1. QuantumNAS[HPCA'22]

[Make the architecture of parameterized quantum circuit robust to noise]

Quantum computing development is fast.
Quantum computing (QC) is witnessing fast development recent years. We can now access to quantum machines with tens or hundreds of qubits.

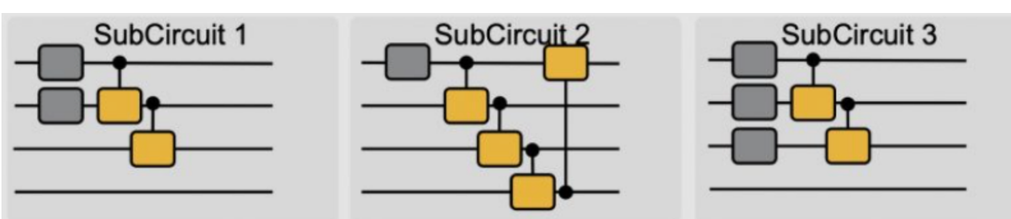
Challenge of NISQ stage and why parameterized quantum circuits (PQC) is promising.
However, we are still expected to be in Noisy Intermediate-Scale Quantum (NISQ) era for multiple years. In this stage, the QC bottlenecks are:

1. The noise of quantum machines are very large (10^-3 to 10^-2 level)
2. The available qubits are not enough to run those famous algorithms such as Shor's algorithm for big number factorization.

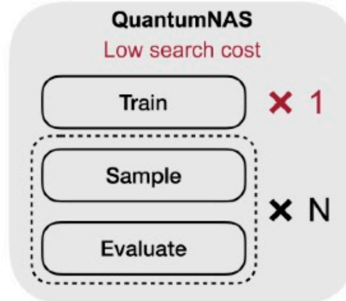
Using parameterized quantum circuits in a hybrid classical-quantum model is more expected to achieve quantum advantage in the near term. Example popular applications are Variational Quantum Eigensolver (VQE) for chemistry simulation, Quantum Approximate Optimization Algorithm (QAOS) for combinatorial optimization problems and Quantum Neural Networks (QNN) for machine learning.

What are parameterized quantum circuits (PQC):
Just like classical, quantum circuits contain many gates (operation), some of them are fixed, such as NOT gate, some are adjustable, such as rotation gates. The typical pipeline to use PQC is: firstly design the circuit architecture such as numbers/types of gates, the position of each gate in the circuits. Then train the parameters in the PQC towards certain goal.

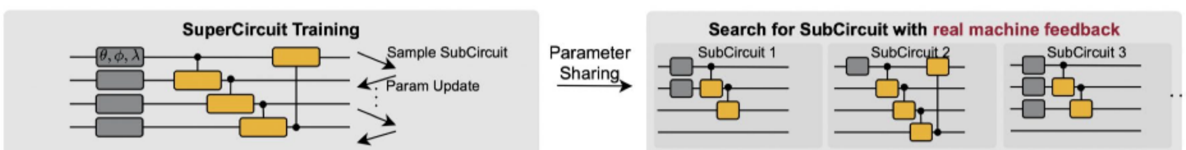
Challenge of PQC.
For PQC, the design of quantum circuit architecture is flexible, different PQC circuits (such as three different circuits below) can all be train to solve the problem. Then the challenge is how to design a good circuit architecture in such a large design space? One way is **naive search,** in which, in one iteration, we can sample many candidates in the design space, then train parameters of each of them, then compare and enter the next iteration of search. This method is very **expensive** because we need to pay them cost of training each candidates.



QuantumNAS is a framework to solve the challenge: efficiently search for PQC architecture.
QuantumNAS **decouples** the parameter training and circuit architecture search as shown below so that we only need to train once but can use it for many times during search. It leverages a **SuperCircuit**-based method.



The QuantumNAS pipeline
Step 1 : Train SuperCircuit.
The SuperCircuit is the **largest** candidate circuit in the design space, so that all the other candidates (called **SubCircuits**) are subsets of the SuperCircuit. We train it by iteratively sampling and updating SubCircuits. The important point is that all SubCircuits **share** the parameters of the common parts. We therefore only train once of the parameters, when we need to evaluate a SubCircuit candidate in the design space, we only need to **inherit** the corresponding parameters from the SuperCircuit and do evaluation.



Step 2: Search for SubCircuit with targeted objective, such as high robustness.
With the SuperCircuit, search will be easy and efficient. For example, the objective could be the accuracy under quantum noise impact. In one iteration, we sample SubCircuit architecture and inherit parameters from SuperCircuit. Then we can evaluate them on target real quantum machine to get the performance under the noise impact. Then select the SubCircuits with high performance and then enter next iteration.

Step 3: Prune away small parameters.
After step 2, we get a searched SubCircuit, then we can train its parameter from scratch and then perform iterative pruning and fine-tuning to remove small parameters. Therefore the number of gates and noise sources are further reduced.

Step 4: Run the search SubCircuit on real machine and obtain final performance.

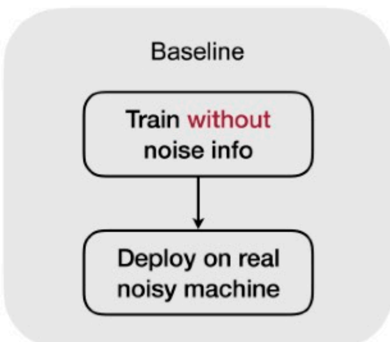
Results on real IBM quantum computers:
1. On quantum neural network task, QuantumNAS achieves 85% 4-class, and 32% 10-class classification accuracy, which is over 35% and 20% higher than human design baseline and noise-unaware search.
2. On chemistry simulation, it achieves 95% accuracy for H2 ground state energy simulation, compared to 84% accuracy of baseline methods.

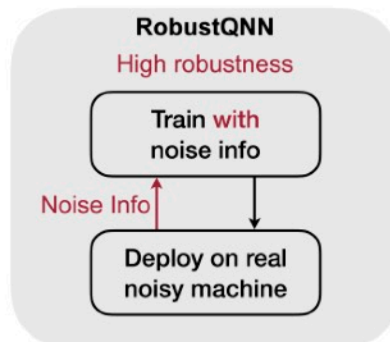
2. TorchQuantum: library for Quantum ML and ML for Quantum.

1. Parameterized quantum circuit construction, search, train and deployment on real quantum machine, including QNN, VQE.
2. Fast quantum simulation on GPU/CPU
3. Examples of using ML models to optimize quantum computer system.
4. Provide extensive tutorials and examples to help the users understand quantum ml system.

3. RobustQNN [DAC'22]

[Make the parameters of parameterized quantum circuit robust to noise]





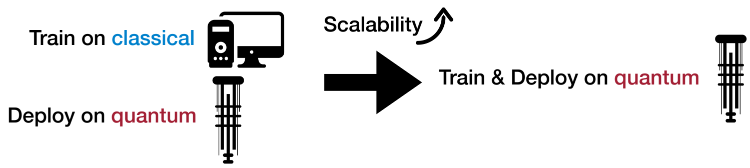
In QuantumNAS, the **circuit architecture** is noise-robustness. In robustQNN, we further propose to make the **circuit parameters** to be robustness to noise.

1. We propose to **inject noise** during the training process so that the parameters will be familiar with the noise that exists in all real machines.
2. We also propose quantization of measurement results, which has a denoting effect on the results, increasing robustness.

4. On-chip QNN [DAC’22]

[Make the training process of parameterized quantum circuit scalable]

Quantum Neural Networks (QNN) is a promising candidate to achieve **quantum advantage** in machine learning. To achieve that, the QNN need to be powerful enough and, by definition of quantum advantage, cannot be efficiently simulated on classical machine.



Therefore, we propose to use quantum computers to train quantum neural networks so it will have high scalability. We are the first experimental demonstration of using a technique called **parameter shift** on quantum machine to train QNN and achieve comparable accuracy to noise-free simulation.