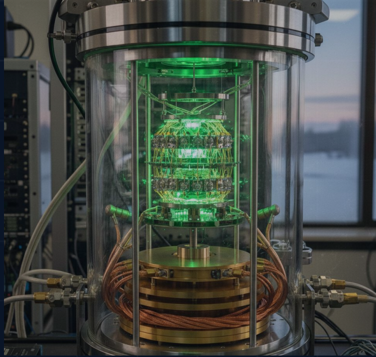


Quantum go-brr

Quantum-GPU Hybrid Acceleration for LABS

Team
Harvard Blocheads

Event
MIT-Iqhack 2026

Meet the Team

The Minds Behind the Project

Emmanuel Rassou
Computer Science/Stats

Project Lead
(Architect)

Tarun Sasirekha
Electrical Engineering

GPU Acceleration PIC
(Builder)

Anmay Gupta
Computer Science/Physics

Quality Assurance PIC
(Verifier)

Hugo Mackay
Physics/Math

Technical Marketing PIC
(Storyteller)

Problem & Motivation

The Low Autocorrelation Binary Sequence (LABS) Challenge

Fundamental approach to the LABS Problem

Inspired by the the first exercise to find patterns and

natural symmetries for small cases of N

Scaling Beyond Classical with Quantum Parallelism

Search space of 2^N grows exponentially. Classically intractable for

N > 60. Current best classical solvers use Memetic Tabu Search

(MTS).

Real-World Impact

Optimal sequences are critical for pulse compression in radar, sonar,

and spread-spectrum communications.

$$E(S) = \sum_{k=1}^{N-1} C_k(S)^2$$

$$C_k(S) = \sum_{i=1}^{N-k} s_i \, s_{i+k}$$

The Plan & The Pivot

From Initial Vision to Execution Reality

Original Approach

Algorithm

QAOA + CD (,DMRG,PCE)

Challenge

QAOA is vulnerable to high-energy adiabatic terms and requires deliberate control, and CD isn’t hardware efficient.

→

Adapted Strategy

Algorithm

Trotterization with Symmetry Exploitation

Ω

Solution

By using Trotterization, we can compute higher N’s with less compute and achieve safe optimizations over the tutorial.

Key Insight *By exploiting symmetries, we can claim easy advantages that allow us to accomplish much harder tasks with much less compute, saving both time and money.*

Quantum Approach I

Method 1: Enforcing Guaranteed Symmetries to Precondition Trotterization

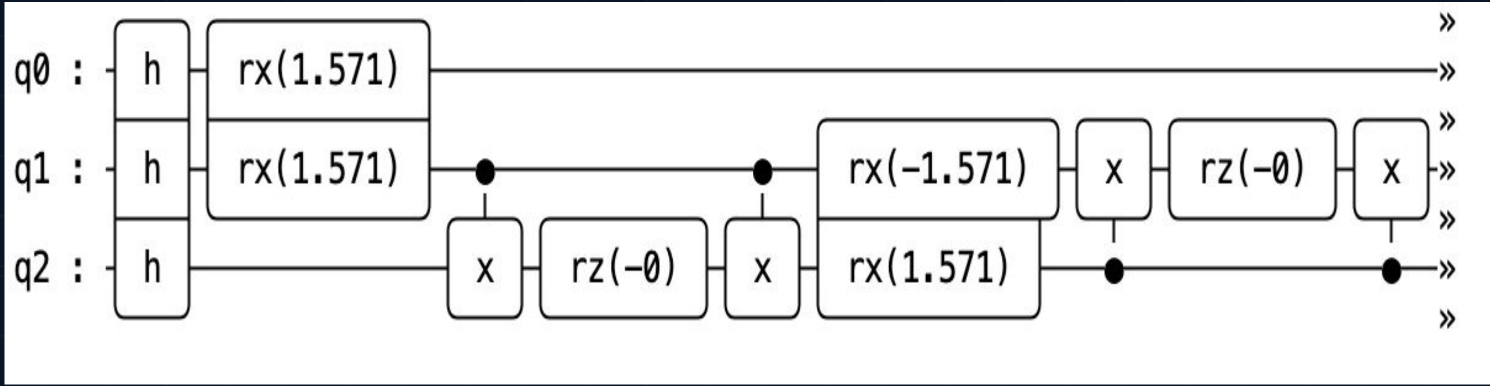
Core Concept

Two Symmetries $\Rightarrow$ set first two qubits to 1

- Flipping of even positions $\longrightarrow$ same energy
- Flipping of odd positions $\longrightarrow$ same energy

Implementation Details

Circuit Depth:	$O(n^2)$
Gate Complexity:	$O(n^3)$
Trotter Steps:	$k = 4$
Error Bound:	$\varepsilon < 10^{-3}$



System Architecture

Quantum-Classical Hybrid Pipeline

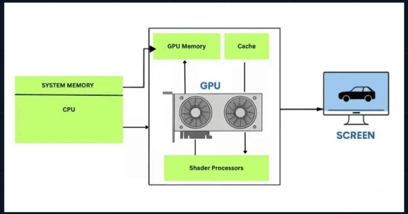
Quantum Layer

QPU / Simulator

Quantum Seed Generator

→

Classical Layer



→

Bridge

Best LABS Sequence

Quantum Execution

Classical Pre-Processing

Hybrid Interface

GPU Acceleration Strategy

GPU-Quantum Workflow Optimization

Circuit Batching

Batch thousands of circuits per launch to saturate GPU compute.
Amortizes launch overhead and maximizes circuits/sec.

GPU Utilization

GPU-optimized state vector and tensor-network kernels.

Performance scales with FLOPs and memory bandwidth (compute-bound).

Quantum Scheduling

Asynchronous CPU–GPU pipeline hides classical latency.
Keeps GPU fully occupied during hybrid execution.

Classical (MTS)

Parallel Neighbor Evaluation: Batches candidate moves (single-bit flips) on the GPU to compute ΔE in parallel.

Incremental Updates: Uses incremental energy updates rather than recomputing full LABS energy from scratch.

CuPy Integration: Rapid development using CuPy for massively parallel computation on sequence vectors.

Hardware Targets

L4: Rapid iteration and batched development.

H100: Heavy compute bursts for raw throughput and final scaling tests.

25x	10^3 -10^4	78%
Speedup vs CPU	Circuits / Sec	GPU Occupancy

Results - performance

Number of Gates vs Problem Size (N)

G=(N-2)^3

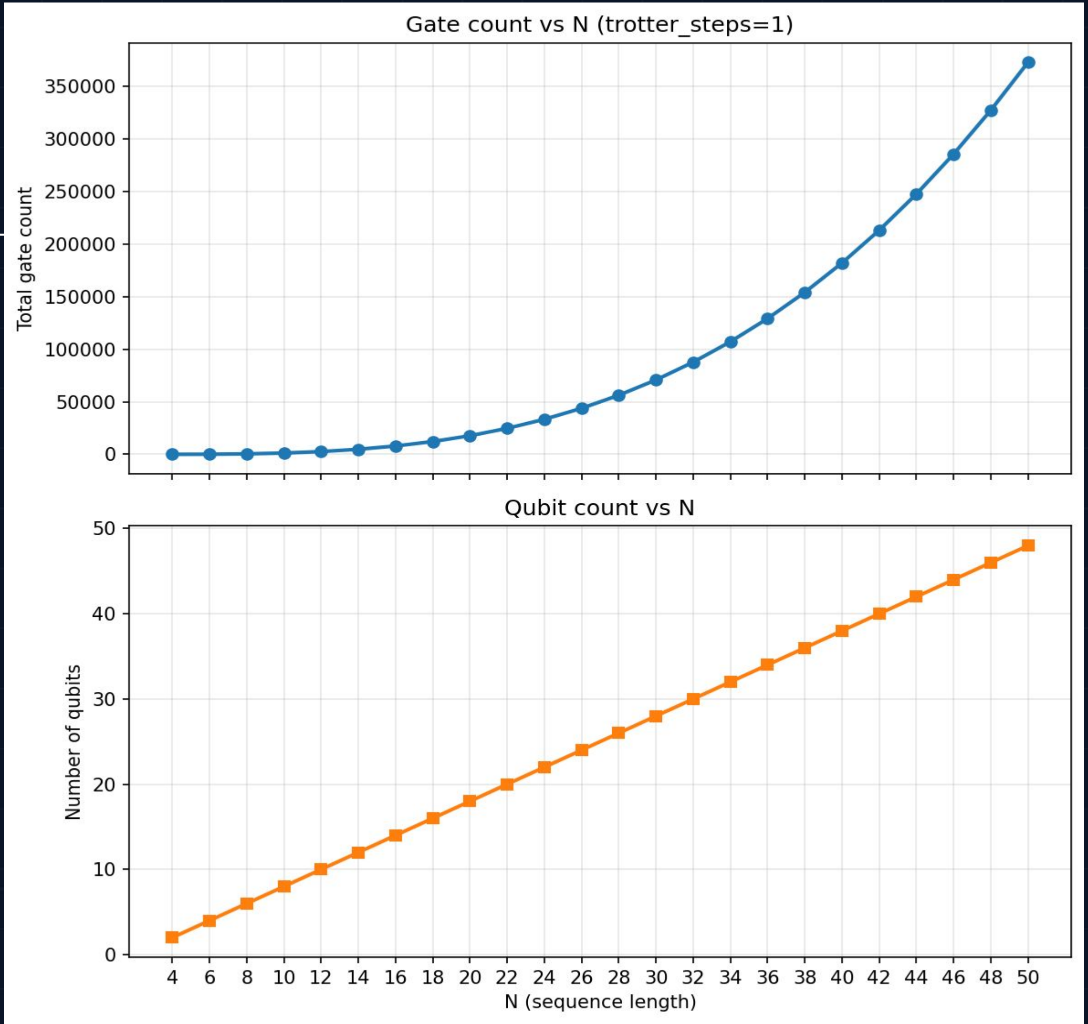
Scaling Law

vs. Classical Baseline

Q=N-2

Linear Qubit Footprint

Linear scaling observed



Results - Accuracy Comparison

Minimizing Normalized Energy Distance

Normalized Energy Distance $$\frac{E_{\text{best}} - E_{\text{optimal}}}{E_{\text{optimal}}}$$

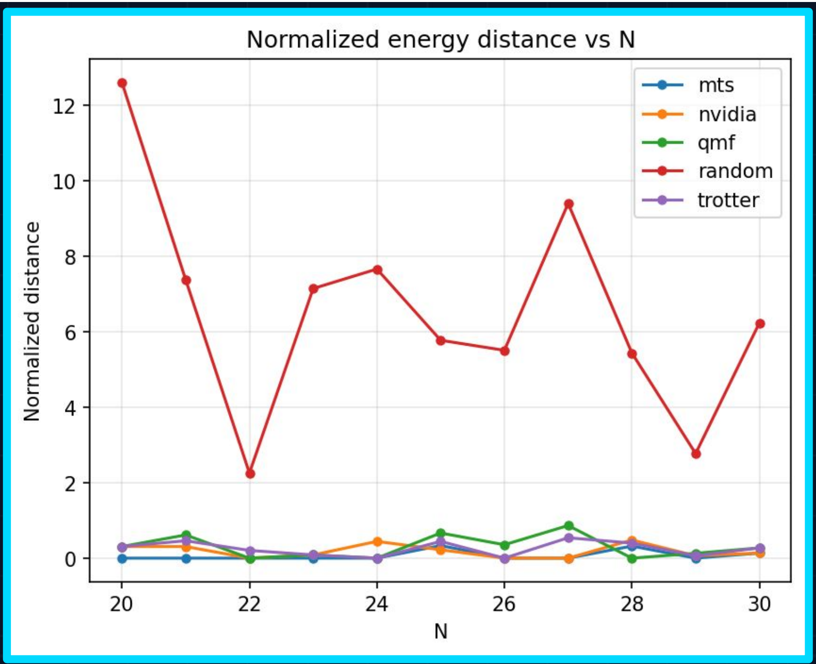
Definition

Equals Nvidia

State of the Art Baseline from Phase 1

Trotter > QMF

Confirm most promising approach to scale up



Strong Scaling

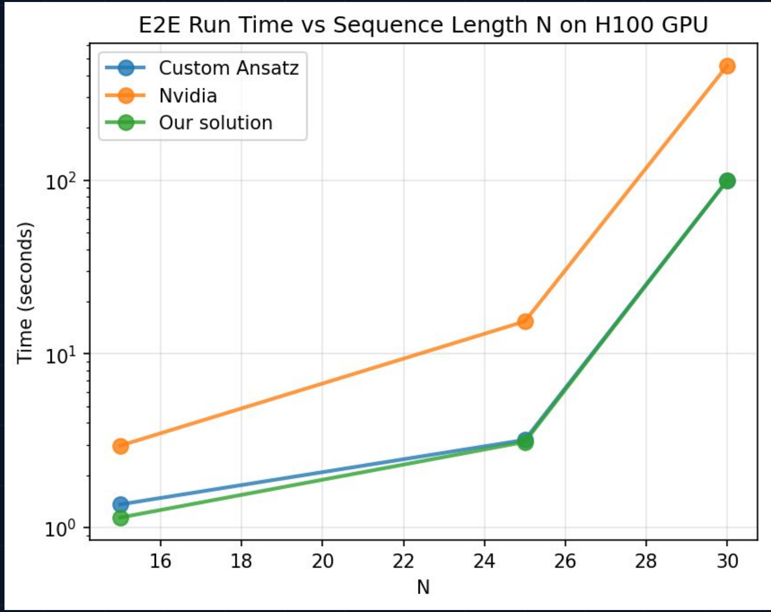
Results - Speed Comparison

Minimizing End To End Time (excl. compilation)

Larger gains with N

6.1x speed-up for N=30

Incremental improvement



Exponential speed-up

Results - GPU Synergy

Time-to-Solution and Scaling Analysis using Brev

13–15x

Speedup Factor

vs. Classical Baseline

~45%-65%

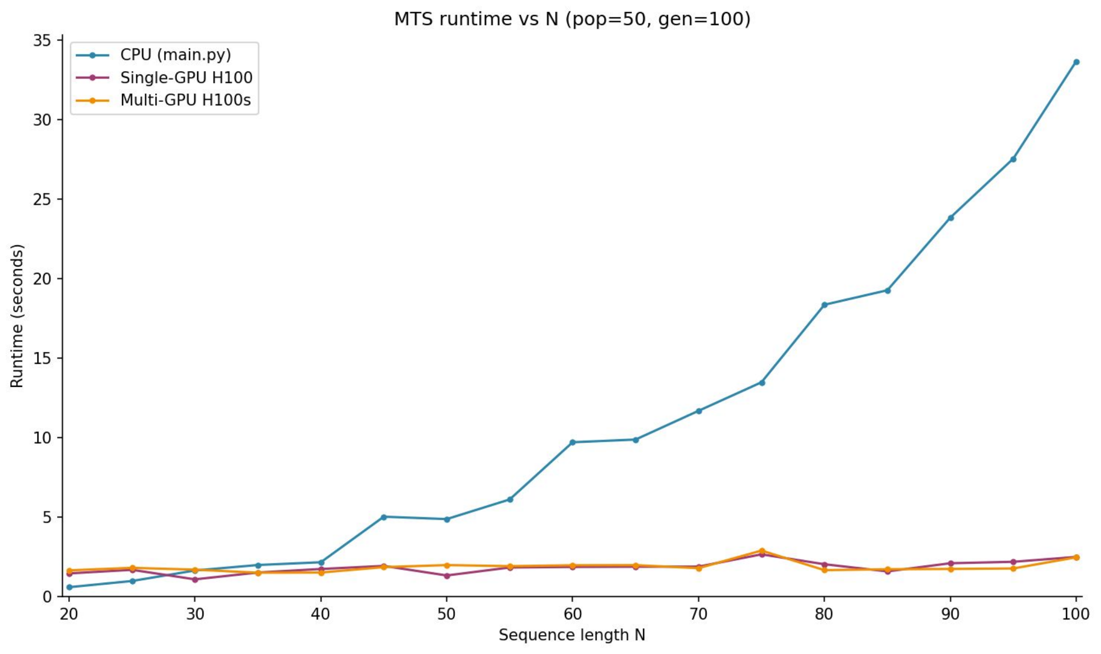
Scaling Efficiency

Linear scaling observed

2.47s

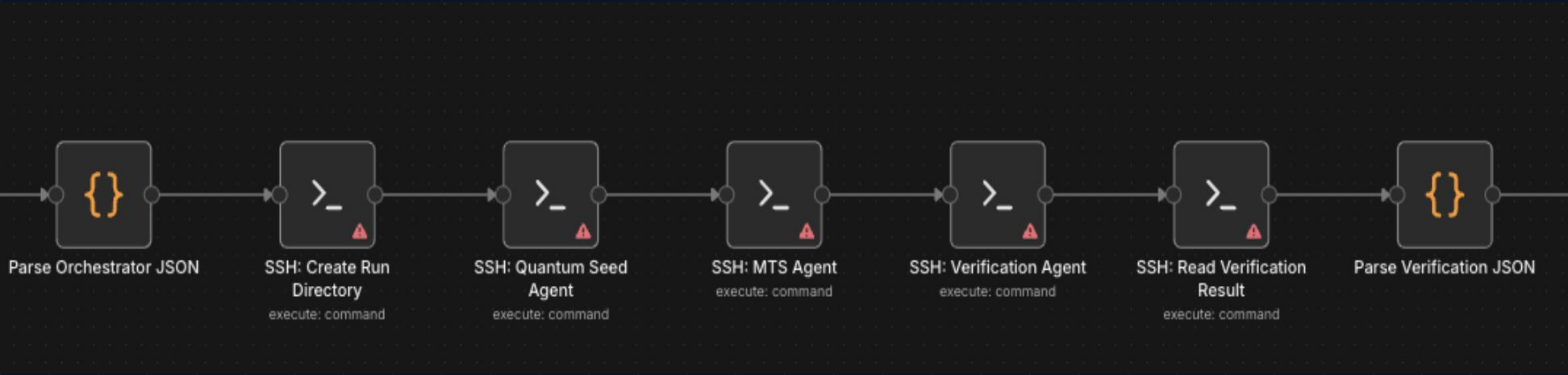
Time-to-Solution-Multi gpus

For max problem size



System Architecture

Quantum-Classical Hybrid Pipeline - AI Agents



Verification & Correctness

Ensuring Result Integrity

Unit Testing Strategy

Automated testing using **pytest**. AI hallucination guardrails ensure API calls and logic remain consistent with CUDA-Q documentation.

Framework: pytest

Physics Symmetry Tests

Tests for complementary and reversal symmetry across sampled sequences

Test: evals/physics_tests.py

Ground Truth Validation

Core evaluation suite matches results against peer-reviewed "golden answers" for N < 66(Packebusch et al., 2016).

Source: evals/answers.csv

Automated Benchmarking

Scripted scheduling of multiple trials and sequence lengths with pipelined plotting

Util: benchmarks/run_benchmark.py

SYSTEM VERIFIED

Key Takeaways

What We Learned

01

Every pattern counts

Simple Optimization Ideas can lead to massive performance gains

02

No need to reinvent the wheel

Extending an established solution allows for robust scaling laws on GPUs

03

Potential to merge with other ideas

Combine initial theoretical exploration of PCE and DMRG with bedrock solution

Literature Review

Foundations of the Hybrid Approach

Shaydulin et al. (2024) - Science Advances

Relevance: Primary justification for "Fixed Parameter QAOA + QMF" showing empirical scaling advantage for LABS.

Chandarana et al. (2022) - Phys. Rev. Research

Relevance: Introduces digitized counterdiabatic QAOA; suppresses transitions to excited states for shallower circuits.

NVIDIA CUDA-Q Documentation

Relevance: Technical foundation for DC-QAOA and QMF implementations in the CUDA-Q ecosystem.

Durr & Hoyer (1996) - arXiv:quant-ph/9607014

Relevance: Foundational paper for the Quantum Minimum Finding (QMF) routine used in our pipeline.

Packebusch & Mertens (2016) - arXiv:1512.0247

Relevance: Provides "golden answers" for sequence lengths up to N=66 for ground truth verification.