Hypothesis Testing for Error Mitigation

Quantum Wednesday

Nate Stemen

Mar 8, 2023

Today's Paper

HYPOTHESIS TESTING FOR ERROR MITIGATION: HOW TO EVALUATE ERROR MITIGATION

Abdullah Ash Saki
Zapata Computing, Inc.
100 Federal Street, 20th Floor
Boston, MA 02110
saki@zapatacomputing.com

Amara Katabarwa
Zapata Computing, Inc.
100 Federal Street, 20th Floor
Boston, MA 02110,
amara@zapatcomputing.com

Salonik Resch
Zapata Computing, Inc.
100 Federal Street, 20th Floor
Boston, MA 02110,
Salonik.Resch@zapatcomputing.com

George Umbrarescu
Department of Physics and Astronomy
University College London
London, United Kingdom
george.umbrarescu.20@ucl.ac.uk

https://arxiv.org/abs/2301.02690

Overview

Two main questions:

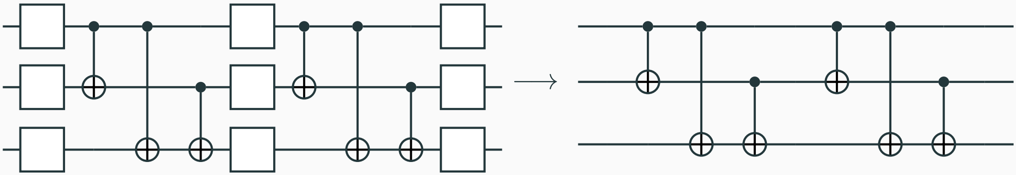
1. How can we understand "stacking" error mitigation techniques?
2. How can we assess the tradeoffs associated with a variety of techniques in a way that captures overhead?

Overview

Two main questions:

1. How can we understand “stacking” error mitigation techniques?
2. How can we assess the tradeoffs associated with a variety of techniques in a way that captures overhead?

Noise Estimation Circuits



$$\langle O \rangle = \frac{\langle O \rangle_{\text{noisy}}}{\frac{\# b_{00\cdots 0}}{N}}$$

Mitigating depolarizing noise on quantum computers with noise-estimation circuits

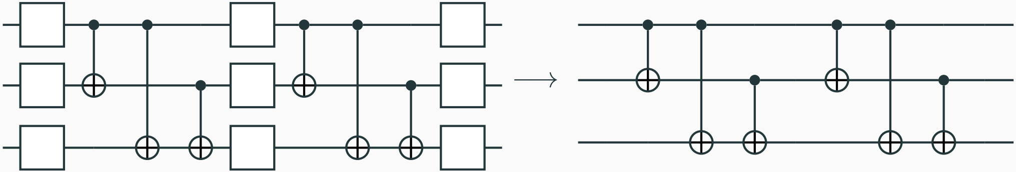
Miroslav Urbanek,[1,*] Benjamin Nachman,[2] Vincent R. Pascuzzi,[2]
Andre He,[2,†] Christian W. Bauer,[2] and Wibe A. de Jong[1,‡]

[1] *Computational Research Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA*
[2] *Physics Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA*

A significant problem for current quantum computers is noise. While there are many distinct noise channels, the depolarizing noise model often appropriately describes average noise for large circuits involving many qubits and gates. We present a method to mitigate the depolarizing noise by first estimating its rate with a noise-estimation circuit and then correcting the output of the target circuit using the estimated rate. The method is experimentally validated on the simulation of the Heisenberg model. We find that our approach in combination with readout-error correction, randomized compiling, and zero-noise extrapolation produces results close to exact results even for circuits containing hundreds of *CNOT* gates.

https://arxiv.org/abs/2103.08591

Noise Estimation Circuits



$$\langle O \rangle = \frac{\langle O \rangle_{\text{noisy}}}{\frac{\# b_{00\cdots 0}}{N}}$$

Mitigating depolarizing noise on quantum computers with noise-estimation circuits

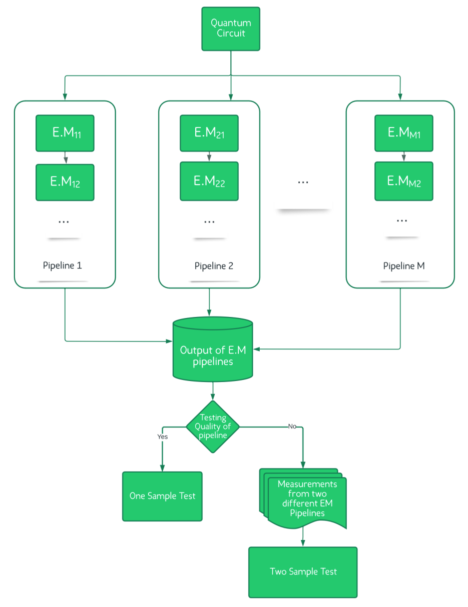
Miroslav Urbanek,[1,*] Benjamin Nachman,[2] Vincent R. Pascuzzi,[2]
Andre He,[2,†] Christian W. Bauer,[2] and Wibe A. de Jong[1,‡]

[1] *Computational Research Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA*
[2] *Physics Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA*

A significant problem for current quantum computers is noise. While there are many distinct noise channels, the depolarizing noise model often appropriately describes average noise for large circuits involving many qubits and gates. We present a method to mitigate the depolarizing noise by first estimating its rate with a noise-estimation circuit and then correcting the output of the target circuit using the estimated rate. The method is experimentally validated on the simulation of the Heisenberg model. We find that our approach in combination with readout-error correction, randomized compiling, and zero-noise extrapolation produces results close to exact results even for circuits containing hundreds of *CNOT* gates.

https://arxiv.org/abs/2103.08591

Error Mitigation Pipelines



Pipeline	Composition
$\mathcal{P}_1$	ZNE
$\mathcal{P}_2$	ZNE + MEM
$\mathcal{P}_3$	ZNE + DD
$\mathcal{P}_4$	ZNE + DD + MEM
$\mathcal{P}_5$	ZNE + RC
$\mathcal{P}_6$	ZNE + RC + MEM
$\mathcal{P}_7$	ZNE + RC + DD
$\mathcal{P}_8$	ZNE + RC + DD + MEM

$\mathcal{P}_i^{(E)}$ indicates extrapolation performed on scaled expectation values.

Error Mitigation Methods

1. Zero-Noise extrapolation
   - Scale factors: `[1, 3, 5]`
   - Folding methods: Global folding, `CNOT` folding
   - Extrapolation methods: Linear, Quadratic
2. Measurement Error Mitigation
   - $C_{\text{mitigated}} = M_{\text{calib}}^{-1} \cdot C_{\text{noisy}}$
3. Randomized Compiling
   - Convert coherent errors into incoherent ones
4. Dynamic Decoupling
   - X-X sequence

```
mitiq.zne.execute_with_zne(
    circuit,
    executor,
    factory=LinearFactory([1.0, 2.0, 3.0]),
    scale_noise=fold_global,
)
```

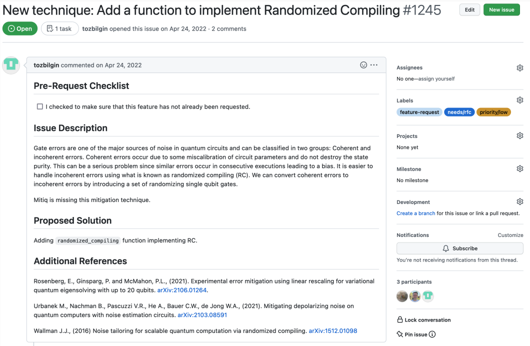
Error Mitigation Methods

1. Zero-Noise extrapolation
   - Scale factors: `[1, 3, 5]`
   - Folding methods: Global folding, `CNOT` folding
   - Extrapolation methods: Linear, Quadratic
2. Measurement Error Mitigation
   - $C_{\text{mitigated}} = M_{\text{calib}}^{-1} \cdot C_{\text{noisy}}$
3. Randomized Compiling
   - Convert coherent errors into incoherent ones
4. Dynamic Decoupling
   - X-X sequence

```
mitiq.rem.execute_with_rem(
    circuit,
    executor,
    observable,
    inverse_confusion_matrix=icm,
)
```

Error Mitigation Methods

1. Zero-Noise extrapolation
   - Scale factors: `[1, 3, 5]`
   - Folding methods: Global folding, `CNOT` folding
   - Extrapolation methods: Linear, Quadratic
2. Measurement Error Mitigation
   - $C_{\text{mitigated}} = M_{\text{calib}}^{-1} \cdot C_{\text{noisy}}$
3. Randomized Compiling
   - Convert coherent errors into incoherent ones
4. Dynamic Decoupling
   - X-X sequence



New technique: Add a function to implement Randomized Compiling #1245 Edit New issue

Open 1 task tozbilgin opened this issue on Apr 24, 2022 · 2 comments

tozbilgin commented on Apr 24, 2022

Pre-Request Checklist

- [ ] I checked to make sure that this feature has not already been requested.

Issue Description

Gate errors are one of the major sources of noise in quantum circuits and can be classified in two groups: Coherent and incoherent errors. Coherent errors occur due to some miscalibration of circuit parameters and do not destroy the state purity. This can be a serious problem since similar errors occur in consecutive executions leading to a bias. It is easier to handle incoherent errors using what is known as randomized compiling (RC). We can convert coherent errors to incoherent errors by introducing a set of randomizing single qubit gates.

Mitiq is missing this mitigation technique.

Proposed Solution

Adding `randomized_compiling` function implementing RC.

Additional References

Rosenberg, E., Ginsparg, P. and McMahon, P.L., (2021). Experimental error mitigation using linear rescaling for variational quantum eigensolving with up to 20 qubits. arXiv:2106.01264.

Urbanek M., Nachman B., Pascuzzi V.R., He A., Bauer C.W., de Jong W.A., (2021). Mitigating depolarizing noise on quantum computers with noise estimation circuits. arXiv:2103.08591

Wallman J.J., (2016) Noise tailoring for scalable quantum computation via randomized compiling. arXiv:1512.01098

Assignees
No one—assign yourself

Labels
feature-request needs/rfc priority/low

Projects
None yet

Milestone
No milestone

Development
Create a branch for this issue or link a pull request.

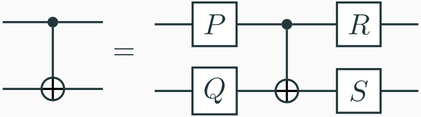
Notifications Customize
Subscribe
You're not receiving notifications from this thread.

3 participants

Lock conversation
Pin issue

Error Mitigation Methods

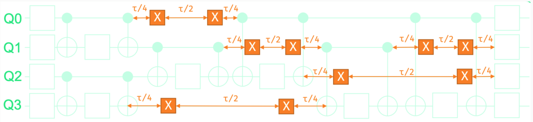
1. Zero-Noise extrapolation
    - Scale factors: `[1, 3, 5]`
    - Folding methods: Global folding, `CNOT` folding
    - Extrapolation methods: Linear, Quadratic
2. Measurement Error Mitigation
    - $C_{\text{mitigated}} = M_{\text{calib}}^{-1} \cdot C_{\text{noisy}}$
3. Randomized Compiling
    - Convert coherent errors into incoherent ones
4. Dynamic Decoupling
    - X-X sequence



P	Q	R	S
I	I	I	I
I	X	I	X
I	Y	Z	Y
$\vdots$			

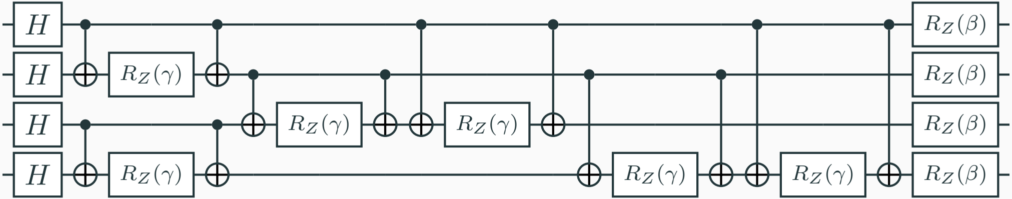
Error Mitigation Methods

1. Zero-Noise extrapolation
   - Scale factors: `[1, 3, 5]`
   - Folding methods: Global folding, `CNOT` folding
   - Extrapolation methods: Linear, Quadratic
2. Measurement Error Mitigation
   - $C_{\text{mitigated}} = M_{\text{calib}}^{-1} \cdot C_{\text{noisy}}$
3. Randomized Compiling
   - Convert coherent errors into incoherent ones
4. Dynamic Decoupling
   - X-X sequence




```python
mitiq.ddd.execute_with_ddd(
    circuit,
    executor,
    rule=mitiq.ddd.rules.xx,
)
```

Circuits



Ten (γ, β) pairs chosen to cover the possible ideal expectation values $(-0.62, 2)$.

Figure of Merit

What's harder

1. running 1 circuit with 10,000 shots, or
2. running 10 circuits with 1,000 shots each?

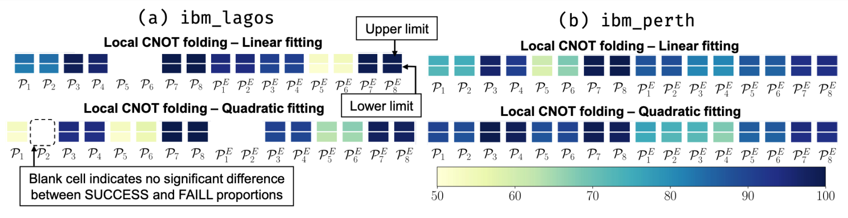
$$R = T(1+S)$$
$$S = -\sum_i p_{C_i} \ln(p_{C_i})$$
$$T = \sum_i N_{C_i} D_{C_i}$$
$$p_{C_i} = \frac{N_{C_i} D_{C_i}}{T}$$
$$M = \frac{PSR(\%)}{\epsilon \cdot R}$$

N_{C_i} = number of shots for circuit C_i

D_{C_i} = duration of circtui C_i

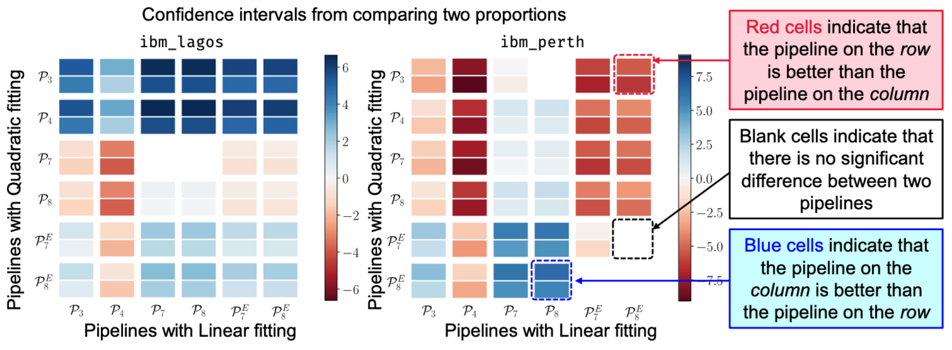
ϵ = median improvement factor (REM)

Results



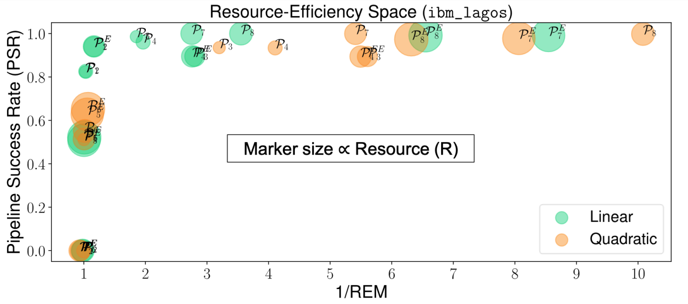
Pipeline	Composition
$\mathcal{P}_1$	ZNE
$\mathcal{P}_2$	ZNE + MEM
$\mathcal{P}_3$	ZNE + DD
$\mathcal{P}_4$	ZNE + DD + MEM
$\mathcal{P}_5$	ZNE + RC
$\mathcal{P}_6$	ZNE + RC + MEM
$\mathcal{P}_7$	ZNE + RC + DD
$\mathcal{P}_8$	ZNE + RC + DD + MEM

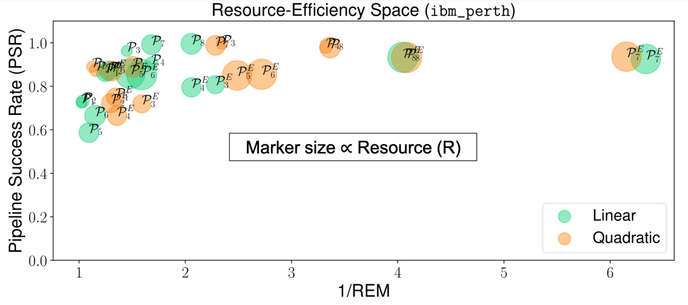
Results



Device	Pipeline	R	REM (lin)	REM (quad)	PSR (lin)	PSR (quad)	M (lin)	M (quad)
ibm_lagos	$\mathcal{P}_3$	2.8494	0.5397	0.3126	0.9858	0.9352	64.1009	104.9981
	$\mathcal{P}_4$	3.8006	0.5097	0.2435	0.9609	0.9339	49.6020	100.9129
	$\mathcal{P}_8$	9.9802	0.2815	0.0992	1.0000	0.9981	35.5945	100.8167
	$\mathcal{P}_7^E$	20.1687	0.1170	0.1240	0.9915	0.9780	42.0164	39.1058
ibm_perth	$\mathcal{P}_3$	2.3061	0.6899	0.4267	0.9623	0.9948	60.4875	101.0925
	$\mathcal{P}_4$	3.1189	0.5950	0.3007	0.9055	0.9811	48.7956	104.6124
	$\mathcal{P}_7^E$	16.5019	0.1578	0.1625	0.9258	0.9363	35.5540	34.9154

Results


Resource-Efficiency Space (ibm_lagos)
Pipeline Success Rate (PSR)
1/REM
Marker size ∝ Resource (R)
Linear
Quadratic


Resource-Efficiency Space (ibm_perth)
Pipeline Success Rate (PSR)
1/REM
Marker size ∝ Resource (R)
Linear
Quadratic

Thank you!