Separating quantum and classical proofs and advice with classical oracles

John Bostanci

Based on work with Jonas Haferkamp, Chinmay Nirkhe, and Mark Zhandry
and with Andrew Huang, and Vinod Vaikuntanathan

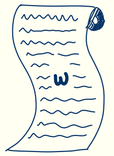
The power of proofs

Efficient Classical Algorithm

P vs NP

Unbounded prover

Classical proof



Efficient Classical Algorithm

The power of proofs

Efficient Classical Algorithm

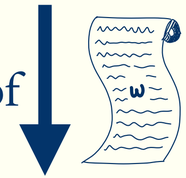
vs

Efficient Quantum Algorithm

vs

Unbounded prover

Classical proof



Efficient Classical Algorithm

The power of proofs

$N = pq$

Efficient Classical Algorithm

vs

Efficient Quantum Algorithm

p, q

vs

Unbounded prover

Classical proof

Efficient Classical Algorithm

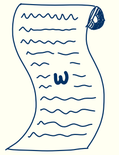
The power of quantum proofs

Efficient Quantum Algorithm

vs

Unbounded prover

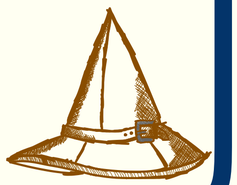
Classical proof



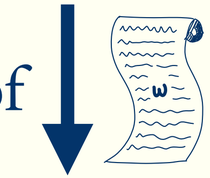
Efficient Quantum Algorithm

The power of quantum proofs

Unbounded prover



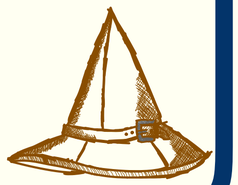
Classical proof



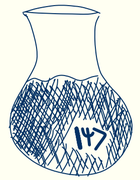
Efficient Quantum Algorithm

vs

Unbounded prover



Quantum proof

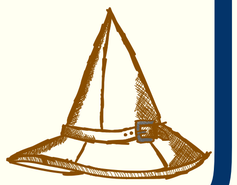


Efficient Quantum Algorithm

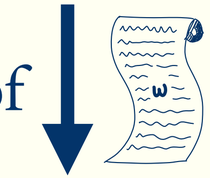
The power of quantum proofs

QCMA vs QMA

Unbounded prover

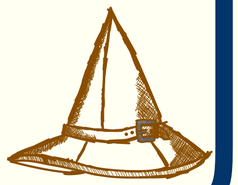


Classical proof

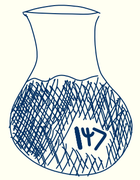


Efficient Quantum Algorithm

Unbounded prover



Quantum proof



Efficient Quantum Algorithm

Quantum versus classical proofs

QMA captures a lot of things we want to know about the world:

- Is there a thing that satisfies some physical property?
- Are two different physical processes doing the same transformation?
- Are local views of a system consistent with some quantum system?

Quantum versus classical proofs

QMA captures a lot of things we want to know about the world:

- Is there a thing that satisfies some physical property?
- Are two different physical processes doing the same transformation?
- Are local views of a system consistent with some quantum system?

→ If there were ways to check these properties with a classical proof, it would say something about the “representation complexity” of quantum systems!

Quantum versus classical proofs

QMA captures a lot of things we want to know about the world:

- Is there a thing that satisfies some physical property?
- Are two different physical processes doing the same transformation?
- Are local views of a system consistent with some quantum system?

→ If there were ways to check these properties with a classical proof, it would say something about the “representation complexity” of quantum systems!

→ If not, then quantum proofs really do allow us to verify something that a classical proof can’t. It will be interesting to understand why classical strings fail!

What can we say?

Unfortunately, proving an unconditional separation between the two classes would imply $\mathsf{P} \neq \mathsf{PSPACE}$, among other things.

$$\mathsf{P} \subseteq \mathsf{QCMA} \subseteq \mathsf{QMA} \subseteq \mathsf{PSPACE}$$

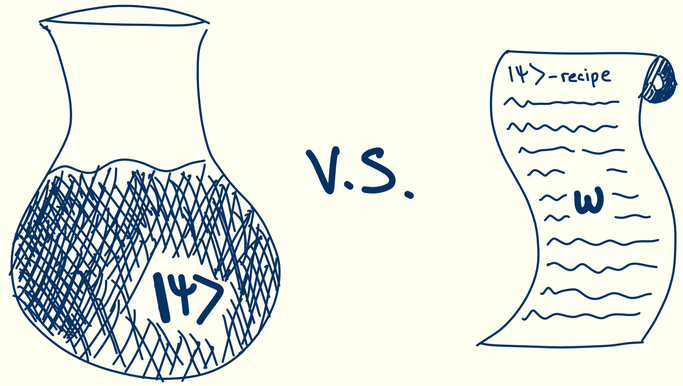
What can we say?

Unfortunately, proving an unconditional separation between the two classes would imply $\mathsf{P} \neq \mathsf{PSPACE}$, among other things.

$$\mathsf{P} \subseteq \mathsf{QCMA} \subseteq \mathsf{QMA} \subseteq \mathsf{PSPACE}$$

Next best thing: Oracle separation!

We will prove that there is a classical oracle relative to which QMA ≠ QCMA



History of the QMA versus QCMA problem

- First proposed in ’02 by **Aharanov and Naveh**.

History of the QMA versus QCMA problem

- First proposed in ’02 by **Aharanov and Naveh**.

- **Aaronson & Kuperberg ’06**: Quantum oracle separation. Each $\mathcal{O}_n = \mathrm{id} - 2\,|\psi_n\rangle\langle\psi_n|$

History of the QMA versus QCMA problem

- First proposed in ’02 by **Aharanov and Naveh**.
- **Aaronson & Kuperberg ’06**: Quantum oracle separation. Each $\mathcal{O}_n = \mathrm{id} - 2\,|\psi_n\rangle\langle\psi_n|$
- **Lutomirski ’11**: Proposed the expander mixing problem as a candidate classical oracle separation.

History of the QMA versus QCMA problem

- First proposed in ’02 by **Aharanov and Naveh**.
- **Aaronson & Kuperberg ’06**: Quantum oracle separation. Each $\mathcal{O}_n = \mathrm{id} - 2\,|\psi_n\rangle\langle\psi_n|$
- **Lutomirski ’11**: Proposed the expander mixing problem as a candidate classical oracle separation.
- **Fefferman & Kimmel ’15**: In-place permutation oracle. Problem corresponds to set size estimation.

History of the QMA versus QCMA problem

- First proposed in ’02 by **Aharanov and Naveh**.
- **Aaronson & Kuperberg ’06**: Quantum oracle separation. Each $\mathcal{O}_n = \mathrm{id} - 2\,|\psi_n\rangle\langle\psi_n|$
- **Lutomirski ’11**: Proposed the expander mixing problem as a candidate classical oracle separation.
- **Fefferman & Kimmel ’15**: In-place permutation oracle. Problem corresponds to set size estimation.
- **Natarajan & Nirkhe ’22**: Distribution testing oracle. Problem corresponds to size estimation of an expander graph.

History of the QMA versus QCMA problem

- First proposed in ’02 by **Aharanov and Naveh**.
- **Aaronson & Kuperberg ’06**: Quantum oracle separation. Each $\mathcal{O}_n = \mathrm{id} - 2\,|\psi_n\rangle\langle\psi_n|$
- **Lutomirski ’11**: Proposed the expander mixing problem as a candidate classical oracle separation.
- **Fefferman & Kimmel ’15**: In-place permutation oracle. Problem corresponds to set size estimation.
- **Natarajan & Nirkhe ’22**: Distribution testing oracle. Problem corresponds to size estimation of an expander graph.
- **Li, Liu, Pelecanos, Yamakawa ’23**: Separation assuming only classical queries. Based on “Verifiable Quantum Advantage without Structure”.

History of the QMA versus QCMA problem

- First proposed in ’02 by **Aharanov and Naveh**.
- **Aaronson & Kuperberg ’06**: Quantum oracle separation. Each $\mathcal{O}_n = \mathrm{id} - 2\,|\psi_n\rangle\langle\psi_n|$
- **Lutomirski ’11**: Proposed the expander mixing problem as a candidate classical oracle separation.
- **Fefferman & Kimmel ’15**: In-place permutation oracle. Problem corresponds to set size estimation.
- **Natarajan & Nirkhe ’22**: Distribution testing oracle. Problem corresponds to size estimation of an expander graph.
- **Li, Liu, Pelecanos, Yamakawa ’23**: Separation assuming only classical queries. Based on “Verifiable Quantum Advantage without Structure”.
- **Ben-David & Kundu ’24**: Bounded adaptivity, based on “Verifiable Quantum Advantage without Structure”.

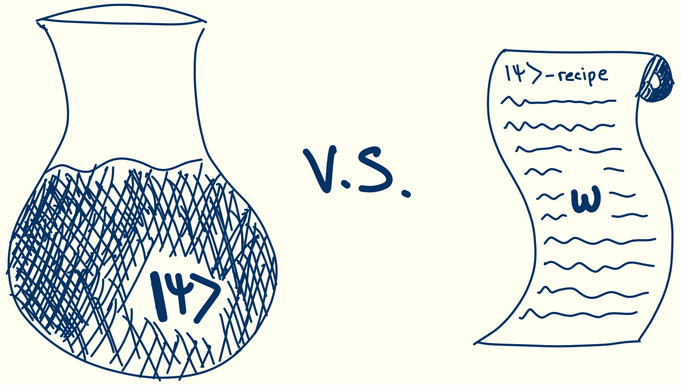
History of the QMA versus QCMA problem

- First proposed in ’02 by **Aharanov and Naveh**.
- **Aaronson & Kuperberg ’06**: Quantum oracle separation. Each $\mathcal{O}_n = \mathrm{id} - 2\,|\psi_n\rangle\langle\psi_n|$
- **Lutomirski ’11**: Proposed the expander mixing problem as a candidate classical oracle separation.
- **Fefferman & Kimmel ’15**: In-place permutation oracle. Problem corresponds to set size estimation.
- **Natarajan & Nirkhe ’22**: Distribution testing oracle. Problem corresponds to size estimation of an expander graph.
- **Li, Liu, Pelecanos, Yamakawa ’23**: Separation assuming only classical queries. Based on “Verifiable Quantum Advantage without Structure”.
- **Ben-David & Kundu ’24**: Bounded adaptivity, based on “Verifiable Quantum Advantage without Structure”.
- **Liu, Mutreja, Yuen ’25**: Aaronson-Ambainis-like conjecture implies QMA QCMA separation. Problem corresponds to size estimation of an expander graph.

Why is this problem so hard?

Why is this problem so hard?

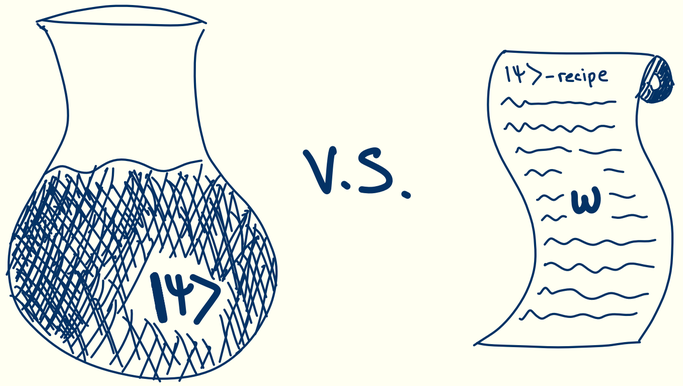
→ We have few techniques that would differentiate quantum and classical proofs!



Why is this problem so hard?

→ We have few techniques that would differentiate quantum and classical proofs!

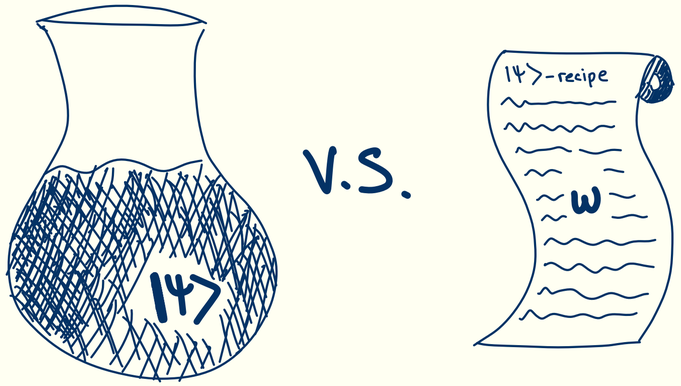
- The typical trick is to “guess the witness”, but this works for a quantum witness.



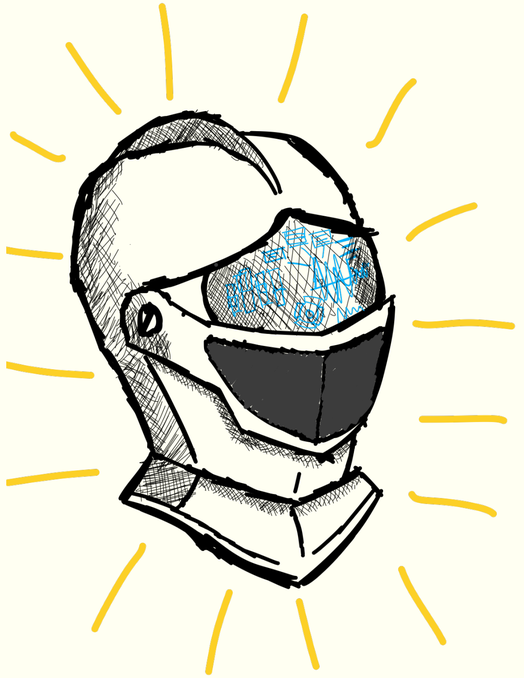
Why is this problem so hard?

→ We have few techniques that would differentiate quantum and classical proofs!

- The typical trick is to “guess the witness”, but this works for a quantum witness.



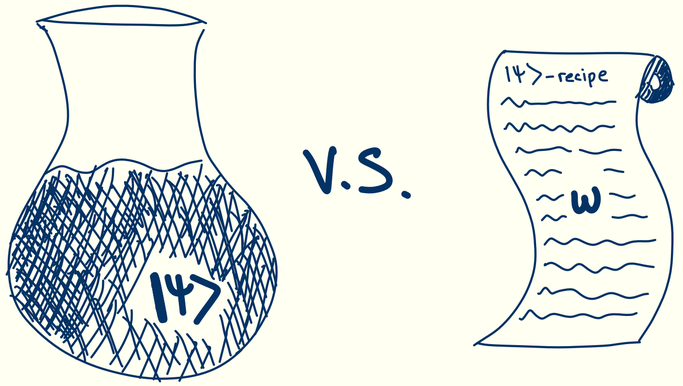
→ A quantum verifier must use their proof in an “interesting” way, can’t just measure their proof, because otherwise I could send the measurement result as a proof!



Why is this problem so hard?

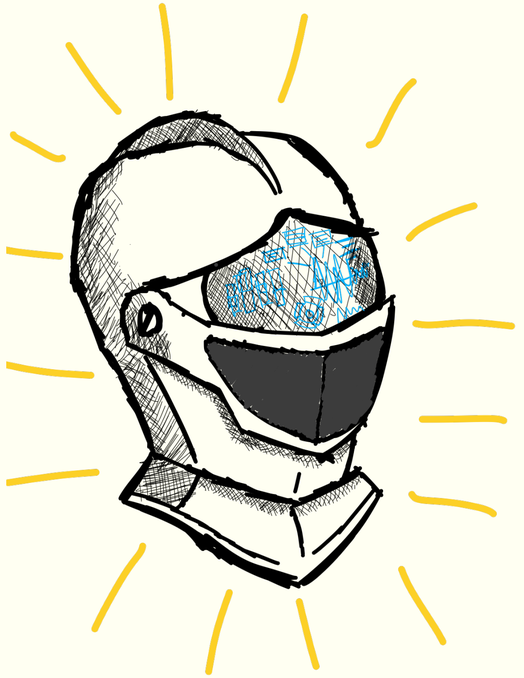
→ We have few techniques that would differentiate quantum and classical proofs!

- The typical trick is to “guess the witness”, but this works for a quantum witness.



→ A quantum verifier must use their proof in an “interesting” way, can’t just measure their proof, because otherwise I could send the measurement result as a proof!

- Quantum algorithms are really good at detecting global structure, so it’s even hard to rule out BQP algorithms for some candidate oracle separations!



A new approach for ruling out QCMA algorithms

Let’s reprove an old result of Fefferman and Kimmel, using a new technique.

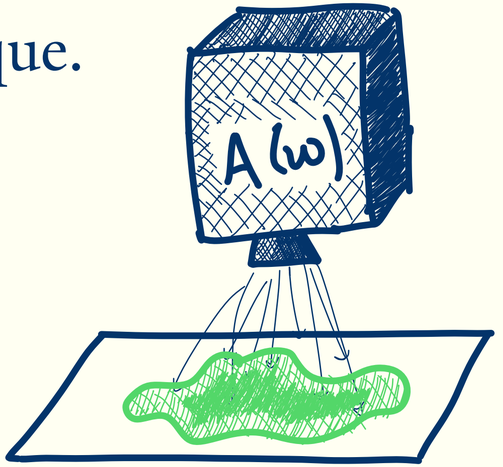
A new approach for ruling out QCMA algorithms

Let's reprove an old result of Fefferman and Kimmel, using a new technique.

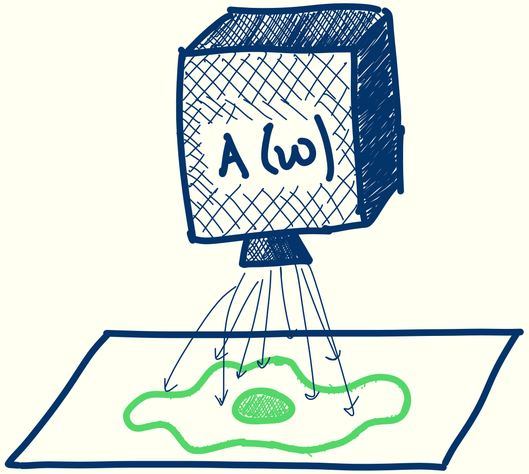
They define an oracle-input problem as follows:

Input: Oracle access to a set S

Output: Is $|S| \geq \ell$ or $|S| \leq \ell/2$, promised one of the two is the case.

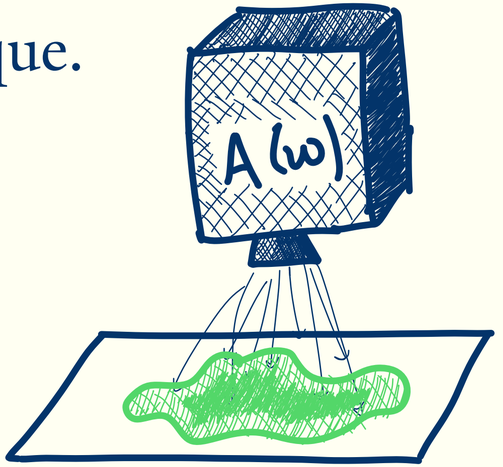


VS



A new approach for ruling out QCMA algorithms

Let's reprove an old result of Fefferman and Kimmel, using a new technique.

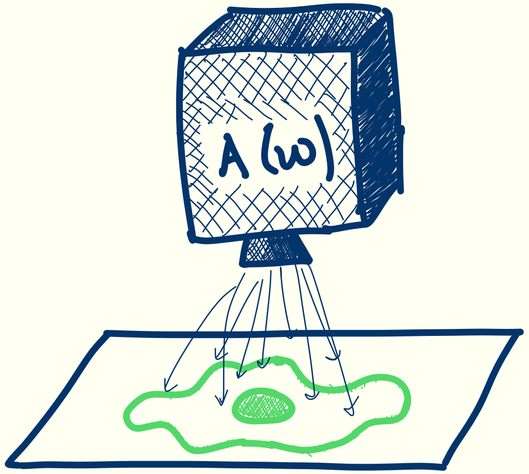


They define an oracle-input problem as follows:

Input: Oracle access to a set S

Output: Is $|S| \geq \ell$ or $|S| \leq \ell/2$, promised one of the two is the case.

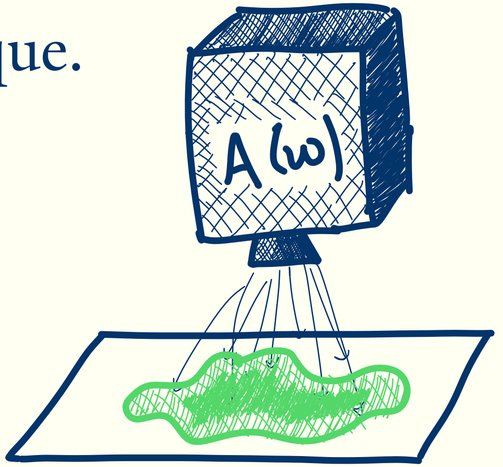
VS



- This problem is in AM by the work of Goldwasser and Sipser.

A new approach for ruling out QCMA algorithms

Let's reprove an old result of Fefferman and Kimmel, using a new technique.

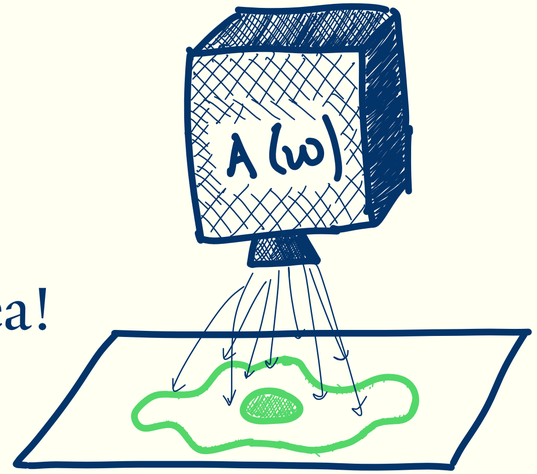


They define an oracle-input problem as follows:

Input: Oracle access to a set S

Output: Is $|S| \geq \ell$ or $|S| \leq \ell/2$, promised one of the two is the case.

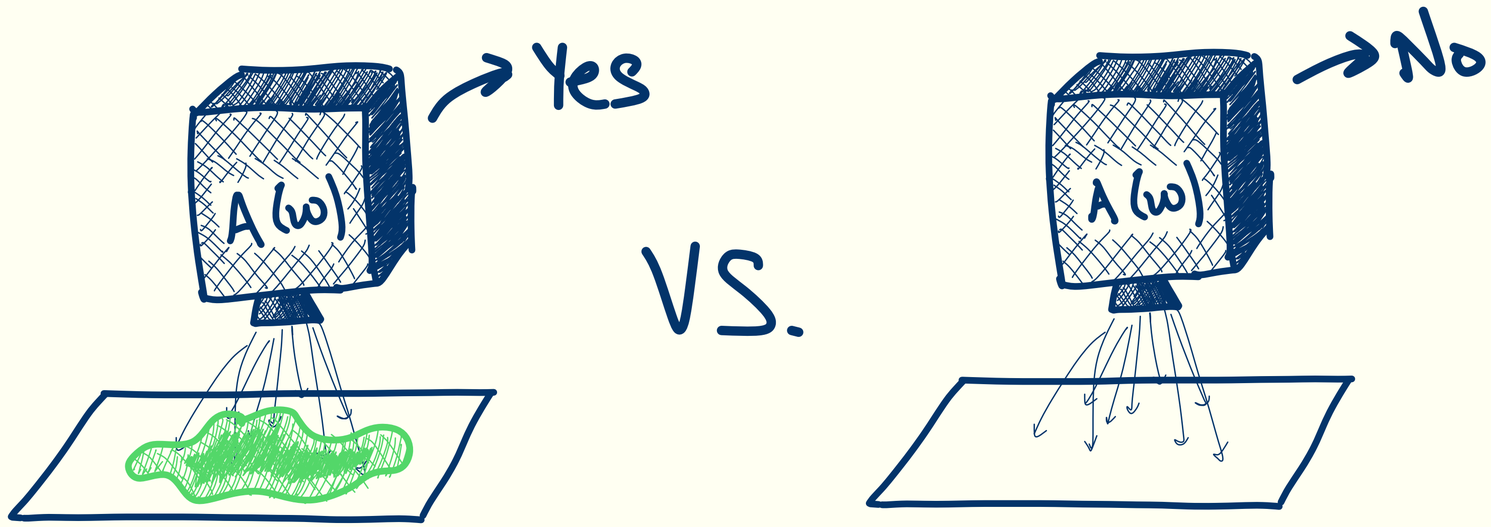
VS



- This problem is in AM by the work of Goldwasser and Sipser.
- Fefferman and Kimmel originally approached this by fixing a "good" witness, and then applying the adversary method. Let's see another idea!

A new approach for ruling out QCMA algorithms

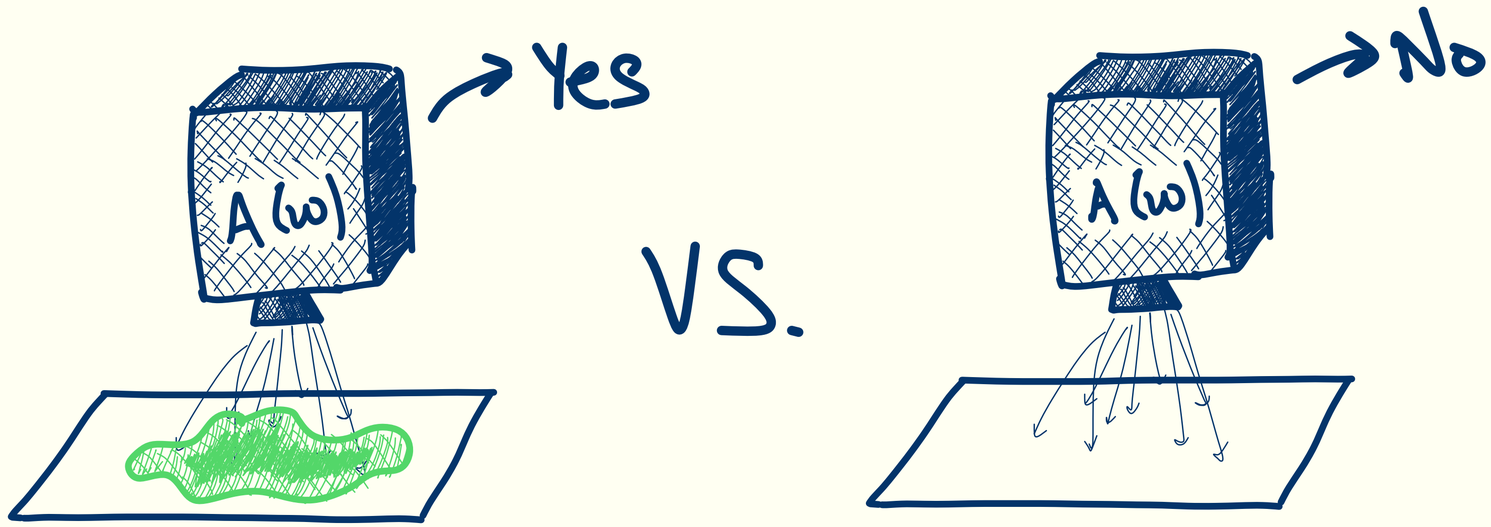
Assume for the sake of contradiction that some algorithm $A(w)$ distinguishes between the two cases. So, $A(w)$ outputs yes when given S, and no when given $\varnothing$.



A new approach for ruling out QCMA algorithms

Assume for the sake of contradiction that some algorithm $A(w)$ distinguishes between the two cases. So, $A(w)$ outputs yes when given S, and no when given $\varnothing$.

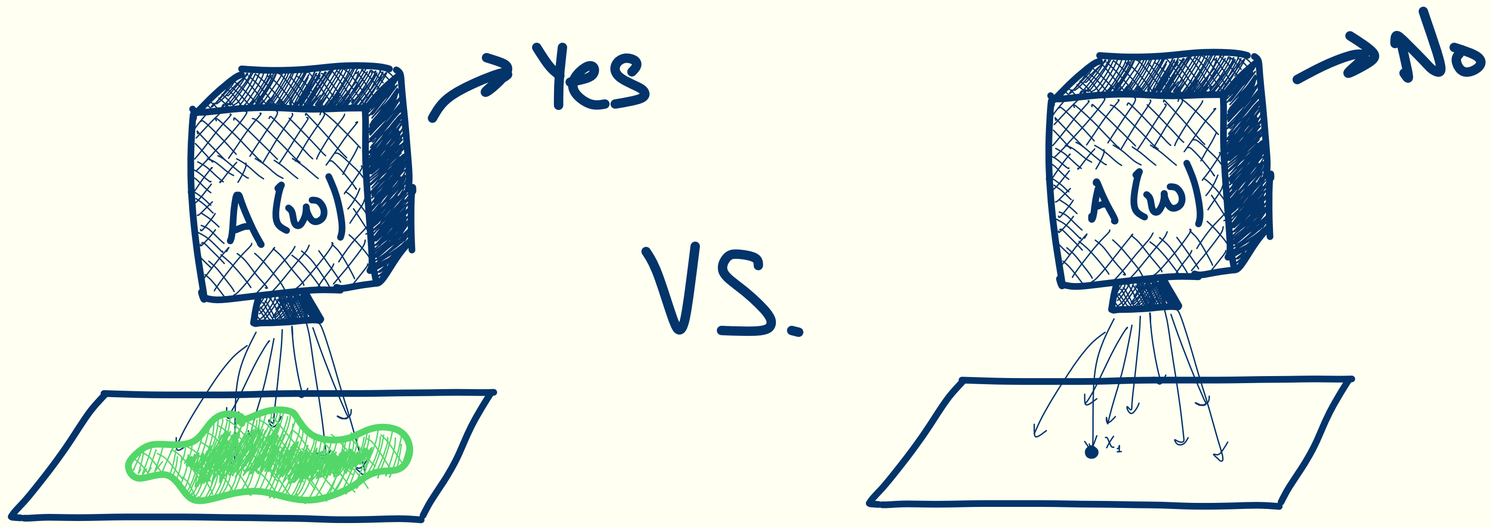
$\rightarrow$ A random query of $A^{\varnothing}(w)$ must query a point of S with at least $1/36t^2$ probability



A new approach for ruling out QCMA algorithms

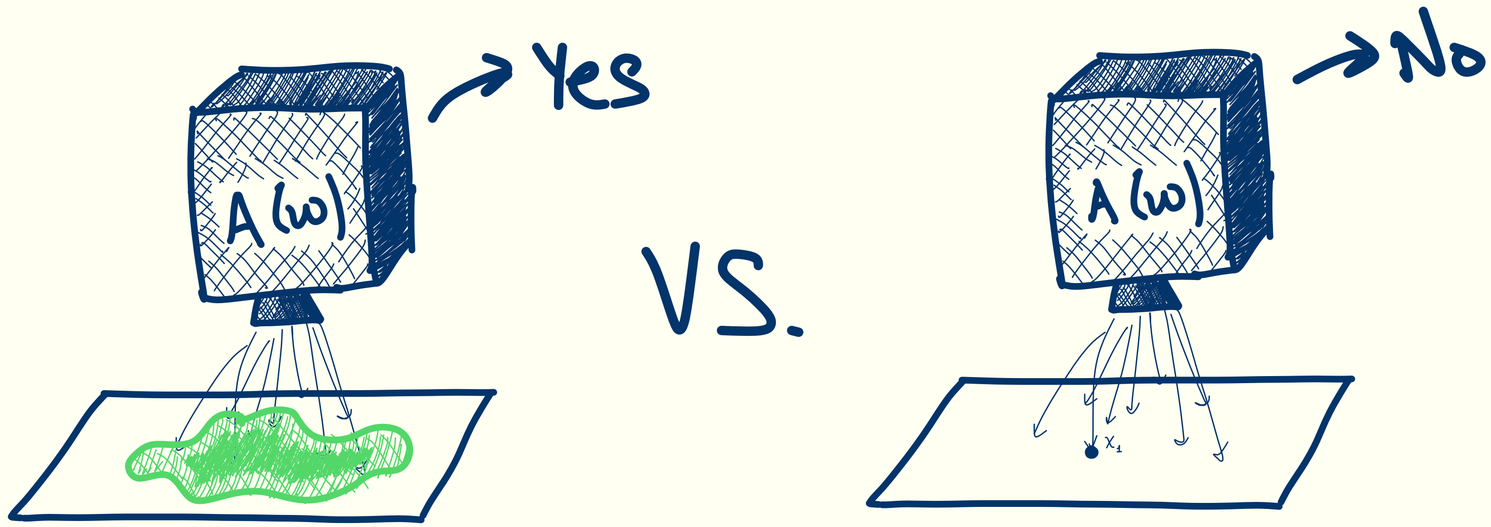
Assume for the sake of contradiction that some algorithm $A(w)$ distinguishes between the two cases. So, $A(w)$ outputs yes when given S, and no when given $\varnothing$.

$\rightarrow$ A random query of $A^{\varnothing}(w)$ must query a point of S with at least $1/36t^2$ probability



A new approach for ruling out QCMA algorithms

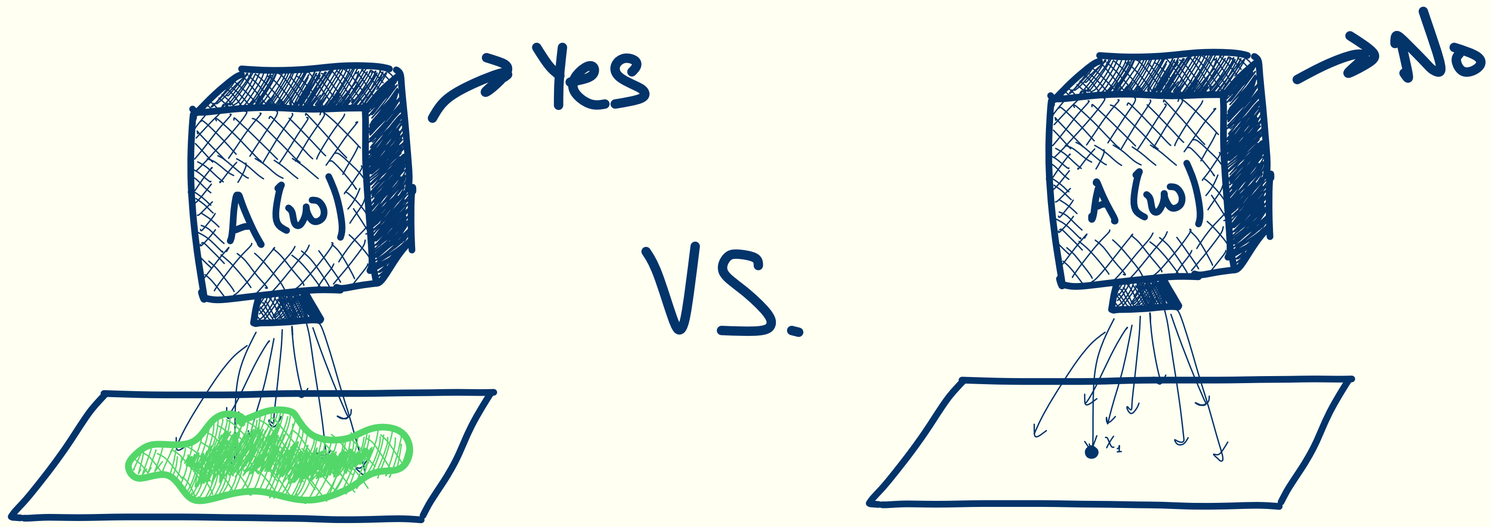
Assume for the sake of contradiction that some algorithm $A(w)$ distinguishes between the two cases. So, $A(w)$ outputs yes when given S, and no when given $\{x_1\}$.



A new approach for ruling out QCMA algorithms

Assume for the sake of contradiction that some algorithm $A(w)$ distinguishes between the two cases. So, $A(w)$ outputs yes when given S, and no when given $\{x_1\}$.

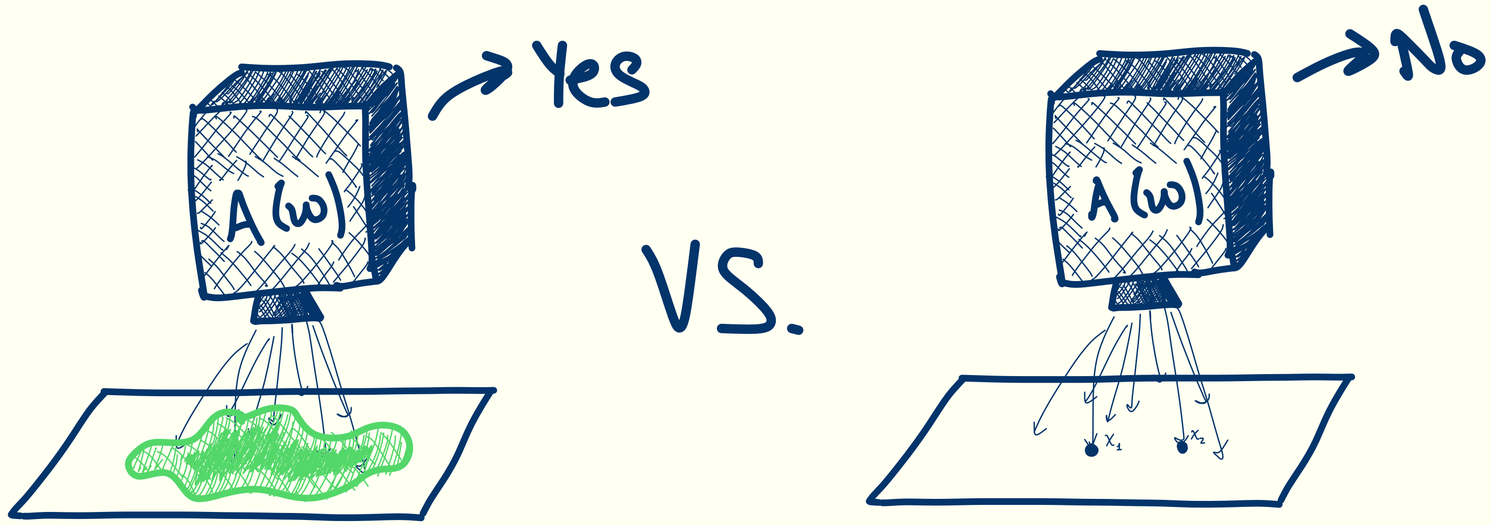
→ A random query of $A^{\{x_1\}}(w)$ must query a point of $S \backslash \{x_1\}$ with $1/36t^2$ probability



A new approach for ruling out QCMA algorithms

Assume for the sake of contradiction that some algorithm $A(w)$ distinguishes between the two cases. So, $A(w)$ outputs yes when given S, and no when given $\{x_1\}$.

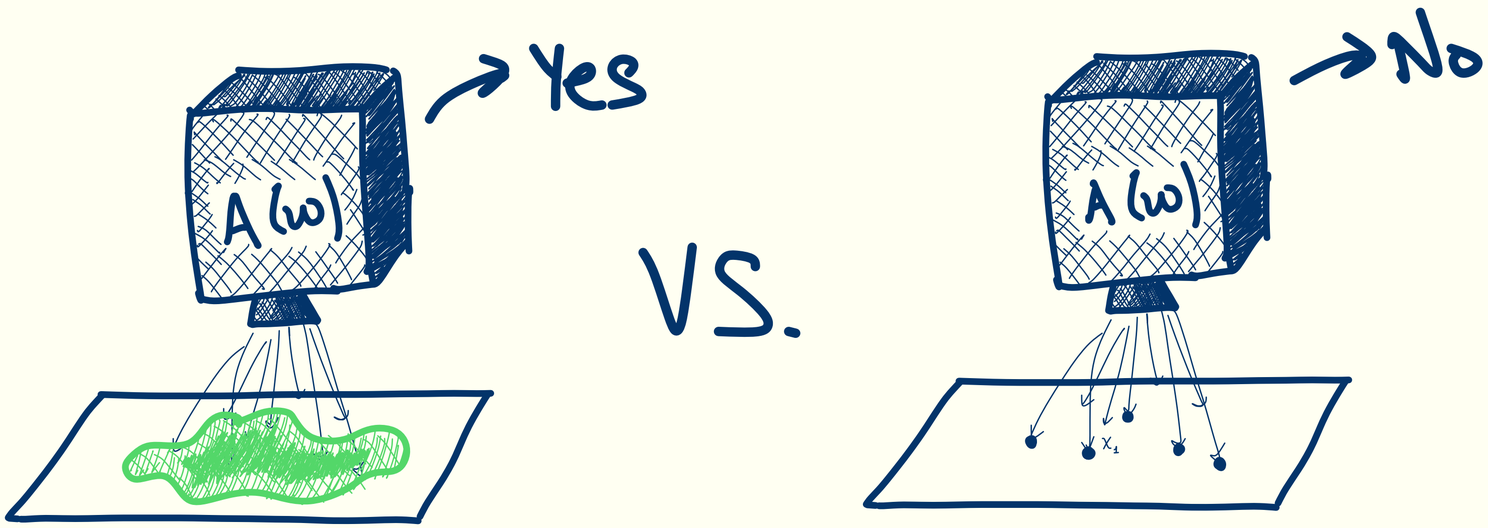
→ A random query of $A^{\{x_1\}}(w)$ must query a point of $S \backslash \{x_1\}$ with $1/36t^2$ probability



A new approach for ruling out QCMA algorithms

We can keep repeating this to get $v \leq \ell/2$ points from S, with probability at least $(1/36t^2)^v$. If we additionally guess the witness, we can sample v points with probability

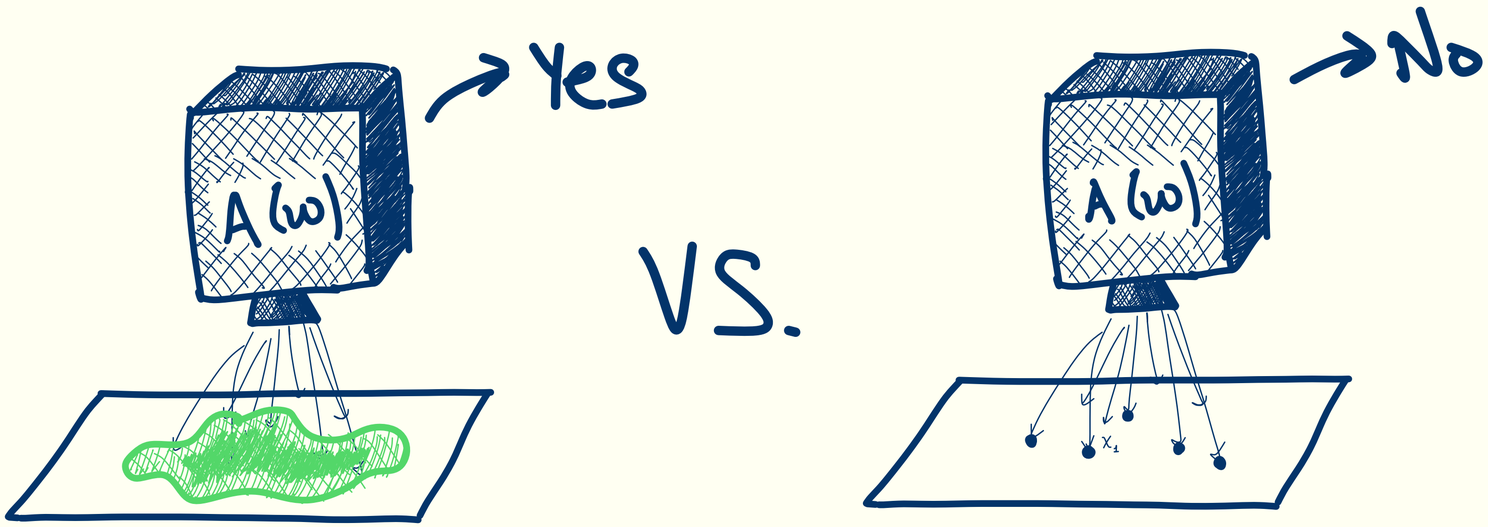
$$\geq 2^{-|w|} \left(1/36t^2\right)^v$$



A new approach for ruling out QCMA algorithms

We can keep repeating this to get $v \leq \ell/2$ points from S, with probability at least $(1/36t^2)^v$. If we additionally guess the witness, we can sample v points with probability

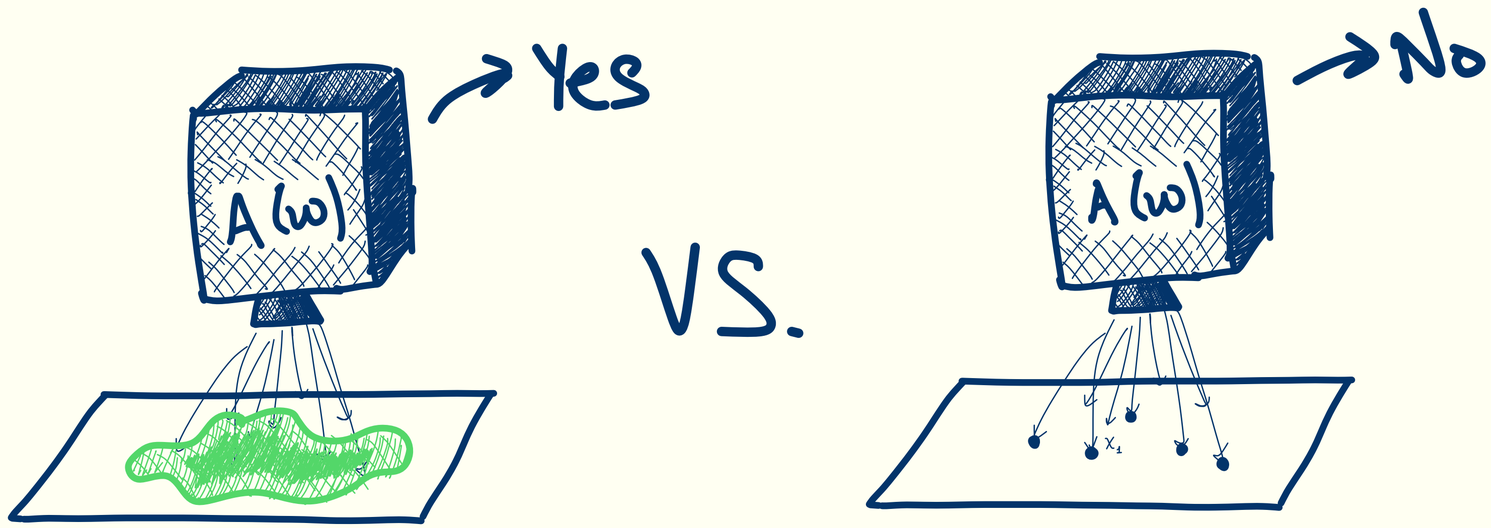
$$\geq 2^{-|w|} \left(1/36t^2\right)^v$$



Our sampler doesn't make any queries to S!

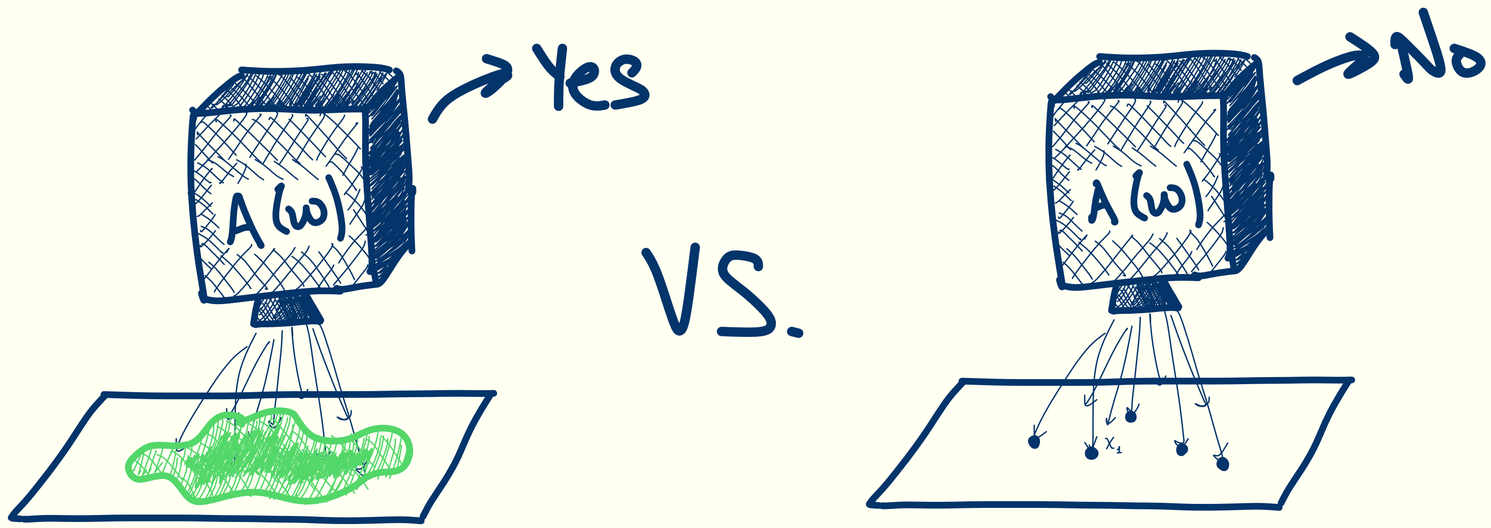
A new approach for ruling out QCMA algorithms

If there is a QCMA verifier for set size verification, we get a sampler, making no queries to S, that samples points from any set S with probability $\geq 2^{-|w|}\left(1/36t^2\right)^v$



A new approach for ruling out QCMA algorithms

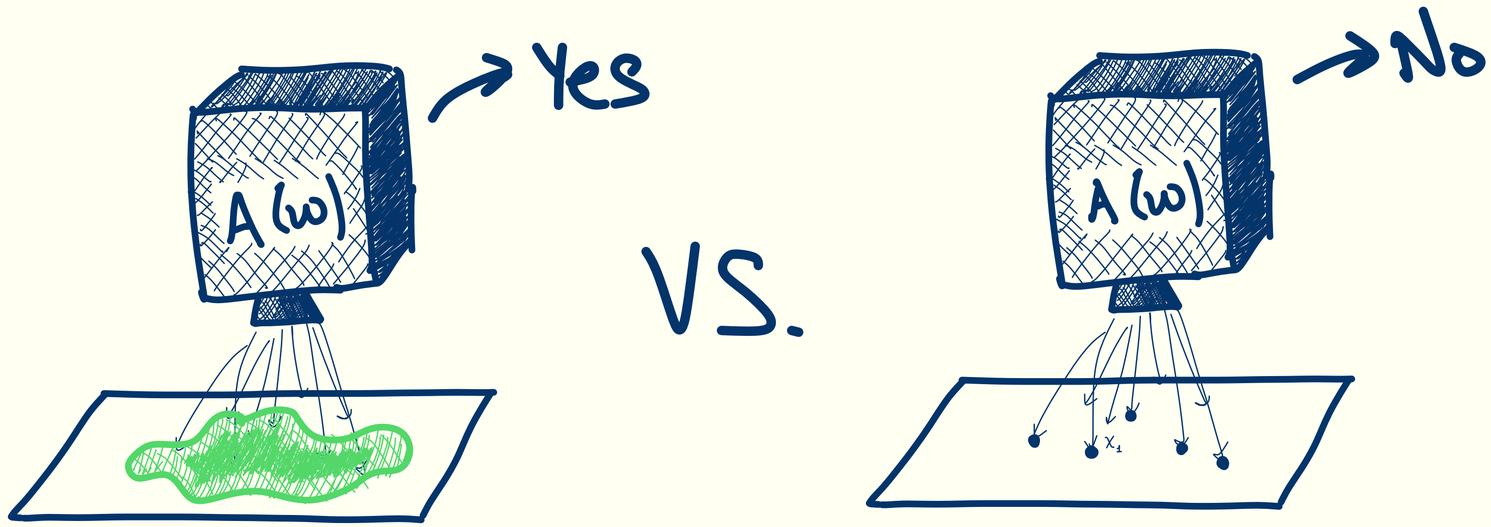
If there is a QCMA verifier for set size verification, we get a sampler, making no queries to S, that samples points from any set S with probability $\geq 2^{-|w|}\left(1/36t^2\right)^v$



But, any algorithm outputting v points can only be right with probability at most $(\ell/2^n)^v$.
$\rightarrow$ When $v \sim O(|w|/n)$, we get a contradiction!

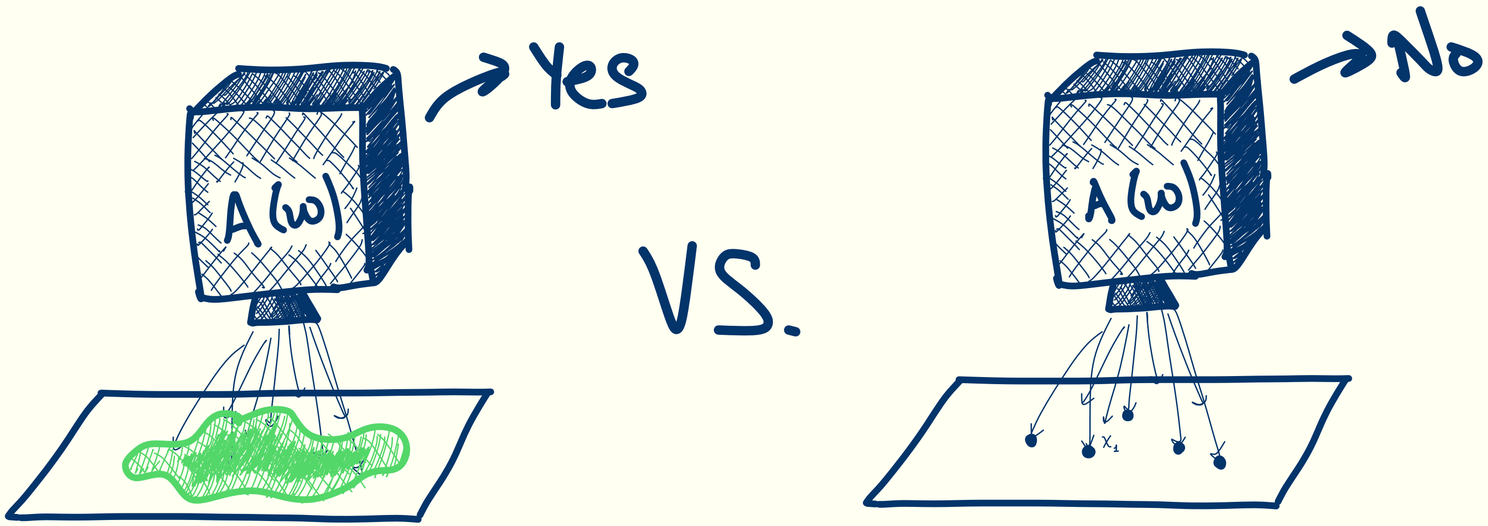
A new approach for ruling out QCMA algorithms

An important thing to note about this proof: It depends on the fact that *w* is "clonable", or that we can re-use it between different rounds. This would not necessarily work for a QMA witness!



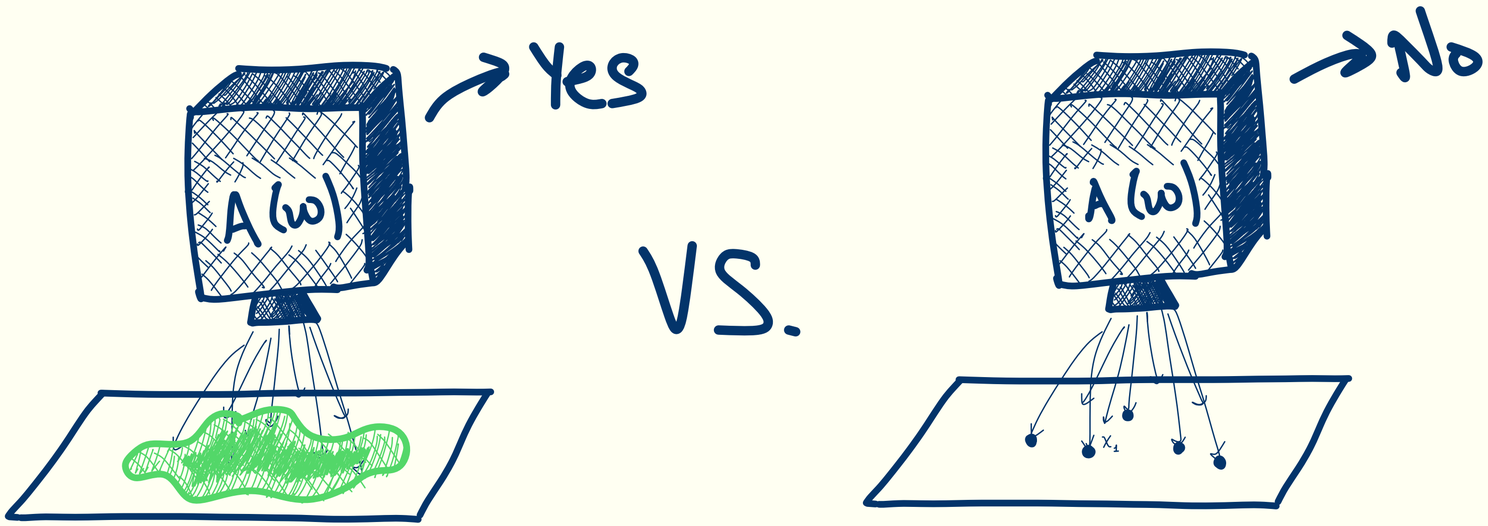
From AM to QMA

We just saw that set size verification is outside of QCMA.



From AM to QMA

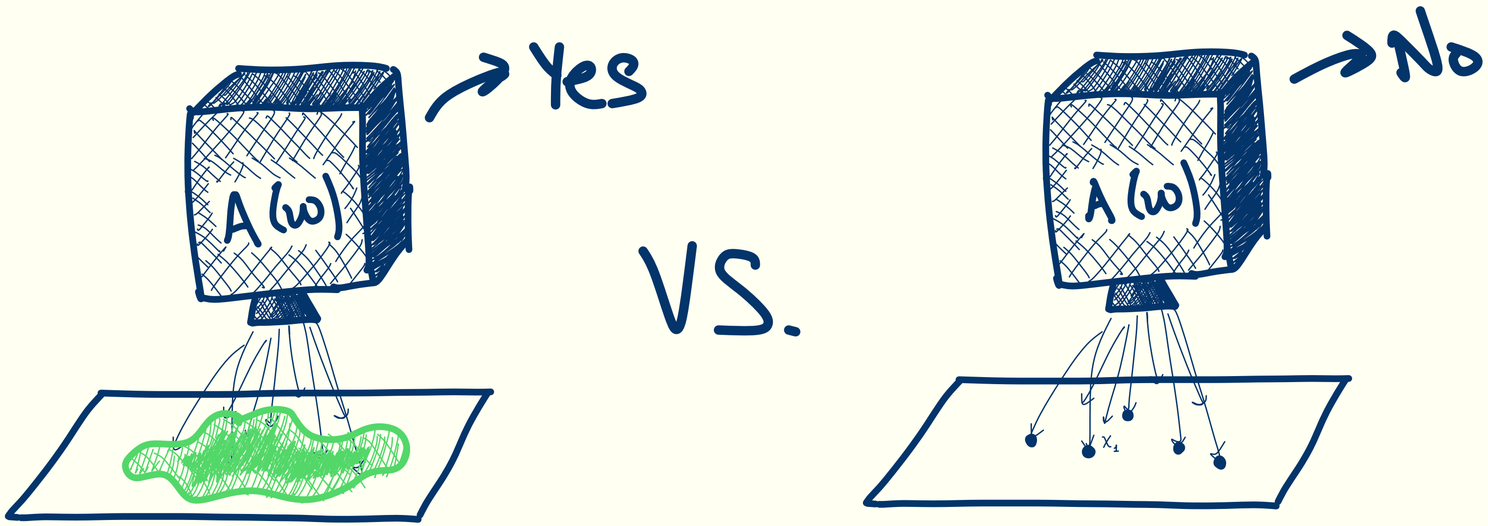
We just saw that set size verification is outside of QCMA.
Unfortunately, it's also not in QMA!



From AM to QMA

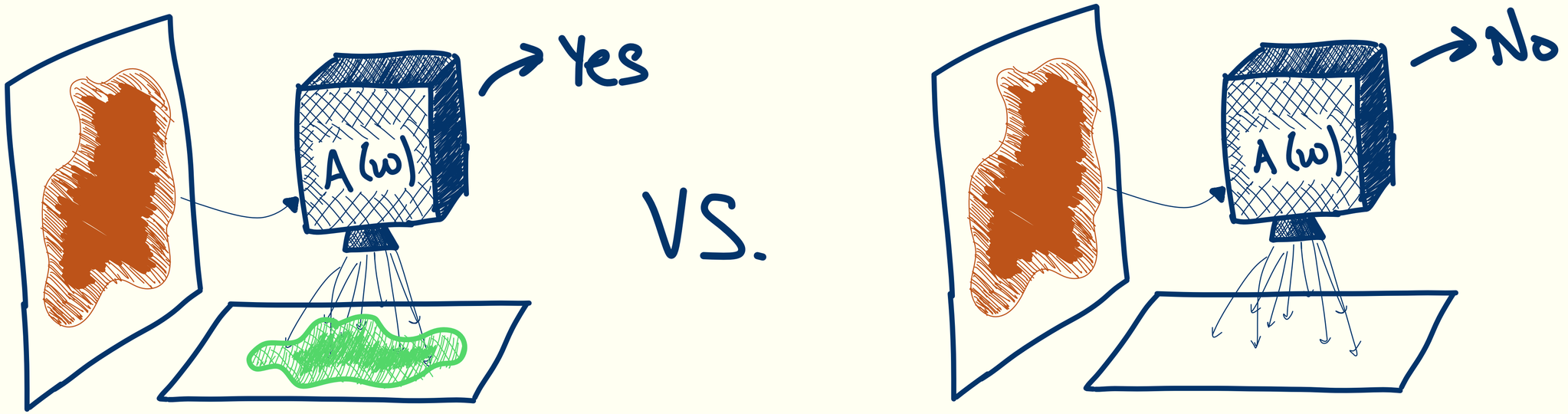
We just saw that set size verification is outside of QCMA.
Unfortunately, it's also not in QMA!

Let's look at one way of putting the problem back in QMA.



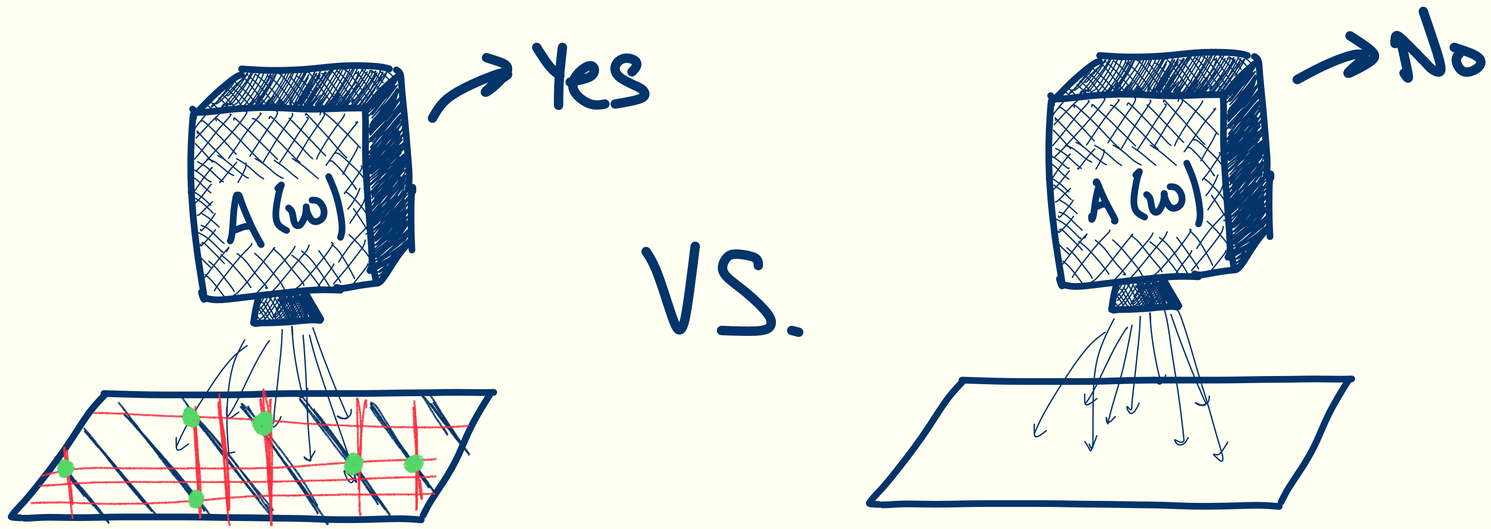
Idea 1: A second oracle [Zhandry'24, BHNZ'25]

Maybe we can add a second oracle that encodes information about the "Fourier transform" of the set S. This results in a problem we call spectral Forrelation.



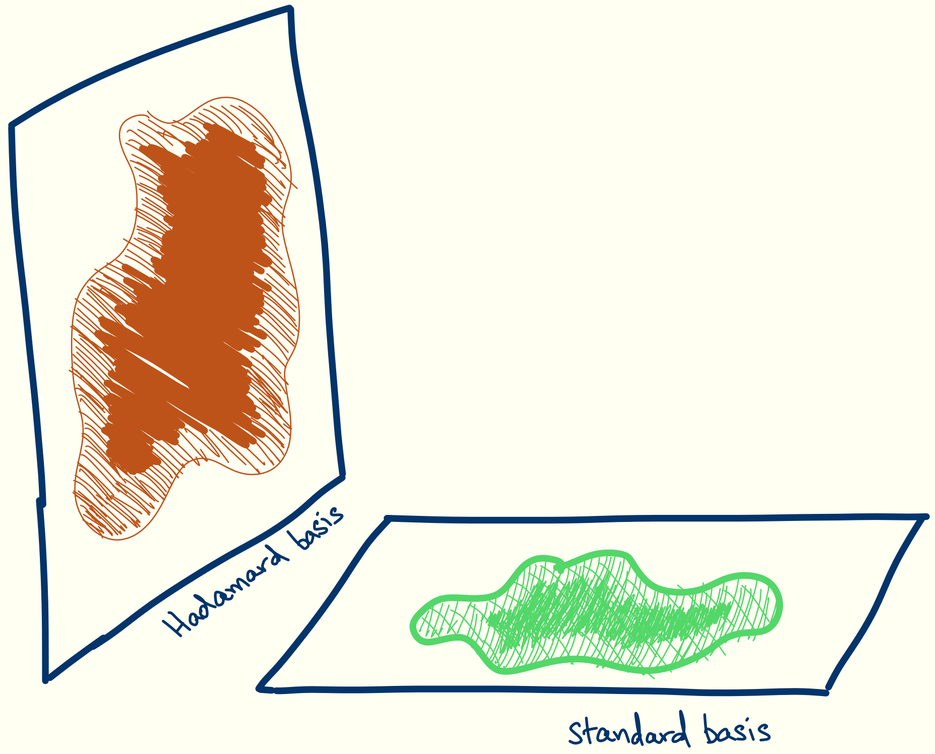
Idea 2: Code intersection [LLPY'23, BK'24, BHV'25]

Another approach will be to add structure to the sparse set S so that a quantum proof can encode many points in the set, but a classical proof can't. This results in a problem called code intersection.



The spectral Forrelation problem

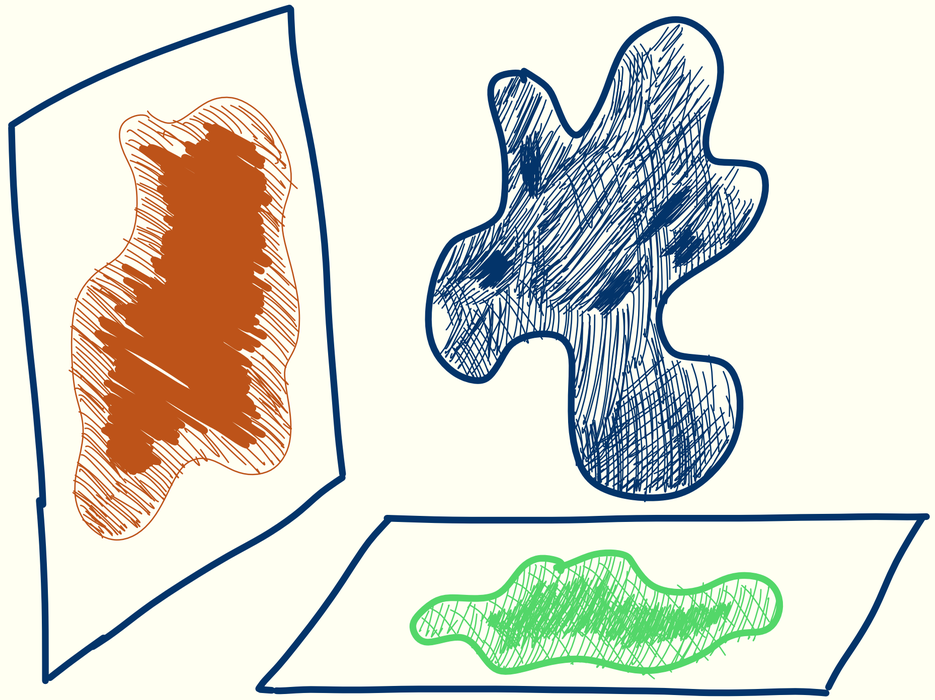
The spectral Forrelation problem is a problem about pairs of sets (S, U), which we treat as oracles through the set membership functions.



The spectral Forrelation problem

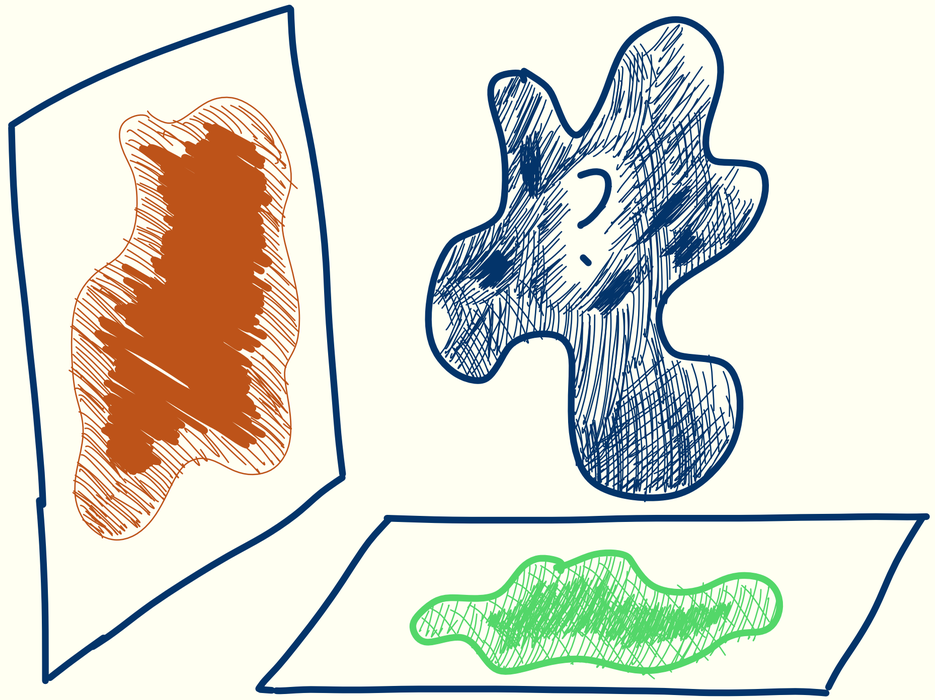
We say that two sets (S, U) are α-spectrally Forrelated if there is a state $|\psi\rangle$ such that

$$\|\Pi_U \cdot H^{\otimes n} \cdot \Pi_S |\psi\rangle\|^2 \geq \alpha$$



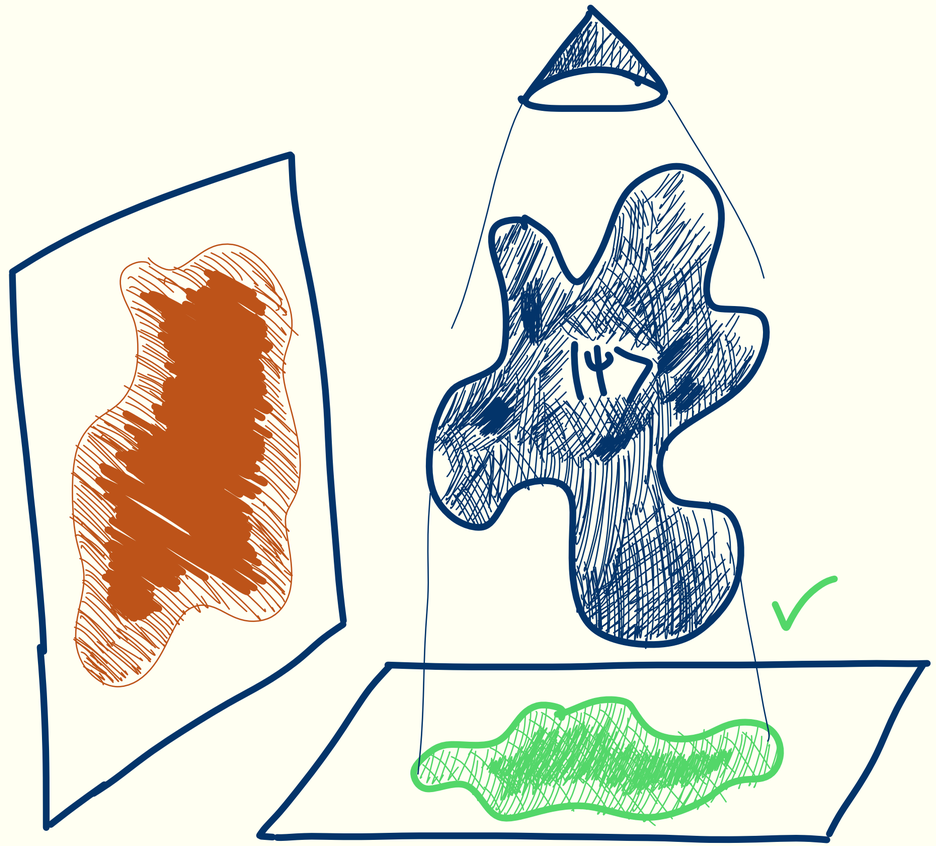
The spectral Forrelation problem

Given oracle access to two sets (S, U) (set membership functions), determine if there is a state $|\psi\rangle$ such that $\|\Pi_U \cdot H^{\otimes n} \cdot \Pi_S |\psi\rangle\|^2$ is large ($\geq 59/100$) or small ($\leq 57/100$), promised that one of the two is the case.



Spectral Forrelation is in QMA

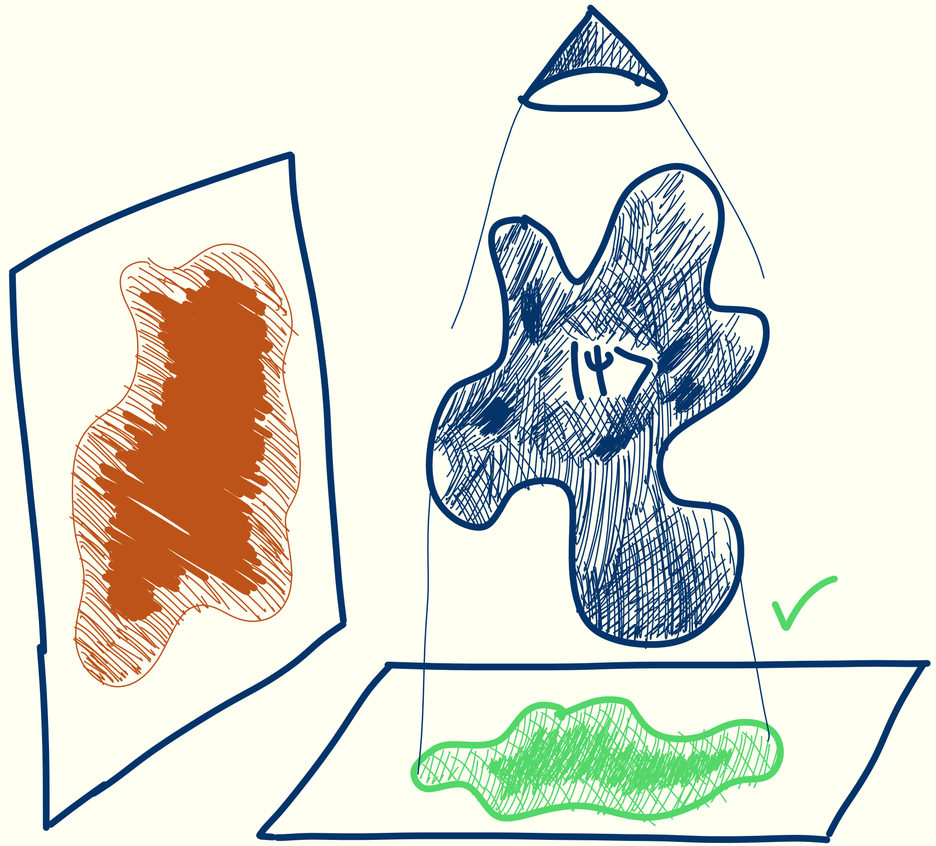
Given a copy of a state $|\psi\rangle$:



Spectral Forrelation is in QMA

Given a copy of a state $|\psi\rangle$:

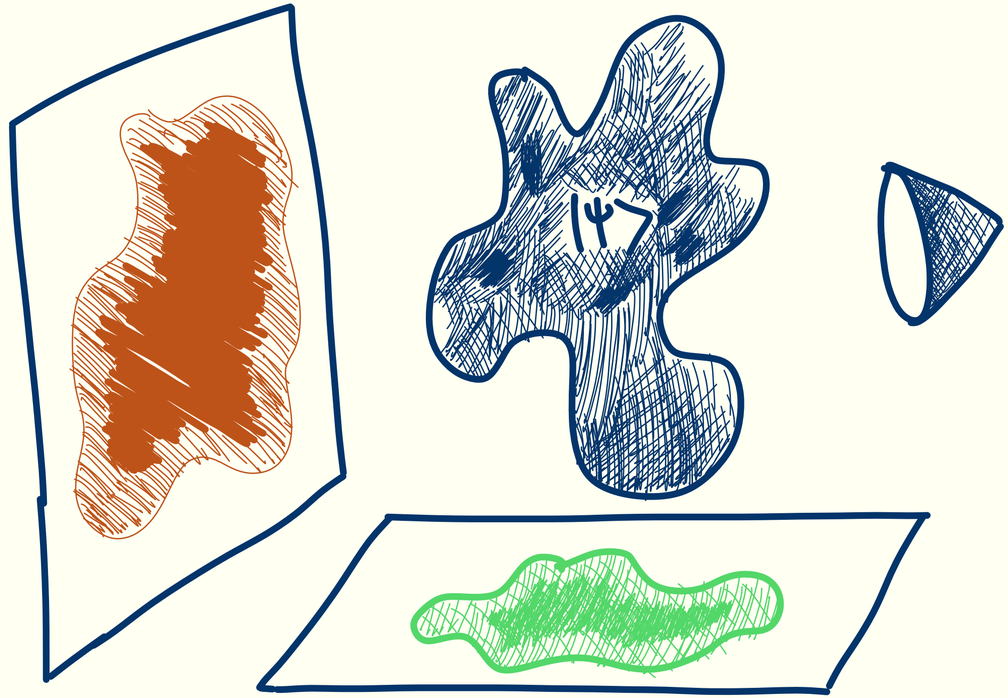
- Use S oracle to measure the POVM $\{\Pi_S, \mathrm{id} - \Pi_S\}$, reject if the outcome is $\mathrm{id} - \Pi_S$.



Spectral Forrelation is in QMA

Given a copy of a state $|\psi\rangle$:

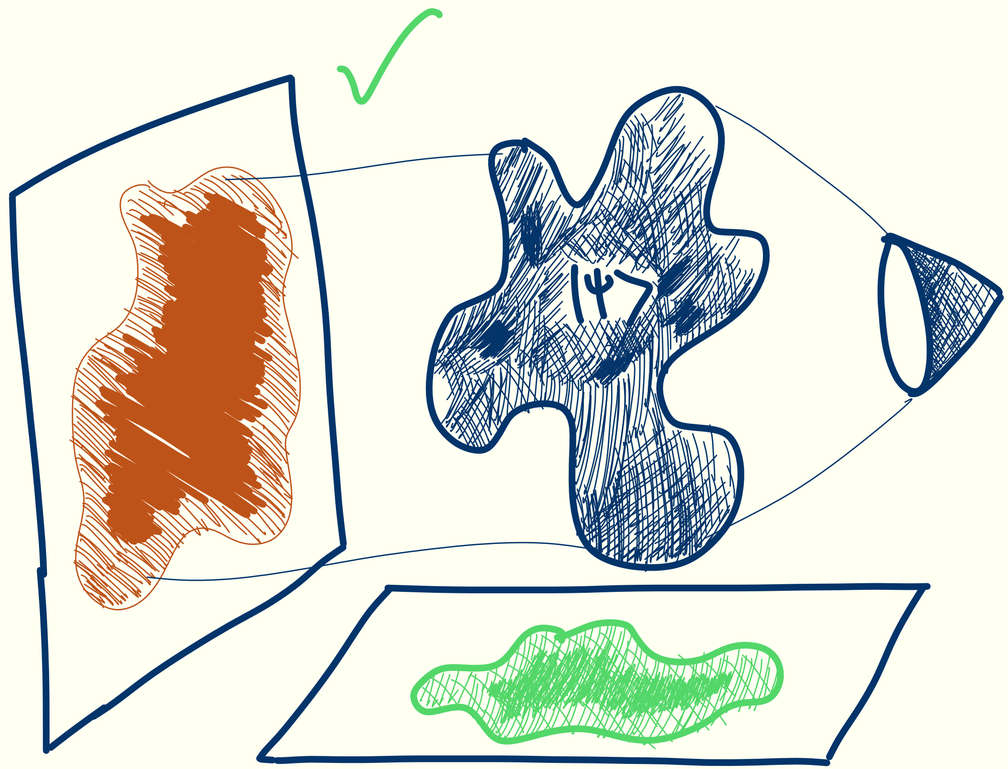
- Use S oracle to measure the POVM $\{\Pi_S, \mathrm{id} - \Pi_S\}$, reject if the outcome is $\mathrm{id} - \Pi_S$.
- Apply $H^{\otimes n}$ to the resulting state.



Spectral Forrelation is in QMA

Given a copy of a state $|\psi\rangle$:

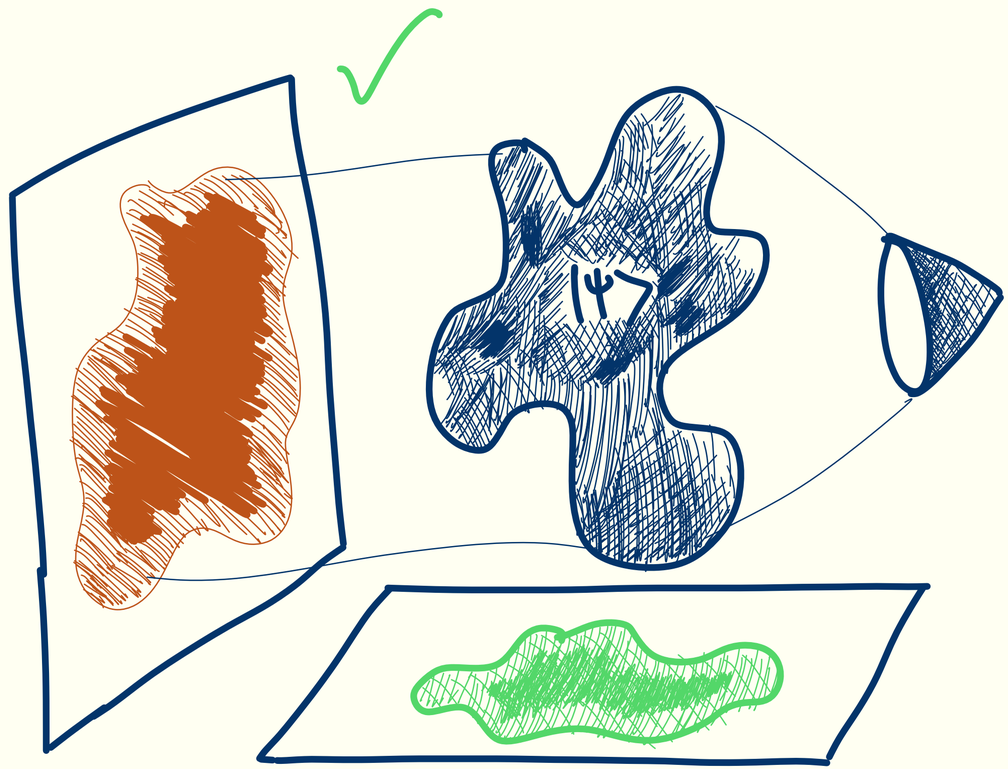
- Use S oracle to measure the POVM $\{\Pi_S, \mathrm{id} - \Pi_S\}$, reject if the outcome is $\mathrm{id} - \Pi_S$.
- Apply $H^{\otimes n}$ to the resulting state.
- Use U oracle to measure the POVM $\{\Pi_U, \mathrm{id} - \Pi_U\}$, reject it the outcome is $\mathrm{id} - \Pi_U$.
- Accept.



Spectral Forrelation is in QMA

Given a copy of a state $|\psi\rangle$:

- Use S oracle to measure the POVM $\{\Pi_S, \mathrm{id} - \Pi_S\}$, reject if the outcome is $\mathrm{id} - \Pi_S$.
- Apply $H^{\otimes n}$ to the resulting state.
- Use U oracle to measure the POVM $\{\Pi_U, \mathrm{id} - \Pi_U\}$, reject it the outcome is $\mathrm{id} - \Pi_U$.
- Accept.

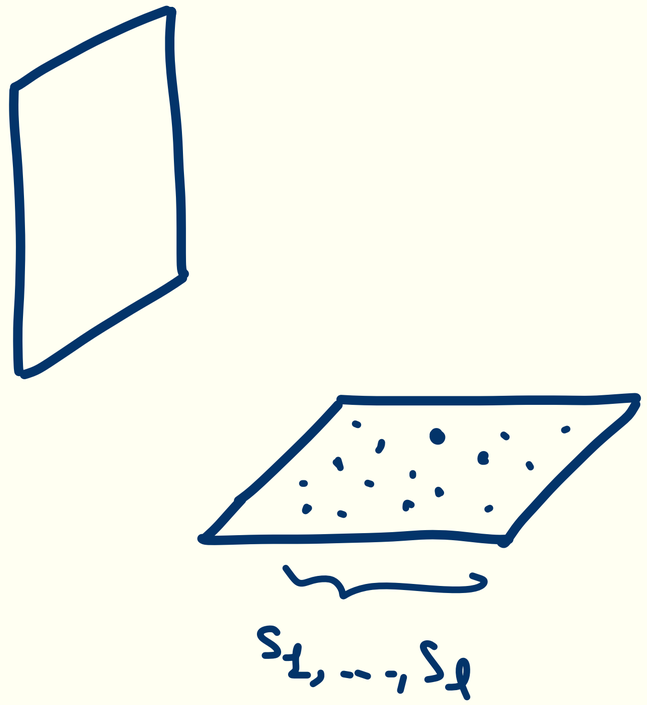


This verifier accepts with probability:
$\|\Pi_U \cdot H^{\otimes n} \cdot \Pi_S |\psi\rangle\|^2$.

Marriot-Watrous amplification can bring this to the standard 2/3 or 1/3.

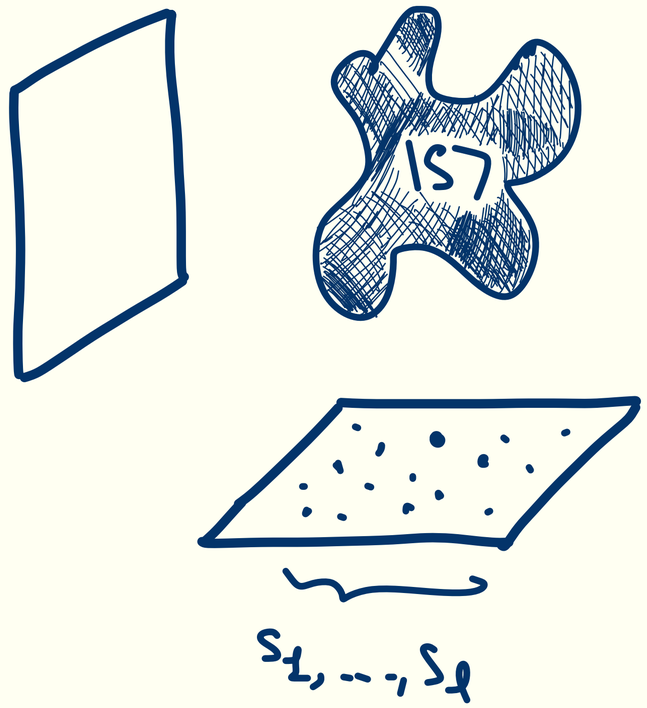
A distribution over spectral Forrelation yes instances

- We will first sample $\ell = 2^{n/10}$ many random elements $s_1, \ldots, s_\ell$.



A distribution over spectral Forrelation yes instances

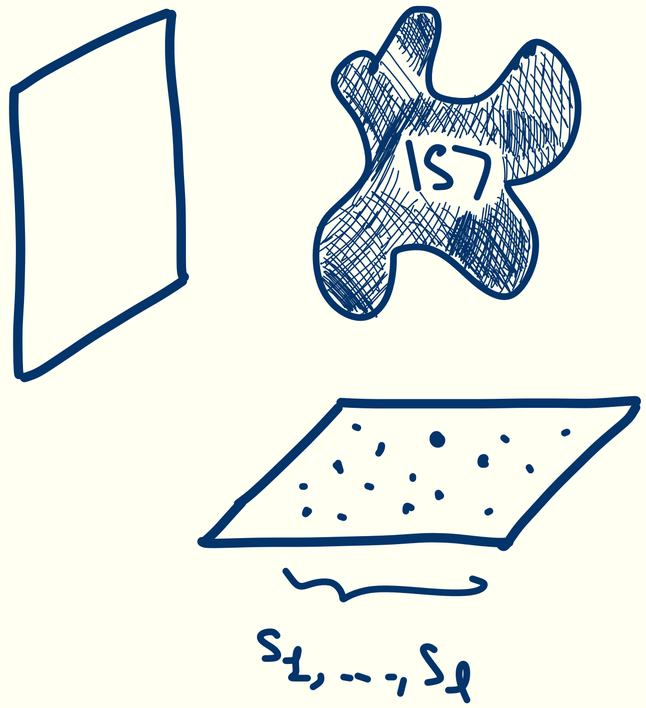
- We will first sample $\ell = 2^{n/10}$ many random elements $s_1, \ldots, s_\ell$. Let $|S\rangle$ be the uniform superposition over the points.



A distribution over spectral Forrelation yes instances

- We will first sample $\ell = 2^{n/10}$ many random elements $s_1, \ldots, s_\ell$. Let $|S\rangle$ be the uniform superposition over the points.
- We take U to be the heavy points of $H^{\otimes n}|S\rangle$, the Hadamard transform of $|S\rangle$:

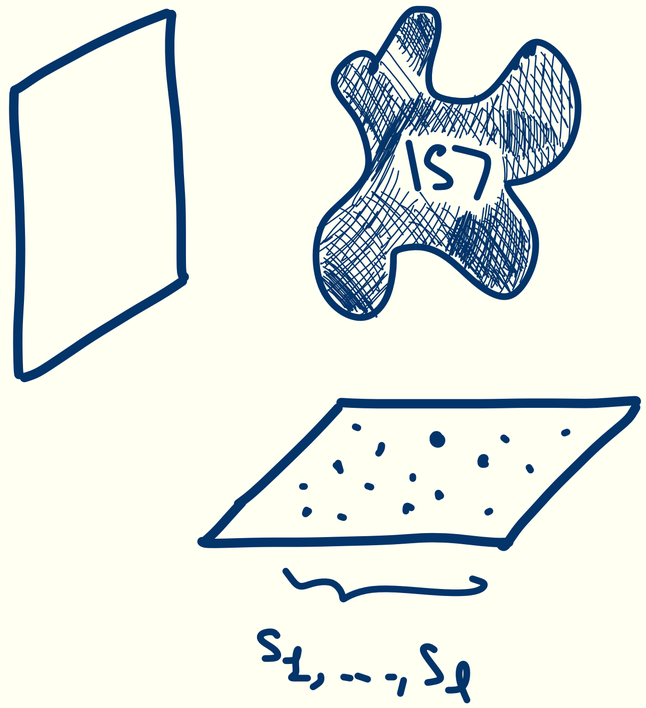
$$\Pr[y \in U] = 1 - \frac{1}{2}\exp\left(-\frac{1}{10}2^n \left|\langle y|H^{\otimes n}|S\rangle\right|^2\right)$$



A distribution over spectral Forrelation yes instances

- We will first sample $\ell = 2^{n/10}$ many random elements $s_1, \ldots, s_\ell$. Let $|S\rangle$ be the uniform superposition over the points.
- We take U to be the heavy points of $H^{\otimes n}|S\rangle$, the Hadamard transform of $|S\rangle$:

$$\Pr[y \in U] = 1 - \frac{1}{2}\exp\left(-\frac{1}{10}2^n \left|\langle y|H^{\otimes n}|S\rangle\right|^2\right)$$
$$= 1 - \frac{1}{2}\exp\left(-\frac{1}{10}\gamma_y^{(S)}\right)$$

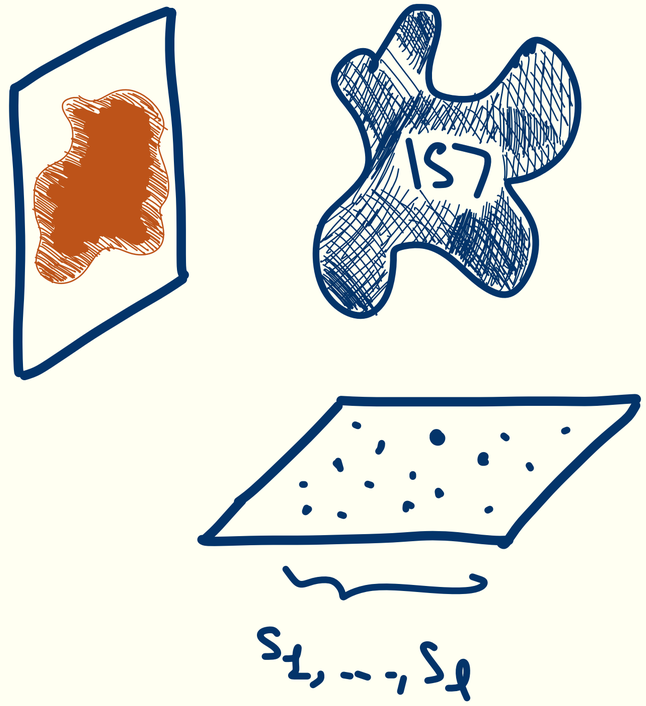


A distribution over spectral Forrelation yes instances

- We will first sample $\ell = 2^{n/10}$ many random elements $s_1, \ldots, s_\ell$. Let $|S\rangle$ be the uniform superposition over the points.
- We take U to be the heavy points of $H^{\otimes n}|S\rangle$, the Hadamard transform of $|S\rangle$:

$$\Pr[y \in U] = 1 - \frac{1}{2}\exp\left(-\frac{1}{10}2^n \left|\langle y|H^{\otimes n}|S\rangle\right|^2\right)$$

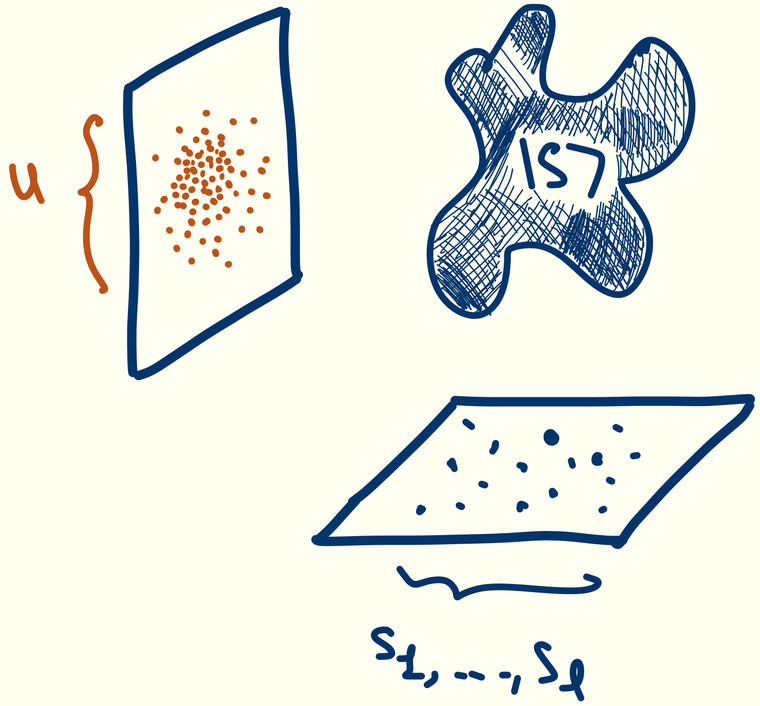
$$= 1 - \frac{1}{2}\exp\left(-\frac{1}{10}\gamma_y^{(S)}\right)$$



A distribution over spectral Forrelation yes instances

- We will first sample $\ell = 2^{n/10}$ many random elements $s_1, \ldots, s_\ell$. Let $|S\rangle$ be the uniform superposition over the points.
- We take U to be the heavy points of $H^{\otimes n}|S\rangle$, the Hadamard transform of $|S\rangle$:

$$\Pr[y \in U] = 1 - \frac{1}{2}\exp\left(-\frac{1}{10}2^n \left|\langle y|H^{\otimes n}|S\rangle\right|^2\right)$$
$$= 1 - \frac{1}{2}\exp\left(-\frac{1}{10}\gamma_y^{(S)}\right)$$



A distribution over spectral Forrelation yes instances

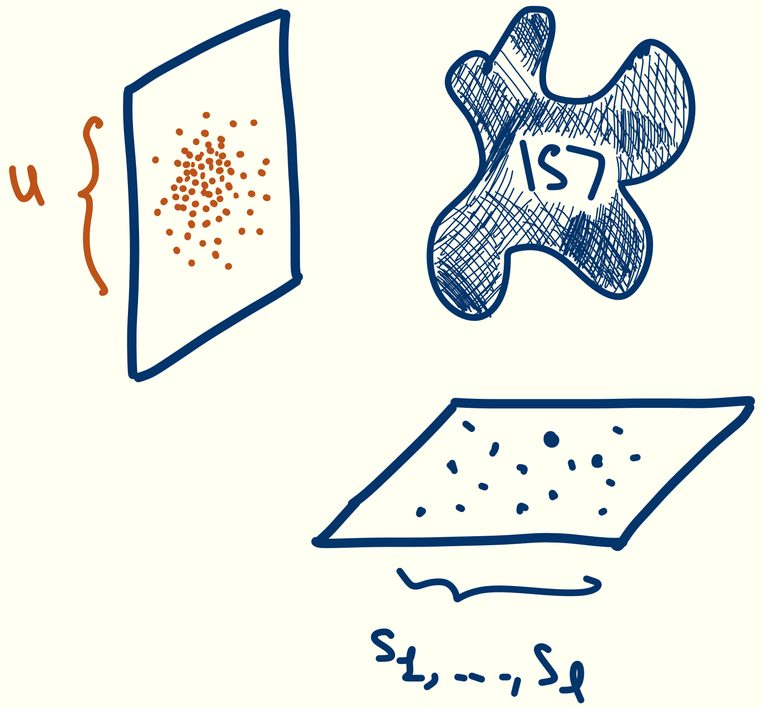
- We will first sample $\ell = 2^{n/10}$ many random elements $s_1, \ldots, s_\ell$. Let $|S\rangle$ be the uniform superposition over the points.

- We take U to be the heavy points of $H^{\otimes n}|S\rangle$, the Hadamard transform of $|S\rangle$:

$$\Pr[y \in U] = 1 - \frac{1}{2}\exp\left(-\frac{1}{10}2^n \left|\langle y|H^{\otimes n}|S\rangle\right|^2\right)$$

$$= 1 - \frac{1}{2}\exp\left(-\frac{1}{10}\gamma_y^{(S)}\right)$$

We call this distribution over oracles the Strong distribution.



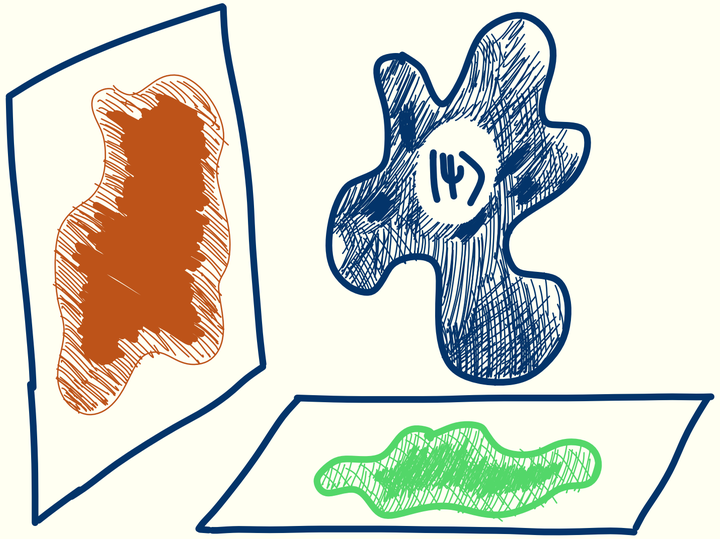
Strong yes instances

A pair (S, U) is a strong yes instance if:

Strong yes instances

A pair (S, U) is a strong yes instance if:

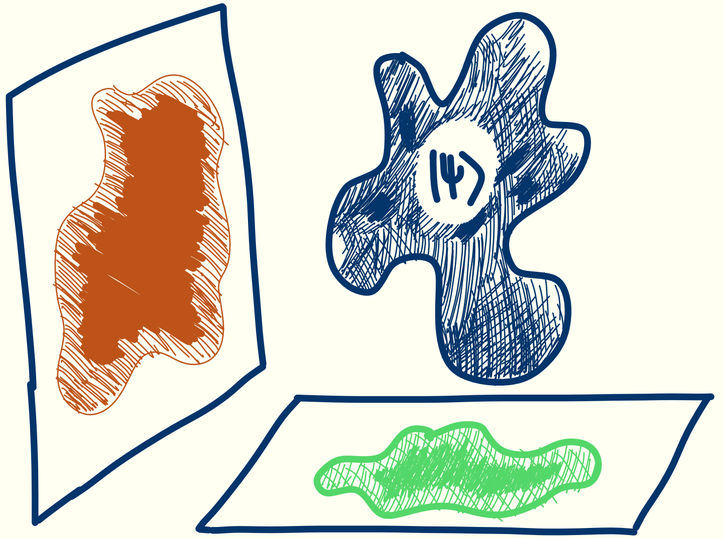
- (S, U) is a yes instance of spectral Forrelation (i.e., $\geq 59/100$ spectrally Forrelated).

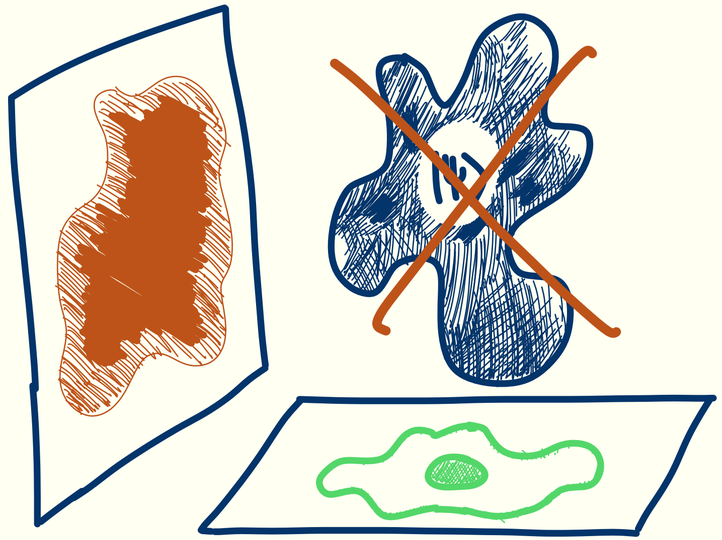


Strong yes instances

A pair (S, U) is a strong yes instance if:

- (S, U) is a yes instance of spectral Forrelation (i.e., $\geq 59/100$ spectrally Forrelated).
- For all $\Delta \subset S$ with $|\Delta| \leq \ell/100$, (Δ, U) is a no instance of spectral Forrelation (i.e., $\leq 57/100$ spectrally Forrelated).





Strong yes instances

Claim: (S, U) sampled from the Strong distribution will be a strong yes instance, except with probability $\ell 2^{n/6}$.

Strong yes instances

Claim: (S, U) sampled from the Strong distribution will be a strong yes instance, except with probability $\ell 2^{n/6}$.

Proof sketch: When we compute the expectation over U of the following operator, we roughly get something that looks like

$$\mathbb{E}_U[\Pi_S \cdot H^{\otimes n} \cdot \Pi_U \cdot H^{\otimes n} \cdot \Pi_S] \approx \frac{1}{10}|S\rangle\langle S| + \frac{1}{2}\mathrm{id}$$

Strong yes instances

Claim: (S, U) sampled from the Strong distribution will be a strong yes instance, except with probability $\ell 2^{n/6}$.

Proof sketch: When we compute the expectation over U of the following operator, we roughly get something that looks like

$$\mathbb{E}_U[\Pi_S \cdot H^{\otimes n} \cdot \Pi_U \cdot H^{\otimes n} \cdot \Pi_S] \approx \frac{1}{10}|S\rangle\langle S| + \frac{1}{2}\mathrm{id}$$

If this was really the matrix, then taking any $\Delta \times \Delta$ sub-matrix only get a small part of the mass of $|S\rangle$, making its operator norm close to 1/2.

Strong yes instances

Claim: (S, U) sampled from the Strong distribution will be a strong yes instance, except with probability $\ell 2^{n/6}$.

Proof sketch: When we compute the expectation over U of the following operator, we roughly get something that looks like

$$\mathbb{E}_U[\Pi_S \cdot H^{\otimes n} \cdot \Pi_U \cdot H^{\otimes n} \cdot \Pi_S] \approx \frac{1}{10}|S\rangle\langle S| + \frac{1}{2}\mathrm{id}$$

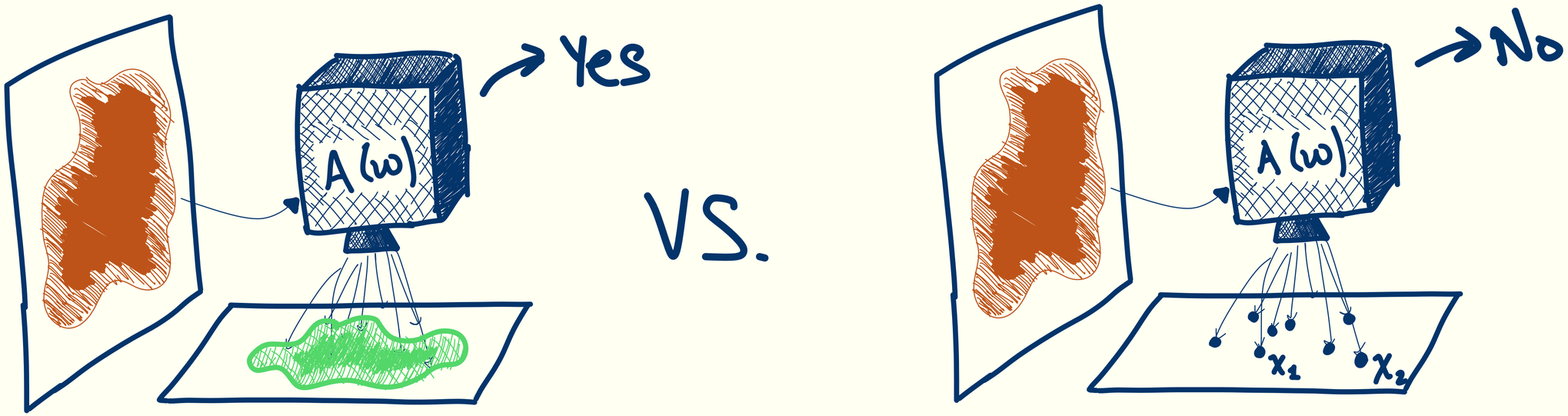
If this was really the matrix, then taking any $\Delta \times \Delta$ sub-matrix only get a small part of the mass of $|S\rangle$, making its operator norm close to $1/2$.

Using concentration bounds, we get that with very high probability, this happens.

Applying the sampler idea to spectral Forrelation

Theorem: If there exists a QCMA algorithm, making t queries to (S, U) and taking a witness of length w, then for all $0 < v < \ell/100$, there is a query algorithm making vt queries to U that outputs v distinct points from S with probability

$$\geq 2^{-w} \left(\frac{1}{36t^2} \right)^v$$



Sampler upper bounds for Strong

The Fock basis is a way to write down a multi-set, similar to how we write subsets of $\{0,1\}^n$ as 2^n bit strings. Given a multi-set with ℓ_x copies of x, we associate it with a vector of 2^n non-negative integers:

$$|\ell_0, \ldots, \ell_{2^n-1}\rangle$$

Sampler upper bounds for Strong

The Fock basis is a way to write down a multi-set, similar to how we write subsets of $\{0,1\}^n$ as 2^n bit strings. Given a multi-set with ℓ_x copies of x, we associate it with a vector of 2^n non-negative integers:

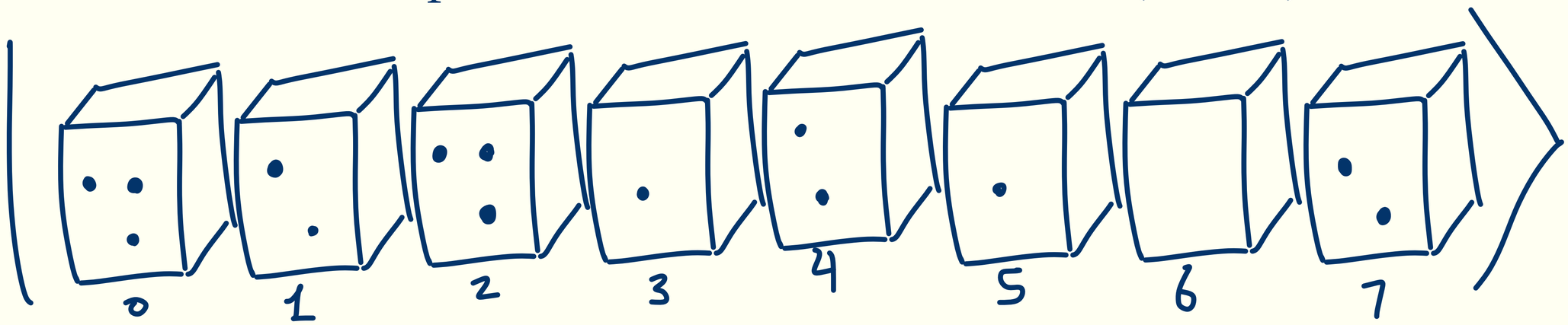
$$|\ell_0, \ldots, \ell_{2^n-1}\rangle$$

Then the uniform superposition over multi-sets is given by

$$\frac{1}{\sqrt{2^n}} \sum_{\vec{\ell}} \sqrt{\frac{\ell!}{\prod_x \ell_x!}} |\ell_0, \ldots, \ell_{2^n}\rangle .$$

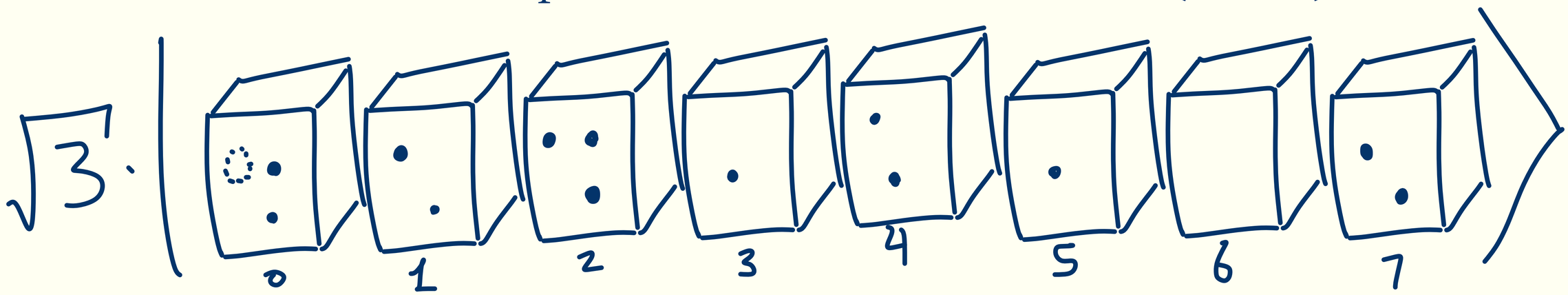
Sampler upper bounds for Strong: Bosons

Bosons are a mathematical representation of multi-sets used in physics. The "annihilation" and "creation" operators add and subtract elements (bosons).



Sampler upper bounds for Strong: Bosons

Bosons are a mathematical representation of multi-sets used in physics. The "annihilation" and "creation" operators add and subtract elements (bosons).

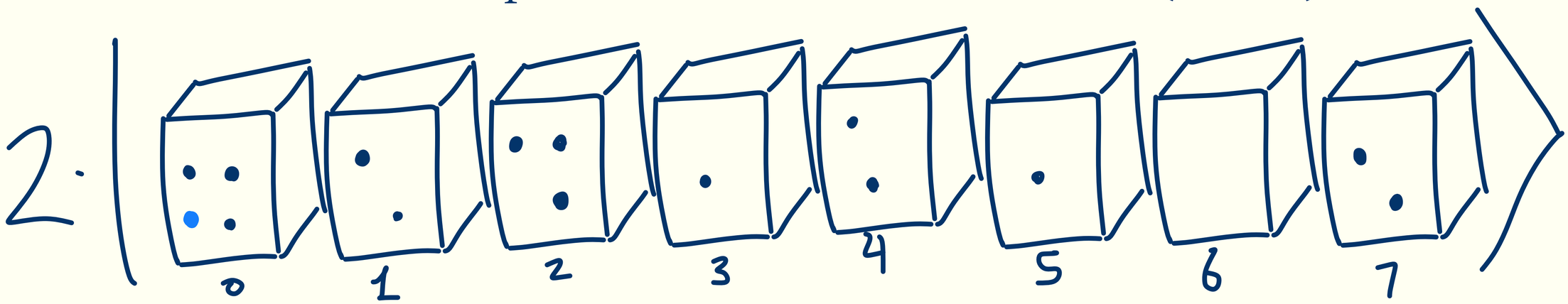


$$\hat{a}_x \,|\, \ell_0, \ldots, \ell_x, \ldots, \ell_{2^n-1}\rangle = \sqrt{\ell_x}\,|\, \ell_0, \ldots, \ell_x - 1, \ldots, \ell_{2^n-1}\rangle$$

(after $\hat{a}_0$)

Sampler upper bounds for Strong: Bosons

Bosons are a mathematical representation of multi-sets used in physics. The "annihilation" and "creation" operators add and subtract elements (bosons).



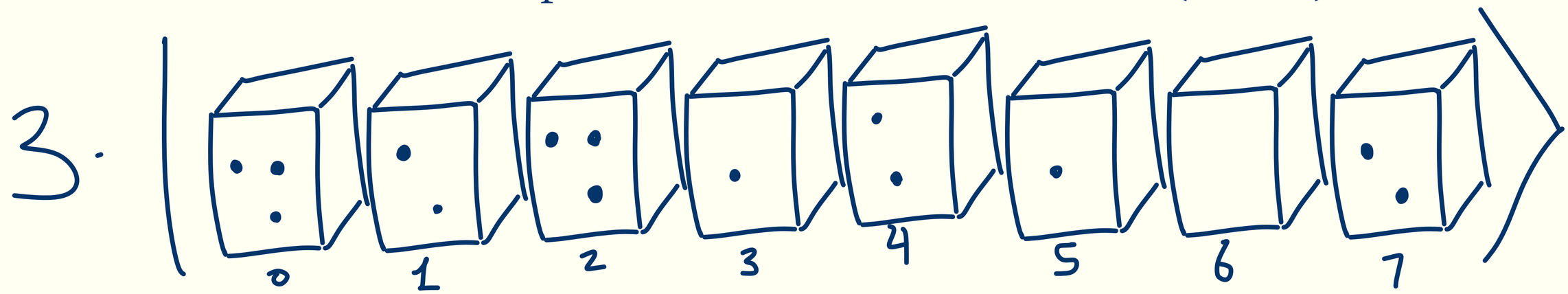
$$\hat{a}_x \,|\, \ell_0, \ldots, \ell_x, \ldots, \ell_{2^n-1} \rangle = \sqrt{\ell_x} \,|\, \ell_0, \ldots, \ell_x - 1, \ldots, \ell_{2^n-1} \rangle$$

$$\hat{a}_x^\dagger \,|\, \ell_0, \ldots, \ell_x, \ldots, \ell_{2^n-1} \rangle = \sqrt{\ell_x + 1} \,|\, \ell_0, \ldots, \ell_x + 1, \ldots, \ell_{2^n-1} \rangle$$

(after $\hat{a}_2^\dagger$)

Sampler upper bounds for Strong: Bosons

Bosons are a mathematical representation of multi-sets used in physics. The "annihilation" and "creation" operators add and subtract elements (bosons).



$$\hat{a}_x \left| \ell_0, \ldots, \ell_x, \ldots, \ell_{2^n-1} \right\rangle = \sqrt{\ell_x} \left| \ell_0, \ldots, \ell_x - 1, \ldots, \ell_{2^n-1} \right\rangle$$

$$\hat{a}_x^\dagger \left| \ell_0, \ldots, \ell_x, \ldots, \ell_{2^n-1} \right\rangle = \sqrt{\ell_x + 1} \left| \ell_0, \ldots, \ell_x + 1, \ldots, \ell_{2^n-1} \right\rangle$$

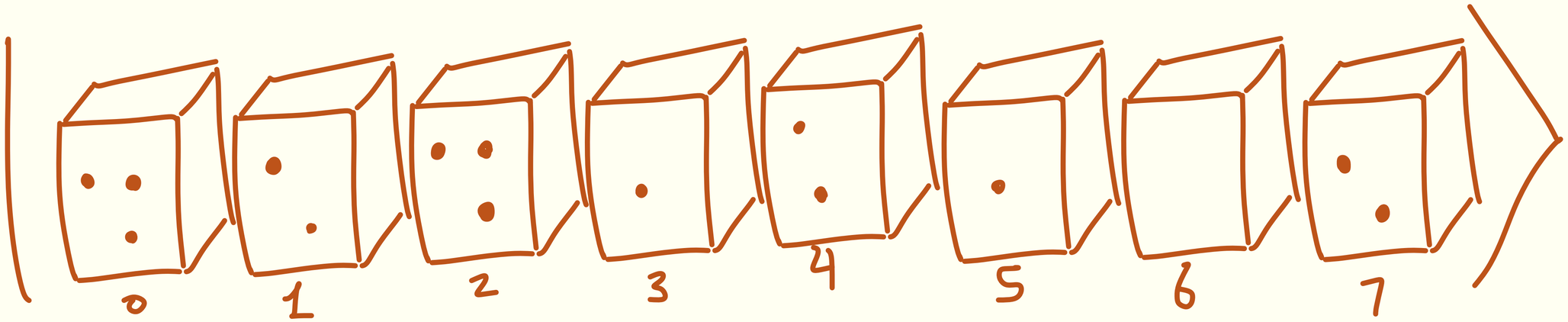
(after $\hat{n}_0$)

Taking the product, the number operator $\hat{n}_x = \hat{a}_x^\dagger \hat{a}_x$ is diagonal in the position Fock basis, and applies a scaling of the number of bosons in the x'th mode.

Sampler upper bounds for Strong: Bosons

We can also define a “Hadamard” basis for the bosons, with the analogous operators:

$$\tilde{a}_y = \frac{1}{\sqrt{2^n}} \sum_x (-1)^{y \cdot x} \hat{a}_x \quad \text{and} \quad \tilde{a}_y^\dagger = \frac{1}{\sqrt{2^n}} \sum_x (-1)^{y \cdot x} \hat{a}_x^\dagger$$

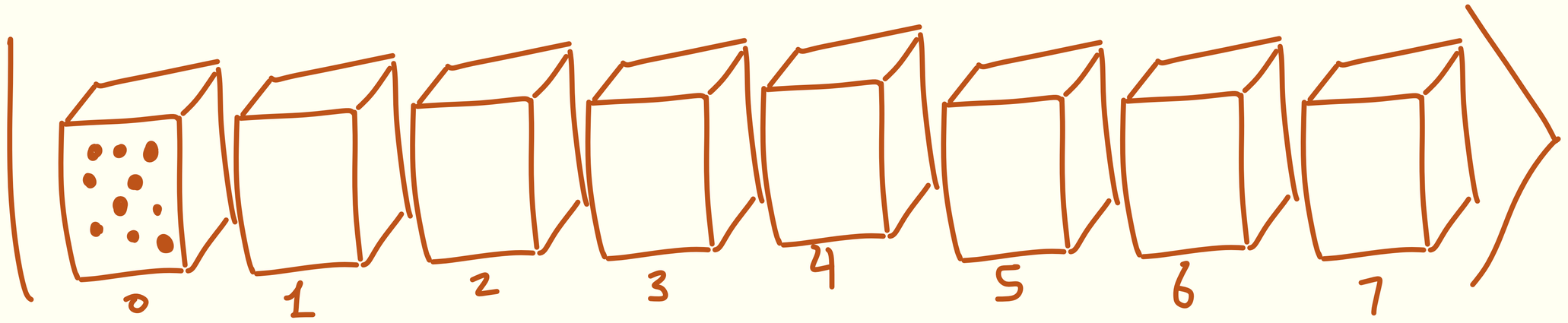


As a matter of notation, we also define $|\text{vac}\rangle$ to be the state $|0,\ldots,0\rangle$.

Sampler upper bounds for Strong: Bosons

Claim: The purification of a random multi-set is:

$$|\,\text{init}\rangle = \frac{1}{\sqrt{\ell\,!}} \left(\tilde{a}_0^\dagger\right)^\ell |\,\text{vac}\rangle$$



Sampler upper bounds for Strong: Bosons

Claim: The purification of a random multi-set is:

$$|\,\text{init}\rangle = \frac{1}{\sqrt{\ell\,!}}\left(\tilde{a}_0^\dagger\right)^\ell |\,\text{vac}\rangle$$

Proof: Expand out the expression:

$$\frac{1}{\sqrt{\ell\,!}}\left(\tilde{a}_0^\dagger\right)^\ell |\,\text{vac}\rangle = \frac{1}{\sqrt{2^{n\ell}\cdot\ell\,!}} \sum_{s_1,\ldots,s_\ell} \hat{a}_{s_1}^\dagger \ldots \hat{a}_{s_\ell}^\dagger |\,\text{vac}\rangle$$

If you deal with the coefficients (and multiplicities of the multi-sets), it works out. 😊

Compressed oracles for Strong: Action of U

Roughly, querying U at some fixed y is like querying the squared Fourier coefficient $\gamma_y^{(S)}$.
What happens when we apply the diagonal matrix

$$\sum_S \gamma_y^{(S)} \,|\mathrm{Fock}(S)\rangle\langle\mathrm{Fock}(S)|\,?$$

Compressed oracles for Strong: Action of U

Roughly, querying U at some fixed y is like querying the squared Fourier coefficient $\gamma_y^{(S)}$. What happens when we apply the diagonal matrix

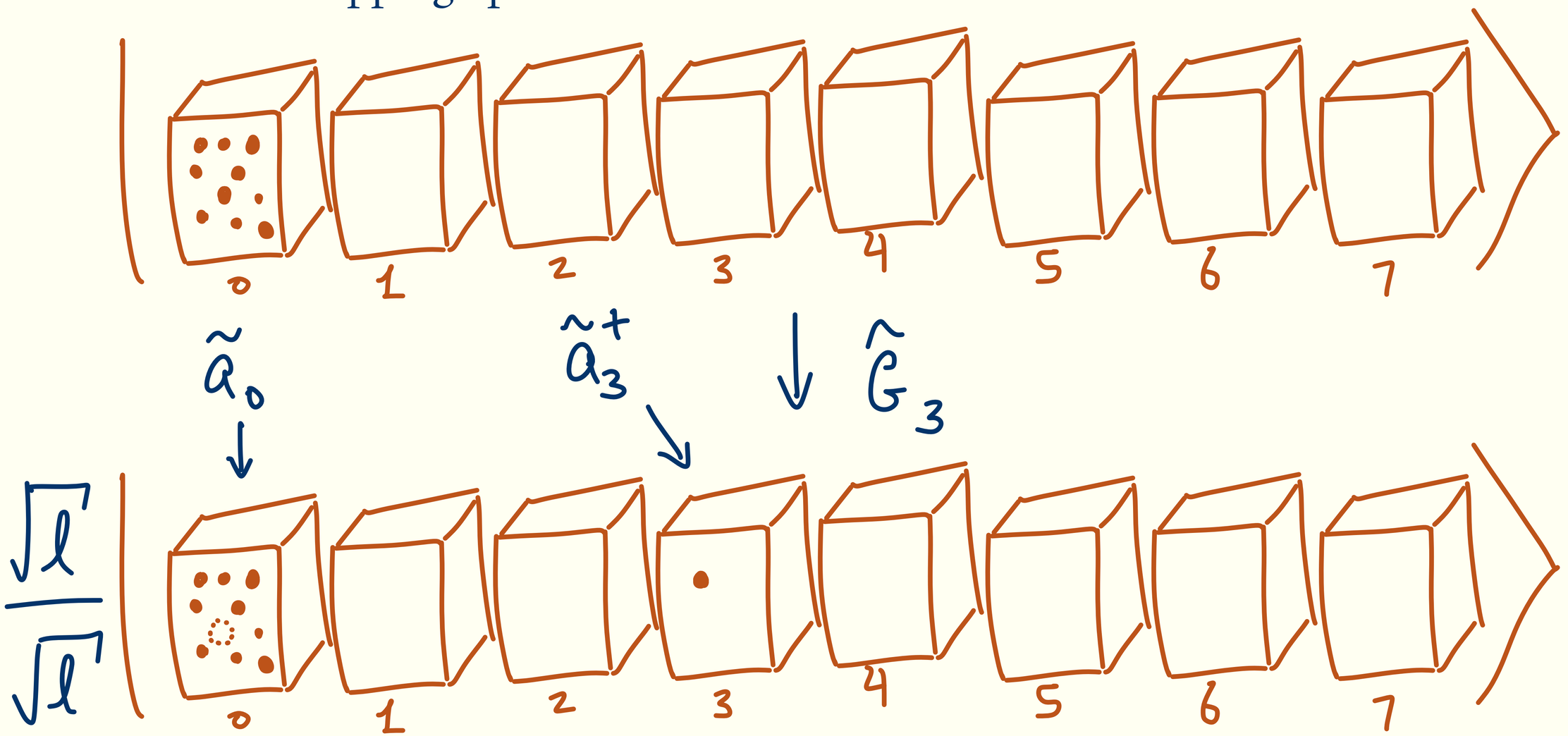
$$\sum_S \gamma_y^{(S)} |\mathrm{Fock}(S)\rangle\langle\mathrm{Fock}(S)| \, ?$$

Let's define the momentum hopping and double hopping operators

$$\widetilde{G}_y = \frac{1}{\sqrt{\ell}} \sum_{x \in \{0,1\}^n} \widetilde{a}^\dagger_{x \oplus y} \widetilde{a}_x \quad \text{and} \quad \widetilde{H}_y = \frac{1}{\ell} \sum_{x,x' \in \{0,1\}^n} \widetilde{a}^\dagger_{x \oplus y} \widetilde{a}^\dagger_{x' \oplus y} \widetilde{a}_x \widetilde{a}_{x'}$$

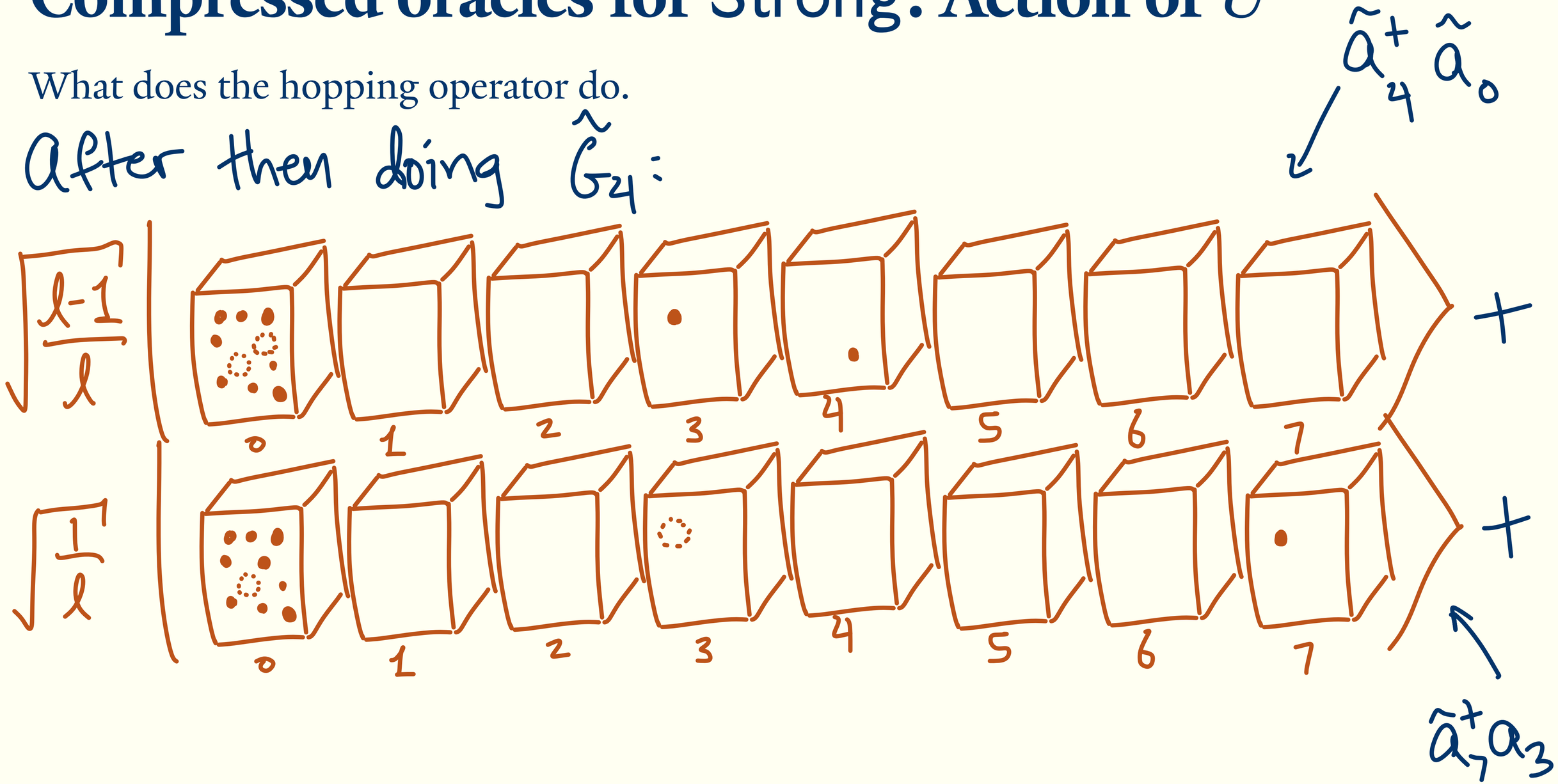
Compressed oracles for Strong: Action of U

What does the hopping operator do.



Compressed oracles for Strong: Action of U

What does the hopping operator do.



Compressed oracles for Strong: Action of U

Claim: The diagonal matrix that applies the squared Fourier coefficient is actually:

$$\sum_S \gamma_y^{(S)} \, |\mathrm{Fock}(S)\rangle\langle\mathrm{Fock}(S)| = \widetilde{G}_y^2$$

Compressed oracles for Strong: Action of U

Claim: The diagonal matrix that applies the squared Fourier coefficient is actually:

$$\sum_S \gamma_y^{(S)} |\mathrm{Fock}(S)\rangle\langle\mathrm{Fock}(S)| = \widetilde{G}_y^2$$

Proof: We can expand out a position Fock state in the momentum basis and directly compute the action of the hopping operator (the double hopping is the square):

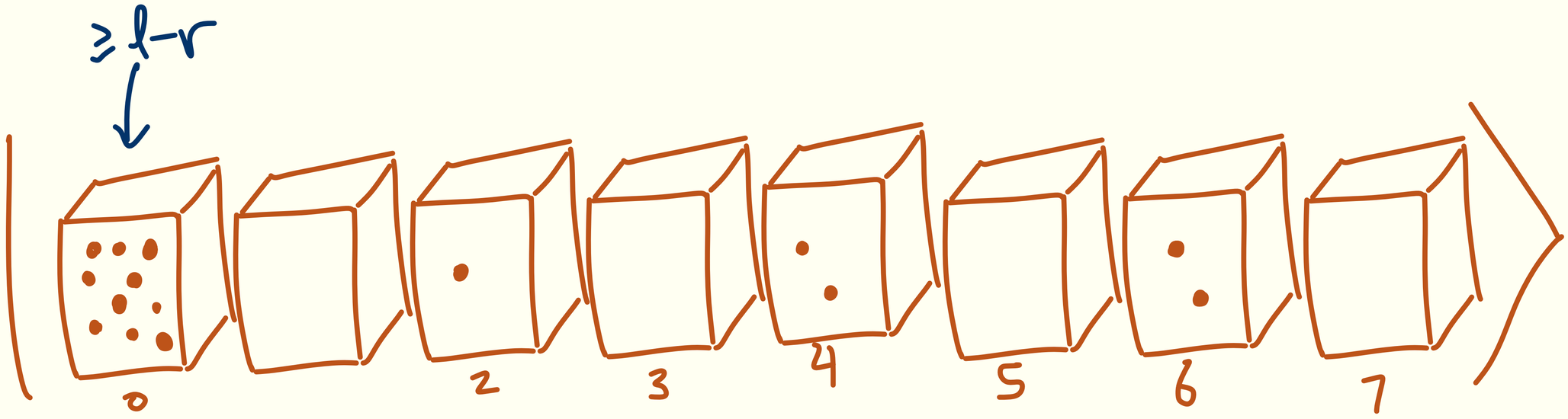
$$\widetilde{G}_y \hat{a}_{s_1}^\dagger \ldots \hat{a}_{s_\ell}^\dagger |\mathrm{vac}\rangle = \widetilde{G}_y \sum_{t_1,\ldots,t_\ell} \left(\prod_i (-1)^{t_i \cdot s_i} \right) \widetilde{a}_{t_1}^\dagger \ldots \widetilde{a}_{t_\ell}^\dagger |\mathrm{vac}\rangle$$

When we apply the hop and re-index the sum, we see that we just get a phase kickback!

Quasi-even condensates

A (r, o)-quasi-even condensate is a momentum Fock state $|\ell_0, \ldots, \ell_{2^n}\rangle$ that satisfies:

Condensate: $\ell_0 \geq \ell - r$, i.e., almost all of the bosons are in their initial position.

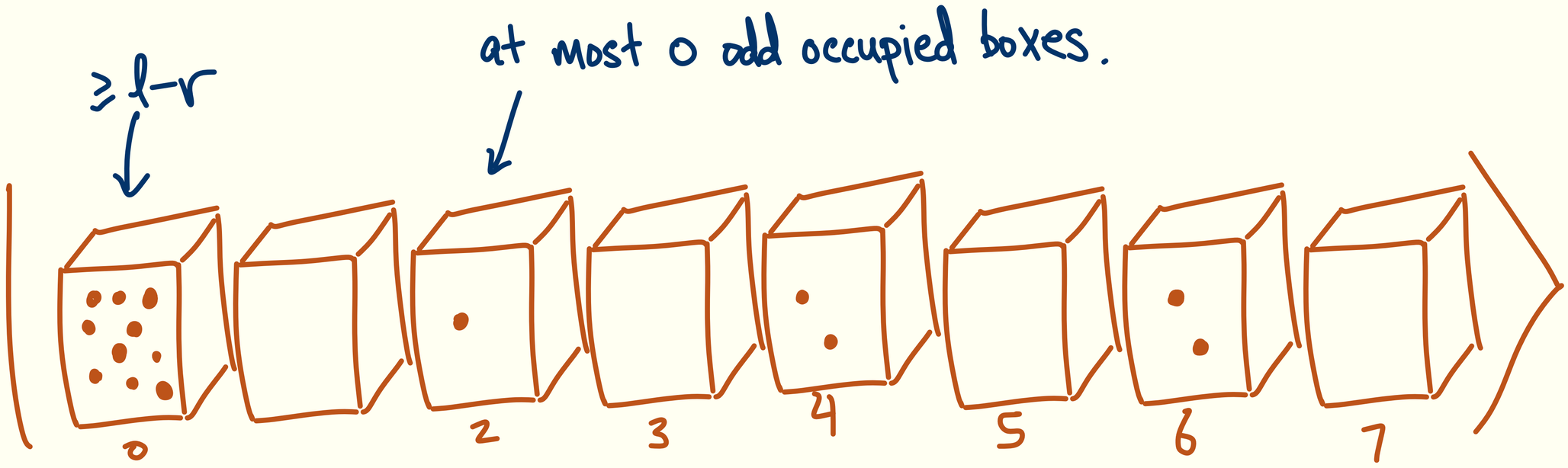


Quasi-even condensates

A (r, o)-quasi-even condensate is a momentum Fock state $|\ell_0, \ldots, \ell_{2^n}\rangle$ that satisfies:

Condensate: $\ell_0 \geq \ell - r$, i.e., almost all of the bosons are in their initial position.

Quasi-even: At most o of the non-zero indices are odd.



Sampling bounds on quasi-even condensates

Claim: Let $|\psi\rangle$ be a state that is supported entirely on (r, o)-quasi-even condensate, then the following bound holds for all collections $z_1, \ldots, z_v \in \{0,1\}^n$:

$$\langle\psi| n_{z_1} \ldots, n_{z_\ell} |\psi\rangle \leq \left(\text{poly}(v, r) \cdot \frac{\sqrt{\ell}}{2^{n/4}} \right)^v$$

Sampling bounds on quasi-even condensates

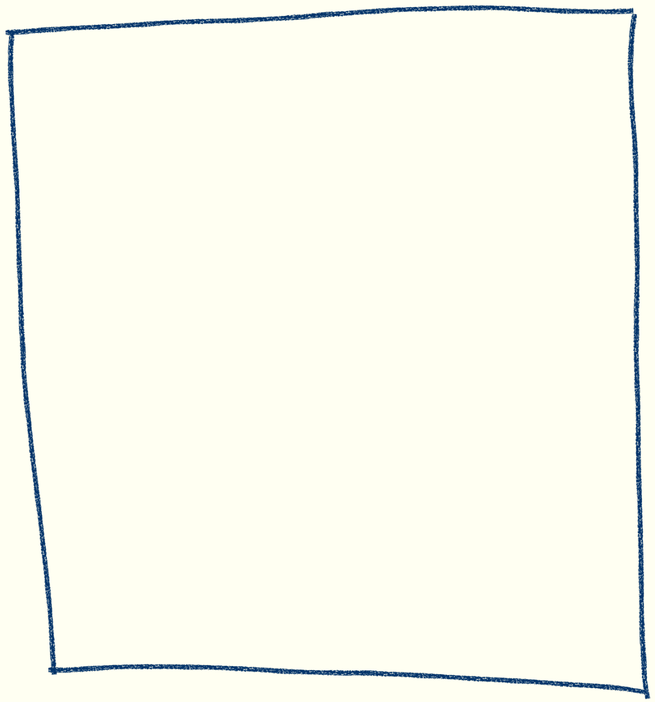
Claim: Let $|\psi\rangle$ be a state that is supported entirely on (r, o)-quasi-even condensate, then the following bound holds for all collections $z_1, \ldots, z_v \in \{0,1\}^n$:

$$\langle\psi| n_{z_1} \ldots, n_{z_\ell} |\psi\rangle \leq \left(\text{poly}(v, r) \cdot \frac{\sqrt{\ell}}{2^{n/4}} \right)^v$$

This number upper bounds the sampling success probability (applying Markov's inequality) → proving that the "purified" state of any algorithm is a quasi-even condensate would complete the proof.

The code intersection problem

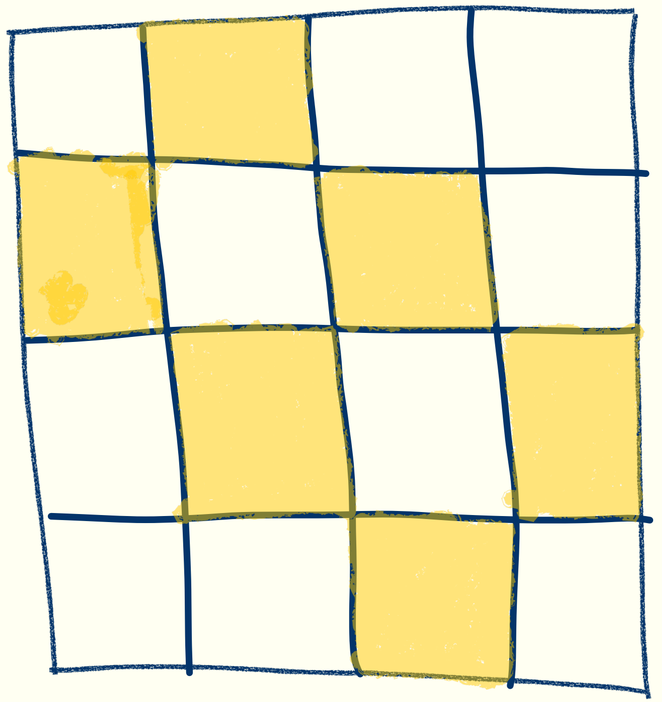
Here is how we can sample an instance of the code intersection problem:



The code intersection problem

Here is how we can sample an instance of the code intersection problem:

1. Take any linear code $C \subseteq (\mathbb{F}_q^s)^n$



$(q=4, n=2)$

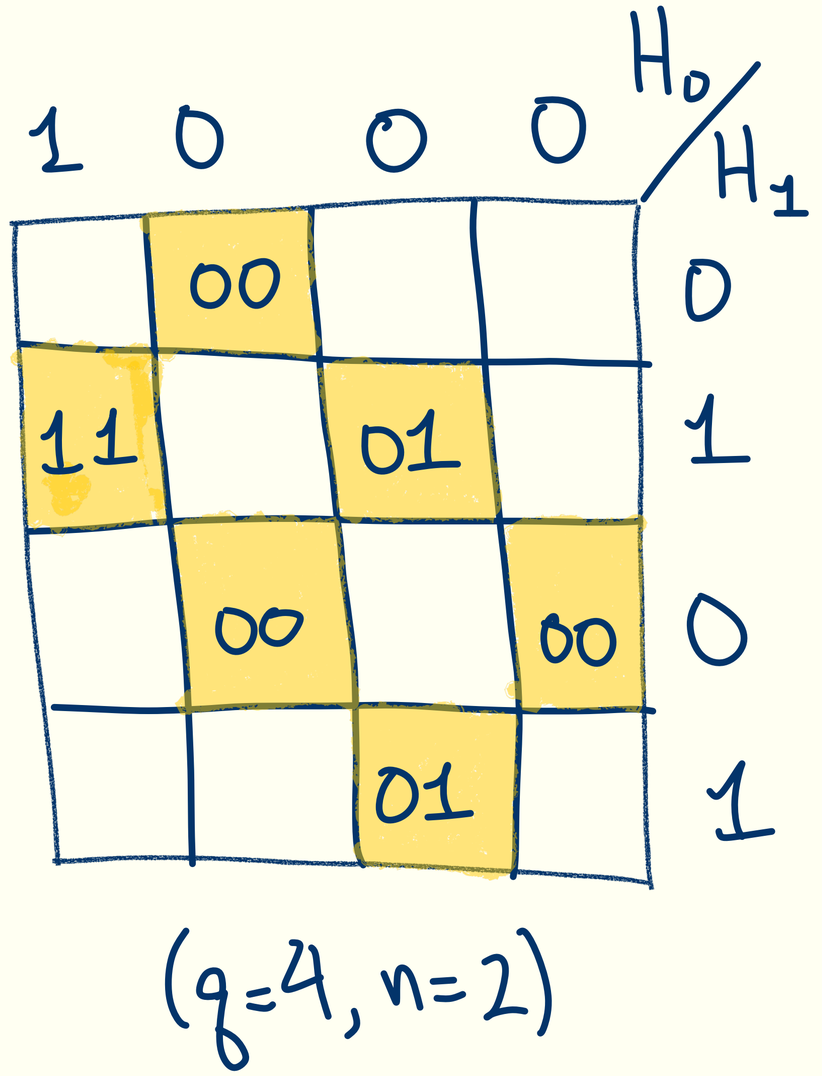
$s=1$

The code intersection problem

Here is how we can sample an instance of the code intersection problem:

1. Take any linear code $C \subseteq (\mathbb{F}_q^s)^n$

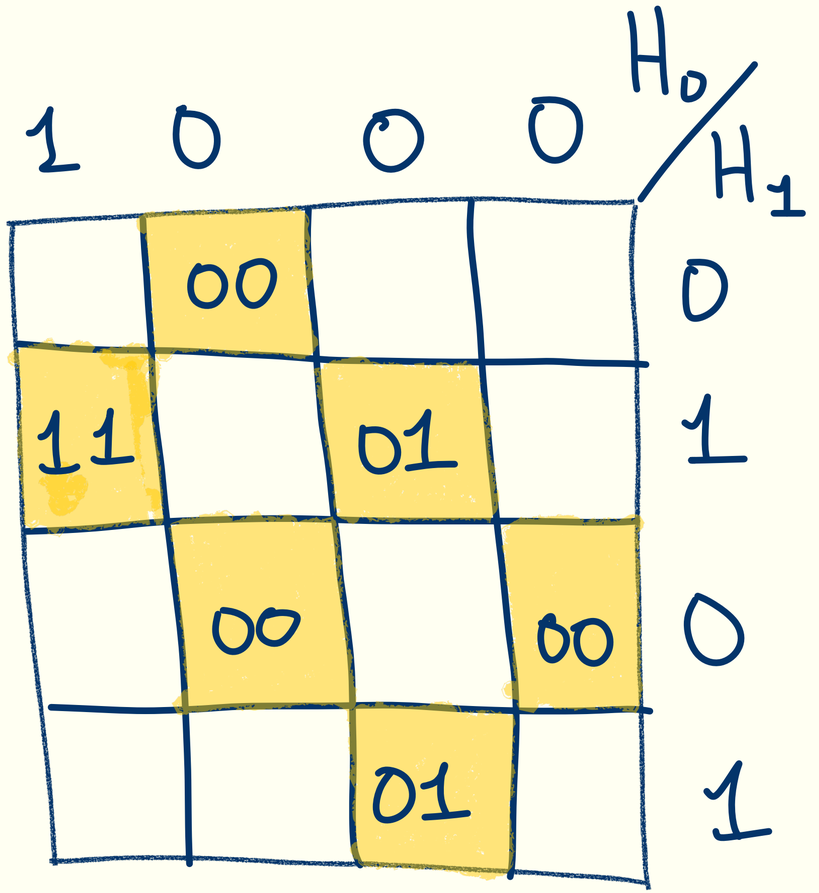
2. Sample random functions $H_i : \mathbb{F}_q^s \mapsto \{0,1\}$



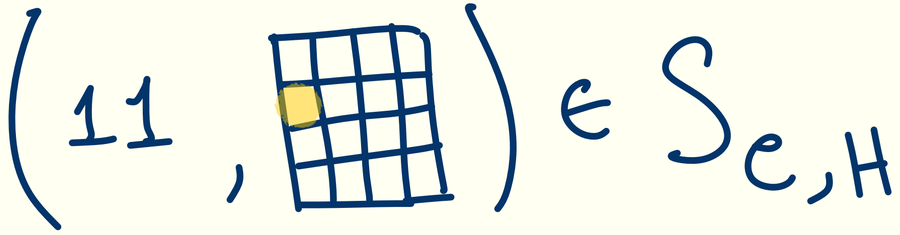
The code intersection problem

Here is how we can sample an instance of the code intersection problem:

1. Take any linear code $C \subseteq (\mathbb{F}_q^s)^n$
2. Sample random functions $H_i : \mathbb{F}_q^s \mapsto \{0,1\}$
3. Let the set be $S_{C,H} = \{(x, c) : c \in C \text{ and } H_i(c_i) = x_i \ \forall i\}$



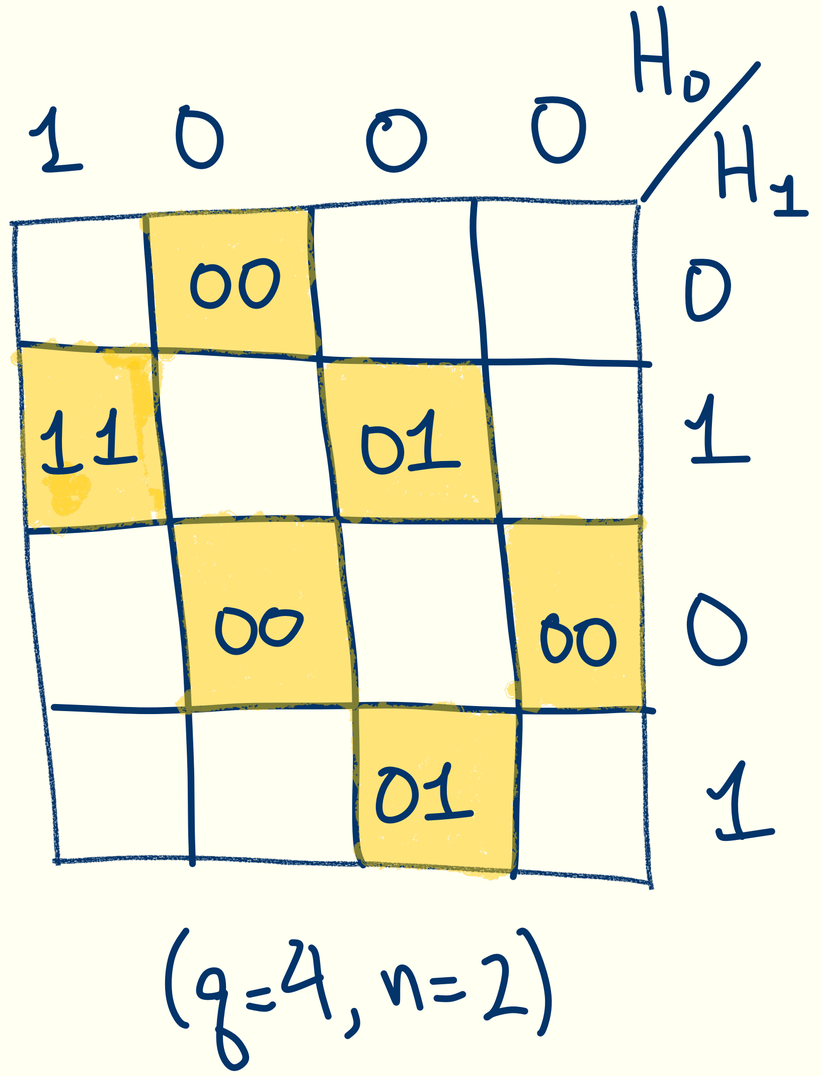
(q=4, n=2)



The code intersection problem

$S_{C,H} = \{(x, v) : v \in C \text{ and } H_i(v_i) = x_i \ \forall i\}$

Input: Oracle access to $S_{C,H} \cap E$ for $E \subseteq \{0,1\}^n \times (\mathbb{F}_q^s)^n$



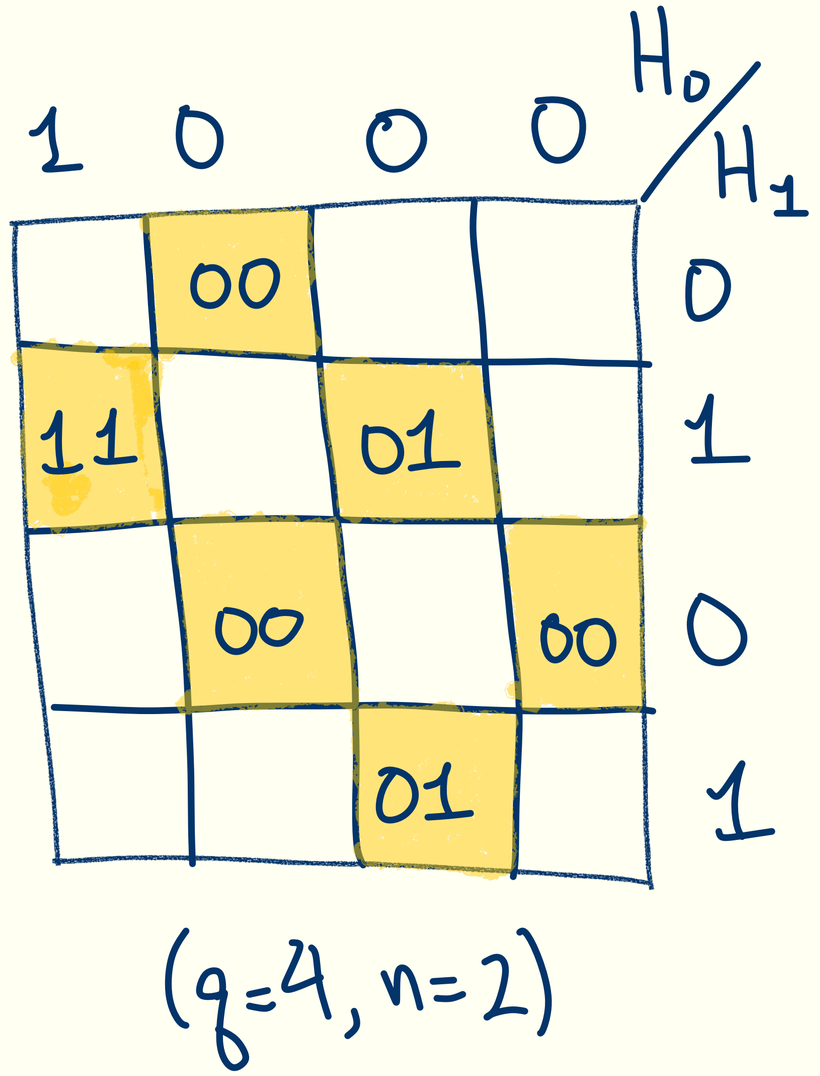
The code intersection problem

$S_{C,H} = \{(x, v) : v \in C \text{ and } H_i(v_i) = x_i \ \forall i\}$

Input: Oracle access to $S_{C,H} \cap E$ for $E \subseteq \{0,1\}^n \times (\mathbb{F}_q^s)^n$

Decide between:

(Yes) every x has a matching $(x, v) \in S_{C,H} \cap E$, i.e., $E = \{0,1\}^n \times (\mathbb{F}_q^s)^n$, or



The code intersection problem

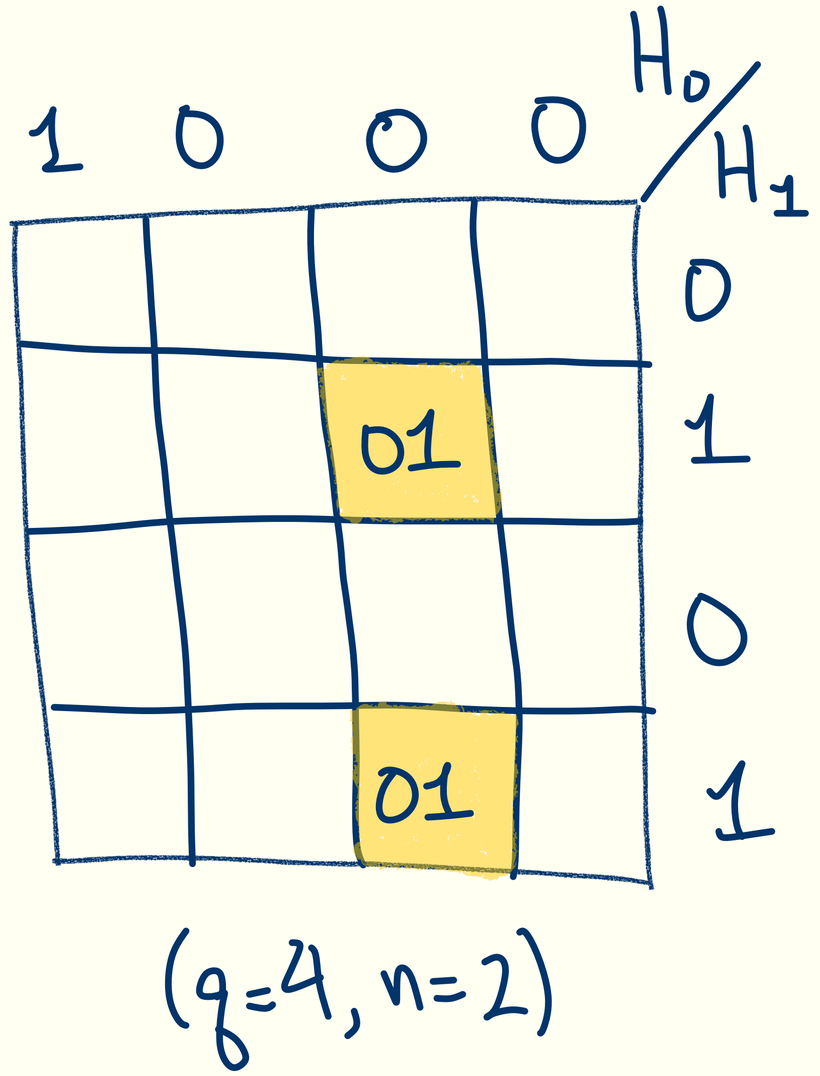
$S_{C,H} = \{(x, v) : v \in C \text{ and } H_i(v_i) = x_i \ \forall i\}$

Input: Oracle access to $S_{C,H} \cap E$ for $E \subseteq \{0,1\}^n \times (\mathbb{F}_q^s)^n$

Decide between:

(Yes) every x has a matching $(x, v) \in S_{C,H} \cap E$, i.e., $E = \{0,1\}^n \times (\mathbb{F}_q^s)^n$, or

(No) at most $1/3$ of $x's$ are in $S_{C,H} \cap E$, i.e., $E \subseteq F \times (\mathbb{F}_q^s)^n$, for some $|F| \leq 2^n/3$, promised one is the case.



Code intersection is in QMA

The code intersection problem (for certain functions and codes) is in QMA because of the Yamakawa-Zhandry algorithm.

Code intersection is in QMA

The code intersection problem (for certain functions and codes) is in QMA because of the Yamakawa-Zhandry algorithm. It works because quantum algorithms can “realize” the convolution trick:

$$|\psi\rangle \odot |\phi\rangle = \mathrm{QFT}^{-1}\left(\mathrm{QFT}\,|\psi\rangle \star \mathrm{QFT}\,|\phi\rangle\right)$$

In the code and has the right hash

Add and decode the dual code

Code intersection is in QMA

The code intersection problem (for certain functions and codes) is in QMA because of the Yamakawa-Zhandry algorithm. It works because quantum algorithms can “realize” the convolution trick:

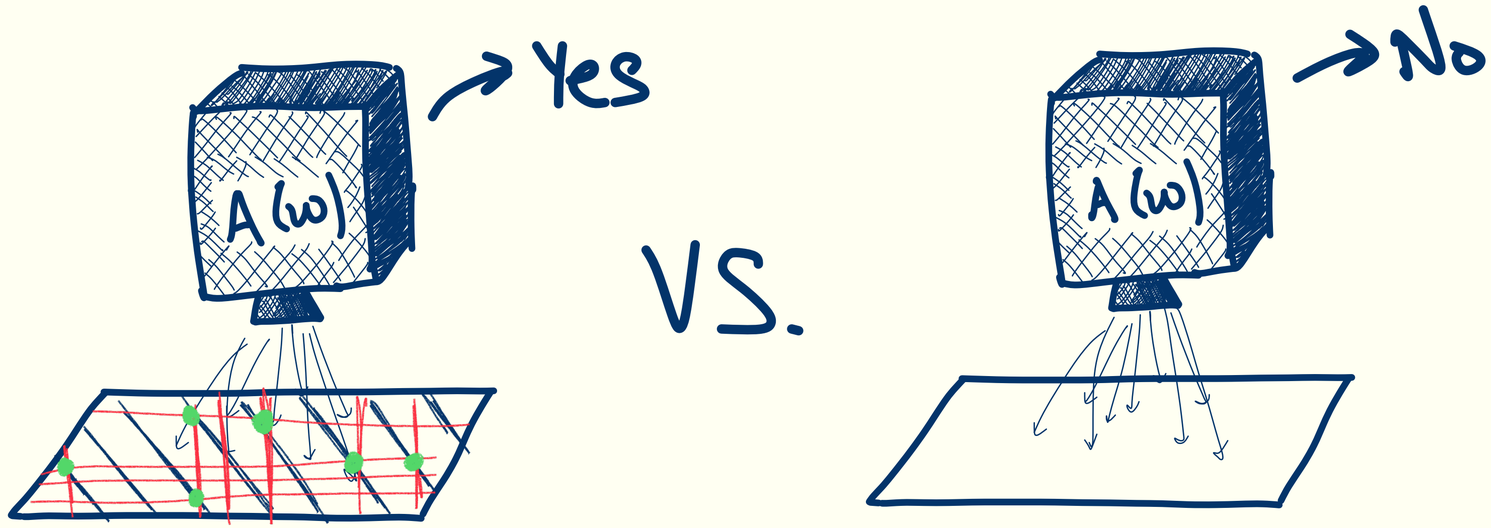
$$\underbrace{|\psi\rangle \odot |\phi\rangle}_{\text{In the code and has the right hash}} = \mathrm{QFT}^{-1}\left(\mathrm{QFT}\,|\psi\rangle \underbrace{\star\, \mathrm{QFT}\,|\phi\rangle}_{\text{Add and decode the dual code}}\right)$$

If the functions are random and the code has high enough distance, the Fourier transform of the witness corresponds to decodable errors.

Applying the sampler idea to code intersection

Theorem: If there is a QCMA algorithm, making t queries to $S_{C,H}$, and taking a witness of length w, then for all $0 \leq v \leq 2^n/3$, there is an algorithm making no queries that outputs v many pairs of distinct codewords and their hashes, (x, c) with probability

$$\geq 2^w \left(\frac{1}{36t^2}\right)^v$$



List recoverability and code intersection

Given a QCMA algorithm, we get a sampler that outputs many **codewords**, but the hard thing to guess are the hashes of the individual **symbols**!

List recoverability and code intersection

Given a QCMA algorithm, we get a sampler that outputs many **codewords**, but the hard thing to guess are the hashes of the individual **symbols**!

The minimum number of symbols that the sampler has to output, given that it outputs v many codewords, is described by the **list recoverability** of the code.

List recoverability and code intersection

Given a QCMA algorithm, we get a sampler that outputs many **codewords**, but the hard thing to guess are the hashes of the individual **symbols**!

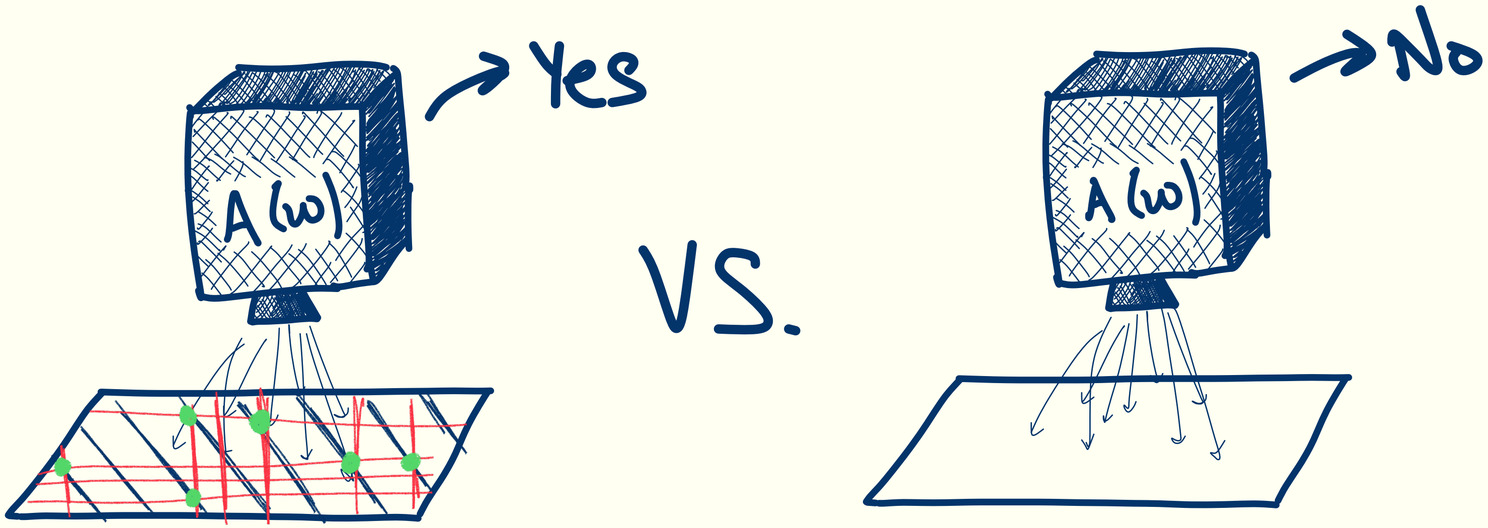
The minimum number of symbols that the sampler has to output, given that it outputs v many codewords, is described by the **list recoverability** of the code.

Definition: A code is (ℓ, L)-list recoverable if, given lists $S_1, \ldots, S_n$ of average size ℓ, the number of codewords you can build from them is at most L.

List recoverability and code intersection

If we instantiate our code intersection problem with a $(v/2,v)$-list recoverable code, then the sampler must guess at least $nv/2$ many hash values, so we can bound its success probability by

$$(1-p)^{nv/2} \approx \left(\frac{1}{2^{\lambda}}\right)^{v}$$



List recoverability and code intersection

Good news: There are families of codes (we take multiplicity codes) that have good list recovery up to $v = 2^n$ many codewords, so we can take those in our code.

$$f(X) \mapsto \begin{pmatrix} f(\alpha_1), \ldots, f(\alpha_n) \\ f'(\alpha_1), \ldots, f'(\alpha_n) \\ f^{(2)}(\alpha_1), \ldots, f^{(2)}(\alpha_n) \\ \cdots \end{pmatrix}$$

List recoverability and code intersection

Good news: There are families of codes (we take multiplicity codes) that have good list recovery up to $v = 2^n$ many codewords, so we can take those in our code.

$$f(X) \mapsto \begin{pmatrix} f(\alpha_1), \ldots, f(\alpha_n) \\ f'(\alpha_1), \ldots, f'(\alpha_n) \\ f^{(2)}(\alpha_1), \ldots, f^{(2)}(\alpha_n) \\ \cdots \end{pmatrix}$$

Bad news: The distance of the dual codes for multiplicity codes is quite small, so it's not clear that a QMA algorithm can actually solve this problem anymore!

List recoverability and code intersection

Good news: There are families of codes (we take multiplicity codes) that have good list recovery up to $v = 2^n$ many codewords, so we can take those in our code.

$$f(X) \mapsto \begin{pmatrix} f(\alpha_1), \ldots, f(\alpha_n) \\ f'(\alpha_1), \ldots, f'(\alpha_n) \\ f^{(2)}(\alpha_1), \ldots, f^{(2)}(\alpha_n) \\ \cdots \end{pmatrix}$$

Bad news: The distance of the dual codes for multiplicity codes is quite small, so it's not clear that a QMA algorithm can actually solve this problem anymore!

Turns out this can be solved without too many modifications to the YZ algorithm.

Takeaways

Takeaways

- **Quantum proofs are really powerful!**
  → That power is what we think makes them not reusable!
  → Our proof finds a task (sampling) that should be really hard, and shows that a reusable proof would be too good to be true.

Takeaways

- **Quantum proofs are really powerful!**
  → That power is what we think makes them not reusable!
  → Our proof finds a task (sampling) that should be really hard, and shows that a reusable proof would be too good to be true.
- **Small structural changes can have a huge impact!**
  → The analysis of spectral Forrelation relies on a seemingly tiny change: allowing the set S to be a multi-set with small probability (i.e., s_i are sampled independently).
  → In the code intersection problem, just adding the derivatives of the polynomial gives us the increased list recovery needed to make the lower bound work.

Takeaways

- **Quantum proofs are really powerful!**
  → That power is what we think makes them not reusable!
  → Our proof finds a task (sampling) that should be really hard, and shows that a reusable proof would be too good to be true.
- **Small structural changes can have a huge impact!**
  → The analysis of spectral Forrelation relies on a seemingly tiny change: allowing the set S to be a multi-set with small probability (i.e., s_i are sampled independently).
  → In the code intersection problem, just adding the derivatives of the polynomial gives us the increased list recovery needed to make the lower bound work.
- **Much more work is needed!**
  → Understanding oracles with structure seems to require an understanding that structure, seem to be annoying to deal with using general methods.
  → We still don't really understand all the "kinds" of structure that a quantum proof can help identify that a classical proof can't.

Open questions

Open questions

- **Can we find new constructions/security proofs for quantum money?**
  → Our ideas lie in the intersection of ideas used for quantum money (subset states ↔ subspace states, Fourier transform of S ↔ Fourier transform for group actions).
  → We also prove a separation between ClonableQMA and QMA, feels like we should be able to say something about quantum money, but what?

Open questions

- **Can we find new constructions/security proofs for quantum money?**
  → Our ideas lie in the intersection of ideas used for quantum money (subset states ↔ subspace states, Fourier transform of S ↔ Fourier transform for group actions).
  → We also prove a separation between UnclonableQMA and QMA, feels like we should be able to say something about quantum money, but what?

- **Can we use our oracle/techniques to solve other problems in query complexity?**
  → QMA search-to-decision?
  → Communication complexity QMA versus QCMA?

Open questions

- **Can we find new constructions/security proofs for quantum money?**
  → Our ideas lie in the intersection of ideas used for quantum money (subset states ↔ subspace states, Fourier transform of S ↔ Fourier transform for group actions).
  → We also prove a separation between UnclonableQMA and QMA, feels like we should be able to say something about quantum money, but what?

- **Can we use our oracle/techniques to solve other problems in query complexity?**
  → QMA search-to-decision?
  → Communication complexity QMA versus QCMA?

- **Is there a connection to the Aaronson-Ambainis conjecture?**
  → Both Liu-Mutreja-Yuen'24 and Zhandry'24 showed that there is a connection between QCMA versus QMA and pseudorandomness against quantum algorithms.
  → Our proof didn't say anything about this, but could you use our techniques?

Thanks for listening!