Quantum Galton Boards: My Understanding and Project Plan

I've been studying how a quantum Galton board (QGB) can reproduce and generalize the classical Galton board's statistics, and how that idea extends to a broader "universal statistical simulator" through modular circuit design and bias control. The central intuition I'm carrying into my project is simple: a classical Galton board explores an exponential number of trajectories as balls cascade through pegs, while a quantum circuit can encode those 2^n paths in superposition with shallow, composable building blocks—making sampling tasks both intuitive and scalable. The Universal Statistical Simulator (USS) paper by Carney and Varcoe reinforces this mental model, showing that a QGB can be built with a small set of gate types, reduced depth compared to prior approaches, and a route to universality by "removing pegs" and adjusting left–right ratios. That blend of conceptual clarity and engineering pragmatism is exactly what I want to implement and demonstrate in my submission.[1][2][3]

At a high level, my working picture is each "peg" is a quantum module that splits amplitude left vs right; stacking these modules level-by-level yields an n-level QGB, with measurement producing a histogram over bins analogous to the classical board's binomial counts. The novelty is not that the output is Gaussian—classically we know it tends to a normal via De Moivre–Laplace—but that the circuit can evaluate all trajectories coherently and do so with carefully managed depth and resources. Carney–Varcoe emphasize that their QGB computes 2^n trajectories with O(n^2) resources using only three gate types and ancilla, and—crucially—that the construction can be generalized into a "universal statistical simulator" by removing pegs (structural pruning) and biasing each peg (amplitude steering). This is a compelling path to turn the Galton board from a single-purpose generator into a family of statistical samplers.[2][1]

My plan follows that architecture in three stages. First, I will implement the symmetric QGB that reproduces the binomial/Gaussian distribution across any number of layers n. Conceptually, I'll use one qubit per "decision" (or adopt the USS peg module with ancilla if I want lower depth at scale), prepare equal superpositions (Hadamard per level in the simple variant), and measure to obtain a distribution over bitstrings, mapped to bins by counting "1"s. Validation will include: overlaying the observed histogram with the binomial PMF, plotting residuals per bin, and reporting distance metrics—Jensen–Shannon (primary), Total Variation Distance, Chi-squared, and max error. I'll run enough shots to stabilize the estimates and will keep circuits shallow through transpilation so the numbers reflect circuit quality rather than sampling variance.[3][4][1][2]

Second, I will demonstrate two non-Gaussian targets on a noiseless, all-to-all simulator to separate design intent from hardware constraints. For an exponential distribution, the idea is to replace uniform splits with biased rotations whose angles are designed (or lightly calibrated) to match a normalized exp(−λk) profile over bins 0..n; this maps cleanly to the USS "alter the left–right ratio" instruction for pegs, but realized with low-depth single-qubit rotations plus minimal entanglement. For the Hadamard quantum walk, I'll adopt the coined walk model: apply a Hadamard coin on a coin qubit followed by a conditional shift on the position register, repeated for T steps, then measure the position distribution. This walk exhibits ballistic spread and interference patterns that are qualitatively different from the Gaussian of classical random walks; it's a natural "second target" to show the framework isn't locked to binomial behavior.[5][6][7][8]

Third, I will introduce a realistic noise model (device-like gate and readout errors) and optimize all three cases—Gaussian, exponential, and Hadamard walk—under noise. Here the USS message about depth matters: shallower circuits accumulate less error, so I'll keep the construction minimal (or swap to the USS peg module/ancilla if it reduces depth for larger n). I'll increase shots to reduce statistical error, use advanced transpilation to trim redundant gates, and compare noisy vs noiseless metrics to quantify degradation. The goal is not just a single accuracy number but an accuracy vs n curve under noise, to show the practical ceiling on board depth with current noise levels and to demonstrate thoughtful engineering choices that stabilize fidelity.[9][1][2][3]

On complexity and scaling, my narrative will be transparent: classical path enumeration is $O(2^n)$, while the quantum circuit cost grows roughly linearly or low-polynomial in n for the sampler's core, compressing trajectory generation into shallow circuits. I'll plot simple speedup curves (e.g., comparing 2^n paths against $O(n)$ gate counts) and include a log-scale resource plot to make the growth rates visible. These are communication devices to support intuition; the formal claim I'll stick to is the USS result that their QGB creates 2^n trajectories with $O(n^2)$ resources and lower depth than earlier designs—a strong engineering statement that's relevant for NISQ feasibility.[1][2][3]

Evaluation will be rigorous but readable. Every target distribution will come with: (1) overlay plots (target vs observed), (2) residuals per bin, and (3) a metrics table with JS/TVD/χ^2/max error. Where practical, I'll use bootstrap resampling over the measured counts to attach uncertainty bands to JS/TVD, so results are not just point estimates. For the Hadamard walk, I'll cross-check qualitative features (ballistic spread, interference peaks, known asymmetries depending on initial coin state) against standard references, and where available, I'll compare to closed-form distributions derived for 1D walks to ensure fidelity of the implementation and the sampling pipeline.[6][7][5]

I also like the USS framing that removing pegs and adjusting biases "universally" spans a broad class of statistical simulators, which I interpret as a practical recipe: the same modular board can be sparsified and retuned to synthesize families of distributions, not just the Gaussian. That means the project is not a one-off demo; it's a platform. In a short project window, I'll keep the scope disciplined—symmetric Gaussian for baseline, exponential for monotone decay, and a Hadamard walk for interference-driven non-Gaussian structure—but the code will be written to accept per-peg bias schedules, optional peg removal, and coin/shift controls, so it's easy to extend. This is faithful to the USS spirit and keeps the door open for additional targets later (e.g., skewed or multi-modal distributions) without redesigning everything from scratch.[2][3][1]

Finally, I'll package the work like a product: a two-page narrative (this summary), clean figures, a reproducible notebook or script with seeded runs, and a JSON of metrics for each experiment. The deliverables will emphasize clarity over flash: the Gaussian baseline will be the anchor that proves correctness; the exponential and Hadamard walk will demonstrate versatility; the noise-aware runs will show pragmatism for NISQ realities; and the scaling visuals will communicate the "why quantum" story simply and honestly. In short, the quantum Galton board is my intuitive bridge between classical combinatorics and quantum superposition, and the USS paper gives me the blueprint to make that bridge both shallow and universal in practice.[3][1][2]

1. https://arxiv.org/abs/2202.01735
2. https://arxiv.org/pdf/2202.01735.pdf
3. https://www.thewiser.org/quantum-walks-monte-carlo
4. https://en.wikipedia.org/wiki/Galton_board
5. https://en.wikipedia.org/wiki/Quantum_walk
6. https://arxiv.org/abs/quant-ph/0010117
7. https://arxiv.org/abs/2308.05213
8. https://quantumai.google/cirq/experiments/quantum_walks
9. https://link.aps.org/doi/10.1103/PhysRevA.102.022223