The Complexity of NISQ

Sitan Chen*
UC Berkeley

Jordan Cotler†
Harvard University

Hsin-Yuan Huang‡
Caltech

Jerry Li§
Microsoft Research

October 14, 2022

* sitanc@berkeley.edu
† jcotler@fas.harvard.edu
‡ hsinyuan@caltech.edu
§ jerrl@microsoft.com

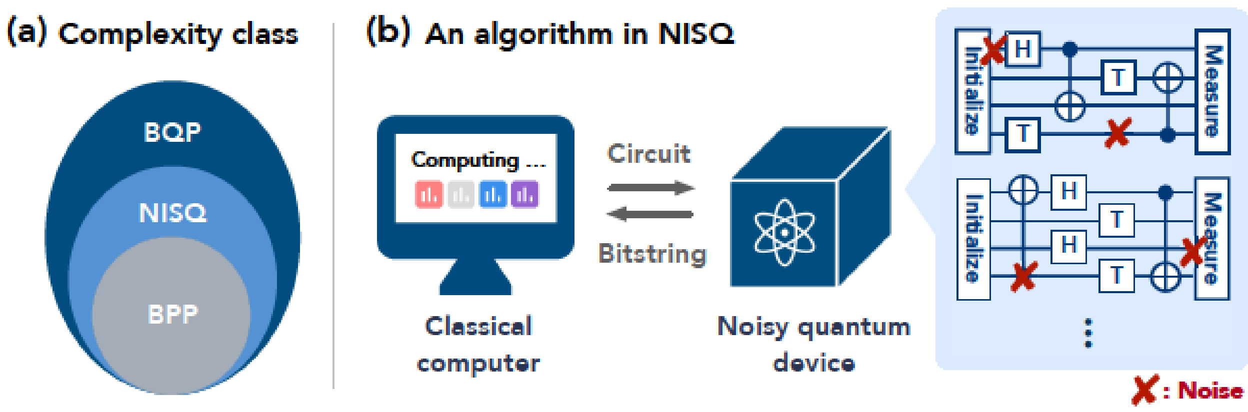

Figure 1: *Illustration of the* NISQ *complexity class:* (a) Complexity classes: NISQ contains problems that can be solved by classical computation (BPP), and some problems that can be solved by quantum computation (BQP). (b) NISQ algorithm: An algorithm in the complexity class NISQ is modeled by a hybrid quantum-classical algorithm, where a classical computer can specify the circuit to run on a noisy quantum device and the device would run a noisy version of the circuit and return a random classical bitstring obtained from noisy measurement.

all qubits simultaneously. From a physical perspective, this constraint arises due to the difficulty of isolating subsets of qubits and measuring them without decohering the residual qubits.

Finally, we consider a classical computer that can repeatedly run the noisy quantum device and analyze the output from the noisy quantum device.

These constraints are chosen to encapsulate the gap between the physical limitations of what we can achieve with existing quantum computers, and general quantum computation. We note these considerations preclude the implementation of all known general fault-tolerant quantum computation schemes [53, 54, 55, 56, 57, 58, 59], but that removing any one of these constraints would already allow for some form of nontrivial quantum fault tolerance [53, 57, 58]. The obstruction to fault tolerance can be understood intuitively. The noisy quantum gates cause all qubits to accrue entropy, which cannot be pumped out until the measurement at the end. Since too much entropy would destroy all useful quantum correlations, it is not possible for the noisy quantum devices under the above constraints to perform an arbitrarily long quantum computation.

Motivated by the above considerations, in Section 2.1 we formally define the NISQ complexity class to be the set of all problems that can be efficiently solved by a classical computer with access to a noisy quantum device that can (i) prepare a noisy poly(n)-qubit all-zero state, (ii) execute noisy quantum gates, and (iii) perform a noisy measurement on all of the poly(n) qubits.

2 Main Results

In Section 2.1 we give an overview of the definition of NISQ. Then, in Section 2.2, we give two modifications of Simon's problem which respectively yield a super-polynomial separation between BPP and NISQ, and an exponential separation between NISQ and BQP. In Section 2.3, we study the NISQ complexity of three well-known problems: unstructured search, Bernstein-Vazirani problem, and shadow tomography. We defer all technical details to the appendix.

$\mathcal{O}$ $\overline{}$

$\mathcal{O}$

$\mathcal{O}$

—

$\mathcal{T}$ $\mathcal{A}$

$\mathcal{T}$

- $\mathcal{T}$

-

-

$\mathcal{A}$

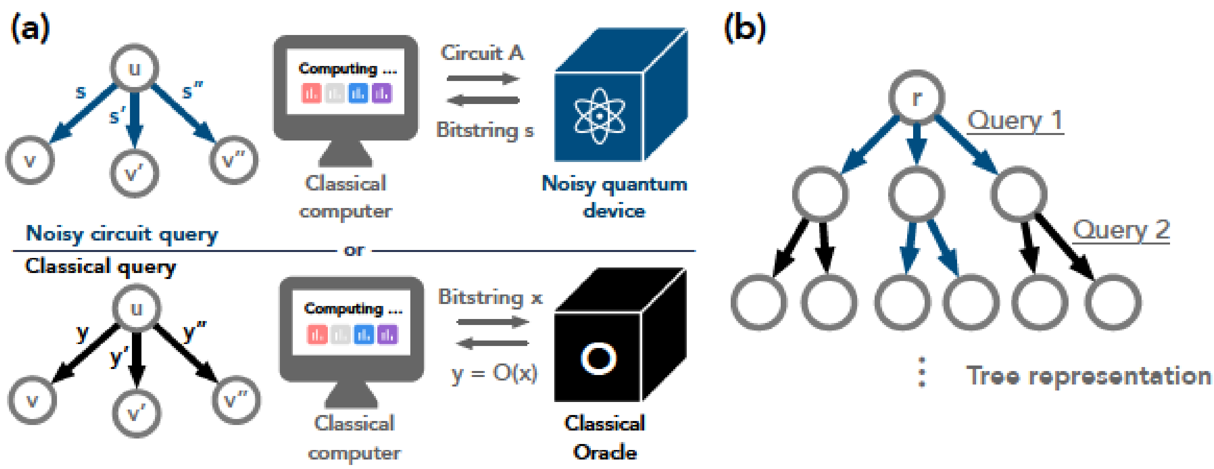

Figure 2: Illustration of the tree representation for NISQ algorithms. (a) At every memory state u of the classical computer/algorithm, it could either make a noisy circuit query or a classical query. (b) The tree representation with a mix of noisy circuit queries and classical queries.

The following lemma shows that slight perturbations to the distributions over children for each node do not change the overall distribution over leaves of $\mathcal{T}$ by too much.

Lemma B.2. *Given learning tree $\mathcal{T}$ corresponding to a NISQ_λ algorithm with query complexity N, suppose $\mathcal{T}'$ is a learning tree obtained from $\mathcal{T}$ as follows. For every node u at which a noisy quantum circuit A is run, replace A by another circuit A' such that the new induced distribution over children of u is at most ε-far from the original distribution in total variation. Then the distributions over leaves of $\mathcal{T}$ and $\mathcal{T}'$ are at most εN-far in total variation.*

Proof. Consider the sequence of trees $\mathcal{T}^{(i)}$ where $\mathcal{T}^{(0)} = \mathcal{T}$ and $\mathcal{T}^{(i)}$ is given by taking all u in layer i of $\mathcal{T}^{(i-1)}$ that run some noisy quantum circuit A and replacing them with the corresponding circuit A' from $\mathcal{T}'$. By design, $\mathcal{T}^{(N)} = \mathcal{T}'$. Let $p^{(i)}$ denote the distribution over leaves of $\mathcal{T}^{(i)}$. It suffices to show that $d_{\mathrm{TV}}(p^{(i)}, p^{(i-1)}) \le \varepsilon$.

Note that $p^{(i-1)}$ specifies some mixture over distributions p_v, where p_v is the distribution over leaves conditioned on reaching node v in the i-th layer. In particular, in this mixture, v is sampled by sampling parent node u by running the NISQ algorithm corresponding to $\mathcal{T}'$ for $i-1$ steps and then running the corresponding quantum circuit A from $\mathcal{T}$. In contrast, $p^{(i)}$ is a mixture over the same distributions p_v, but v is sampled by running the NISQ algorithm corresponding to $\mathcal{T}'$ for i steps and then running the corresponding quantum circuit A' from $\mathcal{T}'$. These two distributions over v are at most ε-far in total variation, so the two mixture distributions are also at most ε-far in total variation as claimed. □

Our lower bounds will be based on Le Cam's method– see Section 4.3 of [48] for an overview in the context of the tree formalism of Definition B.1. In every case we will reduce to some *distinguishing task* in which the algorithm must discern whether the oracle it has access to comes from one family of oracles or from another. For example, for unstructured search, the distinguishing task will be whether the oracle corresponds to some element in the search domain or whether the oracle is the identity channel.

More concretely, given two disjoint sets of oracles S_0, S_1, we will design distributions D_0, D_1 over S_0, S_1. Given any algorithm specified by some $(\mathcal{T}, \mathcal{A})$, we will upper bound the total variation

$\mathcal{T}$

$\mathcal{A}$

$\mathcal{T}$

$\mathcal{E}$ $\mathcal{E}$

$\mathcal{E}$ $\mathcal{E}$

$\mathcal{E}$ $\mathcal{E}$

$\mathcal{E}$ $\mathcal{E}$ $\mathcal{U}$ $\mathcal{U}$

$\mathcal{U}$ $\mathcal{E}$ $\mathcal{U}$ $\mathcal{E}$ $\mathcal{U}$

$\mathcal{U}$ $\mathcal{E}$ $\mathcal{U}$ $\mathcal{E}$ $\mathcal{U}$ $\mathcal{E}$ $\mathcal{U}$ $\mathcal{E}$ $\mathcal{U}$

$\mathcal{E}$ $\mathcal{E}$ $\mathcal{U}$ $\mathcal{E}$ $\mathcal{U}$ $\mathcal{E}$ $\mathcal{U}$ $\mathcal{E}$ $\mathcal{E}$

□

\

\

\ □

$\mathcal{O}$

$\mathcal{O}$

$\mathcal{G}$

$\mathcal{G}$

$\mathcal{G}$

$\mathcal{G}$

$\mathcal{O}$

$\mathcal{O}$

$\mathcal{A}$

$\mathcal{O}$ $\mathcal{D}$

$\mathcal{A}$ $\mathcal{D}$ $\mathcal{D}$

$\mathcal{O}$

$\mathcal{O}$

□

$\mathcal{O}$

$\mathcal{A}$

$\mathcal{D}$

$\mathcal{A}$

$\mathcal{D}$ $\mathcal{O}$

□

$\mathcal{H}$

$\mathcal{H}$ $\mathcal{H}$ $\mathcal{H}$ $\mathcal{H}$ $\mathcal{H}$

$\mathcal{H}$

$\mathcal{H}$

$\mathcal{H}$ $\mathcal{H}$ $\mathcal{H}$ $\mathcal{H}$

$\mathcal{H}$

$\mathcal{H}$

$\mathcal{H}$

$\mathcal{H}$

$\mathcal{H}$

$\mathcal{H}$ $\mathcal{H}$ □

$\mathcal{H}$ $\mathcal{H}$ $\mathcal{H}$ $\mathcal{H}$

$\mathcal{H}$ $\mathcal{H}$

$\mathcal{H}$ $\mathcal{H}$ $\mathcal{H}$ $\mathcal{H}$ $\mathcal{O}$

$\mathcal{H}$

$\mathcal{H}$ $\mathcal{H}$

$\mathcal{H}$

$\mathcal{H}$

$\mathcal{H}$

$\mathcal{H}$ $\mathcal{H}$ $\mathcal{H}$

$\mathcal{H}$

$\mathcal{H}$ $\mathcal{H}$ $\mathcal{H}$

$\mathcal{H}$

$\mathcal{H}$

$\mathcal{O}$

$\mathcal{D}$ $\mathcal{D}$

$\mathcal{D}$

$\mathcal{O}$

$\mathcal{E}$

$\mathcal{E}$

$\mathcal{O}$ $\mathcal{E}$

□

$-$

$-$

$-$

$\mathcal{E}$ $\mathcal{E}$

□

$\mathcal{T}$

$\mathcal{T}$

$\square$

$\mathcal{T}$ $\mathcal{A}$

$\mathcal{T}$

- $\mathcal{T}$

-

-

$\mathcal{A}$

-
-

□

$\mathcal{T}$

$\mathcal{T}$ $\mathcal{A}$

$-$

□

$\mathcal{I}$

$\mathcal{I}$

$\mathcal{T}$

$\mathcal{T}$

$-$ $\mathcal{O}$ $\overline{\quad}$

$\mathcal{T}$

$-$

$\mathcal{T}$

$\mathcal{O}$

$-$

$\mathcal{T}$ $\mathcal{T}$

$-$

$-$

$\overline{\qquad\qquad}$

$-\mathcal{I}$

$\mathcal{T}$

$-$ $\mathcal{O}$ $\overline{\quad}$

$-$

$\square$

$\mathcal{O}$ $\overline{\quad}$

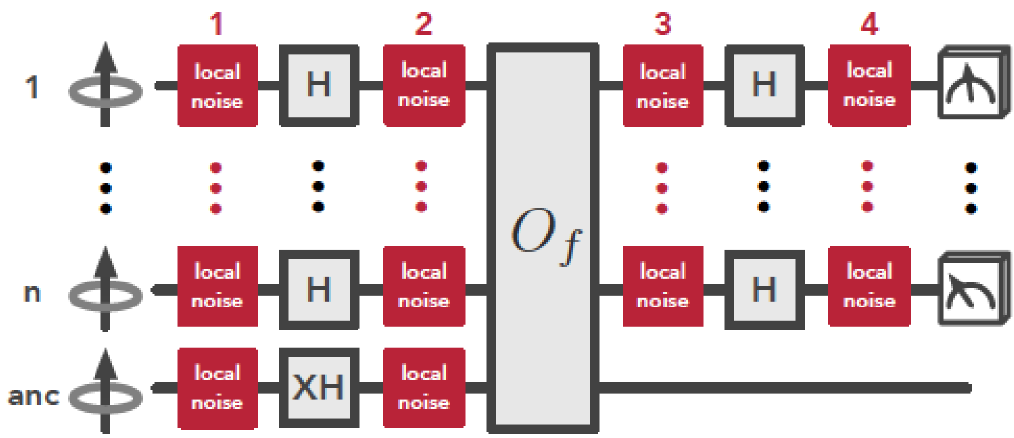

Figure 3: Bernstein-Vazirani algorithm in the presence of arbitrary noise (each box labeled by "local noise" denotes that with probability λ, an arbitrary, adversarially chosen single-qubit operation is applied). We have labeled the layers of noise for ease of reference in the proof.

Definition E.2 (Permutation operators). *For $n > 0$, let S_n be the permutation group on m objects. To each π in S_n we associate an operator acting on $(\mathbb{C}^2)^{\otimes n}$ defined by*

$$\pi\big(|\psi_1\rangle \otimes |\psi_2\rangle \otimes \cdots \otimes |\psi_n\rangle\big) = |\psi_{\pi^{-1}(1)}\rangle \otimes |\psi_{\pi^{-1}(2)}\rangle \otimes \cdots \otimes |\psi_{\pi^{-1}(n)}\rangle\,, \quad \forall\, |\psi_1\rangle, |\psi_2\rangle, ..., |\psi_n\rangle \in \mathbb{C}^2$$

which extends by multilinearity to all of $(\mathbb{C}^2)^{\otimes m}$.

We have a similar definition for permutations acting on bit strings.

Definition E.3 (Permutations acting on bit strings). *Again letting S_n be the permutation group on n objects, to each π in S_n we associate a function $\pi : \{0,1\}^n \to \{0,1\}^n$ defined by*

$$\pi(s_1 s_2 \cdots s_m) = s_{\pi^{-1}(1)} s_{\pi^{-1}(2)} \cdots s_{\pi^{-1}(m)}\,, \quad \forall s_1 s_2 \cdots s_m \in \{0,1\}^n\,.$$

Moreover, if $f : \{0,1\}^n \to \{0,1\}$ is the unknown function in the Bernstein-Vazirani problem, then we define $f_\pi := f \circ \pi$.

Bernstein-Vazirani algorithm. We conclude this subsection by reviewing how the original Bernstein-Vazirani algorithm [92] works, see Figure 3. One begins by preparing the initial state $|+\rangle^{\otimes n} \otimes |-\rangle$, and then acting on it with the oracle. In so doing, we obtain the state

$$\frac{1}{2^{n/2}} \sum_{x \in \{0,1\}^n} (-1)^{s \cdot x} |x\rangle \otimes |-\rangle\,.$$

Tracing out the $|-\rangle$ qubit, we can then apply the Hadamards $H^{\otimes n}$ to the first n qubits to obtain

$$\frac{1}{2^n} \sum_{x,y \in \{0,1\}^n} (-1)^{(s+y)\cdot x} |y\rangle = |s\rangle$$

which gives us the hidden string s.

$\mathcal{O}$

—

$\mathcal{O}$

$\mathcal{O}$

$\mathcal{E}$

□

□

$\mathcal{D}$

$\mathcal{F}$

$\mathcal{F}$

$\mathcal{F}$ $\mathcal{F}$

$\mathcal{F}$ $\mathcal{D}$

□

$\mathcal{F}$ $_{\mathcal{F}}$
$\mathcal{F}$ $\mathcal{F}$ $\mathcal{F}$
$\mathcal{F}$ $\mathcal{D}$

$\mathcal{F}$
$_{\mathcal{F}}$
$\mathcal{F}$

$\mathcal{F}$

$_{\mathcal{F}}$

$_{\mathcal{F}}$ $_{\mathcal{D}}$

$\mathcal{F}$ $\mathcal{F}$

$\mathcal{F}$ $\mathcal{F}$

$\mathcal{F}$ $\mathcal{D}$

$\mathcal{F}$

$\mathcal{F}$ $\mathcal{F}$

$\mathcal{F}$ $\mathcal{F}$

$\mathcal{F}$ _

$\mathcal{F}$ $\mathcal{D}$

$\mathcal{F}$ $\mathcal{D}$

$\mathcal{F}$

$\mathcal{F}$ $\mathcal{D}$

$\mathcal{F}$

$\mathcal{E}$

□

$\mathcal{T}$

$\mathcal{T}$

$\mathcal{T}$ □

$\mathcal{O}$

$\mathcal{O}$

$\mathcal{O}$

$\mathcal{O}$

$\mathcal{O}$