Quantum AI:
The Road Ahead for AI Powered Testing

Dr. Tuhin Chattopadhyay

TUHIN AI ADVISORY

Tuhin Chattopadhyay, Ph.D.

Founder & CEO, Tuhin AI Advisory

An ISO 27001:2013 Certified Company

Top 10 Data Scientists In Analytics India Magazine
India's Top 10 Data Scientists - 2016

KamiKaze B2B Media Big Data, Analytics & Insights Summit
Analytics and Insight Leader of the Year - 2017

Inflection Awards NASSCOM CoE - IoT
Artificial Intelligence Leader of the Year - 2018

Zendesk Awards Digital CX Summit
Digital Transformation Leader of the Year - 2019

CIO Conclave & Awards 2020 UBS Forums
CTO of the Year - 2020

https://www.linkedin.com/in/tuhinai/

International Experience

- 20 years of experience in academia & industry – delivers analytics solutions to organizations across USA, Europe, Australia, Africa & South-East Asia
- Keynote speaker at international conferences like Next Big Tech Asia 17 in 2017 at Kuala Lumpur and Sports Analytics Africa in 2018 at Johannesburg
- Jury Member of data science competitions across Europe & USA

Thought Leadership

- Authored books & more than 30 publications in international journals & conferences
- Editor-in-Chief of International Journal of Business Analytics and Intelligence (IJBAI)
- Interview to DZone, USA – https://dzone.com/articles/coffee-with-a-data-scientist-tuhin

A Paradigm Shift

Current State

- Intelligent Test Automation

The Road Ahead

- Quantum AI

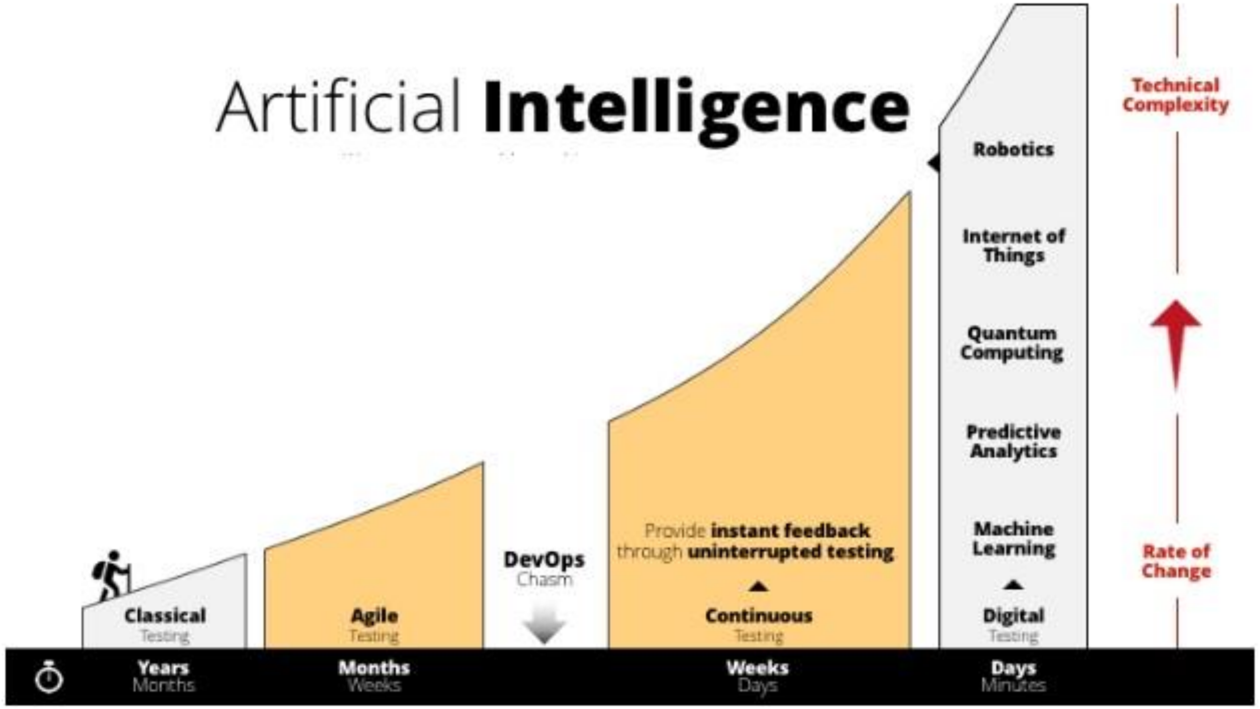


Image Source: https://sdtimes.com/ai/whats-beyond-continuous-testing-ai/

State of AI Test Automation

Integration of design time & run-time evolution

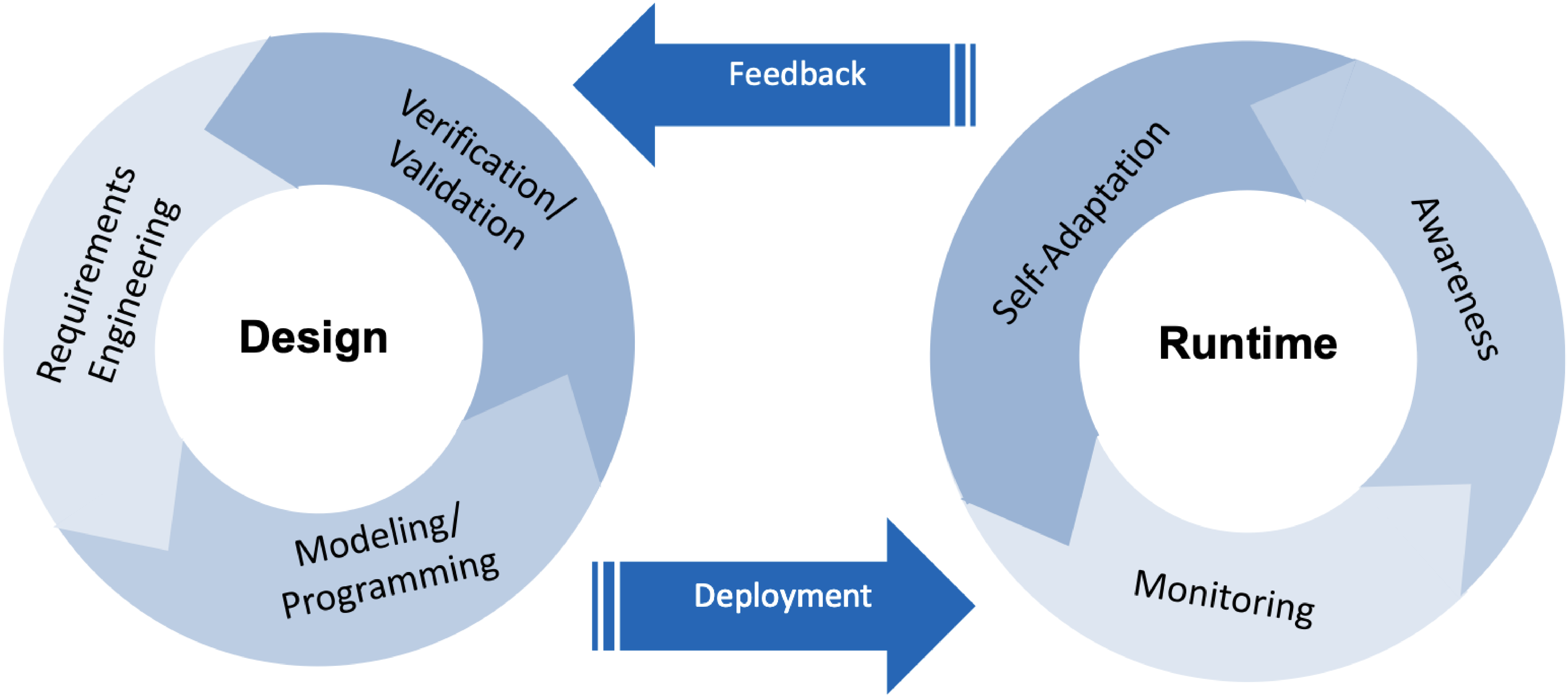


Image Source : https://doi.org/10.1145/3387940.3391469

Applying AI to Software Testing

AI-Driven Testing

- Developing AI **Tools** to test software

Testing AI Systems

- Devising **Methods** to test AI systems

Self-Testing Systems

- **Designing** software capable of **self-testing & self-healing**

Reference: https://www.perfecto.io/blog/ai-in-software-testing

Applying AI to Software Testing

Elastically scale functional, load & performance tests	Seamless integrations with existing CI/CD process	AI test cases update automatically
Intelligent Testing	Root Cause Analysis	Seamless Integrations

AI-Driven Testing Approaches

Differential Testing

- Comparing application versions overbuilds, classifying the differences, and learning from feedback on the classification

Visual Testing

- Leveraging image-based learning and screen comparisons to test the look and feel of an application

Declarative Testing

- Specifying the intent of a test in a natural or domain-specific language, and having the system figure out how to carry out the test

Self-healing Automation

- Auto-correcting element selection in tests when the UI changes

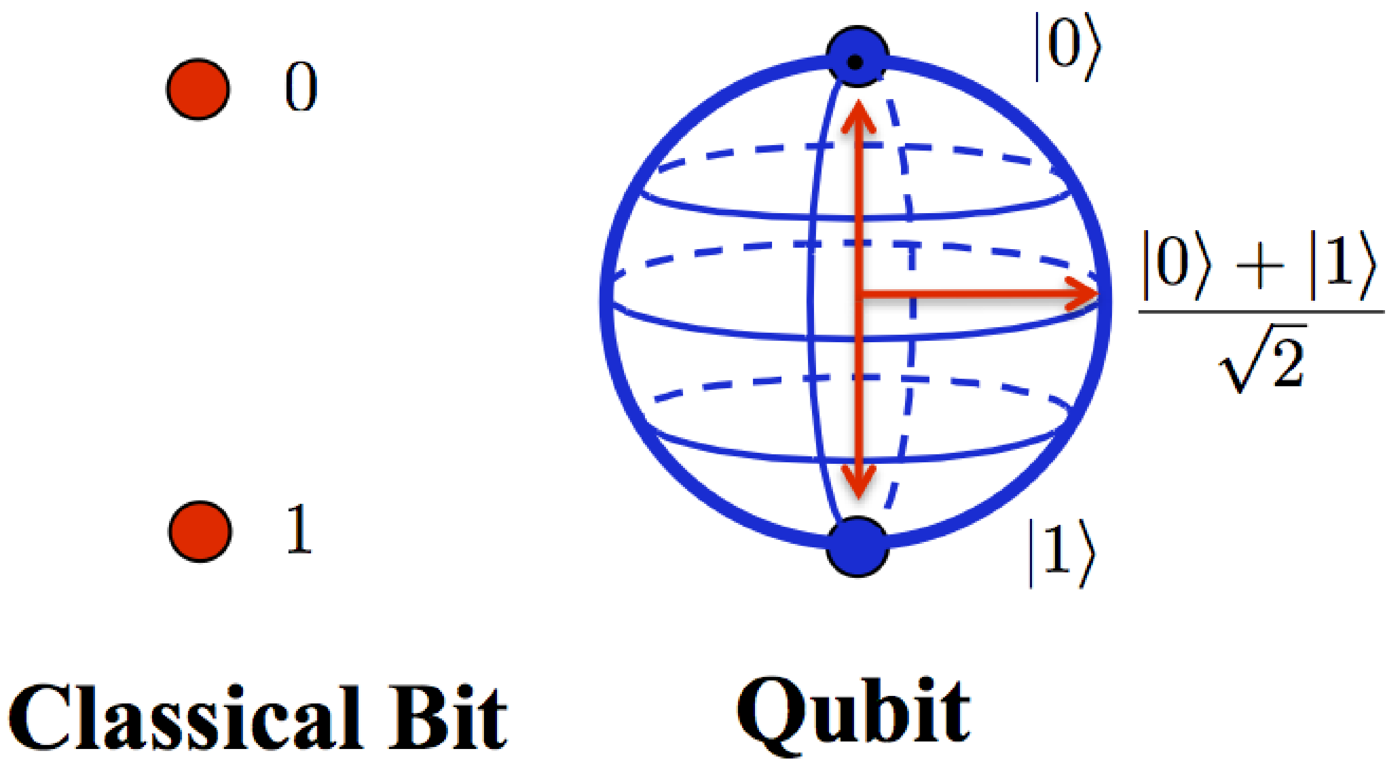
Reference: https://appsierra.com/ai-testing-in-software-testing/

Benefits of AI Powered Testing

Code-less	• No need to memorize any syntax.
Simplicity	• Easy to create a process through simple drag and drop.
Scalability	• It can be achieved by assigning work to multiple workstations.
Cost saving	• Huge reduction in cost as very minimal workforce is required.
Accuracy	• As the tasks are performed by the bots.
Productivity	• As it is automated, productivity will be very high.
Flexibility	• Test process does not depend on the type of software under test, whether it is web based, desktop application or mobile application.

Quantum AI: The Road Ahead

What is Quantum Computing



Quantum Computing	Classical Computing
Calculates with qubits, which can represent 0 and 1 at the same time	Calculates with transistors, which can represent either 0 or 1
Power increases exponentially in proportion to the number of qubits	Power increases in a 1:1 relationship with the number of transistors
Quantum computers have high error rates and need to be kept ultracold	Classical computers have low error rates and can operate at room temp
Well suited for tasks like optimization problems, data analysis, and simulations	Most everyday processing is best handled by classical computers

Image Sources:

1. Hussain, Zahid. (2016). Strengths and Weaknesses of Quantum Computing.. International Journal of Scientific and Engineering Research. 7.
2. https://www.cbinsights.com/research/quantum-computing-classical-computing-comparison-infographic/

How a Quantum Computer Works

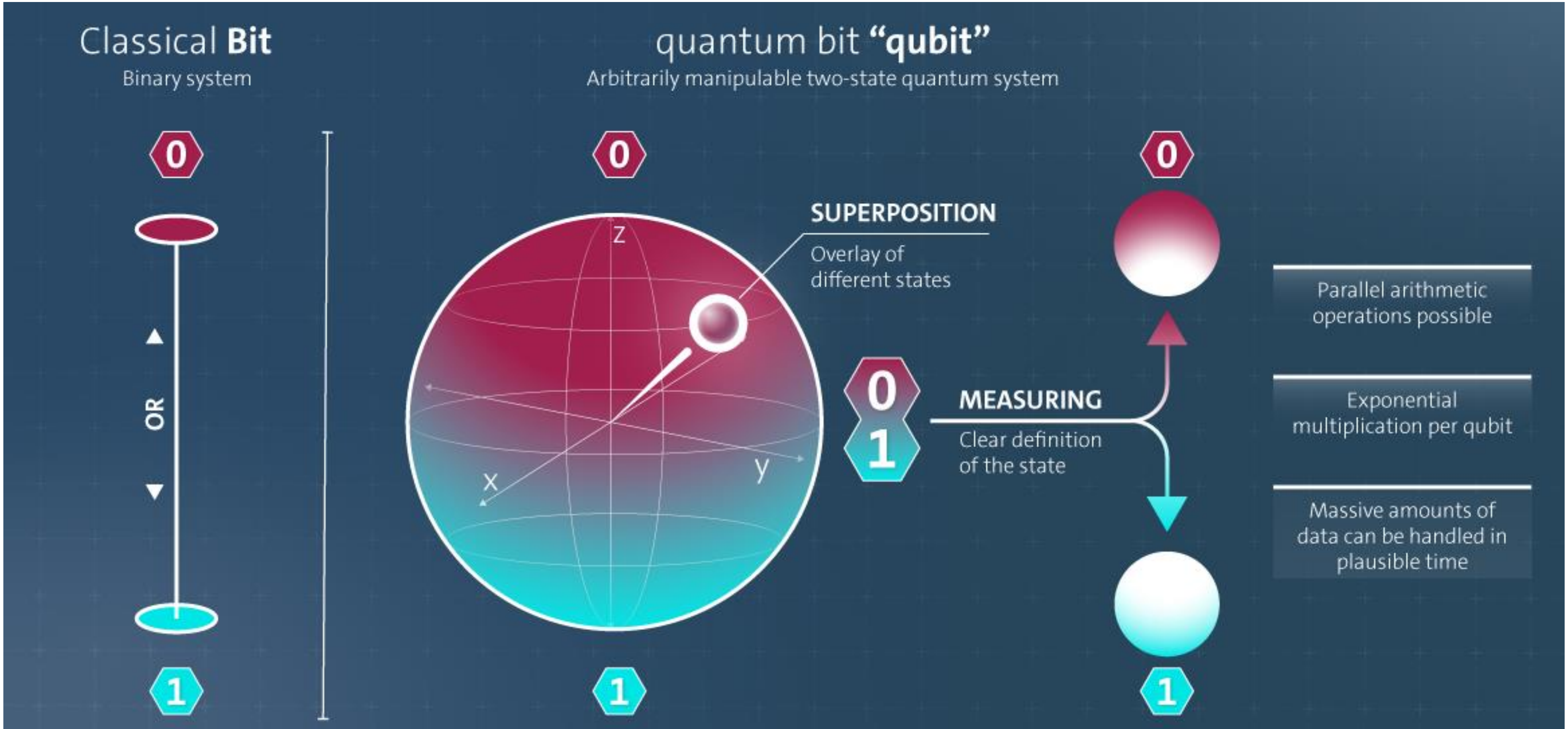


Image Source: https://www.volkswagenag.com/en/news/stories/2019/11/where-is-the-electron-and-how-many-of-them.html

Why is Quantum different?

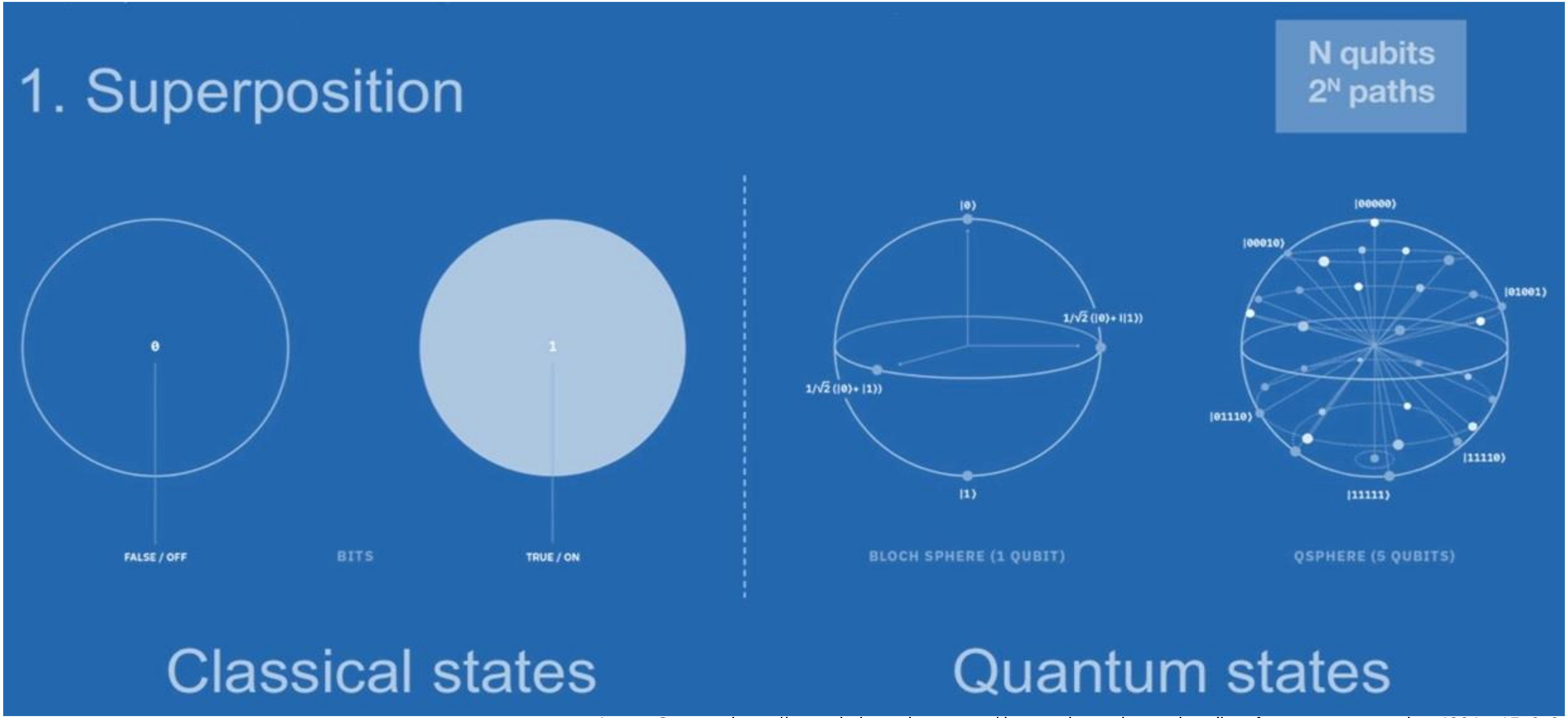


Image Source : https://towardsdatascience.com/the-need-promise-and-reality-of-quantum-computing-4264ce15c6c0

Why is Quantum different?

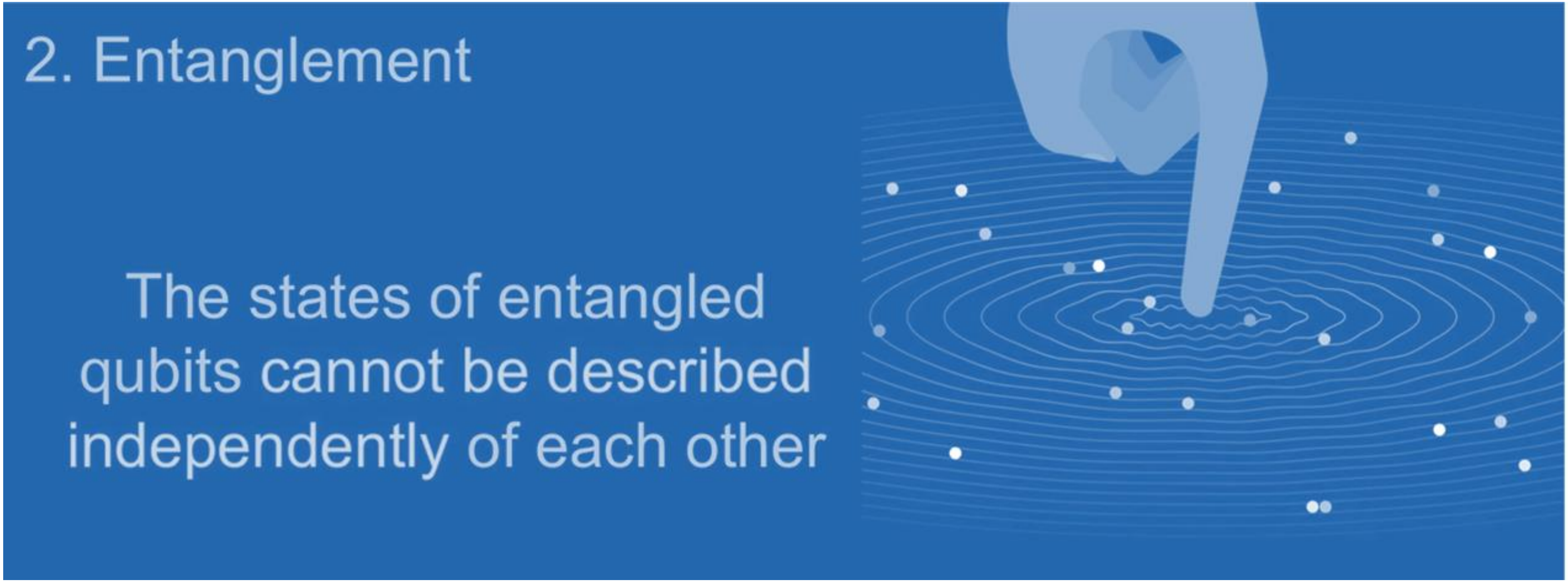
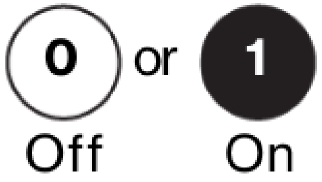


Image Source: https://towardsdatascience.com/the-need-promise-and-reality-of-quantum-computing-4264ce15c6c0

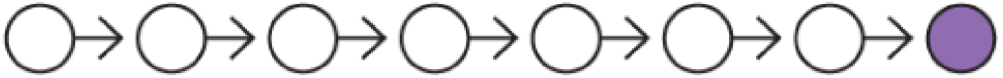
Why Quantum will be quicker

Your PC performs calculations using bits, units of information encoded electrically in either the off (0) or on (1) state.



A qubit, or quantum bit, can exist as 0 or 1 or both simultaneously, a trait known as superposition. Adding more qubits therefore increases the number of states that can be represented exponentially.

A traditional computer tries possible answers one by one until it finds the right one, a process that's far too slow for such complex problems.



Superposition is one of the properties of a quantum computer that enables it to work faster by considering many possibilities at once, sorting through sets of probable outcomes that converge on the correct answer.

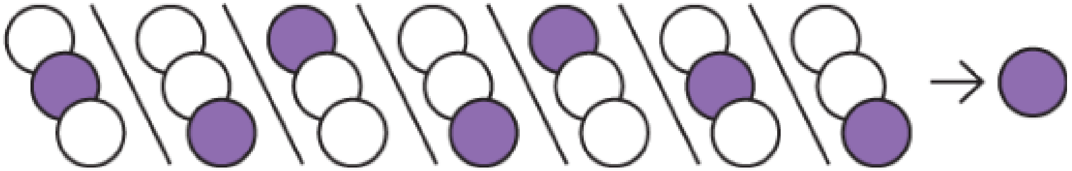


Image Source: https://www.bloomberg.com/news/articles/2019-10-26/why-quantum-computers-will-be-super-awesome-someday-quicktake

What is Quantum AI

Quantum AI is the use of quantum computing for computation of ML algorithms

- Thanks to computational advantages of quantum computing, quantum AI can help achieve results that are not possible to achieve with classical computers

Reference: https://research.aimultiple.com/quantum-ai/

How does Quantum AI Work

Convert quantum data to the quantum dataset

- Quantum data can be represented as a multi-dimensional array of numbers which is called as quantum tensors. TensorFlow processes these tensors in order to represent create a dataset for further use.

Choose quantum neural network models

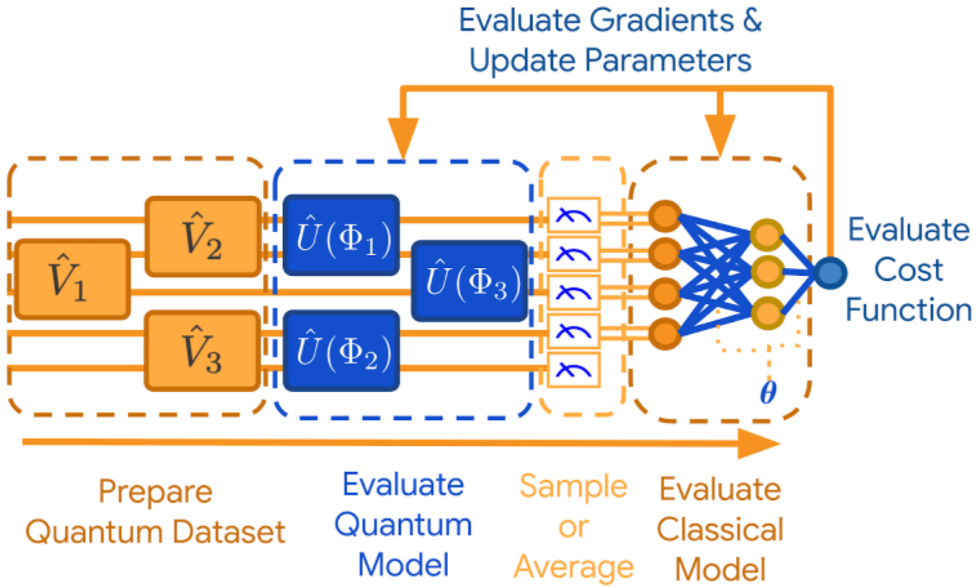
- Based on the knowledge of the quantum data structure, quantum neural network models are selected. The aim is to perform quantum processing in order to extract information hidden in an entangled state

Sample or Average

- Measurement of quantum states extracts classical information in the form of samples from the classical distribution. The values are obtained from the quantum state itself. TFQ provides methods for averaging over several runs involving steps (1) and (2).

Evaluate a classical neural networks model

- Since quantum data is now converted into classical data, deep learning techniques are used to learn the correlation between data.



Reference: https://research.aimultiple.com/quantum-ai/

Challenges of Quantum Powered Artificial Intelligence

Challenge 1 The Feedback Loop

- Replace the feedback loop around training (consisting of the tasks “Select Model/Policy”, “Train”, and “Assess QoS”) entirely with a quantum algorithm

Challenge 2 The Training Data

- Provide means to process (the essence of) large amounts of data on quantum computers

Challenge 3 The Interfaces

- Provide standardized interfaces that allow for dynamic combination of QAI components and (by extension) for experts of different fields to collaborate on QAI algorithms

Challenge 4 The Real Reason

- Keep track of the source of observed improvements

Reference: The Holy Grail of Quantum Artificial Intelligence: Major Challenges in Accelerating the Machine Learning Pipeline
Publication:ICSEW'20: Proceedings of the IEEE/ACM 42nd International Conference on Software Engineering Workshops June 2020 Pages 456–461

Quantum Search Algorithm – Grover's Algorithm

The database of N items can have a single or multiple target elements in it

The elements in a database can have some order (sorted database) or no order (unsorted database) at all

- The unsorted database with single target element can be searched with Grover algorithm in O(√N) steps compared to O(N) steps by a classical computer. This is an example of quadratic speed up in computation time.
- Similarly in the unsorted database with M target elements one of the target elements can be obtained in O√(N/M) steps by Grover algorithm

Reference: Giri, P.R., Korepin, V.E. A review on quantum search algorithms. Quantum Inf Process 16, 315 (2017). https://doi.org/10.1007/s11128-017-1768-7

Quantum Computing Powered Software Testing Benefits

Time Complexity

- Quantum computation has the advantage of speed over its classical counterpart
- Peter Shor showed that it is possible for a quantum algorithm to compute factorization in polynomial time
- Grover showed that it is possible to search for a single target item in an ***unsorted*** database in a time which is quadratically faster than what a classical computer needs to complete the same task

Cost Complexity

- On average, half the cost of developing complex software is on Verification & Validation (VV) and the cost is increasing
- Next gen software VV will require new computing paradigm

Bug Types in Quantum Software Testing

Bug Type	Defense Strategy
(1) Incorrect quantum initial values	Assertion checks for classical and superposition preconditions
(2) Incorrect operations and transformations	Assertion checks for unit testing
(3) Incorrect composition of operations using	iteration Assertion checks for classical intermediate states
(4) Incorrect composition of operations using recursion	Assertion checks for entangled intermediate states
(5) Incorrect composition of operations using mirroring	Assertion checks for product state postconditions
(6) Incorrect classical input parameters	Assertion checks for classical postconditions
(7) Incorrect deallocation of qubits	Assertions on algorithm postconditions

Reference: arXiv:2007.07047

Assertions for Quantum Software

An assertion is a statement about the expected behavior of a software component that must be verified during execution

In software development, a programmer defines an assertion to ensure a specific program state at run time

Assertions have been used extensively for detecting runtime faults, documenting programmer intent, and formally reasoning about the correctness of classical programs

Invariant & Inductive Assertions

- Additive invariants could be derived from additively inductive assertions, and can be generated through an SDP (Semidefinite Programming solver

Applied Quantum Hoare Logic

- Binding QHL to a particular class of pre- and post-conditions (assertions), i.e., projections, to reduce the complexity of quantum program verification and to provide a convenient way used for debugging and testing of quantum software

Reference: arXiv:2007.07047

Assertion Library for Quantum Software

Assert Probability	• Uses a robust statistical method to test the probability of observing a qubit in each state
Assert Entangled	• Takes two qubits as its arguments to test whether they are entangled
Assert Equal	• Tests the equality of two states in quantum programs
Assert Teleported	• Test quantum teleportation as it is a significant protocol in the quantum realm.
Assert Transformed	• Tests the validity of any unitary transformation

Reference: arXiv:2007.07047

Quantum Software Testing

QuanFuzz (Fuzz Testing)

- Define some quantum sensitive information to evaluate the test inputs for quantum software and use a matrix generator to generate test cases with higher coverage

QSharpCheck

- A property-based testing approach for quantum software written in Q#

Functional Testing

- (1) The initial situation of a quantum test case sets up the initial status of the qubits
- (2) Similar to classical testing, the quantum circuit is executed
- (3) The test suite saves the obtained result to calculate the most probable result

White-Boxing Testing

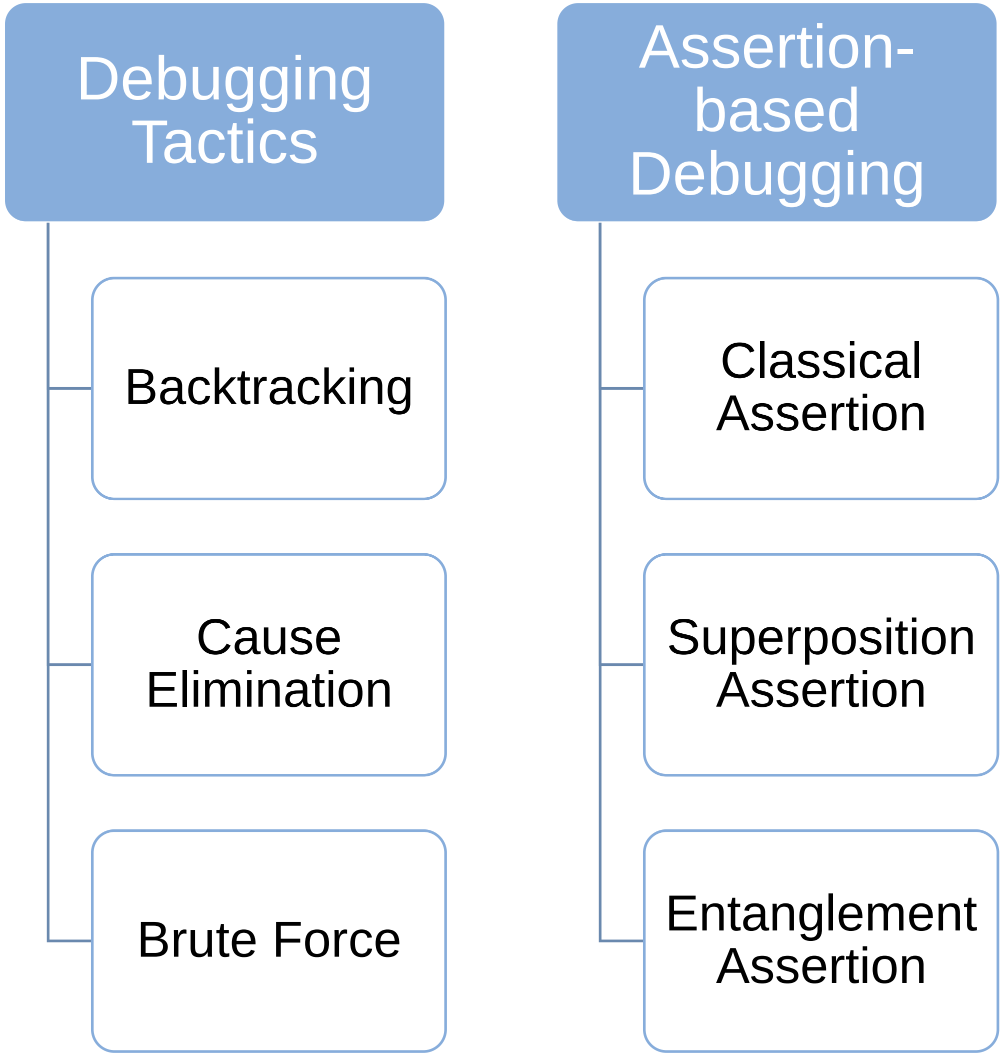
- Tests a software solution’s internal structure, design, and coding.

Model-based Testing

- First use some modeling language, for example, UML, to model the behavior of the quantum software

Reference: arXiv:2007.07047

Quantum Program Debugging



Reference: arXiv:2007.07047

Dr. Tuhin Chattopadhyay

THE THINKER

Thanks