

Deep Learning Approaches to Quantum Error Mitigation

Leonardo Placidi^{1,4,5}, Ifan Williams², Enrico Rinaldi^{3,6}, Daniel Mills², Cristina Cîrstoiu², Vanya Eccles², and Ross Duncan²

¹Quantinuum K.K., Tokyo, Japan; ²Quantinuum Ltd., Cambridge, United Kingdom; ³Quantinuum Ltd., London, United Kingdom; ⁴The University of Osaka, Japan; ⁵QIQB, The University of Osaka, Japan; ⁶RIKEN Center for Quantum Computing, Japan

Abstract

Quantum error mitigation (QEM) is a near-term route to reducing bias in estimates extracted from noisy quantum processing units (QPUs). We present a systematic study of *machine learning (ML)-guided QEM*: deep learning models that mitigate full output probability distributions from measured quantum circuits. The task is deliberately stringent: an improved computational-basis distribution can be used to estimate any diagonal observable, but it requires the model to learn correlations across all bitstrings rather than a single expectation value. We compare fully connected, recurrent, Transformer, noisy-encoder, and Perceiver-style architectures; contrast direct prediction models with multiplicative correction models; and test pre-training, fine-tuning, ablation, and transfer-learning workflows. On noisy simulations parameterised by IBM calibration data and on real data from IBM superconducting QPUs, attention-based architectures – especially Transformer and Perceiver variants – are the most reliable learned models; they are competitive with strong analytic baselines on simulated data, outperform the tested post-processing baselines on real hardware data, and transfer effectively between comparable devices.

Motivation and contribution

This work [1] falls under the research topic of ML-guided QEM: the goal is to reduce noise bias in quantities estimated from noisy samples without requiring full quantum error correction [2, 3]. Many established QEM methods rely on fixed noise assumptions, additional circuit families, or a preselected observable. Here we study a complementary data-driven setting where the mitigation map is learned from data: a neural network receives a circuit description, backend calibration data, and the measured noisy probability distribution, and outputs a mitigated distribution closer to the ideal one. This formulation is more general than mitigating a single observable, but also a harder benchmark for ML because finite-shot noise, circuit structure, calibration drift, and device-specific errors all affect the target distribution.

The study addresses four practical questions for learned QEM: which architectures best exploit the circuit, backend, and noisy-distribution inputs; whether noisy simulations can pre-train models before fine-tuning on scarce hardware data; which inputs drive performance; and whether learned mitigation generalises across circuit families or transfers between similar QPUs. We answer using a large benchmark covering simulated and real hardware data.

Dataset and supervised task

We use two five-qubit circuit families generated up to a family-specific maximum time step T : *Pauli* circuits append random Pauli gadgets per time step, while *Random* circuits append random IBM-native gates. Circuits are compiled with TKET [4] to the native connectivity and gate set of fixed five-qubit subgraphs of `ibm_algiers` and `ibm_hanoi`, with no further optimisation; the resulting circuits span $\sim 20 - 140$ layers.

Let n_ℓ be the number of post-compilation layers. Each circuit is encoded as $C_{\text{array}} \in \mathbb{R}^{n_\ell \times 5 \times 5}$, where each layer is a qubit-by-operation matrix: rows index active qubits, and columns encode X , SX , R_z , CX , and measurement operations, with idle entries zero and R_z/CX entries carrying angle/control-target information. Since layers may contain parallel operations, n_ℓ depends on both T and circuit family. To form fixed-size samples, C_{array} is zero-padded along the layer axis to $C_{\text{pad}} \in \mathbb{R}^{144 \times 5 \times 5}$; $C \in \mathbb{R}^{3600}$ when flattened.

The remaining model inputs are backend calibration data $B \in \mathbb{R}^{101}$ and the measured distribution $P_{\text{noisy}} \in \mathbb{R}^{32}$ over five-qubit bitstrings. Multilayer perceptron (MLP) models use the flattened circuit-backend vector $x_{CB} \in \mathbb{R}^{3701}$. Sequence and attention models instead use an embedded layer-wise circuit-backend sequence

$X_{CB} \in \mathbb{R}^{n_c \times 132}$. The supervised label is $P_{\text{ideal}} \in \mathbb{R}^{32}$, obtained by classical simulation; the model output is the mitigated distribution $P_{\text{mit}} \in \mathbb{R}^{32}$. The workflow is summarised in Fig. 1.

The benchmark contains 246,000 unique circuits. Real-device data were collected over several months with repeated executions and 20,000 shots per circuit. Simulated data are generated from noisy classical simulations parameterised by sampled `ibm_algiers` calibration data. This design exposes the models to temporal drift and hardware variability while retaining exact labels for evaluation. Models are trained with KL-divergence, $\text{KL}(P_{\text{ideal}}, P_{\text{mit}})$, and evaluated using the *L1 Relative Change*

$$\mathcal{R}(P_{\text{mit}}, P_{\text{ideal}}, P_{\text{noisy}}) := \frac{\|P_{\text{ideal}} - P_{\text{mit}}\|_1 - \|P_{\text{ideal}} - P_{\text{noisy}}\|_1}{\|P_{\text{ideal}} - P_{\text{noisy}}\|_1}. \quad (1)$$

Negative $\mathcal{R}$ indicates successful mitigation, and we also report the fraction of test circuits with $\mathcal{R} < 0$.

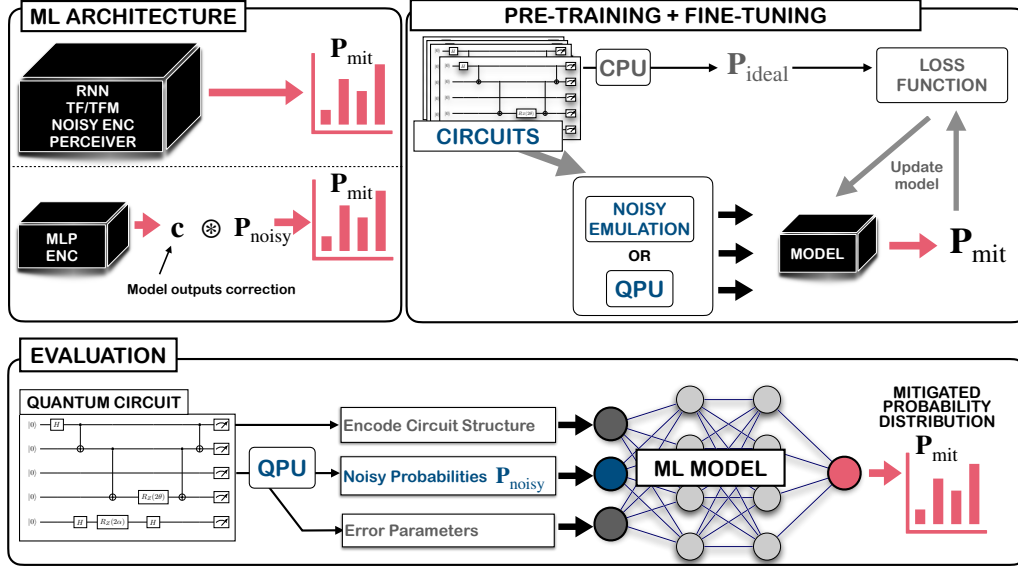


Figure 1: Training and inference pipeline. Prediction models output P_{mit} directly; correction models output c and combine it with P_{noisy} . Models are pre-trained on noisy simulations, fine-tuned on hardware data, and evaluated by ingesting circuit and backend features, and P_{noisy} .

Models and training workflow

We compare two output formulations. Prediction models condition directly on the measured distribution, learning $P_{\text{mit}} = f(C, B, P_{\text{noisy}})$. Correction models instead learn a correction vector $c = f(C, B)$ from circuit and backend features only, and form the mitigated distribution afterwards as $P_{\text{mit}} = \text{softmax}(c \odot P_{\text{noisy}})$; thus P_{noisy} enters the deterministic output layer and loss, but not the learned correction map. The model suite includes MLPs, recurrent neural networks, Transformer encoder-decoders, a Transformer variant augmented with SPAM-mitigated probabilities, a noisy-distribution-only attention encoder, and a Perceiver-style latent-attention model [5–7]. In particular, the Perceiver compresses heterogeneous circuit, backend, and probability-distribution inputs into a fixed latent array using cross-attention, followed by latent self-attention and a probability output head.

We optimise per-dataset, per-architecture hyperparameters by Bayesian search. We compare simulation-to-hardware transfer, hardware-only training, and simulation pre-training plus hardware fine-tuning; results used PyTorch Lightning [8] distributed data-parallel training on NVIDIA A100 GPUs.

Main results

Prediction models tend to outperform correction models, and attention-based models are the most reliable across datasets. On simulated data, the Transformer improves 94.2% of Pauli circuits and the Perceiver

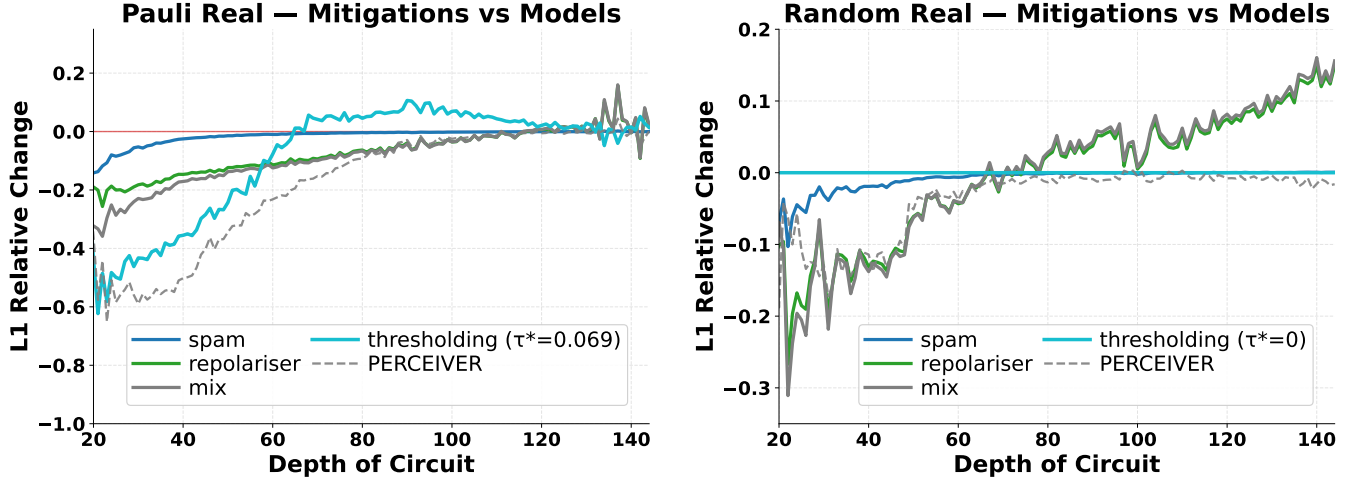


Figure 2: Median L_1 Relative Change versus circuit depth on real `ibm_algiers` Pauli-gadget and Random data. PERCEIVER is compared with SPAM correction, Repolarizer, SPAM+Repolarizer (Mix), and optimised global thresholding. Negative values indicate improvement.

improves 91.7% of Random circuits. On real `ibm_algiers` data, the Perceiver gives the best median performance in both circuit families: $\mathcal{R} = -0.2062$ with 83.2% of Pauli circuits improved, and $\mathcal{R} = -0.0210$ with 64.4% of Random circuits improved. The weaker absolute performance on Random circuits is consistent with their smaller signal: at larger depths, the noisy distribution approaches a near-uniform distribution, leaving little structure for any post-processing method to exploit.

Figure 2 compares the Perceiver with SPAM correction, Repolarizer, SPAM+Repolarizer (Mix), and optimised thresholding on real `ibm_algiers` data. On simulated data, analytical methods remain competitive, with Repolarizer and Mix giving the strongest median improvements, consistent with the effectiveness of simple global-depolarisation models in this setting. On real hardware, especially for Pauli circuits, the advantage shifts to learned models: for real Pauli data, the Perceiver improves over optimised thresholding in both median $\mathcal{R}$ and fraction of improved circuits (-0.2062 , 83.2%, versus -0.1675 , 68.2%). For real Random data, the improvement over SPAM correction is smaller but consistent (-0.0210 , 64.4%, versus -0.0008 , 55.5%).

Ablations show that P_{noisy} is the dominant input: randomising it severely degrades performance, while randomising circuit descriptions has little effect for several recurrent and Transformer models, indicating that the networks are not simply simulating the ideal circuit. Circuit and backend features act as secondary, architecture-dependent conditioning. Cross-family testing is limited, but transfer between `ibm_algiers` and `ibm_hanoi` works well and remains stable after fine-tuning for attention-based models, suggesting learned QEM can transfer across comparable devices without full retraining.

Outlook

This benchmark study identifies useful design principles for ML-guided QEM: use P_{noisy} as the primary input, exploit attention mechanisms for heterogeneous data, pre-train on simulation when possible, and evaluate robustness through ablations and transfer tests. The main limitation is scalability: full output distributions and exact ideal labels are available only for small systems. Larger experiments will require marginals, local observables, tensor-network- or Clifford-assisted labels, mirror-benchmarking targets, or bitstring generative models that assign probabilities to sampled bitstrings without storing the full 2^n -outcome distribution. Future work should also compare simulators of varying fidelity, using simulator–hardware gaps to diagnose missing noise mechanisms and guide improved device error models. Overall, attention-based neural architectures learn useful, transferable post-processing maps for QEM on real QPU data.

References

- [1] Leonardo Placidi, Ifan Williams, Enrico Rinaldi, Daniel Mills, Cristina Cîrstoiu, Vanya Eccles, and Ross Duncan. Deep Learning Approaches to Quantum Error Mitigation, 1 2026. URL <https://arxiv.org/abs/2601.14226>.
- [2] Zhenyu Cai, Ryan Babbush, Simon C Benjamin, Suguru Endo, William J Huggins, Ying Li, Jarrod R McClean, and Thomas E O’Brien. Quantum error mitigation. *Reviews of Modern Physics*, 95(4):045005, 2023.
- [3] Youngseok Kim, Andrew Eddins, Sajant Anand, Ken Xuan Wei, Ewout Van Den Berg, Sami Rosenblatt, Hasan Nayfeh, Yantao Wu, Michael Zaletel, Kristan Temme, et al. Evidence for the utility of quantum computing before fault tolerance. *Nature*, 618(7965):500–505, 2023.
- [4] Seyon Sivarajah, Silas Dilkes, Alexander Cowtan, Will Simmons, Alec Edgington, and Ross Duncan. tket: a retargetable compiler for nisd devices. *Quantum Science and Technology*, 6(1):014003, 2020.
- [5] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.
- [6] Andrew Jaegle, Felix Gimeno, Andrew Brock, Andrew Zisserman, Oriol Vinyals, and João Carreira. Perceiver: General perception with iterative attention. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, pages 5494–5503. PMLR, 2021. URL <https://proceedings.mlr.press/v139/jaegle21a.html>.
- [7] Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Daniel Zoran, Andrew Brock, Evan Shelhamer, Olivier Hénaff, Matthew M. Botvinick, Andrew Zisserman, Oriol Vinyals, and João Carreira. Perceiver io: A general architecture for structured inputs & outputs, 2022. URL <https://arxiv.org/abs/2107.14795>.
- [8] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf.