Summary of Universal Statistical Simulator (Quantum Galton Board) by Mark Carney, Ben Varcoe

1. Introduction

The **Quantum Galton Board (QGB)** is presented as an intuitive quantum computing algorithm that simulates the classical Galton board while demonstrating **exponential speedup** over classical computation. Unlike abstract examples like the Quantum Fourier Transform, this approach is easy to understand without deep complexity theory.
The QGB can be adapted into a **Universal Statistical Simulator** by modifying peg arrangements and left–right bias, enabling applications such as:

- Complex systems simulation
- Graph random walks
- Machine learning
- Stock price modeling
- Cryptography
- Sampling and search problems

2. Classical Galton Board

A classical Galton board drops balls through pegs, producing a binomial distribution (approximating a normal distribution for large n).
The formula for an **n-level** board with equal left/right probability ($p = q = 0.5$) is:

P(k)=12n(nk)P(k) = \frac{1}{2^n} \binom{n}{k}

3. Quantum Galton Board Design

3.1 Quantum Peg

- Mimics a physical peg by using **Hadamard (H)**, **Controlled-SWAP (Fredkin)**, and **CNOT** gates.
- A "ball" is represented by a qubit in the $|1\rangle$ state.
- Superposition enables simultaneous simulation of all trajectories.

3.2 Multi-Peg QGB

- Example: **3-peg (2-level)** circuit with minimal depth.
- Uses **reset** operations to reuse the control qubit between layers.

3.3 Scaling

- Each peg needs ≤ 4 gates.
- Total gates for n levels:

$$\text{Gate count} \le 2n^2 + 5n + 2$$

3.4 Features

- Requires **n ancilla qubits** for n output bits.
- Output encoding uses “one-hot” format (post-processing required).
- Circuit depth is less than half that of previous QGB methods, reducing noise.

4. Experimental Results

4.1 Remote Simulations

- Implemented **4-level QGB** on IBM-QX simulator → output matched expected normal distribution.

4.2 Real Hardware (Single Peg)

- On IBM-Manila, transpilation inflated 5 gates → 64 gates, introducing noise.
- Desired states accounted for ~54% of results.

4.3 Local Simulation with Rescaling

- Re-mapped one-hot outputs to integer values.
- Aggregated outputs produced a normal distribution after rescaling.

5. Biased Quantum Galton Board (B-QGB)

5.1 Biased Quantum Peg

- Replace H gate with **Rx(θ)** rotation to skew left/right probabilities.
- Allows control over final distribution shape.

5.2 Example

- $\theta = 2\pi/3 \rightarrow 75\%$ probability upper path, 25% lower path.

5.3 Gate Count

- 5 gates per peg (RESET, Rx, 2×CSWAP, optional CNOT).
- Max gates for n levels:

3(n2+n)+n+23(n^2 + n) + n + 2

6. Experimental Results for B-QGB

- Real hardware (ibmq-manila): biased outputs visible, but noise still significant.
- Local simulations confirmed skewed distributions (mean ~2.66 for $\theta = 2\pi/3$).

7. Fine-Grained Bias Control

7.1 Per-Peg Bias

- Each peg can have its own Rx(θi) value.
- Requires corrective CNOT and RESET gates between pegs.

7.2 Gate Count

- Approx:

3n2+3n+13n^2 + 3n + 1

- Local simulations verified expected output distributions.

8. Conclusions

- The QGB offers an **efficient**, **intuitive**, and **low-depth** approach to generating normal and biased statistical distributions on quantum hardware.
- Advantages:
    - Minimal circuit depth → reduced noise sensitivity
    - Modular "quantum peg" design
    - Flexible bias control
- Limitations:
    - Requires many ancilla qubits
    - Fredkin gates not natively supported → transpilation inflates gate count, increasing noise
    - Still constrained by NISQ-era device errors
- Potential extensions:
    - Native Fredkin gates or error correction
    - Integration into statistical sampling and random number generation pipelines