

Quantum-PEFT: Ultra parameter-efficient fine-tuning

Toshiaki Koike-Akino^{(1),*}, Francesco Tonin^{(2),*}, Yongtao Wu⁽²⁾, Frank Zhengqing Wu⁽²⁾, Leyla Naz Candogan⁽²⁾, Volkan Cevher⁽²⁾

* Equal contribution

⁽¹⁾ Mitsubishi Electric Research Laboratories (MERL), 201 Broadway, Cambridge, MA 02139, USA

⁽²⁾ LIONS, École Polytechnique Fédérale de Lausanne (EPFL), 1015 Lausanne, Switzerland

Introduction

- Parameter-efficient fine-tuning (**PEFT**) is a cost-effective framework for downstream task to specialize pre-trained large foundation models.
- Low-rank adaptation (**LoRA**) variants achieve excellent performance.
- We propose a novel framework, named **Quantum-PEFT**, that leverages quantum unitary parameterizations, achieving orders-of-magnitudes higher compression rates over state-of-the-art PEFT methods.

Quantum-Inspired Machine Learning

- Quantum machine learning (**QML**) is an emerging framework leveraging quantum processing units (QPUs) for AI tasks.
- QML realizes ultra-efficient operations due to exponential expressivity.
- We introduce generalized QML framework based on alternating RY/CZ simplified two-design ansatz.

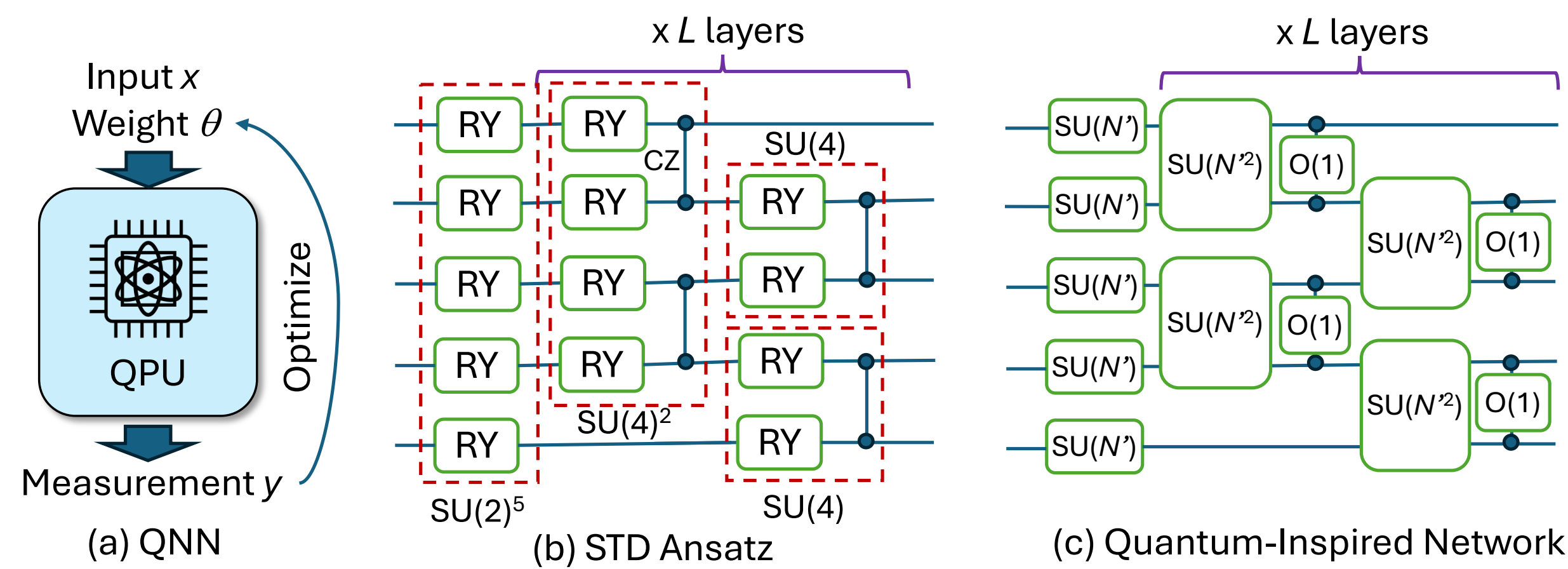
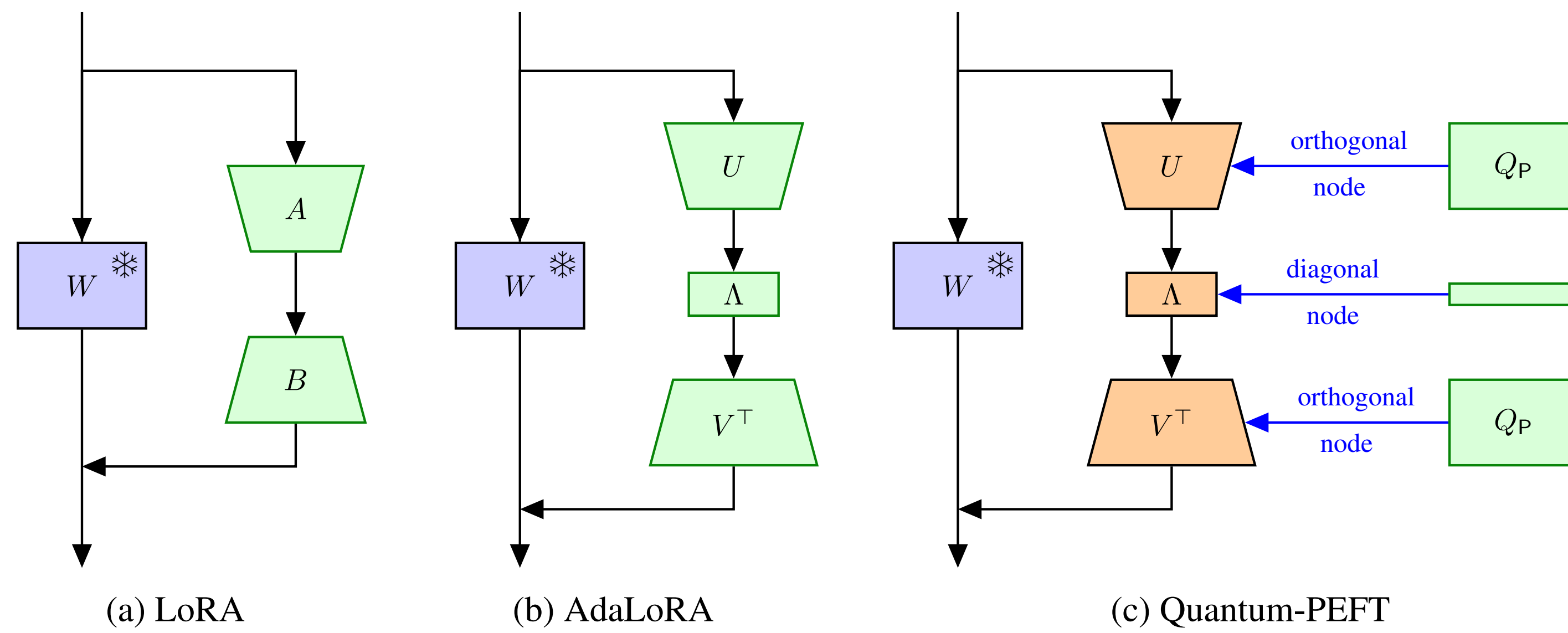


Figure 1: QML: (a) General pipeline for quantum neural network (QNN), embedding classical data x and variational parameters θ to control measurement y . (b) Simplified two-design ansatz. (c) Generalized quantum-inspired network.

Quantum-PEFT: Parameterization



- W is the frozen pretrained weight.
- Green boxes represent trainable parameters.
- LoRA updates W by training the low-rank matrices A, B .
- AdaLoRA introduces the SVD trainable form U, Λ, V with orthogonal regularizer.
- In Quantum-PEFT, the matrices U, V are not trainable parameters, but rather computed through mappings of *orders-of-magnitude smaller* intrinsic parameters.
- Contrary to AdaLoRA, U, V are left-orthogonal by construction.

Quantum-PEFT: Modules

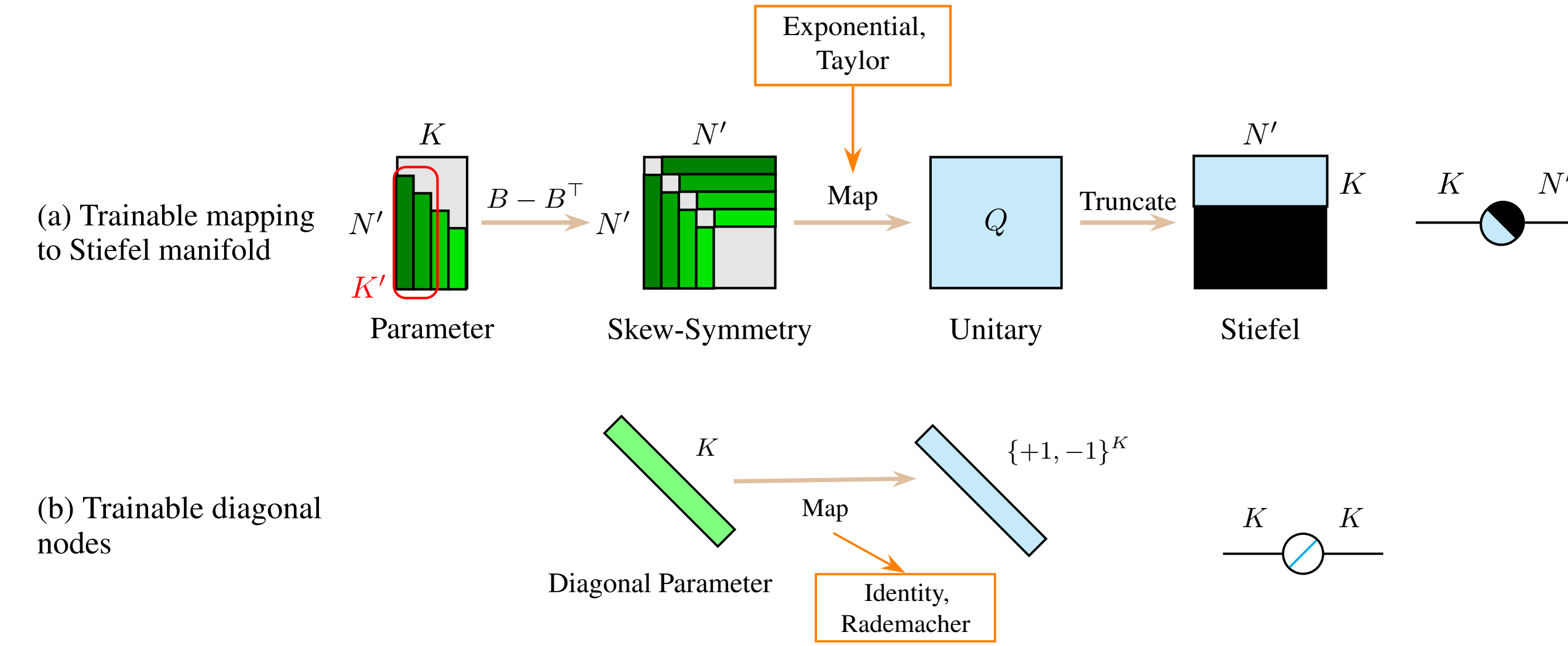
In Quantum-PEFT, we propose to use the Pauli parameterization based on the simplified two-design ansatz:

$$Q_P = \prod_{l=1}^L \left((I \otimes (CZ^{\otimes \frac{q-1}{2}} \bigotimes_{k=2}^q RY(\theta_{k,2l+1}))) ((CZ^{\otimes \frac{q-1}{2}} \bigotimes_{k=1}^{q-1} RY(\theta_{k,2l})) \otimes I) \right) \bigotimes_{k=1}^q (RY(\theta_{k,1})),$$

- L is the number of alternating entanglement layers.
- $q = \log_2(N)$ is the number of qubits.
- The trainable parameters are the rotation angles $\theta_{k,l}$ associated with the employed RY gates (i.e., rotations).

We show how to (i) generalize RY gates by generating unitary matrices from skew-symmetric matrices, and (ii) use recursive cosine-sine decomposition to allow non-power-of-two N .

- Mapping of skew-Hermitian to Stiefel manifold.



- (a) Trainable mapping onto the Stiefel manifold $\mathcal{V}_K(N')$. Intrinsic rank K' : top K' columns are trainable parameters in B .
- (b) Generalized CZ modules for diagonal nodes on either $O(1)^{N'}$ or $\mathbb{R}^{N'}$.
- Allowing arbitrary dimensions.
- We use quantum Shannon decomposition (QSD).
- QSD allows recursive decomposition.
- For example, when $N = 28$, we apply cos-sin decomposition twice to have $N = 2^4 + 2^3 + 2^2$, where the first $(N_1, N_2) = (2^4, 2^3 + 2^2)$ and the second CSD for the N_2 part will be further decomposed as $(N_1, N_2) = (2^3, 2^2)$.

Quantum-PEFT: Parameter efficiency

- Pauli parameterization enables logarithmically fewer number of trainable parameters.

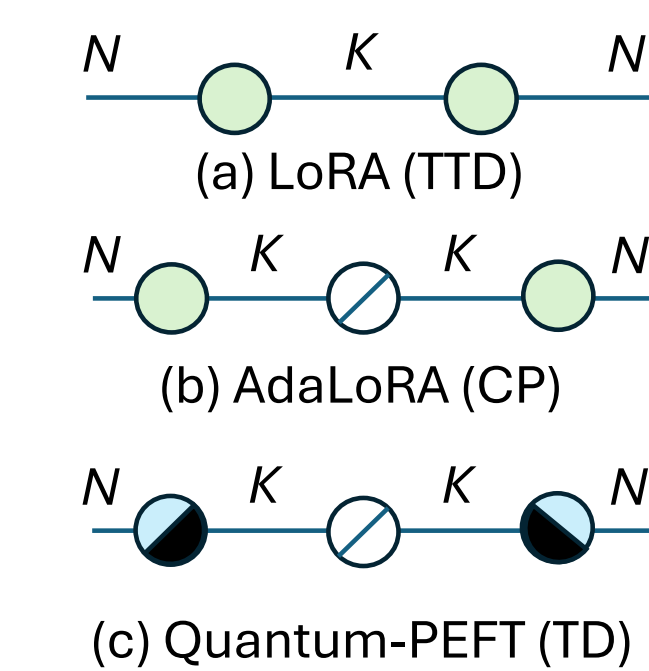


Figure 2: Tensor diagram of LoRA variants.

Table 1: Comparison of different PEFT methods and their computational requirements.

Method	# Trainable Parameters
LoRA (TTD)	$2NK$
AdaLoRA (CP)	$2NK + K$
Quantum-PEFT (TD: Q_P)	$2(2L + 1) \log_2(N) + K$

Experiments

- 3 transfer learning tasks:** LLM GLUE benchmark; E2E challenge; ImageNet to CIFAR10 classification.
- 4 foundation models:** DeBERTaV3; GPT2; Mistral-7B; ViT.
- Baseline methods include:** LoRA; AdaLoRA; LoHa; LoKr; MORA; QuanTA.

Results

- Quantum-PEFT shows competitive performance with extremely fewer number of trainable parameters, with favorable scaling to larger models.

Table 2: Results with DeBERTaV3 base on GLUE benchmark.

Method	# Trainable Parameters	SST-2	CoLA	RTE	MRPC	STS-B	Avg.	Memory
FT	184M	95.63	69.19	83.75	89.46	91.60	85.93	14200×
BitFit	0.1M	94.84	66.96	78.70	87.75	91.35	83.92	7.69×
HAdapter	0.61M	95.30	67.87	85.56	89.22	91.30	85.85	46.92×
PAdapter	0.60M	95.53	69.48	84.12	89.22	91.52	85.97	46.15×
HAdapter	0.31M	95.41	67.65	83.39	89.25	91.31	85.40	23.85×
PAdapter	0.30M	94.72	69.06	84.48	89.71	91.38	85.87	23.08×
LoRA	0.33M	94.95	68.71	85.56	89.71	91.68	86.12	25.38×
AdaLoRA	0.32M	95.80	70.04	87.36	90.44	91.63	87.05	24.62×
LoHa	0.33M	95.50	66.52	80.43	89.95	89.46	84.37	25.38×
LoKr	0.073M	95.07	69.46	85.20	89.71	90.76	86.04	5.62×
MORA	0.49M	95.79	67.13	85.19	89.08	90.13	85.46	37.87×
QuanTA	0.093M	95.30	67.75	84.48	89.22	91.01	85.55	7.15×
Quantum-PEFT	0.013M	95.85	67.85	<u>86.57</u>	90.78	91.06	<u>86.42</u>	1×

Table 3: Results with Mistral-7B on GLUE benchmark.

Method	# Trainable Parameters	SST-2	CoLA	RTE	MRPC	STS-B	Avg.
LoRA	3.54M	96.21	68.83	87.46	87.74	90.70	86.19
AdaLoRA	3.54M	96.90	70.38	87.53	89.52	90.96	87.06
Quantum-PEFT	0.758M	96.67	70.61	88.10	89.94	91.63	87.39

Table 4: Results on the E2E benchmark using GPT2 Medium.

Method	# Trainable Parameters	BLEU	NIST	METEOR	ROUGE-L	CIDEr
FT	354.92M	68.2	8.62	46.2	71.0	2.47
AdaLoRA	0.38M	64.64	8.38	43.49	65.90	2.18
LoRA	0.39M	66.88	8.55	45.48	68.40	2.31
LoHa	0.39M	65.03	8.45	43.76	66.54	2.22
LoKr	0.098M	63.90	8.27	42.35	65.22	2.04
Quantum-PEFT	0.098M	67.46	8.58	<u>45.02</u>	<u>67.36</u>	2.31

- Quantization and mixed-precision Quantum-PEFT keep good performance over full-precision PEFT.

Table 5: Results for ViT transfer learning from ImageNet-21k to CIFAR10. Base ViT is quantized with 3 bits.

Method	Original	FT	LoRA _{K=1}	LoRA _{K=2}	LoRA _{K=4}	Quantum-PEFT
# Parameters	—	85.81M	0.037M	0.074M	0.147M	0.007M
Accuracy	76.21%	98.05%	98.14%	98.30%	98.39%	98.46%

Table 6: Quantization impact on ViT Lie parameters for transfer learning ImageNet-21k to CIFAR10.

Quantization	FP32	INT8	INT4	INT3	INT2	INT1
# Bits per parameter	32	8.25	4.25	3.25	2.25	1.25
Accuracy (Uniform Bit Loading)	98.81%	98.79%	98.78%	98.75%	98.67%	97.96%
Accuracy (Adaptive Bit Loading)	98.81%	98.78%	98.87%	98.80%	98.77%	98.64%