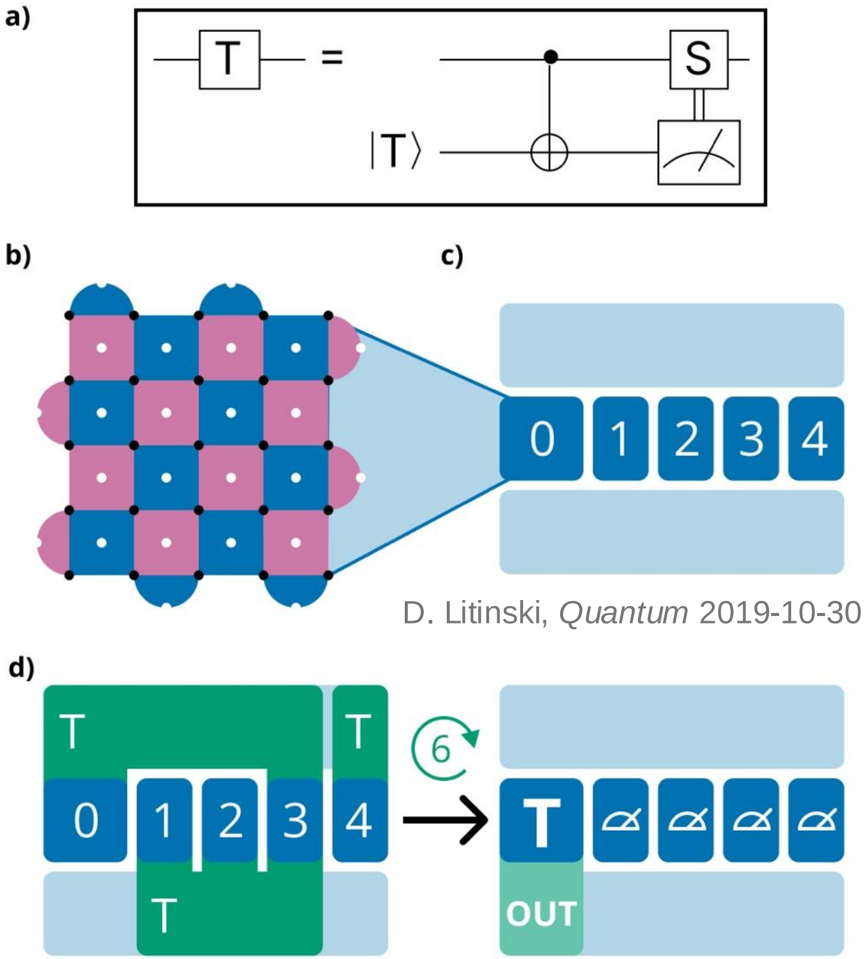
Averting multi-qubit burst errors in surface code magic state factories

Jason D. Chadwick, Christopher Kang, Joshua Viszlai,
Sophia Fuhui Lin, and Frederic T. Chong

arXiv:2405.00146

Background: surface code & magic state distillation

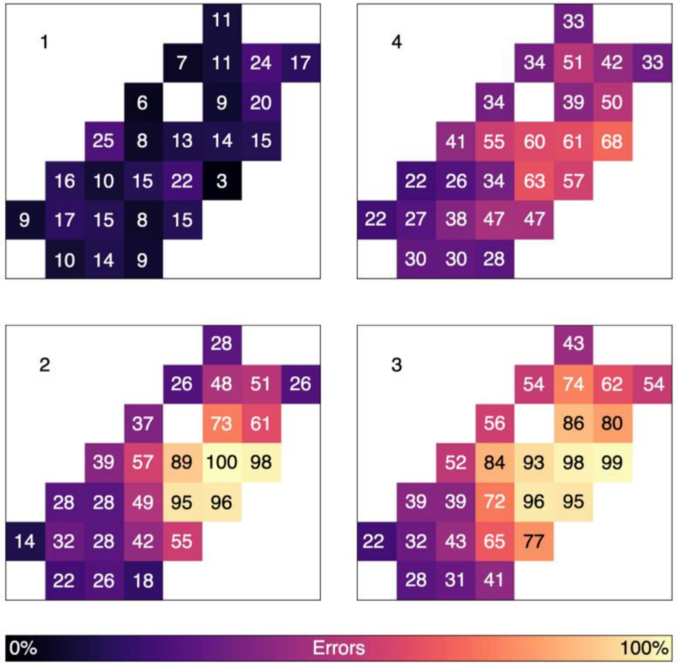
- Surface code: rectangular patch of physical qubits; multi-qubit operations by merging and splitting patches
- Can fault-tolerantly perform Clifford operations, but need T gate to complete universal gate set
- T gate can be performed using a *magic state* $|T\rangle$
- Magic states of high fidelity are generated in *magic state factories* made of surface code tiles
- Magic state factories repeatedly perform *magic state distillation*
    - In this work, we focus on 15-to-1 distillation, which suppresses errors from order p to order p^3
- Magic state distillation estimated to be 60-95% of total program cost (qubitcycles)



D. Litinski, *Quantum* 2019-10-30

Background: multi-qubit burst errors

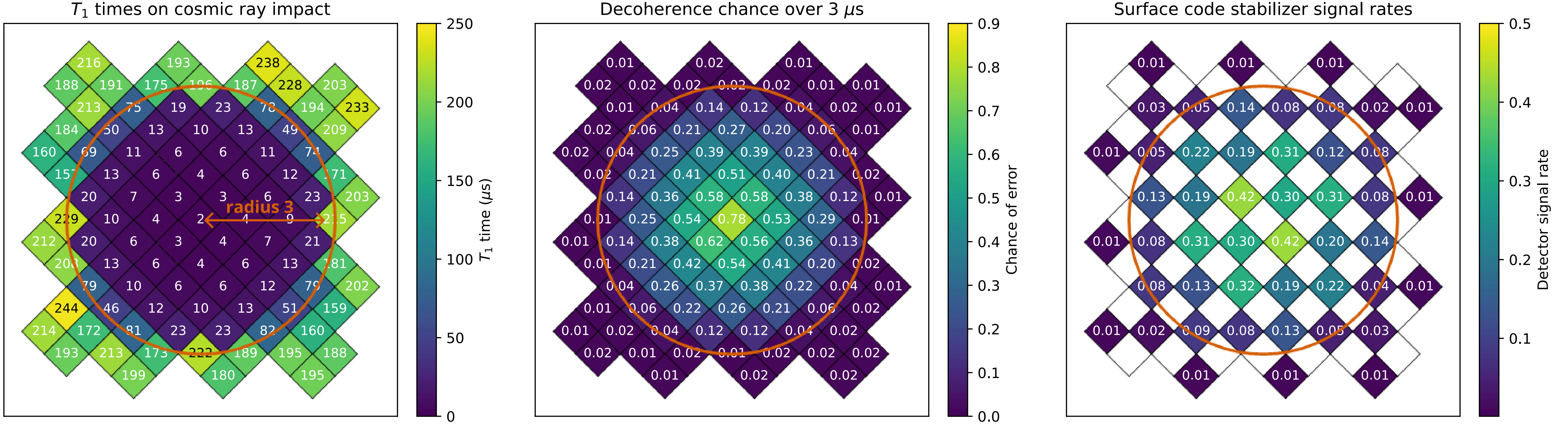
- Quantum error correction relies on sufficiently-small and unchanging physical error rates
- Physical error rates fluctuate significantly on current hardware in a variety of ways
- We focus on *multi-qubit burst errors*: many qubits experiencing an increase in error rate at the same time
- Common source of burst errors in superconducting hardware: cosmic ray impacts
  - Rate of rays can be reduced by shielding, but a single burst error could ruin an hours-long computation
  - Gap engineering can reduce direct sensitivity to radiation, but may come with fabrication tradeoffs and does not solve the whole problem



M. McEwen et al., “Resolving catastrophic error bursts from cosmic rays in large arrays of superconducting qubits,” *Nature Physics* (2022)

Noise model: **Direct**

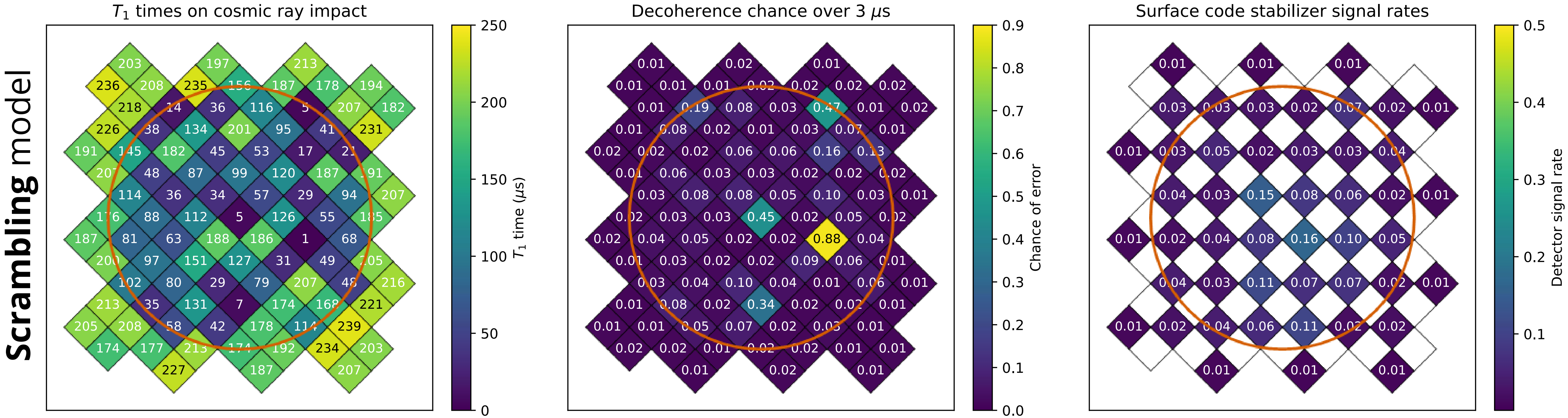
Direct model



Based on M. McEwen et al., “Resolving catastrophic error bursts from cosmic rays in large arrays of superconducting qubits,” Nature Physics (2022).

- Model: ray reduces T_1 times in some radius r_{CRE}. Qubit error rates increase linearly towards center, with maximum reduction at the center of $f_{T_1} T_1^{init}$.
- Ray impacts are Poisson-distributed with rate Γ. Goes away after some time.

Noise model: **TLS Scrambling**



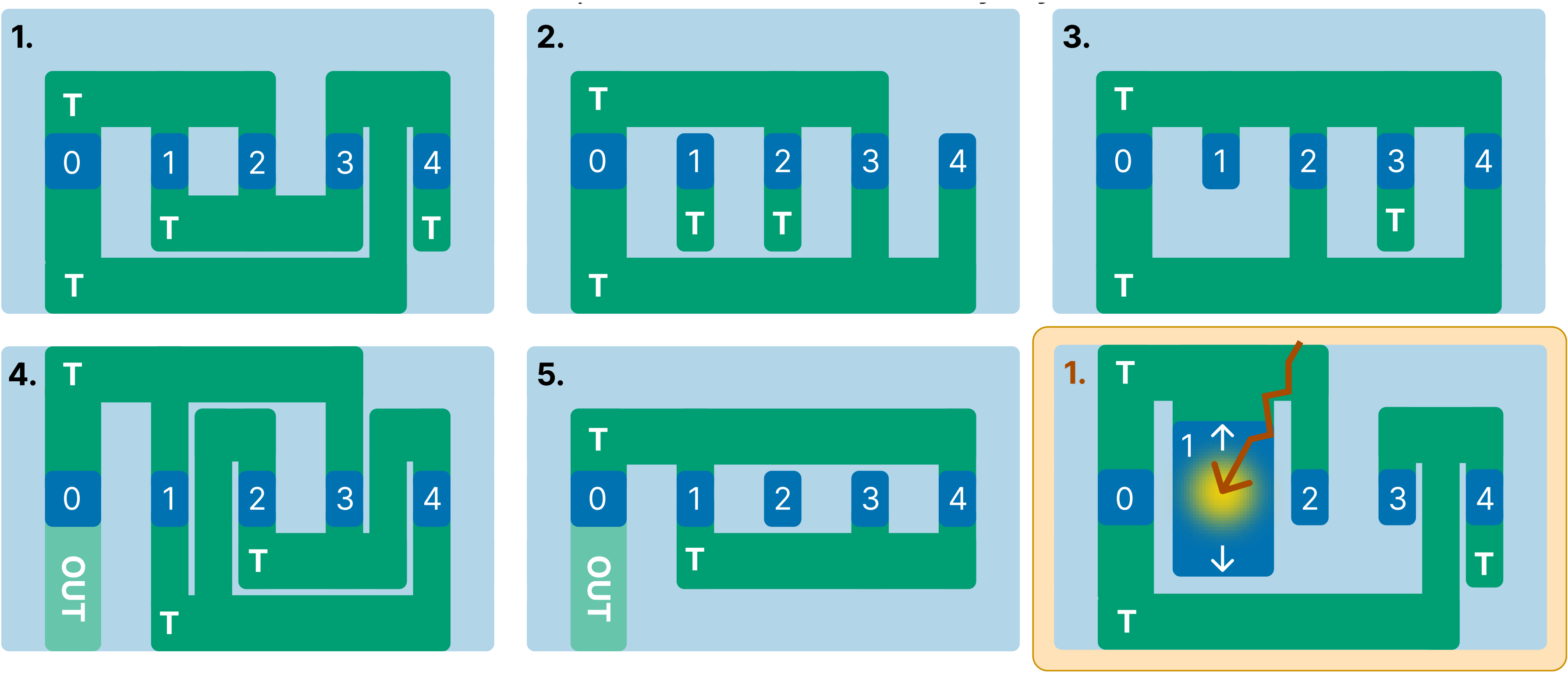
Based on C. D. Wilen et al., “Correlated charge noise and relaxation errors in superconducting qubits,” Nature (2021) and T. Thorbeck et al., “Two-Level-System Dynamics in a Superconducting Qubit Due to Background Ionizing Radiation,” PRX Quantum (2023).

- Model: ray scrambles T_1 times randomly within radius r_{CRE}.
- Ray impacts are Poisson-distributed with rate Γ. Requires active re-calibration to fix.

Baseline: *Code expansion*

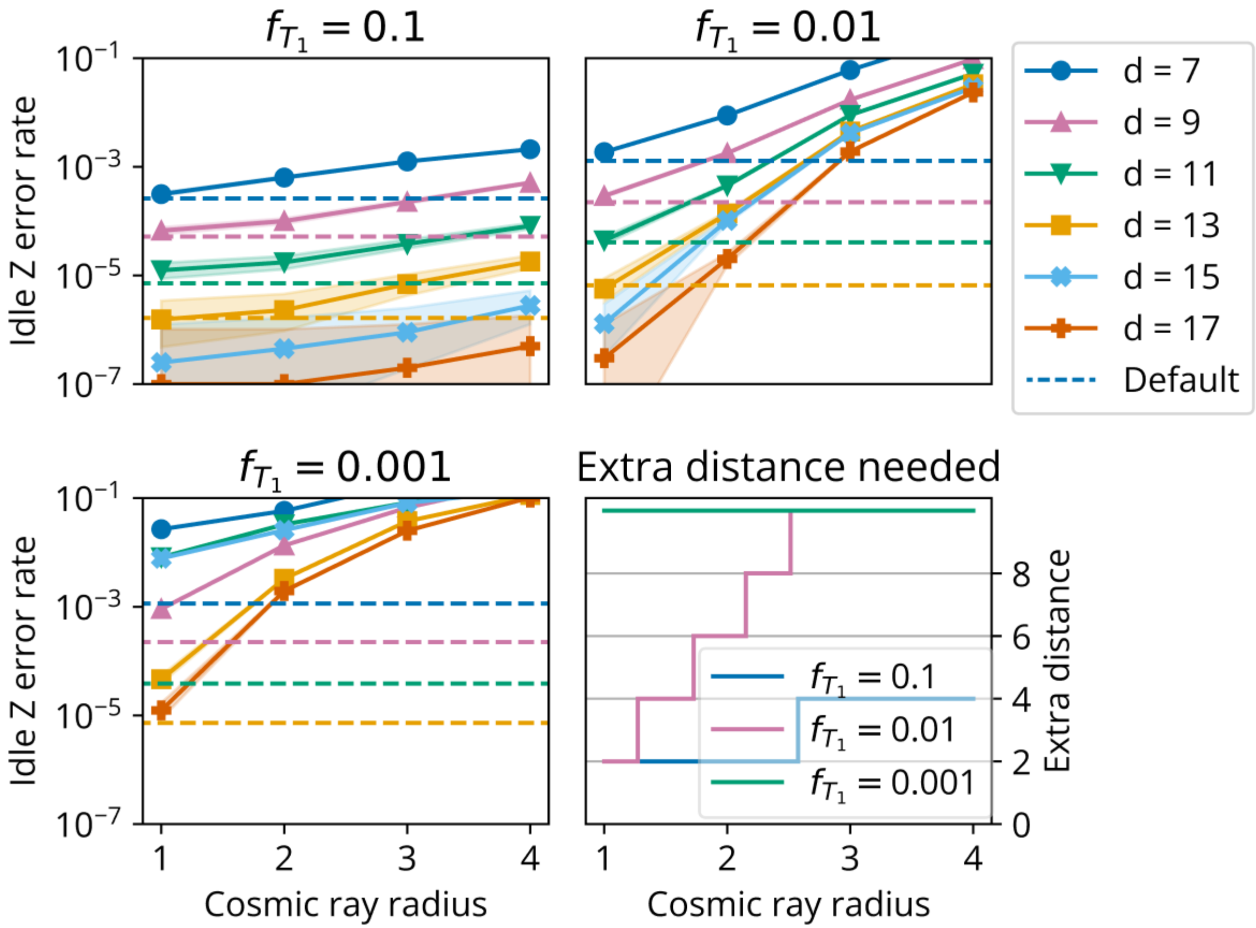
Y. Suzuki et al., “Q3DE: A fault-tolerant quantum computer architecture for multi-bit burst errors by cosmic rays,” MICRO 2022

- Allocate extra buffer space around each patch
  - With enough buffer space, can perform distillation in $5d_m$ steps instead of $6d_m$
- Upon burst event, expand patch to increase error resilience



Baseline: *Code expansion*

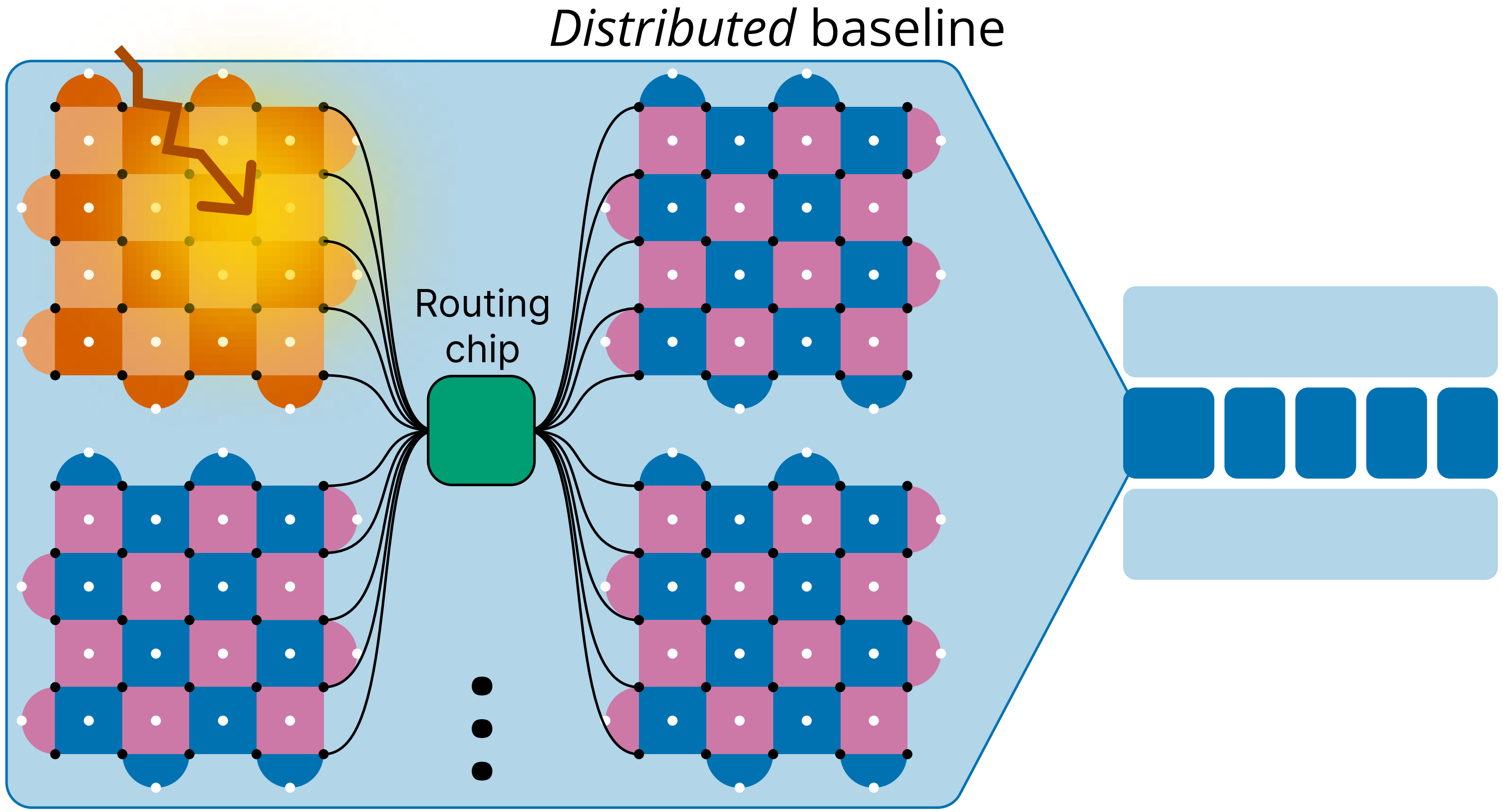
- How much buffer space do we need? Depends on cosmic ray parameters
- For **Direct** model, d_{extra} depends on r_{CRE} and f_{T_1}
- For **Scrambling** model, we assume that added distance must be sufficient for worst-case set of broken qubits, so $d_{\text{extra}} = 2r_{CRE}$
- Assume d_{extra} must be doubled if there is a significant chance of two simultaneous events
    - Depends on $\Gamma \times T_{\text{offline}}$



Baseline: *Distributed*

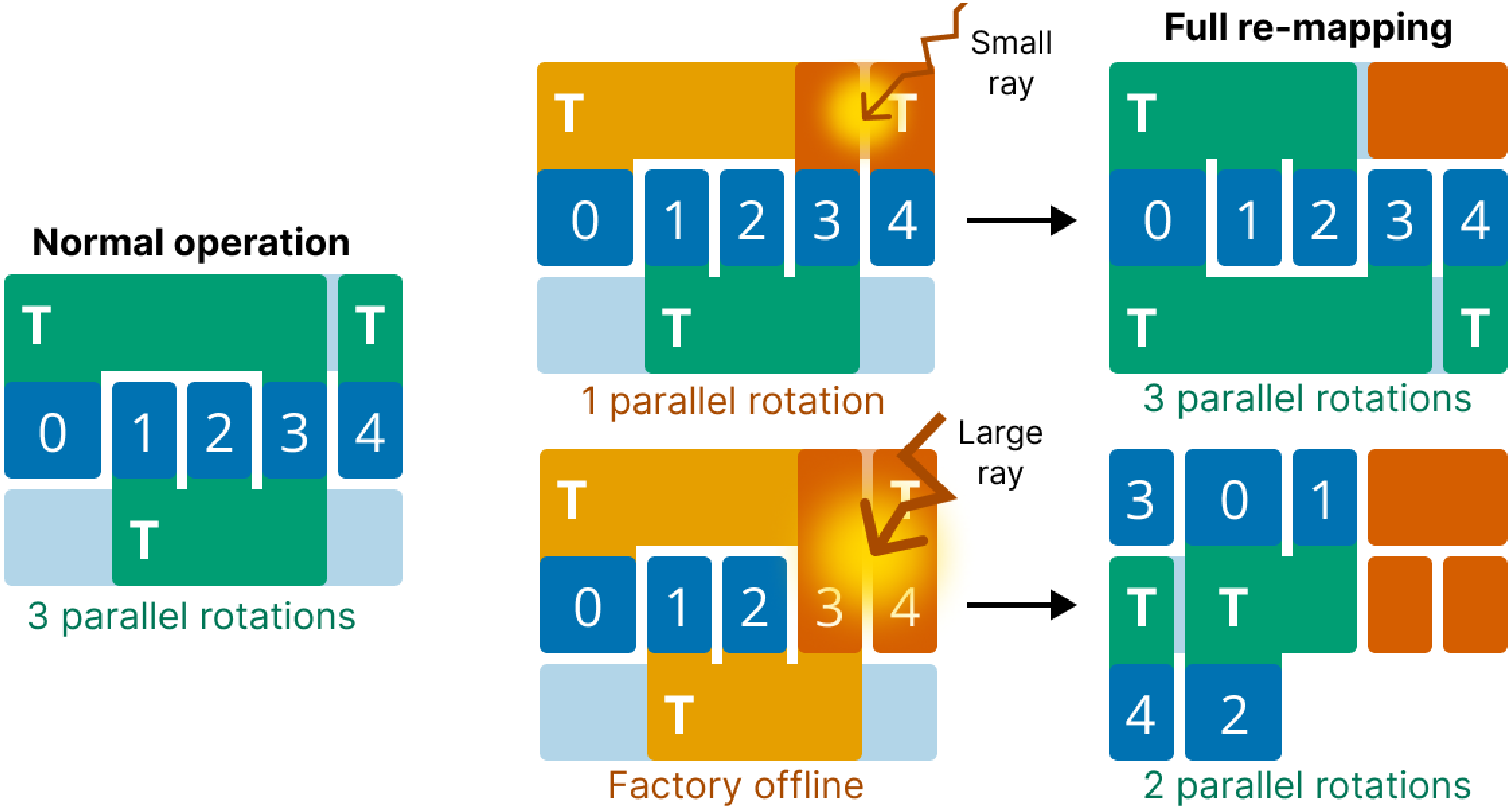
Q. Xu et al., “Distributed Quantum Error Correction for Chip-Level Catastrophic Errors,” *Phys. Rev. Letters* (2022)

- Encode each logical qubit in higher-level distributed code
- A detected burst error is treated as a heralded erasure error (assume entire patch is broken)
- A code with distance d qubits can tolerate $d - 1$ simultaneous erasures
    - $\Gamma \times T_{\text{offline}}$ (probability of simultaneous events) determines required higher-level code

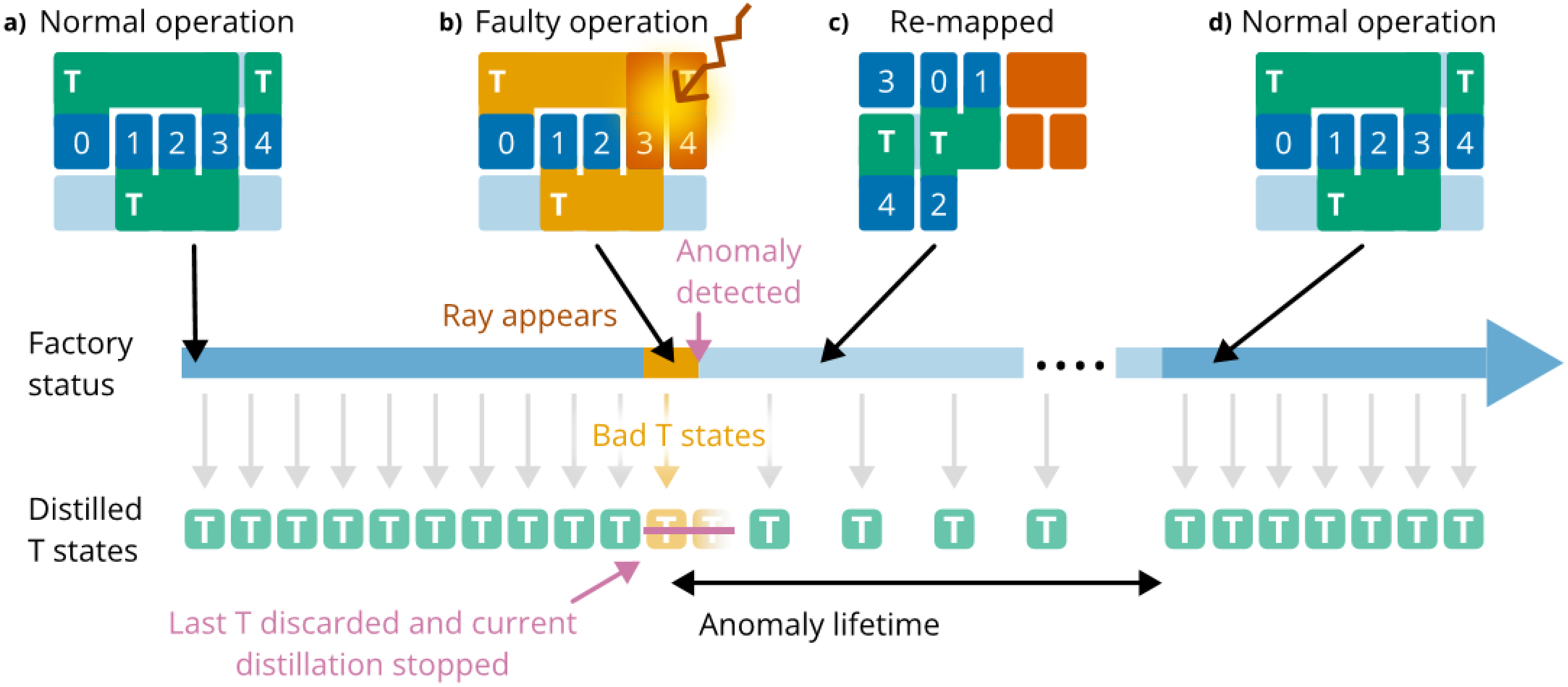


Solution: partially-offline magic state factories

- Magic state factories do not store long-term logical information; we do not have to protect them as carefully
- Idea: if a ray hits, just turn parts of the factory offline until recovery
- Re-mapping allows factory to operate even under more severe disruption

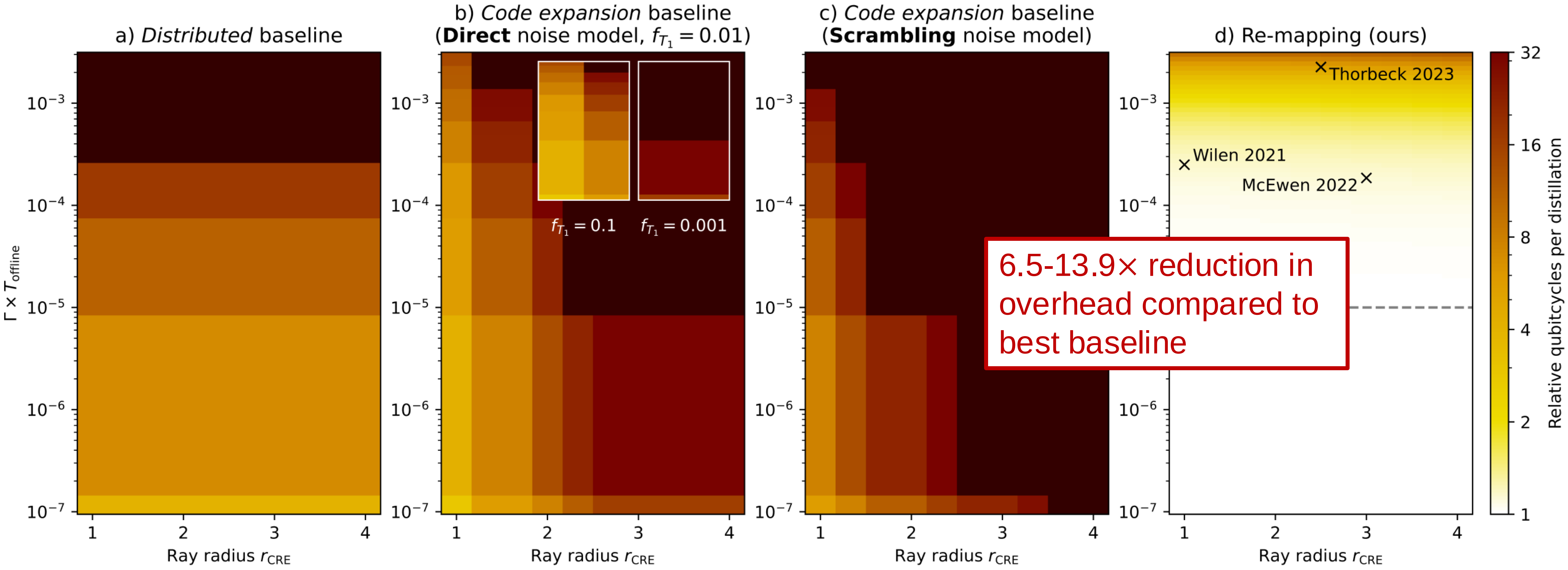


Mitigating burst errors in magic state factories



Comparison to baselines

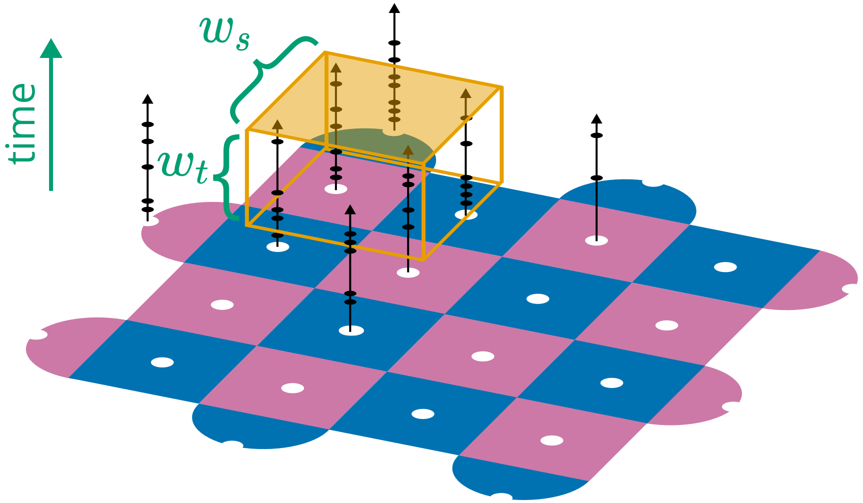
- Both baselines assume **instant** and **complete** detection of burst events, so we compare under that assumption

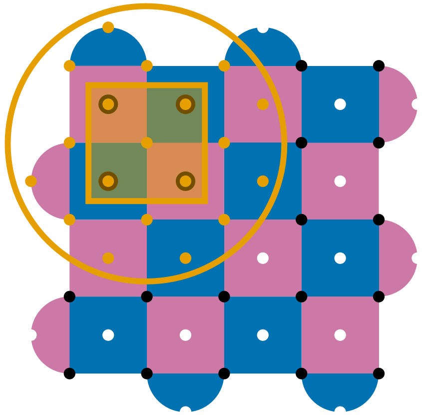


6.5-13.9× reduction in overhead compared to best baseline

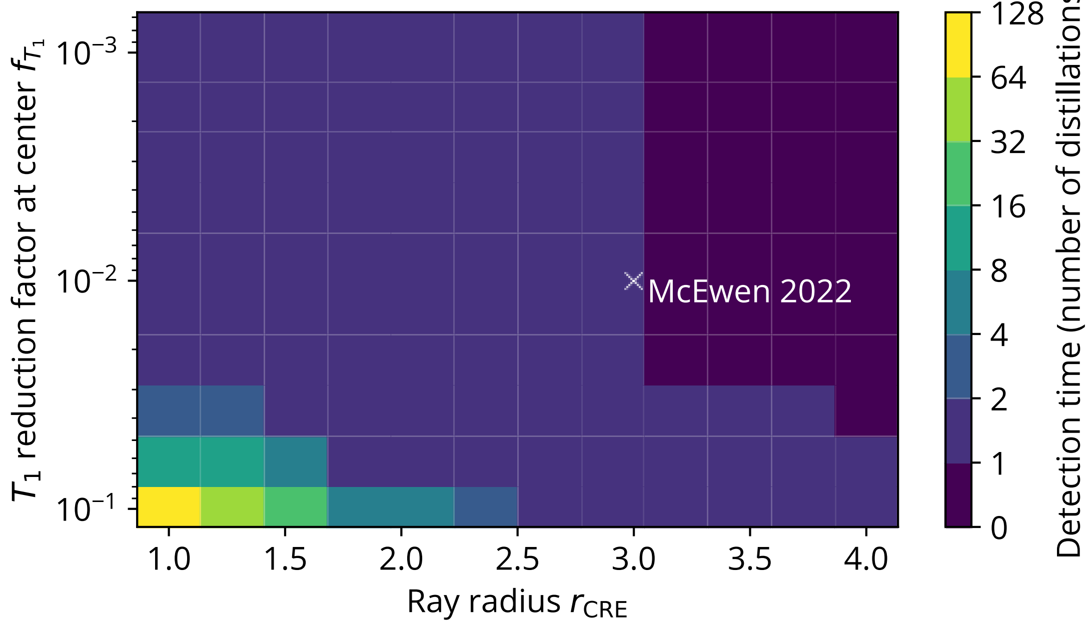
Realistic detection of burst error events

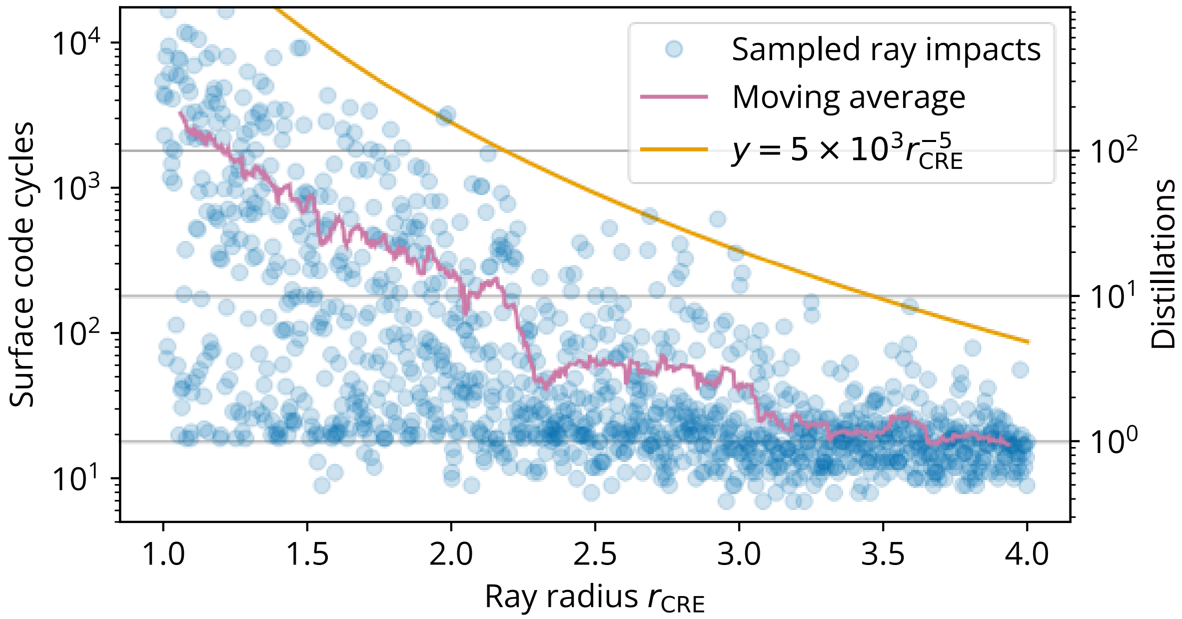
- How quickly can we reliably detect burst errors when our only information is QEC error syndromes?
- Count error syndromes in spatiotemporal windows
- Define spatial windows of size $w_s \times w_s$
- For each spatial window, determine average baseline syndrome rate per stabilizer $p_{\mathrm{syn},i}$
- Define temporal window size w_t and set a threshold number of counts $n_{\mathrm{th},i}$ based on desired false positive rate (FPR)
- Each cycle, count syndromes in each window. If the count exceeds $n_{\mathrm{th},i}$, a detection event is triggered
- Upon a detection event, turn off all qubits with radius r_{off} of the window for duration T_{offline}





Burst error detection latency


Number of distillations to turn off every affected qubit, **Direct** model

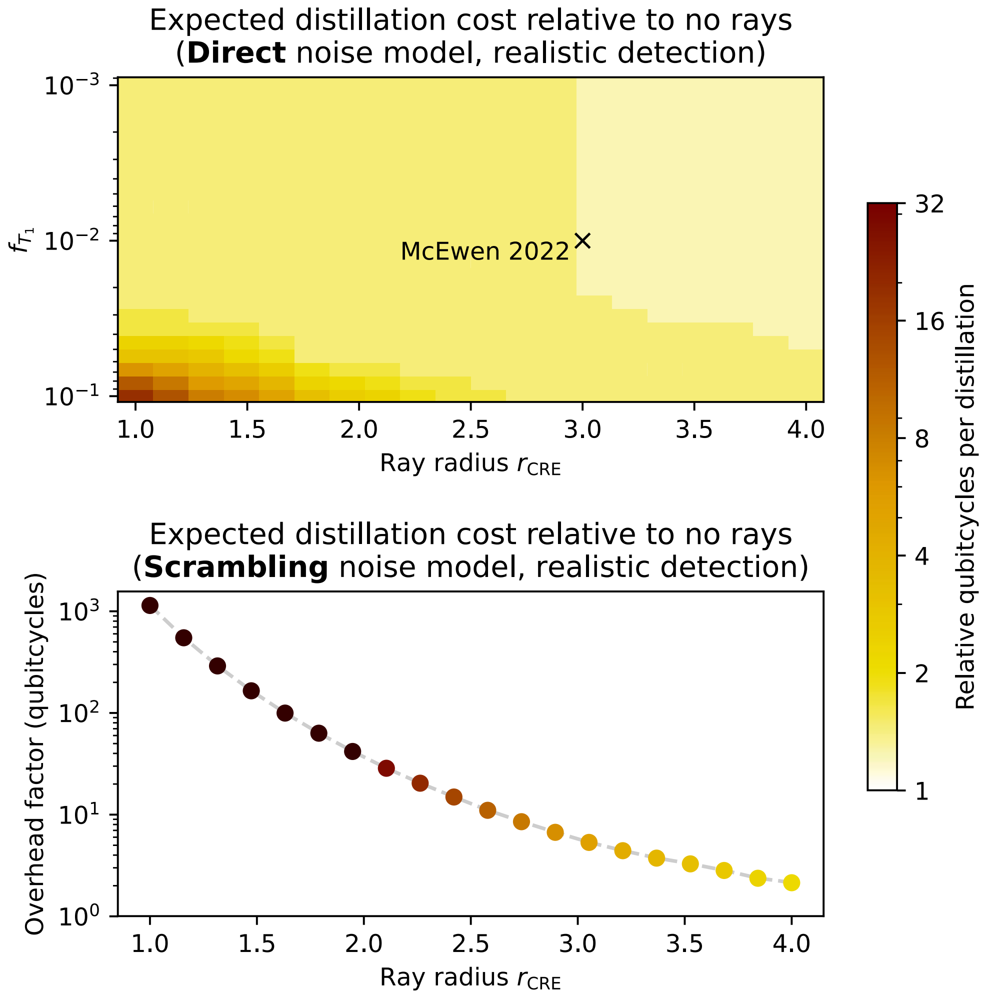

Number of cycles to turn off every affected qubit, **Scrambling** model

High latency for weak burst errors

Scrambling model is hard to reliably detect!

Overhead of re-mapping under realistic detection

- Fixed temporal overhead: $\Gamma \times T_{\mathrm{offline}} = 10^{-5}$
- Spatial overhead determined by T buffer size, which is set by reliable-detection latency
- **Direct** model: latency determined by f_{T_1} and r_{CRE}
  - Less than 2x overhead for most of the studied parameter space, but quickly grows for small and weak rays
- **Scrambling** model: latency determined by r_{CRE}
  - Reliable detection is difficult; need to design for worst-case ray
  - Overhead quickly grows as r_{CRE} decreases



Expected distillation cost relative to no rays
(**Direct** noise model, realistic detection)

Expected distillation cost relative to no rays
(**Scrambling** noise model, realistic detection)

Summary and discussion

- By tailoring burst error mitigation to magic state factories, we reduced mitigation overheads by 6.5-13.9× compared to previous methods
- Scales favorably with Γ (no time overhead until an event happens)
- Easily extends to different magic state factory layouts
  - Overhead factor will *decrease* with increasing factory size, while baseline overheads will *increase*
- Re-mapping factories may be useful for other error sources (fluctuating TLSs, calibration drift, etc.)
- Our method does not apply to logical program qubits – still need a larger-overhead mitigation method for some parts of the processor
- Detection of weaker burst errors is more difficult than previously assumed – we need to carefully study implications for compute qubits