Quantum Computing: Zero to Hero

Georgios Korpas

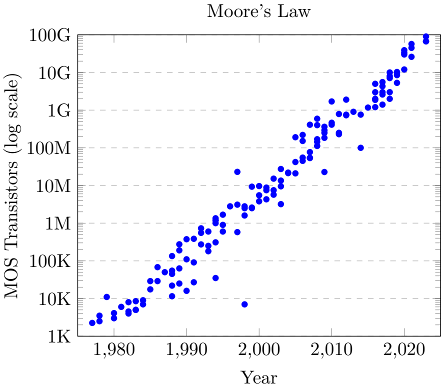
HSBC Lab, Innovation and Ventures, HSBC Bank Plc.
and
Dep. of Computer Science, Czech Technical University

Classical Computers

- Introduction of the Turing machine by Alan Turing [1936]
- von Neumann architecture (stored-program computers) [1947]
- Invention of the transistor at Bell Labs [1947]
- Integrated circuit (IC) [1958]
- Intel 4004 microprocessor [1971]
- IBM Personal Computer (PC) [1981]
- World Wide Web invented by Berners-Lee [1989]
- NVIDIA GeForce 256 GPU [1999]
- Multi-core processors [2000s]
- IBM Watson AI [2011]
- NVIDIA A100 GPU [2020]
- GPT-4, Stable Diffusion, etc [2023]

Limitations of Classical Computers

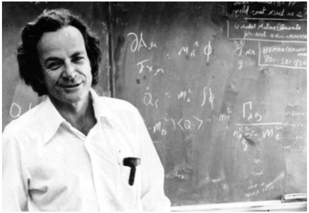
- Exponential growth of resources
- Inability to efficiently simulate quantum systems
- Limitations in solving optimization problems efficiently
- Energy consumption and heat dissipation
- Physical limitations of transistor scaling (Moore's Law)



Quantum Computing

- Manin (computable and uncomputable)
- Feynman (simulating nature)
- Benioff (quantum Turing machine)
- Deutsch (universal quantum computer)
- Schor (factoring)
- Grover (unstructured search algorithm)
- ...

"Nature isn't classical, dammit, and if you want to make a simulation of nature, you'd better make it quantum mechanical, and by golly it's a wonderful problem, because it doesn't look so easy."



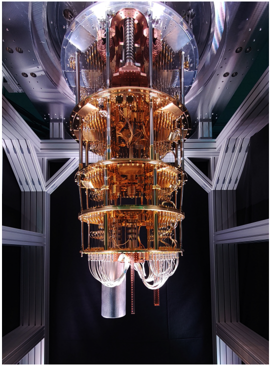
Paradigm Shift

QC is a paradigm of computation that utilizes quantum mechanics to perform calculations that are infeasible for classical computers.

Potential for **exponential** speedups!

Potential for unheard breakthroughs in science, medicine, engineering, ...

An IBM superconducting QPU



Quantum Computing: Theory 1

Church-Turing thesis

Any function "naturally to be regarded as computable" is computable by a (probabilistic) Turing machine.

Extended Church-Turing thesis

Any function naturally to be regarded as "efficiently computable" is efficiently computable by a (probabilistic) Turing machine.

Quantum Computing: Theory 1

Church-Turing thesis

Any function "naturally to be regarded as computable" is computable by a (probabilistic) Turing machine.

Extended Church-Turing thesis

Any function naturally to be regarded as "efficiently computable" is efficiently computable by a (probabilistic) Turing machine.

Quantum Extended Church-Turing thesis

Any function "naturally to be regarded as efficiently computable" is efficiently computable by a quantum Turing machine.

Quantum Computing: Theory 1

Church-Turing thesis

Any function “naturally to be regarded as computable” is computable by a (probabilistic) Turing machine.

Extended Church-Turing thesis

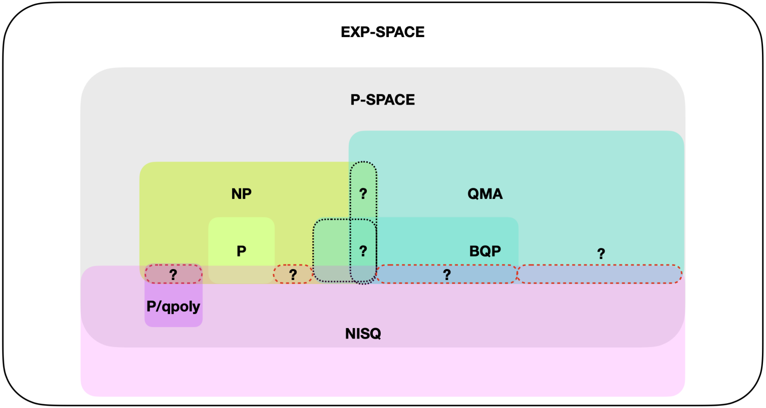
Any function naturally to be regarded as “efficiently computable” is efficiently computable by a (probabilistic) Turing machine.

Quantum Extended Church-Turing thesis

Any function “naturally to be regarded as efficiently computable” is efficiently computable by a quantum Turing machine.

What does this say about quantum computers?

Quantum Computing: Theory 2



Quantum Computing: Terminology

Quantum Supremacy

Experimental verification that a well defined computing task can be performed faster in a quantum computer than a classical computer.

Has it been achieved? Maybe.

Quantum Advantage

Experimental and repeated verification that a practical real world computing task can be performed faster in a quantum computer than a classical computer.

Has it been achieved? Probably not.

Google's Quantum Supremacy

nature

Article | Published: 23 October 2019

Quantum supremacy using a programmable superconducting processor

Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C. Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando G. S. L. Brandao, David A. Buell, Brian Burkett, Yu Chen, Zijun Chen, Ben Chiaro, Roberto Collins, William Courtney, Andrew Dunsworth, Edward Farhi, Brooks Foxen, Austin Fowler, Craig Gidney, Marissa Giustina, Rob Graff, Keith Guerin, … John M. Martinis + Show authors

Nature **574**, 505–510 (2019) | Cite this article

981k Accesses | **2990** Citations | **6298** Altmetric | Metrics

Bits vs. Qubits

	Bits	**Qubits**
Unit of information	$\{0, 1\}$	$\mathbb{CP}^1$
Operation	Deterministic	Stochastic
Processing	Classical gates	Unitary operators
Technology	Semiconductor based	Large number of modalities
Reversibility	No	Yes

Qubits

Quantum systems are discrete. A qubit is a 2-level system described as a vector on a Hilbert space $\mathcal{H} = \mathbb{C}^2$ with basis $\{|0\rangle, |1\rangle\}$.

Qubits

Quantum systems are discrete. A qubit is a 2-level system described as a vector on a Hilbert space $\mathcal{H} = \mathbb{C}^2$ with basis $\{|0\rangle, |1\rangle\}$.

The state of such a 2-level system is generically written as

$$|\psi\rangle = a|0\rangle + b|1\rangle$$

where $a, b \in \mathbb{C}$ are called **amplitudes** and $|a|^2 + |b|^1 = 1$, therefore the state is described as a Riemann sphere $\mathbb{CP}^1$.

Qubits

Quantum systems are discrete. A qubit is a 2-level system described as a vector on a Hilbert space $\mathcal{H} = \mathbb{C}^2$ with basis $\{|0\rangle, |1\rangle\}$.

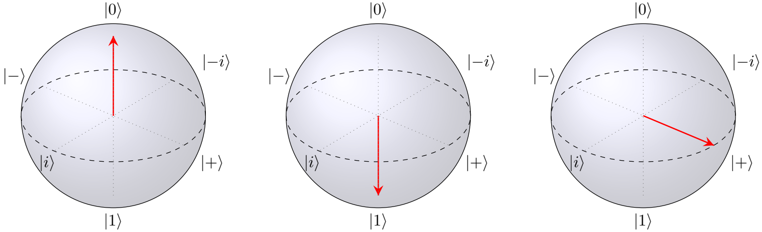
The state of such a 2-level system is generically written as

$$|\psi\rangle = a\,|0\rangle + b\,|1\rangle$$

where $a, b \in \mathbb{C}$ are called **amplitudes** and $|a|^2 + |b|^1 = 1$, therefore the state is described as a Riemann sphere $\mathbb{CP}^1$.

Amplitudes are **super important**: their squares give probabilities to measure the state in either of the basis states.

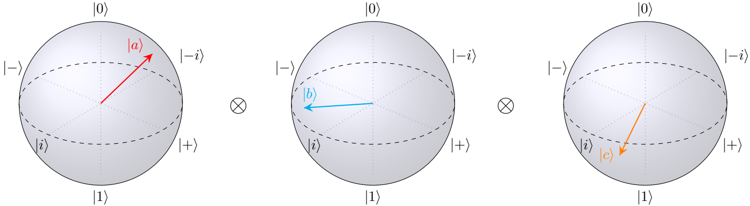
Bloch Sphere



The Bloch sphere $\cong \mathbb{CP}^1$ is an easy visualization of the state space of the qubit, the Hilbert space.

Each vector can be described by two angles, which encode the same information as the amplitude.

Multi-Qubit States



$$|\psi\rangle = |a\rangle \otimes |b\rangle \otimes |c\rangle = |abc\rangle$$

Superposition

In general a 3-qubit state is a superposition of all possible combinations (in the computational basis):

$$|\psi\rangle = \sum_{i} c_{ijk}|ijk\rangle$$

$$|\psi\rangle = c_{000}|000\rangle + c_{001}|001\rangle + \ldots + c_{110}|110\rangle + c_{111} \mid 111\rangle$$

This state is fully described by the $2^3 = 8$ amplitudes where

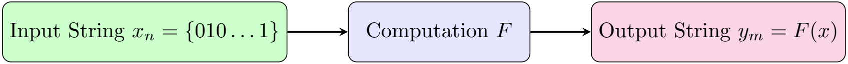
$$|c_{000}|^2+|c_{001}|^2+|c_{010}|^2+|c_{011}|^2+|c_{100}|^2+|c_{101}|^2+|c_{110}|^2+|c_{111}|^2 = 1$$

Classical Computation

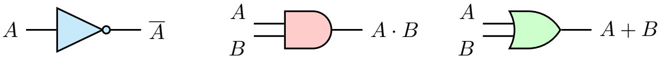
$$\text{Input String } x_n = \{010\ldots1\} \longrightarrow \text{Computation } F \longrightarrow \text{Output String } y_m = F(x)$$

- $x \in \{0,1\}^n$
- $y \in \{0,1\}^m$
- $F : \{0,1\}^n \to \{0,1\}^m$ is the composition of logical gates

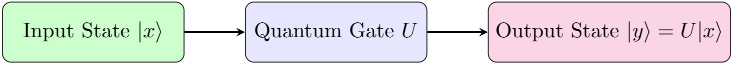
Classical Computation



- $x \in \{0,1\}^n$
- $y \in \{0,1\}^m$
- $F : \{0,1\}^n \to \{0,1\}^m$ is the composition of logical gates



Quantum Computation



- $|x\rangle \in \mathcal{H}_{\text{in}}$, the input Hilbert space of dimension n
- $|y\rangle \in \mathcal{H}_{\text{out}}$, the output Hilbert space of dimension n
- $U : \mathcal{H}_{\text{in}} \to \mathcal{H}_{\text{out}}$,

Quantum Computation

$$\text{Input State } |x\rangle \longrightarrow \text{Quantum Gate } U \longrightarrow \text{Output State } |y\rangle = U|x\rangle$$

- $|x\rangle \in \mathcal{H}_{\text{in}}$, the input Hilbert space of dimension n
- $|y\rangle \in \mathcal{H}_{\text{out}}$, the output Hilbert space of dimension n
- $U : \mathcal{H}_{\text{in}} \to \mathcal{H}_{\text{out}}, \quad U \in \mathrm{U}(n)$ the unitary group

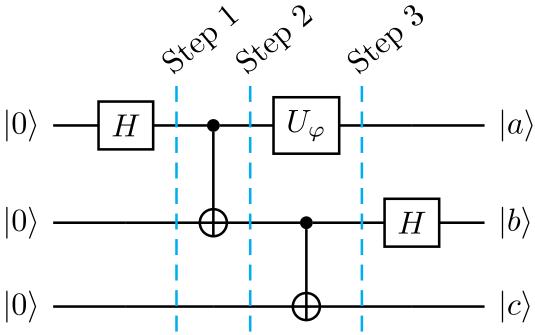
Quantum Computation

$$\boxed{\text{Input State } |x\rangle} \longrightarrow \boxed{\text{Quantum Gate } U} \longrightarrow \boxed{\text{Output State } |y\rangle = U|x\rangle}$$

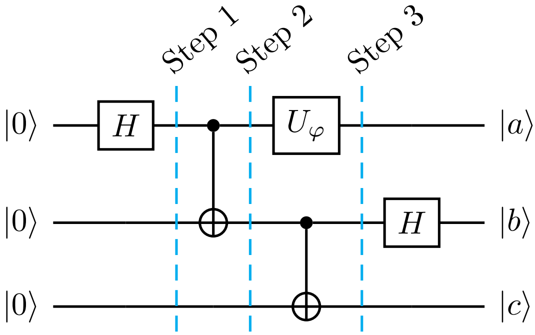
- $|x\rangle \in \mathcal{H}_{\text{in}}$, the input Hilbert space of dimension n
- $|y\rangle \in \mathcal{H}_{\text{out}}$, the output Hilbert space of dimension n
- $U : \mathcal{H}_{\text{in}} \to \mathcal{H}_{\text{out}}, \quad U \in \mathrm{U}(n)$ the unitary group
- $U = e^{-iHt}$, for $H \in \mathrm{Herm}(\mathbb{C}^n)$ the Hamiltonian or some other generator

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \quad \text{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Circuit Model

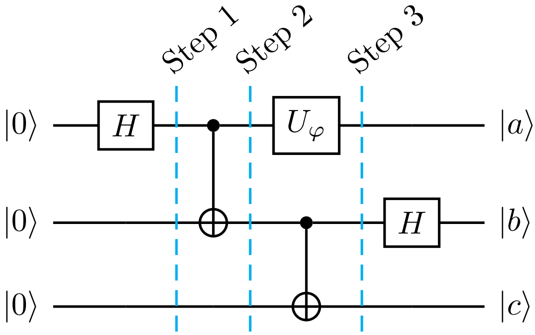


Circuit Model



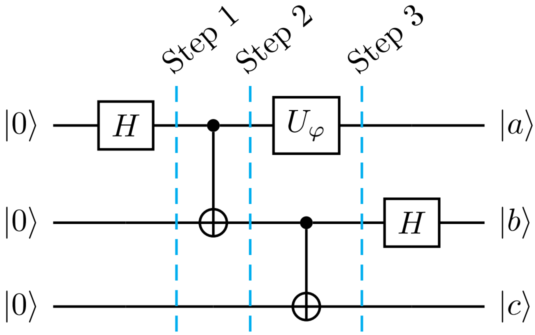
- Start with $|0\rangle^{\otimes 3} = |000\rangle$

Circuit Model



- Start with $|0\rangle^{\otimes 3} = |000\rangle$
- Apply $U = (\mathbf{1} \otimes H \otimes \mathbf{1})(U_\varphi \otimes \mathrm{CNOT})(H \otimes \mathbf{1} \otimes \mathbf{1})$

Circuit Model



- Start with $|0\rangle^{\otimes 3} = |000\rangle$
- Apply $U = (\mathbf{1} \otimes H \otimes \mathbf{1})(U_\varphi \otimes \mathrm{CNOT})(H \otimes \mathbf{1} \otimes \mathbf{1})$
- Obtain $|abc\rangle = U\,|000\rangle$

Quantum Algorithms

Out of this process, one can construct algorithms that do a lot of different things.

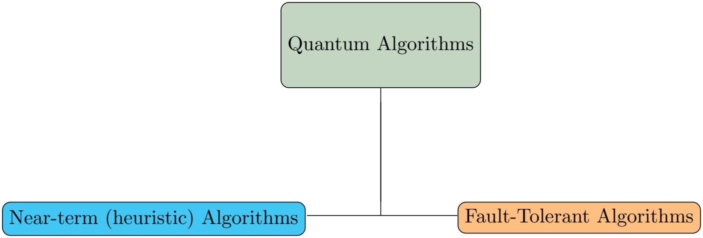
Quantum Algorithms

Out of this process, one can construct algorithms that do a lot of different things.

Algorithm	**Purpose**	**Speedup**
Schor's	Factoring	exp
Quantum Simulation	Simulating Nature	exp
HHL	Solving LSs	exp
Grover's	Search	poly
QAE	Amplify prob. of desired outcome	poly
VQAs	Optimization	unknown
Quantum Annealing	Optimization	unknown

NISQ vs FTEC

- Algorithms are implemented by quantum circuits on actual hardware
- The current era of hardware is called NISQ (issues: noisy, few qubits, ...)
- We are trying to arrive to the FTEC era (millions of qubits, error correction, ...)

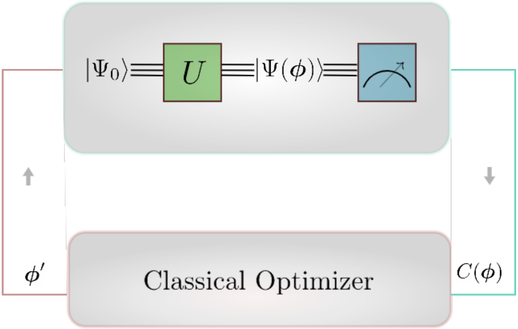


NISQ Era Algorithmic Enhancements

Currently, most people focus on NISQ compatible algorithms to be used already in real-world problems.

- **Optimization**: VQAs, Quantum Annealers, and more generally "Ising Machines"
- **Machine Learning**: Quantum Kernels, VQAs, QNNs, etc
- **Sampling**: QRNGs, Quassian Boson Sampling, Random Circuit Sampling, etc

Variational Quantum Algorithms



Variational Quantum Algorithms

- Consider some optimization problem (e.g. Knapsack):

$$\begin{aligned} \max \quad & \sum_{i=1}^{n} v_i x_i, \\ \text{s.t.} \quad & \sum_{i=1}^{n} w_i x_i \leq W, \end{aligned}$$

Variational Quantum Algorithms

- Then (if possible) formulate it as a QUBO:

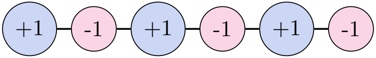
$$f(x) = \sum_{i}^{n} v_i x_i - \lambda_0 \left(\sum_{i}^{n} w_i x_i - W + \sum_{k=1}^{N} 2^{k-1} s_k \right)^2$$

Variational Quantum Algorithms

- Then map to an Ising Hamiltonian:

$$H(s) = \sum_{i}^{n} \left(\frac{v_i}{2}\right) \hat{s}_i + \sum_{i<j}^{n} \left(\frac{v_i v_j + \lambda_0 w_i w_j}{4}\right) \hat{s}_i \hat{s}_j$$

- Implement a **QAOA** certain type of VQA



NISQ Quantum Algorithms

However, more approaches exists:

Choose either:

- Quantum Annealing implementation (e.g. D-Wave)
- Digital Annealing implementation (e.g. Fujitsu)
- TensorTrain implementation (e.g. TerraQuantum)
- QAOA implementation (e.g. IBM, Quantinuum, ...)

Quantum Advantage

Applications of quantum computing:

- finance and logistic
- chemistry and material science
- automotive and aviation
- ...

McKinsey: “ the four industries likely to see the earliest economic impact from quantum computing—automotive, chemicals, financial services, and life sciences—stand to potentially gain up to $1.3 trillion in value by 2035.”

Problems

Here are (some) issues:

- FTEC computers still far ...

Problems

Here are (some) issues:

- FTEC computers still far ...
- "hybrid" solutions (TensorTrains, Digital Annealers) not fundamentaly different paradigm

Problems

Here are (some) issues:

- FTEC computers still far ...
- "hybrid" solutions (TensorTrains, Digital Annealers) not fundamentaly different paradigm
- VQAs NP-Hard to train!

arXiv > quant-ph > arXiv:2101.07267

Search...
Help | Advar

Quantum Physics

[Submitted on 18 Jan 2021 (v1), last revised 14 Apr 2022 (this version, v2)]

Training variational quantum algorithms is NP-hard

Lennart Bittel, Martin Kliesch

Variational quantum algorithms are proposed to solve relevant computational problems on near term quantum devices. Popular versions are variational quantum eigensolvers and quantum ap- proximate optimization algorithms that solve ground state problems from quantum chemistry and binary optimization problems, respectively. They are based on the idea of using a classical computer to train a parameterized quantum circuit. We show that the corresponding classical optimization problems are NP-hard. Moreover, the hardness is robust in the sense that, for every polynomial time algorithm, there are instances for which the relative error resulting from the classical optimization problem can be arbitrarily large assuming $P \neq NP$. Even for classically tractable systems composed of only logarithmically many qubits or free fermions, we show the optimization to be NP-hard. This elucidates that the classical optimization is intrinsically hard and does not merely inherit the hardness from the ground state problem. Our analysis shows that the training landscape can have many far from optimal persistent local minima. This means that gradient and higher order descent algorithms will generally converge to far from optimal solutions.

Problems (it's even worse?)

- Conjecture [G.K., Kungurtsev, Marecek, Zhu]:

Theorem 14 (Undecidability of VQAs). *Assuming that the Multivariate Diophantine Reducibility Conjecture 13 holds, the decision version of digitized VQA minimization (Problem 11) is undecidable for $L = 58$ layers.*

Theorem 14 then implies that there is no finite-time algorithm to solve the problem, i.e.,

Corollary 15 (Uncomputability of VQAs). *Assuming that the Multivariate Diophantine Reducibility Conjecture 13 holds, there exists no recursive function to decide digitized VQA minimization in the decision version (Problem 11) for $L = 58$ layers. Consequently, the optimization version is uncomputable for $L = 58$ layers.*

Problems (it's even worse?)

- Conjecture [G.K., Kungurtsev, Marecek, Zhu]:

Theorem 14 (Undecidability of VQAs). *Assuming that the Multivariate Diophantine Reducibility Conjecture 13 holds, the decision version of digitized VQA minimization (Problem 11) is undecidable for $L = 58$ layers.*

Theorem 14 then implies that there is no finite-time algorithm to solve the problem, i.e.,

Corollary 15 (Uncomputability of VQAs). *Assuming that the Multivariate Diophantine Reducibility Conjecture 13 holds, there exists no recursive function to decide digitized VQA minimization in the decision version (Problem 11) for $L = 58$ layers. Consequently, the optimization version is uncomputable for $L = 58$ layers.*

- It's fine; LLMs should not work either (I guess)

Problems (it's even worse?)

- Conjecture [G.K., Kungurtsev, Marecek, Zhu]:

Theorem 14 (Undecidability of VQAs). *Assuming that the Multivariate Diophantine Reducibility Conjecture 13 holds, the decision version of digitized VQA minimization (Problem 11) is undecidable for $L = 58$ layers.*

Theorem 14 then implies that there is no finite-time algorithm to solve the problem, i.e.,

Corollary 15 (Uncomputability of VQAs). *Assuming that the Multivariate Diophantine Reducibility Conjecture 13 holds, there exists no recursive function to decide digitized VQA minimization in the decision version (Problem 11) for $L = 58$ layers. Consequently, the optimization version is uncomputable for $L = 58$ layers.*

- It's fine; LLMs should not work either (I guess)
- Monthly "mini-breakthroughs" on arxiv/Nature/PRL

Problems (it's even worse?)

- Conjecture [G.K., Kungurtsev, Marecek, Zhu]:

Theorem 14 (Undecidability of VQAs). *Assuming that the Multivariate Diophantine Reducibility Conjecture 13 holds, the decision version of digitized VQA minimization (Problem 11) is undecidable for $L = 58$ layers.*

Theorem 14 then implies that there is no finite-time algorithm to solve the problem, i.e.,

Corollary 15 (Uncomputability of VQAs). *Assuming that the Multivariate Diophantine Reducibility Conjecture 13 holds, there exists no recursive function to decide digitized VQA minimization in the decision version (Problem 11) for $L = 58$ layers. Consequently, the optimization version is uncomputable for $L = 58$ layers.*

- It's fine; LLMs should not work either (I guess)
- Monthly "mini-breakthroughs" on arxiv/Nature/PRL
- Conjecture [G.K., Kungurtsev, Marecek, Zhu]: VQAs can decide undecidable languages

Problems (it's even worse?)

- Conjecture [G.K., Kungurtsev, Marecek, Zhu]:

Theorem 14 (Undecidability of VQAs). *Assuming that the Multivariate Diophantine Reducibility Conjecture 13 holds, the decision version of digitized VQA minimization (Problem 11) is undecidable for $L = 58$ layers.*

Theorem 14 then implies that there is no finite-time algorithm to solve the problem, i.e.,

Corollary 15 (Uncomputability of VQAs). *Assuming that the Multivariate Diophantine Reducibility Conjecture 13 holds, there exists no recursive function to decide digitized VQA minimization in the decision version (Problem 11) for $L = 58$ layers. Consequently, the optimization version is uncomputable for $L = 58$ layers.*

- It's fine; LLMs should not work either (I guess)
- Monthly "mini-breakthroughs" on arxiv/Nature/PRL
- Conjecture [G.K., Kungurtsev, Marecek, Zhu]: VQAs can decide undecidable languages
- Slow but steady adoption by top firms

What does a bank do with quantum?

There are several (other) use cases:

- Fraud detection (QML - NISQ/FTEC)
- Collateral optimization (VQAs - NISQ)
- Portfolio optimization (VQAs - NISQ)
- Risk Management: CVA (QAE - FTEC)
- Cybersecurity (Schor - FTEC)

Quantum Option Pricing

Quantum
the open journal for quantum science

CALL FOR EDITORS! PAPERS PERSPECTIVES

Option Pricing using Quantum Computers

Nikitas Stamatopoulos[1], Daniel J. Egger[2], Yue Sun[1], Christa Zoufal[2,3], Raban Iten[2,3], Ning Shen[1], and Stefan Woerner[2]

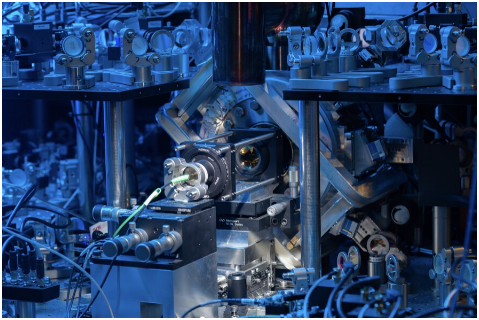
[1]Quantitative Research, JPMorgan Chase & Co., New York, NY, 10017
[2]IBM Quantum, IBM Research – Zurich
[3]ETH Zurich

Published:	2020-07-06, volume 4, page 291
Eprint:	arXiv:1905.02666v5
Doi:	https://doi.org/10.22331/q-2020-07-06-291
Citation:	Quantum 4, 291 (2020).

Hardware issues

One of the main problems with quantum computing is hardware.

There are **several** different modalities with many pros and cons.



Issues: scalability, noise reduction, error correction, ...

Talent Sparsity

Another problem with quantum computing is talent sparsity.

There are **several** needs across the industry

- quantum physicists,
- quantum algorithm developer,
- quantum transpilation engineers,
- software engineering,
- ...

Summary

Quantum computing is still a bet, to some extent.

Summary

Quantum computing is still a bet, to some extent.

While many achievements happen and landmarks are reached, we are yet to see the **game changing application** with significant **quantum advantage**...

Summary

Quantum computing is still a bet, to some extent.

While many achievements happen and landmarks are reached, we are yet to see the **game changing application** with significant **quantum advantage**...

Large institutions, governments, and corporations are convinced though. It's an exciting field that has the potential to make LLMs look like toys.

Topics not discussed

There are several "fundamental" topics that were not discussed in this talk including:

- initial quantum state preparation
- quantum optimal control and quantum SysId
- quantum compilation and circuit optimization
- important algorithms and algorithm development
- other universal models (quantum Turing machine, quantum walks, etc)
- hardware modalities and peculiarities
- quantum startups, business, VCs, etc
- ...

Thank you!

Find me on Linkedin:
https://www.linkedin.com/in/georgios-korpas/