# A rigorous and robust quantum speed-up in supervised machine learning

Yunchao Liu (UC Berkeley)

Joint work with Srinivasan Arunachalam and Kristan Temme (IBM Research)

arxiv: 2010.02174

# Exciting developments in NISQ experiments

- Quantum supremacy [Arute et al’19, Zhong et al’20]

- Quantum chemistry [Arute et al’20]
- Combinatorial optimization [Harrigan et al’21]
- Machine learning [Peters et al’21]
- etc…

# Exciting developments in NISQ experiments

- Quantum supremacy [Arute et al'19, Zhong et al'20]

- Quantum chemistry [Arute et al'20]
- Combinatorial optimization [Arute et al'21]
- Machine learning [Peters et al'21]
- etc…

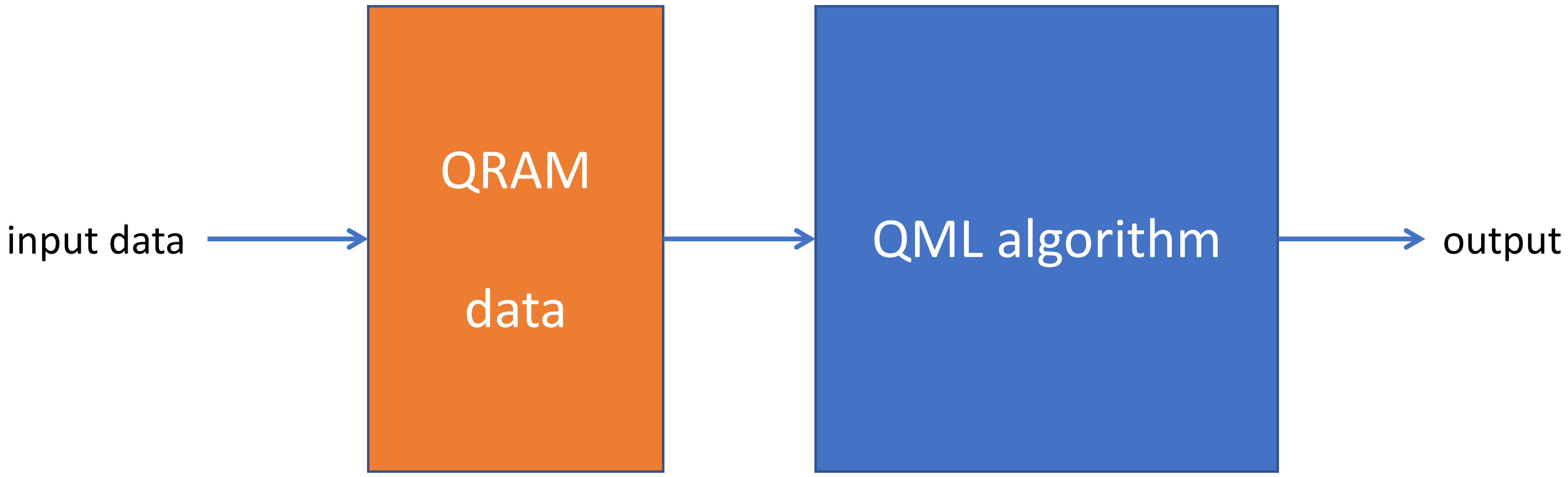
This talk: theoretical evidence of quantum advantage using quantum kernel methods

# Quantum machine learning algorithms

- QRAM-based algorithms [HHL'09, etc…]
  - Amplitude encoding: $n$ dimensional vector stored in $\log n$ qubits
  - Pros: $\mathrm{polylog}(n)$ running time

# Quantum machine learning algorithms

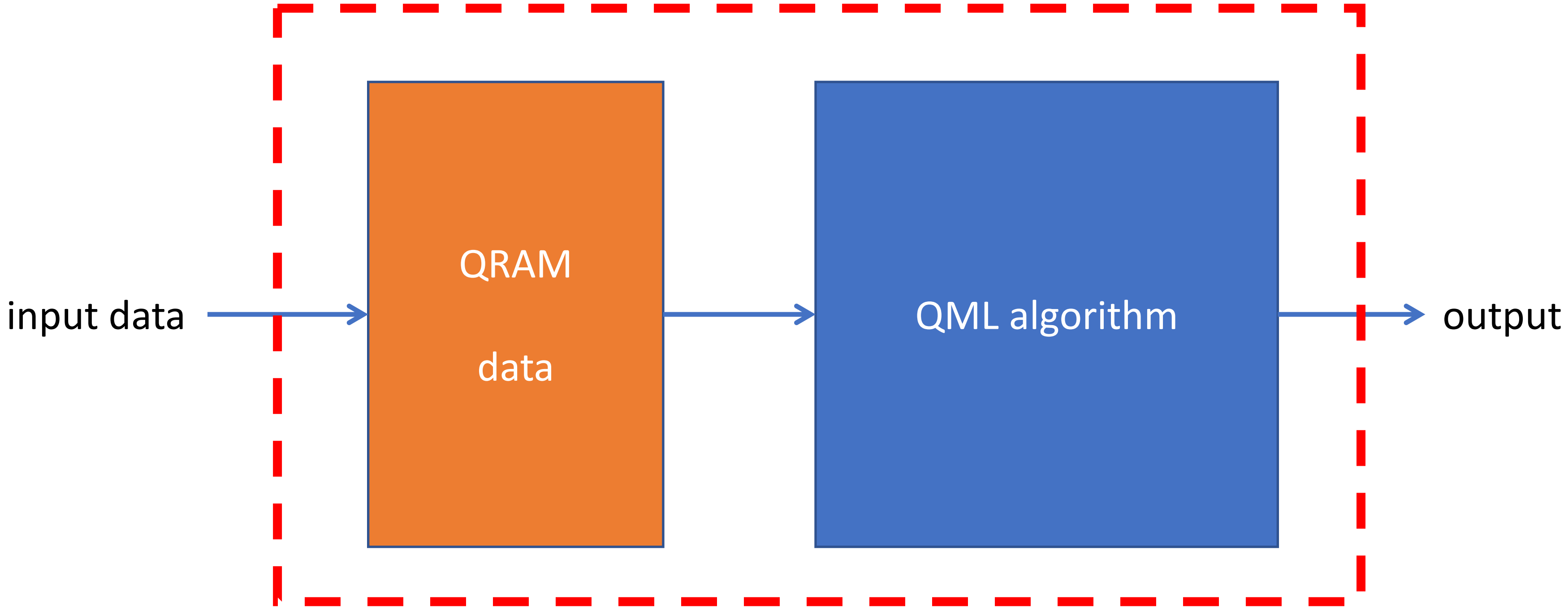
- QRAM-based algorithms [HHL'09, etc...]




Limitations: QRAM is hard to implement

# Quantum machine learning algorithms

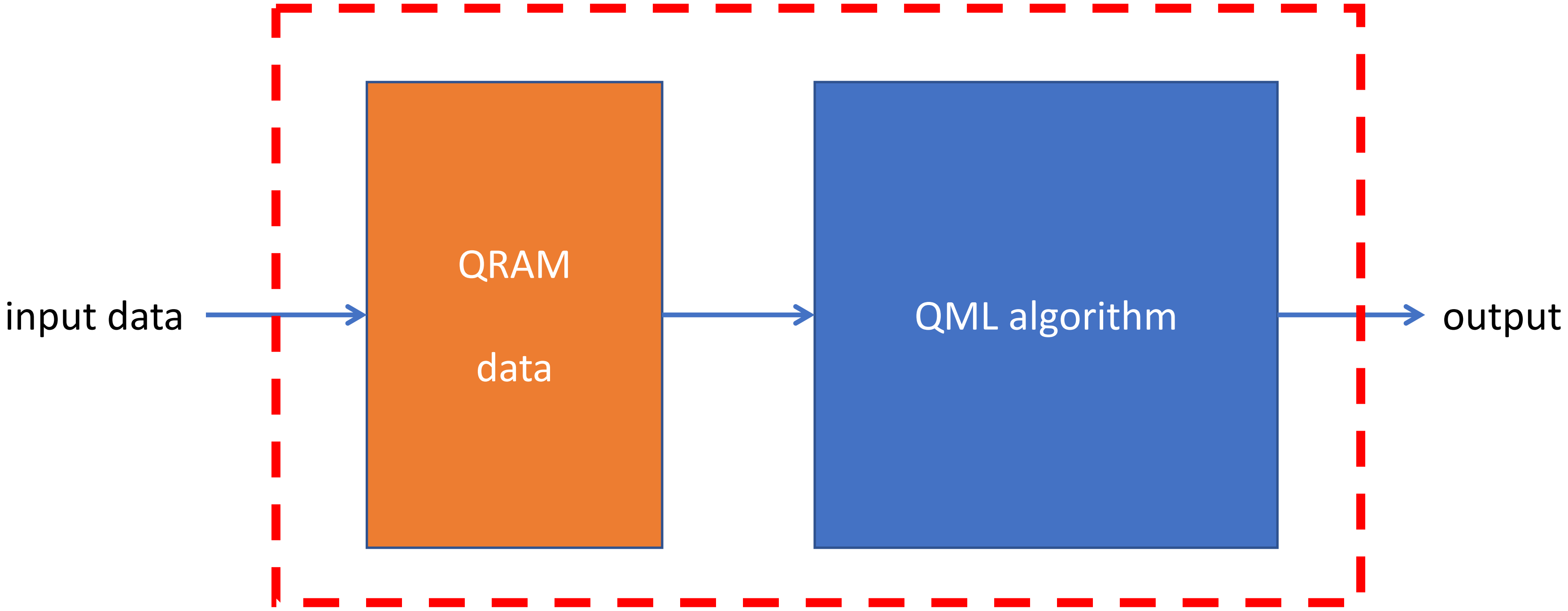
- QRAM-based algorithms [HHL'09, etc…]




Limitations: no provable end-to-end exponential speed-up [Aaronson'15]

# Quantum machine learning algorithms

- QRAM-based algorithms [HHL'09, etc...]




Limitations: dequantization argument [Tang'18]

# Quantum machine learning algorithms

- QRAM-based algorithms [HHL'09, etc...]
  - Amplitude encoding: $n$ dimensional vector stored in $\log n$ qubits
  - Pros: $\mathrm{polylog}(n)$ running time
  - Cons: hard to implement, not end-to-end, dequantization

- Heuristic QML algorithms
  - QNN, QGAN, kernel methods, etc...
  - Works on classical data
  - Pros: can be implemented on near-term hardware
  - Cons: lack of evidence for quantum advantage

# Results

**Quantum kernel methods**

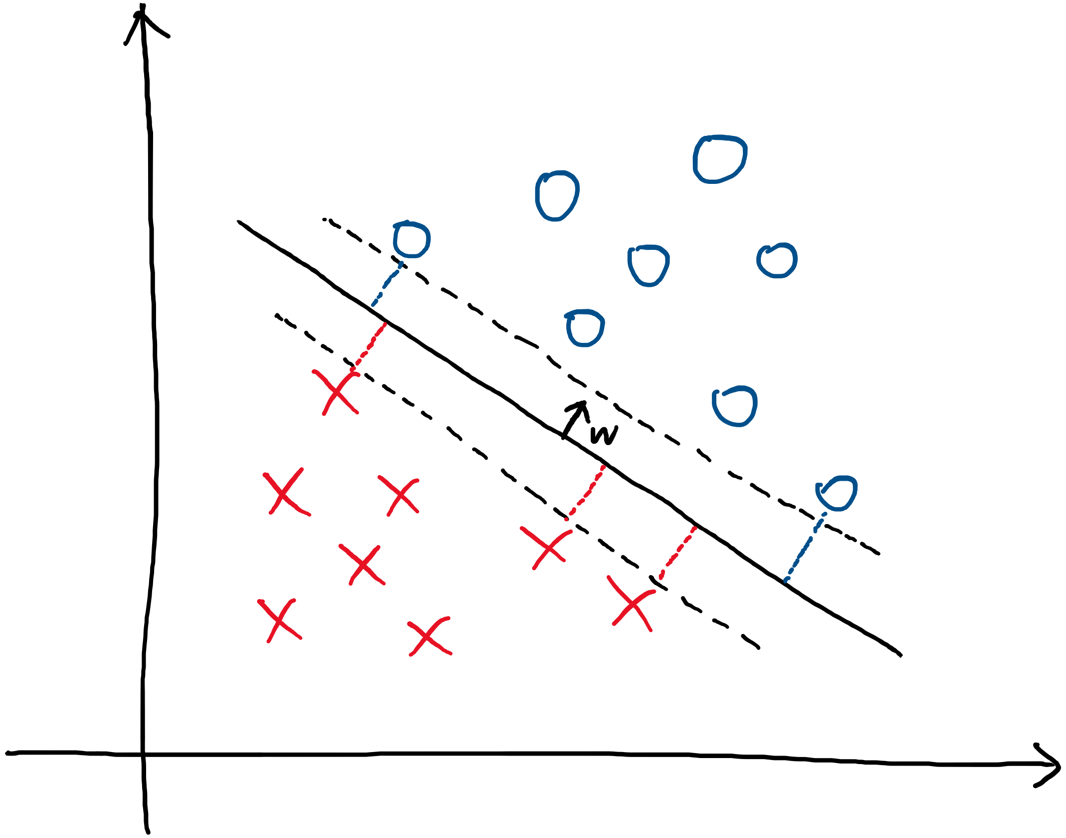
**[Havlíček et al’19, Schuld et al’19]**

- Classical data vectors are mapped to quantum states via a quantum feature map
- A linear classifier in Hilbert space can be efficiently obtained via the kernel method

**Our results**

- We show this algorithm can provably solve a classification problem, and this problem is hard for all classical algorithms
- Evidence of end-to-end quantum speed-up

# Support vector machines and kernel methods



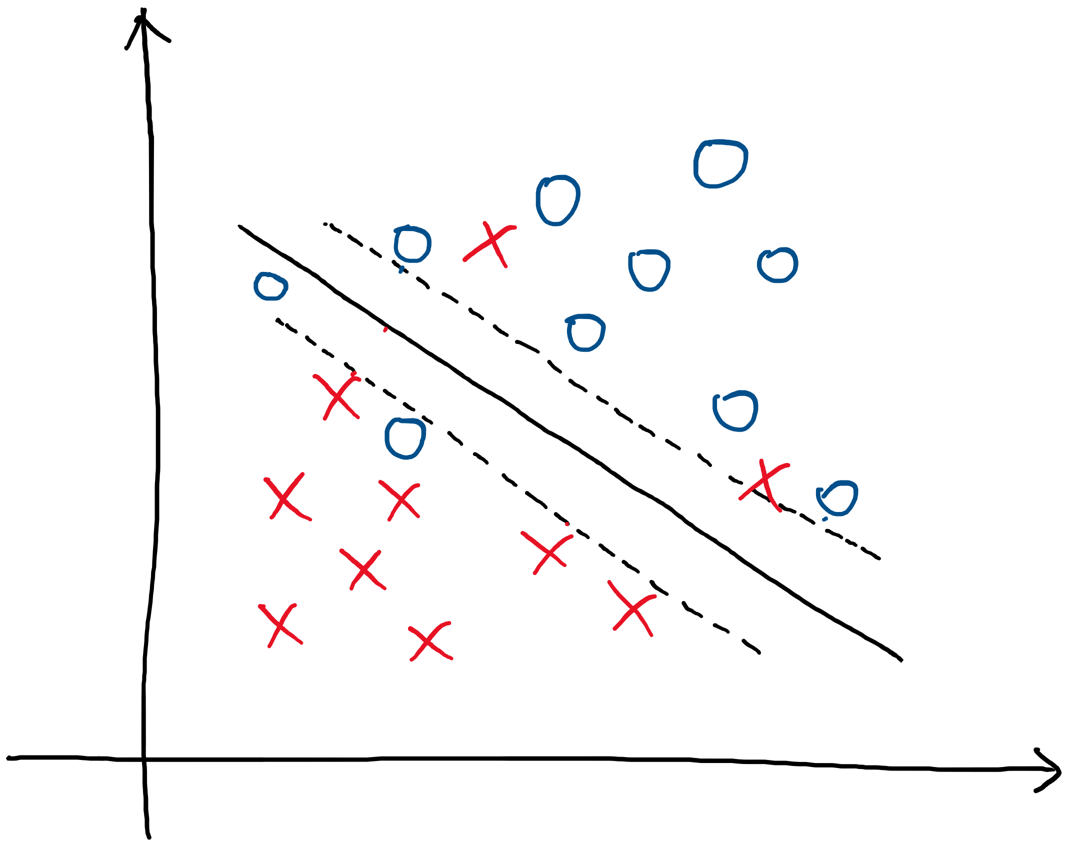

training data $S \subseteq \mathbb{R}^d \times \{\pm 1\}$

$$\begin{cases} \min \ \frac{1}{2}\|w\|_2^2 \\ \text{st} \quad y_i(\langle x_i, w\rangle + b) \geq 1, \quad \forall (x_i, y_i) \in S \end{cases}$$

$$\text{margin}(x, y) = \frac{1}{\|w\|_2} y(\langle w, x\rangle + b)$$

$$\rightarrow \max \min_{(x,y)\in S} \text{margin}(x, y)$$

# Support vector machines and kernel methods



training data $S \subseteq \mathbb{R}^d \times \{\pm 1\}$

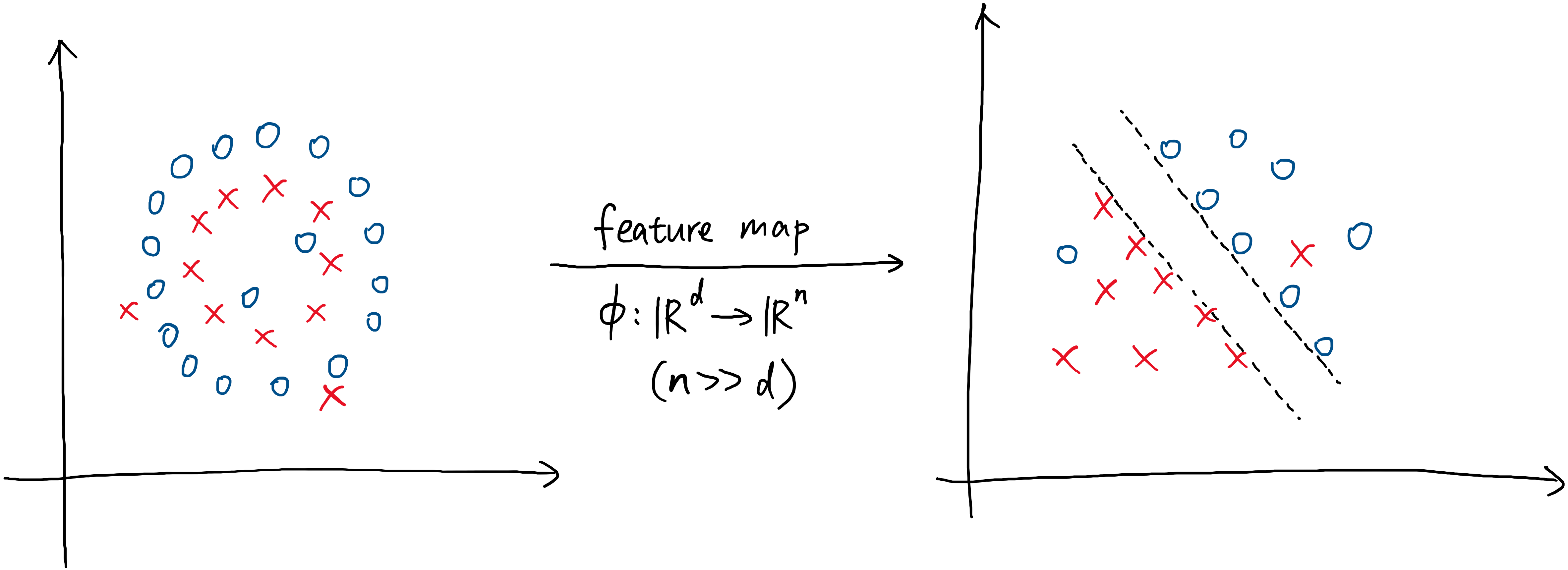
$$\min \quad \frac{1}{2}\|w\|_2^2 + \frac{\lambda}{2}\sum_i \xi_i^2$$

$$\text{st} \quad y_i(\langle x_i, w\rangle + b) \geq 1 - \xi_i$$

$$\xi_i \geq 0$$

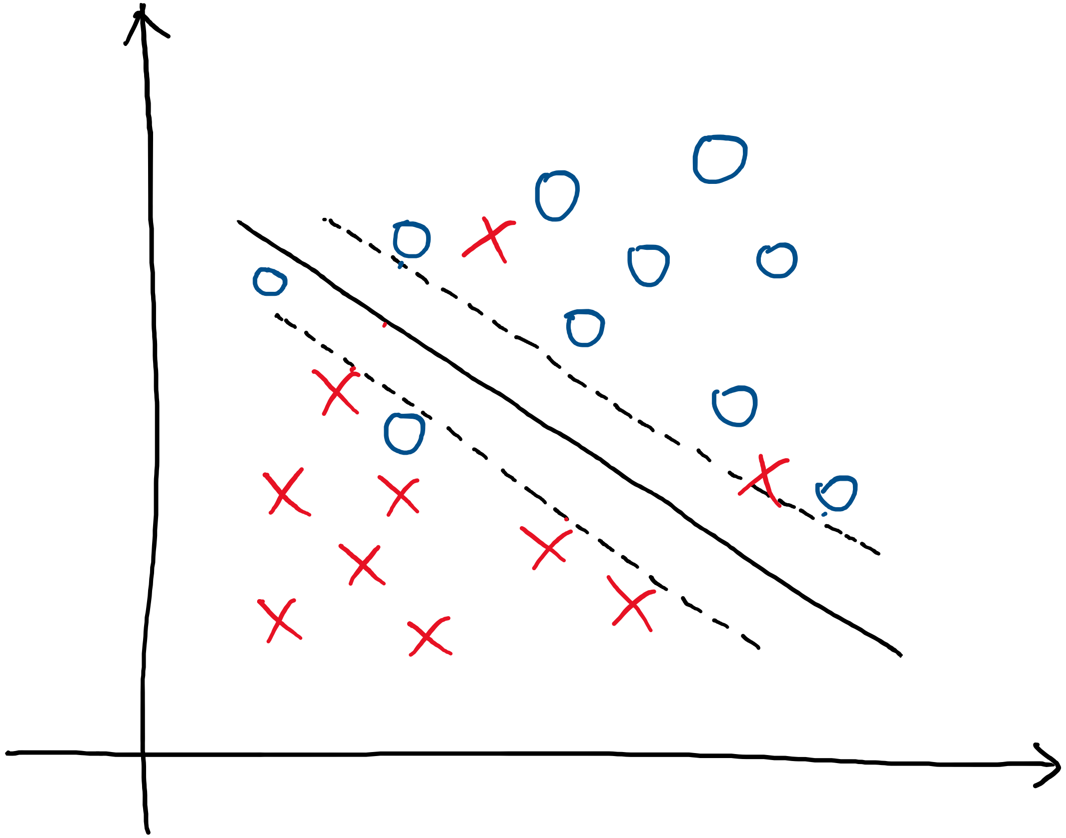
$$\text{margin}(x, y) = \frac{1}{\|w\|_2} y(\langle w, x\rangle + b)$$

# Support vector machines and kernel methods




Problem: how to do optimization in high dimensional feature space?

# Support vector machines and kernel methods



$$\min_{w,\xi} \; \frac{1}{2}\|w\|_2^2 + \frac{\lambda}{2}\sum_i \xi_i^2$$

$$\text{st} \quad y_i\left(\langle \phi(x), w \rangle\right) \geq 1 - \xi_i$$

$$\xi_i \geq 0$$

↕ duality

$$\max_{\alpha \geq 0} \; \sum_i \alpha_i - \frac{1}{2\lambda}\sum_i \alpha_i^2 - \frac{1}{4}\sum_{i,j} \alpha_i \alpha_j y_i y_j \left(\langle \phi(x_i), \phi(x_j) \rangle + 1\right)$$

↑ $K(x_i, x_j)$

# Support vector machines and kernel methods

training

$$\min_{w,\xi} \ \frac{1}{2}\|w\|_2^2 + \frac{\lambda}{2}\sum_i \xi_i^2$$

$$\text{st} \quad y_i\left(\langle \phi(x), w\rangle\right) \geq 1 - \xi_i$$

$$\xi_i \geq 0$$

↕ duality

$$\max_{\alpha \geq 0} \ \sum_i \alpha_i - \frac{1}{2\lambda}\sum_i \alpha_i^2$$

$$-\frac{1}{4}\sum_{i,j} \alpha_i\alpha_j y_i y_j \left(\langle \phi(x_i), \phi(x_j)\rangle + 1\right)$$

$K(x_i, x_j)$

testing

$$y_{pred} = \text{sign}\left(\langle w, \phi(x)\rangle\right)$$

↕ duality

$$y_{pred} = \text{sign}\left(\sum_i \alpha_i y_i K(x, x_i)\right)$$

# Support vector machines and kernel methods

- Kernel method: do not specify feature map explicitly; instead, define efficiently computable *kernel function*
  - $K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle$

eg polynomial kernel: $K(x_i, x_j) = (\langle x_i, x_j \rangle + 1)^p$ $\quad \dim \phi \approx \binom{d+p}{p}$

radial basis function (RBF): $K(x_i, x_j) = \exp\left(-\gamma \| x_i - x_j \|_2^2\right)$ $\quad \dim \phi = \infty$

# Quantum kernel methods

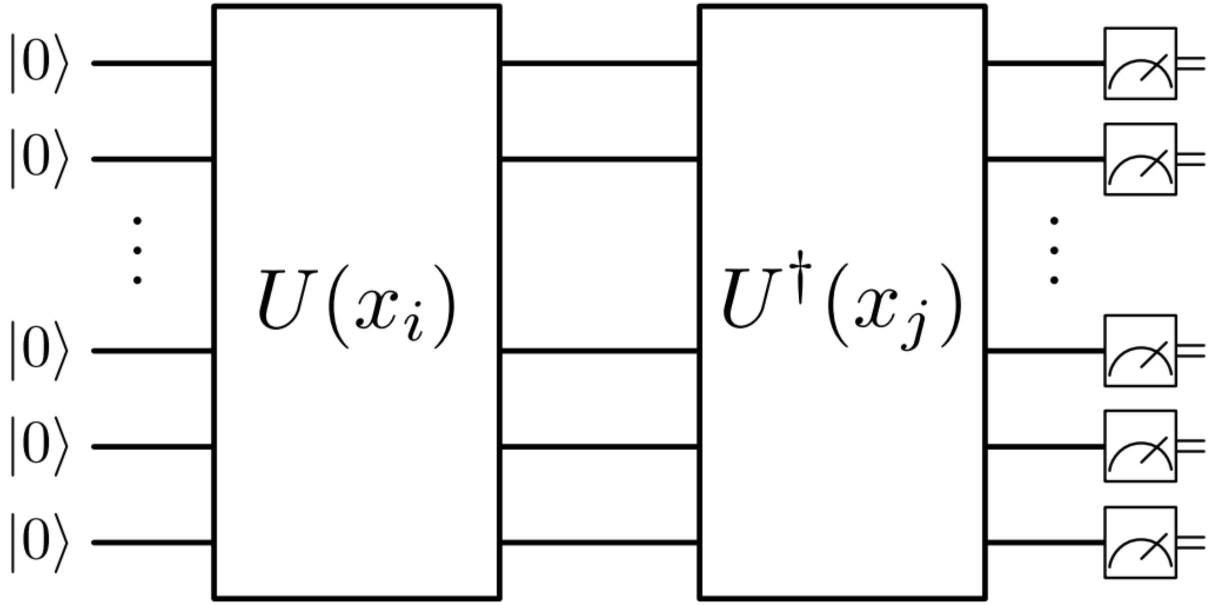
- Kernel method: do not specify feature map explicitly; instead, define efficiently computable *kernel function*
  - $K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle$
- Quantum feature map:

$$x \xrightarrow{\phi} U(x)|0\rangle\langle 0|U(x)^\dagger$$

$$K(x_i, x_j) = \left| \langle 0| U^\dagger(x_j) U(x_i) |0\rangle \right|^2$$

# Quantum kernel methods

$$K(x_i, x_j) = \left| \langle 0 | U^\dagger(x_j) U(x_i) | 0 \rangle \right|^2 \approx$$



# Classical and quantum kernel methods

**Classical SVMs**

- On input a classical training set
- Compute the kernel function for each pair of training data
- Run dual program, obtain classifier
- Compute the kernel function for new data during testing

**Quantum kernel estimation (SVM-QKE)**

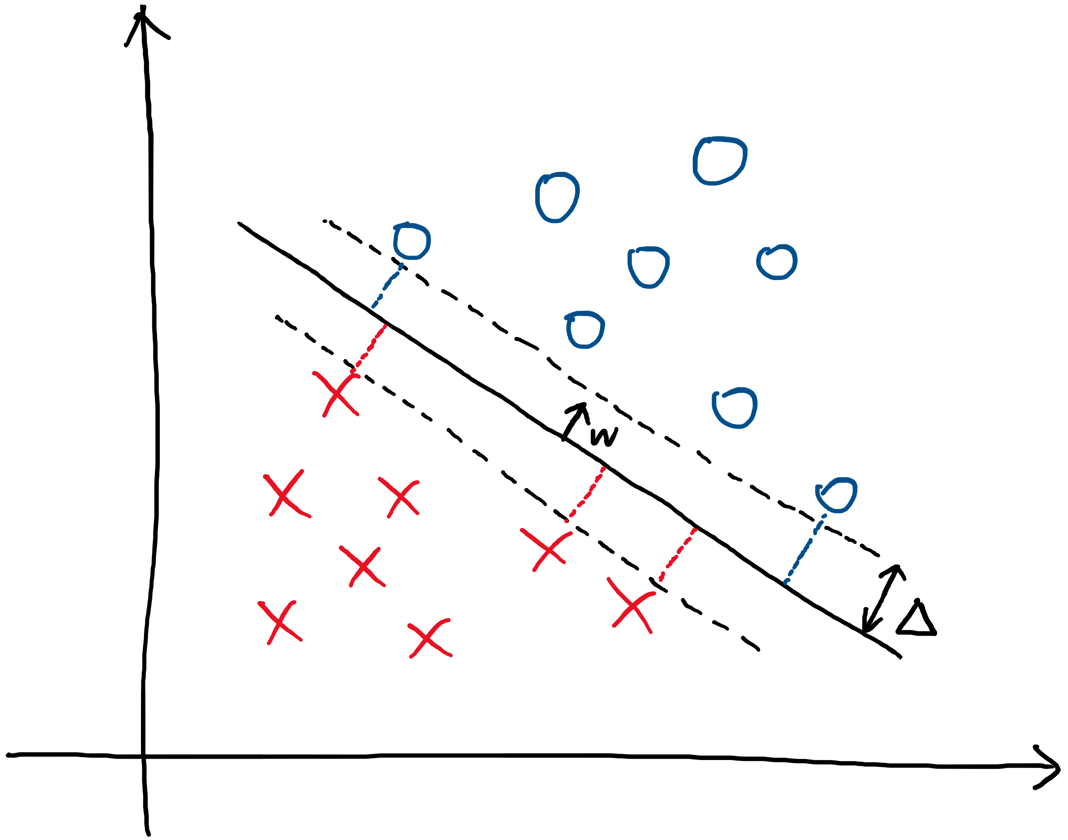
- On input a classical training set
- Estimate the quantum kernel function for each pair of training data
- Run dual program, obtain classifier
- Estimate the kernel function for new data during testing

- Expressivity: quantum feature maps are more expressive than classical feature maps
- Finite sampling noise: quantum kernel estimation has 1/poly sampling noise, even with error corrected quantum computer

# Result: end-to-end quantum speed-up

- Step 1: construct a learning problem that is hard for classical algorithms
  - Sanity check: this problem should be in BQP
  - We construct a classification problem that is as hard as discrete log
- Step 2: solve this problem using quantum kernel estimation
  - Robust to finite sampling noise
- The learning problem itself is not important, the purpose is to show that SVM-QKE is powerful in general

# How to prove that kernel methods work?




$$\min \quad \frac{1}{2}\|w\|_2^2$$

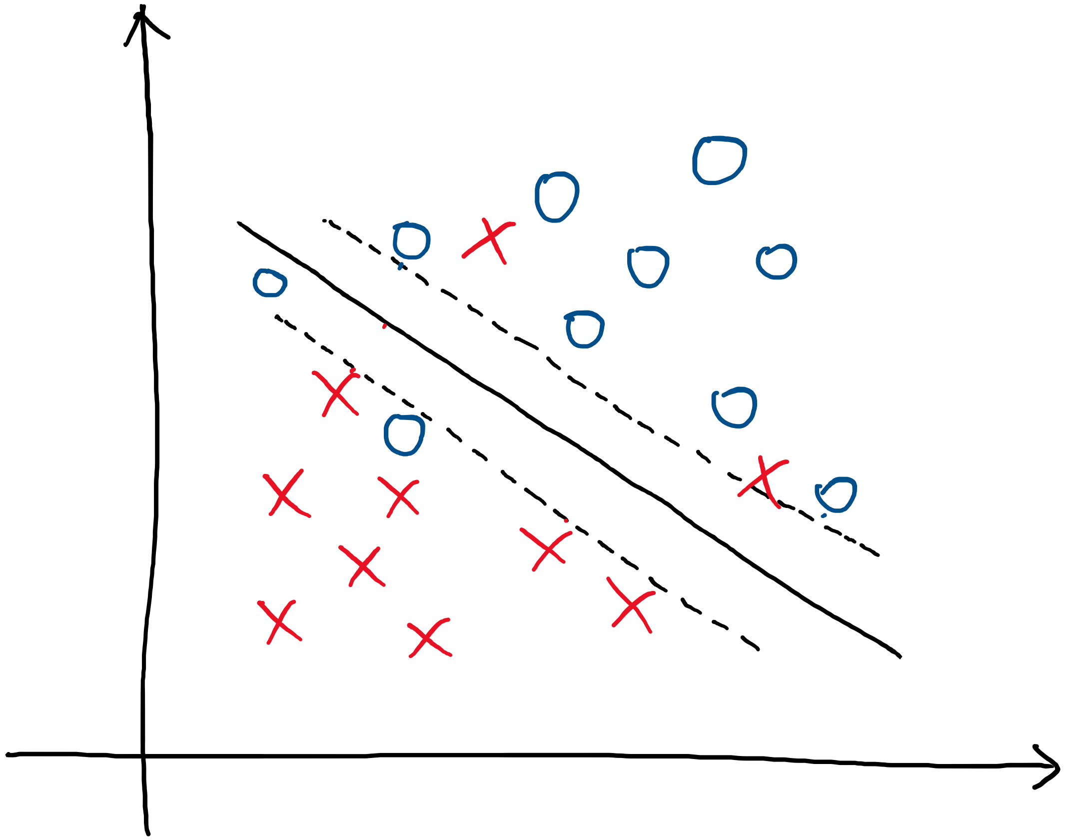
$$\text{st} \quad y_i \langle x_i, w \rangle \geq 1, \quad \forall (x_i, y_i) \in S$$

margin-based generalization bound:

$$\text{test error} \leq O\left(\frac{1}{m\Delta^2}\right)$$

$$= O\left(\frac{\|w\|_2^2}{m}\right)$$

# How to prove that kernel methods work?



$$\min \quad \frac{1}{2}\|w\|_2^2 + \frac{\lambda}{2}\sum_i \xi_i^2$$

$$\text{st} \quad y_i(\langle \phi(x), w\rangle) \geq 1 - \xi_i$$

$$\xi_i \geq 0$$

map to higher dimension and reduce to hard-margin case

$$\text{test error} \leq \frac{1}{m} O(\|w\|_2^2 + \|\xi\|_2^2)$$

Key point: large margin observed in training implies good performance in testing

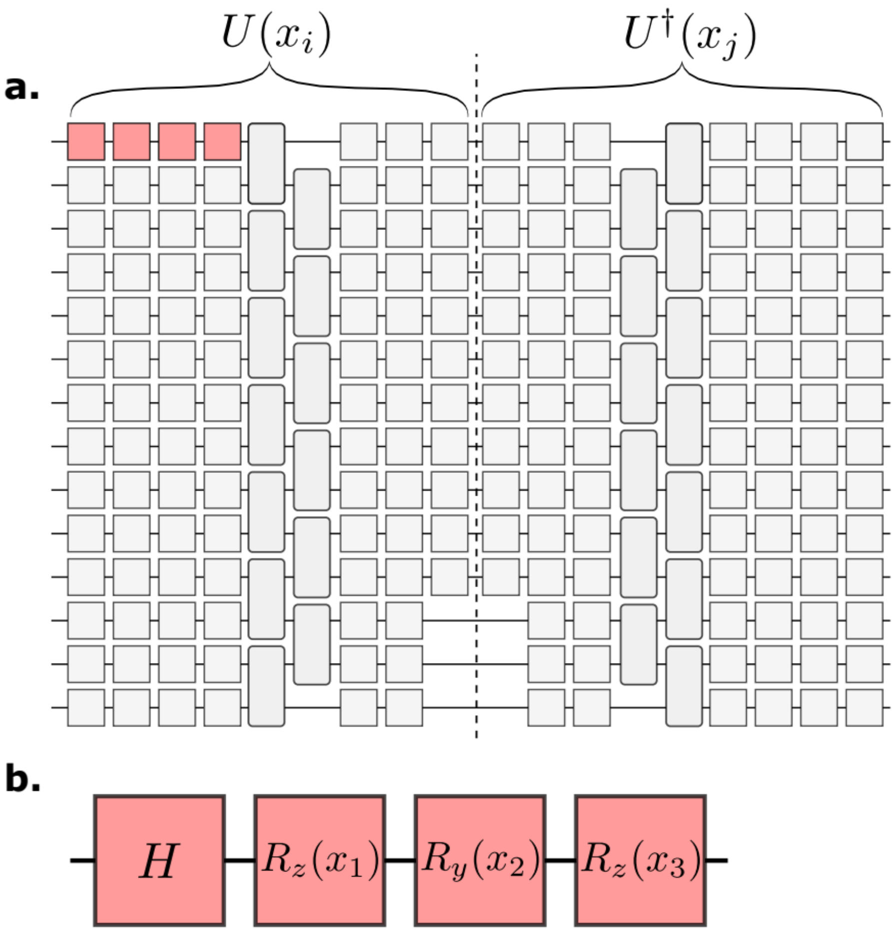
# Proof overview

- We explicitly construct a quantum feature map (kernel function) such that training data is separated by a large margin
- Running the dual program with quantum kernel, we are guaranteed to find a good hyperplane
- Use margin-based generalization bound
- Noise robustness can be obtained by strong convexity
  - Small perturbation in the kernel will only cause small perturbation in the classifier

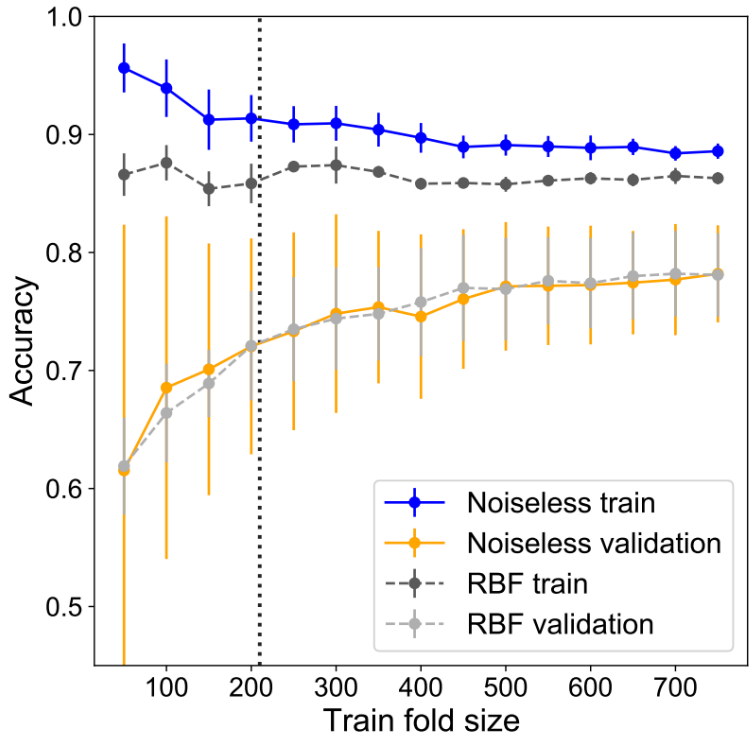
# Prospects and obstacles of quantum advantage with QKE

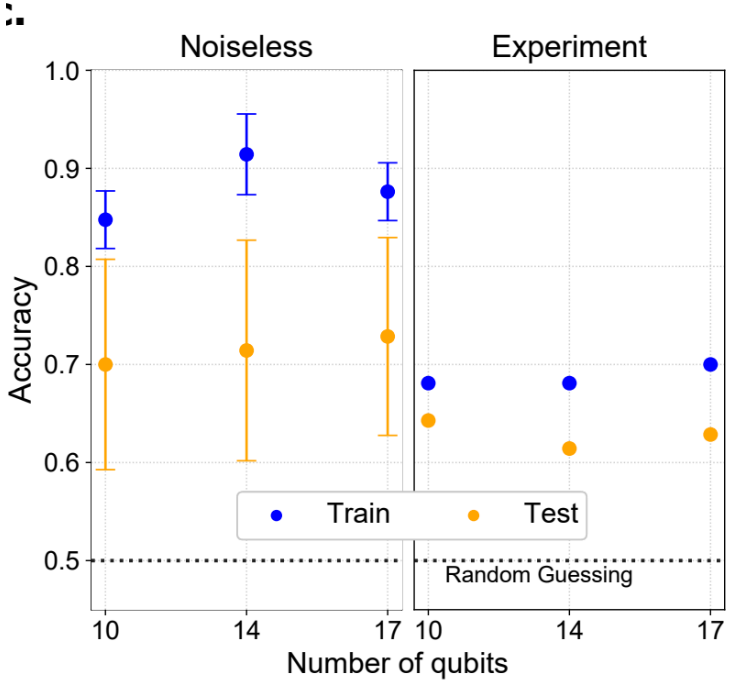
- Future directions:
  - Improve our result to BQP-complete
  - Find practical learning problems that are challenging for classical algorithms
  - Develop “universal” quantum kernels
  - Develop error mitigation techniques suitable for QKE
- Obstacles:
  - Constant depth 2D circuits do not have asymptotic advantage [BGM’19]
  - Already have very powerful classical general-purpose learning algorithms

# Recent experiment [Peters et al’21]




- 17 qubits on Sycamore
- 67 dimensional cosmological dataset
- Binary classification




Peters et al, Machine learning of high dimensional data on a noisy quantum processor, arxiv: 2101.09581

# References


- [Arute et al’19] Quantum supremacy using a programmable superconducting processor, Nature 2019
- [Zhong et al’20] Quantum computational advantage using photons, Science 2020
- [Arute et al’20] Hartree-Fock on a superconducting qubit quantum computer, Science 2020
- [Harrigan et al’21] Quantum approximate optimization of non-planar graph problems on a planar superconducting processor, Nature Physics 2021
- [Peters et al’21] Machine learning of high dimensional data on a noisy quantum processor, arxiv: 2101.09581
- [HHL’09] Quantum Algorithm for Linear Systems of Equations, Physical Review Letters 2009
- [Aaronson’15] Read the fine print, Nature Physics 2015
- [Tang’18] A quantum-inspired classical algorithm for recommendation systems, STOC 2019
- [Havlíček et al’19] Supervised learning with quantum enhanced feature spaces, Nature 2019
- [Schuld et al’19] Quantum machine learning in feature Hilbert spaces, Physical Review Letters 2019
- [BGM’19] Classical algorithms for quantum mean values, Nature Physics 2021