

When can we trust a quantum simulation?

Correlations and the spread of a local disturbance

BITS Pilani, K K Birla Goa Campus | IBM Qiskit+ Fall Fest

Imagine a reversible program running on a graph. Each vertex stores a qubit, and each edge allows a two-qubit operation. Repeating a short update rule produces a probability distribution over bit strings. Predict selected statistics, compare exact and approximate classical algorithms, and test the same program on a quantum processor. Do four-qubit statistics become unreliable before single-qubit averages? Determine whether classical approximation or hardware noise limits the sizes and times you can trust.

Physics translation. The program simulates interacting quantum spins driven away from their initial alignment. We ask how their initial order changes, how correlations develop, and, in an extension, how a local disturbance spreads. These are current research questions in quantum many-body dynamics. A useful result is an honest boundary of reliable computation, even when a classical method performs very well.

Scope. Complete Steps 1–4 and the submission in Step 6. Step 5 is an extension. Use one device geometry and at least one approximate classical method. Each team receives at most **10 minutes of QPU time** in total; Step 4 explains how to plan against it. AI coding tools and existing packages are welcome. Your team must explain the checks and reproduce the results.

1. Fix the graph and the update rule

Choose Boston or Fez (nominally 156 qubits, heavy-hex connectivity), or Phoenix (120 programmable qubits, square connectivity). Check the current backend before choosing a connected patch. An edge means an available physical coupling. Save the exact vertices and edges. Use open boundaries and retain all selected couplings, including those that close cycles. Start with 8–12 qubits, then 16–20, then attempt 32–64. The largest connected region of the device is an optional further patch: it costs no extra QPU time (Step 4), but its classical benchmark is the hard part.

Physics translation. The Hamiltonian specifies the update of n qubits. With X_i the bit-flip operator, $Z_i|0\rangle = |0\rangle$, $Z_i|1\rangle = -|1\rangle$, and E the saved edge set, use

$$H = H_b + H_f, \quad H_b = -J \sum_{(i,j) \in E} Z_i Z_j, \quad H_f = - \sum_i (h_x X_i + h_z Z_i).$$

Set $J = 1$, $\hbar = 1$, and start in $|0\rangle^{\otimes n}$. The positive coupling favours aligned neighbours. The X field changes spins, while the negative Z field favours the opposite alignment. Use the following starting parameters, fixed across sizes within a geometry.

Geometry	h_x/J	h_z/J	Initial $J\delta t$
Heavy-hex	1.25	−0.60	0.20
Square	2.00	−0.90	0.10

These are two benchmark families. Their comparison does not isolate geometry because the fields differ. A geometry-only extension must use identical fields.

Use N steps of duration δt to reach $t = N\delta t$. The second-order split is $U_N = S(\delta t)^N$, where

$$S(\delta t) = e^{-iH_f\delta t/2} e^{-iH_b\delta t} e^{-iH_f\delta t/2}.$$

Implement each field factor exactly as a single-qubit unitary. Adjacent field halves merge. In Qiskit's convention $R_{ZZ}(\theta) = e^{-i\theta ZZ/2}$, each edge needs $R_{ZZ}(-2J\delta t)$. A field interval τ uses rotation angle $-2\sqrt{h_x^2 + h_z^2}\tau$ about $(h_x, 0, h_z)/\sqrt{h_x^2 + h_z^2}$. Check both signs. Schedule disjoint edges together. Ideal bond depth is at most 3 layers on heavy-hex or 4 on square per step. Native compilation can increase this.

Output. A graph drawing, edge list, parameter file, and one verified circuit. Use this same circuit as the target for every circuit benchmark.

2. Define the statistics and verify small systems

Measure every qubit. For each measured bit b_i , write $z_i = (-1)^{b_i}$. Averages below mean averages over runs, or ideal expectations. Before examining hardware data, select all single-site means, edge-pair correlations, and at least three four-site clusters.

Physics translation. A single-site mean measures local spin alignment. Pair correlations reveal shared fluctuations. A four-point cumulant measures information not contained in the means and pair covariances. It is not, by itself, a test of quantum entanglement.

$$m_i = \langle z_i \rangle, \quad M = \frac{1}{n} \sum_i m_i, \quad C_{ij} = \langle z_i z_j \rangle - m_i m_j, \\ G_{ijkl} = \langle z_i z_j z_k z_l \rangle, \quad \kappa_{ijkl} = \langle \delta z_i \delta z_j \delta z_k \delta z_l \rangle - C_{ij} C_{kl} - C_{ik} C_{jl} - C_{il} C_{jk},$$

where $\delta z_i = z_i - m_i$ and the four sites are distinct. Use a centre with three neighbours on heavy-hex, or a square on Phoenix. Report fewer clusters if the smallest patch requires it. All quantities use the same bit strings. The four-point signal may be small.

1. Run ideal Qiskit Aer statevector simulations at $Jt = 0, 0.4, 1, 2, 3, 4, 5$, with $N = t/\delta t$. At $N = 0$, verify $m_i = G_{ijkl} = 1$ and $C_{ij} = \kappa_{ijkl} = 0$. Check qubit-to-bit ordering and that $U_N^\dagger U_N$ returns the initial state.
2. On 8–12 qubits, independently compute $e^{-iHt}|0\rangle^{\otimes n}$ with a sparse matrix exponential applied to a vector. Compare with the circuit. On the largest affordable exact patch, halve δt and compare at the *same physical times*, up to $Jt = 5$ if affordable. Aim for maximum absolute observable difference below 0.02, rather than relative error near zero. Refine further where practical. Small-patch agreement is evidence, not an error bound for all sizes.
3. Plot $M(t)$ and repeat a small-system run with h_z reversed. Measure the contrast without requiring a prescribed decay or plateau. An optional $h_z = 0$ chain is a soluble control, not a test of two-dimensional difficulty.
4. *Optional.* On a 12-qubit patch with no mirror symmetry (or a chain with h_z on one end site scaled by 1.2), compute the eigenvalues $e^{-i\varphi_k}$ of the single step $S(\delta t)$, sort the phases, form consecutive gaps s_k , and average $r_k = \min(s_k, s_{k+1})/\max(s_k, s_{k+1})$. Expect $\langle r \rangle \approx 0.53$ at the table parameters and ≈ 0.39 for the $h_z = 0$ chain. This is linear algebra on a 4096×4096 matrix.

Physics translation. Item 4 is the standard test that the dynamics is not secretly reducible to free particles: repelling energy levels give 0.53, independent levels give 0.39. It certifies that the parameters sit in the hard regime rather than a perturbative one.

Freeze the time step and the observable list after these checks. If smaller steps force a shorter hardware time range, report the new range. Distinguish errors in approximating e^{-iHt} from errors in simulating the fixed circuit.

Output. Exact curves for M , selected C , G , and κ , plus a table of time-step errors.

3. Measure the accuracy and cost of a classical method

Choose one route below. Compare with the exact circuit on small patches, then attempt the larger patches. Retain the original graph and every interaction.

- **Pauli propagation, including monoprop.** Propagate the measured operator backwards as a weighted sum of products of the single-qubit matrices I, X, Y, Z . Truncate this sum to control memory. Test three cutoffs or term limits and record the retained term count.
- **Tensor networks.** Compress the state into linked arrays. Use gate-based time-evolving block decimation (TEBD), or the time-dependent variational principle (TDVP) for Hamiltonian evolution. Test at least three bond dimensions, the sizes of the connecting indices. TDVP comparisons must separate circuit time-step error from compression error. Keep long-range couplings introduced by a one-dimensional ordering.

- **Belief propagation for quantum dynamics.** Pass approximate tensor-network messages between regions. Vary message size and, where supported, cluster size. Message convergence does not eliminate loop errors. Classical Ising sampling is not this calculation.

Output. Accuracy versus runtime and peak memory at each size, with CPU/GPU specifications. Beyond exact validation, cutoff stability is an empirical check. Report failures and resource limits. A timeout or disagreement with noisy hardware does not establish quantum advantage.

4. Run a controlled hardware comparison

Use one small patch with exact answers and one larger patch, preferably 32–64 qubits. A third, optional patch may be the largest connected region of the device. Justify a smaller second patch if calibration or resources require it.

Physics translation. Error mitigation uses calibration and repeated measurements to reduce the effects of gate and measurement errors. It can increase statistical uncertainty, so test it against known answers.

Before submission. Retrieve the backend’s supported gates, coupling map, durations, repetition delay, and calibration. Check support for native tunable R_{ZZ} gates, called fractional gates. Negative angles may need equivalent positive-angle circuits with X conjugations. A single-qubit unitary may use several pulses. If R_{ZZ} decomposes into two CZ gates, count both. Save compiled circuits and report two-qubit counts, depths, and durations.

Depth. Aim for compiled two-qubit depth at most 100, with a ceiling of 300 for *every executed circuit*, including mitigation and reversal. On heavy-hex with $J\delta t = 0.20$, $Jt = 3$ is 45 ideal layers and $Jt = 5$ is 75. On the square lattice with $J\delta t = 0.10$, $Jt = 2$ is 80 layers and $Jt = 3$ is 120, so Phoenix teams treat $Jt = 3$ as optional and run zero-noise extrapolation only at a time whose compiled depth is at most 100, so that three integer folds stay within the ceiling. Depth alone does not ensure accuracy. Run a noisy Aer check on the small patch with a stated noise model.

QPU allocation. Each team has at most 10 minutes of QPU time in total, measured as the usage the service reports for its jobs. Usage is set by shots, the repetition delay between shots (about $250\ \mu\text{s}$ by default on current backends; read it from the backend), readout, and a per-job overhead. At these depths the circuit itself adds only $10\text{--}15\ \mu\text{s}$ per shot, and the qubit count does not enter at all: a 156-qubit patch costs the same minutes as a 12-qubit one. Until measured, plan with $0.5\ \text{ms}$ per shot, that is 2,000 shots per second, or 1.2 million shots in 10 minutes. Keep planned usage at or below 7 minutes and hold the remainder in reserve. Your first job (readout calibration plus $Jt = 0$ and 0.4 on the small patch) doubles as a pilot: read its reported usage, recompute the rate, and rescale the plan before submitting anything else. Submit few large batch jobs, about four per team, rather than many small ones, and never leave an interactive session open. Access ends when the allocation is used.

Shot budget. Use $Jt = 0, 0.4, 1, 2, 3$ on heavy-hex and $Jt = 0, 0.4, 1, 2$ on the square lattice (optional $Jt = 3$), with $N = t/\delta t$, if the Step 2 checks support these ranges. Use 16,384 shots per circuit; the four-point cumulant needs them. Split a circuit’s shots across randomized implementations if used.

Block	Circuits $\times$ shots	Shots	Time at 2,000/s
Readout calibration, two patches	$2 \times 16,384$	32,768	16 s
Forward, two patches, five times (four on Phoenix)	$10 \times 16,384$	163,840	82 s
<i>Required subtotal</i>		196,608	$\approx 100\ \text{s}$
One further correction: reference circuits at three times on two patches, or two extra noise levels at one time on two patches	$\leq 6 \times 16,384$	98,304	49 s
Step 5 extension, one patch, three times, target and reference	$6 \times 16,384$	98,304	49 s
Optional full-device patch, forward and calibration	$6 \times 16,384$	98,304	49 s
<i>Planned total, all blocks</i>		491,520	$\approx 4\ \text{min}$
Reserve: per-job overhead, re-runs after calibration drift, doubled shots at the two latest times			$\geq 3\ \text{min}$
Allocation			10 min

Estimate every job’s usage from shots, repetition delay, and durations before submission, and record the reported usage afterwards. Shot count alone does not determine minutes, but at these depths it dominates. After mentor review, submit early enough for queues.

Required post-processing. Calibrate measurement errors and correct the one- through four-site

outcome probabilities, or use a validated method such as M3. Combining single-qubit calibrations assumes independent readout errors. Check this on the small patch. Compute C and κ from corrected moments and retain raw estimates.

One further correction. The budget includes one of the following. Test it on the small patch, choose settings using some known times, and validate on unused times before applying it at larger sizes.

- **Reference rescaling.** Observable-dependent renormalization (ODR) uses a known reference to estimate attenuation. Match the observable and relevant errors. A Clifford reference uses gates that preserve Pauli strings, including at its boundaries. Changing angles changes noise, so validate the correction. The Step 5 reference does not automatically calibrate forward circuits.
- **Zero-noise extrapolation.** At one time, insert inverse-gate pairs to increase noise and fit towards zero noise using at least three levels. Verify that compilation preserves the pairs. Report the fit and uncertainty. Adding only two-qubit pairs does not scale every error source equally.

Check IBM option compatibility. Current documentation excludes automatic two-qubit Pauli twirling with fractional gates. Readout correction is the required minimum.

Estimate 95% intervals by bootstrapping, meaning resampling shot records with replacement and recomputing all moments together. Include calibration uncertainty and execution blocks. Check cumulant bias on the exact case. Do not clip corrected values or hide unresolved signals.

Output. Raw and corrected curves with intervals, small-patch errors, and resource usage. Separate statistical uncertainty from time-step and classical approximation errors.

5. Optional extension: spread a local disturbance

Run forward, flip an interior qubit i with X_i , run the exact inverse gate sequence, and measure. Repeat without the flip as a reference. Preserve both arms during compilation and inspect the result. The compiler must not cancel the reference to an empty circuit.

Physics translation. Perfect reversal erases the evolution. The flip prevents this locally, and interactions can spread its effect. This measures sensitivity in the chosen state, without establishing thermalization or chaos.

Define $W_i(t) = U_N^\dagger X_i U_N$ and write $\langle \cdot \rangle_0$ for the expectation in $|0\rangle^{\otimes n}$. The measured return polarization and the probability of a flipped output bit are

$$E_{ij}(t) = \langle W_i(t) Z_j W_i(t) \rangle_0 = \text{Re} \langle W_i(t) Z_j W_i(t) Z_j \rangle_0, \quad D_{ij}(t) = \frac{1 - E_{ij}(t)}{2}.$$

The equality uses $Z_j |0\rangle^{\otimes n} = |0\rangle^{\otimes n}$. The four-operator expression is an out-of-time-ordered correlator in this state, rather than an average over inputs. Initially, $D_{ii} = 1$ and $D_{ij} = 0$ for $j \neq i$. The ideal unperturbed reference has $E_j^{\text{ref}} = 1$.

Try $Jt = 0.4, 0.8, 1.2$ on one patch, with $N = t/\delta t$. Target and reference at 16,384 shots each cost 98,304 shots, about 50 s at the planning rate; this block is in the Step 4 table. A second patch doubles it and must fit in the reserve. Reversal roughly doubles two-qubit depth: 36 ideal layers at $Jt = 1.2$ on heavy-hex, 96 on the square lattice. Acquire both circuits close in time. After readout correction, the ratio E_{ij}/E_j^{ref} assumes matching attenuation. Validate it on small systems and propagate both uncertainties. Reject it if validation fails or the reference is less than five standard errors from zero. Do not clip unstable ratios.

Output. Map $D_{ij}(t)$ against time and shortest-path distance from i , with raw, reference, and corrected results where valid. Benchmark the same reversal circuit classically. Measure the spread without assuming it reaches the whole device.

6. Submit a reproducible result

Submit one archive named `team_name.zip`. The archive must contain the following.

1. **src/, README.md, and a pinned environment file.** Supply code and a command to reproduce figures and tables from saved data without QPU submission. Identify reused packages and AI assistance. Include sign, bit-order, $t = 0$, and reversal checks.
2. **results.csv and resources.csv.** In results, give graph ID, n , parameters, δt , N , t , observable, site indices, method, approximation setting, estimate, interval endpoints, and validation status. In resources, give runtime, peak memory, CPU/GPU details, term count or bond/message dimension, compiled gate count/depth, shots, and QPU usage per job as reported, with the estimate made before submission. Use one row per measured quantity or run.
3. **raw/ and metadata.json.** Save bit strings or lossless counts by execution block, calibrations, references, edge lists, compiled circuits, qubit layout, backend, job IDs, timestamps, versions, seeds, and mitigation settings. Exclude credentials. During an outage, organizers may supply clearly labelled hardware data.