Classical machine learning for quantum many-body problems

Laura Lewis

Caltech → Cambridge → University of Edinburgh

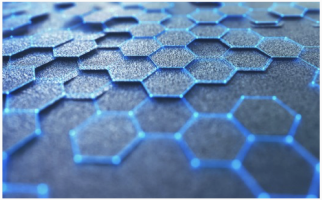
Motivation

A fundamental problem in physics is to learn how the world works.

Motivation

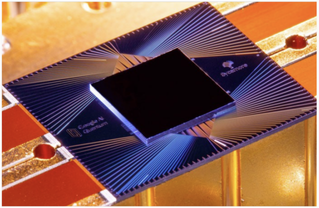
A fundamental problem in physics is to learn how the world works.

Chemistry



Materials Science

Physics



Quantum Devices

Sources: “Chemistry beakers and molecular structure” by Kwanchai Lerttanapunyaporn / EyeEm / Getty, https://www.nature.com/collections/ecjehiebic, https://www6.slac.stanford.edu/media/dwarf-galaxy-discover-largejpeg, https://www.nature.com/articles/d41586-019-03213-z

Motivation

Quantum computers are expected to help solve some important physics problems, but they are currently small and error-prone.

Motivation

Quantum computers are expected to help solve some important physics problems, but they are currently small and error-prone.

Meanwhile, how can we leverage our powerful classical computers?

Outline

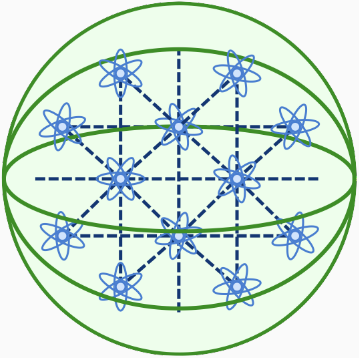
Outline

Classical Shadows

Classical ML for Ground States

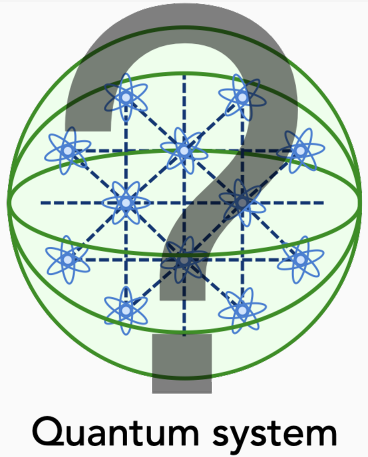
Proof Ideas

Quantum State Tomography

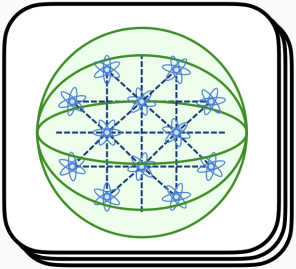


Quantum system

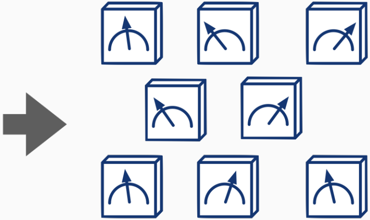
Quantum State Tomography



Quantum State Tomography

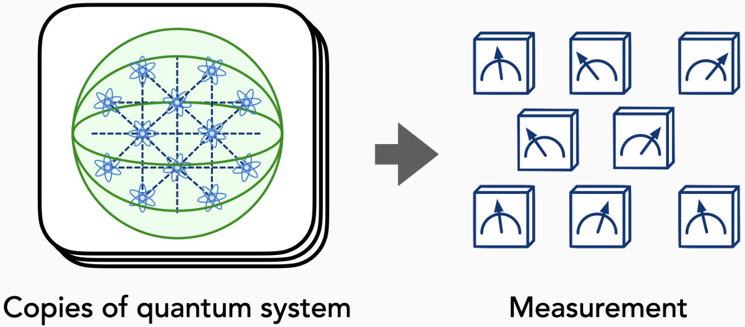


Copies of quantum system



Measurement

Quantum State Tomography



Given N samples of an n-qubit quantum state ρ, learn $\hat{\rho}$ such that

$$\mathrm{d}_{\mathrm{tr}}(\hat{\rho}, \rho) = \frac{1}{2}\|\hat{\rho} - \rho\|_1 < \epsilon.$$

Measures of Complexity

We want to minimize the sample complexity, i.e., the number N of copies of ρ.

Measures of Complexity

We want to minimize the sample complexity, i.e., the number N of copies of ρ.

We can also consider the computational complexity, i.e., the runtime of an algorithm.

Known Results

Quantum state tomography are known to require a sample complexity of $\Theta(4^n)$ in general[1].

[1] [Haah, Harrow, Ji, Wu, Yu, 2017], [O'Donnell and Wright, 2016]

Known Results

Quantum state tomography are known to require a sample complexity of $\Theta(4^n)$ in general[1].

This means that general tomography is practically infeasible.

[1] [Haah, Harrow, Ji, Wu, Yu, 2017], [O'Donnell and Wright, 2016]

Known Results

Quantum state tomography are known to require a sample complexity of $\Theta(4^n)$ in general[1].

This means that general tomography is practically infeasible.

Are there ways to circumvent this?

[1] [Haah, Harrow, Ji, Wu, Yu, 2017], [O'Donnell and Wright, 2016]

Known Results

Quantum state tomography are known to require a sample complexity of $\Theta(4^n)$ in general[1].

This means that general tomography is practically infeasible.

Are there ways to circumvent this?

Key Idea: What if we don't learn a full description of the state?

[1] [Haah, Harrow, Ji, Wu, Yu, 2017], [O'Donnell and Wright, 2016]

Known Results

Quantum state tomography are known to require a sample complexity of $\Theta(4^n)$ in general[1].

This means that general tomography is practically infeasible.

Are there ways to circumvent this?

Key Idea: What if we don't learn a full description of the state?

Can we just learn enough to be useful?

[1] [Haah, Harrow, Ji, Wu, Yu, 2017], [O'Donnell and Wright, 2016]

Known Results

Quantum state tomography are known to require a sample complexity of $\Theta(4^n)$ in general[1].

This means that general tomography is practically infeasible.

Are there ways to circumvent this?

Key Idea: What if we don't learn a full description of the state?

Can we just learn enough to be useful? i.e., to predict properties? [Aaronson, 2018]

[1] [Haah, Harrow, Ji, Wu, Yu, 2017], [O'Donnell and Wright, 2016]

Shadow Tomography

Let $E_1, \ldots, E_M$ be two-outcome measurements.

Shadow Tomography

Let $E_1, \ldots, E_M$ be two-outcome measurements. The goal is to estimate $\text{tr}(E_i \rho)$ up to ϵ-error.

Shadow Tomography

Let $E_1, \ldots, E_M$ be two-outcome measurements. The goal is to estimate $\mathrm{tr}(E_i \rho)$ up to ϵ-error.

Theorem ([Aaronson and Rothblum, STOC 2019])

One can perform shadow tomography with

$$N = \tilde{\mathcal{O}}\left(\frac{n^2 \log^2(M)}{\epsilon^8}\right)$$

copies of the unknown state ρ.

Shadow Tomography

Let $E_1, \ldots, E_M$ be two-outcome measurements. The goal is to estimate $\mathrm{tr}(E_i\rho)$ up to ϵ-error.

Theorem ([Aaronson and Rothblum, STOC 2019])

One can perform shadow tomography with

$$N = \tilde{\mathcal{O}}\left(\frac{n^2 \log^2(M)}{\epsilon^8}\right)$$

copies of the unknown state ρ.

Later improved by [Badescu and O'Donnell, STOC 2021] to

$$N = \tilde{\mathcal{O}}\left(\frac{n \log^2(M)}{\epsilon^4}\right).$$

Drawbacks

1. The protocol requires exponential-size quantum circuits acting on multiple copies of ρ at a time.

Drawbacks

1. The protocol requires exponential-size quantum circuits acting on multiple copies of ρ at a time.
2. Sample complexity scales with system size.

Drawbacks

1. The protocol requires exponential-size quantum circuits acting on multiple copies of ρ at a time.
2. Sample complexity scales with system size.
3. Observables have to be given beforehand.

Classical Shadows [HKP20]

Classical shadows addresses all of these drawbacks.

1. The protocol is very simple and requires only single copies of ρ at a time.

Classical Shadows [HKP20]

Classical shadows addresses all of these drawbacks.

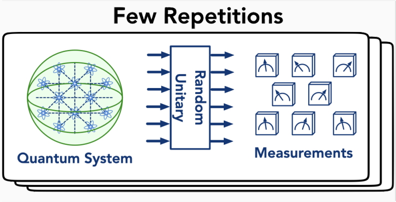
1. The protocol is very simple and requires only single copies of ρ at a time.
2. Sample complexity is independent of system size for broad classes of observables.

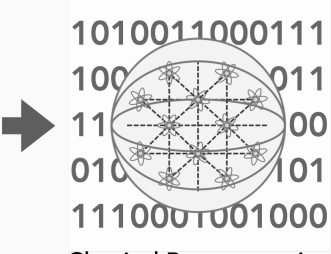
Classical Shadows [HKP20]

Classical shadows addresses all of these drawbacks.

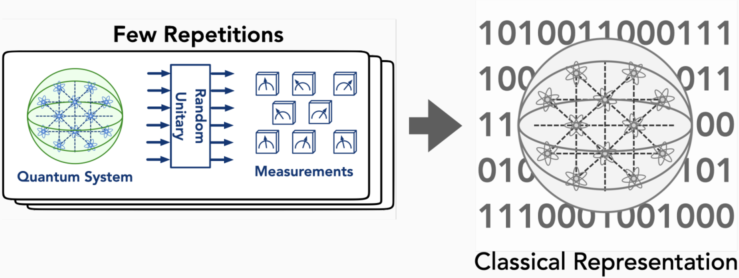
1. The protocol is very simple and requires only single copies of ρ at a time.
2. Sample complexity is independent of system size for broad classes of observables.
3. One can prepare a classical representation of the unknown quantum state, from which properties can be predicted.

Classical Shadows Algorithm [HKP20]



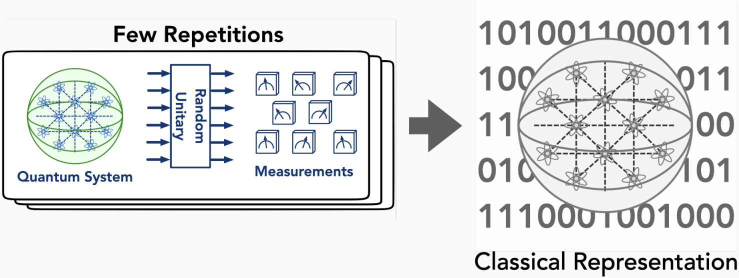


Classical Shadows Algorithm [HKP20]



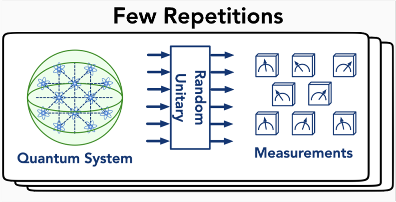
1. Apply a random Clifford U_i to ρ.

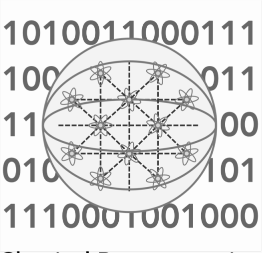
Classical Shadows Algorithm [HKP20]



1. Apply a random Clifford U_i to ρ.
2. Measure in the computational basis to obtain $b_i \in \{0, 1\}^n$.

Classical Shadows Algorithm [HKP20]



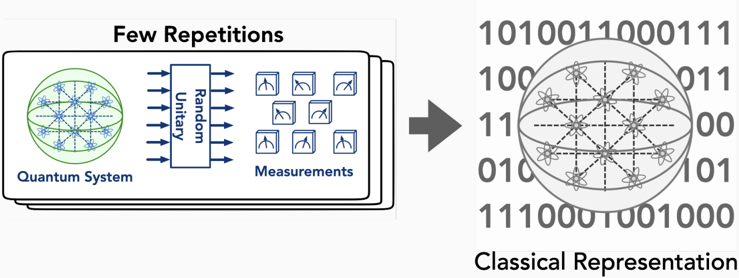


Classical Representation

1. Apply a random Clifford U_i to ρ.
2. Measure in the computational basis to obtain $b_i \in \{0,1\}^n$.
3. This gives a classical snapshot:

$$\hat{\rho}_i = (2^n + 1) U_i^\dagger |b_i\rangle\langle b_i| U_i - I.$$

Classical Shadows Algorithm [HKP20]



1. Apply a random Clifford U_i to ρ.
2. Measure in the computational basis to obtain $b_i \in \{0,1\}^n$.
3. This gives a classical snapshot:

$$\hat{\rho}_i = (2^n + 1) U_i^\dagger |b_i\rangle\langle b_i| U_i - I.$$

4. Repeating this, we obtain the classical shadow of ρ

$$S_N(\rho) = \{\hat{\rho}_1, \ldots, \hat{\rho}_N\}.$$

Classical Shadows Algorithm [HKP20]

Given the classical shadow

$$\mathsf{S}_N(\rho) = \{\hat{\rho}_1, \ldots, \hat{\rho}_N\},$$

one can predict expectation values via median-of-means.

Classical Shadows Algorithm [HKP20]

Given the classical shadow

$$\mathsf{S}_N(\rho) = \{\hat{\rho}_1, \ldots, \hat{\rho}_N\},$$

one can predict expectation values via median-of-means.

1. Compute $Z_i \triangleq \mathrm{tr}(O\hat{\rho}_i)$ for all $i = 1, \ldots, N$.

Classical Shadows Algorithm [HKP20]

Given the classical shadow

$$\mathsf{S}_N(\rho) = \{\hat{\rho}_1, \ldots, \hat{\rho}_N\},$$

one can predict expectation values via median-of-means.

1. Compute $Z_i \triangleq \mathrm{tr}(O\hat{\rho}_i)$ for all $i = 1, \ldots, N$.
2. Predict

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

Classical Shadows Guarantee

Theorem ([Huang, Kueng, Preskill, Nat. Phys. 2020])

Let $O_1, \ldots, O_M$ be observables with $\mathrm{tr}\left(O_i^2\right) \leq B$ for all i. Then, we can estimate expectation values up to ϵ-error using

$$N = \mathcal{O}\left(\frac{B \log M}{\epsilon^2}\right).$$

Classical Shadows Guarantee

Theorem ([Huang, Kueng, Preskill, Nat. Phys. 2020])

Let $O_1, \ldots, O_M$ be observables with $\operatorname{tr}\left(O_i^2\right) \leq B$ for all i. Then, we can estimate expectation values up to ϵ-error using

$$N = \mathcal{O}\left(\frac{B \log M}{\epsilon^2}\right).$$

In general, the type of random measurement affects the algorithm, class of observables, and sample complexity.

Classical Shadows Guarantee

Theorem ([Huang, Kueng, Preskill, Nat. Phys. 2020])

Let $O_1, \ldots, O_M$ be observables with $\mathrm{tr}\left(O_i^2\right) \leq B$ for all i. Then, we can estimate expectation values up to ϵ-error using

$$N = \mathcal{O}\left(\frac{B \log M}{\epsilon^2}\right).$$

In general, the type of random measurement affects the algorithm, class of observables, and sample complexity.

Another important example is to predicting local observables.

Outline

Classical Shadows

Classical ML for Ground States

Proof Ideas

Motivation

Can we design classical ML algorithms to solve difficult quantum physics problems using classical shadows?

Motivation

Can we design classical ML algorithms to solve difficult quantum physics problems using classical shadows?

In particular, we focus on finding ground states.

Motivation

Can we design classical ML algorithms to solve difficult quantum physics problems using classical shadows?

In particular, we focus on finding ground states.

Can classical ML efficiently predict ground states after learning from training data?

(Classical) Machine Learning

Goal: Want to learn some unknown function $c : \mathcal{X} \to \mathcal{Y}$.

(Classical) Machine Learning

Goal: Want to learn some unknown function $c : \mathcal{X} \to \mathcal{Y}$.

Given training data $\{(x_\ell, y_\ell)\}_{\ell=1}^N$, where $x_\ell \sim \mathcal{D}(\mathcal{X})$ are sampled from some distribution over $\mathcal{X}$

(Classical) Machine Learning

Goal: Want to learn some unknown function $c : \mathcal{X} \to \mathcal{Y}$.

Given training data $\{(x_\ell, y_\ell)\}_{\ell=1}^{N}$, where $x_\ell \sim \mathcal{D}(\mathcal{X})$ are sampled from some distribution over $\mathcal{X}$ and $y_\ell = c(x_\ell)$.

(Classical) Machine Learning

Goal: Want to learn some unknown function $c : \mathcal{X} \to \mathcal{Y}$.

Given training data $\{(x_\ell, y_\ell)\}_{\ell=1}^N$, where $x_\ell \sim \mathcal{D}(\mathcal{X})$ are sampled from some distribution over $\mathcal{X}$ and $y_\ell = c(x_\ell)$.

We want to learn a function h^* that has low average prediction error

$$\mathop{\mathbb{E}}_{x \sim \mathcal{D}(\mathcal{X})} |h^*(x) - c(x)|^2 \leq \epsilon$$

(Classical) Machine Learning

Goal: Want to learn some unknown function $c : \mathcal{X} \to \mathcal{Y}$.

Given training data $\{(x_\ell, y_\ell)\}_{\ell=1}^{N}$, where $x_\ell \sim \mathcal{D}(\mathcal{X})$ are sampled from some distribution over $\mathcal{X}$ and $y_\ell = c(x_\ell)$.

We want to learn a function h^* that has low average prediction error

$$\mathop{\mathbb{E}}_{x \sim \mathcal{D}(\mathcal{X})} |h^*(x) - c(x)|^2 \leq \epsilon$$

with as little training data as possible.

(Classical) Machine Learning

Goal: Want to learn some unknown function $c : \mathcal{X} \to \mathcal{Y}$.

Given training data $\{(x_\ell, y_\ell)\}_{\ell=1}^{N}$, where $x_\ell \sim \mathcal{D}(\mathcal{X})$ are sampled from some distribution over $\mathcal{X}$ and $y_\ell = c(x_\ell)$.

We want to learn a function h^* that has low average prediction error

$$\mathop{\mathbb{E}}_{x \sim \mathcal{D}(\mathcal{X})} |h^*(x) - c(x)|^2 \leq \epsilon$$

with as little training data as possible.

The amount of training data N is called the sample complexity.

Learning Ground State Properties

Let H be an n-qubit geometrically-local gapped Hamiltonian that can be written as a sum of few-body interaction terms

$$H = \sum_j h_j.$$

Learning Ground State Properties

Let H be an n-qubit geometrically-local gapped Hamiltonian that can be written as a sum of few-body interaction terms

$$H = \sum_j h_j.$$

Parameterize H by a vector $x \in [-1, 1]^m$

Learning Ground State Properties

Let H be an n-qubit geometrically-local gapped Hamiltonian that can be written as a sum of few-body interaction terms

$$H = \sum_j h_j.$$

Parameterize H by a vector $x \in [-1, 1]^m$

$$H(x) = \sum_j h_j(x).$$

Learning Ground State Properties

Let H be an n-qubit geometrically-local gapped Hamiltonian that can be written as a sum of few-body interaction terms

$$H = \sum_j h_j.$$

Parameterize H by a vector $x \in [-1, 1]^m$

$$H(x) = \sum_j h_j(x).$$

e.g., $H(x) = x_1 Z_1 + x_2 Z_2$.

Learning Ground State Properties

Let H be an n-qubit geometrically-local gapped Hamiltonian that can be written as a sum of few-body interaction terms

$$H(x) = \sum_j h_j(x)$$

parameterized smoothly by $x \in [-1, 1]^m$.

Learning Ground State Properties

Let H be an n-qubit geometrically-local gapped Hamiltonian that can be written as a sum of few-body interaction terms

$$H(x) = \sum_j h_j(x)$$

parameterized smoothly by $x \in [-1, 1]^m$.

Let $\rho(x)$ be the ground state of $H(x)$.

Learning Ground State Properties

Let H be an n-qubit geometrically-local gapped Hamiltonian that can be written as a sum of few-body interaction terms

$$H(x) = \sum_j h_j(x)$$

parameterized smoothly by $x \in [-1, 1]^m$.

Let $\rho(x)$ be the ground state of $H(x)$.

Goal: Want to predict ground state properties $\mathrm{tr}(O\rho(x))$, where O is a sum of geometrically local observables with $\|O\|_\infty \leq 1$.

Learning Ground State Properties

Let H be an n-qubit geometrically-local gapped Hamiltonian that can be written as a sum of few-body interaction terms

$$H(x) = \sum_j h_j(x)$$

parameterized smoothly by $x \in [-1, 1]^m$.

Let $\rho(x)$ be the ground state of $H(x)$.

Goal: Want to predict ground state properties $\mathrm{tr}(O\rho(x))$, where O is a sum of geometrically local observables with $\|O\|_\infty \leq 1$.

Given training data $\{(x_\ell, y_\ell)\}_{\ell=1}^N$, where $x_\ell \sim \mathcal{D}([-1, 1]^m)$ and $y_\ell \approx \mathrm{tr}(O\rho(x))$.

Learning Ground State Properties

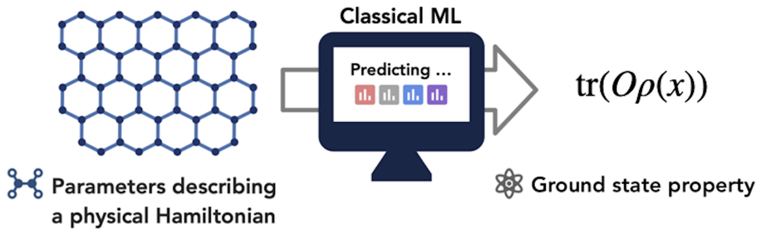
Goal: Want to predict ground state properties $\mathrm{tr}(O\rho(x))$, where O is a sum of geometrically local observables with $\|O\|_\infty \leq 1$.

Given training data $\{(x_\ell, y_\ell)\}_{\ell=1}^N$, where $x_\ell \sim \mathcal{D}([-1,1]^m)$ and $y_\ell \approx \mathrm{tr}(O\rho(x))$.

Learning Ground State Properties

Goal: Want to predict ground state properties $\text{tr}(O\rho(x))$, where O is a sum of geometrically local observables with $\|O\|_\infty \leq 1$.

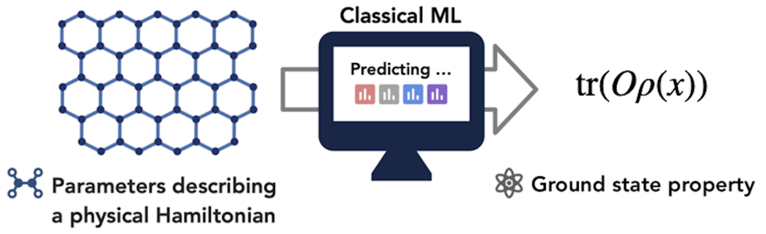
Given training data $\{(x_\ell, y_\ell)\}_{\ell=1}^N$, where $x_\ell \sim \mathcal{D}([-1,1]^m)$ and $y_\ell \approx \text{tr}(O\rho(x))$.



Learning Ground State Properties

Goal: Want to predict ground state properties $\mathrm{tr}(O\rho(x))$, where O is a sum of geometrically local observables with $\|O\|_\infty \leq 1$.

Given training data $\{(x_\ell, y_\ell)\}_{\ell=1}^N$, where $x_\ell \sim \mathcal{D}([-1,1]^m)$ and $y_\ell \approx \mathrm{tr}(O\rho(x))$.

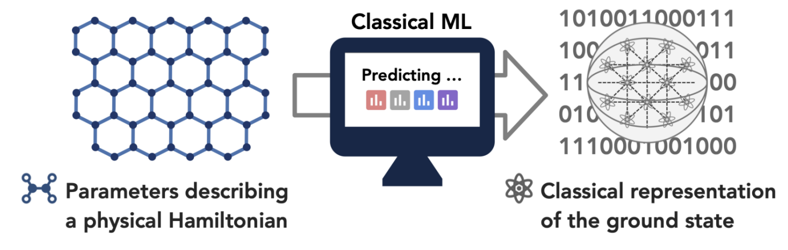


Can classical ML do this efficiently?

Learning Ground State Properties

Goal: Want to predict ground state representations.

Given training data $\{(x_\ell, y_\ell)\}_{\ell=1}^N$, where $x_\ell \sim \mathcal{D}([-1,1]^m)$ and y_ℓ approximates the ground state.



Can classical ML do this efficiently?

Rigorous Guarantees

Theorem ([Huang et al. Science 2022][2])

There exists an efficient classical ML model $g^*(x)$ that achieves

$$\mathbb{E}_{x \sim U([-1,1]^m)} \left| g^*(x) - \mathrm{tr}(O\rho(x)) \right|^2 \leq \epsilon.$$

using $N = n^{\mathcal{O}(1/\epsilon)}$ training data.

[2] [Huang, Kueng, Torlai, Albert, Preskill, Science 2022]

Rigorous Guarantees

Theorem ([Huang et al. Science 2022][2])

There exists an efficient classical ML model $g^*(x)$ that achieves

$$\mathop{\mathbb{E}}_{x\sim U([-1,1]^m)} \left|g^*(x) - \mathrm{tr}(O\rho(x))\right|^2 \leq \epsilon.$$

using $N = n^{\mathcal{O}(1/\epsilon)}$ training data.

Proposition (Computational Hardness)

Assuming RP $\neq$ NP, then for 2D Hamiltonians, no randomized classical algorithm predicting 1-body observables can achieve an average prediction error $\leq 1/4$ within $\mathrm{poly}(n, m)$ time.

[2][Huang, Kueng, Torlai, Albert, Preskill, Science 2022]

Rigorous Guarantees

Key Additional Assumption:

$$H(x) = \sum_j h_j(\vec{x}_j)$$

[3] [Lewis, Huang, Tran, Lehner, Kueng, Preskill, Nat. Commun. 2024]

Rigorous Guarantees

Key Additional Assumption:

$$H(x) = \sum_j h_j(\vec{x}_j)$$

where each h_j depends on a constant number of parameters $\vec{x}_j$.

[3] [Lewis, Huang, Tran, Lehner, Kueng, Preskill, Nat. Commun. 2024]

Rigorous Guarantees

Key Additional Assumption:

$$H(x) = \sum_j h_j(\vec{x}_j)$$

where each h_j depends on a constant number of parameters $\vec{x}_j$.

Theorem ([Lewis et al. Nat. Commun. 2024][3])

There exists a classical ML model $h^*(x)$ that achieves

$$\mathop{\mathbb{E}}_{x \sim \mathcal{D}([-1,1]^m)} |h^*(x) - \mathrm{tr}(O\rho(x))|^2 \leq \epsilon$$

using $N = \log(n) 2^{\mathrm{polylog}(1/\epsilon)}$ training data sampled from an arbitrary distribution.

[3][Lewis, Huang, Tran, Lehner, Kueng, Preskill, Nat. Commun. 2024]

Related Works

- [Wanner, Lewis, Bhattacharyya, Dubhashi, Gheorghiu, NeurIPS 2024]: no system size dependence + neural network guarantees

Related Works

- [Wanner, Lewis, Bhattacharyya, Dubhashi, Gheorghiu, NeurIPS 2024]: no system size dependence + neural network guarantees

- [Onorati, Rouzé, Stilck França, Watson, Nat. Commun. 2024]: similar guarantee for learning Gibbs states with exponentially decaying correlations

Related Works

- [Wanner, Lewis, Bhattacharyya, Dubhashi, Gheorghiu, NeurIPS 2024]: no system size dependence + neural network guarantees
- [Onorati, Rouzé, Stilck França, Watson, Nat. Commun. 2024]: similar guarantee for learning Gibbs states with exponentially decaying correlations
- [Onorati, Rouzé, Stilck França, Watson, arXiv:2311.07506 2024]: similar guarantee for learning Lindbladian phases of matter with local rapid mixing

Outline

Classical Shadows

Classical ML for Ground States

Proof Ideas

Key Idea: Simple Form for Ground State Property

Theorem ([LHT+24])

The ground state property we wish to predict can be approximated by a sum of smooth local functions:

$$\mathrm{tr}(O\rho(x)) \approx f(x).$$

Key Idea: Simple Form for Ground State Property

Theorem ([LHT+24])

The ground state property we wish to predict can be approximated by a sum of smooth local functions:

$$\mathrm{tr}(O\rho(x)) \approx f(x).$$

Write O in the Pauli basis: $O = \sum_{P \in \{I,X,Y,Z\}^{\otimes n}} \alpha_P P$.

Key Idea: Simple Form for Ground State Property

Theorem ([LHT+24])

The ground state property we wish to predict can be approximated by a sum of smooth local functions:

$$\mathrm{tr}(O\rho(x)) \approx f(x).$$

Write O in the Pauli basis: $O = \sum_{P \in \{I,X,Y,Z\}^{\otimes n}} \alpha_P P$.

It suffices to show that

$$\mathrm{tr}(P\rho(x)) \approx f_P(x),$$

for f_P a local function.

Key Idea: Simple Form for Ground State Property

Write O in the Pauli basis: $O = \sum_{P \in \{I,X,Y,Z\}^{\otimes n}} \alpha_P P$.

Key Idea: Simple Form for Ground State Property

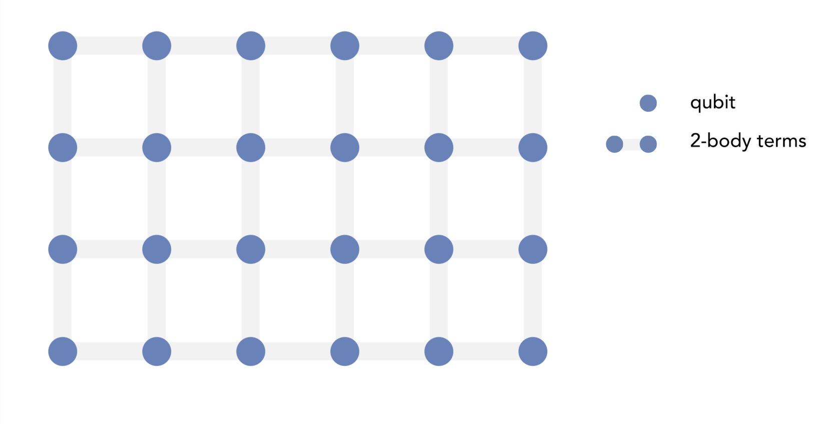
Write O in the Pauli basis: $O = \sum_{P \in \{I,X,Y,Z\}^{\otimes n}} \alpha_P P$.

For each Pauli P, we define a map $\chi_P(x)$ that sets parameters in x that are "far from P" set to 0.

Key Idea: Simple Form for Ground State Property

Write O in the Pauli basis: $O = \sum_{P \in \{I,X,Y,Z\}^{\otimes n}} \alpha_P P$.

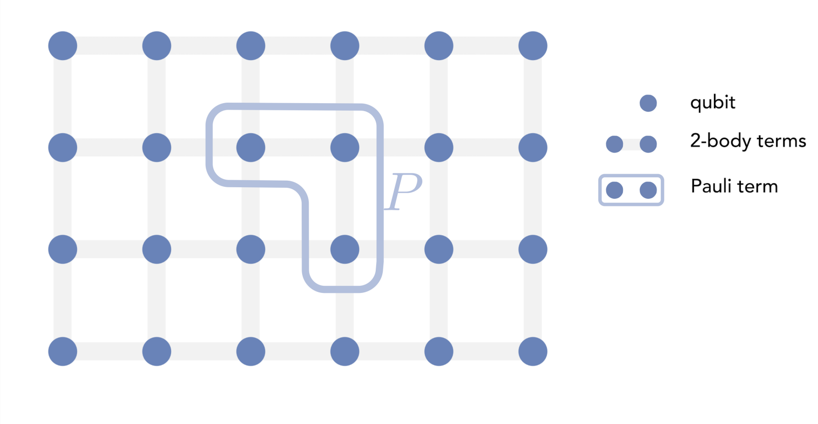
For each Pauli P, we define a map $\chi_P(x)$ that sets parameters in x that are "far from P" set to 0.



Key Idea: Simple Form for Ground State Property

Write O in the Pauli basis: $O = \sum_{P \in \{I,X,Y,Z\}^{\otimes n}} \alpha_P P$.

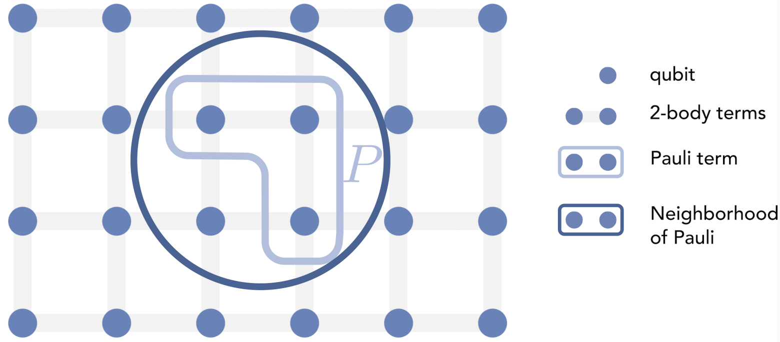
For each Pauli P, we define a map $\chi_P(x)$ that sets parameters in x that are "far from P" set to 0.



Key Idea: Simple Form for Ground State Property

Write O in the Pauli basis: $O = \sum_{P \in \{I,X,Y,Z\}^{\otimes n}} \alpha_P P$.

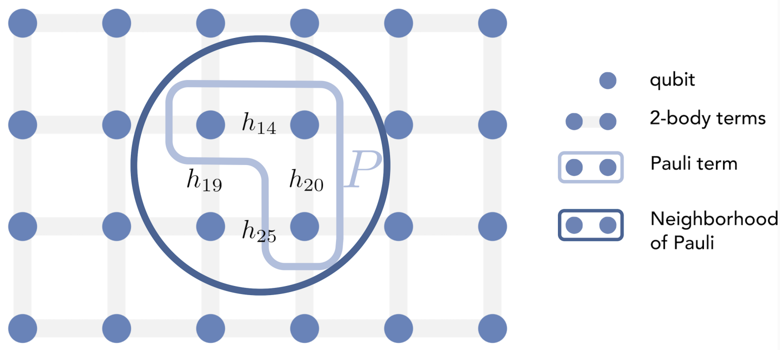
For each Pauli P, we define a map $\chi_P(x)$ that sets parameters in x that are "far from P" set to 0.



Key Idea: Simple Form for Ground State Property

Write O in the Pauli basis: $O = \sum_{P \in \{I,X,Y,Z\}^{\otimes n}} \alpha_P P$.

For each Pauli P, we define a map $\chi_P(x)$ that sets parameters in x that are "far from P" set to 0.



χ_P depends only on $\vec{x}_{14}, \vec{x}_{19}, \vec{x}_{20}, \vec{x}_{25}$

Key Idea: Simple Form for Ground State Property

Lemma

$$\mathrm{tr}(P\rho(x)) \approx f_P(x) = \mathrm{tr}(P\rho(\chi_P(x)))$$

Key Idea: Simple Form for Ground State Property

Lemma

$$\mathrm{tr}(P\rho(x)) \approx f_P(x) = \mathrm{tr}(P\rho(\chi_P(x)))$$

In other words, only parameters "close to P" affect the ground state property.

Algorithm of [LHT+24]

So we get

$$\mathrm{tr}(O\rho(x)) \approx f(x) = \sum_P \alpha_P f_P(x).$$

Algorithm of [LHT+24]

So we get

$$\mathrm{tr}(O\rho(x)) \approx f(x) = \sum_{P} \alpha_P f_P(x).$$

In fact, [LHT+24] shows that

$$\mathrm{tr}(O\rho(x)) \approx \boldsymbol{w}' \cdot \phi(x).$$

Algorithm of [LHT+24]

So we get

$$\mathrm{tr}(O\rho(x)) \approx f(x) = \sum_P \alpha_P f_P(x).$$

In fact, [LHT+24] shows that

$$\mathrm{tr}(O\rho(x)) \approx \boldsymbol{w}' \cdot \phi(x).$$

Algorithm:

1. Apply feature mapping ϕ.

Algorithm of [LHT+24]

So we get

$$\mathrm{tr}(O\rho(x)) \approx f(x) = \sum_P \alpha_P f_P(x).$$

In fact, [LHT+24] shows that

$$\mathrm{tr}(O\rho(x)) \approx \boldsymbol{w}' \cdot \phi(x).$$

Algorithm:

1. Apply feature mapping ϕ.
2. Learn $h^*(x) = \boldsymbol{w}^* \cdot \phi(x)$ using ℓ_1-regularized regression (LASSO) over the feature space.

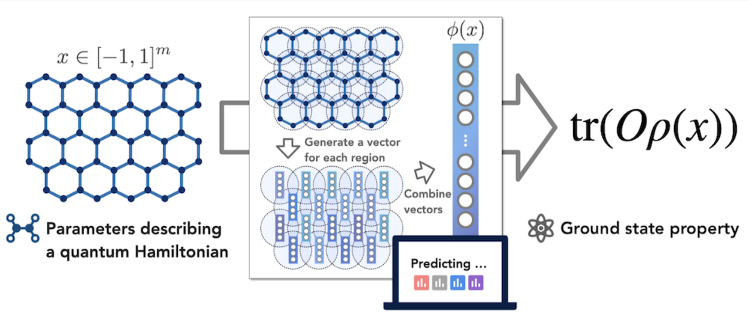
Algorithm of [LHT+24]

So we get

$$\mathrm{tr}(O\rho(x)) \approx f(x) = \sum_P f_P(x).$$

In fact, [LHT+24] shows that

$$\mathrm{tr}(O\rho(x)) \approx \boldsymbol{w}' \cdot \phi(x).$$



Numerical Experiments [LHT+24]

Spins placed on a 2D lattice with Hamiltonian

$$H = \sum_{\langle ij \rangle} J_{ij}(X_i X_j + Y_i Y_j + Z_i Z_j).$$

Numerical Experiments [LHT+24]

Spins placed on a 2D lattice with Hamiltonian

$$H = \sum_{\langle ij \rangle} J_{ij}(X_i X_j + Y_i Y_j + Z_i Z_j).$$

The parameter vector is $x = \{J_{ij}\}$.

Numerical Experiments [LHT+24]

Spins placed on a 2D lattice with Hamiltonian

$$H = \sum_{\langle ij \rangle} J_{ij}(X_i X_j + Y_i Y_j + Z_i Z_j).$$

The parameter vector is $x = \{J_{ij}\}$.

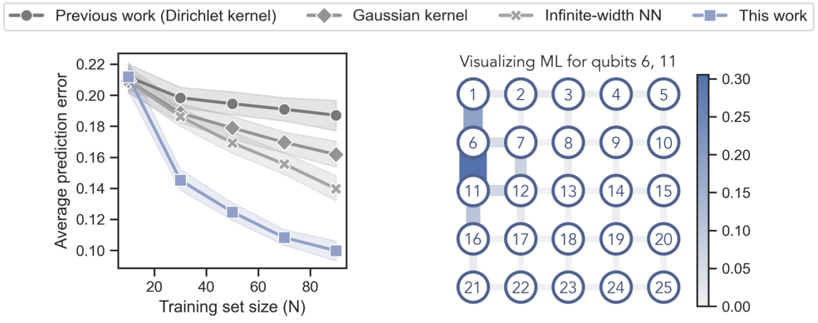
We want to predict two-body correlation functions, i.e., the expectation value of

$$C_{ij} = \frac{1}{3}(X_i X_j + Y_i Y_j + Z_i Z_j).$$

Numerical Experiments [LHT+24]

Spins placed on a 2D lattice with Hamiltonian

$$H = \sum_{\langle ij \rangle} J_{ij}(X_i X_j + Y_i Y_j + Z_i Z_j).$$



Summary

We've discussed how classical ML algorithms can predict ground state properties using very little data.

Summary

We've discussed how classical ML algorithms can predict ground state properties using very little data.

This raises the hope that ML algorithms can address practical problems by learning from the small amount of data available from physical experiments.

Open Questions

- Can quantum ML algorithms predict ground state properties even better?

Open Questions

- Can quantum ML algorithms predict ground state properties even better?
- Can ML learn to predict other physical properties (e.g., low-energy excited state properties)?