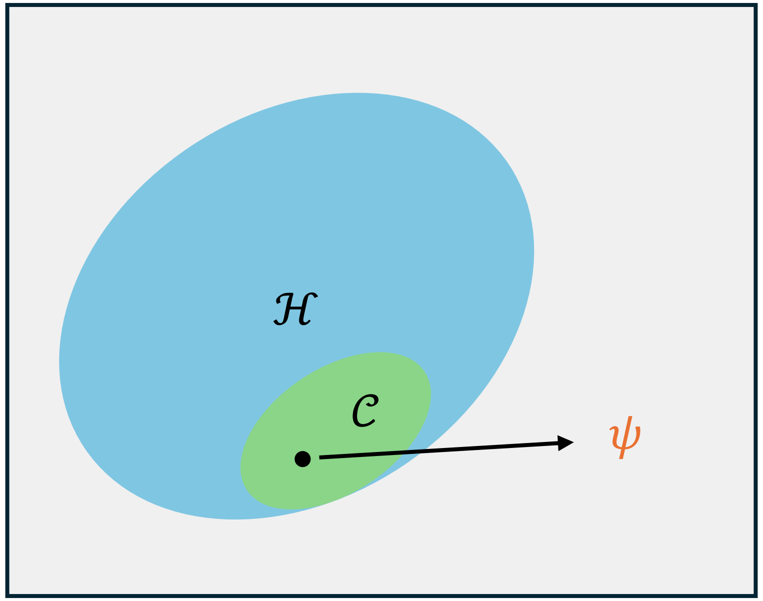
Nearly tight bounds for testing tree tensor network states

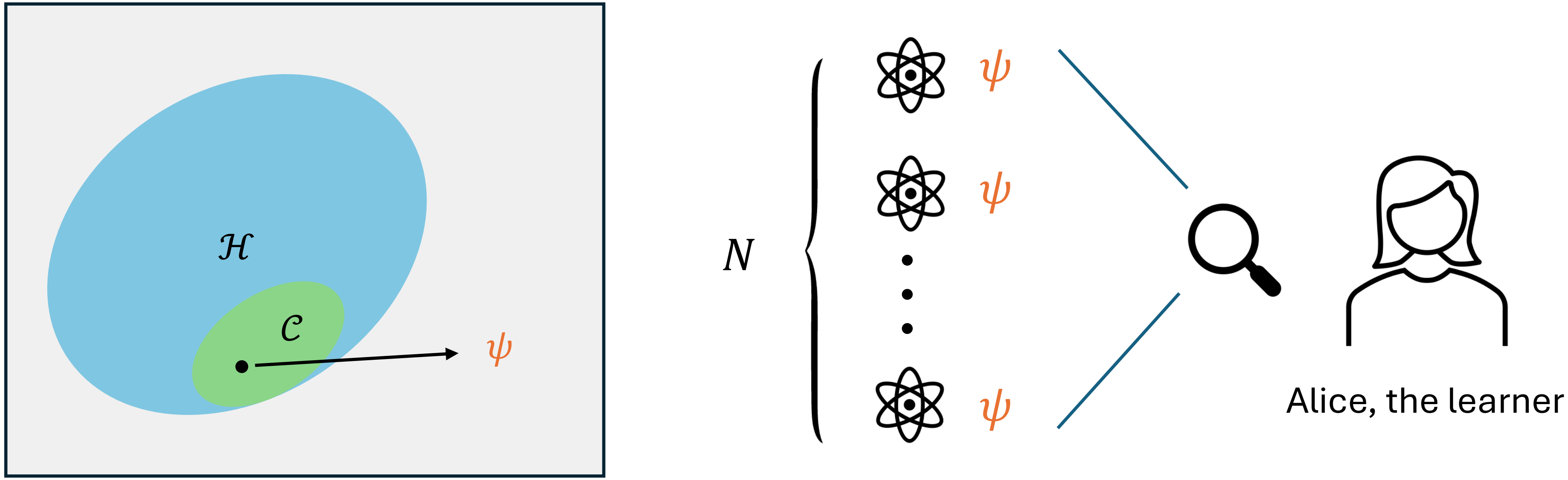
Benjamin Lovitz (Northeastern) and **Angus Lowe** (MIT)

arxiv:2410.21417

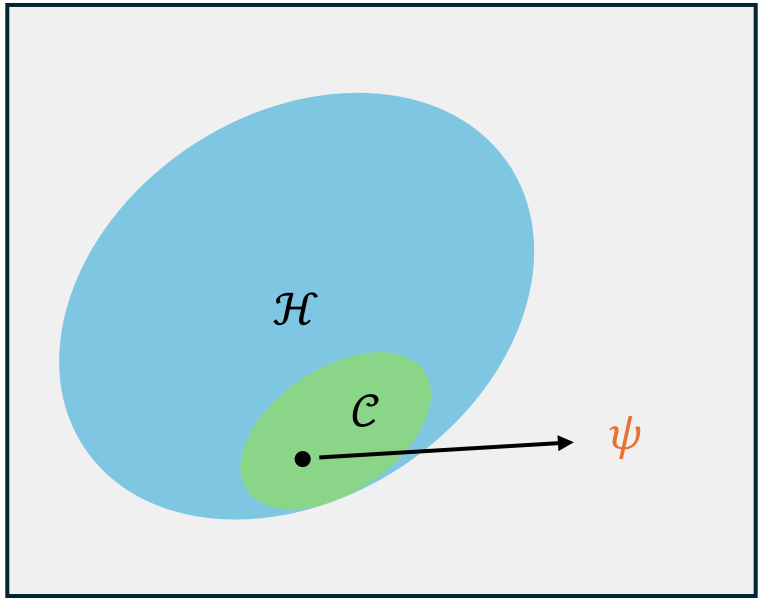
I. Background & Main Result
II. Lower Bound for a Specific Algorithm
III. There is No Better Algorithm
IV. Final Remarks

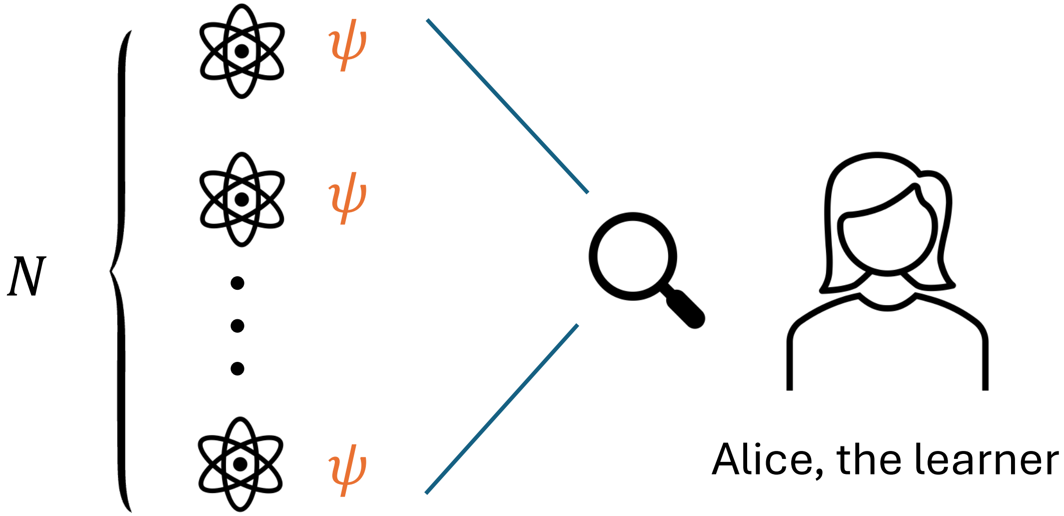


$\mathcal{H}$
$\mathcal{C}$
ψ

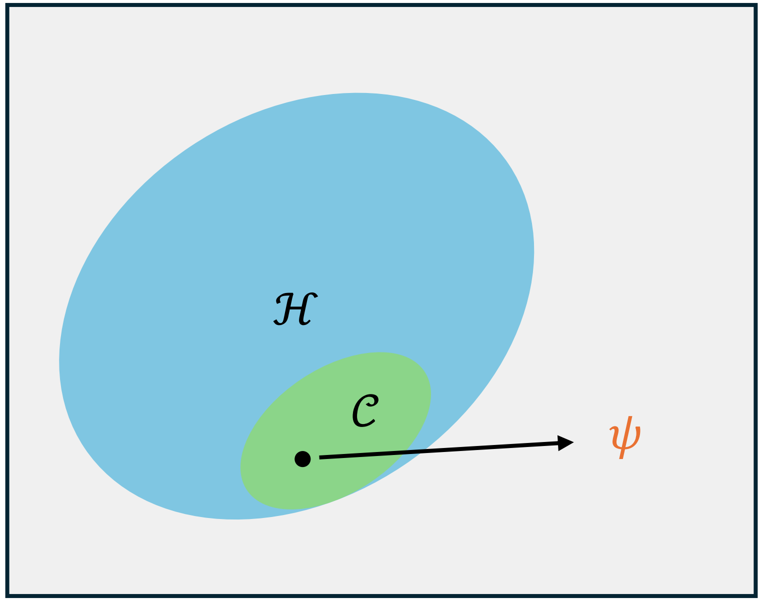


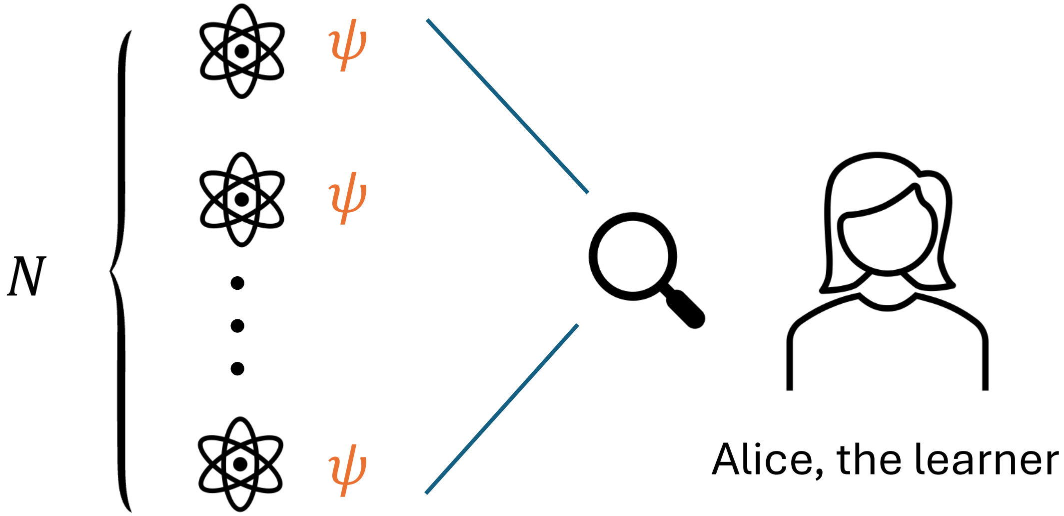
$\mathcal{H}$
$\mathcal{C}$
ψ
N
ψ
ψ
ψ
Alice, the learner





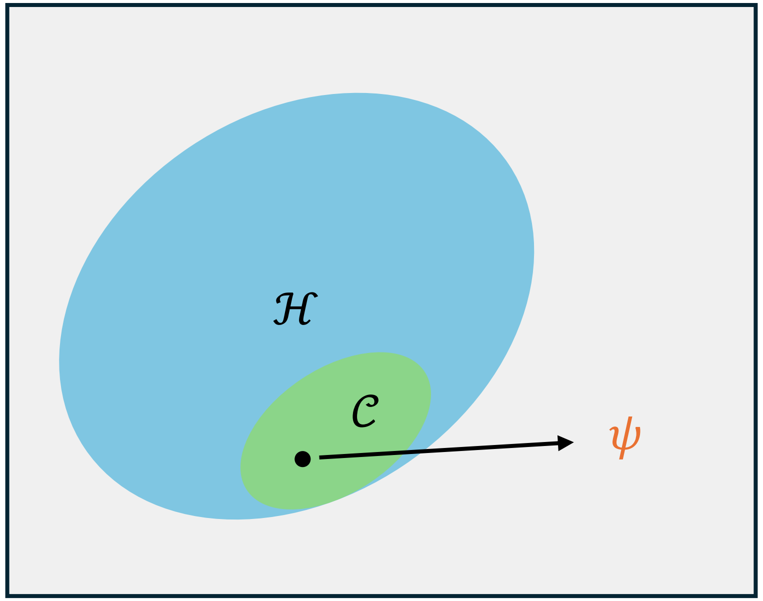
- What are the expected values of some observables? (Shadow tomography)
- How strong is the magnetic field that produces the state ψ? (Quantum sensing)
- What is ψ? (Tomography/learning)
- Is ψ even in $\mathcal{C}$? (Property testing)



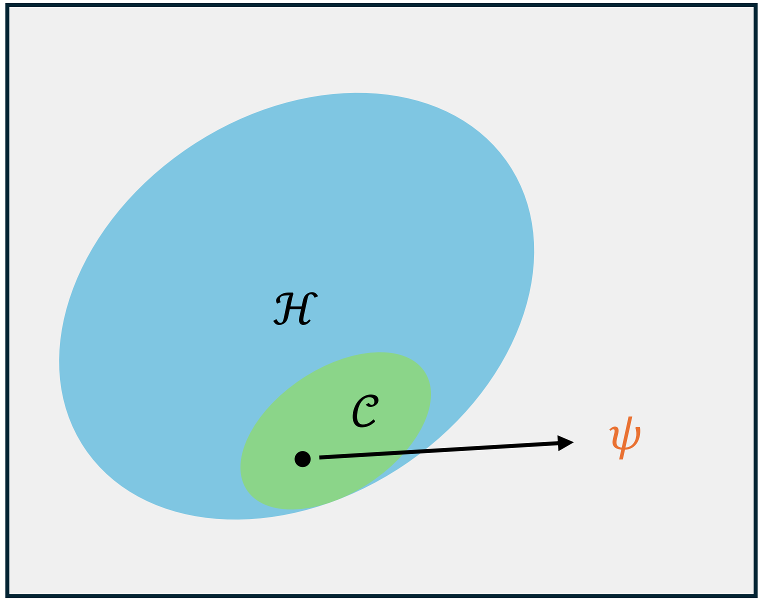


- What are the expected values of some observables? (Shadow tomography)
- How strong is the magnetic field that produces the state ψ? (Quantum sensing)
- What is ψ? (Tomography/learning)
- Is ψ even in $\mathcal{C}$? (Property testing)

Quantum Property Testing

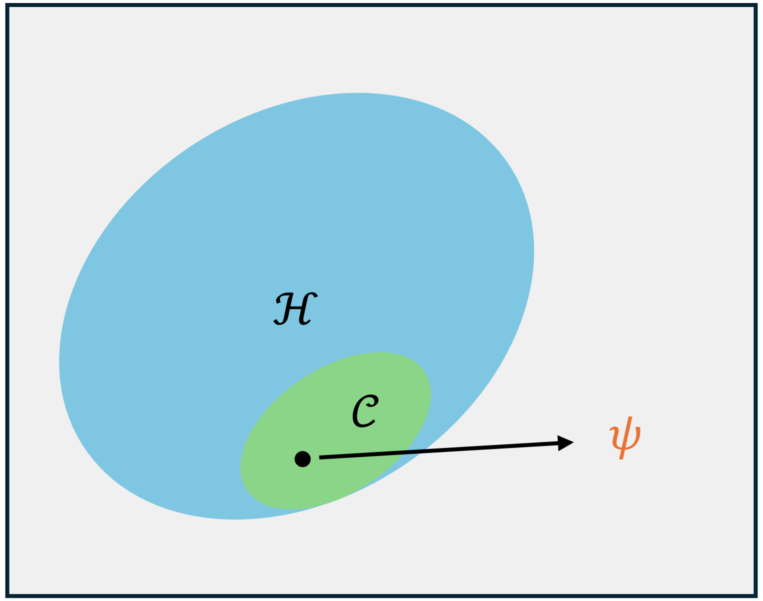


Quantum Property Testing

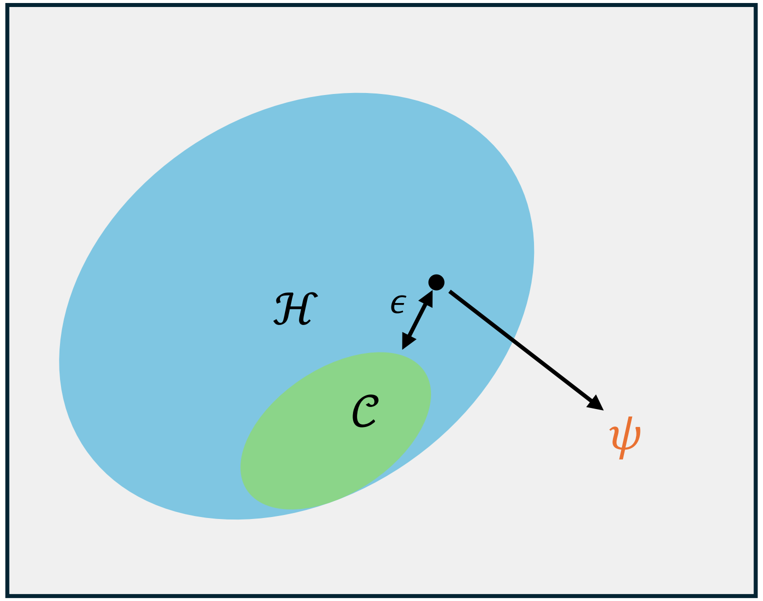


Completeness: $\mathrm{Tr}\left(M\psi^{\otimes N}\right) \geq a$

Quantum Property Testing

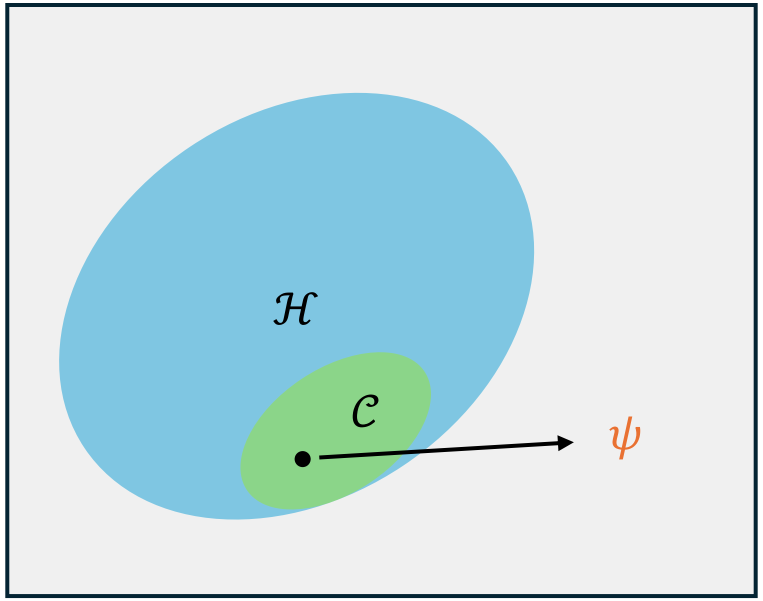


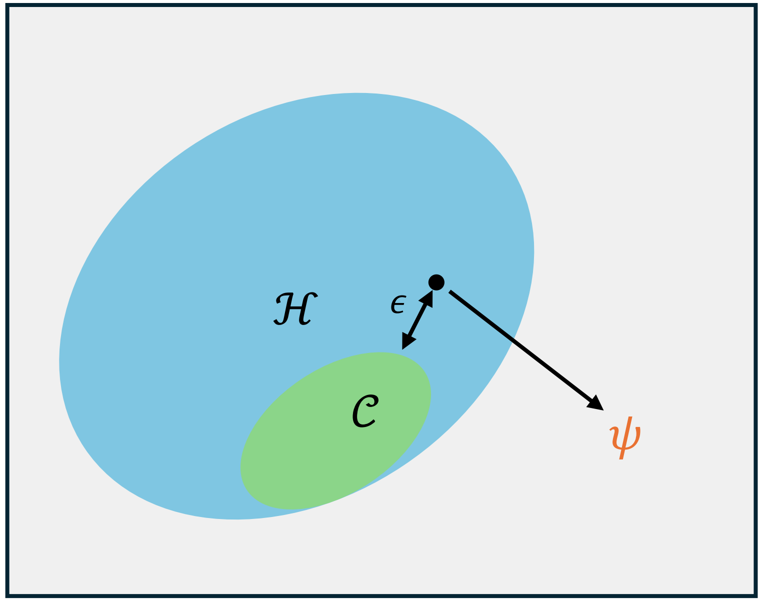
Completeness: $\mathrm{Tr}\left(M\psi^{\otimes N}\right) \geq a$



Soundness: $\mathrm{Tr}\left(M\psi^{\otimes N}\right) \leq b$

Quantum Property Testing



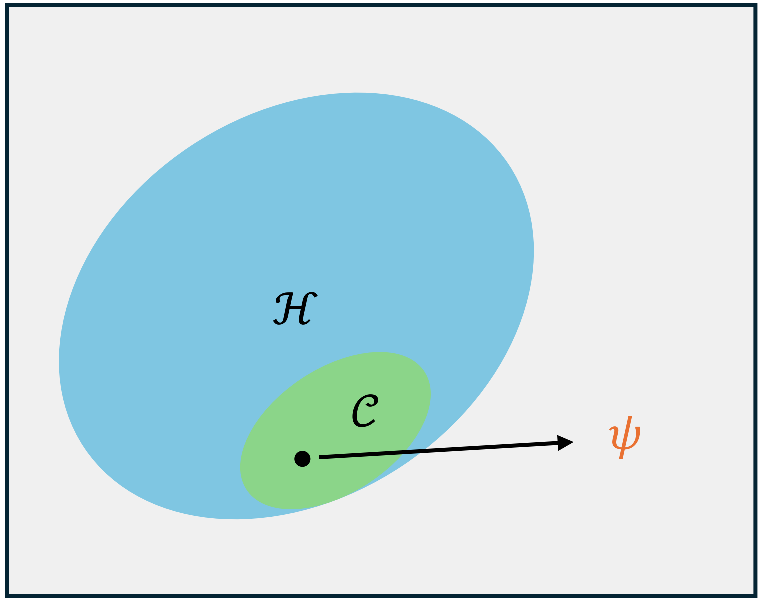


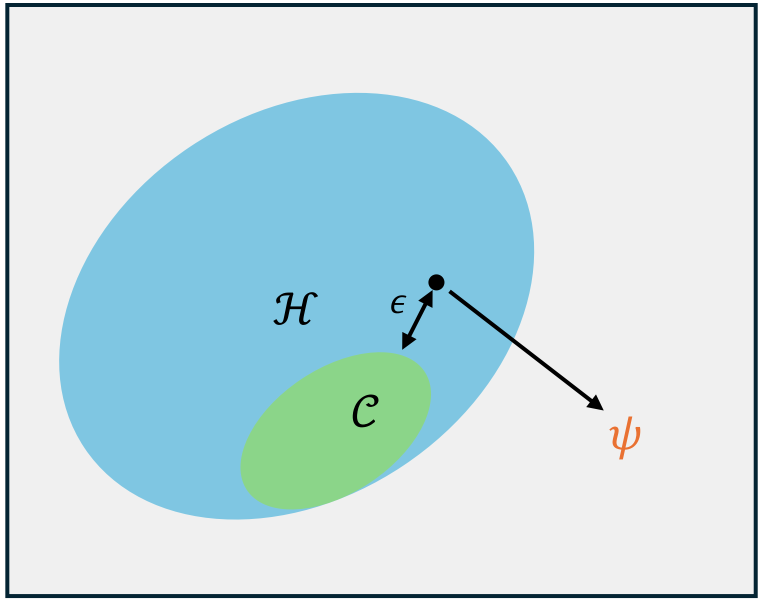
Completeness: $\mathrm{Tr}\left(M\psi^{\otimes N}\right) \geq a$

Soundness: $\mathrm{Tr}\left(M\psi^{\otimes N}\right) \leq b$

- "Successful" if $a - b = \Omega(1)$

Quantum Property Testing



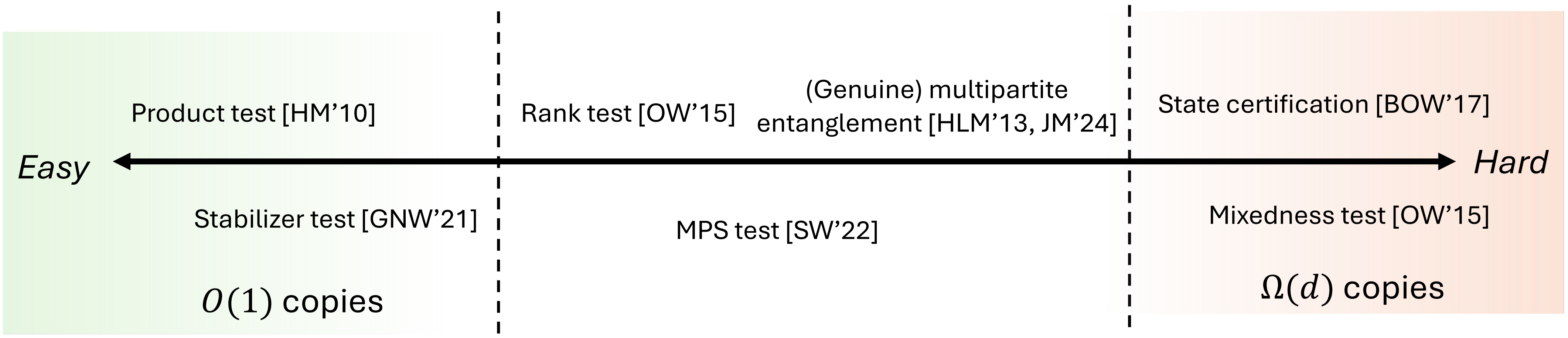


Completeness: $\mathrm{Tr}\left(M\psi^{\otimes N}\right) \geq a$

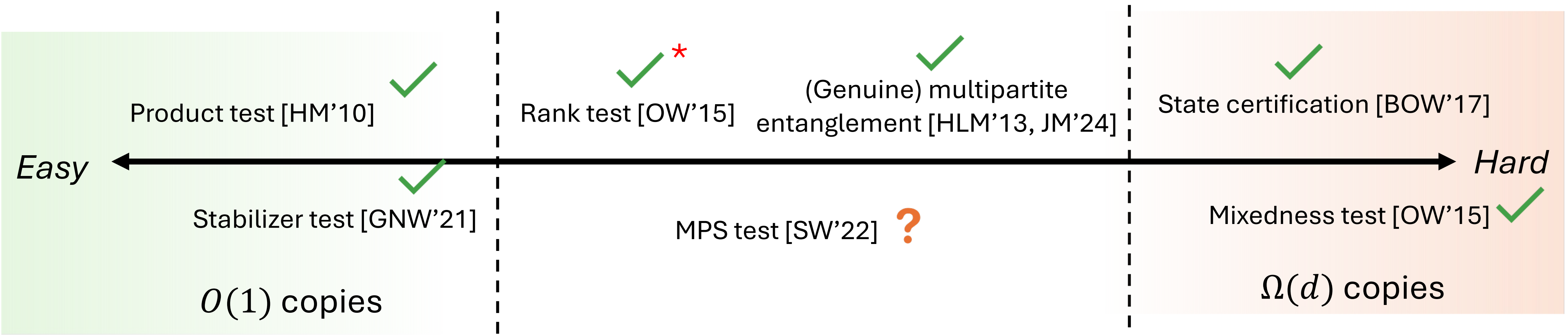
Soundness: $\mathrm{Tr}\left(M\psi^{\otimes N}\right) \leq b$

- "Successful" if $a - b = \Omega(1)$
- "Perfect completeness" if $a = 1$

Selected Prior Work



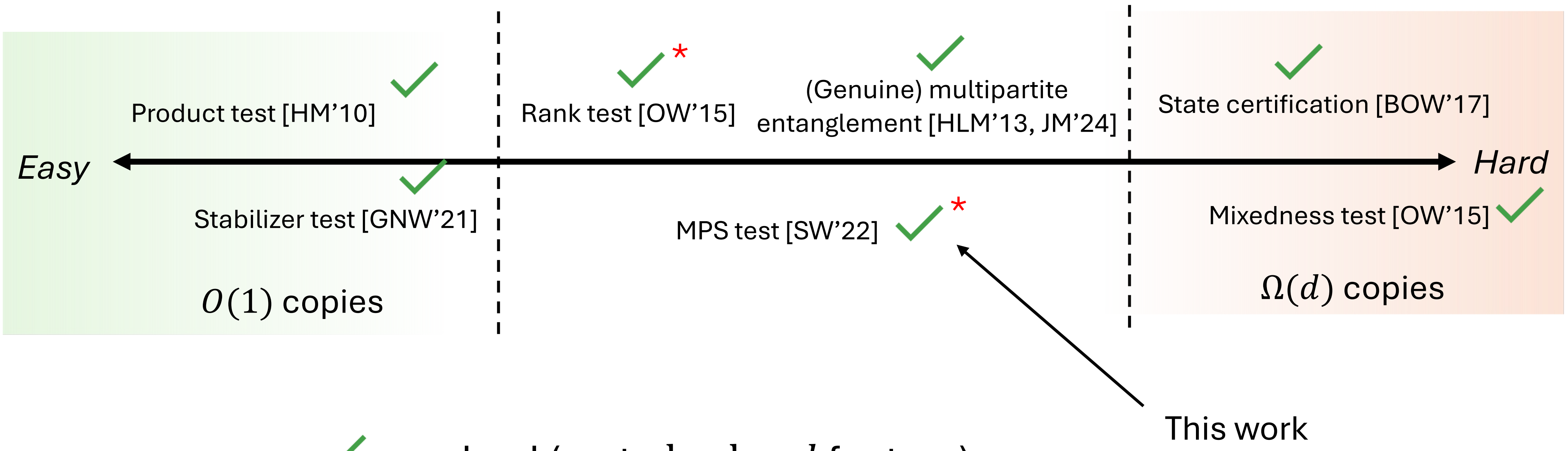
Selected Prior Work



✓ = solved (up to $\log \log d$ factors)

✓* = ...with perfect completeness

Selected Prior Work



✓ = solved (up to $\log \log d$ factors)

✓* = …with perfect completeness

Testing Tensor Network States

Testing Tensor Network States

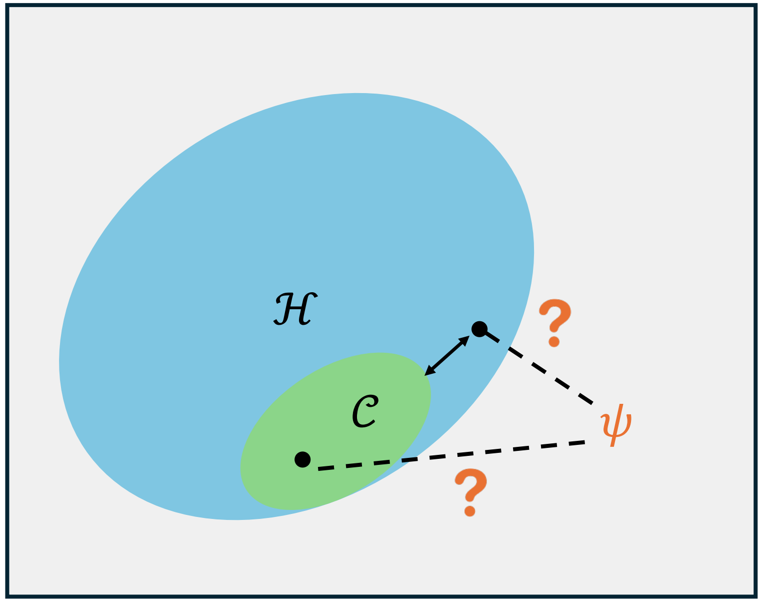
- MPS/TTNS, PEPS, etc. are important classes of states in QI and condensed matter physics

Testing Tensor Network States

- MPS/TTNS, PEPS, etc. are important classes of states in QI and condensed matter physics

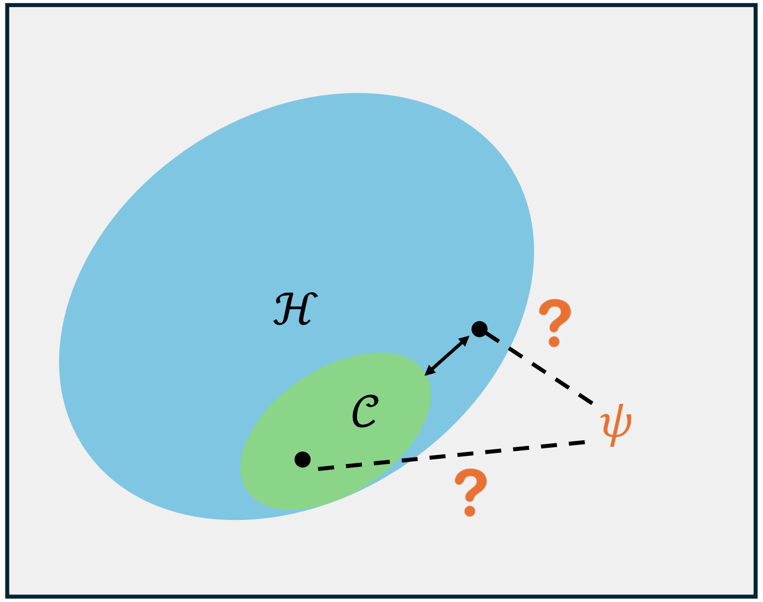
Testing Tensor Network States

- MPS/TTNS, PEPS, etc. are important classes of states in QI and condensed matter physics



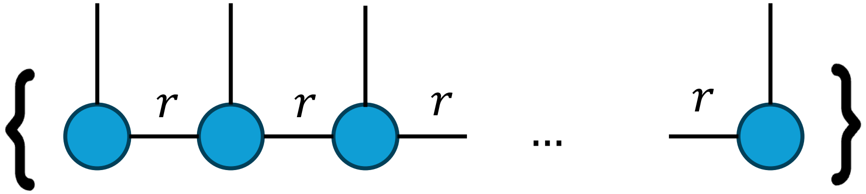
Testing Tensor Network States

- MPS/TTNS, PEPS, etc. are important classes of states in QI and condensed matter physics



MPS testing:

$\mathcal{H} = \mathbb{C}^{d_1} \otimes \mathbb{C}^{d_2} \otimes \cdots \otimes \mathbb{C}^{d_n}$

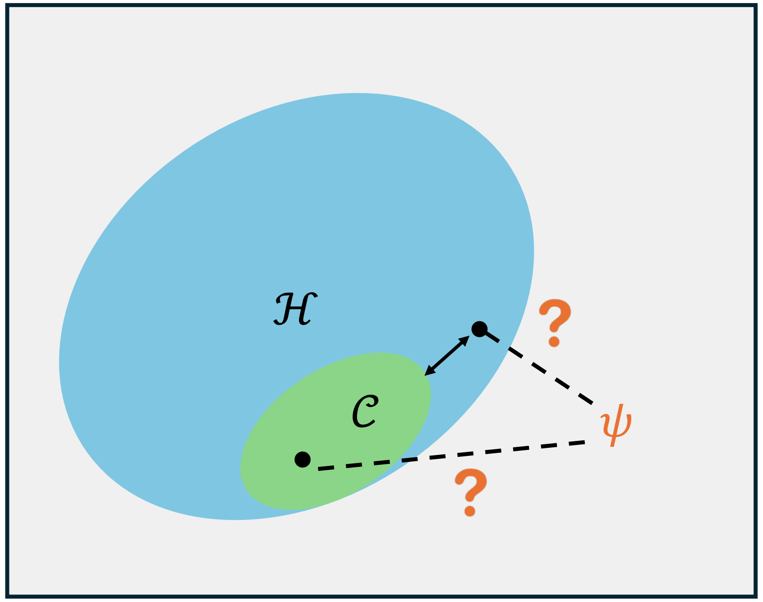
$\mathcal{C} = MPS(r) =$ 

$= \left\{\psi \in \mathcal{H}\colon \psi_{i_1 i_2 \dots i_n} = M^{[1]i_1}_{\alpha_1} M^{[2]i_2}_{\alpha_1\alpha_2} \dots M^{[n]i_{n-1}}_{\alpha_{n-2}\alpha_{n-1}} M^{[1]i_n}_{\alpha_n}\right\}$

$= \{\psi \in \mathcal{H}\colon SR(\psi) \leq r \ \ \forall e \in E\}$

Testing Tensor Network States

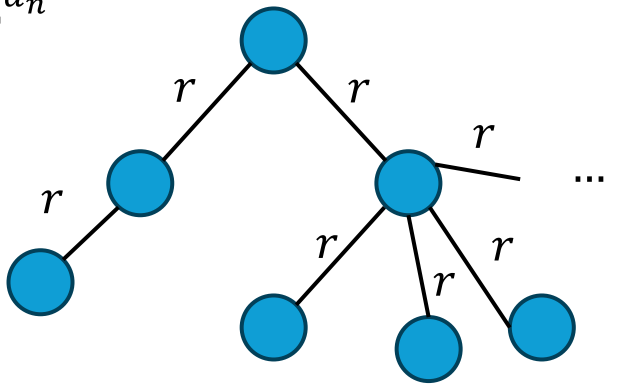
- MPS/TTNS, PEPS, etc. are important classes of states in QI and condensed matter physics



TTNS testing:

$$\mathcal{H} = \mathbb{C}^{d_1} \otimes \mathbb{C}^{d_2} \otimes \cdots \otimes \mathbb{C}^{d_n}$$

$$\mathcal{C} = TTNS(G, r) = \{$$



$$\}$$

$$= \{\psi \in \mathcal{H}:\ SR(\psi) \leq r\ \ \forall e \in E\}$$

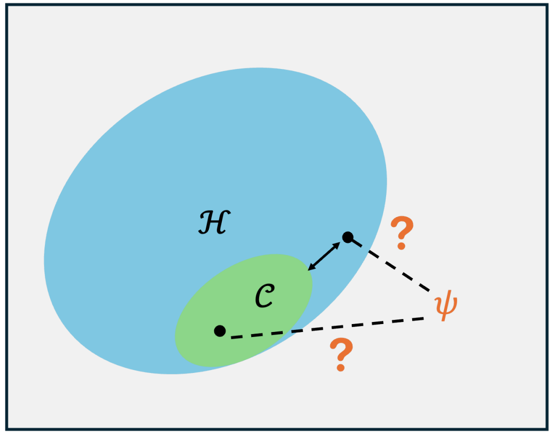
Main Result

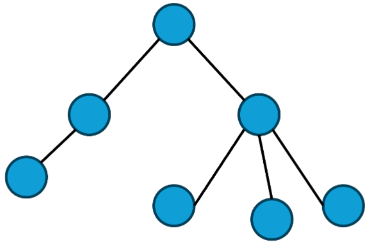
TTNS testing:

Fix any tree graph $G = (V, E)$ with n vertices.

$\mathcal{H} = \mathbb{C}^{d_1} \otimes \mathbb{C}^{d_2} \otimes \cdots \otimes \mathbb{C}^{d_n}$

$\mathcal{C} = TTNS(G, r) = \{\psi \in \mathcal{H}:\ SR(\psi) \leq r\ \ \forall e \in E\}$





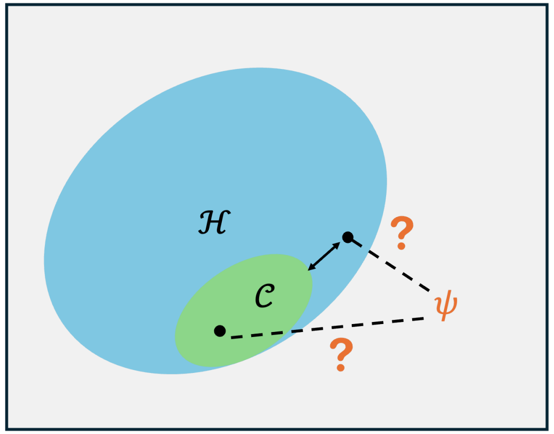
Main Result

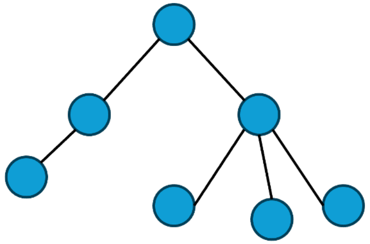
TTNS testing:

Fix any tree graph $G = (V, E)$ with n vertices.

$\mathcal{H} = \mathbb{C}^{d_1} \otimes \mathbb{C}^{d_2} \otimes \cdots \otimes \mathbb{C}^{d_n}$

$\mathcal{C} = TTNS(G, r) = \{\psi \in \mathcal{H}\colon\ SR(\psi) \leq r\ \ \forall e \in E\}$





Brief History

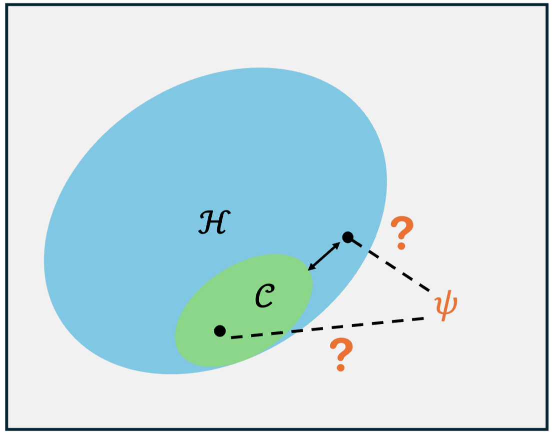
Main Result

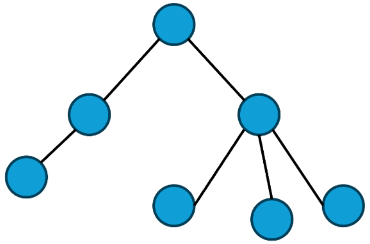
TTNS testing:

Fix any tree graph $G = (V, E)$ with n vertices.

$\mathcal{H} = \mathbb{C}^{d_1} \otimes \mathbb{C}^{d_2} \otimes \cdots \otimes \mathbb{C}^{d_n}$

$\mathcal{C} = TTNS(G, r) = \{\psi \in \mathcal{H}\colon SR(\psi) \le r \ \ \forall e \in E\}$





Brief History

- At $r = 1$ this is product testing: $\Theta(1)$ copies necessary and sufficient **[HM10]**
- For MPS at $r \geq 2$: $\Omega(\sqrt{n})$ copies necessary, and $O(nr^2)$ sufficient **[SW22]**
- For MPS at $r \leq 2^{n/8}$: $\Omega(\sqrt{r})$ copies necessary **[Aar+23]**
- For MPS at $r \geq 2$: $\Omega(\sqrt{nr} + r^2)^*$ copies necessary **[CWZ24]**

* *with perfect completeness*

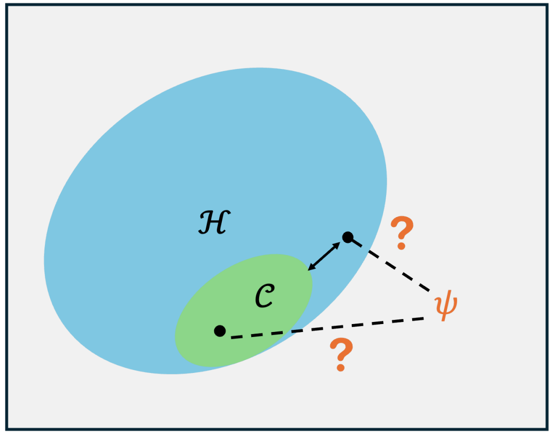
Main Result

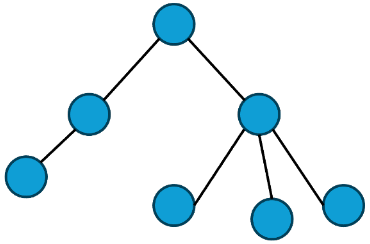
TTNS testing:

Fix any tree graph $G = (V, E)$ with n vertices.

$\mathcal{H} = \mathbb{C}^{d_1} \otimes \mathbb{C}^{d_2} \otimes \cdots \otimes \mathbb{C}^{d_n}$

$\mathcal{C} = TTNS(G, r) = \{\psi \in \mathcal{H} : SR(\psi) \leq r \;\; \forall e \in E\}$





Brief History

- At $r = 1$ this is product testing: $\Theta(1)$ copies necessary and sufficient **[HM10]**
- For MPS at $r \geq 2$: $\Omega(\sqrt{n})$ copies necessary, and $O(nr^2)$ sufficient **[SW22]**
- For MPS at $r \leq 2^{n/8}$: $\Omega(\sqrt{r})$ copies necessary **[Aar+23]**
- For MPS at $r \geq 2$: $\Omega(\sqrt{nr} + r^2)^*$ copies necessary **[CWZ24]**
- For any TTNS at $r \geq 2 + \log(n)$: $\Omega(nr^2/\log n)^*$ copies necessary, and $O(nr^2)$ copies sufficient **[This work]**

* *with perfect completeness*

I. Background & Results

II. Lower Bound for a Specific Algorithm

III. There is No Better Algorithm

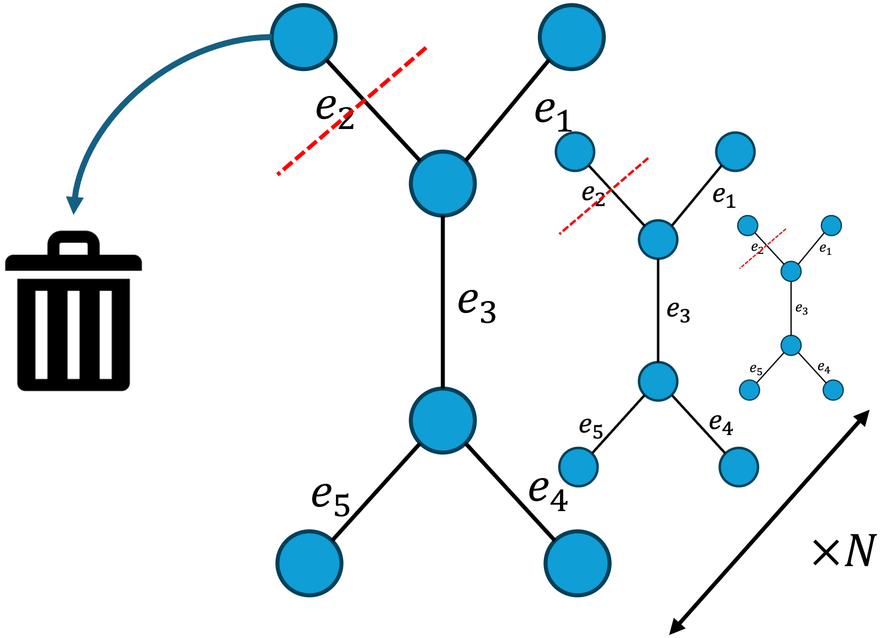
IV. Final Remarks

The Algorithm

Apply the rank test to every edge [SW'22]:

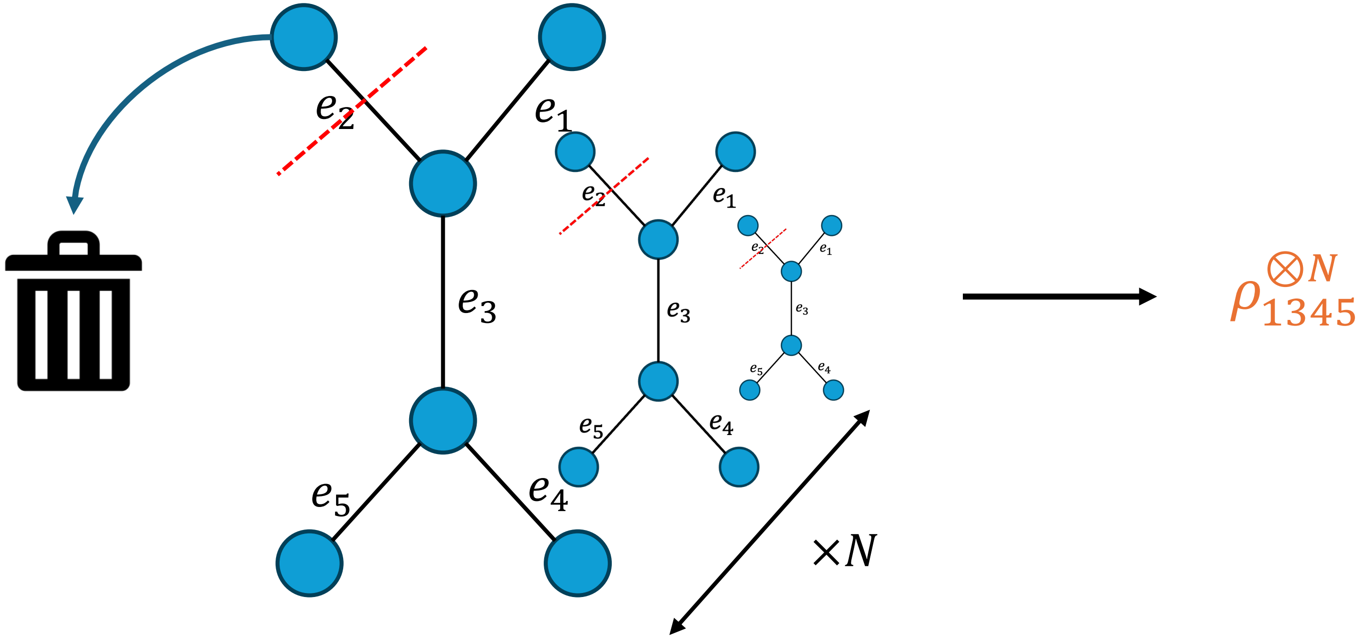
The Algorithm

Apply the rank test to every edge [SW'22]:



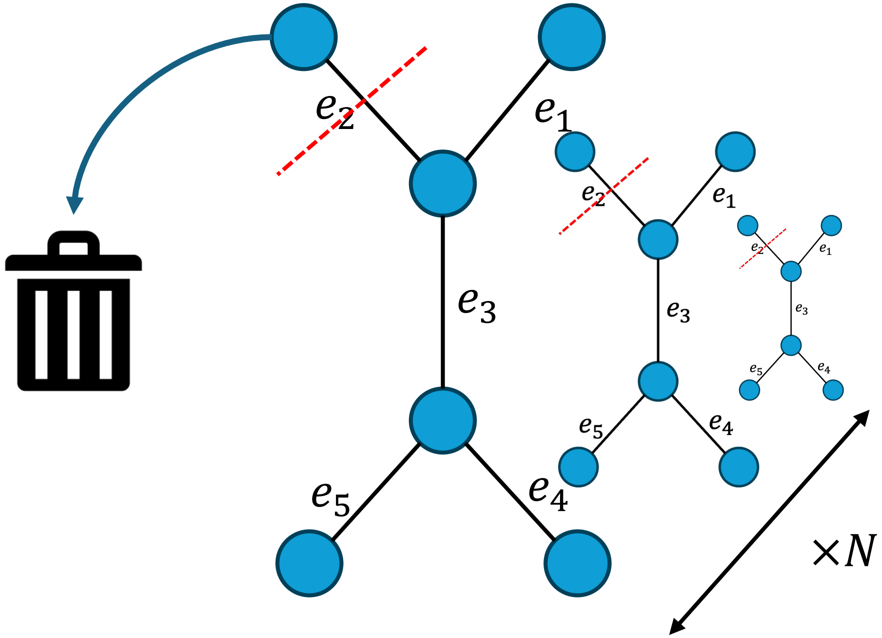
The Algorithm

Apply the rank test to every edge [SW'22]:



The Algorithm

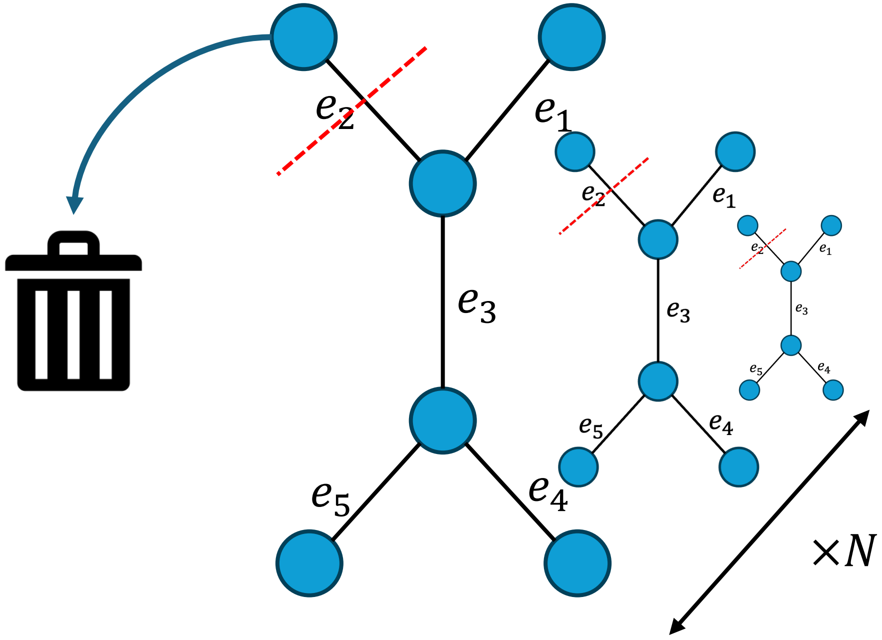
Apply the rank test to every edge [SW'22]:



$\longrightarrow \rho_{1345}^{\otimes N} \longrightarrow$ Is $\text{rank}(\rho) \leq r$?

The Algorithm

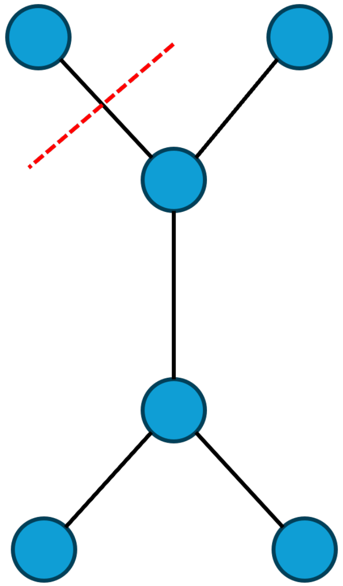
Apply the rank test to every edge [SW’22]:



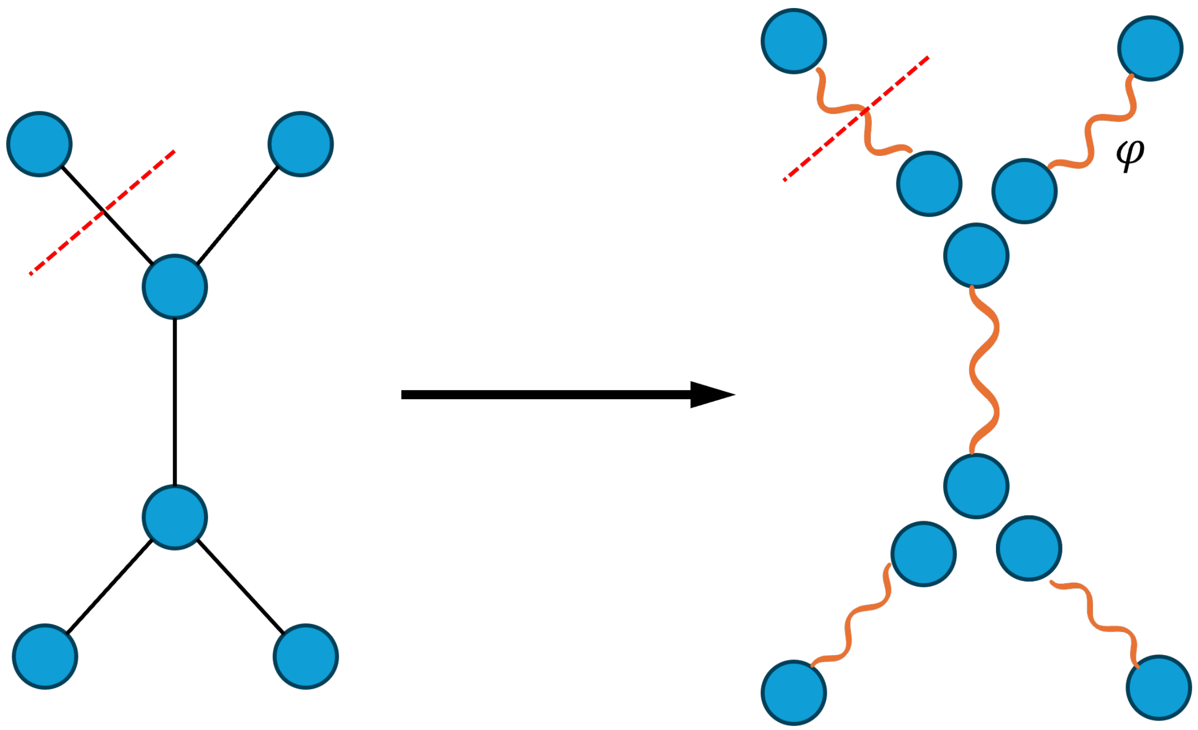
$\longrightarrow$ $\rho_{1345}^{\otimes N}$ $\longrightarrow$ Is $\text{rank}(\rho) \leq r$?

$N = O(nr^2)$ copies suffice to detect far-away state.

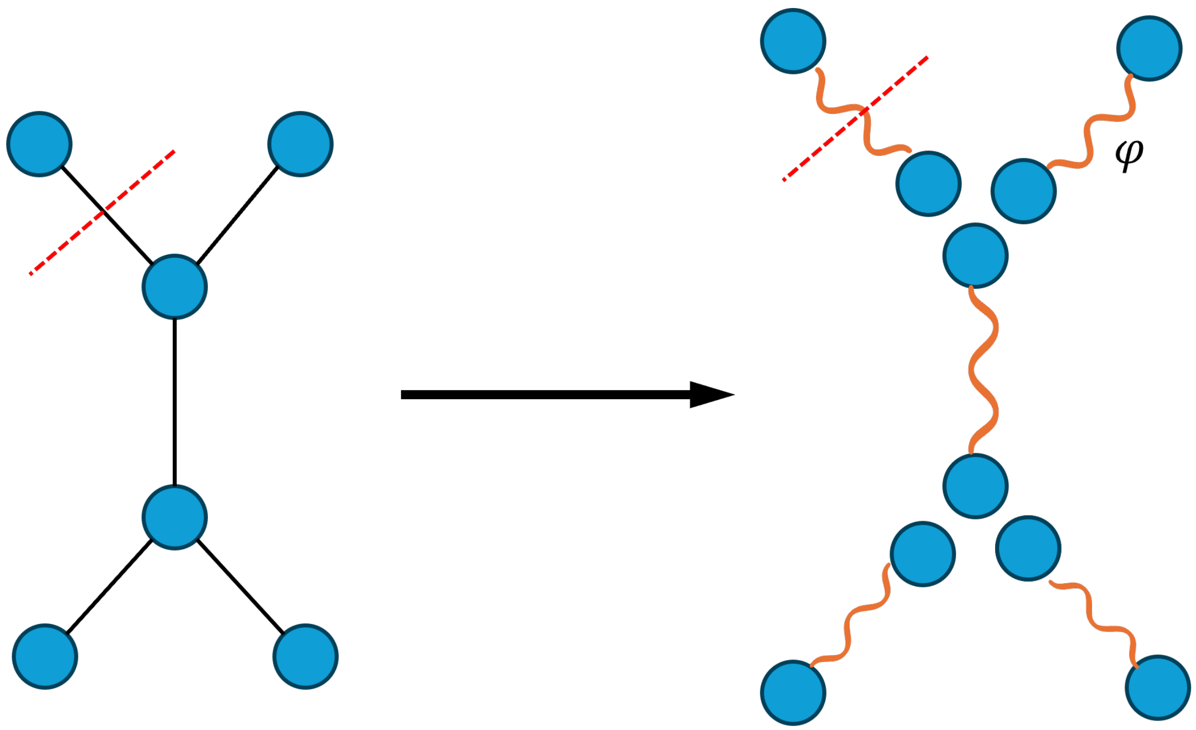
The Hard Instance



The Hard Instance

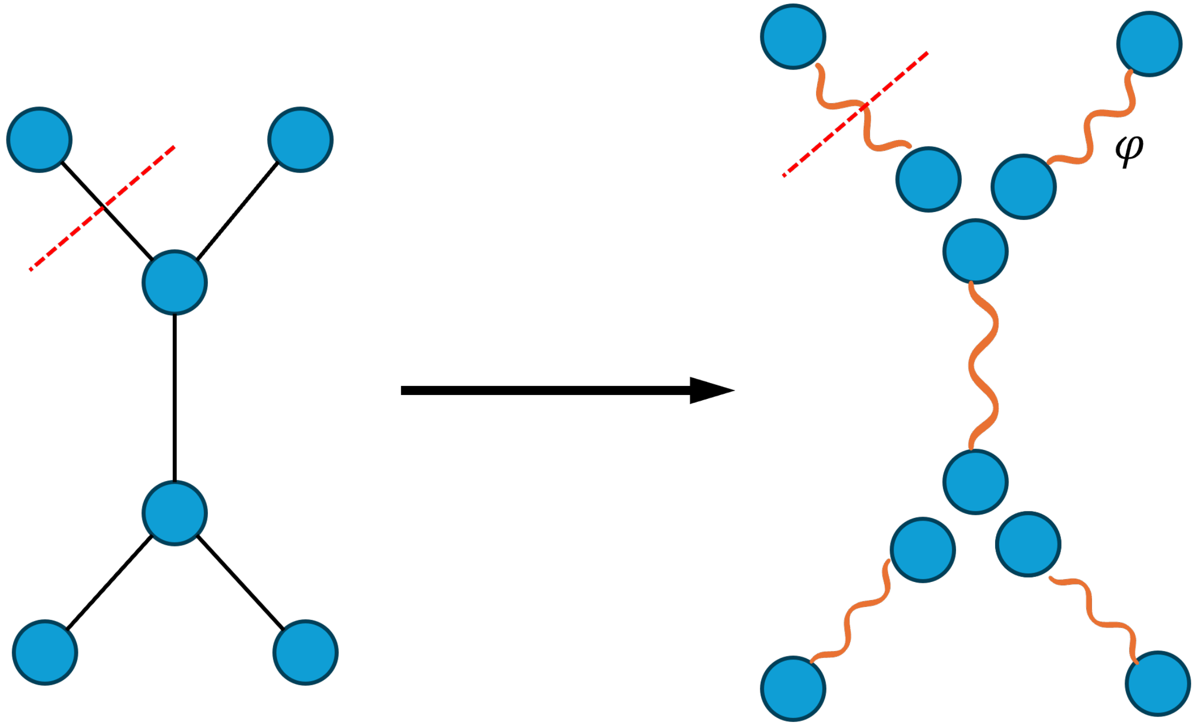


The Hard Instance



$$Schmidt(\varphi) = \left(\sqrt{1-\frac{\epsilon}{n}}, \sqrt{\frac{\epsilon}{n(d-1)}}, \sqrt{\frac{\epsilon}{n(d-1)}}, \dots, \sqrt{\frac{\epsilon}{n(d-1)}}\right)$$

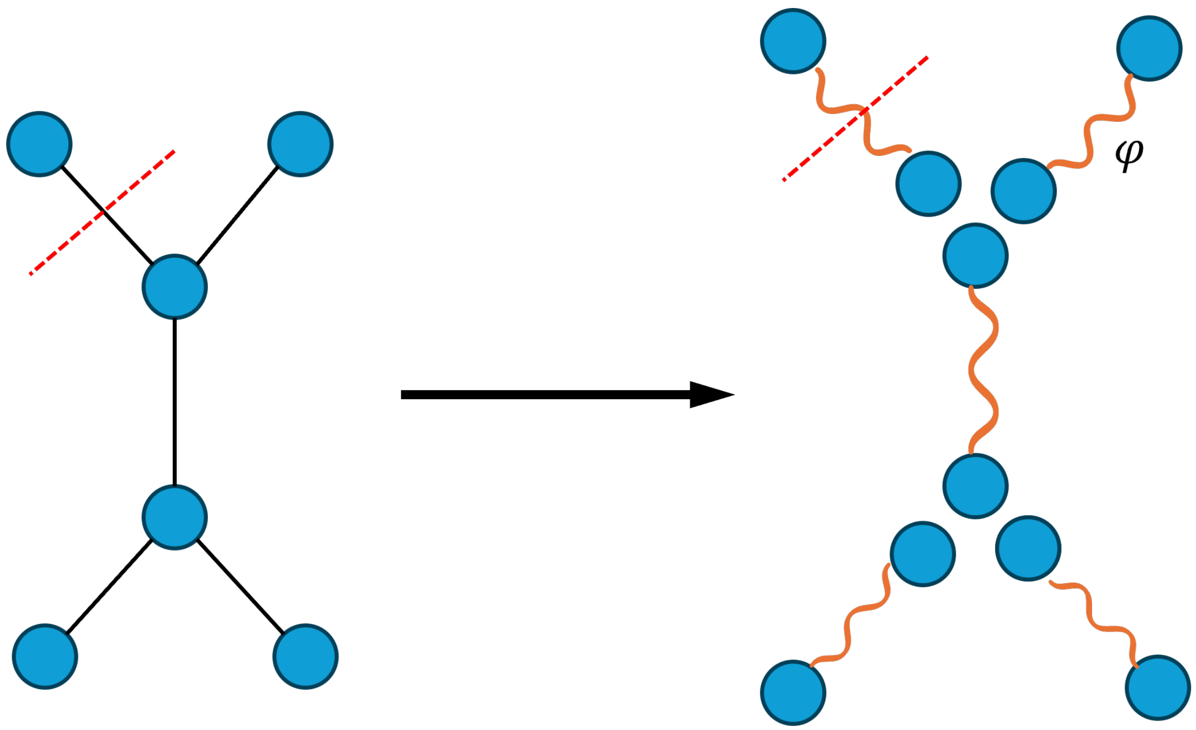
The Hard Instance



$$\mathrm{dist}\left(\varphi^{\otimes n-1}, TTNS\right) \approx \epsilon$$

$$Schmidt(\varphi) = \left(\sqrt{1-\frac{\epsilon}{n}}, \sqrt{\frac{\epsilon}{n(d-1)}}, \sqrt{\frac{\epsilon}{n(d-1)}}, \dots, \sqrt{\frac{\epsilon}{n(d-1)}}\right)$$

The Hard Instance

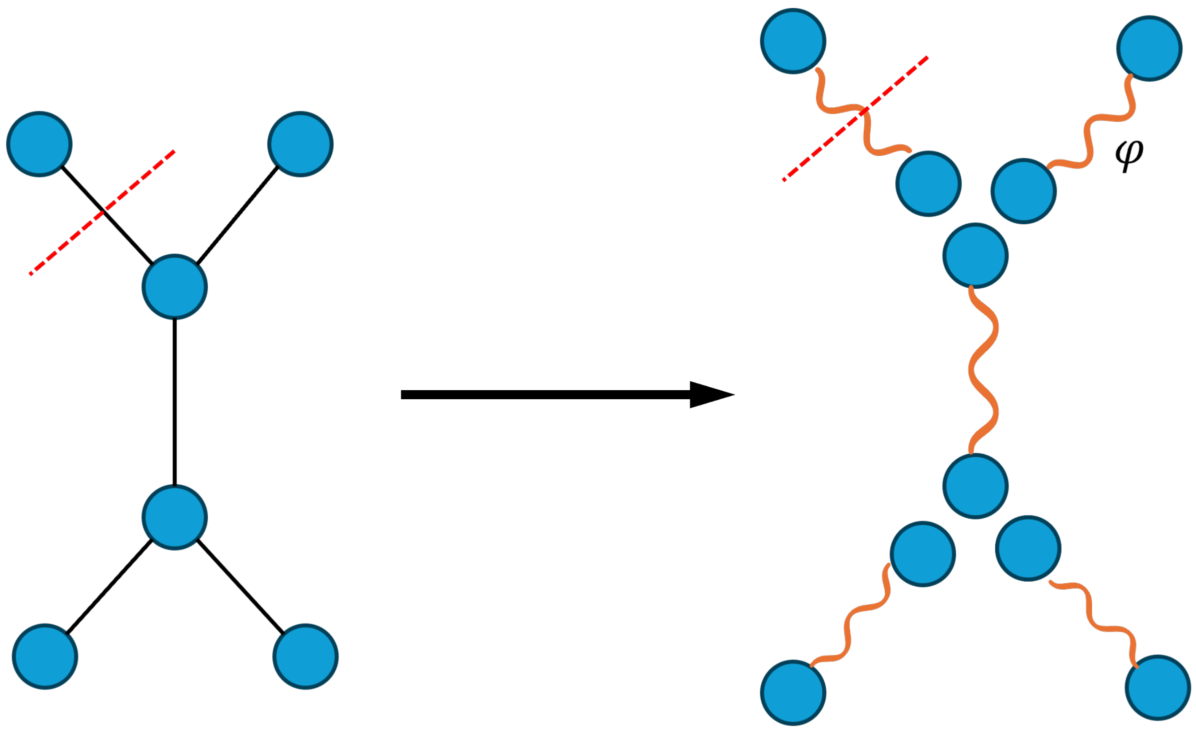


$\mathrm{dist}\left(\varphi^{\otimes n-1}, TTNS\right) \approx \epsilon$

But any reduced density matrix $\rho[\varphi]$ is quite close to rank-r

$$Schmidt(\varphi) = \left(\sqrt{1-\frac{\epsilon}{n}}, \sqrt{\frac{\epsilon}{n(d-1)}}, \sqrt{\frac{\epsilon}{n(d-1)}}, \dots, \sqrt{\frac{\epsilon}{n(d-1)}}\right)$$

The Hard Instance



$\mathrm{dist}(\varphi^{\otimes n-1}, TTNS) \approx \epsilon$

But any reduced density matrix $\rho[\varphi]$ is quite close to rank-r

$\Longrightarrow$ the rank test struggles to reject!

$$Schmidt(\varphi) = \left(\sqrt{1-\frac{\epsilon}{n}}, \sqrt{\frac{\epsilon}{n(d-1)}}, \sqrt{\frac{\epsilon}{n(d-1)}}, \ldots, \sqrt{\frac{\epsilon}{n(d-1)}}\right)$$

Rank Testing

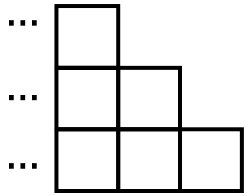
Fact from representation theory:

$$\rho^{\otimes N} \cong \bigoplus_{\lambda \vdash N} I_{\mathcal{P}_\lambda} \otimes q_\lambda^d(\rho)$$

Rank Testing

Fact from representation theory:

$$\rho^{\otimes N} \cong \bigoplus_{\lambda \vdash N} I_{\mathcal{P}_\lambda} \otimes q_\lambda^d(\rho)$$

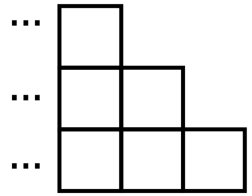


irreps of $\mathcal{S}_N$ and $\mathbb{U}(d)$

Rank Testing

Fact from representation theory:

$$\rho^{\otimes N} \cong \bigoplus_{\lambda \vdash N} I_{\mathcal{P}_\lambda} \otimes q_\lambda^d(\rho)$$



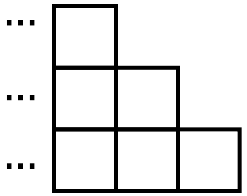
irreps of $\mathcal{S}_N$ and $\mathbb{U}(d)$

Due to symmetry, measuring λ is optimal. Accept if $\ell(\lambda) \leq r$.

Rank Testing

Fact from representation theory:

$$\rho^{\otimes N} \cong \bigoplus_{\lambda \vdash N} I_{\mathcal{P}_\lambda} \otimes q_\lambda^d(\rho)$$



irreps of $\mathcal{S}_N$ and $\mathbb{U}(d)$

Due to symmetry, measuring λ is optimal. Accept if $\ell(\lambda) \leq r$.

Schur polynomial

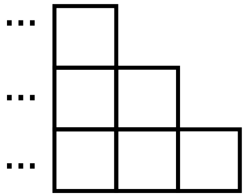
$$\text{Distributed as } p(\lambda) = \mathrm{Tr}\left(I_{\mathcal{P}_\lambda}\right)\mathrm{Tr}\left(q_\lambda^d(\rho)\right) = \dim(\mathcal{P}_\lambda)s_\lambda(p_1, \ldots, p_d)$$

Spectrum of ρ

Rank Testing

Fact from representation theory:

$$\rho^{\otimes N} \cong \bigoplus_{\lambda \vdash N} I_{\mathcal{P}_\lambda} \otimes q_\lambda^d(\rho)$$



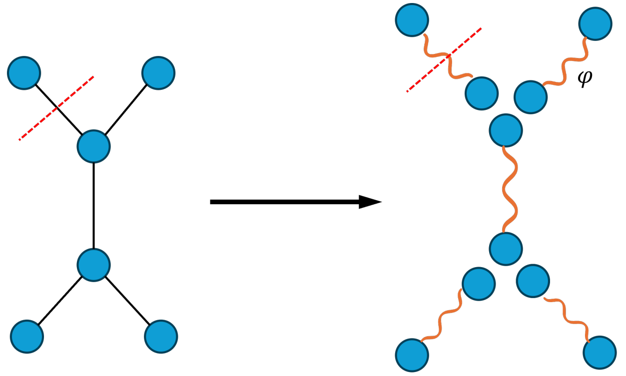
irreps of $\mathcal{S}_N$ and $\mathbb{U}(d)$

Due to symmetry, measuring λ is optimal. Accept if $\ell(\lambda) \leq r$.

Theorem [OW'15]: *Let $p = (p_1, \dots, p_d)$ be the spectrum of ρ and let $X_1, \dots, X_N \sim p$, iid. It holds that*

$$\mathrm{p_{acc}} \equiv \sum_{\lambda \vdash N : \ell(\lambda) \leq r} \mathrm{Tr}\left(\Pi_\lambda \rho^{\otimes N}\right) = \mathrm{Pr}[LDS(X) \leq r]$$

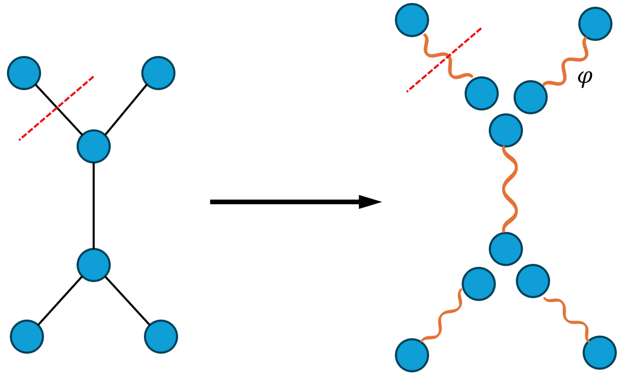
Lower Bound Proof Sketch



$$Schmidt(\varphi) = \left(\sqrt{1-\frac{\epsilon}{n}}, \sqrt{\frac{\epsilon}{n(d-1)}}, \sqrt{\frac{\epsilon}{n(d-1)}}, \dots, \sqrt{\frac{\epsilon}{n(d-1)}}\right)$$

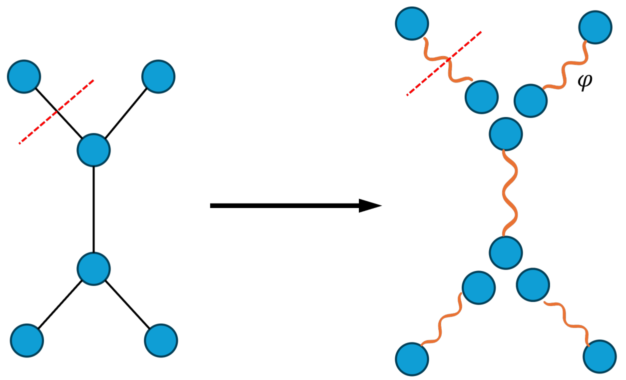
Lower Bound Proof Sketch

The TTNS tester performs the rank test with respect to every cut, so acceptance probability is $(\mathrm{p}_{acc})^{n-1}$.



$$Schmidt(\varphi) = \left(\sqrt{1-\frac{\epsilon}{n}}, \sqrt{\frac{\epsilon}{n(d-1)}}, \sqrt{\frac{\epsilon}{n(d-1)}}, \dots, \sqrt{\frac{\epsilon}{n(d-1)}}\right)$$

Lower Bound Proof Sketch



$$Schmidt(\varphi) = \left(\sqrt{1-\frac{\epsilon}{n}}, \sqrt{\frac{\epsilon}{n(d-1)}}, \sqrt{\frac{\epsilon}{n(d-1)}}, \dots, \sqrt{\frac{\epsilon}{n(d-1)}}\right)$$

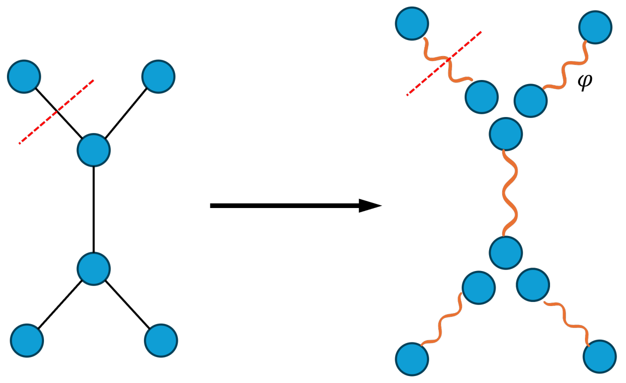
The TTNS tester performs the rank test with respect to every cut, so acceptance probability is $(\mathrm{p}_{acc})^{n-1}$.

Recall: Rank test accepts with probability

$$\mathrm{p}_{acc} = \Pr[LDS(X) \leq r]$$

where X is random word with letters sampled according to $spec(\rho)$.

Lower Bound Proof Sketch



$$Schmidt(\varphi) = \left(\sqrt{1-\frac{\epsilon}{n}}, \sqrt{\frac{\epsilon}{n(d-1)}}, \sqrt{\frac{\epsilon}{n(d-1)}}, \dots, \sqrt{\frac{\epsilon}{n(d-1)}}\right)$$

The TTNS tester performs the rank test with respect to every cut, so acceptance probability is $(\mathrm{p}_{acc})^{n-1}$.

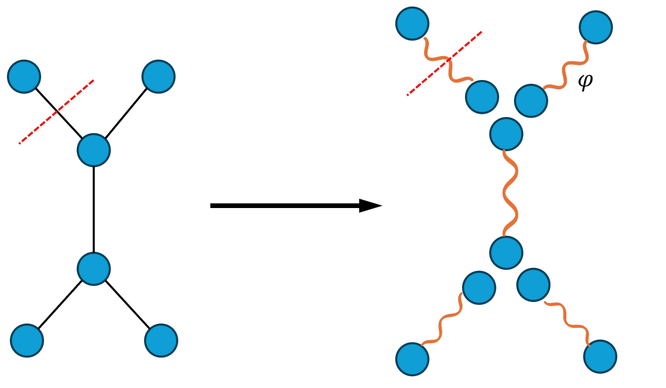
Recall: Rank test accepts with probability

$$\mathrm{p}_{acc} = \Pr[LDS(X) \le r]$$

where X is random word with letters sampled according to $spec(\rho)$.

We get an N-letter word like $X = 1111811111113511121 \ldots$

Lower Bound Proof Sketch



The TTNS tester performs the rank test with respect to every cut, so acceptance probability is $(\mathrm{p}_{acc})^{n-1}$.

Recall: Rank test accepts with probability

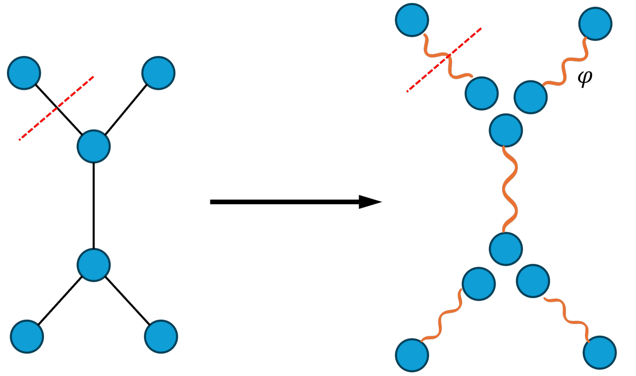
$$\mathrm{p}_{acc} = \Pr[LDS(X) \leq r]$$

where X is random word with letters sampled according to $spec(\rho)$.

We get an N-letter word like $X = 11118111111351112 1 \ldots$

$$Y \equiv 8352 \ldots$$

Lower Bound Proof Sketch



$$Schmidt(\varphi) = \left(\sqrt{1-\frac{\epsilon}{n}}, \sqrt{\frac{\epsilon}{n(d-1)}}, \sqrt{\frac{\epsilon}{n(d-1)}}, \dots, \sqrt{\frac{\epsilon}{n(d-1)}}\right)$$

The TTNS tester performs the rank test with respect to every cut, so acceptance probability is $(\mathrm{p}_{acc})^{n-1}$.

Recall: Rank test accepts with probability

$$\mathrm{p}_{acc} = \Pr[LDS(X) \leq r]$$

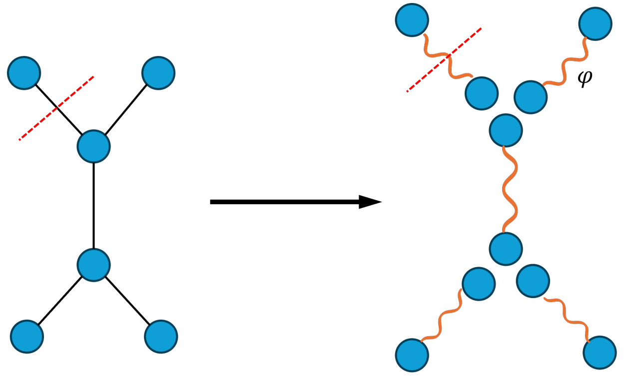
where X is random word with letters sampled according to $spec(\rho)$.

We get an N-letter word like $X = 11118111111351112 1 \dots$

$$Y \equiv 8352 \dots$$

Y is uniformly random, L-letter word → (Counting) $\Pr[LDS(Y) \leq r] \approx 1 - \left(\frac{L}{r^2}\right)^r$

Lower Bound Proof Sketch



$$Schmidt(\varphi) = \left(\sqrt{1-\frac{\epsilon}{n}}, \sqrt{\frac{\epsilon}{n(d-1)}}, \sqrt{\frac{\epsilon}{n(d-1)}}, \dots, \sqrt{\frac{\epsilon}{n(d-1)}}\right)$$

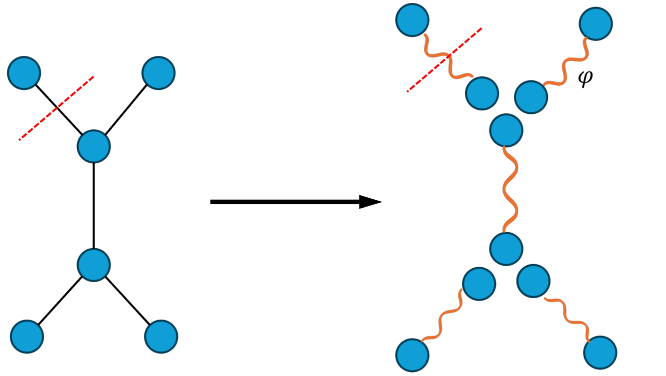
The TTNS tester performs the rank test with respect to every cut, so acceptance probability is $(\mathrm{p}_{acc})^{n-1}$.

We get an N-letter word like $X = 11118111111351112 1 \dots$

$$Y \equiv 8352 \dots$$

Y is uniformly random, L-letter word $\xrightarrow[\text{Counting}]{}$ $\Pr[LDS(Y) \leq r] \approx 1 - \left(\frac{L}{r^2}\right)^r$

Lower Bound Proof Sketch



$$Schmidt(\varphi) = \left(\sqrt{1-\frac{\epsilon}{n}}, \sqrt{\frac{\epsilon}{n(d-1)}}, \sqrt{\frac{\epsilon}{n(d-1)}}, \dots, \sqrt{\frac{\epsilon}{n(d-1)}}\right)$$

The TTNS tester performs the rank test with respect to every cut, so acceptance probability is $(\mathrm{p}_{acc})^{n-1}$.

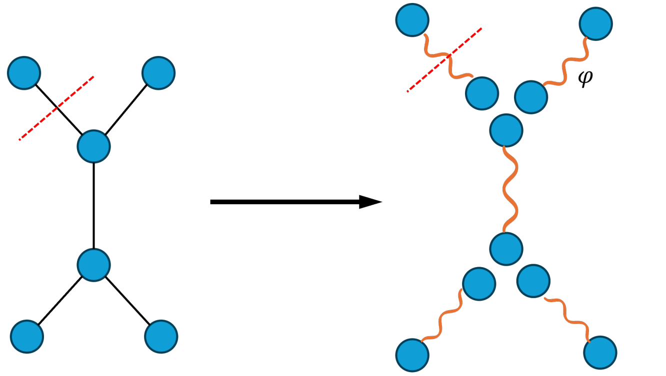
We get an N-letter word like $X = 1111811111135111121\ldots$

$$Y \equiv 8352\ldots$$

Y is uniformly random, L-letter word $\xrightarrow{\text{Counting}}$ $\Pr[LDS(Y) \le r] \approx 1 - \underbrace{\left(\frac{L}{r^2}\right)^r}_{\bigstar}$

L = length of $Y \approx \frac{N\epsilon}{n}$. So suppose $N \ll nr^2/(\log n)$. Then $\bigstar \approx 1 - \left(\frac{1}{\log n}\right)^r$.

Lower Bound Proof Sketch



$$Schmidt(\varphi) = \left(\sqrt{1-\frac{\epsilon}{n}}, \sqrt{\frac{\epsilon}{n(d-1)}}, \sqrt{\frac{\epsilon}{n(d-1)}}, \dots, \sqrt{\frac{\epsilon}{n(d-1)}}\right)$$

The TTNS tester performs the rank test with respect to every cut, so acceptance probability is $(\mathrm{p}_{acc})^{n-1}$.

We get an N-letter word like $X = 11118111111351112 1 \dots$

$$Y \equiv 8352 \dots$$

Y is uniformly random, L-letter word ⟶ (Counting) $\Pr[LDS(Y) \leq r] \approx 1 - \underbrace{\left(\frac{L}{r^2}\right)^r}_{★}$

$L =$ length of $Y \approx \frac{N\epsilon}{n}$. So suppose $N \ll nr^2/(\log n)$. Then ★ $\approx 1 - \left(\frac{1}{\log n}\right)^r$.

So if $r \approx \log n$ then $(\mathrm{p}_{acc})^{n-1} \approx \left(1 - \left(\frac{1}{\log n}\right)^{\log n}\right)^n \to 1.$

I. Background & Results
II. Lower Bound for a Specific Algorithm
III. There is No Better Algorithm
IV. Conclusion

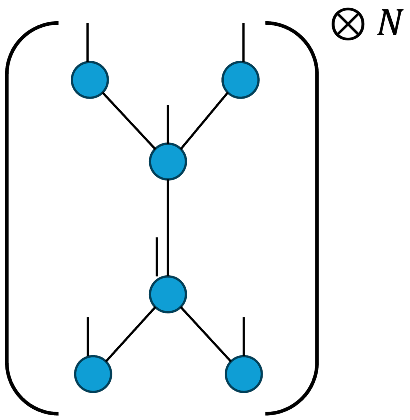
Lemma: *The optimal test with perfect completeness for any class $\mathcal{C}$ of* pure *states is*

$$P_{acc} := \text{proj span}\left\{|\phi\rangle^{\otimes N} : |\phi\rangle \in \mathcal{C}\right\}.$$

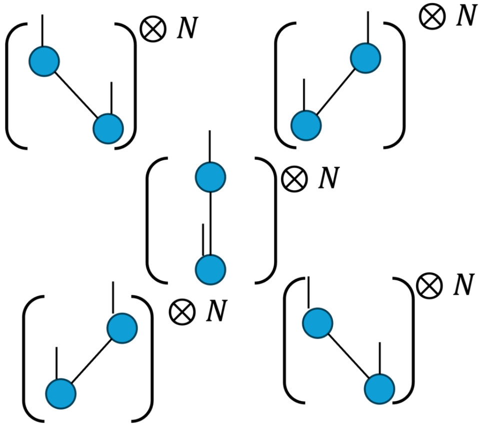
Lemma: *The optimal test with perfect completeness for any class $\mathcal{C}$ of pure states is*

$$P_{acc} \coloneqq \text{proj span}\left\{|\phi\rangle^{\otimes N} : |\phi\rangle \in \mathcal{C}\right\}.$$

$$\text{proj span}\left\{|\phi\rangle^{\otimes N} : |\phi\rangle \in TTNS(G,r)\right\} = \text{proj span}$$

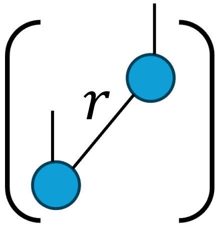


$$\succcurlyeq \text{proj span}$$



Lemma: *The optimal test with perfect completeness for any class $\mathcal{C}$ of pure states is*

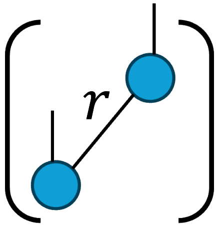
$$P_{acc} := \text{proj span}\left\{|\phi\rangle^{\otimes N} : |\phi\rangle \in \mathcal{C}\right\}.$$

proj span 

$^{\otimes N}$ $= \text{proj span}\left\{|\varphi\rangle^{\otimes N} :\ \text{SR}(|\varphi\rangle) \leq r\right\}$

Lemma: *The optimal test with perfect completeness for any class $\mathcal{C}$ of pure states is*

$$P_{acc} := \text{proj span}\left\{|\phi\rangle^{\otimes N} : |\phi\rangle \in \mathcal{C}\right\}.$$

proj span 

$^{\otimes N}$

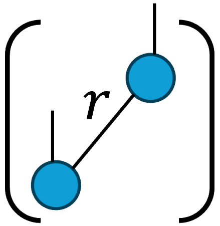
$$= \text{proj span}\left\{|\varphi\rangle^{\otimes N}:\ \text{SR}(|\varphi\rangle) \leq r\right\}$$

$$= \left(\text{RankTest} \otimes I_B^{\otimes N}\right)\Pi_{sym}$$

Lemma: *The optimal test with perfect completeness for any class $\mathcal{C}$ of pure states is*

$$P_{acc} \coloneqq \text{proj span}\left\{|\phi\rangle^{\otimes N} : |\phi\rangle \in \mathcal{C}\right\}.$$

proj span



$\otimes N$

$$= \text{proj span}\left\{|\varphi\rangle^{\otimes N} :\ \text{SR}(|\varphi\rangle) \leq r\right\}$$

$$= \left(\text{RankTest} \otimes I_B^{\otimes N}\right)\Pi_{sym}$$

In general, to test properties of bipartite entanglement, might as well discard Bob's system [MH07, CWZ24, Har05].

I. Background & Results
II. Lower Bound for a Specific Algorithm
III. There is No Better Algorithm
IV. Final Remarks

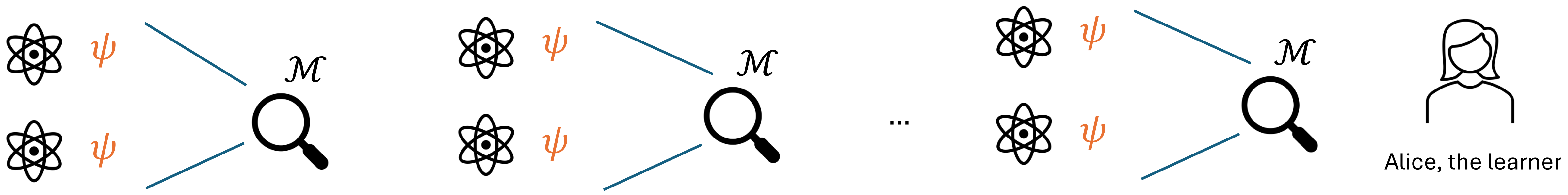
Conclusion

- We give nearly tight bounds for testing MPS/TTNS with one-sided error when bond dimension grows logarithmically.
- We also analyze few-copy tests (not discussed here)
- Open questions:
  - What happens at constant bond dimension? We suspect $O(\sqrt{n})$ copies could suffice. (See bonus slides.)
  - What about two-sided error? (We don't even know the answer for rank testing.)
  - What is the copy complexity of *learning* MPS?
  - What is the copy complexity of learning/testing PEPS, or MPS with CBC?

Bonus slides

Few-copy Tests

- Besides copy complexity, an important resource is quantum memory, i.e., number of copies measured simultaneously
- For the product test (MPS/TTNS testing at $r = 1$), only two copies at a time:

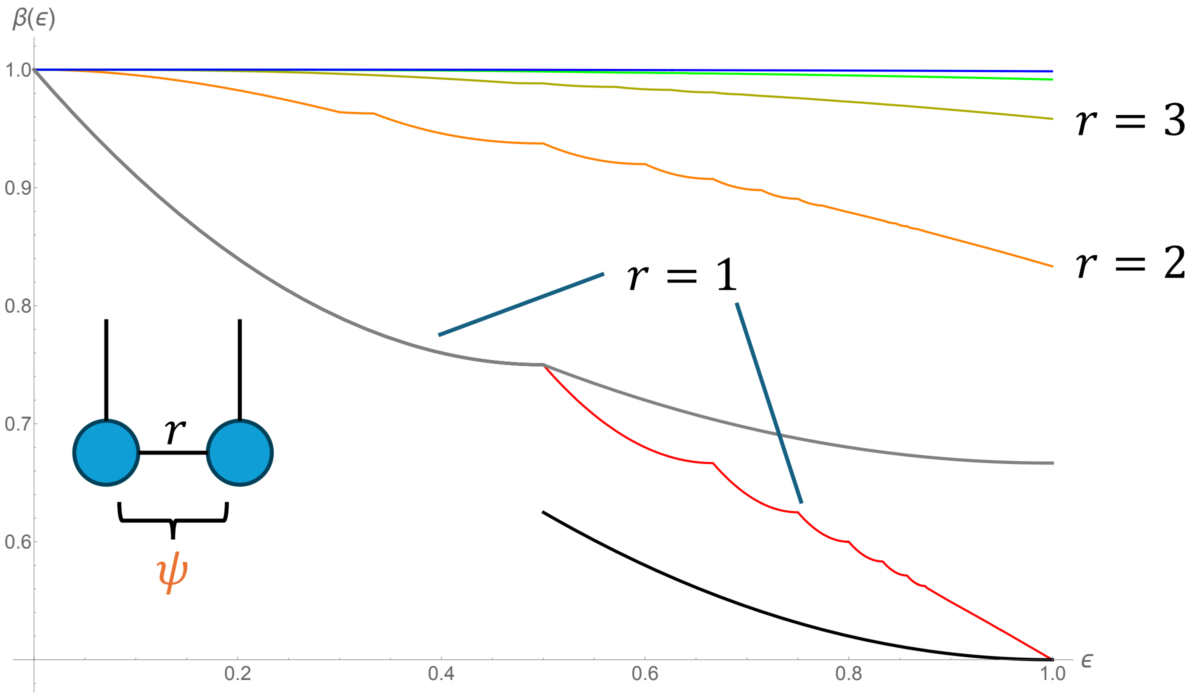


Q: What about $r \geq 2$?

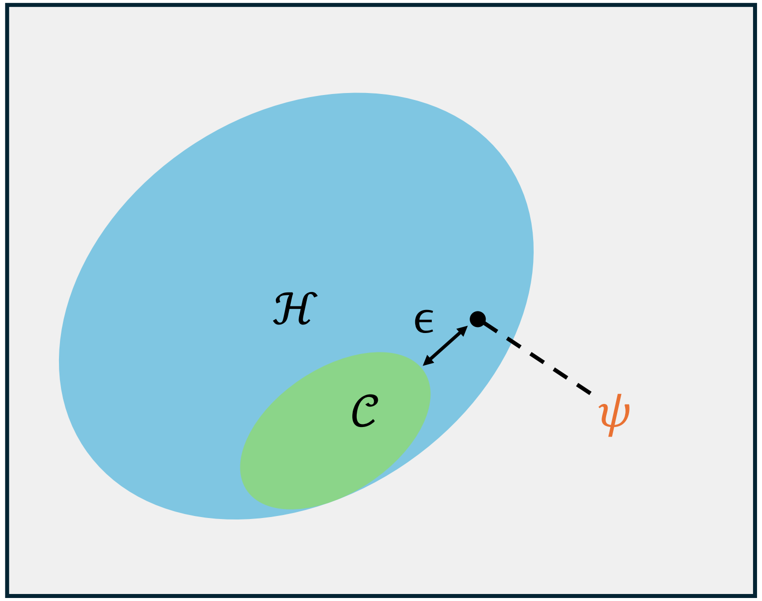
Few-copy Tests

Theorem: *For any $r \geq 2$, consider testing $\psi \in TTNS(G, r)$ with measurements on $(r+1)$ copies at a time (w/ one-sided error). It holds that $O(n^r)$ total copies suffice and $\Omega(n^{r-1})$ copies are necessary.*

$n = 2$, optimal Schmidt-rank test



$\beta(\epsilon)$ = acceptance probability on ϵ-far state



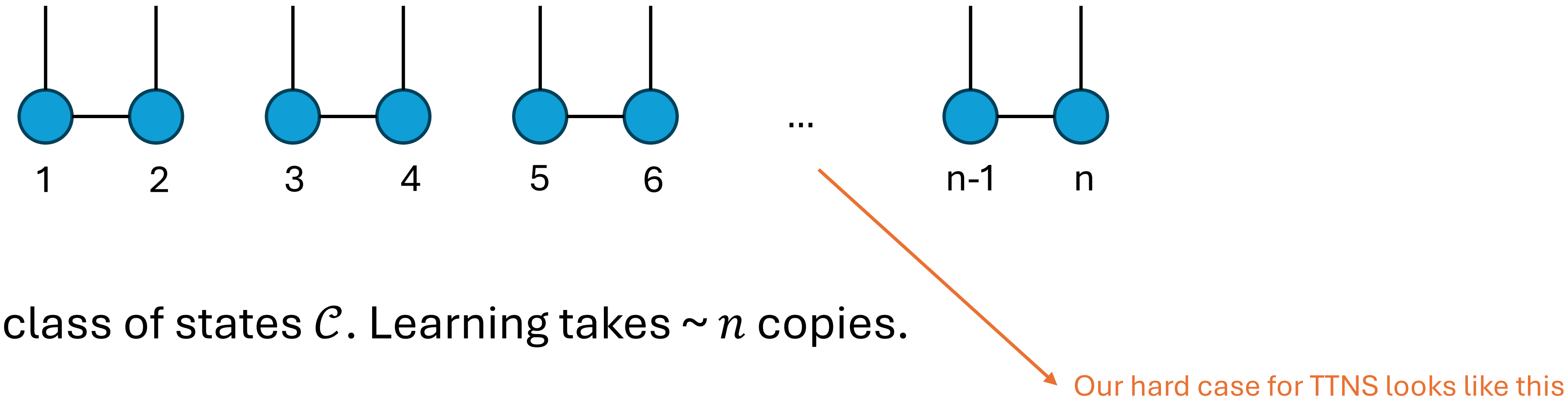
Constant Bond Dimension?

- For any TTNS at $r \geq 2 + \log(n)$: $\Omega(nr^2/\log n)^*$ copies necessary, and $O(nr^2)$ copies sufficient **[This work]**

* *with perfect completeness*

Q: Why do we need $r \geq 2 + log\ n$?

Take $r = 2$ and "forget" half the bonds:



This is a valid class of states $\mathcal{C}$. Learning takes $\sim n$ copies.

This work: testing $\mathcal{C}$ possible using just $O(\sqrt{n})$ copies.