Astronomy and Computing 56 (2026) 101090

Contents lists available at ScienceDirect

Astronomy and Computing

journal homepage: www.elsevier.com/locate/ascom

Full length paper

Comparing Classical and Quantum Deep Learning Techniques for Anomaly Detection of Short-Duration Gamma-Ray Signals ☆

Alessandro Rizzo [a,*], Nicolò Parmiggiani [b], Andrea Bulgarelli [b], Antonio Macaluso [c], Carlo Burigana [d,e], Eric Burns [f], Luca Cappelli [g,h], Vincenzo Fabrizio Cardone [i,j], Farida Farsian [a], Savitri Gallego [k], Giuseppe Murante [g], Giuseppe Sarracino [i], Roberto Scaramella [i,j], Francesco Schillirò [a], Vincenzo Testa [i], Tiziana Trombetti [d,e], Dieter H. Hartmann [m], Carolyn A. Kierans [l], Eliza Neights [l,n], John A. Tomsick [o], Andreas Zoglauer [o]

[a] INAF-Osservatorio Astronomico di Catania, Via S. Sofia 78, Catania, 95123, CT, Italy
[b] INAF-Osservatorio di astrofisica e scienza dello spazio, Via Pietro Gobetti 93/3, Bologna, 40129, BO, Italy
[c] Agents and Simulated Reality Department, German Research Center for Artificial Intelligence (DFKI), Saarbrucken, Saarland, Germany
[d] INAF-Istituto di Radioastronomia, Via Piero Gobetti 101, Bologna, 40129, BO, Italy
[e] INFN Sezione di Bologna, Via Irnerio 46, Bologna, 40127, BO, Italy
[f] Department of Physics and Astronomy, Louisiana State University, Baton Rogue, 70803, LA, USA
[g] INAF-Osservatorio Astronomico di Trieste, Via Tiepolo 11, Trieste, 34131, TS, Italy
[h] Universita degli Studi di Trieste, Dipartimento di Fisica, Via A. Valerio 2, Trieste, 34127, TS, Italy
[i] INAF-Osservatorio Astronomico di Roma, Via Frascati 33, Monte Porzio Catone, 00078, RM, Italy
[j] INFN Sezione di Roma 1, Piazzale Aldo Moro 2, Edificio G. Marconi, Roma, 00185, RM, Italy
[k] Institut für Physik & Exzellenzcluster PRISMA+, Johannes Gutenberg-Universität Mainz, Mainz, 55099, Germany
[l] NASA Goddard Space Flight Center, 8800 Greenbelt Road, Greenbelt, 20771, MD, USA
[m] Clemson University, Department of Physics and Astronomy, Clemson, 29634-0987, SC, USA
[n] George Washington University, 2121 I St NW, Washington, 20052, DC, USA
[o] Space Sciences Laboratory, University of California, 7 Gauss Way, Berkeley, 94720-7450, CA, USA

ARTICLE INFO

Keywords:
Gamma-Ray Bursts
Machine learning
Deep learning
Quantum computing
Quantum machine learning
Anomaly detection
Autoencoder

ABSTRACT

Gamma-Ray Bursts (GRBs) are the most luminous explosions in the universe, and are of great interest in astrophysics research. Despite the development of onboard trigger algorithms, significant obstacles remain in identifying weak events and effectively filtering out false detections. These challenges motivate the exploration of innovative technologies such as deep learning and quantum computing to help traditional triggers as a second verification level and increase reliability across the detection pipeline.

We present a comparison of a classical and a quantum autoencoder running on simulated quantum hardware, tackling an anomaly detection task in a controlled environment. The goal is to compare their performance, with a focus on resource-constrained scenarios involving limited data and trainable parameters. Both approaches were tested using a simulated set of light curves from the anticoincidence shield of the Compton Spectrometer and Imager (COSI) soft gamma-ray telescope. Short-duration GRB signals were simulated using MEGAlib and then modified through data augmentation techniques.

The results indicate that while classical autoencoders reconstruct better, achieving a Mean Squared Error (MSE) as low as 2.97×10^{-5}, the efficiency of the quantum autoencoders is notable with lower numbers of parameters and smaller datasets. Using only 10 trainable parameters and 100 input samples, the quantum model obtained an MSE of 2.24×10^{-4}, outperforming its classical counterpart (MSE of 3.81×10^{-2} with 705 parameters). These results show a promising role for quantum machine learning in astrophysical contexts where data related to GRBs are limited or lightweight model architectures are essential.

☆ This article is part of a Special issue entitled: 'HPC in Cosmology and Astrophysics' published in Astronomy and Computing.

* Corresponding author.
E-mail address: alessandro.rizzo@inaf.it (A. Rizzo).

https://doi.org/10.1016/j.ascom.2026.101090
Received 9 October 2025; Received in revised form 12 February 2026; Accepted 23 February 2026
Available online 4 March 2026

1. Introduction

Quantum computing (Nielsen and Chuang, 2000; Rieffel and Polak, 2011) is an emerging field of computer science and engineering that leverages the principles of quantum mechanics with the goal of solving problems that are beyond the capabilities of even the most powerful classical computers. In recent years, there have been constant advances in this area, such as increased qubit capacity and improved quantum error correction (QEC) techniques, driven by developments in quantum hardware (Roffe, 2019; Bertels et al., 2024; Memon et al., 2024; Gill et al., 2025). This progress is laying the foundation for increasingly reliable and scalable quantum systems in the near future.

However, we are currently in the so-called Noisy Intermediate-Scale Quantum (NISQ) era (Preskill, 2018), where quantum computers lack fault tolerance and are highly prone to errors. The main goal of this discipline, and also one of its most difficult challenges, is to achieve quantum utility, i.e., the point at which a quantum device outperforms a classical computer in solving practical problems.

Several research groups are making significant progress in this direction, introducing various innovations. Google's "Willow" quantum chip[1] (Acharya et al., 2024) and IBM's plan to introduce the first quantum supercomputer[2] are key developments that make quantum computing more accessible, for example, by abstracting complex quantum circuits into simpler functions. In addition, Microsoft recently announced the Majorana-1 quantum chip,[3] based on foundational research published by Aghaee et al. (2024). This could revolutionize quantum computing by using topological qubits that make quantum systems more stable and practical.

Quantum Machine Learning (QML) (Biamonte et al., 2017), and its subfield Quantum Deep Learning (QDL) (Wiebe et al., 2015), can take advantage of this technological progress, aiming to represent an alternative to their respective classical techniques. QML is showing great promise in the field of computing, but also in other scientific domains such as quantum chemistry and high-energy physics (Preskill, 2018). This discipline integrates quantum algorithms into classical machine learning techniques to solve problems more quickly and efficiently than classical approaches (Harrow et al., 2009), exploiting various principles of quantum mechanics such as superposition, entanglement, and quantum measurement. According to different studies, QML already performs very well in certain complex tasks including pattern recognition and anomaly detection, especially in contexts where input data is limited or lightweight models are essential (Schuld and Killoran, 2019; Abbas et al., 2021).

We can find an example of this computational challenge in astrophysics. Gamma-Ray Bursts (GRBs) are among the most energetic astrophysical phenomena observed in the known universe. They consist of short but intense bursts of high-energy gamma rays originating from some of the most violent astrophysical processes. It is believed that most short GRBs, which last no more than 2 s, originate from the merger of neutron stars, which produce gravitational wave signals. A significant demanding task lies in detecting GRBs that originate from weak or distant events (Kumar and Zhang, 2015, Section 6). In fact, although many bright GRBs produce signals that dominate the instrumental background, this is not always the case. Another difficulty consists in separating astrophysical signals from instrumental effects, which can often simulate short-duration events, as well as from the enormous variation in GRB properties. These challenges represent a major obstacle for machine learning techniques, particularly quantum ones. It is therefore important to emphasize that, in this study, we employ a simulated and controlled environment, using a single GRB model. This is obviously an idealized case, but it is well-suited to the exploratory nature of this work.

Previous efforts in GRB detection have leveraged classical machine learning methods. Parmiggiani et al. (2021) introduced a deep learning technique using a Convolutional Neural Network (CNN) to classify AGILE-GRID gamma-ray intensity maps, trained on extensive simulated datasets to distinguish GRBs from background data. These observations originate from the AGILE mission, a satellite launched by the Italian Space Agency (ASI) devoted to high-energy astrophysics (Tavani et al., 2009). This approach demonstrated a significant improvement over traditional methods, particularly in the analysis of short-term observations with low photon counts.

Parmiggiani et al. (2023) developed a CNN-based autoencoder for anomaly detection in the multivariate time series from the AGILE Anticoincidence System (ACS), again training on background-only data to identify GRBs by their reconstruction error, leading to the identification of 72 bursts, 15 of which were not listed in the AGILE-MCAL catalog. Building on this, Parmiggiani et al. (2024) presented an updated deep learning model that predicts AGILE ACS top panel background count rates using satellite orbital parameters as input, detecting GRBs via significant deviations between predicted and acquired rates. This approach allowed for the identification of 39 bursts, four of which were previously undetected by the AGILE team.

Crupi et al. (2023), for the HERMES Pathfinder nano-satellite mission (Fiore et al., 2020, 2021), presented a data-driven approach for detecting GRBs and other high-energy astronomical transients, using a neural network to estimate background count rates, employing a fast change-point and anomaly detection technique to isolate statistically significant excesses from the background. Garcia-Cifuentes et al. (2024) introduced a Python package that applies a machine learning approach to the post-detection classification of GRBs, using the t-SNE algorithm to analyze light curves from the Swift/BAT database.

These studies have already proven that a classical approach is very successful in the context of GRB detection. This work, instead, investigates whether utilizing quantum architectures can offer improvements in certain areas, such as detection accuracy or reductions in model complexity. Similar research on GRB analysis focused on classification tasks. Rizzo et al. (2025) compared the classical networks defined in Parmiggiani et al. (2021) with Quantum Convolutional Neural Networks (QCNNs) applied to the same type of data. This work describes the effectiveness of Quantum Deep Learning in the supervised classification of gamma-ray intensity maps and light curves, highlighting how quantum variational architectures can achieve good accuracy in this classification task, often surpassing their classical counterparts while using fewer trainable parameters and input data. In this study, the QCNNs were trained to perform binary classification between GRB signals and background noise from the AGILE mission. This work was further extended by Farsian et al. (2025), which conducted a benchmark study on QCNNs applied to GRB detection on simulated light curves for the Cherenkov Telescope Array Observatory (CTAO) (Cherenkov Telescope Array Consortium et al., 2019). The authors focused on evaluating different Variational Quantum Circuit architectures to identify the most efficient configuration for temporal signal classification. Also, they explored the impact of different data encoding strategies on the model's ability to extract features from 1D time-series data. These research efforts have demonstrated that Quantum Machine Learning can generalize effectively with a reduced number of trainable parameters and input data, thereby highlighting the potential application of quantum computing to astrophysical data analysis. In this work, we treat this problem as an anomaly detection task, which aims to identify light curves that deviate from the established pattern of background events, thus signaling the potential presence of a GRB signal. We chose to implement this approach rather than a standard supervised classifier because it provides an advantage in terms of dataset requirements. In fact, since GRBs are rare events, it would be difficult to obtain a comprehensive and balanced dataset as required by a supervised model.

[1] https://blog.google/technology/research/google-willow-quantum-chip/

[2] https://www.ibm.com/roadmaps/quantum/2025/

[3] https://news.microsoft.com/source/features/innovation/microsofts-majorana-1-chip-carves-new-path-for-quantum-computing/

Instead, by utilizing an autoencoder for anomaly detection, we reduce the dependence on extensive GRB data.

The scope of this work is limited to short-duration GRBs. However, for brevity, we will refer to them as GRBs in the rest of this work. One of the scientific objectives of the Compton Spectrometer and Imager (COSI) (Tomsick et al., 2019, 2024) satellite, whose simulations we used in our work, is to observe them to further the field of multi-messenger astrophysics.

This paper is divided as follows: Section 2 gives an introduction to machine learning, quantum computing, and how those two disciplines are interconnected. Section 3 describes the dataset used in this work and the implemented models. Section 4 presents the obtained results, comparing the performance of the different approaches. Finally, Section 5 discusses the implications of the main results of this work and proposes potential directions for future research.

2. Methods and algorithms for anomaly detection

Deep Learning (DL), a subset of Machine Learning (ML), uses deep neural networks to handle complex data, attempting to replicate human decision-making. The main strength of DL networks lies in their depth, which can reach over hundreds of layers, allowing for a precise and concise analysis of input data (Goodfellow et al., 2016). These algorithms can learn through two different paradigms. The first is supervised learning, in which the model is trained with previously labeled data. The second is unsupervised learning, which involves identifying patterns and features from raw data and is a fundamental aspect of artificial intelligence because it facilitates the acquisition of information without direct human intervention. For this reason, it is widely used in various applications, such as speech recognition (Liu et al., 2022), image analysis (Liu et al., 2024), and anomaly detection (Berahmand et al., 2024).

2.1. Classical autoencoders

An autoencoder is a type of neural network model in which the network attempts to efficiently compress, or encode, the input data down to its essential features, and then reconstruct, or decode, the original input data from that compressed representation. For a more detailed discussion of classical autoencoders, refer to Appendix A or Goodfellow et al. (2016).

In this work, we applied this machine learning model to an anomaly detection task. This is the process of identifying data points, patterns, or events that deviate from the standard and expected behavior of the input data. These data points, or outliers, are very important because they allow us to identify critical issues such as fraud, system failures, or other anomalies. For this reason, this technique plays a fundamental role in several fields, including cybersecurity (Lunardi et al., 2022), healthcare (Fernando et al., 2021), manufacturing (Yan et al., 2024), and finance (Bakumenko and Elragal, 2022). An autoencoder is the most suitable machine learning model for this type of task because it is trained only on normal data, allowing it to become an expert in reconstructing normal patterns. When we input anomalous data to this model, e.g., a GRB event, it will not reconstruct it correctly, since the key features of the signal are unknown to the trained network. Thus, the reconstruction process will produce a lower quality result with a higher reconstruction error, which will then lead to these data being flagged as anomalies.

2.2. Fundamentals and methods of quantum computing for machine learning

In this study, we implement the quantum counterpart to the classical autoencoder to compare the performance of the two models. Before proceeding, it is important to discuss some of the most relevant concepts of quantum computation, such as qubits, superposition, and entanglement, and their relationship to machine learning. We also discuss quantum circuit operations and the role of these principles in hybrid-quantum classical systems. For a more detailed exposition of these fundamental topics, the reader is referred to Appendix B or Klusch et al. (2024).

At the end of this section, we identify the baseline quantum autoencoder (QAE) and its main components used in this work.

2.2.1. Quantum computing

Quantum computing leverages quantum mechanical principles for computation. To perform operations on quantum information, we need quantum circuits, composed of quantum gates. These architectures represent the quantum analog of classical Boolean circuits (Nielsen and Chuang, 2010).

The fundamental component of a quantum circuit is the qubit, representing the quantum counterpart of a classical bit. A qubit may be in the state $|0\rangle$ or $|1\rangle$, akin to its classical counterpart, but can also exist in a superposition of states. This is expressed as $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, where α and β are complex numbers satisfying the condition $|\alpha|^2 + |\beta|^2 = 1$. The n-qubit quantum state space can scale exponentially, up to as 2^n.

Another very important principle of quantum computing is entanglement, a quantum phenomenon where multiple qubits can become correlated with each other, thus making their individual states indistinguishable in order to create complex relationships between them.

We must perform operations on those qubits to enable computation. This is achieved through operators that delineate the evolution of the quantum state via a unitary transformation, a process that is reversible and ensures the preservation of state normalization. After implementing these operators, to extract information from a quantum system we need to perform a measurement, which is a probabilistic process. When applied to a qubit in superposition, the measurement forces it to collapse into one of its defined ground states, making the result accessible as classical information.

Today's hardware limits the computational potential of these quantum principles. In fact, the number of qubits, short coherence times, and low gate fidelity are constraints for current quantum computers (Preskill, 2018). For this reason, until we are in the NISQ era, hybrid quantum–classical computing has found widespread use, showing great promise for machine learning applications. These models use variational quantum algorithms (VQAs) that implement quantum circuits with trainable parameters, known as parameterized quantum circuits (PQCs) (Cerezo et al., 2021), that are then optimized using classical optimization algorithms.

Quantum embedding is the first step in implementing a VQA. In this phase, classical data is encoded within a quantum state, thus defining the expressive power that the quantum model can offer. In this work, we used amplitude encoding, a technique that allows for exponential compression of input data. It encodes a classical N-dimensional vector x into the amplitudes of a normalized quantum state with n-qubits, where $n = \lceil \log_2(N) \rceil$. We chose this technique because its scaling is particularly advantageous when dealing with large-dimensional data, such as that considered in this study. A more detailed introduction of this embedding process can be found in Appendix B.1. Following the encoding process, the quantum state is processed by the network's PQC. The output of this circuit is then measured to obtain information in a classical state, which will then be evaluated by a classical computer through a hybrid optimizer that will update the circuit's parameters to minimize the chosen cost function.

2.2.2. Quantum autoencoder

As introduced at the beginning of Section 2.2, we implement a Quantum Autoencoder (QAE) for anomaly detection on light curves, in order to identify potential GRB signals by training the model on background-only light curves. We then compared the results with the classical counterpart.

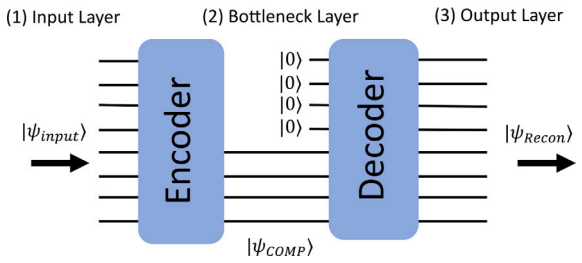


Fig. 1. Representation of a quantum autoencoder. One can notice the similarities with the classical autoencoder, with the circuit having an input state, a bottleneck state, and an output state.
Source: Figure adapted from https://qiskit-community.github.io/qiskit-machine-learning/tutorials/12_quantum_autoencoder.html.

The goal of a quantum autoencoder, defined in Romero et al. (2017), is to reduce the dimensionality of the input in the same way as the classical one. The only difference in this case is that the input is a quantum state. Fig. 1 shows a representation of this model.

This reduction is achieved by a unitary encoding that maps the input to a smaller latent space, followed by a unitary decoding that attempts to reconstruct the initial state from this compressed representation. The goal of training is to maximize the fidelity between the input and the reconstructed quantum state in the output. The circuit of a QAE consists of an input layer that compresses the state, an encoder implemented via a PQC, a bottleneck layer in which some qubits are initialized or traced out, and another PQC decoder that uses an ancilla qubit to reconstruct the original state. For a detailed description of the original QAE architecture, including the mechanism involving reference and auxiliary qubits, the state transformations, and the specifics of the loss calculation on the bottleneck layer, please refer to Appendix C or Romero et al. (2017).

In addition to the standard QAE architecture, we also implemented and tested an improved version based on Feature Vector Encoding (Bravo-Prieto, 2021). This technique modifies the quantum circuit by dynamically setting the rotation angles of the gates based on classical features of the input data, such as the mean and standard deviation of the light curves dataset. A detailed explanation of this method is provided in Appendix C.2.

Similar works employing quantum autoencoders include Ngairangbam et al. (2022), which investigated VQCs for anomaly detection in Large Hadron Collider data. Frehner and Stockinger (2024) explored quantum autoencoders for time series anomaly detection across various benchmark datasets, analyzing reconstruction errors and latent space representations.

3. Methods and experimental setup

This section describes the experimental settings used to implement and evaluate classical and quantum autoencoders. In the following paragraphs, we detail the simulation process employed to generate background and GRB light curves, which will be used for training and evaluating the models, then we analyze the key characteristics of the resulting dataset, and finally address the various pre-processing steps, such as normalization, necessary to feed the data into the implemented architectures.

3.1. Datasets

To train, test, and evaluate the DL models, we simulated two separate datasets: one composed of background-only light curves and the other containing background and GRB events. The former is mainly used during the training phase to learn the best parameters for the model, while the latter is reserved for evaluating the architecture's performance and testing its ability to successfully identify anomalies.

3.1.1. Background-only light curves

To generate the datasets for this research, we relied on simulations for the Compton Spectrometer and Imager (COSI) satellite (Tomsick et al., 2019, 2024). This is a soft gamma-ray survey telescope, operating in the 0.2–5 MeV range, designed to investigate the origins of Galactic positrons, reveal the sites of nucleosynthesis in the Galaxy, carry out innovative studies of gamma-ray polarization, and find counterparts to multi-messenger sources.

Its germanium detectors (GeDs) have an instantaneous field of view of >25% of the sky and exceptional energy resolution (~1%). The detectors are surrounded on the sides and bottom by Bismuth Germanate scintillator (BGO) active shields to reject background charged particles. Those are also utilized for GRB detection through an onboard triggering algorithm. When this system identifies a burst, the satellite promptly transmits data packets to ground stations via the Tracking and Data Relay Satellite System (TDRSS), a dedicated real-time analysis pipeline will then examine it more closely to confirm and get more details about the event, including localizing the GRBs in the COSI field of view to <2.5° accuracy. It is at this stage that the models discussed in this work could be useful. For example, they could provide important secondary verification of the detections made by the onboard trigger, allowing for more reliable GRB alerts before further analysis is initiated. More details about the COSI pipeline can be found in Zoglauer et al. (2024).

COSI's survey mode is characterized by a periodic rocking of its pointing axis, which shifts between 20° North and 20° South of the Earth's zenith over a 12-hour cycle. The background light curves for our models were generated using the COSI BGO background dataset simulated for the Data Challenge 3[4] (Zoglauer et al., 2021; Martinez, 2023; Karwin et al., 2025). This dataset contains 24 h of simulated BGO data, consisting of 14 COSI orbits. However, after discarding the initial orbits due to instrument activation and removing passages through the South Atlantic Anomaly, the remaining data were insufficient for robust model training and evaluation. We therefore selected a 1000-second time window from a single orbit position that represents the average count rate during the orbit and binned the background counts using 1-second bins. This approach is justified because the instrumental background, while dependent on the satellite's pointing and orbital position, can be considered constant over the short timescale of a GRB event, which typically lasts only a few seconds. Once the pointing is set, the satellite's environment does not change significantly during the brief moments in which the burst is visible. We fitted the distribution of these count rates in the defined time window with a Gaussian distribution (Fig. 2), evaluating the fit with the χ^2 test. We obtained a reduced chi-squared (χ^2/ndof) of 0.98, which confirms the goodness of the fit. The obtained mean is 1308, and the standard deviation is 58. We can use the fitted parameters to simulate new background data by sampling from a Gaussian distribution with the same parameters.

During the training, the background-only dataset is split into training and validation datasets. The training dataset allows the model to learn the background data patterns. We then introduced an additional set of 1×10^4 background-only light curve samples for this evaluation phase. This dataset has been generated following the same distribution of the data used for the training process.

A schema of the configurations used for these tests is presented in Table 1. We arbitrarily chose the number of samples generated, limiting ourselves to three specific dataset sizes: 100, 1000, and 5000 samples. By using only 100 samples, we observed the behavior of the models with a very small dataset, thus obtaining information about their performance in scenarios with limited data availability. On the other hand, by using 5000 samples, we tested the model's ability to generalize when trained on a larger dataset. We opted to limit the dataset sample size to 5000 due to the computational constraints of the quantum model training process.

[4] https://github.com/cositools/cosi-data-challenges

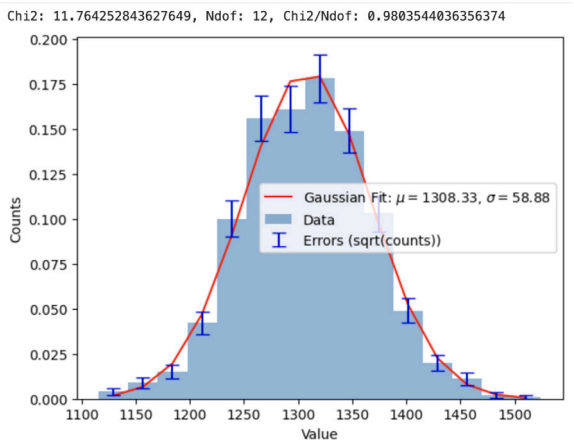


Fig. 2. Gaussian fit to the background count distribution from the COSI BGO_Z1 detector.

Table 1
Used configurations for both the classical and quantum models.

Samples	Training set	Validation set	Test set	GRB set	Bins
100	80	20	1×10^4	1×10^4	32
	80	20	1×10^4	1×10^4	128
1000	800	200	1×10^4	1×10^4	32
	800	200	1×10^4	1×10^4	128
5000	4000	1000	1×10^4	1×10^4	32
	4000	1000	1×10^4	1×10^4	128

We normalized the datasets to facilitate faster and more efficient training for the autoencoder models. We applied Min–Max scaling to the entire datasets, a normalization technique that transforms data features into the range [0,1]. This method adjusts the scale of each feature through a linear transformation, ensuring that the minimum value becomes 0 and the maximum value becomes 1. For a given value x of a dataset X, the normalized value x' is calculated using the formula:

$$x' = \frac{x - \min(X)}{\max(X) - \min(X)}. \tag{1}$$

This technique preserves the relationship between data points, ensuring the structure of the dataset remains intact. It is important to note that the normalization factors are computed on the whole dataset, and not for each light curve. However, the normalization procedure must be different for the quantum approach, as explained in Section 3.3.

As shown in Table 1, we tested several datasets with different numbers of time bins: 32 and 128. This was done to evaluate how the model performs with data of varying dimensions, in terms of efficiency and computational complexity. To modify the number of bins, we divided both the mean and standard deviation of the original fitted Gaussian for 1/0.05 for the 128-bin case and 1/0.15 for the 32-bin one. In this way, we reduced the bin time of the light curve from the initial one-second value.

3.1.2. GRB light curves

We created a dataset of GRBs to evaluate our trained model by summing randomly generated background light curves with GRB signals. The latter are generated starting from the simulation of the GRB 180703B using the MEGAlib software (Zoglauer et al., 2006), with the DC3 COSI mass model. We extracted the light curve template from the Fermi-GBM GRB catalog (von Kienlin et al., 2020). While focusing on a single GRB signal can be considered a limitation, this is an exploratory study, aiming to provide a direct comparison between classical and quantum DL mechanics on a consistent dataset.

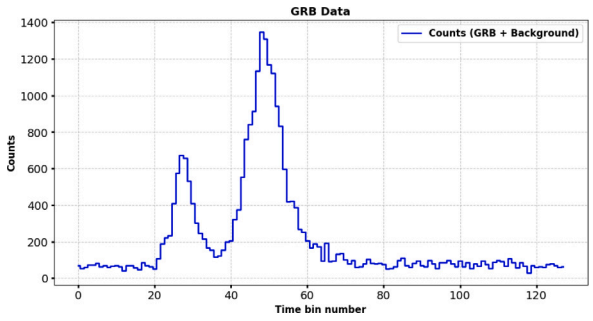


Fig. 3. GRB light curve used for the simulations.

As shown in Fig. 3, this GRB is remarkably strong, exhibiting two peaks of photon counts, where the main one reaches nearly 1400 counts.

We calculated the statistical significance of each source using the Li&Ma method (Li and Ma, 1983). By extracting the *t_on* counts (photons detected during the GRB event window) from the signal window and the *t_off* counts (photons detected during a reference period containing only background noise) from the background window, we obtained an exceptionally high significance for the event. This value suggests a very strong deviation from the background noise, providing clear evidence for the presence of a GRB.

We implemented a data augmentation technique to generate a dataset from the only GRB template available. This was essential to run a full inference on the trained models. For each new sample, the counts of each time bin of the original curve are multiplied by a random factor, extracted from a uniform distribution, which introduces a variable increase between 0% and 10%. In this way, we diversify the samples by adding independent stochastic noise into each bin.

This process was repeated to generate a number of GRB light curves matching the number of time series in the test set,[5] which facilitated a comparison of final results.

We divided the counts of each event by a reduction factor of 30 to reduce the significance of the GRB and obtain a more complex dataset, since the signal becomes less distinguishable from the background noise. In this way, we obtained a mean statistical significance calculated with the Li&Ma method (Li and Ma, 1983) on the whole dataset equal to 7.12 for the 32-bin case, as shown in Fig. 4. As illustrated in Fig. 5, to compute the statistical significance, we made use of the counts within the red-shaded area for the GRB signal, while the blue-shaded part gives us the background count for the comparison in the formula:

$$S = \sqrt{2\left[s \ln\left(\left(\frac{1+\alpha}{\alpha}\right)\frac{s}{s+b}\right) + b \ln\left((1+\alpha)\frac{b}{s+b}\right)\right]}, \tag{2}$$

where $\alpha = t_{on}/t_{off}$, s is the total number of counts in the on-source region, and b is the number of counts in the background region. The variable S then represents the statistical significance of the observed signal, quantifying how unlikely it is to be a random background fluctuation.

Fig. 6 illustrates the signal's composition across three different plots: the background-only light curve, the GRB signal, and the light curve resulting from their sum.

The GRB signal was introduced at 1/6 of the length of the original background light curve. In the case of a 128-bin light curve, the GRB was introduced after the 20th bin. This process is crucial because it allows us to replicate the data an observer would actually measure. On the other hand, we showed the scenario for the 32-bin light curve in Fig. 7. In this case, both the background and the GRB signal were defined with the same number of bins. Since we know the GRB is located in the first half of the signal, we started again the GRB at 1/6

[5] 1×10^4 samples.

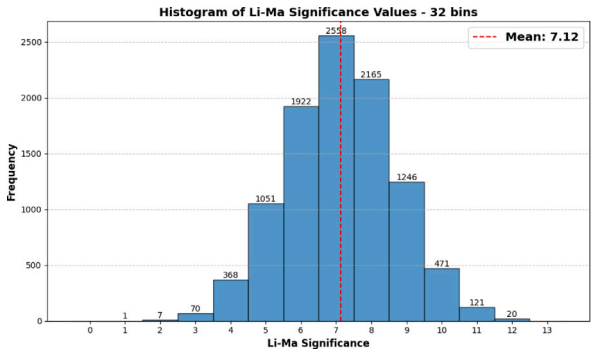


Fig. 4. Histogram of the obtained significance with the Li&Ma method for each simulated GRB in the dataset.

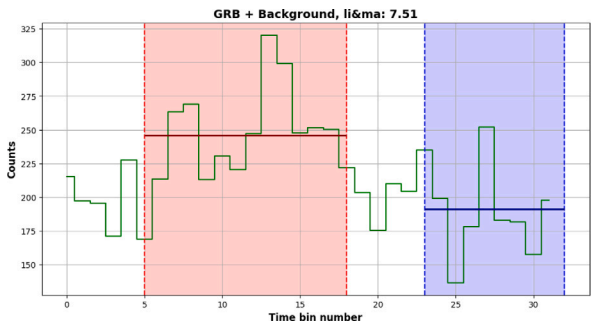


Fig. 5. Example of a GRB light curve composed of 32 bins with the highlighted windows for the computation of the significance with the Li&Ma method.

of the background light curve, and summed the remaining bins up to the 32nd, ignoring the last ones of the GRB event, as they are close to zero. This allows us to know what the correct windows are to compute t_{on} and t_{off} for the Li&Ma method. While this approach might train the model to recognize the signal's location, this is not a limitation for our intended application. It is important to recall that we propose this method as a secondary check, used only after an initial onboard trigger has already detected a GRB and its approximate position in the time series has been determined. As described in Parmiggiani et al. (2026), once the onboard trigger identifies a burst, the relevant data are transmitted to the ground for confirmation and localization. The system provides the specific trigger time (T_0) of the event, meaning this methodology does not imply a blind search. This allows us to place the GRB in a fixed position within the time series in order to standardize the evaluation of the autoencoder's reconstruction fidelity. This setup enabled us to focus on the model's performance in signal reconstruction and anomaly detection.

Finally, we normalized this dataset in the same manner as with the background dataset, using the same normalization values to keep the background at the same level.

3.2. Models

In this section, we outline the models used to address this anomaly detection task. First, we present the classical model, detailing its deep learning architectures and hyperparameters, in order to establish a meaningful basis for comparison. Subsequently, we introduce its quantum counterpart, describing the ansaetze we used, the encoding of classical data into quantum states, and the metrics employed.

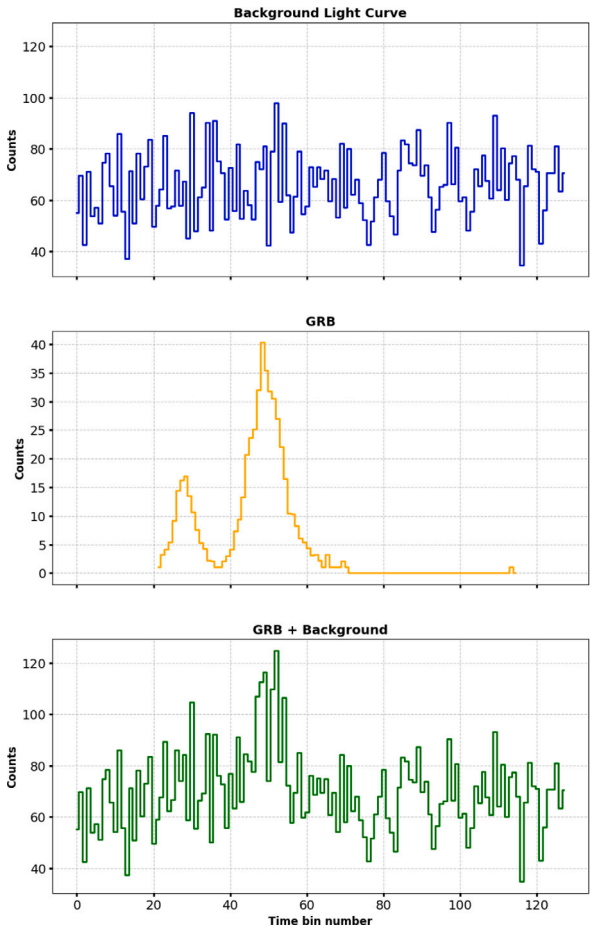


Fig. 6. Background, GRB, and resulting light curves for the 128-bin case. The top panel shows a 128-bin light curve representing the estimated background noise. The middle panel represents the signal model of the GRB, while the bottom panel shows the simulated light curve, created by summing the GRB signal into the background at an offset of 1/6 of the total time window.

The implementation of the classical and quantum models described in this work is available in a GitHub repository upon request.[6]

3.2.1. Classical approach

We implemented a deep learning classical autoencoder using 1D convolutions to process time series data as input, as also discussed in Parmiggiani et al. (2023). Additional details can be found in Goodfellow et al. (2016). This model efficiently extracts local features from adjacent data points, which then serve as fundamental building blocks for the autoencoder to learn a more comprehensive, hierarchical representation.

This architecture is composed of two 1D convolutional layers for the encoder and three transposed 1D convolutional layers for the decoder. The former extracts spatial and temporal features from the input time series data, reducing its dimensionality while preserving essential patterns. On the other hand, the decoder is the inverse of a 1D convolution, and it increases the length of the input sequence rather than reducing it, making it particularly useful for upsampling and decoding, as required in this case.

We used 5 as the kernel size, while the filters are changed to obtain models of varying complexity, in order to compare them with

[6] Please contact the corresponding author at alessandro.rizzo@inaf.it to request access to the GitHub repository.

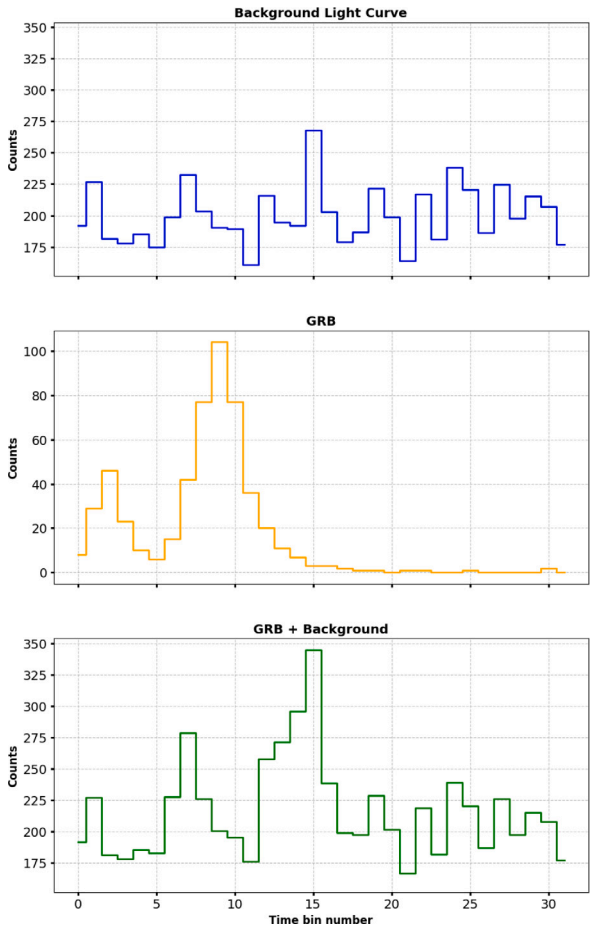


Fig. 7. Background, GRB, and source light curves for the 32-bin case. The description of each plot is the same as in Fig. 6.

their quantum counterparts. Each layer is followed by a dropout layer with a value of 0.2 to avoid overfitting. Those hyperparameters have been chosen by building upon previous work, such as Parmiggiani et al. (2023). We used the Rectified Linear Unit (ReLU) as activation function, which introduces non-linearity into the model by returning the input directly if it is positive, and zero otherwise, defined as $f(x) = \max(0, x)$.

The model was trained for 100 epochs with Early Stopping enabled, a regularization technique that monitors the validation loss during training and halts the process if the model's performance on the validation set does not improve after a specified number of consecutive epochs, preventing overfitting. As the loss function, we used the Mean Squared Error (MSE), defined as:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2, \tag{3}$$

where n is the total number of data points, y_i is the actual value of the ith data point, and $\hat{y}_i$ is its predicted value by the trained model.

For the training process, we used the ADAM optimizer (Kingma and Ba, 2017), setting the learning rate to 1×10^{-4}. We chose 16 as the batch size for the training process.

Fig. 8 shows the structure of an architecture composed of approximately 4×10^4 parameters. The larger model tested contains 6.5×10^5 parameters, while the smallest models have only 705 and 1081 trainable parameters, respectively.

We implemented these architectures using Keras (Chollet et al., 2015) with TensorFlow (Abadi et al., 2015) as the backend.

Layer (type)	Output Shape	Param #
conv1d (Conv1D)	(None, 64, 32)	192
dropout (Dropout)	(None, 64, 32)	0
conv1d_1 (Conv1D)	(None, 32, 64)	10,304
dropout_1 (Dropout)	(None, 32, 64)	0
conv1d_transpose (Conv1DTranspose)	(None, 64, 64)	20,544
dropout_2 (Dropout)	(None, 64, 64)	0
conv1d_transpose_1 (Conv1DTranspose)	(None, 128, 32)	10,272
dropout_3 (Dropout)	(None, 128, 32)	0
conv1d_transpose_2 (Conv1DTranspose)	(None, 128, 1)	161

Total params: 41,473 (162.00 KB)

Trainable params: 41,473 (162.00 KB)

Non-trainable params: 0 (0.00 B)

Fig. 8. Structure of the classical deep learning model composed of 4×10^4 trainable parameters. This table provides a layer-by-layer summary of the model, showing the sequence of operations, the transformation of data's shape, and the parameter count at each step.

3.2.2. Anomaly score

The anomaly score represents a quantitative measure of how much a data point, in this case a light curve, deviates from expected behavior. For autoencoders, this measure corresponds to the reconstruction error between the original inputs and the outputs reconstructed by the model.

In our work, to mitigate potential boundary effects, we excluded from the ends of each light curve three edge bins for the 32-bin case and 11 for the higher-dimensionality curve. We then calculated the squared error for each light curve between the original and reconstructed data points for the remaining bins. Finally, we summed the differences between the input light curve and its autoencoder reconstruction across all bins to distinguish the anomalous time series.

We set the normal reconstruction baseline by analyzing the distribution of total errors from the training data. To find the critical anomaly threshold, we found a balance between minimizing false positives (on the test set) and false negatives (on the GRB set). Thus, a light curve from the train, test, or GRB set will be classified as an anomaly if its absolute reconstruction error exceeds this empirically determined threshold. This was done for both the classical and quantum approaches.

3.3. Quantum approach

We implemented a quantum autoencoder (QAE) as introduced in Section 2.2.2 and expanded in Appendix C, composed of a PQC. To evaluate the impact of model depth, we tested configurations with both a single and three layers. The former offers significant advantages in computational efficiency and training speed due to its reduced parameter count, and despite its lower theoretical capacity, we found this simpler configuration to be effective at learning the key patterns within our data. In contrast, the deeper circuit presented some optimization challenges, even though it provided greater expressive power. We did not go beyond this number of layers due to computational complexity and training time constraints, as well as the model beginning to overfit with increased depth. The number of qubits, as we will explain shortly, depends on the size of the input data and the type of embedding chosen to represent classical data in a quantum state.

To implement the quantum circuits, we used the Qiskit Framework (version 1.3.2) offered by IBM (Javadi-Abhari et al., 2024), which

also allowed us to implement some PQCs already available in its library. We used the quantum simulators offered by Qiskit Aer,[7] a high-performance quantum computing simulator module that is part of the Qiskit environment. It provides interfaces to run quantum circuits with or without noise using multiple different simulation methods, including *Statevector Simulator*, which is the one used in this work. This method computes the full quantum state of a circuit, enabling precise simulations of quantum algorithms without noise. Qiskit Aer also supports GPU acceleration. All runs in this work for the quantum models have been executed on an NVIDIA RTX 4000 SFF Ada Generation GPU and a dual-socket AMD EPYC 9654 CPU.

Preparing data for quantum state encoding requires a specific preprocessing procedure, distinct from methods employed for classical machine learning models, such as Min–Max normalization. The mandatory step for quantum compatibility is normalizing each sample vector x to have a unit L_2 norm, satisfying the condition $\|x\|^2 = 1$, for the state to be physically valid and represent a probability distribution, as described in Appendix B.1. Applying this unit norm normalization was the essential data preparation carried out before the quantum encoding stage in this work.

To optimize the training process, we used the Constrained Optimization By Linear Approximation (COBYLA) optimizer (Powell, 1994, 1998, 2007), which is a numerical optimization method for constrained problems where the derivative of the objective function is not known. This optimizer does not rely on gradients to update the parameters, but approximates the objective function using linear models.

After training, we performed inference to verify whether the model correctly identified GRB light curves as anomalies, similarly to what we did for the classical approach, as described in Section 3.2.2.

Our initial explorations involved the computation of exact final state vectors. From these, we could directly derive fidelity metrics using the SWAP Test, as discussed in Appendix C.1, and MSE values. The process involved parameterizing the quantum gates, binding their parameters, and omitting the final measurement step to maintain a pure state vector representation rather than collapsing it into a set of classical outcomes. After conducting tests with this particular setup, we shifted our main focus to an alternative method that will be detailed shortly, as it proved to be more efficient for achieving high-fidelity light curve reconstructions.

To reconstruct the data, we again applied the chosen parameterized circuit for the encoder as its inverse in the decoder part of the network. This inverse transformation is implemented by applying the conjugate transpose of each gate in the encoder circuit in reverse order, effectively reversing the unitary operations performed during the encoding step. For example, a rotation gate $R_y(\theta)$ in the encoder would be inverted as $R_y(-\theta)$ in the decoder.

The alternative approach we adopted to reconstruct the light curves utilized the Sampler[8] primitive offered by Qiskit. The Sampler's core function is to execute quantum circuits and estimate the probability distribution of their measurement outcomes, effectively simulating the process of sampling from the final quantum state. This function returns quasi-probabilities, which are estimates of measurement probabilities derived from a finite number of shots.

In our specific application, this output distribution directly gives the probability p_k for each of the $2^7 = 128$ or $2^5 = 32$ computational basis states corresponding to our light curve bins ($k = 0$ to $k = 127$ or $k = 32$, for $N = 7$ or $N = 5$ respectively, where N represents the number of qubits). Following our methodology, the reconstructed value for each light curve bin k is then determined by taking the square root of its Sampler-derived probability, $\sqrt{p_k}$.

[7] https://qiskit.github.io/qiskit-aer/

[8] https://docs.quantum.ibm.com/api/qiskit-ibm-runtime/sampler-v2

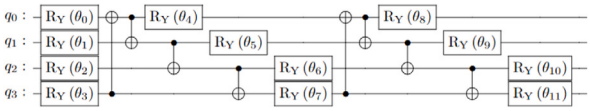


Fig. 9. Example of Real Amplitude ansatz implemented via Qiskit. *Source:* Figure adapted from Yoffe et al. (2023).

3.3.1. Type of ansatz

An ansatz is essentially a PQC, consisting of various quantum gates with tunable parameters. In this work, we implemented and tested several PQCs, differing in the types of quantum gates they include and the number of required trainable parameters.

Some of the ansaetze used in this study were derived from the work presented in Sim et al. (2019). Other tested ansaetze were obtained from the Qiskit library, such as *Real Amplitudes* (an example is shown in Fig. 9) and *Efficient SU2*. Both belong to the n-local circuit class implemented in Qiskit, which is designed to provide a flexible and efficient parameterization of quantum circuits.

The structure of these models consists of alternating rotation and entanglement layers. In the rotation layer, the gate blocks are applied sequentially, stacked on top of each other, while the entanglement layer applies two-qubit gates, such as CNOTs, to interconnect pairs of qubits.

There are different strategies to perform entanglement in an ansatz, and selecting the right one is a critical aspect of PQC design, as it influences its ability to navigate the Hilbert space and converge during variational optimization procedures. For tasks involving anomaly detection or reconstruction of 1D time series data, such as this work, entanglement strategies as linear or circular are likely most appropriate for the training process. In the former, qubits are connected sequentially, while the latter couples qubits in a loop, allowing the last one to interact with the first one.

3.3.2. Architecture choices

The number of qubits initialized for each circuit depends mainly on the type of encoding used. In fact, as discussed in Appendix B.1, in the case of amplitude embedding, the number of qubits is determined by the length of the input data. This encoding has been implemented using the `RawFeatureVector` class offered by Qiskit, which acts as an initialization method for state vectors.

We must also ensure that the number of qubits in the latent space is sufficient to provide adequate representational capacity for this task. For this reason, we need to define the division between trash and latent qubits. The former ones are those discarded in the bottleneck layer, while the latter ones serve as the compressed representation of the input data. Through our experiments, we found that using a greater number of latent qubits than trash qubits leads to better results, as well as shorter training times. This improvement is likely due to the model's ability to preserve critical information from the input during the compression stage of the autoencoder. Having more trash qubits also leads to more complex and computationally expensive optimization processes due to the operation of tracing out a larger number of qubits from the quantum system in the bottleneck layer.

3.3.3. Metrics and losses

In this work, we have explored different types of loss functions and metrics for training and evaluating the quantum autoencoder.

The first metric we examined was the fidelity score, which measures the similarity between the trash space and the input state to identify the best latent space that generalizes the most important input features. For an input state $|\psi_{in}\rangle$ and a reconstructed state $|\psi_{out}\rangle$, the fidelity is defined as $F = |\{\langle\psi_{in}|\psi_{out}\rangle\}|^2$. When the fidelity score is equal to 1, the reconstructed state perfectly matches the input state, while a score of 0 means the reconstructed state is orthogonal to the input state. Typically, this value lies between 0 and 1.

Since we prefer to minimize a loss function C during the training process, we define the loss as:

$$C = -\frac{1}{log_{10}(1-F)}. \tag{4}$$

This definition requires F to be in the open interval (0,1), ensuring the logarithm is well-defined and avoiding division by zero.

After training, we used the MSE to compare the results, evaluating the performance of the autoencoder by calculating the difference between the original light curve and the reconstructed one.

A second way to assess the effectiveness of our model was through the cost function introduced in Bravo-Prieto (2021) and defined in earlier work by Cerezo et al. (2021). This function is constructed from the expectation values of local Pauli Z-operators measured on each trash qubit, and it is designed to ensure that these qubits decouple from the system by actively forcing their collapse into the $|0\rangle$ state. The formal definition of this cost function and its detailed derivation are provided in Appendix C.3.

We assessed the quality of the autoencoder's output curve reconstructions against the original ones using two standard metrics. Alongside the previously mentioned Mean Squared Error (MSE), we also utilized the Mean Absolute Error (MAE). This provides a direct measure of the average magnitude of errors by calculating the mean of the absolute differences between predicted values $\hat{y}_i$ and the actual values y_i for each data point i and it is defined as:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \hat{y}_i\right|. \tag{5}$$

To evaluate the error relative to the magnitude of the actual observations, we also computed the Mean Absolute Percentage Error (MAPE):

$$\text{MAPE} = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right|. \tag{6}$$

To calculate this, we first computed the standard MAE between the original sample and its autoencoder reconstruction, then we derived the arithmetic mean of the data points in the original light curve, finally the sample's MAE is expressed as a percentage of the sample's mean value:

$$\text{Light Curve Relative MAE \%} = \left(\frac{MAE_{lc}}{\mu_{lc}}\right) \times 100. \tag{7}$$

This percentage quantifies the average absolute reconstruction error relative to the sample's intensity level. Afterwards, the final metric for a dataset is obtained by averaging the individual MAE values across all samples belonging to that specific dataset. Normalizing the reconstruction error by each light curve's average intensity allows for a better comparison of the model's performance across signals of different magnitudes. Using this metric to assess the reconstruction quality of our models allowed us to compare the performance of the classical and quantum models on the test set. A lower MAPE value corresponds to a more accurate model.

4. Results

This section outlines the obtained results from both the classical autoencoder, which serves as our baseline, and the quantum approach, in order to investigate whether it is possible to achieve a quantum advantage or to determine if the quantum model can at least replicate the classical performance. The results shown in the following paragraphs refer only to the 32-bin case, unless stated otherwise, the reason for this will be explained in Section 4.2.

4.1. Classical results

As mentioned in Section 3.2, we tested multiple architectures with different numbers of trainable parameters: 6.5×10^5, 4×10^4, 1081, and 705. We also ran the model with various dataset sizes to see how these differences affect the final results.

As shown in Fig. 10, the models successfully converge during the training process, with the architecture containing the largest number of parameters achieving the best performance. It is also important to mention how the validation loss remains lower than the training loss during the process, and this is due to the presence of dropout layers that are only visible to the training dataset but not to the validation set, resulting in lower variability in the validation loss.

The minimum MSE achieved after the training process for both the training and test sets is presented in Table 2. This shows the different tests conducted by varying the size of the input dataset and the number of trainable parameters, while the results are divided between the training and test sets. We recall that, as described in Section 3.1, the test dataset is composed of 1×10^4 samples. The obtained trends indicate that, as might be expected, the model with the highest capacity (determined by its architecture, such as the number of layers and parameters) consistently achieved the lowest MSE values across both the training and test datasets, showing strong learning and generalization capabilities. On the other hand, the architecture with the fewest parameters exhibited a higher MSE, indicating its difficulties in reconstructing the input curves. These results also highlight the benefit of a larger dataset, as utilizing 5000 samples consistently produced lower MSE scores compared to using only 1000 samples.

In Table 3, we present the detected anomalies on the GRB dataset for the trained models that achieved better performance, as well as those incorrectly identified as outliers in the test set (i.e., false positives). It should be noted that models with lower performance were not considered for these and subsequent tables, as they could not adequately reconstruct the curves. The structure is the same as in Table 2.

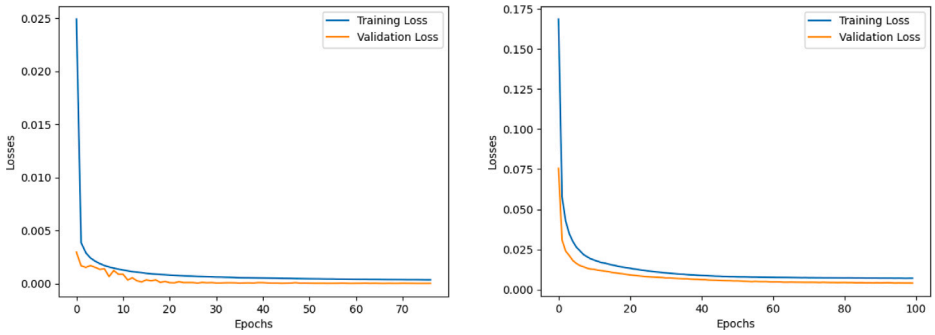
We found a very high detection rate for genuine GRBs across the models tested. This excellent detection accuracy demonstrates the effectiveness of classical autoencoders, which can easily detect anomalous data within the dataset owing to their optimal normal data reconstruction properties. Regarding the false positive rates, the table shows these are generally low for each configuration. It is also interesting to note that in this case, the model with fewer parameters shows a slight improvement for the 5000-sample case, presenting a lower false positive rate (0.15% vs 1.70%) and detecting a few more GRBs (99.52% vs 99.30%). This outcome highlights that although the largest model achieves a better overall reconstruction performance (lowest MSE), the difference, even if very minimal, in detection performance is likely due to the anomaly threshold. Since the largest model reconstructs normal background data with a very low error, it may become more sensitive to minor statistical noise during the inference task. The medium-sized model might offer a cleaner separation between background noise and the anomaly at the specific threshold used.

We further computed additional metrics, including precision, recall, and F1-score, to assess and compare the performance of different architectures in identifying anomalies and accurately classifying them as GRBs. In fact, this problem can be framed as a binary classification task, where the two classes correspond to background noise and GRBs. Precision measures the fraction of detections flagged as anomalous that correspond to true GRB events, a high value thus reflects a low rate of false positives among the identified candidates. Recall, also known as sensitivity, measures the fraction of all actual GRB events that the model successfully captures. A high recall value, therefore, signifies a low rate of false negatives, meaning few true events were missed. Finally, the F1-score is a single measure of overall classification performance since it is calculated as the harmonic mean of precision and recall. Those metrics are defined as follows:

$$\text{Precision} = \frac{\text{TP}}{\text{TP}+\text{FP}}, \tag{8}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP}+\text{FN}}, \tag{9}$$

$$\text{F1-score} = 2\times\frac{\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}}, \tag{10}$$



(a) The largest classical model, with 6.5×10^5 trainable parameters.

(b) The smallest models, with 705 parameters.

Fig. 10. Comparison of loss curves for different classical model sizes.

Table 2
Comparison of training and test errors for different datasets and classical models.

Dataset configuration	Trainable parameters	MSE - training set	MSE - test set
1000 Samples - 32 bins	6.5×10^5	1.65×10^{-4}	1.76×10^{-4}
	4×10^4	5.30×10^{-4}	5.65×10^{-4}
	1081	9.70×10^{-3}	9.78×10^{-3}
	705	1.32×10^{-2}	1.33×10^{-2}
5000 Samples - 32 bins	6.5×10^5	2.65×10^{-5}	2.97×10^{-5}
	4×10^4	1.51×10^{-4}	1.58×10^{-4}
	1081	5.42×10^{-3}	5.50×10^{-3}
	705	4.07×10^{-3}	4.11×10^{-3}

Table 3
Comparison of percentage of light curves flagged as anomalies in the GRB and test datasets for different classical model sizes and dataset configurations.

Dataset configuration	Trainable parameters	GRBs detected (Test set)	GRBs detected (GRB set)
1000 Samples - 32 Bins	6.5×10^5	0.31%	99.15%
	4×10^4	1.44%	98.50%
5000 Samples - 32 Bins	6.5×10^5	1.70%	99.30%
	4×10^4	0.15%	99.52%

where TP = True Positive, FP = False Positive and FN = False Negative. They range between 0 and 1, where 1 represents perfect performance and 0 indicates the worst possible outcome. In our work, we computed the MSE between each original light curve and its corresponding autoencoder reconstruction across the training, test and GRB datasets. An anomaly threshold was then established following the approach described in Section 3.2.2. Applying this single threshold to the MSE scores calculated for the combined test and GRB samples allowed us to assign a binary classification to each sample, and by comparing these assigned labels against the ground truth, we computed the metrics introduced above. Moreover, to assess performance across the entire spectrum of possible thresholds, we generated the Receiver Operating Characteristic (ROC) curve from the test and GRB samples' MSEs and calculated the total Area Under the Curve (AUC), obtaining the plot in Fig. 11. We can notice how the curve is very close to the top-left corner, where the False Positive Rate is 0 and the True Positive Rate is 1, indicating that the model exhibits a high capability to distinguish between normal data and anomalies. This is further supported by the AUC score.

In Table 4, we showed the aforementioned metrics. As already noted, we observe that the two larger models deliver excellent scores across all metrics. Their robust performance is evident for both dataset configurations.

Fig. 12 compares the original light curves with those reconstructed by the decoder, while the green plot represents the absolute difference between the real and reconstructed values for each bin. We can observe how the classical autoencoder performs exceptionally well in

Table 4
Comparison of precision, recall and F1-score for the classical architectures.

Dataset	Trainable parameters	Precision	Recall	F1-Score
1000 Samples - 32 Bins	6.5×10^5	0.997	0.992	0.994
	4×10^4	0.986	0.985	0.985
5000 Samples - 32 Bins	6.5×10^5	0.994	0.953	0.973
	4×10^4	0.998	0.995	0.997

reconstructing the curves, despite its simplicity as a vanilla model with a deterministic latent space. Notably, even the anomalies are reconstructed with high accuracy. On the other hand, the model with 705 parameters struggles to properly reproduce the curves, as shown in Fig. 13. Specifically, it struggles to reconstruct the test set curves, performing significantly worse than models with a greater number of parameters, which results in a loss of detail and poor reconstruction quality. Fig. 14 presents the histograms of the absolute differences between the original light curves and their reconstructed counterparts for the models with 6.5×10^5 and 1081 parameters. Each panel displays histograms corresponding to the training (green), testing (orange), and GRB (purple) datasets, effectively showing the model's reconstruction performance on normal versus anomalous light curves. We can easily notice how the error distributions for the normal background data are clearly separated from the distribution for the anomalous GRB data. The train and test distributions largely overlap and are concentrated at lower error values relative to the GRB distribution, which is shifted towards higher errors. However, these histograms also highlight the

Table 5
Comparison of training and test set reconstruction errors (MSE) for classical models trained with 100 samples.

Dataset	Trainable parameters	Training set (MSE)	Test set (MSE)
100 Samples - 32 Bins	6.5×10^5	1.51×10^{-3}	1.63×10^{-3}
	4×10^4	9.83×10^{-3}	1.04×10^{-2}
	1081	2.96×10^{-2}	2.94×10^{-2}
	705	3.82×10^{-2}	3.81×10^{-2}

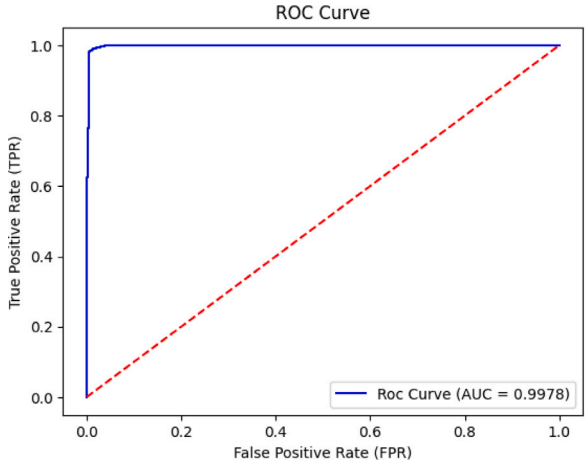


Fig. 11. ROC Curve showing the model's performance in detecting GRBs in the test set for the classical model. The red dashed line represents the performance of a random-guess classifier, which serves as a baseline to evaluate the model's performance.

Table 6
MAPE for different dataset configurations for the classical approach.

Dataset configuration	Trainable parameters	Mean absolute percentage error
100 Samples - 32 Bins	6.5×10^5	6.87%
	4×10^4	17.74%
	1081	29.79%
	705	34.77%
1000 Samples - 32 Bins	6.5×10^5	2.29%
	4×10^4	3.69%
	1081	16.64%
	705	19.72%
5000 Samples - 32 Bins	6.5×10^5	0.82%
	4×10^4	1.80%
	1081	11.76%
	705	9.90%

Table 7
Configurations for the quantum models.

Hyperparameter	Values
Latent Qubits & Trash Qubits	3+2, 4+1, 4+3, 5+2, 6+1
Samples	100, 1000, 5000
Anomaly Samples	20, 200, 1000, 10000
Number of time bins	32, 128
Layers	1, 3
Ansatz ID	Real Amplitudes, Efficient SU2 Two Local, PauliTwoDesign

impact of model capacity on reconstruction fidelity. The largest model separates the signal, reconstructing normal data with minimal error and creating a narrow distribution that is clearly separated from the GRB class. In contrast, as the number of model parameters decreases, we can observe a greater overlap between the two classes, confirming that performance deteriorates as the capacity of the classical architecture decreases.

Table 5 details the reconstruction performance of the classical autoencoders when trained under significantly data-constrained conditions, using only 100 samples. As expected, the reconstruction errors presented here are substantially higher across all model configurations compared to those achieved with the larger 5000-sample dataset. Even the highest-capacity model, which still performs best among the options in this limited data scenario, yields an MSE on the test set equal to 1.63×10^{-3}, while the models with fewer parameters exhibit much poorer reconstruction. Therefore, the classical autoencoder's ability to learn the underlying patterns of the background light curves and generalize is clearly limited in this case. These results serve an important purpose, they establish a baseline for classical performance under data scarcity, providing a critical reference point for the subsequent comparison with the quantum approach, particularly concerning its potential advantages in low-sample scenarios.

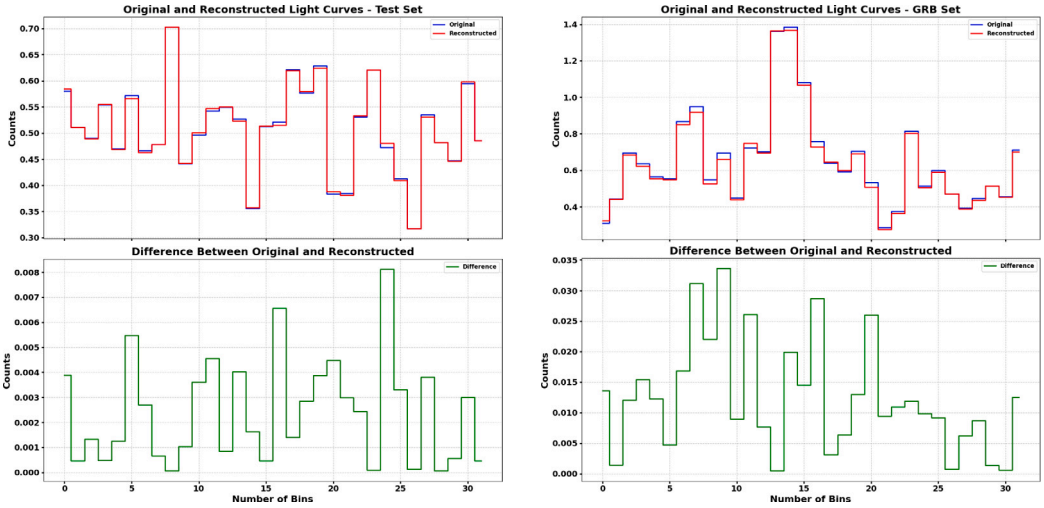
Finally, in Table 6 it is shown the difference between the MAPE computed on all the classical configurations after we trained the model, as described in Section 3.3.3. This metric is significantly higher for the GRB set than for the corresponding test set. However, the table only shows the MAPE obtained on the test set to highlight the model's ability to reconstruct normal data, in order to prove how the trained models manage to learn the distinction between background and GRB sources. We found that increasing the number of training samples from 1000 to 5000 reduces the MAPE for both datasets across all model sizes. For instance, the larger model sees its test error drop from 6.87% to 0.82%. Also, varying the number of trainable parameters confirms that models with higher capacity generally achieve lower relative errors compared to smaller models. In fact, the architectures with only 1081 and 705 parameters consistently struggle the most, exhibiting the highest MAPE, reaching values as high as 19.72% for test data. The challenges associated with limited data become particularly noticeable when studying the 100-sample results. In this scenario, the MAPE escalates dramatically, where even the highest-capacity model shows test set relative errors around 6%–7%, while the smaller models face severe difficulties, with errors becoming extremely large, even exceeding 30%. This observed decrease in performance, particularly the difficulties encountered by models with fewer parameters, sets a clear and challenging baseline for the classical approach.

4.2. Quantum results

This section presents the obtained results with the quantum model. Our initial study covered a wider range of dataset sizes and light curve dimensionalities, but we observed that 32-bin data produced the most promising results, as explained in Appendix D. For this reason, this section focuses on the analysis of only these light curves.

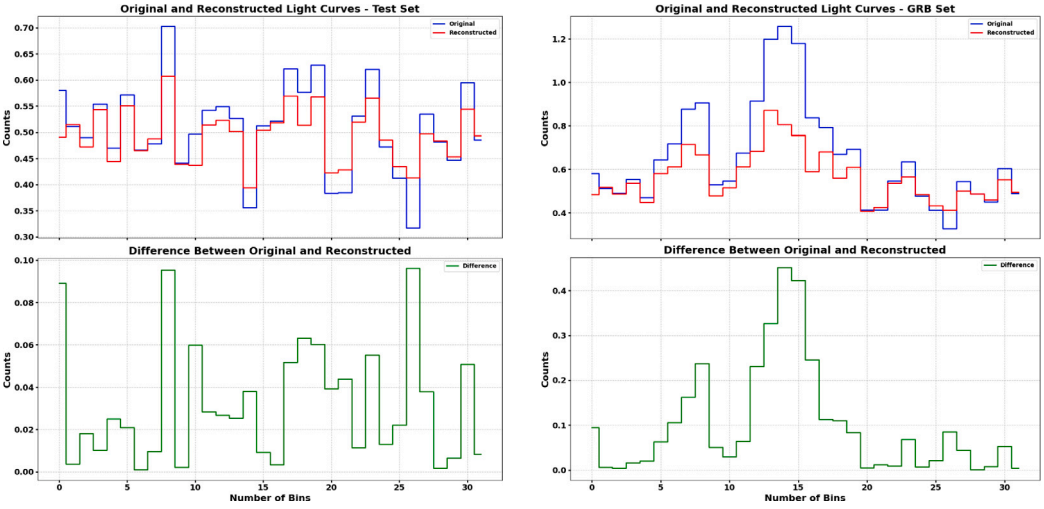
During this research, we analyzed different ansaetze, encoding techniques, and the number of qubits, as well as testing many parameter configurations (Table 7) to identify those that worked better. In this section, we will discuss the setup specifications that have yielded the best results.

Fig. 15 shows the training loss using a dataset of 5000 samples with 32 bins and employing the Real Amplitudes ansatz. It is worth noticing how the training and validation losses are almost the same. This is because we are using a deterministic statevector simulator, as discussed earlier in Section 3.3, which computes the full quantum state



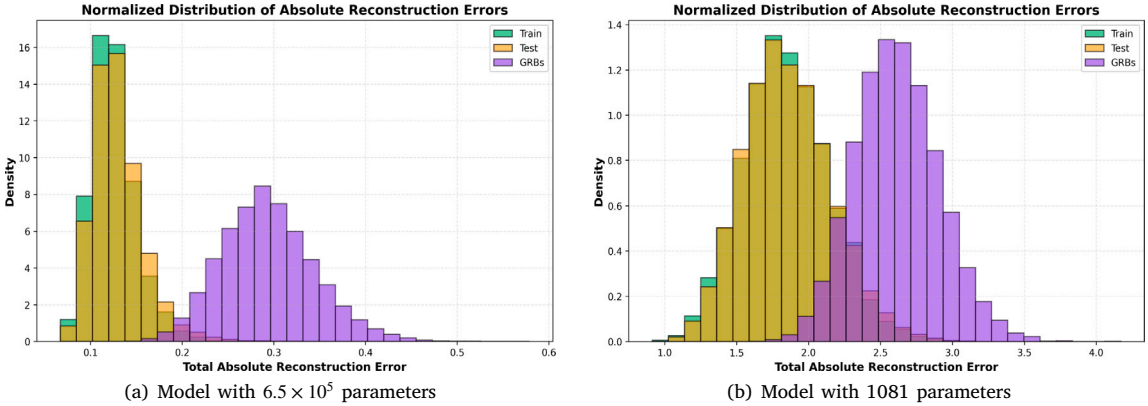
(a) Test Data Reconstruction (b) GRB Data Reconstruction

Fig. 12. Comparison of 32-bin reconstructed curves for test and GRB data using the 4×10^4 parameter model.



(a) Test Data Reconstruction (b) GRB Reconstruction

Fig. 13. Comparison of reconstructed curves for test and GRB data using the 705-parameter model.



(a) Model with 6.5×10^5 parameters (b) Model with 1081 parameters

Fig. 14. Comparison of histograms of differences for various model sizes using 5000 samples as the input dataset.

Table 8
Comparison between different dataset configurations for the quantum models.

Dataset configuration	Qubits	Layers	Ansatz	Reconstruction train	Reconstruction test
100 Samples - 32 Bins	4+1	1	Real Amplitudes	2.17×10^{-4}	2.24×10^{-4}
	4+1	1	Efficient SU2	2.41×10^{-4}	2.34×10^{-4}
	4+1	3	Real Amplitudes	6.49×10^{-4}	1.09×10^{-2}
1000 Samples - 32 Bins	4+1	1	Real Amplitudes	2.24×10^{-4}	2.21×10^{-4}
	4+1	1	Efficient SU2	2.28×10^{-4}	2.27×10^{-4}
	4+1	1	Real Amplitudes	7.88×10^{-4}	1.41×10^{-2}
5000 Samples - 32 Bins	4+1	1	Real Amplitudes	3.31×10^{-4}	3.32×10^{-4}
	4+1	1	Efficient SU2	2.32×10^{-4}	2.32×10^{-4}

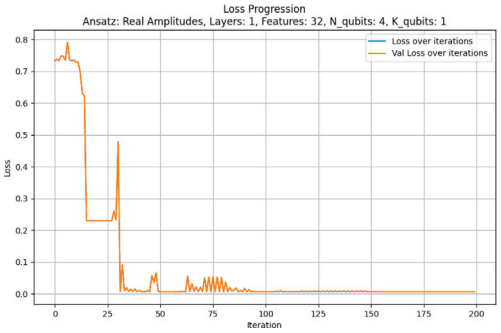


Fig. 15. Training loss using the Real Amplitude ansatz and 5000 samples. *Ansatz* denotes the specific model used, *Layers* the depth of the network, *Features* the number of bins of the light curves, *N_qubits* and *K_qubits* the number of latent and trash qubits of the autoencoder, respectively.

without any statistical noise. The validation loss never diverges from the training loss, even after 100 epochs, meaning that the model is not overfitting on the training data.

As for the latent and trash spaces, as we mentioned in Section 3.2, we found that a higher number of qubits for the latent space yields better final results. Therefore, we only kept the runs with this configuration. We also observed that increasing the number of trash qubits significantly increased training time. Moreover, due to computational complexity constraints, we limited further parameter exploration to only those configurations that proved to be successful. Additionally, we discarded any ansatz with too few trainable parameters, as they failed to achieve proper convergence during training.

In Table 8 we show the best obtained results with different dataset configurations and different ansaetze for the quantum approach. In this case, we found that increasing the number of layers does not always improve performance. This may be due to limited expressivity with only 5 qubits, which makes optimization more challenging in deeper circuits.

The plots in Fig. 16 compare the original and reconstructed signals for test and GRB data. The reconstruction of the test data closely follows the original curves. In contrast, the GRB data exhibit the most pronounced deviations, which serve as a key indicator of the model's success in identifying and differentiating these events as anomalies from the normal dataset. It is interesting to note that even if the model manages to recognize where the main peak is, we also obtain a secondary rise in the reconstructed signal shortly after. Since this is not an artifact of external noise, the reasons likely lie in how the model itself responds to these intense events. In fact, this phenomenon could be caused by difficulties in handling quantum phase information during the autoencoder's decoding process. While the classical output is derived from the amplitudes of the final quantum state, these depend on relative phases between the components of the state. Consequently, the decoder, in addition to having to reproduce the correct amplitudes for the individual bins, also faces the task of recreating the necessary phase coherence for signal reconstruction. Even subtle inaccuracies in this process can lead to interference at later moments, which in this case are represented as an *echo* in later bins.

Fig. 17 instead shows the distribution of absolute differences between the original curves and the reconstructed ones for each different dataset, similarly to the classical approach. The plot clearly reveals that the GRB dataset occupies a region of higher reconstruction errors, separated from the largely overlapping distributions of the train and test datasets, concentrated at lower error values.

Regarding the case with 100 samples, compared with the results obtained with 5000 data points, the reconstruction errors are very similar, again indicating the ability of the quantum model to generalize effectively even from significantly limited training datasets. We also obtain similar results with 1000 samples, where again we can notice how the differences are minimal across all metrics.

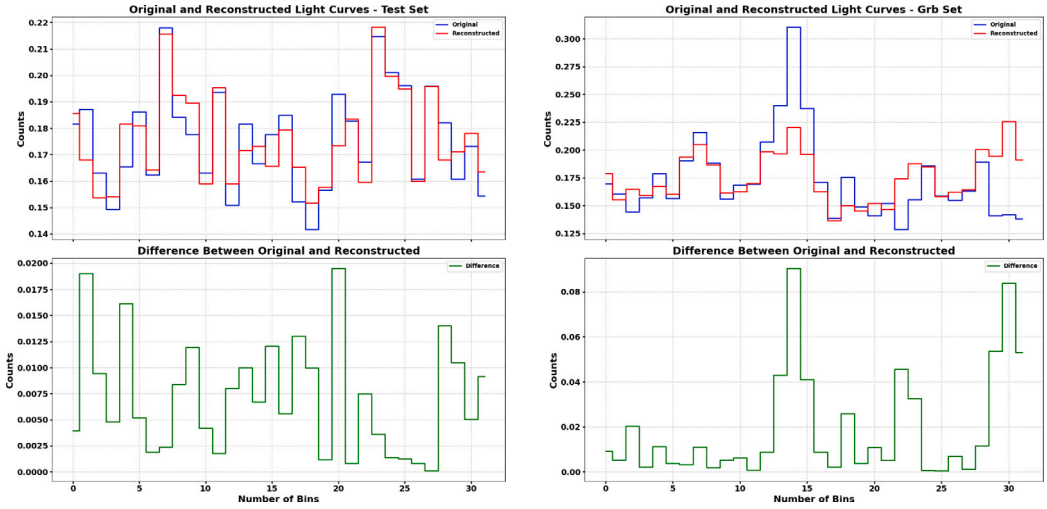
Table 9 shows the percentage of GRBs detected in the test set and in the GRB set. This quantum approach performs well across all dataset sample sizes. The threshold to find the anomalies has been computed following the approach described in Section 3.2.2.

The ROC curve in Fig. 18 shows good classification performance, suggesting that the quantum model is effective at distinguishing between normal and GRB instances across various classification thresholds.

Table 10 details the quantum models' classification performance, revealing consistently high F1-scores, especially for the Efficient SU2 ansatz. Also in this case, it is important to note how the quantum approach shows minimal performance degradation even with significantly reduced training data.

Table 11 shows the number of trainable parameters for each possible configuration. It is easy to notice how this number increases significantly as the depth grows, and consequently, also the training time. The main difference between quantum models lies in architectural choices, such as rotation strategies or the specific entanglement mechanisms. Nevertheless, we can observe that all these models have a significantly lower number of parameters compared to the classical ones.

Finally, Table 12 shows the difference between the MAPEs computed on different quantum configurations after we trained the model, as described in Section 3.3.3, similarly as we did for the classical model. What is particularly noteworthy here is the quantum models' consistent performance on the test set, irrespective of the number of training samples used. For instance, the MAPE on the test set is approximately 7% whether the model was trained on only 100 samples or on a larger dataset. This phenomenon suggests that the quantum approach is capable of learning generalizable features for reconstructing normal background-like signals even when provided with very limited training data, unlike classical models that show a more pronounced degradation in performance as data becomes scarce. This characteristic points towards a potential advantage of quantum models in scenarios where obtaining large, labeled datasets is challenging, as discussed in the following Section 4.3.



(a) Test Data Reconstruction (b) GRB Reconstruction

Fig. 16. Original and reconstructed light curves composed of 32 features, for the quantum model.

Table 9
Comparison of percentage of light curves flagged as anomalies in the GRB and test datasets for different quantum model sizes and dataset configurations.

Number of Samples	Ansatz	GRBs detected (Test set)	GRBs detected (GRB set)
100	Real Amplitudes	9.00%	91.36%
	Efficient SU2	6.00%	93.44%
1000	Real Amplitudes	10.00%	86.71%
	Efficient SU2	8.00%	92.92%
5000	Real Amplitudes	9.00%	91.13%
	Efficient SU2	7.00%	92.11%

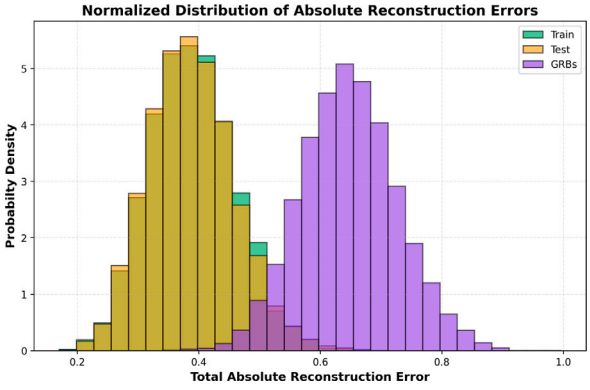


Fig. 17. Histograms of absolute differences for the 32-bin case and 5000 samples as input dataset, for the quantum autoencoder.

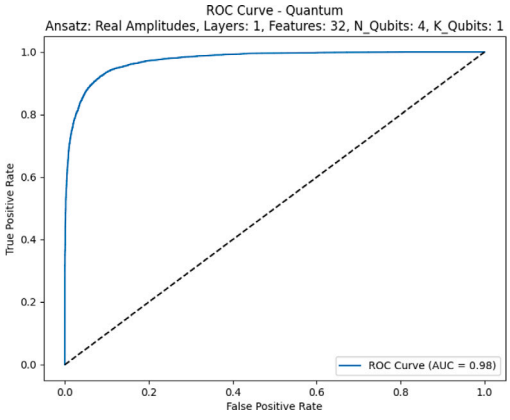


Fig. 18. ROC Curve showing the quantum model's performance in detecting GRBs in the test set. The specific model used is denoted by *Ansatz*, the network depth by *Layers*, and the number of light curve bins by *Features*. For the autoencoder, the number of latent and trash qubits are represented by *N_qubits* and *K_qubits*, respectively.

Table 10
Performance metrics for the quantum model with 100, 1000, and 5000 samples in input.

Dataset configuration	Ansatz	Precision	Recall	F1-Score
100 Samples - 32 Bins	Real Amplitudes	0.910	0.914	0.912
	Efficient SU2	0.940	0.934	0.937
1000 Samples - 32 Bins	Real Amplitudes	0.897	0.867	0.882
	Efficient SU2	0.921	0.929	0.925
5000 Samples - 32 Bins	Real Amplitudes	0.910	0.911	0.911
	Efficient SU2	0.930	0.931	0.931

Table 11
Number of trainable parameters for each model. Q indicates the number of qubits of the circuit, while L means the number of layers.

Model	5q, 1L	7q, 1L	5q, 3L	7q, 3L
Efficient SU2	20	28	40	56
Real Amplitudes	10	14	20	28

Table 12
MAPE on the test set obtained by the quantum approach.

Dataset configuration	Ansatz	Mean absolute error percentage
100 Samples - 32 Bins	Efficient SU2	6.98%
	Real Amplitudes	6.75%
1000 Samples - 32 Bins	Efficient SU2	6.84%
	Real Amplitudes	6.96%
5000 Samples - 32 Bins	Efficient SU2	6.90%
	Real Amplitudes	7.05%

4.3. Comparison between classical and quantum approaches

In this section, we compare the obtained results by the classical model with those of its quantum counterpart. In particular, we focus on the performance achieved by both approaches in scenarios with limited resources, such as the number of trainable parameters, the dataset size, and the dimensionality of the input data.

We have seen how classical models with larger datasets generally achieve superior results, as can be seen from the comparison between Tables 2 and 8. However, the quantum approach has demonstrated a clear advantage in resource-limited scenarios. When compared to a classical autoencoder with fewer trainable parameters, the quantum model achieves better results and outperforms the classical version by more than one order of magnitude. As we have pointed out, the quantum model can achieve an optimal MSE with only 10 trainable parameters, while the classical model struggles with 705, proving the quantum strength in these scenarios.

This efficiency also extends to settings with limited data, such as the 100-sample case. In fact, the results in Table 5 for the classical approach and Table 8 for the quantum one show again that the latter outperforms the former by one order of magnitude on the test set, demonstrating that the quantum approach is more robust and efficient at learning from small datasets.

It is interesting to note that the quantum model offers consistent performance, especially with limited datasets. This is related to the advantages offered by quantum mechanics, discussed in Appendix B. For example, the ability to encode information in superpositions allows for excellent efficiency in extracting data patterns, while quantum entanglement enables the creation of non-classical relationships between features. These properties permit the model to explore a vast computational space with very few parameters, suggesting the potential that quantum approaches have for domains where data collection is limited or costly. However, this quantum advantage is still limited by the current hardware, even on simulators. We have seen, in fact, how a classical model operating at full capacity remains superior.

The practical impact of these quantum advantage and current limitations on the model's performance is clearly illustrated by the reconstruction plots in our work. For instance, Figs. 12 and 16 show a direct comparison of reconstruction quality. The former achieves a better data reconstruction, matching the original signal more accurately. The latter, in contrast, shows greater differences between the original and reconstructed curves. These discrepancies are emphasized in the difference plots, which give an example of high error peaks for the quantum model, especially for the test set. On the other hand, if we consider the plots in Fig. 13, we notice how a smaller classical model struggles significantly to reconstruct the curves, yielding poorer results than the quantum autoencoder. This is an important observation since it confirms that, when operating with fewer parameters, a quantum model can indeed outperform a classical counterpart in reconstruction tasks. These findings are also summed up in Table 13, which illustrates the mean absolute error percentage (MAPE) on the test set for the best classical and quantum models across different dataset configurations. This table confirms the quantum model's advantages in two resource-constrained areas: data scarcity and parameter efficiency. In the 100-sample scenario, the quantum approach, using only 10 parameters, achieves a MAPE of 6.75%, performing similarly with the best classical model that requires 6.5×10^5 parameters. This efficiency is further highlighted when compared to the smaller 705-parameter classical model, which struggles significantly in the same low-data environment, obtaining a MAPE of 34.77%. This advantage is maintained across all dataset sizes, as seen even in the 5000-sample case where the quantum autoencoder achieves a 6.73% MAPE compared to the classical model's 9.90%. However, the table also confirms the superiority of the classical architecture when it can operate at full capacity. As the number of training samples increases from 100 to 5000, the high-capacity classical model's performance scales impressively, with its MAPE dropping to 0.82%.

It is interesting to note that, as also mentioned in Section 4.2, the MAPE of the quantum models does not significantly decrease as the number of input samples increases. This suggests that, due to the limited number of trainable parameters and computational constraints, the models quickly reach their maximum expressive potential. As introduced in Section 4.2, we increased the depth of the circuit, but this did not lead to performance improvements. Instead, it contributed to optimization problems, such as barren plateaus, pointing to a computational limitation in the quantum architectures. However, the stability of the results across varying dataset sizes confirms the efficiency of the quantum models in generalizing information patterns, achieving near-optimal performance even with a limited number of training samples.

It is worth mentioning how the simulated quantum models proved to be significantly slower than their classical counterparts. Regarding the 5000-sample case, the larger classical algorithm achieved convergence in only 47 s. On the other hand, the quantum architecture utilizing the Real Amplitudes ansaetze required 27 min and 52 s to complete training. The memory increment during the training process for the classical model is 513.82 MB, while for the quantum architecture is only 28 MB. This is given by the lower number of trainable parameters employed in the quantum models. Instead, for the 100-sample case, the quantum model only takes 1 min and 27 s, while its classical counterpart requires a training time similar to the 5000-sample case result, indicating that the classical model's primary bottleneck in this case is the number of trainable parameters, given that even the larger dataset size implemented in this work is modest compared to typical deep learning standards. In summary, the classical models converge faster, particularly as the input data size increases. For the quantum models, the main bottleneck remains the data encoding process. This is not surprising and considered to be expected, as also further discussed in Bowles et al. (2024).

5. Conclusion & final discussion

In this work, we outlined a comparative study between classical and quantum autoencoders for the task of detecting GRBs within simulated COSI BGO shield data. Our goal was to compare a quantum approach with a classical one, in order to verify whether the former could offer advantages over classical machine learning techniques already widely used with excellent results in the field of astrophysical anomaly detection.

We confirmed that classical autoencoders are very effective in detecting GRB anomalies, achieving an optimal light curve reconstruction and high accuracy. However, their performance decreases when the amount of input data or the number of trainable parameters is limited and reduced by several orders of magnitude. This helped us establish a baseline for classical models and allowed us to compare them with their quantum counterparts, implemented with parameterized quantum circuits composed of a limited number of qubits and layers due to computational limitations.

Under the same resource-constrained scenarios, the quantum counterpart required fewer trainable parameters to outperform the classical

Table 13
Comparative analysis of MAPE on the test set: classical vs. quantum autoencoders.

Dataset configuration	Approach	Model details	Mean absolute percentage error
100 Samples - 32 Bins	Classical	6.5×10^5 Parameters	6.87%
	Classical	705 Parameters	34.77%
	Quantum	Real Amplitudes (10 parameters)	6.75%
1000 Samples - 32 Bins	Classical	6.5×10^5 Parameters	2.29%
	Classical	705 Parameters	19.72%
	Quantum	EfficientSU2 (20 parameters)	6.84%
5000 Samples - 32 Bins	Classical	6.5×10^5 Parameters	0.82%
	Classical	705 Parameters	9.90%
	Quantum	EfficientSU2 (20 parameters)	6.73%

approach. Similarly, we demonstrated how the quantum autoencoder performs better than the classical model with only 100 input data points. We also found that our quantum models produced more stable training and better results using lower-dimensional input data, such as light curves composed of 32 bins. This is related to the reduced complexity and depth of the quantum circuits needed to process this data, which are therefore less susceptible to potential training issues.

However, when operating at full capacity, the classical autoencoder still achieves higher reconstruction accuracy than quantum circuits based on metrics such as mean squared error (MSE) and mean absolute percentage error (MAPE). In fact, the quantum model only showed an advantage in contexts with limited resources, suggesting a theoretical efficiency in the way quantum circuits can process information.

This finding suggests a promising application of quantum machine learning, particularly in environments where data is exceptionally scarce or where lightweight models are required. However, these results are currently limited by the computational cost in developing quantum models, even when simulated, which proved to be a major obstacle, thus limiting the scale of the models in this study.

Therefore, the path toward using quantum-based architectures in astrophysics remains challenging. If we want to test these models on real quantum processors, and not only simulators, we must take into account the presence of hardware noise, which will require the implementation of quantum error mitigation and correction techniques, making the implementation of these architectures even more difficult. However, in light of recent developments in quantum hardware, this direction remains rich with possibilities. With improvements in coherence, QEC techniques, and the capacity of quantum computers, the exploration of more complex quantum architectures will become increasingly feasible. A possible additional exploratory phase of this work could involve the use of more sophisticated quantum data encoding techniques, in addition to the use of more complex ansaetze. Hybrid quantum–classical models combining the strengths of both categories could also be implemented. The first intended step towards using a real quantum computer is to simulate noise inside the quantum network to mimic the imperfections of current NISQ devices. This can be done directly on a classical computer using Qiskit Aer. In fact, to obtain a realistic baseline of our quantum models' performance, we first need to understand how gate infidelity and decoherence could affect the autoencoder's reconstruction capability, helping to establish whether the implemented architectures can maintain the obtained accuracies in a noisy environment as well. A more comprehensive exploration of quantum variational architectures is required to address this path, which will be the primary goal of upcoming research.

To perform inference on our trained models, we employed data augmentation techniques based on the original GRB template. It is worth recalling that this research is intended as an exploratory study, for this reason, the current dataset was sufficient to establish the models' baseline performance. Future work will expand the dataset with more real-time observations to further validate the efficacy of the proposed methodology.

Future investigations might also extend this research to the classification and detection of additional source models. We have outlined a GRB detection technique based on the analysis of background input sources. However, in addition to these events, there are various possible anomalies, such as soft gamma repeaters, solar flares, and terrestrial gamma-ray flashes. This is because GRB monitors trigger on a variety of signals, making their distinction an important part of GRB detection.

In conclusion, classical autoencoders managed to achieve excellent input curve reconstruction and very high signal recognition. For this reason, they remain the best choice for this anomaly detection task. However, if we reduce the size of the input data and the capacity of the trained models, the performance of classical architectures deteriorates significantly. It is precisely in this context that quantum autoencoders seem to offer advantages, especially when input data is scarce or it is important to implement computationally lightweight models. Although they are not yet able to completely replace their classical counterparts, their efficiency in extracting information from limited resources highlights their great potential. As quantum hardware advances, these models will find new ways to solve complex data analysis problems in astrophysics and related fields.

CRediT authorship contribution statement

Alessandro Rizzo: Writing – review & editing, Writing – original draft, Visualization, Software, Methodology, Investigation, Formal analysis, Data curation. **Nicolò Parmiggiani:** Writing – review & editing, Supervision, Project administration, Conceptualization. **Andrea Bulgarelli:** Writing – review & editing, Supervision, Resources, Project administration, Conceptualization. **Antonio Macaluso:** Writing – review & editing, Supervision. **Carlo Burigana:** Writing – review & editing. **Eric Burns:** Writing – review & editing. **Luca Cappelli:** Writing – review & editing. **Vincenzo Fabrizio Cardone:** Writing – review & editing. **Farida Farsian:** Writing – review & editing. **Savitri Gallego:** Writing – review & editing. **Giuseppe Murante:** Writing – review & editing. **Giuseppe Sarracino:** Writing – review & editing. **Roberto Scaramella:** Writing – review & editing. **Francesco Schillirò:** Writing – review & editing. **Vincenzo Testa:** Writing – review & editing. **Tiziana Trombetti:** Writing – review & editing. **Dieter H. Hartmann:** Writing – review & editing. **Carolyn A. Kierans:** Writing – review & editing. **Eliza Neights:** Writing – review & editing. **John A. Tomsick:** Writing – review & editing. **Andreas Zoglauer:** Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is supported by the ICSC - Centro Nazionale di Ricerca in HPC, Big Data e Quantum Computing, CN00000013, spoke 10 "quantum computing", CUP C53C22000350006. The Compton Spectrometer and Imager is a NASA Explorer project led by the University of California, Berkeley with funding from NASA, United States under contract

80GSFC21C0059. Resources supporting this work were provided by the NASA High-End Computing (HEC) Program through the NASA Advanced Supercomputing (NAS) Division at Ames Research Center. SG acknowledges support by DLR, United States grant 50OO2218. Resources supporting this work were provided by National High Performance Computing (NHR) South-West at Johannes Gutenberg University Mainz. This work has been partially funded by ASI under contract 2024-11-HH.0. E.N. acknowledges NASA, United States funding through Cooperative Agreement 80NSSC22K0982.

Appendix A. Classical autoencoder

This appendix provides an overview of the classical autoencoder, which served as the baseline model for comparison with the quantum approach in our study.

Autoencoders use unsupervised machine learning to compress input data into a lower-dimensional latent space, defined during a training process that minimizes the reconstruction error between the original input and its output. Every autoencoder is built around three fundamental elements, represented in Fig. A.19:

- The *encoder* creates a compressed representation of the input data by reducing its dimensionality. This is achieved through a series of hidden layers where each subsequent layer has fewer nodes than the one before it.
- The *bottleneck* holds the most compressed form of the data. This layer serves as the output of the encoder and the input for the decoder. This stage performs the main objective of an autoencoder, which is to isolate the most important features of the input data, which will be essential for the reconstruction step.
- The *decoder* takes the compressed data from the bottleneck layer and reverses the process, using a series of hidden layers with an increasing number of nodes to expand the data back to its original size. This final output is then compared against the initial input, or ground truth, and the resulting difference is known as the reconstruction error.

More details on this neural network can be found in Hinton and Salakhutdinov (2006) or Goodfellow et al. (2016).

Appendix B. Quantum computing basics

This appendix details the theoretical foundations of our quantum model, from the basic principles of quantum computation to their practical application in machine learning. We begin with an overview of fundamental concepts such as the qubit, superposition, and entanglement. We then build upon this foundation to explain the hybrid quantum–classical approach and the variational algorithms that are central to this work. For more detailed information about quantum computing and quantum artificial intelligence in general, the reader can refer to Klusch et al. (2024).

Quantum computing leverages the principles of quantum mechanics to process information and perform computations. Among the various models, gate-based quantum computing is considered the most promising, using quantum gates to process quantum information inputs. This model allows for the implementation of complex quantum circuits, which act as quantum analogues of Boolean circuits in classical computing.

A single bit, represented by the integers 0 or 1, is the fundamental building block of classical information processing. Bits can exist in only one of two states, so n bits can be described as 2^n unique values, with the bit register being in only one of these possible states. A quantum bit, or qubit, is the quantum analog of a classical bit and can be represented by the two basis states $|0\rangle = \binom{1}{0}$ and $|1\rangle = \binom{0}{1}$. According to the laws of quantum mechanics, qubits can be implemented using a variety of physical systems, including neutral atoms, ion traps, superconducting circuits, and the spin states of subatomic particles.

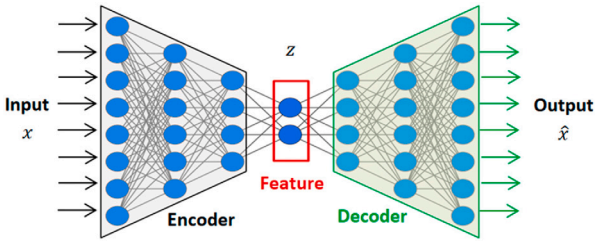


Fig. A.19. Representation of a classical autoencoder structure.
Source: Taken from (Alaghbari et al., 2023).

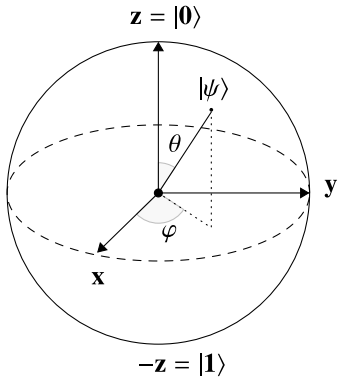


Fig. B.20. Representation of a quantum state $|\psi\rangle$ on the Bloch sphere. The state is described by the angles θ and φ, where θ defines the polar angle from the z-axis and φ defines the azimuthal angle in the xy-plane. The Bloch sphere offers a geometric representation of the pure states of a qubit, where $|0\rangle$ and $|1\rangle$ correspond to the poles of the sphere.
Source: Figure adapted from Klusch et al. (2024).

The ability of qubits to exist in a superposition of both states at once, $|0\rangle$ or $|1\rangle$, is one of their most important characteristics that sets them apart from their classical equivalent. An arbitrary single-qubit state can be expressed as:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle = \begin{pmatrix}\alpha\\ \beta\end{pmatrix}, \tag{B.1}$$

where the coefficients α and β are complex numbers that satisfy the normalization condition $|\alpha|^2 + |\beta|^2 = 1$. Fig. B.20 shows the visual representation of a qubit. In a multi-qubit system, an n-qubit register creates a computational space with 2^n dimensions, known as Hilbert space, where any pure quantum state of the system is defined as:

$$|\psi\rangle = c_0|0\cdots 0\rangle + c_1|0\cdots 1\rangle + \cdots + c_{2^n-1}|1\cdots 1\rangle, \tag{B.2}$$

with $c_i \in \mathbb{C}$ and $\sum_{i=0}^{2^n-1}|c_i|^2 = 1$. The multi-qubit basis states, e.g., $|0\cdots 1\rangle = |0\rangle\otimes|0\rangle\otimes\cdots\otimes|0\rangle\otimes|1\rangle$, are tensor products of the individual qubits. The state $|\psi\rangle \rightarrow [c_0, c_1, \ldots, c_{N-1}]^t$ consists of $N = 2^n$ complex amplitudes, that must satisfy the condition that the sum of their squared absolute values equals one. Thus, based on the superposition principle, an n-qubit system is able to encode information that scales as 2^n, while classical systems are limited to n. If the component qubits of a multi-qubit state cannot be independently described as a tensor product, we say that it is entangled. This very important quantum mechanism enables non-local correlations, where the operation on one qubit alters the state of others instantaneously and independently of their separate locations. An example of this phenomenon is the entangled 2-qubit Bell states.

Computing requires the manipulation of quantum states. This is accomplished by operators that outline the evolution of a closed quantum system using unitary transformations. These must be unitary and

reversible in order to maintain the normalization of the original quantum state. This means that for a quantum operator U must hold that $U^\dagger U = I$. It follows that any operation on qubits can be described as a matrix operator. This is one of the main differences with classical computing, where operations can also be non-unitary or reversible.

Another significant distinction between classical and quantum computing is the ability to access information after it has been processed. In classical computing, it is easy to find the state of each bit without any disturbance since we can directly observe it at any time in a state of 0 or 1. However, we have seen how a qubit can exist simultaneously in a combination of 0 and 1, as it is often in a superposition of different states. To obtain information from any quantum system, we need a measurement of an observable physical quantity, which, from now on, we assume to be the binary number associated with the computational basis states. This measurement process collapses the quantum system from its superposition state to one of the possible definite states. This is a probabilistic process, in fact, one cannot predict the outcome of this measurement since it follows the amplitudes of the quantum state. It is also important to note that after performing the measurement process, it is no longer possible to retrieve the original superposition state of the system. This is an important distinction from the classical approach, where observation is non-intrusive and reversible, and does not alter the state of the system in any way.

Given the quantum state in Eq. (B.1), the measurement will result in one of the basis states $|0\rangle$ or $|1\rangle$ with classical output 0 or 1, respectively. The probability of observing $|0\rangle$ is $|\alpha|^2$, while the probability of observing $|1\rangle$ is $|\beta|^2$. The total probability must sum to 1 ($|\alpha|^2+|\beta|^2 = 1$). After measurement, the qubit will be found in the state corresponding to the measurement outcome. For the state described in Eq. (B.2), the measurement outcome will be one of the 2^n possible basis states, as introduced above. The probability of observing a particular multi-qubit state $|\psi\rangle = |b_1 b_2 \cdots b_n\rangle$ is given by $|c_{\text{index}}|^2$, where *index* represents the binary number corresponding to the combination $b_1 b_2 \cdots b_n$, and the total probability must sum up to 1, ($\sum_{i=0}^{2^n-1} |c_i|^2 = 1$).

Following the measurement, the quantum system collapses into the observed state, determined by the result of the measurement itself. This is the only non-unitary, or irreversible, operation in an ideal quantum computer. For this reason, unlike classical computing, it is impossible to generate an identical copy of an arbitrary quantum state, according to the no-cloning theorem of quantum computing (Wootters and Zurek, 1982).

It should be noted that squaring the amplitude relative to probabilities leads to negative and arbitrary amplitudes with complex values, which are responsible for destructive interference, an advantage of quantum algorithms used to omit unwanted solutions. This phenomenon is a fundamental difference between quantum and classical stochastic algorithms. Another interesting detail concerns the norm that the quantum state vector must respect, which is linked to the physical principle of conservation of total probability and is the reason why quantum operations must be unitary.

The fundamental concepts of quantum computing mentioned above are direct consequences of the principles of quantum mechanics and form a basis for exploring how these principles can be applied in practice through *quantum circuits*, which function as the quantum counterpart of classical algorithms.

The execution of a quantum circuit starts with a defined initial state, which is followed by the implementation of quantum gates that perform computation and transformations on these states. A measurement ends the circuit, extracting classical information by collapsing the qubits into definite classical states. More information about this can be found in Appendix B.1. Quantum algorithms can be visualized via quantum circuit diagrams. Horizontal lines represent individual qubits, while the sequence of operations, or gates, is indicated by their position along these lines. An example of a simple quantum circuit is depicted in Fig. B.21.

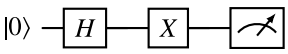


Fig. B.21. A simple quantum circuit demonstrating the combination of basic quantum gates. The circuit starts with a qubit in the $|0\rangle$ state. The Hadamard gate (H) is applied first, creating a superposition of $|0\rangle$ and $|1\rangle$. Following this, a Pauli-X gate (X) is applied, flipping the qubit's state. The circuit concludes with a measurement, represented by the meter symbol, which collapses the qubit's state into either $|0\rangle$ or $|1\rangle$, producing a classical output.

Quantum computing can mimic classical computing via a universal set of quantum gates, and the opposite is also true (Eisert and Wolf, 2006). However, for problems that can be solved in polynomial time by a probabilistic quantum Turing machine with bounded error (error probability less than or equal to 1/4), quantum computing solutions belong to the complexity class of bounded-error quantum polynomial-time. In addition to the P class, this also intersects with NP and PSPACE. This motivates exploration that goes beyond algorithms that already have a proven significant speed-up over their classical counterparts, such as quantum prime factorization (Shor, 1997) and quantum search (Grover, 1996; Pokharel and Lidar, 2024).

B.1. Quantum machine learning with hybrid computation

Quantum algorithms based on the gate model have demonstrated theoretical speedups compared to their classical counterparts in certain types of problems, especially when considering worst-case complexity (Deutsch and Jozsa, 1992; Shor, 1997; Arute et al., 2019). However, the obtained results are based on theoretical cases, such as the availability of accessing large-scale quantum computers with extremely low logical error rates. The construction of these devices necessitates the implementation of quantum error correction techniques to protect the input information from noise, decoherence, and hardware imperfections. The implementation of such processes remains a major engineering challenge.

In machine learning, VQAs are used to train parameterized quantum models to approximate functions, make predictions, or encode and compress data. These models are implemented through PQCs, or variational quantum circuits (VQCs), whose trainable gates are composed of parameters that are optimized using classical algorithms. The structure of a quantum machine learning model is divided into two main parts: an encoding circuit U_S and a variational circuit $U(\theta)$. The former is responsible for mapping classical input data x to a quantum state. This step is referred to as feature encoding or quantum embedding, and is fundamental to defining the expressive power of the model. There are several encoding strategies: basis encoding, amplitude encoding, or more complex feature maps. The choice of encoding affects both the model's performance and the depth of the circuit required to implement it (Weigold et al., 2021). Amplitude encoding is the technique adopted in this research, given its computational potential. This method encodes a classical vector x of length N into the amplitudes of a normalized n-qubit quantum state, where $n = \lceil \log_2(N) \rceil$:

$$|x\rangle = \sum_{i=0}^{N-1} x_i |i\rangle, \tag{B.3}$$

where $|i\rangle$ denotes the computational basis states. Since the classical data becomes the amplitude vector of the quantum state, it must satisfy the normalization condition $\|x\|^2 = 1$. This encoding allows for an exponential compression of information, representing high-dimensional data using a logarithmic number of qubits, and this is particularly advantageous when working with large-scale datasets.

Once the input encoding is complete, the variational unitary $U(\theta)$ applies a series of parameterized quantum gates, initialized with random parameters θ. To calculate an output $f(x;\theta)$, we must first measure the quantum result and process it classically. This output is then evaluated using a cost function that is used to train the model parameters.

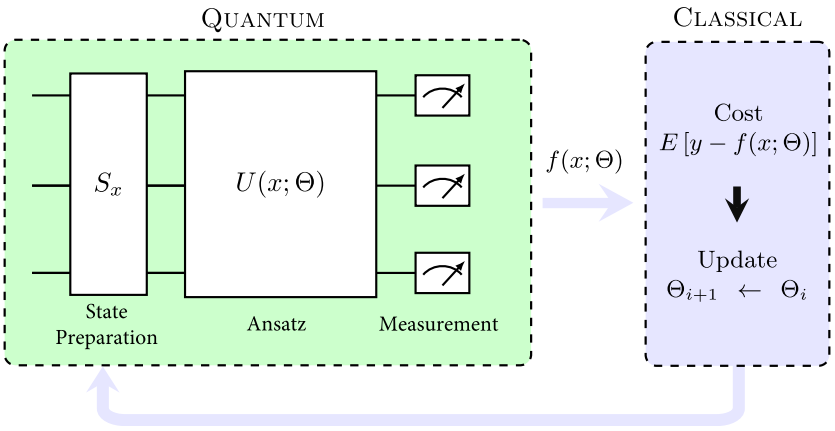


Fig. B.22. A Hybrid Quantum–Classical Optimization Workflow (adapted from Macaluso et al. (2022)). The diagram illustrates a hybrid quantum–classical optimization process. The quantum part is composed of three main steps. **State Preparation**: An initial quantum state is prepared using the unitary operator U_S. **Computation**: The prepared state is processed by a parameterized quantum circuit $U(\Theta)$ to search for the optimal solution x based on the parameters Θ. **Measurement**: The quantum state is measured to obtain the expectation value $\langle M \rangle$. The classical part is also composed of three stages. **Post-Processing**: The measured expectation value $\langle M \rangle$ is processed to extract the classical variable x. **Evaluation**: The function $f(x)$ is evaluated based on the classical variable x. **Update**: The parameters Θ are updated using classical optimization algorithms to improve the search in the next iteration. The process iterates, with the updated parameters Θ_{i+1} being fed back into the quantum circuit, forming a closed-loop optimization cycle between the quantum and classical computations.

This is a hybrid cycle that continues iteratively until we reach convergence or a desired performance threshold. The trainable parameters within the PQC can be optimized for a wide range of machine learning tasks, while the classical optimization level ensures compatibility with current hardware limitations, making this architecture adaptable to every kind of problem. This process is shown in Fig. B.22

Appendix C. Quantum autoencoder

This appendix provides a comprehensive analysis of our quantum autoencoder model. We begin by detailing the model's fundamental architectures and its operational principles. We then describe the SWAP test, and, finally, we introduce the Feature Vector Encoding implementation, a method that makes the quantum circuit informed of classical features from the input data.

This quantum network can be represented graphically as an interconnected group of nodes, where each one of these nodes corresponds to a qubit in the system. We achieve the structure of an autoencoder when some of the input nodes are discarded after the initial encoding step, and this is done by tracing over the qubits that correspond to these nodes. Then, additional qubits, initialized in a given reference state, are prepared and utilized for the final decoding stage, the result of which is then compared with the original state. The learning task of a quantum autoencoder is therefore to identify a unitary transformation that preserves the input's quantum information as it passes through the intermediate latent space, when the input nodes are discarded. For this reason, it is important to quantify the deviation from the initial input state $|\psi_i\rangle$ to the output state ρ_i^{out}. In the original work (Romero et al., 2017), it is used the expected fidelity $F(|\psi_i\rangle, \rho_i^{out}) = \langle\psi_i|\rho_{out}|\psi_i\rangle$ (Wilde, 2012). Therefore, a successful autoencoding is described as one in which $F(|\psi_i\rangle, \rho_i^{out}) \approx 1$ for all the input states. However, we also explored different metrics, as illustrated in Section 3.3.3.

The circuit's architecture consists of different stages. First, at the input layer, we introduce the initial quantum state $|\psi\rangle$, which we want to compress. This is followed by the encoder layer, realized as a parameterized circuit that acts on the input state. The encoder compresses the quantum state, reducing its dimensionality to $n-k$ qubits, yielding a new state of the form $|\psi_{comp}\rangle \otimes |0\rangle^{\otimes k}$, where $|\psi_{comp}\rangle$ contains $n-k$ qubits. This parameterized circuit depends on a set of trainable parameters that represent the nodes of the quantum autoencoder, which are updated during the training process to optimize the loss function.

The next stage is the bottleneck layer, where we disregard the remaining k qubits for the remainder of the circuit. The input state is now compressed. The final step is the decoder, where we add k new qubits to the circuit, all initialized in the $|0\rangle$ state. This layer is composed of a second parameterized circuit that is applied to the compressed state and these new qubits, decompressing the information and reconstructing the original input state. After the decoding is complete, the resulting output state is ideally a reconstruction of the initial input.

However, in a practical implementation, we cannot simply add or discard qubits in the middle of a quantum circuit. For this reason, we must include a reference state as well as some auxiliary qubits at the beginning of this model. We use the reference state as a known quantum state that serves as a baseline during the encoding process, allowing the quantum circuit to measure how closely the input has been transformed and reconstructed. This is initialized to $|0\rangle$, or ground state. Instead, the auxiliary qubit, or ancilla qubit, is used for the SWAP test, discussed in Appendix C.1. This specific qubit is measured to find the overall fidelity of the trash and reference states to guide the optimization process. In the end, the full input state is a combination of the original state, a reference state, an ancilla qubit, and a classical register for performing measurements (see Fig. C.23).

It is important to note that the training loss is computed on the bottleneck layer, rather than on the output layer. This approach is given by the fact that to measure the fidelity score on the trash space, we only need to perform a SWAP test between the k trash qubits and the reference state. This requires substantially fewer computational resources compared to evaluating the whole output, which is composed of $n+k$ qubits. Moreover, preparing multiple copies of a fixed reference state is more feasible than preparing multiple identical copies of a complex quantum state required for output verification. Romero et al. (2017) mathematically proves that this is equivalent to measuring fidelity on the final output. However, this equivalence is intuitive because a successful compression should leave the trash qubits with minimal information, making them equivalent to a fixed reference state. For this reason, measuring their fidelity to this reference state directly quantifies the compression quality.

C.1. SWAP test

The SWAP Test (Barenco et al., 1996; Buhrman et al., 2001) is used to evaluate how well the autoencoder reconstructs quantum states by

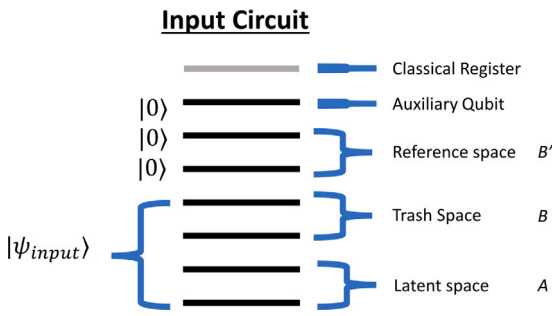


Fig. C.23. Representation of the input layer of the quantum autoencoder. *Source:* Image adapted from https://qiskit-community.github.io/qiskit-machine-learning/tutorials/12_quantum_autoencoder.html.

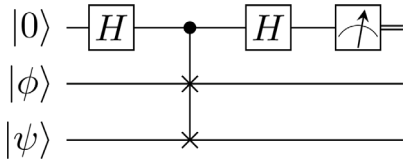


Fig. C.24. Representation of the SWAP test. The top qubit represents an ancilla qubit initialized to $|0\rangle$, the middle one represents the first quantum state $|\phi\rangle$, and the bottom one represents the second quantum state $|\psi\rangle$. Hadamard gates (H) are applied to the ancilla before and after the controlled-SWAP operation to create interference that reveals the similarity between the two states.

comparing the original input state to the reconstructed one, estimating how similar two quantum states are.

This operation can be executed on both a quantum simulator and a real quantum computer. The primary difference between the two setups is the presence of noise and quantum errors within real controlled gates, which can affect the accuracy of the measurements. To perform a SWAP test, as discussed in Section 2.2.2, we need to introduce an ancilla qubit at the start of the circuit, initialized in the state $|0\rangle$. Then, we apply a Hadamard gate on the ancilla qubit, which creates the superposition state $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and enables interference between the two quantum states under comparison. We then use the ancilla as the control qubit for a SWAP operation on the two states $|\psi\rangle$ and $|\phi\rangle$. If the ancilla is $|1\rangle$, then the two states are swapped. Finally, we reapply a Hadamard gate to the ancilla, and we measure it. This second gate converts the interference information into measurable probabilities to obtain the similarity of the two states. This is illustrated in Fig. C.24.

In a quantum autoencoder, the initial state $|\psi_{in}\rangle$ is processed through the encoder and the decoder, producing a reconstructed state $|\psi_{out}\rangle$. The SWAP test then compares these two states by placing them in separate quantum registers and performing the interference-based measurements. The probability of measuring the ancilla in $|0\rangle$ directly corresponds to the fidelity between $|\psi_{in}\rangle$ and $|\psi_{out}\rangle$, serving as the cost function for training the network. The goal is to maximize fidelity so that the model learns to optimize its encoder and decoder, leading to an improvement in the final reconstruction.

C.2. Feature vector encoding

In order to potentially improve the training quality and compression capabilities of this quantum approach, we adopt the Enhanced Feature Quantum Autoencoder (EF-QAE) strategy introduced in Bravo-Prieto (2021). This framework modifies the standard QAE by incorporating classical information directly into the PQC, as shown in Fig. C.25. In this case, the unitary evolution $U(\theta, \mathbf{x})$ applied to the initial quantum state depends not only on a set of trainable parameters θ associated with the ansatz gates but also on a classical feature vector $\mathbf{x}$. This serves to characterize the input state, potentially providing context based on the original data source or its properties. In our specific case, $\mathbf{x}$ is derived from the original time series data w corresponding to the input quantum state $|\psi_{in}\rangle$, and consists of statistical features such as the mean and standard deviation of w computed after scaling.

The mechanism of this approach consists of encoding the feature vector $\mathbf{x}$ into the angles of the single-qubit rotation gates within the variational ansatz, achieved using a linear function that mirrors the input modulation in classical neurons. For each rotation gate l in the ansatz (e.g., an R_y gate), a unique subset of $d_x + 1$ trainable parameters (w_l, b_l) is allocated from the main parameter vector θ. Here, d_x is the dimension of the feature vector $x = (x_1, \ldots, x_{d_x})$, while $w_l = (w_{l,1}, \ldots, w_{l,d_x})$ are trainable weights and b_l is a trainable bias. The rotation angle ϕ_l for that gate is then computed dynamically based on the current sample's feature vector $\mathbf{x}$:

$$\phi_l = \mathbf{w}_l \cdot \mathbf{x} + b_l = \sum_{k=1}^{d_x} w_{l,k} x_k + b_l \tag{C.1}$$

This calculated angle ϕ_l is then used in the corresponding rotation gate, e.g. $R_y(\phi_l)$. The entire set of parameters θ (comprising all $w_{l,k}$ and b_l for all rotations l) is adjusted during the training loop via a classical optimizer, driven by minimizing a cost function $C(\theta)$ evaluated through measurements or simulation of the quantum circuit's output. The use of a feature vector as input to this parameterized transformation is inspired by classical feed-forward networks. In fact, this encoding strategy is similar to how classical neural networks work: the assigned parameters $w_{l,k}$ and b_l can be seen as trainable weights and biases, respectively, while the rotation gate itself acts similarly to a non-linear activation function applied to the linearly computed angle ϕ_l. This approach gives the quantum circuit the ability to respond to the input's context, adapting its transformation based on the classical features $\mathbf{x}$ associated with the quantum state being processed.

C.3. Local cost function

This appendix provides a detailed description of the local cost function used to train the quantum autoencoder, as mentioned in Section 3.3.3 and further discussed in Cerezo et al. (2021) and Bravo-Prieto (2021).

The principle behind this metric is the strong connection between quantum data compression and decoupling. To achieve lossless compression, the autoencoder must decouple the trash qubits by completely isolating them from the rest of the system. During training, we monitor the state of these qubits to measure how effectively this decoupling occurs. The goal is to minimize the probability of measuring any state other than the target state, typically the $|0\rangle^{\otimes n_t}$ state, where n_t is the number of trash qubits. At the end of the compression stage, all trash qubits should be in the $|0\rangle$ state.

This process of decoupling is achieved by training the circuit using a local cost function, instead of a global one, in such a way as to avoid the training issues often associated with the latter, particularly in noisy or shallow quantum circuits. This strategy penalizes the measurement outcomes on individual trash qubits based on their distance from the target $|0\rangle$ state. Specifically, the Hamming distance (d_{H_j}) in the computational basis represents the number of symbols that are different in the binary representation between the measured string and the target string, and it is used to quantify this disparity.

The local cost function C, which the optimization process aims to minimize, is therefore defined as the weighted sum of measurement probabilities on the trash qubits:

$$C \equiv \sum_{k,j} d_{H_j} M_{k,j}, \tag{C.2}$$

where $M_{k,j}$ represents the outcome of the jth measurement on the kth trash qubit in the computational basis, and d_{H_j} is the corresponding Hamming distance from the $|0\rangle^{\otimes n_t}$ state. In the original

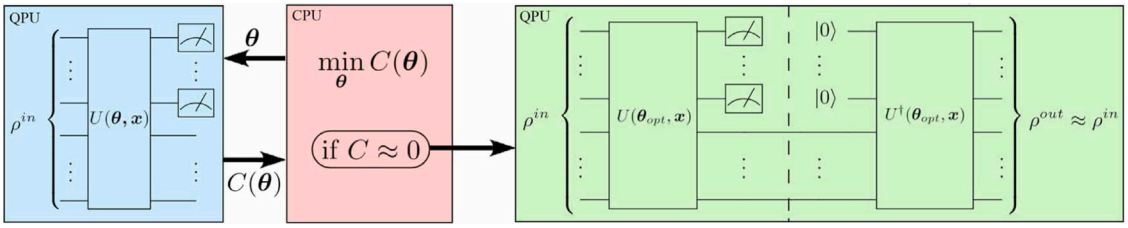


Fig. C.25. Representation of the Enhanced Feature Quantum Autoencoder (Bravo-Prieto, 2021). The system starts with a set of initial quantum states (ρ^{in}), a feature vector $\mathbf{x}$ that defines these states, and a shallow quantum circuit U. This feature vector, along with a set of variational parameters θ, is built into the circuit's logic, and these are updated through a quantum–classical optimization cycle that works to minimize a local cost function $C(\theta)$. Once we identified the optimal parameters θ_{opt}, the resulting circuit $U(\theta_{opt}, \mathbf{x})$ produces compressed quantum states $|\phi\rangle$ representing the specific model. Subsequently, these compressed states are decompressed by applying the adjoint circuit $U^{\dagger}(\theta_{opt}, \mathbf{x})$, yielding an output ρ^{out} that closely approximates the original input ρ^{in}.

Table C.14
MAPE on the test set obtained by the classical and quantum approaches for 128 bins.

Dataset configuration	Approach	Model details	Mean absolute percentage error
1000 Samples - 128 Bins	Classical	1081 Parameters	13.57%
	Classical	705 Parameters	14.00%
	Quantum	Real Amplitudes (10 parameters)	13.12%
5000 Samples - 128 Bins	Classical	1081 Parameters	6.87%
	Classical	705 Parameters	9.05%
	Quantum	Real Amplitudes (10 parameters)	11.72%

Table C.15
Results using the SWAP test as loss metric.

Qubits	Number of bins	Layers	Ansatz	Reconstruction train	Reconstruction test
4+1	32	1	Real Amplitudes	2.20×10^{-4}	2.20×10^{-4}
4+1	32	1	Efficient SU2	2.20×10^{-4}	2.19×10^{-4}
4+1	32	3	Real Amplitudes	7.78×10^{-4}	7.77×10^{-4}

paper (Bravo-Prieto, 2021), it is demonstrated that this cost function can be equivalently expressed using local Pauli-Z operators:

$$C = \frac{1}{2}\sum_{k=1}^{n_t}(1 - \langle Z_k \rangle), \tag{C.3}$$

where $\langle Z_k \rangle$ is the expectation value of the Pauli-Z operator measured on the kth trash qubit. This formulation is particularly advantageous because $\langle Z_k \rangle = +1$ if the qubit is in the $|0\rangle$ state and -1 if it is in the $|1\rangle$ state. Consequently, perfect compression and decoupling are achieved if the cost function C reaches its minimum value, obtained when all the trash qubits collapse to the $|0\rangle$ state. This local approach can be advantageous because it provides direct information about the compression process, as well as more robustness against issues such as barren plateaus and resilience to certain types of noise, for instance, global depolarizing noise.

Appendix D. Additional results

This appendix presents additional experimental results that provide further context for the findings discussed in Section 4.

As illustrated in Table C.14, in the case of 128-bin light curves, and therefore higher dimensionality, the quantum model struggles to outperform even classical models with fewer parameters. This is due to the greater complexity of the input data, which translates into deeper and more elaborate quantum circuits and encoding processes. These networks are harder to optimize and implement, even when running on a simulator.

Table C.15 shows the result obtained with the loss computed using the SWAP test as highlighted in Section 3.3.3, using 5000 samples. We can notice how the results tend to be very similar to the other cases we illustrated. However, the alternative cost function we employed significantly reduced training times. This efficiency comes from its design, which is based on measuring local observables on specific qubits, thereby eliminating the need for the more resource-intensive SWAP test in each iteration. When used for direct fidelity estimation, the latter increases the depth of the quantum circuit and overall complexity due to its need for an auxiliary qubit and controlled multi-qubit gates, resulting in longer execution times for each fidelity estimate.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., Zheng, X., 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. URL https://www.tensorflow.org/. Software available from tensorflow.org.

Abbas, A., Sutter, D., Zoufal, C., Lucchi, A., Figalli, A., Woerner, S., 2021. The power of quantum neural networks. Nat. Comput. Sci. 1 (6), 403–409. doi:10.1038/s43588-021-00084-1.

Acharya, R., Abanin, D.A., Aghababaie-Beni, L., Aleiner, I., Andersen, T.I., Ansmann, M., Arute, F., Arya, K., Asfaw, A., Astrakhantsev, N., Atalaya, J., Babbush, R., Bacon, D., Ballard, B., Bardin, J.C., Bausch, J., Bengtsson, A., Bilmes, A., Blackwell, S., Boixo, S., Bortoli, G., Bourassa, A., Bovaird, J., Brill, L., Broughton, M., Browne, D.A., Buchea, B., Buckley, B.B., Buell, D.A., Burger, T., Burkett, B., Bushnell, N., Cabrera, A., Campero, J., Chang, H.-S., Chen, Y., Chen, Z., Chiaro, B., Chik, D., Chou, C., Claes, J., Cleland, A.Y., Cogan, J., Collins, R., Conner, P., Courtney, W., Crook, A.L., Curtin, B., Das, S., Davies, A., De Lorenzo, L., Debroy, D.M., Demura, S., Devoret, M., Di Paolo, A., Donohoe, P., Drozdov, I., Dunsworth, A., Earle, C., Edlich, T., Eickbusch, A., Elbag, A.M., Elzouka, M., Erickson, C., Faoro, L., Farhi, E., Ferreira, V.S., Burgos, L.F., Forati, E., Fowler, A.G., Foxen, B., Ganjam, S., Garcia, G., Gasca, R., Genois, É., Giang, W., Gidney, C., Gilboa, D., Gosula, R., Dau, A.G., Graumann, D., Greene, A., Gross, J.A., Habegger, S., Hall, J., Hamilton, M.C., Hansen, M., Harrigan, M.P., Harrington, S.D., Heras, F.J.H., Heslin, S., Heu, P., Higgott, O., Hill, G., Hilton, J., Holland, G., Hong, S., Huang, H.-Y., Huff, A., Huggins, W.J., Ioffe, L.B., Isakov, S.V., Iveland, J., Jeffrey, E., Jiang, Z., Jones, C., Jordan, S., Joshi, C., Juhas, P., Kafri, D., Kang, H., Karamlou, A.H., Kechedzhi, K., Kelly, J., Khaire, T., Khattar, T., Khezri, M., Kim, S., Klimov, P.V., Klots, A.R., Kobrin, B., Kohli, P., Korotkov, A.N., Kostritsa, F., Kothari, R., Kozlovskii, B., Kreikebaum, J.M., Kurilovich, V.D., Lacroix, N., Landhuis, D., Lange-Dei, T., Langley, B.W., Laptev, P., Lau, K.-M., Le Guevel, L., Ledford, J., Lee, J., Lee, K., Lensky, Y.D., Leon, S., Lester, B.J., Li, W.Y., Li, Y., Lill, A.T., Liu, W.,

Livingston, W.P., Locharla, A., Lucero, E., Lundahl, D., Lunt, A., Madhuk, S., Malone, F.D., Maloney, A., Mandrà, S., Manyika, J., Martin, L.S., Martin, O., Martin, S., Maxfield, C., McClean, J.R., McEwen, M., Meeks, S., Megrant, A., Mi, X., Miao, K.C., Mieszala, A., Molavi, R., Molina, S., Montazeri, S., Morvan, A., Movassagh, R., Mruczkiewicz, W., Naaman, O., Neeley, M., Neill, C., Nersisyan, A., Neven, H., Newman, M., Ng, J.H., Nguyen, A., Nguyen, M., Ni, C.-H., Niu, M.Y., O'Brien, T.E., Oliver, W.D., Opremcak, A., Ottosson, K., Petukhov, A., Pizzuto, A., Platt, J., Potter, R., Pritchard, O., Pryadko, L.P., Quintana, C., Ramachandran, G., Reagor, M.J., Redding, J., Rhodes, D.M., Roberts, G., Rosenberg, E., Rosenfeld, E., Roushan, P., Rubin, N.C., Saei, N., Sank, D., Sankaragomathi, K., Satzinger, K.J., Schurkus, H.F., Schuster, C., Senior, A.W., Shearn, M.J., Shorter, A., Shutty, N., Shvarts, V., Singh, S., Sivak, V., Skruzny, J., Small, S., Smelyanskiy, V., Smith, W.C., Somma, R.D., Springer, S., Sterling, G., Strain, D., Suchard, J., Szasz, A., Sztein, A., Thor, D., Torres, A., Torunbalci, M.M., Vaishnav, A., Vargas, J., Vdovichev, S., Vidal, G., Villalonga, B., Heidweiller, C.V., Waltman, S., Wang, S.X., Ware, B., Weber, K., Weidel, T., White, T., Wong, K., Woo, B.W.K., Xing, C., Yao, Z.J., Yeh, P., Ying, B., Yoo, J., Yosri, N., Young, G., Zalcman, A., Zhang, Y., Zhu, N., Zobrist, N., 2024. Quantum error correction below the surface code threshold. Nature 638 (8052), 920–926. doi:10.1038/s41586-024-08449-y.

Aghaee, M., Ramirez, A.A., Alam, Z., Ali, R., Andrzejczuk, M., Antipov, A., Astafev, M., Barzegar, A., Bauer, B., Becker, J., Bhaskar, U.K., Bocharov, A., Boddapati, S., Bohn, D., Bommer, J., Bourdet, L., Bousquet, A., Boutin, S., Casparis, L., Chapman, B.J., Chatoor, S., Christensen, A.W., Chua, C., Codd, P., Cole, W., Cooper, P., Corsetti, F., Cui, A., Dalpasso, P., Dehollain, J.P., de Lange, G., de Moor, M., Ekefjärd, A., Dandachi, T.E., Saldaña, J.C.E., Fallahi, S., Galletti, L., Gardner, G., Govender, D., Griggio, F., Grigoryan, R., Grijalva, S., Gronin, S., Gukelberger, J., Hamdast, M., Hamze, F., Hansen, E.B., Heedt, S., Heidarnia, Z., Zamorano, J.H., Ho, S., Holgaard, L., Hornibrook, J., Indrapiromkul, J., Ingerslev, H., Ivancevic, L., Jensen, T., Jhoja, J., Jones, J., Kalashnikov, K.V., Kallaher, R., Kalra, R., Karimi, F., Karzig, T., King, E., Kloster, M.E., Knapp, C., Kocon, D., Koski, J., Kostamo, P., Kumar, M., Laeven, T., Larsen, T., Lee, J., Lee, K., Leum, G., Li, K., Lindemann, T., Looij, M., Love, J., Lucas, M., Lutchyn, R., Madsen, M.H., Madulid, N., Malmros, A., Manfra, M., Mantri, D., Markussen, S.B., Martinez, E., Mattila, M., McNeil, R., Mei, A.B., Mishmash, R.V., Mohandas, G., Mollgaard, C., Morgan, T., Moussa, G., Nayak, C., Nielsen, J.H., Nielsen, J.M., Nielsen, W.H.P., Nijholt, B., Nystrom, M., O'Farrell, E., Ohki, T., Otani, K., Wütz, B.P., Pauka, S., Petersson, K., Petit, L., Pikulin, D., Prawiroatmodjo, G., Preiss, F., Morejon, E.P., Rajpalke, M., Ranta, C., Rasmussen, K., Razmadze, D., Reentila, O., Reilly, D.J., Ren, Y., Reneris, K., Rouse, R., Sadovskyy, I., Sainiemi, L., Sanlorenzo, I., Schmidgall, E., Sfiligoj, C., Shah, M.B., Simoes, K., Singh, S., Sinha, S., Soerensen, T., Sohr, P., Stankevic, T., Stek, L., Stuppard, E., Suominen, H., Suter, J., Teicher, S., Thiyagarajah, N., Tholapi, R., Thomas, M., Toomey, E., Tracy, J., Turley, M., Upadhyay, S., Urban, I., Hoogdalem, K.V., Woerkom, D.J.V., Viazmitinov, D.V., Vogel, D., Watson, J., Webster, A., Weston, J., Winkler, G.W., Xu, D., Yang, C.K., Yucelen, E., Zeisel, R., Zheng, G., Zilke, J., 2024. Interferometric single-shot parity measurement in an inas-al hybrid device. arXiv:2401.09549. URL https://arxiv.org/abs/2401.09549.

Alaghbari, K.A., Lim, H.-S., Saad, M.H.M., Yong, Y.S., 2023. Deep autoencoder-based integrated model for anomaly detection and efficient feature extraction in IoT networks. IoT 4 (3), 345–365. doi:10.3390/iot4030016, URL https://www.mdpi.com/2624-831X/4/3/16.

Arute, F., et al., 2019. Quantum supremacy using a programmable superconducting processor. Nature 574 (7779), 505–510.

Bakumenko, A., Elragal, A., 2022. Detecting anomalies in financial data using machine learning algorithms. Systems 10 (5), doi:10.3390/systems10050130, URL https://www.mdpi.com/2079-8954/10/5/130.

Barenco, A., Berthiaume, A., Deutsch, D., Ekert, A., Jozsa, R., Macchiavello, C., 1996. Stabilisation of quantum computations by symmetrisation. arXiv:quant-ph/9604028. URL https://arxiv.org/abs/quant-ph/9604028.

Berahmand, K., Daneshfar, F., Salehi, E., Li, Y., Xu, Y., 2024. Autoencoders and their applications in machine learning: a survey. 57, doi:10.1007/s10462-023-10662-6,

Bertels, K., Turki, E., Sarac, T., Sarkar, A., Ashraf, I., 2024. Quantum computing – a new scientific revolution in the making. arXiv:2106.11840. URL https://arxiv.org/abs/2106.11840.

Biamonte, J., Wittek, P., Pancotti, N., Rebentrost, P., Wiebe, N., Lloyd, S., 2017. Quantum machine learning. Nature 549 (7671), 195–202. doi:10.1038/nature23474.

Bowles, J., Ahmed, S., Schuld, M., 2024. Better than classical? The subtle art of benchmarking quantum machine learning models. arXiv:2403.07059. URL https://arxiv.org/abs/2403.07059.

Bravo-Prieto, C., 2021. Quantum autoencoders with enhanced data encoding. Mach. Learn.: Sci. Technol. 2 (3), 035028. doi:10.1088/2632-2153/ac0616.

Buhrman, H., Cleve, R., Watrous, J., de Wolf, R., 2001. Quantum fingerprinting. Phys. Rev. Lett. 87 (16), doi:10.1103/physrevlett.87.167902.

Cerezo, M., Sone, A., Volkoff, T., Cincio, L., Coles, P.J., 2021. Cost function dependent barren plateaus in shallow parametrized quantum circuits. Nat. Commun. 12 (1), doi:10.1038/s41467-021-21728-w.

Cherenkov Telescope Array Consortium, Acharya, B.S., Agudo, I., Al Samarai, I., Alfaro, R., Alfaro, J., Alispach, C., Alves Batista, R., Amans, J.-P., Amato, E., Ambrosi, G., Antolini, E., Antonelli, L.A., Aramo, C., Araya, M., Armstrong, T., Arqueros, F., Arrabito, L., Asano, K., Ashley, M., Backes, M., Balazs, C., Balbo, M., Ballester, O., Ballet, J., Bamba, A., Barkov, M., Barres de Almeida, U., Barrio, J.A., Bastieri, D., Becherini, Y., Belfiore, A., Benbow, W., Berge, D., Bernardini, E., Bernardini, M.G., Bernardos, M., Bernlöhr, K., Bertucci, B., Biasuzzi, B., Bigongiari, C., Biland, A., Bissaldi, E., Biteau, J., Blanch, O., Blazek, J., Boisson, C., Bolmont, J., Bonanno, G., Bonardi, A., Bonavolontà, C., Bonnoli, G., Bosnjak, Z., Böttcher, M., Braiding, C., Bregeon, J., Brill, A., Brown, A.M., Brun, P., G., B., Buanes, T., Buckley, J., Bugaev, V., Bühler, R., Bulgarelli, A., Bulik, T., Burton, M., Burtovoi, A., Busetto, G., Canestrari, R., Capalbi, M., Capitanio, F., Caproni, A., Caraveo, P., Cárdenas, V., Carlile, C., Carosi, R., Carquín, E., Carr, J., Casanova, S., Cascone, E., Catalani, F., Catalano, O., Cauz, D., Cerruti, M., Chadwick, P., Chaty, S., Chaves, R.C.G., Chen, A., Chen, X., Chernyakova, M., Chikawa, M., Christov, A., Chudoba, J., Cieślar, M., Coco, V., Colafrancesco, S., Colin, P., Conforti, V., Connaughton, V., Conrad, J., Contreras, J.L., Cortina, J., Costa, A., Costantini, H., Cotter, G., Covino, S., Crocker, R., Cuadra, J., Cuevas, O., Cumani, P., D'Aì, A., D'Ammando, F., D'Avanzo, P., D'Urso, D., Daniel, M., Davids, I., Dawson, B., Dazzi, F., De Angelis, A., de Cássia dos Anjos, R., De Cesare, G., De Franco, A., de Gouveia Dal Pino, E.M., de la Calle, I., de los Reyes Lopez, R., De Lotto, B., De Luca, A., De Lucia, M., de Naurois, M., de Oña Wilhelmi, E., De Palma, F., De Persio, F., de Souza, V., Deil, C., Del Santo, M., Delgado, C., della Volpe, D., Di Girolamo, T., Di Pierro, F., Di Venere, L., Díaz, C., Dib, C., Diebold, S., Djannati-Ataï, A., Domínguez, A., Dominis Prester, D., Dorner, D., Doro, M., Drass, H., Dravins, D., Dubus, G., Dwarkadas, V.V., Ebr, J., Eckner, C., Egberts, K., Eineckes, S., Ekoume, T.R.N., Elsässer, D., Ernenwein, J.-P., Espinoza, C., Evoli, C., Fairbairn, M., Falceta-Goncalves, D., Falcone, A., Farnier, C., Fasola, G., Fedorova, E., Fegan, S., Fernandez-Alonso, M., Fernández-Barral, A., Ferrand, G., Fesquet, M., Filipovic, M., Fioretti, V., Fontaine, G., Fornasa, M., Fortson, L., Freixas Coromina, L., Fruck, C., Fujita, Y., Fukazawa, Y., Funk, S., Füßling, M., Gabici, S., Gadola, A., Gallant, Y., Garcia, B., Garcia López, R., Garczarczyk, M., Gaskins, J., Gasparetto, T., Gaug, M., Gerard, L., Giavitto, G., Giglietto, N., Giommi, P., Giordano, F., Giro, E., Giroletti, M., 2019. Science with the Cherenkov Telescope Array. doi:10.1142/10986.

Chollet, F., et al., 2015. Keras. https://keras.io.

Crupi, R., Dilillo, G., Ward, K., Bissaldi, E., Fiore, F., Vacchi, A., 2023. Searching for long faint astronomical high energy transients: a data driven approach. arXiv:2303.15936. URL https://arxiv.org/abs/2303.15936.

Deutsch, D., Jozsa, R., 1992. Rapid solution of problems by quantum computation. Proc. R. Soc. Lond. Ser. A: Math. Phys. Sci. 439 (1907), 553–558.

Eisert, J., Wolf, M.M., 2006. Quantum computing. In: Handbook of Nature-Inspired and Innovative Computing: Integrating Classical Models with Emerging Technologies. Springer US, Boston, MA, pp. 253–286. doi:10.1007/0-387-27705-6_8.

Farsian, F., Parmiggiani, N., Rizzo, A., Panebianco, G., Bulgarelli, A., Schilliró, F., Burigana, C., Cardone, V., Cappelli, L., Meneghetti, M., Murante, G., Sarracino, G., Scaramella, R., Testa, V., Trombetti, T., 2025. Benchmarking quantum convolutional neural networks for signal classification in simulated Gamma-ray burst detection. In: 2025 33rd Euromicro International Conference on Parallel, Distributed, and Network-Based Processing. PDP, IEEE, pp. 372–380. doi:10.1109/pdp66500.2025.00059.

Fernando, T., Gammulle, H., Denman, S., Sridharan, S., Fookes, C., 2021. Deep learning for medical anomaly detection – a survey. arXiv:2012.02364. URL https://arxiv.org/abs/2012.02364.

Fiore, F., Burderi, L., Lavagna, M., Bertacin, R., Evangelista, Y., Campana, R., Fuschino, F., Lunghi, P., Monge, A., Negri, B., Pirrotta, S., Puccetti, S., Sanna, A., Amarilli, F., Ambrosino, F., Amelino-Camelia, G., Anitra, A., Auricchio, N., Barbera, M., Bechini, M., Bellutti, P., Bertuccio, G., Cao, J., Ceraudo, F., Chen, T., Cinelli, M., Citossi, M., Clerici, A., Colagrossi, A., Curzel, S., Della Casa, G., Demenev, E., Del Santo, M., Dilillo, G., Di Salvo, T., Efremov, P., Feroci, M., Feruglio, C., Ferrandi, F., Fiorini, M., Fiorito, M., Frontera, F., Gacnik, D., Galgoczi, G., Gao, N., Gambino, A.F., Gandola, M., Ghirlanda, G., Gomboc, A., Grassi, M., Guzman, A., Karlica, M., Kostic, U., Labanti, C., La Rosa, G., Lo Cicero, U., Lopez-Fernandez, B., Malcovati, P., Maselli, A., Manca, A., Mele, F., Milankovich, D., Morgante, G., Nava, L., Nogara, P., Ohno, M., Ottolina, D., Pasquale, A., Pal, A., Perri, M., Piazzolla, R., Piccinin, M., Pliego-Caballero, S., Prinetto, J., Pucacco, G., Rashevsky, A., Rashevskaya, I., Riggio, A., Ripa, J., Russo, F., Papitto, A., Piranomonte, S., Santangelo, A., Scala, F., Sciarrone, G., Selcan, D., Silvestrini, S., Sottile, G., Rotovnik, T., Tenzer, C., Troisi, I., Vacchi, A., Virgilli, E., Werner, N., Wang, l., Xu, Y., Zampa, G., Zampa, N., Zanotti, G., 2020. The HERMES-technologic and scientific pathfinder. In: den Herder, J.-W.A., Nakazawa, K., Nikzad, S. (Eds.), Space Telescopes and Instrumentation 2020: Ultraviolet To Gamma Ray. SPIE, doi:10.1117/12.2560680.

Fiore, F., Werner, N., Behar, E., 2021. Distributed architectures and constellations for Gamma-ray burst science. arXiv:2112.08982. URL https://arxiv.org/abs/2112.08982.

Frehner, R., Stockinger, K., 2024. Applying quantum autoencoders for time series anomaly detection. arXiv:2410.04154. URL https://arxiv.org/abs/2410.04154.

Garcia-Cifuentes, K., Becerra, R.L., Colle, F.D., 2024. ClassiPyGRB: Machine learning-based classification and visualization of Gamma ray bursts using t-SNE. J. Open Source Softw. 9 (96), 5923. doi:10.21105/joss.05923.

Gill, S.S., Cetinkaya, O., Marrone, S., Claudino, D., Haunschild, D., Schlote, L., Wu, H., Ottaviani, C., Liu, X., Machupalli, S.P., Kaur, K., Arora, P., Liu, J., Farouk, A., Song, H.H., Uhlig, S., Ramamohanarao, K., 2025. Quantum computing: vision and

challenges. In: Quantum Computing. Elsevier, pp. 19–42. doi:10.1016/b978-0-443-29096-1.00008-8.
Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep Learning. MIT Press, http://www.deeplearningbook.org.
Grover, L.K., 1996. A fast quantum mechanical algorithm for database search. arXiv:quant-ph/9605043. URL https://arxiv.org/abs/quant-ph/9605043.
Harrow, A.W., Hassidim, A., Lloyd, S., 2009. Quantum algorithm for linear systems of equations. Phys. Rev. Lett. 103 (15), doi:10.1103/physrevlett.103.150502.
Hinton, G., Salakhutdinov, R., 2006. Reducing the dimensionality of data with neural networks. Sci. (New York, N.Y.) 313, 504–507. doi:10.1126/science.1127647.
Javadi-Abhari, A., Treinish, M., Krsulich, K., Wood, C.J., Lishman, J., Gacon, J., Martiel, S., Nation, P.D., Bishop, L.S., Cross, A.W., Johnson, B.R., Gambetta, J.M., 2024. Quantum computing with qiskit. arXiv:2405.08810. URL https://arxiv.org/abs/2405.08810.
Karwin, C., Zoglauer, A., Buhler, J., et al., 2025. COSI data challenges. doi:10.5281/zenodo.15126188.
von Kienlin, A., Meegan, C.A., Paciesas, W.S., Bhat, P.N., Bissaldi, E., Briggs, M.S., Burns, E., Cleveland, W.H., Gibby, M.H., Giles, M.M., Goldstein, A., Hamburg, R., Hui, C.M., Kocevski, D., Mailyan, B., Malacaria, C., Poolakkil, S., Preece, R.D., Roberts, O.J., Veres, P., Wilson-Hodge, C.A., 2020. The fourth Fermi-GBM Gamma-ray burst catalog: A decade of data. Astrophys. J. 893 (1), 46. doi:10.3847/1538-4357/ab7a18.
Kingma, D.P., Ba, J., 2017. Adam: A method for stochastic optimization. arXiv:1412.6980. URL https://arxiv.org/abs/1412.6980.
Klusch, M., Lässig, J., Müssig, D., Macaluso, A., Wilhelm, F.K., 2024. Quantum artificial intelligence: A brief survey. arXiv:2408.10726. URL https://arxiv.org/abs/2408.10726.
Kumar, P., Zhang, B., 2015. The physics of gamma-ray bursts & relativistic jets. Phys. Rep. 561, 1–109. doi:10.1016/j.physrep.2014.09.008, URL https://www.sciencedirect.com/science/article/pii/S0370157314003846.
Li, T.P., Ma, Y.Q., 1983. Analysis methods for results in gamma-ray astronomy.. Astrophys. J. 272, 317–324. doi:10.1086/161295.
Liu, S., Mallol-Ragolta, A., Parada-Cabeleiro, E., Qian, K., Jing, X., Kathan, A., Hu, B., Schuller, B.W., 2022. Audio self-supervised learning: A survey. arXiv:2203.01205. URL https://arxiv.org/abs/2203.01205.
Liu, J., Xie, G., Wang, J., Li, S., Wang, C., Zheng, F., Jin, Y., 2024. Deep industrial image anomaly detection: A survey. Mach. Intell. Res. 21 (1), 104–135. doi:10.1007/s11633-023-1459-z.
Lunardi, W.T., Lopez, M.A., Giacalone, J.-P., 2022. ARCADE: Adversarially regularized convolutional autoencoder for network anomaly detection. arXiv:2205.01432. URL https://arxiv.org/abs/2205.01432.
Macaluso, A., Orazi, F., Klusch, M., Lodi, S., Sartori, C., 2022. A variational algorithm for quantum single layer perceptron. In: International Conference on Machine Learning, Optimization, and Data Science. Springer, pp. 341–356.
Martinez, I., 2023. The cosipy library: Cosi's high-level analysis software. In: Proceedings of 38th International Cosmic Ray Conference — PoS(ICRC2023). In: ICRC2023, Sissa Medialab, p. 858. doi:10.22323/1.444.0858.
Memon, Q.A., Al Ahmad, M., Pecht, M., 2024. Quantum computing: Navigating the future of computation, challenges, and technological breakthroughs. Quantum Rep. 6 (4), 627–663. doi:10.3390/quantum6040039, URL https://www.mdpi.com/2624-960X/6/4/39.
Ngairangbam, V.S., Spannowsky, M., Takeuchi, M., 2022. Anomaly detection in high-energy physics using a quantum autoencoder. Phys. Rev. D 105, 095004. doi:10.1103/PhysRevD.105.095004, URL https://link.aps.org/doi/10.1103/PhysRevD.105.095004.
Nielsen, M.A., Chuang, I.L., 2000. Quantum Computation and Quantum Information. Cambridge University Press.
Nielsen, M.A., Chuang, I.L., 2010. Quantum Computation and Quantum Information: 10th Anniversary Edition. Cambridge University Press.
Parmiggiani, N., Bulgarelli, A., Castaldini, L., De Rosa, A., Di Piano, A., Falco, R., Fioretti, V., Macaluso, A., Panebianco, G., Ursi, A., Pittori, C., Tavani, M., Beneventano, D., 2024. A new deep learning model to detect Gamma-ray bursts in the AGILE anticoincidence system. Astrophys. J. 973 (1), 63. doi:10.3847/1538-4357/ad64cd.
Parmiggiani, N., Bulgarelli, A., Fioretti, V., Di Piano, A., Giuliani, A., Longo, F., Verrecchia, F., Tavani, M., Beneventano, D., Macaluso, A., 2021. A deep learning method for AGILE-GRID Gamma-ray burst detection. Astrophys. J. 914 (1), 67. doi:10.3847/1538-4357/abfa15.
Parmiggiani, N., Bulgarelli, A., Panebianco, G., Burns, E., Neights, E., Fioretti, V., Martinez-Castellanos, I., Castaldini, L., Ciabattoni, A., Di Piano, A., Falco, R., Gallego, S., Mustafa, G., Patel, P., Rizzo, A., Wulf, E., Hartmann, D., Kierans, C.A., Tomsick, J.A., Zoglauer, A., 2026. COSI Short Gamma-Ray Burst Localization Using BGO Shield Data. Astrophys. J. 997 (2), 135. doi:10.3847/1538-4357/ae25fc.
Parmiggiani, N., Bulgarelli, A., Ursi, A., Macaluso, A., Piano, A.D., Fioretti, V., Aboudan, A., Baroncelli, L., Addis, A., Tavani, M., Pittori, C., 2023. A deep-learning anomaly-detection method to identify Gamma-ray bursts in the ratemeters of the AGILE anticoincidence system. Astrophys. J. 945 (2), 106. doi:10.3847/1538-4357/acba0a.
Pokharel, B., Lidar, D.A., 2024. Better-than-classical grover search via quantum error detection and suppression. Npj Quantum Inf. 10 (1), 23.
Powell, M.J.D., 1994. A direct search optimization method that models the objective and constraint functions by linear interpolation. In: Advances in Optimization and Numerical Analysis. Kluwer Academic (Dordrecht), pp. 51–67.
Powell, M.J.D., 1998. Direct search algorithms for optimization calculations. Acta Numer. 7, 287–336.
Powell, M.J.D., 2007. A view of algorithms for optimization without derivatives. Technical Report DAMTP 2007/NA03, Cambridge University.
Preskill, J., 2018. Quantum computing in the NISQ era and beyond. Quantum 2, 79. doi:10.22331/q-2018-08-06-79.
Rieffel, E.G., Polak, W.H., 2011. Quantum Computing: a Gentle Introduction. MIT Press.
Rizzo, A., Parmiggiani, N., Bulgarelli, A., Macaluso, A., Fioretti, V., Castaldini, L., Di Piano, A., Panebianco, G., Pittori, C., Tavani, M., Sartori, C., Burigana, C., Cardone, V., Farsian, F., Meneghetti, M., Murante, G., Scaramella, R., Schillirò, F., Testa, V., Trombetti, T., 2025. Quantum Convolutional Neural Networks for the Detection of Gamma-Ray Bursts in the AGILE Space Mission Data. In: Jacques, A., Seaman, R., Gandilo, N., Linder, T. (Eds.), Astronomical Data Analysis Software and Systems XXXIII. In: ASP Conference Series, vol. 541, Astronomical Society of the Pacific, San Francisco, p. 222, Preprint arXiv:2404.14133.
Roffe, J., 2019. Quantum error correction: an introductory guide. Contemp. Phys. 60 (3), 226–245. doi:10.1080/00107514.2019.1667078.
Romero, J., Olson, J.P., Aspuru-Guzik, A., 2017. Quantum autoencoders for efficient compression of quantum data. Quantum Sci. Technol. 2 (4), 045001. doi:10.1088/2058-9565/aa8072.
Schuld, M., Killoran, N., 2019. Quantum machine learning in feature Hilbert spaces. Phys. Rev. Lett. 122 (4), doi:10.1103/physrevlett.122.040504.
Shor, P.W., 1997. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM J. Comput. 26 (5), 1484–1509. doi:10.1137/S0097539795293172, arXiv:https://doi.org/10.1137/S0097539795293172.
Sim, S., Johnson, P.D., Aspuru-Guzik, A., 2019. Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms. Adv. Quantum Technol. 2 (12), doi:10.1002/qute.201900070.
Tavani, M., et al., 2009. The AGILE mission. Astron. Astrophys. 502 (3), 995–1013. doi:10.1051/0004-6361/200810527.
Tomsick, J., Boggs, S., Zoglauer, A., Hartmann, D.H., Ajello, M., Burns, E., Fryer, C., Karwin, C., Kierans, C., Lowell, A., Malzac, J., Roberts, J., Saint-Hilaire, P., Shih, A., Siegert, T., Sleator, C., Takahashi, T., Tavecchio, F., Wulf, E., Beechert, J., Gulick, H., Joens, A., Lazar, H., Neights, E., Martinez Oliveros, J.C., Matsumoto, S., Melia, T., Yoneda, H., Amman, M., Bal, D., von Ballmoos, P., Bates, H., Böttcher, M., Bulgarelli, A., Cavazzuti, E., Chang, H.K., Chen, C., Chu, C.Y., Ciabattoni, A., Costamante, L., Dreyer, L., Fioretti, V., Fenu, F., Gallego, S., Ghirlanda, G., Grove, E., Huang, C.Y., Jean, P., Khatiya, N., Knödlseder, J., Kraus, M., Leising, M., Lewis, T., Lommler, J., Marcotulli, L., Martinez Castellanos, I., Mittal, S., Negro, M., Al Nussirat, S., Nakazawa, K., Oberlack, U., Palmore, D., Panebianco, G., Parmiggiani, N., Pike, S., Rogers, F., Schutte, H., Sheng, Y., Smale, A., Smith, J.R., Trigg, A., Venters, T., Watanabe, Y., Zhang, H., 2024. The Compton Spectrometer and Imager. In: 38th International Cosmic Ray Conference. p. 745. doi:10.48550/arXiv.2308.12362, arXiv:2308.12362.
Tomsick, J.A., Zoglauer, A., Sleator, C., Lazar, H., Beechert, J., Boggs, S., Roberts, J., Siegert, T., Lowell, A., Wulf, E., Grove, E., Phlips, B., Brandt, T., Smale, A., Kierans, C., Burns, E., Hartmann, D., Leising, M., Ajello, M., Fryer, C., Amman, M., Chang, H.-K., Jean, P., von Ballmoos, P., 2019. The compton spectrometer and imager. arXiv:1908.04334. URL https://arxiv.org/abs/1908.04334.
Weigold, M., et al., 2021. Encoding patterns for quantum algorithms. IET Quantum Commun. 2 (4), 141–152. doi:10.1049/qtc2.12032.
Wiebe, N., Kapoor, A., Svore, K.M., 2015. Quantum deep learning. arXiv:1412.3489. URL https://arxiv.org/abs/1412.3489.
Wilde, M.M., 2012. Quantum Information Theory. Cambridge University Press.
Wootters, W.K., Zurek, W.H., 1982. A single quantum cannot be cloned. Nature 299 (5886), 802–803. doi:10.1038/299802a0.
Yan, P., Abdulkadir, A., Luley, P.-P., Rosenthal, M., Schatte, G.A., Grewe, B.F., Stadelmann, T., 2024. A comprehensive survey of deep transfer learning for anomaly detection in industrial time series: Methods, applications, and directions. IEEE Access 12, 3768–3789. doi:10.1109/access.2023.3349132.
Yoffe, D., Natan, A., Makmal, A., 2023. A qubit-efficient variational selected configuration-interaction method. arXiv:2302.06691. URL https://arxiv.org/abs/2302.06691.
Zoglauer, A., Andritschke, R., Schopper, F., 2006. MEGAlib – the medium energy Gamma-ray astronomy library. New Astron. Rev. 50 (7), 629–632. doi:10.1016/j.newar.2006.06.049, URL https://www.sciencedirect.com/science/article/pii/S1387647306000972. Astronomy with Radioactivities.
Zoglauer, A., Siegert, T., Lowell, A., Mochizuki, B., Kierans, C., Sleator, C., Hartmann, D.H., Lazar, H., Gulick, H., Beechert, J., Roberts, J.M., Tomsick, J.A., Leising, M.D., Pellegrini, N., Boggs, S.E., Brandt, T.J., 2021. COSI: From calibrations and observations to all-sky images. arXiv:2102.13158. URL https://arxiv.org/abs/2102.13158.
Zoglauer, A., et al., 2024. The COSI Science Data Pipeline. In: AAS/High Energy Astrophysics Division. p. 105.09.