QGB

First of all, let's understand what a QGB (Quantum Galton Board) is: QGB is essentially a quantum version of the classical Galton board, a device that represents probability distributions. Here, instead of a ball, unlike in the classical version, we have particles or quantum particles, which in computation are qubits. Inside the board. The ball will bounce left or right in a classical version; on the other hand, when we talk about the quantum version, let's imagine a ball that is moving right and left simultaneously, like a ghost ball, due to superposition. At later stages, these balls split further and create interference; their paths overlap, and depending on the phase, either their amplitudes add up (constructive interference) or cancel out (destructive interference), representing a quantum walk.

The logic represented in a paper I read used discrete-time coined quantum walks. Where the H gate is applied to the coin qubit to create a superposition, which represents the movement of a ball from left to right simultaneously. Then, a conditional shift based on the value of the qubit and interference effects leads to different probability distributions. The final position of the walker will be decided only after the measurement. QGB's output depends on both phase and probability amplitudes.

Based on the notions presented in a paper, I started with a small number of layers, I ran the provided QASM code in a paper, and then I started implementing it for n layers of QGB. The algorithm dynamically generates the shift operation for n steps, ensuring that the walker's position register possesses 2n + 1 states. While running the algorithm on a noiseless simulator, a Gaussian distribution is formed where the interference was suppressed, and a balanced Hadamard was used.

To study the versatility of the QGB, we need to alter the coin and phase operations to get the non-Gaussian distribution. Replacing the Hadamard coin with the layer-dependent biasing will bias the walk, which will direct the probability to one side/direction, and we can achieve an exponential distribution using a bias angle chosen carefully. Using the Hellinger distance metric, the two distributions were compared, verifying the successful impact of the probability profile.

I used the FakeAlgiers, IBM backend’s noise model, and then optimized the circuit using transpilation and transpilation passes. Then calculated the Hellinger distance and fidelity between the ideal vs. noisy simulation. For the Hadamard walk, the Hellinger distance between the ideal and noisy result after the optimization was 0.03, which shows a moderate difference in distribution shape, whereas the fidelity was 0.99, showing very high similarity between ideal and noisy simulation outcomes.

Let's see what the key observations are: Fine-grain biasing shapes the distribution output precisely while maintaining coherence. The Hadamard walk maintained the fidelity under realistic noise, demonstrating robustness to modern quantum noise. Hellinger distance is sensitive, as increasing the layers increases the complexity of the interference pattern, which needs to be optimized using proper transpiler passes, error mitigation techniques, etc.

The Quantum Galton Board combines classical statistical physics and quantum processing to provide a simple yet effective setting to study quantum walks, biasing, and noise robustness. This project shows scalable circuit building with arbitrary layers. The distribution is shaped utilizing fine-grained coin bias. Noise tolerance is investigated using real-device error models. These implementations help us understand quantum interference and stochastic processes while also serving as benchmarks for quantum computing power. While classical Galton boards produce Gaussian statistics by default, quantum versions enable designed probability landscapes via superposition, interference, and controlled biasing.

Future extensions might include enhanced results with IBM's error mitigation and AI transpiler passes, followed by multi-coin entangled QGBs for richer dynamics, versatile coin tuning based on real-time noise feedback, and experimental runs on NISQ hardware to validate large-layer performance.