ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

IBM

Master Thesis

Quantum Generative Adversarial Networks for Anomaly Detection in High Energy Physics

Elie Bermot*†

August 30, 2022

Supervisors:
Dr. Ivano Tavernelli *
Dr. Christa Zoufal *
Dr. Francesco Tacchino *
Prof. Renato Renner †

*IBM Quantum, IBM Research Europe – Zurich, Rueschlikon 8803, Switzerland
†ETH Zurich, Zurich 8092, Switzerland

Abstract

The standard model (SM) of particle physics is an accomplishment of decades of theoretical and experimental work. It describes almost all known elementary particles and their interactions: strong, weak and electromagnetic. However, some of the observed events occurring in a particle accelerator, e.g., the Large Hadron Collider, correspond to anomalous and unpredictable events, whereby the underlying physics is not governed by the SM [1]. The detection of anomalous events and, the corresponding potential discovery of new fundamental physics, is far from trivial. In fact – until recently – most of the high energy physics data analysis relied on model-specific selection process. In this work, we consider a quantum generative adversarial network (qGAN) based scoring function that identifies anomalous events by determining whether an event is characteristic for a certain background distribution such as the SM. We implement the qGAN training and the anomaly detection scheme with Qiskit and test the method on proof-of-principle examples using numerical simulations as well as actual IBM quantum processors.

Acknowledgements

First and foremost, I would like to thank Ivano Tavernelli for helping me throughout my project and for giving me the chance to work in the thriving environment of IBM Quantum in Zurich.

I would also like to thank Renato Renner for agreeing to be my ETH supervisor.

Naturally, I could not be more greatful for having Christa Zoufal as my supervisor. Her endless time, her rigour, and her continuous motivation speeches helped me to always challenge myself, and to be proud of my work.

I would also like to thank Michele Grossi, Sofia Vallecorsa and the entire CERN QTI team for their collaboration and their valuable help and advices.

I thank Panagiotis Barkoutsos and Julian Schuhmacher, for sharing their knowledge and engaging in very helpful discussions.

Lastly, I would like to thank the entire team of IBM Quantum for all of the great discussions, coffee breaks, pedantle, and of course the great help that they all gave me. I can easily say that their advice and interest in my work helped me go deeper into my project and to be a better scientist.

Finally, I thank my friends and family for their encouragement and support.

Contents

Abstract i

Acknowledgements iii

List of Figures vii

List of Tables x

1. Introduction 1

2. High Energy Physics 3
1. Standard Model of Particle Physics . . . 3
2. Beyond the Standard Model . . . 4
3. Challenges for future experiments at the LHC . . . 5
   3.1. The LHC . . . 5
   3.2. The future run of the LHC . . . 6

3. Generative Adversarial Models 9
1. Classical GAN . . . 9
   1.1. Definition . . . 9
   1.2. Training with classical GAN . . . 9
   1.3. Training with WGAN . . . 11
2. Quantum GAN . . . 13
   2.1. Related work . . . 13
   2.2. Data embedding . . . 14
   2.3. Definition . . . 16
   2.4. Training . . . 18

4. Anomaly detection with Generative Adversarial Networks 21
1. Related work . . . 21
   1.1. Anomaly detection with classical unsupervised learning . . . 21
   1.2. Anomaly detection with qGANs . . . 22
2. Anomaly score . . . 22
   2.1. Classical anomaly score . . . 22
   2.2. Quantum anomaly score . . . 23
3. Classification . . . 23
   3.1. Metrics . . . 23
   3.2. Anomaly detection algorithm . . . 25

5. Methodology 27
1. Data sets . . . 27
2. Feature Selection . . . 28
3. Quantum representation of the data set . . . 29

6. Results and Discussion **31**
1. Recognizing a Standard Model event 31
2. Classical benchmark 31
2.1. Anomaly detection with 2 features 32
2.2. Anomaly detection with 8 features 33
2.3. Comparison of the different performances w.r.t. dimensions 37
3. Quantum benchmark 37
3.1. Anomaly detection with 2 features 38
3.2. Anomaly detection with 8 features 43
3.3. Comparison of the different performances w.r.t. dimensions 51

7. Conclusion **55**

Bibliography **64**

Declaration of originality **65**

List of Figures

1.1. Workflow of the presented anomaly scoring method. We first reduced the data sets through feature reduction techniques which are applied to testing data sets. Then we train our qGAN on the embedded SM data points. Lastly, we compute the anomaly score on the embedded test data set. We use the value of the anomaly score on the Standard Model (SM) to define a threshold. This threshold is used to classify a data point as an anomaly. 2

2.1. The particle spectrum of the Standard Model. 4
2.2. Slice view of the CMS detector [17] and of the paths of different collision products. Charged particles have a curved path due to the electromagnetic field. 6

3.1. Working principle of the Generative Adversarial Networks (GAN): a generator $\mathcal{G}^{\theta_G}$ samples a data sample, and the discriminator $\mathcal{D}^{\theta_D}$ receives a data sample at a time from either the real data set or from the generated samples and independently classify it. The aim of the training of the GAN is done simultaneously: the generator tries to fool as much as possible the discriminator that its data is the real one while the discriminator tries to distinguish as much as possible the real data from the generated data . . 10
3.2. General Quantum Generative Adversarial Networks (qGAN) to learn the training quantum data set $\{|\mathbf{x}_1\rangle, \ldots, |\mathbf{x}_M\rangle\}$. At each training step, the discriminator receives the generated state $|\mathcal{G}\rangle$ and a batch of the embedded data points $|\mathbf{x}_i\rangle = \mathcal{R}_i |0\rangle$. 17
3.3. Ansatz for the generator $\mathcal{G}(\theta_G)$. $\theta_G^{i,j}$ corresponds to the parameter of R_y rotation acting on the i qubit on the j layer. 17
3.4. Ansatz for the discriminator $\mathcal{D}(\theta_D)$. $\theta_D^{i,j}$ corresponds to the j parameter of R_y rotation acting on the i qubit. 18

6.1. Cumulative explained variance of the test data point w.r.t. the number of principal components. A Principal Component Analysis (PCA) is applied to each data set and the cumulative variance is computed. 31
6.2. Loss functions of different GAN and Wasserstein Generative Adversarial Networks (WGAN) architectures for 2 features: (a) "Vanilla" GAN setup . (b) Sigmoid WGAN setup. In all plots, the blue curve corresponds to the generator loss function and in orange the discriminator loss function. . 33
6.3. Anomaly score scheme with 2 features in the "vanilla" setup of the classical benchmark : (a) Anomaly score w.r.t. α parameters while (b) corresponds to the optimal configuration for all the data point in the test data set and $\alpha = 0$. In all plots, the blue points correspond to the SM data points and the area corresponds to the 95%-confidence interval. The orange points correspond to the graviton data points and the orange area the 95% confidence interval and in orange the discriminator loss function. The green point correspond to the Higgs data points and the green area

denotes the 95% confidence interval. 34
6.4. ROC curves of the different GAN/WGAN architectures with 2 features(a) and with 8 features (b). 34
6.5. Loss functions of different GAN and WGAN architectures with 8 features: (a) "Vanilla" GAN setup and (b) Sigmoid WGAN setup. In all plots, the blue curve corresponds to the generator loss function and in orange the discriminator loss function. 35
6.6. Anomaly score scheme with 8 features in the classical benchmark. (a) shows the anomaly score mean w.r.t. to the α parameters. The optimal configuration for $\alpha = 0$ is shown in the right panel (b) for the whole testing data set. In all plots, the blue points correspond to the SM data points and the area corresponds to the 95%-confidence interval. The orange points correspond to the Graviton data points and the orange area the 95% confidence interval and in orange the discriminator loss function. The green point correspond to the Higgs data points and the green area denotes the 95% confidence interval. 36
6.7. ROC Curve for different dimensions in the classical framework in the vanilla setup (a) and in the WGAN sigmoid setup. 38
6.8. Loss function of the quantum GAN using statevector with 2 features. The blue curve corresponds to the generator loss function and the red one corresponds to the discriminator loss function. 39
6.9. Quantum anomaly score scheme with 2 features on a noiseless simulator. (a) shows the anomaly score mean w.r.t. to the α parameters. The optimal configuration is shown in the right panel (b) for the whole testing data set and $\alpha = 0$. In all plots, the blue points correspond to the SM data points and the area corresponds to the 95%-confidence interval. The orange points correspond to the Graviton data points and the orange area the 95% confidence interval and in orange the discriminator loss function. The green point correspond to the Higgs data points and the green area denotes the 95% confidence interval. 40
6.10. ROC curve in 2 features of the best classical GAN architecture and of the noiseless quantum simulations. In this case, the quantum framework yields similar results to the classical one. 40
6.11. Loss function of the quantum GAN using a noisy simulator with 2 features. The blue curve corresponds to the generator loss function and the red one corresponds to the discriminator loss function. 41
6.12. Quantum anomaly score scheme with 2 features on a noisy simulator. (a) shows the anomaly score mean w.r.t. to the α parameters. The optimal configuration is shown in the right panel (b) for the whole testing data set and $\alpha = 0$. In all plots, the blue points correspond to the SM data points and the area corresponds to the 95%-confidence interval. The orange points correspond to the Graviton data points and the orange area the 95% confidence interval and in orange the discriminator loss function. The green point correspond to the Higgs data points and the green area denotes the 95% confidence interval. 42
6.13. ROC curve in 2 features of the best classical GAN architecture, of the noiseless and noisy quantum simulations. In this case, the classical frame-

work outperforms all of the quantum simulations. 43
6.14. Evolution of the loss functions with 2 features on real quantum hardware. The blue curve corresponds to the generator loss function and the red one corresponds to the discriminator loss function. 44
6.15. Quantum anomaly score scheme with 2 features on a real hardware. (a) shows the anomaly score mean w.r.t. to the α parameters. The optimal configuration is shown in the right panel (b) for the whole testing data set and $\alpha = 0.75$. In all plots, the blue points correspond to the SM data points and the area corresponds to the 95%-confidence interval. The orange points correspond to the Graviton data points and the orange area the 95% confidence interval and in orange the discriminator loss function. The green point correspond to the Higgs data points and the green area denotes the 95% confidence interval. 44
6.16. ROC curve in 2 features of the best classical GAN architecture, of the noiseless and noisy quantum simulations and of the hardware simulation. In this case, the classical framework outperforms all of the quantum simulations. However, the hardware run yields a better AUC score compared to the other quantum simulations. 45
6.17. Loss function of the quantum GAN using statevector with 8 features. The blue curve corresponds to the generator loss function and the red one corresponds to the discriminator loss function. 46
6.18. Quantum anomaly score scheme with 8 features using a noiseless simulator. (a) shows the anomaly score mean w.r.t. to the α parameters. The optimal configuration $\alpha = 0.25$ is shown in the right panel (b) for the whole testing data set. In all plots, the blue points correspond to the SM data points and the area corresponds to the 95%-confidence interval. The orange points correspond to the Graviton data points and the orange area the 95% confidence interval and in orange the discriminator loss function. The green point correspond to the Higgs data points and the green area denotes the 95% confidence interval. 47
6.19. ROC curve in 8 features of the best classical GAN architecture and of the noiseless quantum simulations. In this case, the classical framework gives similar results than with the quantum setting. 48
6.20. Loss function of the quantum GAN using a noisy simulator with 8 features. The blue curve corresponds to the generator loss function and the red one corresponds to the discriminator loss function. 49
6.21. Quantum anomaly score scheme with 8 features on a noisy simulator. (a) shows the anomaly score mean w.r.t. to the α parameters. The optimal configuration is shown in the right panel (b) for the whole testing data set. In all plots, the blue points correspond to the SM data points and the area corresponds to the 95%-confidence interval. The orange points correspond to the Graviton data points and the orange area the 95% confidence interval and in orange the discriminator loss function. The green point correspond to the Higgs data points and the green area denotes the 95% confidence interval. 50
6.22. ROC curve in 8 features of the noisy quantum simulations compared to the best classical GAN architecture and to the noiseless quantum simulations.

In this case, the classical framework gives similar results than with the quantum setting. 51
6.23. ROC Curve for different dimensions in the quantum setting with noiseless simulations (a) and noisy simulations (b). 52

List of Tables

3.1. Comparison of the three encoding strategies for a data point with N features each. Here d_i corresponds to the number of qubits used to describe one feature in the basis encoding scheme. 16

4.1. Confusion matrix of an anomaly detection problem. 24

5.1. High level features (HLFs) of the data set. 28

6.1. Best classical classification metrics for the classification metrics with 2 features. 35
6.2. Best classical classification metrics for the classification metrics with 8 features 36
6.3. Best classification score obtained in the classical framework for each dimension between 2 and 8 features. 37
6.4. Classification metrics for the qGAN approach with 2 features on a noiseless simulator and with the best classification metrics. 41
6.5. Classification metrics for the qGAN approach with 2 features on a noisy simulator, compared to the best classical results and to the noiseless quantum simulations. 42
6.6. Classification metrics for the qGAN approach with 2 features on a real hardware. 45
6.7. Classification metrics for the qGAN approach with 8 features on a noiseless simulator compared with the best classical classification metrics. . . 48
6.8. Classification metrics for the qGAN approach with 8 features on a noisy simulator compared to the noiseless quantum simulation and to the best classical classification metrics. 50
6.9. Best classification score obtained in the quantum framework on noiseless and noisy simulations and in the classical case for each dimension between 2 and 8 features. 52
6.10. Number of trainable parameters in the quantum framework 53

1. Introduction

The development of quantum computers stimulated quantum application research. One of these application areas corresponds to Quantum Machine Learning (QML), i.e., the intersection of quantum computing and machine learning. QML can, for example, be used to address some of the computational challenges faced in particle physics data processing and analysis for High Energy Physics (HEP) [2].

Particle physics aims to understand the most elementary components of matter and forces. These elementary particles can be detected in controlled environments with particle accelerators such as the Large Hadron Collider (LHC). The Standard Model (SM) of particle physics is the most elaborate description of the basic building blocks of the universe. It unifies three of the four fundamental forces that govern the universe: electromagnetism, the strong force, and the weak force. While it is an extremely successful effective theory, it is known to have certain limitations, such as its incompatibility with the gravitational force. In fact, there exist so-called anomalous events which cannot be described by the SM. The main goal of the current data analysis at LHC is to detect Beyond-Standard-Model (BSM) particles using machine learning techniques. More specifically, anomaly detection in HEP consists of looking for rare collisions that could generate different particles w.r.t the ones prescripted by the SM.

Effectively, the goal of anomaly detection is the classification of anomalous event data compared to SM event data. Intuitively, one would expect that supervised machine learning methods could offer a suitable method to approach this problem. However, supervised learning strongly relies on the availability of sufficient data from all data types that are to be characterized. Since it is not a priori known which data events are anomalous, the required training data cannot be easily provided. Therefore, the training of a supervised machine learning model for distinguishing between anomalous and SM data is difficult and, in fact, not practical. Another potential approach for anomaly detection would be to use clustering. These algorithms often require a certain level of prior knowledge such as the number of clusters given in a training data set. Therefore, recent research [3, 4] focused on using generative models to learn the underlying structure of the complex data. A generative model learns the underlying unknown process which created a given data set. Once learnt, this approximate process could sample more data points from a distribution close to the data distribution. This could help to detect more easily anomalies. One particularly promising approach suggested in [4] introduces an anomaly detection method, called *AnoGAN*, with generative adversarial networks (GAN). The model employs an adversarial training of two classical neural networks: a generator and a discriminator. The discriminator is trained to differentiate between the output of the generator and the real data set, and the generator is trained to generate data which the discriminator classifies as real data. Once the generator and the discriminator are trained, both networks can be used to define an anomaly score for new test events. Replacing one or both of the neural networks in the GAN setting with parameterized quantum circuits leads to a quantum Generative Adversarial Networks (GAN), the Quantum Generative Adversarial Networks (qGAN) [5, 6] model. This approach can be interesting because a quantum generator has the advantage of providing a native model for the physical generation processes which are governed by quantum

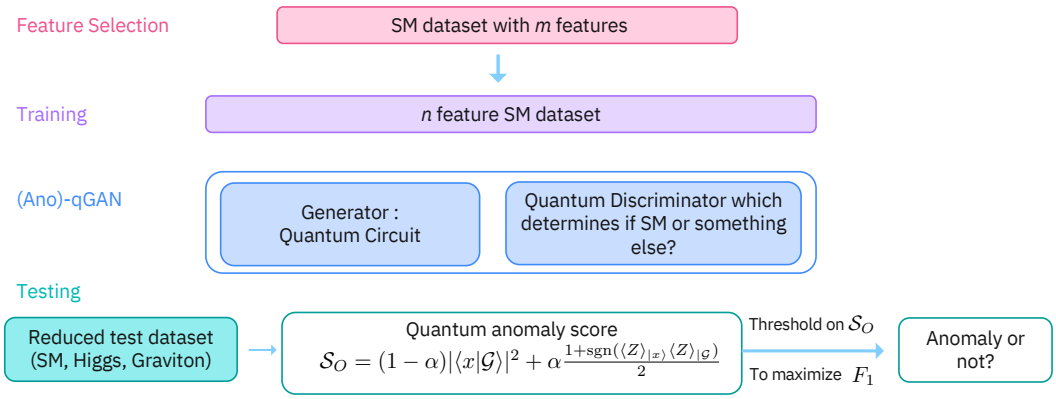


Figure 1.1.: Workflow of the presented anomaly scoring method. We first reduced the data sets through feature reduction techniques which are applied to testing data sets. Then we train our qGAN on the embedded SM data points. Lastly, we compute the anomaly score on the embedded test data set. We use the value of the anomaly score on the SM to define a threshold. This threshold is used to classify a data point as an anomaly.

physics. Hence, we expect that modelling the respective particle physics data with a system that is close to the underlying physics can lead to a beneficial learning behavior.

This work outlines and investigates a qGAN based anomaly score, where a qGAN is trained on an embedded SM data set. Then, the anomaly score for data samples is evaluated by calculating a distance measure between the data samples and states that are given by the trained qGAN. An overview of the method is given in Fig. 1.1. Furthermore, we investigate the practical performance of anomaly detection for Beyond the Standard Model (BSM) data with quantum simulation as well as actual quantum hardware experiments.

This thesis is structured in the following way: we first give an introduction to high energy physics (HEP) (Chapter. 2) and generative models (Chapter. 3). In the latter, we give an overview of classical GAN (Sec. 1 of Chapter. 3) before presenting the quantum framework of GAN (Sec. 2 of Chapter. 3). This is followed by our approach for anomaly detection using GANs (Chapter. 4). First, we introduce the classical approach of the *AnoGAN* technique (Sec. 1.1 of Chapter. 4) and present our approach in the quantum framework (Sec. 1.2 of Chapter. 4). In the quantum approach, the data points of the HEP data sets need to be embedded as presented in Sec. 2.2 of Chapter. 3. This is applied to the data sets presented in Sec. 1 of Chapter 5. These data sets are compressed using a feature selection technique (Sec. 2 of Chapter. 5) and embedded into quantum sets (Sec. 3 of Chapter. 5). We present our result on the HEP data set in the classical framework (Sec. 2 of Chapter. 6) and in the quantum approach (Sec. 3 of Chapter. 6) on noiseless simulations, noisy simulator and in the hardware.

2. High Energy Physics

1. Standard Model of Particle Physics

Particle physics aims at uncovering what the universe is made of. It aims to understand the elementary components of matter and radiation and how they interact with each other [7]. It is also referred to as HEP because many elementary particles do not occur under normal circumstances in nature but can be created and detected during energetic collisions of other particles. These collisions can be produced in controlled environments with particle accelerators such as the LHC in the European Organization for Nuclear Research (CERN) [8] or in uncontrolled environments such as cataclysmic events in the cosmic rays.

First introduced in the early 1970s, the SM of particle physics is the most elaborate description of the basic building blocks of the universe. This model helped determining with high precision known fundamental particles based on their properties and introduces rules that determine which interactions can occur at what rate. The Standard model has been verified experimentally with high precision [9].

The particles in the SM are fermions and bosons whose most fundamental difference is their spin. Bosons are particles with integer spins while half-integer spin particles are fermions. In particular, for particles described by the SM, boson particles only have a spin 0 or 1 while fermions particles have a spin $\frac{1}{2}$.

The SM is a relativistic quantum field theory at the intersection of quantum physics and special relativity. Like in Quantum Electrodynamics (QED), it uses a gauge theory where each interaction between particules is described by the exchange of a boson called a gauge boson. Furthermore, each boson is responsible for mediating a specific force. More specifically, the photons γ mediate the electromagnetic interactions, the $W^{\pm}$ and Z^0-bosons the weak interactions, while the gluons mediating the strong interactions. Plus, a boson can interact with itself if it carries the charges corresponding to the force it mediates (e.g. gluons, $W_{\pm}$, Z^0 bosons).

The Higgs boson is the last discovered element of the SM with the characteristic of having a spin value of 0. The scalar field associated with the Higgs boson explains the masses of other fundamental particles. Therefore, the interaction with the Higgs boson H generates the mass of all particles through the Higgs mechanism [7, 10].

Matter can be described by fermions and by their matching antiparticle with opposite charges. They can be split into two categories: quarks and leptons. Each quark and lepton is then from one of three generations:

- A quark can be either: up u or down d (first generation); charm c and strange s (second generation); top t and bottom b (third generation).

- A lepton can be: electron e and electron neutrino ν_e (first generation) ; muon μ and muon neutrino ν_μ (second generation) ; tau τ and tau neutrino ν_τ (third generation).

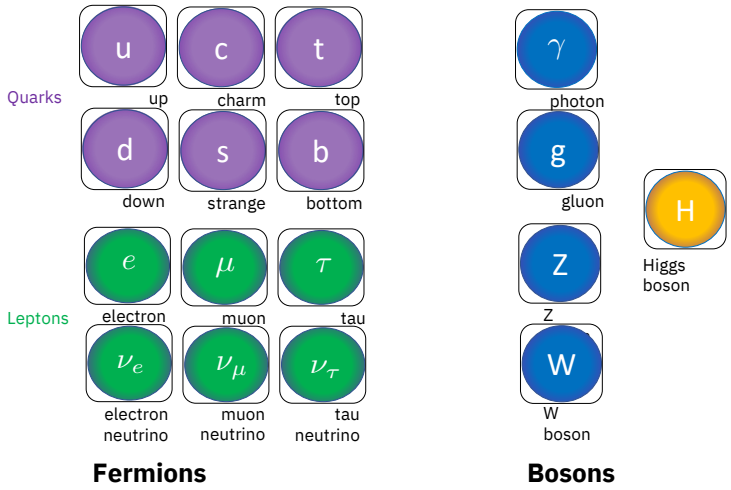


Figure 2.1.: The particle spectrum of the Standard Model.

Particles in different generations have similar properties but usually differ in mass, for example, the u quark is much lighter than the t quark and the mass of the c quark is much greater than the u quark. The main difference between quarks and leptons is that quarks interact with the strong nuclear force, whereas leptons do not. Both quarks and charged leptons can interact via the electromagnetic and weak forces.

2. Beyond the Standard Model

Since its formulation in the 1970s, the SM has predicted the outcome of countless experiments in particle physics with very high accuracy. However, the model still remains incomplete. In fact, gravity and dark matter are the main missing pieces, i.e., they are not described by the SM. Hence, this model leaves many fundamental questions unanswered and even raises some new ones. Some fundamental questions of the universe that cannot be explained by the SM are given below [7, 11, 12]:

- Why has dark matter never been experimentally detected?
- How can we explain the matter/anti-matter asymmetry?
- How can we explain that neutrinos have mass and oscillate from one type to another?
- How can we explain that the SM is incompatible with general relativity?

All of these questions can be answered by the study of theories beyond the SM. These theories are usually called BSM and aim at modifying or extending our picture of the SM.

3. Challenges for future experiments at the LHC

3.1. The LHC

The LHC is the largest and most powerful particle accelerator. First started up on 10 September 2008, the LHC is a 27-kilometer long ring that consists of superconducting magnets with a number of accelerating structures to boost the energy of the particles along the way. It accelerates protons to nearly the velocity of light — in clockwise and anti-clockwise directions — and then collides them at four locations of the ring, corresponding to four different experiments: ALICE [13], ATLAS [14], CMS [15] and LHCb [16].

At first, the LHC project was conceived to explain the mass property of both bosons Z and W and the massless property of the photons. The ATLAS and the Compact Muon Solenoid (CMS) experiments at the LHC are general-purpose detectors located at one of these four collision points (e.g. CMS experiment in Fig. 2.2). Both experiments were designed to search for the Higgs bosons and physics beyond the SM. New physics are expected at the Tera electron-volt (TeV) energy scaled allowed by the LHC. Therefore, these experiments have to be characterized by a high center of mass energy of up to 13 TeV and a luminosity of around $1.2 \times 10^{34}\text{cm}^{-2}\text{s}^{-1}$. This luminosity corresponds to the collision number per area and per second. In practice, the integrated luminosity over time is related to the recorded number of collision events.

In order to build up a picture of events occurring in the LHC, the CMS must find the energies of emerging particles. Electrons and photons are particularly interesting because of their use in finding the Higgs boson (through decay channels involving di-photons for example) and other new physics. Most of the particles produced in the process of the collision of two protons are usually unstable. After several decays, electrons and photons are typical end products of successive particle decays and can easily be detected. These particles are identified using the electromagnetic calorimeter (ECAL). In order to find them with the necessary precision in the very strict conditions of the LHC - a high magnetic field, high levels of radiation and only 50 nanoseconds at most between collisions - it required very particular detector materials. Each element of the detector can help determining one property of the particle, as seen in Fig. 2.2:

- Charged particles are detected in the tracking system.
- Photons or electrons are detected by the ECAL and their light determines the deposited energy.
- Hadrons are detected by the hadronic calorimeter (HCAL) where jets are confined.
- Only the highest-energy hadrons and muon can cross the superconducting solenoid.
- Muons are detected in any muon detector.
- Neutrinos are elusive and can only be detected by computing the imbalance of energy in the the reconstructed event called missing transverse energy (MET).

This entire process is called particle-flow (PF) reconstruction and aims at identifying different parameters of the underlying physical process: momentum of the emitted

particle, pseudorapidity of the emitted particle, number of emitted neutrons, etc. Here, the pseudorapidity η is a space coordinate used to describe the particle's direction with respect to the beam axis. In this thesis, we use data sets from the measurements of the CMS detector.

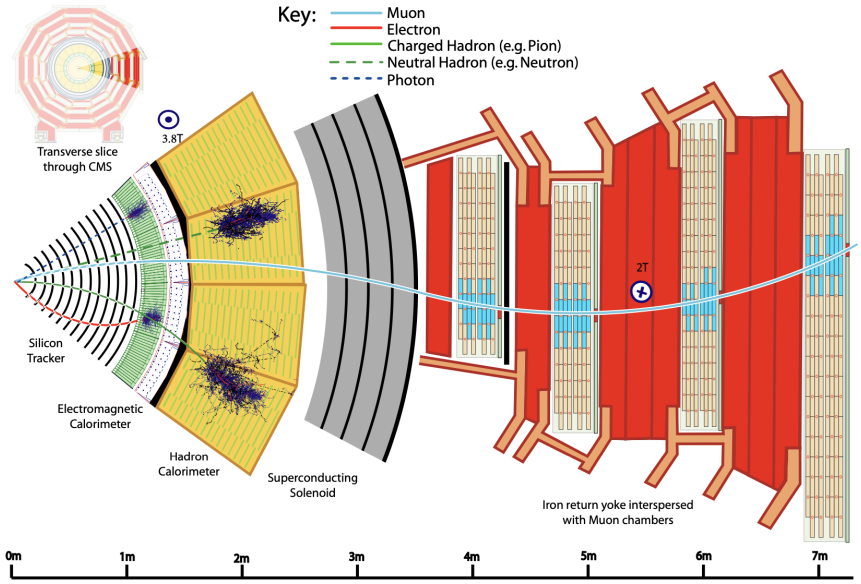


Figure 2.2.: Slice view of the CMS detector [17] and of the paths of different collision products. Charged particles have a curved path due to the electromagnetic field.

3.2. The future run of the LHC

The work from more than 30 institutions has led the first run of the LHC to be successful in the discovery of the Higgs boson confirmed by ATLAS and CMS in 2012. Run 1 (2009-2013) performance for ECAL in particular was the following [18] :

- Electrons and photons were typically well reconstructed up to $\eta < 2.5$.
- Jets were well reconstructed up to $\eta < 3.0$

Thus, the goal for the following runs (called Run 2 and 3 of the LHC) is to maintain (or improve) the ECAL performance that allowed the Higgs boson discovery. However, operating conditions have evolved and are expected to continue evolving with time.

One of the main missions of the LHC is to search for new exciting physics beyond the Standard Model which requires the study of rare events. To witness enough rare events the number of overall events will have to be increased. Therefore, in the future decade, the LHC will be upgraded to the High Luminosity Large Hadron Collider (HL-LHC) to answer its different challenges. This will call for:

- An increased luminosity (e.g. $2 \times 10^{34}\text{cm}^{-2}\text{s}^{-1}$ for Run 2 in 2015-2018). This will also cause an increase in the integrated luminosity: 500 fb^{-1} by the end of 2024. An even bigger increase for the HL-LHC with an instantaneous luminosity

of $7.5 \times 10^{34}\text{cm}^{-2}\text{s}^{-1}$ and an integrated luminosity of 4500fb^{-1} is expected by the end of HL-LHC.

- A higher beam intensity with proton-proton collisions with an energy of 13 TeV in the center of mass.
- Bunches of particles every 25 ns from Run2 onwards (2015-) which means up to 60 simultaneous interactions per bunch crossing. This will lead to more pileup to be dealt with for Run 2 (2015- 2018) and Run 3 (2021-2024), and even more for the High Luminosity LHC (starting in 2027) with up to 200 concurrent interactions per LHC bunch crossing.

Therefore, despite being able to accelerate protons to a high energy of 6 TeV for each proton beam, it operates at the highest collision frequency, producing one proton-beam crossing every 25 nsec. It means that for event, all LHC detectors generates around $\mathcal{O}(1)$ MB of data. Naturally every event could not be efficiently stored because it would require being able to store at a speed of $\mathcal{O}(10)$ TB/sec. The ambition HL-LHC program would require enormous computing resources.

Currently, several techniques are currently used at the LHC to reduce the computation cost of data analysis. It mostly consists of a trigger system [19]: in order to detect a certain type of new physics phenomena, one should specify a signal hypothesis and test of the presence of the predicted kind of events on top of a background from known physics process. Even though this method helps reducing drastically the number of events, it does not help to discover new physics since a hypothesis has to be made prior of the experiment. Therefore, new methods are currently being investigated to improve the particle track reconstruction algorithm, using deep learning [20, 21] or quantum computing techniques [2, 22–24] for example.

3. Generative Adversarial Models

1. Classical GAN

1.1. Definition

In the last decade, the field of machine learning has rapidly evolved due to intense research efforts and has found a wide variety of applications in the industry. Classical machine learning has shown remarkable results at tackling tasks like classification, clustering or regression due to its capacity to scale up efficiently with increasing data set sizes [25]. In particular, the development and study of GAN, which spread out in recent years, has helped to solve many problems of interest, e.g. GAN stem from the field of image generation. It is possible to generate an image from a sentence or phrase that describes its content ("a blue flower"), to generate abstract maps from satellite photos, to predict the next frame in a video and so on. We can also mention their applications in other fields like chemistry [26] or biology [27] where they are used to generate synthetic data.

Another application of generative modelling is data augmentation. In a machine learning procedure, we are given large data sets collected by experiments. The process of the experiments is not always accessible: the input-output of each experiment is the sole knowledge of the data set. However, increasing the data set size is sometimes needed. A tool for approximating the underlying process is given by generative learning, in particular with classical GANs: their goal is to learn the underlying probability distribution of the data set p_{data}. After a generative model is trained, it can be used to sample, i.e., generate new synthetic data that is aligned with the process responsible for the training data. This is applicable with the GAN structure such as with BigGAN [28], StyleGAN [29, 30], or other examples in the literature [25–27].

The GAN working principle [31, 32] consists of two parts, or two players, represented by two differentiable neural networks, the generator ($\mathcal{G}^{\theta_G}$) and the discriminator ($\mathcal{D}^{\theta_D}$), competing with each other as seen in Fig. 3.1. Here, θ_G and θ_D represent the parameters of the generator and discriminator neural networks in this adversarial setting. The generator $\mathcal{G}^{\theta_D}$ takes as input a sample z drawn from fixed prior distribution $p_z(z)$ and aims at generating samples that could be mistaken for being drawn from p_{data}. On the other hand, the discriminator $\mathcal{D}^{\theta_D}$ aims at distinguishing data coming from the generator $\mathcal{G}^{\theta_G}$ and from the training data set. In practice, $\mathcal{D}^{\theta_D}(x)$ represents the probability that x came from the data rather than from the generator.

1.2. Training with classical GAN

Training should improve the performance of both $\mathcal{D}^{\theta_D}$ and $\mathcal{G}^{\theta_G}$ at their respective tasks simultaneously. Different methods exists to train a GAN [33, 34]. Let us first introduce the training method presented in the original work of GAN [31].

The discriminator $\mathcal{D}^{\theta_D}$ is trained in order to maximize the probability of assigning the correct label to both training data points and samples from $\mathcal{G}^{\theta_G}$. The generator $\mathcal{G}^{\theta_G}$ is

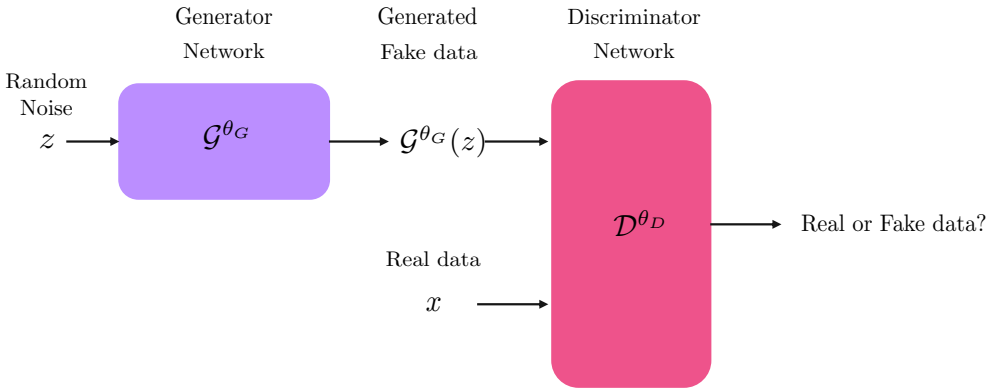


Figure 3.1.: Working principle of the GAN: a generator $\mathcal{G}^{\theta_G}$ samples a data sample, and the discriminator $\mathcal{D}^{\theta_D}$ receives a data sample at a time from either the real data set or from the generated samples and independently classify it. The aim of the training of the GAN is done simultaneously: the generator tries to fool as much as possible the discriminator that its data is the real one while the discriminator tries to distinguish as much as possible the real data from the generated data

trained at the same time in order to minimize the difference between real and fake labels $\log(1-\mathcal{D}^{\theta_D}(\mathcal{G}^{\theta_G}(z)))$, i.e. so that for the discriminator, a generated sample looks like a training data point. In other words, it all comes down to solve a min-max optimization problem with the value function $V(\mathcal{G}^{\theta_G}, \mathcal{D}^{\theta_D})$:

$$\min_{\mathcal{G}^{\theta_G}} \max_{\mathcal{D}^{\theta_D}} V(\mathcal{D}^{\theta_D}, \mathcal{G}^{\theta_G}) = \mathbb{E}_{x\sim p_{data}(x)}[\log(\mathcal{D}^{\theta_D}(x))] + \mathbb{E}_{z\sim p_z(z)}[\log(1-\mathcal{D}^{\theta_D}(\mathcal{G}^{\theta_G}(z))]. \quad (3.1)$$

However, this global min-max optimization is typically not the one being optimized during the training of the GAN. This problem can be reduced to the optimization of two non-saturating loss functions [35] for the generator and for the discriminator respectively.

The discriminator's loss functions reads:

$$\mathrm{C}_D(\theta_G, \theta_D) = -\frac{1}{2}\mathbb{E}_{z\sim p_z(z)} \log(1-\mathcal{D}^{\theta_D}(\mathcal{G}^{\theta_G}(z))) - \frac{1}{2}\mathbb{E}_{x\sim p_{\text{data}}} \log \mathcal{D}^{\theta_D}(x). \quad (3.2)$$

During the training, we do not present training data to the network one item at the time, but the full training data is divided into so-called batches, a group of training data that is fed into the network together. The larger the batch is, the better the training is. However, it brings a higher computational cost. The random partitioning of the training data into batches is kept for a certain number of iterations, before a new partitioning is

chosen.

When sampling a batch of m noisy samples $\{z^{(1)}, \dots, z^{(m)}\}$ drawn from the noise prior $p_z(z)$ and of m training data points $\{x^{(1)}, \dots, x^{(m)}\}$ drawn from the data distribution p_{data}, it yields :

$$\mathrm{C}_D(\theta_G, \theta_D) = -\frac{1}{2m}\sum_{j=1}^{m} \log(1 - D^{\theta_D}(\mathcal{G}^{\theta_G}(z^{(j)}))) - \frac{1}{2m}\sum_{j=1}^{m} \log \mathcal{D}^{\theta_D}(x^{(j)}). \tag{3.3}$$

This corresponds to a minimization of the cross-entropy cost when training a binary classifier with a sigmoid output. During this optimization, only the parameters θ_D are optimized while the parameters θ_G are fixed.

Several techniques exist to specify the cost function of the generator. In the original paper of GAN [31], the generator cost function simply reads:

$$\mathrm{C}_G(\theta_G, \theta_D) = -\mathrm{C}_D(\theta_G, \theta_D). \tag{3.4}$$

This corresponds to a zero-sum game, or a Nash equilibrium, in which the sum of all player's loss function is zero. Learning in this game would ideally converges to the theoretical minimum of the loss function. However, in practice, this does not converge as easily as in the theoretical framework. Therefore, we use for the heuristically generator loss function:

$$\mathrm{C}_G(\theta_G, \theta_D) = -\frac{1}{2}\mathbb{E}_{z\sim p_z(z)} \log \mathcal{D}^{\theta_D}(\mathcal{G}^{\theta_G}(z)). \tag{3.5}$$

When sampling a batch of m noisy samples $\{z^{(1)}, \dots, z^{(m)}\}$ drawn from the noise prior $p_z(z)$ it yields:

$$\mathrm{C}_G(\theta_G, \theta_D) = -\frac{1}{2m}\sum_{j=1}^{m} \log \mathcal{D}^{\theta_D}(\mathcal{G}^{\theta_G}(z^{(j)})). \tag{3.6}$$

The training of both networks is usually done in an alternating fashion as seen in Algorithm 11. Several optimizers are commonly used for the optimization, like Stochastic Gradient Descent (SGD) [36], ADAM [37] or AMSGRAD [38].

1.3. Training with WGAN

Despite being widely used in the machine learning community, the main problem with classical GANs is that they are hard to train because they require large training sets and are highly susceptible to hyperparameters [39, 40]. It is currently an active topic of research to compensate for this with structural modifications and novel loss function formulations. One quite successful loss function is introduced in [34]. Part of the motivations behind this loss is that the original loss functions from [31] have large variance of gradients which make the model unstable.

To solve this issue, the Wasserstein Generative Adversarial Networks (WGAN) were introduced using the optimisation of the Wasserstein distance which has a smoother

input : Dataset $\mathcal{D} = \{x^1, \ldots, x^M\}$, Prior distribution $p_z(z)$, the number of discriminator iterations per generator iteration k, generator optimizer Opt_{θ_G} and its parameters, discriminator optimizer Opt_{θ_D} and its parameters, batch size m.

for number of training steps **do**
  for k steps **do**
    Sample a batch of m noisy samples $\{z^{(1)}, \ldots, z^{(m)}\}$ drawn from the noise prior $p_z(z)$;
    Sample a batch of m training data points $\{x^{(1)}, \ldots, x^{(m)}\}$ drawn from the data distribution p_{data} ;
    Update the discriminator parameters by optimizing $\text{C}_D(\theta_G, \theta_D)$ w.r.t. θ_D with the optimizer Opt_{θ_D} ;
    $\theta_D \leftarrow \theta_D - \text{Opt}_{\theta_D}(\text{C}_D(\theta_G, \theta_D))$
  end
  Sample a batch of m noisy samples $\{z^{(1)}, \ldots, z^{(m)}\}$ drawn from the noise prior $p_z(z)$;
  Update the generator parameters by optimizing $\text{C}_G(\theta_G, \theta_D)$ w.r.t. θ_G with the optimizer Opt_{θ_G} ;
  $\theta_G \leftarrow \theta_G - \text{Opt}_{\theta_G}(\text{C}_G(\theta_G, \theta_D))$
end

Algorithm 1: GAN training algorithm.

gradient behavior [34].

The Wasserstein distance between the data distribution p_{data} and the generator distribution p_g reads:

$$W(p_{\text{data}}, p_g) = \sup_{f:\|f\|_L \leq 1} [\mathbb{E}_{x \sim p_{\text{data}}}[f(x)] - \mathbb{E}_{x \sim p_g}[f(x)]] \tag{3.7}$$

Here the supremum is taken over all 1-Lipschitz functions. This distance can be approximated using a neural network represented by the discriminator $\mathcal{D}^{\theta_D}$ parametrized by the parameters θ_D. The generator distribution corresponds to the distribution of all $\mathcal{G}^{\theta_G}(z)$ for z drawn from the prior distribution $p_z(z)$. Therefore, the discriminator needs to optimize the Wasserstein distance that can be rewritten as:

$$\sup_{\theta_D} [\mathbb{E}_{x \sim p_{\text{data}}}[\mathcal{D}^{\theta_D}(x)] - \mathbb{E}_{z \sim p_z}[\mathcal{D}^{\theta_D}(\mathcal{G}^{\theta_G}(z))]] \tag{3.8}$$

Here $\mathcal{D}^{\theta_D}$ corresponds to a parametrized family of 1-Lipschitz functions. We can choose a family of parametrized neural networks with a sigmoid activation function in the last layer for example. The discriminator loss function reads:

$$\text{C}_D(\theta_G, \theta_D) = \mathbb{E}_{x \sim p_{\text{data}}}[\mathcal{D}^{\theta_D}(x)] - \mathbb{E}_{z \sim p_z}[\mathcal{D}^{\theta_D}(\mathcal{G}^{\theta_G}(z))] \tag{3.9}$$

Similarly the generator aims at minimizing:

$$\min_{\theta_G} [\mathbb{E}_{x \sim p_{\text{data}}}[\mathcal{D}^{\theta_D}(x)] - \mathbb{E}_{z \sim p_z}[\mathcal{D}^{\theta_D}(\mathcal{G}^{\theta_G}(z))]] = -\min_{\theta_G} \mathbb{E}_{z \sim p_z}[\mathcal{D}^{\theta_D}(\mathcal{G}^{\theta_G}(z))] \tag{3.10}$$

Therefore, the generator loss functions reads:

$$\mathrm{C}_G(\theta_G, \theta_D) = -\mathbb{E}_{z\sim p_z}[\mathcal{D}^{\theta_D}(\mathcal{G}^{\theta_G}(z))]. \tag{3.11}$$

However, the constraint of having a family of 1-Lipschitz functions is hard to maintain throughout the training. Furthermore, as stated in [41], a differentiable function f is 1-Lipschitz if and only if it has gradients with norm at most 1 everywhere. Therefore, to ensure this, a gradient penalty term is added to the discriminator loss function which then reads

$$\begin{aligned}\mathrm{C}_D(\theta_G, \theta_D) = \mathbb{E}_{x\sim p_{\mathrm{data}}}[\mathcal{D}^{\theta_D}(x)] - \mathbb{E}_{z\sim p_z}[\mathcal{D}^{\theta_D}(\mathcal{G}^{\theta_G}(z))] \\ + \lambda \mathbb{E}_{\hat{x}\sim p_{\hat{x}}}[(\|\nabla_{\hat{x}}\mathcal{D}^{\theta_D}(\hat{x})\| - 1)^2]\end{aligned} \tag{3.12}$$

Here λ represents the strength of the gradient penalty and $p_{\hat{x}}(\hat{x})$ is the probability distribution of $\hat{x} = x + t(\mathcal{G}^{\theta_G}(z) - x)$ where $x \sim p_{\mathrm{data}}$, $z \sim p_z$ and $t \sim U(0, 1)$.

2. Quantum GAN

2.1. Related work

The data generation process can be interpreted as a black box. Learning the structure of the data with a generative model then relates to approximating the process in the generation black box. In HEP, the measurement of the pp collision at the LHC corresponds to a measurement of a process governed by quantum physics. For example, if a pp collision ends up in a certain detector response, the generative model is able to recreate the event accountable for the detected particles. Since the underlying process is quantum, we expect (or can assume) that modelling the process can be done with a quantum ansatz generative learning.

Quantum computing corresponds to a computational model that is based on the rules of quantum mechanics. For further information, we recommend reading [42]. As this field develops, the search for potential applications of these big conceptual and technical advances continues. In the hope of solving tasks that are hard on classical devices [43, 44], small noisy devices, called Noisy Intermediate-Scale Quantum (NISQ) are already available to test state-of-the-art quantum algorithms. In particular, many machine learning algorithms have been adapted to quantum devices [45–47]—Quantum Machine Learning (QML). More generally, QML may refer to different methods such as using machine learning (ML) techniques to solve quantum problems or to design classical ML techniques influenced by quantum structures. Even if the current NISQ are still limited in the number of qubits [48], they still could bring a potential advantage when combined with QML [49].

In the QML framework, the GAN is a classical ML algorithm that can be adapted in the quantum framework [5, 50]. Several approaches are possible: the input data, the generator and the discriminator can be chosen to be either classical or quantum and combined into an algorithm. It is still ongoing research to find the optimal configuration

between classical and quantum for a given data set. For instance there exists discrete probability distributions that have been proved to be impossible to learn efficiently with a classical generator, but that can be learned efficiently by a quantum one [51]. This creates a theoretical gap in favour of qGAN.

Fully quantum GAN, made with a quantum generator and a quantum discriminator, is an advantageous structure: the quantum generator can be used to learn an underlying quantum process, and a quantum discriminator only needs to measure a single qubit. This discriminator approach helps reducing measurement errors and avoids using global objective functions which could lead in the vanishing gradient problem, so called, barren plateaus [52, 53]. Recent research are also looking at improving the training of qGAN by putting more weight on data points which contribute more to the data distribution [54].

In recent literature we can find numerous examples of attempts to tackle real life problems with qGAN. In [6] a qGAN algorithm, using a quantum generator and a classical discriminator to generate classical distributions and implemented it with numerical simulations as well as on a IBM quantum hardware with 3 qubits. Another work used a qGAN algorithm to learn a continuous distribution for data that aims to simulate HEP detectors, using both hybrid and fully quantum setups [55]. In HEP, qGANs can be combined with a Quantum Amplitude Estimation algorithm [56] for integrating elementary-particle cross sections, using both Monte-Carlo methods and an estimation of the underlying probability distribution of a HEP data set [57]. For image processing, we can cite 2 relevant experiments that aimed at generating the famous MNIST handwritten digit dataset: one which used a hybrid qGAN with 3 to 5 qubits on a superconducting processor, but but showed that qGAN are resilient to a certain level of noise sources contained in a quantum devices [58], and another which used an 8 qubit ion trap processor and obtained high-resolution digits, still using a hybrid setup [59]. However, despite all of the different progress on qGAN, a lot of research is still needed to understand where and why it could outperform classical models.

2.2. Data embedding

In this thesis, we study measurement data of quantum experiments–which is given in a classical form. If we now want to study this data with a quantum framework, one has to design an efficient way to represent the respective classical features with a quantum state. Quantum machine learning algorithms can not ignore how the embedding of a classical memory into the quantum computer is performed. In practice, the procedure of encoding data into the quantum system is part of the algorithm and may account for a crucial part of the complexity. For example, preparing a generic n-qubit quantum state can in the worst case need a exponential number of gates $\mathcal{O}(2^n)$ with $N = \log_2 n$ features [60]–which would impair any possible advantage of a quantum algorithm. Therefore, a quantum encoding is called efficient if its runtime is logarithmic or linear.

Let us consider the data set as $\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_M\}$ where each $\mathbf{x}_i$ consists of a continuous n-feature data point. In practice, each data point is encoded via a state preparation circuit using only one- and two-qubit gates. It maps the data set $\mathcal{D}$ to an n-qubit state

in the Hilbert space $\mathcal{H}$ and reads:

$$\begin{aligned} E: \mathcal{D} &\to \mathcal{H} \\ \mathbf{x} &\mapsto E(\mathbf{x}). \end{aligned}$$

Each encoding needs to have different properties:

- E should have as little as possible qubits but as many as necessary ;
- E should have a number of gates polynomial in the number of qubits to minimize the width and the depth of the quantum circuit ;
- E should be bijective: each data point $\mathbf{x}$ should be represented by a unique quantum state $|\mathbf{x}\rangle$.

In this thesis, we present the basis encoding [61], the angular encoding [62–65] and the amplitude encoding [64–66]. Other embedding exist in the literature [67].

Basis encoding Basis encoding is primarily used for arithmetic operations in a quantum algorithm. One can transform real numbers into their binary encoding and then map them onto a computational basis state of a quantum state. For a continuous data point, its value is approximated by its binary representation.

In other words, each feature $j = 1, \ldots, n$ of the data point $\mathbf{x}_i$ is represented by a quantum state in $\{|0\rangle, \ldots, |2^{d_j} - 1\rangle\}$. Here d_j corresponds to the number of qubits used to describe the corresponding feature. It means that each feature of each data point $\mathbf{x}_i$ is discretized and the binary representation of the discretized version $\tilde{\mathbf{x}}_i$ in d_i bits uses d_i qubits.

For example, in a data set with 2 features x_1 and x_2 with $x_1 \in [2.2, 5.3]$ and $x_2 \in [6.1, 10]$, we can choose to represent the data point $\mathbf{x} = (4.54, 6.5)$ with 2 qubits for the first feature and 3 qubits for the second one. In that case, the domain of the first feature is now $\{2.2, 3.2, 4.2, 5.2\}$ and for the second one it is $\{6.1, 6.58, 7.07, 7.56, 8.04, 8.54, 9.02, 9.51\}$. Therefore, $\mathbf{x}$ is mapped to $\tilde{\mathbf{x}} = (4.2, 6.58)$ with the binary representation 10 and 001. In the end, $\mathbf{x}$ is embedded with the quantum state $|00110\rangle$.
Thus, each data point needs $\sum_{i=1}^{n} d_i$ qubits. While it is simple to understand and requires preparation circuits of depth $\mathcal{O}(1)$, basis encoded data sets requires to discretize the data set which means that for NISQ computers, features taking a lot of different values cannot be efficiently embedded into a quantum state.

Angle encoding [63–65] Due to the NISQ limitations, the angle encoding is the most common and the simplest form of encoding classical data into a quantum state. It uses n qubits to encode an n feature data point as

$$|x\rangle = \bigotimes_{i=1}^{n} R(x_i) \, |0\rangle_i \, . \tag{3.13}$$

Here, $|j\rangle_i$ corresponds to the state $j = 0, 1$ acting on qubit i. R corresponds to one of the single-qubit rotations R_x, R_y or R_z. For this thesis, we choose the angle encoding

with R_y gates. For example the data point $\mathbf{x} = (\pi, \pi, \pi)^T$ is embedded in the quantum state $|\mathbf{x}\rangle = |111\rangle$. The main advantage of this method is that it only needs n qubits and one rotation on each qubit, making it amenable for NISQ computers. The main drawback of this approach is that this encoding does not ensure to have a one-to-one encoding. One must note that even if this embedding is optimal for continuous input features, due to the periodicity of the R_y encoding, the input data value has to be scaled in an interval in $[0, 2\pi]$.

Amplitude encoding This embedding encodes data into the amplitudes of a quantum state. It represents a normalised classical n-dimensional data point, x, as the amplitudes of a $\log_2(n)$ qubit quantum state. Thus, a data point $\mathbf{x} = \{x_1, \ldots, x_n\}$ is represented by

$$|x\rangle = \frac{1}{\|x\|} \sum_{i=1}^{n} x_i |i\rangle . \tag{3.14}$$

Here $\|x\|^2 = \sum_{i=1}^{n} |x_i|^2$. For example, the data point $\mathbf{x} = (1, 1, -1, -1)$ is embedded into the quantum state $|\mathbf{x}\rangle = \frac{1}{2}|00\rangle + \frac{1}{2}|01\rangle - \frac{1}{2}|10\rangle - \frac{1}{2}|11\rangle$. As a system of n qubits provides 2^n amplitudes, amplitude embedding requires $\log_2(n)$ qubits. The main disadvantage of this method is that this encoding requires in general a circuit depth $\mathcal{O}(2^n)$ and can be very hard to construct.

The different strategies are summed up in Table 3.1.

Encoding	Number of qubits	Input features
Basis	$\sum_{i=1}^{n} d_i$	Binary
Angular	n	Continuous
Amplitude	$\log n$	Continuous

Table 3.1.: Comparison of the three encoding strategies for a data point with N features each. Here d_i corresponds to the number of qubits used to describe one feature in the basis encoding scheme.

2.3. Definition

In this thesis, we focus on the training of a fully quantum GAN with a classical input data set given by a quantum process, a quantum generator and a quantum discriminator.

We have to represent the classical input data set with a quantum data set $\{|\mathbf{x}_1\rangle, \ldots, |\mathbf{x}_M\rangle\}$. In the remainder of this work, we shall refer to this data set as the *training quantum data set*.

The generator is a parametrized quantum channel and is trained to transform the basis state $|0\rangle^{\otimes n}$ to an n-qubit output state :

$$|\mathcal{G}\rangle = \mathcal{G}(\theta_G)\,|0\rangle^{\otimes n} . \tag{3.15}$$

The quantum circuit $\mathcal{G}(\theta_G)$ consists of an initialization layer and alternating layers of parametrized single-qubit-rotations and blocks of two-qubit-entanglement gates. In this thesis, we used Pauli-Y gates for the qubit rotations and controlled-Z-gates for the

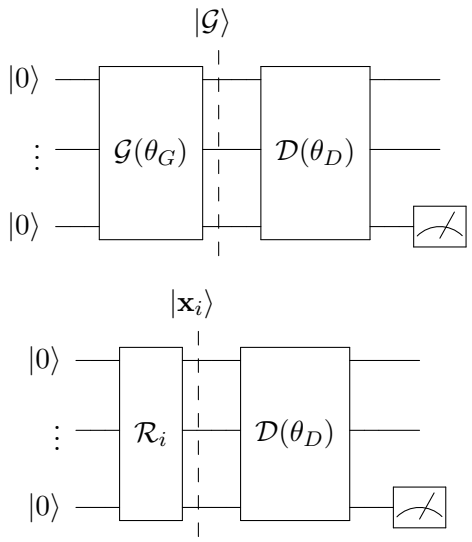


Figure 3.2.: General qGAN to learn the training quantum data set $\{|\mathbf{x}_1\rangle, \ldots, |\mathbf{x}_M\rangle\}$. At each training step, the discriminator receives the generated state $|\mathcal{G}\rangle$ and a batch of the embedded data points $|\mathbf{x}_i\rangle = \mathcal{R}_i |0\rangle$.

entanglement gates [68, 69]. The initialization layer consists of Hadamard gates H [42] on all qubits. In total, the circuit has $(k+1)n$ parametrized Pauli-Y rotation and kn controlled-Z gates. The quantum circuit of the generator is shown in Fig. 3.3.

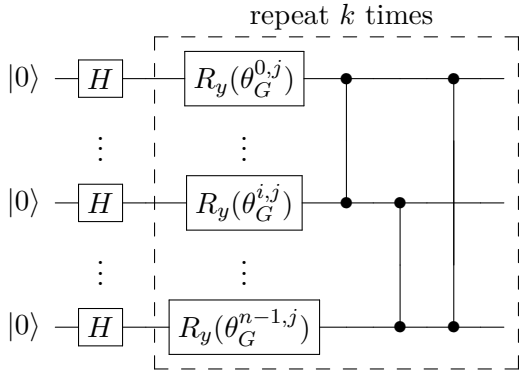


Figure 3.3.: Ansatz for the generator $\mathcal{G}(\theta_G)$. $\theta_G^{i,j}$ corresponds to the parameter of R_y rotation acting on the i qubit on the j layer.

The discriminator is also a parametrized quantum circuit. The variational form of the discriminator consists of parametrized single-qubit rotations, Pauli-X (R_X), Pauli-Y(R_Y) and Pauli-Z(R_Z) rotations on all qubits followed by controlled-X gates (CX) [42] with the target qubit on the last qubit. Finally, the three single-qubit rotations are applied on the last qubit. The discriminator applies the variational form to a n-qubit input quantum state, and labels it as real or fake. This classification is made by a measurement of the last qubit in the computational basis, i.e., in the qubit ordering convention, w.r.t. a $Z \otimes \mathbb{1} \otimes \ldots \otimes \mathbb{1}$ observable, with Z the single qubit Pauli-Z. An input state is classified as real or fake if the measurement outcome is -1 or 1, respectively. In total, the circuit has $3(n+1)$ parametrized single-qubit rotations. The quantum circuit of

the discriminator is shown in Fig. 3.4. Both generator and discriminator are represented

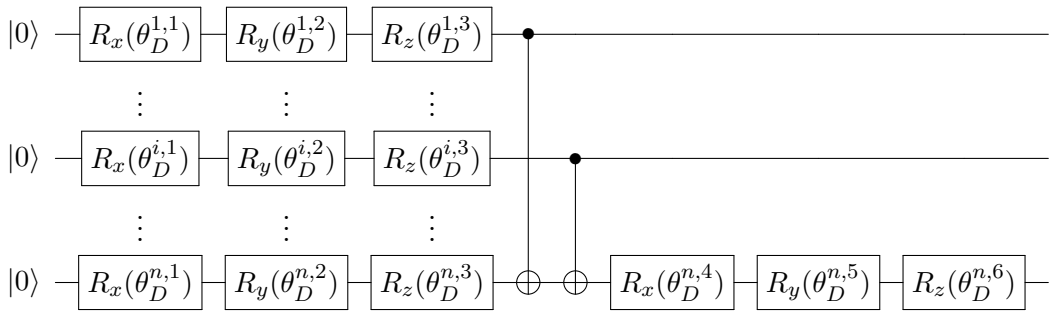


Figure 3.4.: Ansatz for the discriminator $\mathcal{D}(\theta_D)$. $\theta_D^{i,j}$ corresponds to the j parameter of R_y rotation acting on the i qubit.

by hardware-efficient parametrized quantum circuits [68, 70], $\mathcal{G}(\theta_G)$ and $\mathcal{D}(\theta_D)$, as shown in Fig. 3.2.

2.4. Training

At every training step, as in the classical case, the generator aims at fooling the discriminator while the discriminator is trying to distinguish between training and generated data. After convergence, the generator is supposedly able to generate quantum states indistinguishable from the training quantum data set $\{|\mathbf{x}_1\rangle, \ldots, |\mathbf{x}_M\rangle\}$.

To simplify the notation, we now introduce helper functions $\mathrm{C}_D^{\mathcal{S}}$ where $\mathcal{S} \in \{\mathcal{G}; \mathcal{R}\}$

$$\mathrm{C}_D^{\mathcal{G}}(\theta_G, \theta_D) = \frac{1 - \langle \mathcal{G}|\mathcal{D}^\dagger(\theta_D)\,(Z \otimes \mathbb{1} \otimes \ldots \otimes \mathbb{1})\,\mathcal{D}(\theta_D)|\mathcal{G}\rangle}{2} \tag{3.16}$$

$$\mathrm{C}_D^{\mathcal{R}}(\theta_D) = \frac{1}{2}[1 - \frac{1}{M}\sum_{j=1}^{M} \langle \mathbf{x}_i|\mathcal{D}^\dagger(\theta_D)\,(Z \otimes \mathbb{1} \otimes \ldots \otimes \mathbb{1})\,\mathcal{D}(\theta_D)|\mathbf{x}_i\rangle] \tag{3.17}$$

with $|\mathcal{G}\rangle$ denoting the generator state, each $|\mathbf{x}_i\rangle$ denoting each training data point and M denoting the total number of data points. The respective functions determine the probability $\mathbb{P}$ of the discriminator measurement corresponding to -1.

Firstly, the generator $\mathcal{G}(\theta_G)$ intends to generate a state $|\mathcal{G}\rangle$ that reproduces the desired quantum data set $\{|\mathbf{x}_1\rangle, \ldots, |\mathbf{x}_M\rangle\}$. The loss function for the generator reads:

$$\mathrm{C}_G(\theta_G, \theta_D) = -\mathrm{C}_D^{\mathcal{G}}(\theta_G, \theta_D) \tag{3.18}$$

with $\mathrm{C}_D^{\mathcal{G}}$ given by Eq. 3.16. The loss function aims at maximizing the probability for the generated state $|\mathcal{G}\rangle$ being labeled as real by the discriminator.

Secondly, the discriminator $\mathcal{D}(\theta_D)$ tries to discriminate the generator $|\mathcal{G}\rangle$ output from the quantum data set $\{|\mathbf{x}_1\rangle, \ldots, |\mathbf{x}_M\rangle\}$. The loss function of the discriminator reads:

$$\mathrm{C}_D(\theta_G, \theta_D) = \mathrm{C}_D^{\mathcal{R}}(\theta_D) - C_D^{\mathcal{G}}(\theta_G, \theta_D) \tag{3.19}$$

with $C_D^{\mathcal{R}}$ being again given by Eq. 3.16. In other words, the goal is to maximize the probability that each quantum state $|\mathbf{x}_i\rangle$ is labeled as real by the discriminator. Ideally, the generator is learning the state

$$|\mathcal{G}\rangle = \sum_{j=1}^{M} \sqrt{p_j}\,|\mathbf{x}_j\rangle \tag{3.20}$$

with each p_j denoting the sampling probability for the data point $|\mathbf{x}_j\rangle$

Both loss functions of Eqs. 3.18 and 3.19 are optimized in an alternating fashion by computing their analytical gradient [71] offered by the parameter shift rule [72, 73]. In practice, at each optimization step, the discriminator is trained with single samples of the true data, compared to the generator which is trained with the full generator state. Therefore, to improve the stability of the algorithm, this alternating training is done with more optimization iterations for the discriminator. The entire training is shown in Algorithm 2.

input : Training quantum data set $\{|\mathbf{x}_1\rangle, \ldots, |\mathbf{x}_M\rangle\}$, batch size m, the number of discriminator iterations per generator iteration k, generator optimizer Opt_G and its parameters, discriminator optimizer Opt_D and its parameters.

for number of training steps **do**
  for k steps **do**
    Compute $C_D^{\mathcal{G}}(\theta_G, \theta_D) = \frac{1-\langle \mathcal{G}|\mathcal{D}^\dagger(\theta_D)(Z\otimes\mathbb{1}\otimes\ldots\otimes\mathbb{1})\mathcal{D}(\theta_D)|\mathcal{G}\rangle}{2}$;
    Sample a batch of m data points $\{|\mathbf{x}_i\rangle\}_{i=1}^m$.;
    Compute $C_D^{\mathcal{R}}(\theta_D) = \frac{1}{2}[1 - \frac{1}{m}\sum_{j=1}^m \langle \mathbf{x}_i|\mathcal{D}^\dagger(\theta_D)\,(Z\otimes\mathbb{1}\otimes\ldots\otimes\mathbb{1})\,\mathcal{D}(\theta_D)|\mathbf{x}_i\rangle]$;
    Compute the analytical gradients $\nabla_{\theta_D} C_D^{\mathcal{S}}(\theta_G, \theta_D)$ with $\mathcal{S} \in \{\mathcal{R}, \mathcal{G}\}$;
    Update the discriminator parameters by optimizing $C_D(\theta_G, \theta_D)$ w.r.t. θ_D with the gradient-based optimizer Opt_D and analytical gradients ;
    $\theta_D \leftarrow \theta_D - \mathrm{Opt}_D(C_D^{\mathcal{R}}(\theta_D) - C_D^{\mathcal{G}}(\theta_G, \theta_D))$
  end
  Compute $C_D^{\mathcal{G}}(\theta_G, \theta_D) = \frac{1-\langle 0|\mathcal{G}(\theta_G)\mathcal{D}^\dagger(\theta_D)(Z\otimes\mathbb{1}\otimes\ldots\otimes\mathbb{1})\mathcal{D}(\theta_D)\mathcal{G}(\theta_G)|0\rangle}{2}$;
  Compute the analytical gradients $\nabla_{\theta_G} C_D^{\mathcal{G}}(\theta_G, \theta_D)$;
  Update the generator parameters by optimizing $C_G(\theta_G, \theta_D)$ w.r.t. θ_G with the corresponding gradient-based optimizer Opt_G and analytical gradients ;
  $\theta_G \leftarrow \theta_G - \mathrm{Opt}_G(C_G^{\mathcal{G}}(\theta_G, \theta_D))$
end

Algorithm 2: qGAN training algorithm.

4. Anomaly detection with Generative Adversarial Networks

In spite of the existence of many theoretically motivated BSM models, like supersymmetry [74] or loop quantum gravity [75], there was no significant evidence of new physical signals since the last particle of the SM has been discovered in 2012 [76]. The recent developments in machine learning lead to new techniques to detect deviations from the SM [77–81]. The absence of knowledge outside the SM framework motivated the use of unsupervised learning techniques to study and find rare events outside the spectrum of the SM [82–84]. In this chapter, we introduce a method for anomaly detection that employs (quantum) GANs. More explicitly, we outline how to compute an anomaly score based on a (quantum) GAN that is trained on SM data.

1. Related work

In data science, an anomaly corresponds to a data point that significantly deviates from the rest of the data set [85, 86]. Anomaly detection, also known as outlier detection, corresponds to the identification of these outliers. Being able to detect these rare data points is particularly useful in various domains from wireless sensor networks [87], fraud detection, data cleaning, detection of cancer [88], or intrusion attacks in cybersecurity [89].

1.1. Anomaly detection with classical unsupervised learning

Unsupervised methods for anomaly detection try to identify events that do not look like background events without using prior knowledge. One method is to learn the probability distribution of the whole background and to look for events with a very low probability. However, learning these high-dimensional densities is a hard task to accomplish. Thus, to avoid this difficulty, practical anomaly detection rather relies on designing a score to determine whether a particular event is uncharacteristic of a given background distribution. In practice, deep learning methods use a compressed latent space to efficiently learn background events with variational autoencoders [90] or generative adversarial networks [4]. The latent space is an efficient representation of the input data, which captures all important structures, such that the event can be fully reconstructed with the decoder. If the decoded data point does not resemble the input data, then it is identified as an anomaly.

A GAN framework, called *AnoGAN*, introduced in [4] uses the adversarial nature of the two networks to flag anomalous data points. Firstly the adversarial networks learn the structure of the non-anomalous data set. Then, an anomaly score is calculated on the testing data set and/or on new data points to determine if each new data point fit into the learned distribution. This is done by finding the optimal latent variable minimizing the anomaly score for each data point. Depending on the value of this metric, the respective event is labeled as anomalous or normal. Different variations of this approach

are possible, i.e., a WGAN with gradient penalty can be used instead of a standard GAN as in [91]. However, this approach requires an optimization for each testing data point to recover the optimal z_{opt} using stochastic gradient descent for example. This procedure is computationally expensive as every gradient computation requires backpropagation through the generator network. To speed up the anomaly detection scheme, the generator is usually coupled with a variational autoencoder [92] or also with a bi-directional GAN, which is then called the BiGAN model [93, 94].

1.2. Anomaly detection with qGANs

The anomaly detection method introduced above may also be combined with quantum resources, e.g. in the form of a qGAN.

Depending on the nature of the data, different approaches are possible. If the process generating the training data is governed by quantum mechanics, i.e. the data is generated by a quantum channel $\mathcal{G}^*$, then the use of a qGAN can enable the learning of a model for $\mathcal{G}^*$ with a parameterized quantum circuit $\mathcal{G}$ as generator.

In [95], the authors adapt the classical *AnoGAN* approach to the quantum framework and apply their method for fraud detection. The discriminator is a classical neural network while the generator consists of a parameterized quantum circuit with a classical neural net. The quantum circuit operates on a low-dimensional latent space while the classical neural net scales expectation values obtained from this circuit to the higher-dimensional data space. The AnoGAN is then trained on a non-anomalous data set and for each new data point, an optimal latent variable is found to define its anomaly score. Contrary to this approach, in this thesis, we use a fully quantum GAN model, i.e. with a quantum generator and a quantum discriminator. We choose this approach because:

- a quantum generator has the advantage of providing a native model for the physical process of HEP.
- a quantum discriminator has the advantage of measuring a single qubit. It helps to reduce the measurement errors and to avoid to have a global objective function which could suffer from barren plateaus [52, 53].

2. Anomaly score

2.1. Classical anomaly score

In the ideal case, the trained quantum discriminator could directly enable anomaly detection by classifying anomalous data as fake data. In practice, however, it is unlikely that the discriminator is trained perfectly. More explicitly, the adversarial training of both the generator and the discriminator can often lead to unstable learning behavior. As in the *AnoGAN* method [4], for the classical framework, we use a anomaly loss function based on a combination of the influence of the generator and of the discriminator to compensate for instabilities in the generator/discriminator training, i.e.,

$$\mathcal{L}_O(z) = (1-\alpha)\|x - \mathcal{G}(z)\| + \alpha\|\mathcal{D}(x) - \mathcal{D}(\mathcal{G}(z))\|. \tag{4.1}$$

The first term is a *residual loss* and computes the similarity between the generated event $\mathcal{G}(z)$ and the testing data point x. The second term is a *discrimination loss* and

measures the distance between the generated event $\mathcal{G}(z)$ and the testing data point x at the output of the discriminator.

The classical GAN is first trained on the non-anomalous data set as described in Section 3. Then, the trained generator and discriminator are used to find the optimal z_{opt} minimizing $\mathcal{L}_O$. The classical anomaly score of the AnoGAN approach reads as:

$$\mathcal{S}_O = \min_z \mathcal{L}_O(z) = (1-\alpha)\|x - \mathcal{G}(z_{\text{opt}})\| + \alpha\|\mathcal{D}(x) - \mathcal{D}(\mathcal{G}(z_{\text{opt}}))\|. \tag{4.2}$$

For example, for a non-anomalous data point x, we have $x = \mathcal{G}(z_{\text{opt}})$ for a perfect generator and we have $\mathcal{D}(x) = \mathcal{D}(\mathcal{G}(z_{\text{opt}}))$ for a perfect discriminator. If this is not the case, the parameter α weights the importance of both term.

2.2. Quantum anomaly score

Contrary to the approach of [95], the generative model of this thesis is a fully quantum GAN setup as described in section 2 of chapter 3.

Firstly, the qGAN learns the quantum representation of the data set $\{|\mathbf{x}_1\rangle, \ldots, |\mathbf{x}_M\rangle\}$ in the routine described in Algorithm 2. Then, every data point x of the test data set is embedded into a quantum state $|x\rangle$ and the anomaly score for this fully quantum framework reads:

$$\mathcal{S}_O = (1-\alpha)|\langle x|\mathcal{G}\rangle|^2 + \alpha\frac{1+\text{sgn}(\langle Z\rangle_{|x\rangle}\langle Z\rangle_{|\mathcal{G}\rangle})}{2}. \tag{4.3}$$

Like in the classical case, the first term is a *residual score*. It tells us how close the new embedded data sample $|x\rangle$ is to the trained generator state $|\mathcal{G}\rangle$. For a non-anomalous data point x, a perfect generator yields $\langle x|\mathcal{G}\rangle = 1$.

The second term is a *discriminator score*. It represents the influence of the discriminator $\langle Z\rangle_{|x\rangle}$ and is computed as follows: we load the state $|x\rangle$ into the discriminator and measure the last qubit of $\mathcal{D}$. Then the discriminator score is 1 if the Z-measurement w.r.t. a Pauli-Z observable corresponds to the same eigenvalue and 0 otherwise. For a non-anomalous data point $|x\rangle$, a perfect discriminator yields $\frac{1+\text{sgn}(\langle Z\rangle_{|x\rangle}\langle Z\rangle_{|\mathcal{G}\rangle}}{2} = 1$

Finally, the parameter α weights the importance of both networks for the anomaly score.

3. Classification

3.1. Metrics

The performance of an anomaly detection problem is usually measured with different metrics, e.g. Receiver Operator Characteristics (ROC) curve, Accuracy, F1 score, Recall and precision.

This problem corresponds to a binary classification problem : the normal class is labeled 0 and the anomaly class is labeled 1. We first define the confusion matrix for our problem as in Table 4.1.

	Real Anomaly	**Real Non-Anomalous**
Predicted Anomaly	True Positive	False Positive
Predicted Non-Anomalous	False Negative	True Negative

Table 4.1.: Confusion matrix of an anomaly detection problem.

True positive corresponds to the number of the detected anomalies that are correctly predicted by the classifier while false positive corresponds to the number of flagged data points that are actually not anomalies. Similarly, false negative corresponds to the number of detected non-anomalous data points that were anomalies while true negative corresponds to the detected non-anomalous data points that were non-anomalous. The true positive rate (TPR) and the false positive rate (FPR) then read :

$$\text{TPR} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}, \text{FPR} = \frac{\text{False Positive}}{\text{False Positive} + \text{True Negative}}. \quad (4.4)$$

Accuracy Score is the most commonly used metrics in an anomaly detection problem. It corresponds to the proportion of correct predictions and reads

$$\text{Accuracy} = \frac{\text{Correct predictions}}{\text{Total predictions}} \quad (4.5)$$

$$= \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive} + \text{False Negative} + \text{True Negative}}. \quad (4.6)$$

However, the accuracy score can not be the only metrics to measure the quality of a ML problem, especially with imbalanced labels or for which the desired weighting of the labels is unknown [96] as in an anomaly detection problem.

ROC Curve It corresponds to the plot of the true positive rate versus the false positive rate. The ROC curve essentially measures the performance of a binary classifier as its cutoff threshold is varied. In practice, the area under the ROC curve, the Area Under Curve (AUC), is a widely used measure of the potential predictive value of a feature [97, 98].

Recall corresponds to the proportion of true anomalies that was correctly identified. It reads

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}. \quad (4.7)$$

Precision corresponds to the proportion of positive identifications that was actually correct. It reads

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}. \quad (4.8)$$

F1 Score Lastly, the F1-score measures the global performance of the anomaly detection. It corresponds to the harmonic mean of the precision and of the recall. It reads

$$F1 = \frac{2}{\text{Recall}^{-1} + \text{Precision}^{-1}}. \tag{4.9}$$

In all of these metrics, the closer the metrics is to 1, the better the anomaly detection scheme is.

3.2. Anomaly detection algorithm

Both the classical and quantum anomaly detection scheme are similar. The anomaly score is computed for each single point of the training data set and for each single point of the testing data set. The mean of the anomaly scores for the data sets define thresholds which in turn help to identify whether testing data points correspond to anomalies or not. A grid search for the best α parameter can help to improve the anomaly detection by maximizing the AUC.

The difference between the classical and the quantum approach lies in the expected behavior of the anomaly score for anomalies and for non-anomalous data points:

- in the classical case, the anomaly score is expected to be as close as possible to 0 for the non-anomalous data points and way bigger for anomalies ;
- in the quantum case, the anomaly score is expected to be as close as possible to 1 for the non-anomalous data points and as close as possible to 0 for anomalies.

However, in practice, we are most importantly interested in finding a hyperplane separating the non-anomalous data points and the anomalous data points. The anomaly detection procedure is explained in Algorithm 3.

input : Training data set $\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_M\}$ (or $\{|\mathbf{x}_1\rangle, \ldots, |\mathbf{x}_M\rangle\}$ if quantum GAN), testing data set $\mathcal{T}$ the number of discriminator iterations per generator iteration k, generator optimizer Opt_G and its parameters, discriminator optimizer Opt_D and its parameters.

for number of training steps **do**
  The generative structure $(\mathcal{G}, \mathcal{D})$ is trained based of Algorithm. 11 or 2.
end
for $\alpha \in [0; 1]$ **do**
  for $x \in \mathcal{D} \cup \mathcal{T}$ **do**
    if classical **then**
      Compute $\mathcal{S}_O(x) = \min_z[(1-\alpha)\|x - \mathcal{G}(z)\| + \alpha\|\mathcal{D}(x) - \mathcal{D}(\mathcal{G}(z))\|$ by backpropagation ;
    else
      Compute $\mathcal{S}_O = (1-\alpha)|\langle x|\mathcal{G}\rangle|^2 + \alpha\dfrac{1 + \text{sgn}(\langle Z\rangle_{|x\rangle}\langle Z\rangle_{|\mathcal{G}\rangle})}{2}$.
    end
  end
  Compute ROC Curve and AUC(α) score ;
  Compute $F1(\alpha)$, Accuracy(α), Recall(α) and Precision(α) score by varying a cutoff threshold.
end
Compute $\text{AUC}(\alpha_{\max}) = \max_\alpha \text{AUC}(\alpha)$ Return classification score $F1(\alpha_{\max})$, Accuracy($\alpha_{\max}$), Recall($\alpha_{\max}$) and Precision($\alpha_{\max}$)

Algorithm 3: Anomaly detection algorithm.

5. Methodology

1. Data sets

The training data is given by a data set representing a weighted mixture of different SM processes typically observed at 13 TeV with weights given by the production cross section of the corresponding process [99]. This accounts for the most representative processes in the SM. In this thesis, every SM process is the result of simulations made in [99] and do not include the Higgs boson. Therefore, this thesis treats the Higgs boson as an anomaly from the SM.

More specifically, the considered SM event can be generated with a Monte Carlo simulator for a specific physical process that is based on a specific either SM or a BSM theory. It generates the necessary inputs to a detector simulation program with a software such as PYTHIA8 [100]. The particles are then tracked in the detector materials by studying their interaction with material. Another simulation framework such as DELPHES3 [101] creates the detector response and produces the ideal response of the detector. The program is only set up to handle collisions either between hadrons, such as proton p, antiproton $\bar{p}$ or between same-generation leptons. That is, proton-proton pp, proton-antiproton $p\bar{p}$ and electron-positron $e + e-$ beam combinations can be used, but not electron-proton ep, photon-proton γp or proton-proton $\gamma\gamma$. In practice, the simulated detector response can be compared with the detector measurements in real experiments conducted at LHC.

However, HEP theories need physics-motivated high level features to interpret the result of collision measurements. Therefore, a feature reconstruction algorithm, called particle-flow reconstruction algorithm [102] is applied. It takes as input the result of the detector response (from a simulation or from an experiment) and aims at reconstructing and identifying all known stable particles (charged particle, photons and neutral hadrons). It reconstructs the measured energy and flight direction for each identified particles and outputs lists of reconstructed electrons, muons and jets. Each event is characterized by a list of 23 physics-motivated high-level features [103].

Events are post-processed by requiring the emission of an electron, muon, or tau lepton with $p_T > 23GeV$. This selection also requires the presence of one reconstructed electron or muon with an isolation requirement Iso < 0.45. The isolation reads

$$\text{Iso} = \frac{\sum_{p \neq q} p_T^p}{p_T^q}. \tag{5.1}$$

The anomalous data sets correspond to the simulation of BSM processes obtained with PYTHIA8 Monte Carlo simulator [100, 101, 104]: the Higgs boson and the graviton data sets are considered.

	Feature name	Definition
0	H_T	The scalar sum of the transverse momenta p_T of all jets having $p_T > 30$ GeV and $\|\eta\| < 2.4$. H_T characterizes hadronic component of event.
1	$p_{T,\|\|}^{miss}$	The parallel component of p_T^{miss} with respect to the lepton.
2	$p_{T,\perp}^{miss}$	The orthogonal component of p_T^{miss} with respect to the lepton.
3	M_T	The combined transverse mass of the lepton and the E_{miss} system, which is given by $M_T = \sqrt{2p_T^l E_T^{miss}(1-\cos\Delta\phi)}$.
4	N_J	The number of jets entering the H_T sum.
5	N_B	The number of jets identified as originating from a b quark.
6	M_J	The invariant mass of all jets entering the H_T sum.
7	p_T^l	The lepton p_T.
8	η_l	The lepton pseudorapidity.
9	$Isol_{ch}$	The lepton isolation, Iso $= \frac{\sum_{p\neq q} p_T^p}{p_T^q}$ where the sum goes over all leptons except lepton q within $\Delta R < 0.3$.
10	$Isol_\gamma$	Same as $Isol_{ch}$ but with the sum going over all photons.
11	$Isol_{neu}$	Same as $Isol_{ch}$ but with the sum going over all neutral hadrons.
12	q_l	The lepton charge (either -1 or $+1$).
13	$IsEle$	A flag set to 1 if the lepton is an electron, 0 if it is a muon.
14	N_μ	The number of muons entering the sum in $p_{T,TOT}^\mu$.
15	M_μ	The combined invariant mass of all muons entering the sum in $p_{T,TOT}^\mu$.
16	$p_{T,TOT}^\mu$	The vector sum of the p_T of all PF muons in the event with $p_T > 0.5$ GeV.
17	N_e	The number of electrons entering the sum in $p_{T,TOT}^e$.
18	M_e	The combined invariant mass of all electrons entering the sum in $p_{T,TOT}^e$.
19	$p_{T,TOT}^e$	The vector sum of the p_T of all PF electrons in the event with $p_T > 0.5$ GeV.
20	N_{ch}	The number of all charged hadron PF-candidates.
21	N_{neu}	The number of all neutral hadron PF-candidates.
22	N_γ	The number of all photon PF-candidates.

Table 5.1.: High level features (HLFs) of the data set.

2. Feature Selection

Quantum algorithms are currently run on small noisy devices. In this thesis, each feature of the data set is represented by one qubit. This implies that data needs to be efficiently compressed in order to be studied with a quantum approach. This data reduction is done with the Principal Component Analysis (PCA) method [105].

This method tries to find the (combination of) features that vary the most across the data samples. PCA is usually used to visualise the data in a low dimensional space and to reduce the dimensionality of the input data to an amount such that the selected algorithm can handle.

Given a data set $\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$ of m samples with n data features, we can arrange our data in the form of a m by n matrix X. The algorithm of the procedure aims at getting linear combinations of the original features shown in Algorithm 4.

input : Data set matrix X, number of components $n_{\text{components}}$

1 Center the data by subtracting from each column the mean of that column,

$$\mathbf{x}_i \leftarrow \mathbf{x}_i - \frac{1}{m}\sum_{i=1}^{m}\mathbf{x}_i \tag{5.2}$$

2 Form the n by n covariance matrix

$$C = X^T X = \sum_{i=1}^{m}\mathbf{x}_i\mathbf{x}_i^T \tag{5.3}$$

3 Diagonalize C to the form $C = X^T X = W\Lambda W^T$. The columns of W are the normalized eigenvectors, called principal components, and Λ is a diagonal matrix with the corresponding eigenvalues ;

4 Pick the $n_{\text{components}}$ largest eigenvalues $\lambda_1, \ldots, \lambda_{n_{\text{components}}}$ and their corresponding eigenvectors $\mathbf{v}_1, \ldots, \mathbf{v}_{n_{\text{components}}}$. Construct the n by $n_{\text{components}}$ matrix $\tilde{W} = [\mathbf{v}_1, \ldots, \mathbf{v}_{n_{\text{components}}}]$;

5 Dimensional reduction. Transform the data matrix as :

$$\tilde{X} = X\tilde{W} \tag{5.4}$$

Algorithm 4: PCA algorithm to reduce the dimensionality of a problem.

3. Quantum representation of the data set

Data embedding In this thesis, each data point contains features with continuous values. Discretization would imply using a significant amount of qubits to describe every feature. Thus, basis encoding is not considered. Amplitude encoding would imply using large depth quantum circuits to represent each data point compared to the angle encoding and its constant depth $\mathcal{O}(1)$. Thus, we decide to embed the classical data points using the angle encoding with R_y rotation presented in section 2.2 of chapter 3.

Feature scaling Due to the periodicity of the R_y encoding, the training data set was scaled in $[-\frac{\pi}{2}, \frac{\pi}{2}]$ using a "MinMaxScaler" [106]. The transformation is given by

$$X_{\text{scaled}} = \frac{X - \min(X)}{\max X - \min X}\pi - \frac{\pi}{2} \tag{5.5}$$

Once fitted to the training data set, the scaler is applied to the testing data set.

6. Results and Discussion

In this chapter, we present the results of the GAN based anomaly detection workflow on data sets of SM, Higgs and Gravitons events.

1. Recognizing a Standard Model event

In HEP, anomaly detection is efficient if we are able to affirm that an event corresponds to an SM process with a high probability. Specifically, we can characterize a data point well if we have sufficient information about it. More explicitly, we need to consider sufficiently many features to ensure a reliable training of the model. However, it should be noted that if too many features are considered in the training data this might also not be helpful. It is, thus, important to choose the used data features wisely, e.g., by using feature selection. This can be seen by looking at the cumulative explained variance obtained with PCA on the SM data set, w.r.t. the number of principal components. This is shown in Fig. 6.1. It shows that it needs a small amount of principal components to recognize the graviton while 10 principal components are needed to fully describe an SM data point and a Higgs data point.

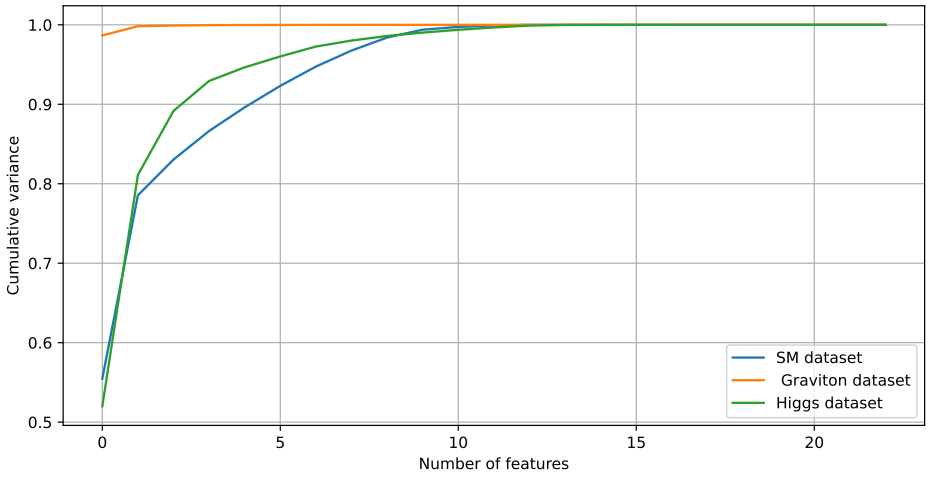


Figure 6.1.: Cumulative explained variance of the test data point w.r.t. the number of principal components. A PCA is applied to each data set and the cumulative variance is computed.

2. Classical benchmark

To put the performance of the quantum approach into context, we benchmark it against a comparable classical approach. The comparison is done by using the same amount of

parameters as in the quantum case. We look at following setups of classical GANs (W and b are the weight and bias parameters of a layer in the GAN):

- A reference GAN where every layer has the form $f(x) = \text{LeakyReLU}_\alpha(Wx + b)$. We refer to this setup as the "vanilla" setup.
- The reference GAN with a dropout layer with probability $\frac{1}{4}$. This means that the dropout layer ignores a randomly selected subset of neuron outputs in the network only during training with probability $\frac{1}{4}$. We refer this setup as the dropout setup.
- A GAN where all layer have the form $f(x) = \text{ReLU}(Wx + b)$. We refer this setup as the ReLU setup.
- A GAN where all layer have the form $f(x) = \sigma(Wx + b)$. We refer to this setup as the sigmoid setup.

Here, we have :

$$\text{ReLU}(x) = \max(x, 0)\,, \tag{6.1}$$

$$\text{LeakyReLU}_\alpha(x) = \begin{cases} x & \text{if } x \geq 0 \\ \alpha x & \text{if } x < 0 \end{cases}\,, \tag{6.2}$$

$$\sigma(x) = \frac{1}{1 + \exp(-x)}\,. \tag{6.3}$$

In all setups, the last layer uses a sigmoid activation function.

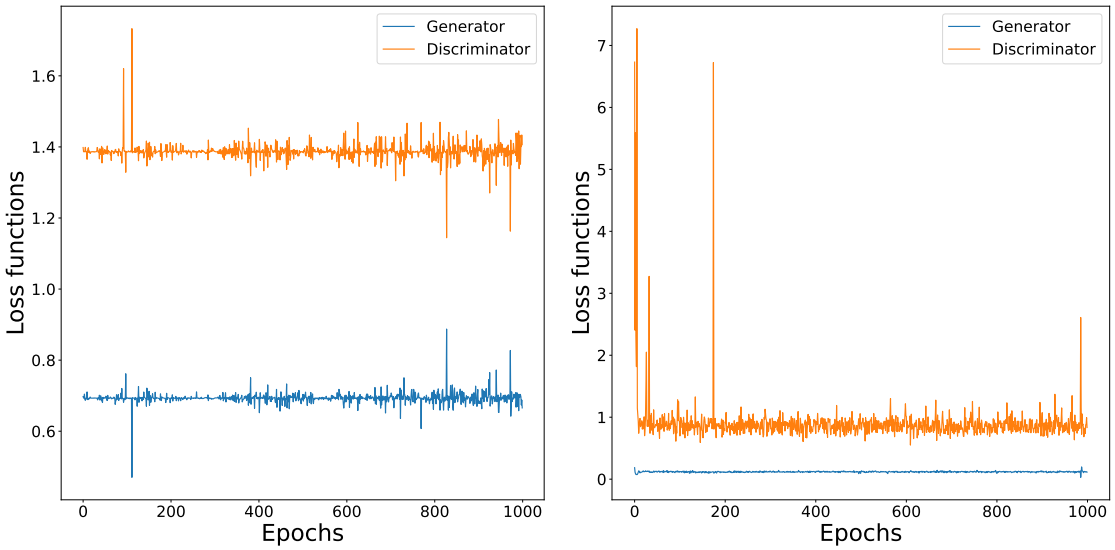
We also look at the same four setups but with WGANs. In the following, we look at the results for two features and for eight features.

2.1. Anomaly detection with 2 features

With two features, we use 6 trainable parameters for the generator and 11 for the discriminator. All of the LeakyReLU functions in the following have a parameter $\alpha = 0.2$. Firstly, we train all of the neural networks for 1000 epochs using the Adam optimizer [37] with a learning rate of 10^{-3} and momentum terms $(\beta_1, \beta_2) = (0.7, 0.99)$. The loss functions of two different GAN/WGAN architectures converge at the end of the training procedure and are shown in Fig. 6.2.

In all cases, the generator and discriminator loss functions do not evolve that much. The best training behavior is not the main goal of this thesis, as long as the classification is possible.

Once the training is finished, the anomaly score is computed for each single point in the training and testing data set as in [4]. Each anomaly value is computed by optimizing the anomaly loss function defined in section 1.1 of chapter 4. This is done with the Adam optimizer with learning rate 10^{-3} and momentum terms $(\beta_1, \beta_2) = (0.7, 0.99)$ and during 1000 epochs. The mean of the anomaly scores for the training data sets defines thresholds which in turn help to identify whether testing data points correspond to anomalies or not. A grid search for the best α parameter can help to improve the anomaly detection by maximizing the AUC. Due to their similarities, we restrict the plots to the anomaly score w.r.t. the α parameter to the "vanilla" setup (see Fig. 6.3).



(a) Vanilla GAN **(b)** Sigmoid WGAN

Figure 6.2.: Loss functions of different GAN and WGAN architectures for 2 features: (a) "Vanilla" GAN setup . (b) Sigmoid WGAN setup. In all plots, the blue curve corresponds to the generator loss function and in orange the discriminator loss function.

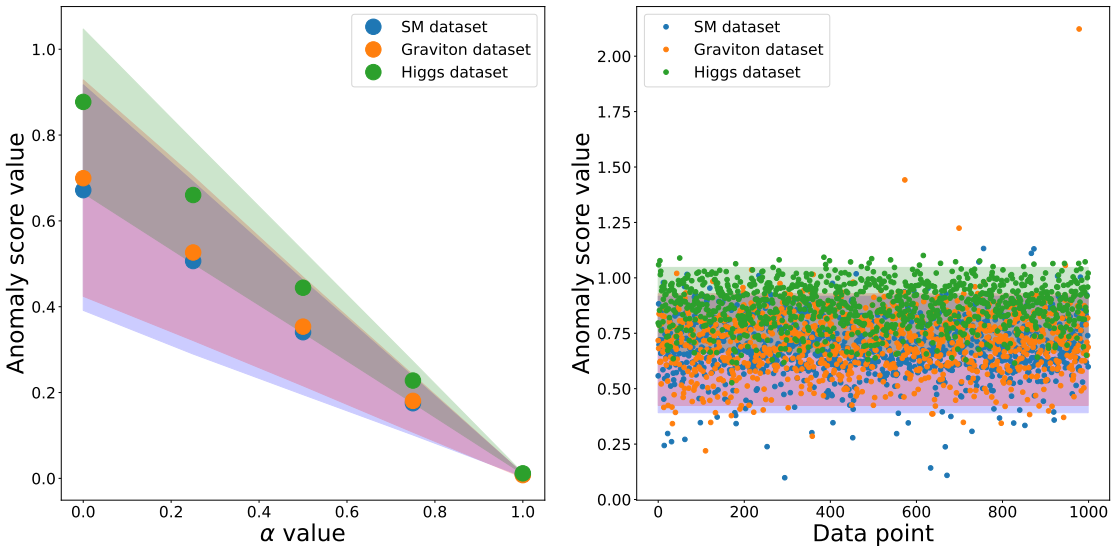
The search of the optimal α is shown in Fig. 6.3 : the left panel represents the mean of the anomaly score of each data set with their 95%-confidence interval w.r.t. the α value. We can see that the optimal configuration is for $\alpha = 0$ giving the maximum AUC score of 0.82 for which the individual anomaly scores are shown in the right panel of Fig. 6.3. In the figure, the anomaly score values of the Higgs events overlap with those of the SM data set, while for the Graviton events, the scores are better separated from the non-anomalous data set. It hints at being able to well classify Graviton events as anomalies while for Higgs events, it is harder.

The performance of the anomaly detection scheme is measured by looking at different metrics for all of the GANs/WGANs (see left panel of Fig. 6.4 and Table. 6.1.) All metrics are computed by varying a cutoff threshold. The Vanilla and Dropout GAN/WGAN all yield a good AUC score of 0.82 but overall, the vanilla and dropout GAN have the best classical performance.

2.2. Anomaly detection with 8 features

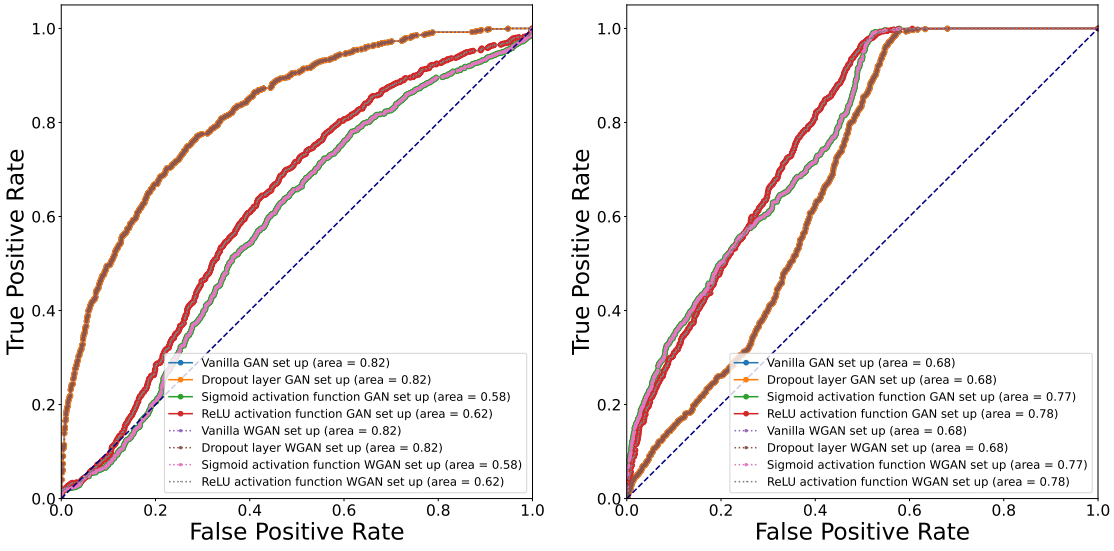
With 8 features, we use 88 variational parameters for the generator and 27 for the discriminator. Firstly, the training of the neural networks are done for 1000 epochs using the Adam optimizer with a learning rate of 10^{-3} and momentum terms $(\beta_1, \beta_2) = (0.7, 0.99)$. Once again, the loss functions are converging to a fixed value while not varying too much during the training (see Fig. 6.5).

We then repeat the procedure of the anomaly detection scheme with 8 features. For each testing data point, we find the anomaly score value by optimizing the corresponding anomaly loss function with the Adam optimizer for 1000 epochs. The situation is similar as with 2 features. The optimal configuration is obtained for $\alpha = 0$ and an AUC score of 0.78 where the SM data points have an anomaly score concentrated around 1 and



(a) **(b)**

Figure 6.3.: Anomaly score scheme with 2 features in the "vanilla" setup of the classical benchmark : (a) Anomaly score w.r.t. α parameters while (b) corresponds to the optimal configuration for all the data point in the test data set and $\alpha = 0$. In all plots, the blue points correspond to the SM data points and the area corresponds to the 95%-confidence interval. The orange points correspond to the graviton data points and the orange area the 95% confidence interval and in orange the discriminator loss function. The green point correspond to the Higgs data points and the green area denotes the 95% confidence interval.

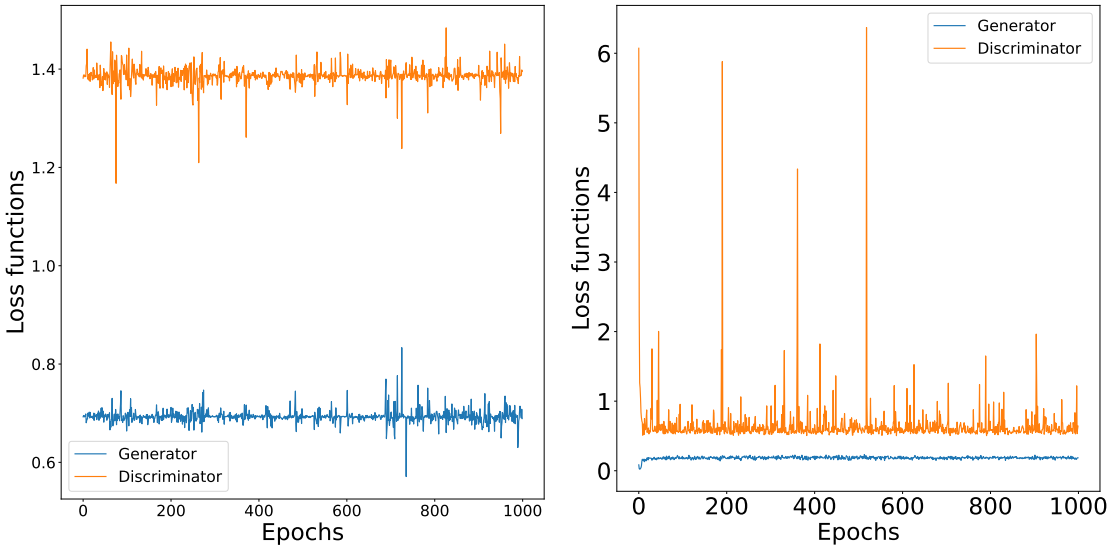


(a) 2 features **(b)** 6 features

Figure 6.4.: ROC curves of the different GAN/WGAN architectures with 2 features(a) and with 8 features (b).

Architecture	F1 score	Accuracy	Precision	Recall	ROC-AUC
Vanilla GAN	0.66	0.77	1	0.96	0.82
Vanilla WGAN	0.5	0.67	0.33	1	0.82
Dropout GAN	0.66	0.77	1	0.96	0.82
Dropout WGAN	0.5	0.67	0.33	1	0.82
Sigmoid GAN	0.52	0.67	1	0.99	0.58
Sigmoid WGAN	0.5	0.67	0.73	1.0	0.58
ReLU GAN	0.54	0.67	1.0	0.95	0.62
ReLU WGAN	0.5	0.67	0.33	1.0	0.62

Table 6.1.: Best classical classification metrics for the classification metrics with 2 features.



(a) Vanilla GAN **(b)** Sigmoid WGAN

Figure 6.5.: Loss functions of different GAN and WGAN architectures with 8 features: (a) "Vanilla" GAN setup and (b) Sigmoid WGAN setup. In all plots, the blue curve corresponds to the generator loss function and in orange the discriminator loss function.

overlaps with almost all of the Higgs data set anomaly score values. However, the Graviton data set anomaly score covers almost all the interval [2, 15] (see Fig. 6.6). This hints at the same situation as with 2 features: while the Graviton events can be easily detected as anomalies the Higgs events cannot.

The classification however is different than with only 2 features (see Table. 6.2). All of the AUC scores are lower than with 2 features (see right panel of Fig. 6.4). In particular, we can see that every F1 score and every recall score are below 0.20 except for the sigmoid GAN case with a F1 score of 0.33 and a recall score of 0.22. Also, we can observe that having a high AUC score does not imply having good classification metrics. The ReLU GAN has the highest AUC score of 0.78 while having a very low recall score of 0.064 and a low F1 score of 0.14. This discrepancy can be explained by the method used to get the metrics. Since they are obtained with a varying cutoff threshold, more threshold points

could give a better estimation of the different metrics. This phenomenon also hints that the more features we use the harder it is to train a model and so the more difficult it is to detect anomalies in a model that is not well-trained.

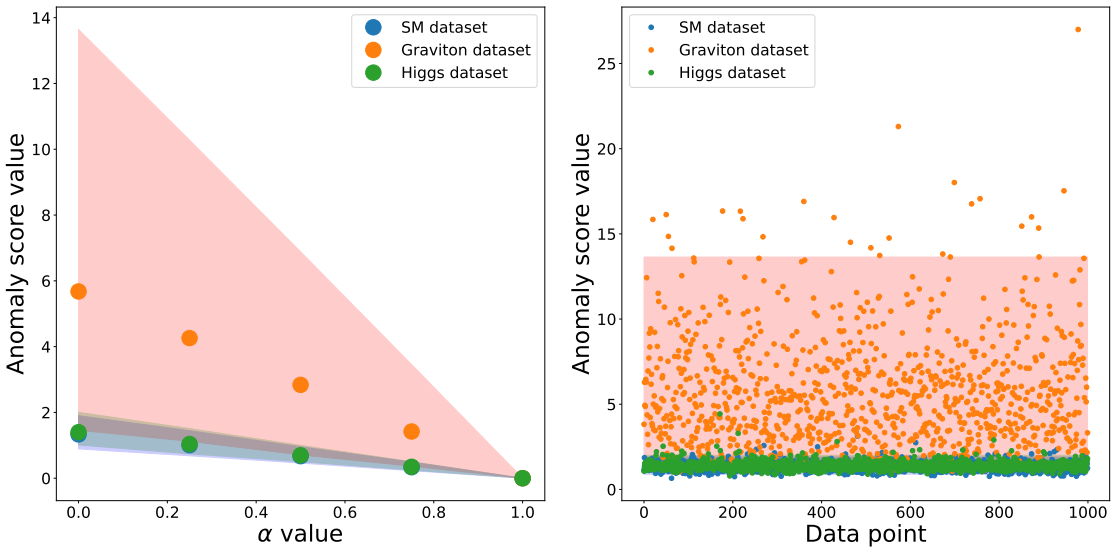


Figure 6.6.: Anomaly score scheme with 8 features in the classical benchmark. (a) shows the anomaly score mean w.r.t. to the α parameters. The optimal configuration for $\alpha = 0$ is shown in the right panel (b) for the whole testing data set. In all plots, the blue points correspond to the SM data points and the area corresponds to the 95%-confidence interval. The orange points correspond to the Graviton data points and the orange area the 95% confidence interval and in orange the discriminator loss function. The green point correspond to the Higgs data points and the green area denotes the 95% confidence interval.

Architecture	F1 score	Accuracy	Precision	Recall	ROC-AUC
Vanilla GAN	0.03	0.67	1	0.01	0.68
Vanilla WGAN	0.04	0.67	1	0.02	0.68
Dropout GAN	0.03	0.67	1	0.01	0.68
Dropout WGAN	0.04	0.67	0.53	0.01	0.68
Sigmoid GAN	0.33	0.71	1	0.22	0.77
Sigmoid WGAN	0.16	0.68	1	0.09	0.77
ReLU GAN	0.14	0.69	1.0	0.08	0.78
ReLU WGAN	0.11	0.68	1	0.06	0.78

Table 6.2.: Best classical classification metrics for the classification metrics with 8 features

Number of features	F1 score	Accuracy	Precision	Recall	ROC-AUC
2	0.655	0.77	1	1	0.82
3	0.75	0.84	1	1	0.90
4	0.74	0.84	1	1	0.90
5	0.96	0.97	1	1	1
6	0.92	0.75	1	1	0.81
7	0.79	0.73	1.0	1	0.89
8	0.33	0.71	1.0	0.22	0.78

Table 6.3.: Best classification score obtained in the classical framework for each dimension between 2 and 8 features.

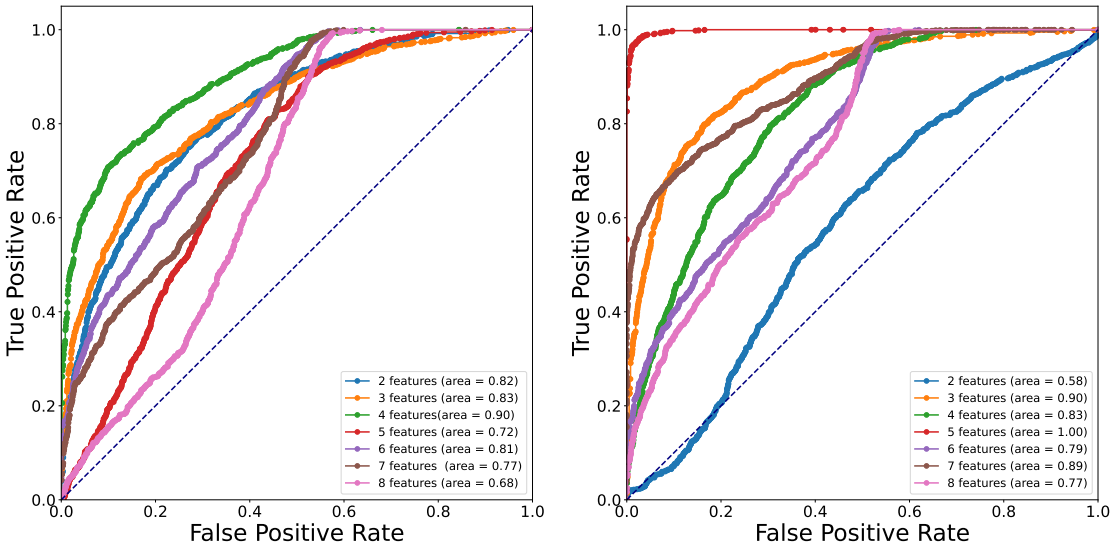
2.3. Comparison of the different performances w.r.t. dimensions

In this thesis, we also try to compare the performance of the classification as we increase the number of features. The more features we add to the model, the easier it should be to recognize an SM event. In particular, the anomaly detection scheme could have a better performance in a high-dimensional framework. However, GAN machine learning models are hard to train, and even if the loss does not converge, the model could still be exploitable for anomaly detection [107]. Hence, one typically studies alternative metrics such as the relative entropy to investigate the training convergence.

The best $F1$, recall, precision, accuracy scores in the classical framework obtained for each dimension are shown in Table. 6.3. Compared to what is previously stated, the different classification metrics do not necessarily increase when we add more features. In particular, we can see that from 2 to 5 features, all of the classification metrics increase. However, for more than 6 features, there is no clear tendency. In particular, in Fig 6.7, we can observe this behavior by looking at the ROC of the "vanilla" GAN and sigmoid WGAN. For the "vanilla" GAN, the ROC-AUC slightly increases as we increase the dimension by going from a ROC-AUC of 0.82 for 2 features to a score of 0.90 with 4 features. However, as we continue to increase the dimension, the score does not continue to increase and rather decreases to a score of 0.68. This all hints at the fact the more features the model has, the harder the training of a GAN.

3. Quantum benchmark

In this section, we present the results of anomaly detection with a qGAN. Our qGAN approach consists of a quantum generator and a quantum discriminator. Every simulation is done using Qiskit [108] and with the same amount of trainable parameters as in the classical case. We compare the results for different numerical simulators: an exact (noiseless) simulator using statevector computations, a noisy simulator that mimic the noise of a quantum hardware, called qasm simulator with noise. The algorithm is also run on IBM processors. Compared to the classical simulations, we limit the training data to a subset of 100 data points. The results presented in this section show that this training data size can be sufficient to achieve similar performance results with a qGAN as classical methods with 10 times as much training data. In every simulation, the discriminator training is done using batches of 30 data points. Due to the current



(a) Vanilla GAN **(b)** WGAN Sigmoid

Figure 6.7.: ROC Curve for different dimensions in the classical framework in the vanilla setup (a) and in the WGAN sigmoid setup.

restriction on IBM hardware, qGAN based anomaly detection is only simulated on IBM hardware with 2 features.

3.1. Anomaly detection with 2 features

The quantum generator is made of 2 repetitions of single-qubit R_Y gates and two-qubit CZ. Compared to simulations with more than 2 features, the quantum discriminator acts on three qubits and is of the form of 3.4. It yields 6 trainable parameters for the generator and 11 for the discriminator. Furthermore, in every training epoch, we have 5 times more optimization updates for the discriminator than for the generator. Every optimization was done in Qiskit [108]. We first present the results in a noiseless, exact simulations with the statevector-simulator.

a. Noiseless simulations

The training corresponds to the Algorithm. 2. We optimize the qGAN structure for 500 epochs with the AMSGRAD optimizer [38], a learning rate of 10^{-3} and $(\beta_1, \beta_2) = (0.7, 0.99)$. As in the classical case, the generator and discriminator loss functions rapidly converge to a fixed value and do not seem to evolve after 200 epochs (see Fig. 6.8). As stated in the classical benchmark, the best learning behavior is not the goal of this thesis, as long as classification is possible.

Once the training is done, we compute the anomaly score for each data point of the testing data set using Eq. 4.3. Contrary to the classical case, the anomaly score can be calculated directly and does not require an optimization over the latent space. This is done by first computing the fidelity between each embedded data point $|\mathbf{x}\rangle$ and the generator state $|\mathcal{G}\rangle$. Then, the discriminator score is computed by loading each

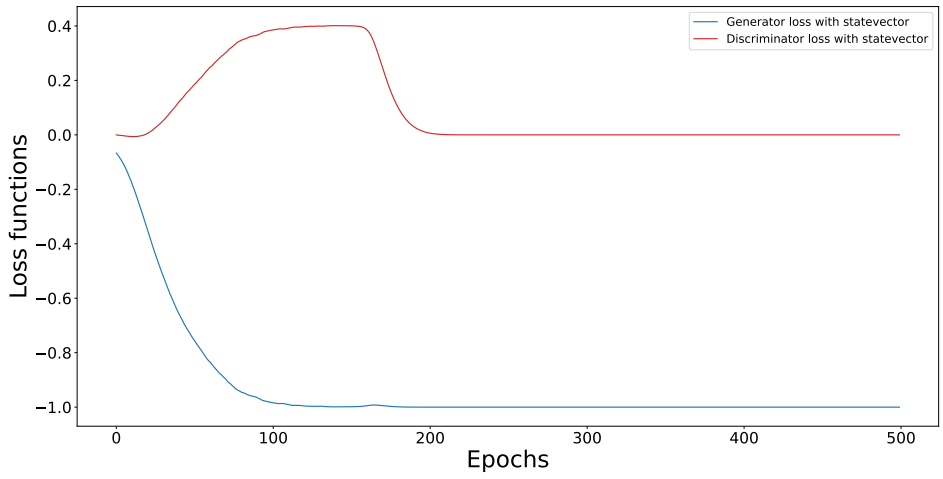
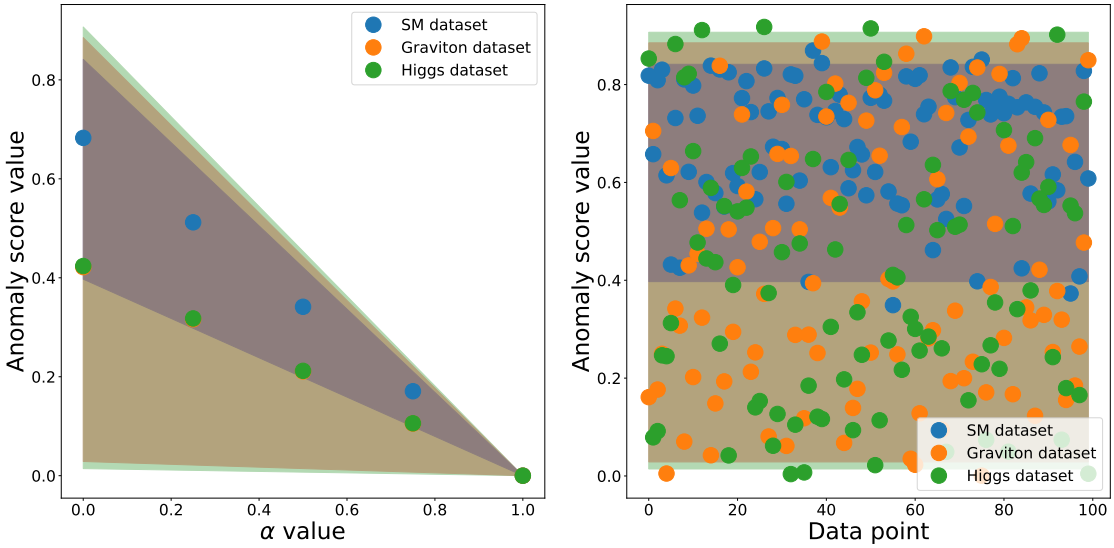


Figure 6.8.: Loss function of the quantum GAN using statevector with 2 features. The blue curve corresponds to the generator loss function and the red one corresponds to the discriminator loss function.

embedded data point $|\mathbf{x}\rangle$ into the discriminator, measuring the last qubit and comparing the measurement result with the one of the generated state $|\mathcal{G}\rangle$. This procedure is repeated for each α value and the best value is obtained by looking at the one maximizing the AUC score (see Fig. 6.9). Here, as in the classical case, the optimal configuration is obtained for $\alpha = 0$ and a AUC score of 0.79. In this setup (see right panel of Fig. 6.9), we can see that the anomaly score values of the SM data set concentrate in the interval $[0.4, 0.82]$. The Graviton and Higgs anomaly scores range in $[0, 0.9]$, but most of their points are located in $[0, 0.4]$. Since the Higgs and the Graviton data sets contain as many points as the SM data set, more data points could be located in the region $[0, 0.4]$ and thus explaining the high classification score (see Table. 6.4) and in the ROC curve (see Fig. 6.10). Moreover, the SM data points overlap with the Graviton anomaly scores and not with the ones of the Higgs. In the quantum setting, the SM scores overlap with both anomalous data points, this explains this discrepancy in the classification metrics between the quantum and the classical results. However, the quantum simulation yields similar metrics to the best classical model. The AUC for both models are similar (0.79 for the noiseless quantum simulations versus 0.82 for the best classical approach). It is also the case for the F1 score, the accuracy and the recall. However, the classical approach yields a better precision score (1 versus 0.61 for the quantum case). Therefore, with less data points, it is possible to obtain a quantum model with a comparable performance as a classical model. A natural extension would be to use bigger data sets and see if this improves the classification.

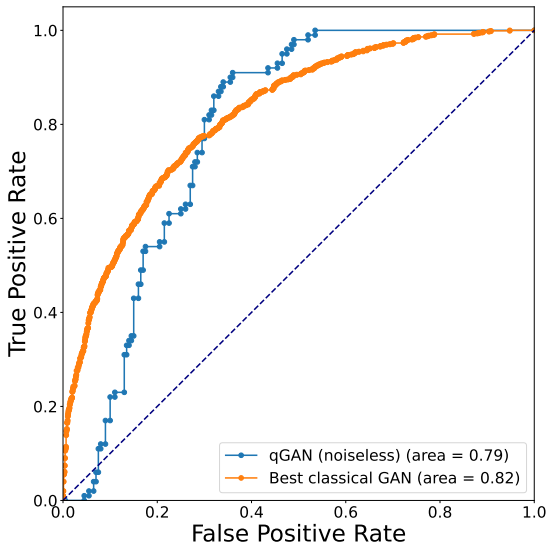
b. Noisy simulator

In this section, we simulate the qGAN training in a noisy simulator which mimics the behavior of a quantum device. We use the same data sets as before. However, due to the noise of the quantum hardware, all of the simulations are done with an increased



(a) (b)

Figure 6.9.: Quantum anomaly score scheme with 2 features on a noiseless simulator. (a) shows the anomaly score mean w.r.t. to the α parameters. The optimal configuration is shown in the right panel (b) for the whole testing data set and $\alpha = 0$. In all plots, the blue points correspond to the SM data points and the area corresponds to the 95%-confidence interval. The orange points correspond to the Graviton data points and the orange area the 95% confidence interval and in orange the discriminator loss function. The green point correspond to the Higgs data points and the green area denotes the 95% confidence interval.



(a)

Figure 6.10.: ROC curve in 2 features of the best classical GAN architecture and of the noiseless quantum simulations. In this case, the quantum framework yields similar results to the classical one.

Framework	F1 score	Accuracy	Precision	Recall	ROC-AUC
Noiseless quantum	0.69	0.74	0.61	1	0.79
Best classical	0.65	0.77	1	1	0.82

Table 6.4.: Classification metrics for the qGAN approach with 2 features on a noiseless simulator and with the best classification metrics.

learning rate of 10^{-2}, 1024 shots and during 100 epochs. With this configuration, the loss functions converges to the same value as in the noiseless simulations and in the classical benchmark (see Fig. 6.11). As previously, after the training of the qGAN, the anomaly score is computed for each point of the data set. Compared to the noiseless simulations, each term of the anomaly score is obtained with 10,000 shots. The more shots we use, the closer the behavior of the algorithm gets to the noiseless simulations. The same α grid search is done to find the one maximizing the AUC value (see left panel of Fig. 6.12). Once again, as in the noiseless simulations and in the classical benchmark, the optimal configuration is obtained with $\alpha = 0$ and a AUC score of 0.52 and shown in the right panel of Fig. 6.12.

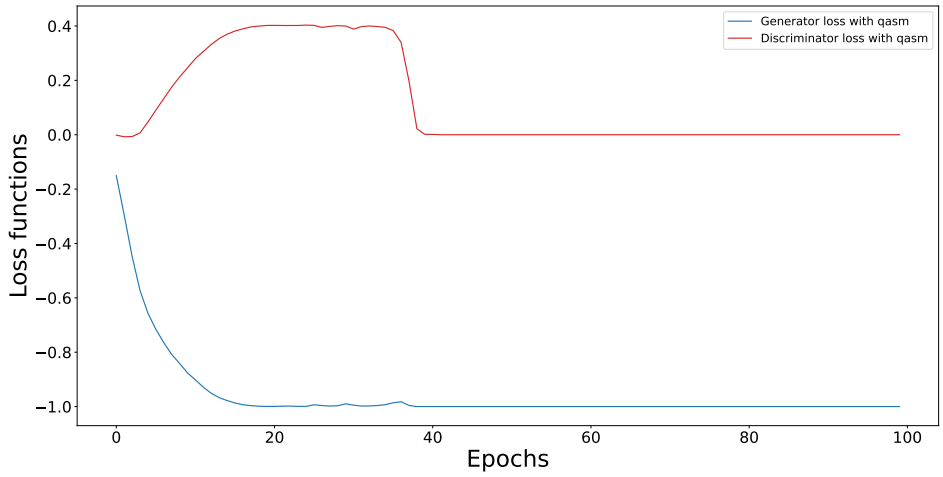


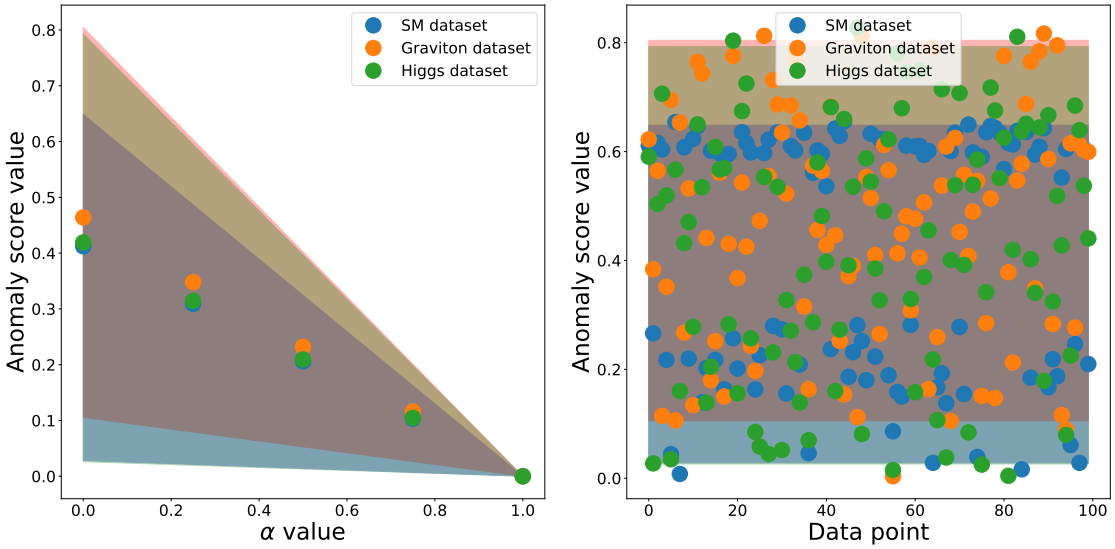
Figure 6.11.: Loss function of the quantum GAN using a noisy simulator with 2 features. The blue curve corresponds to the generator loss function and the red one corresponds to the discriminator loss function.

Compared to the noiseless and classical simulations, the SM data set anomaly scores range in a larger interval. In the region $[0.2, 0.64]$, the SM, the Higgs and the Graviton data points give anomaly scores spread uniformly, making the classification really hard and explaining the bad AUC score of 0.52 (see Fig. 6.13). This can be explained because the different anomaly scores are not concentrated around a certain value as in the classical case. In particular, we can see this tendency by comparing the mean of the anomaly scores versus α for both the quantum and the classical setting. In the left panel of Fig. 6.9, for almost every α, the anomaly score means of all data sets are almost identical while for the classical case (Fig. 6.3), they are distinguishable. As the number of shots is

Framework	F1 score	Accuracy	Precision	Recall	ROC-AUC
Noiseless quantum	0.69	0.74	0.61	1	0.79
Noisy quantum	0.55	0.66	0.46	1	0.52
Best classical	0.65	0.77	1	1	0.82

Table 6.5.: Classification metrics for the qGAN approach with 2 features on a noisy simulator, compared to the best classical results and to the noiseless quantum simulations.

finite, we can expect that the noisy property of the algorithm yields worse results than in the noiseless framework. We can see that every classification metrics of Table. 6.5 is lower to the values obtained in the noiseless case and in the classical case. These scores could be improved if even more shots are used for all components of the anomaly score.

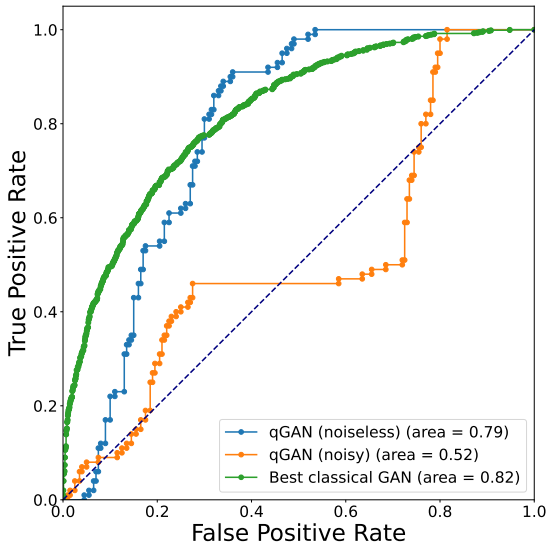


(a) **(b)**

Figure 6.12.: Quantum anomaly score scheme with 2 features on a noisy simulator. (a) shows the anomaly score mean w.r.t. to the α parameters. The optimal configuration is shown in the right panel (b) for the whole testing data set and $\alpha = 0$. In all plots, the blue points correspond to the SM data points and the area corresponds to the 95%-confidence interval. The orange points correspond to the Graviton data points and the orange area the 95% confidence interval and in orange the discriminator loss function. The green point correspond to the Higgs data points and the green area denotes the 95% confidence interval.

c. Hardware runs

Lastly, we run the qGAN based anomaly detection scheme on real hardware. As with the noisy simulator, the training is made with the same training data set and the optimization is done with an increase learning rate of 10^{-2}, with 1024 shots and with error



(a)

Figure 6.13.: ROC curve in 2 features of the best classical GAN architecture, of the noiseless and noisy quantum simulations. In this case, the classical framework outperforms all of the quantum simulations.

mitigation to improve the robustness of the quantum hardware noise. However, due to the current limitations on IBM hardware, the training of the qGAN is performed on 20 epochs and seems unfinished since the loss functions do not converge (see Fig. 6.14). Once the training is done, the anomaly detection computation is also done on real hardware. We load the parameters of the generator and of the discriminator quantum circuits. Then, we compute the fidelity between each $|\mathbf{x}\rangle$ and $|\mathcal{G}\rangle$ with 10,000 shots and the same for the discriminator score as previously described. The α search of the optimal configuration giving the best AUC score is obtained for $\alpha = 0.75$ (see Fig. 6.15). In this case, more SM data points are gathered in the region $[0, 0.2]$ and overlap with both the Higgs and the Graviton (as in the classical case). Also, no points lie in the region $[0.25; 0.8]$, making it a good region to put the hyperplane separating the SM points and the anomalous ones. Contrary to the noisy simulation, the hardware run gives better classification metrics (see Fig. 6.16 and Table. 6.6). In particular, the AUC score is notably higher than the one with the noisy simulation. However, the other classification metrics (F1 score, accuracy, precision and recall) are similar in the hardware run and in the noisy simulations (see Table. 6.6). This discrepancy of the performances in all of the quantum simulations can be explained because the loss functions in the hardware run do not converge and so, maybe running the simulation until we reach convergence could yield similar performances than in the other quantum simulations. However, despite this better performance in the hardware run, the best classical simulation is still outperforming the quantum framework.

3.2. Anomaly detection with 8 features

In this section, as previously presented, the quantum framework is done using both a quantum generator and a quantum discriminator.

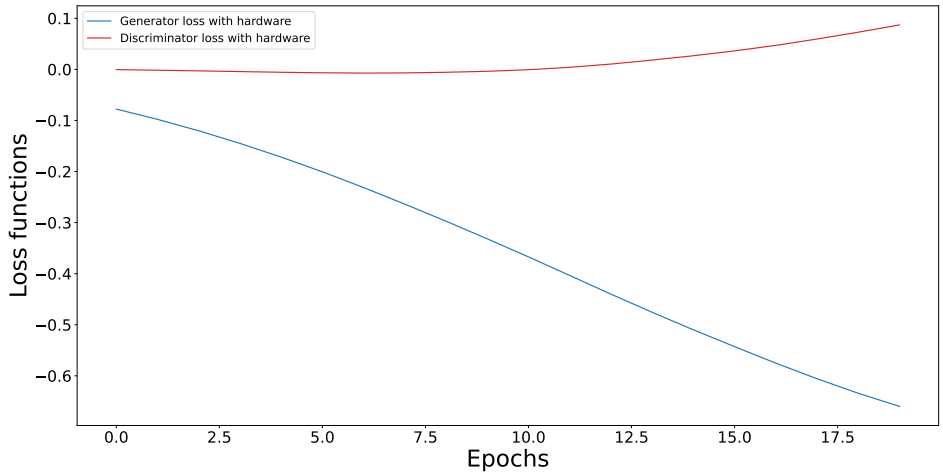
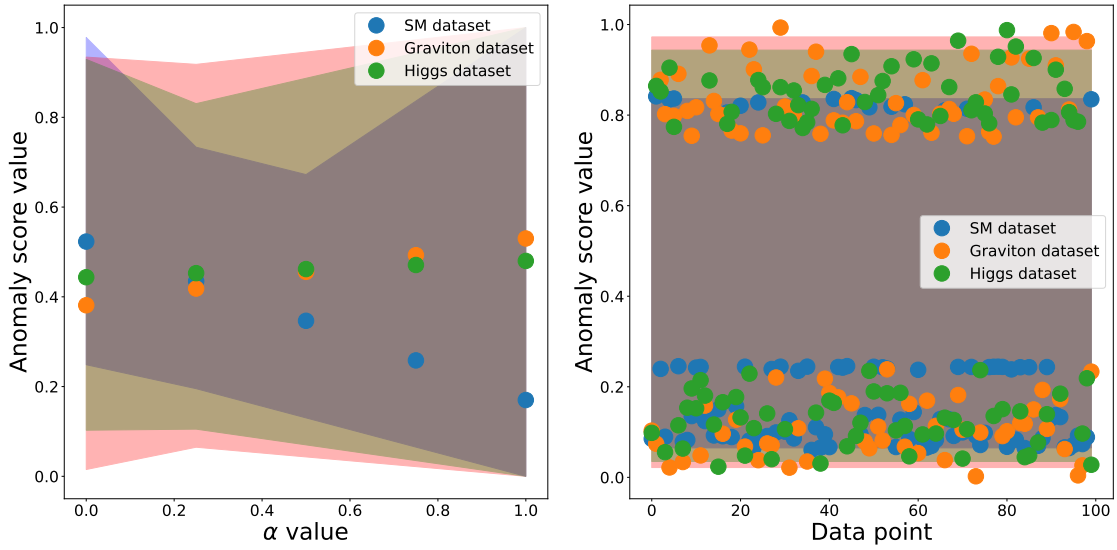
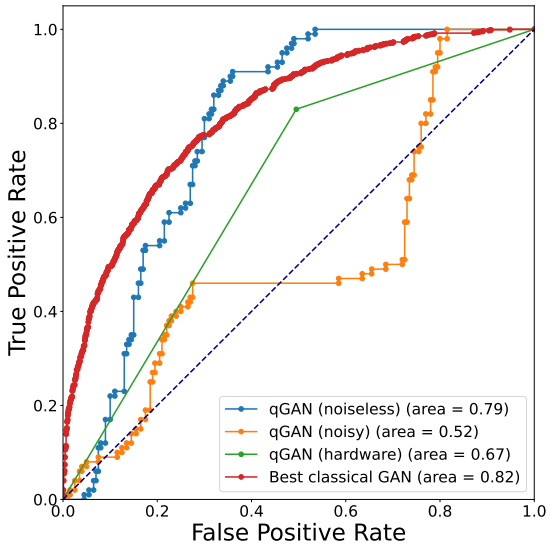


Figure 6.14.: Evolution of the loss functions with 2 features on real quantum hardware. The blue curve corresponds to the generator loss function and the red one corresponds to the discriminator loss function.



(a) **(b)**

Figure 6.15.: Quantum anomaly score scheme with 2 features on a real hardware. (a) shows the anomaly score mean w.r.t. to the α parameters. The optimal configuration is shown in the right panel (b) for the whole testing data set and $\alpha = 0.75$. In all plots, the blue points correspond to the SM data points and the area corresponds to the 95%-confidence interval. The orange points correspond to the Graviton data points and the orange area the 95% confidence interval and in orange the discriminator loss function. The green point correspond to the Higgs data points and the green area denotes the 95% confidence interval.



(a)

Figure 6.16.: ROC curve in 2 features of the best classical GAN architecture, of the noiseless and noisy quantum simulations and of the hardware simulation. In this case, the classical framework outperforms all of the quantum simulations. However, the hardware run yields a better AUC score compared to the other quantum simulations.

Framework	F1 score	Accuracy	Precision	Recall	ROC-AUC
Noiseless quantum	0.69	0.74	0.61	1	0.79
Noisy quantum	0.55	0.66	0.46	1	0.52
Hardware quantum	0.59	0.66	0.46	1	0.67
Best classical	0.65	0.77	1	1	0.82

Table 6.6.: Classification metrics for the qGAN approach with 2 features on a real hardware.

The quantum generator is made of 10 repetitions of single-qubit R_Y gates and two-qubit CZ. The quantum discriminator acts on 8 qubits and is of the form of 3.4. Every optimization is done with Qiskit. The training of the discriminator is made with batches of 30 data points from the full data set of 100 points. Plus, we use 15 times more optimization iterations for the discriminator. We simulate the results in a noiseless simulator and in a noisy simulator.

a. Noiseless simulations

The training corresponds to the Algorithm. 2. In our case, we optimize the qGAN structure with AMSGRAD [38], a learning rate of 10^{-3} and $(\beta_1, \beta_2) = (0.7, 0.99)$ and during 500 epochs. As in the classical case, the generator and discriminator loss functions quickly converge to a fixed value and do not evolve after 100 epochs, which is faster compared to the case with 2 features (see Fig. 6.17). As stated in the classical benchmark, the best learning behavior is not the goal of this thesis, as long as classification is possible.

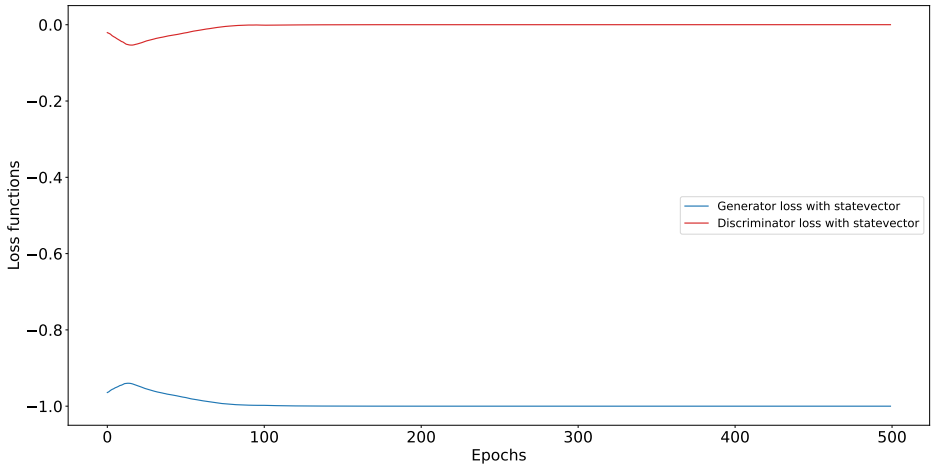


Figure 6.17.: Loss function of the quantum GAN using statevector with 8 features. The blue curve corresponds to the generator loss function and the red one corresponds to the discriminator loss function.

We then repeat the procedure of the quantum anomaly detection scheme with 8 features. Each anomaly score value is obtained by the evaluation of both the fidelity between the embedded state $|\mathbf{x}\rangle$ of the testing data set and the trained generator state $|\mathcal{G}\rangle$ and of discriminator score. This is done for every $\alpha \in [0, 1]$ to find the configuration giving the best AUC score (see left panel of Fig. 6.18). Contrary to the classical case, the optimal configuration is obtained for $\alpha = 0.25$ that yields an AUC score of 0.78 (see right panel of Fig. 6.18). In this setup, as in the classical benchmark, the SM anomaly score values are concentrated in the region [0, 0.05] while the anomalous data points are mostly concentrated in the region [0.05;0.35]. However, the Higgs and the Graviton anomaly scores still have a strong overlap with the non-anomalous data points. The main difference between the classical and quantum settings is that in the classical set-

ting, all of the Higgs data points overlap with the SM and with the Graviton. In the quantum case, a small portion of the anomalous data points overlaps with all of the SM. This explains the high classification scores (see Table. 6.7) and the ROC curve (see Fig. 6.19). In this case, we can see that the quantum simulation gives similar results to the best classical one (a AUC score of 0.78 for the classical versus 0.77 for the quantum). However, except for the precision score (0.67 for the quantum setting versus 1 for the best classical), the classification metrics score are higher in the quantum setting than in the best classical one (see Table. 6.7): the F1 score and the recall score are more important than for the classical setting (0.67 versus 0.33 for the F1 score and 1 versus 0.22 for the recall score). The accuracy of both setting are almost equal to 0.72. Since these quantum scores gives similar or even slightly better than with 2 features, it hints that as we increase the number of features, the classification scores can yield a better anomaly detection. As with 2 features, these simulation are obtained for a subset of the data set used for the classical benchmark. It indicates that the quantum algorithm can give similar or even better results despite learning on a smaller amount of data points.

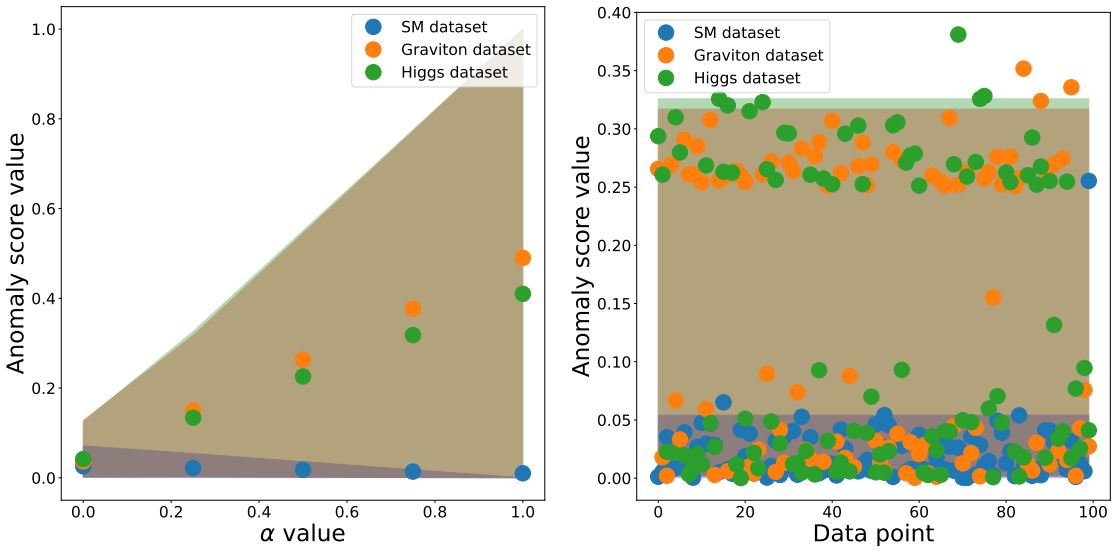


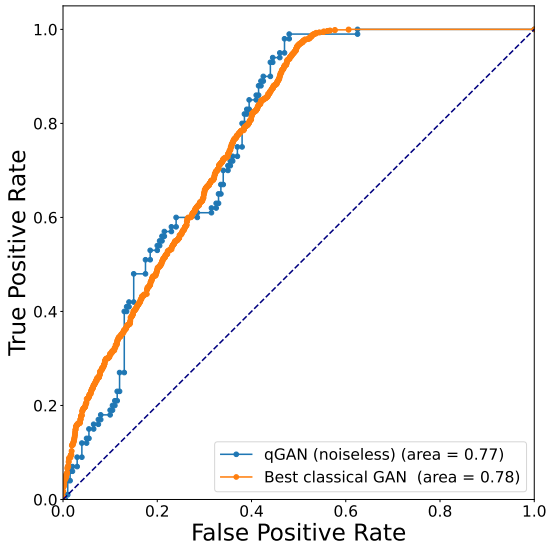
Figure 6.18.: Quantum anomaly score scheme with 8 features using a noiseless simulator. (a) shows the anomaly score mean w.r.t. to the α parameters. The optimal configuration $\alpha = 0.25$ is shown in the right panel (b) for the whole testing data set. In all plots, the blue points correspond to the SM data points and the area corresponds to the 95%-confidence interval. The orange points correspond to the Graviton data points and the orange area the 95% confidence interval and in orange the discriminator loss function. The green point correspond to the Higgs data points and the green area denotes the 95% confidence interval.

b. Noisy simulator

As with 2 features, we also simulate the qGAN training in a noisy simulator that mimic the behavior of a quantum device. The training is done with an increased learning rate of 10^{-2}, 1024 shots and for 100 shots. As observed in the noiseless case and in the

Framework	F1 score	Accuracy	Precision	Recall	ROC-AUC
Noiseless quantum	0.67	0.72	0.67	1	0.77
Best classical	0.33	0.71	1.0	0.22	0.78

Table 6.7.: Classification metrics for the qGAN approach with 8 features on a noiseless simulator compared with the best classical classification metrics.



(a)

Figure 6.19.: ROC curve in 8 features of the best classical GAN architecture and of the noiseless quantum simulations. In this case, the classical framework gives similar results than with the quantum setting.

classical benchmark, the loss functions converge to the same value but stop to evolve after 20 epochs (see Fig. 6.20).

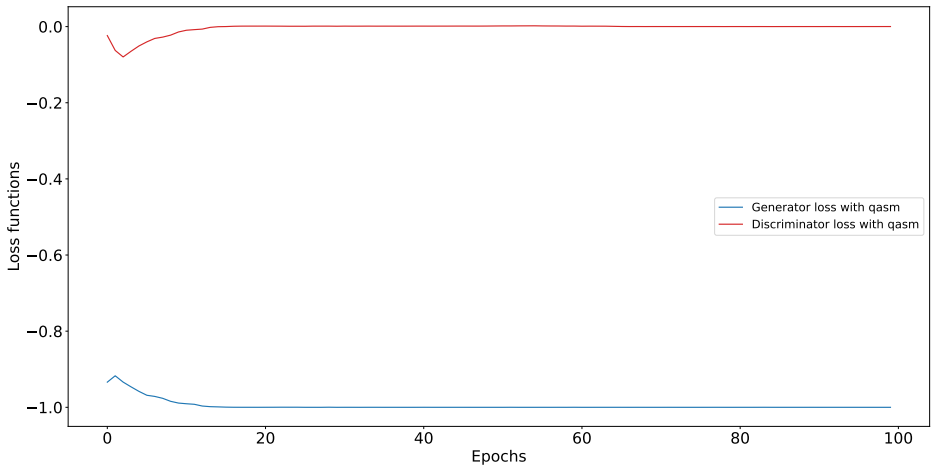
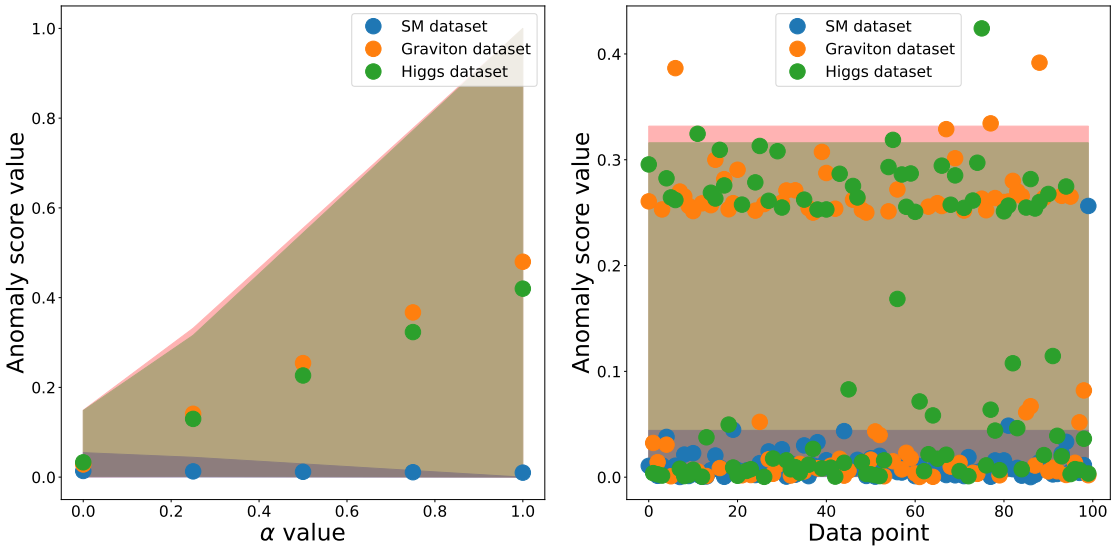


Figure 6.20.: Loss function of the quantum GAN using a noisy simulator with 8 features. The blue curve corresponds to the generator loss function and the red one corresponds to the discriminator loss function.

As before, once the qGAN is trained on the non-anomalous SM data set, we compute the anomaly score on the testing data set. Each term of the score is computed using 10,000 shots. This high number of shots is used to approach as much as possible the behavior of the exact noiseless simulations. The anomaly score is computed for each α-value to maximize the AUC score (left panel of Fig. 6.21). As in the noiseless simulations, the optimal configuration is obtained for $\alpha = 0.25$ yielding a AUC score of 0.75 (right panel of Fig. 6.21). In this setup, the SM data point gather in the region [0, 0.05] while the anomalous data points are spread in the region [0.05, 0.32] while still having an overlap with the non-anomalous data points. However, contrary to the classical case, there is no point lying in the region [0.05, 0.26], hinting that it would be straightforward to separate the SM and the anomalous data points. This explains the relatively high AUC score (see Fig. 6.22). As stated in the noisy simulation case with 2 features, as the number of shots is finite, we expect to have similar or slightly lower classification scores than in the noiseless setting (see Table. 6.8 and Fig. 6.22). Even in this configuration, except for the precision as in the noiseless case, every classification metrics in the noisy setting are better than for the best classical simulation: the F1 score and the recall are still higher (0.67 versus 0.33 for the F1 score and 1 versus 0.22 for the recall), the accuracy is still equivalent in all cases (0.69 versus 0.71). As with 2 features, we can hope that using more shots to compute the anomaly score and for the training of the qGAN could yield even more similar results than in the noiseless case. As we used the same data in both noiseless and noisy setting, a natural extension to this work would be to use more data points for both the training and the testing data set to see whether the performance of the qGAN based anomaly detection scheme is better.

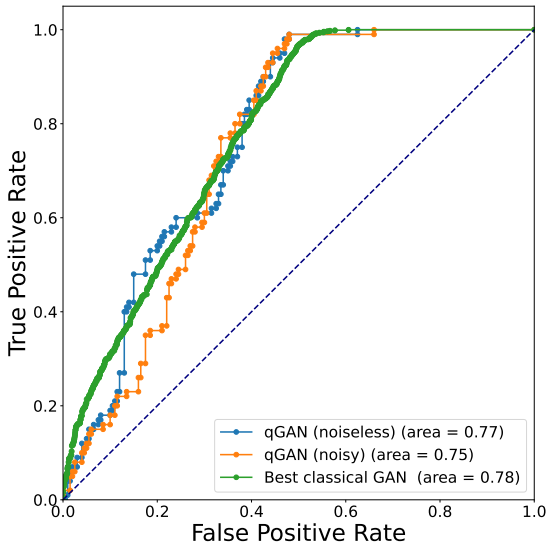


(a) (b)

Figure 6.21.: Quantum anomaly score scheme with 8 features on a noisy simulator. (a) shows the anomaly score mean w.r.t. to the α parameters. The optimal configuration is shown in the right panel (b) for the whole testing data set. In all plots, the blue points correspond to the SM data points and the area corresponds to the 95%-confidence interval. The orange points correspond to the Graviton data points and the orange area the 95% confidence interval and in orange the discriminator loss function. The green point correspond to the Higgs data points and the green area denotes the 95% confidence interval.

Framework	F1 score	Accuracy	Precision	Recall	ROC-AUC
Noiseless quantum	0.67	0.72	0.67	1	0.77
Noisy quantum	0.67	0.69	0.55	1	0.75
Best classical	0.33	0.71	1.0	0.22	0.78

Table 6.8.: Classification metrics for the qGAN approach with 8 features on a noisy simulator compared to the noiseless quantum simulation and to the best classical classification metrics.



(a)

Figure 6.22.: ROC curve in 8 features of the noisy quantum simulations compared to the best classical GAN architecture and to the noiseless quantum simulations. In this case, the classical framework gives similar results than with the quantum setting.

3.3. Comparison of the different performances w.r.t. dimensions

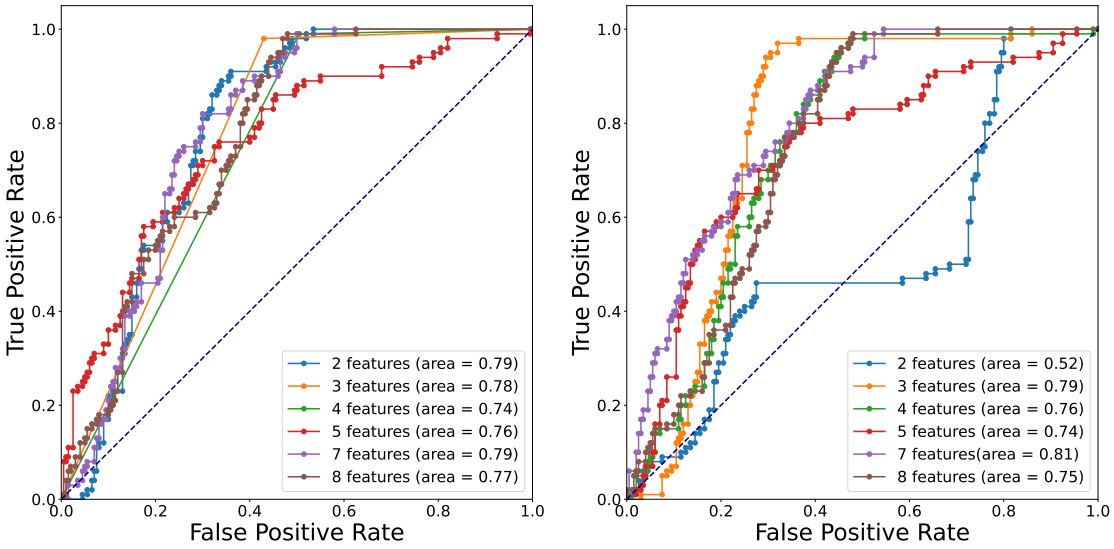
As in the classical case, due to the difficulty of an efficient training, qGAN do not significantly improve their performance when increasing the number of features. We perform the anomaly detection scheme for every feature between 2 and 8 features. The more features we add to the simulations, the more resources we need to efficiently perform the classification between anomalies and non-anomalous data points. In every quantum experiment, we use a smaller data set of 100 data points of the full data set of 1000 points used for the classical simulations. The training of the discriminator is always performed with batches of 30 points. We can see that as we increase the number of features, the classification metrics of the noiseless and noisy quantum setting remain slightly the same (see Table. 6.9 and Fig. 6.23).

For the classical case, we can see that from 2 to 5 features, the classification metrics increase and decrease from 5 to 8 features. However, since the quantum scores remain stable or slightly increase, we can see that it is possible to find a setup where the quantum setting outperforms the classical case: with 8 features, almost all classification metrics for the noiseless and noisy simulations are higher than for the best classical case. The precision, however, remains under for the quantum setting (approximately 0.6 versus 1 in the classical case). Since the data sets used for the quantum approach are subsets of the ones used for the classical one, a natural extension is to study the influence of the size of the batch size.

Number of parameters Another study can also be made in the scalability of this approach by looking at the number of parameters. This is studied in the Table. 6.10. This method requires a linear circuit depth for the discriminator and a bigger circuit depth for the generator. More research has to be made to reduce the gate-count and the

Dimensions	Simulations	F1 score	Accuracy	Precision	Recall	ROC-AUC
2 features	Noiseless	0.69	0.74	0.61	1	0.79
	Noisy	0.55	0.67	0.47	1	0.52
	Best classical	0.65	0.77	1	1	0.82
3 features	Noiseless	0.69	0.71	0.53	1	0.78
	Noisy	0.74	0.78	0.61	1	0.79
	Best classical	0.75	0.84	1	1	0.90
4 features	Noiseless	0.67	0.67	0.5	1	0.74
	Noisy	0.67	0.70	0.56	1	0.76
	Best classical	0.74	0.84	1	1	0.90
5 features	Noiseless	0.62	0.74	1	1	0.76
	Noisy	0.63	0.75	1	1	0.74
	Best classical	0.96	0.97	1	1	1
7 features	Noiseless	0.67	0.74	0.6	1	0.79
	Noisy	0.66	0.75	0.75	1	0.81
	Best classical	0.79	0.73	1.0	1	0.89
8 features	Noiseless	0.67	0.72	0.67	1	0.77
	Noisy	0.67	0.69	0.55	1	0.75
	Best classical	0.33	0.71	1.0	0.22	0.78

Table 6.9.: Best classification score obtained in the quantum framework on noiseless and noisy simulations and in the classical case for each dimension between 2 and 8 features.



(a) Noiseless simulations

(b) Noisy simulations

Figure 6.23.: ROC Curve for different dimensions in the quantum setting with noiseless simulations (a) and noisy simulations (b).

circuit depth of both generator and discriminator in order to find shallower circuits to get similar or even better results than in the framework presented in this thesis.

Dimensions	Parameters	Generator	Discriminator
2 features	Number of qubits	2	3
	Gate count	6 R_y 4 CZ	12 single-qubit, 2 CX
	Circuit depth	6	9
3 features	Number of qubits	3	3
	Gate count	12 R_y and 9 CZ	12 single-qubit, 2 CX
	Circuit depth	14	9
4 features	Number of qubits	4	4
	Gate count	28 R_y, 24 CZ	15 single-qubit, 3 CX
	Circuit depth	32	10
5 features	Number of qubits	5	5
	Gate count	30 R_y, 25 CZ	18 single-qubit, 4 CX
	Circuit depth	32	11
6 features	Number of qubits	6	6
	Gate count	54 R_y, 48 CZ	21 single-qubit, 5 CX
	Circuit depth	58	12
7 features	Number of qubits	7	7
	Gate count	70 R_y, 63 CZ	24 single-qubit, 6 CX
	Circuit depth	74	13
8 features	Number of qubits	8	8
	Gate count	88 R_y, 80 CZ	27 single-qubit, 7 CX
	Circuit depth	92	14

Table 6.10.: Number of trainable parameters in the quantum framework

7. Conclusion

In this work, we present an approach that uses generative modeling in order to detect BSM anomalies in HEP data sets. They consist of measurement results of experiments conducted at the LHC. These experiments can be seen as a black box whose principle is governed by quantum physics. Since the underlying process is quantum, we investigate if the modelling of the process can be done with generative learning based on a quantum ansatz, specifically with qGANs. A qGAN is a generative QML algorithm consisting of the alternative training of two quantum circuits: a quantum generator which generates new data instances and a quantum discriminator which labels a quantum state as drawn from the data set or not. Both generator and discriminator are represented by parametrized quantum circuits. The generator is a quantum channel providing a native model for the physical process of HEP. The quantum discriminator classifies by measuring a single qubit, which mainly helps to reduce measurement errors and to avoid having a global objective function which leads to barren plateaus [52, 53]. Compared to other approaches [4], one advantage of the presented approach is to use a fully quantum setup, with both a quantum generator and discriminator. Unlike other approaches, this method does not rely on a randomness prior given as low-dimensional latent variables which would have to be optimize during the anomaly detection scheme. The training of the qGAN is done on a data set made of data points embedded in a quantum state using angle encoding. This specific choice of embedding has the advantage of supporting continuous input variables and requiring only constant-depth $\mathcal{O}(1)$ circuits. On the contrary, a basis encoding could be more relevant if we need a one-to-one mapping between classical data and a quantum state. Notably, angle encoding distorts the data which does not enable to map a certain basis state to a data point. A qGAN, trained on the SM data set, can then be used to identify HEP processes that do not conform with the SM. The classification is based on an anomaly score, which serves as a distance measure between SM events and the events of interest. We verify the proposed anomaly detection scheme by identifying Higgs and Graviton events. From the point of view of the qGAN, the Higgs and Graviton events represent BSM processes, as they were not included in the SM data set used during the training of the qGAN.

All simulations of a qGAN are done with an ideal noiseless and noisy simulator. The anomaly detection algorithm is also run on hardware for 2 features. We conducted experiments with up to 8 input features and compared the qGANs to classical GANs, using the same number of trainable parameters in both cases.

In this work, we found that the proposed quantum anomaly scheme with 2 features gives slightly worse classification results than in the best classical case in a noiseless simulation. In our case, the noise introduced by the quantum hardware gives a worse performance than for the noiseless for an efficient classification. Increasing the number of shots would help us to approach the results from the noiseless simulations. For the classical setting, we observe that from 2 to 5 features, the anomaly detection gets better and depreciates from 5 to 8 features. This might be due to the fact that the more features we had to the model, the harder it is to train the model. However, in the quantum case, the performance of the algorithm remains similar or slightly better as we increase the number of features. In particular, with 8 features, the quantum model can outperform

the best classical results by giving a F1 score of 0.67 compared to the 0.33 obtained in the best classical case. Notably, the qGANs and classical benchmarks achieved similar performances, although the qGAN used much less training data. This motivates future studies on the influence of required number of data points to train classical respectively quantum models.

The classical and quantum anomaly scheme differ by how they are implemented. Classical anomaly detection requires an optimization of the anomaly loss function for each testing data while the quantum case only requires the measurement of quantum circuits.

However, better quantum performance could be reached with a better tuning of the different hyperparameters (optimizer parameters, ratio between the number of generator and discriminator iteration steps). A natural extension to this work would be to scale up the results on IBM processors when better hardware is available. Then, we could couple this qGAN system to a variational auto-encoder or to other discriminators in order to improve the performance of the classification.

While here we mainly focus on the detection of anomalies, this qGAN framework could be modified to facilitate the generation of events beyond the standard model: once trained on the SM, one could try to generate events on an orthogonal channel for the learned data representation. In doing so, it could be possible to generate anomalies from the SM data set.

Moreover, this work sets the ground for future investigations towards finding a potential quantum advantage for anomaly detection in a HEP data set. Firstly, a quantum approach is useful if there are quantum correlations in the data sets. In principle, this is measurable using Bell's triangle inequalities [109]. These correlations could potentially be efficiently captured with the proposed anomaly detection scheme. This quantum advantage could also be studied with the effective dimension [70]. This metric gives an useful insight on the expressibility of quantum neural network. This could help developing a better understanding of potential advantages of QML compared to classical ML.

Bibliography

[1] Collaboration, T. Aaltonen, and al., "High-precision measurement of the w boson mass with the cdf ii detector," Science, vol. 376, no. 6589, pp. 170–176, 2022.

[2] W. Guan, G. Perdue, A. Pesah, M. Schuld, K. Terashi, S. Vallecorsa, and J.-R. Vlimant, "Quantum machine learning in high energy physics," Machine Learning: Science and Technology, vol. 2, no. 1, p. 011003, mar 2021.

[3] K. Fraser, S. Homiller, R. K. Mishra, B. Ostdiek, and M. D. Schwartz, "Challenges for unsupervised anomaly detection in particle physics," Journal of High Energy Physics, vol. 2022, no. 3, mar 2022.

[4] T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs, "Unsupervised anomaly detection with generative adversarial networks to guide marker discovery," 2017.

[5] S. Lloyd and C. Weedbrook, "Quantum generative adversarial learning," Physical Review Letters, vol. 121, no. 4, jul 2018.

[6] C. Zoufal, A. Lucchi, and S. Woerner, "Quantum generative adversarial networks for learning and loading random distributions," npj Quantum Information, vol. 5, no. 1, nov 2019.

[7] M. Thomson, Modern Particle Physics. Cambridge University Press, 9 2013.

[8] L. Evans and P. Bryant, "LHC machine," Journal of Instrumentation, vol. 3, no. 08, pp. S08 001–S08 001, aug 2008.

[9] M. Baak, , J. Cúth, J. Haller, A. Hoecker, R. Kogler, K. Mönig, M. Schott, and J. Stelzer, "The global electroweak fit at NNLO and prospects for the LHC and ILC," The European Physical Journal C, vol. 74, no. 9, sep 2014.

[10] W. N. Cottingham and D. A. Greenwood, An Introduction to the Standard Model of Particle Physics. Cambridge University Press, 2007.

[11] R. Barbieri, Ten Lectures on the ElectroWeak Interactions. Scuola Normale Superiore, 2007.

[12] P. Langacker, The Standard Model and Beyond. Taylor & Francis, 6 2017.

[13] G. Martinez and A. collaboration, "Alice, the heavy ion experiment at lhc," 2001.

[14] T. A. Collaboration, G. Aad, and al., "The ATLAS experiment at the CERN large hadron collider," Journal of Instrumentation, vol. 3, no. 08, pp. S08 003–S08 003, aug 2008.

[15] T. C. Collaboration, S. Chatrchyan, and al., "The CMS experiment at the CERN LHC," Journal of Instrumentation, vol. 3, no. 08, pp. S08 004–S08 004, aug 2008.

[16] H. Dijkstra, H. J. Hilke, T. Nakada, and T. Ypsilantis, "LHCb Letter of Intent, LHCb Collaboration," CERN, Geneva, Tech. Rep., Jan 1995.

[17] A. M. Sirunyan and al ., "Particle-flow reconstruction and global event description with the cms detector," Journal of Instrumentation, vol. 12, 10 2017.

[18] S. Chatrchyan et al., "Energy calibration and resolution of the CMS electromagnetic calorimeter in pp collisions at $\sqrt{s}$= 7 TeV," Journal of Instrumentation, vol. 8, no. 09, pp. P09 009–P09 009, sep 2013.

[19] "Procedure for the LHC Higgs boson search combination in Summer 2011," CERN, Geneva, Tech. Rep., Aug 2011.

[20] D. Guest, K. Cranmer, and D. Whiteson, "Deep learning and its application to LHC physics," Annual Review of Nuclear and Particle Science, vol. 68, no. 1, pp. 161–181, oct 2018.

[21] S. Farrell et al., "Novel deep learning methods for track reconstruction," in 4th International Workshop Connecting The Dots 2018, 10 2018.

[22] C. Tüysüz, C. Rieger, K. Novotny, B. Demirköz, D. Dobos, K. Potamianos, S. Vallecorsa, J.-R. Vlimant, and R. Forster, "Hybrid quantum classical graph neural networks for particle track reconstruction," 2021.

[23] C. Tüysüz, F. Carminati, B. Demirköz, D. Dobos, F. Fracas, K. Novotny, K. Potamianos, S. Vallecorsa, and J.-R. Vlimant, "Particle track reconstruction with quantum algorithms," EPJ Web of Conferences, vol. 245, p. 09013, 2020.

[24] K. Terashi, M. Kaneda, T. Kishimoto, M. Saito, R. Sawada, and J. Tanaka, "Event classification with quantum machine learning in high-energy physics," Computing and Software for Big Science, vol. 5, no. 1, jan 2021.

[25] G. H. Yann LeCun, Yoshua Bengio, "Deep learning," Nature, 2015.

[26] A. Kadurin, S. Nikolenko, K. Khrabrov, A. Aliper, and A. Zhavoronkov, "drugan: An advanced generative adversarial autoencoder model for de novo generation of new molecules with desired molecular properties in silico," Molecular Pharmaceutics, vol. 14, no. 9, pp. 3098–3104, 2017.

[27] N. Killoran, L. Lee, A. Delong, D. Duvenaud, and B. Frey, "Generating and designing dna with deep generative models," Dec 2017.

[28] A. Brock, J. Donahue, and K. Simonyan, "Large scale gan training for high fidelity natural image synthesis," 2018.

[29] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," 2018.

[30] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of stylegan," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2020.

[31] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2, ser. NIPS'14. Cambridge, MA, USA: MIT Press, 2014, p. 2672–2680.

[32] K. Kurach, M. Lučić, X. Zhai, M. Michalski, and S. Gelly, "A large-scale study on regularization and normalization in GANs," in Proceedings of the 36th International Conference on Machine Learning, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. PMLR, 09–15 Jun 2019, pp. 3581–3590.

[33] W. Fedus*, M. Rosca*, B. Lakshminarayanan, A. M. Dai, S. Mohamed, and I. Goodfellow, "Many paths to equilibrium: GANs do not need to decrease a divergence at every step," in International Conference on Learning Representations, 2018.

[34] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein gan," 2017.

[35] I. Goodfellow, "Nips 2016 tutorial: Generative adversarial networks," 2017.

[36] H. E. Robbins, "A stochastic approximation method," Annals of Mathematical Statistics, vol. 22, pp. 400–407, 2007.

[37] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," CoRR, vol. abs/1412.6980, 2015.

[38] S. J. Reddi, S. Kale, and S. Kumar, "On the convergence of adam and beyond," 2019.

[39] L. Mescheder, A. Geiger, and S. Nowozin, "Which training methods for GANs do actually converge?" in Proceedings of the 35th International Conference on Machine Learning, ser. Proceedings of Machine Learning Research, J. Dy and A. Krause, Eds., vol. 80. PMLR, 10–15 Jul 2018, pp. 3481–3490.

[40] M. Arjovsky and L. Bottou, "Towards principled methods for training generative adversarial networks," 2017.

[41] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, "Improved training of wasserstein gans," 2017.

[42] M. A. Nielsen and I. L. Chuang, Quantum Computation and Quantum Information: 10th Anniversary Edition, 10th ed. USA: Cambridge University Press, 2011.

[43] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," SIAM Journal on Computing, vol. 26, no. 5, p. 1484–1509, Oct 1997.

[44] S. Hallgren, "Polynomial-time quantum algorithms for pell's equation and the principal ideal problem," J. ACM, vol. 54, no. 1, mar 2007.

[45] P. N. e. a. Biamonte J., Wittek P., "Quantum machine learning," Nature, 2017.

[46] C. Zoufal, A. Lucchi, and S. Woerner, "Variational quantum boltzmann machines," Quantum Machine Intelligence, vol. 3, no. 1, feb 2021.

[47] S. Mensa, E. Sahin, F. Tacchino, P. K. Barkoutsos, and I. Tavernelli, "Quantum machine learning framework for virtual screening in drug discovery: a prospective quantum advantage," 2022.

[48] A. Perdomo-Ortiz, M. Benedetti, J. Realpe-Gómez, and R. Biswas, "Opportunities and challenges for quantum-assisted machine learning in near-term quantum computers," Quantum Science and Technology, vol. 3, no. 3, p. 030502, Jun 2018.

[49] Y. Liu, S. Arunachalam, and K. Temme, "A rigorous and robust quantum speed-up in supervised machine learning," Nature Physics, vol. 17, no. 9, pp. 1013–1017, jul 2021.

[50] P.-L. Dallaire-Demers and N. Killoran, "Quantum generative adversarial networks," Physical Review A, vol. 98, no. 1, jul 2018.

[51] R. Sweke, J.-P. Seifert, D. Hangleiter, and J. Eisert, "On the quantum versus classical learnability of discrete distributions," 2021.

[52] J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven, "Barren plateaus in quantum neural network training landscapes," Nature Communications, vol. 9, no. 1, nov 2018.

[53] M. Cerezo, A. Sone, T. Volkoff, L. Cincio, and P. J. Coles, "Cost function dependent barren plateaus in shallow parametrized quantum circuits," Nature Communications, vol. 12, no. 1, mar 2021.

[54] Y. Du, M.-H. Hsieh, and D. Tao, "Efficient online quantum generative adversarial learning algorithms with applications," 2019.

[55] S. Y. Chang, S. Vallecorsa, E. F. Combarro, and F. Carminati, "Quantum generative adversarial networks in a continuous-variable architecture to simulate high energy physics detectors," 2021.

[56] G. Brassard, P. Høyer, M. Mosca, and A. Tapp, "Quantum amplitude amplification and estimation," pp. 53–74, 2002.

[57] G. Agliardi, M. Grossi, M. Pellen, and E. Prati, "Quantum integration of elementary particle processes," Physics Letters B, vol. 832, p. 137228, 2022.

[58] H.-L. Huang, Y. Du, M. Gong, Y. Zhao, Y. Wu, C. Wang, S. Li, F. Liang, J. Lin, Y. Xu, R. Yang, T. Liu, M.-H. Hsieh, H. Deng, H. Rong, C.-Z. Peng, C.-Y. Lu, Y.-A. Chen, D. Tao, X. Zhu, and J.-W. Pan, "Experimental quantum generative adversarial networks for image generation," 2020.

[59] M. S. Rudolph, N. B. Toussaint, A. Katabarwa, S. Johri, B. Peropadre, and A. Perdomo-Ortiz, "Generation of high-resolution handwritten digits with an ion-trap quantum computer," 2020.

[60] M. Plesch and Č . Brukner, "Quantum-state preparation with universal gate decompositions," Physical Review A, vol. 83, no. 3, mar 2011.

[61] E. Farhi and H. Neven, "Classification with quantum neural networks on near term processors," 2018.

[62] R. LaRose and B. Coyle, "Robust data encodings for quantum classifiers," Physical Review A, vol. 102, no. 3, sep 2020.

[63] E. Grant, M. Benedetti, S. Cao, A. Hallam, J. Lockhart, V. Stojevic, A. G. Green, and S. Severini, "Hierarchical quantum classifiers," npj Quantum Information, vol. 4, no. 1, dec 2018.

[64] E. Stoudenmire and D. J. Schwab, "Supervised learning with tensor networks," in Advances in Neural Information Processing Systems, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, Eds., vol. 29. Curran Associates, Inc., 2016.

[65] S. Cao, L. Wossnig, B. Vlastakis, P. Leek, and E. Grant, "Cost-function embedding and dataset encoding for machine learning with parametrized quantum circuits," Physical Review A, vol. 101, no. 5, may 2020.

[66] A. W. Harrow, A. Hassidim, and S. Lloyd, "Quantum algorithm for linear systems of equations," Physical Review Letters, vol. 103, no. 15, oct 2009.

[67] F. P. Maria Schuld, Supervised Learning with Quantum Computers. Springer Cham, 9 2018.

[68] J. R. McClean, J. Romero, R. Babbush, and A. Aspuru-Guzik, "The theory of variational hybrid quantum-classical algorithms," New Journal of Physics, vol. 18, no. 2, p. 023023, feb 2016.

[69] M. Benedetti, E. Lloyd, S. Sack, and M. Fiorentini, "Parameterized quantum circuits as machine learning models," Quantum Science and Technology, vol. 4, no. 4, p. 043001, nov 2019.

[70] A. Abbas, D. Sutter, C. Zoufal, A. Lucchi, A. Figalli, and S. Woerner, "The power of quantum neural networks," Nature Computational Science, vol. 1, no. 6, pp. 403–409, jun 2021.

[71] I. Kerenidis and A. Prakash, "Quantum gradient descent for linear systems and least squares," Phys. Rev. A, vol. 101, p. 022316, Feb 2020.

[72] M. Schuld, V. Bergholm, C. Gogolin, J. Izaac, and N. Killoran, "Evaluating analytic gradients on quantum hardware," Phys. Rev. A, vol. 99, p. 032331, Mar 2019.

[73] K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii, "Quantum circuit learning," Phys. Rev. A, vol. 98, p. 032309, Sep 2018.

[74] M. F. Sohnius, "Introducing Supersymmetry," Phys. Rept., vol. 128, pp. 39–204, 1985.

[75] C. Rovelli, "Loop quantum gravity," Living Reviews in Relativity [electronic only], vol. 1, pp. 1002–, 04 1998.

[76] S. Chatrchyan and al., "Observation of a new boson at a mass of 125 GeV with the CMS experiment at the LHC," Physics Letters B, vol. 716, no. 1, pp. 30–61, sep 2012.

[77] B. Nachman, "Anomaly detection for physics analysis and less than supervised learning," 2020.

[78] G. Aad and al (ATLAS), "Dijet resonance search with weak supervision using $\sqrt{s} = 13$ *tev pp* collisions in the atlas detector," Physical Review Letters, vol. 125, no. 13, sep 2020.

[79] A. S. et al. (CMS), "Music: a model-unspecific search for new physics in proton-proton collisions at $\sqrt{s}$= 13 tev," Eur. Phys. J.C, vol. 81, 2020.

[80] G. Kasieczka and al., "The LHC olympics 2020 a community challenge for anomaly detection in high energy physics," Reports on Progress in Physics, vol. 84, no. 12, p. 124201, dec 2021.

[81] T. Aarrestad and al., "The dark machines anomaly score challenge: Benchmark data and model independent event classification for the large hadron collider," SciPost Physics, vol. 12, no. 1, jan 2022.

[82] K. Fraser, S. Homiller, R. K. Mishra, B. Ostdiek, and M. D. Schwartz, "Challenges for unsupervised anomaly detection in particle physics," Journal of High Energy Physics, vol. 2022, no. 3, mar 2022.

[83] A. Andreassen, I. Feige, C. Frye, and M. D. Schwartz, "JUNIPR: a framework for unsupervised machine learning in particle physics," The European Physical Journal C, vol. 79, no. 2, feb 2019.

[84] T. Dorigo, M. Fumanelli, C. Maccani, M. Mojsovska, G. C. Strong, and B. Scarpa, "Ranbox: Anomaly detection in the copula space," 2021.

[85] F. E. Grubbs, "Procedures for detecting outlying observations in samples," Technometrics, vol. 11, no. 1, pp. 1–21, 1969.

[86] G. Enderlein, "Hawkins, d. m.: Identification of outliers. chapman and hall, london – new york 1980, 188 s., £ 14, 50," Biometrical Journal, vol. 29, pp. 198–198, 1987.

[87] S. Rajasegarar, C. Leckie, and M. Palaniswami, "Anomaly detection in wireless sensor networks," Wireless Commun., vol. 15, no. 4, p. 34–40, aug 2008.

[88] G. Quellec, M. Lamard, M. Cozic, G. Coatrieux, and G. Cazuguel, "Multiple-instance learning for anomaly detection in digital mammography," IEEE Transactions on Medical Imaging, vol. 35, no. 7, pp. 1604–1614, 2016.

[89] R. Zuech, T. M. Khoshgoftaar, and R. Wald, "Intrusion detection and big heterogeneous data: a survey," Journal of Big Data, vol. 2, pp. 1–41, 2015.

[90] M. Farina, Y. Nakai, and D. Shih, "Searching for new physics with deep autoencoders," Physical Review D, vol. 101, no. 7, apr 2020.

[91] T. Schlegl, P. Seeböck, S. M. Waldstein, G. Langs, and U. Schmidt-Erfurth, "f-anogan: Fast unsupervised anomaly detection with generative adversarial networks," Medical Image Analysis, vol. 54, pp. 30–44, 2019.

[92] F. Di Mattia, P. Galeone, M. De Simoni, and E. Ghelfi, "A survey on gans for anomaly detection," 2019.

[93] H. Zenati, C. S. Foo, B. Lecouat, G. Manek, and V. R. Chandrasekhar, "Efficient gan-based anomaly detection," 2018.

[94] J. Donahue, P. Krähenbühl, and T. Darrell, "Adversarial feature learning," 2016.

[95] D. Herr, B. Obert, and M. Rosenkranz, "Anomaly detection with variational quantum generative adversarial networks," Quantum Science and Technology, vol. 6, no. 4, p. 045004, jul 2021.

[96] C. Cortes and M. Mohri, "Auc optimization vs. error rate minimization," in Advances in Neural Information Processing Systems, S. Thrun, L. Saul, and B. Schölkopf, Eds., vol. 16. MIT Press, 2003.

[97] J. Hanley and B. Mcneil, "The meaning and use of the area under a receiver operating characteristic (roc) curve," Radiology, vol. 143, pp. 29–36, 05 1982.

[98] E. R. DeLong, D. M. DeLong, and D. L. Clarke-Pearson, "Comparing the areas under two or more correlated receiver operating characteristic curves: A nonparametric approach," Biometrics, vol. 44, no. 3, pp. 837–845, 1988.

[99] Knapp, O., Cerri, O., Dissertori, G., Nguyen, T. Q., Pierini, M., and Vlimant, J. R., "Adversarially learned anomaly detection on cms open data: re-discovering the top quark," Eur. Phys. J. Plus, vol. 136, no. 2, p. 236, 2021.

[100] A. Buckley, J. Butterworth, S. Gieseke, D. Grellscheid, S. Höche, H. Hoeth, F. Krauss, L. Lönnblad, E. Nurse, P. Richardson, S. Schumann, M. H. Seymour, T. Sjöstrand, P. Skands, and B. Webber, "General-purpose event generators for LHC physics," Physics Reports, vol. 504, no. 5, pp. 145–233, jul 2011.

[101] J. de Favereau, , C. Delaere, P. Demin, A. Giammanco, V. Lemaître, A. Mertens, and M. Selvaggi, "DELPHES 3: a modular framework for fast simulation of a generic collider experiment," Journal of High Energy Physics, vol. 2014, no. 2, feb 2014.

[102] A. Sirunyan and al., "Particle-flow reconstruction and global event description with the CMS detector," Journal of Instrumentation, vol. 12, no. 10, pp. P10 003–P10 003, oct 2017.

[103] O. Cerri, T. Q. Nguyen, M. Pierini, M. Spiropulu, and J.-R. Vlimant, "Variational autoencoders for new physics mining at the large hadron collider," Journal of High Energy Physics, vol. 2019, no. 5, may 2019.

[104] N. Bakhet, M. Y. Khlopov, and T. Hussein, "Phenomenology of large extra dimensions models at hadrons colliders using monte carlo techniques (spin-2 graviton)," 2015.

[105] H. Hotelling, "Analysis of a complex of statistical variables into principal components." Journal of Educational Psychology, vol. 24, pp. 498–520, 1933.

[106] Scikit learn. sklearn.preprocessing.minmaxscaler — scikit-learn 1.0.2 documentation.

[107] P. Grnarova, K. Y. Levy, A. Lucchi, N. Perraudin, I. Goodfellow, T. Hofmann, and A. Krause, "A domain agnostic measure for monitoring and evaluating gans," 2018.

[108] G. Aleksandrowicz, T. Alexander, P. Barkoutsos, L. Bello, Y. Ben-Haim, D. Bucher, F. J. Cabrera-Hernández, J. Carballo-Franquis, A. Chen, C.-F. Chen, J. M. Chow, A. D. Córcoles-Gonzales, A. J. Cross, A. Cross, J. Cruz-Benito, C. Culver, S. D. L. P. González, E. D. L. Torre, D. Ding, E. Dumitrescu, I. Duran, P. Eendebak, M. Everitt, I. F. Sertage, A. Frisch, A. Fuhrer, J. Gambetta, B. G. Gago, J. Gomez-Mosquera, D. Greenberg, I. Hamamura, V. Havlicek, J. Hellmers, Łukasz Herok, H. Horii, S. Hu, T. Imamichi, T. Itoko, A. Javadi-Abhari, N. Kanazawa, A. Karazeev, K. Krsulich, P. Liu, Y. Luh, Y. Maeng, M. Marques, F. J. Martín-Fernández, D. T. McClure, D. McKay, S. Meesala, A. Mezzacapo, N. Moll, D. M. Rodríguez, G. Nannicini, P. Nation, P. Ollitrault, L. J. O'Riordan, H. Paik, J. Pérez, A. Phan, M. Pistoia, V. Prutyanov, M. Reuter, J. Rice, A. R. Davila, R. H. P. Rudy, M. Ryu, N. Sathaye, C. Schnabel, E. Schoute, K. Setia, Y. Shi, A. Silva, Y. Siraichi, S. Sivarajah, J. A. Smolin, M. Soeken, H. Takahashi, I. Tavernelli, C. Taylor, P. Taylour, K. Trabing, M. Treinish, W. Turner, D. Vogt-Lee, C. Vuillot, J. A. Wildstrom, J. Wilson, E. Winston, C. Wood, S. Wood, S. Wörner, I. Y. Akhalwaya, and C. Zoufal, "Qiskit: An Open-source Framework for Quantum Computing," Jan. 2019.

[109] J. S. Bell, "On the einstein podolsky rosen paradox," Physics Physique Fizika, vol. 1, pp. 195–200, Nov 1964.

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Declaration of originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor.

Title of work (in block letters):

Quantum Generative Adversarial Networks for Anomaly Detection in High Energy Physics

Authored by (in block letters):

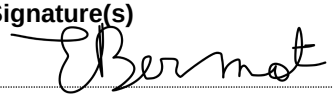
For papers written by groups the names of all authors are required.

Name(s):	**First name(s):**
Bermot	Elie, Bernard, Cadok

With my signature I confirm that

- I have committed none of the forms of plagiarism described in the 'Citation etiquette' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for plagiarism.

Place, date	**Signature(s)**
Zürich, 29.08.2022	

For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.