MIP* = RE

By Zhengfeng Ji, Anand Natarajan, Thomas Vidick, John Wright, and Henry Yuen

Note from the Research Highlights Co-Chairs:
A Research Highlights paper appearing in *Communications* is usually peer-reviewed prior to publication. The following paper is unusual in that it is still under review. However, the result has generated enormous excitement in the research community, and came strongly nominated by SIGACT, a nomination seconded by external reviewers.

Abstract

The complexity class NP characterizes the collection of computational problems that have *efficiently verifiable solutions*. With the goal of classifying computational problems that seem to lie beyond NP, starting in the 1980s complexity theorists have considered extensions of the notion of efficient verification that allow for the use of *randomness* (the class MA), *interaction* (the class IP), and the possibility to interact with *multiple proofs, or provers* (the class MIP). The study of these extensions led to the celebrated PCP theorem and its applications to hardness of approximation and the design of cryptographic protocols.

In this work, we study a fourth modification to the notion of efficient verification that originates in the study of *quantum entanglement*. We prove the surprising result that every problem that is recursively enumerable, including the Halting problem, can be efficiently verified by a classical probabilistic polynomial-time verifier interacting with two all-powerful but noncommunicating provers sharing entanglement. The result resolves long-standing open problems in the foundations of quantum mechanics (Tsirelson's problem) and operator algebras (Connes' embedding problem).

1. INTERACTIVE PROOF SYSTEMS

An *interactive proof system* is an abstraction that generalizes the intuitively familiar notion of *proof*. Given a formal statement z (e.g., “this graph admits a proper 3-coloring”), a proof π for z is information that enables one to check the validity of z more efficiently than without access to the proof (in this example, π could be an explicit assignment of colors to each vertex of the graph, which is generally easier to verify than to find).

Complexity theory formalizes the notion of proof in a way that emphasizes the role played by the verification procedure. To explain this, first recall that a *language* L is identified with a subset of $\{0, 1\}^*$, the set of all bit strings of any length, that represents all problem instances to which the answer should be “yes.” For example, the language $L = 3\text{-COLORING}$ contains all strings z such that z is the description (according to some prespecified encoding scheme) of a 3-colorable graph G . We say that a language L admits efficiently verifiable

proofs if there exists an algorithm V (formally, a polynomial-time Turing machine) that satisfies the following two properties: (i) for any $z \in L$ there is a string π such that $V(z, \pi)$ returns 1 (we say that V “accepts” the proof π for input z) and (ii) for any $z \notin L$, there is no string π such that $V(z, \pi)$ accepts. Property (i) is generally referred to as the *completeness* property and (ii) is the *soundness*. The set of all languages L such that there exists a verifier satisfying both completeness and soundness is the class NP.

Research in complexity and cryptography in the 1980s and 1990s led to a significant generalization of this notion of “efficiently verifiable proof.” The first modification is to allow *randomized* verification procedures by relaxing (i) and (ii) to *high probability* statements: every $z \in L$ should have a proof π that is accepted *with probability at least c* (the completeness parameter), and for no $z \notin L$ should there be a proof π that is accepted *with probability larger than s* (the soundness parameter). A common setting is to take $c = \frac{2}{3}$ and $s = \frac{1}{3}$; standard amplification techniques reveal that the exact values do not significantly affect the class of languages that admit such proofs, provided that they are chosen within reasonable bounds.

The second modification is to allow *interactive* verification. Informally, instead of receiving a proof string π in its entirety and making a decision based on it, the verification algorithm (called the “verifier”) now communicates with another algorithm called a “prover,” and based on the interaction decides whether $z \in L$. There are no restrictions on the computational power of the prover, whereas the verifier is required to run in polynomial time.^a We let IP (for “Interactive Proofs”) denote the class of languages having interactive, randomized polynomial-time verification procedures.

The third and final modification is to consider interactions with *two* (or more) provers. In this setting, the provers are not allowed to communicate with each other and the verifier may “cross-interrogate” them in order to decide if $z \in L$. The provers are allowed to coordinate a joint strategy ahead of time, but once the protocol begins, they can only interact with the verifier. The condition that the provers cannot communicate with each other is a powerful constraint that can be leveraged by the verifier to detect attempts at making it accept a false claim, that is whenever $z \notin L$.

^a The reader may find the following mental model useful: in an interactive proof, an all-powerful prover is trying to convince a skeptical, but computationally limited, verifier that a string z (known to both) lies in the set L , even when it may be that in fact $z \notin L$. By interactively interrogating the prover, the verifier can reject false claims, i.e. determine with high statistical confidence whether $z \in L$ or not. Importantly, the verifier is allowed to probabilistically and adaptively choose its messages to the prover.

The original version of this paper appeared on the quant-ph arXiv as arXiv:2001.04383.

Work in the 1980s and 1990s in complexity theory has shown that the combination of randomness, interaction, and multiple provers leads to an *exponential* increase in verification power. Concretely, the class of problems that can be verified with all three ingredients together, denoted MIP (for Multiprover Interactive Proofs), was shown to equal the class NEXP¹ of languages that admit exponentially long “traditional” proofs verifiable in exponential time.

2. OUR RESULTS

We now introduce a fourth modification to the class MIP, leading to the class MIP* that is the focus of our work. Informally the class MIP* contains all languages that can be decided by a classical polynomial-time verifier interacting with multiple *quantum* provers sharing *entanglement*. To be clear: compared with MIP, the only difference is that the provers may use entanglement, both in the case when $z \in L$ (completeness) and when $z \notin L$ (soundness).

The study of MIP* is motivated by a long line of works in the foundations of quantum mechanics around the topic of *Bell inequalities*. Informally, a Bell inequality is a linear function that separates two convex sets: on the one hand, the convex set of all families of distributions that can be generated locally by two isolated parties using shared randomness; on the other hand, the convex set of all families of distributions that can be generated locally by two isolated parties using quantum entanglement. (In neither case is there any computational restriction; the only difference is in the kind of shared resource available.) The most famous Bell inequality is the *CHSH inequality*⁵; we will describe another example later, in Section 4.1. The study of Bell inequalities is relevant not only to quantum foundations, where they are a tool to study the nonlocal properties of entanglement, but also in quantum cryptography, where they form the basis for cryptographic protocols, for example, quantum key distribution,⁹ and in the study of entangled states of matter in physics. The connection with complexity theory was first made by Cleve⁶ using the language of two-player games.

The introduction of entanglement in the setting of interactive proofs has interesting consequences for complexity theory; indeed, it is not *a priori* clear how the class MIP* compares to MIP. This is because in general the use of entanglement may increase the odds of the provers to convince the verifier of the validity of any given statement, true or false. Such an increase may render a previously sound proof system unsound, because the provers are able to make the verifier accept false statements. Conversely, it is conceivable that new proof systems may be designed for which the completeness property (valid statements have proofs) holds only when the provers are allowed to use entanglement. As a consequence, MIP* could a priori be smaller, larger, or incomparable to MIP. The only clear inclusions are $IP \subseteq MIP^*$, because if the verifier chooses to interact with a single prover, then entanglement does not play any role, and $MIP^* \subseteq RE$, the class of recursively enumerable languages, that is languages L such that there exists a Turing machine $\mathcal{M}$ such that $z \in L$ if and only if $\mathcal{M}$ halts and accepts on input z . (The inclusion $MIP^* \subseteq RE$ will be justified once we introduce the model more formally. At the moment we only point out that the

reason that no time-bounded upper bound holds in a self-evident manner is because there is no a priori bound on the complexity of near-optimal provers in an MIP* interactive proof system. This is in contrast to MIP where any prover can be represented by its question–answer response function, an exponential-size object.)

In¹³ the first nontrivial lower bound on MIP* was obtained, establishing that $MIP = NEXP \subseteq MIP^*$. This was shown by arguing that the completeness and soundness properties of the proof system of Babai¹ are maintained in the presence of shared entanglement between the provers. Following¹³ a sequence of works established progressively stronger lower bounds, culminating in¹⁸ which showed the inclusion $NEEXP \subseteq MIP^*$, where NEEXP stands for nondeterminist doubly exponential time. Since it is known that $NEXP \subsetneq NEEXP$ it follows that $MIP \neq MIP^*$.

Our main result takes this line of work to its limit to show the exact characterization

$$MIP^* = RE.$$

A complete problem for RE is the Halting problem. Thus a surprising consequence of the equality $MIP^* = RE$ is that there exists a (classical) polynomial-time transformation that takes as input any Turing machine $\mathcal{M}$ and returns the description of a classical randomized polynomial-time (in the description size of $\mathcal{M}$) verification procedure such that the existence of quantum provers sharing entanglement that are accepted by this verification procedure is equivalent to the fact that the Turing machine halts. Since the actions of quantum provers are (in principle) physically realizable, the verification procedure can be interpreted as the specification of a physical experiment that could be used to certify that a Turing machine halts. The reason that this is surprising is because the Halting problem does not refer to any notion of time bound (and, as shown by Turing, is in fact undecidable), whereas the verification procedure is time-bounded. Yet all true statements (halting Turing machines) have valid proofs in this model, while false statements (non-halting Turing machines) do not.

Before proceeding with a description of the main ideas that go in the proof of this result, we highlight some consequences and describe open questions. Our result is motivated by a connection with Tsirelson’s problem from quantum information theory, itself related to Connes’ Embedding Problem in the theory of von Neumann algebras.⁸ In a celebrated sequence of papers, Tsirelson²² initiated the systematic study of quantum correlation sets, for which he gave two natural definitions. The first definition, referred to as the “tensor product model,” constrains isolated systems to be in tensor product with one another: if two parties Alice and Bob are space-time isolated, then any measurement performed by Alice can be modeled using an operator A on Alice’s Hilbert space $\mathcal{H}_A$, while any measurement performed by Bob can be modeled using an operator B on Bob’s Hilbert space $\mathcal{H}_B$; studying the correlations between Alice and Bob then involves forming the tensor product $\mathcal{H}_A \otimes \mathcal{H}_B$ and studying operators such as $A \otimes B$. The second definition, referred to as the “commuting model,” is more

general because it only makes use of a single Hilbert space $\mathcal{H}$ on which both Alice's operators A and Bob's operators B act simultaneously; the constraint of "isolation" is enforced by requiring that A and B commute, $AB = BA$, so that neither operation can have a causal influence on the other.^b The question of equality between the families of distributions that can be generated in either model is known as *Tsirelson's problem*.²³ As already noted by Fritz,¹⁰ the undecidability of MIP* implies that Tsirelson's two models are finitely separated (i.e., there is a constant-size family of distributions that can be realized in the second model but is within a constant distance of any distribution that can be realized in the first model); thus, our result that $\text{RE} \subseteq \text{MIP}^*$ resolves Tsirelson's problem in the negative. Through a sequence of previously known equivalences,¹⁹ we also resolve Connes' Embedding Problem in the negative. As a consequence, our result may lead to the construction of interesting objects in other areas of mathematics. In particular an outstanding open question that may now be within reach is the problem of constructing a finitely presented group that is not sofic, or even not hyperlinear.

3. PROOF OVERVIEW

For simplicity, we focus on the class $\text{MIP}^*(2,1)$, which corresponds to the setting of one-round protocols with two provers sharing entanglement. (A consequence of our results is that this setting has equal verification power to the general setting of polynomially many provers and rounds of interaction.) The verifier in such a protocol can be described as the combination of two procedures: a *question sampling* procedure S that samples a pair of questions (x, y) for the provers according to a distribution μ and a *decision* procedure that takes as input the provers' questions and their respective answers a, b and evaluates a predicate $D(x, y, a, b) \in \{0, 1\}$ to determine the verifier's acceptance or rejection. (In general, both procedures also take the problem instance z as input.)

Our results establish the existence of transformations on *families* of two-prover one-round protocols having certain properties. In order to keep track of efficiency (and ultimately, computability) properties, it is important to have a way to specify such families in a uniform manner. Toward this we introduce the following formalism. A *normal form verifier* is specified by a pair of Turing machines $\mathcal{V} = (S, D)$ that satisfy certain conditions. The Turing machine S (called a *sampler*) takes as input an index $n \in \mathbb{N}$ and returns the description of a procedure that can be used to sample questions (x, y) (this procedure itself obeys a certain format associated with "conditionally linear" distributions, defined in Section 4.3 below). The Turing machine D (called a *decider*) takes as input an index n , questions (x, y) , and answers (a, b) and returns a single-bit decision. The sampling and decision procedures are required to run in time polynomial in the index n . Note that normal form verifiers do not take any input other than the index n ; we explain later in this section how the actual problem instance z is taken into account. Given a normal form verifier $\mathcal{V} = (S, D)$ and an

index $n \in \mathbb{N}$, there is a natural two-prover one-round protocol $\mathcal{V}_n$ associated to it. We let $\text{val}^*(\mathcal{V}_n)$ denote the maximum success probability of quantum provers sharing entanglement in this protocol.

Our main technical result is a *gap-preserving compression* transformation on normal form verifiers. The following theorem presents an informal summary of the properties of this transformation. For a verifier $\mathcal{V}_n$ and a probability $0 \leq p \leq 1$, we let $\mathcal{E}(\mathcal{V}_n, p)$ denote the minimum local dimension of an entangled state shared by provers that succeed in the protocol executed by $\mathcal{V}_n$ with probability at least p .

THEOREM 3.1 (GAP-PRESERVING COMPRESSION). *There exists a polynomial-time Turing machine **Compress** that, when given as input the description of a normal form verifier $\mathcal{V} = (S, D)$, returns the description of another normal form verifier $\mathcal{V}' = (S', D')$ that satisfies the following properties: for all $n \in \mathbb{N}$, letting $N = 2^n$*

1. (**Completeness**) *If $\text{val}^*(\mathcal{V}_N) = 1$ then $\text{val}^*(\mathcal{V}'_N) = 1$.*
2. (**Soundness**) *If $\text{val}^*(\mathcal{V}_N) \leq \frac{1}{2}$ then $\text{val}^*(\mathcal{V}'_N) \leq \frac{1}{2}$.*
3. (**Entanglement**) *$\mathcal{E}(\mathcal{V}'_n, \frac{1}{2}) \geq \max\{\mathcal{E}(\mathcal{V}_N, \frac{1}{2}), 2^{2^{O(n)}}\}$.*

The terminology *compression* is motivated by the fact, implicit in the (informal) statement of the theorem, that the time complexity of the verifier's sampling and decision procedures in the protocol $\mathcal{V}'_n$, which (as required in the definition of a normal form verifier) is polynomial in n , is exponentially smaller than the time complexity of the verifier $\mathcal{V}_N$, which is polynomial in N and thus exponential in n . As such our result can be understood as an efficient delegation procedure for (normal form) interactive proof verifiers. In the next section, we describe how the presence of entanglement between the provers enables such a procedure. First, we sketch how the existence of a Turing machine **Compress** with the properties stated in the theorem implies the inclusion $\text{RE} \subseteq \text{MIP}^*$.

To show this, we give an $\text{MIP}^*(2,1)$ protocol for the Halting problem, which is a complete problem for RE. Precisely, we give a procedure that given a Turing machine $\mathcal{M}$ as input returns the description of a normal form verifier $\mathcal{V}^{\mathcal{M}} = (S^{\mathcal{M}}, D^{\mathcal{M}})$ with the following properties. First, if $\mathcal{M}$ does eventually halt on an empty input tape, then it holds that for all $n \in \mathbb{N}$, $\text{val}^*(\mathcal{V}_n^{\mathcal{M}}) = 1$. Second, if $\mathcal{M}$ does not halt then for all $n \in \mathbb{N}$, $\text{val}^*(\mathcal{V}_n^{\mathcal{M}}) \leq \frac{1}{2}$. It follows that $\mathcal{M} \mapsto \mathcal{V}_1^{\mathcal{M}}$ is a valid $\text{MIP}^*(2,1)$ proof system for the Halting problem.

We describe the procedure that achieves this. Informally, the procedure returns the specification of a verifier $\mathcal{V}^{\mathcal{M}} = (S^{\mathcal{M}}, D^{\mathcal{M}})$ such that $D^{\mathcal{M}}$ proceeds as follows: on input (n, x, y, a, b) it first executes the Turing machine $\mathcal{M}$ for n steps. If $\mathcal{M}$ halts, then $D^{\mathcal{M}}$ accepts. Otherwise, $D^{\mathcal{M}}$ computes the description of the compressed verifier $\mathcal{V}' = (S', D')$ that is the output of **Compress** on input $\mathcal{V}^{\mathcal{M}}$, then executes the decision procedure $D'(n, x, y, a, b)$ and accepts if and only if D' accepts.^c

^b Precisely, commutation implies that the joint distribution of outcomes obtained by performing one measurement and then the other is independent of the order chosen.

^c The fact that the decider $D^{\mathcal{M}}$ can invoke the **Compress** procedure on itself follows from a well-known result in computability theory known as *Kleene's recursion theorem* (also called *Roger's fixed point theorem*).^{15,21}

To show that this procedure achieves the claimed transformation, consider two cases. First, observe that if $\mathcal{M}$ eventually halts in some number of time steps T , then by definition $\text{val}^*(\mathcal{V}_n^{\mathcal{M}}) = 1$ for all $n \geq T$. Using Theorem 3.1 along with an inductive argument it follows that $\text{val}^*(\mathcal{V}_n^{\mathcal{M}}) = 1$ for all $n \geq 1$. Second, if $\mathcal{M}$ never halts, then observe that for any $n \geq 1$ Theorem 3.1 implies two separate lower bounds on the amount of entanglement required to win the protocol $\mathcal{V}_n^{\mathcal{M}}$ with probability at least $\frac{1}{2}$: the dimension is (a) at least $2^{2^{n(n)}}$ and (b) at least the dimension needed to succeed in the protocol $\mathcal{V}_n^{\mathcal{M}}$ with probability at least $\frac{1}{2}$. By induction, it follows that an *infinite* amount of entanglement is needed to succeed in the protocol $\mathcal{V}_n$ with any probability greater than $\frac{1}{2}$. By continuity, a sequence of finite-dimension prover strategies for $\mathcal{V}_n$ cannot lead to a limiting value larger than $\frac{1}{2}$, and $\text{val}^*(\mathcal{V}_n^{\mathcal{M}}) \leq \frac{1}{2}$.

4. USING ENTANGLEMENT TO COMPRESS QUANTUM INTERACTIVE PROOFS

In the language introduced in the previous section, the inclusion $\text{NEXP} \subseteq \text{MIP}$ implies that for any NEXP -complete language L , there is a pair of polynomial-time Turing machines $\mathcal{V} = (\mathcal{S}, \mathcal{D})$ such that the following hold. The Turing machine $\mathcal{S}$ takes as input an integer $n \in \mathbb{N}$ and returns a (probabilistic) circuit that can be used to sample questions $(x, y) \in \mathcal{X} \times \mathcal{Y}$ according to some distribution $\mathcal{S}_n$. The Turing machine $\mathcal{D}$ on input $n \in \mathbb{N}$ as well as an instance $z \in \{0, 1\}^n$ and $(x, y, a, b) \in \mathcal{X} \times \mathcal{Y} \times \mathcal{A} \times \mathcal{B}$ returns a decision bit in $\{0, 1\}$. This proof system is such that whenever $z \in L$ there are “proofs” $f_A : \mathcal{X} \rightarrow \mathcal{A}$ and $f_B : \mathcal{Y} \rightarrow \mathcal{B}$ and such that for all (x, y) in the support of $\mathcal{S}_n$, $\mathcal{D}(1^n, z, x, y, f_A(x), f_B(y)) = 1$, and whenever $z \notin L$ then for any purported “proofs” $f_A : \mathcal{X} \rightarrow \mathcal{A}$ and $f_B : \mathcal{Y} \rightarrow \mathcal{B}$ it holds that

$$\mathbb{E}_{(x,y) \sim \mathcal{S}_n} \mathcal{D}(1^n, z, x, y, f_A(x), f_B(y)) \leq \frac{1}{2}. \quad (1)$$

While the definition of MIP allows for more general proof systems, for example, the sampling of questions may depend on the input z and there may be multiple rounds of interaction, subsequent refinements of the original proof that $\text{NEXP} \subseteq \text{MIP}$ imply that proof systems of the form above are sufficient to capture the entire class.

The maximum of the expression on the left-hand side of (1) can be computed exactly in nondeterministic exponential time by guessing an optimal choice of (f_A, f_B) and evaluating the expression. Therefore, the class MIP does not allow verification of languages beyond NEXP . In particular, Theorem 3.1 is clearly not possible in this model, because by applying it to the verifier for NEXP described in the preceding paragraph one would obtain a polynomial-time verifier with the ability to decide a complete language for NEXP ; however, it is known that $\text{NEXP} \subsetneq \text{NEEXP}$. To go beyond NEXP and prove Theorem 3.1, we must find ways for the verifier to constrain the provers to make use of entanglement so that even more complex computations can be delegated to them.

To prove Theorem 3.1, we will show how the actions of both the sampler $\mathcal{S}$ and the decider $\mathcal{D}$ can be delegated to the provers and their correct execution verified in time polylogarithmic in the sampler and decider’s time complexity. This will enable a polynomial-time verifier to force the

provers to “simulate” the action of $\mathcal{S}$ and $\mathcal{D}$ on inputs of size $N = 2^n$ using polynomial resources, as required by Theorem 3.1. While the techniques for delegating deterministic procedures such as $\mathcal{D}$ are already present, to some extent, in the proof of $\text{MIP} = \text{NEXP}$ (whose implications for delegated computation are well-known, see, e.g., Goldwasser¹²), for the inherently probabilistic procedure $\mathcal{S}$, we are faced with an entirely new challenge: the delegation of *randomness generation*. In particular, it is clearly not reasonable to give the provers access to the random seed used by $\mathcal{S}$, as this would provide them perfect knowledge of both questions and therefore allow them to easily defeat the protocol. In order to ensure that the provers sample from the correct distribution while each obtaining exactly the right amount of information about their respective input, we leverage entanglement in an essential way. In the next section, we give a classic example of how randomness can be certified by using entanglement. In Section 4.2, we introduce an asymptotically efficient entanglement test that builds on the example. In Section 4.3, we describe a class of question distribution that can be delegated (we will say “introspected”) by making use of the entanglement test. In Section 4.5, we explain a final step of parallel repetition, and in Section 4.6, we conclude.

4.1. The Magic Square game

We introduce a two-player game known as the “Magic Square game.” To specify the rules of the game, we first describe a (classical) constraint satisfaction problem that consists of 6 linear equations defined over 9 variables that take values in the binary field $\mathbb{F}_2$. The variables are associated with the cells of a 3×3 grid, as depicted in Figure 1. Five of the equations correspond to the constraint that the sum of the variables in each row and the first two columns must be equal to 0, and the last equation imposes that the sum of the variables in the last column must be equal to 1.

This system of equations is clearly unsatisfiable. Now consider the following game associated to it. In the game, the trusted referee first samples one of the 6 equations uniformly at random and then one of the three variables appearing in the constraint uniformly at random. It asks one prover for an assignment of values to all variables in the chosen constraint, and the other prover for the selected variable. It is not hard to see that no deterministic or randomized strategy can succeed in this game with probability larger than $\frac{17}{18}$, since there are 18 pairs of questions in total and any deterministic or randomized strategy that succeeds with a strictly larger probability is easily seen to imply a satisfying assignment to the Magic Square.

What makes the game “magic” is the surprising fact, first demonstrated by Mermin and Peres,^{16,20} that this game has

Figure 1. The Magic Square game.

x_1	x_2	x_3
x_4	x_5	x_6
x_7	x_8	x_9

a perfect quantum strategy: two noncommunicating players sharing entanglement can win with probability 1. Moreover, this can be achieved by sharing a simple 4-dimensional quantum state (two EPR pairs) and making local measurements on it (i.e., each prover only makes an observation on their local share of the state; see Section 4.2 for the mathematical formalism of quantum strategies). The outcomes of local measurements are not causally related and entanglement does not allow the provers to communicate information. However, their outcome distribution can be correlated: in the Mermin-Peres quantum strategy for the Magic Square game, for any pair of questions selected by the referee the joint distribution of the player's answers happens to be uniform over all pairs of possible answers.^d

The Magic Square game is an example of a Bell inequality (specifically, the inequality is that classical strategies cannot win with probability larger than $\frac{17}{18}$). The game can be interpreted as a two-prover verification protocol for the satisfiability of the underlying system of 9 equations, in which case the protocol is sound with classical provers but unsound with quantum provers. However, for our purposes, this is not a productive observation—we are interested in finding *positive* uses for entanglement. Is there anything useful that can be certified using this game?

We reproduce a crucial observation due to⁷: the *only* way that quantum provers are able to succeed in the Magic Square game beyond what is possible classically is by generating answers that are *intrinsically random*. This is because an intrinsically random strategy (by which we mean that the computation of each answer *must* involve injecting fresh randomness, that was not determined prior to the questions being generated) is the only possibility for evading the classical impossibility (the aforementioned “Bell inequality”). Indeed if the two players always give a deterministic and valid answer to their questions then listing those 18 pairs of answers will lead to a satisfying assignment for the magic square constraints. Since this is not possible it must be that each pair of answers is generated “on the fly,” so that any two answers are correlated in the right way but there is no meaningful correlation between pairs of answers obtained to different pairs of questions (e.g., in different runs of the game). In other words, quantum randomness cannot be “fixed,” and this is what allows quantum provers to succeed in the game.

Based on this observation and with further technical work, it is possible to use the Magic Square game to develop a simple two-prover test (i.e., a small self-contained protocol that can be used as a building block towards a larger MIP* protocol) that constrains the provers to obtain, and report to the verifier, identical uniformly random outcomes. Note that this is stronger than merely allowing the provers to use shared randomness, in the sense that it is *verifiable*: upon seeing either prover's answer, the verifier has the guarantee that it has been chosen near-uniformly at random—the provers cannot bias the randomness in any way (unless they pay a price by failing in the test with positive probability).

^d Not every game has such a perfect quantum strategy. For example, a game in which the second player has to answer the first player's question: this would violate the nonsignaling principle.

While this is already a genuinely quantum possibility that transforms the realm of achievable protocols, it does not suffice for our purposes, for two reasons. First of all the distributions that we aim to sample from, that is the distributions $\mathcal{S}_n$, are more complicated than uniform shared randomness. We discuss this point in detail in Section 4.3. Second, executing as many copies of the Magic Square game as there are bits of randomness required to sample from $\mathcal{S}_n$ would not lead to the complexity savings that we are aiming to achieve. This problem can be solved by employing a more efficient means of randomness and entanglement generation that builds simultaneously on the Magic Square game and on a quantum version of the classical low-degree test, a key component in the proof of $\text{NEXP} \subseteq \text{MIP}$, which we repurpose for this new goal. We explain this in the next section.

4.2. The quantum low-degree test

The quantum low-degree test, first introduced by Natarajan¹⁷ and analyzed by Ji,¹⁴ is one of the core technical components behind the proof of Theorem 3.1. The test provides an efficient means of certifying entanglement (and, as a corollary, randomness generation) between two provers. The test builds upon classical results in the property testing and probabilistically checkable proofs literature for testing low-degree polynomials, a line of work initiated by Gemmel¹¹ building upon the multilinearity test by Babai.¹ In this section, we first state the “quantum” version of these results and then explain its use for delegating the sampling procedure.

To state the quantum low-degree test in sufficient technical depth, we introduce the standard formalism used to model quantum strategies. A strategy for two quantum provers (in a one-round protocol) consists of (i) a quantum state shared by the provers, which is represented by a unit vector $|\psi\rangle \in \mathcal{H}_A \otimes \mathcal{H}_B$ where $\mathcal{H}_A$ and $\mathcal{H}_B$ are finite-dimensional Hilbert spaces associated with each prover (e.g., if $|\psi\rangle$ consists of n EPR pairs then $\mathcal{H}_A = \mathcal{H}_B = \mathbb{C}^{2^n}$) and (ii) for each question to a prover, say $x \in \mathcal{X}$ to the first prover, a measurement $\{A_a^x\}_{a \in \mathcal{A}}$ that the prover performs on their share of the quantum state to produce an answer $a \in \mathcal{A}$. Mathematically each A_a^x is a positive semidefinite operator acting on $\mathcal{H}_x$ such that $\sum_{a \in \mathcal{A}} A_a^x = I$. The measurement rule states that two provers modeled in this way generate answers (a, b) in response to questions (x, y) with probability

$$p(a, b | x, y) = \langle \psi | A_a^x \otimes B_b^y | \psi \rangle.$$

Clearly two strategies generate the same distributions if they are unitarily equivalent, that is, exchanging $|\psi\rangle$ for $U_A \otimes U_B |\psi\rangle$ for unitaries U_A on $\mathcal{H}_A$ and U_B on $\mathcal{H}_B$ and conjugating all operators by U_A or U_B leads to the same strategy. So any characterization of near-optimal strategies will only be “up to local unitaries.” More generally, the provers may have access to a larger space that is not directly used in their strategy. For this reason, a characterization of near-optimal quantum strategies is generally formulated “up to local isometries,” where a local isometry is a linear map $\phi_A : \mathcal{H}_A \rightarrow \mathcal{H}_{A'} \otimes \mathcal{H}_{A''}$ (resp. $\phi_B : \mathcal{H}_B \rightarrow \mathcal{H}_{B'} \otimes \mathcal{H}_{B''}$).

The quantum low-degree test comes in the form of a *rigidity* result, which guarantees that any near-optimal strategy

in a certain two-prover protocol is essentially unique—up to local isometries. The test is parametrized by a tuple $\text{qldparams} = (q, m, d)$ where q is a field size, m a number of variables, and d a degree satisfying some conditions that we omit from this abstract. We denote the associated two-prover one-round verifier $\mathcal{V}_{\text{qldparams}}^{\text{PAULI}}$. In the associated protocol, whose exact description we also omit, the provers are expected to (i) measure $2^m \log q$ shared EPR pairs in a certain basis (either the computational basis or the Hadamard basis, which is its Fourier transform), obtaining an outcome $a \in \mathbb{F}_q^{2^m}$; (ii) interpolate a polynomial $g_a: \mathbb{F}_q^m \rightarrow \mathbb{F}_q$ of individual degree at most d ; and (iii) return the restriction of g_a to a line or point in $\mathbb{F}_q^m$ that is provided to them as their question. This is almost exactly the same as the classic test from Babai¹ and Gemmel¹¹; the “quantum” part of the test lies in the fact that the specification a for the polynomial g_a used by the provers should be obtained as a result of a measurement on an entangled state. This measurement can be required to be made in different, incompatible bases (depending on the question), and thus its outcome cannot be fixed a priori, before the protocol starts. (In contrast, in the classical case, it is assumed that a is a fixed string on which the provers pre-agree.) Tests based on the Magic Square game are performed in addition to the above to enforce that the provers make their measurements in the right bases.

THEOREM 4.1. *There exists a function*

$$\delta_{\text{QLD}}(\varepsilon, q, m, d) = a(md)^a(\varepsilon^b + q^{-b} + 2^{-bmd})$$

for universal constants $a \geq 1$ and $0 < b < 1$ such that the following holds. For all admissible parameter tuples $\text{qldparams} = (q, m, d)$ and for all strategies $\mathcal{S} = (|\psi\rangle, A, B)$ for $\mathcal{V}_{\text{qldparams}}^{\text{PAULI}}$ that succeed with probability at least $1 - \varepsilon$, there exist local isometries $\phi_A: \mathcal{H}_A \rightarrow \mathcal{H}_{A'} \otimes \mathcal{H}_{A''}$, $\phi_B: \mathcal{H}_B \rightarrow \mathcal{H}_{B'} \otimes \mathcal{H}_{B''}$ and a state $|AUX\rangle \in \mathcal{H}_{A'} \otimes \mathcal{H}_{B'}$ such that

$$\| |\phi_A \otimes \phi_B | \psi \rangle - |AUX\rangle \otimes |EPR\rangle^{\otimes M \log q} \| \leq \delta_{\text{QLD}}(\varepsilon, q, m, d),$$

where $|EPR\rangle$ denotes an EPR pair. Moreover, letting $\tilde{A}_a^x = \phi_A^x A_a^x \phi_A^{\dagger}$ and $\tilde{B}_b^y = \phi_B^y B_b^y \phi_B^{\dagger}$ we have for $W \in \{X, Z\}$

$$\begin{aligned} \tilde{A}_u^W \otimes I_{B'B''} &\approx_{\delta_{\text{QLD}}} (\sigma_u^W)_{A''} \otimes I_{A'B'B''} \\ I_{A'A''} \otimes \tilde{B}_u^W &\approx_{\delta_{\text{QLD}}} I_{A'A''B'} \otimes (\sigma_u^W)_{B''}. \end{aligned}$$

In the theorem statement, the approximation $\approx_{\delta_{\text{QLD}}}$ is measured in a norm that depends on the state shared by the provers and is appropriate for arguing that the two measurement operators on the left and right of the approximation sign lead to similar outcome distributions, even when some other operator is applied by the other prover. The measurement operators $\tilde{A}_u^W$ and $\tilde{B}_u^W$ on the left-hand side are the provers’ measurement operators associated with a designated pair of questions $W = X, Z$ in the protocol. The operators σ_u^W that appear on the right-hand side denote tensor products of single-qubit Pauli measurements in the basis W , which is the computational basis in case $W = Z$ and the Hadamard basis in case $W = X$. Here u is an element of $\mathbb{F}_q^{2^m}$, which can be interpreted as an $(2^m \log q)$ -bit string.

Informally, the theorem guarantees that any prover strategy that achieves a high probability of success in the test is “locally equivalent” to a strategy that consists in measuring $2^m \log q$ EPR pairs in the appropriate basis, which is a strategy of the “honest” form described above. Crucially, the number of EPR pairs tested is exponential in the verifier complexity, which scales polynomially in $\log q, m$, and d .

We turn our attention back to the problem of delegating the sampling from the distribution $\mathcal{S}_n$ to the provers. Referring to the provers producing their own questions, we call this procedure “introspection.” In general, it is not known how to introspect arbitrary unstructured distributions $\mathcal{S}_n$. The verifier that underlies the protocol from Babai¹ chooses most of its question from the “line-point” distribution $\mathcal{S}_n$, which is the distribution over pairs (x, y) such that x is the description of a uniformly random line $\ell \in \mathbb{F}_q^m$ and y is a uniformly random point on ℓ . It is natural to first consider introspection of this distribution. This is done by Natarajan¹⁸ to “compress” the protocol from Babai¹ and thereby show the inclusion $\text{NEEXP} \subseteq \text{MIP}^*$. The compressed protocol relies on the test from Theorem 4.1 to efficiently sample its questions to the provers; the randomness used by the provers for sampling is based on their measurement outcomes in the test, which are bitstrings of length $2^m \log q$.

Unfortunately the distribution used to sample the questions in the compressed protocol no longer has such a nice form as the line-point distribution. In particular, it includes the question distribution from the quantum low-degree test, which itself incorporates the Magic Square game question distribution. Since our goal is to compress protocols iteratively (recall the end of Section 3), it is essential to identify a class of distributions that is sufficiently general and “closed under introspection” in the sense that any distribution from the family can be tested using a distribution from the same family with reduced randomness complexity. For this, we introduce the class of *conditionally linear distributions*, which generalizes the low-degree test distribution. We show that conditionally linear distributions can be “introspected” using conditionally linear distributions only, enabling recursive introspection. (As we will see later, other closure properties of conditionally linear distributions, such as taking direct products, play an important role as well.) We describe this family of distributions next.

4.3. Conditionally linear distributions

Fix a vector space V that is identified with $\mathbb{F}_q^m$, for a finite field $\mathbb{F}_q$ and integer m . Informally, a function L on V is *conditionally linear* (CL for short) if it can be evaluated by a procedure that takes the following form: (i) read a substring $z^{(1)}$ of z ; (ii) evaluate a linear function L_1 on $z^{(1)}$; and (iii) repeat steps (i) and (ii) with the remaining coordinates $z \setminus z^{(1)}$, such that the next steps are allowed to depend in an arbitrary way on $L_1(z^{(1)})$ but not directly on $z^{(1)}$ itself. What distinguishes a function of this form from an arbitrary function is that we restrict the number of iterations of (i)—(ii) to a constant number (at most 9, in our case). (One may also think of CL functions as “adaptively linear” functions, where the number of “levels” of adaptivity is the number of iterations of (i)—(ii).) A distribution μ over pairs $(x, y) \in V \times V$ is called conditionally linear

if it is the image under a pair of conditionally linear functions $L^A, L^B : V \rightarrow V$ of the uniform distribution on V , that is, $(x, y) \sim (L^A(z), L^B(z))$ for uniformly random $z \in V$.

An important class of CL distributions are low-degree test distributions, which are distributions over question pairs (x, y) where y is a randomly chosen affine subspace of $\mathbb{F}_q^m$ and x is a uniformly random point on y . We explain this for the case where the random subspace y is one-dimensional (i.e., a line). Let $V = V_x \otimes V_v$ where $V_x = V_v = \mathbb{F}_q^m$. Let L^A be the projection onto V_x (i.e., it maps $(x, v) \rightarrow (x, 0)$ where $x \in V_x$ and $v \in V_v$). Define $L^B : V \rightarrow V$ as the map $(x, v) \mapsto (L_v^{L^N}(x), v)$ where $L_v^{L^N} : V_x \rightarrow V_x$ is a linear map that, for every $v \in V_v$, projects onto a complementary subspace to the one-dimensional subspace of V_x spanned by v (one can think of this as an “orthogonal subspace” to the span of $\{v\}$). L^B is conditionally linear because it can be seen as first reading the substring $v \in V_v$ (which can be interpreted as specifying the *direction* of a line), and then applying a linear map $L_v^{L^N}$ to $x \in V_x$ (which can be interpreted as specifying a canonical point on the line $\ell = \{x + tv : t \in \mathbb{F}\}$). It is not hard to see that the distribution of $(L^A(z), L^B(z))$ for z uniform in V is identical (up to relabeling) to the low-degree test distribution (x, ℓ) where ℓ is a uniformly random affine line in $\mathbb{F}_q^m$, and x is a uniformly random point on ℓ .

We show that any CL distribution μ , associated with a pair of CL functions (L^A, L^B) over a linear space $V = \mathbb{F}_q^m$, can be “introspected” using a CL distribution that is “exponentially smaller” than the initial distribution. Slightly more formally, to any CL distribution μ , we associate a two-prover test in which questions from the verifier are sampled from a CL distribution μ' over $\mathbb{F}_q^{m'}$ for some $m' = \text{poly} \log(m)$ and such that in any successful strategy, when the provers are queried on a special question labeled **INTRO** they must respond with a pair (x, y) that is approximately distributed according to μ . (The test allows us to do more: it allows us to conclude how the provers obtained (x, y) —by measuring shared EPR pairs in a specific basis—and this will be important when using the test as part of a larger protocol that involves other checks.) Crucially for us, the distribution μ' only depends on a size parameter associated with (L^A, L^B) (essentially, the integer m together with the number of “levels” of adaptivity of L^A and L^B), but not on any other structural property of (L^A, L^B) . Only the decision predicate for the associated protocol depends on the entire description of (L^A, L^B) .

We say a few words about the design of μ' and the associated test, which borrow heavily from Natarajan.¹⁸ Building on the quantum low-degree test introduced in Section 4.2, we already know how a verifier can force a pair of provers to measure m EPR pairs in either the computational or Hadamard basis and report the (necessarily identical) outcome z obtained, all the while using questions of length polylogarithmic in m only. The added difficulty is to ensure that a prover obtains, and returns, precisely the information about z that is contained in $L^A(z)$ (resp. $L^B(z)$), and not more. A simple example is the line-point distribution described earlier: there, the idea to ensure that, for example, the “point” prover only obtains the first component, x of $(x, v) \in V_x \otimes V_v$, the verifier demands that the “point” prover measures their qubits associated with the space V_v

in the Hadamard, instead of computational, basis; due to the uncertainty principle, this has the effect of “erasing” the outcome in the computational basis. The case of the “line” prover is a little more complex: the goal is to ensure that, conditioned on the specification of the line ℓ received by the “line” prover, the point x received by the “point” prover is uniformly random within ℓ . This was shown to be possible in [18].

4.4. Answer reduction

The previous section sketches how we are able to use entanglement testing techniques to delegate the sampling of questions directly to the provers, with an exponential savings in the verifier’s effort. Let $\mathcal{V}_n^{\text{INTRO}}$ denote the resulting “question-reduced” verifier. However, the players still respond with poly(N)-length answers, which the verifier has to check satisfies the decision predicate of the original protocol. We explain how to obtain a similar exponential savings in the verification of the answers.

For this, we use probabilistically checkable proofs (PCPs) as follows. The verifier in $\mathcal{V}_n^{\text{AR}}$ samples questions as $\mathcal{V}_n^{\text{INTRO}}$ would and sends them to the players. Instead of receiving the introspected questions and answers (x, y, a, b) for the original verifier $\mathcal{V}_n$ and running the decision procedure $\mathcal{D}(N, x, y, a, b)$, the verifier instead asks the provers to compute a PCP Π for the statement that the original decider $\mathcal{D}$ accepts the input (N, x, y, a, b) in time $T = \text{poly}(N)$. The verifier then samples additional questions for the provers that ask them to return specific entries of the proof Π . Finally, upon receipt of the provers’ answers, the verifier executes the PCP verification procedure. Because of the efficiency of the PCP, both the sampling of the additional questions and the decision procedure can be executed in time poly(n).^e

This sketch presents some difficulties. A first difficulty is that in general no prover by themselves has access to the entire input (N, x, y, a, b) to $\mathcal{D}$, so no prover can compute the entire proof Π . This issue is addressed using oracularization, which is a classical technique and so we omit the details (there are some subtleties specific to the quantum case). A second difficulty is that a black-box application of an existing PCP, as done by Natarajan,¹⁸ results in a question distribution for $\mathcal{V}_n^{\text{AR}}$ (i.e., the sampling of the proof locations to be queried) that is rather complex—and in particular, it may no longer fall within the framework of CL distributions for which we can do introspection. To avoid this, we design a bespoke PCP based on the classical MIP for NEXP (in particular, we borrow and adapt techniques from Ben-Sasson^{3,4}). Two essential properties for us are that (i) the PCP proof is a collection of several low-degree polynomials, two of which are low-degree encodings of each prover’s uncompressed answers, and (ii) verifying the proof only requires (a) running low-degree tests, (b) querying all polynomials at a uniformly random point, and (c) performing simple consistency checks. Property (i) allows us to eliminate the extra layer of encoding by Natarajan,¹⁸ who had to consider a PCP

^e This idea is inspired by the technique of composition in the PCP literature, in which the complexity of a verification procedure can be reduced by composing a proof system (often a PCP itself) with another PCP.

of proximity for a circuit applied to the low-degree encodings of the provers' uncompressed answers. Property (ii) allows us to ensure the question distribution employed by the verifier remains conditionally linear.

4.5. Parallel repetition

The combined steps of question reduction (via introspection) and answer reduction (via PCP composition) result in a protocol whose verifier $\mathcal{V}_n^{\text{AR}}$ has complexity poly(n). Furthermore, the property of having a perfect quantum strategy is preserved by the reduction. Unfortunately, the sequence of transformations incurs a loss in the soundness parameters: if $\text{val}^*(\mathcal{V}_n) \leq \frac{1}{2}$, then we can only establish that $\text{val}^*(\mathcal{V}_n^{\text{AR}}) \leq 1 - C$ for some positive constant $C < \frac{1}{2}$ (we call C the soundness gap). Such a loss would prevent us from recursively applying the compression procedure Compress an arbitrary number of times, which is needed to obtain the desired complexity results for MIP*.

To overcome this, we need a final transformation to restore the soundness gap after answer reduction to a constant larger than $\frac{1}{2}$. To achieve this, we use the technique of parallel repetition. The parallel repetition of a two-prover one-round protocol $\mathcal{G}$ is another protocol $\mathcal{G}^k$, for some number of repetitions k , which consists of executing k independent and simultaneous instances of $\mathcal{G}$ and accepting if and only if all k instances accept. Intuitively, parallel repetition is meant to decrease the prover's maximum success probability in a protocol $\mathcal{G}$ exponentially fast in k , provided $\text{val}^*(\mathcal{G}) < 1$ to begin with. However, it is an open question of whether this is generally true for the entangled value val^* .

Nevertheless, some variants of parallel repetition are known to achieve exponential amplification. We use a variant called “anchored parallel repetition” introduced by Bavarian.² This allows us to devise a transformation that efficiently amplifies the soundness gap to a constant. The resulting protocol $\mathcal{V}_n^{\text{REP}}$ has the property that if $\text{val}^*(\mathcal{V}_n^{\text{AR}}) = 1$, then $\text{val}^*(\mathcal{V}_n^{\text{REP}}) = 1$ (and moreover this is achieved using a commuting and consistent strategy), whereas if $\text{val}^*(\mathcal{V}_n^{\text{AR}}) \leq 1 - C$ for some universal constant $C > 0$, then $\text{val}^*(\mathcal{V}_n^{\text{REP}}) \leq \frac{1}{2}$. Furthermore, we have the additional property, essential for us, that good strategies in $\mathcal{V}_n^{\text{REP}}$ require as much entanglement as good strategies in $\mathcal{V}_n^{\text{AR}}$ (which in turn require as much entanglement as good strategies in $\mathcal{V}_n$). The complexity of the verifier in $\mathcal{V}_n^{\text{REP}}$ remains poly(n).

The anchored parallel repetition procedure, when applied to a normal form verifier, also yields a normal form verifier: this is because the direct product of CL distributions is still conditionally linear.

4.6. Putting it all together

This completes the overview of the transformations performed by the compression procedure Compress of Theorem 3.1. To summarize, given an input normal form verifier $\mathcal{V}$, question and answer reduction are applied to obtain $\mathcal{V}^{\text{AR}}$, and anchored parallel repetition is applied to obtain $\mathcal{V}^{\text{REP}}$, which is returned by the compression procedure. Each of these transformations preserves completeness (including the commuting and consistent properties of a value-1 strategy) as well as the entanglement requirements of each protocol; moreover, the overall transformation preserves soundness.

References

- Babai, L., Fortnow, L., Lund, C. Non-deterministic exponential time has two-prover interactive protocols. *Comput. Complex.* 1, 1 (1991), 3–40.
- Bavarian, M., Vidick, T., Yuen, H. Hardness amplification for entangled games via anchoring. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, ACM, New York, NY, USA, 2017, 303–316.
- Ben-Sasson, E., Goldreich, O., Harsha, P., Sudan, M., Vadhan, S. Robust PCPs of proximity, shorter PCPs, and applications to coding. *SIAM J. Comput.* 36, 4 (2006), 889–974.
- Ben-Sasson, E., Sudan, M. Simple PCPs with poly-log rate and query complexity. In *Proceedings of the Thirty-seventh Annual ACM Symposium on Theory of Computing*, ACM, New York, NY, USA, 2005, 266–275.
- Clauser, J.F., Horne, M.A., Shimony, A., Holt, R.A. Proposed experiment to test local hidden-variable theories. *Phys. Rev. Lett.* 23, 15 (1969), 880.
- Cleve, R., Hoyer, P., Toner, B., Watrous, J. Consequences and limits of nonlocal strategies. In *Proceedings. 19th IEEE Annual Conference on Computational Complexity*, 2004, IEEE, Washington, DC, USA, 2004, 236–249.
- Colbeck, R. Quantum and relativistic protocols for secure multi-party computation. Ph. D. Thesis, 2009.
- Connes, A. Classification of injective factors cases II₁, II_∞, III_λ, λ ≠ 1. *Ann. Math.* 104 (1976), 73–115.
- Ekert, A.K. Quantum cryptography based on Bell's theorem. *Phys. Rev. Lett.* 67, 6 (1991), 661.
- Fritz, T., Netzer, T., Thom, A. Can you compute the operator norm? *Proc. Am. Math. Soc.* 142, 12 (2014), 4265–4276.
- Gemmel, P., Lipton, R., Rubinfeld, R., Sudan, M., Wigderson, A. Self testing/correcting for polynomials and for approximate functions. In *Proceedings of the 23rd STOC*. ACM, New York, NY, USA, 1991, 32–42.
- Goldwasser, S., Kalai, Y.T., Rothblum, G.N. Delegating computation: interactive proofs for muggles. *J. ACM* 62, 4 (2015), 1–64.
- Ito, T., Vidick, T. A multi-prover interactive proof for NEXP sound against entangled provers. In *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*, IEEE, Washington, DC, USA, 2012, 243–252.
- Ji, Z., Natarajan, A., Vidick, T., Wright, J., Yuen, H. Quantum soundness of the classical low individual degree test. arXiv preprint arXiv:2009.12982 (2020).
- Kleene, S.C. Introduction to Metamathematics. *J. Symb. Log.* 19, 3 (1954), 215–216.
- Mermin, D. Simple unified form for the major no-hidden-variables theorems. *Phys. Rev. Lett.* 65, 27 (1990), 3373.
- Natarajan, A., Vidick, T. Two-player entangled games are NP-hard. In *Proceedings of the 33rd Computational Complexity Conference*, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 2018, 20.
- Natarajan, A., Wright, J. NEXP ⊆ MIP*. arXiv preprint arXiv:1904.05870v3 (2019).
- Ozawa, N. About the Connes embedding conjecture. *Jpn. J. Math.* 8, 1 (2013), 147–183.
- Peres, A. Incompatible results of quantum measurements. *Phys. Lett. A* 151, 3–4 (1990), 107–108.
- Rogers Jr., H. *Theory of Recursive Functions and Effective Computability*. MIT Press, Cambridge, MA, USA, 1987.
- Tsirelson, B.S. Some results and problems on quantum Bell-type inequalities. *Hadronic J. Suppl.* 8, 4 (1993), 329–345.
- Tsirelson, B.S. Bell inequalities and operator algebras, 2006. Problem statement from website of open problems at TU Braunschweig, 2006. Available at <http://web.archive.org/web/20090414083019/http://www.imaph.tu-bs.de/qi/problems/33.html>.

Zhengfeng Ji (zhengfeng.ji@uts.edu.au), School of Computer Science, University of Technology Sydney, Sydney, Australia.

Anand Natarajan (anandn@caltech.edu), Department of EECS, Massachusetts Institute of Technology, Cambridge, MA, USA.

Thomas Vidick (vidick@caltech.edu), Department of Computing and

Mathematical Sciences, California Institute of Technology, Pasadena, CA, USA.

John Wright (wright@cs.utexas.edu), Department of Computer Science, University of Texas at Austin, Austin, TX, USA.

Henry Yuen (hyuen@cs.columbia.edu), Department of Computer Science, Columbia University, New York, NY, USA.