

Scalable On-Hardware Training of Quantum Neural Networks: Application to Clinical Data Imputation

Natansh Mathur^{1,2}, Panagiotis Kl. Barkoutsos³, Masako Yamada³, Martin Roetteler³, Iordanis Kerenidis^{2,4}

¹ QC Ware Corp., Paris, France
³ IonQ Inc.

² IRIF, CNRS - Université Paris Cité, France
⁴ Quantum Signals

Abstract

Training quantum neural networks (QNNs) on quantum hardware is currently bottlenecked by the cost of gradient estimation: standard parameter-shift methods require a number of circuit evaluations that grows quadratically with the number of trainable parameters, making hardware-based optimisation impractical beyond small system sizes. In this work, we introduce a training framework that reduces this cost to logarithmic in the number of qubits. Our framework combines three co-designed ingredients: (i) a structured, subspace-preserving Butterfly circuit architecture with $O(n \log n)$ parameters and logarithmic depth; (ii) a layer-wise training strategy that confines on-hardware optimisation to one small, well-structured layer at a time; and (iii) a parallelised parameter-shift rule that exploits the commuting structure within each Butterfly layer to extract all gradients in a constant number of circuit executions. Together these reduce the number of distinct circuit evaluations per optimisation step from $O(n^2)$ to $O(\log n)$. We validate the framework on clinical data imputation using the MIMIC-III electronic health record dataset. Hybrid classical-quantum models are trained directly on IonQ Forte Enterprise trapped-ion hardware at 16 qubits without performance degradation relative to ideal or noisy simulation and via tensor-network simulation at 32 qubits, with 32-qubit inference executed on hardware. The resulting models match or exceed strong classical neural baselines in downstream patient survival prediction, demonstrating that the proposed framework enables practical, scalable QNN training under realistic hardware constraints.

Quantum Neural Network Architecture The QNN architecture is designed around three guiding principles: (i) controllable expressivity to mitigate barren plateaus, (ii) favourable parameter and depth scaling for hardware execution, and (iii) structural compatibility with efficient gradient evaluation. The model is organised as a modular pipeline consisting of a non-Gaussian state initialisation stage, a data-loading layer, and a parametrised, subspace-preserving quantum circuit arranged in a Butterfly architecture (Figure 1), allowing each component to be independently motivated and optimised while ensuring coherent end-to-end operation on hardware.

Layer-Wise Training We adopt a *layer-wise training* strategy that combines classical, simulated, and on-hardware optimisation, exploiting the fact that the Butterfly architecture has only $O(\log n)$ layers. Rather than optimising all parameters simultaneously, the model is trained progressively: two independent $n/2$ -qubit Butterfly subcircuits are trained first (classically or via quantum simulation), and their parameters are then frozen. A new coupling layer of $n/2$ two-qubit gates is added to join the two subcircuits into an n -qubit Butterfly circuit, and only this newly introduced layer is optimised, while all previously trained layers remain fixed.

Parallel Parameter-Shift Rule The Butterfly circuit naturally realises a commuting-block structure [BWP23, CRM⁺24]: each of its $\log(n)$ sub-layers consists of Reconfigurable Beam Splitter (RBS) gates acting on disjoint pairs of qubits, so the generators within a sub-layer mutually commute and each Butterfly layer constitutes a commuting block.

The results of [BWP23] therefore apply whenever the measurement Hamiltonian $\mathcal{H}$ commutes or anticommutes with the RBS generator $G_j \propto Y_{j_1} \otimes X_{j_2} - X_{j_1} \otimes Y_{j_2}$. For the global observable $Z^{\otimes n}$, $[Z_j^{\otimes 2}, G_j] = 0$, so this condition is satisfied and the parallel parameter-shift rule applies (see [CRM⁺24] for details).

This structural property allows us to apply the parameter-shift rule to all parameters in a Butterfly layer simultaneously. For an individual RBS gate with parameter θ_i , the parameter-shift rule is derived

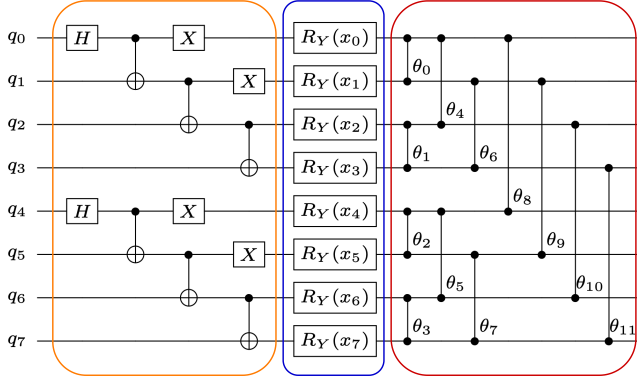


Figure 1: The quantum component of the hybrid imputation pipeline, shown for $n = 8$ qubits, has three modules. (i) *Non-Gaussian state initialisation*: groups of four qubits are prepared in the entangled state $|\phi\rangle = (|0011\rangle + |1100\rangle)/\sqrt{2}$ using the magic-state loader of [ODMZ22], applied in parallel across $n/4$ qubit groups, placing the register in a non-Gaussian, Hamming-weight-preserving state that is classically hard to simulate. (ii) *Data-loading layer*: each feature x_i is angle-encoded via $R_Y(2\pi x_i)$ on qubit i . (iii) *Trainable Butterfly circuit*: $O(\log n)$ layers of RBS gates with parameters θ mix information within the Hamming-weight-preserving subspace at $O(n \log n)$ total parameters. The quantum model serves as a learnable conditional estimator for one selected feature within an iterative-imputation step, while remaining features are handled classically.

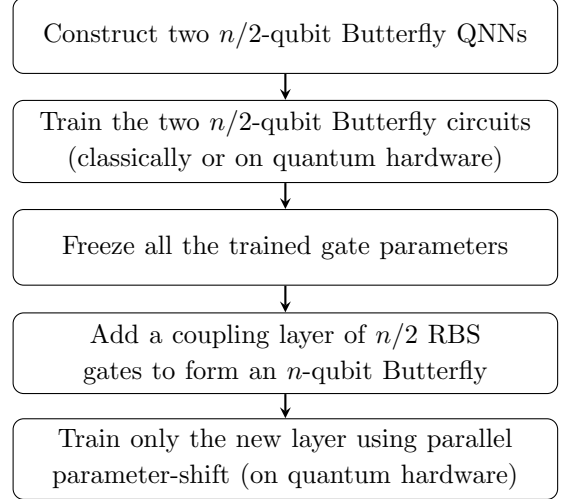


Figure 2: Layer-wise training workflow: smaller $n/2$ -qubit Butterfly subcircuits are trained first (classically or on hardware) and frozen. A coupling layer then forms an n -qubit circuit, and only this new layer is optimised on hardware using parallel parameter-shift, substantially reducing measurement and optimisation cost.

in [AWGP21]. Exploiting the commuting structure of the Butterfly layer, all parameters $\{\theta\}$ in the layer can be shifted simultaneously, yielding

$$\frac{\partial f(\theta)}{\partial \theta} = \left[f\left(\theta + \frac{\pi}{4}\right) - f\left(\theta - \frac{\pi}{4}\right) \right] - \frac{\sqrt{2}-1}{2} \left[f\left(\theta + \frac{\pi}{2}\right) - f\left(\theta - \frac{\pi}{2}\right) \right]$$

Individual gradients are recovered by diagonalising the corresponding commuting gradient observables.

Experiment Application: Medical Data Imputation We instantiate the training-efficient QNN framework on imputing missing values in clinical tabular data (Figure 3), a task that probes both the expressive capacity of QNNs and the practical viability of training them under realistic hardware constraints. Imputation quality is assessed indirectly via a clinically meaningful proxy task, binary patient survival prediction, following established practice of evaluating how imputation strategy affects downstream classifier performance, measured by the area under the ROC curve (AUC). This protocol enables a fair, application-relevant comparison between classical and hybrid classical-quantum models.

Our experiments use the Medical Information Mart for Intensive Care III (MIMIC-III) database [JPS⁺16], which contains de-identified electronic health records for ICU admissions.

Results The central result is a feasibility demonstration: gradient-based training of a 16-qubit QNN via the parallel parameter-shift rule is viable on current trapped-ion hardware. The dominant bottleneck preventing on-hardware gradient optimisation at this scale has been the $O(n^2)$ circuit evaluation cost of standard parameter-shift, which makes iterative optimisation on real devices prohibitively slow and noise-sensitive beyond modest qubit counts. Our framework reduces this cost to $O(\log n)$, and the experiments reported here confirm that this reduction is sufficient to make direct hardware training tractable at 16 qubits without requiring gradient pruning, zero-order approximations, or simulation fallback.

These results confirm that gradient-based training on IonQ Forte Enterprise, facilitated by the parallel parameter-shift rule, is not only feasible but yields models competitive with classical networks

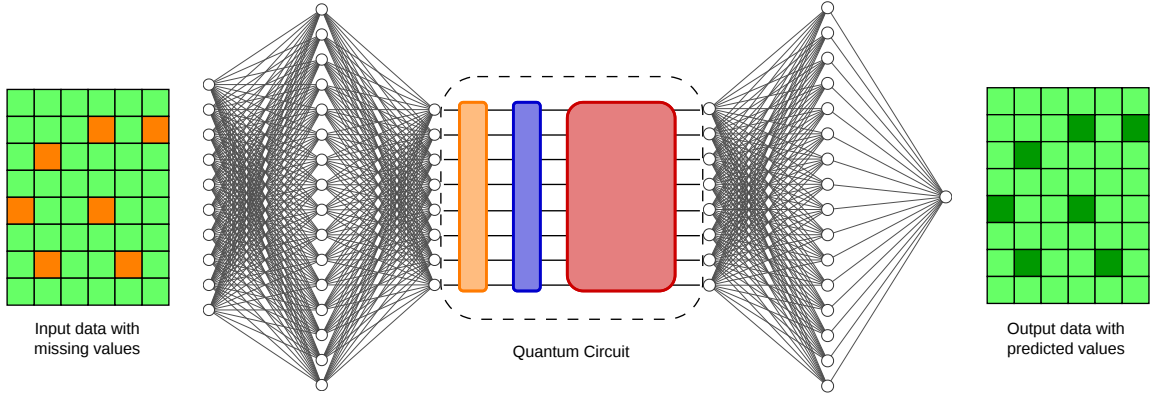


Figure 3: Overview of the hybrid classical-quantum imputation pipeline: observed features (green) are used as inputs to predict missing entries (orange); the quantum circuit components are described in Figure 1.

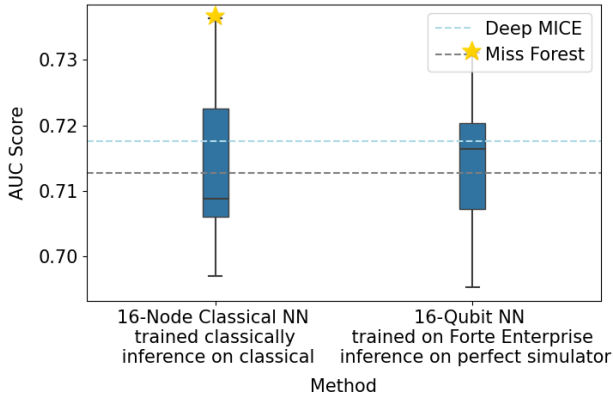


Figure 4: Downstream survival prediction AUC for a classical 16×16 neural network versus a hybrid classical-quantum network (16-Qubit Hybrid DeepImputer) with its quantum layer trained directly on IonQ Forte Enterprise. Both share the same architecture, with inference on a classical simulator in both cases. The hybrid model achieves a higher median AUC with reduced variance across seeds.

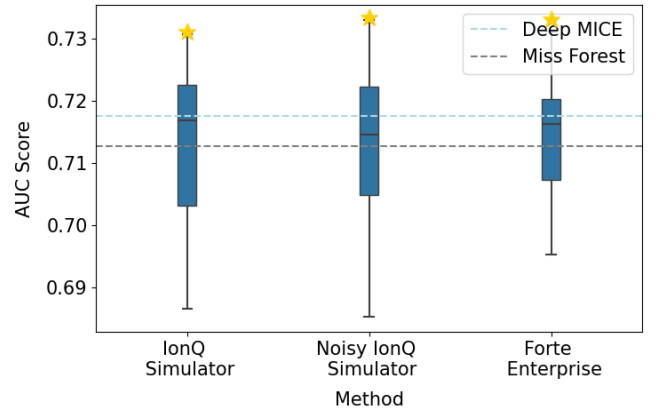


Figure 5: 16-Qubit hybrid model performance when the quantum layer is trained via ideal simulation, noisy simulation, and direct execution on IonQ Forte Enterprise. No statistically significant degradation is observed moving from simulation to hardware; inference uses the classical simulator throughout.

of comparable width. Indeed, the hardware-trained model achieves a marginally higher median AUC (+1%) with lower variance (-1.6% in standard deviation) relative to the classical reference, suggesting a mild stabilising effect of on-hardware optimisation at this scale. Figure 5 compares the downstream survival prediction AUC of models trained on an ideal simulator, noisy simulator, and hardware execution on IonQ Forte Enterprise. No statistically significant degradation is observed across these three training conditions, confirming that the $O(\log n)$ training framework is robust to realistic hardware noise at this scale.

Conclusion This work introduces a hardware-compatible training framework for gradient-based optimisation of quantum neural networks on near-term processors. By combining a structured, logarithmic-depth Butterfly architecture with layer-wise training and a parallelised parameter-shift rule that exploits the commuting structure within each layer, the number of distinct circuit evaluations per optimisation step scales as $O(\log n)$ in the number of qubits — down from $O(n^2)$ for generic architectures — and remains constant regardless of layer width. This reduction constitutes the primary contribution of the framework and is the enabling condition for the hardware training results reported here.

References

- [AWGP21] Gian-Luca R Anselmetti, David Wierichs, Christian Gogolin, and Robert M Parrish. Local, expressive, quantum-number-preserving vqe ansätze for fermionic systems. *New Journal of Physics*, 23(11):113010, 2021.
- [BWP23] Joseph Bowles, David Wierichs, and Chae-Yeun Park. Backpropagation scaling in parameterised quantum circuits. *arXiv preprint arXiv:2306.14962*, 2023.
- [CRM⁺24] Brian Coyle, Snehal Raj, Natansh Mathur, El Amine Cherrat, Nishant Jain, Skander Kazdaghi, and Iordanis Kerenidis. Training-efficient density quantum machine learning. *arXiv preprint arXiv:2405.20237*, 2024.
- [JPS⁺16] Alistair EW Johnson, Tom J Pollard, Lu Shen, Li-wei H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3(1):1–9, 2016.
- [ODMZ22] Michał Oszmaniec, Ninnat Dangniam, Mauro ES Morales, and Zoltán Zimborás. Fermion sampling: a robust quantum computational advantage scheme using fermionic linear optics and magic input states. *PRX Quantum*, 3(2):020328, 2022.