Quantum Galton Boards: Implementation and Simulation

August 11, 2025

Abstract

We present a compact implementation and evaluation of *quantum* Galton boards, a quantum analogue of the classical Galton (bean) machine. We describe circuit constructions that realize classical-like binomial spreads, an exponential bias variant, and a Hadamard-driven quantum walk. Results from Qiskit's Aer simulator are shown, and practical notes on noise modeling and reproducibility are provided.

1 Introduction

The Galton board demonstrates how independent binary events produce a binomial (and asymptotically Gaussian) distribution. Recasting this process in the quantum domain lets a single quantum state traverse superposed paths, producing distributions shaped by interference. Quantum Galton boards can be useful for exploring quantum walks, sampling strategies, and Monte Carlo primitives on near-term devices.

2 Method

2.1 Encoding and circuit layout

Our implementation encodes path positions in a contiguous register of qubits; a single control qubit is reused across layers to perform conditional swaps that propagate amplitude between adjacent position qubits. Each layer corresponds to a decision point (left/right). The high-level steps are:

1. Initialize position register with a single excited qubit at the middle bin.
2. For each layer: reset and prepare the control qubit, create a superposition for the split, perform controlled-swap operations to move amplitude between adjacent positions, and optionally apply coin rotations or phase gates.
3. Measure the position register to obtain the histogram.

A short pseudocode for one layer:

```
reset(ctrl); H(ctrl)
for dest in active_positions:
  CSWAP(ctrl, dest, dest+1)
  CX(dest+1, ctrl)
if variant=="exponential": RY(...)
if variant=="hadamard": H(ctrl)
```

2.2 Variants

Gaussian-like (Hadamard coin): use H on the control each layer.
Exponential: apply a controlled RY with angle that grows with layer index to bias transitions.
Hadamard quantum walk: additional H (or phase) on the control to introduce interference.

3 Experiments and Results

We run each circuit on Qiskit's Aer simulator with a fixed random seed to ensure reproducibility. Histograms for three variants (Gaussian-like, exponential, Hadamard walk) are produced with 5000 shots each. Figures below are produced by running the provided Python script and saving PNGs.

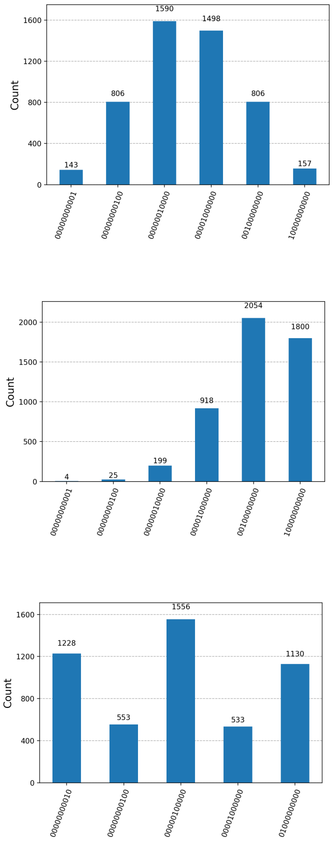

Figure 1: Simulated distributions for $n_{\text{layers}} = 6$ (top: Gaussian-like, middle: exponential bias, bottom: Hadamard walk).