Youngstown State University

DEXA CONFERENCE 2025

Exploring Quantum Bootstrap Sampling for AQP Error Assessment

Dr. Feng George Yu and Raya Jahan
Computer Science and Information Systems
Youngstown State University
Youngstown, OH USA

Introduction

Background

- Approximate Query Processing **(AQP)** uses sampling method to execute complex big data queries quickly but must assess the error in those approximations.
- Bootstrap sampling is a robust yet resource-intensive error estimation method.
- This study explores how **Quantum Computing** can accelerate bootstrap sampling for AQP.

Objective

- Design a Quantum Bootstrap Sampling (**QBS**) framework for AQP error assessment that uses Superposition, QRAM, Quantum Counters, and a Ripple-carry adder.
- Leverage quantum computing to:

    1) Generate bootstrap resamples more efficiently.

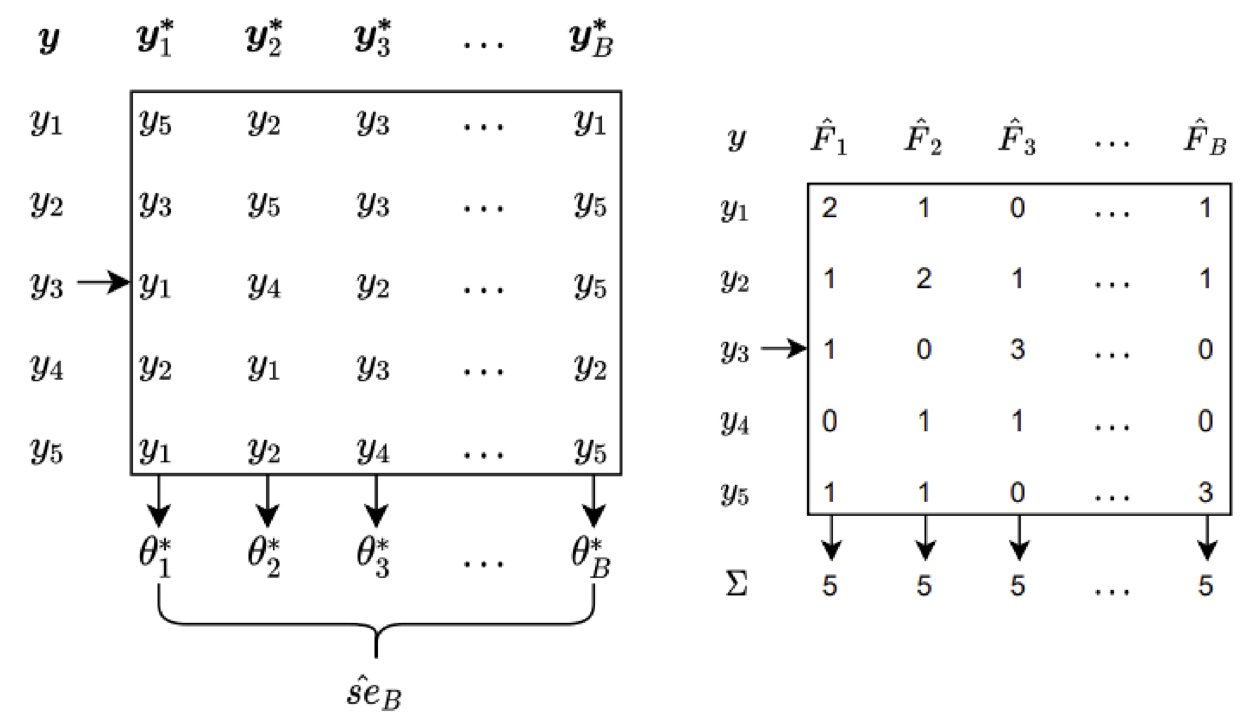
    2) Implement quantum circuits that replicate classical bootstrap sampling tasks.

Bootstrap Sampling

Given a sample $\vec{\mathbf{y}} = (\mathbf{y_1, y_2, \ldots, y_n})$ from an unknown distribution **F**,
A **bootstrap sample** $\vec{\mathbf{y}}^*$ = $(y_1^*, y_2^*, \ldots, y_n^*)$ is obtained by randomly sampling n times with replacement from $\vec{y}$.

Example (n = 5)
Possible bootstrap samples:
$\vec{\mathbf{y}}_1^* = (y_5, y_3, y_1, y_2, y_1)$
$\vec{\mathbf{y}}_2^* = (y_2, y_5, y_4, y_1, y_2)$
$\vec{\mathbf{y}}_3^* = (y_3, y_3, y_2, y_3, y_4)$

Estimated Frequency Distribution:
F^ = (f1^, f2^, . . .),
where f_k^ = #{y_i* = yk}/n



(a) Bootstrap Samples (b) Bootstrap Distributions

Fig. 1: Example: Bootstrap Sampling

Research Problem

SELECT Agg(attributes) FROM table WHERE conditions;

Problem-1: High computational cost of classical bootstrap sampling
Problem-2: Lack of efficient quantum-enabled frameworks for AQP error assessment

Goal: Can **quantum bootstrap sampling** framework be effectively leveraged to reduce the time complexity of bootstrap-based error assessment in AQP?

Literature Review

Theoretical Background

- Yu et al. (**2022**) used optimized bootstrap sampling for AQP but relied on classical computation.
- Phalak et al. (**2023**) compared QRAM architectures (Bucket-Brigade, Fanout, Flip-Flop, EQGAN, PQC) showing QRAM's advantage in speed and scalability over classical RAM.
- Wu et al. (**2024**) proposed a hybrid classical-quantum amplifier using Grover's operator and QRAM for rare group detection and amplification in AQP.
- Jiang et al. (**2024**) developed a simplified quantum counter that can act as an accumulator for summing binary outcomes based on Grover's algorithm to reduce gate complexity in NP-hard problems.

Research Gaps

While prior works have explored quantum sampling and memory access, none applied quantum methods to accelerate bootstrap-based error estimation in Approximate Query Processing.

Research Methodology

A **hybrid** quantum-classical framework is introduced to accelerate bootstrap-based error estimation for AQP.

It comprises **two** main quantum modules:

1) **Quantum ReSAmpler (QSA)**

2) **Quantum Counter (QC)**

The complete circuit integrates multiple parallel QSA modules feeding into a QC. The final count is measured and scaled on a classical computer.

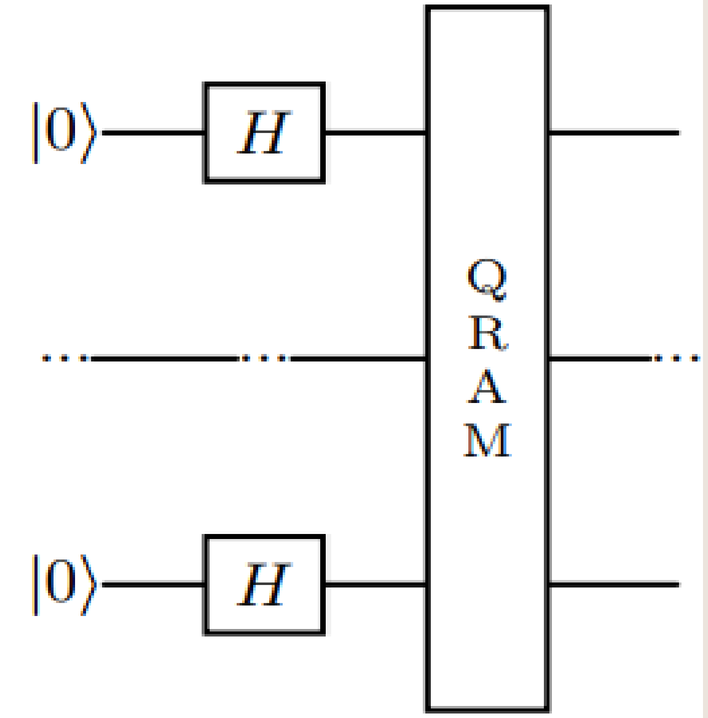
Experiments: Quantum circuit design

Figure (a) depicts the circuit design of the first step of the quantum sampling framework. It is the quantum resampler (QSA).

i) The input states are initialized in $|0\rangle^{\otimes n}$.

ii) Hadamard gates are used to simulate the randomly sampled indices with replacement.

iii) After that, the QRAM will translate the random sampling indices to the sample tuple results.



(a) Quantum reSAmpler (QSA)

Experiments: Quantum circuit design

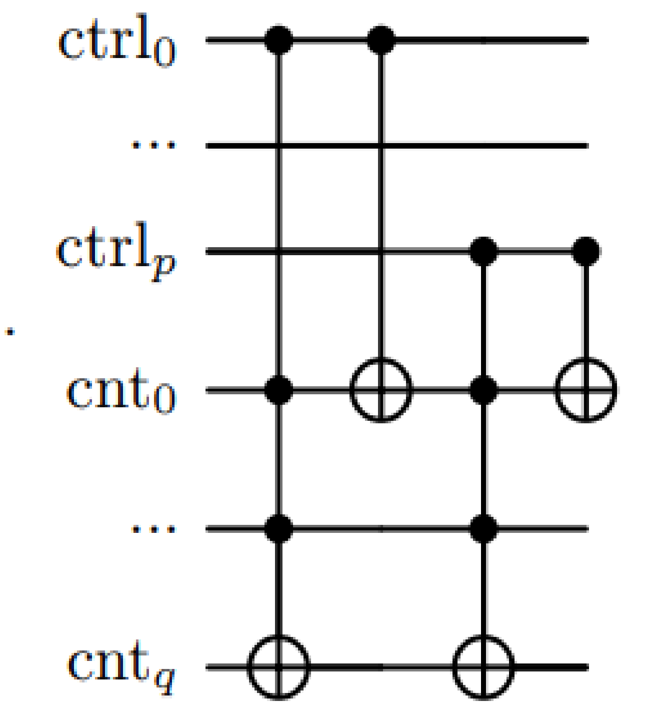
Figure (b) depicts a simplified quantum counter circuit to count the number of qubits in $|1\rangle$ state from the output of a quantum re-sampler.

i) The control qubits can be either $|0\rangle$ or $|1\rangle$ state. The counter qubits are initialized as $|0\rangle$ state.

ii) It accumulates how many control qubits are in the state $|1\rangle$.

iii) The output number represents the total of tuple results of 1, or selected, in the COUNT query, which can be used to compute the bootstrap replication.

iv) The counter qubits are connected to each control qubit using Toffoli and control-NOT gates.



(b) Quantum Counter (QC)

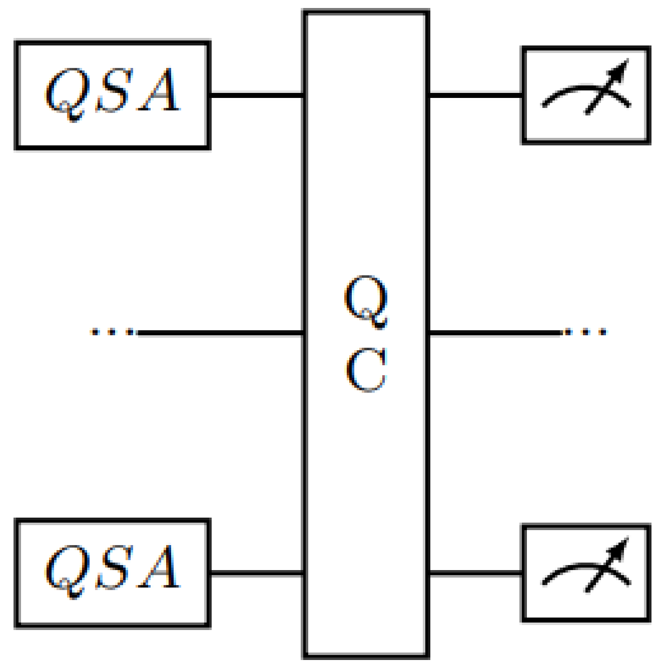
Experiments: Quantum circuit design

Figure (c) illustrates the overall quantum bootstrap sampling framework.

i) The sampled tuple results from multiple QSA in parallel will be fed into a quantum counter QC.

ii) The quantum counter will count the total of non-zero tuple results in each bootstrap resample.

iii) It can be measured and converted to a classical bit value in LSB format to produce a bootstrap replication.



(c) Quantum Bootstrap Replication

Circuits: QRAM & Quantum Counter

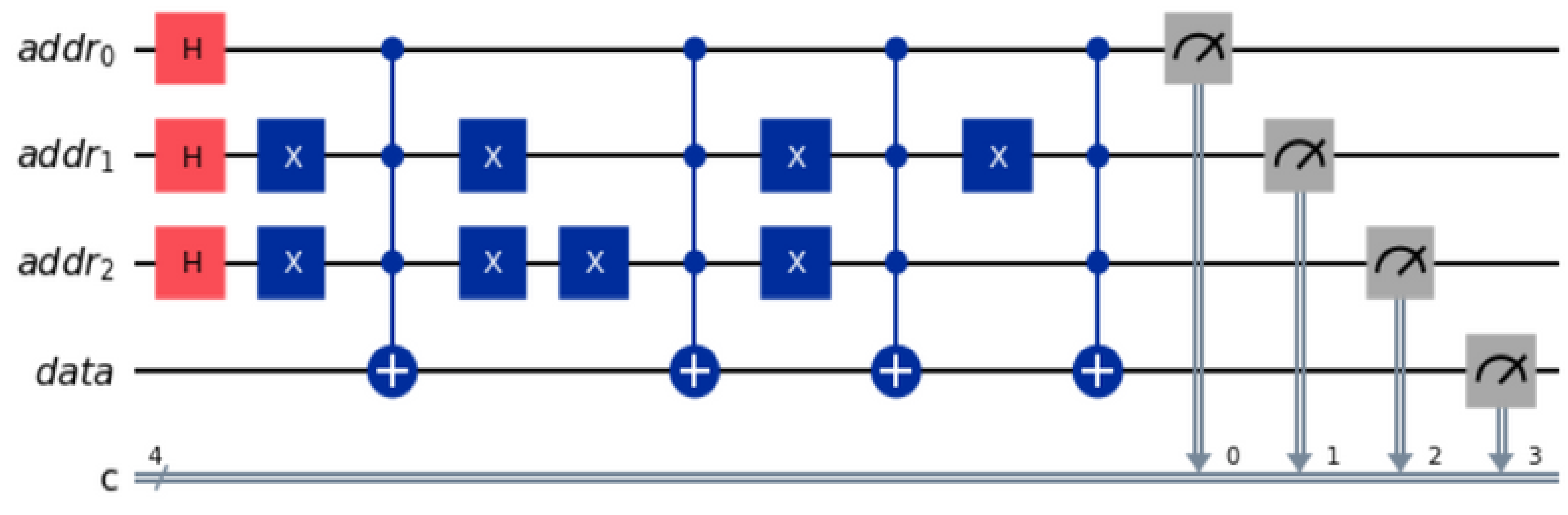


Fig. 3: An Implemented QRAM Circuit for Bootstrap Resampling

Circuits: QRAM & Quantum Counter

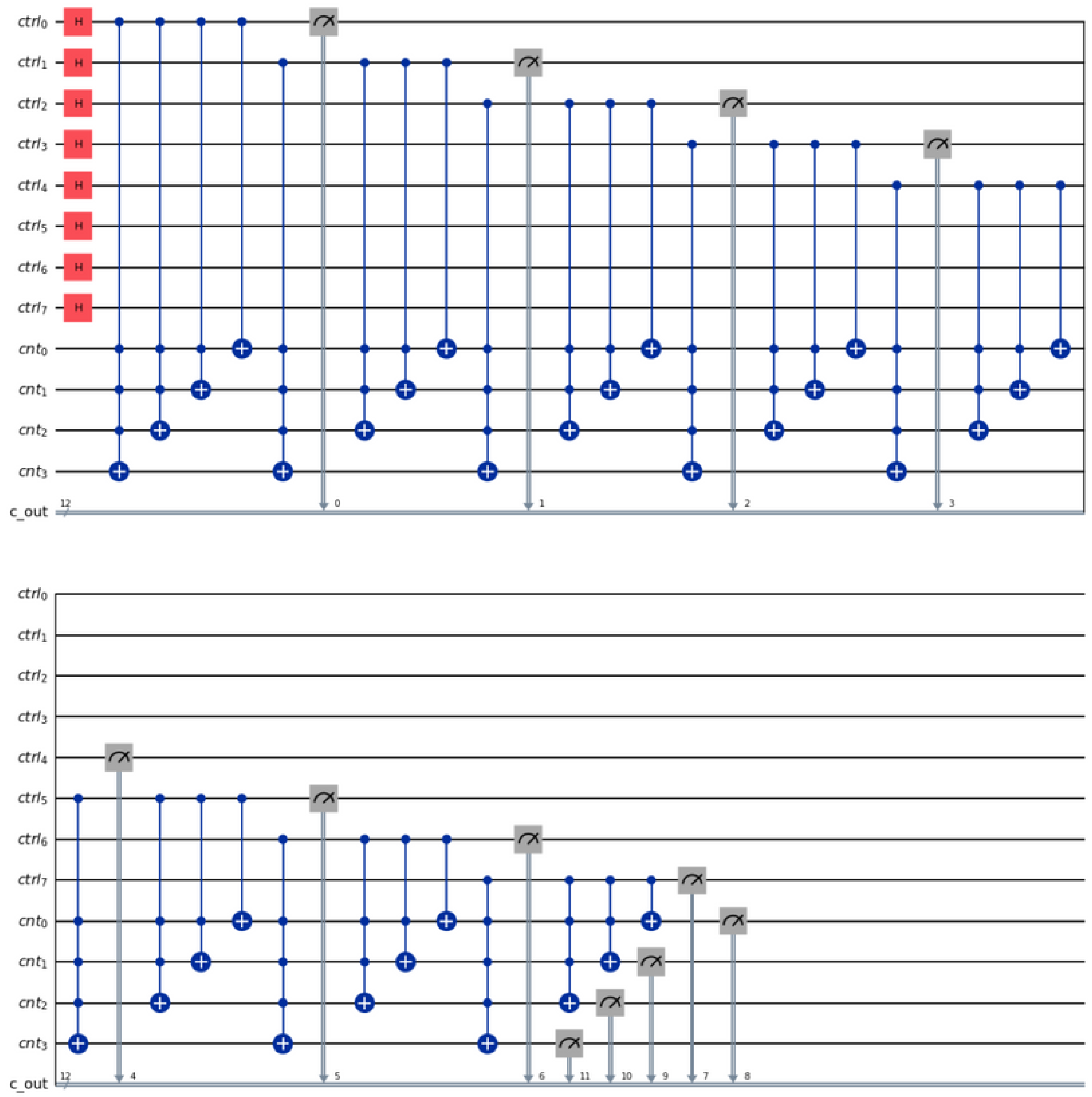


Fig. 5: A Simplified Quantum Counter to Count Qubits ($n \leq 8$) in $|1\rangle$ State

Data Collection & Analysis

- Conducted **1024** bootstrap replications using IBM Qiskit with three qubits representing $n = 2^3 =$ **8 sample tuples**.
- **Sample Encoding**: Initialized a static array for QRAM input where **odd** memory addresses are assigned a value of **1** and **even** addresses are assigned **0**.
- **QSA** generated uniform random tuple indices with replacement via Hadamard gates and QRAM.
- **QC** computed the count of selected tuples (value = 1) by summing up qubit states from the resampled data, implementing multi-controlled gates.

Execution: Used Qiskit's AerSimulator for circuit execution and measurement, reversing bit order to align with human-readable outputs.

Results

- **Uniform Sampling with Replacement:** Achieved via quantum superposition of address qubits passed through QRAM.
- **Tuple Counting:** The Quantum Counter accurately computes the number of 1's in the resampled tuples, representing selected rows.
- **Immediate Bootstrap Replication:** Each measurement directly yields a bootstrap replication without further post-processing.
- **Table 1** – QRAM Output Validation
- **Table 2** – Quantum Counter Simulation

Table 1: Measured QRAM Output States

Address (Binary)	Address (Decimal)	Data Qubit	Count
000	0	0	128
010	2	0	135
100	4	0	125
110	6	0	131
001	1	1	129
011	3	1	116
101	5	1	126
111	7	1	134

Table 2: One Result of Quantum Counter Simulation

Description	Value
Full bitstring measured	$\|010100011111\rangle$ (last bit to first bit)
Control bits measured	$\|00011111\rangle$ (5 ones)
Counter bits measured	$\|0101\rangle$ (binary of value 5)

Results (continued)

Figure 4 – Output Distribution: Displays nearly uniform sampling frequencies across 8 memory addresses, validating superposition and entanglement functionalities.

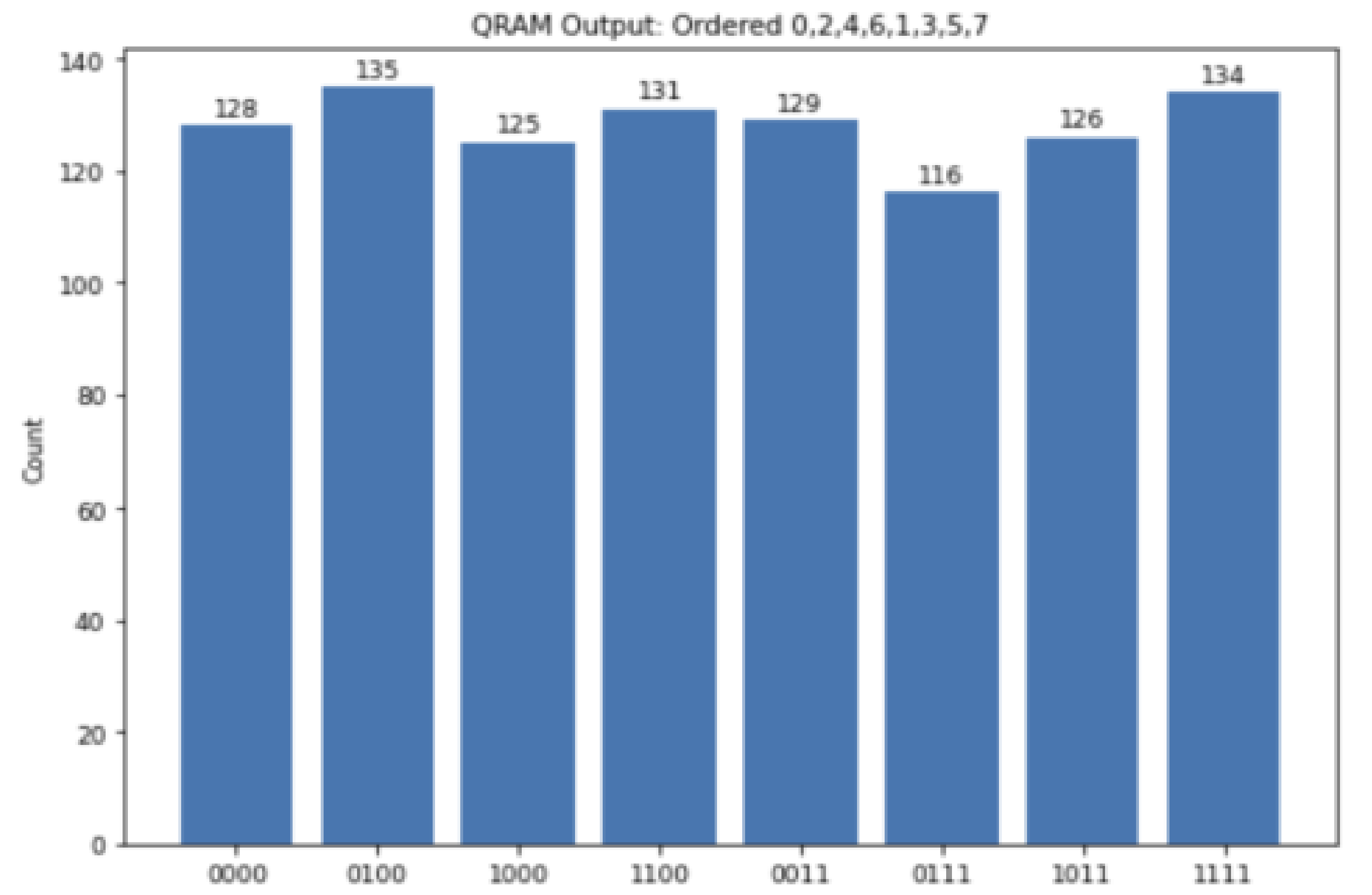


Fig. 4: A Distribution of QRAM Output States

SUM and AVG Queries

For **SUM** and **AVG** queries, **numeric values** yi ∈ R are used instead of binary values (0 or 1) in the input sample: S = {u1,u2,...,u_n}

Encoding in QRAM:
Fixed-width binary format for storing values
Example: 5 qubits for values up to 31

Mapping: QRAM maps address state |i⟩ → data value |yi⟩ using Toffoli and X gates

Example: Address |001⟩ → Data |10100⟩ for y1 = 20

Accumulation:
Quantum **ripple-carry adder circuit** is used to sum QRAM outputs

SUM: Direct total of values
AVG: SUM ÷ number of samples

Discussion

Practical/Theoretical Implications

- **Efficiency**: Quantum bootstrap replication is executed in constant time O(1) per sample due to quantum parallelism.
- **Scalability**: The proposed quantum framework supports parallel generation of bootstrap samples via multiple QSA circuits.

Discussion (Continued)

Limitations

- **Hardware Constraints**: Current NISQ devices support limited qubits
- Our research simulates the COUNT aggregation only. Encoding data for **SUM/AVG** remains future work.
- The experiments are based on **simulated tuple values**. Real-world datasets need to be integrated.

Future Work

- **Extension to SUM and AVG Aggregation functions**: Leverage quantum ripple-carry adder circuits and multi-bit QRAM data encoding for numeric tuple values.
- **Test on Real-World Datasets**: Integrate real data to evaluate the system under realistic database workloads.
- **Scale Hardware Implementation**: Transition from simulation to execution on real quantum processors as they evolve.

Conclusion

- Proposed a novel Quantum Bootstrap Sampling (QBS) framework combining quantum resampling and counting for error estimation in AQP tasks.
- Implemented quantum circuits using Qiskit to replicate classical bootstrap sampling via quantum methods.
- Verified through multiple measurements and tests that the system reproduces expected sampling behavior and count accuracy.

Thank You!

Feng “George” Yu, fyu@ysu.edu
Raya Jahan, rjahan01@student.ysu.edu