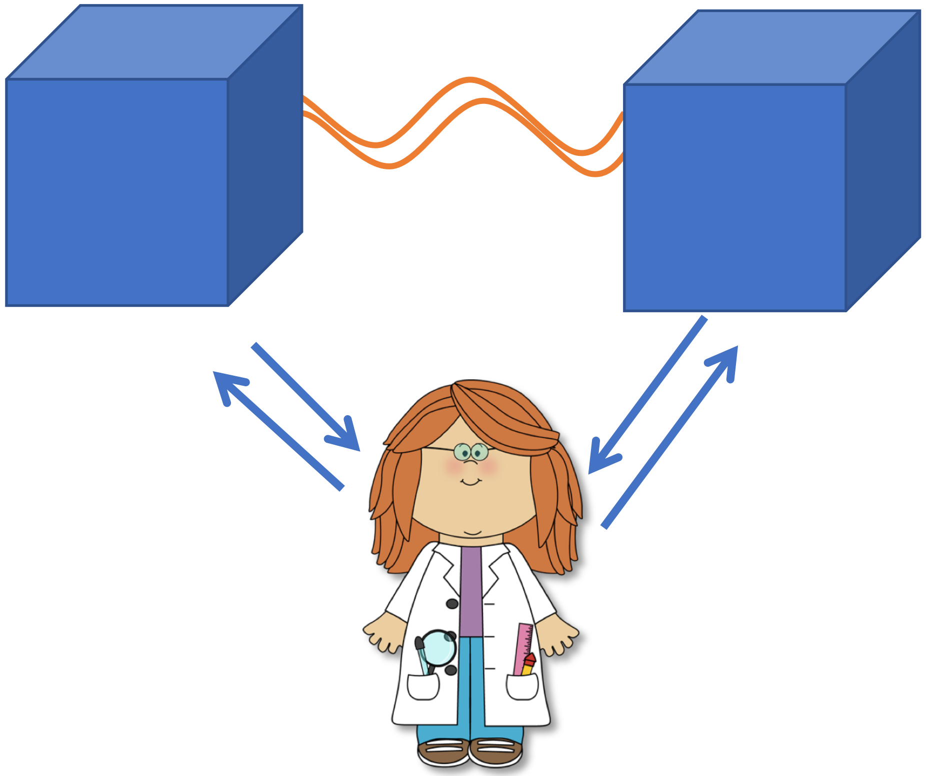


Multiprover Protocols

Part II

A lens on complexity, cryptography, and beyond

Henry Yuen

University of Toronto

Testing for many qubits, redux

- Parameters of N-fold Magic Square
  - Certifies $2N$ EPR pairs, and X/Z Pauli measurements on those EPR pairs
  - Questions/answer length: $O(N)$ bits
  - Robustness: $1-\epsilon$ winning probability $\Rightarrow O(N^2\sqrt{\epsilon})$-close to textbook strategy

State-of-the-art

- Quantum low-degree test (Natarajan-Vidick 2018):
  - Certifies N EPR pairs, and X/Z Pauli measurements on those EPR pairs
  - Questions/answer length: $polylog(N)$ bits

    Exponentially shorter messages
  - Robustness: $1 - \epsilon$ winning probability $\Rightarrow O(\epsilon^{\alpha})$-close to textbook strategy

    Robustness independent of N!

Classical low-degree test

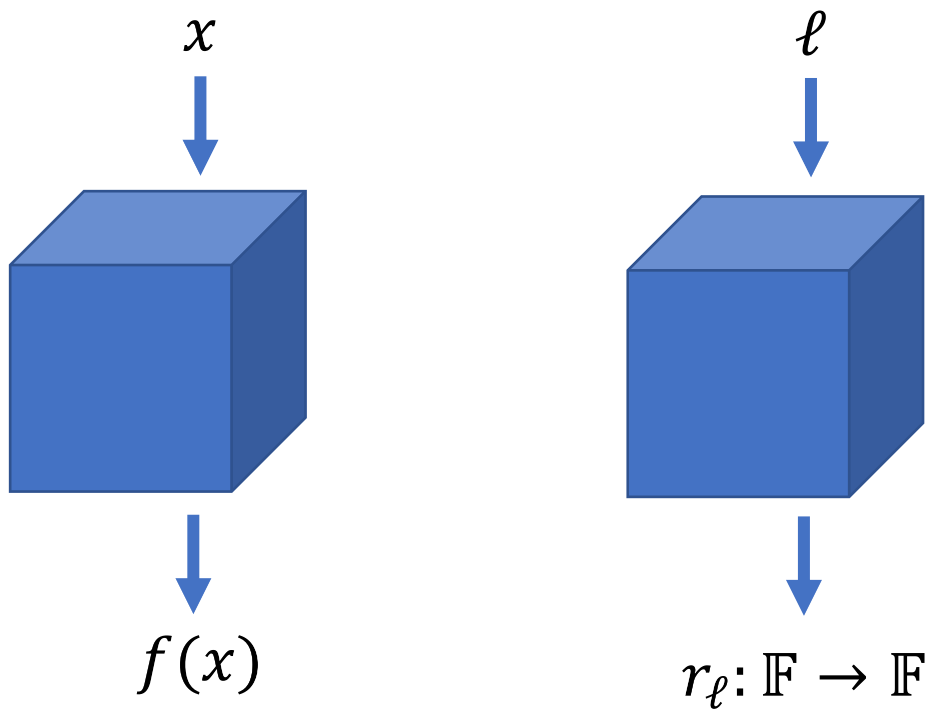
- A protocol with “classical rigidity” properties
- Crucial in *probabilistically checkable proofs*
- Assume provers are deterministic
- **Goal**: test whether their responses are consistent with a $\mathbb{F}$-valued degree-d polynomial over $\mathbb{F}^m$

Classical low-degree test

- Verifier picks random
  - Point $x \in \mathbb{F}^m$
  - Line $\ell = \{u + vt : t \in \mathbb{F}\}$ containing x

- Prover A responds with $f(x) \in \mathbb{F}$
- Prover B responds with univariate degree-d polynomial $r_\ell : \mathbb{F} \to \mathbb{F}$

- Provers win if $f(x) = r_\ell(x)$

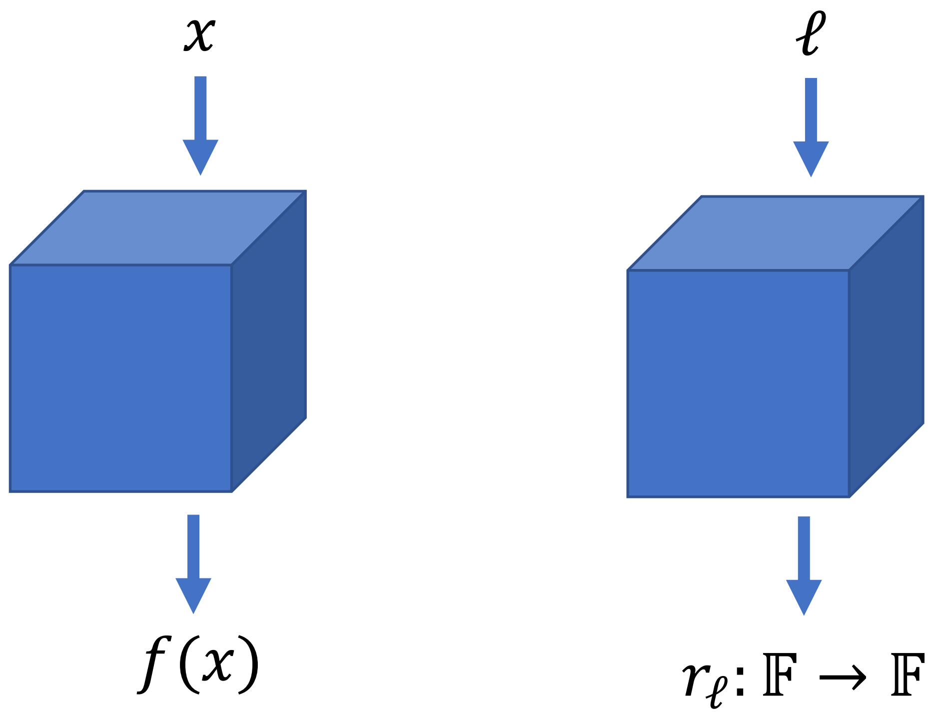


If f is degree-d: r_ℓ should be $f|_\ell$

Classical low-degree test

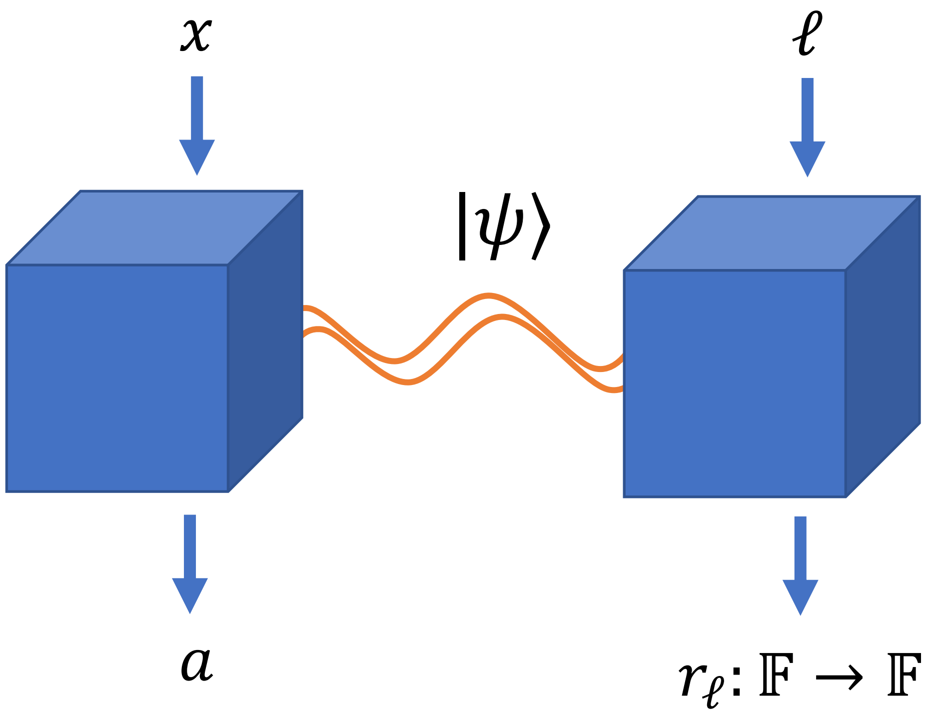
Theorem (AS, ALMSS, RS, …): If provers win w.p. $\geq 1 - \epsilon$, then f is $O(\epsilon)$-close to degree-d.

- Extremely efficient test for structure!
- Description of degree-d function: $\binom{m+d}{d}$
- Questions/answers in CLD: $O(m \log|\mathbb{F}|)$



Classical low-degree test, entangled provers

Theorem (NV18): If provers win w.p. $\geq 1 - \epsilon$, then provers' measurements are $O(\epsilon^{\alpha})$-consistent with degree-d polynomial.



"Classical rigidity" phenomenon persists even in presence of entangled provers!

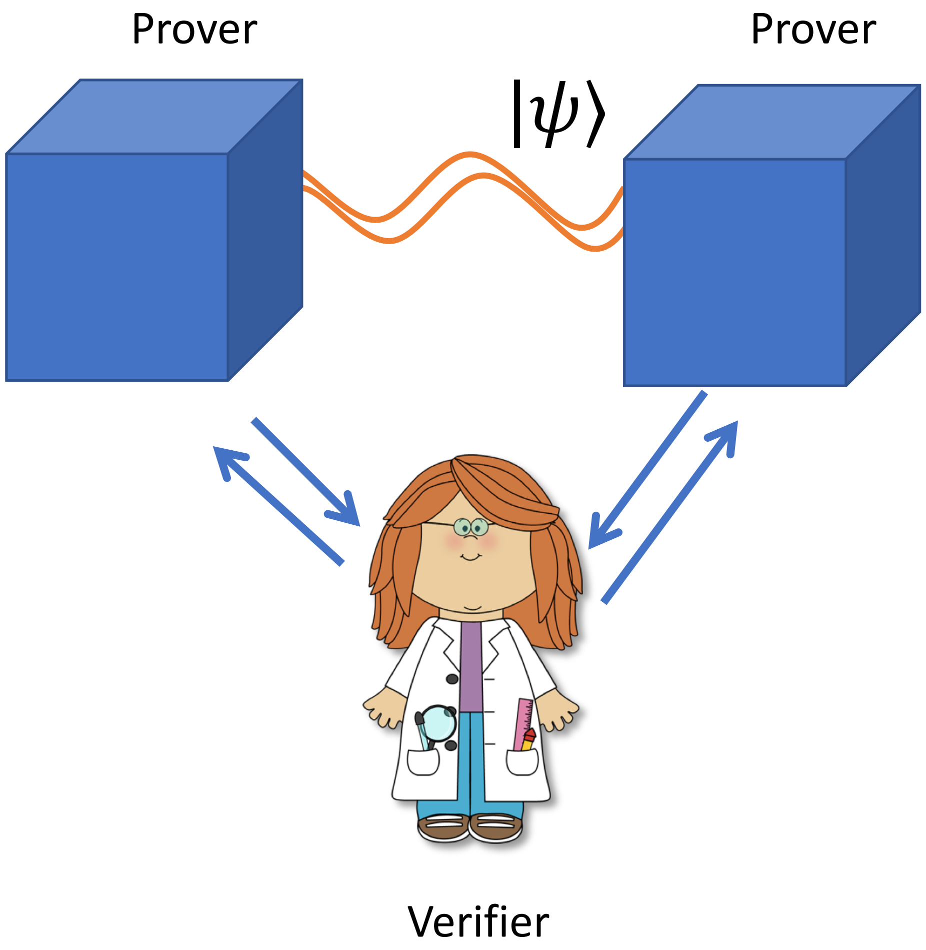
From classical to quantum low-degree testing

Very roughly:

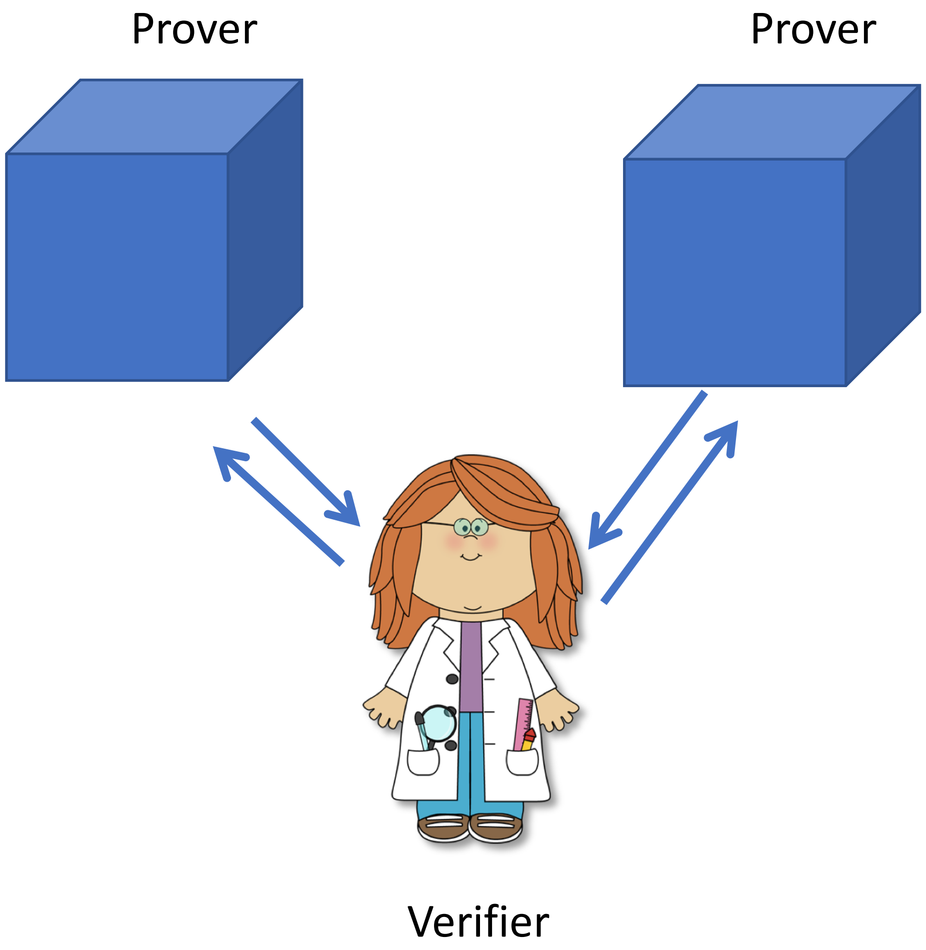
- Perform classical low-degree testing in X and Z bases separately
- Relate the two bases using Magic Square game.

Complexity of interactive proofs

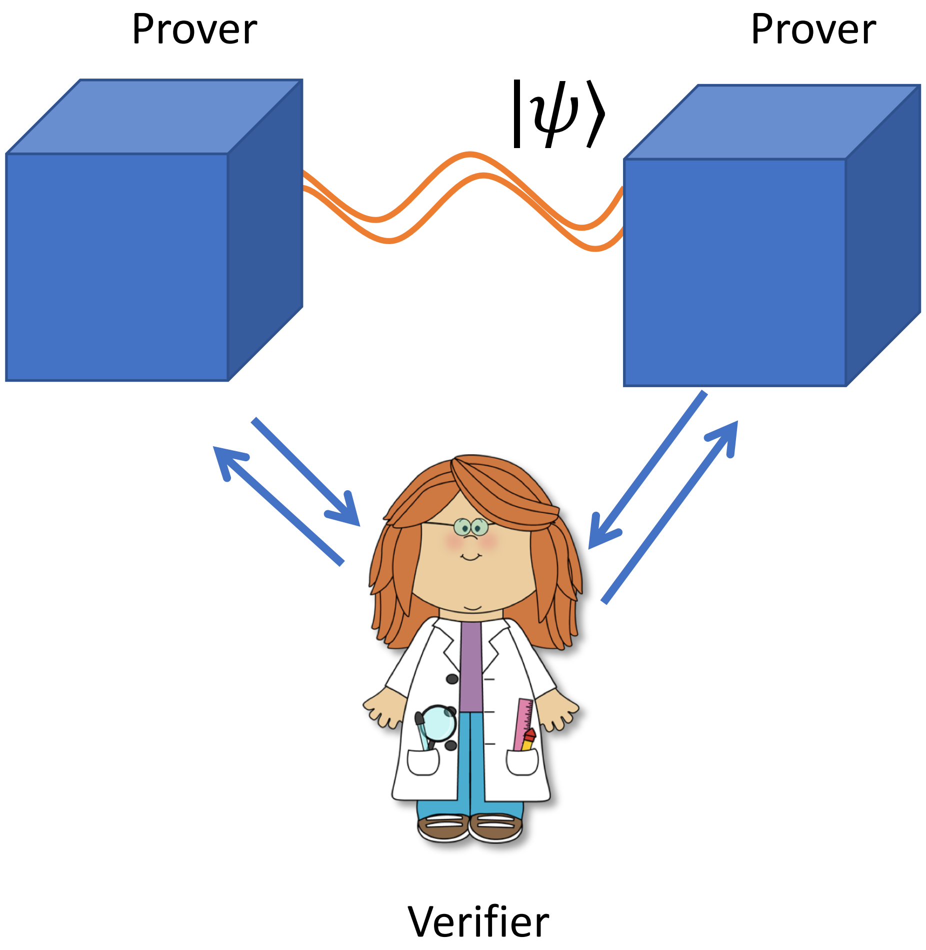
with entangled provers



- Recall: provers want to convince verifier of statement *X, e.g.,*
  - *"N is product of two primes"*
  - *"quantum circuit C accepts whp"*
  - *"graphs G, H are isomorphic"*
- What statements can be verified using multiprover protocols with entangled provers?
- No assumptions on complexity of provers
- Protocol must be **complete** and **sound**



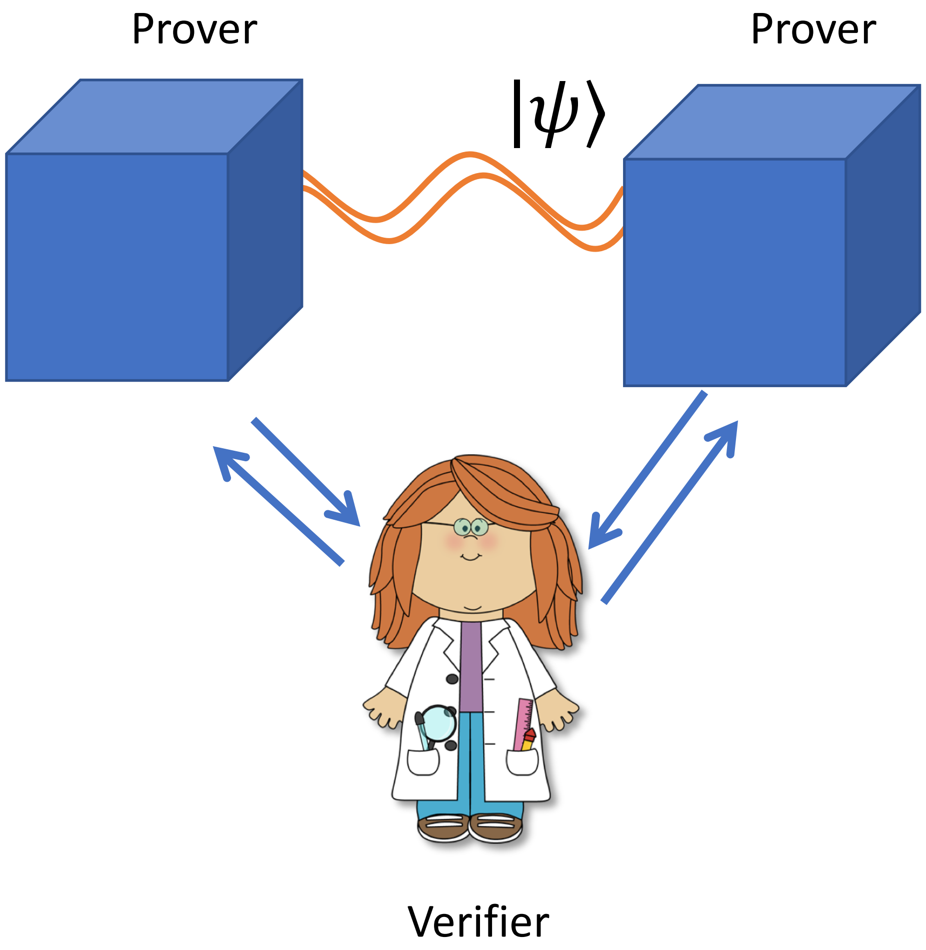
- **MIP** denotes complexity class of problems that can be verified with classical multiprover protocols
- Classical provers = deterministic
- Babai, Fortnow, Lund '91: **MIP = NEXP**
- Polynomial-time verifier can check statements like *"Turing machine M outputs 42 after exponentially many steps".*
- Crucial component: classical low-degree test



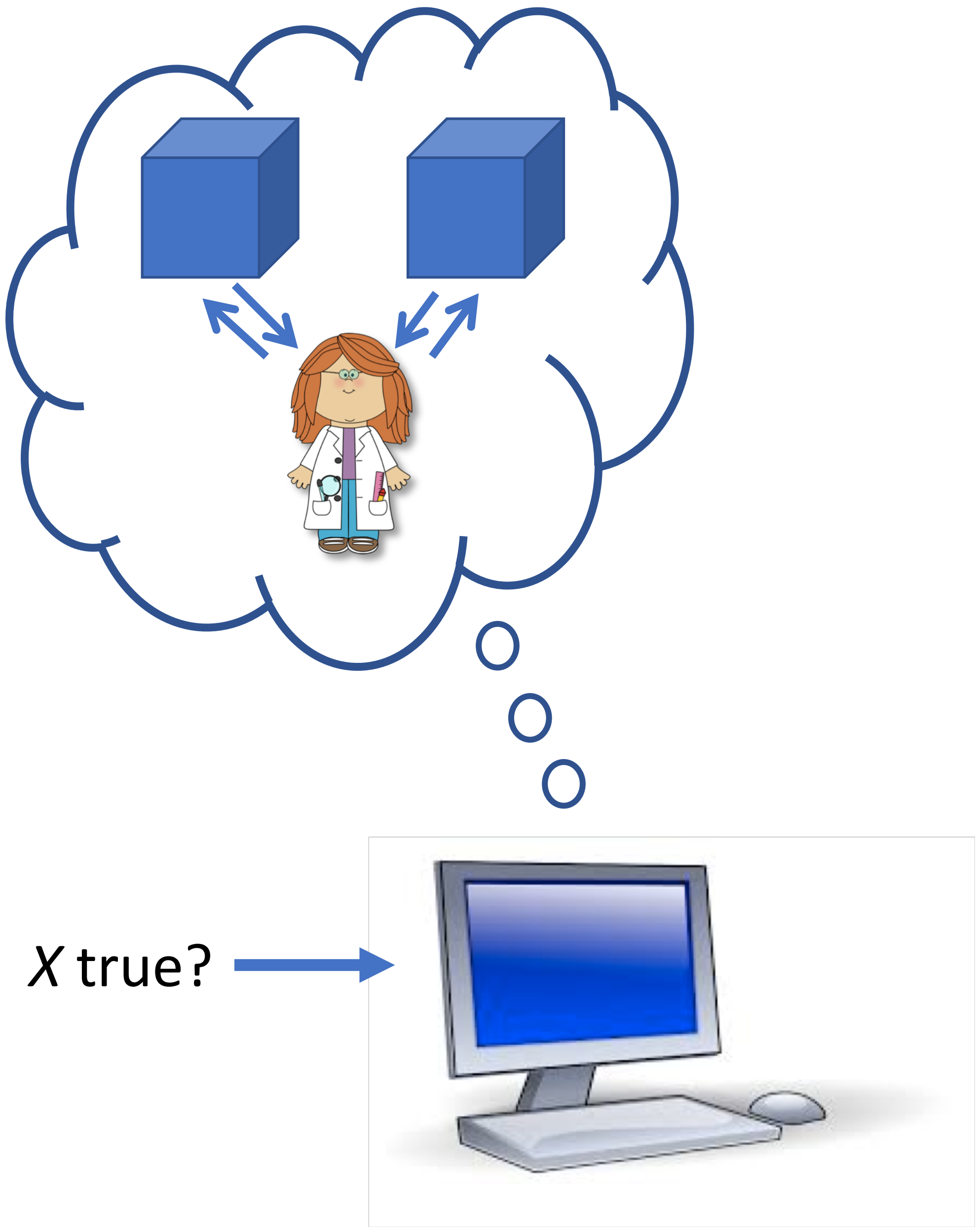
- **MIP*** denotes complexity class of problems that can be verified with ***entangled-prover*** protocols

- **MIP** vs **MIP***? Consider classical **MIP** protocol to verify statement X.
  - If X true, then quantum provers can also prove X to verifier.
  - If X false, then verifier rejects all classical provers, but *entangled provers* may be able to cause verifier to accept!

- **Soundness of classical MIP protocols may no longer hold against entangled provers!**



- [Ito-Vidick 2012] **NEXP ⊆ MIP***
- Showed classical protocol of Babai, Fortnow, Lund still safe against entangled provers.
  - Classical low-degree test still guarantees structure even with entangled provers.
- Entanglement cannot ***reduce*** complexity of multiprover protocols.
- …can entanglement ***expand*** their complexity?



- Algorithmic upper bounds on **MIP***?

- Compare: **MIP ⊆ DOUBLY-EXP**
- Proof: Suppose *X* can be verified by classical **MIP** protocol *P*. Then implies doubly exponential time procedure to compute whether *X* is true:

  Enumerate over all possible deterministic provers for *P*, and calculate acceptance probability of verifier.

- Why doesn't this work for **MIP***?
- Space of provers is infinite; no upper bound on amount of entanglement needed.

- Best upper bound known: **MIP*** $\subseteq$ **RE**

- [Ji-Natarajan-Vidick-Wright-Y.] **MIP* = RE**

- Complexity-theoretic implications

  - Classical, polynomial-time verifier with entangled provers can verify *X* = "Turing machine *M* eventually halts"

  - There is no computable upper bound on amount of entanglement needed in general **MIP*** protocols.

  - No computable upper bound on **MIP***

Using entanglement in MIP*

- [Natarajan-Wright] **NEEXP ⊆ MIP*** shows how verifier can use entangled provers to its advantage.

- **Key idea**: using rigidity, force provers to simulate exponentially large verification protocol.

Using entanglement in **MIP***

- Goal: verify *X* = "Turing machine *M* accepts after 2^N steps"

- *X* is **NEXP** statement, so there exists **MIP*** protocol with
  - 1 round
  - Verifier runs in $poly(N)$ time.
  - Based on classical low-degree test.

Using entanglement in MIP*

- Goal: verify *X* = “Turing machine *M* accepts after 2^{2^N} steps”

- There exists protocol P_{Big} where:
  - 1 round
  - Verifier runs in 2^N time.
  - Based on classical low-degree test.

- Want a protocol P_{Small} that verifies *X* using $poly(N)$-time verifier.

Question and answer reduction

P_{Big} — Questions: $\exp(N)$-bits; Answers: $\exp(N)$-bits

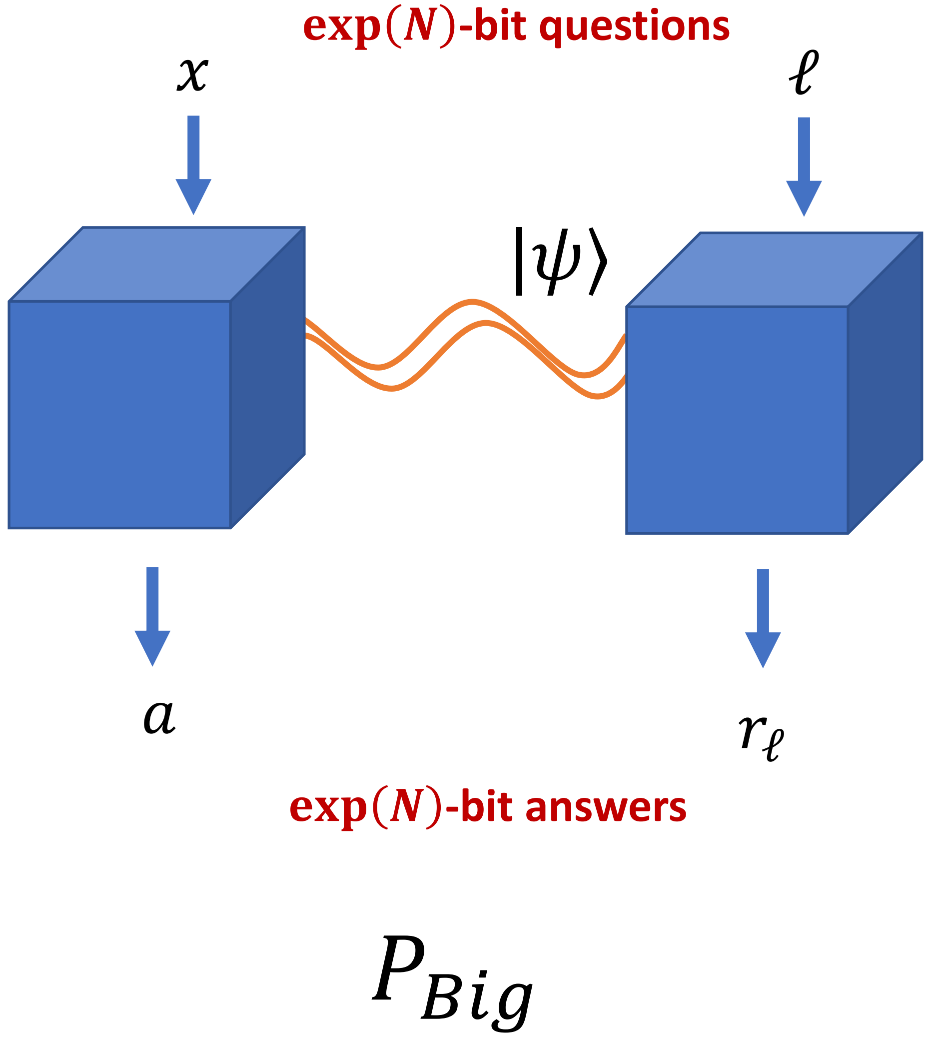
$\Downarrow$

P_{int} — Questions: $\mathrm{poly}(N)$-bits; Answers: $\exp(N)$-bits

$\Downarrow$

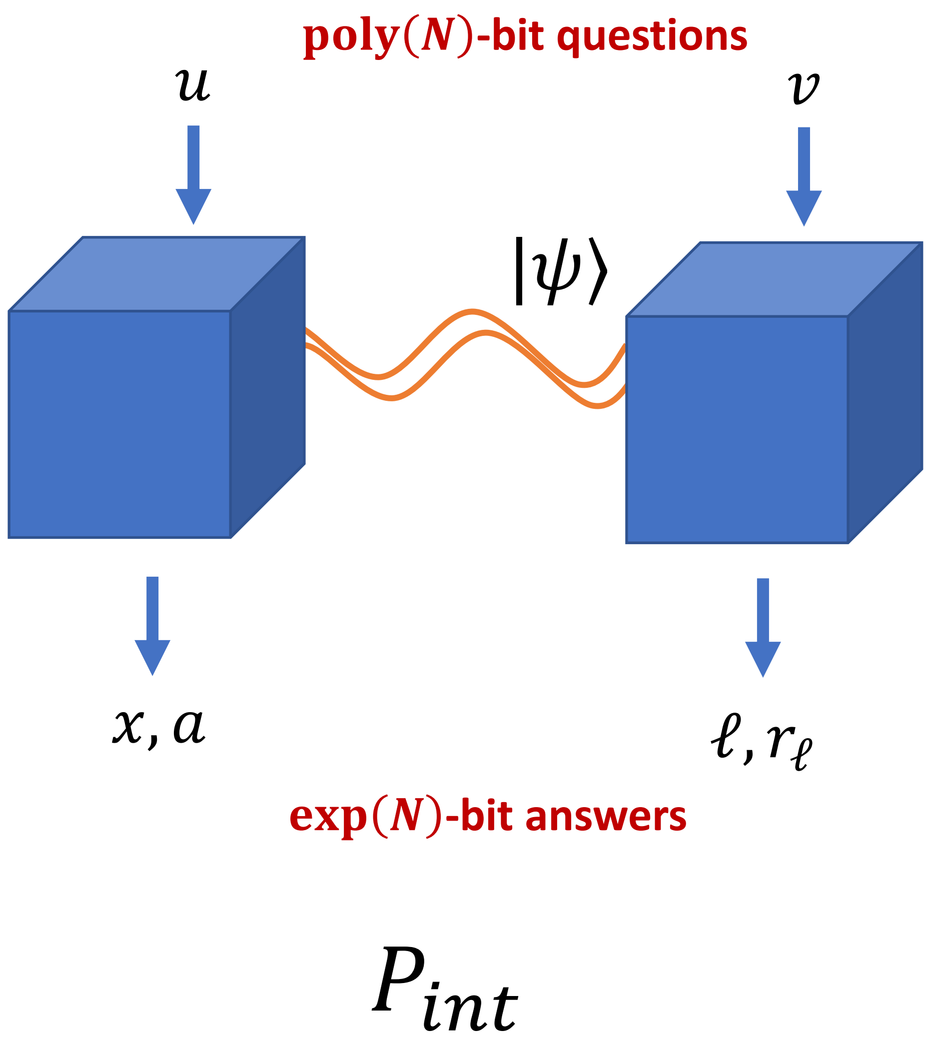
P_{Small} — Questions: $\mathrm{poly}(N)$-bits; Answers: $poly(N)$-bits

Reducing question size



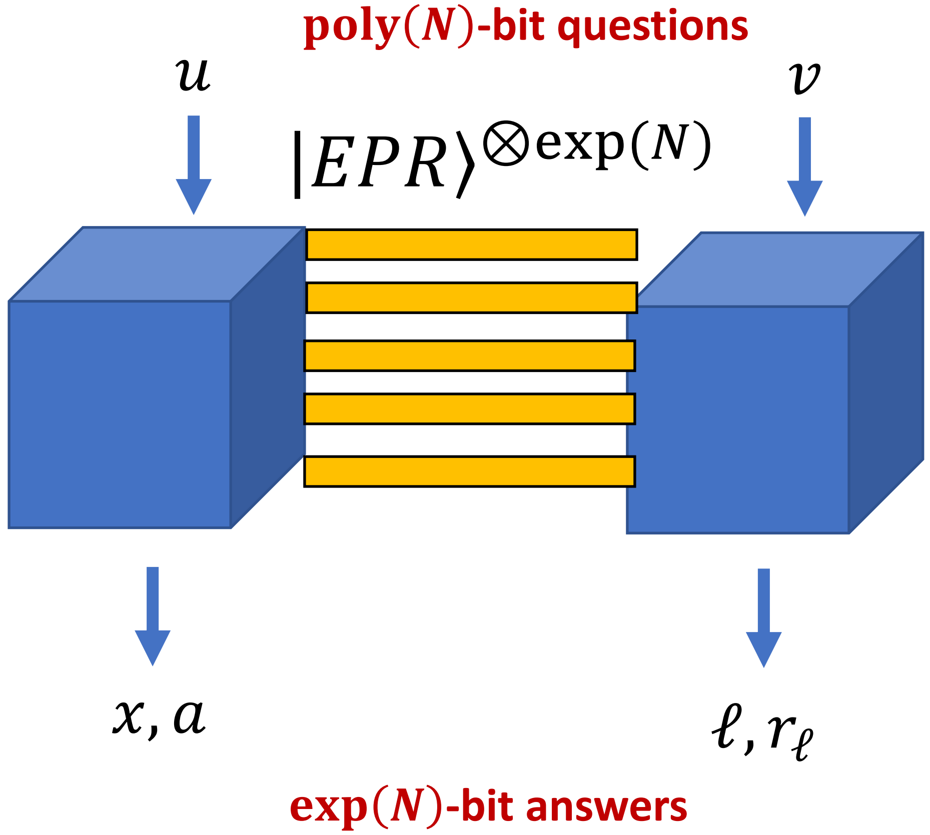
- In P_{Big}, point/line questions are exponential length
  - E.g., $x \in \mathbb{F}^m$ where $m = \exp(N)$

Reducing question size



- In P_{Big}, point/line questions are exponential length
  - E.g., $x \in \mathbb{F}^m$ where $m = \exp(N)$
- Intermediate protocol P_{int}: forces provers to sample questions (x, ℓ) themselves, and then generate answers (a, r_ℓ) to their own questions ("***introspection***")
- Verifier in P_{int} uses questions (u, v) of length $poly(N)$.

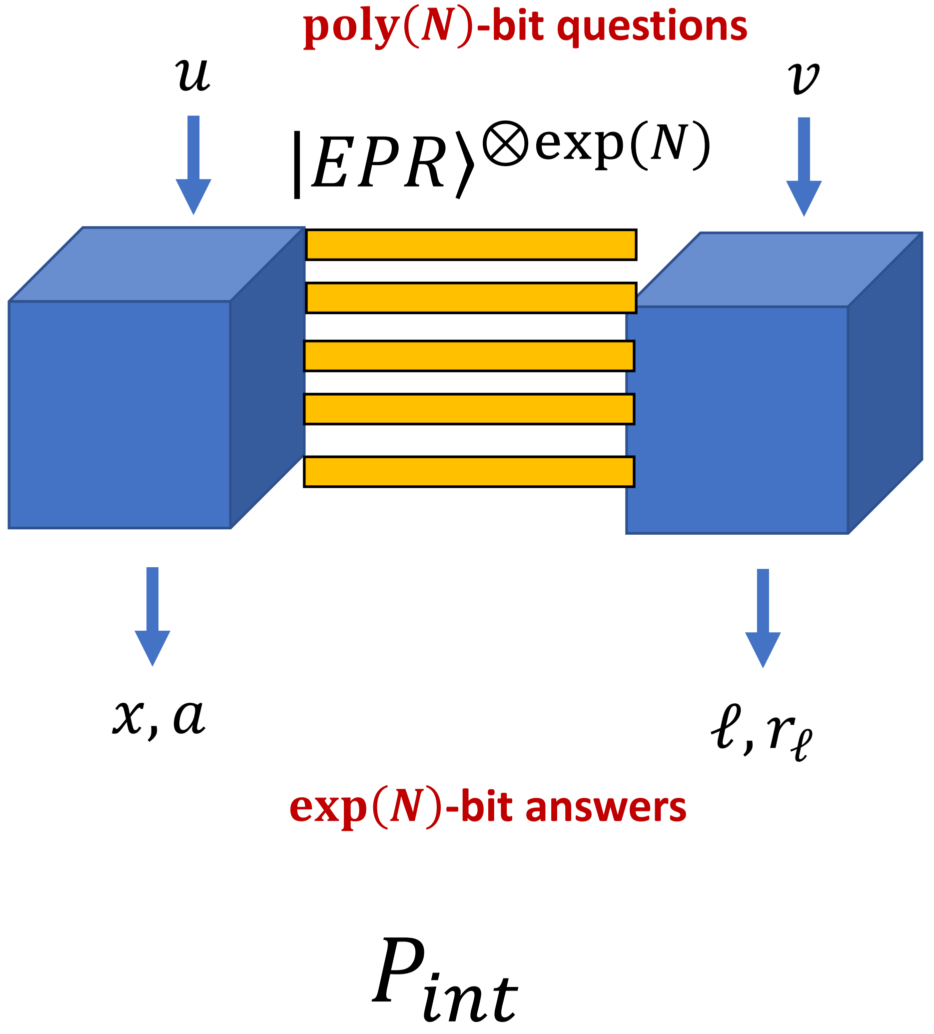
Reducing question size



P_{int}

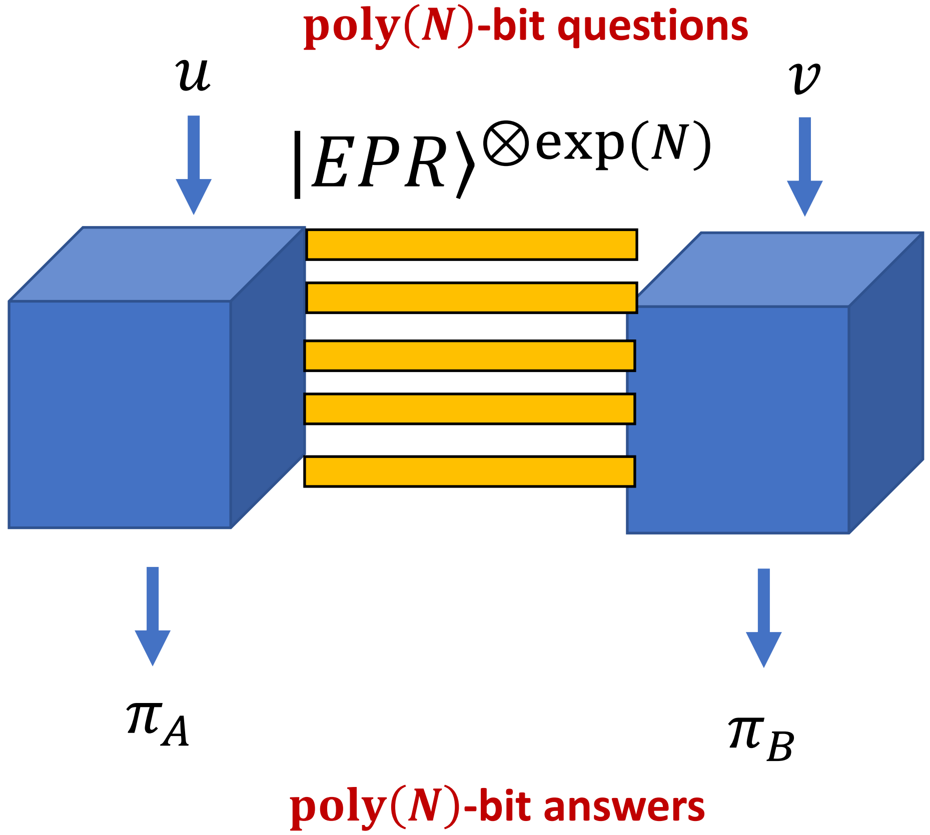
- P_{int} protocol
  - With prob. ½, run Quantum Low-Degree Test to certify $\exp(N)$ EPR pairs
  - With prob. ½, run Introspection protocol to certify:
    - Provers sample point/line distribution (x, ℓ) as in P_{Big}.
    - Provers' answers (a, r_ℓ) to introspected questions (x, ℓ) satisfy verifier in P_{Big}

Reducing question size



- P_{int} protocol
  - exp(N) EPR pairs are used as a source of randomness to generate (x, ℓ)
  - **Challenge**: needs to certify that prover A only samples point x, prover B only samples line ℓ, and there is no leakage of information!
  - Solution crucially relies on special structure of point-line distribution!

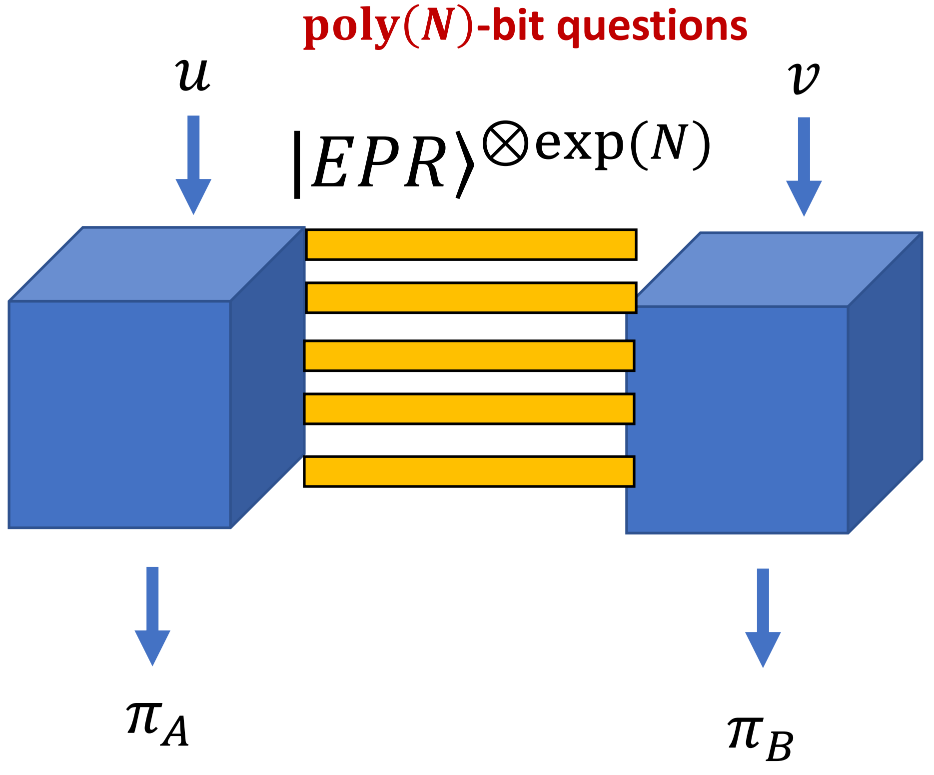
Reducing answer size



P_{Small}

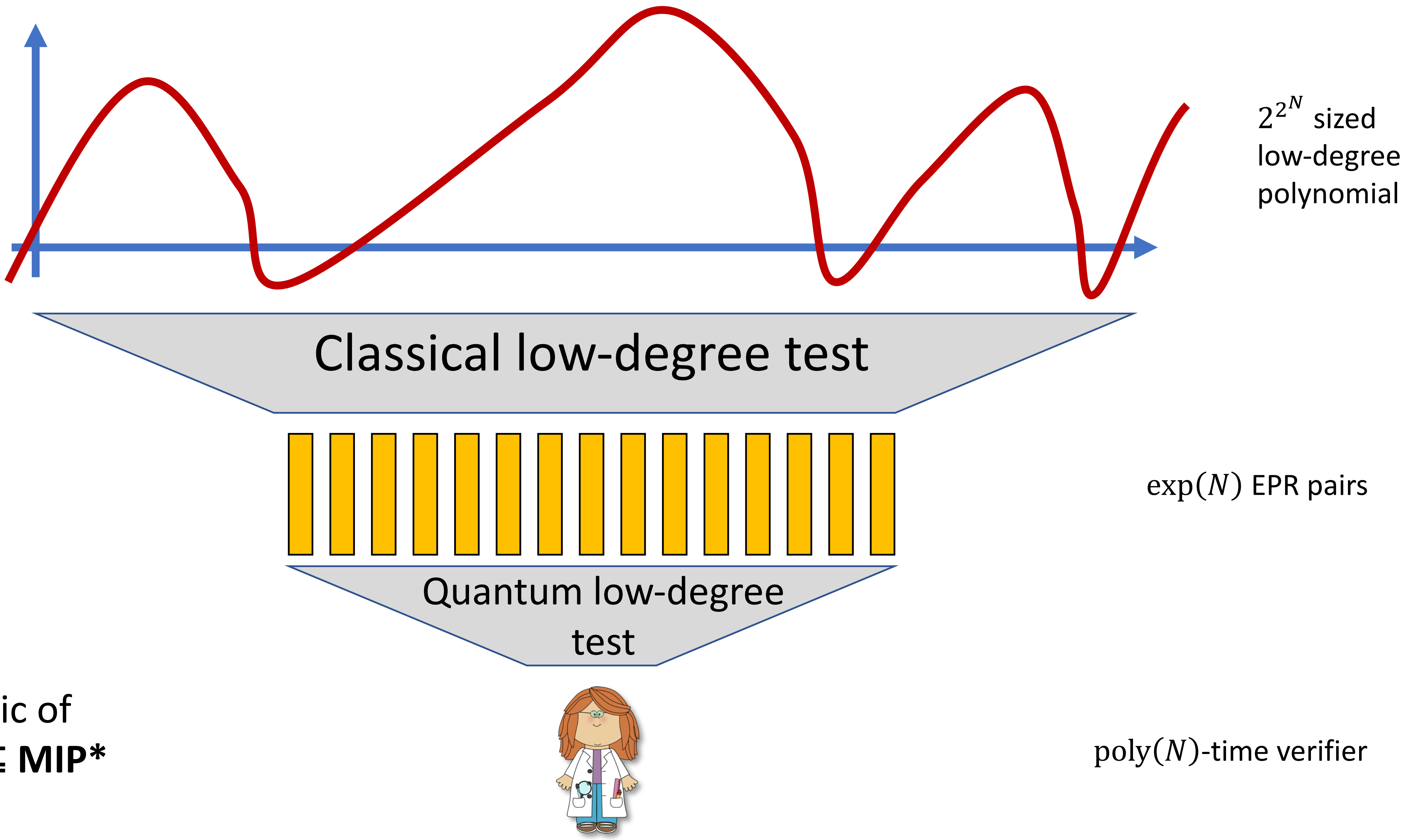
- P_{Small} protocol
  - Provers give succinct proofs (π_A, π_B) that they would've given accepting answers in P_{int}
  - Based on probabilistically checkable proofs (PCPs)
  - Proofs are $poly(N)$ bits long

Final protocol

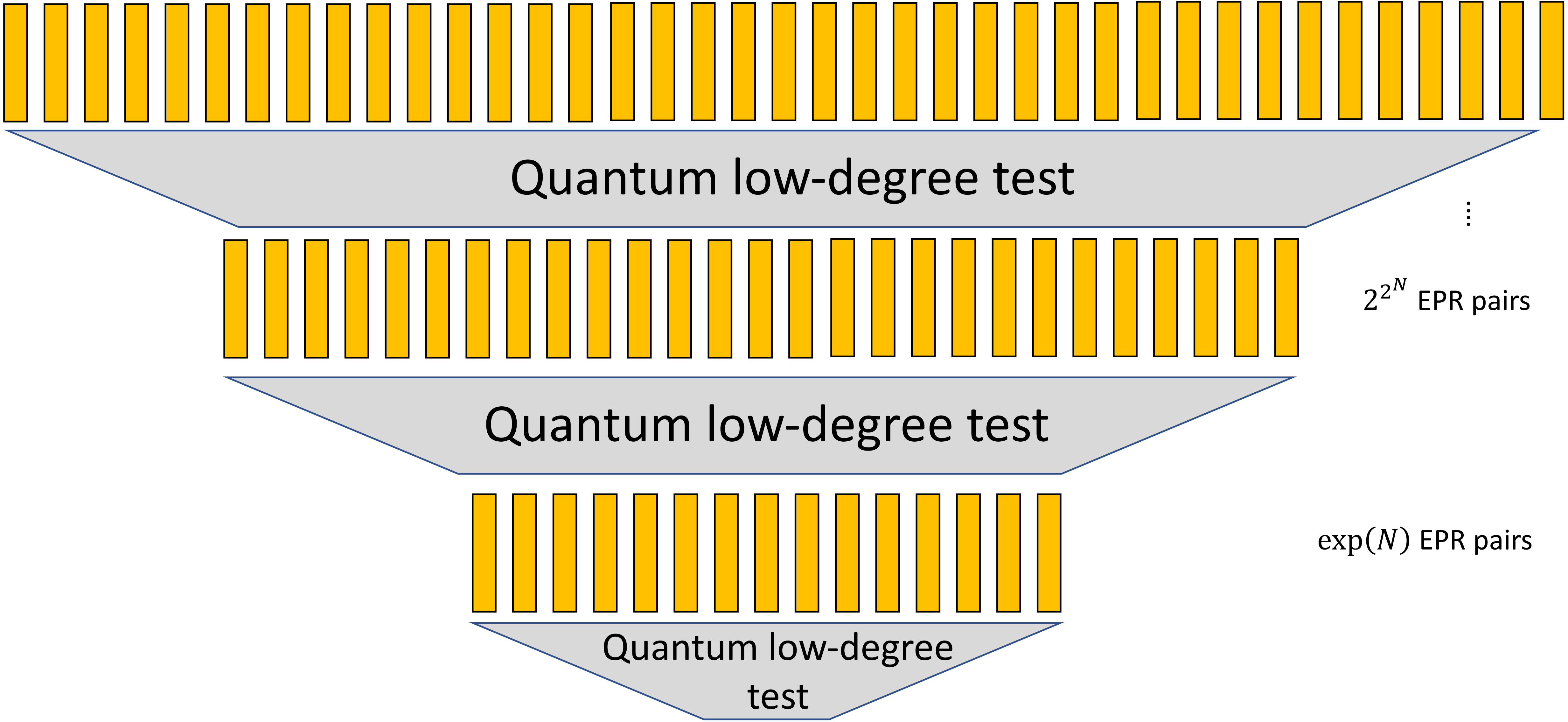


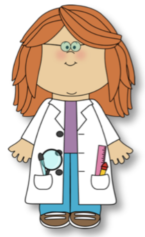
P_{Small}

- $poly(N)$-time verifier can verify
  X = "Turing machine M accepts after 2^{2^N} steps"
- Uses Quantum Low-Degree test to certify $\exp(N)$ EPR pairs using $poly(N)$ question length
- Uses Introspection to certify sampling from exponentially large *classical* low-degree test questions from EPR pairs
- Use PCPs to reduce answer size to $poly(N)$.



Schematic of
NEEXP ⊆ MIP*





$\mathrm{poly}(N)$-time verifier

Why not iterate…?

Consequences

- Recursively iterating the Introspection technique from Natarajan-Wright yields **MIP*** protocol for verifying the Halting Problem.
- No computable upper bound on **MIP***
- Resolves questions in three different areas:
  - Complexity of **MIP*** (Computer science)
  - Tsirelson’s Problem (Mathematical physics)
  - Connes’ Embedding Problem (Pure mathematics)

Unexpected connections

Connes' Embedding Problem (1974)

Do all separable type II_1 factors embed into an ultrapower of the hyperfinite II_1 factor?

Tsirelson's Problem (2006)

Are all quantum correlations in the commuting operator model approximable in finite dimensions?

Complexity of **MIP*** (2004)

*Is there an algorithmic upper bound on **MIP*?***

Thanks!

The multiprover lens

- Cryptography
  - Delegated quantum computation
  - Randomness expansion
  - Device indepen
    cryptography
  - Zero knowledge
- Complexity theo y
  - Complexity of MIP*
  - Hamiltonian complexity

- Foundations of quantum mechanics
  - Rigidity of quantum correlations
  - Finite vs infinite dimensional quantum

  s
  - sis
  - Representation theory
  - Algebra
  - Noncommutative optimization

Central tool:

Rigidity of quantum correlations