

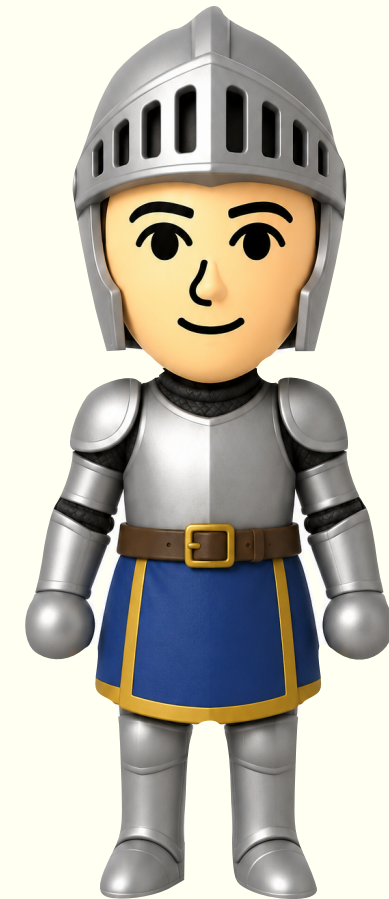
Classical oracle separations between QMA and QCMA

John Bostanci

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

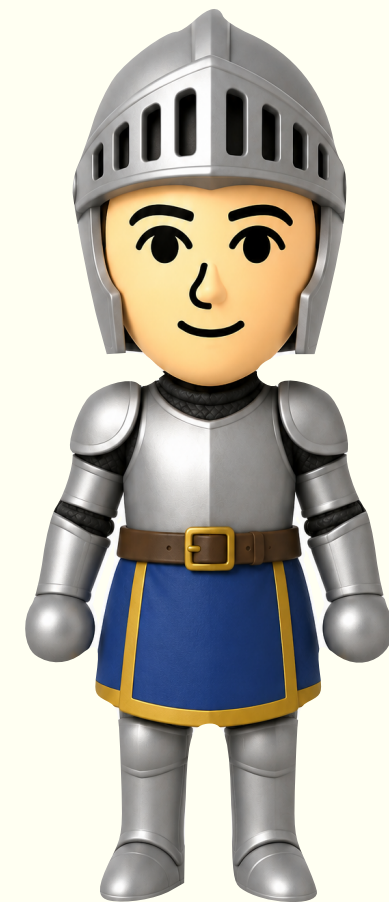
Proofs in computer science

Efficient Classical
Algorithm



Proofs in computer science

Efficient Classical
Algorithm



vs

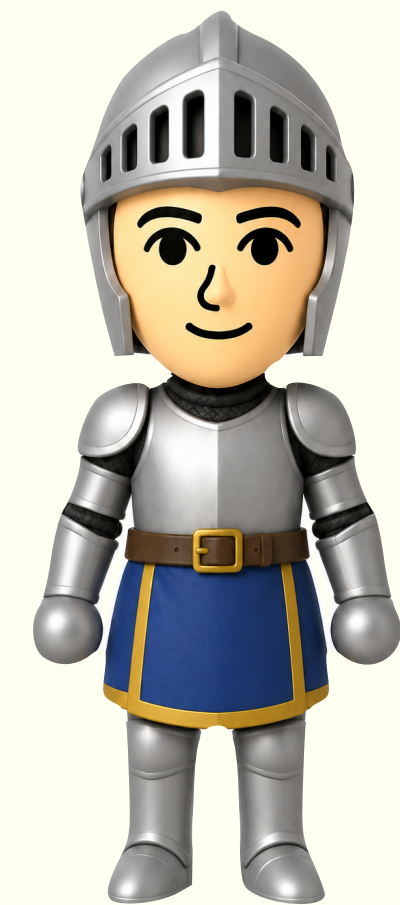
Unbounded
prover



Classical proof



Efficient Classical
Algorithm



Proofs in quantum computer science

Efficient Quantum
Algorithm



vs

Unbounded
prover



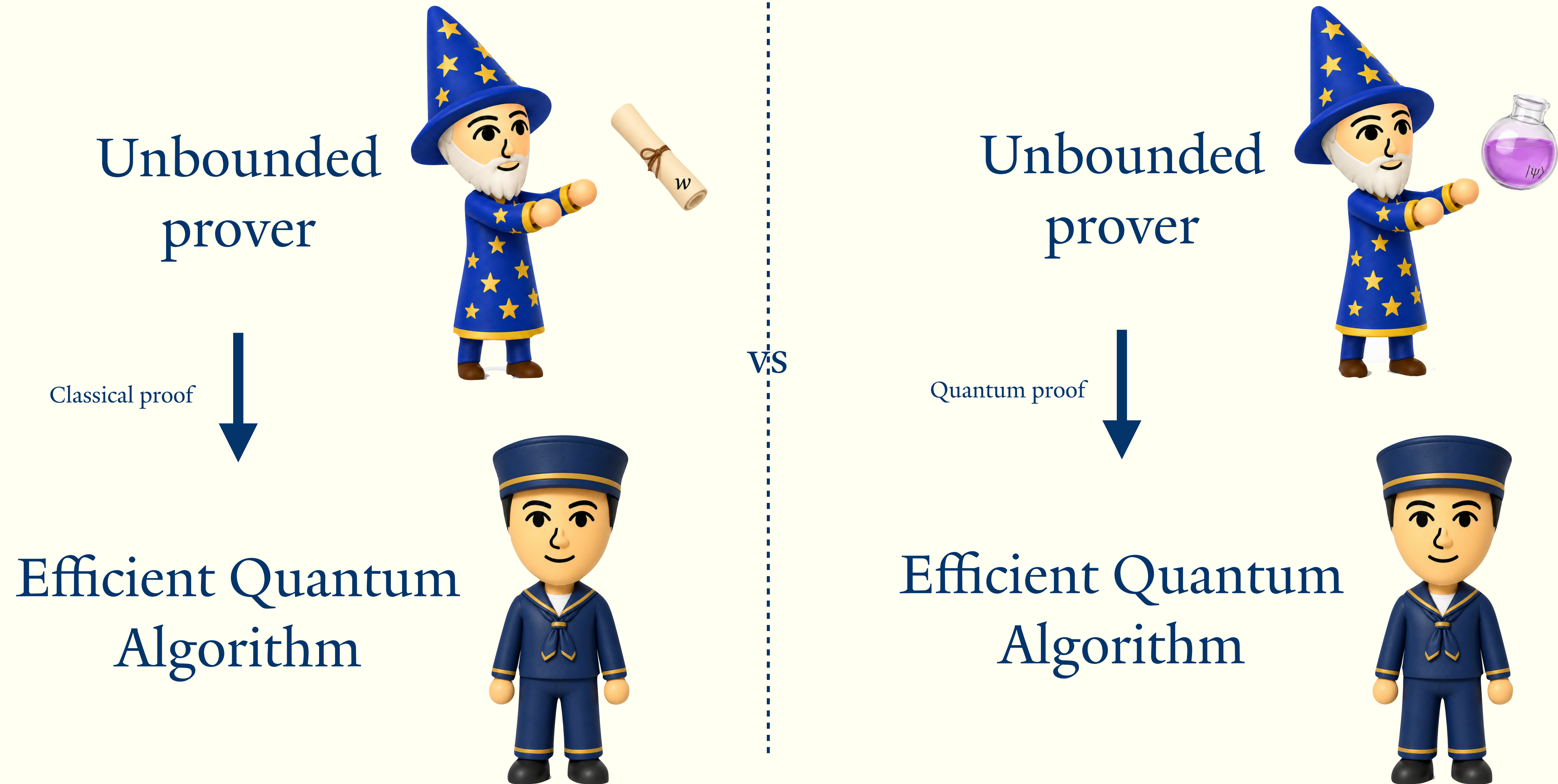
Classical proof



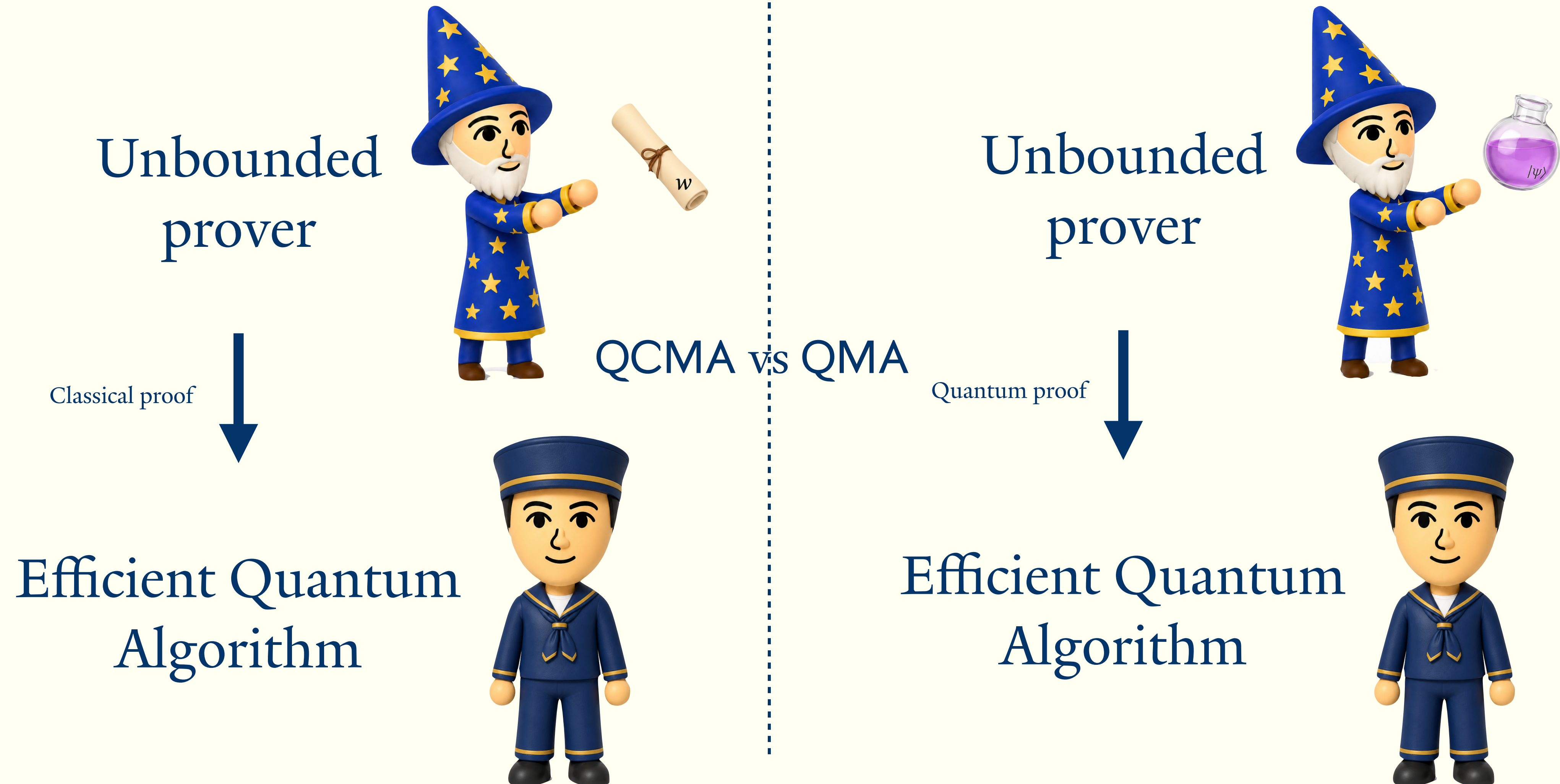
Efficient Quantum
Algorithm



Proofs in quantum computer science

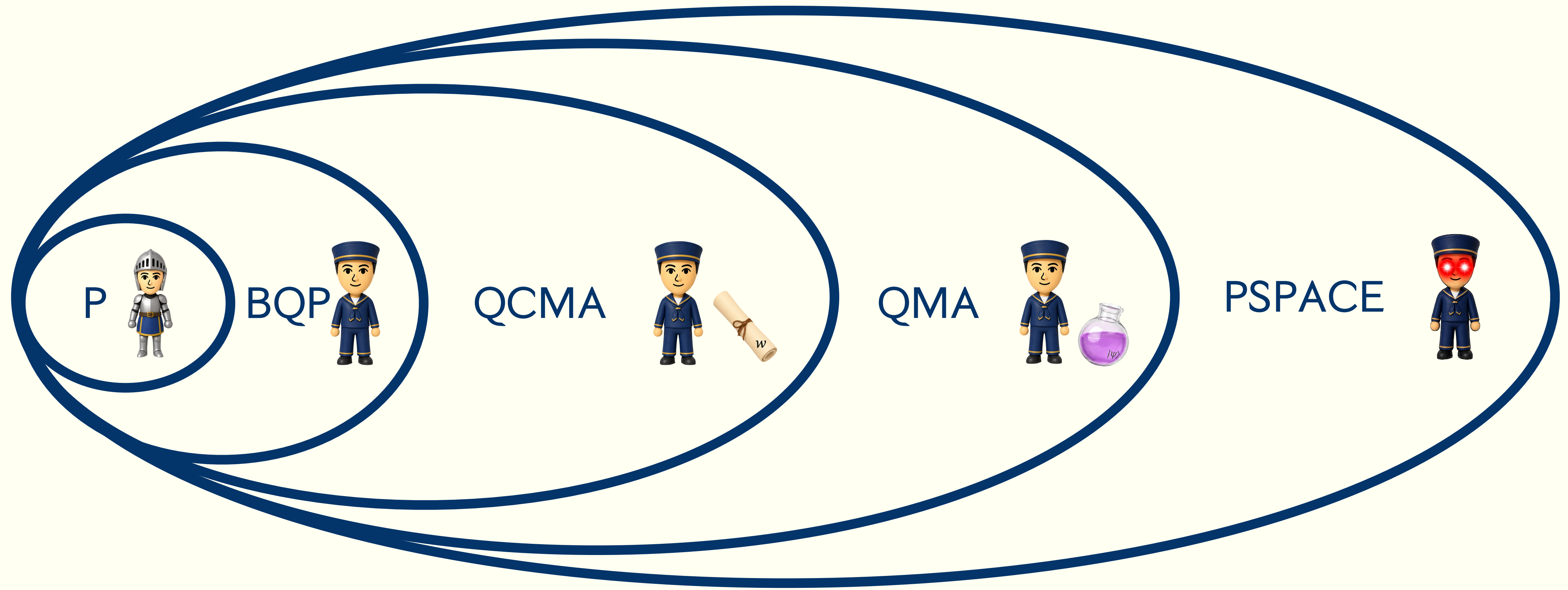


Proofs in quantum computer science



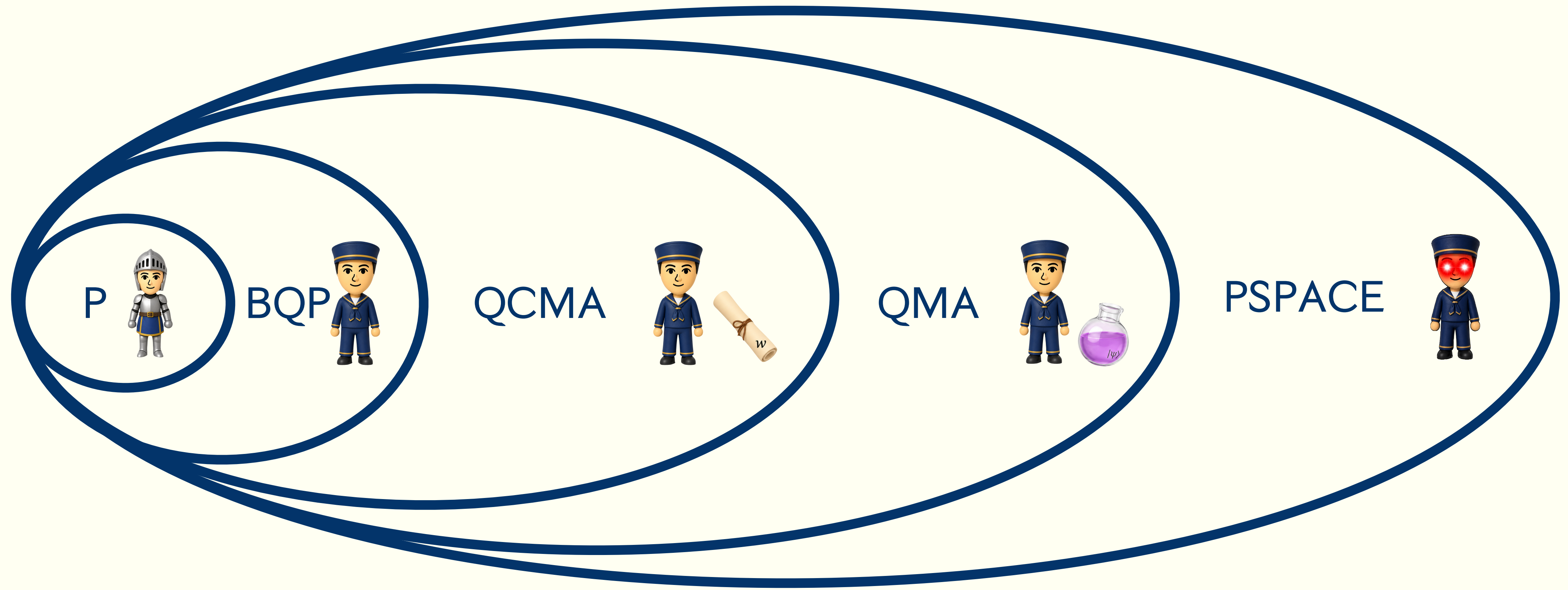
Separating QMA from QCMA

Unfortunately, both QMA and QCMA are both between P and PSPACE, so separating them unconditionally seems out of reach.



Separating QMA from QCMA

Unfortunately, both QMA and QCMA are both between P and PSPACE, so separating them unconditionally seems out of reach.



The next best thing would be an oracle separation!

History of the QMA versus QCMA problem

- First proposed in '02 by **Aharanov and Naveh**.
- **Aaronson & Kuperberg '06**: Quantum oracle separation. Each $\mathcal{O}_n = \text{id} - 2|\psi_n\rangle\langle\psi_n|$
- **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, Pelecinos, 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 do we care about QMA versus QCMA

- It would be surprising if the ground energy of local Hamiltonians could be efficiently checked without a quantum state (say, a copy of the ground state). We hope that understanding QMA versus QCMA can give us evidence for or against that.



Why do we care about QMA versus QCMA

- It would be surprising if the ground energy of local Hamiltonians could be efficiently checked without a quantum state (say, a copy of the ground state). We hope that understanding QMA versus QCMA can give us evidence for or against that.



- Properties we care about, like non-collapsing, un-clonability, un-telegraphicability, are requirements for a separation between QMA and QCMA, and are poorly understood and important for many applications of quantum computing to cryptography!



Why is this problem so hard?

Why is this problem so hard?

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



VS



Why is this problem so hard?

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

- One idea is to “guess the witness”, but this works for a quantum witness.



VS

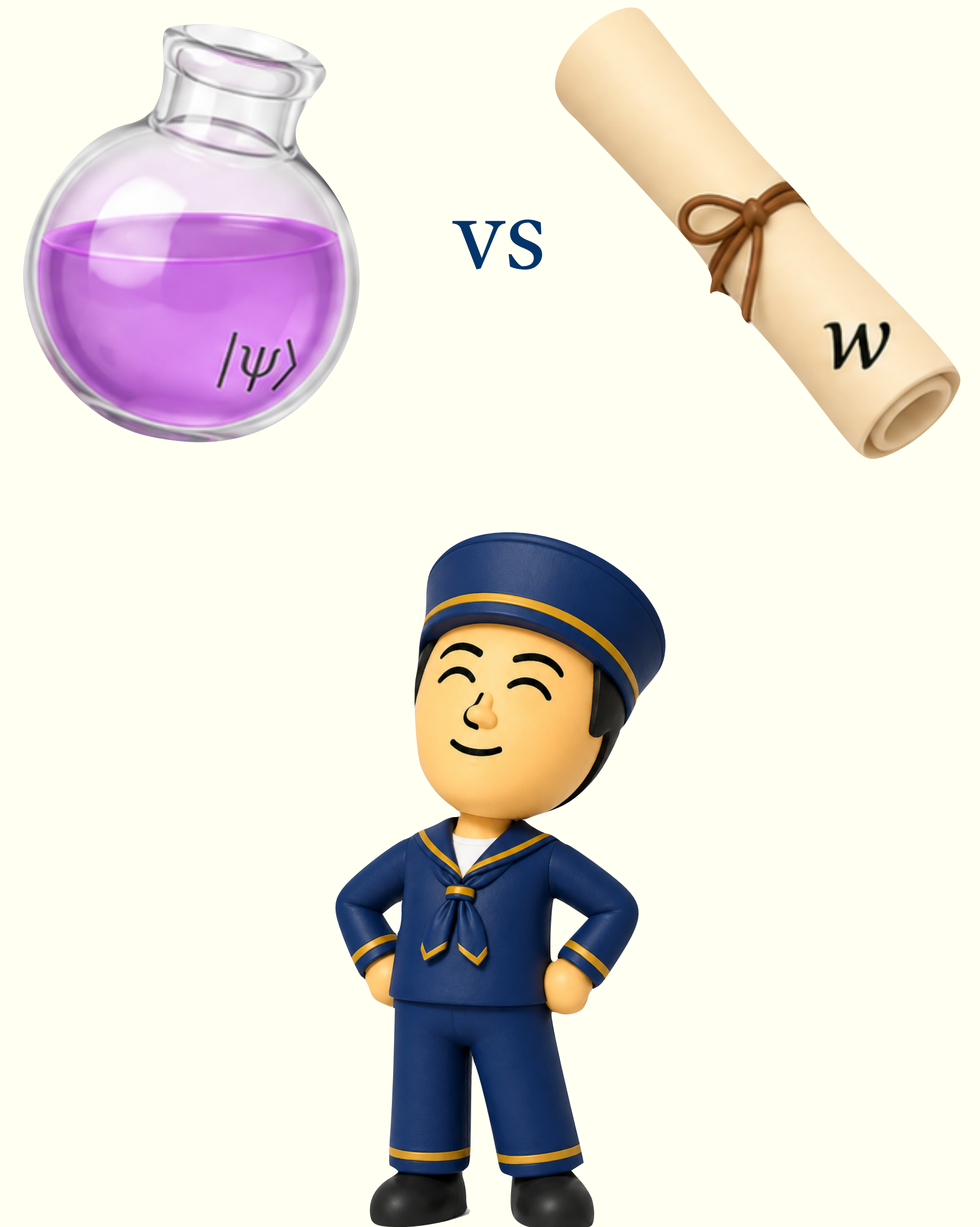


Why is this problem so hard?

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

- One idea is to “guess the witness”, but this works for a quantum witness.

→ The separation should have some kind of interesting structure that makes it so the verifier needs to do more than measure their proof in the standard basis!



Why is this problem so hard?

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

- One idea is to “guess the witness”, but this works for a quantum witness.

→ The separation should have some kind of interesting structure that makes it so the verifier needs to do more than measure their proof in the standard basis!

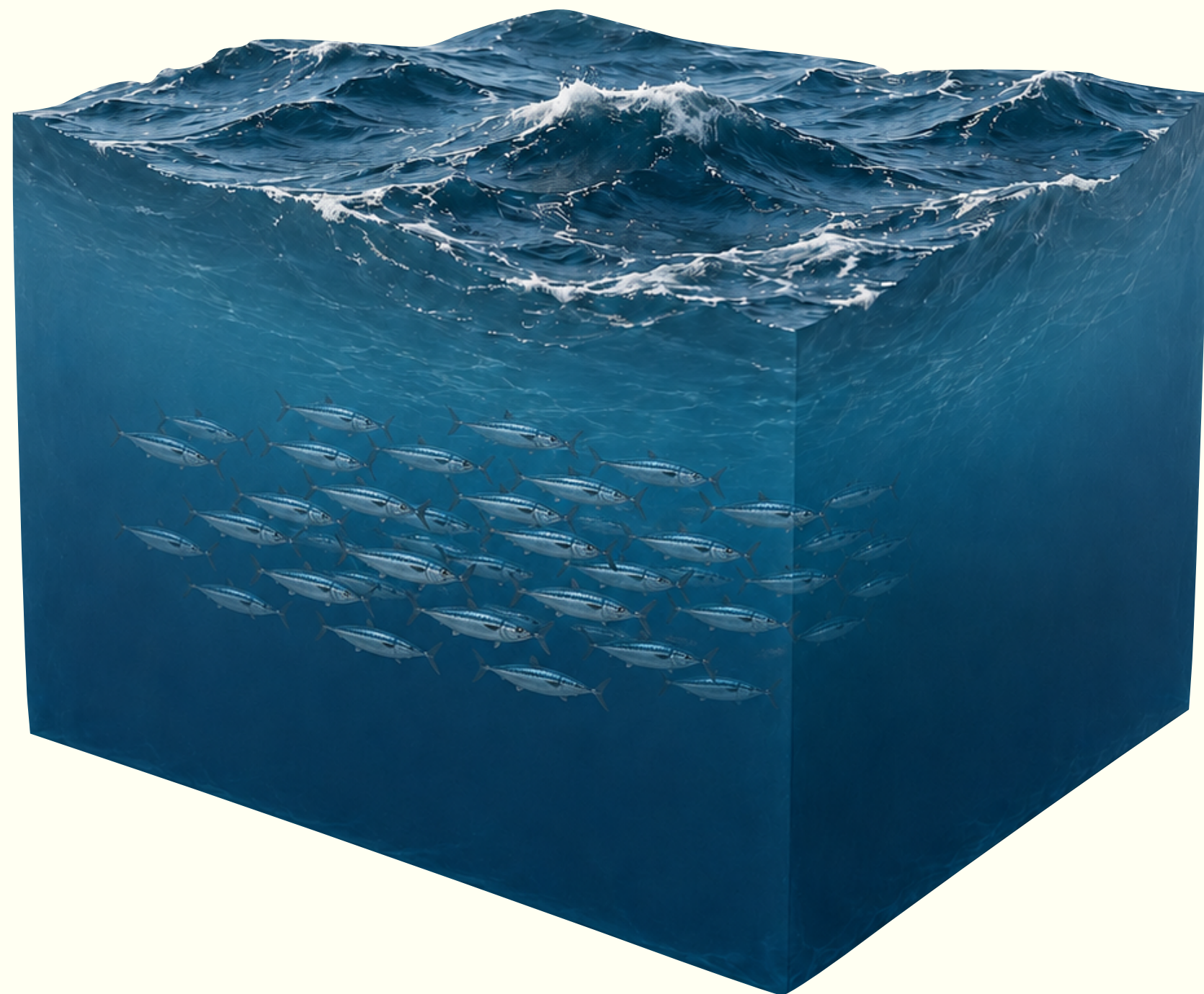
- Quantum algorithms are really good at detecting global structure, so it's even hard to rule out BQP algorithms for some candidate oracle separations!



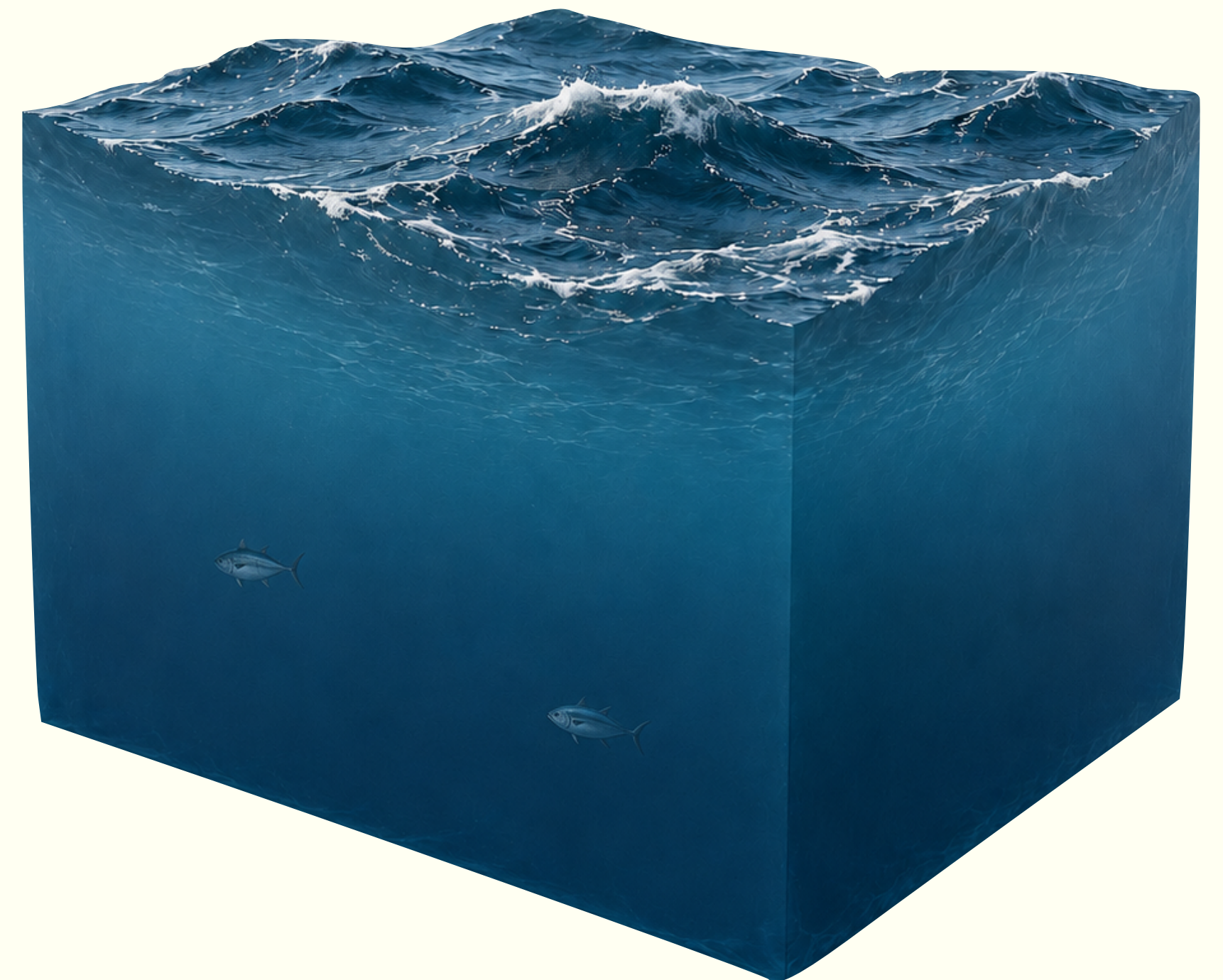
A hard problem for QCMA



A hard problem for QCMA



VS



A new approach for ruling out QCMA

In slightly more detail, we are considering the following problem:

Input: Oracle access to a set S (where fish are), and size ℓ (huge!)

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

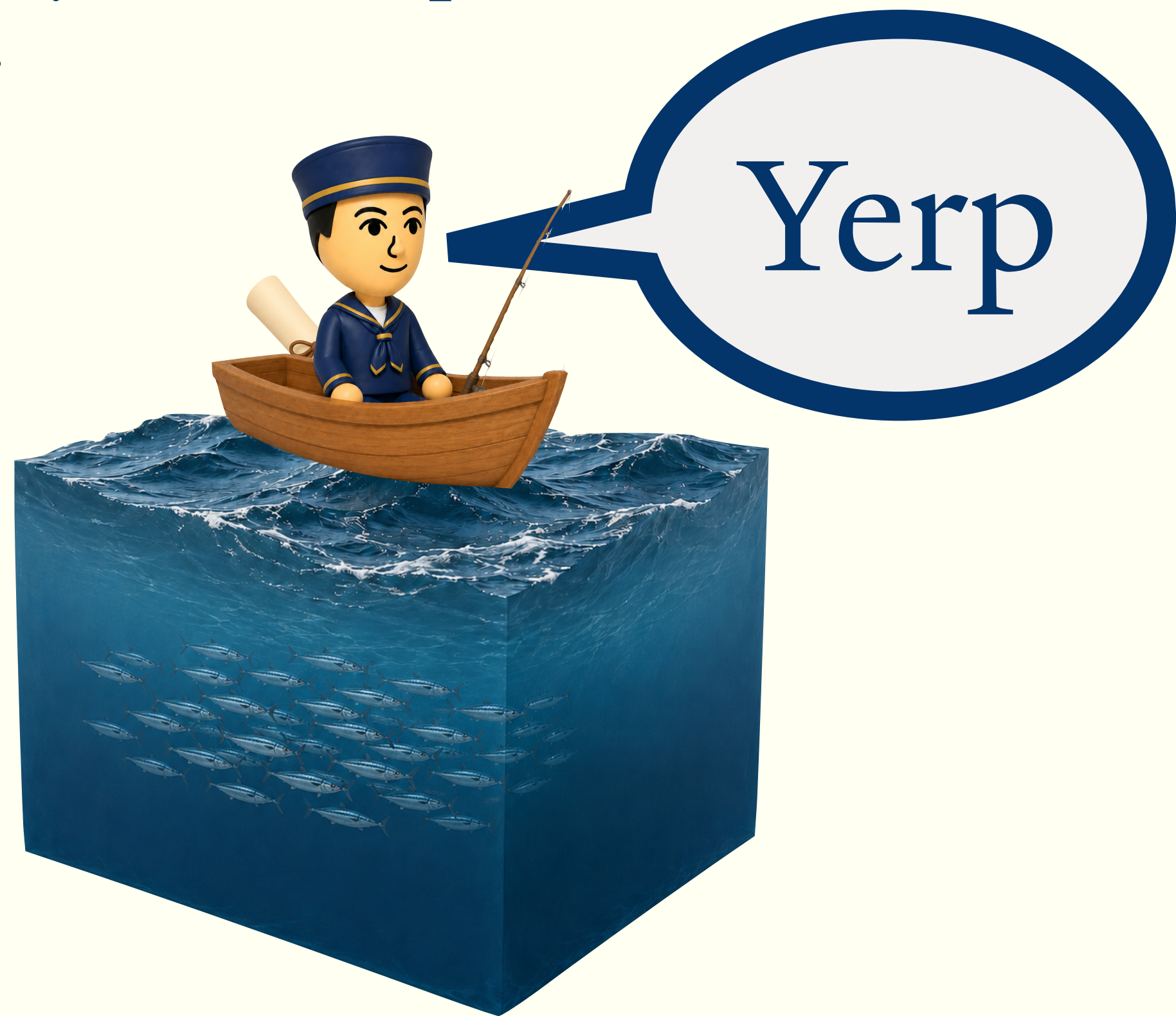


VS



A new approach for ruling out QCMA

Assume for the sake of contradiction that the sailor (with classical witness) would successfully solve this problem, then he definitely distinguishes between many fish and no fish...



A new approach for ruling out QCMA

Assume for the sake of contradiction that the sailor (with classical witness) would successfully solve this problem, then he definitely distinguishes between many fish and no fish...



If the sailor never catches a fish, they would not distinguish between the two cases!

A new approach for ruling out QCMA

Imagine “simulating” the problem in a pool: Let the sailor move around as if they were in the ocean, but any time they fish they won’t catch anything.



A new approach for ruling out QCMA

Imagine “simulating” the problem in a pool: Let the sailor move around as if they were in the ocean, but any time they fish they won’t catch anything.



Because they (by assumption) distinguished between the two cases, if we look at where they want to fish, we will likely catch a fish! Say they fished at position $x_1 \dots$

A new approach for ruling out QCMA

Now, by assumption the should distinguishes between many fish and one fish in the spot he just told us to fish in...



If the sailor doesn't find **another** fish in a different spot, they can't distinguish them!

A new approach for ruling out QCMA

Now simulate this problem in a pool: Let the sailor move around as if they were in the ocean, but any time they fish they won't catch anything, **unless they fish at x_1** .



If we look at where they want to fish, we will likely catch a fish in a different location! Say they fished at position x_2 ...

A new approach for ruling out QCMA

We can keep repeating this process: Simulate the sailor in a pool to get another position where a fish will be, then re-do the simulation with more fake fish!



A new approach for ruling out QCMA

We can keep repeating this process: Simulate the sailor in a pool to get another position where a fish will be, then re-do the simulation with more fake fish!



Even if the map contains enough information to tell the sailor about v fish...
We can repeat this simulation to learn where $\gg v$ fish are, without even fishing at all!

A new approach for ruling out QCMA

We can keep repeating this process: Simulate the sailor in a pool to get another position where a fish will be, then re-do the simulation with more fake fish!



Contradiction!

Even if the map contains enough information to tell the sailor about v fish...
We can repeat this simulation to learn where $\gg v$ fish are, without even fishing at all!

A new approach for ruling out QCMA

An important thing to note about this proof: It depends on the fact that the witness is “re-usable” between the different simulations!



A new approach for ruling out QCMA

An important thing to note about this proof: It depends on the fact that the witness is “re-usable” between the different simulations!



If (instead) we had a potion that the sailor had to drink, once they drink it on the first round we wouldn't be able to run the second simulation!

Getting back to QMA

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



VS



Getting back to QMA

We just saw that set size verification is outside of QCMA.
Unfortunately, it's also not in QMA [AKKT'20]!



VS



Getting back to QMA

We just saw that set size verification is outside of QCMA.
Unfortunately, it's also not in QMA [AKKT'20]!

Let's look at two ways of putting the problem back in QMA.

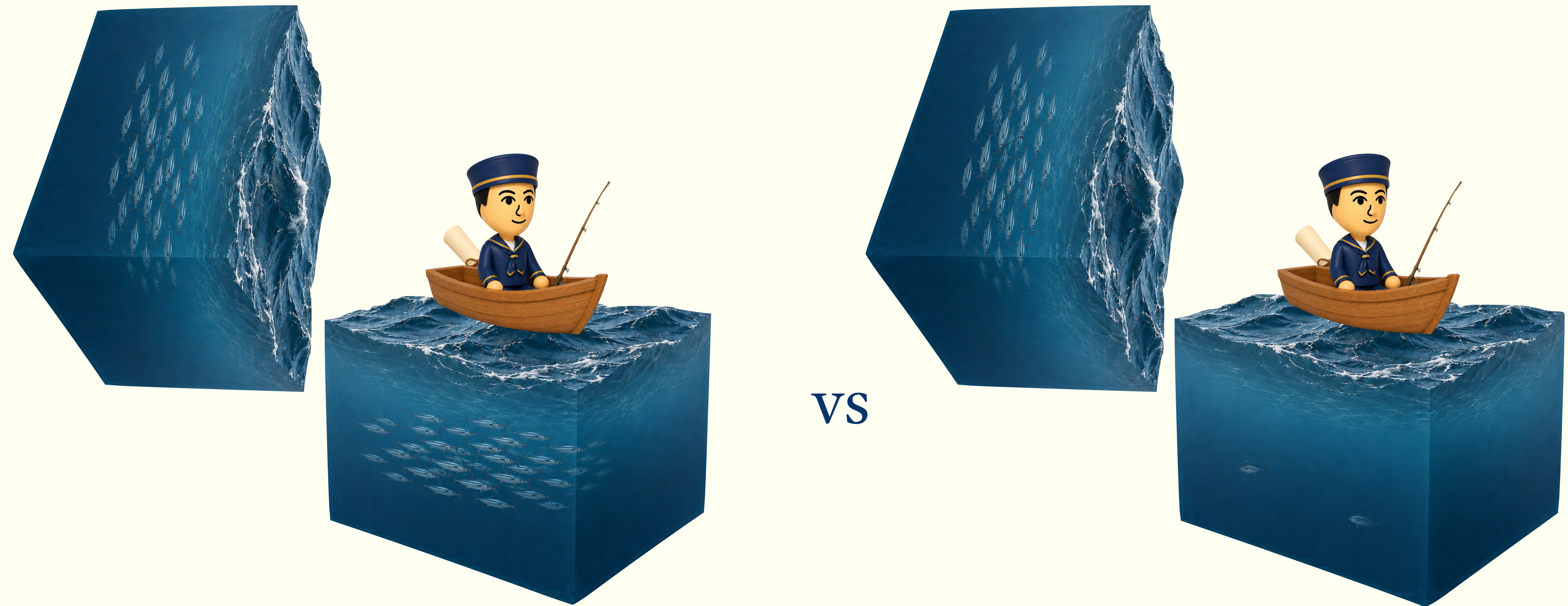


VS



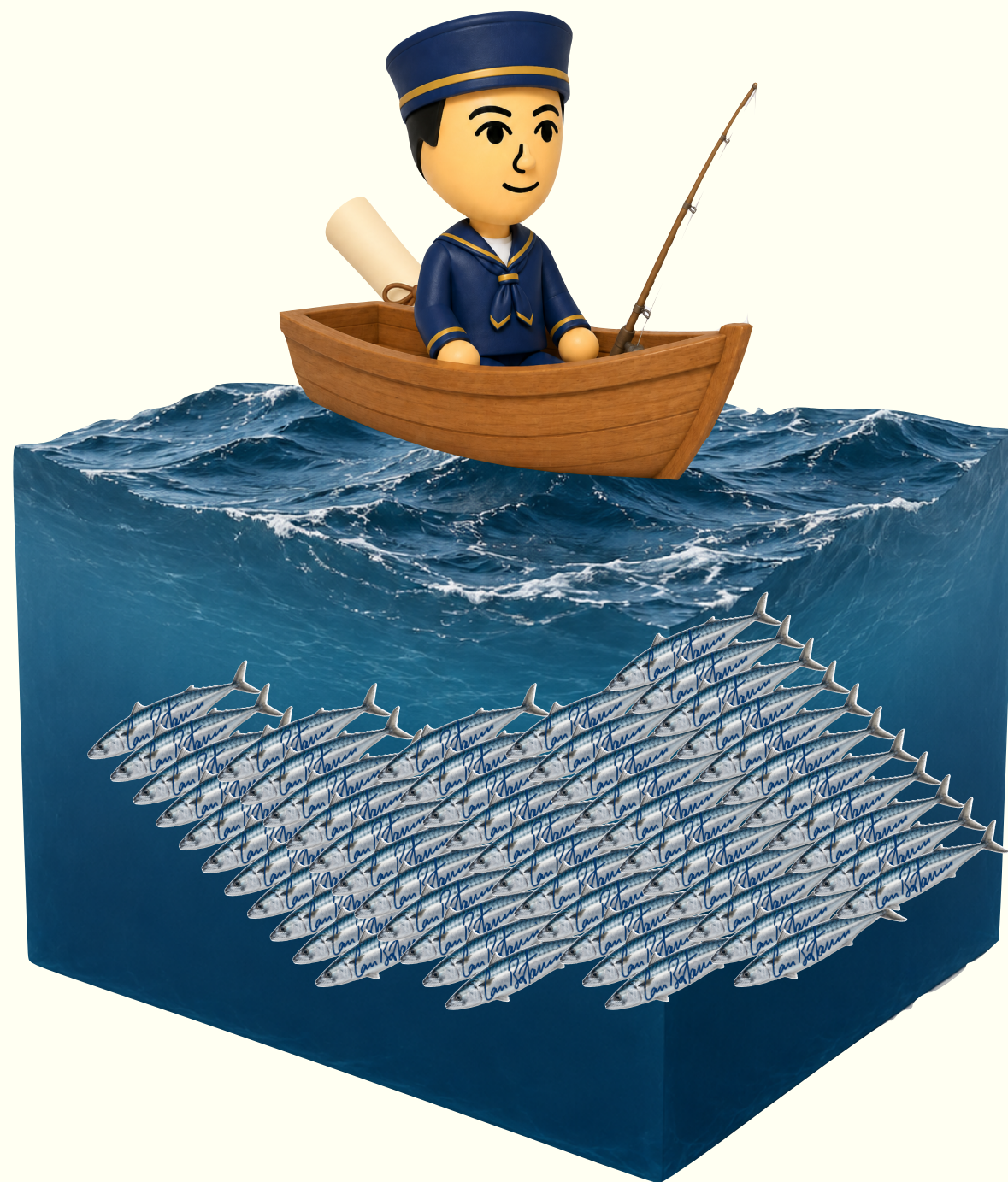
Idea 1: A second oracle [BHNZ'26]

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 [BHV'26]

Another approach will be to add signatures to the sparse set S so that a quantum proof lets someone sign any point, but a classical proof can't. This results in a problem called **code intersection**.

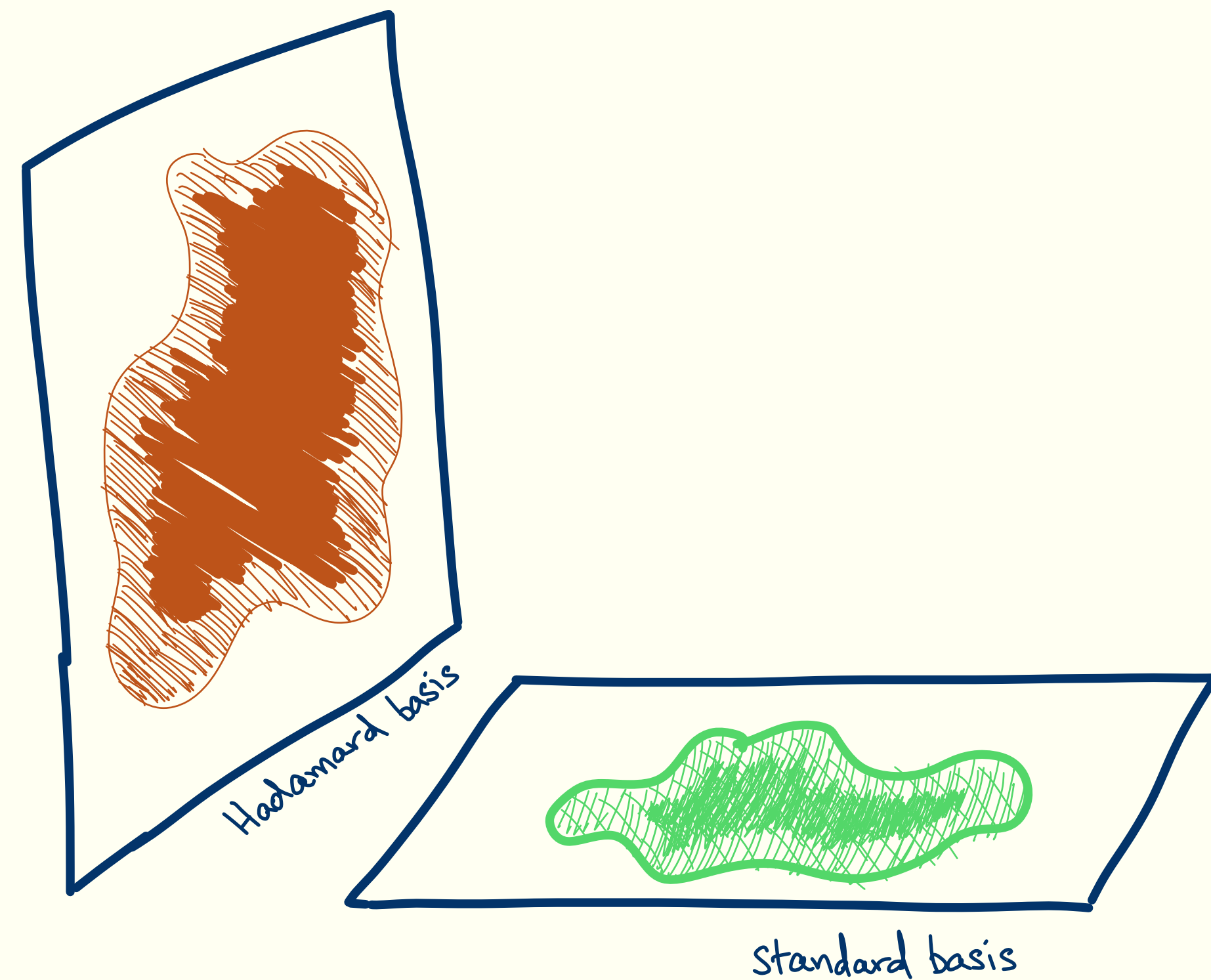


VS



The spectral Forrelation problem

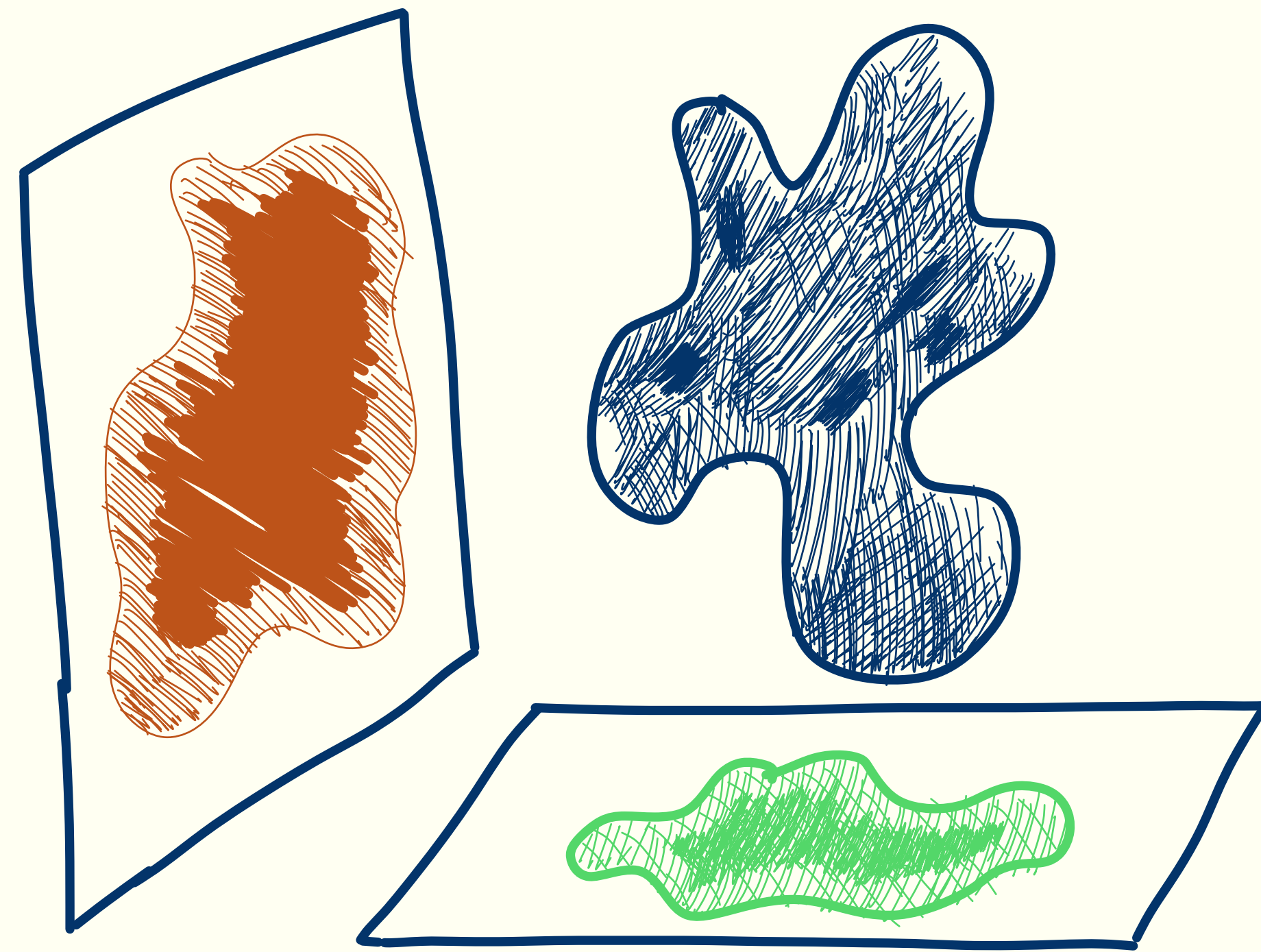
The spectral Forrelation problem is a problem about pairs of sets (S, U) , which we treat as oracles through the set membership functions.



The spectral Forrelation problem

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

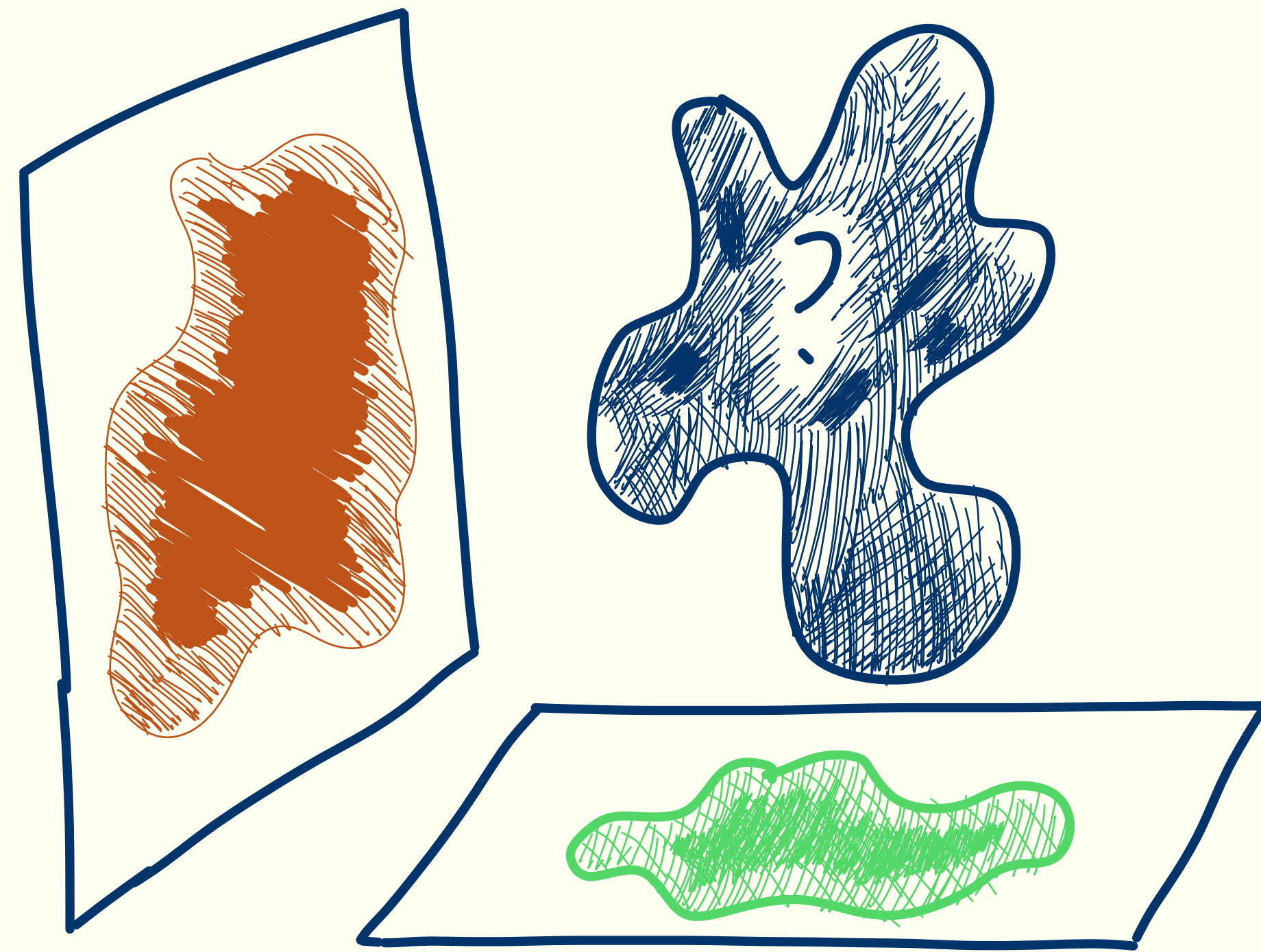
1. $|\psi\rangle$ is mostly supported on S
2. The Fourier transform of $|\psi\rangle$ is mostly supported on U .



The spectral Forrelation problem

Input: Oracle access to two sets (S, U) (via set membership functions)

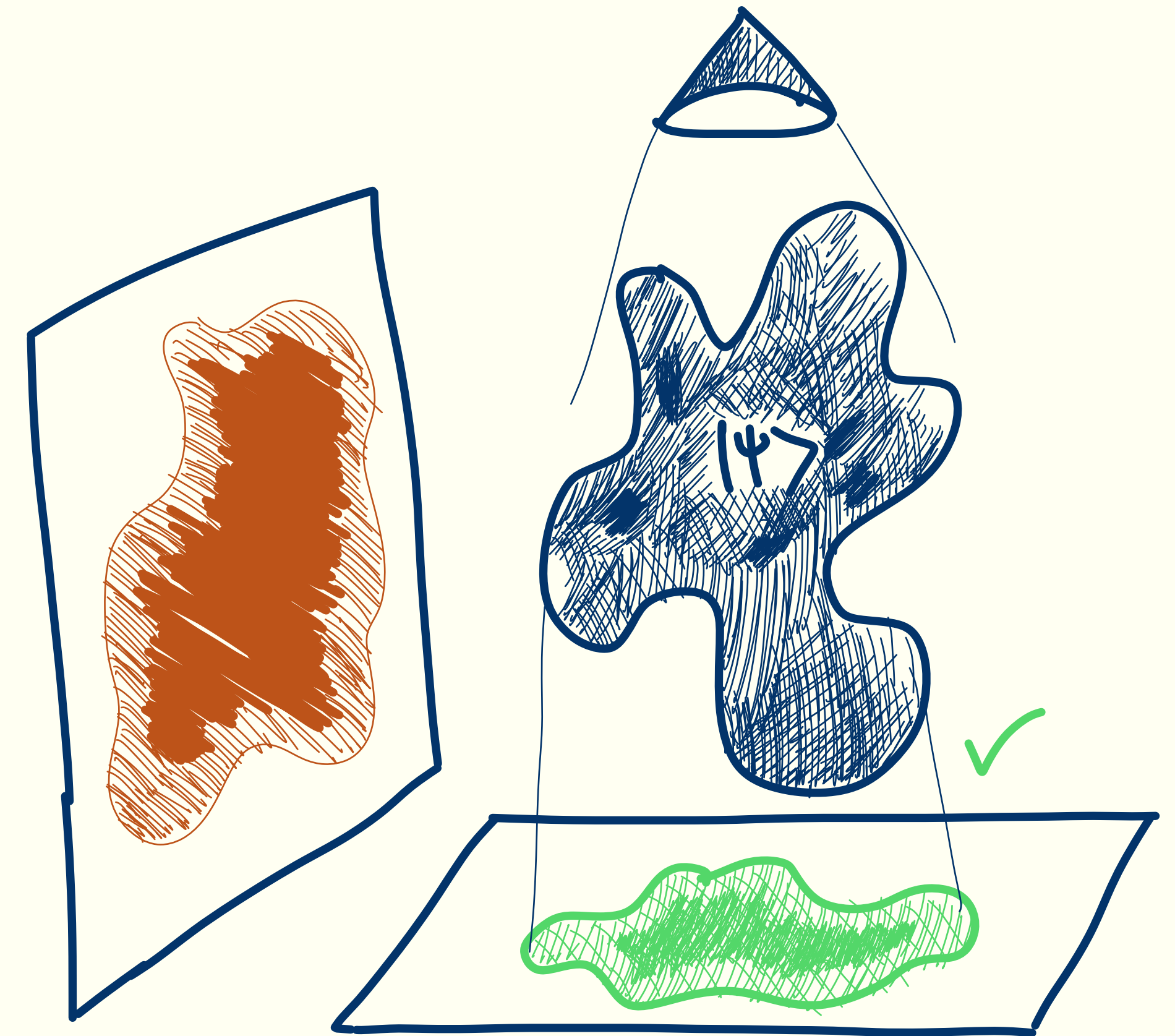
Output: Are they spectrally Forrelated or not?



Spectral Forrelation is in QMA

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

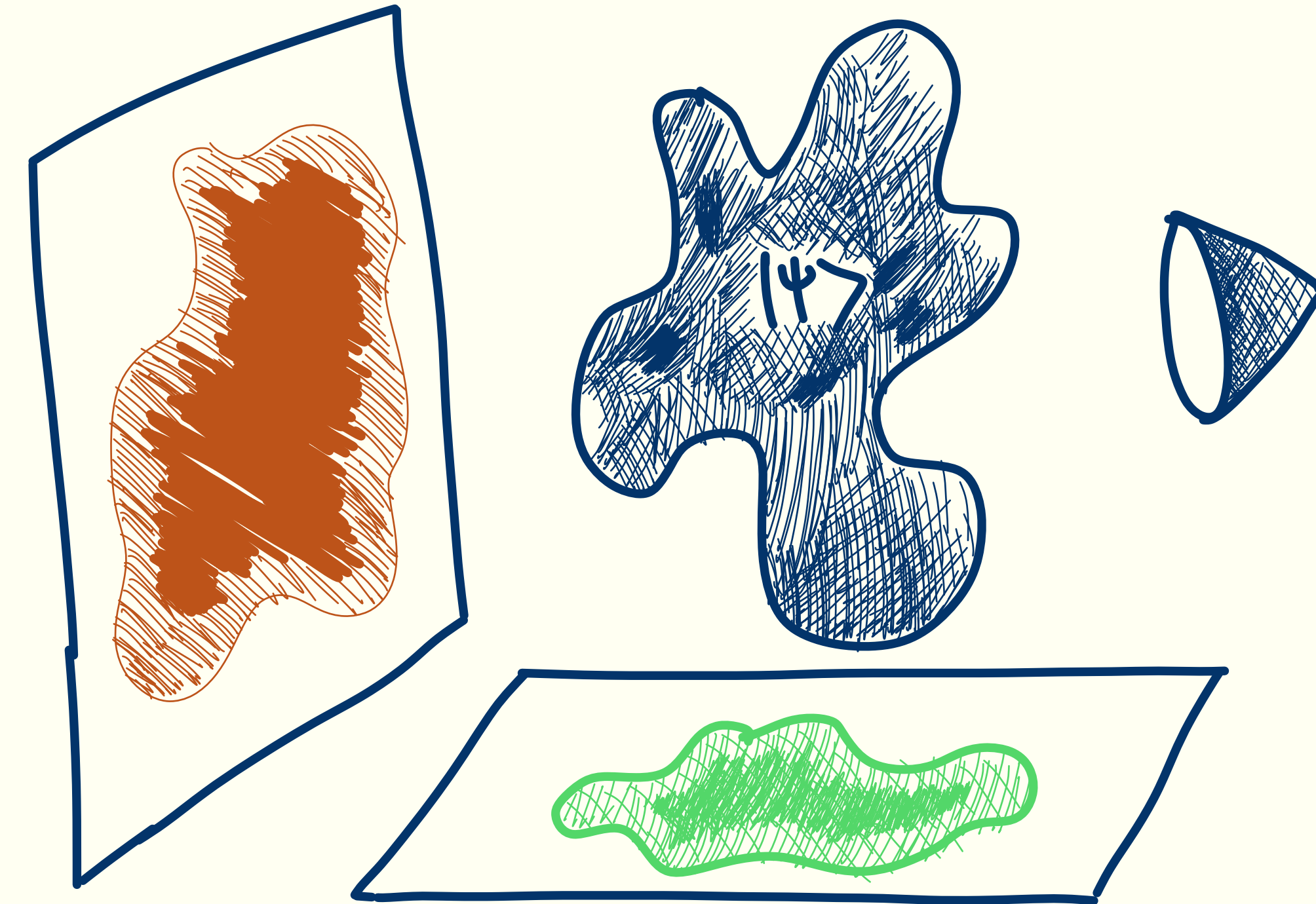
- Check if the state is supported on S , reject if not.



Spectral Forrelation is in QMA

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

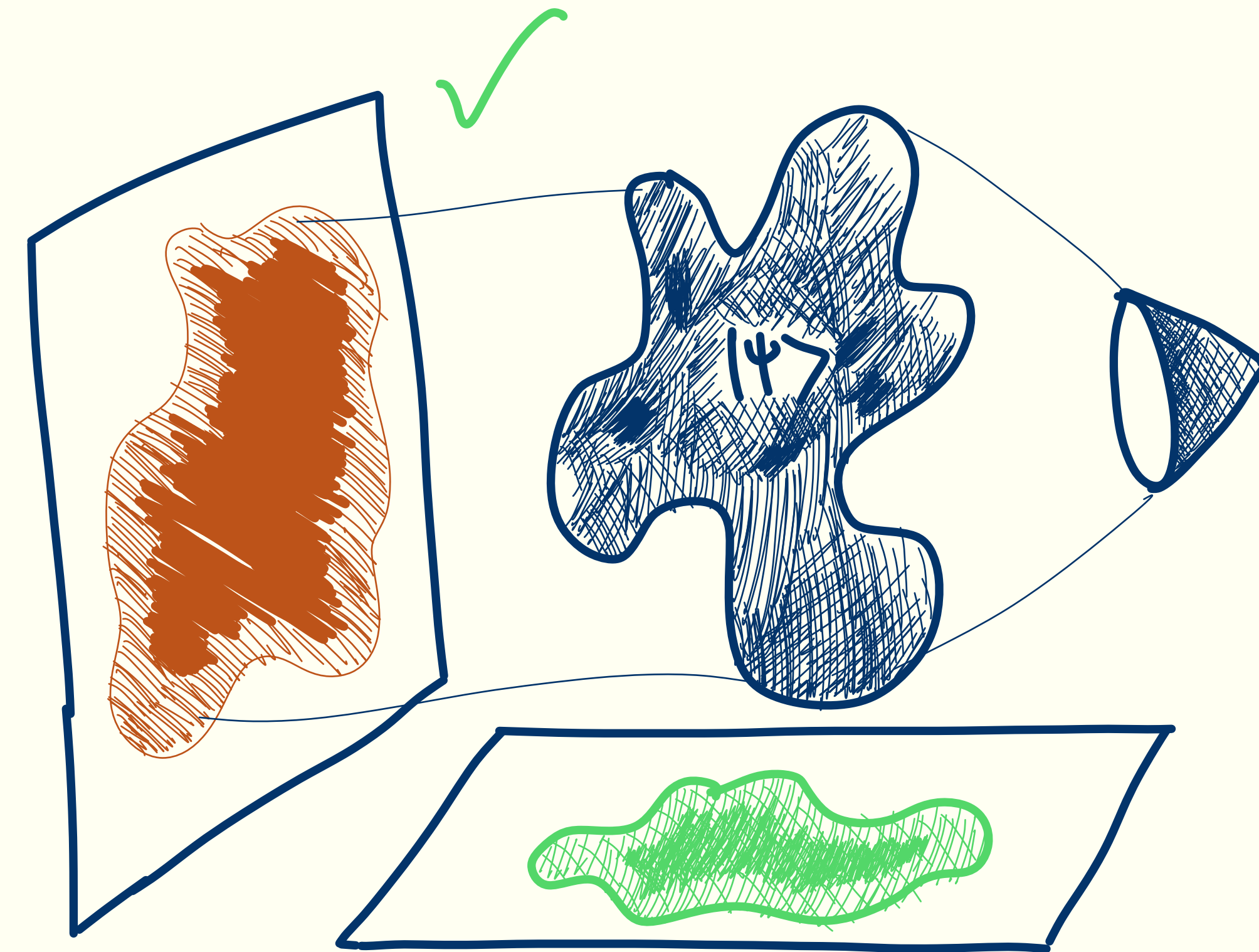
- Check if the state is supported on S , reject if not.
- Fourier transform the state.



Spectral Forrelation is in QMA

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

- Check if the state is supported on S , reject if not.
- Fourier transform the state.
- Check if the state is supported on U , reject if not.
- Accept.



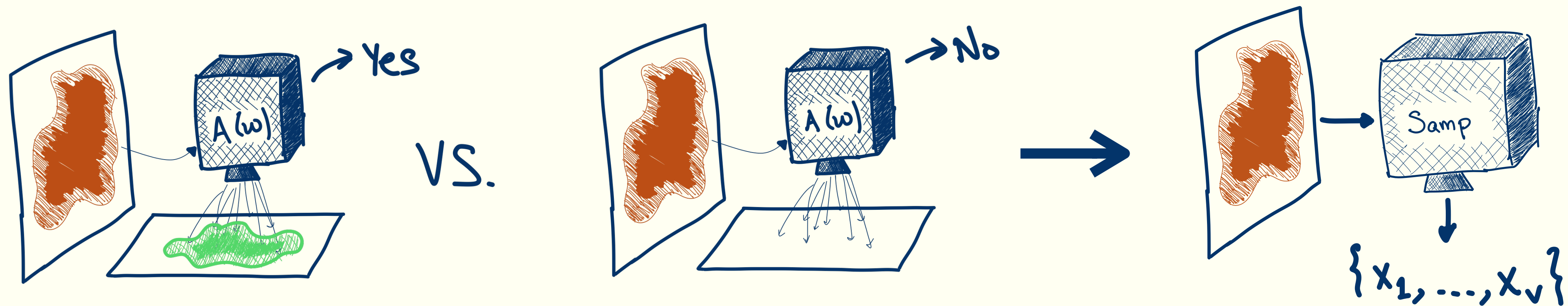
Spectral Forrelation is not in QCMA

We rule out QCMA algorithms for spectral Forrelation in two steps:

Spectral Forrelation is not in QCMA

We rule out QCMA algorithms for spectral Forrelation in two steps:

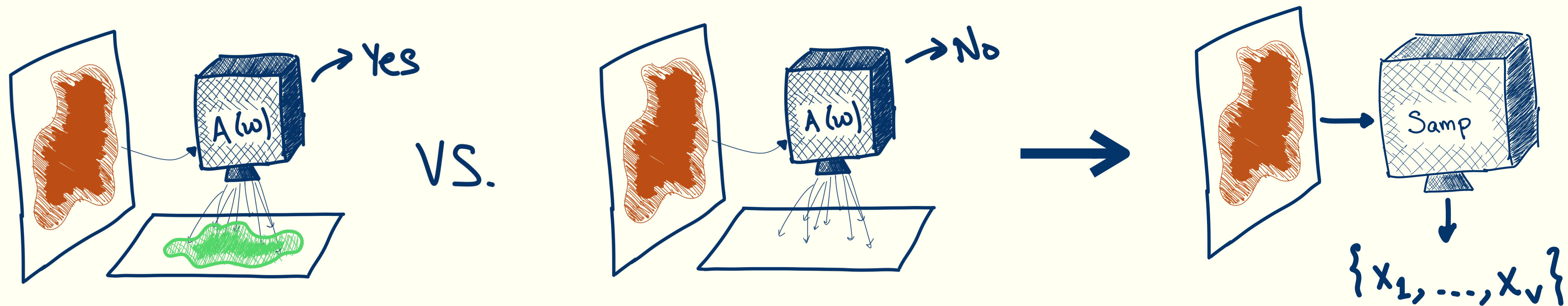
Theorem 1: A QCMA algorithm implies a good sampler that only looks at U .



Spectral Forrelation is not in QCMA

We rule out QCMA algorithms for spectral Forrelation in two steps:

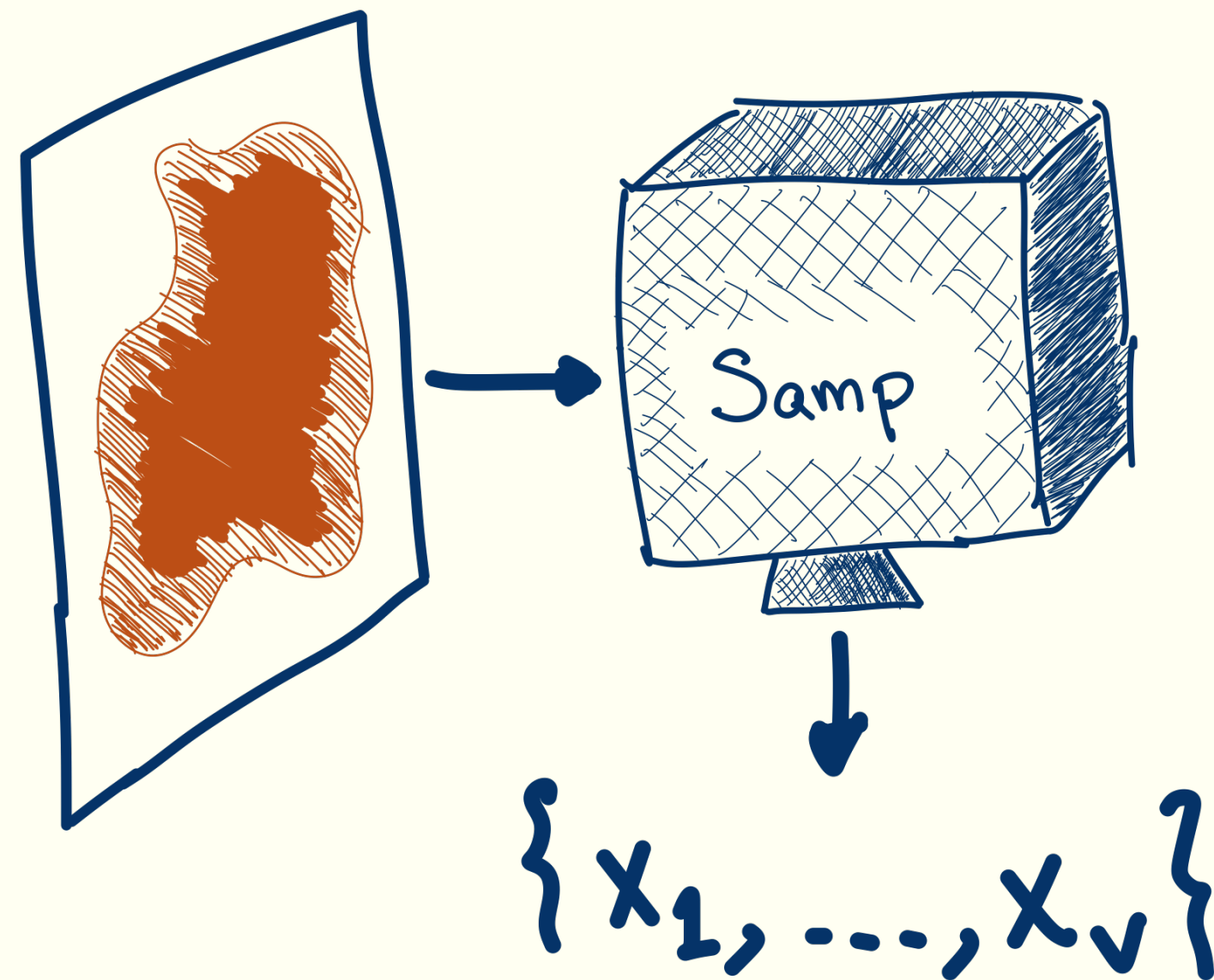
Theorem 1: A QCMA algorithm implies a good sampler that only looks at U .



Theorem 2: A few queries to U can not help you sample points from S .

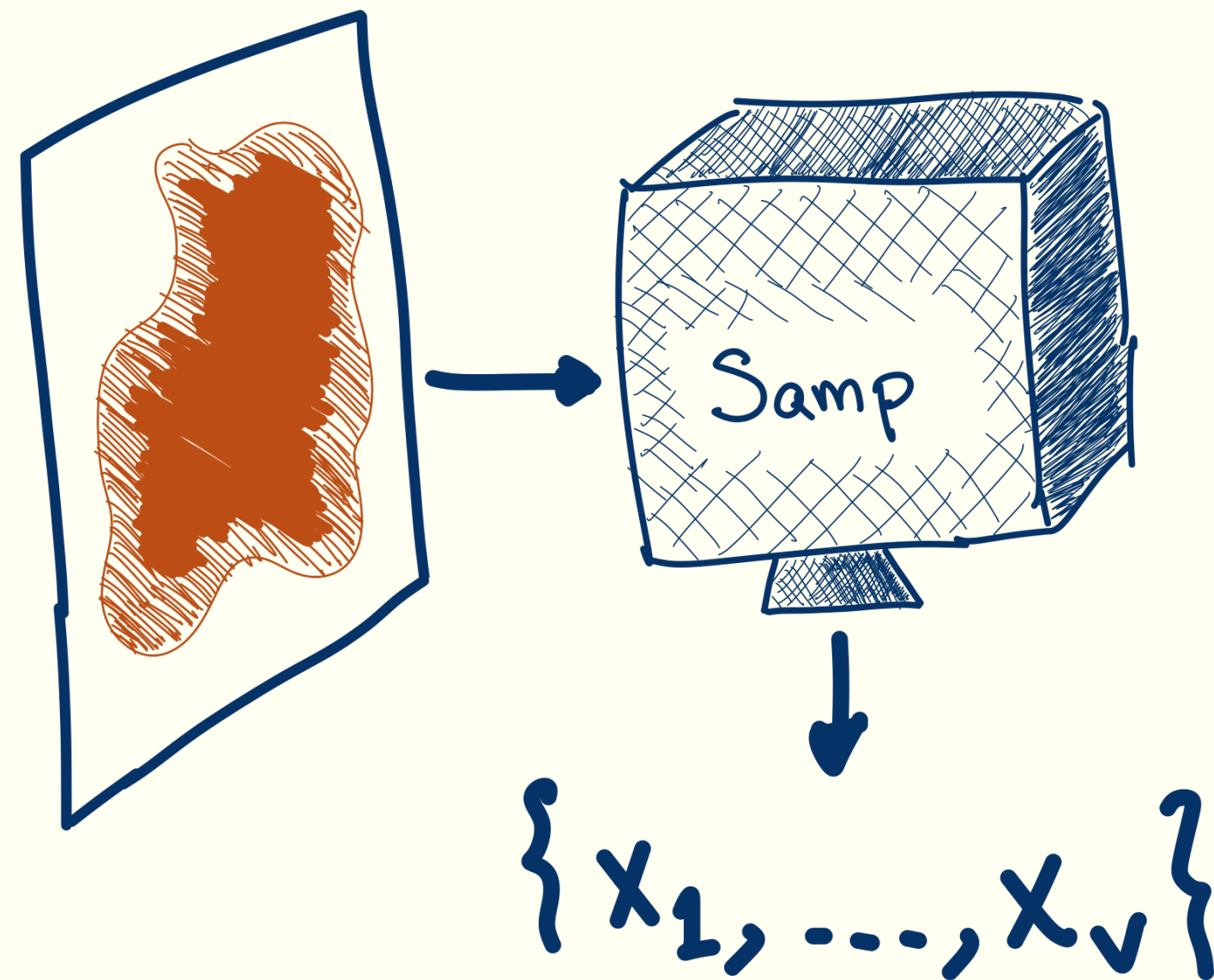
Spectral Forrelation is not in QCMA

Theorem 2: A few queries to U can not help you sample points from S .



Spectral Forrelation is not in QCMA

Theorem 2: A few queries to U can not help you sample points from S .



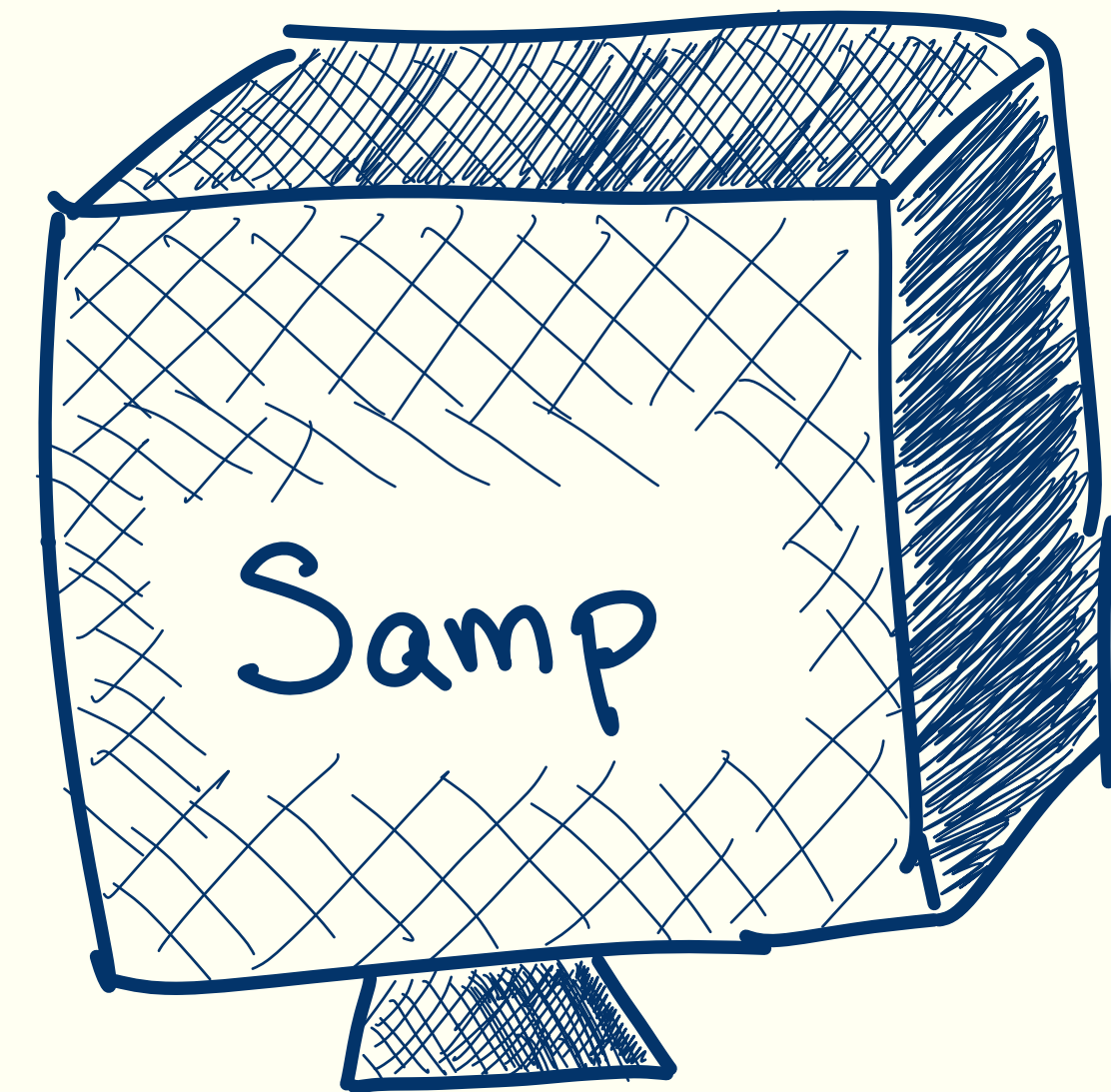
This is more tricky than before: U can be correlated to S .

→ We need to understand what the sampler knows about S when they look at U !

Lazy sampling

Classically, when someone interacts with a random function, we don't write down every entry. Lazy sampling is how we only store what the adversary knows about the function.

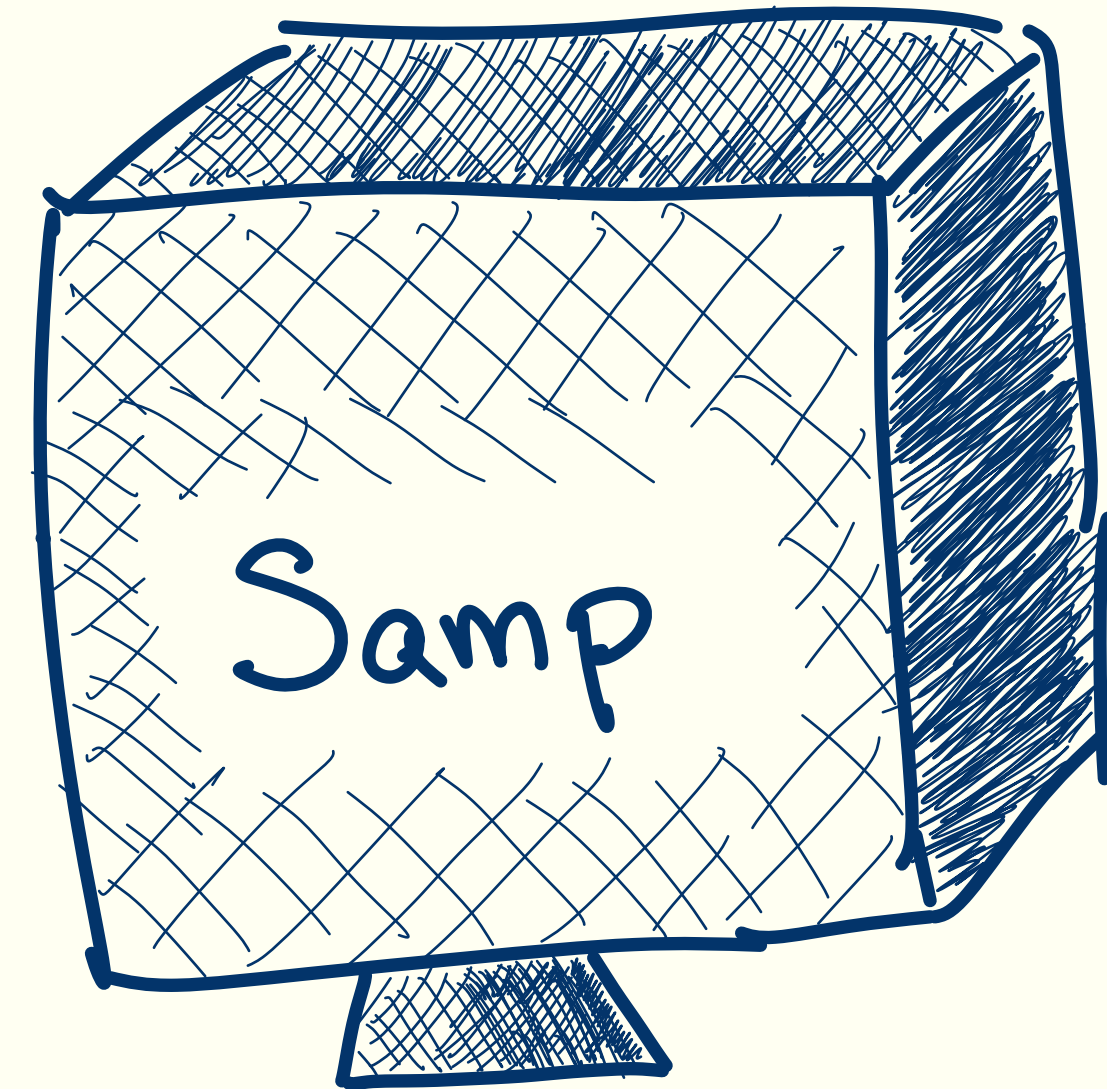
x	$f(x)$
z	1
w	0



Lazy sampling

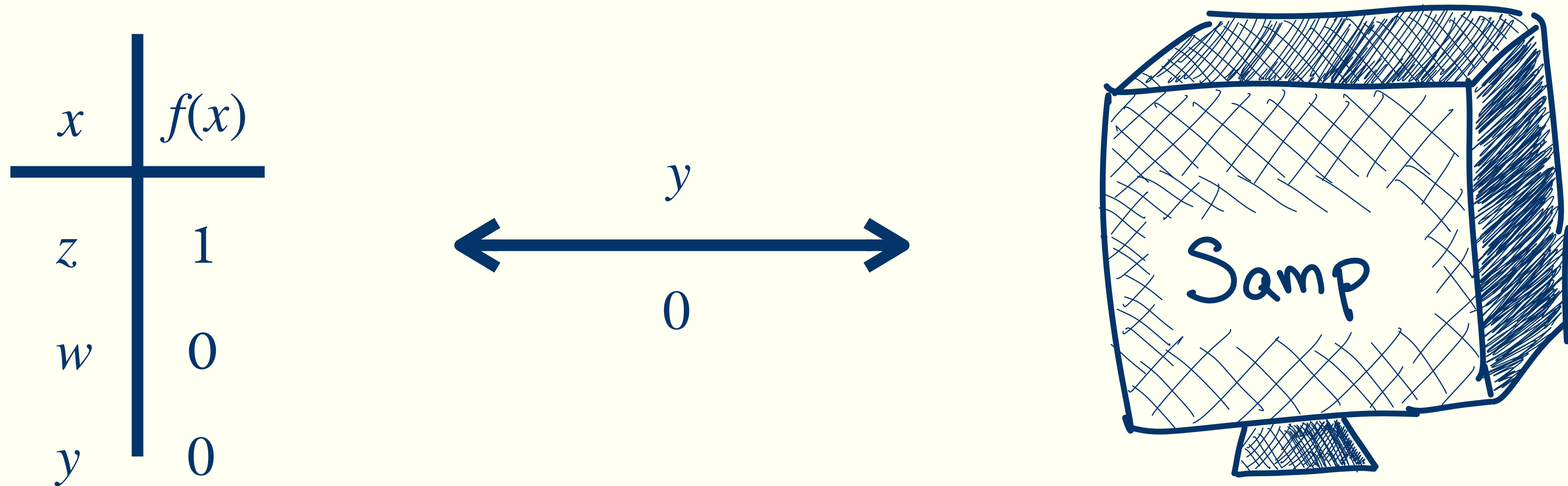
Classically, when someone interacts with a random function, we don't write down every entry. Lazy sampling is how we only store what the adversary knows about the function.

x	$f(x)$
z	1
w	0



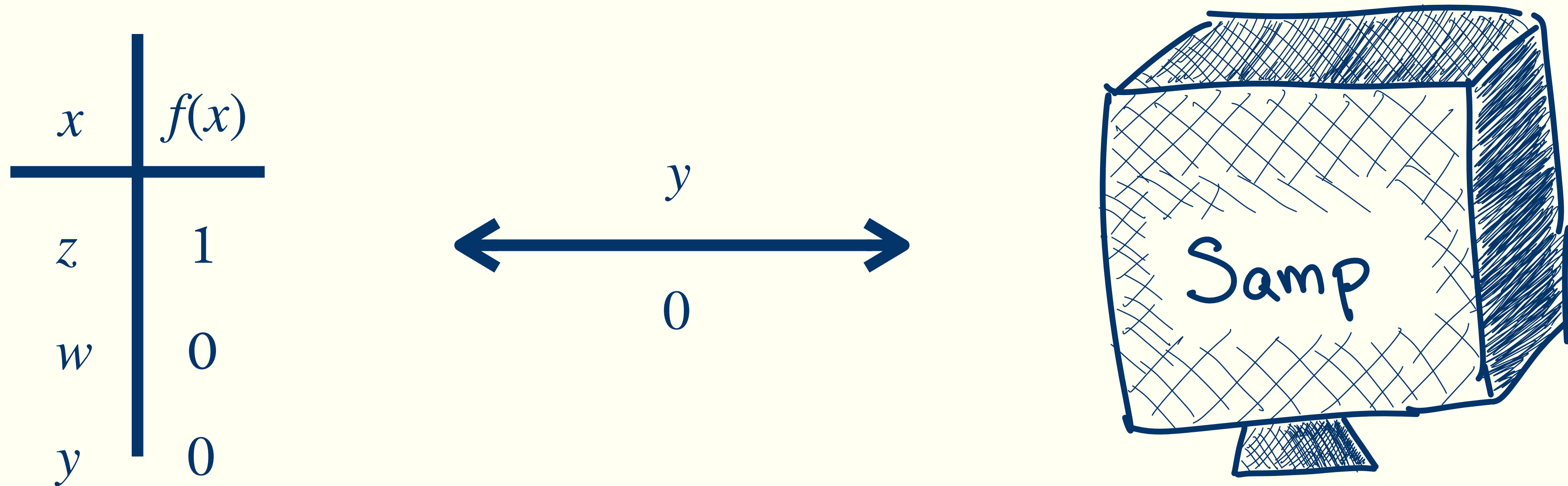
Lazy sampling

Classically, when someone interacts with a random function, we don't write down every entry. Lazy sampling is how we only store what the adversary knows about the function.



Lazy sampling

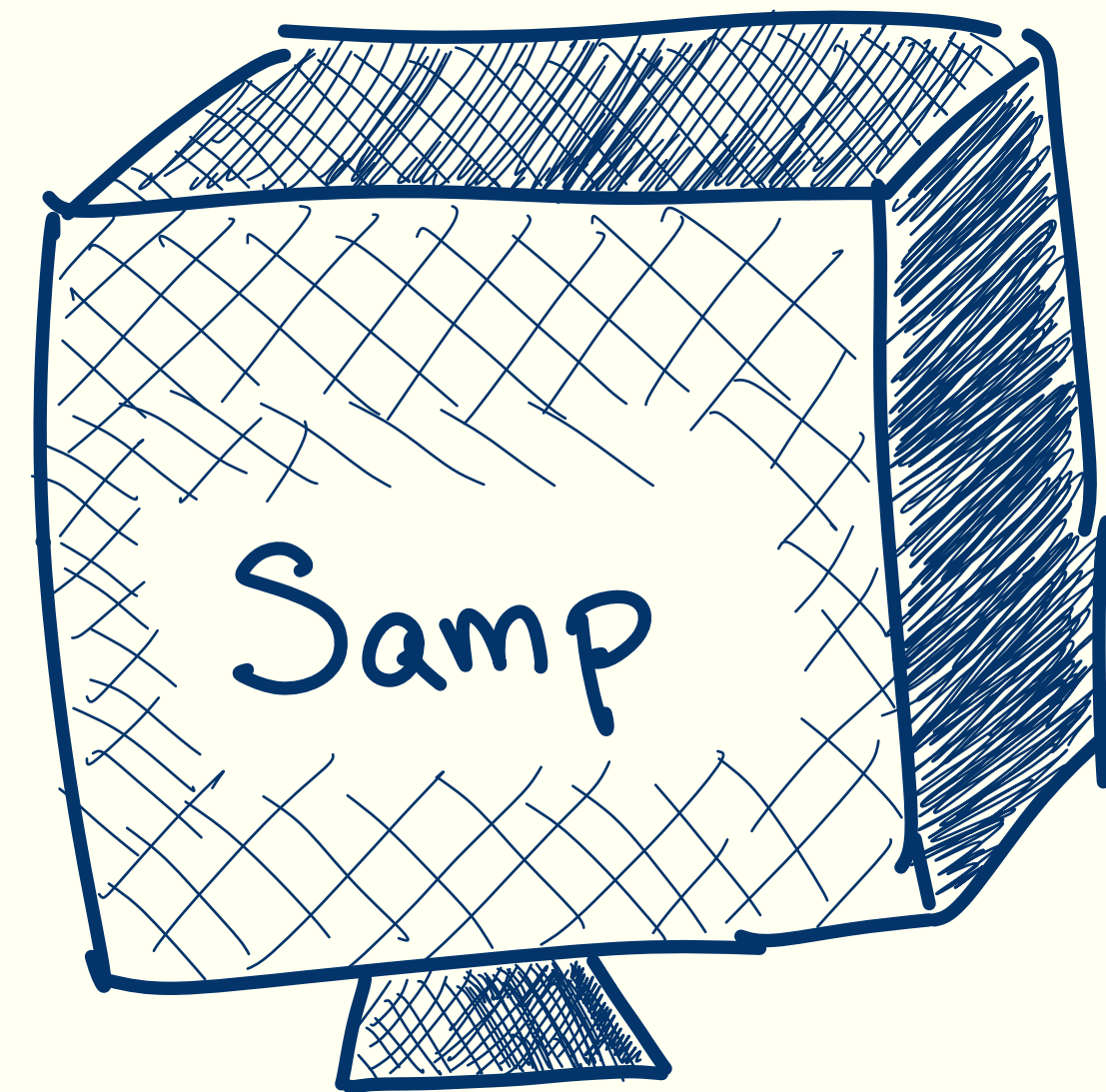
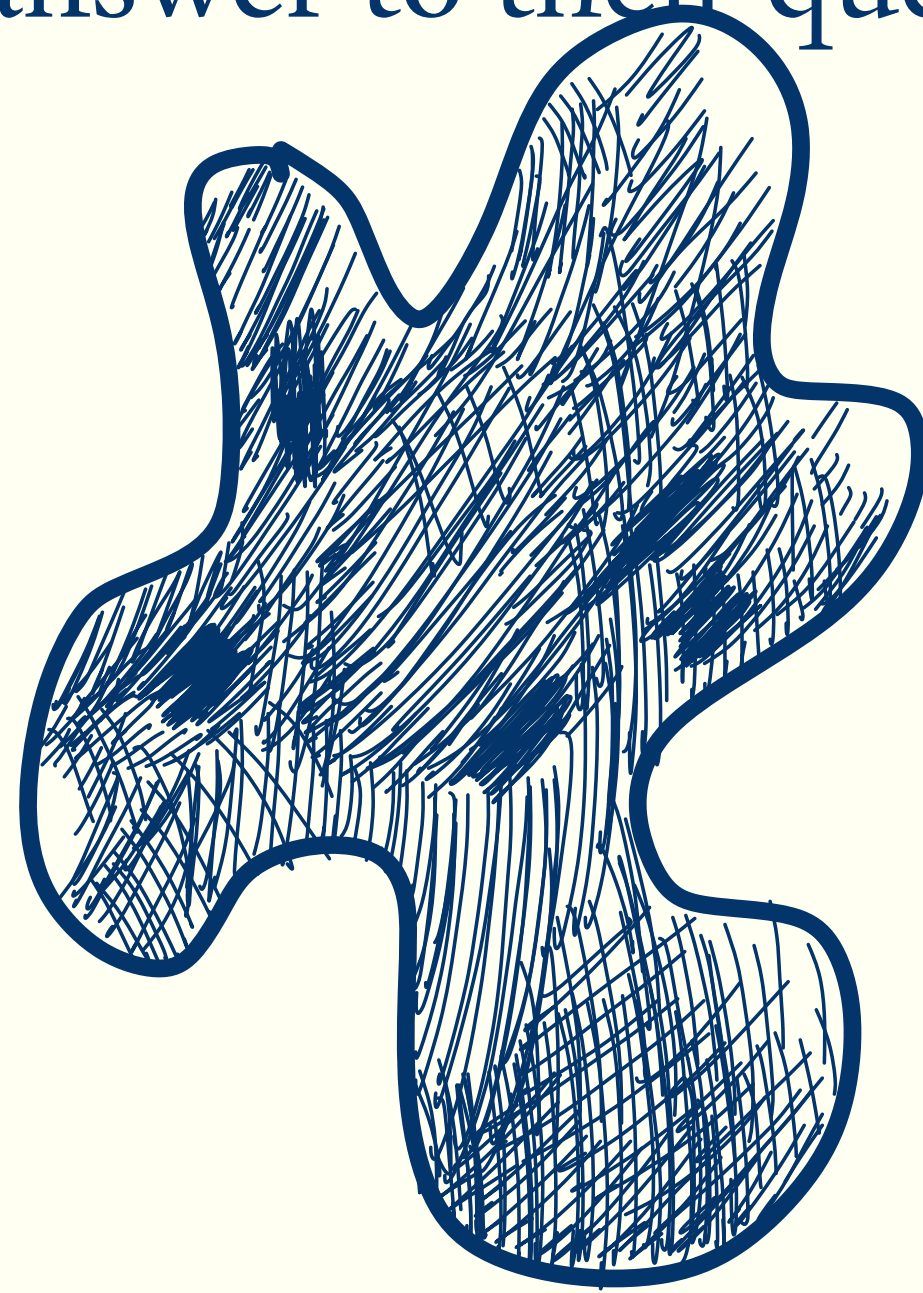
Classically, when someone interacts with a random function, we don't write down every entry. Lazy sampling is how we only store what the adversary knows about the function.



A similar technique exists quantumly, called the compressed oracle technique.

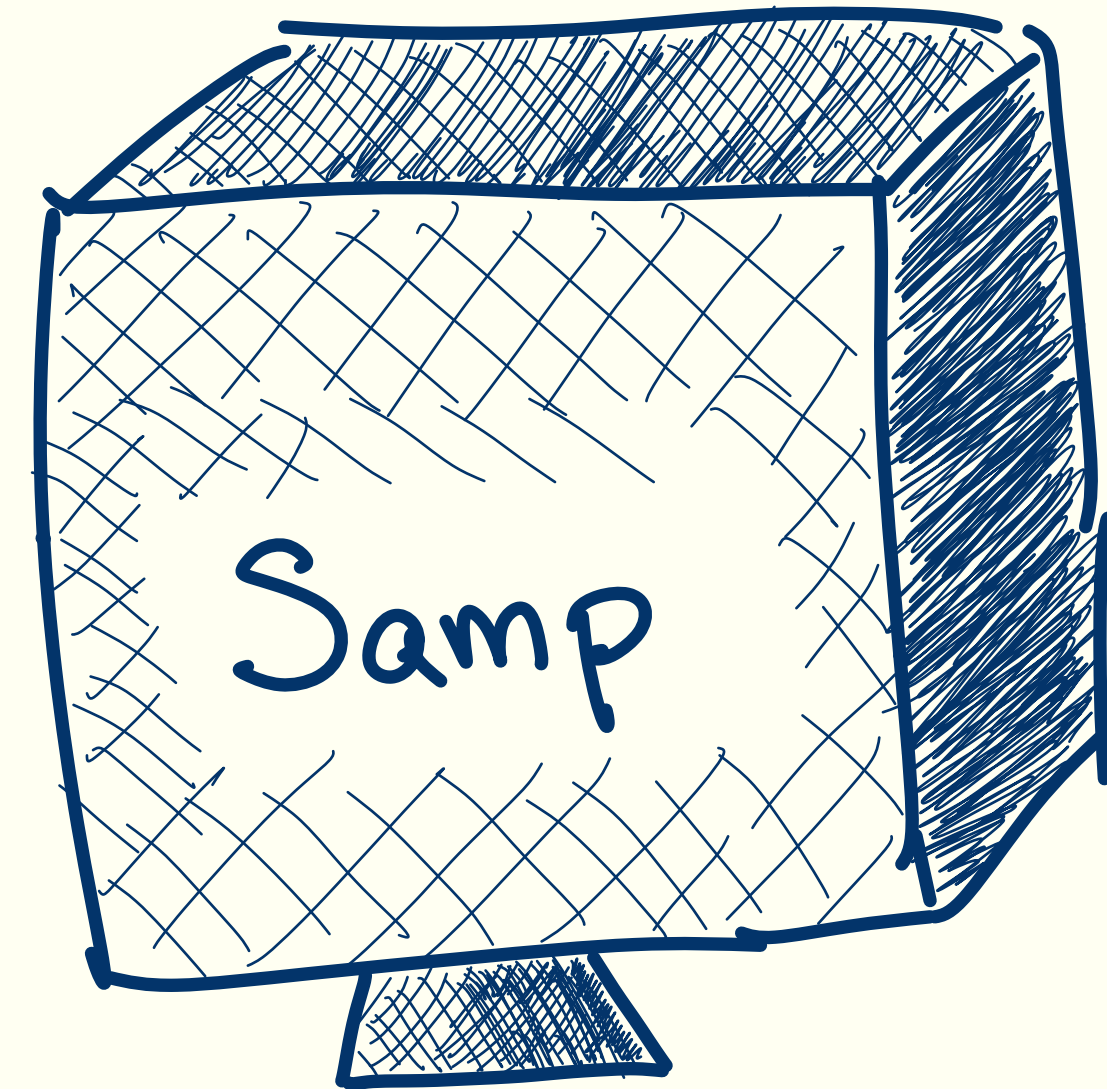
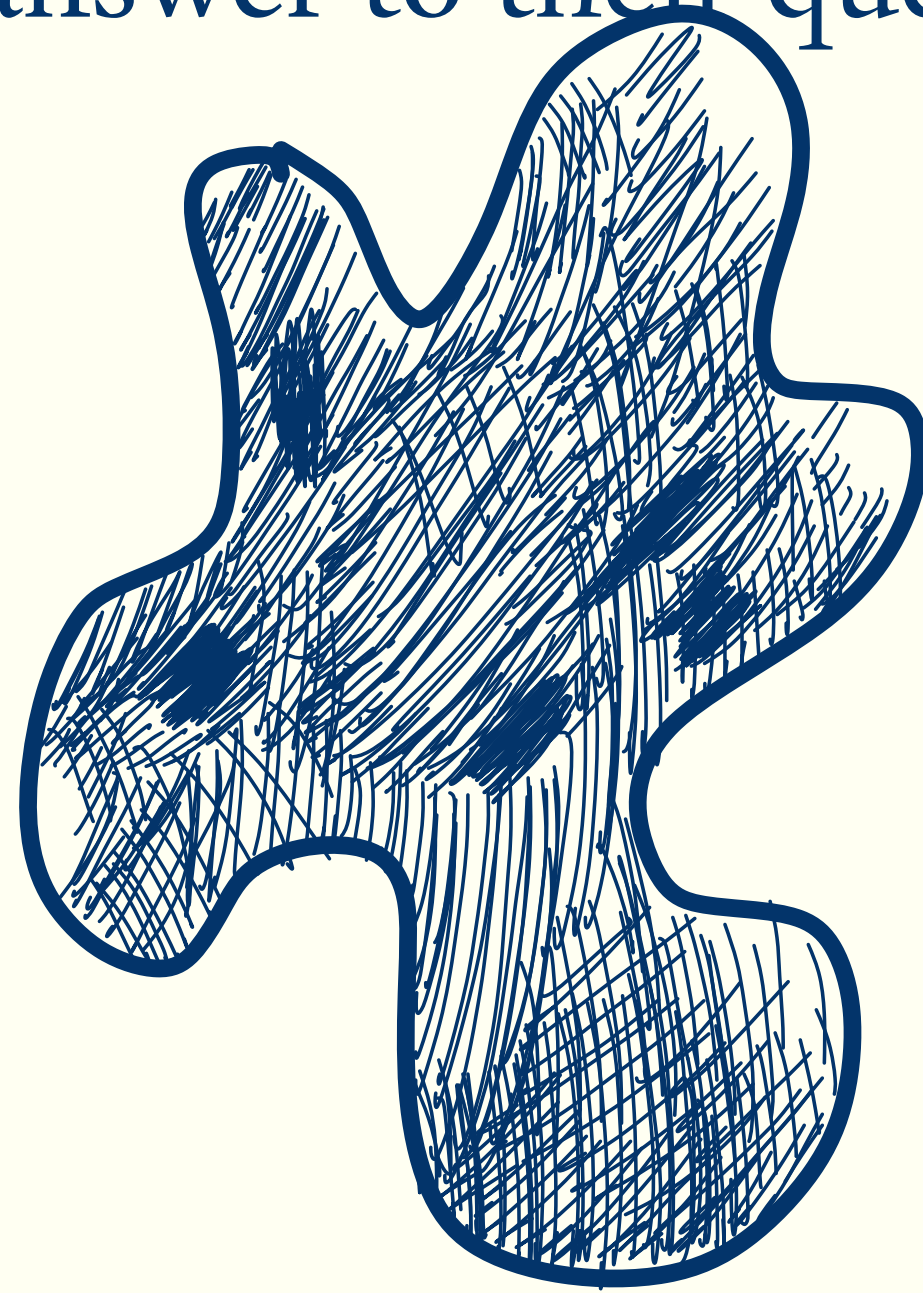
Compressed oracles

In the compressed oracle technique, the sampler interacts with an unknown quantum system. When they query, we instead “condition” on the system being consistent with the answer to their query!



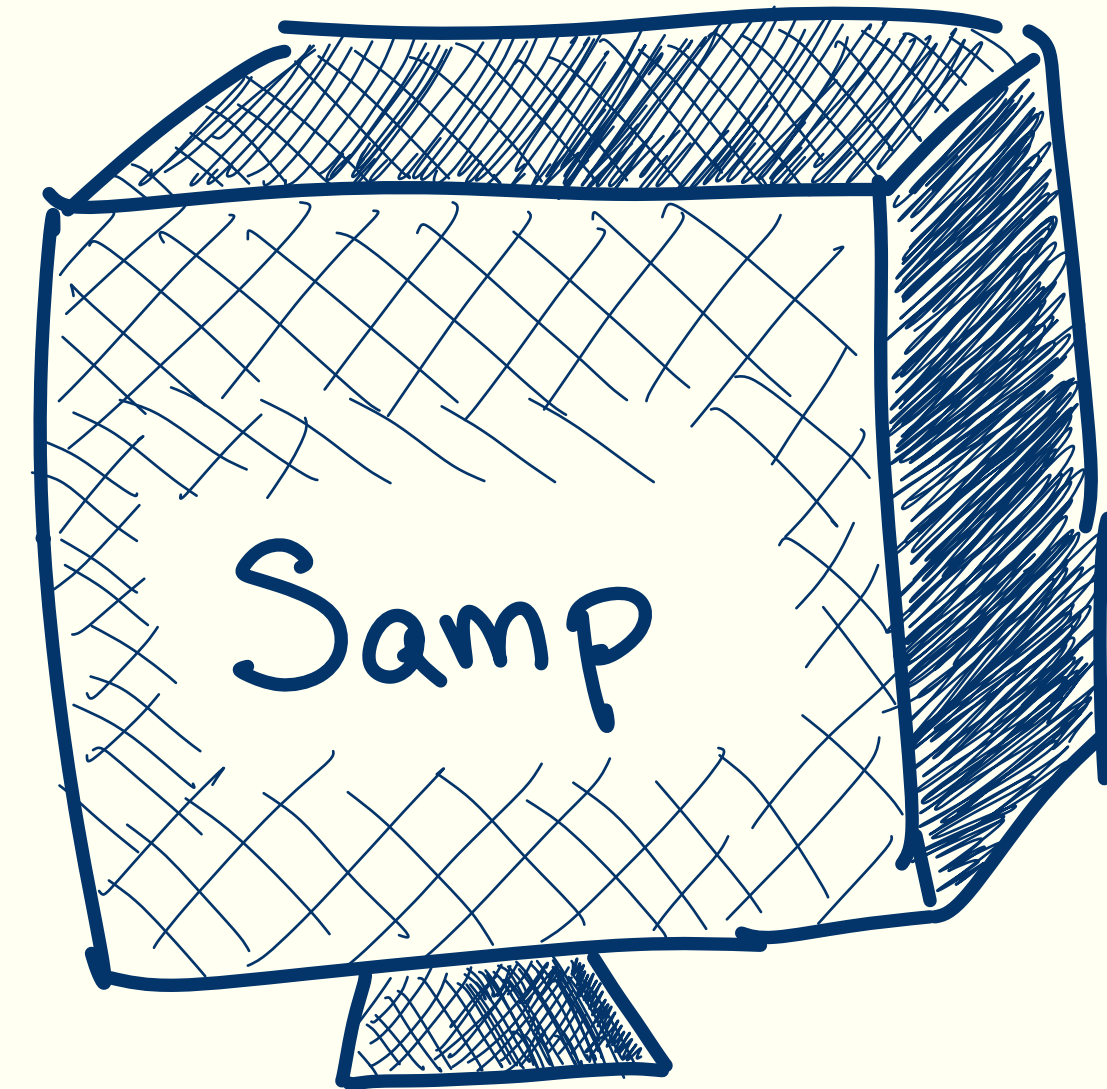
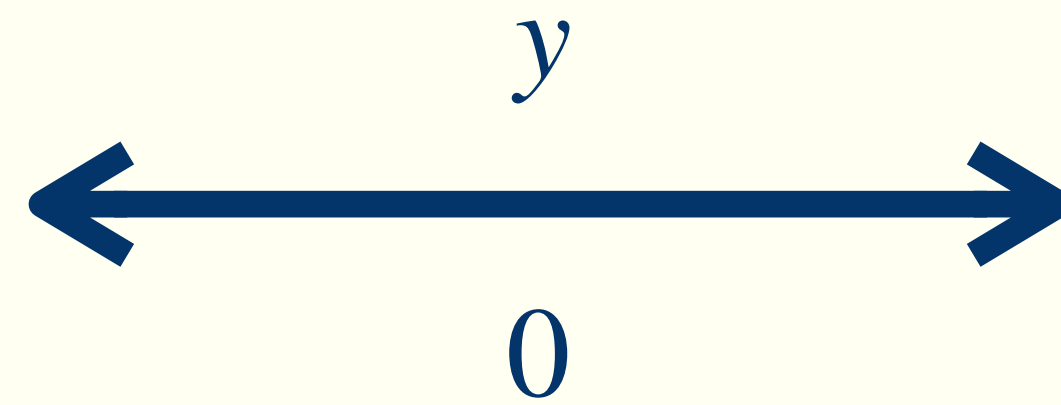
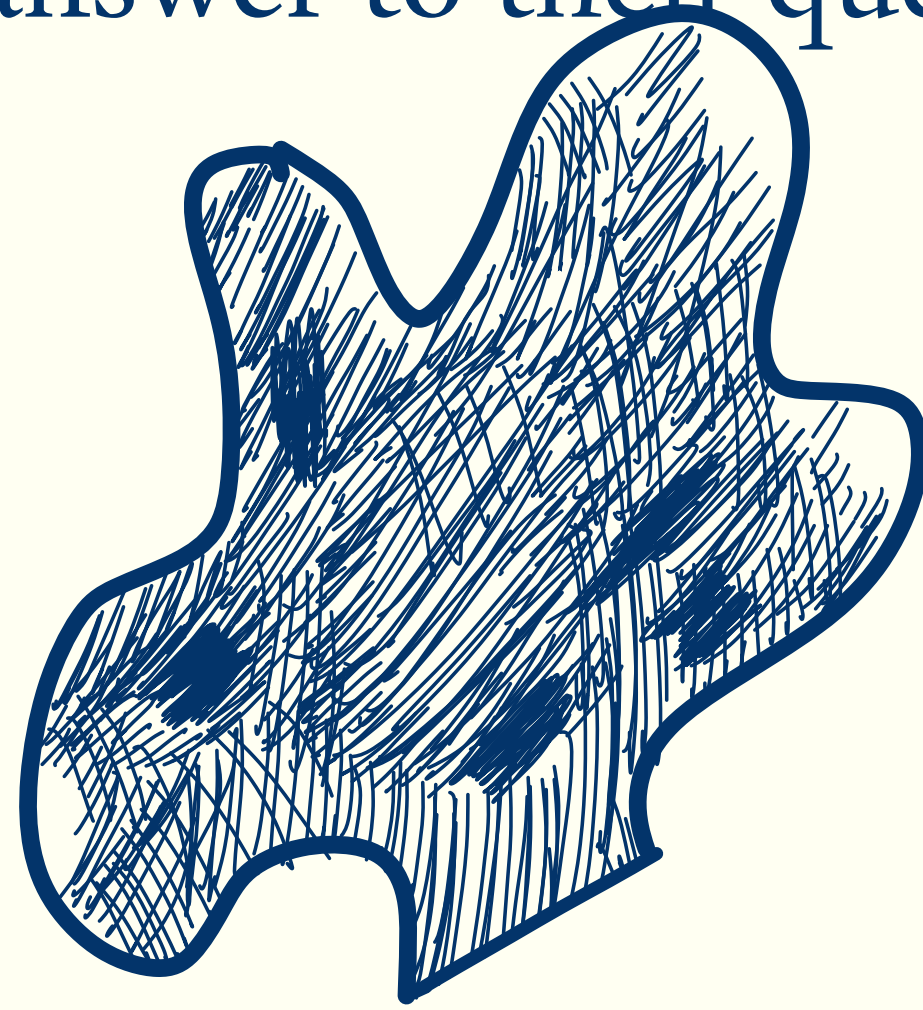
Compressed oracles

In the compressed oracle technique, the sampler interacts with an unknown quantum system. When they query, we instead “condition” on the system being consistent with the answer to their query!



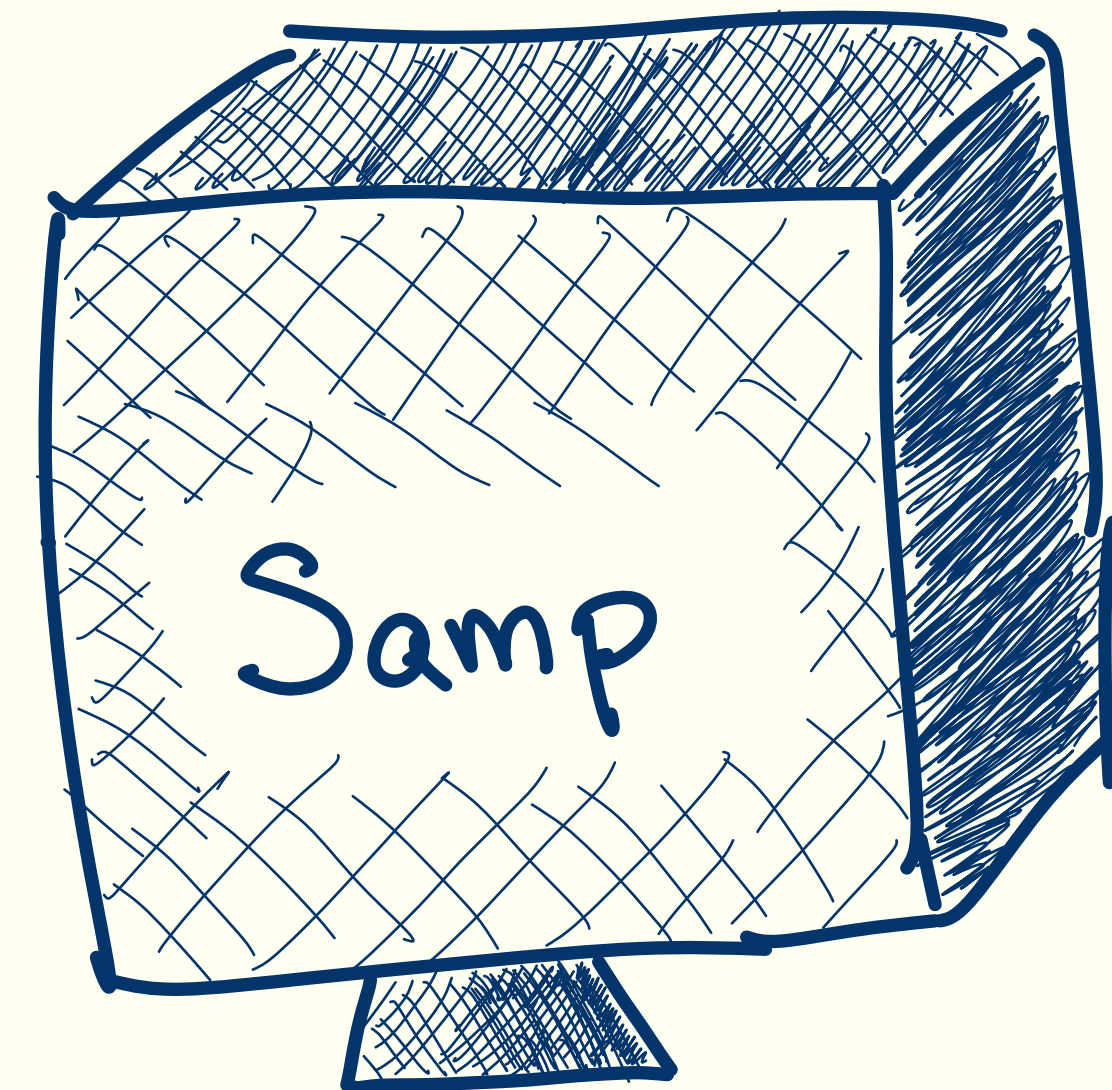
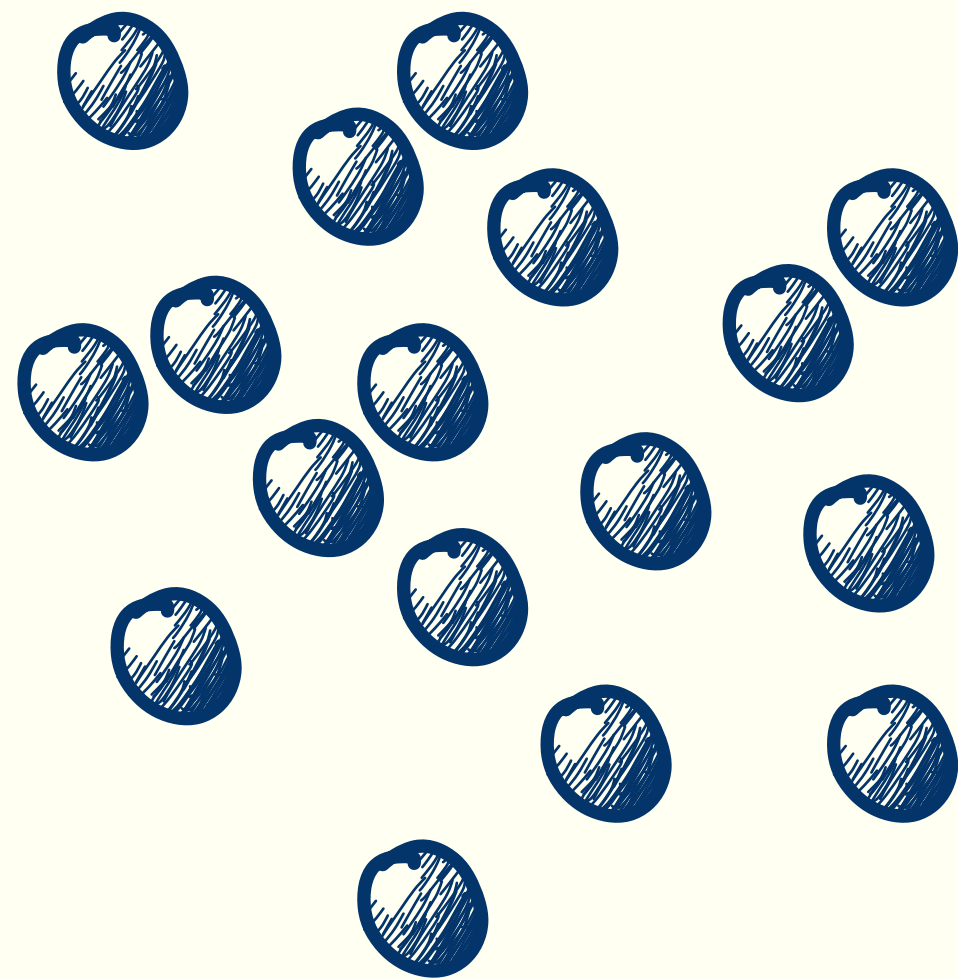
Compressed oracles

In the compressed oracle technique, the sampler interacts with an unknown quantum system. When they query, we instead “condition” on the system being consistent with the answer to their query!



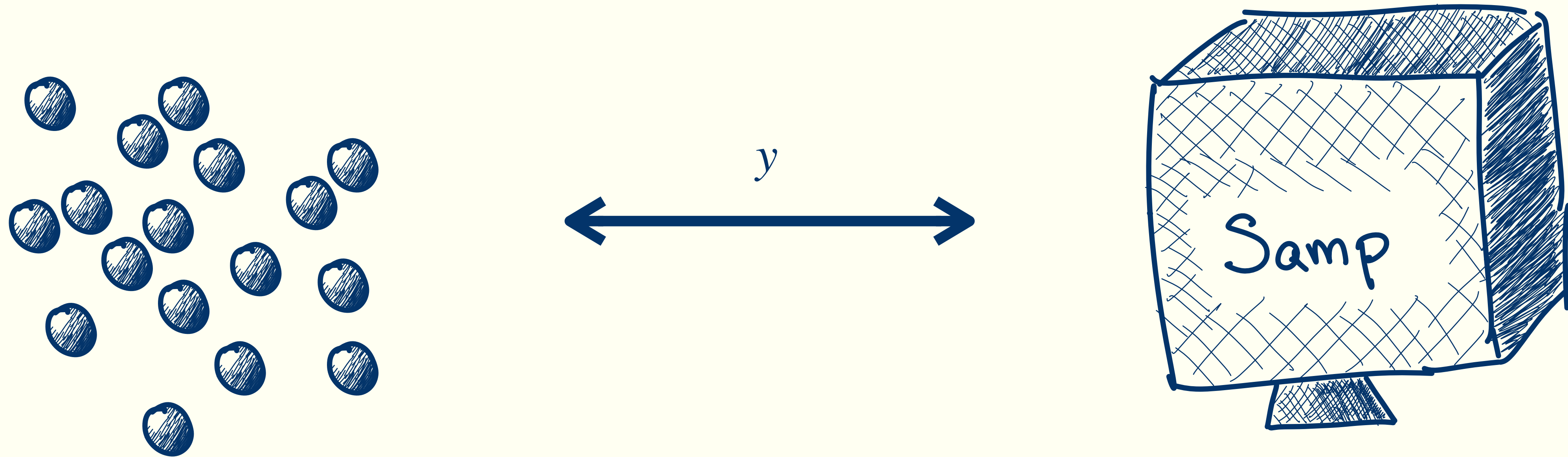
Compressed oracles for spectral Forrelation

It turns out that doing this for spectral Forrelation leads to an interesting connection with physics! We model the quantum system as a collection of “bosons” (the physics version of multisets) at random positions...



Compressed oracles for spectral Forrelation

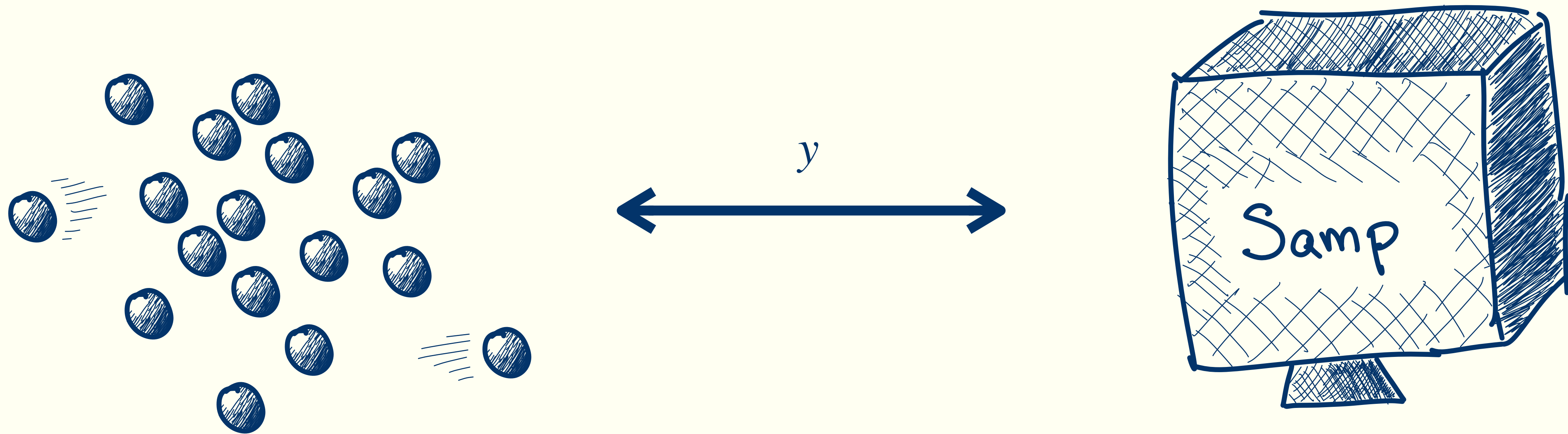
It turns out that doing this for spectral Forrelation leads to an interesting connection with physics! We model the quantum system as a collection of “bosons” (the physics version of multisets) at random positions...



When the algorithm queries U at y , we have to give random pairs of bosons additional momentum y (these form pairs, which I've been told are like Cooper pairs).

Compressed oracles for spectral Forrelation

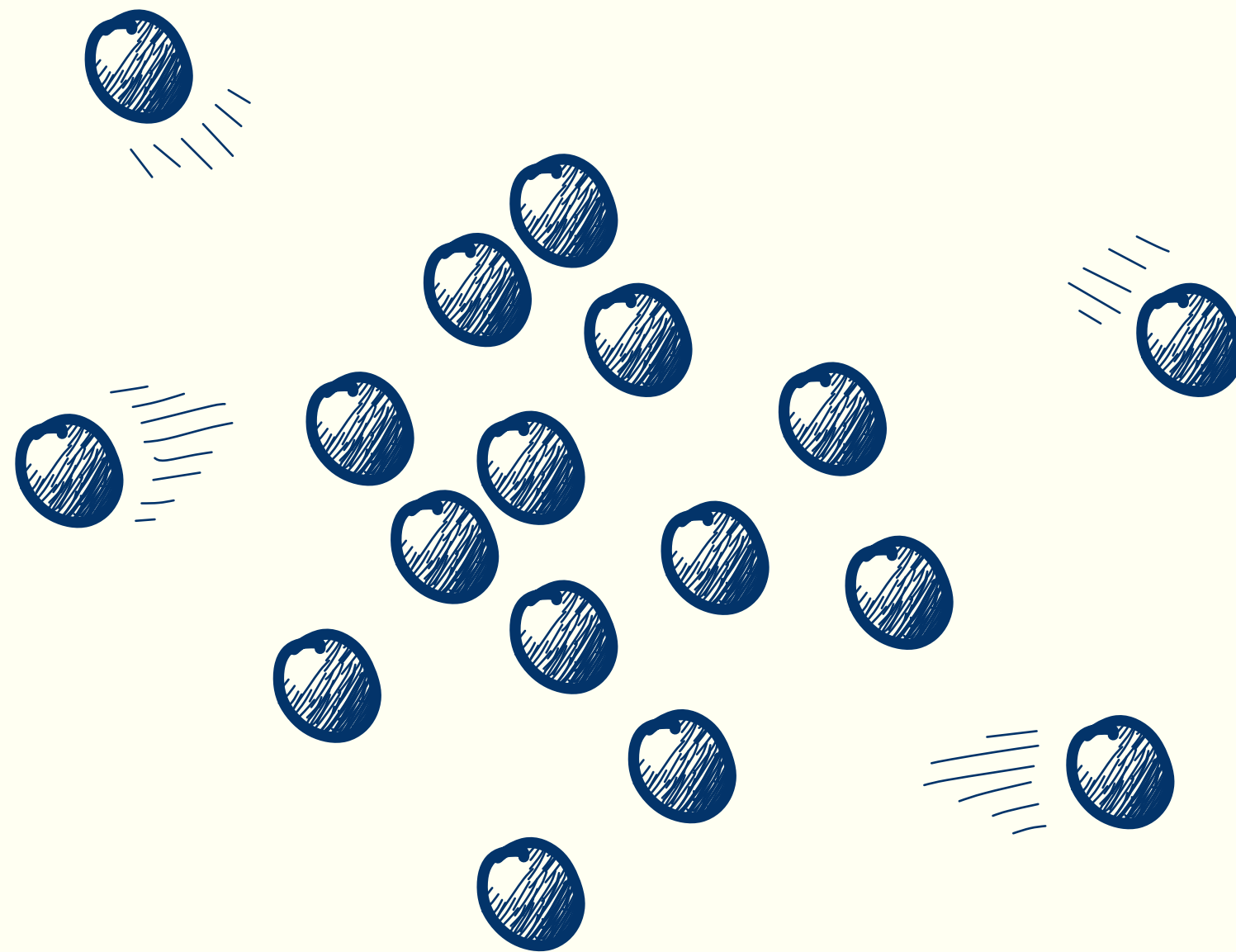
It turns out that doing this for spectral Forrelation leads to an interesting connection with physics! We model the quantum system as a collection of “bosons” (the physics version of multisets) at random positions...



When the algorithm queries U at y , we have to give random pairs of bosons additional momentum y (these form pairs, which I've been told are like Cooper pairs).

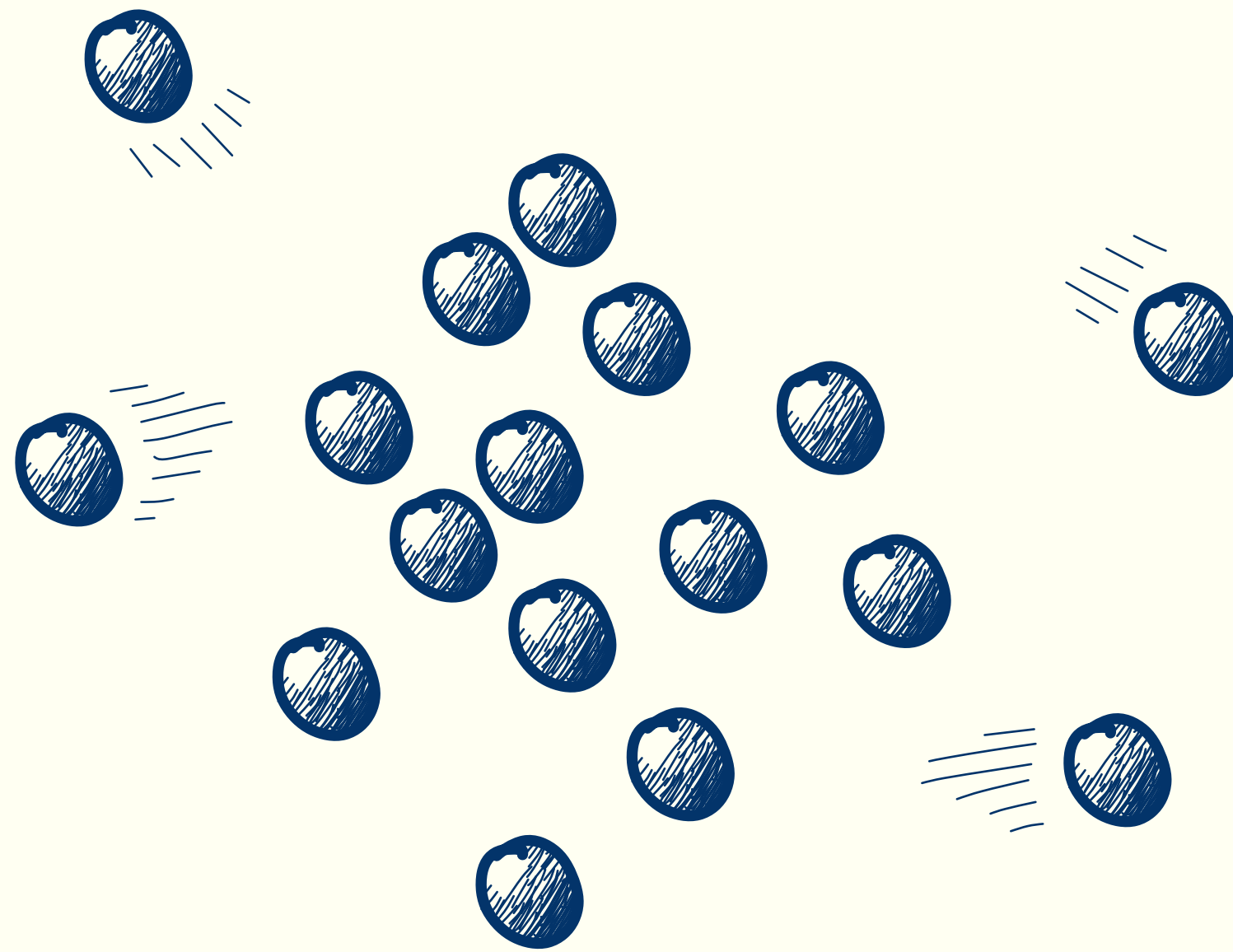
Why can't QCMA solve spectral Forrelation?

We show that knowing about the **momentum of pairs** of bosons tells you nothing about the **positions** of those bosons. As long as the bosons remain paired up, you can't find their positions.



Why can't QCMA solve spectral Forrelation?

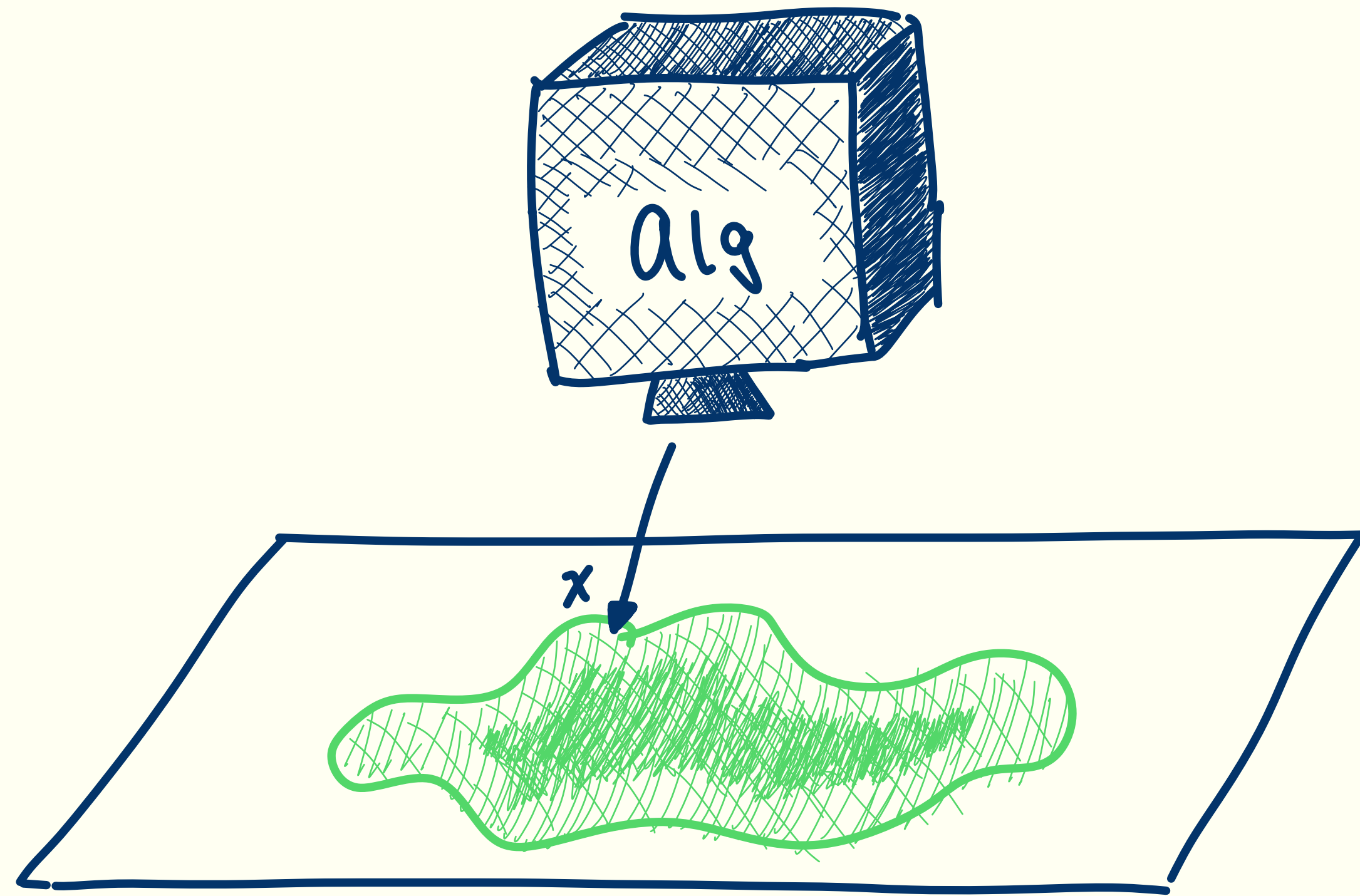
We show that knowing about the **momentum of pairs** of bosons tells you nothing about the **positions** of those bosons. As long as the bosons remain paired up, you can't find their positions.



Not everything is always paired: the sampler might “double bounce”, i.e., add momentum to a boson that already was moving. But using fancy math, we can show this almost never happens → The sampler implies a **contradiction**!

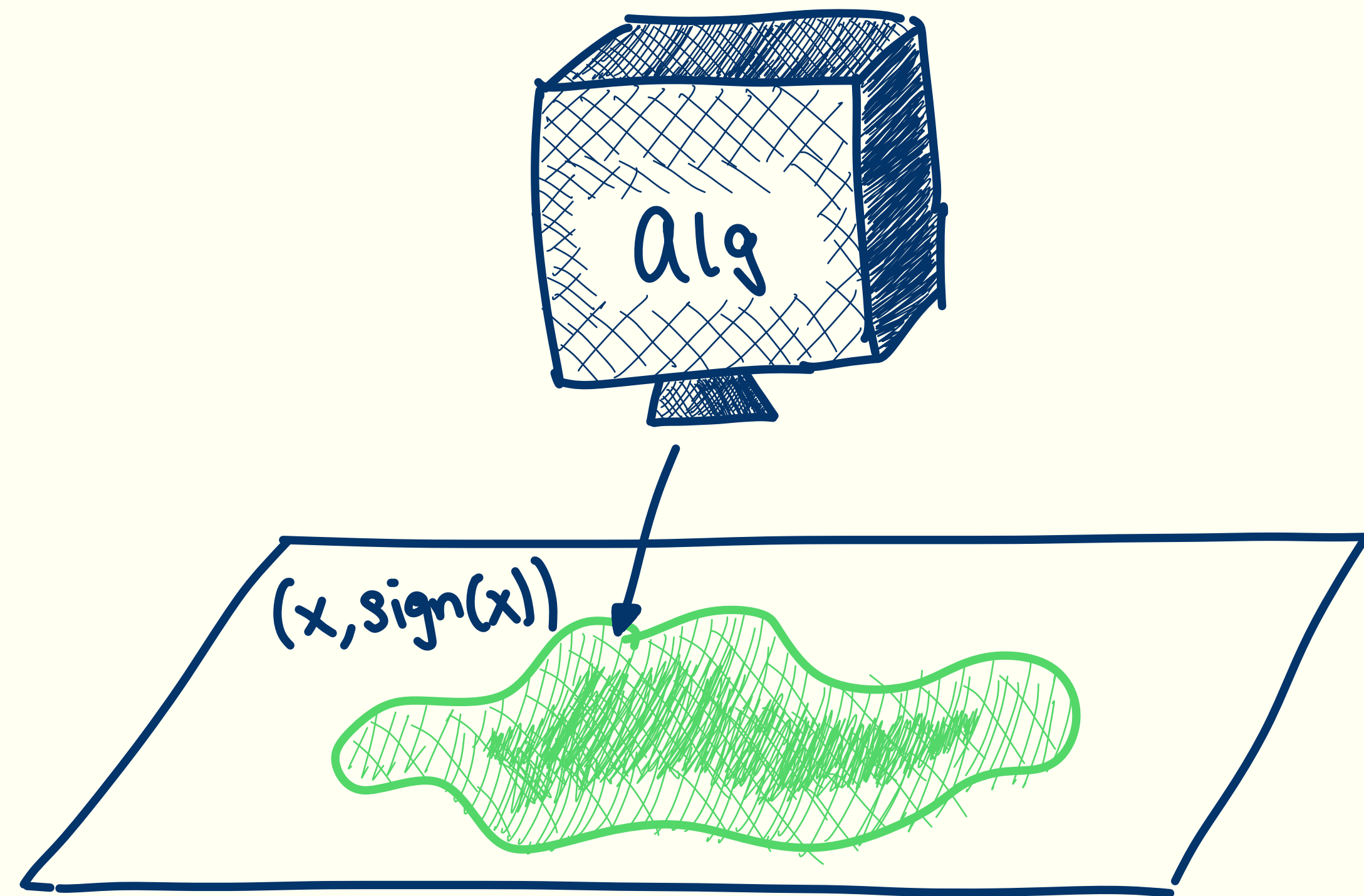
The code intersection size problem

For the code intersection size problem, we start with the subset size problem...



The code intersection size problem

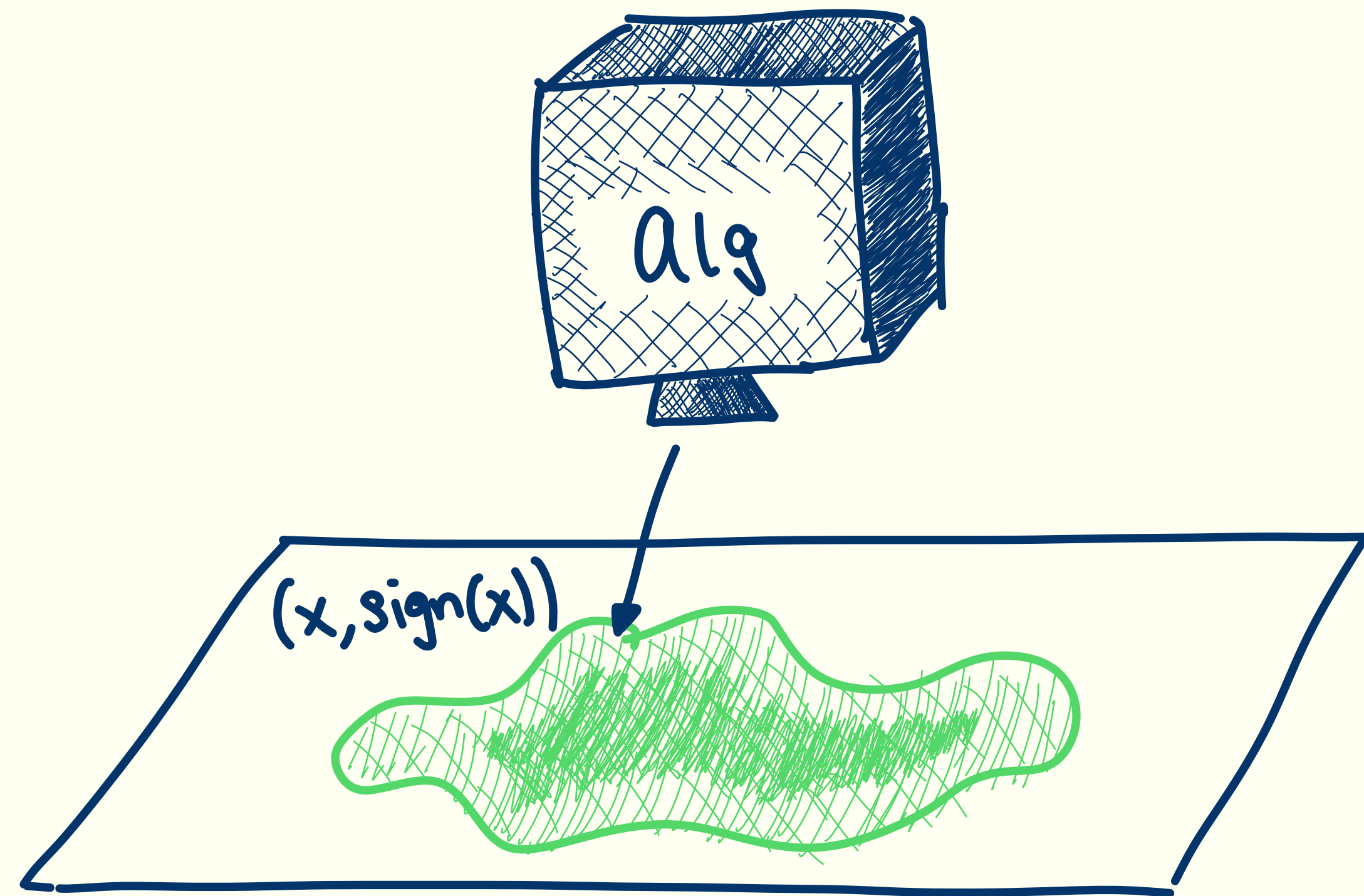
For the code intersection size problem, we start with the subset size problem...



And we add an additional check to each point, similar to forcing the algorithm to sign each x it wants to query (otherwise the oracle will always say 0).

The code intersection size problem

For the code intersection size problem, we start with the subset size problem...



Fix a linear code $C \subseteq \Sigma^n$, and functions $f_1, \dots, f_n : \Sigma \mapsto \{0,1\}$.

The signature of a point x has to satisfy both:

$$(1) \ v = (v_1, \dots, v_n) \in C$$

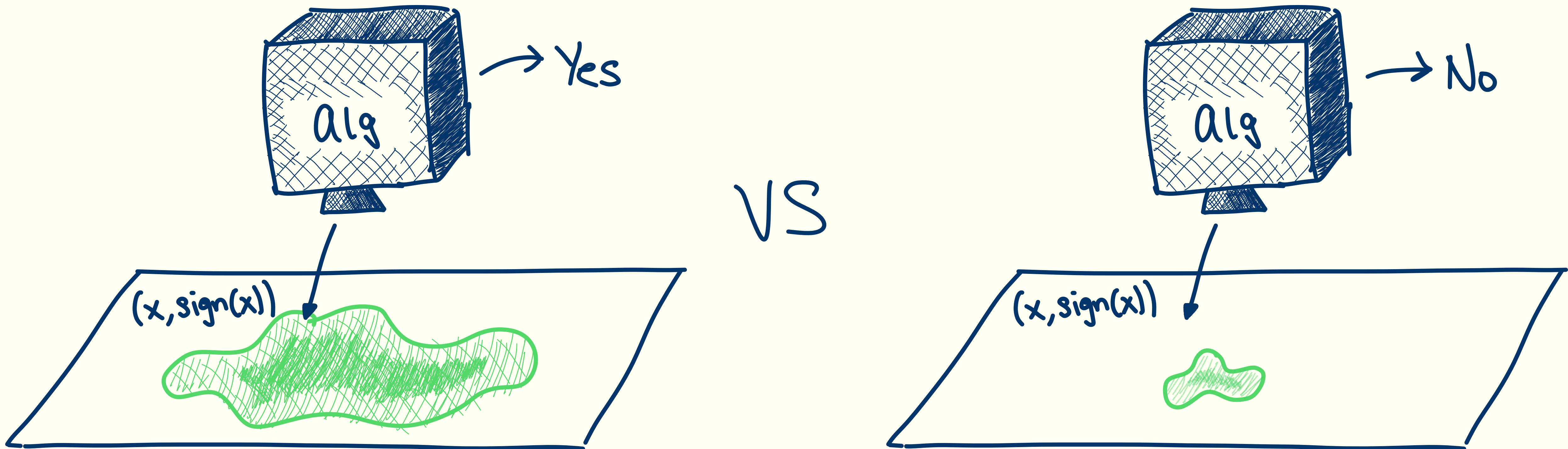
$$(2) \ f_i(v_i) = x_i \text{ for all } i.$$

And we add an additional check to each point, similar to forcing the algorithm to sign each x it wants to query (otherwise the oracle will always say 0).

The code intersection size problem

Input: Oracle access to a signed set oracle $O_{S, f_1, \dots, f_n}$ (for a fixed code C).

Output: Is $|S| = 2^n$ or $|S| \leq 2^n/3$?



Code intersection is in QMA

The Yamakawa-Zhandry algorithm lets a verifier find a valid signature for every x , given the following quantum state:

$$\bigotimes_{i=1}^n |f_i^{-1}(0)\rangle \otimes |f_i^{-1}(1)\rangle$$

Code intersection is in QMA

The Yamakawa-Zhandry algorithm lets a verifier find a valid signature for every x , given the following quantum state:

$$\bigotimes_{i=1}^n |f_i^{-1}(0)\rangle \otimes |f_i^{-1}(1)\rangle$$

This works because quantum algorithms can “realize” the convolution trick:

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

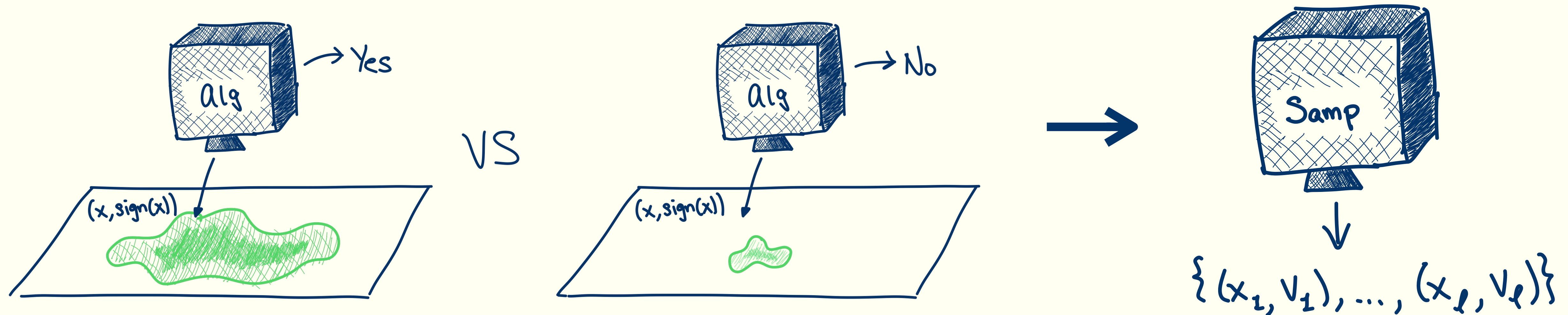
In the code and
has the right hash

Add and decode
the dual code

Code intersection is not in QCMA

We rule out QCMA algorithms for code intersection size in two steps:

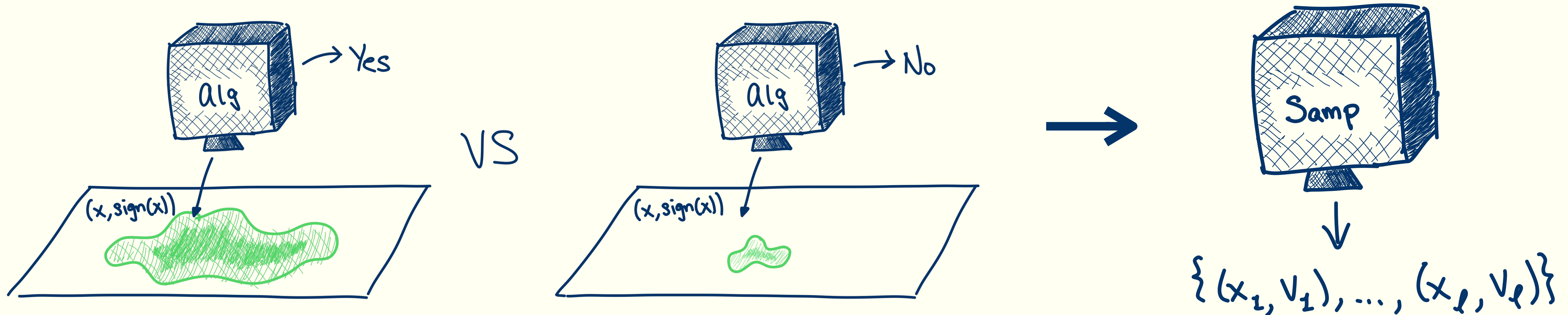
Theorem 1: A QCMA algorithm implies a good signature (codewords and their hashes) sampler that makes no oracle queries.



Code intersection is not in QCMA

We rule out QCMA algorithms for code intersection size in two steps:

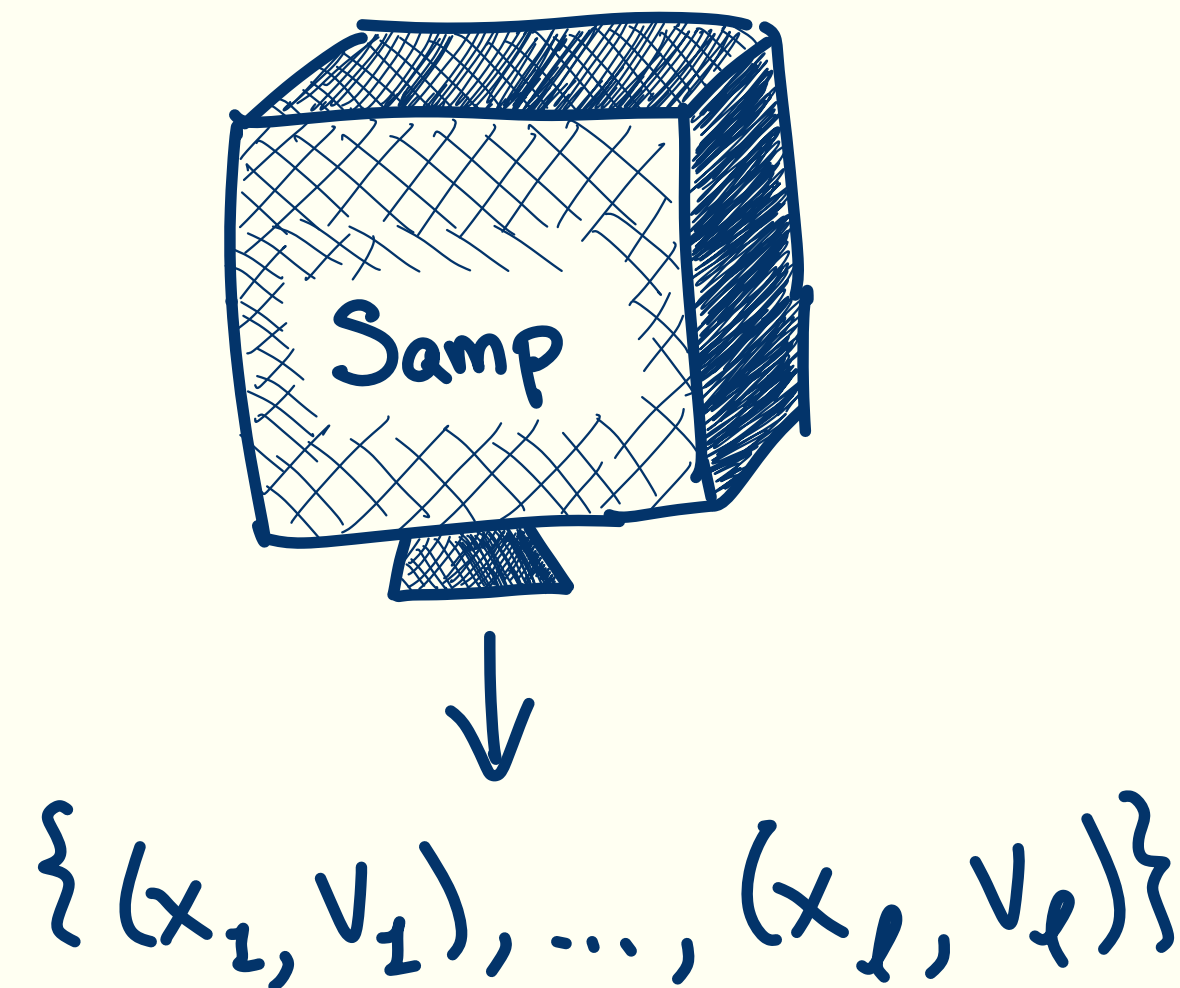
Theorem 1: A QCMA algorithm implies a good signature (codewords and their hashes) sampler that makes no oracle queries.



Theorem 2: Finding many distinct codeword and their hashes means you found many symbols and their hashes (for certain codes).

Code intersection is not in QCMA

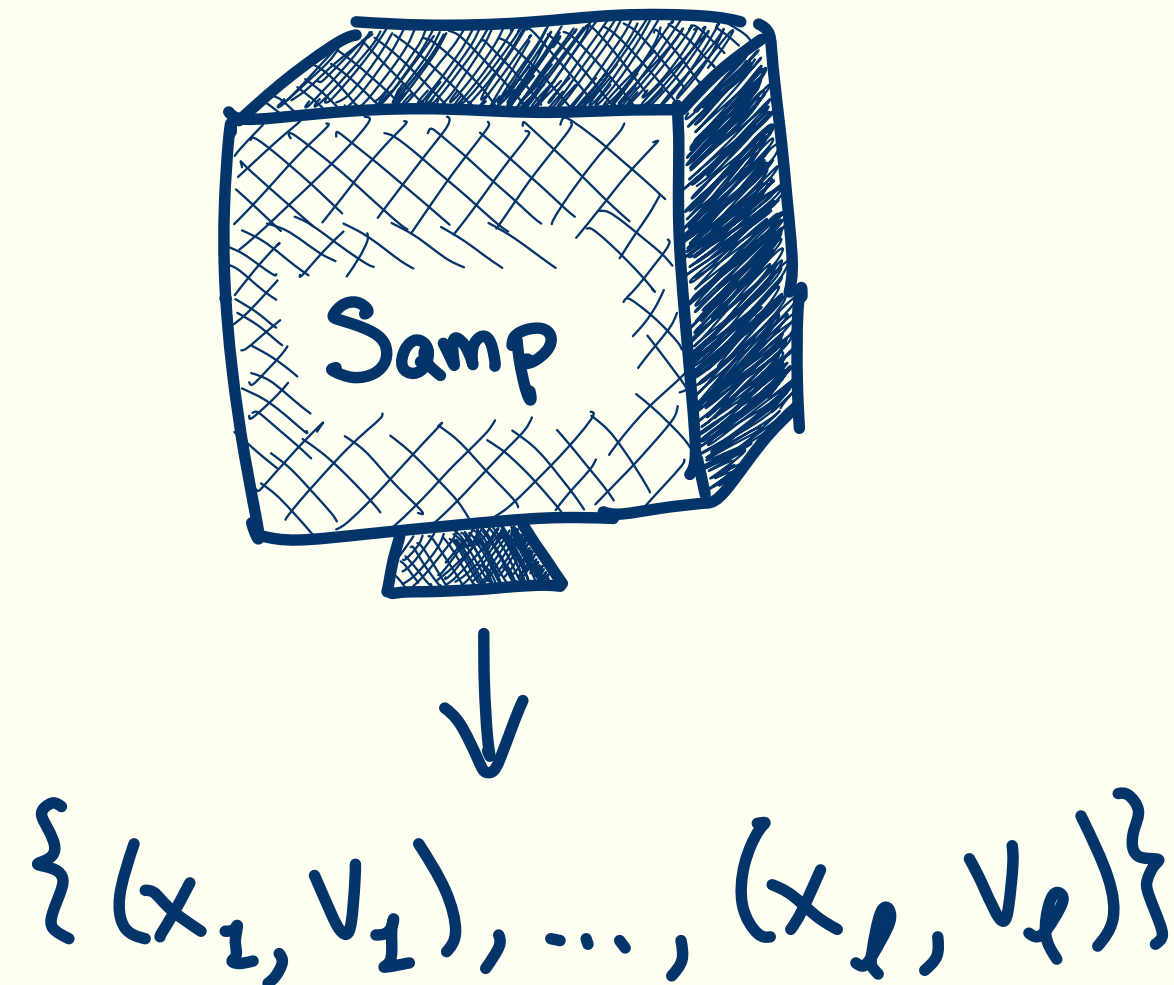
Theorem 2: Finding many distinct codeword and their hashes means you found many symbols and their hashes (for certain codes).



The number of distinct symbols that a collection of codewords contain is called the codes list recoverability.

Code intersection is not in QCMA

Theorem 2: Finding many distinct codeword and their hashes means you found many symbols and their hashes (for certain codes).

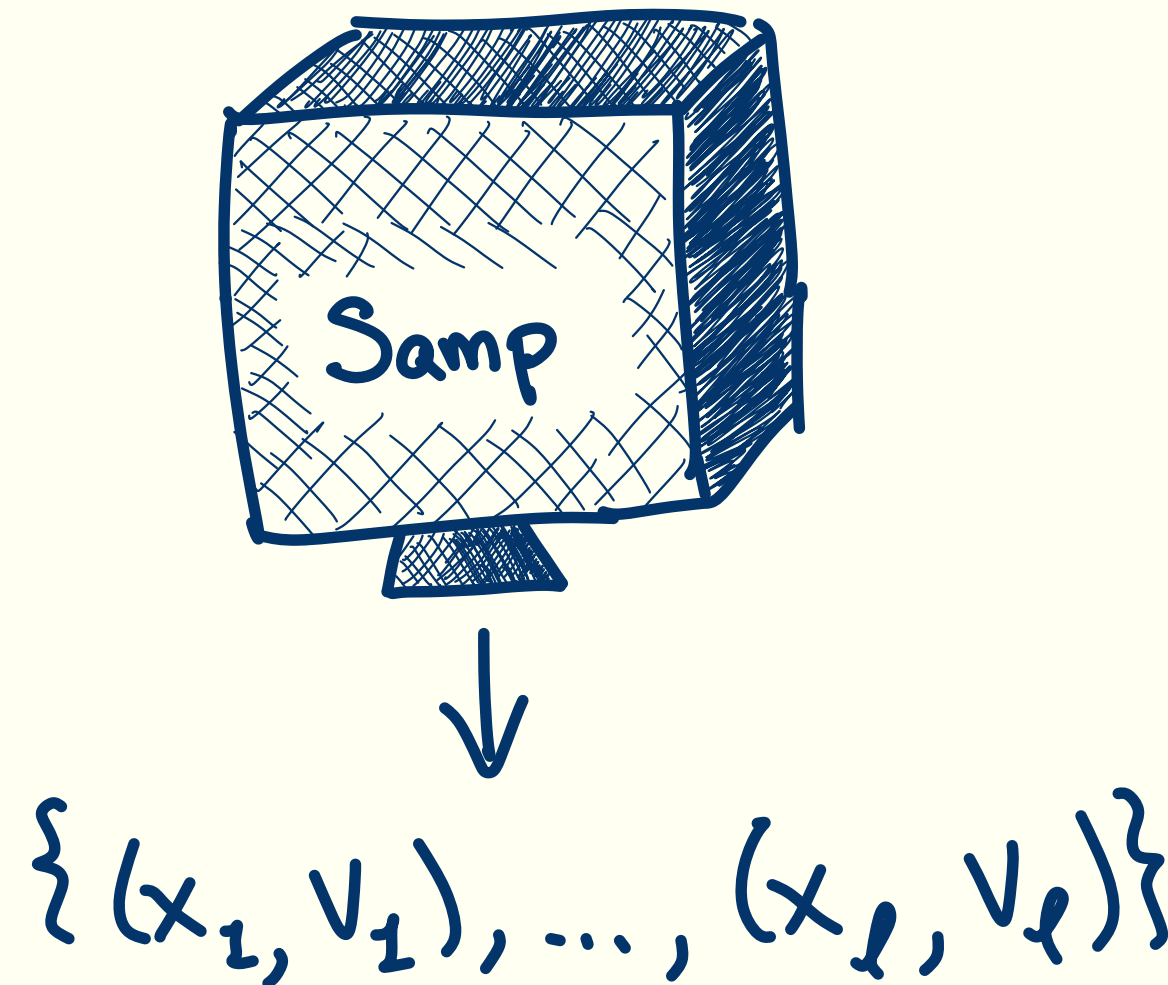


The number of distinct symbols that a collection of codewords contain is called the codes list recoverability. Optimal list recoverability means ℓ codewords have $\sim \ell n$ symbols.

Code intersection is not in QCMA

Theorem 2: Finding many distinct codeword and their hashes means you found many symbols and their hashes (for certain codes).

Thm 1: Correct (i.e., $F(v_i) = x_i$) with high probability!

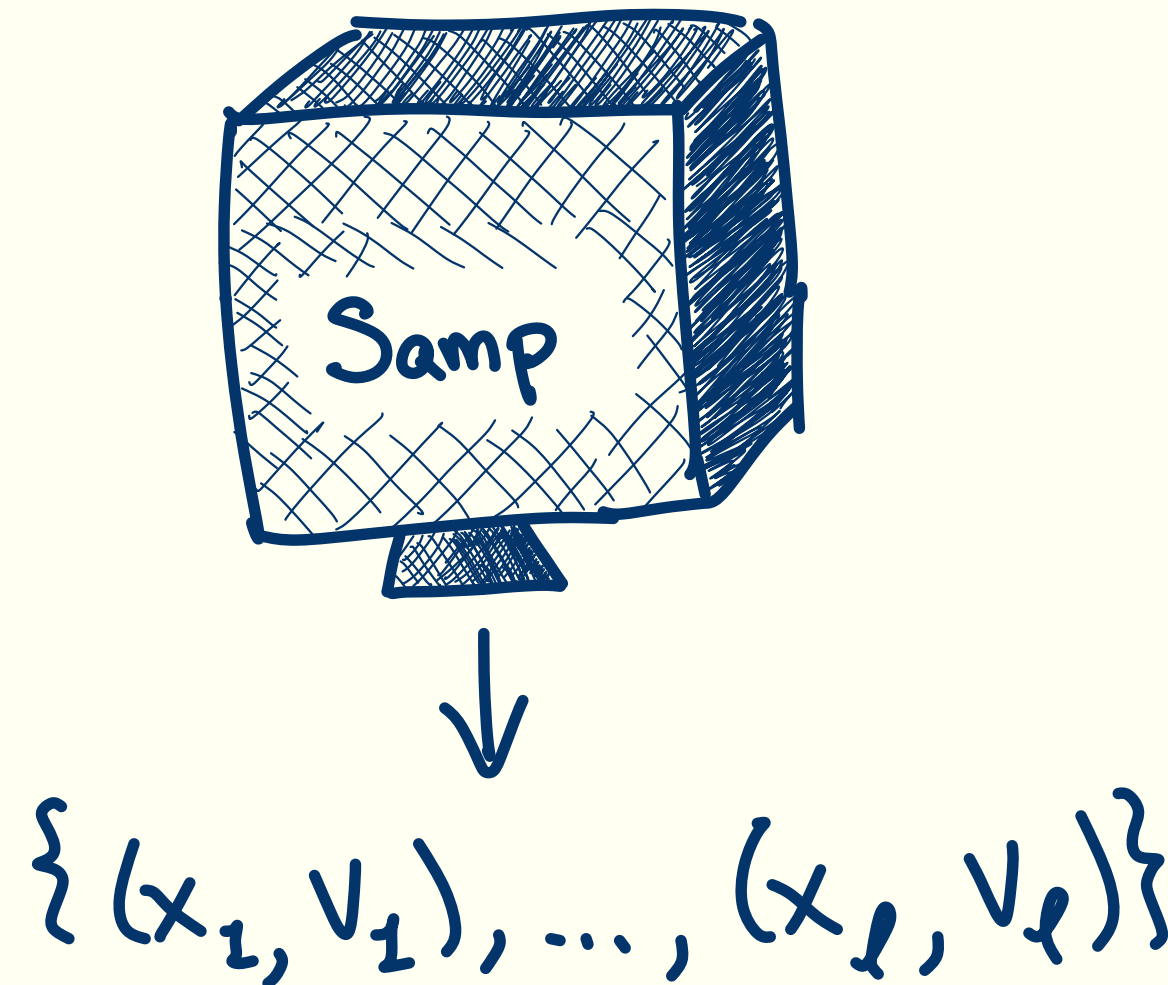


The number of distinct symbols that a collection of codewords contain is called the codes list recoverability. Optimal list recoverability means ℓ codewords have $\sim \ell n$ symbols.

Code intersection is not in QCMA

Theorem 2: Finding many distinct codeword and their hashes means you found many symbols and their hashes (for certain codes).

Thm 1: Correct (i.e., $F(v_i) = x_i$) with high probability!



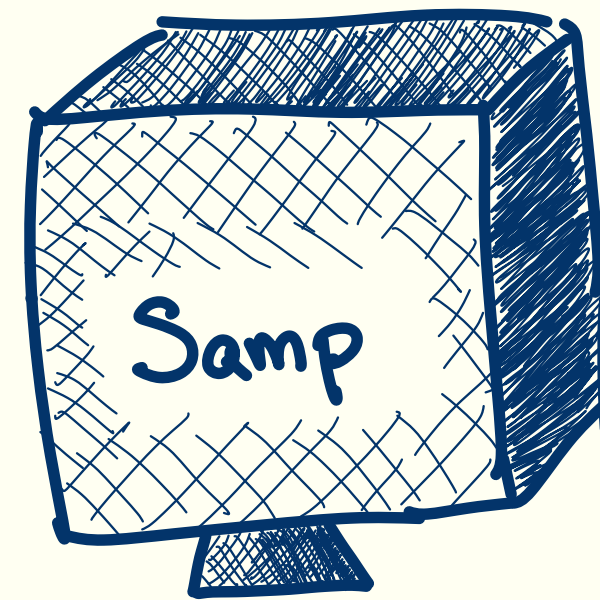
Thm 2: Must have guessed $\ell \cdot n$ random bits!

The number of distinct symbols that a collection of codewords contain is called the codes list recoverability. Optimal list recoverability means ℓ codewords have $\sim \ell n$ symbols.

Code intersection is not in QCMA

Theorem 2: Finding many distinct codeword and their hashes means you found many symbols and their hashes (for certain codes).

Thm 1: Correct (i.e., $F(v_i) = x_i$) with high probability!



$\{(x_1, v_1), \dots, (x_\ell, v_\ell)\}$

Thm 2: Must have guessed $\ell \cdot n$ random bits!

Contradiction!

The number of distinct symbols that a collection of codewords contain is called the codes list recoverability. Optimal list recoverability means ℓ codewords have $\sim \ell n$ symbols.

Code intersection is not in QCMA

The properties you need to get a separation are:

- Good enough distance (inverse polynomial in n).
- Dual is efficiently decodable (not really, but makes things easier).
- Really good list recoverability.

Code intersection is not in QCMA

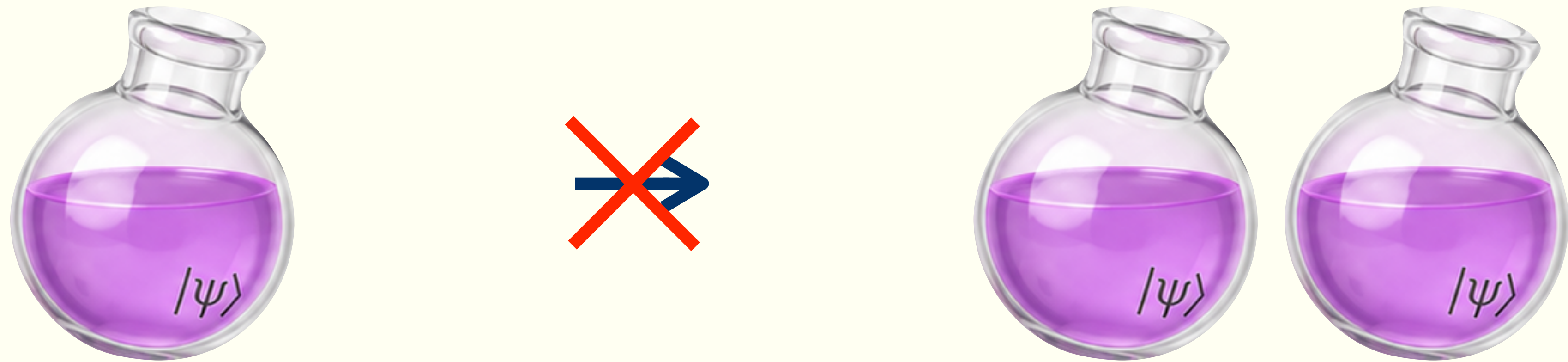
The properties you need to get a separation are:

- Good enough distance (inverse polynomial in n).
- Dual is efficiently decodable (not really, but makes things easier).
- Really good list recoverability.

It turns out that there is a family of codes, multiplicity codes, that have all of these properties. So we get a separation.

Takeaways and future directions

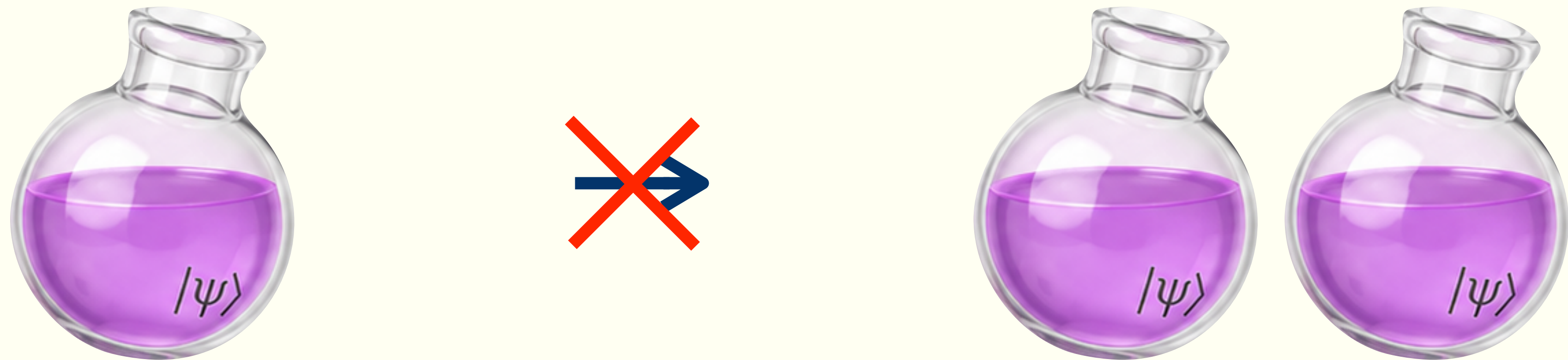
Some problems are only meant to be proven once!



Takeaways and future directions

Some problems are only meant to be proven once!

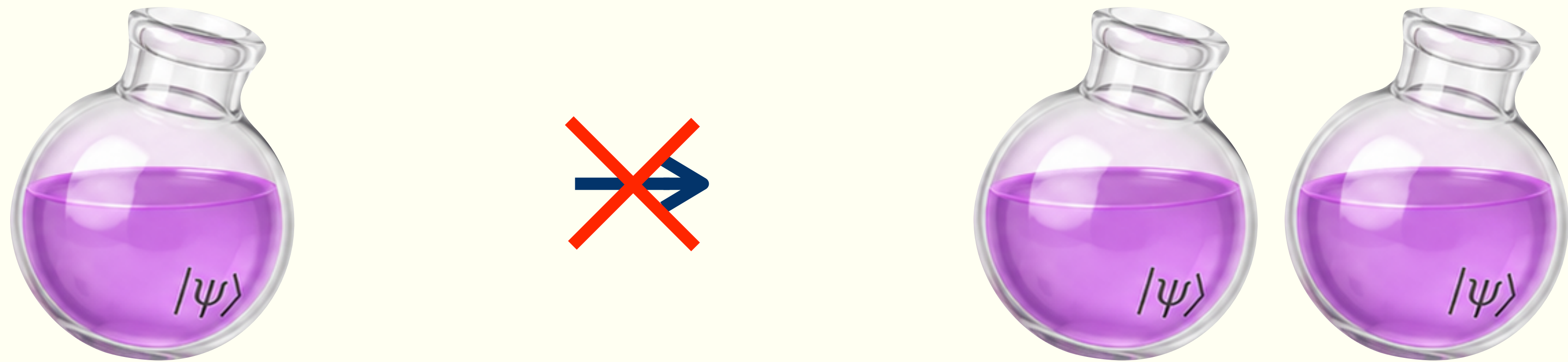
Our work shows that quantum proofs might be more powerful than classical states for solving problems.



Takeaways and future directions

Some problems are only meant to be proven once!

Our work shows that quantum proofs might be more powerful than classical states for solving problems.



Unfortunately, this power actually prevents us from making them clonable! Our ideas show that there are some problems that can only be proven once, otherwise they would lead to impossible situations!

Takeaways and future directions

But, sometimes that is a good thing!

Takeaways and future directions

But, sometimes that is a good thing!

1. We would like to have money that is publicly verifiable, but impossible to counterfeit (clone). Quantum money is the idea that we might be able to use quantum proof to achieve this.



Takeaways and future directions

But, sometimes that is a good thing!

1. We would like to have money that is publicly verifiable, but impossible to counterfeit (clone). Quantum money is the idea that we might be able to use quantum proof to achieve this.
2. We want to be able to publicly verify quantum computation without having to own a quantum computer ourselves. This usually requires getting a quantum computer to verify problems that can't be solved with a classical proof.



Takeaways and future directions

But, sometimes that is a good thing!

1. We would like to have money that is publicly verifiable, but impossible to counterfeit (clone). Quantum money is the idea that we might be able to use quantum proof to achieve this.

2. We want to be able to publicly verify quantum computation without having to own a quantum computer ourselves. This usually requires getting a quantum computer to verify problems that can't be solved with a classical proof.

In these settings, we actually want there to not be a clonable proof! Maybe our ideas can give us insight into how to build quantum money and verify quantum computation!



Takeaways and future directions

But, sometimes that is a good thing!

1. We would like to have money that is publicly verifiable, but impossible to counterfeit (clone). Quantum money is the idea that we might be able to use quantum proof to achieve this.

2. We want to be able to publicly verify quantum computation without having to own a quantum computer ourselves. This usually requires getting a quantum computer to verify problems that can't be solved with a classical proof.

In these settings, we actually want there to not be a clonable proof! Maybe our ideas can give us insight into how to build quantum money and verify quantum computation!

These problems are still really hard though...



Takeaways and future directions

Some more reasonable problems to shoot for:

Takeaways and future directions

Some more reasonable problems to shoot for:

1. QMA search-to-decision: Can you find a **classical** oracle input problem that can be solved in QMA, but a witness can not be synthesized in BQP^{QMA} ? (Or find a reduction that relativizes)

Takeaways and future directions

Some more reasonable problems to shoot for:

1. QMA search-to-decision: Can you find a **classical** oracle input problem that can be solved in QMA, but a witness can not be synthesized in BQP^{QMA} ? (Or find a reduction that relativizes)
2. QMA versus QCMA communication complexity: Two parties get N -bit inputs x and y , and want to compute $f(x, y)$ using $\text{poly log}(N)$ quantum communication, with the help of either a quantum proof (QMA), which should be impossible with a classical proof (QCMA).

Takeaways and future directions

Some more reasonable problems to shoot for:

1. QMA search-to-decision: Can you find a **classical** oracle input problem that can be solved in QMA, but a witness can not be synthesized in BQP^{QMA} ? (Or find a reduction that relativizes)
2. QMA versus QCMA communication complexity: Two parties get N -bit inputs x and y , and want to compute $f(x, y)$ using $\text{poly log}(N)$ quantum communication, with the help of either a quantum proof (QMA), which should be impossible with a classical proof (QCMA).
3. QMA versus IP: Can you prove that spectral Forrelation (without sparse S) is outside of IP? What about BQP versus IP? These problems probably require a better understanding of IP rather than QMA or BQP.