MIP* = RE: a Lean blueprint

The MIP* = RE formalization project

September 11, 2026

1 Introduction

This blueprint organizes a Lean formalization of the theorem MIP* = RE of Ji, Natarajan, Vidick, Wright and Yuen [1]. The main result is that there is a computable map from Turing machines to nonlocal games that sends halting machines to games of (quantum) value 1 and non-halting machines to games of value at most $\frac{1}{2}$; in particular, approximating the quantum value of a nonlocal game is an uncomputable problem. Through the work of Fritz [15], Junge et al. [14] and Ozawa [16], this refutes Tsirelson's problem and Connes' Embedding Problem.

The main theorem is stated in the self-contained (other than importing Mathlib) file `HaltingGameValue.lean`. Every definition it contains (POVMs, synchronous games, game descriptions) is elementary linear algebra and computability. The goal of this projects are to (a) complete a formalization of this theorem and (b) do it in such a way that the required effort benefits the community down-the-line, e.g. by making the proof modular, stating and proving as many facts as possible in a manner that is reusable, etc. Finally, (c) we expect multiple simplifications and clarifications to the proof to accompany the effort.

This initial blueprint is a coarse sketch. It maps out the main steps of the proof, in the attempt to modularize it and provide a diversity of entrypoints to the formalization effort. The most important definitions are complete, but most theorems appear only in "headline" form, with the decomposition into formalizable lemmas left as a project task.

1.1 Overview of the proof

The inclusion MIP* $\subseteq$ RE is elementary: the value of a game can be approximated from below by enumerating tensor-product strategies (Section 3.11). The substance is RE $\subseteq$ MIP*, which is proved by a reduction from the halting problem. The central ingredient in the proof is a *gap-preserving compression* procedure for "normal form" verifiers (Section 6.6): a polynomial-time transformation that maps a verifier V to a verifier $\mathsf{V}^{\textsc{Compr}}$ which, at index n, simulates the exponentially larger game V_{2^n}, preserving the value-1 property and the property of having value at most $\frac{1}{2}$. Given compression, an abstract fixed-point argument (Lin's *compressibility criterion*, Section 4.3, applied via Kleene's recursion theorem) produces the halting reduction.

Compression itself is assembled from three transformations, each described in Section 6: *question reduction* by introspection (the players sample their own questions, certified by a Pauli-basis self-test built on the quantum low individual degree test), *answer reduction* by PCP composition (the players prove, via a probabilistically checkable proof, that their answers would have been accepted), and *gap amplification* by parallel repetition (anchored in [1], direct here). Each of these steps has one central ingredient that can be formalized in an independent manner: the low individual degree test, the PCP theorem (in a specific form), and a parallel repetition theorem (Section 5; the direct, value-form theorems are formalized).

1.2 Entry points

Contributors with different scientific backgrounds can enter the project in different places, largely independently.

Quantum information. The foundations chapter (Section 2) and the required results on self-testing: the Magic Square game (Section 3.4), the Gowers–Hatami theorem (Section 3.1), the orthonormalization lemma (Section 3.3), and the low individual degree test (Section 3.6). These are self-contained statements about states and measurements, formalizable without any complexity theory.

Complexity and computability theory. The computability toolkit (Section 4.2), the abstract recursive compression lemma (Section 4.3), succinct descriptions and the Cook–Levin machinery feeding answer reduction (Section 3.8). These are classical statements with no quantum content, and several partially exist in Mathlib.

Operator algebras. The synchronous framework of Section 2, the bridge between synchronous and tensor-product values (Section 3.2), and the downstream consequences (Section 8), which connect the main theorem to Tsirelson's problem and Connes' Embedding Problem.

In addition, contributors can participate at different levels. Contributors with a high degree of Lean expertise may want to focus on the foundations and concentrate on those elements of the proof that are of broader interest and may later be upstreamed to Mathlib, Physlib, CSLib, etc. For example, the Gowers Hatami theorem, general facts about Turing machines and computability, the PCP theorem, formalization of quantum games and quantum game values, etc.

1.3 Departures from the paper

The formalization is not intended to follow [1] line by line. Since the initial publication of the paper a number of simplifications have appeared in the literature. In particular, we plan to incorporate the following.

Synchronous strategies throughout. The paper works with bipartite tensor-product strategies $(|\psi\rangle, A, B)$ on $\mathcal{H}_A \otimes \mathcal{H}_B$. We instead take *synchronous* strategies — a single family of measurements $\{M_a^x\}$ on one space, evaluated against the tracial state — as the basic object, and define the synchronous value val^s (Definition 2.10). This eliminates the duplication of Hilbert spaces, states, and consistency bookkeeping that pervades the paper's soundness analyses, and it matches the main Lean statement. The cost is a single bridging step: the almost-synchronous correlations theorem (Section 3.2) shows that for synchronous games the two values agree closely enough to transport completeness and soundness. All games constructed in the proof are synchronous (or are made so by symmetrization), so this is a genuine simplification.

Projective strategies via orthonormalization. Rather than invoking Naimark dilation to replace POVMs by projective measurements (which enlarges the space and interacts poorly with the tracial framework), we use the orthonormalization lemma (Section 3.3) to replace almost-projective POVMs by nearby projective measurements on the *same* space. All soundness analyses should be carried out for projective strategies from the start.

Computable rather than polynomial-time, where possible. The final undecidability statement only needs the halting reduction to be *computable*; polynomial-time bounds matter only inside the compression recursion, where the runtime of Compress must be smaller than the succinctness it exploits (see the discussion in [8]). The blueprint keeps time bounds where

they are load-bearing (normal form verifiers, compression) and drops them everywhere else. In particular the main theorem (Section 7) is stated with a computable reduction.

Abstract compression, in value form. The fixed-point argument that turns compression into a halting reduction is stated once, abstractly, as Lin's compressibility criterion [10] (Section 4.3): a compression procedure preserving "perfect PCC strategy" and "value at most $\frac{1}{2}$", together with the semidecidability of "value above $\frac{1}{2}$", yields the reduction. This isolates all uses of Kleene's recursion theorem in one quantum-free lemma, and it removes the entanglement bookkeeping of [1] from the pipeline: no transformation needs an entanglement clause, and gap amplification uses direct rather than anchored parallel repetition, for which complete formalizations exist (Section 5.1). The recursive compression lemma of Marks–Nezhadi–Yuen [8], instantiated with f = entanglement requirement, is retained as the entanglement-form alternative (Remark 6.8).

Other candidate simplifications (to be decided). Using de la Salle's quantitative improvements to stability (Gowers–Hatami) where they shorten proofs; other simplifications that are present in the follow-up proof of $\mathrm{MIP}^{\mathrm{co}} = \mathrm{coRE}$ [9, 10].

1.4 Conventions

Every definition and theorem carries a `\label`, and `\uses` annotations record the dependency graph rendered on the project web page. Lean declaration names (`\lean` tags) are deliberately absent at this stage; they will be added as the corresponding declarations are created. Statements in Sections 3–5 are external results: the project may initially axiomatize them (each section records the literature source and comments on formalization), while statements in Section 6 are proof obligations of the project. Numerical constants in headline statements (levels, error exponents) are copied from [1] and should be treated as provisional until the corresponding section is worked out in detail.

2 Foundations

This chapter fixes the objects that everything else refers to: games, strategies and values (in the synchronous framework, see Section 1.3), the computability conventions, and normal form verifiers. Definitions here are meant to be final; they are the first formalization targets of the project.

Statements marked ↑ QuantumLib are general-purpose quantum information content: they are intended to be developed in, or contributed to, the QuantumLib (Lean-QuantumInfo) library rather than to remain specific to this project. The badge marks the general-purpose layer — games, strategies, values, and the rigidity theorems of Section 3 — as opposed to project-specific machinery such as normal form verifiers and the game transformations.

2.1 Games, strategies, and values

2.1.1 Games

Definition 2.1 (Nonlocal game)**.** ↑ *QuantumLib* *A two-player one-round game $\mathfrak{G}$ is a tuple $(\mathcal{X}, \mathcal{Y}, \mathcal{A}, \mathcal{B}, \mu, D)$ where $\mathcal{X}, \mathcal{Y}$ are finite question alphabets, $\mathcal{A}, \mathcal{B}$ are finite answer alphabets, μ is a probability distribution on $\mathcal{X} \times \mathcal{Y}$, and $D : \mathcal{X} \times \mathcal{Y} \times \mathcal{A} \times \mathcal{B} \to \{0, 1\}$ is the decision predicate (1 = accept).*

Definition 2.2 (Synchronous game)**.** ↑ QuantumLib *A game is* synchronous *if* $\mathcal{X} = \mathcal{Y}$, $\mathcal{A} = \mathcal{B}$, *and unequal answers to equal questions are rejected:* $D(x, x, a, b) = 0$ *whenever* $a \neq b$. *We write a synchronous game as* $\mathfrak{G} = (\mathcal{X}, \mathcal{A}, \mu, D)$.

2.1.2 Strategies

Definition 2.3 (Tensor-product strategy)**.** ↑ QuantumLib *A* tensor-product strategy *for* $\mathfrak{G}$ *consists of finite-dimensional Hilbert spaces* $\mathcal{H}_{\mathrm{A}}$, $\mathcal{H}_{\mathrm{B}}$, *a unit vector* $|\psi\rangle \in \mathcal{H}_{\mathrm{A}} \otimes \mathcal{H}_{\mathrm{B}}$, *and projective measurements (PVMs)* $\{A_a^x\}$ *on* $\mathcal{H}_{\mathrm{A}}$ *and* $\{B_b^y\}$ *on* $\mathcal{H}_{\mathrm{B}}$, *one for each* x *(resp.* y*): each* A_a^x *is an orthogonal projection and* $\sum_a A_a^x = I$.

Outcome probabilities are computed using the Born rule: the players answer (a, b) *to questions* (x, y) *with probability* $\langle\psi|A_a^x \otimes B_b^y|\psi\rangle$. *Restricting to projective measurements follows [1]; by Naimark dilation, allowing general POVMs would define the same quantum value.*

Definition 2.4 (Synchronous strategy)**.** ↑ QuantumLib *A* synchronous strategy *for a synchronous game* $\mathfrak{G} = (\mathcal{X}, \mathcal{A}, \mu, D)$ *is a dimension* $d \geq 1$ *together with a family of projective measurements (PVMs)* $\{M_a^x\}_{a \in \mathcal{A}}$ *on* $\mathbb{C}^d$, *one for each* $x \in \mathcal{X}$*: each* M_a^x *is an orthogonal projection and* $\sum_a M_a^x = I$.

Outcome probabilities are computed with the tracial state $\tau(M) = \mathrm{tr}(M)/d$*: the players answer* (a, b) *to questions* (x, y) *with probability* $\tau(M_a^x M_b^y)$.

Definition 2.5 (Commuting strategy)**.** ↑ QuantumLib *A* commuting strategy *for a synchronous game* $\mathfrak{G} = (\mathcal{X}, \mathcal{A}, \mu, D)$ *is a unital* C^**-algebra* $\mathscr{A}$ *with a tracial state* τ *(a positive linear functional with* $\tau(1) = 1$ *and* $\tau(ab) = \tau(ba)$ *for all* $a, b \in \mathscr{A}$*), together with a family of projective measurements* $\{M_a^x\}_{a \in \mathcal{A}}$ *in* $\mathscr{A}$, *one for each* $x \in \mathcal{X}$*: each* M_a^x *is a self-adjoint idempotent and* $\sum_a M_a^x = 1$.

Outcome probabilities are computed with τ*: the players answer* (a, b) *to questions* (x, y) *with probability* $\tau(M_a^x M_b^y)$, *a nonnegative real number since* $\tau(M_a^x M_b^y) = \tau(M_a^x M_b^y M_a^x)$.

Remark 2.6. *A synchronous strategy (Definition 2.4) is exactly the finite-dimensional case* $\mathscr{A} = \mathbb{C}^{d \times d}$, $\tau = \mathrm{tr}\,/d$. *Via the GNS representation, commuting strategies capture exactly the synchronous commuting-operator correlations of the usual two-prover picture, with Bob's measurements acting in the commutant on* $L^2(\mathscr{A}, \tau)$ *— [18, Theorem 5.5]; see also [9] and [10, Section 3.4].*

Definition 2.7 (PCC strategy)**.** ↑ QuantumLib *A synchronous strategy* $\mathscr{S} = (d, M)$ *for a synchronous game* $\mathfrak{G} = (\mathcal{X}, \mathcal{A}, \mu, D)$ *is* PCC *(*projective, consistent, and commuting*, following [1, Section 5.2]) if the measurements associated with any pair of questions asked with positive probability commute: for all* (x, y) *in the support of* μ,

$$M_a^x M_b^y = M_b^y M_a^x \qquad \text{for all } a, b \in \mathcal{A}\,.$$

Remark 2.8. *In [1, Section 5.2] a PCC strategy is a tensor-product strategy (Definition 2.3) with* $\mathcal{H}_{\mathrm{A}} = \mathcal{H}_{\mathrm{B}}$ *that is* projective *(all measurements are PVMs),* consistent *(*$A_a^x \otimes I|\psi\rangle = I \otimes B_a^x|\psi\rangle$ *for all* x, a*), and* commuting *(*$[A_a^x, B_b^y] = 0$ *for all* (x, y) *in the support of* μ *and all* a, b*). In the synchronous framework the first two conditions are automatic: a projective, consistent strategy on a maximally entangled state is exactly a synchronous strategy ([1, Remark 5.13]; see also [18, Theorem 5.5]), so only the commutation condition remains, and we keep the name PCC for alignment with the paper. The "commuting" in PCC is a condition on question pairs in the support of* μ, *unrelated to the commuting (operator) strategies of Definition 2.5. Finally, since*

a synchronous strategy describes both players by the same family of measurements, symmetry is automatic, and we do not distinguish between the PCC and symmetric PCC (SPCC) strategies of [1].

2.1.3 Values

Definition 2.9 (Quantum value)**.** ↑ QuantumLib *The value of a tensor-product strategy* $\mathscr{S} = (\mathcal{H}_{\mathrm{A}}, \mathcal{H}_{\mathrm{B}}, \psi, A, B)$ *in* $\mathfrak{G}$ *is*

$$\mathrm{val}^*(\mathfrak{G}, \mathscr{S}) = \sum_{x,y,a,b} \mu(x,y)\, D(x,y,a,b)\, \langle \psi | A_a^x \otimes B_b^y | \psi \rangle \ ,$$

and the quantum value $\mathrm{val}^*(\mathfrak{G})$ *is the supremum of* $\mathrm{val}^*(\mathfrak{G}, \mathscr{S})$ *over all tensor-product strategies.*

Definition 2.10 (Synchronous value)**.** ↑ QuantumLib *The value of a synchronous strategy* $\mathscr{S} = (d, M)$ *in* $\mathfrak{G}$ *is*

$$\mathrm{val}^s(\mathfrak{G}, \mathscr{S}) = \sum_{x,y,a,b} \mu(x,y)\, D(x,y,a,b)\, \tau\big(M_a^x M_b^y\big) \ ,$$

and the synchronous value *of* $\mathfrak{G}$ *is* $\mathrm{val}^s(\mathfrak{G}) = \sup_{\mathscr{S}} \mathrm{val}^s(\mathfrak{G}, \mathscr{S})$.

Definition 2.11 (Commuting value)**.** ↑ QuantumLib *The value of a commuting strategy* $\mathscr{S} = (\mathscr{A}, \tau, M)$ *in* $\mathfrak{G}$ *is*

$$\mathrm{val}^{\mathrm{co}}(\mathfrak{G}, \mathscr{S}) = \sum_{x,y,a,b} \mu(x,y)\, D(x,y,a,b)\, \tau\big(M_a^x M_b^y\big) \ ,$$

and the commuting value *of* $\mathfrak{G}$ *is* $\mathrm{val}^{\mathrm{co}}(\mathfrak{G}) = \sup_{\mathscr{S}} \mathrm{val}^{\mathrm{co}}(\mathfrak{G}, \mathscr{S})$.

Definition 2.12 (Entanglement requirement)**.** ↑ QuantumLib *For a game* $\mathfrak{G}$ *and* $\nu \in [0,1]$, $\mathrm{Ent}(\mathfrak{G}, \nu)$ *is the least* d *such that there exists a synchronous strategy (Definition 2.4) with dimension at most* d *achieving value at least* ν *in* $\mathfrak{G}$*, and* ∞ *if no synchronous strategy achieves* ν.

2.1.4 Relationship lemmas

Lemma 2.13 (Synchronous value lower-bounds the quantum value)**.** ↑ QuantumLib *For every synchronous game* $\mathfrak{G}$, $\mathrm{val}^s(\mathfrak{G}) \le \mathrm{val}^*(\mathfrak{G})$.

Proof. Given a synchronous strategy $\{M_a^x\}$ on $\mathbb{C}^d$, take the tensor-product strategy with $\mathcal{H}_{\mathrm{A}} = \mathcal{H}_{\mathrm{B}} = \mathbb{C}^d$, $A_a^x = M_a^x$, $B_b^y = (M_b^y)^{\mathsf{T}}$, and $|\psi\rangle = d^{-1/2} \sum_i |i\rangle|i\rangle$; then $\langle \psi | A_a^x \otimes B_b^y | \psi \rangle = \tau(M_a^x M_b^y)$, so the values agree. □

The converse direction — every good tensor-product strategy for a synchronous game can be rounded to a good synchronous strategy — is the almost-synchronicity theorem, imported as Theorem 3.3.

Lemma 2.14 (Synchronous value lower-bounds the commuting value)**.** ↑ QuantumLib *For every synchronous game* $\mathfrak{G}$, $\mathrm{val}^s(\mathfrak{G}) \le \mathrm{val}^{\mathrm{co}}(\mathfrak{G})$.

Proof. A synchronous strategy (d, M) is a commuting strategy with $\mathscr{A} = \mathbb{C}^{d \times d}$ and $\tau = \mathrm{tr}\,/d$, with the same value (in Lean, `MIPRE.SyncStrategy.toCommutingStrategy`; the value is preserved by definitional equality). □

2.2 Linear constraint system games

A well-studied class of nonlocal games arises from systems of linear equations over $\mathbb{F}_2$: the referee asks Alice for a satisfying assignment to a randomly chosen equation and Bob for the value of a randomly chosen variable in it, and checks consistency. The Mermin–Peres magic square is the canonical example. The formalization of this subsection was contributed by Sean Perazzolo (`MIPRE.LCS`); beyond the definitions below it includes observable- and projector-based strategy formalisms and their equivalence, a sum-of-squares decomposition of the loss operators, the extraction of operator identities from perfect play on the EPR state, matrix representations of solution groups, and the magic square as a worked example (`MIPRE.LCS.MagicSquare.merminPeresStrategy`).

Definition 2.15 (LCS instance)**.** *A* binary linear constraint system *consists of r equations over s $\mathbb{F}_2$-valued variables, specified by the set $V_i \subseteq \{1, \dots, s\}$ of variables occurring in each equation i together with a right-hand side $b \in \mathbb{F}_2^r$; equation i reads $\sum_{j \in V_i} x_j = b_i$.*

Definition 2.16 (LCS game)**.** *The* nonlocal game *of an LCS instance: the referee samples an equation i uniformly at random, then a uniformly random variable $j \in V_i$; Alice answers an assignment to the variables, Bob answers a value for x_j, and they win if Alice's assignment satisfies equation i and agrees with Bob's answer on x_j.*

Definition 2.17 (Solution group)**.** *The* solution group *of an LCS instance is the group presented by generators $g_1, \dots, g_s$ and J subject to the relations: all generators are involutions, J is central, variables occurring in a common equation commute, and $\prod_{j \in V_i} g_j = J^{b_i}$ for each equation i. Operator solutions of the system correspond to representations sending $J \mapsto -I$ (in Lean, `MIPRE.LCS.SolutionGroup.solutionGroupRepresentationOfEPRLoss`).*

Theorem 2.18 (Perfect strategies from operator solutions)**.** *If a bipartite observable strategy for an LCS instance is perfect — its local loss operators annihilate the EPR vector — then the associated LCS game has a tensor-product strategy of value* 1.

Proof. Take both players' spaces to be $\mathbb{C}^n$ with the normalized EPR state; Alice measures with the joint projective measurements of her equations and Bob with the binary measurements of his observables. The Born-rule value then equals the EPR expectation of the winning operator, which is 1 by hypothesis. □

2.3 Distance measures

Definition 2.19 (Normalized Hilbert–Schmidt norm)**.** ↑ QuantumLib *For matrices $A, B \in \mathbb{C}^{d \times d}$, let $\langle A, B \rangle_{hs} = \tau(A^\dagger B)$ and $\|A\|_{hs}^2 = \tau(A^\dagger A)$, where $\tau = \mathrm{tr}/d$ is the dimension-normalized trace.*

Definition 2.20 (POVM distance)**.** ↑ QuantumLib *Let μ be a distribution on a finite set $\mathcal{X}$, and $\{M_a^x\}$, $\{N_a^x\}$ families of POVMs. We say M and N are δ-*close*, written $M_a^x \approx_\delta N_a^x$, if $\mathbb{E}_{x \sim \mu} \sum_a \|M_a^x - N_a^x\|_{hs}^2 \le \delta$. Since each difference $M_a^x - N_a^x$ is Hermitian, the summand equals $\tau((M_a^x - N_a^x)^2)$.*

Definition 2.21 (Inconsistency)**.** *Let μ be a distribution on a finite set $\mathcal{X}$, let $|\psi\rangle \in \mathbb{C}^{d_A} \otimes \mathbb{C}^{d_B}$, and let $\{M_a^x\}$, $\{N_a^x\}$ be families of POVMs on the two factors with a common question set $\mathcal{X}$ and a common answer set. Their* inconsistency *on $|\psi\rangle$ is*

$$\mathop{\mathbb{E}}_{x \sim \mu} \sum_{a \neq b} \langle \psi | M_a^x \otimes N_b^x | \psi \rangle,$$

*the probability that the two measurements return different answers. We say M and N are δ-*consistent *if their inconsistency is at most δ.*

Comments. This is Definition 4.8 of [2], in the two-space form used by Theorem 3.7. It is a different notion from the POVM distance of Definition 2.20: consistency compares two families across the two factors of the state, whereas $\approx_\delta$ compares them as operator families.

2.4 Computability conventions

Turing machines are represented by their Gödel numbers; in Lean the intended model is Mathlib's `Nat.Partrec.Code`, with ϕ_e the partial function computed by code e and "e halts on empty input" meaning (`e.eval 0`).`Dom`. Running times are counted as in [8]; polynomial-time computability of a function of strings and integers means time polynomial in the lengths of the string inputs and in log of the integer inputs, unless an integer is explicitly given in unary.

Definition 2.22 (Recursively enumerable)**.** *A language $L \subseteq \{0,1\}^*$ is in* RE *if it is the domain (equivalently, the range) of a partial computable function. The* halting problem *— the set of e such that ϕ_e halts on the empty input — is* RE*-complete under many-one reductions.*

Definition 2.23 (Succinct description)**.** *Following [8], a pair (e, n) of a Turing machine e and integer n is a* succinct description *of the string $x \in \{0,1\}^*$ if $|x| \le 2^n$, the runtime of e on input m is at most $(n+1)(|m|+1)^2$, and $\phi_e(m)$ equals the m-th bit of x for $m < |x|$, and 2 for $m \ge |x|$. ([8] require runtime at most n on every input; in a model that charges for reading its input this is vacuous for $|m| > n$, and the polynomial slack in $|m|$ is what the ambient model's bit-indexing program achieves. Any fixed polynomial in n and $|m|$ would do, provided the same one is used in Theorem 6.6.)*

Definition 2.24 (Game description)**.** *A* game description *is a tuple $g = (n_{\mathcal{X}}, n_{\mathcal{A}}, w, \mathrm{acc})$ of two integers, a finite list w of weighted question pairs, and a finite list* acc *of accepted answer tuples. It determines a synchronous game $\mathfrak{G}_g$ on question alphabet $\{0, \dots, n_{\mathcal{X}}\}$ and answer alphabet $\{0, \dots, n_{\mathcal{A}}\}$: the question distribution is w normalized (a fixed point mass if w has total weight zero), and the predicate accepts exactly the tuples listed in* acc*, except that unequal answers to equal questions always lose. Game descriptions are first-order data and carry a canonical Gödel numbering (`GameData` in the Lean file).*

2.5 Normal form verifiers

Games produced in the proof are presented by pairs of Turing machines: a *sampler*, generating questions by a rigid linear-algebraic procedure that introspection can certify, and a *decider*, checking answers.

Definition 2.25 (Sampler)**.** *An ℓ-*level sampler *with field size $q(n) = 2$ is a Turing machine $\mathcal{S}$ that on input n specifies a linear space $V = \mathbb{F}_2^{s(n)}$ and a pair of ℓ-level conditionally linear functions $L^{\mathrm{A}}, L^{\mathrm{B}}$ on V (Definition 6.1); the question distribution it samples is $\mu_n = (L^{\mathrm{A}}(z), L^{\mathrm{B}}(z))$ for uniform $z \in V$. Formally $\mathcal{S}$ answers dimension, marginal, and linear-evaluation queries as in [1]; the precise query interface is fixed in Section 6.1. $\mathsf{TIME}_{\mathcal{S}}(n)$ denotes its maximal running time on queries with index n.*

Definition 2.26 (Decider)**.** *A* decider *is a 5-input Turing machine $\mathcal{D}$, whose input is interpreted as (n, x, y, a, b) with n an integer index and x, y, a, b strings. It accepts (output 1) or rejects. $\mathsf{TIME}_{\mathcal{D}}(n)$ denotes its maximal running time over inputs with index n.*

Definition 2.27 (Normal form verifier)**.** *A* normal form verifier *is a pair $\mathsf{V} = (\mathcal{S}, \mathcal{D})$ of a sampler and a decider. For each n, V determines a game V_n with question distribution given by $\mathcal{S}$ at index n and decision predicate $D(x, y, a, b) = \mathcal{D}(n, x, y, a, b)$, where answers longer than*

$\mathsf{TIME}_{\mathcal{D}}(n)$ *are rejected. Symmetrizing the roles of the two players makes* V_n *a synchronous game; we write* $\mathrm{val}^*(\mathsf{V}_n)$ *and* $\mathrm{val}^s(\mathsf{V}_n)$ *for its values. The description length* $|\mathsf{V}| = \max\{|\mathcal{S}|, |\mathcal{D}|\}$*, and the number of levels of* V *is that of* $\mathcal{S}$*.*

Definition 2.28 (λ-bounded verifier)**.** *For* $\lambda \in \mathbb{N}$*, a normal form verifier* $\mathsf{V} = (\mathcal{S}, \mathcal{D})$ *is* λ-bounded *if* $\mathsf{TIME}_{\mathcal{S}}(n), \mathsf{TIME}_{\mathcal{D}}(n) \le n^{\lambda}$ *for all* $n \ge 2$*, and* $|\mathsf{V}| \le \lambda$*.*

Definition 2.29 (The class MIP*)**.** *A language* L *is in* MIP* *(more precisely,* $\mathrm{MIP}^*_{1,1/2}(2,1)$*) if there is a polynomial-time computable map* $z \mapsto \mathfrak{G}_z$ *from strings to (descriptions of) games — given by a sampler and decider running in time* $\mathrm{poly}(|z|)$ *— such that* $z \in L$ *implies* $\mathrm{val}^*(\mathfrak{G}_z) = 1$ *and* $z \notin L$ *implies* $\mathrm{val}^*(\mathfrak{G}_z) \le \frac{1}{2}$*.*

3 Background results

This chapter collects theorems that the proof relies on but whose proofs appear in other papers. Each entry gives a full statement, a pointer to the literature, and comments on its role and on formalization. The project may initially assume these as axioms; discharging them are independent sub-projects. Three of these results are larger sub-projects and are developed in their own sections: the computability results in Section 4 and parallel repetition in Section 5; they are included in the summary table below for completeness. The general-purpose quantum information facts feeding the parallel repetition proof (fidelity and Uhlmann's theorem, a relative entropy toolkit, quantum Raz's lemma) are collected in Section 5.2 and likewise appear in the table. Statements are given in the form closest to how they will be used; where the source states a bipartite version and the project needs a synchronous one (see Section 1.3), the translation is part of the formalization task and is noted in the comments.

Each entry carries an estimate of the formalization effort: ● ○ ○ easy for self-contained arguments of at most a few pages, ● ● ○ medium for substantial but well-delimited developments, and ● ● ● hard for results whose formalization is a project in itself; ✓ (mostly) in Mathlib marks statements essentially available in Mathlib. The estimates are for the statement as it will be used (including any synchronous translation) and are of course provisional. Independently of the effort estimate, the ↑ QuantumLib badge (introduced in Section 2) marks results of general quantum-information interest that are intended to be upstreamed to QuantumLib.

Result	Statement	Source	Effort
Gowers–Hatami	Thm. 3.2	[4, 5]	✓
Almost-synchronous correlations	Thm. 3.3	[6]	hard
Orthonormalization	Thm. 3.4	[5]	medium
Magic Square rigidity	Thm. 3.5	[11]	easy
Low individual degree test (classical)	Thm. 3.7	[2]	✓
Low individual degree test (Pauli)	Thm. 3.8	[2]	hard
Anchored parallel repetition	Thm. 5.22	[7]	hard
Direct parallel repetition (commuting-operator)	Thm. 5.5	[31]	✓
Direct parallel repetition (entangled)	Thm. 5.6	[30]	✓†
Tracial density	Thm. 5.8	[9, 31]	✓
Uhlmann, Fuchs–van de Graaf	Thm. 5.11, Lem. 5.12	[32, 33]	medium
Relative entropy toolkit	Lems. 5.14–5.17	[35, 34]	medium
Quantum Raz's lemma	Lem. 5.18	[7]	easy
Holenstein conditioning lemmas	Lem. 5.19	[28]	easy
Recursive compression lemma	Lem. 4.10	[8]	✓
Compressibility criterion	Lem. 4.11	[10]	✓
Efficient computability toolkit	Lems. 4.7–4.9	[8]	hard
Halting undecidability	Thm. 3.9	Mathlib	✓
Succinct Cook–Levin	Thm. 3.10	[20, 19]	hard
Self-dual normal bases	Lem. 3.11	[21, 22]	medium
Schwartz–Zippel	Lem. 3.12	[23, 24]	✓
Value approximable from below	Lem. 3.13	folklore	medium

†The Lean proof of Theorem 5.6 is complete modulo Lemma 5.7, which is stated in Lean but not yet proved. Theorem 5.22 and the results of Section 5.2 are no longer on the path to the main theorem (Remark 5.9).

3.1 The Gowers–Hatami theorem

↑ QuantumLib

Definition 3.1 (Approximate representation)**.** ↑ QuantumLib *Let G be a finite group and $\varepsilon \in \mathbb{R}$. An* ε-approximate representation *of G is a function $f : G \to U(d)$ such that*

$$\mathop{\mathbb{E}}_{x,y\sim G} \operatorname{Re}\langle f(x)f(y), f(xy)\rangle_{hs} \geq 1-\varepsilon \ ,$$

where $\langle\cdot,\cdot\rangle_{hs}$ is the normalized Hilbert–Schmidt inner product of Definition 2.19. Since each value of f is unitary, $\|f(x)f(y)-f(xy)\|_{hs}^2 = 2 - 2\operatorname{Re}\langle f(x)f(y), f(xy)\rangle_{hs}$, so the condition is equivalent to the defect form $\mathbb{E}_{x,y\sim G}\|f(x)f(y)-f(xy)\|_{hs}^2 \leq 2\varepsilon$.

Theorem 3.2 (Gowers–Hatami)**.** *Let G be a finite group and $f : G \to U(d)$ an ε-approximate representation of G. Then there exist $d' \geq d$, an isometry $V : \mathbb{C}^d \to \mathbb{C}^{d'}$, and a representation $\rho : G \to U(d')$ such that*

$$\mathop{\mathbb{E}}_{x\sim G} \left\|f(x) - V^\dagger \rho(x) V\right\|_{hs}^2 \leq 2\varepsilon \ .$$

Proof. Let $L(G,\mathbb{C}^d) \cong \mathbb{C}^{|G|d}$ carry the inner product normalized by $1/|G|$, let $V : \mathbb{C}^d \to L(G,\mathbb{C}^d)$ be the map $u \mapsto (x \mapsto f(x)u)$, and let ρ be the right regular representation, $(\rho(x)F)(y) = F(yx)$. Then V is an isometry, and the compression $V^\dagger\rho(x)V = \mathbb{E}_{y\sim G} f(y)^\dagger f(yx)$ is an average of unitaries, hence has squared Hilbert–Schmidt norm at most 1. Moreover $\mathbb{E}_x \operatorname{Re}\langle f(x), V^\dagger\rho(x)V\rangle_{hs} = \mathbb{E}_{x,y}\operatorname{Re}\langle f(x)f(y), f(xy)\rangle_{hs} \geq 1-\varepsilon$ by hypothesis, and the claim follows by expanding $\left\|f(x) - V^\dagger\rho(x)V\right\|_{hs}^2$. The construction gives $d' = |G|\,d$. □

Source. Gowers and Hatami [4]. A quantitative form suited to applications in nonlocal games, and a version adapted to the tracial setting, is given by de la Salle [5] and in [6].

Comments. This is the stability result underlying all rigidity (self-testing) arguments in the proof: an approximate representation of the Pauli group $\mathbb{F}_2^{2k}$ (with its cocycle) is close to an exact one, which forces the presence of EPR pairs. The statement above is fully formalized (statement and proof) in `MIPRE/Background/GowersHatami/Basic.lean`, following the regular-representation argument sketched in the proof; the intermediate statement, with the exact representation living on the concrete index set $G \times \{1, \dots, d\}$, is `MIPRE.gowers_hatami_prod`. Two remarks on the statement. (i) It asserts only $d' \geq d$, with witness $d' = |G|d$; it does not include the dimension bound $d' \leq (1 + C\varepsilon)d$ of [4], whose proof goes through the Fourier-analytic route via irreducible representations. Whether the sharper bound is needed downstream (in the Pauli-group application feeding Theorem 3.8) is to be revisited; if so it will be added as a separate strengthening. (ii) In the Lean encoding the isometry is stored as the $d \times d'$ matrix of its adjoint, so the isometry condition reads $VV^\dagger = I$ and the compression is written $V\rho(x)V^\dagger$. The normalized Hilbert–Schmidt norm is the tracial form needed for the synchronous framework [5, 6].

3.2 Almost-synchronous correlations

• • • hard | ↑ QuantumLib

Theorem 3.3 (Almost-synchronous strategies are near synchronous strategies)**.** *There are universal constants $C > 0$ and $0 < c \leq 1$ such that for every synchronous game $\mathfrak{G}$,*

$$1 - \mathrm{val}^s(\mathfrak{G}) \;\leq\; C\big(1 - \mathrm{val}^*(\mathfrak{G})\big)^c \,.$$

More precisely, every tensor-product strategy for $\mathfrak{G}$ with value $1 - \varepsilon$ is, up to local isometries, well-approximated on average by a convex combination of synchronous strategies each of value at least $1 - C\varepsilon^c$.

Source. Vidick [6], building on [17, 18]; the statement above is the corollary needed here, with constants independent of the game's size.

Comments. The proof uses the orthonormalization lemma (Section 3.3) and a rounding argument.

3.3 The orthonormalization lemma

• • ○ medium | ↑ QuantumLib

Theorem 3.4 (Orthonormalization)**.** *There is a universal constant $C > 0$ such that the following holds. Let $(\mathcal{M}, \tau)$ be a von Neumann algebra with a normal faithful tracial state (for the finite-dimensional case, $\mathcal{M} = M_d(\mathbb{C})$ with the normalized trace). Let $\{M_a\}_{a \in \mathcal{A}}$ be a POVM in $\mathcal{M}$ that is ε-nearly projective:*

$$\sum_{a \in \mathcal{A}} \tau\big(M_a - M_a^2\big) \leq \varepsilon \,.$$

Then there is a projective measurement $\{P_a\}_{a \in \mathcal{A}}$ in $\mathcal{M}$ with

$$\sum_{a \in \mathcal{A}} \|M_a - P_a\|_{L^2(\tau)}^2 \leq C\varepsilon \,.$$

The same holds, on average, for a family $\{M_a^x\}$ indexed by $x \sim \mu$.

Source. de la Salle [5], which gives a dimension-free bound with explicit constants; earlier versions with worse parameters appear in [17] and in [1].

Comments. Used constantly: soundness analyses produce POVMs that are only approximately projective, and this lemma replaces them by genuinely projective measurements on the same space, which is what makes the "projective strategies throughout" convention of Section 1.3 viable (avoiding Naimark dilation). The finite-dimensional case suffices for the main theorem and is a clean, self-contained formalization target in linear algebra.

3.4 Magic Square rigidity

• ○ ○ easy | ↑ QuantumLib

Theorem 3.5 (Magic Square rigidity)**.** *Let $\mathscr{S} = (|\psi\rangle, A, B)$ be a strategy that succeeds in the Magic Square game $\mathfrak{G}^{\mathrm{MS}}$ with probability $1-\varepsilon$. Then there exist local isometries $\phi_{\mathrm{A}} : \mathcal{H}_{\mathrm{A}} \to (\mathbb{C}^2)^{\otimes 2} \otimes \mathcal{H}_{\mathrm{A}''}$, $\phi_{\mathrm{B}} : \mathcal{H}_{\mathrm{B}} \to (\mathbb{C}^2)^{\otimes 2} \otimes \mathcal{H}_{\mathrm{B}''}$ and a state $|\mathrm{AUX}\rangle \in \mathcal{H}_{\mathrm{A}''} \otimes \mathcal{H}_{\mathrm{B}''}$ such that*

$$\big\|\phi_{\mathrm{A}} \otimes \phi_{\mathrm{B}}|\psi\rangle - |\mathrm{EPR}_2\rangle^{\otimes 2} \otimes |\mathrm{AUX}\rangle\big\| \leq O(\sqrt{\varepsilon}) \,,$$

and, writing $\tilde{A} = \phi_{\mathrm{A}} A \phi_{\mathrm{A}}^{\dagger}$, the observables for the questions $\textsc{Variable}_1$ and $\textsc{Variable}_5$ satisfy $\tilde{A}_b^{\textsc{Variable}_1} \approx_{\sqrt{\varepsilon}} \sigma_b^X \otimes I$ and $\tilde{A}_b^{\textsc{Variable}_5} \approx_{\sqrt{\varepsilon}} \sigma_b^Z \otimes I$ on $|\mathrm{EPR}_2\rangle^{\otimes 2} \otimes |\mathrm{AUX}\rangle$ (and symmetrically for $\tilde{B}$). In particular the corresponding ± 1-observables approximately anticommute: $\tilde{A}^{\textsc{Variable}_1}\tilde{A}^{\textsc{Variable}_5} \approx_{\sqrt{\varepsilon}} -\tilde{A}^{\textsc{Variable}_5}\tilde{A}^{\textsc{Variable}_1}$.

Source. Wu, Bancal, McKague and Scarani [11]; the game is the Mermin–Peres magic square. The form above is [1, Section 8].

Comments. This is the elementary seed of all rigidity in the proof: it certifies a single pair of anticommuting observables, which the low individual degree test then amplifies to many qubits. The game is constant-sized, the proof is a page of algebra, and it is an excellent first self-testing target. The synchronous restatement (one measurement family, tracial state) should be derived once and reused as a template for the other rigidity statements.

3.5 The classical low individual degree test

• • • hard

Definition 3.6 (Low individual degree test)**.** *Let $\mathbb{F}$ be a finite field with q elements and let $m \geq 1$ and d be integers. The* (m, q, d)-low individual degree test *is the two-player game in which the verifier picks one of the following three sub-tests with probability $\frac{1}{3}$ each, having sampled a uniform point $u \sim \mathbb{F}^m$.*

1. ***Axis-parallel lines test.*** *Sample a uniform direction $i \sim [m]$ and a uniform role. One player receives the line $\ell = \{u + te_i\}$ and answers a univariate polynomial f of degree at most d; the other receives u and answers $a \in \mathbb{F}$. Accept if $f(u) = a$.*
2. ***Self-consistency test.*** *Both players receive u and answer elements of $\mathbb{F}$. Accept if the answers are equal.*

3. ***Diagonal lines test.*** *Sample a uniform $j \sim [m]$, a uniform direction $v \in \mathbb{F}^m$ whose coordinates after the j-th vanish, and a uniform role. One player receives the line $\ell = \{u + tv\}$ and answers a univariate polynomial of degree at most md; the other receives u and answers $a \in \mathbb{F}$. Accept if $f(u) = a$.*

Comments. This is Figure 1 of [2]. The direction v of the diagonal lines test may be zero, in which case the "line" is the single point u; this degenerate case is part of the test as stated. In the formalization, line questions are the lines themselves, presented canonically (the direction is rescaled so that its first nonzero coordinate is 1, and the base point is the point of the line whose corresponding coordinate vanishes), so that the same line asked by two sub-tests is the same question; a line answer is a polynomial in the parameter of that canonical parametrization.

Theorem 3.7 (Quantum soundness of the classical low individual degree test)**.** *Let*

$$\delta_{lidt}(\varepsilon, m, d, q, k) = 10^5\, k^2 m^4 \big(\varepsilon^{1/40000} + (d/q)^{1/40000} + e^{-k/(2560000m^2)}\big).$$

Let $\mathscr{S} = (|\psi\rangle, A, B)$ be a tensor-product strategy for the (m, q, d)-low individual degree test that succeeds with probability at least $1 - \varepsilon$, and let k be an integer with $k \geq 400md$ and $k > 0$. Then there exist projective measurements $G^{\mathrm{A}} = \{G^{\mathrm{A}}_g\}$ on Alice's space and $G^{\mathrm{B}} = \{G^{\mathrm{B}}_g\}$ on Bob's space, both with outcomes the polynomials g in m variables of individual degree at most d, such that, writing $\delta = \delta_{lidt}(\varepsilon, m, d, q, k)$ and letting $u \sim \mathbb{F}^m$ be uniform:

1. *Alice's point measurement A^u and the evaluation $G^{\mathrm{B}}_{[g(u)=\cdot]}$ are δ-consistent;*
2. *the evaluation $G^{\mathrm{A}}_{[g(u)=\cdot]}$ and Bob's point measurement B^u are δ-consistent;*
3. *G^{A} and G^{B} are δ-consistent.*

Proof. This is Theorem 1.3 of [2] (`thm:main-formal` there), whose proof is the MIPStarRE formalization vendored into this repository. The Lean proof translates a strategy for our game into MIPStarRE's strategy container (the state as a density matrix; the line measurements as coarse-grainings of the measurements at the canonical line questions along the change of parametrization, which makes them covariant under rebasing), shows that MIPStarRE's failure surrogate is at most the rejection probability $1 - \mathrm{val}$ (the consistency defect of each branch is at most the rejection probability of that branch, since acceptance forces the coarse-grained answers to agree), applies their main theorem, and reads the resulting measurements back into our vocabulary. □

Source. Ji, Natarajan, Vidick, Wright and Yuen [2], Theorem 1.3 (the formal version for general, not necessarily symmetric, strategies). The proof is a formalization by Sirui Lu and collaborators (https://github.com/LionSR/MIPStarRE), vendored into this repository with the authors' permission at `MIPRE/Background/LIDT/MIPStarRE/`; the statement above is this project's own, and is connected to theirs by an explicit bridge.

Comments. Two hypotheses correct the printed statement, as established by that formalization: the sampling parameter satisfies $k \geq 400md$ where the paper prints $k \geq md$, and k must be positive, since the printed error vanishes at $k = 0$. The parameter k remains free: the error carries both a factor k^2 and a term $e^{-k/(2560000m^2)}$, so no single choice of k reduces it to a function of (ε, m, d, q); a corollary in the shape of δ_{qld} below is a separate derivation.

3.6 The quantum low individual degree test

• • • hard | ↑ QuantumLib

Theorem 3.8 (Pauli basis test / quantum low individual degree test)**.** *There exists a function*

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

for universal constants $a \geq 1$ and $0 < b < 1$ such that the following holds. For all admissible parameter tuples $\mathsf{qldparams} = (q, m, d)$ *and all strategies $\mathscr{S} = (|\psi\rangle, A, B)$ for the game $\mathfrak{G}^{\textsc{Pauli}}_{\mathsf{qldparams}}$ that succeed with probability at least $1 - \varepsilon$, there exist local isometries $\phi_{\mathrm{A}}, \phi_{\mathrm{B}}$ mapping into $\mathcal{H}' \otimes (\mathbb{C}^q)^{\otimes M}$ with $M = 2^m$, and a state $|\textsc{aux}\rangle$, such that $\left\|\phi_{\mathrm{A}} \otimes \phi_{\mathrm{B}}|\psi\rangle - |\textsc{aux}\rangle \otimes |\mathrm{EPR}_q\rangle^{\otimes M}\right\| \leq \delta_{qld}$, and for $W \in \{X, Z\}$ the measurements for the questions $(\textsc{Pauli}, W)$ are δ_{qld}-close to the Pauli basis measurements τ_u^W, $u \in \mathbb{F}_q^M$, acting on $|\mathrm{EPR}_q\rangle^{\otimes M}$.*

Source. Ji, Natarajan, Vidick, Wright and Yuen [2], which proves quantum soundness of the classical low individual degree test and derives the Pauli basis test; this corrects the analysis of the total degree test relied on in early versions of [1]. The reduction from the Pauli test to the low individual degree test follows [1, Section 7], with the Magic Square theorem (Theorem 3.5) providing the base anticommutation.

Comments. This is the largest and hardest imported result: its proof is an induction on the number of variables with a delicate self-improvement argument. It is the workhorse of question reduction (introspection certifies the players' sampling of their own questions against Pauli basis measurements). Its classical ingredient is now available as Theorem 3.7; what remains is the reduction from the Pauli basis test, which uses Theorem 3.5 for the base anticommutation and Theorem 3.2 and Theorem 3.4 as black boxes, together with a corollary of Theorem 3.7 collapsing the sampling parameter k into a single error function.

3.7 Undecidability of the halting problem

✓ (mostly) in Mathlib

Theorem 3.9 (Turing)**.** *The halting problem is r.e. but not decidable; it is* RE*-complete under many-one (indeed polynomial-time) reductions.*

Source. Turing (1936); in Mathlib as part of the `Computability` library.

Comments. Listed to close the dependency graph: combined with the halting reduction (Theorem 6.9) it yields undecidability of approximating the game value, and RE-completeness gives $\mathrm{RE} \subseteq \mathrm{MIP}^*$.

3.8 Cook–Levin and succinct satisfiability

• • • hard

Theorem 3.10 (Explicit succinct Cook–Levin)**.** *There is a polynomial-time algorithm that, given a decider $\mathcal{D}$, an index n, a time bound T (in binary) and strings x, y, outputs a succinct description (in the sense of Definition 2.23, via Boolean formulas) of a 3SAT instance φ of size $2^{\mathrm{poly}(\log T)}$ such that φ is satisfiable if and only if there exist strings a, b of length at most T with $\mathcal{D}(n, x, y, a, b) = 1$ within T steps; moreover satisfying assignments of φ encode the computation tableau together with (a, b).*

Source. Cook [20] and Levin for the tableau reduction; Papadimitriou and Yannakakis [19] for succinct representations and their complexity; the explicit form needed here is [1, Section 10].

Comments. This is the classical core of answer reduction: the players prove satisfiability of φ (arithmetized over a low-degree extension) instead of sending (a, b). The statement is long but entirely combinatorial. Cook–Levin has been formalized in other proof assistants; a Lean formalization with the explicit-uniformity guarantees above is a substantial but well-specified sub-project.

3.9 Self-dual normal bases

● ● ○ medium

(Thomas: Is this needed?)

Lemma 3.11 (Deterministic self-dual normal basis)**.** *There is a deterministic algorithm that, given an odd integer $k > 0$, outputs in time* poly(k) *a self-dual normal basis of $\mathbb{F}_{2^k}$ over $\mathbb{F}_2$ together with its multiplication tables.*

Source. Combines Shoup's deterministic irreducible-polynomial construction [21], Lenstra's isomorphism algorithm [22], and the classical existence and construction of self-dual normal bases in characteristic 2 for odd k; see the discussion in [1, Section 3].

Comments. Needed so that verifiers can do field arithmetic over $\mathbb{F}_{2^k}$ using only $\mathbb{F}_2$-data (the "downsize" maps), uniformly and deterministically. This is why field sizes are restricted to $q = 2^k$, k odd ("admissible"). Self-contained algebra/algorithmics; independent of everything quantum.

3.10 The Schwartz–Zippel lemma

✓ (mostly) in Mathlib

Lemma 3.12 (Schwartz–Zippel)**.** *Let $f \neq g : \mathbb{F}_q^m \to \mathbb{F}_q$ be polynomials of total degree at most d. Then* $\Pr_{x \sim \mathbb{F}_q^m}[f(x) = g(x)] \leq d/q$.

Source. [23, 24]. A version is available in Mathlib (multivariate polynomials over finite fields).

Comments. Used throughout the low-degree machinery (distance of the Reed–Muller code, soundness of arithmetization). The variant for individual degree should be derived alongside it.

3.11 Approximating the value from below ($\mathrm{MIP}^* \subseteq \mathrm{RE}$)

● ● ○ medium

Lemma 3.13 (The value is approximable from below)**.** *The set $\{(g, p) : g$ a game description, $p \in \mathbb{Q}, \mathrm{val}^s(\mathfrak{G}_g) > p\}$ is recursively enumerable, and likewise for* val^*.

Source. Folklore; see [1, Section 12]. Enumerate tensor-product strategies with entries in $\mathbb{Q}[i]$ and evaluate; density of such strategies among all tensor-product strategies gives completeness of the enumeration.

Comments. This is the easy inclusion $\mathrm{MIP}^* \subseteq \mathrm{RE}$ and one half of Theorem 7.3. The commuting-operator analogue (approximating $\mathrm{val}^{\mathrm{co}}$ from *above* via the NPA hierarchy [25]) is *not* needed for the main theorem, but becomes relevant for the downstream consequences in Section 8.

4 Computability

This chapter collects the computability-theoretic ingredients of the proof: classical statements with no quantum content, several of which partially exist in Mathlib. Like the results of Section 3, these are external results that the project may initially assume as axioms, and each entry carries the same formalization-effort estimate. The concrete machine model and its canonical codes come first; then the toolkit of resource-bounded computability results; the recursive compression lemma, which builds on the toolkit, isolates the self-referential part of the proof.

4.1 The machine model

The machine-level side of the project is formalized directly: a concrete multi-tape machine model with several input tapes, a canonical first-order syntax of machine *codes* with an exact binary serialization, and a reference evaluator.

Definition 4.1 (Multi-input Turing machine)**.** *A* multi-input Turing machine *with i input tapes and w work tapes has i two-way read-only input tapes (heads clamped one blank cell beyond either end of the input), w bidirectional work tapes over an alphabet extended with a blank symbol, and a write-only output stream. A transition may write and move on every work tape, move every input head, emit at most one output symbol, and either continue or halt. Time is counted in steps, and space as the number of work-tape cells visited.*

At $i = 1$ this is exactly the CSLib multi-tape model.

Definition 4.2 (Machine codes)**.** *A* machine code *is first-order data: a header (number of work tapes, alphabet size, number of states, start state) and a dense transition table with one entry per observation, in a fixed canonical order. A well-formed code is interpreted as a multi-input machine, with binary input and output through two reserved alphabet symbols. Codes serialize exactly: the* description $\alpha \in \{0,1\}^*$ *of a code has size $|\alpha|$, and* total decoding *interprets every string — $[\alpha]_i$ denotes the coded machine with i input tapes described by α, with every malformed string denoting one fixed machine that immediately rejects.*

Definition 4.3 (Runs of coded machines)**.** *Let c be a machine code with i input tapes, $x = (x_1, \ldots, x_i)$ bit strings and $y \in \{0,1\}^*$. The code c* produces *y on x if its run on x halts with output y (in some time and space). The* budgeted evaluation *of c on x with budget T returns the output if c has halted within T steps, and a timeout marker otherwise; it is total and executable, and the two notions agree: c produces y if and only if some budget suffices to observe it.*

Lemma 4.4 (Canonical serialization)**.** *Exact decoding inverts encoding, distinct codes have distinct descriptions, and every description accepted by the exact decoder is the canonical encoding of the code it parses to.*

Proof. Compositional, from paired prefix and soundness lemmas for each field of the description format. □

The following two statements are the machine-level counterparts of Lemma 4.7.

Lemma 4.5 (Universal machine)**.** *For each arity i there are a machine code U with $i+1$ input tapes and a polynomial P such that for all $\alpha \in \{0,1\}^*$, all inputs x and all y: U on $(\alpha; x)$ produces y if and only if $[\alpha]_i$ on x produces y; and whenever $[\alpha]_i$ halts on x within t steps, U halts on $(\alpha; x)$ within $P(|\alpha| + t)$ time and space.*

Proof. TODO □

Lemma 4.6 (Bounded universal machine)**.** *For each arity i there are a machine code U with $i+2$ input tapes and a polynomial P such that on input $(\alpha; T$ in binary$; x)$, the machine U always halts within $P(|\alpha| + T)$ steps, outputting the canonically encoded budgeted evaluation of $[\alpha]_i$ on x with budget T.*

Proof. TODO □

4.2 Efficient computability toolkit

• • • hard (infrastructure: requires a cost model for computation)

Lemma 4.7 (Efficient universal Turing machine)**.** *There is a program u and a polynomial p such that $\phi_u(e, x) = \phi_e(x)$ for all e, x, with* $\mathrm{runtime}(u, (e, x)) \le p\big(|e|, |x|, \mathrm{runtime}(e, x)\big)$.

Proof. Both the universal program and its clocked variant are self-interpreters of the ambient list language, written in the language itself (`MIPRE.Cost.Machine.univProg`, `MIPRE.Cost.Machine.univTProg`). Evaluation is first presented as an abstract machine on configurations (a control "evaluate p" or "return v", an environment and a stack of frames) whose steps are charged exactly the costs of the evaluation rules, so that a derivation of cost t corresponds to a machine run of at most $3t$ steps reaching a final configuration, and conversely every run reaching a final configuration comes from a derivation. Configurations are encoded as data and the step function is implemented by a fixed program (`stepProg`) at cost quadratic in the size of the configuration. Along the run simulating a derivation of cost t on the input x, every configuration has size polynomial in $|e| + |x| + t$: values are results of sub-derivations, hence of size at most t; environments and stacks grow by at most one entry per unit of cost. The universal program iterates `stepProg` until a final configuration is reached, so its runs are polynomially bounded, and any of its halting runs is a machine run reaching a final configuration, hence a derivation with the same result. The clocked variant additionally carries, in unary, the remaining cost budget (decremented by the exact cost of each step), a step budget $3k + 1$ guaranteeing termination, and a size guard Θ, a fixed polynomial in $k + |e| + |x|$ dominating the configuration sizes of any run of cost at most k; it returns the timeout value when a budget or the guard is exceeded. The guard is checked on the configuration produced by each step, whose size is bounded by the cost of producing it, so the loop only handles configurations of polynomial size even when the simulated run does not halt. □

Lemma 4.8 (Efficient s-m-n)**.** *There is a polynomial-time computable s such that $\phi_{s(e,x)}(y) = \phi_e(x, y)$ for all e, x, y, with the runtimes of e and $s(e, x)$ polynomially equivalent.*

Proof. In the ambient list language, $s(e, x)$ is the program that pairs the literal x with its input and runs e on the pair (`MIPRE.Cost.hardcode`). The time overhead is additive and linear in $|x|$ and the input size, in both directions, and the description of $s(e, x)$ is that of e plus $|x|$ plus a constant. The map s itself is computed in linear time by a fixed program assembling the description (`smnProg`). □

Lemma 4.9 (Efficient Kleene recursion theorem)**.** *For every polynomial-time computable $f : \{0,1\}^* \to \{0,1\}^*$ there is a program e with $\phi_e = \phi_{f(e)}$ and the runtime of e polynomially bounded in terms of that of $f(e)$. (This is the direction used by Lemma 4.10; [8] states the runtimes as polynomially equivalent, and the converse direction would require a simulator that is never faster than the simulated program.)*

Proof. The classical construction. Let G be the program that on input (x, v) computes $s(x, x)$ (Lemma 4.8), applies f to it, and runs the resulting program on v through the universal machine (Lemma 4.7); put $e = s(G, G)$. Then e on v is G on (G, v), which runs $f(s(G, G)) = f(e)$ on v: the two programs halt on the same inputs with the same results (the backward direction inverts the run of G step by step and uses that the universal machine halts only if the simulated program does), and a run of $f(e)$ of time t gives a run of e within $p(|v|, t)$, where p is the universal machine's overhead polynomial shifted by the fixed size of $f(e)$, plus a linear term for the copies of the input and a constant for the fixed computation of $f(e)$. □

Source. Folklore; stated in this efficient form in [8], Lemmas 2.1–2.3.

Comments. Mathlib contains the non-resource-bounded versions (universal machine, s-m-n and the Rogers fixed-point theorem for `Nat.Partrec.Code`). The efficient versions require a cost model for the chosen machine model; this is real infrastructure work and a prerequisite for everything involving "polynomial-time". The formalization's ambient model is not a Turing machine but a small first-order list-processing language (`MIPRE.Cost.Prog`, over binary trees) with a unit-cost semantics in which reading a value costs its size; it is polynomially equivalent to Turing-machine time, programs are their own tree serializations (so "a program e" and "$|e|$" are literal), and the s-m-n theorem is a linear-overhead syntactic operation. Mathlib's `Turing.ToPartrec.Code` was tried first and rejected: its programs cannot rebuild a list in polynomial time. A pragmatic first pass can replace "polynomial time" by "computable" wherever Section 1.3 permits, deferring the cost model. The machine-level counterpart, for the concrete model of Section 4.1, is stated as Lemmas 4.5 and 4.6; the pipeline consumes only the present ambient version.

4.3 The abstract compression lemmas

● ○ ○ easy (given the toolkit of Section 4.2)

Lemma 4.10 (Recursive compression)**.** *Let $f : \{0,1\}^* \to \mathbb{N} \cup \{\infty\}$ be a function and $A \subseteq \{x : f(x) < \infty\}$ a nonempty language. Suppose there is a polynomial-time computable function $\textsc{Compr} : \{0,1\}^* \times \mathbb{N} \to \{0,1\}^*$ such that whenever $\hat{x}$ is a succinct description (Definition 2.23) of a string x and $y = \textsc{Compr}(\hat{x}, n)$:*

1. *$f(y) \geq \max\{f(x), n\}$, and*
2. *if $x \in A$ then $y \in A$.*

Then there is a polynomial-time computable function g reducing the halting problem to A: for every Turing machine e, if ϕ_e halts on the empty input then $g(e) \in A$, and otherwise $f(g(e)) = \infty$.

Proof. The construction of [8], with two harmless modifications: the "next level" of the recursion is $2n + 1$ instead of $n + 1$ (prepending a bit to a binary numeral is one step in the list language, an increment is not), and the threshold below is explicit. Let β be the fixed *bit-query* program that, on input $(((c', e), n), m)$, runs the universal machine on c' with input (e, n) to obtain a

string y and outputs the m-th bit of y (or 2 if $m \geq |y|$). Let a be the polynomial-time *decider* that on $(c', (e, n))$ outputs a fixed $y_0 \in A$ if e halts on the empty input within $\log n$ steps, and $\textsc{Compr}\big((s(\beta, ((c', e), 2n+1)), n), n\big)$ otherwise, where s is the s-m-n map of Lemma 4.8. By Lemma 4.9 there is a program c with $\phi_c(e, n) = \phi_a(c, (e, n)) =: h(e, n)$, running in time polynomial in $|e| + \log n$. Hence for n above a threshold $r(e) = 2^{K+1+|e|}$ (with K a constant depending only on the polynomial overheads), $(s(\beta, ((c, e), 2n+1)), n)$ is a succinct description of $h(e, 2n+1)$: the string has length at most 2^n and the bit-query program answers within the budget of Definition 2.23. The reduction is $g(e) = h(e, r(e))$. If e does not halt, then for every $n \geq r(e)$ the decider compresses, so $f(h(e, n)) \geq \max\{f(h(e, 2n+1)), n\}$, and iterating gives $f(h(e, n)) \geq n + k$ for all k, i.e. $f(g(e)) = \infty$. If e halts in T steps, then $h(e, n) = y_0 \in A$ once $n \geq 2^T$, and by downward induction along the levels $n, 2n+1, 4n+3, \ldots$ the second property of $\textsc{Compr}$ gives $h(e, n) \in A$ for every $n \geq r(e)$, in particular $g(e) \in A$. □

Source. Marks, Nezhadi and Yuen [8], Lemma 5.1 (the variant of their Lemma 3.1 with compression parameter n); the technique is implicit in [26, 12, 1]. The proof is a fixed-point construction using the efficient universal machine, s-m-n and Kleene recursion theorems (Section 4.2).

Comments. This is the entire "self-referential" part of the proof, isolated in a statement with no quantum content. In the application (Section 6.7), A is the set of normal form verifier descriptions with $\mathrm{val}^s = 1$ and finite entanglement requirement, and $f(\mathsf{V}) = \mathrm{Ent}(\mathsf{V}, \frac{1}{2})$ (suitably formalized); the compression procedure is Theorem 6.6. The value-form pipeline of Section 6 uses Lemma 4.11 below instead; this lemma is retained as the entanglement-form route (Remark 6.8), where the factor $\frac{1}{2}$ in the entanglement bound of compression requires the variant with hypothesis $f(y) \geq \frac{1}{2}\max\{f(x), G(n)\}$ for a function G growing faster than the levels. Note that [8] also proves a converse (their Lemma 4.1): a halting reduction of this kind exists *only if* a compression procedure does, which is evidence that this decomposition of the proof is canonical. The lemma and its proof are a self-contained computability project, an ideal early milestone.

Lemma 4.11 (Compressibility criterion)**.** *Let $A, B \subseteq \{0,1\}^*$ be languages with distinguished elements $y_{\mathrm{yes}} \in A$ and $y_{\mathrm{no}} \in B$, and suppose that the complement of B is recursively enumerable: there is a program S that halts on input x if and only if $x \notin B$. Suppose there is a polynomial-time computable function $\textsc{Compr} : \{0,1\}^* \times \mathbb{N} \to \{0,1\}^*$ such that whenever $\hat{x}$ is a succinct description (Definition 2.23) of a string x and $y = \textsc{Compr}(\hat{x}, n)$:*

1. *if $x \in A$ then $y \in A$, and*
2. *if $x \in B$ then $y \in B$.*

Then there is a polynomial-time computable function g reducing the halting problem to (A, B): for every Turing machine e, if ϕ_e halts on the empty input then $g(e) \in A$, and otherwise $g(e) \in B$.

Proof. The construction of Lemma 4.10, with a third branch. Let β be the bit-query program of that proof, and let σ be the program that, on $((c', (e, (m, m))), \cdot)$, runs the universal machine on c' with input $(e, (m, m))$ to obtain a string x and then runs S on x. Let a be the polynomial-time decider that on $(c', (e, (m, n)))$ outputs y_{yes} if e halts on the empty input within $\log n$ steps; otherwise y_{no} if $s(\sigma, (c', (e, (m, m))))$ halts within $\log n$ steps; and otherwise $\textsc{Compr}\big((s(\beta, ((c', e), (m, 2n+1))), n), n\big)$, where s is the s-m-n map of Lemma 4.8. By Lemma 4.9 there is a program c with $\phi_c(e, (m, n)) = \phi_a(c, (e, (m, n))) =: h(e, m, n)$, running in time polynomial in $|e| + \log m + \log n$. As before, for n above a threshold $r(e) = 2^{K+1+|e|}$ and $m = r(e)$, the pair $(s(\beta, ((c, e), (m, 2n+1))), n)$ is a succinct description of $h(e, m, 2n+1)$; the start level m is carried as data so that the threshold can be chosen from the running time of a afterwards.

The reduction is $g(e) = h(e, r(e), r(e)) =: x_e$; the second branch tests, at every level, whether S halts on x_e.

If e does not halt, suppose $x_e \notin B$. Then S halts on x_e in some time T, so at every level n with $\log n \geq T$ the decider outputs $y_{\text{no}} \in B$, and by downward induction along the levels $r(e), 2r(e)+1, 4r(e)+3, \ldots$ from such a level, property 2 of COMPR gives $h(e, r(e), n) \in B$ at every level, in particular $x_e \in B$, a contradiction. If e halts in T steps, then $h(e, r(e), n) = y_{\text{yes}} \in A$ once $\log n \geq T$, and by downward induction property 1 gives $h(e, r(e), n) \in A$ at every level, provided the second branch never fires at a level where the first has not. If it fired at such a level n_1, then S halts on x_e, so $x_e \notin B$; but the output $y_{\text{no}} \in B$ at level n_1 and property 2, by downward induction along the levels between $r(e)$ and n_1 (at which the first branch does not fire either), give $x_e \in B$, a contradiction. Hence $g(e) = x_e \in A$. □

Source. Lin [10], the compression criterion for RE-complete problems (direction "weakly compressible implies RE-complete"), whose self-referential sequence has exactly these three branches; the formulation through succinct descriptions is that of [8] and of Lemma 4.10. Lin's theorem is an equivalence: given that the no-instances form a coRE set, a decision problem is RE-complete if and only if it is compressible.

Comments. This is the lemma the main theorem uses (Section 6.7): A is the set of normal form verifier descriptions with a value-1 PCC strategy, B the set of those with $\text{val}^s \leq \frac{1}{2}$, and S the enumeration of Lemma 3.13. Compared with Lemma 4.10, the growth hypothesis on a measure is replaced by the semidecidability of the complement of B — both directions of the specification of S are used — and nothing about entanglement is required of the compression procedure. The Lean proof reuses the toolkit of Section 4.2: the search branch runs the semidecision procedure through the clocked universal machine, and the start level of the recursion is carried as data next to the current level.

5 Parallel repetition

This chapter states the parallel repetition theorems available for gap amplification in the compression procedure (Section 6.5). It has two parts. Section 5.1 states *direct* (unanchored) parallel repetition in value form, for entangled and for commuting-operator strategies, together with Lin's tracial density theorem. These theorems are formalized: two external Lean developments proving them are vendored into this repository, and the section records their statements in the vocabulary of Section 2 together with the bridges connecting the two. Sections 5.2–5.4 state the *anchored*, entanglement-preserving theorem of Bavarian, Vidick and Yuen [7] used in [1]: repeating the anchored version of a game amplifies the completeness-soundness gap while preserving entanglement lower bounds. That theorem is a blueprint-only external result, and it is no longer on the path to the main theorem: the pipeline of Section 6 is stated in value form and uses the direct theorems (Remark 5.9). The anchored sections are kept as the record of the entanglement-form route of [1], whose remaining interest is the entanglement-requirement corollaries (Remark 6.8). Section 5.2 collects the background results the anchored proof relies on — entropy inequalities, Uhlmann's theorem — most of which are natural QuantumLib content, and some of which already exist there. Section 5.3 defines the anchoring transformation and states the theorem, and Section 5.4 decomposes the proof into its main steps.

5.1 Direct parallel repetition

The n-fold direct repetition of a game asks n independent question pairs at once and accepts if all n answer pairs are accepted. Two recent Lean developments prove that the value of the repeated game decays exponentially in n, for every game, with no anchoring and at a rate depending only on the gap $1 - \mathrm{val}(\mathfrak{G})$ and on the size of the answer alphabets: one for tensor-product (entangled) strategies, by OpenAI [30], and one for commuting-operator strategies [31], the latter building on the former and on Lin's tracial density theorem [9]. Both are vendored into this repository (`MIPRE/Background/Repetition/`, with provenance recorded in the `README.md` of each directory) and connected to the definitions of Section 2 by explicit bridges; the statements below are the bridged forms. The commuting-operator theorem is recorded now, although only the entangled one enters $\mathrm{MIP}^* = \mathrm{RE}$, because it is the form needed for the commuting-operator counterpart $\mathrm{MIP}^{\mathrm{co}} = \mathrm{coRE}$ [10] (Remark 8.5).

Definition 5.1 (Direct repetition)**.** *Let $\mathfrak{G} = (\mathcal{X}, \mathcal{Y}, \mathcal{A}, \mathcal{B}, \mu, D)$ be a game and $n \geq 0$ an integer. The* n-fold direct repetition *of $\mathfrak{G}$ is the game $\mathfrak{G}^n$ with question alphabets $\mathcal{X}^n$, $\mathcal{Y}^n$ and answer alphabets $\mathcal{A}^n$, $\mathcal{B}^n$, question distribution $\mu^n(\bar{x}, \bar{y}) = \prod_{i=1}^n \mu(x_i, y_i)$, and decision predicate $D^n(\bar{x}, \bar{y}, \bar{a}, \bar{b}) = \bigwedge_{i=1}^n D(x_i, y_i, a_i, b_i)$. The direct repetition of a synchronous game is synchronous, with the same data.*

The entangled theorem is stated for the quantum value of Definition 2.9, whose strategies are bipartite. The commuting-operator theorem needs the bipartite counterpart of the commuting value, which Section 2 defines in synchronous (tracial) form only (Definition 2.11); the two are related by Lemma 5.4.

Definition 5.2 (Commuting-operator strategy)**.** ↑ *QuantumLib* *A* commuting-operator strategy *for $\mathfrak{G}$ consists of a Hilbert space $\mathcal{H}$ (a complete inner product space over $\mathbb{C}$, of any dimension), a unit vector $|\psi\rangle \in \mathcal{H}$, and POVMs $\{E_a^x\}_{a \in \mathcal{A}}$ and $\{F_b^y\}_{b \in \mathcal{B}}$ on $\mathcal{H}$, one for each x (resp. y) — each E_a^x and F_b^y is a positive bounded operator, $\sum_a E_a^x = I$ and $\sum_b F_b^y = I$ — such that $E_a^x F_b^y = F_b^y E_a^x$ for all x, y, a, b. The players answer (a, b) to questions (x, y) with probability $\langle\psi| E_a^x F_b^y |\psi\rangle$.*

Definition 5.3 (Commuting-operator value)**.** ↑ *QuantumLib* *The value of a commuting-operator strategy $\mathscr{S} = (\mathcal{H}, \psi, E, F)$ in $\mathfrak{G}$ is*

$$\omega^{\mathrm{co}}(\mathfrak{G}, \mathscr{S}) = \sum_{x,y,a,b} \mu(x, y)\, D(x, y, a, b)\, \langle\psi| E_a^x F_b^y |\psi\rangle \ ,$$

and the commuting-operator value $\omega^{\mathrm{co}}(\mathfrak{G})$ *of $\mathfrak{G}$ is the supremum of $\omega^{\mathrm{co}}(\mathfrak{G}, \mathscr{S})$ over all commuting-operator strategies.*

Comments. As in Definition 2.9, the formalization takes the real part of the (real, nonnegative) probability. The Hilbert space of a strategy lives in the lowest universe, as in the vendored development; this is no restriction for the suprema, which are bounded by 1. The notation ω^{co} follows [31] and keeps the bipartite value apart from the tracial value $\mathrm{val}^{\mathrm{co}}$ of Definition 2.11.

● ● ○ medium ↑ QuantumLib

Lemma 5.4 (Tracial strategies are commuting-operator strategies)**.** ↑ *QuantumLib* *For every synchronous game $\mathfrak{G}$, $\mathrm{val}^{\mathrm{co}}(\mathfrak{G}) \leq \omega^{\mathrm{co}}(\mathfrak{G})$.*

Proof. Let $(\mathscr{A}, \tau, M)$ be a commuting strategy (Definition 2.5) and let $\mathcal{H} = L^2(\mathscr{A}, \tau)$ be the completion of $\mathscr{A}$ for the inner product $\langle a, b\rangle = \tau(a^*b)$, with the class of 1 as the unit vector ξ. Left multiplication $\pi(a) : b \mapsto ab$ and right multiplication $\rho(a) : b \mapsto ba$ extend to bounded operators on $\mathcal{H}$ (for ρ this uses traciality: $\|ba\|^2 = \tau(a^*b^*ba) = \tau(b^*b\,aa^*) \le \|a\|^2\,\|b\|^2$), commute with each other, and satisfy $\rho(p)^* = \rho(p^*)$ (again by traciality) so that ρ maps projections to projections. Setting $E_a^x = \pi(M_a^x)$ and $F_b^y = \rho(M_b^y)$ gives a commuting-operator strategy with $\langle\xi|E_a^x F_b^y|\xi\rangle = \langle 1, M_a^x M_b^y\rangle = \tau(M_a^x M_b^y)$, hence with the same value as $(\mathscr{A}, \tau, M)$. Taking suprema yields the claim. □

Comments. This is the direction needed to apply Theorem 5.5 to the tracial value used in the pipeline: a bound on $\omega^{\mathrm{co}}(\mathfrak{G}^n)$ is a bound on $\mathrm{val}^{\mathrm{co}}(\mathfrak{G}^n)$. The reverse comparison (from ω^{co} to $\mathrm{val}^{\mathrm{co}}$, which is false in general and true up to error for synchronous games) is the commuting-operator analogue of Theorem 3.3, proved by Lin [9]; it is part of the transport discussed in Section 1.3 and not of this section. The Lean statement of this lemma and its GNS construction are a sub-project of their own.

● ● ● hard ↑ QuantumLib

Theorem 5.5 (Direct parallel repetition for commuting-operator strategies). ↑ QuantumLib *There is a universal constant $c > 0$ such that for every game $\mathfrak{G}$ with nonempty alphabets and every integer $n \ge 1$, writing $\varepsilon = 1 - \omega^{\mathrm{co}}(\mathfrak{G})$,*

$$\omega^{\mathrm{co}}\big(\mathfrak{G}^n\big) \;\le\; \exp\Big(-\frac{c\,\varepsilon^7\,n}{\varepsilon + \log\big(|\mathcal{A}|\;|\mathcal{B}|\big)}\Big)\,.$$

Proof. This is the main theorem of [31] (Theorem 7.1 there), whose Lean proof is vendored at `MIPRE/Background/Repetition/CommutingRepetition/`. The bridge identifies a game of Definition 2.1 with the game of the vendored development having the $\{0,1\}$-valued payoff D, a commuting-operator strategy of Definition 5.2 with a strategy of the vendored development (field by field, so that the two suprema defining the value range over the same set of numbers), and the two direct repetitions with each other. When $\varepsilon = 0$ and $|\mathcal{A}| = |\mathcal{B}| = 1$ the formalization reads the quotient as 0 and the bound is trivial. □

Theorem 5.6 (Direct parallel repetition for entangled strategies). ↑ QuantumLib *There is a universal constant $c > 0$ such that for every game $\mathfrak{G}$ with nonempty answer alphabets and $\mathrm{val}^*(\mathfrak{G}) < 1$, and every integer $n \ge 1$, writing $\varepsilon = 1 - \mathrm{val}^*(\mathfrak{G})$,*

$$\mathrm{val}^*\big(\mathfrak{G}^n\big) \;\le\; \exp\Big(-\frac{c\,\varepsilon^{13}\,n}{\varepsilon + \log\big(|\mathcal{A}|\;|\mathcal{B}|\big)}\Big)\,.$$

Proof. This is the main theorem of Chapter 6 of [30], whose Lean proof is vendored at `MIPRE/Background/Repetition/`. Its entangled value is defined with a density matrix on the product of two finite sets and POVMs; Lemma 5.7 identifies it with val^* (Definition 2.9), after which the two direct repetitions are literally the same construction. The Lean proof is complete modulo that lemma. □

● ● ○ medium ↑ QuantumLib

Lemma 5.7 (Mixed states and POVMs do not change the quantum value). ↑ QuantumLib *Let $\mathrm{val}^*_{\mathrm{POVM}}(\mathfrak{G})$ be the supremum of the winning probabilities $\sum_{x,y,a,b} \mu(x,y)\,D(x,y,a,b)\,\mathrm{tr}\big((A_a^x \otimes B_b^y)\,\rho\big)$ over all finite sets S_{A}, S_{B}, density matrices ρ on $\mathbb{C}^{S_{\mathrm{A}}} \otimes \mathbb{C}^{S_{\mathrm{B}}}$ and POVMs $\{A_a^x\}$ on $\mathbb{C}^{S_{\mathrm{A}}}$, $\{B_b^y\}$ on $\mathbb{C}^{S_{\mathrm{B}}}$. Then $\mathrm{val}^*_{\mathrm{POVM}}(\mathfrak{G}) = \mathrm{val}^*(\mathfrak{G})$.*

Proof. A tensor-product strategy (Definition 2.3) is a strategy of the second kind, with $S_{\mathrm{A}} = \{1, \dots, d_{\mathrm{A}}\}$, $S_{\mathrm{B}} = \{1, \dots, d_{\mathrm{B}}\}$, $\rho = |\psi\rangle\langle\psi|$ and projective measurements, which gives $\mathrm{val}^* \leq \mathrm{val}^*_{\mathrm{POVM}}$. Conversely, given a strategy of the second kind, purify ρ on a reference system placed on Bob's side (Bob's POVM acts as the identity on it), then dilate the two POVMs to projective measurements on enlarged spaces (Naimark); neither step changes the outcome probabilities or the tensor-product structure, and reindexing the finite sets by initial segments of $\mathbb{N}$ yields a tensor-product strategy with the same value. □

Comments. The lemma is the only piece of the entangled bridge that is stated but not yet proved in Lean; it is a general fact of independent interest. The vendored definition of the entangled value has the strategy's two finite sets as arbitrary types in the lowest universe, and the bridge reindexes them.

• • • hard ↑ QuantumLib

Theorem 5.8 (Lin's tracial density theorem). ↑ *QuantumLib* *Let $\mathcal{X}$ and $\mathcal{A}$ be finite nonempty sets. Call a correlation $q(a, b \mid x, y)$ on question alphabets $\mathcal{X}, \mathcal{X}$ and answer alphabets $\mathcal{A}, \mathcal{A}$* tracially embeddable *if there are: a unital $*$-algebra $\mathscr{A}$ over $\mathbb{C}$ with a tracial state τ ($\tau(1) = 1$, $\tau(ab) = \tau(ba)$, $\tau(a^*) = \overline{\tau(a)}$); a Hilbert space $\mathcal{H}$ with a linear map $\iota : \mathscr{A} \to \mathcal{H}$ of dense range satisfying $\langle \iota a, \iota b\rangle = \tau(a^* b)$, on which $\mathscr{A}$ acts on the left by bounded operators $L(a)$ with $L(a)\,\iota b = \iota(ab)$; a positive element $\sigma \in \mathscr{A}$ with $\tau(\sigma^*\sigma) = 1$; POVMs $\{E^x_a\}_a \subseteq \mathscr{A}$ of positive elements summing to 1; and POVMs $\{G^y_b\}_b$ of positive bounded operators on $\mathcal{H}$ summing to I and commuting with $L(\mathscr{A})$, such that*

$$q(a, b \mid x, y) = \big\langle \iota\sigma,\ L(E^x_a)\, G^y_b\, \iota\sigma \big\rangle\ .$$

Then every commuting-operator correlation $p(a, b \mid x, y) = \langle\psi| E^x_a F^y_b |\psi\rangle$ on these alphabets (Definition 5.2) is an ℓ^1-limit of tracially embeddable correlations: for every $\delta > 0$ there is a tracially embeddable q with $\sum_{x,y,a,b} |p(a, b \mid x, y) - q(a, b \mid x, y)| < \delta$.

Proof. This is Theorem 3.2 of [9], as formalized in [31] (the density programme of the vendored development, `Tracial/Density/Main.lean`). The statement above is that of the vendored development, in its own vocabulary — its tracially embeddable correlations have the shape of Lin's Definition 3.1, with Bob's measurement taken in the commutant of the left action; restating it through the definitions of Section 2 is pending. □

Source. Direct parallel repetition for entangled strategies is Chapter 6 of OpenAI's *Ten advances* [30], with the Lean proof in the accompanying repository; for commuting-operator strategies it is [31], whose proof reduces to the tracially embeddable case by Theorem 5.8 and then rounds by adapting the argument of [30]. Earlier value-form theorems cover restricted classes of games or give polynomial rates; the anchored theorem of Section 5.3 is the one used in [1]. The rates ε^{13} and ε^7 are those of the sources; any polynomial rate suffices for the pipeline.

Comments. Both vendored developments are sorry-free and use only the standard axioms; the file `MIPRE/Background/Repetition/Axioms.lean` guards the axioms of the bridged commuting-operator theorem and of Theorem 5.8 (the entangled theorem joins it once Lemma 5.7 is proved). Vendoring is done by `scripts/vendor-repetition.py`, never by hand; the copies are read-only and the only local changes are recorded in each directory's `README.md`. Nothing outside `MIPRE/Background/Repetition/` refers to the vendored namespaces.

Remark 5.9 (Direct versus anchored repetition)**.** *Theorems 5.6 and 5.5 are statements about values, while Theorem 5.22 is a statement about entanglement requirements: the anchored proof extracts from a good strategy for the repeated game a strategy for the original game of* no larger dimension*, whereas the proofs of the direct theorems round through auxiliary resources (an embezzling state in [30], a tracial approximation in [31]) and give no control of the dimension. Lin's proof of* $\mathrm{MIP}^{\mathrm{co}} = \mathrm{coRE}$ *[10] shows that the entanglement clauses of [1] are dispensable: with completeness and soundness of compression stated in value form (perfect strategies are preserved, value at most* $\frac{1}{2}$ *is preserved), the halting reduction follows from the enumeration of Lemma 3.13 and Kleene's recursion theorem alone (Lemma 4.11), and gap amplification then needs only a value statement — exactly the theorems of this section, applied to the game as it is, without anchoring. The pipeline of Section 6 is stated in that form (Section 1.3): Theorem 6.5 rests on Theorem 5.6, and Theorem 5.22 with its toolkit (Sections 5.2–5.4) is not on the path to the main theorem; the entanglement clauses are recorded in Remark 6.8. In either form, the transport between bipartite and synchronous strategies (Section 1.3) is the pipeline's responsibility.*

5.2 Background results

The proof of Theorem 5.22 draws on a toolkit of quantum information theory (collected in [7, Section 5.1]) together with two classical probability lemmas of Holenstein. Everything in this subsection is standard and independent of nonlocal games; entries carry the same effort estimates as in Section 3.

5.2.1 Fidelity and Uhlmann's theorem

• • ○ medium ↑ QuantumLib

Definition 5.10 (Fidelity)**.** ↑ QuantumLib *The* fidelity *between positive semidefinite matrices* ρ, σ *on* $\mathbb{C}^d$ *is* $F(\rho,\sigma) = \left\|\sqrt{\rho}\sqrt{\sigma}\right\|_1$*, where* $\|A\|_1 = \operatorname{tr}\sqrt{A^\dagger A}$ *denotes the trace norm.*

Theorem 5.11 (Uhlmann)**.** ↑ QuantumLib *Let* $|\psi\rangle, |\varphi\rangle \in \mathcal{H}_A \otimes \mathcal{H}_B$ *be unit vectors and let* ρ, σ *denote their reduced density matrices on* $\mathcal{H}_A$*. Then there exists a unitary* V *on* $\mathcal{H}_B$ *such that*

$$\langle\varphi|\big(I \otimes V\big)|\psi\rangle = F(\rho,\sigma)\ .$$

Lemma 5.12 (Fuchs–van de Graaf)**.** ↑ QuantumLib *For all density matrices* ρ, σ *on* $\mathbb{C}^d$*,*

$$1 - F(\rho,\sigma) \ \le\ \frac{1}{2}\left\|\rho - \sigma\right\|_1 \ \le\ \sqrt{1 - F(\rho,\sigma)^2}\ .$$

Source. Uhlmann [32]; Fuchs and van de Graaf [33]. The finite-dimensional statements above are Theorem 9.2.1 and Section 9.3 of [35].

Comments. This is the tool that converts closeness of *reduced* states into *local unitaries*: combined with $\||\psi\rangle - |\varphi\rangle\|^2 = 2(1 - \mathrm{Re}\langle\varphi|\psi\rangle)$, Uhlmann's theorem turns a fidelity lower bound between reduced density matrices into a unitary on the complementary system carrying one purification close to the other. The proof of Lemma 5.24 applies it three times. The finite-dimensional proof is a page of linear algebra (Schmidt decomposition, polar decomposition, Cauchy–Schwarz) and, like everything in this subsection, should be stated for matrices, with no game-theoretic content.

5.2.2 The relative entropy toolkit

• • ○ medium ↑ QuantumLib

Definition 5.13 (Relative entropy and mutual information). ↑ QuantumLib *For positive semidefinite ρ, σ on $\mathbb{C}^d$, the* relative entropy *is $D(\rho \,\|\, \sigma) = \operatorname{tr}\big(\rho(\log \rho - \log \sigma)\big)$ if the support of ρ is contained in that of σ, and $+\infty$ otherwise; the* relative min-entropy *is $D_\infty(\rho \,\|\, \sigma) = \min\{\lambda : \rho \le 2^\lambda \sigma\}$, and $+\infty$ if no such λ exists. For a bipartite state ρ^{AB} the* mutual information *is $I(A:B)_\rho = D(\rho^{AB} \,\|\, \rho^A \otimes \rho^B)$, and for a tripartite state ρ^{ABC} the* conditional mutual information *is $I(A:B\,|\,C)_\rho = I(A:BC)_\rho - I(A:C)_\rho$.*

Lemma 5.14 (Basic properties of the relative entropy). ↑ QuantumLib *For all density matrices on finite-dimensional spaces, with subsystems as indicated:*

1. *(**Nonnegativity**) $D(\rho \,\|\, \sigma) \ge 0$;*
2. *(**Monotonicity under partial trace**) $D(\rho^X \,\|\, \sigma^X) \le D(\rho^{XY} \,\|\, \sigma^{XY})$;*
3. *(**Composition with the min-entropy**) if $D(\rho \,\|\, \sigma) \le \lambda_1$ and $D_\infty(\sigma \,\|\, \tau) \le \lambda_2$ then $D(\rho \,\|\, \tau) \le \lambda_1 + \lambda_2$;*
4. *(**Mutual information minimizes over product states**) for all density matrices σ^A, τ^B, $D(\rho^{AB} \,\|\, \sigma^A \otimes \tau^B) \ge I(A:B)_\rho$.*

Lemma 5.15 (Quantum Pinsker inequality). ↑ QuantumLib *For all density matrices ρ, σ on $\mathbb{C}^d$,*

$$\frac{1}{2 \ln 2} \|\rho - \sigma\|_1^2 \;\le\; D(\rho \,\|\, \sigma)\,.$$

Lemma 5.16 (Chain rule over a classical register). ↑ QuantumLib *Let $\rho = \sum_x P(x)|x\rangle\langle x| \otimes \rho_x$ and $\sigma = \sum_x Q(x)|x\rangle\langle x| \otimes \sigma_x$ be classical-quantum states. Then*

$$D(\rho \,\|\, \sigma) = D(P \,\|\, Q) + \mathop{\mathbb{E}}_{x \sim P} D(\rho_x \,\|\, \sigma_x)\,,$$

where $D(P \,\|\, Q)$ is the Kullback–Leibler divergence. In particular, for a classical-quantum state ρ^{XE}, $I(X:E)_\rho = \mathbb{E}_x\, D(\rho_x^E \,\|\, \rho^E)$.

Lemma 5.17 (Strong subadditivity). ↑ QuantumLib *For all tripartite density matrices ρ^{ABC}, $I(A:B\,|\,C)_\rho \ge 0$.*

Source. Standard; the statements above are exactly the toolkit collected in [7, Section 5.1], with textbook treatment in [35, Chapter 11]. Strong subadditivity is due to Lieb and Ruskai [34].

Comments. Core QuantumLib material, and a substantial part of it *already exists* there (including strong subadditivity and much of the relative entropy machinery), so the formalization task is largely one of interfacing: fixing a convention for classical-quantum states, conditioning on classical registers, and expectation notation. Strong subadditivity is the only deep theorem in the list; everything else is a page or two of matrix analysis.

5.2.3 Quantum Raz's lemma

● ○ ○ easy (given the toolkit) ↑ QuantumLib

Lemma 5.18 (Quantum Raz's lemma). ↑ *QuantumLib* *Let* $\rho = \rho^{X_1 \cdots X_n A}$ *and* $\sigma = \sigma^{X_1} \otimes \cdots \otimes \sigma^{X_n} \otimes \sigma^A$ *be states with* $X = X_1 \cdots X_n$ *classical in both. Then*

$$\sum_{i=1}^{n} I(X_i : A)_\rho \;\le\; D\big(\rho^{XA} \,\|\, \sigma^{XA}\big) \,.$$

Source. [7, Lemma 5.10]; it is the quantum analogue of the classical lemma at the heart of Raz's and Holenstein's proofs of parallel repetition [27, 28]. The proof is a page, by induction from the chain rule (Lemma 5.16), item 4 of Lemma 5.14, and strong subadditivity (Lemma 5.17).

Comments. The information-theoretic heart of the argument. In the application, ρ is the state of the players' question tuple together with one player's entanglement register, conditioned on winning the coordinates in a set C; σ is the corresponding unconditioned (product) state; and the relative entropy between the two is at most $\log \frac{1}{\Pr(W_C)}$ plus an answer-length term. The lemma then says that *most coordinates* i carry almost no information about the entanglement register. Reusable in any parallel-repetition or direct-product argument, hence a prime upstreaming candidate.

5.2.4 Conditioning independent random variables

● ○ ○ easy

Lemma 5.19 (Holenstein's conditioning lemmas). *Write* $\|P - Q\|$ *for the total variation distance between distributions. Let* $U = (U_1, \ldots, U_m)$ *be independent random variables and* W *an event with* $\Pr(W) > 0$. *Then*

$$\sum_{i=1}^{m} \big\| P_{U_i|W} - P_{U_i} \big\|^2 \;\le\; \log \frac{1}{\Pr(W)} \,.$$

More generally, let T, V *be random variables such that* $P_{TUV} = P_T \prod_i P_{U_i|T} \cdot P_{V|TU}$ *(the* U_i *are independent conditioned on* T*). Then*

$$\frac{1}{m} \sum_{i=1}^{m} \big\| P_{TU_iV|W} - P_{TV|W}\, P_{U_i|T} \big\| \;\le\; \sqrt{\frac{1}{m}\Big(\log|\mathcal{V}| + \log \frac{1}{\Pr(W)}\Big)} \,,$$

where $\mathcal{V}$ *is the support of* V.

Source. Holenstein [28], Lemma 4.1 and Corollary 4.3; restated in the form used here in [7, Section 4.3].

Comments. Purely classical probability (the proof is Kullback–Leibler divergence, classical Pinsker, and Jensen); the natural home is Mathlib rather than QuantumLib, which is why the entry carries no upstreaming badge. The argument also uses, silently, the basic calculus of total variation distance: marginalization is contractive, and appending a shared conditional kernel to both distributions preserves the distance.

Finally, the proof freely uses elementary facts about pure states that should be stated once and for all: every bipartite pure state can be brought to the symmetric Schmidt form $\sum_j \sqrt{\lambda_j}|v_j\rangle|v_j\rangle$ by a local unitary; for unit vectors, $\|\psi - \varphi\|_1 \le 2\,\||\psi\rangle - |\varphi\rangle\|$; the trace distance is contractive under measurements; and the "transpose trick" relating operators applied to either half of a state in symmetric Schmidt form. These are one-line lemmas of Mathlib-style linear algebra.

5.3 The anchoring transformation and the theorem

• ○ ○ easy ↑ QuantumLib

Definition 5.20 (Anchoring). ↑ QuantumLib *Let $\mathfrak{G} = (\mathcal{X}, \mathcal{Y}, \mathcal{A}, \mathcal{B}, \mu, D)$ be a game and $0 < \alpha < 1$. The* α-anchored version *of $\mathfrak{G}$ is the game $\mathfrak{G}_\perp$ with question alphabets $\mathcal{X} \cup \{\perp\}$ and $\mathcal{Y} \cup \{\perp\}$ and the same answer alphabets, in which the referee samples $(x, y) \sim \mu$ and then, independently for each player, replaces that player's question by the anchor symbol $\perp$ with probability α; the decision predicate accepts whenever either question is $\perp$, and otherwise applies D. Unless indicated otherwise, $\mathfrak{G}_\perp$ denotes the $\frac{1}{2}$-anchored version.*

Lemma 5.21 (Anchoring preserves the value). ↑ QuantumLib *For every game $\mathfrak{G}$ and $0 < \alpha < 1$,*

$$\mathrm{val}^*(\mathfrak{G}_\perp) = 1 - (1-\alpha)^2\big(1 - \mathrm{val}^*(\mathfrak{G})\big)\,.$$

Moreover the relation holds at the level of strategies, without change of dimension: extending a strategy for $\mathfrak{G}$ by arbitrary answers on $\perp$, or restricting a strategy for $\mathfrak{G}_\perp$ to the original questions, translates value p in one game into value $1-(1-\alpha)^2(1-p)$ in the other. The same relation holds for the classical value.

Proof. With probability $1-(1-\alpha)^2$ at least one question is $\perp$ and the referee accepts regardless of the answers; conditioned on the complement, the question pair is distributed according to μ and the predicate is that of $\mathfrak{G}$. Extending a value-p strategy for $\mathfrak{G}$ by a fixed answer on $\perp$ therefore achieves $1-(1-\alpha)^2(1-p)$ in $\mathfrak{G}_\perp$, and restricting a strategy for $\mathfrak{G}_\perp$ to the questions of $\mathfrak{G}$ — same state, same measurements, so the same dimension — inverts the relation. Taking suprema over strategies yields the claim. □

The theorem below is the form of the parallel repetition theorem invoked by the compression procedure of [1]. It is an entanglement statement, not merely a value statement: this is what soundness of the compression recursion requires when it is stated in terms of Ent (Remark 6.8); the value-form pipeline of Section 6 uses Theorem 5.6 instead.

• • • hard ↑ QuantumLib

Theorem 5.22 (Bavarian–Vidick–Yuen). *There exists a universal constant $c > 0$ such that for every two-player game $\mathfrak{G}$, all positive integers k, all $0 < \varepsilon \le 1$, and all p satisfying*

$$p > \frac{4}{\varepsilon}\exp\!\Big(-\frac{c\,\varepsilon^{17}\,k}{s}\Big)\,,$$

where s is the bit-length of the players' answers in $\mathfrak{G}$ and $\mathfrak{G}_\perp$ is the anchored version of $\mathfrak{G}$ (Definition 5.20, with $\alpha = \frac{1}{2}$), it holds that

$$\mathrm{Ent}\big(\mathfrak{G}_\perp^k, p\big) \ge \mathrm{Ent}(\mathfrak{G}, 1-\varepsilon)\,.$$

Source. Bavarian, Vidick and Yuen [7]; the entanglement-preserving form quoted here is the one used in [1, Section 10]. The paper proves the theorem for a general anchoring parameter α, with an α^{48} dependence in the exponent that is absorbed into c at $\alpha = \frac{1}{2}$; passing from $\mathrm{Ent}(\mathfrak{G}_\perp, 1-\varepsilon')$ to $\mathrm{Ent}(\mathfrak{G}, 1-\varepsilon)$ costs only the constant factor of Lemma 5.21, likewise absorbed. An alternate proof of the value form of the theorem was given by Jain and Kundu [29] and may be worth evaluating as a formalization route.

Comments. Used for gap amplification: after answer reduction the completeness-soundness gap has shrunk to 1 vs. $1 - 1/\mathrm{poly}$, and repetition of the anchored game restores the gap to 1 vs. $\frac{1}{2}$ while preserving entanglement lower bounds. The proof is information-theoretic and long; axiomatize first. Direct, value-form repetition is formalized (Section 5.1) and has replaced this theorem in the pipeline (Remark 5.9); this theorem is retained for the entanglement form of the argument (Remark 6.8). Two translation points for the formalization: the source works with bipartite tensor-product strategies while Ent (Definition 2.12) is defined via synchronous strategies, so the statement must be transported, consistent with the convention of Section 1.3; and the anchored version of a synchronous game is not literally synchronous (it accepts unequal answers on the question pair $(\perp, \perp)$), which is repaired by fixing a canonical accepted answer on $\perp$. The exponent 17 and the constants are provisional, as per Section 1.4.

5.4 Decomposition of the proof

This subsection records the structure of the proof of Theorem 5.22 following [7], at the level of its two main intermediate statements. It is not yet a formalization roadmap, but it identifies where each background result of Section 5.2 enters.

Shape of the argument. Like all known proofs of parallel repetition, the proof is a *rounding* argument. Let $\mathscr{S}$ be a dimension-d strategy for the repeated game $\mathfrak{G}_\perp^n$ with value at least p. One shows: if p is above the threshold of the theorem, then there is a strategy for $\mathfrak{G}_\perp$ of dimension *at most* d with value at least $1 - \varepsilon$. By Lemma 5.21 this in turn yields a strategy for $\mathfrak{G}$, still of dimension at most d, with value $1 - O(\varepsilon)$; the entanglement inequality of the theorem is the contrapositive. Everything hinges on the extracted strategy reusing the *same* entangled state (suitably measured) and the same measurements (suitably conjugated), so that the dimension does not grow.

A counting argument ([7, Proposition 6.5]) reduces the problem to a conditional one: there exists a set $C \subseteq [n]$ of size $O\big(\frac{1}{\varepsilon} \log \frac{1}{\varepsilon\, p}\big)$ such that, conditioned on the event W_C of winning all coordinates in C, a uniformly random other coordinate i is won with probability at least $1 - \varepsilon/2$. The players would therefore like to play coordinate i of $\mathfrak{G}_\perp^n$ "conditioned on W_C". The obstruction is that neither the conditioned question distribution nor the conditioned post-measurement state is locally sampleable. All error terms are controlled by the single quantity

$$\delta = \frac{1}{n - |C|} \Big(\log \frac{1}{\Pr(W_C)} + |C| \log |\mathcal{A}|\, |\mathcal{B}| \Big) , \tag{1}$$

whose answer-length term is where the parameter s of Theorem 5.22 enters.

The classical step: dependency-breaking variables. Following Raz and Holenstein, one introduces for each coordinate $i \notin C$ a *dependency-breaking variable* $\Omega_i = (D_i, M_i)$: D_i points to one of the two players uniformly at random, and M_i is a noisy copy of that player's question, where the noise rate $\eta = \alpha/2$ is tuned to the anchoring probability so that the marginal of the

question pair remains that of $\mathfrak{G}_\perp$. Conditioned on Ω_i the two players' questions in coordinate i are *independent*. Write R_{-i} for the tuple collecting the Ω_j for $j \notin C \cup \{i\}$ together with the questions and answers in the coordinates of C: this is the shared classical data that the extracted strategy will (non-uniformly) fix.

• • ○ medium

Lemma 5.23 (Conditioning does not skew individual coordinates)**.** *Fix a strategy for $\mathfrak{G}_\perp^n$, a set $C \subseteq [n]$ with $\Pr(W_C) > 0$, and let δ be as in (1) and $m = n - |C|$. Then, writing $\mathbb{E}_i$ for the expectation over a uniform $i \notin C$:*

1. $\mathbb{E}_i \left\| P_{\Omega_i X_i Y_i | W_C} - P_{\Omega_i X_i Y_i} \right\| \le \sqrt{\delta}$;
2. $\mathbb{E}_i \left\| P_{X_i Y_i}\, P_{R_{-i} | X_i = \perp, Y_i = \perp, W_C} - P_{X_i Y_i}\, P_{R_{-i} | X_i, Y_i, W_C} \right\| = O\big(\sqrt{\delta}/\alpha^2\big)$.

Source. [7, Lemma 4.6], items condensed; the proof is classical probability from Lemma 5.19.

Comments. Item 1 says conditioning on W_C barely disturbs any single coordinate; item 2 says the shared data R_{-i} can be sampled without knowing the players' actual questions (x, y) — by pretending both are the anchor — which is the classical half of the correlated sampling problem. This is where anchoring first pays: switching a question to $\perp$ costs only a factor $\mathrm{poly}(1/\alpha)$ because every question profile has the anchor in its support.

The quantum step: dependency-breaking states. The quantum analogue must also "condition the entangled state on W_C". For a realization $r_{-i} = (\omega_{-i}, a_C, b_C)$ of R_{-i} and questions x, y for coordinate i, define the *dependency-breaking state*

$$|\tilde{\Phi}_{r_{-i},x,y}\rangle \;\propto\; \big(A_{\omega_{-i},x}(a_C)\big)^{1/2} \otimes \big(B_{\omega_{-i},y}(b_C)\big)^{1/2} |\psi\rangle \,,$$

where $A_{\omega_{-i},x}(a_C)$ denotes the POVM element of the strategy producing answers a_C in the coordinates of C, averaged over all questions outside coordinate i consistent with ω_{-i} and with $X_i = x$ (and symmetrically for B). This is the post-measurement state of $|\psi\rangle$ given that the answers on C were (a_C, b_C), as seen when coordinate i carries (x, y). The extracted strategy would like to hold $|\tilde{\Phi}_{r_{-i},x,y}\rangle$, but the state depends on *both* questions; the key technical statement is that the players can reach it from the question-independent state $|\tilde{\Phi}_{r_{-i},\perp,\perp}\rangle$ by *local* unitaries.

• • • hard

Lemma 5.24 (Existence of local unitaries)**.** *In the setting above there exist, for every $i \notin C$, r_{-i}, x and y, unitaries $U_{r_{-i},x}$ on the first player's space and $V_{r_{-i},y}$ on the second player's space such that*

$$\mathop{\mathbb{E}}_{i} \mathop{\mathbb{E}}_{r_{-i} | W_C} \mathop{\mathbb{E}}_{(x,y)} \Big\| \big(U_{r_{-i},x} \otimes V_{r_{-i},y}\big) |\tilde{\Phi}_{r_{-i},\perp,\perp}\rangle - |\tilde{\Phi}_{r_{-i},x,y}\rangle \Big\| = O\big(\delta^{1/16} \alpha^{-3}\big) \,,$$

where (x, y) is distributed according to the question distribution of $\mathfrak{G}_\perp$.

Source. [7, Proposition 5.1], proved in Sections 5.2–5.4 there.

Comments. This is the technical core of the paper, and the second place anchoring pays: the state is moved one question at a time, using the anchor as a "home base" connected to every question ($\tilde{\Phi}_{\perp,\perp} \to \tilde{\Phi}_{x,\perp} \to \tilde{\Phi}_{x,y}$). Each switch is an application of Uhlmann's theorem (Theorem 5.11, via Lemma 5.12): the required closeness of reduced density matrices comes from quantum Raz's lemma (Lemma 5.18) applied to the state of questions and one player's register conditioned on W_C, converted to trace distance by Pinsker's inequality (Lemma 5.15). A separate argument compares the normalization factors of the unnormalized states (γ-comparison, [7, Lemma 5.17]). The exponent $\frac{1}{16}$ is an artifact of chaining these approximations and is provisional.

Assembly. Given Lemmas 5.23 and 5.24, the extracted strategy for $\mathfrak{G}_\perp$ is: fix (by averaging) a good coordinate i and realization r_{-i}; share the state $|\tilde{\Phi}_{r_{-i},\perp,\perp}\rangle$; on questions (x, y), apply $U_{r_{-i},x}$ and $V_{r_{-i},y}$; then measure with the original measurements for coordinate i, conditioned on the fixed answers (a_C, b_C) (a renormalized, "pretty good" version of the strategy's POVMs). The main lemma of [7, Section 6] shows the resulting question–answer distribution is $O(\delta^{1/16}/\alpha^3)$-close in total variation to $P_{X_iY_iA_iB_i|W_C}$, whose winning probability is at least $1 - \varepsilon/2$ by the counting argument; the parameters of Theorem 5.22 are set so that the total loss is at most ε. The state and measurements act on the original d-dimensional spaces, so the extracted strategy has dimension at most d, giving the entanglement form of the theorem.

6 Structure of the proof

This chapter records the main transformations of [1] at headline level: for each, what it does, what it relies on, and one theorem statement so that the dependency graph reflects the architecture. Working out each section — restating it in the synchronous framework, decomposing it into lemmas, and setting the constants — is a main task of the project. The paper factors several of these transformations through *typed* verifiers and a detyping step; that machinery is not reproduced here and should be introduced locally where needed.

Following Lin [10], every soundness statement below is in *value form*: a transformation preserves the property "value at most $1-\varepsilon$" up to a quantitative loss, and compression preserves "value at most $\frac{1}{2}$". No entanglement lower bound is tracked. Remark 5.9 explains why none is needed, Section 4.3 states the abstract lemma (Lin's compressibility criterion) that turns value-form compression into the halting reduction, and the entanglement clauses of [1] are recorded for reference in Remark 6.8.

6.1 Conditionally linear functions and samplers

Question distributions must be structured enough that players can sample them *on their own* in a certifiable way (introspection), yet expressive enough to implement all the verifiers in the pipeline. The right class is that of conditionally linear distributions: linear maps applied in stages, where each stage may depend on the output of the previous ones.

Definition 6.1 (Conditionally linear function)**.** *Let $V = \mathbb{F}_q^s$ and $\ell \geq 1$. An* ℓ-level conditionally linear (CL) function *on V is defined recursively: a 1-level CL function is a linear map $L : V \to V$; an ℓ-level CL function is given by a direct sum decomposition $V = V_1 \oplus V_{>1}$, a linear map L_1 on V_1, and, for each value u of L_1, an $(\ell-1)$-level CL function $L^u_{>1}$ on $V_{>1}$; it maps $z = z_1 + z_{>1}$ to $L_1(z_1) + L^{L_1(z_1)}_{>1}(z_{>1})$. A* CL distribution *is the distribution of $(L^{\mathrm{A}}(z), L^{\mathrm{B}}(z))$ for uniformly random $z \in V$ and CL functions $L^{\mathrm{A}}, L^{\mathrm{B}}$.*

An ℓ-level sampler (Definition 2.25) presents a CL distribution via Turing machines answering structured queries (dimensions, marginals, and evaluations of the linear maps and their canonical complements). This section of the project should fix the query interface once and for all, and prove the closure properties used later: products, concatenation, and reduction of the field size to $q = 2$ ("downsizing", using Lemma 3.11).

6.2 Question reduction: introspection

Introspection removes the verifier's cost of sampling questions: instead of receiving questions for V_{2^n}, the players *sample the questions themselves* by measuring shared EPR pairs, and the new verifier uses the Pauli basis test (Theorem 3.8) to force honest sampling. The CL structure of the sampler is what makes commitment to each stage of the sampled question enforceable (the "hiding" directions of the CL decomposition). The result is a verifier whose questions are exponentially smaller, at the price of a fixed number of levels.

Theorem 6.2 (Introspection)**.** *There is a polynomial-time Turing machine* ComputeIntroVerifier *that on input $(\mathsf{V}, \lambda, \ell)$ returns a 5-level normal form verifier $\mathsf{V}^{\textsc{Intro}} = (\mathcal{S}^{\textsc{Intro}}, \mathcal{D}^{\textsc{Intro}})$ with $\mathsf{TIME}_{\mathcal{S}^{\textsc{Intro}}}(n) = \mathrm{poly}(n, \lambda, \ell)$, $\mathsf{TIME}_{\mathcal{D}^{\textsc{Intro}}}(n) = \mathrm{poly}(2^{\lambda n}, \ell)$ and $\left|\mathcal{D}^{\textsc{Intro}}\right| = \mathrm{poly}(\lambda, \ell)$, such that for every λ-bounded ℓ-level verifier V there is a function $\delta(\varepsilon, n) = a\big((\lambda n)^a \varepsilon^b + (\lambda n)^{-b}\big)$ (constants a, b depending only on ℓ) with, for all n:*

1. *(**Completeness**) If V_{2^n} has a value-1 PCC strategy, so does $\mathsf{V}_n^{\textsc{Intro}}$.*
2. *(**Soundness**) If $\mathrm{val}^s(\mathsf{V}_n^{\textsc{Intro}}) > 1 - \varepsilon$ then $\mathrm{val}^s(\mathsf{V}_{2^n}) \geq 1 - \delta(\varepsilon, n)$.*

The soundness analysis is a long rigidity argument [1, Section 8]. It builds on techniques introduced in [3], and it may be adviseable to formalize some of the key lemmas there first. These include the Pauli basis test together with a toolkit of approximate-measurement lemmas (twirling, commutation, sandwiching). For this project, those lemmas should be stated and proved in the tracial/cynchronous setting.

6.3 Oracularization

Oracularization converts a normal form verifier into one where a single player (the "oracle") receives both original questions and answers both, while the other player receives one of the two questions and must answer consistently. This makes the game's predicate a pointwise function of one player's answer — the format required by answer reduction. This is probably the easiest transformation in the paper and could be attempted first to fix the semantics.

Theorem 6.3 (Oracularization)**.** *There is a polynomial-time Turing machine* ComputeOracleVerifier *mapping a normal form verifier $\mathsf{V} = (\mathcal{S}, \mathcal{D})$ to a normal form verifier $\mathsf{V}^{\textsc{Orac}}$ such that, for some $\delta(\varepsilon) = \mathrm{poly}(\varepsilon)$ and all n:*

1. *(**Completeness**) If V_n has a value-1 PCC strategy then $\mathsf{V}_n^{\textsc{Orac}}$ has a value-1 symmetric PCC strategy.*
2. *(**Soundness**) If $\mathrm{val}^s(\mathsf{V}_n^{\textsc{Orac}}) > 1 - \varepsilon$ then $\mathrm{val}^s(\mathsf{V}_n) \geq 1 - \delta(\varepsilon)$.*
3. *(**Complexity**) The sampler of $\mathsf{V}^{\textsc{Orac}}$ depends only on $\mathcal{S}$, with TIME within a constant factor and the same number of levels.*

6.4 Answer reduction

After introspection the decider still runs in time exponential in n (it simulates $\mathcal{D}$ at index 2^n), and the players' answers are correspondingly long. Answer reduction composes the game with a PCP: by Theorem 3.10 the statement "the original decider would have accepted our answers" is a succinct 3SAT instance; the players commit to a low-degree encoding of a satisfying assignment (their original answers plus the computation tableau) and the new decider spot-checks it, using the low individual degree test to certify the encoding and Lemma 3.12 for the arithmetization. Answers and decider time become polylogarithmic in the original.

Theorem 6.4 (Answer reduction)**.** *There is a polynomial-time Turing machine* $\mathsf{ComputeAnsRedVerifier}$ *that on input* $(\mathsf{V}, \lambda, \mu, \sigma)$*, for* V *an* ℓ*-level normal form verifier with* $|\mathcal{D}| \le \sigma$*,* $\mathsf{TIME}_{\mathcal{S}}(n) \le (\lambda n)^{\mu}$ *and* $\mathsf{TIME}_{\mathcal{D}}(n) \le (2^{\lambda n})^{\mu}$ *for* $n \ge 2$*, outputs a* $\max\{\ell + 2, 5\}$*-level normal form verifier* V^{ar} *with* $\mathsf{TIME}_{\mathcal{S}^{\mathsf{ar}}}(n), \mathsf{TIME}_{\mathcal{D}^{\mathsf{ar}}}(n) = \mathrm{poly}\big((\lambda n)^{\mu}, \sigma\big)$*, whose sampler depends only on* $(\mathcal{S}, \lambda, \mu, \sigma)$*, and such that for some* $\delta(\varepsilon, n) = \sigma^{a}\big((\lambda n)^{\mu a}\varepsilon^{b} + (\lambda n)^{-\mu b}\big)$ *(universal constants* a, b*) and all* n*:*

1. *(**Completeness**) If* V_n *has a value-*1 *symmetric PCC strategy then* $\mathsf{V}^{\mathsf{ar}}_n$ *has a value-*1 *PCC strategy.*
2. *(**Soundness**) If* $\mathrm{val}^{s}(\mathsf{V}^{\mathsf{ar}}_n) > 1 - \varepsilon$ *then* $\mathrm{val}^{s}(\mathsf{V}_n) \ge 1 - \delta(\varepsilon, n)$*.*

This is the most technically heterogeneous section of the paper ([1, Section 10]): half classical (tableaux, Tseitin encoding, arithmetization — see Section 3.8) and half quantum (a low-degree "sandwiching" argument recombining the players' committed polynomials). It may also be the hardest.

6.5 Gap amplification: parallel repetition

Answer reduction leaves a shrunken gap (1 vs. $1 - 1/\mathrm{poly}(n)$). Direct parallel repetition (Theorem 5.6, formalized) restores soundness $\frac{1}{2}$ with polynomially many repetitions, at polynomial cost in complexity and without anchoring; the anchored, entanglement-form theorem used in [1] (Theorem 5.22) is not needed (Remark 5.9).

Theorem 6.5 (Parallel repetition of normal form verifiers)**.** *There are universal constants* $b, c, c' > 0$ *and a polynomial-time Turing machine* $\mathsf{ComputeRepeatedVerifier}$ *that on input* $(\mathsf{V}, \lambda, \tau)$ *outputs a normal form verifier* $\mathsf{V}^{\mathrm{Rep}}$ *whose game* $\mathsf{V}^{\mathrm{Rep}}_n$ *is the* $k(n)$*-fold direct repetition (Definition 5.1) of* V_n*, with* $k(n) = (\lambda n)^{(1+c')\tau}$*, such that if* V *is* ℓ*-level then* $\mathsf{V}^{\mathrm{Rep}}$ *is* ℓ*-level with* $\mathsf{TIME}_{\mathcal{S}^{\mathrm{Rep}}}(n) = O(k(n)\mathsf{TIME}_{\mathcal{S}}(n))$ *and* $\mathsf{TIME}_{\mathcal{D}^{\mathrm{Rep}}}(n) = O\big(k(n)\max(\mathsf{TIME}_{\mathcal{D}}(n), (\lambda n)^{\tau})\big)$*, the sampler of* $\mathsf{V}^{\mathrm{Rep}}$ *depends only on* $(\mathcal{S}, \lambda, \tau)$*, and for all* n*:*

1. *(**Completeness**) If* V_n *has a value-*1 *PCC strategy and* $\mathsf{TIME}_{\mathcal{D}}(n) \le (\lambda n)^{\tau}$*, then* $\mathsf{V}^{\mathrm{Rep}}_n$ *has a value-*1 *PCC strategy.*
2. *(**Soundness**) For all* $\varepsilon > 0$*, if* $\mathrm{val}^{s}(\mathsf{V}_n) \le 1 - \varepsilon$ *and* $\mathsf{TIME}_{\mathcal{D}}(n) \le (\lambda n)^{\tau}$*, then* $\mathrm{val}^{s}(\mathsf{V}^{\mathrm{Rep}}_n) \le \exp\big(-c\,\varepsilon^{b}\,k(n)/(\lambda n)^{\tau}\big)$*; in particular* $\mathrm{val}^{s}(\mathsf{V}^{\mathrm{Rep}}_n) \le \frac{1}{2}$ *whenever* $\varepsilon \ge (\lambda n)^{-\tau}$*, for* c' *large enough and* $n \ge 2$*.*

The game $\mathsf{V}^{\mathrm{Rep}}_n$ is $\mathsf{V}^{k(n)}_n$: its sampler runs $k(n)$ independent copies of $\mathcal{S}$ (a product of conditionally linear functions of the same level) and its decider runs $\mathcal{D}$ on every coordinate. Completeness: the $k(n)$-fold tensor power of a value-1 PCC strategy is a value-1 PCC strategy for the repeated game. Soundness: by the almost-synchronous transport (Theorem 3.3), $\mathrm{val}^{s}(\mathsf{V}_n) \le 1 - \varepsilon$ gives $\mathrm{val}^{*}(\mathsf{V}_n) \le 1 - \varepsilon'$ with $\varepsilon' = \Omega(\varepsilon^{a})$ for a universal a; Theorem 5.6 bounds $\mathrm{val}^{*}(\mathsf{V}^{k(n)}_n)$, where the

answer alphabets of V_n have size at most $2^{\mathsf{TIME}_{\mathcal{D}}(n)}$ so that $\log(|\mathcal{A}|\,|\mathcal{B}|) \le 2(\lambda n)^{\tau}$; and $\mathrm{val}^s \le \mathrm{val}^*$ (Lemma 2.13) concludes, with $b = 13a$. The exponent b and the constants are provisional (Section 1.4).

6.6 The compression theorem

Composing the four transformations — introspection, oracularization, answer reduction, repetition — yields gap-preserving compression: the paper-level form of the compression hypothesis of Lemma 4.11. The soundness clause is the value-form clause that [1] state first and then strengthen to an entanglement bound (Remark 6.8).

Theorem 6.6 (Gap-preserving compression)**.** *There is a universal constant $C_0 > 0$ and a polynomial-time Turing machine* Compress *that on input (V, λ) outputs a 9-level normal form verifier $\mathsf{V}^{\textsc{Compr}} = (\mathcal{S}^{\textsc{Compr}}, \mathcal{D}^{\textsc{Compr}})$ with $\mathsf{TIME}_{\mathcal{S}^{\textsc{Compr}}}(n), \mathsf{TIME}_{\mathcal{D}^{\textsc{Compr}}}(n) = \mathrm{poly}(n, \lambda)$, where $\mathcal{S}^{\textsc{Compr}}$ is independent of V and computable from λ in time $\mathrm{polylog}(\lambda)$. If moreover V is a λ-bounded 9-level normal form verifier, then for all $n \ge C_0$ and $N = 2^n$:*

1. *(**Completeness**) If V_N has a value-1 PCC strategy, then so does $\mathsf{V}_n^{\textsc{Compr}}$.*
2. *(**Soundness**) If $\mathrm{val}^s(\mathsf{V}_N) \le \frac{1}{2}$ then $\mathrm{val}^s(\mathsf{V}_n^{\textsc{Compr}}) \le \frac{1}{2}$.*

Remark 6.7 (Relation to the abstract compression lemma)**.** *Theorem 6.6 instantiates the hypotheses of Lemma 4.11: take A to be the set of (descriptions of) normal form verifier games with a value-1 PCC strategy and B the set of those with $\mathrm{val}^s \le \frac{1}{2}$, with the trivially accepting and the trivially rejecting game as the distinguished elements; the complement of B is semidecided by enumerating synchronous strategies (Lemma 3.13); and a description $(\mathcal{S}, \mathcal{D}, n)$ with n in binary is precisely a succinct description of the game V_{2^n} played by a λ-bounded verifier. Completeness of compression is the preservation of A and soundness the preservation of B. Making this dictionary precise (in particular the encoding of games by verifier descriptions and the role of λ) is a formalization task of this section. The entanglement form of [1] (Remark 6.8) instantiates Lemma 4.10 instead, with $f = \mathrm{Ent}(\cdot, \frac{1}{2})$; it is not used by the main theorem.*

Remark 6.8 (The entanglement form of [1])**.** *The paper states the soundness of each transformation with an additional entanglement clause, obtained from the same proofs by tracking the dimension of the extracted strategies:* $\mathrm{Ent}(\mathsf{V}_n^{\textsc{Intro}}, 1-\varepsilon) \ge \max\{\mathrm{Ent}(\mathsf{V}_{2^n}, 1-\delta(\varepsilon,n)),\ (1-\delta(\varepsilon,n))\,2^{2^{\lambda n}}\}$ *for introspection;* $\mathrm{Ent}(\mathsf{V}_n^{\textsc{Orac}}, 1-\varepsilon) \ge \mathrm{Ent}(\mathsf{V}_n, 1-\delta(\varepsilon))$ *for oracularization;* $\mathrm{Ent}(\mathsf{V}_n^{\mathsf{ar}}, 1-\varepsilon) \ge \mathrm{Ent}(\mathsf{V}_n, 1-\delta(\varepsilon,n))$ *for answer reduction; Theorem 5.22 for anchored repetition; and, composed,* $\mathrm{Ent}(\mathsf{V}_n^{\textsc{Compr}}, \frac{1}{2}) \ge \frac{1}{2}\max\{\mathrm{Ent}(\mathsf{V}_N, \frac{1}{2}),\ 2^{N^{\lambda}-1}\}$ *for compression. Fed into the variant of Lemma 4.10 tolerating the factor $\frac{1}{2}$ (harmless, since the second term grows much faster than the levels of the recursion), with $f = \mathrm{Ent}(\cdot, \frac{1}{2})$, this yields Theorem 6.9 with the stronger conclusion $\mathrm{Ent}(\mathfrak{G}_{\mathcal{M}}, \frac{1}{2}) = \infty$ for non-halting $\mathcal{M}$. The value form adopted here needs none of it; the entanglement clauses remain of interest for the corollaries of Remark 8.5.*

6.7 The halting reduction

Theorem 6.9 (Halting reduction)**.** *There is a computable (indeed polynomial-time) map $\mathcal{M} \mapsto \mathfrak{G}_{\mathcal{M}}$ from Turing machines to synchronous game descriptions, with games given by normal form verifiers satisfying the efficiency requirements of Definition 2.29, such that:*

1. *if $\mathcal{M}$ halts on the empty input then $\mathrm{val}^s(\mathfrak{G}_{\mathcal{M}}) = 1$, witnessed by a PCC strategy;*
2. *if $\mathcal{M}$ does not halt on the empty input then $\mathrm{val}^s(\mathfrak{G}_{\mathcal{M}}) \le \frac{1}{2}$.*

The proof is Lemma 4.11 applied to the instantiation of Remark 6.7; the semidecidability hypothesis of the lemma is Lemma 3.13. The entanglement form of the statement, with the stronger conclusion $\mathrm{Ent}(\mathfrak{G}_{\mathcal{M}}, \frac{1}{2}) = \infty$ for non-halting $\mathcal{M}$, is discussed in Remark 6.8.

7 The main theorem

The statement below is the formalization target of the whole project. It matches, definition for definition, the Lean file `MIPRE/HaltingGameValue.lean`, which states it on top of Mathlib alone (theorem `halting_reduces_to_gameValue`): Turing machines are Mathlib's `Nat.Partrec.Code`, games are game descriptions (Definition 2.24), and the value is the synchronous value (Definition 2.10).

Theorem 7.1 (Halting reduces to the game value; Theorem 12.2 of [1], synchronous form)**.** *There is a computable map g from Turing machines to game descriptions such that for every Turing machine $\mathcal{M}$:*

1. *(**Completeness**) if $\mathcal{M}$ halts on the empty input, then $\mathrm{val}^s(\mathfrak{G}_{g(\mathcal{M})}) = 1$;*
2. *(**Soundness**) if $\mathcal{M}$ does not halt on the empty input, then $\mathrm{val}^s(\mathfrak{G}_{g(\mathcal{M})}) \le \frac{1}{2}$.*

Theorem 7.1 follows from Theorem 6.9 by converting the normal form verifier at a fixed index into an explicit game description (a finite object obtained by running the sampler and decider), and relaxing polynomial-time computability of the reduction to computability, per Section 1.3.

Corollary 7.2 (Uncomputability of the game value)**.** *There is no algorithm that, given a game description g with the promise that $\mathrm{val}^s(\mathfrak{G}_g) = 1$ or $\mathrm{val}^s(\mathfrak{G}_g) \le \frac{1}{2}$, decides which is the case.*

Theorem 7.3 ($\mathrm{MIP}^* = \mathrm{RE}$)**.** $\mathrm{MIP}^* = \mathrm{RE}$.

For $\mathrm{RE} \subseteq \mathrm{MIP}^*$: the halting problem is RE-complete (Theorem 3.9) and Theorem 6.9 provides a polynomial-time verifier for it; the bridge results (Lemma 2.13 and Theorem 3.3) transfer the completeness–soundness gap from val^s to val^* as required by Definition 2.29. For $\mathrm{MIP}^* \subseteq \mathrm{RE}$: given a verifier for L and an input z, enumerate tensor-product strategies (Lemma 3.13) and accept upon finding one with value greater than $\frac{1}{2}$; this semi-decides L.

Remark 7.4. *The polynomial-time version of Theorem 7.1 (Theorem 12.2 of [1] as stated there) is recorded by Theorem 6.9; the computable version above is what the Lean main statement asserts, and is sufficient for every consequence in Section 8.*

8 Downstream results

Potential extensions of the project once the main theorem is in place. They are listed roughly in order of increasing formalization cost. The first two stay within finite-dimensional linear algebra plus elementary functional analysis, while the operator-algebraic consequences require von Neumann algebra infrastructure that largely does not yet exist in Mathlib and would constitute a major sub-project of independent value.

8.1 Uncomputability of the value

Corollary 7.2 is already a headline consequence requiring nothing further: no algorithm can approximate the (synchronous or quantum) value of a nonlocal game to within any constant better than $\frac{1}{2}$.

8.2 Separation of quantum and commuting values

Theorem 8.1 (Separation)**.** *There is an explicit synchronous game* $\mathfrak{G}^{\mathrm{SEP}}$ *with* $\mathrm{val}^*(\mathfrak{G}^{\mathrm{SEP}}) \leq \frac{1}{2}$ *and* $\mathrm{val}^{\mathrm{co}}(\mathfrak{G}^{\mathrm{SEP}}) = 1$*.*

Sketch: apply the reduction of Theorem 6.9 to a machine that searches for a proof of its own non-halting (or any non-halting machine whose game retains a perfect commuting-operator strategy), and track through the pipeline that completeness holds not just for PCC strategies but for commuting-operator (equivalently, tracial) strategies. This *commuting completeness* property is an additional obligation on Sections 6.2–6.6 that the blueprint should make explicit when those sections are worked out; it is the only new ingredient beyond the main theorem. An alternative route: if $\mathrm{val}^* = \mathrm{val}^{\mathrm{co}}$ for all games, then combining Lemma 3.13 (value approximable from below) with the NPA hierarchy [25] (commuting value approximable from above) would make the value computable, contradicting Corollary 7.2; this gives a non-explicit separation with much less work, and is the recommended first target.

8.3 Failure of Tsirelson's problem

Corollary 8.2 (Tsirelson's problem, negative answer)**.** *There exist finite question and answer sets for which the closure* C_{qa} *of the set of finite-dimensional quantum correlations is strictly contained in the set* C_{qc} *of commuting-operator correlations.*

Immediate from Theorem 8.1: the value gap is witnessed by a correlation in C_{qc} at distance bounded away from C_{qa}. Requires only the definitions of the correlation sets. (Non-closure of the unclosed set C_q itself is Slofstra's earlier theorem [13], which also follows, with quantitative bounds, from the machinery here.)

8.4 Refutation of Connes' Embedding Problem

Corollary 8.3 (Connes' Embedding Problem, negative answer)**.** *There is a separable* II_1 *factor (indeed, a tracial von Neumann algebra generated by finitely many projections) that does not embed into an ultrapower* $\mathcal{R}^\omega$ *of the hyperfinite* II_1 *factor.*

Source for the implication. Tsirelson's problem is equivalent to CEP by Fritz [15] and Junge et al. [14], with the converse direction completed by Ozawa [16]. In the synchronous framework the implication is particularly direct: a synchronous correlation is precisely given by projections in a tracial von Neumann algebra summing to 1 [18, 17], and C_{qa}-type correlations are exactly those arising from $\mathcal{R}^\omega$-embeddable algebras, so Corollary 8.2 for synchronous correlations yields a non-embeddable algebra.

Comments. This is the deepest formalization challenge on the list: it needs tracial von Neumann algebras, ultraproducts, and the synchronous-correlation correspondence. The synchronous framework adopted in this blueprint (Section 1.3) was chosen partly because it makes this bridge as short as it can be. A reasonable staging: (i) define C_{qa}, C_{qc} and synchronous correlations; (ii) prove the synchronous correlation–tracial algebra correspondence; (iii) formalize the ultraproduct $\mathcal{R}^\omega$ and the equivalence. Step (iii) alone is a landmark for operator algebras in Lean.

8.5 Failure of Kirchberg's QWEP conjecture

Corollary 8.4 (QWEP)**.** *Kirchberg's QWEP conjecture fails: there is a C^*-algebra that is not a quotient of a C^*-algebra with the weak expectation property; equivalently, $C^*(F_2) \otimes_{\min} C^*(F_2) \neq C^*(F_2) \otimes_{\max} C^*(F_2)$.*

Equivalent to CEP by Kirchberg's theorem (see [16]); no new input from the project, but the equivalence itself is another substantial operator-algebra formalization.

8.6 Further directions

Remark 8.5. *Other candidate consequences to record as the project matures: undecidability of the existence of* perfect *strategies for synchronous games and its group-theoretic reformulations (via the synchronous algebra of a game); quantitative non-closure of C_q and rates for the NPA hierarchy; uncomputability results for entanglement requirements (*$\mathrm{Ent}(\cdot, \frac{1}{2})$ *grows faster than any computable function along the halting games), which need the entanglement form of the pipeline (Remark 6.8) rather than the value form adopted here; the commuting-operator counterpart* $\mathrm{MIP}^{\mathrm{co}} = \mathrm{coRE}$ *[9, 10], for which direct parallel repetition for commuting-operator strategies (Theorem 5.5) and Lin's tracial density theorem (Theorem 5.8) are already formalized; and, more speculatively, connections to the undecidability of the spectral gap and to the existence of non-hyperlinear groups, the latter being* not *implied by CEP's failure and best recorded as an open problem.*

References

[1] Z. Ji, A. Natarajan, T. Vidick, J. Wright, H. Yuen. $\mathrm{MIP}^* = \mathrm{RE}$. arXiv:2001.04383, 2020 (revised 2022).

[2] Z. Ji, A. Natarajan, T. Vidick, J. Wright, H. Yuen. Quantum soundness of the classical low individual degree test. arXiv:2009.12982, 2020.

[3] A. Natarajan, J. Wright. $\mathrm{NEEXP} \subseteq \mathrm{MIP}^*$. In *Proc. 60th FOCS*, 2019. arXiv:1904.05870.

[4] W. T. Gowers, O. Hatami. Inverse and stability theorems for approximate representations of finite groups. *Sbornik: Mathematics*, 208(12):1784–1817, 2017. arXiv:1510.04085.

[5] M. de la Salle. Orthogonalization of positive operator valued measures. arXiv:2103.14126, 2021.

[6] T. Vidick. Almost synchronous quantum correlations. *Journal of Mathematical Physics*, 63(2):022201, 2022. arXiv:2103.02468.

[7] M. Bavarian, T. Vidick, H. Yuen. Hardness amplification for entangled games via anchoring. In *Proc. 49th STOC*, 2017. arXiv:1509.07466.

[8] A. Marks, S. S. Nezhadi, H. Yuen. The recursive compression method for proving undecidability results. Manuscript.

[9] J. Lin. Tracially embeddable strategies: lifting MIP^* tricks to $\mathrm{MIP}^{\mathrm{co}}$. arXiv:2304.01940, 2023.

[10] J. Lin. $\mathrm{MIP}^{\mathrm{co}} = \mathrm{coRE}$. In *Proc. 58th STOC*, 2026. arXiv:2510.07162.

[11] X. Wu, J.-D. Bancal, M. McKague, V. Scarani. Device-independent parallel self-testing of two singlets. *Physical Review A*, 93:062121, 2016. arXiv:1512.02074.

[12] J. Fitzsimons, Z. Ji, T. Vidick, H. Yuen. Quantum proof systems for iterated exponential time, and beyond. In *Proc. 51st STOC*, 2019. arXiv:1805.12166.

[13] W. Slofstra. The set of quantum correlations is not closed. *Forum of Mathematics, Pi*, 7:E1, 2019. arXiv:1703.08618.

[14] M. Junge, M. Navascués, C. Palazuelos, D. Pérez-García, V. Scholz, R. Werner. Connes' embedding problem and Tsirelson's problem. *Journal of Mathematical Physics*, 52(1):012102, 2011.

[15] T. Fritz. Tsirelson's problem and Kirchberg's conjecture. *Reviews in Mathematical Physics*, 24(05):1250012, 2012.

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

[17] S.-J. Kim, V. Paulsen, C. Schafhauser. A synchronous game for binary constraint systems. *Journal of Mathematical Physics*, 59(3):032201, 2018.

[18] V. Paulsen, S. Severini, D. Stahlke, I. Todorov, A. Winter. Estimating quantum chromatic numbers. *Journal of Functional Analysis*, 270(6):2188–2222, 2016.

[19] C. Papadimitriou, M. Yannakakis. A note on succinct representations of graphs. *Information and Control*, 71(3):181–185, 1986.

[20] S. Cook. The complexity of theorem-proving procedures. In *Proc. 3rd STOC*, 1971.

[21] V. Shoup. New algorithms for finding irreducible polynomials over finite fields. *Mathematics of Computation*, 54(189):435–447, 1990.

[22] H. W. Lenstra, Jr. Finding isomorphisms between finite fields. *Mathematics of Computation*, 56(193):329–347, 1991.

[23] J. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *Journal of the ACM*, 27(4):701–717, 1980.

[24] R. Zippel. Probabilistic algorithms for sparse polynomials. In *Symbolic and Algebraic Computation*, pages 216–226, 1979.

[25] M. Navascués, S. Pironio, A. Acín. A convergent hierarchy of semidefinite programs characterizing the set of quantum correlations. *New Journal of Physics*, 10(7):073013, 2008.

[26] B. Durand, A. Romashchenko, A. Shen. Fixed-point tile sets and their applications. *Journal of Computer and System Sciences*, 78(3):731–764, 2012.

[27] R. Raz. A parallel repetition theorem. *SIAM Journal on Computing*, 27(3):763–803, 1998.

[28] T. Holenstein. Parallel repetition: simplification and the no-signaling case. *Theory of Computing*, 5(1):141–172, 2009.

[29] R. Jain, S. Kundu. A direct product theorem for one-way quantum communication. In *Proc. 36th CCC*, 2021. arXiv:2008.08963.

[30] OpenAI. Ten advances in mathematics and theoretical computer science. Chapter 6: Exponential parallel repetition for all two-player entangled games. 2026. https://cdn.openai.com/pdf/ten-proofs-oai.pdf; Lean formalization at https://github.com/openai/ten-proofs.

[31] T. Vidick. Uniform direct parallel repetition for two-player commuting-operator strategies. Manuscript with Lean formalization, 2026. https://github.com/vidick/commuting-repetition.

[32] A. Uhlmann. The "transition probability" in the state space of a $*$-algebra. *Reports on Mathematical Physics*, 9(2):273–279, 1976.

[33] C. A. Fuchs, J. van de Graaf. Cryptographic distinguishability measures for quantum-mechanical states. *IEEE Transactions on Information Theory*, 45(4):1216–1227, 1999.

[34] E. H. Lieb, M. B. Ruskai. Proof of the strong subadditivity of quantum-mechanical entropy. *Journal of Mathematical Physics*, 14(12):1938–1941, 1973.

[35] M. M. Wilde. *Quantum Information Theory*. Cambridge University Press, 2013. arXiv:1106.1445.