Introduction to Hybrid Quantum-Classical Machine Learning with Applications

Presenter: Samuel Yen-Chi Chen

Quantum Computing and Quantum Machine Learning Algorithms for High Energy Physics Workshop

Date 2022-06-01

I. Introduction
II. Quantum Computing (QC)
III. Quantum Machine Learning
IV. Applications-Classification
V. Applications-Sequential Learning
VI. Applications-Reinforcement Learning
VII. Conclusion and Outlook

I. Introduction
II. Quantum Computing (QC)
III. Quantum Machine Learning
IV. Applications-Classification
V. Applications-Sequential Learning
VI. Applications-Reinforcement Learning
VII. Conclusion and Outlook

2016 AlphaGo

ARTICLE

doi:10.1038/nature16961

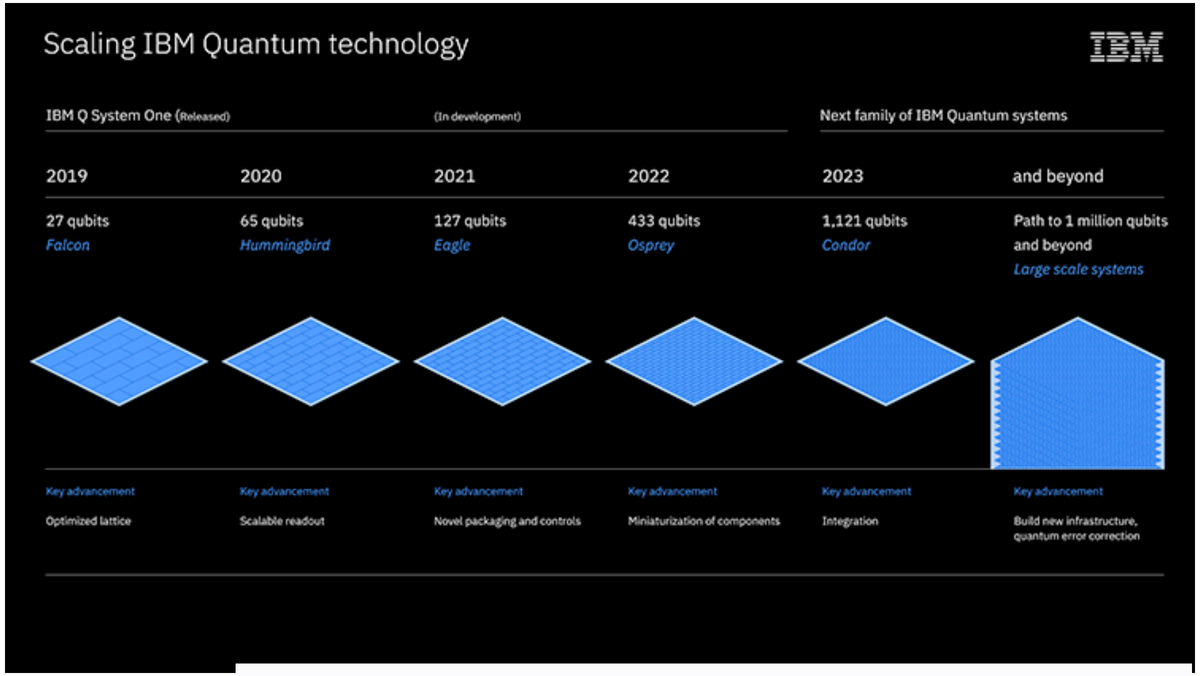
Mastering the game of Go with deep neural networks and tree search

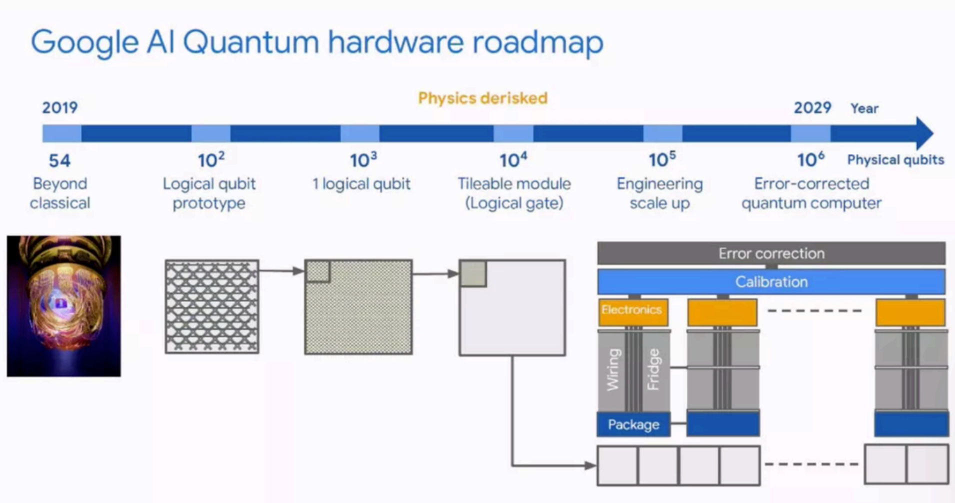
David Silver[1*], Aja Huang[1*], Chris J. Maddison[1], Arthur Guez[1], Laurent Sifre[1], George van den Driessche[1], Julian Schrittwieser[1], Ioannis Antonoglou[1], Veda Panneershelvam[1], Marc Lanctot[1], Sander Dieleman[1], Dominik Grewe[1], John Nham[2], Nal Kalchbrenner[1], Ilya Sutskever[2], Timothy Lillicrap[1], Madeleine Leach[1], Koray Kavukcuoglu[1], Thore Graepel[1] & Demis Hassabis[1]

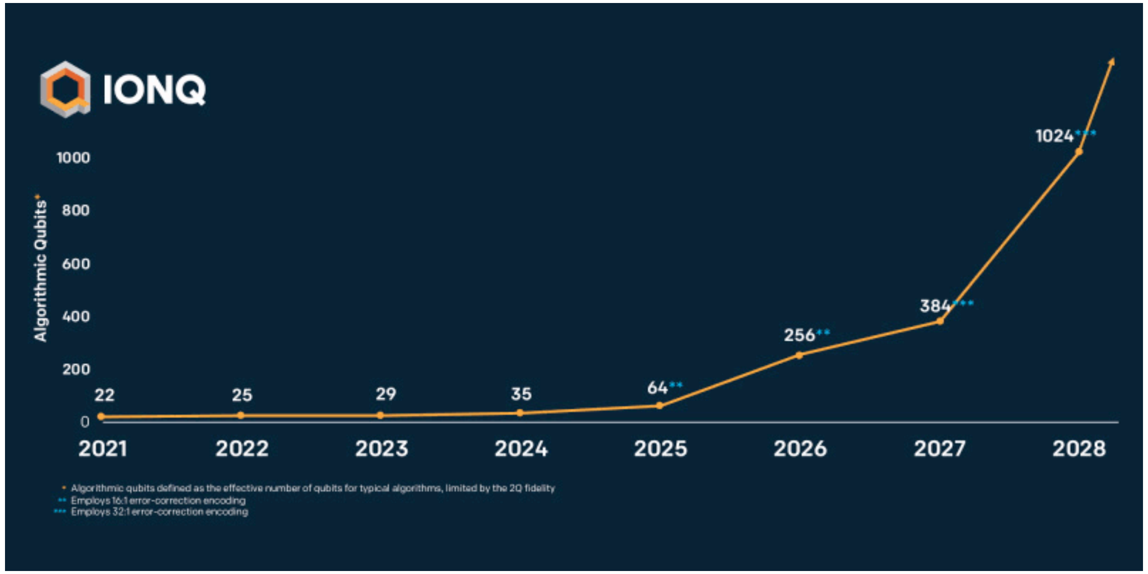
The game of Go has long been viewed as the most challenging of classic games for artificial intelligence owing to its enormous search space and the difficulty of evaluating board positions and moves. Here we introduce a new approach to computer Go that uses 'value networks' to evaluate board positions and 'policy networks' to select moves. These deep neural networks are trained by a novel combination of supervised learning from human expert games, and reinforcement learning from games of self-play. Without any lookahead search, the neural networks play Go at the level of state-of-the-art Monte Carlo tree search programs that simulate thousands of random games of self-play. We also introduce a new search algorithm that combines Monte Carlo simulation with value and policy networks. Using this search algorithm, our program AlphaGo achieved a 99.8% winning rate against other Go programs, and defeated the human European Go champion by 5 games to 0. This is the first time that a computer program has defeated a human professional player in the full-sized game of Go, a feat previously thought to be at least a decade away.

Brookhaven National Laboratory

2017~ QC hardware







I. Introduction
II. **Quantum Computing (QC)**
III. Quantum Machine Learning
IV. Applications-Classification
V. Applications-Sequential Learning
VI. Applications-Reinforcement Learning
VII. Conclusion and Outlook

Quantum Computing

- Classical computers: Classical bits 0 vs 1
- Quantum computers: Quantum bits (qubit) $|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle$ where α and β are complex numbers $\mathbb{C}$
- Quantum **entanglements**: A unique property of quantum physics —> No analog in the classical computer
- Famous algorithms:
    - Shor's algorithm: Can be used to break the state-of-the-art public key cryptography systems such as RSA
    - Grover's algorithm: Quadratic speedup in unstructured search

Brookhaven National Laboratory

- Designing a quantum algorithm is non-trivial task
- Even harder in the noisy quantum machines

Quantum States

Single Qubit State

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$|1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Two Qubit State

$$|0\rangle \otimes |0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

N **Qubit State**

$$\underbrace{|0\rangle \otimes |0\rangle \otimes \cdots \otimes |0\rangle}_{N} = \underbrace{\begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \cdots \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix}}_{N}$$

Quantum States

Density Operators

$$\rho = \sum_j p_j \left|\psi_j\right\rangle \left\langle\psi_j\right|$$

$\left|\psi_j\right\rangle$ **Basis state**

p_j **Probability**

Examples:

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \qquad \langle 0| = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

$$\downarrow$$

$$|0\rangle\langle 0| = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$$

Quantum Operations

$\boxed{X}$ $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$

$\boxed{Y}$ $\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$

$\boxed{Z}$ $\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$

$\boxed{H}$ $\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$

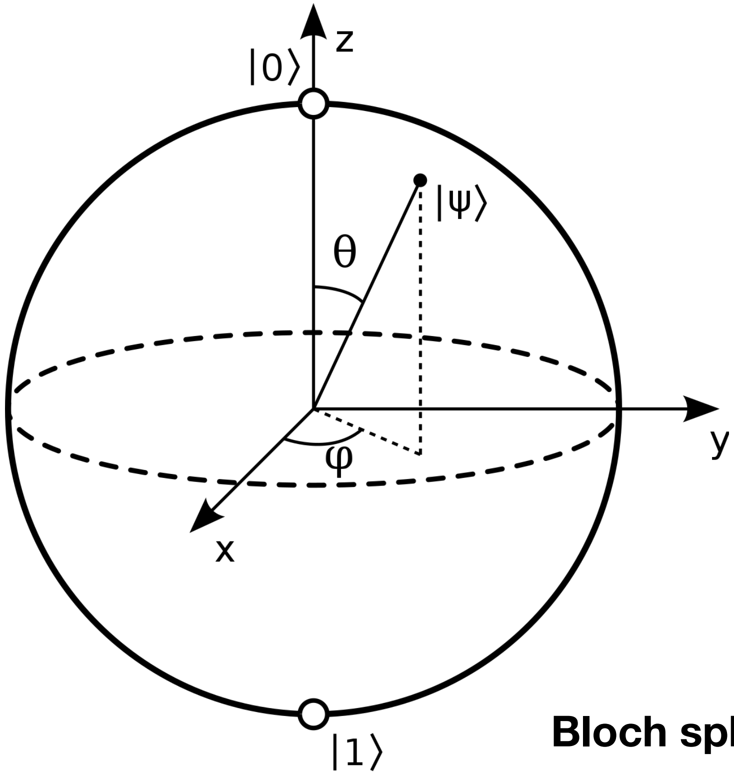
Example:

$|0\rangle$ —— $\boxed{X}$

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} = |1\rangle$$

Quantum Operations

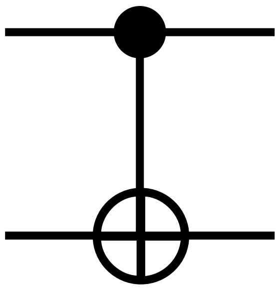
$$\boxed{R(\phi,\theta,\omega)} \qquad \begin{bmatrix} e^{-i(\phi+\omega)/2}\cos(\theta/2) & e^{-i(\phi-\omega)/2}\sin(\theta/2) \\ e^{-i(\phi-\omega)/2}\sin(\theta/2) & e^{i(\phi+\omega)/2}\cos(\theta/2) \end{bmatrix}$$



Bloch sphere

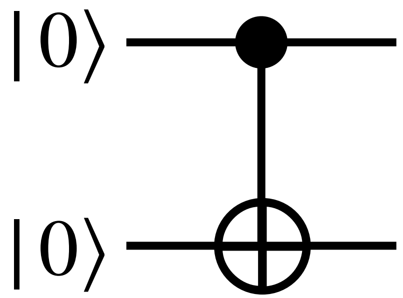
Quantum Operations

CNOT



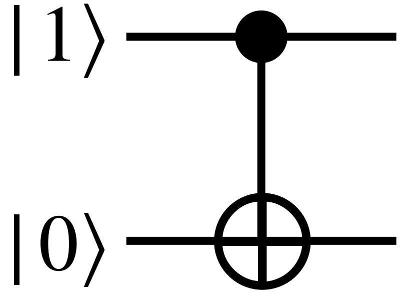
$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Result



$|0\rangle$

$|0\rangle$

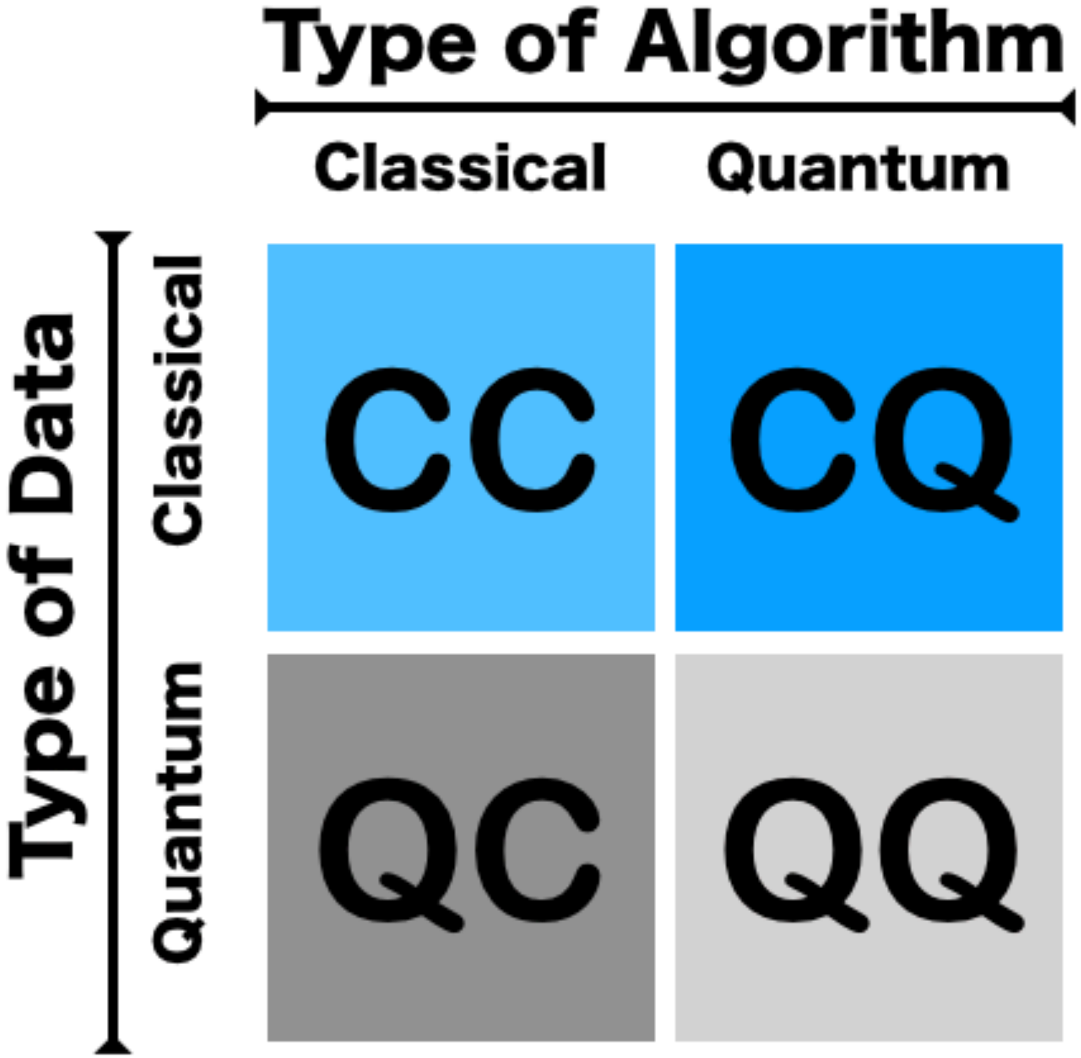


$|1\rangle$

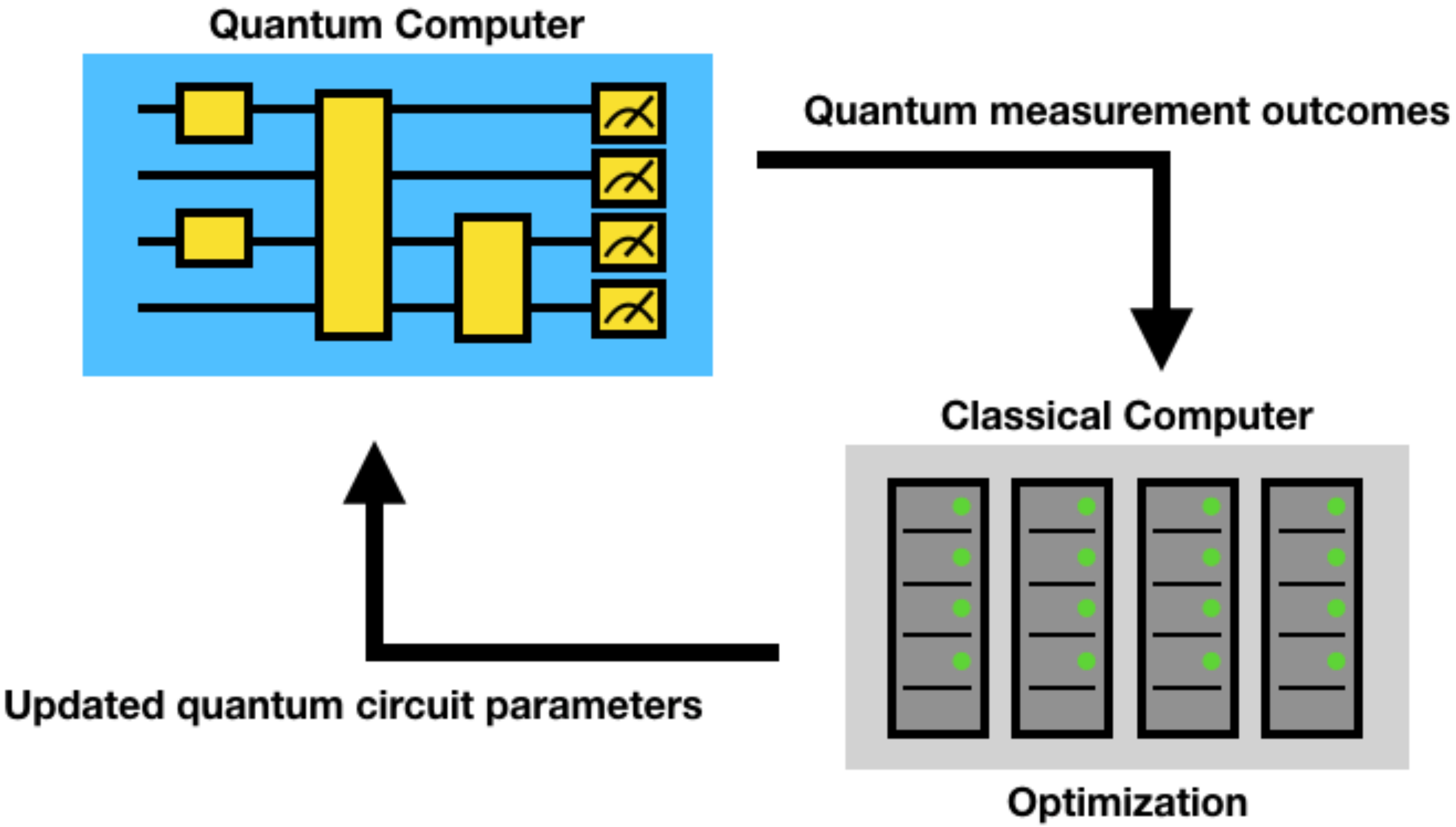
$|1\rangle$

Brookhaven National Laboratory

Quantum Machine Learning

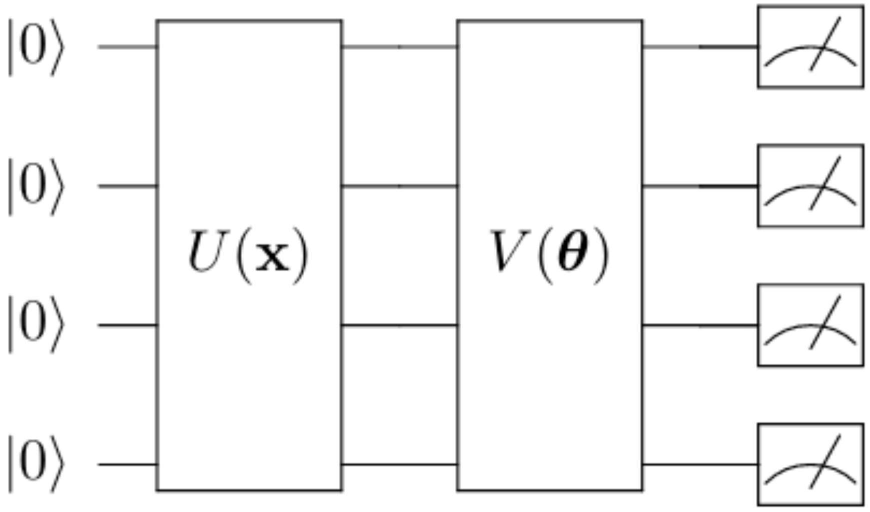


Hybrid Quantum-Classical Paradigm



Variational Quantum Circuits (VQC)

- Quantum circuits with **tunable** parameters.
- Subject to iterative optimization procedures (on classical computers).
- $U(\mathbf{x})$: encoding circuit
- $V(\theta)$: variational circuit
-  : measurement



Quantum Encoding and State Preparation

A general N qubit quantum state can be represented as:

$$|\psi\rangle = \sum_{(q_1,q_2,\cdots,q_N)\in\{0,1\}} c_{q_1,q_2,\cdots,q_N} \left|q_1\right\rangle \otimes \left|q_2\right\rangle \otimes \cdots \otimes \left|q_N\right\rangle$$

where $c_{q_1,\cdots,q_N} \in \mathbb{C}$ is the complex amplitude for each basis state and each $q_i \in \{0,1\}$

The total probability is equal to 1: $\sum_{(q_1,\cdots,q_N)\in\{0,1\}} \| c_{q_1,\cdots,q_N} \|^2 = 1$

Brookhaven National Laboratory

Quantum Encoding and State Preparation

Amplitude Encoding

Encode a vector $\left(\alpha_0, \cdots, \alpha_{2^n-1}\right)$ into a n-qubit quantum state:

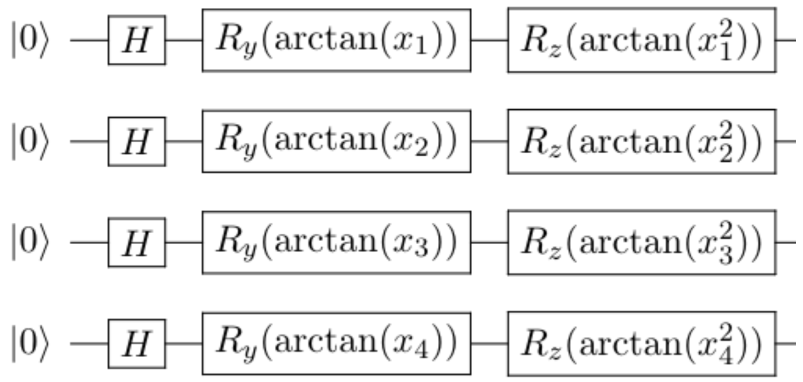
$$|\Psi\rangle = \alpha_0 |00\cdots0\rangle + \cdots + \alpha_{2^n-1} |11\cdots1\rangle$$

where α_i are real numbers and $\left(\alpha_0, \cdots, \alpha_{2^n-1}\right)$ is normalized

N-dimensional vector will require only $\log_2(N)$ qubits to encode

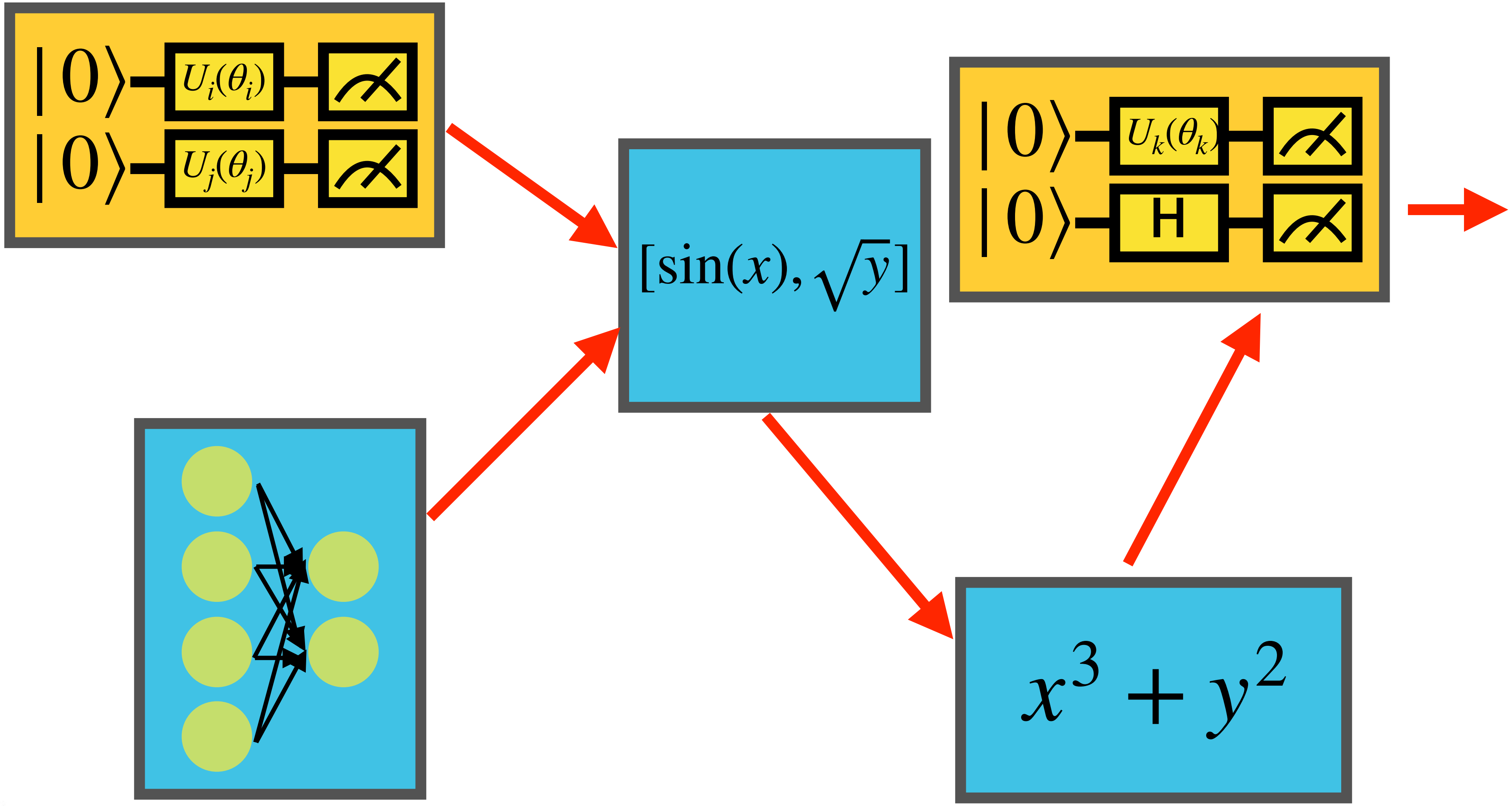
Variational Encoding

Input numbers $x_1 \cdots x_n$ are used as quantum rotation angles



Simpler implementation than amplitude encoding

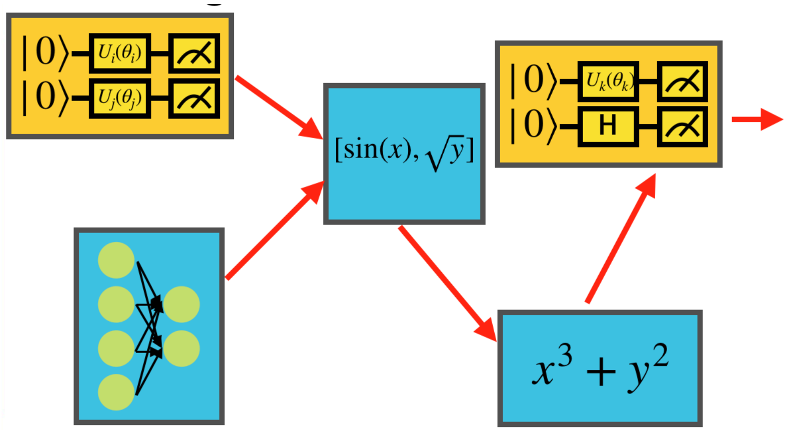
Interfacing with Classical ML



Brookhaven National Laboratory

Interfacing with Classical ML

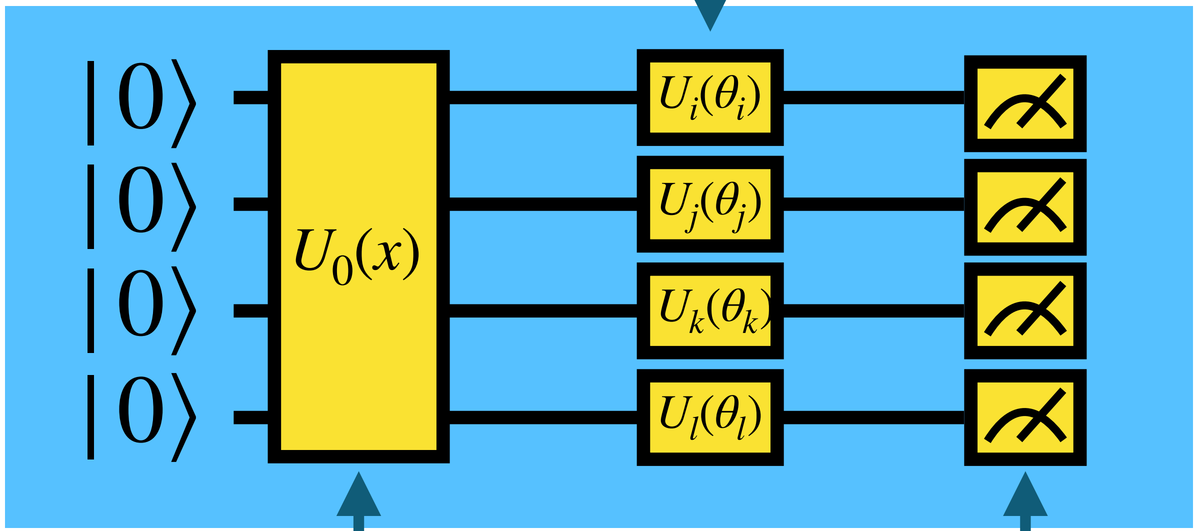
- Mixing *classical* and *quantum* computing components
- These classical and quantum nodes are arranged in a *directed acyclic graph* (DAG)
- The hybrid architecture is similar to the one in *deep learning* models
- The whole model can be trained with *backpropagation* method or other gradient-free methods such as evolutionary optimization
- The next question is “*How to calculate the gradient of a quantum node?*”



Brookhaven National Laboratory

Quantum Gradients

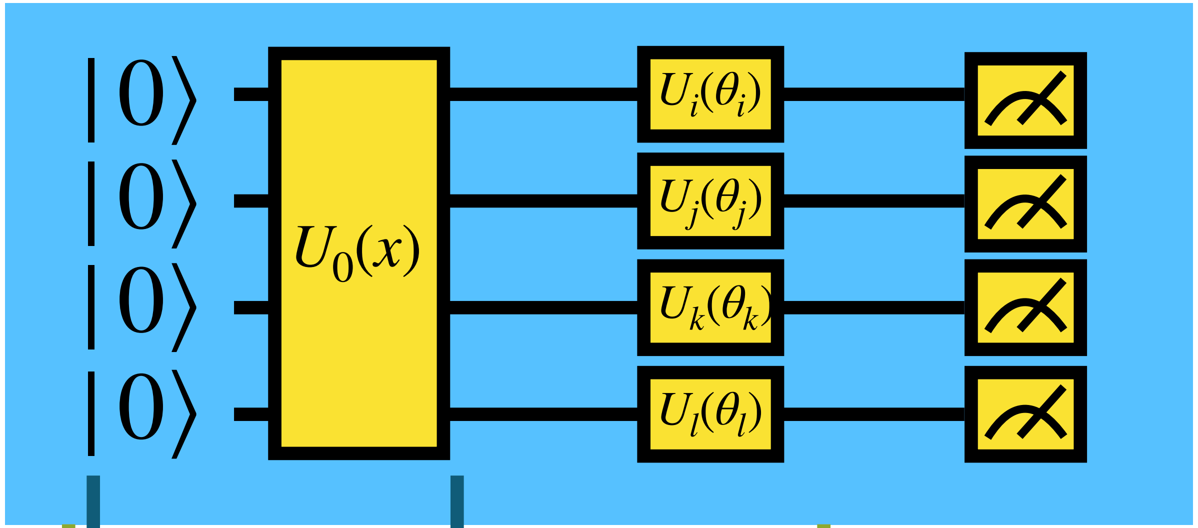
Learnable quantum circuit parameters



Quantum encoding / state preparation circuit

Quantum measurements

Quantum Gradients



$$U_0(x)\,|0\rangle$$

$$U_i\left(\theta_i\right)\,U_0(x)\,|0\rangle$$

$$\left\langle 0 \left| U_0^\dagger(x) U_i^\dagger\left(\theta_i\right) \hat{B} U_i\left(\theta_i\right) U_0(x) \right| 0 \right\rangle$$

Brookhaven National Laboratory

Quantum Gradients

$$f\left(x;\theta_i\right) = \left\langle 0 \left| U_0^\dagger(x) U_i^\dagger\left(\theta_i\right) \hat{B} U_i\left(\theta_i\right) U_0(x) \right| 0 \right\rangle = \left\langle x \left| U_i^\dagger\left(\theta_i\right) \hat{B} U_i\left(\theta_i\right) \right| x \right\rangle$$

- x: input value
- $U_0(x)$: encoding circuit
- i: circuit parameter index
- $U_i(x_i)$: single-qubit rotation generated by the Pauli operators

PRA 98, 032309 (2018)
PRA 99, 032331 (2019)

Quantum Gradients

The gradient of f with respect to the parameter θ_i is:

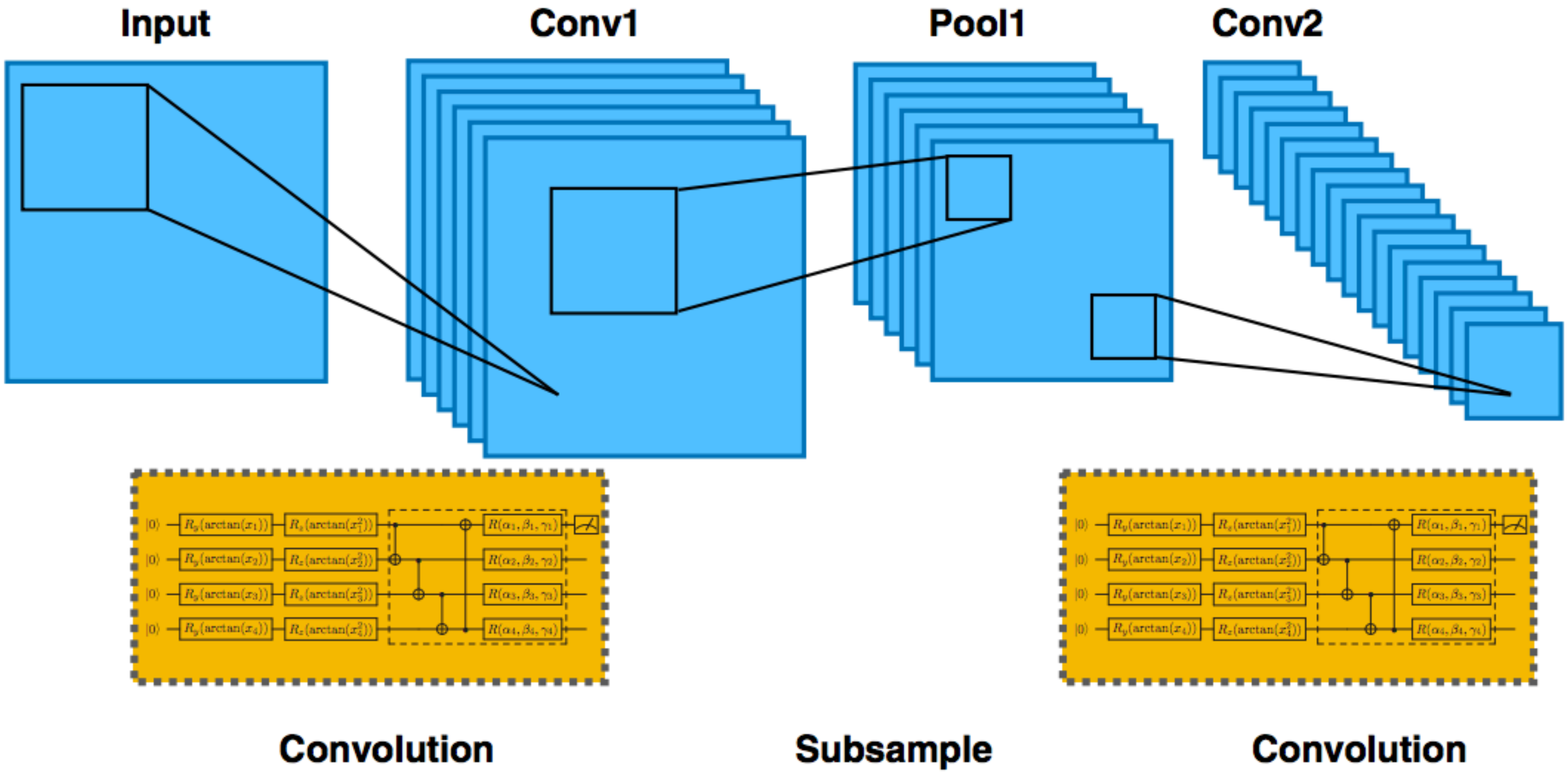
$$\nabla_{\theta_i} f(x;\theta_i) = \frac{1}{2}\left[f\left(x;\theta_i + \frac{\pi}{2}\right) - f\left(x;\theta_i - \frac{\pi}{2}\right)\right]$$

This value can be calculated via running two quantum circuits with shifted parameters, the so-called *parameter-shift* rule.

PRA 98, 032309 (2018)
PRA 99, 032331 (2019)

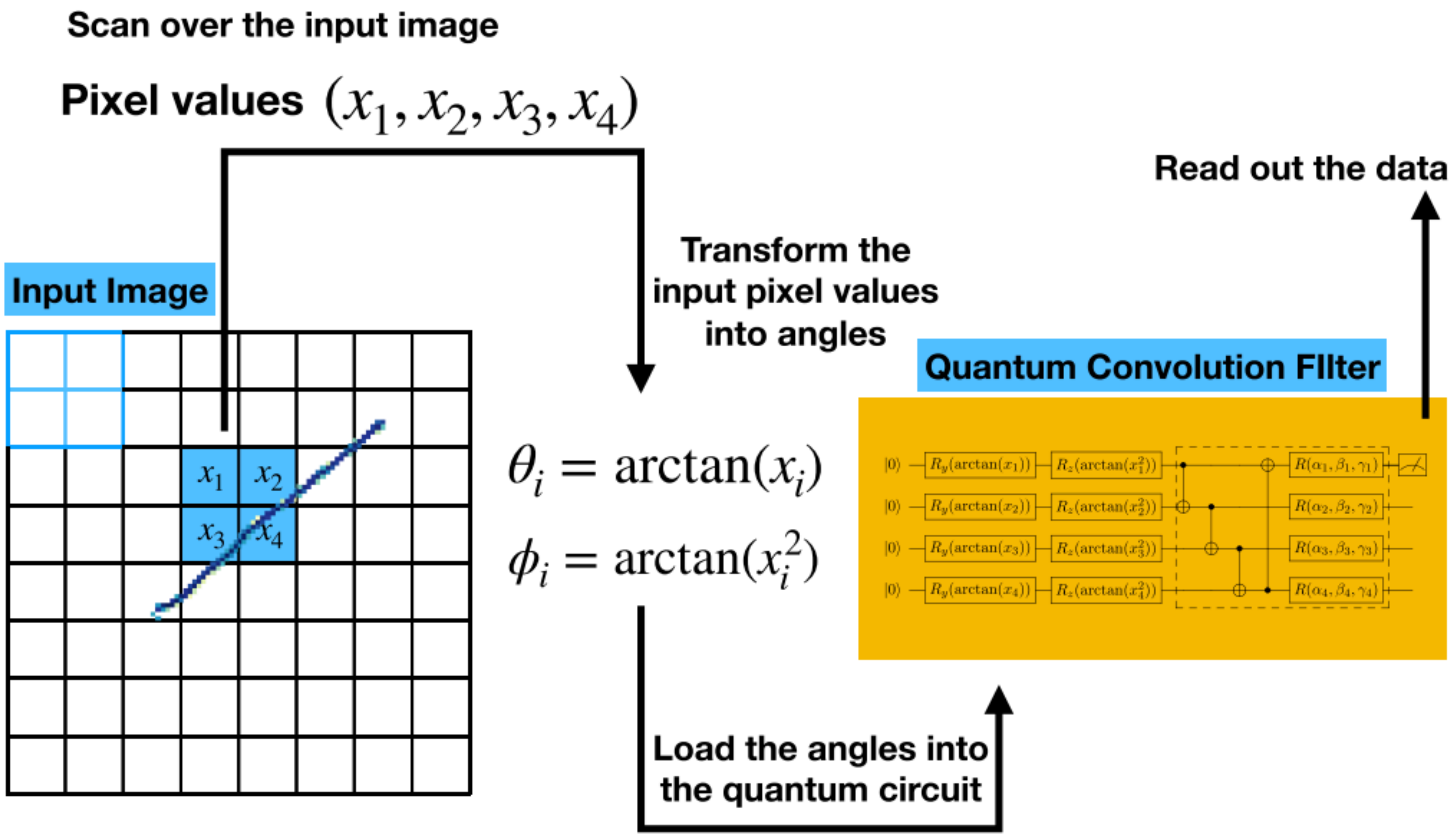
Brookhaven National Laboratory

Quantum Convolutional Neural Network

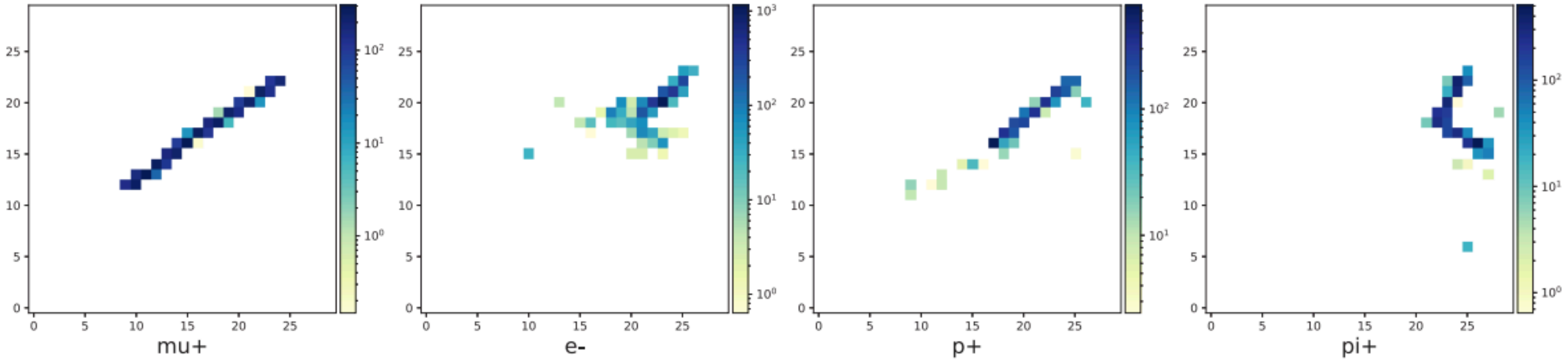


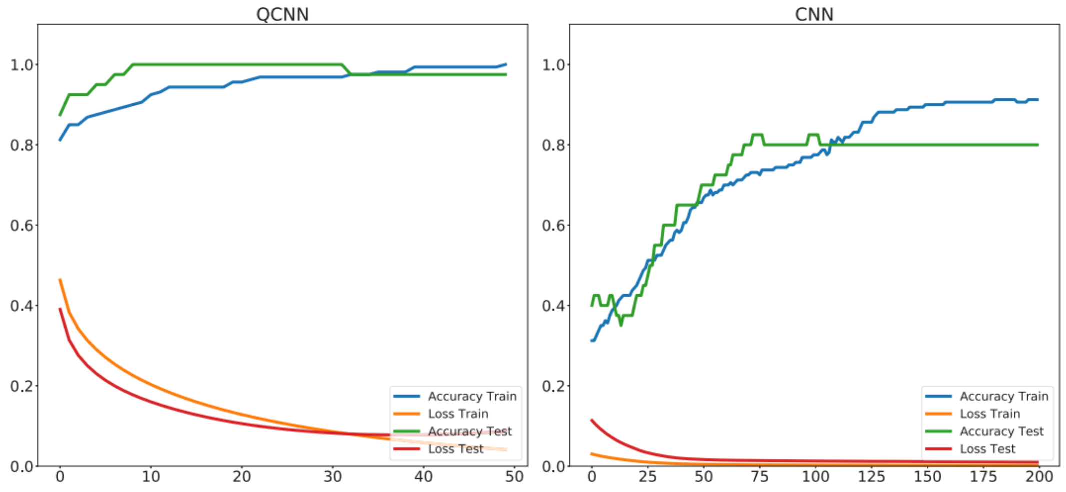
Chen, S. Y. C., Wei, T. C., Zhang, C., Yu, H., & Yoo, S. (2022). Quantum convolutional neural networks for high energy physics data analysis. *Physical Review Research*, *4*(1), 013231.

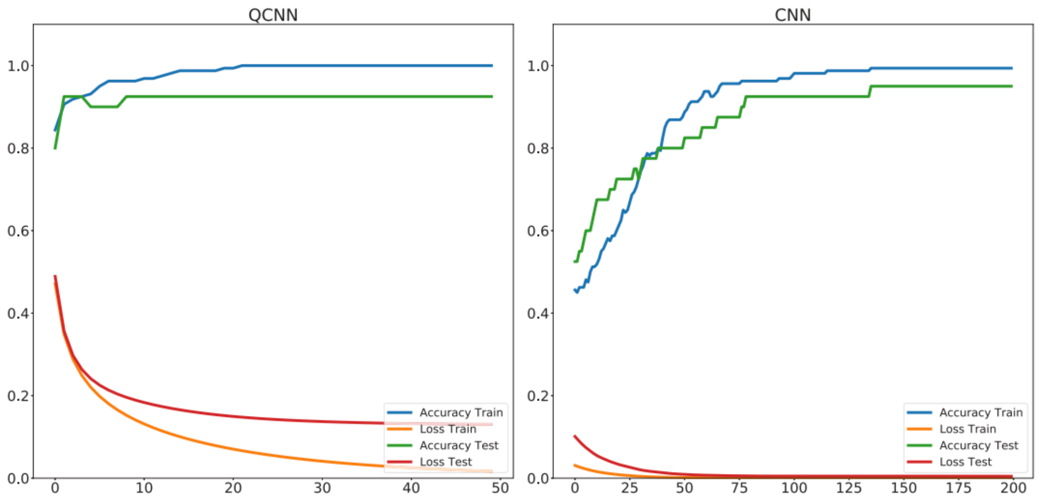
Quantum Convolutional Neural Network



Brookhaven National Laboratory







mu+ vs e-

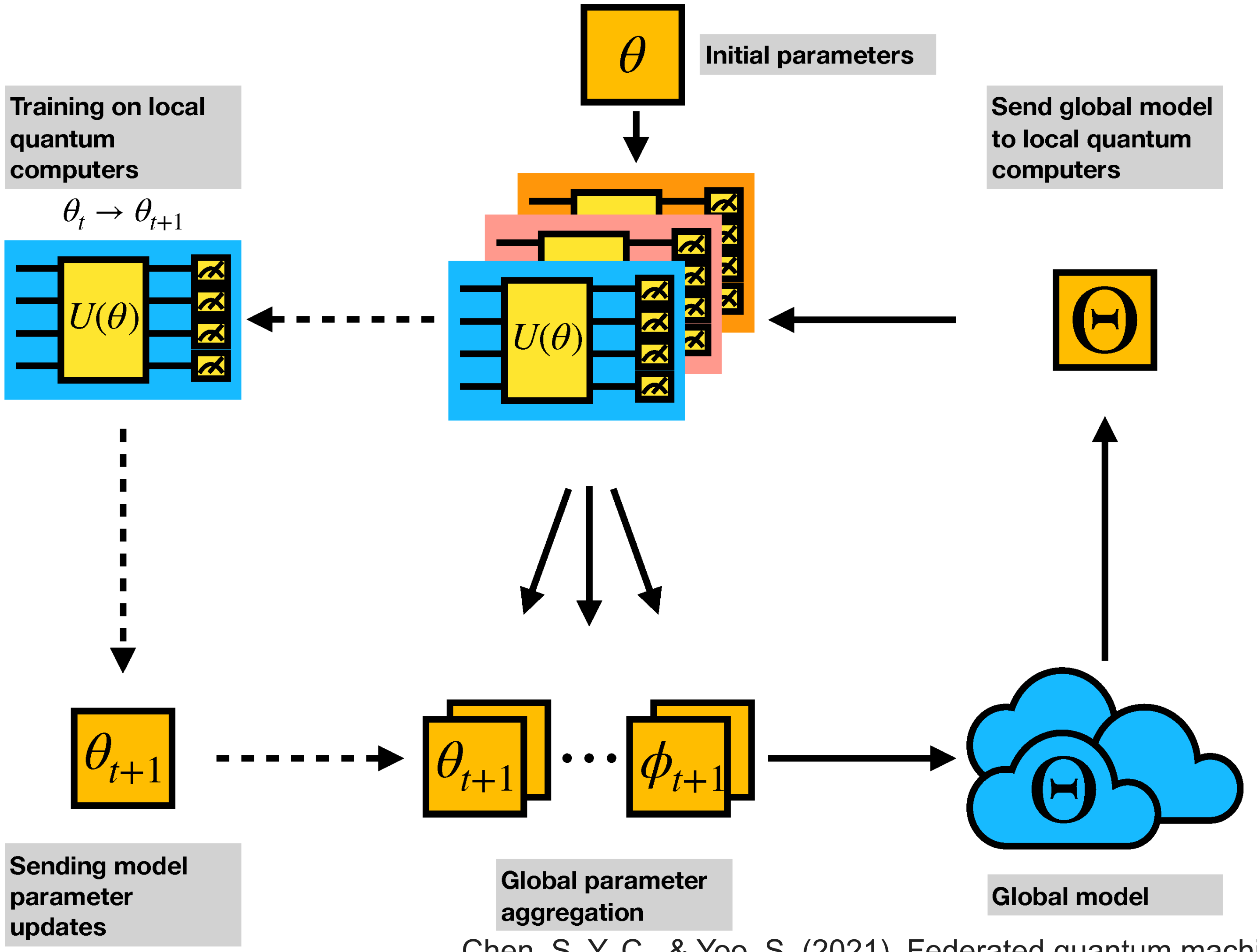
mu+ vs p

TABLE I. The comparison of model architectures between QCNN and CNN used in this paper.

	First conv layer (# of channels)	Filter size	Second conv layer (# of channels)	Filter size	Classical part (# of params)	Total number of params
QCNN	1	3 × 3	1	2 × 2	394	472
CNN	4	5 × 5	2	5 × 5	198	498

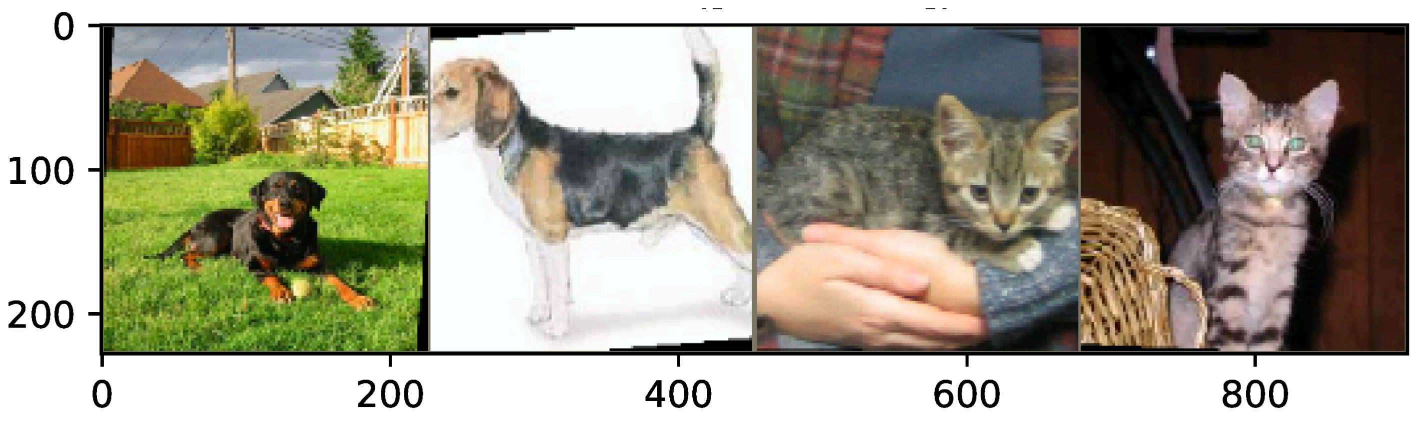
Brookhaven National Laboratory

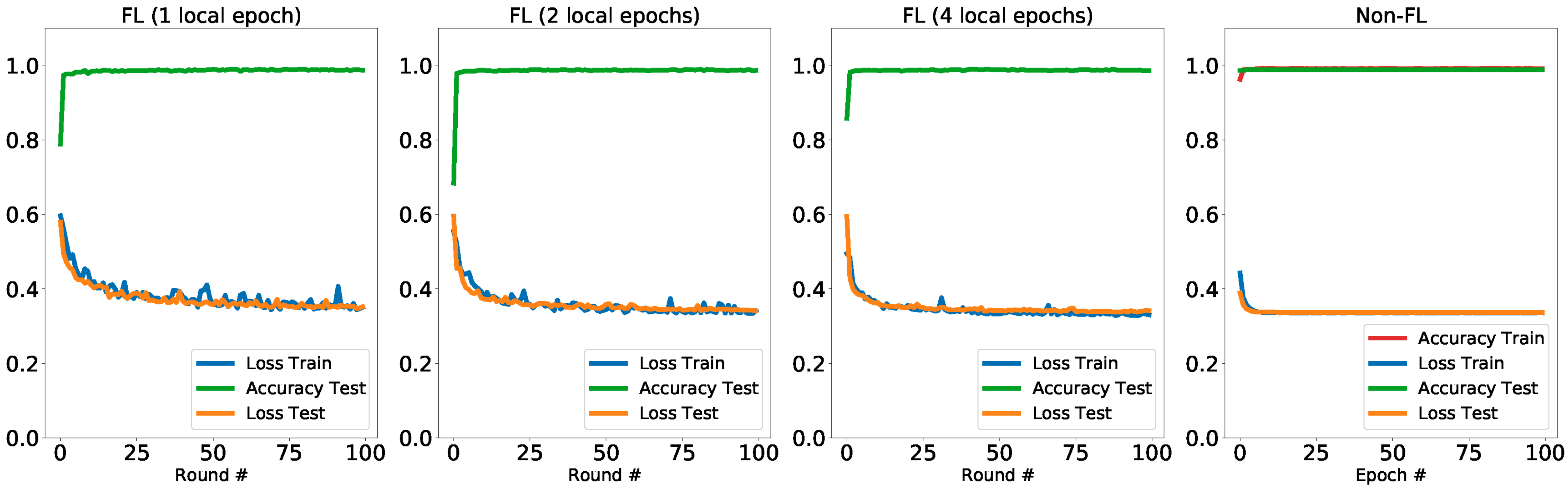
Federated Quantum Machine Learning

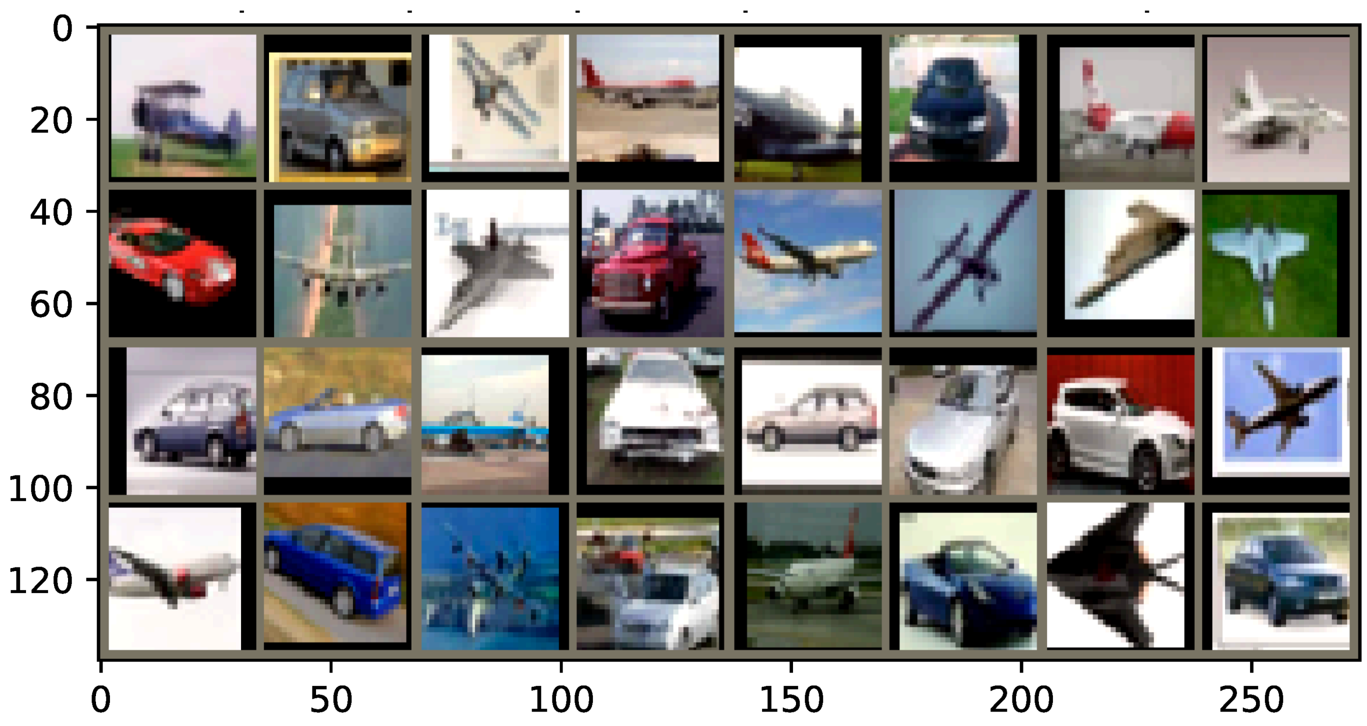


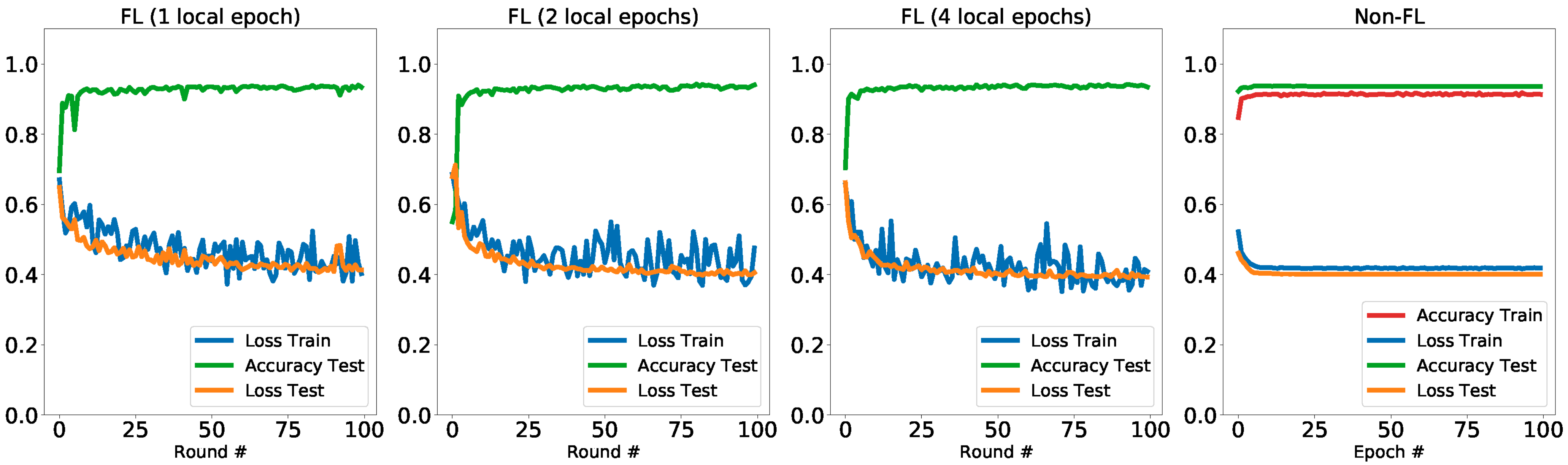
Chen, S. Y. C., & Yoo, S. (2021). Federated quantum machine learning. *Entropy*, *23*(4), 460.

Federated Quantum Machine Learning


0
100
200
0
200
400
600
800


FL (1 local epoch)
FL (2 local epochs)
FL (4 local epochs)
Non-FL
1.0
0.8
0.6
0.4
0.2
0.0
0
25
50
75
100
Round #
Epoch #
Loss Train
Accuracy Test
Loss Test
Accuracy Train





Brookhaven National Laboratory

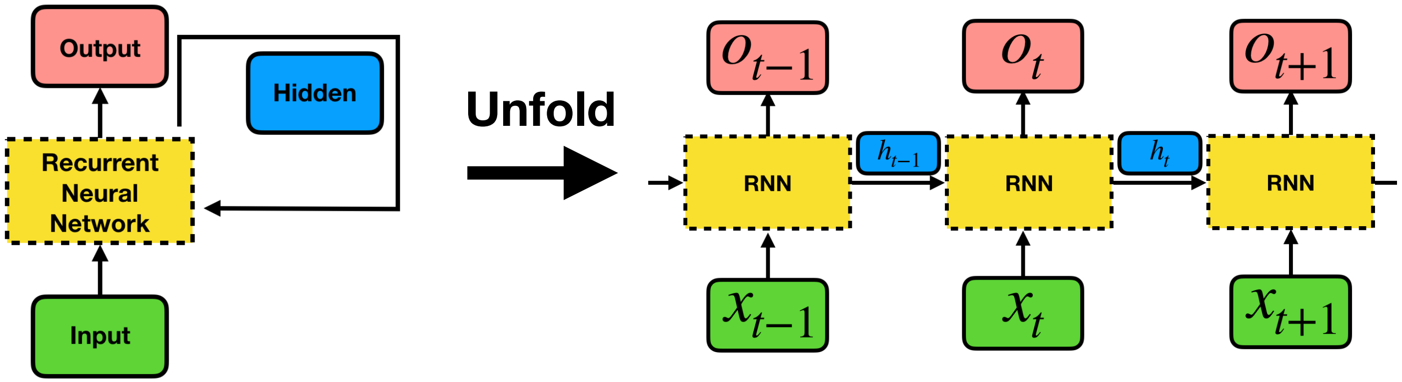
I. Introduction
II. Quantum Computing (QC)
III. Quantum Machine Learning
IV. Applications-Classification
V. Applications-Sequential Learning
VI. Applications-Reinforcement Learning
VII. Conclusion and Outlook

Brookhaven National Laboratory

Sequential Learning: Motivation

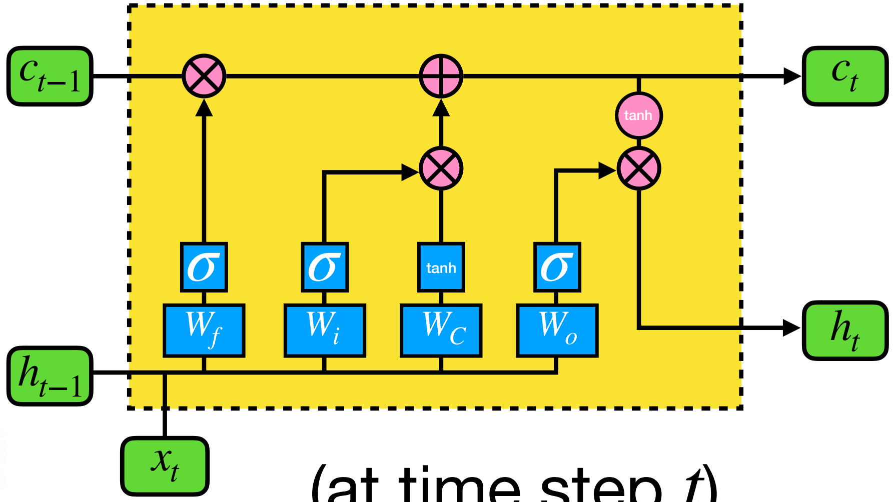
- Natural language processing: question answering, machine translation, …
- Physical dynamics: quantum dynamics, quantum optimal control, …
- Reinforcement learning (RL) agents also need a **memory**
- Goal #1: to predict time series accurately
- Goal #2: to explore if quantum computers could deliver better outcome

Recurrent Neural Network (RNN) and LSTM



Cons of RNN:
1. not suitable for long memory time
2. not easy to train (vanishing gradient)

(Classical) Long short-term memory (LSTM)



(at time step t)

c_t **: Cell state**

h_t **: Hidden state**

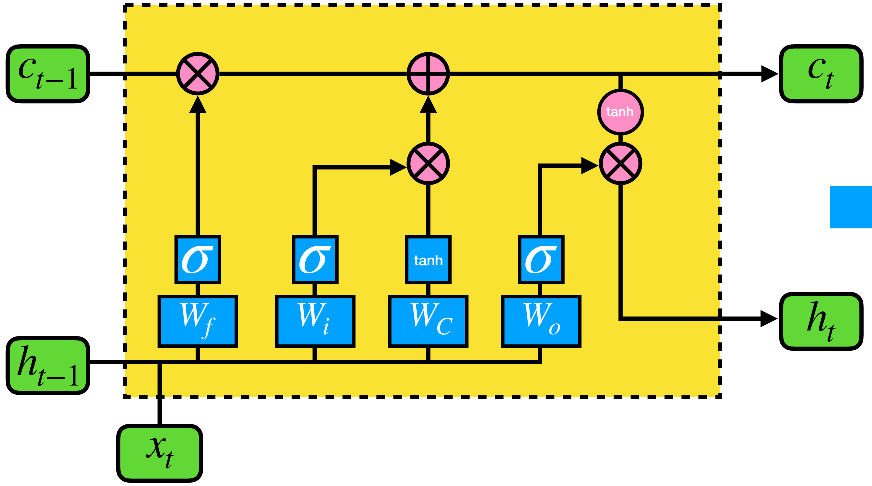
x_t **: Input**

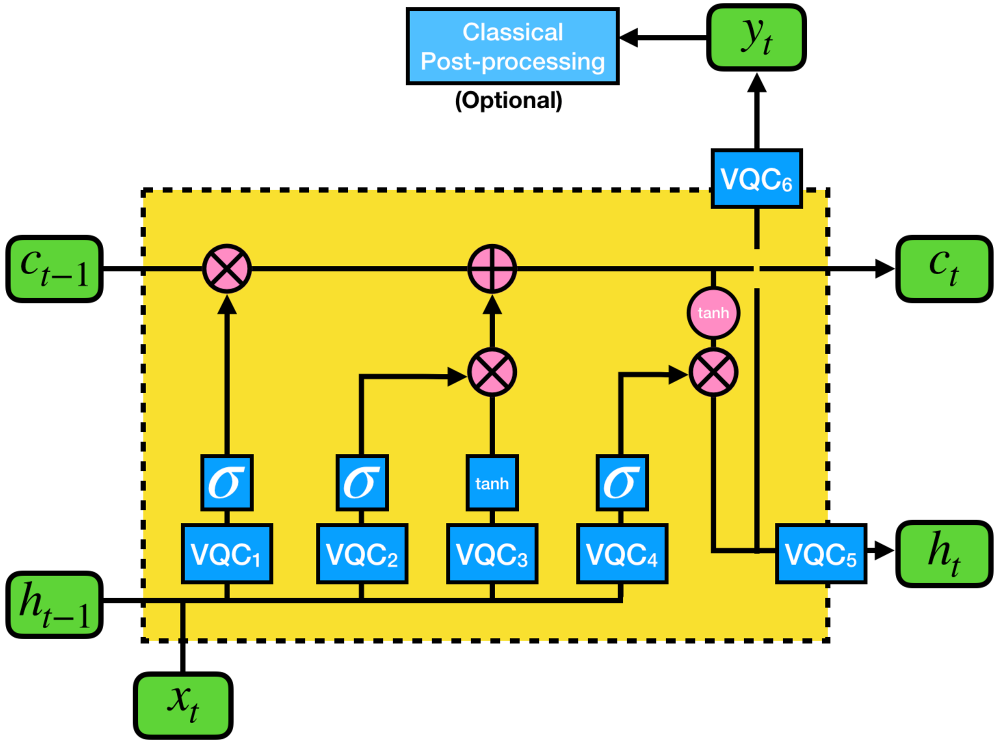
$\{W\}$**: Neural networks**

Use nonlinear activation functions (σ & tanh)

Quantum LSTM (QLSTM)

- Replace the classical neural networks with variational quantum circuits (VQC)
- Other components (input, output, activation functions, internal states) are still kept “classical” to avoid no-clone



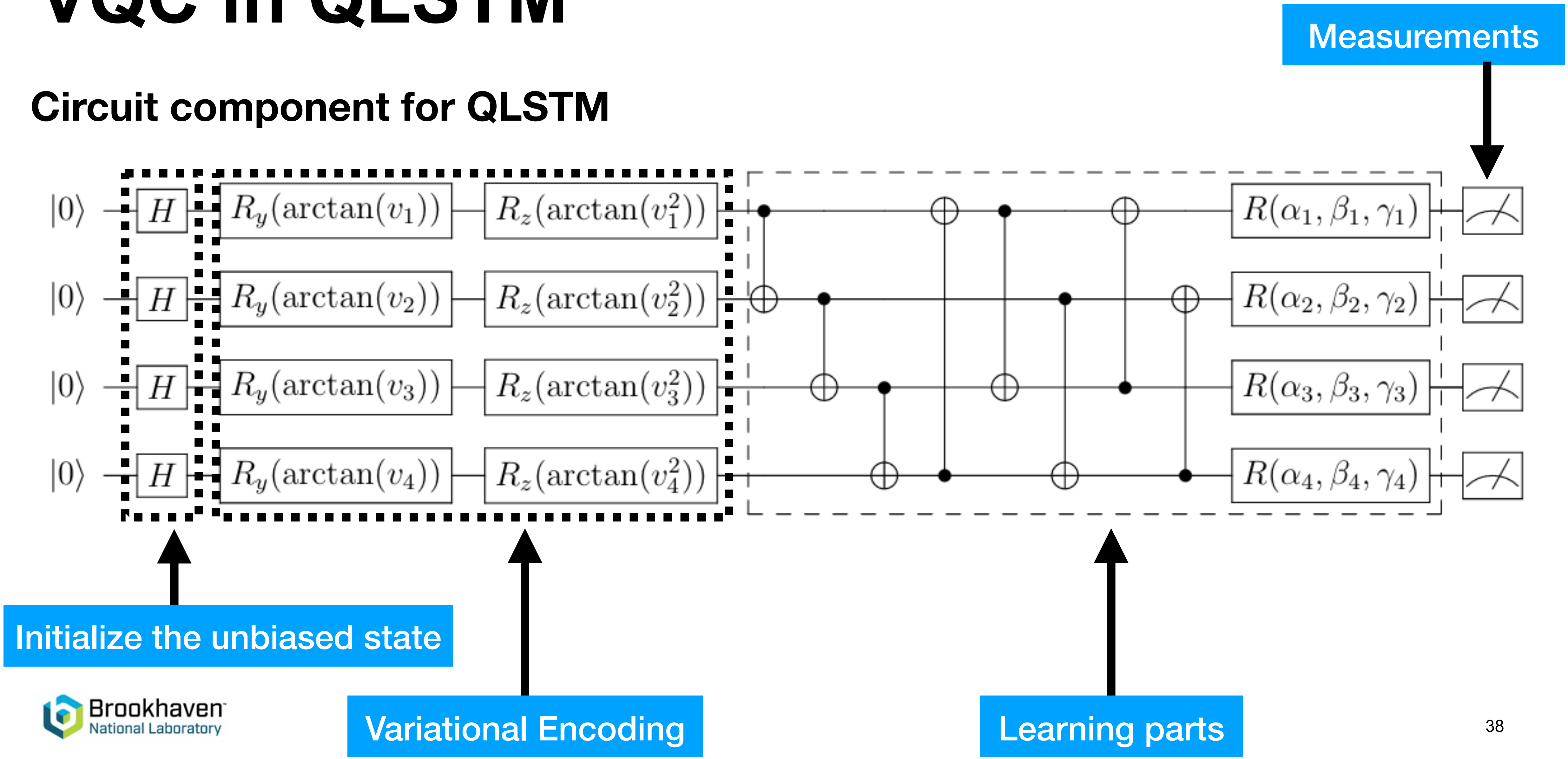


Brookhaven National Laboratory

(at time step t)

VQC in QLSTM

Circuit component for QLSTM



Experiments

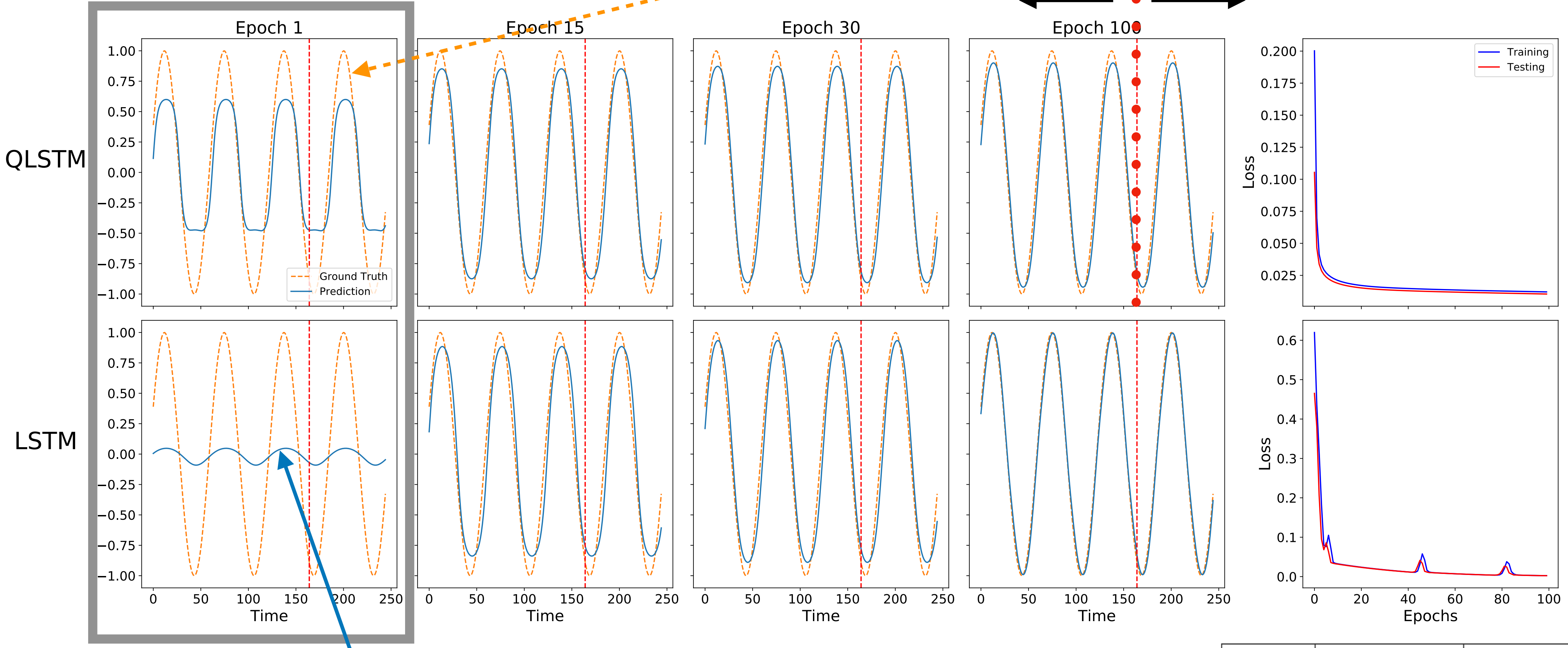
- Goal is to model/predict time series in a sliding window fashion
- Given 4 points from previous time steps $x_{t-4}, x_{t-3}, x_{t-2}, x_{t-1}$, predict x_t
- The time series data is loaded into the (Q)LSTM sequentially
- In each time step t, the concatenation of x_t and hidden state from previous time step h_{t-1}, $[x_t, h_{t-1}]$ is loaded into the quantum circuit
- Compare QLSTM with LSTM that has roughly the same number of params
- Loss function = Mean Square Error (MSE)
- Optimizer = RMSProp
- Noiseless quantum simulator

$$\textbf{time series} = \underbrace{\boxed{x_1, x_2, x_3, x_4}}_{\textbf{given}}, \overbrace{\boxed{x_5}}^{\textbf{predict}}, x_6, x_7, \cdots$$

Test case: $\sin(t)$

ground truth

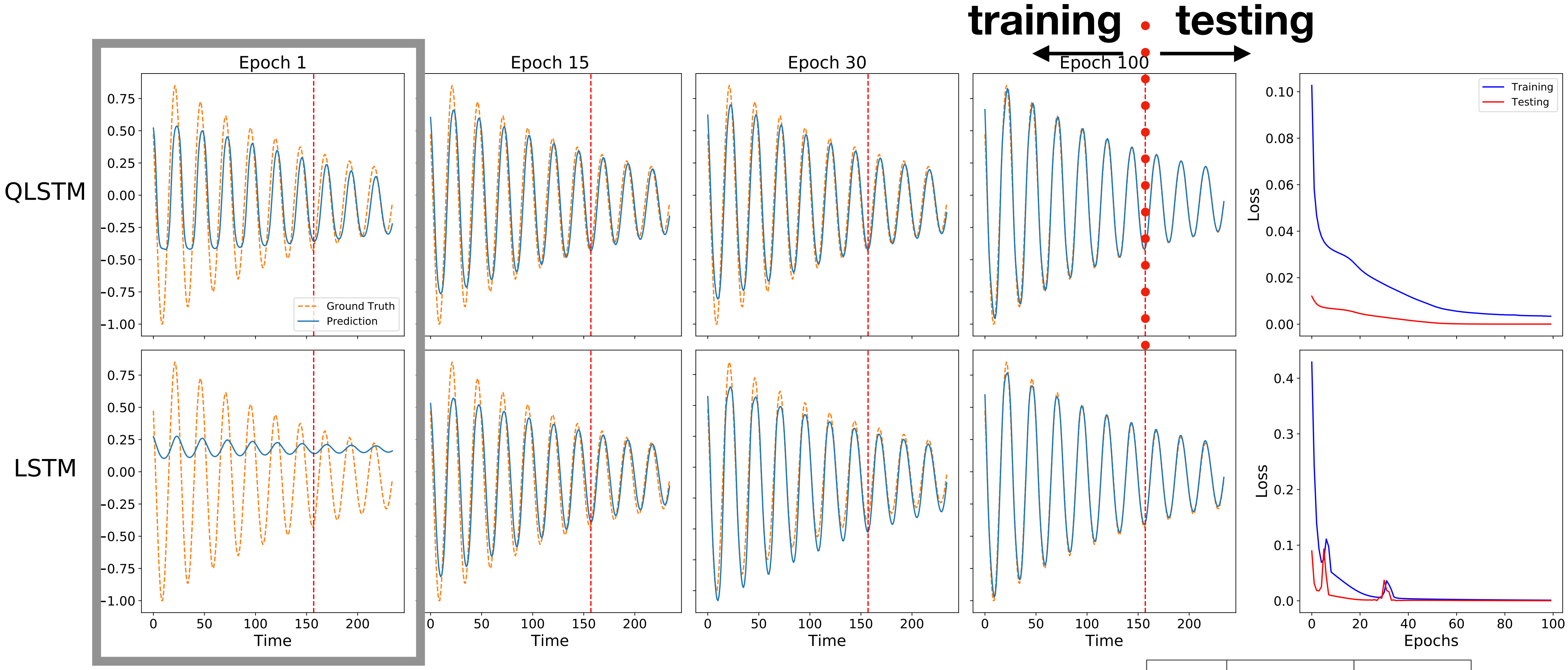
training testing



prediction

	Training Loss	Testing Loss
QLSTM	1.89×10^{-2}	1.69×10^{-2}
LSTM	2.86×10^{-2}	2.81×10^{-2}

Test case: damped oscillation

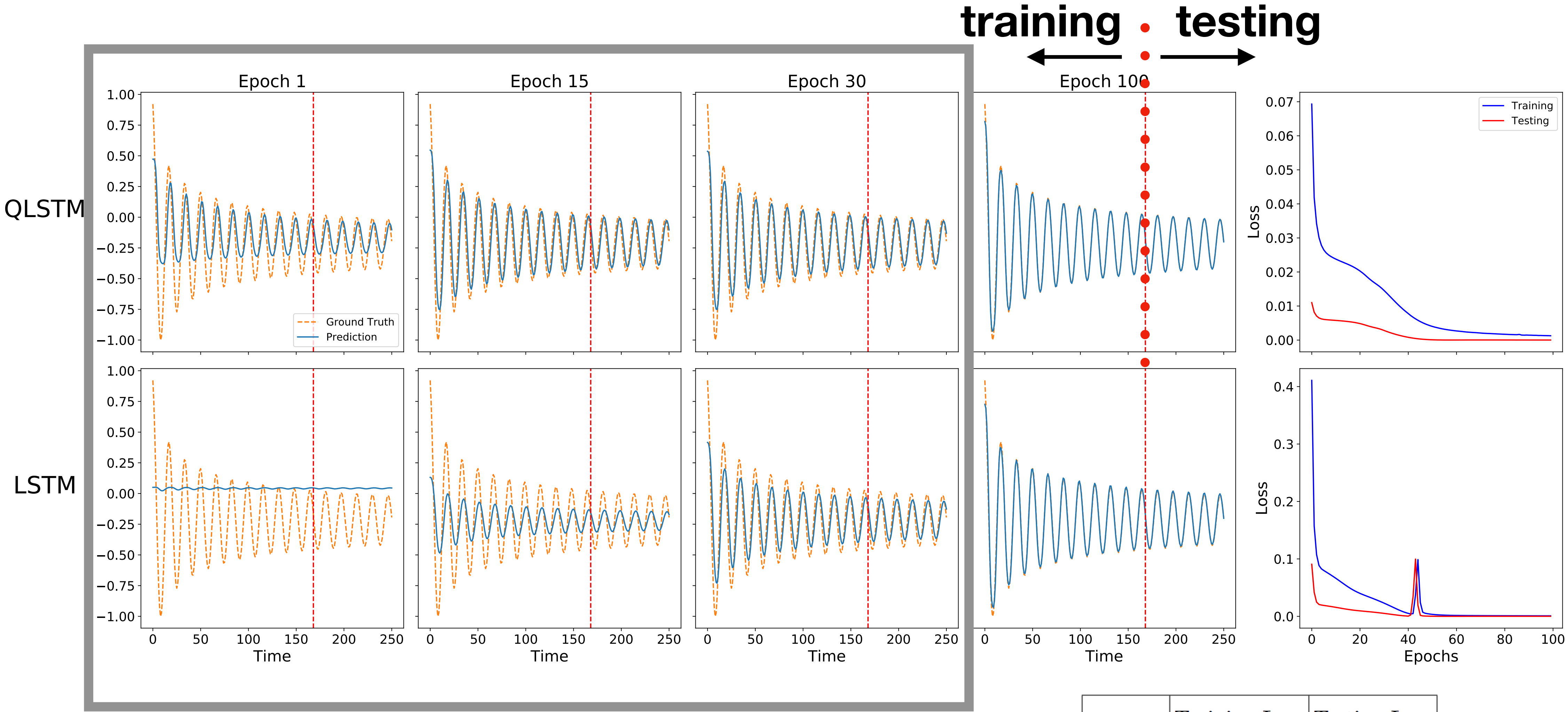


$$\frac{d^2\theta}{dt^2} + \frac{b}{m}\frac{d\theta}{dt} + \frac{g}{L}\sin\theta = 0$$

	Training Loss	Testing Loss
QLSTM	2.92×10^{-2}	6×10^{-3}
LSTM	3.15×10^{-2}	5×10^{-3}

Brookhaven National Laboratory

Test case: Bessel function $J_2(t)$

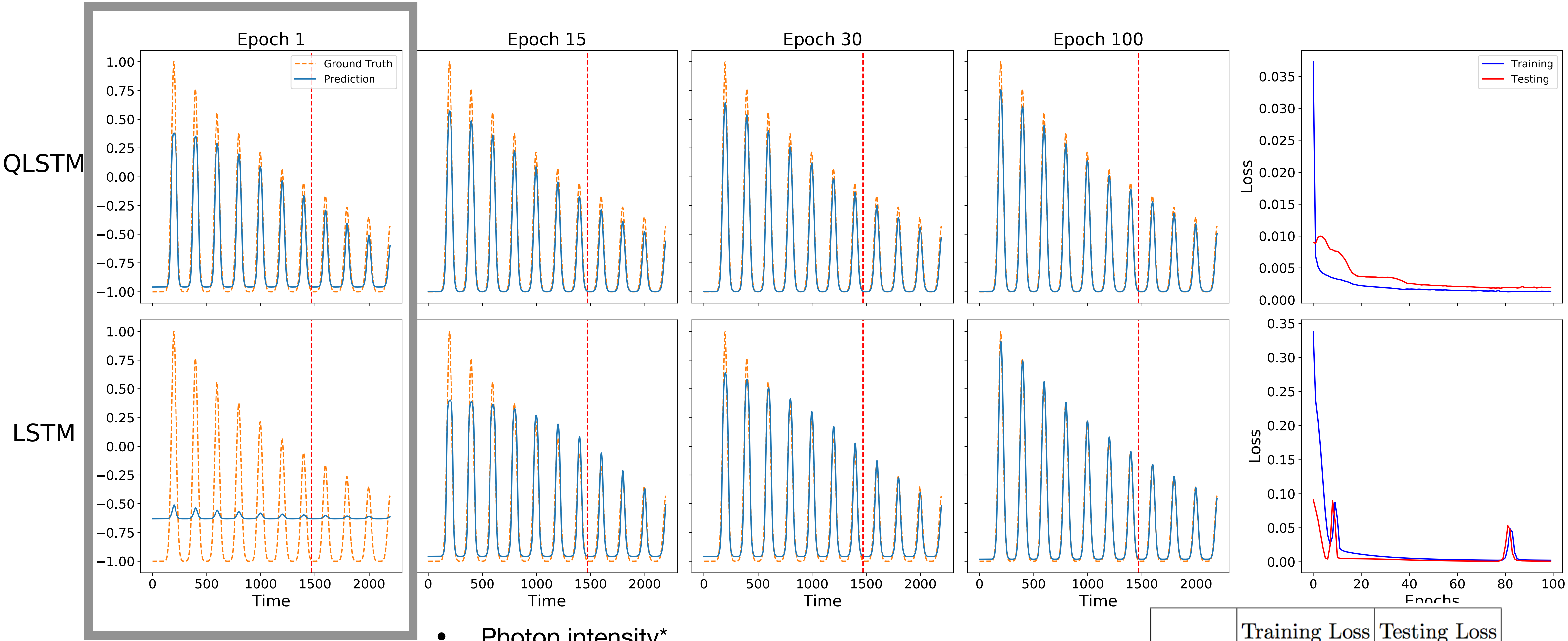


$$J_\alpha(x) = \sum_{m=0}^{\infty} \frac{(-1)^m}{m!\Gamma(m+\alpha+1)} \left(\frac{x}{2}\right)^{2m+\alpha}$$

	Training Loss	Testing Loss
QLSTM	2.26×10^{-2}	5.5×10^{-3}
LSTM	5.43×10^{-2}	1.28×10^{-2}

Brookhaven National Laboratory

Test case: photon trapping



- Photon intensity*
- Example of non-Markovian quantum dynamics (due to delayed quantum feedback)

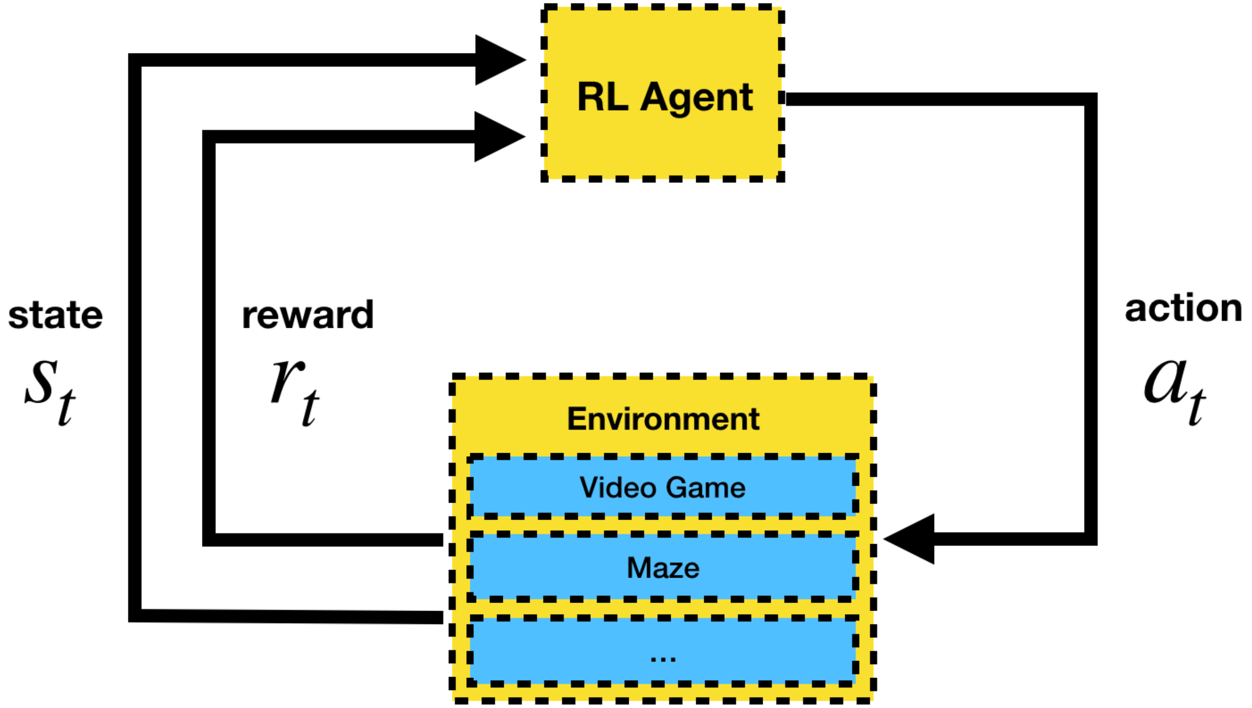
*See, e.g., Tufarelli et al, PRA 87, 013820 (2013)

	Training Loss	Testing Loss
QLSTM	2.88×10^{-3}	5.7×10^{-3}
LSTM	1.44×10^{-2}	4.7×10^{-3}

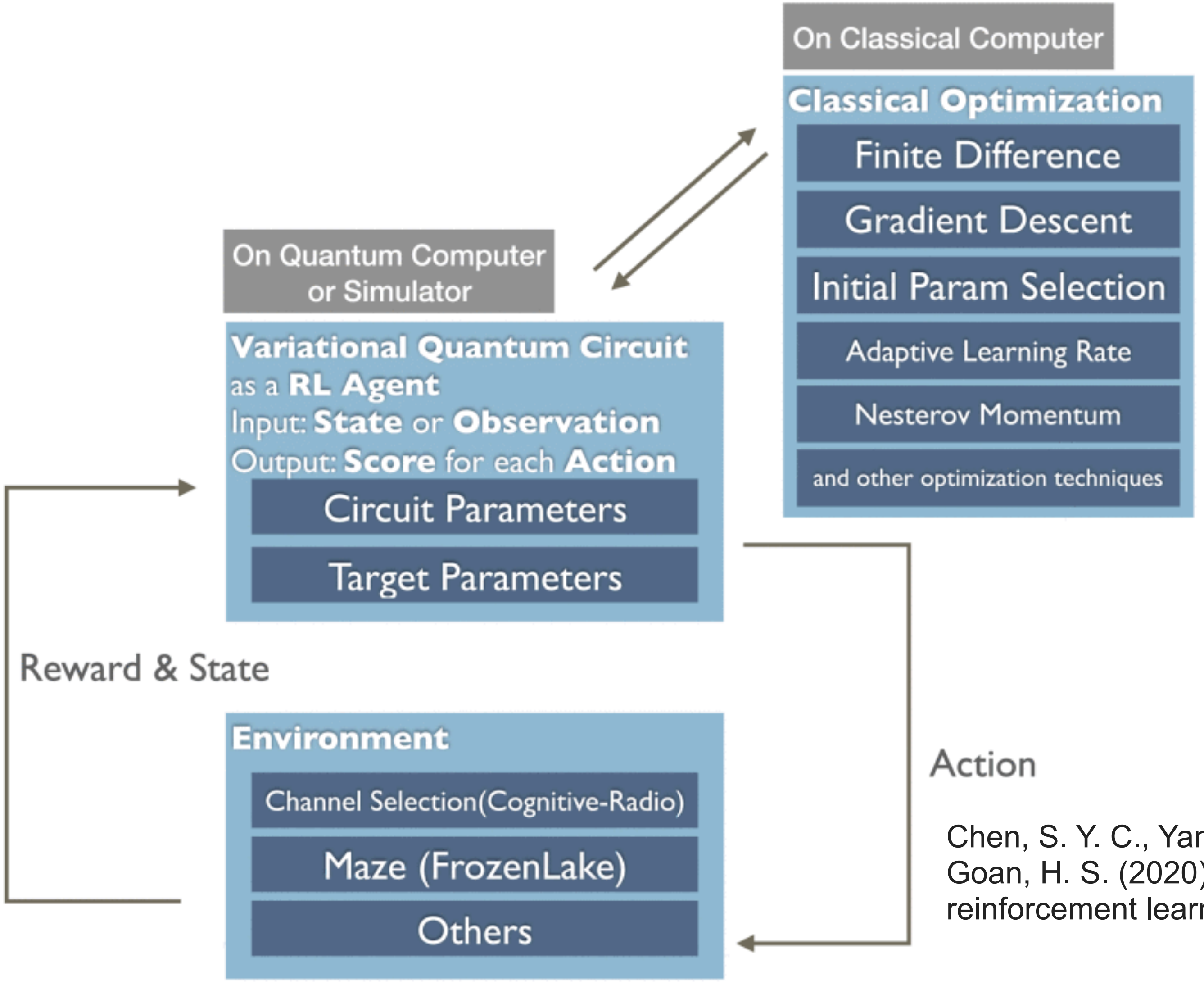
Brookhaven National Laboratory

Reinforcement Learning (RL)

- RL: An agent interacts with an ***environment*** $\mathscr{E}$ over a number of discrete time steps.
- The agent receives ***state*** or ***observation*** s_t and then chooses an ***action*** a_t from a set of actions $\mathscr{A}$ according to its ***policy*** π.
- Goal: Maximize the ***total discounted return*** $R_t = \sum_{t'=t}^{T} \gamma^{t'-t} r_{t'}$



Quantum Reinforcement Learning



Chen, S. Y. C., Yang, C. H. H., Qi, J., Chen, P. Y., Ma, X., & Goan, H. S. (2020). Variational quantum circuits for deep reinforcement learning. *IEEE Access*, *8*, 141007-141024.

Brookhaven National Laboratory

Quantum Deep Q-Learning

Algorithm 1 Variational Quantum Deep Q Learning

Initialize replay memory $\mathcal{D}$ to capacity N
Initialize action-value function quantum circuit Q with random parameters
for episode $= 1, 2, \ldots, M$ **do**
  Initialise state s_1 and encode into the quantum state
  for $t = 1, 2, \ldots, T$ **do**
    With probability ϵ select a random action a_t
    otherwise select $a_t = \max_a Q^*(s_t, a; \theta)$ from the output of the quantum circuit
    Execute action a_t in emulator and observe reward r_t and next state s_{t+1}
    Store transition (s_t, a_t, r_t, s_{t+1}) in $\mathcal{D}$
    Sample random minibatch of transitions $\left(s_j, a_j, r_j, s_{j+1}\right)$ from $\mathcal{D}$
    Set $y_j = \begin{cases} r_j & \text{for terminal } s_{j+1} \\ r_j + \gamma \max_{a'} Q(s_{j+1}, a'; \theta) & \text{for non-terminal } s_{j+1} \end{cases}$
    Perform a gradient descent step on $\left(y_j - Q(s_j, a_j; \theta)\right)^2$
  end for
end for

Brookhaven National Laboratory

Chen, S. Y. C., Yang, C. H. H., Qi, J., Chen, P. Y., Ma, X., & Goan, H. S. (2020). Variational quantum circuits for deep reinforcement learning. *IEEE Access*, *8*, 141007-141024.

Quantum Deep Q-Learning

- Env: FrozenLake
- 16 discrete states
- 4 actions

S	F	F	F
F	H	F	H
F	F	F	H
H	F	F	G

(a)

S	F	H	F
F	F	F	H
F	F	F	H
H	F	F	G

(b)

S	F	F	F
F	F	F	H
F	H	F	H
H	F	F	G

(c)

Location	Reward
HOLE	-0.2
GOAL	1.0
OTHER	-0.01

Chen, S. Y. C., Yang, C. H. H., Qi, J., Chen, P. Y., Ma, X., & Goan, H. S. (2020). Variational quantum circuits for deep reinforcement learning. *IEEE Access*, *8*, 141007-141024.

Environment with 16 states.

States numbered as 0~15

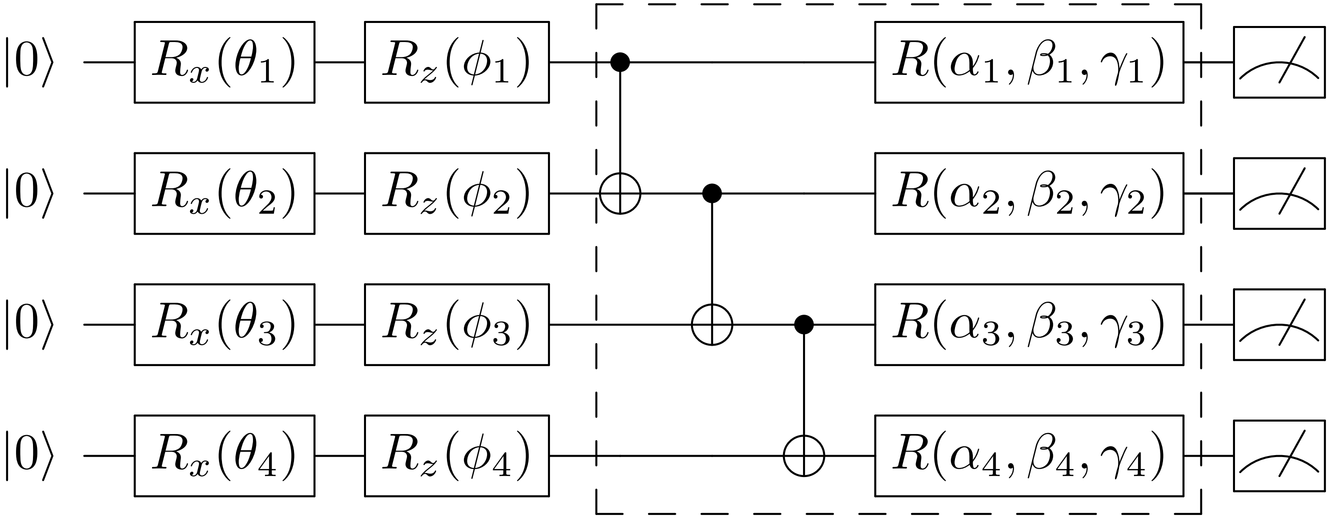
Example: State 12 : 1100 ->1,1,0,0

Rotation :

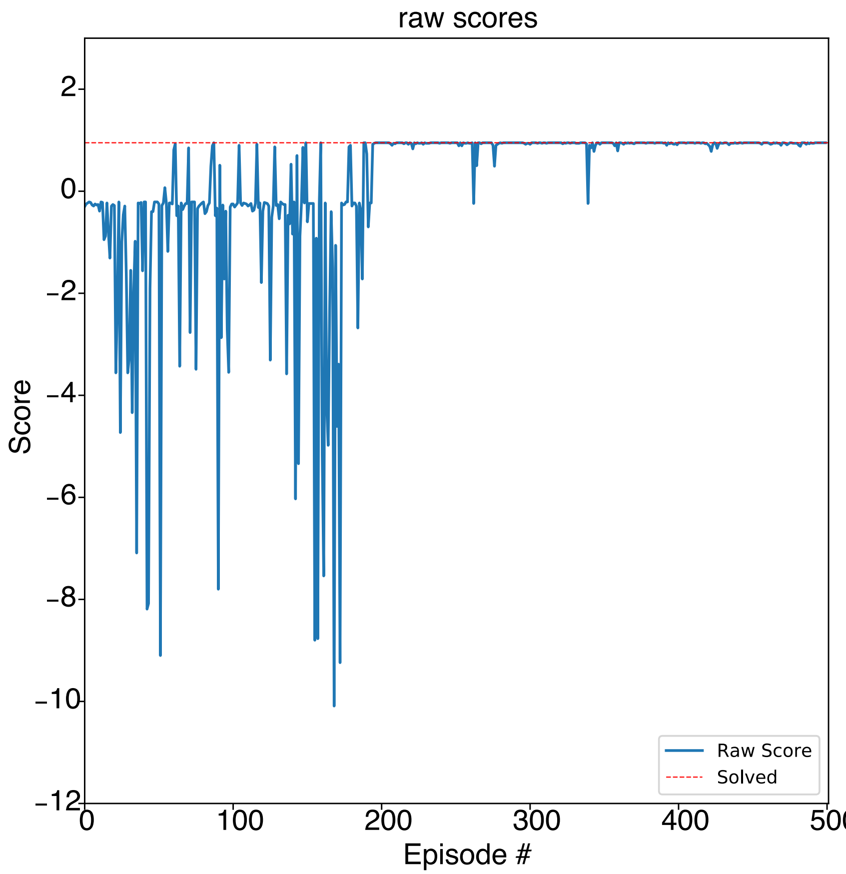
$$\theta_i = \pi \times b_i$$

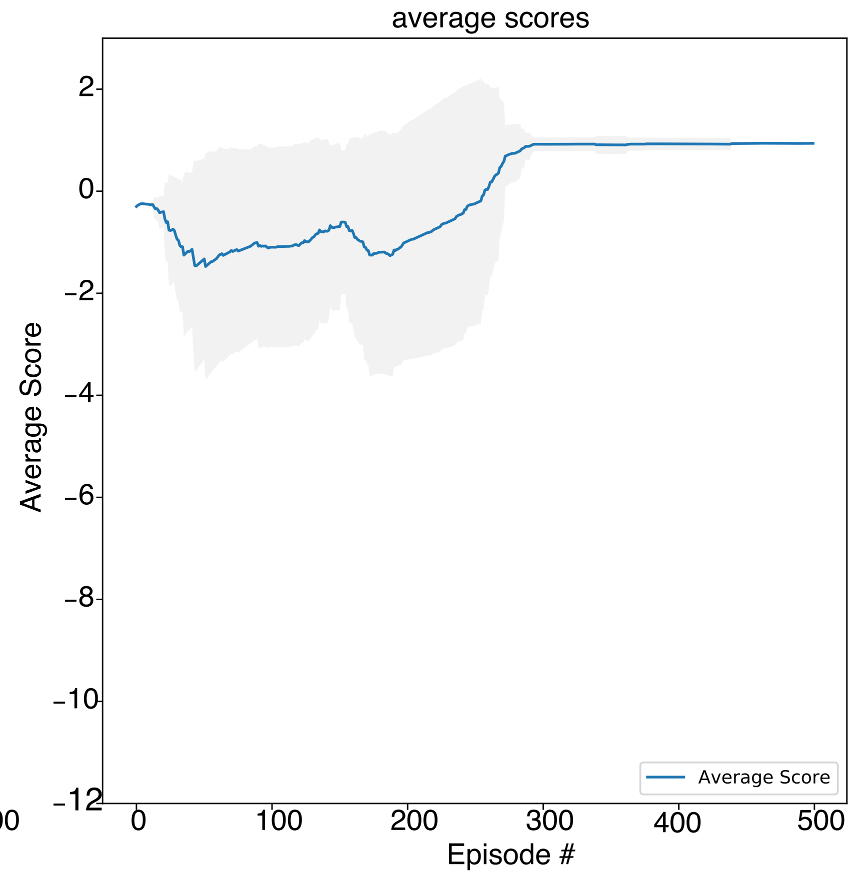
$$\phi_i = \pi \times b_i$$

Result : $|1\rangle \otimes |1\rangle \otimes |0\rangle \otimes |0\rangle$

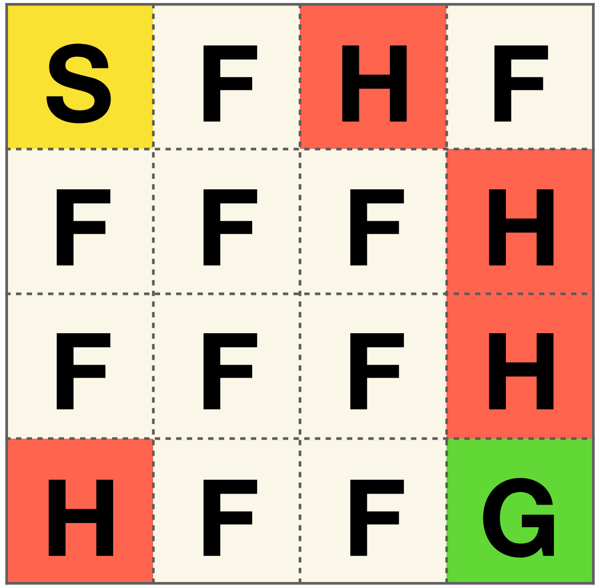


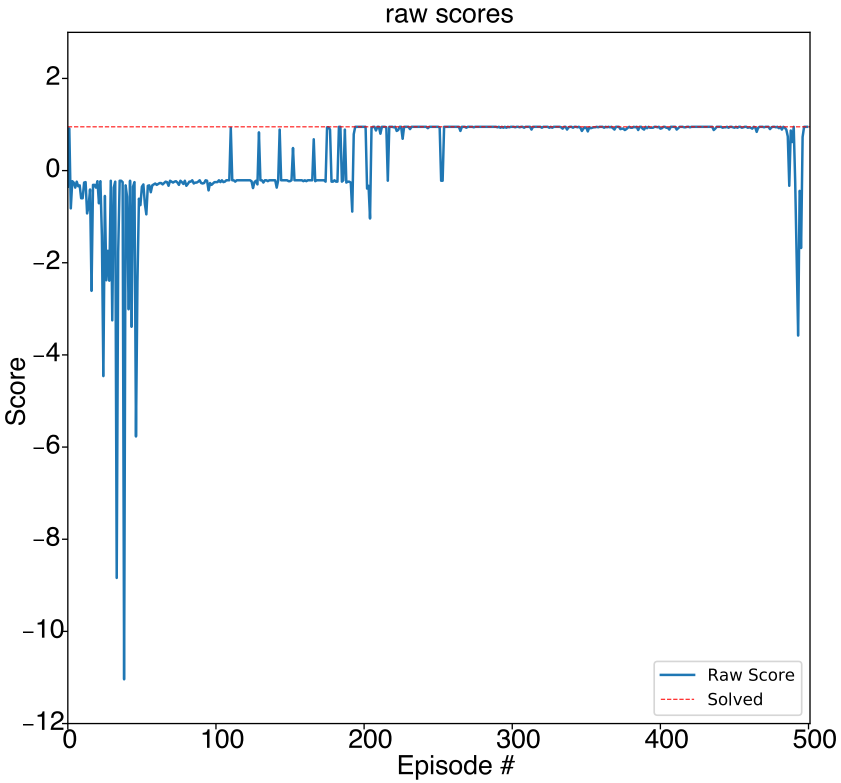
S	F	F	F
F	H	F	H
F	F	F	H
H	F	F	G

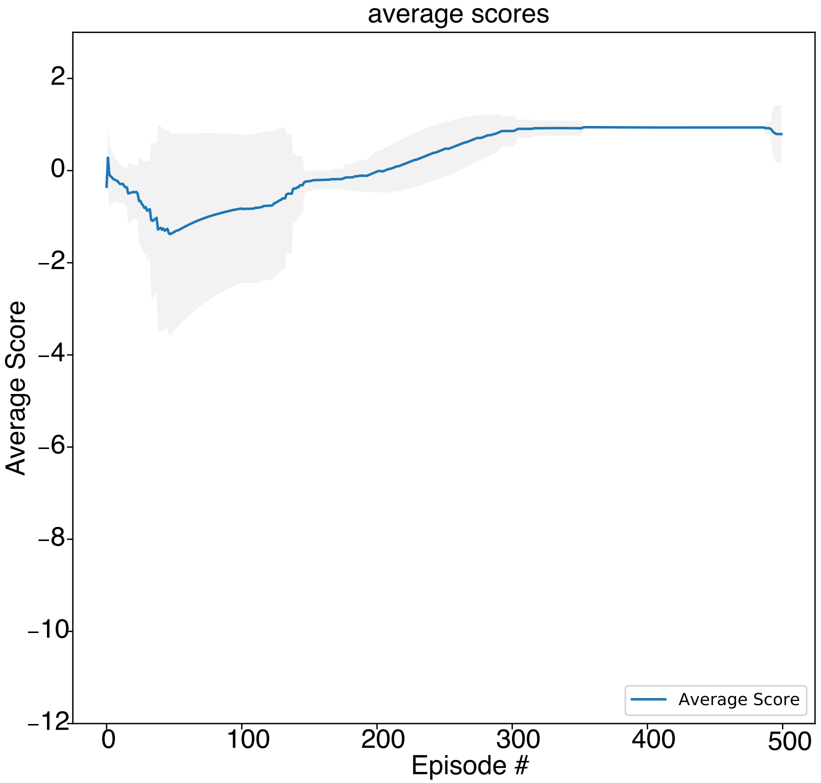




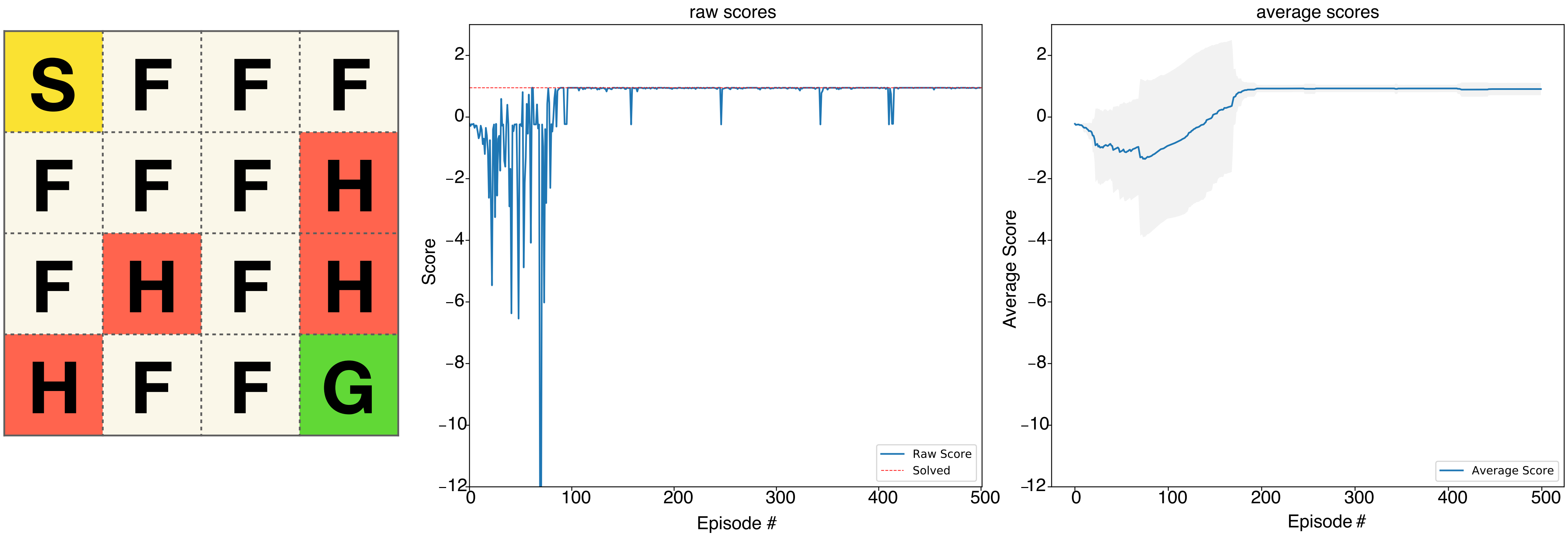
Chen, S. Y. C., Yang, C. H. H., Qi, J., Chen, P. Y., Ma, X., & Goan, H. S. (2020). Variational quantum circuits for deep reinforcement learning. *IEEE Access*, *8*, 141007-141024.





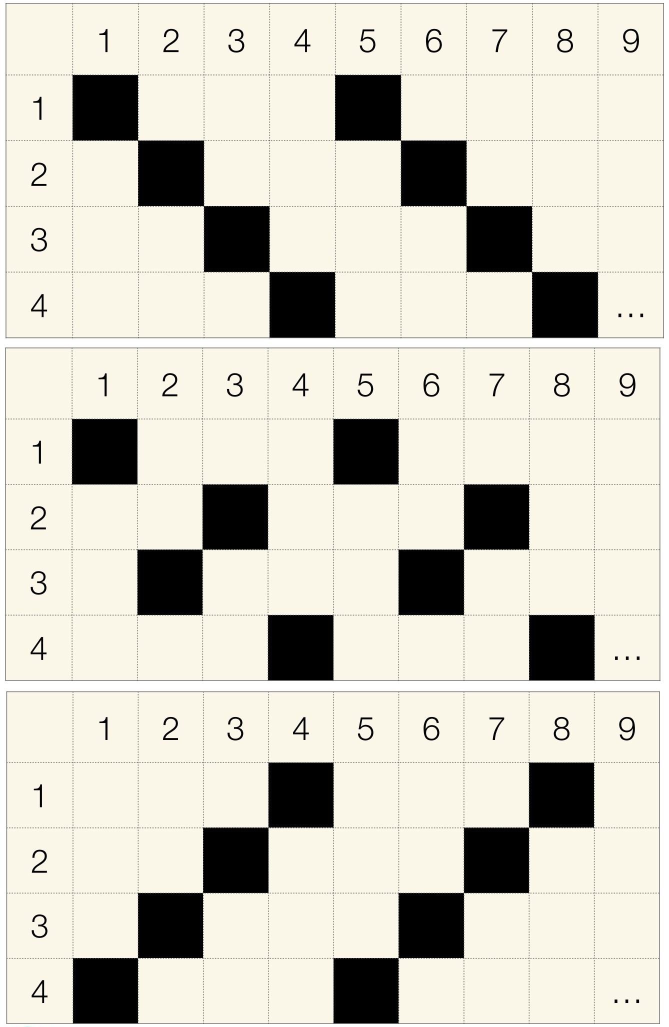


Chen, S. Y. C., Yang, C. H. H., Qi, J., Chen, P. Y., Ma, X., & Goan, H. S. (2020). Variational quantum circuits for deep reinforcement learning. *IEEE Access*, *8*, 141007-141024.



Chen, S. Y. C., Yang, C. H. H., Qi, J., Chen, P. Y., Ma, X., & Goan, H. S. (2020). Variational quantum circuits for deep reinforcement learning. *IEEE Access*, *8*, 141007-141024.

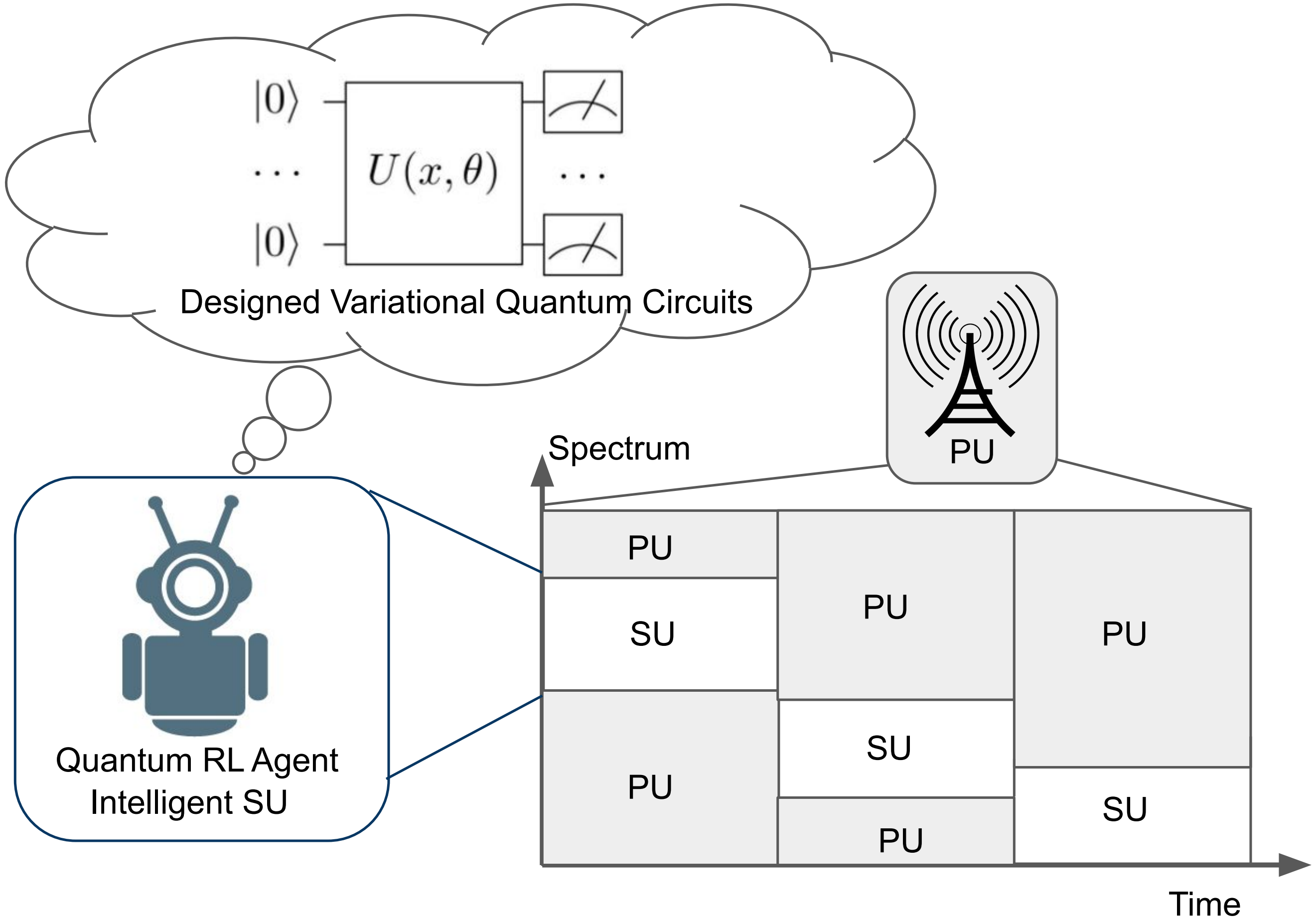
- Env: CognitiveRadio



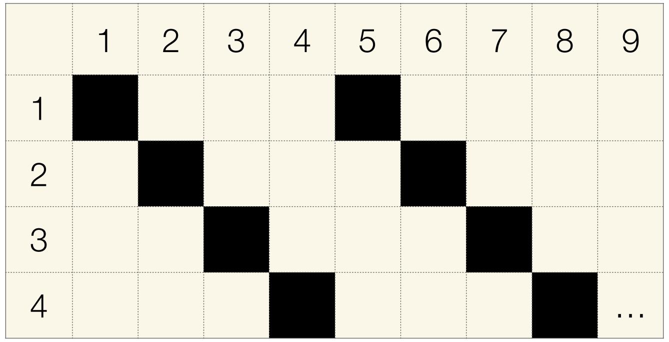
(a)

(b)

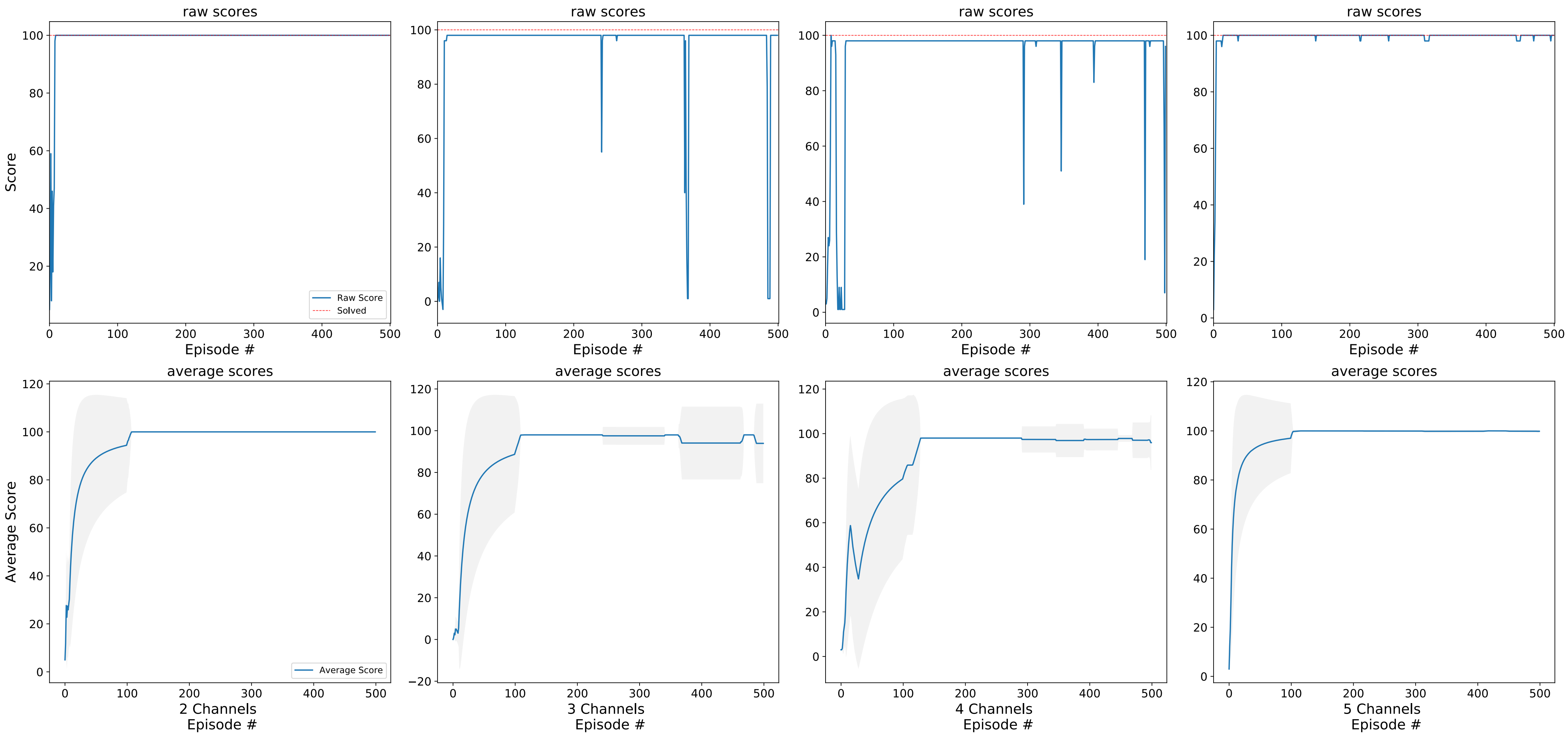
(c)

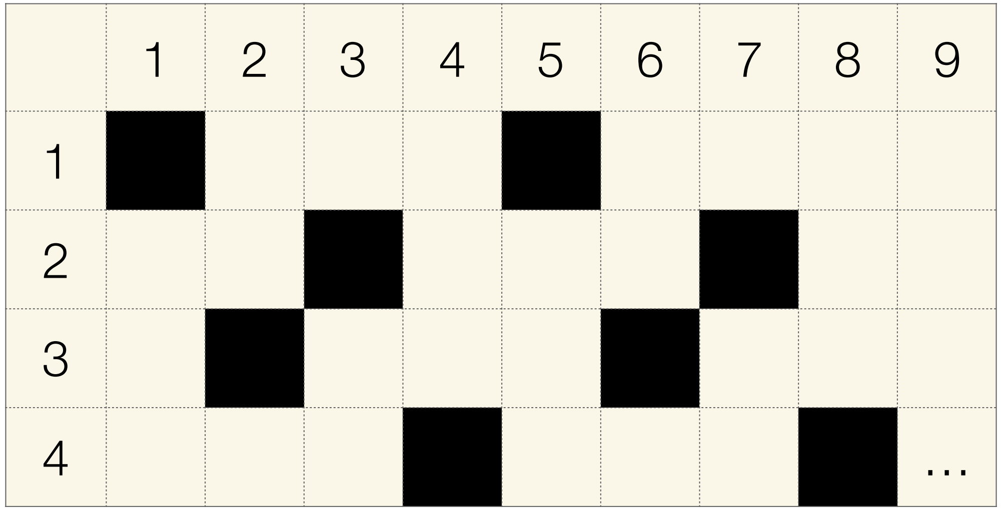


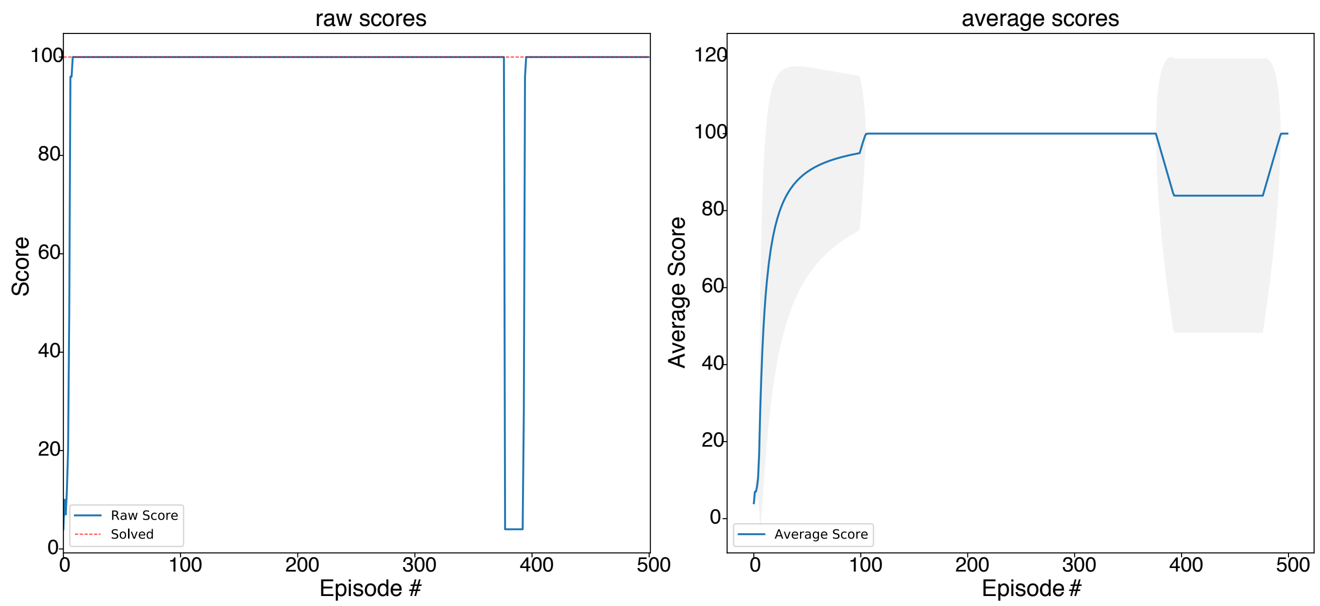
Chen, S. Y. C., Yang, C. H. H., Qi, J., Chen, P. Y., Ma, X., & Goan, H. S. (2020). Variational quantum circuits for deep reinforcement learning. *IEEE Access*, *8*, 141007-141024.


1
2
3
4
5
6
7
8
9
1
2
3
4
…

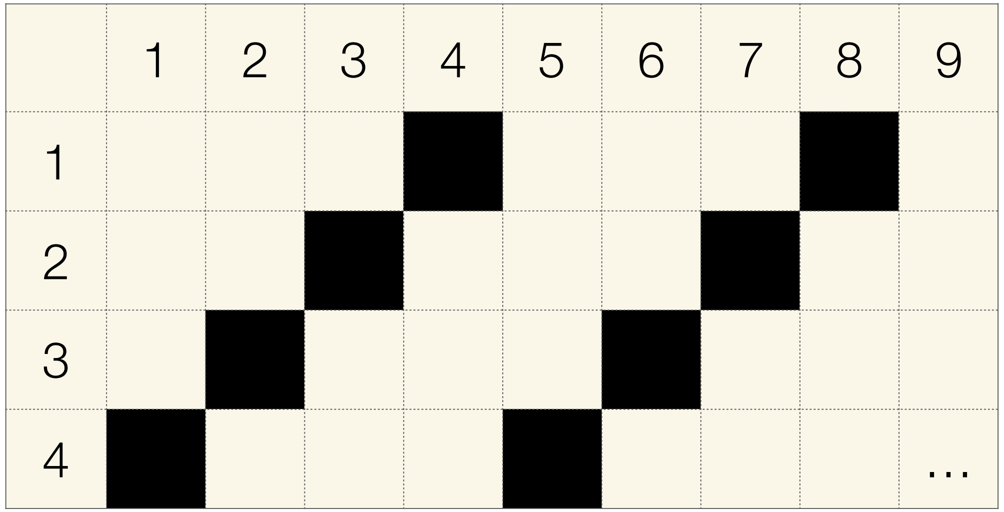
(a)

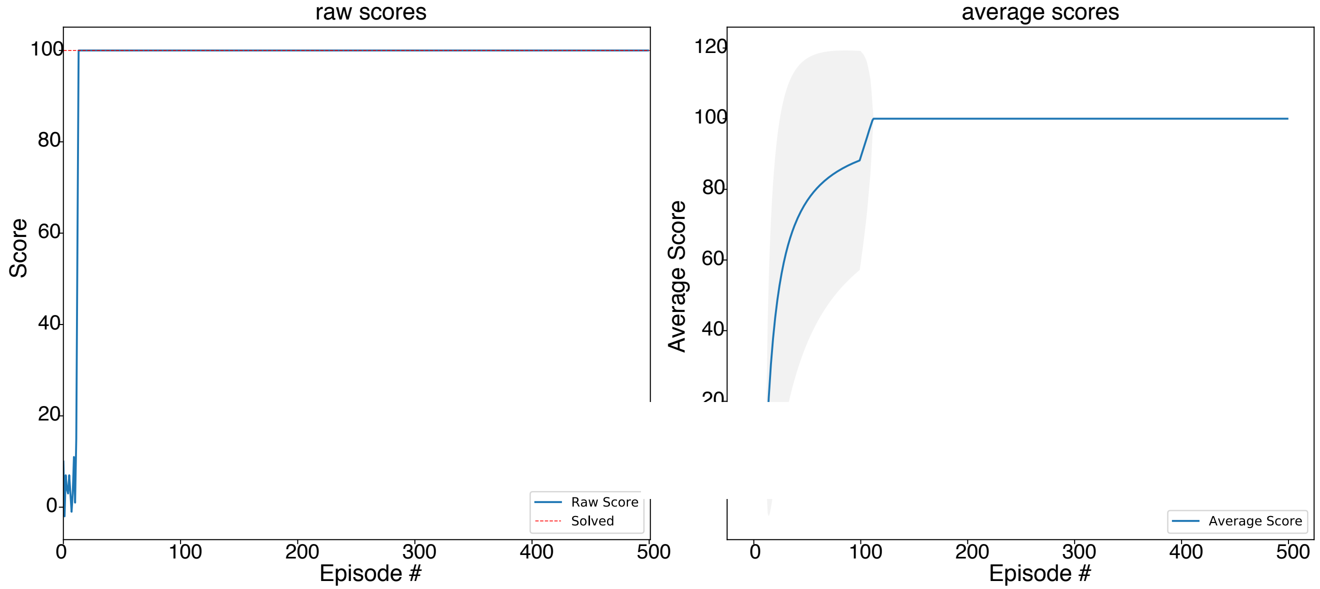

raw scores
Score
Episode #
Raw Score
Solved
average scores
Average Score
Average Score
2 Channels
3 Channels
4 Channels
5 Channels
Episode #





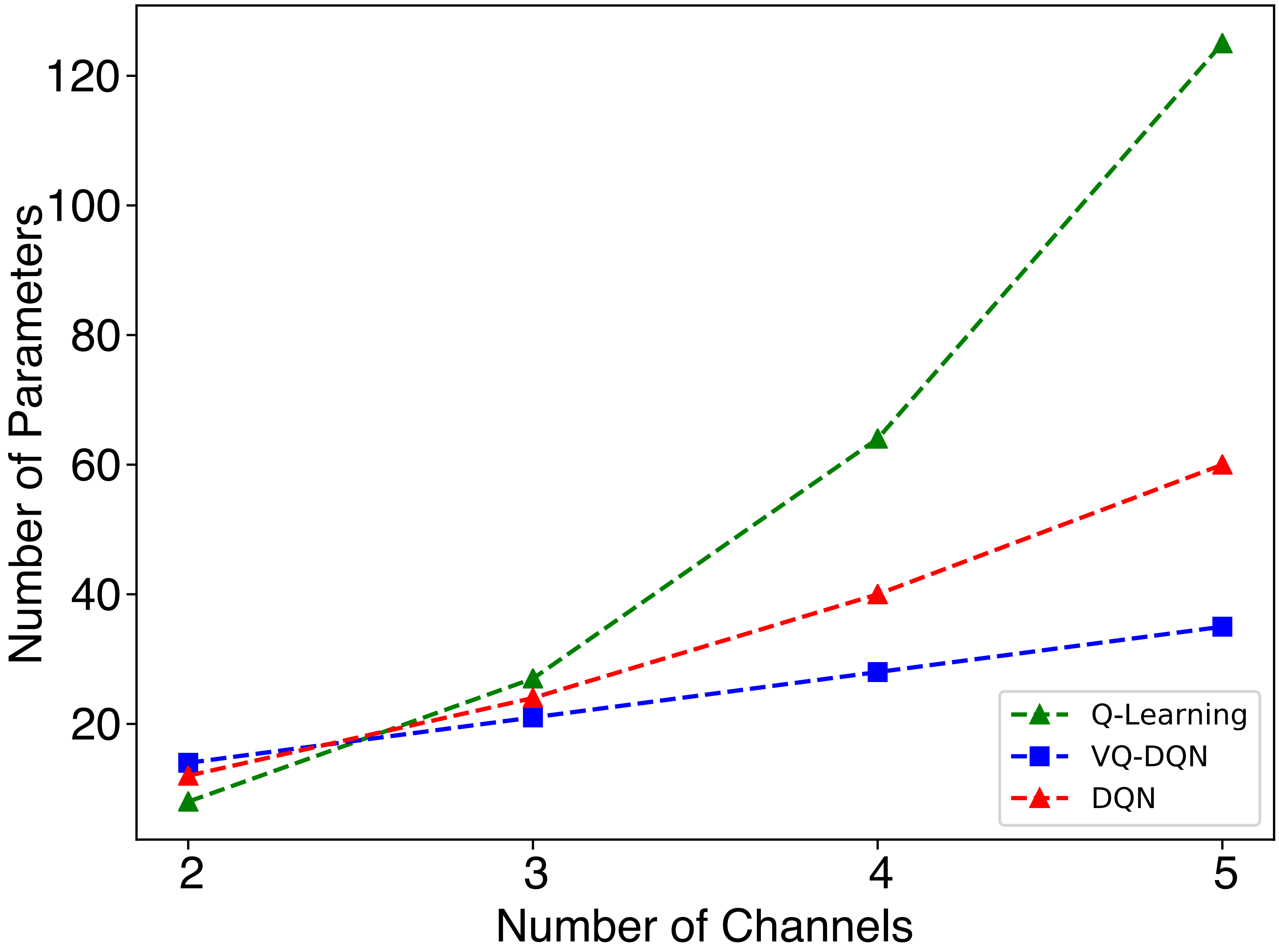
(b)





Brookhaven National Laboratory

(c)



Brookhaven National Laboratory

Evolutionary Quantum RL

- Why?
  - Gradient-based methods may suffer from local optima.
  - Certain QRL models are difficult to train via gradient-based methods.
  - In classical RL, evolutionary optimization can beat gradient-based methods in some hard tasks.

Chen, S. Y. C., Huang, C. M., Hsing, C. W., Goan, H. S., & Kao, Y. J. (2022). Variational quantum reinforcement learning via evolutionary optimization. *Machine Learning: Science and Technology*, *3*(1), 015025.

Evolutionary Optimization

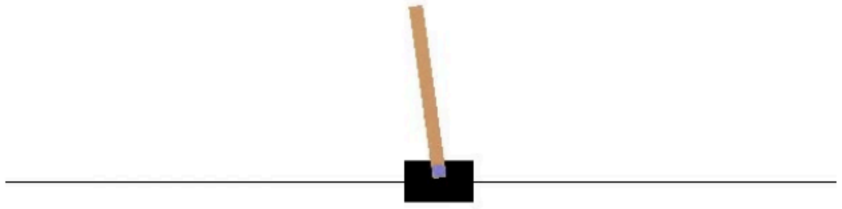
- **Initialization:**

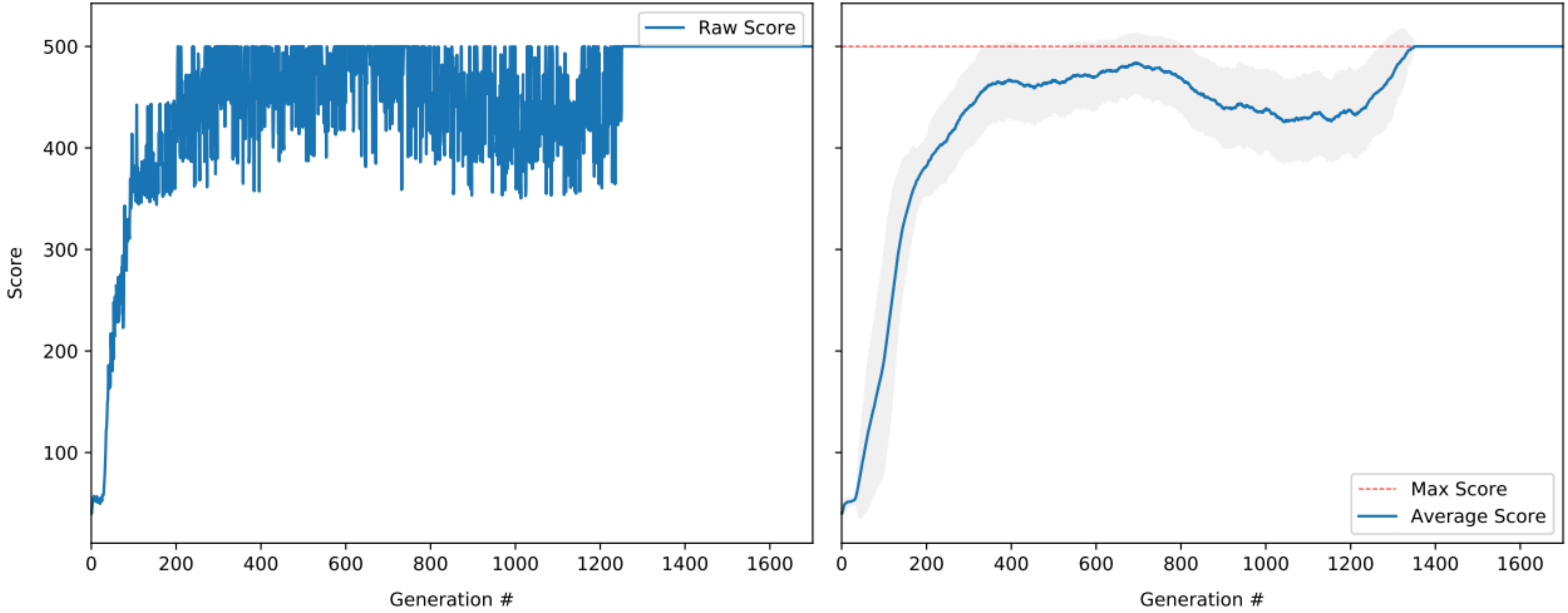
  Initialize the population $\mathscr{P}$ of N agents with each of them given randomly generated initial parameters θ, which are sampled from $\mathscr{N}(0, I)$
- **Running and evaluating the agents:**
  - Each agent plays the game R_1 times and get the average score $S_i^{avg} = \frac{1}{R_1} \sum_{r=1}^{R_1} S_{i,r}$
  - Top T agents age selected to be the *parents* to generate the next generation
- **Mutation and the next generation:**
  - $N - 1$ children: Each child is generated via a randomly selected agent from the parent group and slightly mutated according to $\theta \leftarrow \theta + \sigma\epsilon$ where σ is the mutation power and ϵ is the Gaussian noise
  - The *elite* or N^{th}- child is the best performing from the parent group

Brookhaven National Laboratory

Environments-CartPole

- **Observation:** A four dimensional vector s_t comprising values of the cart position, cart velocity, pole angle and pole velocity at the top.
- **Action:** There are two actions: pushing to the *right* or *left*.
- **Reward:** A reward +1 is given for every time step where the pole close to being upright.





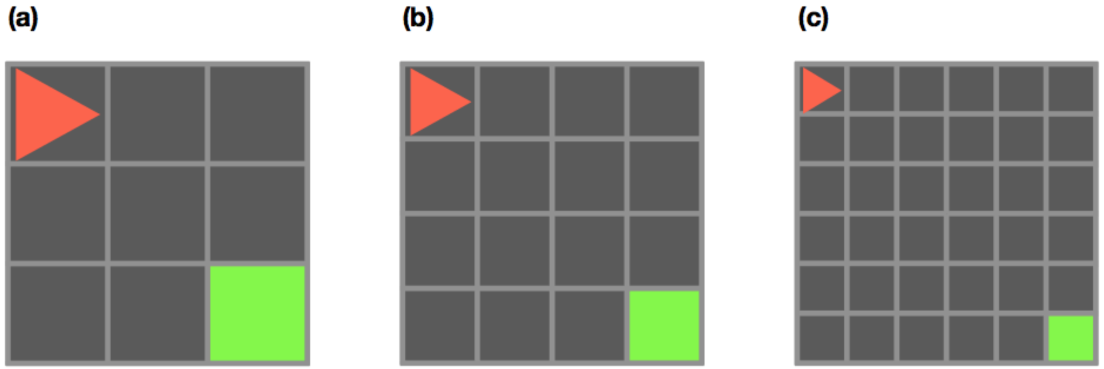
Brookhaven National Laboratory

Chen, S. Y. C., Huang, C. M., Hsing, C. W., Goan, H. S., & Kao, Y. J. (2022). Variational quantum reinforcement learning via evolutionary optimization. *Machine Learning: Science and Technology*, *3*(1), 015025.

Environments-MiniGrid

- **Observation:** A 147 dimensional vector s_t
- **Action:** There are 6 actions:
  - Turn left
  - Turn right
  - Move forward
  - Pick up an object
  - Drop the object
  - Toggle
- **Reward:** A reward of 1 is given when the agent reaches the goal. A penalty is subtracted from the reward according to:

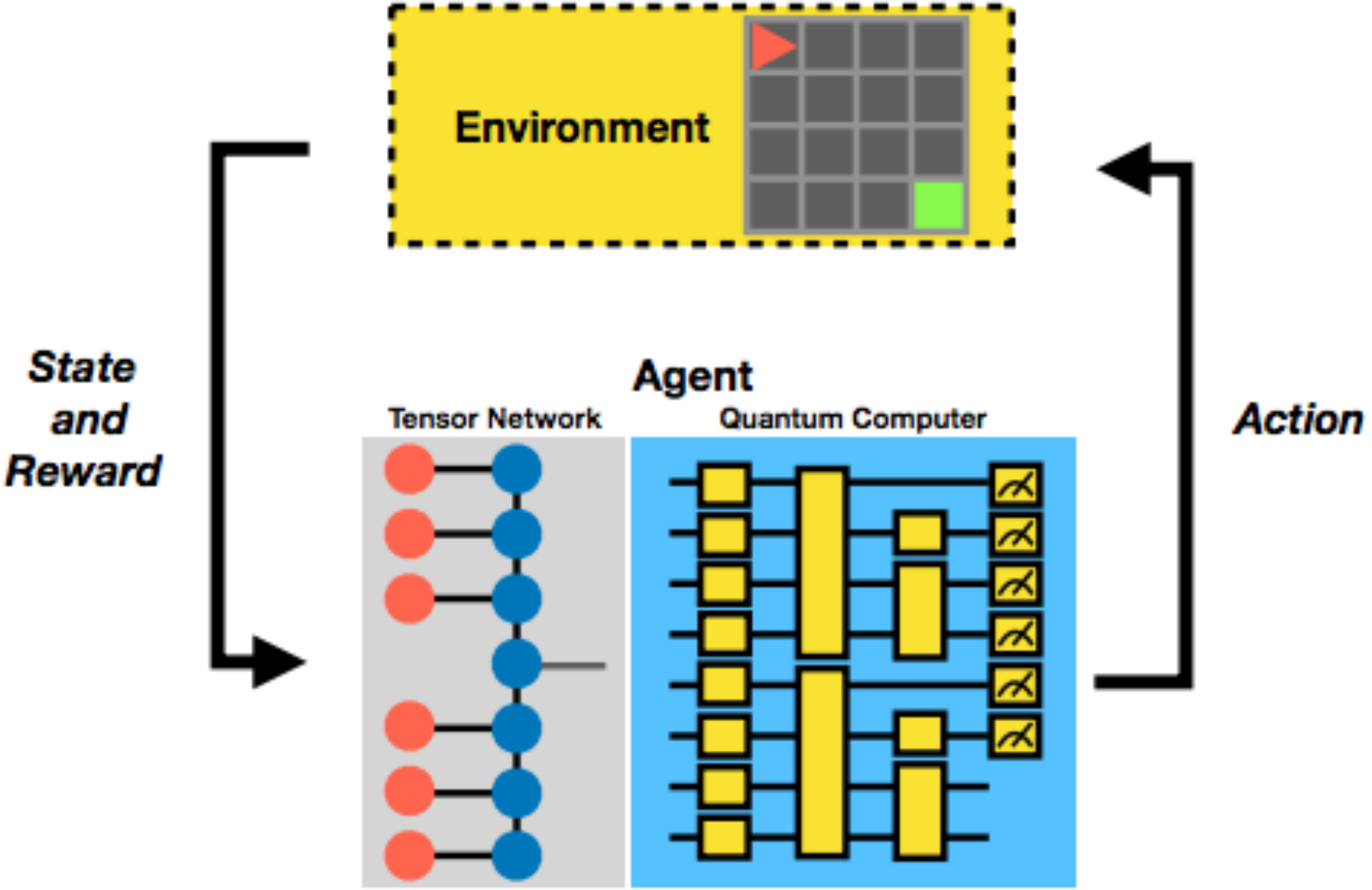
  1 − 0.9 × (*number of steps/max steps allowed*)



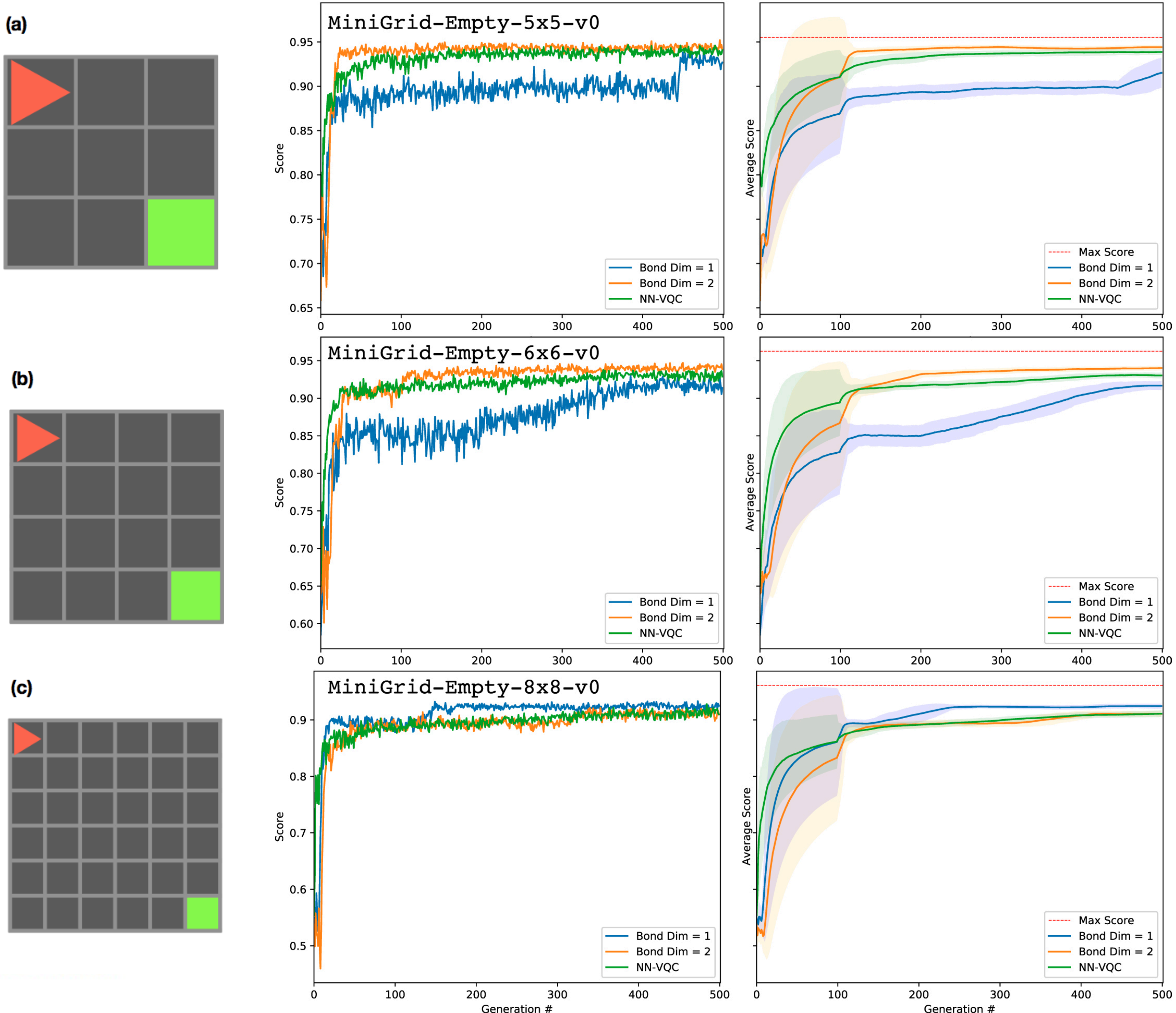
Brookhaven National Laboratory

Chen, S. Y. C., Huang, C. M., Hsing, C. W., Goan, H. S., & Kao, Y. J. (2022). Variational quantum reinforcement learning via evolutionary optimization. *Machine Learning: Science and Technology*, *3*(1), 015025.

Hybrid TN-VQC model



Chen, S. Y. C., Huang, C. M., Hsing, C. W., Goan, H. S., & Kao, Y. J. (2022). Variational quantum reinforcement learning via evolutionary optimization. *Machine Learning: Science and Technology*, *3*(1), 015025.



(a)
(b)
(c)
MiniGrid-Empty-5x5-v0
MiniGrid-Empty-6x6-v0
MiniGrid-Empty-8x8-v0
Score
Average Score
Generation #
Max Score
Bond Dim = 1
Bond Dim = 2
NN-VQC

I. Introduction
II. Quantum Computing (QC)
III. Quantum Machine Learning
IV. Applications-Classification
V. Applications-Sequential Learning
VI. Applications-Reinforcement Learning
VII. Conclusion and Outlook

- Quantum encoding / embedding methods are critical.
- Using VQC to replace NN in CNN can improve the performance when the number of parameters is similar.
- With careful design, quantum reinforcement learning can learn a similar task with fewer model parameter.
- QML models can be trained in federation to preserve data privacy.

Acknowledgements

- Brookhaven National Laboratory LDRD #20-024
- U.S. Department of Energy, Office of Science, DE-SC-0012704
- U.S. Air Force Office of Scientific Research, FA2386-20-1-4033
- Ministry of Science and Technology (MOST) of Taiwan:
    - 107-2112-M-002-016-MY3
    - 108-2112-M-002-020-MY3
    - 109-2112-M-002-023-MY3
    - 109-2627-M-002-003
    - 107-2627-E-002-001-MY3
    - 109- 2622-8-002-003
- National Taiwan University grant No. NTUCC-110L890102

Brookhaven National Laboratory