Hardware-Aware Magic-State Injection on the Rotated Surface Code
High-Level Idea, Theoretical Model, and Numerical Simulation Plan

Draft research outline

February 16, 2026

Abstract

This document outlines a Physical Review A–level research direction centered on hardware-aware magic-state injection for the rotated surface code. The project extends Li's post-selected encoding protocol on the regular surface code and Lao–Criger's magic-state injection schemes on the rotated surface code to realistic, biased, and connectivity-constrained noise models. We describe (i) the high-level research idea and its relation to key references, (ii) a concrete theoretical model design including circuit-level noise and protocol parameterization, and (iii) a detailed plan for stabilizer-based numerical simulations on CPU and GPU backends.

Contents

1 High-Level Research Idea and Key References

1.1 Motivation: Reducing Magic-State Overhead in Surface-Code Architectures

Fault-tolerant surface-code architectures require a continual supply of high-fidelity magic states to implement non-Clifford gates such as the T gate via gate teleportation. The conventional approach prepares "raw" encoded magic states with relatively poor fidelity and then applies one or more rounds of magic-state distillation, which dominates the qubit-time overhead in many architectures.

The central idea of this project is to *optimize magic-state injection itself* on the rotated surface code, under realistic hardware constraints, so that the logical error rate p_L of the raw injected states is reduced and their logical noise may be biased in a favorable way. This directly lowers the cost of subsequent magic-state distillation, potentially saving orders of magnitude in qubit-time volume.

Concretely, we will:

- Define an architecture-specific circuit-level noise model where single-qubit gates and measurements are cheaper than CNOTs, and include possible noise bias and limited-range connectivity.
- Treat the injection patch geometry, stabilizer-measurement order, and number of post-selection rounds as optimization variables.
- Analytically count first-order error patterns that survive post-selection to obtain closed-form expressions for p_L as functions of hardware parameters.
- Validate and refine these expressions via stabilizer-based numerical simulations at the circuit level, and propagate the resulting p_L and acceptance probabilities into standard magic-state distillation cost models.

The result will be a family of hardware-optimized injection schemes on the rotated surface code (and potentially XZZX-rotated variants) with quantitatively demonstrated advantages over existing protocols.

1.2 Li (2015): Post-Selected Encoding on the Regular Surface Code

Li's protocol [1] provides the key conceptual building blocks: a two-phase, post-selected encoding procedure on the regular surface code, together with a circuit-level noise analysis.

Protocol structure. A single-qubit magic state $|m\rangle$ is initialized on a physical qubit in the top-left corner of a distance-d_1 surface-code patch. The remaining qubits in the small patch are initialized in $|0\rangle$ or $|+\rangle$ such that the target logical state after encoding is $|m\rangle_L$. Two rounds of stabilizer measurements are performed, using carefully ordered CNOT circuits. Runs are discarded (post-selected out) whenever the measured stabilizers are inconsistent with the initialization pattern or with the previous round's measurement outcomes.

If no syndrome is detected in this first phase, the code distance is then grown from d_1 to a larger d_2 by initializing additional data qubits and performing one or more rounds of full-lattice stabilizer measurements. After this second phase, the logical magic state is protected by standard surface-code error correction.

Circuit-level noise model. Li assumes independent depolarizing noise after each operation:

- Qubit initialization in $|0\rangle$ is flipped with probability p_I.
- Measurements in the computational basis are flipped with probability p_M.
- Single-qubit gates suffer depolarizing noise with probability p_1: a random non-identity Pauli acts with probability $p_1/3$.
- Two-qubit CNOT gates suffer two-qubit depolarizing noise with probability p_2: a random non-identity two-qubit Pauli acts with probability $p_2/15$.

Leading-order logical error rate. Under this model, Li analytically counts single-fault events that both (i) are not detected by post-selection and (ii) result in a logical error on the encoded magic state. Neglecting higher-order contributions, the resulting logical error rate has the form

$$p_L \approx \frac{2}{5}p_2^2 + 2p_I + \frac{2}{3}p_1 + \dots, \tag{1}$$

where the dots denote subleading terms in p_2 and contributions from weight-2 or higher faults. In the idealized limit where single-qubit gates, initialization, and measurement are perfect ($p_1 = p_I = p_M = 0$), Li finds that

$$p_L \to \frac{2}{5}p_2^2 \tag{2}$$

as $p_2 \to 0$, a result confirmed by stabilizer-based simulations.

A notable conclusion is that with realistic parameters where $p_1 \ll p_2$, the infidelity of the encoded magic state can be significantly *lower* than p_2, even though the encoding circuit uses many CNOT gates.

Impact on distillation overhead. Li couples the analytic expression for p_L to standard 15-to-1 Bravyi–Kitaev magic-state distillation, estimating the number of raw magic states and qubit-time volume required to reach a target logical error rate per output state. Improved raw magic-state fidelity translates directly into fewer distillation rounds and reduced hardware footprint.

1.3 Lao–Criger (2022): Magic-State Injection on the Rotated Surface Code

Lao and Criger [2] adapt Li's post-selection protocol to the rotated surface code and introduce new injection schemes that exploit the more qubit-efficient rotated geometry.

Rotated surface-code geometry. A distance-d rotated planar surface code encodes one logical qubit using only $2d^2$ data qubits (versus $2d^2 - 2d + 1$ for the regular layout), while retaining local stabilizer measurements on a 2D grid. The logical operators are strings of Pauli operators along orthogonal boundaries of the code.

CR and MR injection schemes. Two main schemes are introduced:

- *CR (corner-rotated)*: the magic state is initialized on a data qubit located at a corner of the rotated lattice. The rest of the patch is initialized in $|0\rangle$ or $|+\rangle$ such that, after two rounds of stabilizer measurements and post-selection, the logical state is an eigenstate of a logical operator supported along the top and left boundaries.

- *MR (middle-rotated)*: the magic state is initialized on a data qubit in the middle of the rotated lattice. The initialization pattern is chosen so that the logical operator is supported along the central horizontal and vertical lines passing through the magic qubit. This reduces the number of "sensitive" data qubits whose single-qubit faults can induce undetected logical errors.

Both schemes apply Li's two-round post-selection rule: reject runs whenever stabilized plaquettes that should be satisfied based on initialization give the wrong eigenvalue, or when outcomes between the first and second measurement rounds disagree.

Circuit-level noise and analytic p_L. Using a similar circuit-level depolarizing model, Lao and Criger enumerate single-operation errors that lead to logical faults without triggering post-selection. For example, in the CR scheme they find a leading-order logical error rate of the form

$$p_L^{\mathrm{CR}} \approx \frac{3}{5}p_2^2 + 2p_I + \frac{2}{3}p_1 + \dots, \tag{3}$$

while the MR scheme yields

$$p_L^{\mathrm{MR}} \approx \frac{3}{5}p_2^2 + p_I + \frac{2}{3}p_1 + \dots, \tag{4}$$

with the precise coefficients depending on which CNOTs and data qubits can host undetected faults.

They also analyze biased noise models where $p_1 \ll p_2$ and/or p_I differs significantly from p_2. Under realistic biased regimes motivated by superconducting devices, the MR scheme can outperform both the CR scheme and Li's original protocol on the regular surface code.

Numerical validation. Stabilizer-based Monte Carlo simulations under the circuit-level noise model confirm that the numerically estimated p_L converges to the analytic leading-order expressions at small p_2. Lao and Criger also study how p_L and rejection probabilities depend on the code distance, and observe that (in their setting) distance primarily affects acceptance probability rather than the leading-order contribution to p_L.

1.4 Berthusen *et al.* (2025): 2D Local qLDPC Architectures and Stacked Stabilizers

Berthusen *et al.* [3] focus on implementing quantum LDPC codes under strict 2D local connectivity using a bilayer architecture and teleportation-based routing. While their primary subject is not the surface code, their work provides important architectural ideas that we can adapt to model connectivity constraints in our injection protocols.

Key points include:

- A bilayer architecture with data and check qubits in one layer and routing ancillas in another, enabling long-range CNOTs via teleportation with a distance-dependent error cost.

- The "stacked model" and *masking*: measuring geometrically local stabilizer generators frequently, while measuring large-radius generators less often to reduce time overhead.

- Circuit-level depolarizing noise for single- and two-qubit gates, measurements, and idling, with detailed circuit-level simulations on qLDPC codes.

For our purposes, their framework suggests:

- Modeling architectures where some stabilizer measurements in the injection patch are significantly more expensive (in depth and error) than others.
- Exploring post-selection protocols in which only a subset of "cheap" stabilizers are measured in early rounds, with "expensive" stabilizers measured less frequently or only before acceptance.

1.5 Biased Surface Codes and Magic-State Preparation

Recent work on biased-noise surface codes and magic-state preparation (e.g., XZZX surface codes and biased-noise magic-state gadgets) demonstrates that encoding and error-correction schemes tailored to a dominant dephasing channel can achieve substantially improved thresholds and logical error rates. This motivates explicitly incorporating bias into our noise and optimization models.

In particular, we can draw on:

- XZZX surface-code constructions with high thresholds under dephasing-dominated noise.
- Biased-noise magic-state preparation schemes that achieve quadratic improvements in magic-state infidelity by exploiting noise bias.
- Analyses of logical noise bias in magic-state injection, which quantify how the injection gadget transforms physical bias into logical bias.

2 Theoretical Model Design

2.1 Architecture-Specific Circuit-Level Noise Model

We now specify a flexible circuit-level noise model suitable for several hardware regimes (e.g., superconducting qubits, trapped ions, neutral atoms). The model distinguishes error rates by operation type and allows for Pauli bias.

2.1.1 Operation types

We consider the following set of operations on physical qubits:

- Initialization in $|0\rangle$ or $|+\rangle$.
- Single-qubit Clifford and non-Clifford gates (Hadamard, phase, T, etc.).
- Two-qubit CNOT (or CZ) gates, restricted to nearest neighbors in a 2D lattice, or implemented via teleportation in a bilayer architecture.
- Measurement in the computational basis (and potentially in the X basis).
- Idle (no gate) steps, during which decoherence acts.

2.1.2 Error channels

For each operation type, we assign an independent noise channel acting after the ideal operation:

- **Initialization:** a qubit intended to be initialized in $|0\rangle$ is instead in $X|0\rangle$ with probability p_I, modeled as a classical bit-flip. Similarly for initialization in $|+\rangle$, with a Z-flip probability.

- **Measurement:** an ideal projective measurement in the computational basis is followed by classical outcome flip with probability p_M.
- **Single-qubit gates:** after an ideal single-qubit unitary U, a Pauli error acts according to

$$\mathcal{E}_1(\rho) = (1 - p_1)\,\rho + p_1 \sum_{P \in \{X,Y,Z\}} q_P P \rho P, \tag{5}$$

  where q_P are non-negative and sum to 1. Depolarizing noise corresponds to $q_X = q_Y = q_Z = 1/3$. Biased noise can be modeled by choosing, for example, $q_Z \gg q_X, q_Y$.

- **Two-qubit CNOT gates:** after an ideal CNOT, a two-qubit Pauli error acts as

$$\mathcal{E}_2(\rho) = (1 - p_2)\,\rho + p_2 \sum_{P \in \mathcal{P}_2 \setminus \{II\}} r_P P \rho P, \tag{6}$$

  where $\mathcal{P}_2$ is the two-qubit Pauli group and the r_P define a distribution. Two-qubit depolarizing noise corresponds to $r_P = 1/15$ for all non-identity P. Biased noise may favor certain correlated Z errors.

- **Idle steps:** during an idle time step of duration Δt, a dephasing channel acts on each qubit:

$$\mathcal{E}_{\text{idle}}(\rho) = (1 - p_{Z,\text{idle}})\,\rho + p_{Z,\text{idle}} Z \rho Z, \tag{7}$$

  where $p_{Z,\text{idle}} \approx 1 - e^{-\Delta t / T_2}$. Amplitude damping or more general decoherence channels can be used if needed.

2.1.3 Connectivity and effective two-qubit error rates

For a planar architecture with only nearest-neighbor two-qubit gates, stabilizer measurement circuits are composed of local CNOTs according to a prescribed CNOT ordering. In a bilayer architecture with teleportation-based routing, a logical long-range CNOT between distant qubits is implemented by a sequence of Bell-pair preparations, Bell measurements, and local CNOT/CZ gates.

If a long-range CNOT between qubits separated by Manhattan distance L is realized using teleportation, the effective two-qubit error rate can be modeled as

$$p_2^{\text{eff}}(L) = 1 - (1 - p_2)^{n(L)}, \tag{8}$$

where $n(L)$ is the number of two-qubit gates required in the teleportation circuit. In the presence of entanglement purification, $p_2^{\text{eff}}(L)$ becomes a more complicated function of L, p_2, and the purification protocol.

For the purposes of magic-state injection on the rotated surface code, we may initially restrict to a 2D nearest-neighbor layout (no long-range CNOTs), and later incorporate bilayer effects if needed.

2.2 Parametric Family of Injection Protocols

We now define a general family of injection protocols P on the rotated surface code, of which Li's and Lao–Criger's schemes are special cases.

2.2.1 Rotated surface-code patch and logical operators

Consider a distance-d_1 rotated planar surface code patch encoding a single logical qubit. Let $\mathcal{D}$ denote the set of data-qubit indices and $\mathcal{S}^X, \mathcal{S}^Z$ the sets of X- and Z-type stabilizer generators, respectively. Logical operators are strings of X and Z along opposite boundaries of the patch.

We parameterize the injection patch by:

- The choice of magic-qubit location $j_M \in \mathcal{D}$ (corner, edge, or middle).
- The initialization pattern for all other data qubits: for each $j \in \mathcal{D} \setminus \{j_M\}$, we choose $|0\rangle$ or $|+\rangle$ such that the joint state is an eigenstate of a desired logical operator when the magic qubit is initialized in $|m\rangle$.

2.2.2 Stabilizer-measurement circuits and scheduling

For each stabilizer generator $S \in \mathcal{S}^X \cup \mathcal{S}^Z$, we choose a measurement circuit consisting of:

- An ancilla qubit initialized in $|0\rangle$ (for Z-type) or $|+\rangle$ (for X-type) as appropriate.
- A sequence of CNOT gates between the ancilla and the data qubits in the support of S, with a specified order.
- A final single-qubit measurement of the ancilla.

Let $\mathcal{C}(S)$ denote the ordered list of CNOTs for stabilizer S. A *schedule* for a single round of stabilizer measurements is a partition of all CNOTs $\bigcup_S \mathcal{C}(S)$ into time steps such that no qubit participates in more than one CNOT per time step.

The schedule strongly influences the set of single-fault locations that can lead to undetected logical errors, as shown by Li and Lao–Criger. An important part of the project is to search over a restricted space of schedules (e.g., those with minimal depth) to minimize the leading-order coefficients in p_L.

2.2.3 Post-selection rounds and stabilizer subsets

A protocol P is specified by:

$$P = (d_1, j_M, \{\mathcal{C}(S)\}_S, r, \{\mathcal{S}_t\}_{t=1}^r), \tag{9}$$

where:

- d_1 is the initial code distance.
- j_M is the magic-qubit location.
- $\{\mathcal{C}(S)\}_S$ are the measurement circuits (including CNOT orderings) for each stabilizer.
- r is the number of post-selection rounds.
- $\mathcal{S}_t \subseteq \mathcal{S}^X \cup \mathcal{S}^Z$ is the subset of stabilizers measured in round t.

Li and Lao–Criger correspond to $r = 2$ and $\mathcal{S}_1 = \mathcal{S}_2 = \mathcal{S}^X \cup \mathcal{S}^Z$, up to stabilizers that are already satisfied by the initialization pattern. More general protocols may, for example, measure only a subset of "cheap" stabilizers in the first round, and the full set in subsequent rounds.

2.2.4 Acceptance condition

Given the measured syndromes $\{s_t(S)\}$, a run is accepted if and only if:

1. For each S that is satisfied by the initialization pattern, the first-round measurement satisfies $s_1(S) = +1$.
2. For each S measured in both rounds t and t', the outcomes agree: $s_t(S) = s_{t'}(S)$.

If any of these conditions fail, the run is discarded and all data qubits are reinitialized for another attempt.

2.3 Analytic First-Order Error Counting

Given a protocol P and noise model parameters $\theta = (p_I, p_M, p_1, p_2, \dots)$, we wish to obtain a leading-order expression for the logical error rate $p_L(P, \theta)$ and, if possible, the logical noise bias. The analysis follows Li and Lao–Criger but is generalized to the larger protocol family and biased noise.

2.3.1 Heisenberg-picture tracking of stabilizers and logical operators

We describe the code in terms of its stabilizer group $\mathcal{S}$ and logical Pauli operators $\overline{X}, \overline{Z}$. In the Heisenberg picture, we track how these operators transform under the unitary part of the injection and measurement circuits.

For each time step in the circuit (including initialization, magic-state rotation, CNOT layers, and measurement), we update the stabilizers and logical operators according to the conjugation action of the gates. For example, for a CNOT with control qubit c and target qubit t, the Pauli operators transform as

$$X_c \mapsto X_c X_t, \qquad Z_c \mapsto Z_c, \tag{10}$$

$$X_t \mapsto X_t, \qquad Z_t \mapsto Z_c Z_t. \tag{11}$$

By propagating all stabilizers and logical operators through the first round of measurements, we can identify:

- Which stabilizers have deterministic $+1$ outcomes given the ideal circuit and initialization.
- The final form of the logical operator that corresponds to the encoded magic state.

2.3.2 Fault locations and undetected logical errors

A *fault location* is a specific operation (initialization, single-qubit gate, CNOT, measurement) at which a Pauli error may occur. To leading order in the noise parameters, we retain only events where a single fault occurs somewhere in the circuit.

For each fault location ℓ and each non-identity Pauli error E_ℓ that can occur there (with probability depending on $p_I, p_1, p_2, \ldots$), we determine:

1. Whether E_ℓ changes any of the measured stabilizer outcomes in such a way that the run is rejected.
2. Whether E_ℓ changes the eigenvalue of the final logical operator (e.g., $\overline{Z}$ for a T-type magic state) on accepted runs.

Formally, let $\mathcal{A}$ be the set of *accepted* single-fault events:

$$\mathcal{A} = \{(\ell, E_\ell) : \text{the run is accepted and the logical state is flipped}\}. \tag{12}$$

The leading-order logical error probability is then

$$p_L(P,\theta) \approx \sum_{(\ell,E_\ell)\in\mathcal{A}} p(E_\ell), \tag{13}$$

where $p(E_\ell)$ is the probability of that particular error under the noise model.

2.3.3 Symbolic expressions in terms of noise parameters

Carrying out this enumeration yields symbolic expressions of the form

$$p_L(P,\theta) \approx \alpha_2(P)p_2 + \alpha_1(P)p_1 + \alpha_I(P)p_I + \alpha_M(P)p_M + \dots, \tag{14}$$

where the coefficients $\alpha_\cdot(P)$ are non-negative rational numbers that count the number of dangerous fault locations weighted by the probabilities of their associated Pauli components. In regimes where the first-order contributions cancel or are absent, the leading term may instead be quadratic, e.g.,

$$p_L(P,\theta) \approx c(P)p_2^2 + O(p_2^3). \tag{15}$$

By computing these coefficients for different protocols P and hardware parameter regimes θ, we can compare and optimize injection schemes analytically.

2.3.4 Logical noise bias

In settings with biased noise, it is often useful to characterize the *logical* noise after injection in terms of its Pauli components. Let the effective logical channel after injection (conditioned on acceptance) be

$$\mathcal{E}_L(\rho) = (1-p_L)\rho + \sum_{P\in\{\overline{X},\overline{Y},\overline{Z}\}} p_P P\rho P. \tag{16}$$

We can define a logical bias parameter, for example

$$\eta_L = \frac{p_{\overline{Z}}}{p_{\overline{X}}}, \tag{17}$$

which quantifies whether the logical noise is predominantly phase-like or bit-flip-like. The same single-fault enumeration used for p_L can be refined to keep track of which logical Pauli each undetected fault induces, thereby giving leading-order estimates for $p_{\overline{X}}, p_{\overline{Y}}, p_{\overline{Z}}$.

2.4 From Injection to Magic-State Distillation Cost

To demonstrate the practical impact of improved injection, we will couple the obtained $p_L(P,\theta)$ to standard magic-state distillation protocols. For example, in the Bravyi–Kitaev 15-to-1 protocol, the output error rate scales as

$$p_{\text{out}} \approx 35p_{\text{in}}^3, \tag{18}$$

for small input error p_{in}. Given a target magic-state error p_{target}, we can determine the number of distillation rounds required starting from p_L, and estimate the total number of raw magic states and qubit-time volume needed.

This analysis enables quantitative comparison between different injection protocols and architectures, in terms of the ultimate cost per high-fidelity T gate.

3 Numerical Simulations

3.1 Goals of the Simulation Study

The numerical simulations serve three main purposes:

1. **Validation of analytic leading-order expressions:** check that Monte Carlo estimates of $p_L(P, \theta)$ converge to the analytic predictions as physical error rates tend to zero.

2. **Characterization beyond leading order:** quantify the impact of weight-2 and higher faults, and explore regions where analytic approximations break down.

3. **Exploration of parameter space:** study how p_L, logical noise bias, and rejection probabilities depend on code distance d_1, number of post-selection rounds r, noise bias, and connectivity constraints.

3.2 Simulation Framework and Backends

3.2.1 Stabilizer-based simulation

Because the injection circuits consist solely of Clifford operations (initializations, Clifford gates, CNOTs, and Pauli measurements), and the injected magic state itself is a single-qubit non-Clifford state, the overall circuit is amenable to efficient stabilizer-based simulation with Pauli frame tracking and sampling.

Two main approaches are possible:

- **Direct stabilizer simulation with non-Clifford inputs:** represent the initial magic state as a state vector or as a probabilistic mixture of stabilizer states, and propagate through the Clifford circuit using a stabilizer simulator (e.g., a Stim-like library). This may require handling a small non-stabilizer component explicitly.

- **Pauli-twirled approximation:** approximate the effect of the non-Clifford injection gate and noise as a Pauli channel acting on an otherwise stabilizer state. This is valid for estimating logical error rates when coherent phases are not the focus.

In practice, many existing fault-tolerance simulation libraries support circuit-level noise on Clifford circuits and can be extended to handle a single magic-state qubit.

3.2.2 CPU and GPU execution

The simulations can be parallelized over many independent Monte Carlo samples. Each sample consists of:

1. Sampling a noise realization according to the circuit-level noise model.

2. Evolving the state (or stabilizer tableau) through the injection circuit.

3. Computing the measured stabilizers and deciding acceptance or rejection.

4. For accepted runs, determining whether a logical error has occurred (by comparing the final logical operator eigenvalues to the ideal case).

Since these samples are embarrassingly parallel, they can run efficiently on:

- Multi-core CPUs using thread-level parallelism (e.g., OpenMP, multiprocessing).

- GPUs using CUDA, HIP, or other accelerator frameworks, particularly if the simulator supports batched updates of many stabilizer tableaus in parallel.

For moderate code distances (e.g., $d_1 = 3, 5, 7$) and target statistical precision (e.g., order 10^{-4} in p_L), millions of samples per parameter point may be required. GPU acceleration can significantly reduce wall-clock time in such regimes.

3.3 Simulation Algorithm Outline

We outline a generic Monte Carlo procedure to estimate $p_L(P, \theta)$ and the rejection probability $p_{\mathrm{rej}}(P, \theta)$.

3.3.1 Circuit construction

For a chosen protocol P and noise parameters θ:

1. Construct the full injection circuit as a list of time steps, where each time step contains a set of commuting operations (single-qubit gates, CNOTs, measurements).

2. For each operation, record its type, target/control qubits, and position in the lattice (for potential connectivity-aware noise models).

3.3.2 Single-sample simulation

For each Monte Carlo sample:

1. Initialize the physical qubits in their nominal states: the magic qubit in $|m\rangle$, others in $|0\rangle$ or $|+\rangle$. Sample and apply initialization errors according to p_I.

2. For each time step in the circuit:

   2.(a) Apply the ideal Clifford operations of that time step to the stabilizer tableau or state representation.

   2.(b) For each operation, sample whether a noise event occurs; if so, apply the corresponding Pauli error (or more general channel) to the tableau.

3. Record the measurement outcomes for all stabilizer and data-qubit measurements.

4. Apply the post-selection rule: if the outcomes violate any acceptance condition, label the run as rejected and discard it.

5. For accepted runs, infer whether a logical error has occurred. This can be done by:

   - Comparing the final logical Pauli eigenvalues to those of the ideal encoded magic state.
   - Alternatively, decoding the final syndrome to estimate the most likely logical Pauli and checking whether it differs from identity.

3.3.3 Estimators

Let N be the total number of Monte Carlo samples, N_{acc} the number of accepted runs, and N_{err} the number of accepted runs with a logical error. Then estimators for the acceptance probability and conditional logical error rate are

$$\hat{p}_{\mathrm{acc}} = \frac{N_{\mathrm{acc}}}{N}, \tag{19}$$

$$\hat{p}_L = \frac{N_{\mathrm{err}}}{N_{\mathrm{acc}}}. \tag{20}$$

Standard binomial confidence intervals (e.g., Clopper–Pearson) can be used to estimate the statistical uncertainty in these quantities as functions of N.

3.4 Code Structure for Simulations

3.4.1 Data structures

A practical implementation (e.g., in Python with a C++ backend, or in C++ directly) can use the following data structures:

- `QubitLattice`: stores the positions and indices of physical qubits in the rotated patch.
- `Stabilizer`: stores the support and type (X or Z) of each stabilizer generator.
- `CircuitOp`: represents a single operation (initialization, gate, measurement) with metadata.
- `Circuit`: an ordered list of `CircuitOp`s grouped into time steps.
- `Tableau`: a stabilizer tableau representation of the quantum state, with methods to apply Clifford gates and Pauli errors efficiently.

3.4.2 Pseudo-code sketch

A high-level pseudo-code sketch for the Monte Carlo loop is:

```
for theta in noise_parameter_grid:
    P = choose_protocol(theta)
    circuit = build_injection_circuit(P)
    N_acc = 0
    N_err = 0
    for sample in range(N_samples):
        tableau = initialize_tableau(P)
        apply_initialization_noise(tableau, theta)
        accepted = True
        for timestep in circuit.timesteps:
            apply_ideal_gates(tableau, timestep)
            apply_noise(tableau, timestep, theta)
            record_measurements(tableau, timestep)
        if not acceptance_condition_satisfied():
            accepted = False
        if accepted:
            N_acc += 1
            if logical_error_occurred(tableau, P):
                N_err += 1
    p_acc_hat = N_acc / N_samples
    p_L_hat   = N_err / max(N_acc, 1)
    store_results(theta, P, p_acc_hat, p_L_hat)
```

3.4.3 GPU considerations

On a GPU, one can batch many independent samples by representing multiple stabilizer tableaus in parallel. For example:

- Represent each tableau as a set of bit-packed rows (for X and Z components) and store a batch of them in GPU memory.
- Implement Clifford gate and Pauli-error updates as bitwise operations over 32- or 64-bit words, applied in parallel to the batch.

- Use random-number generation on the GPU (e.g., cuRAND) to sample noise events per operation and per sample.

The main complexity lies in handling measurements and post-selection efficiently across the batch, since different samples will be accepted or rejected at different times.

3.5 Appendix: Example Analytic Coefficients

As a concrete example, consider a distance-3 rotated patch with MR injection, under an unbiased depolarizing model with perfect single-qubit gates and measurement ($p_1 = p_I = p_M = 0$). Following Lao–Criger, one finds that the leading-order logical error rate behaves as

$$p_L^{\mathrm{MR}} \approx c_{\mathrm{MR}} p_2^2, \tag{21}$$

with a coefficient c_{MR} that can be obtained by counting the number of two-qubit Pauli errors on specific CNOTs that induce logical faults without triggering rejection.

In our generalized setting, we would re-derive such coefficients for each candidate protocol P and then compare their performance under realistic biased and connectivity-constrained noise models.

Acknowledgments

This outline is designed as a starting point for developing a full-length manuscript. Detailed derivations of the analytic coefficients, explicit circuit diagrams, and extensive numerical results would be added in subsequent iterations.

References

[1] Y. Li, "A magic state's fidelity can be superior to the operations that created it," *New J. Phys.* **17**, 023037 (2015).

[2] L. Lao and B. Criger, "Magic state injection on the rotated surface code," in *Proceedings of the 19th ACM International Conference on Computing Frontiers* (CF '22), 2022.

[3] N. Berthusen, D. Devulapalli, E. Schoute, A. M. Childs, M. J. Gullans, A. V. Gorshkov, and D. Gottesman, "Toward a 2D local implementation of quantum low-density parity-check codes," *PRX Quantum* **6**, 010306 (2025).