# NVIDIA Day

CINECA Practical Quantum Computing School 2nd Edition

Dec 2nd 2022

# Agenda

| | |
|---|---|
| Introduction | |
| NVIDIA cuQuantum | Carlo Nardone, NVIDIA |
| NVIDIA cuQuantum: cuTensorNet | Andreas Hehn, NVIDIA |
| NVIDIA QODA | Zohim Chandani, NVIDIA |
| Qibo and cuQuantum integration | Andrea Pasquale, UniMI<br>Stavros Efthymiou, TII |
| Hands-on exercises with Qibo | |

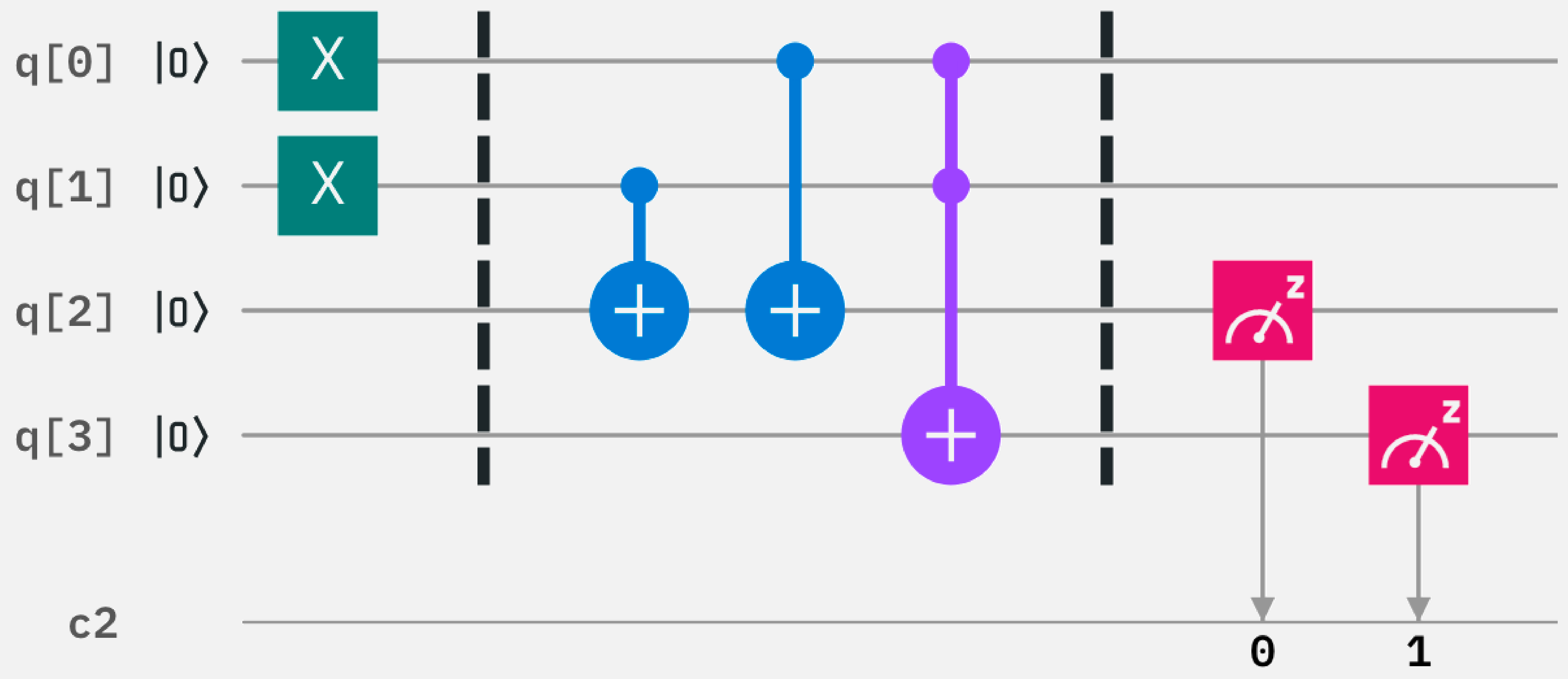
# Quantum Computing Simulation

# GPU-based Supercomputing in the Quantum Computing Ecosystem

Researching the quantum computer of tomorrow with the supercomputers of today

## QUANTUM CIRCUIT SIMULATION

Critical tool for answering today's most pressing questions in Quantum Information Science (QIS):



- What quantum algorithms are most promising for near-term or long-term quantum advantage?
- What are the requirements (number of qubits and error rates) to realize quantum advantage?
- What quantum processor architectures are best suited to realize valuable quantum applications?

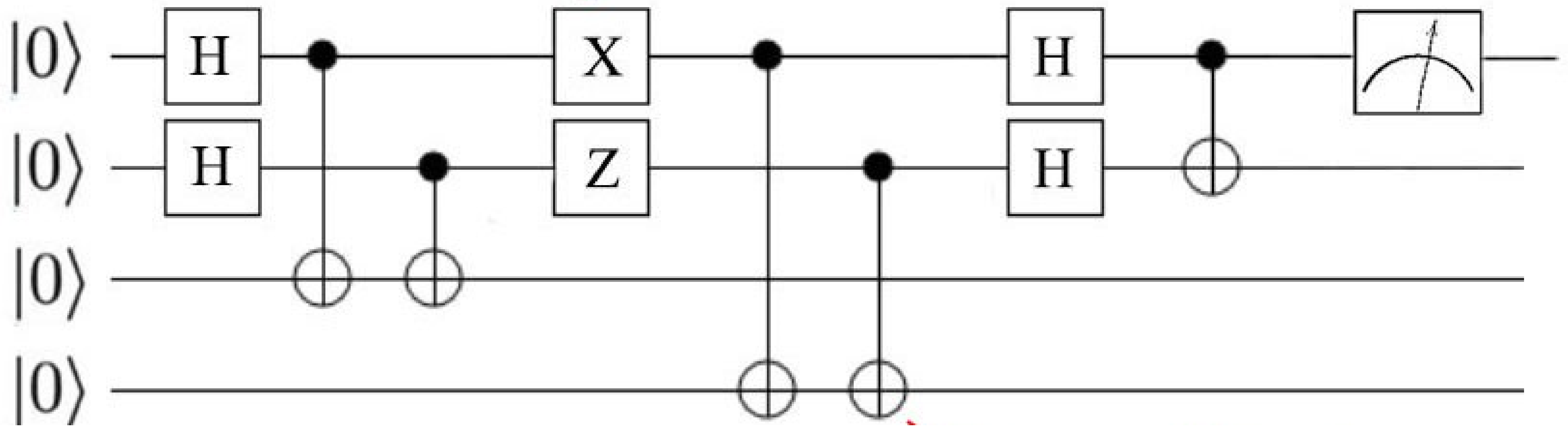
## HYBRID CLASSICAL/QUANTUM APPLICATIONS

Impactful QC applications (e.g. simulating quantum materials and systems) will require classical supercomputers with quantum co-processors

+

- How can we integrate and take advantage of classical HPC to accelerate hybrid classical/quantum workloads?
- How can we allow domain scientists to easily test coprogramming of QPUs with classical HPC systems?
- Can we take advantage of GPU acceleration for circuit synthesis, classical optimization, and error correction decoding?

# Two Leading Quantum Circuit Simulation Approaches



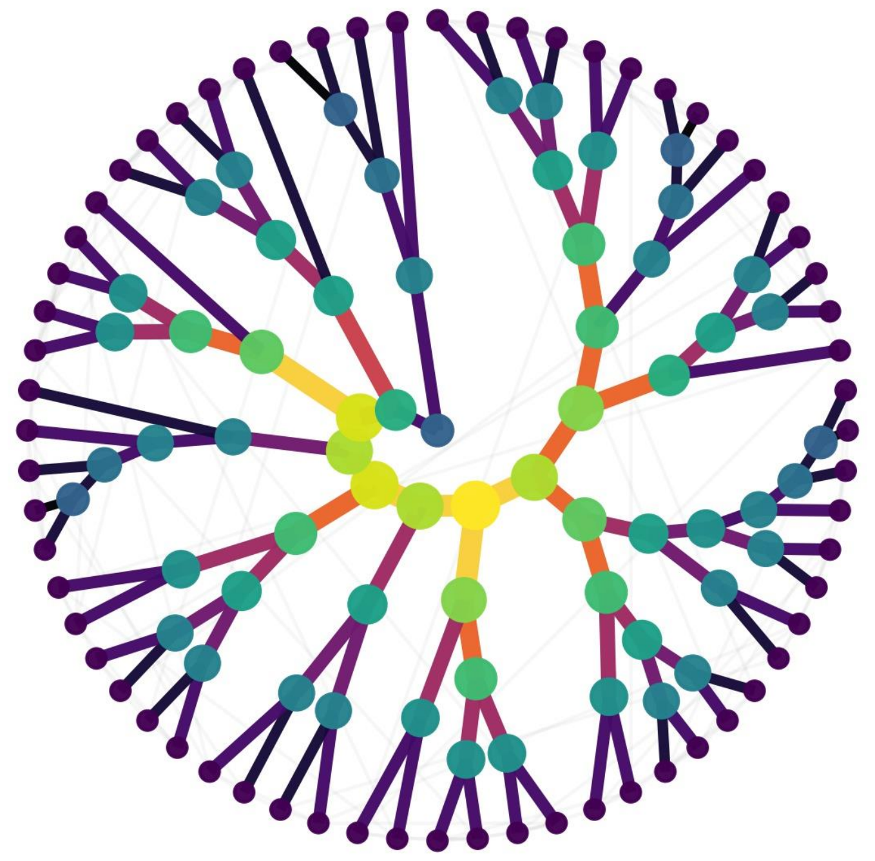
## State vector simulation

**“Gate-based emulation of a quantum computer”**

- Maintain full $2^n$ qubit vector state in memory
- Update all states every timestep, probabilistically sample n of the states for measurement

Memory capacity & time grow exponentially w/ # of qubits - practical limit around 50 qubits on a supercomputer

Can model either ideal or noisy qubits



## Tensor networks

**“Only simulate the states you need”**

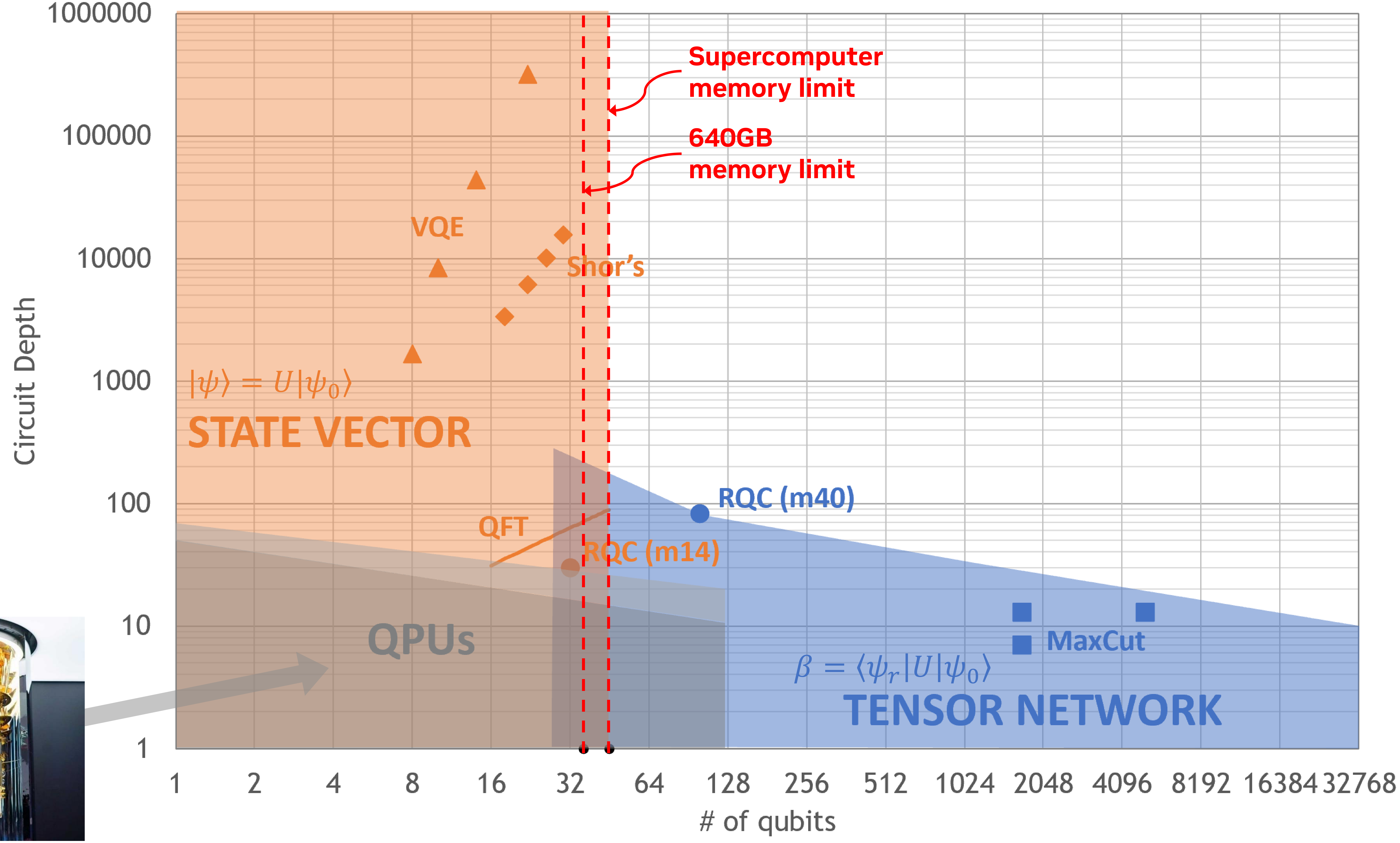
- Uses tensor network contractions to dramatically reduce memory for simulating circuits
- Can simulate 100s or 1000s of qubits for many practical quantum circuits

*GPUs are a great fit for either approach*

*Source of Tensor Network image: Quimb - quimb.readthedocs.io*

# State Vector vs Tensor Network for Quantum Circuit Simulation

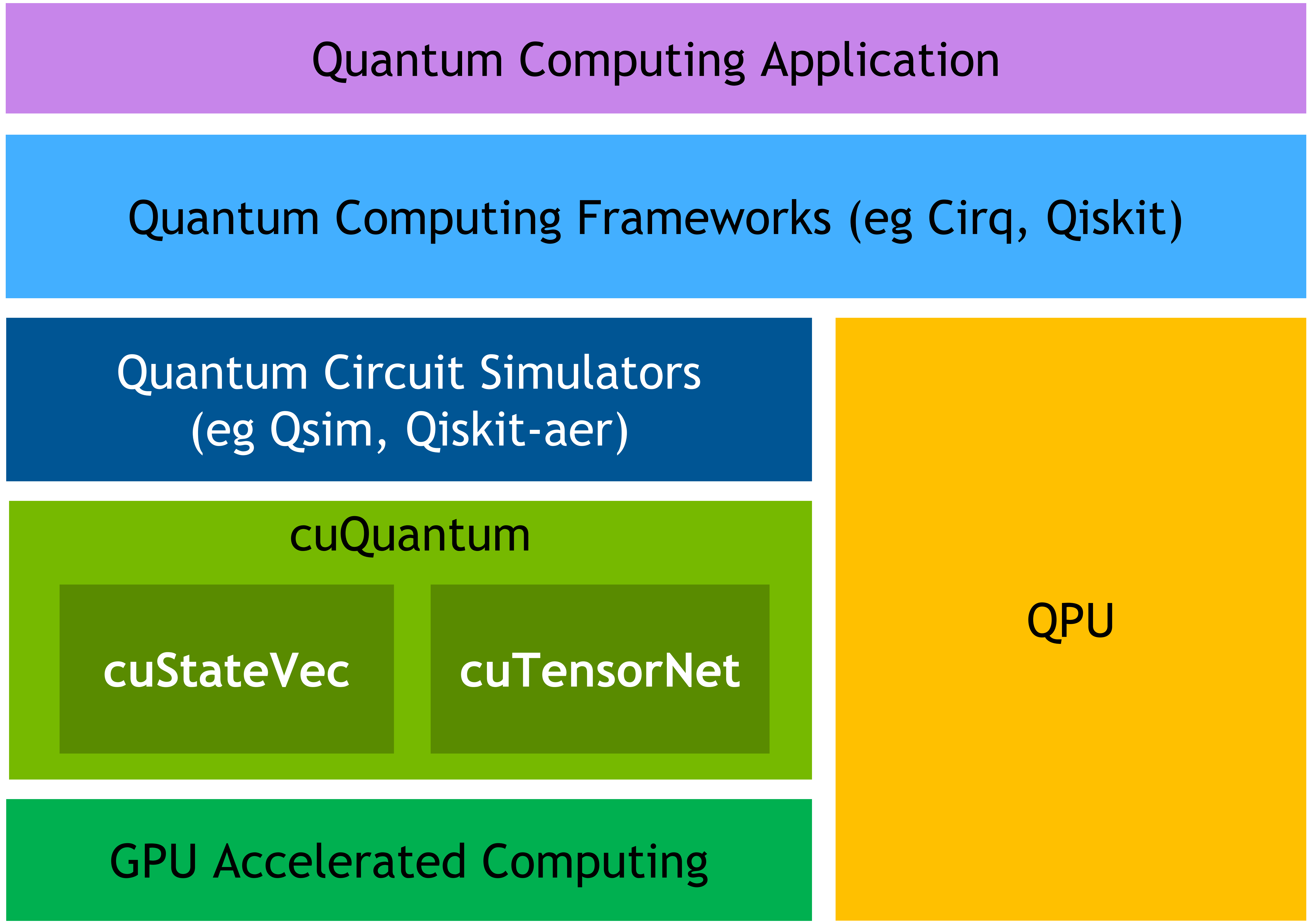
R&D for the computers of tomorrow requires powerful simulations today



# NVIDIA cuQuantum

# Introducing cuQuantum

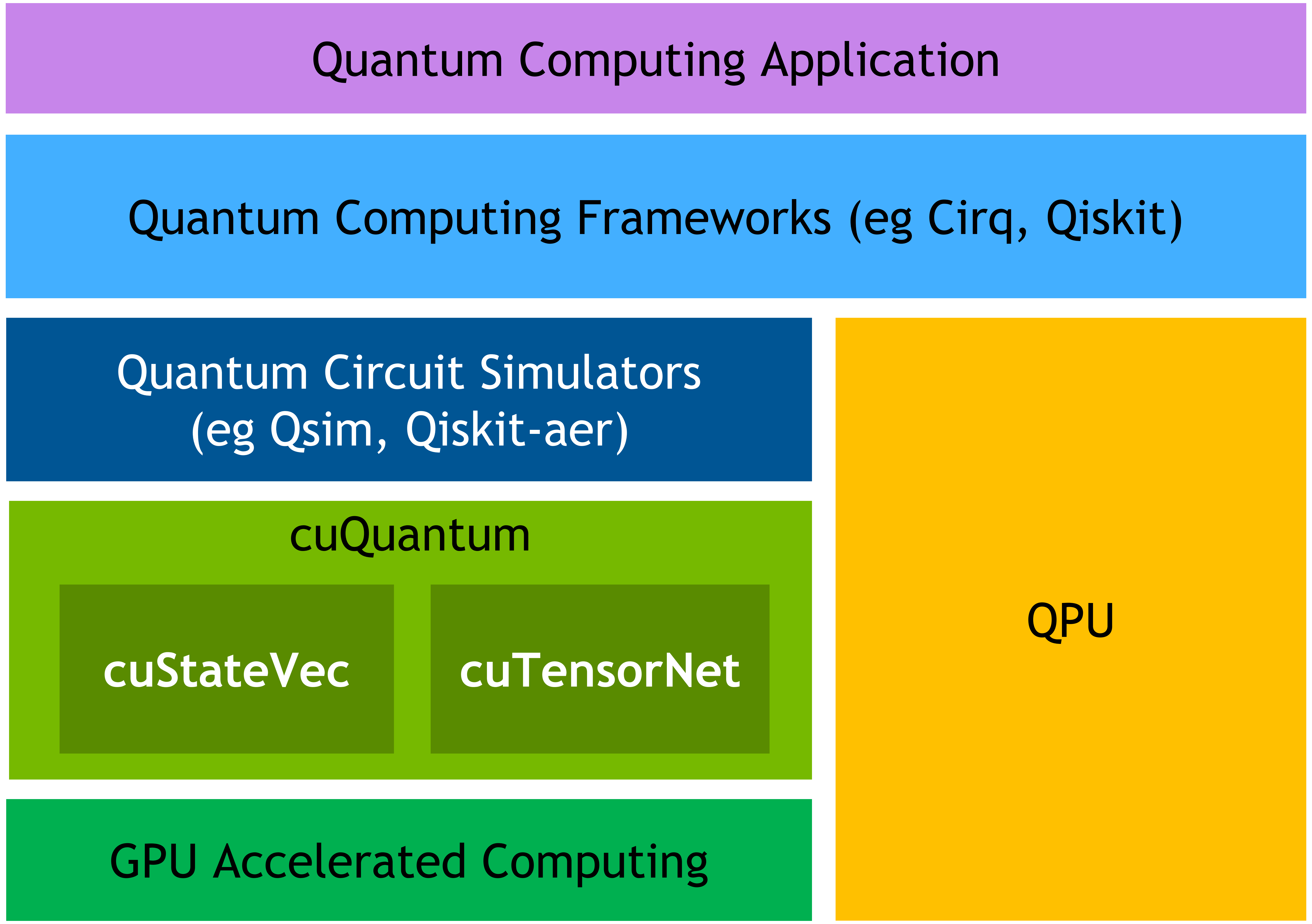
- cuQuantum is an SDK of **optimized libraries and tools** for accelerating Quantum Computing workflows
- cuQuantum is **not** a:
  - Quantum Computer
  - Quantum Computing Framework
  - Quantum Circuit Simulator
- Very similar approach to what NVIDIA has done in the past in the CUDA ecosystem
  - cuBLAS, cuFFT …
  - cuDNN invoked by all major Deep Learning frameworks (PyTorch, TensorFlow, etc.)


Quantum Computing Application
Quantum Computing Frameworks (eg Cirq, Qiskit)
Quantum Circuit Simulators (eg Qsim, Qiskit-aer)
cuQuantum
cuStateVec
cuTensorNet
GPU Accelerated Computing
QPU

# Introducing cuQuantum

- cuQuantum is a platform for Quantum Computing research
  - Accelerate Quantum Circuit Simulators on GPUs
  - Simulate ideal or noisy qubits
  - Enable algorithms research with scale and performance not possible on quantum hardware or on simulators today
- GA availability, integrated with
  - Google Cirq
  - IBM Qiskit
  - Xanadu PennyLane
- DGX Quantum Appliance container available on NGC: catalog.ngc.nvidia.com/orgs/nvidia/containers/cuquantum-appliance
- Full documentation at docs.nvidia.com/cuda/cuquantum

Cirq Qiskit PENNYLANE


Quantum Computing Application
Quantum Computing Frameworks (eg Cirq, Qiskit)
Quantum Circuit Simulators (eg Qsim, Qiskit-aer)
cuQuantum
cuStateVec
cuTensorNet
GPU Accelerated Computing
QPU

# cuQuantum Ecosystem

## Frameworks

aws Amazon Braket

Cirq

Qiskit

PENNYLANE

Orquestra®

QIBO

XACC QUANTUM FRAMEWORK

### HPC Centers

Argonne NATIONAL LABORATORY

JÜLICH Forschungszentrum

OAK RIDGE National Laboratory

NERSC

## Other Power Users

IBM

Google Quantum AI

rigetti

QUANTINUUM

IONQ

XANADU

Ψ PsiQuantum

PASQAL

QUANTUM BRILLIANCE

CLASSIQ

QCWARE

ZAPATA

menten.AI

# cuQuantum Performance

Enabling speedups for a range of use cases and users

PASQAL

Faster Quantum Algorithm for Physics-ML

100X

Faster Time-to-solution

24X

More Circuit Depth

Amazon Braket aws

PENNYLANE

New PennyLane Integration via AWS Braket

900X

Faster Time-to-solution

3.5X

Lower Costs

ZAPATA

Orquestra Platform Integration

100X

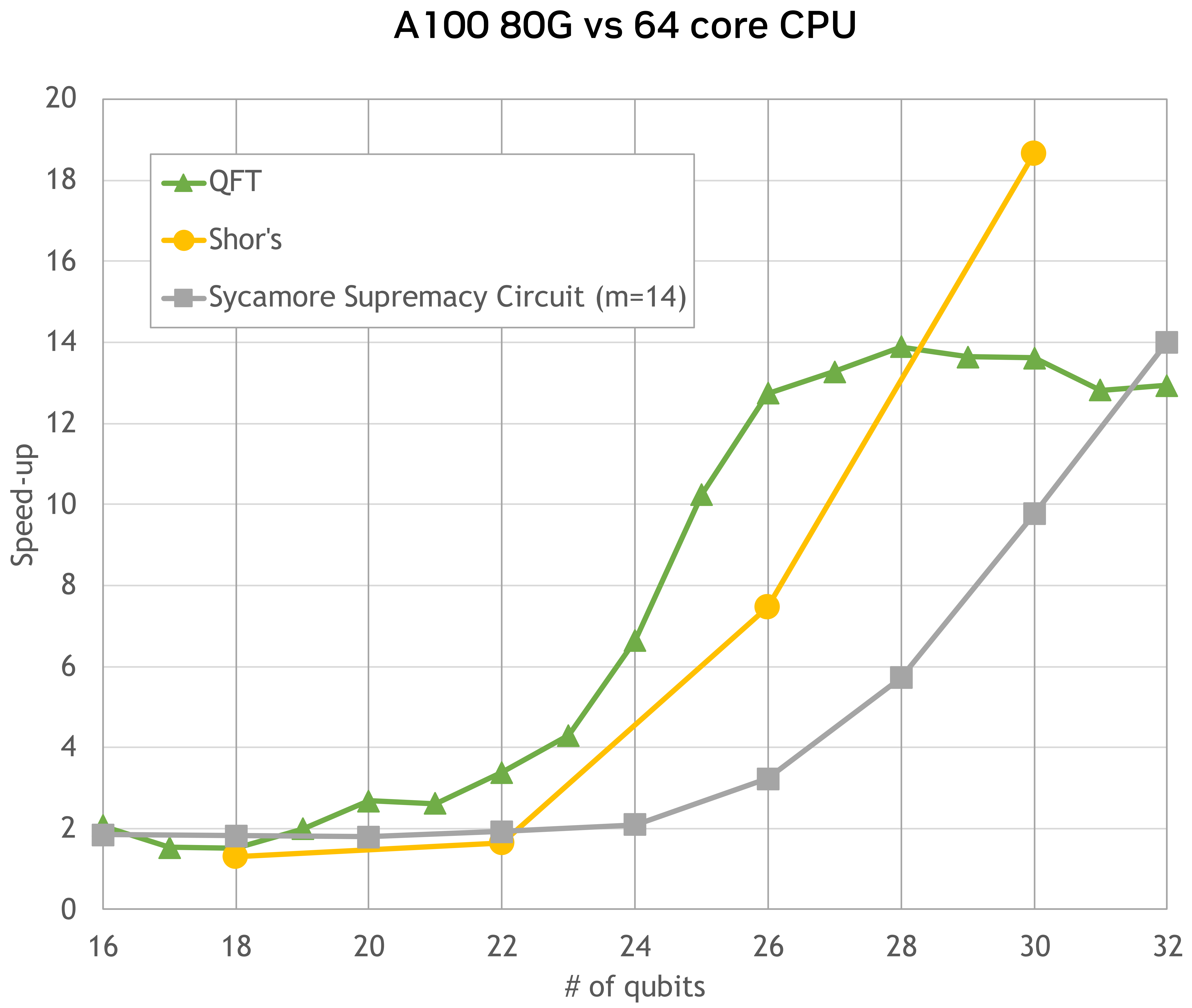
Faster Time-to-solution

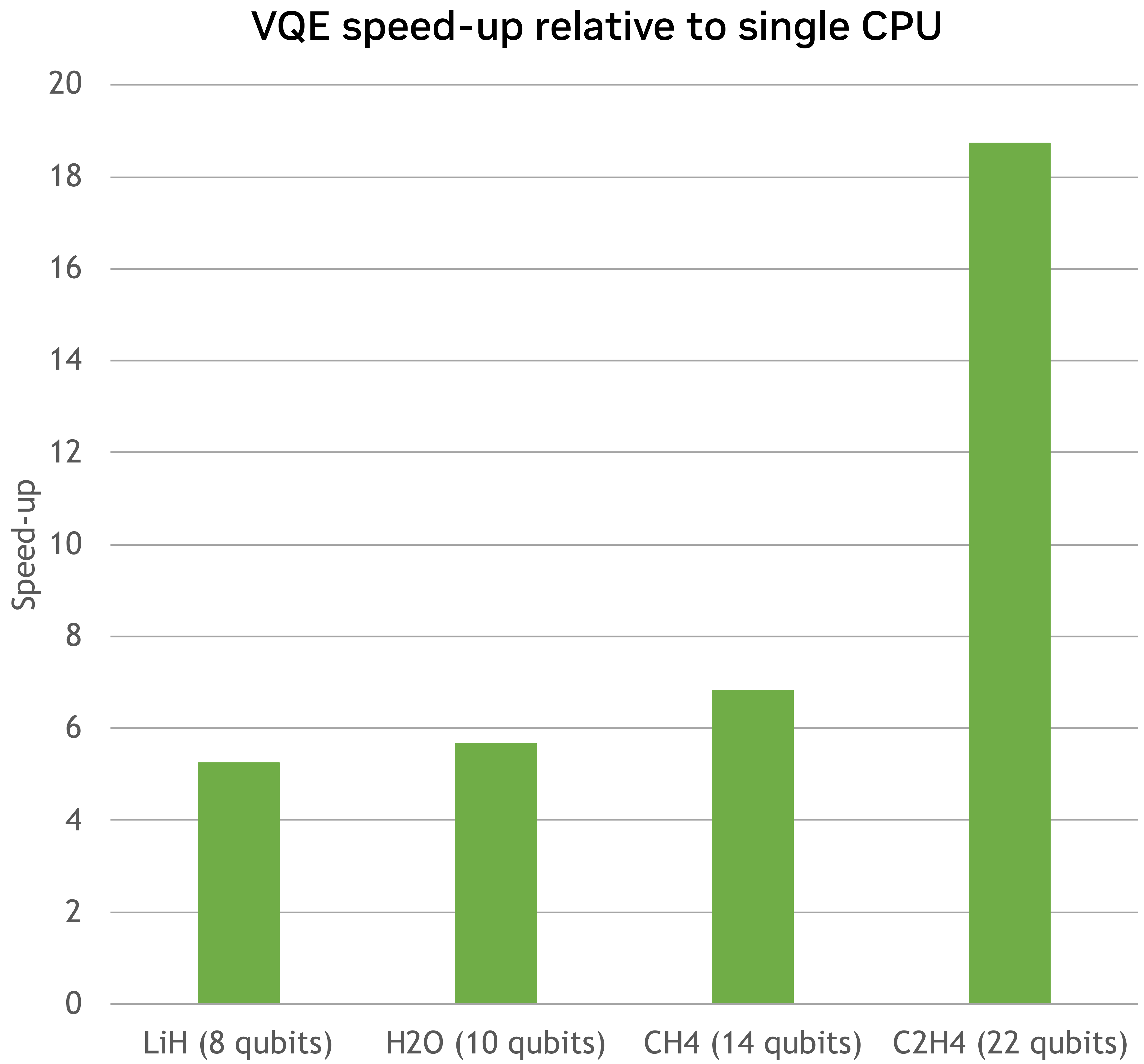
1.5X

More Qubits

# cuStateVec – Single GPU Performance

## Preliminary performance of Cirq/Qsim + cuStateVec on NVIDIA A100





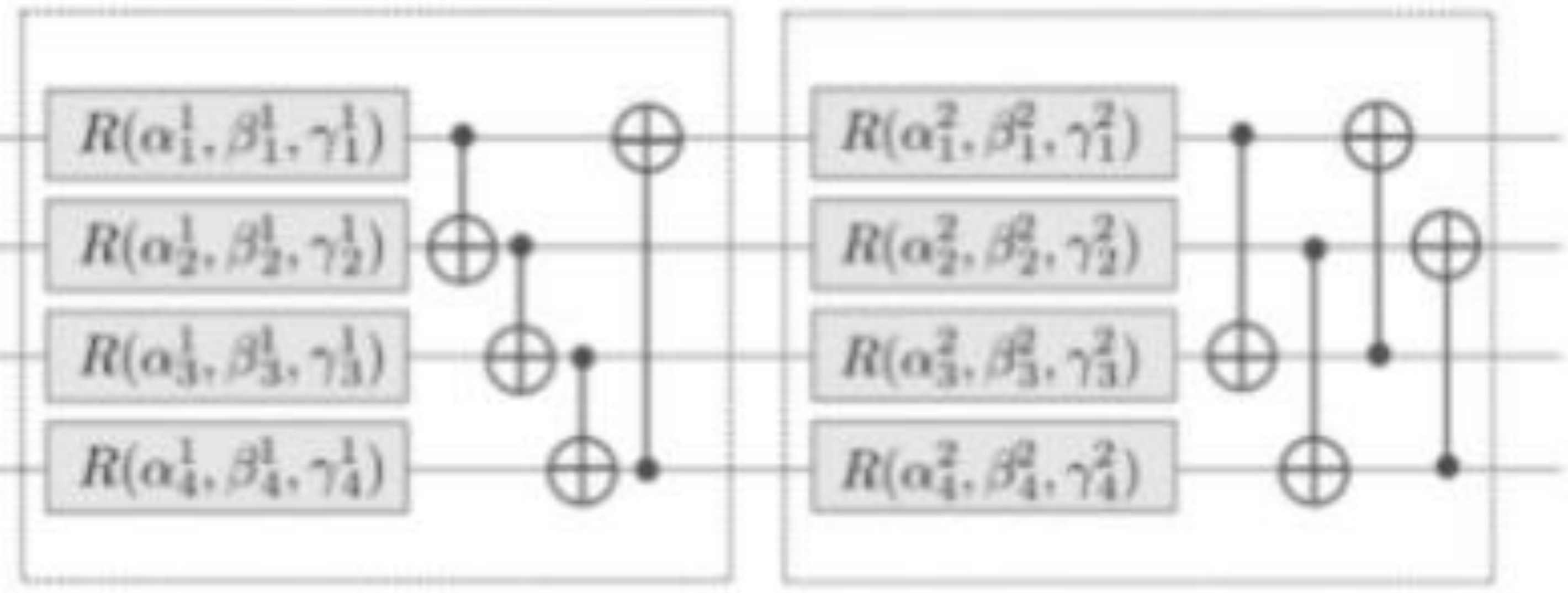
Benchmarks run using cirq/qsim with modifications to integrate cuStateVec
CPUs used were AMD EPYC 7742 with 64 cores
QFT circuit with 32 qubits and depth 63
Shor’s circuit with 30 qubit and depth 15560 (integer factorized: 65)
Sycamore supremacy circuit m=14 with 7480 gates

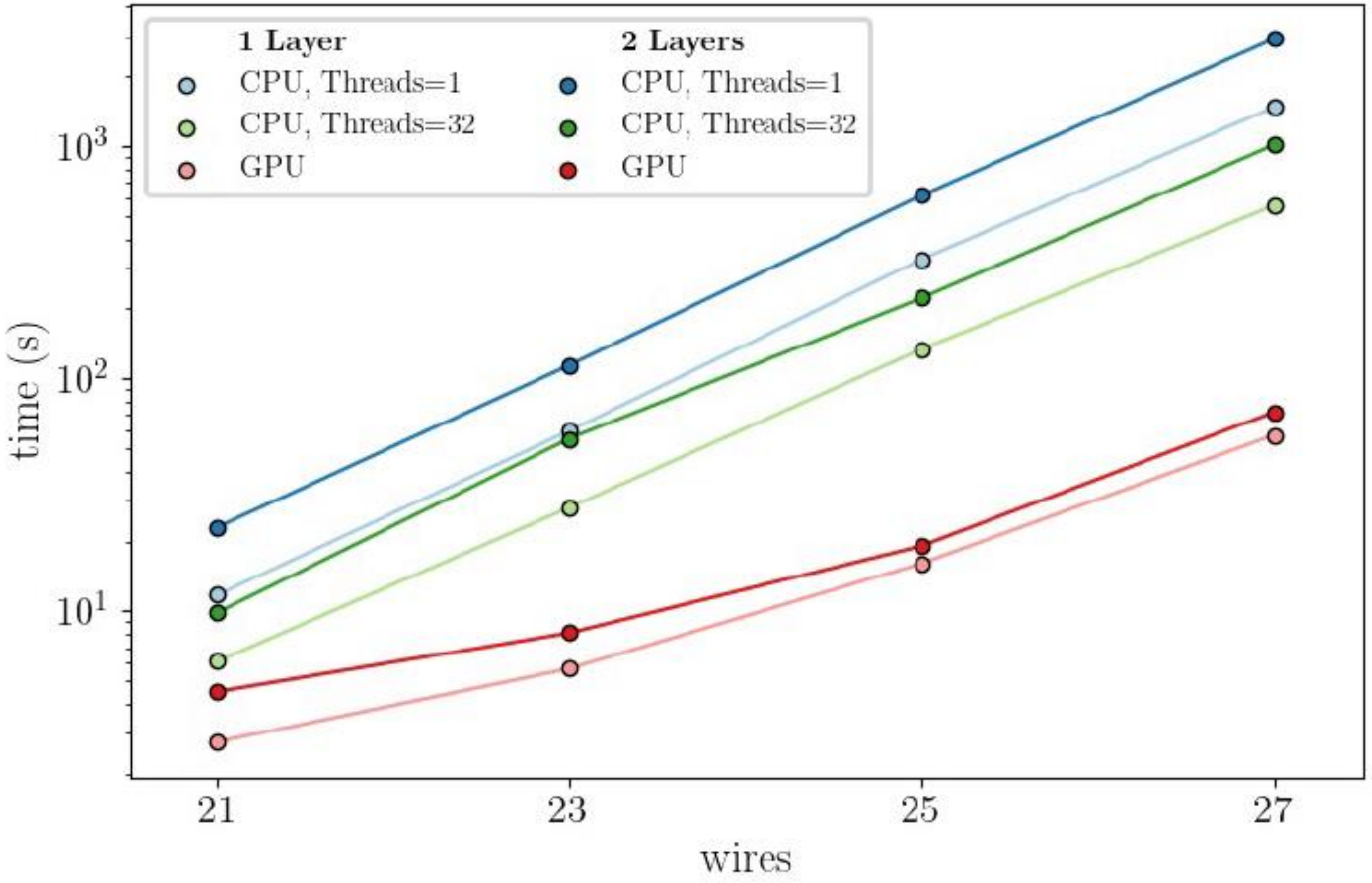
VQE benchmarks have all orbitals and results were measured for the energy function evaluation

# cuQuantum Support for PennyLane

- Leading open-source framework for quantum machine learning and quantum chemistry, built by Xanadu
    - Train Quantum Computers in the same way as Neural Networks
- New simulator *lightning.gpu* with cuQuantum support, available now:
    - [xanadu.ai/products/lightning](xanadu.ai/products/lightning)
- 10x speedup for QML circuits

PENNYLANE

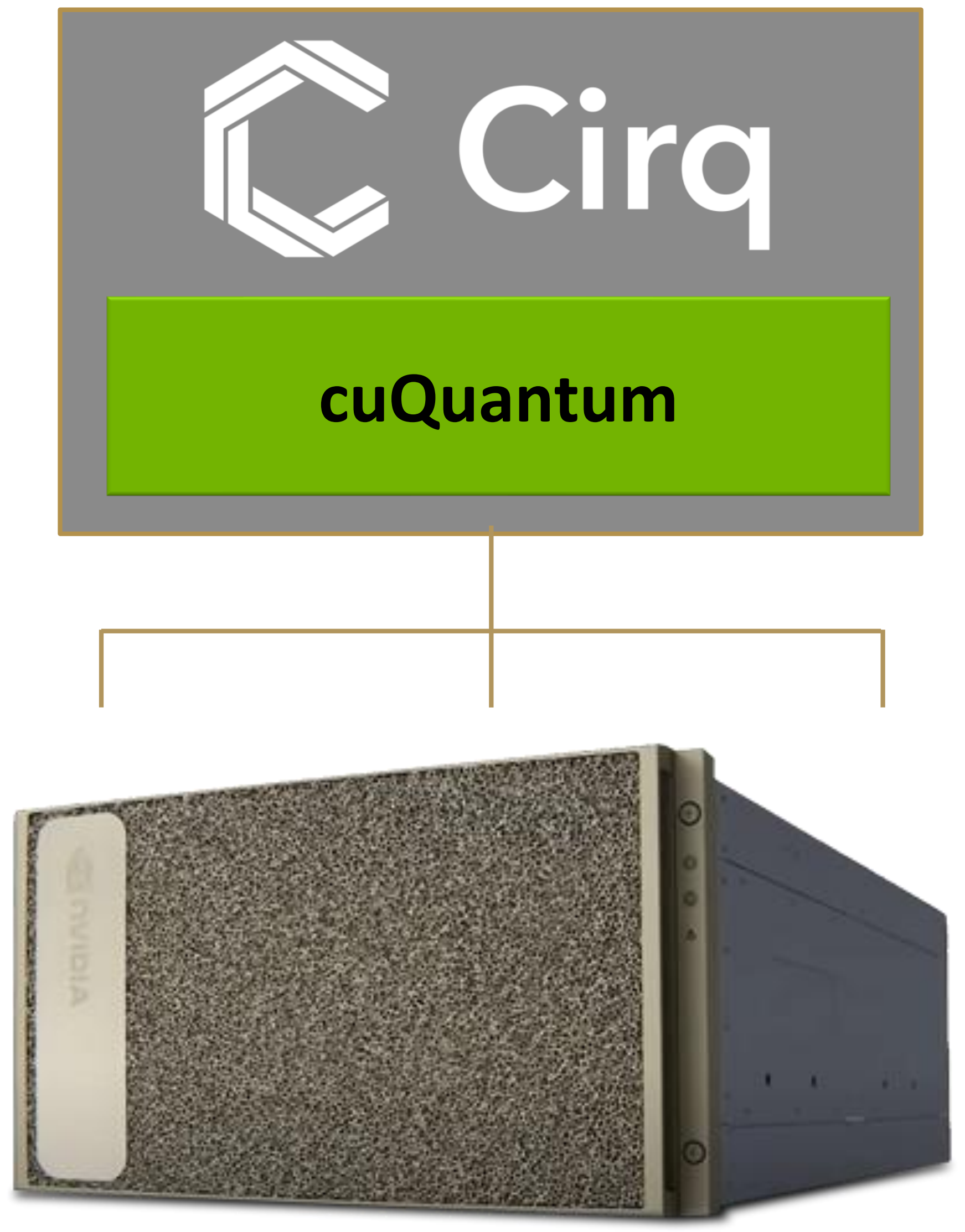

$R(\alpha_1^1, \beta_1^1, \gamma_1^1)$
$R(\alpha_2^1, \beta_2^1, \gamma_2^1)$
$R(\alpha_3^1, \beta_3^1, \gamma_3^1)$
$R(\alpha_4^1, \beta_4^1, \gamma_4^1)$
$R(\alpha_1^2, \beta_1^2, \gamma_1^2)$
$R(\alpha_2^2, \beta_2^2, \gamma_2^2)$
$R(\alpha_3^2, \beta_3^2, \gamma_3^2)$
$R(\alpha_4^2, \beta_4^2, \gamma_4^2)$

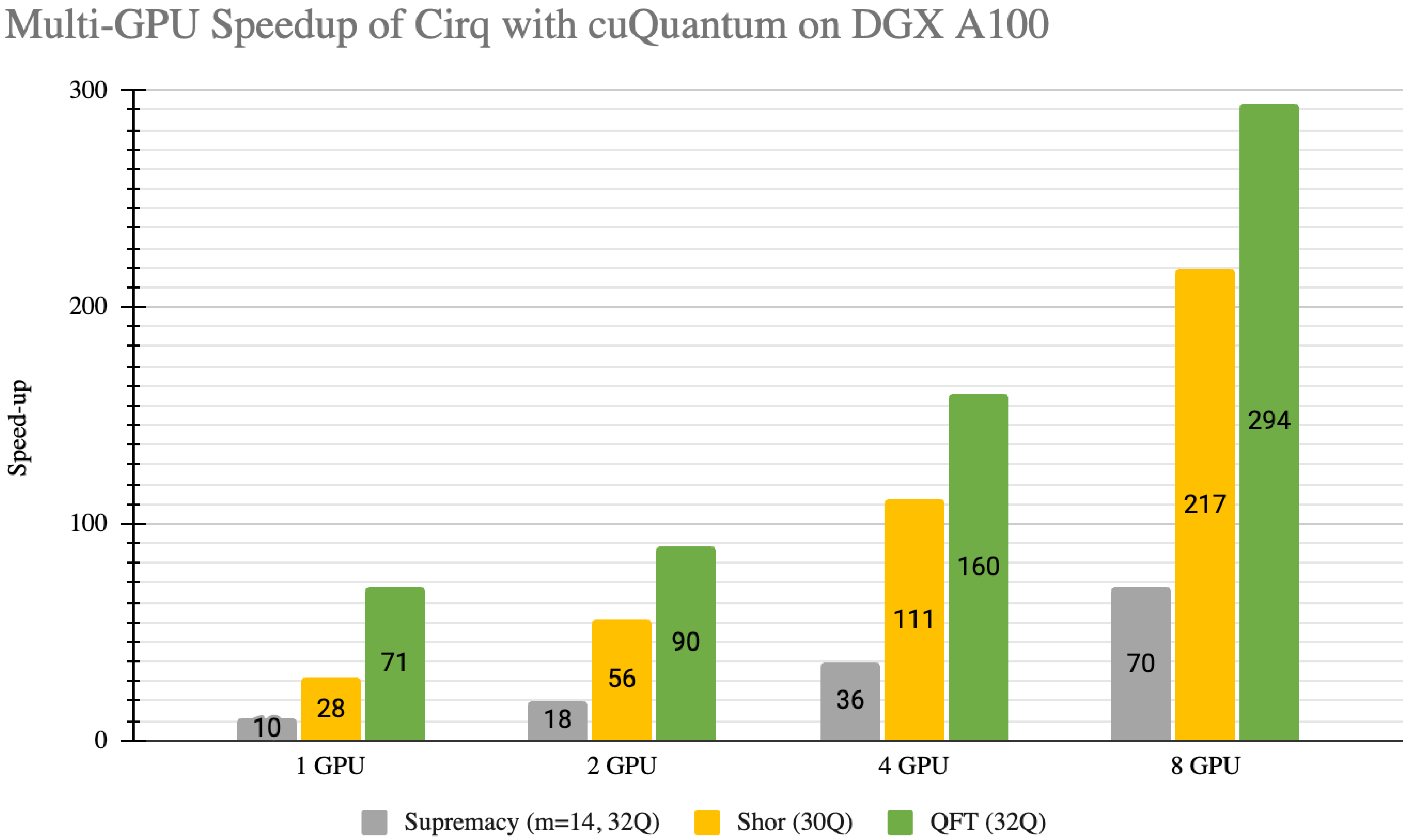


# DGX cuQuantum Appliance

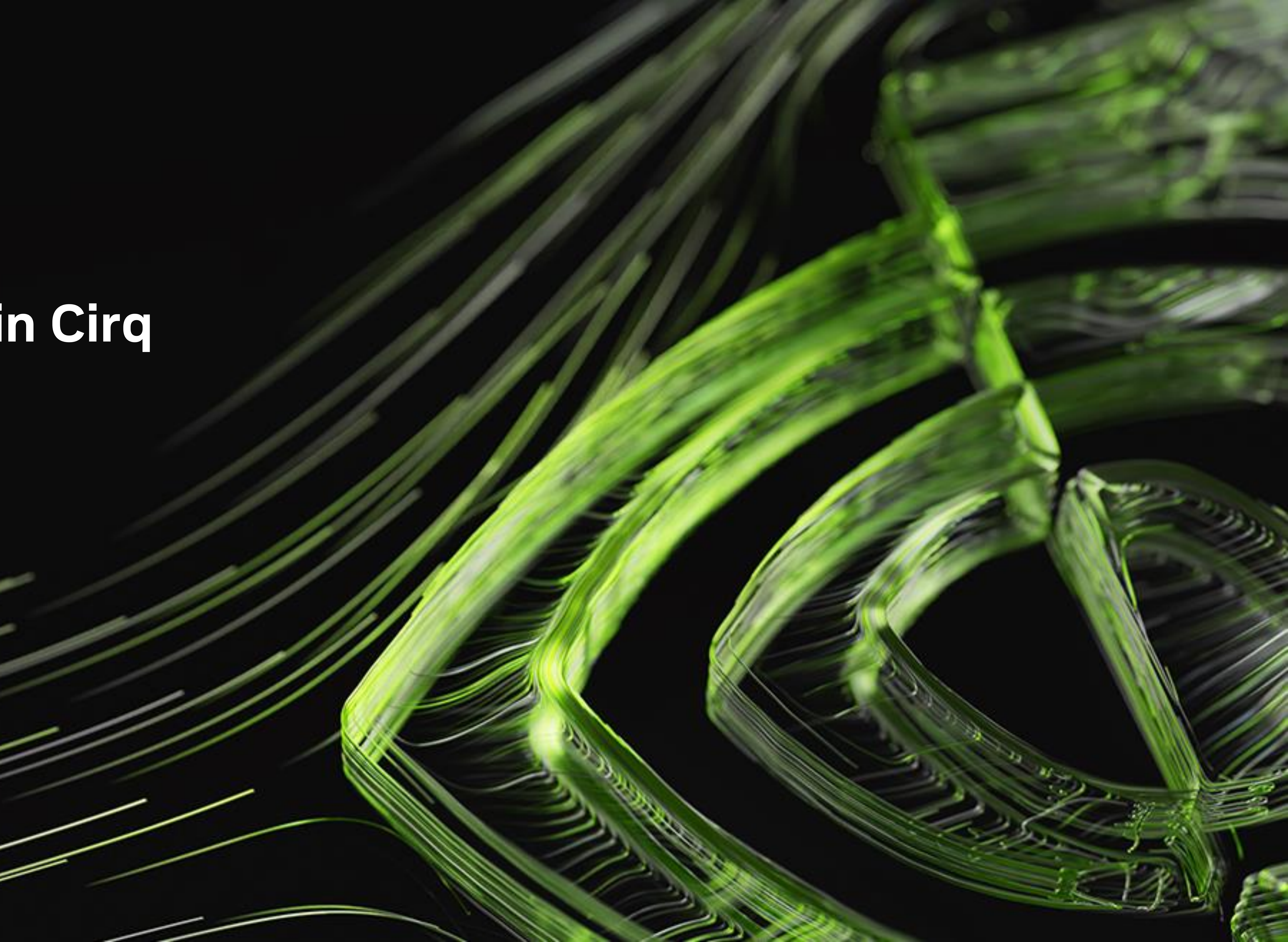
Multi-GPU container with cuQuantum + integrated Cirq/Qsim

- Full Quantum Simulation stack with a Cirq/Qsim frontend
  - other frontends will be available in future releases
- World class performance on key quantum algorithms, multi-GPU optimized
- Available now on NGC: catalog.ngc.nvidia.com/orgs/nvidia/containers/cuquantum-appliance





# Demo 1:
# cuQuantum in Cirq



# Demo 2:
# QML with PennyLane

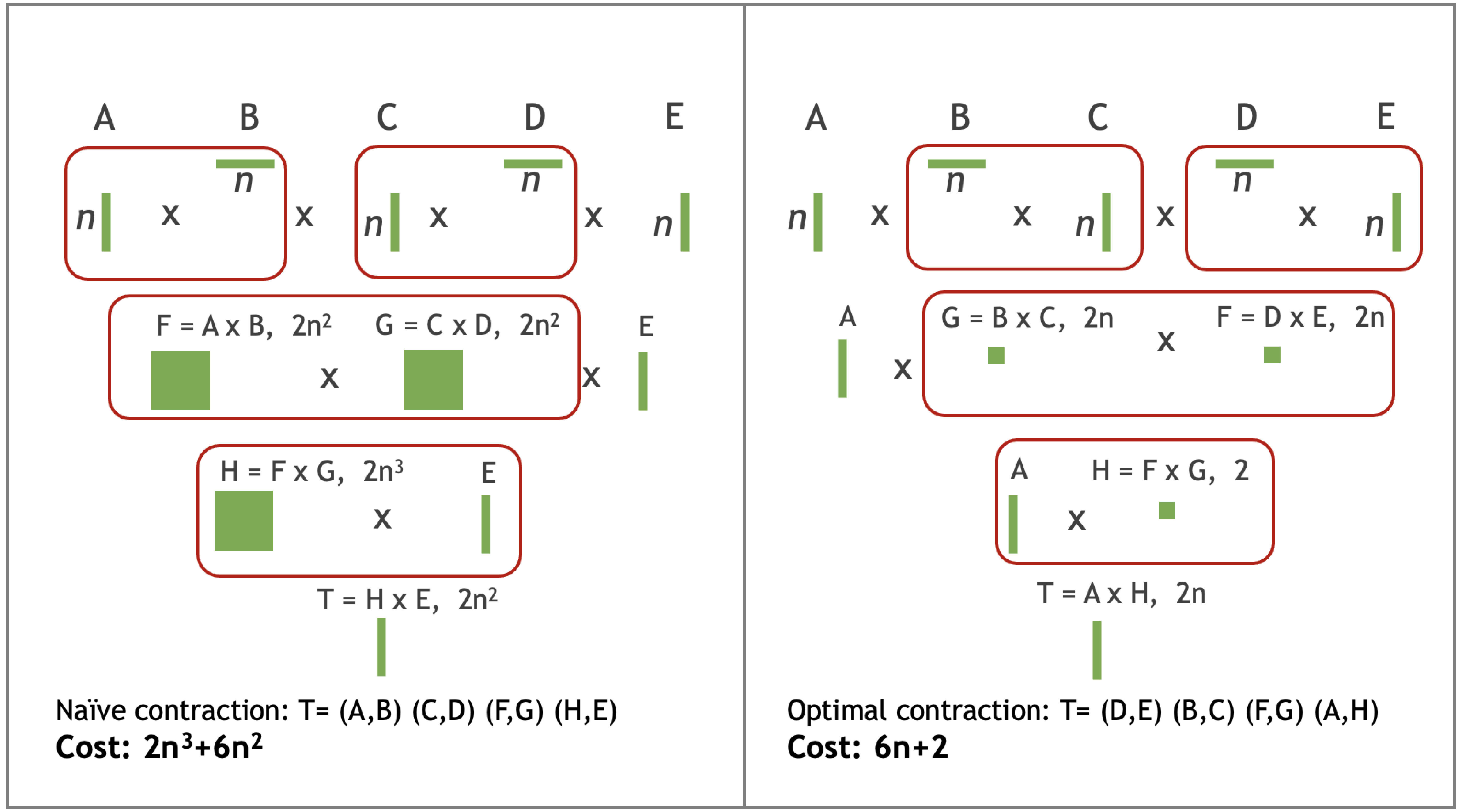
## Demo 3:
## VQE circuit with cuStateVec

# Tensor Networks & MaxCut

# cuTensorNet

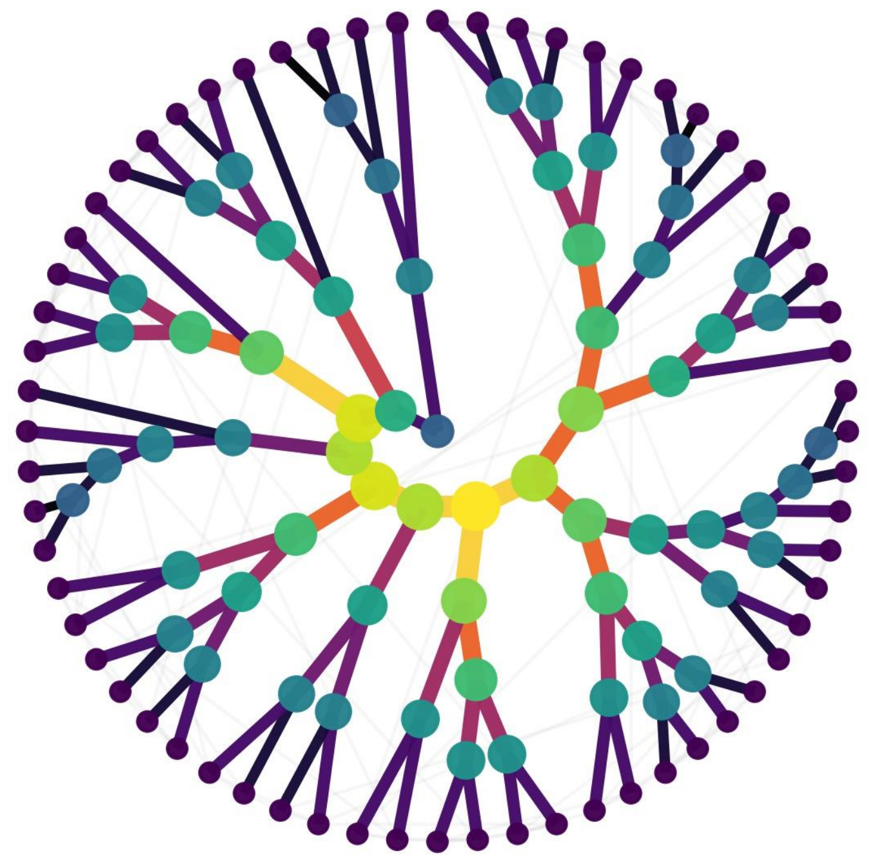
## A library to accelerate Tensor Network based Quantum Circuit simulation

- For many practical quantum circuits, tensor networks enable scaling of simulation to 100s or 1000s of qubits
- cuTensorNet provides APIs to:
  - convert a circuit written in Cirq or Qiskit to a tensor network
  - calculate an optimal path for the contraction
    - hyper-optimization is used to find contraction path with lowest total cost (eg FLOPS or time estimate)
    - slicing is introduced to create parallelism or reduce maximum intermediate tensor sizes
  - calculate an execution plan and execute the TN contraction
    - leverages cuTENSOR heuristics
- Checkout technical blogpost on NVIDIA Devblog: developer.nvidia.com/blog/scaling-quantum-circuit-simulation-with-cutensornet



Naïve contraction: T= (A,B) (C,D) (F,G) (H,E)
**Cost: $2n^3+6n^2$**

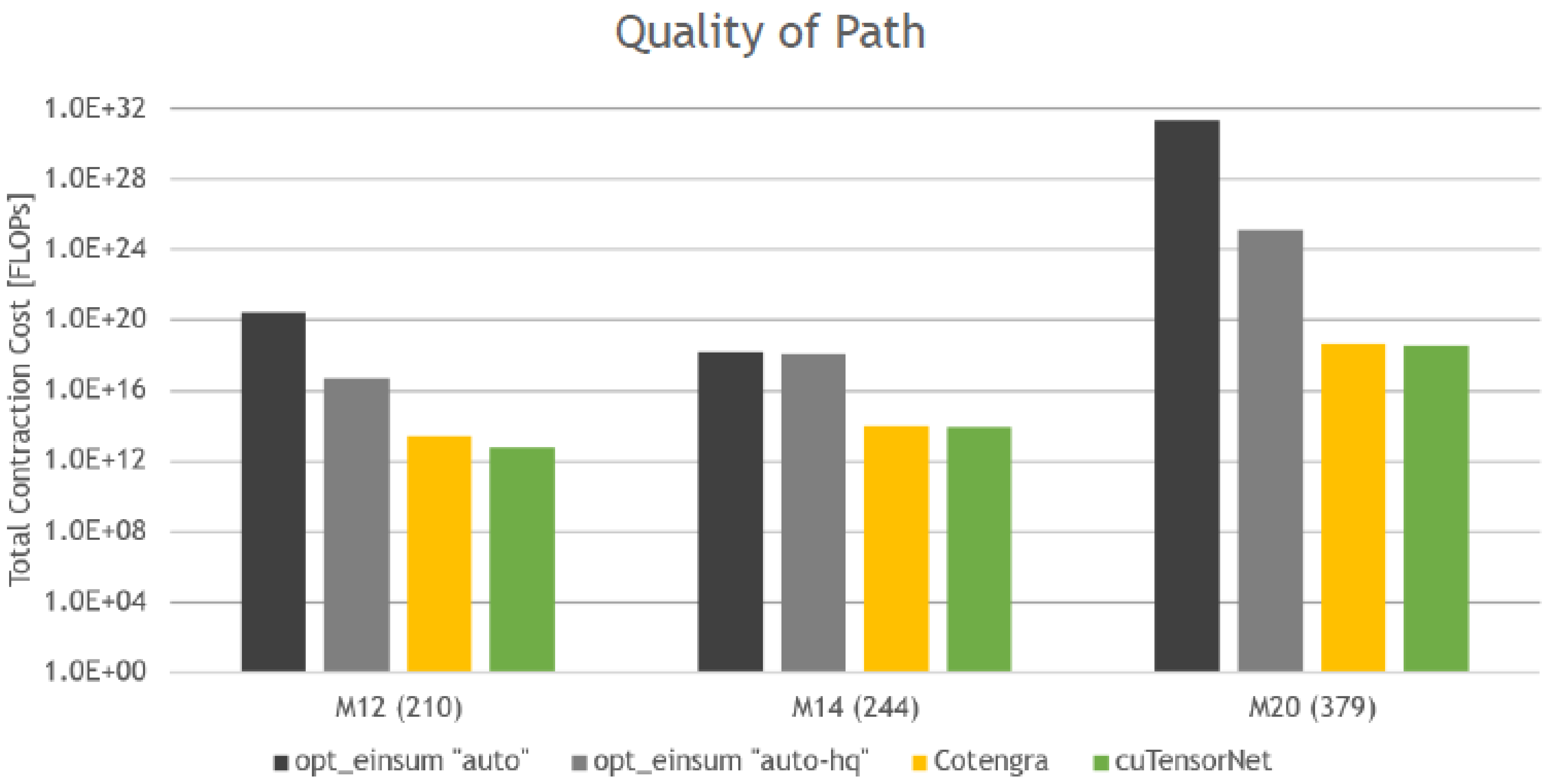
Optimal contraction: T= (D,E) (B,C) (F,G) (A,H)
**Cost: 6n+2**

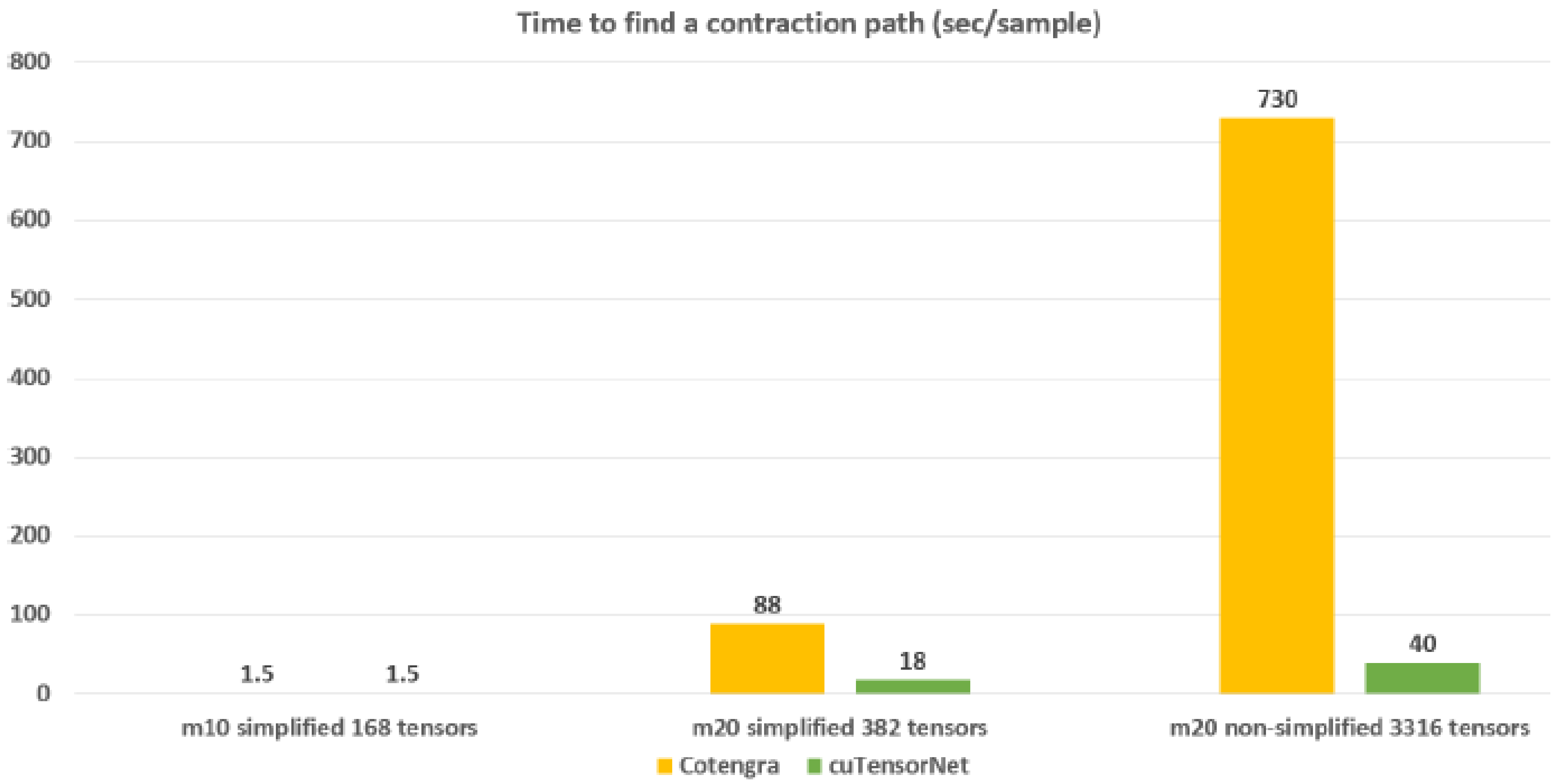


*Source of Tensor Network image: Quimb -* quimb.readthedocs.io

# cuTensorNet

## Tensor Network path optimization performance

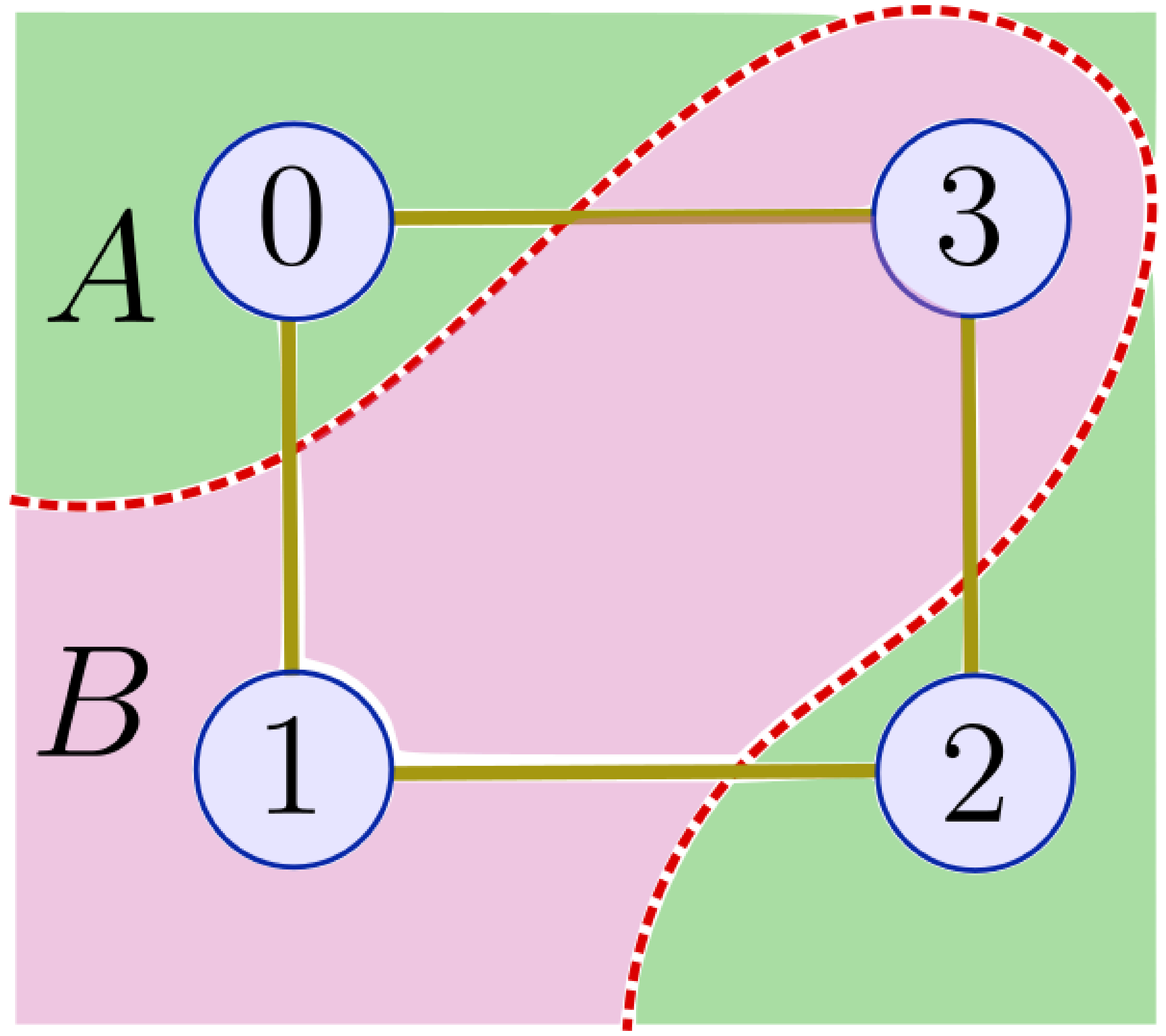


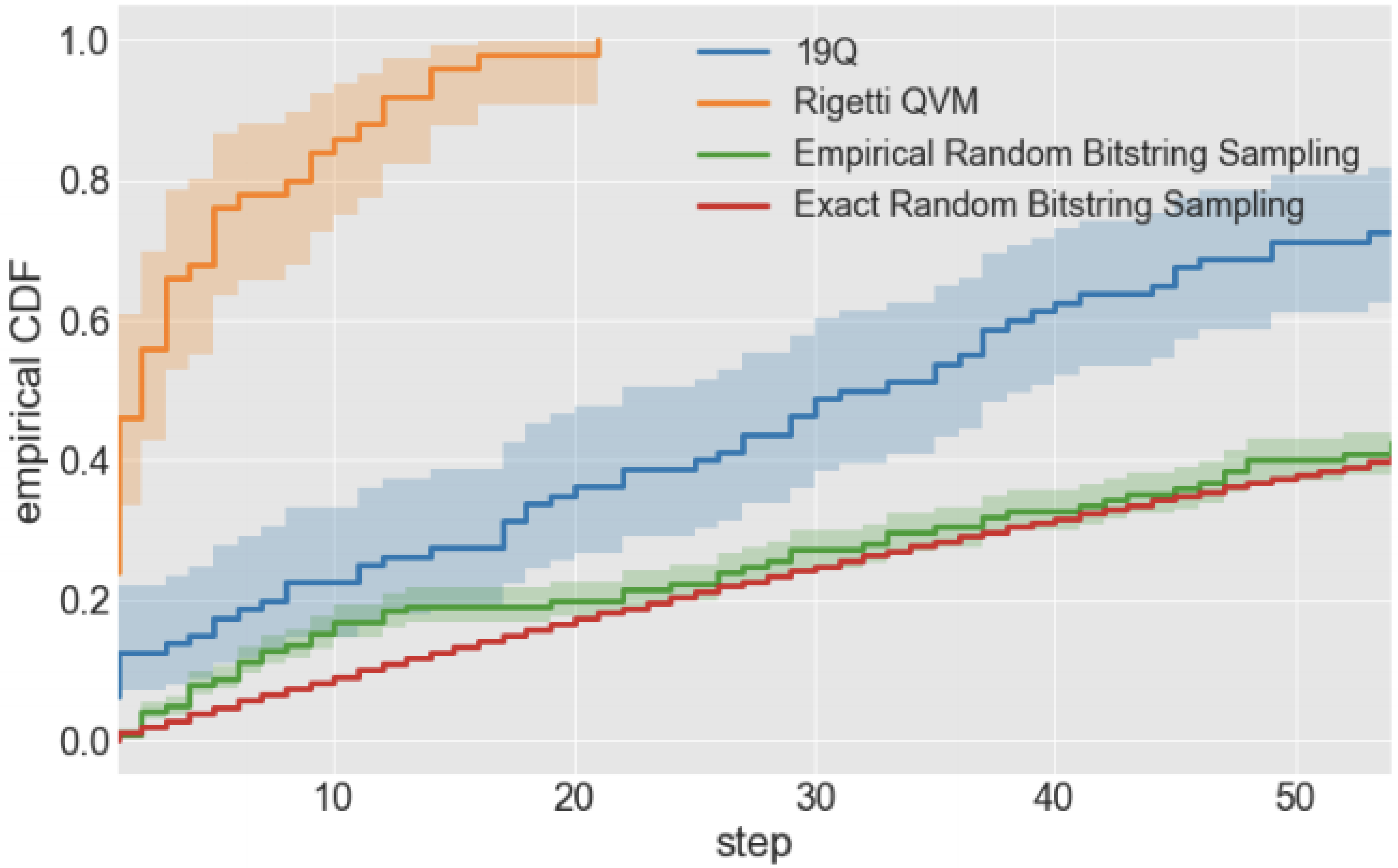


**cuTensorNet achieves SotA pathfinding results dramatically faster, and does better with more complex networks**

[1] Gray & Kourtis, Hyper-optimized tensor network contraction, 2021. URL: quantum-journal.org/papers/q-2021-03-15-410/pdf
[2] opt-einsum, URL: pypi.org/project/opt-einsum

# The MaxCut Problem





- NP-Complete combinatorial optimization problem
- Applications include clustering, network design, Statistical Physics, and more

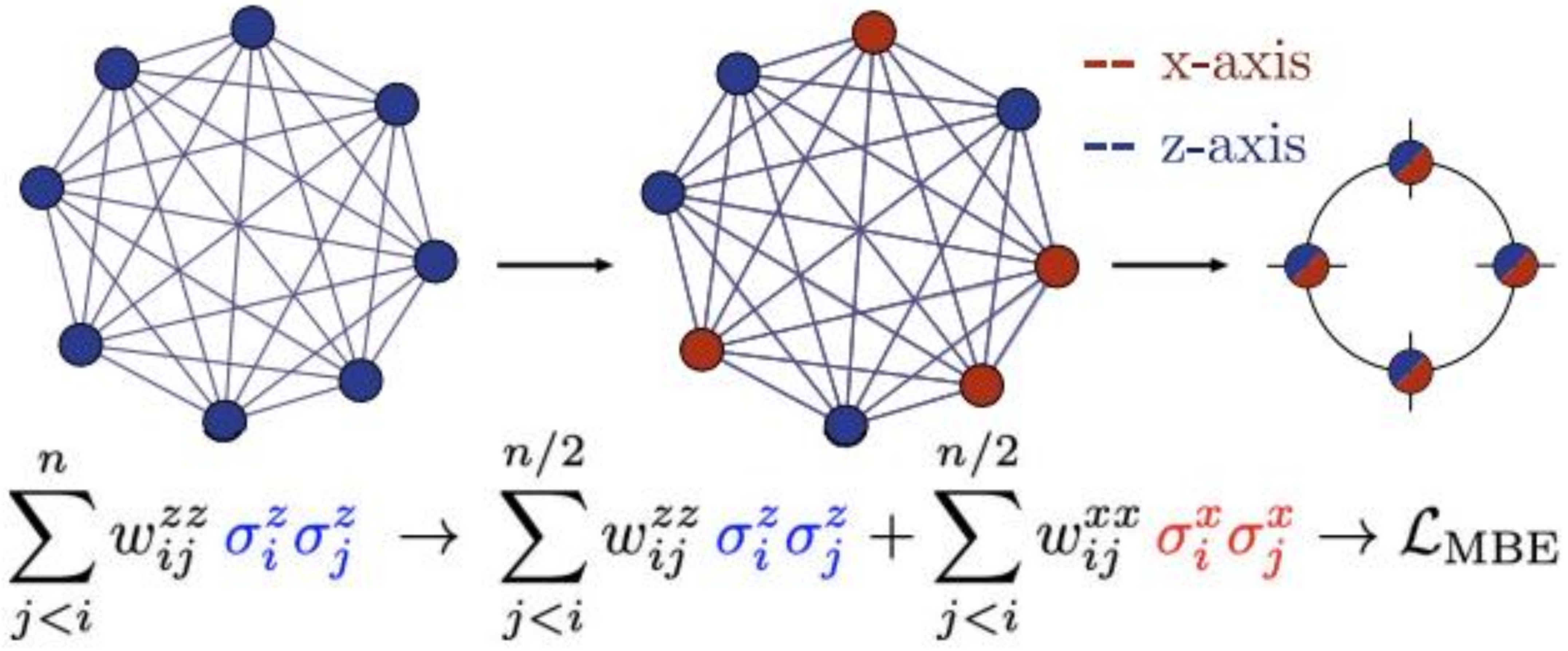
- Early target for hybrid variational quantum algorithms
- QAOA proposed by Farhi et al: arXiv:1411.4028
- Several HW demonstrations, including on Rigetti 19Q chip in 2017

*Source of images:* Xanadu Inc. - pennylane.ai/qml/demos/tutorial_qaoa_maxcut.html (left) | Otterbach et al, Unsupervised Machine Learning on a Hybrid Quantum Computer, arXiv:1712.05771

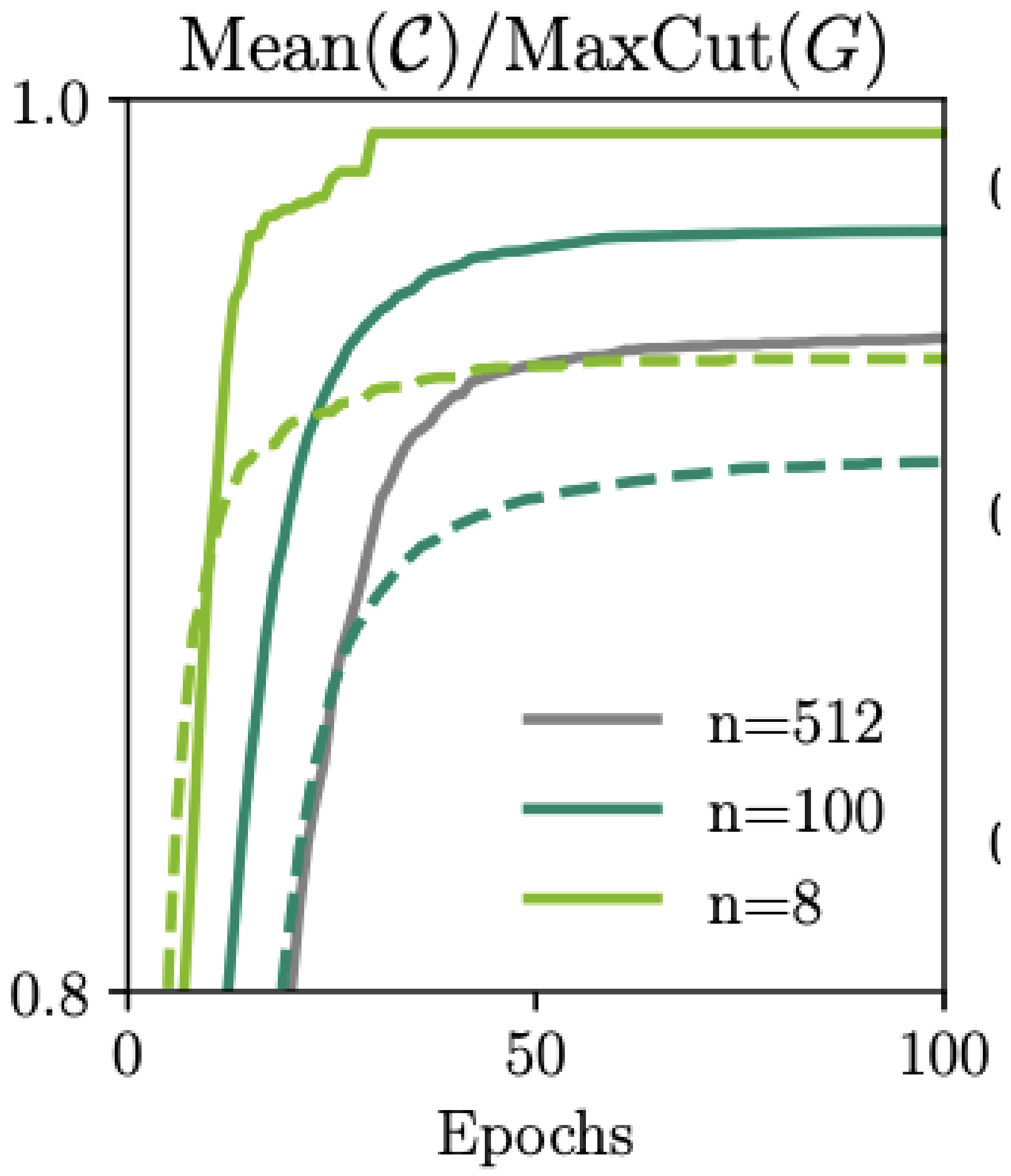
# Simulating MaxCut using Tensor Networks

- Tensor Networks are a natural fit for MaxCut
  - Fried et. al. (2017) [arxiv.org/abs/1709.03636](arxiv.org/abs/1709.03636)
  - Huang et. al (2019) [arxiv.org/abs/1909.02559](arxiv.org/abs/1909.02559)
  - Lykov et. al. (2020) [arxiv.org/abs/2012.02430](arxiv.org/abs/2012.02430)

- Patti et. al.(2021): NVIDIA Research proposes a novel variational quantum algorithm
  - Based on 1D tensor ring representation
  - Multibasis encoding
  - Able to find accurate solution for 512 vertices (256 qubits) on a single GPU
  - Paper: [arxiv.org/abs/2106.13304](arxiv.org/abs/2106.13304)
  - Code: [github.com/tensorly/quantum](github.com/tensorly/quantum)



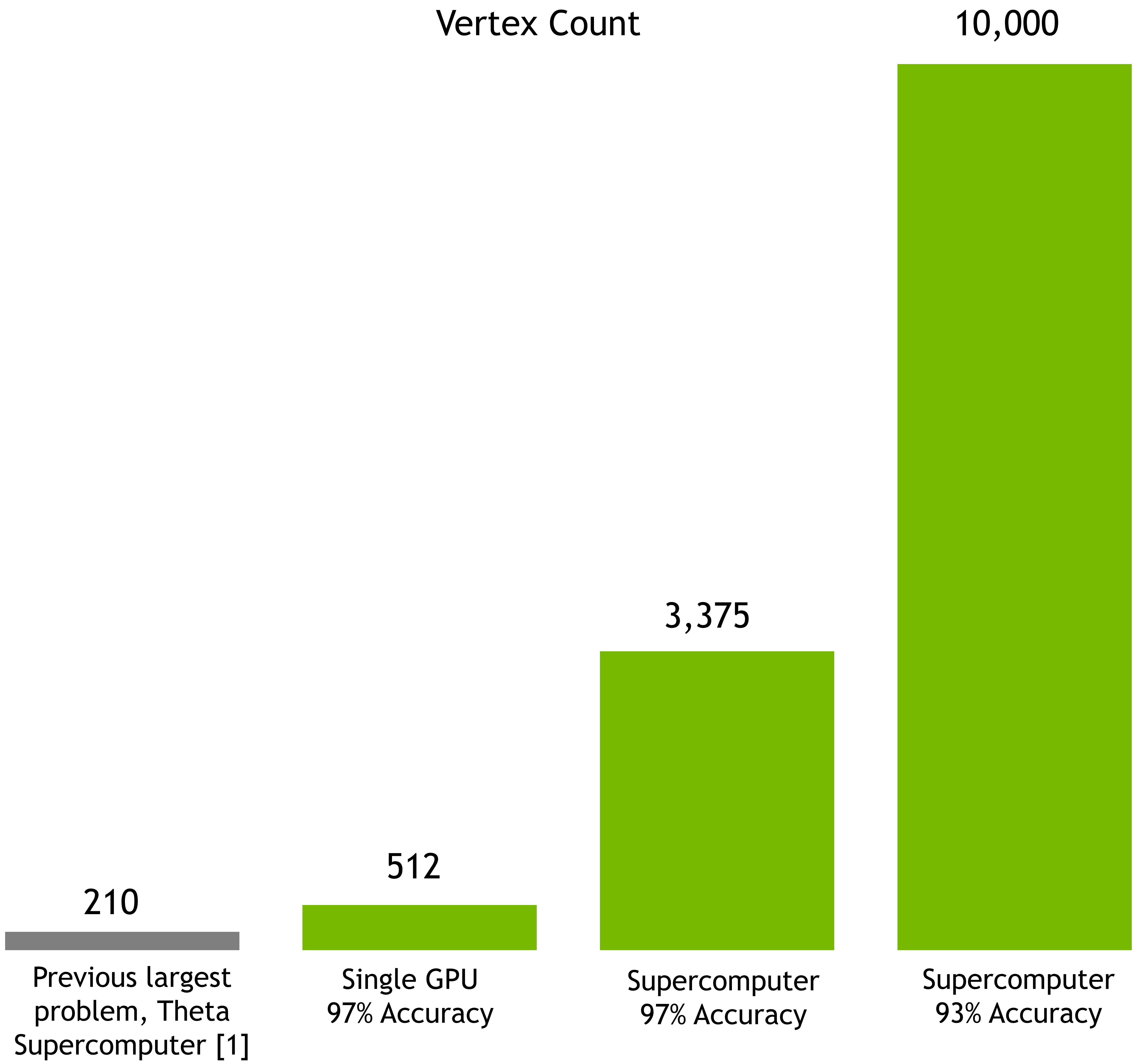
$$\sum_{j<i}^{n} w_{ij}^{zz}\,\sigma_i^z\sigma_j^z \rightarrow \sum_{j<i}^{n/2} w_{ij}^{zz}\,\sigma_i^z\sigma_j^z + \sum_{j<i}^{n/2} w_{ij}^{xx}\,\sigma_i^x\sigma_j^x \rightarrow \mathcal{L}_{\mathrm{MBE}}$$



# Scaling to a GPU Supercomputer: NVIDIA DGX SuperPOD

NVIDIA's Selene DGX SuperPOD based supercomputer

- Using NVIDIA's Selene supercomputer
- Solved a 3,375 vertex problem (1,688 qubits) with 97% accuracy
- Solved a 10,000 vertex problem (5,000 qubits) with 93% accuracy



[1] Danylo Lykov et al, Tensor Network Quantum Simulator With Step-Dependent Parallelization, 2020 arxiv.org/abs/2012.02430

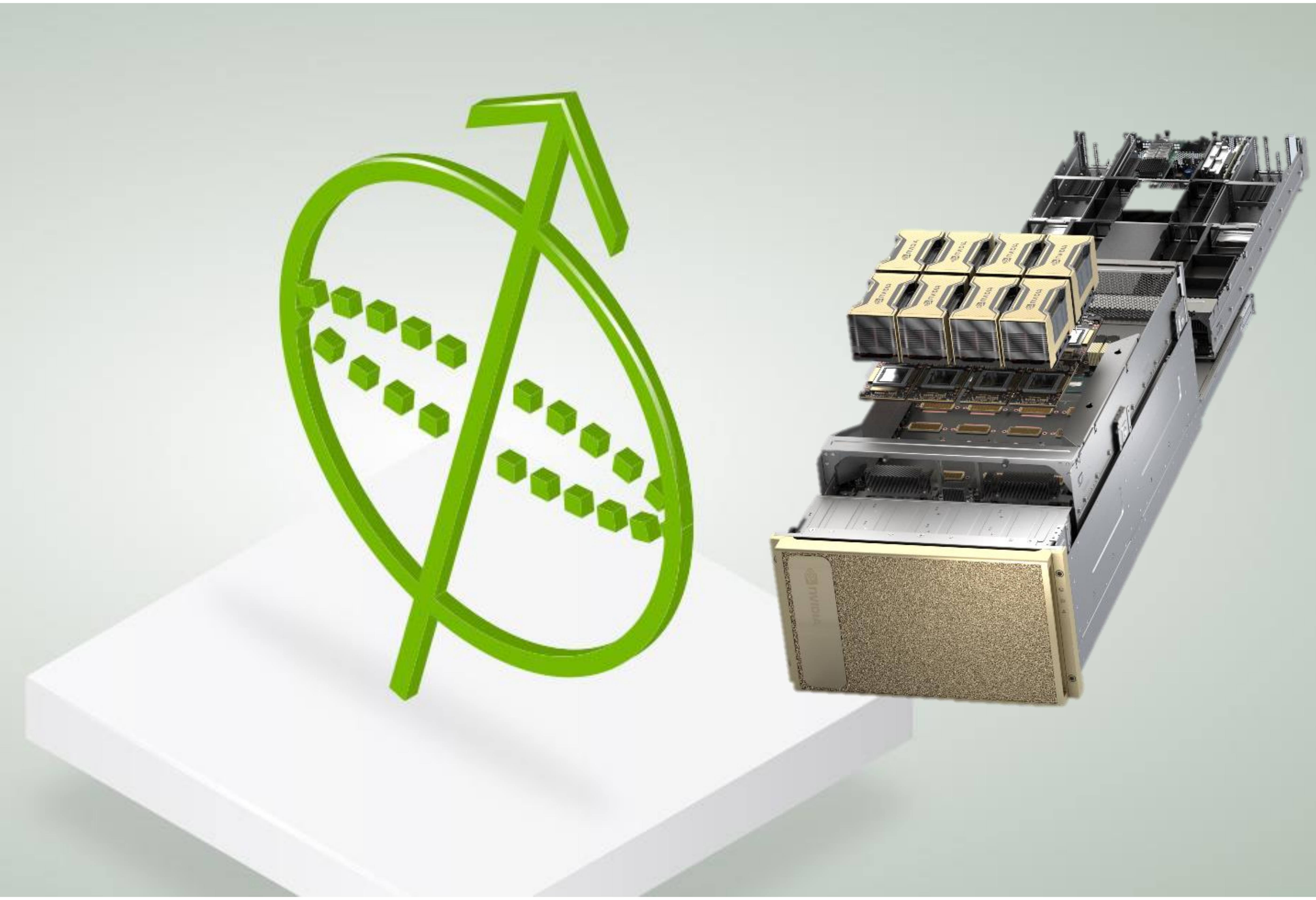
# Summary

- Quantum circuit simulation is an approach to conduct quantum computation with classical computer processors like CPUs and GPUs

- cuQuantum makes it easy for anyone with NVIDIA hardware to accelerate and scale their simulations more than previously possible

- An expanding ecosystem is using cuQuantum to enable quantum research

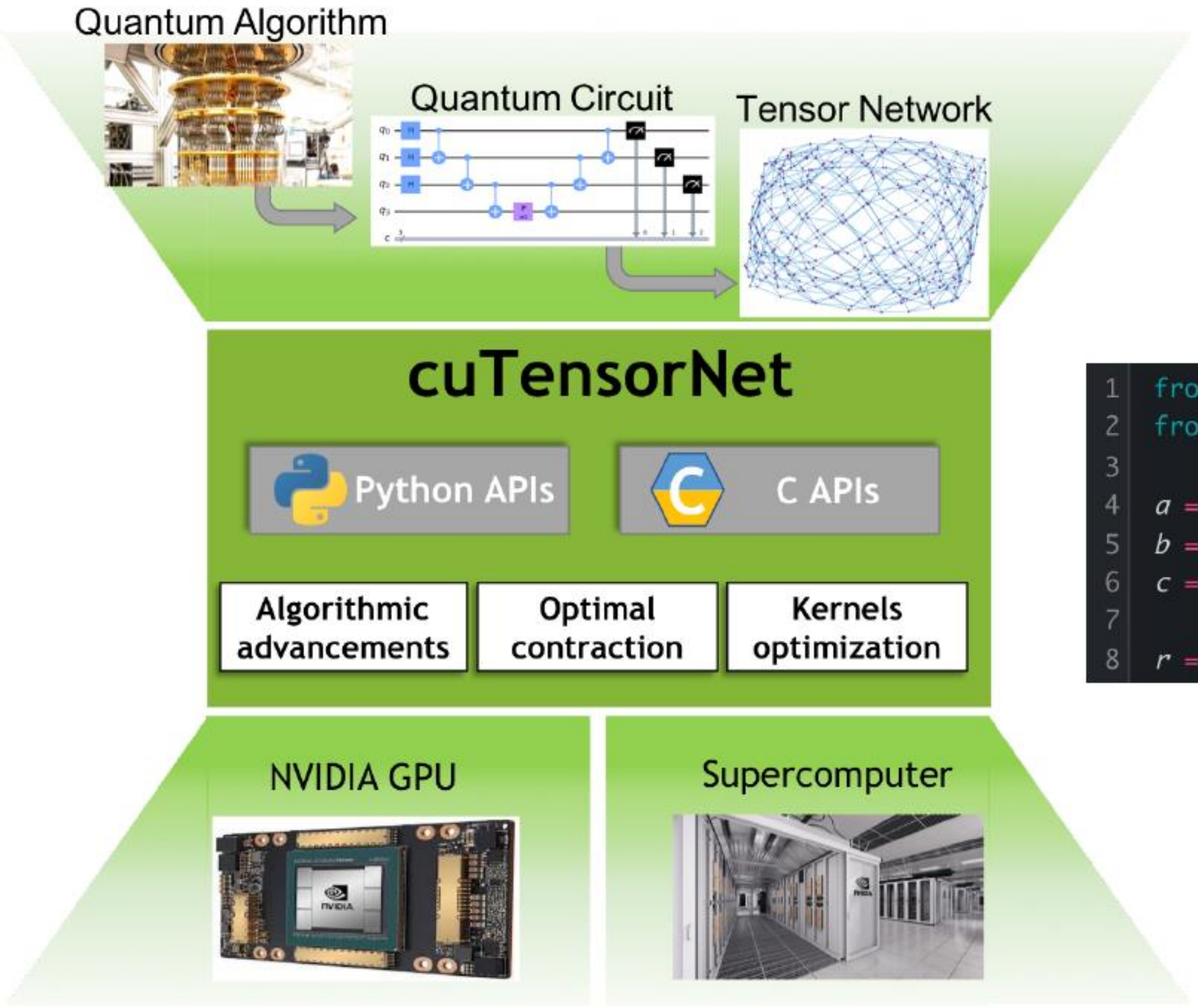
- Get stated with cuQuantum today by pulling our container from NGC, downloading the SDK from our DevZone, via pip or conda install, or through other frameworks

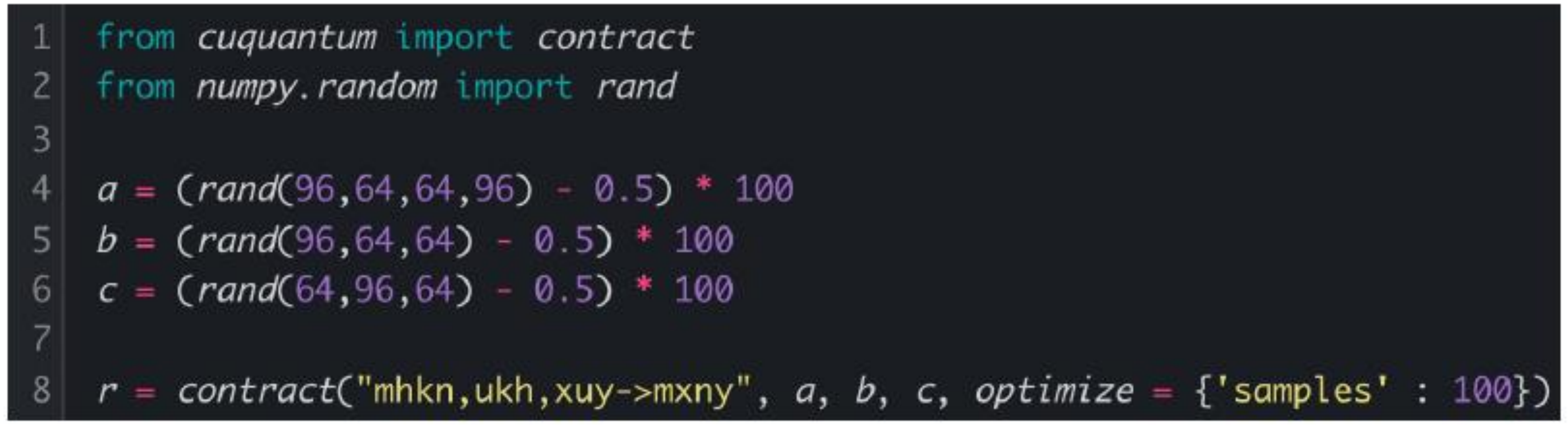


# Tensor Networks & cuTensorNet

# cuTensorNet

A library to accelerate Tensor Network based Quantum Circuit simulation





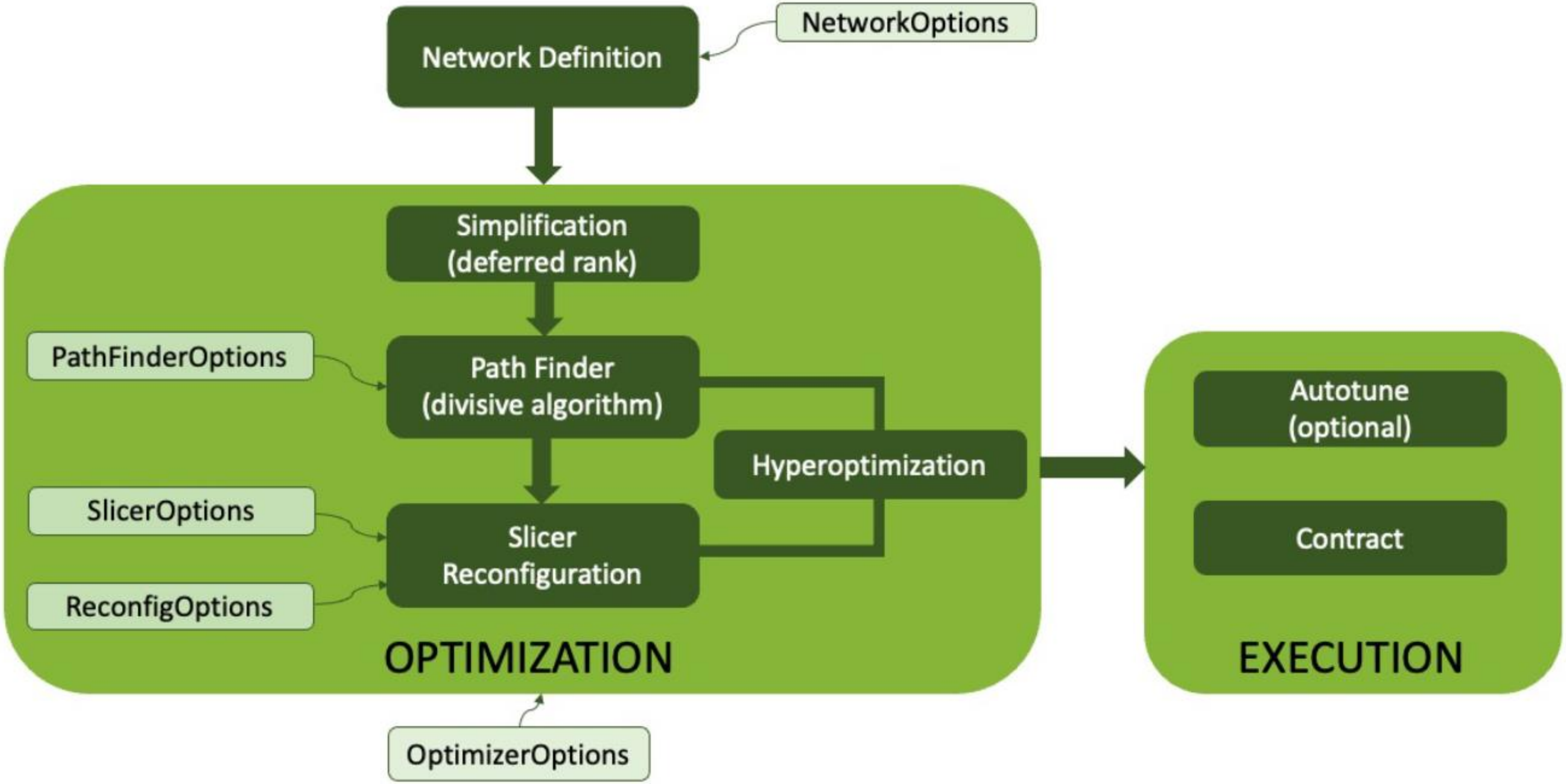
```
from cuquantum import contract
from numpy.random import rand

a = (rand(96,64,64,96) - 0.5) * 100
b = (rand(96,64,64) - 0.5) * 100
c = (rand(64,96,64) - 0.5) * 100

r = contract("mhkn,ukh,xuy->mxny", a, b, c, optimize = {'samples' : 100})
```

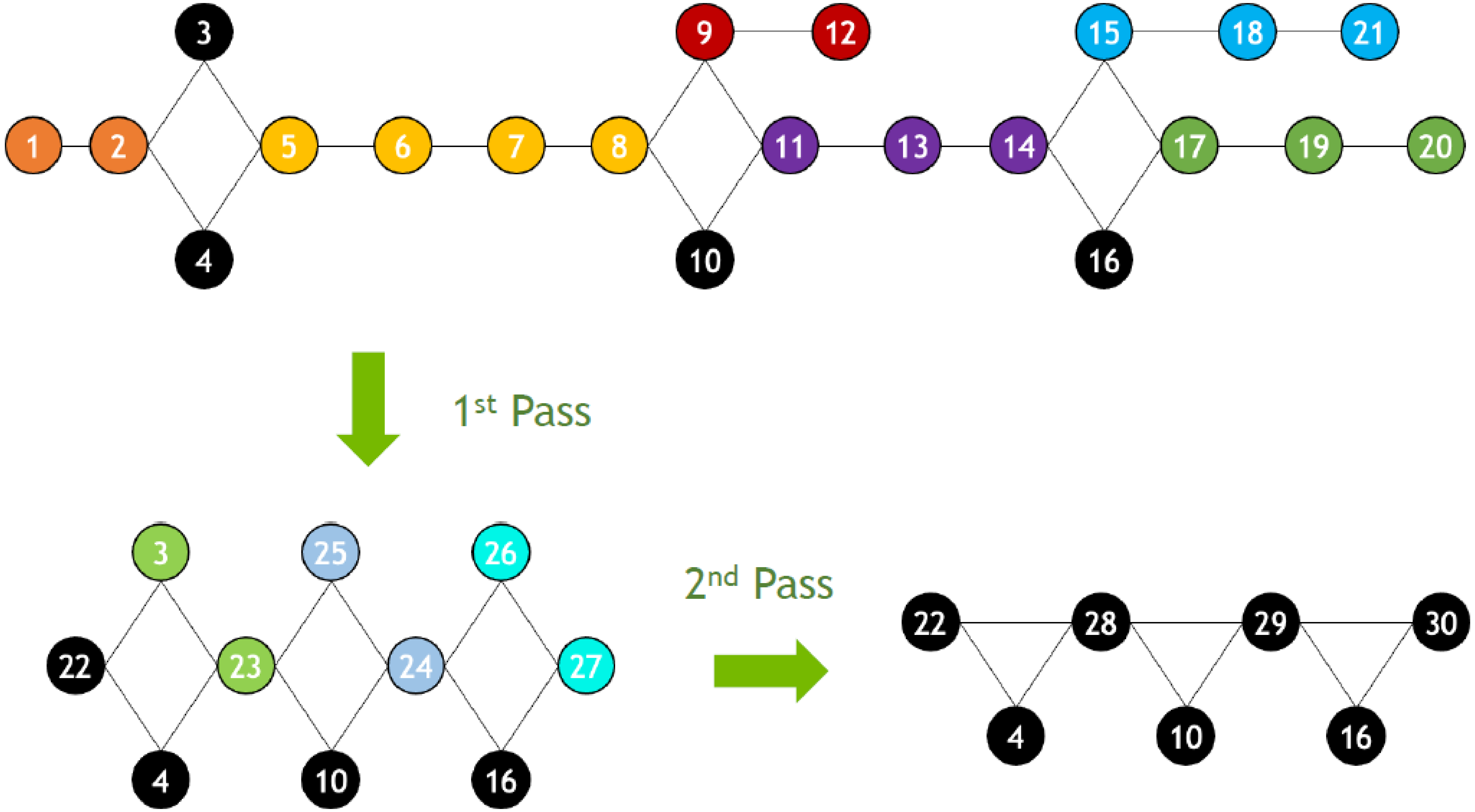
developer.nvidia.com/blog/scaling-quantum-circuit-simulation-with-cutensornet

# cuTensorNet Optimization & Flowchart



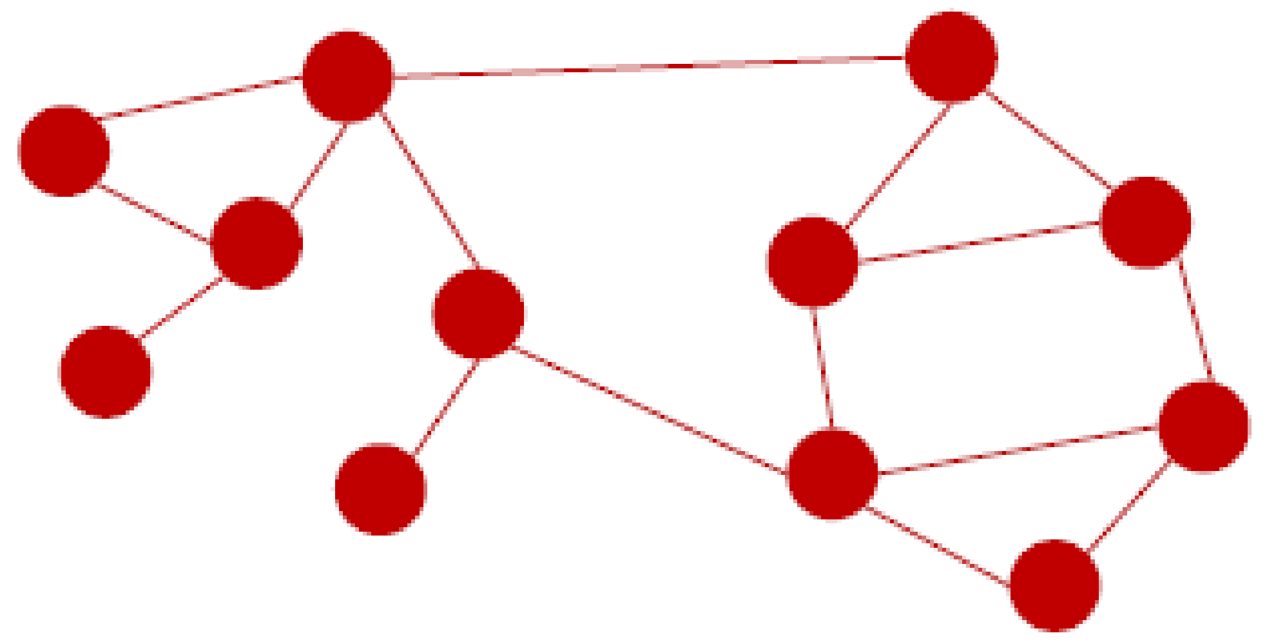
# Tensor Network Simplification

- Simplification aims to reduce the computational cost of contracting the tensor network through preprocessing.
- cuTensorNet implements deferred rank-simplification, which identifies those pairwise contractions that do not increase the rank (number of dimensions) of the resulting tensor and sequences them to be performed first as a path prefix. This essentially creates a smaller network for the divisive algorithm as well as for reconfiguration to process.

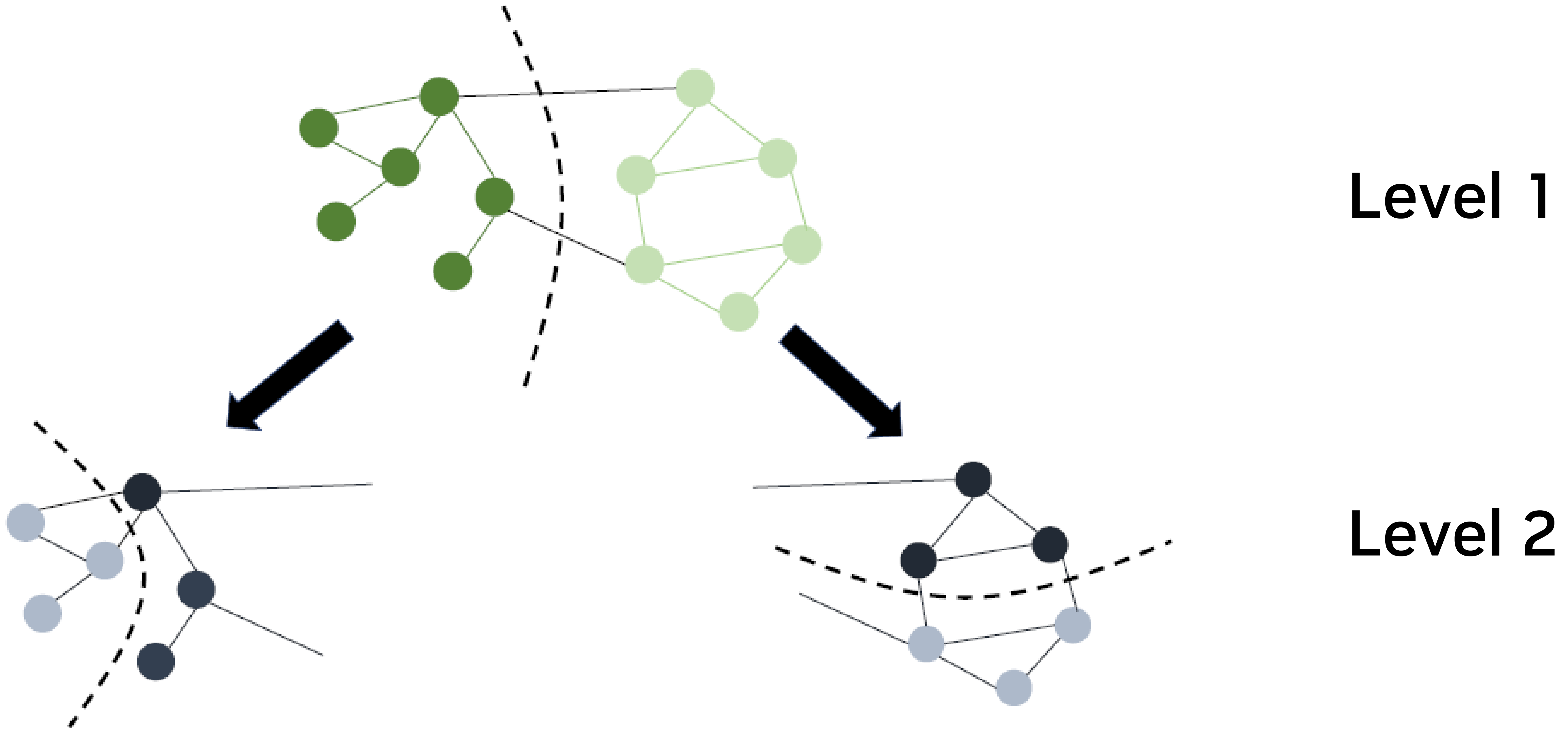


# cuTensorNet Path Finder (Divisive Algorithm)

- The tensor network is represented as a graph, with tensors as the vertices and modes that are contracted as the edges.



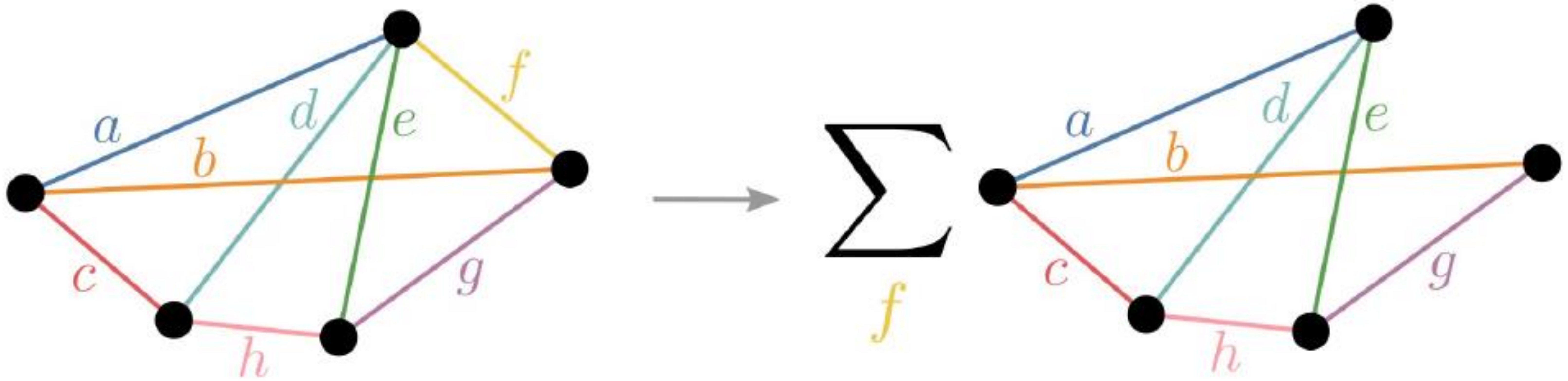
- The graph is partitioned into the specified number of partitions (2 shown) recursively until the size of each partition is less than or equal to the specified cutoff size (3 shown). Exhaustive search or an agglomerative algorithm is used to find the contraction order within as well as between partitions, from which the contraction order for the complete tensor network is built.



*The colors map to the partitioning level, and the shades at each level distinguish different partitions.*

# Tensor Network Slicing for Parallelism & Minimizing Memory Requirements

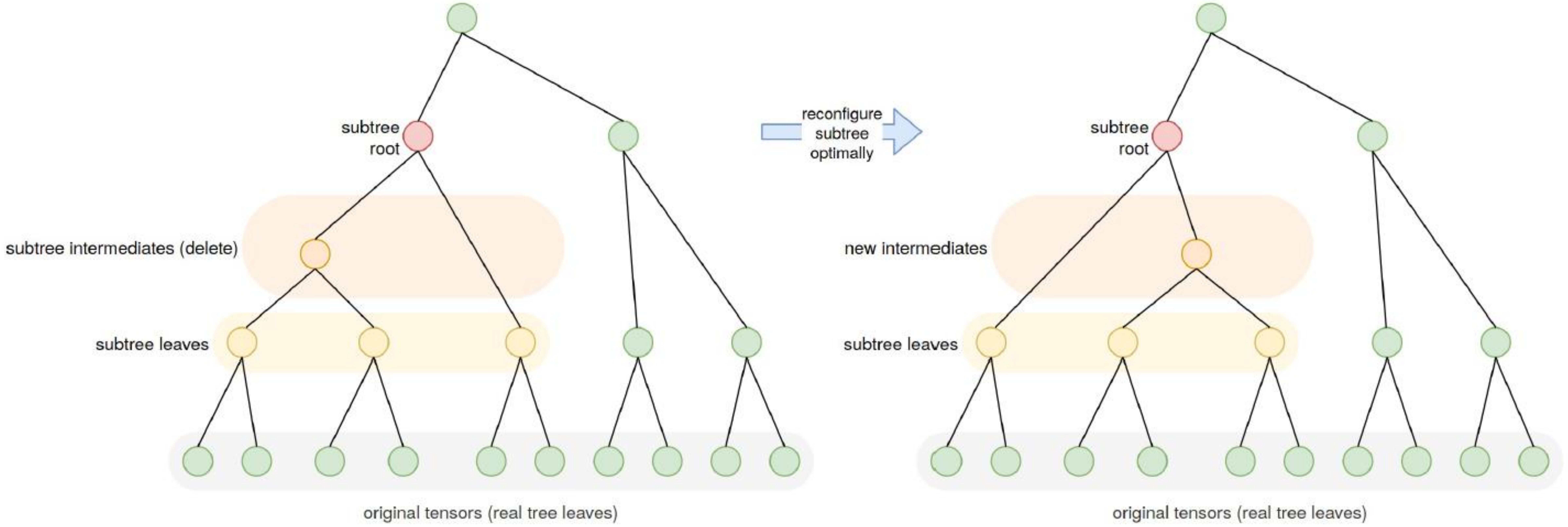
- *Slicing* is a technique to select a subset of edges from a tensor network (corresponding to mode labels) for explicit summation.
- A sliced network:
  - 1. results in lower memory requirements (often with some computational overhead), and
  - 2. allows for parallel execution.
- cuTensorNet implements *dynamic slicing*, which interleaves slicing with reconfiguration.



Source of image: github.com/jcmgray/cotengra by Johnnie Gray

# Tensor Network Reconfiguration

- The divisive algorithm computes a contraction path, which is a linearization of the contraction tree. The basic idea behind reconfiguration is to reduce the total contraction cost by reducing the contraction cost of portions (subtrees) of the contraction tree. The number of leaves in the subtree is typically chosen to be small enough so that the optimal algorithm can be used, and multiple iterations of reconfiguration are performed on different subtrees.
- As mentioned earlier, if slicing is active cuTensorNet interleaves reconfiguration with slicing to keep the contraction cost low.


subtree root
subtree intermediates (delete)
subtree leaves
original tensors (real tree leaves)
reconfigure subtree optimally
subtree root
new intermediates
subtree leaves
original tensors (real tree leaves)

*Source of image: github.com/jcmgray/cotengra by Johnnie Gray*

# Demo 4:
# VQE circuit with cuTensorNet

# Demo 5:
# M10 Sycamore circuit

# NVIDIA QODA

# Useful References

- cuQuantum SDK web page with download and conda install: developer.nvidia.com/cuquantum-sdk
- DGX Quantum Appliance container available on NGC (ngc.nvidia.com):
  - catalog.ngc.nvidia.com/orgs/nvidia/containers/cuquantum-appliance
  - includes Cirq and Qsim
- New PennyLane simulator *lightning.gpu* with cuQuantum support, available now from Xanadu:
  - xanadu.ai/products/lightning
- Full documentation at docs.nvidia.com/cuda/cuquantum
- cuStateVec technical article on NVIDIA Devblog:
  - developer.nvidia.com/blog/accelerating-quantum-circuit-simulation-with-nvidia-cuStateVec
- cuTensorNet technical article on NVIDIA Devblog:
  - developer.nvidia.com/blog/scaling-quantum-circuit-simulation-with-cutensornet
- Tensor Network contraction optimization paper:
  - Johnnie Gray and Stefanos Kourtis, "Hyper-optimized tensor network contraction“, Quantum, volume 5, 2021.
- What is a QPU? blogs.nvidia.com/blog/2022/07/29/what-is-a-qpu
- NVIDIA QODA: developer.nvidia.com/qoda
- QODA technical article on NVIDIA Devblog:
  - developer.nvidia.com/blog/introducing-qoda-the-platform-for-hybrid-quantum-classical-computing

# Qibo

# Thank you!

cnardone @ nvidia.com
ahehn @ nvidia.com
zchandani @ nvidia.com
andrea.pasquale @ unimi.it
stavros.efthymiou @ tii.ae