QAMP 2025

Reinforcement learning based selection of error mitigation parameters for Zero Noise Extrapolation #56

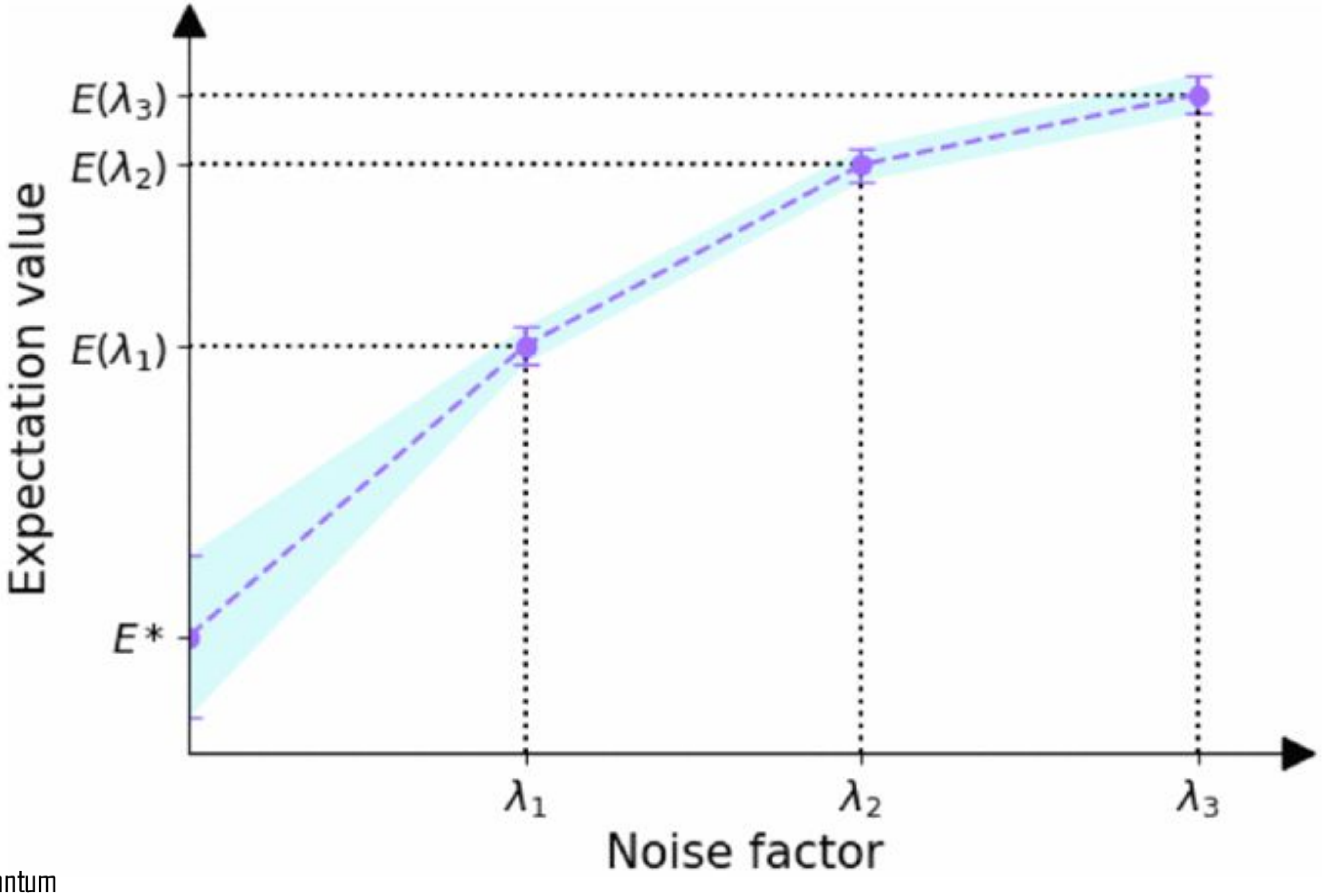
Speaker name: Abdullah K

Team member: Deenathayalan A

Mentors: Dr. Ritajit Majumdar, Dr. Anupama Roy

Introduction

In digital ZNE (dZNE), the "zero-noise" expectation value of the target circuit is extrapolated from the expectation values of mitigation circuits where the noise has been amplified by inserting additional digital quantum gates.



Ref: Majumdar et al. Best practices for quantum error mitigation with digital zero noise extrapolation

Motivation

Consider a mirrored QAOA circuit, where we are calculating the average expectation value of all weight-1 Z-type observables. Since it is a mirrored circuit, the ideal expectation value is +1.

We evaluate on Hamiltonian-simulation circuits using a Trotterized ansatz with varying trotter steps.

8×4 circuit, noise factors [1,3,5]

```
EXPECTATION VALUES:
  - 0.992008792128204 (linear)
  - 0.9861691478707368 (polynomial_degree_2)
  - 1.0228199598770722 (exponential)
```

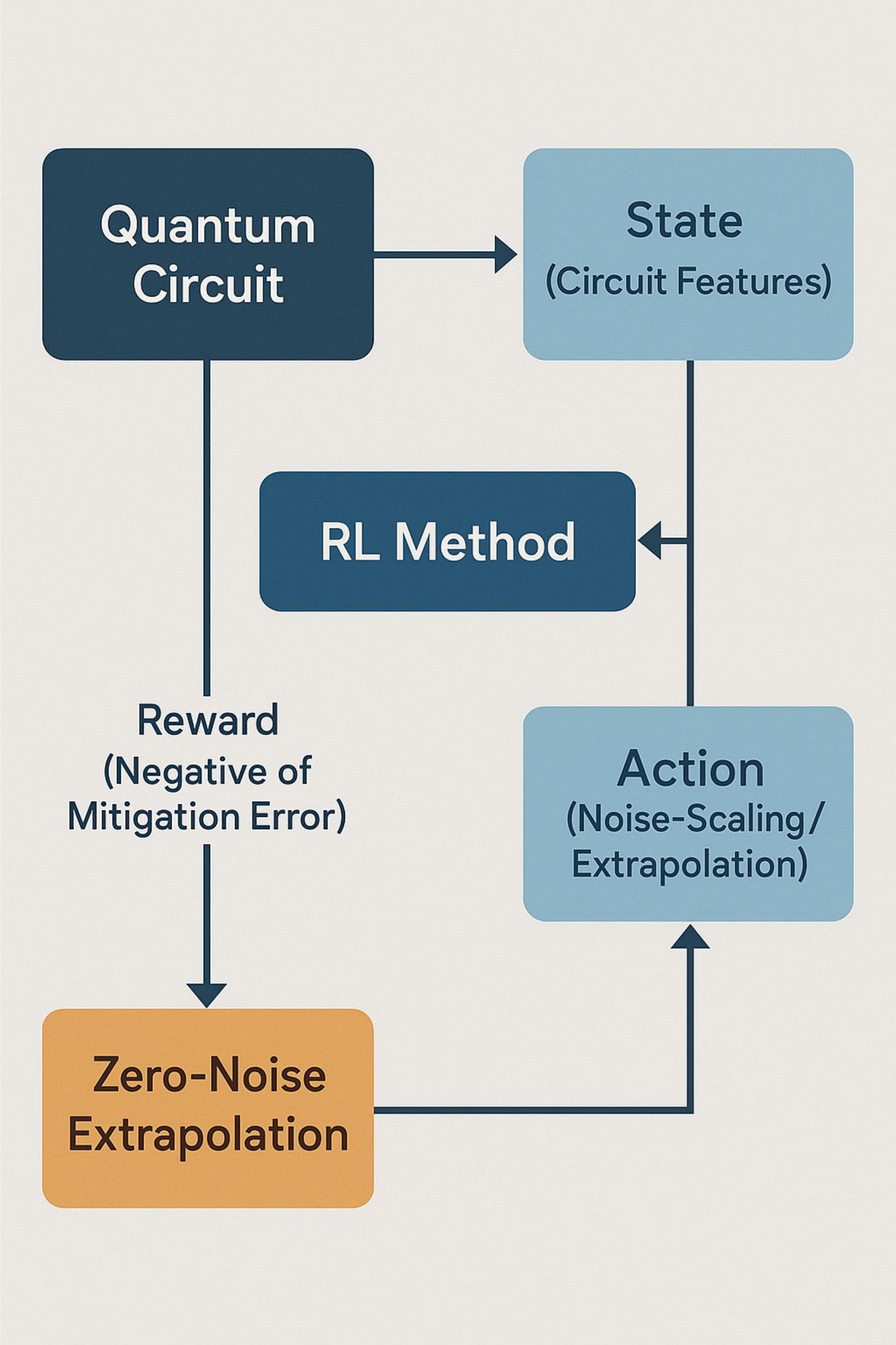
40×40 circuit

```
EXPECTATION VALUES:
  - 0.5814274221284977 (((1, 3, 5), 'linear'))
  - 23.8150497875442 (((1, 3, 5), 'exponential'))
  - 1.2644406687742398 (((1, 1.2, 1.4), 'linear'))
  - 5.484386280057926e+24 (((1, 1.2, 1.4), 'exponential'))
```

- The noise factors (λ) and the extrapolator (e) which works very well for an 8×4 circuit doesn't work well for the 40×40 one.
- **Question**: How does a user, who is not an expert or has not built the intuition regarding this, decide which noise factors and extrapolator (setting) to use for his/her problem?

Solution *: Use a reinforcement learning–based method that automatically selects the optimal noise-scaling factors and extrapolation strategy for each circuit.*

Workflow



Input: Quantum Circuit
- Parse the target circuit and extract structural info (qubit count, gate counts, depth, parameterized gates, CX density, etc.).

State (Circuit Features)
- Convert extracted metrics into a normalized state vector that represents the circuit for the RL agent.

RL Method
- The RL agent receives the state and selects an action. (We keep this generic so different RL algorithms can be tested.)

Action: Noise-Scaling / Extrapolation Choice
- Action defines the noise-amplification template (e.g., stretch/folding factors) and the extrapolation family (Richardson / polynomial / exponential).

Zero-Noise Extrapolation (ZNE) Execution
- Run the scaled/noisier circuits, collect measurements, and apply the chosen extrapolation to estimate the zero-noise value.

Reward & Learning
- Compute reward as negative mitigation error (reward = −error). Feed reward back to the RL agent to update policy. Store transitions for replay / training.

Loop & Evaluation
- Repeat across episodes; log metrics and compare against baseline extrapolations for validation and model selection.

Note *: To scale up we plan to use cliffordization so parts of the pipeline can be validated on larger circuits.*

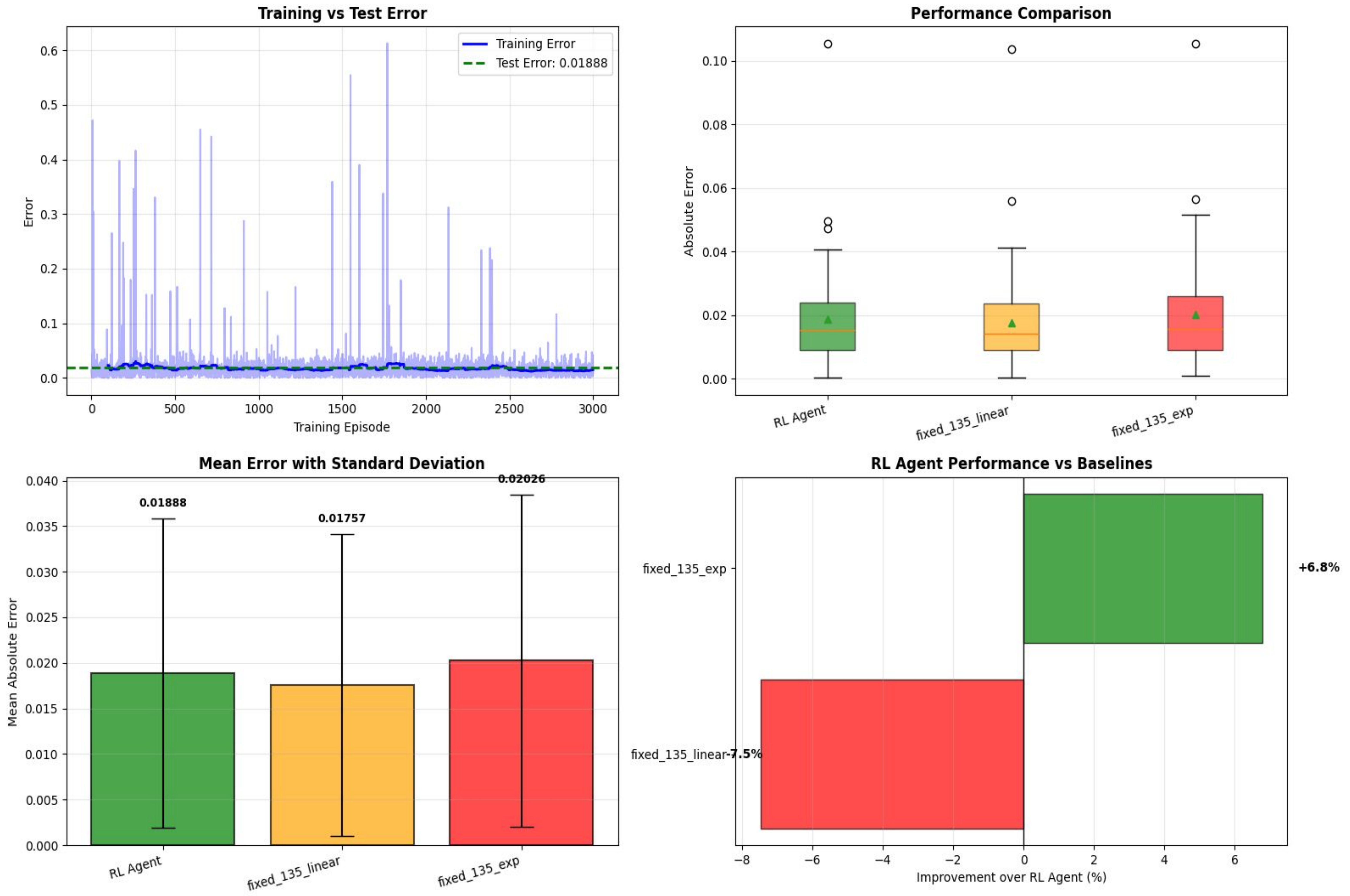
RL Agent vs Baselines (Preliminary Results)

Stable Learning:
- Training error decreases over episodes and test error stabilizes around **0.0188**, indicating consistent policy learning.

Comparable Performance:
- Box-plots show the RL agent performs **similarly** to baseline templates (fixed_135_linear, fixed_135_exp) with overlapping error distributions.

Mean Error Analysis:
- Average absolute error is close across methods — RL (**0.01888**), linear baseline (**0.01757**), exponential baseline (**0.02026**).

Relative Improvement:
- RL outperforms exponential baseline by **+6.8%**, while linear baseline performs **~7.5% better** than RL



- *Observed top RL-selected templates (most-used): [1.0, 3.5, 6.0] (polynomial) and [1.0, 4.0, 5.0] (polynomial).*
- ***Note:** linear [1,3,5] still outperformed RL in this run; we are tuning the RL policy (action space / reward shaping / curriculum) to find optimal results.*

QISKIT ADVOCATE

2025 2025

MENTORSHIP PROGRAM

THANK YOU