DyREM: Dynamically Mitigating Quantum Readout Error with Embedded Accelerator

Kaiwen Zhou[1] , Liqiang Lu[1] * , Hanyu Zhang[1] , Debin Xiang[1] , Chenning Tao[1], Xinkui Zhao[1], Size Zheng[2], Jianwei Yin[1]

[1]Zhejiang University, [2]ByteDance Ltd

Abstract

Quantum readout error significantly reduces measurement fidelity. Existing methods face high latency due to the dynamic generation of mitigation matrices. DyREM proposes a software-hardware co-design approach that mitigates errors by exploiting sparsity in the nonzero probability distribution of quantum states and calculating the tensor product on an embedded accelerator. The dataflow of DyREM dynamically downsamples the original mitigation matrix, dramatically reducing memory requirements. The accelerator architecture of DyREM also flexibly gates redundant computation. Experiments demonstrate that DyREM outperforms existing methods in both speed (9.6x to 2000x speedup) and fidelity (1.03x to 1.15x improvement).

Motivation

As shown in Figure 1, the motivation of DyREM mainly consists of two parts:

- **Long Mitigation Latency.** Prior software mitigation methods, such as Mthree, IBU, and QuFEM, suffer from long latency, which accounts for the majority of end-to-end latency for executing a quantum circuit, hindering quantum supremacy. Although SpREM proposes a hardware architecture to accelerate the process, it suffers from poor scalability.
- **Static Qubit Group.** Prior methods, such as IBU and QuFEM, apply a static qubit group for readout error mitigation. However, the actual measured qubits dynamically change in different circuit executions, so these methods fail to mitigate readout error.

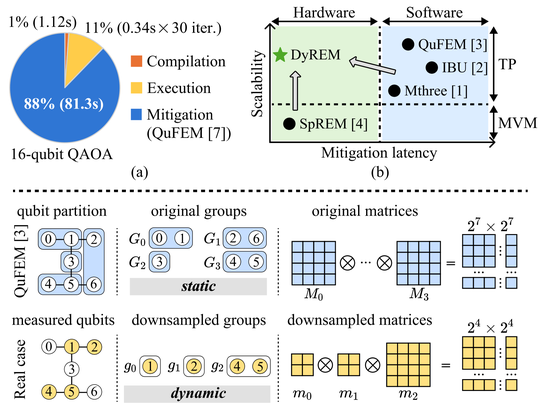

Figure 1: Motivation of DyREM.

DyREM Dataflow Overview

To address the above problems, we propose DyREM dataflow, as shown in Figure 2. The input consists of measured qubits and a noisy distribution. The output is a mitigated distribution. The dataflow can be divided into three steps:

- **Downsampled Groups Generation:** We partition the measured qubits according to the original groups. Then, we can obtain downsampled groups, which serve as the input for matrix downsampling.
- **Matrix Downsampling:** We categorize the downsampled groups into three types. Then, we adopt three strategies to generate the downsampled matrices, serving as the input for the nonzero state-oriented computation.
- **Nonzero State-oriented Computation:** We first obtain the output sparsity from the noisy distribution. Then, we compress the full-dimensional matrix based on the output sparsity. Next, we calculate the mitigation matrix by selecting the required elements from the downsampled matrices and performing multiplication.

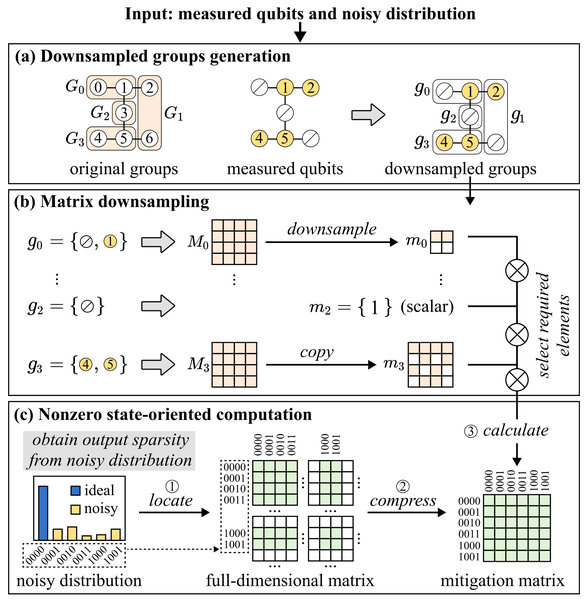

Figure 2: The details of DyREM dataflow.

Nonzero State-Oriented Computation

To enable on-chip mitigation matrix calculation, we propose nonzero state-oriented computation, comprising two steps:

- **Mitigation Matrix Compression:** We observe that the ideal distribution usually lies in the noisy distribution, which only contains a few nonzero values. Therefore, we exploit this output sparsity from the noisy distribution to compress the mitigation matrix.
- **Nonzero State Similarity Detection:** We identify the similarity by partitioning the states into different windows and computing a similar table for our accelerator. We avoid redundant calculations within each window to effectively calculate the mitigation matrix.

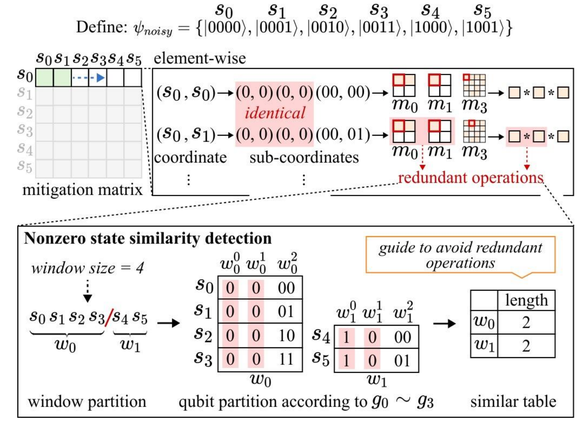

Figure 3: Illustration of nonzero state similarity detection.

DyREM Hardware Architecture

We design an accelerator architecture to implement DyREM, primarily consisting of two components:

- **Nonzero State Detector:** The nonzero state detector efficiently calculates the length of the longest identical segment for each window. It integrates an early termination unit with an adder. The early termination unit consists of a comparator and a NOT gate.
- **Mitigation Core Array:** The mitigation core array receives the values from downsampled matrices and noisy probabilities. Each mitigation core (MC) employs a three-layer multiplier to compute the mitigation matrix and the MVM step effectively. Within the MC, we set up a FIFO for each multiplier (1st layer) to stockpile the values from downsampled matrices. In particular, the FIFO in the first row is used to calculate the reused data.

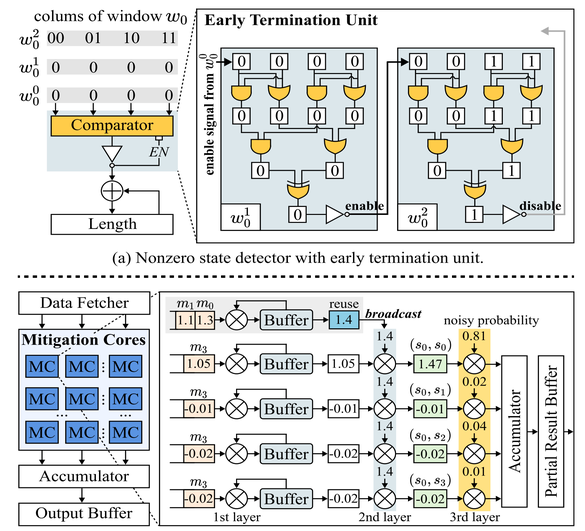

Figure 4: The detailed hardware design of DyREM.

Evaluation

- **Mitigation Latency:** DyREM achieves an average speedup of 9.6X ~ 2000X.
- **Q-throughput (states/s):** DyREM achieves an average improvement of 1.5X ~ 2726X.
- **Fidelity**: DyREM achieves average improvements of 1.15X, 1.13X, 1.09X, and 1.03X.

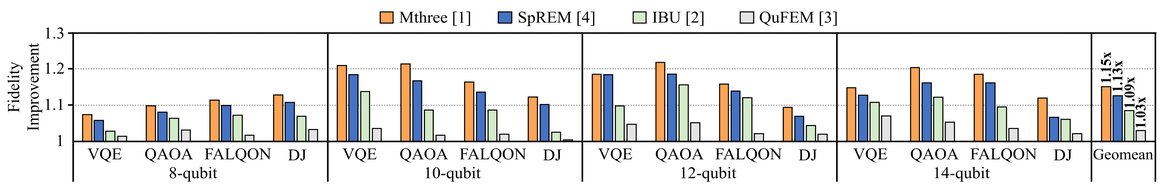

Figure 5: The fidelity improvement over baselines.

References

[1] P. D. Nation et al., "Scalable mitigation of measurement errors on quantum computers," PRX Quantum, vol. 2, p. 040326, 2021.

[2] B. Pokharel et al., "Scalable measurement error mitigation via iterative bayesian unfolding," Phys. Rev. Res., vol. 6, p. 013187, 2024.

[3] S. Tan et al., "Qufem: Fast and accurate quantum readout calibration using the finite element method," in ASPLOS, 2024, p. 948–963.

[4] H. Zhang et al., "Sprem: Exploiting hamming sparsity for fast quantum readout error mitigation," in DAC, 2024.

Welcome to Follow!

Personal Website: https://kaiwenzhou2003.github.io/

Advisor's Website (Prof. Liqiang Lu): https://liqianglu-zju.github.io/

HPCA 2025 Tutorial: https://janusq.github.io/HPCA_2025_Tutorial/home

My Email: kaiwenzhou@zju.edu.cn