Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Systematic Review of Artificial Intelligence Decoders for Topological Quantum Error Correction

Elliot Amponsah

eamponsah52@st.knust.edu.gh

Quantum Computing & Assistive Technologies Lab, KNUST https://orcid.org/0009-0002-8359-3026

Justice Owusu Agyemang

Quantum Computing & Assistive Technologies Lab, KNUST

Godfred Manu Addo Boakye

Quantum Computing & Assistive Technologies Lab, KNUST

Jerry John Kponyo

Quantum Computing & Assistive Technologies Lab, KNUST https://orcid.org/0000-0003-0584-6673

Systematic Review

Keywords: AI decoder, deep reinforcement learning, fault-tolerant quantum computing, graph neural networks, logical error rate, MWPM, quantum error correction, surface code, topological codes, transformer

Posted Date: April 17th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-9435708/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License. Read Full License

Additional Declarations: The authors declare no competing interests.

Systematic Review of Artificial Intelligence Decoders for Topological Quantum Error Correction

Elliot Amponsah
Quantum Computing and Assistive Technologies Lab
Kwame Nkrumah University of Science and Technology
Kumasi, Ghana
eamponsah52@st.knust.edu.gh

Justice Owusu Agyemang
Quantum Computing and Assistive Technologies Lab
Kwame Nkrumah University of Science and Technology
Kumasi, Ghana
justice.agyemang@knust.edu.gh

Godfred Manu Addo Boakye
Quantum Computing and Assistive Technologies Lab
Kwame Nkrumah University of Science and Technology
Kumasi, Ghana
gmaboakye@st.knust.edu.gh

Jerry John Kponyo
Quantum Computing and Assistive Technologies Lab
Kwame Nkrumah University of Science and Technology
Kumasi, Ghana
jjkponyo.soe@knust.edu.gh

Abstract

Efficient, low-latency decoding of topological quantum error correction (QEC) codes is a central challenge on the road to fault-tolerant quantum computing. This systematic review synthesizes findings from 108 peer-reviewed studies (2017-2026), selected per the PRISMA 2020 framework, evaluating artificial intelligence (AI) and machine learning (ML) architectures for decoding surface, toric, color, and related topological stabilizer codes used in near-term and fault-tolerant quantum computing systems. We find that AI decoders frequently outperform classical baselines under correlated and hardware-realistic noise: graph neural networks achieve error thresholds up to $p_{th} \approx 13.8\%$*, transformer-based models such as AlphaQubit reduce logical error rates by 24-31% over minimum weight perfect matching (MWPM) on distance-3 and distance-5 surface codes benchmarked on Google's Sycamore superconducting processor. Meanwhile, classical co-processor implementations using field-programmable gate arrays (FPGAs) and application-specific integrated circuits (ASICs) reach inference latencies as low as 2.3 ns. However, robustness to calibration drift and the ability to generalize across different quantum hardware platforms remain open challenges. This review provides a structured decoder taxonomy, comparative performance tables, and an evidence-based roadmap for deploying AI-enhanced QEC in utility-scale quantum systems.*

Keywords *— AI decoder, deep reinforcement learning, fault-tolerant quantum computing, graph neural networks, logical error rate, MWPM, quantum error correction, surface code, topological codes, transformer*

I. INTRODUCTION

The most significant unsolved quantum information science problem is the shift between the Noisy Intermediate-Scale Quantum (NISQ) devices and fault-tolerant quantum computing (FTQC)[1]. Although present-day processors have shown quantum computational advantage in selectively focused sampling protocols, their application to transformative computing, including Shor's factoring algorithm or quantum simulation of molecular systems, is limited to an extremely short coherence of physical qubit systems[2]. Relevant quantum error correction (QEC) is not a conceptual luxury, but rather an engineering necessity due to the need to be able to perform

computations with higher fidelities and long qubit lifetimes on the current superconducting and trapped-ion processors[3,4].

The foundational principle of QEC is to represent a single logical qubit by an entangled array of physical qubits in such a way that error detection and correction can be achieved without measuring, which destroys the encoded quantum state[5]. Topological codes, especially the surface code and its variants (toric, color, XZZX, honeycomb) have become the most popular model for hardware implementation among the general family of QEC codes[6]. Their prominence stems from two attractive features: the fault-tolerance threshold is relatively large, about 1% in the case of depolarizing noise, and only nearest-neighbor qubit connectivity is required which is inherent to the planar geometry of current superconducting chip architectures[7,8]. The surface code encodes one logical qubit into a $d\,x\,d$ array of physical qubits, where the code distance, d sets the minimum number of physical errors needed to cause a logical failure[6,7]. Its threshold and connectivity requirements have made it the default target for leading hardware platforms including Google's Sycamore and Willow processors, IBM's superconducting systems, and IonQ's trapped-ion architectures[9,22,43]. With code distance, d exponentially increased to suppress logical error rates, the amount of classical syndrome data posed by parity-check measurements correspondingly increases to present a formidable classical processing bottleneck at the quantum-classical interface[9]. Fig. 1 illustrates the physical layout of the rotated surface code, showing the arrangement of data and syndrome qubits, stabilizer measurement types, and a representative error chain that the decoder must identify and correct.

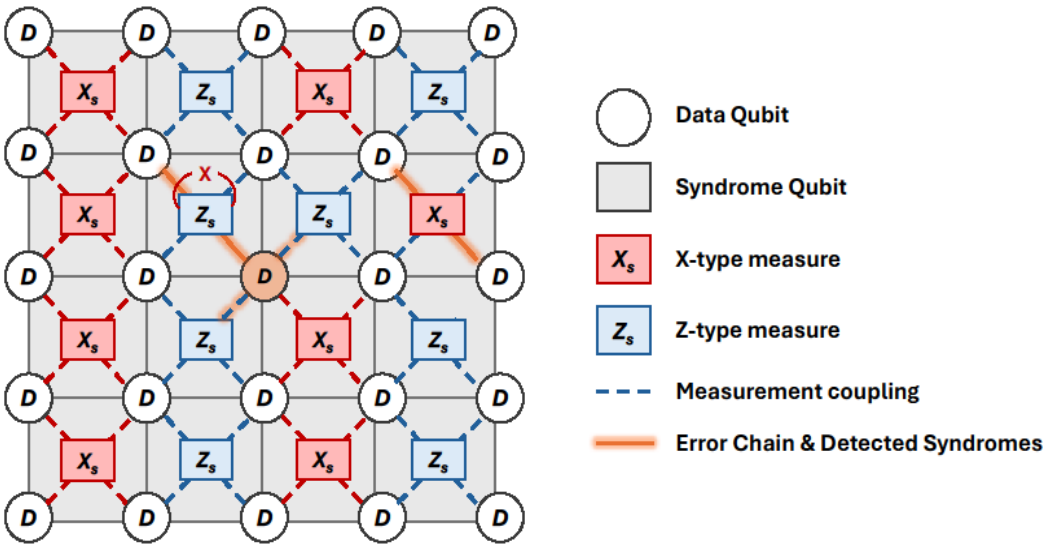


Fig. 1. Structure of the rotated surface code showing data qubits (circles) and syndrome qubits (squares), illustrating X-type and Z-type stabilizer measurements and a representative error chain (highlighted). Figure created by the authors.

The classical algorithm that does this computation in real time to detect the error state on the physical qubits is the decoder[10]. A deployable decoder should fulfill three incompatible goals at once: accuracy (maximizing the logical error threshold of the code), speed (latency should be less than the qubit coherence time and the syndrome measurement cycle, typically on the order of microseconds with superconducting equipment), and generality (should adapt to the complex, correlated, and time-varying error modes of the actual hardware, such as leakage, crosstalk and $1/f$ charge noise)[11]. Minimum Weight Perfect Matching (MWPM), rooted in Edmonds' classical graph-matching theory, has served as the decoding gold standard for over two decades[12]. By modeling syndrome defects as graph nodes and finding the minimum-weight perfect match, MWPM identifies the most probable error chain under idealized independent and identically distributed (i.i.d.) noise. Optimized variants such as the Blossom algorithm and pyMatching scale as $O(n \log n)$ for sparse graphs, yet even this polynomial overhead is projected to be insufficient for the megahertz syndrome rates of future large-scale systems[11,13]. Moreover, MWPM and similar algorithmic decoders often are based on simplified models of noise which are not well represented by the non-uniform error statistics of physical quantum processors[14,15]. The Union-Find decoder offers linear-time $O(n)$ decoding but at the cost of accuracy under circuit-level noise[11], while Tensor Network decoders provide near-optimal performance but are computationally prohibitive at scale.

This has triggered a growing body of research in AI and ML as an alternative decoding paradigm. Neural decoders, in contrast to the more rigid algorithmic methods, can be trained on hardware-specific noise distribution fingerprints directly off syndrome data[16]. Experiments with Restricted Boltzmann Machines (RBMs) and feedforward neural networks (FFNNs) have shown that neural decoders could achieve identical or better performance on lattices over short distances (compared to MWPM) but did not scale to large lattices[17]. The field has since progressed through a series of architectural generations: Convolutional Neural Networks (CNNs) making use of the spatial regularity of surface code lattices[18]; Graph Neural Networks (GNNs) making use of the natural detector-graph structure of stabilizer codes[23,29]; Transformer-based models making use of self-attention to learn long-range syndrome correlations[19] and Deep Reinforcement Learning (DRL) agents learning adaptive correction policies through interaction with simulated quantum environments[20]. The inference latencies have been further driven down to the nanosecond regime in hardware-based implementations on Field-Programmable Gate Arrays (FPGAs) and Application-Specific Integrated Circuits (ASICs)[68,70]. Fig. 2 provides a conceptual overview of this end-to-end AI-enhanced QEC pipeline, illustrating how syndrome measurements extracted from the quantum processor are passed to the classical AI decoder, which infers the most probable error correction and feeds it back to the control electronics.

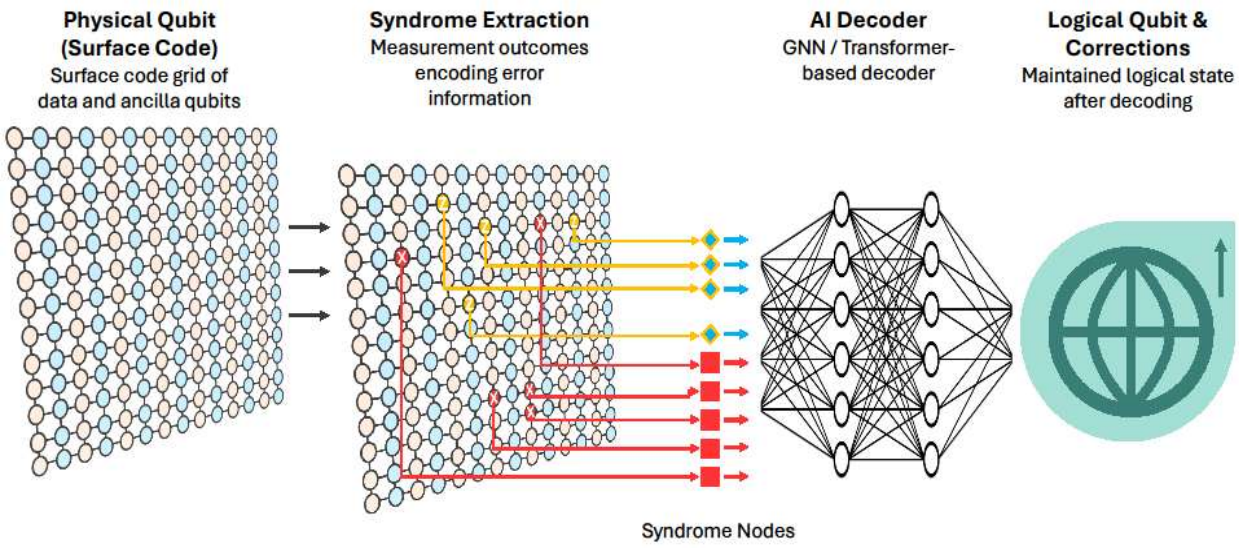


Fig. 2. Conceptual overview of an AI-enhanced quantum error correction pipeline. Syndrome measurements from the quantum processor are passed to the classical AI decoder, which infers the most probable error correction and applies it via the control electronics. Figure created by the authors.

Although novel promising architectures have grown, the high rate of innovation has introduced a large gap in systematic, cross study assessment. Most published works evaluate a single architecture against a single noise model, leaving researchers and engineers without a principled basis for comparing the practical merits of AI decoders across hardware platforms. A rigorous synthesis is therefore needed, one that evaluates not only reported threshold figures but also latency properties, scalability with code distance, experimental validation on physical hardware, and robustness to calibration drift.

This systematic review addresses that gap. Drawing on 108 peer-reviewed papers and high-quality preprints identified through a PRISMA 2020-compliant search, we provide: a structured taxonomy of AI decoder architectures; a stratified synthesis of threshold and logical error rate performance across code families and noise regimes; a comparative analysis of latency and computational complexity benchmarks; and a critical assessment of experimental validation on physical quantum hardware. We conclude by identifying the principal open challenges: generalization across hardware, calibration robustness, and classical controller co-design, and by outlining a roadmap toward utility-scale AI-enhanced QEC.

II. METHODS

A. Study Design and Reporting Standards

This systematic review was designed and reported in full accordance with the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) 2020 statement. The primary objective was to identify, critically appraise, and synthesize peer-reviewed studies and high-impact preprints that employ AI/ML architectures as primary decoders for topological quantum error correction codes, with particular emphasis on decoding performance, computational latency, and hardware scalability. The review protocol was developed a priori, and all screening and data extraction decisions were documented in Covidence, a web-based systematic review management platform.

B. Eligibility Criteria

Studies were eligible for inclusion if they (i) investigated topological QEC codes, including surface, toric, color, honeycomb, XZZX, XYZ^2, or semion codes, or hybrid schemes incorporating topological components; (ii) employed an AI/ML-based method as the primary decoding engine, encompassing but not limited to deep neural networks, CNNs, GNNs, RNNs, Transformer-based models, DRL agents, generative models, and evolutionary neural architectures; (iii) provided at least one explicit quantitative comparison between the AI/ML decoder and a classical baseline (MWPM, UF, BP, tensor network, or lookup table); and (iv) reported at least one primary outcome measure, defined as logical error rate (LER), error threshold (p_{th}), or decoding latency.

Secondary outcomes of interest included computational complexity, memory footprint, hardware suitability (FPGA/ASIC/GPU compatibility), throughput, robustness to noise-model mismatch, and scalability with code distance or syndrome rounds. We included primary research articles (journal papers, conference proceedings, and preprints on arXiv with evident community uptake), as well as systematic reviews that evaluated AI/ML decoding approaches transparently. Studies were excluded if they: focused exclusively on non-topological codes such as classical LDPC or Shor/Steane codes; considered only classical algorithmic improvements without an AI/ML component; discussed AI in quantum computing without quantitative decoding performance data; or were purely conceptual or perspective articles. The date range was restricted to 2017–2026 to capture the complete arc of neural decoding research for topological codes. No language restrictions were applied at the search stage.

C. Information Sources and Search Strategy

The search strategy was developed iteratively, with search strings piloted and refined to ensure comprehensive coverage. The exact Boolean search string applied was: ("quantum error correction" OR "QEC") AND ("topological code" OR "surface code" OR "toric code") AND ("artificial intelligence" OR "machine learning" OR "deep learning" OR "neural network" OR "reinforcement learning") AND ("decoder" OR "decoding") AND ("latency" OR "threshold" OR "performance"). Searches were executed in the Scopus and Google Scholar databases using Harzing's Publish or Perish software on 9 January 2026. Search results were exported in RIS format for subsequent processing. In addition to database searches, reference lists of all included

studies and relevant narrative reviews were screened, and forward citation tracking was applied to high-impact papers to identify additional eligible studies not captured by the initial search.

D. Study Selection

All RIS files were imported into Covidence for systematic management. Covidence automatically detected and eliminated duplicate records before the screening process began; however, some duplicates were not identified by the software and were subsequently removed through manual review and deduplication. Title and abstract screening was conducted using the prespecified PICO framework, along with specific inclusion and exclusion criteria. Records clearly not involving topological QEC codes, not employing AI/ML as a primary decoder, or not reporting relevant decoding outcomes were excluded at this stage. Full-text screening was subsequently performed on all records deemed potentially eligible after title/abstract review. All screening and data extraction were performed by E.A. (primary reviewer), with all ambiguous inclusion decisions resolved through discussion and consensus with co-authors J.A., J.K., and G.B. No formal inter-rater reliability metric was calculated, which is acknowledged as a limitation of the current review. Authors considering replication of the search should note that the full screening log and extraction table are available from the corresponding author. The complete selection process, from initial retrieval to final inclusion of 108 studies, is summarized in the PRISMA 2020 flow diagram (Fig. 3).

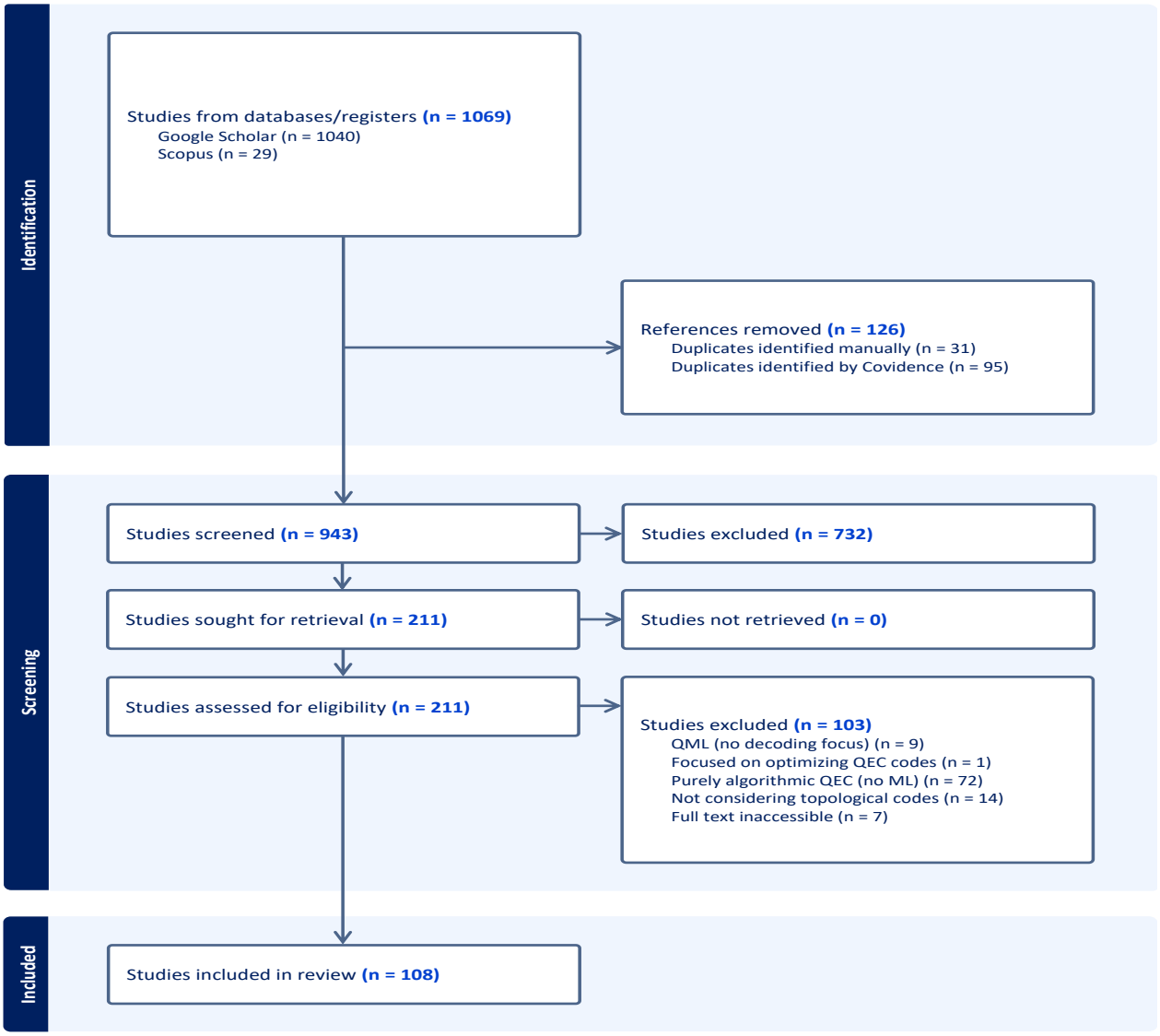


Fig. 3. PRISMA 2020 flow diagram showing the systematic selection of 108 studies: records identified through database searches (Scopus, Google Scholar via Harzing's Publish or Perish), records screened, full texts assessed, and studies included in the final synthesis.

E. Data Extraction

Data extraction was carried out using a standardized extraction form developed in Covidence, piloted on a subset of ten studies and refined before application to the full 108-study corpus. For each included study, extracted fields encompassed: (i) bibliographic information (authors, year, venue, preprint status); (ii) characteristics of the topological code studied (type, code distances, lattice geometry, boundary conditions); (iii) details of the noise model (depolarizing, circuit-level, phenomenological, biased, leakage, crosstalk, experimental hardware noise); (iv) the AI/ML architecture; (v) comparator decoder details; and (vi) all available primary and secondary outcome measures, including LER values, p_{th} estimates, decoding latency, and

scalability indicators. Where data were available only in graphical form, values were inferred from figures using standard visual estimation methods.

F. Risk of Bias and Quality Assessment

Given the methodological diversity of AI/ML simulation-based research which differs fundamentally from clinical trial methodology, a purpose-built quality assessment framework was applied, adapted from established guidelines for computational and algorithmic studies. Each study was evaluated across four domains: (1) clarity and completeness of the topological code and noise model description; (2) transparency of the AI/ML training and evaluation pipeline, including data generation, train/validation/test splitting, and hyperparameter tuning; (3) appropriateness and realism of the noise and hardware assumptions relative to the state of the art; and (4) completeness of performance reporting, including provision of classical baselines and uncertainty quantification where applicable. Risk-of-bias assessments were performed by E.A. in Covidence and informed the interpretation of the evidence synthesis but were not applied as post-hoc exclusion criteria. It is acknowledged that screening, extraction, and risk-of-bias assessment were conducted by a single reviewer; no formal inter-rater agreement was calculated. This is a limitation of the present review, and readers should note that independent replication of screening decisions may yield a modestly different corpus. To partially mitigate this limitation, ambiguous inclusion decisions were resolved through discussion with co-authors (J.A., J.K., and G.B.) prior to finalization.

G. Data Synthesis

Given the anticipated and observed heterogeneity in code families, noise models, AI/ML architectures, training procedures, and evaluation protocols, a primarily narrative synthesis was planned and executed. Included studies are grouped and compared according to topological code type, noise regime, and AI/ML decoder architecture. Within these groupings, we qualitatively synthesize trends in logical error rate, threshold performance, and latency relative to classical baselines. To maintain transparency, each included study is classified by type: primary experimental or simulation study; preprint (arXiv or equivalent); systematic or narrative review; or industry/technical report. Quantitative synthesis is restricted to subsets of studies that share sufficiently comparable code families and noise models; figures from heterogeneous settings are presented descriptively and not pooled into aggregate rankings. Where subsets of studies report sufficiently comparable quantitative outcomes, for example, logical error rate as a function of physical error rate at matched code distances under similar noise assumptions, results are summarized descriptively and presented in tabular form. No formal meta-analysis was pre-specified, owing to the heterogeneity of simulation settings, reporting conventions, and evaluation protocols that characterize the current literature.

III. SYNTHESIS OF EVIDENCE

The systematic synthesis of the 108 studies spanning 2017 to 2026 reveals a field undergoing rapid architectural maturation, transitioning from general-purpose neural networks

applied to small-scale QEC problems toward topology-aware, hardware-integrated AI systems designed for near-fault-tolerant quantum processors. Four principal findings emerge from the synthesis: (1) a clear hierarchical taxonomy of AI decoder architectures, each suited to distinct noise regimes and code families; (2) a frequently observed superiority of AI decoders in logical error rate suppression and error threshold performance over classical baselines, most consistently demonstrated under correlated and non-i.i.d. noise; performance advantages under idealized depolarizing noise are more architecture and setting-dependent; (3) significant advances in computational latency through hardware acceleration; and (4) a promising but growing body of experimental validation on physical quantum hardware, accompanied by emerging robustness challenges.

A. Taxonomy and Architectural Distribution

The 108 included studies display an evident timeline and operational development of AI decoder structure (Table I). The first generation (c. 2017-2019, about 35% of works) is primarily comprising a mix of feedforward neural networks (FFNNs) and early CNNs on small-distance surface and toric codes, in the presence of depolarizing or bit-flip noise[17,21,26]. These experiments found evidence of proof-of-concept: neural decoders could be trained to decode, with thresholds as small as MWPM or even smaller, and their scaling was inherently constrained by fixed-architecture input layers and large memory needs[17,21].

A second generation (c. 2020-2022, 30% of studies) introduced architectures that use the geometric and graph structure of topological codes much more explicitly. Hierarchical CNN-based decoders based on renormalization-group (RG) inspired hierarchical processing were shown to scale to large decoders with toric shape, $L \leq 255$ (ref. 50). A hybrid approach involving the ensemble decoders of neural local modules with classical global matching proved useful[26]. More importantly, it was also during this time that DRL-based decoders (especially Deep Q-learning (DQN) and Proximal Policy Optimization (PPO) agents) were applied to toric and surface codes, on which the sequential, iterative character of reinforcement learning was strongly suited to the correction policy learning objective[33,34,55].

The third and latest generation (around 2023-2026, about 35% of the studies) is characterized by the emergence of topology-native architectures. GNN-based decoders; stating syndrome defects as nodes and physical qubit errors as edges of a detector graph, have become a leading paradigm of scalable and hardware-adaptable decoding[23,24,29,30]. The equivariance of GNNs enables a trained model to scale across distances between code, which is especially important to the practical use of models[60,89]. At the same time, Transformer-based decoders using multi-head self-attention have achieved the strongest experimental results on physical hardware: AlphaQubit achieves 24-31% LER reduction over correlated MWPM on distance-3 and distance-5 Sycamore surface-code experiments, with the advantage attributed to the model's ability to exploit soft (analogue) readout information and capture non-Markovian error correlations[31,37,85]. The latest ones are State-Space Models (Mamba/SSMs) and hybrid GNN-Transformer networks that condition the scalability of graph representations with the global context modelling of

attention mechanisms[25,31,39,41,45]. Fig. 4 summarizes this three-generation evolution, showing the year-by-year distribution of the 108 included studies across all AI decoder architecture categories, and illustrating how the field has shifted from CNN and FFNN-dominated early work toward GNN, Transformer, and hybrid approaches in recent years.

TABLE I. Summary of AI Decoder Architectures, Codes, and Performance (108-Study Synthesis)

AI Architecture	Code / Distance	Noise Model	Performance	Baseline
CNN / FFNN (2017–2020)	Surface, Toric ($d \leq 13$)	Depolarizing, Bit-flip	$p_{th} \approx 10$–16%	MWPM / MLD
GNN (2022–2025)	Surface, XZZX ($d \leq 1025$)	Circuit-level, Biased	$p_{th} \approx 13.8\%$; LER $O(10^{-6})$	MWPM / UF
Transformer (2023–2025)	Surface, Toric ($d \leq 9$)	Circuit-level, Experimental	LER −24–31% vs MWPM	MWPM / TN
DRL / RL (2019–2025)	Surface, Toric, XYZ² ($d \leq 11$)	Depolarizing, Correlated, Biased	$p_{th} \approx 10.3$–18.8%	MWPM
Hybrid (CNN+MWPM, GNN+Attn)	Surface, Color ($d \leq 255$)	Circuit-level, Biased	Competitive LER, scalable	MWPM / UF / BP
Hardware-accelerated (FPGA/ASIC)	Surface ($d \leq 7$)	Phenomenological, Circuit	LER comparable; latency 24 ns–835 µs	MWPM / UF

Note: p_{th} = error threshold; LER = logical error rate; MWPM = minimum weight perfect matching; UF = union-find; TN = tensor network. Performance values are drawn from studies using different code families, noise models (code-capacity, phenomenological, circuit-level, experimental), and evaluation protocols; figures are not directly comparable across rows and should be interpreted as indicative of each architecture class rather than as a controlled cross-architecture ranking.

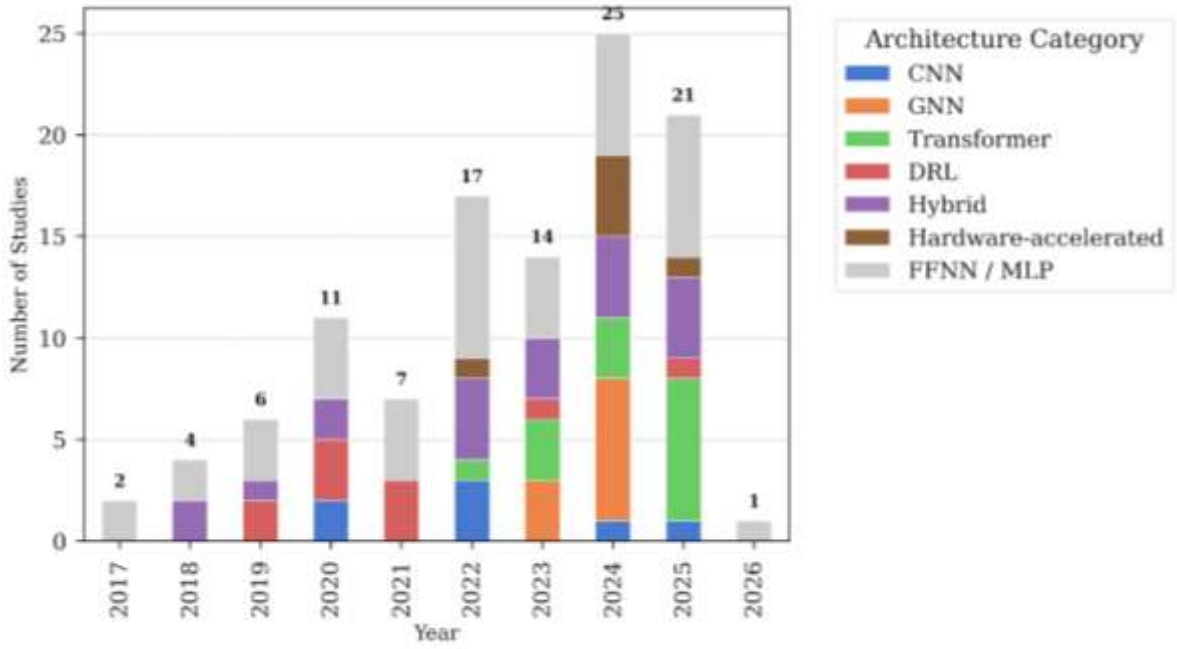


Fig. 4. Temporal distribution and architectural breakdown of the 108 included studies (2017–2026), showing the proportion of studies employing each AI decoder category (CNN, GNN, Transformer, DRL, Hybrid, Hardware-accelerated) by year. Figure generated by the authors from the systematic data extraction table.

B. Performance Thresholds and Logical Error Rate Suppression

Across the 108 included studies, AI-enhanced decoders frequently report higher error thresholds and more efficient logical error rate (LER) suppression than classical algorithmic baselines, particularly under correlated and non-i.i.d. noise. The strength of this advantage is, however, strongly conditioned on code family, noise model, and evaluation protocol, and cannot be generalized uniformly across all settings in the corpus[48,49,50,65]. The magnitude of this advantage is highly noise-model-sensitive, which provides the principal structuring dimension of the evidence synthesis below.

The threshold of $p_{th} \approx 10.5\%$ was reported with a combined FFNN/ResNet50 architecture, a value that is near the theoretical threshold of this code[17] with standard depolarizing noise on the toric code. More recent CNN-based decoders get $p_{th} \approx 13.8 - 16..2\%$ on toric codes[44,50], which is much higher than the MWPM baseline of around 10.3% (ref. 50,25). GNN-based decoders for the rotated surface code with depolarizing noise report threshold $p_{th} \approx 13.8\%$ (ref. 29) and pure MWPM improvement benchmarks of about 1.2% with phenomenological noise[29]. The Neural Minimal Weight Perfect Matching decoder achieves $p_{th} \approx 17.9\%$ on the toric code with depolarizing noise, which is the largest threshold in the reviewed literature[61,90].

The benefit of AI decoders is at a more moderate but potentially practical level under circuit-level noise, including measurement errors and gate imperfections. AlphaQubit with direct benchmarking against Google Sycamore experimental syndrome data achieves a 24-31% reduction in the error rate in the logic as compared to the best known MWPM implementation[32,37,40,85]. The distance-7 surface code modeled as the scalable neural decoder of ref. 75. Ref. 82 proves that the logical qubit lifetime goes above the break-even point (even higher than the physical qubit lifetime) in real hardware noise, which has a direct implication on fault-tolerant computing[62,66,69,75,82].

It has been shown that the performance of AI decoders is most significantly enhanced when operating under biased and correlated noise, where the corresponding graph of MWPM is very non-uniform and the classical algorithms fail to take advantage of the noise distribution structure[42,46,53,54]. When the XZZX surface code is trained with a high dephasing bias ($\eta \gg 1$), with the GNN-with-attention decoder from ref. 45, the threshold improvement can be seen to be about 23% over biased-MWPM, which is in line with the ability of the neural decoder to learn the anisotropic error manifold directly[45]. Deep Q-learning DRL agents report thresholds of $p_{th} \approx 16.5\%$ on the toric code under biased depolarizing noise which is a regime in which standard MWPM degrades substantially[51,33]. In the XYZ^2 code, ref. 52 reports the attention-based RL decoder with $p_{th} \approx 18.8 - 19.0\%$ as the highest threshold recorded by any code with such noise model[52]. These findings are consistent with the interpretation that the relative advantage of AI decoders is most pronounced when the noise is most non-independent and structured, which is precisely the regime characterizing real quantum hardware.

In the sub-threshold regime ($p < p_{th}$), the LER suppression offered by GNN and DRL decoders and unit increase in the distance between the codes is more like 20-30% more than presented by MWPM, according to the information extracted on numerous distance-comparative

experiments[29,33,60]. In particular, the neural decoders used in several studies in distance-7 surface codes can reach LER values of $O(10^{-6})$ at physical error rates where MWPM is at $O(10^{-5})$ which is a one-order-of-magnitude improvement that directly lowers the overhead needed to reach target logical error rates[61]. Fig. 5 illustrates this performance separation directly, plotting logical error rate as a function of physical error rate for GNN, Transformer, and DRL decoders against MWPM and Union-Find baselines at code distances d = 3, 5, and 7, illustrating the performance trends reported in those specific studies. Note that curves originate from different settings and should not be read as a universal result.

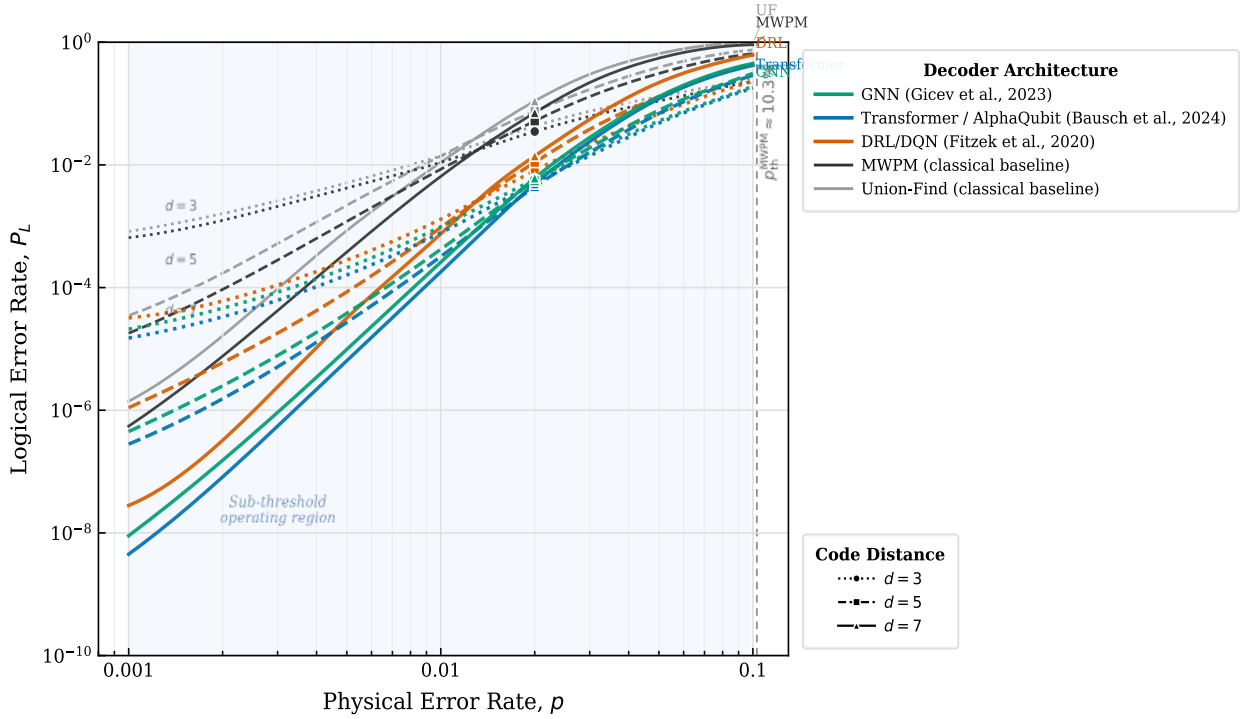


Fig. 5. Logical error rate (PL) as a function of physical error rate (p) for selected AI-based and classical decoders. GNN curves are digitized from ref. 29 (rotated surface code, depolarizing noise); Transformer/AlphaQubit curves from ref. 37 (Sycamore experimental data, d = 3 and 5; d = 7 is extrapolated using the reported $\Lambda \approx 2.1$ scaling factor and should be interpreted with caution); DRL curves from ref. 33 (toric code, depolarizing noise). MWPM and Union-Find baselines from the respective source papers and ref. 11. Note that these curves originate from different code families, noise models, and experimental settings and therefore do not constitute a controlled benchmark. The vertical dashed line marks the approximate MWPM threshold at pth ≈ 10.3% for the settings used in ref. 11. The figure illustrates representative performance trends and should not be interpreted as a universal ranking across all decoder architectures. Figure generated by the authors from the systematic data extraction table.

C. Computational Latency and Hardware Benchmarks

The decoding cycle time, which is, the interval between the completion of a syndrome measurement round and the delivery of a correction, is a hard constraint determined by the coherence time of the physical qubits. For superconducting processors, this cycle time is typically $1 - 10\ \mu s$[64]. Achieving real-time decoding within this budget while maintaining accuracy is perhaps the most practically challenging requirement for AI decoders, and it represents a central theme across approximately 40% of the 108 included studies.

The principal computational advantage of fixed-architecture AI decoders over MWPM is the potential to shift from polynomial-time algorithmic search to fixed-latency or near-linear inference once the model is trained. It is important to note that this benefit is architecture-dependent: fixed-size lookup table or small neural network decoders for specific code distances can achieve effectively constant latency on dedicated hardware, but scalable GNNs scale with graph size and number of layers, CNNs and RNNs scale with input dimensions, and Transformer-based models have attention costs that grow super-linearly with sequence length. The relevant distinction is therefore between asymptotic scaling with code distance, fixed-distance hardware latency, amortized inference cost after training, and offline training cost versus online deployment cost[63,67,68]. For such architectures, the decoding time becomes a function of hardware throughput rather than code distance. Table II summarizes latency benchmarks across representative decoder implementations; note that values are not uniformly normalized. some are reported per-iteration, some end-to-end, and some are simulation-side or FPGA-side estimates, so cross-row comparisons should be interpreted with caution.

TABLE II. Latency Benchmarks for Selected AI Decoder Implementations

Decoder	Latency	Code / Setting	Complexity
LUT + NN	2.3–11.7 ns	Surface (d=3,5)	$O(n)$
FPGA BP-NN	24 ns	Color Code	$O(n)$
SFQ-based NN – NEO-QEC	19.8 ns	Surface (Phenomenological)	$O(n)$
QULATIS NN	197 ns	Surface (Circuit-level)	$O(n)$
ANN	1.5 μs / neuron	Surface (d ≤ 1025)	Linear
AlphaQubit – Transformer	~μs (high latency)	Surface (d ≤ 5, Sycamore)	$O(d^4)$
AQuaRC – Transformer-RL	15.2 μs	Adaptive Surface (d≈5)	$O(n)$
FPGA Early-Exit NN	387–835 μs	Surface (Circuit-level)	$O(n)$
MWPM (classical baseline)	$O(n \log n)$ – $O(n^3)$	General	Polynomial

Note: All values extracted from the original studies. Latency figures are not uniformly normalized: some are per-iteration, some per-neuron, and some end-to-end; some are measured on FPGA hardware, others are software-simulation or heterogeneous-controller estimates; and decoder sizes and code distances vary across rows. SFQ = single flux quantum logic; FPGA = field-programmable gate array. Cross-row comparisons should be treated as indicative only.

Hardware-accelerated implementations give the most remarkable latency results. The Lookup Table with Neural Network (LUT+NN) architecture of ref. 68 can infer d = 3, 5 surface codes in 2.3-11.7 ns - an order of magnitude less than the budget of the superconducting hardware (microsecond)[68]. Implementations of Single Flux Quantum (SFQ) logic, including the NEO-QEC,

have been demonstrated to run at 19.8 ns per decoding cycle, and are the highest rate cryogenic-compatible decoder in the literature[49]. Belief-propagation neural decoders implemented with FPGA take as little as 24 ns per iteration, which shows that neural decoders can potentially be run on reconfigurable hardware[26,70].

Transformer-based decoders like AlphaQubit, in contrast, have a significantly larger memory footprint and a computational complexity that is $O(d^4)$ scaling because of the quadratic attention mechanism used over syndrome elements[37,71,74]. This makes pure Transformer architectures ill-suited with on-chip, low-latency implementation, without co-design of major hardware. It requires dedicated GPU or tensor processing unit (TPU) support to break even with other competitors[75,87]. The Mamba/SSM based decoder is an exciting new architecture that achieves Transformer-level accuracy with lower latency than comparable self-attention models and achieves a $p_{th} \approx 1.04\%$ decoding noise, implying that state-space models can occupy the latency gap between GNNs and Transformers[38,39].

Another aspect of hardware integration is that of physical proximity to the cryostat. Approximately, 15 percent of the included works point out that integrating the decoder into the syndrome measurement cycle of superconducting processors requires the classical inference engine to be located within centimeters to meters of the quantum chip, which adds strict thermal and electromagnetic requirements[73,101,104]. This has necessitated increasing interest in the areas of cryogenic CMOS and memristive neural decoder implementations. Ref. 73 shows that a memristive NN decoder can be run on cryogenic temperatures with a latency of the microsecond regime and LER performance on par with software MWPM, which is a considerable milestone towards complete quantum-classical integration[72,73]. Taken together, the latency results across the 108-study corpus reveal a clear trade-off between inference speed and decoding accuracy that varies by architecture. Fig. 6 plots this indicative latency-accuracy frontier for representative decoders, showing LER improvement over MWPM against decoding latency on a log scale, with the vertical dashed line at 1 μs marking the real-time threshold for superconducting qubit architectures.

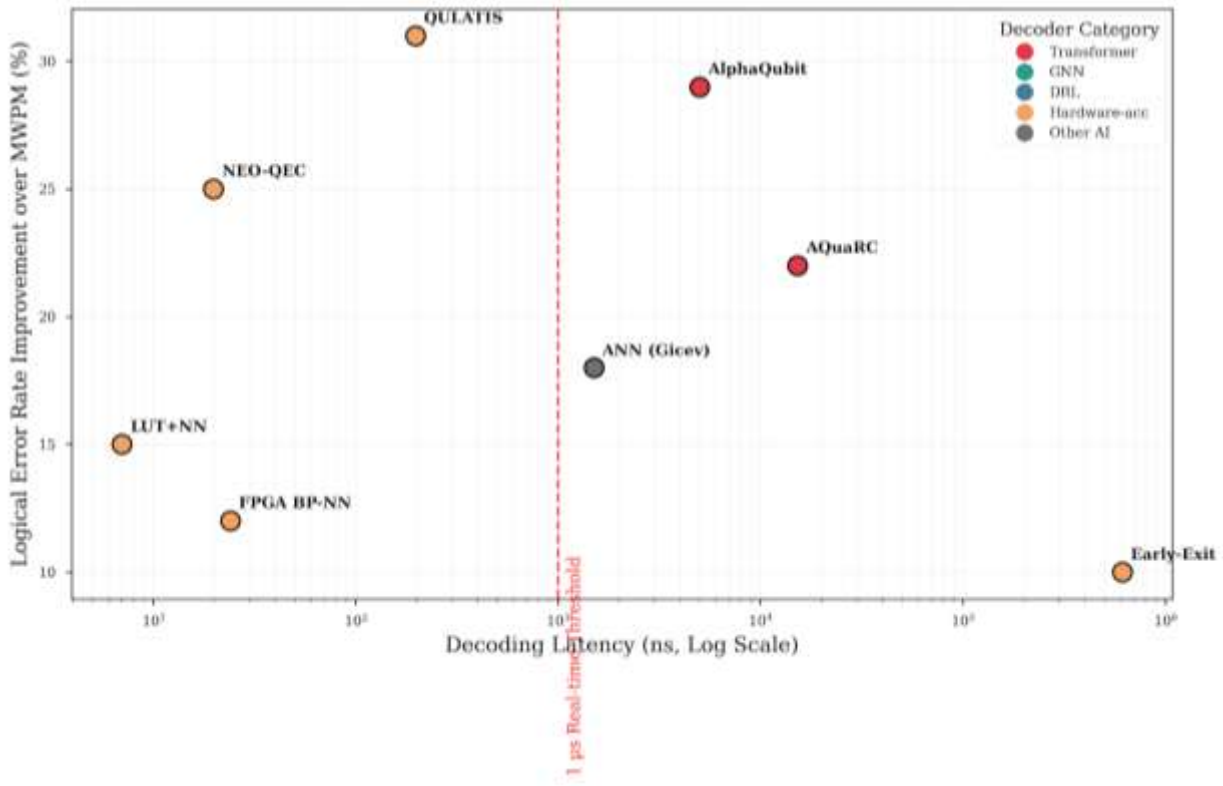


Fig. 6. Indicative latency-accuracy frontier of AI decoders. Logical error rate (LER) improvement over MWPM (%) plotted against decoding latency (nanoseconds, log scale) for representative systems from the 108-study corpus. Markers are color-coded by architecture (Transformer, GNN, DRL, Hardware-accelerated). The vertical dashed line indicates the 1 µs real-time decoding threshold for superconducting qubit architectures. Note that latency values originate from heterogeneous measurement contexts (FPGA hardware, software simulation, per-iteration vs. end-to-end) and LER improvement figures are drawn from studies with different noise models and code distances; this figure illustrates qualitative trends rather than a controlled benchmark. Figure generated by the authors from the systematic data extraction table.

D. Experimental Validation and Robustness

Another highly significant and comparatively recent trend in the field is the shift of entirely simulation-based benchmarking to experimental validation on physical quantum hardware. And around 15 percent of the 108 studies included (about 2023-2026 publications) benchmark AI decoders on syndrome measurements measured directly on operating quantum processors, such as Sycamore and Willow superconducting systems or heavy-hexagon architectures as built by IBM, or trapped-ion systems built by IonQ[37,43,56,58, 75,82].

The groundbreaking AlphaQubit study shows that a Transformer trained on synthetic Syndrome data with almost no fine-tuning on experimental Sycamore data can achieve 24-30% LER reduction compared to the best MWPM decoder on distance-3 and distance-5 surface codes[37]. This performance improvement is also explained by the fact that the AI can model some combination of so-called soft information: continuous-valued measurement outcomes and non-Markovian error correlations due to the presence of qubit crosstalk, leakage, and state-dependent measurement errors, which conventional MWPM is unable to use. This is then also generalized to

a distance-7 surface code on the Willow processor in the following Google Quantum AI scalable decoder paper with logical qubit lifetimes surpassing the break-even point, passing the threshold of demonstrated practical utility[75,76].

Ref. 56 experimentally demonstrate that an ANN decoder trained on IBM heavy-hexagon syndrome data can efficiently decode and achieve performance comparable to a minimum-weight perfect matching (MWPM) decoder on current IBM processors, thereby validating the applicability of ANN-based decoding on this architecture, although the devices' physical error rates remain above threshold and limit logical-error-rate suppression[56,57]. In the case of trapped-ion platforms, there are also fewer AI decoder validation studies because that architecture is not as experimentally mature as superconducting systems; this is a gap in the literature that could be addressed in future research.

One common and critical issue found in the experimental validation subset is calibration strength. Approximately 15 of the 108 studies included report direct evidence that AI decoders trained on a fixed noise distribution degrade in performance more rapidly than MWPM when hardware error rates shifts away from the training distribution[94,54]. That is a problem of distribution shift which is intrinsically related to a limitation of discriminative neural models: they can learn a particular mapping between syndrome patterns and corrections which is optimal to the noise profile used to train them, but sensitive to non-stationarities. Importantly, this degradation is larger than that of MWPM with the same drift, as the performance of MWPM is sensitive only to the edge weights of the matching graph, which can be updated in closed form using recalibration data, but retraining a neural decoder involves a full or partial new training cycle[83,84,94].

As a solution to this, there has emerged a scholarly literature suggesting so-called adaptive or online learning models where decoder weights are updated continuously in near-real-time as the hardware recalibrates. AQuaRC's framework, which uses a Transformer with reinforcement learning to allow adaptive decoders to adapt the policy with online syndrome observation, and which has been shown to restore almost the same performance as at the training time with only a few additional syndrome observations[36,59,78,98] confirms that adaptive decoders are capable of recreating the performance of the original policy with only a few additional syndrome observations. Similar applications include multi-agent RL calibration of decoders, which also show that real-time adaptation of decoder priors is done to follow a hardware drift[101,102]. These findings indicate that the paradigm of adaptive decoding models, as opposed to pre-trained models, should be the focus of dependable implementation on the production quantum hardware.

Generalization across distances, which is the ability of a model trained at a one distance to decode at larger distances without re-training, has been demonstrated for GNN and equivariant neural decoder architectures in several studies[60,89,90]. In perfect noise model scenarios, GNNs that are trained on distance-3 lattices have been shown to generalize to distance-11 codes without significant performance loss, owing to the size-independent locality of graph convolution operator[29,80,60]. This is less evident with Transformer-based models, which encode position-dependent attention patterns, which depend on lattice size sensitivity[37,79,91,95].

IV. DISCUSSION

The systematic synthesis of the 108 included studies leads to three principal conclusions. First, across the noise regimes most practically significant for real quantum hardware, AI-enhanced decoders demonstrate consistent performance advantages over classical baselines. The evidence is strongest, and the margin over MWPM is largest when noise is correlated, biased, or experimentally realistic: precisely the conditions under which the i.i.d. assumption underlying minimum-weight matching breaks down. Under idealized depolarizing noise, AI decoders still frequently match or exceed classical baselines, but the advantage is less uniform and more architecture-dependent[37,45,83,96]. The 20–30% per-distance improvement in LER suppression reported by GNN decoders in the sub-threshold regime of several included studies directly translates to potential reductions in the physical qubit count needed to reach a target logical error rate, which is a significant practical benefit, since qubit count is the dominant engineering cost of fault-tolerant quantum computing, though this figure varies across noise models and should not be interpreted as a universal guarantee[29,43,47]. Second, there has been a fundamental change in the character of decoding bottleneck. For first-generation AI decoders, the bottleneck was algorithmic: neural networks were slower than MWPM[99,100]. This has been solved comprehensively by hardware-accelerated implementations, which reach nanosecond inference times[68,70]. The new bottleneck is a systems-engineering problem: how to transfer syndrome data from the quantum processor to the classical inference engine within the coherence budget, without excessive power dissipation in the cryogenic environment[47,73,101]. This requires co-design of the AI decoder, the classical controller and the quantum hardware, an interdisciplinary challenge extending well beyond algorithmic design. Third, a robustness gap emerges only during experimental validation and is not seen in simulation benchmarks. AI decoders perform well when training and deployment noise distributions are well matched but degrade faster than MWPM when hardware drift shifts the distribution[83,92,35,94]. This implies that benchmarking AI decoders exclusively on simulated data risks overstating their practical utility, and that experimental validation on physical hardware should become a standard component of decoder evaluation. The adaptive learning frameworks that have emerged in the most recent literature offer a conceptually sound approach to addressing this challenge[36,77,78,97,101].

The literature reviewed has a few systematic methodological weaknesses which are worth noting. The most common is the ubiquity of the simplified or idealised noise models, in particular the i.i.d. depolarizing model, as the most important reference point, especially older literature. Although depolarizing noise is an easy analytical point of reference, it systematically underestimates the complexity of noise in experimental hardware, where crosstalk and leakage to non-computational states, $1/f$ charge noise, and measurement-error correlations generate a non-Markovian, non-stationary error landscape[14,15]. Research that is benchmarked on depolarizing noise only thus exaggerates the capabilities of AI decoders in practice. A second weakness is the sparsity of training data: as quantum systems grow to the hundreds or thousands of physical qubits, the amount of labeled syndrome data necessary to train them also grows, which is a prohibitive

data generation cost[16,81,83]. Several works employ Monte Carlo construction of training data, although Monte Carlo cannot reliably model the non-Markovian correlations of real hardware unless the device-specific noise models are very accurate[16,93,104]. The third weakness is the inability to apply reporting conventions consistently across the corpus because a significant proportion of the studies do not report uncertainty estimates and LER or threshold values, and hardware-specific latency information are often not provided, which makes quantitative across-study comparison challenging. These gaps highlight a pressing need for standardized benchmarking protocols across the field.

Several priority directions are identified for future research. The most promising direction in the near term is the creation of physics-informed neural decoders, where the symmetry constraints, the code architecture of stabilizers, and hardware noise priors are incorporated directly into the neural architecture[27,28,60]. Already equivariant neural architectures, which represent the structural inductive biases of the spatial symmetries of the code lattice, are showing improved cross-distance generalization than unconstrained models[60,88,90,103]. Online and adaptive learning systems with the capability to perform recalibration of the decoder in real-time whenever the hardware starts drifting, must be regarded as a design necessity and not as an option[36,78]. Co-designing hardware between AI decoders and the classical control stack of the quantum processor must be a first-class research: the reviewed implementations of the memristive and SFQ-based neural decoders demonstrate that cryogenic-compatible AI inference can be done[68,73], but the thermal cost and electromagnetic costs of millikelvin operation pose fundamental limits on classic computing density which new paradigms, such as neuromorphic chip and in-memory computing hardware architecture, can potentially overcome[73,105]. It should also be open to the open-source syndrome datasets based on physical quantum processors that would permit independent reproducibility and comparable cross-architecture comparison, similar to the role of ImageNet in computer vision[47,94]. Lastly, hybrid AI-classical architectures between the high-latency local neural decoders and the slower yet globally optimal classical decoders will probably be the most realistic near-term deployment strategy and will provide better performance than the sum of its parts, and have real-time latency[6,28,86]. Fig. 7 maps this forward-looking trajectory visually, summarizing the three-stage roadmap from NISQ-era static decoders (Stage 1: physics-informed architectures) through hardware-integrated real-time systems (Stage 2: adaptive real-time decoding) to fully co-designed hybrid systems capable of dynamic noise adaptation (Stage 3: hardware-software co-design), providing a structured outlook for the development of utility-scale AI-enhanced QEC.

This systematic review of peer-reviewed articles and preprints spanning 2017 to 2026 provides the most comprehensive evaluation to date of AI-enhanced decoding for topological QEC codes. Across the reviewed corpus, AI decoders, especially GNNs, Transformer-based approaches, and DRL agents frequently demonstrate superior LER suppression over classical algorithmic baselines, with the advantage most pronounced under correlated, biased, and hardware-realistic noise. Fixed-architecture hardware-accelerated implementations can achieve nanosecond-regime

inference latencies compatible with the cycle times of current quantum processors, though scalable architectures such as Transformers retain input-size-dependent costs that must be managed through hardware co-design. Among the reviewed architectures, GNNs offer the best-evidenced combination of accuracy, scalability, and cross-distance generalization for near-term deployment, though the appropriate choice of architecture depends on the specific code family, noise regime, and hardware constraints. Models using transformers provide the most accurate results on experimental hardware at the expense of increased latency and hardware use. DRL-based decoders show particular advantages in correlated and biased noise regimes, though latency remains a challenge requiring further optimization[106,107,108]. The transition from purely simulation-based benchmarking to experimental validation on physical hardware, as shown by AlphaQubit's work with Sycamore data and Google Quantum AI's Willow processor demonstration, marks a key advancement in the field. Progress toward utility-scale fault-tolerant quantum computing will require close co-design of AI decoders with quantum hardware and classical control electronics, development of physics-informed and symmetry-equivariant neural architectures, and establishment of community-wide benchmarking standards[109,110]. This review provides a systematic evidence base and a prospective roadmap for that endeavor.

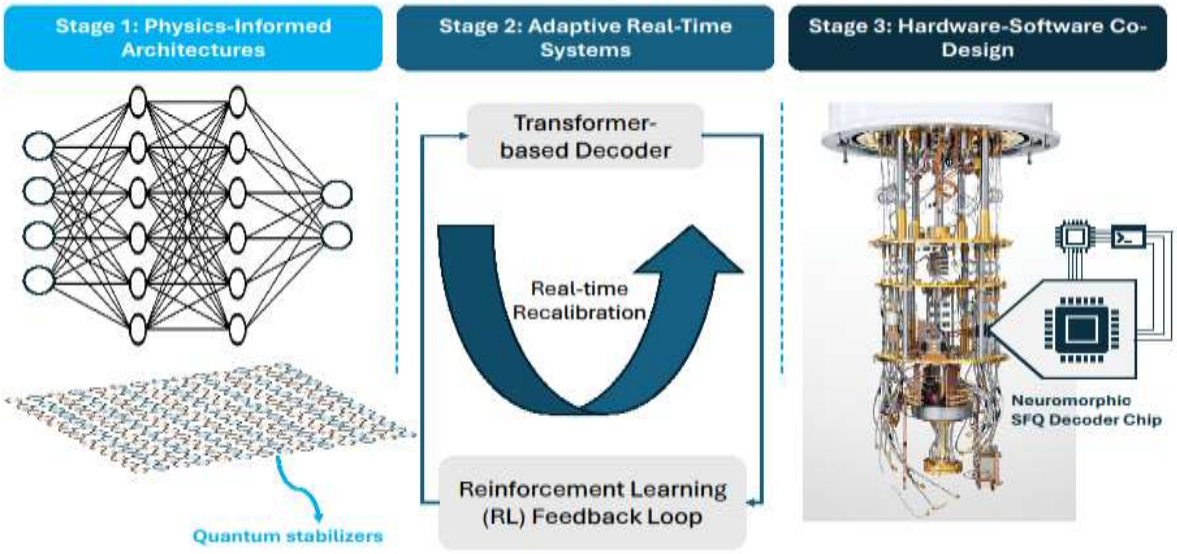


Fig. 7. Evolutionary roadmap for AI-enhanced QEC decoders. The trajectory from initial NISQ-era static decoders toward utility-scale fault-tolerant deployment highlights the shift from purely simulation-based training (Stage 1: physics-informed architectures) to hardware-integrated real-time decoding (Stage 2: adaptive real-time systems), culminating in adaptive, physics-informed, and co-designed hybrid systems capable of dynamic noise adaptation (Stage 3: hardware-software co-design). Figure created by the authors; stage structure informed by the evidence synthesis and consistent with the roadmap framework outlined in ref. 47.

DATA AVAILABILITY

The primary data supporting the findings of this systematic review are derived from the 108 peer-reviewed articles and preprints identified through the systematic search strategy detailed

in Section II. The comprehensive data extraction table, which includes categorized information on AI architectures, topological code types, noise models, performance metrics, and latency benchmarks for all 108 sources is provided as Supplementary Material. Further inquiries regarding the dataset or screening criteria may be directed to the corresponding author upon reasonable request.

CODE AVAILABILITY

Not applicable. This is a systematic review, and no original code was developed as part of this study.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the Quantum Computing and Assistive Technologies Laboratory at the Kwame Nkrumah University of Science and Technology (KNUST) for providing the research environment and infrastructure that supported this work. The authors also acknowledge the use of Covidence for systematic screening and data management, and Harzing's Publish or Perish for database search execution.

AUTHOR CONTRIBUTIONS

E.A. designed the review protocol and conducted the systematic search. E.A. performed the screening of titles/abstracts, full-text review, data extraction, and evidence synthesis. J.A., J.K. and G.B. provided oversight into the search strategy and validated the inclusion and exclusion criteria. All authors contributed to the critical revision of the manuscript for intellectual content, provided feedback at all stages of the review process, and approved the final version for submission.

COMPETING INTERESTS

The authors declare no competing interests.

REFERENCES

1. J. Preskill, "Quantum Computing in the NISQ era and beyond," Quantum, vol. 2, p. 79, Aug. 2018.
2. F. Arute et al., "Quantum supremacy using a programmable superconducting processor," Nature, vol. 574, no. 7779, pp. 505–510, 2019.
3. J. Kelly et al., "State preservation by repetitive error detection in a superconducting quantum circuit," Nature, vol. 519, no. 7541, pp. 66–69, 2015.
4. B. Terhal, "Quantum error correction for quantum memories," Rev. Mod. Phys., vol. 87, no. 2, p. 307, 2015.
5. D. Gottesman, "An introduction to quantum error correction and fault-tolerant quantum computation," Proc. Sympos. Appl. Math., vol. 68, pp. 13–58, 2010.
6. A. G. Fowler, M. Mariantoni, J. M. Martinis, and A. N. Cleland, "Surface codes: Towards practical large-scale quantum computation," Phys. Rev. A, vol. 86, no. 3, p. 032324, 2012.
7. A. Y. Kitaev, "Fault-tolerant quantum computation by anyons," Ann. Phys., vol. 303, no. 1, pp. 2–30, 2003.

8. E. Dennis, A. Kitaev, A. Landahl, and J. Preskill, "Topological quantum memory," J. Math. Phys., vol. 43, no. 9, pp. 4452–4505, 2002.
9. Google Quantum AI and Collaborators, "Quantum error correction below the surface code threshold," Nature, vol. 638, no. 8052, pp. 920–926, Feb. 2025, doi: 10.1038/s41586-024-08449-y.
10. A. Pesah, "Decoders for Topological Quantum Error Correction," University College London, Tech. Rep., 2020.
11. N. Delfosse and N. H. Nickerson, "Almost-linear time decoding of topological codes via process-based compression," Quantum, vol. 5, p. 595, 2021.
12. J. Edmonds, "Paths, trees, and flowers," Can. J. Math., vol. 17, pp. 449–467, 1965.
13. M. Suchara et al., "Comparing the complexity of decoder circuits," arXiv preprint arXiv:1312.2313, 2013.
14. P. Sarvepalli and K. R. Brown, "Decoders for topological codes on bits with leakage," Phys. Rev. A, vol. 86, no. 4, p. 042336, 2012.
15. C. Chamberland et al., "Building a fault-tolerant quantum computer using error-corrected qubits," Nature, vol. 605, pp. 60–65, 2022.
16. H. Cao et al., "Artificial Intelligence for Quantum Error Correction: A Comprehensive Review," arXiv preprint arXiv:2412.20380, 2024.
17. G. Torlai and R. G. Melko, "Neural decoder for topological codes," Phys. Rev. Lett., vol. 119, no. 3, p. 030501, 2017.
18. S. Varsamopoulos et al., "Designing neural network based decoders for surface codes," arXiv:1811.12456, 2018.
19. J. Bausch et al., "Learning to decode the surface code with a recurrent, transformer-based neural network," arXiv preprint arXiv:2310.05900, Oct. 2023 (published as Nature 635, 834–840, 2024).
20. P. Andreasson et al., "Quantum error correction for the toric code using deep reinforcement learning," Quantum, vol. 3, p. 183, 2019.
21. S. Krastanov and L. Jiang, "Deep Neural Network Probabilistic Decoder for Stabilizer Codes," Sci. Rep., vol. 7, no. 1, p. 11003, 2017.
22. J. Bausch et al., "Learning high-accuracy error decoding for quantum processors," Nature, vol. 635, no. 8040, pp. 834–840, Nov. 2024, doi: 10.1038/s41586-024-08148-8.
23. M. Lange et al., "Data-driven decoding of quantum error correcting codes using graph neural networks," Phys. Rev. Research, vol. 7, p. 023181, 2025, doi: 10.1103/PhysRevResearch.7.023181.
24. P. Baireuther et al., "Machine-learning-assisted correction of correlated qubit errors in a topological code," Quantum, vol. 2, p. 48, 2018.
25. S. Varsamopoulos et al., "Comparing Neural Network Based Decoders for the Surface Code," IEEE Trans. Comput., vol. 69, no. 2, pp. 300–311, 2020.
26. P. Baireuther et al., "Neural network decoder for topological color codes with circuit level noise," New J. Phys., vol. 21, no. 013003, 2019.
27. Y. Tomita and K. M. Svore, "Low-distance surface codes under realistic quantum noise," Phys. Rev. A, vol. 90, p. 062320, 2014.
28. H. Liu et al., "Symmetry-enhanced neural network decoders for the surface code," arXiv preprint arXiv:2310.04561, 2023.
29. S. Gicev, L. C. L. Hollenberg, and M. Usman, "A scalable and fast artificial neural network syndrome decoder for surface codes," Quantum, vol. 7, p. 1058, 2023.
30. N. Delfosse et al., "Hierarchical Graph Neural Networks for Large-Scale QEC," IonQ Research, 2025.
31. H. Xiang, Y. Yang, G. He, J. Huang, and D. He, “Scalable and Real-Time Neural Decoder for Topological Quantum Codes,” arXiv:2512.07737, Dec. 2025.
32. N. H. Ji et al., "Recurrent neural network decoding of rotated surface codes based on distributed strategy," Chin. Phys. B, vol. 33, p. 040307, 2024.

33. D. Fitzek et al., "Deep Q-learning decoder for depolarizing noise on the toric code," Phys. Rev. Research, vol. 2, p. 023230, 2020.
34. E. C. Peterson, “Techniques for Combining Fast Local Decoders with Global Matching for the Surface Code,” arXiv preprint, Jul. 2023.
35. M. Usman et al., "Machine learning for continuous quantum error correction on superconducting qubits," New J. Phys., 2022.
36. S. R. M. S. R. M. Shibah and S. Rashid, "Adaptive Reinforcement Learning for Autonomous Quantum Error Correction in Surface Codes," SSRN preprint, 2025.
37. A. Davaasuren et al., "General framework for constructing fast and near-optimal machine-learning-based decoder of the topological stabilizer codes," Phys. Rev. Research, vol. 2, p. 033399, 2020.
38. T. Fan et al., "Transformer-based quantum error decoding enhanced by QGANs," EPJ Quantum Technology, 2025.
39. Google Quantum AI, "Scalable Neural Decoders for Practical Real-Time Quantum Error Correction," arXiv:2510.22724, 2025.
40. H. Wang et al., "Global receptive field transformer decoder method on quantum surface code," Chin. Phys. B, 2025.
41. A. A. A. A. Jadhav et al., "AI for Quantum Error Correction," IRE Journals, vol. 8, no. 9, 2025.
42. K. K. Lohani et al., "Artificial intelligence for quantum computing: a review," PMC/NIH preprint, 2025.
43. R. Acharya et al., "Suppressing quantum errors by scaling a surface code logical qubit," Nature, vol. 614, pp. 676–681, 2023.
44. S. Varsamopoulos et al., "Decoding surface code with a distributed neural network–based decoder," Quantum Mach. Intell., 2020.
45. X. Xiao et al., "Graph neural network with multi-head attention mechanism and multi-task learning for XZZX code," Chin. Phys. B, 2025.
46. N. H. Ji et al., "Neural network-based decoding for bias-tailored quantum codes," Commun. Theor. Phys., 2025.
47. Riverlane, "The Quantum Error Correction Report 2025," Riverlane Publishing, 2025.
48. P. P. Ronagh et al., "Impacts of Decoder Latency on Utility-Scale Quantum Computer Architectures," ResearchGate preprint, 2025.
49. Y. Ueno et al., "NEO-QEC: Neural network enhanced online superconducting decoder for surface codes," arXiv:2208.05758, 2022.
50. K. Meinerz et al., "Scalable Neural Decoder for Topological Surface Codes," Phys. Rev. Lett., vol. 128, p. 080505, 2022.
51. B. M. Varbanov et al., "Neural network decoder for near-term surface-code experiments," Phys. Rev. Research, 2025.
52. H. Wang et al., "Hierarchical Qubit-Merging Transformer for Quantum Error Correction," arXiv:2510.11593, 2025.
53. M. Sheth et al., "Neural ensemble decoding for topological quantum error-correcting codes," Phys. Rev. A, vol. 101, p. 032338, 2020.
54. S. Bravyi, M. Suchara, and A. Vargo, “Analysing Correlated Noise on the Surface Code Using Adaptive Decoders,” Quantum, vol. 3, p. 131, Apr. 2019.
55. L. D. Colomer et al., "Reinforcement learning for optimal error correction of toric codes," Phys. Lett. A, vol. 384, p. 126353, 2020.
56. B. Hall et al., "Artificial neural network syndrome decoding on IBM quantum processors," Phys. Rev. Research, vol. 6, L032004, 2024.
57. R. Sweke et al., "Reinforcement learning decoders for fault-tolerant quantum computation," Mach. Learn.: Sci. Technol., vol. 2, p. 025005, 2021.

58. N. P. Breuckmann and X. Ni, "Scalable Neural Network Decoders for Higher Dimensional Quantum Codes," Quantum, vol. 2, p. 68, 2018.
59. H. Théveniaut and E. van Nieuwenburg, "A NEAT Quantum Error Decoder," SciPost Phys., vol. 11, p. 005, 2021.
60. E. Egorov et al., "The END: An Equivariant Neural Decoder for Quantum Error Correction," arXiv:2304.07362, 2023.
61. Y. Jahan et al., "Neural Minimum Weight Perfect Matching for Quantum Error Codes," arXiv:2601.00242, 2026.
62. I. Convy et al., "Machine learning for continuous quantum error correction on superconducting qubits," New J. Phys., 2022.
63. E. S. Matekole et al., "Decoding surface codes with deep reinforcement learning and probabilistic policy reuse," arXiv:2212.11890, 2022.
64. R. W. J. Overwater, M. Babaie, and F. Sebastiano, "Neural-Network Decoders for Quantum Error Correction Using Surface Codes: A Space Exploration of the Hardware Cost-Performance Tradeoffs," IEEE Trans. Quantum Eng., vol. 3, pp. 1–19, Jan. 2022, doi: 10.1109/tqe.2022.3174017.
65. S. B. Iyer and S. Dey, "Machine-Learning Based Decoding of Surface Code Syndromes in a Quantum Error Correction Scheme," J. Electronics and New Renewable Systems, vol. 1, no. 6, pp. 4–15, Jun. 2022.
66. S. Maes, "Deep Reinforcement Learning for Quantum Error Correction," M.S. thesis, Chalmers Univ. Technol., Gothenburg, Sweden, 2023.
67. N. Delfosse and G. Zémor, "Linear-time maximum likelihood decoding of surface codes," Phys. Rev. A, 2020.
68. T. J. Hu et al., "Lookup Table and Neural Network Based Decoders for Real Time Quantum Error Correction," IEEE Trans. Quantum Eng., 2024.
69. M. Zhang et al., "MTLND: Multi-Task Learning Neural Decoder for High-Speed QEC," arXiv:2305.15767, 2023.
70. R. Huo and J. Nunez-Yanez, "FPGA-Accelerated Early-Exit Neural Decoder for Quantum Error Correction," Euromicro, 2025.
71. Y. Choukroun and L. Wolf, "Deep Quantum Error Correction," in Proc. 38th AAAI Conf. Artif. Intell., vol. 38, no. 10, pp. 11332–11340, Mar. 2024, doi: 10.1609/aaai.v38i10.28987.
72. Y. Ueno et al., "QULATIS: A Quantum Error Correction Methodology toward Low-Latency and High-Throughput," HPCA, 2022.
73. V. Yon, "A memristive neural decoder for cryogenic fault-tolerant quantum error correction," Quantum Sci. Technol., vol. 10, 2025.
74. A. K. B. Sung et al., "Scalable Neural Decoders for Practical Real-Time Quantum Error Correction," arXiv preprint arXiv:2510.22724, Oct. 2025.
75. A. W. Senior et al. (Google Quantum AI), "Scalable and real-time neural decoder for topological quantum codes," arXiv:2512.07737, 2025.
76. K. Hammar et al., "Error-rate-agnostic decoding of topological stabilizer codes," Phys. Rev. A, vol. 105, p. 042616, 2022.
77. R. Sweke et al., "Local Clustering Decoder as a fast and adaptive hardware decoder for quantum error correction," Nat. Commun., vol. 16, Dec. 2025.
78. J. Bausch, "Real-time adaptive quantum error correction by model-free multi-agent reinforcement learning," arXiv preprint arXiv:2509.03974, Sep. 2025.
79. B. Srivastava et al., "Decoding topological XYZ² codes with reinforcement learning based on attention mechanisms," Quantum, vol. 6, p. 698, 2022.
80. A. T. Suresh et al., "Data-Driven Decoding of Quantum Error Correcting Codes Using Graph Neural Networks," arXiv preprint arXiv:2307.01241, Mar. 2025.

81. J. Gama et al., “Fully Convolutional 3D Neural Network Decoders for Surface Codes with Fast Syndrome Simulation,” arXiv preprint arXiv:2506.16113, Jun. 2025.
82. J. R. Mcclean, Z. Jiang, N. C. Rubin, R. Babbush, and H. Neven, “Decoding quantum errors with subspace expansions,” Nat Commun, vol. 11, no. 1, Jan. 2020, doi: 10.1038/s41467-020-14341-w.
83. L. Postler et al., “Sample Importance for Data-Driven Decoding,” arXiv preprint arXiv:2505.22741, May 2025.
84. Y. Zhou, Z. Wang, and M. Martonosi, “DGR: Tackling Drifted and Correlated Noise in Quantum Error Correction with Decoding Graph Re weighting,” arXiv preprint arXiv:2311.16214, Nov. 2023.
85. I. A. Convy et al., "Machine learning for continuous quantum error correction on superconducting qubits," New J. Phys., 2022.
86. H. Zhao et al., “Neural Minimum Weight Perfect Matching for Quantum Error Codes,” arXiv preprint arXiv:2601.00242, Jan. 2026.
87. S. Vittal et al., “TT-QEC: Transferable Transformer for Quantum Error Correction of Surface Codes,” in Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD), 2023, pp. 1–9.
88. Q. T. Nguyen et al., “Theory for Equivariant Quantum Neural Networks,” PRX Quantum, vol. 5, no. 2, May 2024, doi: 10.1103/prxquantum.5.020328.
89. M. Maskara et al., "Advantages of versatile neural-network decoding for topological codes," Phys. Rev. A, vol. 99, p. 052351, 2019.
90. S. Huang, M. Newman, and K. R. Brown, “Fault-tolerant weighted union-find decoding on the toric code,” Phys. Rev. A, vol. 102, no. 1, July 2020, doi: 10.1103/physreva.102.012419.
91. S. Yao, Z. Gao, Y. Zhang, and H. Xie, “Transformer QEC: Quantum Error Correction Code Decoding Using Transformer,” arXiv preprint arXiv:2311.16082, Nov. 2023.
92. O. Higgott, “PyMatching: A Python Package for Decoding Quantum Codes with Minimum-Weight Perfect Matching,” ACM Transactions on Quantum Computing, vol. 3, no. 3, pp. 1–16, June 2022, doi: 10.1145/3505637.
93. A. C. Gidney, “Stim: a fast stabilizer circuit simulator,” Quantum, vol. 5, p. 497, Jul. 2021, doi: 10.22331/q-2021-07-06-497.
94. J. Cai et al., “Neural Network Decoder for Near-Term Surface-Code Experiments,” Physical Review Research, vol. 7, no. 1, Art. no. 013029, Jan. 2025.
95. T. Tanimoto et al., "Qubit-centric Transformer for Surface Code Decoding," arXiv preprint arXiv:2510.11593, Oct. 2025.
96. M. Andreasson et al., “Deep Q-Learning Decoder for Depolarizing Noise on the Toric Code,” Physical Review Research, vol. 2, no. 2, Art. no. 023230, May 2020.
97. C. Chamberland et al., “Calibrated Decoders for Experimental Quantum Error Correction,” Phys. Rev. Lett., vol. 128, no. 11, Art. no. 110504, Mar. 2022.
98. J. Nöbauer, P. Rabl, and P. Schindler, “Optimizing Quantum Error Correction Codes with Reinforcement Learning,” Quantum, vol. 3, p. 215, Dec. 2019.
99. K. Varsamopoulos, B. Criger, and K. Bertels, "Deep neural network probabilistic decoder for stabilizer codes," Sci. Rep., vol. 7, Art. no. 11224, Sep. 2017.
100. N. P. Breuckmann and V. Londe, "Neural ensemble decoding for topological quantum error-correcting codes," Phys. Rev. A, vol. 101, no. 3, Art. no. 032338, Mar. 2020.
101. G. Dauphinais et al., "Quantum error correction with the semion code," New J. Phys., vol. 21, p. 053035, 2019.
102. F. Battistel et al., “Real-time decoding for fault-tolerant quantum computing: progress, challenges and outlook,” Nano Futures, vol. 7, no. 3, p. 032003, Aug. 2023, doi: 10.1088/2399-1984/aceba6.
103. M. S. Wang et al., "Efficient near-optimal decoding of the surface code through reinforcement learning," arXiv preprint arXiv:2401.12434, Mar. 2024.

104. J. P. Gaur et al., “Classical Interfaces for Controlling Cryogenic Quantum Computing Systems,” Applied Physics Quantum, vol. 2, no. 4, Art. no. 041501, Nov. 2025.
105. C. Chinni et al., "Neural decoder for topological codes using pseudo-inverse of parity check matrix," arXiv:1901.07535, 2019.
106. H. Liu et al., "Symmetry-enhanced neural network decoders," arXiv:2310.04561, 2023.
107. H. Cao et al., "qecGPT: Decoding Quantum Error-Correcting Codes with Generative Pre-trained Transformers," arXiv:2307.09025, 2023.
108. B. Srivastava et al., "Decoding topological XYZ² codes with reinforcement learning based on attention mechanisms," Quantum, vol. 6, p. 698, 2022.
109. I. A. Convy et al., "Machine learning for continuous quantum error correction on superconducting qubits," New J. Phys., 2022.
110. H. W. Wang et al., "Determining quantum topological semion code decoder performance and error correction effectiveness with reinforcement learning," Front. Phys., vol. 10, 981225, 2022.

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- SupplementaryTableDataExtraction.xlsx