INTRO TO QUANTUM COMPUTING

Week 24 Lab

NEAR-TERM QUANTUM ALGORITHMS

<insert TA name>

<insert date>

PROGRAM FOR TODAY

- Announcements
- Canvas attendance quiz
- Pre-lab zoom feedback
- Lab content
- Post-lab zoom feedback

Canvas attendance quiz passcode:

ANNOUNCEMENTS

- The last course activity is a panel discussion on Sunday May 9 at 2 p.m. EDT

- We **highly encourage** you to attend the live panel! It will feature several experts in the field of quantum computing, including Scott Aaronson, Clarice Aiello, Will Oliver, and Spiros Michalakis

- This is your chance to ask all your questions!

Canvas attendance quiz passcode:

ANNOUNCEMENTS

- **Schrodinger' Chat** will stream on twitch over the summer.
- First stream on **Saturday, May 22nd at 10 a.m. EDT**
- Follow us on **twitch.tv/qubitbyqubit**

Canvas attendance quiz passcode:

CANVAS ATTENDANCE QUIZ

- Please log into Canvas and answer your lab section's quiz (using the passcode posted below).
  - This is lab number:
  - Passcode:

- **Give your TA some feedback!** Indicate your level of agreement with these statements:
  - My TA helped make the course a great experience
  - My TA helped me feel like I belong in the field of quantum computing
  - I see my TA as a STEM role model

- **This quiz is not graded but counts for your lab attendance!**

Canvas attendance quiz passcode:

PRE-LAB ZOOM FEEDBACK

On a scale of 1 to 5, how would you rate your understanding of this week's content?

- 1 –Did not understand anything
- 2 – Understood some parts
- 3 – Understood most of the content
- 4 – Understood all of the content
- 5 – The content was easy for me/I already knew all of the content

In lecture this week, we discussed near-term quantum algorithms with a focus on QAOA

Canvas attendance quiz passcode:

LEARNING OBJECTIVES FOR LAB 24

- Understanding near-term quantum algorithms
  - Optimization problems
  - QAOA
  - Case study – Portfolio optimization
- Learning how to continue growing beyond this course
  - Making sense of code
  - Self-teaching and bootstrapping

Canvas attendance quiz passcode:

CODING IN THE REAL WORLD

- When trying to implement your own code, you'll often find tutorials, documentation, someone else's implementation, etc.
- These implementations are usually messy/don't do exactly what you want
- How should you start?
- Today, we'll try to make sense of a pre-written implementation of QAOA

Canvas attendance quiz passcode:

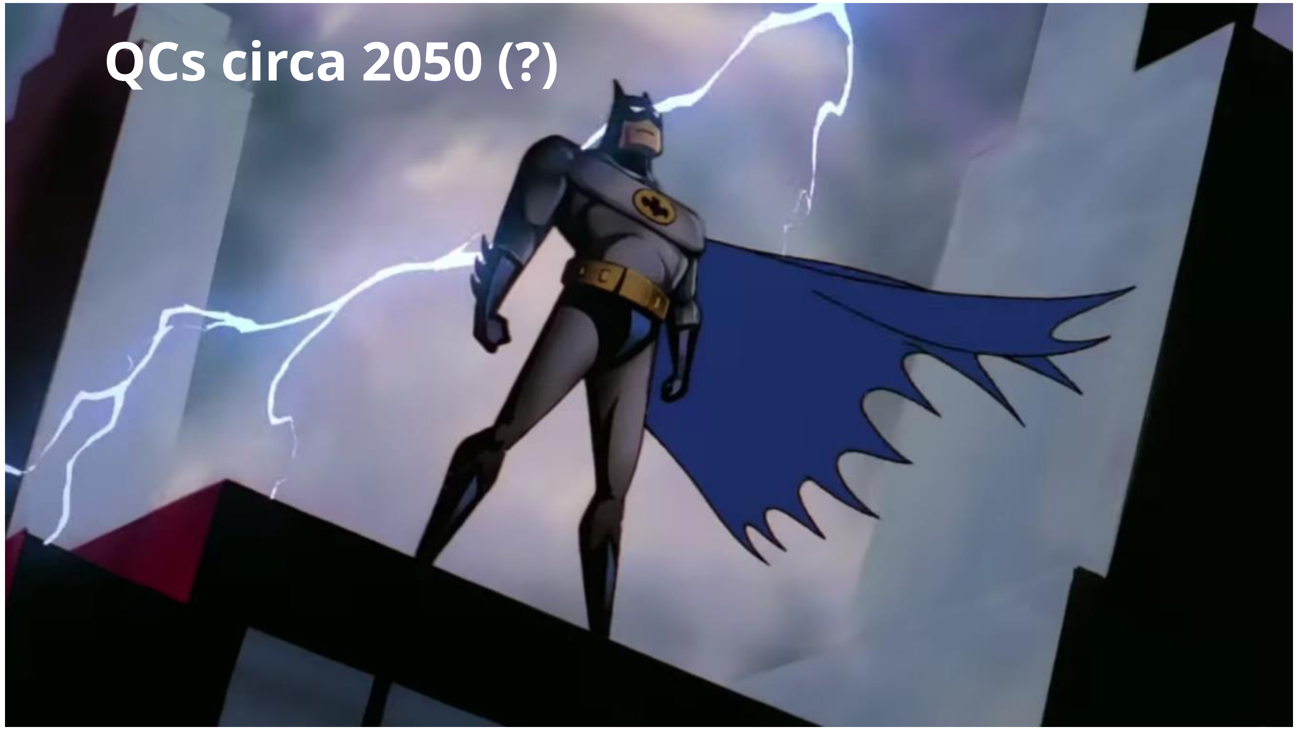
STATUS OF QUANTUM COMPUTERS

Current	Planned	Needed (for Shor's/Grover's)
65 noisy qubits	1,000 noisy qubits (by 2023)	~10,000 error-free qubits

Canvas attendance quiz passcode:

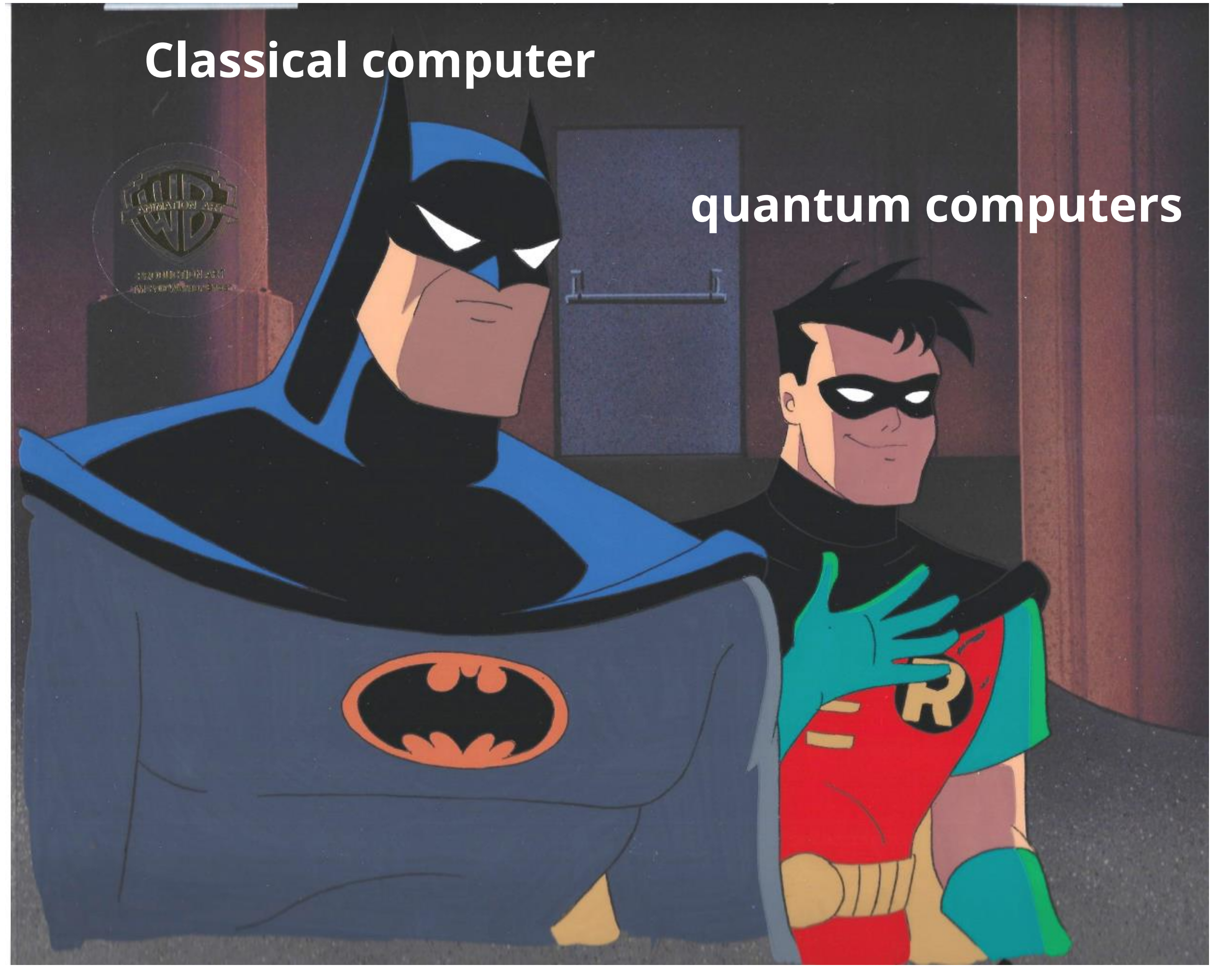
NEAR-TERM QUANTUM ALGORITHMS

- We would like quantum computers to do Shor's, Grover's etc., but the technology is not there yet
- What can we do with the QCs we have right now/ will have within the next 5 years?



NEAR-TERM QUANTUM ALGORITHMS

- Can we use them in combination with classical computers?
- Quantum computers do the part of the job that is difficult for classical computers, and classical computers do the rest
- Hybrid, near-term algorithms



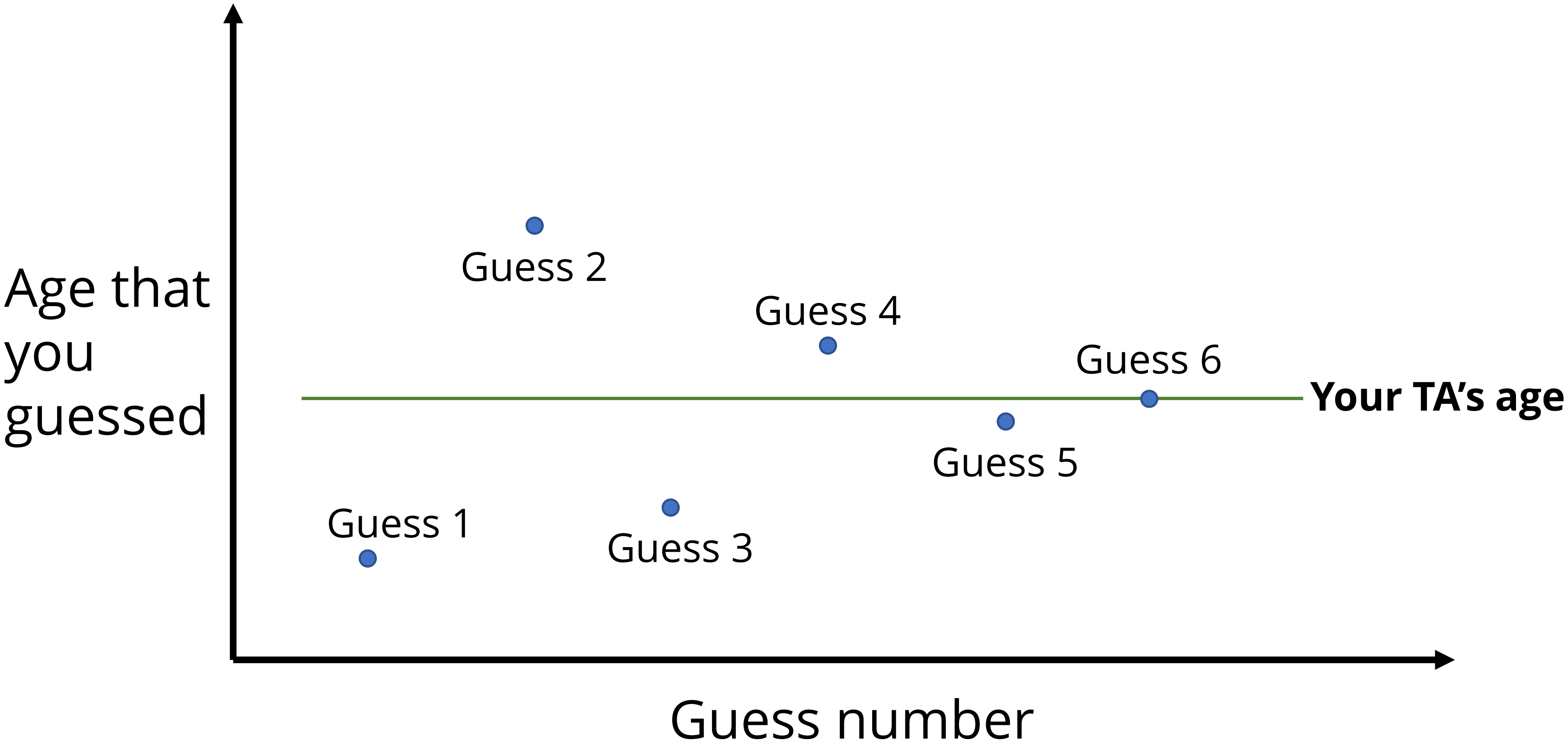
Canvas attendance quiz passcode:

HYBRID ALGORITHMS GAME

- Guess your TA's age! Guess a number, and your TA will tell you if you should go higher or lower
- **You're the quantum computer** – you make and update guesses to solve the problem
- **Your TA is the classical computer** – they listen to your guess and tell you in which direction to move for your next guess until you hit the right solution
- **Optimization problems** are an exciting application of near-term quantum algorithms

Canvas attendance quiz passcode:

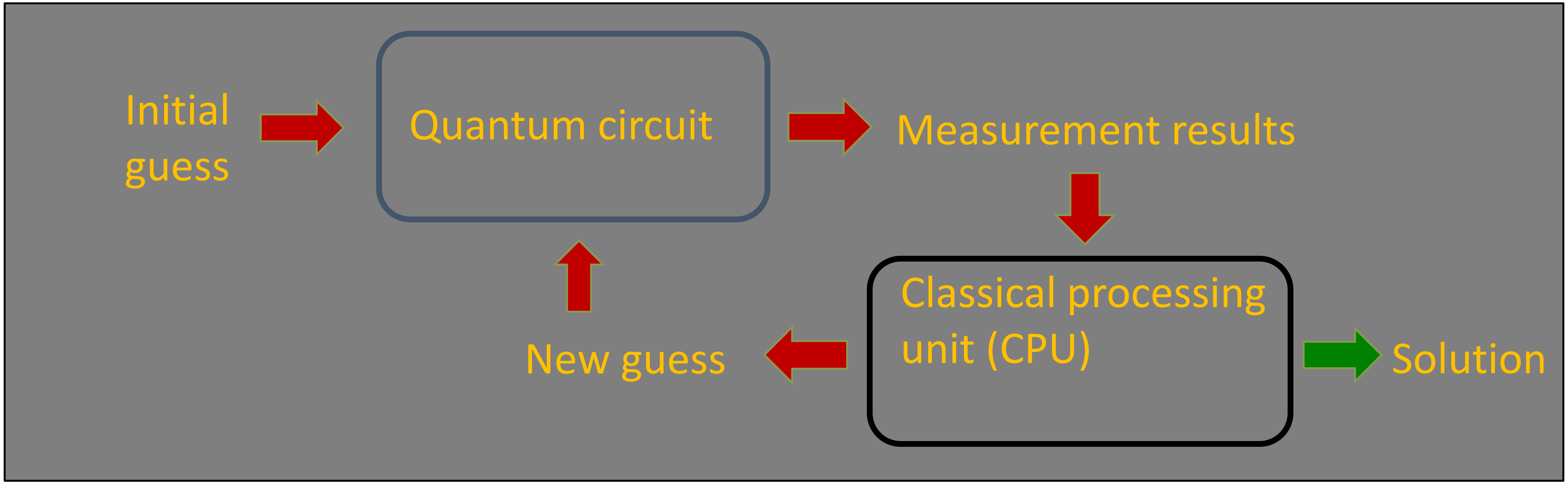
CONVERGING TO AN OPTIMUM



Canvas attendance quiz passcode:

HYBRID ALGORITHMS

Quantum Approximate Optimization Algorithm (QAOA)



Canvas attendance quiz passcode:

FINANCE – PORTFOLIO OPTIMIZATION

- You are a stock trader, and you would like to see what the best selection of stocks (portfolio) would have been for a given day

- **Question:** How many possible combinations of 20 stocks can we make from the 3,300 listed stocks (the number of stocks listed on NASDAQ)?

- This is something classical computers do today. However, as number of choices gets bigger, the optimal solution gets harder to find – need more computing power

PORTFOLIO OPTIMIZATION

- Let's make a simple version of this problem

- You have 5 stocks to choose from
  - Total number of assets (stocks) = 5
- You can only hold a certain number of stocks in your portfolio
  - Budget (max. number of stocks you can hold) = 2

- Your goal is to **maximize your returns**, given your budget and the amount of risk you're willing to take

- What information do you need to pick these stocks?

PORTFOLIO OPTIMIZATION

- You would probably want to look at the historical performance of the stock

- What is the average return on the stock?
  - Greater average return means that you can expect to make more money, on average
- What is the variance in that return?
  - The more the variance, the higher the risk (and reward)! How much risk are you willing to take?

PROCESS OF OPTIMIZATION

1. Create a **cost function,** which is one function that incorporates all our important variables, including:
   1) Average return
   2) Variance in return (amount of risk you're willing to take)
   3) Budget constraints and penalties for going over budget
2. Convert this cost function into a quantum operator (so that the classical and quantum computers can communicate)
3. Implement the operator in a circuit, and measure the outcome (**quantum computing step**)
4. Based on the measured outcome, adjust the quantum operator (**classical computing step**)
5. Repeat steps 3 and 4 until the change in measured outcome is below a threshold → the algorithm has **minimized** the cost function
6. Display the final solution

Canvas attendance quiz passcode:

LET'S CODE!

- Today, we'll look at an implementation of QAOA for portfolio optimization
- We'll try to make sense of what's going on in the implementation. We won't be able to understand everything within the lab time – this is normal!
- Fully understanding every step of the entire algorithm, and every line of code, would take hours/days of work
- This is an active research area, and these implementations are being developed right now!

CODING IN THE WILD

Useful steps in developing code:

- Find open-source code for a similar problem (qiskit textbook/github)
- Run code, check that it works
- See what you can change
- Search for documentation (google/stack exchange)
- Establish dependencies- which part of the code depends on other parts?
- Removing fluff – get to a minimum working example

KEY TAKEAWAYS

- Near-term quantum algorithms (such as VQE and QAOA) aim to use quantum computing resources that are currently available/will be available in the next few years to solve useful problems

- As you make progress in quantum computing, you will come across tutorials/other implementations. Making sense of them is a useful skill, and the process is messy and takes patience

- Google/stackoverflow/qiskit slack are resources to find if issues you've encountered have been solved/are being solved

Canvas attendance quiz passcode:

FURTHER READING AND RESOURCES

- Intro to QAOA and VQE
- Original QAOA paper
- Qiskit textbook page on QAOA
- QC in the news

Canvas attendance quiz passcode:

POST-LAB ZOOM FEEDBACK

After this lab, on a scale of 1 to 5, how would you rate your understanding of this week's content?

- 1 –Did not understand anything
- 2 – Understood some parts
- 3 – Understood most of the content
- 4 – Understood all of the content
- 5 – The content was easy for me/I already knew all of the content

Canvas attendance quiz passcode: