Quantum-PEFT: Ultra parameter-efficient fine-tuning

Toshiaki Koike-Akino [*] [1] [2] **Francesco Tonin** [*] [2] **Yongtao Wu** [2] **Leyla Naz Candogan** [2] **Volkan Cevher** [2]

Abstract

This paper introduces Quantum-PEFT that leverages quantum computations for parameter-efficient fine-tuning (PEFT). Unlike other additive PEFT methods, such as low-rank adaptation (LoRA), Quantum-PEFT exploits an underlying full-rank yet surprisingly parameter-efficient *quantum unitary parameterization* with alternating entanglement. With the use of Pauli parameterization, the number of trainable parameters grows only logarithmically with the ambient dimension, as opposed to linearly as in LoRA-based PEFT methods. Consequently, Quantum-PEFT achieves vanishingly smaller number of trainable parameters than the lowest-rank LoRA as dimensions grow, enhancing parameter efficiency while maintaining a competitive performance. We apply Quantum-PEFT to several transfer learning benchmarks in language and vision, demonstrating significant advantages in parameter efficiency.

1. Introduction

Fine-tuning large pre-trained models is a cost-effective method to adapt a general-purpose model to additional domains and tasks in computer vision and natural language processing (Devlin et al., 2018; Liu et al., 2019; He et al., 2020; Radford et al., 2019; Brown et al., 2020; AI@Meta, 2024). Yet, even the practice of fine-tuning for each application can be costly as models scale to billions or trillions of parameters. The substantial memory requirements, such as GPT-3's 350GB footprint (Brown et al., 2020), can pose significant resource challenges, restricting practical deployment.

Parameter-efficient fine-tuning (PEFT) addresses the resource challenges of task specialization for massive pretrained networks without the need to fine-tune parameters in full model weights dimensions (Aghajanyan et al., 2020; Hu et al., 2021; Edalati et al., 2022). For instance, low-rank adaptation (LoRA) (Hu et al., 2021) uses low-rank decompositions to modify weights, whereby reducing the number of trainable parameters. Despite its efficiency, there are limitations to the number of parameters, which include a compression ratio constrained by rank-1 decompositions and a linear scaling of trainable parameters with weight matrix dimensions.

We introduce Quantum-PEFT, a novel framework that achieves extremely parameter-efficient fine-tuning beyond LoRA-variants by leveraging quantum unitary parameterizations (Biamonte et al., 2017; Schuld et al., 2015). The core idea is to reparameterize the layers of pre-trained networks as generalized quantum circuits capturing complex transformations, which only require a logarithmic number of trainable parameters. The ultra parameter efficiency is enabled by parameterizing the low-rank subspaces via Kronecker products of generalized Pauli rotations. The key contributions of our work include:

- We introduce new quantum-inspired modules based on generalized Pauli parametrization and quantum tensor network. We propose a novel framework, named Quantum-PEFT, that leverages quantum unitary parameterizations for extremely parameter-efficient fine-tuning, achieving orders-of-magnitudes higher compression rates over state-of-the-art PEFT methods.
- Quantum-PEFT with Pauli parameterization enables logarithmic scaling of trainable parameters with respect to the ambient dimension of the model, realizing even smaller parameters than the lowest-rank LoRA.
- Through extensive experiments on language and vision tasks, we show Quantum-PEFT's significant advantage in parameter efficiency, achieving 5 to 25-fold reduction in trainable parameters compared to LoRA, yet maintaining competitive performance.

2. Quantum-PEFT method

Notations: Let $\mathrm{SU}(N)$, $\mathfrak{su}_N$, $\mathrm{SO}(N)$, $\mathrm{O}(N)$, and $\mathcal{V}_K(N)$ denote the special unitary Lie group of size N, its Lie algebra, special orthogonal group, orthogonal group, and real-valued Stiefel manifold having orthonormal K frames, respectively. We denote I, $\mathbb{R}$, $\otimes$, $[\cdot]^\top$, and $\jmath$ as identity matrix

[*] Equal contribution [1] Mitsubishi Electric Research Laboratories (MERL), 201 Cambrdige, MA 02139, USA [2] LIONS, École Polytechnique Fédérale de Lausanne (EPFL), 1015 Lausanne, Switzerland. Correspondence to: Toshiaki Koike-Akino <koike@merl.com>.

ICML 2024 Workshop on Efficient Systems for Foundation Models, Vienna, Austria. Copyright 2024 by the author(s).

of proper size, real numbers field, Kronecker product, transpose, and imaginary number, respectively.

2.1. Quantum machine learning (QML)

Typical QML uses quantum computers as a neural network module where classical data and weight values are embedded into quantum variational parameters such as Pauli rotation angles to control measurement outcomes, as shown in Fig. 1(a). Any quantum circuits can be decomposed (Kitaev, 1997) into a series of single-qubit rotations and two-qubit entanglements. Pauli operators play an important role to generate any unitary rotations up to a global phase. The group $\mathrm{SU}(N)$—the Lie group of unitary $N \times N$ matrices having determinant 1—can be generated by the Lie algebra $\mathfrak{su}_N$, i.e., the set of $N \times N$ skew-Hermitian matrices. For single-qubit rotations over $\mathrm{SU}(2)$, the Lie algebra is a span of $\{\jmath X, \jmath Y, \jmath Z\}$, with Pauli matrices: $X = \left[\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right], Y = \left[\begin{smallmatrix} 0 & -\jmath \\ \jmath & 0 \end{smallmatrix}\right], Z = \left[\begin{smallmatrix} 1 & 0 \\ 0 & -1 \end{smallmatrix}\right]$. The exponential mapping of its linear combinations generates $\mathrm{SU}(2)$. For example, quantum RY rotation gate is given as

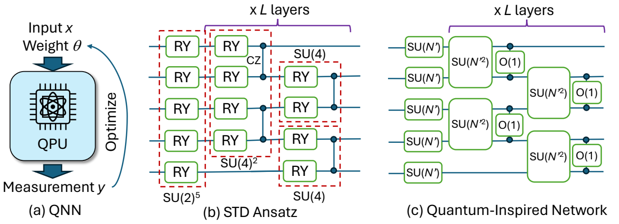
$$\begin{aligned} \mathsf{RY}(\theta) &= \exp(-\jmath \tfrac{\theta}{2} Y) = \exp\left(\begin{bmatrix} 0 & -\theta/2 \\ \theta/2 & 0 \end{bmatrix} \right) \\ &= \begin{bmatrix} \cos(\theta/2) & -\sin(\theta/2) \\ \sin(\theta/2) & \cos(\theta/2) \end{bmatrix}, \end{aligned} \tag{1}$$

which alone spans the special orthogonal group $\mathrm{SO}(2)$ and forms $\mathrm{O}(2)$ along with a reflection Z.

Two-design ansatz (Cerezo et al., 2021) used for QML uses a small number of parameters in order of $\mathcal{O}[\log_2(N)]$ to represent unitary matrices $\mathrm{SU}(N)$ whose statistical properties are identical to ensemble random unitaries with respect to the Haar measure up to the first 2 moments. This property suggests that gradient optimization can uniformly adjust few-parameter Pauli rotation angles along the unitary group $\mathrm{SU}(N)$. Comparing to the full degree of freedoms of $\dim[\mathrm{SU}(N)] = N^2 - 1$ for any skew-Hermitian matrices, the QML has a great potential to realize parameter-efficient representation in its logarithmic order. With q-qubit quantum processing unit (QPU), it can manipulate exponentially large dimensional state space of size $N = 2^q$ simultaneously through Pauli unitary rotations. In the following, we introduce a generalized framework to extend the QML features for extremely parameter-efficient neural network modules, which constitute the building blocks of Quantum-PEFT.

2.2. Quantum-PEFT: Pauli, generalized RY and CZ

Pauli parameterization As shown in Fig. 1(b), the simplified two-design (STD) ansatz (Cerezo et al., 2021) uses an alternating circuit composed of RY and controlled-Z (CZ) entangling gates: $\mathsf{CZ} = \mathrm{diag}[1, 1, 1, -1]$, which is an element of reflection groups $\mathrm{O}(1)^4 = \{\pm 1\}^4$. This ansatz is suited for neural networks as they are real-valued quantum operations over $\mathrm{SO}(N)$, i.e., not complex-valued operations over $\mathrm{SU}(N)$ arising when using RZ or RX rotations. In Quantum-PEFT, we propose to use the Pauli parameterization based on this STD ansatz, as given below:



(a) QNN (b) STD Ansatz (c) Quantum-Inspired Network

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.

$$Q_{\mathsf{P}} = \prod_{l=1}^{L} \Big(\big(I \otimes \mathsf{CZ}^{\otimes \frac{q-1}{2}}\big) \bigotimes_{k=2}^{q} \big(\mathsf{RY}(\theta_{k,2l+1})\big) \\ \big(\mathsf{CZ}^{\otimes \frac{q-1}{2}} \otimes I\big) \bigotimes_{k=1}^{q-1} \big(\mathsf{RY}(\theta_{k,2l})\big) \Big) \bigotimes_{k=1}^{q} \big(\mathsf{RY}(\theta_{k,1})\big), \tag{2}$$

where L is the number of alternating entanglement layers, and $q = \log_2(N)$ is the number of qubits (above equation assumes odd number). This Pauli parameterization has $(2L + 1)\log_2(N) - 2L$ parameters, increasing only logarithmically with the matrix size N. While the tensor rank is 2, the effective rank of the matrix Q_{P} is full of N thanks to the alternating CZ entanglement. Not only parameter efficient, but Pauli parametrization is also computationally efficient as it takes $\mathcal{O}[N \log_2(N) L]$ operations compared to quadratic complexity for unitary matrix rotations. When running on a QPU, it would require only $\mathcal{O}[\log_2(N) L]$ operations. Solovay–Kitaev theorem (Kitaev, 1997) may suggest that the required depth size L to achieve any unitary rotations scales only in a polylog order. Motivated by the STD ansatz, we can further generalize the parameterization from $\mathrm{SU}(2)$ to $\mathrm{SU}(N')$ with an arbitrary size of $N' > 2$ as shown in Fig. 1(c) as a building block to represent a large unitary matrix $\mathrm{SU}(N)$ with a smaller number of unitary factors $\mathrm{SU}(N')$ in a logarithmic scale of $\mathcal{O}[\log_{N'}(N)]$. To this end, we introduce the generalized RY and CZ modules.

Generalized RY modules Generalized RY modules for arbitrary unitary rotations of size N' can be realized by mapping skew-Hermitian matrices for $\mathrm{SU}(N')$ or skew-symmetric matrices for $\mathrm{SO}(N')$. We consider a diverse set of mapping methods below. Let $B \in \mathbb{R}^{N' \times K}$ be a strictly lower-triangular matrix for a rank $K \leq N'$, where the number of non-zero elements is $RN' - R(R+1)/2$, which is identical to $\dim[\mathcal{V}_K(N')]$ for the Stiefel manifold $\mathcal{V}_K(N') \cong \mathrm{SO}(N')/\mathrm{SO}(N' - K)$. Given a skew-symmetric matrix $A = B - B^\top \in \mathbb{R}^{N' \times N'}$, we can generate a corresponding unitary (orthogonal) matrix, e.g., with exponential mapping, Cayley transform, Householder reflection, respectively, as follows:

$$Q_{\mathsf{E}} = \exp(A), \quad Q_{\mathsf{C}} = (I + A)(I - A)^{-1}, \tag{3}$$

$$Q_{\mathsf{H}} = \prod_{k=1}^{K} \big(I - 2\,\mathfrak{N}[B_{:,k}]\mathfrak{N}[B_{:,k}]^\top\big), \tag{4}$$

$$Q_{\mathsf{G}} = \prod_{k=1}^{K} \prod_{n=k+1}^{N} G_{n-k}(B_{n,k}), \tag{5}$$

$$Q_{\mathsf{T}} = \sum_{p=0}^{P} \frac{1}{p!} A^p, \quad Q_{\mathsf{N}} = (I + A) \sum_{p=0}^{P} A^p, \tag{6}$$

where $\mathfrak{N}[\cdot]$ is a normalization operator for canonical coset decomposition (CCD) (Cabrera et al., 2010), and $G_n(\theta)$ denotes the Givens matrix which is identity except that the n and $(n+1)$-th diagonal block is replaced with RY rotation. The mappings of Q_{T} and Q_{N} are respectively approximated versions of Q_{E} and Q_{C} to avoid matrix exponentiation and inversion via Taylor series and Neumann series approximations up to a polynomial order P. Note that Q_{P}, Q_{E} and Q_{G} are identical to RY at $N' = 2$.

Fig. 2(a) illustrates the generalized RY modules to construct trainable orthogonal nodes on Stiefel manifold $\mathcal{V}_K(N')$. After mapping skew-symmetric matrix, truncating the square unitary matrix as $Q_{:K,:}$ can generate right-orthogonal matrix. As all the mappings described above are differentiable, the Lie algebra can be trained via gradient methods. While most mapping methods are studied in other literature (Qiu et al., 2023; Liu et al., 2023b; Chang & Wang, 2021; Wisdom et al., 2016; Bansal et al., 2018; Li et al., 2019), in a PEFT context we can further reduce the number of parameters by masking out the Lie parameters. For example, the top K' columns of B are only trainable, while the other parameters are frozen or null-out. We call K' an intrinsic rank to cover a subset of $\mathcal{V}_K(N')$.

Three potential limitations of this mapping pipeline are i) redundant memory induced before truncation, ii) more computational complexity than the one without mapping, and iii) numerical errors at finite-precision operations. Nevertheless, the memory redundancy will be readily resolved by tensor contraction ordering (Pfeifer et al., 2014), except for Q_{E}. For example, multiplying unitary matrix with a feature vector $x \in \mathbb{R}^{N' \times 1}$ can be recursively contracted as $Q_{\mathsf{T}} x = \sum_p \frac{1}{p!}(B - B^\top)^p x$, which does not require the full matrix Q_{T} but a series of low-rank multiplications with B. Regarding complexity and accuracy, we will discuss in the next subsection, and also quantization impact in Section 3.

The above-mentioned generalized RY modules for $\mathrm{SU}(N')$ are assembled to construct a larger unitary node $\mathrm{SU}(N)$ via generalized STD network shown in Fig. 1(c). However, N should be a power of N'. Using quantum Shannon decomposition (QSD) (Shende et al., 2005) i.e. recursive cosine-sine decomposition (CSD), any unitary matrix $\mathrm{SU}(N)$ can be constructed by $\mathrm{SU}(N_1)$ and $\mathrm{SU}(N_2)$ for lower dimensions such that $N_1 \geq N_2$ and $N_1 + N_2 = N$ for $N > 1$:

$$U = \begin{bmatrix} U_1 & 0 \\ 0 & U_2 \end{bmatrix} \begin{bmatrix} C & -S & 0 \\ 0 & 0 & I \\ S & C & 0 \end{bmatrix} \begin{bmatrix} V_1 & 0 \\ 0 & V_2 \end{bmatrix}, \tag{7}$$

where $U \in \mathrm{SU}(N)$, $U_1, V_2 \in \mathrm{SU}(N_1)$, $U_2, V_1 \in \mathrm{SU}(N_2)$, diagonal cosine and sine matrices such that $C^2 + S^2 = I \in \mathbb{R}^{N_2 \times N_2}$. Hence, power-of-$N'$ rotations such as Kronecker products of Pauli rotations can be still used for arbitrary size of matrices. It hence can solve the power-of-N' limitation.

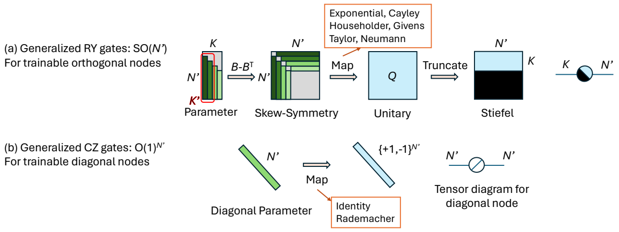


Figure 2. (a) generalized RY modules for orthogonal nodes on Stiefel manifold $\mathcal{V}_K(N')$; (b) generalized CZ modules for diagonal nodes on either $\mathrm{O}(1)^{N'}$ or $\mathbb{R}^{N'}$. Top K' columns are trainable parameters in B as intrinsic rank.

Generalized CZ modules Generalizing CZ modules provides a few options: trainable diagonal matrix in any real number $\mathbb{R}^{N'}$, discrete number, and binary $\{\pm 1\}^{N'}$. Trainable discrete diagonal matrix can be realized e.g. by Gumbel softmax or ReinMax trick (Liu et al., 2024). We refer to a trainable binary diagonal matrix as Rademacher mapping, which can create perfect unitarity and reflection group in $\mathrm{O}(1)^{N'}$. Specifically, Rademacher mapping with ReinMax trick is given as $Q_{\mathsf{R}} = \mathrm{diag}[\mathrm{ReinMax}_\tau([\Lambda, -\Lambda]) \times [+1, -1]]$ with a temperature τ and diagonal parameter $\Lambda \in \mathbb{R}^{N'}$. Fig. 2(b) illustrates diagonal nodes and its tensor diagram. When identity map is used, it can be used as singular values of any matrices under its singular-value decomposition (SVD). Therefore, the use of both trainable unitary matrices and diagonal matrices is sufficient for general representation. It can solve the unitarity limitation of QML.

2.3. Quantum-PEFT: method formulation

PEFT We use the quantum-inspired machine learning modules described above, to realize PEFT. Specifically, we construct a parameter-efficient tensor network by exploiting new modules: trainable unitary nodes parameterized by the Lie algebra to generate Stiefel manifold $\mathcal{V}_K(N)$ via our generalized RY modules; trainable diagonal nodes either on $\mathbb{R}^N$ or $\mathrm{O}(1)^N$ via our generalized CZ modules, and the Pauli parameterization. As one of PEFT tensor networks, we reparametrize the weight updates as a product of trainable unitary matrices $U \in \mathcal{V}_K(N) \subset \mathbb{R}^{N \times K}$ and

Table 1. Comparison of different PEFT methods.

Method	# Trainable Parameters
LoRA (TTD)	$2NK$
AdaLoRA (CP)	$2NK + K$
Quantum-PEFT (TD: Q_{T})	$2NK - K^2$ (for $K' = K$, $N' = N$)
Quantum-PEFT (TD: Q_{P})	$2(2L + 1)\log_2(N) + K$

Table 2. GLUE benchmark. Matthew's correlation for CoLA, average correlation for STS-B, and accuracy for other tasks.

Method	# Trainable Parameters	SST-2	CoLA	RTE	MRPC	STS-B
FT	184M	95.63	69.19	83.75	89.46	91.60
BitFit	0.1M	94.84	66.96	78.70	87.75	91.35
HAdapter	0.61M	95.30	67.87	85.56	89.22	91.30
PAdapter	0.60M	95.53	69.48	84.12	89.22	91.52
HAdapter	0.31M	95.41	67.65	83.39	89.25	91.31
PAdapter	0.30M	94.72	69.06	84.48	89.71	91.38
LoRA	0.33M	94.95	68.71	85.56	89.71	**91.68**
AdaLoRA	0.32M	95.80	**70.04**	**87.36**	90.44	91.63
Quantum-PEFT	**0.013M**	**95.85**	67.85	86.57	**90.78**	91.06

$V \in \mathcal{V}_K(M) \subset \mathbb{R}^{M\times K}$ generated by our generalized RY modules, and a trainable diagonal matrix $\Lambda \in \mathbb{R}^{K\times K}$ generated by our generalized CZ modules. Specifically, the weight update ΔW for a weight matrix $W \in \mathbb{R}^{N\times M}$ is given by: $\Delta W = U\Lambda V^\top$. In this SVD form, the number of trainable parameters depends on the chosen parametrization for the underlying orthogonal matrices. Specifically, the Taylor parametrization Q_{T} for the maximum decomposition size $N' = N$ yields $2NK - K^2$ trainable parameters (with $K' = K$), while the Pauli parametrization Q_{P} achieves an extremely compact representation with only $2(2L+1)\log_2(N)+K$ parameters, scaling logarithmically with the matrix dimension N. The underlying parametrizations induced by our generalized RY modules spanning orthogonal group can effectively capture a full-rank weight update. This contrasts with AdaLoRA (Zhang et al., 2023), which uses approximate orthogonality imposed by regularization terms in loss, failing to reduce the number of trainable parameters being limited by the low-rank decomposition. Consequently, Quantum-PEFT enables orders-of-magnitude parameter reduction compared to conventional LoRA-based approaches, while retaining the expressive power of effectively full-rank representations.

Parameter efficiency Fig. 5 shows tensor diagrams under tensor network interpretation of LoRA variants. As shown in Table 1, the LoRA uses two K-rank matrices, having $2NK$ parameters in total for a matrix size $N \gg K$. This is known as 2-mode tensor train decomposition (TTD). AdaLoRA uses approximated SVD, but unitarity is not perfectly imposed, leading to $K(K+1)$ redundant parameters and extra regularization terms. From the tensor network perspective, AdaLoRA falls under Canonical Polyadic (CP) decomposition which does not strictly assume orthogonality. Using the Lie algebra, Quantum-PEFT can readily realize the non-redundant parameterization for trainable SVD (i.e., 2-mode Tucker decomposition: TD). With QSD, Pauli parameterization can further reduce the number of parameters for arbitrary tensor networks into a logarithmic scale. More discussions of other tensor networks are found in Appendix C.2. When we apply Hadamard product of tensor networks like LoHA (Yeh et al., 2024), Quantum-PEFT can further increase the capacity.

3. Experiments

In this section, we evaluate Quantum-PEFT on the GLUE benchmark (Wang et al., 2019). Our experiments are not to claim that Quantum-PEFT always improves the accuracy compared to LoRA, but to show that Quantum-PEFT can maintain a competitive level of accuracy with orders-of-magnitude fewer parameters. Results on **E2E benchmark** and **Vision Transformer with CIFAR-10** are deferred to Appendices D.2 and D.3, respectively.

Our experiment follows the set-up in (Zhang et al., 2023). The fine-tuning is applied on DeBERTaV3-base (He et al., 2021b). We compare Quantum-PEFT with the following baselines: Full parameters fine-tuning (FT), LoRA (Hu et al., 2021), BitFit (Zaken et al., 2022), adapter tuning with Houlsby adapter (HAdapter) (Houlsby et al., 2019), adapter tuning with Pfeiffer adapter (PAdapter) (Pfeiffer et al., 2021), and AdaLoRA (Zhang et al., 2023). Detailed hyperparameters can be found in Appendix D. The results in Table 2 show that in both SST-2 and MRPC tasks, Quantum-PEFT can outperform AdaLoRA. On other tasks, Quantum-PEFT can still achieve comparable performance with other baselines. Notably, Quantum-PEFT only requires 0.013 million parameters, which are 25 times fewer than LoRA.

4. Conclusions

We introduced Quantum-PEFT, a novel framework leveraging quantum ML principles to achieve extremely parameter-efficient fine-tuning of large pre-trained models. Through reparameterizing neural network layers as generalized quantum circuits, we represent weight updates using highly compact unitary matrix embeddings. Unlike prior low-rank adaptation methods bottlenecked by linear parameter growth, Quantum-PEFT's parameter count scales logarithmically with the model size via Pauli parametrization and can achieve lower parameter number than lowest-rank LoRA. Experiments across language and vision benchmarks show excellent capabilities, achieving orders-of-magnitudes higher compression rates than LoRA while maintaining competitive performance.

References

Aghajanyan, A., Zettlemoyer, L., and Gupta, S. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. *arXiv preprint arXiv:2012.13255*, 2020.

AI@Meta. Llama 3 model card. 2024. URL https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md.

Arjovsky, M., Shah, A., and Bengio, Y. Unitary evolution recurrent neural networks. In *International conference on machine learning*, pp. 1120–1128. PMLR, 2016.

Bansal, N., Chen, X., and Wang, Z. Can we gain more from orthogonality regularizations in training deep CNNs? In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 4266–4276, Red Hook, NY, USA, 2018. Curran Associates Inc.

Bershatsky, D., Cherniuk, D., Daulbaev, T., and Oseledets, I. LoTR: Low tensor rank weight adaptation. *arXiv preprint arXiv:2402.01376*, 2024.

Biamonte, J., Wittek, P., Pancotti, N., Rebentrost, P., Wiebe, N., and Lloyd, S. Quantum machine learning. *Nature*, 549(7671):195–202, 2017.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901, 2020.

Cabrera, R., Strohecker, T., and Rabitz, H. The canonical coset decomposition of unitary matrices through Householder transformations. *Journal of Mathematical Physics*, 51(8), 2010.

Cerezo, M., Sone, A., Volkoff, T., Cincio, L., and Coles, P. J. Cost function dependent barren plateaus in shallow parametrized quantum circuits. *Nature communications*, 12(1):1791, 2021.

Chang, H.-Y. and Wang, K. L. Deep unitary convolutional neural networks. In *Artificial Neural Networks and Machine Learning–ICANN 2021: 30th International Conference on Artificial Neural Networks, Bratislava, Slovakia, September 14–17, 2021, Proceedings, Part II 30*, pp. 170–181. Springer, 2021.

Chavan, A., Liu, Z., Gupta, D., Xing, E., and Shen, Z. One-for-all: Generalized LoRA for parameter-efficient fine-tuning. *arXiv preprint arXiv:2306.07967*, 2023.

Chen, S., Ge, C., Tong, Z., Wang, J., Song, Y., Wang, J., and Luo, P. AdaptFormer: Adapting vision transformers for scalable visual recognition. *Advances in Neural Information Processing Systems*, 35:16664–16678, 2022.

Chen, X., Liu, J., Wang, Y., Brand, M., Wang, G., Koike-Akino, T., et al. SuperLoRA: Parameter-efficient unified adaptation of multi-layer attention modules. *arXiv preprint arXiv:2403.11887*, 2024.

Dallaire-Demers, P.-L. and Killoran, N. Quantum generative adversarial networks. *Physical Review A*, 98(1):012324, 2018.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. ImageNet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Edalati, A., Tahaei, M., Kobyzev, I., Nia, V. P., Clark, J. J., and Rezagholizadeh, M. Krona: Parameter efficient tuning with Kronecker adapter. *arXiv preprint arXiv:2212.10650*, 2022.

Farhi, E. and Neven, H. Classification with quantum neural networks on near term processors. *arXiv preprint arXiv:1802.06002*, 2018.

Hao, T., Chen, H., Guo, Y., and Ding, G. Consolidator: Mergable adapter with group connections for visual adaptation. In *The Eleventh International Conference on Learning Representations*, 2022.

Hayou, S., Ghosh, N., and Yu, B. LoRA+: Efficient low rank adaptation of large models. *arXiv preprint arXiv:2402.12354*, 2024.

He, J., Zhou, C., Ma, X., Berg-Kirkpatrick, T., and Neubig, G. Towards a unified view of parameter-efficient transfer learning. In *International Conference on Learning Representations*, 2021a.

He, P., Liu, X., Gao, J., and Chen, W. DeBERTa: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*, 2020.

He, P., Gao, J., and Chen, W. DeBERTaV3: Improving DeBERTa using electra-style pre-training with gradient-disentangled embedding sharing. *arXiv preprint arXiv:2111.09543*, 2021b.

Helfrich, K., Willmott, D., and Ye, Q. Orthogonal recurrent neural networks with scaled Cayley transform. In *International Conference on Machine Learning*, pp. 1969–1978. PMLR, 2018.

Henderson, M., Shakya, S., Pradhan, S., and Cook, T. Quanvolutional neural networks: powering image recognition with quantum circuits. *Quantum Machine Intelligence*, 2 (1):2, 2020.

Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., Attariyan, M., and Gelly, S. Parameter-efficient transfer learning for NLP. In *International Conference on Machine Learning*, pp. 2790–2799. PMLR, 2019.

Hu, E. J., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W., et al. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2021.

Huang, H., Zhou, X., and He, R. Orthogonal Transformer: An Efficient Vision Transformer Backbone with Token Orthogonalization. In *Advances in Neural Information Processing Systems*, October 2022.

Huggins, W., Patil, P., Mitchell, B., Whaley, K. B., and Stoudenmire, E. M. Towards quantum machine learning with tensor networks. *Quantum Science and technology*, 4(2):024001, 2019.

Jie, S. and Deng, Z.-H. Fact: Factor-tuning for lightweight adaptation on vision transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 1060–1068, 2023.

Jing, L., Shen, Y., Dubcek, T., Peurifoy, J., Skirlo, S., LeCun, Y., Tegmark, M., and Soljačić, M. Tunable efficient unitary neural networks (EUNN) and their application to RNNs. In *International Conference on Machine Learning*, pp. 1733–1741. PMLR, 2017.

Karimi Mahabadi, R., Henderson, J., and Ruder, S. Compacter: Efficient low-rank hypercomplex adapter layers. *Advances in Neural Information Processing Systems*, 34: 1022–1035, 2021.

Kitaev, A. Y. Quantum computations: algorithms and error correction. *Russian Mathematical Surveys*, 52(6):1191, 1997.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

Lester, B., Al-Rfou, R., and Constant, N. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 3045–3059, 2021.

Li, J., Li, F., and Todorovic, S. Efficient Riemannian Optimization on the Stiefel Manifold via the Cayley Transform. In *The 8th International Conference on Learning Representations*, September 2019.

Li, X. L. and Liang, P. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 4582–4597, 2021.

Liu, J., Koike-Akino, T., Wang, P., Brand, M., Wang, Y., and Parsons, K. LoDA: Low-dimensional adaptation of large language models. *NeurIPS'23 Workshop on on Efficient Natural Language and Speech Processing*, 2023a.

Liu, L., Dong, C., Liu, X., Yu, B., and Gao, J. Bridging discrete and backpropagation: Straight-through and beyond. *Advances in Neural Information Processing Systems*, 36, 2024.

Liu, W., Qiu, Z., Feng, Y., Xiu, Y., Xue, Y., Yu, L., Feng, H., Liu, Z., Heo, J., Peng, S., Wen, Y., Black, M. J., Weller, A., and Schölkopf, B. Parameter-Efficient Orthogonal Finetuning via Butterfly Factorization. In *The Twelfth International Conference on Learning Representations*, October 2023b.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Lloyd, S. and Weedbrook, C. Quantum generative adversarial learning. *Physical review letters*, 121(4):040502, 2018.

Mhammedi, Z., Hellicar, A., Rahman, A., and Bailey, J. Efficient orthogonal parametrisation of recurrent neural networks using Householder reflections. In *International Conference on Machine Learning*, pp. 2401–2409. PMLR, 2017.

Novikov, A., Podoprikhin, D., Osokin, A., and Vetrov, D. P. Tensorizing neural networks. *Advances in neural information processing systems*, 28, 2015.

Novikova, J., Dušek, O., and Rieser, V. The E2E dataset: New challenges for end-to-end generation. *arXiv preprint arXiv:1706.09254*, 2017.

Orús, R. A practical introduction to tensor networks: Matrix product states and projected entangled pair states. *Annals of physics*, 349:117–158, 2014.

Pérez-Salinas, A., Cervera-Lierta, A., Gil-Fuster, E., and Latorre, J. I. Data re-uploading for a universal quantum classifier. *Quantum*, 4:226, 2020.

Pfeifer, R. N., Haegeman, J., and Verstraete, F. Faster identification of optimal contraction sequences for tensor networks. *Physical Review E*, 90(3):033315, 2014.

Pfeiffer, J., Kamath, A., Rücklé, A., Cho, K., and Gurevych, I. AdapterFusion: Non-destructive task composition for transfer learning. In *16th Conference of the European Chapter of the Associationfor Computational Linguistics, EACL 2021*, pp. 487–503. Association for Computational Linguistics (ACL), 2021.

Qiu, Z., Liu, W., Feng, H., Xue, Y., Feng, Y., Liu, Z., Zhang, D., Weller, A., and Schölkopf, B. Controlling text-to-image diffusion by orthogonal finetuning. *Advances in Neural Information Processing Systems*, 36:79320–79362, 2023.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Rebentrost, P., Mohseni, M., and Lloyd, S. Quantum support vector machine for big data classification. *Physical review letters*, 113(13):130503, 2014.

Roberts, C., Milsted, A., Ganahl, M., Zalcman, A., Fontaine, B., Zou, Y., Hidary, J., Vidal, G., and Leichenauer, S. TensorNetwork: A library for physics and machine learning. *arXiv preprint arXiv:1905.01330*, 2019.

Romero, J., Olson, J. P., and Aspuru-Guzik, A. Quantum autoencoders for efficient compression of quantum data. *Quantum Science and Technology*, 2(4):045001, 2017.

Schuld, M., Sinayskiy, I., and Petruccione, F. An introduction to quantum machine learning. *Contemporary Physics*, 56(2):172–185, 2015.

Schuld, M., Bergholm, V., Gogolin, C., Izaac, J., and Killoran, N. Evaluating analytic gradients on quantum hardware. *Physical Review A*, 99(3):032331, 2019.

Shende, V. V., Bullock, S. S., and Markov, I. L. Synthesis of quantum logic circuits. In *Proceedings of the 2005 Asia and South Pacific Design Automation Conference*, pp. 272–275, 2005.

Sim, S., Johnson, P. D., and Aspuru-Guzik, A. Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms. *Advanced Quantum Technologies*, 2(12):1900070, 2019.

Suykens, J. A. K. Generating quantum-measurement probabilities from an optimality principle. *Phys. Rev. A*, 87:052134, May 2013. doi: 10.1103/PhysRevA.87.052134. URL https://link.aps.org/doi/10.1103/PhysRevA.87.052134.

Vidal, G. Class of quantum many-body states that can be efficiently simulated. *Physical review letters*, 101(11):110501, 2008.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. R. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=rJ4km2R5t7.

Wisdom, S., Powers, T., Hershey, J., Le Roux, J., and Atlas, L. Full-capacity unitary recurrent neural networks. *Advances in neural information processing systems*, 29, 2016.

Yeh, S.-Y., Hsieh, Y.-G., Gao, Z., Yang, B. B. W., Oh, G., and Gong, Y. Navigating text-to-image customization: From lyCORIS fine-tuning to model evaluation. In *The Twelfth International Conference on Learning Representations*, 2024.

Zaken, E. B., Goldberg, Y., and Ravfogel, S. BitFit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 1–9, 2022.

Zhang, Q., Chen, M., Bukharin, A., He, P., Cheng, Y., Chen, W., and Zhao, T. Adaptive budget allocation for parameter-efficient fine-tuning. In *The Eleventh International Conference on Learning Representations*, 2023.

Zheng, J., Gao, Q., and Lü, Y. Quantum graph convolutional neural networks. In *2021 40th Chinese Control Conference (CCC)*, pp. 6335–6340. IEEE, 2021.

Zhu, J., Greenewald, K., Nadjahi, K., Borde, H. S. d. O., Gabrielsson, R. B., Choshen, L., Ghassemi, M., Yurochkin, M., and Solomon, J. Asymmetry in low-rank adapters of foundation models. *arXiv preprint arXiv:2402.16842*, 2024.

Zi, B., Qi, X., Wang, L., Wang, J., Wong, K.-F., and Zhang, L. Delta-LoRa: Fine-tuning high-rank parameters with the delta of low-rank matrices. *arXiv preprint arXiv:2309.02411*, 2023.

A. Related work

Parameter-efficient fine-tuning (PEFT) Parameter-efficient fine-tuning (PEFT) methods allow significantly lower model training cost for different downstream tasks. A plethora of methods have been proposed for PEFT (Houlsby et al., 2019; Aghajanyan et al., 2020; Hu et al., 2021; Edalati et al., 2022; Lester et al., 2021; Li & Liang, 2021; He et al., 2021a; Karimi Mahabadi et al., 2021; Chen et al., 2022; Jie & Deng, 2023; Hao et al., 2022; Houlsby et al., 2019; Pfeiffer et al., 2021), among which reparameterization-based techniques (Aghajanyan et al., 2020; Hu et al., 2021; Edalati et al., 2022) bear the most relevance to our study, where the model architecture is not changed but reparametrized with a lower number of trainable parameters. Low-rank adaptation (LoRA) (Hu et al., 2021) has shown promising results by updating the pretrained weight matrix through the addition of a product of two low-rank matrices. The simplicity of this low-rank weight reparameterization has led to its widespread adoption (Zi et al., 2023; Chavan et al., 2023; Hayou et al., 2024; Zhu et al., 2024). Many variants were introduced, e.g., methods based on Kronecker product, such as KronA (Edalati et al., 2022) and LoKr (Yeh et al., 2024), Hadamard product, such as LoHA (Yeh et al., 2024), tensor rank decomposition, such as LoTR (Bershatsky et al., 2024) and SuperLoRA (Chen et al., 2024), and nonlinear mappings, such as LoDA (Liu et al., 2023a).

Unitary-constrained PEFT AdaLoRA (Zhang et al., 2023) introduces dynamic rank adjustment during fine-tuning, with additional regularizer for orthogonality. Unlike AdaLoRA that involves inexact orthogonality constraints and extra regularization terms, Quantum-PEFT directly parameterizes full-rank unitary matrices via efficient quantum circuit embeddings. Orthogonal fine-tuning (OFT) (Qiu et al., 2023; Liu et al., 2023b) employs a unitary matrix to transform the pretrained weights, showing stronger generalization than LoRA. Despite its enhanced generalization performance, OFT typically requires more trainable parameters than LoRA, highlighting the need for more parameter-efficient PEFT methods. In addition, OFT methods rely on an expensive Cayley transform and a structured block-diagonal matrix.

Unitary-constrained machine learning Unitary constraints in machine learning have been explored extensively due to their potential to make training more stable and improve generalization. (Chang & Wang, 2021) uses deep unitary convolution based on an exponential map with Lie parameters. In recurrent neural networks (RNNs), several methods with unitary weights have been proposed, such as (Arjovsky et al., 2016), which uses unitary evolution without requiring expensive eigen-value decomposition, and (Jing et al., 2017), which employs unitary neural networks. Different parametrizations have been used, including orthogonal weight matrices through the Cayley transform (Helfrich et al., 2018) and Householder reflection for RNNs (Mhammedi et al., 2017) and ViT (Huang et al., 2022). Optimization of deep learning models over the Stiefel manifold has been studied in multiple works (Wisdom et al., 2016; Bansal et al., 2018; Li et al., 2019).

Quantum machine learning Main relevant concepts in quantum machine learning include expressibility and entangling (Sim et al., 2019). Variational principles for quantum neural networks (QNNs) were studied in (Farhi & Neven, 2018), with extensions for quanvolutional networks (Henderson et al., 2020), quantum autoencoders (QAEs) (Romero et al., 2017), quantum support vector machines (QSVMs) (Suykens, 2013; Rebentrost et al., 2014), quantum graph neural networks (QGNNs) (Zheng et al., 2021), and quantum generative adversarial networks (QGANs) (Lloyd & Weedbrook, 2018; Dallaire-Demers & Killoran, 2018). It was proved that QNNs hold the universal approximation property (Pérez-Salinas et al., 2020). More importantly, quantum circuits can be analytically differentiable with a parameter-shift rule (Schuld et al., 2019) that enables stochastic gradient optimization of QNN.

Tensor network Tensor network (Roberts et al., 2019) provides a way to represent/manipulate multi-dimensional arrays of data by factorizing into a network of lower-dimensional tensors. Many tensor rank decomposition methods are used for tensor networks, including matrix product state (MPS) and tree tensor network (TTN) (Huggins et al., 2019), based on tensor train decomposition (TTD) and Hierarchical Tucker decomposition (HTD), respectively. More sophisticated ones used in QML include multi-scale entanglement renormalization ansatz (MERA) (Vidal, 2008) and projected entangled-pair states (PEPS) (Orús, 2014). Tensorization provides efficient parameterization of DNN architecture (Novikov et al., 2015).

B. Comparison of unitary mapping

Fig. 3 shows the comparison of different unitary mapping methods over different matrix size N for a rank of $K = 4$. We examined the unitarity test and speed bench on RTX6000 GPU for forward and backward processing. The unitarity error measures an averaged ℓ_∞ norm of $\|QQ^\top - I\|_\infty$ over a batch size of 32 and 10 random seeds. The exponential mapping

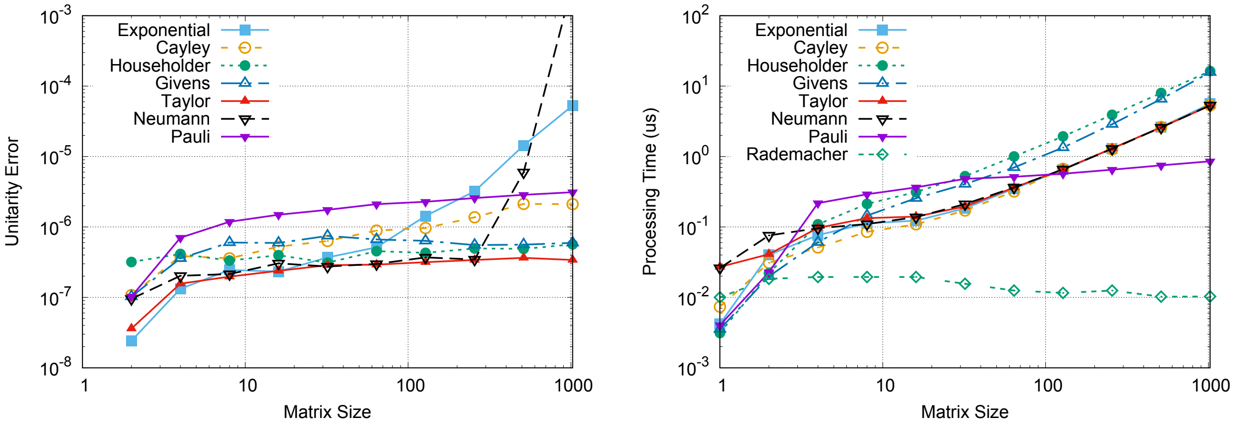


Figure 3. Unitarity error analysis and speed bench including forward and backward passes for different unitary mapping methods as a function of matrix size of N for a rank of $K = 4$ on an NVIDIA RTX6000 GPU 24GB.

uses `torch.linalg.matrix_exp`, and matrix inversion for Cayley transform uses `torch.linalg.solve`. We assume $P = 18$ polynomial order for Taylor and Neumann series. It was found that Neumann series and exponential mapping become inaccurate as the matrix size is increased. While Pauli parameterization has relatively higher error than the rest of methods, it can be much faster in large matrix size. Householder reflections and Givens rotations had slower behaviors due to sequential nature. Although Rademacher diagonal matrix of $\{\pm 1\}^K$ has a low complexity and perfect unitarity (here, we used ReinMax trick), it alone does not cover the Stiefel manifold $\mathcal{V}_K(N)$. Overall, Taylor series method showed a good trade-off between accuracy and speed. Note that most large foundation models use thousands for a matrix size of N per weight. Therefore, the accuracy and speed at large matrix size regimes are important. With these trade-offs in mind, in the experiments we evaluate the Taylor Q_{T} and Pauli Q_{P} parametrizations, where Pauli gives logarithmic number of trainable parameters in the ambient dimension and Taylor shows satisfactory speed for larger models.

C. Further details and discussions on Quantum-PEFT

To further elaborate on Quantum-PEFT, we provide the tensor network diagrams in Figure 4 exemplifying its mechanism w.r.t. other LoRA-based methods.

C.1. Quantum-inspired PEFT modules

Generalized measurements As well as generalized-RY gates and CZ gates, we introduce generalized measurement module. Although quantum operation is linear, quantum measurement can be nonlinear in general. Hence, motivated from the quantum measurement to solve the linearity constraint, we can impose nonlinearity using activation functions. Using log-softmax after squaring corresponds to measuring quantum state probability. For our case, such nonlinear activations can be imposed at any mid-circuit operations. In Fig. 4, we introduce a new tensor diagram with delay symbols representing the nonlinear node. Nonlinear mapping can be also trainable when using another multi-layer perceptron (MLP) as used in LoDA. Letting $f(\cdot)$ be such a nonlinear function, tensor contraction can be done via *nonlinear* Einstein sum: $f^{\text{out}}(\sum f^{\text{in}}(\prod Q^{[k]}_{i,j}))$ for parent tensor nodes $\{Q^{[k]}\}$, where f^{out} and f^{in} denote outer nonlinearity and inner nonlinearity, respectively. Note that the nonlinear nodes can only pass the data after tensor contraction from all ancestor nodes.

Quantization To further save memory, we can use a standard integer quantization for trainable parameter: θ: $\theta_{\text{q}} = \text{round}((\theta - \mu)/\beta)\beta + \mu$, where scale value $\beta = (\theta_{\max} - \theta_{\min})/(2^n - 1)$ and zero value $\mu = \theta_{\min}$ for n-bit quantization. The maximum $\theta_{\max}$ and minimum values $\theta_{\min}$ are obtained in a chunk of group size g. When the quantization is applied on the Lie parameters, we employ the straight-through trick for quantization-aware training (QAT), i.e., $\theta := \theta_{\text{q}} + \theta - \theta.\text{detach}()$, where $.\text{detach}()$ means no gradient passing. Once trained, the required memory will be $n + 32/g$ bits per Lie parameter when β and μ use floating-point (FP) 16 bits precision.

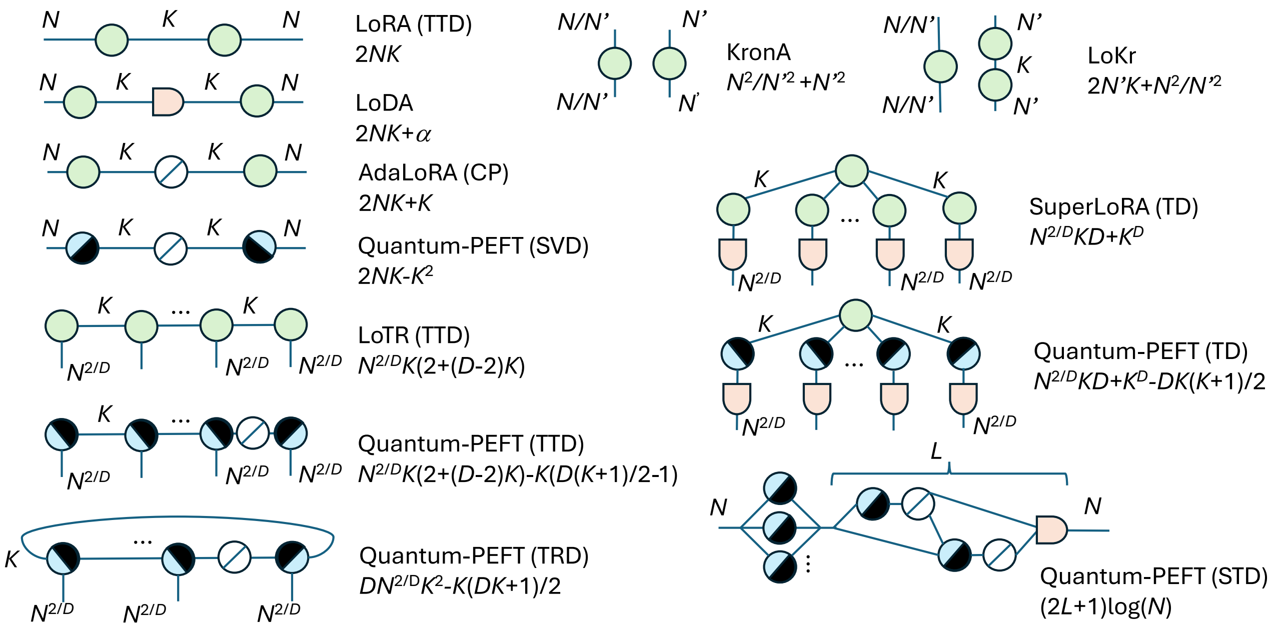


Figure 4. Tensor diagrams of Quantum-PEFT and LoRA variants in tensor network perspectives for a matrix size of N and rank K. The number of parameters are also present. Circle denotes dense multi-linear tensor node. Slashed open circles denote diagonal node. Half-closed circles denote unitary node. Delay symbols denote nonlinear nodes.

C.2. Tensor network implication

Fig. 4 shows tensor diagrams for various LoRA variants. Our Quantum-PEFT framework can unify them with reduced number of parameters by exploiting trainable orthogonal nodes, trainable diagonal nodes, and trainable nonlinear nodes. As mentioned, LoRA uses 2-mode tensor train decomposition (TTD) which is also known as matrix product state (MPS) tensor network. LoDA introduced the nonlinear node in tensor network. AdaLoRA is based on CP decomposition, which has parameter redundant. LoTR extends LoRA towards higher-mode TTD. SuperLoRA uses another tensor network based on higher-oder Tucker decomposition (TD), while nonlinear mapping is optionally introduced. In fact, TTD and TD can be normalized except one node, and hence our Quantum-PEFT based on the Lie algebra can eliminate the redundant parameters to improve the efficiency for LoRA, LoTR and SuperLoRA. Similarly, our framework provides parameter-efficient unitary nodes in most other tensor networks including tensor ring decomposition (TRD), hierarchical Tucker decompostion (HTD) a.k.a. tree tensor network (TTN), multi-scale entanglement renormalization ansatz (MERA), and projected entangled pair states (PEPS). As descussed, Pauli parameterization based on STD ansatz can further reduce the number of parameters for those tensor networks into a logarithmic scale. Note that STD parameterization can be regarded as a renormalization step of each orthogonal node in tensor networks. Fig. 6 shows an example of the STD renormalization step when N is 3-folded into $N' = N^{1/3}$. The total number of parameters to represent the unitary node for $\mathcal{V}_K(N)$ can be reduced in a logarithmic order of $\log_{N'}(N)$. When $K = 1, N = 3^9, N' = 3^2, L = 1$, it becomes 180 from 729. Reducing the size of N' can further improve the parameter efficiency.

Table 3 shows an example result for ViT CIFAR10 transfer learning task, using Taylor parameterization (with $K = K' = 4$ and $P = 18$) for different tensor networks, including CP, TRD, HTD (TTN), TD, and TTD (MPS). We find that all tensor networks offer competitive performance to LoRA.

C.3. Intrinsic rank impact

We introduced an intrinsic rank K' to reduce the trainable parameters than the specified rank K, by masking the top K' columns of Lie parameters. In Table 4, we show the impact of intrinsic rank K' for Taylor parameterization on ViT transfer learning task. We can see that decreasing the intrinsic rank K' gradually degrade the accuracy and the required number of

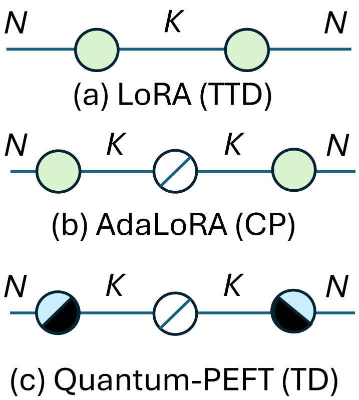


Figure 5. Tensor diagram of LoRA variants.

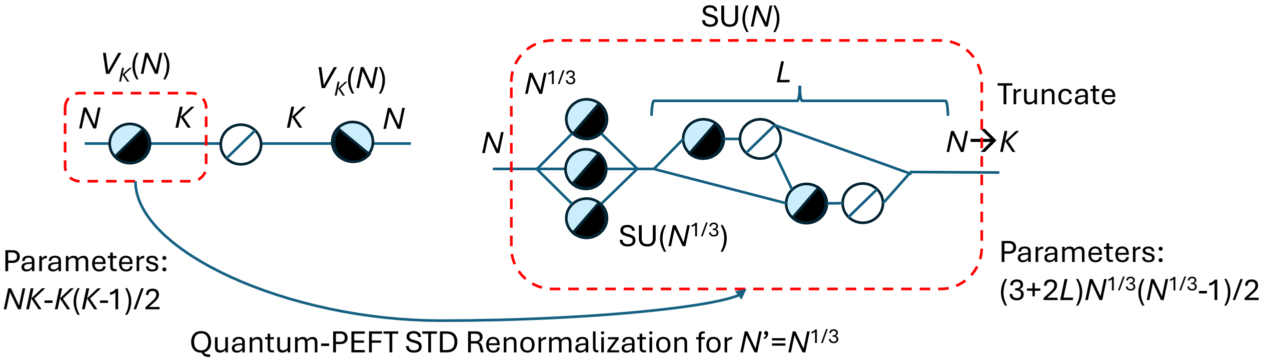


Figure 6. STD renormalization step example when $N' = N^{1/3}$. The total number of parameters is reduced from 729 to 180 for a unitary node when $K = 1, N = 3^6, N' = 3^2, L = 1$.

parameters. While the subspace rank is $K = 8$, the number of parameters can be effectively $K' \leq K$. The performance degradation from $K' = 8$ to $K' = 1$ is only 0.49%, and more importantly the accuracy is much better than LoRA in Table 8. For example, LoRA with $K = 1$ has an accuracy of 98.15%, while Quantum-PEFT Q_T parameterization with $K = 8$ and $K' = 1$ has 98.38%, at the comparable number of parameters. It shows the great potential of masking out the Lie parameters while keeping higher subspace rank.

C.4. Broader impacts and future work

It is interesting to investigate how we can further reduce the memory for trainable parameters by employing quantization or pruning.

Mixed-precision tensor network One could consider a mixed-precision tensor network, where each tensor node and its parameter group can have different precisions. Fig. 7(a) shows an example of Quantum-PEFT in 3-dimensional TRD tensor network. The TRD is formulated by 3 unitary nodes $\{Q^{[k]}\}$ and 1 diagonal node $\varLambda$. Specifically the (i, j, k)-th element is given by nonlinear Einstein sum: $W_{i,j,k} = f^{\text{out}}(\sum_{l,m,n} f^{\text{in}}(Q^{[1]}_{l,i,m} Q^{[2]}_{m,j,n} \varLambda_{n,n} Q^{[3]}_{n,k,l}))$. As shown in Fig. 7(b), each node has trainable parameters θ, and we can adaptively assign more bits or fewer bits depending on the group range $\varDelta_i = \theta_{i,\max} - \theta_{i,\min}$ for the i-th group. For example, the bit loading may use the following strategy: $q_i = \text{round}(q \log_2(\varDelta_i^\kappa / \bar{\varDelta}))$ with an average range $\bar{\varDelta} = \mathbb{E}[\varDelta_i^\kappa]$ where q_i bits are assigned for the i-th group with an exponent $\kappa >= 0$. When $\kappa = 0$, it reduces to uniform bit loading: i.e., $q_i = q$ for all group i. More sophisticated but time-consuming strategy is to consider the quantization error of the weight matrix $\min |W_\text{q} - W|$, which requires combinatorial optimization.

Table 3. Different tensor network results with Taylor parameterization for ViT transfer learning from ImageNet-21k to CIFAR10. Base ViT is not quantized.

Method	CP	TRD	HTD (TTN)	TD	TTD (MPS)
# Parameters	0.074M	0.147M	0.026M	0.074M	0.111M
Accuracy	98.53%	98.14%	98.11%	98.05%	98.81%

Table 4. Impact of intrinsic rank K' for Taylor parameterization for ViT transfer learning from ImageNet-21k to CIFAR10. Base ViT is quantized with 3-bit integers. Tensor rank is $K = 8$.

Intrinsic rank K'	1	2	3	4	5	6	7	8
# Parameters	0.037M	0.074M	0.111M	0.147M	0.184M	0.221M	0.257M	0.294M
Accuracy	98.38%	98.52%	98.76%	98.74%	98.63%	98.79%	98.81%	98.87%

When the bit allocation is zero (i.e., $\varDelta_i$ is close to zero) as shown in Fig. 7(c), it corresponds to structural pruning except that the masked group can still hold non-zero values μ. Further fine-grained pruning is also possible by nulling out θ if the value magnitude is smaller than a threshold. Therefore, it can accomplish an adaptive rank mechanism similar to AdaLoRA.

Pretrained model compression In fact, Quantum-PEFT framework can also be applicable to compress the pretrained model before adaptation. Tensor rank decomposition, quantization and pruning can be applied to pretrained model before transfer learning tasks, similar to Q-LoRA, R-LoDA, and S-LoDA. For ViT transfer learning task, we evaluated 3-bit quantization of pre-trained models.

D. Additional experimental results and detailed setups

D.1. GLUE benchmark

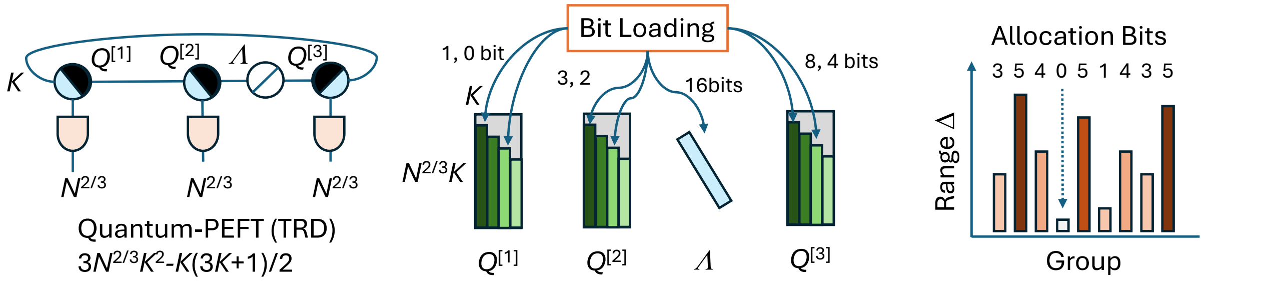
Below, we provide a summary of the tasks in the GLUE benchmark that are used in this work.

- SST-2: stands for The Stanford Sentiment Treebank, a dataset on sentiment analysis tasks with two labels. The size of the training set is 67k, and the size of the test set is 1.8k.
- CoLA, represents The Corpus of Linguistic Acceptability, a dataset on sentence classification with two labels. It consists of 8.5k training data and 1k test data.
- RTE: stands for The Recognizing Textual Entailment, including 2.5k training data points and 3k test data points.
- MRPC: represents The Microsoft Research Paraphrase Corpus, a dataset on pairwise text classification with 3.7k training points and 1.7k test points.
- STS-B: represents The Semantic Textual Similarity Benchmark, a task on measuring text similarity with 7k training points and 1.4k test points.

We fine-tune the query/key/value projection matrices, the output projection in the attention block, and the weight matrices in two-layer MLPs. For all of the baselines, we follow the hyperparameters in (Zhang et al., 2023). For Quantum-PEFT, we use Q_P with $L = 1$ in all tasks. We select the best learning rate by parameters sweep. We conduct five runs with different random seeds and report the mean. We use the same number of training epochs as in AdaLoRA. Due to limited computing resources, we focus on tasks with training instances less than 100k, including SST-2, CoLA, RTE, MRPC, and STS-B. We select the same number of epochs for Quantum-PEFT as in AdaLoRA. We perform a hyperparameters sweep for the learning rate over $\{0.01, 0.03, 0.06, 0.001, 0003, 0.006\}$. We select the best learning rate and the best checkpoints over each epoch. We present the hyperparameters for Quantum-PEFT in Table 5.

D.2. E2E benchmark

We fine-tune GPT-2 (Radford et al., 2019) Medium on the common E2E natural language generation benchmark (Novikova et al., 2017), following the setups of (Hu et al., 2021). GPT2-Medium has 354M parameters with 24 transformer layers.



(a) TRD Tensor Network (b) Mixed-Precision Bit Loading (c) Adaptive Allocation/Pruning

Figure 7. Mixed-precision Quantum-PEFT in 3-dimensional TRD tensor network. Each tensor node and tensor parameter can have non-uniform bit assignments. Adaptive bit loading depends on group range Δ. Assignment of 0 bit corresponds to adaptive structural pruning.

Table 5. Hyperparameter configurations for Quantum-PEFT on the GLUE benchmark.

Hyperparameter	SST-2	CoLA	RTE	MRPC	STS-B
# GPUs	1	1	1	1	1
Optimizer	AdamW	AdamW	AdamW	AdamW	AdamW
Learning Rate Schedule	Linear	Linear	Linear	Linear	Linear
Weight Decay	0.01	0.01	0.01	0.01	0.01
Batch Size	256	128	128	128	128
Epochs	24	25	50	30	25
Warmup ratio	0.1	0.1	0.1	0.1	0.1
Max sequence length	128	64	320	320	128
Rank K	3	3	3	3	3
α	32	32	32	32	32
Learning Rate	0.006	0.01	0.06	0.01	0.03
Unitary Parametrization	Q_P $(L=1)$	Q_P $(L=1)$	Q_P $(L=1)$	Q_P $(L=1)$	Q_P $(L=1)$

The E2E benchmark consists of 42,200 samples for training, 4,600 for validation, and 4,600 for testing. We compare Quantum-PEFTwith LoRA (Hu et al., 2021), AdaLoRA (Zhang et al., 2023), and full FT. Full FT results are sourced from prior works (Zi et al., 2023). For fair comparison, we use the same training settings and hardware for LoRA, AdaLoRA, and Quantum-PEFT. We train LoRA, AdaLoRa, and Quantum-PEFTusing 4 NVIDIA A100 GPUs using the code provided by the respective authors. We apply LoRA, AdaLoRA, and Quantum-PEFTto the query and value projection layers in each attention block and use the same number of training epochs, batch size, and LoRA scaling, except different learning rate. Table 6 lists hyperparameters for the experiment on transfer learning task of E2E benchmark.

Table 7 shows the results for E2E Challenge dataset on 5 evaluation metrics. Quantum-PEFT's performance is on par or better than LoRA with approximately 4 times less trainable parameters. For the BLEU metric, our method obtains 0.58 performance gain compared with LoRA, with comparable results on the other metrics. We report results from the final epoch, whereas (Hu et al., 2021) presented the best performance observed during training, and use 4 GPUs rather than 1 due to time constraints, which may contribute to the observed variances w.r.t. the reported performance in (Hu et al., 2021). These results demonstrate that Quantum-PEFTcan achieve a comparable level of accuracy to the baselines while using significantly fewer parameters.

D.3. ViT CIFAR10 task

We evaluate a transfer learning task of the ViT model pre-trained on ImageNet-21k (Deng et al., 2009) towards CIFAR10 dataset (Krizhevsky et al., 2009). Detailed settings are found in Appendix D. The base model is frozen after being quantized

Table 6. Hyperparameter configurations for LoRA and Quantum-PEFT on the E2E benchmark for GPT2 Medium.

Hyperparameter	LoRA	Quantum-PEFT
# GPUs	4	4
Optimizer	AdamW	AdamW
Learning Rate Schedule	Linear	Linear
Weight Decay	0.01	0.01
Batch Size	8	8
Epochs	5	5
Warmup Steps	500	500
Label Smooth	0.1	0.1
Rank K	4	2 ($K' = 1$)
α	32	32
Learning Rate	0.0002	0.002
Unitary Parametrization	—	Q_{T} ($P = 3$)

Table 7. Results for different adaptation methods on the E2E benchmark and GPT2 Medium model. Quantum-PEFT achieves similar performance as LoRA with 4 times less trainable parameters.

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**
Quantum-PEFT	**0.098M**	**67.46**	**8.58**	45.02	67.36	**2.31**

with 3 bits, and adapters for query and value projections are updated. For Quantum-PEFT, we use Q_{P} parameterization for $K = L = 1$ with 2-split Hadamard product. Table 8 shows the comparison of full FT, LoRA, and Quantum-PEFT. When no fine-tuning was applied, the classification accuracy of the original ViT is poor, and thus fine-tuning is important. Compared to the full FT which requires 95.81M parameters, PEFT can significantly reduce the required number of trainable parameters, especially with our Quantum-PEFT. For example, Quantum-PEFT has 21-hold fewer parameters than LoRA with rank 4. More importantly, Quantum-PEFT shows superior performance despite the fact of the fewest parameters.

Table 9 shows the QAT performance with different number of bits per the Lie parameter for Taylor parameterization ($K = K' = 4$ and $P = 18$). Here, the base ViT model is not quantized, while only adapters are quantized. We use $g = 128$ and FP16 for scale and zero values β and μ. It is observed that reducing the precision for the Lie parameterization can gradually degrade. Nevertheless, thanks to QAT, no significant loss can be seen even with 1-bit integer quantization from FP32: i.e., 0.65% degradation. We also evaluate the performance of mixed-precision Taylor parameterization. One can see that adaptive bit loading can significantly improve the performance at few-bit quantization regimes. For instance, adaptive 1-bit quantization of Lie parameters has just 0.17% loss from FP32, and 0.28% improvement from uniform 1-bit quantization. This may come from the effective pruning gain. More details of model decomposition and quantization are found in Appendix C.2 and C.4.

Table 10 lists hyperparameters for the experiment on transfer learning task of ViT. The base ViT model (`google/vit-base-patch16-224`)[1] pretrained on ImageNet-21k has 12 layers of multi-head attention modules, each of which has 12 heads, 768 features, and a token length of 769. CIFAR10 is an image classification dataset having 10 classes of 32×32 colored images with 50k training samples and 10k test samples. We use up-sampling to 224×224 resolutions with random resized cropping and horizontal flip. The original classifier head has 1000 class output, and we selected 10 outputs based on the prediction score of CIFAR10 training data in prior to PEFT process. All weights and biases of the base ViT model including the classifier head are frozen after being quantized with 3-bit integers via rounding as described in Appendix C.4. Therefore, the base model is compressed from floating-point 32 bits to integer 3 bits (with auxiliary scale and zero values β and μ for $g = 128$ group), i.e., from 330MiB to 34MiB storage. It was confirmed that less than 3-bit

[1] `https://huggingface.co/google/vit-base-patch16-224`

Table 8. 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 9. Quantization impact on Lie parameters with Taylor parameterization for ViT transfer learning from ImageNet-21k to CIFAR10. Base ViT is not quantized.

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%**

quantization for the base ViT model compression had poor performance: 56.0% accuracy with 1 bit and 97.4% with 2 bits. The required run-time on GPU A40 40GB was about 3.37 second per iteration, and 5284.16 second per epoch.

Table 10. Hyperparameter configurations for LoRA and Quantum-PEFT on the CIFAR-10 transfer learning task for ViT.

Hyperparameter	LoRA	Quantum-PEFT
# GPUs	1	1
Optimizer	AdamW	AdamW
Learning Rate Schedule	Constant	Constant
Weight Decay	0.01	0.01
Batch Size	32	32
Epochs	100	100
Patience	5	5
Rank K	1,2,4	1, 4
Learning Rate	0.001	0.003
Unitary Parametrization	—	Q_{P} ($L = 1$), Q_{T} ($P = 18$)