Efficient Parameter-Shift Rule Implementation for Computing Gradient on Quantum Simulators

Hai Vu Tuan; Le Vu Trung Duong; Hoai Luan Pham; Yasuhiko Nakashima

Nara Institute of Science and Technology, Japan

1. Parameterized quantum circuit (PQC) as learning model [1]

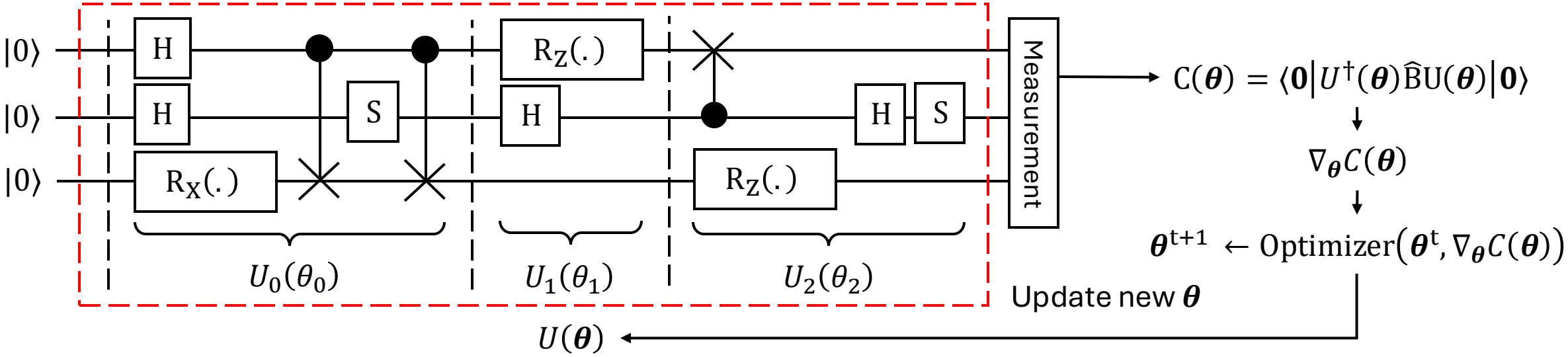


Fig. 1. Left) An example 3-qubits PQC is propose to split into sub-circuits. Each sub-circuit has only one parameter. (Right) The output is used for computing gradient, then back to update PQC's parameters by the classical optimizer.

A PQC $U(\boldsymbol{\theta}) = U_{m-1}U_{m-2} \dots U_0$ can be splitted into m sub-circuit (sub-operator) with $|\boldsymbol{\theta}| = m$.

[1] M. Schuld and F. Petruccione, Machine learning with quantum computers. Springer, 2021, vol. 676

2. Quantum gradient

Computing quantum gradient is the key in various quantum machine learning models:

$$\nabla_{\boldsymbol{\theta}} C(\boldsymbol{\theta}) = \left[\frac{\partial C}{\partial \theta_0} \frac{\partial C}{\partial \theta_1} \quad \cdots \quad \frac{\partial C}{\partial \theta_{m-1}}\right]^{\mathsf{T}}$$

$$\boldsymbol{\theta}_j^{\pm \epsilon_k} = \boldsymbol{\theta} \pm \epsilon_k \times e_j$$

e_j: j^{th} unit vector

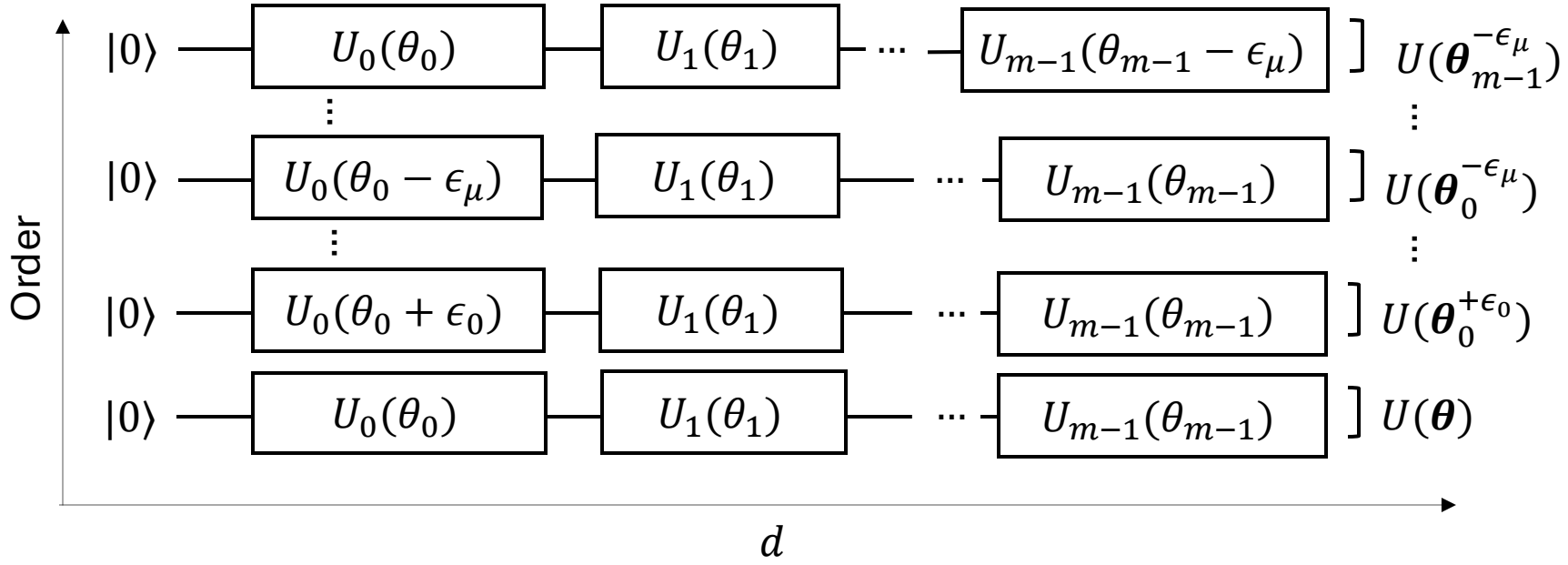
Where:

$$\frac{\partial C}{\partial \theta_j} = \sum_{k=1}^{R} d_k \left(C\left(\theta_j^{+\epsilon_k}\right) - C\left(\theta_j^{-\epsilon_k}\right)\right)$$

Based on general parameter-shift rule technique [2]

[2] D. Wierichs, J. Izaac, C. Wang, and C. Y.-Y. Lin, “General parameter-shift rules for quantum gradients,” Quantum, vol. 6, p. 677, Mar. 2022

3. Computing $\nabla_{\boldsymbol{\theta}} C(\boldsymbol{\theta})$ consume too much resources!



The total number of quantum evaluations is $2 \times R \times (m-1)$ for m parameters, then, number of $2^n \times 2^n$ matrix multiplication for simulate $\left\{U\left(\boldsymbol{\theta}_j^{\pm\epsilon_k}\right)\right\}$ is $\left(2 \times R \times (m-1)\right) \times m$ (n: number of qubits, m: number of parameters).

4. New scheme

For any pair $\left\{U\left(\boldsymbol{\theta}_j^{\pm\epsilon_k}\right)\right\}$, there are a duplication:

$$U\left(\boldsymbol{\theta}_j^{+\epsilon_k}\right) = \mathcal{U}_{j+1:m-1} U\left(\theta_j^{+\epsilon_k}\right) \mathcal{U}_{0:j-1}$$

$$U\left(\boldsymbol{\theta}_j^{-\epsilon_k}\right) = \mathcal{U}_{j+1:m-1} U\left(\theta_j^{-\epsilon_k}\right) \mathcal{U}_{0:j-1}$$

Furthermore:

$$U\left(\boldsymbol{\theta}_{j+1}^{\pm\epsilon_k}\right) = \mathcal{U}_{j+2:m-1} U\left(\theta_{j+1}^{\pm\epsilon_k}\right) \mathcal{U}_{0:j}$$

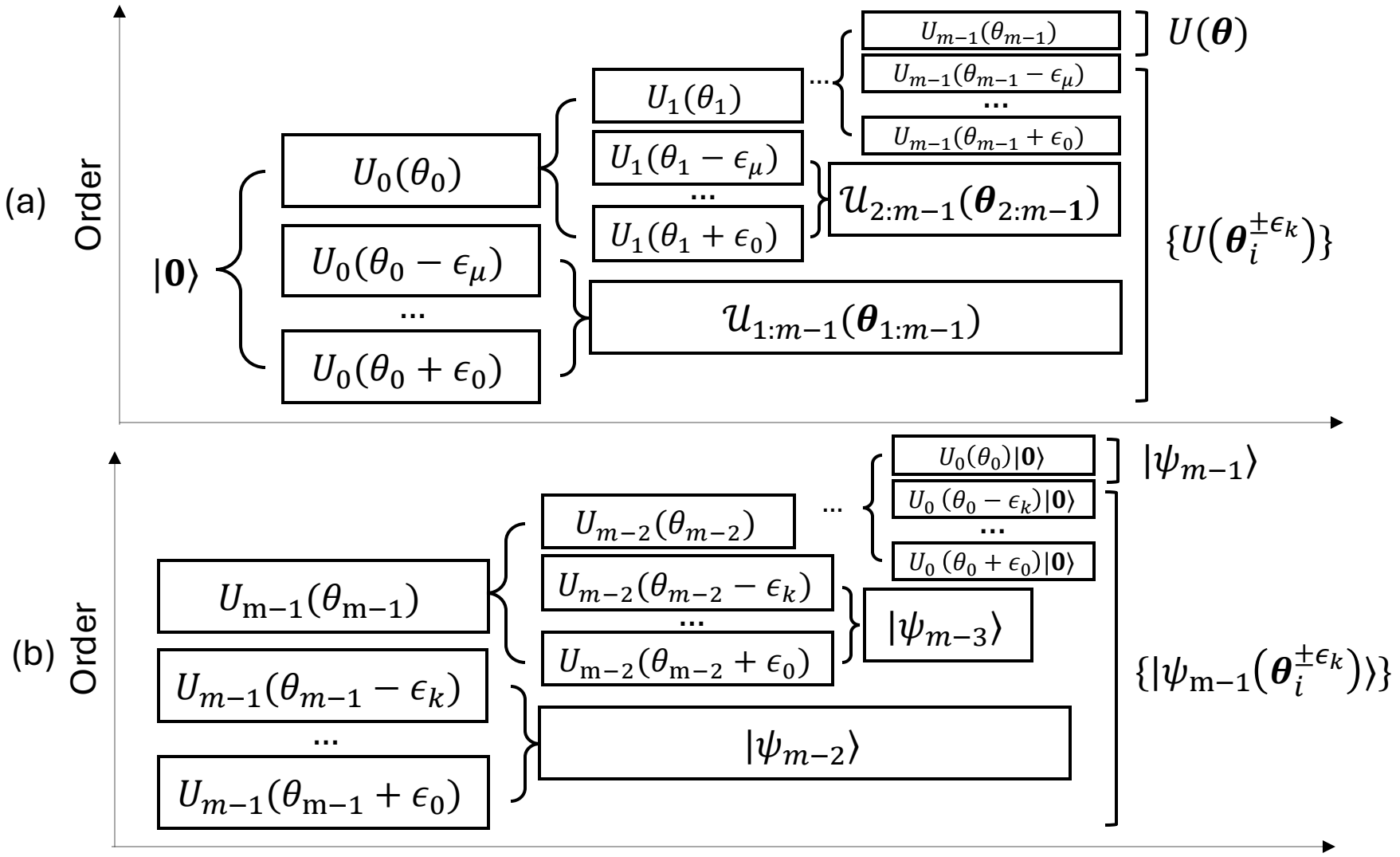
We already know:

$$\mathcal{U}_{j+1:m-1} = U_{j+1}\mathcal{U}_{j+2:m-1} \text{ and } \mathcal{U}_{0:j} = U_j \mathcal{U}_{0:j-1}$$

Notation:

$$\mathcal{U}_{i:j} = U_j U_{j-1} \dots U_i$$

4. Compact scheme



For matrix multiplication-based

For state vector-based

Fig. 2: Computation order from begin to end (y-axis) for (a) Baseline PSR, (b) Proposed PSR for MM-based simulator,

4. Using Look up table (LUT)

j \ i	0	1	2	3	...	$m-1$	$\|\psi_j\rangle$
0	U_0	$\mathbb{I}_{2^n}$	$\mathbb{I}_{2^n}$	$\mathbb{I}_{2^n}$	...	$\mathbb{I}_{2^n}$	$\|\psi_0\rangle$
1	$\mathcal{U}_{0:1}$	U_1	$\mathbb{I}_{2^n}$	$\mathbb{I}_{2^n}$	...	$\mathbb{I}_{2^n}$	$\|\psi_1\rangle$
2	$\mathcal{U}_{0:2}$	$\mathcal{U}_{1:2}$	U_2	$\mathbb{I}_{2^n}$	...	$\mathbb{I}_{2^n}$	$\|\psi_2\rangle$
3	$\mathcal{U}_{0:3}$	$\mathcal{U}_{1:3}$	$\mathcal{U}_{2:3}$	U_3	...	$\mathbb{I}_{2^n}$	$\|\psi_3\rangle$
...	...	...	...	...	...	...	...
$m-1$	U	$\mathcal{U}_{1:m-1}$	$\mathcal{U}_{2:m-1}$	$\mathcal{U}_{3:m-1}$	...	U_{m-1}	$\|\psi_{m-1}\rangle$

TABLE I: Look up table (LUT) for $\{U_{i:j}\}_{i,j} \in [0, m-1]$. Red and Blue elements are required for MM-based and state-based simulators, respectively. Violet elements are required for both. The normal (**Black**) elements do not need to be stored.

To fill this LUT, only take $\sum_{j=0}^{m-1} j = \frac{(m-1)\times m}{2}$ matrix multiplication

To achive $\left\{U\left(\boldsymbol{\theta}_j^{\pm\epsilon_k}\right)\right\}$, this method only take $\frac{(m-1)\times m}{2} + 4R < \left(2 \times R \times (m-1)\right) \times m$

5. Experiments and results

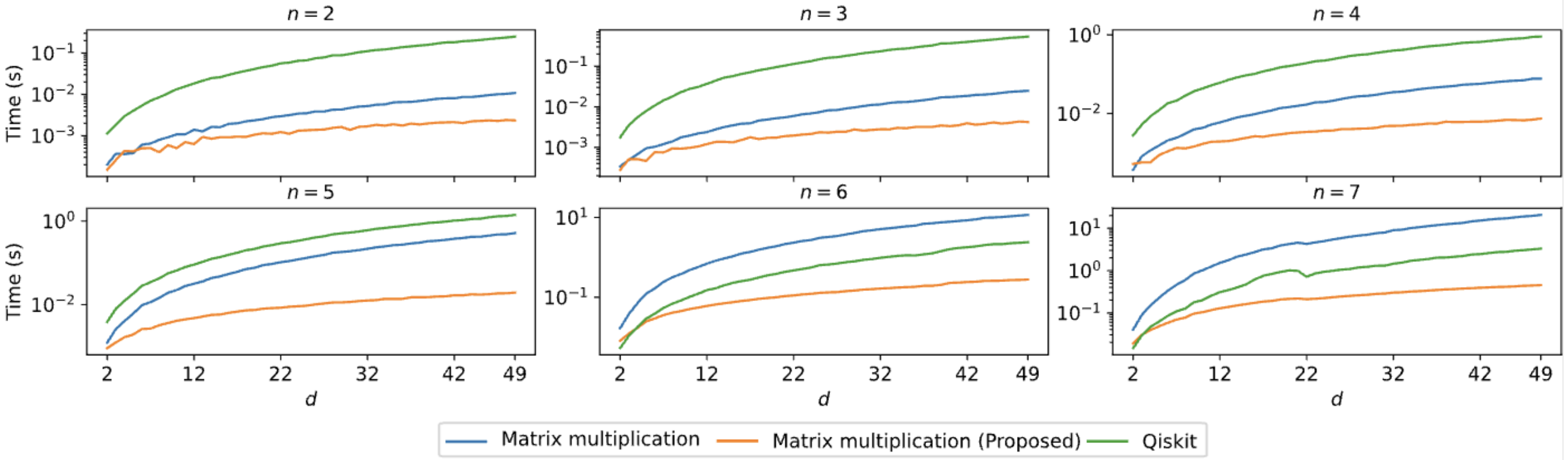


Fig. 3: Execution time (y-axis) from 2 to 7-qubit quantum circuit in log scale, with (x-axis) is depth value from 2 to 49.

5. Experiments and results

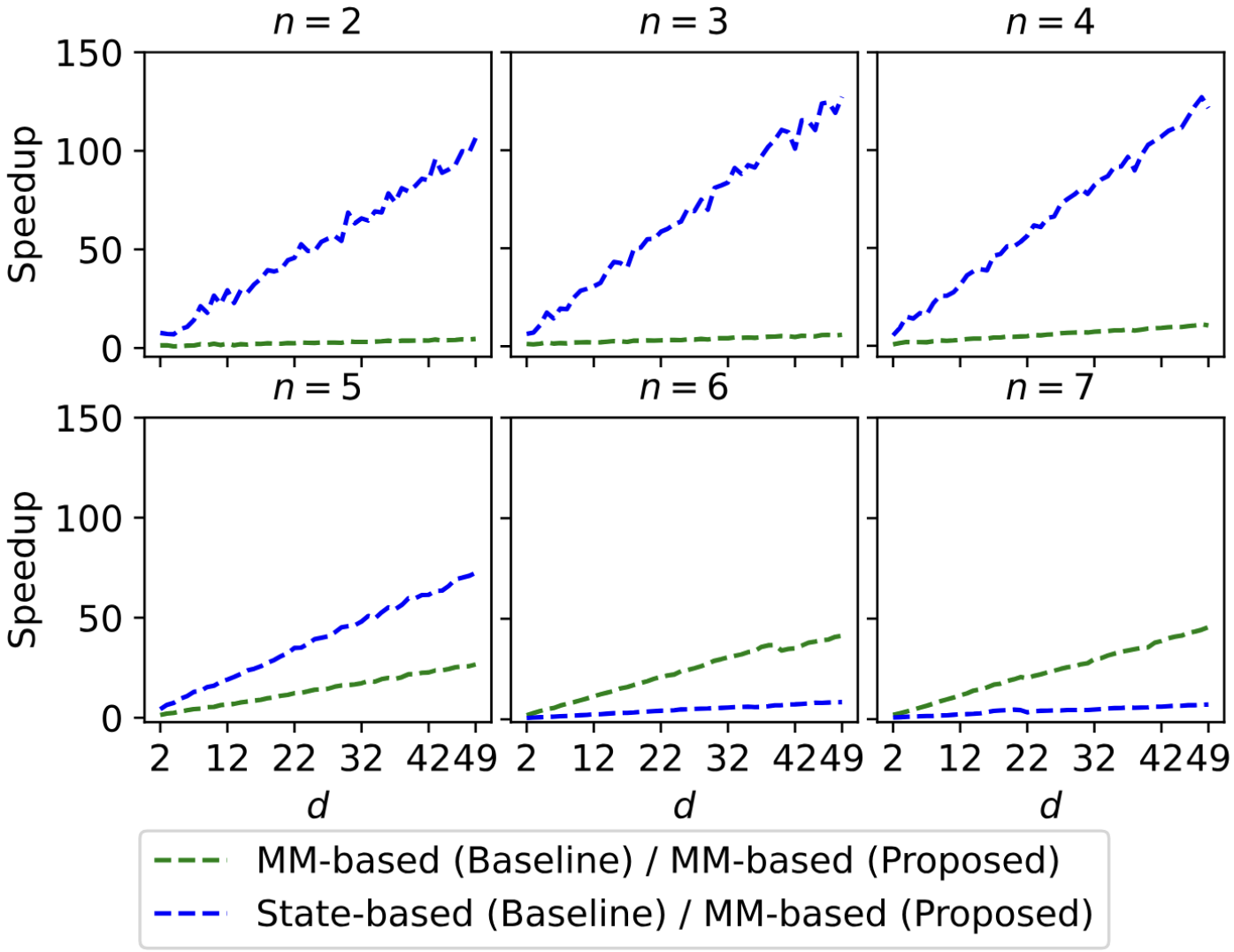


Fig. 4: Speedup (y-axis) compare between proposed method and baselines, from 2 to 7-qubit quantum circuit, with (x-axis) is depth value from 2 to 49.

References

[1] M. Schuld and F. Petruccione, Machine learning with quantum computers. Springer, 2021, vol. 676

[2] D. Wierichs, J. Izaac, C. Wang, and C. Y.-Y. Lin, “General parameter-shift rules for quantum gradients,” Quantum, vol. 6, p. 677, Mar. 2022.

[3] J. Stokes, J. Izaac, N. Killoran, and G. Carleo, “Quantum Natural Gradient,” Quantum, vol. 4, p. 269, May 2020.