

Algorithmic Intent Leakage in Transpiled Quantum Circuits: An Initial Reverse-Engineering Study

Beat Buesser
IBM Research
Zurich, Switzerland
beat.buesser@ibm.com

Anil Kurmus
IBM Research
Zurich, Switzerland
kur@zurich.ibm.com

Abstract—We study whether coarse algorithmic intent can be inferred from final backend-transpiled quantum circuits. The receiver observes only the optimized physical circuit and has no access to the source program, logical-qubit layout, routing history, or intermediate compiler representations. We construct a controlled dataset spanning seven intent classes and transpile circuits to Qiskit’s FakeNighthawk backend for 24 and 48 qubits at optimization levels 0–3. Each circuit is represented by ten normalized features capturing gate composition, temporal two-qubit activity, and spatial hardware-load structure. A Random Forest classifier achieves high accuracy and macro-F1 when trained and tested within the same optimization-level or circuit-size regime. Cross-setting transfer is weaker, except between optimization levels 2 and 3, which show strong mutual generalization. These results indicate that final transpiled circuits retain measurable intent-related structure, but that this leakage depends on transpiler optimization level and circuit-size regime.

Index Terms—Quantum circuit, reverse engineering, machine learning, classification algorithms, data privacy.

I. INTRODUCTION

Quantum circuits are often shared, benchmarked, or executed only after backend transpilation. *We ask whether the resulting physical circuit retains observable structure correlated with the algorithmic intent of the original circuit.*

We formulate this as a controlled intent-recovery problem over seven coarse intent classes, using only normalized structural features extracted from the final backend-transpiled circuit. The objective is not source reconstruction or logical-qubit recovery, but measurement of coarse intent information preserved in backend-visible structure. We evaluate this information under two distribution shifts: transpiler optimization level and circuit size.

A. Related Work

Prior work has shown that quantum compilation artifacts can reveal compiler optimization passes [1], backend or coupling-map information [2], and properties of quantum machine-learning models [3]. Decompile work has also explored recovering interpretable high-level quantum programs from low-level circuit representations [4]. We address the complementary question of whether final transpiled circuits leak coarse algorithmic intent, using only normalized structural features computable from the received physical circuit.

II. EXPERIMENT

We evaluate intent recovery on seven intent classes listed in Table I. Source circuits are transpiled to Qiskit’s FakeNighthawk backend for 24 and 48 qubits at optimization levels 0–3, targeting the native instruction set with single-qubit gates `rz`, `sx`, and `x`, and `cz` as the native two-qubit entangling gate [5].

The detector uses only the final transpiled circuit for a given backend, qubit count, and optimization level; source circuits, logical layouts, routing histories, and intermediate compiler states are excluded.

TABLE I
INTENT LABELS AND CORRESPONDING CIRCUIT GENERATORS

Intent label	Circuit generators
Entanglement generation	GHZ, Bell_Chain, Graph_Line, Graph_Ring, Graph_Star, W_State_Ladder, Cluster_Grid
Fourier-like	QFT_swaps, QFT_no_swaps, iQFT_swaps, iQFT_no_swaps, Approx_QFT, Phase_Estimation, QFT_Phase_Gradient
Phase/oracle-like	Parity, Parity_odd, Parity_even, Phase_Oracle, Phase_Ladder, Controlled_Phase_Ladder, Diagonal_Phase_Pattern
Amplitude amplification	Grover, Grover_Zero, Grover_Two_Iterations, Parity_Amplitude_Amplification, Fixed_Point_Amplitude_Amplification, Amplitude_Estimation_Style
Variational ansatz	HEA_linear, HEA_ring, HEA_brickwork, RealAmplitudes, TwoLocal_RX_RZ, QAOA_Line, QAOA_Ring
Hamiltonian simulation	Trotter_ZZ_XX, Trotter_ZZ_XX_Ring, Trotter_TFIM, Trotter_XXZ, Trotter_Heisenberg, Trotter_Longitudinal_Ising
Arithmetic/permutation	Ripple_Adder, Incrementer, Permutation_Swaps, Controlled_Swap_Network, Comparator, Majority_Logic, Toffoli

A. Feature Selection

Table II defines the ten normalized structural features extracted from each final backend-transpiled circuit. The features cover three observable structure classes: gate composition, temporal two-qubit activity, and spatial hardware load.

The representation uses fractions, normalized activity measures, and Gini coefficients to reduce dependence on absolute size while retaining interpretable gate-composition, temporal-distribution, and spatial-load signals.

B. Machine Learning

Intent recovery is evaluated as supervised multi-class classification using the ten features in Table II. Train/test splits are source-circuit disjoint, and feature scaling is fitted only

TABLE II
FEATURES COMPUTED FROM BACKEND-TRANPILED CIRCUIT.

ID	Feature Definition
1	Fraction of all gates acting on two qubits.
2	Fraction of gates belonging to phase or rotation gate families.
3	Entropy of the distribution over single-qubit gate types.
4	Fraction of circuit layers containing two-qubit operation.
5	Mean simultaneous two-qubit operations per layer.
6	Gini coefficient of two-qubit gate counts across layers.
7	Fraction of edges used by two-qubit operations.
8	Gini coefficient of two-qubit gate counts across edges.
9	Maximum edge load divided by total two-qubit gate count.
10	Gini coefficient of two-qubit counts across qubits.

on the training split. Random Forest hyperparameters are selected by 5-fold cross-validated grid search on the training split. The selected model is retrained on the full training split and evaluated on the held-out test split; we report accuracy and macro-F1. We evaluate generalization across optimization levels 0–3 and across 24-, 48-, and combined 24+48-qubit settings.

III. RESULTS

A. Optimization-Level Generalization

Table III shows strong within-level recovery, with diagonal accuracy and macro-F1 values of 0.93–0.98. Cross-level transfer is substantially weaker, indicating optimization-dependent feature distributions. The main exception is levels 2 and 3, which transfer strongly in both directions, with accuracy/F1 values of 0.937/0.937 and 0.970/0.970.

TABLE III
OPTIMIZATION LEVEL GENERALIZATION: ACCURACY AND MACRO-F1.

Train \ Test	Level 0	Level 1	Level 2	Level 3
Level 0	Acc 0.958 F1 0.958	Acc 0.499 F1 0.474	Acc 0.300 F1 0.271	Acc 0.285 F1 0.261
Level 1	Acc 0.397 F1 0.372	Acc 0.933 F1 0.931	Acc 0.414 F1 0.359	Acc 0.374 F1 0.312
Level 2	Acc 0.295 F1 0.244	Acc 0.357 F1 0.277	Acc 0.982 F1 0.982	Acc 0.937 F1 0.937
Level 3	Acc 0.254 F1 0.212	Acc 0.357 F1 0.278	Acc 0.970 F1 0.970	Acc 0.978 F1 0.979

B. Circuit-Size Generalization

Table IV shows high within-size recovery, with accuracy/F1 values of 0.971/0.972 for 24-qubit circuits and 0.988/0.988 for 48-qubit circuits. Direct cross-size transfer is weaker (0.607/0.560 and 0.579/0.567), so current normalization reduces but does not remove size dependence. Joint training on 24+48 qubits restores high accuracy on both sizes (0.956/0.956 and 0.988/0.987).

IV. CONCLUSIONS

Final backend-transpiled circuits retain measurable intent-related structure under the proposed normalized features. Intent recovery shows high accuracy and macro-F1 within fixed optimization-level and circuit-size regimes, but cross-regime transfer is limited except between optimization levels

TABLE IV
CIRCUIT-SIZE GENERALIZATION: ACCURACY AND MACRO-F1.

Train \ Test	24	48	24+48
24	Acc 0.971 F1 0.972	Acc 0.607 F1 0.560	Acc 0.792 F1 0.792
48	Acc 0.579 F1 0.567	Acc 0.988 F1 0.988	Acc 0.780 F1 0.786
24+48	Acc 0.956 F1 0.956	Acc 0.988 F1 0.987	Acc 0.971 F1 0.972

2 and 3. Joint training across circuit sizes restores high accuracy, indicating that size dependence can be mitigated when multiple regimes are represented during training. These results support treating transpiled circuits as information-bearing artifacts when sharing, benchmarking, or analyzing compiled quantum workloads.

The weak cross-regime transfer also suggests that intent leakage is shaped by the combined algorithm–transpiler–backend pipeline rather than by the source circuit alone. Consequently, publishing or exchanging transpiled circuits may require regime-specific leakage assessment, especially when compilation settings or backend targets are fixed or known.

V. FUTURE WORK

Future work will evaluate feature importance and ablations, expand the dataset to realistic application circuits, test additional backends and coupling topologies, analyze functional circuit regions, and study adversarial circuit generation for intent obfuscation. A key next step is to distinguish features that are stable leakage channels from those that are artifacts of a specific transpiler or backend. This would support the design of mitigation strategies, such as compilation randomization or structure-preserving transformations that reduce intent recoverability while preserving circuit functionality.

ACKNOWLEDGMENT

We thank Julien Gacon, Stefan Wörner, and Salvador de la Puente for helpful discussions and feedback.

REFERENCES

- [1] S. Kundu and S. Ghosh, “Inverse-Transpilation: Reverse-Engineering Quantum Compiler Optimization Passes from Circuit Snapshots,” 2025, arXiv:2504.19113. [Online]. Available: <https://arxiv.org/abs/2504.19113>
- [2] R. Roy, A. Ghosh, and S. Ghosh, “Forensics of Transpiled Quantum Circuits,” 2024, arXiv:2412.18939. [Online]. Available: <https://arxiv.org/abs/2412.18939>
- [3] A. Ghosh and S. Ghosh, “The Quantum Imitation Game: Reverse Engineering of Quantum Machine Learning Models,” 2024, arXiv:2407.07237. [Online]. Available: <https://arxiv.org/abs/2407.07237>
- [4] S. Xie, A. Sarkar, and S. Feld, “DeQompile: Quantum Circuit De-compilation Using Genetic Programming for Explainable Quantum Architecture Search,” 2025, arXiv:2504.08310. [Online]. Available: <https://arxiv.org/abs/2504.08310>
- [5] A. Javadi-Abhari, M. Treinish, K. Krsulic, C. J. Wood, J. Lishman, J. Gacon, S. Martiel, P. Nation, L. S. Bishop, A. W. Cross, B. R. Johnson, and J. M. Gambetta, “Quantum computing with Qiskit,” 2024, arXiv:2405.08810. [Online]. Available: <https://arxiv.org/abs/2405.08810>