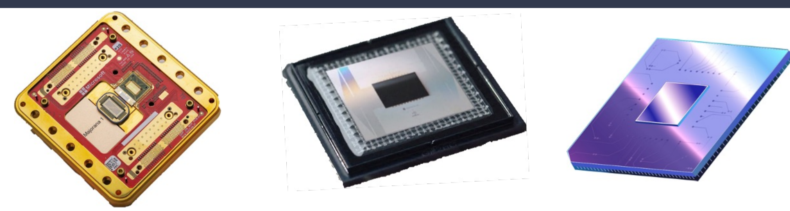
QUANTUMX: FEW-SHOT LEARNING FOR QUANTUM-HPC PERFORMANCE MODELING

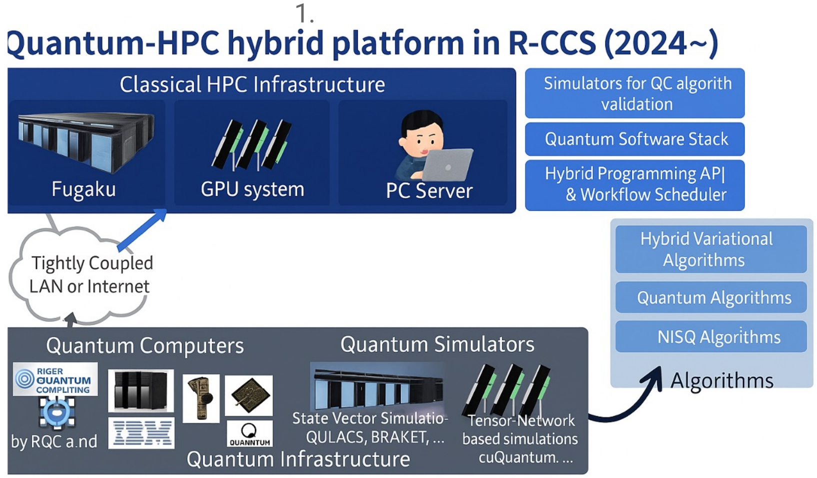
Arunavo Dey[1], Dr. Jae-Seung Yeom[2], Dr. Tapasya Patki[2], Dr. Tanzima Islam[1]
Texas State University[1]
Lawrence Livermore National Laboratory[2]

The work is performed under the auspices of the U.S Department of Energy by Lawrence Livermore National Laboratory under the Contract DE-AC52-07NA27344

The era of NISQ Devices



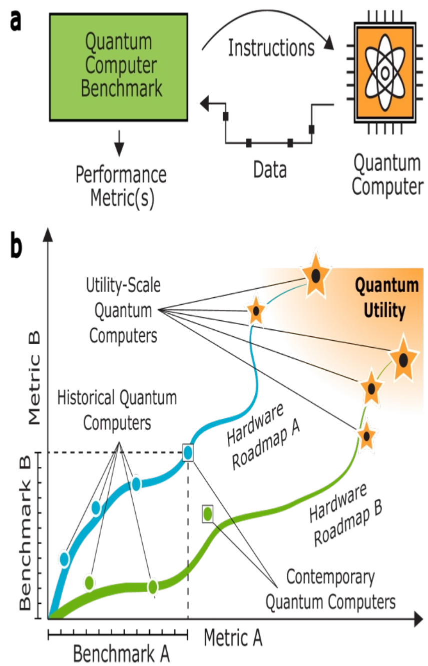
1.



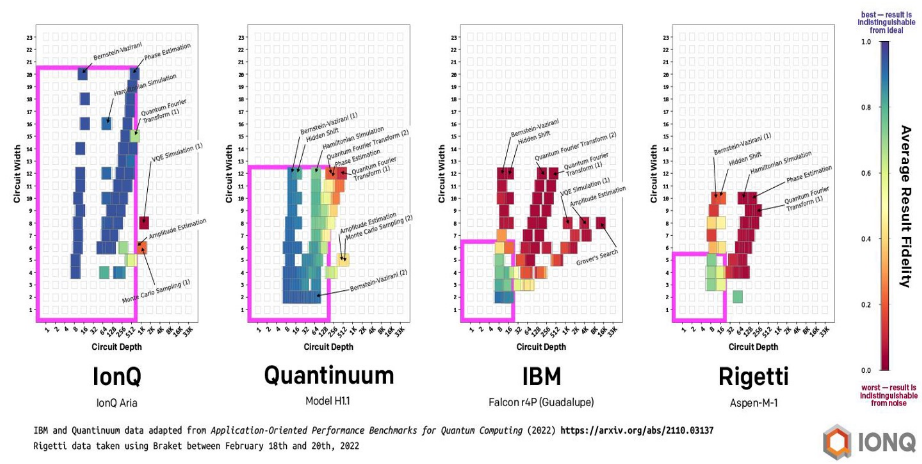
2

(2) https://www.hpcwire.com/2024/05/22/isc-2024-a-few-quantum-gems-and-slides-from-a-packed-qc-agenda/

Why it is important to compare QPU's



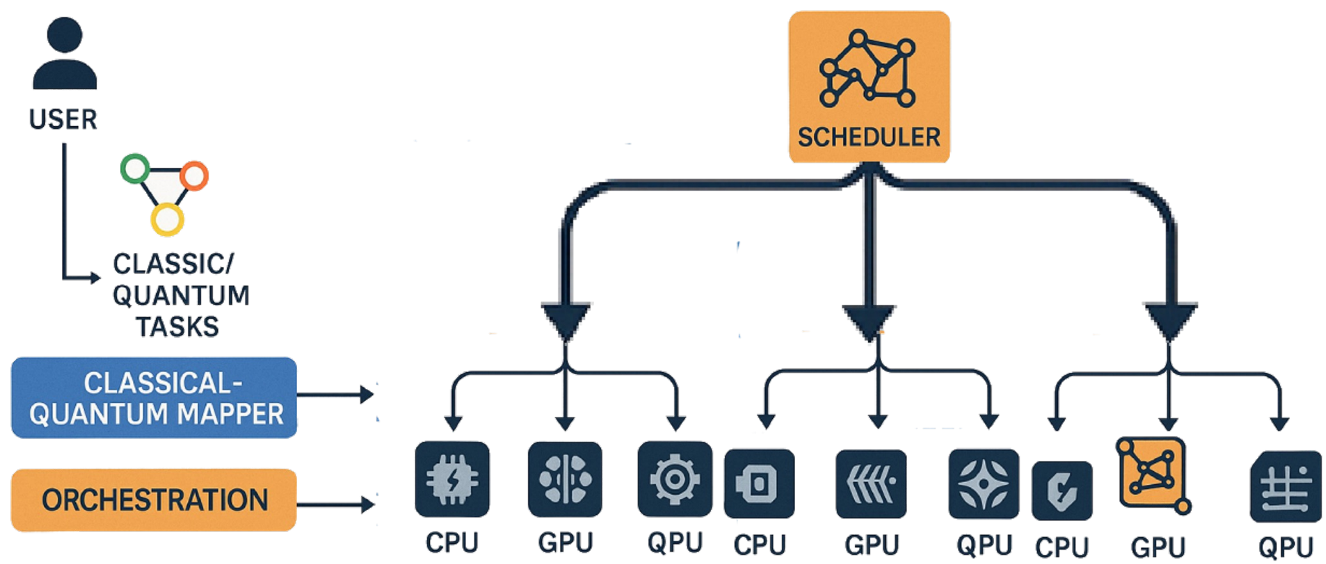
1. Total cost of ownership includes ecosystem, reliability, and support. Procurement decisions must balance NISQ limitations with roadmap alignment.



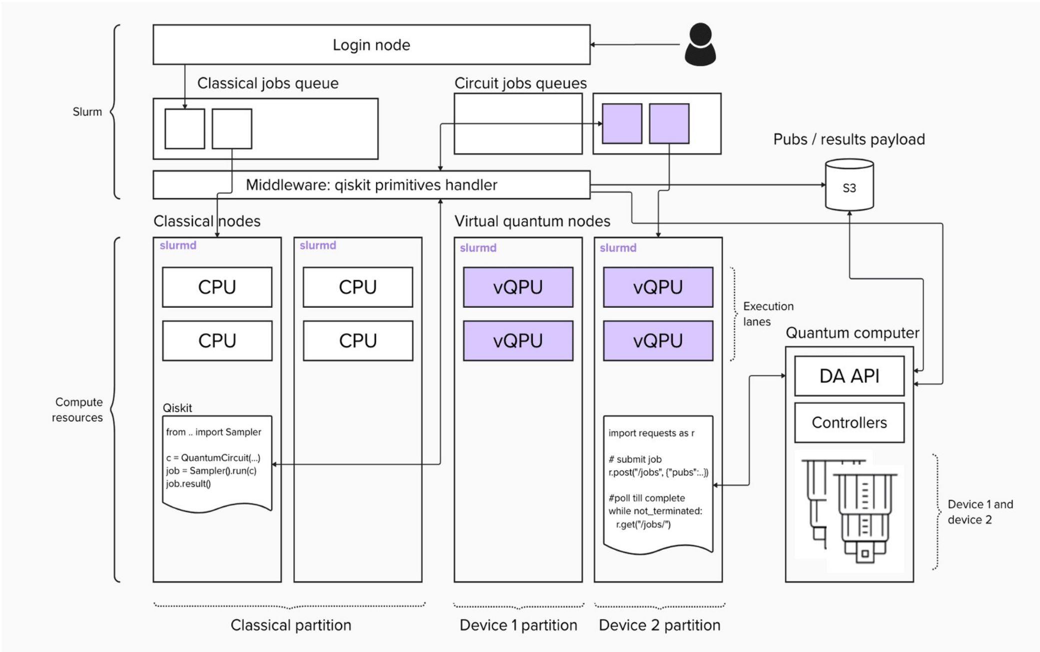
2.Performance depends on more than qubit count—connectivity, fidelity, and error rates matter more. Performance prediction requires cross-QPU generalization models. Hybrid predictors must incorporate classical preprocessing and quantum circuit partitioning

Source: 1. https://www.hpcwire.com/2024/05/22/isc-2024-a-few-quantum-gems-and-slides-from-a-packed-qc-agenda/ 2. https://quantumtech.blog/tag/ibm/

Why it is important to compare QPU's

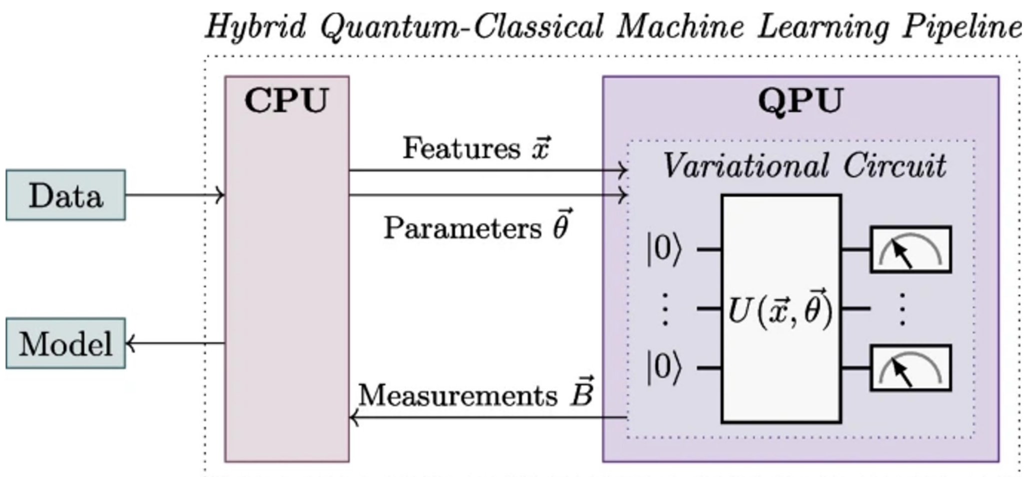


Quantum and HPC



Source: https://www.ibm.com/quantum/blog/supercomputing-24

Quantum in ML



Source: https://link.springer.com/article/10.1007/s42484-025-00254-8

Job Scheduling Challenges in NISQ Devices

1. **First**, the *incompatibility of data representations* across different qubit modalities
2. **Second**, *hardware-specific noise profiles and error channels*
3. **Third**, the *lack of standardized performance metrics*
4. **Fourth**, *variability in gate sets, native operations, and compilation paths*

An approach to solve: QuantumX

- Input alignment-based test-time adaptation with few-shot learning

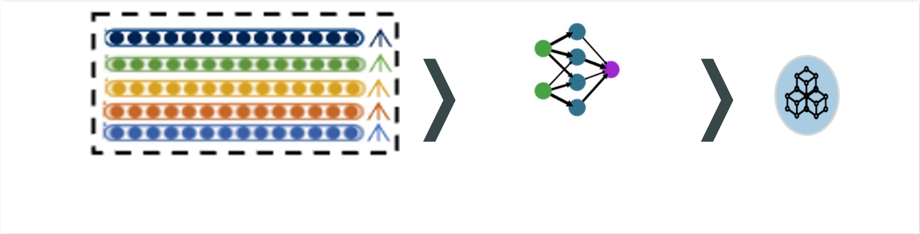
Time-efficient

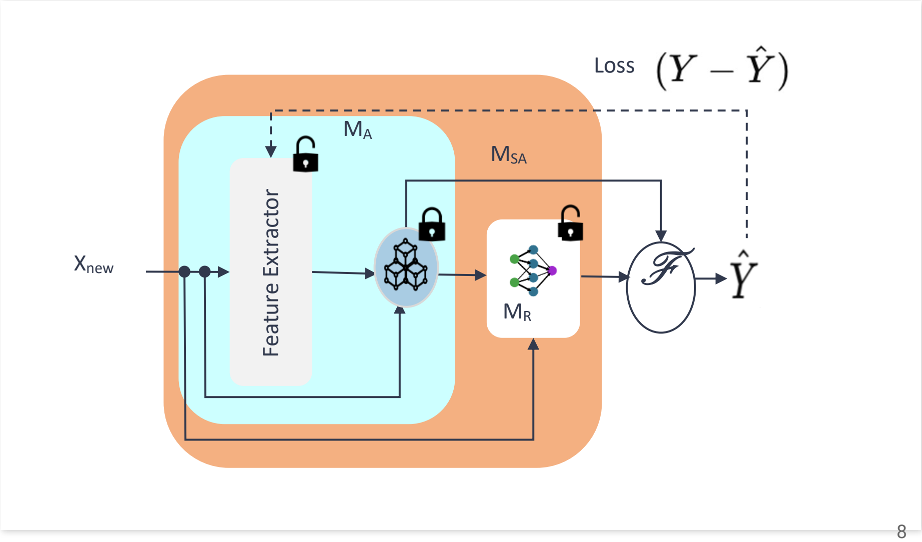
- Build a source model and fine-tune with new test-time samples (test-time adaptation)

Data-efficient

- Adaptation can be done using a few samples
  - Evaluations show promising results with even 1% of samples (=1 sample)

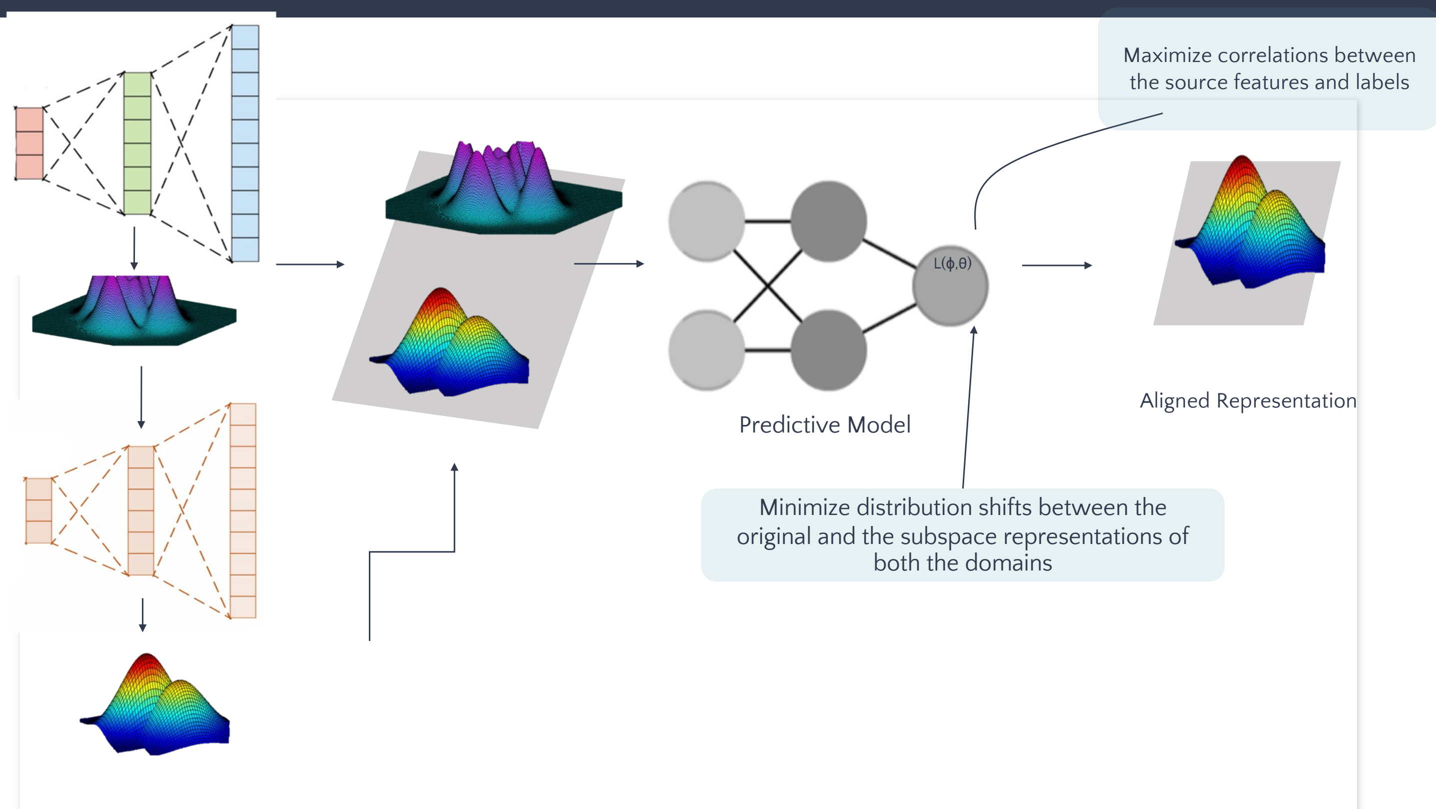
A framework with a Model with input Alignment and Residual Learning





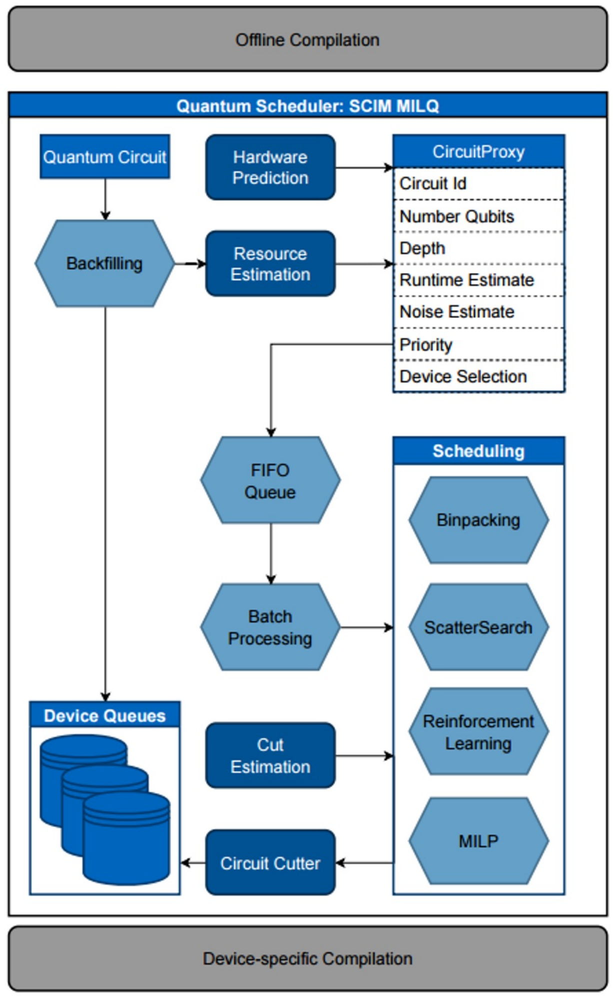
Source: *ModelX*: A Novel Transfer Learning Approach Across Heterogeneous Datasets

Key Capabilities: Input Alignment



KEY CAPABILITIES

- Scheuler Integration



KEY CAPABILITIES

Few shot adaptation and Residual Modeling

Scenerio

- Lets talk about a scenario where we want to transfer knowledge across to QPU'S A and B
  - Algorithm used: VQE
  - Qubits: Superconducting Qubits
  - QPU-A: IBM Kyiv
  - QPU-B: Aspen Rigetti

What features to use if want to build predictive model for runtime prediction?
From where and how do we collect the features?
How the knowledge is used?

For Runtime

Domain Invariant Features

num_Qubits, depth, width, num_1q_gates, num_2q_gates, avg_gate_density, entanglement_connectivity, num_measurements

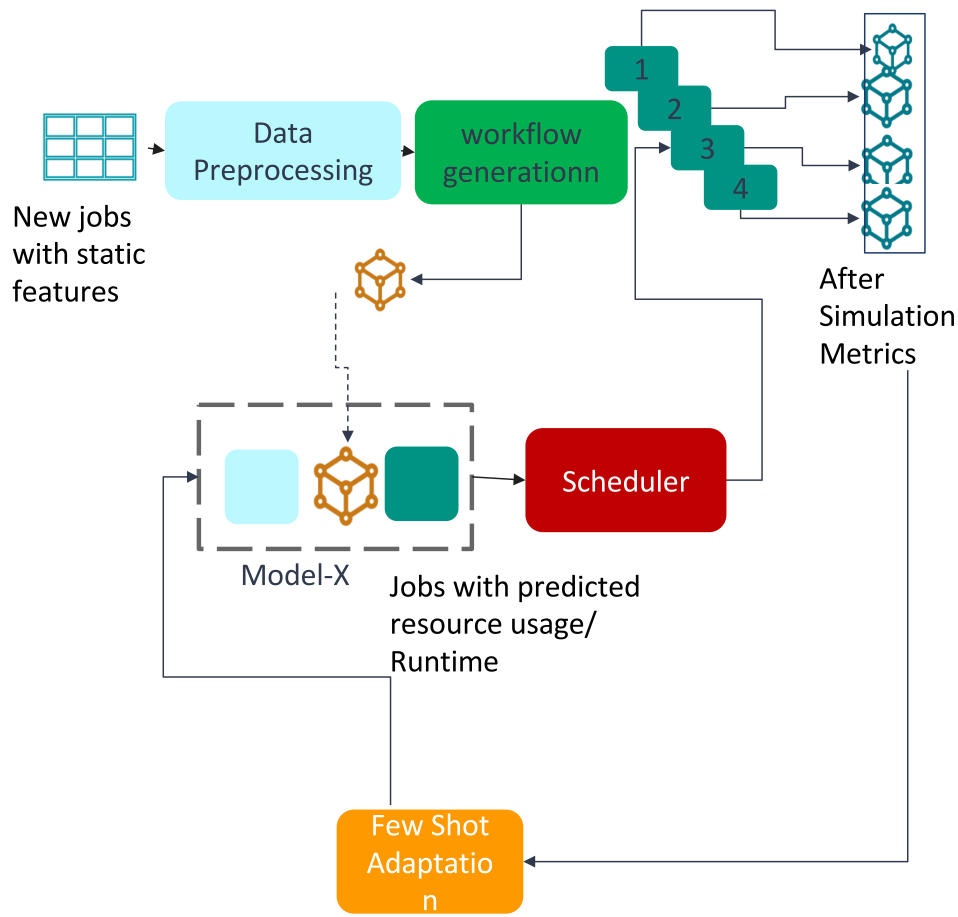
H/w Specific Features

native get set, avg_1q_gate_time, avg_2q_gate_time, readout duration, coherence time avg, error_1q_avg, error_2q_avg, layout_density

cx_directionality, qubit_frequency, gate_schedule_duration, thermal_execution_time, delay_instruction_granularity, pulse_alignment_constraints,

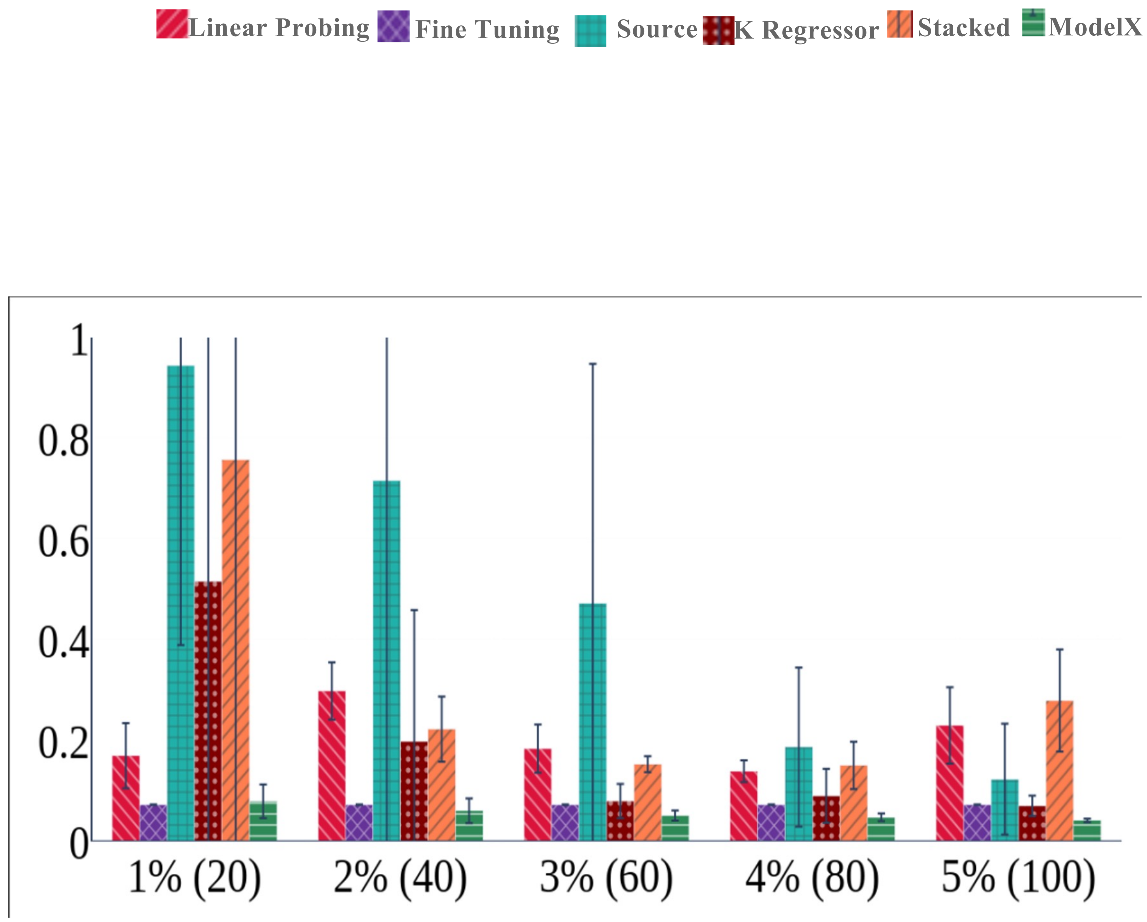
Qubit tuning Scheudle, ramp schedule, Qpu noise floor estimate, quil routing strategy, qubit_tile_mapping, acquire_to_measurement_delay, entanglement connectivity, 2q_gate_pairs, parallelism score, swap estimate, pairwise frequency collision, active neighboirs per gate

Proposed Framework

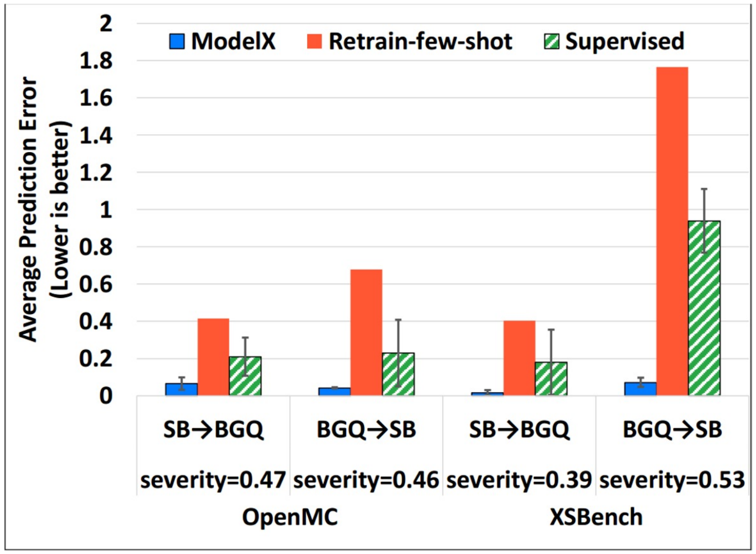


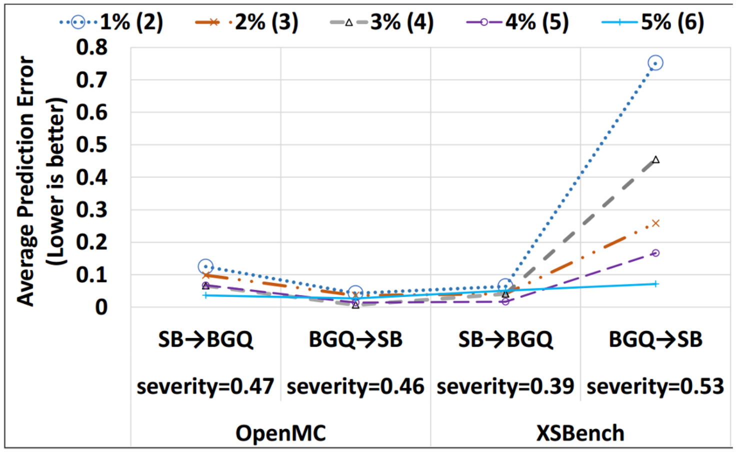
Initial Results on Cross platform prediction

Pascal TO Turing

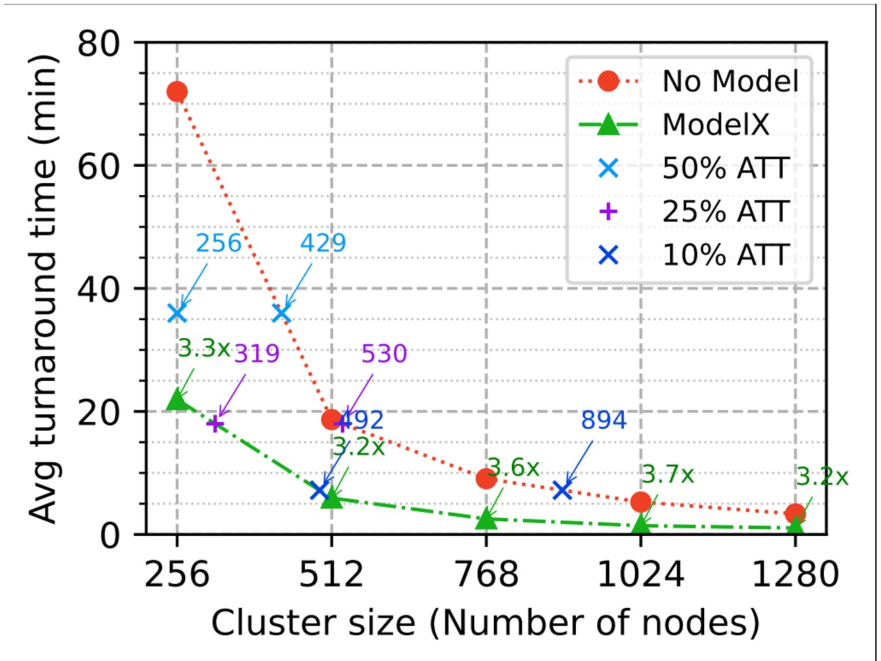


Initial Results on Cross platform prediction





Initial Results on Cross platform prediction



LIMITATIONS

🚀 The Challenge
- Hybrid quantum-classical HPC systems are complex and noisy.
- Task performance is affected by:
  • Quantum-side: Crosstalk (e.g. ZZ interactions), gate errors.
  • Classical-side: GPU stalls, CPU delays.
- Existing ML models need large data, frequent retraining, and struggle with portability.

KEY CAPABILITIES

Predictive Resource and Crosstalk Modeling:

Can QuantumX Do that?
Is Total Error Prediction a Better Idea?
How the existing error detection models can be benefitted using the framework?