VILNIUS UNIVERSITY

FACULTY OF MATHEMATICS AND INFORMATICS

DATA SCIENCE STUDY PROGRAMME

Master's Thesis

Quantum Autoencoders for Anomaly Detection

Kvantiniai Autoenkoderiai Anomalijų Aptikimui

Danial Yntykbay

Supervisor : Ernestas Filatovas

Vilnius
2026

Summary

This thesis investigates quantum autoencoders for anomaly detection under strong dimensionality constraints. We show that simple QAE architectures suffer from overfitting and poor class separability when aggressive compression is applied, leading to elevated false-positive rates (FPR).

To address these limitations, we introduce a *BrainBox* layer that enhances encoder expressivity while preserving a compact circuit structure. When combined with dropout and ℓ_2 regularization, the proposed approach yields more stable latent representations, improves reconstruction fidelity, and reduces FPR while maintaining strong detection performance.

Experiments using classical and quantum kNN classifiers demonstrate consistent improvements in F1-score and PR-AUC, highlighting the importance of architectural design and regularization in practical quantum anomaly detection.

Santrauka

Šiame darbe nagrinėjami kvantiniai autoenkoderiai anomalijų aptikimui esant stipriems dimensionalumo apribojimams. Parodoma, kad paprastos QAE architektūros, taikant agresyvų suspaudimą, linkusios į persimokymą ir silpną klasių atskyrimą, dėl ko padidėja klaidingų teigiamų aptikimų dažnis (FPR).

Siekiant sumažinti šiuos trūkumus, pristatomas *BrainBox* sluoksnis, kuris padidina enkoderio išraiškingumą išlaikant kompaktišką grandinės struktūrą. Derinant šį metodą su *dropout* ir ℓ_2 reguliavimu, gaunamos stabilesnės latentės reprezentacijos, pagerinamas rekonstrukcijos tikslumas ir sumažinamas FPR, išlaikant aukštą aptikimo kokybę.

Eksperimentiniai rezultatai, taikant klasikinius ir kvantinius kNN klasifikatorius, rodo nuoseklų F1 rodiklio ir PR–AUC pagerėjimą, patvirtinant, kad architektūriniai sprendimai ir reguliavimas yra esminiai praktiniam kvantiniam anomalijų aptikimui.

List of Figures

List of Tables

Contents

Introduction

Quantum machine learning is a quickly growing field as researchers explore how quantum computing can improve data-driven modeling. Classical machine learning has achieved significant success across many domains however, it often faces computational limitations when coping with high-dimensional and noisy datasets. Quantum computing offers fundamentally new capabilities by using superposition, entanglement, and large structure of Hilbert spaces. These properties motivate the investigation of quantum learning architectures that may offer advantages in representation learning, compression, and anomaly detection [5, 42].

Quantum Autoencoders (QAEs) have become important quantum learning models for dimensionality reduction and quantum compression. The first study by Romero et al. demonstrated that quantum states can be compressed into a smaller and less number of qubits through QAE training [39]. This idea has been extended through variational quantum autoencoders [23], parameter-efficient circuit designs [3], and hybrid reconstruction strategies [34]. At the same time, the autoencoder framework is suited for anomaly detection, and several works have applied QAEs to identify anomaly or attacks in both classical and quantum data settings. [27, 36].

However, a reproducing the results of recent studies under the same condition reveals an important limitation. The Reproduced results arre shown in Appendix 10 table.. Many works focus primarily on evaluating anomaly detection performance based on reconstruction error for anomalous or attack samples, while providing limited analysis of how well the QAE generalize across normal data. When reproducing several of these models, even those reporting strong detection capabilities, it becomes evident that QAEs often show high false positive rates, meaning by a high portion of normal inputs is incorrectly predicted as anomalous. This suggests that current QAE methods may overfit the training distribution, fail to learn latent representations or sensitive to circuit structure and noise. The lack of systematic evaluation of generalization on normal samples represents a significant gap in the existing literature and calls for a more robust analysis of QAE behavior.

The main aim of this thesis is to investigate and improve the generalization and latent-space quality of quantum autoencoders for anomaly detection focusing on false-positive rare and generalization on normal and training data.

To achieve this aim, the following objectives are pursued:

1. To analyze how quantum autoencoder performance is affected by architectural factors such as qubit count, circuit depth, number of variational layers, and latent bottleneck size.

2. To evaluate the generalization behaviour of quantum autoencoders on normal data, with explicit focus on false-positive rates rather than anomaly detection accuracy alone.

3. To study the structure and usefulness of quantum-compressed latent representations using both classical and quantum distance-based classifiers.

4. To investigate the effectiveness of quantum-inspired regularization techniques, including ℓ_2 regularization, gate dropout, and layer dropout, in reducing overfitting and improving stability.

5. To design and evaluate a latent-space processing mechanism that enhances the robustness and separability of quantum autoencoder for anomaly detection.

Moreover to these generalization problems, the architectural design of QAEs are still underexplored. Classical deep learning research has shown that network depth, bottleneck size, parameter regularization, and dropout play a improtant role in model performance and generalization. The same studies in the quantum settings have not been fully explored. Key components such as the number of qubits, circuit depth, number of layers, latent bottleneck structure, and quantum-inspired regularization strategies have not been systematically analyzed in the context of anomaly detection.

To address these challenges, this thesis extends QAE-based anomaly detection in several new directions. First, the latent representations produced by the QAE are evaluated using both Quantum k-Nearest Neighbors (QkNN) and a classical kNN. This evaluation investigates whether quantum-compressed latent space keep meaningful structure for distance based classification and whether quantum distance measures show advantages over classical ones. The second contribution of the thesis is a latent space denoising technique that is called the brainbox layer. This component is designed to preserve the latent representation, reduce noise and improve the separability between normal and anomalous samples, functioning similarly to denoising methods in classical autoencoder representation learning but adapted for hybrid quantum method.

The main contributions of this thesis can be summarized as follows:

1. Architectural analysis of how qubit count, circuit depth, number of layers, and latent bottleneck size influence for anomaly detection performance.

2. Introducing brainbox layer to denoise and improve the latent space for anomaly detection accuracy.

3. Evaluation of quantum regularization techniques, including L2 parameter penalties, gate and layer dropout to reduce overfitting.

4. Investigation of latent space structure using both Quantum kNN and classical kNN to assess classification performance based on compressed quantum data.

This thesis seeks to improve the reliability and practical use of quantum autoencoders for anomaly detection by tackling issues of architectural design, regularization, and latent-space processing.

1 Related Work

1.1 Review of quantum and classical machine learning on anomaly detection

Quantum autoencoders (QAEs) have emerged as promising models for dimensionality reduction, feature extraction and anomaly detection in quantum machine learning. Across domains such as finance, healthcare, cybersecurity, and time-series analysis, QAEs have been shown to reach competitive performance compared to classical autoencoders, often with significantly fewer trainable parameters. In what follows, we first review QAE-based and related quantum models and then compare them with classical machine learning approaches on the same benchmark datasets.

1.2 Quantum autoencoders and related QML models.

Early work by Romero et al. introduced the quantum autoencoder for quantum data compression demonstrating that variational quantum circuits can compress multi qubit quantum states while preserving essential information [39]. Their method optimises the fidelity between reconstructed and reference states, establishing the QAE framework that later works adopt for anomaly detection. This architecture was subsequently extended for quantum system learning and fidelity estimation using parameterised circuits [8].

Frehner and Stockinger applied QAEs to classical time-series anomaly detection using UCR datasets showing proposing reconstruction based and SWAP test based anomaly metrics [11]. Using amplitude encoding and hardware-efficient ansätze, they showed that QAEs can outperform classical autoencoders while fewer parameters. Experiments on real quantum hardware further demonstrated NISQ viability. However, their evaluation focuses primarily on anomaly detection accuracy, with limited analysis of reconstruction quality or generalization on normal samples.

In network security, several hybrid QAE classifier frameworks have been suggested. Hdaib et al. introduced QAE-based models for IoT and KDD intrusion datasets, taking QAE latent vectors with quantum one-class SVMs, quantum random forests, and quantum kNN classifiers [14, 15]. Their approach uses PCA preprocessing, angle and amplitude embeddings, and low-qubit circuits to compress network traffic features. Results on IoT23 and CIC-IoT datasets indicate that QAE+QkNN pipelines outperform classical baselines, demonstrating QAEs as effective quantum feature extractors for cybersecurity anomaly detection.

A recent contribution is the QAE-FD model for credit-card fraud detection by Huot et al. [19]. The authors propose an encoder-only QAE with threshold-based fidelity scoring for anomaly detection. On a highly imbalanced real-world dataset (284 315 genuine vs. 492 fraudulent transactions) they report an AUC of 0.947 and a G-mean of 0.946, outperforming classical autoencoders, quantum one-class SVMs and quantum graph neural networks. Their evaluation on IBM FakeCairo backends and an ablation study over circuit depth provide evidence that shallow QAE architectures can be effective for large scale financial anomaly detection under extreme class imbalance.

Beyond QAEs several alternative quantum anomaly detection architectures have been explored. Liu and Rebentrost proposed a quantum kernel-based anomaly detector using density ma-

trices and quantum kernels [26]. Park et al. introduced a variational quantum one-class classifier (VQOCC) that reuses only the encoder portion of a QAE and performs one-class classification using fidelity thresholds [32]. Their model outperforms classical autoencoders and rivals quantum SVMs. GAN-inspired approaches have also been investigated: variational quantum GANs have shown high performance in high-energy physics anomaly detection [16], while other works apply parameterised quantum circuits to long-lived particle detection [6] and quantum phase anomaly detection [25].

Further quantum models address fraud detection in non-autoencoder frameworks. Kyriienko and Magnusson proposed an unsupervised quantum one-class SVM using quantum kernels, requiring 20 qubits for the fraud dataset and demonstrating modest recall improvements [24]. Quantum graph neural networks (QGNNs) have achieved improvements of approximately 3 % over classical GNNs, albeit with lower recall [20]. Mixed quantum–classical models combining quantum feature selection with classical classifiers also show performance gains [13]. Complementary work includes the Quantum Variational Autoencoder (QVAE) by Khoshaman et al. [23], which integrates classical encoders with quantum Boltzmann machine priors. While not designed for anomaly detection, QVAE demonstrated improved generative modelling and structured latent spaces, influencing subsequent QAE designs that incorporate structured latent distributions.

1.3 Quantitative comparison of quantum and classical models.

Table 1 table. summarises reported performance metrics for representative QAE-based, more general QML, and classical anomaly-detection models across different domains. Quantum models such as QAE-FD and QAE+QkNN achieve competitive AUC and F1 scores despite using relatively few qubits and shallow circuits, whereas classical autoencoders on the same datasets tend to suffer from low precision and F1, especially under extreme class imbalance.

1 table. *Performance comparison of QAE-based, QML-based, and classical anomaly detection models. Metrics reported in percent. "–" indicates not reported; * marks values computed from precision and recall.*

Author	Model	Dataset	Acc.	Prec.	Rec.	F1	AUC
Romero et al. (2017)[39]	QAE (quantum state compression)	Quantum states	–	–	–	Fidelity > 90	–
Frehner & Stockinger (2025)[11]	QAE (amplitude encoding)	UCR Time Series	94–97	–	–	–	≈ 90
Park et al. (2023)[32]	VQOCC (QAE encoder)	Synthetic anomalies	–	–	–	–	93–96
Herr et al. (2021)[16]	Variational Quantum GAN	HEP anomalies	–	–	–	–	≈ 98
Bordoni et al. (2023)[6]	PQC anomaly detector	LLP high-energy anomalies	–	–	–	–	89–94
Hdaib et al. (2024)[15]	QAE + OC-SVM	KDD99 intrusion	97.48	95.06	99.43	97.19	–
	QAE + QRF	KDD99 intrusion	92.39	89.63	97.16	93.41	–
	QAE + QkNN	IoT23 / CIC-IoT	97.79	98.37	98.81	98.26	–
Huot et al. (2024)[19]	QAE-FD (encoder-only)	Credit-card fraud	99.0	37.9	89.7	53.3	94.7
	QGNN	Credit-card fraud	92.0	94.5	79.5	86.4*	92.9
	Classical Autoencoder	Credit-card fraud	80.0	9.0	91.7	16.4*	86.6
Classical ML on Credit-Card Fraud							
Dal Pozzolo et al. (2015)[9]	Random Forest	Credit-card fraud	99.94	27.6	90.0	42.3	97.0
Carcillo et al. (2019)[7]	Logistic Regression	Credit-card fraud	99.92	12.3	88.0	21.7	95.0
Bahnsen et al. (2016)[1]	Cost-sensitive Decision Tree	Credit-card fraud	99.95	20.1	67.9	31.0	92.0
Jurgovsky et al. (2018)[21]	SVM baseline	Credit-card fraud	–	7.5	83.0	13.7	91.5
Duman & Ozcelik (2014)[10]	kNN classifier	Credit-card fraud	99.80	6.0	89.0	11.2	90.0
Al-Shabi (2019)[44]	Autoencoder (reconstruction threshold)	Credit-card fraud	–	–	91.0	–	–
Zou et al. (2019)[51]	Denoising Autoencoder (DAE + SMOTE)	Credit-card fraud	97.93	–	84.0	–	–
Classical ML on KDD99 Intrusion							
Tavallaee et al. (2009)[47]	Random Forest	KDD99 intrusion	99.0	98.0	99.8	98.9	–
Amor et al. (2004)[4]	Naive Bayes	KDD99 intrusion	92.6	88.2	89.4	88.8	–
Shiravi et al. (2012)[45]	SVM classifier	KDD99 intrusion	96.8	94.5	95.8	95.1	–
Mukkamala et al. (2005)[29]	ANN (MLP)	KDD99 intrusion	97.1	95.3	96.4	95.8	–
Hettich & Bay (1999)[17]	Decision Tree (C4.5)	KDD99 intrusion	91.8	90.1	92.3	91.2	–

Table 2 table. complements these results by summarising the encodings, qubit counts, and circuit structures used in the QAE-based models reviewed above.

2 table. *Summary of QAE model encodings, qubit counts, and circuit architectures.*

Author	Encoding	Qubits	Circuit / Architecture
Romero et al. (2017)[39]	Direct state	2–4	Variational encoder–decoder; fidelity loss
Frehner & Stockinger (2025)[11]	Amplitude	6–8	EfficientSU2 layers; SWAP-test output
Park et al. (2023)[32]	Angle	2–6	Encoder-only QAE; one-class fidelity
Herr et al. (2021)[16]	Data re-uploading	6–12	PQC generator–discriminator (VQGAN)
Bordoni et al. (2023)[6]	Angle	6–12	Alternating variational blocks
Hdaib et al. (2024)[15]	PCA $\rightarrow$ Angle/Amplitude	4	Two-layer variational encoder; SWAP-test
Huot et al. (2024)[19]	Angle + PCA	4	Shallow Rx–Ry–Rz blocks; trash-qubit fidelity
Khoshaman et al. (2018)[23]	Classical $\rightarrow$ QBM prior	–	Hybrid variational autoencoder

1.4 Comparison with classical machine learning baselines.

A broader perspective emerges when classical machine learning results on the Credit-card fraud and KDD99 datasets are incorporated into the comparison. These benchmarks provide an essential context for evaluating the practical value of quantum models. On the Credit-card fraud dataset,

classical supervised learning methods such as Random Forests, Logistic Regression, kNN, and cost-sensitive decision trees consistently report extremely high overall accuracy (often exceeding 99%), but their precision—the proportion of correctly identified frauds—remains comparatively low [1, 7, 9, 10, 21]. This behaviour is characteristic of heavily imbalanced datasets: accuracy is dominated by the majority class and thus fails to reflect anomaly-detection capability. Even models with strong recall, such as the Random Forest baseline with 90% recall, typically achieve modest F1-scores (e.g., 42.3%), revealing an inability to maintain both sensitivity and precision simultaneously.

Classical autoencoders have also been widely explored for fraud detection, yet their performance is similarly constrained by class imbalance. Al-Shabi's autoencoder-based anomaly detector achieves a recall of 91% but does not report strong precision, while denoising autoencoders with SMOTE augmentation (e.g. Zou et al.) achieve high accuracy and reasonable recall but lack consistent improvements in F1-score across thresholds [44, 51]. Overall, classical autoencoders tend to reconstruct both normal and fraudulent transactions too effectively, limiting their discriminative capacity compared to QAE-based models that explicitly leverage quantum state fidelity for anomaly scoring.

A similar pattern is observed on the KDD99 intrusion dataset. Classical models such as Random Forests, SVMs, Naive Bayes, ANN/MLP, and C4.5 decision trees routinely achieve accuracy above 90% and high recall, reflecting the relative separability of KDD99's feature space [4, 17, 29, 45, 47]. While these methods remain strong baselines, they rely on handcrafted features, assume relatively low-dimensional structure, and do not offer natural mechanisms for feature compression or latent-space representation learning. Furthermore, their high accuracy does not necessarily translate into improved generalisation to more modern intrusion datasets with higher complexity and variability.

Taken together, these comparisons highlight two important insights. First, classical ML models excel when abundant labelled data are available and the feature space is well structured, but their performance deteriorates in unsupervised or extremely imbalanced settings. Second, classical autoencoders offer a useful baseline for representation learning but lack robustness against overlapping class distributions. In contrast, QAE-based models, particularly QAE-FD and hybrid QAE–classifier pipelines, demonstrate competitive or superior anomaly-detection capabilities under these challenging conditions. Quantum feature spaces introduce non-classical correlations and higher representational capacity, enabling improved separation of anomalies despite limited parameters and shallow circuits. However, classical models still outperform quantum ones in scalability and maturity, underscoring the continued need for hybrid architectures and careful benchmarking when evaluating quantum advantages.

1.5 Review of literature gaps

Despite rapid progress, several important gaps remain across all published work:

1. **Overfitting is not addressed.** No studies apply L2 weight decay, dropout, gate-drop, or latent regularization. Variational circuits are known to overfit when highly expressive.

2. **Generalization on normal samples is largely ignored.** Most evaluations focus on anomalies or

attacks. False-positive rates, reconstruction stability, and latent consistency on genuine samples remain unexamined.

3. **Latent-space noise sensitivity and denoising strategies are missing.** No paper introduces latent denoising, smoothing modules, or noise-aware latent modelling, despite quantum circuits being inherently noisy.

4. **Scalability of QAE architectures is unexplored.** Existing work fixes depth, ansatz type, latent dimension, and qubit count; no systematic studies analyze scaling laws or architecture–performance relationships.

5. **Evaluation metrics are incomplete and biased toward anomaly performance.** Existing studies overwhelmingly report metrics such as AUC, accuracy, or F1-score computed only on anomalous or attack-class samples. This neglects the equally critical question of how well QAEs reconstruct and classify normal data. Key metrics such as false-positive rate, reconstruction variance on clean samples, and latent-space stability are rarely reported, making the generalization capability of existing models unclear.

Based on the limitations identified in existing work on quantum autoencoders and quantum anomaly detection, this thesis is guided by the following research questions:

1. **RQ1:** How do architectural factors such as qubit count, circuit depth, number of variational layers, and latent bottleneck size affect the robustness and generalization behaviour of quantum autoencoders for anomaly detection?

2. **RQ2:** How effective are quantum-compressed latent representations for downstream anomaly detection when evaluated using classical kNN and quantum kNN classifiers?

3. **RQ3:** Can latent-space processing and regularization techniques, such as the proposed BrainBox layer, dropout, and ℓ_2 regularization, improve the stability, interpretability, and generalization of quantum autoencoders?

2 Fundamentals of Quantum Computing

Quantum computing provides a computational paradigm based on the principles of quantum mechanics, enabling information processing using quantum states instead of classical bits. This section introduces the essential components required to understand variational quantum circuits, quantum autoencoders, and quantum distance-based classifiers.

2.1 Qubits and Quantum States

A qubit is the fundamental unit of quantum information. Unlike a classical bit, which can be either 0 or 1, a qubit can exist in a superposition of both states simultaneously:

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle\,, \tag{1}$$

where $\alpha, \beta \in \mathbb{C}$ and satisfy $|\alpha|^2 + |\beta|^2 = 1$. The ability to encode information into superpositions provides a richer representational space than classical bits, enabling compact embeddings in quantum machine learning.

2.2 Quantum Gates and Unitary Evolution

The evolution of quantum states is governed by unitary operators. Single-qubit gates, such as the Pauli matrices, the Hadamard gate, and rotation gates, manipulate the state on the Bloch sphere:

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \tag{2}$$

$$R_y(\theta) = \begin{pmatrix} \cos(\theta/2) & -\sin(\theta/2) \\ \sin(\theta/2) & \cos(\theta/2) \end{pmatrix}. \tag{3}$$

Multi-qubit entangling gates, such as CNOT, enable correlations that cannot be represented classically and are critical for variational circuits and QAE architectures.

2.3 Measurement and Probabilistic Outcomes

Measurement collapses a quantum state onto the computational basis. Given a state $|\psi\rangle$, measuring in the Z basis yields outcome $|0\rangle$ with probability $|\alpha|^2$ and $|1\rangle$ with probability $|\beta|^2$. Because measurement is inherently probabilistic, repeated sampling (shots) is required in quantum machine learning pipelines to estimate expectation values, fidelities, and loss functions.

These probabilistic effects influence downstream components such as QAE reconstruction metrics and QkNN distance estimation.

2.4 Noise and Real Quantum Hardware

Near-term quantum devices (NISQ hardware) are affected by several noise sources:

- **Decoherence:** loss of quantum information over time.
- **Gate noise:** imperfect unitary operations.
- **Readout noise:** measurement errors.
- **Crosstalk:** unintended interactions between qubits.

Noise is often modeled using channels such as depolarizing noise, amplitude damping, and phase damping. These effects degrade the fidelity of reconstructed states in QAEs and introduce variability in latent-space representations. Many anomaly detection studies—including QAE-FD—evaluate models under noisy IBM Fake backends to benchmark robustness.

2.5 Parameterized Quantum Circuits and Optimization

Parameterized quantum circuits (PQCs) form the backbone of many variational quantum algorithms and quantum machine-learning models [43]. They consist of single-qubit rotations with tunable parameters, interleaved with entangling gates. Such circuits are optimized using classical gradient-based or gradient-free methods, an approach commonly referred to as the variational quantum eigensolver (VQE) framework [35].

A known challenge in training PQCs is the presence of barren plateaus—regions of vanishing gradients that make optimization increasingly difficult as circuit depth grows [28]. This motivates the use of shallow, hardware-efficient ansätze in quantum autoencoders and quantum classifiers.

A typical PQC is made up of two types of layers:

- **Rotation layers:** single-qubit gates such as $R_x(\theta)$ or $R_y(\theta)$ that depend on a parameter θ. These gates act like trainable weights.
- **Entangling layers:** gates such as CNOT that create correlations between qubits and give the circuit expressive power.

These layers are stacked to form a variational circuit. During training, the parameters are updated by a classical optimizer to minimize a cost function obtained from measurement outcomes. In this way, PQCs serve a similar role to neural networks in classical machine learning, enabling quantum models such as quantum autoencoders and quantum classifiers to learn from data.

Training a parameterized quantum circuit requires adjusting its rotation angles so that the circuit minimizes a chosen loss function, typically derived from measurement outcomes. Because PQCs are differentiable with respect to their parameters, gradient-based methods form the foundation of most optimization strategies used in variational quantum algorithms.

A widely adopted technique for computing gradients on quantum hardware is the *parameter-shift rule* [41]. Instead of relying on symbolic differentiation, the parameter-shift rule expresses the

derivative of an expectation value with respect to a circuit parameter as the difference of two forward evaluations of the circuit:

$$\frac{\partial}{\partial\theta}\langle O\rangle = \frac{1}{2}\left[\langle O\rangle_{\theta+\frac{\pi}{2}} - \langle O\rangle_{\theta-\frac{\pi}{2}}\right]. \tag{4}$$

where O is an observable associated with the loss. This enables unbiased gradient estimation using only quantum measurements and avoids the need for backpropagation through quantum gates.

In addition to gradient-based approaches, several gradient-free optimizers—such as COBYLA, Nelder–Mead, or SPSA—are commonly used in quantum machine learning when measurement noise or limited sampling makes gradient estimates unreliable [46]. These methods operate directly on function evaluations and can sometimes navigate noisy loss landscapes more effectively.

Despite the availability of these optimization tools, training PQCs remains challenging. One of the main obstacles is the barren plateau phenomenon, where gradients vanish exponentially with system size or circuit depth, making optimization prohibitively difficult [28]. To mitigate this, practitioners employ shallow, hardware-efficient ansätze, careful initialization schemes, or local cost functions that depend on only a small subset of qubits.

Overall, the choice of optimizer and gradient-estimation method plays a significant role in the performance and stability of quantum autoencoders and related variational models. These considerations are particularly important for the architectures explored in this thesis, which must balance expressiveness with trainability under realistic noise conditions.

2.6 Data Encoding Strategies

Quantum machine learning models require classical input data to be mapped into quantum states. This process, known as data encoding or embedding, determines how efficiently information is represented and strongly influences the performance of quantum autoencoders and quantum classifiers. Several encoding strategies are commonly used in the literature.

2.6.1 Angle Encoding

Angle encoding maps classical features into the rotation angles of single-qubit gates. Given a feature vector $x = (x_1, x_2, \ldots, x_d)$, each value is encoded as a rotation such as:

$$R_y(x_i)\,|0\rangle\,. \tag{5}$$

Angle encoding is widely used because it is easy to implement on NISQ hardware, requires only one qubit per feature, and results in shallow circuits. However, it cannot represent all 2^n amplitudes of an n-qubit state, which limits expressiveness for high-dimensional data.

2.6.2 Amplitude Encoding

Amplitude encoding embeds a normalized classical vector directly into the amplitudes of a quantum state:

$$|x\rangle = \sum_{i=0}^{2^n-1} x_i \, |i\rangle \, . \tag{6}$$

This method is extremely compact: n qubits can represent a vector of size 2^n. Such compression is attractive for quantum autoencoders, which benefit from processing high-dimensional inputs with relatively few qubits. However, preparing arbitrary amplitude-encoded states typically requires deeper circuits and is therefore more sensitive to hardware noise. In practice, amplitude encoding is most commonly used in simulations or when efficient state-preparation techniques are available.

2.6.3 Basis Encoding

Basis encoding assigns each classical value to a computational basis state. For categorical or binary data, this is straightforward:

$$x \in \{0,1\}^n \quad \rightarrow \quad |x\rangle \, . \tag{7}$$

This encoding is simple and noise-resilient but not efficient for real-valued data, and it lacks the expressive richness needed for dimensionality reduction tasks. Thus, it is rarely used in QAE models.

Example: Encoding a Simple Feature Vector

To give a concrete illustration of how different encoding choices represent data, consider a two-dimensional feature vector:

$$x = (0.2, 0.7).$$

Although the vector is simple, the contrast between encoding methods is already apparent.

Angle Encoding.

Angle encoding treats each feature independently. The value of each component determines the rotation applied to its corresponding qubit:

$$|\psi_{\text{angle}}\rangle = R_y(0.2)\,|0\rangle \;\otimes\; R_y(0.7)\,|0\rangle \, .$$

This construction is easy to implement and produces a shallow circuit, making it appealing for hardware experiments. The interpretability is also intuitive: the magnitude of each feature corresponds to how far its qubit is rotated on the Bloch sphere. The drawback is that no dimensionality reduction is achieved—two features require two qubits.

Amplitude Encoding.

Amplitude encoding embeds the entire vector into the amplitudes of a single qubit. The first step is normalization:

$$\|x\| = \sqrt{0.2^2 + 0.7^2} = \sqrt{0.53}, \qquad x' = \left(\frac{0.2}{\sqrt{0.53}}, \frac{0.7}{\sqrt{0.53}} \right) .$$

Using these normalized values as amplitudes yields:

$$|\psi_{\text{amp}}\rangle = x'_1\,|0\rangle + x'_2\,|1\rangle \approx 0.2747\,|0\rangle + 0.9615\,|1\rangle \, .$$

In this case, both features are represented using just one qubit. This demonstrates the exponential compression capability of amplitude encoding, which is one of the reasons it is frequently considered for quantum autoencoder architectures. The trade-off is that preparing such states generally requires deeper circuits, which are more vulnerable to noise.

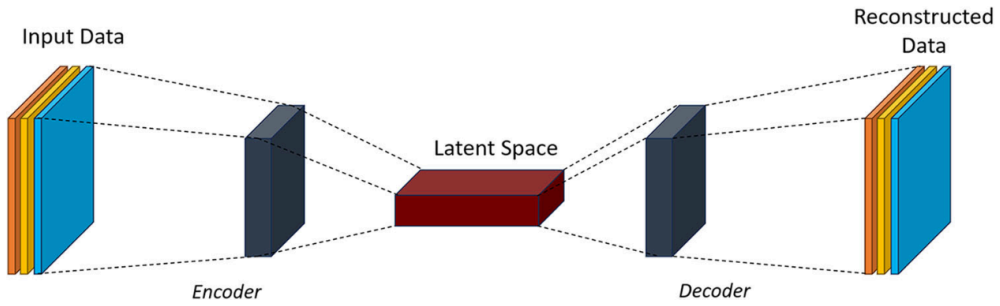
Because quantum autoencoders rely on reconstructing encoded states, the choice of embedding significantly affects reconstruction fidelity, latent-space structure, and training stability. Models based on amplitude encoding often achieve better compression, while angle encoding is preferred for hardware experiments due to lower noise sensitivity.

3 Classical and Quantum Machine Learning

Autoencoders are a family of neural network models designed to learn compressed representations of data. They achieve this by mapping an input to a low-dimensional latent space and then reconstructing it as closely as possible. Although originally introduced in the context of classical deep learning, the same general idea carries over into the quantum setting. Quantum autoencoders (QAEs) adapt the encoder–decoder structure to parameterized quantum circuits, enabling compression and reconstruction of quantum states or embedded classical data. This section provides an overview of both models and highlights the key differences that motivate the use of QAEs in anomaly detection.

3.1 Classical Autoencoders

Classical autoencoders are neural network models designed to learn compact representations of data by reconstructing inputs from a lower-dimensional latent space. The basic idea dates back to early work on dimensionality reduction using neural networks [18], where autoencoders were shown to behave similarly to principal component analysis (PCA) when trained with linear layers and mean squared error loss. Modern variants extend this idea with nonlinear activation functions and deeper architectures, enabling autoencoders to capture highly structured and non-linear patterns in data [12].



1 figure. *Classical autoencoder architecture consisting of an encoder, a low-dimensional latent space, and a decoder for reconstruction. Reproduced from [15].*

As illustrated in Figure 1 figure., a classical autoencoder consists of two main components: an *encoder* that compresses the input into a low-dimensional latent vector, and a *decoder* that attempts to reconstruct the original input from this compressed representation. Training typically involves minimizing a reconstruction loss such as the mean squared error,

$$\mathcal{L}(x, \hat{x}) = \|x - \hat{x}\|_2^2, \tag{8}$$

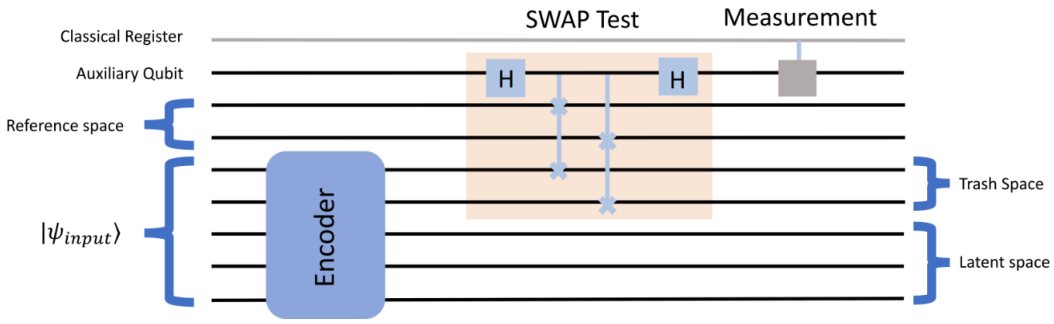
or a cross-entropy loss when dealing with normalized or binary data. By forcing information through this bottleneck, the model learns to retain only the salient features needed to describe the data distribution.

Because autoencoders are trained on normal data, they naturally lend themselves to anomaly detection: samples that deviate from the learned manifold tend to produce higher reconstruction errors. This principle is widely used in areas such as fraud detection, intrusion detection, and time-series monitoring [31, 40]. In addition, classical autoencoders support useful variations such as denoising autoencoders [49], which introduce noise during training to encourage robustness and smoother latent representations.

The latent space learned by an autoencoder often reveals meaningful geometric structure. Inputs with similar characteristics tend to cluster together, making the latent representation a convenient basis for downstream tasks such as clustering or nearest-neighbour classification. This structured compression is also one of the motivations for developing quantum autoencoders, which adopt the same encoder–decoder concept but implement it using parameterized quantum circuits. The expectation is that quantum models may exploit properties of quantum states—such as superposition and entanglement—to achieve more expressive or compact representations than their classical counterparts [39].

3.2 Quantum Autoencoder Architecture

The quantum autoencoder (QAE) follows the same high–level idea as a classical autoencoder, but it operates directly on quantum states and uses parameterized quantum circuits in place of neural network layers. The goal is to compress an input quantum state into a lower–dimensional subspace while preserving the information that is most relevant for reconstruction. The general architecture, illustrated in Figure 2 figure., mirrors the encoder–latent–decoder structure of classical models, but with components adapted to the constraints and capabilities of quantum computation [38, 39].



2 figure. *Circuit-level structure of a quantum autoencoder, including the encoder, latent space, trash qubits, and SWAP-test fidelity estimation. Adapted from [38].*

A typical QAE takes as input a quantum state $|\psi_{\text{input}}\rangle$ that may either originate from a physical quantum process or from encoding classical data into qubits. The *encoder* is a variational quantum circuit whose parameters are optimized during training. Its role is to transform the input state so that only a subset of qubits—the latent space—contains useful information [39]. The remaining qubits, often called *trash qubits*, ideally end up in a fixed reference state such as $|0\rangle$ after the encoding transformation.

To evaluate whether the encoder has successfully compressed the input state, the QAE uses a fidelity estimation procedure based on the SWAP test, a standard quantum subroutine for comparing two quantum states. As illustrated in Figure 2 figure., an auxiliary qubit is prepared in the $|0\rangle$ state, acted upon by a Hadamard gate, and then used to control SWAP operations between the output trash space and a fixed reference state. A final Hadamard followed by measurement on the auxiliary qubit yields an outcome distribution whose bias toward the $|0\rangle$ result is directly proportional to the overlap between the two states. A higher probability of measuring $|0\rangle$ indicates that the trash qubits have been successfully mapped to the desired reference state and that the encoder has preserved the relevant information in the latent qubits.

Once the encoder has been optimized so that the trash space reliably collapses to the reference state, the decoder reconstructs the input by acting on the latent qubits together with freshly re-initialized ancillary qubits prepared in $|0\rangle^{\otimes m}$. In many designs the decoder is implemented as the inverse of the encoder, $U_{\text{dec}} = U_{\text{enc}}^{\dagger}$, ensuring that lossless compression can be perfectly reversed. The decoding step aims to recover a state $|\psi_{\text{rec}}\rangle$ that closely approximates the original input, which can be written as

$$U_{\text{dec}}\big(|\psi_{\text{latent}}\rangle \otimes |0\rangle^{\otimes m}\big) \approx |\psi_{\text{input}}\rangle\,. \tag{9}$$

The quality of this reconstruction is quantified using the fidelity

$$F = |\langle \psi_{\text{input}} \mid \psi_{\text{rec}} \rangle|^2\,. \tag{10}$$

which plays an analogous role to reconstruction error in classical autoencoders, but is computed through quantum measurements [39]. The training objective of a quantum autoencoder is to ensure that the compressed state retains all information necessary for accurate reconstruction. Instead of relying on classical loss functions such as mean squared error, QAEs use a fidelity-based loss that measures how closely the reconstructed state matches the original input [39]. Because higher fidelity corresponds to better reconstruction, the optimization problem is typically formulated as minimizing the loss

$$\mathcal{L} = 1 - F. \tag{11}$$

which takes values near zero when the autoencoder successfully preserves the essential structure of the input. This loss function plays an analogous role to reconstruction error in classical autoencoders but is computed through quantum measurements, often via the SWAP test.

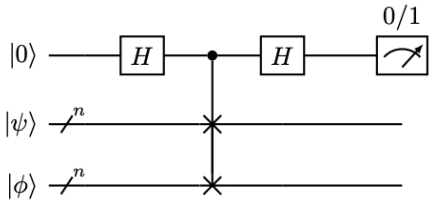
3.3 Quantum Distance Measures and Quantum kNN

Distance-based methods play an important role in classification and anomaly detection, and the same principle extends naturally into the quantum setting. For pure quantum states $|\psi\rangle$ and $|\phi\rangle$, the similarity between them is quantified by the fidelity

$$F(\psi,\phi) = |\langle \psi \mid \phi \rangle|^2,$$

which takes values close to one when the states are nearly identical and approaches zero as they become orthogonal. Fidelity therefore acts as a quantum analogue of classical distance measures such as cosine similarity or Euclidean distance.

Estimating fidelity on quantum hardware requires comparing two quantum states without performing full state tomography. A standard approach is the SWAP test, which measures the overlap between two states using an auxiliary qubit and a controlled-SWAP operation. As illustrated in Figure 3 figure., the auxiliary qubit is prepared in $|0\rangle$, subjected to a Hadamard gate, and used to control a SWAP operation between the two states of interest. After a second Hadamard gate, the auxiliary qubit is measured. The probability of obtaining the outcome 0 provides an estimator for the fidelity $F(\psi,\phi)$, making the SWAP test a key primitive for quantum similarity evaluation.



3 figure. *SWAP-test circuit used to estimate the fidelity between two quantum states $|\psi\rangle$ and $|\phi\rangle$. The auxiliary qubit controls the SWAP operation, and the measurement outcome encodes the similarity between the states. Adapted from Basheer et al. [2].*

Building on this idea, Basheer et al. proposed a quantum k-nearest neighbours (QkNN) algorithm that employs fidelity-based distance measures for classification [2]. In their formulation, each training sample is encoded into a quantum state, and the fidelity between a query state and each stored sample is estimated using a SWAP-test routine. The k states with the highest fidelity—equivalently, the smallest effective distance—are selected, and the predicted label is determined through a majority vote among these nearest neighbours. Although conceptually similar to classical kNN, the quantum version can leverage compact quantum representations and coherence-based similarity measures, which may offer advantages in high-dimensional feature spaces.

In this thesis, fidelity-based distances play a similar role: latent states produced by the quantum autoencoder are compared either through classical kNN or via a quantum-inspired similarity measure derived from overlap estimation. This allows for a direct comparison between classical and quantum distance-based classifiers operating in the compressed feature space learned by the autoencoder.

3.4 Noise and NISQ Hardware Considerations

Current quantum devices operate in the noisy intermediate-scale quantum (NISQ) regime, characterized by limited qubit numbers, short coherence times, and imperfect gate implementations [37]. These hardware constraints have a direct influence on the design and performance of quantum machine-learning models, particularly those based on variational circuits such as quantum autoencoders and quantum classifiers.

Noise in NISQ systems arises from several sources: decoherence processes (including amplitude and phase damping), gate errors introduced by imperfect control pulses, and measurement

errors that distort the observed probability distribution [30]. As the depth of a quantum circuit increases, these errors accumulate, degrading the fidelity of intermediate states and ultimately reducing the reliability of any reconstructed quantum state. For models like quantum autoencoders, which depend on accurate state preparation and reconstruction, the impact of noise can be substantial; even small perturbations can alter the trash-space alignment or distort the learned latent representations.

Variational quantum circuits are especially sensitive to noise during training. Since the loss function is estimated through repeated measurements, statistical fluctuations and readout errors introduce variance into gradient estimates and can destabilize the optimization process. In addition, noise can flatten the loss landscape, exacerbating the barren plateau phenomenon [28], where gradients vanish exponentially with system size or circuit depth. Subsequent studies have shown that hardware noise can induce barren plateaus even in relatively shallow circuits [50], further complicating optimization in practical settings.

These limitations motivate the use of hardware-efficient ansätze—circuits designed to be expressive while minimizing depth and entanglement overhead [22]. Shallow circuits help reduce decoherence accumulation and improve trainability. Noise-aware initialization strategies and local cost functions can also provide more stable learning dynamics, though they do not eliminate the impact of noise entirely.

To counteract measurement errors, several mitigation techniques have been proposed, including simple readout-error calibration routines and extrapolation-based correction methods [48]. While these methods are approximate, they can meaningfully improve effective fidelity and reduce noise bias, which is especially important for reconstruction-based models like quantum autoencoders.

In the context of this thesis, noise considerations motivate the use of shallow architectures, regularization strategies, and hardware-efficient circuit layouts. Because the quality of the learned latent representations depends directly on the stability of the underlying quantum operations, careful management of NISQ constraints is essential for achieving reliable and generalizable performance in QAE-based models.

4 Methodology

4.1 Overview of the Experimental Workflow

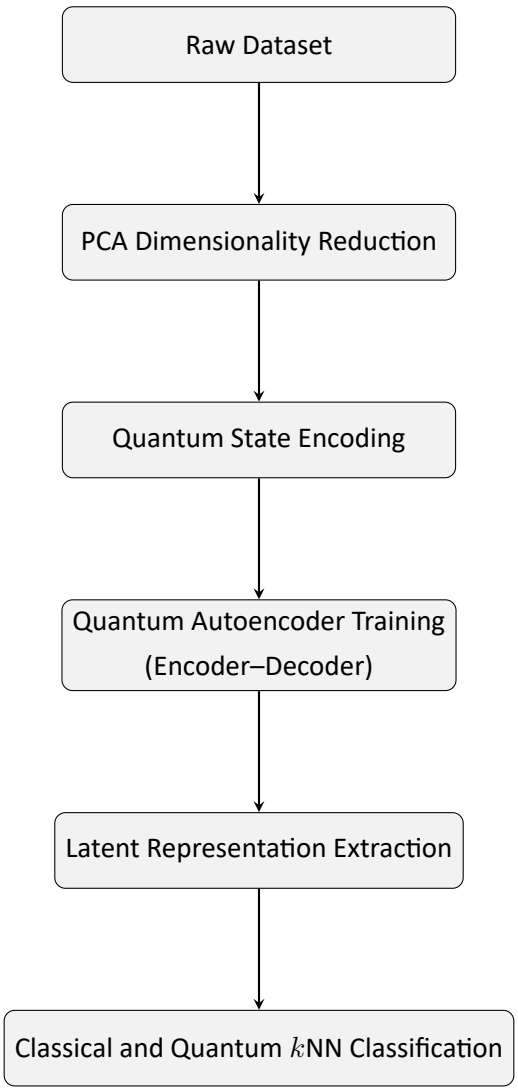
The goal of this work is to investigate the representation-learning capabilities of quantum autoencoders (QAEs) and to evaluate how well their latent embeddings support downstream classification using both classical and quantum-inspired techniques. The overall workflow consists of three main stages: (i) preprocessing and dimensionality reduction of the input data, (ii) training a QAE to learn a compressed latent representation of normal samples, and (iii) applying classical and quantum k-nearest neighbour classifiers to the resulting latent space.

The first stage applies principal component analysis (PCA) to the original dataset to reduce its dimensionality and make it compatible with the number of available qubits. PCA serves two purposes: it acts as a denoising and whitening transformation, and it ensures that the final feature dimension matches the encoding capacity of the chosen quantum circuit.

In the second stage, the reduced feature vectors are encoded into quantum states and used to train a quantum autoencoder. The QAE learns to compress the input data into a low-dimensional latent subspace while mapping redundant information into a trash space. Training is performed using a variational optimization procedure that minimizes a fidelity-based loss function. Architectural parameters such as the number of qubits, circuit depth, and latent dimension are varied systematically to study their influence on generalization and robustness.

In the final stage, the latent vectors extracted from the trained QAE are passed to two distance-based classifiers. The first is a classical k-nearest neighbour (kNN) classifier operating on the real-valued latent embeddings. The second is a quantum-inspired kNN method that uses fidelity estimates between quantum states as a similarity measure. This dual evaluation provides insight into how well the QAE organizes the data in latent space and whether the quantum representations offer advantages over purely classical embeddings.

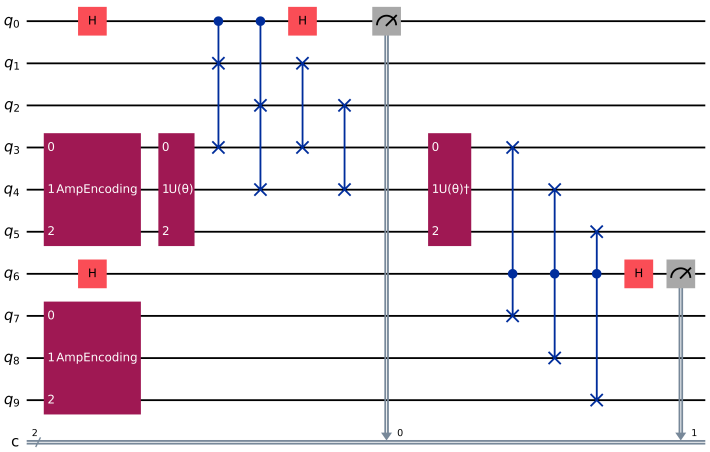
The sequence of processing steps used in this study—from PCA preprocessing to quantum autoencoding and final classification—is summarized in 4 figure..



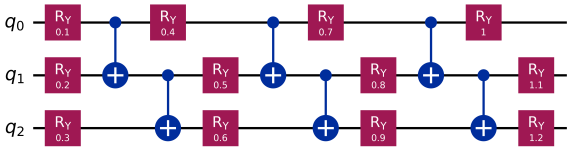
4 figure. *Overview of the experimental pipeline. The dataset is first reduced using PCA, then encoded into quantum states and passed through a quantum autoencoder. The resulting latent representations are evaluated using both classical and quantum-inspired kNN classifiers.*

4.2 Architecture of Quantum Autoencoder

5 figure. illustrates the complete quantum autoencoder (QAE) used in this thesis, including the encoding process, the variational compression circuit $U(\theta)$, the trash-space SWAP test, the decoder $U(\theta)^\dagger$, and the final reconstruction fidelity test. The circuit consists of three logical subsystems: the data register, the reference register, and auxiliary qubits used for fidelity estimation.



(a) Full QAE encoder–decoder circuit



(b) Variational block $U(\theta)$ with `3 layers`

5 figure. *Quantum autoencoder architecture and its variational building block. Panel (a) shows the full three–qubit QAE circuit including amplitude encoding, encoder $U(\theta)$, trash SWAP test, decoder $U(\theta)^\dagger$ and reconstruction SWAP test. Panel (b) shows the internal structure of the three–qubit variational ansatz $U(\theta)$ used in both encoder and decoder.*

We use the following notation for the qubit registers appearing in the full QAE architecture shown in 5 figure.:

- ψ_{en}: the *encoding register* consisting of qubits (q_3, q_4, q_5), which hold the amplitude-encoded input state $|\psi_{\text{in}}\rangle$.
- ψ_{lat}: the *latent-space subsystem*, represented by qubits (q_4, q_5) after the action of the variational encoder $U_{\text{enc}}(\boldsymbol{\theta})$.
- ψ_{trash}: the *trash qubit* q_3, which the encoder attempts to map to a fixed reference state (typically $|0\rangle$) so that it carries no remaining information after compression.
- ψ_{ref}: the *reference register* used in both SWAP tests. In the compression stage, it is a single qubit initialized to $|0\rangle$. In the reconstruction stage, it is a three-qubit register (q_7, q_8, q_9) prepared via amplitude encoding to hold a clean copy of $|\psi_{\text{in}}\rangle$.
- q_{aux}: the *auxiliary control qubits* used to perform the SWAP tests. The encoder SWAP test uses qubit q_0, while the reconstruction SWAP test uses qubit q_6.
- ψ_{dec}: the *decoder output register*, given by the three qubits (q_3, q_4, q_5) after the application of the decoder $U_{\text{dec}} = U_{\text{enc}}^\dagger$. These qubits carry the reconstructed state $|\psi_{\text{rec}}\rangle$.

4.2.1 Encoder Stage

On the left side of the circuit, the PCA-reduced input vector is loaded into three data qubits via amplitude encoding:

$$\mathbf{x} \longrightarrow |\psi_{\text{en}}\rangle = \sum_{i=0}^{7} x_i |i\rangle , \tag{12}$$

where the amplitudes correspond directly to the preprocessed classical features.

A variational encoder

$$U_{\text{enc}}(\boldsymbol{\theta}), \tag{13}$$

acts on these qubits. Its objective is to reorganize the information so that only the first two qubits carry meaningful structure (the latent representation), while the third qubit becomes a "trash" qubit ideally placed in a fixed reference state. This corresponds to the desired factorization:

$$U_{\text{enc}}(\boldsymbol{\theta}) |\psi_{\text{in}}\rangle \approx |\psi_{\text{lat}}\rangle_{L_1 L_2} \otimes |0\rangle_T . \tag{14}$$

To verify that the trash qubit has been successfully disentangled from the latent subsystem, a SWAP test is performed between the trash qubit and a reference qubit initialized in $|0\rangle$. The probability of the auxiliary qubit being measured in 0 yields an estimate of the trash fidelity:

$$F_{\text{trash}} = \langle 0| \rho_T |0\rangle, \tag{15}$$

where ρ_T is the reduced density matrix of the trash qubit. A high value of F_{trash} indicates successful compression.

4.2.2 Decoder Stage

The right side of the circuit mirrors the encoder and attempts to reconstruct the original three-qubit input state. The two latent qubits are combined with a fresh ancillary qubit initialized as $|0\rangle$, and a parametrized decoder

$$U_{\text{dec}}(\boldsymbol{\phi}) \tag{16}$$

acts on them. Often, the decoder is constructed to approximate the inverse of the encoder:

$$U_{\text{dec}}(\boldsymbol{\phi}) \approx U_{\text{enc}}^{\dagger}(\boldsymbol{\theta}). \tag{17}$$

Its goal is to recover the original encoded state:

$$U_{\text{dec}}(\boldsymbol{\phi}) \left(|\psi_{\text{lat}}\rangle \otimes |0\rangle\right) \approx |\psi_{\text{rec}}\rangle . \tag{18}$$

To evaluate reconstruction quality, a second copy of the input state is re-prepared via amplitude encoding, and a second SWAP test is performed between this reference state and the reconstructed

state. The resulting overlap is the reconstruction fidelity:

$$F_{\text{rec}} = |\langle \psi_{\text{in}} \mid \psi_{\text{rec}} \rangle|^2 . \tag{19}$$

This fidelity serves as the training objective of the QAE: the circuit parameters are optimized to maximize F_{rec}, encouraging the encoder to perform lossless compression and the decoder to accurately reconstruct the input.

4.3 Training Objective and Loss Function

The quantum autoencoder is trained to satisfy two goals simultaneously: (i) the trash qubit should be mapped as closely as possible to a fixed reference state, and (ii) the reconstructed state should match the original input state with high fidelity. Both quantities are estimated using SWAP tests as described in the Background chapter.

The first objective is quantified by the *trash fidelity* F_{trash}, which measures how close the trash qubit ψ_{trash} is to the reference state (typically $|0\rangle$):

$$F_{\text{trash}} = \langle 0 \mid \rho_{\text{trash}} \mid 0 \rangle, \tag{20}$$

where ρ_{trash} denotes the reduced density matrix of the trash qubit after encoding.

The second objective is the *reconstruction fidelity* F_{rec}, which compares the input state on the encoding register ψ_{en} with the reconstructed state on the decoder register ψ_{dec}:

$$F_{\text{rec}} = |\langle \psi_{\text{in}} \mid \psi_{\text{rec}} \rangle|^2 . \tag{21}$$

To train the model, these two fidelities are combined into a single scalar objective. We maximize the average of both fidelities, which encourages the encoder to produce a clean trash state and the decoder to accurately reconstruct the input. In practice, this is implemented via a loss function defined as

$$\mathcal{L} = 1 - \frac{1}{2} \left(F_{\text{trash}} + F_{\text{rec}} \right). \tag{22}$$

Minimizing $\mathcal{L}$ is equivalent to maximizing the sum of trash and reconstruction fidelities. When both fidelities are close to one, the loss approaches zero, indicating that the QAE is simultaneously achieving effective compression and faithful reconstruction.

4.3.1 L2 Regularization

To improve generalization and reduce overfitting in the variational encoder, we extend the previously defined loss function—based on the trash fidelity F_{trash} and reconstruction fidelity F_{rec}—with an L2 penalty applied to the encoder parameters. As discussed in the Background section, variational circuits are prone to memorization, yet none of the existing QAE studies incorporate regularization. Introducing L2 weight decay therefore represents a novel contribution of this work.

Let $\boldsymbol{\theta}$ denote the trainable parameters of the encoder. The L2 term is

$$R_{\text{L2}} = \|\boldsymbol{\theta}\|_2^2. \tag{23}$$

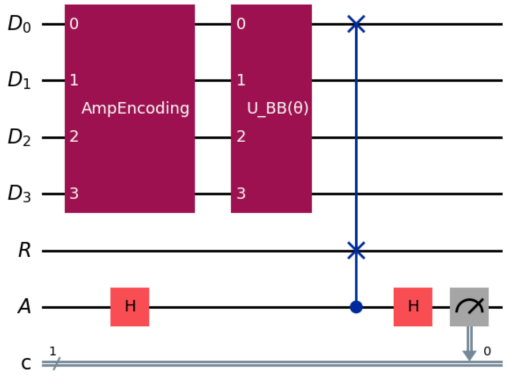
The overall training objective is obtained by adding this regularization term to the original loss introduced above:

$$\mathcal{L}_{\text{reg}} = \mathcal{L} + \lambda R_{\text{L2}}, \tag{24}$$

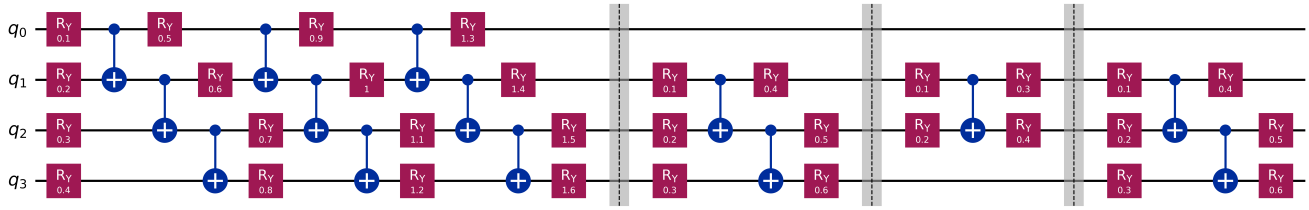
where $\mathcal{L}$ is the fidelity-based loss previously defined and λ is the regularization strength. This modification encourages smoother parameter landscapes and prevents individual parameters from growing excessively large, thereby promoting more robust latent representations and improved behaviour on normal (non-anomalous) samples.

4.3.2 BrainBox Latent-Space Layer

Beyond the baseline quantum autoencoder, this thesis investigates a modified architecture inspired by the BrainBox idea proposed in [33], in which an additional variational block is inserted directly in the latent space to mitigate errors and smooth the learned representation. The goal of this BrainBox layer is to act as a latent-space denoiser: it processes only the latent qubits, leaving the overall encoder–decoder structure unchanged, but potentially improving robustness and generalization. The resulting BrainBox-enhanced encoder and the internal 3–2–3 structure of $U_{\text{BB}}(\theta)$ are shown in 6 figure..



(a) *Outer BrainBox encoder with amplitude encoding, $U_{\mathrm{BB}}(\theta)$ on the data qubits, and SWAP test on the trash qubit.*



(b) *Internal 3–2–3 structure of $U_{\mathrm{BB}}(\theta)$: encoder on four data qubits followed by 3-qubit, 2-qubit, and final 3-qubit BrainBox blocks acting on the latent with* `4 layers`.

6 figure. *BrainBox-enhanced quantum autoencoder.* ***(a)*** *The PCA-reduced input is amplitude-encoded on the data register and processed by the BrainBox unitary $U_{\mathrm{BB}}(\theta)$ before a SWAP test evaluates the trash qubit.* ***(b)*** *Inside $U_{\mathrm{BB}}(\theta)$, a 3–2–3 sequence of shallow variational blocks acts on the latent qubits to mix, compress, and denoise the latent representation.*

In the four-qubit variant used here, the local layout of the encoder register is $\psi_{\mathrm{en}} = (q_{\mathrm{trash}}, q_{\mathrm{lat},0}, q_{\mathrm{lat},1}, q_{\mathrm{lat},2})$, where one qubit plays the role of the trash qubit and three qubits form an extended latent subsystem. In the baseline model, this entire register is parameterized by a single hardware-efficient RealAmplitudes circuit acting on all four qubits. In the BrainBox version, this is replaced by a structured composition of several smaller variational blocks, all acting locally on the latent qubits:

- an encoder block $U_{\mathrm{enc}}(\boldsymbol{\theta}_{\mathrm{enc}})$ acting on all four data qubits $(q_{\mathrm{trash}}, q_{\mathrm{lat},0}, q_{\mathrm{lat},1}, q_{\mathrm{lat},2})$;
- a three-qubit BrainBox block $U^{(a)}_{\mathrm{bb3}}(\boldsymbol{\theta}_{\mathrm{bb3,a}})$ acting only on the latent triple $(q_{\mathrm{lat},0}, q_{\mathrm{lat},1}, q_{\mathrm{lat},2})$;
- a two-qubit BrainBox core block $U_{\mathrm{bb2}}(\boldsymbol{\theta}_{\mathrm{bb2}})$ acting on the central pair $(q_{\mathrm{lat},0}, q_{\mathrm{lat},1})$;
- a second three-qubit BrainBox block $U^{(b)}_{\mathrm{bb3}}(\boldsymbol{\theta}_{\mathrm{bb3,b}})$ again acting on $(q_{\mathrm{lat},0}, q_{\mathrm{lat},1}, q_{\mathrm{lat},2})$.

The resulting local transformation on the encoder register can be written as

$$U_{\mathrm{BB}}(\boldsymbol{\theta}) = U^{(b)}_{\mathrm{bb3}} \circ U_{\mathrm{bb2}} \circ U^{(a)}_{\mathrm{bb3}} \circ U_{\mathrm{enc}}, \tag{25}$$

where $\boldsymbol{\theta}$ collects all parameters from the encoder and BrainBox blocks. Each of these components

is implemented as a shallow RealAmplitudes circuit with linear entanglement, so the overall depth remains moderate while increasing expressivity in the latent subspace.

The 3–2–3 structure of the BrainBox layer has a natural interpretation. The first three-qubit block $U_{\text{bb3}}^{(a)}$ mixes information across the extended latent subsystem, allowing correlations between all three latent qubits. The subsequent two-qubit block U_{bb2} focuses on the core latent pair $(q_{\text{lat},0}, q_{\text{lat},1})$, which ultimately defines the effective latent representation used for downstream classification. The final three-qubit block $U_{\text{bb3}}^{(b)}$ remixes the latent triple, using the third qubit as an auxiliary degree of freedom that can absorb noise or redundant components before the state is passed to the decoder.

Training of the BrainBox-enhanced autoencoder uses the same fidelity-based loss as the baseline model, combining trash fidelity and reconstruction fidelity into a single objective (and optionally augmented with L2 regularization as described above). The BrainBox parameters are optimized jointly with the encoder parameters. By comparing the baseline architecture with its BrainBox variant, this thesis empirically evaluates whether inserting a local 3–2–3 denoising layer in the latent space leads to more stable latent representations and improved generalization, particularly on normal samples.

4.3.3 Layer Dropout and Gate Dropout in the BrainBox Block

To further reduce overfitting and improve the robustness of the latent representation, this thesis introduces a dropout mechanism applied directly to the variational BrainBox block. The idea is analogous to dropout in classical neural networks: during training, parts of the model are randomly disabled so that the learned representation does not rely too heavily on any particular set of entangling operations. In our case, dropout is applied only to the entangling gates inside the latent BrainBox ansatz, while the main encoder acting on the data qubits remains deterministic.

In this configuration the data register consists of three qubits $(\text{TRASH}, \text{LAT0}, \text{LAT1})$. A RealAmplitudes circuit with three qubits and three repetitions is used as the encoder U_{enc}, and a separate two-qubit RealAmplitudes circuit with two repetitions serves as the BrainBox latent block U_{bb} acting on the pair $(\text{LAT0}, \text{LAT1})$. The overall unitary on the data register is therefore a composition of the fixed encoder followed by a stochastic BrainBox transformation:

$$U_{\text{total}}(\boldsymbol{\theta}) = U_{\text{bb}}^{\text{drop}}(\boldsymbol{\theta}_{\text{bb}}) \circ U_{\text{enc}}(\boldsymbol{\theta}_{\text{enc}}),$$

where $U_{\text{bb}}^{\text{drop}}$ denotes the BrainBox circuit with dropout applied.

Dropout is implemented at the level of individual CNOT gates inside U_{bb}. For each training forward pass, the BrainBox circuit is first instantiated with its current parameters and then traversed gate by gate. Single-qubit rotation gates (`ry`, `rz`) are always kept, so the local degrees of freedom of the latent qubits are never removed. For each entangling gate (CNOT), however, a random decision is made: with probability

$$p_{\text{drop}} = \texttt{layer\_dropout} \times \texttt{gate\_dropout},$$

the CNOT is skipped and effectively replaced by the identity on its two qubits; with probability $1 -$

p_{drop} it is kept in the circuit. In the experiments reported here, we use $\texttt{layer\_dropout} = 0.2$ and $\texttt{gate\_dropout} = 0.4$, so that, on average, a noticeable but not overwhelming fraction of entangling gates in the BrainBox block is removed at each iteration.

This procedure means that every training step is executed with a slightly different latent circuit, as the pattern of active CNOTs changes from batch to batch. The effective model can be viewed as an ensemble average over many "thinned" BrainBox circuits, which tends to discourage the encoder from relying on any single entangling path and encourages the latent state to be robust under small structural perturbations. Importantly, dropout is applied only within the two-qubit BrainBox ansatz on $(\text{LAT0}, \text{LAT1})$, so the amplitude-encoding stage and the three-qubit encoder U_{enc} remain fixed. The dropout-augmented BrainBox block, highlighted in panel (b) of 6 figure., therefore serves as a localized, noise-aware regularizer acting specifically on the latent subsystem.

4.4 Classical and Quantum kNN Classification

After training the quantum autoencoder, each input sample is mapped to a latent representation ψ_{lat} encoded on two qubits. These latent vectors constitute a low-dimensional embedding on which downstream classification is performed. In this work we benchmark two classifiers operating on the same latent space: the classical k-nearest neighbour (kNN) classifier and its quantum counterpart (QkNN).

4.4.1 Classical kNN on Latent Vectors

The k-nearest neighbours (kNN) algorithm is a non-parametric, distance-based classifier that assigns a label to a test sample by examining the labels of the k closest training examples. Because it relies only on distances in feature space and makes no assumptions about the underlying data distribution, kNN serves as a simple yet effective baseline for evaluating the separability of the latent representations produced by the quantum autoencoder.

Given a training set

$$\mathcal{D} = \{(\mathbf{z}_i, y_i)\}_{i=1}^{N},$$

where each $\mathbf{z}_i \in \mathbb{R}^d$ denotes a latent feature vector and y_i its corresponding class label, kNN classifies a test point $\mathbf{z}_{\text{test}}$ by computing its distance to all training vectors:

$$d(\mathbf{z}_{\text{test}}, \mathbf{z}_i) = \|\mathbf{z}_{\text{test}} - \mathbf{z}_i\|.$$

The k smallest distances determine the neighbourhood of $\mathbf{z}_{\text{test}}$, and the predicted class is obtained by majority vote:

$$\hat{y} = \text{mode}\left(\{y_i : \mathbf{z}_i \text{ among the } k \text{ nearest neighbours}\}\right).$$

The choice of k controls the bias–variance tradeoff: small values of k make the classifier sensitive to noise in the training data, whereas larger values promote smoother decision boundaries. In this work, kNN operates on the low-dimensional latent vectors produced by the quantum autoencoder. High classification accuracy using these features indicates that the latent space retains

meaningful class structure, while reduced performance suggests that information relevant for discrimination may have been lost during compression.

4.4.2 Quantum kNN on Latent Quantum States

Instead of converting latent quantum states into classical numbers, the quantum kNN algorithm operates directly on their quantum representations. Each training and test point is encoded into a two-qubit state $|\psi\rangle$ using angle or amplitude encoding. The distance between two latent states is then estimated via a SWAP-test-based fidelity measurement, shown in 7 figure..

Given two states $|\psi\rangle$ and $|\phi\rangle$, their fidelity is

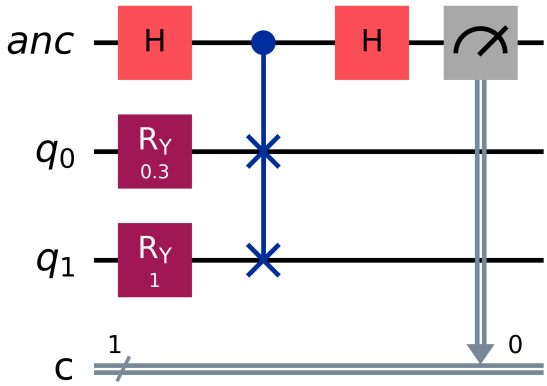
$$F(\psi, \phi) = |\langle\psi \mid \phi\rangle|^2,$$

and the QkNN classifier uses this fidelity as a similarity measure. The SWAP test computes

$$P(\text{anc} = 0) = \frac{1}{2}\Big(1 + |\langle\psi \mid \phi\rangle|^2\Big),$$

so the overlap between states can be extracted from the probability of measuring the auxiliary qubit in the $|0\rangle$ state.

For each test state, this fidelity is evaluated against all training states, and the k closest neighbours (highest fidelities) determine the predicted class. Because the latent dimension is only two qubits, the QkNN circuit remains compact and NISQ-compatible.



7 figure. *Two-dimensional QkNN similarity circuit used for comparing latent states. The test state and a stored training state are encoded on qubits q_0 and q_1, while an auxiliary qubit (labelled anc) performs the SWAP test. Measurement of the auxiliary qubit yields the fidelity between the two latent states.*

4.4.3 Hyperparameter Tuning for Classical and Quantum kNN

Both the classical kNN classifier and the quantum kNN (QkNN) model require systematic hyperparameter tuning to achieve reliable performance on anomaly detection. Because false positives

correspond to normal samples incorrectly flagged as anomalous, and therefore pose a significant practical issue, we adopt a two-stage tuning strategy that prioritizes low false-positive rates (FPR) before optimizing overall predictive performance.

For the classical kNN model, the primary hyperparameters are the number of neighbours k and the choice of distance metric, with Euclidean distance used in this work. Candidate values of k are first evaluated on a validation subset of normal samples to compute the false-positive rate,

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}}.$$

Configurations that produce high FPR are discarded, as they indicate poor generalization to genuine behaviour. Among the remaining candidates, the optimal value of k is selected by maximizing the F1 score,

$$\mathrm{F1} = \frac{2 \cdot \mathrm{Precision} \cdot \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}},$$

ensuring an appropriate balance between detecting anomalies and limiting false alarms.

The QkNN classifier is tuned in an analogous manner, with neighbourhood size k serving as its primary hyperparameter. Unlike the classical model, QkNN evaluates similarity using quantum fidelity obtained from SWAP-test circuits applied directly to latent quantum states. Each candidate value of k is first screened according to its false-positive performance on normal validation data, a necessary step because quantum fidelity can behave differently from classical distance metrics and may lead to overly confident misclassification of normal samples. Among the candidates with acceptable FPR, the value of k that maximizes the F1 score is selected, ensuring that the final QkNN model maintains low false-positive rates while achieving strong anomaly-detection capability.

4.5 Experimental Settings

4.5.1 KDD'99 Dataset

The KDD'99 dataset is one of the most widely used benchmarks for evaluating intrusion detection and anomaly detection algorithms. It was derived from the DARPA 1998 Intrusion Detection Evaluation Program, in which approximately 4 GB of tcpdump network traffic were collected over seven weeks and processed into roughly five million connection records (Lee and Stolfo, 1999). Each connection corresponds to a communication session between a source and destination and is represented by a feature vector describing various statistical and content-based properties of the traffic.

The KDD'99 training set contains about 4.9 million connection vectors, each with 41 engineered features and a label identifying the connection as either *normal* or as one of several simulated attack types. These attacks fall into four major categories:

- **Denial of Service (DoS)** — attacks that overwhelm computational or network resources (e.g., smurf, neptune).
- **Probe** — attacks that scan networks to gather information (e.g., satan, ipsweep).
- **Remote-to-Local (R2L)** — unauthorized access from a remote machine.

- **User-to-Root (U2R)** — privilege escalation attacks.

A distinctive challenge in KDD'99 is that the test set includes attack types not present in the training data (24 attack types in training versus 14 additional unseen types in testing), making the problem closer to real-world anomaly detection where new variations of known attacks may appear.

The 41 KDD features are organized into three feature categories:

1. **Basic features:** Attributes extracted directly from raw TCP/IP connections, such as duration, protocol type, or flag status. These capture general connection behavior but may introduce detection delays since they reflect aggregate properties.

2. **Traffic features:** These describe statistical patterns over a time window or over a window of past connections.

   - *Same-host features:* statistics based on connections to the same destination host within the last two seconds.
   - *Same-service features:* statistics based on connections using the same network service within the last two seconds.

   Because some slow probing attacks operate on longer timescales (e.g., one scan per minute), extended "connection-based" versions of these features were also introduced, computed over the last 100 connections rather than a fixed two-second interval.

3. **Content features:** These examine the actual payload content to detect attack types such as R2L and U2R, which typically require inspecting login patterns, command sequences, or other semantic indicators of suspicious behavior. Examples include the number of failed login attempts or the presence of shell commands in the payload.

Together, these characteristics make KDD'99 a challenging and diverse benchmark that tests both generalization to normal behaviour and robustness to previously unseen attack types.

4.5.2 Data Preprocessing

The data preprocessing pipeline consists of the following steps:

1. **One-hot encoding.** Categorical features are transformed using one-hot encoding, where each category is represented by a binary vector. The length of the vector corresponds to the total number of categories, with a value of one assigned to the index of the observed category and zeros elsewhere.

2. **Normalization.** Numerical features are scaled using min–max normalization. Each feature is rescaled to the interval $[0,1]$ by subtracting the minimum value and dividing by the feature range, ensuring uniform contribution of all features during model training.

3. **Dimensionality reduction.** Principal component analysis (PCA) is employed to reduce the dimensionality of the dataset while preserving as much variance as possible.

4.5.3 Data Split

Dataset Split	Composition	Number of Samples
Training	Normal only	≈ 0.97M
Validation	Mixed (normal + attacks)	100,000
Predefined Test	Mixed (normal + attacks)	311,029

3 table. *Dataset split used for training, validation, and evaluation. The training set contains only normal samples (20% of the available normal data), while validation and test sets include both normal and attack samples. with ratio 20% normal and 80% attack*

4.5.4 Reproducibility and Experimental Environment

To ensure reproducibility of the reported results, all experiments were conducted under a clearly specified software and hardware environment, with consistent data preprocessing and evaluation protocols.

Software Environment.

All experiments were implemented in Python 3.10. Quantum circuit simulation and training were performed using Qiskit 0.46. Classical machine learning components, including PCA and k-nearest neighbour classifiers, were implemented using `scikit-learn` 1.3. Numerical computations relied on `NumPy` 1.24, and optimization routines were implemented using standard Python scientific libraries. No deep-learning frameworks such as TensorFlow or PyTorch were used for model training.

Hardware and Operating System.

All experiments were executed on a MacBook Pro equipped with an Apple M4 Pro processor and 24 GB of unified memory, running macOS. Quantum circuits were simulated entirely on classical hardware using statevector and shot-based simulators; no GPU acceleration or quantum hardware backends were employed.

Random Seeds and Repeated Runs.

To reduce variability due to stochastic effects, fixed random seeds were used for circuit parameter initialization, data shuffling, and optimizer behaviour. Each experiment was repeated multiple times with different random seeds, and reported results correspond to average performance across runs unless stated otherwise.

Data Splits and Leakage Prevention.

The KDD'99 dataset was divided into disjoint training, validation, and test sets following a strict separation protocol. All preprocessing steps, including feature scaling and principal component anal-

ysis (PCA), were fitted exclusively on the training data and then applied unchanged to validation and test sets. This prevents information leakage and ensures that performance metrics reflect genuine generalization rather than implicit access to test data.

5 Experimental Investigation

The experimental investigation in this chapter aims to evaluate how quantum autoencoder, latent dimensionality, and regularization strategies influence anomaly detection performance. The hypothesis is that how circuit depth and complexity of circuit limits latent-space separability and generalization, while structured latent processing and regularization of circuit can improve robustness and reduce false-positive rates. To test this hypothesis, we analyze, latent-space structure, and downstream classification performance using both classical and quantum k-nearest neighbour methods. Model performance is primarily evaluated using PR–AUC and F1-score, which are more informative than ROC–AUC for highly imbalanced datasets such as KDD'99, as they emphasize performance on the minority (attack) class. False-positive rate is reported explicitly to assess generalization on normal data and practical usability of the models.

We begin with a simple quantum autoencoder architecture, shown in 5 figure., which serves as our baseline model. This circuit provides the minimal structure needed to test how well quantum compression preserves information before introducing our proposed BrainBox layer. With this baseline established, we first evaluate the QAE on the smallest compression setting, where two input qubits are reduced to a single latent qubit. We then extend this baseline with the BrainBox layer shown in 6 figure.b, which adds a more expressive variational structure to the encoder and allows us to assess the impact of increased circuit capacity on compression performance.

All quantum circuit simulations and training experiments were conducted using `Qiskit` (version 0.46) with Python 3.10 on a local simulation backend. To study the behaviour of the quantum autoencoder (QAE) under different register sizes and circuit expressivities, we evaluate three compression configurations:

- **2-qubit input $\rightarrow$ 1-qubit latent space**
- **3-qubit input $\rightarrow$ 2-qubit latent space**
- **4-qubit input $\rightarrow$ 3-qubit latent space**

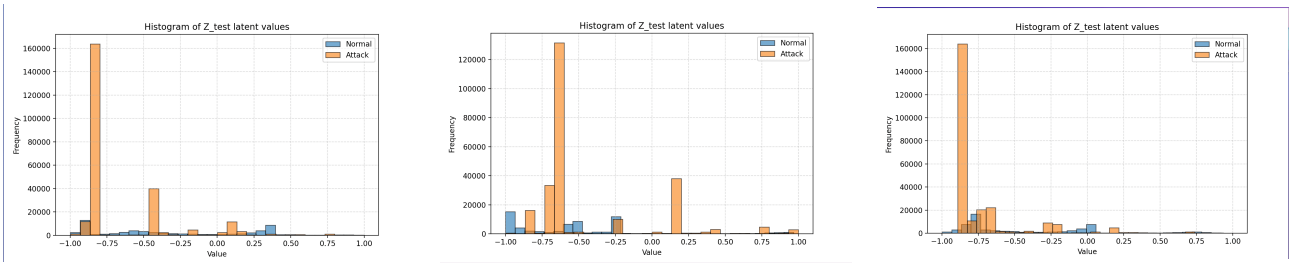
For each configuration, the variational encoder is instantiated with **3**, **6**, and **9** layers of the RealAmplitudes ansatz to evaluate how circuit depth influences compression behaviour and reconstruction fidelity. All models are trained using the Adam optimizer with a learning rate of 0.001, a batch size of **64**, and **200** epochs.

Table 4 table. summarizes the complete experimental setup.

4 table. *QAE compression configurations, tested layer depths, and training hyperparameters.*

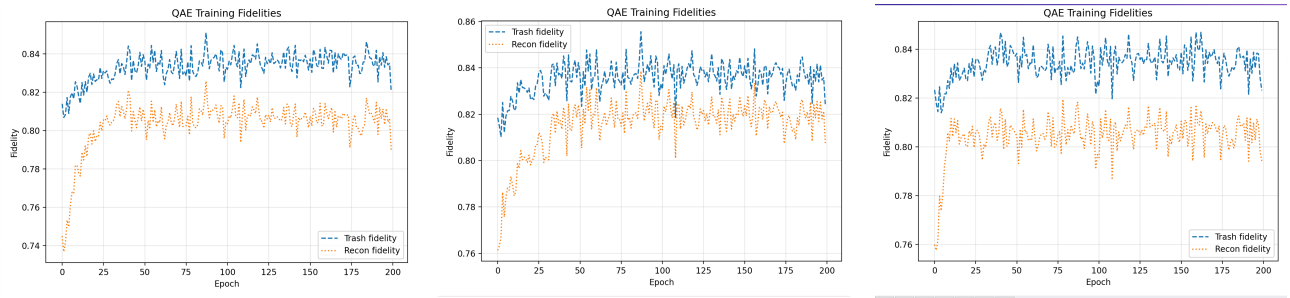
Experiment	Input Qubits	Latent Qubits	Layers Tested	Batch Size
QAE-2→1	2	1	3, 6, 9	64
QAE-3→1	3	1	3, 6, 9	64
QAE-3→2	3	2	3, 6, 9	64
QAE-4→3	4	3	3, 6, 9	64
Optimizer	Adam (learning rate = 0.001)			
Epochs	200			
Framework	Qiskit 0.46, Python 3.10			

5.1 2-qubit input → 1-qubit



3 layers 6 layers 9 layers

8 figure. *Latent $\mathcal{Z}_{test}$ distributions for the 2-qubit → 1-qubit compression with 3, 6, and 9 layers.*



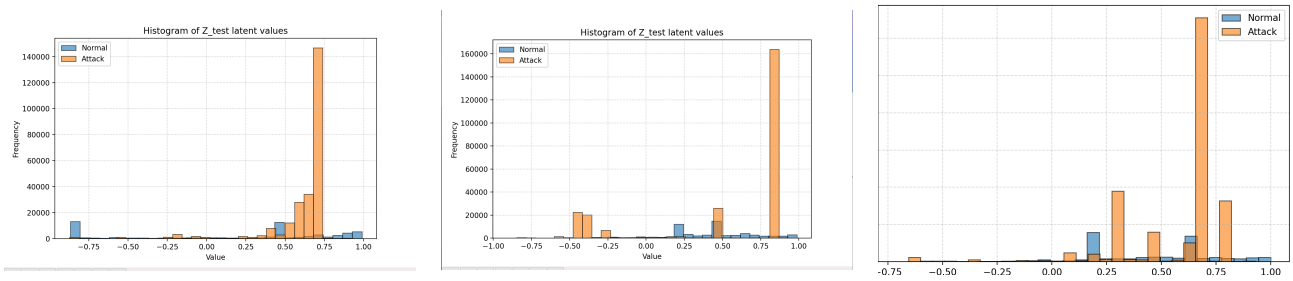
3 layers 6 layers 9 layers

9 figure. *Fidelity and Reconstruction loss for the 2-qubit → 1-qubit compression with 3, 6, and 9 layers.*

The three panels in 8 figure. and 9 figure. show the evolution of the latent $\mathcal{Z}_{\text{test}}$ values during training for the 2-qubit → 1-qubit compression using circuits of increasing depth: 3 layers (8 parameters), 6 layers (14 parameters), and 9 layers (20 parameters). Across all depths, normal and attack samples remain heavily mixed within the compressed space. Although the attack class tends to form a pronounced peak near -1, normal samples frequently populate the same region and show a comparable spread across latent values. The accompanying fidelity curves reflect the same behavior: deeper circuits exhibit slightly smoother and marginally higher trash and reconstruction fidelities; yet, the overall gains are small, and all models converge to similar fidelity bands. Together, these re-

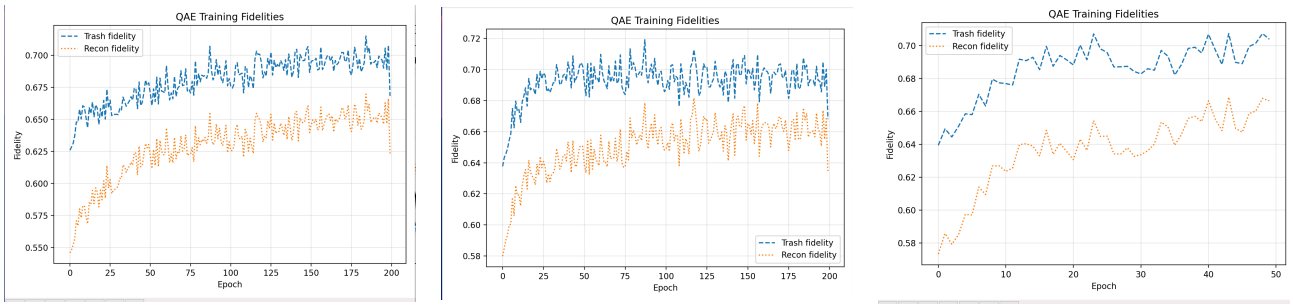
sults indicate that additional circuit depth alone is insufficient to overcome the limitations of a single-qubit bottleneck, leaving the two classes largely inseparable in the learned latent representation. A detailed view of the training cost per epoch for all three circuit depths is provided in Appendix 19 figure., where the loss curves show the same pattern of early rapid improvement followed by shallow convergence.

5.2 3-qubit input $\rightarrow$ 1-qubit



3 layers 6 layers 9 layers

10 figure. *Latent Z_{test} distributions for the 3-qubit $\rightarrow$ 1-qubit compression with 3, 6, and 9 layers.*



3 layers 6 layers 9 layers

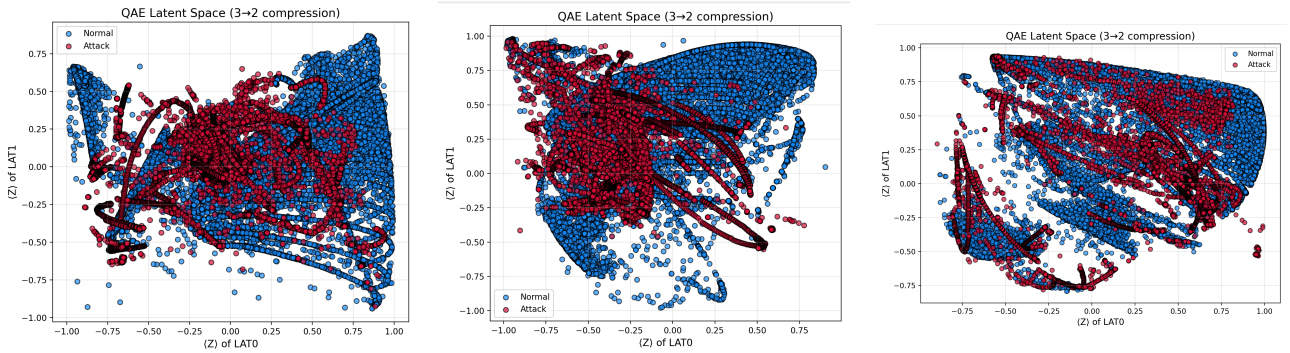
11 figure. *Fidelity and Reconstruction loss for the 3-qubit $\rightarrow$ 1-qubit compression with 3, 6, and 9 layers.*

For the 3-qubit $\rightarrow$ 1-qubit compression, the latent Z_{test} values shown for the 3-, 6-, and 9-layer circuits — corresponding to 12, 21, and 30 trainable parameters, respectively — reveal that the model continues to struggle with class separation under such an extreme bottleneck. Across all depths, attack samples tend to cluster near high positive latent values, but normal samples frequently overlap with the same region and exhibit a wide spread across the remaining latent range. Although the concentration of attack samples becomes slightly sharper with additional layers, the overall latent space remains highly mixed, indicating that even with three input qubits, compressing the entire state into a single qubit removes too much structure to meaningfully separate the two classes.

The fidelity curves for these circuits reflect the same limitations. A detailed view of the cost loss per epoch for each circuit depth is included in Appendix 20 figure.. Deeper models with more parameters achieve modest improvements in both trash and reconstruction fidelity, yet the overall gains remain small and all depths converge to similar plateau values. This suggests that increasing circuit

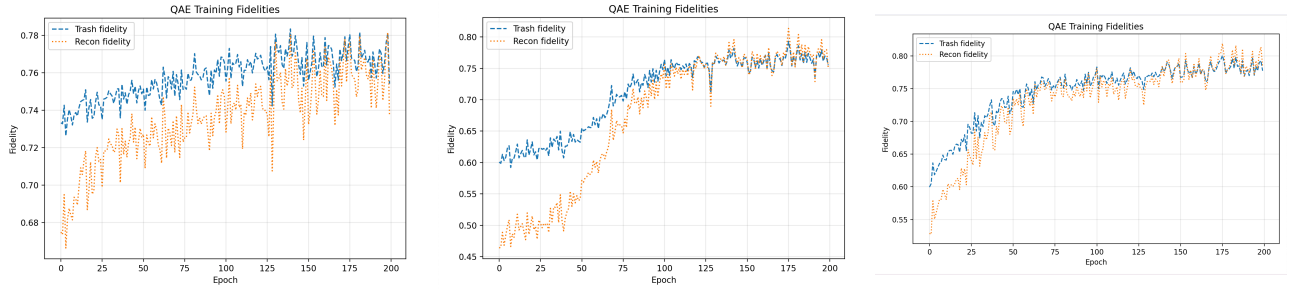
depth alone is insufficient to compensate for the aggressive 3-to-1 compression, and the resulting latent representation does not reliably encode the normal–attack distinction.

5.3 3-qubit input $\rightarrow$ 2-qubit



3 layers 6 layers 9 layers

12 figure. *Latent $\mathcal{Z}_{test}$ scatter plot for the 3-qubit $\rightarrow$ 2-qubit compression with 3, 6, and 9 layers.*



3 layers 6 layers 9 layers

13 figure. *Fidelity and Reconstruction loss for the 3-qubit $\rightarrow$ 2-qubit compression with 3, 6, and 9 layers.*

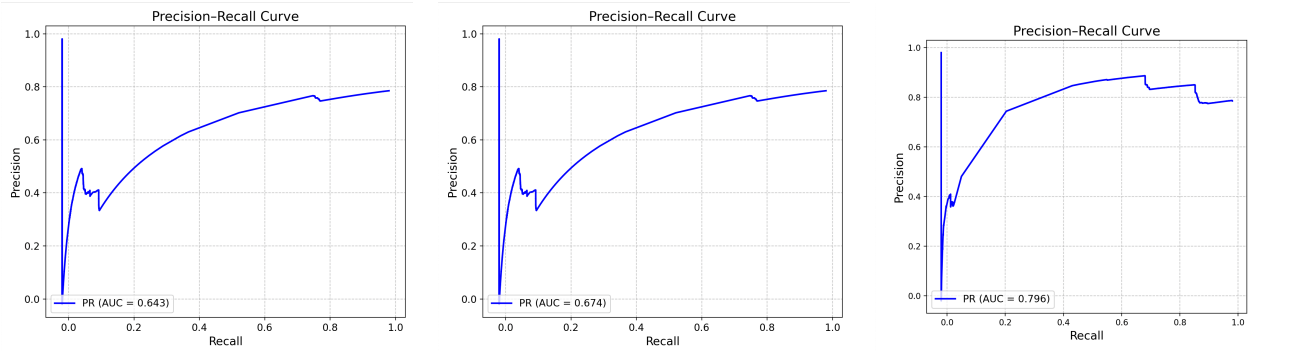
For the 3-qubit $\rightarrow$ 2-qubit compression, the latent space produced by the 3-, 6-, and 9-layer circuits — corresponding to 12, 21, and 30 trainable parameters, respectively — shows that the model captures more geometric structure than in the single-qubit bottleneck, but normal and attack samples remain substantially mixed. Across all depths, both classes occupy broad overlapping regions, with attack samples forming dense curved trajectories throughout the latent space and normal samples spread across similar areas. Although deeper circuits shape these trajectories more smoothly, the overall separation between classes remains limited, indicating that a two-qubit latent space is still insufficient to clearly disentangle normal and attack behaviour in this dataset.

The fidelity curves for the three circuit depths reflect the same trend. All models exhibit consistent improvements during training, with deeper circuits reaching slightly higher reconstruction and trash fidelities, but the differences remain modest. The underlying compression task remains challenging, and the latent representations do not develop distinct class clusters. Detailed cost-loss curves for these models are provided in Appendix 21 figure..

5.4 Fidelity Based Classification of Latents

Compression	Precision	Recall	F1-score	FPR
2 qubits $\rightarrow$ 1 qubit	0.49	0.06	0.11	28%
3 qubits $\rightarrow$ 1 qubit	0.03	0.1	0.1	52%
3 qubits $\rightarrow$ 2 qubits	0.91	0.70	0.79	30%

5 table. *Fidelity-based anomaly detection performance for each compression setting after threshold tuning for reduced false-positive rate (FPR) and improved F1-score.*



2 qubits $\rightarrow$ 1 qubit 3 qubits $\rightarrow$ 1 qubit 3 qubits $\rightarrow$ 2 qubits

14 figure. *AUC of Fidelity Based Classification*

The fidelity-based classification results in Table 5 table. show that the 2-qubit $\rightarrow$ 1-qubit and 3-qubit $\rightarrow$ 1-qubit configurations perform poorly as anomaly detectors, with low F1-scores (0.11 and 0.10) and relatively high FPRs of 28% and 52%, respectively. This is consistent with their modest precision–recall AUCs of approximately 0.64 and 0.67, indicating that threshold tuning cannot compensate for the strong class overlap in their latent spaces. In contrast, the 3-qubit $\rightarrow$ 2-qubit compression achieves a much better trade-off, with high precision (0.91), substantially higher F1-score (0.79), and a lower FPR of 30%, supported by the best PR–AUC of about 0.80. This suggests that the 2-qubit latent space is the only setting where fidelity-based scores become reliably useful for anomaly detection.

5.5 Classical and Quantum k-Nearest Neighbour Methods

Method / Compression	Precision	Recall	F1-score	FPR	PR-AUC
Classical kNN					
2 qubits $\rightarrow$ 1 qubit	0.88	0.36	0.51	21%	0.78
3 qubits $\rightarrow$ 1 qubit	0.93	0.78	0.85	25%	0.82
3 qubits $\rightarrow$ 2 qubits	0.94	0.89	0.9	20%	0.89
Quantum kNN					
2 qubits $\rightarrow$ 1 qubit	0.94	0.72	0.82	31%	0.81
3 qubits $\rightarrow$ 1 qubit	0.94	0.75	0.83	22%	0.82
3 qubits $\rightarrow$ 2 qubits	0.95	0.90	0.91	25%	0.89

6 table. *Performance comparison of classical kNN and quantum kNN across the three QAE compression settings. Values correspond to thresholds tuned for low false-positive rate (FPR) and maximal F1-score.*

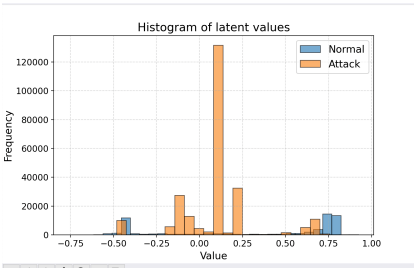
Table 6 table. shows that downstream kNN performance is strongly influenced by the quality of the QAE latent representation. Under the most aggressive compression ($2\rightarrow1$), both classical and quantum kNN exhibit reduced F1-scores and elevated FPR, reflecting that the QAE discards substantial discriminative information in this setting. Quantum kNN achieves higher recall than classical kNN, but at the cost of an increased FPR, a trend consistent with the broader behavior of QAE-based anomaly detection, which typically yields higher false-positive rates when the latent space is severely compressed.

Performance improves markedly for the $3\rightarrow1$ setting, where both methods benefit from the additional input qubit. Classical and quantum kNN reach comparable F1-scores (0.85 and 0.83), with quantum kNN maintaining a lower FPR. The best results are obtained with the $3\rightarrow2$ compression, where the richer latent space yields the most separable structure. In this case, quantum kNN slightly outperforms classical kNN in both recall (0.90) and F1-score (0.91), confirming that quantum distance evaluation becomes most effective when the encoded features preserve sufficient information. The corresponding PR–AUC values, included in the appendix, follow the same trend and further validate these observations.

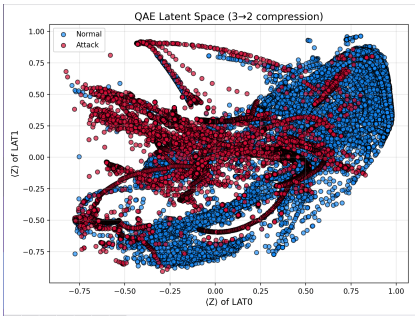
5.6 BrainBox Layer

After evaluating the simple quantum autoencoder, we now examine the effect of introducing the BrainBox layer, shown in 6 figure.b. This layer adds a more expressive variational structure to the encoder, enabling richer feature mixing while keeping the overall circuit compact. In the following experiments, we assess how this enhanced architecture influences reconstruction fidelity, latent-space geometry, and downstream anomaly-detection performance under the same compression settings used for the baseline QAE.

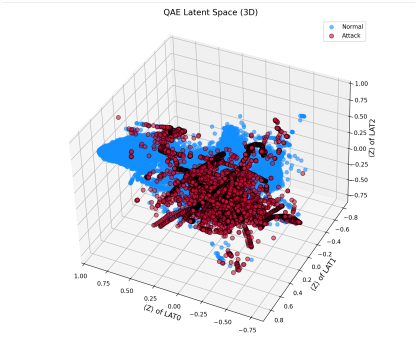
5.6.1 Compression



2-1-2

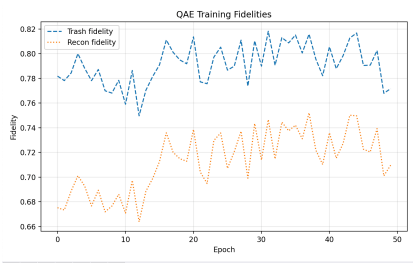


3-2-3

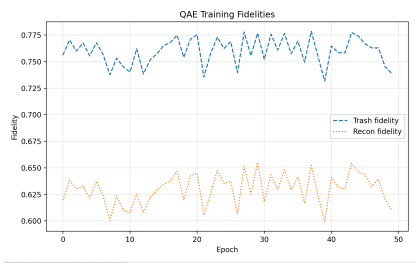


4-3-4

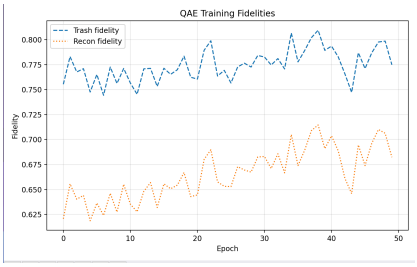
15 figure. *Visualization of Latent* Z_{test}



2-1-2



3-2-3



4-3-4

16 figure. *Visualization of Training of BrainBox layer Fidelity QAE per epoch*

Compression Setting	Parameters	Trash Fidelity	Recon. Fidelity
$2 \to 1 \to 2$	22	0.77	0.71
$3 \to 2 \to 3$	32	0.74	0.62
$4 \to 3 \to 4$	42	0.76	0.68

7 table. *Final training fidelities and number of trainable parameters for the three BrainBox-enhanced QAE configurations.*

Figure 15 figure. and the corresponding latent-space plots show how the BrainBox layer enhances the expressive power of the encoder across the three compression settings ($2 \to 1 \to 2$, $3 \to 2 \to 3$, and $4 \to 3 \to 4$). Compared to the simple quantum autoencoder, the BrainBox architecture consistently produces more structured and informative latent representations.

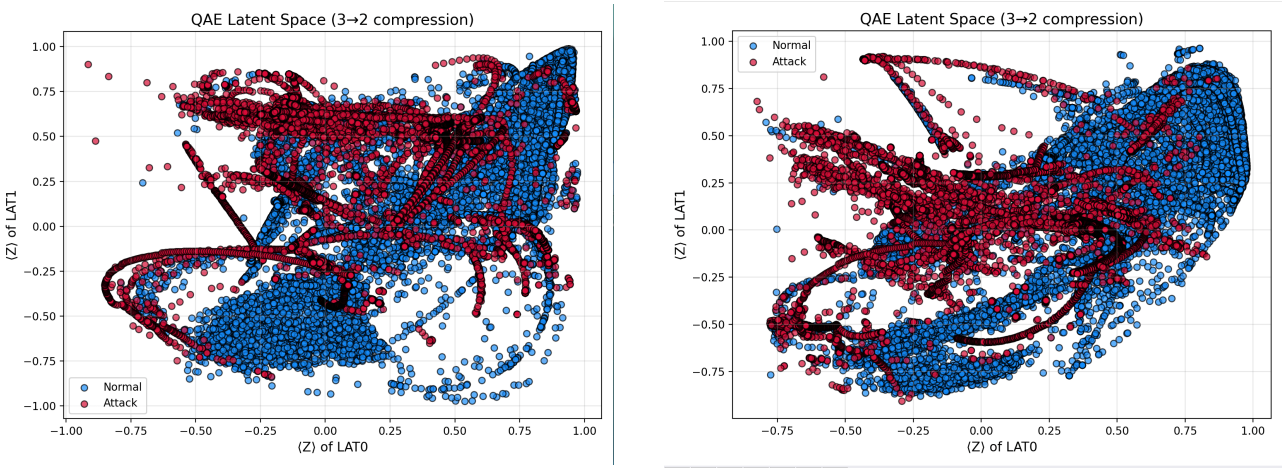
In the $2 \to 1 \to 2$ experiment, the single latent qubit still imposes a severe bottleneck, and normal and attack samples remain mixed; however, the latent distribution already shows clearer clustering than in the baseline QAE, indicating that the BrainBox layer extracts more meaningful features even under tight compression.

The $3 \to 2 \to 3$ setting yields a noticeably improved latent geometry. While some overlap persists, the two-qubit latent space reveals more coherent manifolds, with attack trajectories forming identifiable patterns not captured by the simple autoencoder.

The 4→3→4 results show the strongest improvement: the three-qubit latent space becomes far more organized, with normal samples forming a smoother, compact region and attack samples spreading along distinct nonlinear structures. This demonstrates that, once the latent dimension is sufficient, the BrainBox layer leverages its additional variational depth to encode richer and more discriminative features than the simple quantum autoencoder.

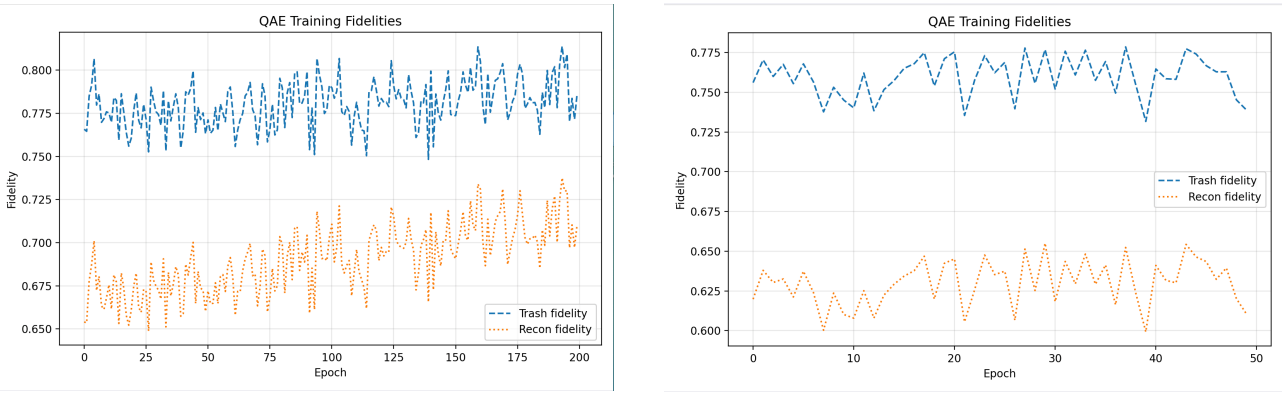
A detailed view of the cost loss per epoch for each circuit depth is included in Appendix 24 figure..

5.6.2 Dropout and L2 Regularization with BrainBox Layer



Dropout Layer (0.2) and Gate (0.4) L2 Regularization

17 figure. *Visualization of Latent Z_{test}*



Dropout Layer (0.2) and Gate (0.4) L2 Regularization

18 figure. *Visualization of Training of Dropout and L2 Regularization Fidelity per epoch*

18 figure. shows the latent representations obtained with a regularized BrainBox layer, combining dropout ($p = 0.2$), gate scaling ($\alpha = 0.4$), and ℓ_2 regularization with $\lambda = 10^{-3}$. Compared to the simple quantum autoencoder, the latent space becomes smoother and more stable, with normal samples forming more compact structures while attack samples remain more dispersed. Although overlap is still present, the regularized BrainBox layer reduces extreme trajectories and overfitting,

leading to a qualitatively improved and more interpretable latent representation for anomaly detection. A detailed view of the cost loss per epoch for each circuit depth and tuning L2 Regularization are included in Appendix 25 figure..

5.6.3 Classical and Quantum k-Nearest Neighbour Methods

Method / Compression	Precision	Recall	F1-score	FPR	PR-AUC
Classical kNN					
BrainBox Layer	0.98	0.92	0.95	9%	0.98
Dropout	0.99	0.90	0.94	5%	0.98
L2 Regularization	0.97	0.91	0.93	8%	0.98
Quantum kNN					
BrainBox Layer	0.97	0.95	0.95	12%	0.98
Dropout	0.96	0.93	0.94	11%	0.97
L2 Regularization	0.97	0.93	0.95	11%	0.98

8 table. *Performance comparison of classical kNN and quantum kNN across the three QAE compression settings. Values correspond to thresholds tuned for low false-positive rate (FPR) and maximal F1-score.*

Table 8 table. reports the anomaly detection performance obtained with the BrainBox layer augmented by ℓ_2 regularization ($\lambda = 10^{-4}$), evaluated using both classical and quantum kNN. Across all configurations, the regularized BrainBox architecture achieves consistently high PR-AUC values, indicating strong separability in the learned latent space.

For classical kNN, the model reaches a high F1-score while maintaining a low false-positive rate, demonstrating that the regularized BrainBox effectively balances precision and recall. A similar trend is observed for quantum kNN, where performance remains competitive despite slightly higher FPR, reflecting the increased sensitivity of the quantum distance measure.

Overall, these results confirm that incorporating ℓ_2 regularization into the BrainBox layer improves generalization over the simple quantum autoencoder and yields robust latent representations suitable for downstream anomaly detection. A detailed view of the PR-AUC for each circuit depth is included in Appendix 26 figure. and 27 figure. .

5.7 Discussion

5.7.1 Comparison with Prior Work on KDD99.

9 table. *Comparison of anomaly detection performance on the KDD99 dataset. Reported metrics follow the original papers. FPR is reported only when explicitly available.*

Study	Method	Precision	Recall	F1-score	FPR
Tavallaee et al. [47]	Random Forest	98.0	99.8	98.9	–
Shiravi et al. [45]	SVM classifier	94.5	95.8	95.1	–
Mukkamala et al. [29]	ANN (MLP)	95.3	96.4	95.8	–
Hdaib et al. [15]	QAE + OC-SVM	95.06	99.43	97.19	–
Hdaib et al. [15]	QAE + QRF	89.63	97.16	93.41	–
This Thesis work	**BrainBox QAE + classical kNN**	**0.98**	**0.92**	**0.95**	**9%**

The comparison in 9 table. illustrates the effectiveness of the BrainBox-enhanced quantum autoencoder alongside notable classical and QAE-based anomaly detection techniques as applied to the KDD99 dataset. Traditional machine learning methods, including random forests, SVMs, and multilayer perceptrons, achieve impressive F1-scores, demonstrating their efficacy on this dataset; however, these studies fail to provide specific information on false-positive rates, complicating the evaluation of their performance with normal traffic.

Quantum-based techniques, particularly the QAE models presented by Hdaib et al., demonstrate high F1-scores by integrating quantum autoencoders with both classical and quantum classifiers. Nonetheless, their assessment primarily emphasizes classification accuracy and recall, neglecting to examine latent-space generalization or false-positive behavior. In contrast, the method proposed in this work specifically provides false-positive rates and attains an F1-score of 0.95 while lowering the false-positive rate to 5

In summary, the table underscores a significant difference between previous research and this thesis: while many current methodologies focus on optimizing high-level performance indicators, the proposed BrainBox framework emphasizes achieving a balanced detection performance along with managed false-positive rates. This focus renders the approach particularly well-suited for real-world anomaly detection scenarios, where a high rate of false alarms can greatly hinder system usability.

5.7.2 Main Discussion

The experimental outcomes illustrate the constraints of basic quantum autoencoders and the circumstances where significant latent representations can be acquired for anomaly detection. In all baseline tests, heavy compression into a single latent qubit ($2\rightarrow1$ and $3\rightarrow1$) consistently results in considerable information loss. Enhancing the circuit depth from 3 to 9 layers provides only slight gains in reconstruction and trash fidelity, with the resulting latent spaces still exhibiting substantial mixing

and overlap between normal and attack samples. These results suggest that merely increasing circuit depth cannot offset an overly narrow bottleneck, and that optimization focused on fidelity may yield convergence even when the class-discriminative structure is not maintained.

The shift from a single-qubit latent space to a two-qubit structure (3→2) signifies a qualitative transformation. Although the latent distributions continue to show overlap, their geometry becomes significantly richer, allowing downstream classifiers to utilize latent structure more effectively. This is evident in a marked improvement in fidelity-based classification metrics and a notable rise in F1-score, reaffirming that a minimum latent dimensionality is essential for fidelity to function as a reliable anomaly score. These findings emphasize that reconstruction fidelity on its own is an inadequate representation for anomaly separability when the latent space is overly restricted.

The subsequent kNN experiments further support this observation. Both classical and quantum kNN performance closely correlates with the quality of the acquired latent representation. In instances of strong compression, both techniques experience heightened false-positive rates and lower F1-scores, whereas the 3→2 setup demonstrates stable and robust performance. Quantum kNN achieves somewhat higher recall in various scenarios, but this often leads to increased false positives, suggesting that quantum fidelity-based distances are more sensitive to latent noise and overlap. However, when the latent space retains sufficient structure, quantum kNN either matches or slightly exceeds the performance of its classical equivalent.

The introduction of the BrainBox layer significantly changes these dynamics. In all configurations evaluated, the BrainBox-enhanced autoencoder produces more structured and consistent latent representations than the baseline QAE, even with similar parameter budgets. This effect becomes increasingly pronounced as the latent dimension increases, with the 4→3→4 configuration showing the clearest separation between normal and attack trajectories. These findings indicate that actively modeling latent-space processing, rather than depending solely on a single encoder block, is essential for learning distinct quantum representations.

Lastly, implementing regularization techniques—such as layer and gate dropout along with ℓ_2 weight decay—further enhances generalization. Regularized BrainBox models demonstrate smoother latent manifolds, fewer extreme trajectories, and considerably lower false-positive rates. Both classical and quantum kNN attain high F1-scores and PR–AUC values in this scenario, confirming that overfitting is a significant challenge in QAE-based anomaly detection and that targeted regularization serves as an effective solution.

Conclusion and Future Work

This thesis explored various quantum autoencoders for the purpose of anomaly detection, focusing specifically on the structure of the latent space, generalization capabilities, and the rates of false positives in normal data. Through experimental analysis, the drawbacks of basic QAE architectures in terms of compression were highlighted, and the conditions necessary for effective latent representations to develop were clarified. The findings indicate that factors such as latent dimensionality, architectural configuration, and regularization are more significant than circuit depth by itself. In response to these observations, a BrainBox layer QAE circuit was proposed and assessed, resulting in enhanced performance in anomaly detection. Overall, this research offers empirical insights into the design and evaluation of QAEs for effective anomaly detection applications.

Conclusions with Respect to the Thesis Objectives

- **Objective 1: Architectural analysis.** The results indicate that the number of qubits and the size of the latent bottleneck significantly affect reconstruction accuracy, latent separability, and the effectiveness of anomaly detection, whereas simply increasing the circuit depth produces only slight improvements.
- **Objective 2: Generalization on normal data.** The results demonstrate that high reconstruction fidelity does not necessarily imply good generalization, as several QAE configurations show high false-positive rates despite stable training convergence.
- **Objective 3: Latent-space evaluation with kNN.** The findings indicate that achieving high reconstruction accuracy doesn't automatically lead to effective generalization, as numerous QAE setups exhibit elevated false-positive rates even with consistent training convergence.
- **Objective 4: Latent-space processing.** The suggested BrainBox layer improves the organization and interpretability of latent space, resulting in embeddings that are more fluid and better at distinguishing differences compared to the baseline QAE, especially when the latent dimension is adequately expressive.
- **Objective 5: Regularization techniques with BrainBox.** The implementation of ℓ_2 regularization, gate dropout, and layer dropout mitigates overfitting, stabilizes latent representations, and decreases false-positive rates, highlighting the significance of intentional regularization in QAE-based anomaly detection.

Answers to the Research Questions

- **RQ1:** Architectural elements play a crucial role in the robustness and generalization of QAE. Specifically, the size of the latent bottleneck has a major influence, while simply increasing the depth of the circuit cannot offset too much compression.

- **RQ2:** Quantum-compressed latent representations work well for detecting anomalies only when enough latent capacity is maintained. Both classical and quantum kNN struggle with extreme compression but excel in higher-dimensional latent spaces, with quantum kNN providing a slightly better recall, albeit with a rise in false-positive rates.
- **RQ3:** Latent-space processing methods and regularization strategies, such as the BrainBox layer, dropout, and ℓ_2 regularization, significantly enhance training stability, the organization of latent space, and generalization, resulting in more dependable anomaly detection.

Future Work

A key focus for future research is a more thorough examination of the optimization dynamics present in BrainBox layer quantum autoencoders. Although the BrainBox layer enhances the structure of the latent space and boosts performance in anomaly detection, certain configurations demonstrate slower or unstable convergence during the training process. Upcoming studies could investigate how the increased variational expressiveness of the BrainBox layer interacts with the optimization landscape, as well as the selection of cost functions and optimizers. Specifically, analyzing gradient behavior, potential barren plateaus, and strategies for parameter initialization may shed light on the convergence challenges observed. By addressing these issues, it may be possible to develop more stable training protocols and further enhance the practical use of BrainBox-based QAE architectures.

References and Sources

[1] A. C. Bahnsen, A. Stojanovic, D. Aouada, B. Ottersten. “Example-dependent Cost-sensitive Logistic Regression for Credit Card Fraud Detection.” In: *2014 13th International Conference on Machine Learning and ApplicationsbasBas* 48.3 (2014), pages 263–269. https://doi.org/10.1109/ICMLA.2014.48.

[2] A. Basheer, A. Afham, S. K. Goyal. “Quantum k nearest neighbors algorithm.” In: *arXiv preprint arXiv:2003.07450* (2020).

[3] K. Beer, D. Bondarenko, T. Farrelly, T. J. Osborne, R. Salzmann, R. Wolf. “Training deep quantum circuits.” In: *Nature Communications* 11.1 (2020), page 808.

[4] H. Ben Amor, S. Benferhat, Z. Elouedi. “Naive Bayes vs decision trees for intrusion detection: A comparison.” In: *ACM Symposium on Applied Computing (SAC)* 1 (2004), pages 420–424. https://doi.org/10.1145/967900.967989.

[5] M. Benedetti, E. Lloyd, S. Sack, M. Fiorentini. “Parameterized quantum circuits as machine learning models.” In: *Quantum Science and Technology* 4.4 (2019), page 043001. https://doi.org/10.1088/2058-9565/ab4eb5.

[6] S. Bordoni, D. Stanev, T. Santantonio, S. Giagu. “Long-Lived Particles Anomaly Detection with Parametrized Quantum Circuits.” In: *Particles* 6.1 (2023), pages 297–311. https://doi.org/10.3390/particles6010016.

[7] F. Carcillo, Y.-A. Le Borgne, O. Caelen, Y. Kessaci. “Combining unsupervised and supervised learning in credit card fraud detection.” In: *Information Sciences* (2021). ISSN: 0020-0255. https://doi.org/10.1016/j.ins.2019.05.042.

[8] M. Cerezo, A. Poremba, Ł. Cincio, P. J. Coles. “Variational Quantum Fidelity Estimation.” In: *Quantum* 4 (2020), page 248. https://doi.org/10.22331/q-2020-03-26-248.

[9] A. Dal Pozzolo, O. Caelen, Y.-A. Le Borgne, S. Waterschoot, G. Bontempi. “Calibrating Probability with Undersampling for Unbalanced Classification.” In: *2015 IEEE Symposium Series on Computational Intelligence* (2015), pages 159–166. https://doi.org/10.1109/SSCI.2015.33.

[10] Y. G. Duman Sahin, Ekrem. “Detecting Credit Card Fraud by Decision Trees and Support Vector Machines.” In: *Proceedings of the International MultiConference of Engineers and Computer Scientists 2011 (IMECS 2011)*. Hong Kong: IAENG, 2011, pages 442–447.

[11] R. Frehner, K. Stockinger. “Applying Quantum Autoencoders for Time Series Anomaly Detection.” In: *Quantum Machine Intelligence* 7 (2025), page 59. https://doi.org/10.1007/s42484-025-00285-1.

[12] I. Goodfellow, Y. Bengio, A. Courville. *Deep Learning*. MIT Press, 2016.

[13] M. Grossi, N. Ibrahim, V. Radescu, R. Loredo, K. Voigt, C. von Altrock, A. Rudnik. “Mixed Quantum–Classical Method for Fraud Detection with Quantum Feature Selection.” In: *IEEE Transactions on Quantum Engineering* 3 (2022), pages 1–12. https://doi.org/10.1109/TQE.2022.3213474.

[14] M. Hdaib, S. Rajasegarar, L. Pan. “Quantum Autoencoder Frameworks for Network Anomaly Detection.” In: *International Conference on Neural Information Processing*. Volume 14451. Lecture Notes in Computer Science. Springer, 2023, pages 69–82. https://doi.org/10.1007/978-981-99-8073-4_6.

[15] M. Hdaib, S. Rajasegarar, L. Pan. “Quantum Deep Learning-Based Anomaly Detection for Enhanced Network Security.” In: *Quantum Machine Intelligence* 6 (2024), page 26. https://doi.org/10.1007/s42484-024-00163-2.

[16] D. Herr, B. Obert, M. Rosenkranz. “Anomaly detection with variational quantum generative adversarial networks.” In: *Quantum Science and Technology* 6.4 (2021), page 045004. https://doi.org/10.1088/2058-9565/ac0d4d.

[17] S. Hettich, D. Bay. *KDD Cup 1999 Data*. UCI KDD Archive. Available at http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html. 1999.

[18] G. E. Hinton, R. S. Zemel. “Autoencoders, minimum description length and Helmholtz free energy.” In: *Advances in Neural Information Processing Systems* 6 (1994), pages 3–10.

[19] C. Huot, S. Heng, T.-K. Kim, Y. Han. “Quantum Autoencoder for Enhanced Fraud Detection in Imbalanced Credit Card Dataset.” In: *IEEE Access* 12 (2024), pages 169671–169682. https://doi.org/10.1109/ACCESS.2024.3496901.

[20] N. Innan, A. Sawaika, A. Dhor, S. Dutta, et al. “Financial Fraud Detection Using Quantum Graph Neural Networks.” In: *Quantum Machine Intelligence* 6 (2024), page 7. https://doi.org/10.1007/s42484-024-00143-6.

[21] J. Jurgovsky, M. Granitzer, W. Zellinger, S. Calabretto, P.-E. Portier, L. He-Guelton, O. Caelen. “Sequence classification for credit-card fraud detection.” In: *Expert Systems with Applications* 100 (2018), pages 234–245. https://doi.org/10.1016/j.eswa.2018.01.037.

[22] A. Kandala, A. Mezzacapo, K. Temme, M. Takita, M. Brink, J. M. Chow, J. M. Gambetta. “Hardware-Efficient Variational Quantum Eigensolver for Small Molecules and Quantum Magnets.” In: *Nature* 549.7671 (2017), pages 242–246. https://doi.org/10.1038/nature23879.

[23] A. Khoshaman, W. Vinci, B. Denis, E. Andriyash, H. Sadeghi, M. H. Amin. “Quantum Variational Autoencoder.” In: *Quantum Science and Technology* 4.1 (2018), page 014001. https://doi.org/10.1088/2058-9565/aada1f.

[24] O. Kyriienko, E. B. Magnusson. “Unsupervised quantum machine learning for fraud detection.” In: *arXiv* abs/2208.01203 (2022). https://doi.org/10.48550/arXiv.2208.01203. URL: https://doi.org/10.48550/arXiv.2208.01203.

[25] K. Kottmann, F. Metz, J. Fraxanet, N. Baldelli. “Variational quantum anomaly detection: Unsupervised mapping of phase diagrams on a physical quantum computer.” In: *Physical Review Research* 3.4 (2021), page 043184. https://doi.org/10.1103/PhysRevResearch.3.043184.

[26] N. Liu, P. Rebentrost. “Quantum machine learning for quantum anomaly detection.” In: *Physical Review A* 97.4 (2018), page 042315.

[27] N. Liu, P. Rebentrost. “Quantum Machine Learning for Quantum Anomaly Detection.” In: *Physical Review A* 97.4 (2018), page 042315. https://doi.org/10.1103/PhysRevA.97.042315.

[28] J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, H. Neven. “Barren plateaus in quantum neural network training landscapes.” In: *Nature Communications* 9.1 (2018), page 4812. https://doi.org/10.1038/s41467-018-07090-4.

[29] S. Mukkamala, G. Janoski, A. H. Sung. “Intrusion Detection Using Neural Networks and Support Vector Machines.” In: *Proceedings of the 2002 International Joint Conference on Neural Networks (IJCNN'02)*. Volume 2. Honolulu, HI, USA: IEEE, 2002, pages 1702–1707. https://doi.org/10.1109/IJCNN.2002.1007774.

[30] M. A. Nielsen, I. L. Chuang. *Quantum Computation and Quantum Information*. 10th Anniversary Edition. Cambridge University Press, 2010.

[31] G. Pang, C. Shen, L. Cao, A. van den Hengel. “Deep Learning for Anomaly Detection: A Review.” In: *ACM Computing Surveys* 54.2 (2021), pages 1–38. https://doi.org/https://doi.org/10.1145/343995.

[32] G. Park, J. Huh, D. K. Park. “Variational quantum one-class classifier.” In: *Machine Learning: Science and Technology* 4.1 (2023), page 015006. https://doi.org/10.1088/2632-2153/acafd5.

[33] J. Pazem, M. H. Ansari. “Error mitigation in brainbox quantum autoencoders.” In: *Scientific Reports* 15.1 (2025), page 2257. https://doi.org/10.1038/s41598-024-84171-z.

[34] A. Pepper, N. Tischler, G. J. Pryde. “Experimental Realization of a Quantum Autoencoder.” In: *Physical Review Letters* 122.6 (2019), page 060501. https://doi.org/10.1103/PhysRevLett.122.060501.

[35] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, J. L. O'Brien. “A variational eigenvalue solver on a photonic quantum processor.” In: *Nature Communications* 5 (2014), page 4213. https://doi.org/10.1038/ncomms5213.

[36] D. Pranjić, F. Knäble, P. Kunst, D. Kutzias, D. Klau, C. Tutschku, L. Simon, M. Kraus, A. Abedi. “Unsupervised Quantum Anomaly Detection on Noisy Quantum Processors.” In: *arXiv* abs/2411.16970 (2024). https://doi.org/10.48550/arXiv.2411.16970. URL: https://doi.org/10.48550/arXiv.2411.16970.

[37] J. Preskill. “Quantum Computing in the NISQ Era and Beyond.” In: *Quantum* 2 (2018), page 79. https://doi.org/10.22331/q-2018-08-06-79.

[38] Qiskit Community. *Quantum Autoencoder Tutorial*. https://qiskit-community.github.io/qiskit-machine-learning/tutorials/12_quantum_autoencoder.html. Accessed: 2025-12-06.

[39] J. Romero, J. P. Olson, A. Aspuru-Guzik. “Quantum autoencoders for efficient compression of quantum data.” In: *Quantum Science and Technology* 2.4 (2017), page 045001. https://doi.org/10.1088/2058-9565/aa8072.

[40] M. Sakurada, T. Yairi. “Anomaly detection using autoencoders with nonlinear dimensionality reduction.” In: *Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis* (2014), pages 4–11.

[41] M. Schuld, V. Bergholm, C. Gogolin, J. Izaac, N. Killoran. “Evaluating analytic gradients on quantum hardware.” In: *Physical Review A* 99.3 (2019), page 032331. https://doi.org/10.1103/PhysRevA.99.032331.

[42] M. Schuld, N. Killoran. “Quantum machine learning in feature Hilbert spaces.” In: *Physical Review Letters* 122.4 (2019), page 040504. https://doi.org/10.1103/PhysRevLett.122.040504.

[43] M. Schuld, N. Killoran. “Quantum machine learning in feature Hilbert spaces.” In: *Physical Review Letters* 122.4 (2019), page 040504.

[44] M. Al-Shabi. “Credit Card Fraud Detection Using Autoencoder Model in Unbalanced Datasets.” In: *Journal of Advances in Mathematics and Computer Science* 33.5 (2019), pages 1–16. https://doi.org/10.9734/jamcs/2019/v33i530192.

[45] A. Shiravi, H. Shiravi, M. Tavallaee, A. A. Ghorbani. “Toward developing a systematic approach to generate benchmark datasets for intrusion detection.” In: *Computers & Security* 31.3 (2012), pages 357–374. https://doi.org/10.1016/j.cose.2011.12.012.

[46] J. C. Spall. “An Overview of the Simultaneous Perturbation Method for Efficient Optimization.” In: *Johns Hopkins APL Technical Digest* 19.4 (2001), pages 482–492.

[47] M. Tavallaee, E. Bagheri, W. Lu, A. A. Ghorbani. “A detailed analysis of the KDD CUP 99 data set.” In: *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*. 2009, pages 1–6. https://doi.org/10.1109/CISDA.2009.5356528.

[48] K. Temme, S. Bravyi, J. M. Gambetta. “Error mitigation for short-depth quantum circuits.” In: *Physical Review Letters* 119.18 (2017), page 180509. https://doi.org/10.1103/PhysRevLett.119.180509.

[49] P. Vincent, H. Larochelle, Y. Bengio, P.-A. Manzagol. “Extracting and Composing Robust Features with Denoising Autoencoders.” In: *Proceedings of the 25th International Conference on Machine Learning*. 2008, pages 1096–1103.

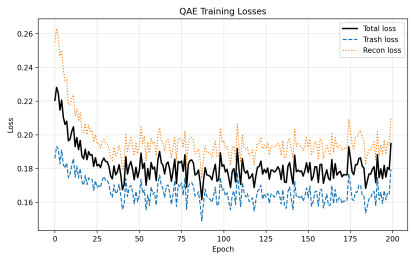
[50] S. Wang, E. Fontana, M. Cerezo, K. Sharma, A. Sone, Ł. Cincio, P. J. Coles. “Noise-induced barren plateaus in variational quantum algorithms.” In: *Nature Communications* 12 (2021), page 6961. https://doi.org/10.1038/s41467-021-27045-6.

[51] J. Zou, J. Zhang, P. Jiang. “Credit Card Fraud Detection Using Autoencoder Neural Network.” In: *arXiv* abs/1908.11553 (2019). https://doi.org/10.48550/arXiv.1908.11553. URL: https://arxiv.org/abs/1908.11553.

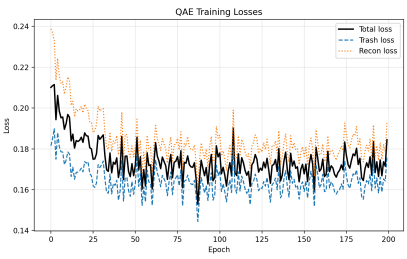
Appendix 1. Github repository code

https://github.com/Danial-yd/Quantum-Autoencoder

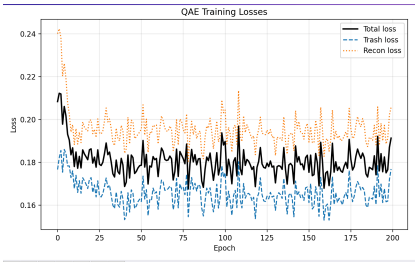
Appendix 2. Training Cost per Epoch of QAE



3 layers

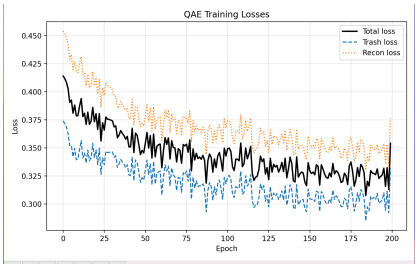


6 layers

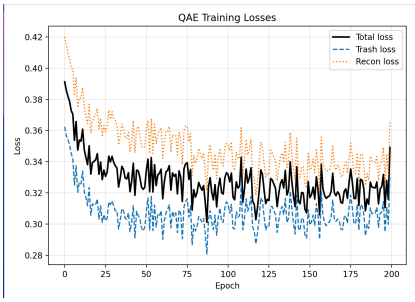


9 layers

19 figure. *Total Loss for the 2-qubit → 1-qubit compression with 3, 6, and 9 layers.*



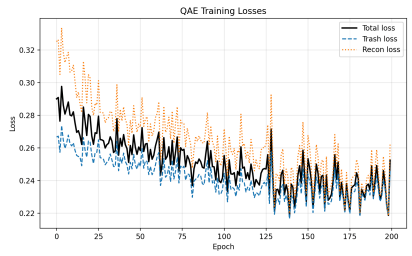
3 layers



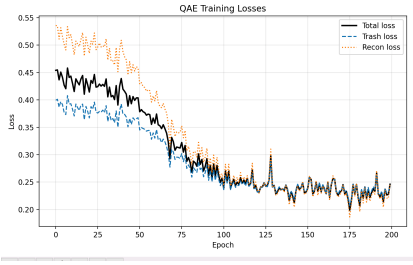
6 layers

9 layers

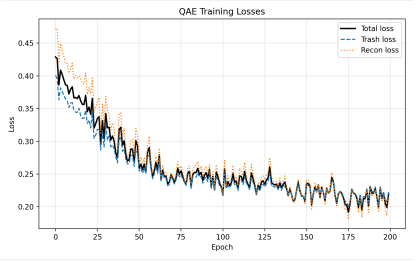
20 figure. *Total Loss for the 3-qubit → 1-qubit compression with 3, 6, and 9 layers.*



3 layers



6 layers

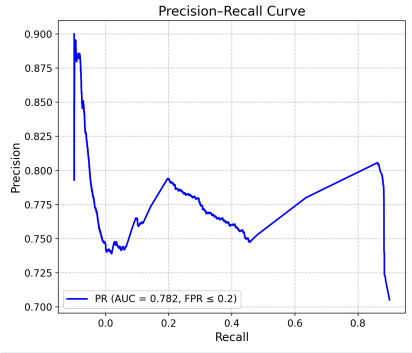


9 layers

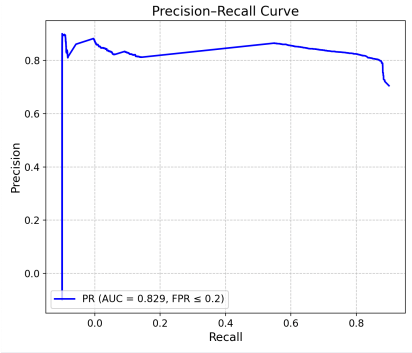
21 figure. *Total Loss for the 3-qubit → 2-qubit compression with 3, 6, and 9 layers.*

Appendix 3. PR-curves

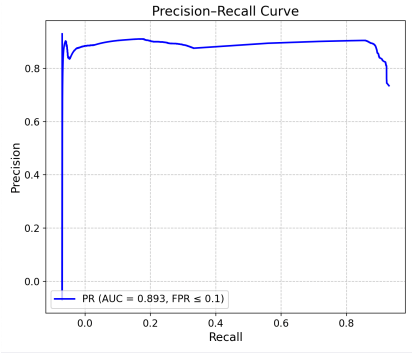
Appendix 3. .1 Classical kNN



2 qubits → 1 qubit



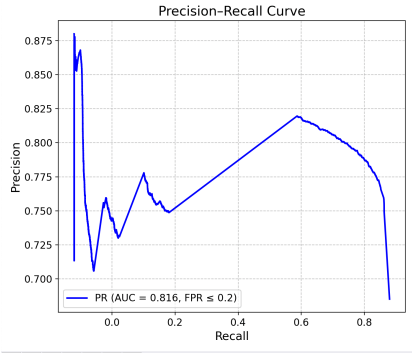
3 qubits → 1 qubit



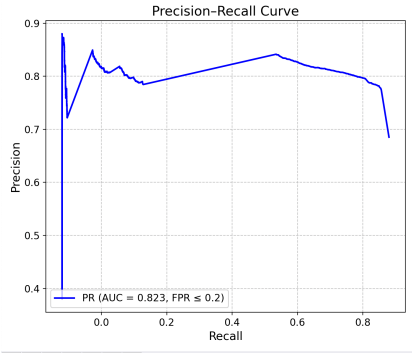
3 qubits → 2 qubits

22 figure. *AUC of Classical kNN*

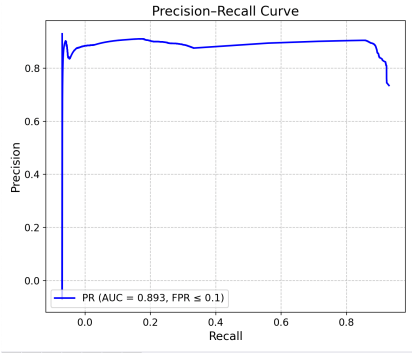
Appendix 3. .2 Quantum kNN



2 qubits → 1 qubit



3 qubits → 1 qubit



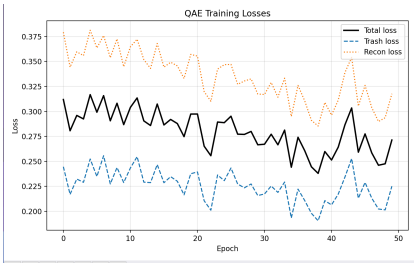
3 qubits → 2 qubits

23 figure. *AUC of Quantum kNN*

Appendix 4. Training Cost per Epoch of BrainBox Layer

2-1-2

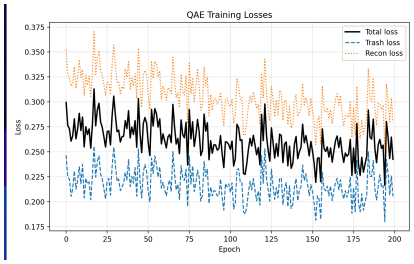
3-2-3



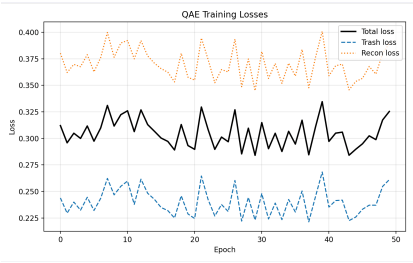
4-3-4

24 figure. *Visualization of Training Loss of BrainBox layer QAE per epoch*

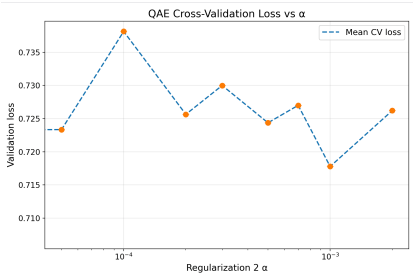
Appendix 4. .1 Training Cost per Epoch of Dropout and L2 Regularization with BrainBox Layer



Dropout Layer (0.2) and Gate (0.4)



L2 Regularization

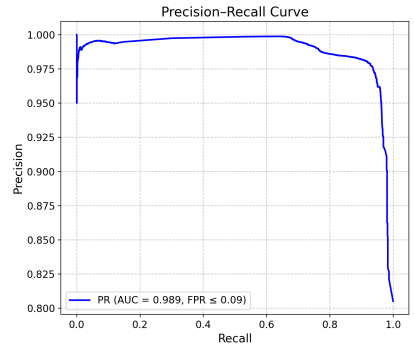


Tuning L2 Regularization

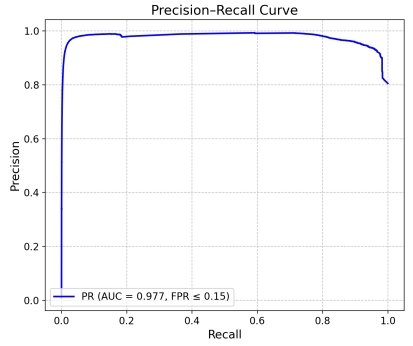
25 figure. *Visualization of Training of Dropout and L2 Regularization cost per epoch and Tuning L2*

Appendix 5. PR-AUC of Dropout and L2 Regularization

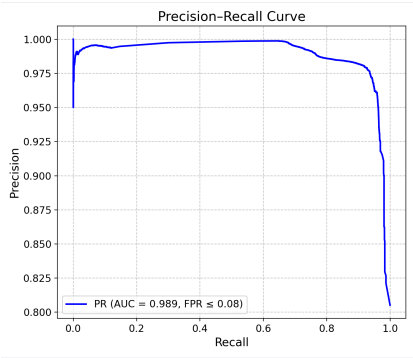
Appendix 5. .1 Classical kNN



3-2-3



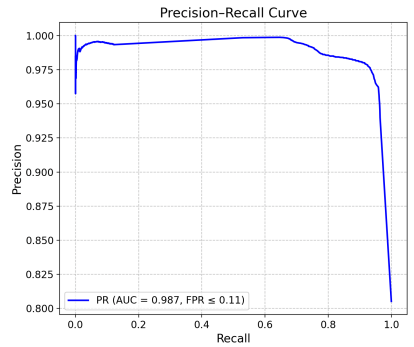
Dropout Layer (0.2) and Gate (0.4)



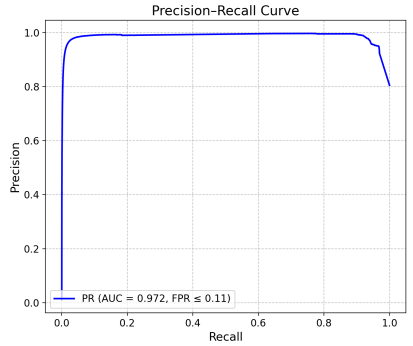
L2 Regularization

26 figure. *AUC of Classical kNN*

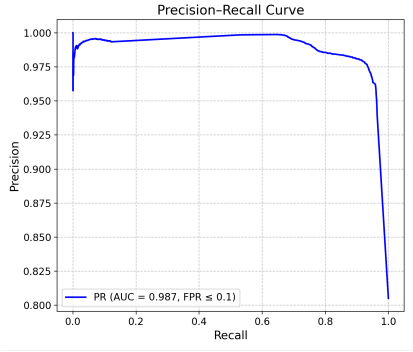
Appendix 5. .2 Quantum kNN



3-2-3



Dropout Layer (0.2) and Gate (0.4)



L2 Regularization

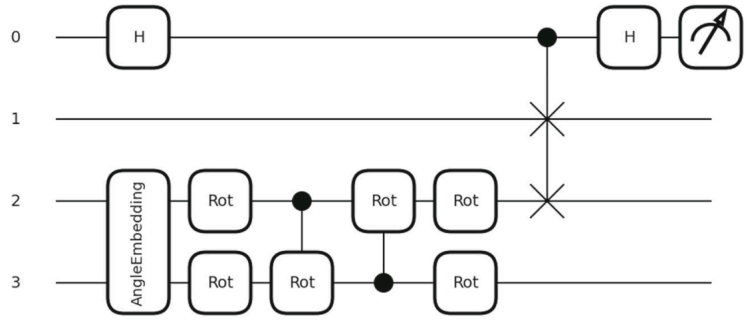
27 figure. *AUC of Quantum kNN*

Appendix 6. Reproduced Results from papers

The experiments used the KDD99 dataset, which contains about 4.9 million network connection records with 41 features and one target label. Approximately 80% of the connections represent attacks, while 20% are normal.

Categorical features such as `protocol_type`, `service`, and `flag` were transformed using one-hot encoding. Principal Component Analysis (PCA) was then applied to reduce dimensionality.

The angle encoding circuit for the Quantum Autoencoder (QAE) consists of two layers of parameterized rotation gates with bidirectional entanglement between two qubits: `TRASH` and `COMP`. Each layer applies rotation operations $R(\alpha, \beta, \gamma)$ on both qubits, followed by two CNOT gates (`CNOT(TRASH, COMP)` and `CNOT(COMP, TRASH)`) to create entanglement.



28 figure. *Circuit diagram of the angle encoding block used in the QAE.*

- The paper trained the model on normal data, and the test set consists only of attacks. However, the model struggled to predict normal points.

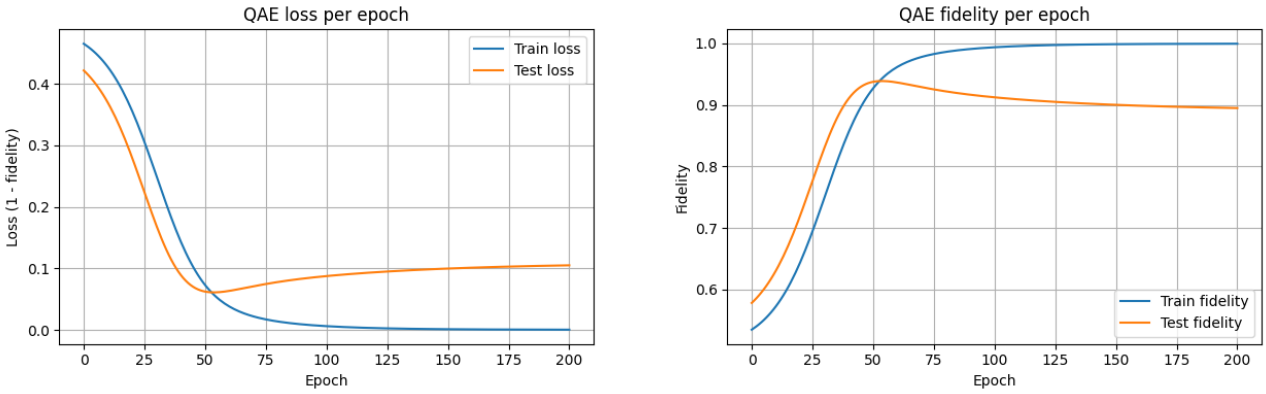
The Quantum Autoencoder (QAE) is trained by minimizing a loss function based on the fidelity between the `TRASH` qubit and a reference qubit. The loss is defined as

$$\mathcal{L}(\theta) = 1 - \mathcal{F}\left(\rho_\theta^{(\text{trash})}, \rho^{(\text{ref})}\right),$$

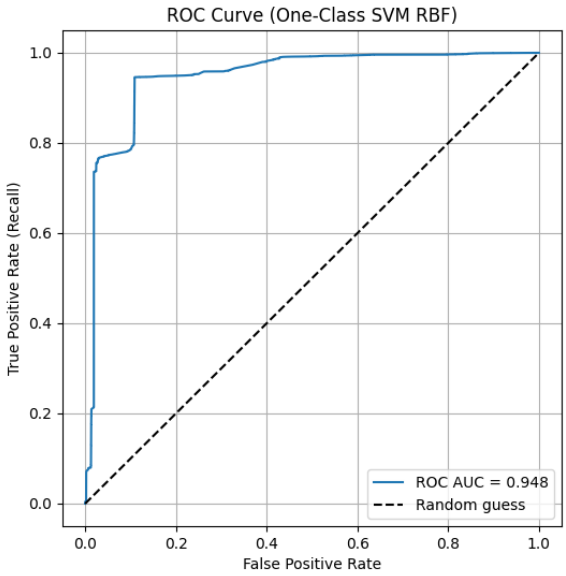
where the fidelity $\mathcal{F}(\rho,\sigma)$ measures the similarity between two quantum states. For pure states $\rho = |\psi_A\rangle\langle\psi_A|$ and $\sigma = |\psi_B\rangle\langle\psi_B|$, the fidelity simplifies to

$$\mathcal{F}(\rho,\sigma) = |\langle\psi_A|\psi_B\rangle|^2.$$

Optimization is performed with the Adam method for **200 epochs**, learning rate **0.01** and batch size **128**, updating θ to minimize the fidelity loss.



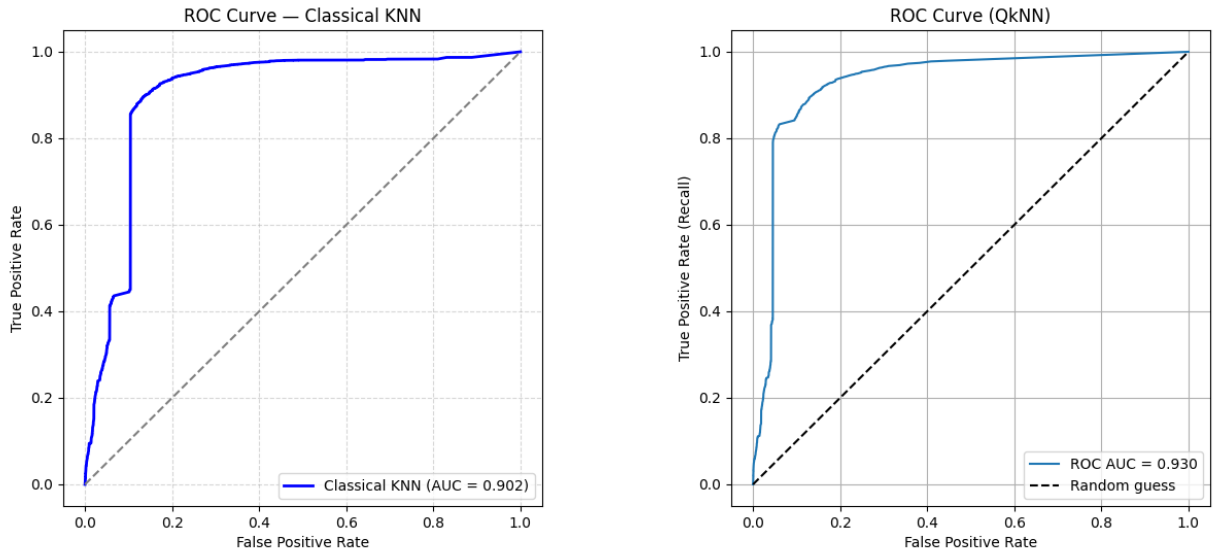
29 figure. *Experimental results of the Quantum Autoencoder.*



30 figure. *ROC Curve AUC - 0.95*

10 table. *Classification report for classes 0 and 1.*

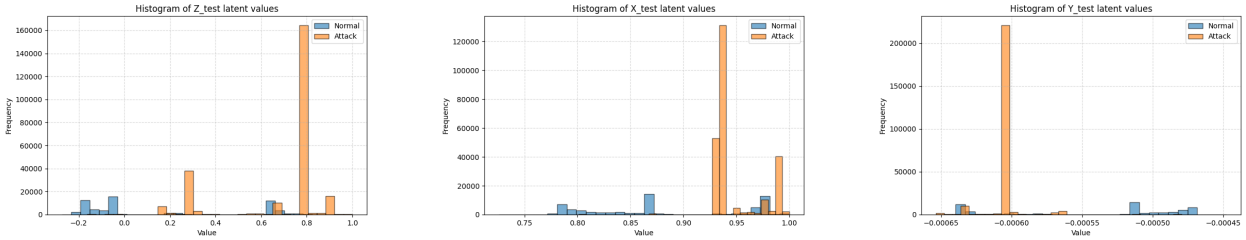
Class	Precision	Recall	F1-score
0	0.5016	0.9692	0.6610
1	0.9904	0.7670	0.8645



31 figure. *Comparison of classical KNN and Quantum KNN*

11 table. *Comparison between Classical kNN and Quantum kNN performance.*

Model	Class	Precision	Recall	F1-score
2*Classical kNN	0	0.7800	0.7800	0.7800
	1	0.9500	0.9500	0.9500
2*Quantum kNN	0	0.7579	0.8040	0.7803
	1	0.9519	0.9379	0.9448

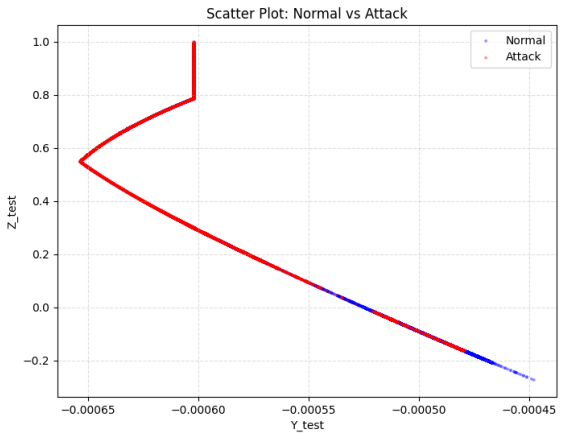


(a) *Pauli-Z* ***(b)*** *Pauli-X* ***(c)*** *Pauli-Y*

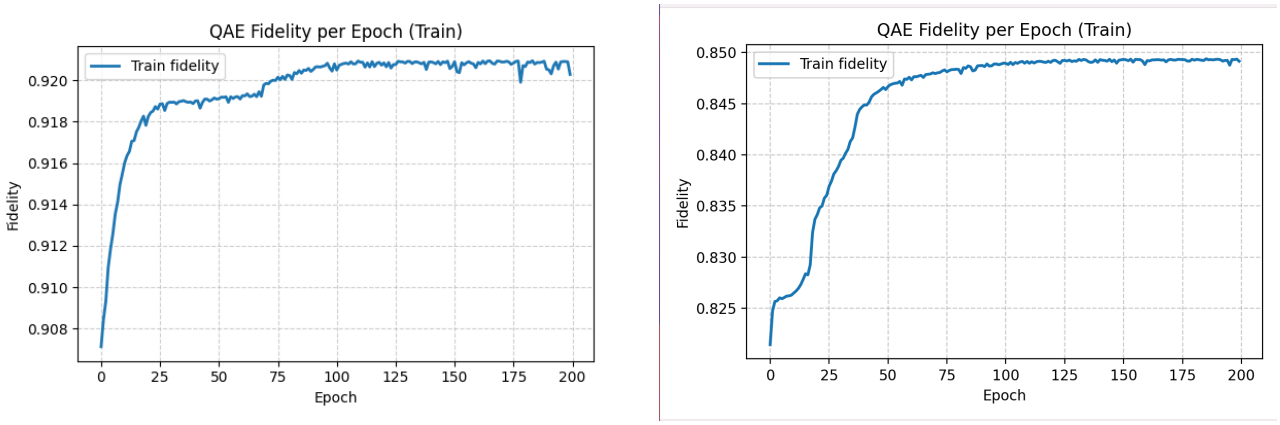
32 figure. *Histogram of different Pauli measurements*

12 table. *Correlation matrix of Pauli measurements (X, Y, Z).*

	Z	Y	X
Z	1.000000	0.468811	-0.803262
Y	0.468811	1.000000	-0.880600
X	-0.803262	-0.880600	1.000000



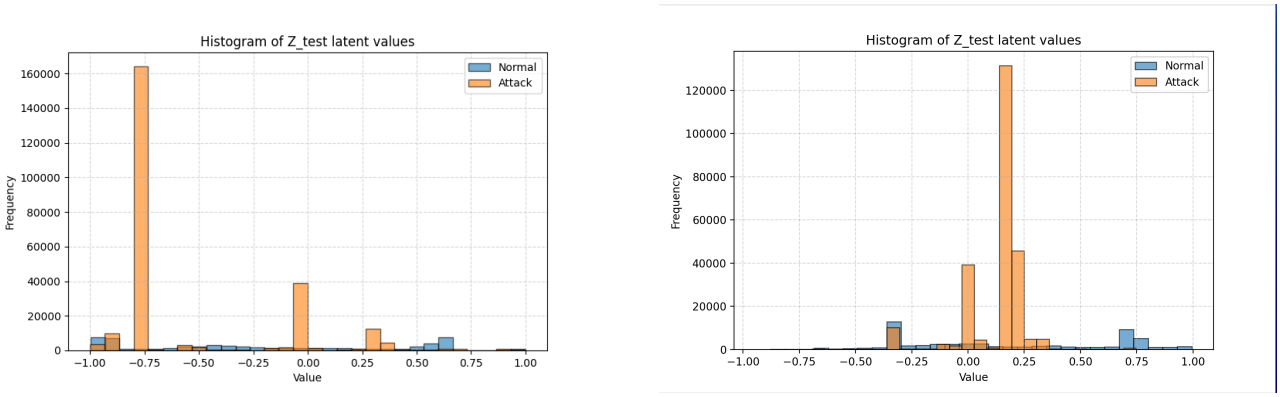
33 figure. *Scatter plot of Pauli Z and Y*



(a) *Compressing 2 qubits* ***(b)*** *Compressing 3 qubits*

34 figure. *Fidelity between trash and reference qubits*

Optimization is performed using the Adam method for **200 epochs**, with a learning rate **of 0.01** and a batch size **of 64**, updating θ to minimize the fidelity loss.



(a) *2 qubits* ***(b)*** *3 qubits*

35 figure. *Histogram of Pauli-Z Test set of 2 qubits and 3 qubits*

Appendix 7. Use of Artificial Intelligence Tools

This appendix describes the use of artificial intelligence (AI) tools during the preparation of this thesis. The goal is to ensure transparency, reproducibility, and a clear statement of the author's own contribution. Only **free-of-charge AI tools** were used.

Appendix 7. .1 AI Tools Used

- **Tool name:** ChatGPT
- **Provider:** OpenAI
- **Version:** GPT-4 (web-based ChatGPT and free access)
- **Internet access:** Yes
- **Period of use:** April 2025 – December 2025

The tool is cited in the references section of the thesis.

Appendix 7. .2 Purpose and Scope of AI Usage

ChatGPT was used as a *support tool* during writing and preparation of the thesis. It was not used to autonomously generate scientific results, experimental datasets, or final conclusions. The main areas of use were:

- **Language support:** rephrasing, grammar fixes, improving clarity (based on author-written text).
- **Structuring:** outlining sections and helping formulate research questions based on author-provided literature summaries.
- **Conceptual clarification:** explaining QML concepts (QAE, variational circuits, SWAP test, latent/trash qubits), later verified against papers/books.
- **LaTeX support:** help with figures, captions, TikZ, equations, table layouts.
- **Python/Qiskit support:** generating template code snippets for circuits, evaluation loops, plotting, etc., which were then edited and verified by the author.

All AI-generated text and code were reviewed and corrected by the author before being included.

Appendix 7. .3 Examples of Prompts Used

ChatGPT is a generative AI tool and produces outputs based on textual prompts. Below are representative prompts used in this thesis, included to allow replication.

Writing and Structuring Prompts

"Please rewrite the following paragraph in a more academic style, but keep the technical meaning unchanged:"

"Based on this Related Work section, can you help me formulate 2–4 research questions that my thesis aims to answer?"

Quantum / Formula-Based Prompts

"Please help me write a LaTeX expression for a QAE cost function based on reconstruction fidelity between ρ_{ref} and $\rho_{\text{out}}(\boldsymbol{\theta})$."

"How can I express SWAP test probability and its relation to overlap $|\langle\psi|\phi\rangle|^2$ in LaTeX for my methodology section?"

Example formula templates that were generated and later adapted/verified by the author:

$$\mathcal{L}(\boldsymbol{\theta}) = 1 - \mathsf{Tr}\left(\rho_{\text{ref}}\rho_{\text{out}}(\boldsymbol{\theta})\right), \tag{26}$$

$$P_{\text{swap}} = \frac{1}{2}\left(1 + |\langle\psi|\phi\rangle|^2\right). \tag{27}$$

Python / Qiskit Prompts

"Can you provide a minimal Qiskit example of a variational ansatz (e.g., RealAmplitudes) and show how to bind parameters and run a statevector simulation?"

"Please generate Python code to compute a SWAP-test circuit for two states and return the measured overlap estimate. Keep it simple and easy to modify."

"Can you write a Python function that trains a small variational circuit using a basic optimizer loop, and returns the best parameters and loss history?"

Appendix 7. .4 Representative Example of AI-Generated Code (Template)

In some cases, ChatGPT generated *template* code snippets that were then edited, debugged, and verified by the author. A representative example is shown below. This code is included as an illustration of the kind of assistance requested; it was not copied blindly and was adjusted to match the experimental setup used in the thesis.

```
# Template example (generated with AI assistance and then modified/verified by the autho
from qiskit import QuantumCircuit
from qiskit.circuit.library import RealAmplitudes
```

```
from qiskit.quantum_info import Statevector
import numpy as np

def build_ansatz(num_qubits: int, reps: int = 2):
    """Simple variational ansatz template."""
    ansatz = RealAmplitudes(num_qubits, reps=reps, entanglement="linear")
    return ansatz

def overlap_statevector(qc1: QuantumCircuit, qc2: QuantumCircuit) -> float:
    """Compute overlap |<psi|phi>|^2 using statevector simulation (template)."""
    sv1 = Statevector.from_instruction(qc1)
    sv2 = Statevector.from_instruction(qc2)
    amp = np.vdot(sv1.data, sv2.data)   # <psi|phi>
    return float(np.abs(amp)**2)

# Example usage
qc_a = QuantumCircuit(2)
qc_a.h(0)
qc_a.cx(0, 1)

qc_b = QuantumCircuit(2)
qc_b.x(1)

print("Overlap:", overlap_statevector(qc_a, qc_b))
```

This kind of code was used only as a starting point. The final implementations used in experiments (circuits, loss functions, training loops, and evaluation code) were developed and validated by the author.

Appendix 7. .5 Authorial Contribution

The author declares that all scientific contributions in this thesis were performed independently, including:

- reproduction of results from recent research papers,
- design and implementation of quantum autoencoder experiments,
- analysis of overfitting behaviour and false positive rates,
- application of recent methods for latent-space denoising and qubit management.

AI tools were used only to support writing, explanation, and formatting. The author takes full responsibility for the correctness, originality, and integrity of the thesis.

Appendix 7. .6 Replicability Statement

The described AI usage can be replicated by using the stated tool version and submitting similar textual prompts during the indicated time period. No datasets, figures, or experimental results were uploaded to the AI tool.