Title: Provably efficient machine learning for quantum many-body problems - Hsin-Yuan Huang, California Institute of Technology

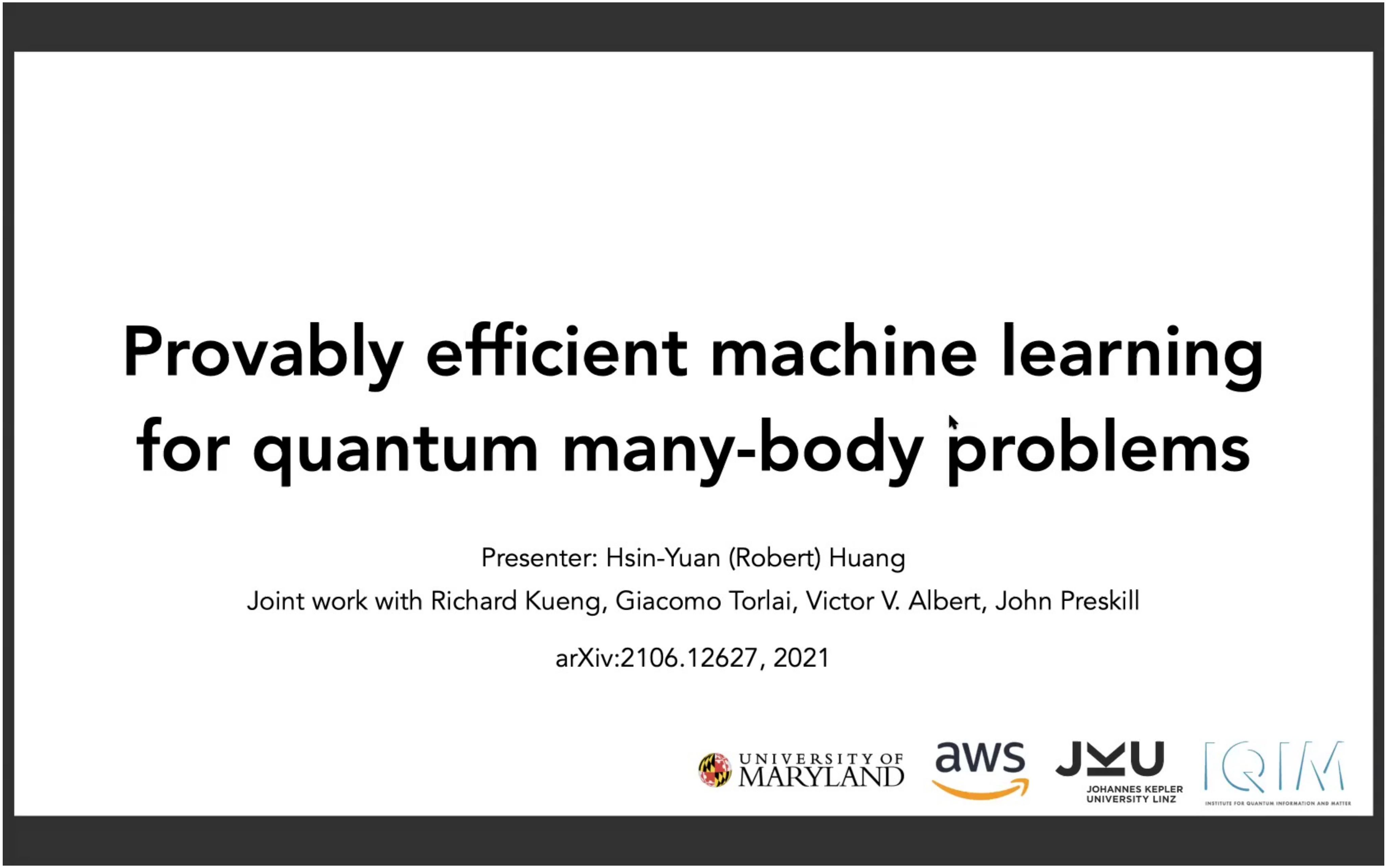
Speakers:

Series: Perimeter Institute Quantum Discussions

Date: October 20, 2021 - 3:30 PM

URL: https://pirsa.org/21100010

Abstract: Classical machine learning (ML) provides a potentially powerful approach to solving challenging quantum many-body problems in physics and chemistry. However, the advantages of ML over more traditional methods have not been firmly established. In this work, we prove that classical ML algorithms can efficiently predict ground state properties of gapped Hamiltonians in finite spatial dimensions, after learning from data obtained by measuring other Hamiltonians in the same quantum phase of matter. In contrast, under widely accepted complexity theory assumptions, classical algorithms that do not learn from data cannot achieve the same guarantee. We also prove that classical ML algorithms can efficiently classify a wide range of quantum phases of matter. Our arguments are based on the concept of a classical shadow, a succinct classical description of a many-body quantum state that can be constructed in feasible quantum experiments and be used to predict many properties of the state. Extensive numerical experiments corroborate our theoretical results in a variety of scenarios, including Rydberg atom systems, 2D random Heisenberg models, symmetry-protected topological phases, and topologically ordered phases.



Provably efficient machine learning for quantum many-body problems

Presenter: Hsin-Yuan (Robert) Huang

Joint work with Richard Kueng, Giacomo Torlai, Victor V. Albert, John Preskill

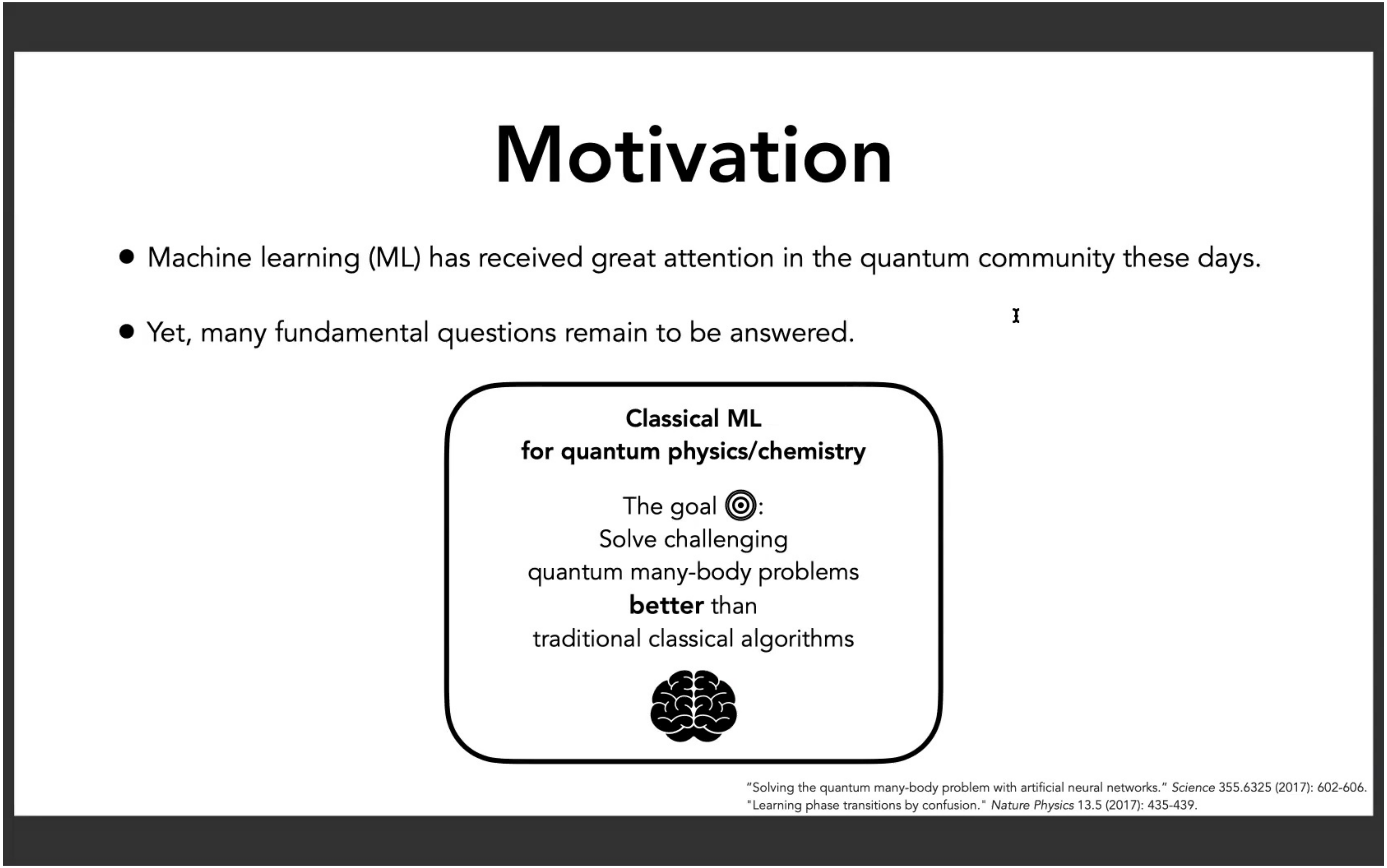
arXiv:2106.12627, 2021

UNIVERSITY OF MARYLAND

aws

JKU JOHANNES KEPLER UNIVERSITY LINZ

IQIM INSTITUTE FOR QUANTUM INFORMATION AND MATTER



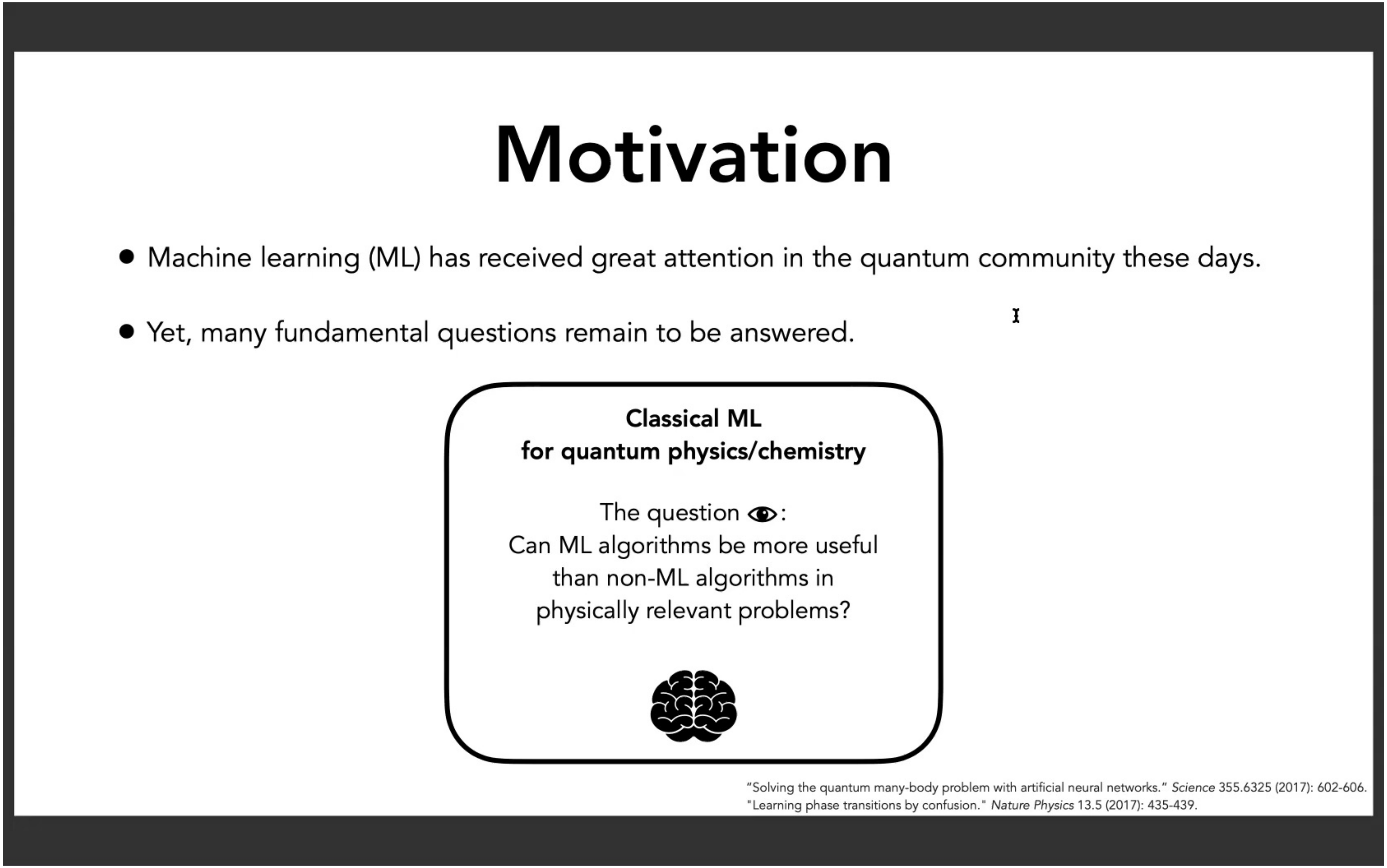
Motivation

- Machine learning (ML) has received great attention in the quantum community these days.
- Yet, many fundamental questions remain to be answered.

Classical ML
for quantum physics/chemistry

The goal ◎:
Solve challenging
quantum many-body problems
better than
traditional classical algorithms

"Solving the quantum many-body problem with artificial neural networks." *Science* 355.6325 (2017): 602-606.
"Learning phase transitions by confusion." *Nature Physics* 13.5 (2017): 435-439.



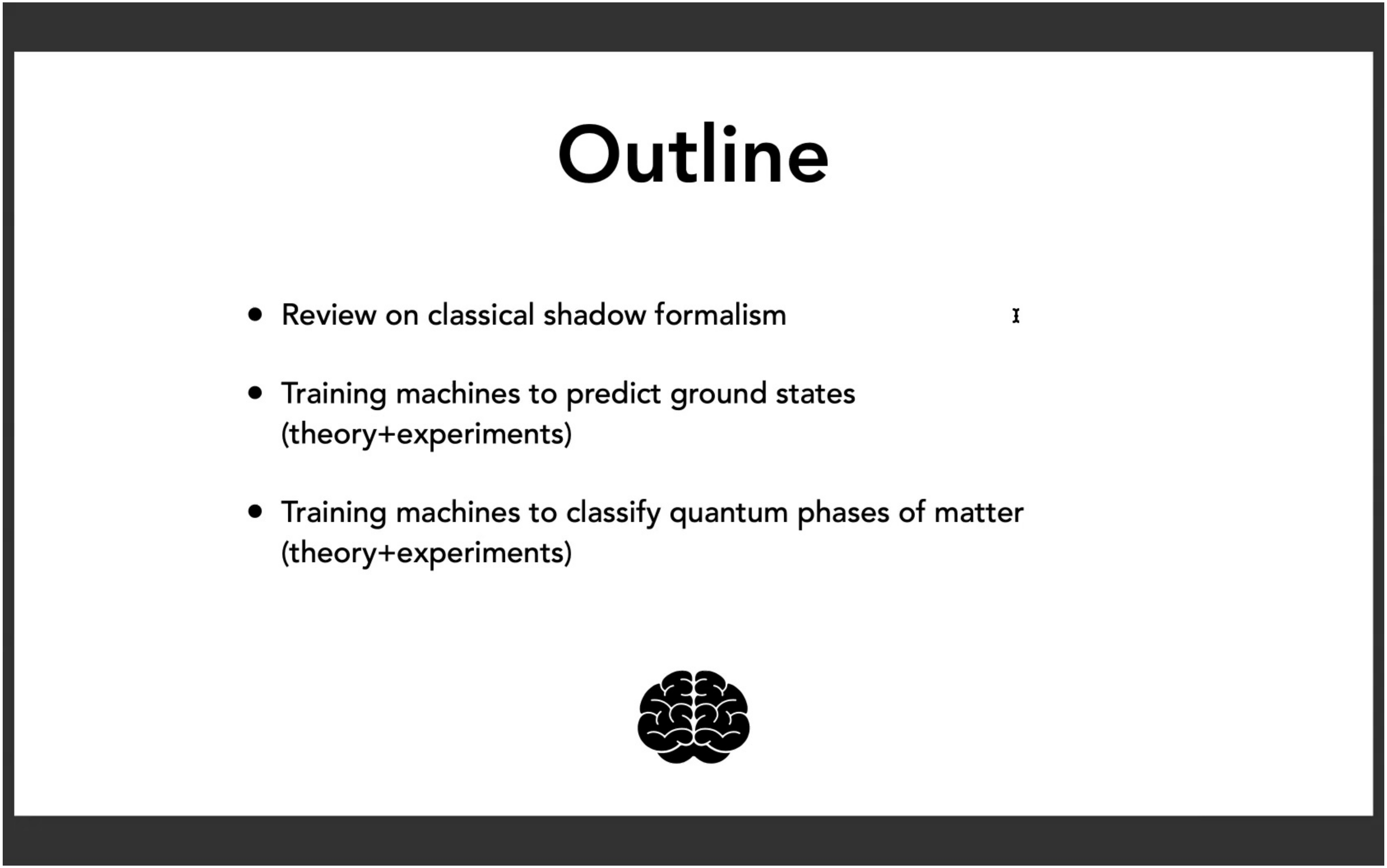
Motivation

- Machine learning (ML) has received great attention in the quantum community these days.
- Yet, many fundamental questions remain to be answered.

Classical ML
for quantum physics/chemistry

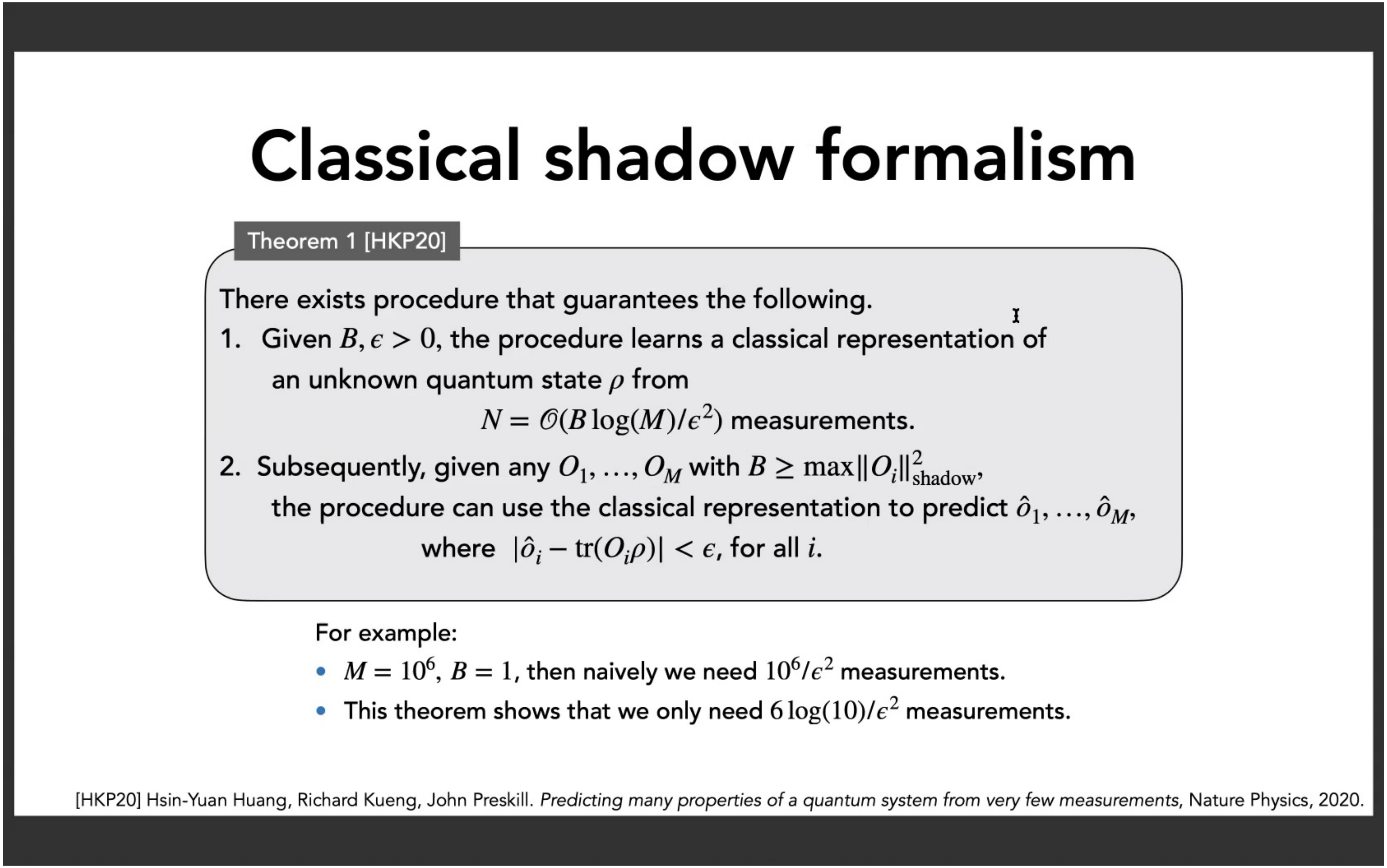
The question 👁:
Can ML algorithms be more useful
than non-ML algorithms in
physically relevant problems?

"Solving the quantum many-body problem with artificial neural networks." *Science* 355.6325 (2017): 602-606.
"Learning phase transitions by confusion." *Nature Physics* 13.5 (2017): 435-439.



Outline

- Review on classical shadow formalism
- Training machines to predict ground states (theory+experiments)
- Training machines to classify quantum phases of matter (theory+experiments)



Classical shadow formalism

Theorem 1 [HKP20]

There exists procedure that guarantees the following.

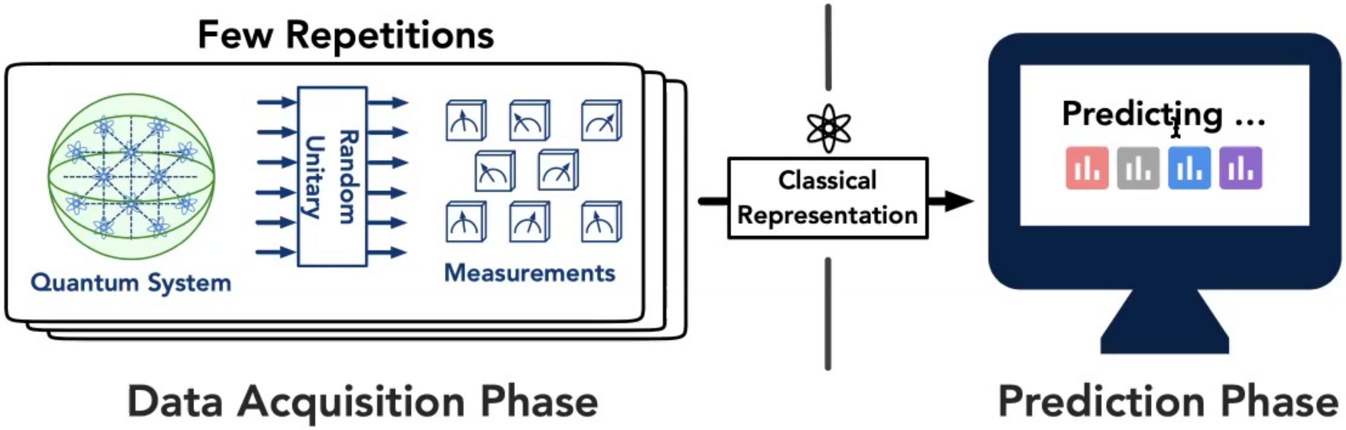
1. Given $B, \epsilon > 0$, the procedure learns a classical representation of an unknown quantum state ρ from
$$N = \mathcal{O}(B \log(M)/\epsilon^2) \text{ measurements.}$$
2. Subsequently, given any $O_1, \ldots, O_M$ with $B \geq \max \|O_i\|^2_{\text{shadow}}$, the procedure can use the classical representation to predict $\hat{o}_1, \ldots, \hat{o}_M$,
$$\text{where } \; |\hat{o}_i - \text{tr}(O_i\rho)| < \epsilon, \text{ for all } i.$$

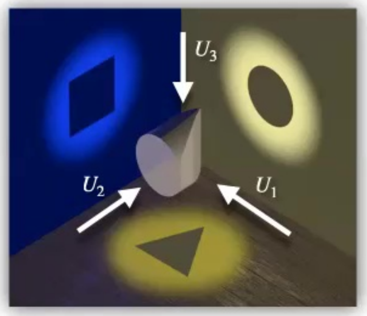
For example:

- $M = 10^6$, $B = 1$, then naively we need $10^6/\epsilon^2$ measurements.
- This theorem shows that we only need $6\log(10)/\epsilon^2$ measurements.

[HKP20] Hsin-Yuan Huang, Richard Kueng, John Preskill. *Predicting many properties of a quantum system from very few measurements*, Nature Physics, 2020.

Classical shadow formalism



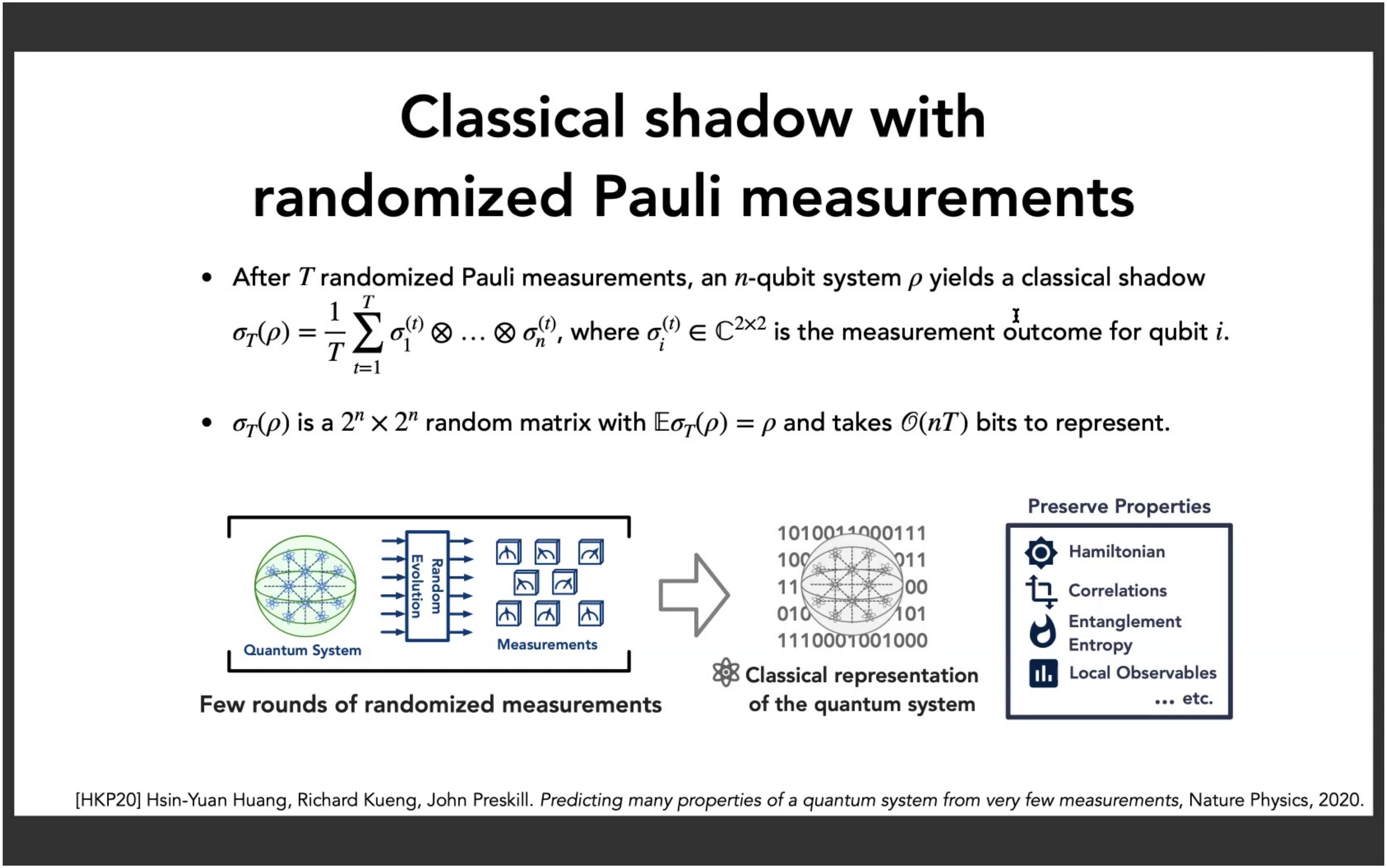


Algorithm for predicting $\mathrm{tr}(O\rho)$: (median-of-means)

Compute $X_i = \mathrm{tr}(O\mathcal{M}^{-1}(|s_i\rangle\langle s_i|)), \forall i = 1,\ldots,N.$

Predict $\hat{o} = \mathrm{median}\left(\frac{1}{N/K}\sum_{i=1}^{N/K} X_i, \ldots, \frac{1}{N/K}\sum_{i=N-N/K+1}^{N} X_i\right).$

[HKP20] Hsin-Yuan Huang, Richard Kueng, John Preskill. *Predicting many properties of a quantum system from very few measurements*, Nature Physics, 2020.



Classical shadow with randomized Pauli measurements

- After T randomized Pauli measurements, an n-qubit system ρ yields a classical shadow $\sigma_T(\rho) = \frac{1}{T}\sum_{t=1}^{T} \sigma_1^{(t)} \otimes \ldots \otimes \sigma_n^{(t)}$, where $\sigma_i^{(t)} \in \mathbb{C}^{2\times 2}$ is the measurement outcome for qubit i.
- $\sigma_T(\rho)$ is a $2^n \times 2^n$ random matrix with $\mathbb{E}\sigma_T(\rho) = \rho$ and takes $\mathcal{O}(nT)$ bits to represent.

Few rounds of randomized measurements

Classical representation of the quantum system

Preserve Properties

- Hamiltonian
- Correlations
- Entanglement Entropy
- Local Observables
- ... etc.

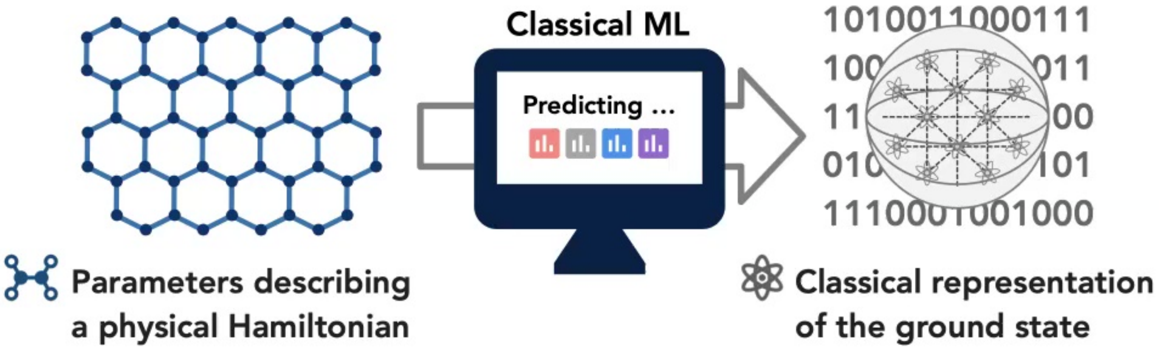
[HKP20] Hsin-Yuan Huang, Richard Kueng, John Preskill. *Predicting many properties of a quantum system from very few measurements*, Nature Physics, 2020.

Predicting ground states: Task

- Given parameters x that describes a Hamiltonian $H(x)$, the machine needs to predict a classical representation of the ground state $\rho(x)$ of $H(x)$.

- $x \in \mathbb{R}^m$ describes laser intensities, few-body interactions, magnetic fields, etc.

- We assume that $x \mapsto H(x)$ is not known exactly. And we represent $\rho(x)$ on a classical computer using its classical shadow $\sigma_T(\rho(x))$.

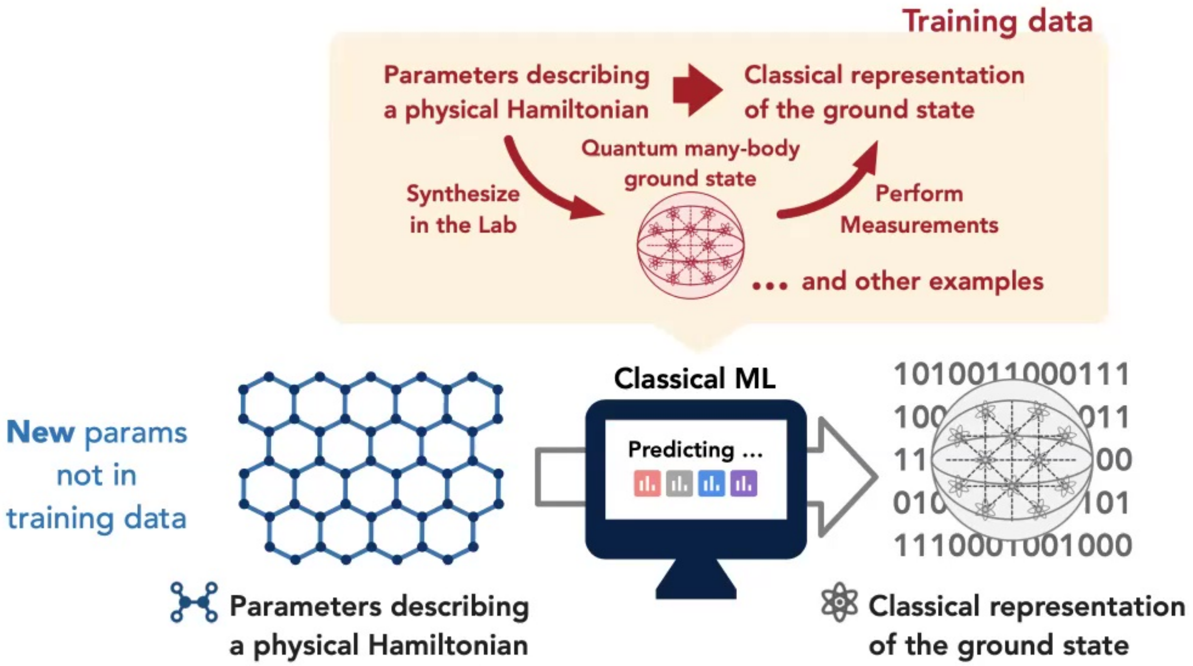


Parameters describing a physical Hamiltonian

Classical representation of the ground state

Predicting ground states: Task

- $x \in \mathbb{R}^m$ describes laser intensities, few-body interactions, magnetic fields, etc. We will normalize such that $x \in [-1,1]^m$.
- Training data: examples of params and associated ground state $\{x_\ell \to \sigma_T(\rho(x_\ell))\}_{\ell=1}^N$.



Training data

Parameters describing a physical Hamiltonian → Classical representation of the ground state

Synthesize in the Lab

Quantum many-body ground state

Perform Measurements

... and other examples

New params not in training data

Classical ML

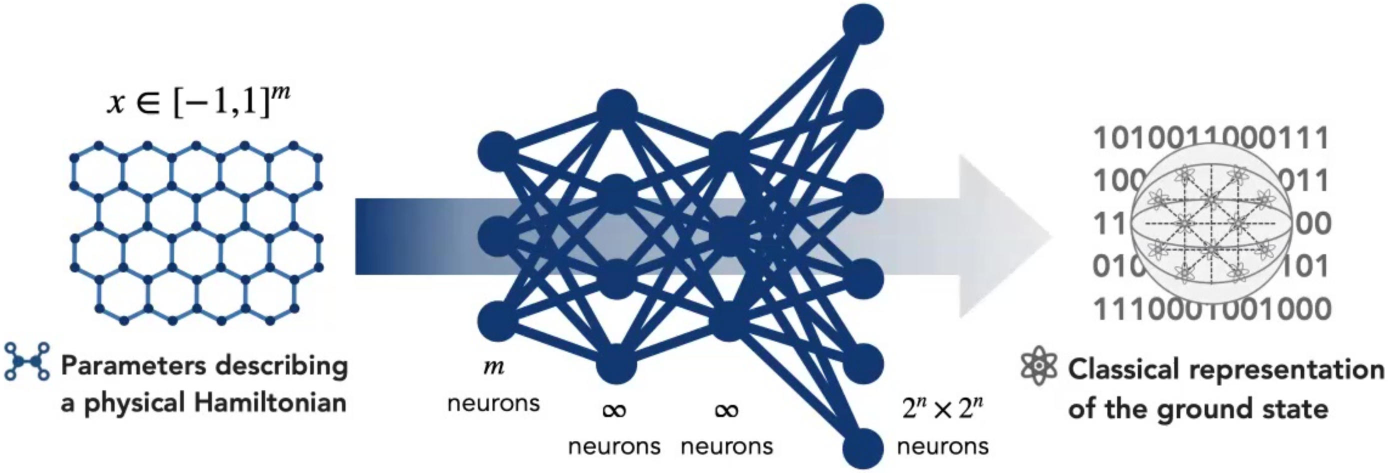
Predicting ...

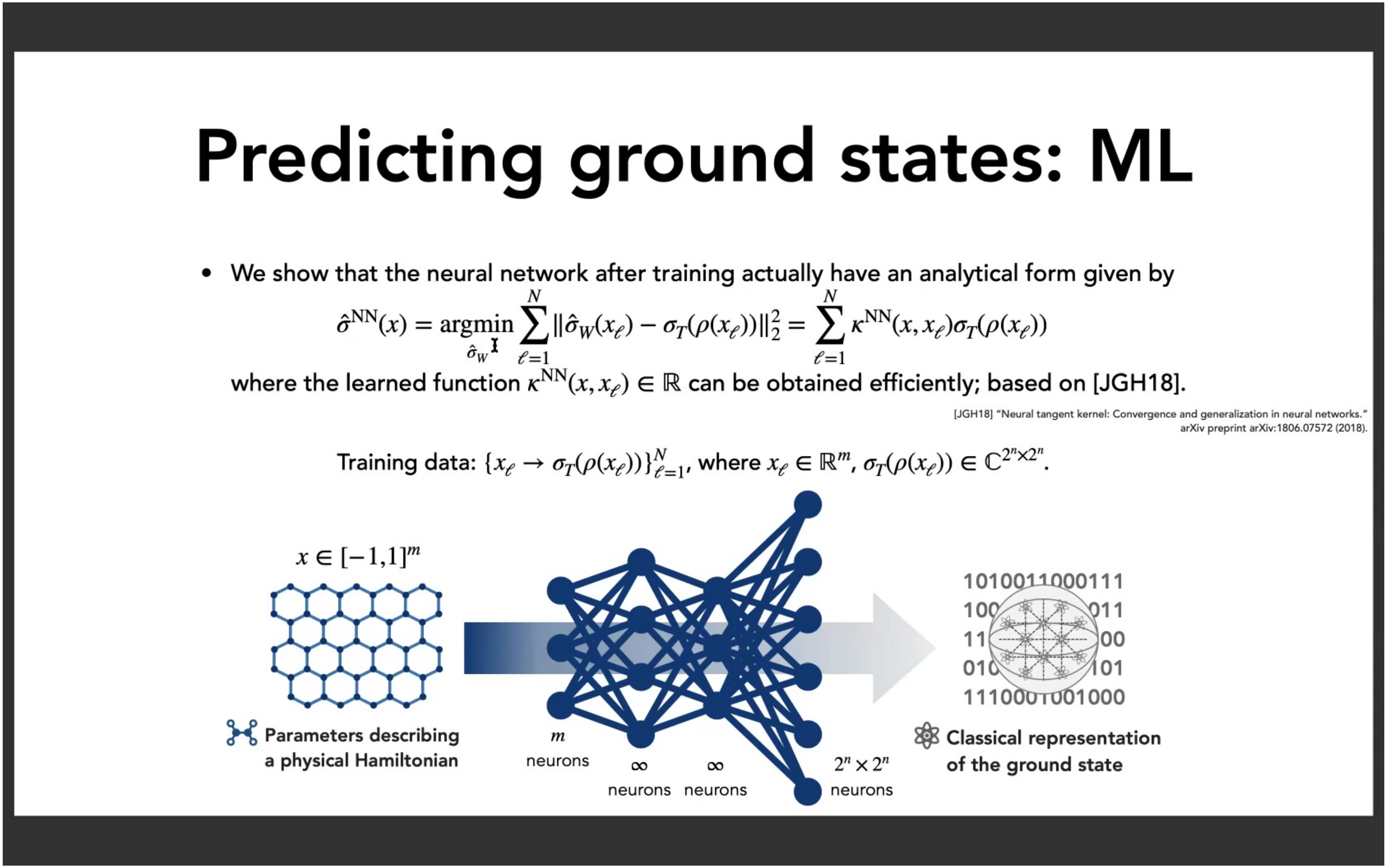
Parameters describing a physical Hamiltonian

Classical representation of the ground state

Predicting ground states: ML

- Training data: $\{x_\ell \to \sigma_T(\rho(x_\ell))\}_{\ell=1}^N$, where $x_\ell \in \mathbb{R}^m$, $\sigma_T(\rho(x_\ell)) \in \mathbb{C}^{2^n \times 2^n}$.
- We consider training an ML model that takes in an m-dim vector x and outputs a $2^n \times 2^n$-size matrix $\hat{\sigma}(x)$; more precisely, an efficient representation of $\hat{\sigma}(x)$.
- The ML model needs to be trained within time polynomial in n, m.





Predicting ground states: ML

- We show that the neural network after training actually have an analytical form given by

$$\hat{\sigma}^{\mathrm{NN}}(x) = \underset{\hat{\sigma}_W}{\operatorname{argmin}} \sum_{\ell=1}^{N} \|\hat{\sigma}_W(x_\ell) - \sigma_T(\rho(x_\ell))\|_2^2 = \sum_{\ell=1}^{N} \kappa^{\mathrm{NN}}(x, x_\ell)\sigma_T(\rho(x_\ell))$$

where the learned function $\kappa^{\mathrm{NN}}(x, x_\ell) \in \mathbb{R}$ can be obtained efficiently; based on [JGH18].

[JGH18] "Neural tangent kernel: Convergence and generalization in neural networks." arXiv preprint arXiv:1806.07572 (2018).

Training data: $\{x_\ell \rightarrow \sigma_T(\rho(x_\ell))\}_{\ell=1}^{N}$, where $x_\ell \in \mathbb{R}^m$, $\sigma_T(\rho(x_\ell)) \in \mathbb{C}^{2^n \times 2^n}$.

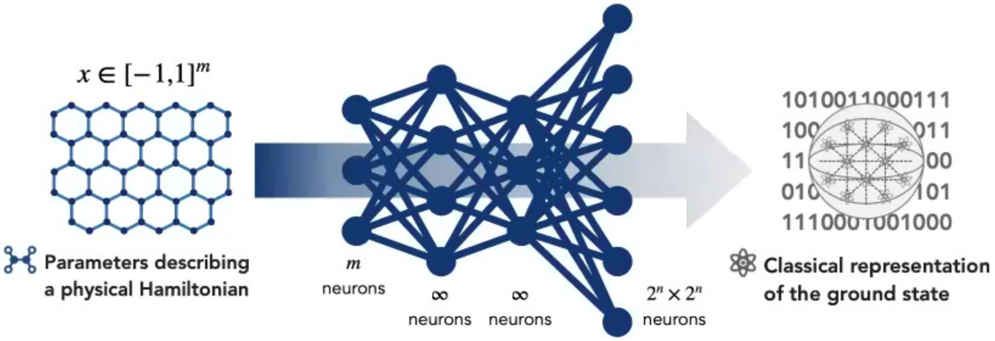
Predicting ground states: ML

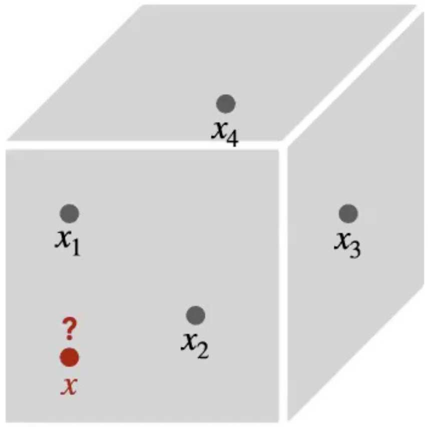
- Furthermore, various machine learning models (kernel methods, infinite-width neural networks, etc.) can be shown to yield an analytical form as the global minimum of the optimization (training):

$$\hat{\sigma}(x) = \sum_{\ell=1}^{N} \kappa(x, x_\ell)\sigma_T(\rho(x_\ell))$$

where $\kappa(x, x_\ell) \in \mathbb{R}$ is a learned function for how to extrapolate the known examples to the full space.

Training data: $\{x_\ell \to \sigma_T(\rho(x_\ell))\}_{\ell=1}^{N}$, where $x_\ell \in \mathbb{R}^m$, $\sigma_T(\rho(x_\ell)) \in \mathbb{C}^{2^n \times 2^n}$.





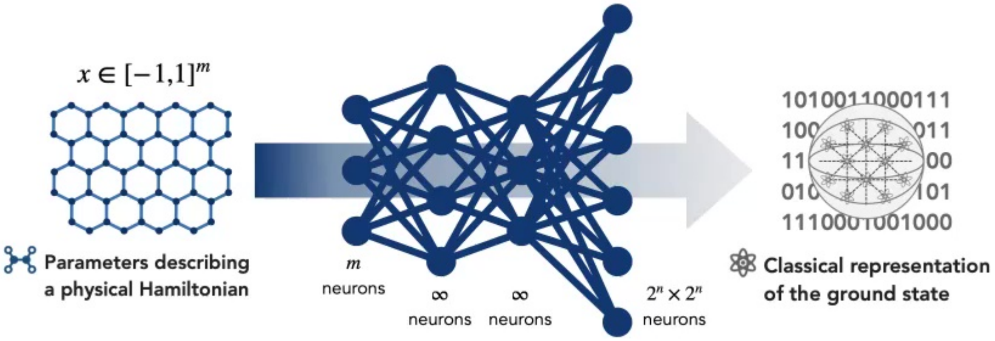
Predicting ground states: ML

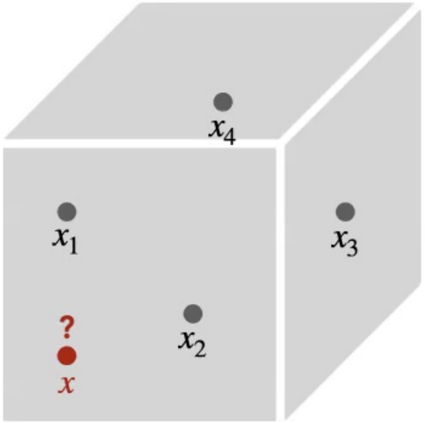
- As long as $\kappa(x, x_\ell) \in \mathbb{R}$ is efficiently computable, the ML model's prediction

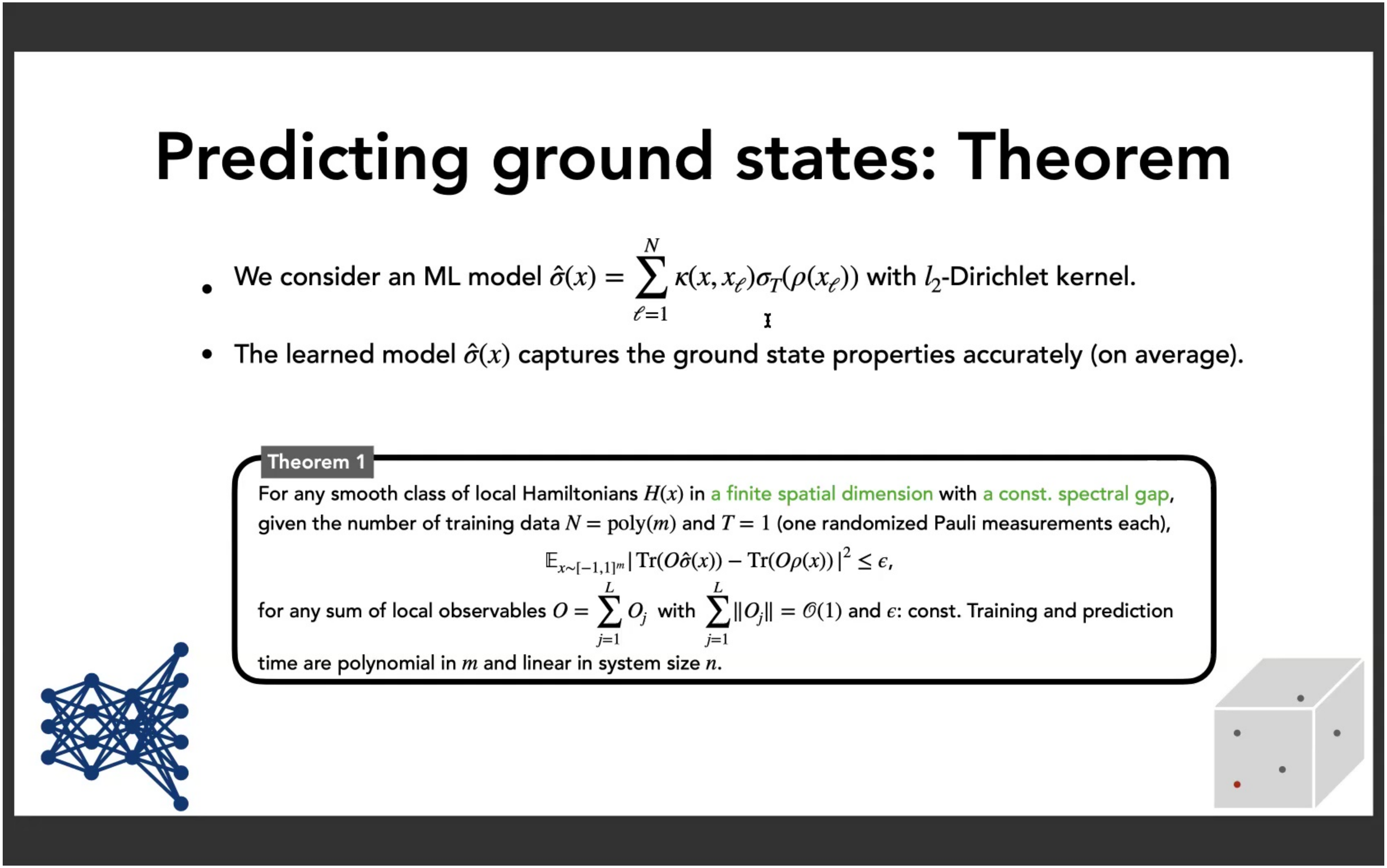
$$\hat{\sigma}(x) = \sum_{\ell=1}^{N} \kappa(x, x_\ell)\sigma_T(\rho(x_\ell))$$

can be represented efficiently with $\mathcal{O}(nTN)$ bits; recall $\sigma_T(\rho(x_\ell))$ only require $\mathcal{O}(nT)$ bits.

Training data: $\{x_\ell \to \sigma_T(\rho(x_\ell))\}_{\ell=1}^{N}$, where $x_\ell \in \mathbb{R}^m$, $\sigma_T(\rho(x_\ell)) \in \mathbb{C}^{2^n \times 2^n}$.







Predicting ground states: Theorem

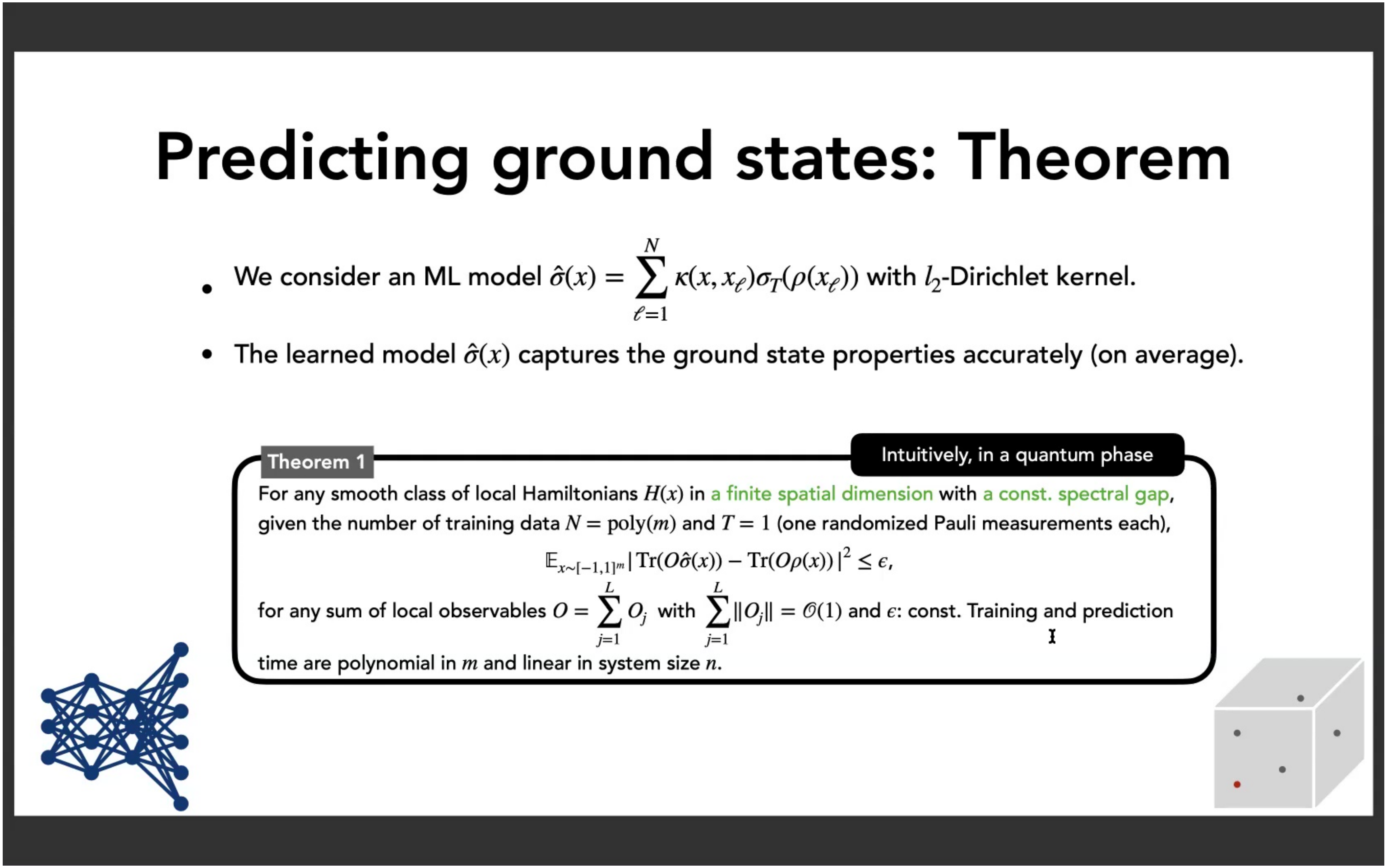
- We consider an ML model $\hat{\sigma}(x) = \sum_{\ell=1}^{N} \kappa(x, x_\ell)\sigma_T(\rho(x_\ell))$ with l_2-Dirichlet kernel.
- The learned model $\hat{\sigma}(x)$ captures the ground state properties accurately (on average).

Theorem 1

For any smooth class of local Hamiltonians $H(x)$ in a finite spatial dimension with a const. spectral gap, given the number of training data $N = \mathrm{poly}(m)$ and $T = 1$ (one randomized Pauli measurements each),

$$\mathbb{E}_{x\sim[-1,1]^m} |\operatorname{Tr}(O\hat{\sigma}(x)) - \operatorname{Tr}(O\rho(x))|^2 \leq \epsilon,$$

for any sum of local observables $O = \sum_{j=1}^{L} O_j$ with $\sum_{j=1}^{L} \|O_j\| = \mathcal{O}(1)$ and ϵ: const. Training and prediction time are polynomial in m and linear in system size n.



Predicting ground states: Theorem

- We consider an ML model $\hat{\sigma}(x) = \sum_{\ell=1}^{N} \kappa(x, x_\ell)\sigma_T(\rho(x_\ell))$ with l_2-Dirichlet kernel.
- The learned model $\hat{\sigma}(x)$ captures the ground state properties accurately (on average).

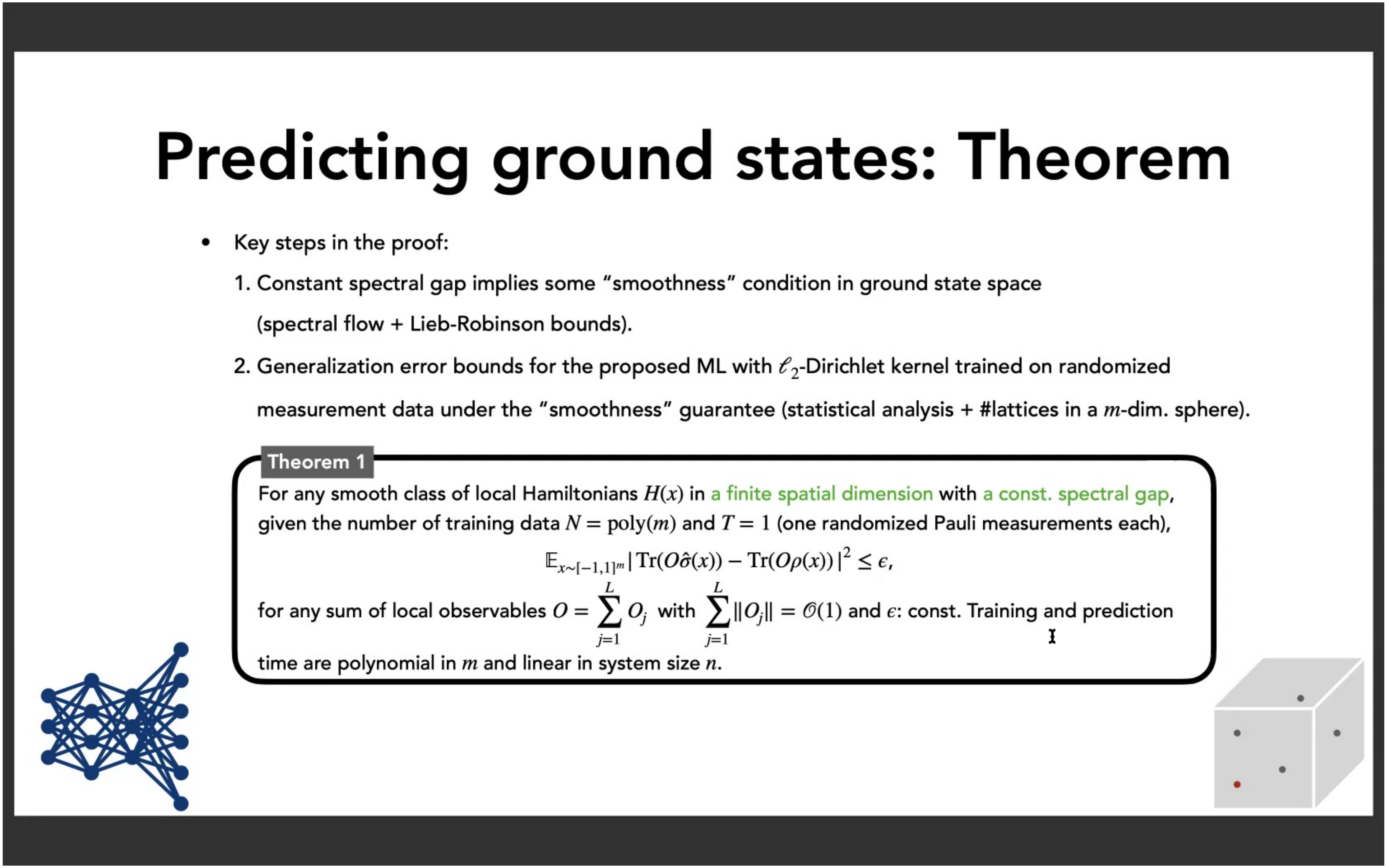
Theorem 1

Intuitively, in a quantum phase

For any smooth class of local Hamiltonians $H(x)$ in a finite spatial dimension with a const. spectral gap, given the number of training data $N = \mathrm{poly}(m)$ and $T = 1$ (one randomized Pauli measurements each),

$$\mathbb{E}_{x\sim[-1,1]^m} |\,\mathrm{Tr}(O\hat{\sigma}(x)) - \mathrm{Tr}(O\rho(x))\,|^2 \leq \epsilon,$$

for any sum of local observables $O = \sum_{j=1}^{L} O_j$ with $\sum_{j=1}^{L} \|O_j\| = \mathcal{O}(1)$ and ϵ: const. Training and prediction time are polynomial in m and linear in system size n.



Predicting ground states: Theorem

- Key steps in the proof:

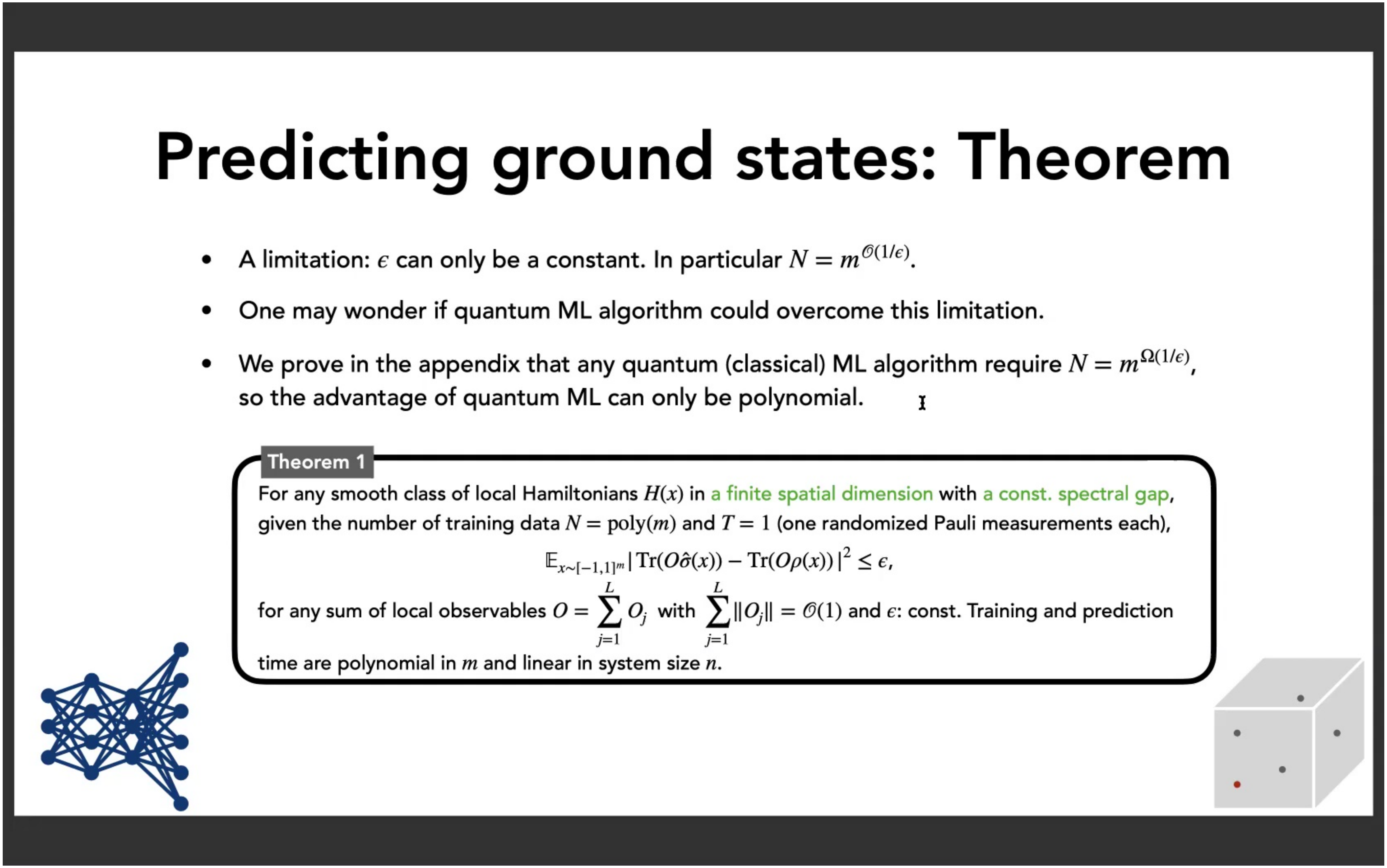
1. Constant spectral gap implies some "smoothness" condition in ground state space (spectral flow + Lieb-Robinson bounds).
2. Generalization error bounds for the proposed ML with ℓ_2-Dirichlet kernel trained on randomized measurement data under the "smoothness" guarantee (statistical analysis + #lattices in a m-dim. sphere).

Theorem 1

For any smooth class of local Hamiltonians $H(x)$ in a finite spatial dimension with a const. spectral gap, given the number of training data $N = \mathrm{poly}(m)$ and $T = 1$ (one randomized Pauli measurements each),

$$\mathbb{E}_{x\sim[-1,1]^m} | \mathrm{Tr}(O\hat{\sigma}(x)) - \mathrm{Tr}(O\rho(x)) |^2 \leq \epsilon,$$

for any sum of local observables $O = \sum_{j=1}^{L} O_j$ with $\sum_{j=1}^{L} \|O_j\| = \mathcal{O}(1)$ and ϵ: const. Training and prediction time are polynomial in m and linear in system size n.



Predicting ground states: Theorem

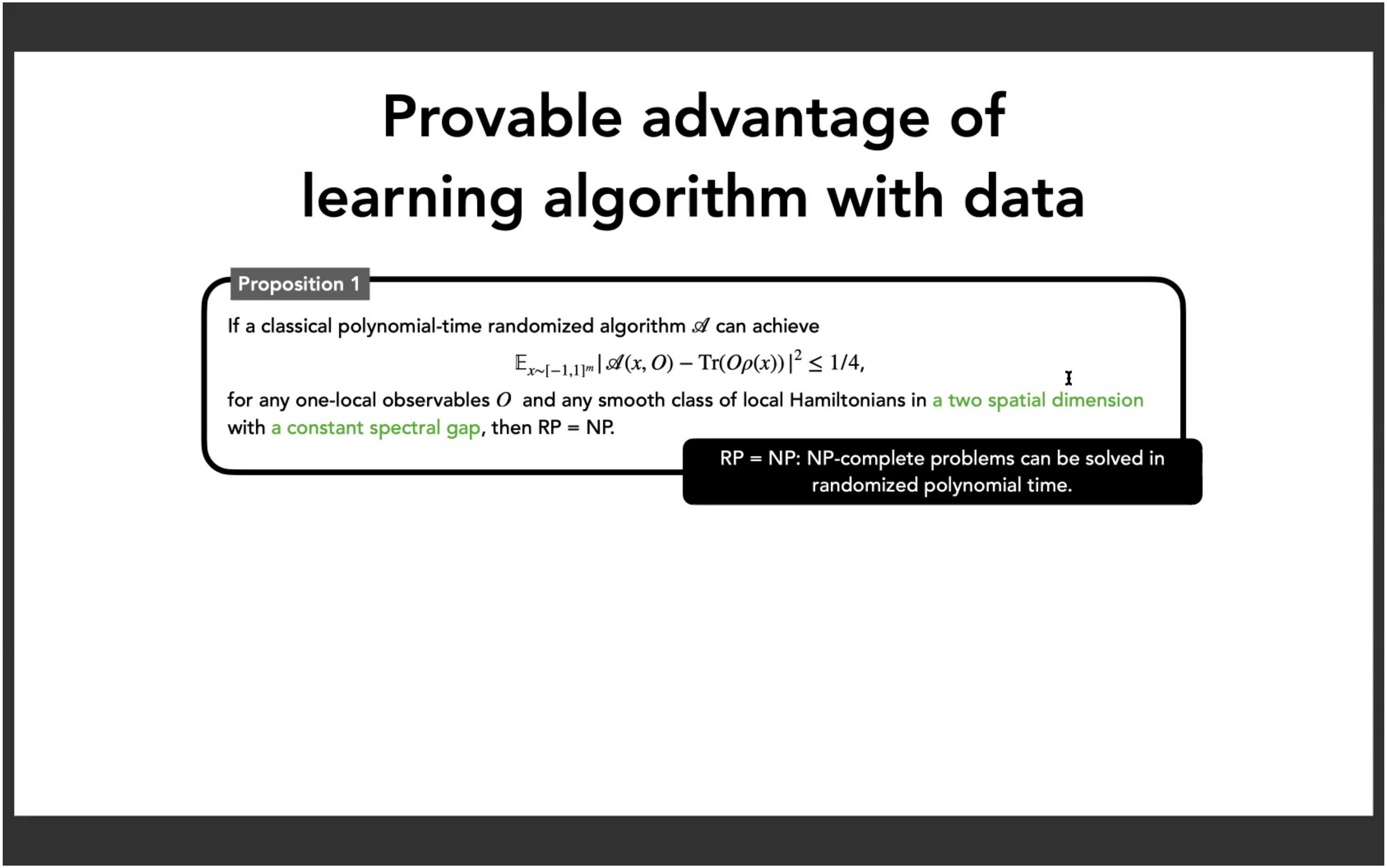
- A limitation: ϵ can only be a constant. In particular $N = m^{\mathcal{O}(1/\epsilon)}$.
- One may wonder if quantum ML algorithm could overcome this limitation.
- We prove in the appendix that any quantum (classical) ML algorithm require $N = m^{\Omega(1/\epsilon)}$, so the advantage of quantum ML can only be polynomial.

Theorem 1

For any smooth class of local Hamiltonians $H(x)$ in a finite spatial dimension with a const. spectral gap, given the number of training data $N = \mathrm{poly}(m)$ and $T = 1$ (one randomized Pauli measurements each),

$$\mathbb{E}_{x \sim [-1,1]^m} |\operatorname{Tr}(O\hat{\sigma}(x)) - \operatorname{Tr}(O\rho(x))|^2 \leq \epsilon,$$

for any sum of local observables $O = \sum_{j=1}^{L} O_j$ with $\sum_{j=1}^{L} \|O_j\| = \mathcal{O}(1)$ and ϵ: const. Training and prediction time are polynomial in m and linear in system size n.



Provable advantage of learning algorithm with data

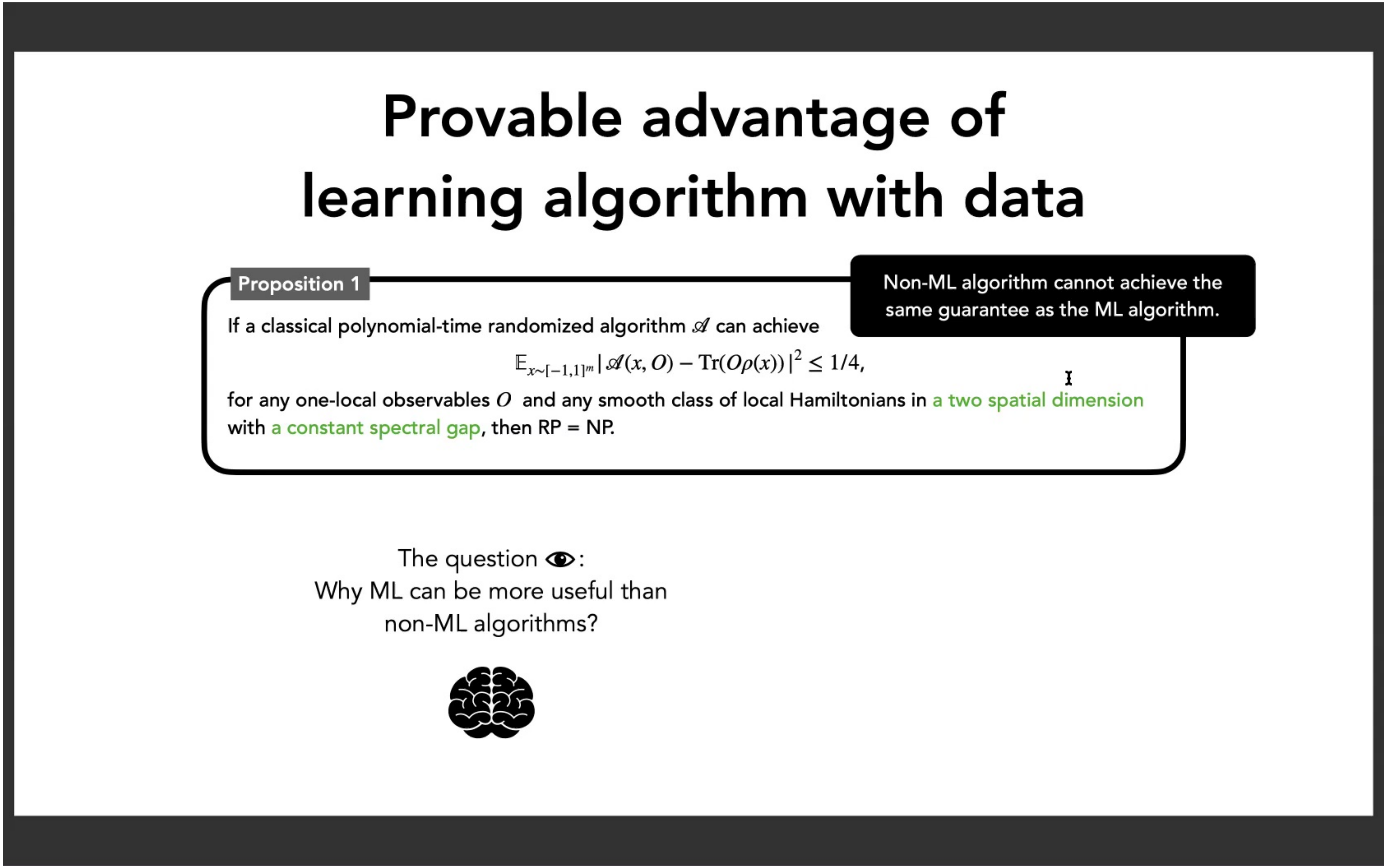
Proposition 1

If a classical polynomial-time randomized algorithm $\mathscr{A}$ can achieve

$$\mathbb{E}_{x\sim[-1,1]^m} | \mathscr{A}(x, O) - \mathrm{Tr}(O\rho(x)) |^2 \leq 1/4,$$

for any one-local observables O and any smooth class of local Hamiltonians in a two spatial dimension with a constant spectral gap, then RP = NP.

RP = NP: NP-complete problems can be solved in randomized polynomial time.



Provable advantage of learning algorithm with data

Non-ML algorithm cannot achieve the same guarantee as the ML algorithm.

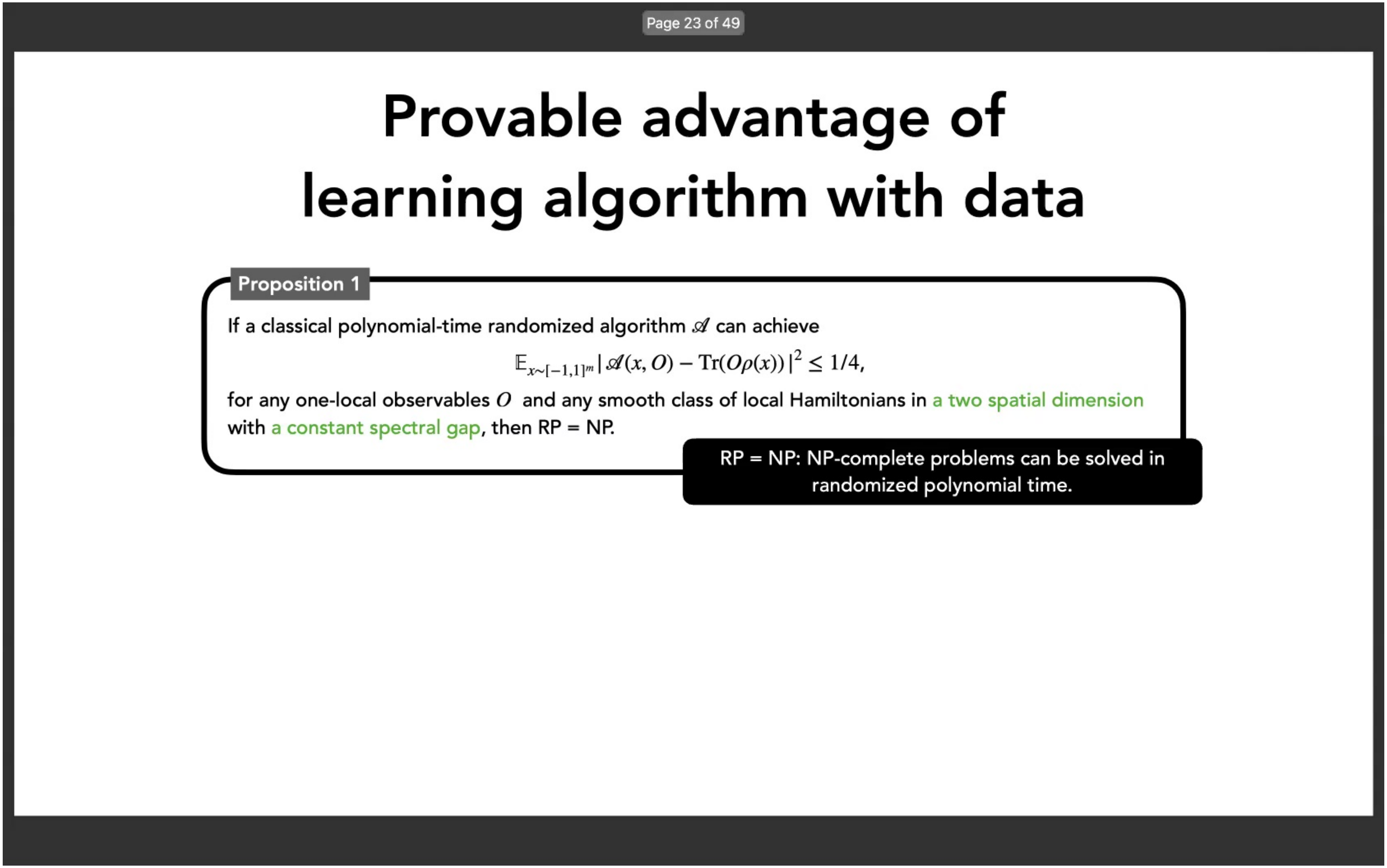
Proposition 1

If a classical polynomial-time randomized algorithm $\mathscr{A}$ can achieve

$$\mathbb{E}_{x\sim[-1,1]^m} | \mathscr{A}(x, O) - \mathrm{Tr}(O\rho(x)) |^2 \leq 1/4,$$

for any one-local observables O and any smooth class of local Hamiltonians in a two spatial dimension with a constant spectral gap, then RP = NP.

The question 👁:
Why ML can be more useful than
non-ML algorithms?



Provable advantage of learning algorithm with data

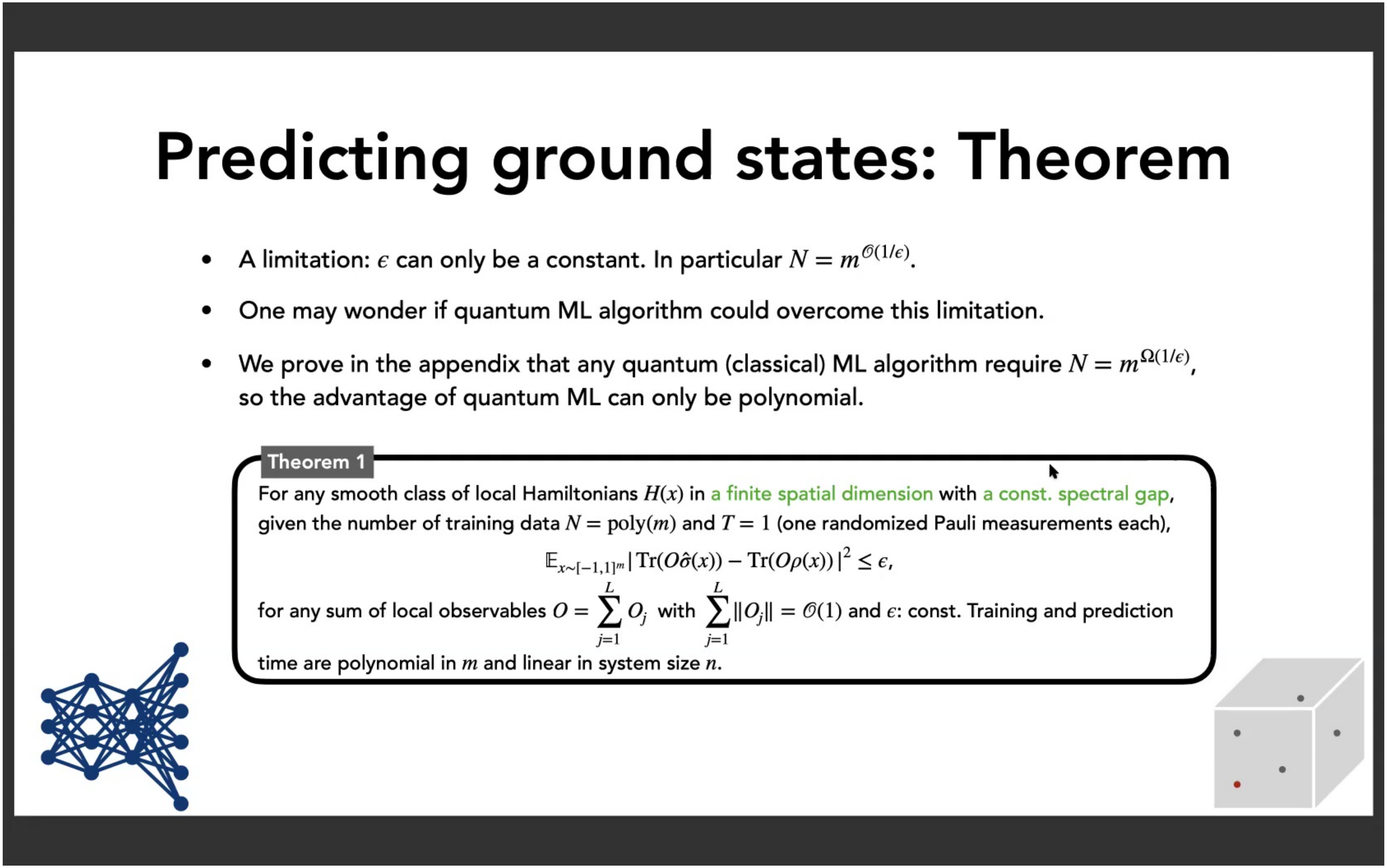
Proposition 1

If a classical polynomial-time randomized algorithm $\mathscr{A}$ can achieve

$$\mathbb{E}_{x\sim[-1,1]^m}\,|\,\mathscr{A}(x,O)-\mathrm{Tr}(O\rho(x))\,|^2\leq 1/4,$$

for any one-local observables O and any smooth class of local Hamiltonians in a two spatial dimension with a constant spectral gap, then RP = NP.

RP = NP: NP-complete problems can be solved in randomized polynomial time.



Predicting ground states: Theorem

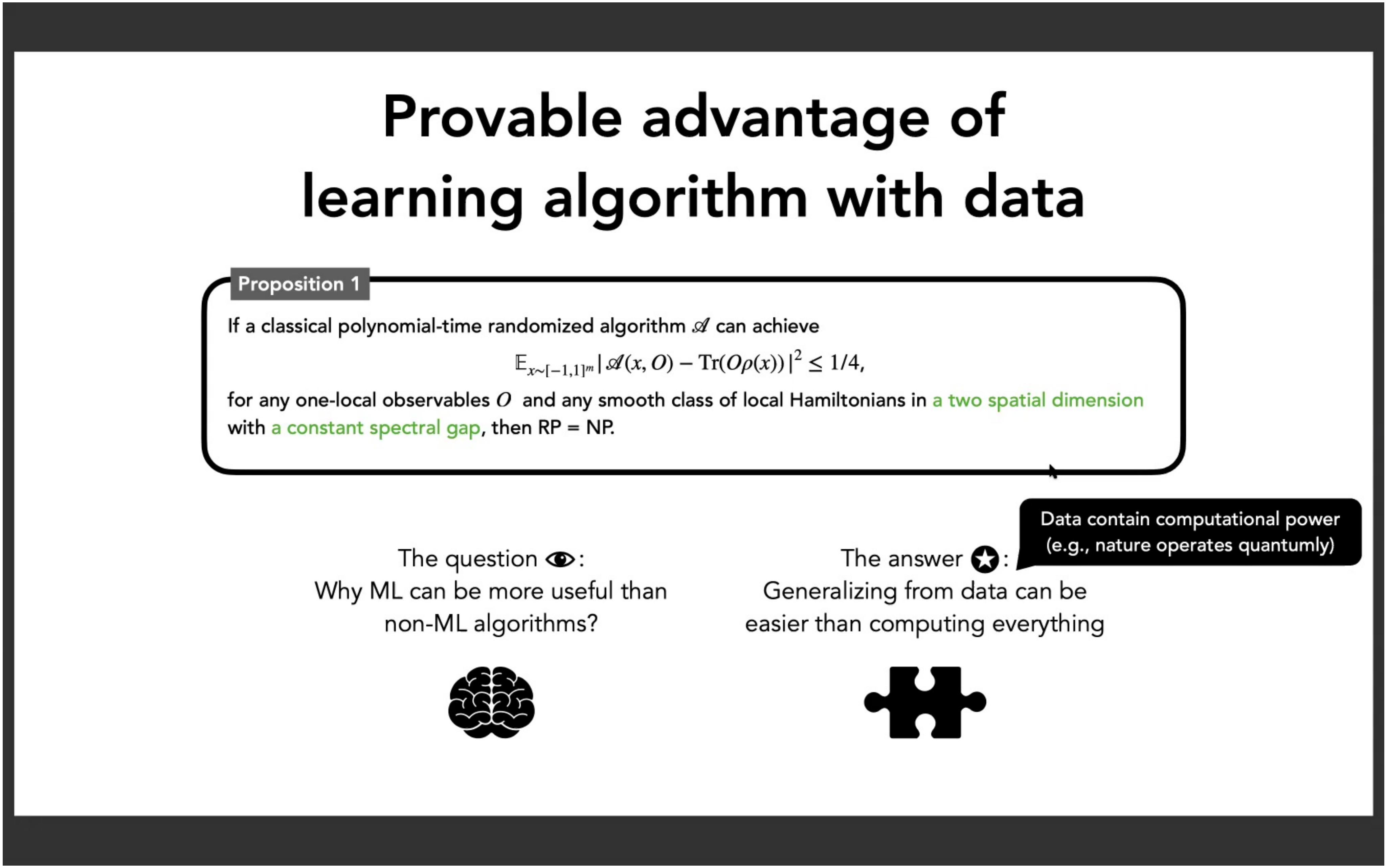
- A limitation: ϵ can only be a constant. In particular $N = m^{\mathcal{O}(1/\epsilon)}$.
- One may wonder if quantum ML algorithm could overcome this limitation.
- We prove in the appendix that any quantum (classical) ML algorithm require $N = m^{\Omega(1/\epsilon)}$, so the advantage of quantum ML can only be polynomial.

Theorem 1

For any smooth class of local Hamiltonians $H(x)$ in a finite spatial dimension with a const. spectral gap, given the number of training data $N = \text{poly}(m)$ and $T = 1$ (one randomized Pauli measurements each),

$$\mathbb{E}_{x\sim[-1,1]^m} | \operatorname{Tr}(O\hat{\sigma}(x)) - \operatorname{Tr}(O\rho(x)) |^2 \leq \epsilon,$$

for any sum of local observables $O = \sum_{j=1}^{L} O_j$ with $\sum_{j=1}^{L} \|O_j\| = \mathcal{O}(1)$ and ϵ: const. Training and prediction time are polynomial in m and linear in system size n.



Provable advantage of learning algorithm with data

Proposition 1

If a classical polynomial-time randomized algorithm $\mathscr{A}$ can achieve

$$\mathbb{E}_{x\sim[-1,1]^m} |\mathscr{A}(x, O) - \mathrm{Tr}(O\rho(x))|^2 \leq 1/4,$$

for any one-local observables O and any smooth class of local Hamiltonians in a two spatial dimension with a constant spectral gap, then RP = NP.

The question 👁:
Why ML can be more useful than non-ML algorithms?

The answer ★:
Generalizing from data can be easier than computing everything

Data contain computational power (e.g., nature operates quantumly)

Predicting ground states: Theorem

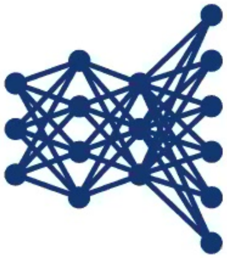
- A limitation: ϵ can only be a constant. In particular $N = m^{\mathcal{O}(1/\epsilon)}$.
- One may wonder if quantum ML algorithm could overcome this limitation.
- We prove in the appendix that any quantum (classical) ML algorithm require $N = m^{\Omega(1/\epsilon)}$, so the advantage of quantum ML can only be polynomial.

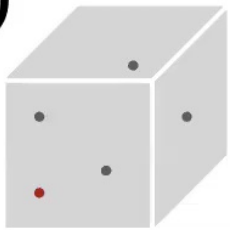
Theorem 1

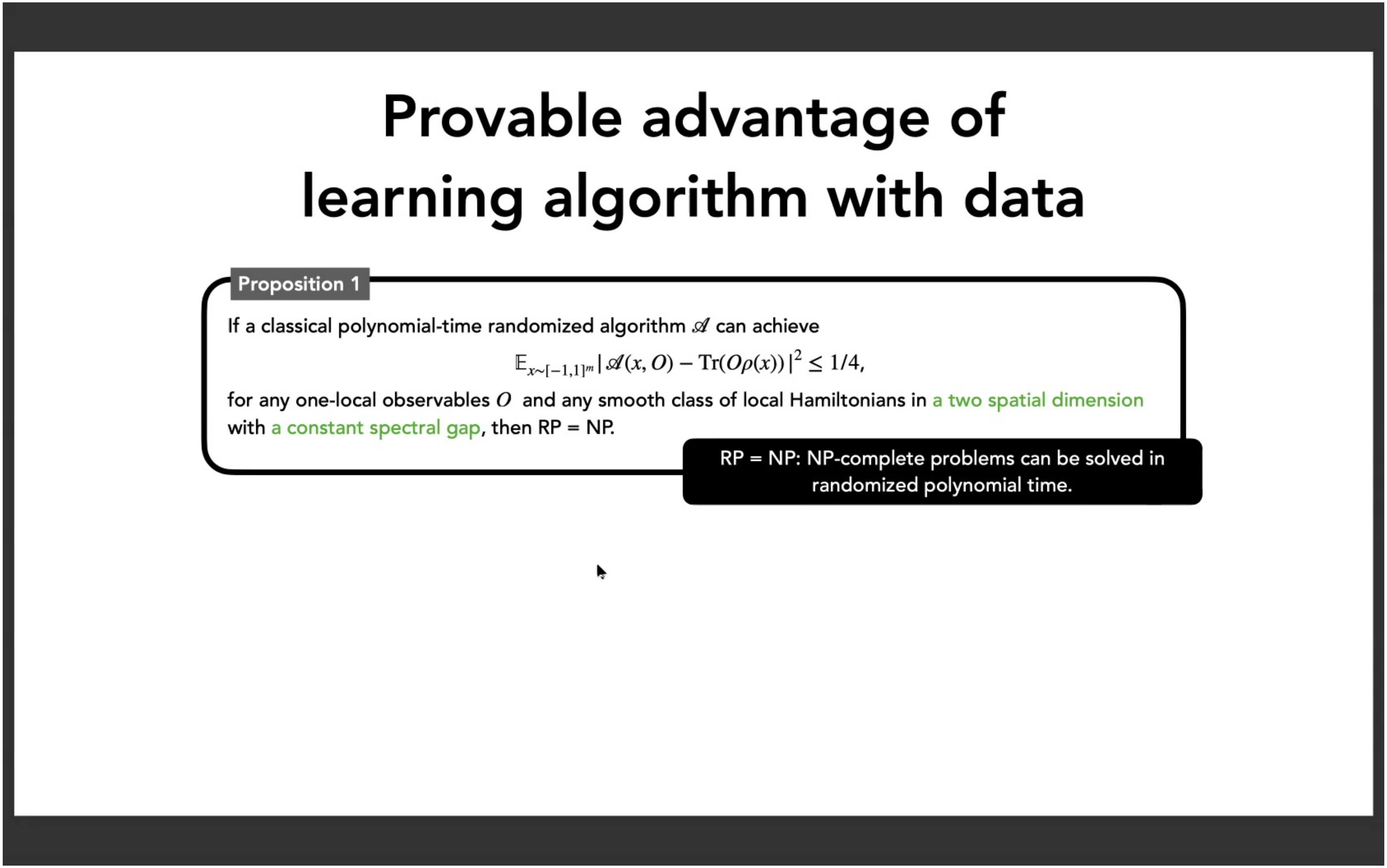
For any smooth class of local Hamiltonians $H(x)$ in a finite spatial dimension with a const. spectral gap, given the number of training data $N = \mathrm{poly}(m)$ and $T = 1$ (one randomized Pauli measurements each),

$$\mathbb{E}_{x \sim [-1,1]^m} |\operatorname{Tr}(O\hat{\sigma}(x)) - \operatorname{Tr}(O\rho(x))|^2 \leq \epsilon,$$

for any sum of local observables $O = \sum_{j=1}^{L} O_j$ with $\sum_{j=1}^{L} \|O_j\| = \mathcal{O}(1)$ and ϵ: const. Training and prediction time are polynomial in m and linear in system size n.







Provable advantage of learning algorithm with data

Proposition 1

If a classical polynomial-time randomized algorithm $\mathscr{A}$ can achieve

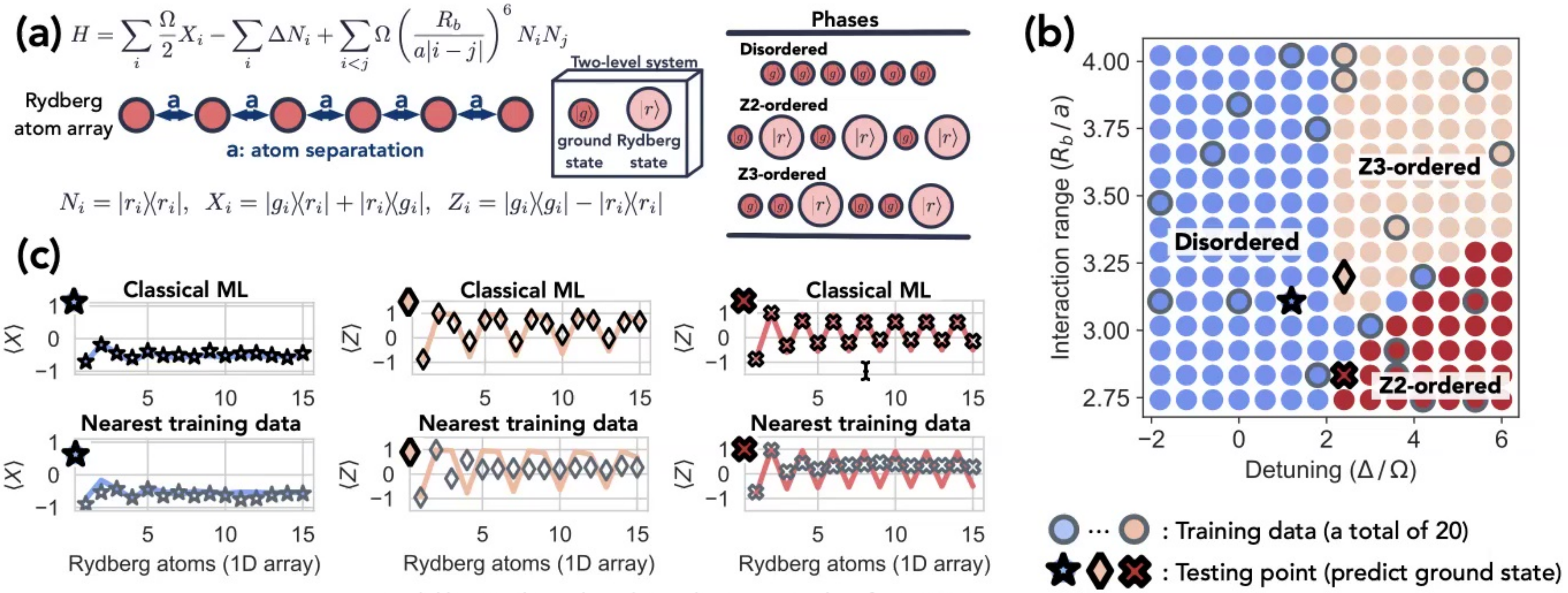
$$\mathbb{E}_{x\sim[-1,1]^m}\,|\,\mathscr{A}(x,O)-\mathrm{Tr}(O\rho(x))\,|^2 \leq 1/4,$$

for any one-local observables O and any smooth class of local Hamiltonians in a two spatial dimension with a constant spectral gap, then RP = NP.

RP = NP: NP-complete problems can be solved in randomized polynomial time.

1D Rydberg atom array

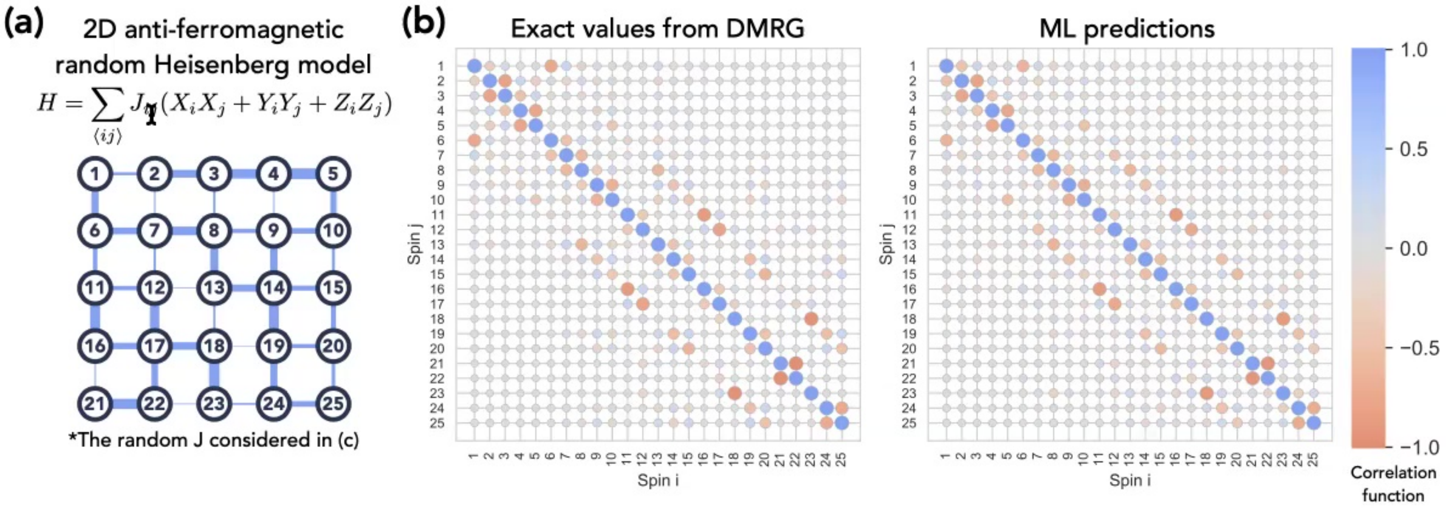
We consider training data size $N = 20$, $T = 500$ randomized measurements for constructing classical shadows.
The best ML model is chosen from Gaussian kernel method, infinite-width neural networks, and l_2-Dirichlet kernel.

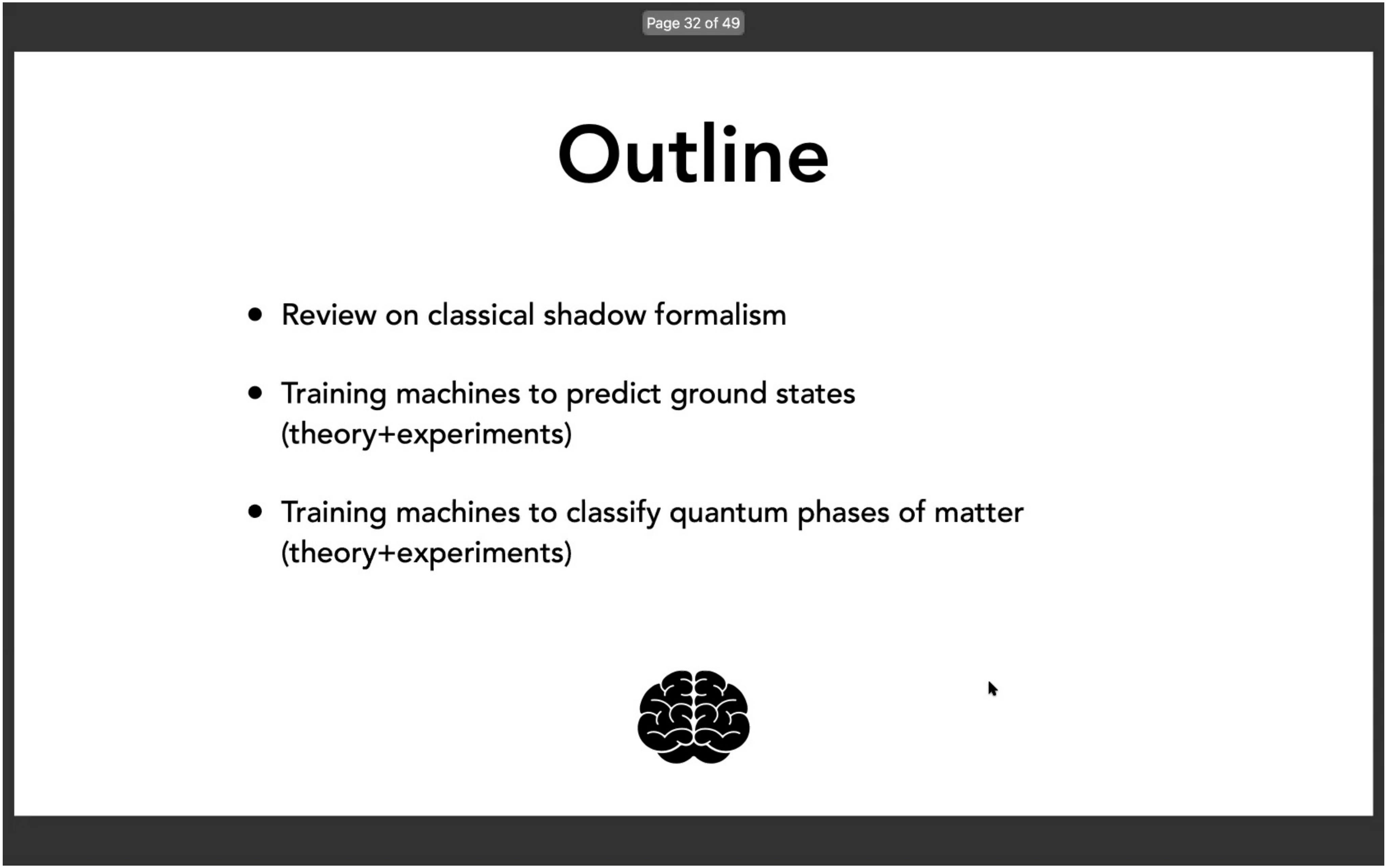


*Solid lines in the six line plots indicate exact values from DMRG

2D random Heisenberg model

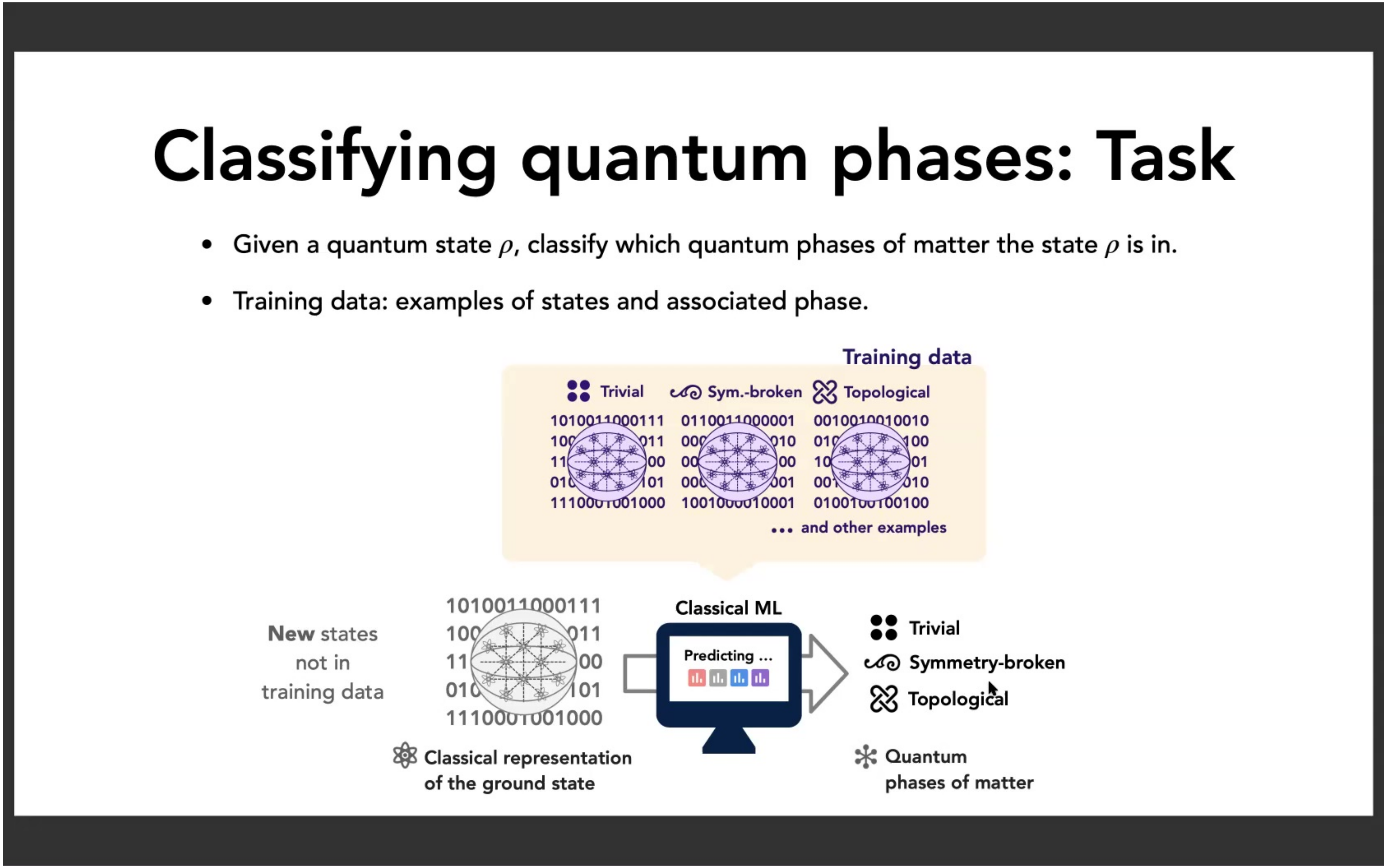
We consider training data size $N = 100$, $T = 500$ randomized measurements for constructing classical shadows. The best ML model is chosen from Gaussian kernel method, infinite-width neural networks, and l_2-Dirichlet kernel.





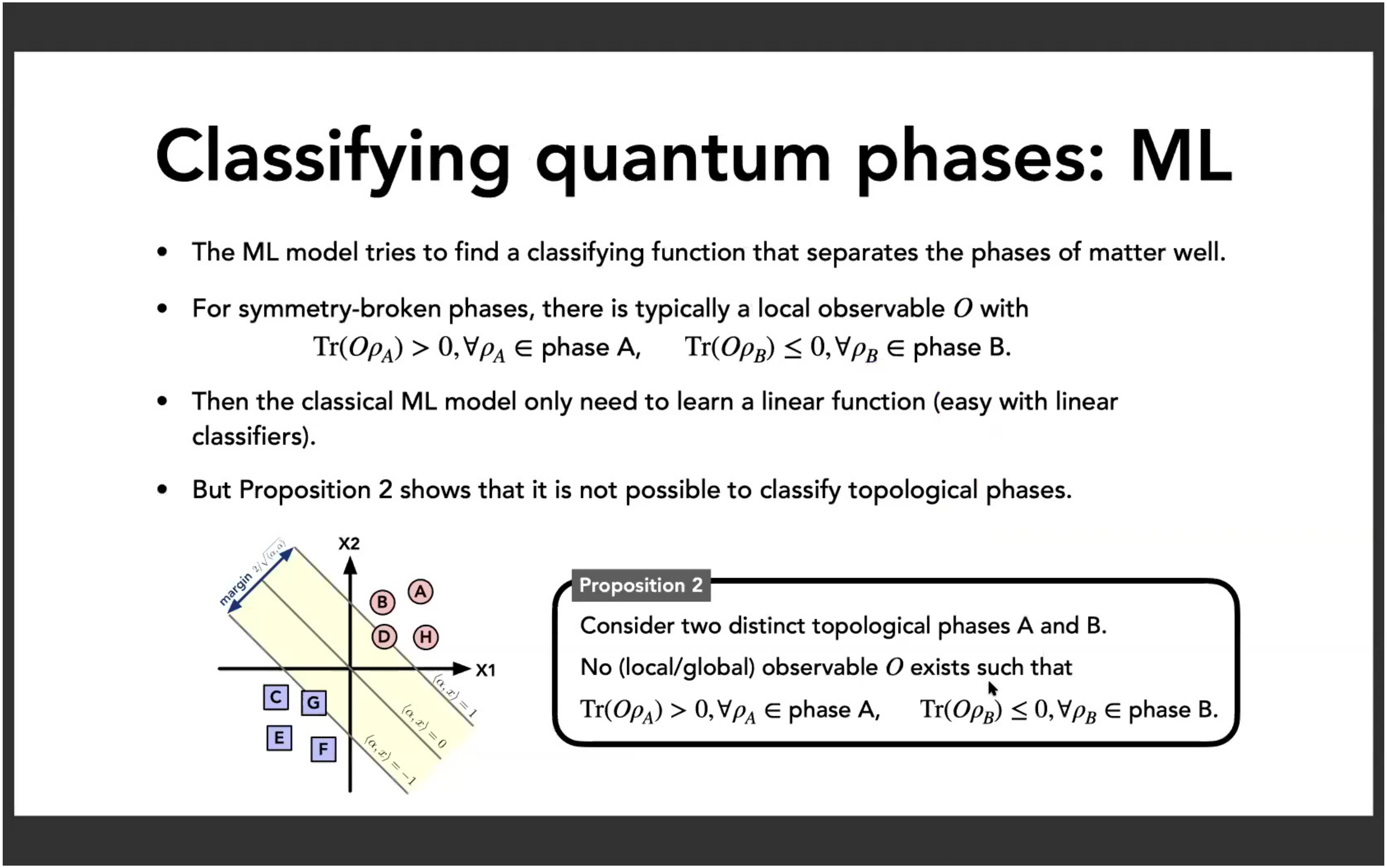
Outline

- Review on classical shadow formalism
- Training machines to predict ground states (theory+experiments)
- Training machines to classify quantum phases of matter (theory+experiments)



Classifying quantum phases: Task

- Given a quantum state ρ, classify which quantum phases of matter the state ρ is in.
- Training data: examples of states and associated phase.



Classifying quantum phases: ML

- The ML model tries to find a classifying function that separates the phases of matter well.
- For symmetry-broken phases, there is typically a local observable O with
  $$\mathrm{Tr}(O\rho_A) > 0, \forall \rho_A \in \text{phase A}, \qquad \mathrm{Tr}(O\rho_B) \leq 0, \forall \rho_B \in \text{phase B}.$$
- Then the classical ML model only need to learn a linear function (easy with linear classifiers).
- But Proposition 2 shows that it is not possible to classify topological phases.

margin $2/\sqrt{\langle\alpha,\alpha\rangle}$
X2
X1
A
B
D
H
C
G
E
F
$\langle\alpha,x\rangle = 1$
$\langle\alpha,x\rangle = 0$
$\langle\alpha,x\rangle = -1$

Proposition 2

Consider two distinct topological phases A and B.

No (local/global) observable O exists such that

$$\mathrm{Tr}(O\rho_A) > 0, \forall \rho_A \in \text{phase A}, \qquad \mathrm{Tr}(O\rho_B) \leq 0, \forall \rho_B \in \text{phase B}.$$

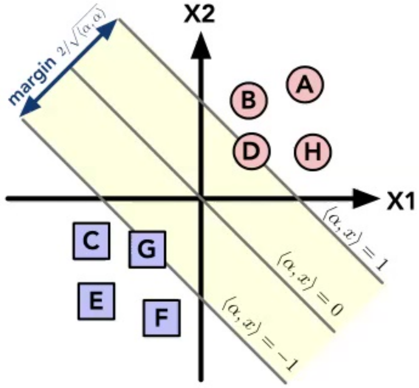
Classifying quantum phases: ML

- We need a more powerful ML model that can learn nonlinear functions, such as
  $\mathrm{Tr}(O\rho \otimes \rho)$, $\mathrm{Tr}(O\rho^{\otimes d})$, or a general analytic function $f(\rho)$.

- To do so, we consider learning a linear function in an ∞-dim space, where each state ρ is mapped to

$$\phi^{(\mathrm{shadow})}(S_T(\rho)) = \lim_{D,R\to\infty} \bigoplus_{d=0}^{D} \sqrt{\frac{\tau^d}{d!}} \left(\bigoplus_{r=0}^{R} \sqrt{\frac{1}{r!}\left(\frac{\gamma}{n}\right)^r} \bigoplus_{i_1=1}^{n} \dots \bigoplus_{i_r=1}^{n} \mathrm{vec}\left[\frac{1}{T}\sum_{t=1}^{T} \bigotimes_{\ell=1}^{r} \sigma_{i_\ell}^{(t)}\right]\right)^{\otimes d}.$$

- It consists of arbitrarily-large r-body reduced density matrices and arbitrarily-high-degree expansion.


margin $2/\sqrt{\langle \alpha, \alpha \rangle}$
X2
X1
A
B
D
H
C
G
E
F
$\langle \alpha, x \rangle = 1$
$\langle \alpha, x \rangle = 0$
$\langle \alpha, x \rangle = -1$

Proposition 2

Consider two distinct topological phases A and B.

No (local/global) observable O exists such that

$\mathrm{Tr}(O\rho_A) > 0, \forall \rho_A \in$ phase A, $\mathrm{Tr}(O\rho_B) \leq 0, \forall \rho_B \in$ phase B.

Classifying quantum phases: ML

- We need a more powerful ML model that can learn nonlinear functions, such as $\mathrm{Tr}(O\rho \otimes \rho)$, $\mathrm{Tr}(O\rho^{\otimes d})$, or a general analytic function $f(\rho)$.

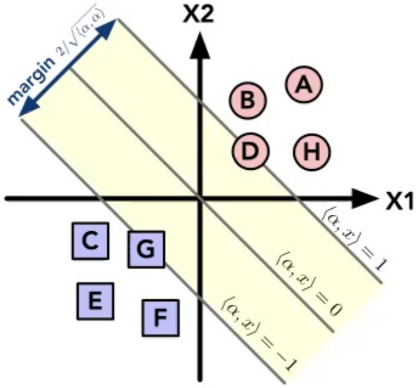
- To do so, we consider learning a linear function i

Concatenate all degree

ρ is mapped to

$$\phi^{(\mathrm{shadow})}(S_T(\rho)) = \lim_{D,R\to\infty} \bigoplus_{d=0}^{D} \sqrt{\frac{\tau^d}{d!}} \left(\bigoplus_{r=0}^{R} \sqrt{\frac{1}{r!}\left(\frac{\gamma}{n}\right)^r} \bigoplus_{i_1=1}^{n} \cdots \bigoplus_{i_r=1}^{n} \mathrm{vec}\left[\frac{1}{T}\sum_{t=1}^{T}\bigotimes_{\ell=1}^{r}\sigma_{i_\ell}^{(t)}\right]\right)^{\otimes d}.$$

- It consists of arbitrarily-large r-body reduced density matrices and arbitrarily-high-degree expansion.



X2
X1
margin $2/\sqrt{\langle\alpha,\alpha\rangle}$
A
B
D
H
C
G
E
F
$\langle\alpha,x\rangle=1$
$\langle\alpha,x\rangle=0$
$\langle\alpha,x\rangle=-1$

Proposition 2

Consider two distinct topological phases A and B.

No (local/global) observable O exists such that

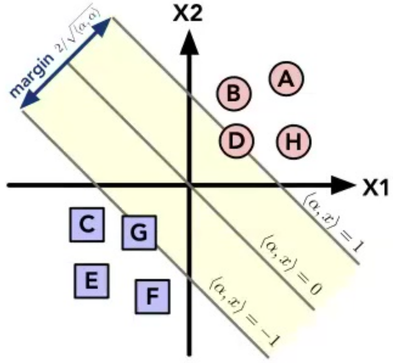
$\mathrm{Tr}(O\rho_A) > 0, \forall \rho_A \in$ phase A, $\quad \mathrm{Tr}(O\rho_B) \leq 0, \forall \rho_B \in$ phase B.

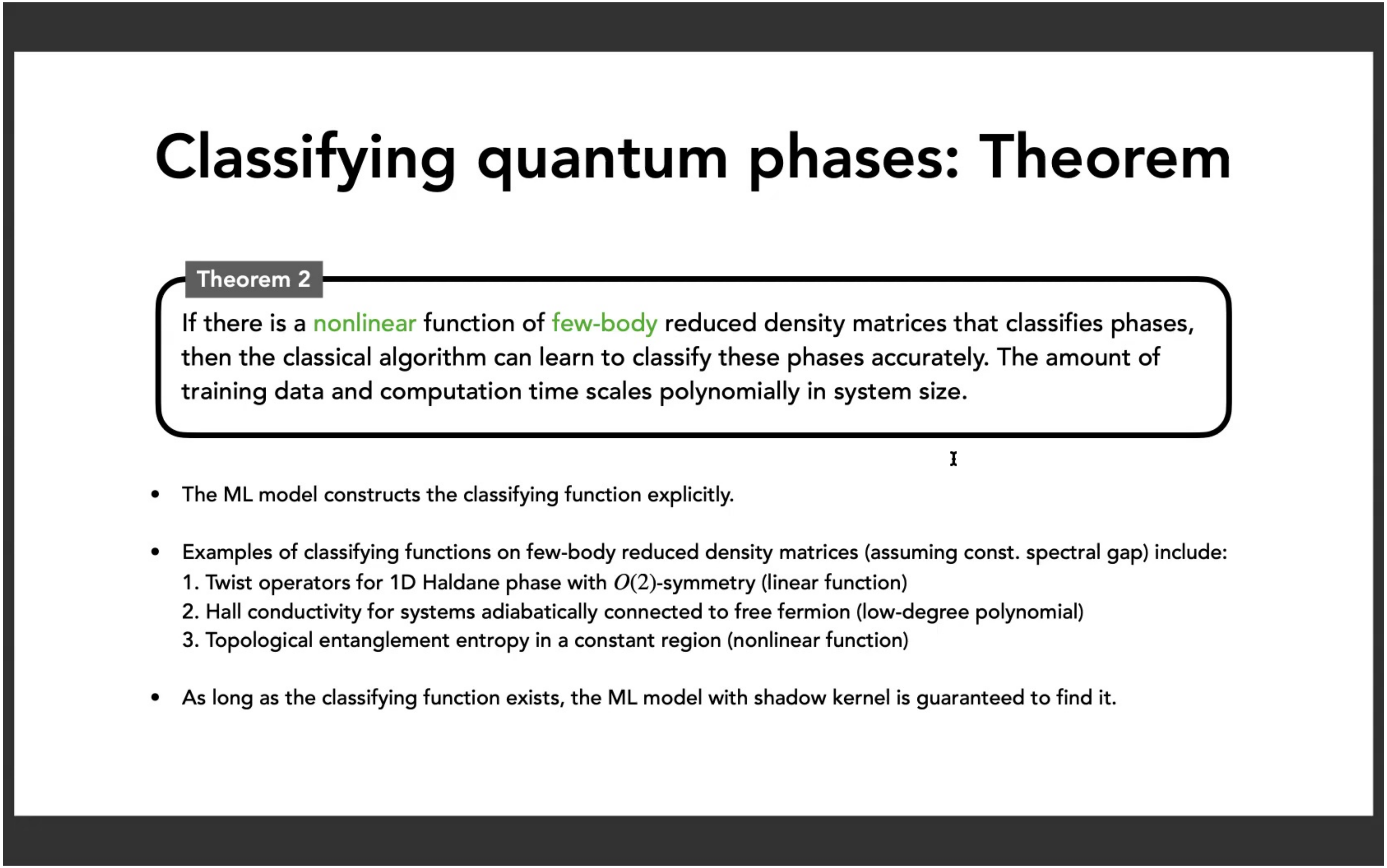
Classifying quantum phases: ML

- Classical ML model: Learn a linear function in $\phi^{(\text{shadow})}(S_T(\rho))$ equiv. a nonlinear function in ρ.
- All we need is to efficiently compute the inner product (referred to as shadow kernel)

$$\langle \phi^{(\text{shadow})}(S_T(\rho)), \phi^{(\text{shadow})}(S_T(\tilde{\rho})) \rangle = \exp\left(\frac{\tau}{T^2} \sum_{t,t'=1}^{T} \exp\left(\frac{\gamma}{n} \sum_{i=1}^{n} \mathrm{Tr}\left(\sigma_i^{(t)} \tilde{\sigma}_i^{(t')} \right) \right) \right) \equiv k^{(\text{shadow})}\left(S_T(\rho), S_T(\tilde{\rho}) \right).$$

- Computing shadow kernels only take time $\mathcal{O}(nT^2)$.
- Training the classical ML model only take time polynomial in n, T, N (and extremely efficient in practice).


X2
X1
margin $2/\sqrt{\langle \alpha, \alpha \rangle}$
A
B
D
H
C
G
E
F
$\langle \alpha, x \rangle = 1$
$\langle \alpha, x \rangle = 0$
$\langle \alpha, x \rangle = -1$



Classifying quantum phases: Theorem

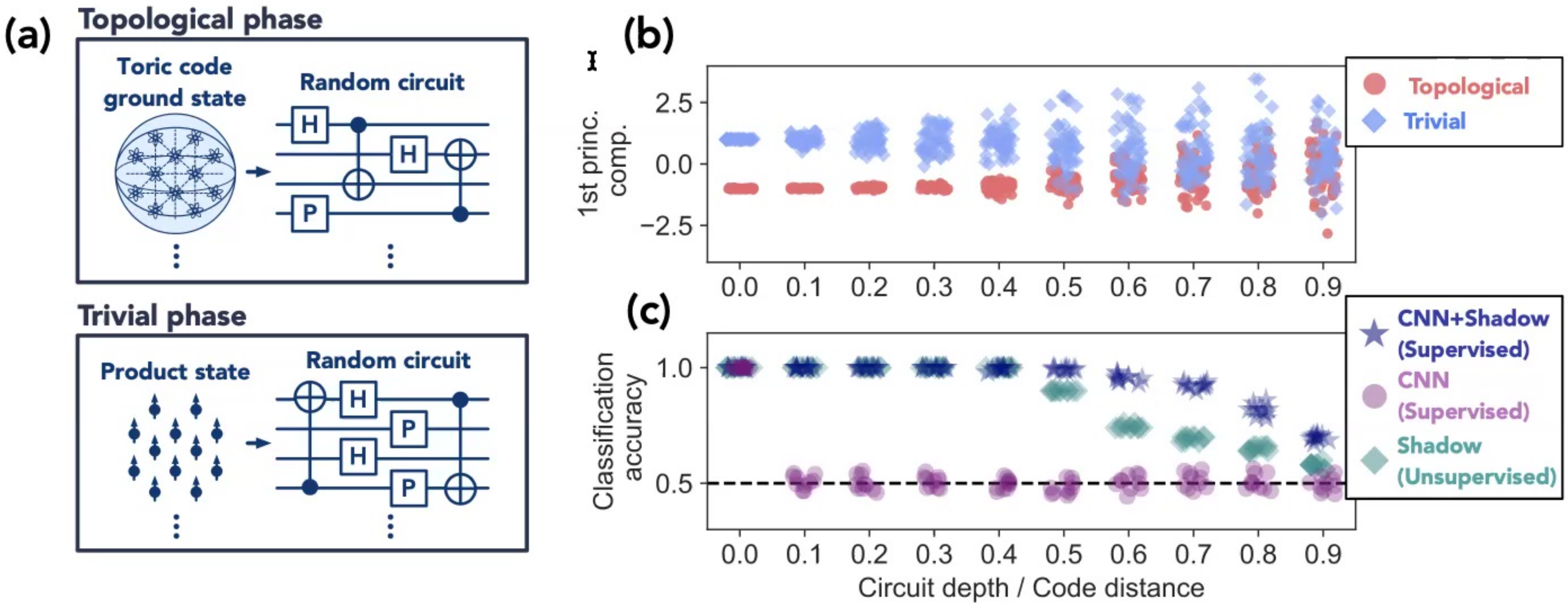
Theorem 2

If there is a nonlinear function of few-body reduced density matrices that classifies phases, then the classical algorithm can learn to classify these phases accurately. The amount of training data and computation time scales polynomially in system size.

- The ML model constructs the classifying function explicitly.
- Examples of classifying functions on few-body reduced density matrices (assuming const. spectral gap) include:
  1. Twist operators for 1D Haldane phase with $O(2)$-symmetry (linear function)
  2. Hall conductivity for systems adiabatically connected to free fermion (low-degree polynomial)
  3. Topological entanglement entropy in a constant region (nonlinear function)
- As long as the classifying function exists, the ML model with shadow kernel is guaranteed to find it.

2D topologically-ordered phases

We consider $T = 500$ randomized measurements to construct classical shadows for each state.
The classical unsupervised ML model is a kernel PCA using the shadow kernel.



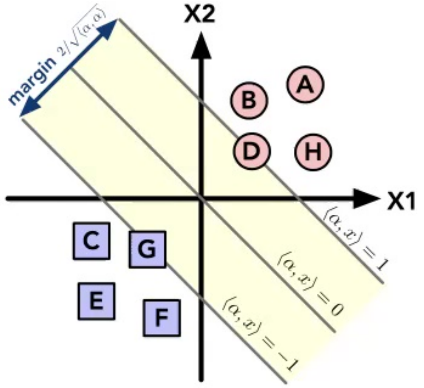
Classifying quantum phases: ML

- We need a more powerful ML model that can learn nonlinear functions, such as $\mathrm{Tr}(O\rho \otimes \rho)$, $\mathrm{Tr}(O\rho^{\otimes d})$, or a general analytic function $f(\rho)$.

- To do so, we consider learning a linear function in an ∞-dim space, where each state ρ is mapped to

$$\phi^{(\mathrm{shadow})}(S_T(\rho)) = \lim_{D,R\to\infty} \bigoplus_{d=0}^{D} \sqrt{\frac{\tau^d}{d!}} \left(\bigoplus_{r=0}^{R} \sqrt{\frac{1}{r!}\left(\frac{\gamma}{n}\right)^r} \bigoplus_{i_1=1}^{n} \cdots \bigoplus_{i_r=1}^{n} \mathrm{vec}\left[\frac{1}{T} \sum_{t=1}^{T} \bigotimes_{\ell=1}^{r} \sigma_{i_\ell}^{(t)} \right] \right)^{\otimes d}.$$

- It consists of arbitrarily-large r-body reduced density matrices and arbitrarily-high-degree expansion.

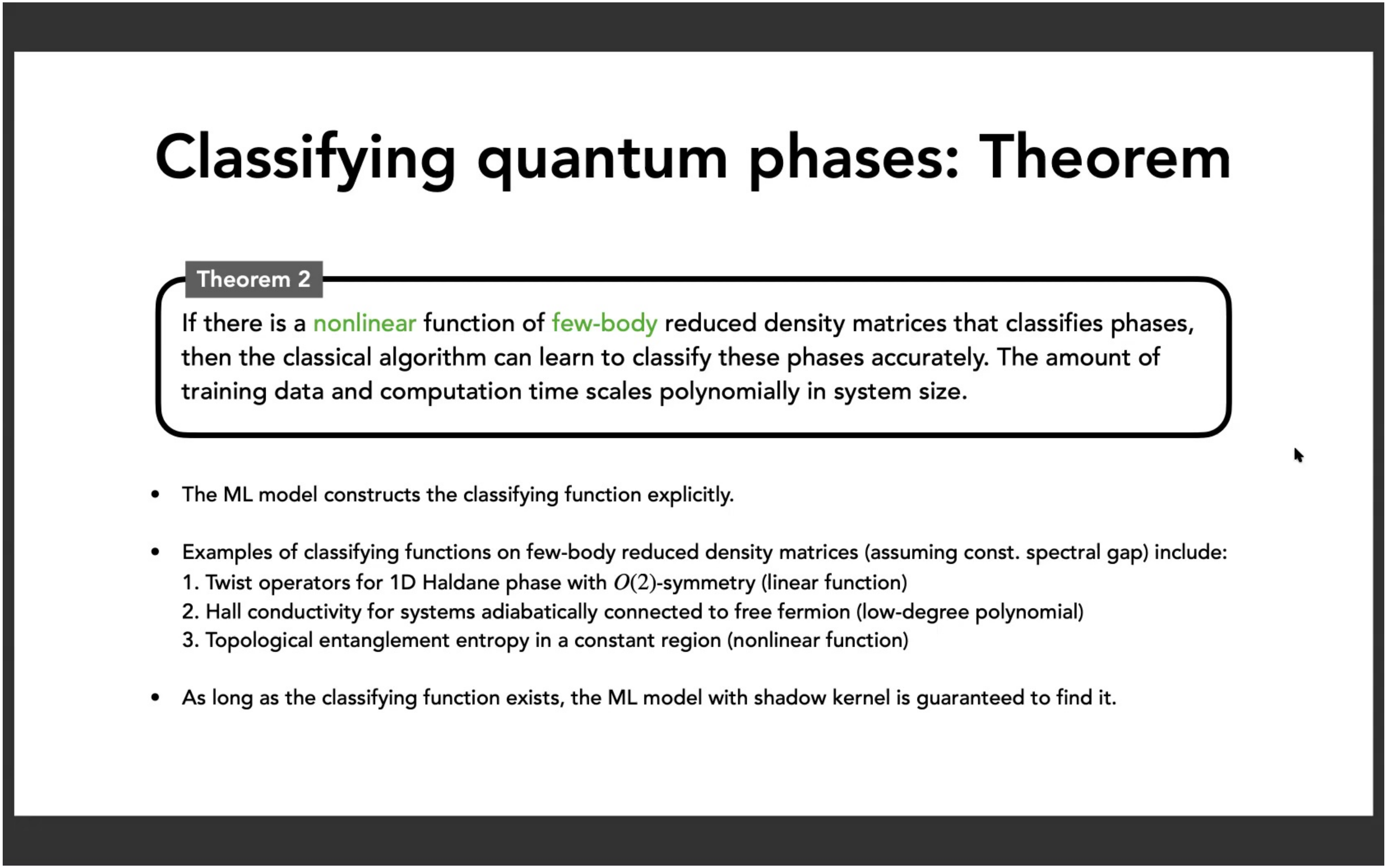

margin $2/\sqrt{\langle\alpha,\alpha\rangle}$
X2
X1
A
B
D
H
C
G
E
F
$\langle\alpha,x\rangle = 1$
$\langle\alpha,x\rangle = 0$
$\langle\alpha,x\rangle = -1$

Proposition 2

Consider two distinct topological phases A and B.

No (local/global) observable O exists such that

$\mathrm{Tr}(O\rho_A) > 0, \forall \rho_A \in$ phase A, $\quad \mathrm{Tr}(O\rho_B) \leq 0, \forall \rho_B \in$ phase B.



Classifying quantum phases: Theorem

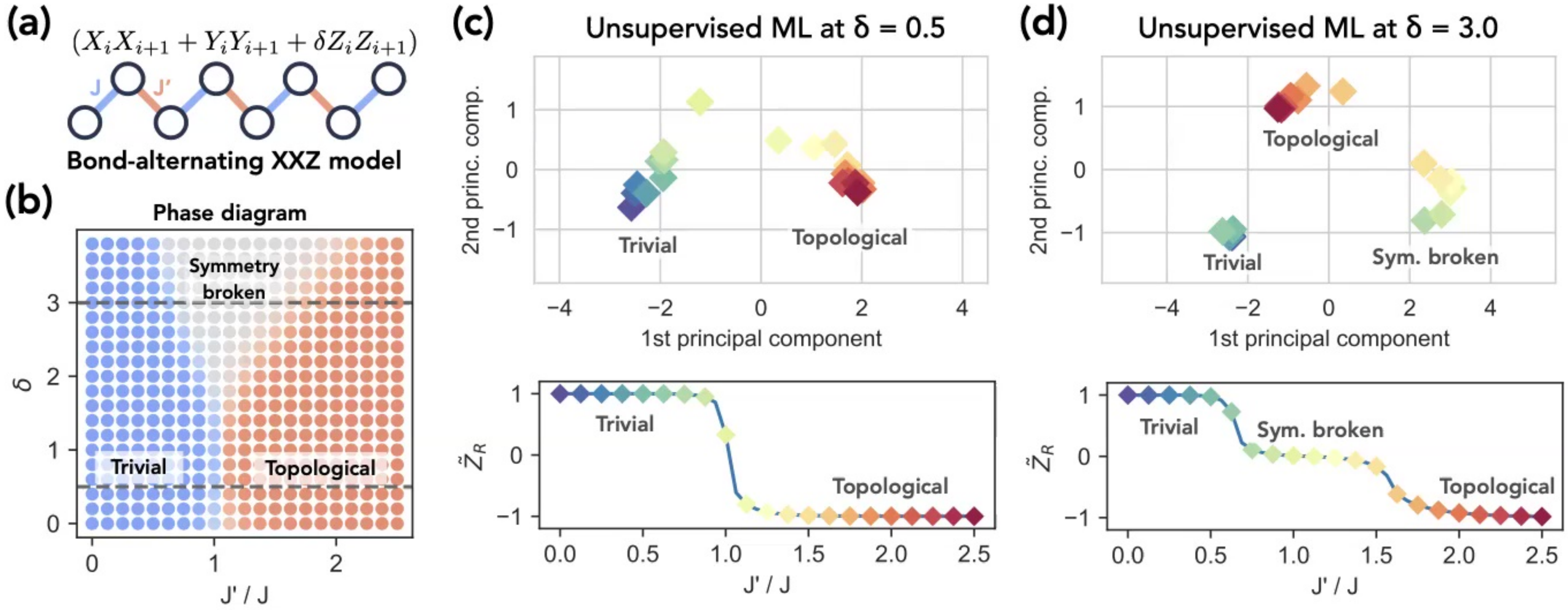
Theorem 2

If there is a nonlinear function of few-body reduced density matrices that classifies phases, then the classical algorithm can learn to classify these phases accurately. The amount of training data and computation time scales polynomially in system size.

- The ML model constructs the classifying function explicitly.
- Examples of classifying functions on few-body reduced density matrices (assuming const. spectral gap) include:
  1. Twist operators for 1D Haldane phase with $O(2)$-symmetry (linear function)
  2. Hall conductivity for systems adiabatically connected to free fermion (low-degree polynomial)
  3. Topological entanglement entropy in a constant region (nonlinear function)
- As long as the classifying function exists, the ML model with shadow kernel is guaranteed to find it.

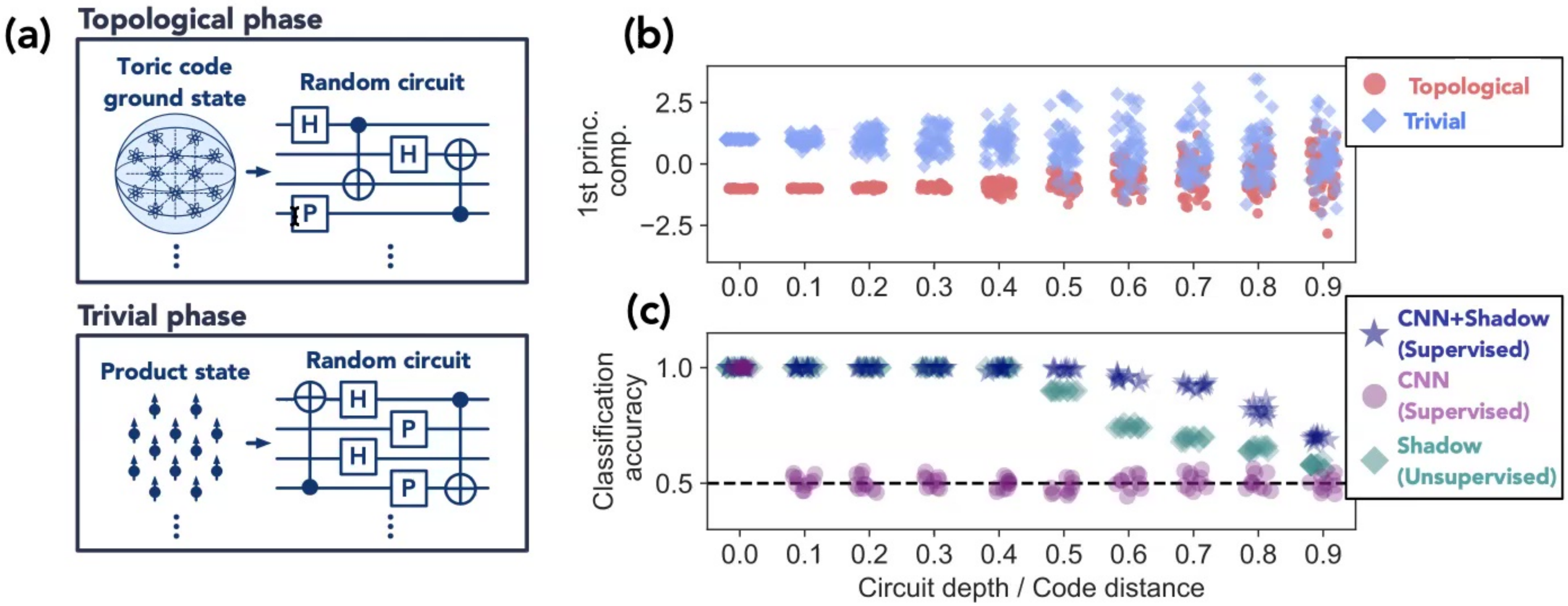
1D Symmetry protected topological phases

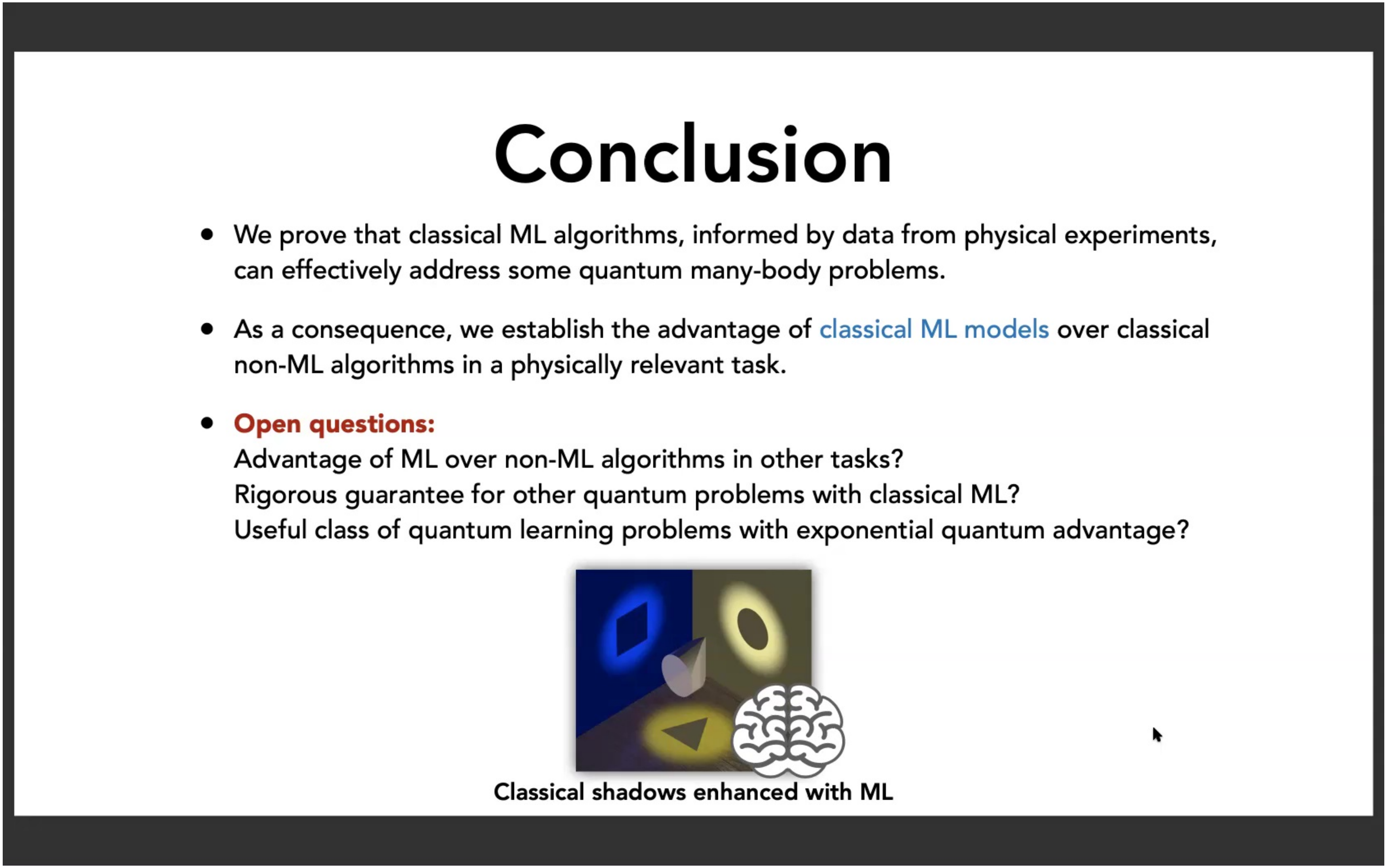
We consider $T = 500$ randomized measurements to construct classical shadows for each state.
The classical unsupervised ML model is a kernel PCA using the shadow kernel.



2D topologically-ordered phases

We consider $T = 500$ randomized measurements to construct classical shadows for each state.
The classical unsupervised ML model is a kernel PCA using the shadow kernel.





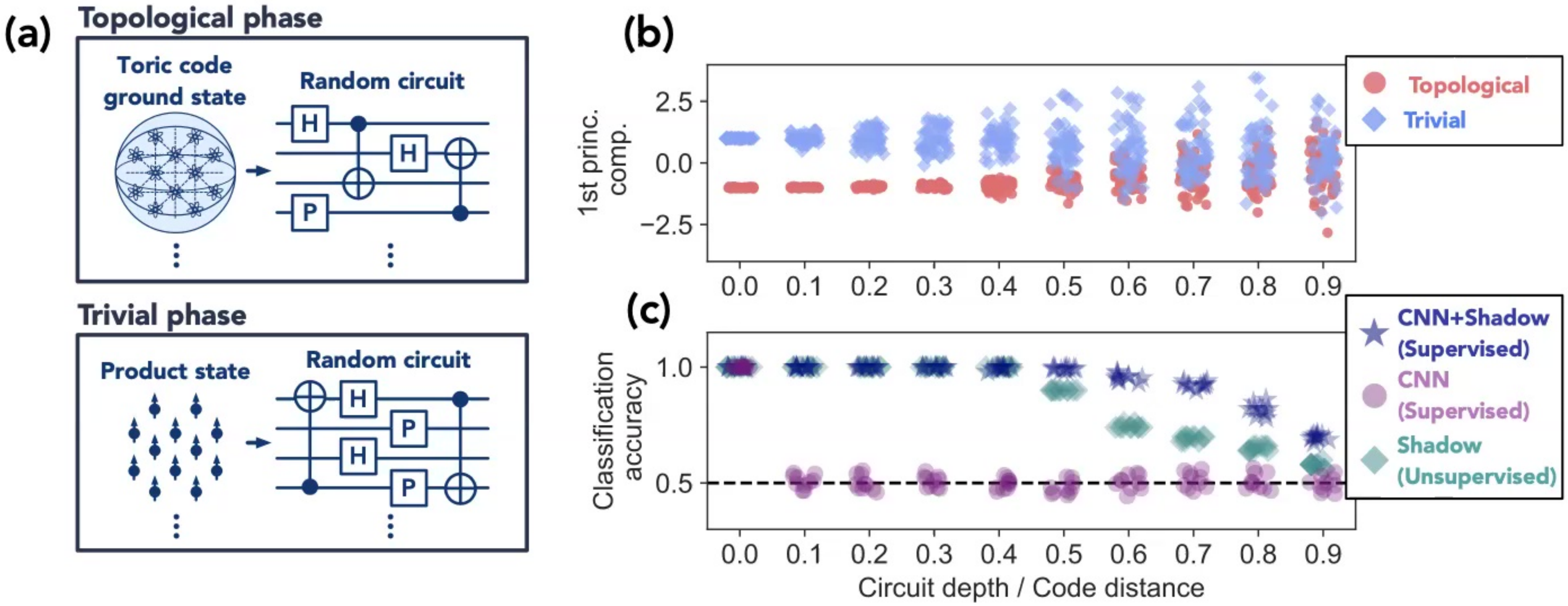
Conclusion

- We prove that classical ML algorithms, informed by data from physical experiments, can effectively address some quantum many-body problems.
- As a consequence, we establish the advantage of classical ML models over classical non-ML algorithms in a physically relevant task.
- **Open questions:**
  Advantage of ML over non-ML algorithms in other tasks?
  Rigorous guarantee for other quantum problems with classical ML?
  Useful class of quantum learning problems with exponential quantum advantage?

Classical shadows enhanced with ML

2D topologically-ordered phases

We consider $T = 500$ randomized measurements to construct classical shadows for each state.
The classical unsupervised ML model is a kernel PCA using the shadow kernel.


(a)
Topological phase
Toric code ground state
Random circuit
H
H
P
Trivial phase
Product state
Random circuit
H
H
P
P
(b)
1st princ. comp.
Topological
Trivial
(c)
Classification accuracy
Circuit depth / Code distance
CNN+Shadow (Supervised)
CNN (Supervised)
Shadow (Unsupervised)