Frobenius Normalization Enables Stable Training for Quantum State Denoising

Amitav Krishna

***Abstract*— Quantum computers offer asymptotic speedups for problems like molecular simulation and integer factorization, but noise limits their near-term utility. Quantum Error Correction can address this, but the number of extra qubits required makes it infeasible for near-term use while the cost per qubit remains high. Quantum Error Mitigation is a near-term alternative, but only recovers expectation values, not the full quantum state needed for extracting accurate results from simulations run on quantum hardware. For these, we need quantum state reconstruction. Quantum state tomography estimates a state by measuring the system across many bases, but these measurements are inherently noisy; quantum state reconstruction recovers a clean density matrix from this noisy data. Neural networks have shown promise here, but previous work has only been demonstrated up to 5 qubits. We find that Frobenius normalization (scaling inputs to unit purity) removes this bottleneck, enabling scaling to 8-qubit systems while maintaining a large improvement in fidelity.**

I. Introduction

Quantum computers hold the potential to enable exponential speedups of many tasks through their exploitation of the principles of quantum mechanics [1], the ability to explore possibilities in superposition, with paths that interfere to amplify correct answers and cancel wrong ones. However, this interference is both a blessing and a curse. After enough time, the computer will inevitably become fully entangled with the environment, locking the information in correlations with the rest of the universe and throwing away the key, leaving us with nothing to work with.

Quantum Error Correction is a principled, scalable method for achieving low error rates however requires many redundant physical qubits, making it infeasible in the near-term. Quantum Error Mitigation is another method for reducing noise of quantum measurements, in the near term, while qubits remain expensive, using classical computation, however it acts only on expectation values, making it less useful for tasks such as verifying the configuration of a specific quantum computer or extracting detailed simulation results, that require access to the full quantum state. Quantum State Tomography is the solution to this, taking in many measurements of a quantum system and returning a probable state for that system, however these predictions are often degraded by various types of noise.

Correspondence: krishna@amitav.net

Quantum State Reconstruction addresses this gap by learning to reconstruct the true state given a noisy tomographic estimate. Recent work has shown that neural networks can learn to invert noise channels [2, 3, 4]. However, previous neural network approaches have only been demonstrated up to 5 qubits. We introduce a data normalization method that enables scaling to 8-qubit systems in our experiments.

We generate synthetic datasets by simulating random quantum circuits composed of single-qubit and two-qubit entangling gates. Each circuit is simulated twice: once exactly to produce a clean density matrix, and once with noise channels (depolarizing, amplitude damping, phase damping, bit flip) applied after each layer to produce a noisy counterpart. We train neural networks to map noisy inputs to clean targets, using Frobenius fidelity as the loss function. The key preprocessing step is Frobenius normalization: we scale each density matrix to unit norm before training. The Frobenius norm of a density matrix equals the square root of its purity. Since noisy states have norms up to ~16 times smaller than clean states for 8-qubit systems (a physical consequence of decoherence pushing states toward the maximally mixed state), this normalization decouples scale recovery from structural denoising, allowing the model to focus on restoring coherence structure rather than learning a global rescaling.

On 5-qubit systems, normalization reduces validation loss by 25% and enables both MLP and Transformer architectures to achieve >3 times improvement in Uhlmann fidelity over the noisy baseline. On 8-qubit systems (256×256 density matrices), training without normalization fails entirely, while normalized models achieve 3–4 times improvement over baseline. Our results suggest that for neural QSR, input conditioning matters more than architectural choice, as both MLPs and Transformers achieve similar performance once normalized. The scale mismatch caused by decoherence is a domain-specific challenge that requires domain-specific preprocessing, and we recommend Frobenius normalization as a standard step for any neural approach to quantum state reconstruction.

II. Related Work

Input normalization is well-established in deep learning. Batch normalization [5] and layer normalization [6] address internal covariate shift, while input standardization (zero mean, unit variance) is standard practice for feedforward networks. However, quantum density matrices present a unique challenge: the scale difference between noisy and clean states is not statistical variation but a *physical consequence* of

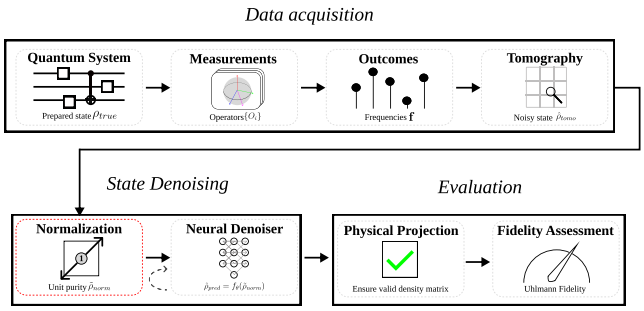


Fig. 1: Overview of the training and evaluation pipeline. Quantum states are measured, reconstructed via tomography, and normalized to unit Frobenius norm. A neural denoiser maps the normalized noisy state to a denoised estimate, which is then projected to a valid density matrix before computing Uhlmann fidelity.

decoherence. Our work identifies Frobenius normalization as the appropriate preprocessing for this domain, demonstrating that it is essential for stable training at larger scales.

For computational tractability at larger qubit counts, we adopt patch-based tokenization inspired by Vision Transformers (ViT) [7] and hierarchical variants like the Swin Transformer [8]. We maintain a fixed token count (64) by adjusting patch size with matrix dimension, enabling scaling to 8-qubit systems where element-wise processing is infeasible.

Machine learning has been applied to quantum information theory, including for syndrome decoding in quantum error correction [9]. Recent work by Lin et al. [10] has shown success in using convolutional neural networks for classical read-out error mitigation. In addition, machine learning has been applied to a related problem to quantum state reconstruction, the quantum marginal problem, with success at 8 qubits [11].

Data-driven approaches to quantum state reconstruction have been explored for small numbers of qubits. Bondarenko and Feldmann [12] used quantum autoencoders to denoise GHZ states up to 4 qubits. Morgillo et al. [2] use a multilayer perceptron to denoise systems of up to 3 qubits, while Kendre [3] instead uses a convolutional architecture to scale up to 5 qubits. Another notable work here is Wang [13], in which they do mixed state to mixed state denoising, as opposed to the mixed state to pure state denoising that is most common, and that our own work covers. Finally, another quantum state reconstruction work (though with a more tomographic framing) is Lohani et al. [14], notable for their use of Cholesky output factors to enforce physicality of the outputs. This Cholesky-output approach was later carried into attention-based architectures by Palmieri et al. [15]. Previous density matrix to density matrix reconstruction has only been demonstrated up to 5 qubits.

III. Methods

To investigate the effectiveness of Frobenius normalization for improving neural state denoisers, we generate two datasets consisting of temporally-independent measurement probabilities and coherence values derived from quantum circuits. We generate 100,000 samples for each dataset, each a pair of density matrices from the same circuit under ideal and noisy evolution, which we denote ρ_{true} and $\hat{\rho}_{\text{tomo}}$, respectively. Each circuit has a depth uniformly sampled from 6 to 9 (inclusive), and each layer consists of applying an independently uniformly sampled single-qubit gate to every qubit, as well as random two-qubit gates applied to each of $\lfloor n/2 \rfloor$ randomly chosen adjacent pairs.

We consider four standard single-qubit Markovian channels: depolarizing, amplitude damping, phase damping, and bit flip, each applied independently to each circuit layer with strength p. Additionally, we have a mixed channel consisting of those four channels applied sequentially, each with strength $p/4$. We generate datasets for four noise intensities $p \in \{0.05, 0.10, 0.15, 0.20\}$. The result is a dataset stratified into 20 "noise cells" (5 types $\times$ 4 levels), with equal representation. For both the 5 qubit and 8 qubit datasets, we use 100,000 samples. For both datasets we use an 80/10/10 split for training, validation, and testing stratified by noise cell to ensure balanced evaluation across all noise types and intensities. We normalize to unit Frobenius norm, a standard measure of matrix scale, prior to training.

$$\tilde{\rho}_{\text{norm}} = \frac{\hat{\rho}_{\text{tomo}}}{\|\hat{\rho}_{\text{tomo}}\|_F}, \quad \|\rho\|_F = \sqrt{\sum_{ij} |\rho_{ij}|^2}. \tag{1}$$

Because density matrices are Hermitian ($\rho^\dagger = \rho$), the Frobenius norm reduces to the square root of the purity: $\|\rho\|_F = \sqrt{\text{Tr}(\rho^\dagger \rho)} = \sqrt{\text{Tr}(\rho^2)}$. Since our clean target states are always pure, Frobenius normalization leaves them unchanged ($\|\rho_{\text{true}}\|_F = 1$ already). Normalization only affects the noisy inputs, whose Frobenius norms are suppressed by decoherence. The maximally mixed state sets the lower bound at $2^{-n/2}$, which is ~ 0.18 for 5 qubit states and ~ 0.06 for 8 qubit states, showing that the noisy norm shrinks with the system size. This normalization discards purity information, which is acceptable since the purity of the clean state is known *a priori*. The practical effect is to bring all inputs to unit scale, which we demonstrate is critical to stable optimization.

For training, we minimize $1 - F_F$, where F_F is the normalized Frobenius fidelity:

$$F_F(\rho, \sigma) = \frac{\text{Tr}(\rho\sigma)}{\sqrt{\text{Tr}(\rho^2)\text{Tr}(\sigma^2)}}. \tag{2}$$

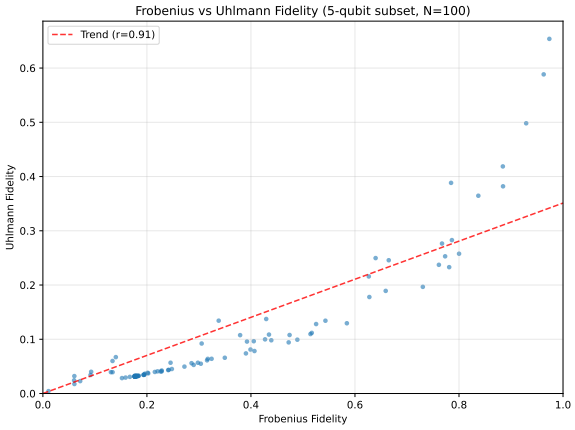


Fig. 2: Correlation between Frobenius fidelity (normalized) and Uhlmann fidelity (physical) on a random subset of the 5-qubit dataset ($r = 0.91$).

For evaluation, we use the Uhlmann fidelity:

$$F_U(\rho, \sigma) = \left(\mathrm{Tr} \sqrt{\sqrt{\rho}\, \sigma\, \sqrt{\rho}} \right)^2 . \tag{3}$$

The Frobenius fidelity includes coherence terms algebraically, but it does not model their physical significance, whereas the Uhlmann fidelity does. On the other hand, Uhlmann fidelity requires multiple eigendecompositions, making it expensive to run for training, so we run it only for evaluation. To validate Frobenius fidelity as a training proxy, we analyzed its correlation with the ground-truth Uhlmann fidelity on 100 randomly sampled 5-qubit density matrices (Figure 2), finding a positive association, implying that reductions to the Frobenius loss are, in general, also reductions in Uhlmann fidelity, which we find sufficient to justify the Frobenius fidelity as a training surrogate.

To understand why normalization helps, consider a neural network f_θ trained to minimize a loss $\mathcal{L}(f_\theta(x), y)$. Our training loss F_F is scale-invariant (it is a cosine similarity), so the loss surface is identical regardless of input magnitude. The benefit of normalization is therefore not about what the loss measures, but about how the optimizer navigates that surface.

Consider the first layer of f_θ, which computes $h = W^{(1)}x + b^{(1)}$. The gradient of the loss with respect to these weights is:

$$\frac{\partial \mathcal{L}}{\partial W^{(1)}} = \frac{\partial \mathcal{L}}{\partial h} \cdot x^T . \tag{4}$$

This gradient is *linearly proportional* to the input x. When inputs are noisy density matrices with $\|x\|_F \approx 0.06$ (8-qubit systems), the gradient magnitude $\|\partial \mathcal{L}/\partial W^{(1)}\|$ is suppressed by the same factor relative to unit-scale inputs. With a fixed learning rate η, the effective parameter update $\Delta W^{(1)} = -\eta\, \partial \mathcal{L}/\partial W^{(1)}$ is ~16× smaller than it would be for normalized inputs. This is equivalent to an implicit ~16× reduction in learning rate for the first layer.

The problem is compounded by weight initialization. Our models use PyTorch's default Kaiming uniform initialization, calibrated to maintain unit-scale activations and gradients under the assumption that inputs have roughly unit variance. When inputs are 16× smaller, the initialization places the network in a regime it was not designed for: activations are suppressed from the first layer onward, and the gradient signal attenuates further as it propagates through subsequent layers.

This effect worsens exponentially with system size. The Frobenius norm of the maximally mixed state is $1/\sqrt{2^n}$, so gradient suppression scales as $O(2^{-n/2})$. This explains why normalization is merely helpful at 5 qubits (~5 times suppression) but essential at 8 qubits (~16 times suppression). By normalizing inputs to unit Frobenius norm, we restore gradient magnitudes and initialization assumptions to their intended operating regime.

Note that Frobenius-normalized matrices are not physically valid density matrices as their trace is not equal to one. We use them only as an intermediate representation during training. At evaluation time, we project the denoiser output $\hat{\rho}_{\text{pred}} = f_\theta(\tilde{\rho}_{\text{norm}})$ back to a valid density matrix by Hermitianizing, normalizing the trace, and clamping negative eigenvalues:

$$\frac{\hat{\rho}_{\text{pred}} + \hat{\rho}_{\text{pred}}^\dagger}{2} \to \rho_{\text{herm}}, \tag{5}$$

$$\frac{\rho_{\text{herm}}}{\mathrm{Tr}(\rho_{\text{herm}})} \to \rho_{\text{proj}}, \tag{6}$$

$$\max(0,\, \lambda_i) \to \lambda_i, \quad V\, \mathrm{diag}(\lambda_{\text{clamped}})\, V^\dagger \to \rho_{\text{psd}}, \tag{7}$$

where λ_i and V are the eigenvalues and eigenvectors of ρ_{proj}, respectively. Note that clamping can slightly reduce the trace (by the magnitude of the clamped eigenvalues, typically 10^{-7}); we do not re-normalize afterward, as the effect on fidelity is negligible. This projection prevents evaluation from returning nonsensical values, such as fidelities greater than one.

Both architectures share a common patch tokenization scheme. The input density matrix is divided into non-overlapping patches, each embedded via a linear projection into 64 tokens of dimension 128 (5-qubit: 4×4 patches; 8-qubit: 32×32 patches). Each decoder token is projected back to its corresponding patch via a linear layer, then patches are reassembled into the full matrix.

The Transformer uses an encoder-decoder structure: 4 encoder layers and 4 decoder layers, each with 8 attention heads and FFN dimension 512. The decoder uses cross-attention to the encoder. The MLP uses an MLP-Mixer-style architecture [16]: token mixing (hidden dim 384) that mixes across patches, and channel mixing (hidden dim 320) applied per-patch. Parameter counts are matched: 1.09 M (5-qubit) and 1.61 M (8-qubit), with the difference coming from patch embedding/unembedding layers that scale with patch size.

To test whether larger models benefit more from normalization, we trained two additional MLP variants with identical optimization hyperparameters. The "Wide" variant scales hidden dimensions by 4 times (token hidden 1536, channel hidden 1280), yielding 5.29 M parameters, nearly five times the baseline. The "Deep" variant doubles the

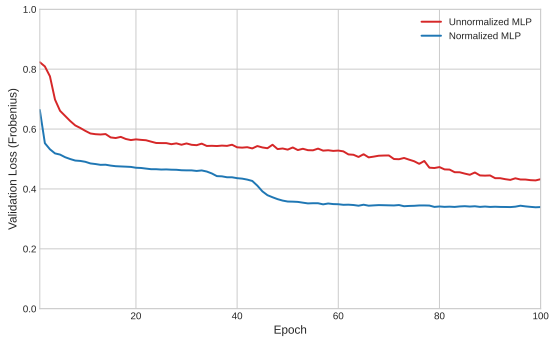


Fig. 3: Validation loss curves for the 5-qubit MLP with and without Frobenius normalization. Normalization enables faster convergence and a 25% lower final loss.

layer count (8+8 encoder-decoder blocks), yielding $2.15\,\mathrm{M}$ parameters.

All models are trained with AdamW (lr 3×10^{-4}, weight decay 10^{-5}), batch size 256, for up to 100 epochs with early stopping (patience 15 on validation loss). The training loss is $1 - F_F$ (Frobenius fidelity).

IV. Results and Discussion

Our central finding is that Frobenius normalization improves training stability and final model performance. Figure 3 shows validation loss with and without normalization on the 5-qubit dataset.

Figure 3 illustrates the training dynamics: the normalized MLP converges faster and to a lower loss, while the unnormalized MLP plateaus early and exhibits greater instability.

Without normalization, the small input magnitudes (~0.18 for 5 qubits, ~0.06 for 8 qubits) suppress gradient flow through early layers, as described in the Methods section. This gradient suppression prevents the model from learning fine-grained coherence recovery. With Frobenius normalization, both architectures improve over the noisy baseline:

TABLE I: 5-qubit model performance with Frobenius normalization. Both models use identical patch tokenization (4×4 patches, 64 tokens) and matched parameter counts.

Model	Params	Uhlmann Fidelity	vs Baseline
Baseline (noisy input)	-	0.167	1.00 times
MLP (hierarchical)	1.09M	0.516	3.09 times
Transformer (hierarchical)	1.09M	0.525	3.14 times

Comparing to the unnormalized MLP (0.463 Uhlmann fidelity), normalization improved the MLP from 0.463 to 0.516 (+11%) and the Transformer from 0.420 to 0.525 (+25%). The Transformer's larger gain likely reflects a conflict between its internal LayerNorm layers, which assume a consistent input scale across the token sequence, and the variable magnitudes of the unnormalized data. Both architectures achieve similar performance with normalization, confirming that input conditioning and not architectural choice is the critical factor for this task.

To confirm that the performance ceiling without normalization is an optimization problem rather than a capacity limitation, we trained wider (4× hidden dims, 5.29M params) and deeper (2× layers, 2.15M params) MLP variants under identical unnormalized conditions:

TABLE II: Capacity ablations on unnormalized 5-qubit data. Larger models perform worse, confirming an optimization pathology.

Model	Params	Uhlmann Fidelity	vs Baseline
MLP (matched)	1.09M	0.463	2.77 times
MLP (wide)	5.29M	0.310	1.86 times
MLP (deep)	2.15M	0.407	2.44 times

Both larger models perform worse, with the wide MLP dropping 33% (0.310 vs 0.463) and the deep MLP dropping 12% (0.407 vs 0.463). If the performance ceiling were due to insufficient capacity, adding parameters should help, not hurt. The fact that it hurts confirms that the bottleneck is optimization, not capacity.

Adding normalization to these same models confirms this diagnosis selectively: the matched MLP improves from 0.463 to 0.516 (+11%), but the wide MLP reaches only 0.344 and the deep MLP only 0.349—both far below the matched model. Normalization resolves optimization pathologies that arise from small input magnitudes (Section 3), but it cannot resolve optimization pathologies from other sources.

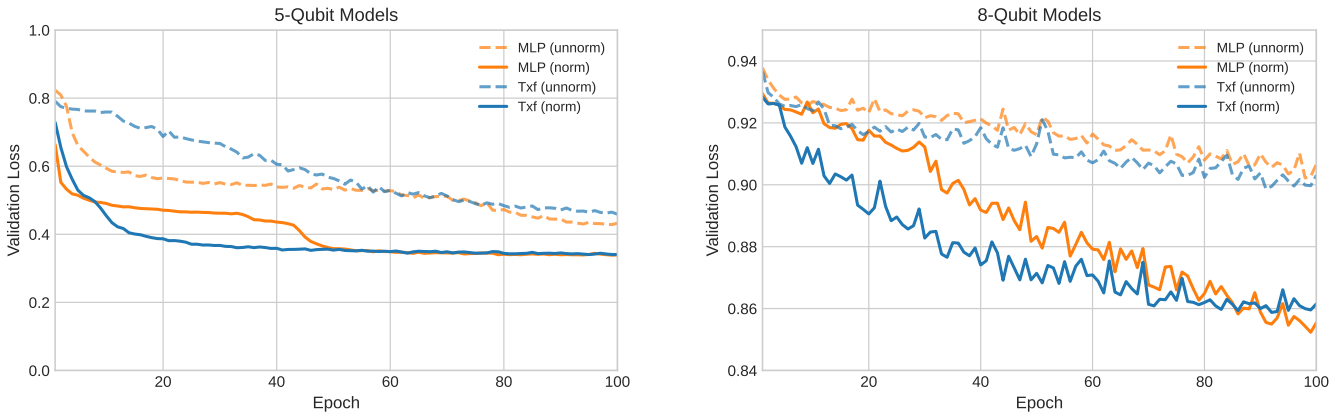
The strongest demonstration of normalization's importance is at 8 qubits. Previous attempts to train on 8-qubit density matrices (256×256, 65,536 complex elements) without normalization failed—models produced outputs no better than the noisy input. With normalization, both architectures improve over baseline:

TABLE III: 8-qubit scaling results with Frobenius normalization. Despite the exponentially harder task (baseline 0.006), both models achieve more than 3 times improvement.

Model	Uhlmann Fidelity	vs Baseline
Baseline (noisy)	0.00635	1.00 times
MLP (hierarchical)	0.02385	3.76 times
Transformer (hierarchical)	0.01901	3.00 times

The 8-qubit task is exponentially harder: the baseline fidelity drops from 0.167 to 0.006, and the density matrix has 64 times more elements. Yet with normalization, the MLP achieves 3.76 times improvement over baseline—comparable relative improvement to the 5-qubit case. Unnormalized 8-qubit models failed to converge (Figure 4b), so we do not report Uhlmann fidelity for these models.

Figure 4 shows validation loss for all four model/qubit combinations with and without normalization. The 5-qubit normalized curves show rapid early improvement followed by gradual refinement, while their unnormalized counterparts converge more slowly to a higher loss. The 8-qubit contrast is starker: although all four 8-qubit curves are noisy, by epoch 30 the normalized models have overtaken the unnormalized models, which plateau near 0.91 while normalized models steadily decrease to ~0.85. To verify that stable convergence is not seed-dependent, we retrained 5-qubit models with two additional initialization seeds (100 and 200) while keeping the data split fixed (seed=42). All runs converged to similar validation loss (0.32–0.35), confirming reproducibility.



(a) 5-qubit models

(b) 8-qubit models (y-axis truncated)

Fig. 4: Validation loss curves for unnormalized (dashed) and normalized (solid) models. (a) 5-qubit models show a clear separation between normalized and unnormalized training. (b) 8-qubit models reveal that normalized models steadily improve while unnormalized models plateau.

Init Seed	MLP Val Loss	Transformer Val Loss
42	0.340	0.328
100	0.337	0.347
200	0.323	0.345

For computational tractability at larger qubit counts, we use patch-based tokenization that maintains a fixed token count (64) regardless of system size, trading spatial resolution for scalability (5-qubit: 4×4 patches, 32 complex values per patch; 8-qubit: 32×32 patches, 2048 complex values per patch). The 8-qubit results validate this approach: despite extreme compression (2048:1), both models achieve more than 3 times baseline improvement.

Our 8-qubit results also expose a limitation of hierarchical tokenization: when patches become too large, the amount of information decreases sharply. This approach requires patches small enough to preserve local structure while remaining computationally tractable. As the size of a density matrix is exponential in the number of qubits, future work will have to explore methods of balancing compression and tractability.

Our synthetic noise model assumes independent, Markovian noise channels acting on each qubit (depolarizing, amplitude damping, phase damping). Real quantum processors exhibit crosstalk (correlated errors between qubits) and non-Markovian memory effects. Extending our normalization approach to models trained on realistic correlated noise is an important direction for future work.

All experiments in this work rely on classical simulation of quantum states and noise. Deploying these models on actual hardware introduces challenges such as calibration drift, where the noise characteristics of the device change over time. A model trained on a static distribution of noise levels may degrade as the hardware drifts. Future work will investigate strategies for online adaptation or robust training across a wider envelope of noise parameters to mitigate this issue.

One ripe area for future exploration is increasing the effectiveness of our physical projections. As noted in Section 2, the Cholesky factorization has been used in adjacent QST work [14, 15] to enforce physicality by construction at small qubit counts; adapting it to our QSR setting is a natural next step. In addition, our method for enforcing positive eigenvalues is quite crude and can result in unphysical output states that are also far from what the network originally outputted, with prior work by Smolin et al. providing an efficient method for finding valid density matrices from Hermitian one-trace matrices [17].

V. Conclusion and Future Directions

We have applied Frobenius normalization to the preprocessing of quantum state data for quantum state reconstruction. It is observed that multilayer perceptrons and transformers trained on normalized data scale better than their unnormalized counterparts. In this work, every model must learn a mapping for every noise channel simultaneously. Previous work by Morgillo et al. [2] has demonstrated the effectiveness of neural networks for the classification of different kinds of noise channels. Additional improvements may come from having an architecture with multiple noise-channel specific neural networks paired with a noise-channel classifier that functions as a router, analogous to a mixture of experts architecture [18]. Other areas of future work will include scaling up the datasets to larger numbers of qubits and applying more sophisticated physical projection methods.

VI. Acknowledgements

We thank Ravi Ravindran and Shea Tough for feedback on early drafts of this paper. We would also like to thank the reviewers for their helpful feedback on our initial IEEE qCCL 2026 submission. We used a large language model to edit parts of this work for clarity in presentation, for monitoring jobs, and assisting with the code implementation. The technical content, experimental design, and interpretations are our own, and we take

full responsibility for this work. Code and data are available at https://github.com/Amitav-Krishna/transformer_for_reducing_quantum_noise.

References

[1] Richard P. Feynman. "Simulating physics with computers". In: *International Journal of Theoretical Physics* 21.6-7 (1982), pp. 467–488.

[2] Angela Rosy Morgillo et al. "Quantum state reconstruction in a noisy environment via deep learning". In: *Quantum Machine Intelligence* 6.2 (July 2024). ISSN: 2524-4914. DOI: 10.1007/s42484-024-00168-x. URL: http://dx.doi.org/10.1007/s42484-024-00168-x.

[3] Karan Kendre. *Machine Learning for Quantum Noise Reduction*. 2025. arXiv: 2509.16242 [quant-ph]. URL: https://arxiv.org/abs/2509.16242.

[4] Giacomo Torlai et al. "Neural-network quantum state tomography". In: *Nature Physics* 14.5 (Feb. 2018), pp. 447–450. ISSN: 1745-2481. DOI: 10.1038/s41567-018-0048-5. URL: http://dx.doi.org/10.1038/s41567-018-0048-5.

[5] Sergey Ioffe and Christian Szegedy. "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift". In: *Proceedings of the 32nd International Conference on Machine Learning (ICML)*. PMLR. 2015, pp. 448–456.

[6] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. "Layer Normalization". In: *arXiv preprint arXiv:1607.06450* (2016).

[7] Alexey Dosovitskiy et al. "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale". In: *arXiv preprint arXiv:2010.11929* (2020).

[8] Ze Liu et al. "Swin Transformer: Hierarchical Vision Transformer using Shifted Windows". In: *arXiv preprint arXiv:2103.14030* (2021).

[9] Hanrui Wang et al. *Transformer-QEC: Quantum Error Correction Code Decoding with Transferable Transformers*. 2023. arXiv: 2311.16082 [quant-ph]. URL: https://arxiv.org/abs/2311.16082.

[10] Xiao-Dao Lin et al. "Quantum Error Mitigation via Autoencoder Neural Networks". In: *2025 IEEE International Conference on Quantum Control, Computing and Learning (qCCL)*. June 2025, pp. 58–64. DOI: 10.1109/qCCL65142.2025.11158291. (Visited on 01/22/2026).

[11] Daniel Uzcategui-Contreras et al. *Machine Learning approach to reconstruct Density Matrices from Quantum Marginals*. 2024. DOI: 10.48550/ARXIV.2410.11145. arXiv: 2410.11145 [quant-ph]. URL: https://arxiv.org/abs/2410.11145.

[12] Dmytro Bondarenko and Polina Feldmann. "Quantum Autoencoders to Denoise Quantum Data". In: *Phys. Rev. Lett.* 124 (13 2020), p. 130502. DOI: 10.1103/PhysRevLett.124.130502. URL: https://link.aps.org/doi/10.1103/PhysRevLett.124.130502.

[13] Ming-Ming Wang. "Denoising quantum mixed states using quantum autoencoders". In: *Quantum Information Processing* 23 (2 2024), p. 30. DOI: 10.1007/s11128-023-04239-z. URL: https://doi.org/10.1007/s11128-023-04239-z.

[14] Sanjaya Lohani et al. "Machine learning assisted quantum state estimation". In: *Machine Learning: Science and Technology* 1 (3 2020), p. 035007. DOI: 10.1088/2632-2153/ab9a21. URL: https://doi.org/10.1088/2632-2153/ab9a21.

[15] Adriano Macarone Palmieri et al. *Enhancing quantum state tomography via resource-efficient attention-based neural networks*. 2024. arXiv: 2309.10616 [quant-ph]. URL: https://arxiv.org/abs/2309.10616.

[16] Ilya O. Tolstikhin et al. "MLP-Mixer: An all-MLP Architecture for Vision". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 24261–24272.

[17] John A. Smolin, Jay M. Gambetta, and Graeme Smith. "Efficient Method for Computing the Maximum-Likelihood Quantum State from Measurements with Additive Gaussian Noise". In: *Phys. Rev. Lett.* 108 (7 Feb. 2012), p. 070502. DOI: 10.1103/PhysRevLett.108.070502. URL: https://link.aps.org/doi/10.1103/PhysRevLett.108.070502.

[18] Robert A. Jacobs et al. "Adaptive Mixtures of Local Experts". In: *Neural Computation* 3.1 (1991), pp. 79–87.