QuantumNAT: Quantum Noise-Aware Training with Noise Injection, Quantization and Normalization

Hanrui Wang¹, Jiaqi Gu², Yongshan Ding³, Zirui Li⁴, Frederic T. Chong⁵, David Z. Pan², Song Han¹

¹Massachusetts Institute of Technology, ²University of Texas at Austin, ³Yale University,
⁴Shanghai Jiao Tong University, ⁵University of Chicago
Outline

• Overview
• Background
• QuantumNAT Methodology
• Evaluation
• TorchQuantum Library
• Conclusion
Outline

• Overview
• Background
• QuantumNAT Methodology
• Evaluation
• TorchQuantum Library
• Conclusion
Overview: Noise-Aware Training

• Quantum circuits are noisy
  • Noise severely *degrades* the circuit performance
Overview: Noise-Aware Training

- Quantum circuits are noisy
  - Noise severely **degrades** the circuit performance
- Add real device noise during circuit training on classical simulator
  - Improve robustness on real quantum machines
Outline

• Overview
• Background
• QuantumNAT Methodology
• Evaluation
• TorchQuantum Library
• Conclusion
NISQ Era

• Noisy Intermediate-Scale Quantum (NISQ)
  • **Noisy**: qubits are sensitive to environment; quantum gates are unreliable

![Gate Error Rate](https://quantum-computing.ibm.com/)

**Single-qubit Pauli-X error**
Avg 1.718e-3
min 1.470e-4 max 7.486e-2

**CNOT error**
Avg 6.973e-2
min 5.403e-3 max 1.000e+0
NISQ Era

- Noisy Intermediate-Scale Quantum (NISQ)
  - **Noisy**: qubits are sensitive to environment; quantum gates are unreliable
  - **Limited number** of qubits: tens to hundreds of qubits

Gate Error Rate
https://quantum-computing.ibm.com/

Google Sycamore
https://www.nature.com/articles/s41586-019-1666-5

IBM Washington
https://quantum-computing.ibm.com/
Parameterized Quantum Circuits (PQC)

- Parameterized Quantum Circuits (PQC)
- Quantum circuit with fixed gates and parameterized gates
Parameterized Quantum Circuits (PQC)

- Parameterized Quantum Circuits (PQC)
- Quantum circuit with fixed gates and parameterized gates

- PQCs are commonly used in hybrid classical-quantum models and show promises to achieve quantum advantage
  - Variational Quantum Eigensolver (VQE)
  - Quantum Neural Networks (QNN)
  - Quantum Approximate Optimization Algorithm (QAOA)
Quantum Neural Networks (QNN)

- QNN is one kind of PQC for machine learning tasks
  - Encoder
  - Trainable Quantum Layers
  - Measurements
Challenge of PQC: Noise

- Noise *degrades* the PQC reliability
- Large *gap* between the noise-free simulation and real deployment

![Graph showing the comparison between Noise-Free Simulation and Measured on IBMQ-Yorktown accuracy against number of parameters. The graph highlights a large gap due to gate errors.]
Outline

- Overview
- Background
- QuantumNAT Methodology
- Evaluation
- TorchQuantum Library
- Conclusion
Three Techniques in QuantumNAT

1. Post-Measurement Normalization
   - Noise-free
   - Sensitive Info Loss
   - Noisy Norm.
   - Match

2. Real QC-backed Noise Injection
   - Sensitive
   - Error Margin
   - Small Margin
   - Inject Real QC Noise into Training
   - Robust
   - Margin

3. Post-Measurement Quantization
   - Quantum Error
   - Denoising via Quantization
Post-Measurement Normalization

- Normalize the measurement outcome (expectation value)
  - Along the **batch** dimension
- Measurement outcome distribution of 50 quantum circuits:

![Graph showing measurement outcome distribution](image)

- No normalization
- Noise-Free Simulation
- Real Device

**Qubit 1**
**SNR=0.89**
Post-Measurement Normalization

- Normalize the measurement outcome (expectation value)
  - Along the **batch** dimension
- Measurement outcome distribution of 50 quantum circuits:

![Histograms showing SNR comparison](image)

- **No normalization**
  - Qubit 1
    - SNR=0.89
  - Qubit 1
    - SNR=5.75

- **With normalization**
  - Qubit 1
    - SNR=5.75
Noise Injection

- Inject noise during training on classical simulator
  - Pauli error
  - Readout error
Noise Injection

- Inject noise during training on classical simulator
  - Pauli error
  - Readout error
Noise Injection

- Inject noise during training on classical simulator
  - Pauli error
  - Readout error
Noise Injection

- Inject noise during training on classical simulator
  - Pauli error
  - Readout error
Post-Measurement Quantization

- Quantize measurement outcomes (expectation values)
  - Denoising effect
  - Small errors will be mitigated

![Diagram showing quantum error denoising via quantization]

**Quantum Error**

**Denoising via Quantization**
Post-Measurement Quantization

- Quantize measurement outcomes (expectation values)
  - Denoising effect
  - Small errors will be mitigated

- **Loss** term to encourage measurement outcomes to be close to **centroids**
Post-Measurement Quantization

- Quantization reduces errors and improves SNR

<table>
<thead>
<tr>
<th>Batch</th>
<th>Qubit</th>
<th>Errors Before Quantize</th>
<th>MSE=0.235, SNR=4.256</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0.8</td>
<td>-0.3</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>-0.3</td>
<td>0.5</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0.3</td>
<td>-0.2</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>0.1</td>
<td>-0.3</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>-0.1</td>
<td>1.1</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0.1</td>
<td>-1.1</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0.4</td>
<td>0.1</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>0.0</td>
<td>-0.0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1.1</td>
<td>-0.0</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>-1.1</td>
<td>0.1</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0.0</td>
<td>0.1</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>0.0</td>
<td>-0.3</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>-0.3</td>
<td>0.2</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>1.2</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0.0</td>
<td>1.0</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Batch</th>
<th>Qubit</th>
<th>Errors After Quantize</th>
<th>MSE=0.167, SNR=6.455</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1.0</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0.0</td>
<td>1.0</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>
Outline

• Overview
• Background
• QuantumNAT Methodology
• Evaluation
• TorchQuantum Library
• Conclusion
Evaluation

• Benchmarks
  • Quantum Machine Learning task: MNIST 10-class, 4-class, 2-class, Fashion MNIST 10-class, 4-class, 2-class, Vowel 4-class, Cifar-2 class

• Quantum Devices
  • IBMQ
  • #Qubits: 5 to 15
  • Quantum Volume: 8 to 32
Evaluation

- QuantumNAT significantly improves real measurement accuracy

Severe accuracy drop because of quantum errors on real devices

Table 2

<table>
<thead>
<tr>
<th></th>
<th>Noise-Free Simulation</th>
<th>IBMQ-Yorktown</th>
<th>Lima</th>
<th>Santiago</th>
</tr>
</thead>
<tbody>
<tr>
<td>MNIST-4</td>
<td>0.8666666667</td>
<td>0.2333333333</td>
<td>0.5566666667</td>
<td>0.7333333333</td>
</tr>
<tr>
<td>1-Qubit Gate Error Rate</td>
<td>1.01E-03</td>
<td>4.84E-04</td>
<td>2.03E-04</td>
<td>0.00E+00</td>
</tr>
<tr>
<td>Accuracy</td>
<td>0.87</td>
<td>0.64</td>
<td>0.56</td>
<td>0.73</td>
</tr>
</tbody>
</table>

+QuantumNAT
Consistent Improvements on Various Benchmarks

- Different quantum devices
- Different models
- Different tasks

<table>
<thead>
<tr>
<th>Model</th>
<th>Method</th>
<th>MNIST-4</th>
<th>Fash.-4</th>
<th>Vow.-4</th>
<th>MNIST-2</th>
<th>Fash.-2</th>
<th>Cifar-2</th>
</tr>
</thead>
<tbody>
<tr>
<td>2B×12L</td>
<td>Baseline</td>
<td>0.30</td>
<td>0.32</td>
<td>0.28</td>
<td>0.84</td>
<td>0.78</td>
<td>0.51</td>
</tr>
<tr>
<td></td>
<td>+ Post Norm.</td>
<td>0.41</td>
<td>0.61</td>
<td>0.29</td>
<td>0.87</td>
<td>0.68</td>
<td>0.56</td>
</tr>
<tr>
<td>Santiago</td>
<td>+ Gate Insert.</td>
<td>0.61</td>
<td>0.70</td>
<td>0.44</td>
<td>0.93</td>
<td>0.86</td>
<td>0.57</td>
</tr>
<tr>
<td></td>
<td>+ Post Quant.</td>
<td>0.68</td>
<td>0.75</td>
<td>0.48</td>
<td>0.94</td>
<td>0.88</td>
<td>0.59</td>
</tr>
<tr>
<td>2B×2L</td>
<td>Baseline</td>
<td>0.43</td>
<td>0.56</td>
<td>0.25</td>
<td>0.68</td>
<td>0.70</td>
<td>0.52</td>
</tr>
<tr>
<td>Yorktown</td>
<td>+ Post Norm.</td>
<td>0.57</td>
<td>0.60</td>
<td>0.38</td>
<td>0.86</td>
<td>0.72</td>
<td>0.56</td>
</tr>
<tr>
<td></td>
<td>+ Gate Insert.</td>
<td>0.58</td>
<td>0.60</td>
<td>0.45</td>
<td>0.91</td>
<td>0.85</td>
<td>0.57</td>
</tr>
<tr>
<td></td>
<td>+ Post Quant.</td>
<td>0.62</td>
<td>0.65</td>
<td>0.44</td>
<td>0.93</td>
<td>0.86</td>
<td>0.60</td>
</tr>
<tr>
<td>2B×6L</td>
<td>Baseline</td>
<td>0.28</td>
<td>0.26</td>
<td>0.20</td>
<td>0.46</td>
<td>0.52</td>
<td>0.50</td>
</tr>
<tr>
<td>Belem</td>
<td>+ Post Norm.</td>
<td>0.52</td>
<td>0.57</td>
<td>0.33</td>
<td>0.81</td>
<td>0.62</td>
<td>0.51</td>
</tr>
<tr>
<td></td>
<td>+ Gate Insert.</td>
<td>0.52</td>
<td>0.60</td>
<td>0.37</td>
<td>0.84</td>
<td><strong>0.82</strong></td>
<td>0.57</td>
</tr>
<tr>
<td></td>
<td>+ Post Quant.</td>
<td><strong>0.58</strong></td>
<td><strong>0.62</strong></td>
<td><strong>0.41</strong></td>
<td><strong>0.88</strong></td>
<td>0.80</td>
<td><strong>0.61</strong></td>
</tr>
<tr>
<td>3B×10L</td>
<td>Baseline</td>
<td>0.29</td>
<td>0.36</td>
<td>0.21</td>
<td>0.54</td>
<td>0.46</td>
<td>0.49</td>
</tr>
<tr>
<td>Athens</td>
<td>+ Post Norm.</td>
<td>0.44</td>
<td>0.46</td>
<td>0.37</td>
<td>0.51</td>
<td>0.51</td>
<td>0.50</td>
</tr>
<tr>
<td></td>
<td>+ Gate Insert.</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>+ Post Quant.</td>
<td><strong>0.56</strong></td>
<td><strong>0.64</strong></td>
<td><strong>0.41</strong></td>
<td><strong>0.87</strong></td>
<td><strong>0.64</strong></td>
<td><strong>0.53</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Model</th>
<th>Method</th>
<th>MNIST-10</th>
<th>Fash.-10</th>
<th>Avg.-All</th>
</tr>
</thead>
<tbody>
<tr>
<td>2B×2L</td>
<td>Baseline</td>
<td>0.11</td>
<td>0.09</td>
<td>0.42</td>
</tr>
<tr>
<td>Melbo.</td>
<td>+ Post Norm.</td>
<td>0.08</td>
<td>0.12</td>
<td>0.52</td>
</tr>
<tr>
<td></td>
<td>+ Gate Insert.</td>
<td>0.25</td>
<td>0.24</td>
<td>0.61</td>
</tr>
<tr>
<td></td>
<td>+ Post Quant.</td>
<td><strong>0.34</strong></td>
<td><strong>0.31</strong></td>
<td><strong>0.64</strong></td>
</tr>
</tbody>
</table>
### Consistent Improvements on Various Benchmarks
- Different gate set design spaces

<table>
<thead>
<tr>
<th>Design Space</th>
<th>MNIST-4</th>
<th>Fashion-2</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Yorktown</td>
<td>Santiago</td>
<td>Yorktown</td>
</tr>
<tr>
<td>‘ZZ+RY’ +QuantumNAT</td>
<td>0.43</td>
<td>0.57</td>
<td>0.80</td>
</tr>
<tr>
<td>‘RXYZ’ +QuantumNAT</td>
<td>0.57</td>
<td>0.61</td>
<td>0.88</td>
</tr>
<tr>
<td>‘ZX+XX’ +QuantumNAT</td>
<td>0.29</td>
<td>0.51</td>
<td>0.52</td>
</tr>
<tr>
<td>‘RXYZ+U1+CU3’ +QuantumNAT</td>
<td>0.28</td>
<td>0.25</td>
<td>0.48</td>
</tr>
</tbody>
</table>
Ablation Study on Noise Injection Method

- Gate insertion is better than rotation angle perturbation
Ablation Study on Noise Injection Method

- Gate insertion is better than measurement outcome perturbation.
Visualization

- QuantumNAT stretches the distribution of features
- MNIST-2 classification task
Outline

• Overview
• Background
• QuantumNAT Methodology
• Evaluation
• TorchQuantum Library
• Conclusion
Open-source: TorchQuantum

- TorchQuantum — An open-source library for interdisciplinary research of quantum computing and machine learning
- [https://github.com/mit-han-lab/torchquantum](https://github.com/mit-han-lab/torchquantum)
Open-source: TorchQuantum

- TorchQuantum — An open-source library for interdisciplinary research of quantum computing and machine learning
- https://github.com/mit-han-lab/torchquantum

- Quantum ML
  - Quantum neural networks
  - Quantum kernel methods

Faster
Higher Scalability
Open-source: TorchQuantum

- TorchQuantum — An open-source library for interdisciplinary research of quantum computing and machine learning
- [https://github.com/mit-han-lab/torchquantum](https://github.com/mit-han-lab/torchquantum)

- ML for Quantum
  - ML optimizes quantum compilation
TorchQuantum Features

- Features
  - Easy construction of **parameterized quantum circuits** such as Quantum Neural Networks in PyTorch
  - Support **batch mode inference and training** on GPU/CPU, supports highly-parallelized training
  - Support **easy deployment** on real quantum devices such as IBMQ
  - Provide tutorials, videos and example projects of QML and using ML to optimize quantum computer system problems
Outline

• Overview
• Background
• QuantumNAT Methodology
• Evaluation
• TorchQuantum Library
• Conclusion
Conclusion

- **QuantumNAT**: makes PQC **parameters** more noise-robust
  - Post-measurement Normalization
  - Noise injection
  - Post-measurement Quantization
- Achieve 94% 2-class and 34% 10-class classification accuracy
- Open-sourced **TorchQuantum** library for Quantum + ML research

https://github.com/mit-han-lab/torchquantum

gmlsys.mit.edu