

## How Far are We from “Quantum Supremacy?”

Instructor: Xiao Liang

Table 1: Different Types of Quantum Supremacy

Type	Theoretically	Practically
Chemistry Simulation	yes	Progressing, no advantage yet
Optimization	unclear	Classical still better
Machine learning	unclear	Classical still better
Quantum Supremacy Experiments (RCS)	yes with warns	Demonstrated, but not useful
Quantum supremacy Experiments (Boson sampling)	yes	Demonstrated, but not useful
Cryptographic Classical Proof of Quantumness	yes	out of the reach of today's quantum computer

# 1 Quantum Computing for Quantum Chemistry

## 1.1 Chemistry is governed by Quantum Mechanics

### Main Takeaway:

*Chemistry is governed by electrons, and electrons are governed by quantum Mechanics!*

To see why Chemistry is governed by electrons, see the following example:

#### Protocol 1: Why Do Two Hydrogen Atoms Form an H<sub>2</sub> Molecule?

##### Key Idea: Atoms Want to Be Stable

A single hydrogen atom consists of:

- 1 proton in the nucleus
- 1 electron in the 1s orbital

To achieve stability, hydrogen “wants” to have 2 electrons. This is known as the **duet rule**. We now just believe it, and don't question it further

When two hydrogen atoms approach each other:

- Each atom has 1 unpaired electron
- They can **share** their electrons to form a **covalent bond**
- Each H atom now feels like it has 2 electrons (its own + the shared one)

This bond **lowers the total energy** of the system, which is why it forms **spontaneously** in

nature when two hydrogen atoms come close.

### Why Is $H_2$ Lower in Energy Than Two H Atoms?

- When the two H atoms share electrons, **electrostatic forces** (the attraction between nuclei and shared electrons) pull the atoms together
- This **reduces potential energy**
- The system becomes **more stable** than two separate atoms

## 1.2 Why and How to perform chemistry simulation?

- Drug discovery: Simulating proteins, enzymes, or drug-receptor interactions
- Materials science: Designing catalysts, superconductors, batteries

### Classical methods do work — but with limits.

Classical computers can simulate molecules using approximation methods such as:

- Hartree-Fock (HF)
- Density Functional Theory (DFT)
- Coupled Cluster (e.g., CCSD(T))
- Full Configuration Interaction (FCI) (only for very small systems)

These methods are highly effective for many chemical systems, but they scale poorly with system size and electron correlation complexity.

### The Core Problem: Exponential Scaling

Chemistry process is governed by electrons. Electrons obey quantum mechanics. So, if you want to simulate the chemistry process, you're basically performing computation following Schrödinger equation.

Examples:

- A modest molecule like caffeine has 24 electrons in 100 orbitals.
- The number of electron configurations can exceed  $10^{20}$ .
- Storing or processing that on a classical computer becomes infeasible.

### Why we have not achieved quantum supremacy in this area

- Current quantum devices are noisy and have:
  - Limited qubit counts (less than 50 logical qubits)
  - Short coherence times (quantum states decay quickly, in around 0.2 milliseconds)
  - Gate errors and measurement noise
- Benchmarks are Still Small: So far, quantum simulations have only been demonstrated for:  $H_2$ ,  $H_2O$ , etc. Mainly toy examples.  
Few electrons, few orbitals (e.g.  $< 10$ )  
These are trivial problems for classical computers, so quantum doesn't shine yet.
- Classical algorithms are highly optimized and perform very well for problems of moderate size.

## 2 Quantum Optimization (and Quantum Machine Learning)

Related topics including quantum approximate optimization algorithm (QAOA), and variational algorithm. But I do not have time to talk about them.

### 2.1 Adiabatic Quantum Optimization

Steps:

- Take your favorite NP complete problem, convert it to a “final” Hamiltonian  $H_f$  such that the ground state  $j_f$  tells you the solution to your NP problem.
- Prepare a “initial” Hamiltonian  $H_0$  such that you know its ground state  $j_0$ .
- Let the initial ground state and let it evolve according to the Schrödinger equation

$$\partial_t j_t = -iH(t)j_t;$$

where  $H(t) = (1 - t/T)H_0 + (t/T)H_f$

- **Adiabatic Theorem:** by taking the “step-size”  $t/T$  small enough, the procedure will evolve slow enough so that the state, through its transformation, remains the ground state
- Eventually,  $j_0$  will evolve into the ground state  $j_f$  of  $H(T) = H_f$ .

**Potential Issues:** How slow should the adiabatic process be? There is a trade-off between speed and performance.

- Adiabatic optimization gives quadratic speedup for search, but exponential time in general (in black-box?) [vMV01]. It essentially recover Simon’s algorithm without running it explicitly. All it does is to do adiabatic procedure.
- Exponential time for NP-complete problems, but can tunnel through local optima in certain special circumstances [Rei04].
- Anderson localization based arguments that it **typically** gets stuck in local optima [AKR10].

**The short story is:**

- quantum optimization algorithms cannot solve NP-complete problems.
- The best we can hope for is: they performs well on specific optimization problems in an approximate, heuristic sense. However, even for that, they can now apply only to “toy examples.” They didn’t beat classical algorithms yet.

## 3 “Quantum Supremacy” Experiments

### 3.1 Random Circuit Sampling

**Steps of Random Circuit Sampling:**

1. The idea is to randomly sample quantum gates (e.g., Haar random 2-qubit gates) to form a circuit.
2. execute the circuit and measure the output to get classical outputs.

3. Finally, test a statistical property of the classical output to make sure that it is really comes from a random. A typical choice for such statistical testing is Cross-Entropy Benchmark (XEB) score. It only requires classical machines to compute the XEB score. It is a score between 0 and 1, where “1” means quantum and “0” means classical.

#### Experiment 1:

- Machine: Willow from Google
- Year: Dec, 2024
- RCS in 5 mins, vs  $10^{25}$  (or 10 septillion) years on today’s fastest supercomputer.

#### Experiment 2:

- Machine: Zuchongzhi 3.0 from USTC, a 105-qubit machine
- Year: March, 2025
- an 80-qubit RCS in “a few” mins, vs 6.4 billion years ( $6.4 \times 10^9$ ) on today’s fastest supercomputer.

#### Warns for this type of results:

- The theoretical foundation for “ideal RCS” has been established by [AC17, AG20], where “strong” hardness assumptions were made.
- These assumptions appear to be questionable by recent theoretical studies [GKC<sup>+</sup>24].
- There is a recent work STOC’23 [AGL<sup>+</sup>23] that show counter examples for “noisy RCS with constant errors.” Roughly, they show how to achieve the same XEB using only a classical algorithm for the constant-noise RCS problem.

## 3.2 Boson Sampling

We do not dive into details. This is a physics-heavy topic we cannot cover in this short talk.

**Just say:** It applies only to Linear Optical Quantum Computing (LOQC).

#### Experiments done:

- Machine: Jiuzhang (76 qubits) from USTC
- Year: 2020
- Jiuzhang finished a Boson sampling instance in 200 seconds, and “claimed” that a classical computer need 2.5 billion years to do so.

**Complexity:** The complexity of Boson Sampling has been rigorously established in the work [AA11]:

- If Boson sampling is easy for classical computers, then  $P^{\#P} = BPP^{NP}$  (which implies that The Polynomial Hierarchy collapses to the 3rd level).

## 4 Classical Proof of Quantumness, Classical Verification of Quantum Computation (CVQC)

### 4.1 Naive Idea: Why cannot we run Shor's algorithm?

- Craig Gidney and Martin Ekerå, How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits. *Quantum*, 2021
- Élie Gouzien and Nicolas Sangouard. Factoring 2048-bit RSA integers in 177 days with 13,436 qubits and a multimode memory. *Physical review letters*, 2021

In contrast, the recent model **IBM - Condor** processor, published in 2023-12-04, support only 1,121 qubits.

### 4.2 Classical Proof of Quantumness

If you are fine with making hardness assumptions like RSA or Factoring, why don't we utilize cryptography?!

We show the largest breakthrough in recent years, in the field of theoretical quantum computing — [Mah18]

#### Trapdoor Claw-Free Function:

Definition:

- $(pk; sk) \leftarrow KGen$
- Each  $y$  has exactly 2 pre-image  $x_0$  and  $x_1$ .
- If have  $sk$ , it is easy to invert the function to learn both of the two preimages.
- For any Adv with  $pk$  only, It is hard to output both (1) an  $(x; y)$  pair, and (2) a string  $d$  such that  $hd; x_0 \quad x_1 i = 0$

The existence of Trapdoor Claw-Free functions follow from standard cryptographic assumptions such as LWE.

#### Mahadev's protocol (oversimplified):

1. V sends  $pk$
2. P sends  $y$
3. V asks randomly (w.p.  $\frac{1}{2}$  each) for either
  - $(x; y)$  pair (V can verify by  $f_{pk}(x) = y$ ), or
  - the string " $d$ " (V can check using  $sk$ )

## 5 security proof

**No classical P can win with probability better than  $1/2$ :** If so, must be winning the second challenge (for string  $d$ ) with non-zero probability. Then, **Xiao: explain the rewinding argument** to extract both  $(x; y)$  and  $d$ . Xiao!

**On the other hand, a quantum P can win with probability  $1$ !**

**Claim 5.1.** Given  $\frac{1}{2}(jx_1i + jx_2i)$  it is easy to find a string  $d$  such that  $hx_0; d + hx_1; d = 0$ .

*Proof.*

$$\begin{aligned} H^{\frac{jx_0i + jx_1i}{2}} &= \frac{1}{2} \sum_{d \in \{0,1\}^n} \left( (-1)^{hx_0;d} + (-1)^{hx_1;d} \right) |d\rangle \\ &= \frac{1}{2^{n+1}} \sum_{d \in \{0,1\}^n} \left( (-1)^{hx_0;d} + (-1)^{hx_1;d} \right) |d\rangle \end{aligned}$$

Measure the final register, we will obtain an  $d$  such that

$$(-1)^{hx_0;d} + (-1)^{hx_1;d} \neq 0:$$

For such an  $d$ , it must hold that

$$\begin{aligned} &hx_0; d = hx_1; d \\ &hx_0; d - hx_1; d = 0 \\ &hx_0 - x_1; d = 0 \end{aligned}$$

□

Note that this above step has already appeared in the famous Simon's algorithm.

## 6 How many qubits to execute Mahadev's protocol?

Xiao: I don't know the exact number. According to google, roughly 3,000 qubits.

Xiao!

## References

- [AA11] Scott Aaronson and Alex Arkhipov. The computational complexity of linear optics. In Lance Fortnow and Salil P. Vadhan, editors, *43rd Annual ACM Symposium on Theory of Computing* pages 333–342, San Jose, CA, USA, June 6–8, 2011. ACM Press. doi : [10.1145/1993636.1993682](https://doi.org/10.1145/1993636.1993682). 4
- [AC17] Scott Aaronson and Lijie Chen. Complexity-theoretic foundations of quantum supremacy experiments. In *32nd Computational Complexity Conference*, page 1, 2017. 4
- [AG20] Scott Aaronson and Sam Gunn. On the classical hardness of spoofing linear cross-entropy benchmarking. *Theory Comput.*, 16:1–8, 2020. URL: <https://doi.org/10.4086/toc.2020.v016a011>, doi : [10.4086/TOC.2020.V016A011](https://doi.org/10.4086/TOC.2020.V016A011). 4
- [AGL<sup>+</sup>23] Dorit Aharonov, Xun Gao, Zeph Landau, Yunchao Liu, and Umesh V. Vazirani. A polynomial-time classical algorithm for noisy random circuit sampling. In Barna Saha and Rocco A. Servedio, editors, *Proceedings of the 55th Annual ACM Symposium on Theory of Computing STOC 2023, Orlando, FL, USA, June 20–23, 2023* pages 945–957. ACM, 2023. doi : [10.1145/3564246.3585234](https://doi.org/10.1145/3564246.3585234). 4
- [AKR10] Boris Altshuler, Hari Krovi, and Jérémie Roland. Anderson localization makes adiabatic quantum optimization fail. *Proceedings of the National Academy of Sciences*, 107(28):12446–12450, 2010. 3
- [GKC<sup>+</sup>24] Xun Gao, Marcin Kalinowski, Chi-Ning Chou, Mikhail D Lukin, Boaz Barak, and Soonwon Choi. Limitations of linear cross-entropy as a measure for quantum advantage. *PRX Quantum*, 5(1):010334, 2024. 4
- [Mah18] Urmila Mahadev. Classical verification of quantum computations. In Mikkel Thorup, editor, *59th Annual Symposium on Foundations of Computer Science*, pages 259–267, Paris, France, October 7–9, 2018. IEEE Computer Society Press. doi : [10.1109/FOCS.2018.00033](https://doi.org/10.1109/FOCS.2018.00033). 5
- [NC11] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, USA, 10th edition, 2011.
- [Rei04] Ben Reichardt. The quantum adiabatic optimization algorithm and local minima. In László Babai, editor, *36th Annual ACM Symposium on Theory of Computing* pages 502–510, Chicago, IL, USA, June 13–16, 2004. ACM Press. doi : [10.1145/1007352.1007428](https://doi.org/10.1145/1007352.1007428). 3
- [vMV01] Wim van Dam, Michele Mosca, and Umesh V. Vazirani. How powerful is adiabatic quantum computation? In *42nd Annual Symposium on Foundations of Computer Science*, pages 279–287, Las Vegas, NV, USA, October 14–17, 2001. IEEE Computer Society Press. doi : [10.1109/SFCS.2001.959902](https://doi.org/10.1109/SFCS.2001.959902). 3

## FiXme Information

### List of Corrections

Xiao: explain the rewinding argument . . . . .	5
Xiao: I don't know the exact number. According to google, roughly 3,000 qubits. . . . .	6