QGB Summary

Prawjal Neupane **Suramya Raj Angdembay**

August 11, 2025

1 Introduction

The *Universal Statistical Simulator* paper starts with the famous example of the quantum Fourier transform, which is the first demonstration of a useful algorithm where a quantum computer is exponentially faster than a classical computer. However, this comes with the caveats that current classical systems could be improved upon and the complexity of the classical calculations not being fully understood. So, the authors introduce a quantum Galton board (QGB) simulator, which is exponentially faster than its classical counterpart and can be intuitively understood without requiring an understanding of computational complexity.

A Galton board (GB) can be thought of as an arrangement of pegs in n levels, with the number of pegs in each level controlled by the triangular numbers ($t_n = \frac{n^2+n}{2}$). Each peg can be thought of as a coin flip, which can be made biased. Starting with an object just above $n = 1$, 1-peg, each peg either pushes the object left or right into another peg at each of these positions. If we are at the n-th level of an n-level GB, we place the object in buckets/baskets (make a reading) instead of sending it to another peg. Subsequently, for an n-level board, we will observe the distribution

$$\frac{1}{2^n}\binom{n}{k},$$

which, for large n, via the central limit theorem, approaches the normal distribution. This binomial form follows directly from the binomial theorem, and its convergence to the normal distribution for large n can be formally shown using the De Moivre–Laplace theorem.

The paper replicated the output of a classical GB in a quantum circuit that creates a superposition of all possible trajectories. By sampling the measurement multiple times, the QGB statistically matches the distribution of a GB while also allowing individual trajectories to be tailored through quantum superposition and operations on quantum gates to control bias, something which is not possible in a GB without physically modifying the pegs. The quantum circuit to perform the GB was achieved through a combination of three Clifford operations and ancilla qubits. To perform an n-level QGB, we require $n(2n-1)$ gates and $2n$ qubits.

Beyond reproducing the classical GB statistics, the QGB benefits from the intrinsic randomness of quantum measurement, avoiding the subtle biases inherent in pseudo-random number generation on classical machines. In addition, it implements a quantum walk, in which all possible left–right sequences are explored simultaneously in superposition, producing interference patterns not seen in classical random walks. This quantum parallelism enables the simulation of 2^n possible paths using only $O(n^2)$ resources, underpinning the exponential speedup over a direct classical simulation.

2 Core Structure of a QGB

The QGB is constructed from modular units called quantum pegs. Each peg simulates a probabilistic left/right decision using a controlled-swap (CSWAP) gate, driven by a control qubit prepared in superposition. This control qubit enables quantum parallelism by allowing the ball's amplitude to branch in both directions simultaneously.

Rather than assigning a dedicated control qubit to each peg, the QGB architecture reuses a single shared control qubit. After each layer of pegs, the control is rebalanced using a CNOT and reset, ensuring that it remains unbiased for the next layer. This design minimizes qubit usage and preserves coherence across multiple layers.

The number of pegs in an n-level board follows the triangular number sequence:

$$T_n = \frac{n(n+1)}{2}.$$

Each level adds one more peg than the previous, mirroring the layered structure of the classical Galton board.

The required qubit count scales linearly with depth. Specifically, an n-level QGB uses:

$$2(n+1) \text{ qubits}$$

comprising:

- 1 control qubit
- n peg qubits (intermediate qubits between layers)
- $n+1$ bucket qubits (final measurement layer)

Qubits are typically arranged in an alternating pattern:

$$\begin{aligned} q[0] &— \text{control} \\ q[1] &— \text{bucket0} \\ q[2] &— \text{peg} \\ q[3] &— \text{bucket1} \\ &\vdots \\ q[2n+1] &— \text{bucket}n \end{aligned}$$

The ball is initialized in the center peg (e.g., $q[n+1]$), and through successive levels of controlled entangling operations, it spreads its amplitude across all buckets. This structure ensures a compact, scalable, and interference-sensitive quantum simulation of the classical GB.

Biasing and Fine-Grained Control. In a biased QGB, the Hadamard gate on the control qubit can be replaced by a rotation about the x-axis, $R_x(\theta)$, to adjust the left–right branching probability:

$$R_x(\theta) = \begin{pmatrix} \cos\frac{\theta}{2} & -i\sin\frac{\theta}{2} \\ -i\sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix}.$$

By tuning θ, skewed distributions can be generated (e.g., $\theta = \frac{2\pi}{3}$ yields a 75% / 25% split). Bias can be applied uniformly across all pegs or varied per peg for fine-grained control, enabling simulation of more complex statistical behaviors without physical changes to the board.

One approach to fine-tuning is to let θ depend on the depth of the current peg in the board. If the current peg is at *level* in a board of n total levels, we define the normalized depth as

$$\text{depth} = \frac{\text{level}}{n}.$$

A possible choice for the bias angle is then

$$\theta = \arcsin\left(\sqrt{\frac{1}{\lambda \cdot n} \cdot \text{depth}}\right),$$

where λ is a tunable scaling parameter. This allows the bias to vary systematically with board depth, providing gradual adjustment of branching probabilities across layers.

3 Circuit Construction and Scaling

Each peg module uses three qubits—two targets for output paths and a shared control qubit—and is built from controlled-swap (Fredkin) gates, CNOTs, and resets. The control qubit is placed in superposition (Hadamard) to branch amplitudes, then rebalanced for reuse at each layer via CNOT and reset, ensuring statistical independence across layers.

An n-level QGB requires $\frac{n(n+1)}{2}$ such modules, so the number of operations scales as $O(n^2)$ while circuit depth grows linearly. The reused control qubit greatly reduces qubit overhead, a key advantage for NISQ-era hardware. Measuring only the bucket qubits yields a one-hot encoding whose distribution approximates the classical binomial law and converges to normality as n increases.

References

[1] M. Carney and B. Varcoe, "Universal Statistical Simulator," *arXiv preprint*, 2022. Available: `https://arxiv.org/abs/2202.01735`.