Introduction to photonic quantum machine learning

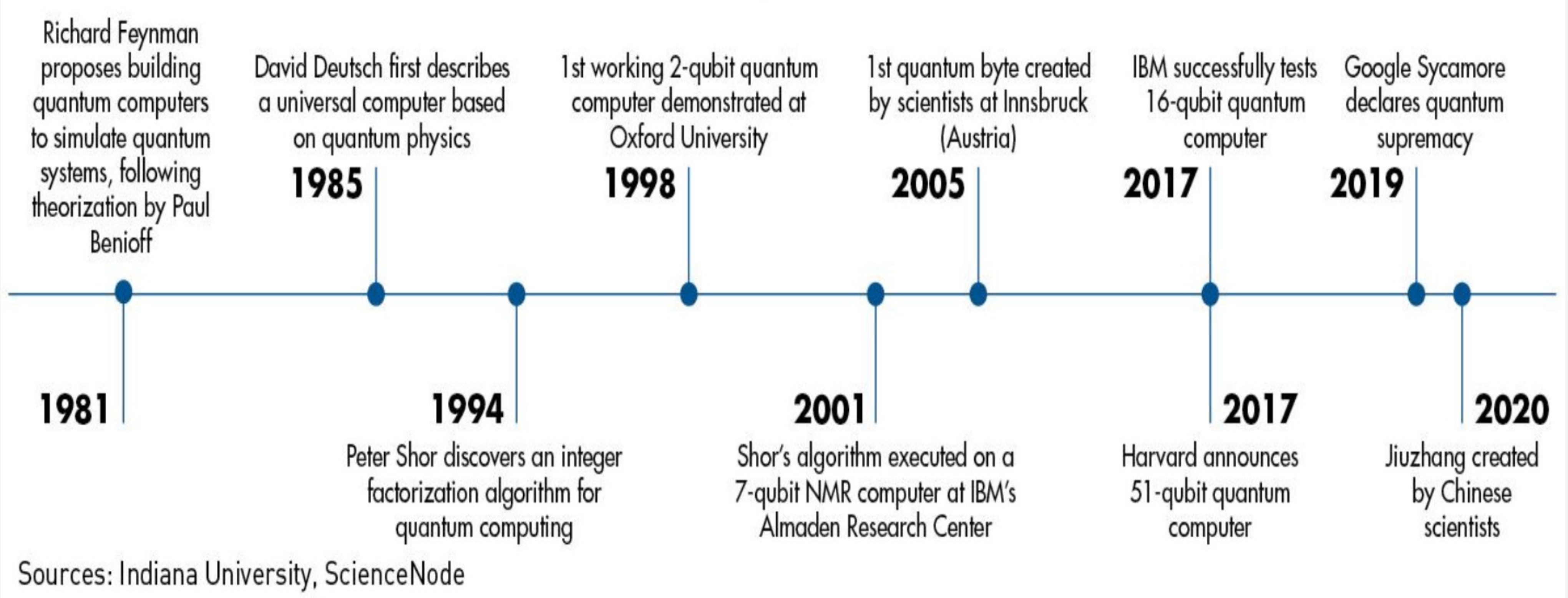
Dániel Nagy

Wigner GPU Day, 2021

Agenda

- Motivation
- Basic ML Concepts
- Quantum Computing with Photons
- Photonic Quantum Machine Learning
- Numerical Experiments

Motivation



- Google q-supremacy (2019) https://www.nature.com/articles/s41586-019-1666-5
- Photonic device by USTC (Jiuzhang 2.0) (2021) https://journals.aps.org/prl/abstract/10.1103/PhysRevLett.127.180502

Motivation

- **Today: Noisy Intermediate-scale devices (NISQ)**
  - <1000 qubits
  - Short coherence-time
  - Cross-talk
  - Gate & readout noise
  - High photon loss rate (above 50%)
  - Dark counts

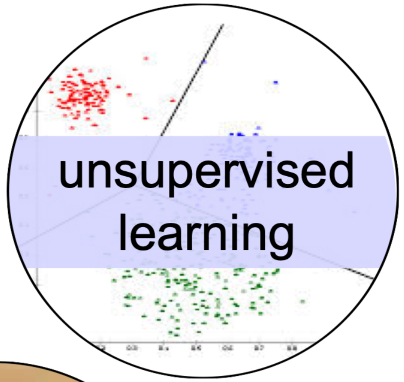
Motivation

- **NISQ-era candidates for practical quantum advantage:**
  - Simulation of quantum chemistry and many-body systems
  - Variational quantum optimization methods like QAOA
  - **Quantum Machine Learning**

Machine Learning

- Instead of implementing rules, let the computer learn the rules.

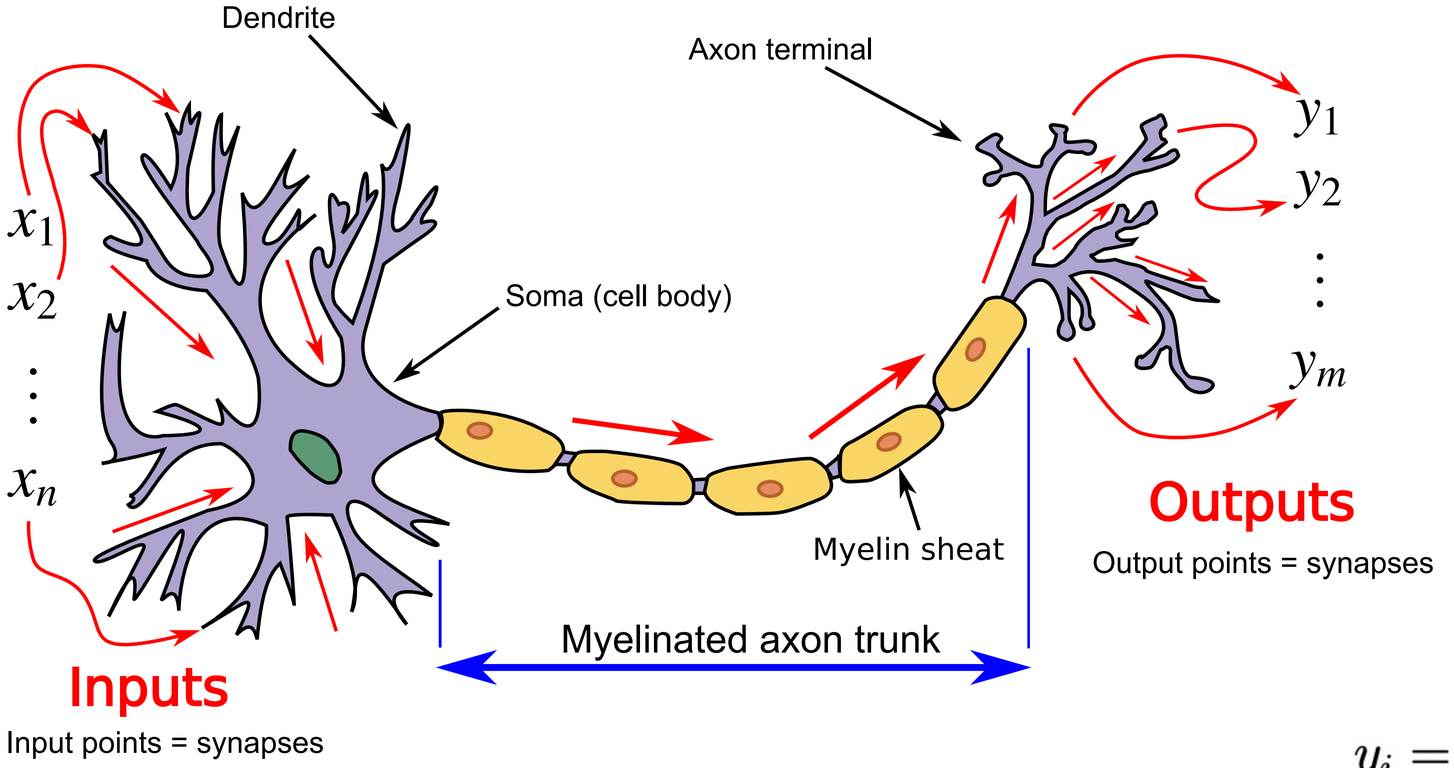
Learn patterns in
labeled data
e.g. cats vs dogs



Learn patterns in
unlabeled data
e.g. clustering

Learn to solve a control
problem or to navigate in an
environment
e.g. Atari games

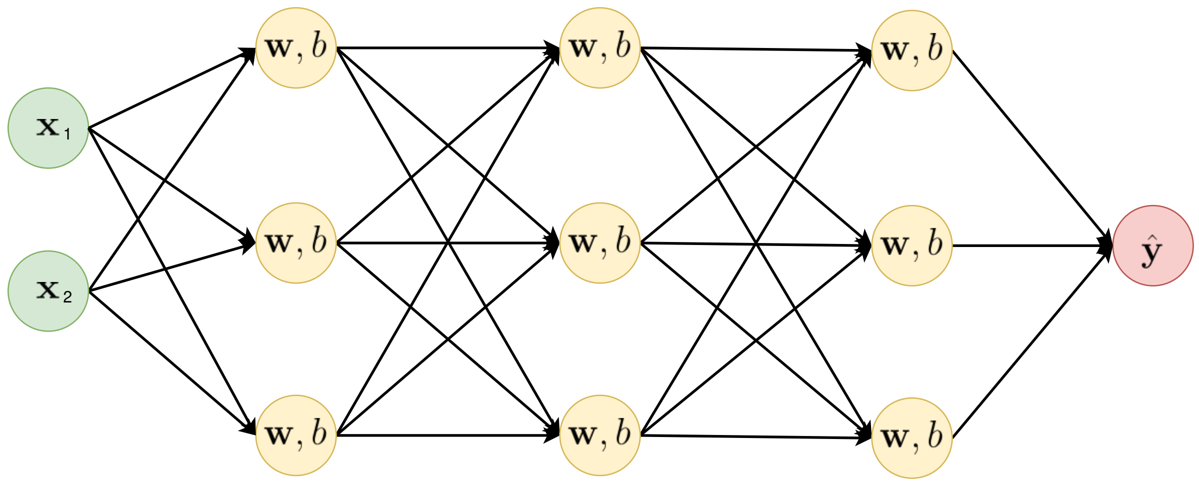
Machine Learning – Neural networks



- Our brain is good at solving problems. Idea: try to model the brain and simulate it.
- Input dendrides receive voltage spikes with frequencies x_j
- Output synapses fire with frequencies y_i
- U_0 is the threshold potential
- θ is the step funcition

$$y_i = \theta\left(\sum_j J_{ij} x_j - U_0\right)$$

Machine Learning – Neural networks

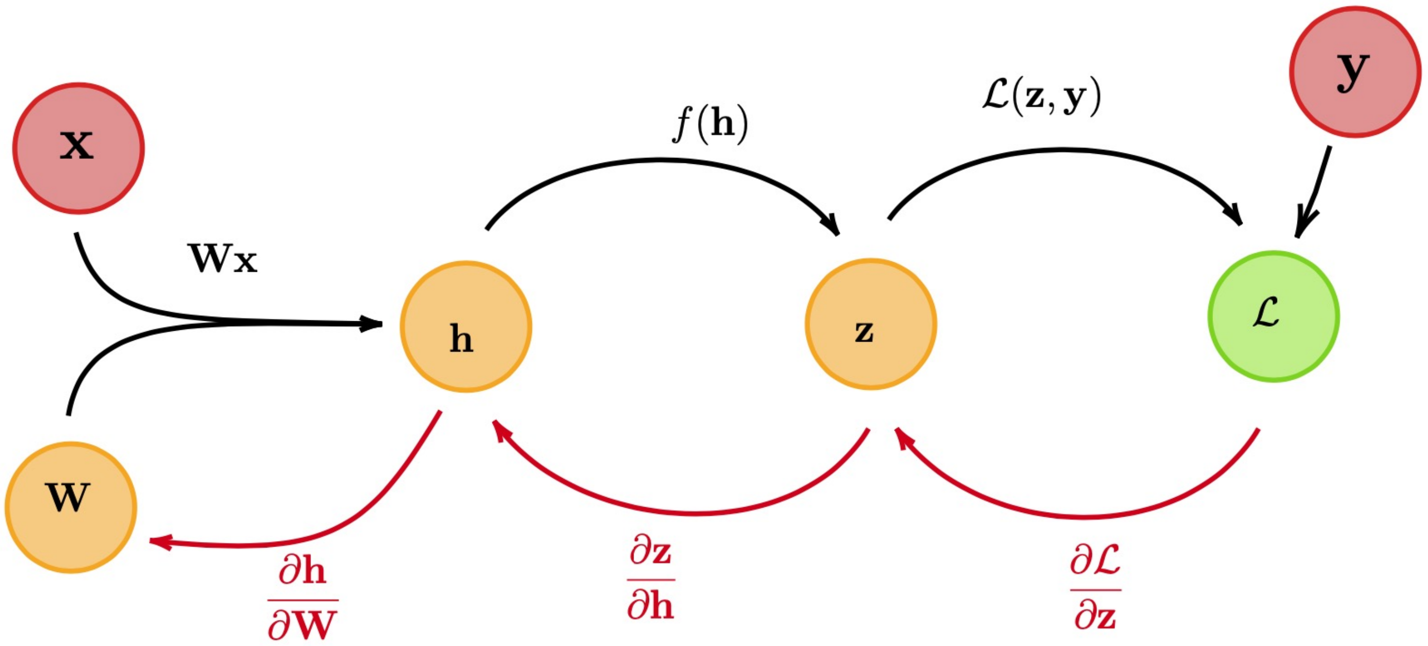


$$\mathbf{y} = g\left(\mathbf{W}^\top \mathbf{x} + \mathbf{b}\right)$$

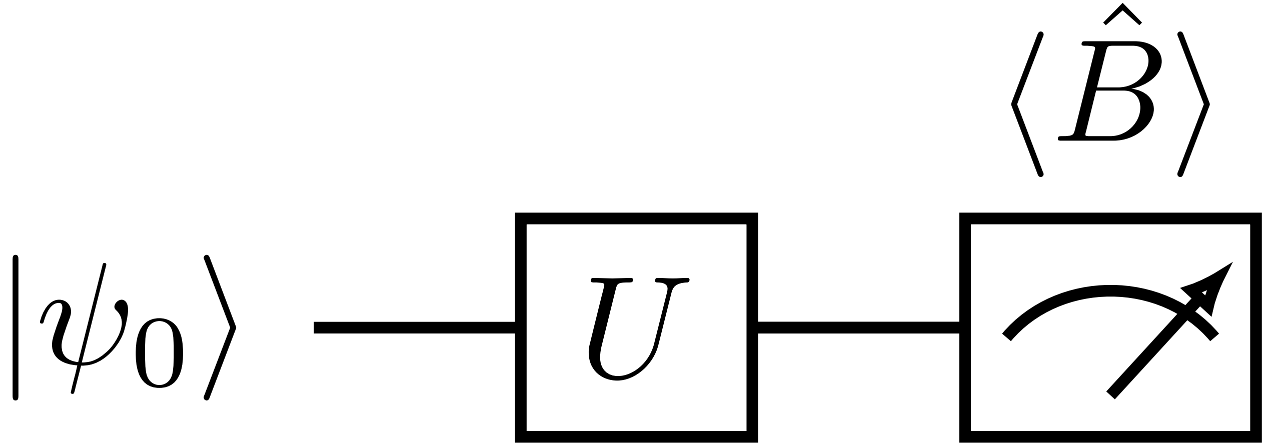
- Goal: find W and b that minimize the error.
- Neural Networks are universal function approximators

How to train Neural Networks?

- Define a loss function: $L = \frac{1}{N}\sum_{k=1}^{N} \mathcal{L}(\mathbf{y}_k, f(\mathbf{x}_k; \boldsymbol{\theta}^{(t)}))$
- Update parameters: $\theta_j^{(t+1)} \leftarrow \theta_j^{(t)} - \alpha_j^{(t)} \left. \frac{\partial L}{\partial \theta_j} \right|_{\theta_j = \theta_j^{(t)}}$



Quantum Computation – Circuit Model



Photonic Quantum Computation

Qubit paradigm

- Qubit basis: $\{|0\rangle, |1\rangle\}$
- $\dim(H) = 2^N$
- Can be realized by photons
- Common errors:
  - Bit-flip error
  - Phase-shift error
  - Gate errors
  - Measurement errors
- Cryogenic Temperature

Qumode paradigm

- Fock basis: $\{|0\rangle, |1\rangle, \ldots\}$
- $\dim(H) = \infty$
- $\dim(H) = D^N$
- Best for describing photons
- Common errors:
  - Photon loss
  - Gate errors
  - Measurement errors
- Room Temperature

Photonic Quantum Computation

Equivalent descriptions

Fock-space description ⟷ **Phase-space description**

$$\hat{a}_k = \frac{\hat{X}_k + i\hat{P}_k}{\sqrt{2}}$$

$$\hat{a}_k^\dagger = \frac{\hat{X}_k - i\hat{P}_k}{\sqrt{2}}$$

Quadrature operators →

$$\hat{X}_k = \frac{\hat{a}_k + \hat{a}_k^\dagger}{\sqrt{2}}$$

$$\hat{P}_k = \frac{\hat{a}_k - \hat{a}_k^\dagger}{i\sqrt{2}}$$

Defines an infinite dimensional Hilbert space

$$|n_1, \ldots, n_k, \ldots\rangle$$

Continuous basis →

$2N$-dimensional symplectic phase-space

$$|\mathbf{x}\rangle \qquad |\xi\rangle$$

Linear operators can have non-linear counterparts →

Phase-space description

- We work in the quadrature basis: $$\hat{X}_j = X_j \left|\mathbf{x}\right\rangle$$

- We can calculate the Wigner-function:

$$W_\rho(\mathbf{x}, \mathbf{p}) = \frac{1}{\pi^N} \int_{\mathbb{R}^N} \left\langle \mathbf{x} + \mathbf{y} \right| \rho \left| \mathbf{x} - \mathbf{y} \right\rangle e^{-2i\mathbf{p}\mathbf{y}} \mathrm{d}^N \mathbf{y}$$

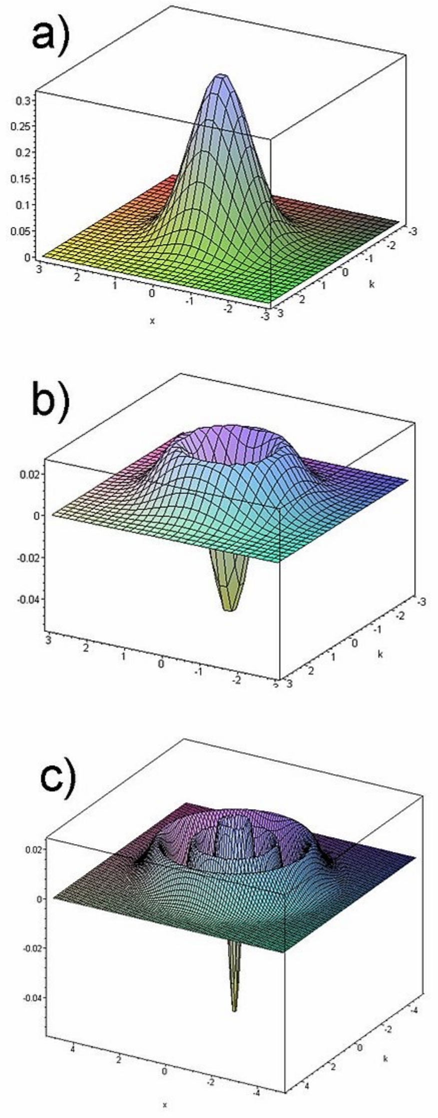
Phase-space description

- The Wigner function can be used to calculate expectation values:

$$\langle X_1 \rangle = \int_{\mathbb{R}^{2N-1}} W_\rho(\mathbf{x}, \mathbf{p}) \mathrm{d}p_1 \cdots \mathrm{d}p_N \mathrm{d}x_2 \cdots \mathrm{d}x_N$$

- And also the purity of the state:

$$\mu_\rho = \mathrm{Tr}\left[\rho^2\right] = (2\pi)^N \int_{\mathbb{R}^{2N}} \left(W_\rho(\mathbf{x}, \mathbf{p})\right)^2 \mathrm{d}^N\mathbf{x}\mathrm{d}^N\mathbf{p}$$



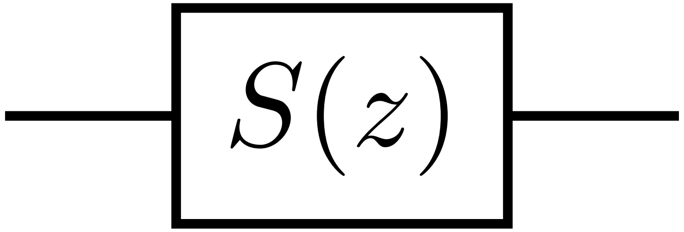
Source: Wikipedia

Gaussian states

- Quadrature vector: $$\mathbf{R}^\top = (\hat{X}_1, \hat{P}_1, \hat{X}_2, \hat{P}_2, ..., \hat{X}_k, \hat{P}_k, ..., \hat{X}_M, \hat{P}_M)$$
- First moment: $$d_j = \langle \hat{R}_j \rangle_\rho$$
- Second moment: $$\sigma_{ij} = \langle \hat{R}_i \hat{R}_j + \hat{R}_j \hat{R}_i \rangle_\rho - 2\langle \hat{R}_i \rangle_\rho \langle \hat{R}_j \rangle_\rho$$
- Gaussian state: $$W_\rho(\boldsymbol{\xi}) = \frac{1}{\pi^N \sqrt{\det \boldsymbol{\sigma}}} e^{-(\boldsymbol{\xi}-\mathbf{d})^\top \boldsymbol{\sigma}^{-1} (\boldsymbol{\xi}-\mathbf{d})}$$
- Vacuum state: $$W_{|0\rangle}(X, P) = \frac{1}{\pi} e^{-X^2 - P^2}$$

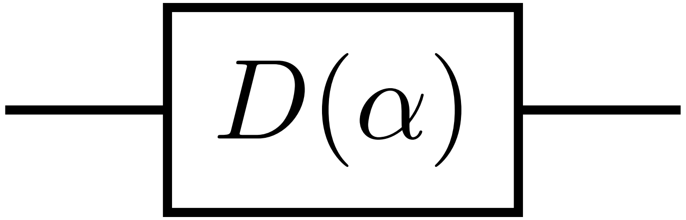
Universal Photonic Quantum Gate Set

Squeezing



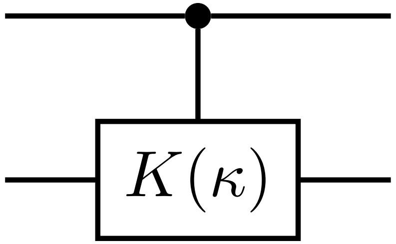
$$\hat{S}(z) = \exp\left[\frac{1}{2}\left(z^*\hat{a}^2 - z\hat{a}^{\dagger 2}\right)\right]$$

Displacement



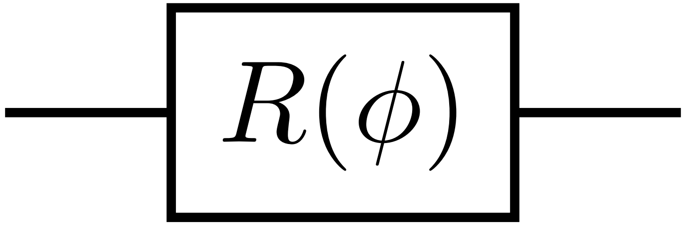
$$\hat{D}(\alpha) = e^{\alpha\hat{a}^\dagger - \alpha^*\hat{a}}$$

Cross-Kerr



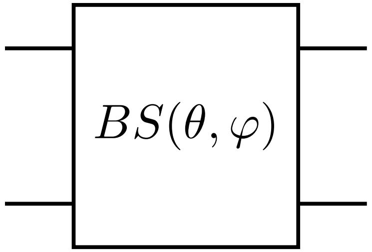
$$\hat{CK}(\kappa) = e^{i\kappa\hat{n}_j\hat{n}_k}$$

Rotation



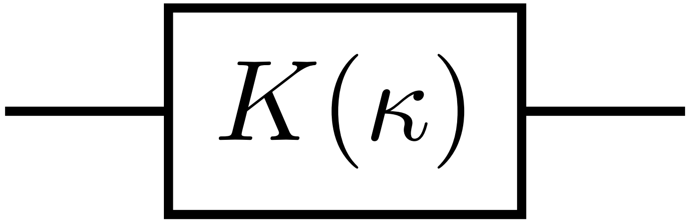
$$\hat{R}(\phi) = e^{i\phi\hat{a}^\dagger\hat{a}}$$

Beam-splitter



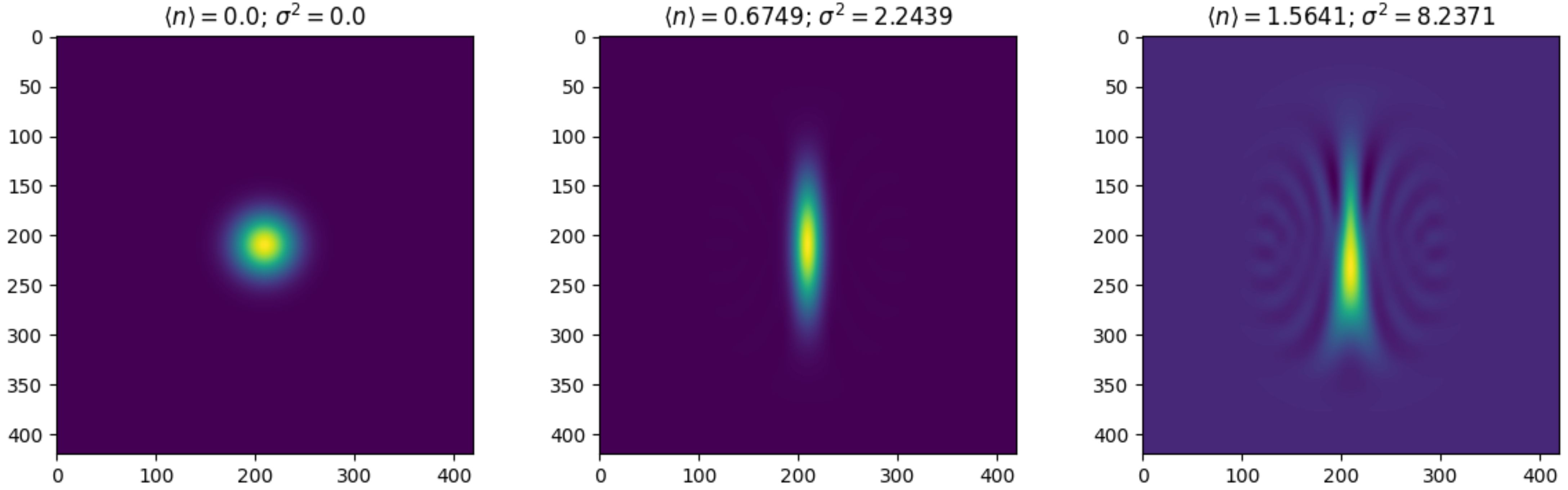
$$\hat{BS}(\theta,\varphi) = \exp\left[\theta\left(e^{i\varphi\hat{a}_j\hat{a}_k^\dagger} - e^{-i\varphi\hat{a}_j^\dagger\hat{a}_k}\right)\right]$$

Kerr



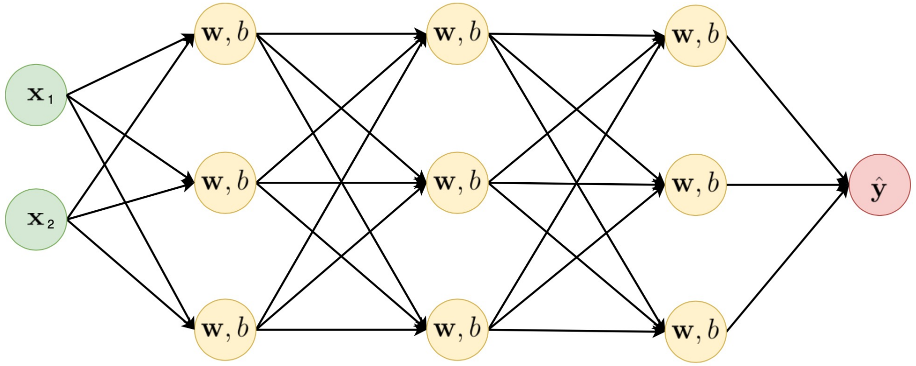
$$\hat{K}(\kappa) = e^{i\kappa\hat{n}^2}$$

Gaussian & Non-Gaussian States



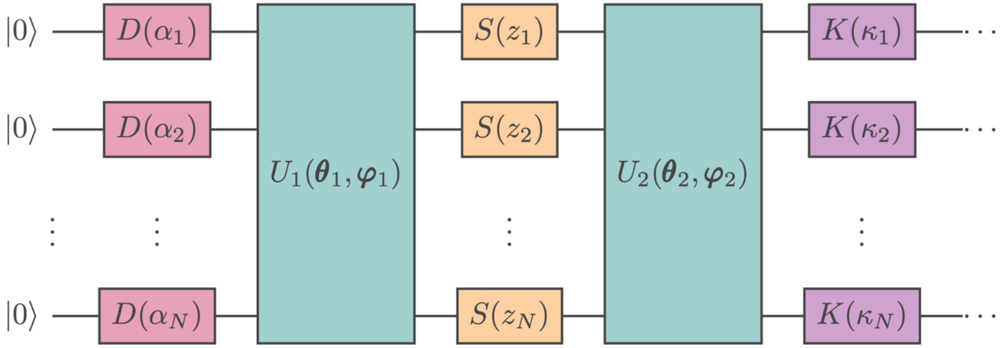
Cubic phase: $V(\gamma) = e^{i\gamma x^3/(3\hbar)}$

Continuous-Variable QNNs



$$\mathcal{L}(\mathbf{x}) = g(\mathbf{W}^\top \mathbf{x} + \mathbf{b})$$

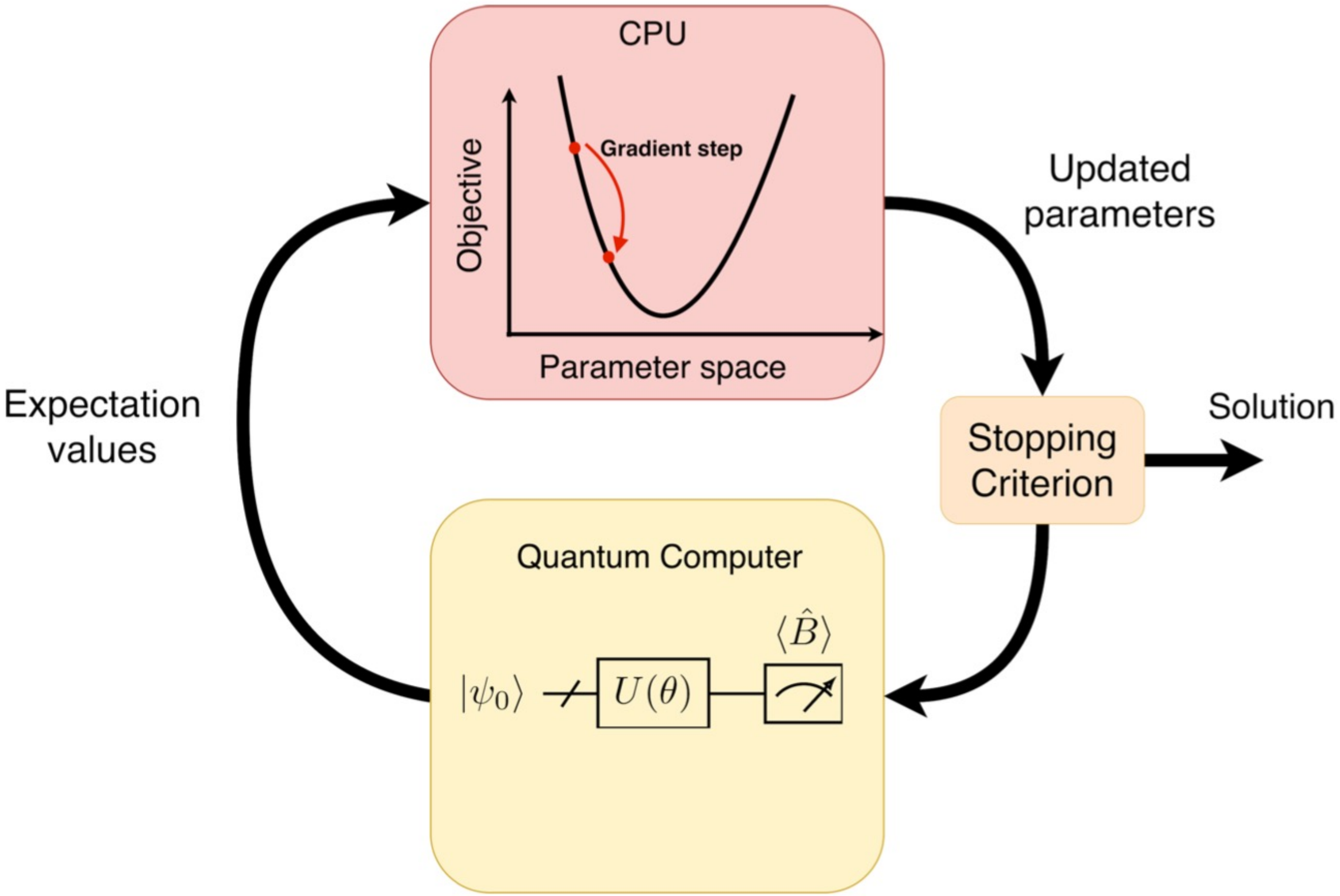
Universal function approximator



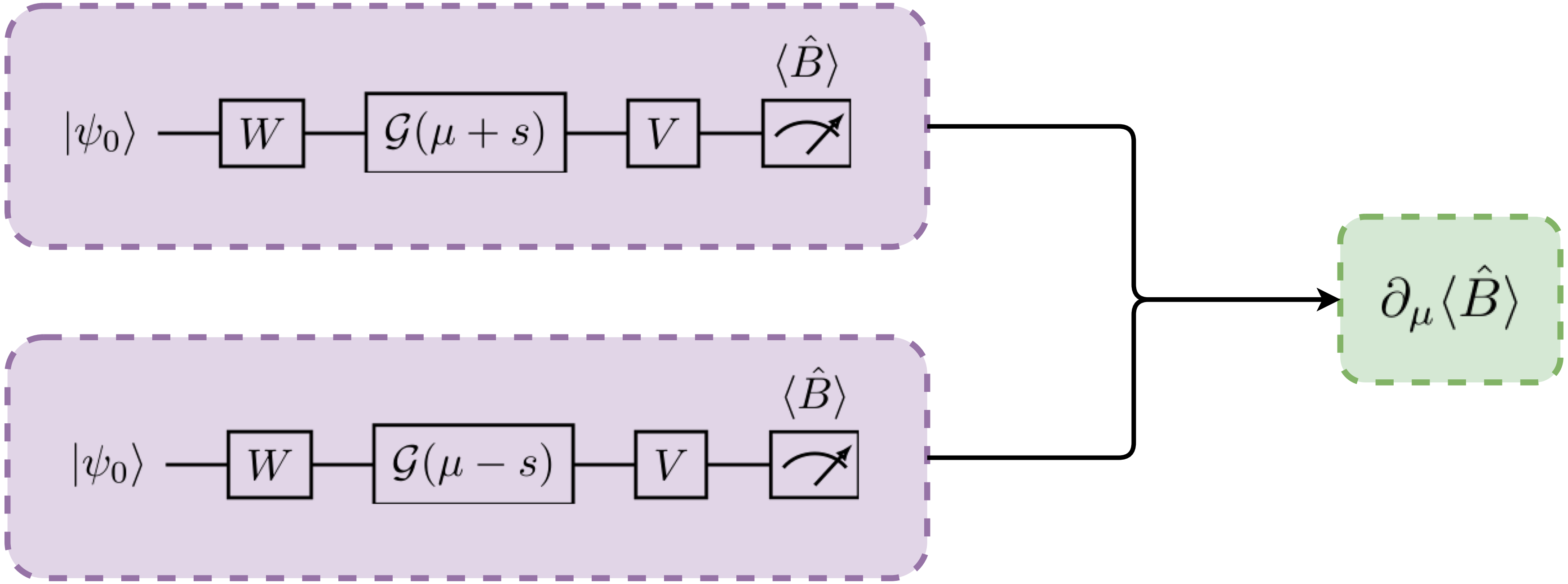
$$\mathcal{L}(\boldsymbol{\xi}) = K(\mathbf{M}\boldsymbol{\xi} + \boldsymbol{\alpha})$$

Universal unitary approximator

Hybrid Training Loop



Training Quantum Neural Networks



Evaluating analytic gradients on quantum hardware

Maria Schuld,* Ville Bergholm, Christian Gogolin, Josh Izaac, and Nathan Killoran
Xanadu Inc., 372 Richmond St W, Toronto, Canada M5V 1X6

(Received 9 January 2019; published 21 March 2019)

Numerical experiments

- Classification
- Regression
- VQE
- Reinforcement Learning

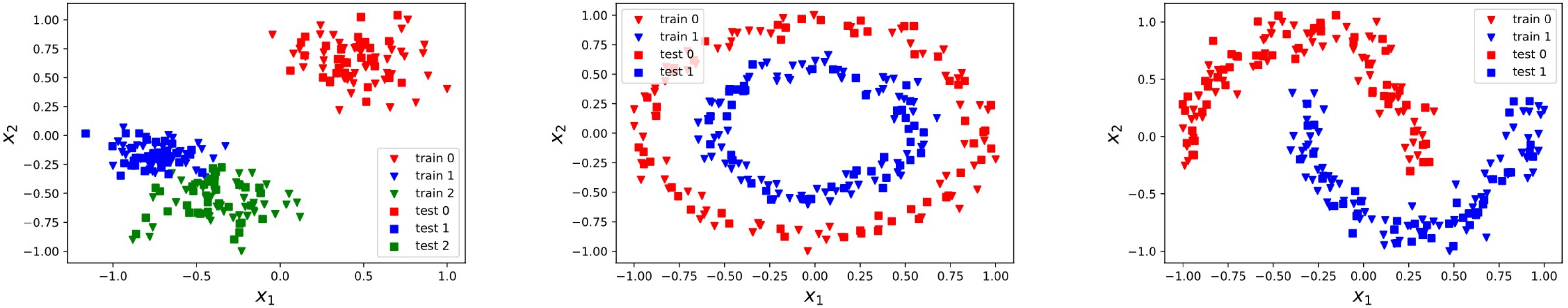
Strawberry Fields:
A Software Platform for Photonic Quantum Computing

Nathan Killoran, Josh Izaac, Nicolás Quesada, Ville Bergholm, Matthew Amy, and Christian Weedbrook

Xanadu, 372 Richmond St W, Toronto, M5V 1X6, Canada

TensorFlow

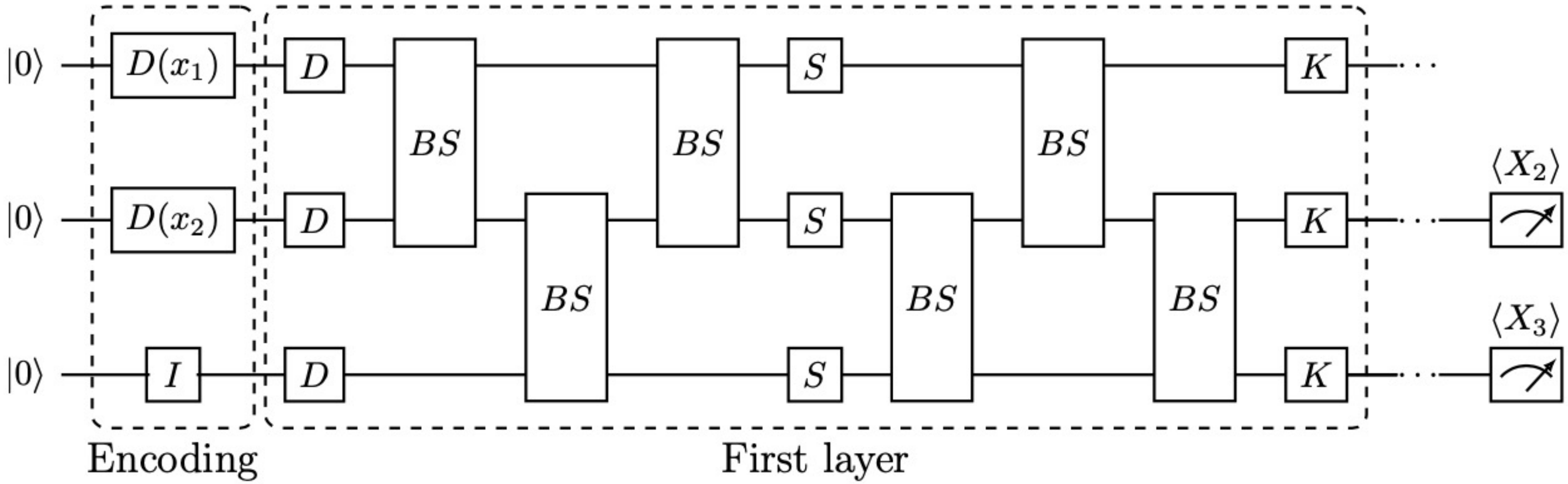
Classification



Encoding spacial data: $$(x_1, x_2) \mapsto \hat{D}_1(x_1) \otimes \hat{D}_2(x_2) \otimes I_3 \left|0, 0, 0\right\rangle$$

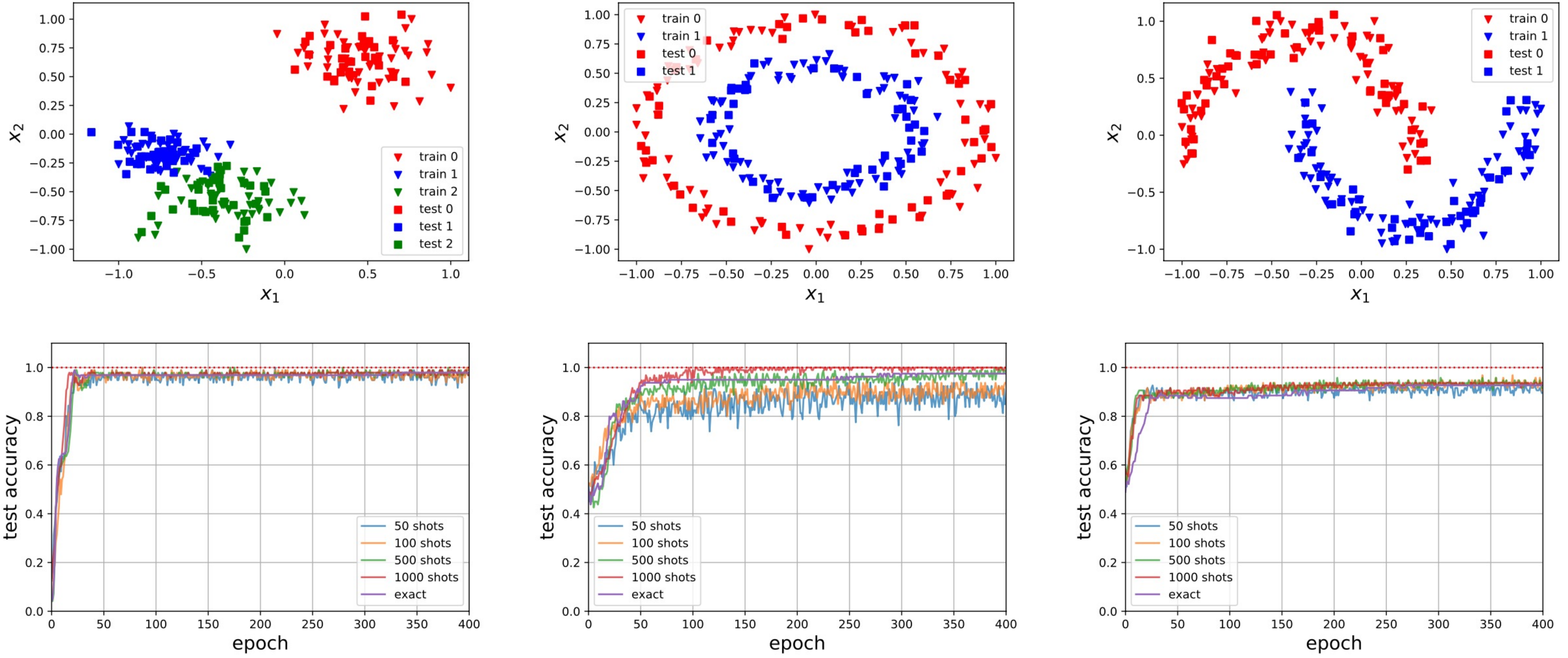
Predicted categories: $$\mathbf{y}^{\text{pred}} = \texttt{Softmax}\begin{bmatrix} \text{abs}\left(\langle \hat{X}_2 \rangle_{\rho,k}\right) \\ \text{abs}\left(\langle \hat{X}_3 \rangle_{\rho,k}\right) \end{bmatrix}$$

Classification

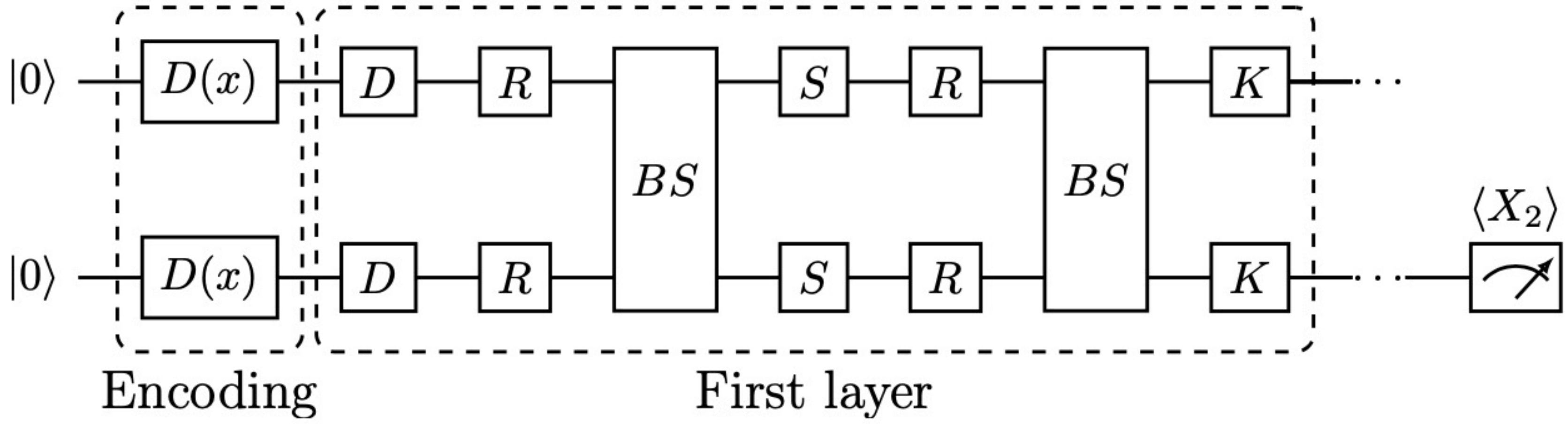


$$\mathcal{L} = \texttt{CrossEntropy}(\mathbf{y}^{\mathrm{pred}}, \mathbf{y}^{\mathrm{true}}) + \frac{\lambda}{|\mathcal{B}|} \sum ||W^{\mathrm{active}}||^2$$

Classification Results



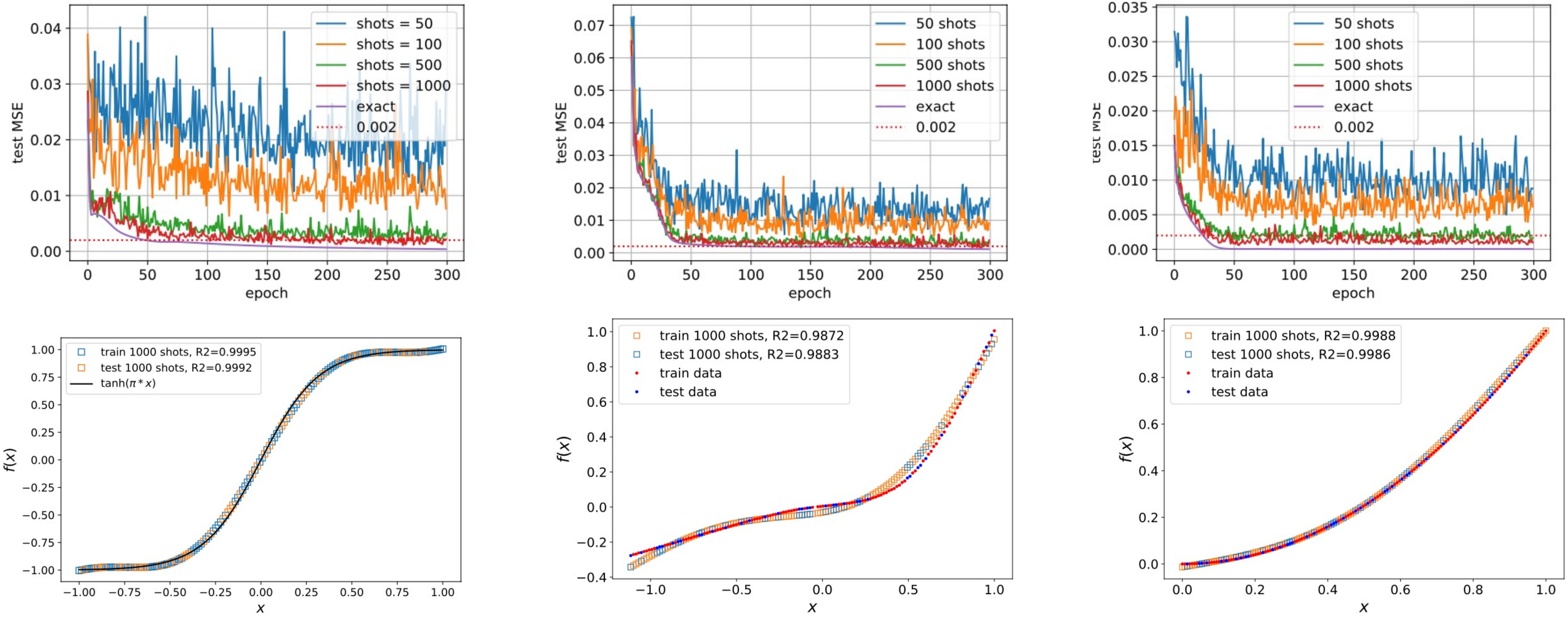
Regression



$$x \mapsto \hat{D}_1(x) \otimes \hat{D}_2(x)\,|0,0\rangle$$

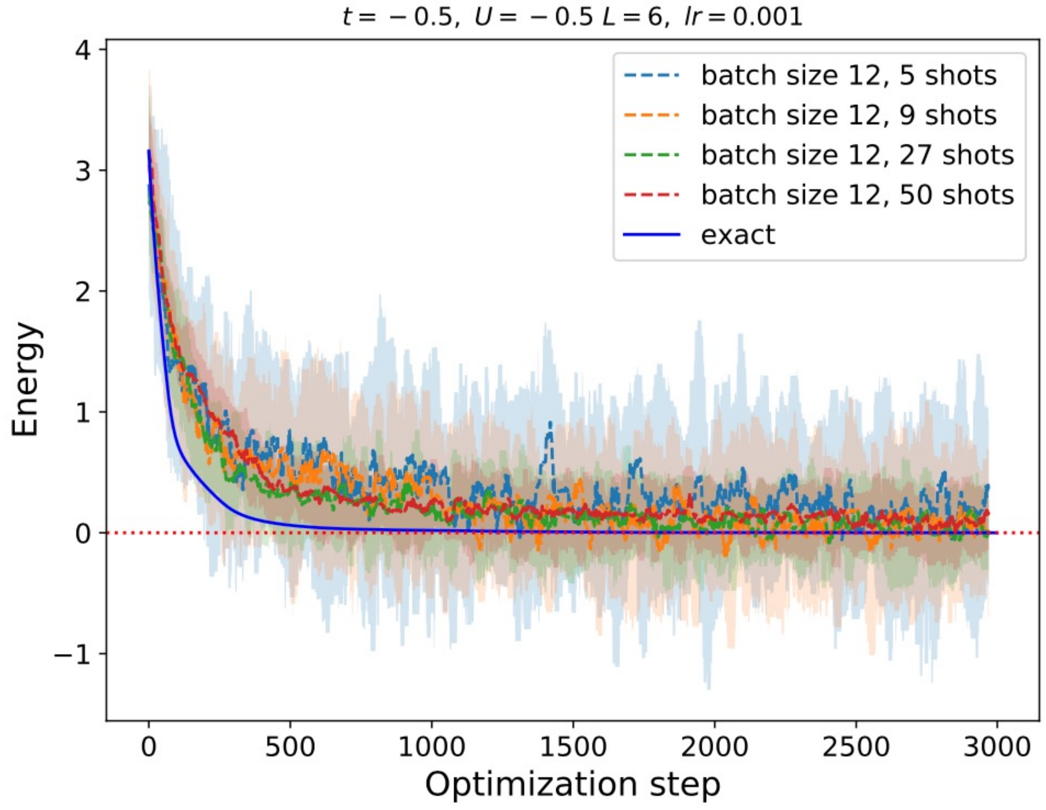
$$\mathcal{L} = \frac{1}{|\mathcal{B}|} \sum_{(x,y)\in\mathcal{B}} \left(y - \langle \hat{X}_2 \rangle_{\rho,k}\right)^2 + \frac{\lambda}{|\mathcal{B}|} \sum ||W^{\text{active}}||^2$$

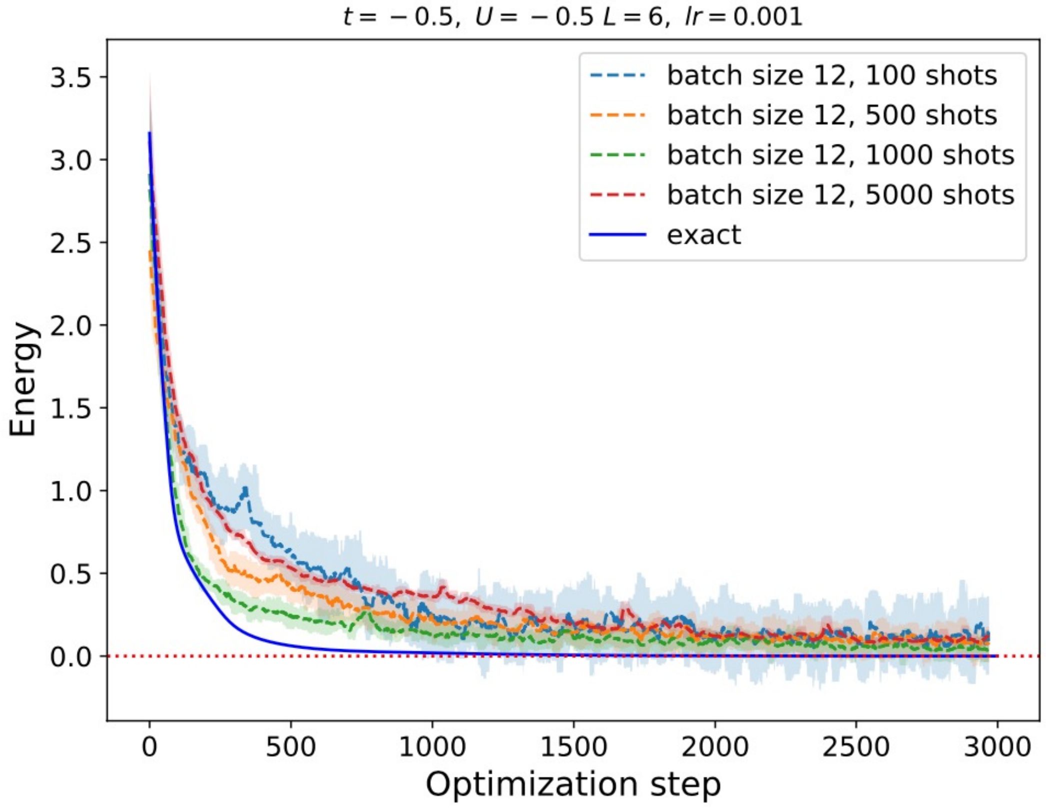
Regression Results



Bose-Hubbard model Results (VQE)

$$\hat{H} = -t \sum_{i=1}^{M-1} (\hat{a}_i^\dagger \hat{a}_{i+1} + \hat{a}_{i+1}^\dagger \hat{a}_i) + \frac{U}{2} \sum_{i=1}^{M} \hat{n}_i^2 \qquad \mathcal{L}(\boldsymbol{\theta}) = \langle \hat{H} \rangle_\rho$$





Conclusion

- We presented CV-quantum computing
- We showed the CV-QNN architecture which is analogous to the classical fully connected NNs
- We presented numerical experiments in which QNNs were used on basic datasets

Outlook

- Testing QNNs on more complex data
- Improving the QNN architecture
- Quantum-specific optimization algorithms

Thank you!

References

[1] John Preskill. Quantum Computing in the NISQ era and beyond.
Quantum, 2:79, August 2018. ISSN 2521-327X.
doi: 10.22331/q-2018-08-06-79. URL https://doi.org/10.22331/ q-2018-08-06-79.

[2] Alessio Serafini. Quantum Continuous Variables.
CRC Press, July 2017. doi: 10.1201/ 9781315118727. URL https://doi.org/10.1201/9781315118727.

[3] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. Nature, 521(7553):436–444,
May 2015. doi: 10.1038/nature14539. URL https://doi.org/10.1038/nature14539.

[4] Killoran, N., Bromley, T. R., Arrazola, J. M., Schuld, M., Quesada, N., & Lloyd, S. (2019).
Continuous-variable quantum neural networks.
Phys. Rev. Research, 1, 033063. doi:10.1103/PhysRevResearch.1.033063