



GPU-Acceleration in Quantum Computing Applications

Esperanza Cuenca-Gomez, Developer Relations Manager for Quantum Computing

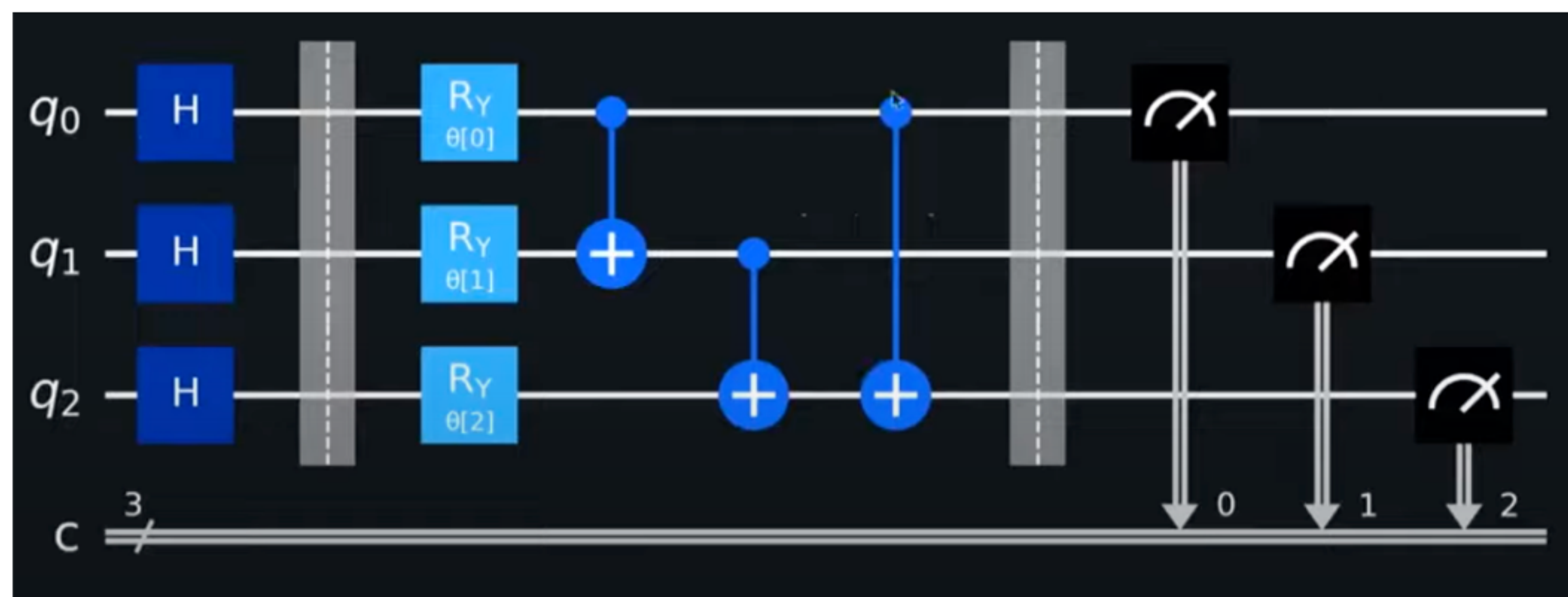
VQE

Overview and Challenges

Overview

Algorithm which estimates the **ground state of a molecule** by **optimizing a parameterized ansatz** through **iterative loops** involving **expectation values** computed on a **QPU** and **classical parameter updates** performed by a **supercomputer**

Variational Quantum Circuit



Initial state preparation

Parameterized circuit $U(\theta)$

Measurement

$$E(\theta) = \langle \psi(\theta) | \hat{H} | \psi(\theta) \rangle$$

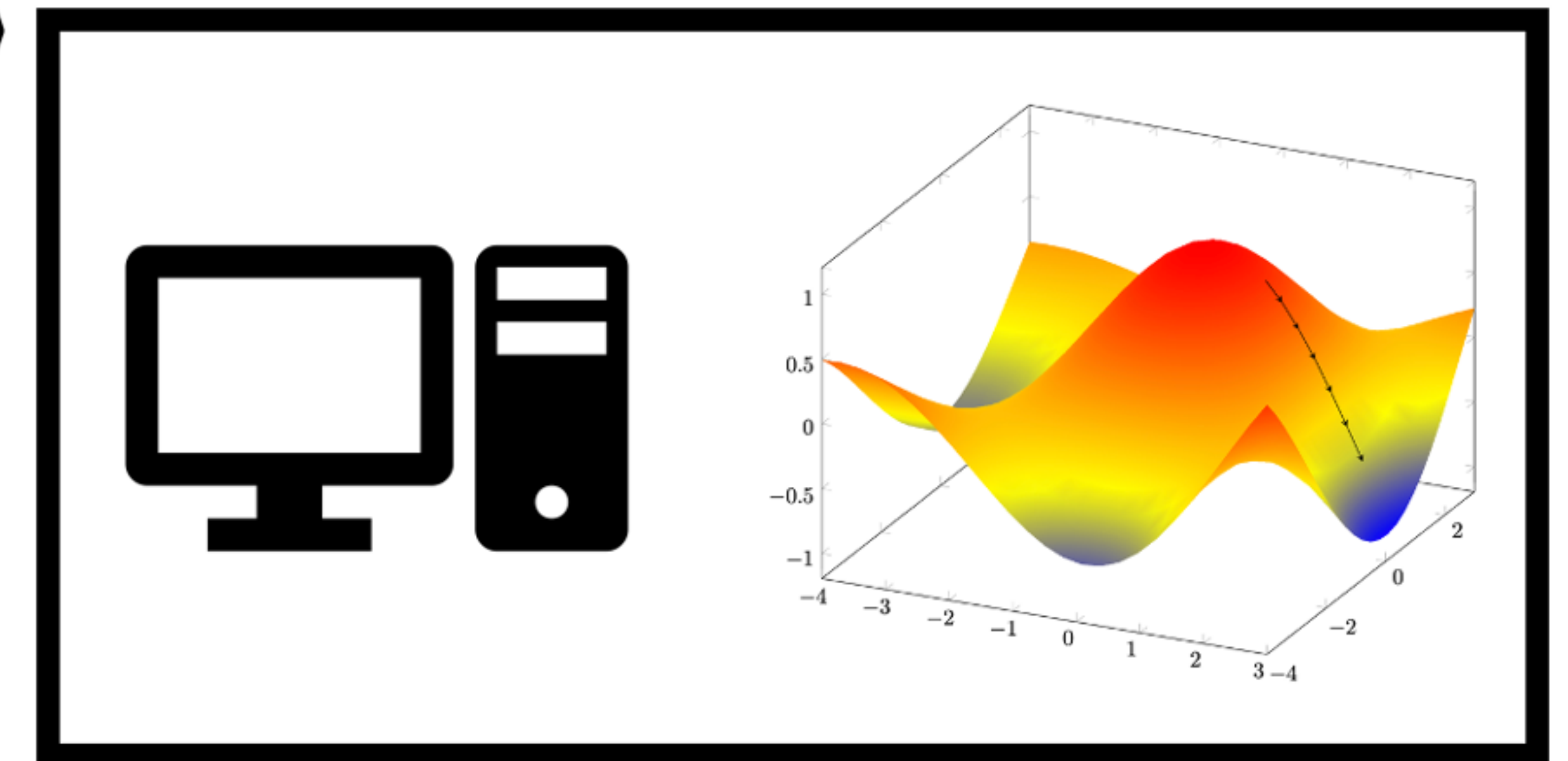


Parameter update



$$\theta_i \rightarrow \theta_{i+1}$$

Classical optimizer



Adapted from [“A variational eigenvalue solver on a quantum processor”](#), Peruzzo et al, 2013

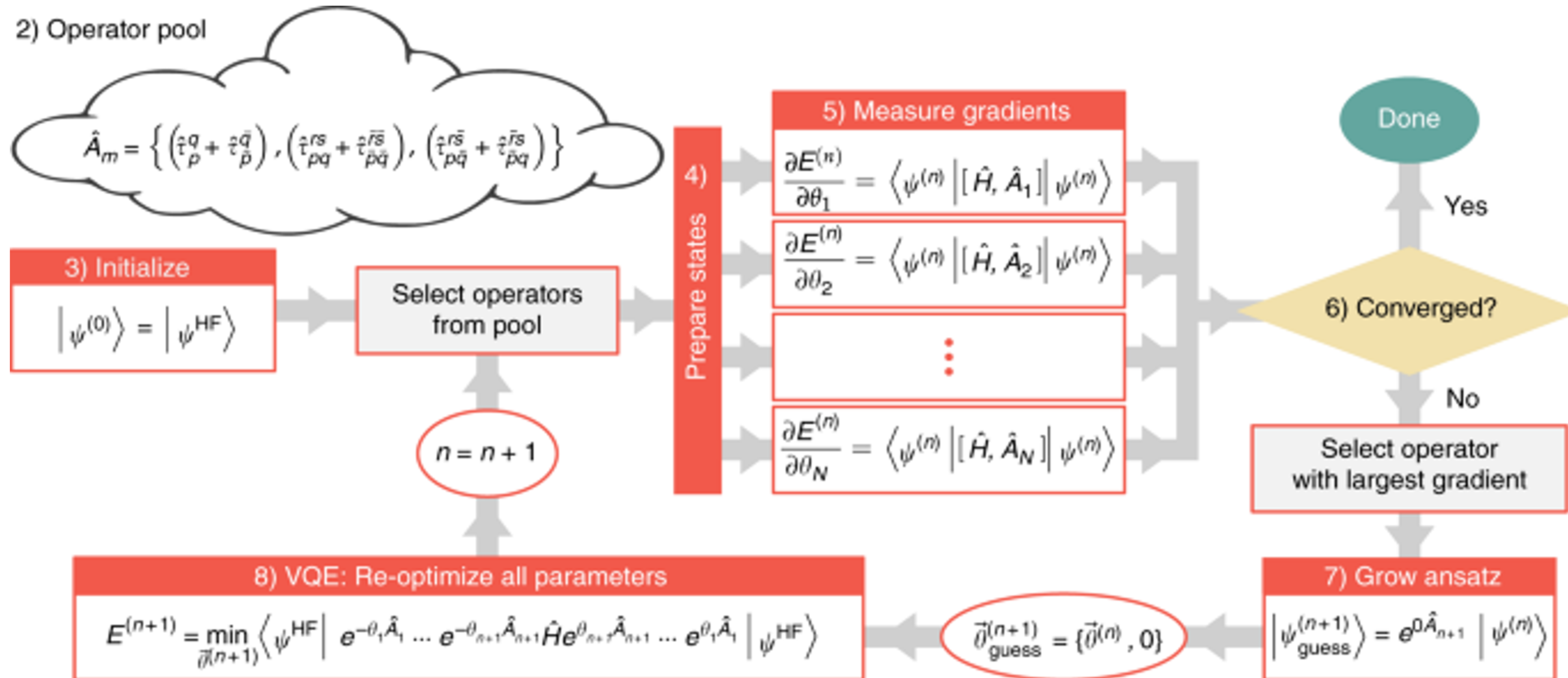
Challenges

- **Costly sampling** of the Hamiltonian terms
- **Barren plateau problem**: even modest problems sizes do not converge due to the flat optimization landscape.

ADAPT-VQE

Adaptively Building the Ansatz

Core idea: “adaptively” build the ansatz over the course of the algorithm



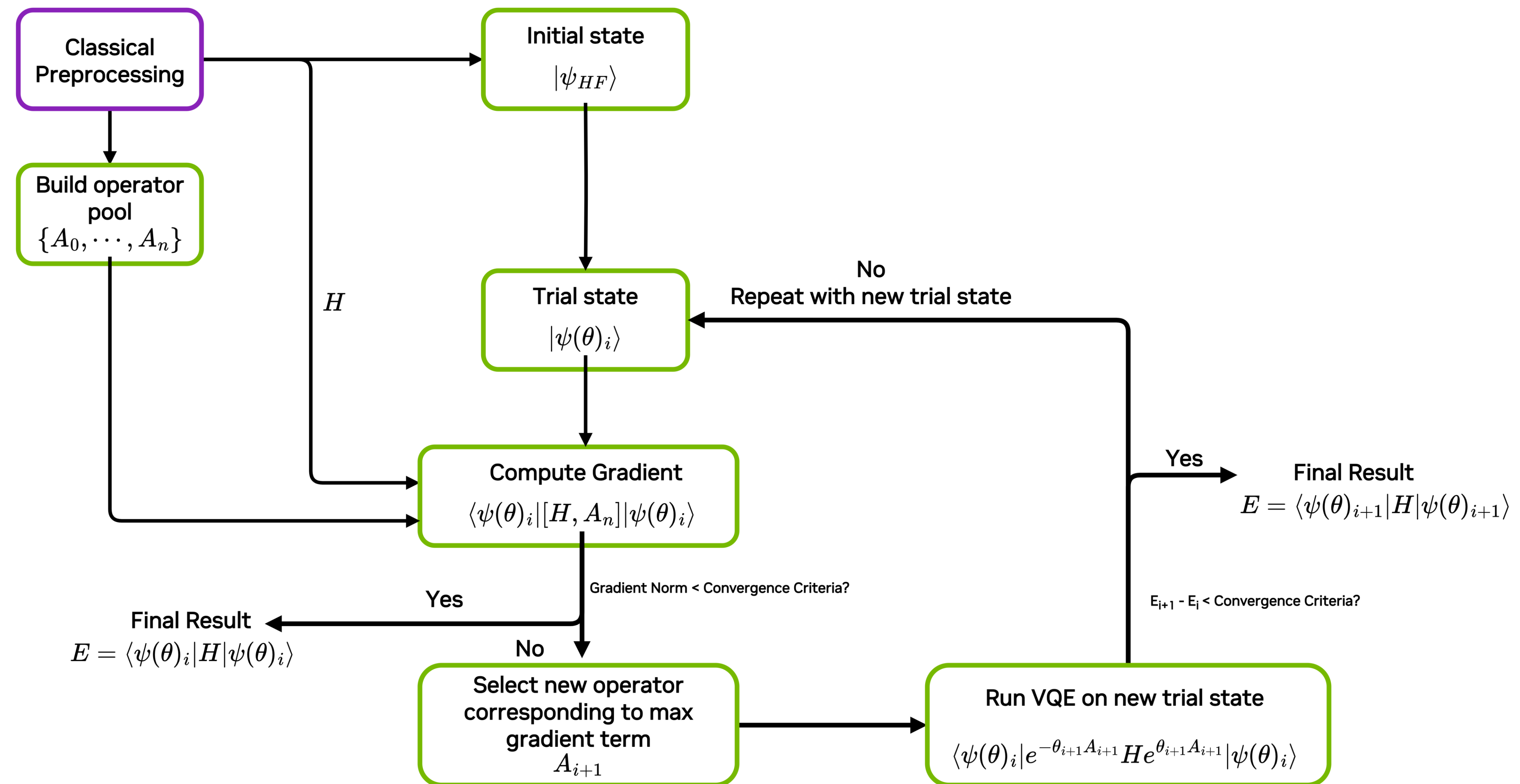
Source: [“An adaptive variational algorithm for exact molecular simulations on a quantum computer”](#), Grimsley et al, 2019.

ADAPT-VQE Step-by-Step

Adaptively Building the Ansatz

Steps

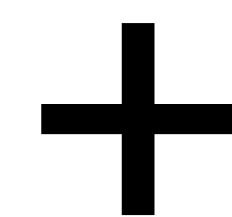
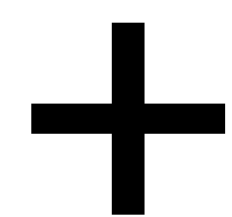
1. On classical hardware, compute the one and two-electron integrals and transform the Hamiltonian to a qubit representation.
2. Build an operator pool, usually corresponding to physical excitations from techniques like unitary coupled cluster with singles and doubles (UCCSD).
3. Initialize qubits in a reference Hartree Fock state.
4. Measure the commutator of the Hamiltonian with each operator in the operator pool to compute the gradient.
5. If the norm of the gradient is below a threshold, then exit. Otherwise, add the operator with the largest gradient contribution along with a new variational parameter.
6. Run VQE on this circuit to obtain optimized variational parameters.
7. Repeat steps 4 through 6 until convergence.



The Generative Quantum Eigensolver

First demonstration of GPT-generated circuits

CUDA-Q



Challenge

- Variational quantum algorithms offered promise for running drug-discovery applications on small quantum devices - but **suffer from serious optimization issues**
- Many of these problems are tied to how circuits are parametrized.

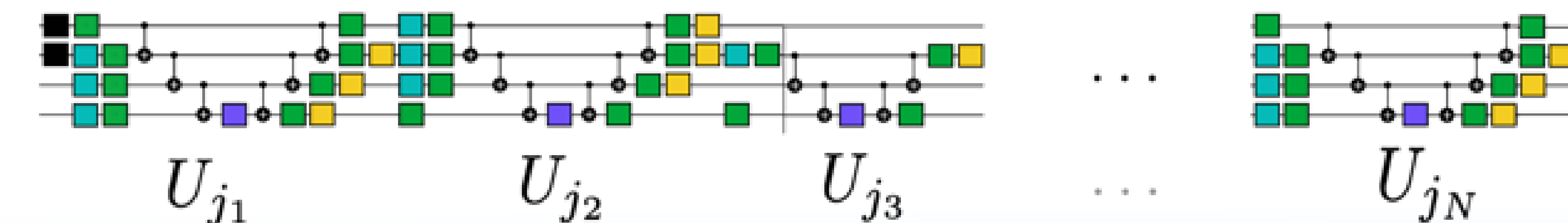
Solution

- The generative quantum eigensolver acts like a Large Language Model – but generating quantum circuits from quantum operations, rather than sentences from words.
- Using a generative model like GPT to create quantum circuits **avoids the limitations of traditional variational quantum algorithms**

$\vec{j} \sim$ "Once upon a time . . . happily ever after"
word_{j1} word_{j2} word_{j3} word_{j4} . . . word_{jN-2} word_{jN-1} word_{jN}

LLM

$\vec{j} \sim$



GPT-QE



New approach in Using AI for building quantum applications



Can be extended to various application areas

40X

Speedup over CPU when running GQE on GPU

The Generative Quantum Eigensolver

First demonstration of GPT-generated circuits in the literature

Goal: Find a circuit producing e.g. ground state
Optimization of some possible space of circuits for desired output

GPT-QE specifically employs a GPT model

- Recent advances in attention-based transformer models can be leveraged

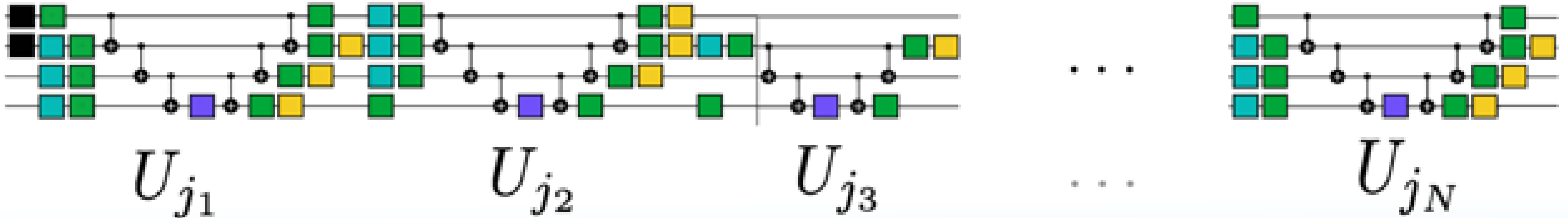
Analogy to Large Language Models (LLMs)

- Quantum operations are analogous to words
- Quantum circuits analogous to sentences
- When trained, GPT-QE generates a sequence of operations to form a circuit

Training learns weights in GPT model

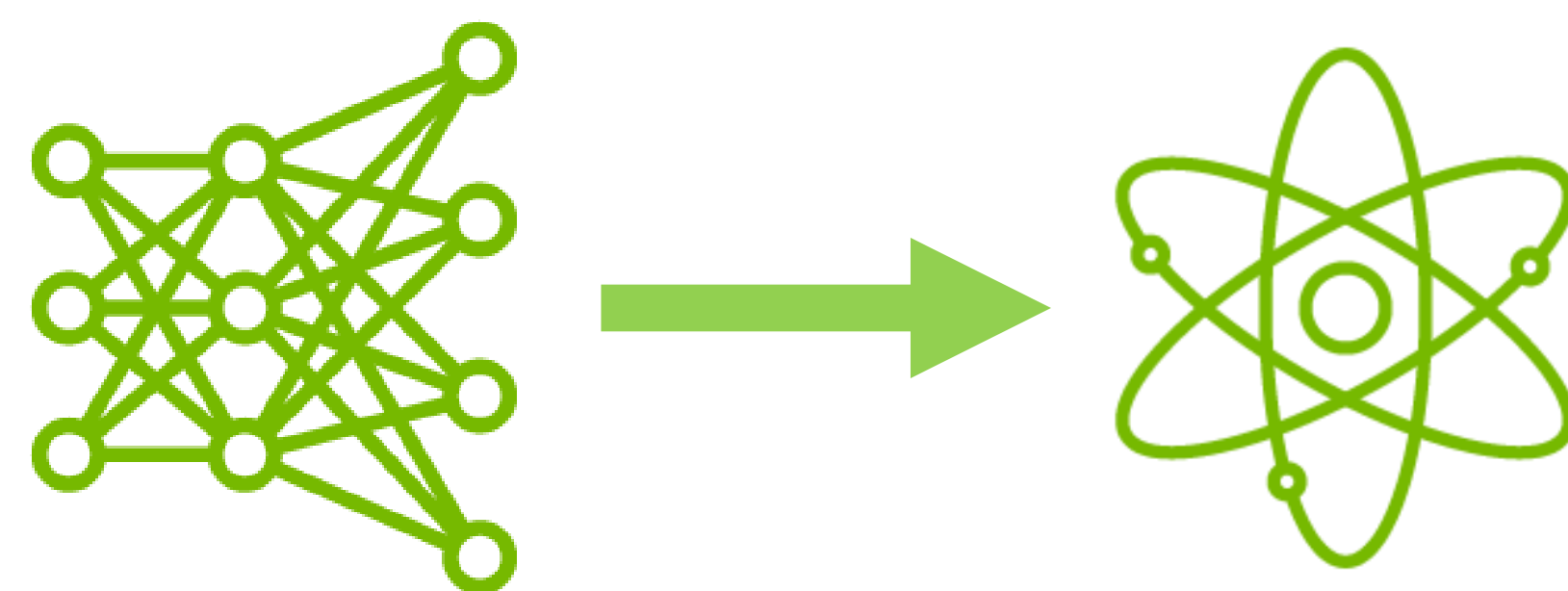
- Cost function compares current prediction to measure of desired circuit output

$\vec{j} \sim$ "Once upon a time . . . happily ever after"
word_{j₁} word_{j₂} word_{j₃} word_{j₄} . . . word_{j_{N-2}} word_{j_{N-1}} word_{j_N} **LLM**

$\vec{j} \sim$

GPT-QE

Extension to other domains

Pretraining databases can be extended to other applications



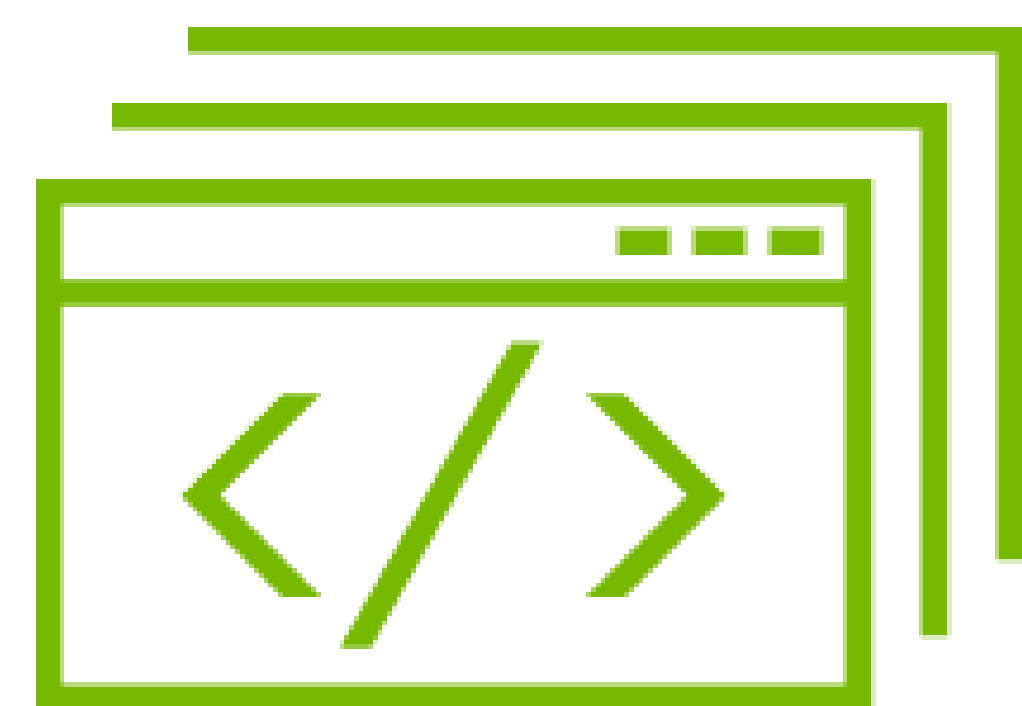
Results of current GPT-QE work is a first step to understanding how GPT can accelerate quantum computing

$\{U_1, U_2, \dots, U_L\}$



$\{\tilde{U}_1, \tilde{U}_2, \dots, \tilde{U}_L\}$

Hamiltonian and operator pool (GPT gate-set) can be switched out to explore applications beyond Chemistry and also accommodate different hardware



Lab 1 - Solving the Ground State Problem with VQE and AI (Generative Quantum Eigensolver) at

https://github.com/NVIDIA/cuda-q-academic/blob/main/chemistry-simulations/vqe_and_gqe.ipynb



Thank you