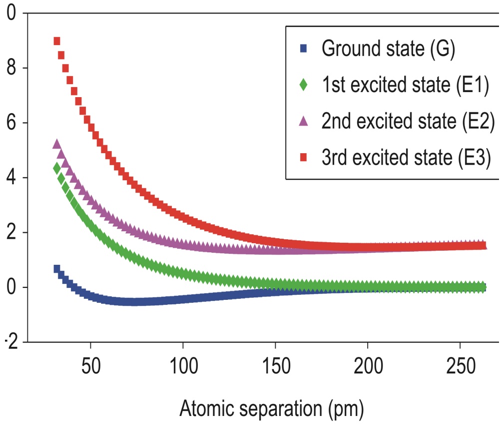
Logic Formulas as Program Abstractions for Quantum Circuits:
A Case Study in Noisy Variational Algorithm Simulation

Yipeng Huang

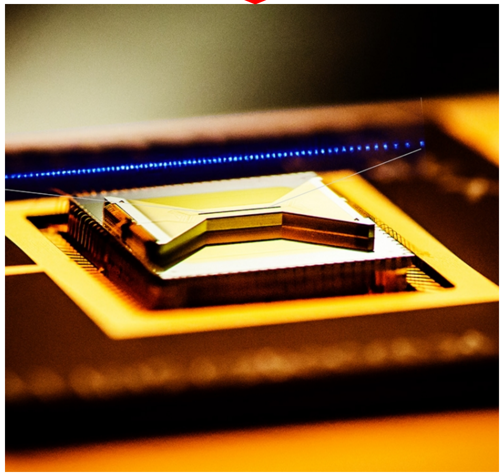
Rutgers CS Systems Reading Group

October 28, 2020

RUTGERS



Quantum computer systems



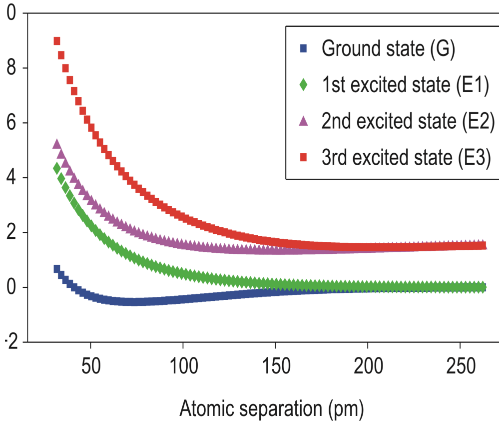
Awe-inspiring quantum algorithms

Chemistry simulations from governing equations	Shor’s algorithm for factoring integers	Hundreds more near-term and far-future algorithms
Quantum computers as quantum mechanics simulator	Surpasses any known classical algorithm	QuantumAlgorithmZoo.org

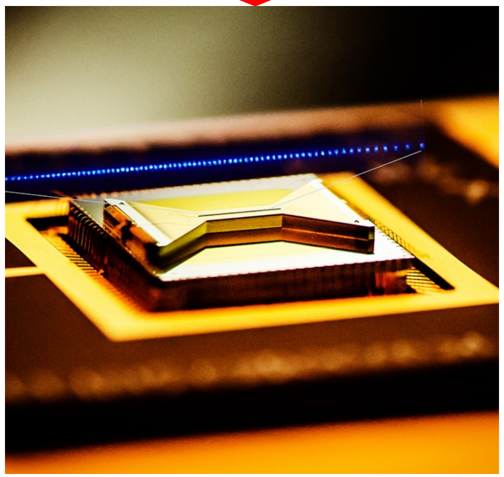
Now-viable quantum prototypes

Superconducting qubits	Trapped ion qubits	Dozens of candidate qubit technologies
IBM, Google, Rigetti, …	IonQ, UMD, Honeywell, …	May yet surpass current leaders in capacity and reliability

Image sources: Lanyon and Whitfield et al., 2010, IonQ



Quantum computer systems



Awe-inspiring quantum algorithms

Chemistry simulations from governing equations	Shor's algorithm for factoring integers	Hundreds more near-term and far-future algorithms
Quantum computers as quantum mechanics simulator	Surpasses any known classical algorithm	QuantumAlgorithmZoo.org

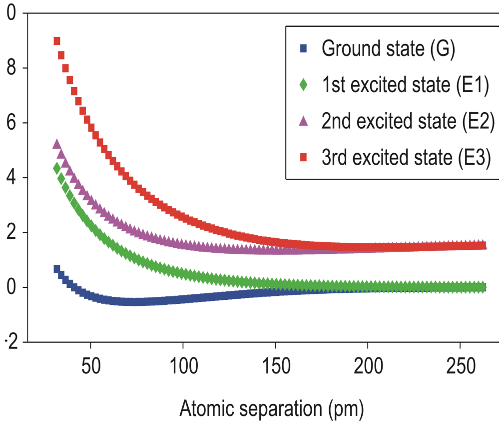
Quantum software-hardware gap

Quantum software frameworks	Quantum programming languages	Quantum architectures and microarchitectures
Simulators. Open source frameworks. Cloud accessible quantum prototypes.	Higher level programming abstractions. Debuggers. Intermediate representations.	Universal logical gate sets. Optimized place and route. Analog-digital quantum interface.

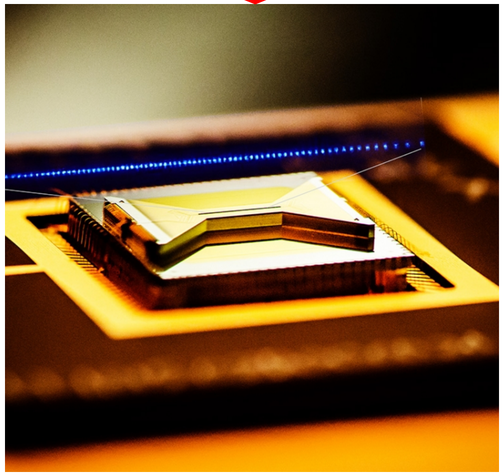
Now-viable quantum prototypes

Superconducting qubits	Trapped ion qubits	Dozens of candidate qubit technologies
IBM, Google, Rigetti, …	IonQ, UMD, Honeywell, …	May yet surpass current leaders in capacity and reliability

Image sources: Lanyon and Whitfield et al., 2010, IonQ



Quantum computer systems



Awe-inspiring quantum algorithms

Chemistry simulations from governing equations	Shor's algorithm for factoring integers	Hundreds more near-term and far-future algorithms
Quantum computers as quantum mechanics simulator	Surpasses any known classical algorithm	QuantumAlgorithmZoo.org

Quantum software-hardware gap

Quantum software frameworks	Quantum programming languages	Quantum architectures and microarchitectures
Simulators. Open source frameworks. Cloud accessible quantum prototypes.	Higher level programming abstractions. Debuggers. *Intermediate representations.*	Universal logical gate sets. Optimized place and route. Analog-digital quantum interface.

Now-viable quantum prototypes

Superconducting qubits	Trapped ion qubits	Dozens of candidate qubit technologies
IBM, Google, Rigetti, …	IonQ, UMD, Honeywell, …	May yet surpass current leaders in capacity and reliability

Image sources: Lanyon and Whitfield et al., 2010, IonQ

What this talk is about:

Using classical probabilistic inference techniques as an abstraction for quantum computing.

- A new way to represent noisy quantum circuits as probabilistic graphical models.
- A new way to encode quantum circuits as conjunctive normal forms and arithmetic circuits.
- A new way to manipulate quantum circuits using logical equation satisfiability solvers.
- Improved simulation and sampling performance for important near-term quantum algorithms.

Where we are going:

What are quantum variational algorithms?

- Why are they different and important?

What is quantum circuit simulation?

- Why are the conventional techniques insufficient?

How do we represent quantum circuits as logic formulas?

- Why does it help with variational algorithm simulation, and by how much?

Where we are going:

<u>What are quantum variational algorithms?</u>

- *<u>Why are they different and important?</u>*

What is quantum circuit simulation?

- Why are the conventional techniques insufficient?

How do we represent quantum circuits as logic formulas?

- Why does it help with variational algorithm simulation, and by how much?

Limitations of near-term quantum computers

- Limited number of qubits (the fundamental information units and devices in quantum computing).
- Noisy, unreliable operations.
- Limited operations on each qubit.
- Error correction too costly (needs ~million qubits), not available.

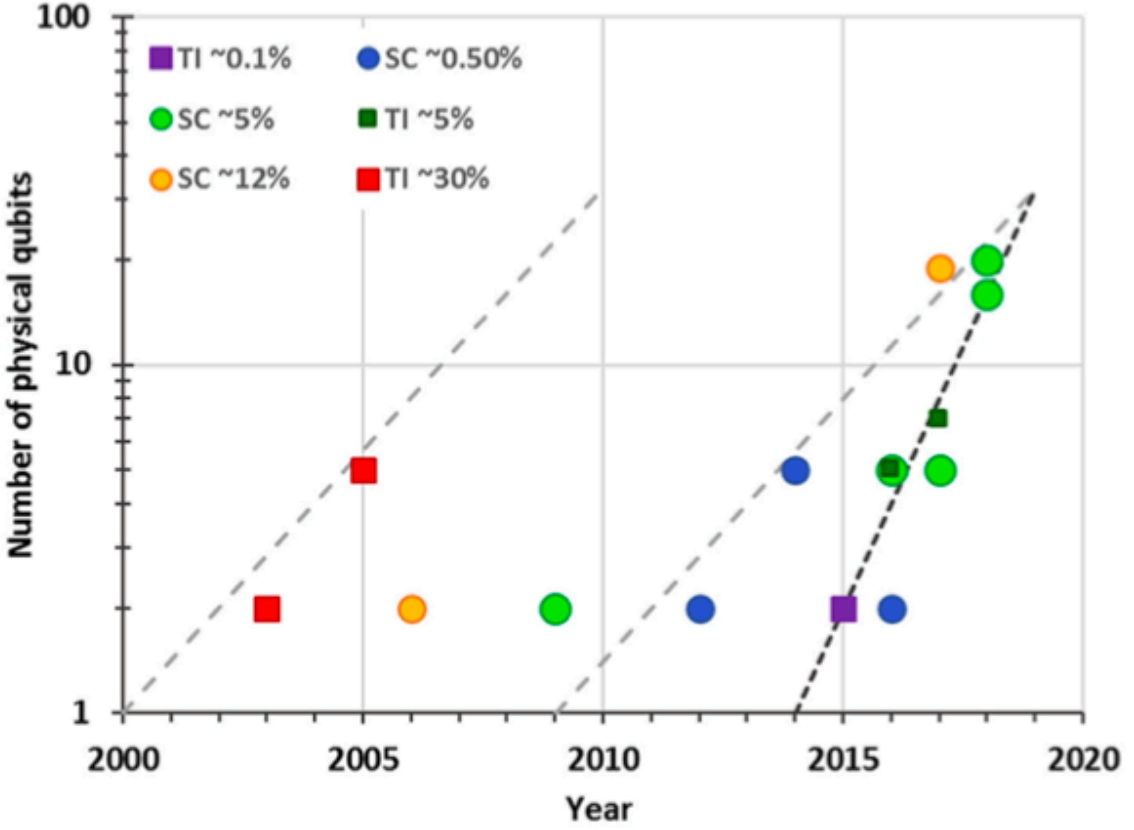


FIGURE 7.2 The number of qubits in superconductor (SC) and trapped ion (TI) quantum computers versus year; note the logarithmic scaling of the vertical axis. Data for trapped ions are shown as squares and for superconducting machines are shown as circles. Approximate average reported two-qubit gate error rates are indicated by color; points with the same color have similar error rates. The dashed gray lines show how the number of qubits would grow if they double every two years starting with one qubit in 2000 and 2009, respectively; the dashed black line indicates a doubling every year beginning with one qubit in 2014. Recent superconductor growth has been close to doubling every year. If this rate continued, 50 qubit machines with less than 5 percent error rates would be reported in 2019. SOURCE: Plotted data obtained from multiple sources [9].

- Reminiscent of Moore's bold prediction.
- Exponential growth of qubits, each providing exponentially more computing capacity.

Image source: Quantum Computing Progress and Prospects. National Academies Press. 2019.

Limitations of near-term quantum computers

- Limited number of qubits (the fundamental information units and devices in quantum computing).
- Noisy, unreliable operations.
- Limited operations on each qubit.
- Error correction too costly (needs ~million qubits), not available.

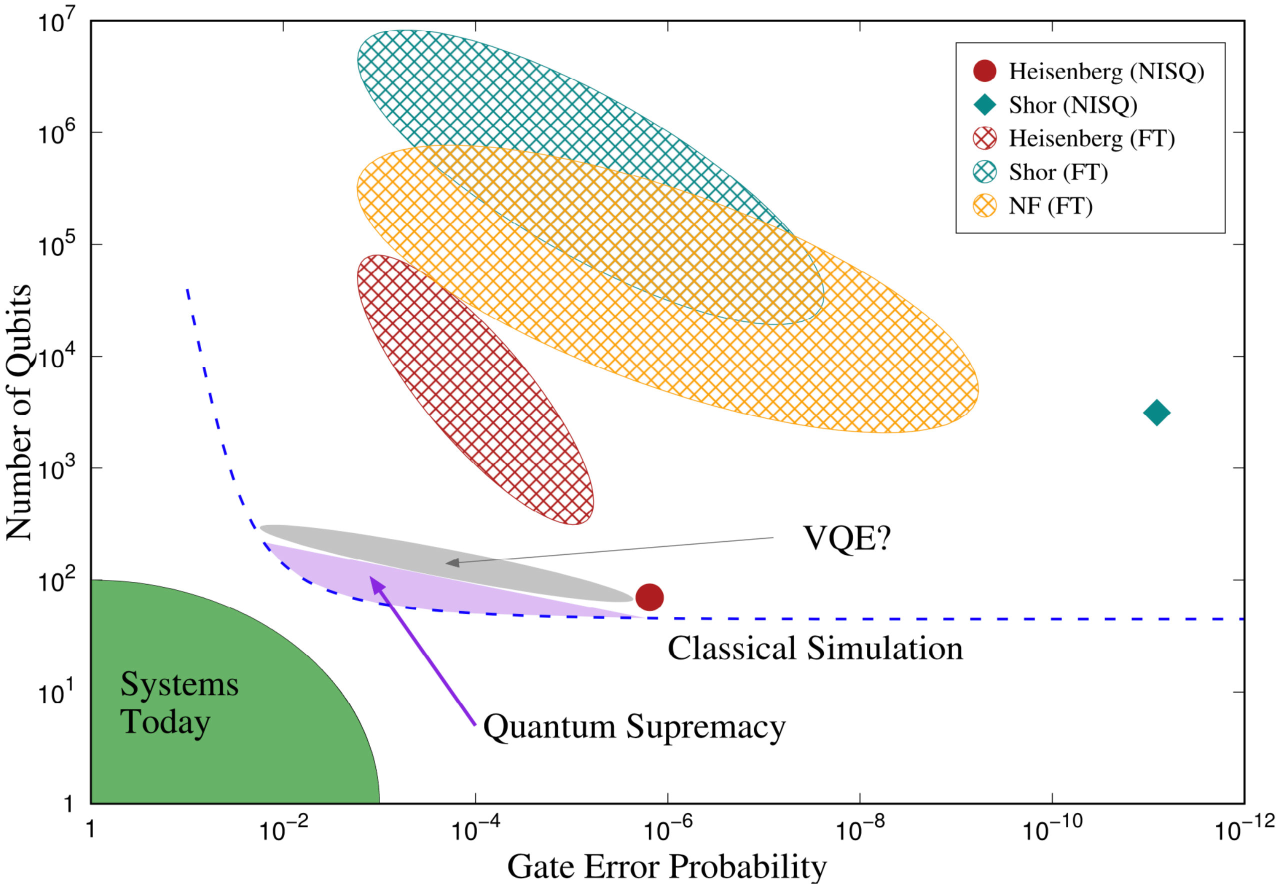


Fig. 2. *Performance space of quantum computers, measured by the error probability of each entangling gate in the horizontal axis (roughly inversely proportional to the total number of gates that can be executed on a NISQ machine), and the number of qubits in the system in the vertical axis. Blue dotted line approximately demarcates quantum systems that can be simulated using best classical computers, while the green colored region shows where the existing quantum computing systems with verified performance numbers lie (as of September 2018). Purple shaded region indicates computational tasks that accomplish the so-called "quantum supremacy," where the computation carried out by the quantum computer defies classical simulation regardless of its usefulness. The different shapes illustrate resource counts for solving various problems, with solid symbols corresponding to the exact entangling gate counts and number of qubits in NISQ machines, and shaded regions showing approximate gate error requirements and number of qubits for an FT implementation (not pictured are the regions where the error gets too close to the known fault-tolerance thresholds): cyan diamond and shaded region correspond to factoring a 1024-bit number using Shor's algorithm [14], magenta circle and shaded region represent simulation of a 72-spin Heisenberg model [20], and orange shaded region illustrates NF simulation [21].*

An Outlook for Quantum Computing. Maslov (IBM) et al. 2018.

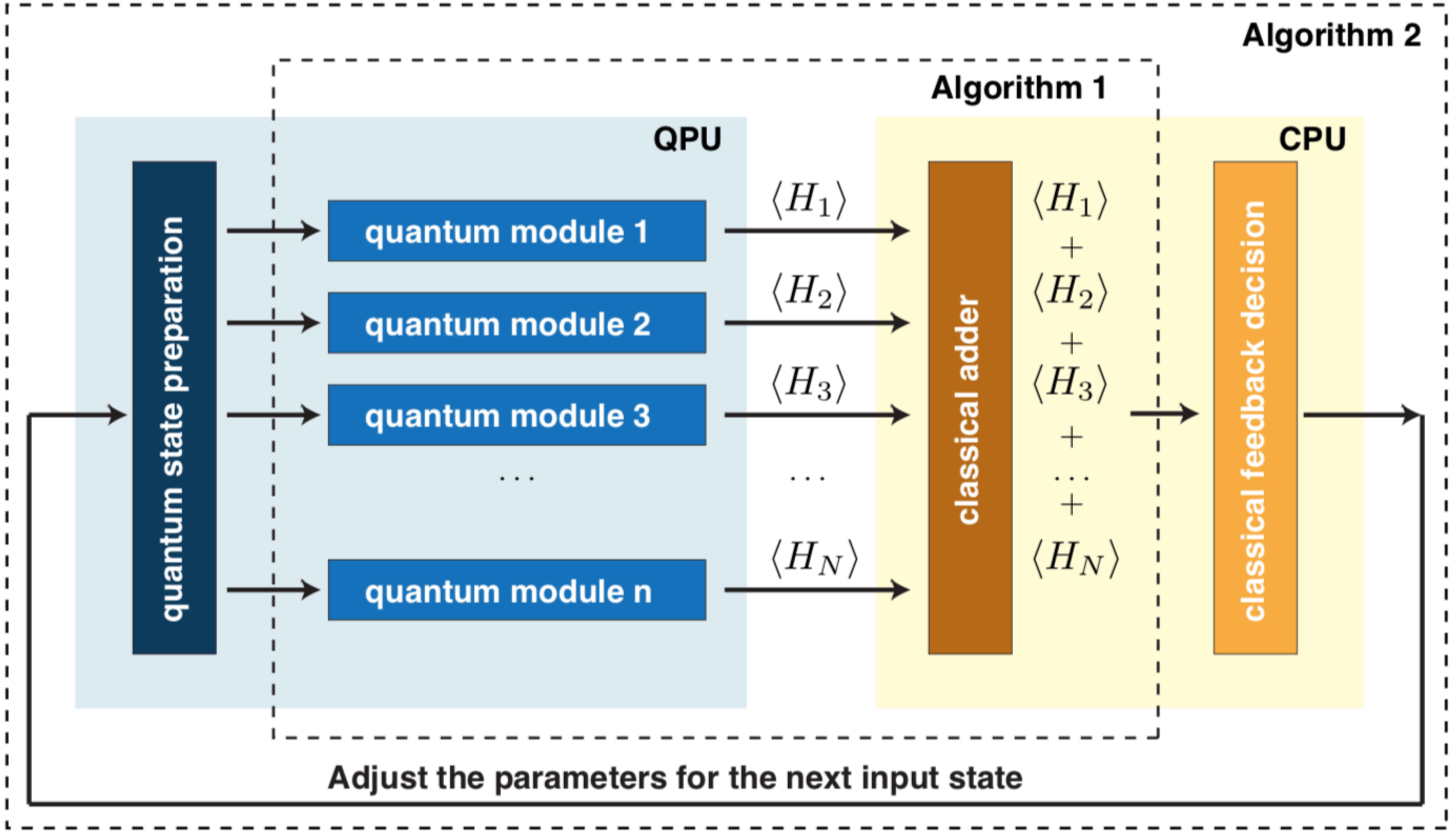
NISQ systems target variational algorithms.

Near-term Intermediate Scale Quantum (NISQ) systems have ~100 qubits with at best 0.1% error rate.

With that capacity and reliability, error correction, along with famous algorithms such as Grover’s search and Shor’s factoring are infeasible.

The soonest candidates for useful quantum computation involve quantum-classical variational algorithms.

Hybrid quantum-classical variational algos

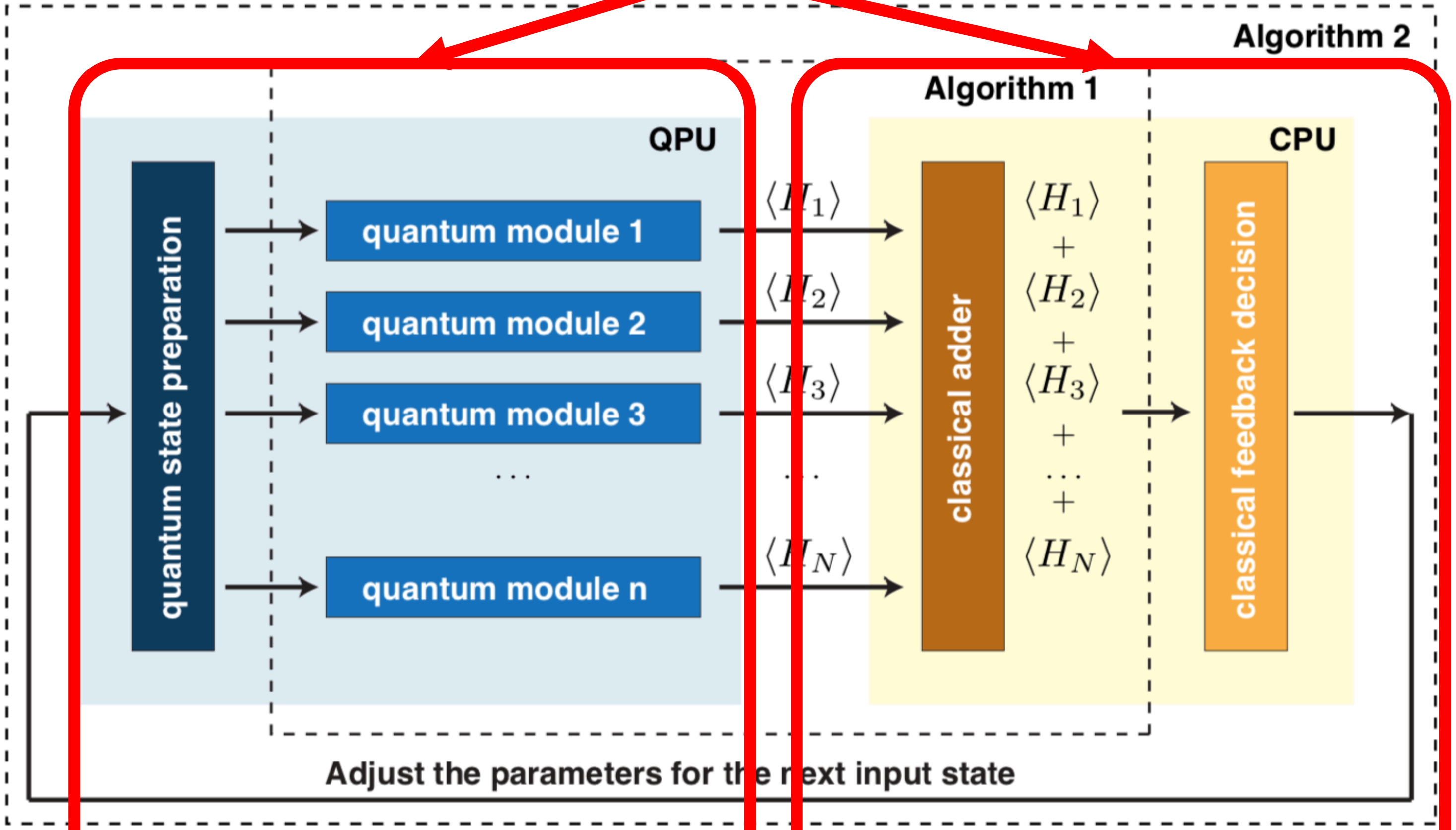


It's like using a classical computer to train a quantum neural network.

Image source: Peruzzo et al., 2013

Hybrid quantum-classical variational algos

Use quantum & classical computation



Quantum evaluates an objective function

Classical optimizes for better parameters

It's like using a classical computer to train a quantum neural network.

Image source: Peruzzo et al., 2013

Unique traits of variational algorithms

Provides meaningful results with noise even without error correction.

Draws on strengths of quantum and classical:

- Repeatedly prepare and measure quantum states.
- Optimize for a set of optimal parameters based on classical measurements.

Wide but shallow circuits (not many operations on many qubits).

Specific examples of variational algorithms

Variational quantum eigensolver (VQE)

Simulate quantum mechanics.

Quantum approximate optimization algorithm (QAOA)

Approximate solutions to constraint satisfaction problems (CSPs).

Where we are going:

What are quantum variational algorithms?

- Why are they different and important?

<u>What is quantum circuit simulation?</u>

- *<u>Why are the conventional techniques insufficient?</u>*

How do we represent quantum circuits as logic formulas?

- Why does it help with variational algorithm simulation, and by how much?

Role of simulation in classical computer systems

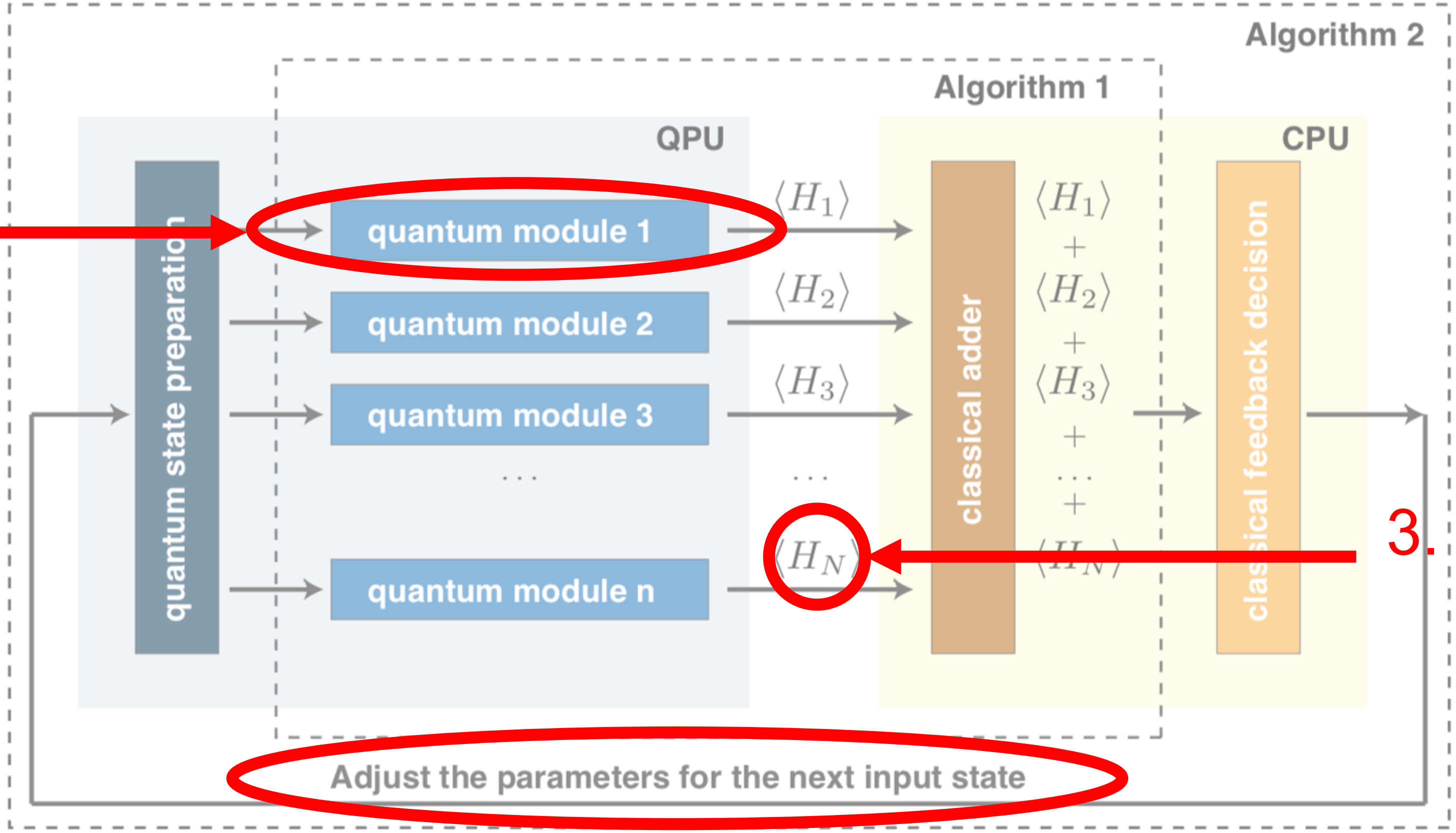
- VirtualBox
- Gem5
- Synopsys / Cadence
- HSpice

Image credit: Warner Brothers Pictures

- Even once we have quantum computer systems, building & testing quantum computers will rely in part on classical computer systems.

The unique challenge of simulating noisy variational algorithms

1. Needs to simulate noise, and quantum circuits are wide but shallow



3. Only need samples, not full wavefunctions.

2. Require repeated simulation with different parameters

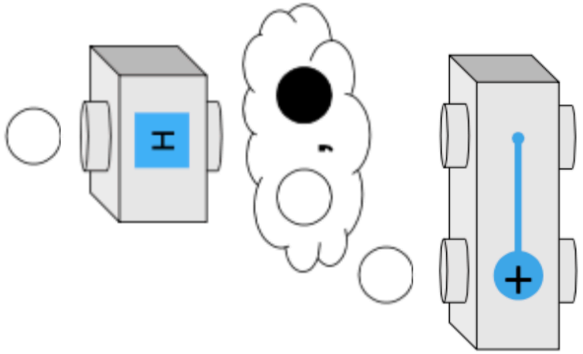
Image source: Peruzzo et al., 2013

Rock, paper, scissors: Existing simulation techniques are not suited for variational algorithms

<u>Schrödinger simulation</u>

QuEST, IBM, Google; parallel matrix vector multiplication

Schrödinger quantum circuit simulation



$$\mathrm{CNOT}(\mathrm{H} \otimes \mathrm{I}|00\rangle) = \mathrm{CNOT}(\mathrm{H}|0\rangle \otimes \mathrm{I}|0\rangle) = \mathrm{CNOT}\begin{bmatrix} \frac{1}{\sqrt{2}}\begin{bmatrix}1\\0\end{bmatrix} \\ \frac{1}{\sqrt{2}}\begin{bmatrix}1\\0\end{bmatrix} \end{bmatrix} = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & & 1 \\ & & 1 & \end{bmatrix}\begin{bmatrix} 1/\sqrt{2} \\ 0 \\ 1/\sqrt{2} \\ 0 \end{bmatrix} = \begin{bmatrix} 1/\sqrt{2} \\ 0 \\ 0 \\ 1/\sqrt{2} \end{bmatrix} = \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle$$

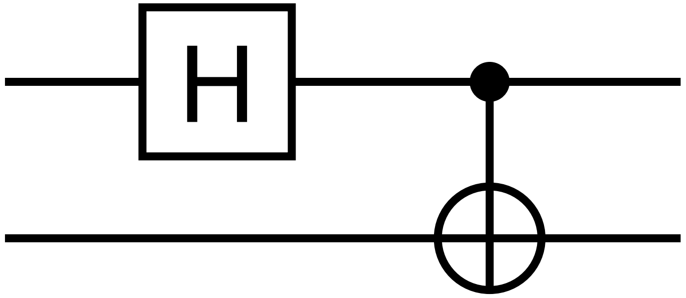
Rock, paper, scissors: Existing simulation techniques are not suited for variational algorithms

	Schrödinger simulation QuEST, IBM, Google; parallel matrix vector multiplication
1. Does it excel at simulating wide but shallow circuits?	✗
2. Does it extract structure for repeated simulation with different parameters?	✗
3. Does it efficiently sample from the final wavefunction?	✓

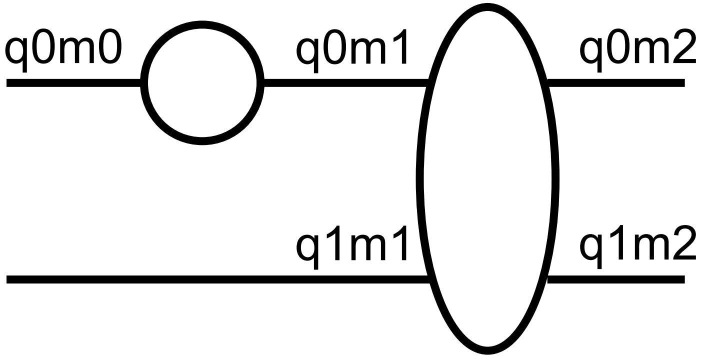
Rock, paper, scissors: Existing simulation techniques are not suited for variational algorithms

	Schrödinger simulation	Feynman simulation
	QuEST, IBM, Google; parallel matrix vector multiplication	qTorch; graphical model tensor network contraction
1. Does it excel at simulating wide but shallow circuits?	✗	
2. Does it extract structure for repeated simulation with different parameters?	✗	
3. Does it efficiently sample from the final wavefunction?	✓	

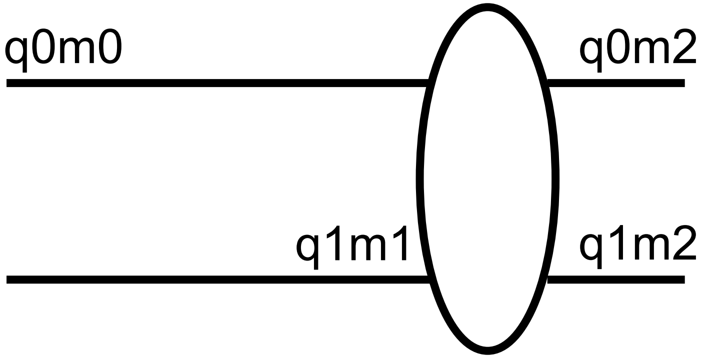
Feynman quantum circuit simulation



	q0m0=$\|\mathbf{0}\rangle$	q0m0=$\|\mathbf{1}\rangle$
q0m1=$\|\mathbf{0}\rangle$	$\frac{1}{\sqrt{2}}$	$\frac{1}{\sqrt{2}}$
q0m1=$\|\mathbf{1}\rangle$	$\frac{1}{\sqrt{2}}$	$\frac{-1}{\sqrt{2}}$



		q0m1=$\|\mathbf{0}\rangle$		q0m1=$\|\mathbf{1}\rangle$	
		q1m1=$\|\mathbf{0}\rangle$	q1m1=$\|\mathbf{1}\rangle$	q1m1=$\|\mathbf{0}\rangle$	q1m1=$\|\mathbf{1}\rangle$
q0m2=$\|\mathbf{0}\rangle$	q1m2=$\|\mathbf{0}\rangle$	1	0	0	0
	q1m2=$\|\mathbf{1}\rangle$	0	1	0	0
q0m2=$\|\mathbf{1}\rangle$	q1m2=$\|\mathbf{0}\rangle$	0	0	0	1
	q1m2=$\|\mathbf{1}\rangle$	0	0	1	0



		q0m0=$\|\mathbf{0}\rangle$		q0m0=$\|\mathbf{1}\rangle$	
		q1m1=$\|\mathbf{0}\rangle$	q1m1=$\|\mathbf{1}\rangle$	q1m1=$\|\mathbf{0}\rangle$	q1m1=$\|\mathbf{1}\rangle$
q0m2=$\|\mathbf{0}\rangle$	q1m2=$\|\mathbf{0}\rangle$	$1/\sqrt{2}$	0	$1/\sqrt{2}$	0
	q1m2=$\|\mathbf{1}\rangle$	0	$1/\sqrt{2}$	0	$1/\sqrt{2}$
q0m2=$\|\mathbf{1}\rangle$	q1m2=$\|\mathbf{0}\rangle$	0	$1/\sqrt{2}$	0	$-1/\sqrt{2}$
	q1m2=$\|\mathbf{1}\rangle$	$1/\sqrt{2}$	0	$-1/\sqrt{2}$	0

Rock, paper, scissors: Existing simulation techniques are not suited for variational algorithms

	Schrödinger simulation QuEST, IBM, Google; parallel matrix vector multiplication	Feynman simulation qTorch; graphical model tensor network contraction
1. Does it excel at simulating wide but shallow circuits?	✗	✓
2. Does it extract structure for repeated simulation with different parameters?	✗	?
3. Does it efficiently sample from the final wavefunction?	✓	✗

Rock, paper, scissors: Existing simulation techniques are not suited for variational algorithms

	Schrödinger simulation QuEST, IBM, Google; parallel matrix vector multiplication	Feynman simulation qTorch; graphical model tensor network contraction	Binary decision diagram simulation QUIDD, Viamontes; Zulehner, Wille et al.
1. Does it excel at simulating wide but shallow circuits?	✗	✓	?
2. Does it extract structure for repeated simulation with different parameters?	✗	?	✓
3. Does it efficiently sample from the final wavefunction?	✓	✗	?

Where we are going:

What are quantum variational algorithms?

- Why are they different and important?

What is quantum circuit simulation?

- Why are the conventional techniques insufficient?

<u>How do we represent quantum circuits as logic formulas?</u>

- Why does it help with variational algorithm simulation, and by how much?

Our toolchain: Bayesian network knowledge compilation for noisy quantum circuit simulation and sampling

1. Noisy quantum circuits to Bayesian network
2. Bayesian networks to conjunctive normal form (CNF)
3. CNF to arithmetic circuit (AC)
4. Exact inference on AC for quantum circuit simulation
5. Gibbs sampling on AC to sample from final wavefunction

Our toolchain: Bayesian network knowledge compilation for noisy quantum circuit simulation and sampling

1. Noisy quantum circuits to Bayesian network
2. Bayesian networks to conjunctive normal form (CNF)
3. CNF to arithmetic circuit (AC)
4. Exact inference on AC for quantum circuit simulation
5. Gibbs sampling on AC to sample from final wavefunction

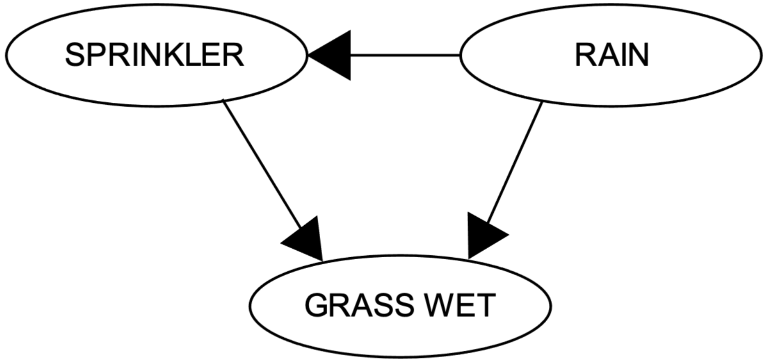
1. Needs to simulate noise
2. Repeated simulation with different parameters
3. Only need samples, not full wavefunctions

Our toolchain: Bayesian network knowledge compilation for noisy quantum circuit simulation and sampling

1. ***Noisy quantum circuits to Bayesian network***
2. Bayesian networks to conjunctive normal form (CNF)
3. CNF to arithmetic circuit (AC)
4. Exact inference on AC for quantum circuit simulation
5. Gibbs sampling on AC to sample from final wavefunction

Bayesian networks: AI models that encode probabilistic knowledge in a factorized format

	SPRINKLER	
RAIN	T	F
F	0.4	0.6
T	0.01	0.99

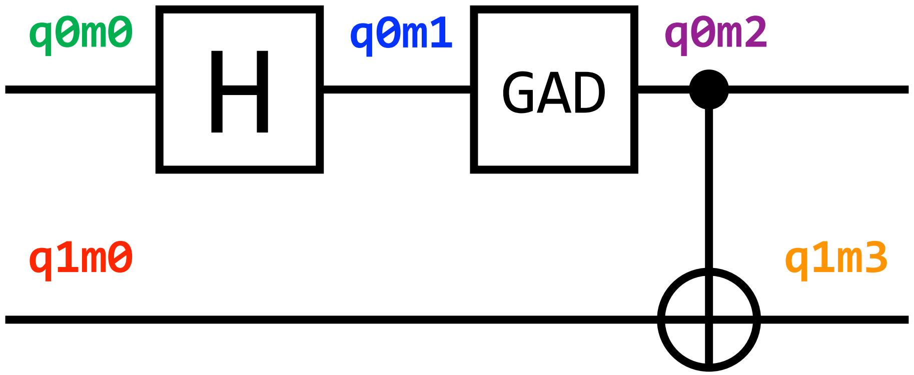


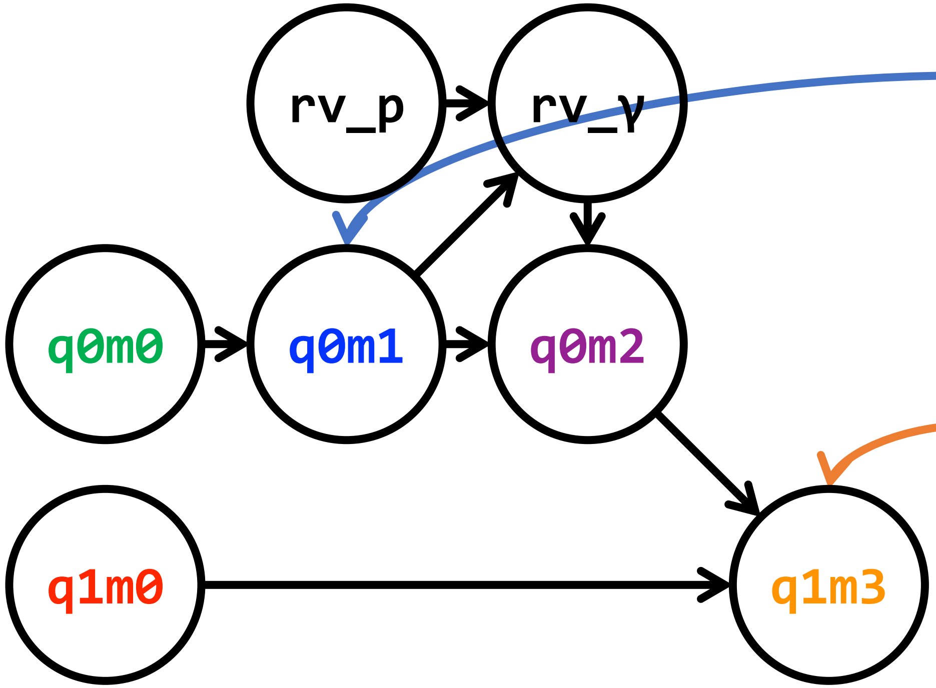
RAIN	
T	F
0.2	0.8

		GRASS WET	
SPRINKLER	RAIN	T	F
F	F	0.0	1.0
F	T	0.8	0.2
T	F	0.9	0.1
T	T	0.99	0.01

Image source: Wikimedia

Noisy quantum circuits to Bayesian network





q0m0	P(q0m1=|0⟩)	P(q0m1=|1⟩)
|0⟩	$+1/\sqrt{2}$	$+1/\sqrt{2}$
|1⟩	$+1/\sqrt{2}$	$-1/\sqrt{2}$

Control q0m2	Target q1m0	P(q1m3=|0⟩)	P(q1m3=|1⟩)
|0⟩	|0⟩	1.	0.
|0⟩	|1⟩	0.	1.
|1⟩	|0⟩	0.	1.
|1⟩	|1⟩	1.	0.

Connection between quantum circuits and probabilistic graphical models

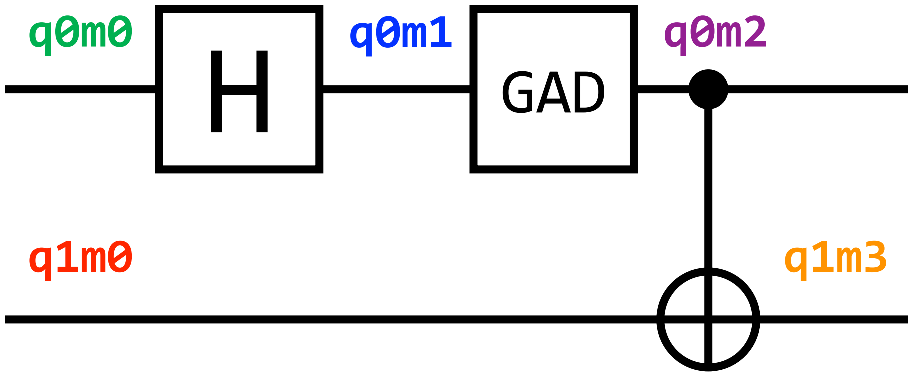
	Quantum	Probabilistic
Key analogies	program simulation qubits amplitudes operator unitary matrices superposition states entangled qubits measurement	inference random variables probabilities conditional probability tables probability distributions dependent random variables sampling & conditioning
Key distinctions	amplitudes are complex-valued squares of amplitudes sum to 1 interference (canceling of amplitudes) possible	probabilities between 0 and 1 probabilities sum to 1 interference impossible

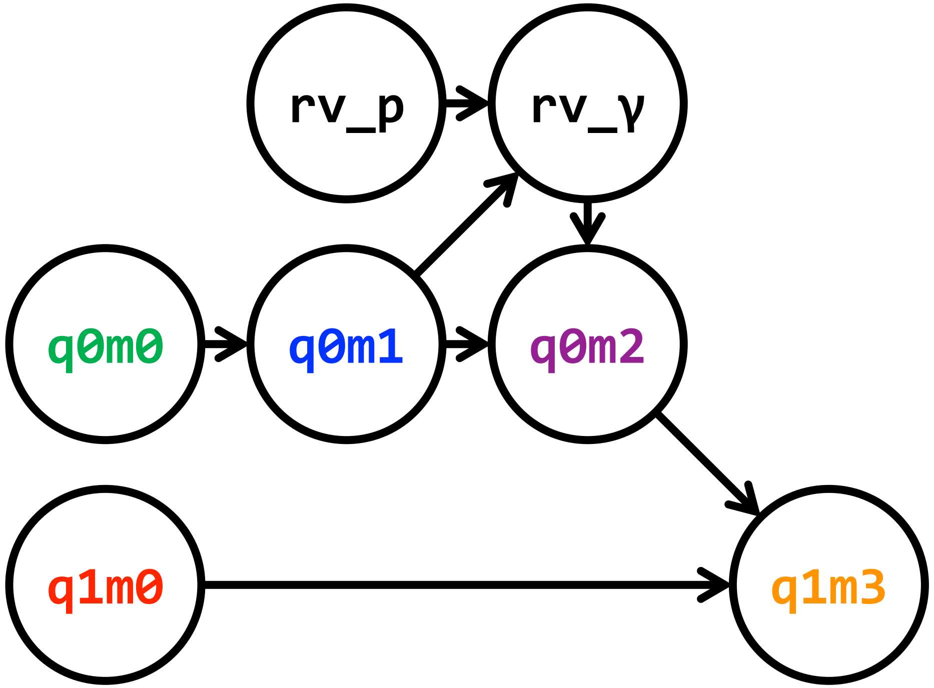
Quantum / probabilistic:
Separated by Gottesman-Knill theorem, ideas can cross-pollinate

Our toolchain: Bayesian network knowledge compilation for noisy quantum circuit simulation and sampling

1. Noisy quantum circuits to Bayesian network
2. ***<u>Bayesian networks to conjunctive normal form (CNF)</u>***
3. CNF to arithmetic circuit (AC)
4. Exact inference on AC for quantum circuit simulation
5. Gibbs sampling on AC to sample from final wavefunction

Bayesian networks to conjunctive normal form (CNF)





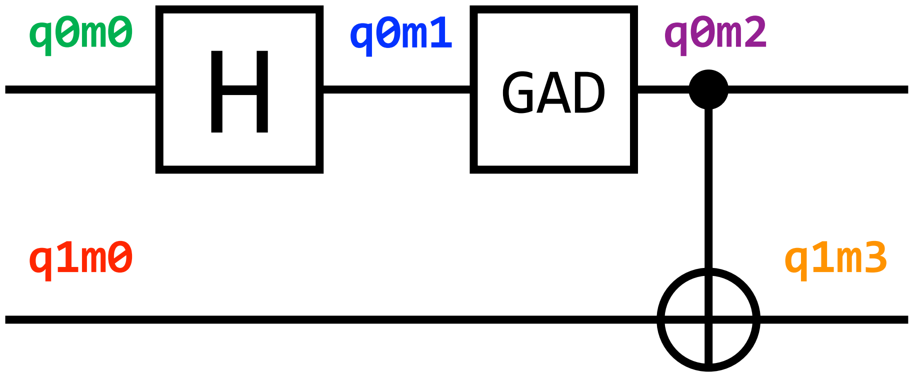
Think about circuit as logic equation

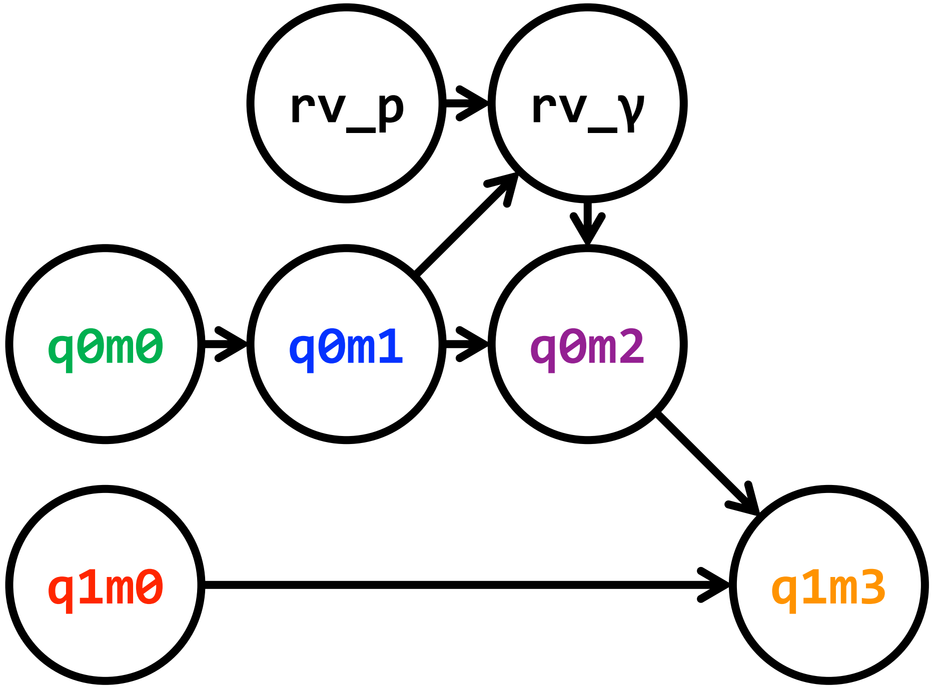
Compile & minimize this logic equation

Variable assignments that satisfy CNF are valid Feynman paths through algorithm

- Model count on variable assignments yields quantum circuit simulation

Bayesian networks to conjunctive normal form (CNF)



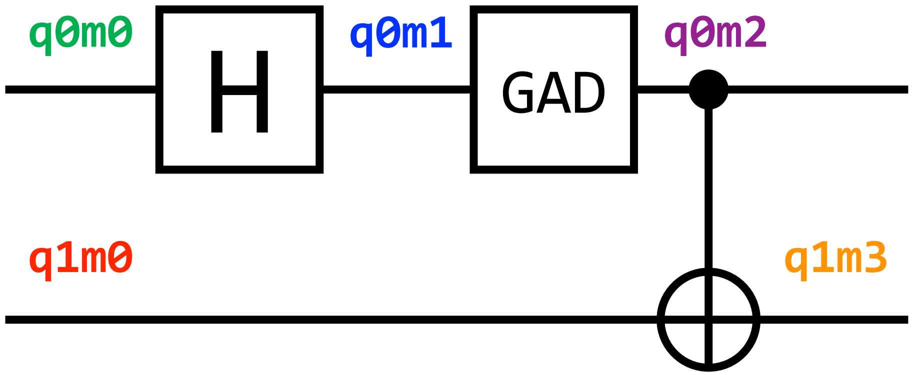


Qubits take on binary values:

q0m0=|0> XOR q0m0=|1>
q0m1=|0> XOR q0m1=|1>
q0m2=|0> XOR q0m2=|1>

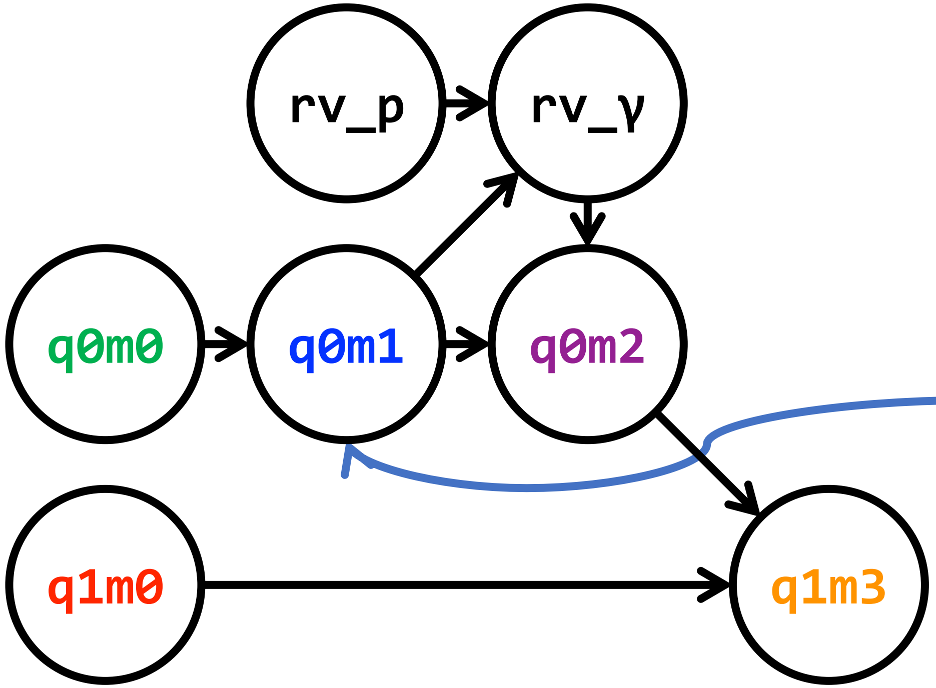
q1m0=|0> XOR q1m0=|1>
q1m3=|0> XOR q1m3=|1>

Bayesian networks to conjunctive normal form (CNF)



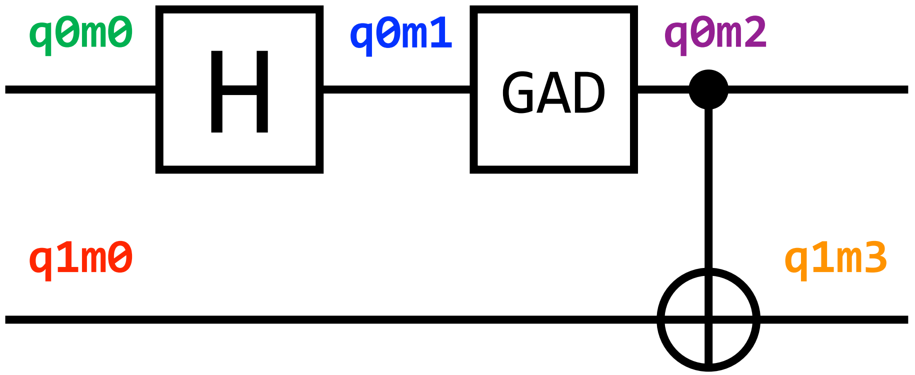
The Hadamard gate:

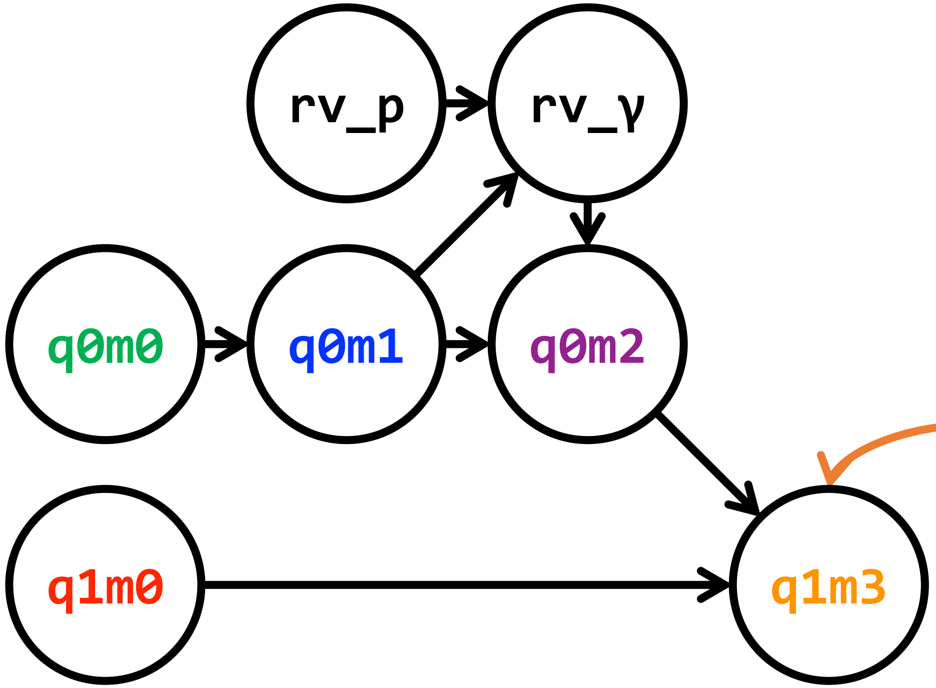
```
q0m0=|0> AND q0m1=|0> -> +1/sqrt(2)
q0m0=|0> AND q0m1=|1> -> +1/sqrt(2)
q0m0=|1> AND q0m1=|0> -> +1/sqrt(2)
q0m0=|1> AND q0m1=|1> -> -1/sqrt(2)
```



q0m0	P(q0m1=$\|0\rangle$)	P(q0m1=$\|1\rangle$)
$\|0\rangle$	$+1/\sqrt{2}$	$+1/\sqrt{2}$
$\|1\rangle$	$+1/\sqrt{2}$	$-1/\sqrt{2}$

Bayesian networks to conjunctive normal form (CNF)



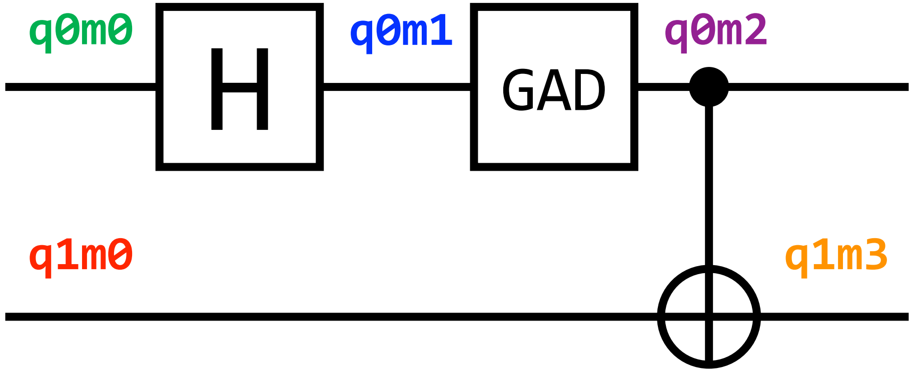


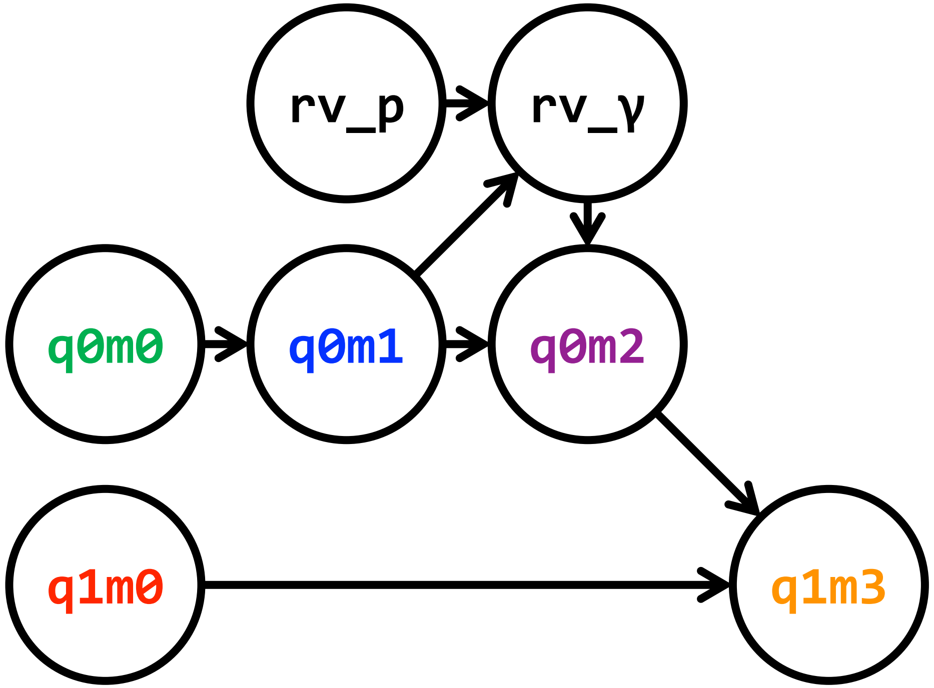
The CNOT gate:

q0m2=|0> AND q1m0=|0> -> q1m3=|0>
q0m2=|0> AND q1m0=|1> -> q1m3=|1>
q0m2=|1> AND q1m0=|0> -> q1m3=|1>
q0m2=|1> AND q1m0=|1> -> q1m3=|0>

Control q0m2	Target q1m0	P(q1m3=|0⟩)	P(q1m3=|1⟩)
|0⟩	|0⟩	1.	0.
|0⟩	|1⟩	0.	1.
|1⟩	|0⟩	0.	1.
|1⟩	|1⟩	1.	0.

Bayesian networks to conjunctive normal form (CNF)





Put all the sentences together!

Convert logical implications "→" to logical disjunctions

Conjoin all the disjunctive clauses together to form CNF
(i.e., AND all the ORs together)

Our toolchain: Bayesian network knowledge compilation for noisy quantum circuit simulation and sampling

1. Noisy quantum circuits to Bayesian network
2. Bayesian networks to conjunctive normal form (CNF)
3. ***CNF to arithmetic circuit (AC)***
4. Exact inference on AC for quantum circuit simulation
5. Gibbs sampling on AC to sample from final wavefunction

CNF to arithmetic circuit (AC)

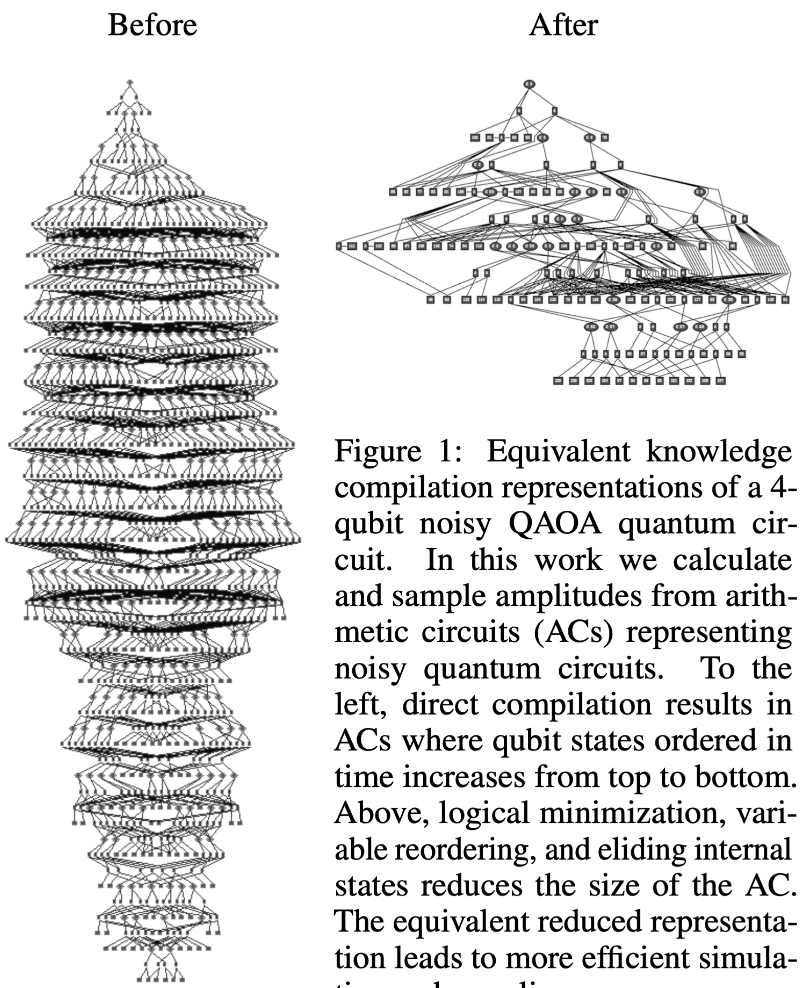


Figure 1: Equivalent knowledge compilation representations of a 4-qubit noisy QAOA quantum circuit. In this work we calculate and sample amplitudes from arithmetic circuits (ACs) representing noisy quantum circuits. To the left, direct compilation results in ACs where qubit states ordered in time increases from top to bottom. Above, logical minimization, variable reordering, and eliding internal states reduces the size of the AC. The equivalent reduced representation leads to more efficient simulation and sampling.

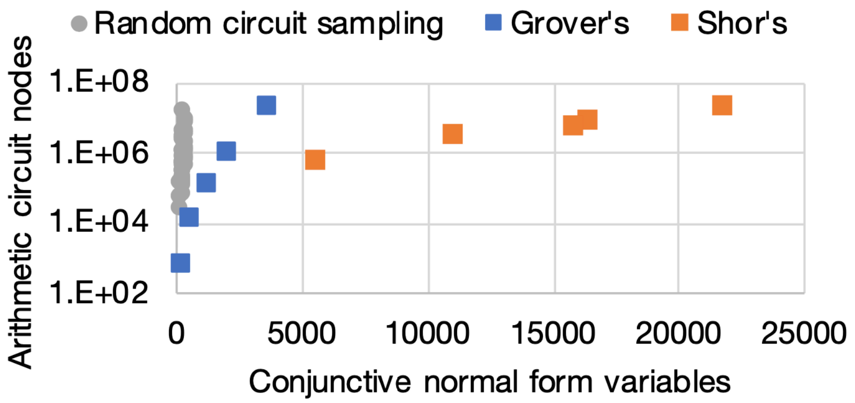
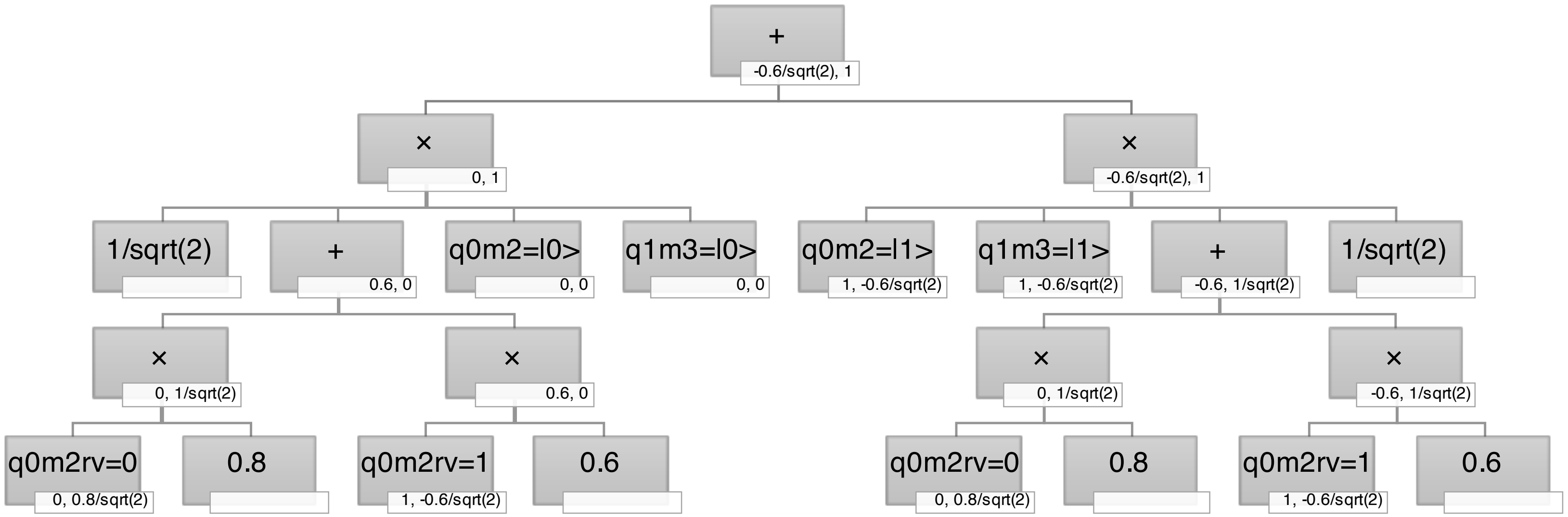


Figure 6: Simulation resource requirements vs. quantum circuit size for three quantum algorithms

Our toolchain: Bayesian network knowledge compilation for noisy quantum circuit simulation and sampling

1. Noisy quantum circuits to Bayesian network
2. Bayesian networks to conjunctive normal form (CNF)
3. CNF to arithmetic circuit (AC)
4. ***Exact inference on AC for quantum circuit simulation***
5. Gibbs sampling on AC to sample from final wavefunction

Exact inference on AC for quantum circuit simulation



- Quantum simulation becomes tree traversal on AC

Exact inference on AC for quantum circuit simulation

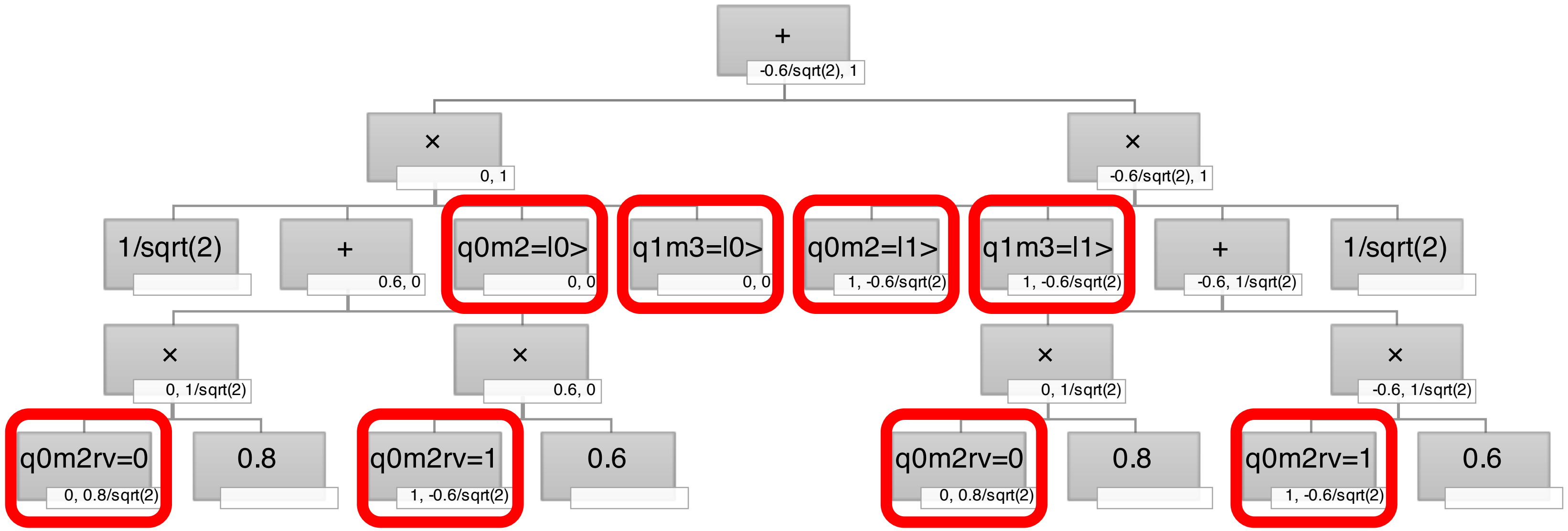


Table 9: Upward pass for finding amplitudes

q0m2rv	q0m2	q1m3	amplitude	density matrix component
0	$\lvert 0\rangle$	$\lvert 0\rangle$	$0.8\frac{1}{\sqrt{2}}$	$0.64\begin{bmatrix}+1/2 & 0 & 0 & +1/2\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ +1/2 & 0 & 0 & +1/2\end{bmatrix}$
0	$\lvert 0\rangle$	$\lvert 1\rangle$	0	
0	$\lvert 1\rangle$	$\lvert 0\rangle$	0	
0	$\lvert 1\rangle$	$\lvert 1\rangle$	$0.8\frac{1}{\sqrt{2}}$	
1	$\lvert 0\rangle$	$\lvert 0\rangle$	$0.6\frac{1}{\sqrt{2}}$	$0.36\begin{bmatrix}+1/2 & 0 & 0 & -1/2\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ -1/2 & 0 & 0 & +1/2\end{bmatrix}$
1	$\lvert 0\rangle$	$\lvert 1\rangle$	0	
1	$\lvert 1\rangle$	$\lvert 0\rangle$	0	
1	$\lvert 1\rangle$	$\lvert 1\rangle$	$0.6\frac{-1}{\sqrt{2}}$	

- Quantum simulation becomes tree traversal on AC
- ***Quantum measurement outcomes are probabilistic evidence***

Exact inference on AC for quantum circuit simulation

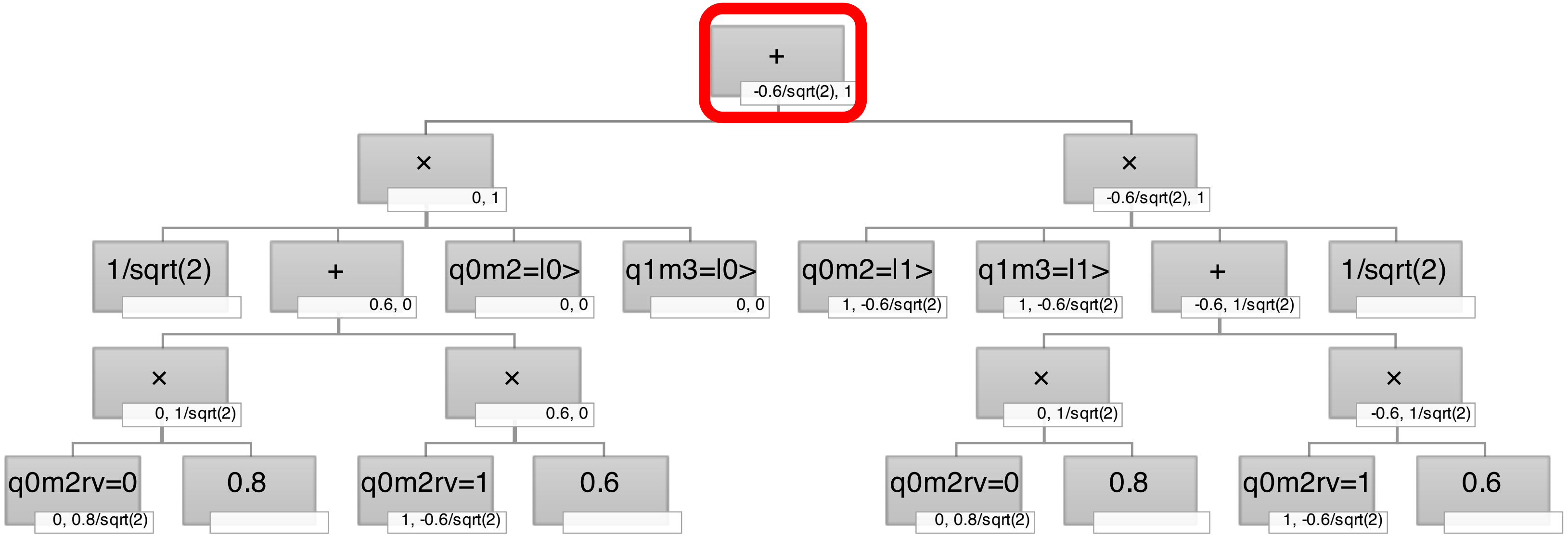


Table 9: Upward pass for finding amplitudes

q0m2rv	q0m2	q1m3	amplitude	density matrix component
0	$\|0\rangle$	$\|0\rangle$	$0.8\frac{1}{\sqrt{2}}$	$0.64\begin{bmatrix}+1/2 & 0 & 0 & +1/2\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ +1/2 & 0 & 0 & +1/2\end{bmatrix}$
0	$\|0\rangle$	$\|1\rangle$	0	
0	$\|1\rangle$	$\|0\rangle$	0	
0	$\|1\rangle$	$\|1\rangle$	$0.8\frac{1}{\sqrt{2}}$	
1	$\|0\rangle$	$\|0\rangle$	$0.6\frac{1}{\sqrt{2}}$	$0.36\begin{bmatrix}+1/2 & 0 & 0 & -1/2\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ -1/2 & 0 & 0 & +1/2\end{bmatrix}$
1	$\|0\rangle$	$\|1\rangle$	0	
1	$\|1\rangle$	$\|0\rangle$	0	
1	$\|1\rangle$	$\|1\rangle$	$0.6\frac{-1}{\sqrt{2}}$	

- Quantum simulation becomes tree traversal on AC
- Quantum measurement outcomes are probabilistic evidence
- ***Amplitude for given outcome comes from root node***

Our toolchain: Bayesian network knowledge compilation for noisy quantum circuit simulation and sampling

1. Noisy quantum circuits to Bayesian network
2. Bayesian networks to conjunctive normal form (CNF)
3. CNF to arithmetic circuit (AC)
4. Exact inference on AC for quantum circuit simulation
5. ***Gibbs sampling on AC to sample from final wavefunction***

Gibbs sampling on AC to sample from final wavefunction

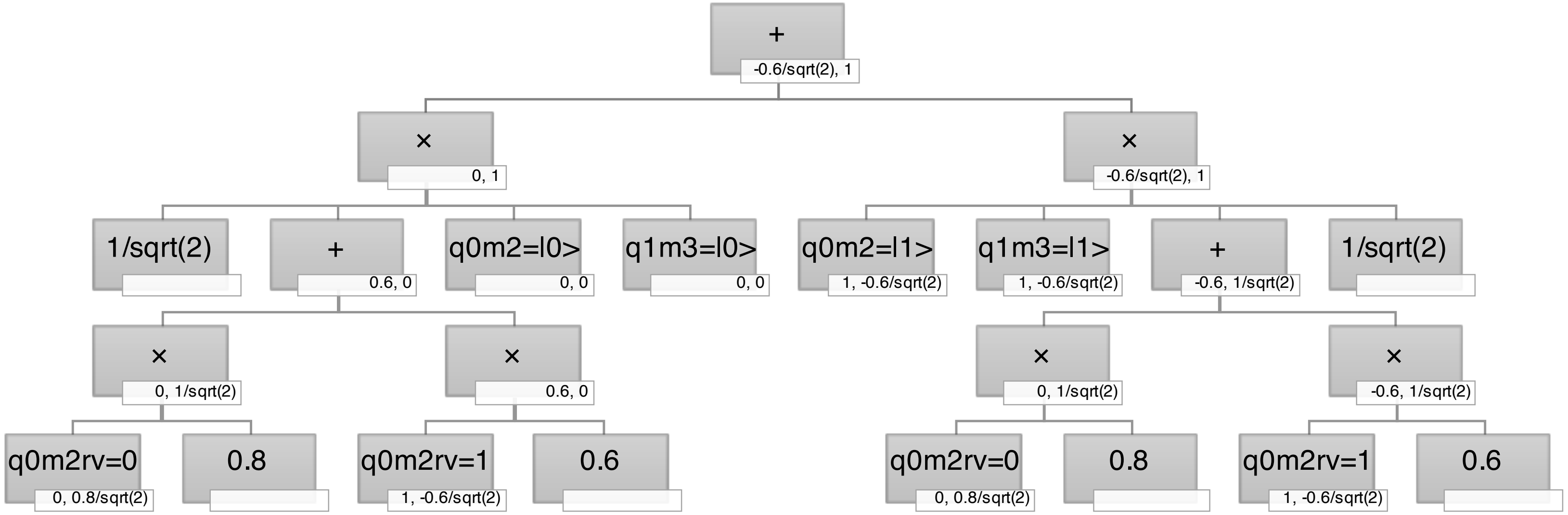
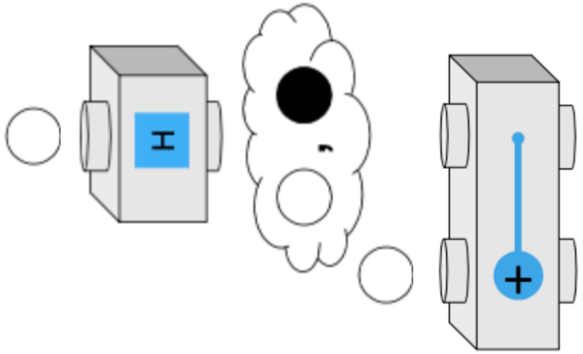


Table 10: Downward pass for finding derivatives for Gibbs sampling MCMC

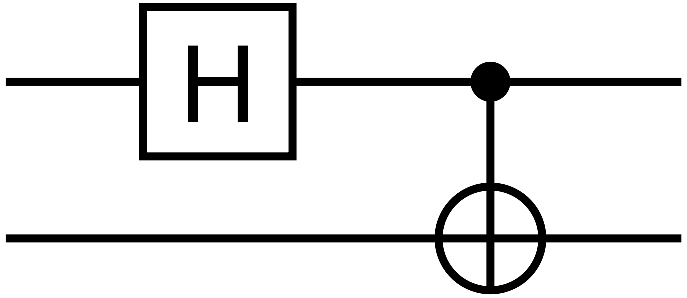
	q0m2rv	q0m2	q1m3	amplitude
Present sample	1	$\vert 1\rangle$	$\vert 1\rangle$	$0.6\frac{-1}{\sqrt{2}}$
Gibbs sample noise	0	$\vert 1\rangle$	$\vert 1\rangle$	$0.8\frac{1}{\sqrt{2}}$
Gibbs sample qubits	1	$\vert 0\rangle$	$\vert 1\rangle$	0
Gibbs sample qubits	1	$\vert 1\rangle$	$\vert 0\rangle$	0

Schrödinger quantum circuit simulation

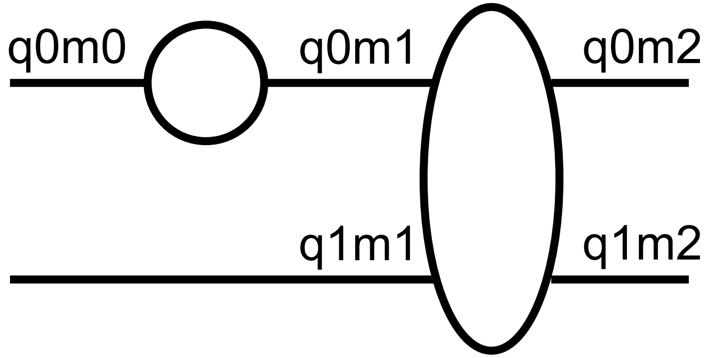


$$\mathrm{CNOT}(\mathrm{H} \otimes \mathrm{I}|00\rangle) = \mathrm{CNOT}(\mathrm{H}|0\rangle \otimes \mathrm{I}|0\rangle) = \mathrm{CNOT}\begin{bmatrix} \frac{1}{\sqrt{2}}\begin{bmatrix}1\\0\end{bmatrix} \\ \frac{1}{\sqrt{2}}\begin{bmatrix}1\\0\end{bmatrix} \end{bmatrix} = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & & 1 \\ & & 1 & \end{bmatrix}\begin{bmatrix} 1/\sqrt{2} \\ 0 \\ 1/\sqrt{2} \\ 0 \end{bmatrix} = \begin{bmatrix} 1/\sqrt{2} \\ 0 \\ 0 \\ 1/\sqrt{2} \end{bmatrix} = \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle$$

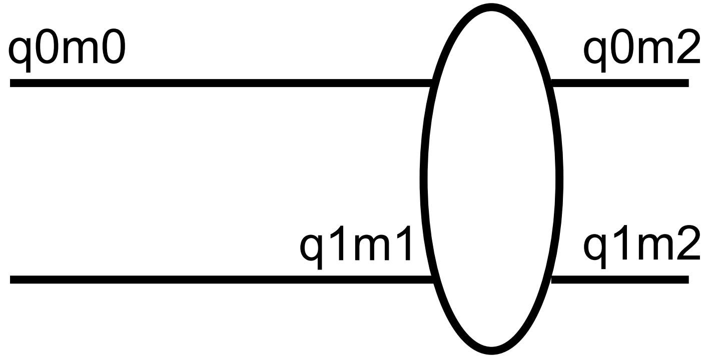
Feynman quantum circuit simulation



	q0m0=$\lvert\mathbf{0}\rangle$	q0m0=$\lvert\mathbf{1}\rangle$
q0m1=$\lvert\mathbf{0}\rangle$	$\frac{1}{\sqrt{2}}$	$\frac{1}{\sqrt{2}}$
q0m1=$\lvert\mathbf{1}\rangle$	$\frac{1}{\sqrt{2}}$	$\frac{-1}{\sqrt{2}}$



		q0m1=$\lvert\mathbf{0}\rangle$		q0m1=$\lvert\mathbf{1}\rangle$	
		q1m1=$\lvert\mathbf{0}\rangle$	q1m1=$\lvert\mathbf{1}\rangle$	q1m1=$\lvert\mathbf{0}\rangle$	q1m1=$\lvert\mathbf{1}\rangle$
q0m2=$\lvert\mathbf{0}\rangle$	q1m2=$\lvert\mathbf{0}\rangle$	1	0	0	0
	q1m2=$\lvert\mathbf{1}\rangle$	0	1	0	0
q0m2=$\lvert\mathbf{1}\rangle$	q1m2=$\lvert\mathbf{0}\rangle$	0	0	0	1
	q1m2=$\lvert\mathbf{1}\rangle$	0	0	1	0



		q0m0=$\lvert\mathbf{0}\rangle$		q0m0=$\lvert\mathbf{1}\rangle$	
		q1m1=$\lvert\mathbf{0}\rangle$	q1m1=$\lvert\mathbf{1}\rangle$	q1m1=$\lvert\mathbf{0}\rangle$	q1m1=$\lvert\mathbf{1}\rangle$
q0m2=$\lvert\mathbf{0}\rangle$	q1m2=$\lvert\mathbf{0}\rangle$	$1/\sqrt{2}$	0	$1/\sqrt{2}$	0
	q1m2=$\lvert\mathbf{1}\rangle$	0	$1/\sqrt{2}$	0	$1/\sqrt{2}$
q0m2=$\lvert\mathbf{1}\rangle$	q1m2=$\lvert\mathbf{0}\rangle$	0	$1/\sqrt{2}$	0	$-1/\sqrt{2}$
	q1m2=$\lvert\mathbf{1}\rangle$	$1/\sqrt{2}$	0	$-1/\sqrt{2}$	0

Where we are going.

What are quantum variational algorithms?

- Why are they different and important?

What is quantum circuit simulation?

- Why are the conventional techniques insufficient?

<u>How do we represent quantum circuits as logic formulas?</u>

- Why does it help with variational algorithm simulation, and by how much?

Where we are going.

What are quantum variational algorithms?

- Why are they different and important?

What is quantum circuit simulation?

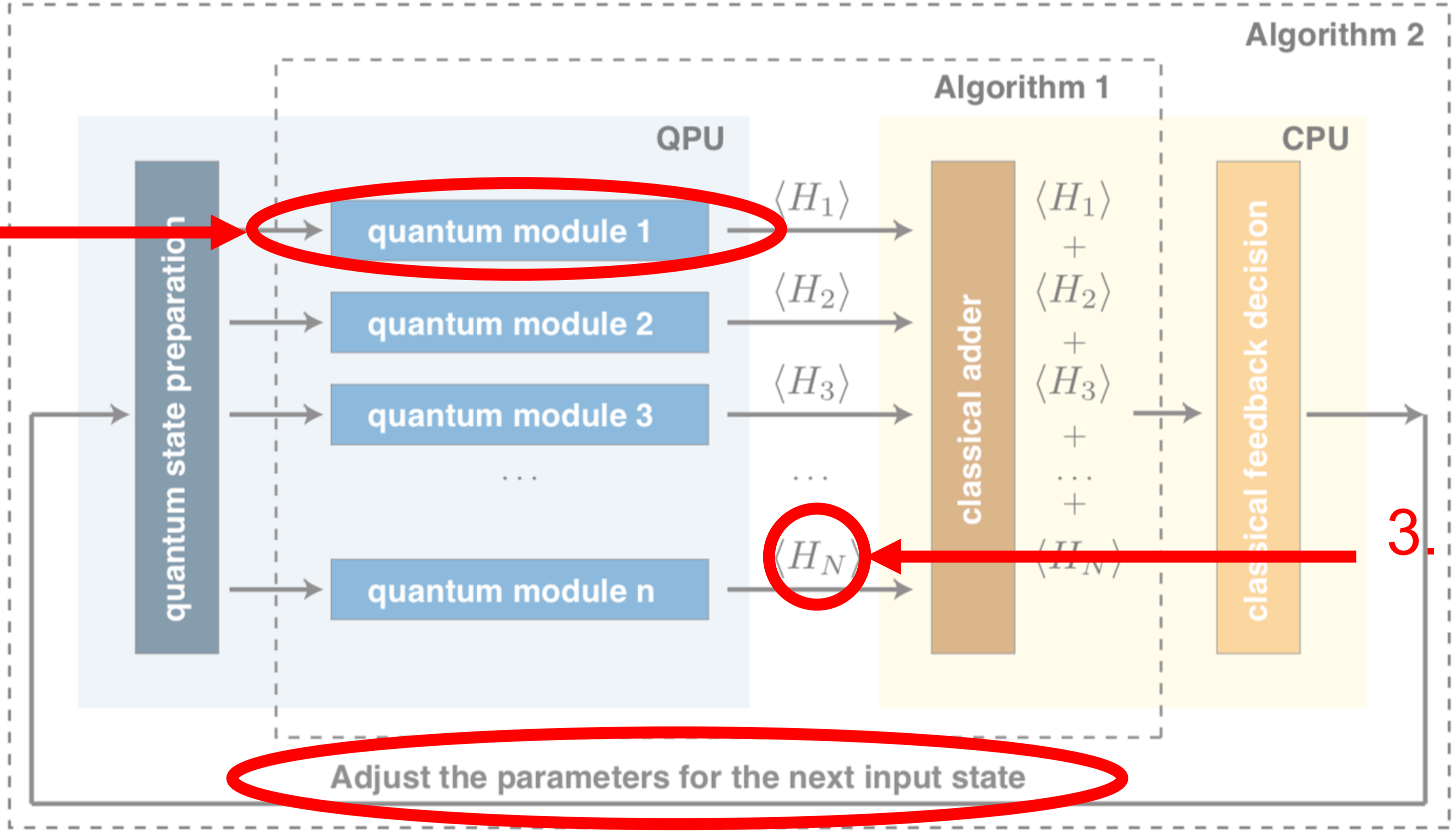
- Why are the conventional techniques insufficient?

How do we represent quantum circuits as logic formulas?

- *Why does it help with variational algorithm simulation, and by how much?*

The unique challenge of simulating noisy variational algorithms

1. Needs to simulate noise, and quantum circuits are wide but shallow



3. Only need samples, not full wavefunctions.

2. Require repeated simulation with different parameters

Image source: Peruzzo et al., 2013

Our toolchain: Bayesian network knowledge compilation for noisy quantum circuit simulation and sampling

1. Noisy quantum circuits to Bayesian network
2. Bayesian networks to conjunctive normal form (CNF)
3. CNF to arithmetic circuit (AC)
4. Exact inference on AC for quantum circuit simulation
5. Gibbs sampling on AC to sample from final wavefunction

1. Needs to simulate noise
2. Repeated simulation with different parameters
3. Only need samples, not full wavefunctions

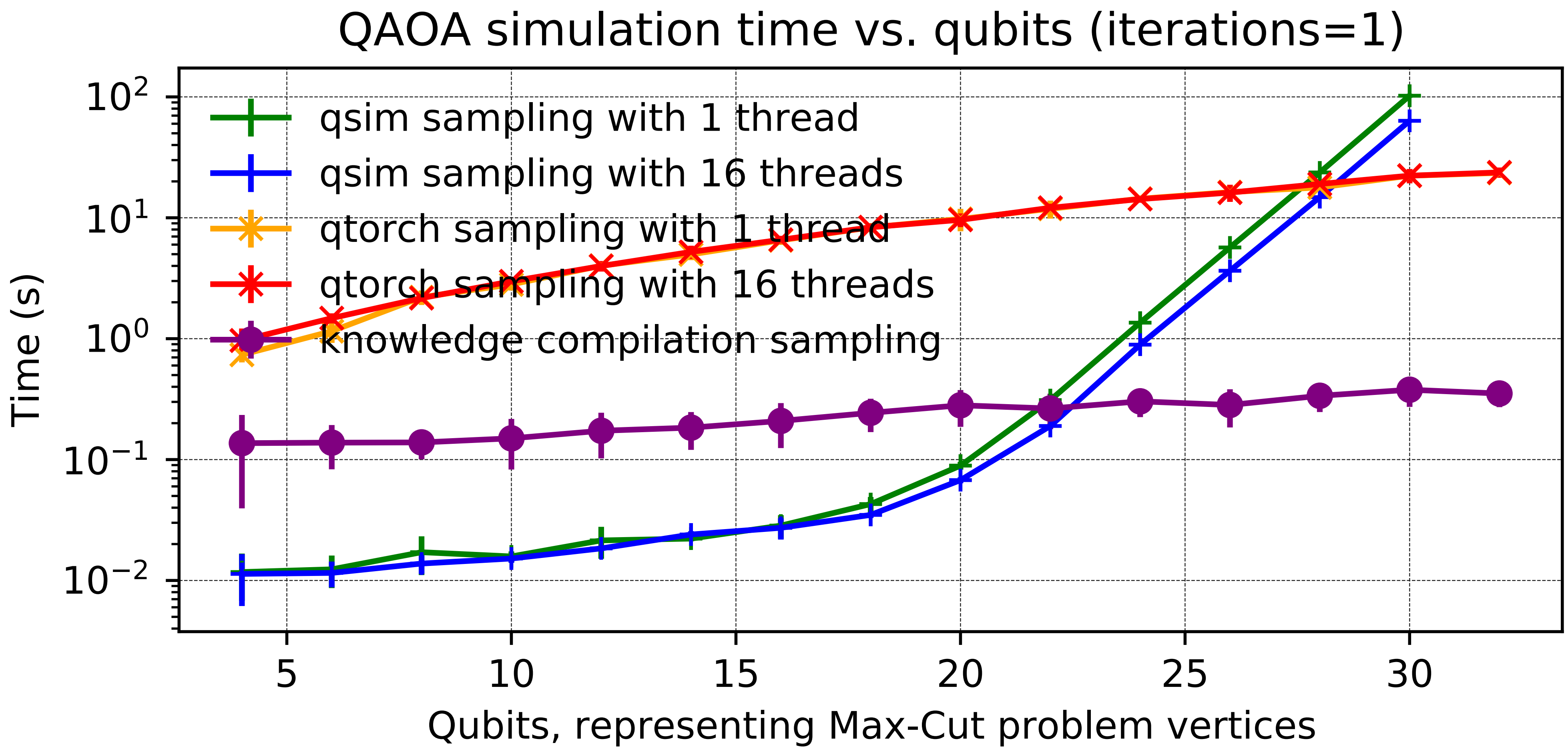
Result 1: It works!

With minimal modification, knowledge compilation exact inference can be repurposed for quantum simulation

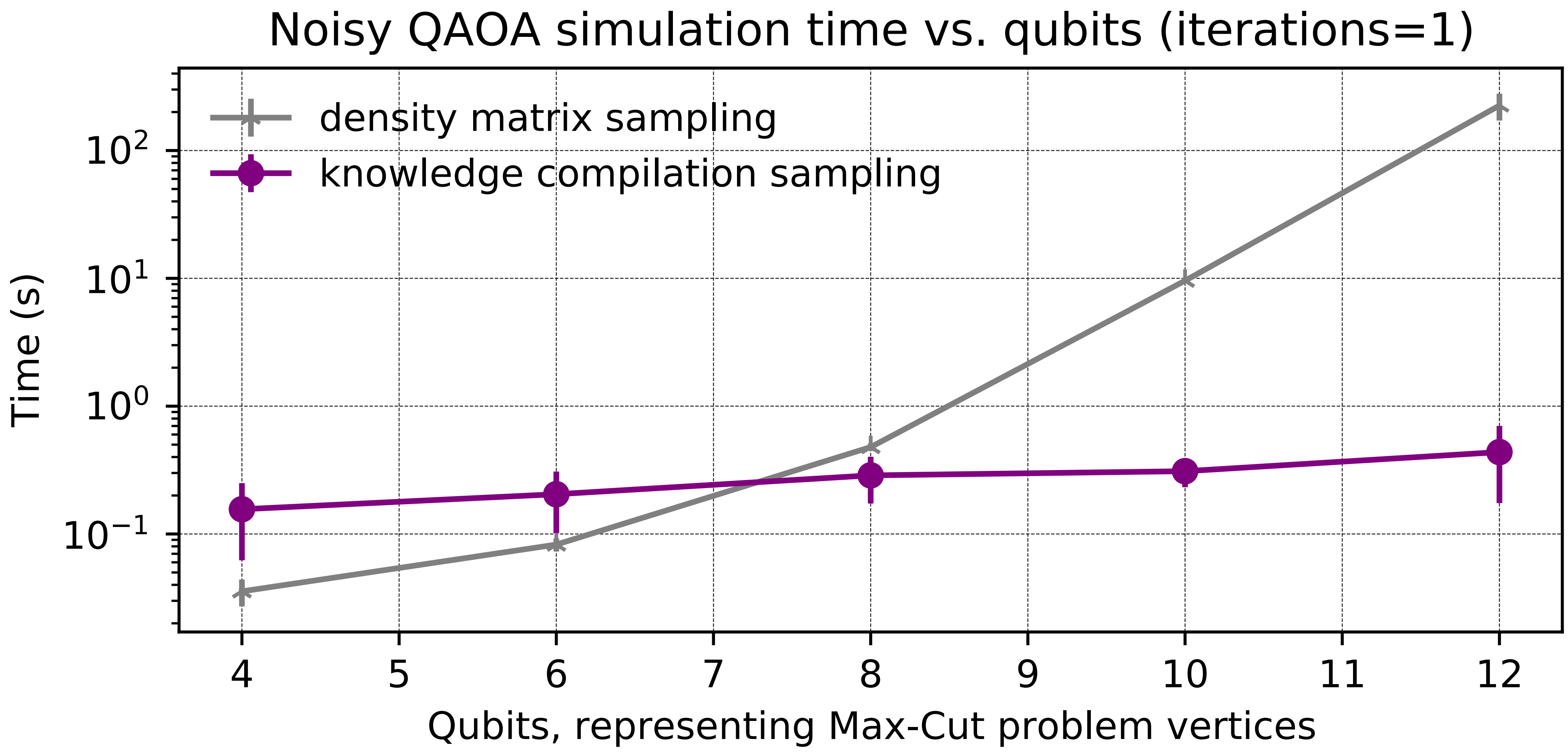
- Can accurately simulate Pauli gates, CNOT, CZ, phase kickback, Toffoli, CHSH protocol, Deutsch-Jozsa, Bernstein-Vazirani, hidden shift, quantum Fourier transform, Shor’s, Grover’s…

- Passes Google Cirq’s suite of test harness for quantum simulators

Result 2: Ideal circuit simulation



Result 2: Noisy circuit simulation



What this talk was about:

Using classical probabilistic inference techniques as an abstraction for quantum computing.

- A new way to represent noisy quantum circuits as probabilistic graphical models.
- A new way to encode quantum circuits as conjunctive normal forms and arithmetic circuits.
- A new way to manipulate quantum circuits using logical equation satisfiability solvers.
- Improved simulation and sampling performance for important near-term quantum algorithms.

Where we have gone:

What are quantum variational algorithms?

- Why are they different and important?

What is quantum circuit simulation?

- Why are the conventional techniques insufficient?

How do we represent quantum circuits as logic formulas?

- Why does it help with variational algorithm simulation, and by how much?

Thank you to my collaborators

Margaret Martonosi; Princeton

Steven Holtzen, Todd Millstein, Guy Van den Broeck; UCLA

Members of the EPiQC team

- This work is funded in part by EPiQC, an NSF Expedition in Computing, under grant 1730082.

Broader research agenda: new representations for quantum computing

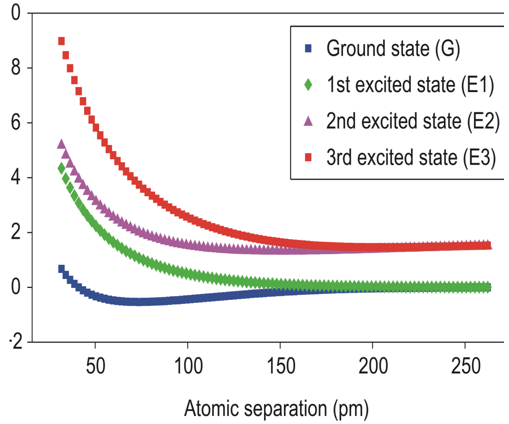
Schrödinger: state vectors and density matrices

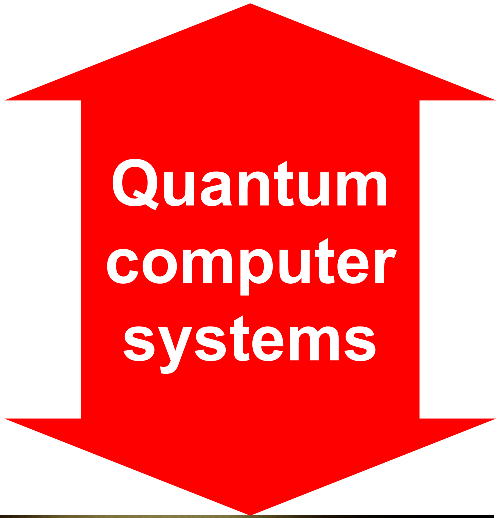
Heisenberg: stabilizer formalism

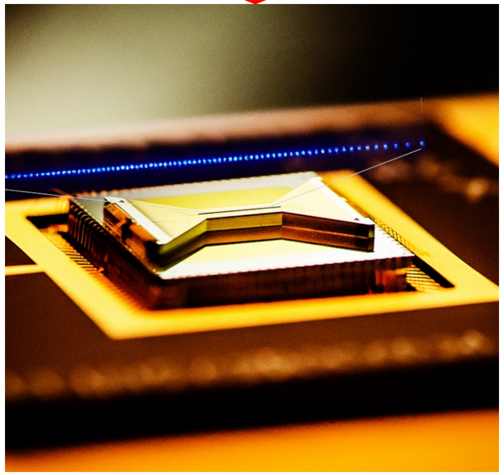
Feynman: tensor-network path sums

Logical satisfiability equations (this work; new?)

Binary decision diagrams (new?)







Awe-inspiring quantum algorithms

Chemistry simulations from governing equations	**Shor's algorithm for factoring integers**	**Hundreds more near-term and far-future algorithms**
Quantum computers as quantum mechanics simulator	Surpasses any known classical algorithm	QuantumAlgorithmZoo.org

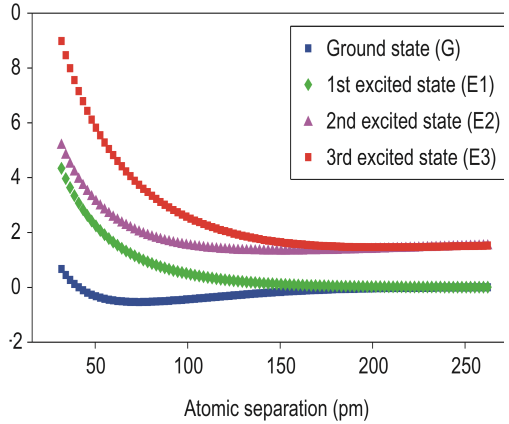
Quantum software-hardware gap

Quantum software frameworks	**Quantum programming languages**	**Quantum architectures and microarchitectures**
Simulators. Open source frameworks. Cloud accessible quantum prototypes.	Higher level programming abstractions. Debuggers. *Intermediate representations.*	Universal logical gate sets. Optimized place and route. Analog-digital quantum interface.

Now-viable quantum prototypes

Superconducting qubits	**Trapped ion qubits**	**Dozens of candidate qubit technologies**
IBM, Google, Rigetti, …	IonQ, UMD, Honeywell, …	May yet surpass current leaders in capacity and reliability

Image sources: Lanyon and Whitfield et al., 2010, IonQ



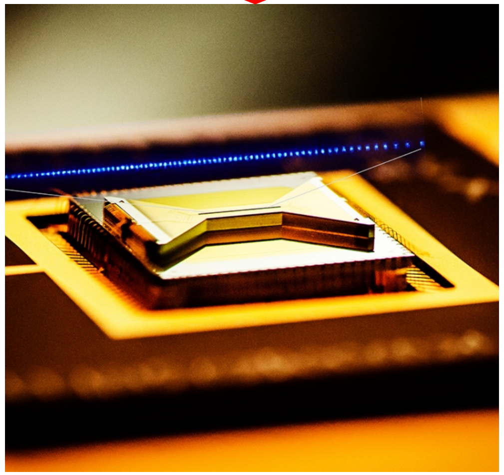
Awe-inspiring quantum algorithms

Chemistry simulations from governing equations	Shor's algorithm for factoring integers	Hundreds more near-term and far-future algorithms
Quantum computers as quantum mechanics simulator	Surpasses any known classical algorithm	QuantumAlgorithmZoo.org

Quantum computer systems

Quantum software-hardware gap

Quantum software frameworks	Quantum programming languages	Quantum architectures and microarchitectures
Simulators. Open source frameworks. Cloud accessible quantum prototypes.	Higher level programming abstractions. Debuggers. Intermediate representations.	Universal logical gate sets. Optimized place and route. Analog-digital quantum interface.



Now-viable quantum prototypes

Superconducting qubits	Trapped ion qubits	Dozens of candidate qubit technologies
IBM, Google, Rigetti, …	IonQ, UMD, Honeywell, …	May yet surpass current leaders in capacity and reliability

Image sources: Lanyon and Whitfield et al., 2010, IonQ

Course 16:198:672
Quantum Computing: Programs and Systems

Graduate seminar on latest developments in quantum computer engineering

What is quantum computer engineering??

- realizing quantum algorithms
- on prototype quantum computers

—a rapidly growing field!!

Goals of the course:

- read and discuss recent developments
- build foundation for you to pursue research or to be experts in industry

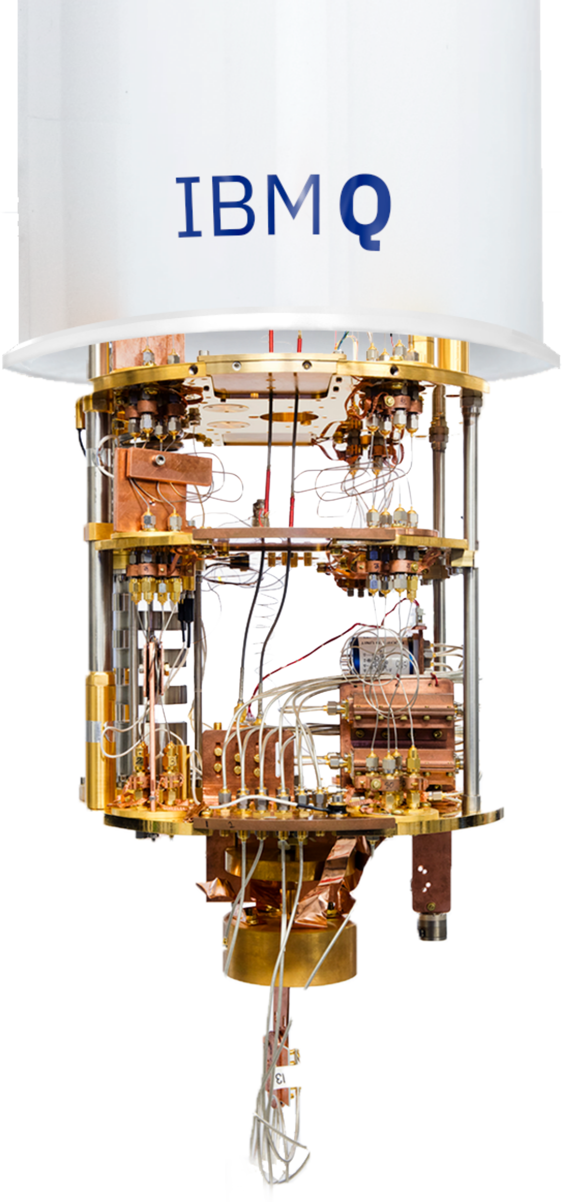


Image source: IBM

Course 16:198:672 Quantum Computing: Programs and Systems

- A systems view of quantum computer engineering
- Near-term intermediate-scale quantum algorithms
- Programming frameworks
- Emerging languages and representations
- Claims and counter claims for quantum advantage
- Extracting success
- Prototypes