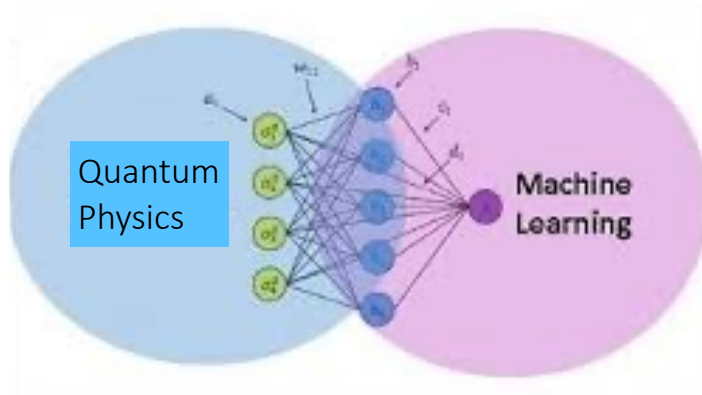# Neural-network-assisted parameter estimation for quantum detection and quantum machine learning algorithms

Yue Ban

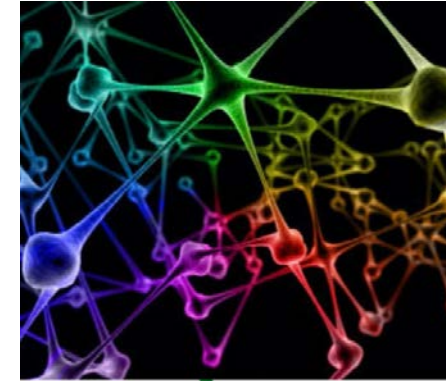Departamento de Física, Universidad Carlos III de Madrid

ybanxc@gmail.com

Quantum machine learning

Quantum Neural Network


To address complex problems in quantum physics by classical machine learning.

To apply quantum resources to enhance the performance of classical machine learning.

Neuromorphic quantum hardwares can offer the advantage as it is expected to be trained on multiple batches of real world data in parallel.

Neural networks are valuable in distinct quantum sensing scenarios.
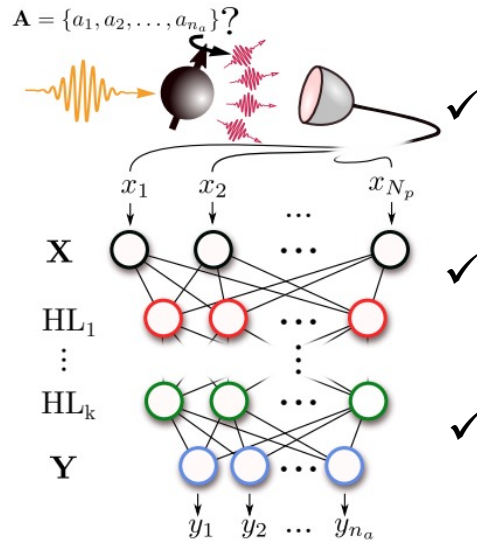
✓ A Neural Network Assisted $^{171}Yb^+$ Quantum Magnetometer

npj Quantum Inf. 8, 152 (2022).   Quantum Sci. Technol. 6, 045012 (2021).

✓ Detection of Nuclear Spins at Arbitrary Magnetic Fields

Phys. Rev. Lett. 132, 150801 (2024).

✓ Neural networks for Bayesian quantum many-body magnetometry
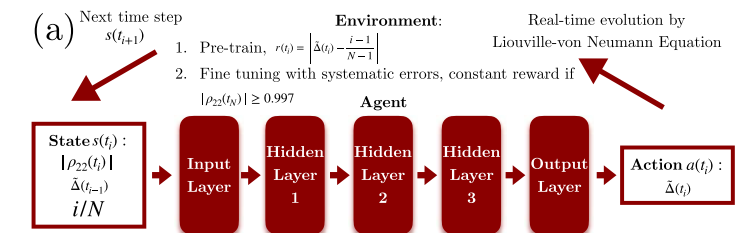
arXiv: 2212.12058.

Deep learning are useful for quantum control.

✓ Breaking adiabatic quantum control with deep learning

Phys. Rev. A 103, L040401 (2021).   Sci. China. Phys. Mech. Astron. 65, 250312 (2022).

✓ Machine-learning-assisted quantum control in a random environment

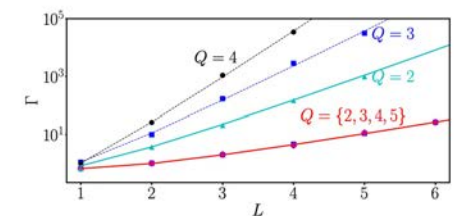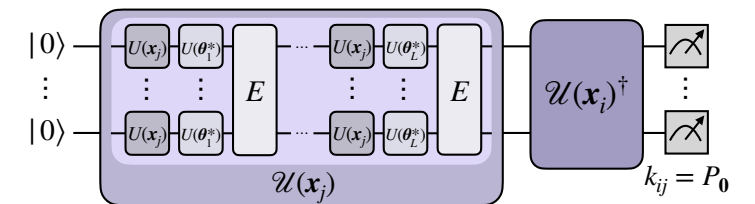Phys. Rev. Applied 17, 024040 (2022).

Digitized quantum simulation for material models

✓ Optimizing edge-state transfer in a topological SSH chain

Phys. Rev. Applied 21, 034033 (2024).

✓ Exploring Ground States of FH Model on Honeycomb Lattices

arXiv:2405.09225, accepted by npj Quantum Materials.

✓ Time-optimal control of driven oscillators by variational circuit learning

Phys. Rev. Research 5, 023173 (2023).
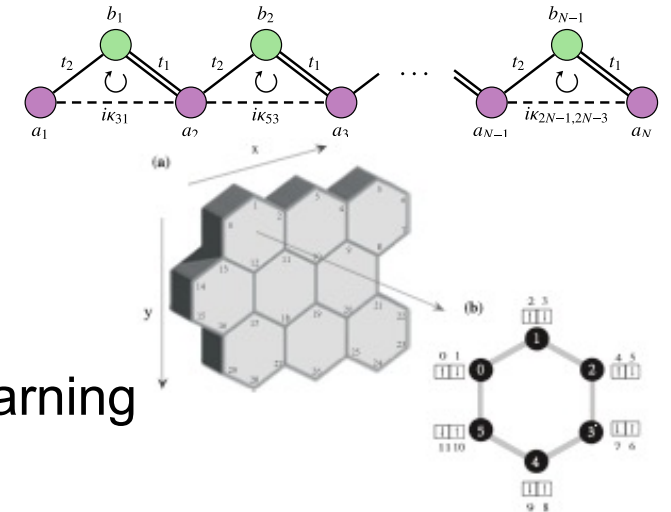
Quantum machine learning algorithms

✓ Tranining embedding quantum kernels with data reuploading quantum neural networks

arXiv:2401.04642.

✓ Satellite image classification with neural quantum kernels

arXiv:2409.20356.

✓ Regressions on quantum neural networks at maximal expressivity

arXiv:2311.06090.

n-to-n

1) Training of multi-qubit QNN

Find $\boldsymbol{\theta}*, \boldsymbol{\varphi}*$

$|0\rangle$

$|0\rangle$

2) EQK construction

$|0\rangle$

$QNN_{\boldsymbol{\theta}*,\boldsymbol{\varphi}*}(\boldsymbol{x}_j)$ $\quad QNN_{\boldsymbol{\theta}*,\boldsymbol{\varphi}*}(\boldsymbol{x}_i)^{\dagger}$

$|0\rangle$

$k_{ij} = P_{\mathbf{0}}$

Direct EQK

1-to-n

1) Training of single-qubit QNN

$|0\rangle$ — $U(\boldsymbol{x})$ $U(\boldsymbol{\theta}_1)$ $\quad U(\boldsymbol{x})$ $U(\boldsymbol{\theta}_L)$ $\quad$ Find $\boldsymbol{\theta}*$

Layer 1 $\qquad$ Layer L

$QNN_{\boldsymbol{\theta}}(\boldsymbol{x})$

$k_{ij} = P_{\mathbf{0}}$

In this talk, I will introduce:

➢ Neural Network assisted quantum sensing/metrology

   ✓ Extending the sensing working regime beyond their standard harmonic behavior.

   ✓ Characterizing target fields in the presence of large shot noise.

   ✓ Reducing computational cost for a quantum many-body magnetometer assisted by NNs.

➢ Quantum machine learning algorithms

   ✓ New algorithms for embedding quantum kernels.

   ✓ Real-data application: Satellite image classification.

   ✓ Quantum Active Learning implemented in EQNNs.

➢ **Neural Network assisted quantum sensing/metrology**

- ✓ Extending the sensing working regime beyond their standard harmonic behavior.

- ✓ Characterizing target fields in the presence of large shot noise

- ✓ Reducing computational cost for a quantum many-body magnetometer assisted by NNs.

➢ Quantum machine learning algorithms

- ✓ New algorithms for embedding quantum kernels.

- ✓ Real-data application: Satellite image classification.

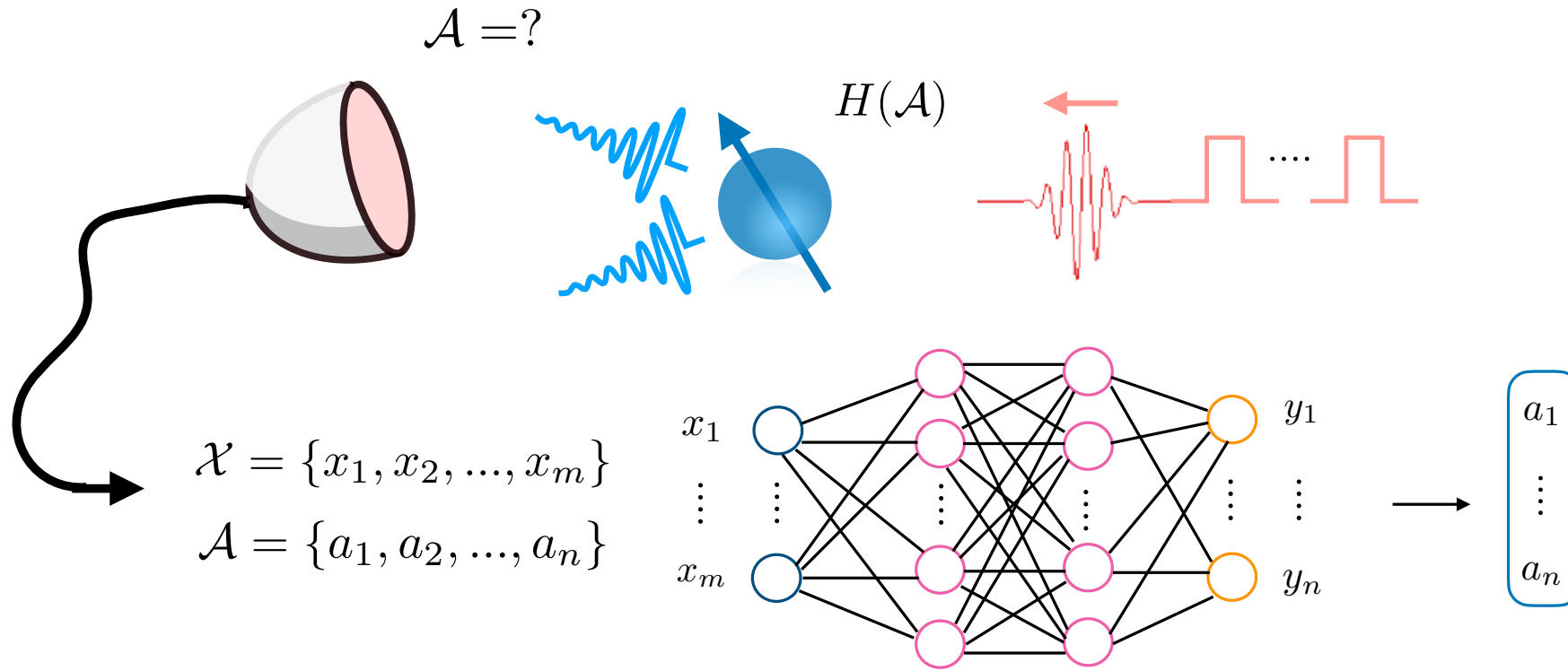- ✓ Quantum Active Learning implemented in EQNNs.

Quantum sensing is the natural playground of neural networks.

$$\mathcal{A} = ?$$

$$H(\mathcal{A})$$

$$\mathcal{X} = \{x_1, x_2, ..., x_m\}$$

$$\mathcal{A} = \{a_1, a_2, ..., a_n\}$$

x_1

x_m

y_1

y_n

a_1

a_n

Phase estimation
Parameter estimation
Sensors calibration …

A Neural Network Assisted ^{171}Yb$^+$ Quantum Magnetometer

In collaboration with

the experimental group from the University of Science and Technology of China(USTC)
Yan Chen, Ran He, Jin-Ming Cui, Yun-Feng Huang, Chuan-Feng Li, Guang-Can Guo

Prof. Guang-Can Guo
USTC, Hefei

Prof. Yun-Feng Huang
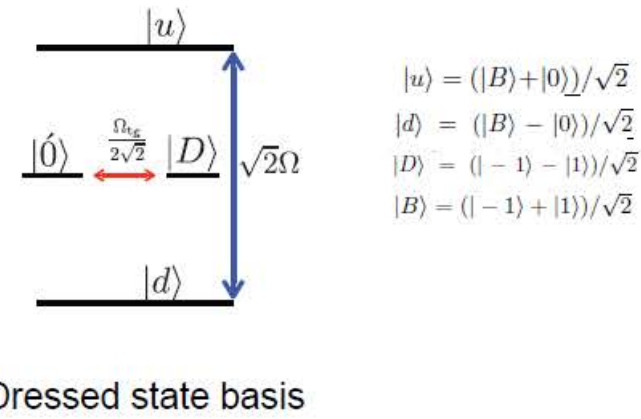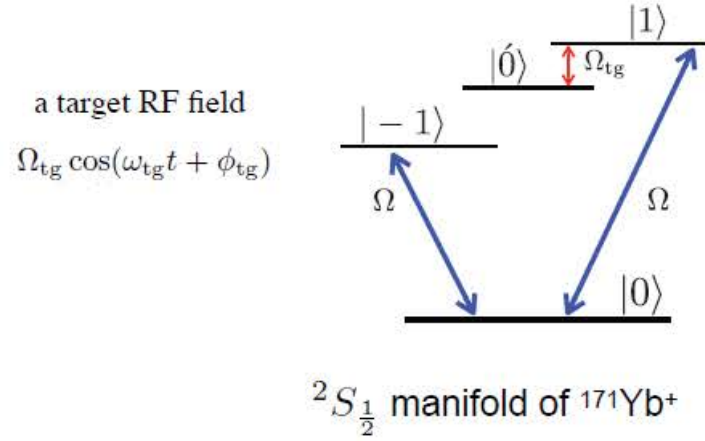USTC, Hefei

the theoreticians from Spain

Dr. Jorge Casanova
UPV/EHU, Bilbao

Dr. Ricardo Puebla
UC3M, Madrid

Y. Chen, Y. Ban, R. He, *et. al.*, npj Quantum Inf. 8, 152 (2022).

Y. Ban, *et. al.*, Quantum Sci. Technol. 6, 045012 (2021).

a target RF field

$$\Omega_{\mathrm{tg}} \cos(\omega_{\mathrm{tg}} t + \phi_{\mathrm{tg}})$$

$^{2}S_{\frac{1}{2}}$ manifold of ^{171}Yb$^+$

$$|u\rangle = (|B\rangle + |0\rangle)/\sqrt{2}$$
$$|d\rangle = (|B\rangle - |0\rangle)/\sqrt{2}$$
$$|D\rangle = (|-1\rangle - |1\rangle)/\sqrt{2}$$
$$|B\rangle = (|-1\rangle + |1\rangle)/\sqrt{2}$$
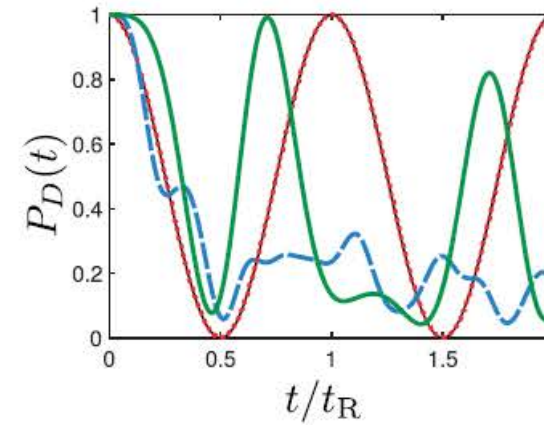
Dressed state basis

Detection of radio frequency fields
Phys. Rev. Lett. 116, 240801 (2016).

Dressed state qubit, a robust register
Nature 476, 185 (2011); Phys. Rev. Lett. 117, 220501 (2016).

Dressed state qubit approach is restricted to a narrow working regime

→ harmonic sensor responses

$$\Omega_{\mathrm{tg}} \ll \Omega \ll \omega_{\mathrm{tg}} \qquad H = -\frac{\Omega_{\mathrm{tg}}}{2\sqrt{2}}(|D\rangle\langle\acute{0}| + |\acute{0}\rangle\langle D|).$$

$$P_D(t) = \cos^2(\pi t/t_{\mathrm{R}}) \quad t_{\mathrm{R}} = 2\pi\sqrt{2}/\Omega_{\mathrm{tg}}$$

RWA

Y. Chen, Y. Ban, R. He, et. al., npj Quantum Inf. 8, 152 (2022).

Y. Ban, et. al., Quantum Sci. Technol. 6, 045012 (2021).

A Neural Network Assisted ^{171}Yb$^+$ Quantum Magnetometer

$$\mathbf{X} = \{P_1, P_2, ..., P_{N_p}\} \quad \mathbf{?} \quad \mathbf{Y} \approx \mathbf{A} = \{\Omega_{\text{tg}}, \xi\}$$

an ^{171}Yb$^+$ atomic sensor

\+

adequately trained neural networks

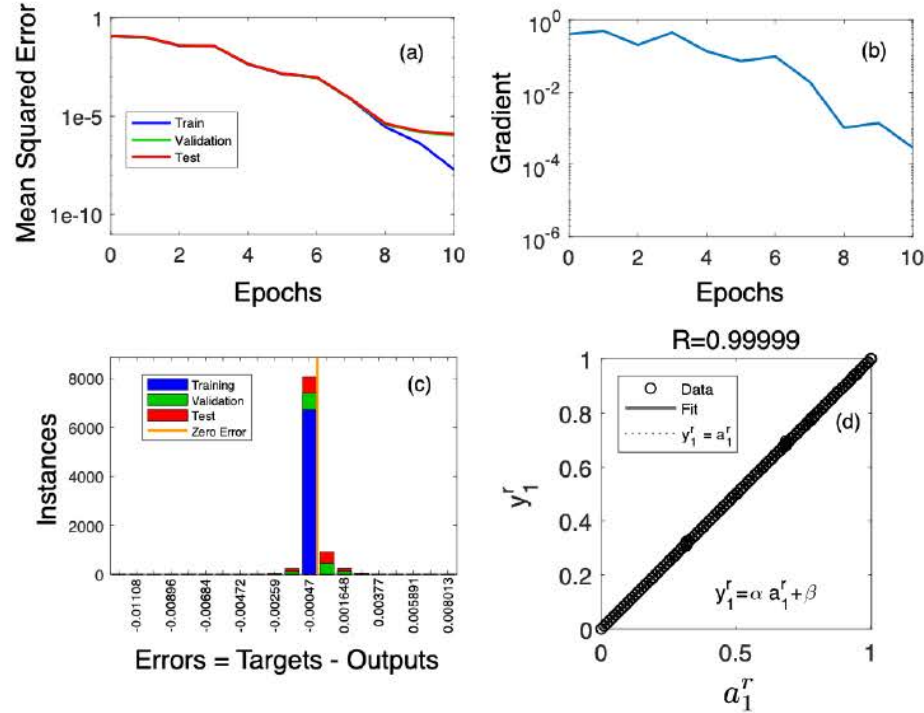A versatile magnetometer enables
the characterization of target fields.
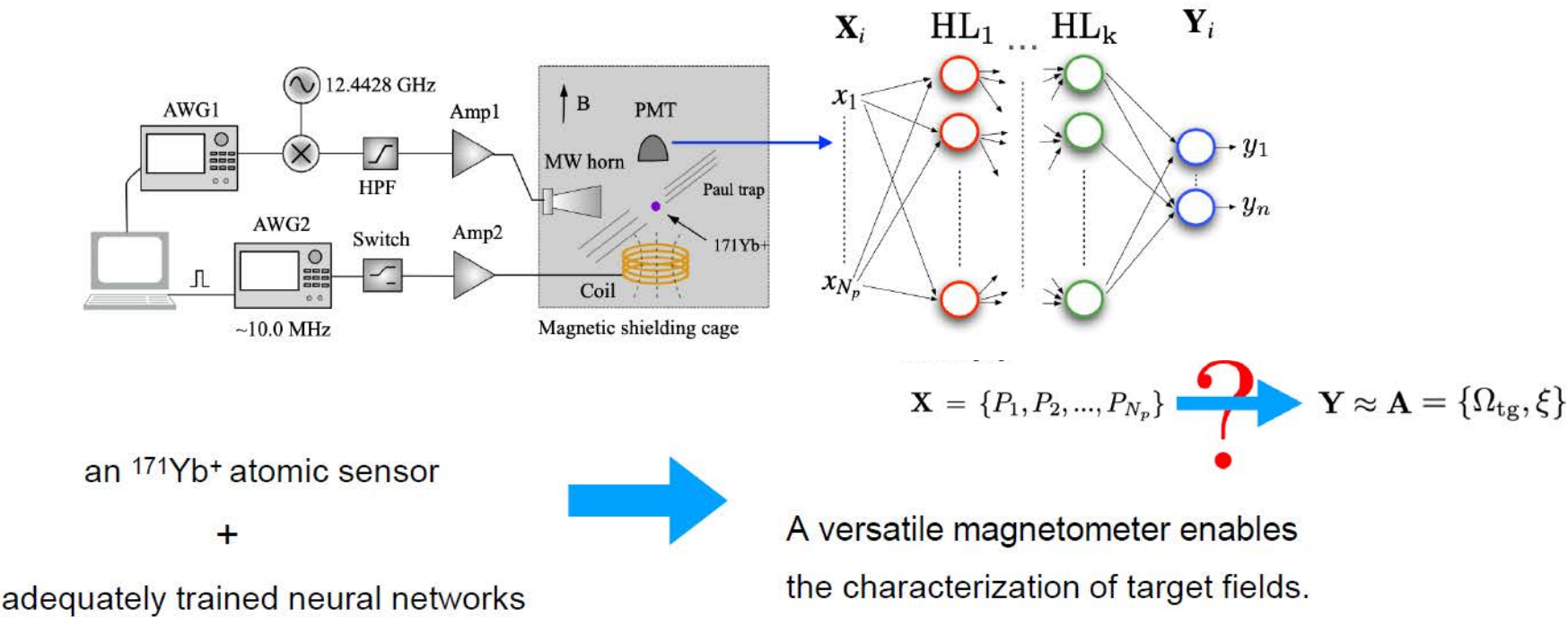
Scenario I: a reduced number of measurements.

Scenario II: Continuous data acquisition.

the working regime: responses beyond the harmonic behavior!

Scenario I: A reduced number of measurements.

Training results:

$$\phi_{\text{tg}} = 0 \quad \omega_{\text{tg}} = (2\pi) \times 10.56 \text{ MHz}$$

Outputs obtained from the NN, when the input data are the experimental responses

| $a_1(\times 2\pi \text{ kHz})$	y_1 ($\times 2\pi$ kHz) with $N_m = 100$	y_1 ($\times 2\pi$ kHz) with $N_m = 30$
1.1487	1.1827	1.1731
1.7229	1.7473	1.8060
2.2566	2.3109	2.3207
2.8760	2.8616	2.8527
3.4429	3.4961	3.4947
4.0098	4.0391	4.0502
4.5778	4.6283	4.6386
5.1834	5.1856	5.2208
5.7140	5.7448	5.7297
6.2797	6.1482	6.2134
6.8397	6.8358	6.6896
7.3927	7.3086	7.3471
7.9319	8.0864	8.1129
8.4527	8.3775	8.3870
8.9493	8.8414	8.7825

$$\bar{F} = \frac{1}{N}\sum_{j=1}^{N} F_j \quad F_j = 1 - |y_1^j - a_1^j|/a_1^j \quad a_1 = \Omega_{\text{tg}}$$

$N_m = 100 \quad \bar{F} = 98.76\% \quad SD = 0.7762\%$

$N_m = 30 \quad \bar{F} = 98.31\% \quad SD = 1.1483\%$

11

Scenario I: A reduced number of measurements.

$$\omega_{\mathrm{tg}} = (2\pi) \times 10.56 \text{ MHz} \qquad \phi_{\mathrm{tg}} = 0 \qquad \Omega = (2\pi) \times 5.5 \text{kHz}$$
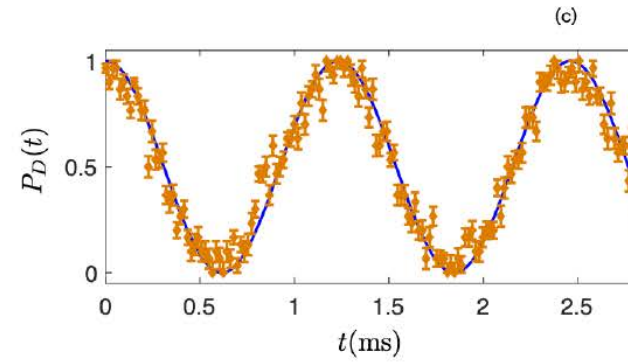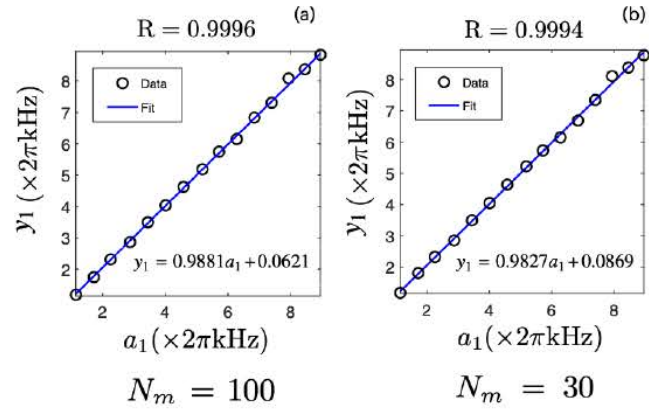
$$N_m = 30 \qquad \Omega_{\mathrm{tg}} = (2\pi) \times 1.1487 \text{ kHz} \qquad y_1 = (2\pi) \times 1.1731 \text{ kHz}$$

$$P_D(t) = \cos^2(\pi t / t_{\mathrm{R}})$$
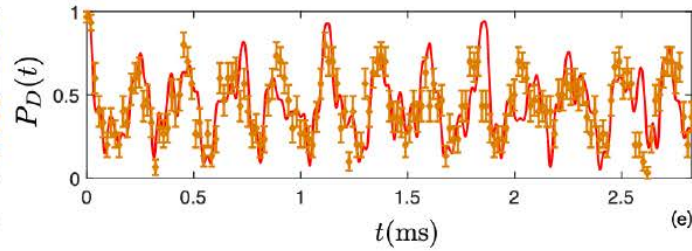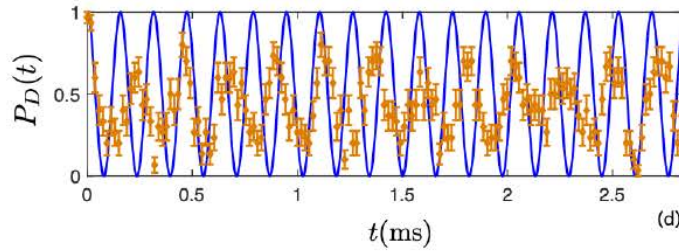
$$N_m = 30 \qquad \Omega_{\mathrm{tg}} = (2\pi) \times 8.9493 \text{ kHz} \qquad y_1 = (2\pi) \times 8.7825 \text{ kHz}$$

Scenario II: Continuous data acquisition.

no reinitialization of the RF field is possible; $N_m = 1$

RF source is always on.

RF field $\Omega_{tg}\cos(\omega_{tg}t)$ $t_2 \in [0,6]$ms

Stage 1 $t_1 = 6.097$ms   **Stage 2** $t_D = 6$ms   **Stage 3** $t_3 = 2.447$ms

Preparing $|D\rangle$ state   Interaction between the sensor and the RF field   Measuring $|D\rangle$ state

To get one input string:

251 times of repetition for three stages

96 Ω_{tg} values $\in 2\pi\times[0.5, 10]$ kHz

To get training/validation/test datasets

1800 times of repetition for each Ω_{tg}
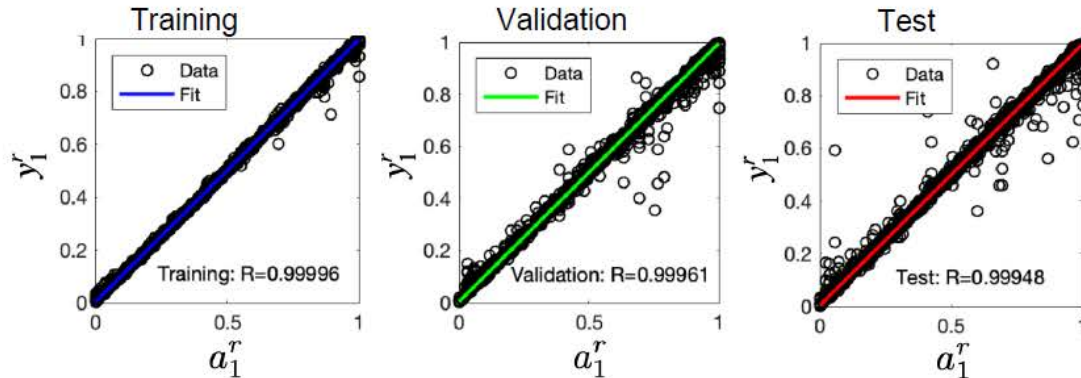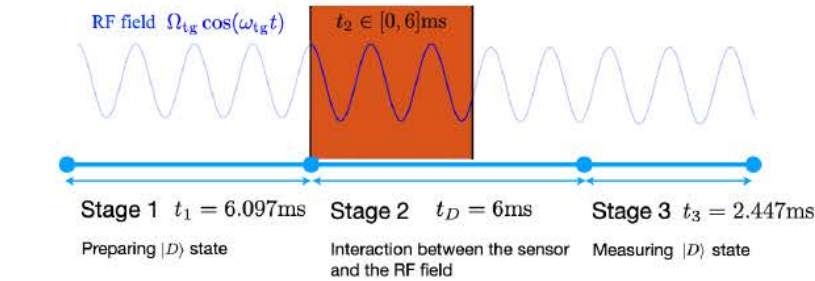
Targets $a_1 (\times 2\pi$ kHz)	Average values $\bar{y}_1$ ($\times 2\pi$ kHz)	Standard deviation SD ($\times 2\pi$ kHz)
0.7542	0.8417	0.0690
1.1840	1.2759	0.0833
1.4044	1.4384	0.0222
1.6206	1.6542	0.0660
2.1572	2.1720	0.0543
4.2265	4.2761	0.0594
6.3960	6.2988	0.0776
8.3689	8.2255	0.1531

For each Ω_{tg}, 20 experimentally obtained strings

Training: R=0.99996   Validation: R=0.99961   Test: R=0.99948

$R = 0.9993$ (a)

$y_1 = 0.9720a_1 + 0.1002$

(b)

$a_1 = (2\pi) \times 2.1572$kHz
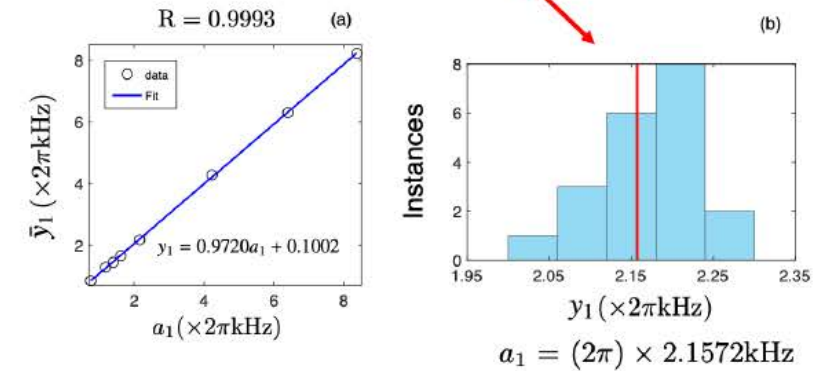
Other estimators: e.g. Bayesian inference.

$$p(\theta|\mathbf{X}) \propto p(\mathbf{X}|\theta)p(\theta)$$

$\mathbf{X}$: data obtained by interrogating quantum sensor at different time instants

Versatile Atomic Magnetometry Assisted by Bayesian Inference

R. Puebla, Y. Ban, J.F. Haase, M.B. Plenio, M. Paternostro, and J. Casanova

Phys. Rev. Applied **16**, 024044 (2021).

A Bayesian analysis and a NN provide a comparable precision for the estimators.

	Bayesian estimator	Neural Networks
Accurate microscopic model	Necessary	Not necessary
Prior knowledge	More	Less
Operation time	Long	Short (once trained)
Computational cost	More	Less

Automatic Detection of Nuclear Spins at Arbitrary Magnetic Fields via Signal-to-Image AI Model

a quantum node: an NV and n ^{13}C nearby nuclear spins

$$P_x = \frac{1}{2}\left(1 + \prod_{j=1}^{n} M_j\right)$$

$$M_j = 1 - m_{j,x}^2 \frac{(1 - \cos\alpha_j)(1 - \cos\beta)}{1 + \cos\alpha_j\cos\beta - m_{j,z}\sin\alpha_j\sin\beta}\sin\frac{N\phi_j^2}{2}$$
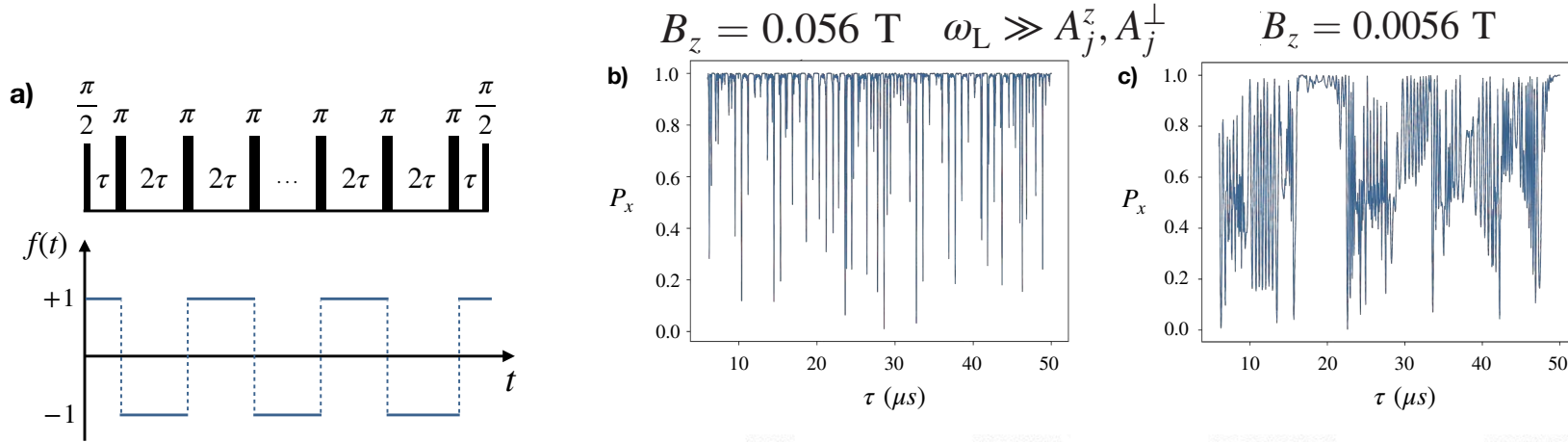
(i) high accuracy in predicting hyperfine vectors over a wide range

(ii) handles noisy signals

(iii) does not require prior knowledge,

(iv) performs well in low-field conditions
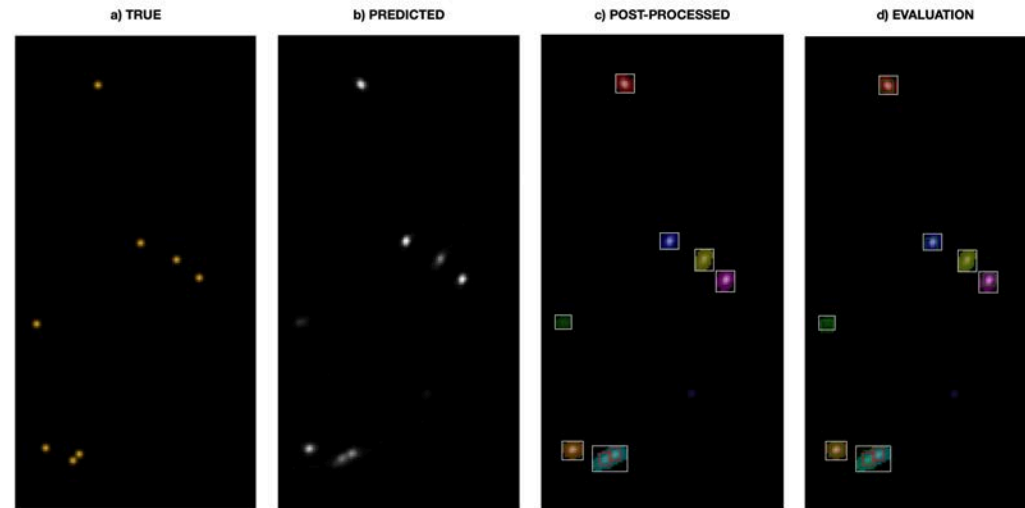
Bayesian Inference Assisted by Neural Networks

Input $\quad \Theta = \{\theta_1, \ldots, \theta_M\}$

Output $\mathbf{Y} = F(\Theta)$

θ_1
θ_2
$\vdots$
θ_M

Y_1
Y_2
$\vdots$
Y_{N_T}

Training of the NN:
Calibration stage

Quantum many-body system
as a sensor: $\mathbf{D} \rightarrow \Theta$?

Likelihood using the suitably
trained NN

Bayesian inference
$P(\Theta|\mathbf{D}) \propto P(\mathbf{D}|\Theta)P(\Theta)$

Estimated target parameters
beyond standard quantum limit
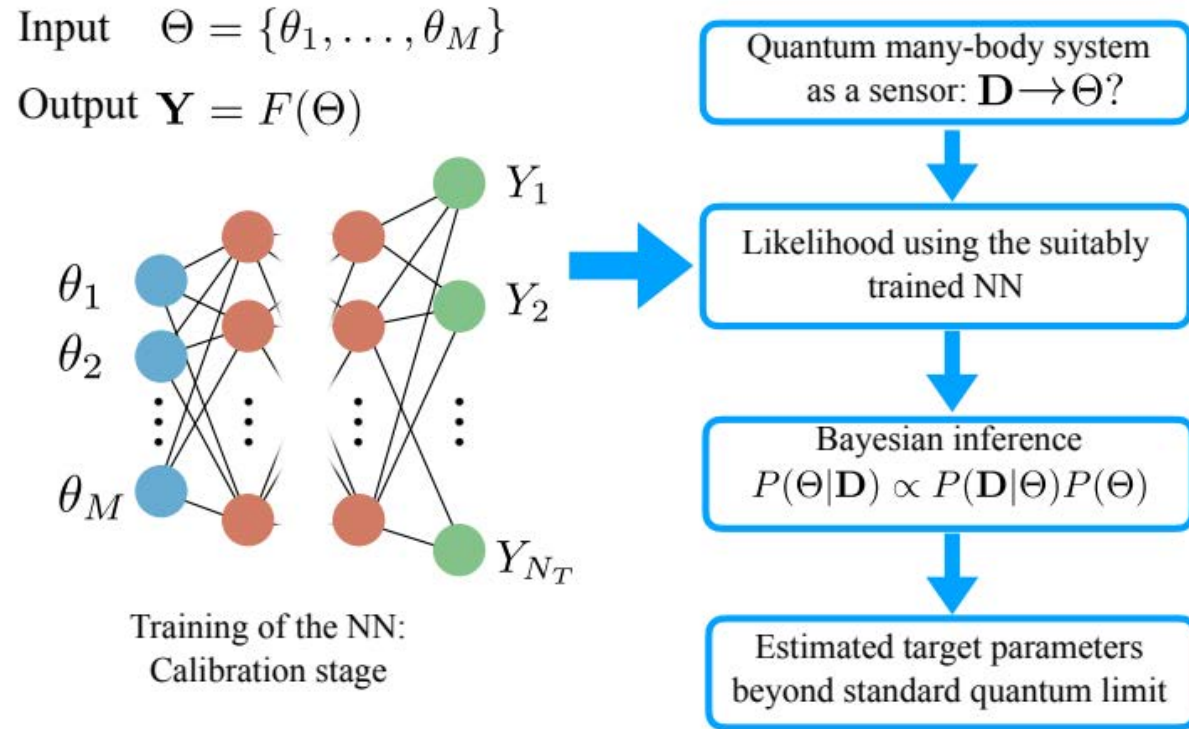
$$P(\mathbf{D}|\Theta) = \Pi_{j=1}^{N_T} f(X_j, N_m, \langle A(t_j; \Theta)\rangle).$$

$$\langle \hat{A}(t; \Theta)\rangle = \mathrm{Tr}\left[\hat{A}\hat{U}(t)\hat{\rho}_0\hat{U}^\dagger(t)\right]$$

$$\mathbf{Y}_{j=1,\ldots,N_T} \equiv F(\Theta)_{j=1,\ldots,N_T} \approx \langle \hat{A}(t_{j=1,\ldots,N_T}; \Theta)\rangle$$

$$P(\mathbf{D}|\Theta) = \Pi_{j=1}^{N_T} f(X_j, N_m, F(\Theta)_{j,\ldots,N_T}).$$

Neural networks faithfully reproduce the dynamics of quantum many-body sensors, thus allowing for an efficient Bayesian analysis.

arXiv: 2212.12058.

Simulation of quantum many-body dynamics by neural networks

XXZ spin-1/2 chain

Input $\Theta = \{\theta_1, \ldots, \theta_M\}$

Output $\mathbf{Y} = F(\Theta)$
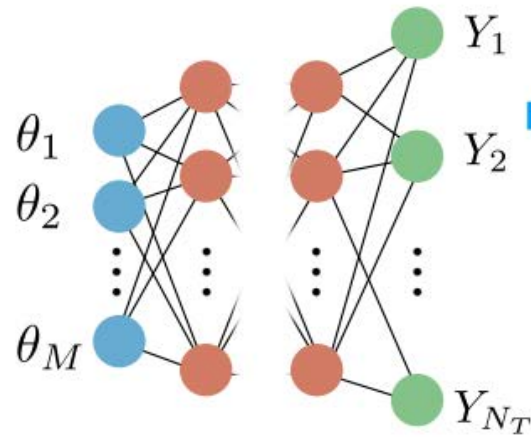
Y_1

θ_1

θ_2

Y_2

θ_M

Y_{N_T}

Training of the NN:
Calibration stage

Quantum many-body system
as a sensor: $\mathbf{D} \rightarrow \Theta$?

Likelihood using the suitably
trained NN

Bayesian inference
$P(\Theta|\mathbf{D}) \propto P(\mathbf{D}|\Theta)P(\Theta)$

Estimated target parameters
beyond standard quantum limit

1D external magnetic field

$$\hat{H}_1(g_x) = -\sum_{i=1}^{N-1}(\hat{\sigma}_i^x \hat{\sigma}_{i+1}^x + \hat{\sigma}_i^y \hat{\sigma}_{i+1}^y + J_z \sigma_i^z \sigma_{i+1}^z) + g_x \sum_{i=1}^{N} \hat{\sigma}_i^x.$$

$$F_1(g_x)_{j=1,\ldots,N_T} \approx \langle \hat{A}_1(t_{j=1,\ldots,N_T}; g_x) \rangle$$

$$\hat{A}_1 = (\hat{\sigma}_{N/2}^x + 1)/2$$

$$P(\mathbf{D}|\Theta) = \Pi_{j=1}^{N_T} f(X_j, N_m, F(\Theta)_{j,\ldots,N_T}).$$

arXiv: 2212.12058.

Estimation on 2D, 3D external magnetic fields also work!

Simulation of quantum many-body dynamics by neural networks

XXZ spin-1/2 chain

Bayes' theorem

$$P(\Theta|\mathbf{D}) \propto P(\mathbf{D}|\Theta)P(\Theta).$$

$$\theta_j^{\text{est}} = \int d\theta_j \theta_j P(\theta_j|\mathbf{D}) \qquad (\Delta\theta_j^{\text{est}})^2 = \int d\theta_j (\theta_j - \theta_j^{\text{est}})^2 P(\theta_j|\mathbf{D}),$$
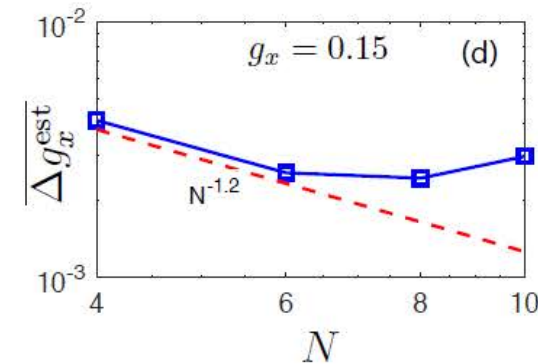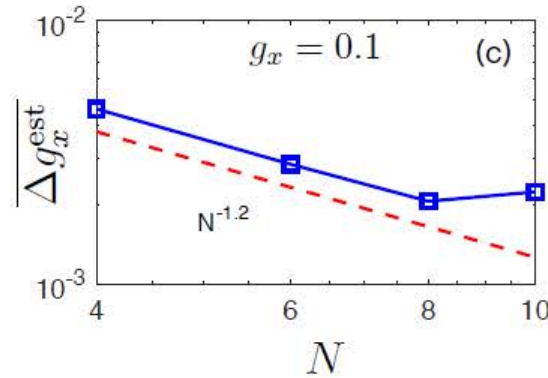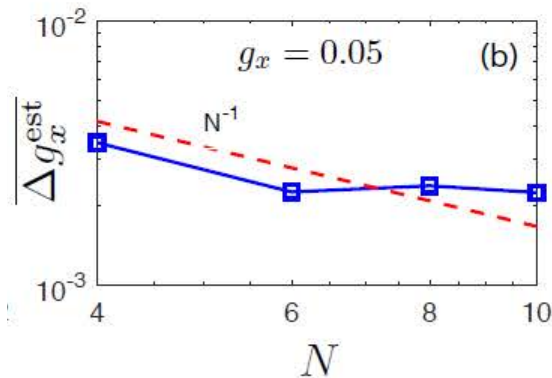
Standard quantum limit $\Delta\theta \propto N^{-1/2}$ Heisenberg limit $\Delta\theta \propto N^{-1}$

Summary on NN-assisted quantum sensing/metrology

✓ Using NNs to decipher the information contained in the sensor responses.

✓ Continuous data acquisition and precision.

✓ Extending the working regime of quantum sensors with Robustness against noise.

✓ Reproduction of microscopic modelling of the quantum many-body system by neural networks.

➢ Neural Network assisted quantum sensing/metrology

   ✓ Extending the sensing working regime beyond their standard harmonic behavior.

   ✓ Characterizing target fields in the presence of large shot noise

   ✓ Reducing computational cost for a quantum many-body magnetometer assisted by NNs.

➢ **Quantum machine learning algorithms**

   ✓ New algorithms for embedding quantum kernels.

   ✓ Real-data application: Satellite image classification.

   ✓ Quantum Active Learning implemented in EQNNs.

1) Training of single-qubit QNN

$|0\rangle$ $U(\boldsymbol{x})$ $U(\boldsymbol{\theta}_1)$

Layer 1   Layer L

$\text{QNN}_{\boldsymbol{\theta}}(\boldsymbol{x})$

Quantum kernel and its application to real data
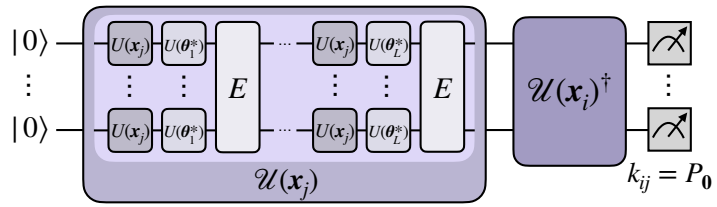
Training embedding quantum kernels with data re-uploading quantum neural networks

arXiv:2401.04642

$k_{ij} = P_{\mathbf{0}}$
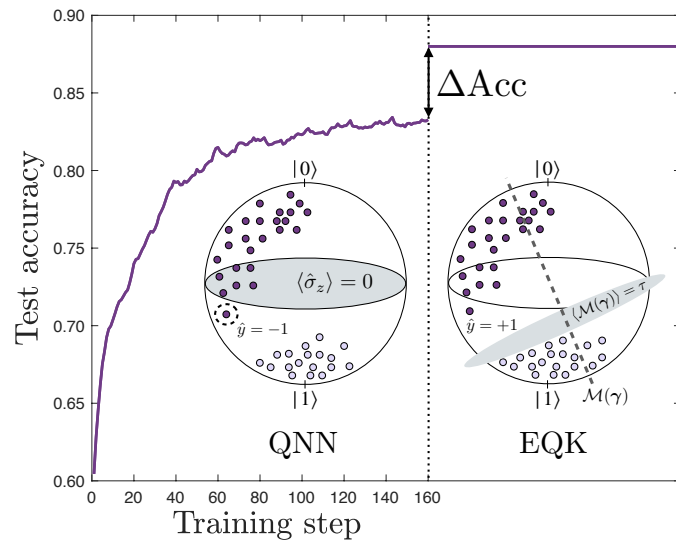
Satellite image classification with neural quantum kernels   arXiv:2409.20356

Without PV panel

With PV panel

Quantum machine learning algorithms

Data-reuploading quantum neural networks (Explicit model)
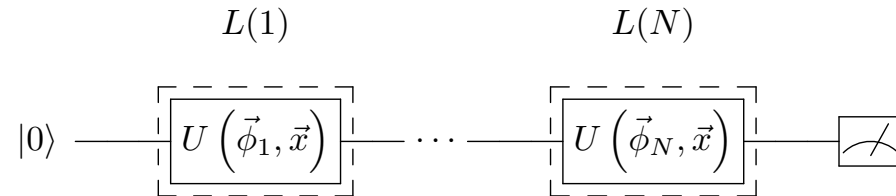
Unifying three approaches in the framework of linear models
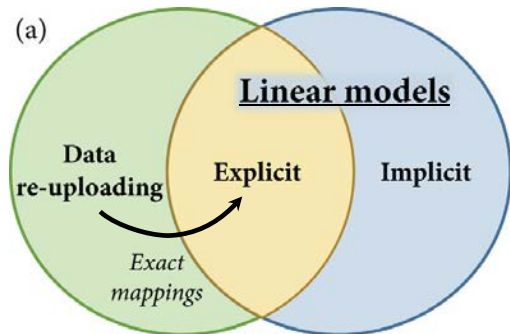
Nature Commun. 14, 517 (2023)
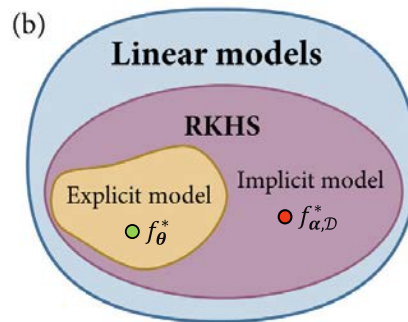
Representer Theorem

Explicit Model $\rho(\boldsymbol{x}) = S(\boldsymbol{x})|0\rangle\langle 0|^{\otimes n} S^{\dagger}(\boldsymbol{x})$

Implicit Model $f_{\boldsymbol{\alpha},X} = \sum_{i=1}^{M} \alpha_i\, k(\boldsymbol{x}, \boldsymbol{x}_i)$

$k(\boldsymbol{x}, \boldsymbol{x}_i) = \mathrm{tr}(\rho(\boldsymbol{x})\rho(\boldsymbol{x}_i))$ $\{\boldsymbol{x}_i\}_{i=1}^{M}$

Quantum kernel methods

Feature space = Hilbert space

Quantum embedding $S(\boldsymbol{x})$

$S(\boldsymbol{x})|\mathbf{0}\rangle = |\phi(\boldsymbol{x})\rangle$

Classical data $\boldsymbol{x}$

Look for linear separation in Hilbert space

$k(\boldsymbol{x}, \boldsymbol{x}')$
Overlaps

$k(\boldsymbol{x}, \boldsymbol{x}')$ Overlaps $\longrightarrow$ K $\longrightarrow$ Predictions

SVM

$$K = \begin{pmatrix} k(\boldsymbol{x}_1, \boldsymbol{x}_1) & \cdots & k(\boldsymbol{x}_1, \boldsymbol{x}_M) \\ \vdots & \ddots & \vdots \\ k(\boldsymbol{x}_M, \boldsymbol{x}_1) & \cdots & k(\boldsymbol{x}_M, \boldsymbol{x}_M) \end{pmatrix}$$

$k(\boldsymbol{x}, \boldsymbol{x}') = |\langle \phi(\boldsymbol{x}) | \phi(\boldsymbol{x}') \rangle|^2$

$\mathcal{O}(M^2)$ overlap evaluations

$|0\rangle$ ⋮ $|0\rangle$ — $S(\boldsymbol{x}')$ — $S^\dagger(\boldsymbol{x})$ — P_0

Embedding quantum kernel (EQK) $k(\boldsymbol{x}, \boldsymbol{x}')$

X
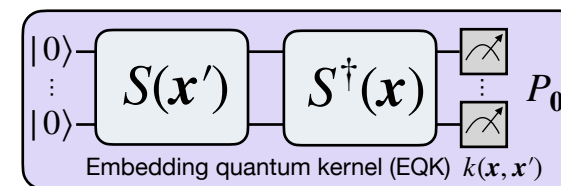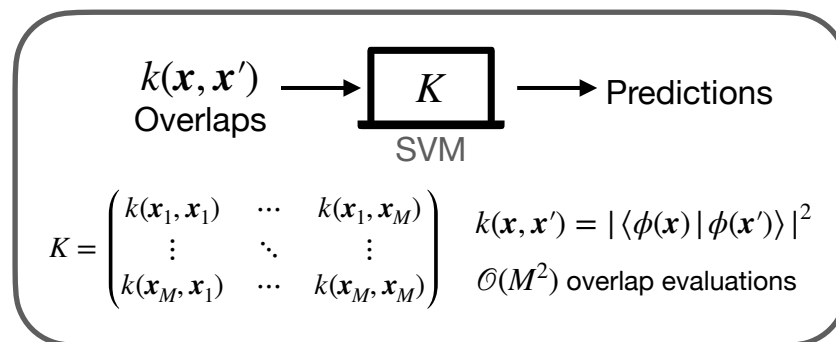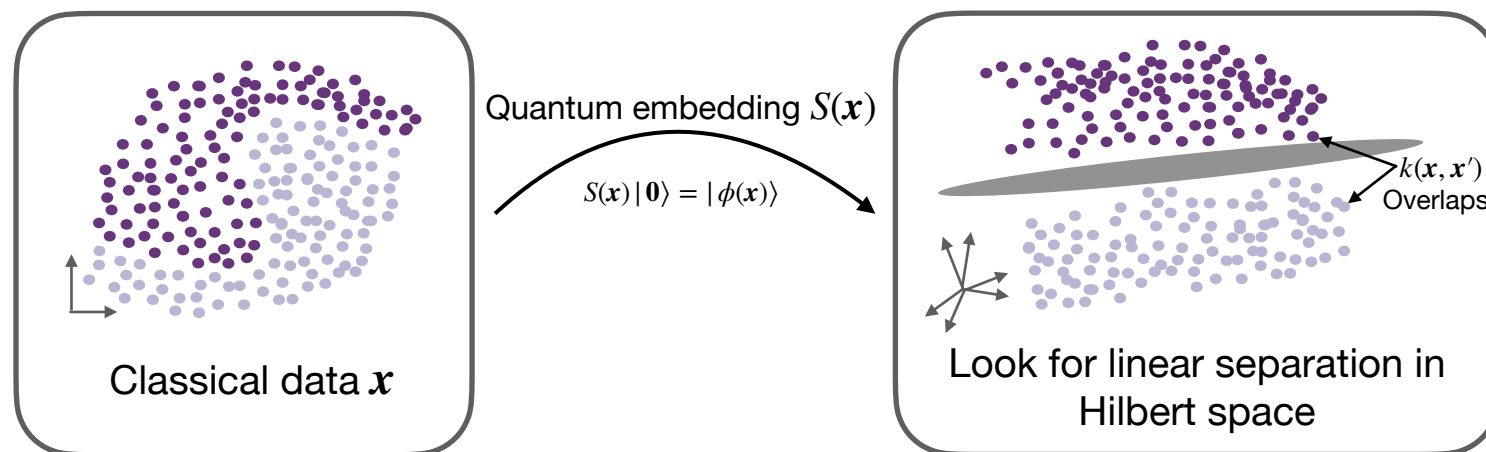
Our protocol: neural quantum kernels

Constructing an EQK from the training of a quantum neural network (QNN)

So far

Ansatz
$S_{\boldsymbol{\varphi}} \to k_{\boldsymbol{\varphi}}$

Kernel matrix
$K_{\boldsymbol{\varphi}}$

→ Predictions

Training loop

Our approach

QNN Ansatz
$S_{\boldsymbol{\varphi}}$

Kernel matrix
$K_{\boldsymbol{\varphi}}$

→ Predictions

Training loop

Encoding information about the problem

$|0\rangle$

$\langle \hat{\sigma}_z \rangle = 0$

$\hat{y} = -1$

$|1\rangle$

$|0\rangle$

$\langle \mathcal{M}(\gamma) \rangle = \tau$

$\hat{y} = +1$

$|1\rangle \quad \mathcal{M}(\gamma)$

pablojesus.rodriguez@ehu.eus
pablojesus.rodriguez@ehu.eus
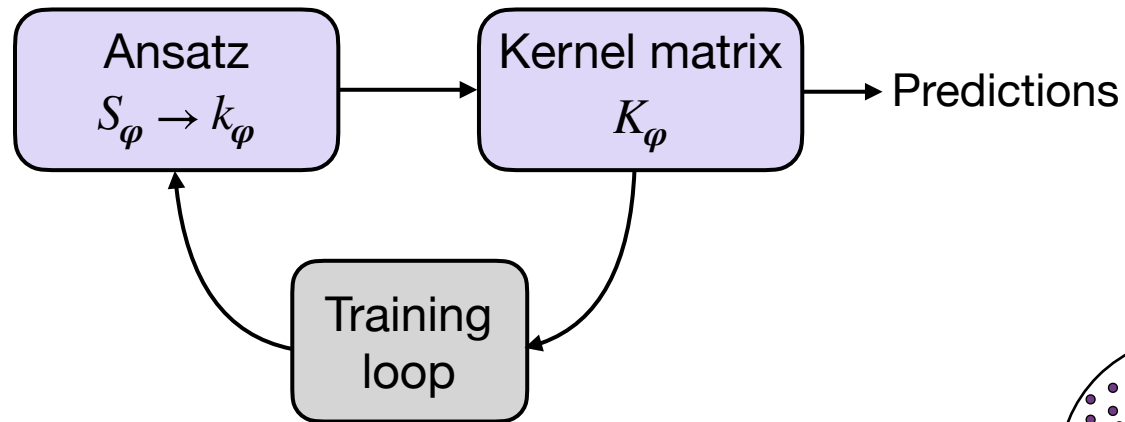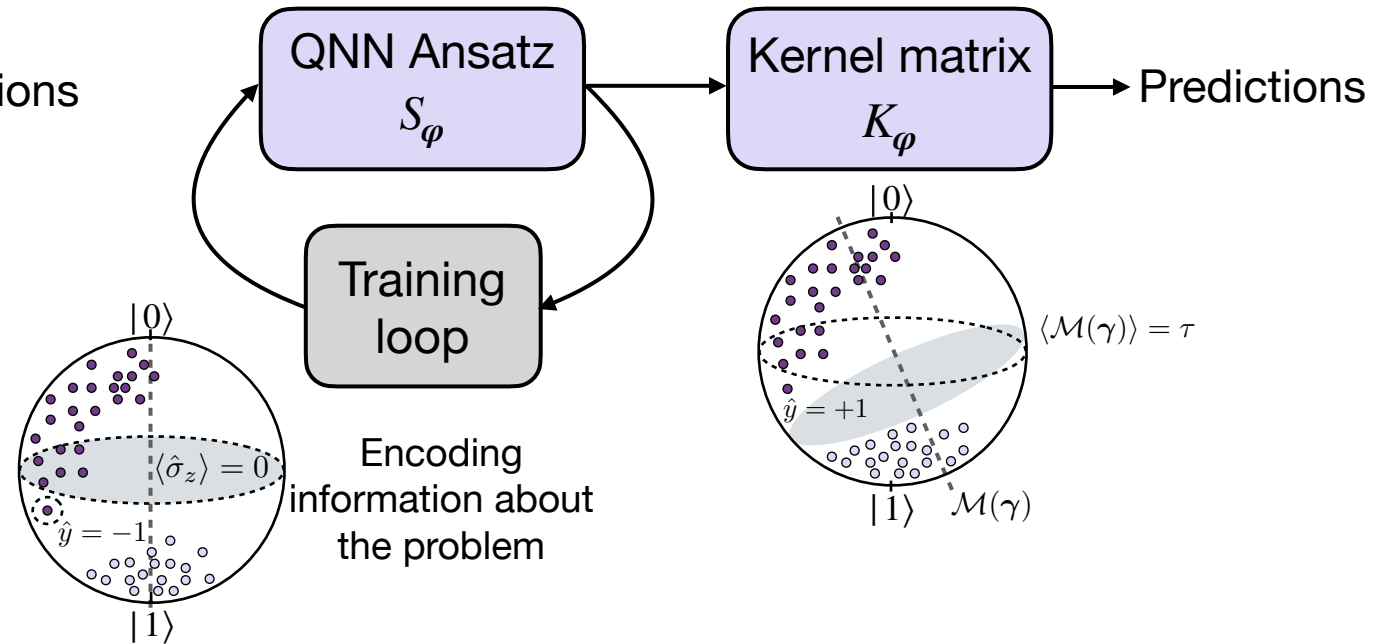
Constructing Embedded Quantum Kernel (EQK) from QNN training

A single-qubit QNN $\longrightarrow$ A single-qubit EQK
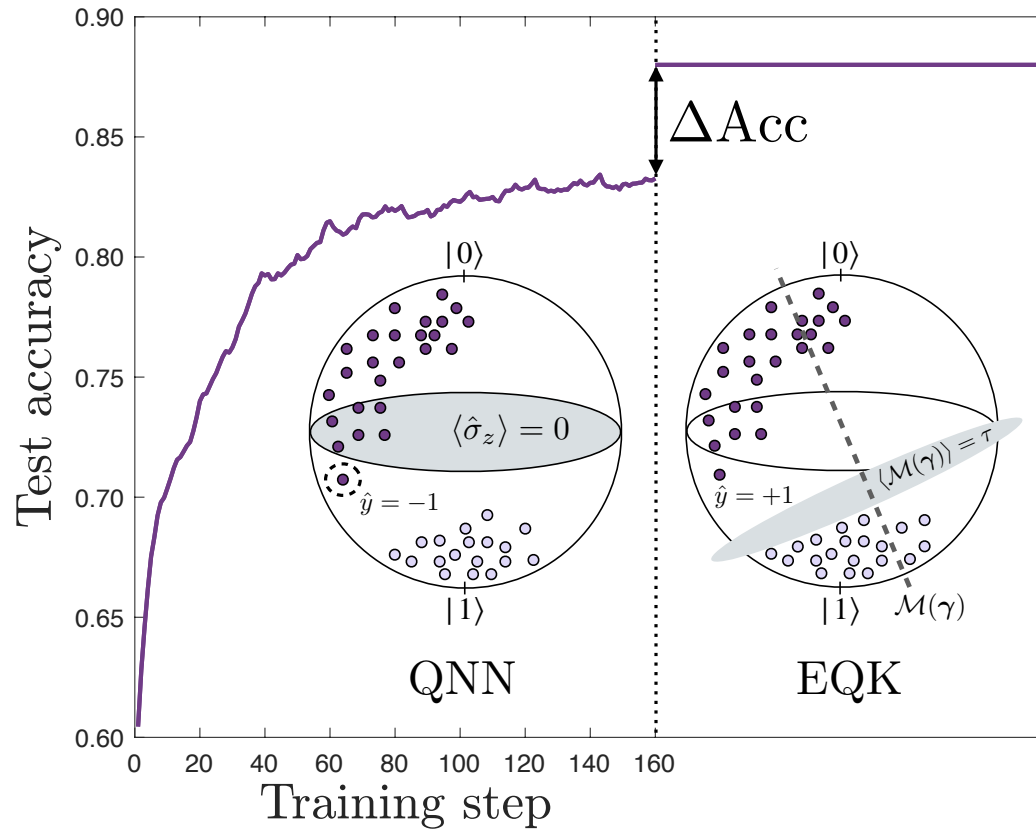
$$\mathrm{QNN}_{\boldsymbol{\theta}}(\boldsymbol{x}) \equiv \prod_{l=1}^{L} U(\boldsymbol{\theta}_l)\, U(\boldsymbol{x}) = U(\boldsymbol{\theta}_L)\, U(\boldsymbol{x}) \ldots U(\boldsymbol{\theta}_1)\, U(\boldsymbol{x})$$

$$|\phi_{\boldsymbol{\theta}}(\boldsymbol{x}_i)\rangle = U(\boldsymbol{\theta}_L)\, U(\boldsymbol{x}) \ldots U(\boldsymbol{\theta}_1)\, U(\boldsymbol{x})|0\rangle$$

$$f_{\mathrm{cost}} = \frac{1}{M} \sum_{i=1}^{M} \left(1 - |\langle \phi_l^i | \phi_{\boldsymbol{\theta}}(\boldsymbol{x}_i)\rangle|^2\right) \quad \hat{y}[\boldsymbol{x}_t] = \begin{cases} +1 & \text{if } |\langle 0|\phi(\boldsymbol{x}_t)\rangle|^2 \geq 1/2, \\ -1 & \text{if } |\langle 0|\phi(\boldsymbol{x}_t)\rangle|^2 < 1/2. \end{cases}$$

SVM separates +1 class (SV$_{+1}$) and -1 class (SV$_{-1}$)

$$\sum_{i \in \mathrm{SV}_{+1}} \alpha_i\, y_i\, |\langle \phi(\boldsymbol{x}_i)|\phi(\boldsymbol{x})\rangle|^2 + \sum_{i \in \mathrm{SV}_{+1}} \alpha_i\, y_i\, |\langle \phi(\boldsymbol{x}_i)|\phi(\boldsymbol{x})\rangle|^2$$
$$= N_{\mathrm{SV}} \alpha\, |\langle 0|\phi(\boldsymbol{x})\rangle|^2 - N_{\mathrm{SV}} \alpha\, |\langle 1|\phi(\boldsymbol{x})\rangle|^2 = 0$$

Equivalent to the hyperplane $\quad |\langle 0|\phi(\boldsymbol{x})\rangle|^2 = |\langle 1|\phi(\boldsymbol{x})\rangle|^2$

arXiv:2401.04642

$\langle \hat{\sigma}_z \rangle = 1/2$

25

Training of the *n*-qubit QNN: an iterative construction starting from a single-qubit QNN

$$\text{QNN}_{\boldsymbol{\theta}}(\boldsymbol{x}) \equiv \prod_{l=1}^{L} U(\boldsymbol{\theta}_l) \, U(\boldsymbol{x}) = U(\boldsymbol{\theta}_L) \, U(\boldsymbol{x}) \dots U(\boldsymbol{\theta}_1) \, U(\boldsymbol{x})$$

$$\text{QNN}_{\boldsymbol{\theta},\boldsymbol{\varphi}}(\boldsymbol{x}) = \prod_{l=1}^{L} \left(\prod_{s=1}^{n-1} \text{CU}_{s+1}^{s}(\boldsymbol{\varphi}_l^{(s)}) \left(\bigotimes_{r=1}^{n} U(\boldsymbol{\theta}_l^{(r)}) \right) U(\boldsymbol{x})^{\otimes n} \right)$$

Step 1: Train single-qubit QNN

$\boldsymbol{\theta}* = \arg\min f_{\text{cost}}(\boldsymbol{\theta}^{(1)})$

Step 2: Initialize two-qubit QNN using previous parameters

Initialize $\begin{cases} \boldsymbol{\theta}^{(1)} = \boldsymbol{\theta}* \\ \boldsymbol{\theta}^{(2)} = \boldsymbol{\varphi} = \boldsymbol{0} \end{cases}$

Update parameters $f_{\text{cost}}(\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}, \boldsymbol{\varphi})$
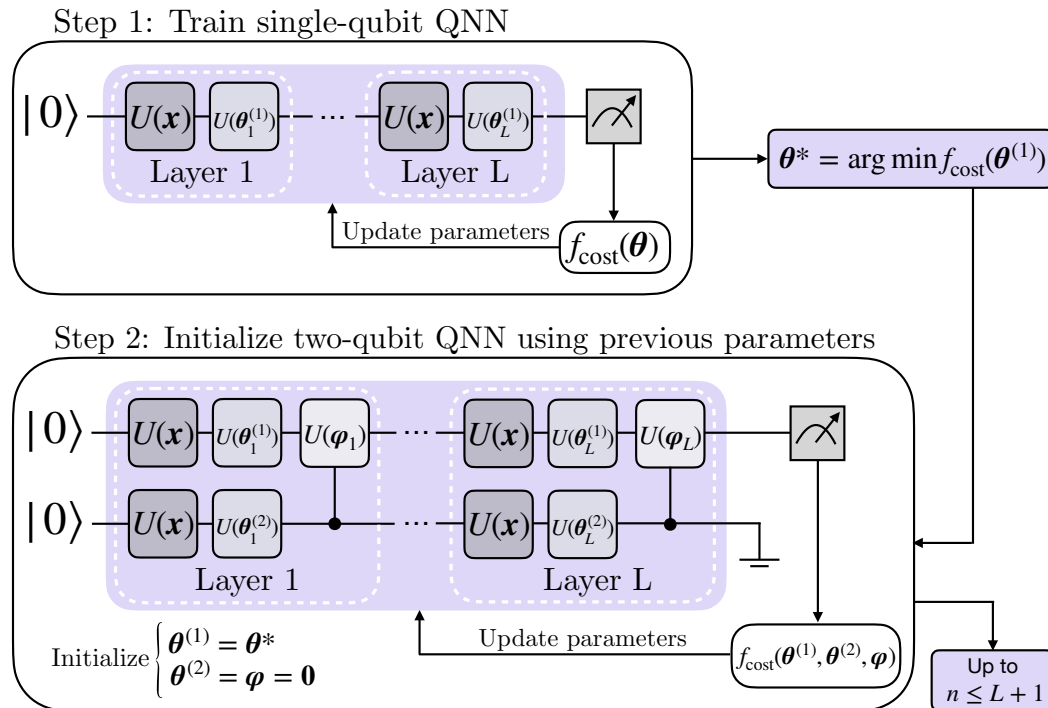
Up to $n \le L + 1$

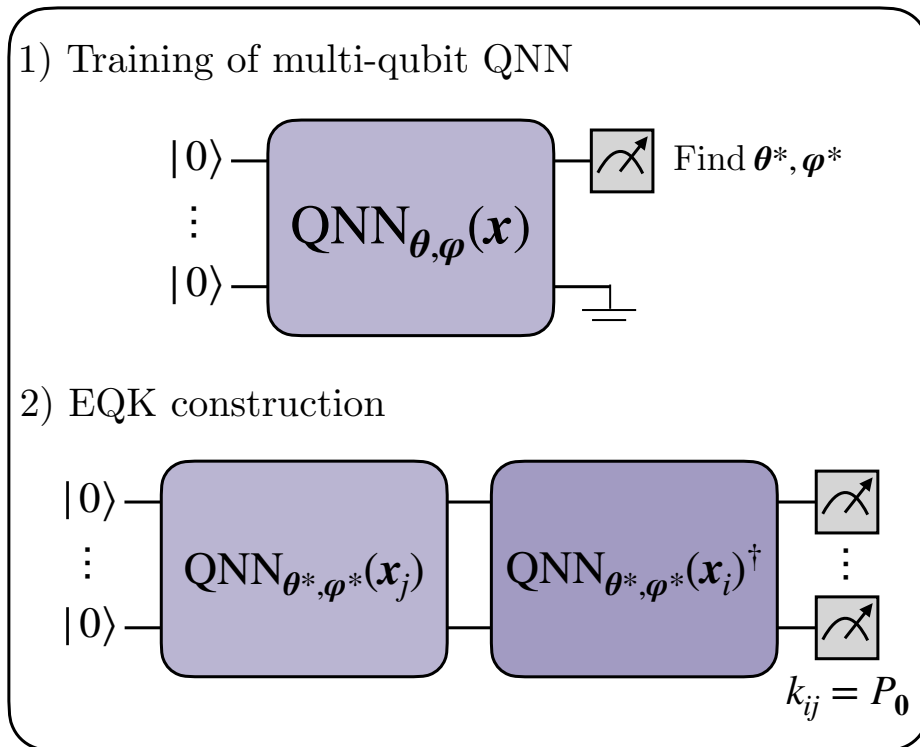Step 1: a single-qubit QNN is trained to obtain optimal parameters.

Step 2: initializing new extra parameters to 0, while the ones of the first qubit are set the same as the trained ones.

......

Scale up the QNN up to n = L + 1 qubits.

Constructing Embedded Quantum Kernel (EQK) from QNN training

n-to-*n* approach: an n-qubit QNN ⟶ EQK of n qubits

n-to-n

1) Training of multi-qubit QNN

$|0\rangle$ — $QNN_{\theta,\varphi}(x)$ — Find $\theta*, \varphi*$

2) EQK construction

$|0\rangle$ — $QNN_{\theta*,\varphi*}(x_j)$ — $QNN_{\theta*,\varphi*}(x_i)^\dagger$ —
$k_{ij} = P_{\mathbf{0}}$

1-to-n

1) Training of single-qubit QNN

$$k_{ij} = |\langle \mathbf{0}| QNN_{\theta*,\varphi*}(x_i)^\dagger QNN_{\theta*,\varphi*}(x_j)|\mathbf{0}\rangle|^2$$

$|0\rangle$ — $U(x)$ $U(\theta_1)$ ... $U(x)$ $U(\theta_L)$ — Find $\theta*$
Layer 1   Layer L

kernel matrix element k_{ij} is defined as the probability of measuring all qubits in the state $|0\rangle$.

2) EQK construction

✓ Scale the QNN as much as possible during training

✓ Utilize the trained feature map to construct an EQK

$|0\rangle$ — $U(x_j)$ $U(\theta_1^*)$ ... $U(x_j)$ $U(\theta_L^*)$ —
$|0\rangle$ — $U(x_j)$ $U(\theta_1^*)$ ... $U(x_j)$ $U(\theta_L^*)$ —
$\mathcal{U}(x_j)$

$\mathcal{U}(x_j)^\dagger$

$k_{ij} = P_{\mathbf{0}}$

Direct EQK

Constructing Embedded Quantum Kernel (EQK) from QNN training

1-to-*n* approach: an 1-qubit QNN ⟶ EQK of n qubits
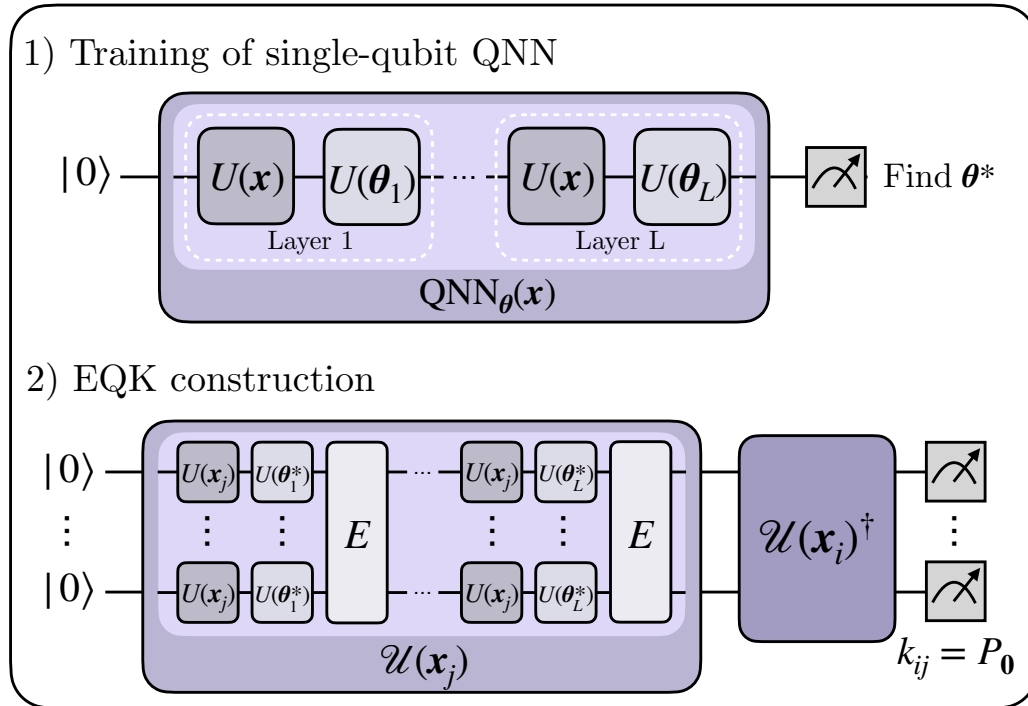
$$k_{ij} = |\langle \mathbf{0} | \prod_{l=1}^{L-1} \left(U(\mathbf{x}_i)^{\dagger \otimes n} \, U(\boldsymbol{\theta}_l^*)^{\dagger \otimes n} \, E \right) U(\mathbf{x}_i)^{\dagger \otimes n}$$

$$U(\mathbf{x}_j)^{\otimes n} \left(\prod_{l=1}^{L-1} E \, U(\boldsymbol{\theta}_l^*) \, U(\mathbf{x}_j) \right) |\mathbf{0}\rangle|^2,$$

E denotes an entangling operation

✓ the training is conducted in a single-qubit QNN

  and does not explicitly consider entanglement.

✓ combine both the n-to-n and 1-to-n architectures

  to generalize it to n-to-m*n.

arXiv:2401.04642

Image classification with a few qubits

Select Cities and Imagery

Data Input

California Solar Initiative Data
Known solar installations by city

USGS Image Data
Aerial Imagery
Geospatial metadata

Annotate Solar Arrays

Annotator Pool
A B C D E F G
Two annotators

Merge Annotations

Annotators A and B produce different polygons
A   B

Compile Data

Output a fully annotated set of 19,863 solar array polygons

classifying satellite photos with/without PV panel

California

Four CA cities selected
yes → Most solar arrays? ← yes

Resolution ≤ 0.3 meters?
yes ↓
Date ≥ 2013?

○ City
○ Selected City

1) Reduce dimensionality using PCA to 2 features:

Image → Classical preprocessing
Dimensionality reduction to 2 or 3 features
→ Dataset $X = \{\vec{x}_i \in \mathbb{R}^{2,3}\}_{i=1}^{M}$

$|0\rangle - QNN_{\vec{\theta}}(X)$ — Single-qubit QNN → 1-to-n NQK

$|0\rangle \vdots |0\rangle - QNN^{(n)}_{\vec{\theta},\vec{\varphi}}(X)$ — n-qubit QNN → n-to-n NQK

Resolution Ir
Oxnard
Modesto

hmark

Accuracy

acy = 85

arXiv:2409.20356   Collaborating with GMV

X   pablojesus.rodriguez@ehu.eus

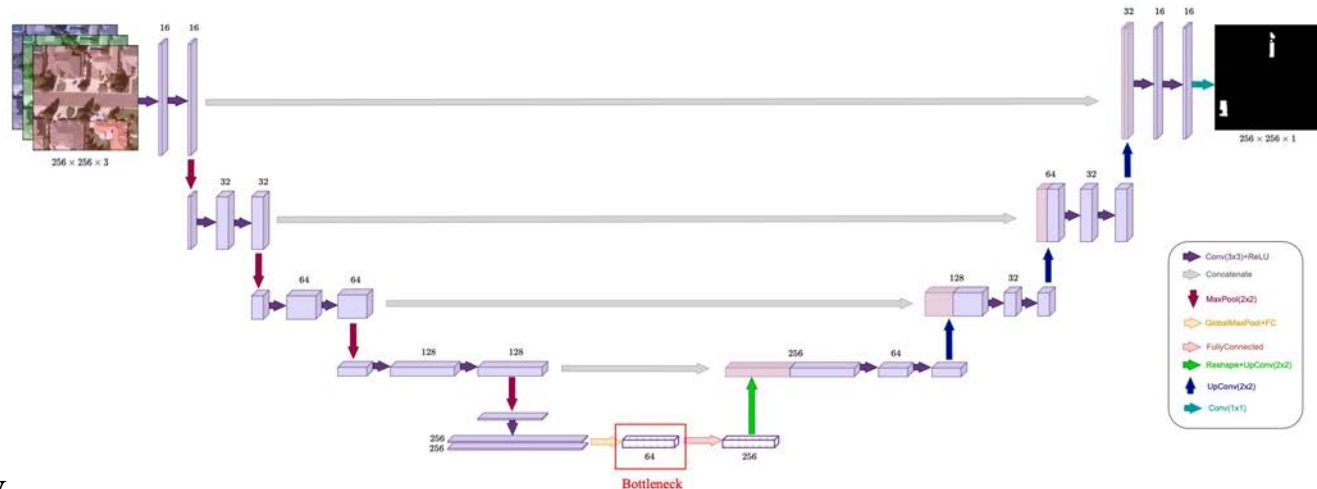# Image classification with a few qubits

classifying satellite phots with/without PV panel
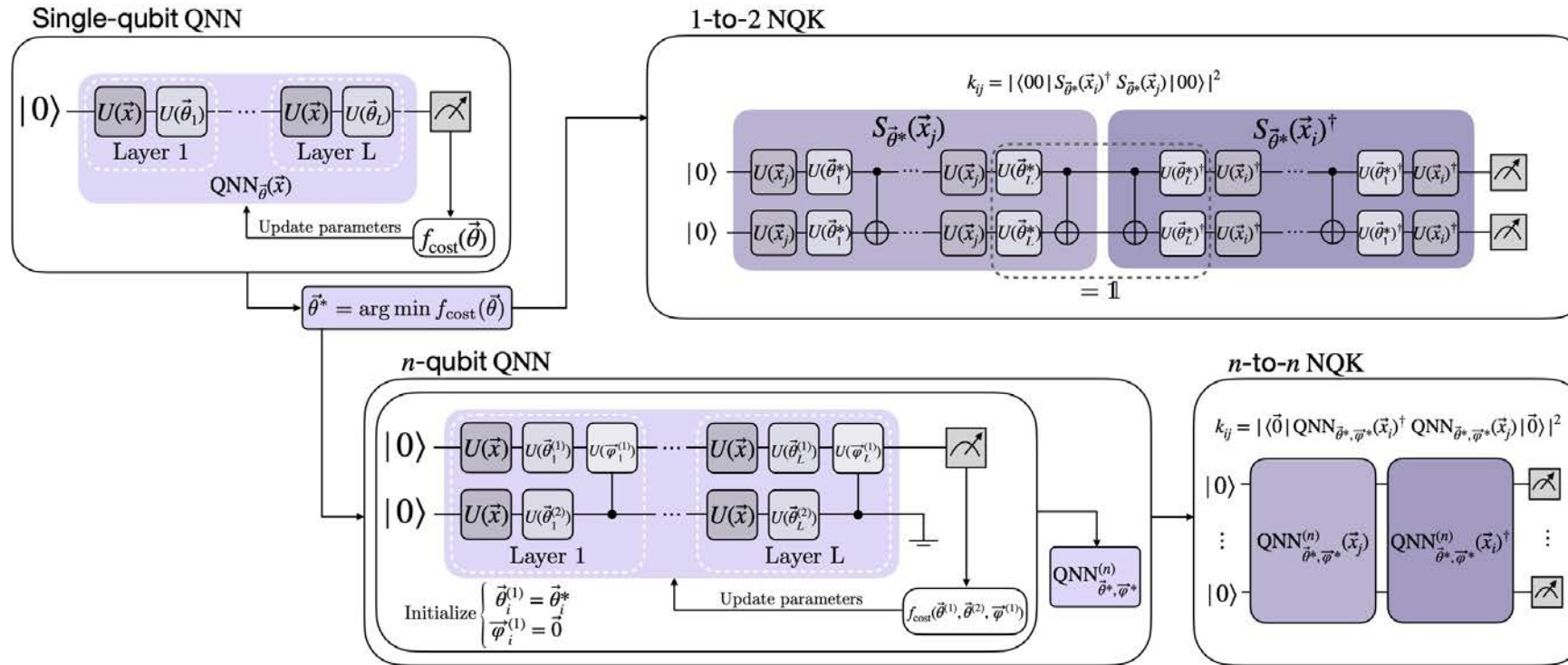
1) Reduce dime

2) Binary classi

arXiv:2409.20356   Collaborating with GMV

Image classification with a few qubits

classifying satellite phots with/without PV panel

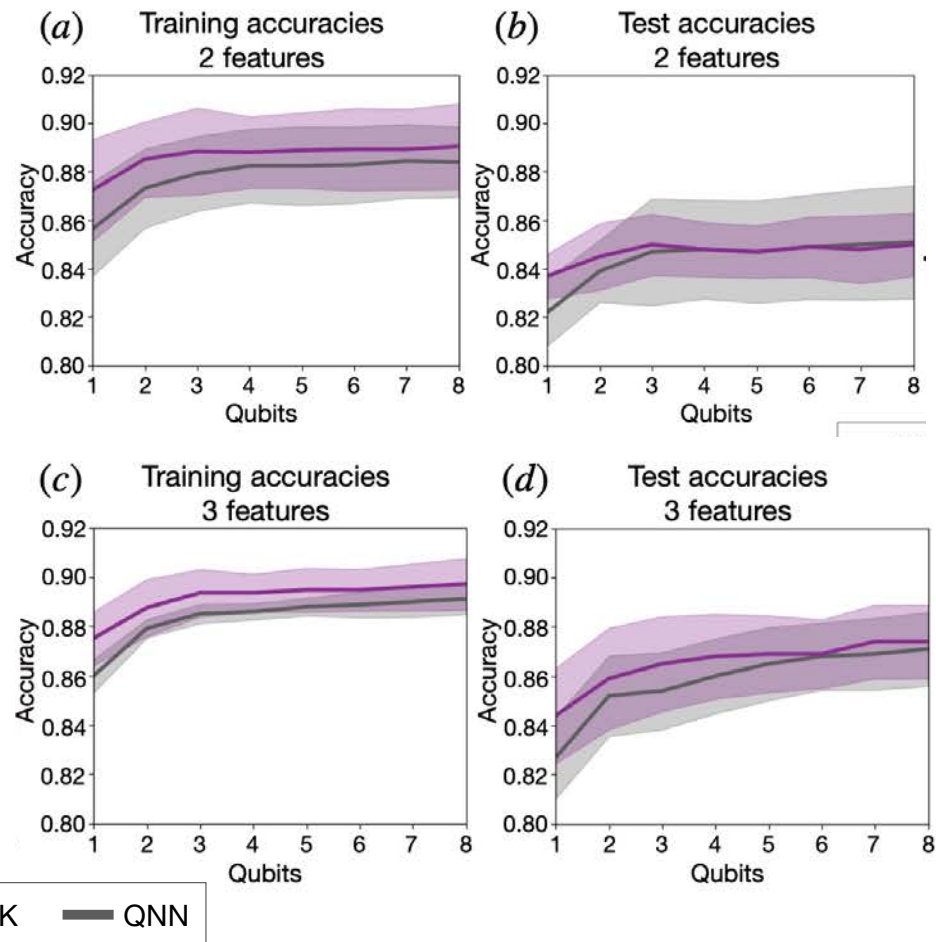Data Input

California Solar Initiative Data
Known solar installations by city

USGS Image Data
Aerial Imagery
Geospatial metadata

Resolution ≤ 0.3 meters?
yes

Annotator Pool
A B C D E F G

outline each solar array in each image

Annotators A and B produce different polygons

Output a fully annotated set of 19,863 solar array polygons

Training   Test

(a) Training accuracies 2 features

(b) Test accuracies 2 features

(c) Training accuracies 3 features

(d) Test accuracies 3 features

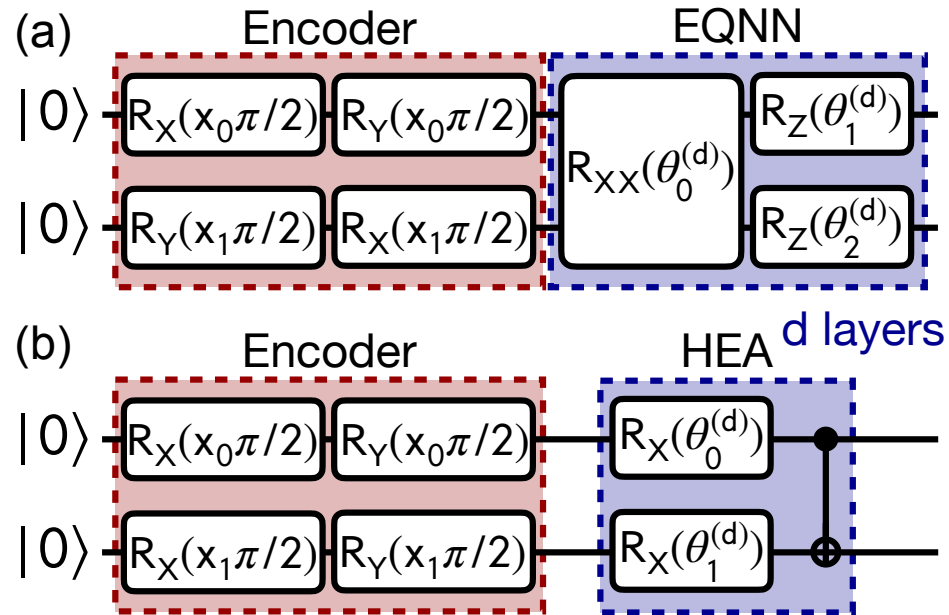| Model	n components	Train	Test
SVC	$p = 2$	86.1±0.3	86.2±2.5
	$p = 3$	88.0±0.3	88.1±2.7
	$p = 45$	89.4±0.3	89.4±2.5
Random Forest	$p = 2$	91.1±0.3	87.2±2.7
	$p = 3$	93.0±0.3	87.6±2.0
	$p = 10$	96.9±0.2	88.3±2.8
to-2 NQK	$p = 2$	86.6±0.3	86.5±2.5
	$p = 3$	81.7±1.8	81.1±2.9
to-3 NQK	$p = 2$	86.6±0.3	86.5±2.6
	$p = 3$	85.9±0.6	85.6±2.7
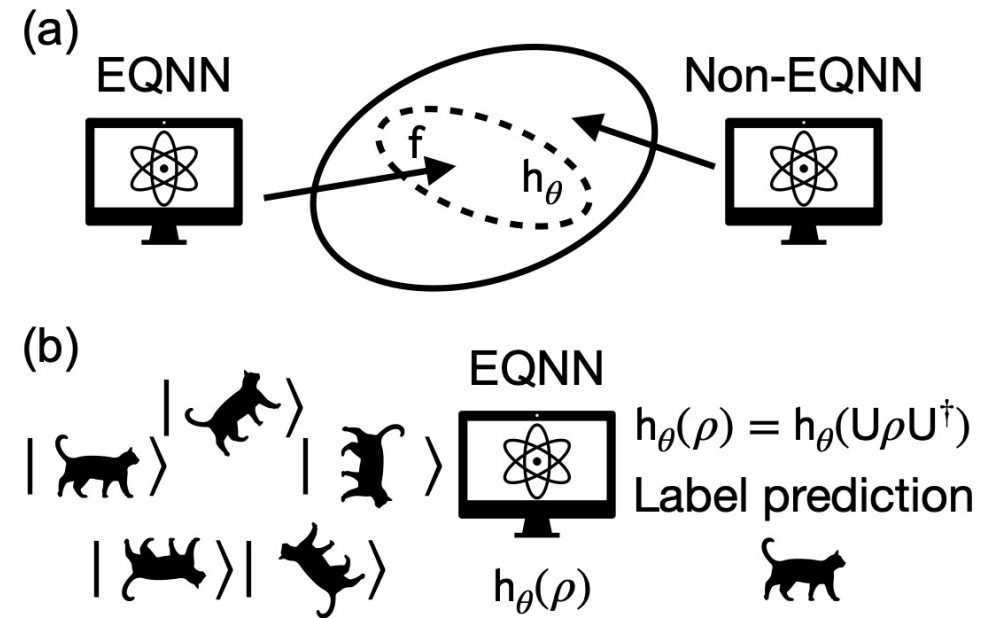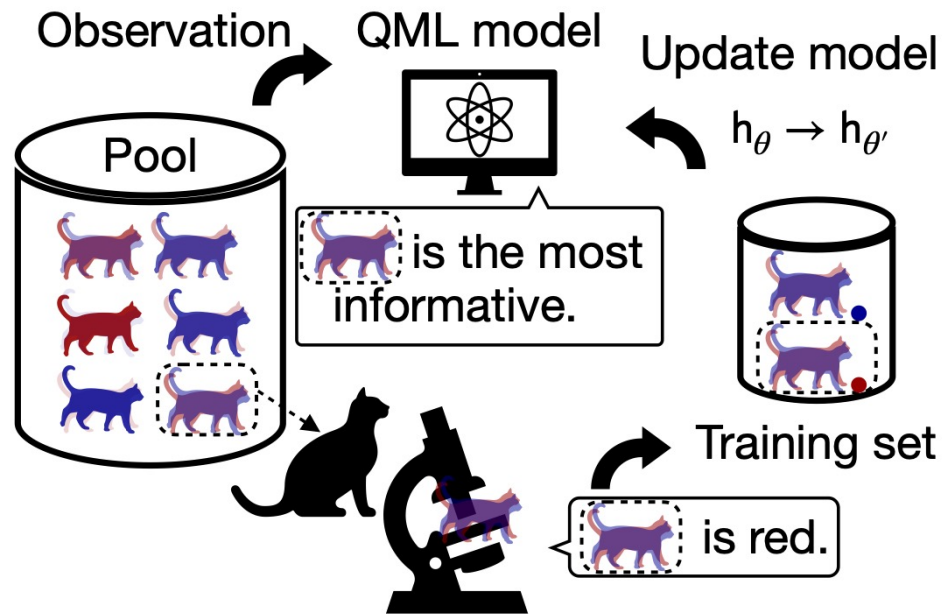
NQK   QNN

Number of Qubits

arXiv:2409.20356

pablojesus.rodriguez@ehu.eus

Universidad del Pais Vasco / Euskal Herriko Unibertsitatea

EHU QC
EHU Quantum Center

Quantum Active Learning

arXiv:2405.18230

(b)

ing

)

$\langle \hat{y}_i \rangle = (\langle \hat{O} \rangle + 1)/2$

$\hat{O} = \langle \hat{Z}_0 + \hat{Z}_1 \rangle /2 \in [-1, 1]$

Data set

(a)

$|0\rangle$

$|0\rangle$

(b)

$|0\rangle$

$|0\rangle$

$\theta_1^{(d)}$

$\theta_2^{(d)}$

layers

$(1 - \langle \hat{y}_i \rangle)]$
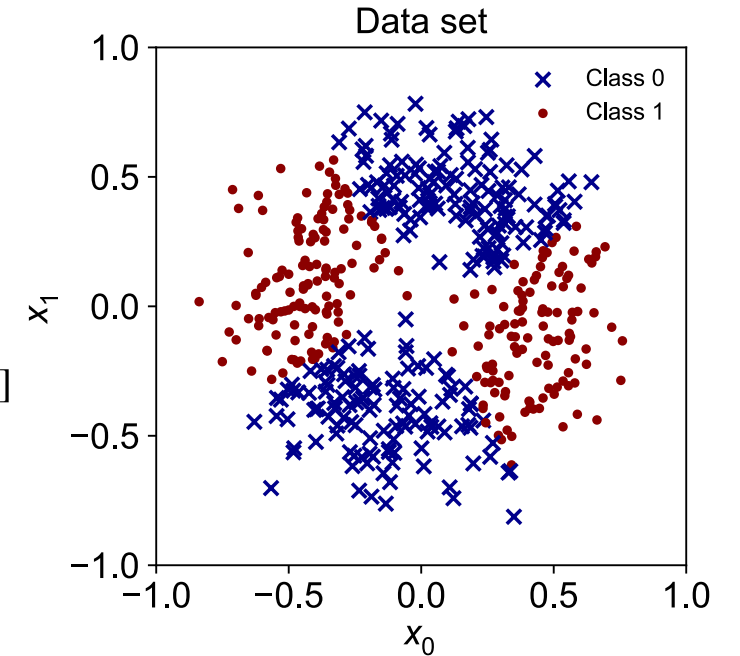
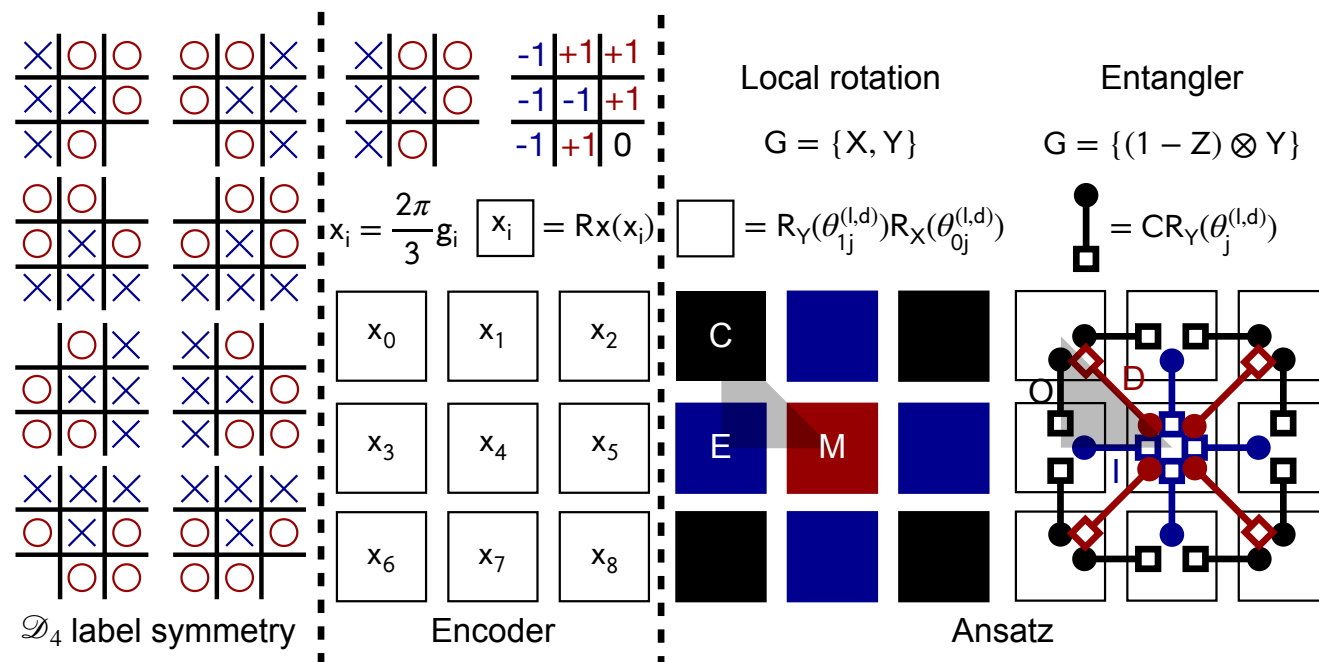training accuracy 91.63 ± 4.24% (EQNN) and 77.30 ± 2.06% (HEA)

Using a uncertainty sampling strategy

Test accuracies 95% (EQNN) and 69% (HEA)

arXiv:2405.18230

33

Dihedral group $\mathcal{D}_4$ Symmetry

tic-tac-toe board

$$\mathbf{g}_i \in \mathcal{U}$$

Achieving accuracy 76.89 ± 6.54% after training 127.58 ± 61.20 epochs.

$$\hat{O}_\circ = \frac{1}{4}\sum_C Z_j = \frac{1}{4}[Z_0 + Z_2 + Z_6 + Z_8],$$

$$\hat{O}_- = Z_M = Z_4,$$

$$\hat{O}_\times = \frac{1}{4}\sum_E Z_j = \frac{1}{4}[Z_1 + Z_3 + Z_5 + Z_7]$$

$$\hat{\mathbf{y}} = [\langle\hat{O}_\circ\rangle, \langle\hat{O}_-\rangle, \langle\hat{O}_\times\rangle]$$

Uncertainty Sampling by entropy

$$\mathbf{g}_{ES} = \underset{\mathbf{g}}{\arg\max}\left[-\sum_j P_\theta(y_j|\mathbf{g})\log P_\theta(y_j|\mathbf{g})\right]$$

$$P_{y_i} = \frac{\exp(y_i)}{\exp(\langle\hat{O}_\circ\rangle) + \exp(\langle\hat{O}_-\rangle) + \exp(\langle\hat{O}_\times\rangle)}$$

Local rotation   **Entangler**

$G = \{X, Y\}$ $G = \{(1 - Z) \otimes Y\}$

$x_i = \frac{2\pi}{3}g_i$ $\boxed{x_i} = Rx(x_i)$ $\square = R_Y(\theta_{1j}^{(l,d)})R_X(\theta_{0j}^{(l,d)})$ $= CR_Y(\theta_j^{(l,d)})$

-1	+1	+1
-1	-1	+1
-1	+1	0

x_0	x_1	x_2
x_3	x_4	x_5
x_6	x_7	x_8

C   E   M   D   O   I

Summary on quantum machine learning

✓ Training an embedding quantum kernel (EQK) from data re-uploading QNNs for classification.

✓ Constructing the kernel matrix only once, offering improved efficiency.

✓ Two specific cases:
n-to-n: the output of the training of an n-qubit QNN directly as a feature map.
1-to-n: creating EQKs with entanglement from the training of a single-qubit QNN.

✓ EQK with classical PCA for classification of satellite images with accuracy approaching 90%.

✓ QAL can achieve performance comparable to that on fully labeled datasets

✓ Intriguing to investigate the potential of alternative QML models.

Thanks for your attention!