CMSC 591: Introduction to Quantum Computing Project Proposal

Gokul Chaluvadi, Ayush Kumar Lnu, Denique Black
Department of Computer Science
Virginia Commonwealth University
Richmond, VA 23284
{chaluvadig, kumara48, blackd7}@vcu.edu

Abstract

This project focuses on Quantum Autoencoders (QAE). QAEs are quantum machine learning techniques that leverage variational quantum circuits to compress quantum information into fewer qubits. The issue is ensuring that the compression of quantum information preserves the fidelity of the original quantum state, allowing for accurate reconstruction afterwards. The goal of this project is to explore the trade-offs between compression fidelity and performance degradation, as well as in the presence of noise. Our group will aim to understand our objective through different approaches. First, we will compare different variational ansätze across compression ratios to gain a better understanding of their efficiency. Second, we will evaluate the performance of quantum autoencoders when applied to different families of quantum states to try and understand their generalization. Third, we will test the noise resilience of the compression methods using IBM Quantum's realistic noise models. Our results should provide an analysis of compression ratios, fidelity under noise, and scenarios where quantum compression outperforms classical techniques.

1 Introduction

1.1 Background and Motivation

Current quantum computing faces many challenges, such as noisy qubits and limited computational time. These challenges prevent the potential of quantum computers from performing quantum computations effectively. To handle these issues, many rely on QAEs as they compress quantum information from many to fewer qubits, all while maintaining the important details when reconstructing. This is similar to how classical autoencoders work, but with the addition of preserving quantum properties, which classical methods can't achieve.

QAEs are significantly useful in many applications. Such application being NISQ devices. It helps tasks such as quantum simulations, machine learning, and communication. Since these use cases are useful, they should come with different viewpoints/questions to understand the underlying foundation. How do different circuit designs affect the compression? How does quantum entanglement impact compressibility?

This project is focused on Quantum Machine Learning. We aim to help fill the gaps by studying how quantum autoencoders work on noise, with the goal of both understanding and improving their performance for practical applications.

The motivation for this project is to obtain a solid understanding of quantum data compression and the deployment of autoencoders on real-world quantum hardware to overcome the noise and imperfections in current systems.

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

1.2 Problem Statement

This project will maintain a goal to build a quantum QAE. The QAE will contain two quantum circuits, an encoder and a decoder. The encoder is used to reduce the dimensionality of quantum states, while the decoder reconstructs the original state as accurately as possible.

Quantum Autoencoder Components:

1. **Encoder (E):** The encoder will reduce the quantim state from n qubits to k qubits. This will occur through a paramterized quantum circuit.
2. **Decoder (D):** The deocder will then take the compressed k-qubit state and reconstructs it back into an n-qubit state. $(n - k)$ ancilla qubits initialized to $|0\rangle$.

The reconstruction of quantum states is measured by quantum state fidelity. This will ensure a quantifiable value to determine how close the reconstructed state is to the original. The goal is to optimize the circuit parameters to minimize the reconstruction error.

Key Research Questions:

1. **Ansatz Selection:** How do the different circuit designs affect the trade-off between the compression quality, fidelity, and the ability to train the model efficiently?
2. **Entanglement & Compressibility:** How does the structure of the entanglement in quantum states impact the efficiency of compression?
3. **Noise Robustness:** How does noise in quantum hardware affect compression performance, and can the quantum autoencoder still outperform classical methods under these noisy conditions?

Input/Output Specifications:

- **Input:** The model takes the quantum states from different families. Each state is prepared on $n \in \{4, 6\}$ qubits.
- **Output:** A compressed k-qubit representation, where $k \in \{n/2, n/3\}$.
- **Constraints:**
  - Circuit depth limited to fewer than 100 gates
  - Training expected to converge within 200 optimization steps
  - Total qubit usage, including reference and auxiliary qubits, not exceeding 10 qubits

Success Criteria:

- Achieve a reconstruction fidelity of greater than 0.95 for the product states across all the tested circuit designs. This will demonstrate effective basic compression.
- Achieve a fidelity greater than 0.85 for moderately entangled W states using the best-performing circuit. This will demonstrate the ability to capture moderate entanglement.
- Achieve a fidelity greater than 0.70 for highly entangled GHZ states under realistic noise. This will demonstrate robustness on NISQ hardware.

Classical Baselines for Comparison:

- **Principal Component Analysis:** Projects quantum state amplitudes onto top principal components, then reconstructs to measure error.
- **Random Unitary Compression:** Uses an untrained random unitary transformation as a baseline for minimum expected performance.

1.3 Related Work

- **Quantum approaches:** Variational quantum circuits are widely used in quantum autoencoders to efficiently compress quantum states (1; 2; 5).

- **Experiments on quantum hardware:** Locher et al. (3) demonstrated error correction using quantum autoencoders on NISQ devices. It showed good fidelity even with noise. The Qiskit tutorial (1) provides examples on simulators and real quantum hardware.
- **Classical methods and limits:** Classical autoencoders and PCA can compress classical data but have difficulty with highly entangled quantum states, as shown in studies on quantum image data (4).
- **Current challenges:** Research highlights the need to balance circuit depth, compression fidelity, and noise resilience (5). Scaling to larger qubit counts while keeping high fidelity is still difficult.
- **Our contribution:** This project builds on these approaches by testing variational quantum circuits for different qubit sizes and entanglement types, comparing with classical baselines, and evaluating robustness under realistic noise.
- **References:** Key techniques and frameworks come from the recent works cited above (1; 2; 3; 4; 5).

2 Quantum Computing Framework

2.1 Quantum Formulation

Classical Machine Learning Task: In classical machine learning, the unsupervised dimensionality reduction is used to learn compressed data representations. These representations retain the key information needed to accurately reconstruct the original data.

Quantum State Encoding: For quantum data, encoding is not needed since the inputs are already quantum states. On the other hand, when we work with classical data, we use amplitude encoding to represent the data in a quantum format.

Variational Quantum Circuit Architecture:

1. **Encoder $E(\boldsymbol{\theta}_E)$:** Maps n-qubit input to k-qubit latent space plus $(n-k)$-qubit trash space.
2. **Trash Removal:** Partial trace over trash qubits.
3. **Decoder $D(\boldsymbol{\theta}_D)$:** Reconstructs n-qubit state from k-qubit latent plus $(n-k)$ ancilla qubits in $|0\rangle$ state.

Ansatz Designs (Comparison):

1. **RealAmplitudes:** A basic circuit that uses simple rotation and connection layers.
2. **EfficientSU2:** A more flexible design that adds extra rotation layers, harder to train.
3. **Hardware-Efficient:** Built to match how real IBM quantum devices are wired, so it runs faster and with fewer errors.

Training and Loss Function: The model's parameters $\boldsymbol{\theta}$ are adjusted using the COBYLA algorithm to make the output state as close as possible to the input state. Another term for this is fidelity, which is measured with a SWAP test. The SWAP test checks how similar the trash part of the system is to a $|\text{ref}\rangle_{n-k} = |0\rangle^{\otimes(n-k)}$. The loss function is then defined based on the fidelity.

$$\mathcal{L}(\boldsymbol{\theta}) = 1 - \frac{1}{N}\sum_{i=1}^{N} \Pr(\text{aux} = 0|\, |\psi_i\rangle\,, \boldsymbol{\theta}) \tag{1}$$

Classical Optimizer: COBYLA, A gradient-free optimizer, works well even with the quantum results are noisy or uncertain.

Evaluation Metrics:

1. **Reconstruction Fidelity:** Measures how closely the output state matches the input state.

2. **Worst-Case Fidelity:** Checks the lowest performance across all test cases to measure consistency.
3. **Circuit Metrics:** Evaluates circuit size, depth, and gate count to estimate hardware cost.
4. **Training Efficiency:** Tracks how quickly and smoothly the model learns during optimization.
5. **Noise Robustness:** Tests how well the model performs when there is noise in the quantum system.

Comparison with Classical Baselines:

- **PCA:** A simple classical method that reduces data dimensions using linear transformations.
- **Metric Conversion:** Ensures that results from quantum and classical models can be fairly compared using the same metrics.

2.2 Quantum Resources

Quantum Platform: IBM Quantum (free tier) **Execution Strategy:**

- **Primary (95%):** Qiskit Aer simulators (local, unlimited, free)
- **Hardware Validation (5%, optional):** 2-3 validation runs on IBM quantum hardware within free tier limits

Qubit Requirements:

- Initial experiments: $n = 4$ input qubits, $k = 2$ latent $\rightarrow$ 6 total qubits (4 data + 2 reference + 1 auxiliary)
- Extended experiments: $n = 6$ input, $k = 3$ latent $\rightarrow$ 10 total qubits

Circuit Depth & Gate Count Estimates:

- Encoder depth: $5 - 10$ layers $\times$ (1 entanglement + 1-2 rotations/qubit) $= 30 - 80$ gates
- Decoder depth: Similar to encoder
- SWAP test: $\approx 2(n - k)$ CSWAP gates + 2 Hadamards
- Total depth: $60 - 180$ gates (within NISQ limits)

Software Stack:

- **Qiskit:** Circuit construction, transpilation, execution
- **Qiskit Machine Learning:** `SamplerQNN` for training loop
- **Qiskit Aer:** High-performance simulators with noise models
- **SciPy:** Classical optimization
- **NumPy/Pandas:** Data processing and analysis
- **Matplotlib/Seaborn:** Visualization of results
- **Scikit-learn:** Classical ML baselines

Noise Models:

- **IBM Fake Backends:** `FakeManilaV2` (5 qubits), `FakeSherbrooke` (127 qubits, use subset)
- **Error Rates:** Gate fidelity $\sim 99\%$ (1-qubit), $\sim 97\%$ (2-qubit); T1/T2 $\sim 100\mu s$
- **Readout Errors:** $\sim 1 - 3\%$ misclassification

3 Methodology

3.1 Implementation Approach

Software Architecture:

```
project/
 src/
    ansatze/
       real_amplitudes.py      # RealAmplitudes ansatz
       efficient_su2.py        # EfficientSU2 ansatz
       hardware_efficient.py   # IBM-optimized ansatz
    circuits/
       encoder.py              # Encoder construction
       decoder.py              # Decoder construction
       qae_circuit.py          # Full QAE with SWAP test
    state_preparation/
       product_states.py       # Simple product states
       entangled_states.py     # W, GHZ states
    training/
       qnn_trainer.py          # SamplerQNN training loop
       cost_function.py        # SWAP test cost
       optimizer_config.py     # COBYLA settings
    evaluation/
       fidelity_metrics.py     # Fidelity calculations
       circuit_metrics.py      # Depth/gate analysis
       noise_analysis.py       # Noise robustness
    baselines/
        classical_pca.py       # PCA baseline
 experiments/
    exp1_ansatz_comparison.py
    exp2_compression_ratios.py
    exp3_noise_robustness.py
 notebooks/
    analysis_visualization.ipynb
```

Circuit Construction Workflow:

1. Instantiate ansatz from library with specified parameters
2. Construct encoder by applying ansatz to input qubits
3. Add reference state preparation (initialize to $|0\rangle$)
4. Construct decoder (inverse ansatz or separate parameterization)
5. Append SWAP test between trash and reference qubits
6. Add measurement on auxiliary qubit

Parameterization Strategy:

- **Shared Parameters:** Encoder and decoder share parameters (adjoint architecture), reduces parameter space, faster training
- We will test both strategies and report trade-offs

Optimization Strategy:

- **Optimizer:** COBYLA
- **Initialization:** Random parameters with 3 random seeds for statistical robustness
- **Convergence Criterion:** $\mathcal{L} < 0.05$ or plateau detection.
- **Batch Training:** Train on small state batches to reduce evaluation cost

3.2 Experimental Design

Experiment 1: Ansatz Comparison (Primary Contribution): Systematically compare three ansätze across compression tasks.

Setup:

- States: Product states (10 samples), W states (5 samples)
- Compression: $4 \to 2$ qubits
- Ansätze: RealAmplitudes, EfficientSU2, Hardware-Efficient
- Layers: $L \in \{3, 5\}$ for each ansatz
- Execution: Ideal simulator, 3 random seeds per configuration

Metrics:

- Average fidelity F_{avg} for each ansatz-layer combination
- Training convergence (iterations to $\mathcal{L} < 0.05$)
- Circuit depth and two-qubit gate count

Experiment 2: Entanglement Structure Study: Characterize compression difficulty based on entanglement.

Setup:

- State families:
  - Product: $|0110\rangle, |1001\rangle, \ldots$ (separable, 10 samples)
  - W states: $(|1000\rangle + |0100\rangle + |0010\rangle + |0001\rangle)/2$ (limited entanglement)
  - GHZ: $(|0000\rangle + |1111\rangle)/\sqrt{2}$ (maximal entanglement)
- Compression: $4 \to 2$ qubits
- Ansatz: Best from Exp 1 (likely EfficientSU2)
- Execution: Ideal simulator

Metrics:

- Fidelity vs entanglement entropy: $S(\rho_A) = -\text{Tr}(\rho_A \log_2 \rho_A)$
- Compression quality for each state family

Experiment 3: Noise Robustness Analysis (Primary Contribution): Characterize performance degradation under realistic noise.

Setup:

- States: Product states, W states
- Compression: $4 \to 2$ qubits
- Ansatz: Best from Exp 1
- Noise models:
  - `FakeManilaV2` (5-qubit device, moderate noise)
  - Custom noise: Vary gate error rates $\epsilon \in \{0.001, 0.01\}$
- Execution: 5 trials per noise level with 1024 shots

Metrics:

- Fidelity degradation: $\Delta F(\epsilon) = F_{\text{ideal}} - F_{\text{noisy}}(\epsilon)$
- Comparison with classical methods under same noise (shot noise for sampling)

3.3 Dataset Description

Quantum State Families:

1. **Product States:**
   - Generation: Random bit strings $|b_1 b_2 \dots b_n\rangle$ where $b_i \in \{0, 1\}$
   - Properties: Zero entanglement, easy to compress
   - Samples: 10 random instances per qubit size
2. **W States:**
   - Definition: $|W_n\rangle = \frac{1}{\sqrt{n}}(|10\dots0\rangle + |01\dots0\rangle + \dots + |0\dots01\rangle)$
   - Properties: Symmetric, limited entanglement, robust to particle loss
   - Samples: 5 instances with random single-qubit rotations applied
3. **GHZ States:**
   - Definition: $|\text{GHZ}_n\rangle = \frac{1}{\sqrt{2}}(|0\rangle^{\otimes n} + |1\rangle^{\otimes n})$
   - Properties: Maximal entanglement, fragile, difficult to compress
   - Samples: 5 pure GHZ

State Preparation Circuits:

- Product states: Direct basis state initialization or single-qubit X gates
- W states: Recursive construction using controlled rotations
- GHZ states: Hadamard on first qubit + CNOT cascade

4 Project Plan and Timeline

4.1 Task Breakdown

Phase 1: Foundation & Baseline (Days 1-5)

- Days 1-2: Literature review (recent QAE papers), environment setup
- Days 3-4: Reproduce Qiskit QAE tutorials, validate understanding
- Day 5: Implement state preparation functions for all state families

Phase 2: Ansatz Implementation (Days 6-11)

- Days 6-7: Implement all ansätze comparisons
- Day 8: Build training framework with optimizer and logging
- Days 9-11: Experiment 1 (ansatz comparison across layers and compression ratios)
- Deliverable: Comparative plots showing fidelity vs depth for each ansatz

Phase 3: Entanglement Study & Noise Analysis (Days 12-15)

- Day 12: Experiment 2 (compression of different state families)
- Days 13-14: Implement noise models and Experiment 3 (noise robustness)
- Day 15: Classical baseline implementation (PCA)
- Deliverable: Fidelity vs entanglement entropy plots

Phase 4: Analysis & Documentation (Days 16-18)

- Day 16: Compile all results
- Day 17: Create figures, statistical analysis, write key findings
- Day 18: Finalize report and presentation slides, code documentation
- Deliverable: Complete project report and presentation

4.2 Timeline

Week 1 (November 10-16):

- Mon-Tue: Submit proposal, literature review, Qiskit tutorial reproduction
- Wed-Thu: State preparation implementation, baseline QAE testing
- Fri-Sun: All three ansätze implemented and tested on simple cases
- Milestone: Working QAE with at least 2 ansätze, product state compression $F > 0.9$

Week 2 (November 17-23):

- Mon-Tue: Complete Experiment 1 (ansatz comparison on ideal simulator)
- Wed: Experiment 2 (entanglement structure study)
- Thu-Fri: Experiment 3 (noise robustness analysis)
- Weekend: Classical baselines setup
- Milestone: All experimental data collected, preliminary analysis complete

Week 3 (November 24-28):

- Mon: Final data collection
- Tue-Wed: Statistical analysis, figure generation, result interpretation
- Thu: Report writing (methodology, results, discussion sections)
- Milestone: Submit complete project by November 28

4.3 Team Member Responsibilities

Gokul Chaluvadi:

- Lead implementation of ansatz architectures (RealAmplitudes, EfficientSU2, Hardware-Efficient)
- Design and execute Experiment 1 (ansatz comparison across compression ratios)
- Develop circuit construction workflow and parameterization strategies
- Coordinate software architecture and module integration

Ayush Kumar Lnu:

- Implement quantum state preparation functions (product, W, GHZ states)
- Design and execute Experiment 2 (entanglement structure study)
- Develop training framework with COBYLA optimizer and cost function implementation
- Compute and analyze entanglement entropy metrics and fidelity measurements

Denique Black:

- Implement noise models (FakeManilaV2, custom noise configurations)
- Design and execute Experiment 3 (noise robustness analysis)
- Develop classical baseline implementations (PCA comparison)
- Conduct performance evaluation across ideal and noisy simulators

5 Conclusion

This project will try and aims to provide an analysis of Quantum Autoencoders for practical solutions for quantum data compression on NISQ devices. We will systematically compare variational ansatz design, evaluate compression performances across different entanglement structures, and characterize noise robustness under realistic hardware conditions.

Expected Contributions:

- Ansatz Performance Characterization
- Entanglement-Compressibility Relationship
- Noise Resilience Analysis
- Reproducible Framework

Our project will help bridge the gap by understanding the trade-offs between compression accuracy, circuit complexity, and noise is key for practical quantum computing. The results we provide will guide the design of quantum algorithms for simulations, communication, and other quantum-classical tasks. Our work will help bridge the gap between theory and deployable quantum applications. It should address the challenge of efficiently representing and transmitting quantum information on limited, noisy qubits, thus providing insight/results for future development.

References

[1] Qiskit Community. (2023). *The Quantum Autoencoder Tutorial*. `https://qiskit-community.github.io/qiskit-machine-learning/tutorials/12_quantum_autoencoder.html`. Accessed: 2025-11-10.

[2] Zhuang, S., Wu, Y., Cadet, X. F., Huynh, D. Q., Liu, W., Charton, P., Damour, C., & Wang, J. B. (2025). Quantum Autoencoder: An efficient approach to quantum feature map generation. *arXiv preprint arXiv:2509.19157*. `https://arxiv.org/abs/2509.19157`

[3] Locher, D. F., Cardarelli, L., & Müller, M. (2023). Quantum Error Correction with Quantum Autoencoders. *Quantum*, 7, 942. `https://quantum-journal.org/papers/q-2023-03-09-942/`

[4] Asaoka, H., & Kudo, K. (2025). Quantum Autoencoders for Image Classification. *arXiv preprint arXiv:2502.15254*. `https://arxiv.org/abs/2502.15254`

[5] Author, E., & Author, F. (2025). Quantum State Compression via Variational Circuits. In *2025 IEEE International Conference on Quantum Computing*. IEEE. `https://ieeexplore.ieee.org/document/10752554`