Quantum Error Mitigation

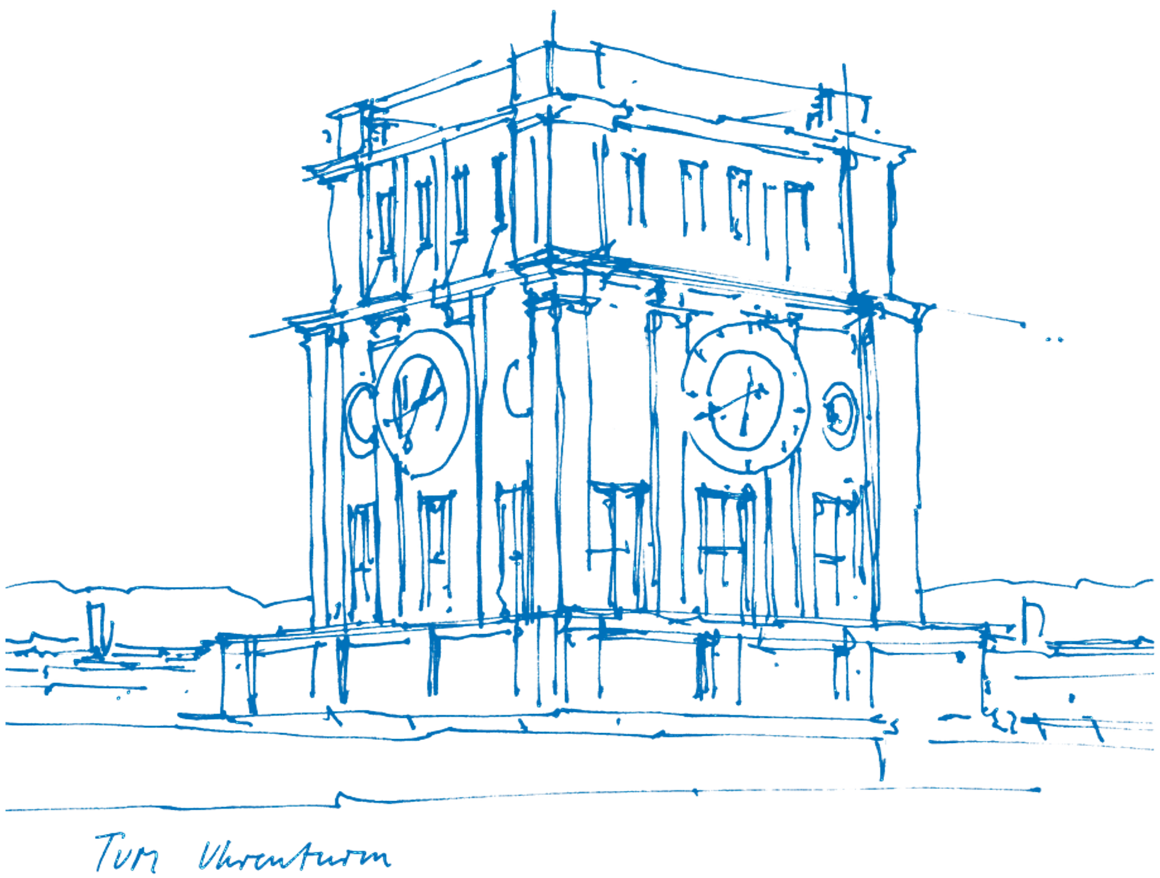
Tianyi Wang

Technical University of Munich

School of Computation, Information and Technology

Department of Informatics 5

09.12.2022



Outline

- Introduction to Quantum Error Mitigation
  - Definition
  - Difference with Quantum Error Correction
  - Why do we use it?

- Various QEM techniques in chronological order
  - Qiskit Error Mitigation
  - General Error Mitigation
  - M3 and Others

- Conclusion

Introduction

- Fact: (present) Quantum computers have errors
  - ⚠: new types of errors! (bit **and phase**)
  - Therefore: need error handling

Interviewer: It says here you're extremely fast at factoring, what are the factors of 9025?

Quantum computer: 7 and 11.

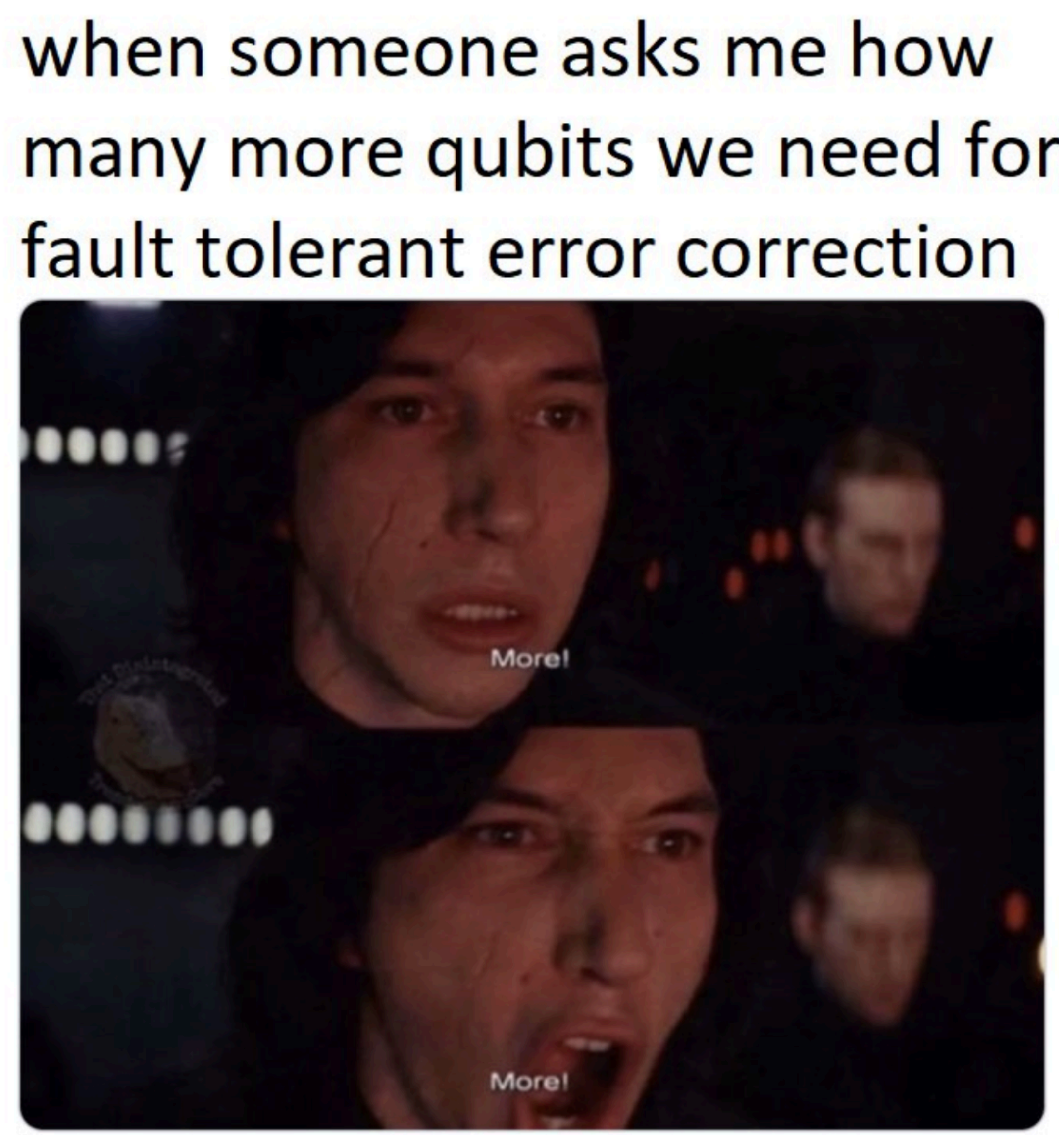
Interviewer: that's not even close

Quantum computer: yeah, but it was fast.

Source: https://twitter.com/quantummemeing/status/1111309373693935616?lang=bn

Introduction

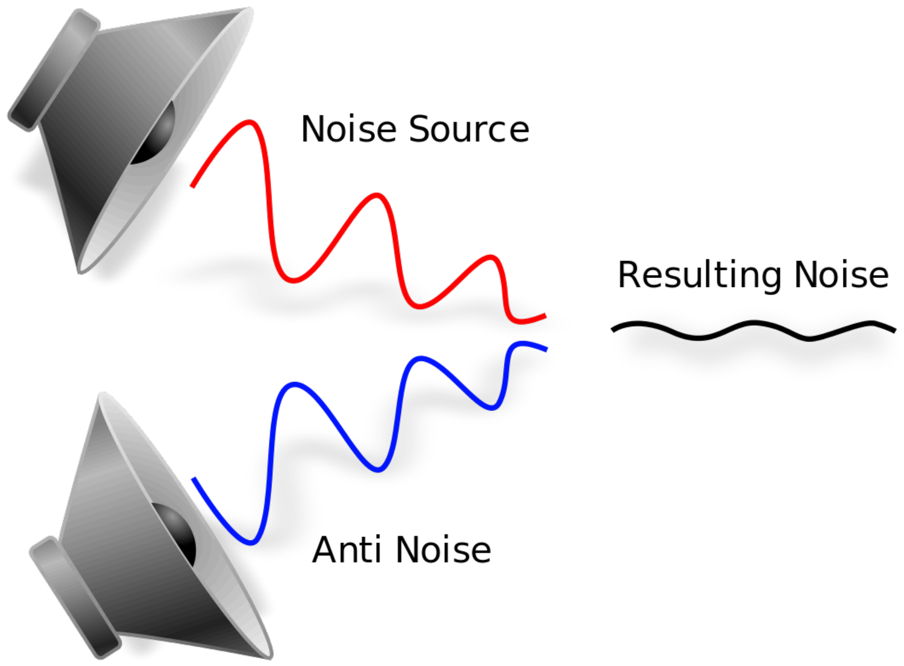
- Quantum Error Correction
  - Requires additional hardware
  - ❗ : not enough (physical) qubits!
  - Therefore: error mitigation!



Source: https://twitter.com/quantummemeing/status/1260955451325325313

Introduction

- Quantum Error Mitigation
  - Using the outputs of (calibration) circuits to reduce/ eliminate the error effects
  - **Does not** require additional hardware
  - Feasible on the near term
  - Common error types:
    A. State preparation and measurement errors (SPAM)
    B. Gate errors



Source: https://upload.wikimedia.org/wikipedia/commons/7/7d/Active_Noise_Reduction.svg

Qiskit Error Mitigation

- Premise
  - Assume N qubits, 2^N possible states
  - Our quantum device produces result counts ($v_1, v_2, \ldots, v_{2^N}$), which differ from the ideal (exact) data ($e_1, e_2, \ldots, e_{2^N}$).
  - Construct (normalized) column vectors

$$V = \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_{2^N} \end{pmatrix}, \quad E = \begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ e_{2^N} \end{pmatrix}$$

  - Postulate the existence of a $2^N \times 2^N$ matrix M s.t.

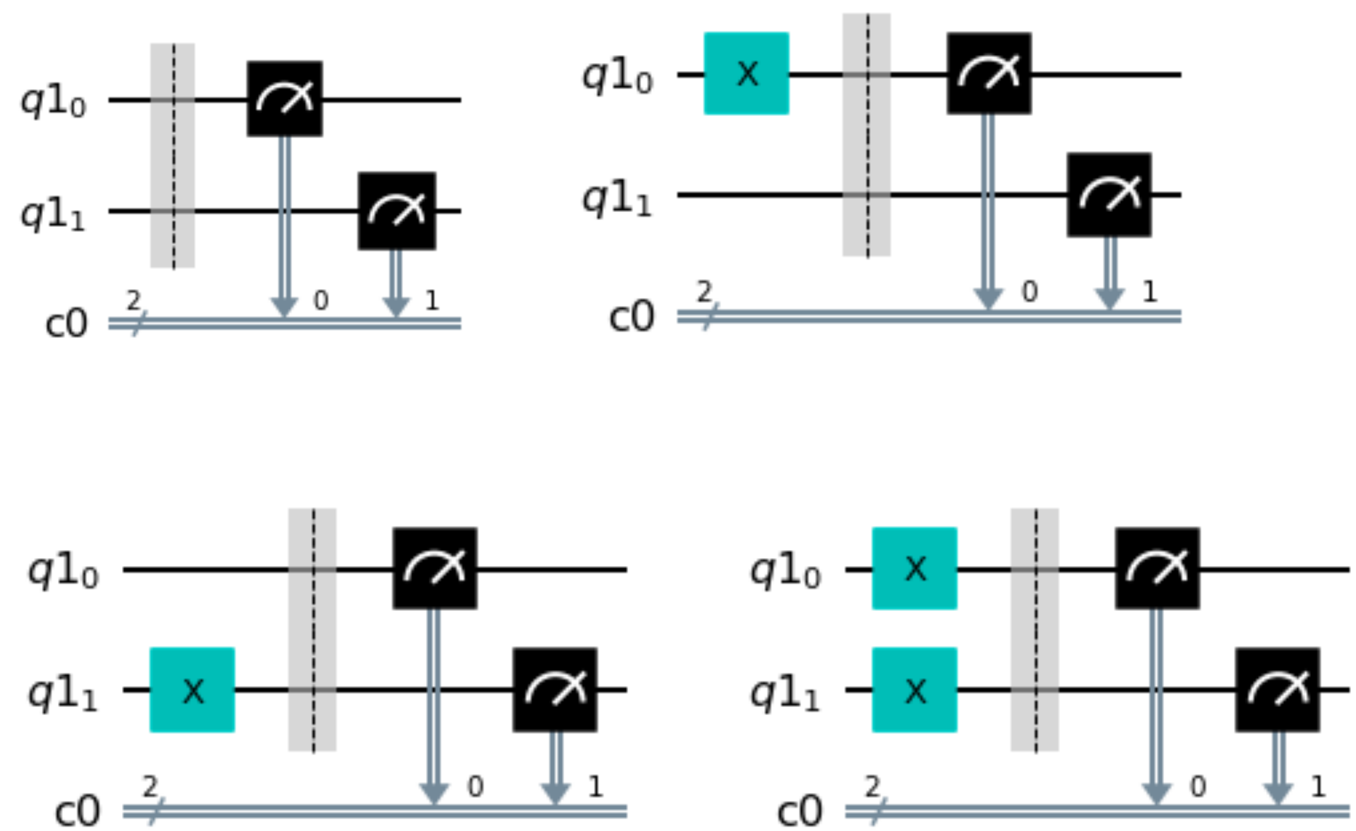
$$\mathbf{ME = V,}$$

  - Remarks:
    1. If device not error prone: M = I
    2. If device error prone, M has non-zero off-diagonal elements.

Qiskit Error Mitigation

- Calibration and mitigation
  - Prepare qubits in all possible 2^N states and measure each state.

- Example (for 2 qubits)
  - Possible states: $|00\rangle, |01\rangle, |10\rangle, |11\rangle$
  - Calibration circuits:



Qiskit Error Mitigation

- Calibration and mitigation
  - Enter the data from each calibration circuit into columns of M, where the j-th column, starting from left, takes data from the circuit whose state is given by the binary representation of j, for all j = 1,..., 2^N. M is the *Qiskit calibration matrix*
  - Note that

$$ME = V,$$

  - Therefore, noting $X = (x_1, x_2, \ldots, x_{2^N})$ as mitigated data,

$$MX = V,$$

  - Therefore: using M^{-1},

$$\mathbf{X = M^{-1}V}$$

- Example (for 2 qubits)
  - M with shots=10000 on ibmq_quito

$$\begin{bmatrix} 0.9583 & 0.0661 & 0.0516 & 0.0025 \\ 0.0166 & 0.9139 & 0.0017 & 0.0517 \\ 0.0245 & 0.0012 & 0.9277 & 0.0674 \\ 0.0006 & 0.0188 & 0.019 & 0.8784 \end{bmatrix}$$

Intuition: "'00' becomes '01' 166 times"

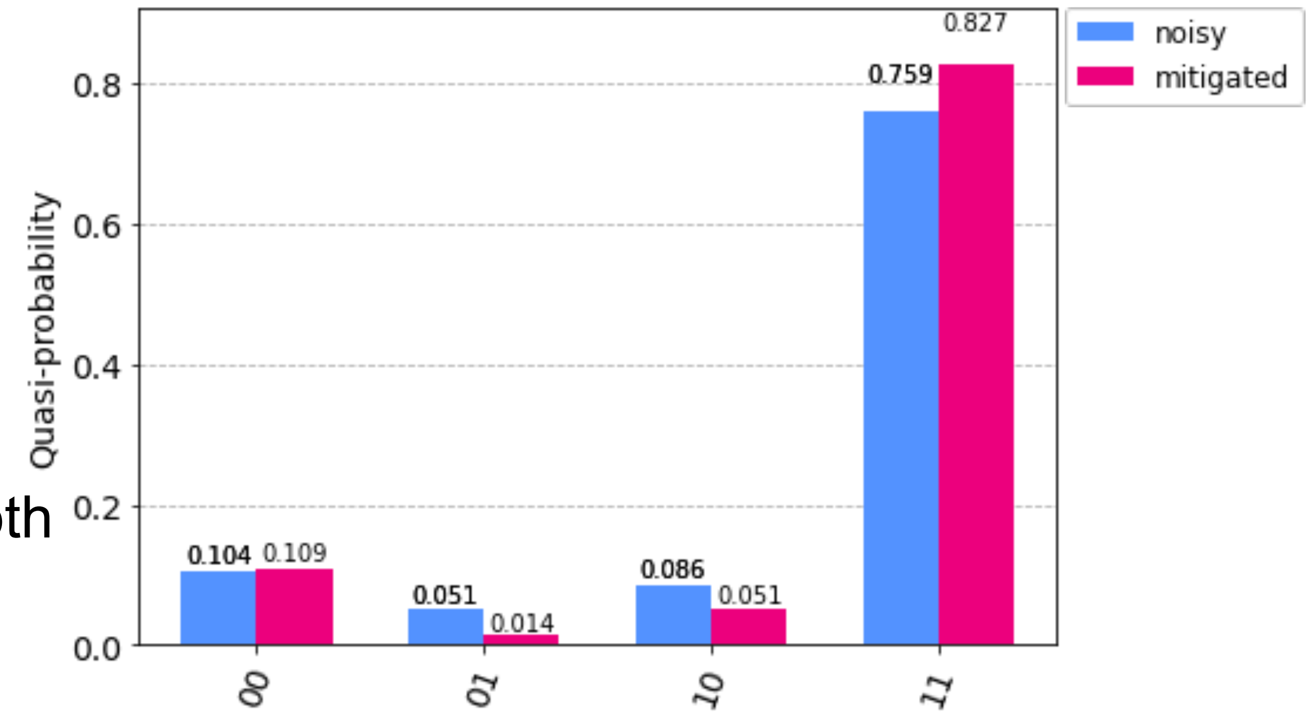
Side note: on aer_simulator:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Qiskit Error Mitigation

- Good:
  - 🙂 Hands-on (directly from Qiskit)
  - 🙂 Easy to understand
  - 🙂 The method does not depend on circuit depth
- But…
  - 🙁 Only addresses SPAM errors
  - 🙁 Does not work well for circuits with greater depths

    when SPAM no longer major source of error
- Therefore: General Error Mitigation!

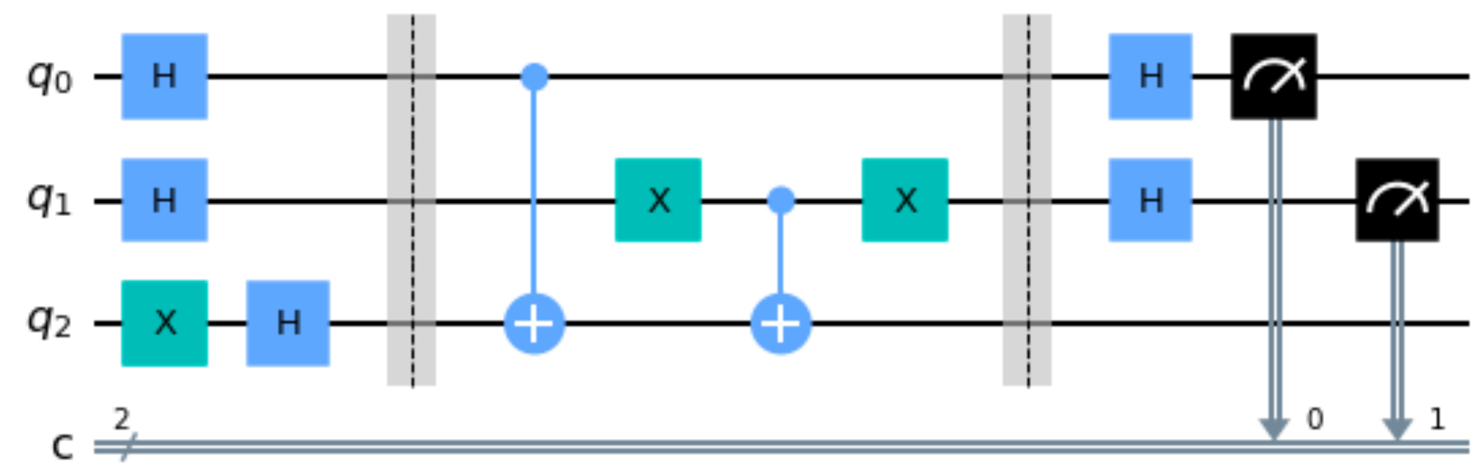


Deutsch-Jozsa 2-qubit balanced run on ibmq_manila

General Error Mitigation

- Calibration and mitigation (simplified)
  - Assume circuit $\mathbf{C_g}$ of depth D and with N qubits.
  - Prepare possible states **twice**.
  - Break C_g into two halves up to depth $\lfloor D/2 \rfloor$, add to calibration circuits.
  - Add inverse gates of the gates added.
  - Measure the calibration circuits and record the data in calibration matrix M_1
  - Analogously, for the remaining half of the gates on the remaining calibration circuits, and name the new matrix M_2.
  - Calculate the matrix $M_G = (M_1 + M_2)/2$. M_G serves the same purpose as M in QiEM.

- Example (for Deutsch-Jozsa circuit for 2 qubits, balanced)

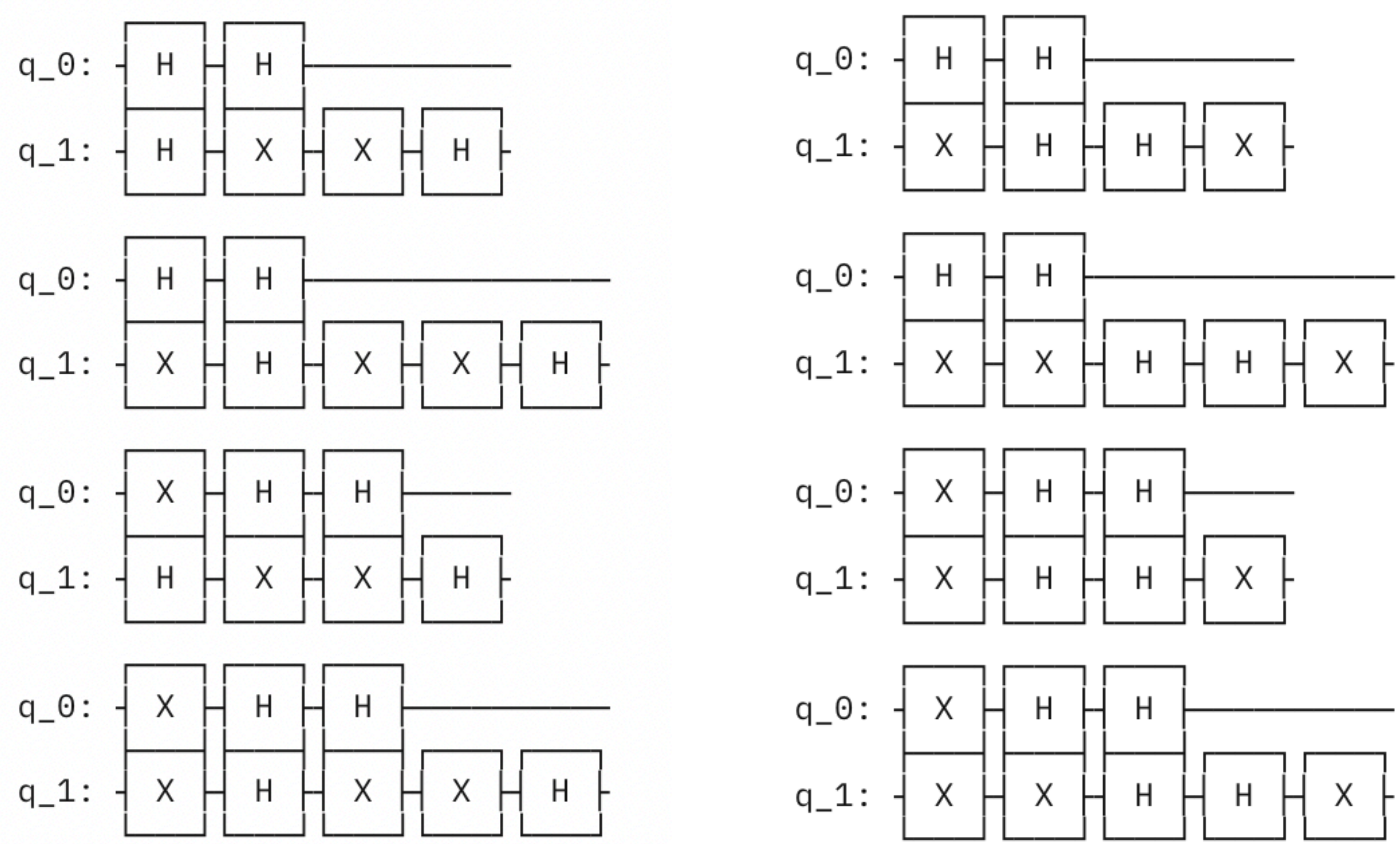


  - Break down to two halves =>
  - First half has h(0), h(1), x(1)
  - Second half has x(1), h(1), h(0)

General Error Mitigation

- Calibration and mitigation (simplified)
  - Assume circuit C_g of depth D and with N qubits.
  - Prepare possible states **twice**.
  - Break C_g into two halves up to depth $\lfloor D/2 \rfloor$, add to calibration circuits.
  - Add inverse gates of the gates added.
  - Measure the calibration circuits and record the data in calibration matrix M_1
  - Analog for the remaining half of the gates on the remaining calibration circuits, and name the new matrix M_2.
  - Calculate the matrix $M_G = (M_1 + M_2)/2$. M_G serves the same purpose as M in QiEM.

- Hence the calibration circuits:



- Pack measurement outputs of the circuits on the left into the columns of 4 x 4 matrix M_1, the right M_2, Then determine M_g

General Error Mitigation

- Discussions
  - 🙂 Performs well also in deeper circuits
  - 🙂 Does not require *a priori* assumption of a specific error model
  - 🙁 Has to be implemented from scratch

General Error Mitigation

- GEM vs QiEM

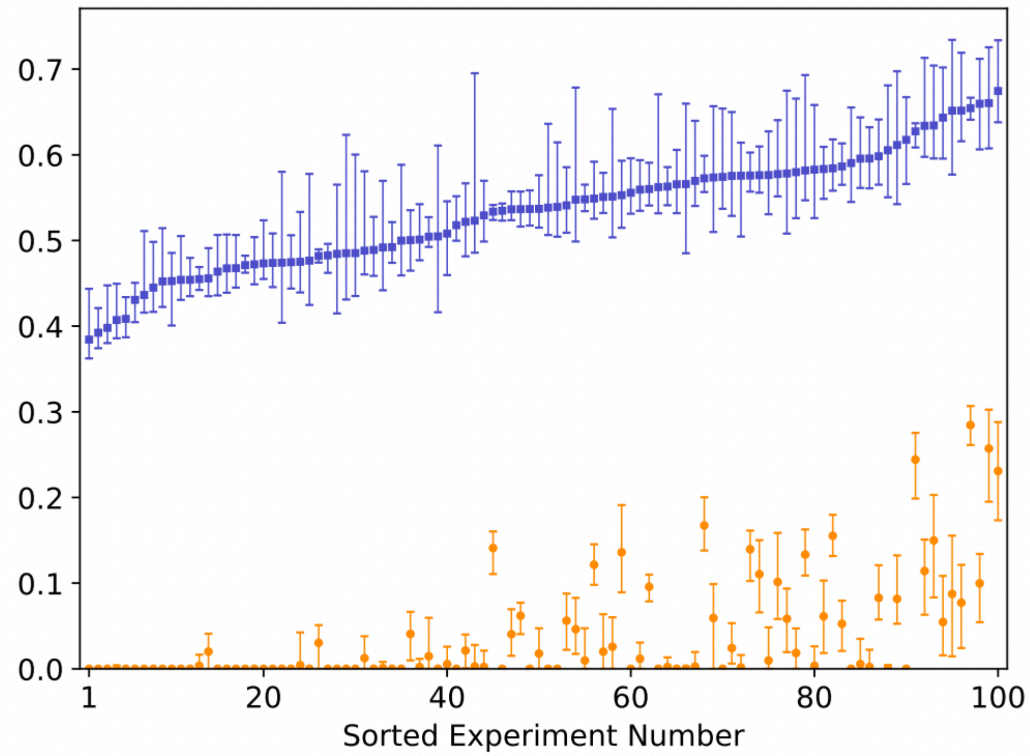


FIG. 7. Same as Fig. 4 except for $N = 2$ and $D \in [74, 80]$ with $\bar{D} = 79.38$. The **H** gate was not used in the gate-set. The device used was IBM Q Burlington [34].

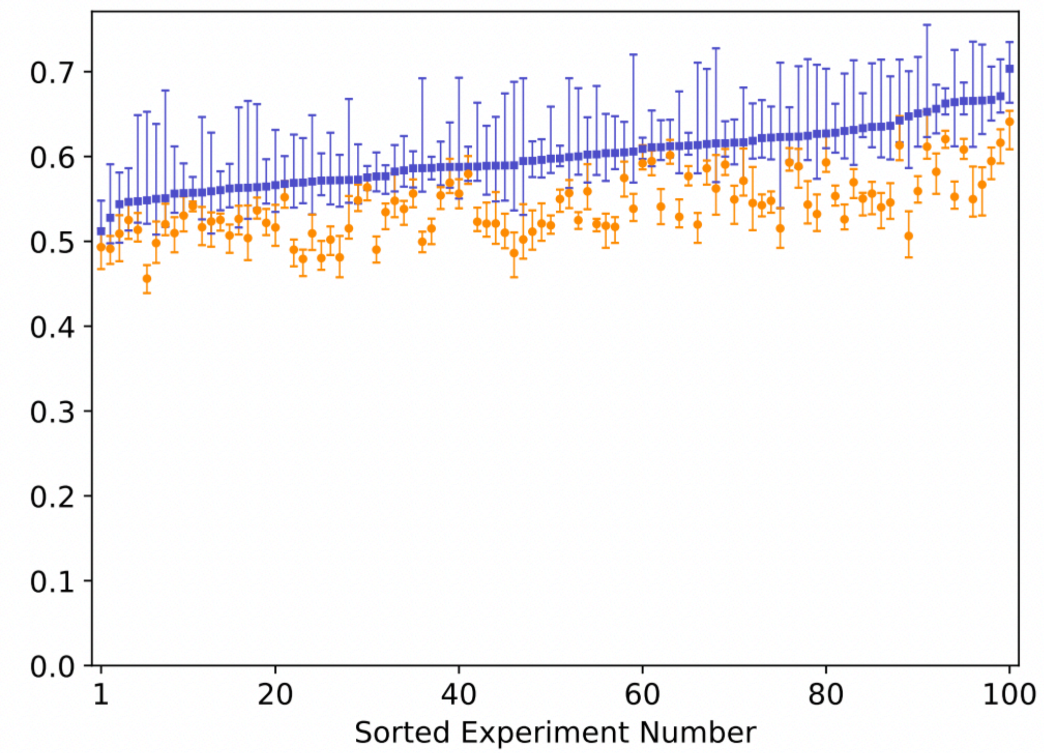


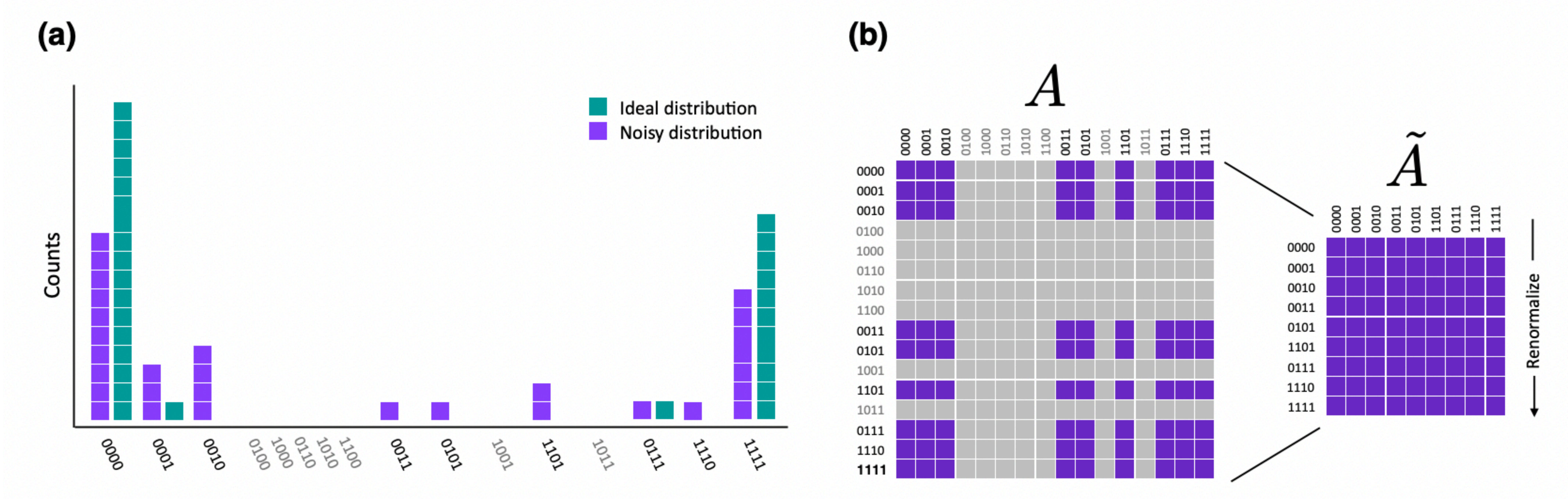
FIG. 10. Same as Fig. 7 except that Qiskit error mitigation was used for $D \in [72, 80]$ with $\bar{D} = 77.50$. The device used was IBM Q Burlington [34].

Source: https://arxiv.org/pdf/2011.10860.pdf

Blue: not mitigated; orange: mitigated; 2 qubits, similar D

M3: the future?

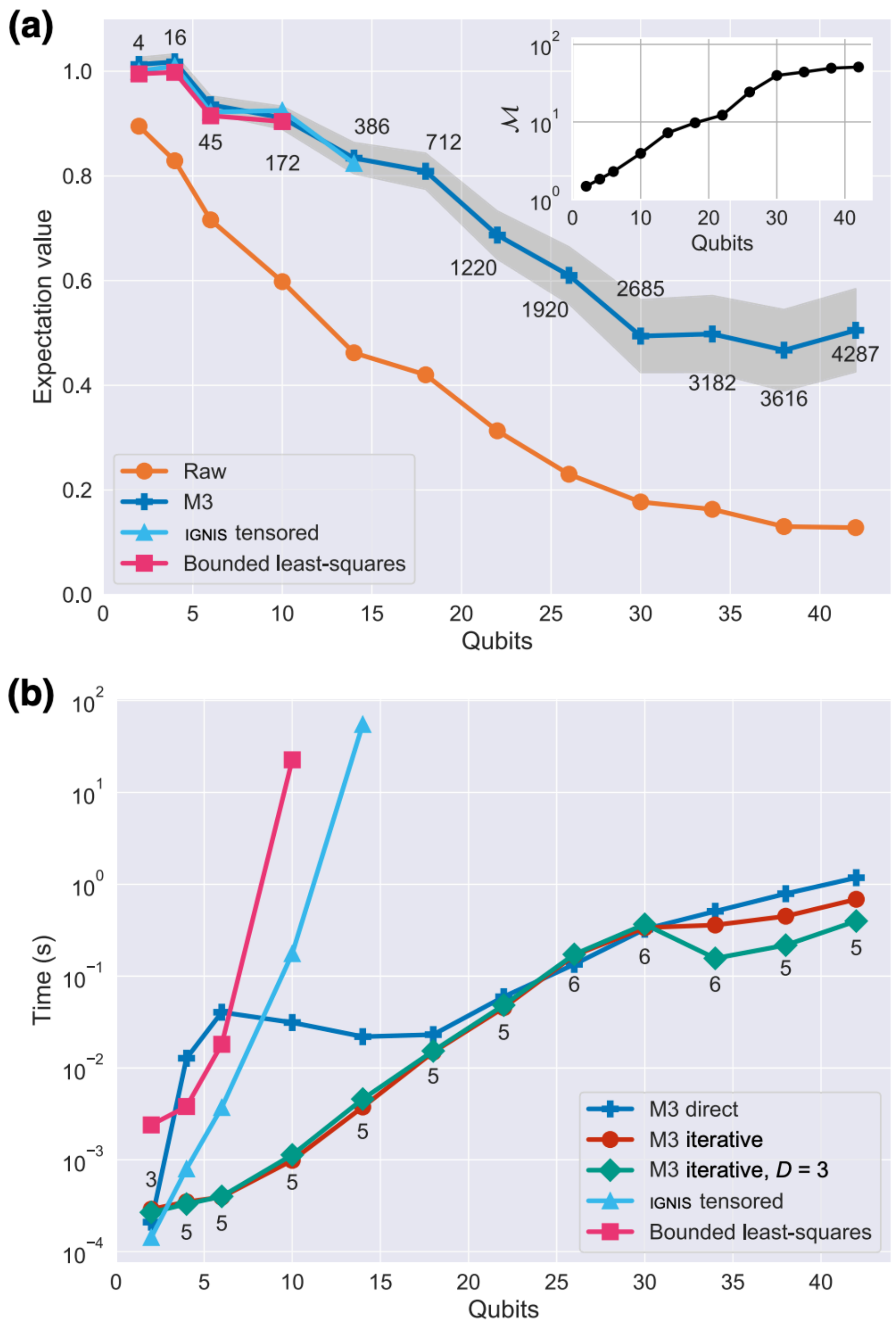
- Proposed by IBM Quantum researchers in Nov. 2021 and integrated in a Qiskit Runtime release
- Addresses the overhead introduced by the matrix-based approaches
- M3 stands for **m**atrix-free **m**easurement **m**itigation
- Subspace reduction, then solve the matrix. Convergence after $O(1)$



Source: https://journals.aps.org/prxquantum/pdf/10.1103/PRXQuantum.2.040326

- Remark: Qiskit Runtime offers other error mitigation methods besides M3, e.g. T-Rex, ZNE…

M3: the future?



Source: https://journals.aps.org/prxquantum/pdf/10.1103/PRXQuantum.2.040326

Conclusion

- Not requiring additional hardware, Quantum Error Mitigation might be a solution to error-tolerant quantum computing in the near term
- Qiskit error mitigation offers hands-on, adequate and basic measurement error mitigation
- With general error mitigation, better performance can be achieved in deeper circuits. It is independent of error models.
- M3 gives an outlook with its novel, matrix-free, hence overhead-reducing approach.

	Matrix-free?	Available in Qiskit?	Error model independent?
QiEM	👎	👍	👎
GEM	👎	👎	👍
M3	👍	👍	👎

References

- https://research.ibm.com/blog/quantum-error-suppression-mitigation-correction
- https://arxiv.org/pdf/2011.10860.pdf
- https://arxiv.org/pdf/2210.00921.pdf
- https://qiskit.org/textbook/ch-quantum-hardware/measurement-error-mitigation.html
- https://journals.aps.org/prxquantum/pdf/10.1103/PRXQuantum.2.040326
- https://quantum-computing.ibm.com/services/programs/docs/runtime/programs/
- https://qiskit.org/documentation/partners/qiskit_ibm_runtime/how_to/error-mitigation.html