Project Brief: Machine Learning for Quantum State Tomography (ML-QST)

October 2025

1 Project Overview

Quantum State Tomography (QST) is the essential process of reconstructing the density matrix (ρ) of an unknown quantum state from a set of experimental measurements. While fundamental, conventional QST suffers from the "curse of dimensionality," requiring an exponentially growing number of measurements and computational resources as the number of qubits (N) increases.

This project focuses on leveraging advanced Machine Learning (ML) techniques—specifically deep learning and neuromorphic computing—to achieve **highly efficient, scalable, and real-time Quantum State Tomography**. The goal is to move beyond conventional methods to enable state certification on Noisy Intermediate-Scale Quantum (NISQ) and future fault-tolerant devices.

2 Key Applications and Impact

The developed ML-QST methods will have immediate and critical applications across the quantum ecosystem:

- **Quantum Algorithm Verification:** Efficiently certifying that the output state of a complex quantum circuit matches the theoretical expectation, a vital step in benchmarking and debugging.
- **Qubit Characterisation:** Providing rapid, high-fidelity measurement of the quantum state to determine parameters like purity, entanglement, and fidelity, crucial for validating hardware performance.
- **Quantum Sensing:** Utilizing reconstructed states to precisely estimate external parameters (e.g., magnetic fields) by identifying minute changes in the quantum system's density matrix.
- **Quantum Error Suppression (QES):** Real-time QST can monitor the state fidelity during computation, allowing for adaptive control pulses or immediate suppression strategies based on instantaneous state knowledge.
- **Neural Quantum Interface:** Developing ML models that serve as a compact, classical representation of a quantum state, creating an efficient interface between quantum hardware (measurement data) and classical control systems.

3 Technical Focus Tracks

The project will be executed across three parallel and collaborative technical tracks, targeting algorithmic, hardware acceleration, and novel computing approaches.

3.1 Track 1: Neural Shadow Tomography (NSQST)

- **Objective:** Implement and optimize Neural-Shadow Quantum State Tomography (NSQST), which leverages the proven sample-efficiency of the Classical Shadows formalism combined with neural networks for state reconstruction.
- **Methodology:** The neural network will be trained to reconstruct the density matrix (ρ) or predict key observables directly from classical measurement outcomes ("shadows") generated by randomized unitaries. The training loss will be based on minimizing the infidelity, estimated using the shadow data.
- **Key Deliverable:** A robust Python/Julia package for high-fidelity QST of up to 20 qubits using minimal measurement overhead.

3.2 Track 2: FPGA Accelerated Tomography

- **Objective:** Accelerate the inference phase of ML-QST algorithms (e.g., small feed-forward networks) using Field-Programmable Gate Arrays (FPGAs) to achieve sub-millisecond reconstruction times.
- **Methodology:** Quantize pre-trained neural network models (from Track 1) to fixed-point precision. Implement the high-level synthesis (HLS) design for the neural network on the FPGA fabric to enable massive parallel processing of measurement data streams.
- **Key Deliverable:** A low-latency hardware core capable of real-time state parameter estimation from high-throughput quantum measurement data.

3.3 Track 3: Neuromorphic Co-processor-based SNN for QST

- **Objective:** Develop a Spiking Neural Network (SNN) model and an associated simulation/deployment pipeline optimized for Quantum State Tomography, leveraging the low-power, event-driven architecture of neuromorphic co-processors.
- **Methodology:** Model the QST problem (mapping measurement outcomes to state parameters) as a pattern recognition task suitable for SNNs. Utilize event-based data representations and train SNNs using specialized frameworks to exploit the sparsity and temporal dynamics of quantum measurements.
- **Key Deliverable:** A prototype SNN model demonstrating improved energy efficiency compared to equivalent classical Deep Neural Networks (DNNs) for state reconstruction.

4 Technical Requirements and Software Stack

The following tools and libraries are required for the implementation of the technical tracks:

Category	Tool/Library	Purpose
Deep Learning	`torch` (PyTorch)	Core framework for implementing and training the ne
SNN Modeling	`snntorch`	Specialized library for fast and efficient simulation anc
Neuromorphic Simulation	`brian2`	Python-based simulator for large-scale, equation-base
High-Level Programming	`python`	Primary language for algorithm development, classica
High-Performance Computing	`julia`	Secondary language for high-performance numerical
Hardware Implementation	`verilog`	Hardware Description Language (HDL) required for im

Table 1: Technical Requirements and Software Stack for ML-QST.