Machine Learning Enhanced Post-selection in Quantum Networks

Huiping Lin
Tsinghua University
lhp22@mails.tsinghua.edu.cn

Zhengfeng Ji
Tsinghua University, Zhongguancun Laboratory
jizhengfeng@tsinghua.edu.cn

Abstract

Quantum networks are being developed to support secure communication and distributed computation, but their performance is limited by noise in both local operations and transmission. Quantum error correction (QEC) is the standard tool for managing noise, and post-selection adds a front-end stage to the QEC pipeline: instead of decoding every received round, the receiver places a filter in front of the decoder that accepts only trustworthy syndromes for decoding and asks the sender to retry on the rest. In this paper, we examine how machine learning can improve the post-selection process modeled by the filter. We design two such neural network filters, a 3D-CNN and a transformer with FiLM noise conditioning, and evaluate them using the rotated surface code together with an AlphaQubit-style neural network decoder. As a baseline we take the filter built from the decoder's own soft output, which is itself already a strong post-selection signal; against this baseline, our two filters together cut the conditional logical error rate by roughly 2–10× at 10–20% acceptance. Taken together, these results position machine learning enhanced post-selection as a concrete link-layer mechanism for quantum networks, with the practical caveat that the right architecture depends on the operating noise regime.

CCS Concepts

• **Hardware → Quantum error correction and fault tolerance**; *Quantum communication and cryptography*; • **Computing methodologies** → *Neural networks.*

Keywords

quantum networks, quantum error correction, quantum communication, machine learning



This work is licensed under a Creative Commons Attribution 4.0 International License.
QuNet '26, August 17–21, 2026, Denver, CO, USA
© 2026 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-2892-1/26/08
https://doi.org/10.1145/3833409.3833445

1 Introduction

Quantum networks are being developed for secure communication, distributed quantum computing, and a future quantum Internet. Like every quantum technology, their performance is limited by noise from both the transmission channel and the local gates that prepare and measure the transmitted qubits. Quantum error correction (QEC), the workhorse for noise on the computing side, has therefore been steadily imported into network protocols. Encoded quantum repeaters, code-based entanglement distribution, and QKD that uses QEC all build on the same logical-qubit machinery developed for fault-tolerant computation [8, 10, 17, 24–26, 28, 32]. These efforts, however, have focused mainly on what code to use, ranging from surface codes to continuous variable codes, and on how to perform logical operations across the channel, in particular the non-local CNOTs that generate encoded EPR pairs [2, 11, 12, 14, 16]. By contrast, the post-processing of measurement information at the receiver in the quantum-network setting has received much less attention.

Post-selection has a long history in fault-tolerant computing, where magic distillation rejects suspicious copies and retries [6, 7, 22]. Concurrent work [21] applies this idea to quantum networks, where it fits naturally: in QKD and entanglement distribution the transmitted state is known by protocol, so the receiver may freely ask for a retransmission without leaking information. In the formulation they study, the receiver places a *filter* in front of the decoder: the filter reads the measured syndrome of each round and decides whether to accept the round for decoding or reject it and ask the sender to retransmit, and the decoder runs only on accepted rounds. The cost is a reduction in throughput; the payoff is a much lower error rate on accepted rounds.

We focus on the filter side of this formulation, motivated by the rapid progress recently made on the decoder side. The most accurate decoders for surface codes today are themselves neural networks, including AlphaQubit [1] and its successors [35], and they emit a soft confidence score together with each correction. That score serves as a post-selection signal at no additional inference cost, and the same concurrent work [21] establishes that it is already a remarkably strong filter, beating both the syndrome weight baseline and the logical gap heuristic in conditional logical error rate. A natural follow-up question is whether a *separately trained* filter

can do better. Our results show that it can: machine learning enhanced filters further reduce the conditional logical error rate relative to the soft-output baseline, with reductions of roughly 50–90% at strict acceptance rates. The improvement is architecture-dependent, with different models performing best in different noise regimes and the advantage decreasing as the acceptance rate is loosened.

Contributions.

- We design two complementary machine learning filters, a 3D-CNN and a FiLM-conditioned transformer, both built to share data with an AlphaQubit-style neural network decoder as the backend.
- We evaluate the two filters on the rotated surface code family across two tasks and show 2–10× reductions in conditional logical error rate at 10–20% acceptance, with the gain bounded and architecture-dependent rather than uniform across the noise spectrum.

To our knowledge, this is the first machine learning filter designed specifically for syndrome level post-selection in any QEC setting. Beyond the network setting, the same noise-conditioned filter recipe transfers naturally to magic state distillation and other "known-state" tasks in quantum computation, where suspicious copies can be rejected and regenerated by design; we leave this direction to future work.

2 Related Work

QEC in quantum networks. A line of work starting with the encoded quantum repeater [24–26] has demonstrated how stabilizer codes can be used to improve entanglement distribution beyond simple distillation schemes. Subsequent work has explored alternative codes, hardware-tailored constructions, and specific network primitives such as encoded teleportation [12, 16, 20, 34]. More recent network-architecture work organises these protocols into a layered stack, with an explicit link layer for entanglement generation and higher layers for application-level primitives [9, 18]; QEC sits naturally at the boundary between the physical and link layers, where it provides the per-round acceptance signal on which any retransmission policy ultimately depends. Existing designs, however, typically fix a standard decoder and focus on the protocol above it, leaving post-selection as an implicit afterthought rather than a first-class architectural knob. A complementary line distills high-fidelity Bell pairs from many noisy copies using high-rate codes [5, 15, 27], which is orthogonal to and composable with the single-state, syndrome-level filtering we study.

Syndrome level post-selection in QEC tasks. We focus on post-selection in QEC tasks at the syndrome level, where a filter discards rounds based on the measured syndrome before decoding. In this setting, post-selection has a long history in distillation and error detection [3, 7, 22]. The cheapest filter is the syndrome weight: count the non-zero stabilizer measurements and reject above a threshold. More refined filters use the decoder's own internal information, such as the logical gap [4], i.e., the cost difference between the best and second-best matching: a larger gap means higher confidence in the chosen correction. More generally, any decoder that exposes a self-confidence signal can be used for post-selection, including the soft output of a neural-network decoder [1, 23]. More recent work has increasingly focused on practical trade-offs between runtime and accuracy [19, 29, 33]. These efforts mostly start from the magic state distillation setting and are benchmarked on the memory experiment; the move to quantum networks, together with a unified link-layer formulation and a hardness analysis, was carried out in concurrent work [21]. The present paper builds directly on that line and pushes the filter side further: rather than rely on existing decoder-derived heuristics, we ask whether training a dedicated machine learning filter, conditioned on the noise regime, can sharpen the trade-off at the precision-extreme end.

Neural network decoders. Neural network decoders for surface codes have moved from proof-of-concept [30, 31] to matching or exceeding optimal-matching accuracy at experimentally relevant noise levels [1]. Today they are arguably the most accurate option in the high-precision regime, despite their training and inference cost. Beyond a hard recovery operator, a neural network decoder emits a calibrated probability that the logical observable is flipped on the current round; the magnitude of the corresponding logit is a natural per-round confidence score, and thresholding it yields a post-selection filter at no additional inference cost. Concurrent work [21] establishes that this soft-output filter is already a remarkably strong baseline, outperforming both the trivial syndrome weight filter and the logical gap filter by one to three orders of magnitude in conditional logical error rate on surface code families, across the experimentally relevant noise range and different acceptance rates. In our evaluation, we build on a re-implementation of the AlphaQubit-style transformer decoder for the rotated surface code, and use it both as the backend decoder and as the source of our SoftOut baseline filter.

3 Problem Definition

We adopt the post-selection formalism developed in [21], focusing on the simplest network protocol that exposes the question cleanly: Alice prepares a known logical state and transmits it to Bob through a noisy channel. This abstracts QKD, entanglement distribution, and any network primitive in which the transmitted state is fixed by protocol and may be reprepared on demand.

State-Encoding Protocol with post-selection.

- Alice and Bob agree on a stabilizer code C with parameters $[[n, k, d]]$.
- Alice encodes the agreed state into C and sends the n physical qubits through the quantum channel.
- Bob receives the qubits and performs syndrome extraction by measurements.
- Bob feeds the syndrome to a filter F that returns accept or reject, which is exactly the post-selection step. If accepted, Bob runs the decoder D and outputs the corrected state; if rejected, Bob requests a retransmission.

Let $\sigma \in \{0,1\}^m$ be the syndrome measured on a single round, and let the filter $F : \{0,1\}^m \to \{0,1\}$ decide whether to accept it; the round is decoded by a fixed decoder D only when $F(\sigma) = 1$, and is otherwise dropped and retransmitted. We characterise the pair (F, D) by two operational quantities. The *acceptance rate*

$$\mathrm{Acc}(F) = \Pr[F(\sigma) = 1]$$

is the fraction of rounds the filter keeps, equal to the protocol's effective throughput after retransmission. The *conditional logical error rate*

$$\mathrm{condL}(F, D) = \Pr\big[D \text{ fails on } \sigma \;\big|\; F(\sigma) = 1\big]$$

is the probability that the decoder fails on an accepted round, where "D fails on σ" means that the recovery operator returned by D, applied to the noisy encoded state, still leaves a non-trivial logical Pauli on the encoded qubits. All probabilities are taken over the underlying physical noise, which mixes transmission and local circuit errors; we treat that randomness as a placeholder for "whatever stochastic event happened on this round" and never need to look inside it.

For a target acceptance rate $\alpha \in (0,1)$ chosen by the application, we seek the filter that minimises condL subject to $\mathrm{Acc}(F) \geq \alpha$. The optimum is the Bayes-risk threshold rule

$$F^\star_\alpha(\sigma) = \mathbf{1}\big[\Pr[D \text{ fails on } \sigma] \leq \tau_\alpha\big],$$

which accepts a syndrome iff its posterior decoder-failure probability $r(\sigma)$ falls below a level τ_α chosen so that the accept set has measure exactly α. In words, $F^\star_\alpha$ ranks all observable syndromes from least to most risky and keeps the safest α-fraction. The catch is that evaluating $r(\sigma)$ exactly is #P-hard [21], so practical filters approximate it.

Because the Bayes-optimal filter is a threshold rule on a real-valued posterior, every filter F in this paper is realised by learning a scalar confidence score $C : \{0,1\}^m \to \mathbb{R}$ that estimates how "reliable" each syndrome is, and then thresholding C to recover the operational filter at any target α. Therefore, our filters are designed to learn a per-syndrome reliability score rather than directly training a binary classifier with a fixed acceptance rate.

4 Key Techniques

We instantiate three filters that share a common confidence-then-threshold scaffold (§4.1). The first reuses the decoder's own soft output (§4.2). The other two are dedicated risk networks: a 3D-CNN suited to dense, locally-correlated failures (§4.3) and a FiLM-conditioned transformer suited to sparse, long-range failures (§4.4).

4.1 Confidence-threshold Scaffold

Following the score-then-threshold approach motivated in §3, each filter assigns a real-valued confidence score $C(\sigma)$ to every measured syndrome σ and accepts the round when $C(\sigma) \geq \tau$. The threshold τ is calibrated for a target acceptance rate α on a held-out calibration set $\{C_i\}_{i=1}^N$: sort the scores, locate the $(1-\alpha)N$-th order statistic $s_{(k)}$ with $k = \lfloor (1-\alpha)N \rfloor$, and set $\tau = \big(s_{(k)} + s_{(k+1)}\big)/2$. The same scaffold accommodates any choice of C.

4.2 Soft-output Baseline (SoftOut)

Unlike a matching decoder, which returns only a hard recovery operator, a neural-network decoder additionally produces a calibrated probability $p_D(\sigma) \in (0,1)$ that the logical observable is flipped on a given round. We use a reimplementation of the AlphaQubit-style transformer decoder trained per distance, and define

$$C_{\textsc{SoftOut}}(\sigma) \;=\; \left|\log \frac{1 - p_D(\sigma)}{p_D(\sigma)}\right|.$$

This quantity is the magnitude of the decoder's logit, which measures how decisively the decoder commits one way or the other: it vanishes at $p_D = 1/2$ when the decoder is maximally uncertain, and grows without bound as p_D approaches 0 or 1 when the decoder is highly confident in either direction. Filtering by $C_{\textsc{SoftOut}} \geq \tau$ therefore admits exactly the rounds on which the decoder is most decisive. The score requires no additional parameters or training, since p_D is already computed in the decoder's forward pass. This baseline is simple but already very strong: recent work has shown that on QEC-protected network channels, $C_{\textsc{SoftOut}}$ outperforms the syndrome weight filter and the logical gap filter by one to three orders of magnitude in conditional logical error rate [21]. We therefore take SoftOut as the strong baseline that any new filter has to beat, and design the rest of this paper around it.

4.3 CNN Risk Filter (CNN-Risk)

CNN-Risk is the first of the two risk networks; it is designed to capture the dense, locally correlated failure modes that dominate at high noise. Its architecture is shown in Figure 1. The input is the syndrome event volume of one round, a tensor of shape $N \times (d{+}1) \times (d{+}1)$ where N is the number of

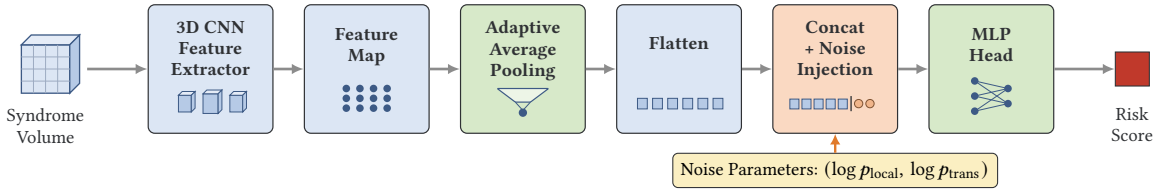


Figure 1: CNN-Risk architecture. The syndrome event volume passes through a stack of 3D convolutions and an adaptive average pool, is flattened, and is then concatenated with the log-noise pair $(\log p_l, \log p_t)$ before a small MLP produces the scalar risk score. Box colors group stages by role: blue for syndrome feature extraction, orange for noise conditioning, and green for spatial aggregation and the readout head.

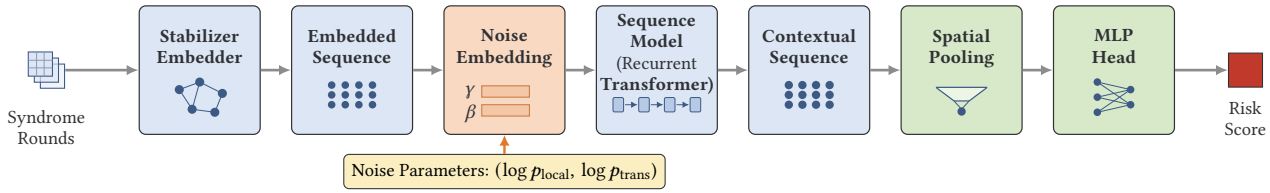


Figure 2: TF-Risk architecture. The per-round syndrome grids are embedded by a stabilizer embedder; the recurrence is initialized by a noise embedding (γ, β) derived from $(\log p_l, \log p_t)$, applied in a FiLM-style modulation to the decoder state so that every attention head and dilated convolution operates on noise-adapted features. A recurrent transformer over the N rounds, a spatial pool, and an MLP produce the scalar risk score. Color scheme is shared with Fig. 1: blue for feature extraction, orange for noise conditioning, and green for aggregation/readout.

stabilizer measurement rounds, together with the noise pair $(\log p_l, \log p_t)$ that characterizes the local and transmission noise rates. A stack of three 3D convolutions with kernel $3 \times 3 \times 3$ and 128 channels is followed by an adaptive 3D average pool to a fixed $128 \times 2 \times 2 \times 2$ descriptor, so the same architecture serves every code distance d. The descriptor is flattened, concatenated with the noise pair, and passed through a two-layer MLP to a scalar risk logit. The full network has roughly 1.0M parameters.

The inductive bias is deliberate. Translation-equivariant 3D convolutions plus pooling extract *bulk*, locally correlated statistics of the syndrome volume; concatenating the noise pair only at the head lets the MLP set a noise-dependent decision boundary without further reshaping the spatial features. This is well matched to the high-noise regime, where logical failures are dense and distributed across the lattice, so the spatial structure that matters is, in a real sense, the average.

4.4 Transformer Risk Filter (TF-Risk)

TF-Risk is the second risk network; it is designed to detect the rare, long-range correlated failures that dominate below threshold and are easily lost to local pooling. Its architecture is shown in Figure 2. TF-Risk reuses the decoder's attention-plus-dilated-convolution backbone at a reduced model width D=96. A stabilizer-grid embedder lifts each per-round syndrome to a D-dimensional feature grid of size $(d{+}1) \times (d{+}1)$; two transformer-style blocks (multi-head self-attention, feed-forward, and 2D convolutions with dilations 1, 2, 4) then process the grid. A recurrence over the N rounds of one shot accumulates evidence,

$$s_{t+1} \;=\; \mathrm{Block}\big(s_t + \mathrm{Embed}(\sigma_t)\big), \qquad t = 0, \ldots, N{-}1,$$

and a distance-independent global average pool over the $(d{+}1) \times (d{+}1)$ grid feeds a small two-layer MLP that emits the scalar risk logit. The full network has roughly 850K parameters, comparable in size to CNN-Risk.

What distinguishes TF-Risk from a smaller copy of the decoder is how the noise pair enters. A two-layer MLP maps $(\log p_l, \log p_t)$ to a FiLM pair $(\gamma, \beta) \in \mathbb{R}^D \times \mathbb{R}^D$. Because the recurrence is iterated N times and every block contains both attention and convolution, this initial conditioning propagates into all downstream features: every attention head and dilated convolution operates on noise-aware representations. In CNN-Risk, by contrast, the noise pair is concatenated only at the head, leaving the convolutional features themselves noise-agnostic. The distinction matters most in the low-noise regime, where failures are rare and the correct inductive bias is to let the noise level reshape what counts as a suspicious pattern globally, rather than only biasing the final score. The architecture's other strength is range: self-attention together with progressively dilated convolutions captures multi-round correlations across the lattice, which is precisely the structure of the chain-like failures that survive at low physical noise. §5 confirms that this architectural choice pays off in the low-noise regime.

Training a separate risk network from sparse failure labels is harder than training the underlying decoder, and two design choices proved essential. First, hard negatives must be drawn from high-confidence correct shots rather than low-confidence ones; at strict acceptance rates the filter only ever sees the high-confidence tail, so this is the region in which training must teach the correct ranking. Second, validation must be tracked per noise configuration, otherwise a single easy configuration with abundant data masks catastrophic failure on configurations where positives are scarce.

Summary. CNN-Risk and TF-Risk are not redundant. They target opposite ends of the noise spectrum (bulk dense failures versus rare long-range failures), and §5 confirms this empirically: each wins decisively in its regime, and neither dominates the other.

5 Experimental Evaluation

5.1 Setup

We evaluate on the rotated surface code at distances $d \in \{5, 7, 9\}$[1], simulated with `stim` [13]. Our primary task is the *state-encoding* protocol of §3: Alice encodes a logical state into the rotated surface code and transmits the n physical qubits to Bob through a noisy channel. We expose two noise axes, a circuit-level depolarizing rate p_l on every local operation (preparation, gates, measurement) and a transmission depolarizing rate p_t applied to every qubit during transit, and sweep p_t at fixed p_l. As a complementary computation-side experiment, we also evaluate on a standard d-round memory circuit with circuit-level noise p_l alone (no transmission stage), to check that our methods carry beyond the network setting. Note that we model transmission as depolarizing noise, whereas loss-dominated photonic channels produce heralded erasures that an erasure-aware filter could exploit; we leave this direction to future work.

We treat p_l and p_t as separate axes because in realistic networks the transmission channel is the dominant and most variable knob, while local operations have a typically lower and more stable error budget. The backend decoder is the AlphaQubit-style transformer described in §4.2, trained per distance. All filters share the threshold-calibration scaffold of §4.1; we report the conditional logical error rate at fixed acceptance rates $\alpha \in \{10\%, 20\%, 50\%, 80\%\}$.

Baseline. As discussed in §4.2, prior work [21] has already established that SoftOut, the decoder's own soft output, dominates the natural alternatives by one to three orders of magnitude in conditional logical error rate on the rotated surface code at distances $d \in \{5, 7, 9\}$, circuit-level noise rates $p_l \in [3\times10^{-3}, 10^{-2}]$, transmission noise rates $p_t \in [0.08, 0.16]$, and acceptance rates $\alpha \in \{10\%, 20\%, 50\%, 80\%\}$: it sits well below both the syndrome weight filter and the logical gap filter when each is paired with the same neural network decoder backend. We therefore do not re-run that comparison here; in everything that follows, SoftOut is the strong baseline against which CNN-Risk and TF-Risk are measured.

[1]We omit d=3: with only four X- and Z-type stabilizers, the post-selection task is too easy and every filter we tested operates near-optimally.

5.2 Comparing Risk Filters with SoftOut

We compare our two risk filters against SoftOut in three settings. First, a low-noise, below-threshold state-encoding regime in which TF-Risk dominates (Fig. 3). Second, a high-noise, above-threshold state-encoding regime in which CNN-Risk dominates (Fig. 4). Third, the memory experiment where TF-Risk again wins, confirming that the recipe generalizes beyond the network setting (Fig. 5).

TF-Risk dominates below threshold. Figure 3 fixes p_l=0.003 and sweeps $p_t \in [0.08, 0.12]$. Both panels (acceptance rate α=10% and α=20%) show TF-Risk strictly below SoftOut for every (d, p_t) point. The largest improvement sits where failures are rarest: at α=10%, p_t=0.08, d=9, TF-Risk reaches condL near 10^{-7} while SoftOut sits near 10^{-6}, an order of magnitude reduction. The gap narrows toward higher p_t where SoftOut already concentrates probability mass on bad rounds. This is the predicted behavior from §4.4: in the below-threshold regime, logical failures are the result of rare, multi-round correlated chains; long-range attention with FiLM noise conditioning catches them where a calibrated logit alone does not. TF-Risk loses this advantage at high p_t combined with high α, where it no longer reliably beats SoftOut; we therefore restrict Fig. 3 to α=10% and α=20%.

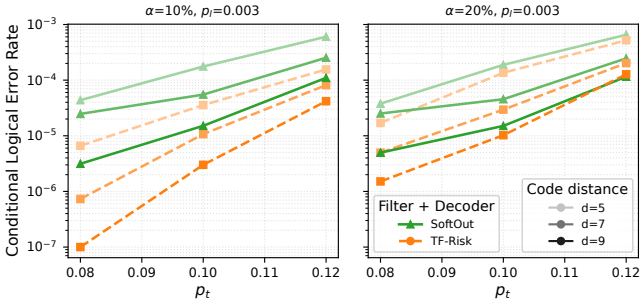


Figure 3: Low-noise, below-threshold regime (p_l=0.003). TF-Risk versus SoftOut on the state-encoding circuit. TF-Risk wins at every operating point in both panels; the gap is $\sim 10\times$ at α=10%, p_t=0.08, d=9, narrowing to $\sim 2\times$ at higher p_t and α=20%.

CNN-Risk dominates above threshold. Figure 4 fixes p_l=0.01 and sweeps $p_t \in [0.08, 0.16]$. CNN-Risk wins at every point in both panels. The win is largest at α=20% ($\sim$ 2–3× across distances) and persists, smaller, at α=50%. This regime is above the surface-code threshold for our noise model: larger

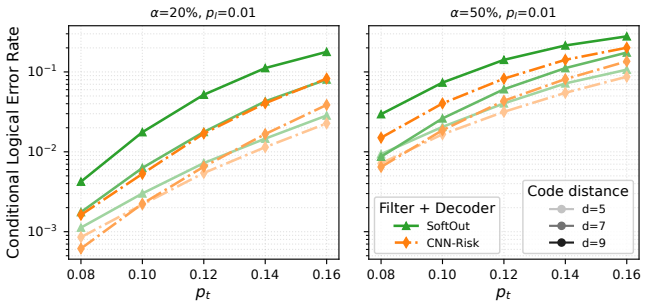


Figure 4: High-noise, above-threshold regime (p_l=0.01). CNN-Risk versus SoftOut on the state-encoding circuit. CNN-Risk wins at every operating point in both α=20% and α=50% panels, with the largest margin at α=20%.

d increases the raw conditional error rate. The fact that post-selection still produces a coherent, monotone improvement in condL across all distances is itself notable. Practically, this is a "break-even" use case: even when local-noise constraints prevent operating below threshold, the receiver can still recover usable rounds by paying a 5× retransmission overhead. The CNN's bulk-statistics inductive bias matches the dense, locally-correlated failures of this regime, exactly as predicted in §4.3. CNN-Risk also beats SoftOut at α=10% and α=80% across the same sweep, with the gap narrowing toward α=80%; we omit those panels for space.

Generalization to memory. Figure 5 repeats the comparison on the memory circuit, which has no transmission stage and is a standard QEC benchmark. The same TF-Risk model architecture is retrained for this circuit. At α=10%, TF-Risk improves on SoftOut consistently for $p_l \geq 0.008$, by up to $\sim 3\times$ at the high-p_l end with d=9. At α=20% the gap is smaller, vanishing at the low-p_l side where the raw condL already approaches 10^{-6} and there is little room left to improve. Although memory is not a network task, this confirms that the recipe transfers: Risk filters are useful wherever the soft-output baseline leaves measurable headroom.

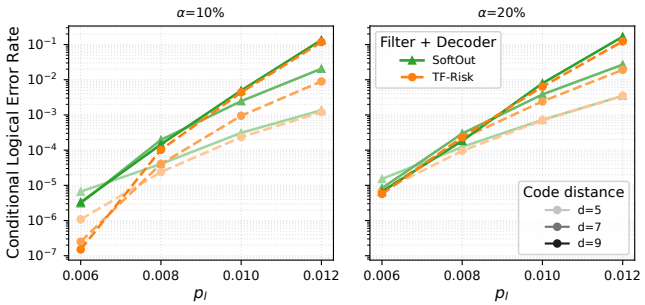


Figure 5: Memory experiment. TF-Risk versus SoftOut, sweeping $p_l \in [0.006, 0.012]$, at α=10% and α=20%. TF-Risk wins clearly at α=10% across moderate-to-high p_l; at α=20% the two are comparable on the low-p_l side and TF-Risk pulls ahead at the high-p_l side.

5.3 Limitations and Discussion

We close with three honest take-homes.

(1) A risk filter trained separately from the decoder gives a clear improvement over SoftOut. Across the regimes we plot, risk filters consistently beat SoftOut, but the gain is moderate, typically within an order of magnitude. At very low raw condL the score distribution of accepted shots is already concentrated on rounds the decoder will get right, so any filter has little headroom to improve over SoftOut. Risk filters only offer a limited improvement, not a dramatic one.

(2) Architecture must match the noise regime. CNN-Risk and TF-Risk address different problems: the first is bulk pattern recognition over a noisy volume, the second is rare-event detection over a noisy time series. In the high-noise regime (p_l=0.01), TF-Risk still beats SoftOut, but by a smaller margin than CNN-Risk; in the low-noise regime (p_l=0.003), CNN-Risk has no advantage over SoftOut at all, which is precisely what motivated TF-Risk in the first place. Forcing one network to win in both regimes is less productive than maintaining both and routing rounds to the right filter at evaluation time.

(3) Higher acceptance rates are structurally harder. Fig. 4 shows CNN-Risk's gap shrinking from α=20% to α=50%, and the same trend holds at α=80%, where CNN-Risk still wins but by a smaller margin. For TF-Risk at p_l=0.003 the effect is more severe: the advantage completely disappears at α=50% and above.

A separate cost worth stating plainly: training and serving a risk filter roughly doubles inference compute over SoftOut and adds a non-trivial training pipeline. Whether this is worth it depends on how much the application values fidelity over throughput: for example, in QKD-like settings where retransmission is cheap, the trade is favorable; in higher-throughput entanglement distribution it requires care.

6 Conclusion

We studied machine learning enhanced post-selection in QEC-protected quantum networks. Taking the soft output of an AlphaQubit-style neural network decoder as a strong baseline, we asked whether a separately trained, noise-conditioned risk filter can do better. The answer is yes: the improvement is architecture-dependent, with a 3D-CNN dominating the high-noise, dense-failure regime and a FiLM-conditioned transformer dominating the low-noise, sparse-failure regime. And as the acceptance rate is loosened, the CNN keeps a positive but shrinking margin over SoftOut at every acceptance rate we tested, whereas the transformer's advantage vanishes with larger acceptance rates. Further exploration includes designing and training better-calibrated risk filters, and transferring this recipe to QEC-protected quantum computation tasks such as magic state distillation.

References

[1] Johannes Bausch, Andrew W Senior, Francisco JH Heras, Thomas Edlich, Alex Davies, Michael Newman, Cody Jones, Kevin Satzinger, Murphy Yuezhen Niu, Sam Blackwell, et al. 2024. Learning high-accuracy error decoding for quantum processors. *Nature* 635, 8040 (2024), 834–840.

[2] Charles H Bennett, Gilles Brassard, Sandu Popescu, Benjamin Schumacher, John A Smolin, and William K Wootters. 1996. Purification of noisy entanglement and faithful teleportation via noisy channels. *Physical review letters* 76, 5 (1996), 722.

[3] Héctor Bombín. 2015. Single-shot fault-tolerant quantum error correction. *Physical Review X* 5, 3 (2015), 031043.

[4] Héctor Bombín, Mihir Pant, Sam Roberts, and Karthik I. Seetharam. 2022. Fault-tolerant Post-Selection for Low Overhead Magic State Preparation. *arXiv preprint arXiv:2212.00813* (2022).

[5] J. Pablo Bonilla Ataides, Hengyun Zhou, Qian Xu, Gefen Baranes, Bikun Li, Mikhail D. Lukin, and Liang Jiang. 2025. Constant-Overhead Fault-Tolerant Bell-Pair Distillation Using High-Rate Codes. *Phys. Rev. Lett.* 135 (2025), 130804. https://doi.org/10.1103/s39k-r2kq

[6] Sergey Bravyi and Jeongwan Haah. 2012. Magic-state distillation with low overhead. *Physical Review A—Atomic, Molecular, and Optical Physics* 86, 5 (2012), 052329.

[7] Sergey Bravyi and Alexei Kitaev. 2005. Universal quantum computation with ideal Clifford gates and noisy ancillas. *Physical Review A—Atomic, Molecular, and Optical Physics* 71, 2 (2005), 022316.

[8] H-J Briegel, Wolfgang Dür, Juan I Cirac, and Peter Zoller. 1998. Quantum repeaters: the role of imperfect local operations in quantum communication. *Physical Review Letters* 81, 26 (1998), 5932.

[9] Axel Dahlberg, Matthew Skrzypczyk, Tim Coopmans, Leon Wubben, Filip Rozpędek, Matteo Pompili, Arian Stolk, Przemysław Pawełczak, Robert Knegjens, Joaquin de Oliveira Filho, Ronald Hanson, and Stephanie Wehner. 2019. A link layer protocol for quantum networks. In *Proceedings of the ACM Special Interest Group on Data Communication (SIGCOMM)*. 159–173.

[10] Wolfgang Dür and Hans J Briegel. 2007. Entanglement purification and quantum error correction. *Reports on Progress in Physics* 70, 8 (2007), 1381–1424.

[11] Jens Eisert, Kurt Jacobs, Polykarpos Papadopoulos, and Martin B Plenio. 2000. Optimal local implementation of nonlocal quantum gates. *Physical Review A* 62, 5 (2000), 052317.

[12] Austin G Fowler, David S Wang, Charles D Hill, Thaddeus D Ladd, Rodney Van Meter, and Lloyd CL Hollenberg. 2010. Surface code quantum communication. *Physical review letters* 104, 18 (2010), 180503.

[13] Craig Gidney. 2021. Stim: a fast stabilizer circuit simulator. *Quantum* 5 (2021), 497.

[14] Daniel Gottesman, Alexei Kitaev, and John Preskill. 2000. Encoding a qubit in an oscillator. *arXiv preprint quant-ph/0008040* (2000).

[15] Andi Gu, Lorenzo Leone, Kenneth Goodenough, and Sumeet Khatri. 2026. Constant Overhead Entanglement Distillation via Scrambling. *Phys. Rev. Lett.* 136 (2026), 110805. https://doi.org/10.1103/3q4z-llv8

[16] Liang Jiang, Jacob M Taylor, Kae Nemoto, William J Munro, Rodney Van Meter, and Mikhail D Lukin. 2009. Quantum repeater with encoding. *Physical Review A—Atomic, Molecular, and Optical Physics* 79, 3 (2009), 032325.

[17] Yumang Jing, Daniel Alsina, and Mohsen Razavi. 2020. Quantum key distribution over quantum repeaters with encoding: Using error detection as an effective postselection tool. *Physical Review Applied* 14, 6 (2020), 064037.

[18] Wojciech Kozlowski and Stephanie Wehner. 2019. Towards large-scale quantum networks. In *Proceedings of the Sixth Annual ACM International Conference on Nanoscale Computing and Communication (NANOCOM)*. 1–7.

[19] Seok-Hyung Lee, Lucas H English, and Stephen D Bartlett. 2026. Efficient post-selection for general quantum LDPC codes. *npj Quantum Information* (2026).

[20] Huiping Lin, Ruixuan Deng, Chris Z Yao, Zhengfeng Ji, and Mingsheng Ying. 2025. Control flow adaption: An efficient simulation method for noisy quantum networks. In *IEEE INFOCOM 2025-IEEE Conference on Computer Communications*. IEEE, 1–10.

[21] Huiping Lin, Kai Zhang, Jianxin Chen, and Zhengfeng Ji. 2026. A Code-Centric Design Space for the Quantum Network Link Layer. (2026). Manuscript under submission.

[22] Daniel Litinski. 2019. Magic state distillation: Not as costly as you think. *Quantum* 3 (2019), 205.

[23] Kai Meinerz, Chae-Yeun Park, and Simon Trebst. 2022. Scalable neural decoder for topological surface codes. *Physical Review Letters* 128, 8 (2022), 080505.

[24] WJ Munro, KA Harrison, AM Stephens, SJ Devitt, and Kae Nemoto. 2010. From quantum multiplexing to high-performance quantum networking. *Nature Photonics* 4, 11 (2010), 792–796.

[25] Sreraman Muralidharan, Jungsang Kim, Norbert Lütkenhaus, Mikhail D Lukin, and Liang Jiang. 2014. Ultrafast and fault-tolerant quantum communication across long distances. *Physical review letters* 112, 25 (2014), 250501.

[26] Sreraman Muralidharan, Linshu Li, Jungsang Kim, Norbert Lütkenhaus, Mikhail D Lukin, and Liang Jiang. 2016. Optimal architectures for long distance quantum communication. *Scientific reports* 6, 1 (2016), 20463.

[27] Christopher A. Pattison, Gefen Baranes, J. Pablo Bonilla Ataides, Mikhail D. Lukin, and Hengyun Zhou. 2025. Constant-Rate Entanglement Distillation for Fast Quantum Interconnects. In *Proceedings of the 52nd Annual International Symposium on Computer Architecture (ISCA)*. 257–270. https://doi.org/10.1145/3695053.3731069

[28] Nicolas Sangouard, Christoph Simon, Hugues De Riedmatten, and Nicolas Gisin. 2011. Quantum repeaters based on atomic ensembles and linear optics. *Reviews of Modern Physics* 83, 1 (2011), 33–80.

[29] Noah Shutty, Michael Newman, and Benjamin Villalonga. 2026. Efficient near-optimal decoding of the surface code through ensembling. *Physical Review Letters* 136, 7 (2026), 070603.

[30] Giacomo Torlai and Roger G Melko. 2017. Neural decoder for topological codes. *Physical review letters* 119, 3 (2017), 030501.

[31] Savvas Varsamopoulos, Ben Criger, and Koen Bertels. 2017. Decoding small surface codes with feedforward neural networks. *Quantum Science and Technology* 3, 1 (2017), 015004.

[32] Zachary D Walton, Ayman F Abouraddy, Alexander V Sergienko, Bahaa EA Saleh, and Malvin C Teich. 2003. Decoherence-free subspaces in quantum key distribution. *Physical review letters* 91, 8 (2003), 087901.

[33] Haipeng Xie, Nobuyuki Yoshioka, Kento Tsubouchi, and Ying Li. 2026. Simple, Efficient, and Generic Post-Selection Decoding for qLDPC codes. *arXiv preprint arXiv:2601.17757* (2026).

[34] Nengkun Yu, Ching-Yi Lai, and Li Zhou. 2021. Protocols for packet quantum network intercommunication. *IEEE Transactions on Quantum Engineering* 2 (2021), 1–9.

[35] Kai Zhang, Zhengzhong Yi, Shaojun Guo, Linghang Kong, Situ Wang, Xiaoyu Zhan, Tan He, Weiping Lin, Tao Jiang, Dongxin Gao, et al. 2026. Learning to Decode in Parallel: Self-Coordinating Neural Network for Real-Time Quantum Error Correction. *arXiv preprint arXiv:2601.09921* (2026).