Unsupervised Anomaly Detection for the ATLAS Level-1 Trigger

Presented by:
Thomas Stern

Prepared for:
Dr. A. Mishra
Dr. J. Keavney
Dr. F. Nicolls
Dept. of Electrical and Electronics Engineering
University of Cape Town

Submitted to the Department of Electrical Engineering at the University of Cape Town in fulfilment of the academic requirements for a Masters of Science degree in Electrical Engineering

April 23, 2025

Declaration

1. I know that plagiarism is wrong. Plagiarism is to use another's work and pretend that it is one's own.
2. I have used the IEEE convention for citation and referencing. Each contribution to, and quotation in, this report from the work(s) of other people has been attributed, and has been cited and referenced.
3. This report is my own work.
4. I have not allowed, and will not allow, anyone to copy my work with the intention of passing it off as their own work or part thereof.

Signature: Signed by candidate

Thomas Stern

Date: April 23, 2025

Acknowledgments

I would like to express my sincere gratitude to my supervisors Amit Mishra and James Keaveney. Their support and encouragement had been fundamental. Amit, thank you for your excitement and commitment to the project, our meeting are always interesting and fruitful. James, thank you for your time, and your interest in helping me develop the underlying models as well as your help in facilitating my journey to CERN.

I would like to thank my parents; Matthew and Elizabeth for supporting me through these two years. You guys are my bedrock.

Lastly I would like to thank the Posse, Eric, Mahmood, Sylla, Mack, Saskia, Andy, Tash, Nick, Gill, Max, Alex, Fred, Nana, and Nono. You guys made the last two years special. Thank You.

Abstract

The Standard Model has governed particle physics for over four decades, yet certain physical phenomena remain unexplained. Therefore, the search for new processes that could elucidate gaps in our current understanding is crucial. At CERN's Large Hadron Collider, search efforts mainly focus on discovering scientifically well-motivated experimental signatures. Yet, in the absence of predefined targets, reliance on models may create blind spots in the data. Searching these blind spots could potentially reveal new physics, a possibility that is especially compelling when considering the low-level data directly read out by the ATLAS detector. The detector collects far more data than can be processed, resulting in over 99% of all data being deleted in real time by the Level-1 Trigger—a chain of field-programmable gate arrays optimized to accept data relevant to the physics processes under study and reject unwanted data. Hundreds of millions of events are rejected every second, possibly discarding something new.

Anomaly detection has become a popular approach for searching for new physics without depending on theorized models, thereby maximizing search sensitivity. Deep learning models based on autoencoders have been researched as mechanisms for detecting specific anomalies. However, while these autoencoder-based methods are effective in representation learning and reconstruction, they may fall short in providing a tailored solution for anomaly detection. These methods rely on the availability of clean training data to teach the model what "normal" samples are. This requirement necessitates the development of large, curated datasets, which would inhibit the development and flexibility of an anomaly detection-based Level-1 Trigger. Furthermore, a preselected background may introduce bias into the detection algorithm, thereby reintroducing model dependence.

A Latent Outlier Exposure-based Level-1 Trigger is proposed to train an anomaly detector in the presence of unlabeled physics anomalies. Latent Outlier Exposure involves simultaneously inferring a binary label for each data point, indicating whether it is anomalous, while updating the model parameters. This is achieved by applying a combination of two losses that share parameters: one for the inferred normal data and one for the inferred anomalous data. This approach was tested on three different anomaly detection systems, including a novel modification to the variational autoencoder's reparameterization trick tailored for anomaly detection. The models were tested on a dataset containing a mixture of simulated Standard Model particle content and postulated, but still unobserved, particle content. Experimental results reveal substantial benefits, especially in addressing the formidable challenge of developing an effective, signal-agnostic Level-1 Trigger.

Contents

List of Figures

List of Tables

Abbreviations

2D	2-dimensional
AD	anomaly detection
AE	Autoencoder
ASIC	application-specific integrated circuit
ATLAS	A Toroidal LHC Apparatus
AUC	area under the ROC curve
AXOL1TL	Anomaly eXtraction Online Level-1 Trigger Algorithm
BF	baseline floating-point
BP	baseline pruned
BSM	Beyond the Standard Model
CMS	Compact Muon Solenoid
CNN	convolutional neural network
CPU	central processing unit
CTP	Central Trigger Processor
DASVDD	deep autoencoding support vector data descriptor
DNN	dense feed-forward neural networks

DSVDD	deep SVDD
ELBO	evidence lower bound
FF	Flip-Flop
FPGA	Field Programmable Gate Arrays
FPR	false positive rate
GM	Gaussian Mixture
GPU	graphics processing unit
HEP	High Energy Physics
HLT	High-Level Trigger
HLS	high-level synthesis
hls4ml	High-level synthesis for machine learning
IO	input-output
KL	Kullback–Leibler
L1	level-1
L1Calo	calorimeters
L1Muon	muon detectors
L1T	level-1 trigger
L1Topo	L1 topological processor
LHC	Large Hadron Collider
LOE	Latent Outlier Exposure
LUT	Look-Up Table
MSE	Mean Square Error
ML	Machine Learning

MUCTPI	Muon-to-Central Trigger Processor Interface
PTQ	Post-training quantization
QAT	quantization-aware training
QCD	Quantum Chromodynamics
ROC	receiver operating characteristics
SM	Standard Model
SVDD	support vector data descriptor
SWAE	Sliced-Wasserstein Autoencoders
SWD	Sliced Wasserstein Distance
TC	Test Crate
TDAQ	Trigger and Data Acquisition
TPR	true positive rate
VAE	Variational Auto Encoder
WAE	Wasserstein Auto Encoders
BRAM	Block Random Access Memory
DPS	Digital Signal Processing slices
URAM	UltraRAM
SLR	Super Logic Region

Glossary

Azimuthal angle Azimuthal angle measured in the transverse plane from the x-axis, in radians within the interval $[-\pi, \pi]$. 39

Bandwidth Bandwidth refers to the amount of data transferred per unit time. 4, 5, 10, 39, 40, 64, 67, 71

Cartesian coordinate system A coordinate system that specifies each point uniquely in a plane by a pair of numerical coordinates (x, y), and in space by (x, y, z). 39

Edge A computing paradigm where data processing occurs close to the source of data generation (e.g., sensors or devices), rather than relying on centralized cloud servers. It helps reduce latency and bandwidth usage. 24

Firmware Software programmed into hardware devices to control low-level functions, often stored in non-volatile memory. v, 4–6, 11, 13, 14, 24, 51, 52, 67, 68, 71, 73

Hardware-in-loop A testing setup where physical hardware is integrated into a simulation loop to evaluate real-time performance and behavior. 5

Jet A collimated spray of particles originating from a quark or gluon, reconstructed using clustering algorithms. 36, 39

Latency Latency is the time delay in data processing or transmission. xi, 3–5, 9–13, 15, 23, 24, 30, 40, 48, 52, 54, 64, 67, 69–71, 73

Missing Transverse Energy The vector that is equal and opposite to the vectorial sum of the transverse momenta of all reconstructed particles in an event. 39

Mixed continuous-discrete optimization An optimization technique involving both continuous and discrete variables, often used in hyperparameter tuning or resource-constrained designs. 4, 72

Muon A heavy lepton, similar to the electron, detected using muon chambers. 35–37, 39

Off-manifold Refers to data points that lie outside the distribution learned by a model, particularly in latent space representations. 3, 9

Phase space A multidimensional space representing all possible states (e.g., positions and momenta) of a physical system. 3, 8–10, 20

Polar angle Polar angle relative to the beam axis (the z-axis). 39

Pruning A model compression technique that removes redundant or unimportant parameters to reduce model size and complexity. ix, 4, 5, 14, 15, 25, 51, 66, 67, 71, 73, 74

Pseudorapidity Pseudorapidity, defined as $\eta = -\log\left(\tan\left(\frac{\theta}{2}\right)\right)$, where θ is the polar angle. 39

Quantization A technique that reduces the precision of model parameters and activations, typically converting floating-point values to integers. 4, 5, 13, 14, 24, 25, 52, 66, 71, 73

Transverse momentum Transverse momentum, the projection of a particle's momentum onto the (x, y) plane. 39

Trigger In high-energy physics, a system that selects potentially interesting events from a high-rate data stream for recording. 3–5, 8–10, 12, 13, 20, 22, 23, 30, 38, 46, 50, 52, 63, 69, 71, 72, 74, 75

Unsupervised A type of machine learning where models are trained on data without labeled outputs. 3–5, 8, 9, 38, 44, 63, 71–74

For Grunto

Chapter 1

Introduction

1.1 Background to the study

The primary goal of particle physics is to understand the origin, characteristics, and interactions of elementary particles, which are considered the fundamental components of the universe. Many of the behaviors and properties of these particles are accurately described by the Standard Model (SM), a well-established theoretical framework. Due to its remarkable success in predicting experimental outcomes, the SM is often regarded as one of the most effective scientific theories ever proposed [1]. However, there are still several phenomena that the SM cannot fully explain, such as the nature of gravity, the asymmetry between matter and antimatter, and the mystery of dark matter. Additionally, certain aspects of the SM, including the Hierarchy Problem [2], require further investigation. To address these gaps, many theoretical extensions of the SM have been suggested, collectively known as Beyond the Standard Model (BSM) physics. Prominent examples of such extensions include Supersymmetry, extra dimensions, and composite models [3].

The process of testing these extensions experimentally typically follows this procedure: Particle theorists first develop precise mathematical predictions for data that could be observed in collider experiments. For instance, a prediction might involve the creation of a new, massive particle, which would lead to a noticeable increase in certain collision rates with a characteristic experimental signature. Subsequently, experimentalists create data analysis techniques to assess the statistical significance of this predicted signature. If the statistical analysis yields convincing results and systematic uncertainties are adequately

controlled, the hypothesis can be either confirmed or rejected.

The Large Hadron Collider (LHC) at CERN, operating since 2010, is capable of accelerating protons to an unprecedented energy of 13.6 TeV, the highest achieved in a laboratory setting. This allows it to generate datasets far larger than those of any previous collider experiments, opening up exceptional opportunities for discovering evidence of BSM physics. To date, analyses of LHC data in search of BSM phenomena have largely followed the traditional model-dependent approach described above. However, no definitive evidence of BSM physics has been found in these experiments [1]. This lack of concrete results has prompted the need for new methods to explore BSM scenarios that have not been constrained by prior theoretical models.

While the traditional approach aligns with the scientific method, it has limitations in that it restricts investigations to predefined BSM scenarios. Moreover, the sensitivity of these analyses to potential BSM signals is often constrained by the accuracy of background process predictions from the SM.

Consequently, there may exist a substantial number of BSM signatures that have either not been considered or are indistinguishable from background processes due to insufficiently precise SM predictions. If these signatures exist, they may have eluded detection in previous LHC analyses. As a result, analyzing LHC data without relying on established BSM models could significantly enhance the chances of uncovering new discoveries.

1.2 Objectives of this study

1.2.1 Problems to be investigated

Within the realm of Machine Learning (ML), the Autoencoder (AE) has surfaced as a potent tool for anomaly detection across diverse domains. AEs learn to reconstruct input data x from a learned lower dimensional representation z [4]. Harnessing their capacity to grasp the intrinsic data distribution and reconstruct input samples, AEs excel in the identification of anomalies or outliers. While AE-based methods are effective in representation learning and reconstruction, they may fall short in providing a tailored solution for detecting anomalies. Another issue is that these methods are heavily dependent on background models due to their reliance on simulations. This can reduce search

sensitivity; as it is difficult to develop accurate simulations that can characterize systematic uncertainties over thousands of final states [5]. This is specially problematic when dealing with the vast data space of the LHC. Even if real LHC data was used, this would be under the assumption that clean training data available for the model to learn the background characteristics. In practice, datasets are often extensive and uncurated, potentially already containing some of the anomalies that one aims to detect [6]. Further, the choice of the background may introduce a bias into the model, thereby dictating which parts of the data space should be explored. This may result in redundant AD searches, rather than the signal agnostic approach the paper aims to achieve.

Another major consideration is that in High Energy Physics (HEP) physicists are almost never able to declare a discovery with a single collision. This is due to the [1]*look elsewhere effect*. Strange collisions are only useful if we can quantify their strangeness and filter useful data from the non-interesting anomalous data. Physicists believe that new physics will manifest as an 'over-density' in Phase space rather than being 'Off-manifold'.

Lastly, the A Toroidal LHC Apparatus (ATLAS) detector outputs massive volumes of data, requiring that 98% of the throughput be rejected. Therefore the model must reject data in real-time while not missing out on all the interesting physics. As collisions take place every 25 nanoseconds, this requires microsecond Latency. Further, the Trigger must be stable as any error may result in valuable data to be lost. Lastly, as the Trigger hardware is underground and currently in use, there will be limited space available for novel AD systems. This further limits the design constraints, as the model must be lightweight to increase the likelihood of implementation [8], [9].

1.2.2 Purpose of the study

To improve feature selection, it is suggested that the model training should be driven by an objective function based on AD [10], [11]. This endows models with the ability to learn rich features catered for AD purposes [12]. Second, this work aims to use Unsupervised AD methods as to not introduce any bias into the AD algorithm. Therefore, this work introduces LOE to adapt standard AE based AD models so they are better suited for a dataset contaminated by unseen anomalies. During training, LOE simultaneously infers

[1]The look elsewhere effect refers to the statistical phenomenon where, in the context of searching for significant signals in various regions, the probability of finding a seemingly significant result by chance increases [7]

anomalous data within the training set and updates its parameters by solving a Mixed continuous-discrete optimization problem, iteratively refining the model and its predicted anomalies. This encourages the model to create a latent distribution in which a portion of the encoded data is pushed out from the main cluster. Further, this encourages the model to create distributions that are better suited to single class AD. AD-optimized latent distributions could also address the *look-elsewhere effect*; by understanding and manipulating where anomalous data clusters in the encoding space, it may be possible to avoid entirely non-interesting anomalous data.

To comply with the strict Latency requirements of the L1T, High-level synthesis for machine learning (hls4ml), is used to integrate ML models on FPGAs as electronic circuits. [13]–[15]. The utilization of hls4ml promotes inference speed, making it ideal for the constraints of the L1T. Further, Pruning and Quantization will be performed on the models to reduce the model footprint as much as possible, without degrading the performance.

This study aims to bring state of the art Unsupervised AD to the ATLAS L1T. The implementation of unrestricted search methods could contribute to the development of new theoretical models in HEP. While the emphasis of ML research has historically centered on the HLT; deploying ML in the vast unknown of the L1T data space is an exciting prospect [16].

1.3 Scope and Limitations

This study aims to design and evaluate resource-efficient AE-based AD models for application within the L1T system in high-energy physics. The focus is on building Unsupervised deep learning models that can be compressed, synthesized in Firmware, and meet strict Latency and Bandwidth constraints. While the work demonstrates promising results in simulation and Firmware synthesis, it does not extend to real-time deployment or integration within operational Trigger hardware. The scope and limitations of the research are detailed below.

1.3.1 Scope

1. Design and implement AE models guided by recent research on AD in HEP.
2. Integrate LOE to enhance the Unsupervised performance of these models.
3. Tune and evaluate various models to identify the best-performing architecture(s).
4. Analyze the learned encoding space to interpret model behavior in an Unsupervised setting.
5. Apply Pruning and Quantization techniques to optimize the best-performing model(s) for hardware deployment.
6. Validate model performance post-compression to ensure minimal accuracy degradation.
7. Translate the optimized model(s) into Firmware using hls4ml.
8. Evaluate the synthesized models for compatibility with Trigger hardware constraints.
9. Verify that the deployed model(s) meet Latency and Bandwidth requirements critical to L1T systems.

1.3.2 Limitations

- The study does not cover deployment in a real-time L1T testbed or on-detector environment.
- Only a subset of AE variants (primarily DNN and VAE-based models) are explored, with no exhaustive comparison across all anomaly detection architectures.
- Latency estimates are based on synthesis reports rather than full Hardware-in-loop validation.
- Data used for model training and validation is simulated or preprocessed; real detector noise and conditions are not incorporated.

1.4 Plan of development

Following the Introduction, Chapter 2 outlines the current state of AD in HEP, as well as novel approaches for training AD systems in the presence of unlabeled anomalies. Chapter 3 covers the ATLAS L1T, hls4ml and different flavours of autoencoders. Chapter 4 provides a summary of the methodology: this section gives a full description of the dataset, model design and testing methods and the steps taken to implement the model in FPGA Firmware. Chapter 5 presents the results, along with an in-depth analysis of the findings. Chapter 6 concludes the report and presents the recommendations for future research. Figure 1.1, below, provides a graphical representation of the plan of development.

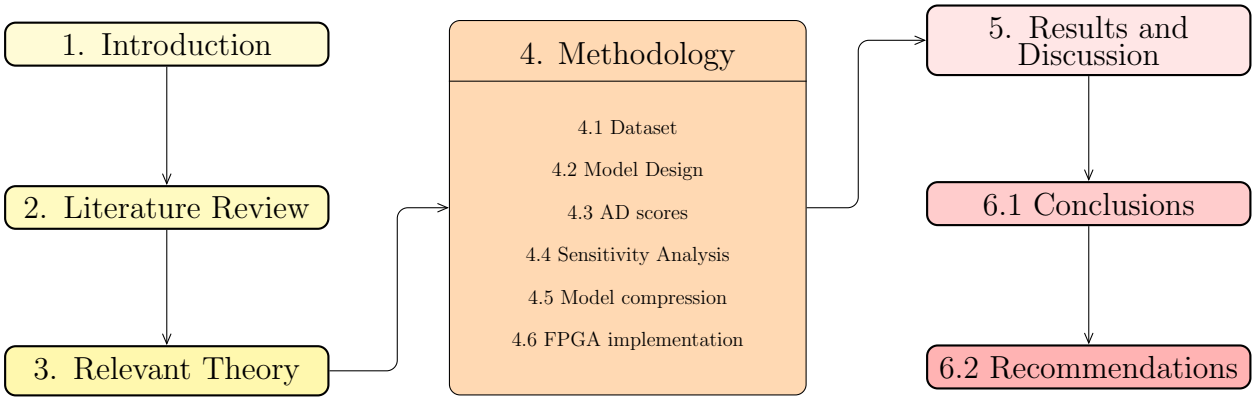


Figure 1.1: Block diagram depicting the report outline.

Chapter 2

Literature Review

2.1 Introduction

In the LHC, protons are accelerated to extremely high energies before they are collided. To study the complex events resulting from these collisions, nine detectors have been installed. Among these, two large general-purpose detectors stand out: the ATLAS and CMS detectors [17]. This research focuses on the ATLAS detector.

The search for BSM physics is a major goal of the experiments at the LHC [18]–[21]. The current, top-down methodology of searches at the LHC targets specific signal models, which are developed based on experimental or theoretical motivations for BSM physics. Using these signal models, physicists generate simulated or synthetic data. This generated signal data is often mixed with synthetic background events to develop a strategy for data analysis. The analysis strategy typically involves selecting events that resemble the signal and implementing techniques to adjust the background rate, ensuring an unbiased statistical analysis. This strategy is then applied to real LHC data [5].

Despite efforts to reduce model dependence, both event selection and background estimation remain highly model-dependent. While current search methods are continually evolving and must continue to do so as new data is acquired, it is evident that HEP requires a complementary search paradigm to fully explore the complex data from the LHC. The reliance on existing search techniques might explain why no new physics has been discovered; model dependence could lead to blind spots that obscure BSM physics signatures

[5].

To date, despite countless BSM searches, a vast Phase space of LHC data remains completely unexplored. In the ATLAS experiment, the two-level Trigger system reduces an initial bunch crossing rate of 40 MHz to 1 kHz, rejecting almost all raw data [22], [23]. The combination of large volumes of unseen data and a limited understanding of how new physics might manifest has inspired a revolution in model-independent searches [24].

Despite recent spikes in interest, model-independent searches have a long history in HEP, dating back to the discovery of the meson [25]. Another example of a model-agnostic approach was the CMS Exotica hotline, where snapshots from the last 24 hours of particle collisions were collected for review. These visual representations were scrutinized to assess the significance of the observed phenomena. The hotline served as an early warning system to highlight potential physics discoveries or anomalies [26], [27]. Additionally, generic searches that make few assumptions about the signal have been used with great success, including the discovery of new particles such as the Higgs boson [28], [29]. In generic searches, such as the *bump hunt*, physicists search for localized enhancements within a smooth background distribution [5].

The major drawback of generic searches is their limited sensitivity, as these searches focus primarily on resonant features and typically do not consider other event properties [5]. Other strategies employed by experiments at the LHC, such as those by CMS [30]–[32] and ATLAS [33]–[35], involve more differential signal model-independent searches. These investigations directly compare experimental data with simulated results across a wide array of distinct final states or bins. While these methods are almost entirely signal model-independent, excluding the feature selection process, the large number of bins can lead to sensitivity issues due to the look-elsewhere effect [7]. Additionally, these methods are heavily dependent on background models due to their reliance on simulation, which can further reduce sensitivity; accurately simulating and characterizing systematic uncertainties across thousands of final states is a significant challenge [5].

ML has become the preferred approach for advancing model-independent searches. Semi-supervised, weakly supervised, and Unsupervised learning methods can enhance sensitivity to subtle or intricate signals while requiring fewer model assumptions compared to conventional search techniques. In particular, the use of AD ML models has garnered significant interest from the HEP community [5], [36], [37].

2.2 Anomaly detection in physics

The challenge of designing searches that are capable of detecting novel physics has led to community-driven initiatives such as the Dark Machines Anomaly Score Challenge [37] and the LHC Olympics [5]. These efforts focus on AD in HEP using Unsupervised offline ML techniques.

AD aims to highlight events that are unexpected when compared to 'normal' data. In this context, Unsupervised ML techniques describe the background event space in a manner independent of the signal. The goal is for BSM physics signals to stand out as atypical compared to the learned background event space [36]. AD methods can be divided into two main categories. The first category involves signal events that are similar to background events. In these cases, information about the expected probability distribution of the background must be used to uncover the signal [5], [38]–[42]. Conversely, when signal events are characteristically different from background events, methods are implemented to identify individual events as anomalous [9], [37], [43]–[45].

AD is a common problem in ML research; however, the unique conditions and requirements in HEP necessitate dedicated approaches. A major differentiator between HEP AD and industry-standard methods is that, in HEP, a single event is often uninformative. An anomaly typically becomes apparent only within the context of a statistical ensemble. This is why HEP searches often target 'over-densities' rather than 'Off-manifold' features as done in industry. Furthermore, data in HEP is consistently distinct from the common data types used for AD ML in industry [5].

Another major design consideration is where AD is implemented in the LHC data processing pipeline. Most AD in HEP is conducted as part of offline analysis [5], [37], where data collected by Trigger algorithms is processed afterwards. While this approach is valid, it overlooks a vast Phase space of data that is deleted in real-time by Trigger systems, which are optimized to accept only the physics processes currently under study. AD could provide a signal-agnostic alternative to the existing model-dependent paradigm, offering a new perspective on the primary data captured in ATLAS. Recent innovations further support this, as ML inference can now meet the LHC low-level Trigger Latency requirements, making a complementary approach to offline analysis feasible [13], [46].

However, it is important to note that algorithms designed for offline scenarios may not

be suitable for online applications due to the constraints of online processing, especially at the L1T. For example, offline methods often involve multiple passes through the dataset to detect anomalous regions in the Phase space, whereas the Trigger observes each collision only once. The Trigger system environment is also highly constrained in terms of Bandwidth and Latency, further limiting the range of AD models that can be implemented. Additionally, even if anomalies can be identified within the required Bandwidth, they are only useful if their level of strangeness can be quantified. As mentioned earlier, in HEP, a discovery cannot be declared based on a single collision. The following analogy from the LHC Olympics paper effectively encapsulates this concept [5]:

> *"We are not looking for flying elephants, but instead a few extra elephants than usual at the local watering hole. The only way to know that the number of elephants is anomalous is to have a precise understanding of the usual rate of elephant"*

2.3 Latent AD

It is standard practice to conduct AD using AE reconstruction errors by considering all features of the input and reconstructed data [47]. However, this approach may result in sub-optimal AD performance, as not all features of the input data are equally necessary or useful [48]. The Mean Square Error (MSE) is a commonly used reconstruction error and anomaly metric [49]. However, in an N-dimensional feature space, the MSE may provide limited information regarding the direction of a distortion vector, potentially restricting its effectiveness in distinguishing between meaningful and trivial anomalies [50]. To address this limitation, researchers utilize the latent space structure to identify anomalous data [51]–[53]. Anguilli et al. [54] introduced a method that incorporates both reconstruction error and latent space geometry, resulting in improved AD, particularly as the dimensionality of the dataset increases. This improvement is illustrated below in Figure 2.1.

Nonetheless, reconstruction error is not always necessary for effective anomaly discrimination [9], [54]–[56]. While combining reconstruction error with latent metrics can achieve state-of-the-art performance, models that rely on reconstruction during inference are not always strictly deterministic, rendering them unsuitable for the L1T [48]. In contrast, models that utilize only latent-based approaches are typically deterministic and eliminate the

need for a decoder, making them an appropriate choice for the L1T.

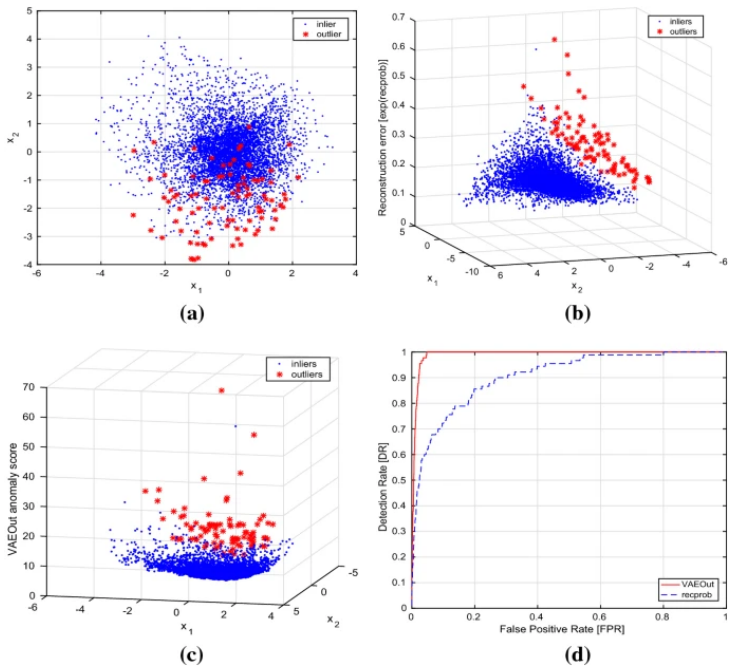


Figure 2.1: Combing the latent space geometry, shown in (a), (b), and (C), with reconstruction error leads to better anomaly discrimination. The AUC curve is shown in (d) [54].

2.4 Online anomaly detection

In HEP, it is standard practice to run ML programs using central processing unit (CPU)s and graphics processing unit (GPU)s, which typically result in inference latencies on the order of milliseconds. However, in the L1T, the event rate must be reduced with an extremely low Latency of just a few microseconds [9]. To meet such stringent Latency requirements, FPGAs or application-specific integrated circuit (ASIC)s are employed for HEP ML applications [13], [37], [57], [58].

An important development in the field of real-time AD for HEP is hls4ml [9]. hls4ml is an open-source software library that translates neural networks [13], [57], [59] into FPGA Firmware. As demonstrated by Govorkova et al. [9], hls4ml can synthesize on-chip implementations of AD models that adhere to both Latency and resource constraints of FPGA implementations in the L1T

Govorkova et al. [9] successfully synthesized both CNN and DNN versions of a supervised VAE and an AE for AD in the ATLAS L1T. The results indicated that all models were viable, except for the CNN AE, which required more resources than were available. The models were synthesized using Vivado HLS 2020.1 [60] for a Xilinx Virtex UltraScale+ VU9P (xcvu9p-flgb2104-2-e) FPGA with the clock frequency set to 200 MHz. One important design constraint, noted by Govorkova et al. [9], is the issue of determinism. The L1T decision must be deterministic, as must all Triggers used in physics studies. This precludes the use of random sampling in the reparameterization trick of the VAE [61]. Consequently, any AD metric that measures the distance between the input and output of a VAE cannot be employed. To address this, the full VAE model is not used. Instead, the AD score is based on the μ and σ values returned by the encoder, leveraging the encoding space to identify anomalies. Two variants of this method were tested: the first variant uses the Kullback–Leibler (KL) divergence loss of the VAE, while the second method employs the z-score of the origin $\vec{0}$ in the latent space, relative to a Gaussian distribution defined by μ and σ. Omitting the decoder maintains determinism and offers additional benefits, such as avoiding the need for buffering data during MSE loss computation, and reducing both resource usage and Latency.

The quest for a model-agnostic L1T is also shared by other experiments at the LHC, not just ATLAS. Zipper et al. [62] developed and tested a VAE for implementation on the CMS Level-1 Global Trigger. Like the ATLAS detector, the CMS detector [63], [64] outputs massive volumes of data, requiring 99% of the throughput be rejected. The CMS L1T rejects data in real-time, on a chain of FPGAs [65] with the added objective of not missing any interesting physics. The CMS Trigger must operate within the constraints of the LHC clock cycle. As collisions take place every 25 nanoseconds, this requires microsecond Latency. Further, the Trigger must be stable as any error that results in 'dead time" will result in the loss of valuable data. Given these considerations, the CMS Trigger is under almost identical design constraints to the ATLAS Trigger. Therefore, research on the CMS Trigger serves as a valuable reference for developing a similar model-agnostic approach for the ATLAS Trigger.".

The VAE architecture, depicted below in Figure 2.2, incorporates an information bottleneck through the implementation of a compact latent space. This design enhances data encoding efficiency and helps the model identify the characteristics of anomalous events. For this implementation, referred to as Anomaly eXtraction Online Level-1 Trigger Algorithm (AXOL1TL), measures were taken to meet resource utilization and Latency criteria. Specifically, the decoder was removed, and the latent space loss term was simplified

during inference. The loss function is simplified by utilizing only the mean-squared term $\sum_i \mu^2$ from the KL-divergence. This adjustment did not degrade performance.

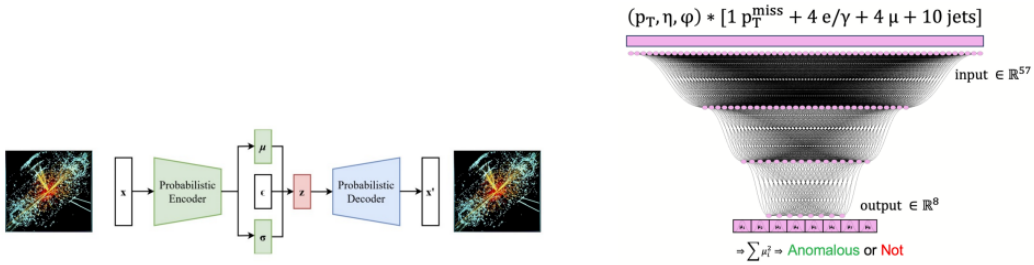


Figure 2.2: Variational Auto Encoder implemented on CMS Global Trigger TC [62].

Zipper et al.[62] successfully developed a L1T model that is signal-agnostic and highly sensitive, enhancing signal efficiency for various physics signatures [9], [13]. The model was implemented in Firmware and integrated into the CMS L1T architecture. It was tested by deploying the Firmware on the CMS Global Trigger[1] TC. The implementation performed robustly when tested on collisions from 2023. However, further updates to the algorithm and downstream Trigger logic are needed to fully integrate the algorithm into the L1T system.

2.5 Model Compression

The quest for better performance in deep learning has led to the development of larger, more intricate models. However, edge devices implemented in the LHC require highly efficient inference, which necessitates reducing Latency, model size, and energy consumption. One effective method for limiting model size is Quantization. Fortunately, hls4ml supports both PTQ and quantization-aware training (QAT), allowing neural networks to be compressed significantly, which reduces resource usage on an FPGA while preserving model performance [9], [15], [66]–[68].

There are several possible flavours of quantisation:

- PTQ [13], [69]–[72], the simplest Quantization approach, involves reducing the baseline model precision, such as lowering the bit-width or converting from floating-point to fixed-point. PTQ generally maintains good algorithm stability, though

[1]The TC consists of identical MP7 boards used as backups for the production system and for experimenting with new Trigger strategies during testing [62].

some accuracy reduction is expected. Typically, the lower the precision, the greater the accuracy loss.

- QAT constrains the model to fixed-point precision during training. This can be achieved using the QKeras [68] library, which helps reduce accuracy loss at higher levels of Quantization, outperforming the weight configurations generated using PTQ. However, inconsistent AD performance can occur when implementing QAT with VAEs. This inconsistency arises because QAT prioritizes optimal input-to-output reconstruction, which does not always lead to improved AD outcomes. Ultimately, the stability of the model will depend on the nature of the anomaly [9].

- Knowledge distillation with QAT involves reframing the optimization objective. Rather than focusing on reducing the quantized model's loss by fixing its weights, the primary goal is to minimize the difference between the loss incurred by the quantized model and the original floating-point model when processing identical inputs. This approach entails training a distinct model that predicts the floating-point outputs of the original model, given the same input, instead of merely training a quantized copy. The aim is to match the floating-point AD performance with a different model that meets the performance constraints of the L1T.

- Anomaly classification with QAT: This involves re-framing the QAT approximated loss regression into a classification problem. Instead of seeking an approximation for the floating-point decision, one could attempt to obtain a binary yes/no response to an alternative query: "Would the floating-point algorithm return an AD score larger than a threshold for this event"[9]. This approach enables setting the threshold based on the precise floating-point model, ensuring satisfactory anomaly acceptance accuracy without the need to predict the precise AD score across various orders of magnitude [9].

Govorkova et al. [9] focused on the first two approaches. The initial step was to develop a reference model against which the performance of QAT and PTQ could be compared. This was achieved by Pruning 50% of the connections in the BF models' layers using magnitude-based Pruning, which involves setting the smallest weights in a tensor to zero, thereby eliminating redundant weights [14], [73]–[76]. This is especially effective, as the hls4ml library excludes all multiplications with zero weights when converting the network into Firmware, reducing the number of floating-point operations required and conserving substantial FPGA resources [14]. Pruning is applied using the polynomial decay method

integrated within the TensorFlow Pruning API, which provides a straightforward drop-in replacement for Keras layers [77]. The resulting model is known as the BP model. Pruning is applied exclusively to the encoder, as this component is intended for FPGA deployment. Thereafter, the QKeras library [68] is used to perform QAT, while fixed-point precision is applied to the BP floating-point model for PTQ. In both methods, bit precision is swept from 2 to 16 with a step size of 2, with Pruning again targeting 50% sparsity.

The results indicate that VAEs are not stable as a function of bit width when using QAT. This likely results from the discrepancy between the AD figure of merit used for inference and the objective function employed during training [9]. Consequently, only PTQ is suitable for a VAE, while both PTQ and QAT were found to be stable when implemented on AEs. A key finding is that utilizing only the encoder, rather than the complete VAE, results in a Latency reduction of 50% while maintaining performance. All pruned models satisfied the Latency requirements, and all models met resource requirements apart from the CNN AE model, which utilized more resources than permitted by the L1T constraints.

2.6 AD based objective functions

In AE-based deep methods [78]–[81], models learn to represent normal instances by minimizing reconstruction errors. These models then aim to discriminate between normal and abnormal data based on reconstruction performance. While AE-based methods are effective for representation learning and reconstruction, they may fall short in providing a tailored solution for AD. The emphasis on mimicking the data distribution, combined with reliance solely on reconstruction errors, may not be sufficient for achieving accurate AD [82], [83].

Another caveat is that features are learned indiscriminately, possibly leading to unstable performance [12]. To address this issue, hybrid models have been developed. In these models, AE-based deep methods extract features, which are then used for AD with traditional techniques [84]–[87]. However, since the features extracted in a hybrid model are separate from the AD component, they may not be relevant or optimal for AD [10]. To enhance feature selection, it is recommended that the model training be guided by an objective function specifically designed for AD [10], [11]. This approach enables models to learn features that are better suited for AD [12].

Hojati et al. [88] developed a model called the DASVDD, illustrated in Figure 2.3, which integrates a traditional AD technique (SVDD) with a feature learning model (AE). This approach addresses the limitation of hybrid models by eliminating the separation between feature learning and AD. The DASVDD is a single-class AD technique that trains an AE model, which simultaneously functions as a SVDD on the latent representations generated by the AE. The AE learns to map normal data to a minimum volume enclosing hypersphere, with the SVDD error serving as the guiding metric by quantifying the distance of the encoded data from the center of the hypersphere. The loss function, which also serves as the anomaly score, combines the input-output (IO) reconstruction error with the SVDD error. By including the reconstruction error in the loss function, the model avoids hypersphere collapse and prevents the simplistic solution of mapping all inputs to a fixed point in the latent representation. The results show that DASVDD outperforms several state-of-the-art AD algorithms.

A VAE-based implementation of an SVDD autoencoder was successfully developed by Zhou et al. [12]. However, the AE-based implementation mentioned above offers specific advantages, particularly for deployment in the L1T. The DASVDD model avoids the reparameterization step required in a VAE, allowing both the IO anomaly metric and the latent distribution to be utilized for AD. This dual approach provides two distinct reference points for detecting anomalies, potentially making it more robust.

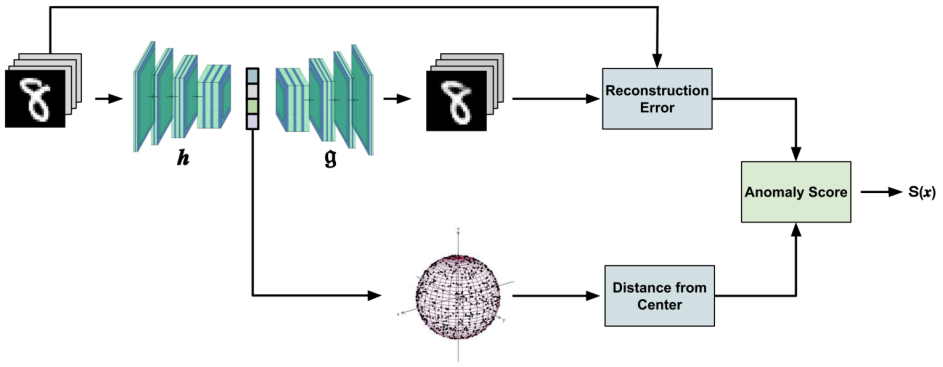


Figure 2.3: Overview of DASVDD model [88]

When both normal and anomalous data are available for training, it is standard practice to employ a binary classification network, trained using a supervised learning approach [89]. However, anomalies are typically rare in real-world scenarios, making it challenging

to gather sufficient abnormal samples for training neural networks. Consequently, training deep learning methods solely on normal data becomes an attractive option [11], [48], [90]. This approach assumes the availability of clean training data for the model to learn the characteristics of 'normal' samples [91]. In practice, this assumption is often challenged, as datasets are frequently extensive and uncurated, potentially containing anomalies that the model is intended to detect [6].

To address this challenge, Qiu et al. [6], drawing inspiration from Outlier Exposure by Hendrycks et al. [92], propose a methodology for training an AD system in the presence of unlabeled anomalies, that is applicable to a wide range of models. The core concept involves simultaneously inferring a binary label for each data point to indicate whether it is anomalous, while iteratively adjusting the model. This approach utilizes a pair of loss functions that share parameters: one for normal data and another for anomalous data.

Like most work in deep AD, a loss function $L^n_\theta(x) \equiv L_n(f_\theta(x))$ is minimized over the normal data. The function $f_\theta(x)$ is employed as a feature extractor for the data x. Additionally, a secondary loss function for anomalies, denoted as $L^a_\theta(x) \equiv L_a(f_\theta(x))$ is used. The feature extractor $f_\theta(x)$ is shared with both losses. When trained exclusively on normal data, the trained loss will output larger values when encountering anomalous data, allowing the loss to be used as an anomaly score. The anomaly loss $L^a_\theta(x)$ acts contrarily to $L^n_\theta(x)$ resulting in a large loss for normal data, and a smaller loss for anomalous data. Assuming, in the interim, that the binary label for each event y is known, the joint is loss function is as follows:

$$L(\theta, y) = \sum_{i=1}^{N} (1 - y_i) L^n_\theta(x_i) + y_i L^a_\theta(x_i). \tag{2.1}$$

Optimizing the shared loss in Equation 2.1 over θ results in better AD than either loss function, L^n_θ or L^a_θ, in isolation [6].

By virtue of L^a_θ, the known anomalies provide an additional training signal to L^n_θ. Through L^a_θ, the labeled anomalies offer supplementary training information to L^n_θ, as the shared parameters guide L^n_θ in learning where normal data should not appear in the latent space. This concept underpins the Outlier Exposure method [92], which generates artificial labeled anomalies to enhance detection accuracy.

Qiu et al. [6] advanced this concept by considering the case where each anomaly label, y_i,

is unobserved. Consequently, the *latent* assignment variables y must be inferred alongside the learning parameters θ. This approach is referred to as LOE. There are two variations of LOE: "Hard" LOE (LOE_H) and "Soft" LOE (LOE_S).

In LOE_H, a constrained set is introduced based on an assumed fixed rate of anomalies, α, in the training data. In this approach, each y_i is assigned a value of either 1 or 0 at a rate consistent with α.

$$Y = \left\{ y \in \{0,1\}^N : \sum_{i=1}^{N} y_i = \alpha N \right\}. \tag{2.2}$$

In practice, LOE_H may show excessive confidence in assigning y, which can lead to sub-optimal performance. To address this issue, LOE_S is introduced. LOE_S makes a simple adjustment to the constraint set:

$$\mathcal{Y}_0 = \left\{ y \in \{0,0.5\}^N : \sum_{i=1}^{N} y_i = 0.5\alpha N \right\}. \tag{2.3}$$

The result is that an anomaly's presence leads to an equal combination of both losses, $0.5(L^n_\theta(x_i) + L^a_\theta(x_i))$. LOE_S introduces uncertainty in classifying x_i as either a regular or anomalous data point, treating both possibilities with equal likelihood. This approach has demonstrated improved performance on certain datasets.

An example of the latent representations resulting from applying LOE to a Deep SVDD is shown below in Figure 2.4. In a Deep SVDD, the normal loss function is defined as $L^n_\theta(x) = ||f_\theta(x) - c||^2$, which aims to pull normal data towards the center of the distribution. Conversely, the abnormal loss function is defined as $L^a_\theta(x) = \frac{1}{||f_\theta(x)-c||^2}$, which aims to push abnormal data away from the distribution center.

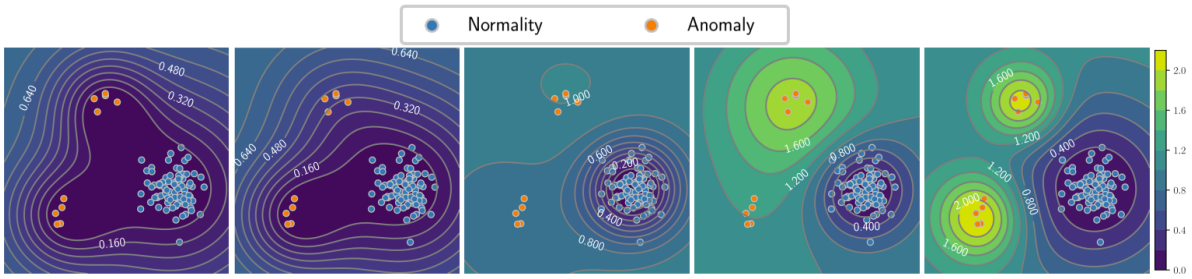


Figure 2.4: SVDD trained using a range of techniques: (a) Blind approach, where all data points are considered normal; (b) "Refine" method, which eliminates specific anomalies; (c) LOE$_S$, which assigns soft labels to anomalies; (d) LOE$_H$, which assigns hard labels; and (e) supervised AD using ground truth labels for comparison. The application of LOE led to improved delineation of region boundaries. [6]

Qiu et al. [6] highlight the versatility of LOE in its application to various AD benchmarks and loss functions. Notably, LOE can outperform results obtained from clean data when trained on contaminated datasets, suggesting that latent anomalies provide valuable learning signals. Furthermore, LOE consistently delivers significant performance improvements across different data types, including images, tabular data, and videos.

2.7 Summary

The LHC generates vast amounts of data in an attempt to unravel the mysteries of BSM physics. Traditional search methodologies for analyzing this data often rely on predefined signal models, resulting in model dependence in both event selection and background estimation. Despite a long history of scientific discoveries using traditional search methods, the recent standstill in progress may indicate that an alternative approach is needed.

Model-independent search paradigms have gained traction in recent times, emphasizing the need for AD techniques that can efficiently navigate the ever-expanding seas of data. However, much of this enthusiasm has been directed toward offline AD approaches, neglecting the enormous Phase space of data deleted by the Trigger system. With advancements in hardware acceleration and model compression techniques, it is now possible to implement AD algorithms directly within the LHC Trigger pipeline, even at the L1T. This shift opens up new avenues for research, enabling real-time AD and potentially uncovering insights from data streams that were previously inaccessible.

Finally, to address the challenge of unlabeled and potentially contaminated training data, novel approaches for training AD systems in the presence of unlabelled anomalies have emerged. One such method, LOE, involves the concurrent optimization of loss functions for both normal and anomalous data. By effectively leveraging both background and signal data, LOE improves a system's ability to generalize and detect previously unseen anomalies, even in the absence of fully labeled datasets.

Overall, this review provides insights into the evolving landscape of AD methodologies in HEP, highlighting the potential for innovative techniques to revolutionize data analysis at the LHC and pave the way for new discoveries in fundamental physics.

Chapter 3

Development of relevant theory

3.1 The Trigger system

At the LHC, proton beams intersect at a rate of 40 million times per second, with each crossing potentially generating multiple collision events. Each event results in several proton pairs colliding, producing numerous particles that are detected by the sensors at the center of each hall. The detectors, equipped with an array of sensors, capture the particles' emergence as electronic signals, accumulating an overall data volume of approximately $\mathcal{O}(1\text{ MB})$ per event. This leads to a data throughput of about 40 TB/sec, which is far too large to be processed, reconstructed, and analyzed directly [62]. To manage the immense data volume, advanced detectors like ATLAS implement real-time data processing systems that filter out all but a small fraction of events. This reduces the event rate to approximately 1 kHz, which is manageable for downstream computing resources [16]. Consequently, only a maximum of 0.25% of beam crossings are selected for further analysis [93].

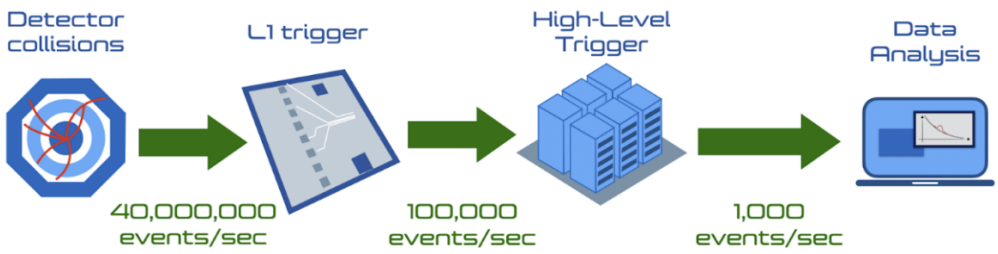


Figure 3.1: An overview of the flow of real-time data processing in ATLAS experiment, from the L1T to the output of the HLT [16]

This filtration system is known as the Trigger. As shown in Figure 3.1, the Trigger comprises two stages of selection. The first stage is known as the L1T. The L1T utilizes algorithms implemented as logic circuits on specialized electronic boards equipped with FPGAs. This stage reduces the data rate to 100 kHz by rejecting over 98% of events. As of July 2024, the target Trigger rate for the L1T system has been set to 85 kHz, as detailed in [94]. Given the limited buffer capabilities and the short 25 ns interval between collisions, arising from the 40MHz collision rate, it is necessary for the complete pipeline of L1T algorithms to be executed within $\mathcal{O}(1)$ μs. The second stage is the HLT. The HLT consists of a computer farm that processes events using commercial CPUs, executing hundreds of intricate selection algorithms within a time-frame of approximately $\mathcal{O}(100)$ ms [16]. Additionally, the Central Trigger Processor (CTP) manages the implementation of dead time, a mechanism aimed at restricting the occurrence of nearby level-1 (L1) accepts [95]. For further details, see Ref [96].

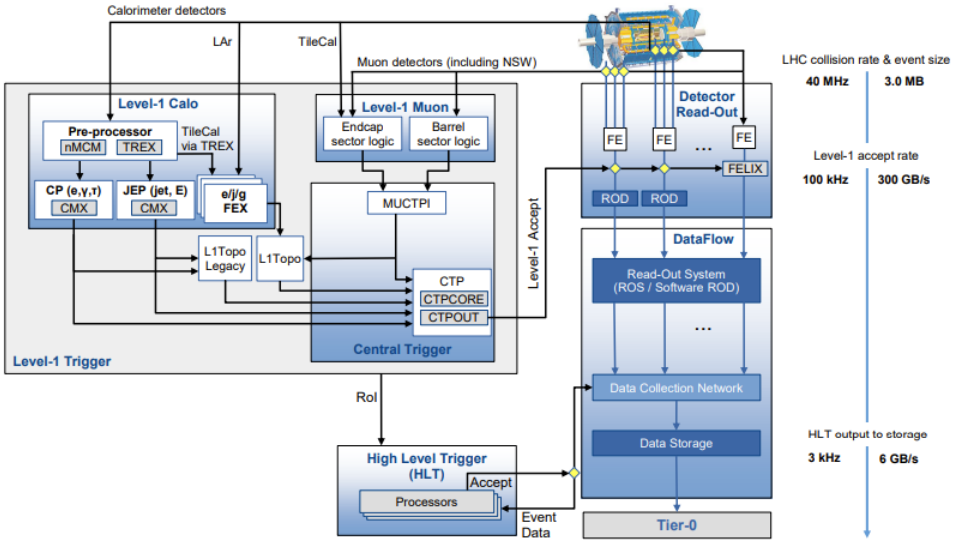


Figure 3.2: The ATLAS TDAQ system in Run 3 with emphasis on the components relevant for triggering as well as the detector read-out and data flow [96]

The Trigger forms the initial stage of data collection in the Trigger and Data Acquisition (TDAQ) system, shown in Figure 3.2. The L1T is largely composed of two independent systems. These systems Trigger on data with reduced granularity originating from either the calorimeters (L1Calo) or the muon detectors (L1Muon) systems, and they are realized using specialized custom electronics. The L1 topological processor (L1Topo) applies topological selections using the kinematic data from the L1Calo and L1Muon systems. The CTP makes the L1 Trigger decision. The determination is guided by inputs received from the L1Calo and L1Muon Trigger systems via the Muon-to-Central Trigger Processor

Interface (MUCTPI) [97], along with inputs from the L1Topo system and various other subsystems.

Selection algorithms in the Trigger system are designed to maximize the acceptance rate for the physics processes under investigation. However, in the absence of strong theoretical guidance, these Trigger systems might inadvertently reject potentially significant events. To address this, contemporary efforts in AD using ML aim to derive a metric directly from LHC data, allowing events to be ranked based on their typicality. By implementing AD at the L1T stage, an unbiased dataset can be presented to the AD algorithm before any event is discarded [16], [98]. This approach could enable the collection of rare event topologies within a dedicated data stream, where the analysis of these anomalous events might lead to the development of new theoretical models in HEP, which could be tested in future data-taking campaigns. While most research has focused on applying this strategy at the HLT stage, deploying it at the L1T level—before any selection bias is introduced—could significantly enhance its effectiveness [16].

The L1T processes coarse-grained data from calorimeters (electrons, photons, jets, MET) and the muon spectrometer (muon candidates), alongside inputs from Minimum Bias Trigger Scintillators for luminosity monitoring. Traditional L1T algorithms rely on fixed kinematic thresholds (e.g., $p_T > 20$ GeV for muons) optimized for known SM processes, introducing an inherent bias against unconventional signatures. This limitation underscores a critical gap: while effective for targeted searches, threshold-based Triggers risk discarding rare or anomalous events that could signal new physics. AD at L1T addresses this by leveraging ML to identify deviations from typical collision data without theoretical assumptions. Unlike conventional Triggers, AD operates model-agnostically, using techniques like AEs to flag outliers in real time, and preserves statistically unbiased events for offline study [99].

To comply with the strict Latency requirements of the L1T, Trigger decision algorithms are implemented in hardware as logic circuits. Integrating ML algorithms into the L1T FPGAs could enhance the complexity and potentially the accuracy of these selection algorithms. The hls4ml library is a software package that enables users to deploy ML models on FPGAs as electronic circuits [13]–[15]. Utilizing hls4ml promotes inference speed, making it well-suited to the constraints of the L1T.

Following these developments, an optimal AD strategy for the L1T is possibly within reach [16].

3.2 hls4ml

hls4ml [13], [37] is an open-source software library designed to facilitate the deployment of ML models on FPGAs, with an emphasis on applications requiring low-Latency and low-power operation at the Edge. By using an ML model as input, hls4ml generates C/C++ code that can be converted into FPGA Firmware via a high-level synthesis (HLS) library. The development of hls4ml was driven by the need to integrate ML into the initial phase of real-time data processing for particle physics experiments at the LHC. To handle the substantial data throughput, LHC experiments employ real-time event selection in the L1T, which necessitates the use of low-Latency ML Firmware.

The architecture of the hls4ml software includes several back-ends, each designed to support different HLS libraries and cater to various FPGA vendors. Currently, the primary focus is on the Vivado HLS [60] back-end, which is specifically tailored for Xilinx FPGAs. hls4ml is compatible with ML models developed using frameworks such as Keras [77], PyTorch [100], and TensorFlow [101].

hls4ml emphasizes deploying neural network architectures entirely on-chip. This approach avoids the Latency overhead associated with transferring data between embedded processing elements and off-chip memory, thereby reducing overall inference Latency. However, this strategy imposes limitations on the complexity and size of models that can be effectively supported by the HLS conversion process.

3.2.1 hls4ml Compression

The hls4ml library offers the capability to define varying numerical precisions for different parts of the network, a technique known as heterogeneous Quantization. This is particularly useful in cases where PTQ applied to activation functions might cause greater accuracy reduction compared to PTQ applied to weights [57]. By default, hls4ml allocates a total of 16 bits per layer, with 6 bits reserved for the integer part. Achieving a high level of compression through PTQ requires balancing compression with accuracy, which depends on the specific application.

The hls4ml library also supports QAT [102] through its interface to QKERAS [15]. In this approach, quantized weights and biases are utilized during the forward pass of training,

while full precision is retained during the backward pass to aid in minimizing the loss towards the optimal point [103]. Aarrestad et al. [37] recommend using QAT through QKERAS for Pruning and Quantization before deploying models with hls4ml on FPGA platforms. Despite slightly lower accuracy compared to baselines, models using QAT require significantly fewer resources. In contrast, PTQ methods, such as ternary and binary Quantization, can result in a drop in model accuracy and increased statistical uncertainty, as the network may lack sufficient information to accurately classify unseen data.

Further, another tool available is AUTOQKERAS [15], which provides a method for performing heterogeneous QAT. AUTOQKERAS performs hyperparameter optimization across varying Quantization conditions, addressing the vast number of configurations in a deep network [104]. It treats layer precision as a hyperparameter and aims to find the Quantization strategy that minimizes the model's bit size while maximizing its accuracy. This optimization process employs a Bayesian strategy to balance accuracy and resource utilization, utilizing a metric derived from both factors [77]. The use of AUTOQKERAS to explore various Quantization configurations for different network components results in an optimally heterogeneously quantized QKERAS model [9].

3.3 Autoencoder-based Anomaly Detection

As introduced in Sections 2.4 and 2.6, autoencoders compress input data into lower-dimensional representations. In this section, we expand on their architecture, training objectives, and how reconstruction error serves as a proxy for anomaly scores.

3.3.1 Autoencoder [105]

Dimensionality reduction in AD aims to find a subspace where normal and anomalous data exhibit distinct characteristics. Suppose we have a normal training set $\{x_1, x_2, \ldots, x_n\}$, where each x_i represents a vector ($x_i \in \mathbb{R}^d$) of dimension d.

During training, a model is developed to map the training data to a lower-dimensional subspace and then reconstruct the original data. The model is optimized to minimize the reconstruction error, striving to achieve the most effective latent space. The reconstruction

error metric is defined as follows:

$$\epsilon(x_i, \hat{x}_i) = \sum_{j=1}^{d} (x_i - \hat{x}_i)^2. \tag{3.1}$$

During the testing phase, normal data in the test dataset conform to the established normal profile from the training phase, resulting in smaller reconstruction errors. Conversely, anomalous data exhibit higher reconstruction errors. Consequently, we can effectively classify anomalous data by applying a threshold to the reconstruction error.

$$c(x_i) = \begin{cases} normal & \epsilon_i < \theta \\ anomalous & \epsilon_i > \theta \end{cases} \tag{3.2}$$

An Autoencoder usually consists of a encoder and decoder network as depicted in Figure 3.3.

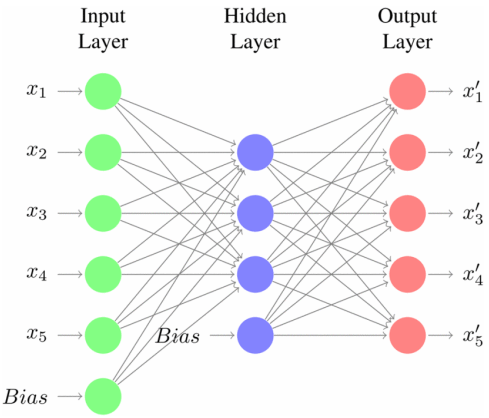


Figure 3.3: Autoencoder network depiction [105]

Encoder: In the encoder network, input vectors $x_i \in \mathbb{R}^d$ are condensed into m neurons in the hidden layer, where $m < d$; seen in Figure 3.3. The i-th neuron's activation in the latent space is expressed as:

$$h_i = f_\theta(x) = s\left(\sum_{j=1}^{n} W_{ij}^{\text{input}} x_j + b_i^{\text{input}}\right). \tag{3.3}$$

In Equation 3.3: x denotes the input vector, θ represents parameters $\{b^{\text{input}}, W^{\text{input}}\}$, W denotes the encoder weight matrix of size $m \times d$, and b is a bias vector with dimensionality m. Consequently, the input vector is transformed into a lower-dimensional vector through encoding.

Decoder: The encoded representation h_i is then decoded, resulting in the original input. This process is defined as follows:

$$x_i = g_{\theta'}(h) = s\left(\sum_{j=1}^{n} W_{ij}^{\text{hidden}} h_j + b_i^{\text{hidden}}\right). \tag{3.4}$$

The decoder's parameter set is denoted as $\theta' = \{b^{\text{hidden}}, W^{\text{hidden}}\}$. The AE is trained to minimize reconstruction error with respect to the parameters θ and θ'. This results in the following optimization problem:

$$\theta^*, \theta'^* = \arg\min_{\theta,\theta'} \frac{1}{n}\sum_{i=1}^{n} \epsilon(x_i, x_i') = \arg\min_{\theta,\theta'} \frac{1}{n}\sum_{i=1}^{n} \epsilon(x_i, g_{\theta'}(f_\theta(x_i))). \tag{3.5}$$

From Equation 3.1 ϵ is the reconstruction error, and the objective is to find the optimal parameters θ^* and θ'^*. Once training is completed, unseen anomalous data can be fed into the network and identified by utilizing the reconstruction error combined with an anomaly threshold.

It is important to note that the activation functions, f and g, must be nonlinear to capture any nonlinear relationships in the data.

3.3.2 VAE [12], [106]

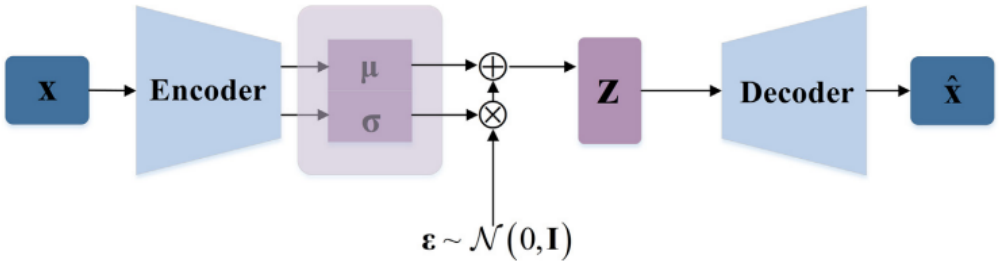


Figure 3.4: VAE Architecture [12]

A notable drawback of utilizing latent space representations in AEs is the potential risk of overfitting, which may lead to the learning of overly sparse representations that fail to generalize effectively. This can reduce the semantic richness of the encoded data, potentially impairing the model's ability to generalize to new inputs that differ from the training data. Therefore, in the context of AD, it is essential to employ a model that

exhibits robustness to input data variations, ensuring that normal fluctuations are not erroneously classified as anomalies.

VAEs, first introduced by Diederik et al. [61], were designed to produce a regularized encoding space that ensures neighboring latent encodings capture similar semantic information. In a VAE, depicted in Figure 3.4, the latent space is modeled as a product of Gaussian distributions, allowing for continuous and probabilistic representations of the latent variables. Essentially, the encoder only needs to learn the parameters, μ and σ, of each Gaussian distribution. During inference and training, the decoder network uses a latent vector sampled from the learned Gaussian parameters to reconstruct the input data. The KL divergence, which quantifies the difference between the estimated and true distributions, is incorporated into the AE loss function, facilitating the regularization of the latent space. This inclusion adds semantic significance to the latent space and ensures efficient encoding [62], [107].

In the context of a dataset $D_n = \{x_1, x_2, ..., x_n\}$, the primary aim of the VAE lies in maximizing the likelihood function $\sum_{i=1}^{n} \log p_\theta(x_i)$ with respect to the encoder's parameters ϕ and the decoder's parameters θ. Here, $\log p_\theta(x_i)$ is defined as:

$$\log p_\theta(x_i) = D_{KL}\left(q_\phi(z|x_i) \| p_\theta(z)\right) + \mathcal{L}(\theta, \phi, x_i). \tag{3.6}$$

Here, $p_\theta(z)$ represents the prior distribution over the latent variables and $D_{KL}(\cdot)$ is the KL divergence. $\mathcal{L}(\theta, \phi, x_i)$ is the evidence lower bound (ELBO) of the datum x_i. As the KL divergence is always non-negative, Equation 3.6 can thus be expressed as:

$$\log p_\theta(x_i) \geq \mathcal{L}(\theta, \phi, x_i). \tag{3.7}$$

Given the intractable marginal likelihood [61], the ELBO is maximized as an alternative estimate of the maximum likelihood $\log p_\theta(x_i)$:

$$\mathcal{L}(\theta, \phi, x_i) = \mathbb{E}_{q_\phi(z|x_i)}\left[\log p_\theta(x_i|z)\right] - D_{KL}\left(q_\phi(z|x_i) \| p(z)\right). \tag{3.8}$$

The first term in Equation 3.8 represents the expected negative reconstruction error from input to output, relying on the sampling of a stochastic latent variable z from the approximated posterior distribution $p_\theta(z|x_i)$. However, traditional back-propagation mechanisms are incompatible with random variables like z [61]. To mitigate this, when $q_\phi(z|x_i)$ follows a Gaussian distribution $\mathcal{N}(z; \mu, \sigma^2)$, each stochastic variable z_i an be expressed as a differentiable function of a noise variable $\epsilon_i \sim \mathcal{N}(0, I)$: $z_{l_i} = \mu_i + r\sigma_i \odot \epsilon_i$.

This is known as the *The reparameterization trick* [81]. This allows the ELBO to be reformulated as:

$$\mathcal{L}(\theta, \phi, x_i) = D_{KL}\left(q_\phi(z|x_i)\|p(z)\right) + \frac{1}{L}\sum_{l=1}^{L} \log p(x_i|z_i^l). \tag{3.9}$$

The prior $p(z)$ is chosen to be isotropic unit Gaussian $\mathcal{N}(0, I)$, where I represents the identity matrix. Maximizing $D_{KL}\left(q_\phi(z|x_i)\|p(z)\right)$ entails aligning the distribution of $q_\phi(z|x_i)$ with that of $p(z)$. Choosing the prior distribution $p(z)$ to be Gaussian, the encoder outputs the mean μ and the standard deviation σ, which parameterize the approximate posterior distribution $q_\phi(z|x) \sim \mathcal{N}(z; \mu, \sigma^2)$. The KL-divergence can therefore be explicitly expressed as:

$$-D_{KL}\left(q_\phi(z|x_i)\|p(z)\right) = \frac{1}{2}\sum\sum_{j=1}^{J}\left(1 + \log((\sigma_i^j)^2) - (\mu_j^i)^2 - (\sigma_i^j)^2\right). \tag{3.10}$$

In Equation 3.10 the dimension of the latent variable z is represented by J. Here, σ_i^j and μ_i^j represent the jth element of the vectors σ_i and μ_i, respectively. The VAE objective function for x_i can be expressed as:

$$\mathcal{L}(\theta, \phi, x_i) = C \cdot \frac{1}{2}\sum_{j=1}^{J}\left(1 + \log((\sigma_i^j)^2) - (\mu_j^i)^2 - (\sigma_i^j)^2\right) + \frac{1}{L}\sum_{l=1}^{L} \log p(x_i|z_i^l). \tag{3.11}$$

The second term corresponds to the reconstruction loss, while the first term measures the discrepancy between the latent prior and the variables generated by the encoder. The reconstruction loss aligns the decoder's output with the original input vector, while the KL divergence term ensures that latent variables remain close to the origin point [106]. These losses are balanced by a suitable parameter C, which allows for tuning their mutual relevance [108]. Prioritizing reconstruction loss may ignore the distribution of the latent space, potentially leading to issues with data generation. On the other hand, emphasizing the KL divergence often results in more disentangled features and a smoother, more normalized latent space, though this can lead to a noisier encoding [109], [110].

In AD, the simplest strategy is to measure the magnitude of the reconstruction loss. After training, normal samples can be effectively reconstructed with minimal reconstruction error. In contrast, anomalies should result in a larger reconstruction loss due to poor reconstruction. However, in the context of a VAE implemented in the L1T, it is not feasible to utilize an reconstruction loss as an AD strategy, as this would necessitate

sampling random numbers on the FPGA. Such an approach would lead to non-deterministic Trigger decisions. Additionally, storing random numbers on the FPGA would consume resources and increase Latency. Researches avoid this by simply using the latent encoding generated by the VAE encoder to identify anomalies [9], [62]. However, the latent space generated by a VAE may be sub-optimal for single class AD.

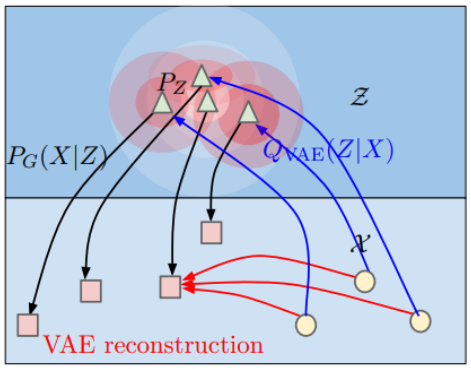
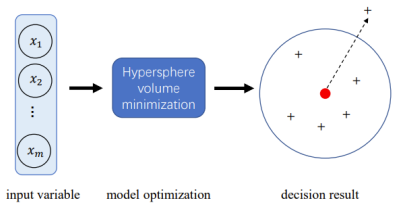


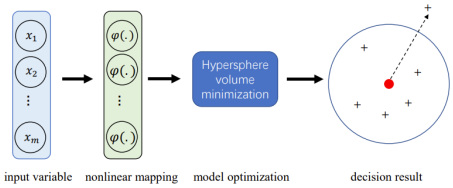
Figure 3.5: The VAE compels $Q(Z|X = x)$ to align with P_Z across all input examples x drawn from P_X. This is shown in Figure (a), where each red ball is adjusted to fit the white shape representing P_Z. As a result, the red balls begin to overlap, causing issues with reconstruction. [111]

Shown in Figure 3.5, the nature of the VAE encourages the emergence of a complex latent distribution, made up of many overlapping distributions rather than a continues mixture. This can make it harder to distinguish between normal and anomalous points, as the latent space may not be well-structured or separate enough for clear classification.

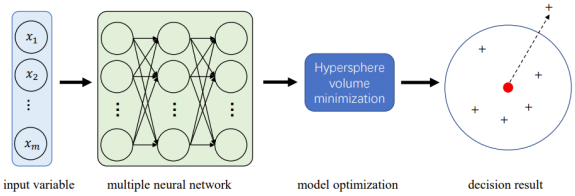
3.4 SVDD



(a) Linear SVDD



(b) Kernel SVDD



(c) Deep SVDD

Figure 3.6: The varying depth profiles for each of the main classes of SVDD [112]

3.4.1 Traditional SVDD models

The SVDD, a variant of the support vector machine, is a widely used one-class classification algorithm [113], [114]. The primary objective of a SVDD is to detect anomalies by defining the smallest possible hypersphere that encompasses the positive samples within the feature space. Traditional SVDDs can be categorized into two main types: linear and kernel-based nonlinear SVDDs. These methods are elaborated upon further below:

3.4.1.1 Linear SVDD

Shown in Figure 3.6a, Given a set of normal training samples denoted as $x_1, x_2, \ldots, x_n$ where $x_i \in \mathbb{R}^m$, $i = 1, 2, \ldots, n$. Linear SVDD optimization attempts to attain a precise representation of the data, as outlined below:

$$\begin{cases} \min R^2 + \frac{\gamma}{n} \sum_{i=1}^{n} \sigma_i, \\ \text{s.t.} \quad ||x_i - o||^2 \leq R^2 + \sigma_i, \\ \sigma_i \geq 0, \end{cases} \tag{3.12}$$

In Equation 3.12, γ serves as a trade-off parameter that balances between the modeling error and the volume of the hypersphere, R represents the radius of the hypersphere and σ_i represents the relaxation variable. The term $\sum_{i=1}^{n} \sigma_i$ acts as a penalty term, accommodating outliers.

3.4.1.2 Kernel SVDD

The optimization described above is applicable solely in the linear scenario. In the presence of a nonlinear data relationship, where the training data is not spherically distributed, a hypersphere cannot effectively isolate anomalies. Hence the need for kernel SVDD.

Shown in Figure 3.6b, In a kernel SVDD a nonlinear mapping function $\phi(\cdot)$ is hypothesized to transform these samples into a new feature space: $x_i \rightarrow \phi(x_i)$, where all samples exhibit a linear relationship. Subsequently, the fundamental SVDD optimization is executed. Further details regarding kernel SVDD can be found in ref [112].

3.4.2 Deep SVDD

To enhance the extraction of data features, Ruff et al. [11] propose deep SVDD (DSVDD), a variant of SVDD structured with deep neural networks, Shown in Figure 3.6c. Analogous to linear and kernel SVDD, DSVDD seeks to identify the smallest possible hypersphere within the feature space. The distinction lies in the employment of deep neural networks to perform more intricate data transformations. Denoting $\Phi(x; W)$ as the transformation of the data through the network, the objective of DSVDD is to minimize the volume of the hypersphere surrounding normal data. The objective, defined by Ruff et al. [11], is as follows:

$$\min_{W} R^2 + \frac{\gamma}{n} \sum_{i=1}^{n} \max\{0, ||\Phi(x_i; W) - c||^2 - R^2\} + \frac{\lambda}{2} \sum_{k} ||W||_F^2. \tag{3.13}$$

In Equation 3.13, λ and γ are balancing parameters and W is the network's weight matrix. If most of the training data is considered normal, Equation 3.13 can be rewritten as the

following simplified optimization [11]:

$$\min_{W} \frac{1}{n} \sum_{i=1}^{n} ||\Phi(x; W) - c||^2 + \frac{\lambda}{2} \sum_{k} ||W||_F^2. \tag{3.14}$$

Equation 3.14 aims to reduce the size of the hypersphere by minimizing the average distance of all training data points to its center, while also incorporating the network weights as a regularization term. Once trained, the network can be utilized for AD. The anomaly score $D(x)$ is defined as the distance to the center of the hypersphere in the feature space:

$$||D(x) = k\Phi(x; W^*) - c||^2. \tag{3.15}$$

Here, W^* represents the weights of the trained network. If a data point is deemed abnormal, the corresponding anomaly score will be higher; otherwise, it will be considered normal.

3.5 SVDD Autoencoder

Hojjati et al. [88] introduce the DASVDD, a method that simultaneously optimizes the parameters of an autoencoder and minimizes the volume of a bounding hypersphere around its latent space representation. The following section provides a detailed exploration of the DASVDD implementation.

3.5.1 DASVDD Anomaly Score and Objective Function

Let the encoding functions of the AE be represented as $h(\cdot)$ and $g(\cdot)$, with θ_e and θ_d denoting their respective training parameters. Given an input x, the encoder computes its latent representation $z = h(x; \theta_e)$. Subsequently, the output reconstruction is obtained as $\hat{x} = g(z; \theta_d)$. The AD metric is formulated to incorporate both the distance of a latent representation from the center of the hypersphere and the reconstruction error. For an input x, the AD metric $S(x)$ is expressed as:

$$S(x) = ||\hat{x} - x||^2 + \gamma||z - c^*||^2 = ||g(h(x; \theta_e^*); \theta_d^*) - x||^2 + \gamma||h(x; \theta_e^*) - c^*||^2. \tag{3.16}$$

Where c represents the hypersphere center and γ serves as a hyperparameter that balances the contribution of the two terms. The asterisk symbol * denotes the optimal value of the

parameter. Note that Equation 3.16 is divided into two terms: the first term represents the reconstruction error, and the second term is the SVDD error. It is crucial not to set γ to an excessively high value. Doing so ensures that the reconstruction error term retains a meaningful contribution and prevents the hypersphere from collapsing.

The network aims to minimize the AD score when processing normal data. Therefore, the objective function is set to be equal to the AD score. This is defined below, where n denotes the batch size:

$$\min_{\theta_e,\theta_d,c} \frac{1}{n}\sum_{i=1}^{n} ||g(h(x_i;\theta_e);\theta_d) - x_i||^2 + \gamma||h(x_i;\theta_e) - c||^2. \tag{3.17}$$

In Equation 3.17, weight decay regularization can be incorporated by adding a term $\lambda|\Theta|_F$. Here, λ is the hyperparameter representing the weight decay, and Θ denotes the matrix formed by concatenating the weights of both the decoder and encoder.

The SVDD term penalizes the hypersphere radius, and minimizing this term effectively reduces the average distance of samples from the hypersphere center c, thereby shrinking the hypersphere containing normal data points. The incorporation of the reconstruction error term in the objective function of the DASVDD framework mitigates the risk of the hypersphere collapsing to zero by preventing all weights from being set to zero solely to minimize the hypersphere's volume.

By projecting the data onto a latent representation near the center c, the network captures the shared factors of variation in normal data. Given that anomalies have distinct characteristics, the network should struggle with reconstructing anomalies and/or place them further from the center of the hypersphere in the latent space.

Chapter 4

Methodology

4.1 Dataset

With the goal of stimulating a community-based effort in fast ML for the L1T, Govorkova et al. [16] developed a dataset that mimics typical data seen by the L1T, pre-filtered by mandating the presence of at least one electron or Muon (referred to as a lepton filter), this dataset offers opportunities for devising innovative event selection methods and evaluating their capability to detect new phenomena.

4.1.1 Physics Content of the Dataset

Collisions between protons at the LHC have the potential to generate and detect a wide array of processes anticipated by the SM [115]–[117]. A concise overview of the particle content of the SM is detailed in Refs. [118], [119]. The SM predicts the rate of each physics process which is subsequently verified through experimental measurements [120]. Through this robust understanding of expected physics processes, realistic physics datasets can be created.

This dataset targets events that involve electrons (e) and Muons (μ), light particles, that along with taus (τ) and their neutrinos, comprise the three lepton families. Although it may have been feasible to consider a dataset without any filtering, such an approach would require computational resources exceeding current technological capabilities for

data generation. Therefore, a lepton filter is adopted [9]. In the confines of a standard LHC detector, electrons and Muons stand as stable particles, observable directly as they traverse the detector components without undergoing decay. In contrast, τ leptons, being substantially heavier than electrons and Muons, swiftly decay into different particles. A fraction of these decays yield electrons and Muons. Within the LHC, the primary source of high-energy leptons lies in the generation of W and Z bosons [121], which rank among the heaviest particles in the SM. Upon their creation, these bosons promptly decay into other particles, including leptons. The production of W and Z bosons predominantly occurs via direct proton collisions. These processes constitute a significant portion of the dataset.

The decay process of top quarks (t) and anti-quarks ($\bar{t}$) give rise to a significant proportion of W bosons. Given the high mass and instability of the top quark, it rapidly decays into a W boson and a bottom quark, creating observable signatures featuring either Jets exclusively or one of the leptons e, μ, or τ, accompanied by a neutrino and a Jet. Leptons also emerge from less common W and Z production channels, the occurrence rates are comparatively low that these avenues are excluded from the dataset [16].

A significant source of leptons arises from the generation of gluons and light quarks, described by the theory of Quantum Chromodynamics (QCD) [122]. Due to color confinement, and the net color charge possessed by each of the quarks and gluons, they are incapable of existing independently; rendering direct observation impossible. Instead, through the process of hadronization, they amalgamate to form color-neutral hadrons resulting in the formation of a collimated spray of hadrons termed as a Jet. Although leptons are seldom generated within Jets, rather primarily through the decay of unstable hadrons, it's noteworthy that in the LHC QCD multijet production stands as the predominant process, thus making a significant contribution to the design of the dataset [16].

4.1.2 Dataset description

This study replicates the setup as found in Refs. [9], [98], [123]. The physics processes outlined above are the primary constituents of the generated e or μ data stream. Each sample in the dataset corresponds to a standard proton-proton collision. The dataset is pre-filtered based on the transverse momenta p_T as well as pseudorapidity η. The specific selection criteria are outlined in further detail in Section 4.1.3. In the real-world the

p_T requirement implemented will likely not be mandated. However the η requirements remain consistent in real-world applications, as they are inherent consequences arising from detector geometry.

The dataset incorporates the following SM processes, the percentage contribution of which is indicated in parentheses. The following list is directly extracted from Ref. [16]:

- Inclusive W boson production, where the W boson decays to a charged lepton (ℓ) and a neutrino (ν), (59.2% of the dataset). The lepton could be an electron (e), a Muon (μ), or a tau (τ) lepton.
- Inclusive Z boson production, with $Z \to \ell\ell$ ($\ell = e, \mu, \tau$) (6.7% of the dataset),
- $t\bar{t}$ production (0.3% of the dataset), and
- QCD multijet production (33.8% of the dataset).

The amalgamation of these samples generates a plausible L1T data stream populated with established SM processes (when combined, this steam is referred to as the *background*). Further, a second dataset of novel lepton-production phenomena are provided (referred to as the *signal*). These phenomena involve theoretical but as-yet-unseen particles, offering theoretically motivated irregularities to assess the performance of an AD system. More details about these phenomena can be found in Refs. [98], [124]. The following breakdown of the dataset is extracted from Ref. [9]:

- A leptoquark (LQ) with a mass of 80 GeV, decaying to a b quark and a τ lepton [125],
- A neutral scalar boson (A) with a mass of 50 GeV, decaying to two off-shell Z bosons, each forced to decay to two leptons: $A \to 4\ell$ [126],
- A scalar boson with a mass of 60 GeV, decaying to two tau leptons: $h^0 \to \tau\tau$ [127],
- A charged scalar boson with a mass of 60 GeV, decaying to a tau lepton and a neutrino: $h^{\pm} \to \tau\nu$ [128].

There are 8 million background events in the dataset. 4 million are used to define the training dataset. The remaining event are combined with new physics processes, to

generate a blackbox [129] dataset. When training non-contaminated models: half the events make up the training set, 40% make up the test set, and the last 10% constitute the validation set. The new physics scenarios are used for performance evaluation. During the training of contaminated models, background and signal events are mixed, with a fraction α of the dataset consisting of anomalous BSM data, while maintaining identical proportional splits across training, testing, and validation sets. A key distinction of this study, in contrast to previous research conducted on this dataset, is the implementation of Unsupervised training on a mixed dataset, further justified below.

In this work, a threshold of $\alpha = 1\%$ was selected for evaluating model performance. This choice was motivated by several considerations. First, it ensures that the models are calibrated to anticipate low rates of anomalies, thereby ensuring the models are exposed to a data environment comparable to the L1T. True anomalies in the L1T are anticipated to be considerably rarer than 1%. Nonetheless, the capability to generate a dataset that contains unbiased, unseen, and potentially anomalous events, while conforming to the L1T Trigger acceptance rate, is invaluable for physics research.

Second, specifically for the LOE based models, incorporating an expected rate of anomalies encourages the model to prioritize the identification of infrequent events within the data stream, rather than merely learning the underlying background data manifold as traditional AE models do. This strategic shift enhances the model's capacity to differentiate and categorize rare anomalies from more prevalent data patterns, thereby improving its effectiveness in AD tasks. Moreover, this approach facilitates the construction of the latent distribution in a way that favors the selection of rare occurrences, at the specified rate, while ensuring that the Trigger rate does not exceed the established limit of 2%. Ultimately, this strategy is designed to optimize the model for unbiased AD within the L1T environment. Additionally, it is anticipated that this methodology will bolster the model's selectivity in defining what constitutes an anomaly, which is particularly critical when dealing with unfiltered L1T data.

4.1.3 Event structure

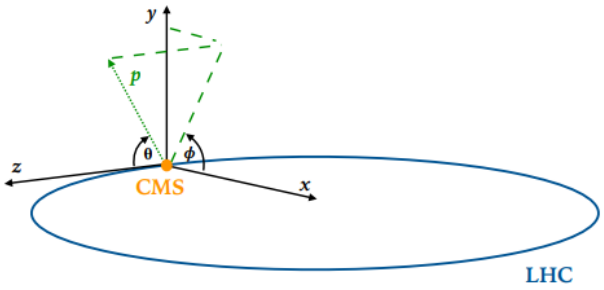


Figure 4.1: The coordinate system used to define the momentum of the particles in the dataset. [16]

As shown in Figure 4.1, the dataset employs a Cartesian coordinate system. The x and y axes form the transverse plane, with the z-axis aligned along the beam direction. The Azimuthal angle ϕ is measured relative to the x-axis and is expressed in radians within the interval $[-\pi, \pi]$. The Pseudorapidity η, is computed using the Polar angle θ, where $\eta = -\log\left(\tan\left(\frac{\theta}{2}\right)\right)$. The Transverse momentum p_T represents the component of the particle's momentum projected onto the (x, y) plane.

The events in the dataset are characterized by a list of the aforementioned four-momenta corresponding to high-level reconstructions; These reconstructions include: electrons, Muons, and Jets. To simulate the typical Bandwidth constraints in the L1T, each event is limited to the first 4 electrons, 4 Muons, and 10 Jets. The events are then arranged in descending order according to p_T. If there are less than 18 particles in the event, zero-padding is used to maintain the input size, mimicking real-world L1T systems. Each item in an event is described by its p_T, η, and ϕ. Additionally, the absolute value and ϕ coordinates of the Missing Transverse Energy are considered.

For the inclusion of an event in the dataset, the following is mandated: the presence of an electron or Muon with $p_T > 23$ GeV, $|\eta| < 3$ for the electron , and $|\eta| < 2.1$ for the Muon. Additionally, Jets must have a $p_T > 15$ GeV with $|\eta| < 4$. Further, events may contain four Muons with $|\eta| < 2.1$ and $p_T > 3$ GeV, and four electrons with $|\eta| < 3$ and $p_T > 3$ GeV. These criteria ensure that the SM processes in the dataset offer a realistic representation of L1T data.

4.1.4 Data records

The entire dataset contains six data entries: one entry comprising the blend of SM processes, four distinct entries for each of the BSM processes mentioned earlier, and one entry for the blackbox data. These are detailed in Table 4.1, along with the total event counts. The data entries are publicly available on Zenodo [125]–[130].

Sample name	Number of samples	Type
SM processes	4,000,000	B
LQ $\rightarrow$ bτ	340,544	S
A $\rightarrow 4\ell$	55,969	S
$h^0 \rightarrow \tau\tau$	691,283	S
$h^{\pm} \rightarrow \tau\nu$	760,272	S
blackbox	4,210,492	S+B

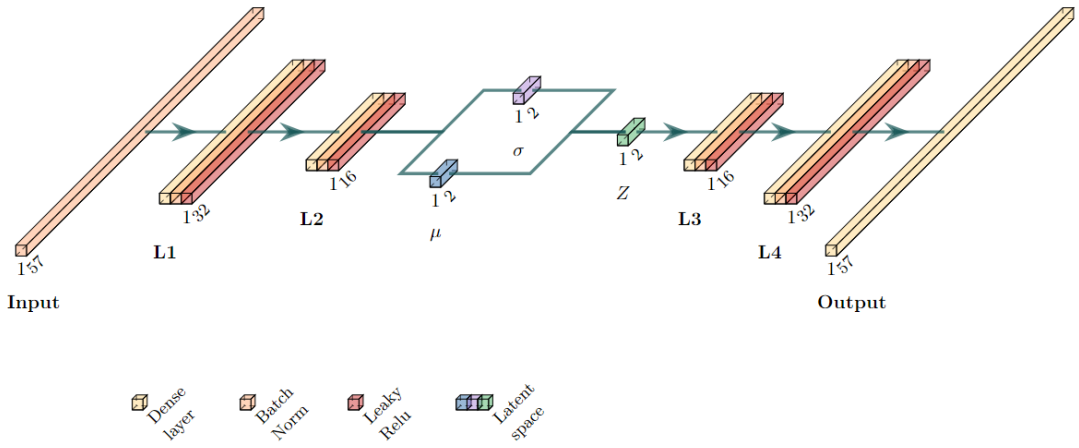
Table 4.1: The names and corresponding number of collision events. Signal and background are denoted by S and B respectively [16]

4.2 Model design

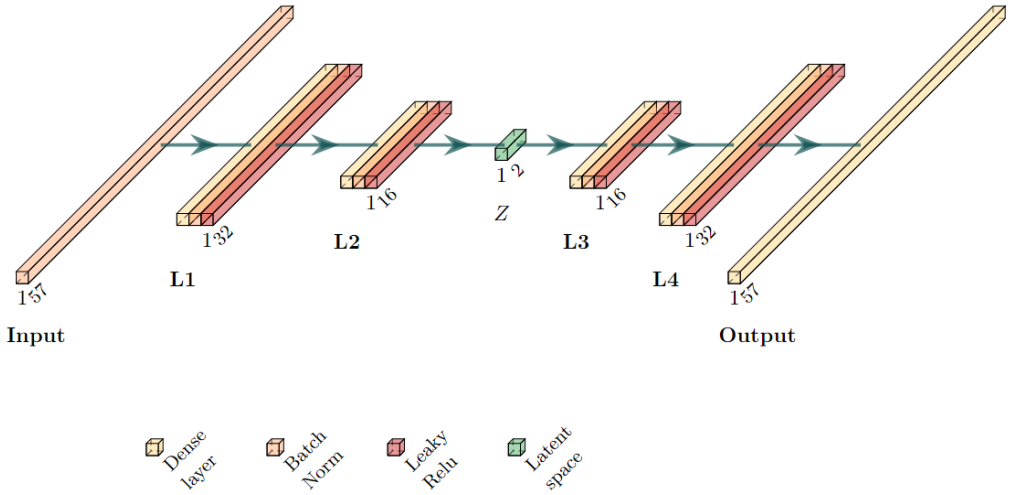
We examine two categories of architectures: non-contaminated and contaminated models. Contaminated models implement LOE, while non-contaminated models do not. Identical inputs are fed into each model, specifically the (p_T, η, ϕ) values for 18 reconstructed objects as mentioned in the previous section. The input initially has a shape of $(19, 3)$, to make this work with the DNN architecture, the four-vector of each reconstructed object is flattened and concatenated into a 1D array, resulting in a 57-dimension input vector.

4.2.1 Non-contaminated

Prior to investigating contaminated models. A standard VAE, and, SVDD based AE were tested on a mixed dataset. This provided a performance baseline for comparison against the models using LOE. The standard AE models were implemented as both DNNs and CNNs, however, like Govorkova et al. [9]; no performance benefit from the use of CNNs was found. Therefore, to minimize model Bandwidth, DNNs are the focus of this research. A latent dimension of 2 is chosen to aid in visualising the latent distribution. To address resource consumption and Latency during data pre-processing, batch normalization [131] is incorporated as the as the initial layer in each model.



(a) VAE



(b) DASVDD

Figure 4.2: Model architectures for a) VAE and b) DASVDD, detailing the dimensions of each layer and the implemented activation functions.

In the DASVDD), the encoder returns the coordinates of the latent space encoding. In the case of the VAE, the encoder yields the standard deviation $\tilde{\sigma}$ and mean $\tilde{\mu}$ of a N-dimensional Gaussian distribution. The distribution represents the probability density function in the encoded space associated with a given event. To achieve this, the DASVDD implements a single 2-dimensional (2D) dense layer in the latent space, while the VAE adheres to the standard VAE approach: two 2D fully connected layers generate the $\tilde{\mu}$ and $\tilde{\sigma}$ vectors, which are then used to sample Gaussian latent variables.

The architectures of the DNN-VAE and DASVDD are depicted in Figures 4.2a and 4.2b, respectively. Both models following an identical structure, with the exception of differences in the latent space. Following the implementation by Govorkova et al. [9], all inputs undergo batch normalization and traverse through a sequence of fully connected layers, with 32 and 16 nodes respectively, before reaching the encoding space. Each

layer's output is batch normalized, followed by the leaky ReLU activation function [132]. The latent layers and the final output layer do not use activation functions. The encoder generates a 2D output, representing the projection of the input in the latent space. This projection varies slightly in structure across different implementations, as previously discussed. Employing a 2D latent space enables the visualization and detailed analysis of the distribution of latent encodings, facilitating the study of their underlying structure and relationships through graphical representation. After encoding, the decoder network in both models comprises fully connected layers with 16 and 32 nodes, respectively, followed by an output layer of dimension 57. Batch normalization and leaky ReLU activations are also applied.

Each model is constructed using TensorFlow [101] and trained on a contaminated dataset with contamination ratios ranging from $\alpha = 0.005$ to $\alpha = 0.05$. The Adam optimizer [81] is employed for training. The implemented loss function for the VAE is expressed as follows:

$$L = (1 - \beta)\text{MSE}(\text{Output}, \text{Input}) + \beta\text{DKL}(\tilde{\mu}, \tilde{\sigma}). \tag{4.1}$$

MSE signifies the reconstruction loss, while DKL stands for KL regularization [133]. The KL divergence term, is defined as:

$$\text{DKL}(\tilde{\mu}, \tilde{\sigma}) = -\frac{1}{2}\sum_{i}\left(\log(\sigma_i^2) - \sigma_i^2 - \mu_i^2 + 1\right). \tag{4.2}$$

Here, $\tilde{\sigma}$ and $\tilde{\mu}$ denote the standard deviation and mean vectors, respectively. β is a hyperparamter constrained within the interval [0, 1] [109].

The resulting loss function for the DASVDD follows a similar structure to the VAE:

$$L = (1 - \beta)\text{MSE}(\text{Output}, \text{Input}) + \beta\text{DSVDD}(\tilde{c}, \tilde{z}). \tag{4.3}$$

The MSE is the Reconstruction term, while the DSVDD term represents the SVDD regularization [88]. This is defined as follows:

$$\text{DSVDD}(\tilde{c}, \tilde{z}) = ||z_i - c_i||^2. \tag{4.4}$$

Here, $\tilde{c}$ denotes the center of the enclosing hypersphere in the latent representation, while $\tilde{z}$ is the encoded latent vector. β is again a hyperparamter constrained within the interval [0, 1].

Both models undergo training for 100 epochs, utilizing a batch size of 1000. Early

stopping is applied, ending training if no improvement in loss is noted after ten epochs. Training across all models is executed with floating-point precision on an NVIDIA V100 GPU.

4.2.2 Contaminated [6]

Prior the development of the LOE-based models, a variety of alternative designs were systematically investigated and subsequently dismissed for various reasons. A detailed description of these models, including their performance metrics and the rationale for their exclusion, is presented in Appendix B.

To incorporate an objective function rooted in AD principles, LOE is integrated into the non-contaminated models. As discussed in the literature review, for the contaminated models, two losses are implemented. A loss function $L_\theta^n(x) \equiv L_n(f_\theta(x))$ minimized over "normal" data, where $f_\theta(x)$ acts as a feature extractor. A second loss for anomalies $L_\theta^a(x) \equiv L_a(f_\theta(x))$, sharing the feature extractor, is also introduced. The resulting joint loss functions is as follows:

$$L(\theta, y) = \sum_{i=1}^{N} (1 - y_i) L_\theta^n(x_i) + y_i L_\theta^a(x_i). \tag{4.5}$$

Given the objective of utilizing both losses L_θ^n and L_θ^a to discern and assess anomalies. $L_\theta^n(x_i) - L_\theta^a(x_i)$ is expected to be large when encountering anomalies, while $L_\theta^a(x_i) - L_\theta^n(x_i)$ should be large for normal data. To optimize these losses with respect to θ, training the AD model involves minimizing Equation 4.5 over y, where y represents the assignment variable. Considering the constraints imposed by LOE_H leads to the following minimization problem:

$$\min_\theta \min_{y \in Y} L(\theta, y). \tag{4.6}$$

This work introduces LOE as a systematic approach to address the intricate problem of AD at the LHC. The LOE methodology enhances AE-based AD models by adapting them for unsupervised learning on unlabelled datasets that may contain a small, unidentified proportion of anomalies. Throughout training, LOE detects anomalous samples and improves prediction accuracy by solving a hybrid continuous–discrete optimization task. This process iteratively refines both the model and the anomaly labels, encouraging the latent representation to distinguish anomalous inputs from the primary data cluster and

structuring the latent space to facilitate single-class AD. Moreover, LOE-shaped latent distributions may help counteract the *look-elsewhere effect* by influencing where anomalies group in the encoding space, thus decreasing the likelihood of spurious outliers.

The central goal of this work, through the LOE framework, is to design AD models with heightened sensitivity to anomalous patterns, optimizing the true positive rate (TPR) while maintaining a fixed false positive rate (FPR) at a stringent level of 10^{-5}. This low FPR threshold is crucial, as it limits the expected background to around 1000 events per month in the target dataset [9], ensuring that detected anomalies are highly significant and reducing the chance of overwhelming the system with false alarms. Balancing high anomaly detection performance with minimal false positives supports effective operation under the demanding real-time conditions of the LHC. By focusing on this balance and limiting dependence on prior data knowledge, the models are designed to deliver robust and controlled Unsupervised AD performance in live experimental environments, where anomalies are both rare and critical to identify accurately.

4.2.2.1 Model architecture

The architecture of the LOE infused models (contaminated models) remains the same as the non-contaminated models, shown in Figure 4.2. However, as mentioned above, a second loss must be introduced. For the VAE, the loss function remains the same when encountering "normal" data. However, in the case of anomalies, a variation of the KL-divergence is introduced: DKL_A. The reconstruction error remains the same for both anomalies and "normal" data ensuring relevant features are extracted from the data. The DKL_A regularization aims to push anomalous samples away from the distribution of normal data by replacing the μ_i^2 in the KL divergence, from Equation 4.2, with $1/\mu_i^2$. This is summarised below in Equation 4.7, where $y_i = 1$ represents an anomaly:

$$L = \begin{cases} (1-\beta)\mathrm{MSE}(\mathrm{Output}, \mathrm{Input}) + \beta \mathrm{DKL}(\tilde{\mu}, \tilde{\sigma}), & \text{if} \quad y_i = 0 \\ (1-\beta)\mathrm{MSE}(\mathrm{Output}, \mathrm{Input}) + \beta \mathrm{DKL}_A(\tilde{\mu}, \tilde{\sigma}), & \text{if} \quad y_i = 1 \end{cases} \tag{4.7}$$

where:

$$\mathrm{DKL}_A(\tilde{\mu}, \tilde{\sigma}) = -\frac{1}{2} \sum_i \left(\log(\sigma_i^2) - \sigma_i^2 - \frac{1}{\mu_i^2} + 1 \right). \tag{4.8}$$

The same process is applied to the DASVDD where DSVDD_A is the inverse of the DSVDD requlariza tion applied in Equation 4.4. This is shown below in Equation 4.9:

$$L = \begin{cases} (1-\beta)\text{MSE}(\text{Output}, \text{Input}) + \beta\text{DSVDD}(\tilde{c}, \tilde{z}), & \text{if} \quad y_i = 0 \\ (1-\beta)\text{MSE}(\text{Output}, \text{Input}) + \beta\text{DSVDD}_A(\tilde{c}, \tilde{z})), & \text{if} \quad y_i = 1 \end{cases} \tag{4.9}$$

where:

$$\text{DSVDD}_A(\tilde{c}, \tilde{z}) = \frac{1}{||z_i - c_i||^2}. \tag{4.10}$$

Finally a third LOE model is introduced called the *Shifty VAE*, shown below in Figure 4.3. This model introduces a third learnable parameter δ into the latent space.

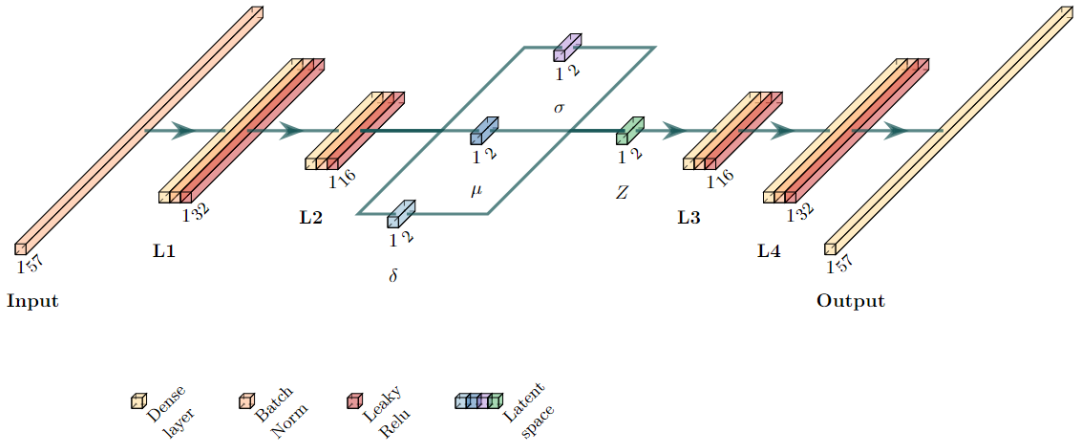


Figure 4.3: Architecture of *Shifty VAE*, detailing the dimensions of each layer and the implemented activation functions.

The parameter δ is responsible for adjusting the mean of the sampling Gaussian distribution based on whether each datum is classified as an anomaly by the model. This approach aims to encourage the model to encode anomalous and normal data in distinct regions of the latent space.

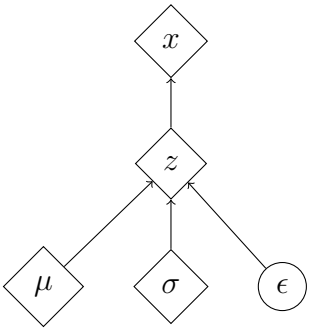


Figure 4.4: Reparameterization trick in VAE [61]

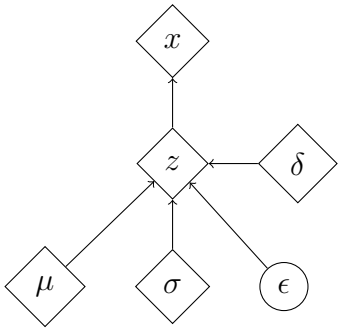


Figure 4.5: Reparameterization trick in Shifty VAE

As depicted in Figures 4.4 and 4.5, the following adaptation to the reparameterization trick is introduced when converting the standard VAE to the *Shifty* VAE: The original equation $z_i^l = \mu_i + \sigma_i \odot \epsilon_i$ is modified to include a shift applied to the mean, resulting in the new formulation:

$$z_i^l = \delta_i + \mu_i + \sigma_i \odot \epsilon_i. \tag{4.11}$$

This results in the following LOE loss:

$$L = \begin{cases} (1-\beta)\text{MSE}(\text{Output}, \text{Input}) + \beta((1-C)\text{DKL}(\tilde{\mu}, \tilde{\sigma}) + C\text{DS}(\tilde{\delta})), & \text{if} \quad y_i = 0 \\ (1-\beta)\text{MSE}(\text{Output}, \text{Input}) + \beta((1-C)\text{DKL}_A(\tilde{\mu}, \tilde{\sigma}) + C\text{DS}_A(\tilde{\delta})), & \text{if} \quad y_i = 1 \end{cases} \tag{4.12}$$

where:

$$\text{DS}(\tilde{\delta}) = ||\delta_i - shift||^2, \quad and \quad \text{DS}_A(\tilde{\delta}) = ||\delta_i + shift||^2. \tag{4.13}$$

DS aims to shift the distribution of the background data towards the positive quadrant of the latent space, while DS_A aims to encourage anomalous data to be encoded in the negative quadrant. C and β are hyperparamters constrained within the interval [0, 1]. The resulting distribution should facilitate the use of a straightforward AD metric while ensuring that a high rate of anomalies is sampled during Triggering, rather than a mixture of normal and anomalous data.

4.2.2.2 Block Coordinate Descent [6]

To achieve discrete optimization with shared loss function, a sequence of parameters θ^t and corresponding labels y^t are considered; employing alternating updates. When updating θ, y^t is fixed and $L(\theta, y^t)$ is minimized over θ.

To update y given θ_t, the same function in minimized subject to the label constraint of LOE_H: $Y = \left\{ y \in \{0,1\}^N : \sum_{i=1}^N y_i = \alpha N \right\}$. Therefore the training anomaly scores are defined as:

$$S_{\text{train}_i} = (\gamma) L_\theta^n(x_i) - (1-\gamma) L_\theta^a(x_i). \tag{4.14}$$

Where γ is a hyperparamter constrained within the interval [0, 1]. The S_{train_i} scores measure the impact of y_i on minimizing Equation 4.5. These scores are ranked and the corresponding labels y_i are assigned. The value 0 is given if the scores fall in the $(1-\alpha)$-quantile. The remaining highest scores are assigned the value 1. This strategy effectively minimizes the loss function while adhering to the label constraint. Assuming

that all involved losses are bounded from below, the block coordinate descent converges to a local optimum since every update improves the loss. The training process is outlined in Algorithm 1.

Stochastic gradient descent is employed to optimize Equation 4.5 using mini-batch processing. The label constraint is applied to each mini-batch, followed by the alternating optimization of the parameters θ and the assignment variable y. The bias introduced in this process diminishes as the size of the mini-batches increases, thereby enhancing the stability of the optimization. Both models are trained for a total of 100 epochs, utilizing a batch size of 100,000, with early stopping implemented to halt training if no improvement in the loss function is observed over a period of ten epochs. All models are executed with floating-point precision on an NVIDIA RTX 2080 GPU.

Algorithm 1: Training Process of LOE [6]

Input : Contaminated training set D (α_0 anomaly rate), Hyperparameter α
Models: Deep anomaly detector with parameters θ
foreach *Epoch* **do**
  foreach *Mini-batch* M **do**
    Calculate the anomaly score S_{train_i} for $x_i \in M$;
    Estimate the label y_i given S_{train_i} and α ;
    Update the parameters θ by minimizing $L(\theta, y)$;
  end
end

4.3 Anomaly detection scores

4.3.1 Non-contaminated

The objective of the DASVDD is to encode information necessary for precise input data reconstruction. Novel features should result in the model struggling to generalize during inference. As such, a prevalent approach in AD involves pinpointing cases where the decoded output diverges notably from the input. A simple strategy is to employ the metric utilized in the training loss function, in this case: the MSE between input and output, known as IO AD [9].

As the DASVDD has a regularised latent space, the latent distribution can too be leveraged to perform AD. In the DASVDD, the model is incentivized to cluster normal data tightly around a sphere-like region in the latent space. Hence, instances can be flagged as anomalous based on their encoded distance from the sphere's center. This leads to the following anomaly metric, where z represents the latent encoding and c denotes the center of the hypersphere:

$$R_{svdd} = \|z_i - c\|^2 \tag{4.15}$$

In the case of the DASVDD, the two metrics can also be combined, giving a third possible AD metric: IO + R_{svdd}

As discussed in the literature review, utilizing the IO AD method with a VAE presents challenges due to the need for deterministic behavior. Furthermore, implementing IO AD would necessitate storing random numbers on the FPGA, leading to resource consumption and increased Latency. To address this, an AD score based on the μ and σ values provided by the encoder is implemented. Specifically, two options are considered: The first option involves utilizing the KL divergence term from the VAE loss, while the second option employs only the mean squared term $\sum_i \mu^2$ of the KL divergence, designated as KL_μ, to accommodate Latency considerations [62].

As Gaussian sampling is circumvented, substantial savings in resources and Latency are achieved through bypassing the evaluation of the decoder. Additionally, there is no necessity to buffer the input data for computing the MSE.

The VAE and DASVDD models, along with the corresponding AD metrics, are specifically

designed to operate with uncontaminated data [9], [88]. The models are trained exclusively on background samples, enabling the bottleneck of the networks to learn a representation that captures the underlying characteristics of the "normal" data. This process facilitates the model's ability to detect instances of data that do not conform to this learned representation. However, in this work, a contaminated dataset is used. This may result in performance losses when applying standard AD techniques to a mixed dataset. LOE is introduced help improve the performance of AD in the presence of unlabeled anomalies.

4.3.2 Contaminated

According to Hojati et al. [88], inferring the most likely label can be employed to detect anomalies, as outlined in the previous section. However, it may be imprudent to assume that anomalies encountered during inference will resemble those present in the training set. Consequently, L_θ^n is utilized as the AD metric instead of L_θ^a. Due to parameter sharing, the joint training of L_θ^a alongside L_θ^n has already facilitated the desired transfer of information between the two loss functions.

Consequently, the LOE VAE and LOE DASVDD models implement the same AD metrics as those used in the non-contaminated models. The *Shifty* VAE adopts the same metrics as the LOE VAE while additionally incorporating the shift loss as a third testable anomaly metric:

$$R_{shift} = ||\delta_i - shift||^2 \tag{4.16}$$

4.3.3 Performance at floating point precision

The model's effectiveness is evaluated using a contaminated mixture dataset containing the four novel physics benchmark models as well as the background SM event data. In this study, the following anomaly detection scores are used:

- IO, R_{SVDD} and IO + R_{SVDD} are used for the DASVDD and the LOE DASVDD,
- KL and KL_μ are used for the VAE and LOEVAE,
- KL, KL_μ and R_{Shift} are used for the *Shifty* VAE

The AUC metric is used to evaluate the performance of each model to provide a single number evaluation of classifier performance [134].

The AD performance is further quantified by finding the TPR corresponding to an FPR working point of 10^{-5}. This FPR working point corresponds to reducing the background rate to approximately 1000 events per month in this dataset [9]. Using Govorkova et al. [9] as a baseline, the state-of-the-art performance of a floating-point model for the TPR corresponding to a FPR working point of 10^{-5} is approximately [1]0.002459, or 0.2459%.

This metric is significant as it is closely related to the objective of implementing LOE within the context of the L1T. The goal of LOE is to enhance the latent space distribution, enabling the sampling of anomalies at lower FPR working points without compromising overall performance. This capability is particularly advantageous in the context of the L1T, where high data rates combined with an elevated FPR could potentially result in uncontrolled Trigger rates [93]. Such scenarios may lead to the saturation of the Trigger system, resulting in the loss of valuable data.

To account for variations in training, each model is subjected to ten independent training iterations. The optimal performance metrics are recorded, specifically focusing on the TPR corresponding to an FPR working point of 10^{-5}. Additionally, the average performance across all trials is documented to account for variation. For enhanced understanding and analysis, the latent distributions of the highest-performing non-contaminated VAE and DASVDD models are analyzed.

The optimal-performing non-contaminated models, are evaluated based on the TPR metric. Subsequently, LOE is applied to these selected models. Finally, a sensitivity analysis is conducted on the two best-performing LOE models.

4.4 Sensitivity analysis

The sensitivity analysis evaluates the robustness of the model under various true and assumed contamination ratios. Specifically, both the true anomaly ratio α_0 and the hyperparameter α are systematically varied within the same dataset. To account for performance variability, each model is trained ten times for each combination of the

[1]Govorkova et al. [9] make use of an identically structured VAE on the same dataset in a supervised learning context.

anomaly ratios, with the best results being recorded. The model exhibiting the highest performance in this sensitivity analysis is subsequently advanced to the stages of compression and FPGA implementation. This decision is informed not only by the TPR metrics at each combination of contamination ratios but also by the consistency of the AUC results and the reliability of the best-performing AD metric.

4.5 Model compression and Firmware synthesis

In the initial stage of compression, the model identified as the best performer from the sensitivity analysis undergoes Pruning. In accordance with the guidelines outlined in the TensorFlow Pruning guide, fine-tuned Pruning is applied to the pre-trained model to optimize its performance while reducing its complexity [135].

The model underwent Pruning for 30 epochs using the same dataset, with a batch size of 100,000. The TensorFlow Pruning guide further recommends focusing Pruning efforts exclusively on dense layers or targeting only non-critical layers. Based on these recommendations, three distinct Pruning methods were tested. The first method involved reducing the connections of all layers by 50%. The second method focused exclusively on Pruning dense layers. The third method entailed Pruning all layers except those representing the latent variables. The results of the Pruning process are compared to the performance of the selected non-pruned model, referred to as the BF model. The TPR of the pruned model at a FPR working point of 10^{-5} is evaluated relative to that of the BF model by calculating the ratio between the two.

The best-pruned model, referred to as the BP model, is selected for PTQ. QAT is not employed, as Govorkova et al. [9] demonstrated that QAT performance exhibited instability as a function of bit width when applied to VAE models. Preliminary investigations indicated that QAT was highly unstable and yielded poor performance, leading to the decision not to implement it fully. The model is subsequently translated into Firmware utilizing hls4ml, which facilitates the implementation of PTQ on the BP model. Upon translation via hls4ml, all performance metrics reported correspond to the model's execution on FPGA Firmware, validated through bit-accurate emulation [136]. In this study, a Xilinx Virtex UltraScale+ VU9P (`xcvu9p-flgb2104-2-e`) FPGA with a clock frequency of 200 MHz is utilized. This FPGA selection aligns with that of Govorkova et al. [9], facilitating direct comparisons of resource utilization and ensuring that resource

consumption remains within the operational constraints of the LHC Trigger system.

A bit precision scan is subsequently conducted to identify the optimal trade-off between Quantization levels and model performance. The bit width is systematically varied from 2 to 16 bits in increments of 2. Additionally, the allocation of bits between the fractional and integer components of the fixed-point representation is adjusted at each specified bit width. Only the optimal ratios between fixed-point and fractional components at each bit width are documented. The TPR of the quantized model at an FPR working point of 10^{-5} is evaluated by computing the ratio of the TPR for the quantized model to that of the BP model.

Again using Govorkova et al. [9] as a baseline, the state-of-the-art performance for the TPR at a FPR working point of 10^{-5} for a model that has been quantized and implemented in Firmware is approximately 0.001999, or 0.1999%.

4.6 Porting models to FPGAs

Once translated into Firmware with PTQ applied. The model is then synthesized with Vivado HLS 2020.1 A summary of the resource consumption and Latency for the synthesized model is then presented, where resource utilization is expressed as a percentage of the total available resources on the FPGA.

Chapter 5

Results and Discussion

5.1 Non-contaminated models

In Table 5.1, multiple DNN architectures are systematically evaluated using critical performance metrics, including the TPR at a fixed FPR of 10^{-5} and the AUC. Additionally, various AD metrics are assessed to provide a comprehensive analysis of model performance. A similar assessment for CNN architectures is provided in Table 5.2. Figures 5.1a–5.1d present the AUC curves of these models, allowing for a visual comparison of their anomaly detection capabilities. Further insights into the models' latent representations are depicted in Figures 5.2 and 5.3, which illustrate the distributions of latent encodings for both background and signal samples. These figures provide a deeper understanding of the models' ability to separate signal from background in their latent space.

5.1.1 Model Performance

As illustrated in Tables 5.1 and 5.2, along with the corresponding AUC plots in Figure 5.1, the performance of DNN models significantly exceeds that of CNN models, with an average improvement factor of 3.5 in the peak TPR at a FPR of 10^{-5}. This substantial improvement highlights the superior effectiveness of DNN architectures within the specific AD context being investigated. This is also true with respect to the average TPR @ FPR 10^{-5} metric, where the DNN models outperformed the CNN models by a factor of 12.5. The AUC metric was comparative between the DNN and CNN models. Given

these results, the CNN models were not explored further. Upon examining Table 5.1, it becomes evident that the DNN VAE exhibits a high level of consistency across the various AD metrics. Notably, the KL and KL_μ metrics demonstrate nearly identical performance in terms of both TPR and AUC, as shown in Figure 5.1a.

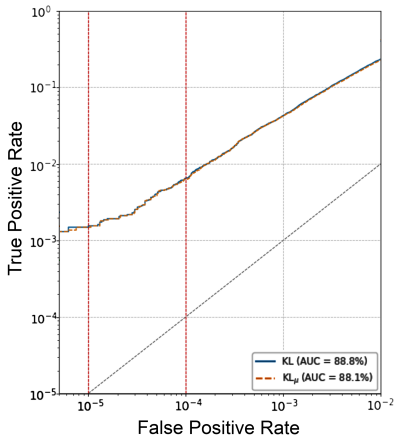
The DNN DASVDD, however, exhibits significantly greater variability across the AD metrics, as illustrated in Figure 5.1b. When the IO AD approach is employed within the DNN DASVDD, it yields the highest peak TPR performance. However, the resulting AUC is inferior to that of any other AD metric applied to the DNN model. The R_{SVDD} + IO and R_{SVDD} metrics perform identically, likely as the IO component effectively renders the R_{SVDD} metric redundant. The average TPR at an FPR of 10^{-5} for the DNN VAE and DNN DASVDD is notably similar, but this similarity is observed exclusively when the IO metric is incorporated. However, when comparing the two models, the DNN VAE KL and KL_μ metrics offer distinct advantages over the DNN DASVDD, which relies on the IO metric for comparative performance. The implementation of either KL metric would substantially reduce inference Latency and on-chip resource utilization, as only the encoder portion of the network needs to be implemented and processed. Additionally, buffering the input to calculate an MSE for the IO metric is rendered unnecessary.

Table 5.1: Performance assessment of the Non-contaminated DNN models

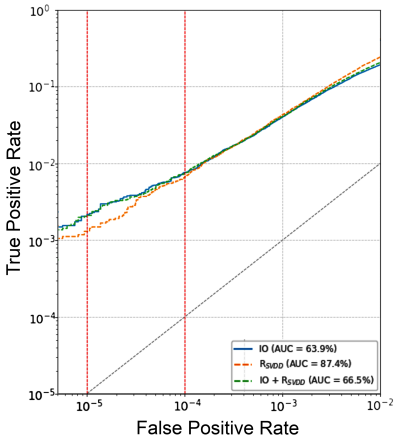
Model		Performance Metric		
Architecture	AD score	TPR @ FPR 10^{-5} [%]	avg. TPR @ FPR 10^{-5} [%]	AUC [%]
VAE	KL	0.1485	0.0835	88.8
VAE	KL_μ	0.1485	0.0835	81.7
DASVDD	IO	0.2042	0.0922	63.9
DASVDD	R_{SVDD}	0.1299	0.0699	87.4
DASVDD	R_{SVDD} + IO	0.2042	0.0922	66.5

Table 5.2: Performance assessment of the Non-contaminated CNN models

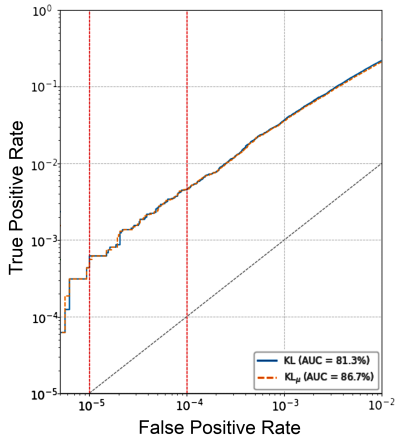
Model		Performance Metric		
Architecture	AD score	TPR @ FPR 10^{-5} [%]	avg. TPR @ FPR 10^{-5} [%]	AUC [%]
VAE	KL	0.0433	0.0161	81.3
VAE	KL_{μ}	0.0433	0.0149	86.7
DASVDD	IO	0.0555	0.0388	85.3
DASVDD	R_{SVDD}	0.0370	0.0481	79.8
DASVDD	R_{SVDD} + IO	0.0555	0.0364	85.3



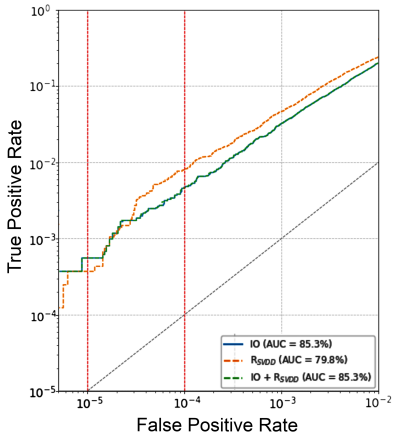
(a) DNN VAE



(b) DNN DASVDD



(c) CNN DASVDD

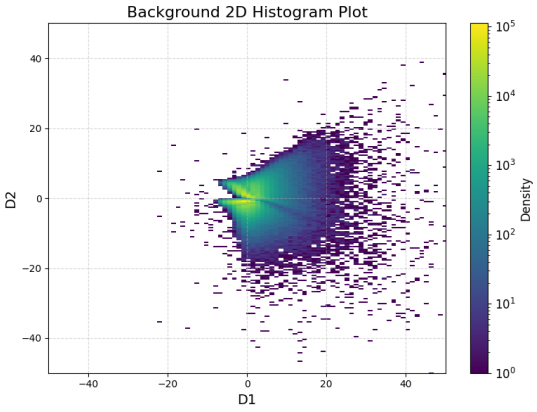


(d) CNN DASVDD

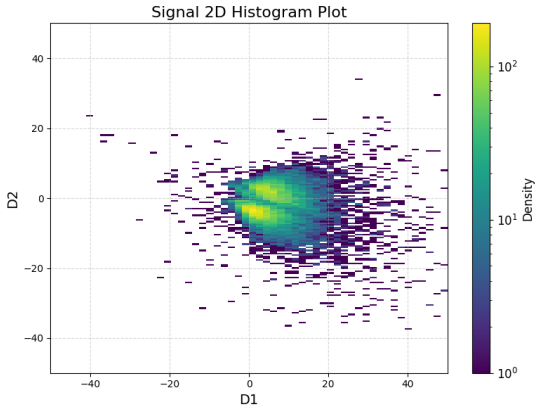
Figure 5.1: AUC curves of non-contaminated models under varying AD metrics. The red line in each plot indicates the TPR @ FPR 10^{-5}. The DNN models are illustrated in the upper half of the figure, whereas the CNN models are displayed in the lower half.

5.1.2 Latent Distributions

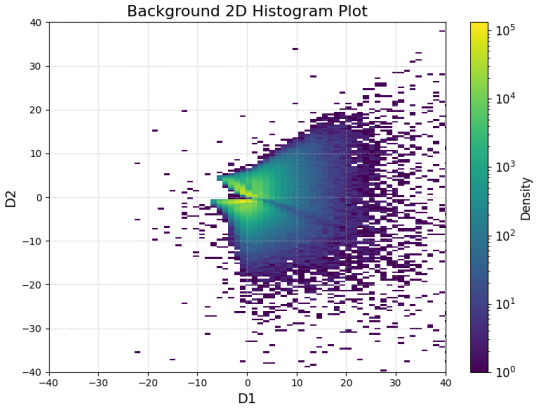
As the top-performing models, the latent distributions of the DNN VAE and DNN DASVDD are presented in Figures 5.2 and 5.3, respectively. Examining the distributions of the z encodings for the DNN VAE (Figures 5.2a and 5.2b) and the DNN DASVDD (Figures 5.3a and 5.3b), it is observed that the distributions exhibit complex shapes, with the majority of encodings clustering around the center. The extent of the high-density region is represented by the bright yellow areas in both sets of plots. This high-density region of the encoding space is significantly larger in the distribution of the signal encodings compared to that of the background for both the DNN VAE and the DNN DASVDD. The DNN VAE distributions exhibit less overlap in the high-density areas between background and signal compared to the DNN DASVDD. This reduced overlap may be associated with the improved TPR at an FPR of 10^{-5} observed when employing an AD metric that incorporates latent outlier detection.



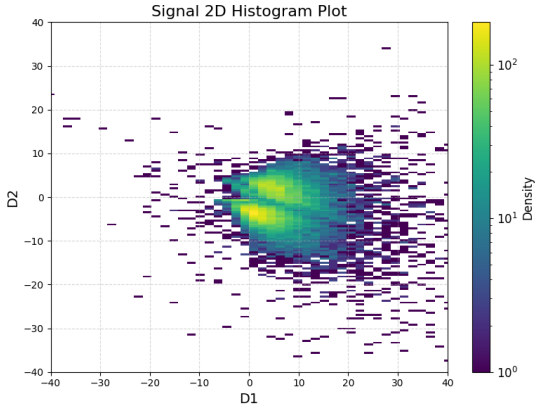
(a) z encodings - Background



(b) z encodings - Signal



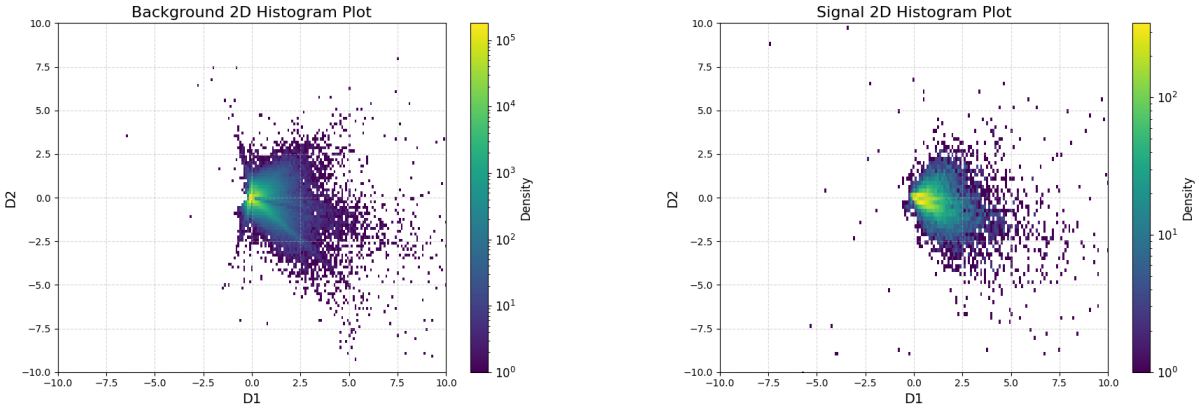
(c) μ encodinsg - Signal



(d) μ encodings - Background

Figure 5.2: Distribution of latent encodings of best performing non-contaminated VAE model with respect to TPR @ FPR 10^{-5} metric (DNN VAE). (a) and (b) show the distribution of the sampled encodings that are passed to the decoder network, while (c) and (d) show the distribution of the encoder mean vectors.

The distribution of the mean vectors of the DNN VAE, depicted in Figures 5.2c and 5.2d, closely aligns with the background and signal distributions of the encoded z vectors in Figures 5.2a and 5.2b. This alignment illustrates why both the KL and KL_μ metrics provide similar performance. At this stage, none of the models achieve the target established in the methodology of a TPR at an FPR of 10^{-5} of 0.2459%. Consequently, the DNN VAE and DASVDD models are converted into contaminated models through the application of LOE. Additionally, the *Shifty* VAE is introduced. The results of these modifications are discussed in Section 5.2.



(a) z encodings - Background

(b) z encodings - Signal

Figure 5.3: Distribution of the latent encodings of best performing non-contaminated DASVDD model with respect to TPR @ FPR 10^{-5} metric (DNN DASVDD). (a) and (b) show the distributions of the encodings passed to the decoder network.

5.2 Contaminated models

The performance of the different LOE AD models are summarized in Table 5.3, which provides a comparative assessment of various architectures and AD scoring techniques under contaminated data conditions. Performance is evaluated based on the best and average TPR at an FPR of 10^{-5}, as well as the AUC. Figure 5.4 illustrates the AUC curves for these models under different AD metrics. Additionally, Figures 5.5 through 5.7 present the latent encodings of background and signal data for each model.

5.2.1 Model Performance

Referring to Table 5.3, the TPR at an FPR of 10^{-5} metric indicates that LOE produced mixed results. Notably, the LOE VAE significantly outperformed its DNN counterpart by at least a factor of 2. Similarly, the *Shifty* VAE demonstrated an improvement over the standard DNN VAE by a comparable margin; however, the *Shifty* VAE the is sensitive to the choice of AD metric. Conversely, the LOE DASVDD exhibited inferior performance compared to the original DNN DASVDD across all metrics, except for the AUC metric. Consequently, further discussion of the LOE DASVDD will be limited.

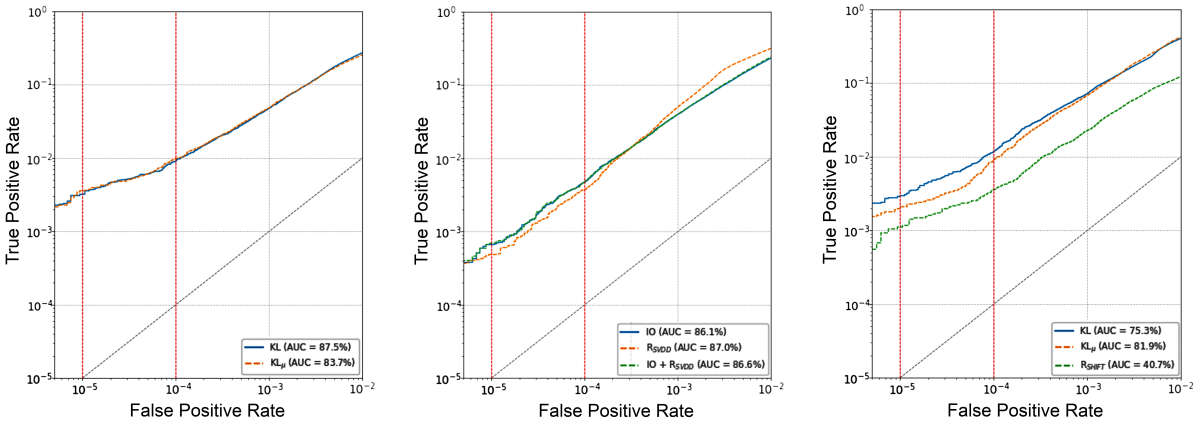
Second, the average TPR at an FPR of 10^{-5} metric demonstrated similar performance gains following the implementation of LOE. The LOE VAE again achieved an improvement of at least a factor of 2, while the *Shifty* VAE exhibited only marginal improvement over the standard DNN VAE. The AUC values of all the LOE-based models are comparable to those of their DNN counterparts, except for the *Shifty* VAE when the R_{Shift} metric is employed, resulting in an AUC of 40.7%. This discrepancy can be attributed to the fact that the R_{Shift} component is intended to enhance the mean component of the *Shifty* VAE rather than independently capture the underlying data structure.

Furthermore, as illustrated in Figure 5.4c, the *Shifty* VAE exhibits significant fluctuations in performance across the various AD metrics. The introduction of the *Shift* vector disrupts the correlation between the KL and KL_{μ} metrics observed in both the LOE and standard VAE models. This indicates a fundamental shift in the distribution of information within the latent space.

Similar to the DNN VAE, the LOE VAE demonstrates consistent performance across the various AD metrics. The KL and KL_{μ} metrics yield nearly identical performance in terms of both TPR at an FPR of 10^{-5} and the AUC, seen in Figure 5.4a. The LOE VAE and *Shifty* VAE exceed the TPR at an FPR of 10^{-5} target established in the methodology, achieving a maximum TPR of 0.3218 and 0.2908, respectively.

Table 5.3: Performance assessment of the contaminated models

Model		Performance Metric		
Architecture	AD score	TPR @ FPR 10^{-5} [%]	avg. TPR @ FPR 10^{-5} [%]	AUC [%]
LOE VAE	KL	0.3218	0.1566	87.5
LOE VAE	KL_μ	0.3589	0.1473	83.7
LOE DASVDD	IO	0.0665	0.0439	86.1
LOE DASVDD	R_{SVDD}	0.0475	0.0539	87.0
LOE DASVDD	R_{SVDD} + IO	0.0689	0.0441	86.6
Shifty VAE	KL	0.2908	0.0983	75.3
Shifty VAE	KL_μ	0.1980	0.0934	81.9
Shifty VAE	R_{Shift}	0.1113	0.0786	40.7



(a) LOE VAE (b) LOE DASVDD (c) *Shifty* VAE

Figure 5.4: AUC curves of LOE models under varying AD metrics. The red line in each plot indicates the TPR @ FPR 10^{-5}.

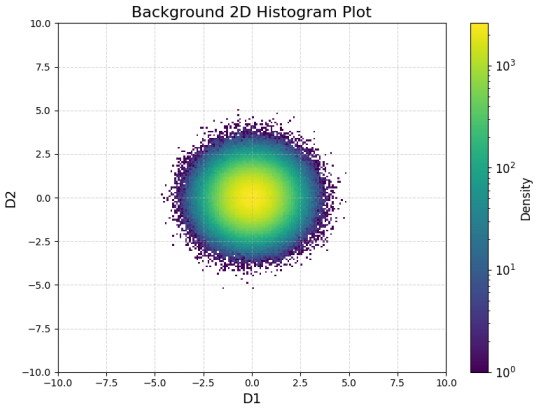
5.2.2 Latent Distributions

The latent distributions of the LOE VAE and *Shifty* VAE are presented in Figures 5.5 and 5.7, respectively. Upon examining the latent encoding distribution of the LOE VAE, a distinct single continuous Gaussian distribution is observed in both the background (Figure 5.5a) and the signal (Figure 5.5b). This contrasts with the distributions observed in the encoding space of the DNN VAE, which exhibits a more complex structure, as illustrated in Figure 5.2. This observation indicates a transformation in the latent structure compared to the vanilla VAE, where the distribution has been streamlined

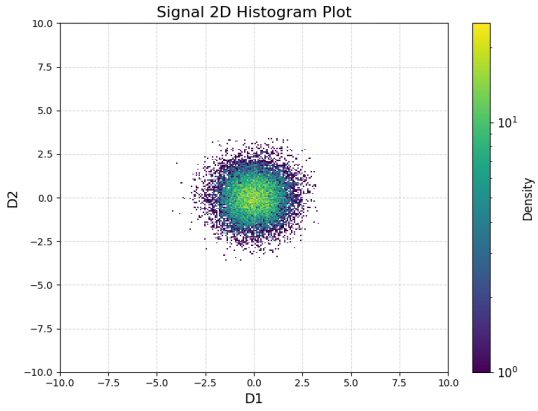
for AD, resulting in a more simplified representation. This simplification in latent structure is similarly observed in the distribution of the LOE DASVDD model compared to the DNN DASVDD, as illustrated in Figures 5.6 and 5.3, respectively. However, it is important to note that this alteration does not necessarily translate to enhanced AD performance. The simplification of the latent structure in the LOE VAE demonstrated significantly greater effectiveness compared to the LOE DASVDD, resulting in a clear and characterizable distribution. This enhancement in the latent space was correlated with improved AD performance. This observation may suggest a potential relationship between the quality of the simplified distribution and the efficacy of AD. In the LOE VAE, the background distribution is characterized by a Gaussian greater concentration around the peak, whereas the signal distribution is a more uniformly spread Gaussian. This shape is also reflected in the distribution of the mean encodings, which elucidates why both the KL and KL_μ metrics yield comparable performance.

The latent distribution of the *Shifty* VAE adheres to the intended design, wherein the *Shift* layer has resulted in the emergence of two distinct peaks. The positioning of the peaks in the latent space has been intentionally structured, with the peak in the positive quadrant corresponding to the encoding of background data, while the peak in the negative quadrant is intended for the encoding of signal data. In the background encoding, as illustrated in Figure 5.7a, the positive peak exhibits the highest density, confirming that the *Shift* vector is effectively influencing the location of the background encodings. Conversely, in the signal encoding depicted in Figure 5.7b, while there remains a significant density in the positive quadrant, an additional peak has emerged in the negative quadrant, indicating a concentration of anomalies as intended. The shape of the distribution is controlled by the *Shift* vector, the distributions of which are shown in Figures 5.7e and 5.7f. Comparing the distributions of the z encodings and the *Shift* vector encodings reveals a clear correlation; the *Shift* vector effectively adjusts the latent z encodings based on the model's classification of the data as anomalous or non-anomalous. This interaction highlights how the *Shift* vector influences the placement of the encodings within the latent space, reflecting the model's perception of the underlying data distribution.

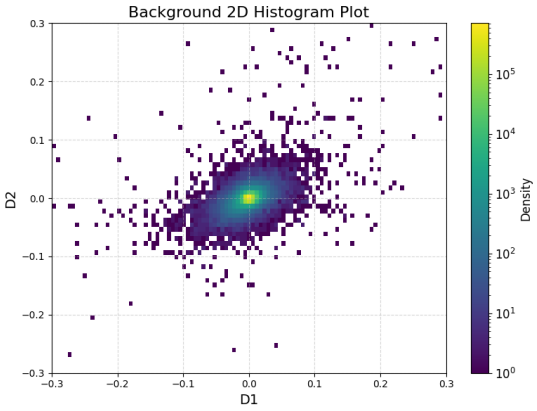
Next, the LOE VAE and *Shifty* VAE models undergo a sensitivity analysis to evaluate their robustness against varying anomaly rates. The outcomes of this analysis are discussed in Section 5.3.



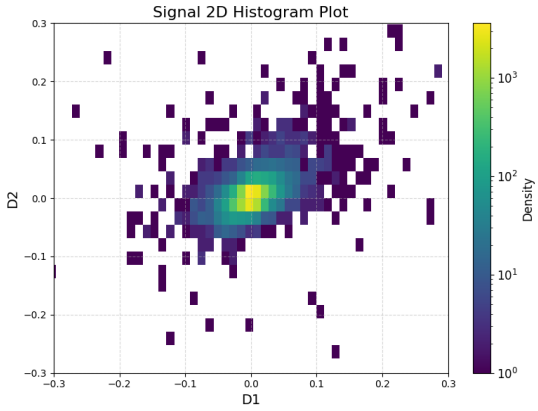
(a) z encodings - Background



(b) z encodings - Signal

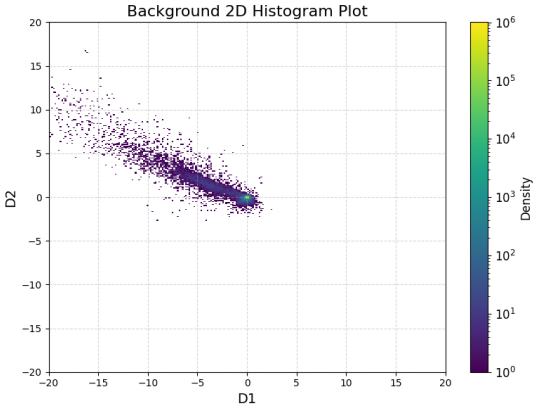


(c) μ encodinsg - Signal

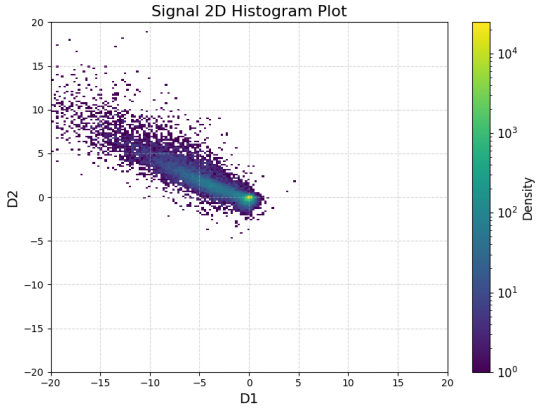


(d) μ encodings - Background

Figure 5.5: Distribution of latent encodings of LOE VAE. (a) and (b) show the distributions of the sampled encodings that are passed to decoder network, while (c) and (d) show the distributions of the encoder mean vectors.

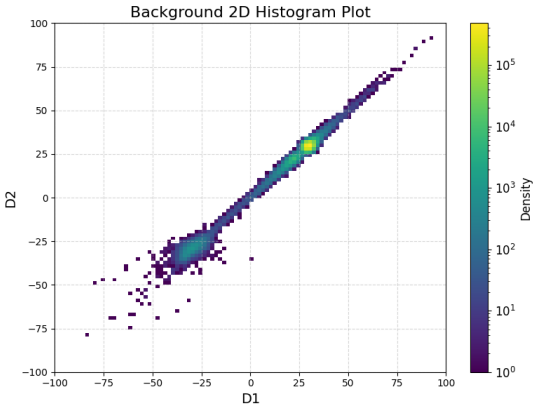


(a) z encodings - Background

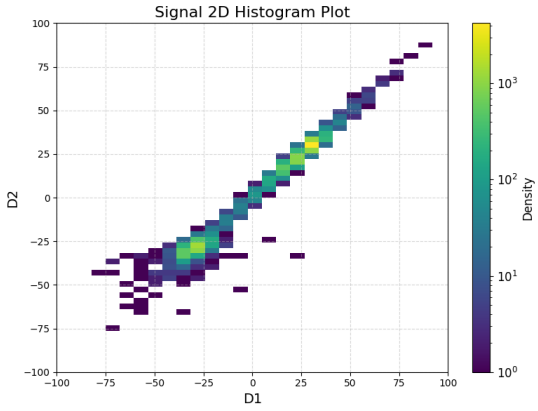


(b) z encodings - Signal

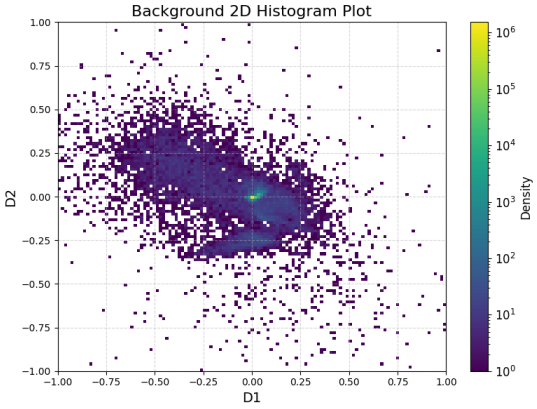
Figure 5.6: Distribution of latent encodings of LOE DASVDD. (a) and (b) show the distributions of the encodings passed to decoder network.



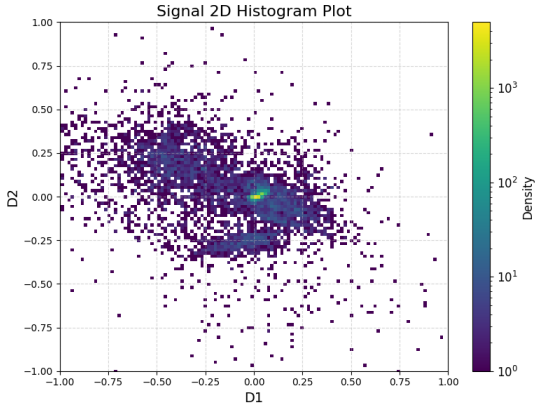
(a) z encodings - Background



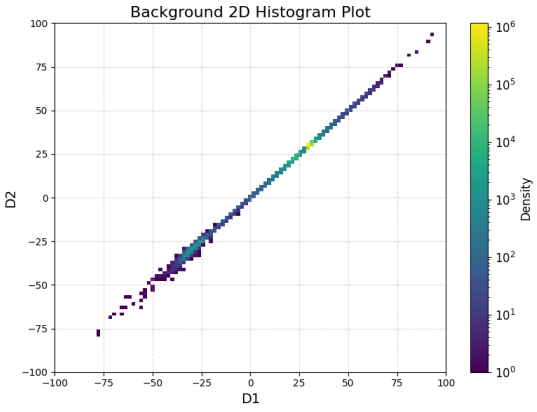
(b) z encodings - Signal



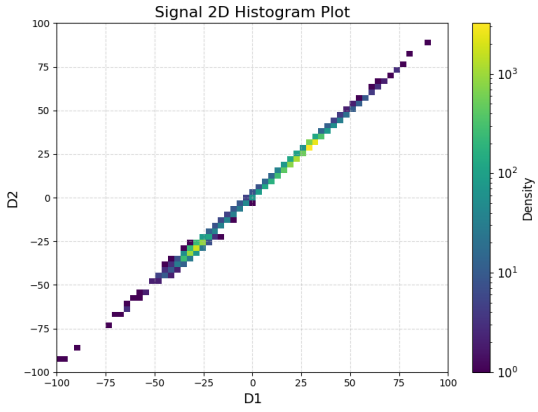
(c) μ encodinsg - Signal



(d) μ encodings - Background



(e) *Shift* encodings - Background



(f) *Shift* encodings - Background

Figure 5.7: Distribution of latent encodings of *Shifty* VAE. (a) and (b) show the distributions of the sampled encodings that are passed to decoder network, while (c) and (d) show the distributions of the encoder mean vectors, lastly (e) and (f) show the distribution of the encoder *Shift* vectors.

5.3 Sensitivity Analysis

The results of the sensitivity analysis for both LOE VAE and *Shifty* VAE models are presented in Tables 5.4, 5.5, and 5.6. These tables compare the models' robustness to varying contamination ratios across three key metrics: the TPR at a FPR of 10^{-5}, shown in Table 5.4; the AUC in Table 5.5; and the AD metric performance in Table 5.6. Each sub-table provides results for different combinations of true and assumed contamination ratios, denoted as α_0 and the hyperparameter α, respectively. Table 5.5 presents the AUC values corresponding to the optimal TPR at an FPR working point of 10^{-5}, while Table 5.4 details the TPR performance metrics. Table 5.6 identifies the AD metric that yielded the best TPR performance. The diagonal entries of the matrix represent the outcomes when the contamination ratio is accurately specified. The hyperparameter α denotes the presumed fraction of anomalies present in the training data. This comparison helps evaluate the performance of the models under varying contamination conditions, which can be considered analogous to varying Triggering or detector conditions in the L1T.

Referring to Tables 5.4, which presents the highest TPR at an FPR of 10^{-5} after ten training cycles, both models demonstrate sensitivity to the true anomaly rate (α_0), the expected anomaly rate (α), and the selected AD metric. In particular, examining the performance of the LOE VAE, Figure 5.4a illustrates that as the discrepancy between α_0 and α increases, the corresponding performance notably declines. Furthermore, the LOE VAE model exhibited greater performance degradation when α_0 significantly exceeded the expected contamination ratio, α. This suggests that the hyper-parameter tuning in the loss function has optimized the model for lower contamination ratios, as its optimal performance is observed when the true contamination ratio is 0.01 or lower. This trend is also observed for the *Shifty* VAE, as shown in Figure 5.4b, where the model achieves its best performance at low true contamination ratios. Regarding the best TPR, both models demonstrate competitive performance, particularly when the contamination ratio is at optimized at 0.01 for both α_0 and α.

Table 5.6b illustrates the AD metric that yielded the highest TPR performance for the *Shifty* VAE. The top-performing AD metric, shown in Table 5.6b, for the *Shifty* VAE varied significantly, leading to substantial variations in both TPR and AUC outcomes, as detailed in Table 5.5b. This variability renders the model impractical in an Unsupervised setting, as it is not possible to determine which AD metric will be effective in the presence of unlabelled anomalies. Referring to Table 5.6a, the LOE VAE demonstrated

a tendency to alternate between the KL and KL_μ metrics as the best-performing AD metric. The observed alternation between the metrics is attributed to their nearly identical performance, indicating a lack of significant differentiation in their effectiveness. Furthermore, the consistency of the AUC, shown in Table 5.5a, between the two metrics reinforces their reliability. Therefore the LOE VAE was chosen as the final model to undergo compression and FPGA implementation. Moving forward, the KL_μ is used for performance analysis, due to the Latency and Bandwidth benefits, however, the KL metric will still be displayed as a reference.

(a) LOE VAE

α true \ α assumed	0.5	1	2	5
0.5	0.30	0.24	0.29	0.21
1	0.28	0.36	0.22	0.25
2	0.15	0.11	0.23	0.18
5	0.10	0.13	0.14	0.17

(b) *Shifty* VAE

α true \ α assumed	0.5	1	2	5
0.5	0.22	0.21	0.26	0.16
1	0.24	0.29	0.24	0.29
2	0.17	0.16	0.20	0.16
5	0.26	0.13	0.14	0.11

Table 5.4: A sensitivity study evaluates the robustness of LOE to varying contamination ratios in terms of the TPR @ FPR 10^{-5} [%]. The true anomaly rate (α true) is shown on the y-axis, while the expected anomaly rate (α assumed) is shown on the x-axis. Both α true and α assumed range from 0.5 to 5 times the original rate of 1%. For visual clarity, different colours indicate TPR performance levels: dark green for TPR ≥ 0.25, light green for $0.20 \leq$ TPR < 0.24, orange for $0.15 \leq$ TPR < 0.20, and red for TPR < 0.15.

(a) LOE VAE

α true \ α assumed	0.5	1	2	5
0.5	83.9	84.6	82.4	82.6
1	89.0	83.7	78.2	85.3
2	85.5	86.1	81.3	86.0
5	85.1	85.4	85.9	82.1

(b) *Shifty* VAE

α true \ α assumed	0.5	1	2	5
0.5	29.1	80.3	75.7	84.2
1	72.1	75.3	80.0	91.1
2	30.9	89.2	36.5	44.9
5	81.2	45.9	41.6	33.2

Table 5.5: A sensitivity study evaluates the robustness of LOE to varying contamination ratios in terms of the TPR @ FPR 10^{-5} [%]. The definitions of α true and α assumed are provided in Table 5.4. For visual clarity, different colors indicate AUC performance levels: dark green for AUC $\geq 85\%$, light green for $80\% \leq$ AUC $< 85\%$, orange for $75\% \leq$ AUC $< 80\%$, and red for AUC $< 75\%$.

(a) LOE VAE

α true \ α assumed	0.5	1	2	5
0.5	KL	KL_μ	KL	KL
1	KL	KL_μ	KL	KL_μ
2	KL_μ	KL	KL_μ	KL
5	KL_μ	KL_μ	KL	KL_μ

(b) *Shifty* VAE

α true \ α assumed	0.5	1	2	5
0.5	R_{Shift}	KL	KL	KL_μ
1	KL	KL	KL_μ	KL
2	R_{Shift}	KL_μ	R_{Shift}	R_{Shift}
5	KL_μ	R_{Shift}	R_{Shift}	R_{Shift}

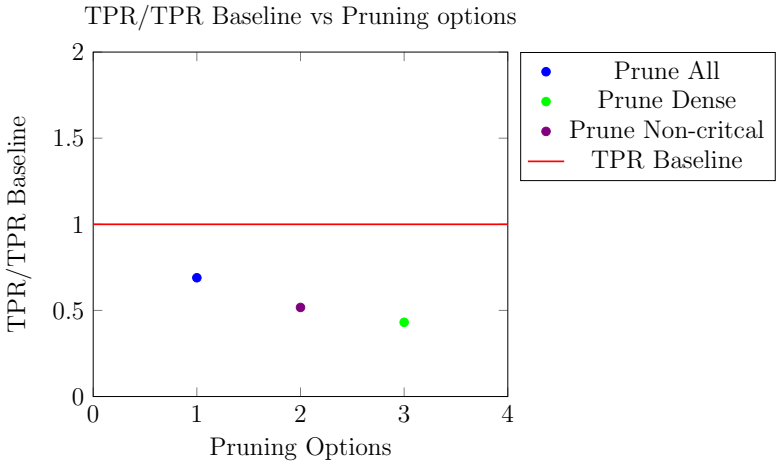
Table 5.6: A sensitivity study evaluates the robustness of LOE to varying contamination ratios in terms of the TPR @ FPR 10^{-5} [%]. The definitions of α true and α assumed are provided in Table 5.4. The color coding follows that of Table 5.4 to facilitate comparison between TPR performance and the choice of AD metric.

5.4 Model Compression

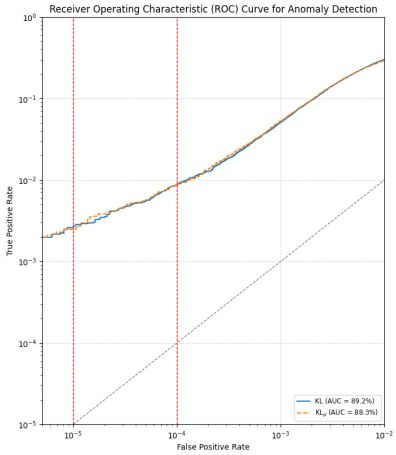
The following section presents the results of applying Pruning and PTQ techniques to the standard LOE VAE model, which will be referred to as the BF LOE VAE model moving forward.

5.4.1 Pruning

The impact of Pruning on model performance is shown in Figure 5.8, where various Pruning methods are compared in terms of TPR normalized by the TPR baseline. Additionally, the AUC for the best performing pruned model is displayed. These results are further summarized in Table 5.7. Similarly, the impact of PTQ is demonstrated in Figure 5.9, where model performance is analyzed as a function of bit width, focusing on TPR normalized by the TPR Baseline. The best-performing bit width is further assessed in terms of AUC, and these findings are consolidated in Table 5.8, which compares the performance metrics at the optimal bit width. The Pruning and Quantization results provide insight into the trade-offs between model compression and AD accuracy.



(a) TPR against TPR Baseline



(b) AUC with all layers pruned

Figure 5.8: Plots showing how different Pruning methods impact performance. a) shows the TPR @ FPR 10^{-5} [%] normalised by the TPR baseline of 0.3589% from the BF model, where KLμ was used as the anomaly metric. b) shows the AUC of the top performing pruned model, with the red line in each plot indicating the TPR @ FPR 10^{-5}.

Table 5.7: Performance assessment of the BP Models

Model		Performance Metric	
Architecture	AD score	TPR @ FPR 10^{-5} [%]	AUC [%]
LOE VAE	KL	0.2598	89.2
LOE VAE	KL_{μ}	0.2475	88.3

Referring to Figure 5.8a, the best-performing Pruning strategy was found to be uniform Pruning of 50% of the connections across all layers in the LOE VAE model. This aggressive reduction of network parameters resulted in a 31% drop in TPR at a fixed FPR of 10^{-5} when compared to the unpruned baseline model (referred to as the BF model). Despite this reduction, the final TPR achieved by the pruned model was 0.2475%, which remains well above the predefined performance threshold of 0.1999% for deployment in the L1T system.

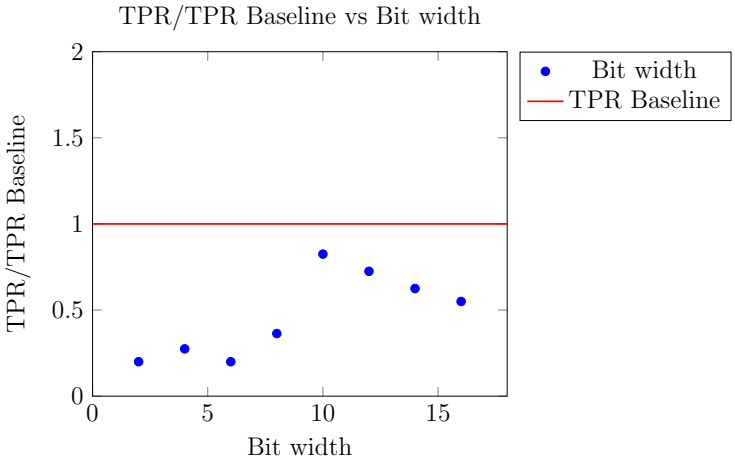
Additionally, the area under the ROC curve AUC—shown in Figure 5.8b—remains relatively high at 88.3%. This suggests that the pruned model retains a strong ability to distinguish between anomalous and background events, even with a significant reduction in complexity. The success of this Pruning strategy is significant, as it directly translates into reduced Latency and lower resource utilization when deploying the model on hardware.

The resulting pruned model is henceforth referred to as the BP model. Given its robust performance under reduced complexity, the BP model is selected for further optimization via PTQ. This step is essential for converting the model from floating-point to fixed-point representation, ensuring compatibility with FPGA architectures. The hls4ml tool is used for this purpose, simultaneously quantizing the model and generating synthesizable Firmware for FPGA deployment. This sets the stage for subsequent analysis of the model's hardware performance in terms of resource usage, Latency, and Bandwidth.

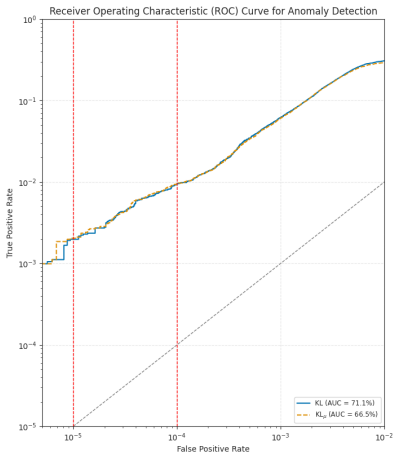
5.4.2 PTQ

The results of the bit precision scan, as illustrated in Figure 5.9a, indicate that a bit width of 10 was the only configuration that achieved the quantized model's TPR goal of 0.1999%. The optimal split between the fixed-point and fractional bits was found to be 6 and 4, respectively. The AUC of the quantized model is shown in 5.9b while the results

are recorded in Table 5.8. The BP model demonstrates state-of-the-art performance once implemented in Firmware. The final step involves evaluating the resources required to synthesize the model on an FPGA. This includes analyzing the utilization of logic elements, memory, and power consumption to ensure efficient deployment on the hardware.



(a) TPR against TPR Baseline



(b) AUC at best bit width

Figure 5.9: Plots showing performance as a function of bit width. a) shows the TPR @ FPR 10^{-5} [%] normalised by the TPR baseline of 0.2475% from the BP model, where KLμ was used as the anomaly metric and only the optimal fixed/fraction split is shown. b) shows the AUC of the top performing quantized model, with the red line in each plot indicating the TPR @ FPR 10^{-5}.

Table 5.8: Performance assessment of quantized model at best bit width (10,6)

Model		Performance Metric	
Architecture	AD score	TPR @ FPR 10^{-5} [%]	AUC [%]
LOE VAE	KL	0.1918	71.7
LOE VAE	KL_μ	0.2042	66.5

5.5 Porting models to FPGAS

The tables presented below provide a comprehensive overview of the FPGA resource utilization and timing performance of the implemented model, targeting a Xilinx Virtex UltraScale+ VU9P (`xcvu9p-flgb2104-2-e`). Table 5.9 summarizes the resource usage of key FPGA components, including Block Random Access Memory (BRAM), Digital Signal Processing slices (DPS) slices, Flip-Flops (FFs), Look-Up Tables (LUTs), and UltraRAM

(URAM). The table highlights both the number of resources used and the corresponding utilization percentages, relative to the available resources in a single Super Logic Region (SLR) and across the entire FPGA device. Following the resource breakdown, Table 5.10 presents the timing and Latency summary of the FPGA implementation. It details the target and estimated clock periods, with uncertainty measurements, for the primary clock signal. Additionally, the Latency section outlines the minimum and maximum Latency in clock cycles, along with the corresponding absolute Latency and pipeline interval, providing insights into the timing efficiency and throughput of the design. The full synthesis report can be found in Appendix A.

Referring to Table 5.9, the design utilizes a small fraction of the total available FPGA resources, demonstrating high efficiency. Specifically, no BRAM or URAM resources are used, leaving all of these resources available for future use. Only 0.82% of the DSP48E blocks are utilized, indicating very low usage and high availability. FF usage is minimal at 0.17%, and LUT utilization stands at just 3.45%, meaning the vast majority of these resources remain available. Overall, this efficient resource utilization leaves significant capacity for additional logic or future enhancements. This meets the goal of 12% or less, from Govorkova et al. [9], ensuring that the model is of acceptable size for use in the LHC Trigger environment.

The timing and Latency summary of the design, presented in Table 5.10, indicates that the target clock period was 5.00 ns, while the estimated clock period achieved was 4.330 ns with an uncertainty of 0.62 ns. The design has a consistent Latency of 5 cycles, corresponding to an absolute Latency of 25.000 ns at a clock frequency of 40 MHz, which equates to 1 clock cycle given the 25 ns clock period. The interval between operations is fixed at 25.000 ns with a pipeline depth of 1. These results demonstrate that the design meets the desired timing constraints with a margin and maintains a consistent and predictable Latency profile, again confirming the suitability for the LHC Trigger.

Table 5.9: FPGA Resource Utilization

Resource	**Used**	**Available SLR**	**Utilization SLR (%)**	**Available**	**Utilization (%)**
BRAM_18K	0	1440	0	4320	0
DSP48E	56	2280	2	6840	~ 0
FF	3921	788160	~ 0	2364480	~ 0
LUT	40752	394080	10	1182240	3
URAM	0	320	0	960	0

Table 5.10: Timing and Latency Summary

Clock	**Target**	**Estimated**	**Uncertainty**
ap_clk	5.00 ns	4.330 ns	0.62 ns

Latency (cycles)		**Latency (absolute)**		**Interval**		**Pipeline**
min	**max**	**min**	**max**	**min**	**max**	**Type**
5	5	25.000 ns	25.000 ns	1	1	function

Chapter 6

Conclusions and Recommendations

6.1 Conclusions

In this work, the objective was to enhance AE-based detection strategies for deployment within the L1T infrastructure. Specifically, the goal was to develop an Unsupervised AD system utilizing a DNN or CNN on an FPGA. Ensuring the system operates in an Unsupervised manner is essential to maintain signal agnosticism and prevent the introduction of background bias into the algorithm. Once model development is completed, the `hls4ml` library is utilized to synthesize the model into Firmware, achieving $\mathcal{O}(1)\,\mu$s Latency with minimal resource utilization following Quantization and Pruning. This optimization is crucial, as the L1T Trigger environment imposes stringent Latency and Bandwidth requirements.

Initially, a variety of AEs were tested as both CNN and DNN models. The best-performing models were identified as the DNN VAE and the DNN DASVDD, with the results presented in Tables 5.1 and 5.2. Both models are able to utilize the projected representation of a given input in the latent space to classify anomalous data. This approach is particularly advantageous for L1T FPGA implementation. In the VAE, this approach eliminates the need to sample Gaussian-distributed pseudorandom numbers, ensuring that the Trigger decision remains deterministic. Furthermore, for both models, it obviates the necessity of running the decoder, leading to significant resource savings. However, these models are not specifically designed for Unsupervised AD in the L1T. In this work, the objective is to adapt the initial models explicitly for application within the

L1T framework. This involves training using an adapted objective function tailored for Unsupervised AD. Additionally, the model must Trigger at a controlled rate, as anomalies in the L1T are expected to be extremely rare. This context of the anticipated Trigger rate will also be integrated into the adapted objective function. The incorporation of this additional information endows the model with the ability to learn rich features catered for AD in the LHC.

LOE is selected to adapt the original models, enhancing their suitability for datasets contaminated by unidentified anomalies. During training, LOE simultaneously infers anomalous data within the training set while updating its parameters by solving a Mixed continuous-discrete optimization problem. This process iteratively refines both the model and its predictions of anomalies. In each batch, the model must identify a portion of the data as anomalous, adhering to the maximum L1T acceptance rate of 1%. This requirement encourages the model to generate a distribution in which a subset of the encoded data is pushed away from the main cluster. Furthermore, it promotes the formation of distributions that are tailored for single-class AD. LOE was applied to the two autoencoder models discussed above, as well as to a novel model referred to as the *Shifty* VAE.

The performance benefits of LOE are evident in both the anomaly detection performance presented in Table 5.3 and the latent distributions illustrated in Figures 5.5, 5.3, and 5.7. These results demonstrate that LOE effectively enhances the model's ability to detect anomalies while also refining the structure of the latent space, contributing to a more effective AE framework. LOE resulted in the Unsupervised DNN VAE outperforming supervised versions on the exact dataset; with regards to the TPR @ FPR 10^{-5} metric. The *Shifty* VAE also showed potential for state of the art TPR results, but was unreliable when subjected to sensitivity analysis, in Section 5.3. The latent distributions that resulted from the implementation of LOE showed simplified structure compared to that of the original AE models. This phenomenon was particularly pronounced in the LOE VAE, where a continuous Gaussian distribution emerged in the latent space. Anomalies were effectively pushed towards the tails of this distribution, facilitating their identification. This configuration is optimal for the L1T, where single-class anomaly detection with a low false positive rate is essential. All LOE models utilize the projected representation of inputs to classify anomalies, thereby benefiting from the advantages discussed earlier. As the best-performing model, the LOE VAE was selected for FPGA implementation, further enhancing its applicability in real-time environments.

To optimize the LOE VAE for deployment, the model underwent Pruning and Quantization. Several Pruning strategies were explored, with the most effective approach being a uniform 50% reduction in connections across all layers, which yielded minimal degradation in performance, as demonstrated in Figure 5.8. Subsequent to Pruning, PTQ was performed, utilizing a bit precision scan to determine the optimal Quantization level. As illustrated in Figure 5.9, a precision of 10 bits was identified as the most effective, offering the best trade-off between model accuracy and hardware efficiency. Simultaneously, the model was synthesized into Firmware using the hls4ml framework and subjected to testing. The quantized model demonstrated state-of-the-art performance, achieving a TPR of 0.1999% at a FPR of 10^{-5}, underscoring its suitability for deployment in resource-constrained environments such as the L1T.

Finally, an analysis of resource utilization, as presented in Table 5.9, along with the Latency metrics summarized in Table 5.10, reveals that the LOE VAE model effectively utilizes less than 3% of the available resources on the Xilinx VU9P FPGA. Moreover, the observed Latency of the model is less than 5 ns, which aligns with the stringent initiation interval requirements dictated by the frequency of bunch crossings at the LHC. This performance demonstrates the model's suitability for deployment in high-frequency environments while maintaining a lightweight architecture.

The LOE framework facilitated the development and optimization of an Unsupervised AD system specifically tailored for deployment within the L1T infrastructure. By incorporating LOE, the models were effectively adapted to address the challenges posed by contaminated datasets, yielding substantial performance improvements, even when compared to their supervised counterparts. The final implementation on FPGA demonstrated exceptional efficiency, utilizing minimal resources while meeting the stringent Latency requirements of the L1T environment. These results underscore the potential of the LOE VAE for real-time, high-performance Unsupervised AD in the demanding conditions of the LHC. True signal-agnostic AD presents a promising approach for elucidating potential new physics phenomena that may be concealed within the data.

6.2 Recommendations

Deploying the proposed models within the L1T system on real hardware presents additional challenges. The limited resources of FPGAs, such as memory capacity and computational throughput, require significant model optimisation. Techniques like quantisation and Pruning, although effective for reducing resource usage, may introduce approximation errors that could impact the model's detection performance [37]. After the LOE VAE model has been successfully converted to hardware using `hls4ml` [13] and its performance tested at various quantisation levels to minimise the resource footprint in simulation, it is crucial to deploy the model on a replicated ATLAS L1T setup [62], [137]. This ensures that the hardware implementation matches the emulation and is essential for validating new L1T configurations before their deployment in ATLAS [138]. In line with the approach used by Zipper et al. [62] in the CMS experiment, the model should be tested on a test beam. This method involves collecting real collision data without recording outputs, enabling the monitoring of actual Trigger rates. This process ensures that the model avoids excessive Triggering and performs as expected when exposed to real data at true 40MHz data rates. Future research should focus on systematic testing under realistic hardware constraints to ensure the models' robustness and flexibility.

The *Shifty* VAE shows promise in generating latent spaces suitable for anomaly detection (AD). However, it currently does not offer the same level of reliability as the LOE VAE. The *Shifty* VAE has significant potential in its ability to create clusters of anomalies, which may help mitigate the *look-elsewhere effect* by ensuring anomalies are not isolated, thus aiding in their identification and localisation within the dataset. Further development is needed to stabilise its performance and drive progress in Unsupervised AD. Enhancements to model robustness and adaptability could substantially increase the effectiveness of Unsupervised methods, allowing them to better handle varied datasets and operational conditions.

Although the proposed models exhibit strong anomaly detection performance within the L1T framework, they come with certain limitations. These models are designed to detect anomalies that diverge from the learned latent distribution; however, their dependence on latent space representations may reduce sensitivity to anomalies that resemble normal variations or involve subtle, localised deviations. This issue is especially pertinent for complex signatures that overlap with background fluctuations, potentially resulting in misclassification. Moreover, the models are optimised for scenarios where contamination rates are controlled, typically ranging from 1% to 5%. Their generalisability to datasets

with significantly different anomaly frequencies remains uncertain. Additionally, while the models can be retrained for varying Trigger rates, their performance under extreme luminosity variations and diverse background conditions has not been thoroughly investigated. Future studies should explore adaptive retraining methods and integrate supplementary detection techniques to enhance robustness across a wider range of anomaly types, including those present in non-physics datasets.

Bibliography

[1] C. Patrignani, "Review of particle physics," *Chinese Physics C*, vol. 40, no. 10, p. 100 001, 2016. DOI: 10.1088/1674-1137/40/10/100001. [Online]. Available: https://doi.org/10.1088/1674-1137/40/10/100001.

[2] H.-C. Cheng and I. Low, "Tev symmetry and the little hierarchy problem," *Journal of High Energy Physics*, vol. 2003, no. 09, p. 051, 2003. DOI: 10.1088/1126-6708/2003/09/051. arXiv: hep-ph/0308199. [Online]. Available: https://doi.org/10.1088/1126-6708/2003/09/051.

[3] D. E. Morrissey, T. Plehn, and T. M. Tait, "Physics searches at the lhc," *Physics Reports*, vol. 515, pp. 1–113, 2012. DOI: 10.1016/j.physrep.2012.02.007. arXiv: 0912.3259 [hep-ph]. [Online]. Available: https://doi.org/10.1016/j.physrep.2012.02.007.

[4] D. H. Ballard, "Modular learning in neural networks," ser. AAAI'87, Seattle, Washington: AAAI Press, 1987, pp. 279–284, ISBN: 0934613427.

[5] G. Kasieczka, B. Nachman, D. Shih, *et 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. 124 201, Dec. 2021. DOI: 10.1088/1361-6633/ac36b9. [Online]. Available: https://doi.org/10.1088%2F1361-6633%2Fac36b9.

[6] C. Qiu, A. Li, M. Kloft, M. Rudolph, and S. Mandt, *Latent outlier exposure for anomaly detection with contaminated data*, 2022. arXiv: 2202.08088 [cs.LG].

[7] E. Gross and O. Vitells, "Trial factors for the look elsewhere effect in high energy physics," *The European Physical Journal C*, vol. 70, no. 1, pp. 525–530, Nov. 2010, ISSN: 1434-6052. DOI: 10.1140/epjc/s10052-010-1470-8. [Online]. Available: https://doi.org/10.1140/epjc/s10052-010-1470-8.

[8] E. Fortin, "Machine learning for real-time processing of atlas liquid argon calorimeter signals with fpgas," in *Proceedings of the TIPP 2023 Conference*, On behalf of

the ATLAS Liquid Argon Calorimeter Group, European Organization for Nuclear Research (CERN), Cape Town, Sep. 2023.

[9] E. Govorkova, E. Puljak, T. Aarrestad, *et al.*, "Autoencoders on field-programmable gate arrays for real-time, unsupervised new physics detection at 40 mhz at the large hadron collider," *Nature Machine Intelligence*, vol. 4, no. 2, pp. 154–161, Feb. 2022, ISSN: 2522-5839. DOI: 10.1038/s42256-022-00441-3. [Online]. Available: http://dx.doi.org/10.1038/s42256-022-00441-3.

[10] R. Chalapathy, A. Menon, and S. Chawla, "Anomaly detection using one-class neural networks," *arXiv preprint arXiv:1802.06360*, 2018.

[11] L. Ruff, R. Vandermeulen, N. Goernitz, *et al.*, "Deep one-class classification," in *International Conference on Machine Learning*, PMLR, 2018, pp. 4393–4402.

[12] Y. Zhou, X. Liang, W. Zhang, L. Zhang, and X. Song, "Vae-based deep svdd for anomaly detection," *Neurocomputing*, vol. 453, pp. 131–140, 2021, ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2021.04.089. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231221006470.

[13] J. Duarte, S. Han, P. Harris, *et al.*, "Fast inference of deep neural networks in fpgas for particle physics," *Journal of Instrumentation*, vol. 13, no. 07, P07027–P07027, Jul. 2018, ISSN: 1748-0221. DOI: 10.1088/1748-0221/13/07/p07027. [Online]. Available: http://dx.doi.org/10.1088/1748-0221/13/07/P07027.

[14] T. Aarrestad, V. Loncar, M. Pierini, S. Summers, J. Ngadiuba, and C. P. and, "Fast convolutional neural networks on fpgas with hls4ml," *CoRR*, vol. abs/2101.05108, 2021. arXiv: 2101.05108. [Online]. Available: https://arxiv.org/abs/2101.05108.

[15] C. Coelho, A. Kuusela, S. Li, *et al.*, "Automatic heterogeneous quantization of deep neural networks for low-latency inference on the edge for particle detectors," *Nature Machine Intelligence*, vol. 3, pp. 1–12, Aug. 2021. DOI: 10.1038/s42256-021-00356-5.

[16] E. Govorkova, E. Puljak, T. Aarrestad, M. Pierini, K. A. Woźniak, and J. Ngadiuba, *Lhc physics dataset for unsupervised new physics detection at 40 mhz*, 2021. arXiv: 2107.02157 [physics.data-an].

[17] A. Seiden, "Characteristics of the ATLAS and CMS detectors," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 370, no. 1961, G. Kalmus, R. Brown, D. Evans, V. Gibson, and R. Nickerson, Eds., Physics at the high-energy frontier: the Large Hadron Collider project 2012. DOI: 10.1098/rsta.2011.0461.

[18] ATLAS Collaboration, *Higgs and diboson searches*, https://twiki.cern.ch/twiki/bin/view/AtlasPublic/HDBSPublicResults, 2019.

[19] CMS Collaboration, *Cms exotica public physics results*, https://twiki.cern.ch/twiki/bin/view/CMSPublic/PhysicsResultsEXO, 2019.

[20] ATLAS Collaboration, *Supersymmetry searches*, https://twiki.cern.ch/twiki/bin/view/AtlasPublic/SupersymmetryPublicResults, 2019.

[21] ATLAS Collaboration, *Exotic physics searches*, https://twiki.cern.ch/twiki/bin/view/AtlasPublic/ExoticsPublicResults, 2019.

[22] T. A. collaboration, "Operation of the atlas trigger system in run 2," *Journal of Instrumentation*, vol. 15, no. 10, P10004, Aug. 2020. DOI: 10.1088/1748-0221/15/10/P10004. [Online]. Available: https://dx.doi.org/10.1088/1748-0221/15/10/P10004.

[23] R. Hauser, "The atlas trigger system," *The European Physical Journal C - Particles and Fields*, vol. 34, no. 1, s173–s183, Jul. 2004, ISSN: 1434-6052.

[24] M. Aaboud, G. Aad, B. Abbott, O. Abdinov, B. Abeloos, and Abidi, "A strategy for a general search for new phenomena using data-derived signal regions and its application within the atlas experiment," *The European Physical Journal C*, vol. 79, no. 2, Feb. 2019, ISSN: 1434-6052. DOI: 10.1140/epjc/s10052-019-6540-y. [Online]. Available: http://dx.doi.org/10.1140/epjc/s10052-019-6540-y.

[25] J. Button, G. R. Kalbfleisch, G. R. Lynch, B. C. Magli ć, A. H. Rosenfeld, and M. L. Stevenson, "Pion-pion interaction in the reaction $\overline{p}+p \to 2\pi^+ + 2\pi^- + n\pi^0$," *Phys. Rev.*, vol. 126, pp. 1858–1863, 5 Jun. 1962. DOI: 10.1103/PhysRev.126.1858. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRev.126.1858.

[26] F. Poppi, "Is the bell ringing?" *CERN Bull.*, p. 14, 2010, BUL-NA-2010-317.

[27] "CMS Exotica hotline leads hunt for exotic particles," *Symmetry Magazine*, 2010.

[28] G. Aad, B. Abbott, D. C. Abbott, *et al.*, "Performance of electron and photon triggers in atlas during lhc run 2," *The European Physical Journal C*, vol. 80, no. 1, Jan. 2020, ISSN: 1434-6052. DOI: 10.1140/epjc/s10052-019-7500-2. [Online]. Available: http://dx.doi.org/10.1140/epjc/s10052-019-7500-2.

[29] S. Chatrchyan, V. Khachatryan, A. Sirunyan, *et al.*, "Observation of a new boson at a mass of 125 gev with the cms experiment at the lhc," *Phys. Lett. B*, vol. 716, pp. 30–61, 2012. DOI: 10.1016/j.physletb.2012.08.021. eprint: 1207.7235.

[30] A. M. Sirunyan, A. Tumasyan, W. Adam, *et al.*, "MUSiC: a model unspecific search for new physics in proton-proton collisions at $\sqrt{s} = 13$ TeV," 2020. eprint: arXiv:2010.02984.

[31] CMS Collaboration. "MUSiC, a model unspecific search for new physics, in *pp* collisions at $\sqrt{s} = 13$ TeV." (2020), [Online]. Available: https://cds.cern.ch/record/2718811.

[32] CMS Collaboration. "Model Unspecific Search for New Physics in *pp* Collisions at $\sqrt{s} = 7$ TeV." (2011), [Online]. Available: http://cds.cern.ch/record/1360173.

[33] ATLAS collaboration. "A general search for new phenomena with the ATLAS detector in *pp* collisions at $\sqrt{s} = 7$ TeV." (Aug. 2012), [Online]. Available: https://cds.cern.ch/record/1472686.

[34] ATLAS collaboration. "A general search for new phenomena with the ATLAS detector in *pp* collisions at $\sqrt{s} = 8$ TeV." (Mar. 2014), [Online]. Available: https://cds.cern.ch/record/1666536.

[35] M. Aaboud, G. Aad, B. Abbott, *et al.*, "A strategy for a general search for new phenomena using data-derived signal regions and its application within the atlas experiment," *Eur. Phys. J. C*, vol. 79, p. 120, 2019. eprint: 1807.07447.

[36] 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, ISSN: 1029-8479. DOI: 10.1007/jhep03(2022)066. [Online]. Available: http://dx.doi.org/10.1007/JHEP03(2022)066.

[37] T. Aarrestad, V. Loncar, N. Ghielmetti, *et al.*, "Fast convolutional neural networks on fpgas with hls4ml," *Machine Learning: Science and Technology*, vol. 2, no. 4, p. 045 015, Jul. 2021, ISSN: 2632-2153. DOI: 10.1088/2632-2153/ac0ea1. [Online]. Available: http://dx.doi.org/10.1088/2632-2153/ac0ea1.

[38] J. H. Collins, K. Howe, and B. Nachman, "Anomaly detection for resonant new physics with machine learning," *Phys. Rev. Lett.*, vol. 121, p. 241 803, 2018, [INSPIRE]. eprint: arXiv:1805.02664.

[39] R. T. D'Agnolo and A. Wulzer, "Learning new physics from a machine," *Phys. Rev. D*, vol. 99, p. 015 014, 2019, [INSPIRE]. eprint: arXiv:1806.02350.

[40] A. D. Simone and T. Jacques, "Guiding new physics searches with unsupervised learning," *Eur. Phys. J. C*, vol. 79, p. 289, 2019, [INSPIRE]. eprint: arXiv:1807.06038.

[41] A. Casa and G. Menardi, "Nonparametric semisupervised classification for signal detection in high energy physics," [INSPIRE]. eprint: arXiv:1809.02977.

[42] J. H. Collins, K. Howe, and B. Nachman, "Extending the search for new resonances with machine learning," *Phys. Rev. D*, vol. 99, p. 014 038, 2019, [INSPIRE]. eprint: arXiv:1902.02634.

[43] S. Caron, L. Hendriks, and R. Verheyen, "Rare and different: Anomaly scores from a combination of likelihood and out-of-distribution models to detect new physics at the lhc," [INSPIRE]. eprint: arXiv:2106.10164.

[44] J. Gonski, J. Lai, B. Nachman, and I. Ochoa, "High-dimensional anomaly detection with radiative return in e^+e^- collisions," [INSPIRE]. eprint: arXiv:2108.13451.

[45] B. Ostdiek, "Deep set auto encoders for anomaly detection in particle physics," *SciPost Phys.*, vol. 12, p. 045, 2022, [INSPIRE]. eprint: arXiv:2109.01695.

[46] CMS Collaboration, "The phase-2 upgrade of the cms level-1 trigger," CERN, Tech. Rep. CERN-LHCC-2020-004. CMS-TDR-021, 2020.

[47] T. Finke, M. Krämer, A. Morandini, A. Mück, and I. Oleksiyuk, "Autoencoders for unsupervised anomaly detection in high energy physics," *Journal of High Energy Physics*, vol. 2021, no. 6, Jun. 2021. DOI: 10.1007/jhep06(2021)161. [Online]. Available: https://doi.org/10.1007%2Fjhep06%282021%29161.

[48] Y. Liao, A. Bartler, and B. Yang, "Anomaly detection based on selection and weighting in latent space," *CoRR*, vol. abs/2103.04662, 2021. arXiv: 2103.04662. [Online]. Available: https://arxiv.org/abs/2103.04662.

[49] T. Aarrestad, M. van Beekveld, M. Bona, *et 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. DOI: 10.21468/scipostphys.12.1.043. [Online]. Available: https://doi.org/10.21468%2Fscipostphys.12.1.043.

[50] Z. Wang and A. C. Bovik, "Mean squared error: Love it or leave it? a new look at signal fidelity measures," *IEEE Signal Processing Magazine*, vol. 26, no. 1, pp. 98–117, 2009. DOI: 10.1109/MSP.2008.930649.

[51] R. Corizzo, M. Ceci, and N. Japkowicz, "Anomaly detection and repair for accurate predictions in geo-distributed big data," *Big Data Research*, vol. 16, pp. 18–35, 2019, ISSN: 2214-5796. DOI: https://doi.org/10.1016/j.bdr.2019.04.001. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2214579618302119.

[52] Z. Zhang, T. Jiang, S. Li, and Y. Yang, "Automated feature learning for nonlinear process monitoring – an approach using stacked denoising autoencoder and k-nearest neighbor rule," *Journal of Process Control*, vol. 64, pp. 49–61, 2018, ISSN: 0959-1524. DOI: https://doi.org/10.1016/j.jprocont.2018.02.004. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S095915241830026X.

[53] J. Guo, G. Liu, Y. Zuo, and J. Wu, "An anomaly detection framework based on autoencoder and nearest neighbor," Jul. 2018, pp. 1–6. DOI: 10.1109/ICSSSM.2018.8464983.

[54] F. Angiulli, F. Fassetti, and L. Ferragina, "Latentout: An unsupervised deep anomaly detection approach exploiting latent space distribution," eng, *Machine learning*, 2022, ISSN: 0885-6125.

[55] S. N. Marimont and G. Tarroni, "Anomaly detection through latent space restoration using vector quantized variational autoencoders," in *2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI)*, 2021, pp. 1764–1767. DOI: 10.1109/ISBI48211.2021.9433778.

[56] L. Vu, V. L. Cao, Q. U. Nguyen, D. N. Nguyen, D. T. Hoang, and E. Dutkiewicz, "Learning latent representation for iot anomaly detection," *IEEE Transactions on Cybernetics*, vol. 52, no. 5, pp. 3769–3782, 2022. DOI: 10.1109/TCYB.2020.3013416.

[57] J. Ngadiuba, V. Loncar, M. Pierini, *et al.*, "Compressing deep neural networks on fpgas to binary and ternary precision with ¡tt¿hls4ml¡/tt¿," *Machine Learning: Science and Technology*, vol. 2, no. 1, p. 015 001, Dec. 2020, ISSN: 2632-2153. DOI: 10.1088/2632-2153/aba042. [Online]. Available: http://dx.doi.org/10.1088/2632-2153/aba042.

[58] D. Rankin, J. Krupa, P. Harris, *et al.*, "Fpgas-as-a-service toolkit (faast)," in *2020 IEEE/ACM International Workshop on Heterogeneous High-performance Reconfigurable Computing (H2RC)*, IEEE, Nov. 2020. DOI: 10.1109/h2rc51942.2020.00010. [Online]. Available: http://dx.doi.org/10.1109/H2RC51942.2020.00010.

[59] Y. Iiyama, G. Cerminara, A. Gupta, *et al.*, "Distance-weighted graph neural networks on fpgas for real-time particle reconstruction in high energy physics," *Frontiers in Big Data*, vol. 3, Jan. 2021, ISSN: 2624-909X. DOI: 10.3389/fdata.2020.598927. [Online]. Available: http://dx.doi.org/10.3389/fdata.2020.598927.

[60] Xilinx, *Vivado design suite user guide: High-level synthesis*, 2020. [Online]. Available: https://www.xilinx.com/support/documentation/sw_manuals/xilinx2020_1/ug902-vivado-high-level-synthesis.pdf.

[61] D. P. Kingma and M. Welling, *Auto-encoding variational bayes*, 2013. arXiv: 1312.6114 [stat.ML].

[62] N. Zipper, *Testing a neural network for anomaly detection in the cms global trigger test crate during run 3*, 2023. arXiv: 2312.10009 [hep-ex].

[63] C. Collaboration, "The cms experiment at the cern lhc," *Journal of Instrumentation*, vol. 3, S08004, 2008.

[64] C. Collaboration, "Development of the cms detector for the cern lhc run 3," CERN, Geneva, Tech. Rep. CMS-PRF-21-001, 2023, CMS-PRF-21-001-003, CERN-EP-2023-136.

[65] C. Collaboration, "The phase-2 upgrade of the cms level-1 trigger," CERN, Geneva, Tech. Rep. CERN-LHCC-2020-004, 2020, CMS-TDR-021.

[66] A. Heintz, V. Razavimaleki, J. Duarte, *et al.*, *Accelerated charged particle tracking with graph neural networks on fpgas*, 2020. arXiv: 2012.01563 [physics.ins-det].

[67] S. Summers, G. D. Guglielmo, J. Duarte, *et al.*, "Fast inference of boosted decision trees in fpgas for particle physics," *Journal of Instrumentation*, vol. 15, no. 05, P05026–P05026, May 2020, ISSN: 1748-0221. DOI: 10.1088/1748-0221/15/05/p05026. [Online]. Available: http://dx.doi.org/10.1088/1748-0221/15/05/p05026.

[68] C. Coelho, *Qkeras*, https://github.com/google/qkeras, 2019.

[69] S. Han, H. Mao, and W. J. Dally, *Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding*, 2016. arXiv: 1510.00149 [cs.CV].

[70] E. Meller, A. Finkelstein, U. Almog, and M. Grobman, "Same, same but different: Recovering neural network quantization error through weight factorization," in *Proceedings of the 36th International Conference on Machine Learning*, K. Chaudhuri and R. Salakhutdinov, Eds., PMLR, vol. 97, Long Beach, CA, USA, Jun. 2019, p. 4486. eprint: 1902.01917. [Online]. Available: http://proceedings.mlr.press/v97/meller19a.html.

[71] M. Nagel, M. van Baalen, T. Blankevoort, and M. Welling, "Data-free quantization through weight equalization and bias correction," in *2019 IEEE/CVF International Conference on Computer Vision*, Seoul, South Korea, Aug. 2019, p. 1325. eprint: 1906.04721.

[72] R. Zhao, Y. Hu, J. Dotzel, C. D. Sa, and Z. Zhang, "Improving neural network quantization without retraining using outlier channel splitting," in *Proceedings of the 36th International Conference on Machine Learning*, K. Chaudhuri and R. Salakhutdinov, Eds., PMLR, vol. 97, Long Beach, CA, USA, Jun. 2019, p. 7543. eprint: 1901.09504. [Online]. Available: http://proceedings.mlr.press/v97/zhao19c.html.

[73] Y. LeCun, J. S. Denker, and S. A. Solla, "Optimal brain damage," in *Advances in Neural Information Processing Systems 2*, D. S. Touretzky, Ed., Morgan-Kaufmann, 1990, p. 598.

[74] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding," in *4th International Conference on Learning Representations (ICLR) 2016, Conference Track Proceedings*, arXiv:1510.00149, 2016.

[75] C. Louizos, M. Welling, and D. P. Kingma, "Learning sparse neural networks through l0 regularization," in *6th International Conference on Learning Representations*, arXiv:1712.01312, vol. 12, 2018.

[76] T. Yang, Y. Chen, and V. Sze, "Designing energy-efficient convolutional neural networks using energy-aware pruning," in *CVPR*, arXiv:1611.05128, 2017.

[77] F. Chollet, "Keras," 2015. [Online]. Available: https://keras.io.

[78] A. Borghesi, A. Bartolini, M. Lombardi, M. Milano, and L. Benini, "Anomaly detection using autoencoders in high-performance computing systems," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 9428–9433. DOI: 10.1609/aaai.v33i01.33019428.

[79] J. Chow, Z. Su, J. Wu, P. Tan, X. Mao, and Y. Wang, "Anomaly detection of defects on concrete structures with the convolutional autoencoder," *Advanced Engineering Informatics*, vol. 45, p. 101 105, 2020. DOI: 10.1016/j.aei.2020.101105.

[80] J. Kolberg, M. Grimmer, M. Gomez-Barrero, and C. Busch, "Anomaly detection with convolutional autoencoders for fingerprint presentation attack detection," *IEEE Transactions on Biometrics, Behavior, and Identity Science*, vol. 3, no. 2, pp. 190–202, 2021. DOI: 10.1109/TBIOM.2021.3050036.

[81] D. P. Kingma and J. Ba, *Adam: A method for stochastic optimization*, 2017. arXiv: 1412.6980 [cs.LG].

[82] B. Zong, Q. Song, M. R. Min, *et al.*, "Deep autoencoding gaussian mixture model for unsupervised anomaly detection," in *International Conference on Learning Representations*, 2018.

[83] X. Wang, Y. Du, S. Lin, P. Cui, Y. Shen, and Y. Yang, "Advae: A self-adversarial variational autoencoder with gaussian anomaly prior knowledge for anomaly detection," *Knowledge-Based Systems*, vol. 190, p. 105 187, 2020. DOI: 10.1016/j.knosys.2019.105187.

[84] A. Alfeo, M. Cimino, G. Manco, E. Ritacco, and G. Vaglini, "Using an autoencoder in the design of an anomaly detector for smart manufacturing," *Pattern Recognition Letters*, vol. 136, pp. 272–278, 2020. DOI: 10.1016/j.patrec.2020.06.008.

[85] Z. Wang and Y.-J. Cha, "Unsupervised deep learning approach using a deep autoencoder with a one-class support vector machine to detect structural damage," *Structural Health Monitoring*, p. 1 475 921 720 934 051, 2020. DOI: 10.1177/1475921720934051.

[86] C. Ieracitano, A. Adeel, F. Morabito, and A. Hussain, "A novel statistical analysis and autoencoder-driven intelligent intrusion detection approach," *Neurocomputing*, vol. 387, pp. 51–62, 2020. DOI: 10.1016/j.neucom.2019.11.016.

[87] K. V. Dutta and M. Choras, "Hybrid model for improving the classification effectiveness of network intrusion detection," in *13th International Conference on Computational Intelligence in Security for Information Systems (CISIS 2020)*, 2019.

[88] H. Hojjati and N. Armanfard, "Dasvdd: Deep autoencoding support vector data descriptor for anomaly detection," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–12, 2024, ISSN: 2326-3865. DOI: 10.1109/tkde.2023.3328882. [Online]. Available: http://dx.doi.org/10.1109/TKDE.2023.3328882.

[89] A. Bartler, L. Mauch, B. Yang, M. Reuter, and L. Stoicescu, "Automated detection of solar cell defects with deep learning," in *2018 26th European Signal Processing Conference (EUSIPCO)*, IEEE, 2018, pp. 2035–2039.

[90] P. Schlachter, Y. Liao, and B. Yang, "Deep one-class classification using intra-class splitting," in *2019 IEEE Data Science Workshop (DSW)*, IEEE, 2019, pp. 100–104.

[91] L. Ruff, J. R. Kauffmann, R. A. Vandermeulen, *et al.*, "A unifying review of deep and shallow anomaly detection," *Proceedings of the IEEE*, vol. 109, no. 5, pp. 756–795, May 2021, ISSN: 1558-2256. DOI: 10.1109/jproc.2021.3052449. [Online]. Available: http://dx.doi.org/10.1109/JPROC.2021.3052449.

[92] D. Hendrycks, M. Mazeika, and T. G. Dietterich, "Deep anomaly detection with outlier exposure," *CoRR*, vol. abs/1812.04606, 2018. arXiv: 1812.04606. [Online]. Available: http://arxiv.org/abs/1812.04606.

[93] W. H. Smith, "Triggering at the LHC," *Annual Review of Nuclear and Particle Science*, vol. 66, pp. 123–141, 2016, Volume publication date October 2016. First published as a Review in Advance on June 01, 2016. © Annual Reviews. DOI: 10.1146/annurev-nucl-102115-044713. [Online]. Available: https://doi.org/10.1146/annurev-nucl-102115-044713.

[94] ATLAS Collaboration, *Atlas public results on trigger operation*, https://twiki.cern.ch/twiki/bin/view/AtlasPublic/TriggerOperationPublicResults, [Accessed: 7-October-2024], 2023.

[95] H. Bertelsen, G. C. Montoya, P.-O. Deviveiros, *et al.*, "Operation of the upgraded atlas central trigger processor during the lhc run 2," *Journal of Instrumentation*, vol. 11, no. 02, p. C02020, Feb. 2016. DOI: 10.1088/1748-0221/11/02/C02020. [Online]. Available: https://dx.doi.org/10.1088/1748-0221/11/02/C02020.

[96] A. Collaboration, *The atlas trigger system for lhc run 3 and trigger performance in 2022*, 2024. arXiv: 2401.06630 [hep-ex].

[97] A. Armbruster, G. Carrillo-Montoya, M. Chelstowska, *et al.*, "The atlas muon to central trigger processor interface upgrade for the run 3 of the lhc," in *2017 IEEE Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC)*, 2017, pp. 1–5. DOI: 10.1109/NSSMIC.2017.8532707.

[98] 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, ISSN: 1029-8479. DOI: 10.1007/jhep05(2019)036. [Online]. Available: http://dx.doi.org/10.1007/JHEP05(2019)036.

[99] A. Sirunyan, A. Tumasyan, W. Adam, *et al.*, "Performance of the cms level-1 trigger in proton-proton collisions at $\sqrt{s} = 13$ tev," *Journal of Instrumentation*, vol. 15, no. 10, P10017, 2020, © 2020 CERN for the benefit of the CMS collaboration. DOI: 10.1088/1748-0221/15/10/P10017. [Online]. Available: https://doi.org/10.1088/1748-0221/15/10/P10017.

[100] A. Paszke, S. Gross, F. Massa, *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, H. Wallach, Ed., Curran Associates, Inc., 2019, p. 8024.

[101] M. Abadi, A. Agarwal, P. Barham, *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous systems," 2015. arXiv: 1603.04467 [cs.DC]. [Online]. Available: https://tensorflow.org.

[102] B. Jacob, S. Kligys, B. Chen, *et al.*, "Quantization and training of neural networks for efficient integer-arithmetic-only inference," 2018. arXiv: 1712.05877 [cs.LG].

[103] M. Courbariaux, Y. Bengio, and J.-P. David, "Binaryconnect: Training deep neural networks with binary weights during propagations," in *Advances in Neural Information Processing Systems*, C. Cortes, Ed., vol. 28, MIT Press, 2015, p. 3123.

[104] N. Ghielmetti, V. Loncar, M. Pierini, *et al.*, "Real-time semantic segmentation on fpgas for autonomous vehicles with hls4ml," *Machine Learning: Science and Technology*, vol. 3, no. 4, p. 045 011, Nov. 2022. DOI: 10.1088/2632-2153/ac9cb5. [Online]. Available: https://dx.doi.org/10.1088/2632-2153/ac9cb5.

[105] Z. Chen, C. K. Yeo, B. S. Lee, and C. T. Lau, "Autoencoder-based network anomaly detection," in *2018 Wireless Telecommunications Symposium (WTS)*, 2018, pp. 1–5. DOI: 10.1109/WTS.2018.8363930.

[106] Y. Kawachi, Y. Koizumi, and N. Harada, "Complementary set variational autoencoder for supervised anomaly detection," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 2366–2370. DOI: 10.1109/ICASSP.2018.8462181.

[107] E. Prifti, J. P. Buban, A. S. Thind, and R. F. Klie, "Variational convolutional autoencoders for anomaly detection in scanning transmission electron microscopy," *Small*, 2023, Open Access. DOI: 10.1002/smll.202205977. [Online]. Available: https://doi.org/10.1002/smll.202205977.

[108] A. Asperti, D. Evangelista, and E. Loli Piccolomini, "A survey on variational autoencoders from a green ai perspective," *SN Computer Science*, vol. 2, no. 4, p. 301, 2021, ISSN: 2661-8907. DOI: 10.1007/s42979-021-00702-9. [Online]. Available: https://doi.org/10.1007/s42979-021-00702-9.

[109] I. Higgins, L. Matthey, A. Pal, *et al.*, "beta-VAE: Learning basic visual concepts with a constrained variational framework," in *Proc. ICLR*, 2017.

[110] C. Burgess, I. Higgins, A. Pal, *et al.*, "Understanding disentangling in beta-vae," 2018.

[111] I. Tolstikhin, O. Bousquet, S. Gelly, and B. Schoelkopf, *Wasserstein auto-encoders*, 2019. arXiv: 1711.01558 [stat.ML].

[112] Z. Zhang and X. Deng, "Anomaly detection using improved deep svdd model with data structure preservation," *Pattern Recognition Letters*, vol. 148, pp. 1–6, 2021, ISSN: 0167-8655. DOI: https://doi.org/10.1016/j.patrec.2021.04.020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167865521001598.

[113] D. Tax and R. Duin, "Support vector domain description," *Pattern Recognition Letters*, vol. 20, pp. 1191–1199, 1999.

[114] F. Bovolo, G. Camps-Valls, and L. Bruzzone, "A support vector domain method for change detection in multitemporal images," *Pattern Recognition Letters*, vol. 31, pp. 1148–1154, 2010.

[115] S. L. Glashow, "Partial-symmetries of weak interactions," *Nucl. Phys.*, vol. 22, pp. 579–588, 1961. DOI: 10.1016/0029-5582(61)90469-2.

[116] S. Weinberg, "A model of leptons," *Phys. Rev. Lett.*, vol. 19, pp. 1264–1266, 1967. DOI: 10.1103/PhysRevLett.19.1264.

[117] "Elementary particle theory: Relativistic groups and analyticity. proceedings of the eighth nobel symposium held may 19-25, 1968 at aspenäsgarden, lerum, in the county of älvsborg, sweden / edited by nils svartholm," Nobelstiftelsen, Stockholm : New York ; London, Tech. Rep., 1968.

[118] N. Wolchover, "A new map of all the particles and forces," *Quanta Mag.*, 2020.

[119] C. Quigg, "The double simplex," in *GUSTAVOFEST: Symposium in Honor of Gustavo C. Branco: CP Violation and the Flavor Puzzle*, hep-ph/0509037, 2005.

[120] P. Zyla, R. Barnett, J. J. Beringer, and O. Dahl, "Review of particle physics," *PTEP*, vol. 2020, p. 083C01, 2020. DOI: 10.1093/ptep/ptaa104.

[121] *Summary of the cross section measurements of standard model processes*, 2021.

[122] R. K. Ellis, W. J. Stirling, and B. R. Webber, *QCD and Collider Physics* (Cambridge Monographs on Particle Physics, Nuclear Physics and Cosmology). Cambridge University Press, 1996.

[123] T. Q. Nguyen, D. Weitekamp, D. Anderson, *et al.*, "Topology classification with deep learning to improve real-time event selection at the lhc," *Computational Software for Big Science*, vol. 3, no. 12, p. 1807.00083, 2019. eprint: 1807.00083.

[124] O. Knapp, G. Dissertori, O. Cerri, T. Q. Nguyen, J.-R. Vlimant, and M. Pierini, *Adversarially learned anomaly detection on cms open data: Re-discovering the top quark*, 2020. arXiv: 2005.01598 [hep-ex].

[125] E. Govorkova, E. Puljak, T. Aarrestad, M. Pierini, K. A. Woźniak, and J. Ngadiuba, "Unsupervised new physics detection at 40 mhz: $h^0 \to \tau\tau$ Signal benchmark dataset," *Zenodo*, 2021, https://doi.org/10.5281/zenodo.5061633. DOI: 10.5281/zenodo.5061633.

[126] E. Govorkova, E. Puljak, T. Aarrestad, M. Pierini, K. A. Woźniak, and J. Ngadiuba, "Unsupervised new physics detection at 40 mhz: $h^+ \to \tau\nu$ Signal benchmark dataset," *Zenodo*, 2021, https://doi.org/[PleaseaddthespecificDOIforthisdataset]. DOI: 10.5281/zenodo.[PleaseaddthespecificDOIforthisdataset].

[127] E. Govorkova, E. Puljak, T. Aarrestad, M. Pierini, K. A. Woźniak, and J. Ngadiuba, "Unsupervised new physics detection at 40 mhz: $LQ \to b\tau$ Signal benchmark dataset," *Zenodo*, 2021, https://doi.org/10.5281/zenodo.5055454. DOI: 10.5281/zenodo.5055454.

[128] E. Govorkova, E. Puljak, T. Aarrestad, M. Pierini, K. A. Woźniak, and J. Ngadiuba, "Unsupervised new physics detection at 40 mhz: $A \to 4$ Leptons signal benchmark dataset," *Zenodo*, 2021, https://doi.org/10.5281/zenodo.5046446. DOI: 10.5281/zenodo.5046446.

[129] T. E. Govorkova Puljak Aarrestad, M. Pierini, K. Woźniak, and J. Ngadiuba, *Unsupervised new physics detection at 40 mhz: Black box dataset*, Zenodo, 2021. DOI: 10.5281/zenodo.5070455. [Online]. Available: https://doi.org/10.5281/zenodo.5070455.

[130] T. E. Govorkova Puljak Aarrestad, M. Pierini, K. Woźniak, and J. Ngadiuba, *Unsupervised new physics detection at 40 mhz: Training dataset*, https://doi.org/10.5281/zenodo.5046389, 2021. DOI: 10.5281/zenodo.5046389.

[131] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32nd International Conference on Machine Learning*, 1502.03167, PMLR, vol. 37, 2015, p. 448. [Online]. Available: http://proceedings.mlr.press/v37/ioffe15.html.

[132] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *ICML Workshop on Deep Learning for Audio, Speech and Language Processing*, 2013. [Online]. Available: https://.

[133] J. M. Joyce, *Kullback-Leibler Divergence*. Springer Berlin Heidelberg, 2011, pp. 720–722. [Online]. Available: https://doi.org/10.1007/978-3-642-04898-2_327.

[134] A. P. Bradley, "The use of the area under the roc curve in the evaluation of machine learning algorithms," *Pattern Recognition*, vol. 30, no. 7, pp. 1145–1159, 1997, ISSN: 0031-3203. DOI: https://doi.org/10.1016/S0031-3203(96)00142-2. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0031320396001422.

[135] TensorFlow, *Pruning comprehensive guide*, https://www.tensorflow.org/model_optimization/guide/pruning/comprehensive_guide, Accessed: 2024-06-05, 2024.

[136] F. Fahim, B. Hawks, C. Herwig, *et al.*, *Hls4ml: An open-source codesign workflow to empower scientific low-power machine learning devices*, 2021. arXiv: 2103.05579 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2103.05579.

[137] O. Penc, "Atlas level-1 trigger menu testing," in *42nd International Conference on High Energy Physics (ICHEP2024)*, Presented on behalf of the L1 Central Trigger group (F. Bonini, S. Haas, A. Koulouris, A. Kulinska, L. Marsella, A. Marzin, K. Mihule, T. Pauly, O. Penc, V. Ryjov, L. Sanfilippo, R. Simoniello, R. Spiwoks, P. Vichoudis, and T. Wengler), L1 Central Trigger group, Prague, Jul. 17–24, 2024.

[138] J. Kočka and O. Penc, "New interface for atlas l1 trigger menu testing," ATLAS Collaboration, Summer Student Project, 2023, Supervised by Ondřej Penc.

[139] D. Binu and B. R. Rajakumar, Eds., *Artificial Intelligence in Data Mining: Theories and Applications*. Elsevier, 2021.

[140] E. Taylor and J. Wheeler, *Spacetime Physics: Introduction to Special Relativity*. New York: W. H. Freeman and Company, 1992, p. 191, ISBN: 978-0-7167-2327-1.

[141] M. Cacciari, G. P. Salam, and G. Soyez, "The anti-k_t jet clustering algorithm," *JHEP*, vol. 04, p. 063, 2008. DOI: 10.1088/1126-6708/2008/04/063. arXiv: 0802.1189.

[142] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, vol. 12, no. 61, pp. 2121–2159, 2011.

[143] O. N. N. E. Collaboration. "Onnx." (2017), [Online]. Available: https://onnx.ai/.

[144] S. Han, J. Pool, J. Tran, and W. J. Dally, "Learning both weights and connections for efficient neural networks," in *Advances in Neural Information Processing Systems 28 (NIPS 2015)*, arXiv:1506.02626, 2015.

[145] S. Shin, Y. Boo, and W. Sung, "Knowledge distillation for optimization of quantized deep neural networks," in *2020 IEEE Workshop on Signal Processing Systems (SiPS)*, 2020, p. 1.

[146] A. Polino, R. Pascanu, and D. Alistarh, "Model compression via distillation and quantization," in *International Conference on Learning Representations*, 2018. [Online]. Available: https://openreview.net/forum?id=S1XolQbRW.

[147] M. Gao, Y. Shen, Q. Li, C. C. Loy, and X. Tang, "An embarrassingly simple approach for knowledge distillation," 2019. eprint: 1812.01819.

[148] A. Mishra and D. Marr, "Apprentice: Using knowledge distillation techniques to improve low-precision network accuracy," in *International Conference on Learning Representations*, 2018. [Online]. Available: https://openreview.net/forum?id=B1ae1lZRb.

[149] T. Golling, T. Nobe, D. Proios, *et al.*, *The mass-ive issue: Anomaly detection in jet physics*, 2023. arXiv: 2303.14134 [hep-ph].

[150] P. Jawahar, T. Aarrestad, N. Chernyavskaya, *et al.*, "Improving variational autoencoders for new physics detection at the lhc with normalizing flows," *Frontiers in Big Data*, vol. 5, 2022, ISSN: 2624-909X. DOI: 10.3389/fdata.2022.803685. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fdata.2022.803685.

[151] G. Kasieczka, B. Nachman, and D. Shih, *New methods and datasets for group anomaly detection from fundamental physics*, 2021. arXiv: 2107.02821 [stat.ML].

[152] G. Aad, T. Abajyan, B. Abbott, *et al.*, "Observation of a new particle in the search for the standard model higgs boson with the atlas detector at the lhc," *Phys. Lett. B*, vol. 716, pp. 1–29, 2012.

[153] D. Zimmerer, S. A. A. Kohl, J. Petersen, F. Isensee, and K. H. Maier-Hein, *Context-encoding variational autoencoder for unsupervised anomaly detection*, 2018. arXiv: 1812.05941 [cs.LG].

[154] H. Xu, W. Chen, N. Zhao, *et al.*, "Unsupervised anomaly detection via variational auto-encoder for seasonal kpis in web applications," in *Proceedings of the 2018 World Wide Web Conference*, ser. WWW '18, Lyon, France: International World Wide Web Conferences Steering Committee, 2018, pp. 187–196, ISBN: 9781450356398. DOI: 10.1145/3178876.3185996. [Online]. Available: https://doi.org/10.1145/3178876.3185996.

[155] J. An and S. Cho, "Variational autoencoder based anomaly detection using reconstruction probability," 2015. [Online]. Available: https://api.semanticscholar.org/CorpusID:36663713.

[156] A. Sidoti, "Minimum bias trigger scintillators in atlas run ii," *JINST*, vol. 9, no. 2014, p. C10020,

[157] L. Adamczyk, E. Banaś, A. Brandt, and M. Bruschi, "Technical design report for the atlas forward proton detector," CERN-LHCC-2015-009, ATLAS-TDR-024, Tech. Rep., 2015. [Online]. Available: https://cds.cern.ch/record/2017378.

[158] S. A. Khalek, B. Allongue, F. Anghinolfi, *et al.*, "The alfa roman pot detectors of atlas," *Journal of Instrumentation*, vol. 11, no. 11, P11013–P11013, Nov. 2016, ISSN: 1748-0221. DOI: 10.1088/1748-0221/11/11/p11013. [Online]. Available: http://dx.doi.org/10.1088/1748-0221/11/11/P11013.

[159] G. Avoni, M. Bruschi, G. Cabras, *et al.*, "The new lucid-2 detector for luminosity measurement and monitoring in atlas," *Journal of Instrumentation*, vol. 13, no. 07, P07017, Jul. 2018. DOI: 10.1088/1748-0221/13/07/P07017. [Online]. Available: https://dx.doi.org/10.1088/1748-0221/13/07/P07017.

[160] ATLAS Collaboration, "Luminosity determination in pp collisions at $\sqrt{s} = 13$ tev using the atlas detector at the lhc," *Eur. Phys. J. C*, vol. 83, p. 982, 2023. arXiv: 2212.09379 [hep-ex].

[161] ATLAS Collaboration, "Zero degree calorimeters for atlas," CERN-LHCC-2007-01, Tech. Rep., 2007. [Online]. Available: https://cds.cern.ch/record/1009649/.

[162] K. Zhao, F. Jia, and H. Shao, "A novel conditional weighting transfer wasserstein auto-encoder for rolling bearing fault diagnosis with multi-source domains," *Knowledge-Based Systems*, vol. 262, p. 110 203, 2023, ISSN: 0950-7051. DOI: https://doi.org/10.1016/j.knosys.2022.110203. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950705122012990.

[163] S. Kolouri, G. K. Rohde, and H. Hoffmann, "Sliced wasserstein distance for learning gaussian mixture models," *CoRR*, vol. abs/1711.05376, 2017. arXiv: 1711.05376. [Online]. Available: http://arxiv.org/abs/1711.05376.

[164] M. Arjovsky, S. Chintala, and L. Bottou, *Wasserstein gan*, 2017. arXiv: 1701.07875 [stat.ML].

[165] S. Kolouri, C. E. Martin, and G. K. Rohde, "Sliced-wasserstein autoencoder: An embarrassingly simple generative model," *CoRR*, vol. abs/1804.01947, 2018. arXiv: 1804.01947. [Online]. Available: http://arxiv.org/abs/1804.01947.

[166] Q. Luo, J. Chen, Y. Zi, Y. Chang, and Y. Feng, "Multi-mode non-gaussian variational autoencoder network with missing sources for anomaly detection of complex electromechanical equipment," *ISA Transactions*, vol. 134, pp. 144–158, 2023, ISSN: 0019-0578. DOI: https://doi.org/10.1016/j.isatra.2022.09.009. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0019057822004669.

[167] Z. Li and M. van Leeuwen, "Explainable contextual anomaly detection using quantile regression forests," *Data Mining and Knowledge Discovery*, vol. 37, pp. 2517–2563, 2023. DOI: 10.1007/s10618-023-00967-z.

[168] A. Thin, N. Kotelevskii, A. Doucet, A. Durmus, E. Moulines, and M. Panov, *Monte carlo variational auto-encoders*, 2021. arXiv: 2106.15921 [stat.ML].

[169] C. F. Ciușdel, L. M. Itu, S. Cimen, *et al.*, "Normalizing flows for out-of-distribution detection: Application to coronary artery segmentation," *Applied Sciences*, vol. 12, no. 8, 2022, ISSN: 2076-3417. DOI: 10.3390/app12083839. [Online]. Available: https://www.mdpi.com/2076-3417/12/8/3839.

[170] Y. Su, Y. Zhao, M. Sun, *et al.*, "Detecting outlier machine instances through gaussian mixture variational autoencoder with one dimensional cnn," *IEEE Transactions on Computers*, vol. 71, no. 4, pp. 892–905, 2022. DOI: 10.1109/TC.2021.3065073.

[171] A. Paszke, S. Gross, F. Massa, *et al.*, *Pytorch: An imperative style, high-performance deep learning library*, 2019. arXiv: 1912.01703 [cs.LG].

[172] R. Brun and F. Rademakers, "Root - an object oriented data analysis framework," *Nucl. Inst. & Meth. in Phys. Res. A*, vol. 389, pp. 81–86, 1997, Proceedings AIHENP'96 Workshop, Lausanne, Sep. 1996.

[173] A. E. Bayer, U. Seljak, and J. Robnik, "Self-calibrating the look-elsewhere effect: fast evaluation of the statistical significance using peak heights," *Monthly Notices of the Royal Astronomical Society*, vol. 508, no. 1, pp. 1346–1357, Sep. 2021, ISSN: 0035-8711. DOI: 10.1093/mnras/stab2331. eprint: https://academic.oup.com/mnras/article-pdf/508/1/1346/40508999/stab2331.pdf. [Online]. Available: https://doi.org/10.1093/mnras/stab2331.

[174] S. Algeri, D. van Dyk, J. Conrad, and B. Anderson, "On methods for correcting for the look-elsewhere effect in searches for new physics," *Journal of Instrumentation*, vol. 11, no. 12, P12010, Dec. 2016. DOI: 10.1088/1748-0221/11/12/P12010. [Online]. Available: https://dx.doi.org/10.1088/1748-0221/11/12/P12010.

[175] J. Doe. "An interesting online article." (2022), [Online]. Available: https://www.example.com/article.

[176] D. Addo, S. Zhou, J. K. Jackson, *et al.*, "Evae-net: An ensemble variational autoencoder deep learning network for covid-19 classification based on chest x-ray images," *Diagnostics*, vol. 12, no. 11, 2022, ISSN: 2075-4418. DOI: 10.3390/diagnostics12112569. [Online]. Available: https://www.mdpi.com/2075-4418/12/11/2569.

Appendix A

Vivado HLS Report for LOE VAE

A.1 Synthesis Report

```
================================================================
== Vivado HLS Report for 'myproject'
================================================================
* Date:           Fri Jul 26 16:29:31 2024

* Version:        2020.1 (Build 2897737 on Wed May 27 20:21:37 MDT 2020)
* Project:        myproject_prj
* Solution:       solution1
* Product family: virtexuplus
* Target device:  xcvu9p-flgb2104-2-e


================================================================
== Performance Estimates
================================================================
+ Timing:
* Summary:
+--------+---------+----------+------------+
|  Clock |  Target | Estimated| Uncertainty|
+--------+---------+----------+------------+
|ap_clk  | 5.00 ns | 4.330 ns |   0.62 ns  |
+--------+---------+----------+------------+

+ Latency:
* Summary:
+---------+---------+-----------+-----------+-----+-----+----------+
|  Latency (cycles) |   Latency (absolute)  |  Interval | Pipeline |
|   min   |   max   |    min    |    max    | min | max |   Type   |
+---------+---------+-----------+-----------+-----+-----+----------+
|        5|        5| 25.000 ns | 25.000 ns |    1|    1| function |
+---------+---------+-----------+-----------+-----+-----+----------+
```

```
+ Detail:
* Instance:
+-----------------------------------------------------------------------------------------------------+
|                                                                                                     |
|                                              Instance                                               |
+-----------------------------------------------------------------------------------------------------+
|grp_dense_latency_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_fu_82    |
|call_ret3_dense_latency_ap_fixed_ap_fixed_config7_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_fu_136   |
|call_ret_normalize_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_s_fu_172|
|call_ret2_leaky_relu_ap_fixed_ap_fixed_10_6_5_3_0_LeakyReLU_config6_s_fu_178                         |
|call_ret6_dense_latency_ap_fixed_ap_fixed_16_6_5_3_0_config13_0_0_0_fu_214                           |
|call_ret5_dense_latency_ap_fixed_ap_fixed_16_6_5_3_0_config11_0_0_0_fu_227                           |
|call_ret4_leaky_relu_ap_fixed_ap_fixed_10_6_5_3_0_LeakyReLU_config10_s_fu_243                        |
+-----------------------------------------------------------------------------------------------------+

* Instance:
+-----------------------------------------------+---------+---------+----------+----------+-----+-----+----------+
|                                               |  Latency (cycles) |  Latency (absolute) |  Interval | Pipeline |
|                    Module                     |   min   |   max   |    min   |    max   | min | max |   Type   |
+-----------------------------------------------+---------+---------+----------+----------+-----+-----+----------+
|dense_latency_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_ |        1|        1| 5.000 ns | 5.000 ns |    1|    1| function |
|0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0  |         |         |          |          |     |     |          |
|dense_latency_ap_fixed_ap_fixed_config7_0_0_   |        0|        0|   0 ns   |   0 ns   |    1|    1| function |
|0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0  |         |         |          |          |     |     |          |
|normalize_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_ |        0|        0|   0 ns   |   0 ns   |    1|    1| function |
|0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_s|         |         |          |          |     |     |          |
|leaky_relu_ap_fixed_ap_fixed_10_6_5_3_0_Leak   |        0|        0|   0 ns   |   0 ns   |    1|    1| function |
|yReLU_config6_s                                |         |         |          |          |     |     |          |
|dense_latency_ap_fixed_ap_fixed_16_6_5_3_0_c   |         |         |          |          |     |     |          |
|onfig13_0_0_0                                  |        0|        0|   0 ns   |   0 ns   |    1|    1| function |
|dense_latency_ap_fixed_ap_fixed_16_6_5_3_0_c   |         |         |          |          |     |     |          |
|onfig11_0_0_0                                  |        0|        0|   0 ns   |   0 ns   |    1|    1| function |
|leaky_relu_ap_fixed_ap_fixed_10_6_5_3_0_Leak   |         |         |          |          |     |     |          |
|yReLU_config10_s                               |        0|        0|   0 ns   |   0 ns   |    1|    1| function |
-+----------------------v----------------------+---------+---------+----------+----------+-----+-----+----------+

* Loop:
N/A

================================================================
== Utilization Estimates
================================================================
* Summary:
+---------------------+---------+-------+---------+---------+-----+
|         Name        | BRAM_18K| DSP48E|    FF   |   LUT   | URAM|
+---------------------+---------+-------+---------+---------+-----+
|DSP                  |        -|      -|        -|        -|    -|
|Expression           |        -|      -|        0|        6|    -|
|FIFO                 |        -|      -|        -|        -|    -|
|Instance             |        -|     56|     1818|    40710|    -|
|Memory               |        -|      -|        -|        -|    -|
|Multiplexer          |        -|      -|        -|       36|    -|
|Register             |        -|      -|     2103|        -|    -|
+---------------------+---------+-------+---------+---------+-----+
|Total                |        0|     56|     3921|    40752|    0|
```

Available SLR	1440	2280	788160	394080	320
Utilization SLR (%)	0	2	~0	10	0
Available	4320	6840	2364480	1182240	960
Utilization (%)	0	~0	~0	3	0

+ Detail:
* Instance:

Instance
grp_dense_latency_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_fu_82
call_ret5_dense_latency_ap_fixed_ap_fixed_16_6_5_3_0_config11_0_0_0_fu_227
call_ret6_dense_latency_ap_fixed_ap_fixed_16_6_5_3_0_config13_0_0_0_fu_214
call_ret3_dense_latency_ap_fixed_ap_fixed_config7_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_fu_136
call_ret4_leaky_relu_ap_fixed_ap_fixed_10_6_5_3_0_LeakyReLU_config10_s_fu_243
call_ret2_leaky_relu_ap_fixed_ap_fixed_10_6_5_3_0_LeakyReLU_config6_s_fu_178
call_ret_normalize_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_s_fu_172
Total

Module	BRAM_18K	DSP48E	FF	LUT	URAM
dense_latency_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0	0	0	1818	30574	0
dense_latency_ap_fixed_ap_fixed_16_6_5_3_0_config11_0_0_0	0	0	0	479	0
dense_latency_ap_fixed_ap_fixed_16_6_5_3_0_config13_0_0_0	0	0	0	491	0
dense_latency_ap_fixed_ap_fixed_config7_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0	0	0	0	6606	0
leaky_relu_ap_fixed_ap_fixed_10_6_5_3_0_LeakyReLU_config10_s	0	14	0	322	0
leaky_relu_ap_fixed_ap_fixed_10_6_5_3_0_LeakyReLU_config6_s	0	32	0	736	0
normalize_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_s	0	10	0	1502	0
	0	56	1818	40710	0

Appendix B

Alternative options explored

B.1 Models, Descriptions and Performance

1. **Model Name:** GM VAE

   - **Description:** The VAE is notably effective model in the AD landscape. However, complex physics signals may not follow Gaussian distributions, posing challenges for traditional Gaussian VAEs. To capture more meaningful latent distributions in complex data environments, a GM VAE can be employed. This model represents the overall population as a mixture of multiple Gaussian-distributed sub-populations, allowing it to effectively handle data with diverse underlying distributions [139]. This approach involves mapping the input to a Gaussian mixture distribution in the latent space using an encoder, and then sampling from this distribution for reconstruction [166].
   - **Performance:** Referring to Table B.1, the GM VAE algorithm exhibited inferior performance compared to the standard DNN VAE, as detailed in Table 5.1, particularly concerning the TPR at a FPR of 10^{-5}. In contrast, the AUC metric reflects comparable performance between the GM VAE and the standard VAE.
   - **Reason for not choosing:** Despite the competitive AUC metric, the TPR performance at low FPR rates renders the model unsuitable for the LHC trigger environment, where maintaining a low FPR at the desired TPR is essential to prevent over-triggering.

2. **Model Name:** Latent*Out* VAE

   - **Description:** AD methods utilizing AEs have demonstrated strong performance; however, deep non-linear architectures can achieve significant dimensionality reduction while maintaining low reconstruction error. This characteristic may inadvertently diminish the outlier detection capabilities of AEs. To address this issue, Angiulli et al. [54] demonstrate that outliers are often found in the sparsest regions of the combined latent/error space. They propose new unsupervised anomaly detection algorithms known as Latent*Out*, which identify outliers by estimating density within this augmented feature space. This algorithm was applied to a VAE feature space to enhance AD performance.
   - **Performance:** The Latent*Out* algorithm did not demonstrate significant performance improvements over the standard VAE. Additionally, due to the computational demands of density estimations, it required considerably more time to execute compared to the use of KL or KL_μ metrics. Consequently, the algorithm's substantial execution time and resource requirements hindered its full implementation, resulting in the absence of finalized results.
   - **Reason for not choosing:** The algorithm increased the time required for AD calculations from seconds to minutes, rendering it unsuitable for the low-latency environment of the LHC trigger. Furthermore, when implemented on an FPGA, the algorithm would demand significant resources, which is not feasible given the limited bandwidth in the L1T.

3. **Model Name:** SWAE [111], [165]

   - **Description:** The Wasserstein Auto Encoders (WAE) is a variant of the AE that minimizes the Wasserstein distance between the encoded data distribution and a specified target distribution. In a VAE, the goal is to ensure that the distribution $Q(Z \mid X = x)$ aligns with the distribution P_Z for every input example x that is sampled from the distribution P_X. However, a WAE compels the continuous mixture $Q_Z = \int Q(Z \mid X) dP_X$ to match P_Z. This approach allows latent codes of different examples to remain distant from each other, promoting better reconstruction. By providing a more flexible and accurate latent space representation, WAE can better distinguish between normal and anomalous data, enhancing AD performance. However, Directly computing the full Wasserstein distance in high-dimensional spaces can be computationally prohibitive due to the complexity of solving the optimal transport problem. Instead, the Sliced Wasserstein Distance (SWD) can be used, which

approximates the Wasserstein distance by projecting high-dimensional data onto one-dimensional subspaces and then computing the Wasserstein distance in these subspaces. This significantly reduces computational complexity and is more scalable for large datasets. This approach allows for any samplable latent distribution to be tested. In this experiment, a Gaussian distribution was employed. By selecting a continuous Gaussian as the latent distribution, anomalies are expected to manifest in the outskirts of the distribution. This characteristic facilitates a straightforward and effective approach to AD.

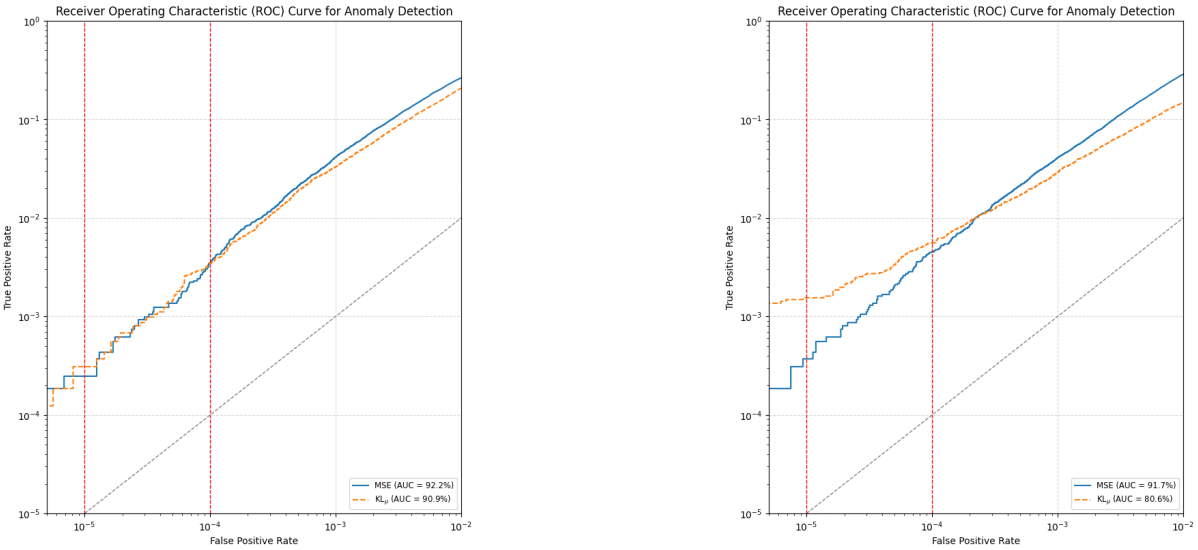
- **Performance:** The SWAE exhibited reasonable performance concerning the TPR at a FPR of 10^{-5} when employing the KL_μ metric, but demonstrated poor performance when assessed using the MSE. This is illustrated in Table B.1 below. Additionally, the AUC metric indicates comparable performance between the SWAE and the standard VAE.
- **Reason for not choosing:** Despite demonstrating a competitive AUC metric and a TPR performance at low FPR rates compared to the standard DNN VAE, the results did not approach state-of-the-art performance, which was the objective of the paper. The state-of-the-art is estimated to achieve a TPR of 0.2459% at a FPR of 10^{-5} [9].

B.2 Results

The following models were tested on the same mixed dataset with the contamination ratio set to 0.01.

Table B.1: Performance assessment of alternative models

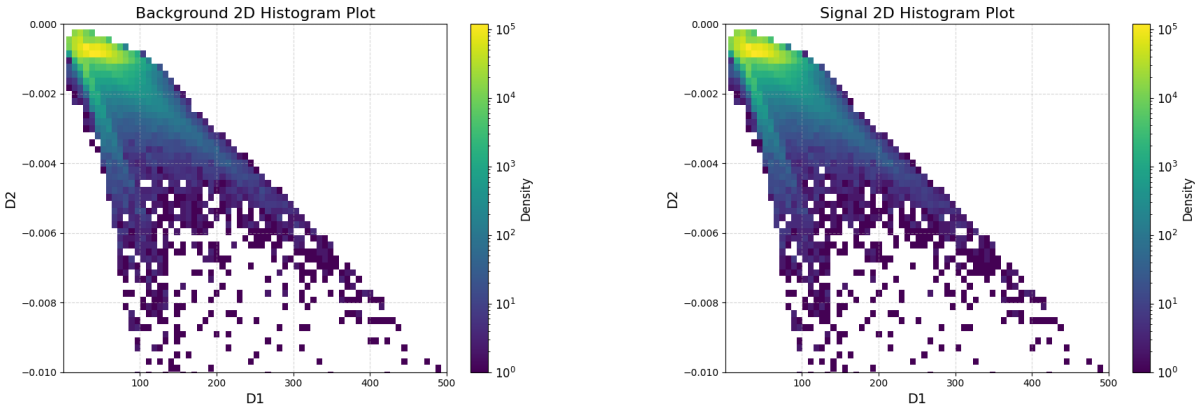
Model		Performance Metric	
Architecture	AD score	TPR @ FPR 10^{-5} [%]	AUC [%]
GM VAE	MSE	0.0248	92.2
GM VAE	KL_μ	0.0186	90.9
SWAE	MSE	0.0309	91.7
SWAE	KL_μ	0.1485	80.6



(a) GM VAE (b) SWAE

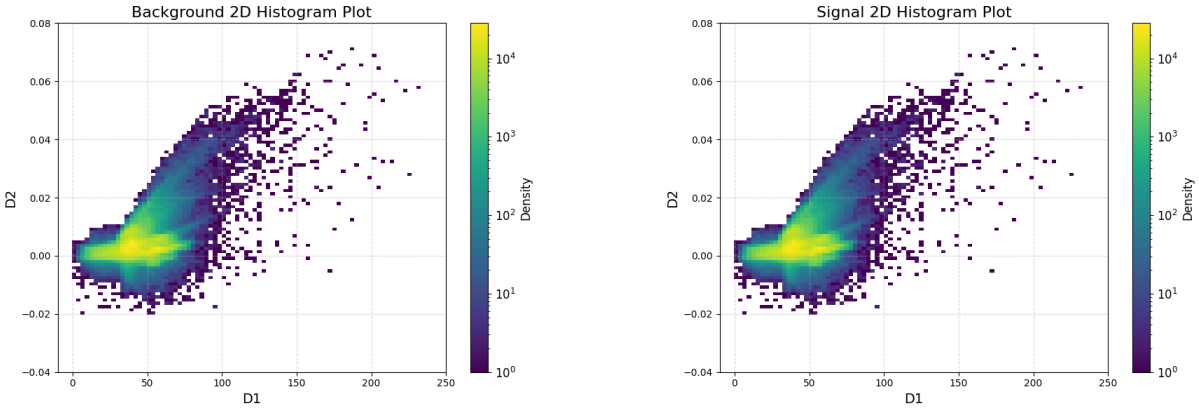
Figure B.1: AUC curves of alternate models under varying AD metrics. The red line in each plot indicates the TPR @ FPR 10^{-5}.

B.2.1 Latent Distributions



(a) z encodings - Background (b) z encodings - Signal

Figure B.2: Distribution of latent encodings for the GM VAE. (a) and (b) show the distributions of the encodings passed to the decoder network.



(a) z encodings - Background

(b) z encodings - Signal

Figure B.3: Distribution of latent encodings for the SWAE. (a) and (b) show the distributions of the encodings passed to the decoder network.

Appendix C

Code and Data availability

C.1 Code

1. The QKeras library is available at: `https://github.com/google/qkeras`
2. The hls4ml library is available at: `https://github.com/fastmachinelearning/hls4ml`
3. The models developed in this paper are available at: `https://gitlab.com/TSpank/unsupervised-anomaly-detection-for-the-atlas-level-1-trigger`

C.2 Data

The datasets used in this work are openly available at Zenodo at Ref. [125]–[128].