Fast Unitary Preparation of Surface-Code Logical States on Neutral-Atom Hardware

Ludwig Schmid[1,*] and Robert Wille[1,2]
[1]Chair for Design Automation, Technical University of Munich, Munich, Germany
[2]MQSC, Garching near Munich, Germany
{ludwig.s.schmid, robert.wille}@tum.de [*] Corresponding author

***Abstract*—Preparing logical states is a constantly recurring primitive at the start of any surface-code-based quantum computation. The standard measurement-based protocol is expensive on neutral atoms, since measurements are orders of magnitude slower than gates and, on single-zone architectures, additionally require shuttling the atoms to a readout zone. In this work, we present a measurement-free, unitary preparation of surface-code Pauli eigenstates tailored to neutral atoms. A *bidirectional* stabilizer-expanding CNOT cascade grows the patch from its middle line outward in depth $(d+3)/2$, while retaining fault distance d in the protected error direction. Each layer maps to one collective atom move followed by a single global Rydberg pulse, with about one row pickup per two layers. We compile and simulate the resulting schedules with the open-source `bloqade` toolchain under a hardware-calibrated, circuit-level neutral-atom noise model. The bidirectional cascade achieves the lowest logical error rate at every distance and is the only construction tested that operates below threshold for both logical states at the considered device-level noise.**

***Index Terms*—surface code, logical state preparation, fault tolerance, neutral atoms, quantum error correction**

I. Introduction

Quantum error correction with surface codes has reached the sub-threshold regime in experiment [1], [2], moving the cost of the underlying logical primitives into focus. Among these, preparing fresh logical states is ubiquitous. The standard protocol initializes all data qubits and runs $\sim d$ rounds of stabilizer measurement. This is particularly expensive on neutral atoms, where measurement is orders of magnitude slower than gates [2], [3] and, on single-zone architectures, additionally requires shuttling the atoms to a readout zone [2].

Unitary, measurement-free preparation avoids this cost entirely. Instead of measuring the stabilizers into their eigenstates, the idea is to *grow* them. A CNOT cascade expands the stabilizers of an initial product state step by step to the full code state. A recent proposal showed that this can be done fault-tolerantly against one error type at depth $\mathcal{O}(d)$ [4], [5]. Related unitary and measurement-free encoders reach similar depths [6]–[8]. These constructions, however, are not optimized for neutral-atom machines. The latter offer a specific, highly parallel operation set of global Rydberg CZ layers [9] and collective, order-preserving row and column moves of atoms held in acousto-optic-deflector (AOD) tweezers [10], and a preparation scheme matched to these capabilities has been missing.

In this work, we adapt and improve the unitary stabilizer-expanding preparation for single-zone neutral-atom hardware. Our construction combines a middle-out growth pattern, employed previously in a different context for the preparation of topologically ordered states on a superconducting processor [11], with the fault-tolerant expansion rule of [4]. Growing the patch from its middle line in both directions at once halves the depth and yields exactly the collective, nearest-neighbour move structure that AOD shuttling favours. In summary, we contribute:

- a bidirectional middle-out cascade of depth $(d+3)/2 = \mathcal{O}(d/2)$ with unchanged protected-direction fault distance d (compared to d and $3(d-1)$ [4]);
- an AOD-native, single-zone realization with one collective nearest-neighbour move and one global Rydberg pulse per layer, and about one row pickup per two layers;
- a validation using the open-source `bloqade` toolchain [12] with a hardware-calibrated neutral-atom noise model, in which the bidirectional cascade beats both baselines and is the only construction below threshold for both logical states at the considered device-level noise.

II. Background

A. Rotated surface code

The surface code [13]–[15] in its rotated layout [16] places data qubits on a $d_x \times d_z$ grid with X- and Z-type plaquette stabilizers arranged in a checkerboard (Fig. 1). The logical operators are strings of X (X_L) and Z (Z_L) Paulis along the two grid directions. The code distance against Z errors is the number of columns d_z, and against X errors the number of rows d_x. Throughout, we call the error type against which a preparation retains its full fault distance the *protected* direction and the other type the *conjugate* direction.

B. Unitary stabilizer-expanding preparation

Unitary preparation [4], [5] initializes qubits in $|0\rangle$ or $|+\rangle$ and grows the code stabilizers by CNOT layers. For the surface code, the circuit seeds one line of the patch, and each layer entangles the next line of fresh qubits, extending every stabilizer on the growth front until all stabilizers reach full weight. Figure 2(a) shows this growth directly as stabilizer (detector) slices. Since a CNOT copies X operators from control to target and Z operators from target to control, the

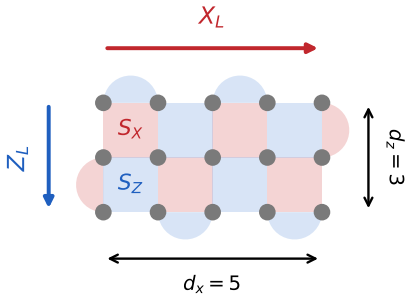


Fig. 1. Rotated surface-code patch (d_x=5 rows, d_z=3 columns): data qubits (grey), X-type (red) and Z-type (blue) stabilizers, and the logical strings X_L/Z_L along the two directions.

two stabilizer types expand in opposite roles across the same CNOT layer (top vs. bottom row). The crucial ingredient is a careful arrangement of the CNOTs such that errors propagate only in a correctable way while the stabilizers still expand [4], as discussed in Sec. III-A.

C. Neutral-atom hardware and AOD shuttling

Neutral-atom processors hold qubits in static optical-tweezer (SLM) traps and move selected atoms with AOD tweezers. The native operations are global and local single-qubit rotations and CZ gates, applied by a single Rydberg pulse to all pairs within blockade radius [2], [9], [17]. AOD moves act on whole rows or columns collectively, must preserve the trap order, and cannot cross [10], [18], [19]. Architectures use either a single zone or separate storage, entangling, and readout zones [10]. Since measurement is orders of magnitude slower than gates [3], measurement-free preparation is attractive, especially on single-zone machines with no readout zone at all.

III. Fast Bidirectional Preparation

Recently, a unitary surface-code preparation was proposed [4], which we call *single-pivot*. It grows the patch from a single corner qubit, where attaching each new line costs three sequential CNOT layers, so the full patch takes $3(d-1)$ layers. A first improvement is to exploit parallelism. Once a qubit has been entangled, it can itself drive the CNOTs of the next layer, so the qubits entangled in one layer expand the patch together in the next. The resulting *unidirectional* cascade attaches one fresh line per CNOT layer and sweeps across the patch in d layers. Both constructions use the same CNOT arrangement as [4] and are therefore fault-tolerant in the protected direction.

But nothing forces the sweep to start at a boundary. Our *bidirectional* cascade seeds the *middle* line of the patch and expands in both directions simultaneously. After the two innermost attachments, which share the seed line and therefore run sequentially, each layer attaches one fresh line on each side. This halves the depth to $(d+3)/2 = \mathcal{O}(d/2)$ without extra qubits and with the same protected-direction distance, since each half of the patch is a unidirectional cascade. Figure 2 shows a 5×3 patch completing in $(5+3)/2 = 4$ CNOT layers, and Table I compares the scaling of the methods.

TABLE I
Schedule scaling of the preparation methods.

Construction	CZ layers	AOD shuttles	Row pickups
Single-pivot [4]	$3(d-1)$	$4(d-1)$	$d-1$
Unidirectional	d	$\approx \frac{3}{2}d$	$\approx \frac{1}{2}d$
Bidirectional (this work)	$(d+3)/2$	$\approx \frac{3}{4}(d+3)$	$\approx \frac{1}{4}(d+3)$

The construction extends directly to rectangular patches, whose depth follows the swept dimension only. Since the preparation protects one error type anyway, asymmetric patches giving more distance to the dominant type are a natural fit under biased noise [20].

A. Fault tolerance and the conjugate direction

The fault tolerance of these stabilizer-expanding cascades has been established in [4], and since each half of the bidirectional cascade is such a cascade, the arguments carry over to our construction. Why the protected direction keeps its full distance is visible directly in the error propagation through the cascade (Fig. 3, top, for a $|0\rangle_L$ patch). A single X fault copies along the CNOTs of subsequent layers. Because fresh qubits are only targets, its final support, reduced modulo the code stabilizers, is either trivial (a stabilizer as in Fig. 3(top, a)), a weight-1 error, or a low-weight error supported *orthogonally* to the logical string as in Fig. 3(top, b). All of these are correctable, so the fault distance in the protected direction remains d [4].

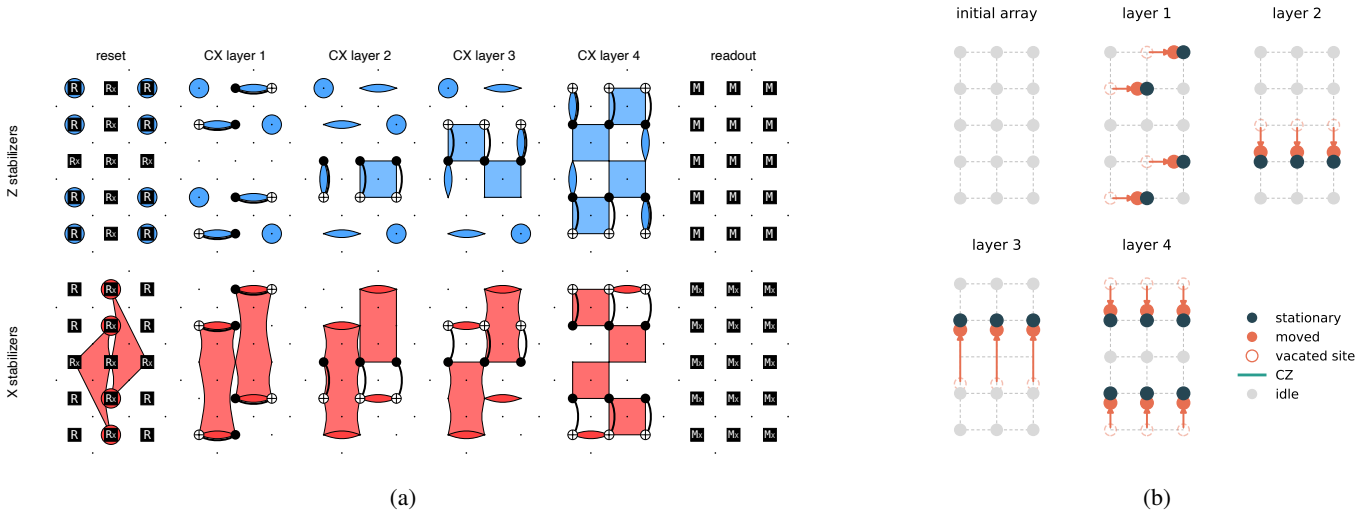
The conjugate type behaves differently (Fig. 3, bottom). Z faults copy in the opposite direction and spread *along* the logical Z_L. A fault on the seed line grows into the full logical as in Fig. 3(bottom, a), which acts trivially on $|0\rangle_L$ and is therefore harmless. A fault on a later line, however, can leave an irreducible weight-2 error along the logical as in Fig. 3(bottom, b), the same error propagation structure identified in [4]. The conjugate direction is therefore not fault-tolerant. For a stored $|0\rangle_L$ this is benign, since conjugate errors commute with the state and its readout, but it precludes using the fresh patch directly as a Steane-type ancilla [21] for a transversal CX [2] without an intermediate QEC round after the state preparation.

IV. Neutral-Atom Realization

A. Mapping the cascade to atom moves

We now discuss how the bidirectional cascade maps efficiently onto neutral-atom hardware. Each CNOT layer compiles to single-qubit basis changes plus one CZ layer, executed by a single global Rydberg pulse over all paired atoms [9]. Bringing the pairs together requires exactly one collective nearest-neighbour line shift per layer [10]. The moving line is displaced by one lattice step onto its partner line, entangled, and later returned (Fig. 2(b)). All moves are parallel shifts of whole lines, so they preserve the trap order and never cross, fulfilling the AOD constraints.

The shuttling cost amortizes further. The first layer costs one row pickup and two shuttles (onto the partner and back).



(a) (b)

Fig. 2. Bidirectional preparation of a 5×3 $|+\rangle_L$ patch. (a) Construction as stim detector slices, where the Z-type (top, blue) and X-type (bottom, red) stabilizers grow with each CNOT layer. (b) The same cascade as AOD moves, one panel per layer. Each layer is a single collective nearest-neighbour shift followed by a global-Rydberg CZ.

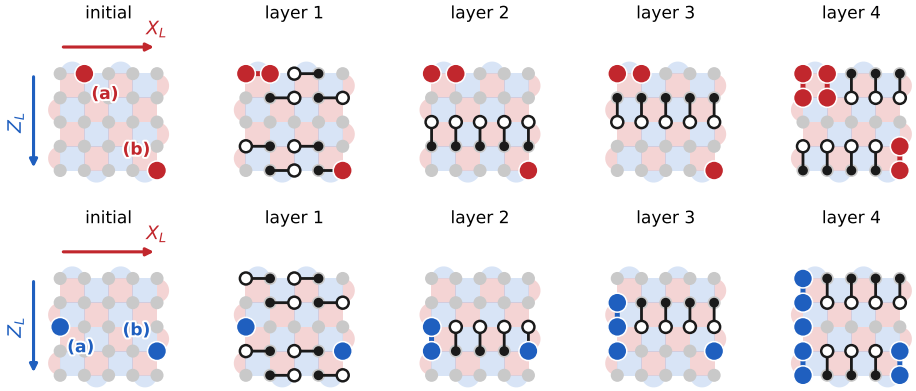


Fig. 3. Single-fault propagation through the cascade of a $|0\rangle_L$ patch, one panel per CNOT layer (propagating CNOTs highlighted). **Top (X, protected):** a fault ends as (a) a stabilizer or (b) a correctable weight-2 error orthogonal to the logical. **Bottom (Z, conjugate):** a fault spreads along Z_L to (a) the full logical operator, acting trivially (harmless), or (b) an irreducible weight-2 error (not fault tolerant).

Afterwards, a picked row serves two consecutive CZ layers by moving onto its first partner, directly on to its second, and only then back, which amounts to three shuttles and one pickup per two layers, as visible in Fig. 2(b). For the bidirectional cascade this yields $\approx \frac{3}{4}(d+3)$ collective shuttles and $\approx \frac{1}{4}(d+3)$ pickup–drop cycles, so global pulses, shuttles, and pickups all scale as $\mathcal{O}(d/2)$. The unidirectional sweep follows the same accounting at twice the layer count, and the single-pivot construction needs four shuttles and one pickup per attached line (Table I).

B. Hardware noise model

We evaluate the schedule under the hardware-calibrated, circuit-level noise model of the open-source `bloqade` SDK [12], which compiles the circuit to the native gate set and attaches Pauli channels to every gate, move, and idle period. Table II lists the key rates. The dominant channels (CZ and shuttling) are strongly Z-biased, and the CZ additionally carries a correlated two-qubit channel dominated by its $IZ/ZI/ZZ$ terms.

TABLE II
KEY PAULI RATES OF THE NEUTRAL-ATOM NOISE MODEL [12].

Channel (per qubit)	$p_X = p_Y$	p_Z
CZ, paired atoms	6.5×10^{-4}	3.2×10^{-3}
CZ, unpaired atoms	5.1×10^{-4}	2.2×10^{-3}
Move (AOD shuttle)	8.1×10^{-4}	2.5×10^{-3}
Idle	3.1×10^{-4}	4.6×10^{-4}
1q gate, local	4.1×10^{-4}	4.1×10^{-4}

V. EVALUATION

To quantify the benefit of the shorter schedules, we compare all three constructions for both $|0\rangle_L$ and $|+\rangle_L$ at distances 3, 7, and 11, sweeping a global scaling factor on all noise rates of Table II. Each experiment prepares the logical state, measures all data qubits in the matching basis, and decodes the reconstructed stabilizers with minimum-weight perfect matching (Stim [22] + PyMatching [23]), with up to 10^7 shots per point. Code and evaluation data are available at https://github.com/lsschmid/fast-surface-code.

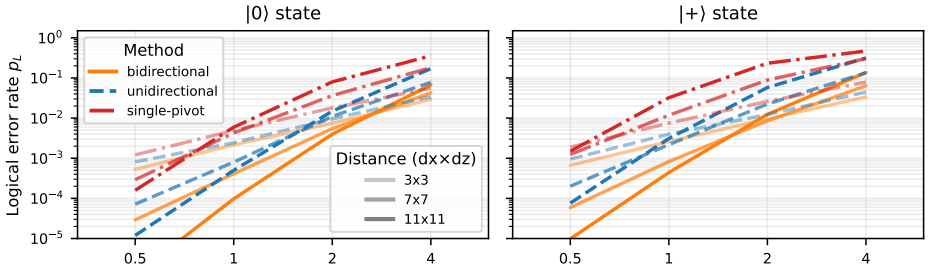


Fig. 4. Logical error rate vs. noise scaling factor (1 = device-level neutral-atom noise). Colour and line style denote the construction, line opacity the distance.

Figure 4 shows the logical error rates for both states. At every distance and noise scale, the constructions order as bidirectional $<$ unidirectional $<$ single-pivot. This ordering reflects the circuit volume, since fewer layers mean fewer global pulses, shuttles, and idle periods, and therefore less exposure to the dominant noise channels of Table II.

We further observe that $|0\rangle_L$ reaches substantially lower error rates than $|+\rangle_L$, which follows from the noise bias. The dominant channels are Z-biased, and Z errors are harmless for $|0\rangle_L$, since Z_L stabilizes the state and the Z-basis readout is blind to them, whereas for $|+\rangle_L$ they are exactly the dangerous error type.

Finally, at the considered device-level noise (scaling factor 1), only the bidirectional construction is below threshold for both states, meaning its logical error rate falls with increasing distance. The unidirectional cascade achieves this only for $|0\rangle_L$, the single-pivot for neither. This shows that, at current hardware error rates, optimizing the state preparation for neutral atoms is essential for sub-threshold behaviour.

VI. Discussion and Conclusion

In this work, we improved the unitary, measurement-free preparation of surface-code logical states of [4] and adapted it to neutral-atom hardware and its AOD constraints. Growing the patch bidirectionally from its middle line halves the preparation depth and maps onto $\mathcal{O}(d/2)$ global Rydberg pulses, collective nearest-neighbour shuttles, and row pickups, without any measurement or transport to a readout zone. With the open-source neutral-atom toolchain `bloqade`, we compiled the schedules to the native gate set and evaluated them under hardware-calibrated device noise. The proposed construction outperforms the original single-pivot proposal of [4] and the unidirectional sweep at every distance, and is the only construction tested that operates below threshold for both logical states at the considered device-level noise, making it a practical candidate for logical-state preparation in experiments. The preparation is fault-tolerant against one error type, while the conjugate type spreads along the logical but stabilizes the prepared state. A single subsequent QEC round restores fault tolerance against both types [4].

Acknowledgment

L.S. would like to thank Tom Peham and Katharina Köstler for helpful discussions. The authors acknowledge funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program grant agreement No. 101001318 and No. 101114305, and the Munich Quantum Valley (MQV), which is supported by the Bavarian state government with funds from the Hightech Agenda Bayern Plus. Furthermore, this work was supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under grant numbers 563402549. The software used to generate the figures and evaluation data in this work, and initial drafts of the manuscript text, were prepared with the assistance of a generative AI system (Anthropic Claude [24]); all code and results were verified, and the manuscript reworked and finalized, by the authors, who take responsibility for all content.

References

[1] Google Quantum AI and Collaborators, "Quantum error correction below the surface code threshold," *Nature*, vol. 638, pp. 920–926, 2025.
[2] D. Bluvstein *et al.*, "Logical quantum processor based on reconfigurable atom arrays," *Nature*, vol. 626, pp. 58–65, 2024.
[3] T. M. Graham *et al.*, "Midcircuit measurements on a single-species neutral alkali atom quantum processor," *Phys. Rev. X*, vol. 13, no. 4, p. 041051, 2023.
[4] L. Colmenarez, R. Zen, J. Olle, F. Marquardt, and M. Müller, "Unitary fault-tolerant encoding of Pauli states in surface codes," 2026.
[5] R. Zen, J. Olle, L. Colmenarez, M. Puviani, M. Müller, and F. Marquardt, "Quantum circuit discovery for fault-tolerant logical state preparation with reinforcement learning," *Phys. Rev. X*, vol. 15, no. 4, p. 041012, 2025.
[6] O. Higgott *et al.*, "Optimal local unitary encoding circuits for the surface code," *Quantum*, vol. 5, p. 517, 2021.
[7] P.-K. Tsai and S. Puri, "A unitary encoder for surface codes," 2025.
[8] H. Goto, Y. Ho, and T. Kanao, "Measurement-free fault-tolerant logical-zero-state encoding of the distance-three nine-qubit surface code in a one-dimensional qubit array," *Phys. Rev. Research*, vol. 5, no. 4, p. 043137, 2023.
[9] S. J. Evered *et al.*, "High-fidelity parallel entangling gates on a neutral-atom quantum computer," *Nature*, vol. 622, pp. 268–272, 2023.
[10] D. Bluvstein *et al.*, "A quantum processor based on coherent transport of entangled atom arrays," *Nature*, vol. 604, pp. 451–456, 2022.
[11] K. J. Satzinger *et al.*, "Realizing topologically ordered states on a quantum processor," *Science*, vol. 374, no. 6572, pp. 1237–1241, 2021.
[12] QuEra Computing, "Bloqade: the neutral-atom SDK," https://github.com/QuEraComputing/bloqade-circuit, 2025.
[13] S. B. Bravyi and A. Y. Kitaev, "Quantum codes on a lattice with boundary," 1998.
[14] E. Dennis, A. Kitaev, A. Landahl, and J. Preskill, "Topological quantum memory," *J. Math. Phys.*, vol. 43, no. 9, pp. 4452–4505, 2002.
[15] 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.
[16] H. Bombin and M. A. Martin-Delgado, "Optimal resources for topological two-dimensional stabilizer codes: Comparative study," *Phys. Rev. A*, vol. 76, no. 1, p. 012305, 2007.
[17] H. Levine, A. Keesling, G. Semeghini, A. Omran, T. T. Wang, S. Ebadi, H. Bernien, M. Greiner, V. Vuletić, H. Pichler, and M. D. Lukin, "Parallel implementation of high-fidelity multiqubit gates with neutral atoms," *Phys. Rev. Lett.*, vol. 123, no. 17, p. 170503, 2019.
[18] L. Schmid, D. F. Locher, M. Rispler, S. Blatt, J. Zeiher, M. Müller, and R. Wille, "Computational capabilities and compiler development for neutral atom quantum processors: connecting tool developers and hardware experts," *Quantum Sci. Technol.*, vol. 9, no. 3, p. 033001, 2024.
[19] D. B. Tan, D. Bluvstein, M. D. Lukin, and J. Cong, "Compiling quantum circuits for dynamically field-programmable neutral atoms array processors," *Quantum*, vol. 8, p. 1281, 2024.
[20] J. P. Bonilla Ataides *et al.*, "The XZZX surface code," *Nat. Commun.*, vol. 12, p. 2172, 2021.
[21] A. M. Steane, "Active stabilization, quantum computation, and quantum state synthesis," *Phys. Rev. Lett.*, vol. 78, no. 11, pp. 2252–2255, 1997.
[22] C. Gidney, "Stim: a fast stabilizer circuit simulator," *Quantum*, vol. 5, p. 497, 2021.
[23] O. Higgott, "PyMatching: A Python package for decoding quantum codes with minimum-weight perfect matching," *ACM Trans. Quantum Comput.*, vol. 3, no. 3, pp. 1–16, 2022.
[24] Anthropic, "Claude (opus 4.8)," https://www.anthropic.com, 2026.