Optimizing Quantum Error Correction Codes with Reinforcement Learning

- **Authors:** H. S. Anwar, E. T. Campbell, D. E. Browne
- **Source:** arXiv:1812.08451

Neeraj Kumar
NS25Z054@smail.iitm.ac.in
NS25Z054

07-Apr-2025

Motivation

- Quantum computers are inherently noisy.
- Quantum Error Correction (QEC) is essential for fault tolerance.
- Surface codes are leading QEC candidates due to locality and high thresholds.
- However, standard surface codes are not optimized for realistic, hardware-specific noise.

Why Surface Codes?

- **"…due to locality…"**

In surface codes:

- **Qubits only interact with their nearest neighbors** (e.g., in a 2D grid).
- This is called **local interaction**.

Why it matters:

Most quantum hardware (like superconducting qubits) can only do **local operations** — they can't connect far-away qubits easily.

Cont...

- **“…and high thresholds.”**

This refers to the **error threshold**:

- It’s the maximum error rate below which error correction **still works reliably**.

- Surface codes can tolerate physical qubit error rates up to **~1% or more**, which is **very high** compared to other codes.

Why it matters:

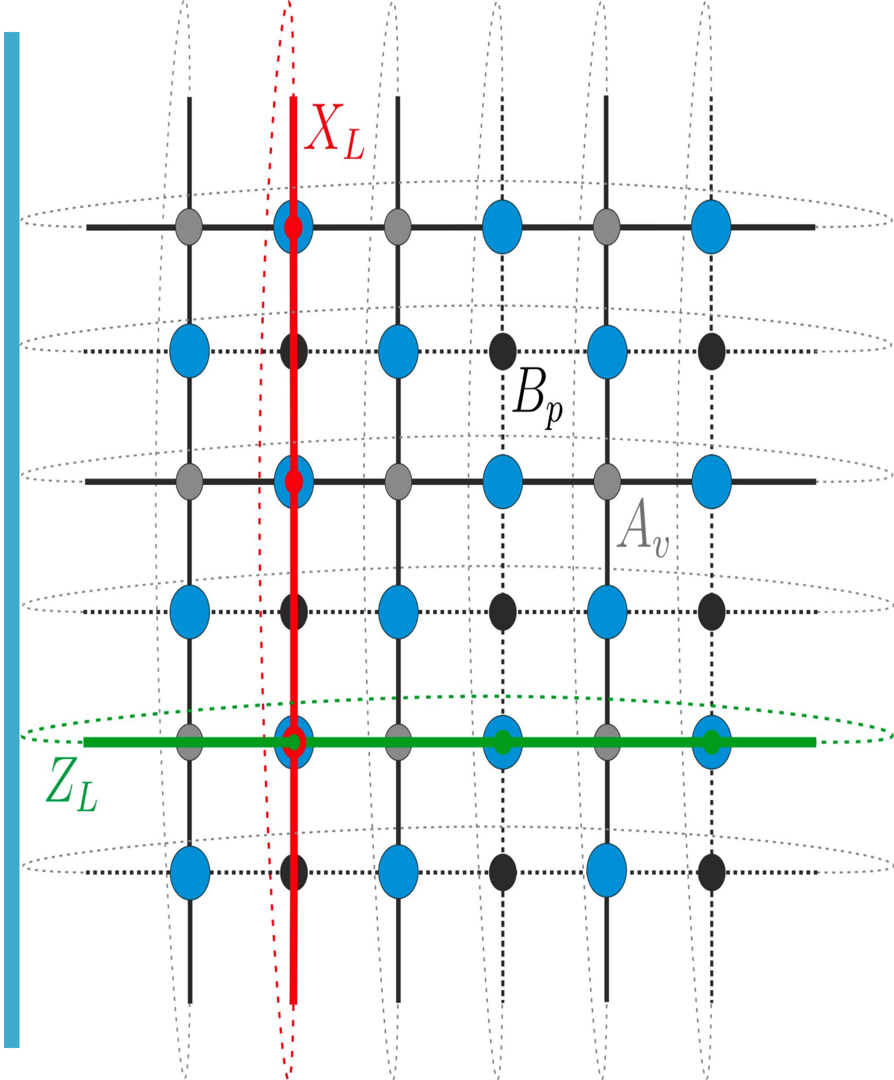
The higher the threshold, the **less perfect** your qubits need to be for QEC to work — that makes surface codes more **robust and scalable**.

Research Goal

- **Objective:** Optimize the layout of surface codes for specific noise characteristics using Reinforcement Learning (RL).
- **Idea:** Different code layouts may perform better under different error models.

Surface Codes Refresher

- Surface codes use data and syndrome qubits.
- Two stabilizer types:
  - Z-type (vertex): detect bit-flip (X) errors.
  - X-type (plaquette): detect phase-flip (Z) errors.
- Logical operators are strings that wrap around the code.
- **Figure:**
  - Shows 3x3 toric surface code on a torus.
  - Blue = data qubits, black = Z-syndrome, gray = X-syndrome

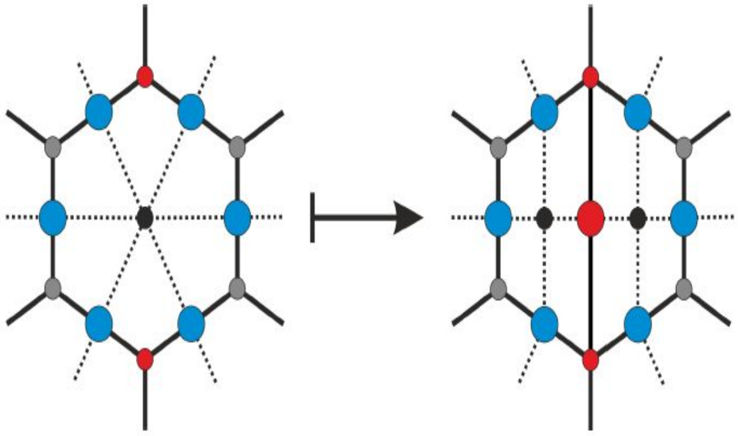


Noise Models

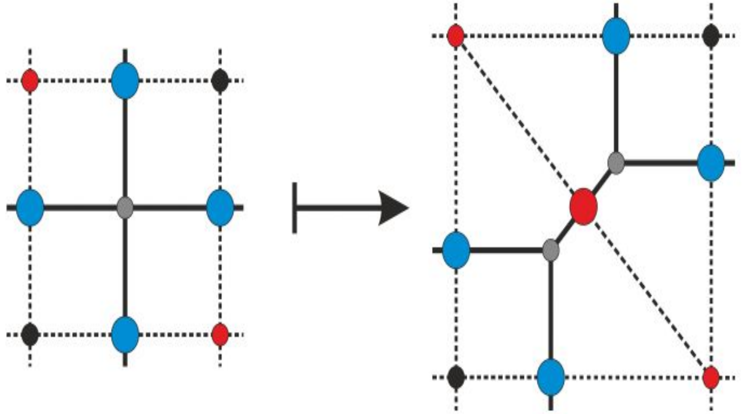
- Considered Pauli noise models:
  - Bit-flip (X), phase-flip (Z), depolarizing noise.
  - Biased noise models: e.g., Z errors >> X errors.
- In practice, noise is hardware-dependent and often biased.

Adaptable Surface Code

- Authors define a set of **alternative surface code layouts**.
- These layouts have the same number of data qubits but different stabilizer placements.
- RL agent selects which layout to use.
- **Goal:** find layout optimal for given noise type.
- **Figure:**
    - An illustration of the basic moves that fault-tolerantly map surface codes to surface codes while changing the underlying lattice.
    - Blue = data qubits, black = Z-syndrome, gray = X-syndrome, red = visual center.



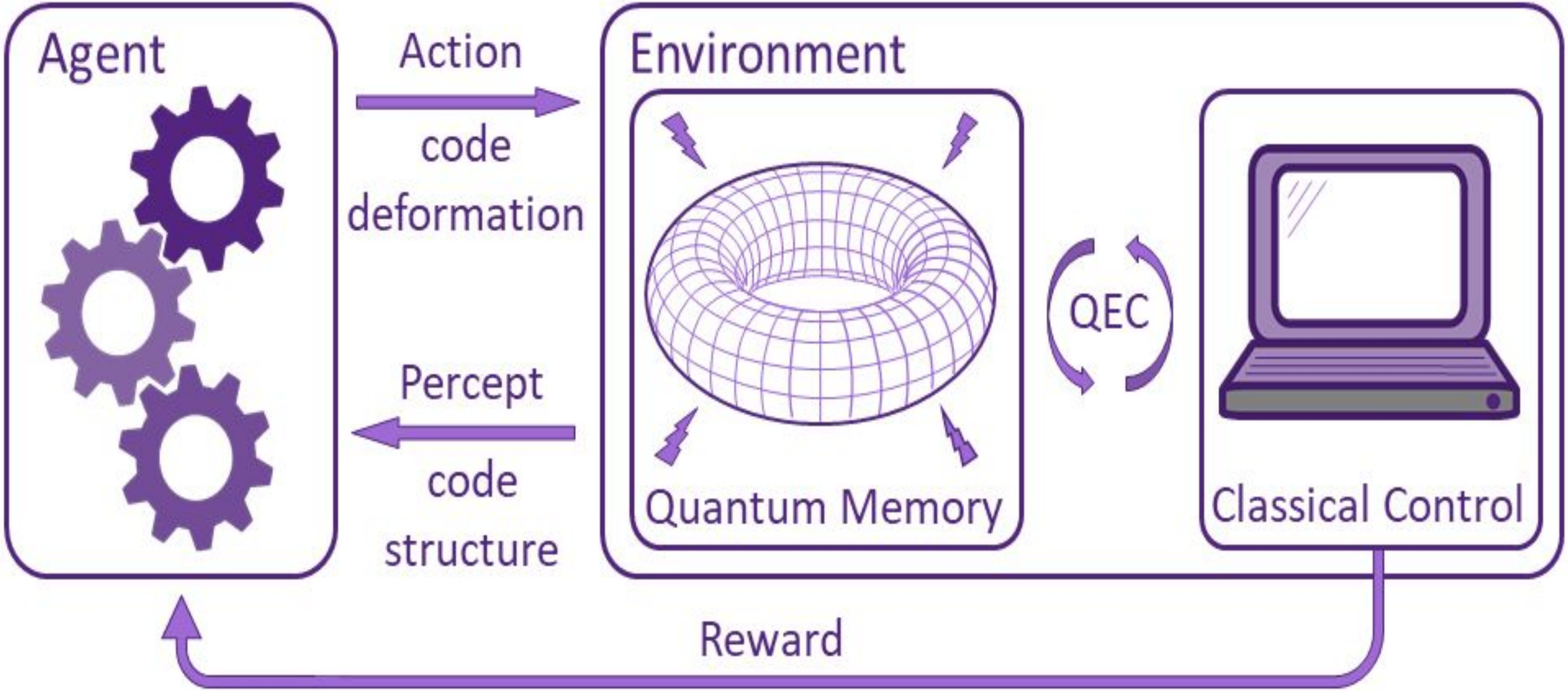
(a)



(b)

RL Framework

- **State:** Current code layout
- **Action:** Modify the layout (swap stabilizers, reconfigure)
- **Reward:** Negative of logical error rate (measured via decoder)
- **RL Agent:** Uses policy gradient methods to optimize code layout



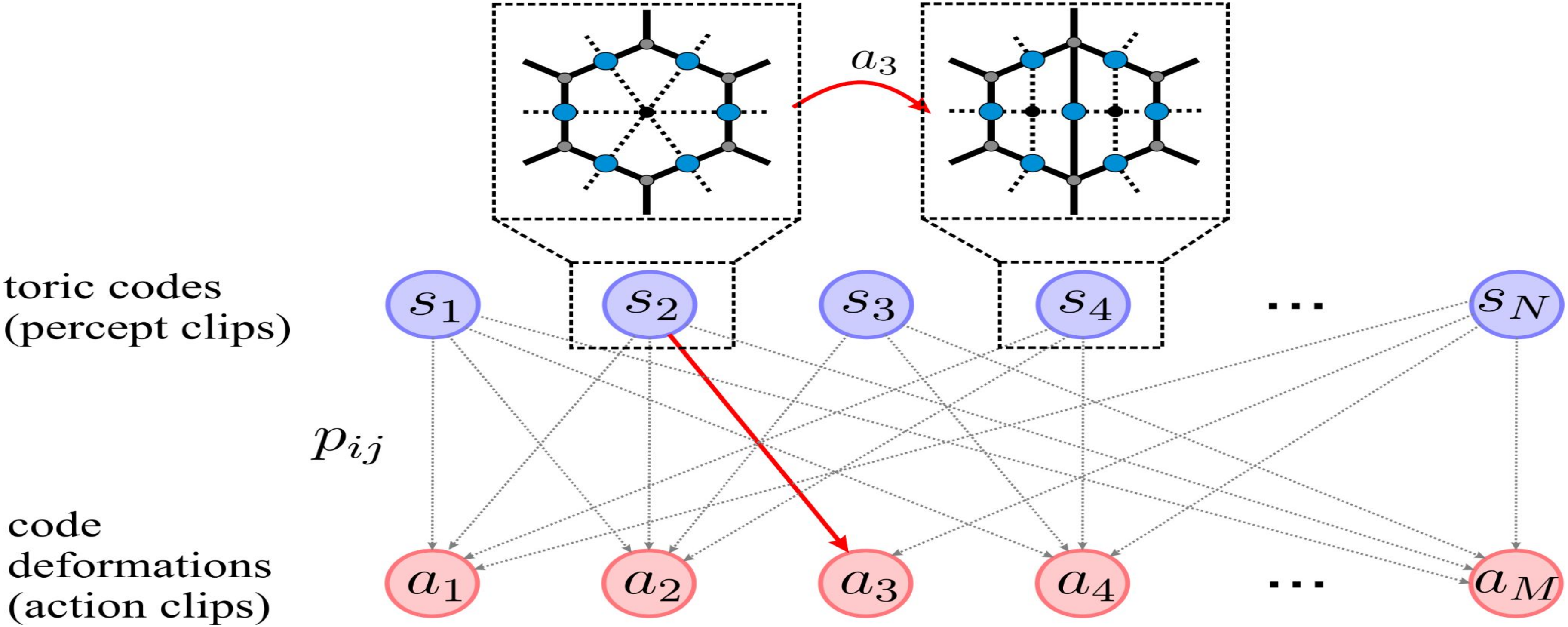
Illustrates the RL setup for layout optimization

Training and Evaluation

- RL trained on simulated quantum hardware.
- Evaluation metric: Logical error rate under given noise model.
- Uses **SQUAB** decoder (fast lookup-table decoder for toric codes).
- Compared standard vs. learned layouts.

Cont...

- Agent starts with an empty or simple memory layout.
- It performs actions (e.g., add connections or qubits).
- After each layout change, it simulates error correction using a decoder.
- Based on the logical error rate, it gets a reward.
- Over many episodes, it learns which layouts give low error rates — i.e., good quantum memories.

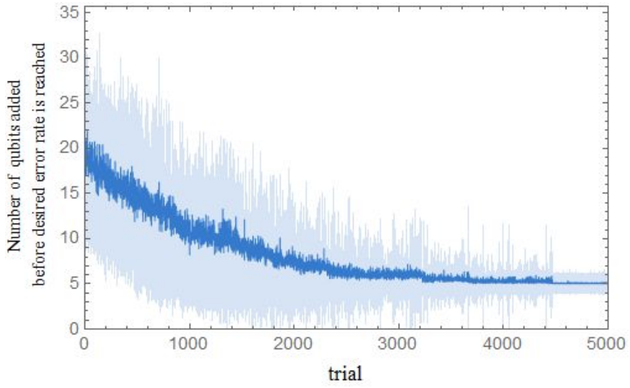


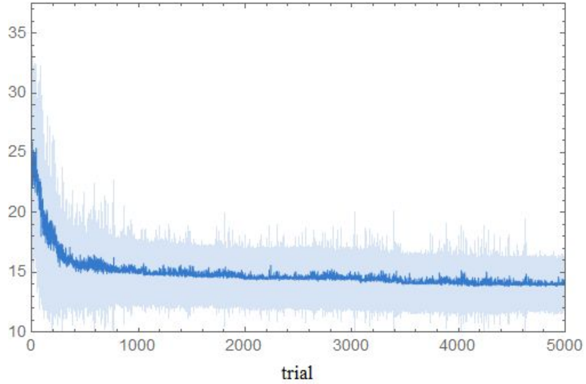
Two-layered clip network of a PS agent

Results and Discussion

- RL-discovered layouts consistently show **lower logical error rates** under specific biased noise conditions.
- **Figure:**
  - Performance improvement over standard code for various noise biases.
- Notable findings:
  - For Z-biased noise, layouts optimized by RL significantly outperform standard toric code.
  - RL learns to cluster Z-stabilizers in critical regions to better detect Z-errors.
- Improvement is **layout-specific**, showing RL effectively tailors the code to the noise model.
- Demonstrates that even small-scale modifications in layout can result in significant fault-tolerance gains.

Cont...





- **Task 1: Increasing Memory Size**
- **Task 2: Biased Noise Channel Adaptation**

Practical Implications

- Suggests a new paradigm: adaptive QEC codes optimized for each hardware's unique noise profile.
- Procedure:
  - Perform noise characterization on hardware.
  - Match with closest theoretical noise model.
  - Use pre-trained or further trained RL models to select optimal layout.
- Improves logical fidelity without needing more physical qubits or deeper circuits.

Conclusion

- Reinforcement learning enables **adaptive optimization** of surface code layouts.
- Learned layouts outperform standard codes under realistic noise models.
- Offers practical benefits in noisy intermediate-scale quantum (NISQ) devices.

Future Directions

- Extend to **larger surface codes**.
- Explore generalization to **multiple noise types**.
- Experiment with actual **quantum hardware** to validate.

References

H. S. Anwar, E. T. Campbell, D. E. Browne, "Optimizing Quantum Error Correction Codes with Reinforcement Learning,"
Link to paper: arXiv:1812.08451

Questions?

- Thank you!
- Happy to take questions.