Baltic J. Modern Computing, Vol. 14 (2026), No. 4, pp. 742–763
https://doi.org/10.22364/bjmc.2026.14.4.02

Benchmarking Quantum Hardware with Grover-Search Workloads Derived from Lights Out

Maksims DIMITRIJEVS[1], Maria PALCIHA[2], Abuzer YAKARYILMAZ[1]

[1] Center for Quantum Computer Science, University of Latvia, Latvia
[2] Riga Purvciems Secondary School, Latvia

maksims.dimitrijevs@lu.lv, marija.palciha@gmail.com,
abuzer.yakaryilmaz@lu.lv

ORCID 0000-0002-4225-7889, ORCID 0009-0009-7528-5147, ORCID 0000-0002-2372-252X

Abstract. Application-level quantum-hardware benchmarks are often summarized using circuit metrics such as depth and two-qubit-operation count, but these metrics may not fully describe algorithm-level performance. We construct 9- and 16-qubit Grover-search workloads derived from Lights Out and test them on publicly accessible IBM and IQM processors using different compilation and execution settings and calibration periods. Several compiled versions have similar circuit depth and two-qubit-operation counts but produce different algorithm-level results. Relative device performance also changes across settings and calibration periods, showing that processor revision alone does not determine circuit-specific performance. In addition, nontrivial total marked-state gain does not necessarily mean that the output distribution is close to the ideal Grover result. Smaller Grover circuits are used as diagnostic controls when the main workloads produce near-uniform outputs. These results support using workload-specific measures and considering the device, compilation settings, and calibration period when interpreting quantum-hardware benchmarks.

Keywords: Lights Out, Grover's Search, IBM, IQM, quantum computer

1 Introduction

Current quantum processors are affected by gate and measurement errors, finite relaxation and dephasing times, and hardware-constrained qubit connectivity (Aseguinolaza et al., 2024). Workloads compiled for these processors can be summarized using post-transpilation quantities such as circuit depth and two-qubit-operation count. However, aggregate circuit statistics do not capture all factors that can influence the behavior of a complete algorithm, including circuit structure and device-, compilation-,

and calibration-dependent effects. To evaluate how effectively a processor preserves intended algorithmic behavior, application-level benchmarks are therefore needed. Consequently, researchers have benchmarked quantum algorithms using noisy simulators and real quantum hardware (Mills et al., 2021), (Lubinski et al., 2024), (Kordzanganeh et al., 2023). Yet, interpreting such benchmarks remains challenging when systems or compiled workloads with similar aggregate metrics show different algorithm-level behavior.

We address two research questions:

- RQ1. To what extent do aggregate post-transpilation circuit metrics, such as circuit depth and two-qubit-operation count, explain the algorithm-level performance of related Grover-search workloads on current quantum processors?
- RQ2. How strongly does the observed benchmark performance depend on the target device, compilation and execution configurations, and calibration period?

To address these questions, we use the graph formulation of the Lights Out problem, which lets us vary the oracle structure, search-register size, and active-qubit footprint. We construct two compact graph-based Lights Out workloads: a reflexive instance on a 2×2 grid and a non-reflexive instance on a six-vertex Möbius ladder. The resulting complete Grover-search circuits use 9 and 16 qubits, respectively, and differ in active-qubit footprint and oracle structure, while having similar logical-circuit complexity and, in several compiled configurations, comparable aggregate post-transpilation statistics. This lets us test whether similar conventional circuit summaries correspond to similar algorithm-level behavior.

We evaluate the workloads on publicly accessible IBM and IQM processors with different architectures under multiple compilation and execution settings and across different calibration periods. When the main Lights Out workloads produce near-uniform outputs, we use smaller Grover circuits as diagnostic controls to determine whether amplitude amplification remains observable at lower circuit complexity.

The results show that comparable aggregate circuit statistics do not necessarily correspond to comparable algorithm-level behavior. The two workloads show different relationships among compiled circuit statistics, normalized marked-state gain, and distributional quality, and nontrivial total marked-state gain does not necessarily imply an output distribution close to the ideal Grover result. Relative device performance also varies across compilation and execution settings and calibration periods, indicating that processor revision alone does not determine circuit-specific benchmark performance. Together, these observations support interpreting application-level benchmark results using workload-specific measures and in the context of device, compilation, mapping, and calibration conditions.

The main contributions of this paper are as follows:

- We show, for the tested workloads and processors, that comparable aggregate depth and two-qubit-operation counts do not by themselves explain algorithm-level benchmark behavior. The observed dependence on workload, device, compilation configuration, and calibration period supports the use of workload-specific algorithm-level measures and context-aware interpretation of benchmark results.

- We construct a paired, classically verifiable benchmark from reflexive and non-reflexive Lights Out instances, yielding related Grover-search circuits with different oracle structures and active-qubit footprints while retaining comparable conventional circuit summaries in several compiled configurations.
- We evaluate these workloads across publicly accessible IBM and IQM processors and multiple compilation/execution settings, including repeated measurements across calibration periods, and use smaller Grover circuits as diagnostic controls for near-uniform outputs.

The code, experimental results, and recorded device-calibration data files are publicly available at `https://github.com/infenrio/lights_out_quantum`, and the repository commit `9d64c7c2426b7d298d5cca4e3ed67b5d86d95458` corresponds to this paper.

The remainder of the paper is organized as follows. Section 2 describes Grover's Search and the Lights Out problem. Section 3 constructs the benchmark workloads. Section 4 describes compilation and hardware-execution methods, and Section 5 presents the experimental setup. Section 6 reports the results, Section 7 analyzes and discusses them, and Section 8 concludes the paper.

2 Grover's Search and Lights Out

2.1 Grover's Search

Grover's search provides a quadratic speedup over classical algorithms for unstructured search problems. Implementing a problem-specific oracle together with amplitude amplification produces a complete algorithmic workload involving state preparation, oracle evaluation, uncomputation, diffusion, routing, and measurement. Grover-search circuits are therefore useful as application-level benchmarks: their width and depth can be varied through the problem instance, while their measured behavior remains classically verifiable (AbuGhanem, 2025).

Previous studies have tested quantum computers using Grover's Search for instances of different computational problems. For example, implementations of Grover's Search for different SAT problems highlighted that small instances of problems are difficult for available devices (Vinod and Shaji, 2021), (Bennakhi et al., 2024). Different designs of the SAT problem were considered to improve the solvability of the problem with existing quantum hardware (Alemany Manzanaro and Mora, 2025).

The work in (Joshi et al., 2024) studied how a quantum computer's coupling map (qubit connectivity graph) and noise influence Grover's algorithm. The work in (Akmal Zulfaizal Fadillah et al., 2021) showed that simpler architectures can yield higher success probabilities than bigger but noisier ones. Although noise can reduce the practical advantage of Grover search on current hardware, Grover-based methods continue to be studied for different combinatorial problem classes (Jipipob and Chongstitvatana, 2025), (Haverly and López, 2021). Grover-based subroutines have also been applied to graph-theoretic optimization problems. Ambainis and Repko developed quantum algorithms for the DOMATIC NUMBER problem using quantum maximum finding, which is based on Grover search (Ambainis and Repko, 2024).

Our work complements previous studies by constructing a paired benchmark from two graph-based variants from the Lights Out family and comparing workloads whose conventional circuit summaries can be similar while their active-qubit footprints and oracle structures differ.

2.2 Lights Out

The Lights Out game appeared as an electronic puzzle game in 1995, released by Tiger Electronics (Anderson and Feil, 1998). The game field consists of lamps that are arranged in a grid 5×5. When the game starts, some of the lamps are switched on. When a lamp is clicked, the state of it and its adjacent neighbors is changed (on $\leftrightarrow$ off). The objective is to switch all the lamps off. The game can also have an additional objective, such as using the minimum possible moves.

The Lights Out game has been popular within the Mathematics and Computer Science communities, as it represents an interesting combinatorial puzzle. It was mathematically investigated in (Anderson and Feil, 1998). In (Berman et al., 2021), different formulations of Lights Out on different graphs were presented, as well as evaluations of the computational complexity of different formulations of the problem.

The graph-based formulation makes Lights Out suitable for constructing related, classically verifiable benchmark workloads. Different graph structures alter the neighborhood relation encoded by the oracle while preserving a common computational interpretation, and reflexive and non-reflexive variants additionally differ in whether clicking a vertex toggles the clicked vertex itself. An online technical article presented a quantum approach to a 3×3 Lights Out instance (Vidwans, 2021). In this work, we select smaller single-click instances on a 2×2 grid and a six-vertex Möbius ladder so that complete Grover circuits can be executed on publicly accessible hardware. The two instances provide different oracle structures and active-qubit footprints while retaining known ideal outputs.

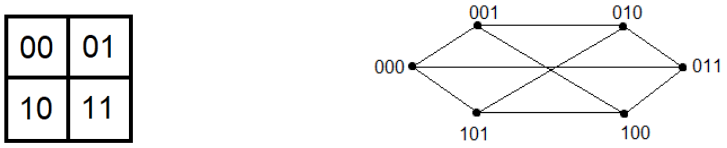
3 Construction of the benchmark workloads

We construct two classically verifiable Lights Out instances. The two workloads share the same single-click search formulation and use different graph rules: the grid instance is reflexive, whereas the Möbius-ladder instance is non-reflexive. The search register encodes a candidate lamp index, and the oracle marks an index if clicking the corresponding lamp transforms the selected initial configuration into the all-off state. We restricted the instances to single-click solutions to keep the search spaces compact, and this permitted a complete implementation with one Grover iteration. The circuits are used as benchmark workloads rather than as evidence of computational speedup over classical Lights Out solvers.

In our design, we use state `0` to mark a specific lamp as switched off, and state `1` to mark a specific lamp as switched on.

3.1 Lights Out instance for 2 × 2 grid

Given an initial configuration of 4 lamps in a 2 × 2 grid, the task is to find which lamp we need to click in order to make all 4 lamps switched off. On a grid, clicking a lamp changes its state and the states of all direct neighbors (right, left, top, or bottom). In the case of a 2 × 2 grid, clicking a specific lamp switches the states of 3 lamps in total, leaving only the opposite-diagonal one unaffected.



(a) Grid with 2 × 2 lamps. **(b)** Möbius ladder with 6 lamps.

Figure 1. Graphs and basis states for the implementation of Lights Out instances.

In our implementation, we use 2 qubits to store the indices of the lamps to be clicked. Lamps are represented by states on the top row as `00` and `01`, and on the bottom row as `10` and `11` (see Fig. 1a). We use 4 qubits to store the initial configuration of 4 lamps and the updated configuration after the click operation. We use one qubit to compute the oracle function value, and this qubit will also flip the phase of the correct solution. We add 2 auxiliary qubits, as we implement the NOT operator controlled by 4 qubits with 3 Toffoli operations.

The implementation of the circuit is shown in Fig. 2. The qubits q_0 and q_1 store the index of the lamp to be clicked, and Grover's Search space is initialized in superposition with Hadamard operators. The qubits q_2, q_3, q_4, and q_5 store the states of the lamps. We apply X gates where needed before the barrier to set the initial configuration. To reduce the number of multi-qubit operations in the oracle, we remark that clicking one lamp changes the states of all lamps except one; therefore, we switch the state of all lamps with an X gate, and switch the state again for the lamp that is not a neighbor of the clicked one. As a result, the circuit contains 14 Toffoli gates in total. The last 5 layers before the measurement operator implement Grover's Diffusion operation.

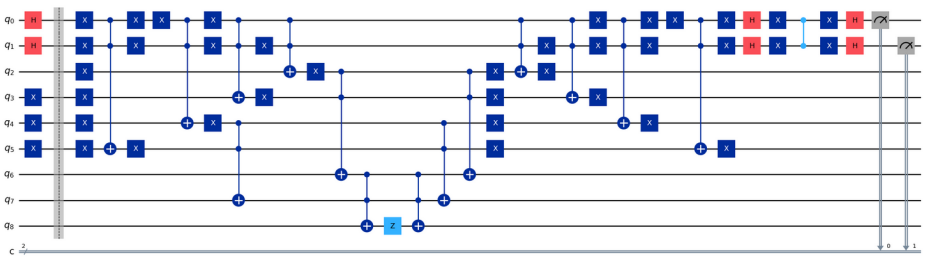


Figure 2. The 9-qubit Grover benchmark circuit for the 2 × 2 Lights Out instance.

The logical circuit uses 9 qubits and, before compilation, contains 14 Toffoli gates, one CZ gate, and 31 layers before measurement. The search space contains one marked state among four candidates. After one Grover iteration, the ideal marked-state `11` probability is 1, whereas a uniform output assigns probability $\frac{1}{4}$ to the marked state.

3.2 Lights Out instance for Möbius ladder with 6 lamps

We are given $n \geq 6$ lamps arranged in a circle, where n is an even number. By clicking a lamp, the states of both neighbors and the lamp on the opposite side of the circle are changed. Therefore, if the lamps are sequentially indexed around the circle and we click a lamp indexed by a, then the states of the lamps indexed by $(a+1) \mod n$, $(a-1) \mod n$, and $(a+n/2) \mod n$ are changed. This design corresponds to non-reflexive Lights Out on a Möbius ladder. Each lamp is connected to three other lamps, and clicking a lamp affects only the states of direct neighbors.

The graph structure for 6 lamps, with their enumeration with basis states, is shown in Fig. 1b. For the six-vertex instance and the indexing shown in Fig. 1b, clicking a lamp toggles the three lamps whose least-significant index bit is opposite to that of the selected lamp. This lets us simplify the circuit. Based on the least-significant bit of the clicked-lamp index, we can switch the states of the corresponding 3 lamps. The invalid indices `110` and `111` are exactly the states for which $q_2 = q_1 = 1$, irrespective of q_0; therefore, we add a validity check that excludes search-register states whose two most-significant index bits are both equal to 1.

In this implementation, we use 3 qubits to store the indices of the lamps to be clicked. We use 6 qubits to store the initial configuration of our lamps, and the updated configuration after the clicking operation. We use one more qubit to store the verification that a lamp with an invalid index (`110` or `111`) has not been clicked. We use the NOT operator, controlled by 7 qubits (six lamp-state qubits and the validity-check qubit); therefore, we add 5 more auxiliary qubits to implement control with 6 RCCX gates (RCCX gates serve as an efficient alternative to the Toffoli gates). The last one is the oracle target qubit that is used for phase marking.

The implementation of the circuit is depicted in Fig. 3. The qubits q_0, q_1, and q_2 store the index of the lamp to be clicked, and the search register is initialized in superposition with Hadamard operators. The qubits q_3, q_4, q_5, q_6, q_7, and q_8 store the states of the lamps. The initial configuration is given as input – X-operations before the barrier on the left side. For the oracle implementation, we implement two checks. First, the state of q_0 affects the choice of 3 lamps for which the state should be changed. Second, q_1 and q_2 are checked to ensure that they are not simultaneously in state `1`; this excludes the invalid indices `110` and `111`, and the result of the check is stored in q_9. The last layers of the circuit on the first 3 qubits implement Grover's Diffusion operation.

The logical circuit uses 16 qubits and, before compilation, contains three Toffoli operations, 12 RCCX gates (in Fig. 3 denoted as R), 12 two-qubit gates (CNOTs), and 32 layers before measurement. The search register represents eight basis states, of which six correspond to lamps, and two are invalid indices. Three valid states are marked (`001`, `011`, and `101`). A uniform distribution therefore assigns total probability $\frac{3}{8} = 0.375$ to the marked set. After one ideal Grover iteration, the total marked-state probability is $\frac{27}{32} = 0.84375$, with probability $\frac{9}{32} = 0.28125$ for each marked state.

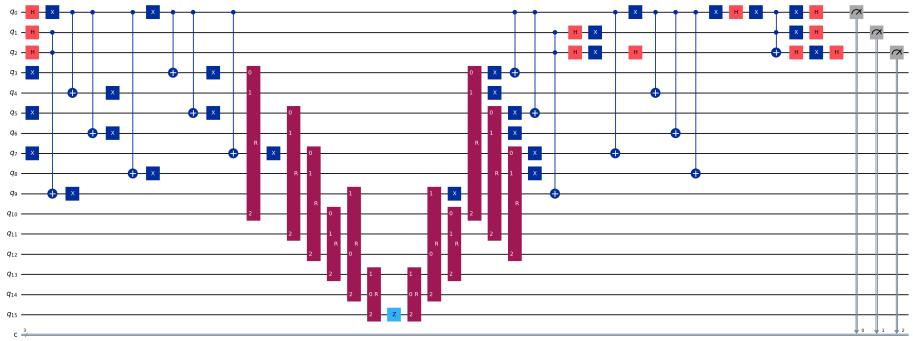


Figure 3. The 16-qubit Grover benchmark circuit for the six-vertex Möbius-ladder instance.

3.3 Diagnostic control circuits

We use two smaller Grover circuits as diagnostic controls when the main benchmark produces a marked-state frequency close to its uniform baseline. The first is a five-qubit Grover circuit for SAT, and the second is a two-qubit Grover circuit. These controls test whether the expected output remains observable at lower circuit complexity.

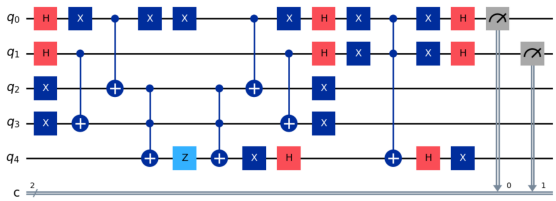


Figure 4. The 5-qubit Grover diagnostic-control circuit for a small SAT problem.

The circuit that we prepared for the small SAT instance is shown in Fig. 4. Here, q_0 and q_1 serve as the formula variables, and the SAT formula to solve is $(q_0) \wedge (\neg q_1)$. q_2 and q_3 store the calculated values for clauses, and q_4 is used to perform phase change. The expected output of the circuit is state `01` with probability 1. The uniform probability of the marked state is $\frac{1}{4}$.

The second diagnostic control is a two-qubit Grover circuit with one marked state among four candidates (see Fig. 5). It contains two two-qubit operations, has marked state `10`, and produces that state with ideal probability 1 after one Grover iteration. The uniform probability of the marked state is $\frac{1}{4}$.

4 Methods for running circuits on quantum hardware

To run the quantum circuit on a quantum computer, the circuit is first translated to the native operations of the computer. Qiskit has the option of using a transpiler that per-

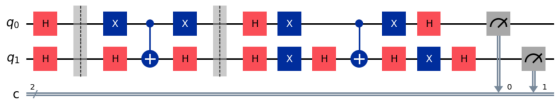


Figure 5. Two-qubit Grover diagnostic-control circuit.

forms the translation of the circuit to map the qubits and operations of the circuit to the actual hardware architecture. The transpiler has an option to provide the *optimization_level* parameter, where higher levels enable more extensive optimization and layout searches in a way that the circuit may have lower depth, number of gates, and use qubits and their connectivity more optimally (e.g., picking qubits that showed better results at calibration benchmarks). Higher levels do not guarantee a monotonic reduction in depth or operation count. Different runs of the transpiler for the same circuit with the same parameters may produce different resulting circuits (i.e., some randomness is involved in the transpilation process, and, additionally, differences in hardware calibration data also affect the choices made by the transpiler).

For the IBM experiments, we use the SamplerV2 primitive, in which case the circuit with measurements is launched on a quantum computer, and we receive measurement outcomes as bitstring distributions over the total number of `shots`. For the 16-qubit workload, we additionally used the IBM EstimatorV2 primitive. Here, EstimatorV2 evaluated a projector onto the three marked states of the logical search register. The observable acts as identity on all other qubits, so its expectation value is exactly the marginal probability mass of the marked outputs. The observable was remapped according to the transpiled circuit layout, and the paper reports the raw Runtime expectation estimate and its Runtime-reported standard error under built-in *resilience_level* set to 0, 1, and 2. The resilience levels invoke IBM's built-in noise-management presets with increasing sampling and postprocessing overhead. No mitigation techniques were implemented or configured individually in this work.

IBM hardware executions used SamplerV2 and, for the 16-qubit workload, EstimatorV2 from Qiskit IBM Runtime version 0.34.0. For 5 consecutive-day experiments on a 2×2 grid, the Qiskit IBM Runtime version was 0.45.0. Circuits were transpiled using Qiskit version 1.1.1 (and Qiskit version 2.3.0 for 5 consecutive-day experiments on a 2×2 grid) with optimization levels reported below. No transpiler seed was fixed. IQM circuits were transpiled with Qiskit and submitted through the IQM Qiskit backend interface (IQMProvider and backend.run) rather than IBM Runtime primitives. Except for the transpilation optimization levels and EstimatorV2 resilience levels specified here, and the shot counts reported in Section 5, no additional execution options were explicitly configured.

The EstimatorV2 with a high *resilience_level* consumes significantly more runtime on IBM hardware than the SamplerV2. This limited the number of experiments that we performed during this research.

5 Setup for experiments

Each benchmark and diagnostic-control circuit described in Section 3 was first validated using AerSimulator as a noiseless reference. We verified the expected marked outputs and their ideal probabilities before executing the circuits on hardware. For shot-based experiments, the primary measured quantity is the marked-state frequency $\hat{p}_{marked} = \frac{N_{marked}}{N}$, where N_{marked} is the number of shots assigned to theoretically marked search-register outputs and N is the total number of shots. For the 9-qubit grid workload, the uniform and ideal marked-state probabilities are $\frac{1}{4}$ and 1, respectively. For the 16-qubit Möbius-ladder workload, they are $\frac{3}{8}$ and $\frac{27}{32}$, respectively.

Because these reference probabilities differ between the workloads, raw marked-state frequencies are not directly comparable across them. For cross-workload comparisons, we therefore use the normalized marked-state gain $G_{norm} = \frac{\hat{p}_{marked} - p_{uniform}}{p_{ideal} - p_{uniform}}$. A value of zero corresponds to the uniform marked-state baseline, and a value of one corresponds to the ideal one-iteration Grover result. Negative values indicate suppression below the uniform baseline. This measure summarizes total marked-state mass and is not a full-distribution fidelity measure.

The 9-qubit grid workload was evaluated on the IBM processors: Heron r2 devices `ibm_marrakesh` and `ibm_fez`, and Heron r1 device `ibm_torino`. The circuit was compiled separately for each device using Qiskit *optimization_level* 0, 1, 2, and 3 and executed with the SamplerV2 primitive. We also executed a circuit prepared using IBM Quantum Composer. To examine temporal variation, we repeated the grid experiment on five consecutive days using *optimization_level* 2 and Composer on all three IBM devices. The corresponding execution dates were January 14-18, 2026.

The 9-qubit workload was also executed on the IQM processors: Emerald and Garnet with superconducting transmon qubits on a square lattice, and IQM star topology quantum processor Sirius. We used *optimization_level* 0, 1, 2, and 3. The marked-state frequencies remained close to the uniform baseline for these executions. We therefore added a five-qubit Grover SAT circuit and a two-qubit Grover circuit as diagnostic controls. The SAT control was executed on all IBM and IQM processors using *optimization_level* 0 and 3. The two-qubit control was executed on all IQM processors using default settings. These controls test whether concentration on the expected output remains observable at lower circuit complexity; they are not used to identify a specific physical error mechanism.

We developed the 16-qubit Möbius-ladder circuit iteratively – the logical implementation was simplified using preliminary executions on the strongest-performing IBM device from the initial grid tests. After the design was finalized, the logical circuit was fixed and evaluated on all reported devices. On IBM hardware, the final workload was executed using SamplerV2 with *optimization_level* 3, EstimatorV2 with *resilience_level* 0, 1, and 2, and IBM Quantum Composer. EstimatorV2 was used only for this workload. The Composer executions were performed on a different date from the SamplerV2 and EstimatorV2 executions. We recorded the calibration data, the actual circuits that were launched, and outcomes. On IQM hardware, the final workload was executed through the IQM backend interface with *optimization_level* 2 and 3.

For each execution, we recorded the calibration data of the computers and the data about the circuits that were actually launched on the devices after the compilation.

As of January 2026, the public-access quotas available to us provided 10 minutes of monthly IBM runtime and approximately 1 minute of monthly IQM runtime. These limits constrained the number of repetitions and motivated different shot budgets across providers. For IBM devices, we launched the circuits with parameter `shots` set to 4000, whereas for IQM devices, parameter `shots` was set to 1000. The sampling uncertainty (1σ) for a reported frequency $\hat{p}$ from N shots is $\sqrt{\hat{p}(1-\hat{p})/N}$, with the worst-case bound $\leq 1/(2\sqrt{N})$, i.e., ≤ 0.0079 for $N = 4000$ and ≤ 0.0158 for $N = 1000$. EstimatorV2 uncertainties are reported separately using the standard errors returned by IBM Runtime.

5.1 Software and reproducibility information

All hardware experiments reported in this work were performed between 9 January and 5 February 2026.

First reported experiments on a 2×2 grid on IBM were performed on 10 January 2026; **reported experiments on Möbius ladder with 6 lamps on IBM** were performed on 12 January 2026 (except experiments that used Composer – those were performed on 23 January 2026), while **reported experiments on diagnostic control circuits on IBM** were performed on 5 February 2026. Implementations used Python version 3.8.8, Qiskit version 1.1.1, qiskit-aer 0.14.2, qiskit-ibm-runtime 0.34.0. **Reported experiments on a 2×2 grid on IBM for 5 consecutive days** were performed between 14 and 18 January 2026 and used Python version 3.12.12, Qiskit version 2.3.0, qiskit-aer 0.17.2, and qiskit-ibm-runtime 0.45.0.

All reported experiments on IQM, except diagnostic control circuit runs, were performed on 1 February 2026. Implementations used Python version 3.11.14, Qiskit version 1.4.2, qiskit-aer 0.17.2, and iqm-client 33.0.2. Diagnostic control circuit runs were performed on 9 January 2026.

6 Results

We report the probability mass assigned to the theoretically marked search-register states as the primary algorithm-level result. For shot-based executions (IBM SamplerV2, IQM backend executions, and IBM Quantum Composer), this quantity is the observed marked-state frequency. For EstimatorV2, it is the expectation value of the marked-subspace projector defined in Section 4. Raw marked-state probabilities are used for within-workload comparisons. For comparisons between the 9- and 16-qubit workloads, we additionally use the normalized marked-state gain G_{norm} defined in Section 5.

Post-transpilation circuit statistics are reported using the number of active physical qubits, two-qubit operations, circuit depth, and circuit size. Circuit depth and size were obtained using Qiskit's `QuantumCircuit.depth()` and `QuantumCircuit.size()` methods after transpilation. The two-qubit-operation count includes all two-qubit operations after transpilation. Measurements were included in depth and size.

The results are presented first within each workload and then across workloads: within-workload comparisons characterize device, configuration, and temporal variation, while the normalized marked-state gain enables comparison of algorithm-level

Table 1. Measured distributions for the 9-qubit grid workload on IBM hardware. For each configuration, the first row reports the count and relative frequency of the marked state `11`; the second row reports the remaining outcomes. Each execution used 4000 `shots`.

Configuration	**ibm_marrakesh**	**ibm_fez**	**ibm_torino**
Composer	**11: 2304 (0.576)**	**11: 1645 (0.411)**	**11: 954 (0.239)**
	00: 515 01: 665 10: 516	00: 759 01: 828 10: 768	00: 1014 01: 950 10: 1082
opt=3	**11: 2018 (0.505)**	**11: 1417 (0.354)**	**11: 1796 (0.449)**
	00: 563 01: 716 10: 703	00: 853 01: 933 10: 797	00: 599 01: 896 10: 709
opt=2	**11: 1896 (0.474)**	**11: 1528 (0.382)**	**11: 1633 (0.408)**
	00: 621 01: 731 10: 752	00: 842 01: 814 10: 816	00: 709 01: 858 10: 800
opt=1	**11: 1780 (0.445)**	**11: 1460 (0.365)**	**11: 1553 (0.388)**
	00: 644 01: 868 10: 708	00: 718 01: 865 10: 957	00: 754 01: 934 10: 759
opt=0	**11: 1370 (0.343)**	**11: 1035 (0.259)**	**11: 1081 (0.270)**
	00: 749 01: 1090 10: 791	00: 937 01: 1097 10: 931	00: 942 01: 1091 10: 886

behavior across the 9- and 16-qubit workloads. Diagnostic controls are reported separately to interpret cases in which the main workloads produce near-uniform outputs.

6.1 Lights Out on 2×2 grid

For the 9-qubit grid workload, one search-register state is marked. The uniform marked-state probability is therefore $\frac{1}{4} = 0.25$, while the ideal probability after one Grover iteration is 1. We use the measured frequency of state `11` as the workload-success measure.

We summarize the results of our experiments on IBM devices in Table 1. The marked-state frequency ranged from 0.239 to 0.576 across the tested devices and execution configurations. The highest value, 0.576, was obtained using Composer on `ibm_marrakesh`. For Qiskit SamplerV2 executions, *optimization_level* 0 produced the lowest marked-state frequency on all three devices. Relative device performance was configuration-dependent: Composer favored `ibm_fez` over `ibm_torino`, whereas *optimization_level* 1-3 favored `ibm_torino` over `ibm_fez`. Overall, `ibm_marrakesh` produced the largest marked-state frequency across all configurations.

We summarize the statistics of the circuits that were actually run on IBM hardware in Table 2. The IBM compilations all used nine active physical qubits. Optimization level 0 produced the largest circuits, with 310-376 two-qubit operations and depths of 712-774. Composer and optimization levels 2-3 produced substantially fewer two-qubit operations, although the lowest aggregate circuit statistics did not correspond to the same relative device performance in every execution configuration.

The results of running the circuits on IQM devices are summarized in Table 3. The marked-state frequencies ranged from 0.218 to 0.289, compared with the uniform baseline of 0.25. Thus, the measured concentration on state `11` was substantially weaker than in the best IBM executions. The frequencies of all states ranged from 0.211 to 0.291.

We report the statistics of the circuits that were actually run on IQM hardware in Table 4. The IQM compiled realizations used between 9 and 12 active physical qubits,

Table 2. Statistics for post-transpilation circuit for a 2×2 grid of lamps (all circuits use 9 active qubits). The data is shown for five different execution configurations – IBM Quantum Composer, and SamplerV2 with *optimization_level* value set to 3, 2, 1, and 0.

	marrakesh			fez			torino		
Configuration	2q op	depth	size	2q op	depth	size	2q op	depth	size
Composer	170	293	633	167	295	631	173	341	650
opt=3	165	397	714	173	437	754	170	430	720
opt=2	175	405	756	177	408	749	175	399	747
opt=1	217	467	932	229	554	962	199	455	882
opt=0	328	715	1560	376	774	1704	310	712	1506

Table 3. Measured counts for the marked state `11` (out of 1000 `shots`) for the circuit for a 2×2 grid of lamps. The data is shown for four different execution configurations – IQM backend execution with *optimization_level* value set to 3, 2, 1, and 0.

Configuration	**Emerald**	**Garnet**	**Sirius**
opt=3	266	258	277
opt=2	218	230	269
opt=1	246	289	277
opt=0	265	284	242

depending on device and *optimization_level* . Higher *optimization_level* generally reduced the two-qubit-operation count and depth relative to *optimization_level* 0. However, these more compact realizations did not produce marked-state frequencies substantially above the uniform baseline.

In the five-day experiment, the marked-state frequency varied from 0.216 to 0.533 on `ibm_marrakesh`, from 0.278 to 0.506 on `ibm_fez`, and from 0.240 to 0.450 on `ibm_torino` (see Table 5). No device ranked first in every day-execution-configuration combination. These repeated measurements show variation across time and configurations; their possible relation to calibration and layout is discussed in Section 7.

The results for the 9-qubit workload show substantial variation across configurations and time. On IBM hardware, marked-state concentration varies across devices, compilation settings, and repeated execution periods, while the IQM executions remain close to the uniform baseline despite comparatively compact compiled circuits. Thus, for this workload, lower aggregate depth or two-qubit-operation count does not consistently correspond to stronger marked-state concentration.

6.2 Lights Out on Möbius ladder with 6 lamps

The 16-qubit workload has three marked search-register states (`001`, `011`, and `101`) among eight basis states. A uniform distribution therefore assigns total probability

Table 4. Statistics for post-transpilation circuit for 2×2 grid of lamps. The data is shown for four different execution configurations – IQM backend execution with *optimization_level* value set to 3, 2, 1, and 0.

	Emerald				Garnet				Sirius			
Configuration	qub	2q op	depth	size	qub	2q op	depth	size	qub	2q op	depth	size
opt=3	9	151	246	403	9	151	232	402	10	202	221	322
opt=2	9	152	230	392	9	145	227	397	10	202	221	322
opt=1	11	166	246	418	10	178	257	454	10	197	215	315
opt=0	12	244	673	1303	9	211	694	1204	10	227	582	968

Table 5. Observed frequencies of the marked state `11` for the circuit for 2×2 grid of lamps. The experiments were run for five consecutive days (D_x denotes Day x), and for two different configurations – SamplerV2 with *optimization_level* 2 and IBM Quantum Composer.

Device	D_1 opt	D_1 comp	D_2 opt	D_2 comp	D_3 opt	D_3 comp	D_4 opt	D_4 comp	D_5 opt	D_5 comp
marrakesh	0.440	0.533	0.227	0.216	0.444	0.278	0.449	0.312	0.453	0.357
fez	0.392	0.309	0.430	0.506	0.456	0.392	0.382	0.284	0.278	0.332
torino	0.450	0.257	0.332	0.406	0.437	0.401	0.331	0.240	0.296	0.259

$\frac{3}{8} = 0.375$ to the marked set. After one ideal Grover iteration, the total marked-state probability is $\frac{27}{32} = 0.84375$, with probability $\frac{9}{32} = 0.28125$ for each marked state.

We summarize the results of our experiments on IBM devices in Table 6. With SamplerV2 at *optimization_level* 3, the total marked-state frequencies were 0.4355, 0.390, and 0.374 on `ibm_marrakesh`, `ibm_fez`, and `ibm_torino`, respectively. The first two values are above the uniform baseline of 0.375, while the `ibm_torino` value is approximately equal to it. Composer produced values at or below the uniform baseline on all three devices. EstimatorV2 produced its highest point estimates at *resilience_level* 2: 0.5358 ± 0.0957, 0.4280 ± 0.1223, and 0.4594 ± 0.0279, respectively. These estimates should be interpreted together with their Runtime-reported standard errors, which are considerably larger than those at *resilience_level* 0 and 1 for `ibm_marrakesh` and `ibm_fez`. All reported point estimates remain substantially below the ideal probability 0.84375.

We summarize the statistics of the circuits that were actually run on IBM hardware in Table 7. The finalized logical workload uses 16 qubits. SamplerV2 and EstimatorV2 compilations used 16 active physical qubits, whereas Composer used 19 qubits on `ibm_marrakesh` and 17 on `ibm_torino`. Across the reported IBM configurations, the circuit contained 191-243 two-qubit operations and had depth 206-257. These aggregate statistics overlap with, and in some cases are smaller than, those of the 9-qubit workload, despite the larger active-qubit footprint.

Table 6. Total marked-state results for the 16-qubit Möbius-ladder workload. The marked set is $S = \{001, 011, 101\}$. SamplerV2 and Composer entries are shot-based frequencies obtained by summing the counts of the three marked outputs. EstimatorV2 entries are expectation estimates of the marked-subspace projector, with the Runtime-reported standard error. EstimatorV2 was used only for this workload on IBM hardware.

Device	Sampler (opt=3)	Estimator (res=0)	Estimator (res=1)	Estimator (res=2)	Comp.
marrakesh	0.4355	0.4637 ± 0.0076	0.4809 ± 0.0085	0.5358 ± 0.0957	0.3753
fez	0.390	0.3398 ± 0.0076	0.3423 ± 0.0074	0.4280 ± 0.1223	0.3345
torino	0.374	0.4293 ± 0.0075	0.4307 ± 0.0074	0.4594 ± 0.0279	0.373

Table 7. The statistics for the post-transpilation circuit for 6 lamps. The data is shown for three different execution configurations – SamplerV2 with *optimization_level* 3, EstimatorV2 with *resilience_level* 2, and IBM Quantum Composer.

	marrakesh				fez				torino			
Configuration	qub	2q op	depth	size	qub	2q op	depth	size	qub	2q op	depth	size
opt=3	16	191	215	802	16	191	213	796	16	203	234	798
res=2	16	191	214	799	16	191	212	793	16	203	228	790
Composer	19	243	227	748	16	219	206	685	17	234	257	739

On IBM hardware, the 16-qubit realizations with *optimization_level* 3 had depths of 213-234, compared with 397-437 for the corresponding 9-qubit realizations. Their two-qubit counts were of similar order: 191-203 for the 16-qubit workload and 165-173 for the 9-qubit workload. Nevertheless, the 16-qubit executions produced lower normalized marked-state gain than the corresponding 9-qubit executions on all three IBM devices.

The results of running the circuits on IQM devices are summarized in Table 8. The total marked-state frequency ranged from 0.305 to 0.475. Several configurations produced total marked-state mass above the uniform baseline. However, the probability was not distributed evenly among the three marked outputs. In every reported execution, state `101` occurred with frequency below its uniform per-state baseline of $\frac{1}{8} = 0.125$, while state `001` was strongly overrepresented in several cases.

We summarize the statistics of the circuits that were actually run on IQM hardware in Table 9. On all three devices, the 16-qubit workload compiled to lower depth than the 9-qubit workload with *optimization_level* 2 and 3. It also used fewer two-qubit operations in most corresponding configurations, despite requiring 16 or 17 active physical qubits rather than 9 or 10. These smaller circuit metrics did not produce output distributions closer to the ideal Grover result.

To compare the two workloads on a common scale, we calculated the normalized marked-state gain G_{norm} (see Table 10). For IBM SamplerV2 executions at *optimization_level* 3, the grid and Möbius values were 0.340 and 0.129 on `ibm_marrakesh`,

Table 8. Marked-state results for the 16-qubit Möbius-ladder workload. The marked set is $S = \{001, 011, 101\}$. The data is shown for `shots` set to 1000 and two different execution configurations – IQM backend execution with *optimization_level* value set to 3 and 2.

Configuration	**Emerald**	**Garnet**	**Sirius**
opt=3	**total: 441 (0.441)** 001: 191 011: 148 101: 102	**total: 397 (0.397)** 001: 111 011: 198 101: 88	**total: 465 (0.465)** 001: 235 011: 148 101: 82
opt=2	**total: 346 (0.346)** 001: 113 011: 130 101: 103	**total: 305 (0.305)** 001: 110 011: 96 101: 99	**total: 475 (0.475)** 001: 238 011: 153 101: 84

Table 9. Statistics for post-transpilation circuit for 6 lamps. The data is shown for two different execution configurations – IQM backend execution with *optimization_level* value set to 3 and 2.

	Emerald				Garnet				Sirius			
Configuration	qub	2q op	depth	size	qub	2q op	depth	size	qub	2q op	depth	size
opt=3	17	106	112	294	16	104	102	285	17	169	179	271
opt=2	16	109	103	304	17	123	106	333	17	169	179	271

0.139 and 0.032 on `ibm_fez`, and 0.265 and approximately 0 on `ibm_torino`, respectively. Thus, on the tested IBM devices, the 16-qubit workload recovered a smaller fraction of the ideal gain above its uniform baseline despite having lower compiled depth and a comparable two-qubit-operation count.

The IQM comparison was different. At *optimization_level* 3, the grid G_{norm} values were 0.021, 0.011, and 0.036 on Emerald, Garnet, and Sirius, whereas the corresponding Möbius values were 0.141, 0.047, and 0.192. The larger total marked-state gain for the Möbius workload did not imply closer agreement with the ideal distribution: Table 8 shows substantial imbalance among its three marked outputs.

The cross-workload comparison shows that smaller aggregate circuit statistics did not consistently correspond to better algorithm-level output. On IBM hardware, the 16-qubit workload has lower compiled depth and comparable two-qubit-operation counts in the optimization-level-3 comparison, yet lower normalized marked-state gain than the 9-qubit workload. On IQM hardware, the 16-qubit workload can show larger total normalized gain while still showing substantial imbalance among the marked outputs. Thus, neither aggregate circuit statistics nor total marked-state probability alone fully describes algorithm-level behavior.

6.3 Diagnostic control circuits (SAT and two-qubit Grover)

Each diagnostic control has one marked output among four possible search-register states. The uniform marked-state probability is therefore 0.25, while the ideal marked-state probability is 1.

Table 10. Cross-workload comparison using shot-based executions at *optimization_level* 3. Comparisons are made within providers; executions were performed on different dates and should be interpreted descriptively rather than as simultaneous controlled trials.

Provider	Device	Grid $\hat{p}_{marked}$	Grid G_{norm}	Möbius $\hat{p}_{marked}$	Möbius G_{norm}
IBM	marrakesh	0.505	0.340	0.4355	0.129
IBM	fez	0.354	0.139	0.390	0.032
IBM	torino	0.449	0.265	0.374	-0.002
IQM	Emerald	0.266	0.021	0.441	0.141
IQM	Garnet	0.258	0.011	0.397	0.047
IQM	Sirius	0.277	0.036	0.465	0.192

Table 11. Measured output distributions (1000 `shots` each) for the SAT diagnostic control circuit. The first line contains the number of counts for the marked state `01`, together with relative frequency, and the second line contains the remaining counts. The data is shown for IQM backend executions with *optimization_level* 3 and 0.

Device	opt=3	opt=0
Emerald	**01: 409 (0.409)**	**01: 241 (0.241)**
	00: 148 10: 135 11: 308	00: 210 10: 264 11: 285
Garnet	**01: 630 (0.630)**	**01: 509 (0.509)**
	00: 105 10: 68 11: 197	00: 128 10: 190 11: 173
Sirius	**01: 258 (0.258)**	**01: 260 (0.260)**
	00: 250 10: 237 11: 255	00: 263 10: 255 11: 222

We summarize the results of running the SAT circuit on IQM devices in Table 11. Garnet produced marked-state frequencies of 0.630 and 0.509 with *optimization_level* 3 and 0, respectively. Emerald produced 0.409 at *optimization_level* 3 but remained close to the uniform baseline at *optimization_level* 0. Sirius remained close to the uniform baseline under both settings. Thus, expected-output concentration was observable on the smaller control for some devices and compilation settings, even when the 9-qubit grid workload remained close to its uniform baseline, and the 16-qubit workload showed distorted marked-state composition.

We summarize the results of running the SAT circuit on IBM devices in Table 12. With *optimization_level* 3, SAT produced marked-state frequencies of 0.860, 0.790, and 0.835 on `ibm_marrakesh`, `ibm_fez`, and `ibm_torino`, respectively. At *optimization_level* 0, the corresponding frequencies were 0.484, 0.242, and 0.548. Therefore, the control also demonstrates substantial sensitivity to compilation settings.

The results for running the two-qubit Grover diagnostic control on IQM devices are summarized in Fig. 6.

The diagnostic controls show that near-uniform or distorted outputs from the main Lights Out workloads do not coincide with a complete absence of observable Grover

Table 12. Measured output distributions (4000 `shots` each) for the SAT diagnostic control circuit. The first line contains the number of counts for the marked state `01`, together with relative frequency, and the second line contains the remaining counts. The data is shown for SamplerV2 configuration with *optimization_level* 3 and 0.

Device	opt=3	opt=0
ibm_marrakesh	**01: 3438 (0.860)**	**01: 1935 (0.484)**
	00: 201 10: 123 11: 238	00: 1084 10: 352 11: 629
ibm_fez	**01: 3161 (0.790)**	**01: 966 (0.242)**
	00: 310 10: 191 11: 338	00: 1791 10: 686 11: 557
ibm_torino	**01: 3339 (0.835)**	**01: 2193 (0.548)**
	00: 259 10: 157 11: 245	00: 735 10: 414 11: 658

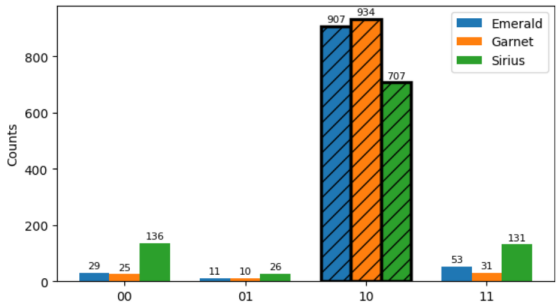


Figure 6. Output distributions for the two-qubit Grover diagnostic control on IQM devices, using 1000 `shots` per device. The expected output is state `10`, with uniform probability 0.25 and ideal probability 1. The measured frequencies of state `10` are 0.907 on Emerald, 0.934 on Garnet, and 0.707 on Sirius.

amplification at lower circuit complexity. They serve as interpretive controls rather than as additional benchmark workloads or evidence for a specific hardware error mechanism.

The results show two main patterns. First, comparable or smaller aggregate depth and two-qubit-operation counts do not necessarily correspond to stronger algorithm-level performance across the paired workloads. Second, relative device performance varies across compilation and execution settings and repeated measurement periods. The IQM results additionally show that total marked-state gain can obscure substantial differences in marked-state composition, while the diagnostic controls confirm that expected-output concentration remains observable at lower circuit complexity.

7 Result analysis and discussion

This section interprets the empirical patterns reported in Section 6 in relation to the two research questions. We first examine device-, configuration-, and time-dependent variation, and then assess whether aggregate circuit statistics fully describe algorithm-level

behavior. We distinguish direct experimental observations from possible hardware-level explanations, which are not isolated independently by the present experiments.

7.1 Device, compilation, and temporal sensitivity

For the 9-qubit workload, the relative device performance depended on the execution configuration. Composer gave the highest marked-state frequency on `ibm_marrakesh`, while `ibm_fez` outperformed `ibm_torino`. In contrast, SamplerV2 with *optimization_level* 1-3 favored `ibm_torino` over `ibm_fez`. Processor revision therefore did not determine a consistent ranking across the tested configurations.

The five-day experiment further demonstrated temporal variation. No device ranked first under every day-execution-configuration combination, and each processor showed a substantial range of marked-state frequencies. Because the daily executions were associated with different calibration snapshots and potentially different physical layouts, this variation cannot be attributed solely to processor revision. The experiment establishes configuration- and time-dependent behavior, but does not isolate the causal contribution of individual calibration metrics.

Our observations answer RQ2 by showing that benchmark performance is not a fixed property of processor revision alone: for the tested workload, relative device performance depends on the execution configuration and measurement period. Consequently, device comparisons should be interpreted as workload- and context-specific rather than as stable global processor rankings.

7.2 Aggregate statistics and workload behavior

The common normalization sharpens the comparison between the paired workloads. On IBM hardware, Möbius-ladder with *optimization_level* 3 executions produced lower normalized marked-state gain than the corresponding grid executions on all three devices. This occurred even though the 16-qubit realizations had lower depth and two-qubit-operation counts of a comparable order. The IBM results therefore show that these aggregate circuit statistics alone do not explain the recovered marked-state gain.

The IQM results show a different pattern. The Möbius-ladder workload had larger point estimates of normalized marked-state gain than the grid workload on all three IQM devices at *optimization_level* 3. However, its marked-state probability was distributed unevenly, with state `101` suppressed below its uniform per-state probability in every reported execution. Thus, the primary IQM limitation is not necessarily the absence of total marked-state gain, but the failure to reproduce the structure of the ideal output distribution.

Together, these results show that neither aggregate circuit statistics nor total marked-state probability alone provides a complete characterization of algorithm-level output quality. With respect to RQ1, the paired workloads therefore show that aggregate post-transpilation metrics are informative but not sufficient on their own. Similar or lower depth and comparable two-qubit-operation counts can occur together with weaker normalized marked-state gain, while nontrivial total gain can occur together with a distorted marked-state distribution. Algorithm-level benchmark interpretation therefore requires workload-specific output measures in addition to aggregate circuit summaries.

One possible explanation is that limited connectivity and the larger active-qubit footprint increase routing and accumulated error in ways not reflected by unweighted depth alone. However, the present experiments do not isolate these effects from calibration, readout, decoherence, or other hardware factors. Related effects of connectivity and noise on Grover circuits were reported in (Joshi et al., 2024).

For the 16-qubit workload, the uniform marked-state probability is $\frac{3}{8}$, while the ideal one-iteration probability is $\frac{27}{32}$. The strongest configurations produce amplification above the uniform baseline, but remain substantially below the ideal result. We therefore regard this workload as being a demanding workload for the tested processors rather than as being almost solved.

7.3 IQM results and diagnostic controls

For the 9-qubit grid workload, the IQM marked-state frequencies remained close to the uniform baseline. The five-qubit SAT control, however, produced clear marked-state concentration on Garnet and, under *optimization_level* 3, on Emerald. The two-qubit circuit provided additional evidence that expected-output concentration remained observable at lower circuit complexity. These controls show that weak Lights Out results do not imply the absence of observable Grover behavior, but they do not identify a particular error mechanism.

For the 16-qubit workload, several IQM configurations produced total marked-state frequency above the uniform marked-set baseline. However, the probability was distributed unevenly among the three marked outputs, and state `101` remained below its uniform per-state probability in every reported execution. Total marked-state probability therefore overstates the similarity to the ideal Grover distribution. The compact IQM transpilation statistics did not translate into output close to the ideal result.

The diagnostic controls help interpret the main benchmarks: near-uniform or distorted Lights Out outputs indicate workload-dependent limitations rather than a complete absence of observable amplitude amplification on the same hardware.

7.4 EstimatorV2 resilience settings and runtime

EstimatorV2 was used only for the 16-qubit workload on IBM hardware. Its marked-subspace projector expectation equals the total probability mass of the three marked search-register states. The tested resilience levels are built-in IBM Runtime presets rather than mitigation methods developed in this work.

The parameter *resilience_level* set to 2 produced the highest point estimate on each IBM processor, but also substantially increased the Runtime-reported standard error for `ibm_marrakesh` and `ibm_fez`. The available measurements therefore do not establish a statistically resolved improvement over the lower resilience levels on those devices. The stronger settings also consumed more runtime – each experiment usually took roughly 3 seconds of runtime on IBM devices (for IQM devices, each of our experiments usually took 1 or 2 seconds), but EstimatorV2 with *resilience_level* set to 1 and 2 each experiment took 12 and 15 seconds, respectively. This shows a trade-off between the estimate, its uncertainty, and runtime cost.

We noticed some exceptions in runtime usage – sometimes the same circuit (that is actually run on hardware) with the same number of `shots` can use a noticeably different amount of runtime on quantum hardware. For example, while on a 9-qubit circuit with SamplerV2 with *optimization_level* 2 used 3 seconds of runtime on IBM devices, there was a case when the experiment on `ibm_fez` used 7 seconds of runtime. A similar case was observed on IQM devices. While the experiment for the SAT instance used up one or two seconds of runtime on all three IQM devices, there was one case when the experiment used 4 seconds on IQM Emerald.

7.5 Benchmarking implications and limitations

The results provide two main answers to the research questions. First, aggregate post-transpilation metrics, such as circuit depth and two-qubit-operation count, do not fully characterize algorithm-level benchmark behavior for the tested workloads. Second, observed performance varies with device, compilation, and execution configuration, and calibration period, so processor revision alone does not define a stable circuit-specific performance ordering.

Our results support interpreting quantum-hardware benchmarks for a specific workload and execution context. Aggregate circuit statistics remain useful descriptors, but they should be interpreted together with algorithm-level output measures and information about the device, compilation configuration, mapping, and calibration period. In practical device selection, an initial screening run may identify a favorable device-compiler combination for a particular workload and measurement period, but such a result should not be interpreted as a stable global ranking of processors.

Our conclusions have several limitations. The experimental design was limited by public runtime quotas, unequal provider shot budgets, a restricted number of repeated executions, and measurements performed on different dates. Cross-platform comparison is further complicated by stochastic and calibration-dependent transpilation and different provider software stacks. Finally, the experiments do not independently separate gate, readout, decoherence, crosstalk, mapping, and calibration effects. Therefore, our conclusions describe the observed benchmark behavior and the limits of aggregate circuit summaries; they do not identify the individual hardware causes. Two-qubit-operation counts also aggregate provider-specific native entangling operations and should not be interpreted as equivalent physical error or duration across hardware platforms.

8 Conclusion

We constructed 9- and 16-qubit Grover-search workloads based on Lights Out instances on a 2×2 grid and a six-vertex Möbius ladder and evaluated them on publicly accessible IBM and IQM processors. The paired workloads were used to examine whether aggregate post-transpilation circuit metrics fully describe algorithm-level benchmark behavior and how strongly observed performance depends on device, compilation, and execution configuration, and calibration period. Smaller Grover circuits were additionally used as diagnostic controls when the main workloads approached their uniform baselines.

With respect to RQ1, the experiments show that aggregate post-transpilation metrics are informative but do not fully describe algorithm-level performance for the tested workloads. In several configurations, the two workloads had comparable aggregate depths and two-qubit-operation counts while showing different normalized marked-state gain and distributional quality. Nontrivial total marked-state gain also did not necessarily correspond to an output distribution close to the ideal Grover result.

With respect to RQ2, relative device performance varied across compilation and execution settings and measurement periods. Processor revision alone therefore did not determine circuit-specific benchmark performance. Taken together, these results support a workload-specific and context-aware approach to application-level quantum-hardware benchmarking. Aggregate circuit summaries should be interpreted together with algorithm-level output measures and information about the device, compilation configuration, mapping, and calibration period rather than being used as standalone indicators of benchmark quality.

The 9-qubit workload provides a comparatively accessible benchmark that distinguishes several device/compiler configurations, whereas the 16-qubit workload probes a more demanding regime in which both normalized marked-state gain and marked-state composition remain informative. Future work should extend the benchmark to additional graph instances and hardware platforms, repeated executions across controlled transpiler seeds and calibration periods, controlled physical layouts, and distribution-level quality measures.

Acknowledgments

The work of M. Dimitrijevs and A. Yakaryılmaz was supported by the Latvian Quantum Initiative under the European Union Recovery and Resilience Facility project no. 2.3.1.1.i.0/1/22/I/CFLA/001. M. Palciha received no specific funding for this work. The funding body had no role in the design of the study, data collection and analysis, interpretation of the results, or preparation of the manuscript.

Declaration on Generative AI

Over the course of preparing this work, the authors used *ChatGPT* for language editing, literature-search queries, and preliminary summaries of candidate sources. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

References

AbuGhanem, M. (2025). Characterizing Grover search algorithm on large-scale superconducting quantum computers, *Sci Rep* **15**(1281) (2025). DOI: 10.1038/s41598-024-80188-6.

Akmal Zulfaizal Fadillah, M. H., Idrus, B., Hasan, M. K., and Mohd, S. M. (2021). Impact of various IBM Quantum architectures with different properties on Grover's algorithm, *2021 International Conference on Electrical Engineering and Informatics (ICEEI)*, Kuala Terengganu, Malaysia, 2021, pp. 1-6. DOI: 10.1109/ICEEI52609.2021.9611142.

Alemany Manzanaro, M., Mora, H. (2025). Assessing the Viability of Quantum SAT Solvers: A Comparative Study Using Grover's Algorithm and Quantum Hardware, *Arai, K. (eds) Proceedings of the Future Technologies Conference (FTC) 2025* **4**. FTC 2025. Lecture Notes in Networks and Systems, **1678**. Springer, Cham (2026). DOI: 10.1007/978-3-032-07992-3_20.

Ambainis, A., Repko, I. (2024). Quantum Algorithm for the Domatic Number Problem, *Baltic Journal of Modern Computing* **12**(3), pp. 259-269. DOI: 10.22364/bjmc.2024.12.3.03.

Anderson, M., Feil, T. (1998). Turning Lights Out with Linear Algebra, *Mathematics Magazine* **71**(4), pp. 300-303 (1998).

Aseguinolaza, U., Sobrino, N., Sobrino, G., Jornet-Somoza, J., and Borge, J. (2024). Error estimation in current noisy quantum computers, *Quantum Information Processing* **23**, 181 (2024). DOI: 10.1007/s11128-024-04384-z.

Bennakhi, A., Byrd, G. T., Franzon, P. (2024). Solving the B-SAT Problem Using Quantum Computing: Smaller Is Sometimes Better, *Entropy 2024* **26**(10). DOI: 10.3390/e26100875.

Berman, A., Borer, F., Hungerbühler, N. (2021). Lights Out on graphs. *Math Semesterber* **68**, pp. 237-255 (2021). DOI: 10.1007/s00591-021-00297-5.

Haverly, A., López, S. (2021). Implementation of Grover's Algorithm to Solve the Maximum Clique Problem, *2021 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, Tampa, FL, USA, 2021, pp. 441-446. DOI: 10.1109/ISVLSI51109.2021.00087.

Jipipob, J. U., Chongstitvatana, P. (2025). A Benchmarking Study of Grover's Algorithm for Solving Boolean SAT with Quantum Circuits, *2025 22nd International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, Bangkok, Thailand, 2025, pp. 1-6. DOI: 10.1109/ECTI-CON64996.2025.11100698.

Joshi, M., Mishra, M. K., Karthikeyan, S. (2024). Impact of Noise and Hardware Connectivity on Grover's Quantum Search Algorithm in NISQ Devices, *Intelligent Human Computer Interaction: 16th International Conference 2024, LNCS* **15558**. DOI: 10.1007/978-3-031-88881-6_4.

Kordzanganeh, M., Buchberger, M., Kyriacou, B., Povolotskii, M., Fischer, W., Kurkin, A., Somogyi, W., Sagingalieva, A., Pflitsch, M., and Melnikov, A. (2023). Benchmarking Simulated and Physical Quantum Processing Units Using Quantum and Hybrid Algorithms, *Advanced Quantum Technologies* **6**(8) (2023). DOI: 10.1002/qute.202300043.

Lubinski, T., Coffrin, C., McGeoch, C., Sathe, P., Apanavicius, J., Bernal Neira, D., and Quantum Economic Development Consortium (QED-C) Collaboration (2024). Optimization Applications as Quantum Performance Benchmarks, *ACM Transactions on Quantum Computing* **5**(3), pp. 1-44 (2024). DOI: 10.1145/3678184.

Mills, D., Sivarajah, S., Scholten, T. L., and Duncan, R. (2021). Application-Motivated, Holistic Benchmarking of a Full Quantum Computing Stack, *Quantum* **5**, p. 415 (2021). DOI: 10.22331/q-2021-03-22-415.

Vidwans, A. (2021). A Quantum Variant for Solving Lights Out Problem, *TDS Archive, Medium*, 18 April 2021. `https://medium.com/data-science/a-quantum-variant-for-solving-lights-out-problem-a0b5c364f91e`, accessed 21 January 2026.

Vinod, G. M., Shaji, A. (2021). Finding Solutions to the Integer Case Constraint Satisfiability Problem Using Grover's Algorithm, *IEEE Transactions on Quantum Engineering* **2** (2021). DOI: 10.1109/TQE.2021.3120449.

Received August 2, 2026 , revised October 1, 2026, accepted November 19, 2026