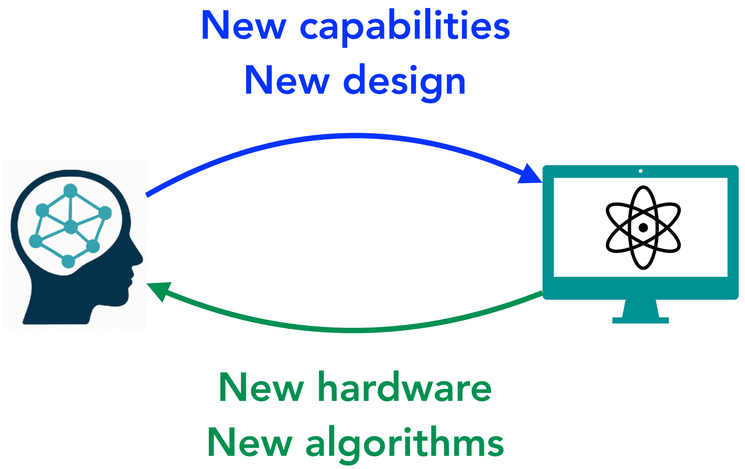
When Quantum Meets AI: Opportunities & Challenges

박경덕

응용통계학과 · 양자정보학과 · 양자정보기술연구원

Summary

- Quantum for AI: New tools for prediction, representation and generation
- AI for Quantum: New kinds of data and new challenges to AI
- Mathematical tools can help analyze and improve quantum computing and AI
- When Quantum meets AI, new algorithms and new mathematical questions emerge



Quantum Computing

- A quantum computer evolves according to Schrödinger's equation.
- Computation is programmed by controlling the Hamiltonian

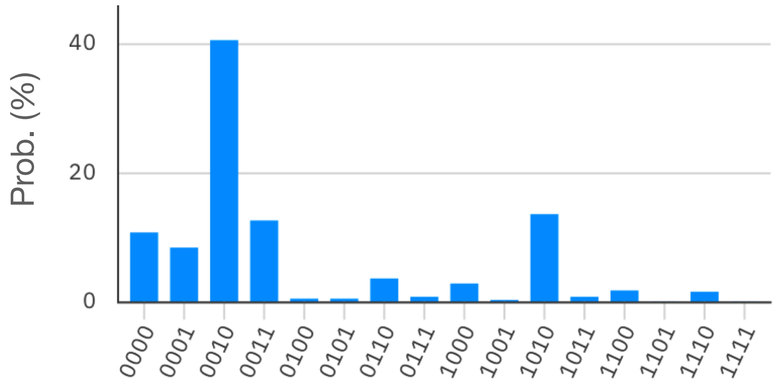
$$i\hbar\frac{d\psi(t)}{dt} = H(\theta(t))\psi(t)$$

$$\overbrace{\begin{bmatrix} h_{11} & \cdots & h_{12^n} \\ \vdots & \ddots & \vdots \\ h_{2^n1} & \cdots & h_{2^n2^n} \end{bmatrix}}^{2^n} \begin{bmatrix} a_1 \\ \vdots \\ a_{2^n} \end{bmatrix} \Bigg\} 2^n$$

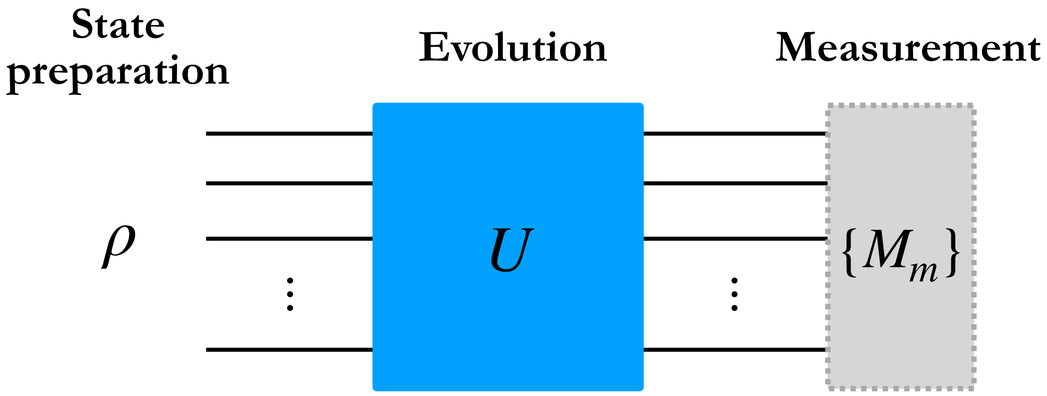
The state space increases exponentially w.r.t the number of qubits!
→ Hard to compute classically in general

- Encode the problem in $H(\theta(t))$ and/or $\psi(0)$
- Control $\theta(t)$ from $t = 0$ to $t = T$
- Measure $\psi(T)$

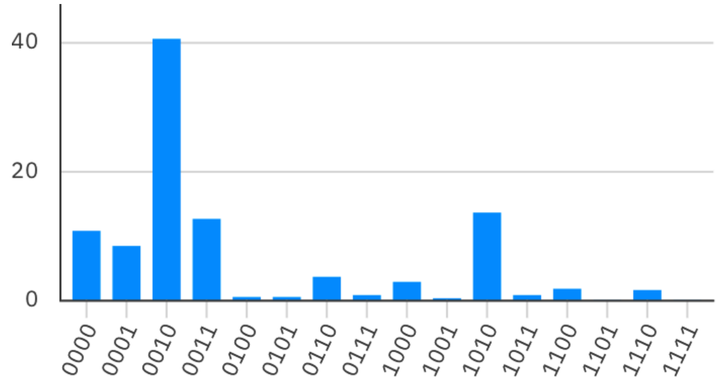
Answer is encoded in the probability distribution
e.g. mode, expectation, etc.



Quantum Circuit Notation



$$\rho, U, M_m \in \mathbb{C}^{2^n \times 2^n}$$



$$\Pr(m) = \mathrm{Tr}\left[M_m^\dagger M_m U \rho U^\dagger\right]$$

$$\rho \geq \mathbf{0},$$
$$\mathrm{Tr}(\rho) = 1,$$
$$\mathrm{Tr}(\rho^2) \leq 1$$

$$U^\dagger = U^{-1}$$

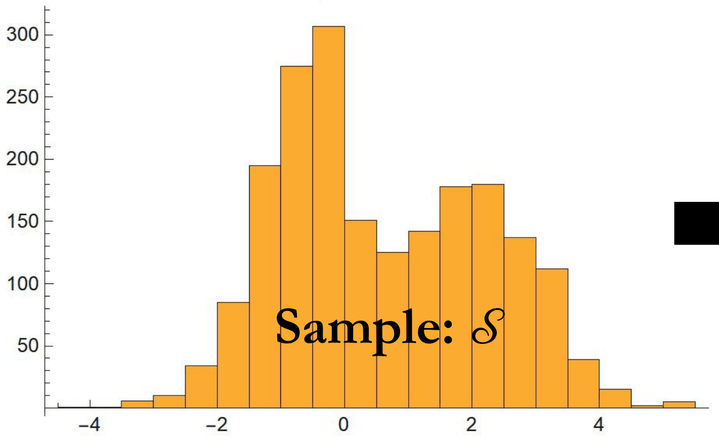
$$\sum_m M_m^\dagger M_m = I$$

A quantum circuit can be used to evaluate quantities of the form $\mathrm{Tr}\left[A^\dagger B\right]$ where $A^\dagger = A$, $B^\dagger = B$

→ This is the Hilbert-Schmidt inner product $\langle A, B \rangle_{\mathrm{HS}}$

Introduction: ML

- Machines learn by solving as an optimization problem:



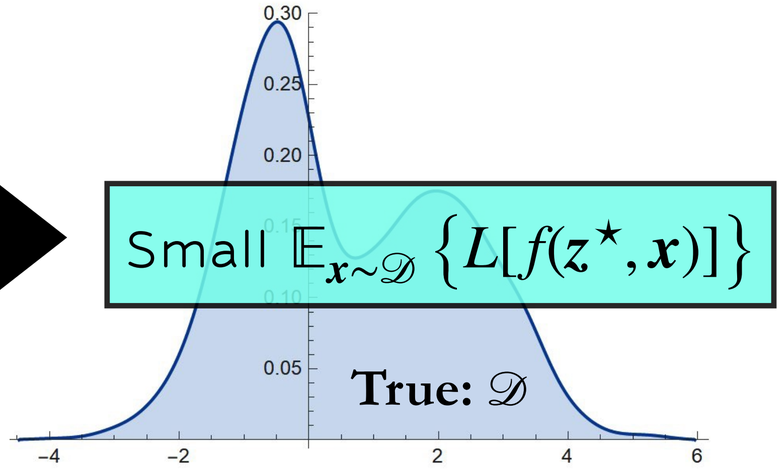
$$\min_{z\in\Omega} \; \mathbb{E}_{x\in\mathcal{S}} \left\{ L\left[h(z,x) \right] \right\}$$

Task

Model

Prior, Constraints

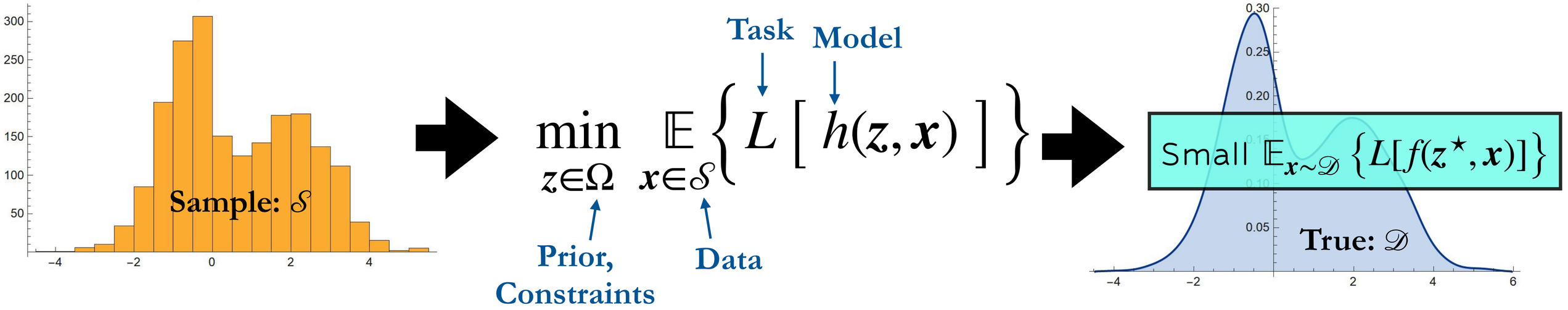
Data



- The goal of ML: Given $\mathcal{S} = \{x_i\}_{i=1}^m \sim \mathcal{D}$, identify $h(z^\star, x)$ that generalizes well for $\mathcal{D}$
  - Search from a good — expressive & efficiently computable — hypothesis class (model)
  - Local solutions may be better than the global solution for generalization
- Faster optimization? Better solution? Better model? What if data is quantum?

Introduction: ML

- Machines learn by solving as an optimization problem:



- The goal of ML: Given $\mathcal{S} = \{x_i\}_{i=1}^m \sim \mathcal{D}$, identify $h(z^\star, x)$ that generalizes well for $\mathcal{D}$
  - Search from a good — expressive & efficiently computable — hypothesis class (model)
  - Local solutions may be better than the global solution for generalization
- Faster optimization? Better solution? Better model? What if data is quantum?

👍 Provable quantum advantages exist

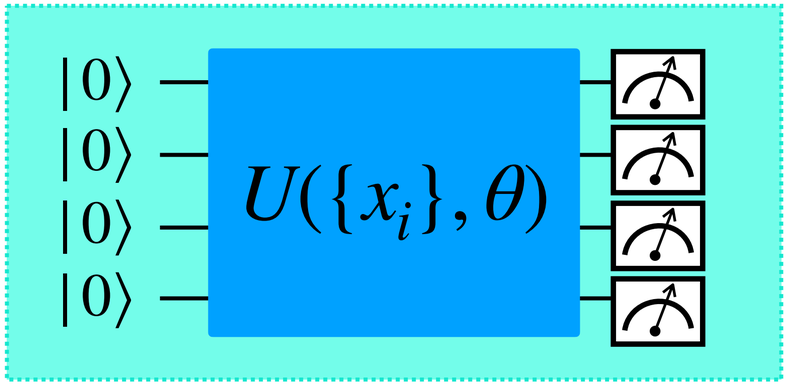
⛔ Not for near-term

$\mathrm{Tr}\left[A^\dagger B\right]$

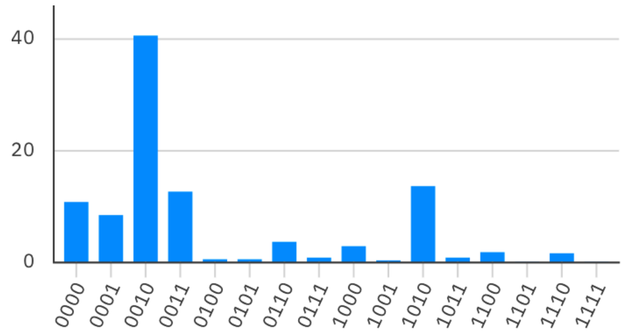
Quantum for AI

Quantum Supervised Learning

- Given: N labelled samples of data $\{(\rho(x_1), y_1), (\rho(x_2), y_2), \ldots, (\rho(x_N), y_N)\} \overset{\text{iid}}{\sim} D$
- Task: Find a prediction function $h(x)$ that minimizes the expected risk $R(h) = \mathbb{E}_{(\rho(x),y)\sim D}|h(x) - y|$



Sample repeatedly



$\langle M \rangle$

(1) Gradient-based optimization of parameterized quantum circuits

$$\langle M \rangle \rightarrow \frac{\partial}{\partial\theta}\mathrm{Tr}\left[MU(\theta)\rho(x_i)U^\dagger(\theta)\right] \Rightarrow h(x,\theta) = \mathrm{Tr}\left[U^\dagger(\theta)MU(\theta)\rho(x)\right] = \sum_{\omega\in\Omega} c(\theta)e^{i\omega x}$$

Fourier expansion

$$\langle M \rangle \rightarrow \mathrm{Tr}\left[\rho(x_i)\rho(x_j)\right] = k(x_i, x_j) \Rightarrow h(x,\alpha) = \mathrm{Tr}\left[\sum_{i=1}^{N} \alpha_i \rho(x_i)\rho(x)\right] = \sum_{i=1}^{N} \alpha_i k(x, x_i)$$

Kernel expansion

(2) Quantum kernel method: Train kernel SVM via convex optimization

Quantum Supervised Learning

- QML builds a model $q(x, \theta) = \mathrm{Tr}(M(\theta)\rho(x))$
- Let $\sigma_i \in \{I, X, Y, Z\}^{\otimes n}$ represent an n-qubit Pauli operator
- $M(\theta) = M(\theta)^\dagger \therefore M(\theta) = \sum_{i=1}^{4^n} \mu_i(\theta)\sigma_i$ where $\mu_i(\theta) \in \mathbb{R}$
- $\rho(x) = \rho(x)^\dagger \therefore \rho(x) = \sum_{i=1}^{4^n} \phi_i(x)\sigma_i$ where $\phi_i(x) \in \mathbb{R}$
- $q(x, \theta) = \sum_{i=1}^{4^n}\sum_{j=1}^{4^n} \mu_i(\theta)\phi_j(x)\mathrm{Tr}(\sigma_i\sigma_j) = \sum_{i=1}^{4^n} \mu_i(\theta)\phi_i(x)$ since $\mathrm{Tr}(\sigma_i\sigma_j) \propto \delta_{ij}$ $\therefore$ $q(x, \theta) = \boldsymbol{\mu}(\theta) \cdot \boldsymbol{\phi}(x)$

Model param

Non-linear feature map

⭐ QML builds a linear model (w.r.t $\boldsymbol{\phi}(x)$) in 4^n real vector space!

↳😲 For binary classification, an analytical solution has been known since the 1960s!

Optimal Solution to Quantum Supervised Learning

- For a set of training samples, $\mathcal{S} = \{\boldsymbol{x}_i^{(+)}, +1\}_{i=1}^{M_+} \cup \{\boldsymbol{x}_i^{(-)}, -1\}_{i=1}^{M_-}$, the loss becomes

$$\begin{aligned}\hat{R}(\mathcal{S}, \boldsymbol{\theta}) &= \frac{1}{M_+ + M_-}\left(\sum_{i=1}^{M_+} \Pr\left[E_-(\boldsymbol{\theta}) \,|\, x_i^{(+)}\right] + \sum_{i=1}^{M_-} \Pr\left[E_+(\boldsymbol{\theta}) \,|\, x_i^{(-)}\right]\right) \\ &= \frac{1}{M_+ + M_-}\left(\sum_{i=1}^{M_+} \mathrm{Tr}\left(E_- U(\boldsymbol{\theta}) \,|\, \boldsymbol{x}_i^{(+)}\rangle\langle \boldsymbol{x}_i^{(+)} \,|\, U^\dagger(\boldsymbol{\theta})\right) + \sum_{i=1}^{M_-} \mathrm{Tr}\left(E_+ U(\boldsymbol{\theta}) \,|\, \boldsymbol{x}_i^{(-)}\rangle\langle \boldsymbol{x}_i^{(-)} \,|\, U^\dagger(\boldsymbol{\theta})\right)\right) \\ &= p_+ \mathrm{Tr}\left(U^\dagger(\boldsymbol{\theta}) E_- U(\boldsymbol{\theta}) \rho_+\right) + p_- \mathrm{Tr}\left(U^\dagger(\boldsymbol{\theta}) E_+ U(\boldsymbol{\theta}) \rho_-\right)\end{aligned}$$

where $\rho_+ = \frac{1}{M_+}\sum_{i=1}^{M_+} |\boldsymbol{x}_i^{(+)}\rangle\langle \boldsymbol{x}_i^{(+)}|$ and $\rho_- = \frac{1}{M_-}\sum_{i=1}^{M_-} |\boldsymbol{x}_i^{(-)}\rangle\langle \boldsymbol{x}_i^{(-)}|$

- $\hat{R}(\mathcal{S}, \boldsymbol{\theta}) \geq \frac{1}{2} - D_{\mathrm{tr}}(p_+\rho_+, p_-\rho_-)$ where $D_{\mathrm{tr}}(A, B) = \frac{1}{2}\mathrm{Tr}\,|\,\mathrm{A} - \mathrm{B}\,|$ is the trace distance

Role of Quantum Data Embedding

- The trace distance is contractive under CPTP maps: $D_{\mathrm{tr}}(\Phi(\rho_+), \Phi(\rho_-)) \leq D_{\mathrm{tr}}(\rho_+, \rho_-)$
- The smallest possible training error is determined by the quantum data embedding
- The trace distance is strictly contractive if Φ is non-unitary
  - Noise will increase the training error
- For classical data classification, quantum data embedding should maximize $D_{\mathrm{tr}}(\rho_+, \rho_-)$

💡Use ML to optimize the data encoding → Neural Quantum Embedding (NQE)

- Improves the theoretical bounds (both training & generalization)
- Improves classification performance against noise

T Hur, IF Araujo, DK Park. Phys. Rev. A **110**, 022411 (2024)
T Hur, DK Park. arXiv:2411.06919 (ICML 2025)

Role of Quantum Data Embedding

- The trace distance is contractive under CPTP maps: $D_{\mathrm{tr}}(\Phi(\rho_+), \Phi(\rho_-)) \leq D_{\mathrm{tr}}(\rho_+, \rho_-)$
- The smallest possible training error is determined by the quantum data embedding
- The trace distance is strictly contractive if Φ is non-unitary
  - Noise will increase the training error
- For classical data classification, quantum data embedding should maximize $D_{\mathrm{tr}}(\rho_+, \rho_-)$

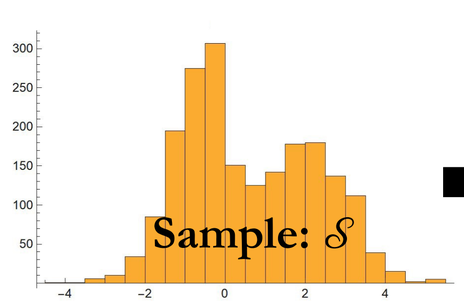
💡Use ML to optimize the data encoding → Neural Quantum Embedding (NQE)

- Improves the theoretical bounds (both training ☑ & generalization)🤔??
- Improves classification performance against noise

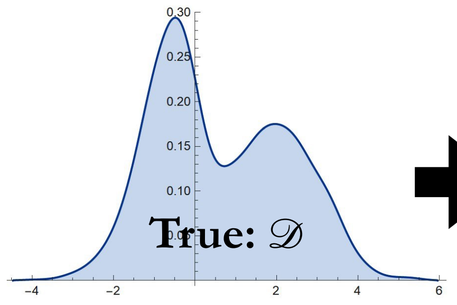
T Hur, IF Araujo, DK Park. Phys. Rev. A **110**, 022411 (2024)
T Hur, DK Park. arXiv:2411.06919 (ICML 2025)

Generalization: Finite Hypothesis Class

- Consider a finite Hypothesis Class $\mathscr{H} = \{h_1, h_2, \ldots, h_N\}$.



$\hat{R}(h)$ vs



$R(h)$ Hope $\hat{R}(h^\star) \approx R(h^\star)$

- For any $\delta \geq 0$, with probability higher than $1 - \delta$,

$$R(h) \leq \hat{R}(h) + \sqrt{\frac{\log |\mathscr{H}| + \log 2/\delta}{2m}}.$$

- Proof Sketch:

$$\Pr\left[\max_{h \in \mathscr{H}} |R(h) - \hat{R}(h)| \geq \epsilon\right] \leq \sum_{h \in \mathscr{H}} \Pr\left[|R(h) - \hat{R}(h)| \geq \epsilon\right] \leq |\mathscr{H}| \times 2\exp\left(-2m\epsilon^2\right)$$

- Here, complexity measure is simply $|\mathscr{H}| = N$. ⚠ **But for QML $|\mathscr{H}| = \infty$.**

Previous Studies on Generalization in QML

$$R(h) \leq \hat{R}(h) + O\left(\sqrt{\frac{T}{m}}\right),$$

T : # of trainable parameters

That is a "uniform bound".
Can be vacuous.

nature communications

Article

https://doi.org/10.1038/s41467-022-32550-3

Generalization in quantum machine learning from few training data

Received: 12 April 2022

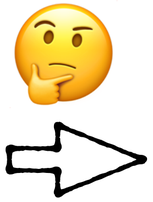
Accepted: 4 August 2022

Published online: 22 August 2022

Check for updates

Matthias C. Caro[1,2] ✉, Hsin-Yuan Huang[3,4], M. Cerezo[5,6], Kunal Sharma[7], Andrew Sornborger[5,8], Lukasz Cincio[9] & Patrick J. Coles[9]

Modern quantum machine learning (QML) methods involve variationally optimizing a parameterized quantum circuit on a training data set, and subsequently making predictions on a testing data set (i.e., generalizing). In this work, we provide a comprehensive study of generalization performance in QML after training on a limited number N of training data points. We show that the generalization error of a quantum machine learning model with T trainable gates scales at worst as $\sqrt{T/N}$. When only $K \ll T$ gates have undergone substantial change in the optimization process, we prove that the generalization error improves to $\sqrt{K/N}$. Our results imply that the compiling of unitaries into a polynomial number of native gates, a crucial application for the quantum computing industry that typically uses exponential-size training data, can be sped up significantly. We also show that classification of quantum states across a phase transition with a quantum convolutional neural network requires only a very small training data set. Other potential applications include learning quantum error correcting codes or quantum dynamical simulation. Our work injects new hope into the field of QML, as good generalization is guaranteed from few training data.



nature communications

Article

https://doi.org/10.1038/s41467-024-45882-z

Understanding quantum machine learning also requires rethinking generalization

Received: 3 July 2023

Accepted: 6 February 2024

Published online: 13 March 2024

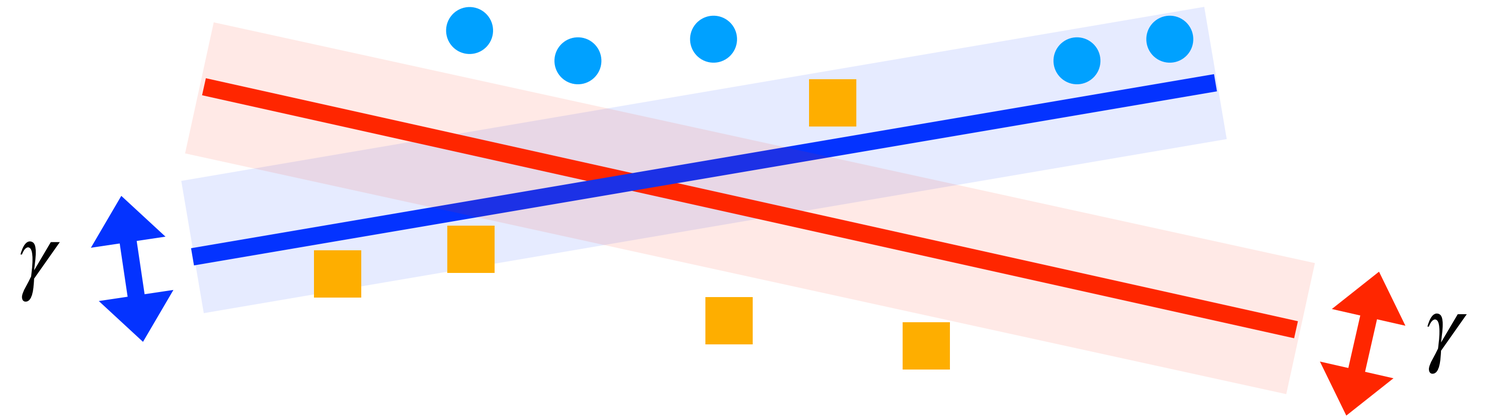
Check for updates

Elies Gil-Fuster[1,2], Jens Eisert[1,2,3] ✉ & Carlos Bravo-Prieto[1] ✉

Quantum machine learning models have shown successful generalization performance even when trained with few data. In this work, through systematic randomization experiments, we show that traditional approaches to understanding generalization fail to explain the behavior of such quantum models. Our experiments reveal that state-of-the-art quantum neural networks accurately fit random states and random labeling of training data. This ability to memorize random data defies current notions of small generalization error, problematizing approaches that build on complexity measures such as the VC dimension, the Rademacher complexity, and all their uniform relatives. We complement our empirical results with a theoretical construction showing that quantum neural networks can fit arbitrary labels to quantum states, hinting at their memorization ability. Our results do not preclude the possibility of good generalization with few training data but rather rule out any possible guarantees based only on the properties of the model family. These findings expose a fundamental challenge in the conventional understanding of generalization in quantum machine learning and highlight the need for a paradigm shift in the study of quantum models for machine learning tasks.

Generalization Bound with Margins

- Generalization in QML can be better characterized using margin [Bartlett et al. NIPS 30 (2017)]
- Consider the general k-class quantum classifier
  - For QML, $h \in \{\rho(x) \mapsto \left[\mathrm{Tr}(E_i U(\boldsymbol{\theta})\rho(x)U^\dagger(\boldsymbol{\theta}))\right]_{i=1:k} : \boldsymbol{\theta} \in \Theta\}$ (probability of classifying $\rho(x)$ as i)
  - Define margin as $\mathscr{M}(\boldsymbol{v}, y) := \boldsymbol{v}_y - \max_{i \neq y} \boldsymbol{v}_i$ (success prob − largest competing error prob)
- Empirical margin loss: $\hat{R}_\gamma(h) = \frac{1}{m}\sum_{i=1}^{m} \mathbb{I}[\mathscr{M}(h(\rho(x_i), y_i) \leq \gamma]$ where $\mathbb{I}[z] = \begin{cases} 1 & \text{if } z \\ 0 & \text{otherwise} \end{cases}$
  - ↳ cf. Empirical loss $\hat{R}(h) = \frac{1}{m}\sum_{i=1}^{m} \mathbb{I}[\arg\max h(\rho(x_i)) \neq y_i]$



Generalization Bound with Margins

- Theorem. Generalization bound with margins:

  Consider an n-qubit QNNs consist of unitary $U \in \mathbb{U}_{\mathrm{QNN}}$ and POVMs $\{E_i\}_{i=1}^k$ for k-class classification.

  Let b be a distance bound such that $\|U - U_{\mathrm{ref}}\|_{2,1} \leq b$ holds for any $U \in \mathbb{U}_{\mathrm{QNN}}$ with a reference unitary matrix U_{ref}.

  Then, for any $\delta > 0$ and $\gamma > 0$, with probability at least $1 - \delta$ over an i.i.d. sample $\mathcal{S}$ of size m, the following holds for all $h \in \{\rho \mapsto \{\mathrm{Tr}(E_i U \rho U^\dagger)\}_{i=1}^k : U \in \mathbb{U}_{\mathrm{QNN}}\}$

$$R(h) \leq \hat{R}_\gamma(h) + \tilde{O}\left(\frac{b}{\gamma}\sqrt{\frac{n\sum_{i=1}^k \|E_i\|^2}{m}} + \sqrt{\frac{\ln(1/\delta)}{m}}\right)$$

T Hur, DK Park. arXiv:2411.06919 (ICML 2025)

Generalization Bound with Margins

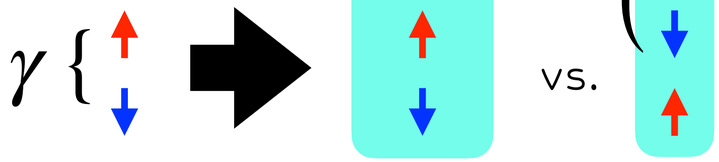
- Theorem. Generalization bound with margins:

  Consider an n-qubit QNNs consist of unitary $U \in \mathbb{U}_{\mathrm{QNN}}$ and POVMs $\{E_i\}_{i=1}^k$ for k-class classification.

  Let b be a distance bound such that $\|U - U_{\mathrm{ref}}\|_{2,1} \leq b$ holds for any $U \in \mathbb{U}_{\mathrm{QNN}}$ with a reference unitary matrix U_{ref}.

  Then, for any $\delta > 0$ and $\gamma > 0$, with probability at least $1 - \delta$ over an i.i.d. sample $\mathcal{S}$ of size m, the following holds for all $h \in \{\rho \mapsto \{\mathrm{Tr}(E_i U \rho U^\dagger)\}_{i=1}^k : U \in \mathbb{U}_{\mathrm{QNN}}\}$

$$R(h) \leq \hat{R}_\gamma(h) + \tilde{O}\left(\frac{b}{\gamma}\sqrt{\frac{n\sum_{i=1}^k \|E_i\|^2}{m}} + \sqrt{\frac{\ln(1/\delta)}{m}}\right)$$



T Hur, DK Park. arXiv:2411.06919 (ICML 2025)

Generalization

Proof Sketch

$\mathfrak{R}$: Rademacher Complexity
$\mathcal{N}$: Covering Number
$\mathcal{F}$: Hypothesis Class of QML model

Takeaway

The proof combines:

- **statistical learning theory**: margin loss, Rademacher complexity
- **entropy methods**: covering numbers, Dudley integral
- **quantum information**: Lipschitz geometry of measurement operators
- **matrix approximation**: Maurey-type sparsification

1. Rademacher Complexity

$$R(h) \leq \hat{R}_\gamma(h) + 2\mathfrak{R}((\mathcal{F}_\gamma)_{|S}) + 3\sqrt{\frac{\ln(2/\delta)}{2m}}$$

2. Dudley's Entropy Integral

$$\mathfrak{R}(U) \leq \inf_{\alpha>0}\left(\frac{4\alpha}{\sqrt{m}} + \frac{12}{m}\int_\alpha^{\sqrt{m}} \sqrt{\ln \mathcal{N}(U,\beta,\|\cdot\|_2)}d\beta\right)$$

3. Covering Number Bound for QML model

$$\ln \mathcal{N}\left((\mathcal{F}_\gamma)_{|S}, \epsilon, ||\cdot||_2\right) \leq \ln \mathcal{N}\left(\{UX : U \in \mathbb{U}_{\text{QNN}}\}, \frac{\epsilon\gamma}{4E}, ||\cdot||_2\right) \leq \left\lceil \frac{32mb^2E^2}{\epsilon^2\gamma^2} \right\rceil \ln 4N^2$$

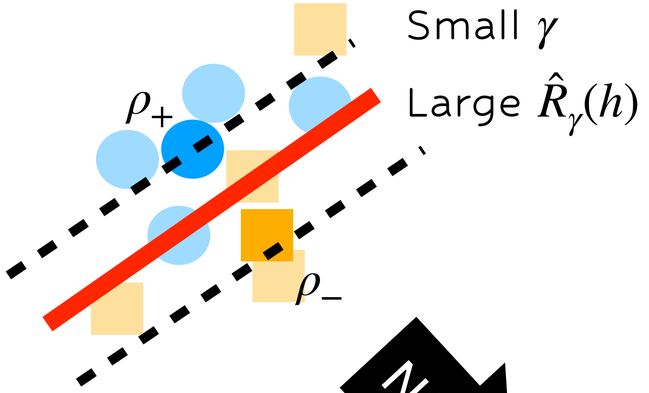
Lipschitz property of quantum measurement

Maurey's Sparsification Lemma

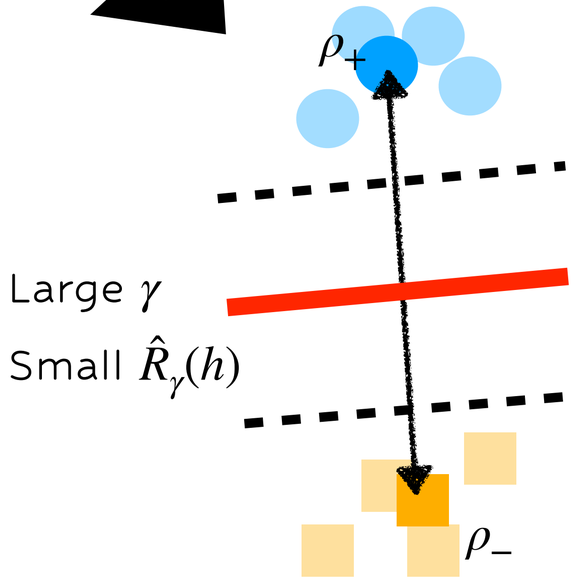
Improving Generalization via NQE

- Theory-guided improvement of generalization in QML:
  - D_{tr} upper-bounds mean (μ) of the margin distribution!

$$\mu = \frac{1}{M}\sum_{i=1}^{M} \mathscr{M}(h(\rho_i), y_i) \leq D_{\text{tr}}(p_-\rho_+, p_-\rho_-)$$

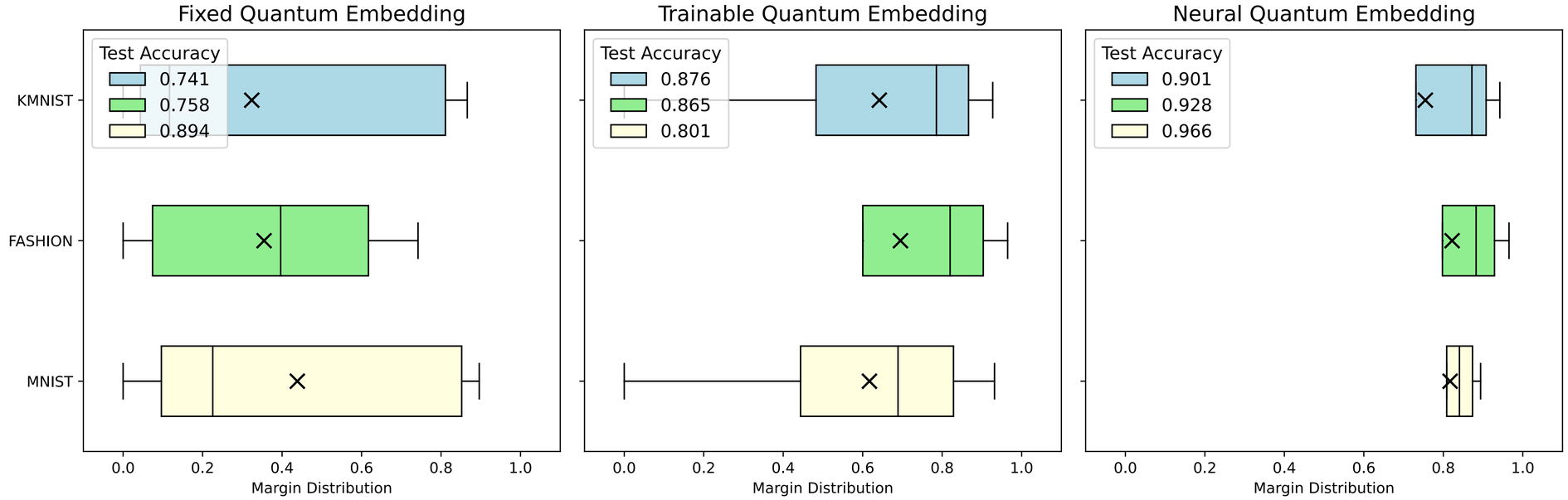


NQE



- 8-qubit simulation results:

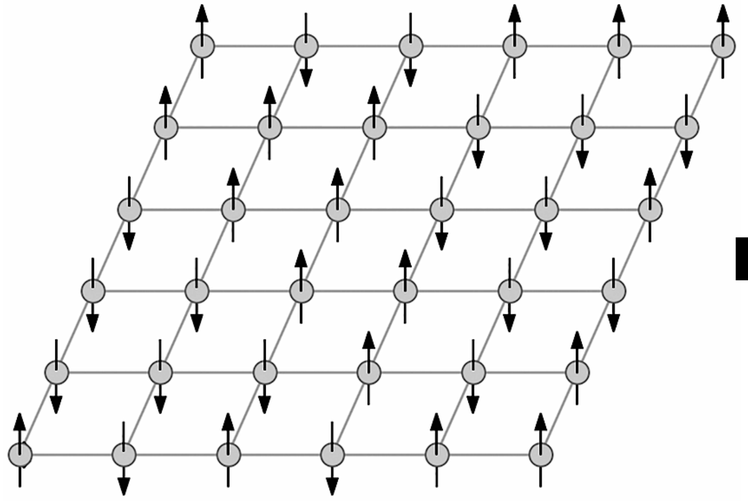
Our method

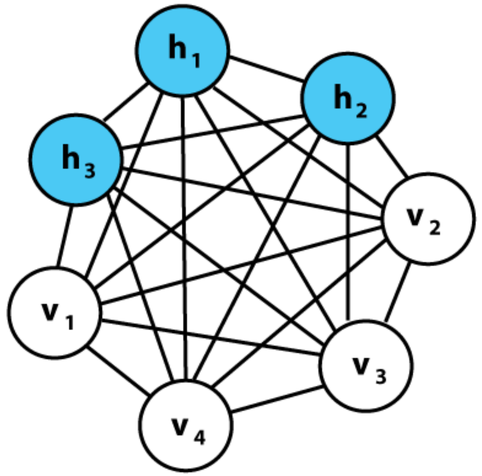


T Hur, DK Park. arXiv:2411.06919 (ICML 2025)

Physics-Inspired Generative Modeling

- An ancestor of the Modern Generative AI: Boltzmann Machines (inspired by physics)



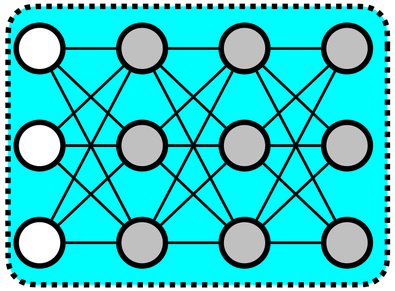


Hard to sample from Boltzmann distribution!

010001
100101

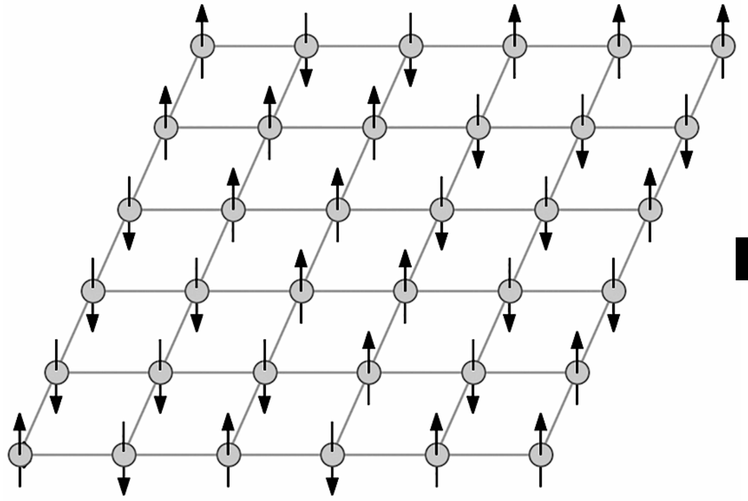
Restrict the network & approximate!

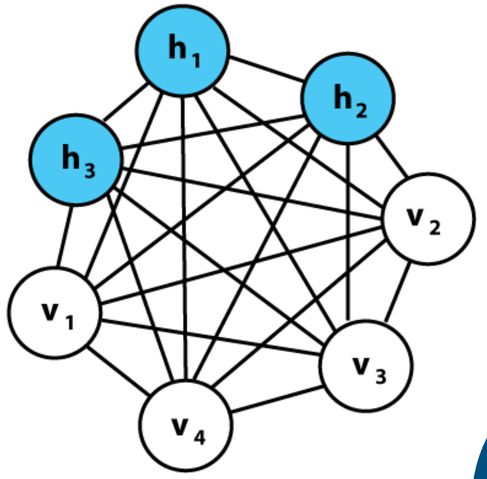
Triggered resurgence of deep learning



Physics-Inspired Generative Modeling

- An ancestor of the Modern Generative AI: Boltzmann Machines (inspired by physics)





Hard to sample from Boltzmann distribution!

010001
100101

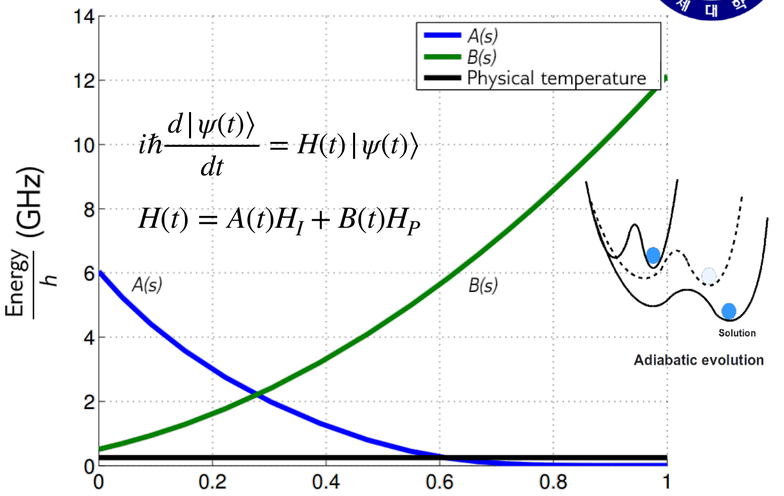
Restrict the network & approximate!

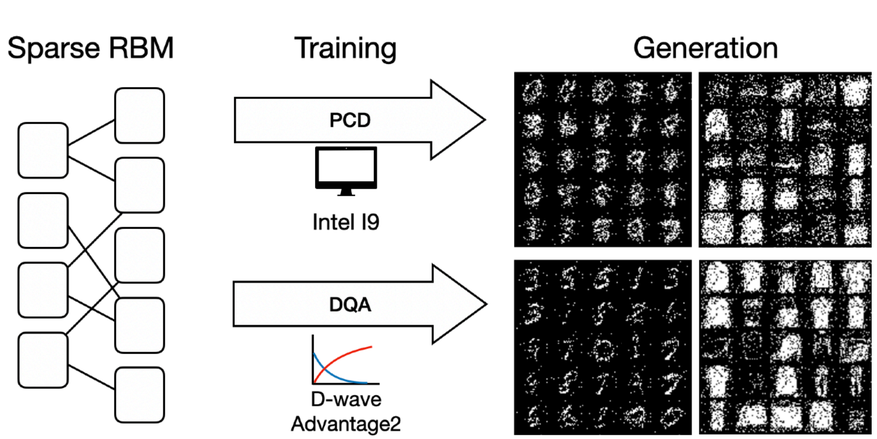
Wouldn't it be more natural to use a physical system that is described by BD?

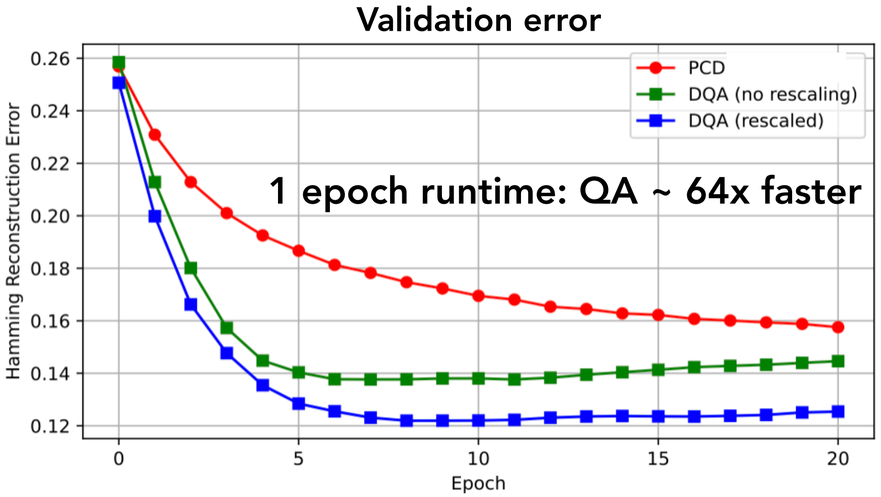
Generative Modeling via Quantum Annealing

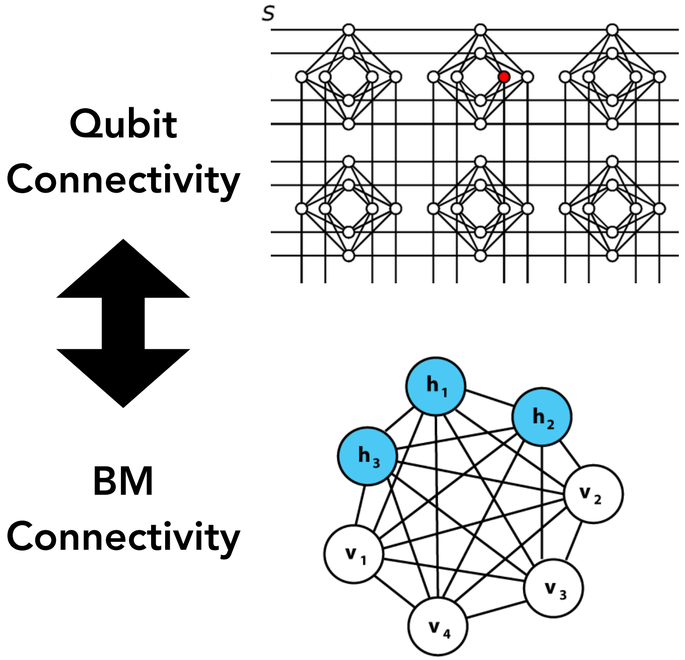
- Quantum annealer:
  - Can be controlled to generate Boltzmann-distributed data
  - Use for training a Boltzmann machine

  → Faster and better training than classical (e.g. MCMC)!





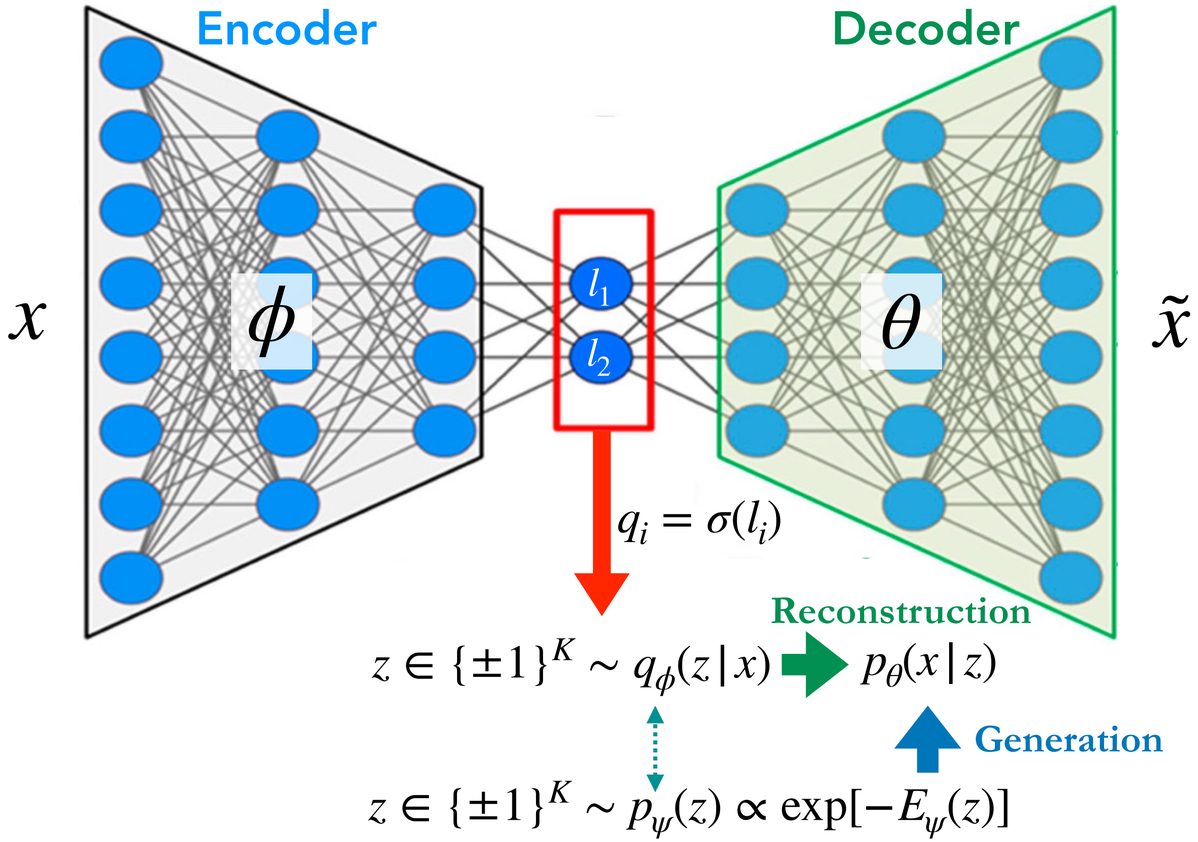


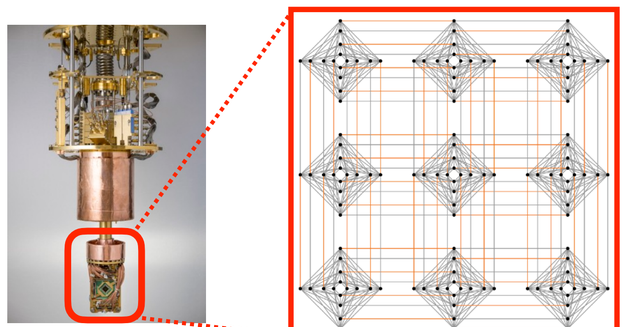


- Outlook: Revival of fully-connected BM (which was ruled out in 1980s!)

G Kim, J Gyhm, DK Park. PRE 113 035302 (2026)

Variational Autoencoders with General Boltzmann Priors





Maximize:

$$\mathscr{L}(\theta,\phi,\psi) = \underbrace{\mathbb{E}_{q_\phi(z|x)}\left[\log p_\theta(x|z)\right]}_{\text{reconstruction}} - \underbrace{D_{\mathrm{KL}}\left(q_\phi(z|x)\,\|\,p_\psi(z)\right)}_{\text{prior matching}}$$

where

$$D_{\mathrm{KL}}\left(q_\phi(z|x)\,\|\,p_\psi(z)\right) = \underbrace{\mathbb{E}_{q_\phi(z|x)}\left[E_\psi(z)\right]}_{\text{energy}} + \log Z_\psi - \underbrace{S\left(q_\phi(z|x)\right)}_{\text{entropy}}$$

Gradient:

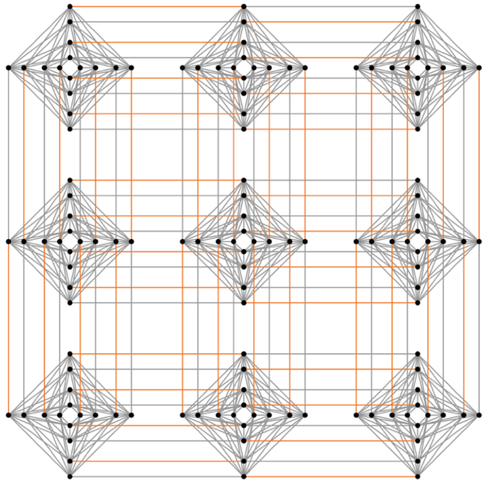
$$\nabla_\psi\left(\mathbb{E}_{q_\phi(z|x)}[E_\psi(z)] + \log Z_\psi\right)$$
$$= \mathbb{E}_{q_\phi(z|x)}[\nabla_\psi E_\psi(z)] - \underbrace{\mathbb{E}_{p_\psi(z)}[\nabla_\psi E_\psi(z)]}_{\text{Boltzmann sampling!}}$$

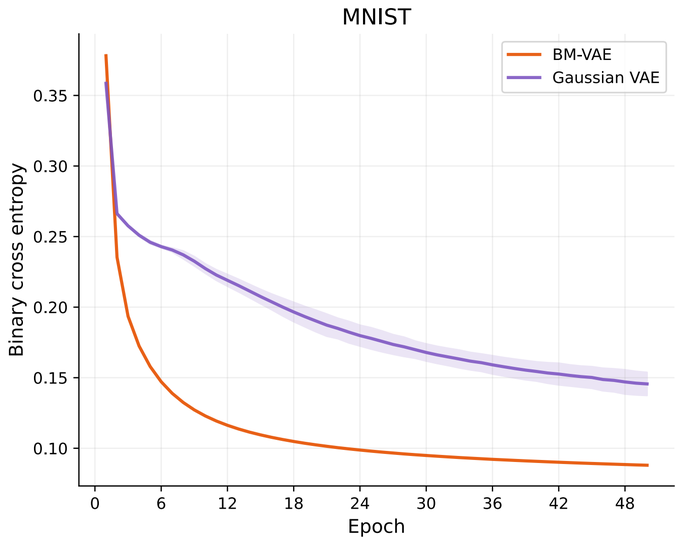
Thermodynamics-inspired representation learning, enabled by quantum computing

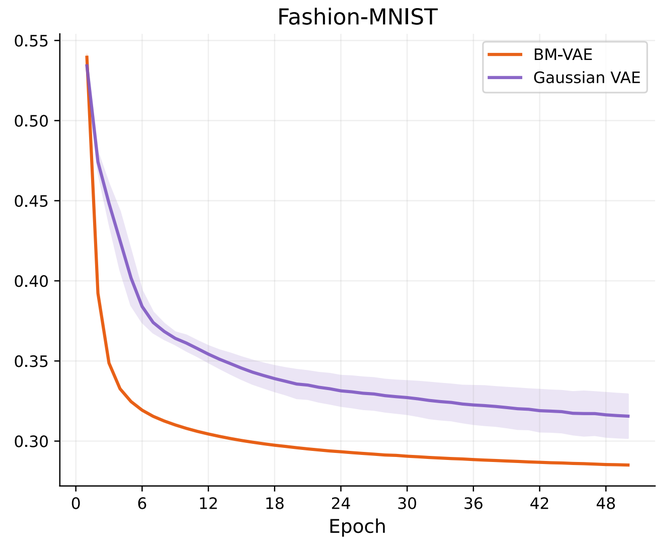
G Kim, DK Park. arXiv:2604.00919

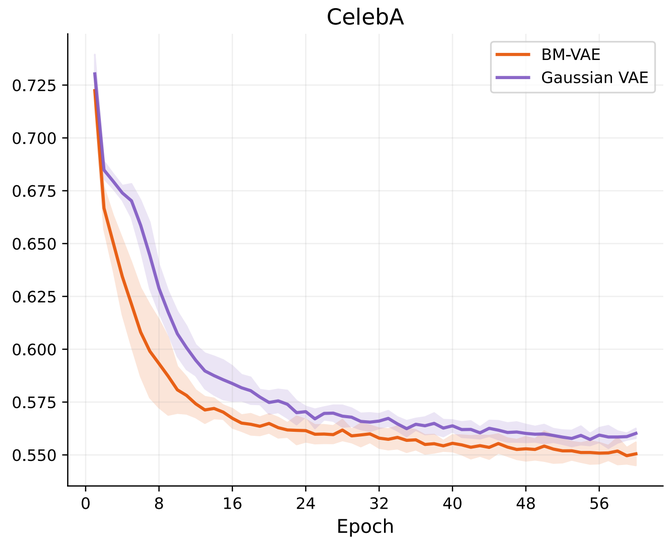
Variational Autoencoders with General Boltzmann Priors

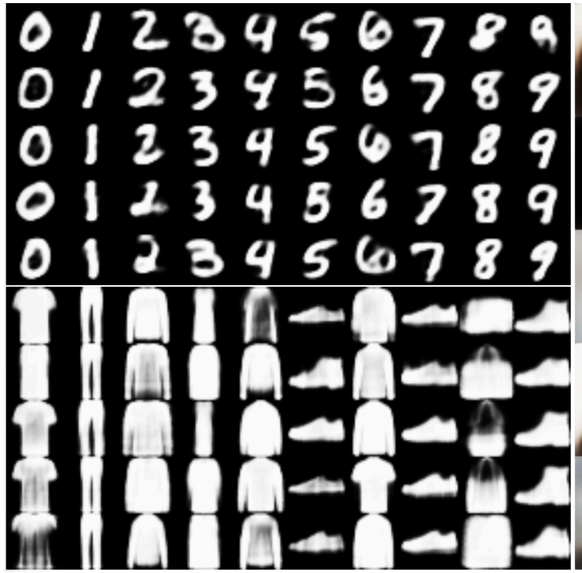
2000 Qubits

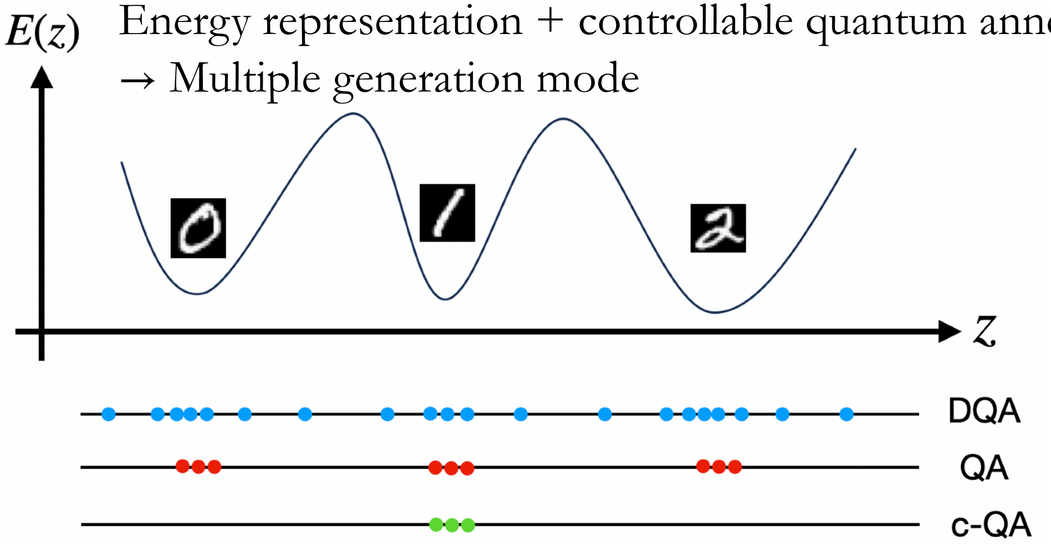










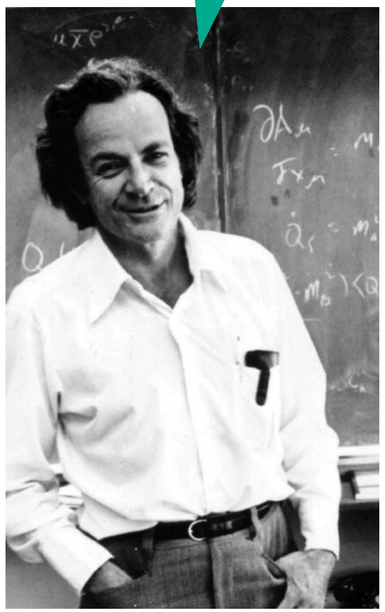


AI for Quantum

Quantum Eigensolvers: Why it Matters?

$$i\hbar\frac{d\psi(t)}{dt} = H(t)\psi(t) \qquad H(t)\psi(t) = E(t)\psi(t)$$

Nature isn't classical, dammit, and if you want to make a simulation of nature, you'd better make it quantum mechanical…



🧪 Chemistry & Materials

- Molecular properties
  - Core to drug discovery, battery and new materials research

🔬 Physics

- Quantum many-body systems
  - Central to exploring new physical phenomena

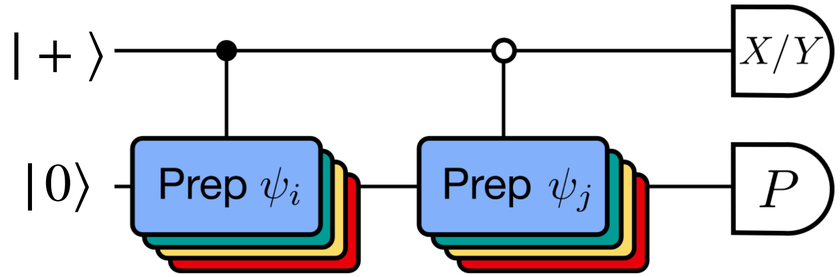
📈 Optimization

- Map combinatorial optimization to Hamiltonian eigenvalue problems

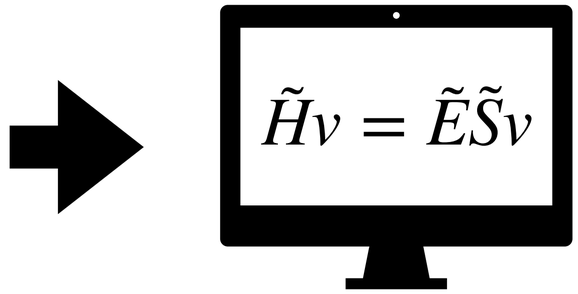
Krylov Subspace Diagonalization

Hamiltonian: $H(\vec{x})$ **Krylov subspace:** $K = \mathrm{span}\{e^{-ijH(\vec{x})\Delta t}|\psi_0\rangle\}, j = 0,1,\ldots,D-1$

$D \ll \dim(H)$ for speed-up



$$\begin{cases} P = H: & \tilde{H}_{jk} = \langle\psi_j|H|\psi_k\rangle \\ P = I: & \tilde{S}_{jk} = \langle\psi_j|\psi_k\rangle \end{cases}$$

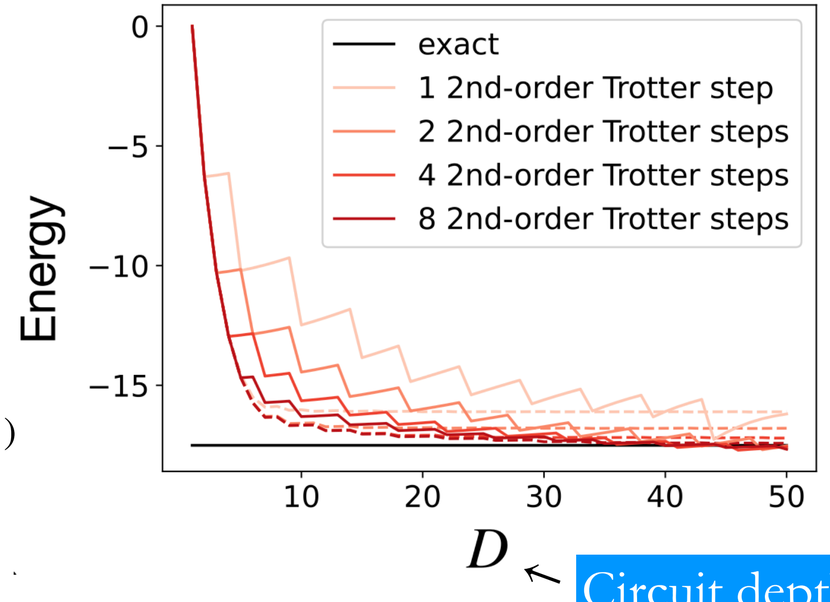


$$H|\phi_i\rangle = E_i|\phi_i\rangle, E_0 \le E_1 \le \ldots \le E_{N-1}$$

$$\Delta E_j = E_j - E_0$$

$$0 \le E_0 - \tilde{E} \le \varepsilon$$

$$\varepsilon = 8\Delta E_{N-1}\left(\frac{1-|\langle\psi_0|\phi_0\rangle|^2}{|\langle\psi_0|\phi_0\rangle|^2}\right)\left(1+\frac{\pi\Delta E_1}{\Delta E_{N-1}}\right)^{-(D-1)}$$



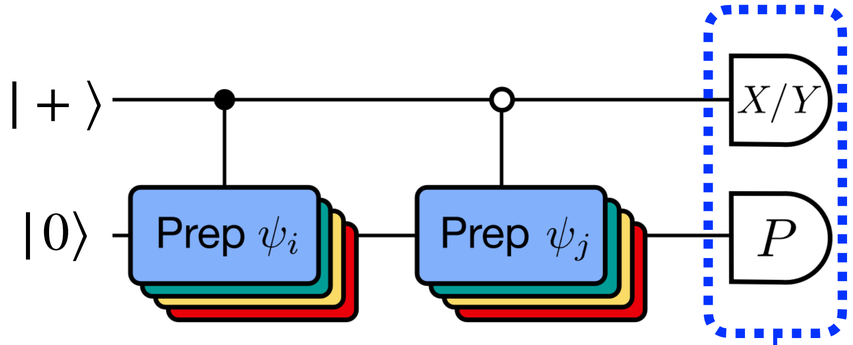
Circuit depth grows linearly with D

Figures from Nature Communications **16** 5014 (2025)

Generative Krylov Subspace Diagonalization

Hamiltonian: $H(\vec{x})$ **Krylov subspace:** $K = \mathrm{span}\{e^{-ijH(\vec{x})\Delta t}|\psi_0\rangle\}, j = 0,1,\ldots,D-1$

$D \ll \dim(H)$ for speed-up



$$\begin{cases} P = H: & \tilde{H}_{jk} = \langle\psi_j|H|\psi_k\rangle \\ P = I: & \tilde{S}_{jk} = \langle\psi_j|\psi_k\rangle \end{cases}$$

$$\tilde{H}v = \tilde{E}\tilde{S}v$$

$\{\vec{a}^{(i)}\}_{i=1,\ldots,N}$

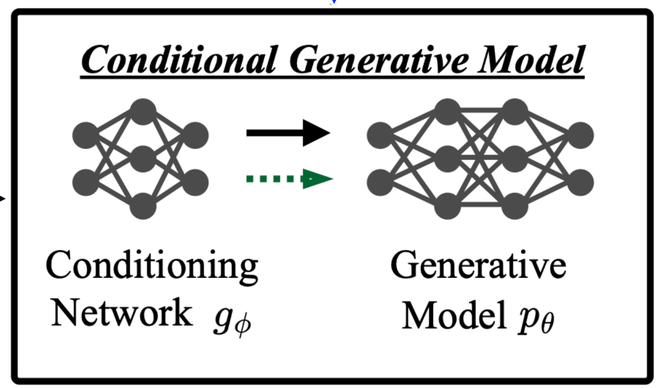
$$\max \sum_i -\log p_\theta[\vec{a}^{(i)}|g_\phi(\vec{x}^{(i)})]$$

$\{\vec{x}^{(i)}\}_{i=1,\ldots,N}$

Train

Test

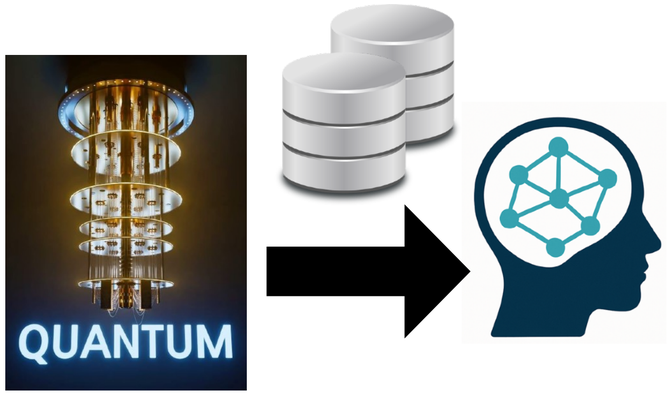
$\vec{x}_{\text{test}}$



$\vec{a}$

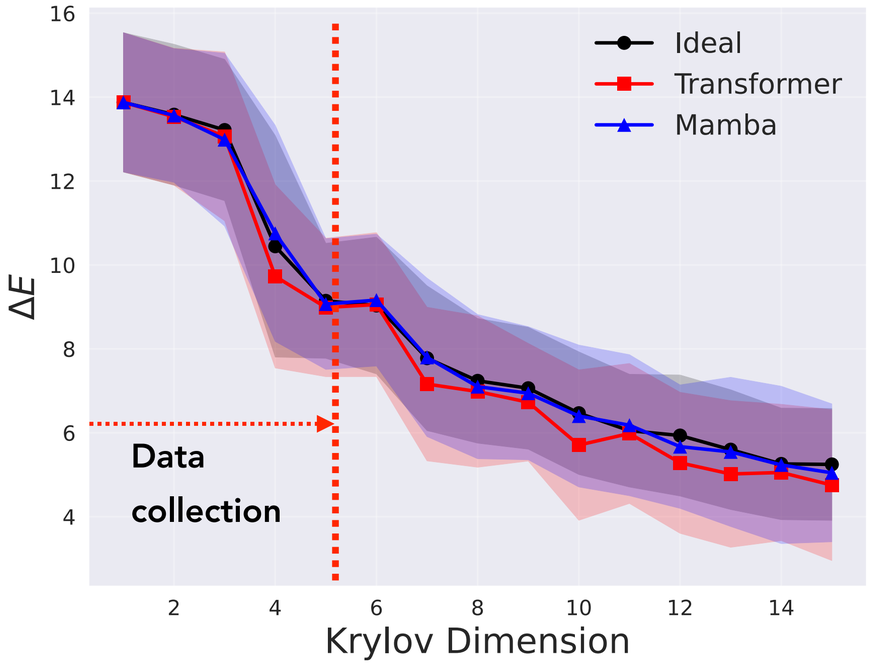
ML-assisted Quantum Eigensolver

- **Training Data**: Approximate solutions of quantum systems pre-computed using a quantum computer
- **Task**: Predict more accurate solutions for new quantum systems
- **Data generation on QC + learning & inference on CC → New paradigm for AI-based quantum simulation**



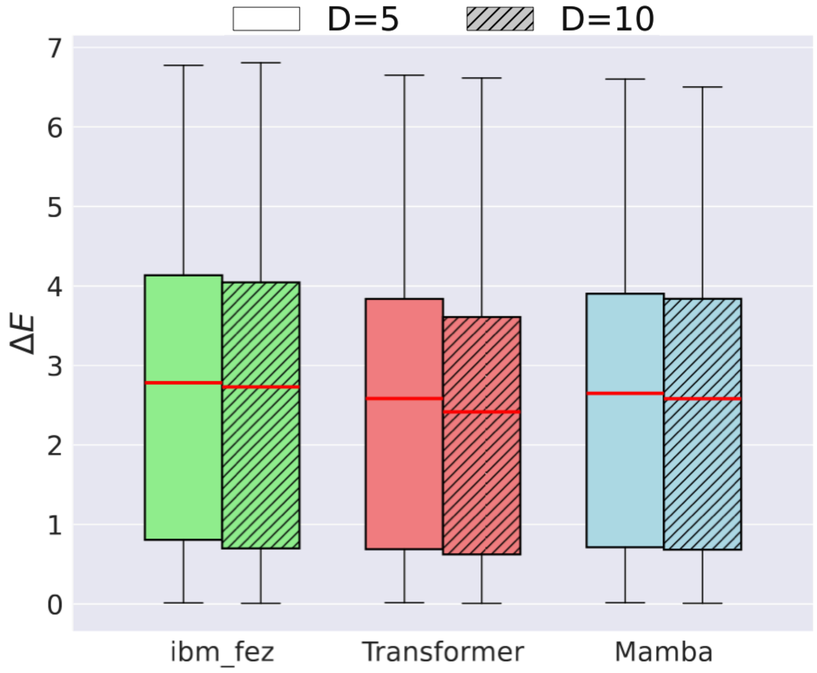
15-qubit simulation

$$H(\vec{J}) = \sum J_{ij}(X_iX_j + Y_iY_j + Z_iZ_j)$$



20-qubit experiment

$$H(\vec{J}) = \sum J_{ij}(X_iX_j + Y_iY_j) + Z_iZ_j$$



30 training / 20 test

C Lee, D K Park. arXiv:2512.19420

Quantum Computing Is Hard in Practice

- Qubits are extremely sensitive to their surroundings → Easy to lose quantumness
- Qubits must interact accurately with each other → Hard to isolate from noise
- Gate and measurement errors are unavoidable

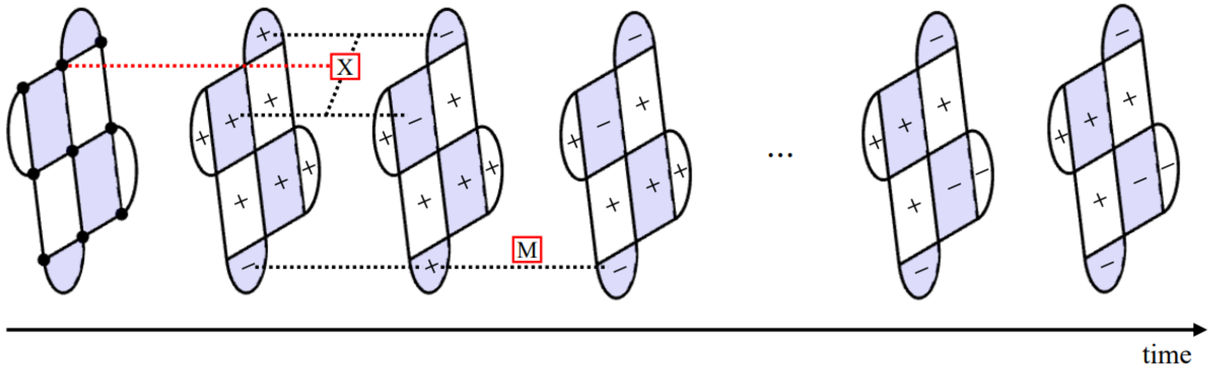
➡**Without error control, quantum advantage cannot be realized in practice** 😰

- How to combat noise & imperfections?
  - Error suppression: Algorithmically reduce system-bath interaction
  - Error mitigation: Post-processing of a large data obtained via controlled repetitions
  - **Error correction ← Enables scalable & arbitrarily long computation**

Basics of Decoding

- Quantum error correction (QEC) is essential for scalable quantum computing
- QEC encodes logical information across many physical qubits and detects errors via syndrome measurement

E.g.) Surface code



Syndrome outcomes

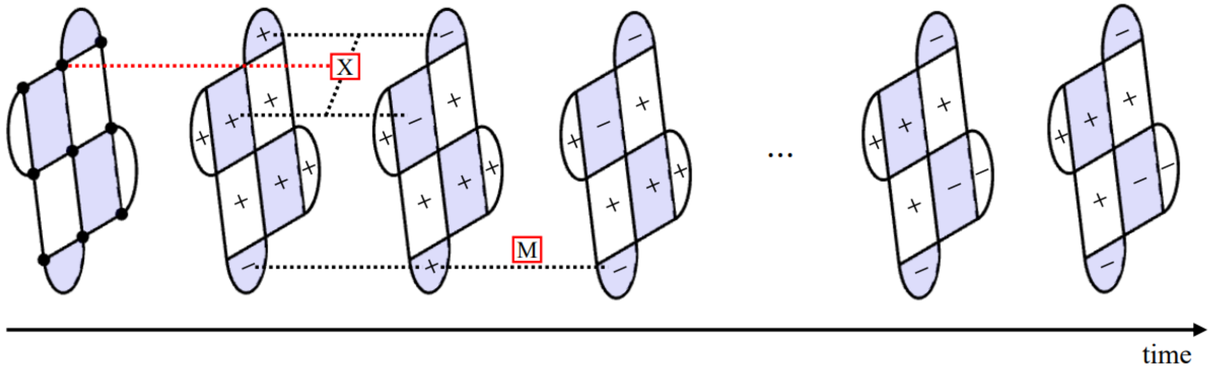
$$
\begin{aligned}
t = 0 \quad &\rightarrow \left[0001\ 0000\right] \\
t = 1 \quad &\rightarrow \left[1100\ 0000\right] \\
t = 2 \quad &\rightarrow \left[1101\ 0000\right] \\
&\ \ \vdots \\
t = n-1 \quad &\rightarrow \left[1011\ 0001\right] \\
t = n \quad &\rightarrow \left[1011\ 0001\right]
\end{aligned}
$$

decode → $\{I_L, X_L, Y_L, Z_L\}$

Basics of Decoding

- Quantum error correction (QEC) is essential for scalable quantum computing
- QEC encodes logical information across many physical qubits and detects errors via syndrome measurement

E.g.) Surface code



Syndrome outcomes

$$
\begin{aligned}
t = 0 \quad &\rightarrow \quad [0001\ 0000] \\
t = 1 \quad &\rightarrow \quad [1100\ 0000] \\
t = 2 \quad &\rightarrow \quad [1101\ 0000] \\
&\vdots \\
t = n-1 \quad &\rightarrow \quad [1011\ 0001] \\
t = n \quad &\rightarrow \quad [1011\ 0001]
\end{aligned}
$$

😰 Not unique!

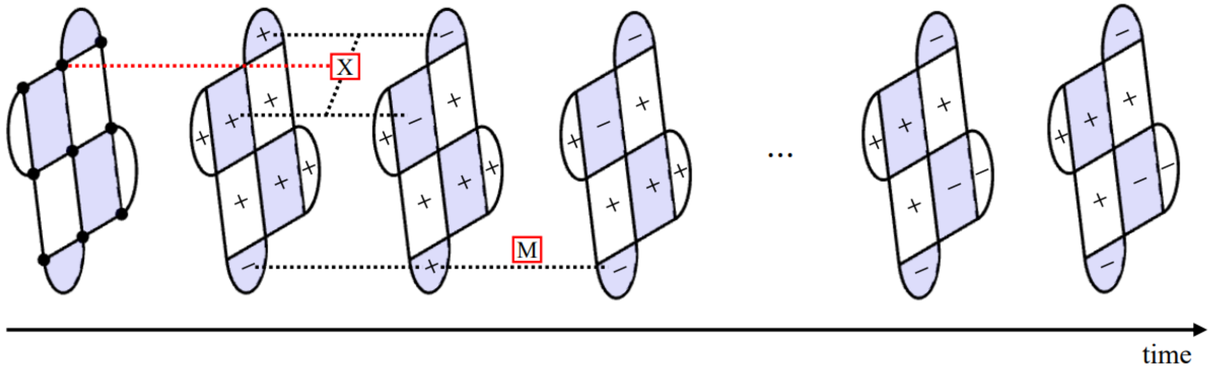
decode → $\{I_L, X_L, Y_L, Z_L\}$

- Maximum likelihood decoding is NP-Hard!
- Approximation needed → Depends on the noise model

Basics of Decoding

- Quantum error correction (QEC) is essential for scalable quantum computing
- QEC encodes logical information across many physical qubits and detects errors via syndrome measurement

E.g.) Surface code



Syndrome outcomes

$$
\begin{aligned}
t = 0 \quad &\rightarrow \quad [0001\ 0000] \\
t = 1 \quad &\rightarrow \quad [1100\ 0000] \\
t = 2 \quad &\rightarrow \quad [1101\ 0000] \\
&\vdots \\
t = n-1 \quad &\rightarrow \quad [1011\ 0001] \\
t = n \quad &\rightarrow \quad [1011\ 0001]
\end{aligned}
$$

decode

$\{I_L, X_L, Y_L, Z_L\}$

😰 Not unique!

Classification!

- Maximum likelihood decoding is NP-Hard!
- Approximation needed → Depends on the noise model
- Neural decoder: Agnostic to hardware noise & adaptable to various settings
- Decoder must be **accurate** and **fast**

ML-assisted Quantum Error Correction

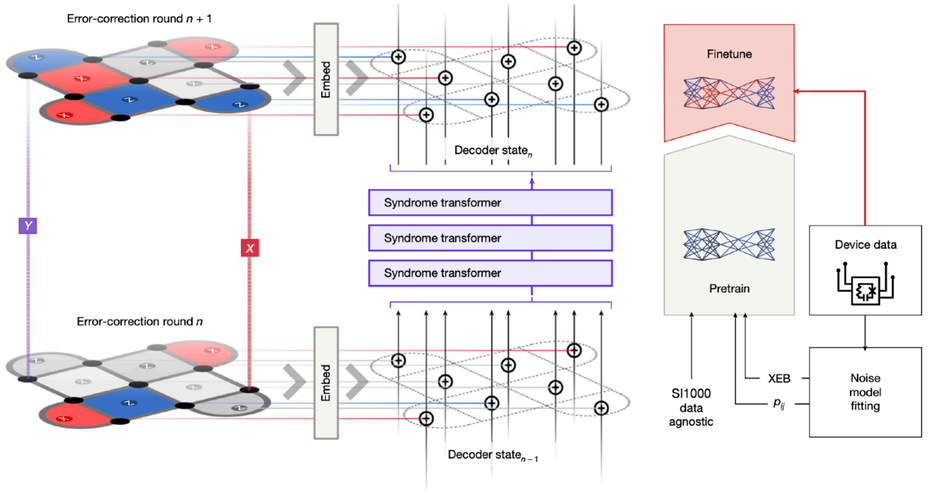
- Decoders based on sequence models (e.g. Transformer, Mamba) have demonstrated excellent performance → AI for the development of reliable and scalable quantum hardware!

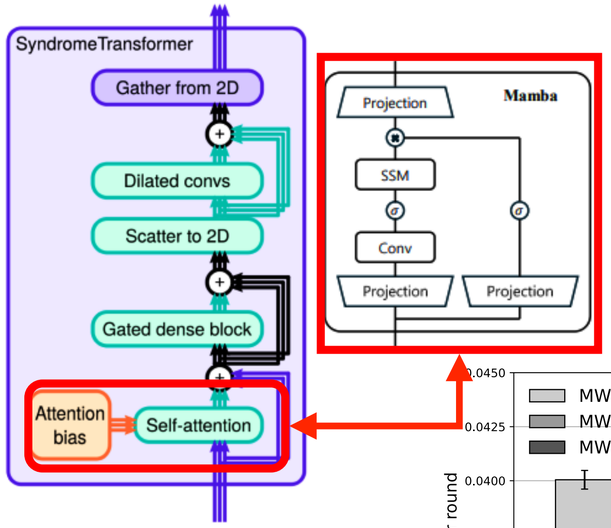
Learning high-accuracy error decoding for quantum processors

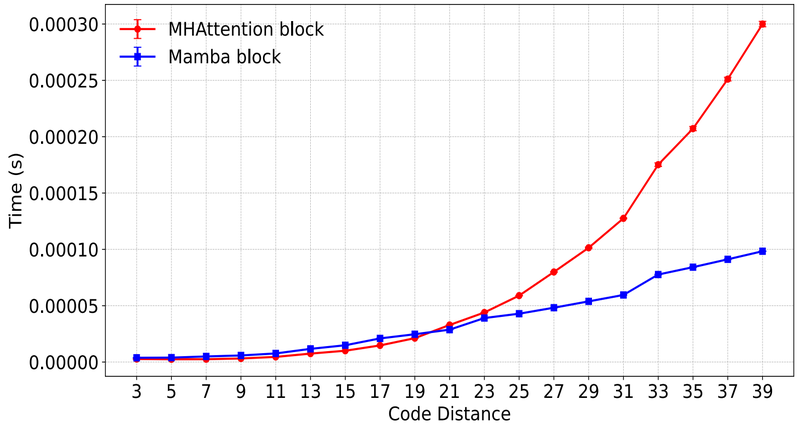
Johannes Bausch, Andrew W. Senior, Francisco J. H. Heras, Thomas Edlich, Alex Davies, Michael Newman, Cody Jones, Kevin Satzinger, Murphy Yuezhen Niu, Sam Blackwell, George Holland, Dvir Kafri, Juan Atalaya, Craig Gidney, Demis Hassabis, Sergio Boixo, Hartmut Neven & Pushmeet Kohli

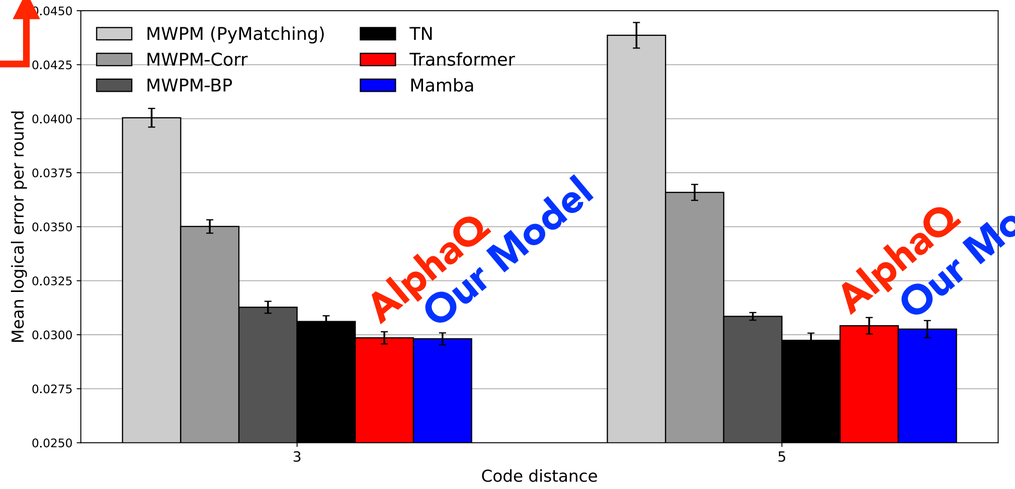
Nature **635**, 834–840 (2024) | Cite this article

a.k.a AlphaQubit









C Lee, T Hur, D Park. arXiv:2510.22724

Summary

- Quantum for AI: Quantum feature maps and energy-based generative models offer new approaches to prediction, representation and generation
- AI for Quantum: Train classical AI on quantum-generated data for scalable quantum simulation & computation
- Mathematical tools can help analyze and improve QML (e.g. margin-based generalization analysis)
- When Quantum meets AI, not only do new algorithms emerge, but so do new mathematical questions, e.g.,
  - For which practical learning tasks do quantum feature maps offer provable advantages?
  - When does classical hardness of quantum sampling differ from its statistical learnability for prediction tasks?
  - Tighter generalization bounds for QML
  - Generalization & sample complexity of energy-based models
  - How should ML models reflect the structures and symmetries of quantum-native tasks and data?
  - How should sample complexity scale with system size in AI for quantum?