Digital zero noise extrapolation for quantum error mitigation

Will Zeng

QRE 2020, in Cyberspace
May 30ish, 2020

Joint work with: Tudor Giurgica-Tiron [Stanford], Yousef Hindy [Stanford], Ryan LaRose [Unitary Fund, UMich], Andrea Mari [Unitary Fund]

Outline

• Why error-mitigation?
• Zero-noise extrapolation (noise-scaling + extrapolation)
• Noise scaling:
  • *Unitary Folding*: a framework for digital noise scaling
  • *Parameter Noise scaling*: a noise model specific framework
• Extrapolation as inference
  • *Non-adaptive Methods*
  • *Adaptive Methods*
• Benchmarks
Error-mitigation is critical for noisy quantum computing

- Probabilistic Error Cancellation [1,2]
- Randomized Compiling [3]
- Dynamical Decoupling [4-7]
- Quantum optimal control [8,9]
- Zero-noise extrapolation

Cross-your-fingers method

- Uses additional qubits
- Requires fast classical control

Zero-noise extrapolation

1. Noise scaling

- Quantum Program
- Noise scaled quantum programs
- Execute on backend
- Noise scaled expectations
- Error mitigated expectation

2. Extrapolation

References:
Zero-noise Extrapolation can work very well

\[ \epsilon = 1 - e^{-2\lambda} \]

Zero-noise extrapolation
Previously introduced and studied with physical level noise scaling & Richardson extrapolation

Recalibration Noise-scaling

Richardson Extrapolation

$$E_K(\lambda) = E^* + \sum_{k=1}^{n} a_k \lambda^k + \mathcal{O}(\lambda^{n+1})$$

$$E_{\text{Rich}}(\lambda) = E_{\text{polv}}^{(d=m-1)}(\lambda) = c_0 + c_1 \lambda + \ldots + c_{m-1} \lambda^{m-1}$$

Zero-noise extrapolation
Previous benchmarks on one and two qubits

We extend and improve both noise scaling & extrapolation

### Recalibration Noise-scaling

- **Incorporates CNOT folding [4]**
- **Incorporates random folding [4]**

### Richardson Extrapolation

- **Extrapolation as inference**
  - Incorporates Exponential Fits [1]

### Parameter Noise Scaling

- Noise scaling can be performed at the instruction set level only.

### Adaptive Extrapolation

- 14-19X more accurately

---

We extend and improve both noise scaling & extrapolation

Recalibration Noise-scaling

Unitary Folding
Incorporates CNOT folding [4]
Incorporates random folding [4]

Parameter Noise Scaling

Richardson Extrapolation

Extrapolation as inference
Incorporates Exponential Fits [1]

Adaptive Extrapolation

Noise scaling can be performed at the instruction set level only.

And we do larger more systematic benchmarks

Outline

• Why error-mitigation?
• Zero-noise extrapolation (noise-scaling + extrapolation)
• Noise scaling:
  • *Unitary Folding*: a framework for digital noise scaling
  • *Parameter Noise scaling*: a noise model specific framework
• Extrapolation as inference
  • *Non-adaptive Methods*
  • *Adaptive Methods*
• Benchmarks
Def: a unitary fold $U \rightarrow U(U^\dagger U)^n$

Method | Subset of indices to fold
--- | ---
From left | $S = \{1, 2, \ldots, s\}$
From right | $S = \{d, d - 1, \ldots, d - s + 1\}$
At random | $S = s$ different indices randomly sampled without replacement from $\{1, 2, \ldots, d\}$.

Fold some subset $n$ times

$d$ scales to $d + 4$
Unitary folding performs well

**RB Circuits (2-qubits)**

97.9% unmitigated to 99.0% mitigated

Exact density matrix simulations with 1% depolarizing as base noise

**Random Circuits (6-qubits)**

0.114 unmitigated to 0.075 mitigated avg error

250 random circuits of depth 40
Unitary folding ZNE improve variational algorithms
Study of MAXCUT solved with QAOA

% closer to optimal with ZNE (p=1)

% closer to optimal with ZNE (p=2)

Exact density matrix simulation of 14 Erdor-Renyi random graphs at each size. Solved with Nelder-Mead optimized QAOA under 2% depolarizing base noise. Used global unitary folding and linear extrapolation.
Outline

• Why error-mitigation?
• Zero-noise extrapolation (noise-scaling + extrapolation)
• Noise scaling:
  • *Unitary Folding*: a framework for digital noise scaling
  • *Parameter Noise scaling*: a noise model specific framework
• Extrapolation as inference
  • *Non-adaptive Methods*
  • *Adaptive Methods*
• Benchmarks
Other instruction layer noise scalings are possible

For particular noise model: parameter noise

- Unitary Folding ZNE assumes that noise scales with gate # / circuit depth
- Other noise models can scale with different parameters

**Parameter Noise**

\[ U(\vec{\theta}) \leftrightarrow U(\vec{\theta} + \vec{\epsilon}') \]

**Parameter Noise Scaling**

Manually apply sampled noise

\[ \vec{\epsilon}' \sim \mathcal{N}(0, \sigma^2)^m \implies \vec{\epsilon}^{\lambda} \sim \mathcal{N}(0, (\lambda \sigma)^2)^m \]

---


Parameter scaled ZNE has similar performance to unitary folding

50 random six-qubit circuits. Underlying noise is an angle noise channel at $\sigma^2 = 0.001$. ZNE with linear extrapolation with noise scale factors $\lambda = \{1, 2, 3\}$. Results were obtained with exact density matrix simulations.
• Why error-mitigation?
• Zero-noise extrapolation (noise-scaling + extrapolation)
• Noise scaling:
  • *Unitary Folding*: a framework for digital noise scaling
  • *Parameter Noise scaling*: a noise model specific framework
• Extrapolation as inference
  • *Non-adaptive Methods*
  • *Adaptive Methods*
• Benchmarks
Extrapolation as Inference: Non-adaptive

Given the curve above 1, infer the intercept
Extrapolation as Inference: Non-adaptive

Given the curve above 1, infer the intercept

**Algorithm 1: Generic non-adaptive extrapolation**

**Data:** A set of increasing noise scale factors 
\[ \lambda = \{\lambda_1, \lambda_2, \ldots \lambda_m\} \], with \( \lambda_j \geq 1 \) and fixed number of samples \( N \) for each \( \lambda_j \).

**Result:** A mitigated expectation value
\[ y \leftarrow 0; \]

begin
for \( \lambda_j \in \lambda \) do
  \[ y_j \leftarrow ComputeExpectation(\lambda_j, N); \]
  Append\((y, y_j)\);
  /* Arbitrary best fit algorithm (e.g., least squares) */
  \[ \Gamma^* \leftarrow BestFit(E_{model}(\lambda; \Gamma), (\lambda, y)); \]
return \( E_{model}(0; \Gamma^*) \);

**ZNE extrapolation comparison on IBMQ Armonk qubit**

53 1-qubit RB circuits of depth 200
Outline

• Why error-mitigation?
• Zero-noise extrapolation (noise-scaling + extrapolation)
• Noise scaling:
  • Unitary Folding: a framework for digital noise scaling
  • Parameter Noise scaling: a noise model specific framework
• Extrapolation as inference
  • Non-adaptive Methods
  • Adaptive Methods
• Benchmarks
Adaptive Zero-noise extrapolation

1. Noise scaling

Quantum Program

Noise scaled quantum programs

Execute on backend

2. Extrapolation

Error mitigated expectation

Noise scaled expectations
Adaptive Zero-noise extrapolation

1. Noise scaling

Loop to convergence

- Quantum Program
- Noise scaled quantum programs

2. Extrapolation

- Error mitigated expectation
- Noise scaled expectations

Execute on backend
Adaptive Zero-noise extrapolation
Optimally choose the next noise scaling (and sample #) based on data seen so far

```
Algorithm 2: Generic adaptive extrapolation

Data: An initial set of $m$ noise scale factors
\[ \lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_m\}, \] with $\lambda_j \geq 1$, $m$ sample numbers $N = \{N_1, N_2, \ldots, N_m\}$ and a maximum number of total samples $N_{max}$.

Result: A mitigated expectation value

begin
  /* Initialization */
  y \leftarrow \emptyset;
  for $\lambda_j \in \lambda$ do
    y_j \leftarrow ComputeExpectation($\lambda_j, N_j$);
    Append ($y, y_j$);
  /* Adaptive loop */
  N_{used} \leftarrow 0;
  while $N_{used} < N_{max}$ do
    \[ \Gamma^* \leftarrow BestFit(E_{model}(\lambda, \Gamma), (\lambda, y)) \];
    \[ \lambda_{next} \leftarrow NewScale(\Gamma^*, \lambda, y) \];
    \[ N_{next} \leftarrow NewNumSamples(\Gamma^*, \lambda, y) \];
    \[ y_{next} \leftarrow ComputeExpectation(\lambda_{next}, N_{next}) \];
    Append ($\lambda, \lambda_{next}$);
    Append ($y, y_{next}$);
    \[ N_{used} \leftarrow N_{used} + N_{next} \];
  return $E_{model}(0; \Gamma^*)$;
```

Loop to convergence

1. Noise scaling
   - Quantum Program
   - Noise scaled quantum programs

2. Extrapolation
   - Error mitigated expectation
   - Noise scaled expectations

Execute on backend
Optimal adaptive exponential ZNE

Optimally choose the next noise scaling (and sample #) based on data seen so far

Exponential measurement model:

\[ y \mid \lambda \sim \mathcal{N} \left( a + b e^{-c \lambda}, \sigma^2 \right) \]

Assumptions:

- Know minimum accessible noise level \( \lambda_1 \)
- Know asymptotic value \( a \)

Can show that it is best to sample at:

\[ \lambda_1 \text{ and } \lambda_2 = \lambda_1 + \frac{1.28}{c} \]

We are interested in the intercept \( a + b \)

We will do inference on \( b \) and \( c \)
Optimal adaptive exponential ZNE

Optimally choose the next noise scaling (and sample #) based on data seen so far

![Graph showing the relationship between observable and noise level.]
Optimal adaptive exponential ZNE

Optimally choose the next noise scaling (and sample #) based on data seen so far

---

**Diagram:**

- **Observable vs. Noise level $\lambda$:**
  - Blue line: true mean
  - Green line: exp. fit
  - Dotted line: $\lambda_{\text{min}}$
  - Black dots: measurements

- **Log-log likelihood (b, θ):**
  - Color scale indicating log(likelihood) values from $-3.434$ to $-3.418$.
Optimal adaptive exponential ZNE

Optimally choose the next noise scaling (and sample #) based on data seen so far
Optimal adaptive exponential ZNE

Optimally choose the next noise scaling (and sample #) based on data seen so far
Optimal adaptive exponential ZNE
Optimally choose the next noise scaling (and sample #) based on data seen so far
Optimal adaptive exponential ZNE

Optimally choose the next noise scaling (and sample #) based on data seen so far
Optimal adaptive exponential ZNE

Optimally choose the next noise scaling (and sample #) based on data seen so far
Optimal adaptive exponential ZNE

Optimally choose the next noise scaling (and sample #) based on data seen so far
Optimal adaptive exponential ZNE

Optimally choose the next noise scaling (and sample #) based on data seen so far
Optimal adaptive exponential ZNE

Optimally choose the next noise scaling (and sample #) based on data seen so far.
Adaptive ZNE can give a 5X speedup

Error by number of total samples taken (proportional to runtime)

5 qubit RB circuits of depth 10 under 5% simulated depolarizing noise
• Why error-mitigation?
• Zero-noise extrapolation (noise-scaling + extrapolation)
• Noise scaling:
  • *Unitary Folding*: a framework for digital noise scaling
  • *Parameter Noise scaling*: a noise model specific framework
• Extrapolation as inference
  • *Non-adaptive Methods*
  • *Adaptive Methods*
• Benchmarks
Exponential extrapolation performs best

<table>
<thead>
<tr>
<th>Scaling</th>
<th>Extrapolation</th>
<th>Error % (dep.)</th>
<th>Error % (amp. damp.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>none</td>
<td>unmitigated</td>
<td>29.9 ± 5.1</td>
<td>16.7 ± 4.0</td>
</tr>
<tr>
<td>circuit</td>
<td>linear ((d = 1))</td>
<td>14.6 ± 4.6</td>
<td>5.40 ± 2.3</td>
</tr>
<tr>
<td>circuit</td>
<td>quadratic ((d = 2))</td>
<td>6.35 ± 3.6</td>
<td>3.53 ± 3.4</td>
</tr>
<tr>
<td>circuit</td>
<td>Richardson ((d = 3))</td>
<td>17.6 ± 11</td>
<td>17.9 ± 16</td>
</tr>
<tr>
<td>circuit</td>
<td>exponential ((a = 0.25))</td>
<td>2.73 ± 1.9</td>
<td>2.06 ± 1.6</td>
</tr>
<tr>
<td>circuit</td>
<td>adapt. exp. ((a = 0.25))</td>
<td><strong>1.27 ± 1.1</strong></td>
<td>2.69 ± 2.8</td>
</tr>
<tr>
<td>at random</td>
<td>linear ((d = 1))</td>
<td>15.6 ± 5.3</td>
<td>5.20 ± 2.4</td>
</tr>
<tr>
<td>at random</td>
<td>quadratic ((d = 2))</td>
<td>5.54 ± 4.4</td>
<td>8.00 ± 8.1</td>
</tr>
<tr>
<td>at random</td>
<td>Richardson ((d = 3))</td>
<td>30.0 ± 24</td>
<td>24.0 ± 18</td>
</tr>
<tr>
<td>at random</td>
<td>exponential ((a = 0.25))</td>
<td><strong>2.84 ± 1.8</strong></td>
<td><strong>0.95 ± 1.0</strong></td>
</tr>
<tr>
<td>at random</td>
<td>adapt. exp. ((a = 0.25))</td>
<td>1.77 ± 1.4</td>
<td>2.18 ± 1.2</td>
</tr>
<tr>
<td>from left</td>
<td>linear ((d = 1))</td>
<td>14.4 ± 4.5</td>
<td>5.16 ± 2.3</td>
</tr>
<tr>
<td>from left</td>
<td>quadratic ((d = 2))</td>
<td>6.73 ± 3.7</td>
<td>3.88 ± 3.7</td>
</tr>
<tr>
<td>from left</td>
<td>Richardson ((d = 3))</td>
<td>18.4 ± 12</td>
<td>16.1 ± 13</td>
</tr>
<tr>
<td>from left</td>
<td>exponential ((a = 0.25))</td>
<td>3.17 ± 2.1</td>
<td>2.19 ± 2.0</td>
</tr>
<tr>
<td>from left</td>
<td>adapt. exp. ((a = 0.25))</td>
<td>1.43 ± 1.1</td>
<td>3.08 ± 3.6</td>
</tr>
</tbody>
</table>

Average of 20 different two-qubit randomized benchmarking circuits with mean depth 27.1%. Depolarizing noise. Amplitude damping channel with \(\gamma = 0.01\). For all non-adaptive methods we used \(\lambda = \{1, 1.5, 2, 2.5\}\). Adaptive extrapolation was iterated up to 4 scale factors.
We can now do zero-noise extrapolation:

with only gate level access

and

14-19X more accurately

Upcoming: *mitiq*