TECHNICAL PAPER

Scaling Quantum Circuit Optimization via Guided Exploration and Global Context Learning

Serena Ma, Chunwei Xia, Zheng Wang

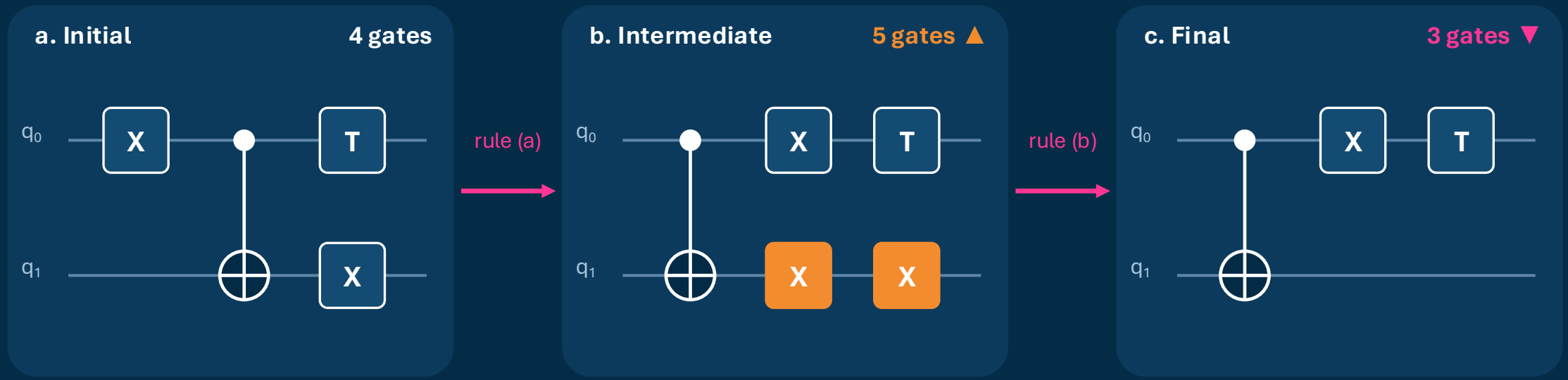
University of Leeds, United Kingdom



SC26 hpc unites | Chicago

Fewer Gates, Fewer Errors: Optimizing by Rewriting

Every gate adds error on noisy hardware and error-correction cost on fault-tolerant hardware. Gate-level rewriting swaps a subcircuit for a cheaper equivalent — but **the best paths often go uphill first.**



Rule (a): $X_c \cdot CX \rightarrow CX \cdot X_c \cdot X_t$ Rule (b): $X \cdot X \rightarrow I$

A greedy optimizer rejects step (a) because it adds a gate — and never reaches the 3-gate circuit.

The Hard Part: Choosing Among Thousands of Rewrites

Rule synthesizers (Quartz, Quanto) generate thousands of verified rewrite rules. The challenge shifts to **which rule, where, and in what order.**

2,198

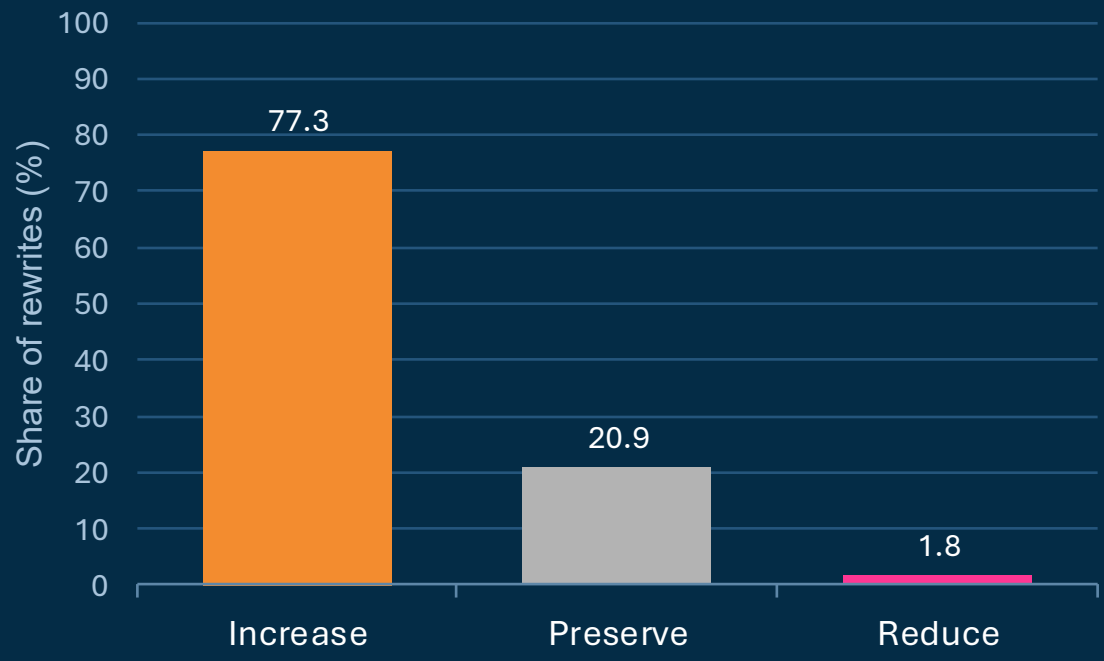
applicable rewrites per step, on average (tof_3, Nam gate set, 13,163 rules)

1.8%

of them reduce the gate count; 77% make the circuit bigger

54 steps

in a row with no gate-reducing rule anywhere in the circuit



Exhaustive BFS over 200 steps on tof_3 (45 gates). Uniform exploration wastes >98% of its compute.

Learning-Based Optimizers Hit Three Walls

State of the art (Quarl): GNN state encoder + RL policy over rewrite actions

1 Local view

The GNN sees only an l-hop neighbourhood. Optimizations that depend on distant gates, like commuting a gate across many operations to cancel its inverse, stay invisible.

GuiderQ
Multi-Head global context + asymmetric actor-critic

2 Unfocused exploration

Rewards are sparse and the landscape is flat. With thousands of candidates per step, standard RL spends its budget on moves that cannot help.

GuiderQ
A learned Guider that prioritizes promising rewrites

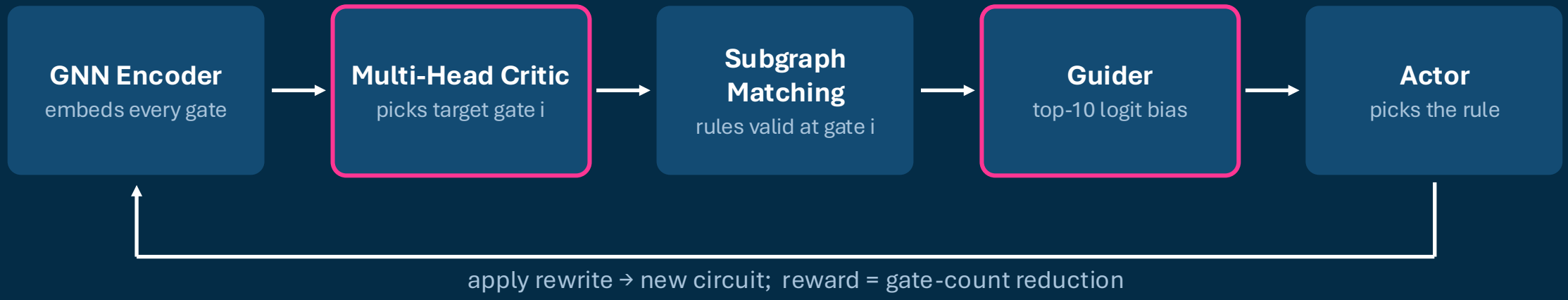
3 Static partitioning

Large circuits are cut into fixed segments optimized independently. Cancellations across cuts are lost, and cuts multiply as circuits grow.

GuiderQ
Recursive partition-and-reassembly that re-optimizes boundaries

GuiderQ at a Glance

Recursive partition-and-reassembly wraps the loop for large circuits



Offline, one-off

Pre-train PPO agent on 6 small circuits; log successful rewrites to train the Guider

Online, per circuit

QASM in → QASM out on top of Qiskit; Quartz rewrite rules; output verified equivalent

Idea 1: Multi-Head Global Context

- GNN embedding h_i describes only the neighbourhood of gate i
- K = 4 attention heads weigh **all** gates, each with a structural bias b_k (gate type, degree) and temperature τ_k
- Concatenated heads give a compact global summary of the circuit

$$\alpha_{i,k} = \mathrm{softmax}_i[\ (\mathrm{MLP}_k(h_i) + b_k(i)) / \tau_k\ ]$$

$$g_k = \Sigma_i\ \alpha_{i,k}\ h_i$$

$$g_{\mathrm{global}} = \mathrm{Norm}[\ g_1 \Vert g_2 \Vert g_3 \Vert g_4\ ]$$

Heads learned on barenco_tof_5, unsupervised

Head 0

Single-qubit rotations (H, T†)

Head 1

CX gates acting as connectivity hubs

Head 2

A few critical-path nodes (sparse)

Head 3

Diffuse view of global structure

0.28 mean pairwise cosine similarity between heads: complementary views

Global Context to Judge, Local Detail to Act

Critic

input: $[\, h_i \,\|\, g_{global} \,]$

Which gate to target? How valuable is this state in the long run?

→ **Global view of the circuit**

PPO advantage

Actor

input: h_i (local only)

Which rule to apply? Needs exact local structure, e.g. do two gates commute?

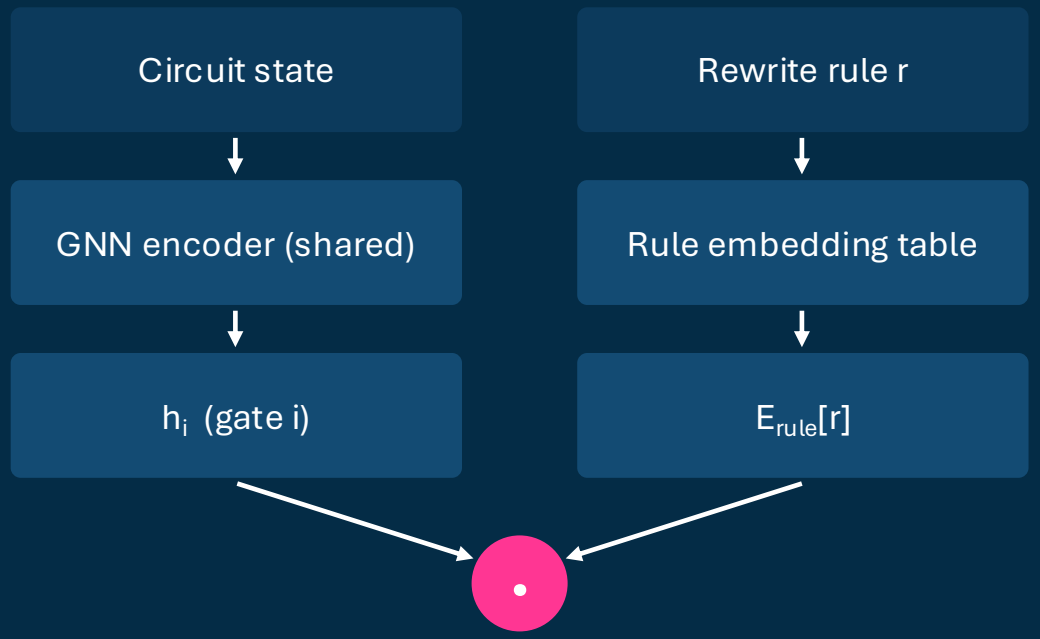
→ **Precise local signal**

Why not share one representation?

In a symmetric design, global features drown out small local changes; our pilot agents looped, applying and reverting the same commutation. Here the Actor learns global value only through the Critic's advantage signal.

Idea 2: A Learned Guider Ranks Rewrites

Dual-tower scorer



$$\text{Score}(i, r) = h_i^T E_{rule}[r]$$

Rule vectors are precomputed, so scoring all candidates is one vector op.

Training: free labels from RL logs

- Positive: the rule that reduced gates; negatives: other rules applicable at that gate
- Margin ranking loss: rank good above bad, no single "correct" rule

Example: gate g_{11} in barenco_tof_3, 12 candidates

r3454 Δ−2	r3458 Δ−2	**r3462 Δ−2 ✓**

+ 9 preserving rules (Δ0). Only r3462 led to further gains, so only it is positive.

Inference: soft bias, not pruning

Top-10 rules get a logit bias in the PPO policy; every valid rule stays selectable to escape local optima

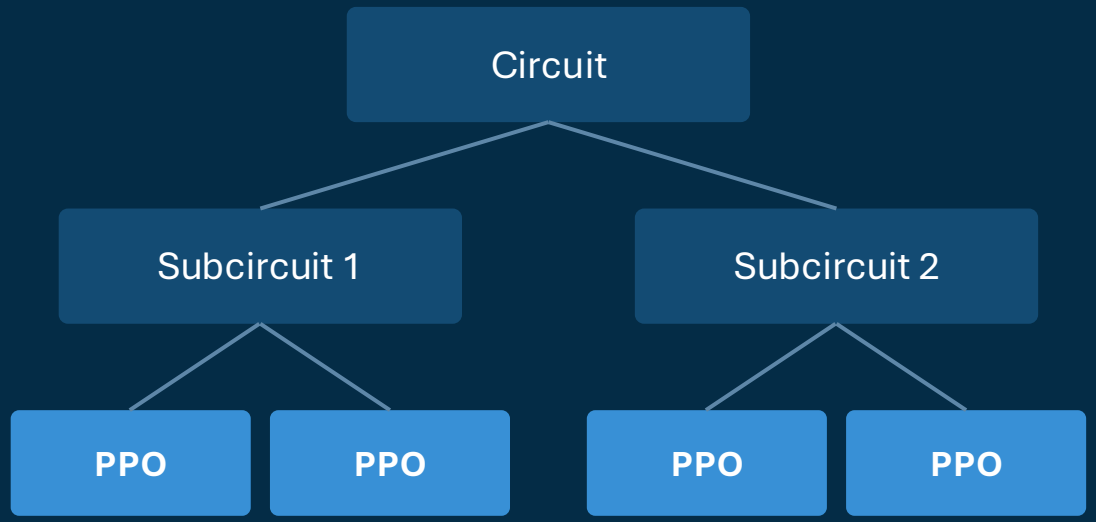
Cascading Search-Space Reduction per Step



Cheap: ≈25 ms per step (GNN 13 ms + rule enumeration 12 ms; heads <0.2 ms). The Guider runs asynchronously (~15 ms, overlapped).

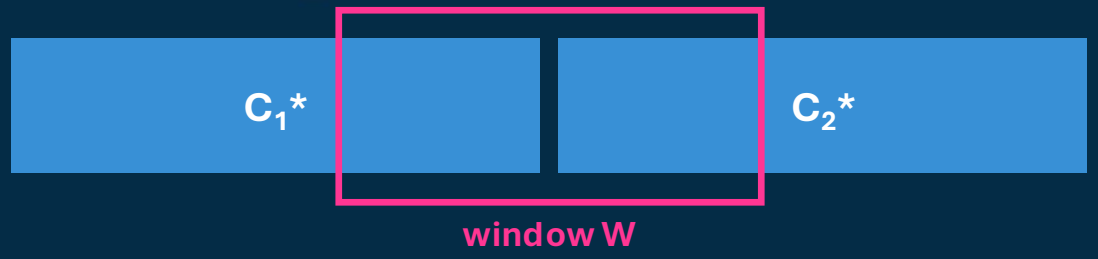
Idea 3: Recursive Partition-and-Reassembly

Bisect by depth until ≤ 2Ω layers (Ω = 40)



Leaves optimized in parallel across GPUs

Meld: re-optimize the boundary



W = last Ω layers of C_1* ‖ first Ω layers of C_2*

Improved? splice W' in and expand outward. Otherwise, concatenate.

One round = compact → segopt → meld

Repeat until gate count stops falling; ≤ M + B(M−1) PPO calls

3.1% → 35.2%

Qiskit -O3 alone vs. inside this scheme, 4 medium circuits, 1–6 s each

Evaluation Setup

62

benchmark circuits: 24 Quartz + 38 MQTBench

3

gate sets: IBM, Nam, TDG (Clifford+T)

27,883

gates in the largest circuit

4× A100

6 h budget per circuit (3 h above 20K gates)

Baselines

Qiskit -O3: greedy rule-based compiler

Quartz: superoptimizer with synthesized rules

Quarl: state-of-the-art RL optimizer, ported to the latest Qiskit

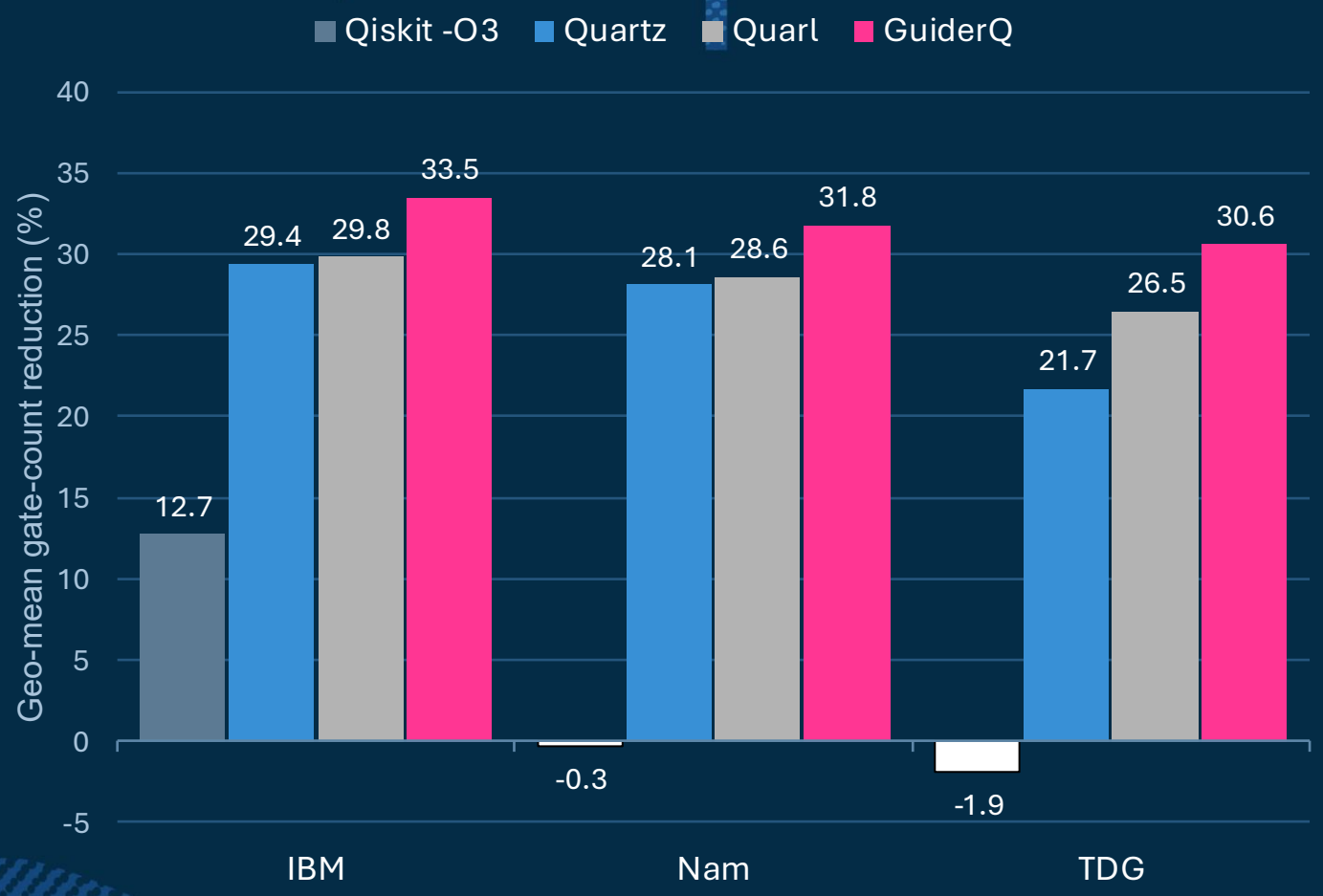
Protocol

Best of 5 runs; run-to-run variance is small

6 Quartz circuits used for pre-training, so held-out results are reported separately

Metrics: gate count, depth, time-to-95%

Quartz Suite: Best on All Three Gate Sets



+11–16%

relative improvement over Quarl (IBM / Nam / TDG)

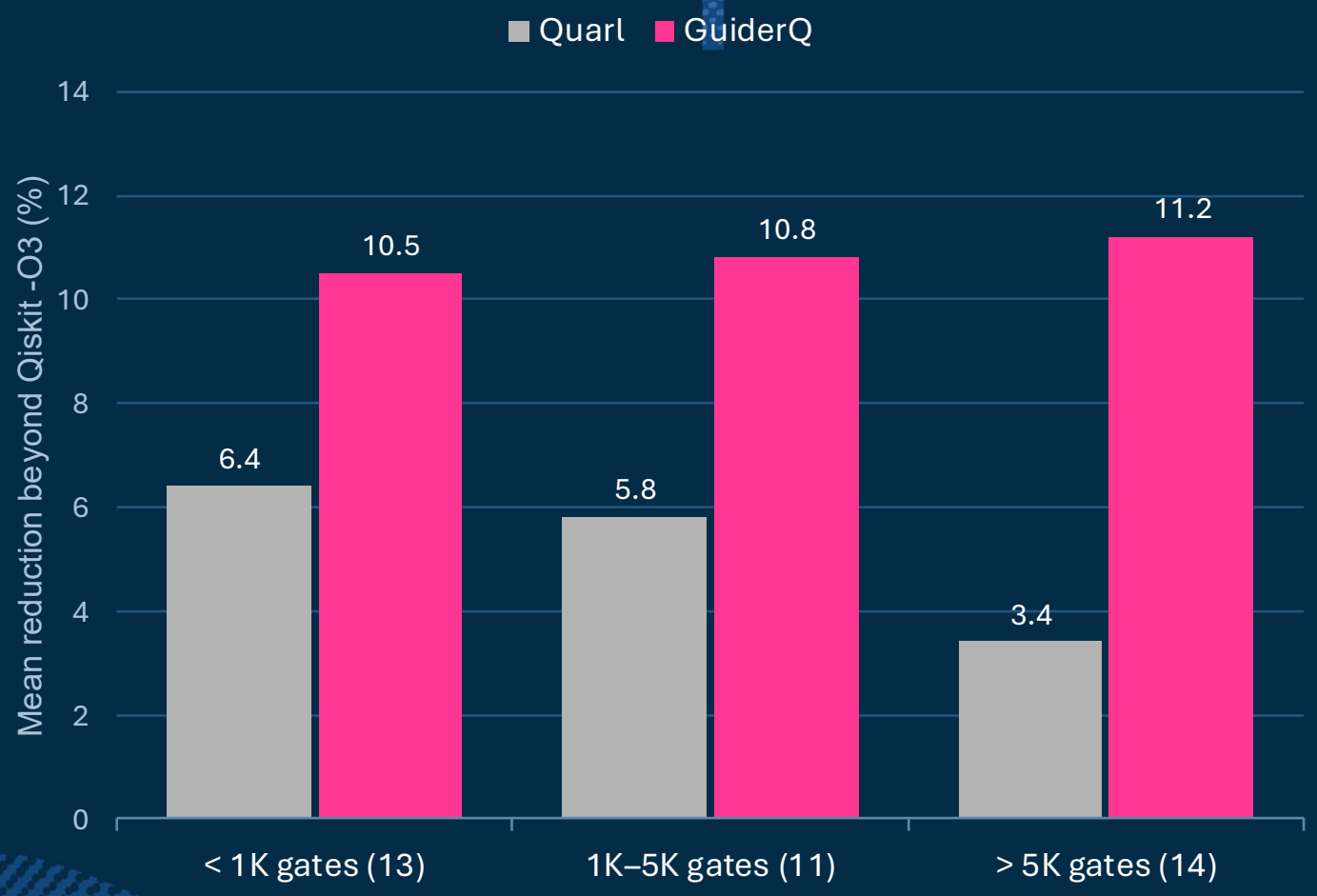
31.4 · 29.9 · 28.3%

on the 18 held-out circuits, within ~2 points of the full mean

22–28%

circuit-depth reduction

Scaling to 27K-Gate Circuits (MQTBench)



5.1% → 10.8%

geo-mean gate reduction, Quarl vs. GuiderQ (38 circuits)

10.2%

geo-mean depth reduction

28.0% vs 4.4%

on tsp_16: structured circuits gain most from global context

Takeaways

1

See globally, act locally

Multi-head context + asymmetric actor-critic finds non-local, uphill rewrite sequences

2

Explore where it pays

A learned Guider focuses search on promising rewrites: up to 3× faster

3

Scale without losing boundaries

Recursive partition-and-reassembly handles 27K+ gates

Limitations & next steps

Optimizes logical gate count; hardware connectivity and noise are not yet modelled

Next: hardware-aware objectives, adaptive partitioning

Code & data

github.com/quantumlearner/sc26ae-515

GUIDERQ · UNIVERSITY OF LEEDS

Thank You. Questions?

Code & data: github.com/quantumlearner/sc26ae-515

Serena Ma (scym@leeds.ac.uk) · Zheng Wang (z.wang5@leeds.ac.uk)



SC26 hpc unites | Chicago