Supplementary Methods: SAGE-QEC Defensible Selective Decoder

S1. Software environment

The SAGE-QEC campaign used Python with Stim for stabilizer-circuit sampling, PyMatching for the MWPM baseline, NumPy for numerical processing, and PyTorch for the candidate ranker. The reference implementation is `AQEC_SAGE_QEC.py`. Exact package versions, random seeds, command lines, and output summaries should be frozen in the final repository before submission.

S2. Circuit and detector records

Each condition uses Stim's rotated surface-code memory circuit with five measurement rounds. The circuit produces a detector record and a logical observable record. The detector record is the only inference-time observation supplied to either decoder. The logical observable is retained by the evaluator and is not supplied to SAGE-QEC during routing, candidate generation, or policy selection.

S3. Data partitioning

For each distance, nominal circuit parameter, and seed, three known regimes are sampled for training and validation: Independent, Measurement, and Burst-2. Four regimes are reserved for blind evaluation: Burst-3, Temporal Drift, Crosstalk, and Mixed Noise. The regime name is not passed to the model. Training and validation use independent simulator draws and seeds.

The blind campaign contains 45 distance-parameter-seed combinations. Each combination contains 450 shots in each of four blind regimes. This yields 1,800 blind shots per combination and 81,000 blind shots overall.

S4. Baseline construction

PyMatching is constructed from the circuit detector error model at the nominal condition. Graphlike decomposition is used for the baseline because the matching solver requires error mechanisms with one or two detection events. The baseline prediction is computed from the same detector record supplied to SAGE-QEC.

The principal qualification gap is that this baseline is not yet a fully calibrated space-time PyMatching model for every structured correlated test regime. A follow-up study must build that

matched baseline and repeat the comparison.

S5. Features and candidate ranker

SAGE-QEC computes detector density, standard deviation, temporal block occupancy, temporal adjacency, connected temporal activity, persistence between halves of the record, and disagreement signals from a bank of PyMatching models with nearby nominal parameters. The ranker receives the resulting feature vector, candidate indicators, and the nominal PyMatching output.

The ranker is a small multilayer perceptron trained with cross-entropy over the finite candidate index. Its purpose is to rank available candidates. It does not create a physical correction chain and it does not receive the true logical outcome.

S6. Policy selection

Confidence, top-two margin, and an out-of-distribution score are selected using validation data. They are frozen before blind testing. A test shot is overridden only if the selected candidate satisfies the policy. Otherwise the system retains the PyMatching decision.

S7. Metrics

The primary metric is logical error rate:

$$LER = N^{-1} \sum_{i} 1[\hat{\ell}_i \neq \ell_i].$$

The relative reduction is:

$$R = (LER_{PM} - LER_{SAGE-QEC})/LER_{PM}.$$

Secondary metrics are override rate, correct overrides, false overrides, override precision, abstention rate, latency, and memory. The evaluator computes these only after inference.

S8. Paired bootstrap

For each bootstrap replicate, the evaluator resamples shot indices with replacement from the paired test records. It computes the mean SAGE-QEC-minus-PyMatching LER difference. The 2.5th and 97.5th percentiles across 800 replicates form the reported interval. A negative interval favors SAGE-QEC.

S9. Noise generators

Burst-3 creates a localized structured event over three early temporal slices. Temporal Drift varies the detector-flip rate between shots. Crosstalk flips a selected detector and neighboring positions. Mixed Noise combines measurement-like flips and crosstalk-like events. These generators are explicit simulator stress tests and should not be described as hardware calibration models.

S10. Required follow-up experiment

The next experiment must replace logical-output proxy candidates with complete physical correction hypotheses. Each candidate must be checked against the parity constraint $Hc = s$ modulo two. Candidate families should include local re-pairing, boundary alternatives, temporal chains, and correlated multi-detector mechanisms extracted from a declared detector error model.

The follow-up must compare nominal PyMatching, matched calibrated space-time PyMatching, local search without machine learning, and the complete proposed decoder. It must impose a predeclared intervention budget and report precision-coverage curves. Test parameters, seeds, and challenge cases must be frozen before execution.

S11. Limitations

The current SAGE-QEC result is simulator-based. It uses a finite candidate set. It does not establish a hardware advantage. It does not establish quantum advantage. It does not establish universal superiority over MWPM. Its high intervention rate is a material deployment limitation.

S12. Supplementary figure legends

Extended Data Fig. 1 | Information boundary of the benchmark. The matrix identifies which regimes are available during training and validation and which are withheld for blind evaluation. The withheld regimes are not passed to the router as metadata.

Extended Data Fig. 2 | Paired bootstrap intervals. Each point is the paired SAGE-QEC-minus-PyMatching logical-error difference for one distance-by-regime aggregate. Horizontal bars are the 2.5th and 97.5th percentiles of 800 shot-level paired bootstrap replicates. The zero line denotes equal performance.

Extended Data Fig. 3 | Override operating characteristics. Each point represents one distance-by-regime aggregate. The x-axis is the fraction of shots on which SAGE-QEC changes the nominal PyMatching decision. The y-axis is the precision of those changes. The plot is descriptive and is not a deployment calibration curve.

Extended Data Fig. 4 | Executed rescue–harm audit. The figure reports the exact counts printed by the reproducibility run. Positive bars are baseline failures rescued by SAGE-QEC; negative bars are harmful overrides. The accompanying intervals and heatmap are computed from the same paired blind cells.

Extended Data Fig. 5 | Noise taxonomy. The figure separates the four nominal Stim circuit-level channels from the post-sampling benchmark perturbations. This distinction prevents the controlled Burst, Drift, Crosstalk and Mixed stress regimes from being misreported as direct hardware noise characterization.

Extended Data Fig. 6 | Per-cell audit. Each panel reports PyMatching LER, SAGE-QEC LER, relative reduction, override rate and override precision for one of the twelve distance-by-regime cells.

S13. Supplementary table legends

Supplementary Table 1 | Condition registry. One row per distance, nominal parameter, seed, and regime, including the sampler seed, number of shots, detector count, and logical-observable count.

Supplementary Table 2 | Decoder configuration. PyMatching graph construction, decomposition settings, SAGE-QEC feature dimensions, candidate count, ranker architecture, and frozen routing thresholds.

Supplementary Table 3 | Paired outcome counts. Shared successes, shared failures, rescues, harmful overrides, override rate, override precision, and logical error rate for every aggregate cell.

Supplementary Table 4 | Audit results. Tests for label leakage, test-time threshold fitting, split contamination, deterministic rerun, syndrome consistency, and candidate validity.

S14. Nature-style reporting summary

The manuscript reports the primary endpoint before secondary operational endpoints. Main figures show the architecture and blind results. Extended data separates the information boundary and diagnostics from the headline result. The supplementary audit explicitly records what is measured, what is not measured, and which future experiments are required before an industrial claim. This separation is intentional: it prevents prospective targets, simulator-only stress tests, and hardware claims from being conflated.