Variational Learning for Quantum Artificial Neural Networks

Stefano Mangini
Quantum Information Theory Group QUit
University of Pavia, Italy

Summer School, *Machine Learning for Quantum Physics and Chemistry*
28/08/2021

stefano.mangini01@universitadipavia.it
@stfnmangini
Stefano Mangini

UNIVERSITÀ DI PAVIA

Introduction

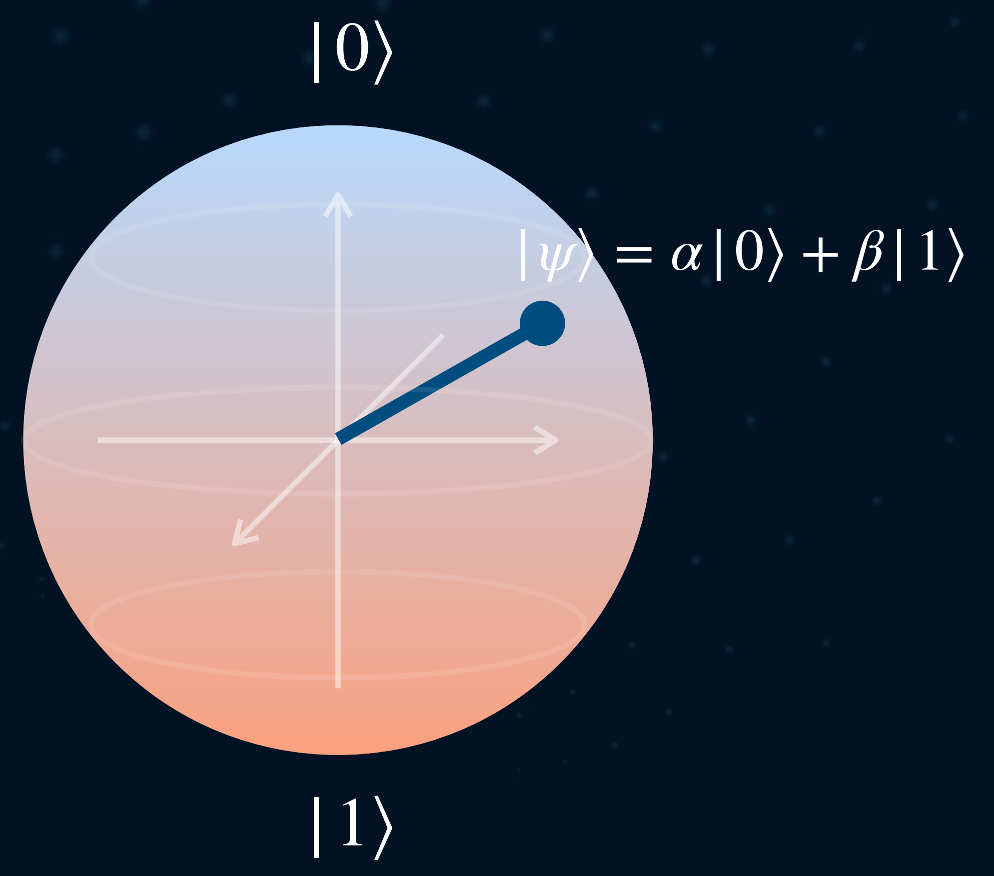
Neuron Model

Variational Learning

Results

Quantum Machine Learning

Quantum computing



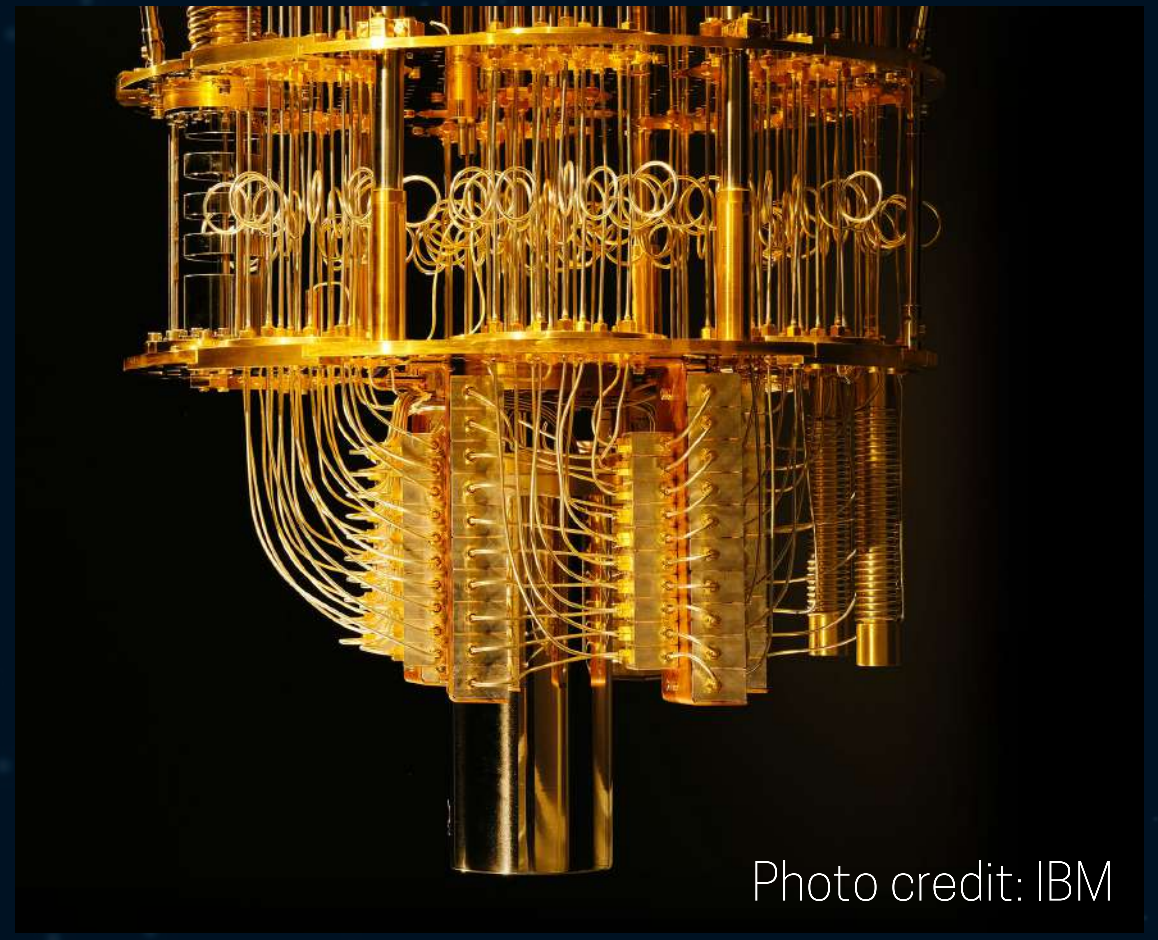
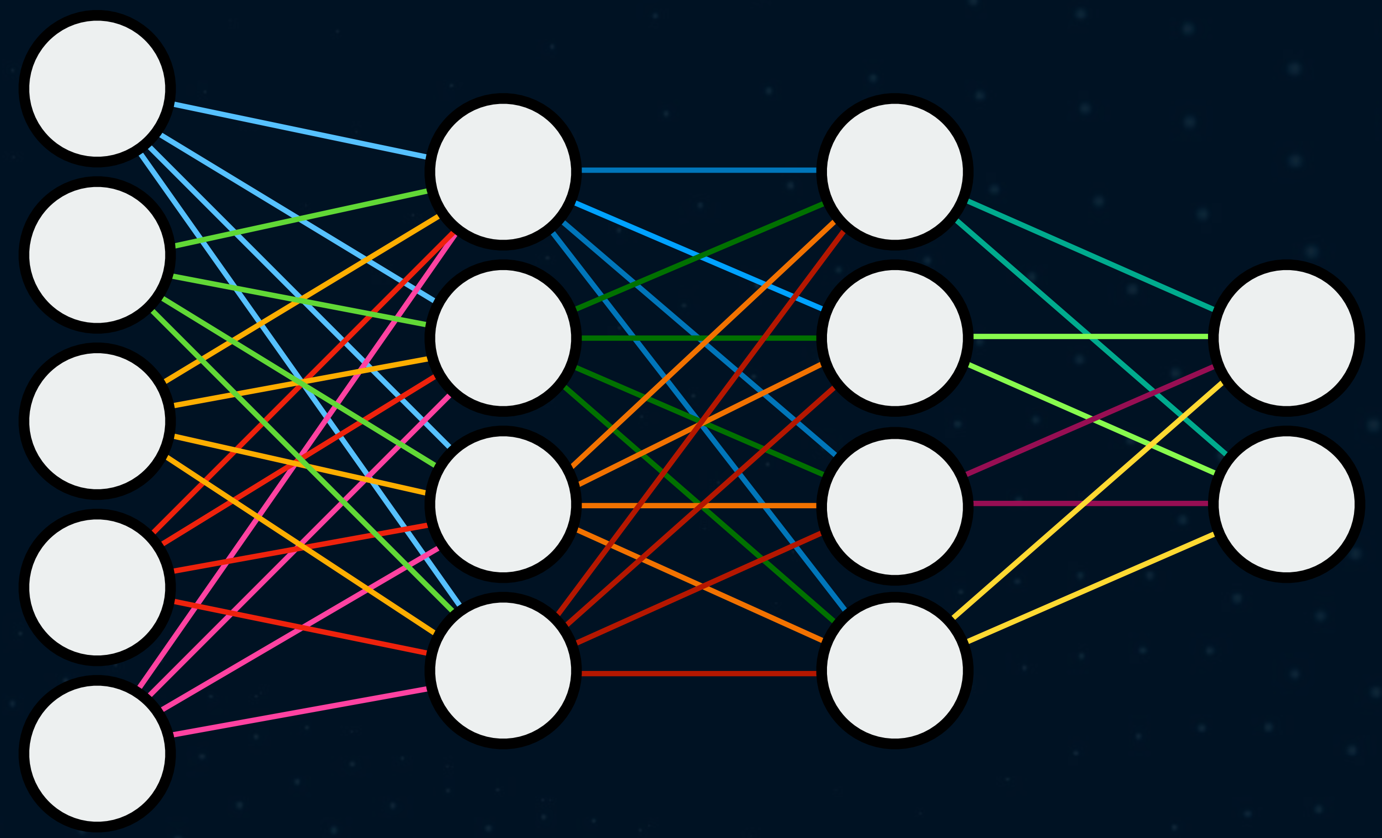
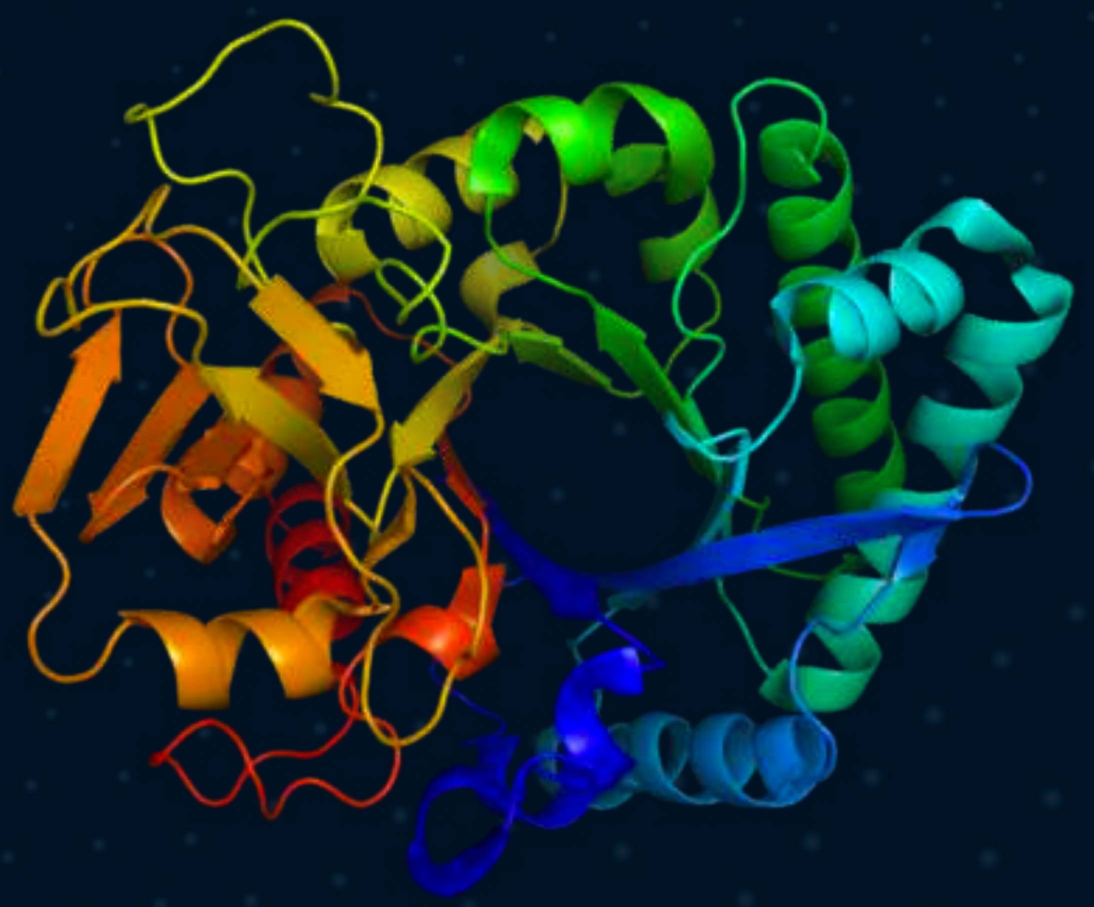


Photo credit: IBM

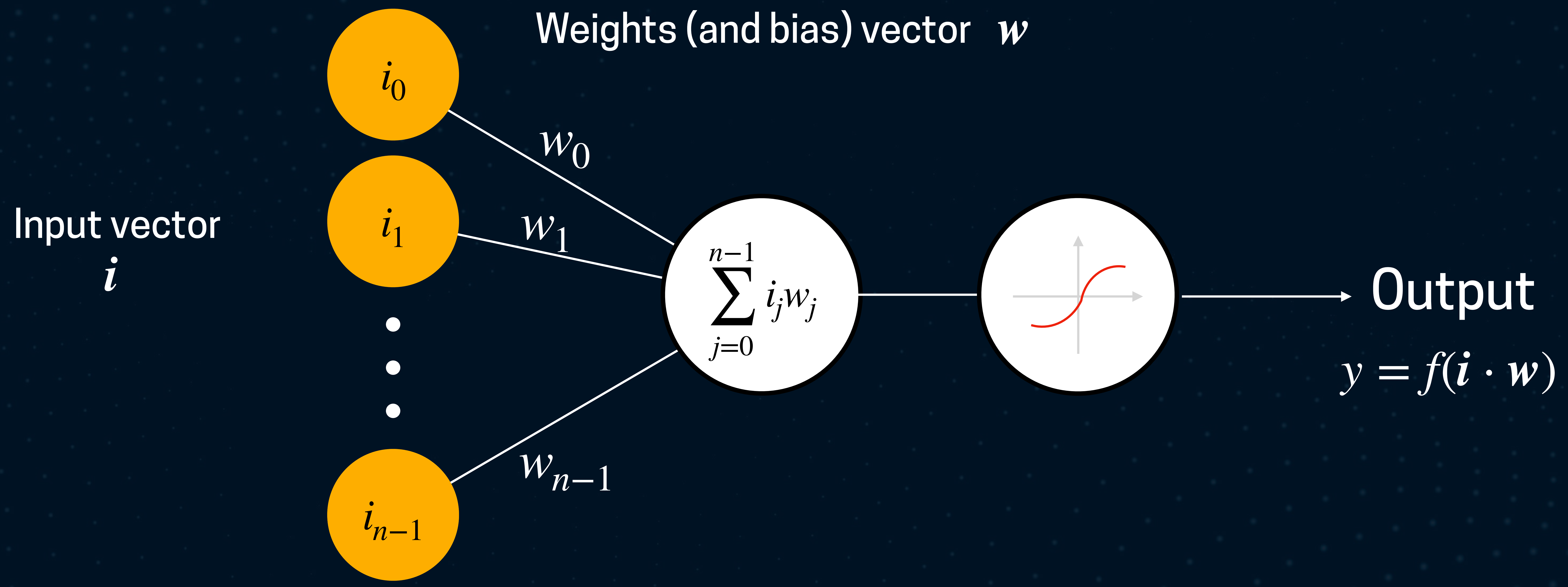
Photo credit: Google

Machine Learning

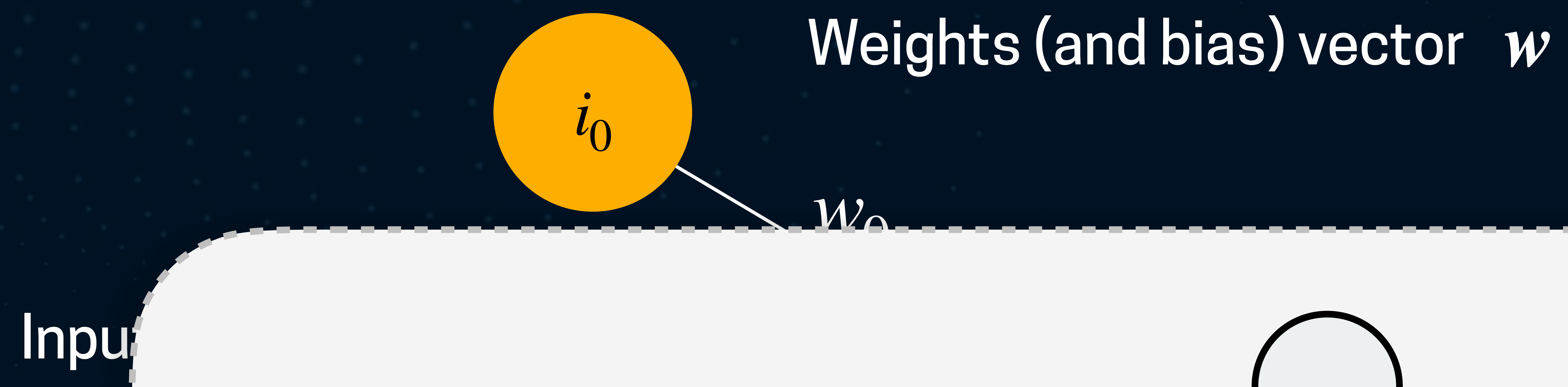




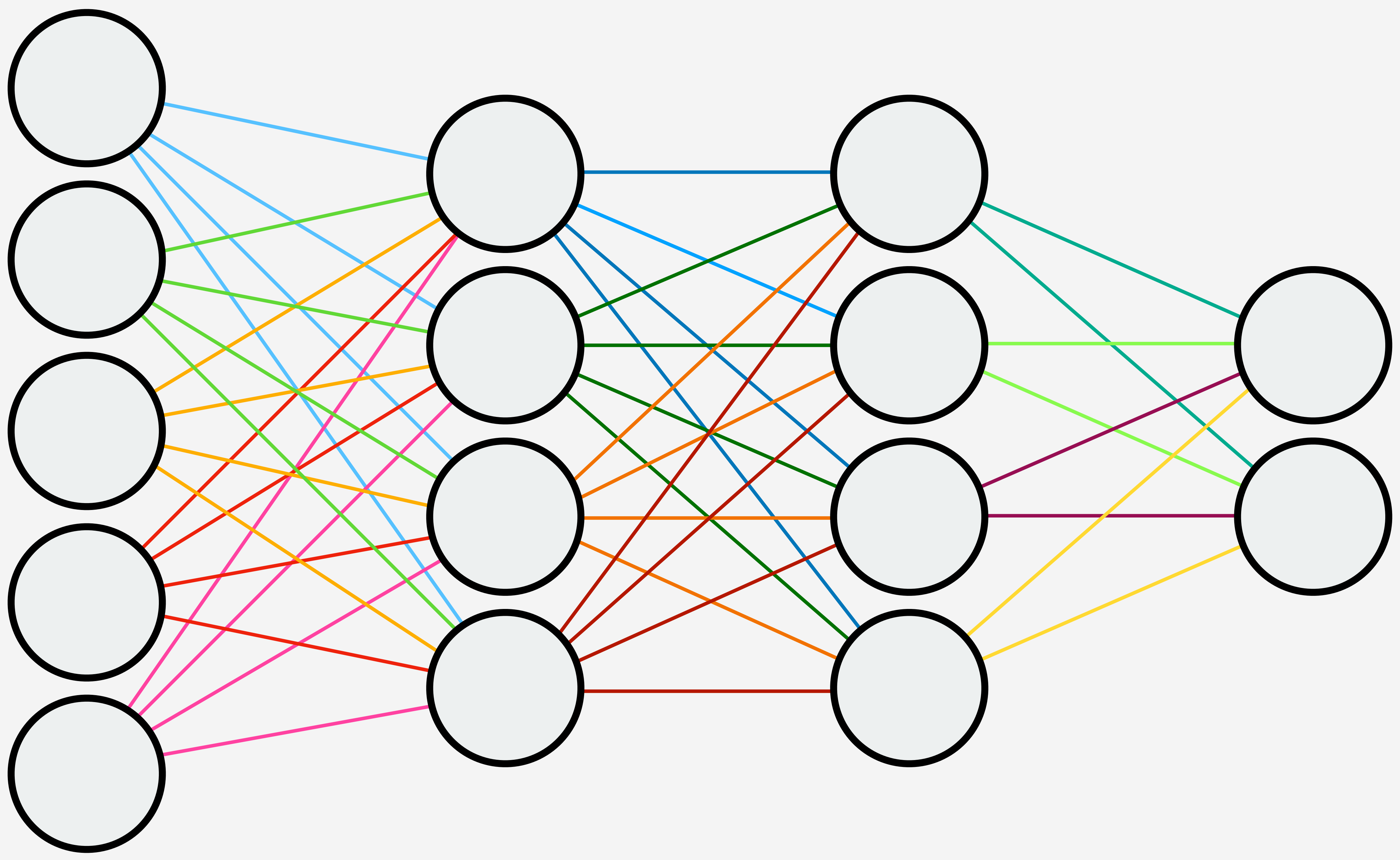
Classical perceptrons



Classical perceptrons

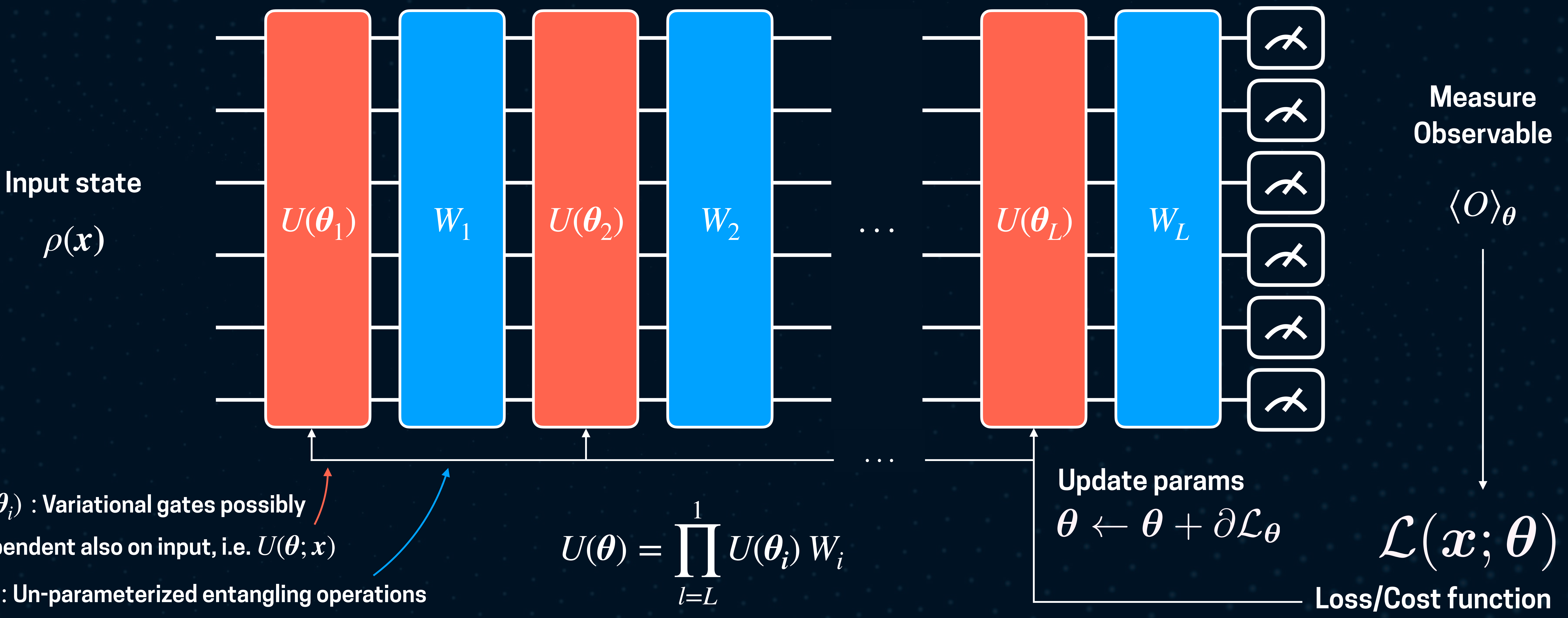


Artificial feedforward neural network



PQCs as QNNs

Parameterized Quantum Circuits (PQC) are often referred to as *Quantum Neural Networks* (QNN)



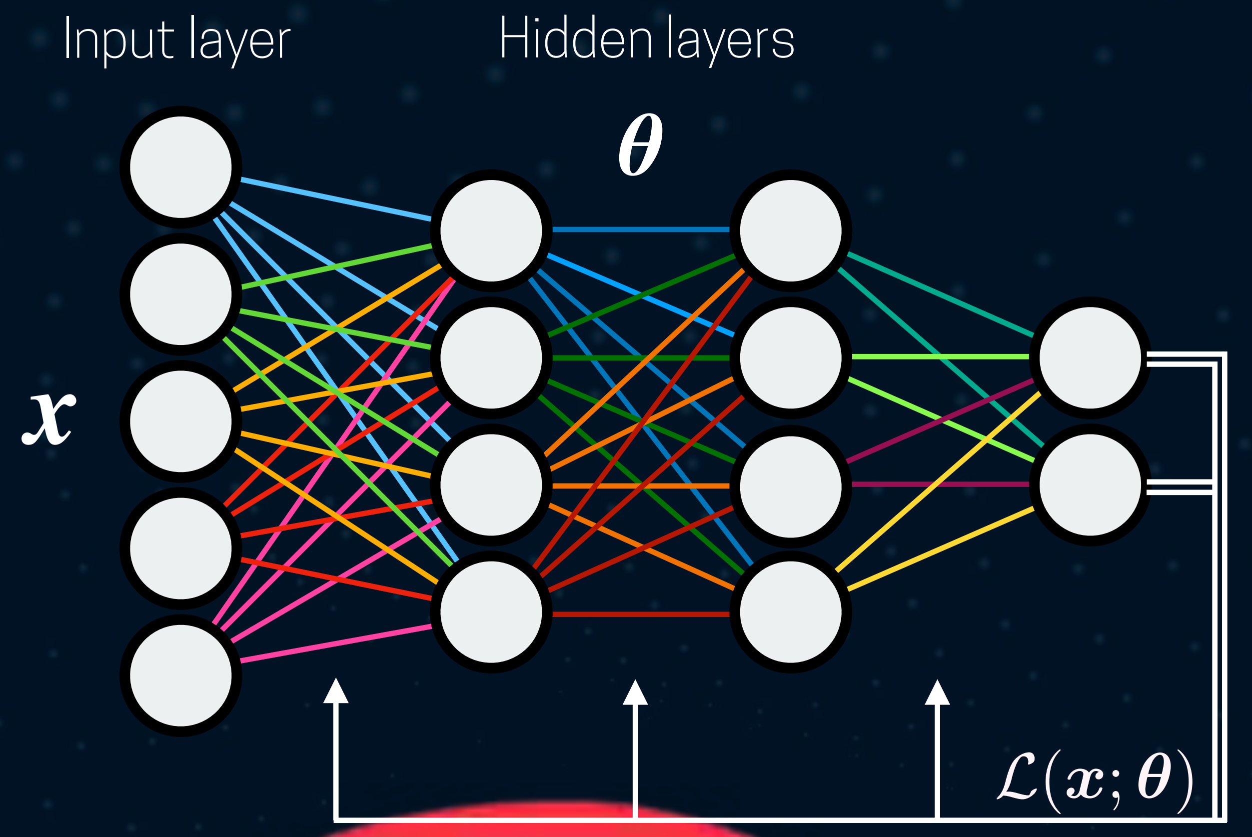
$U(\boldsymbol{\theta}_i)$: Variational gates possibly dependent also on input, i.e. $U(\boldsymbol{\theta}; \boldsymbol{x})$

W_i : Un-parameterized entangling operations

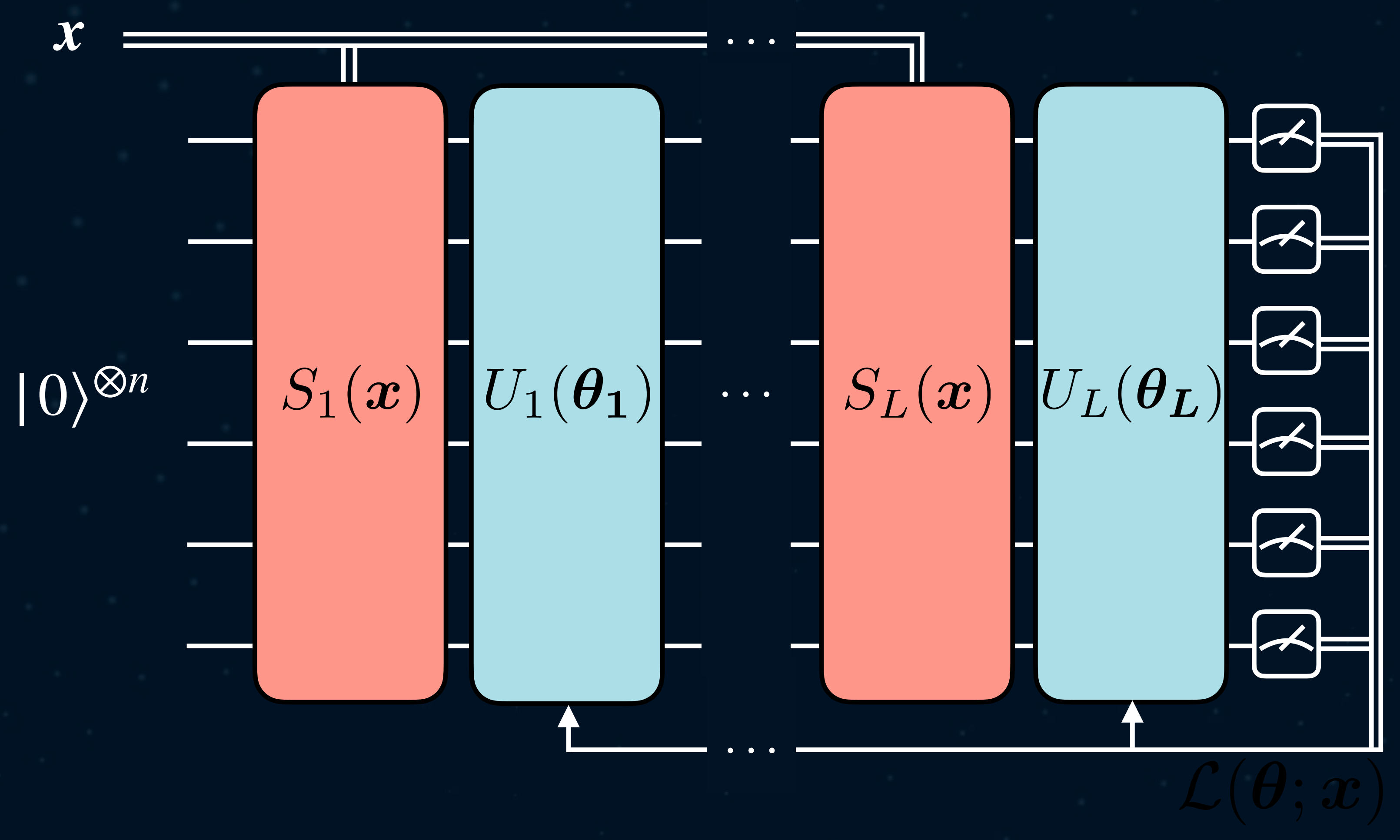
$$U(\boldsymbol{\theta}) = \prod_{l=L}^{1} U(\boldsymbol{\theta}_i)\, W_i$$

Various QNN/QML models [1]

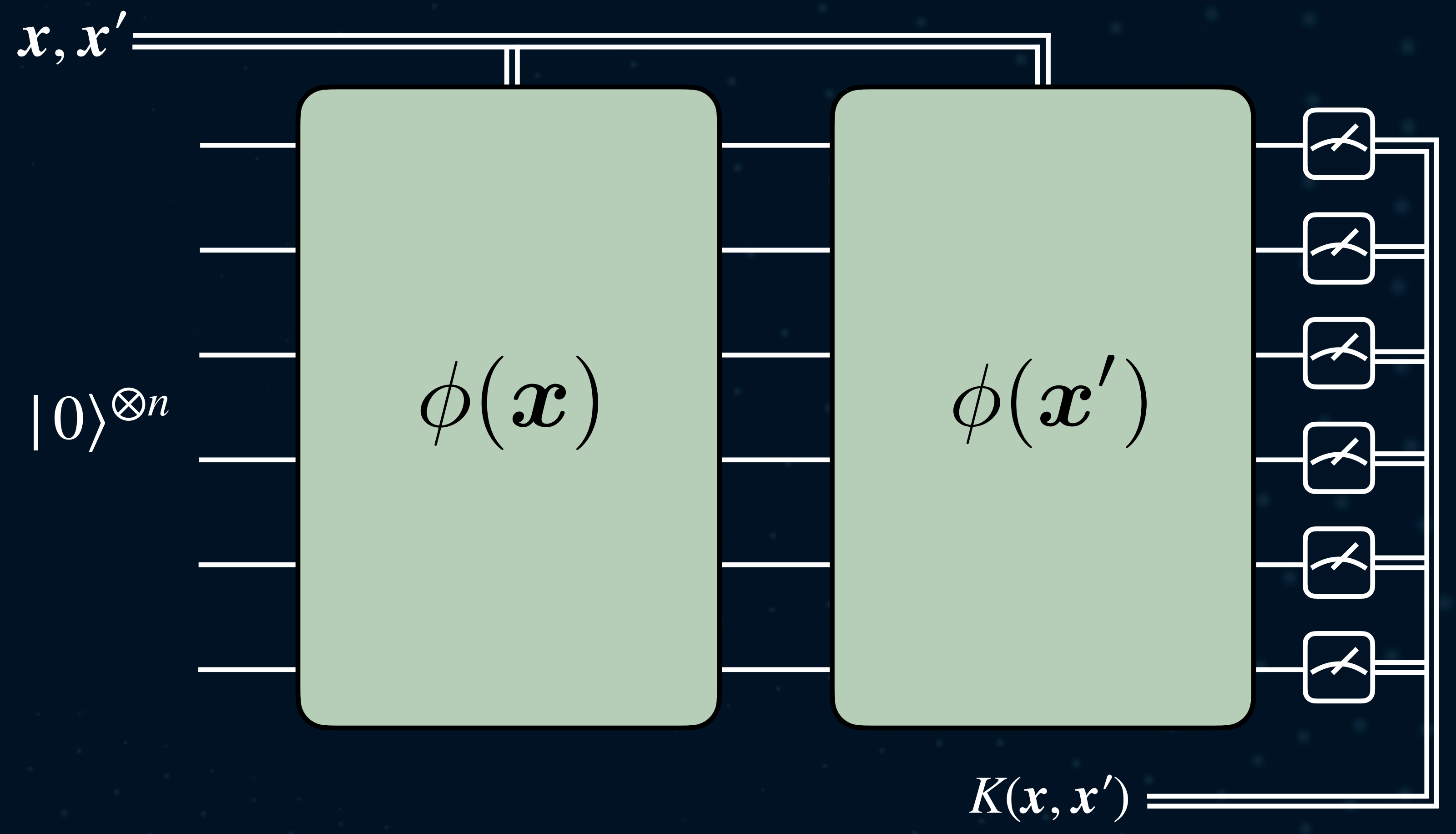
Classical Neural Networks



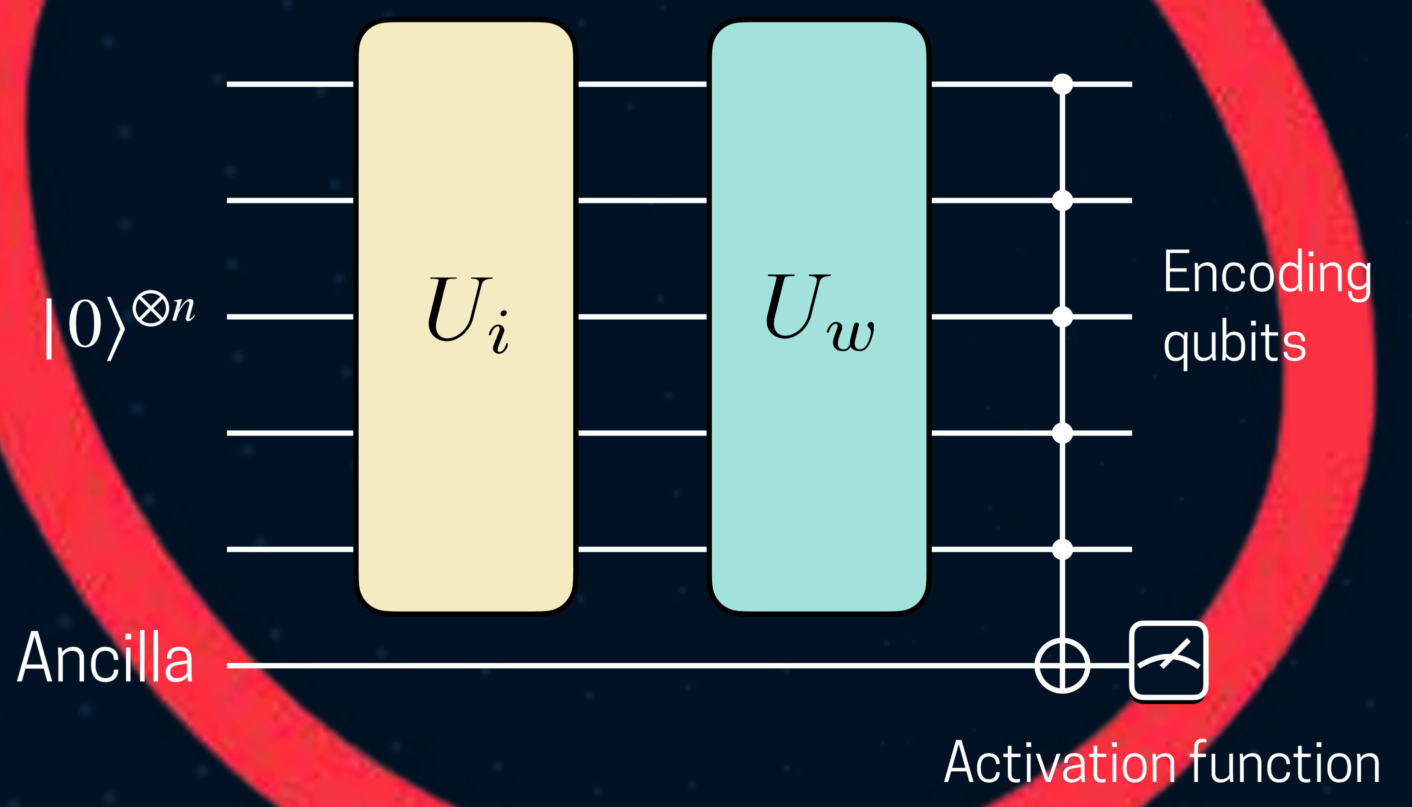
Quantum Neural Networks



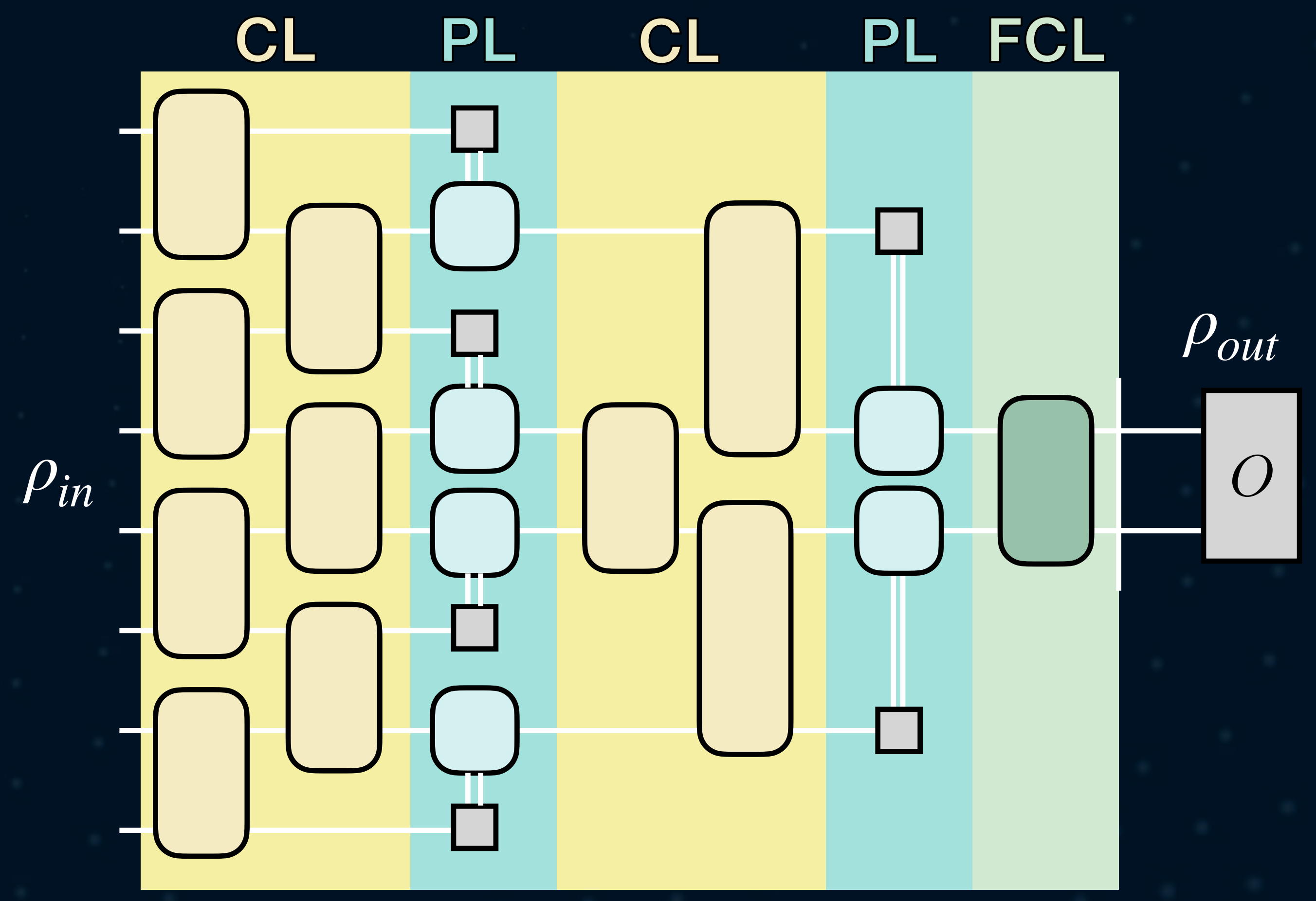
Quantum Kernel Methods



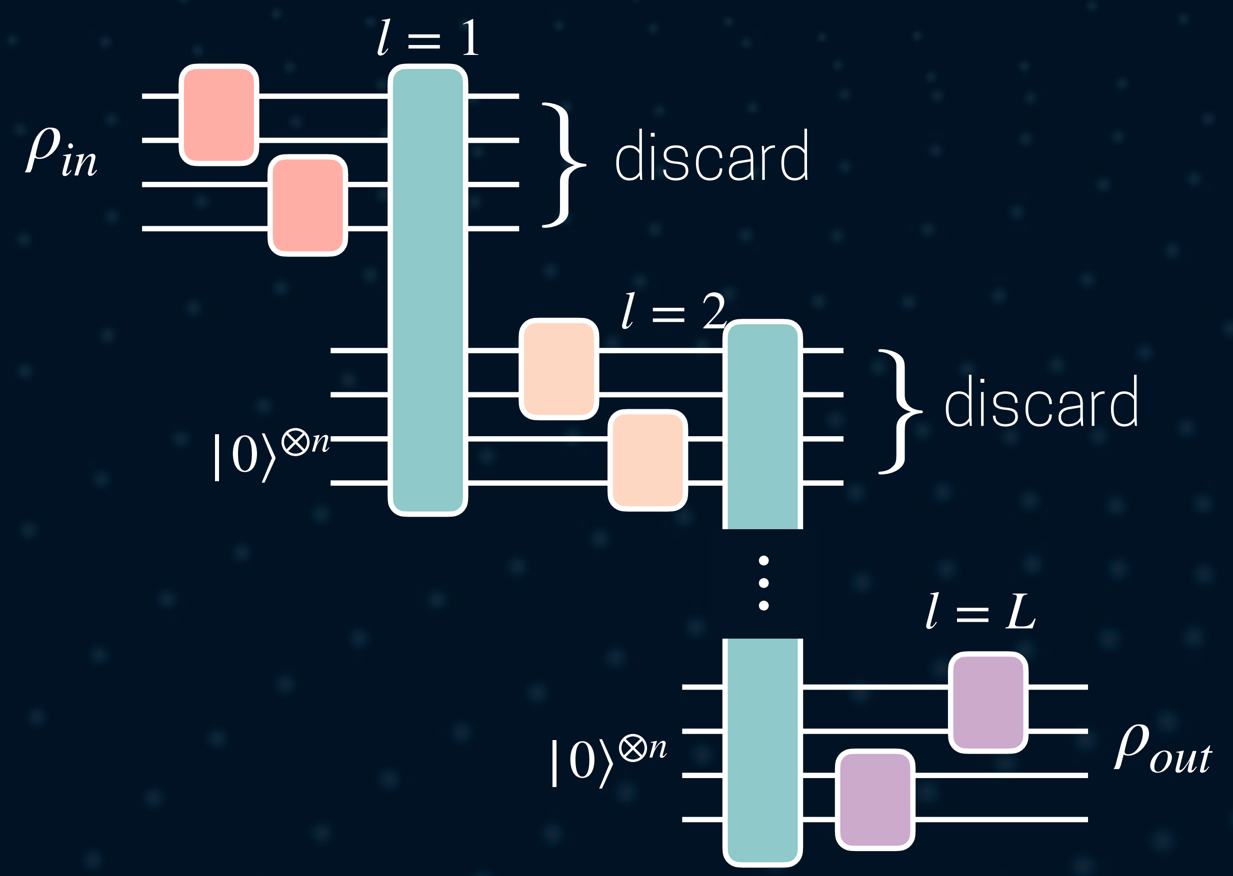
Quantum Perceptrons



Quantum Convolutional Neural Networks



Quantum Dissipative Neural Networks



[1] S. Mangini *et al.*, "Quantum computing models for artificial neural networks", *EPL (Europhysics Letters)* **2**, 1 (2021).

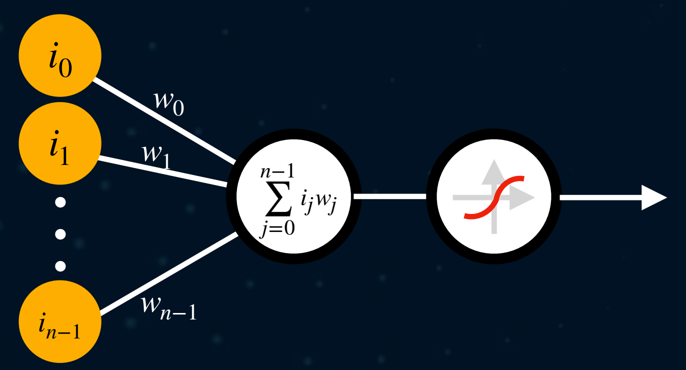
McCulloch-Pitts Quantum Neuron [2]

Discrete inputs and weights

$$\boldsymbol{i} = (i_0, i_1, \ldots, i_{n-1})$$

$$\boldsymbol{w} = (w_0, w_1, \ldots, w_{n-1})$$

$$i_j, w_j \in \{-1,1\}$$



$$|\psi_i\rangle = \frac{1}{\sqrt{2^N}} \sum_{j=0}^{2^N-1} i_j |j\rangle$$

$$|\psi_w\rangle = \frac{1}{\sqrt{2^N}} \sum_{j=0}^{2^N-1} w_j |j\rangle$$

Encode the n-bit long input and weight vector in the amplitudes ± 1 of a balanced superposition of the computational basis states of $N = \log_2 n$ qubits

Real Equally Weighted (REW) states

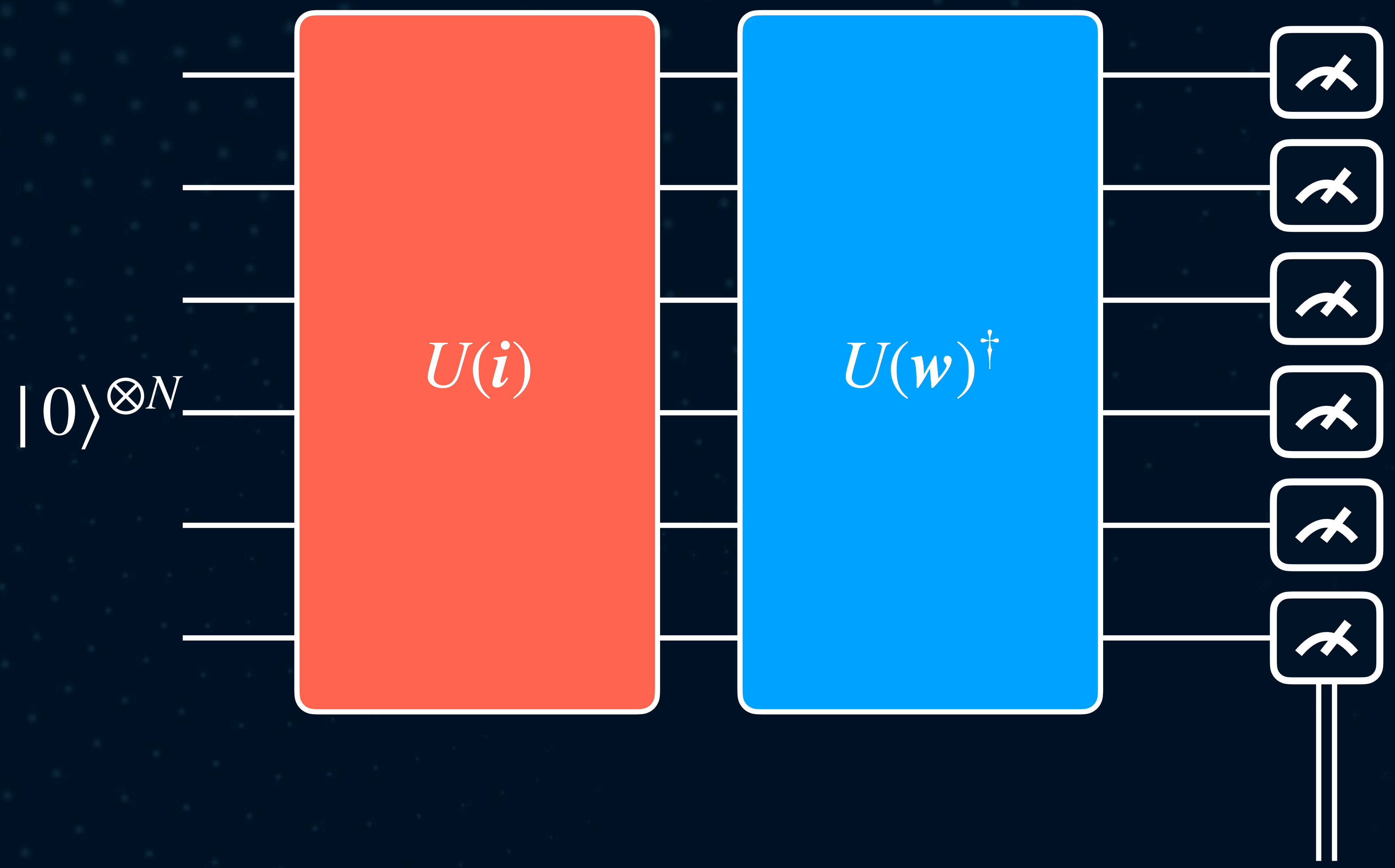
So that…

$$\langle \psi_i | \psi_w \rangle = \sum_{j,k=0}^{2^N-1} i_j w_k \langle j | k \rangle = \sum_{j=0}^{2^N-1} i_j w_j$$

…inner product! Now the activation function!

[2] F. Tacchino *et al.*, "An artificial neuron implemented on an actual quantum processor", *npj Quantum Information* **5**, 26 (2019).

Compute-Uncompute method



of ‘000..0’ measurements = $|\langle\psi_w|\psi_i\rangle|^2$

Non-linear activation function induced by measurement!

Be $U(\cdot)$ the variational unitary preparing the input and weight quantum states

$$U(i)|0\rangle^{\otimes N} = |\psi_i\rangle$$
$$U(w)|0\rangle^{\otimes N} = |\psi_w\rangle$$

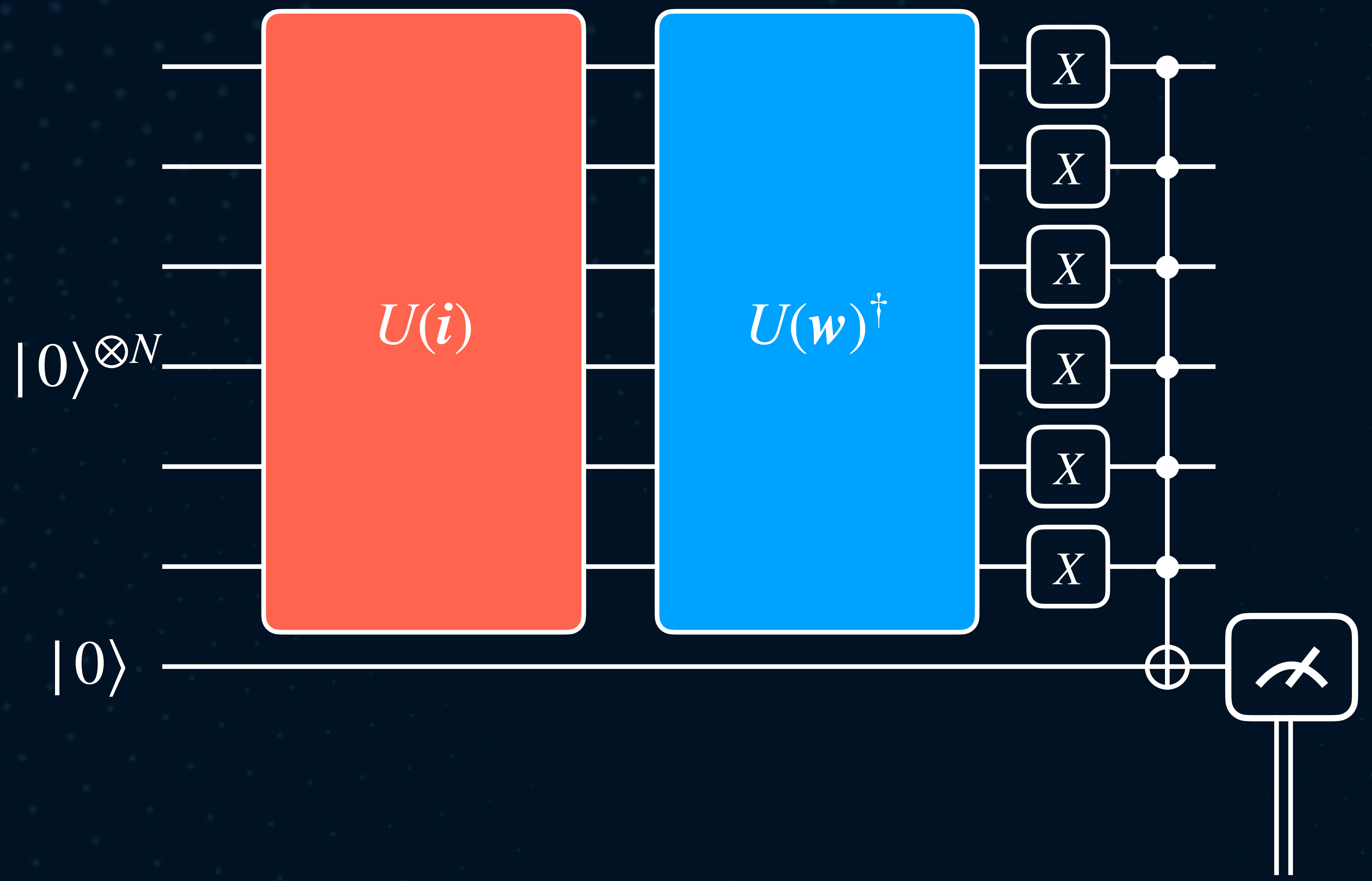
then

$$|\Phi\rangle = U(w)^\dagger U(i)|0\rangle^{\otimes N}$$

its projection on the $|0\rangle^{\otimes N}$ state

$$\langle 0|\Phi\rangle = \underbrace{{}^{\otimes N}\langle 0|U(w)^\dagger}_{\langle\psi_w|}\underbrace{U(i)|0\rangle^{\otimes N}}_{|\psi_i\rangle}$$
$$= \langle\psi_w|\psi_i\rangle$$

Compute-Uncompute with a twist



of '1' measurements = $|\langle\psi_w|\psi_i\rangle|^2$

Adding:

- ancilla qubit
- layer of X
- multi controlled NOT $C^N X$

(or control on $|0\rangle$)

In this way the desired inner product is loaded on the state of the ancilla qubit. Easier to transfer if thinking of a coherent feedforward neural network.

Pattern Recognition

$$\boldsymbol{w} = \begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_{2^N-1} \end{pmatrix} \qquad \boldsymbol{i} = \begin{pmatrix} i_0 \\ i_1 \\ \vdots \\ i_{2^N-1} \end{pmatrix}$$

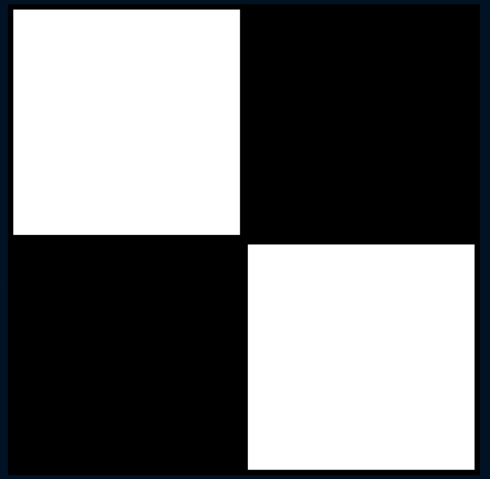
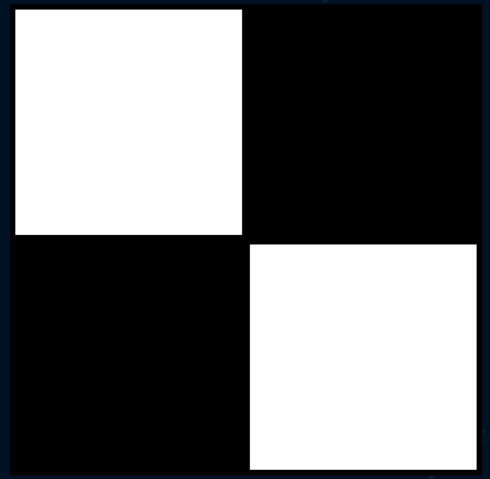
$N = 2$

$\vec{i} = (i_0, i_1, i_2, i_3)$

i_0	i_1
i_2	i_3

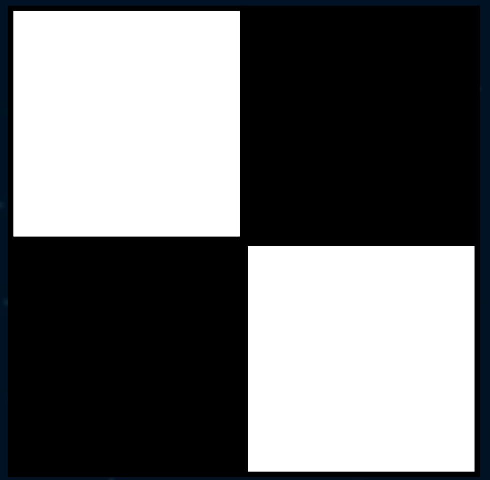
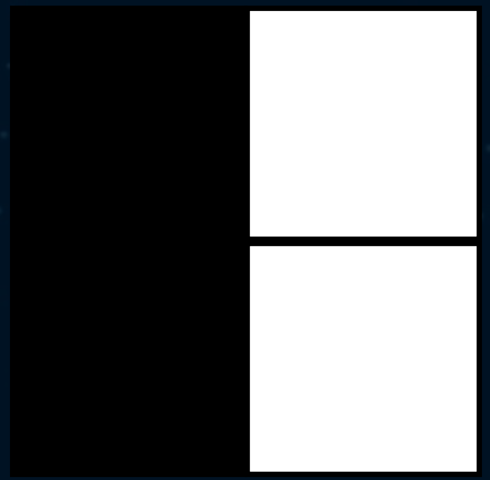
white if $i_j, w_j = +1$

black if $i_j, w_j = -1$

$\boldsymbol{i} =$ $(1,-1,1,-1)$  $\boldsymbol{w} =$ $(1,-1,1,-1)$ 

$\langle \psi_w | \psi_i \rangle = \boldsymbol{i} \cdot \boldsymbol{w} = 1$

✓ Perfect activation

$\boldsymbol{i} =$ $(1,-1,1,-1)$  $\boldsymbol{w} =$ $(-1,1,1,-1)$ 

$\langle \psi_w | \psi_i \rangle = \boldsymbol{i} \cdot \boldsymbol{w} = 0$

✗ No activation

A Continuous Quantum Neuron [3]

Discrete

$$\textcircled{+1} \longrightarrow \textcircled{+}|\psi\rangle$$
$$\textcircled{-1} \longrightarrow \textcircled{-}|\psi\rangle$$

Continuous

$$+1 = e^{i2\pi}$$
$$-1 = e^{i\pi}$$
$$\longrightarrow \quad e^{i\theta}|\psi\rangle$$

Phase Encoding

Encode data on the phases of the quantum state

input $\quad \boldsymbol{\theta} = (\theta_0, \theta_1, \ldots, \theta_{n-1})$

weights $\quad \boldsymbol{\phi} = (\phi_0, \phi_1, \ldots, \phi_{n-1})$

$\theta_j, \phi_j \in [0,\pi]$ (not 2π due to periodicity)

$$|\psi_\theta\rangle = \frac{1}{\sqrt{2^N}} \sum_{j=0}^{2^N-1} e^{i\theta_j} |j\rangle$$

$$|\psi_\phi\rangle = \frac{1}{\sqrt{2^N}} \sum_{j=0}^{2^N-1} e^{i\phi_j} |j\rangle$$

$$\longrightarrow$$

$$|\langle\psi_\phi|\psi_\theta\rangle|^2 = \left|\sum_j^{2^N-1} e^{i(\theta_j-\phi_j)}\right|^2 = \ldots =$$
$$= \frac{1}{2^N} + \frac{1}{2^{2N-1}} \sum_{i<j}^{2^N-1} \cos((\theta_j - \phi_j) - (\theta_i - \phi_i))$$

[3] S. Mangini *et al.*, "Quantum computing model of an artificial neuron with continuously valued input data", *Mach. Learn.: Sci. Technol.* **1** 045008 (2020).

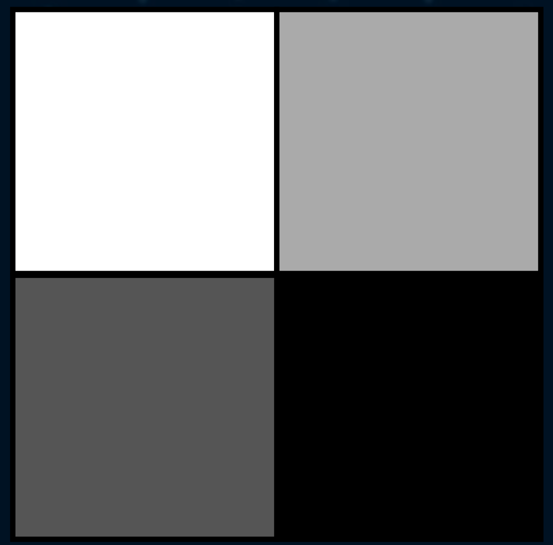
Checkerboard classification [3]

Grayscale images

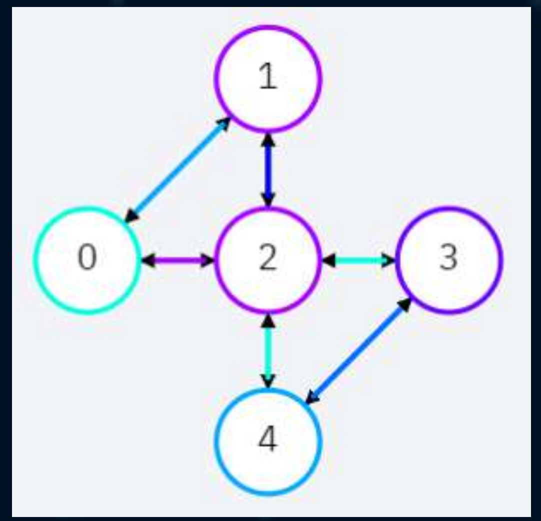
$\boldsymbol{i} = (255,170,85,0)$

$i_j \in [0,255]$

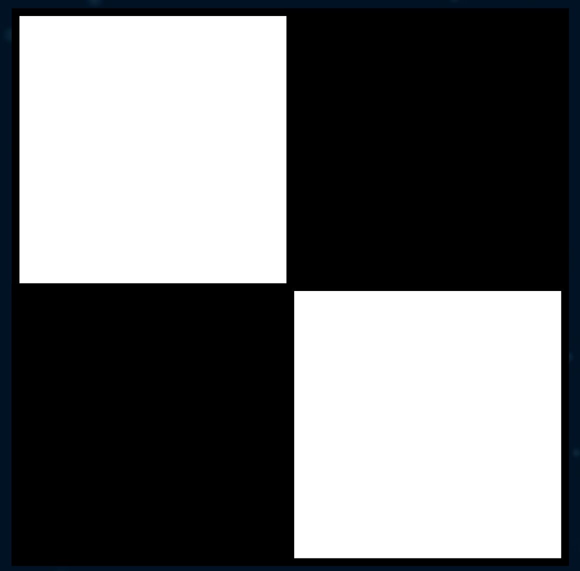
255	170
85	0

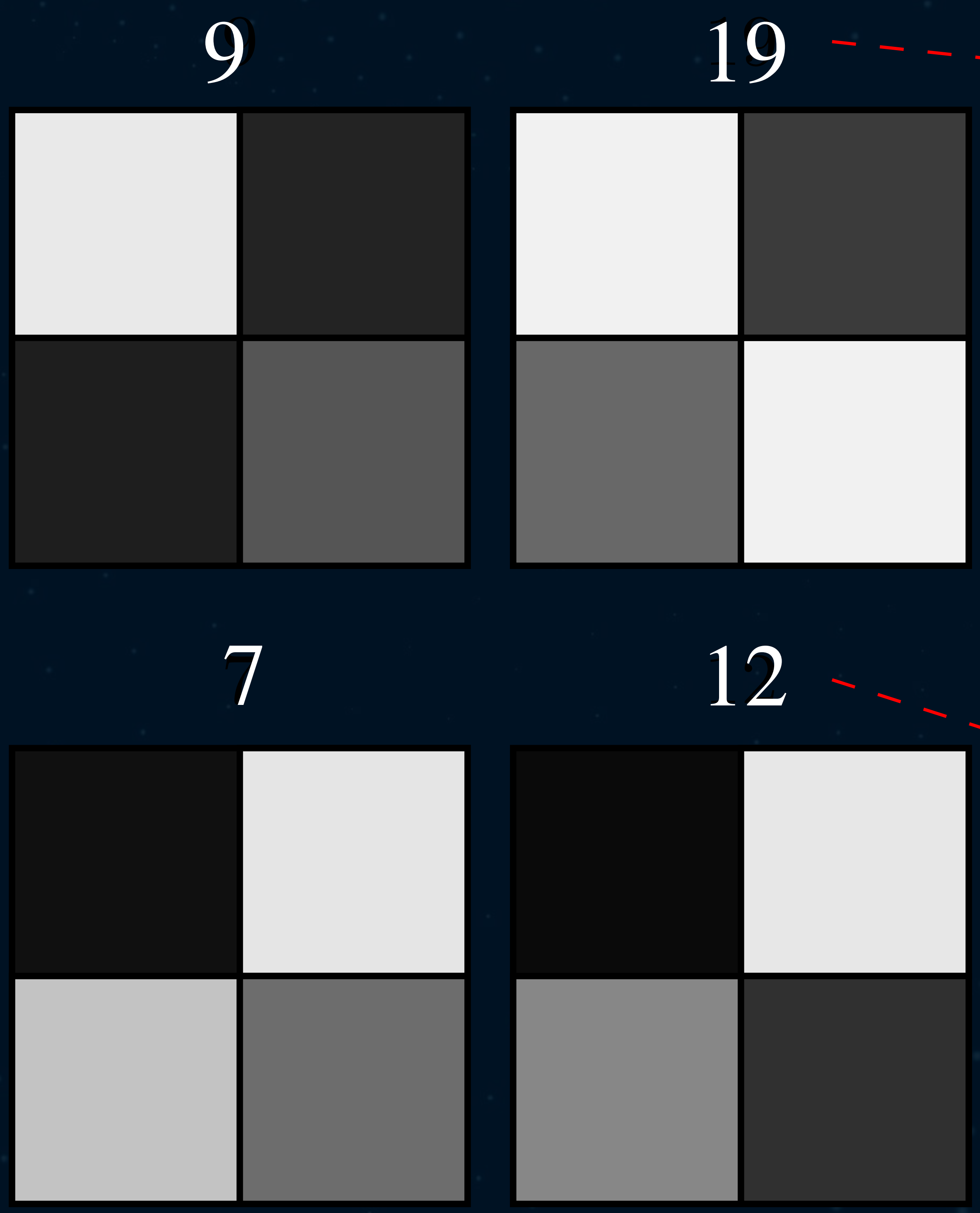


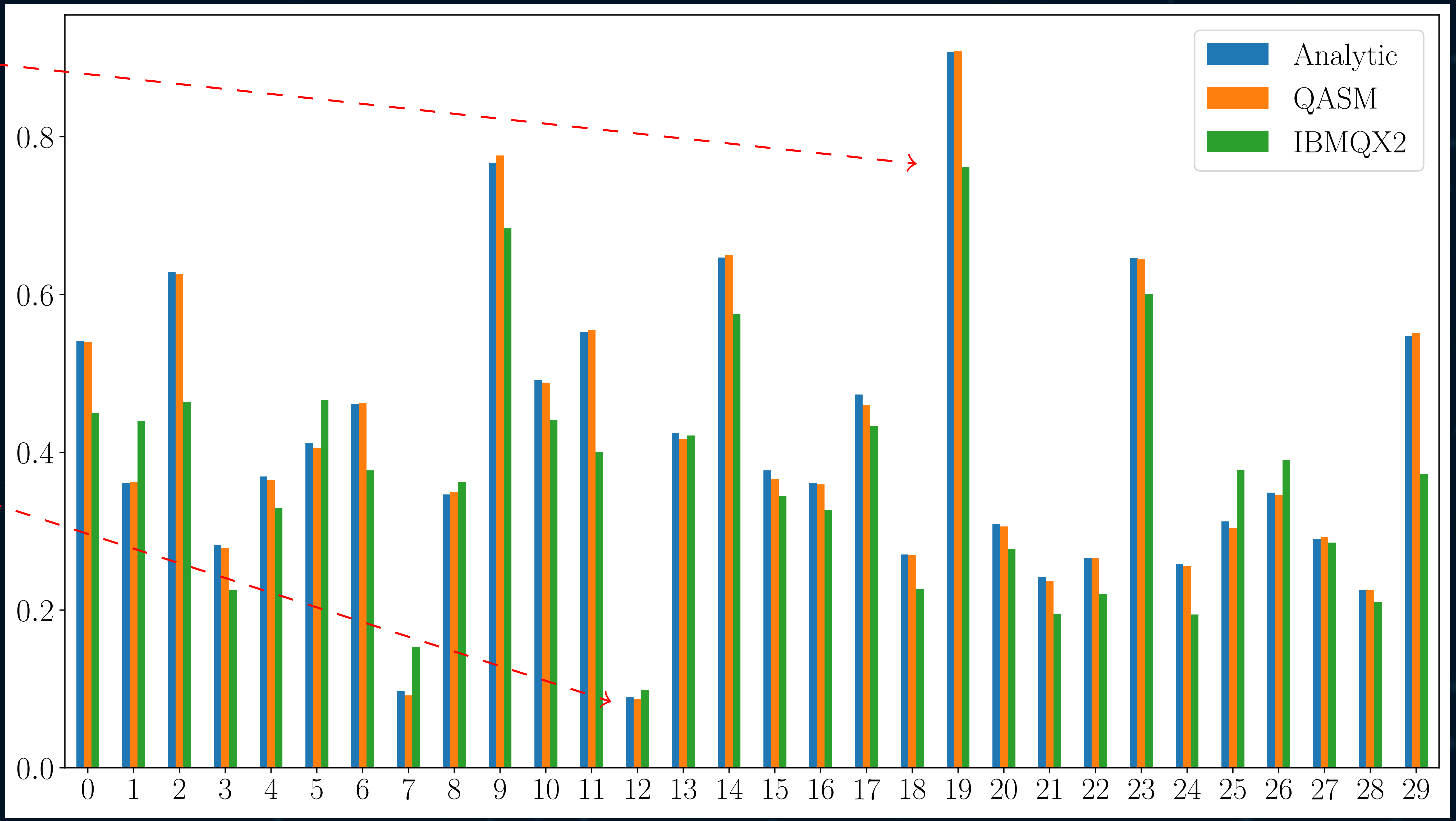
Normalization $\quad \boldsymbol{i} \rightarrow \dfrac{\pi/2}{255}\boldsymbol{i}$



ϕ_{target}







[3] S. Mangini *et al.*, "Quantum computing model of an artificial neuron with continuously valued input data", *Mach. Learn.: Sci. Technol.* **1** 045008 (2020).

Introduction

Neuron Model

Variational Learning

Results

Problem: efficient implementation

State preparation is inefficient in the number of qubits $\quad i = (i_0, i_1, \ldots, i_{n-1})$

# of qubits

$N = \log n$

Brute-force approach

both for discrete and continuous

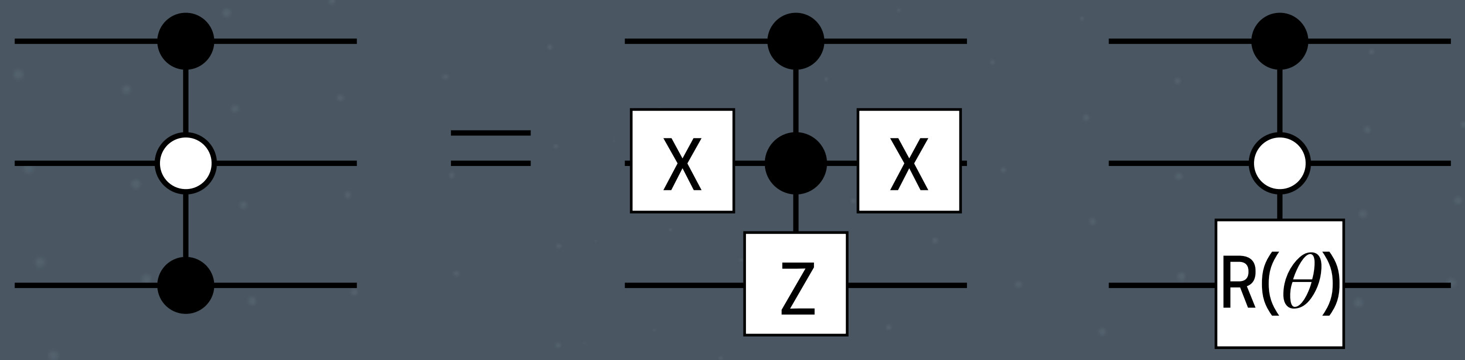
$$\vdots$$

$$|010\rangle \rightarrow i_{010}|010\rangle$$

$$|110\rangle \rightarrow i_{110}|110\rangle$$

$$\vdots$$

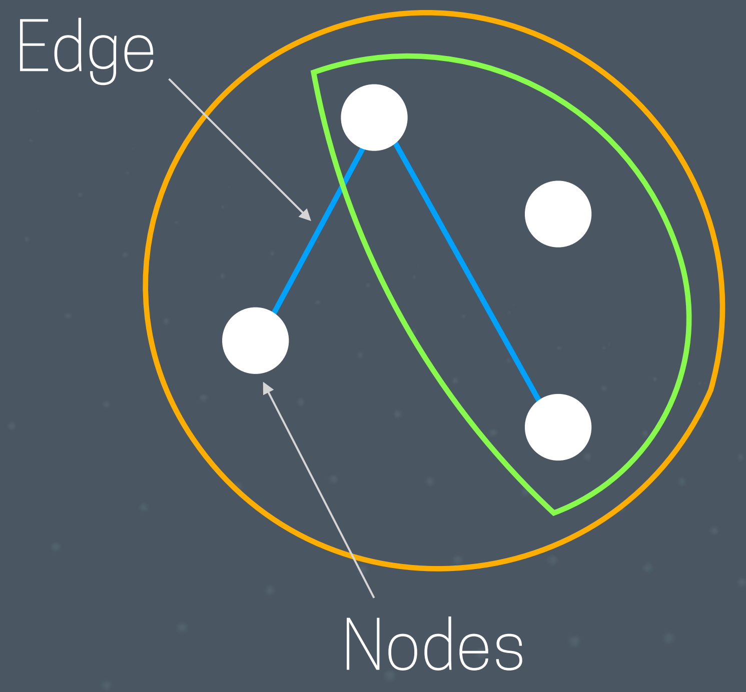
Requires $O(n)$ operations



R(θ) = $\begin{bmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{bmatrix}$

Hypergraph states

only discrete



$\longleftrightarrow$

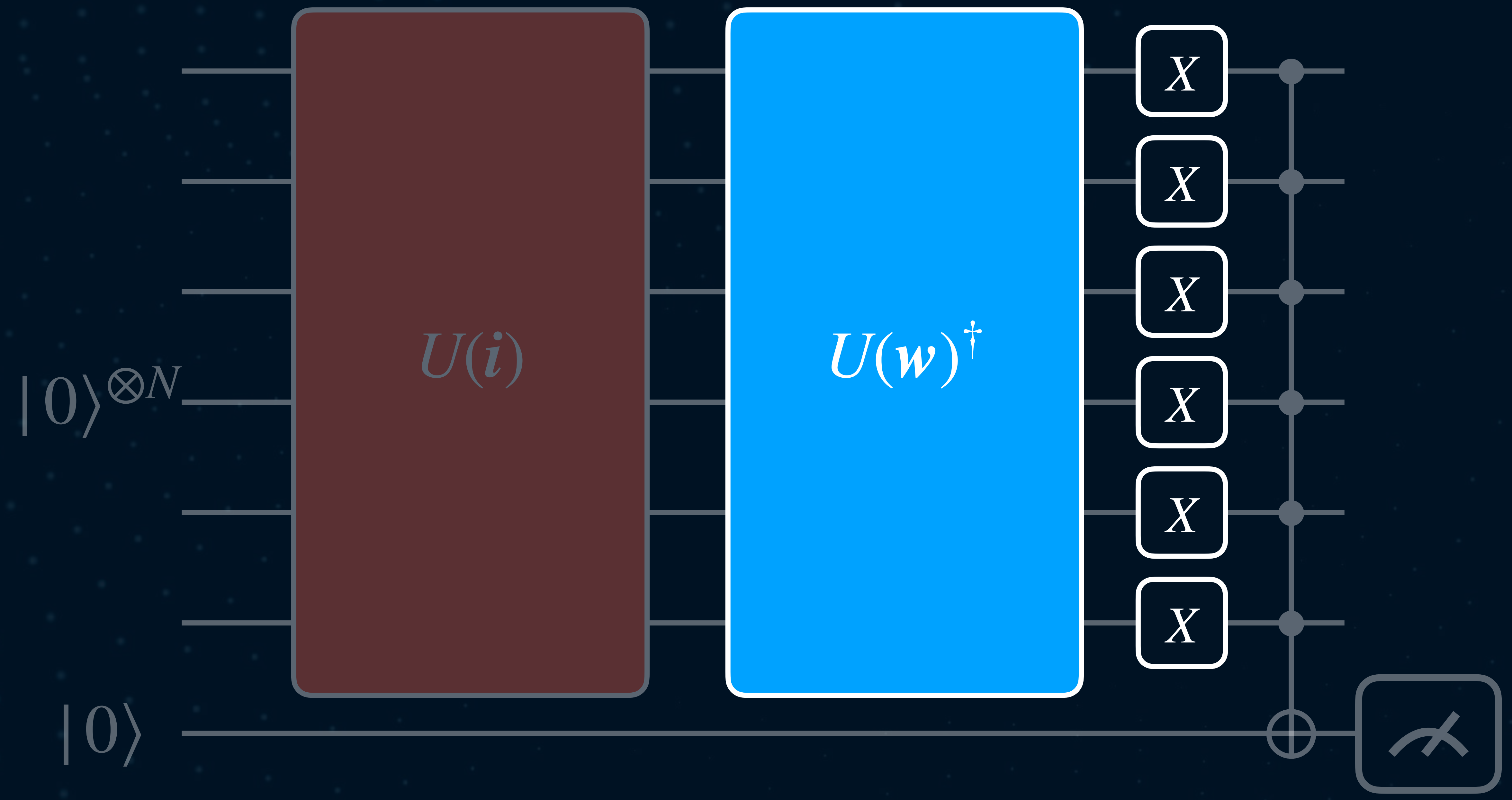
$$|\psi_i\rangle = \frac{1}{\sqrt{2^N}} \sum_{j=0}^{2^N-1} i_j |j\rangle$$

REW state

Requires still $O(n)$ operations but lower multi-qubit operations (at most one N-controlled gate)

Solution: make it variational! [4]

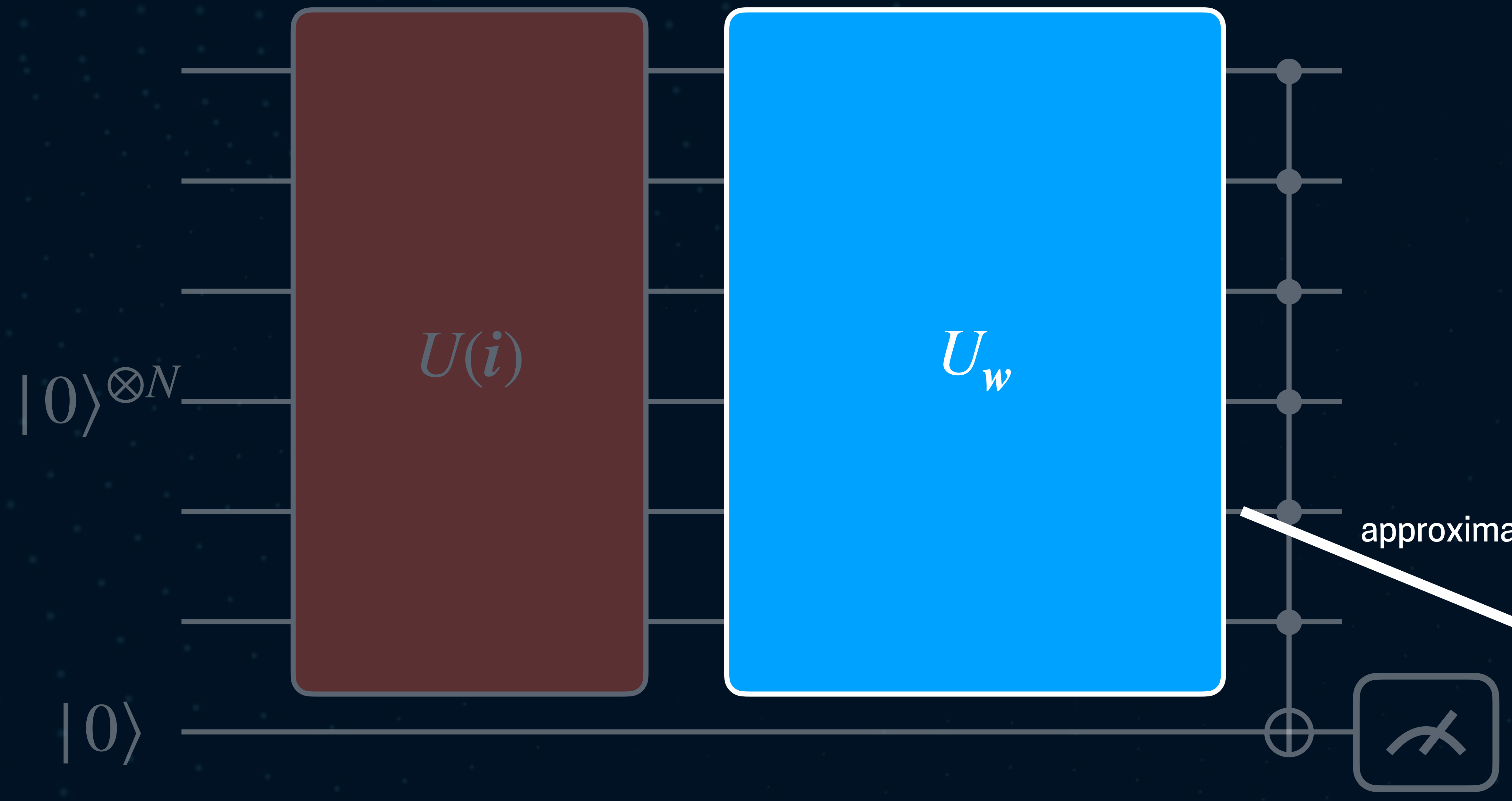
In the following, we focus on creating a variational implementation of U_w, assuming an efficient procedure for loading classical data on the quantum state is available (i.e. U_i is given).



$$U(w)\,|0\rangle^{\otimes N} = |\psi_w\rangle$$

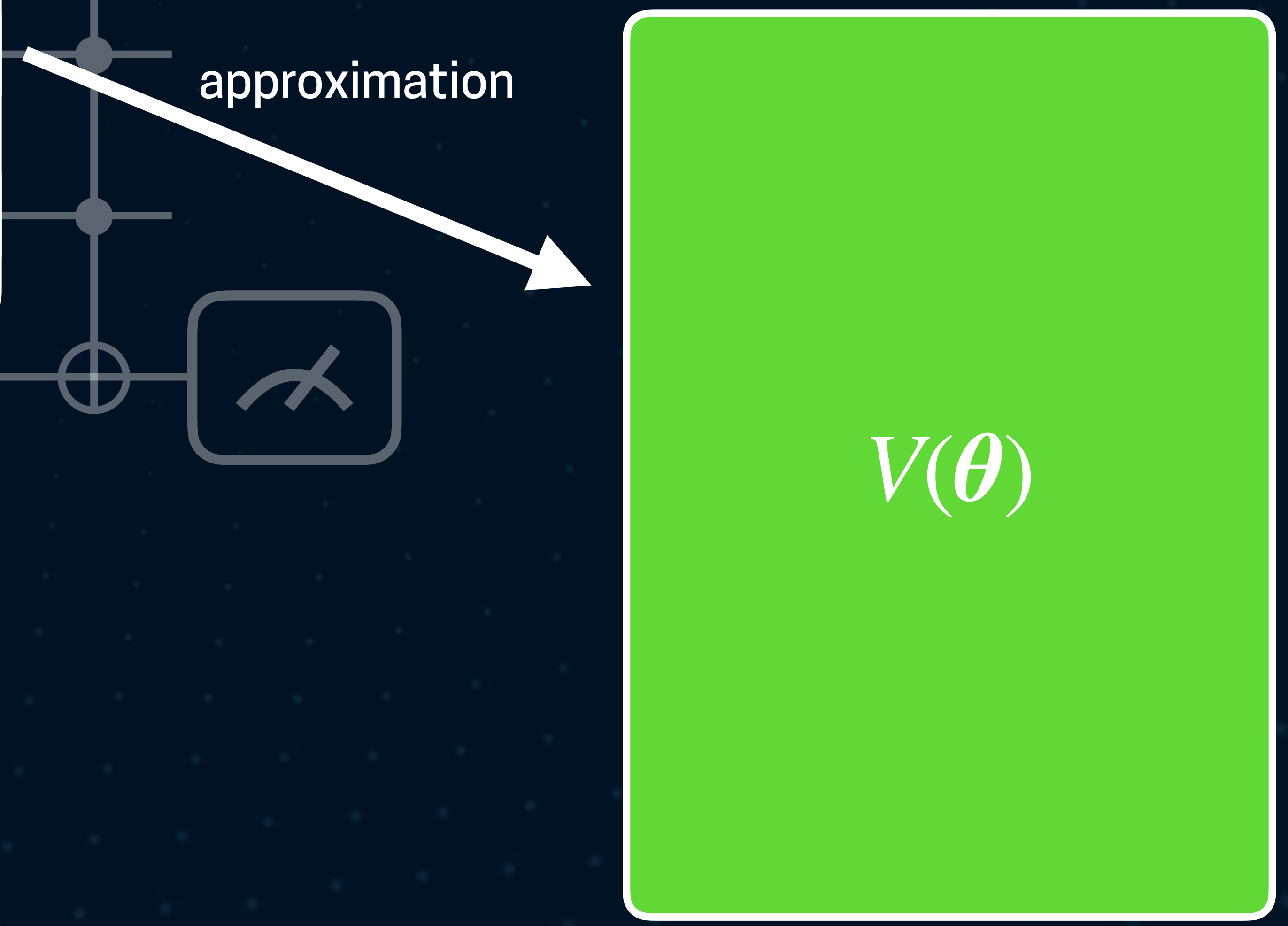
Solution: make it variational! [4]

In the following, we focus on creating a variational implementation of U_w, assuming an efficient procedure for loading classical data on the quantum state is available (i.e. U_i is given).



$$U(w)\,|0\rangle^{\otimes N} = |\psi_w\rangle \longrightarrow U_w = X^{\otimes N} U(w)$$

$$U_w\,|\psi_w\rangle = X^{\otimes N} \underbrace{U(w)\,U(w)^\dagger}_{=\mathbb{1}}\,|0\rangle^{\otimes N} = |1\rangle^{\otimes N}$$



Variational ansatz

$V(\theta^*) \approx U_w$

θ^* optimal angles

Optimization problem:

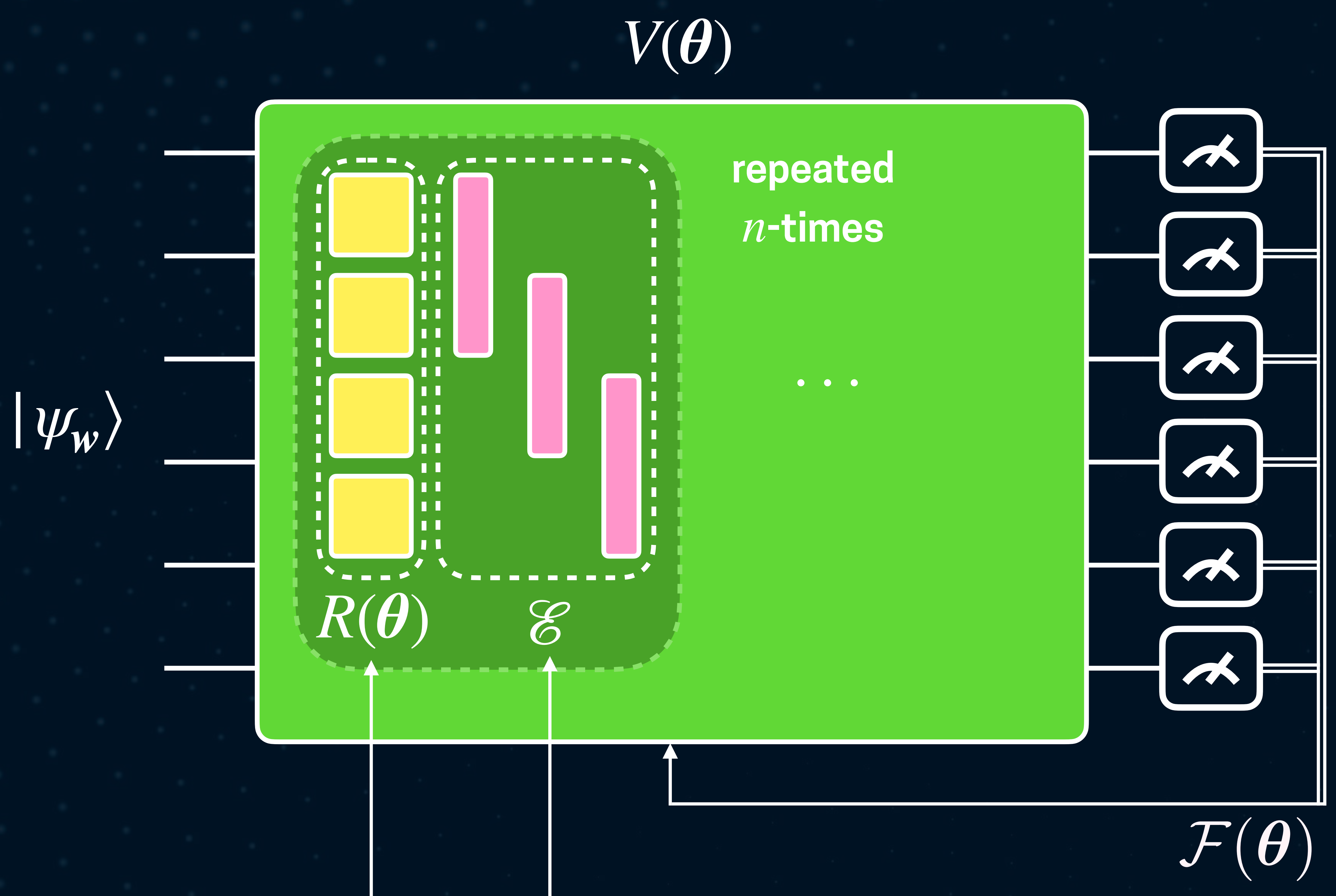
$$\mathcal{F}(\theta) = 1 - |\langle 11\ldots1|V(\theta)|\psi_w\rangle|^2$$

$\theta^* =$ arg min$_\theta$ $\mathcal{F}(\theta)$

Note: optimal values are specific to a w

Global and Local optimization

Global strategy



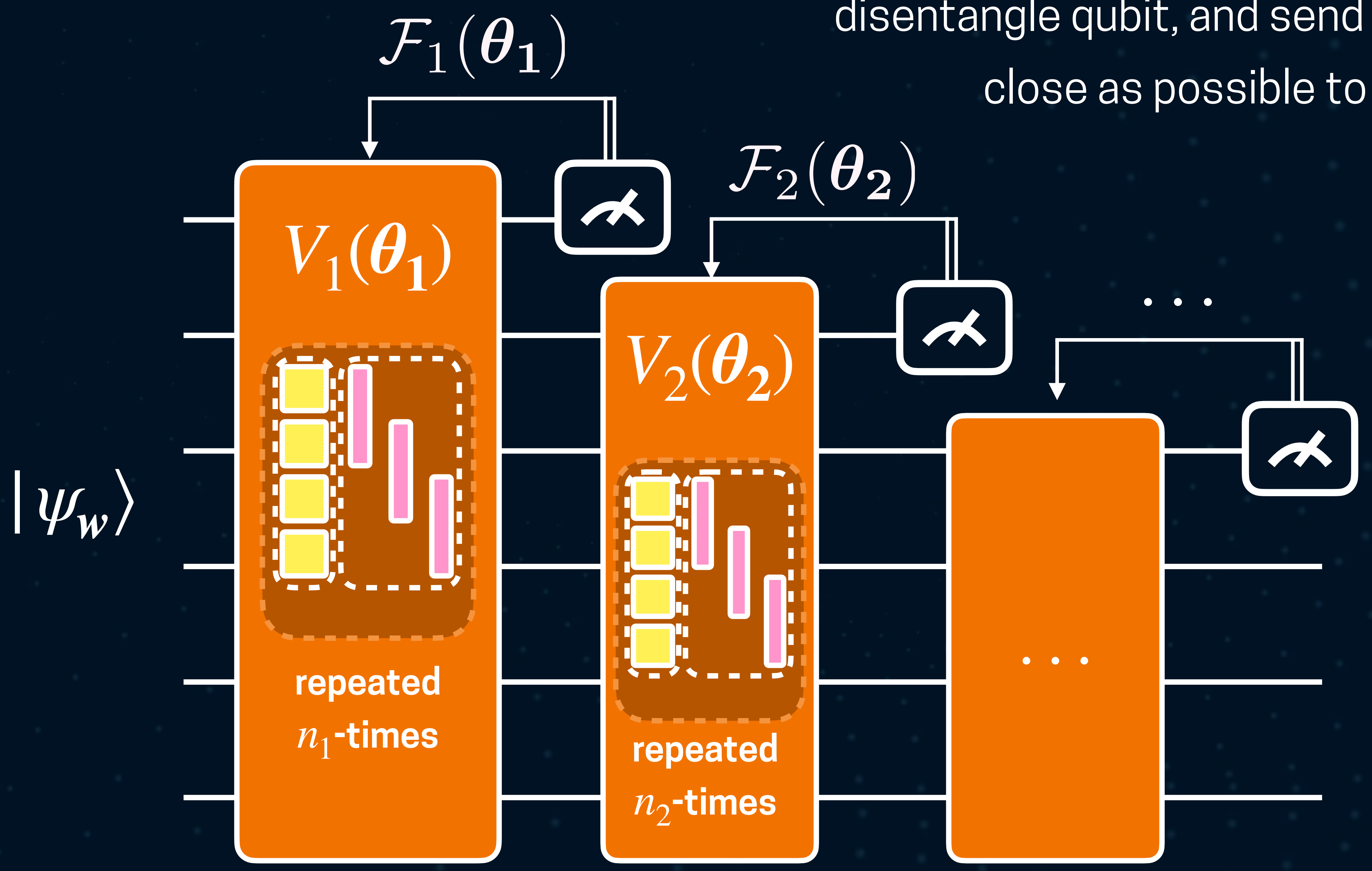
Entangling layer: all-to-all or nearest neighbors

Single qubits rotations (around σ_y)

$$\mathcal{F}(\boldsymbol{\theta}) = 1 - |\langle 11\ldots 1|V(\boldsymbol{\theta})|\psi_{\boldsymbol{w}}\rangle|^2$$

Local strategy

disentangle qubit, and send it as close as possible to $|1\rangle$

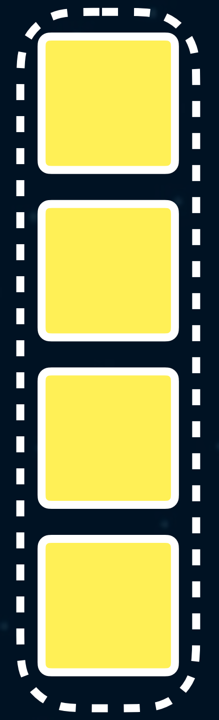


$$\mathcal{F}_j(\boldsymbol{\theta_j}) = 1 - \langle 1|\mathrm{Tr}_{j+1,\ldots,N}[\rho_j]|1\rangle$$

$$\rho_j = \begin{cases} |\psi_{\boldsymbol{w}}\rangle\langle\psi_{\boldsymbol{w}}| & j=1 \\ V_j(\boldsymbol{\theta_j})\rho_{j-1}V_j^\dagger(\boldsymbol{\theta_j}) & j>1 \end{cases}$$

Ansatz: technicalities

Variational gates



$R(\boldsymbol{\theta})$

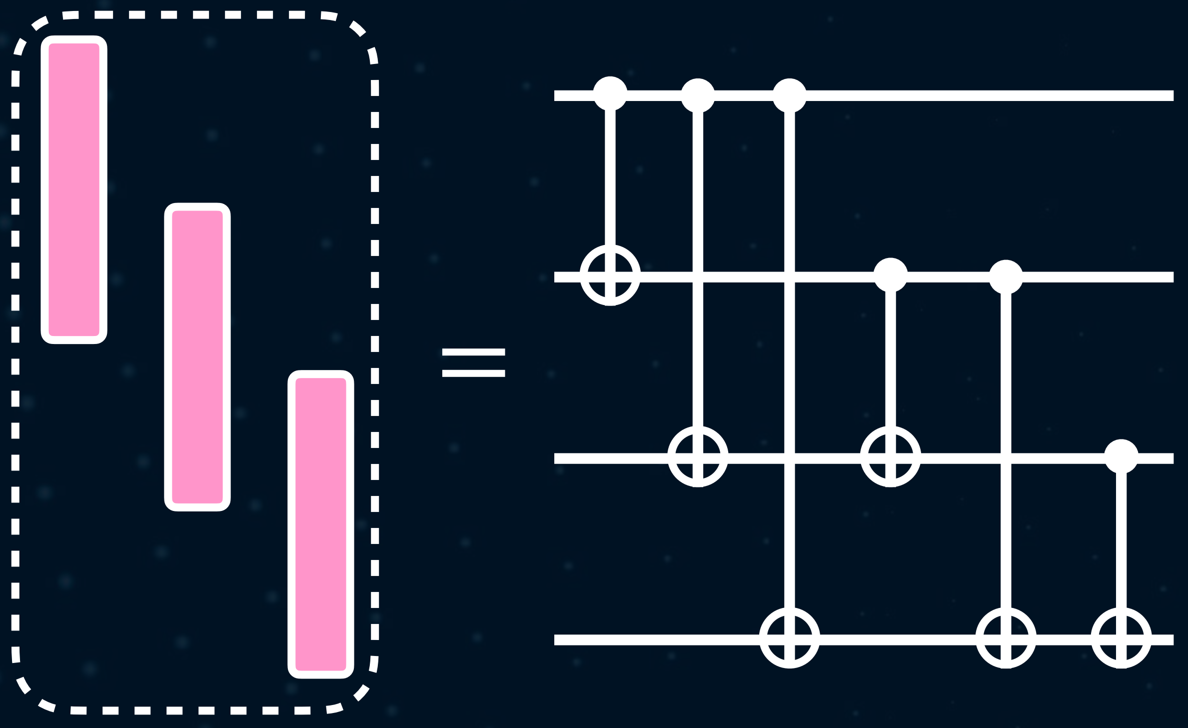
$$R(\theta) = \bigotimes_{q=1}^{4} \exp\left(i\theta_q \sigma_y^{(q)}/2\right)$$

$\sigma_y^{(q)}$:pauli matrix acting on qubit-q

for each layer

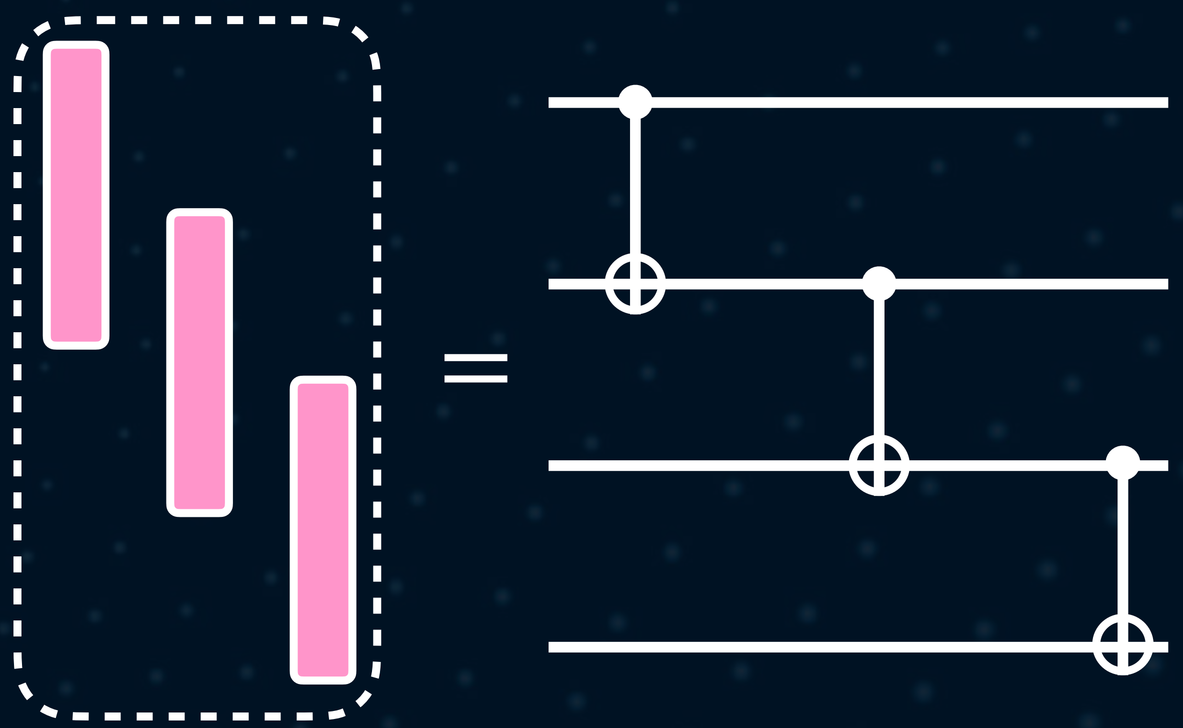
Entangling gates

All to all (*a2a*)



$$\mathscr{E}_{a2a} = \prod_{q} \prod_{q'=q+1}^{4} CNOT_{qq'}$$

Nearest neighbors (*nn*)



$$\mathscr{E}_{nn} = \prod_{q} CNOT_{q,q+1}$$

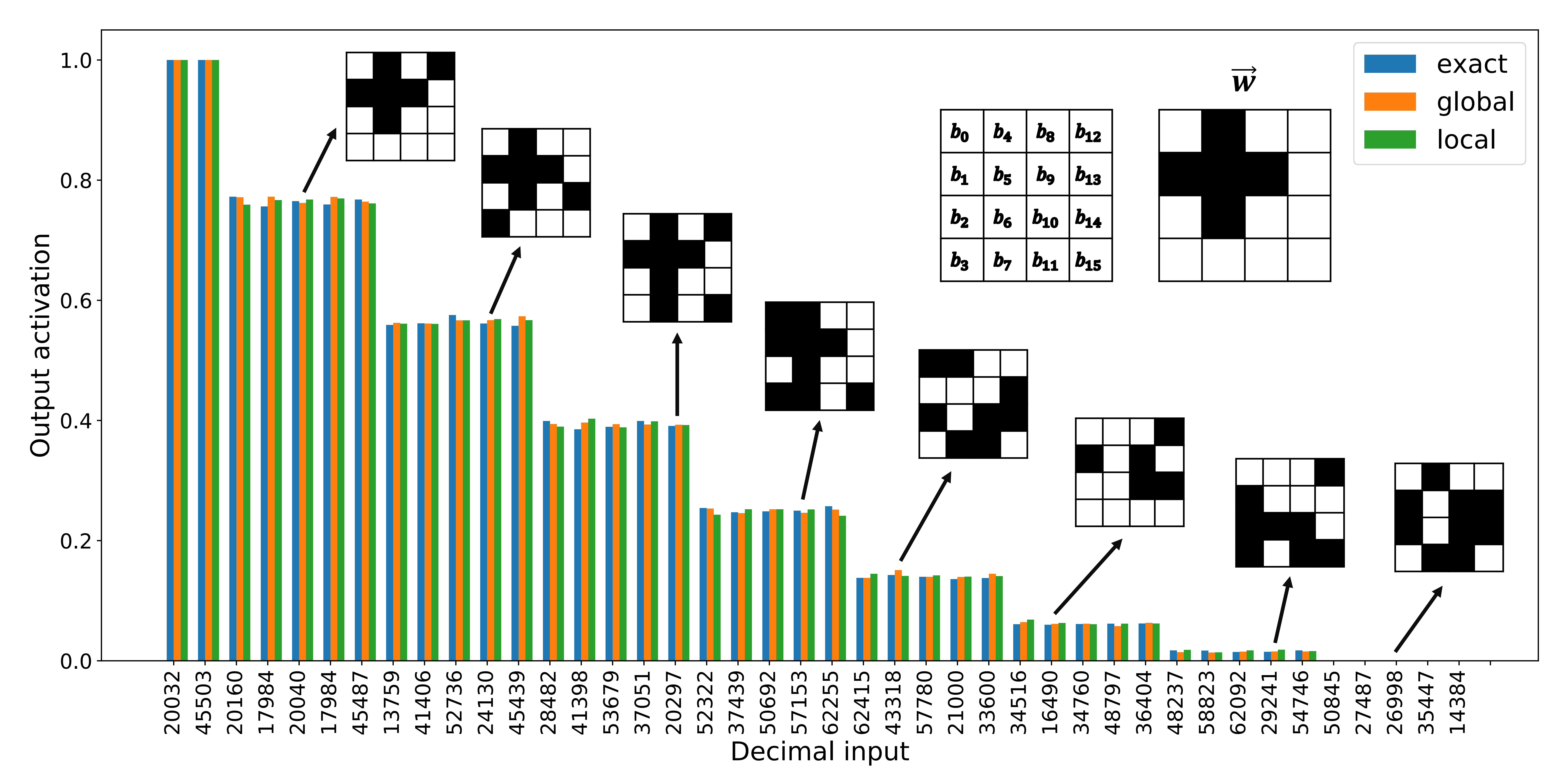
Introduction

Neuron Model

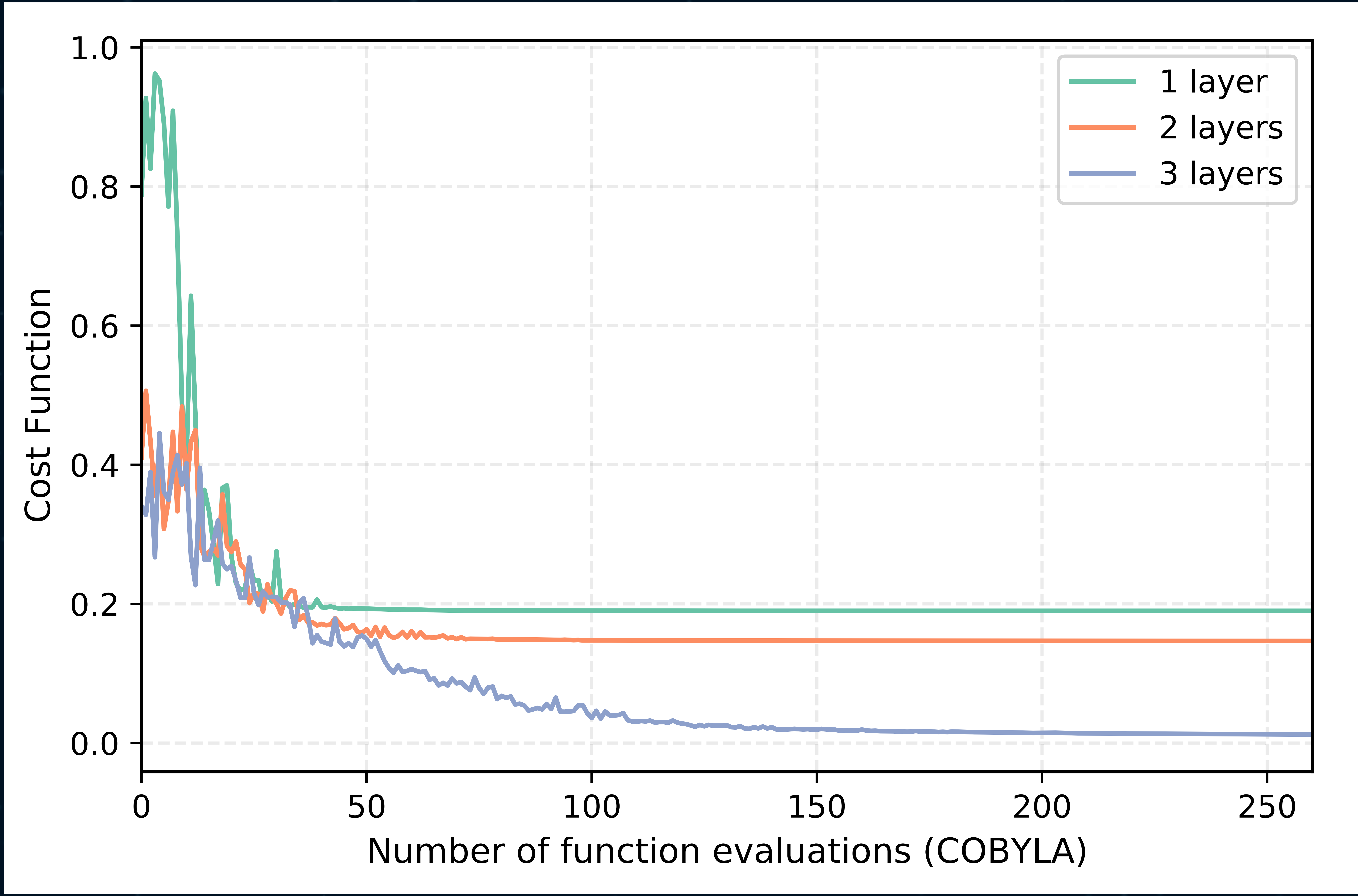
Variational Learning

Results

Comparison with exact method



Optimisation of Global with *nn*

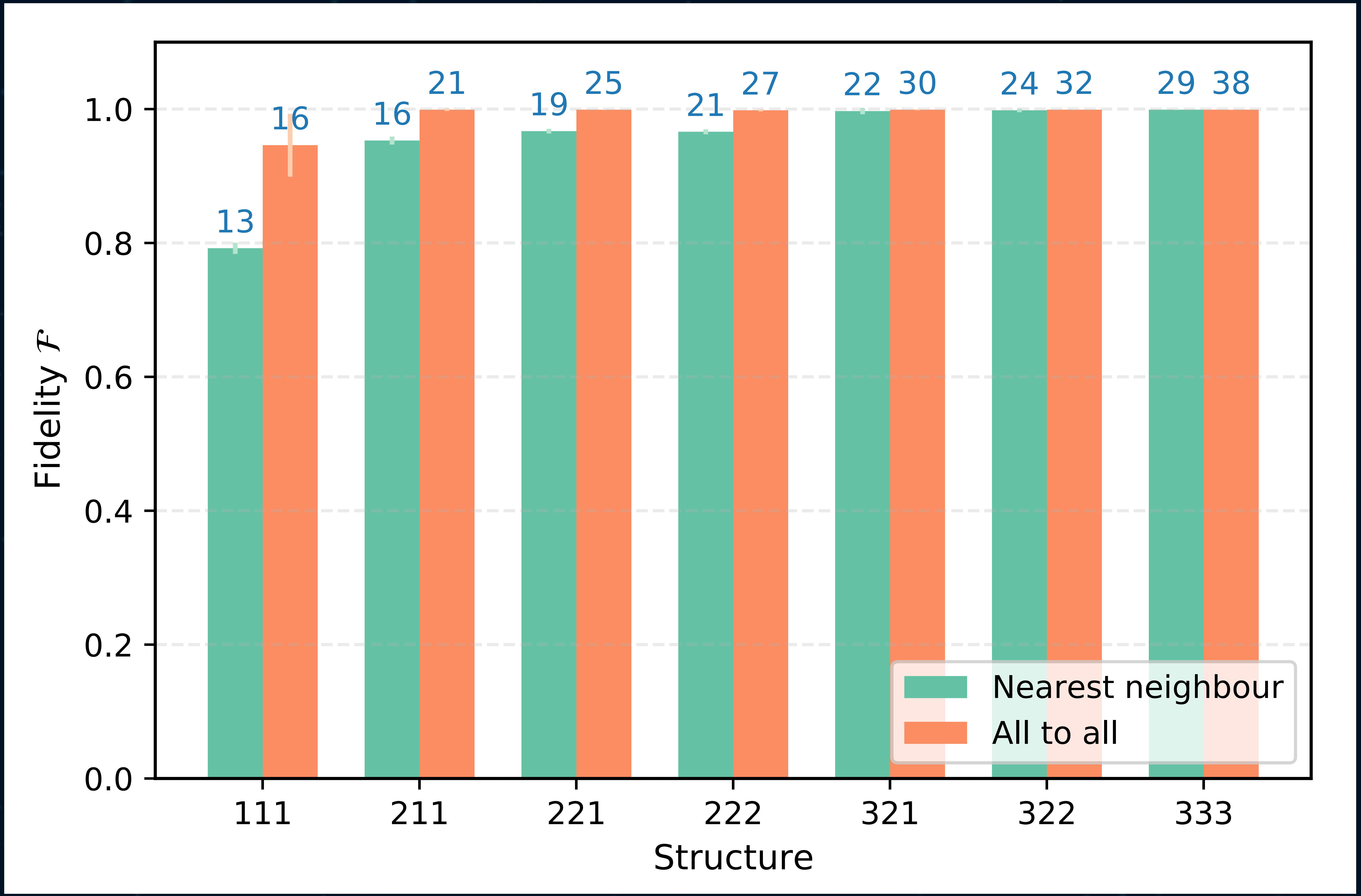


Optimization of Global unitary with *nn* entanglement.

More layers are needed to reach enough expressivity and learn the task

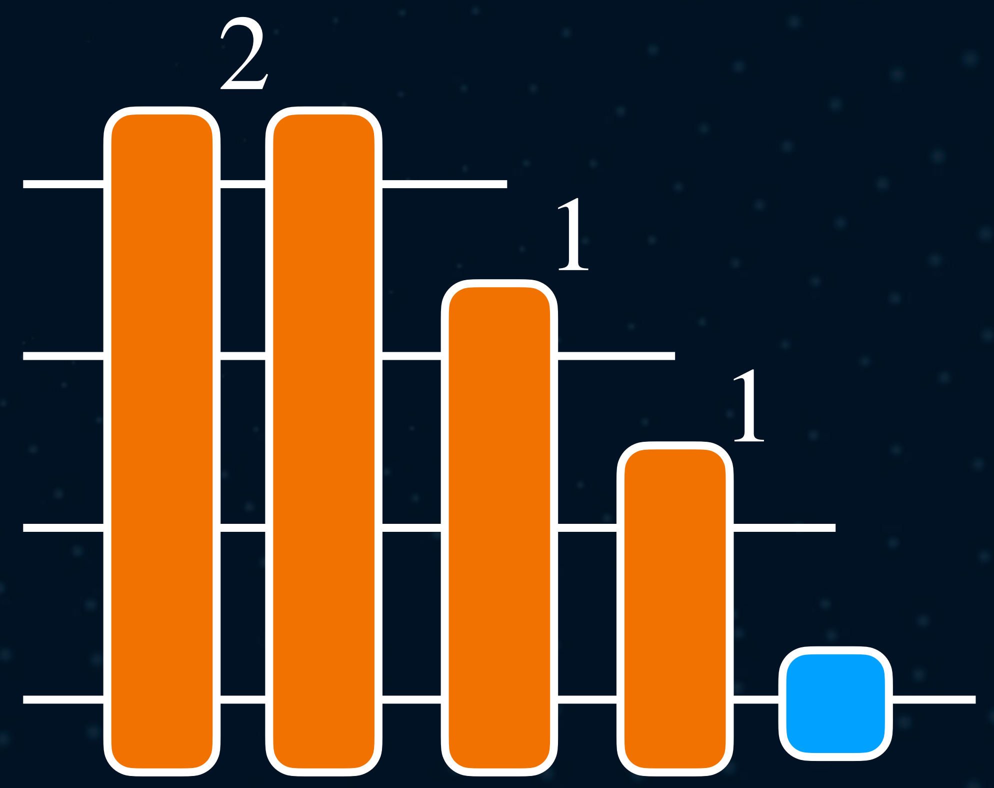
…but using a local approach we can use different depths

Local approach: structure optimization



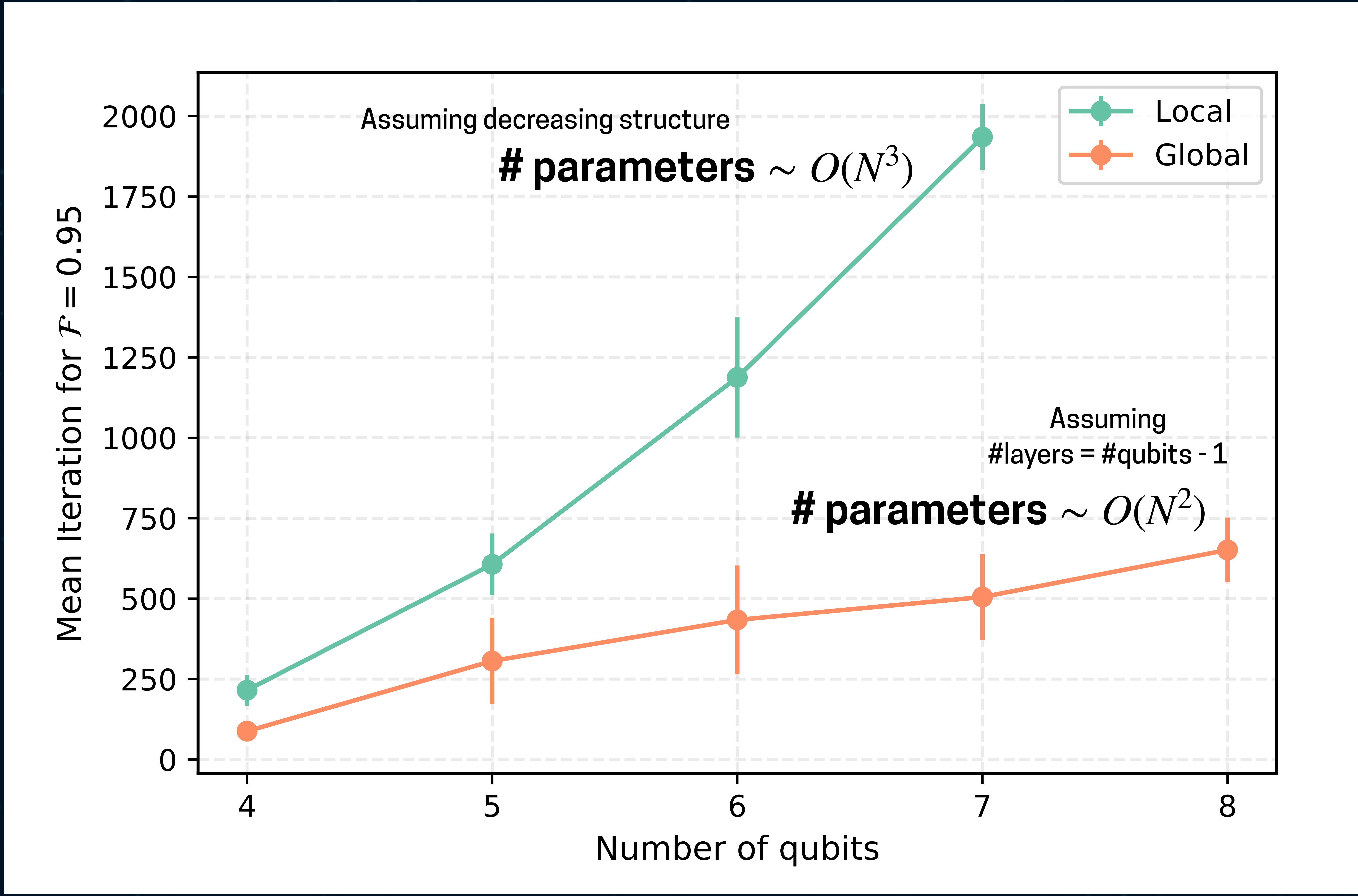
Optimization of local unitary using different structures

A '211' structure is:



optimization run 10 times for each structure

Noise-free: Global vs. Local



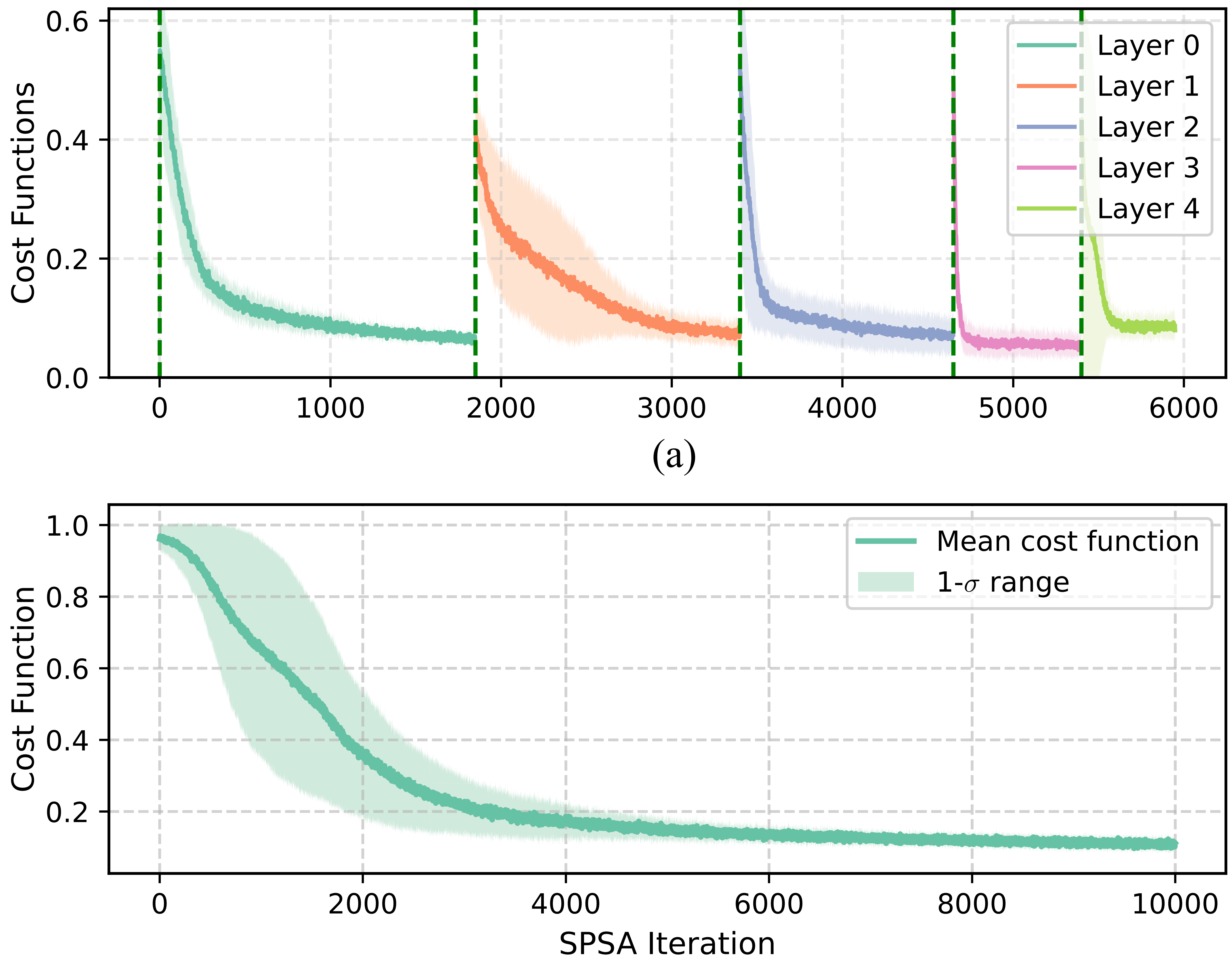
Number of optimization steps to reach target fidelity, in noiseless simulation.

Difference due to different scaling in the number of parameters (assuming decreasing structure for local).

No significant difference between local and global if exact simulation is used [5]

[5] Skolik, A., McClean, J.R., Mohseni, M. *et al.* "Layerwise learning for quantum neural networks", *Quantum Mach. Intell.* 3, 5 (2021)

Shot noise: Global vs. Local



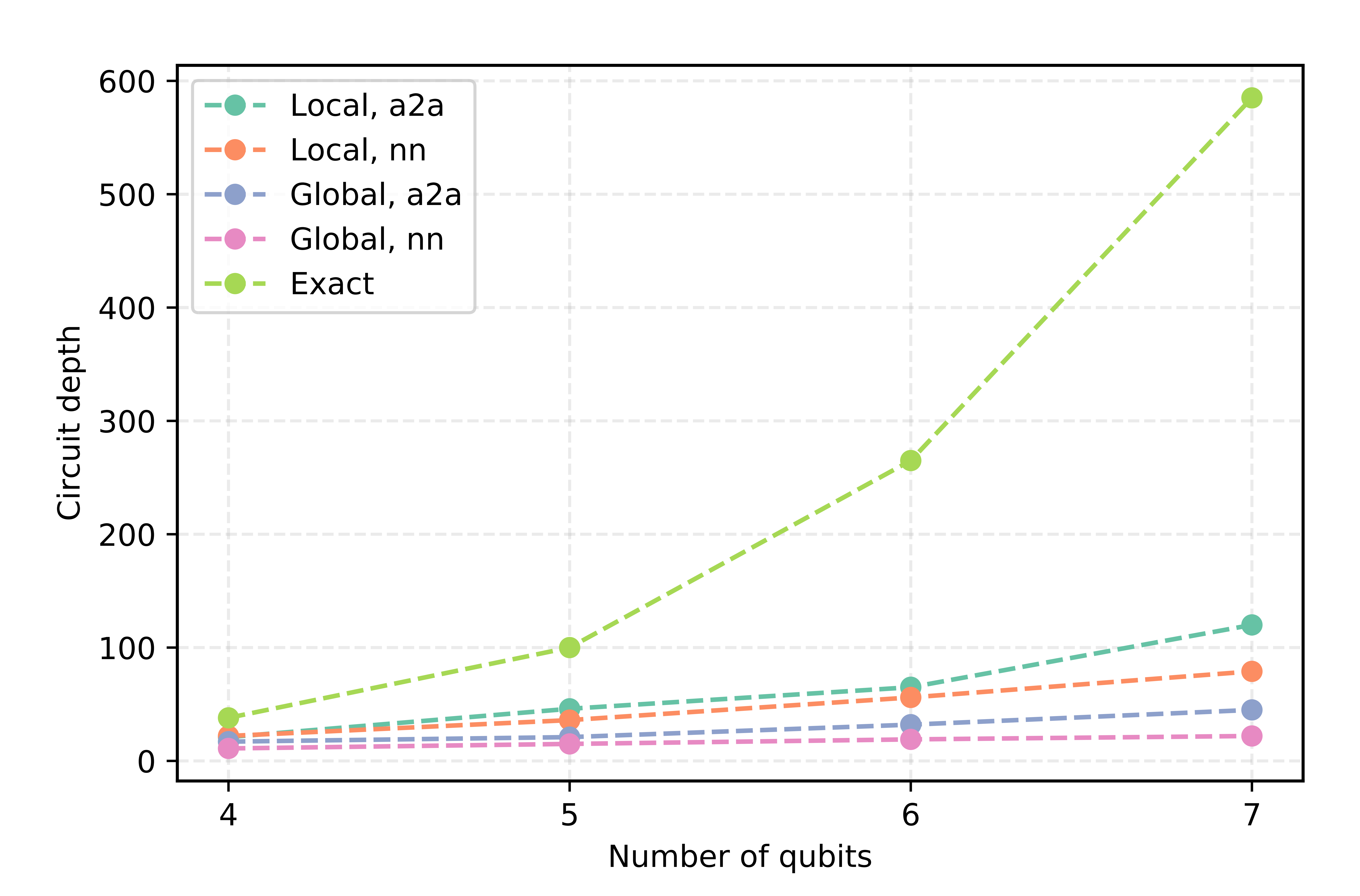
$\mathcal{F}_{\mathrm{local}}$
0.87 ± 0.02

$\mathcal{F}_{\mathrm{global}}$
0.89 ± 0.02

Number of optimization steps to reach target fidelity in presence of measurement noise.

Local approach reaches an optimal solution faster than global optimization!

Scaling



Scaling of the circuit depth wrt the number of qubits

Whatever the structure (global or local, a2a or nn), the variational approach requires much less computation wrt to exact implementation!

Summary and Outlooks

Summary

Introduced variational ansatz substituting exact circuit
Extensive study of different architectures
Noise plays major role in global vs. local
Variational approach lead to NISQ friendlier circuits

Outlooks

Analysis specific to a given pattern and task
Include other source of errors
Deeper theoretical understanding (a2a vs nn, local vs global)

thank you!

References

[1] S. Mangini *et al.*, “Quantum computing models for artificial neural networks”, *EPL (Europhysics Letters)* **2**, 1 (2021).

[2] F. Tacchino *et al.*, “An artificial neuron implemented on an actual quantum processor”, *npj Quantum Information* **5**, 26 (2019).

[3] S. Mangini *et al.*, “Quantum computing model of an artificial neuron with continuously valued input data”, *Mach. Learn.: Sci. Technol.* **1** 045008 (2020).

[4] F. Tacchino, S. Mangini *et al.*, "Variational Learning for Quantum Artificial Neural Networks”, *IEEE Transactions on Quantum Engineering*, vol. 2, pp. 1-10 (2021).

[1] [2] [3] [4]

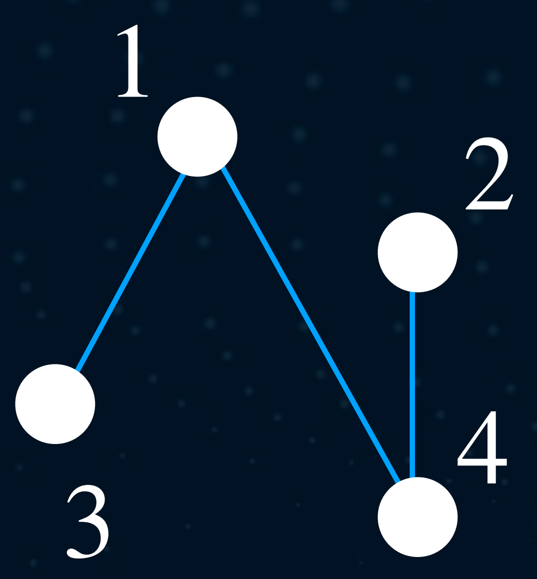
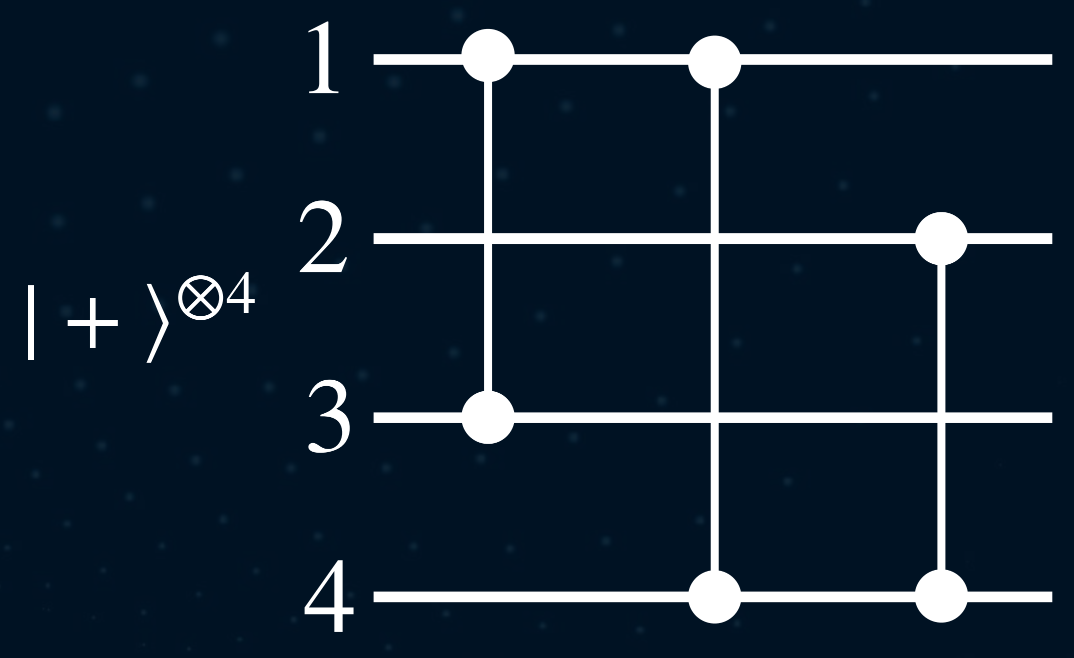
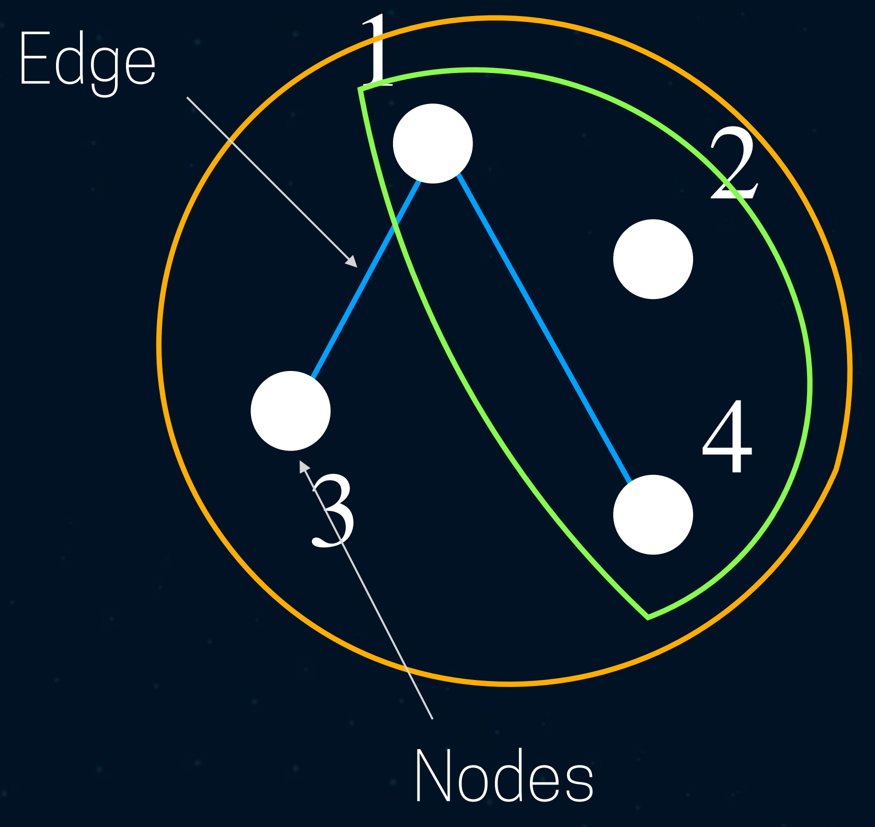
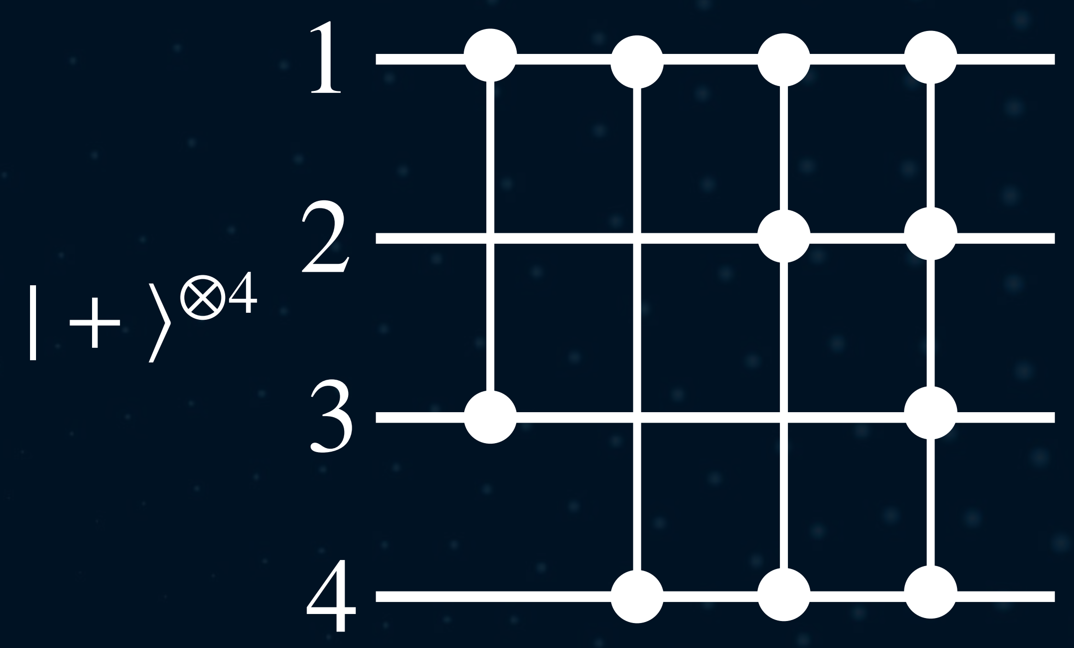
Group members: Chiara Macchiavello, Dario Gerace, Daniele Bajoni (Univ. of Pavia), Francesco Tacchino (IBM Quantum)

Summer School, *Machine Learning for Quantum Physics and Chemistry*
28/08/2021

stefano.mangini01@universitadipavia.it

@stfnmangini Stefano Mangini

Extra\ Hypergraph State Generation [6]

Graph	Quantum Graph State	Hypergraph	Quantum Hypergraph State
			
Edges connect 2 nodes	CZ between connected nodes	Edges connect multiple nodes	C^pZ between connected nodes

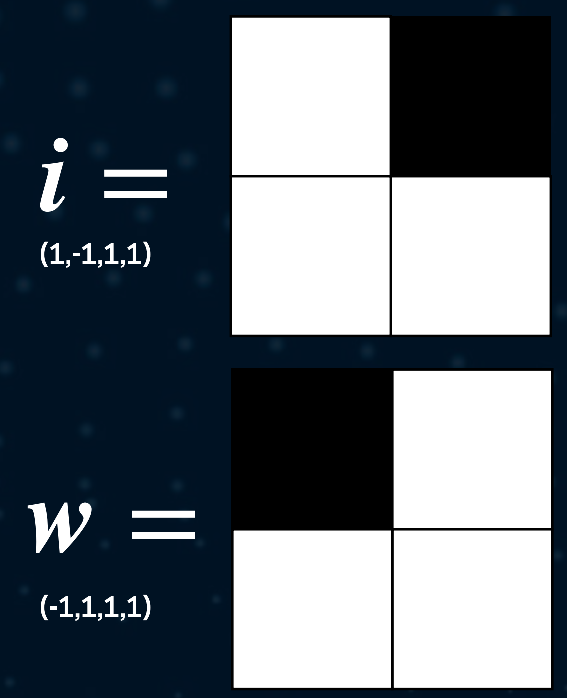
Hypergraph generation algorithm

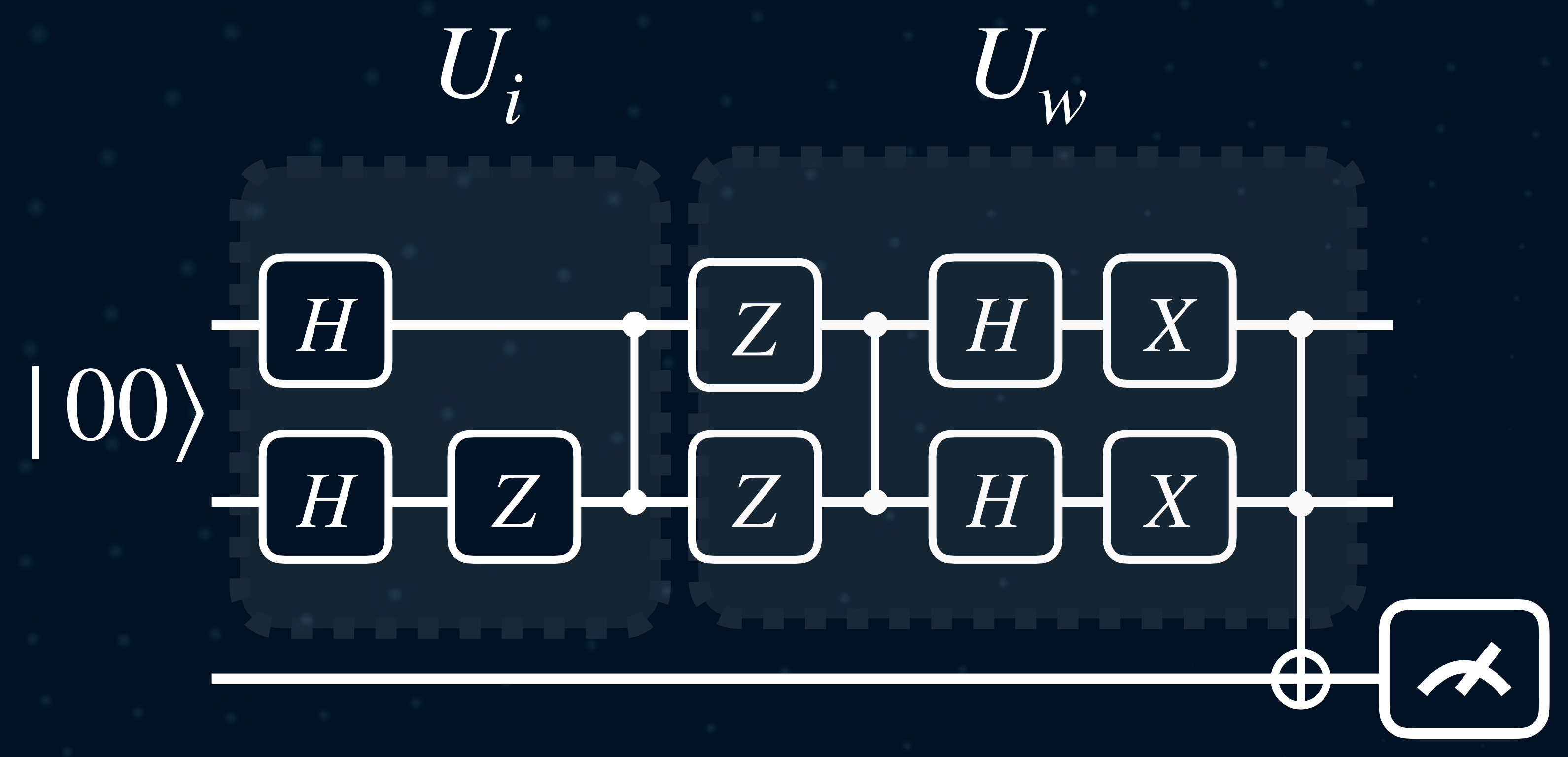
Input: $i \in \{-1,1\}^N$; **Output:** $\lvert \psi_i \rangle = \frac{1}{\sqrt{2^N}} \sum_{j=0}^{2^N-1} i_j \lvert j \rangle$

- Apply Hadamard on all qubits.
- Check for states with only one qubit in 1 requiring a -1 factor (e.g. $\lvert 010 \cdots 0 \rangle$), apply a Z gate on that qubit.
- For $p = 2, \cdots, N$:
  - Check for states with p qubits in 1 requiring a -1 factor, and apply a C^pZ gate on those qubits.
  - If a state has an unwanted -1, correct that applying C^pZ gate.

[6] M. Rossi *et al.*, *"Quantum hypergraph states"*, *New J. Phys.* 15 113022 (2013).

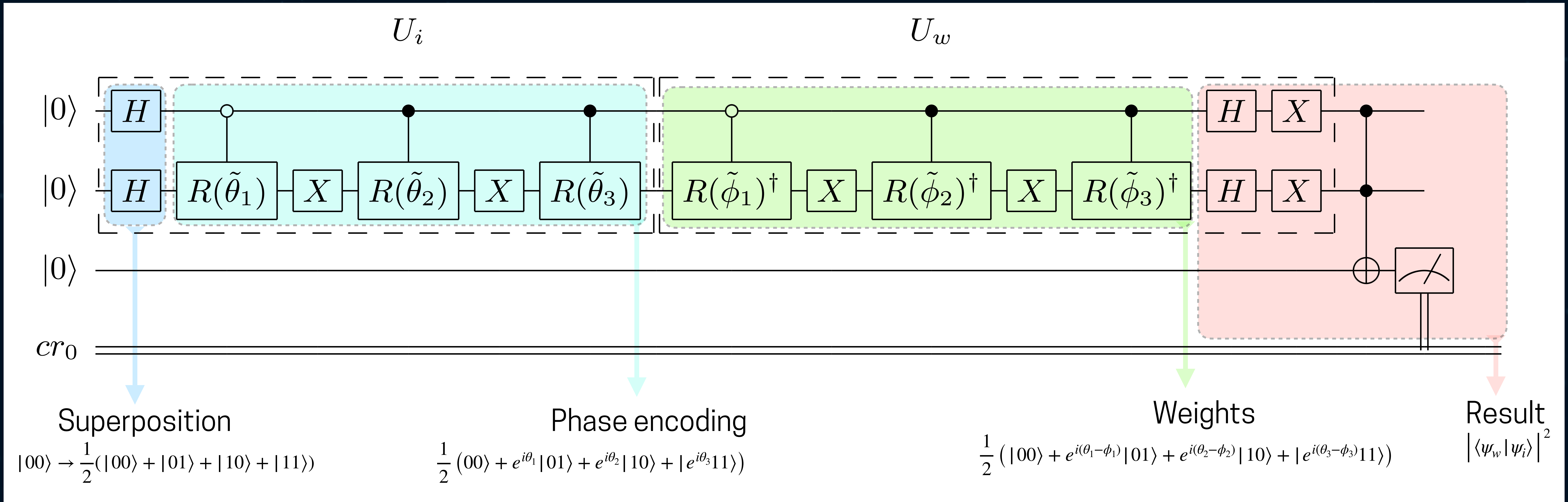
Extra\ Explicit circuits



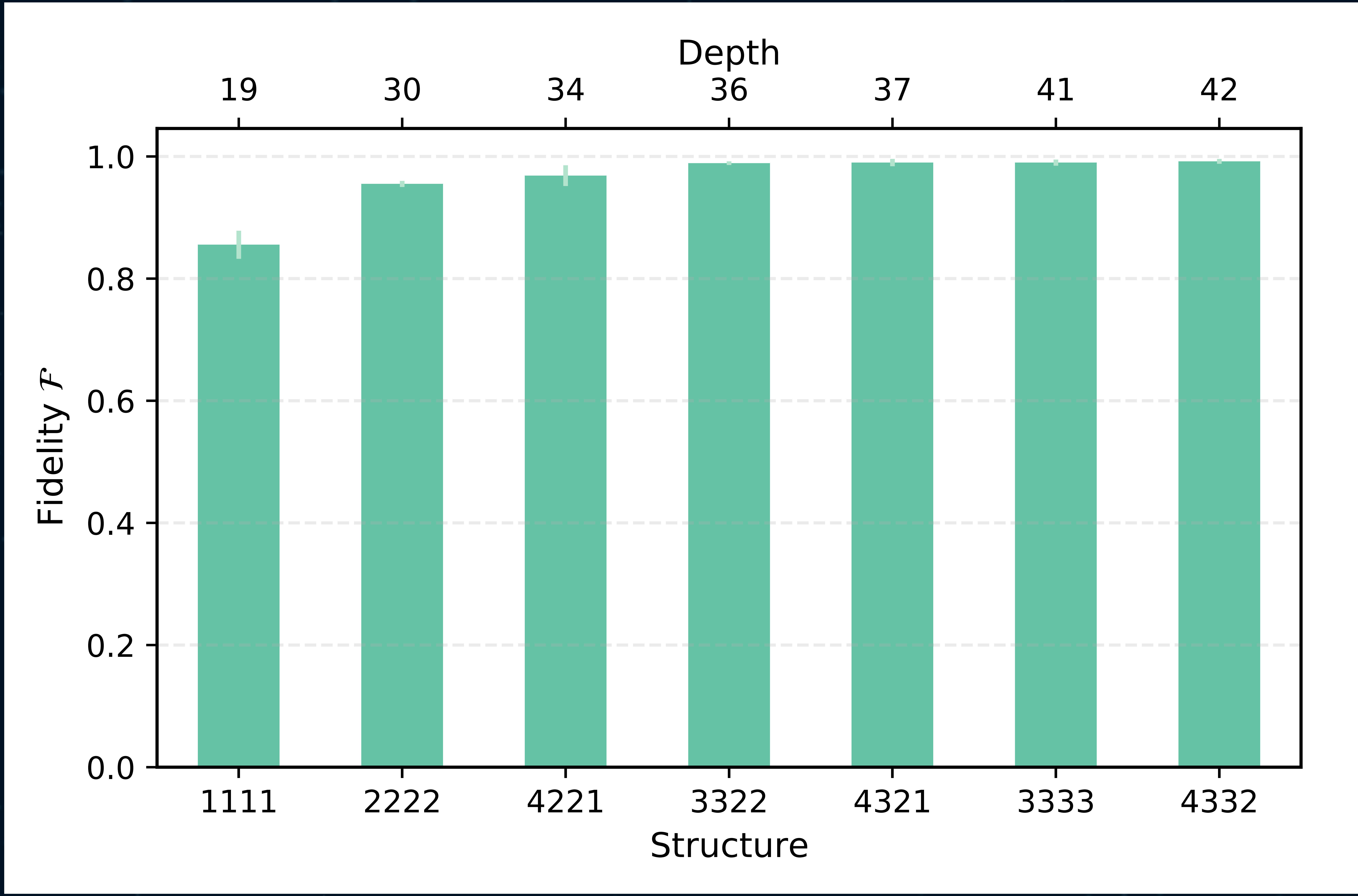


Discrete I/O

Continuous I/O



Extra\Local approach for 5 qubits



Optimization of local unitary for 5 qubits

Again, the most efficient structure is the always decreasing one.

Some cool properties

Activation Function

$$f(\boldsymbol{\theta}, \boldsymbol{\phi}) = \frac{1}{2^N} + \frac{1}{2^{2N-1}} \sum_{i<j}^{2^N-1} \cos((\theta_j - \phi_j) - (\theta_i - \phi_i))$$

- **Shift invariance**

$$f(\boldsymbol{\theta}, \boldsymbol{\theta}) = \frac{1}{2^N} + \frac{1}{2^{2N-1}} \sum_{i<j}^{2^N-1} = \frac{1}{2^N} + \frac{1}{2^{2N-1}} \frac{2^N(2^N-1)}{2} = 1$$

but also for $\boldsymbol{\theta} = \boldsymbol{\phi} + \boldsymbol{\Delta}$ with $\boldsymbol{\Delta}$ a constant! $\boldsymbol{\Delta} = (\Delta, \Delta, \cdots, \Delta)$

The activation function is *color shift invariant, i.e.*

$$f(\boldsymbol{\theta}, \boldsymbol{\phi}) = f(\boldsymbol{\theta}, \boldsymbol{\phi} + \boldsymbol{\Delta})$$

- **Noise resilience**

$$\boldsymbol{\theta} = \boldsymbol{\phi} + \boldsymbol{\Delta}, \quad \Delta_j \sim \mathrm{Unif}(-a/2,\, a/2)$$

$$\langle f(\boldsymbol{\theta}, \boldsymbol{\phi}) \rangle \approx 1 - O(a^2)$$

like *coherent errors* due to under/over-rotations in parametrized gates