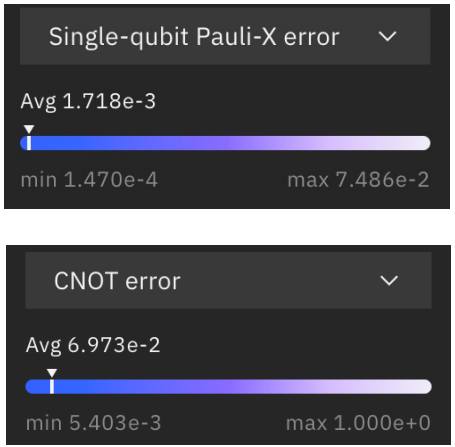
Graph Transformer for Quantum Circuit Reliability Estimation

20221014@IBM Introduction Meeting

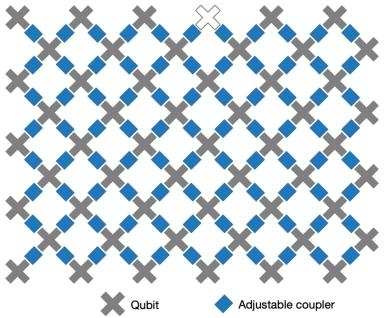
Quantum Computing in NISQ Era

- Noisy Intermediate-Scale Quantum (NISQ)
  - **Noisy**: qubits are sensitive to environment; quantum gates are unreliable
  - **Limited number of qubits**: tens to hundreds of qubits
  - **Limited connectivity:** no all-to-all connections



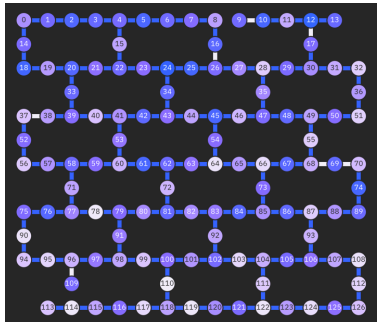
IBMQ Gate Error Rate

https://quantum-computing.ibm.com/



Google Sycamore 53Q

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



IBM Washington 127Q

https://quantum-computing.ibm.com/

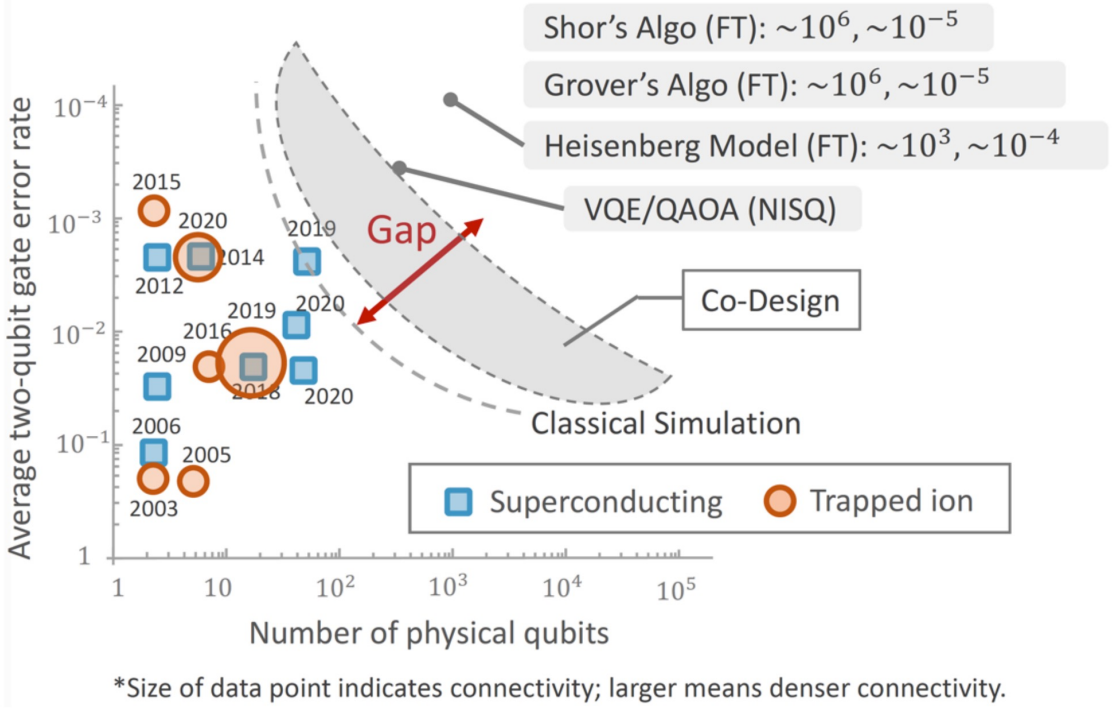
Torch Quantum

Materials at: https://torchquantum.org

MIT HAN LAB

A Large Gap between Powerful Quantum Algorithms and Current Devices

- Close the gap with **machine learning** and **hardware-aware algorithm design**



Fred Chong, https://indico.fnal.gov/event/47147/attachments/138920/174186/Chong-JTF-2021.pdf18.pdf

Materials at: https://torchquantum.org

Good Infrastructure is Critical

- To enable ML-assisted hardware-aware quantum algorithm design
- Need a simulation framework on classical computer
  - Fast
  - PyTorch native
  - Portable
  - Scalable
  - Analyze circuit **behavior**
  - Study **noise** impact
  - Develop **ML model** for quantum optimization

TorchQuantum Library

- A fast library for classical simulation of quantum circuit in **PyTorch**
  - Automatic **gradient** computation for training parameterized quantum circuit
  - **GPU-accelerated** tensor processing with batch mode support
  - **Dynamic computation graph** for easy debugging
  - Easy construction of **hybrid classical and quantum** neural networks
  - **Gate** level and **pulse** level simulation support
  - **Converters** to other frameworks such as IBM Qiskit
  - Examples of ML for Quantum
  - …

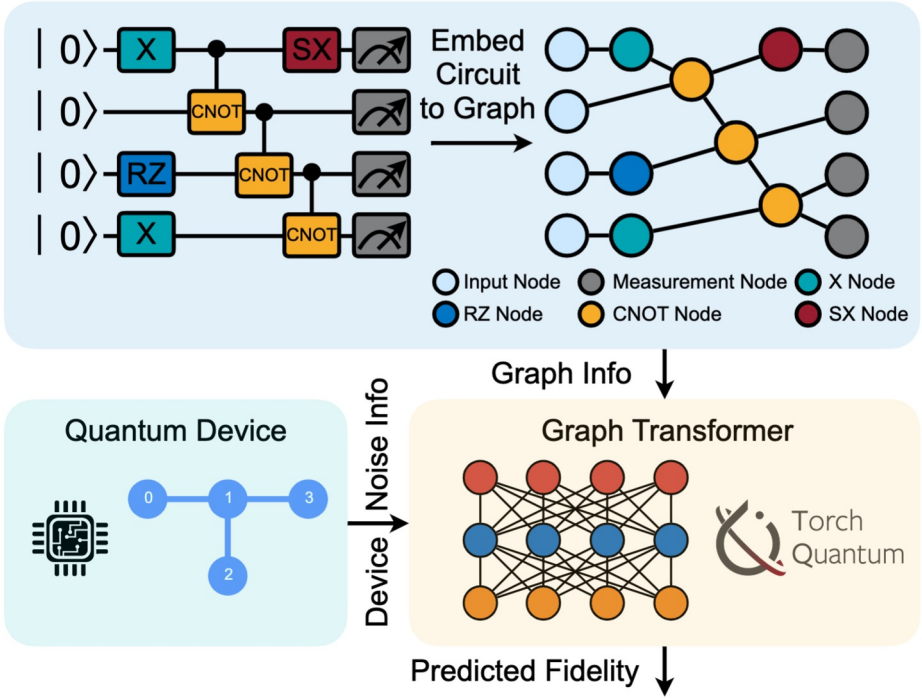
Torch Quantum

Materials at: https://torchquantum.org

MIT HAN LAB

Transformer for circuit reliability prediction

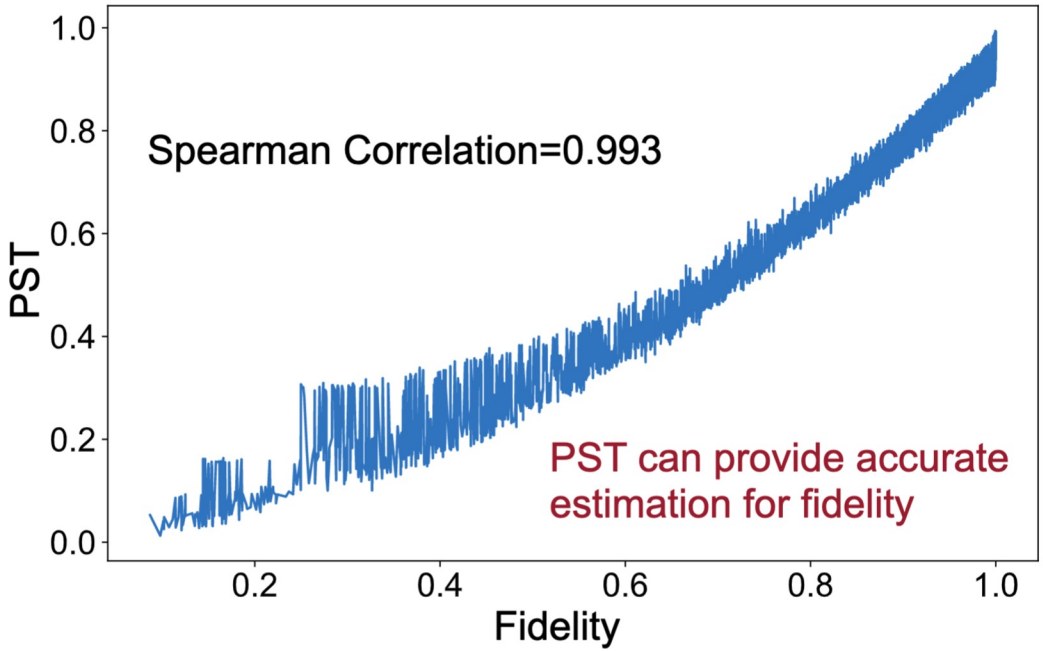
- Use the circuit graph information



Transformer for circuit reliability prediction

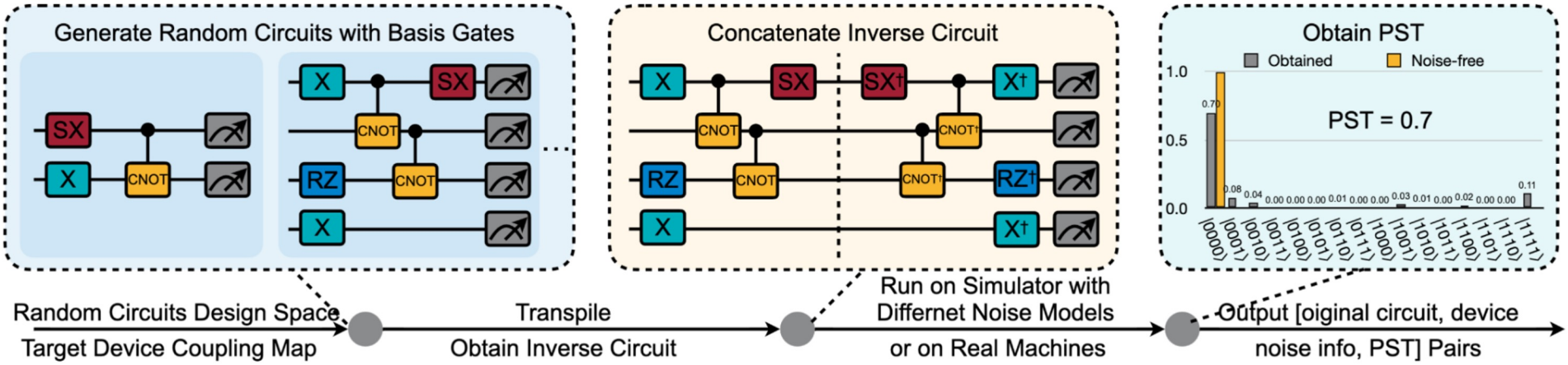
- Use PST as the metrics (same as in the previous Ji and Swamit's papers)

$$PST = \frac{\#Trials\ with\ output\ same\ as\ initial\ state}{\#Total\ trials}$$



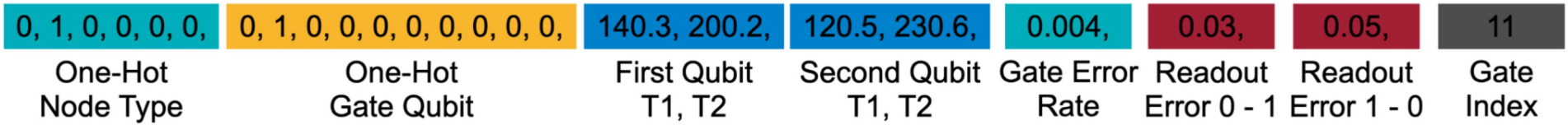
Dataset collection

- Collect dataset on real machine / noisy simulator



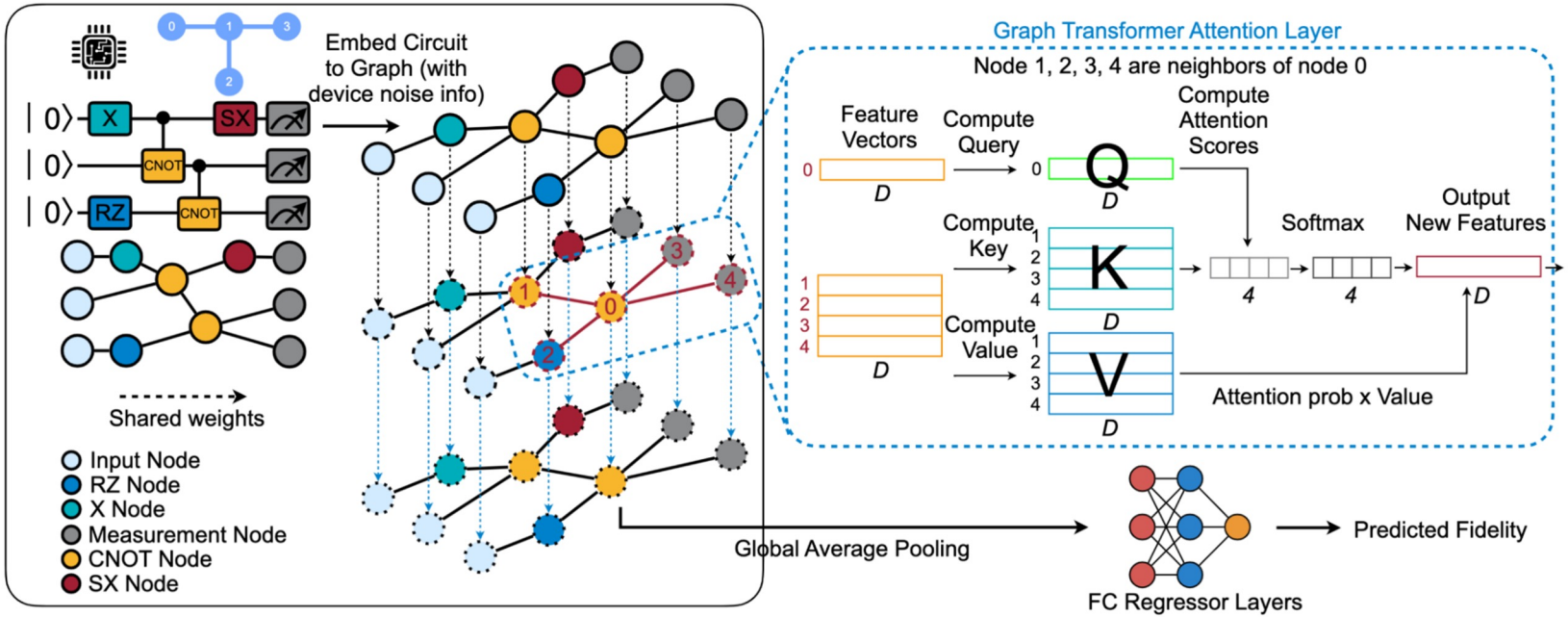
Features on each node

- Use the circuit graph information



Graph Transformer

- Graph transformer layers



Preliminary Results

- Randomly generated circuits