Spirit Sprinters

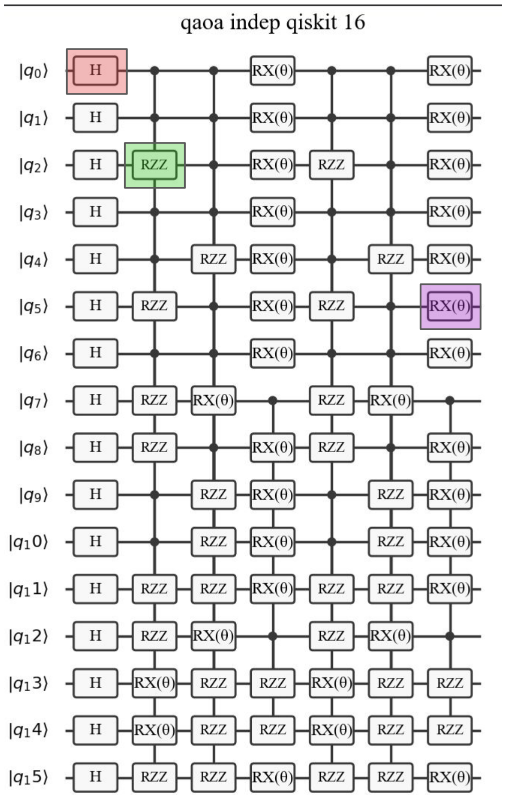
iQuack 2026 Quantum Rings

Quantum Rings

Spirit Sprinters

Quantum Rings iQuHack Contest

What information can we get from a circuit?



- Number of Qubits
- Number of Gates
- Number of Particular Gates:
    - Hadamard
    - RZZ
    - Rx

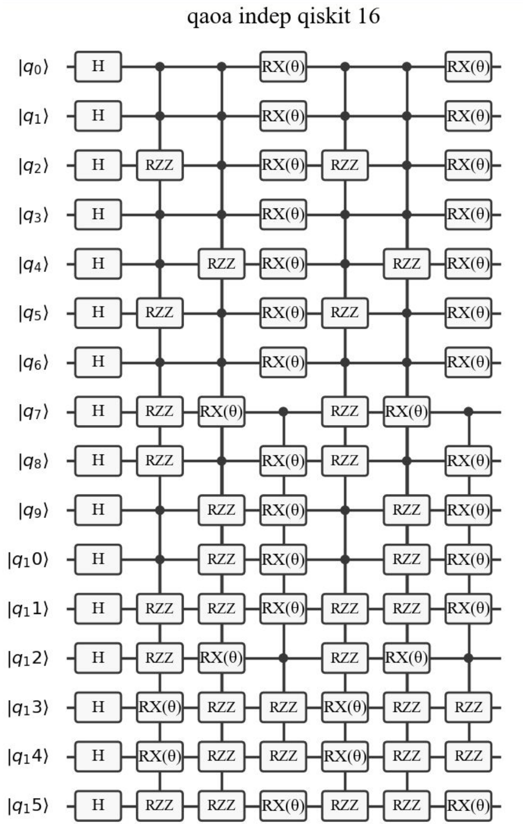
- Degree
- Cut Crossings
- Span

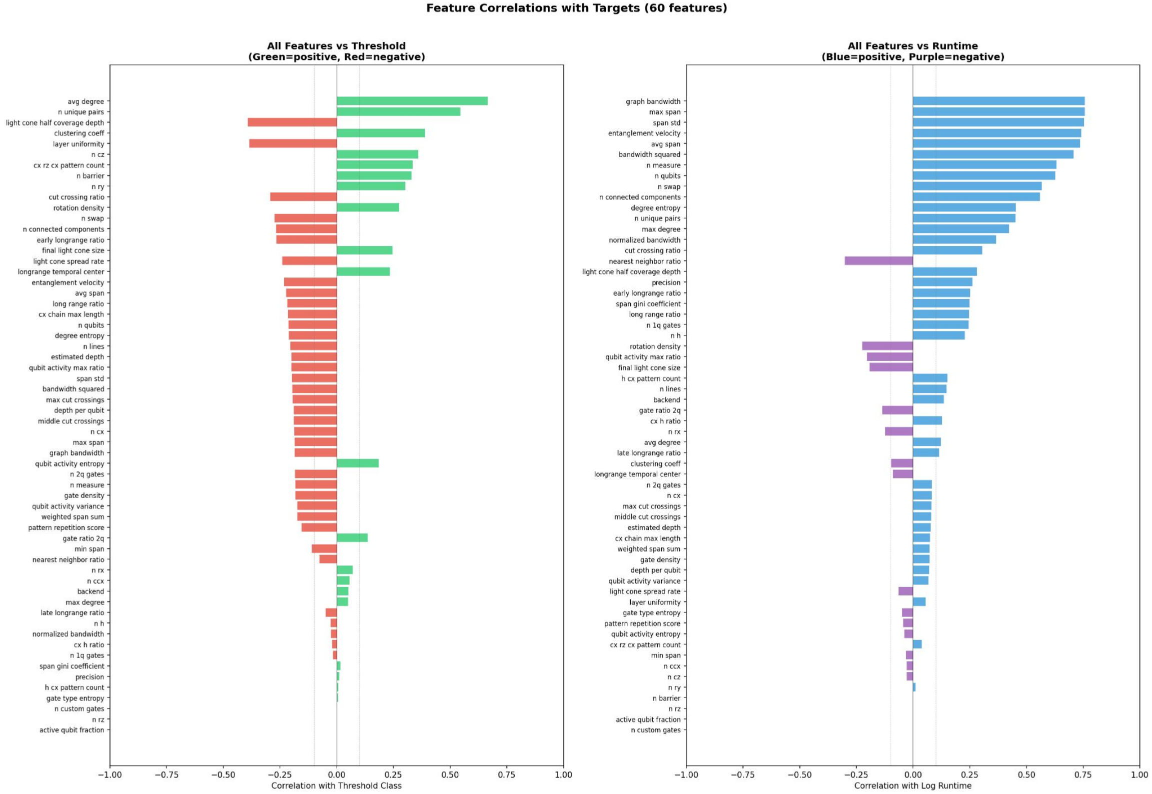
Can also look at the average, density and maximum of these features

- Specific Patterns of Qubits
- Light Cone

We are able to use these parameters (and more) to create a robust feature set!

What information can we get from a circuit?





What information can we get from a circuit?

Engineered Features

All features below are derived from the raw dataset (circuits, results, QASM files). The original data provides: circuit `file`, `backend`, `precision`, `family`, `n_qubits`, `selected_threshold`, `run_wall_s`, and threshold sweep data. Everything in these tables is computed for modeling.

1. Run / task encoding

Encodings of fields that come from the result record (backend, precision, threshold, family).

Name	Description
`backend_idx`	0 = CPU, 1 = GPU.
`precision_idx`	0 = single, 1 = double.
`log2_threshold`	$\log_2$(selected_threshold); used as input for duration prediction.
`family_onehot`	One-hot over circuit family (20 categories, e.g. QFT, GHZ, QAOA).

2. QASM: Basic circuit statistics

Counts and sizes extracted from the QASM file (not stored as such in the original JSON).

Name	Description
`n_lines`	Number of non-empty, non-comment lines in the QASM file.
`n_qubits`	Number of qubits declared in the circuit (`qreg q[N]`).
`n_cx`	Count of CNOT (CX) gates.
`n_cz`	Count of controlled-Z (CZ) gates.
`n_swap`	Count of SWAP gates.
`n_ccx`	Count of Toffoli (CCX) gates.
`n_2q_gates`	Total 2-qubit gates (cx + cz + swap + ccx).
`n_1q_gates`	Total 1-qubit gates (h, x, y, z, s, t, rx, ry, rz, u1, u2, u3).
`n_measure`	Count of measurement operations.
`n_barrier`	Count of barrier instructions.
`n_custom_gates`	Count of custom gate definitions (`gate ...`).
`n_h`	Count of Hadamard gates.
`n_rx`, `n_ry`, `n_rz`	Counts of parameterized rotation gates.

3. QASM: Qubit span

Measures the "reach" of 2-qubit gates across the qubit register (distance between qubit indices).

Name	Description
`n_unique_pairs`	Number of distinct qubit pairs that interact via 2Q gates.
`avg_span`	Mean distance between qubits in 2Q gates.
`max_span`	Maximum span over all 2Q gates.
`min_span`	Minimum span over all 2Q gates.
`span_std`	Standard deviation of spans.

4. QASM: Gate density

Name	Description
`gate_density`	2Q gates per qubit (`n_2q_gates / n_qubits`).
`gate_ratio_2q`	Fraction of all gates that are 2-qubit gates.

5. QASM: Interaction graph

Built from the qubit interaction graph (nodes = qubits, edges = 2Q gate pairs).

Name	Description
`max_degree`	Maximum number of distinct qubits any single qubit interacts with.
`avg_degree`	Average degree over all qubits.
`degree_entropy`	Normalized entropy of the degree distribution.
`n_connected_components`	Number of disconnected subgraphs.
`clustering_coeff`	Graph clustering coefficient.
`max_component_size`	Size of the largest connected component.
`component_entropy`	Normalized entropy of component size distribution.

6. QASM: Depth

Name	Description
`estimated_depth`	Estimated circuit depth (max gate layers on any qubit).
`depth_per_qubit`	Depth divided by qubit count.

QASM: Cut (entanglement pressure)

Counts of 2Q gates crossing bipartitions of the qubit register; relevant for MPS bond dimension.

Name	Description
`middle_cut_crossings`	Number of 2Q gates crossing the middle cut (qubit N/2).
`cut_crossing_ratio`	Fraction of 2Q gates that cross the middle cut.
`max_cut_crossings`	Maximum crossings at any single cut position.

QASM: Graph bandwidth

Name	Description
`graph_bandwidth`	Maximum span in the interaction graph (same as `max_span`).
`normalized_bandwidth`	Bandwidth / n_qubits.
`bandwidth_squared`	Squared bandwidth.

9. QASM: Temporal

When in the circuit long-range gates appear (early vs late).

Name	Description
`early_longrange_ratio`	Fraction of gates in the first third that are long-range (span > N/4).
`late_longrange_ratio`	Fraction of gates in the last third that are long-range.
`longrange_temporal_center`	Normalized position (0–1) of long-range gates in the circuit.
`entanglement_velocity`	Mean increase in cumulative span per 2Q gate.

10. QASM: Qubit activity

Distribution of gate involvement across qubits.

Name	Description
`qubit_activity_entropy`	Normalized entropy of qubit usage (1 = uniform).
`qubit_activity_variance`	Variance of gate counts per qubit.
`qubit_activity_max_ratio`	Fraction of all gate references on the most active qubit.
`active_qubit_fraction`	Fraction of qubits that participate in at least one gate.

11. QASM: Gate patterns

Sequence-level patterns (n-grams) indicative of algorithm type.

Name	Description
`cx_chain_max_length`	Longest consecutive run of CX gates.
`h_cx_pattern_count`	Count of H→CX patterns (Bell/GHZ-like).
`cx_rz_cx_pattern_count`	Count of CX→rotation→CX patterns (e.g. variational)
`rotation_density`	Fraction of gates that are parameterized rotations.
`gate_type_entropy`	Entropy of gate type distribution.
`cx_h_ratio`	Ratio of CX count to H count.

12. QASM: Light cone

How quickly information spreads from a central qubit through 2Q gates.

Name	Description
`light_cone_spread_rate`	Rate of growth of "reached" qubits per 2Q
`light_cone_half_coverage_depth`	Normalized depth to reach half of the qubi
`final_light_cone_size`	Fraction of qubits reached by the end of th

13. QASM: Entanglement structure

Name	Description
`nearest_neighbor_ratio`	Fraction of 2Q gates with span = 1.
`long_range_ratio`	Fraction of 2Q gates with span ≥ N/3.
`span_gini_coefficient`	Inequality in span distribution (0 = even, 1 = concentrated)
`weighted_span_sum`	Sum of squared spans, normalized by n_qubits².

14. QASM: Circuit regularity

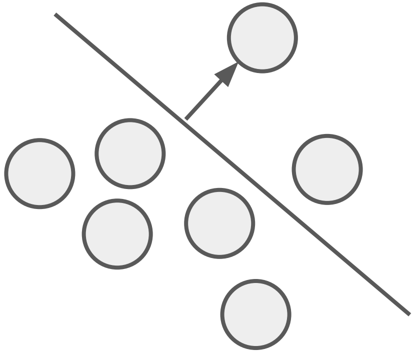
Name	Description
`pattern_repetition_score`	Max repeat count of 4-gate patterns over the circuit.
`barrier_regularity`	Regularity of barrier spacing (1 – normalized std of gaps
`layer_uniformity`	Placeholder for layer-structure regularity (fixed 0.5).

15. QASM: Treewidth

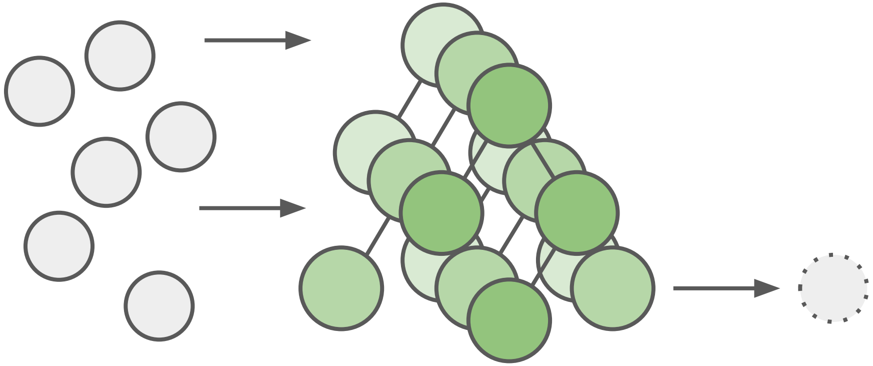
Name	Description
`treewidth_min_degree`	Treewidth estimate from min-degree elimination on the interaction graph.

Feature Vector Models: Four Approaches

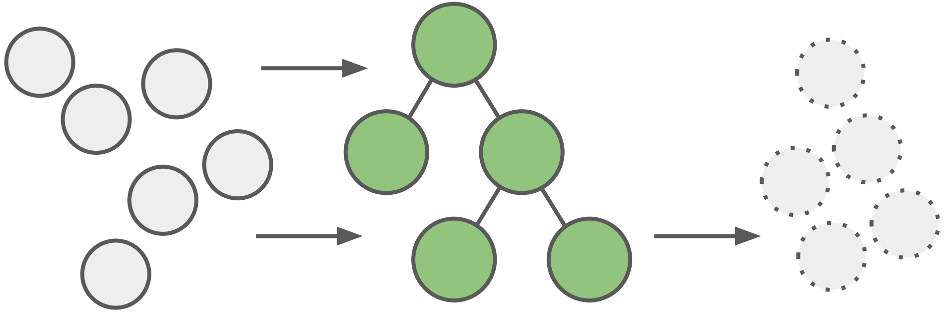
Regression



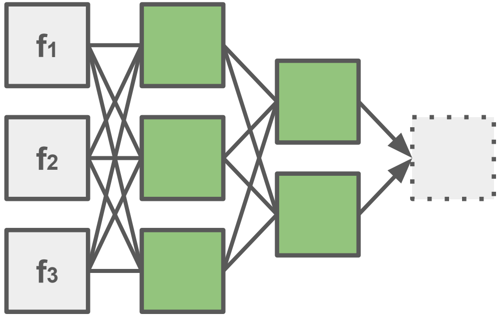
Random Forest



Gradient Boosting



Multi-layer Perceptron (MLP)



Feature Vector Models: Four Approaches

Regression

Model linear relationships in data

Random Forests

Collection of decision trees trained on random data subsets

Ensemble votes for model prediction

Robust to small data size

Gradient Boosting

A sequence of decision trees, where each tree corrects preceding decision errors

Even more robust to small data size

MLP

Multiple layers of linear combinations of variables followed by nonlinear “activations”

Models arbitrary nonlinear relationships in data

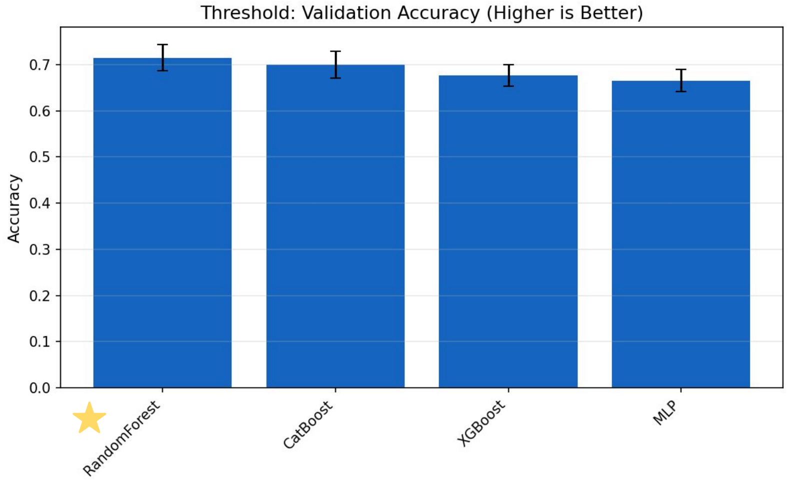
Task 1: Threshold Prediction

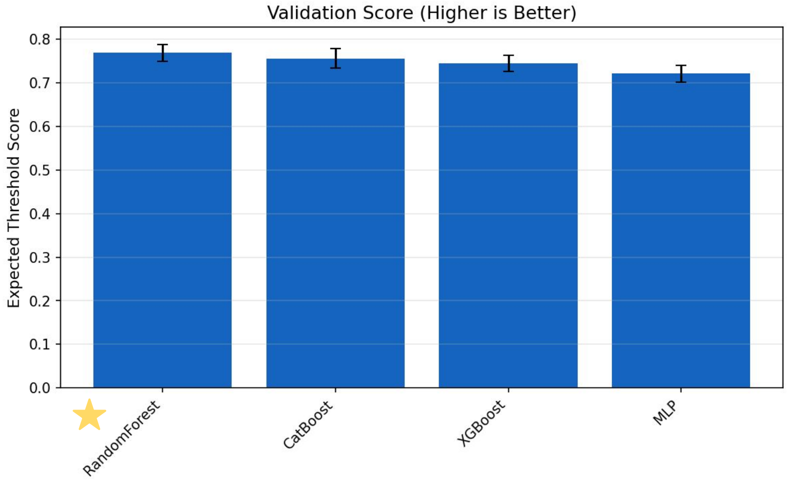
Accuracy: Proportion of the time model correctly predicted the minimum threshold producing 75% fidelity

Score: 0 for underestimation or 2^{-*n*} for *n* class error

Method:

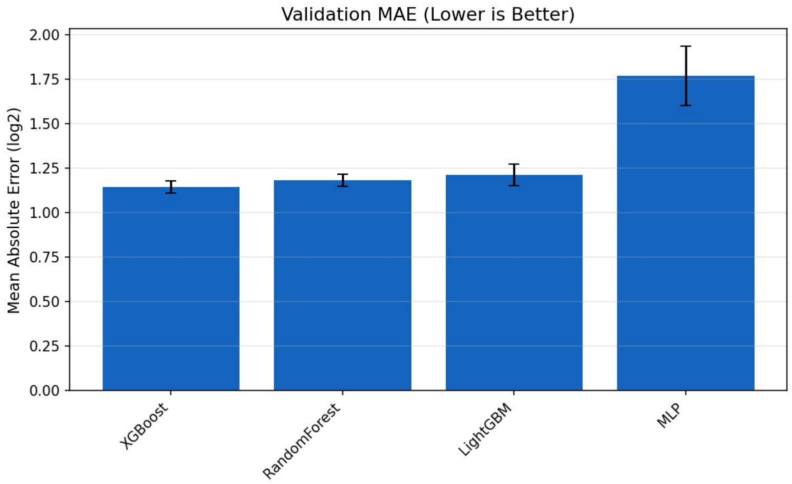
1. Construct a probability distribution *P*
2. Compute score expectation for each possible answer with *P* as a prior
3. Choose maximum expectation class

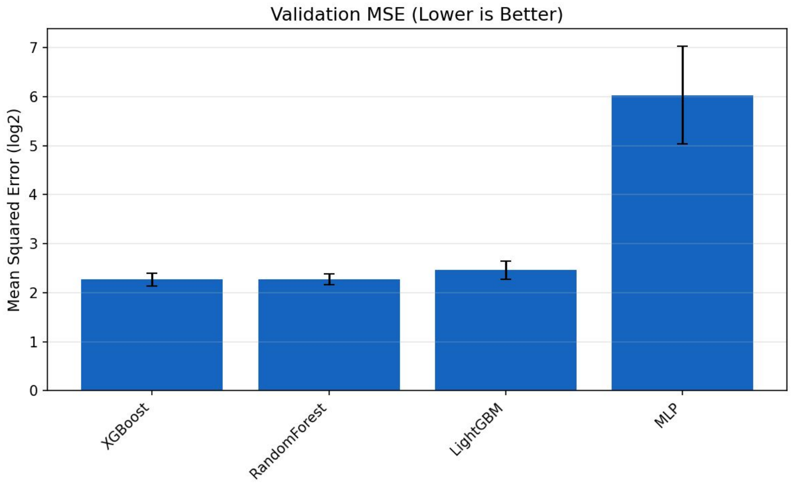




Task 2: Duration Prediction

- For tabular models, our loss function is mean absolute error (MAE) in log space.
- XGBoost provides the best overall prediction, with a log-space MAE of < 1.2.
- This corresponds to real-time predictions on average being 0.44x or 2.3x the true value.
- Not very good!



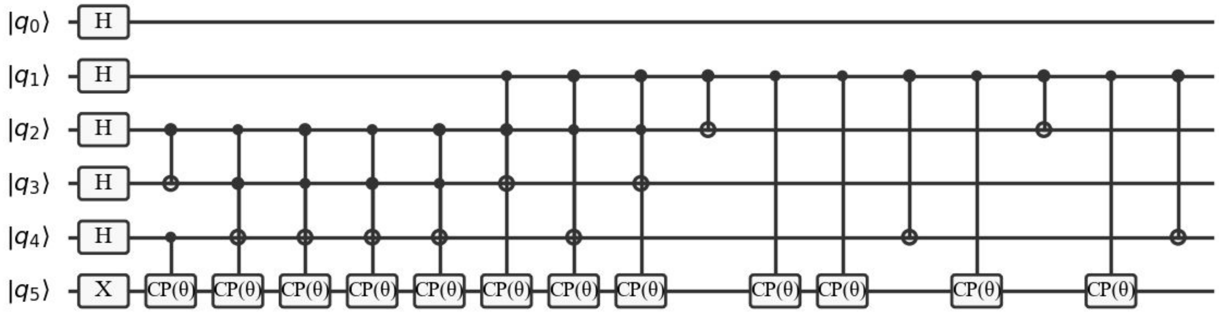


“The bitter lesson is that general methods that leverage computation are ultimately the most effective, and by a large margin.”

— Rich Sutton, *The Bitter Lesson* (2019)

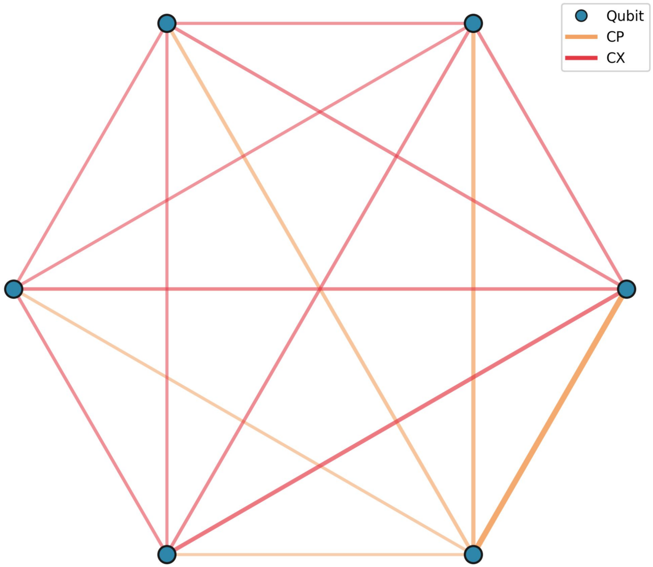
Leveraging Computation

grover-v-chain indep qiskit 7 (first 30 of 310 gates)

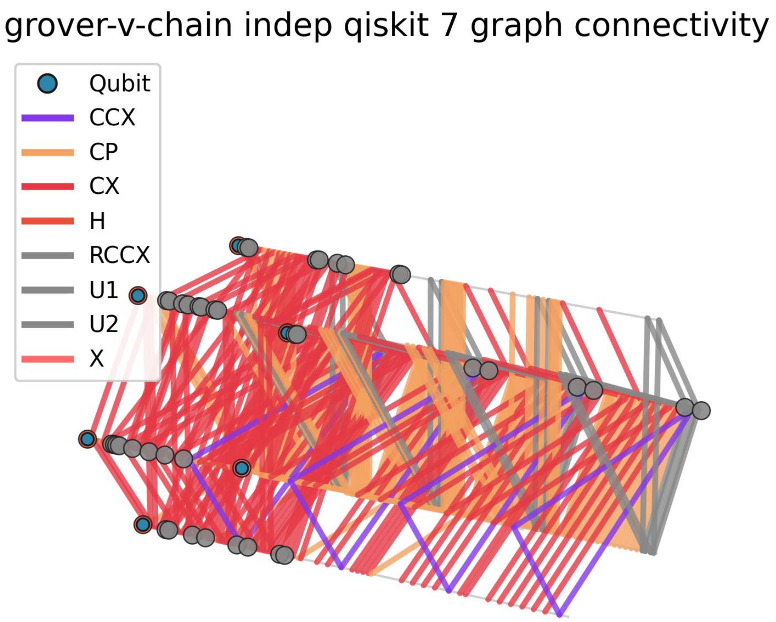


Leveraging Computation

grover-v-chain indep qiskit 7 graph connectivity

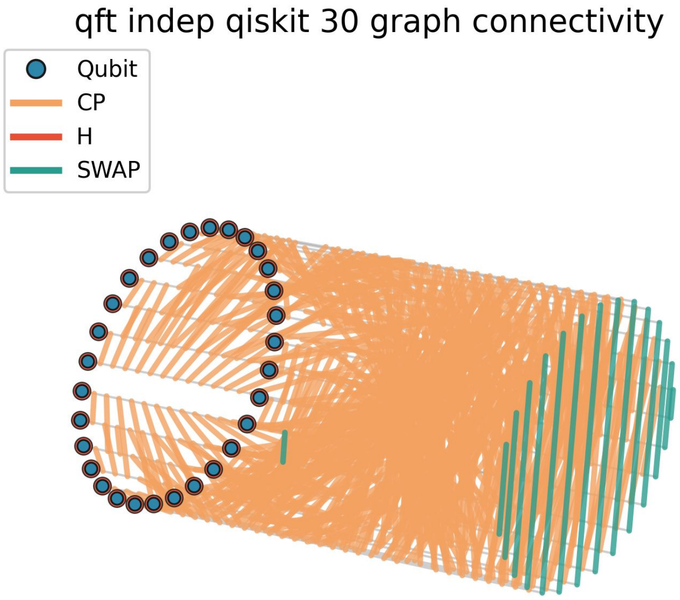


Leveraging Computation



Leveraging Computation

- Qubits are nodes
- Gates are edges
- Some features are conditioned graph-wide (precision, CPU/GPU, etc)
- Use a Graph Neural Network (GNN)
    - Learned message passing between adjacent nodes
- Augment data to fight scarcity
    - Permute qubit indices
    - Small perturbations in graphs



Graph Neural Network (GNN): Four Approaches

Message Passing Neural Network

Each node sums messages from neighbors

Messages modulated by learned gate embeddings

Graph Transformer NN

Global self-attention mechanism

Nodes can look across the graph to learn embeddings

Heterogeneous Graph NN

Edges are grouped into semantic relation types (rotation, pauli, entangle, swap, control, temporal)

Separate processing per relation

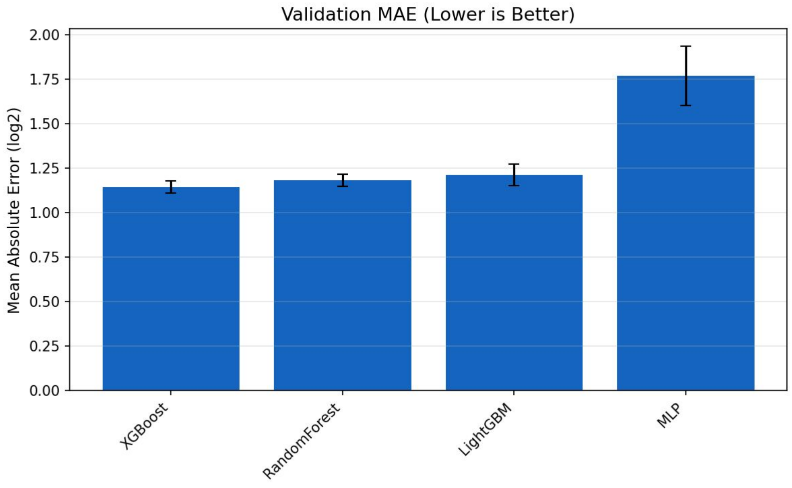
Hierarchical attention

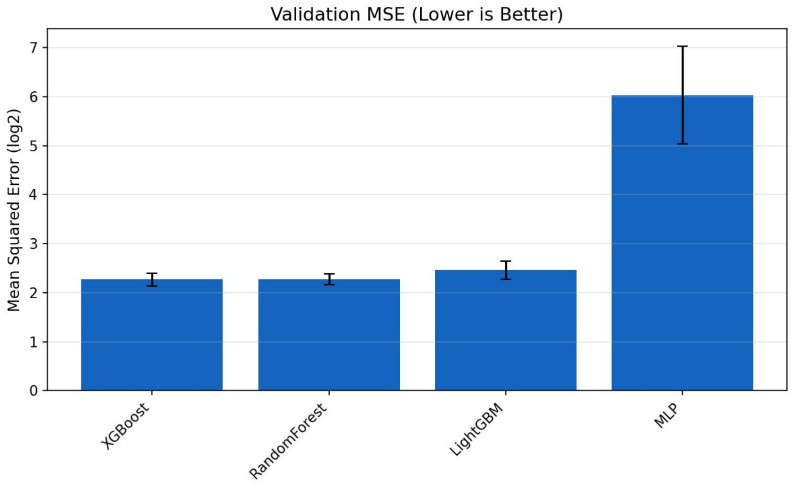
Temporal Graph NN

Temporal positioning on gates respects order

Maintains GRU qubit memory across time

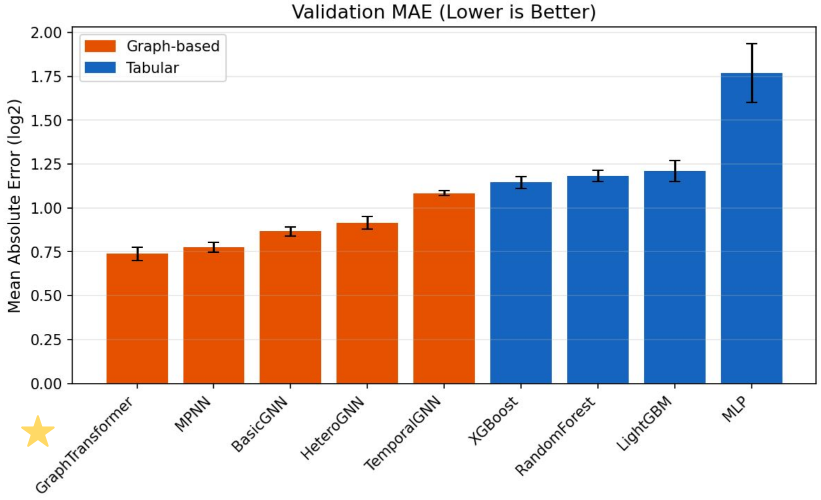
Task 2 Revisited

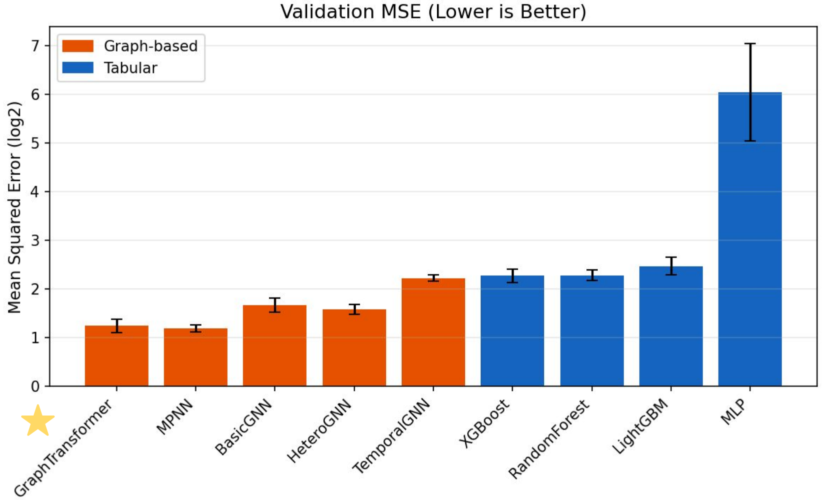




Task 2 Revisited

- GraphTransformer now provides the best overall prediction, with a log-space MAE of < 0.75.
- This corresponds to real-time predictions on average being 0.59x or 1.7x the true value.
- Better result





Questions?