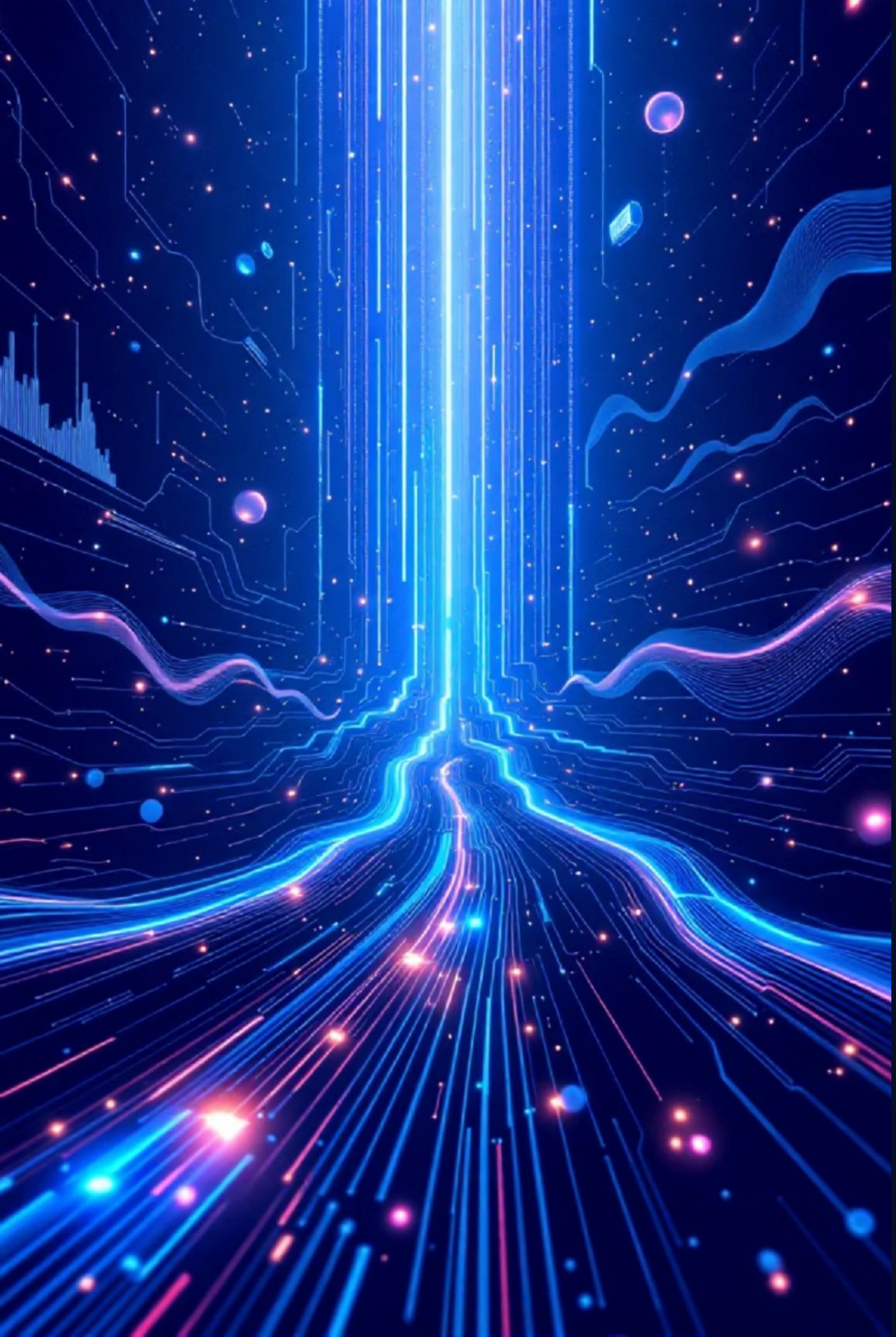


Hybrid Quantum-Classical Portfolio Optimization

A scalable approach to selecting optimal bond portfolios using quantum-inspired algorithms

Beerantum

The Challenge

Project Goal

Select an optimal portfolio of 10 bonds from 31 available assets using efficient computational methods.

The Problem

Portfolio optimization is NP-hard. With 31 assets, there are over 66 million possible combinations—computationally infeasible to check exhaustively.

Beerantum

Initial Approach: Direct QUBO

01

Binary Variables

Each asset represented as $x_i=1$ (selected) or $x_i=0$ (not selected)

02

Cost Function

Financial objectives translated into a mathematical function to minimize

03

Risk Targeting

Penalties added for deviating from target risk within credit quality buckets

04

QAOA Solver

Quantum Approximate Optimization Algorithm applied to find optimal solution

This method worked perfectly for 15 assets on a local machine, validating our QUBO formulation.

The Scaling Wall

Quadratic Growth

QUBO matrix complexity grows quadratically with asset count

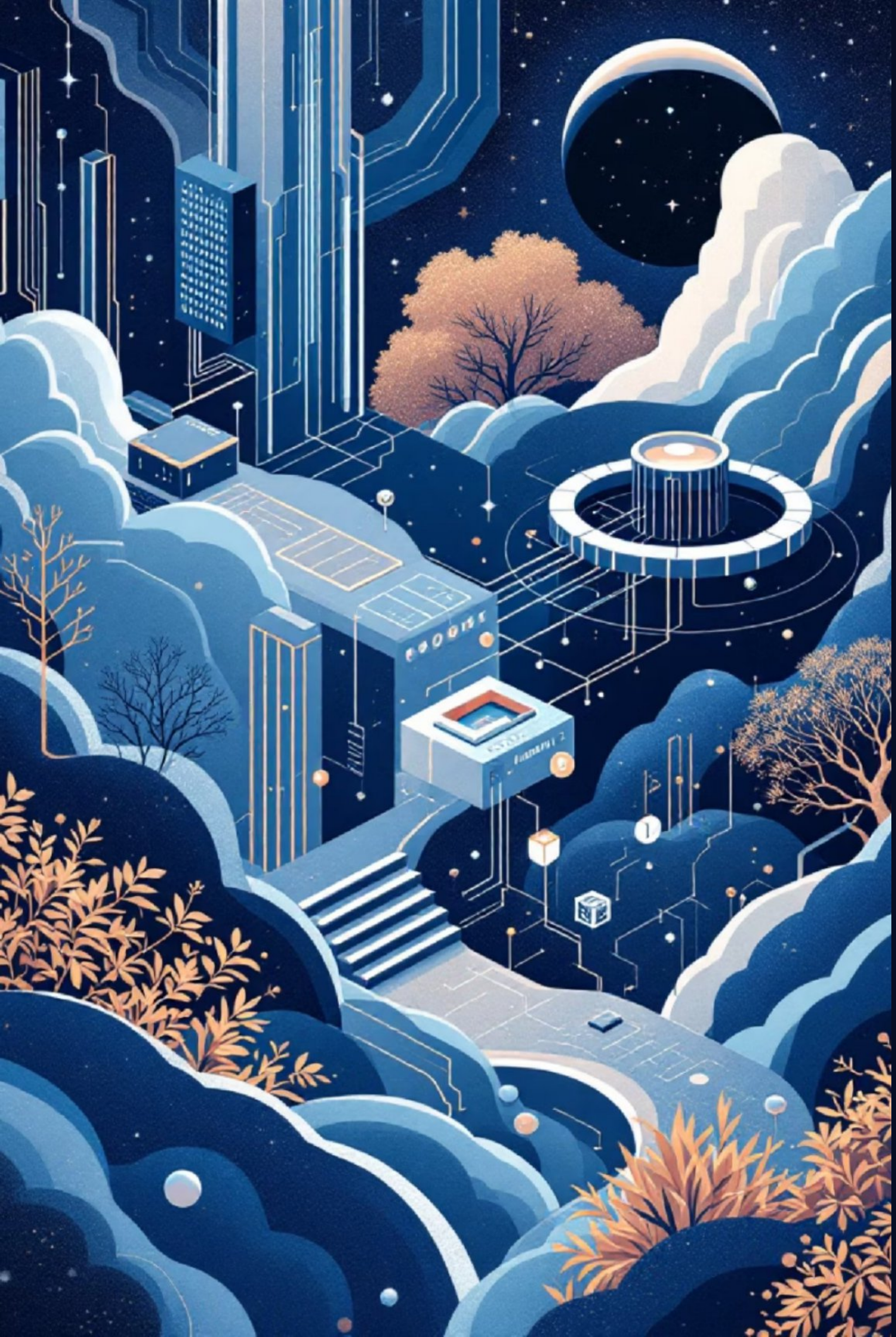
31-Qubit System

Simulating 31 qubits with dense interactions exhausted local machine resources

Kernel Crash

Memory and processing power insufficient for direct approach at scale

> **Key Insight:** Direct quantum algorithms aren't yet feasible for realistic problem sizes on near-term hardware. A smarter strategy was essential.



Our Solution: Divide and Conquer

A hybrid quantum-classical workflow that breaks large problems into solvable pieces, leveraging the strengths of both computational paradigms.

Divide

Classical ML partitions the problem

Conquer

Quantum algorithms solve sub-problems

Combine

Classical methods assemble final portfolio

Step 1: Divide & Cluster

Classical Machine Learning

K-Means clustering algorithm groups 31 assets into 5 distinct clusters based on financial characteristics.

Key Features

- Risk (spreadDur)
- Return (oas)

Outcome

Five independent sub-problems with 5-9 assets each, replacing one massive 31-asset problem.

31

Total Assets

5

Clusters

5-9

Assets per Cluster

Step 2: Conquer with Quantum

1

Build Mini-QUBO

Create a miniature QUBO model for each cluster with local objectives

2

Apply QAOA

Use Quantum Approximate Optimization Algorithm via OpenQAOA library

3

Find Champions

Identify the best assets within each cluster through quantum exploration

Why It Works: Simulating 5-9 qubit systems is computationally trivial, allowing quantum algorithms to explore complex correlations without system crashes.

Step 3: Combine & Finalize

1. **Collect Champions**

   Gather all winning assets from quantum runs into elite candidate pool (11-15 assets)

2. **Classical Selection**

   Sort by return-to-risk ratio using classical methods

3. **Final Portfolio**

   Select optimal 10-bond portfolio from qualified candidates

Implementation Workflow

01_classical_clustering.py

Loads 31 assets and partitions into 5 clusters using K-Means

03_generate_all_qubos.py

Automates creation of unique QUBO model for each cluster

05_solve_all_clusters_local.py

Loops through clusters and solves using OpenQAOA locally

06_assemble_and_visualize.py

Collects results, builds final portfolio, generates risk-return visualization

A Practical Path Forward

Proved Limitations

Direct quantum approaches aren't feasible for realistic problem sizes on current hardware

Designed Solution

Hybrid workflow successfully circumvents computational limitations through intelligent decomposition

Achieved Success

Solved full 31-asset problem efficiently on local machine by combining classical ML and quantum optimization