ECE-382V: Introduction to Quantum Computing Systems
Homework-2: Error mitigation and NISQ
Due Date: 26th October (4.59 PM Central Time). No late submissions will be accepted.

Problem-1: Gate Error Mitigation

(a) What are the benefits of instruction reordering in quantum programs?
(b) Compilers perform gate cancellations in two passes, before and after routing. Why?
(c) Gate nativization is a crucial step in program compilation. What are the trade-offs in nativization on systems that offer multiple native gates to decompose a high-level instruction? What are the drawbacks of the gate selection method proposed in the paper "The Imitation Game"?
(d) Circuit cutting is an emerging error mitigation technique in NISQ-era. What are the overheads of circuit cutting? How can these overheads be reduced?

Problem-2: Measurement Error Mitigation

(a) Why are measurement errors hard to eliminate at the device-level?
(b) What are the drawbacks of the matrix-based measurement error mitigation techniques?
(c) What are the challenges in state-transformation-based measurement error mitigation? How to overcome these challenges?
(d) JigSaw is a measurement error mitigation technique. It uses a recompilation step to measure the program qubits on physical qubits with the lowest measurement errors. How can you reduce this recompilation overhead?
(e) What are the trade-offs in designing measurement classifiers for quantum systems?

Problem-3: Idle/Crosstalk Error Mitigation

(a) Why are idle errors important to eliminate at the application-level?
(b) What are the trade-offs involved in concurrent CNOT scheduling?
(c) What are the trade-offs in incorporating device-level techniques for reducing idle errors naively at the application-level?
(d) Is characterization the right approach to reduce crosstalk/idle errors? Why or why not?

Problem-4: Variational Quantum Algorithms

(a) What are the potential benefits of variational quantum algorithms? What are some of their key drawbacks?
(b) How can you use application specific properties to reduce the impact of hardware errors in variation quantum algorithms?
(c) Google uses three types of graphs to study the performance of QAOA on real systems. What is the specific reason to choose the exact kinds of graphs chosen for this study?
(d) IBM recently introduced the Qiskit Runtime environment to run variational quantum algorithms. What are the benefits of this approach? What are the drawbacks?