Learning the closest product state

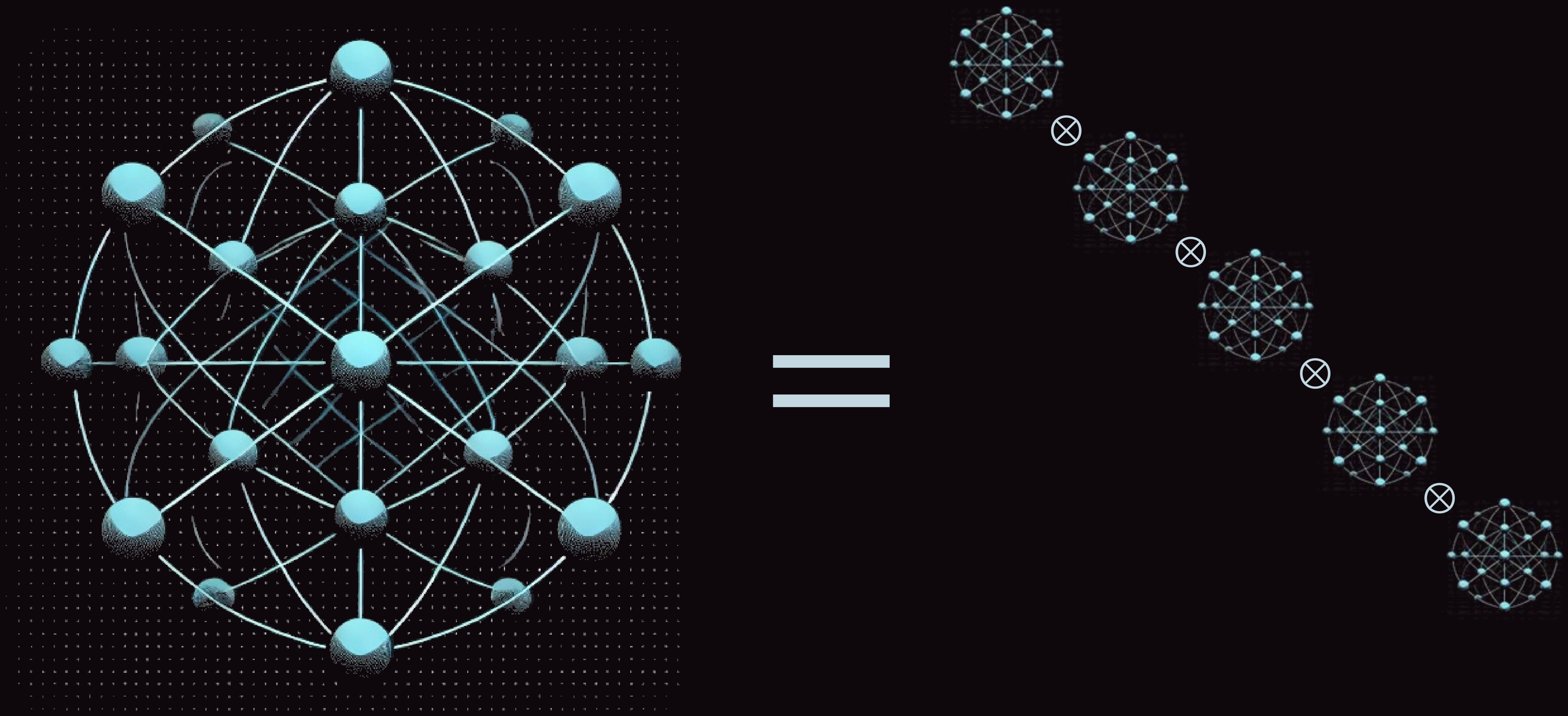
Ainesh Bakshi, John Bostanci, William Kretschmer, Zeph Landau,
Jerry Li, Allen Liu, Ryan O'Donnell, and Ewin Tang

Quantum state learning

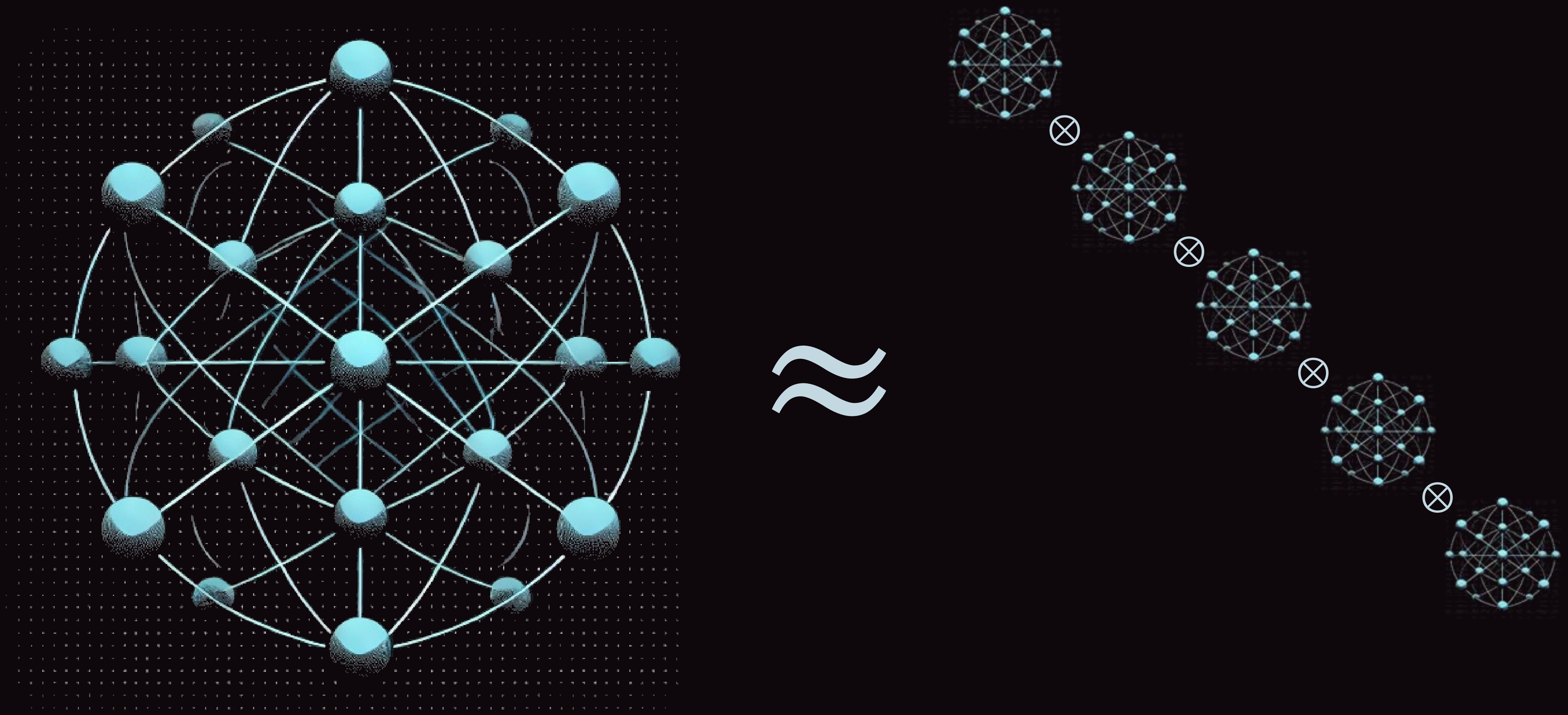
Typical set up:

Photo credits: OpenAI

Quantum state learning



Quantum state learning



Product state learning

Task: Learn a classical description of a product state.

Algorithm: Do state tomography on every register and output the tensor product the reduced states.

Product state learning

Consider the following input:

$$\sqrt{1-\epsilon}|0^n\rangle + \sqrt{\epsilon}|+^n\rangle$$

Product state learning

Consider the following input:

$$\sqrt{1-\epsilon}|0^n\rangle + \sqrt{\epsilon}|+^n\rangle$$

Close to all 0's (product state), but every marginal isn't quite $|0\rangle$.

Product state learning

Moral: Even a little bit of misclassification error can change the nature of quantum state learning problems.

Agnostic quantum state learning

Given a model class C and copies of an arbitrary quantum state, output the description of the closest state in C to the state.

$$(C, \rho^{\otimes n}) \to \underset{|\psi\rangle \in C}{\operatorname{argmax}} \langle\psi|\rho|\psi\rangle$$

Agnostic quantum state learning

Given a model class C and copies of an arbitrary quantum state, output the description of the closest state in C to the state.

$$\left(C, \rho^{\otimes n}\right) \to |\phi\rangle \in C\colon \max_{|\psi\rangle \in C} \langle\psi|\rho|\psi\rangle - \langle\phi|\rho|\phi\rangle \leq \epsilon$$

Agnostic quantum state learning

Given a model class C and copies of an arbitrary quantum state, output the description of the closest state in C to the state.

$$(C, \rho^{\otimes n}) \to |\phi\rangle \in C\colon \max_{|\psi\rangle \in C} \langle\psi|\rho|\psi\rangle - \langle\phi|\rho|\phi\rangle \leq \epsilon$$

If you don't care about runtime, shadow tomography solves this in $O(n \cdot \log^2(|C|) \cdot \epsilon^{-4})$ samples.

Agnostic quantum state learning

Given a model class C and copies of an arbitrary quantum state, output the description of the closest state in C to the state.

$$\left(C, \rho^{\otimes n}\right) \to |\phi\rangle \in C: \max_{|\psi\rangle \in C} \langle\psi|\rho|\psi\rangle - \langle\phi|\rho|\phi\rangle \leq \epsilon$$

Surprisingly, few computationally efficient algorithms exist, even for simple families like product states!

Learning the closest product state

Main result: We provide an algorithm for agnostic learning of product states that has sample complexity and time complexity that is

$$\mathrm{poly}\left(n^{poly\left(\frac{1}{\epsilon}\right)}\right)$$

Learning the closest product state

We can't use the fact that the input is separable.

Learning the closest product state

We can't use the fact that the input is separable, but

Given a state ρ has fidelity at least ½ with some product state.

Learning the closest product state

We can't use the fact that the input is separable, but

Given a state ρ has fidelity at least ½ with some product state.

1. The reduced states of ρ have fidelity at least ½ with some product state too.

Learning the closest product state

We can't use the fact that the input is separable, but
Given a state ρ has fidelity at least ½ with some product state.

1. The reduced states of ρ have fidelity at least ½ with some product state too.

2. There are at most 2 orthogonal product states that have fidelity larger than ½ with ρ, and all of its reduced states.

Learning the closest product state

These observations motivate the following main loop:

For k from 1 through n:

Given a net $\{\pi_i\}$ for first k registers $(\langle \pi_i | \rho_{[k]} | \pi_i \rangle \geq \frac{1}{2}$ and $\langle \pi_i | \pi_j \rangle \approx 0)$,

Find a net for k+1 registers.

Learning the closest product state

High level algorithm:

1. Search in a small ball around $|\pi_i\rangle \otimes |\phi_{k+1}\rangle$ (the root candidate).

Learning the closest product state

High level algorithm:

1. Search in a small ball around $|\pi_i\rangle \otimes |\phi_{k+1}\rangle$ (the root candidate).
2. For a state with high fidelity with ρ.

Learning the closest product state

High level algorithm:

1. Search in a small ball around $|\pi_i\rangle \otimes |\phi_{k+1}\rangle$ (the root candidate).
2. For a state with high fidelity with ρ.
3. That is far from other good candidates we found.

Learning the closest product state

High level algorithm:

1. Search in a small ball around $|\pi_i\rangle \otimes |\phi_{k+1}\rangle$ (the root candidate).
2. For a state with high fidelity with ρ.
3. That is far from other good candidates we found.

Learning the closest product state

High level algorithm:

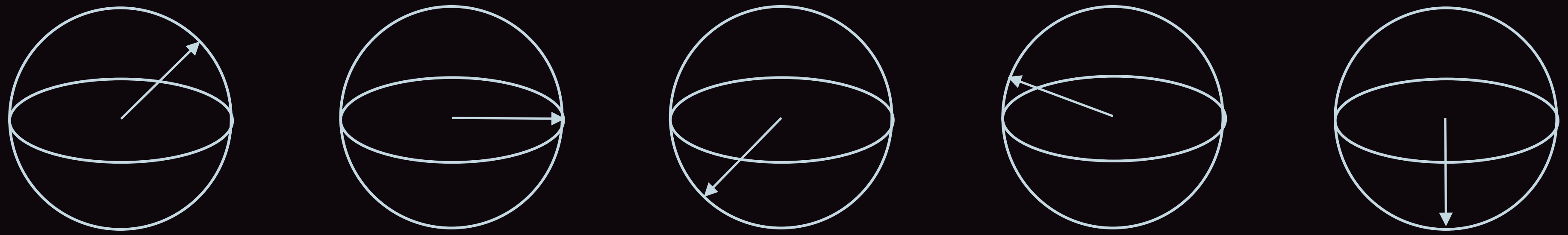
1. Search in a small ball around $|\pi_i\rangle \otimes |\phi_{k+1}\rangle$ (the root candidate).
2. For a state with high fidelity with ρ.
3. That is far from other good candidates we found.

The remaining technical challenge will be turning our objective into a low-degree polynomial and then optimizing that polynomial.

Our algorithm:

1. Search in a small ball around the root candidate.
2. For a state with high fidelity with ρ.
3. That is far from other good candidates we found.

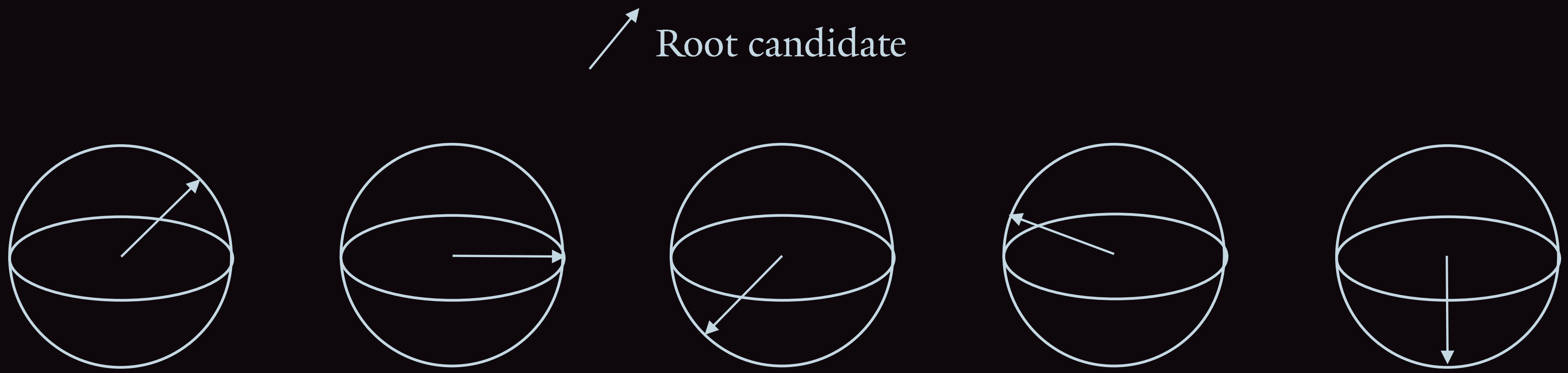
Learning the closest product state



Our algorithm:
1. Search in a small ball around the root candidate.
2. For a state with high fidelity with ρ.
3. That is far from other good candidates we found.

Learning the closest product state

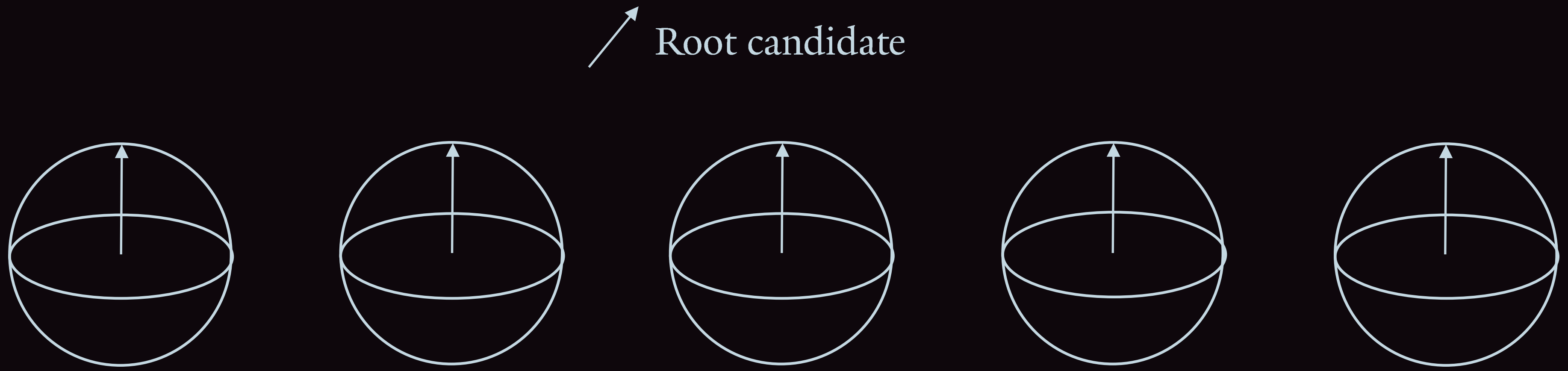
First rotate so that our root candidate is all 0's.



Our algorithm:
1. Search in a small ball around the root candidate.
2. For a state with high fidelity with ρ.
3. That is far from other good candidates we found.

Learning the closest product state

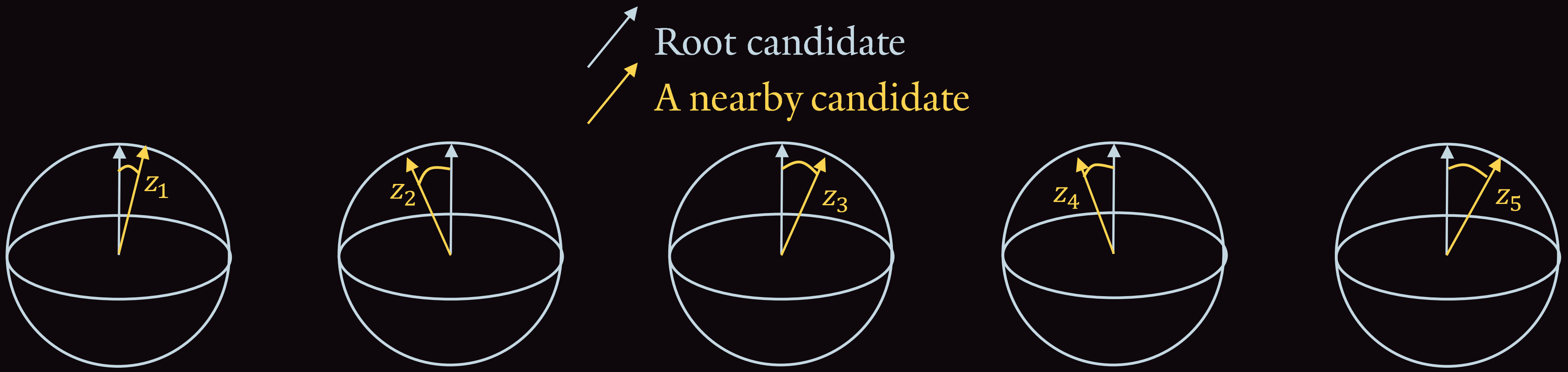
First rotate so that our root candidate is all 0's.



Our algorithm:
1. Search in a small ball around the root candidate.
2. For a state with high fidelity with ρ.
3. That is far from other good candidates we found.

Learning the closest product state

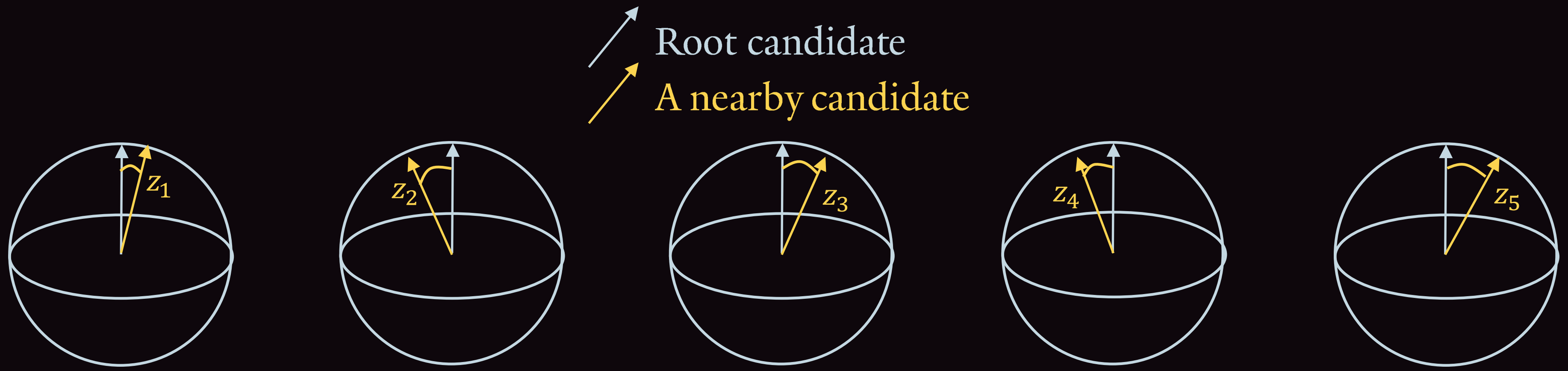
Then the product state ball around $|0^{k+1}\rangle$ will be almost entirely supported on low Hamming weight strings.



Our algorithm:
1. Search in a small ball around the root candidate.
2. For a state with high fidelity with ρ.
3. That is far from other good candidates we found.

Learning the closest product state

Then the product state ball around $|0^{k+1}\rangle$ will be almost entirely supported on low Hamming weight strings.



The "quantum part" of the algorithm will be to do tomography on the low-weight restriction of the input ρ.

Our algorithm:

1. Search in a small ball around the root candidate.
2. For a state with high fidelity with ρ.
3. That is far from other good candidates we found.

Learning the closest product state

Maximizing fidelity ≈ maximizing the following polynomial

$$\max_{\vec{z}\in\mathbb{C}^{k+1}} \sum_{\substack{x,x'\in\{0,1\}^{k+1} \\ |x|,|x|\leq d}} \langle x|\rho_d|x\rangle(\vec{z}^{*,x})(\vec{z}^{x'})$$

Our algorithm:

1. Search in a small ball around the root candidate.
2. For a state with high fidelity with ρ.
3. That is far from other good candidates we found.

Learning the closest product state

Maximizing fidelity ≈ maximizing the following polynomial

$$\max_{\vec{z}\in\mathbb{C}^{k+1}} \sum_{\substack{x,x'\in\{0,1\}^{k+1} \\ |x|,|x|\leq d}} \langle x|\rho_d|x\rangle(\vec{z}^{*,x})(\vec{z}^{x'})$$

Low Degree!

Improvements in some settings

1. Very high fidelity ($\geq \frac{5}{6}$)

Improvements in some settings

1. Very high fidelity ($\geq \frac{5}{6}$)

2. Finitely many, far apart, choices per register

Improvements in some settings

1. Very high fidelity ($\geq \frac{5}{6}$)

2. Finitely many, far apart, choices per register

3. Polynomial bond-dimension MPS

Even more efficient algorithms?

Our algorithm is only polynomial time when ϵ is a constant.

Is there an algorithm that runs in polynomial time when ϵ is small?

Even more efficient algorithms?

Our algorithm is only polynomial time when ϵ is a constant.

Is there an algorithm that runs in polynomial time when ϵ is small?

Not unless NP $\subseteq$ BQP (if $\epsilon \approx \frac{1}{\mathrm{poly}(n)}$).

Even more efficient algorithms?

Our algorithm is only polynomial time when ϵ is a constant.

Is there an algorithm that runs in polynomial time when ϵ is small?

Not unless NP $\subseteq$ BQP (if $\epsilon \approx \frac{1}{\mathrm{poly}(n)}$).

Why? Turns out the connection to tensor optimization goes both ways.