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

THE CHIPS TO SYSTEMS CONFERENCE

SPONSORED BY

Kaiwen Zhou

First-year Ph.D. student in CS, Zhejiang University

Research interests:

- Quantum Computer Architecture
- Quantum Error Correction

Advisor: **Prof. Liqiang Lu**

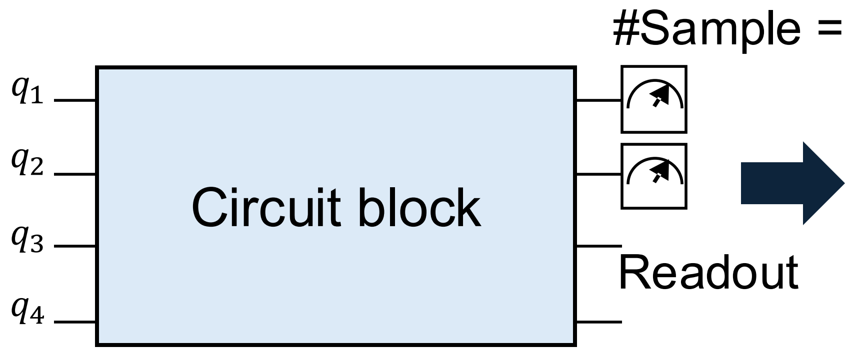
Homepage: https://liqianglu-zju.github.io/

Outline of Presentation

- **Background**
- Motivation
- DyREM Dataflow
- Architecture Design
- Evaluation

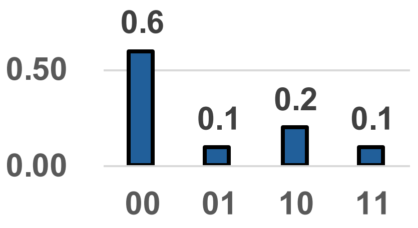
Quantum Readout

- Quantum readout **reads the information from quantum bits to classical bits**.



sample	1	2	3	4	4	…	4999	5000
q_1	0	1	1	1	0	…	1	0
q_2	1	0	1	1	1		1	1

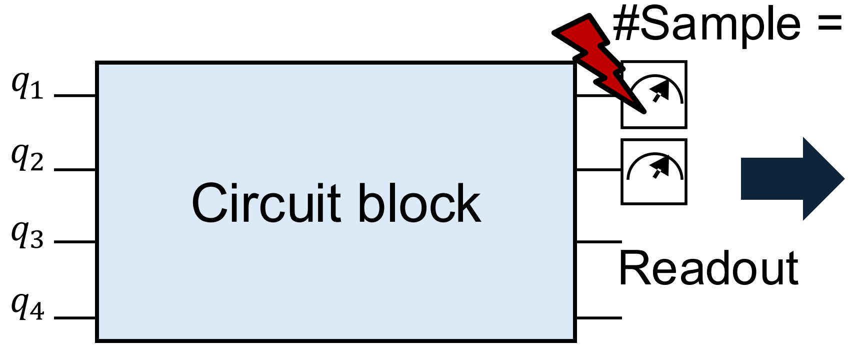
Measured bitstrings in different samples



Probability distribution

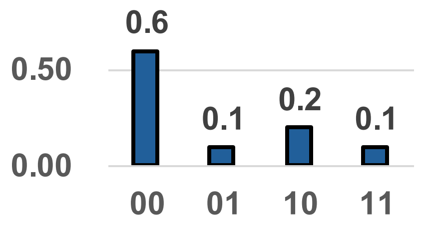
Quantum Readout

- Quantum readout **reads the information from quantum bits to classical bits**.



sample	1	2	3	4	4	…	4999	5000
q_1	0	1	1	1	0	…	1	0
q_2	1	0	1	1	1		1	1

Measured bitstrings in different samples



Probability distribution

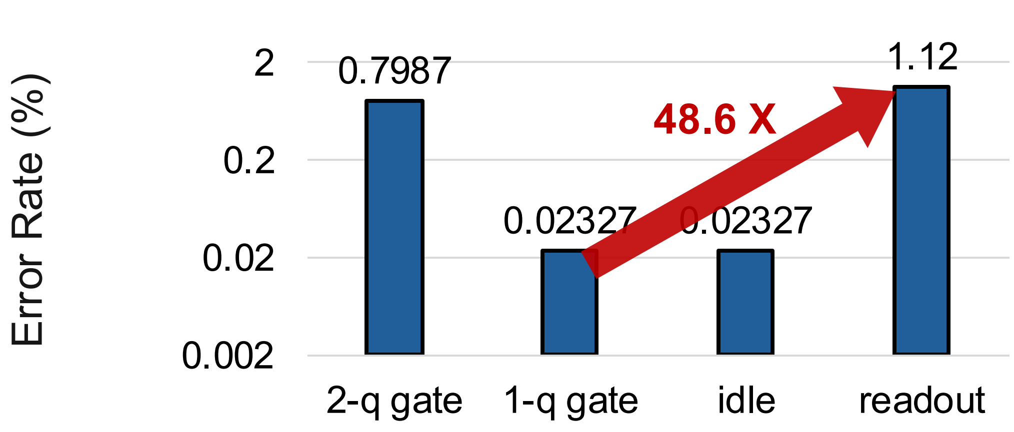
However, the quantum readout process suffers from readout error.

Readout Error

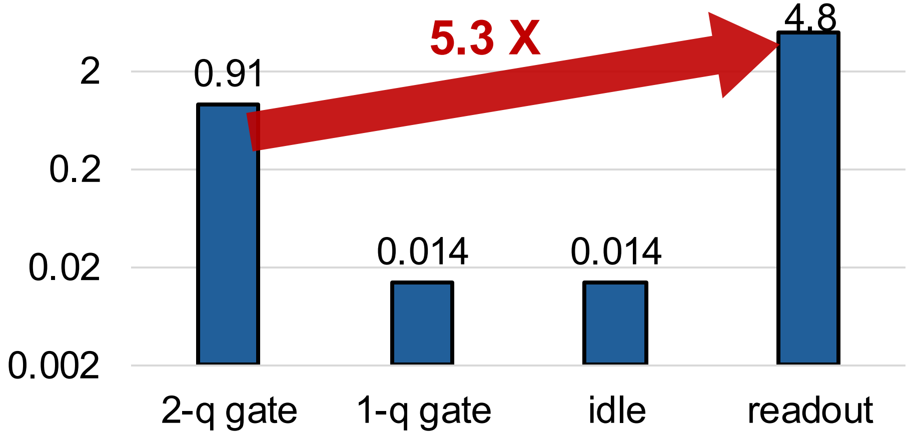
- Error sources

Long readout latency | Crosstalk | Imperfect discriminator

- Readout error is **significant on current quantum hardware.**



Noise on 127-qubit IBM Sherbrooke quantum device



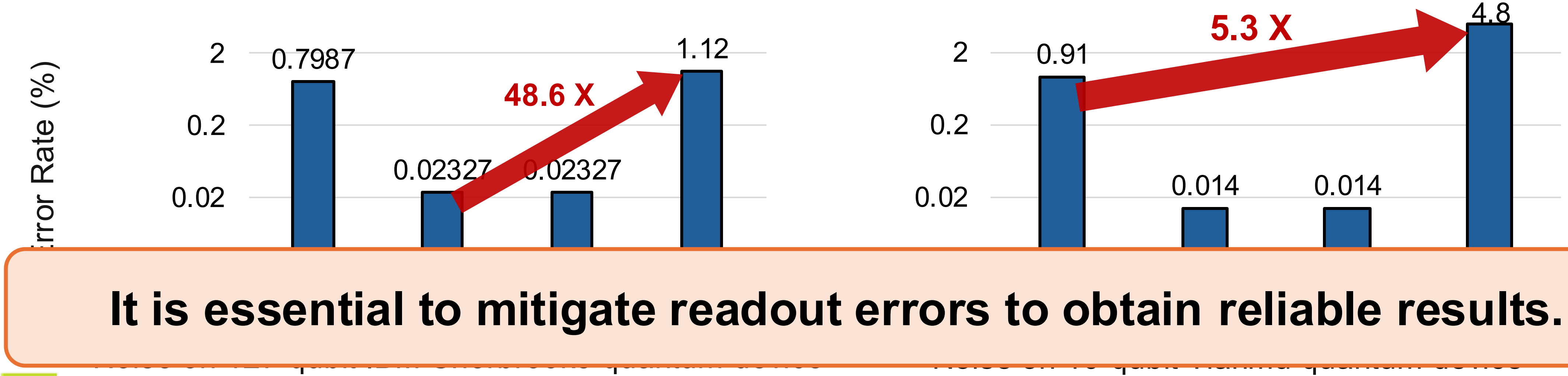
Noise on 10-qubit Tianmu quantum device

Readout Error

- Error sources

Long readout latency | Crosstalk | Imperfect discriminator

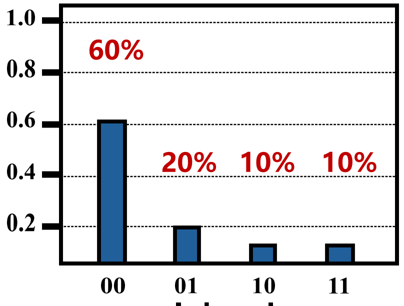
- Readout error is **significant on current quantum hardware.**



It is essential to mitigate readout errors to obtain reliable results.

Matrix-Based Readout Mitigation

Readout with Noise



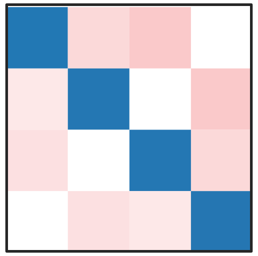
Ideal distribution P_{ideal}

$$\begin{bmatrix} 0.5 \\ 0.1 \\ 0.09 \\ 0.31 \end{bmatrix}$$

Noisy distribution P_{noisy}

$$P_{noisy} = N \cdot P_{ideal}$$

Perform Matrix-Vector Multiplication



$$\bullet \begin{bmatrix} 0.5 \\ 0.1 \\ 0.09 \\ 0.31 \end{bmatrix} = \begin{bmatrix} 0.6 \\ 0.2 \\ 0.1 \\ 0.1 \end{bmatrix}$$

Mitigation matrix M

Noise distribution P_{noisy}

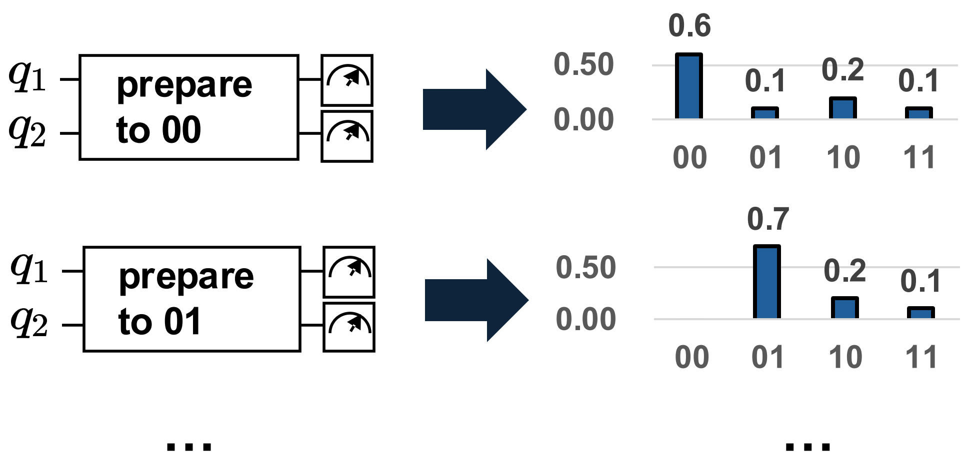
Mitigated distribution P_{miti}

$$P_{miti} = M \cdot P_{noisy}$$

Matrix-Based Readout Mitigation

3 Steps to Obtain Mitigation Matrix M

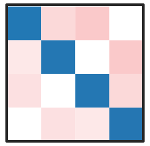
① Prepares qubits to different basis states and apply measurement.



② Fill the noisy matrix

$$N = \begin{bmatrix} 0.6 & 0.1 & 0.2 & 0.1 \\ 0 & 0.7 & 0.2 & 0.1 \\ 0.2 & 0.2 & 0.6 & 0 \\ 0 & 0.1 & 0.1 & 0.8 \end{bmatrix}$$

③ Inverse the noisy matrix

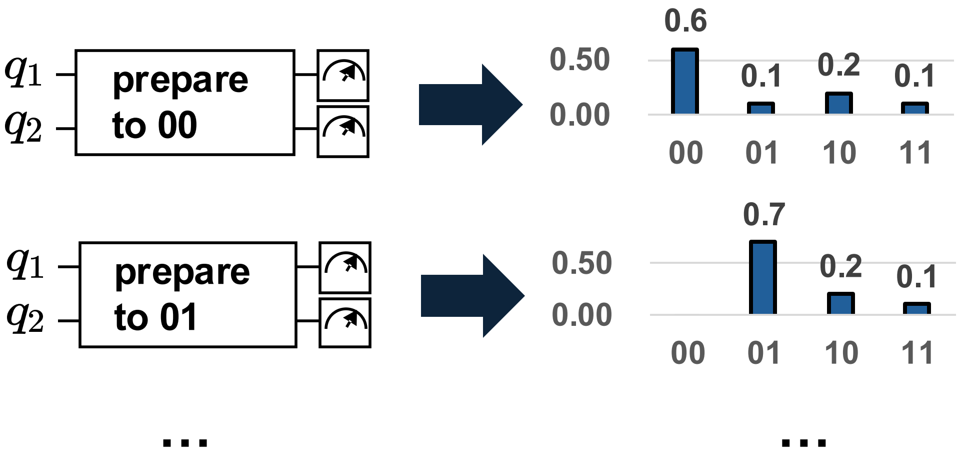
$M = N^{-1} =$ 

Mitigation matrix

Matrix-Based Readout Mitigation

3 Steps to Obtain Mitigation Matrix M

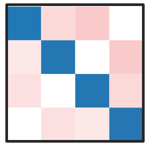
① Prepares qubits to different basis states and apply measurement.



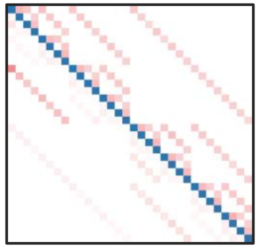
② Fill the noisy matrix

$$N = \begin{bmatrix} 0.6 & 0.1 & 0.2 & 0.1 \\ 0 & 0.7 & 0.2 & 0.1 \\ 0.2 & 0.2 & 0.6 & 0 \\ 0 & 0.1 & 0.1 & 0.8 \end{bmatrix}$$

③ Inverse the noisy matrix

$M = N^{-1} =$ 

Mitigation matrix



5-qubit mitigation matrix: **$2^5 \times 2^5$**

10-qubit mitigation matrix: **$2^{10} \times 2^{10}$**

⋮

The size exponentially increases! ☹

Tensor-Product-Based Readout Mitigation

Key Idea: Tensor-Product Approximation

$$P_{miti} = M \cdot P_{noisy}$$

⬇

$$P_{miti} = \underbrace{(M_1 \otimes M_2 \otimes \cdots \otimes M_k)}_{\text{sub-mitigation matrices}} \cdot P_{noisy}$$

Each submatrix shows exponential reduction in size. ☺

Tensor-Product-Based Readout Mitigation

Key Idea: Tensor-Product Approximation

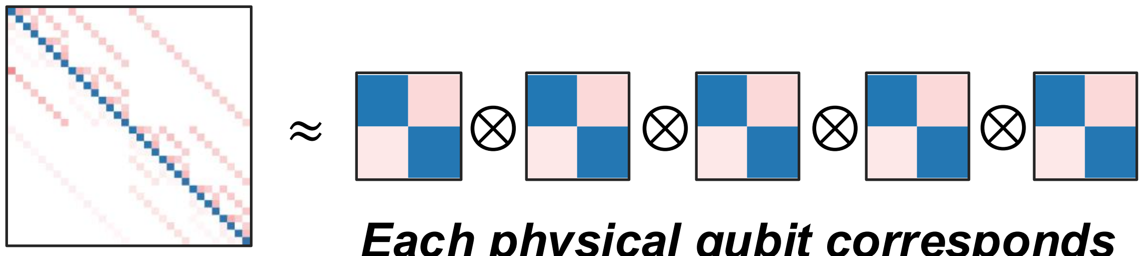
$$P_{miti} = M \cdot P_{noisy}$$

$$P_{miti} = \underbrace{(M_1 \otimes M_2 \otimes \cdots \otimes M_k)}_{\text{sub-mitigation matrices}} \cdot P_{noisy}$$

Each submatrix shows exponential reduction in size.

Qubit Group and Sub-Mitigation Matrix

IBM Mthree [1] and Google IBU [2].



Each physical qubit corresponds to a 2x2 meta matrix.

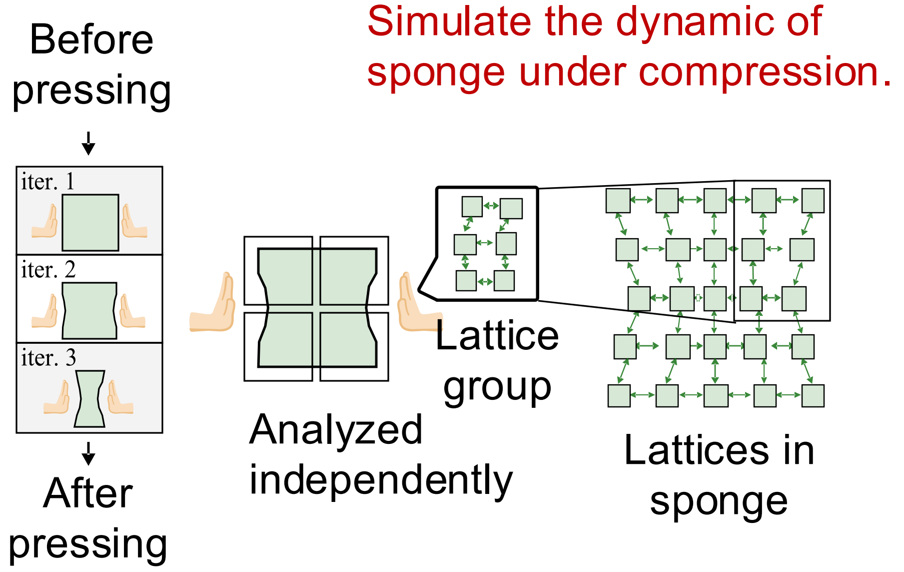
Good scalability, **low fidelity (crosstalk-unaware)**.

[1] Nation, Paul D., et al. "Scalable mitigation of measurement errors on quantum computers." PRX Quantum 2.4 (2021): 040326.
[2] Pokharel, Bibek, et al. "Scalable measurement error mitigation via iterative bayesian unfolding." Physical Review Research 6.1 (2024): 013187.

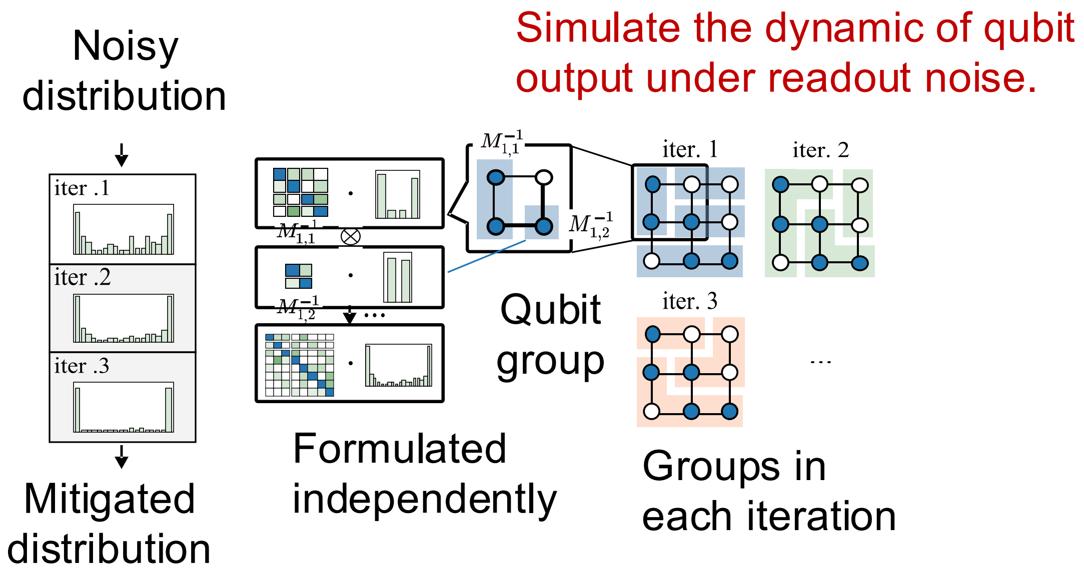
Our Prior Work on Readout Mitigation

- **QuFEM (ASPLOS 2024)**

Classical Finite Element Method



Quantum Finite Element Method

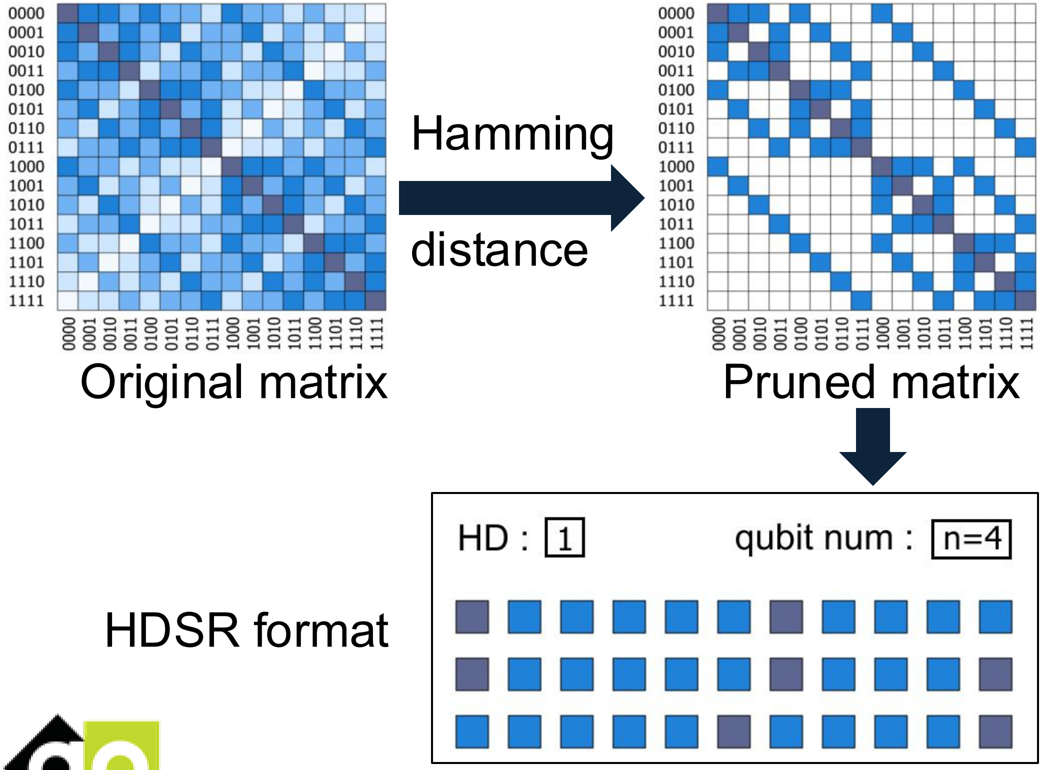


A **divide-and-conquer strategy** to mitigate the noisy distribution.

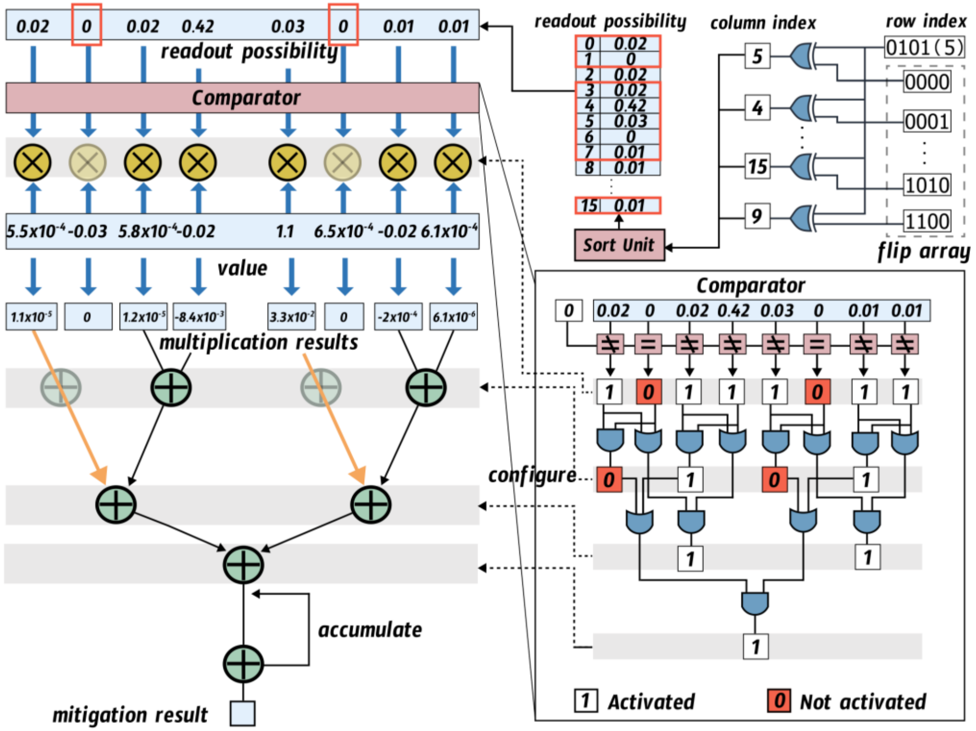
Our Prior Work on Readout Mitigation

- **SpREM (DAC 2024)**

Pruning based on Hamming Distance



Hardware Architecture



Outline of Presentation

- Background
- **Motivation**
- DyREM Dataflow
- Architecture Design
- Evaluation

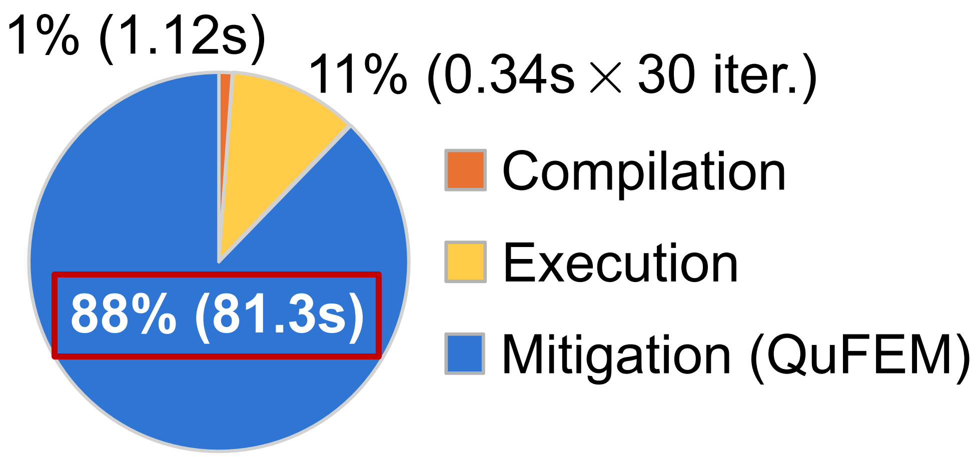
THE CHIPS TO SYSTEMS CONFERENCE
62

SPONSORED BY CEDA sigda

Challenge 1: Long Latency

Breakdown of End-to-End Latency

- 16-qubit QAOA
- Iteration: 30 times
- 3 stages:
  - Compilation, Execution, Mitigation
- Platforms:
  - Quantum: 156-qubit IBM_fez processor
  - Classical: AMD EPYC 9554 64-core CPU

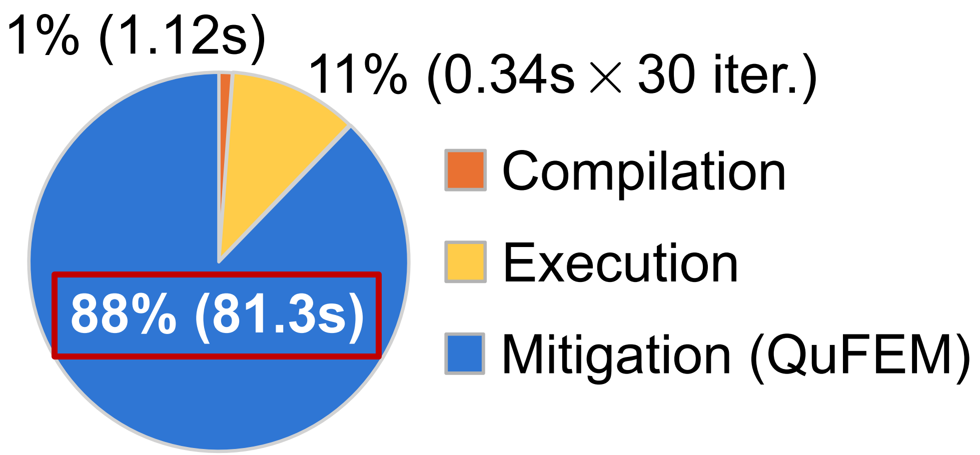


Classical-side mitigation dominates the runtime!

Challenge 1: Long Latency

Breakdown of End-to-End Latency

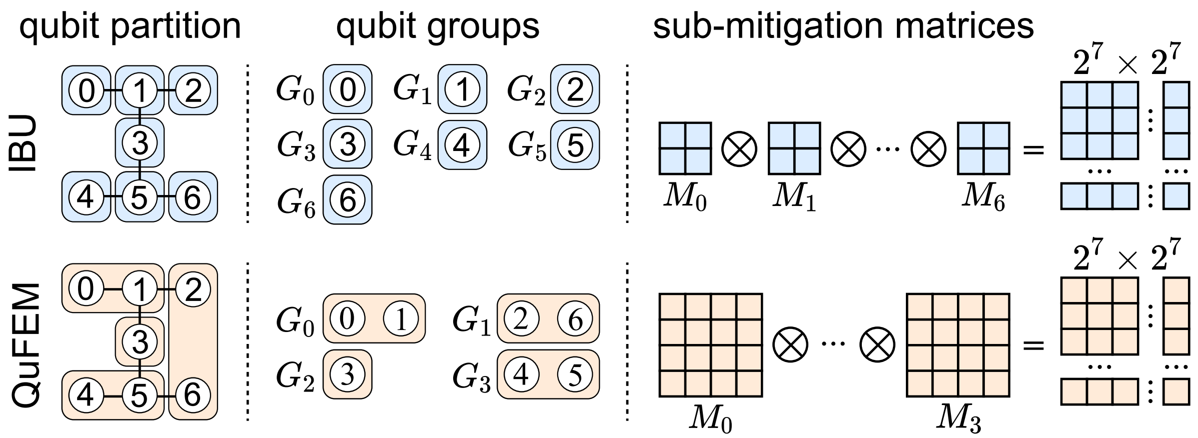
- 16-qubit QAOA
- Iteration: 30 times
- 3 stages:
  - Compilation, Execution, Mitigation
- Platforms:
  - Quantum: 156-qubit IBM_fez processor
  -



Goal: Reduce the mitigation time with a dedicated accelerator.

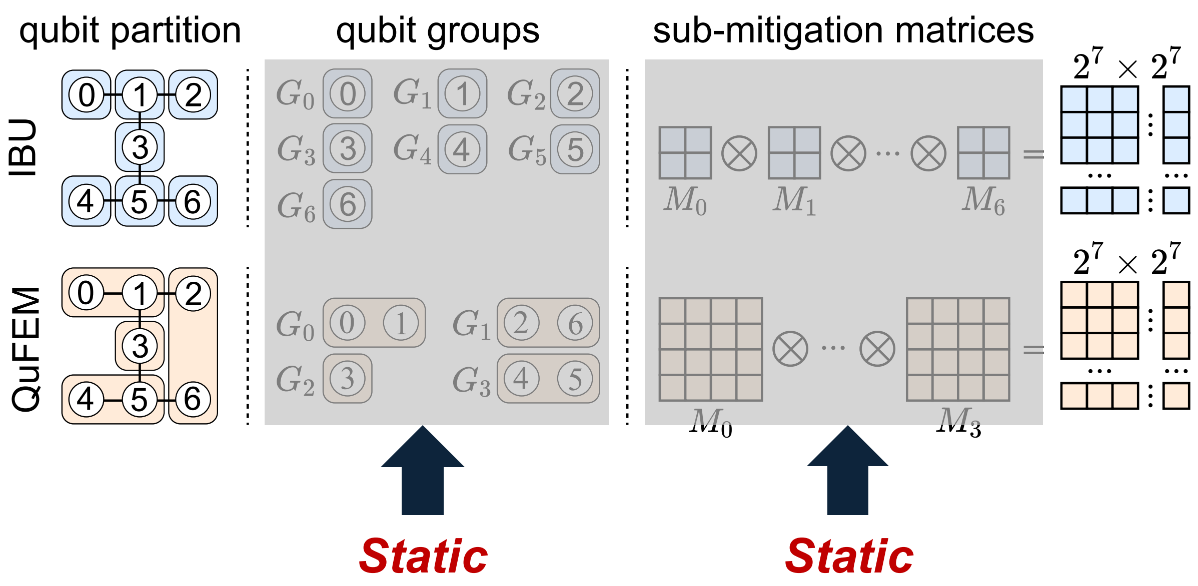
Challenge 2: Static Qubit Group

Qubit Groups of Prior Works



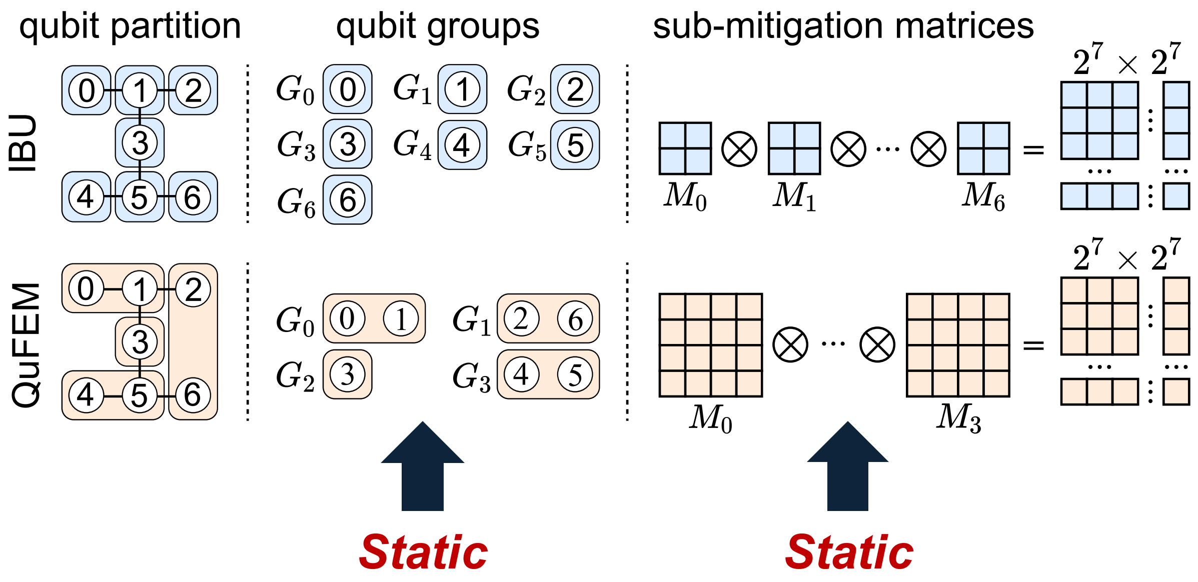
Challenge 2: Static Qubit Group

Qubit Groups of Prior Works

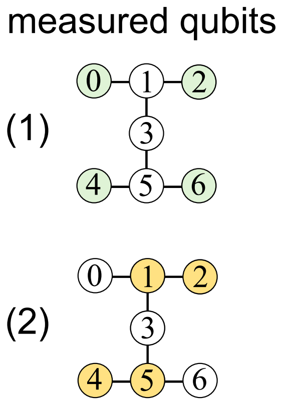


Challenge 2: Static Qubit Group

Qubit Groups of Prior Works



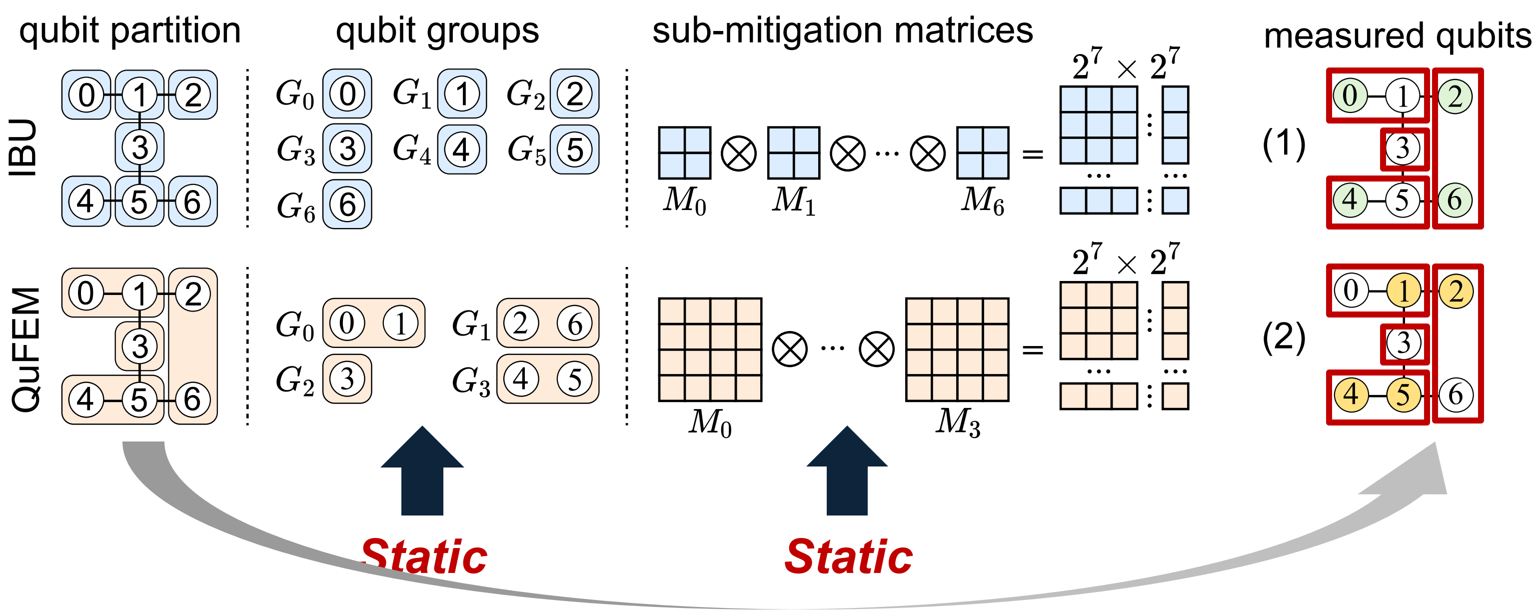
Dynamically Changing Measured Qubits



Challenge 2: Static Qubit Group

Qubit Groups of Prior Works

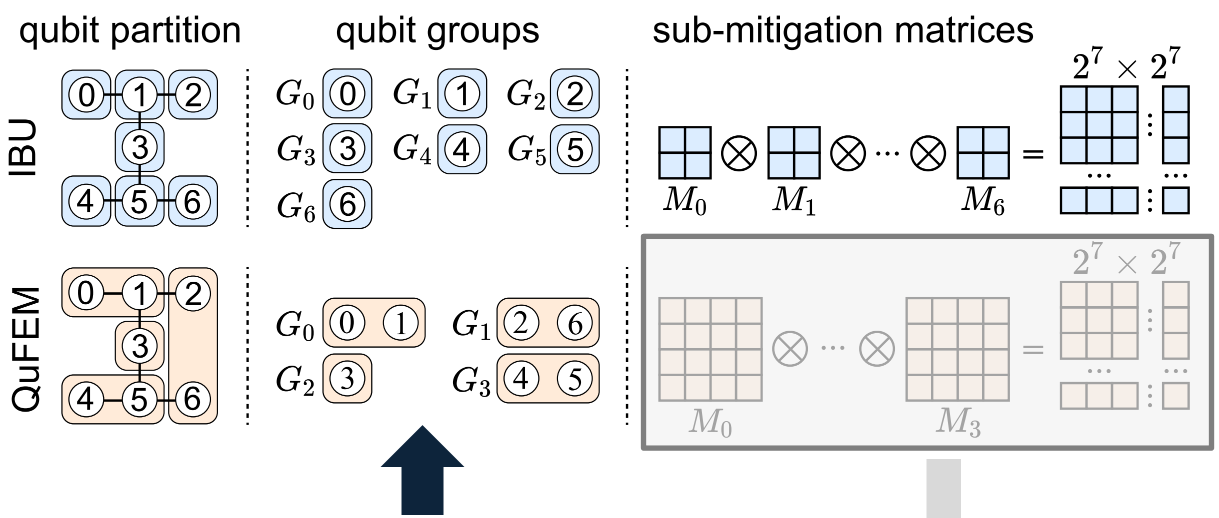
Dynamically Changing Measured Qubits



Apply the grouping scheme of QuFEM
(state-of-the-art)

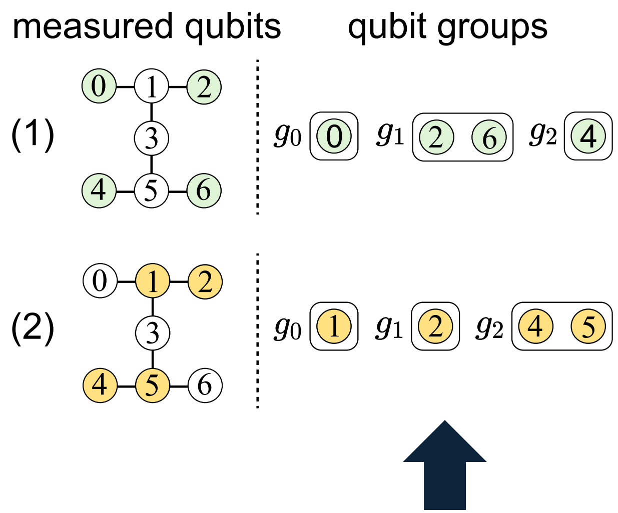
Challenge 2: Static Qubit Group

Qubit Groups of Prior Works



Static

Dynamically Changing Measured Qubits

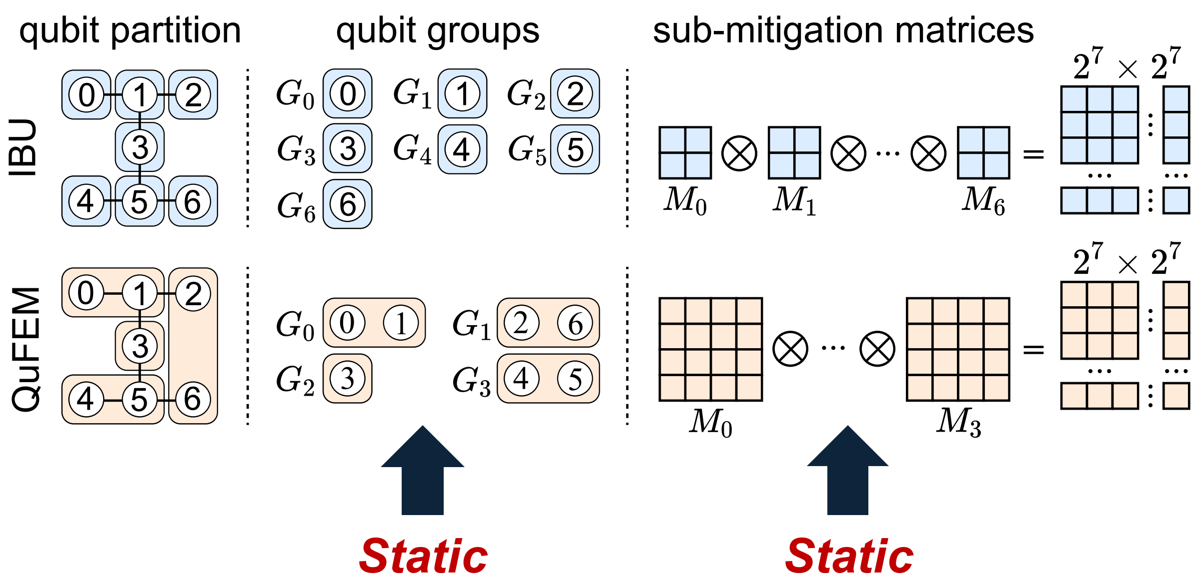


Dynamic

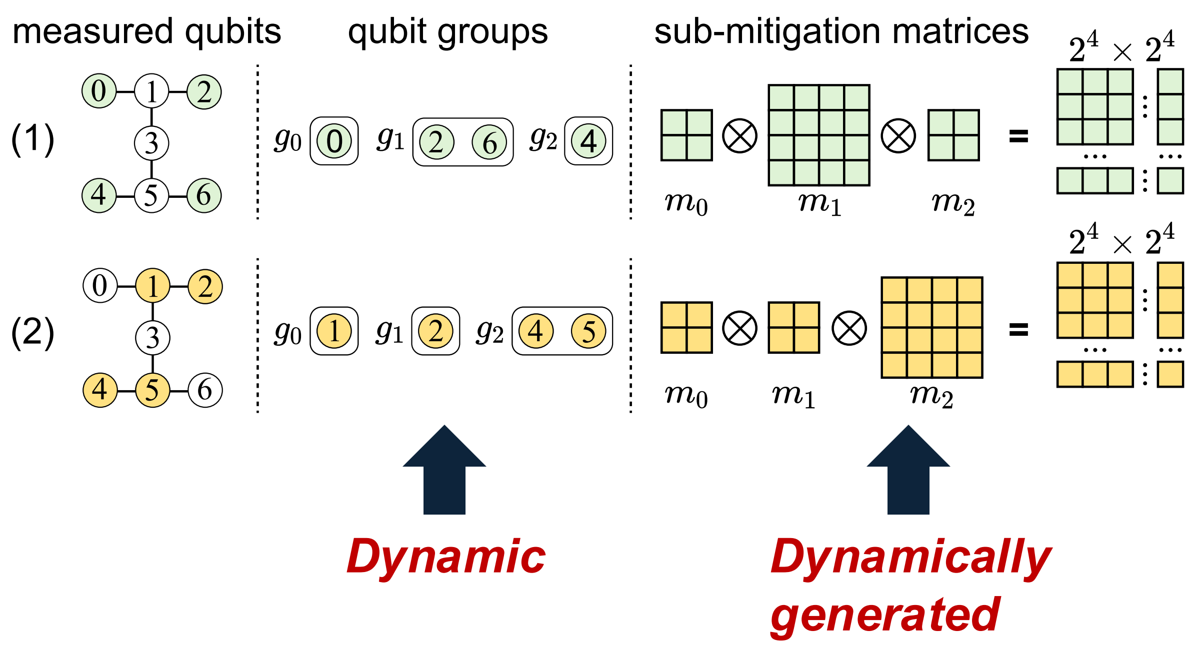
Pre-determined matrices can not be reused ☹

Challenge 2: Static Qubit Group

Qubit Groups of Prior Works

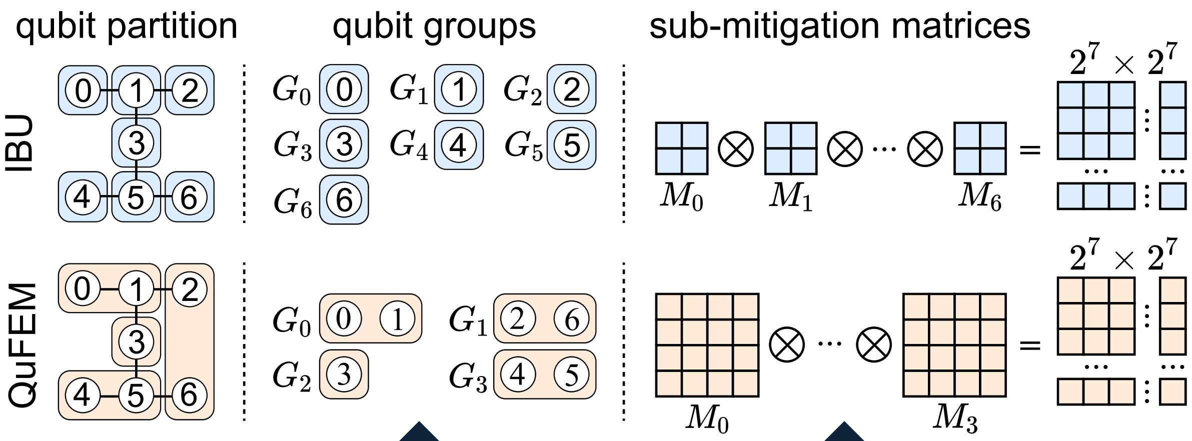


Dynamically Changing Measured Qubits

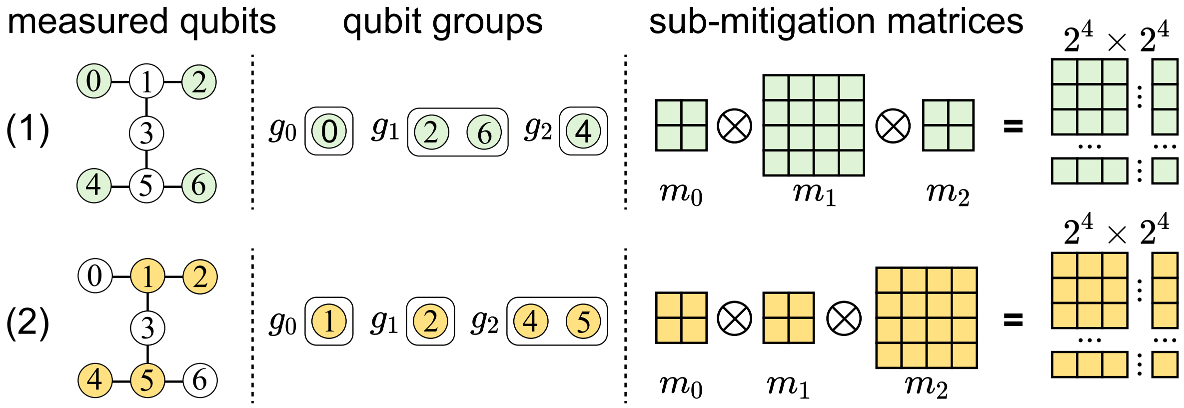


Challenge 2: Static Qubit Group

Qubit Groups of Prior Works



Dynamically Changing Measured Qubits

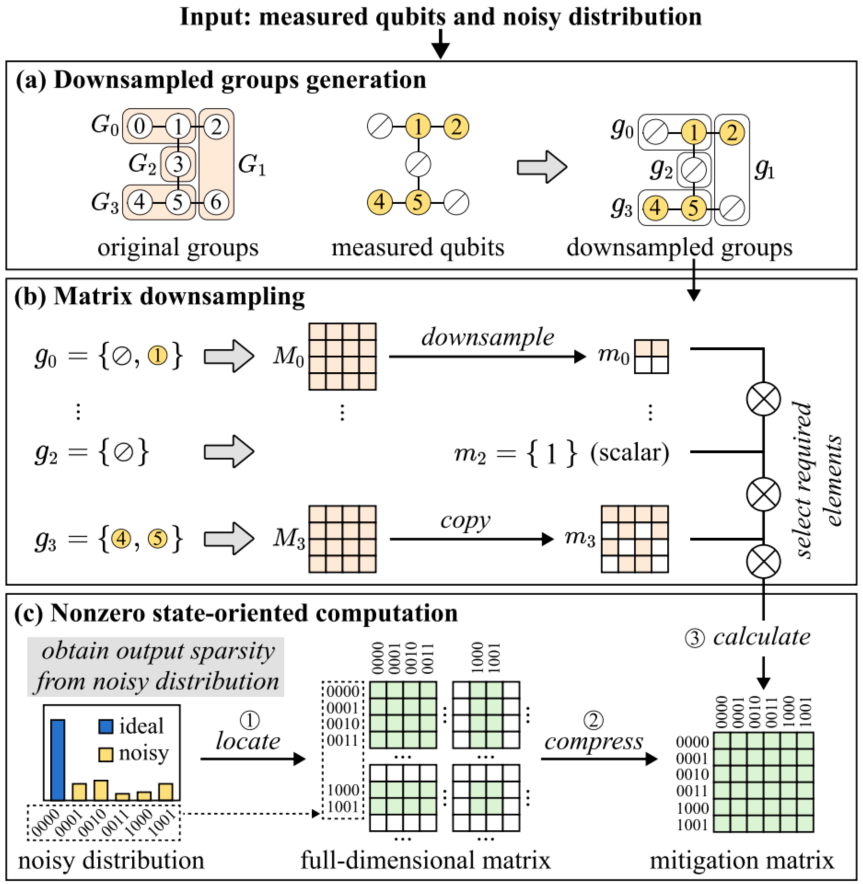


Goal: Enable the dynamic readout error mitigation.

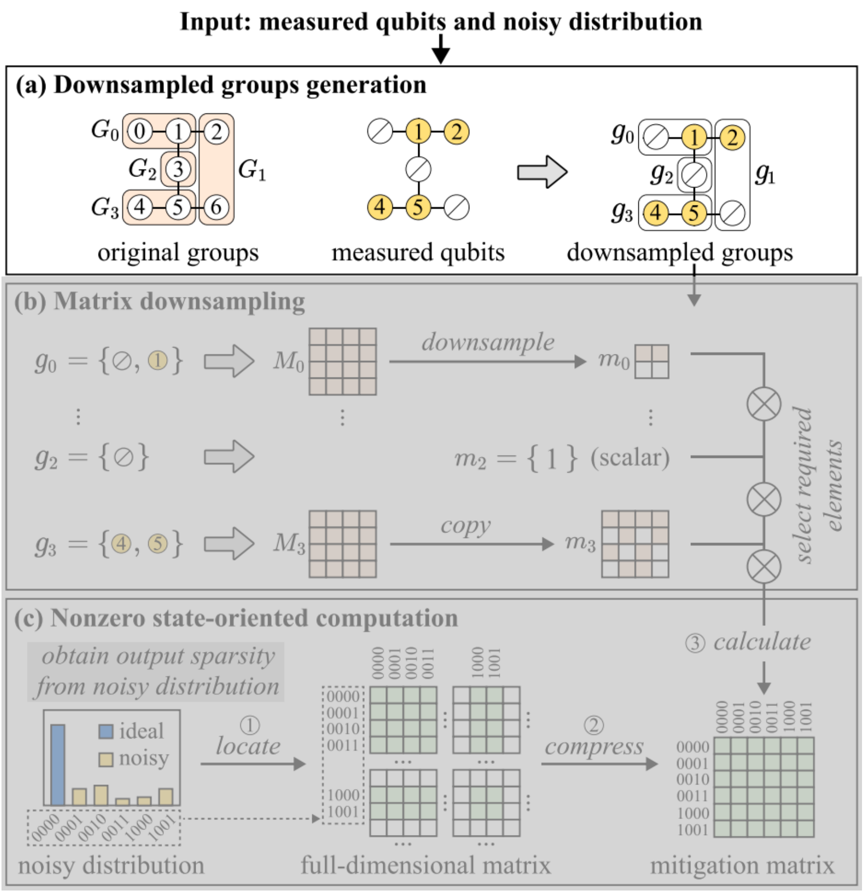
Outline of Presentation

- Background
- Motivation
- **DyREM Dataflow**
- Architecture Design
- Evaluation

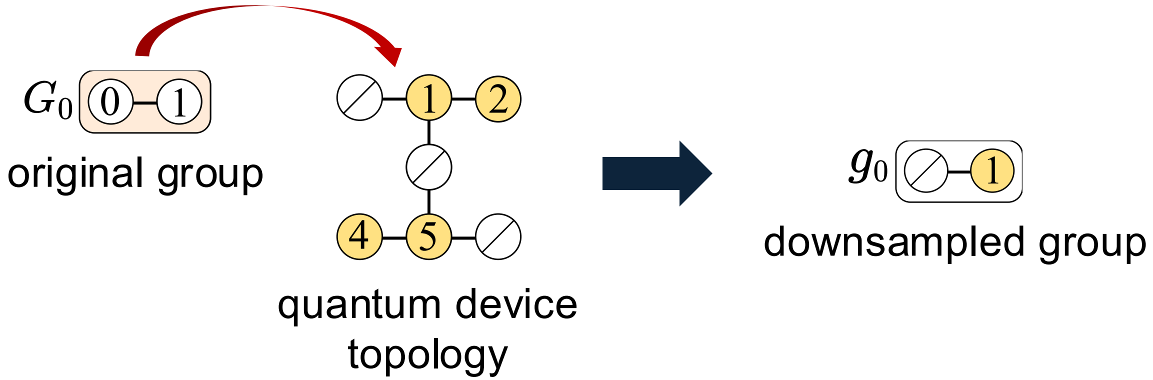
DyREM Dataflow Overview



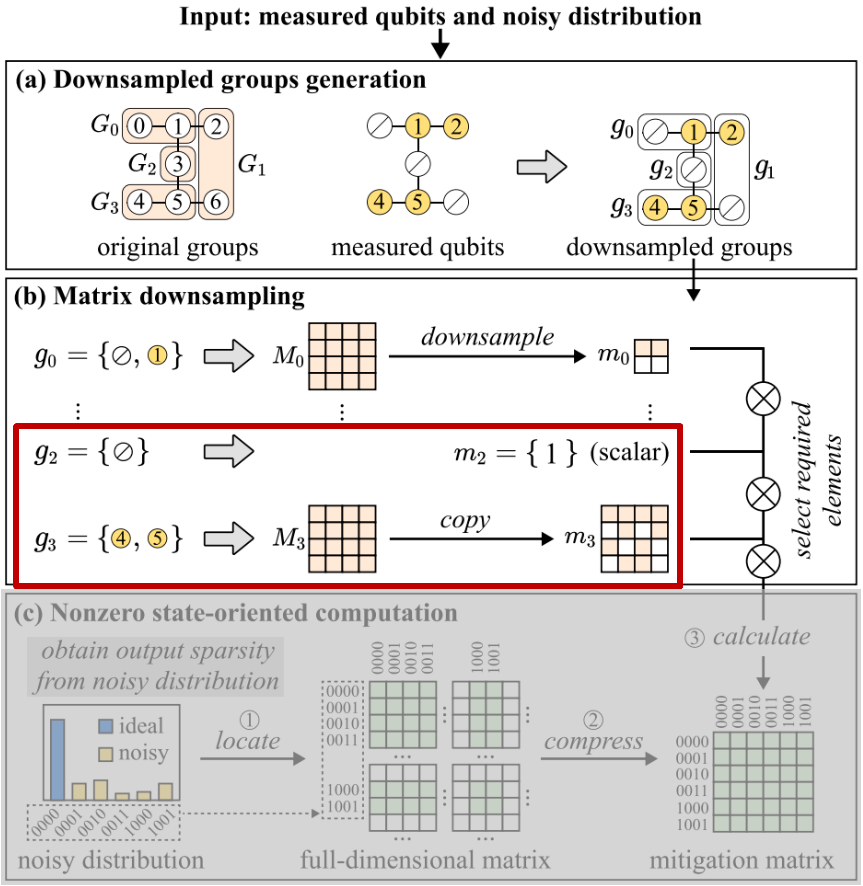
Downsampled Groups Generation



- We define the concept of ***downsampled group*** g_i, determined by the original groups G_i and measured qubits.
- Unmeasured physical qubits are denoted by $\varnothing$



Matrix Downsampling

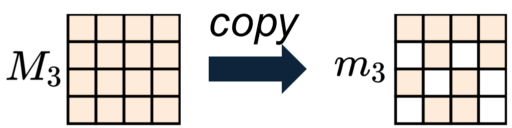


- We categorize downsampled groups into three types:
- Unmeasured (e.g., g_2)
- Fully measured (e.g., g_3)
- Partially measured (e.g., g_0 and g_1)

Unmeasured

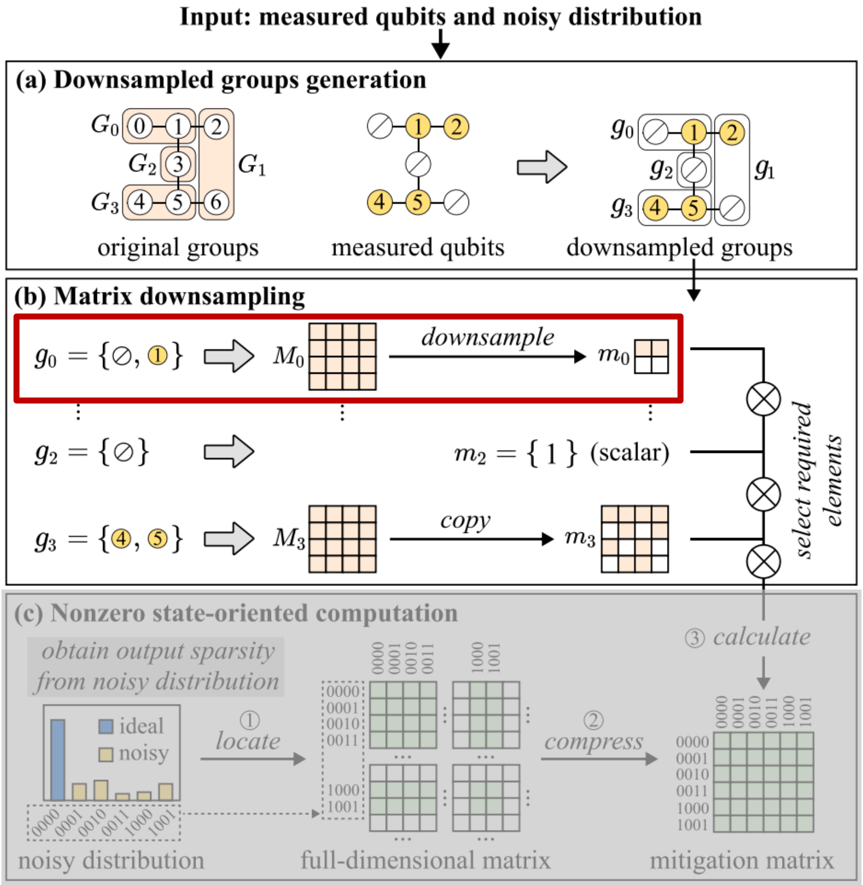
$$m_2 = \{1\}\ (scalar)$$

Fully Measured

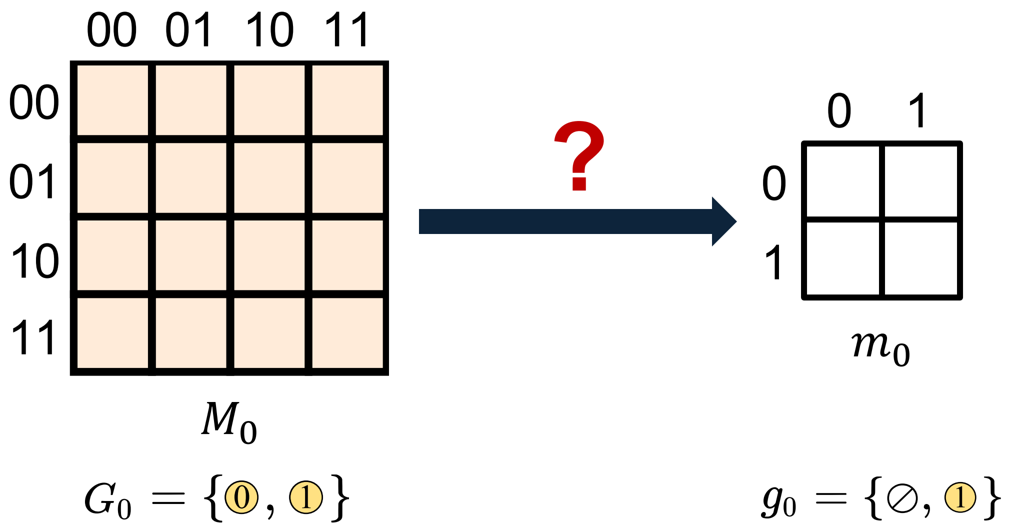


☐ does not involve the subsequent calculation of the mitigation matrix

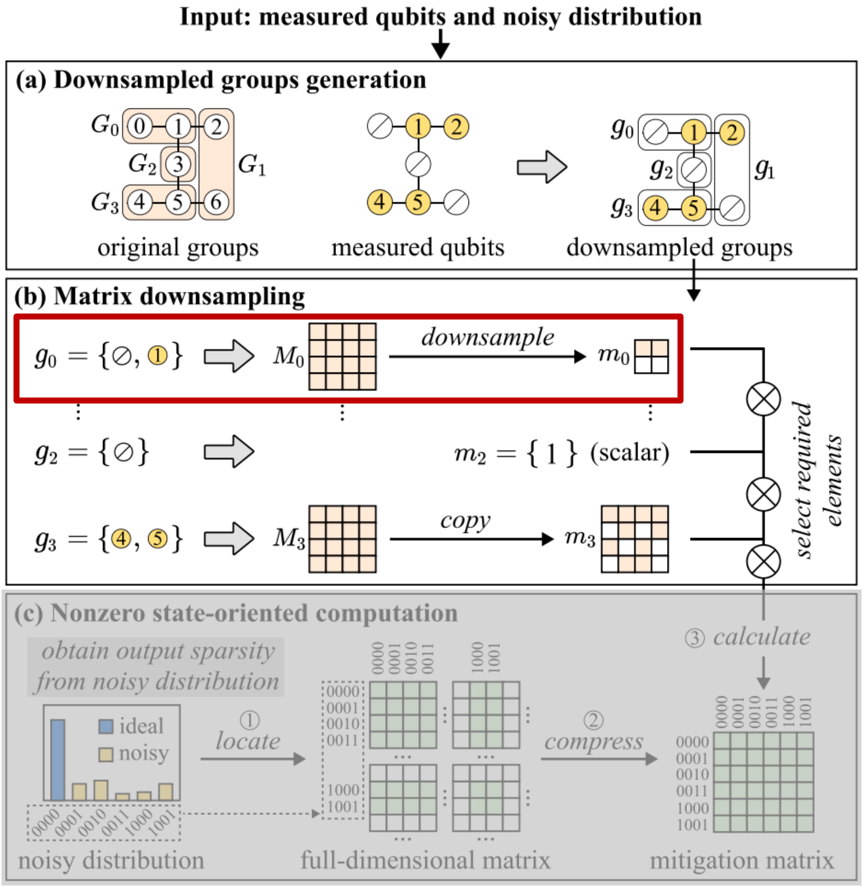
Matrix Downsampling



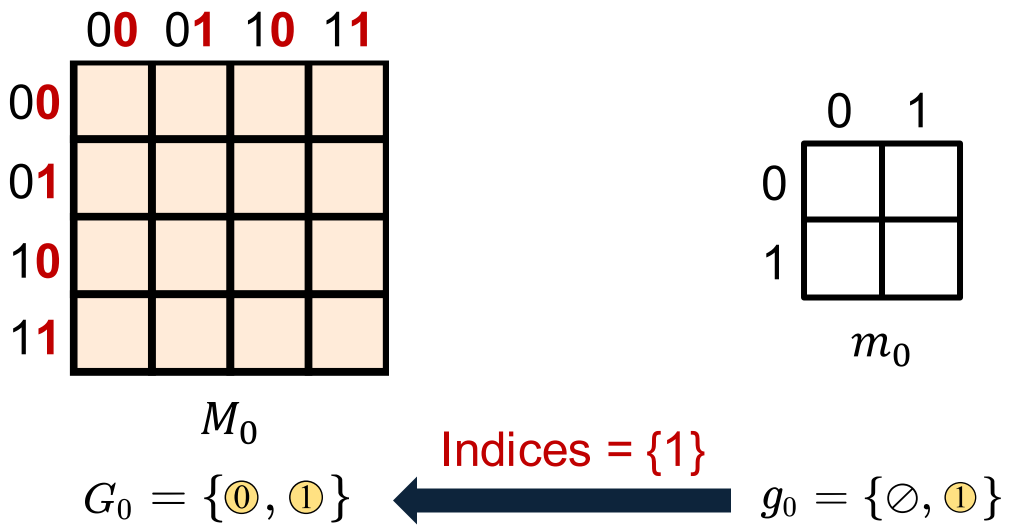
Partially Measured



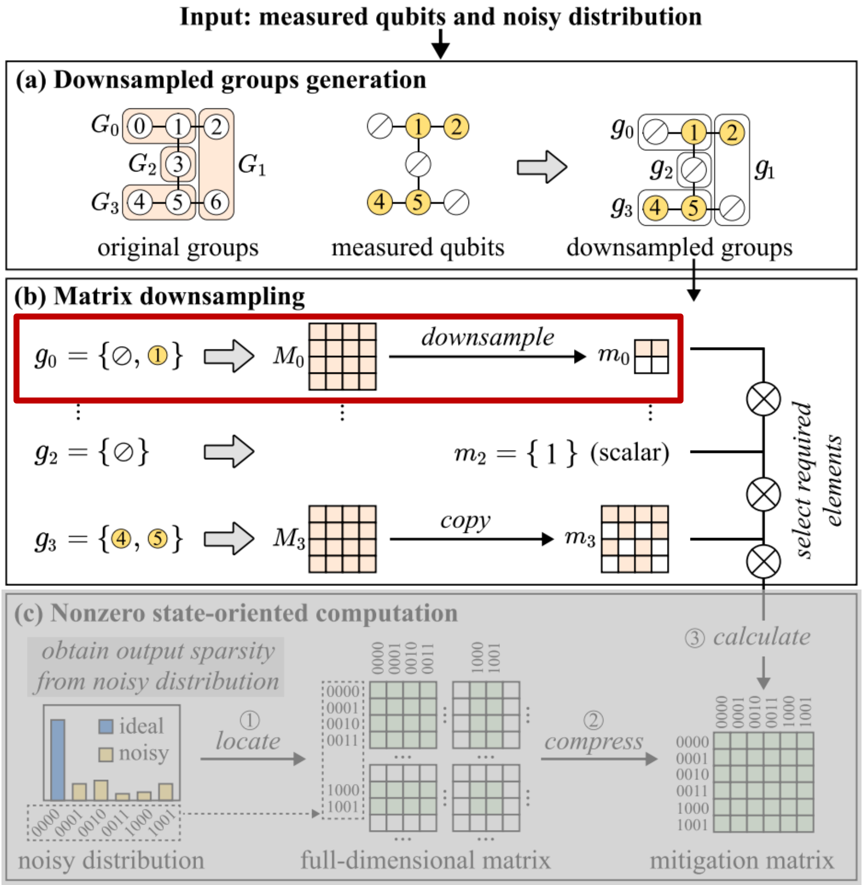
Matrix Downsampling



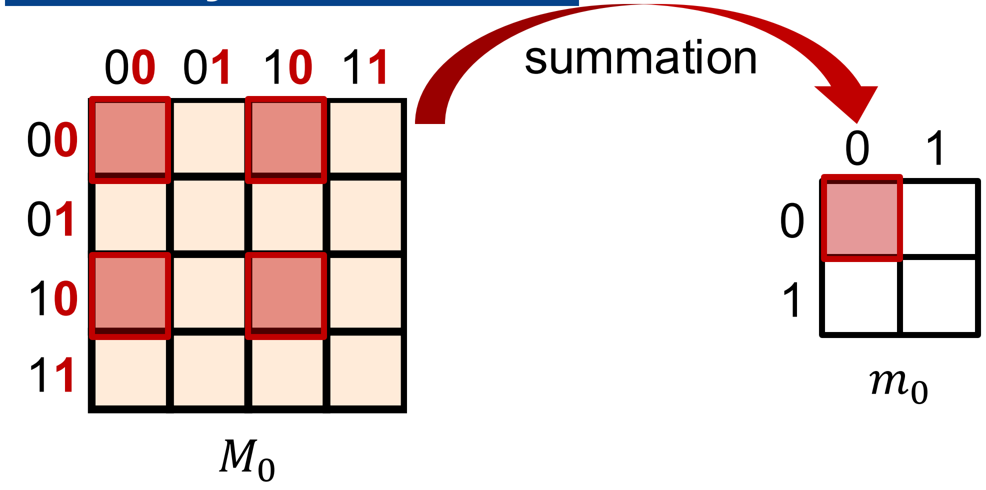
Partially Measured



Matrix Downsampling



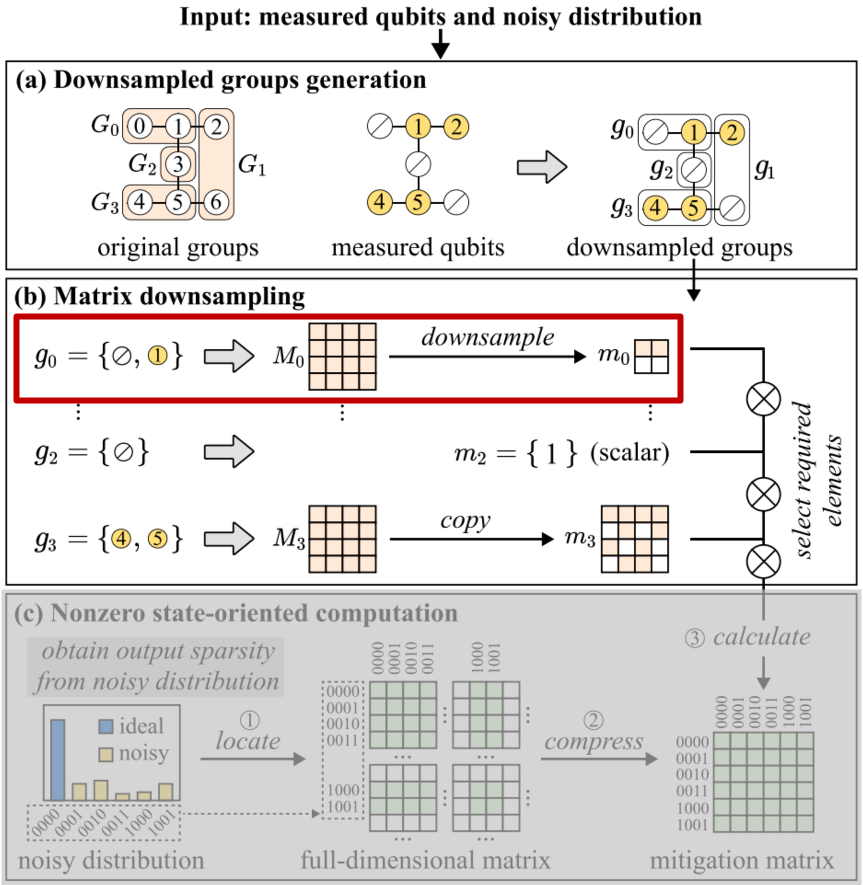
Partially Measured



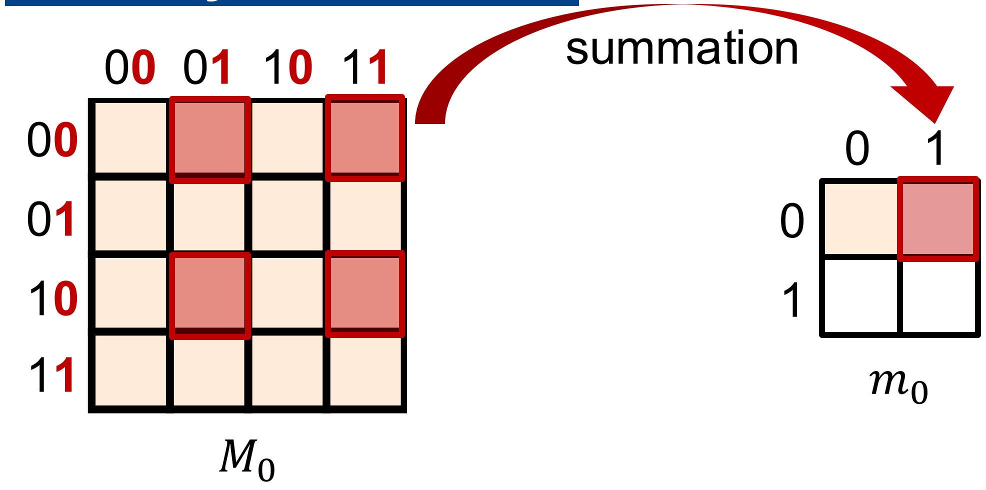
(00, 00)
(00, 10)
(10, 00)
(10, 10)

(0, 0)

Matrix Downsampling



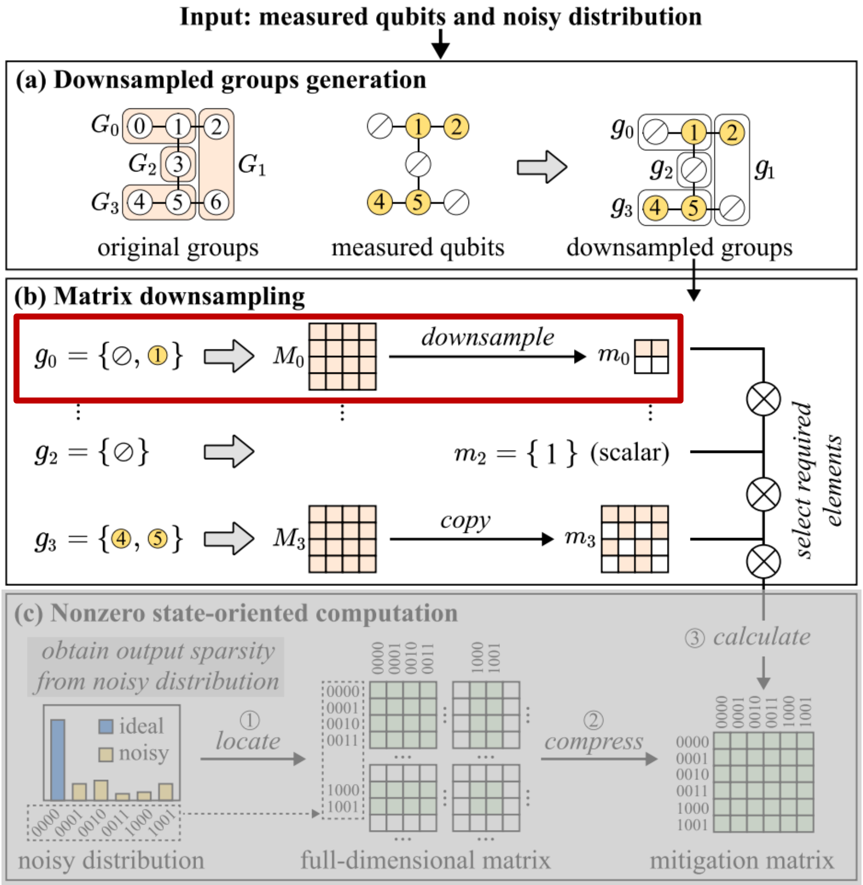
Partially Measured



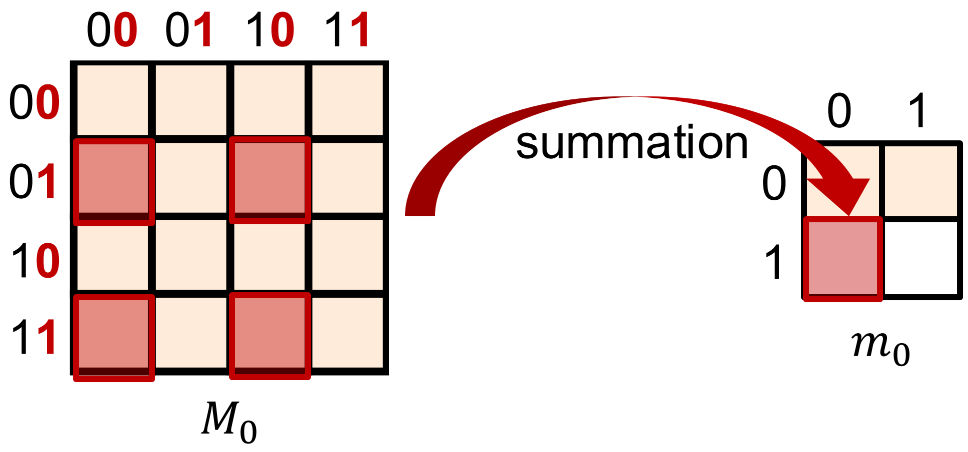
(00, 01)
(00, 11)
(10, 01)
(10, 11)

(0, 1)

Matrix Downsampling



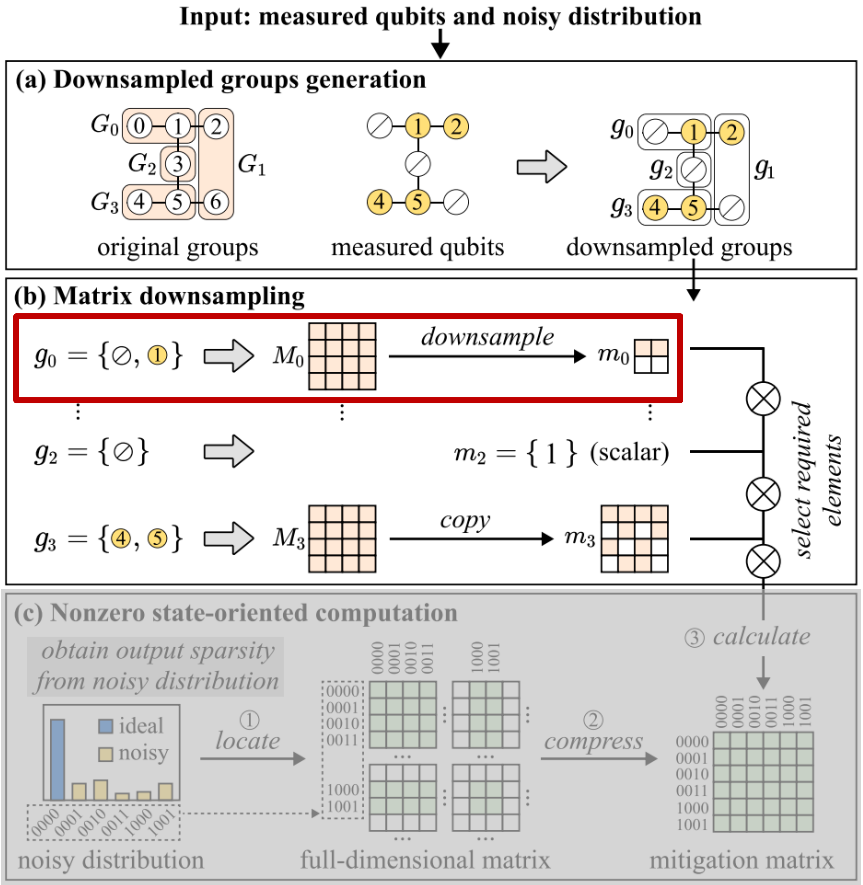
Partially Measured



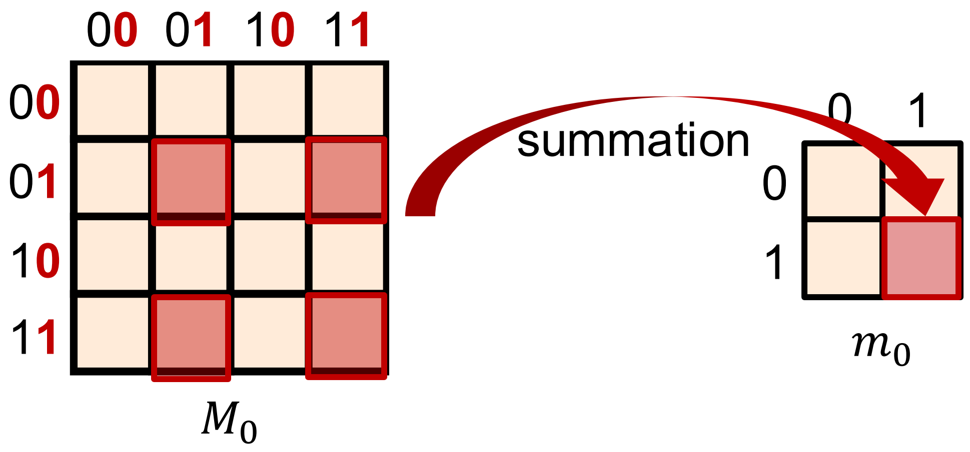
(01, 00)
(01, 10)
(11, 00)
(11, 10)

→ (1, 0)

Matrix Downsampling



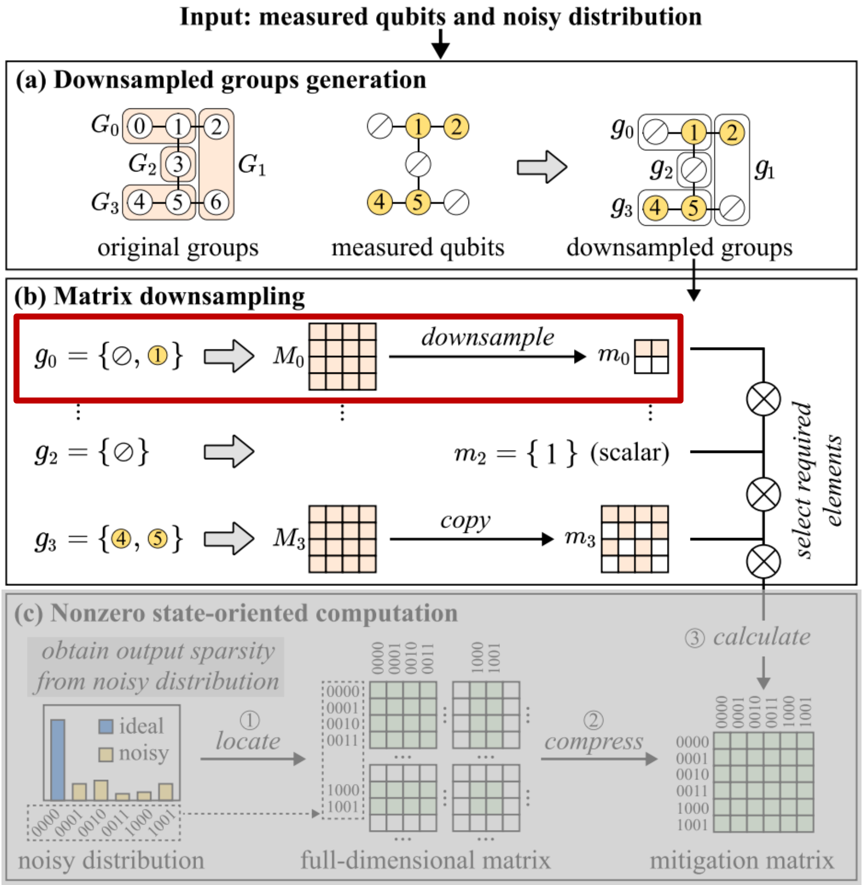
Partially Measured



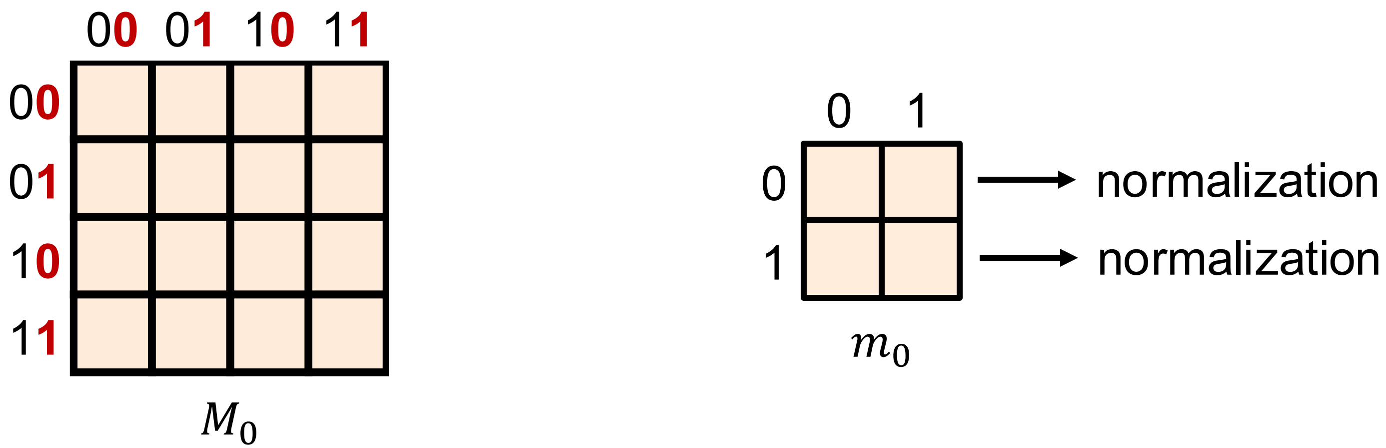
(01, 01)
(01, 11)
(11, 01)
(11, 11)

→ (1, 1)

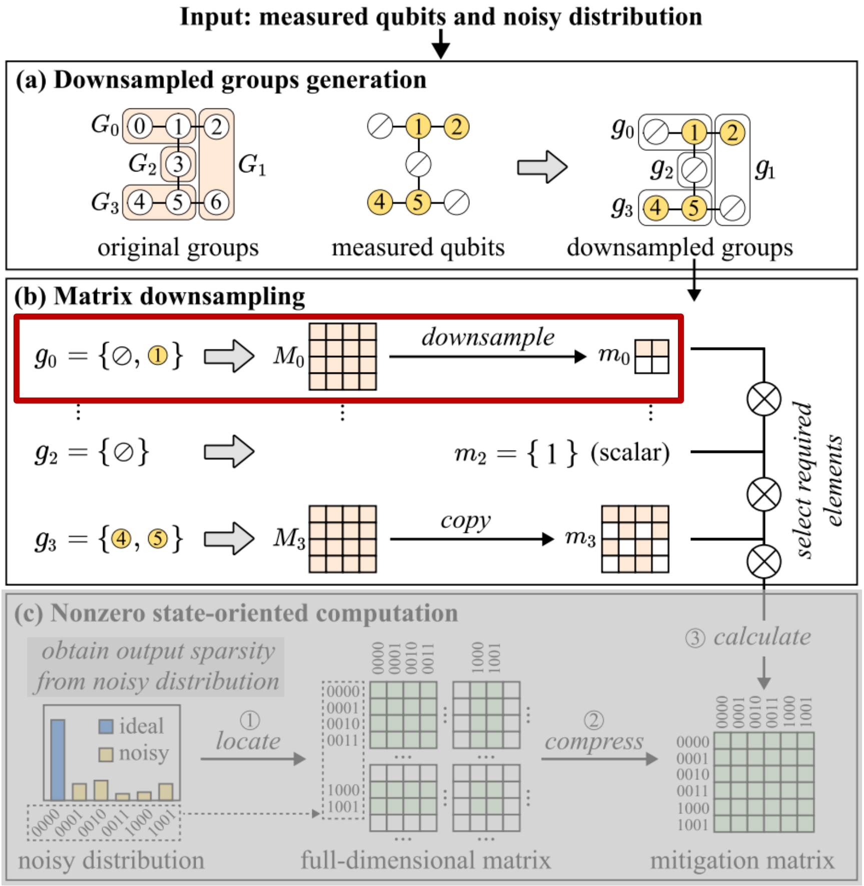
Matrix Downsampling



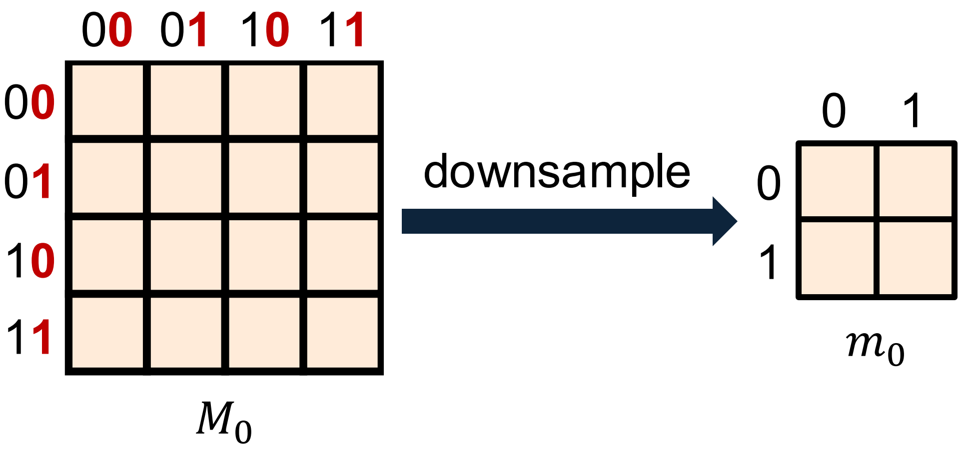
Partially Measured



Matrix Downsampling



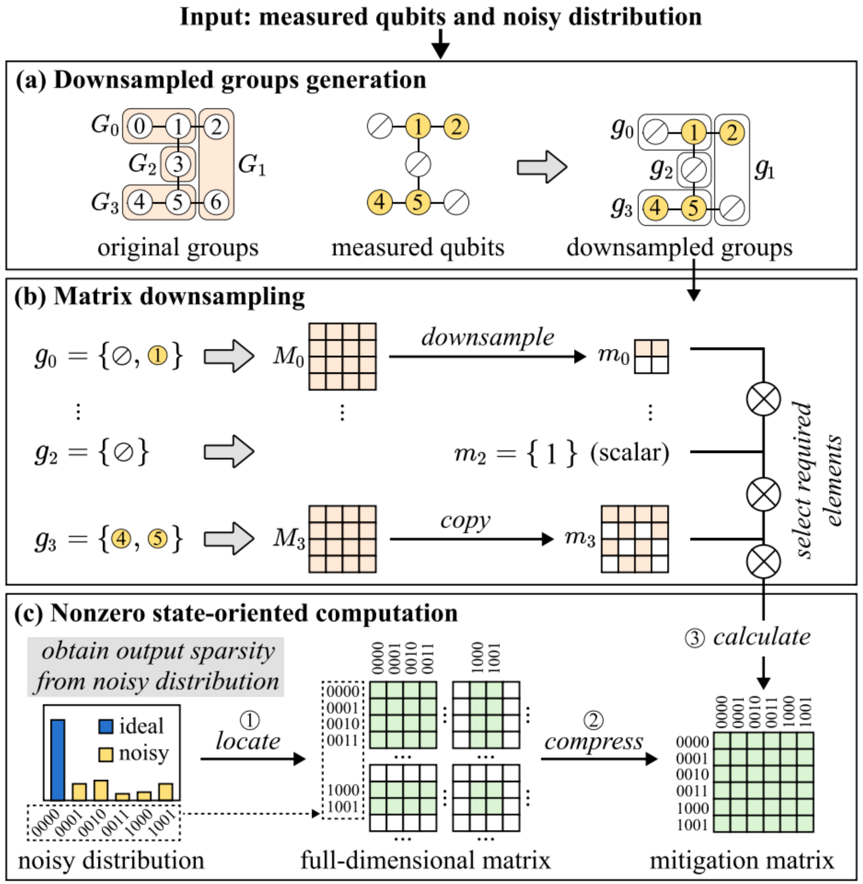
Partially Measured



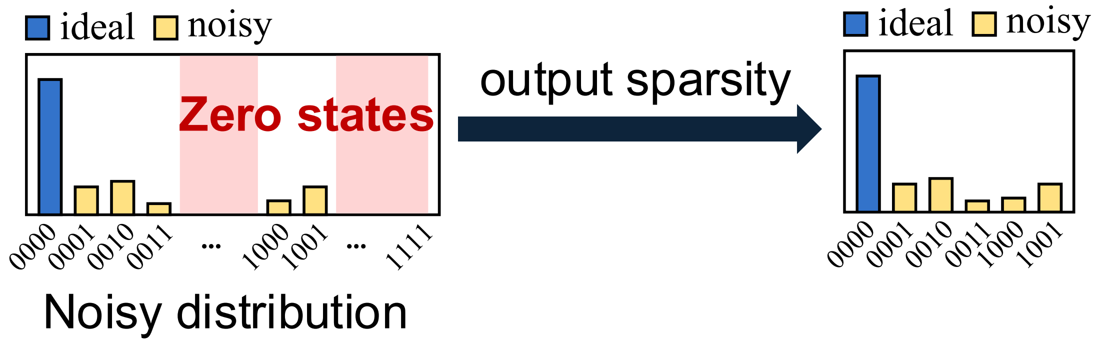
$G_0 = \{⓪, ①\}$ $\quad$ $g_0 = \{\varnothing, ①\}$

Downsampling is essentially a convolution process.
We can compute the kernel size and values based on g_i.

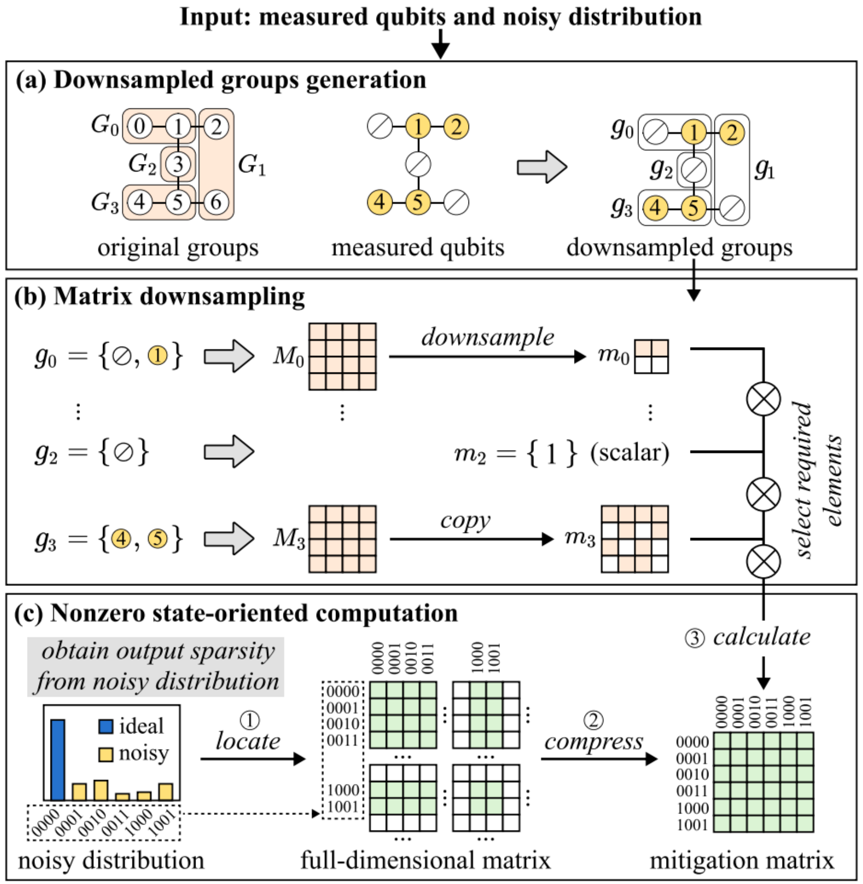
Nonzero State-Oriented Computation



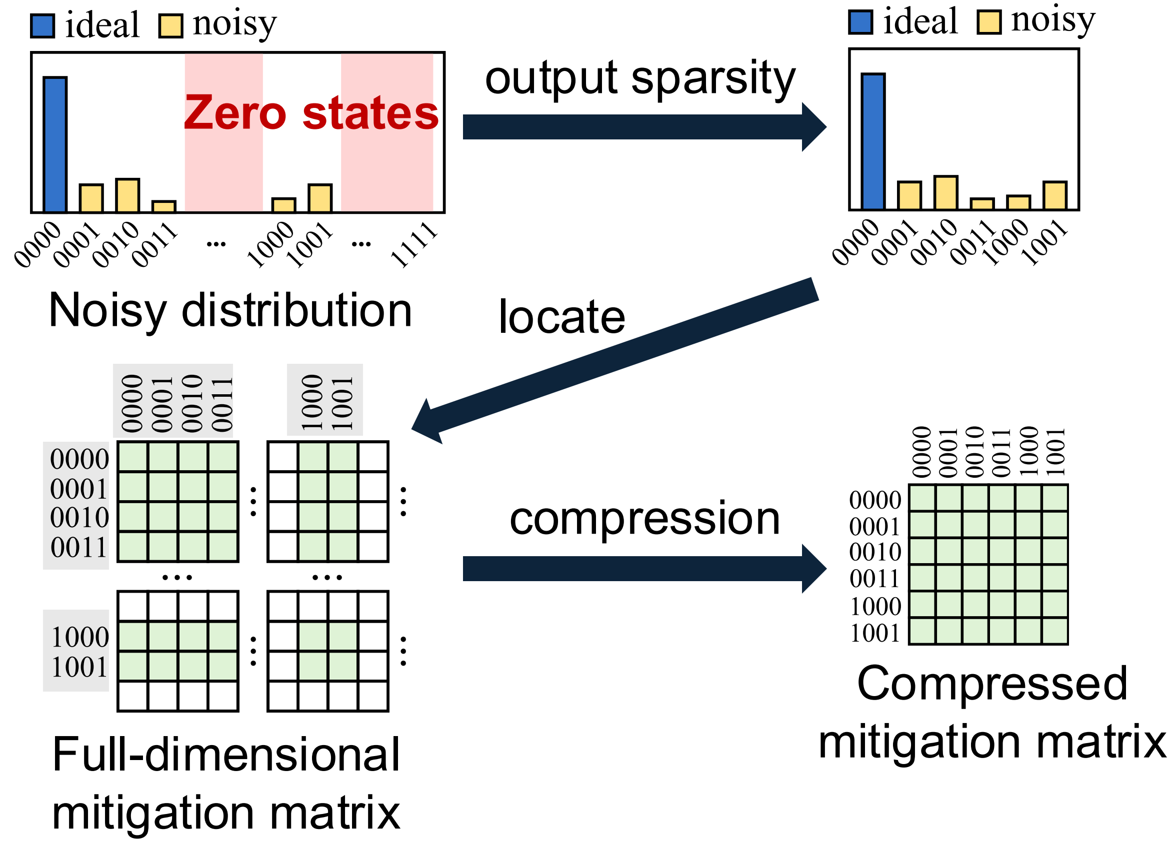
Mitigation Matrix Compression



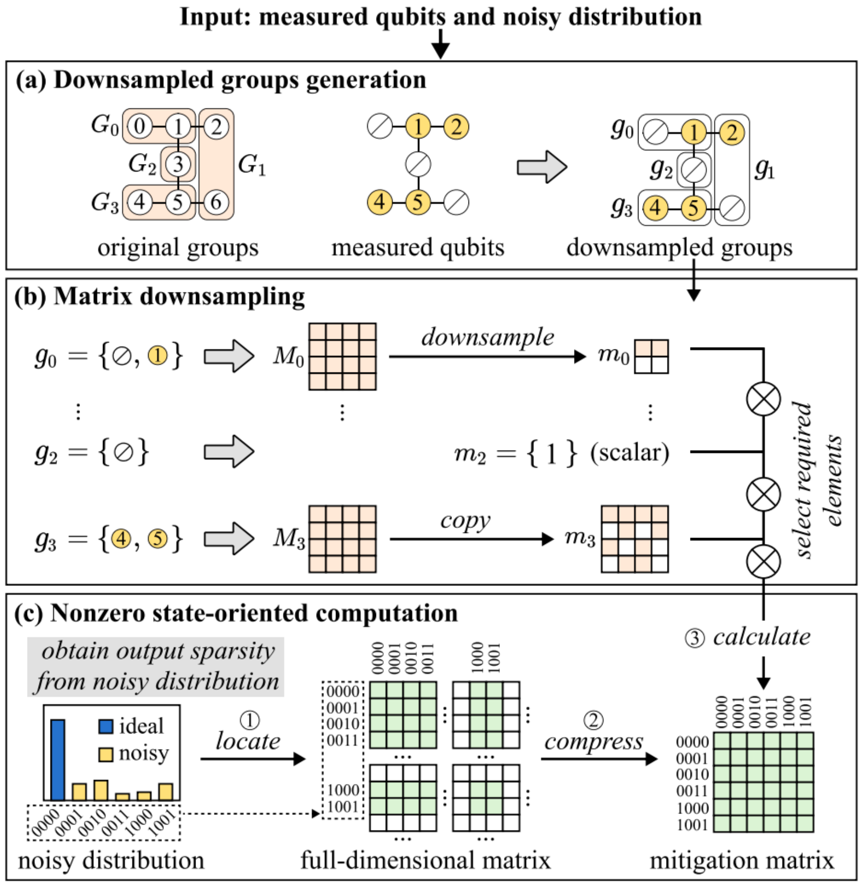
Nonzero State-Oriented Computation



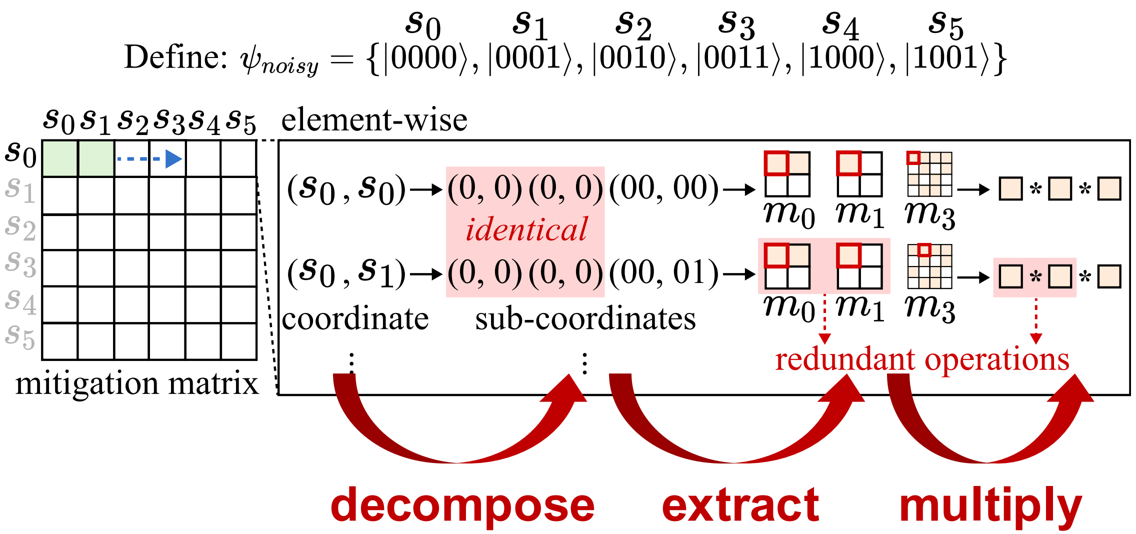
Mitigation Matrix Compression



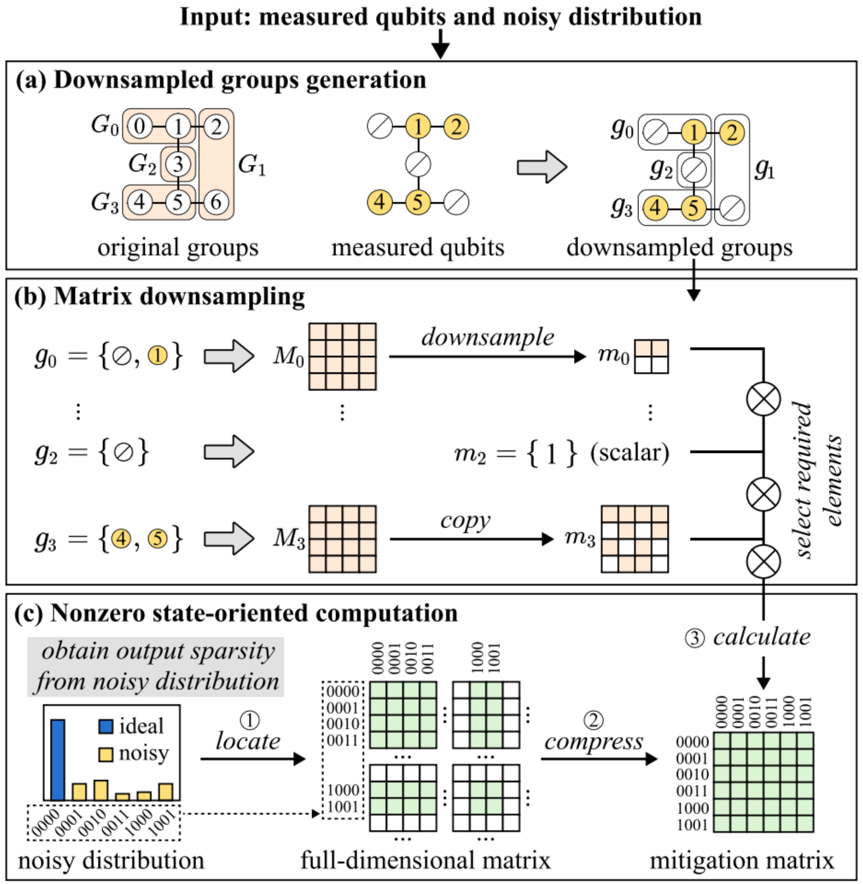
Nonzero State-Oriented Computation



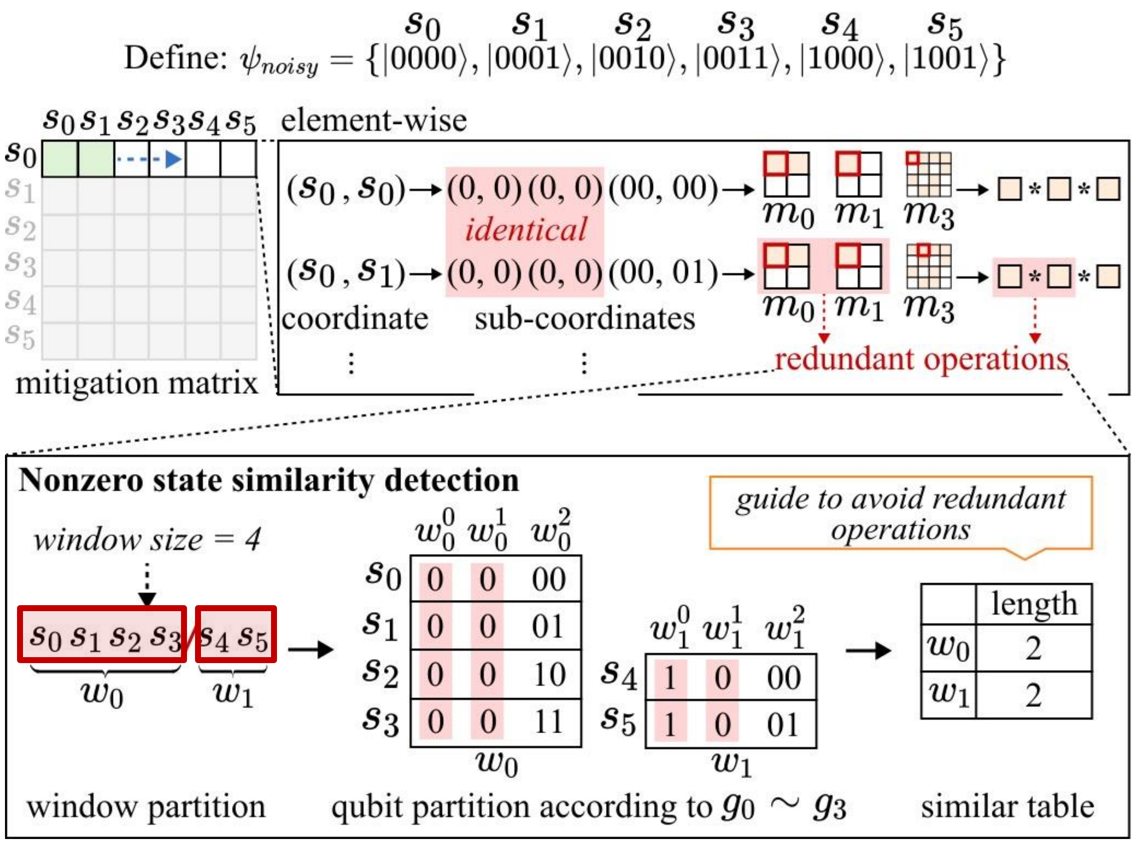
Nonzero State Similarity Detection



Nonzero State-Oriented Computation



Nonzero State Similarity Detection

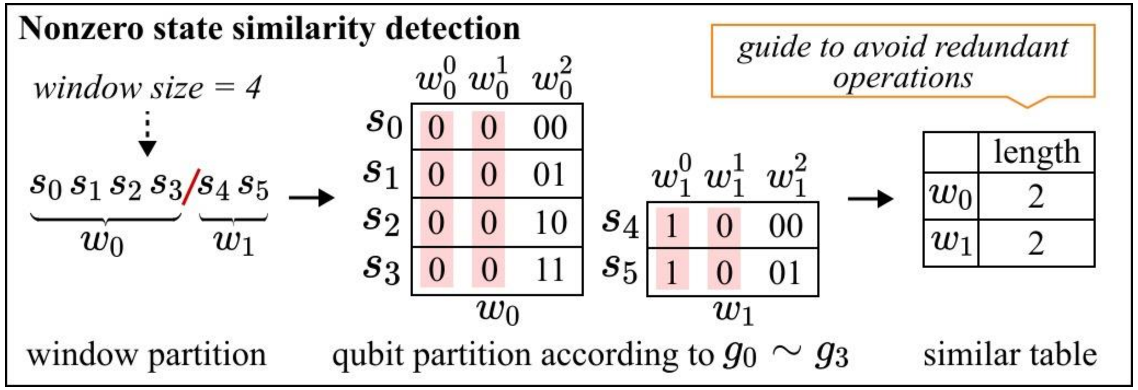


Outline of Presentation

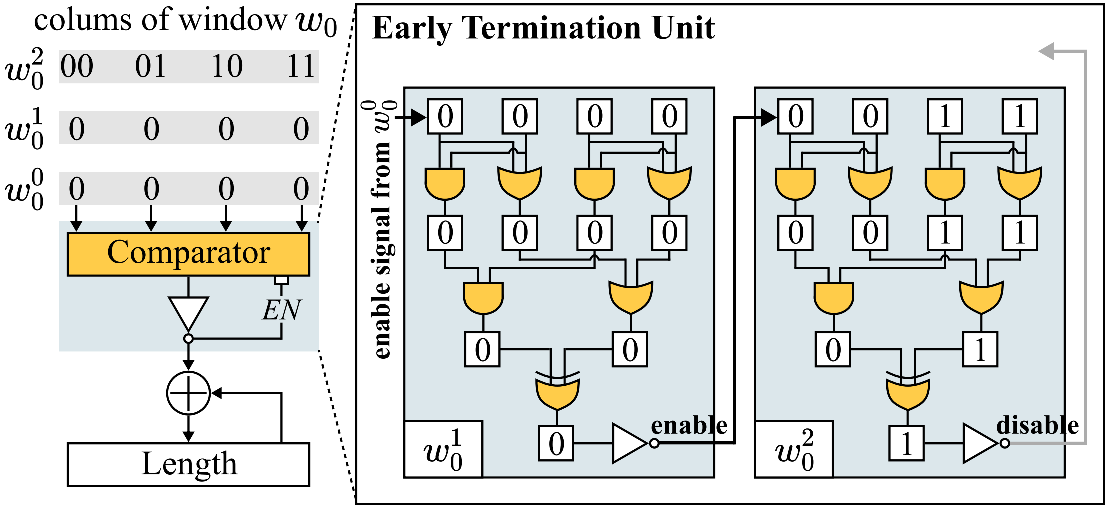
- Background
- Motivation
- DyREM Dataflow
- **Architecture Design**
- Evaluation

Nonzero State Detector

Software Side

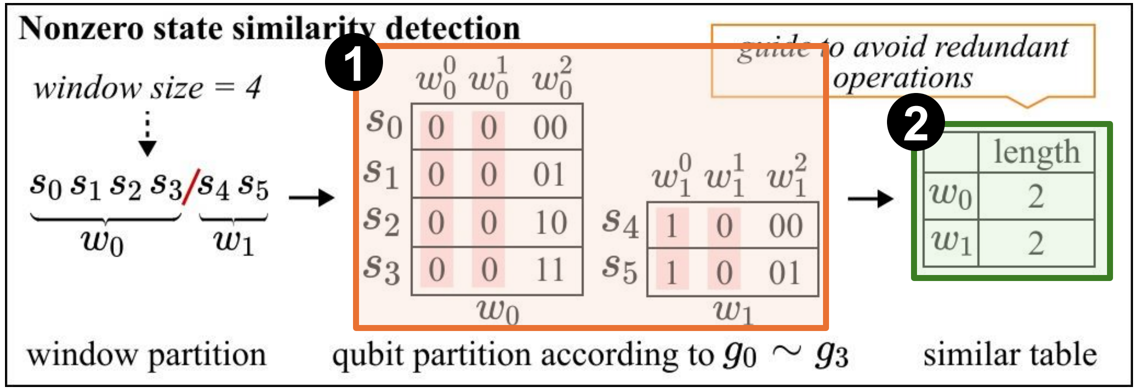


Hardware Side

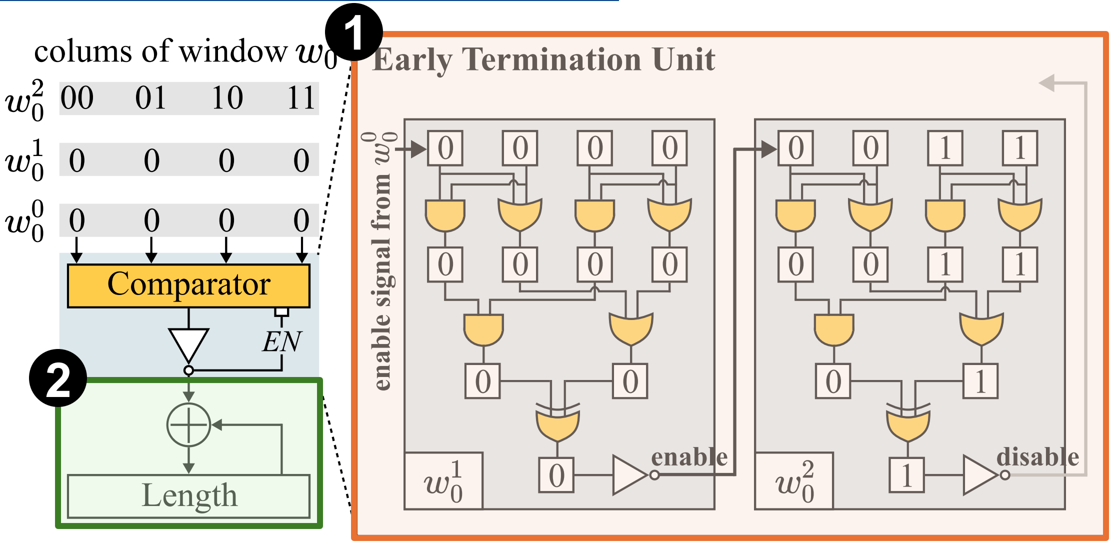


Nonzero State Detector

Software Side



Hardware Side

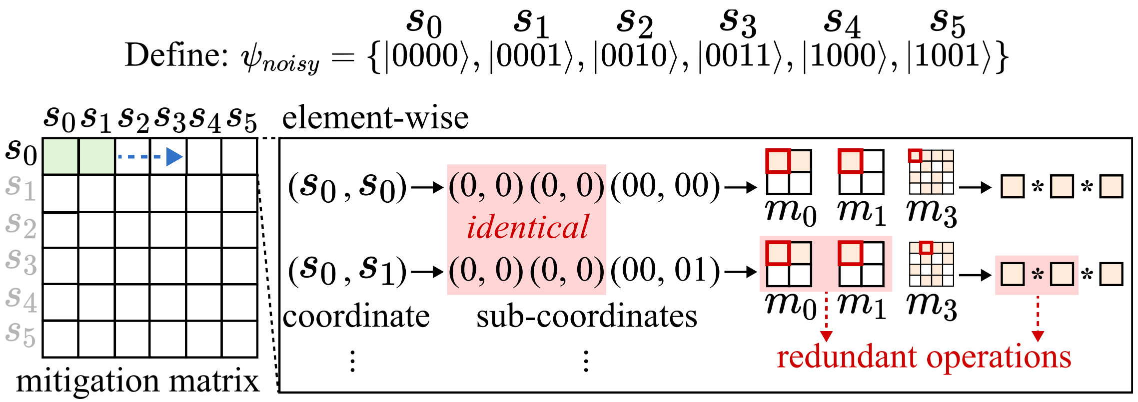


❶ Detection of identical columns in each window

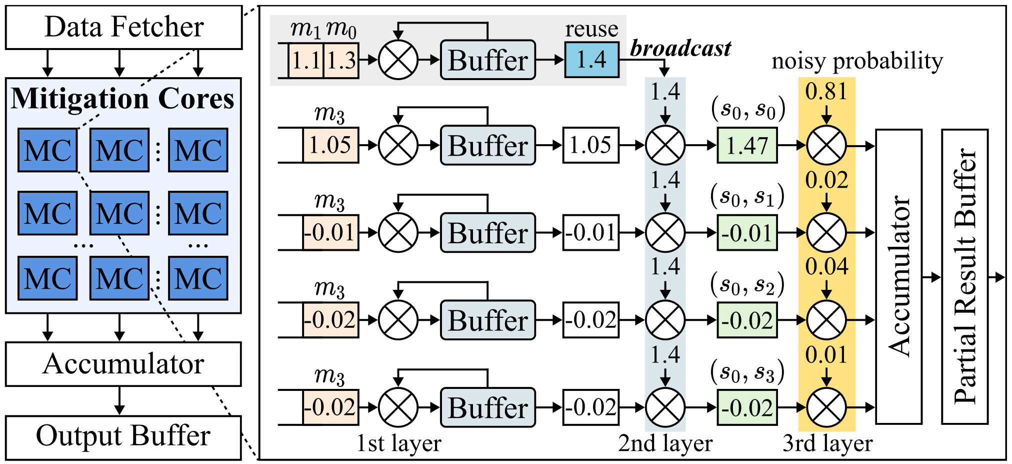
❷ Computation of the similar table

Mitigation Core

Software Side



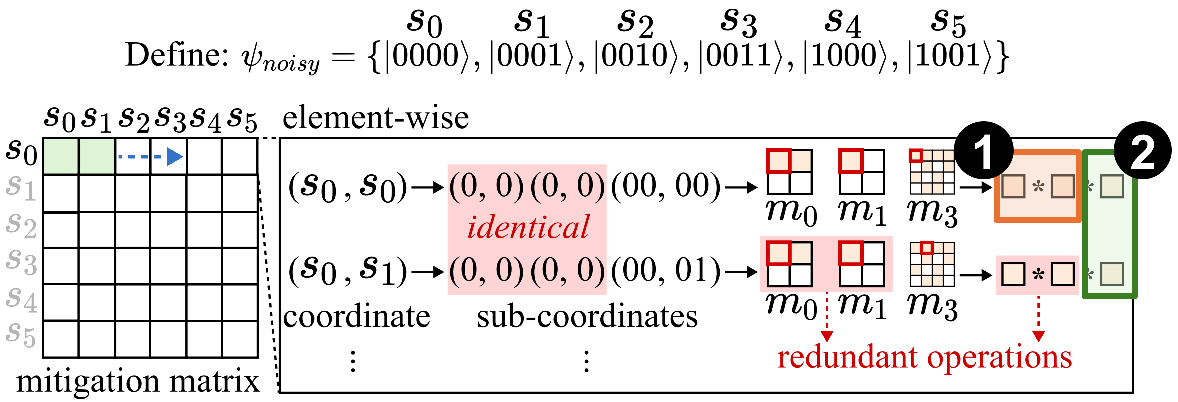
Hardware Side



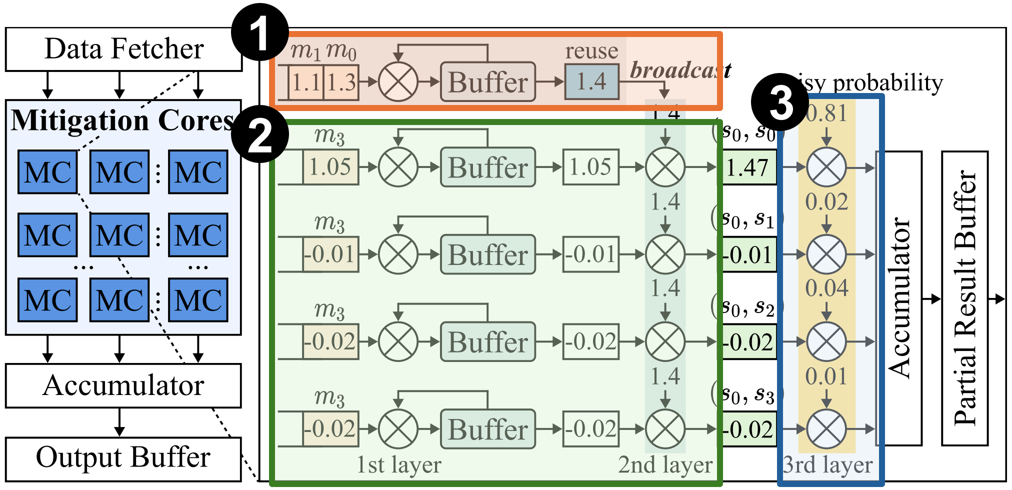
Details of Mitigation Core

Mitigation Core

Software Side



Hardware Side



Details of Mitigation Core

1. The computation of reuse data
2. Element-wise multiplication
3. Matrix-vector multiplication

Outline of Presentation

- Background
- Motivation
- DyREM Dataflow
- Architecture Design
- **Evaluation**

Evaluation Setup

- Benchmarks:
  - VQE, QAOA, FALCON, DJ algorithms
- Baselines:
  - IBM Mthree [1], Google IBU [2], QuFEM [3], SpREM [4]
- Experimental platforms:
  - NVIDIA A100 GPU (Mthree, IBU)
  - AMD EPYC 9554 64-core CPU (QuFEM)
  - Xilinx Alveo U50 FPGA (SpREM, DyREM)

[1] Nation, Paul D., et al. "Scalable mitigation of measurement errors on quantum computers." PRX Quantum 2.4 (2021): 040326.
[2] Pokharel, Bibek, et al. "Scalable measurement error mitigation via iterative bayesian unfolding." Physical Review Research 6.1 (2024): 013187.
[3] Tan, Siwei, et al. "QuFEM: Fast and Accurate Quantum Readout Calibration Using the Finite Element Method." ASPLOS. 2024.
[4] Zhang, Hanyu, et al. "SpREM: Exploiting Hamming Sparsity for Fast Quantum Readout Error Mitigation." DAC. 2024.

Hardware Performance

- Benchmarks:
  - VQE, QAOA, DJ algorithms (16, 20, 24, 28 qubits)
- Metrics:
  - Latency (s), Q-throughput (states/s)

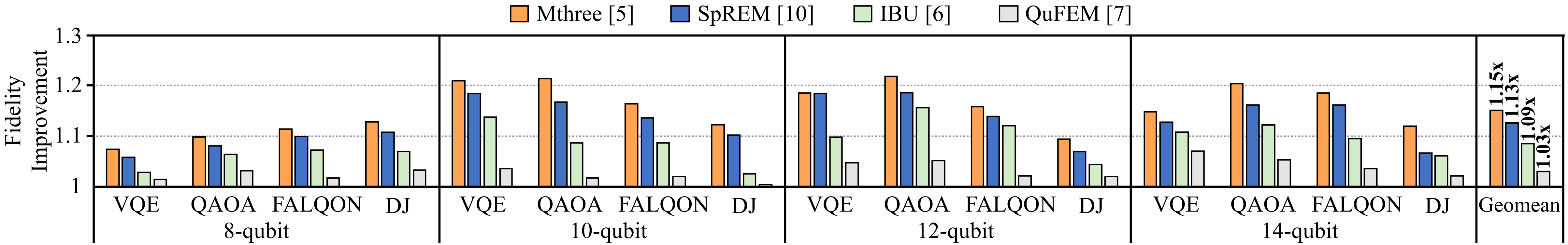
Baseline	Technical feature	VQE [12]		QAOA [13]		DJ [14]	
		Latency (s)	Q-throughput (states/s)	Latency (s)	Q-throughput (states/s)	Latency (s)	Q-throughput (states/s)
Mthree [5]	Hamming pruning	2.52 (384×)	7.27×10^3 (583×)	4.22 (461×)	4.63×10^3 (758×)	0.66 (206×)	2.77×10^4 (192×)
SpREM [10]	HDSR format	0.48 (73.8×)	1.76×10^6 (2.4×)	0.56 (61.5×)	1.49×10^6 (2.3×)	0.031 (9.6×)	3.43×10^6 (1.5×)
IBU [6]	Bayesian unfolding	13.0 (2000×)	3.58×10^5 (11.8×)	17.1 (1879×)	2.72×10^5 (12.9×)	4.59 (1437×)	1.01×10^6 (5.2×)
QuFEM [7]	Finite element analysis	11.7 (1800×)	1.56×10^3 (2726×)	13.5 (1483×)	1.45×10^3 (2420×)	5.42 (1687×)	2.65×10^3 (2002×)
DyREM	Redundancy detection	$\mathbf{6.52 \times 10^{-3}}$	$\mathbf{4.24 \times 10^6}$	$\mathbf{9.12 \times 10^{-3}}$	$\mathbf{3.51 \times 10^6}$	$\mathbf{3.25 \times 10^{-3}}$	$\mathbf{5.31 \times 10^6}$

(1) Average speedup: 9.6X ~ 2000X; (2) Q-throughput improvement: 1.5X ~ 2726X

- Our dataflow leverages the output sparsity and avoids redundant operations.
- We design a dedicated accelerator to support this dataflow.

Mitigation Fidelity

- Benchmarks:
  - VQE, QAOA, FALCON, DJ algorithms (8, 10, 12, 14 qubits)
- Metric:
  - Fidelity

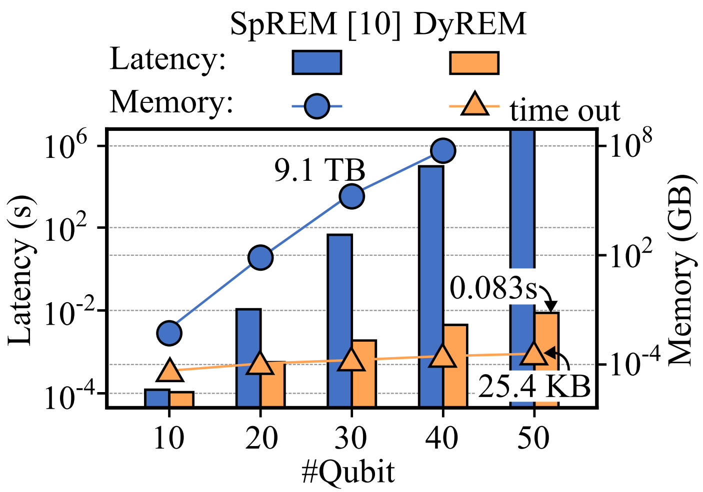


Average fidelity improvement: 1.15X, 1.13X, 1.09X, and 1.03X

- We eliminate the quantum states that do not contribute to fidelity.
- We use the grouping matrix to consider the crosstalk.

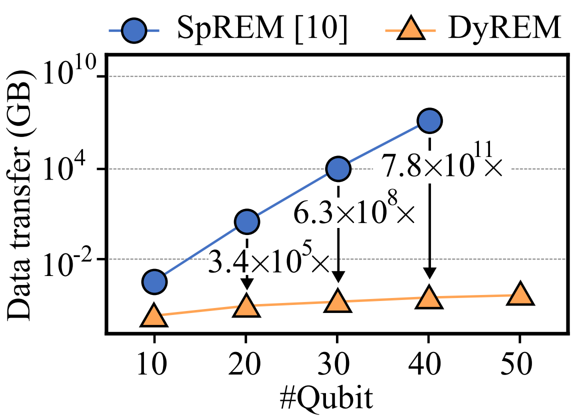
Comparison with SpREM

Latency and Memory



(a) Latency and memory usage of mitigating the DJ algorithm.

Data Transfer



(b) Data transfer volume of mitigating the DJ algorithm.

- Our accelerator calculates the mitigation matrix on-chip, avoiding the limitation of finite bandwidth.

Thanks for listening

Q & A

AI

Security

Systems

EDA

Design

THE CHIPS TO SYSTEMS CONFERENCE

62

SPONSORED BY CEDA sigda