UCLouvain

FACULTE DES SCIENCES

Faculté des sciences

Applicability of nonparametric data-driven background modelling using conditional probabilities for CMS data analysis

Mémoire présenté en vue de l'obtention du grade académique de Master [120] en sciences physiques, finalité approfondie

Auteur : Kaczmarczyk Brieux
Promoteurs : Bethani Agni, Delaere Christophe
Lecteurs : de Wasseige Gwenhaël, Giammanco Andrea
École de physique
Année académique 2023-2024

Acknowledgements

I would like to express my heartfelt thanks to my two supervisors, Agni and Christophe, for their unwavering support and assistance throughout the completion of this project. Their presence allowed me to approach this thesis with minimal stress and pressure, knowing that they would be there to support and advise me at any moment of doubt. Over the course of this project, I realized that the little pressure I felt was the kind that motivates improvement and makes you strive to make those invested in you proud, rather than the stress-inducing kind. I also appreciate their accessibility and relaxed attitude, which helped me view this thesis more as a truly personal project rather than a course unit worth 26 credits. I sincerely believe that I could not have had better supervisors.

I am very grateful to Florian, who provided his assistance and advice for this project, especially when I was in a dead-end. I also want to thank him for his help throughout my physics studies, particularly for organizing the trip to CERN. It is thanks to his guidance (or propaganda, pick one) that I was able to explore and learn more about such a fascinating area of physics.

I would like to thank my two reviewers, Gwenhaël and Andrea, for their time, advice, and insightful comments on my project. Additionally, I want to thank Andrea, even though he might not be aware, for his help on the topic of top quarks. I frequently referred to articles he wrote, which greatly enhanced my understanding of the subject. I believe that good physicists are common, but good physicists who can clearly explain their work are very rare and truly indispensable.
Furthermore, I want to thank Gwenhaël for the various courses she taught during my master's, which presented fresh perspectives and innovative teaching methods. It is also important for me to thank her for her supervision during the personal project at the end of my bachelor's degree. The scientific curiosity she instilled in me during this project allowed me to discover a new area of physics previously unknown to me.

I also want to thank Oğuz for his help and availability during the final stages of the project. I would not have been as proud of my work if it weren't for his support.

I am deeply grateful to my mother for her unwavering support, which extends far beyond the completion of this thesis. She often praised my resilience and perseverance, but these qualities are merely the logical continuation of everything she has given me and done for me over all these years.

I also want to thank Eduardo, Justin, and Sofiane for their support and assistance, which, once again, goes far beyond the completion of this thesis.

I would also like to thank all the friends I met during my physics studies: Chloé, Alexandre, Simon, Hélène, Arthur, Sarah and many others. These five years have not always been easy, but their presence and team spirit have been an indispensable factor in my success.

Lastly, I want to thank the Kot Astro team. Although they disrupted some of my potentially productive mornings, I sincerely believe that this work would not have been as well-executed without their presence.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The interest in studying Higgs pair production is motivated by two aspects. Firstly, this process, involving the coupling of the Higgs boson to itself ($H \rightarrow HH$), offers a unique way to probe the Higgs potential, potentially shedding light on questions relative to the fabric of our Universe. Secondly, Higgs pair production is an extremely rare phenomenon within the Standard Model, rendering it highly sensitive to potential new physics phenomena, especially at unprecedented energy scales. Additionally, the exploration of extended scalar sectors, as predicted by many models beyond the Standard Model, further underscores the significance of studying processes like Higgs pair production. However, this investigation encounters formidable challenges posed by overwhelming backgrounds from standard model processes.

All these experiments are being carried out inside particle accelerators and colliders. These are essential tools in the field of high-energy physics, providing researchers with the means to probe the fundamental constituents of matter and light the mysteries of the universe. Among these, the Large Hadron Collider (LHC) stands out as the most powerful particle accelerator ever built. It accelerates beams of protons to unprecedented energies and collides them head-on at several interaction points, where massive detectors such as the Compact Muon Solenoid (CMS) are positioned.

Nevertheless, the lifetimes of some of these particles are so short that they cannot even reach the first layer of our particle detectors. Hence, their decay products are our only tool to get informations of the presence of particles such as bosons. However, with the multitude of events happening inside a particle detector, the final state signature we are interested in could be faked in many different way. Thus, in high energy physics, distinguishing the background from the signal is a task of an extreme importance. At the moment, the strategy used to filter out this background suffers from different shortcomings. Resulting in a imperfect filtering of background events, and indeed, a mediocre isolation of the targeted final state signature. The goal of this project is to introduce a new and, potentially more effective, background modelling strategy in order to improve the performance of current analysis of the CMS data.

Unfortunately, such as task cannot be succesfully completed by mere human understanding. We will have to use a very powerful tool in order to process the humongous quantity of data we are facing: neural networks. To be more specific, deep

neural networks will be the key to such a problem. These have already proved their worth in the field of high energy physics through modeling, signal identification and more. In this thesis, I will use a specific architecture of deep neural networks: the conditional Generative Adversarial Network (cGAN). This architecture is a variant of the standard GAN, where two deep neural networks are engaged in a competitive learning process. On one hand the generator learns to produce synthetic samples similar to the background distribution, while the discriminator learns to distinguish real data from synthetic ones. By training a cGAN on samples of background events, we aim to develop a sophisticated understanding of the underlying patterns and correlations, allowing us to generate fake but plausible samples reproducing the statistical properties of the background. In order to discriminate this background from the signal more efficiently in the future experiments.

In this thesis, we will start by a succinct explanation of the Standard Model, detailing both its remarkable successes and its recognized shortcomings. We will briefly explore how these deviations from the Standard Model motivate the needs of beyond Standard Model physics. One such theory is known as the 2HDM will be delved for its potential to address multiple of these shortcomings.
Then, we will explore one of the numerous available ways to obtain evidence supporting the proposed theory. We will list the common problems associated with other options and explain why the chosen option appears to be a great trade-off. The current state of research in this field will be discussed, highlighting why it might seem to be at a dead-end and how the approach discussed in this report might pave a new way forward.

Once this theoretical work about physics done, it will be time to dive into the realm of deep neural networks. We will break down the fundamental concepts, all the process leading to the final architecture of the network and finally a short summary of the former. Armed with this technical understanding, a detailed development of the network will be done. We will disucss the main obstacles encountered and how they were overcome. Then, the main results will be presented and discussed.

We will conclude this small chapter in the vast book of Physics by describing what this method has achieved so far, outlining its hard limitations, and exploring the different tools that could be used to address these challenges. By doing so, we aim to illustrate how this method could help others continue writing the forthcoming chapters in the ongoing story of physics.

Chapter 2

Two Higgs Doublet Model (2HDM)

Since its formulation in the mid 70's, the Standard Model (SM) has achieved numerous accomplishements. From the beginning, it has been able to describe three of the fundamental forces being: the electromagnetic, weak and strong interactions ; aswell as classifying all known particles at that time. Since then, the evidence of other particles predicted by the SM such as: W/Z bosons (1983), top quark (1995), tau neutrino (2000) and the Higgs boson (2012) have added further confidence to this theory.

Despite all these succesful achievements, the SM has proven multiple times several shortcomings. The lack of gravitation in its formulation, translated by the incompatibility to reunite the SM with the most succesful theory about gravitation known: the General Relativity. To this day, several experiments have observed neutrinos oscillations, implying the existence of massive neutrinos; which is not part of the original SM. Same goes for the baryon asymmetry or for the existence of dark matter and dark energy. Thus, the necessity of new physics is obvious.

Despite these issues, the SM remains a very solid framework, exhibiting a wide range of phenomena, including spontaneous symmetry breaking, anomalies, and non-perturbative behavior. It can and should be used as a basis for building more exotic models that incorporate hypothetical particles, extra dimensions, and elaborate symmetries to explain experimental results at variance with the SM. Hence, the Standard Model will be the giant's shoulders on which we will stand.

The field of beyond-SM theories is broad with theories such as supersymmetry, String theories or loop quantum gravity which are notable attempts to build a *Theory of everything*. However, we won't be as presumptuous in this work, we will focus on a simpler but yet, promising extension of the SM: the two Higgs Doublet Model (2HDM). As its name suggests, this model contains one more Higgs doublet than the SM, which leads to a richer phenomenology in the existence of five physical states of the Higgs boson: $H, h, A, H^{\pm}$. This model can be described by 6 parameters, four for the masses of the Higgs state: $m_H, m_h, m_A, m_{H^{\pm}}$, one for the ratio of the two vacuum expectation values: $\tan\beta$ and one for the mixing angle which diagonalizes the mass matrix of h and H: α. Instead of only two for the SM: mass of the Higgs: m_h and a single vacuum expectation value: v.

With all these additional parameters taken into account, the 2HDM would be able to explain several shortcomings of the SM, thus, allowing particle physics to perform a great leap forward in the understanding of our Universe. However, despite the mathematical credibility of this model, it lacks something primordial so far: an experimental verification of this theory. But before tackling this, let's delve into the theoretical frameworks of these models.

2.1 The Standard Model

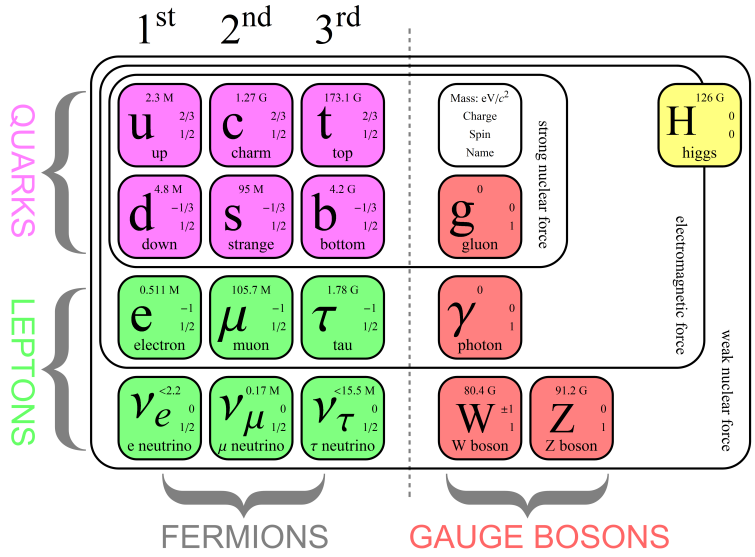


Figure 2.1: The fundamental particles of the Standard Model

Before going into details about the 2HDM, it is important to mention the Standard Model (SM). We will simply go straight to the fundamentals of the SM by mentionning two crucial principles: first the extension of the gauge invariance principle as a local concept, and second the spontaneous symmetry breaking mechanism . The introduction of local gauge invariance generates the gauge bosons as well as the interactions of these gauge bosons with fermions, and also, if the gauge group is non abelian, among the gauge bosons themselves. The combination of local gauge invariance with the spontaneous symmetry breaking mechanism leads to the Higgs mechanism which generates the masses of weak vector bosons and fermions. Since the 2HDM is an extension of the symmetry breaking sector, in this chapter, we are going to review the mechanism of electroweak symmetry breaking and focus on the Higgs particle of the SM.

2.1.1 Starting from electroweak interaction

The Lagrangian of the electroweak interaction is:

$$\mathcal{L}_{EW} = -\frac{1}{4}W_a^{\mu\nu}W^a_{\mu\nu} - \frac{1}{4}B^{\mu\nu}B_{\mu\nu} + |D_\mu\Phi|^2 - V(\Phi), \tag{2.1}$$

$$V(\Phi) = \mu^2 + |\Phi|^2 + \lambda|\Phi|^4. \tag{2.2}$$

where Φ is a complex scalar doublet with hypercharge $Y = +1$ under $SU(2)_L$ and $V(\Phi)$ is the most general, renormalisable potential. The strength tensors are written as:

$$W^a_{\mu\nu} = \partial_\mu W^a_\nu \; - \; \partial_\nu W^a_\mu \; + \; g\epsilon^{abc} W^b_\mu W^c_\nu \tag{2.3}$$

and

$$B_{\mu\nu} = \partial_\mu B_\nu \; + \; \partial_\nu B_\mu, \tag{2.4}$$

$W^a_{\mu\nu}$ and $B_{\mu\nu}$ are the gauge fields of the symmetry group $SU(2)_L$ and $U(1)_Y$, respectively.
It is also important to note that, in order to obtain local gauge invariance, the derivatives need to be changed into covariant derivatives such as:

$$D_\mu \; = \; \partial_\mu - igW^a_\mu T_a - ig'\frac{Y}{2}B_\mu \tag{2.5}$$

where $T^a = \frac{\sigma^a}{2}$ with σ being the Pauli matrices.
Two separate cases emerge here, depending on the sign of μ^2. For $\mu^2 > 0$, the state of the lowest energy achievable corresponds to the annulement of the fields corresponding to:

$$\langle\phi\rangle_0 = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{2.6}$$

thus, there is no spontaneously symmetry breaking.
For $\mu^2 < 0$, , the symmetry is spontaneously broken, in this case the fundamental state is not unique anymore, it is actually a set of degenerated state of minimum energy corresponding to a circle:

$$\langle\phi\rangle_0 = \frac{1}{\sqrt{2}}\begin{pmatrix} 0 \\ v \end{pmatrix} \quad \text{with } v = \sqrt{\frac{-\mu^2}{\lambda}} \text{ (real)} \tag{2.7}$$

with v being the vacuum expectation value (vev), which will be introduced in more details in the following section.

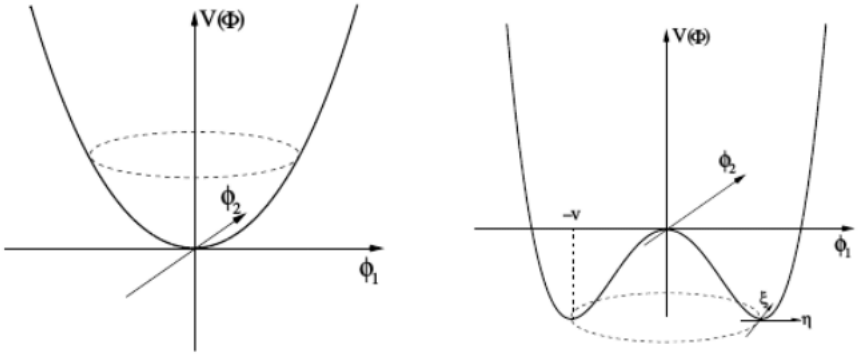


Figure 2.2: Higgs potential without and with spontaneous symmetry breaking

The doublet field ϕ can be expressed using the vev, the Higgs field and three Goldstone bosons $\phi_{1,2,3}$:

$$\Phi = \frac{1}{\sqrt{2}} \begin{pmatrix} \phi_1^+ + i\phi_2 \\ v \ + \ H \ + \ i\phi_3 \end{pmatrix} \tag{2.8}$$

There are three massive vector bosons, which we will define as follows:

$$W_\mu^\pm = \frac{1}{\sqrt{2}}(W_\mu^1 \mp iW_\mu^2), \quad Z_\mu^0 = \frac{1}{\sqrt{g^2+g'^2}}(gW_\mu^3 - g'B_\mu) \tag{2.9}$$

The fourth vector field is orthogonal to Z_μ^0, it doesn't appear in the lagrangian:

$$A_\mu = \frac{1}{\sqrt{g^2+g'^2}}(gW_\mu^3 + g'B_\mu) \tag{2.10}$$

Mass terms are terms thar are bilinear in $W^\pm, Z, A$:

$$m_W = g\frac{v}{2}, \quad m_Z = \frac{v}{2}\sqrt{g^2+g'^2}, \quad m_A = 0. \tag{2.11}$$

Hence, the two massive gauge bosons are related via:

$$\frac{m_W}{m_Z} = \frac{g}{\sqrt{g^2+g'^2}} = \cos\theta_W \tag{2.12}$$

In the case of non-abelian $SU(2)_L \times U(1)_Y$ electroweak theory, three of the gauge bosons require a mass: $W^\pm$ and Z, while the last gauge boson, the γ stays massless knowing that the electric charged must be conserved through an exact symmetry. By spontaneously breaking the symmetry $SU(2)_L \times U(1)_Y$ to $U(1)_{QED}$, three Goldstone bosons have been absorbed by the $W^\pm$ and Z bosons to form their longitudinal components and to get their masses. Since the $U(1)_Q$ symmetry is unbroken, the γ, which is associated to its generator, remains massless as it should be.

The physical bosons are, indeed, the photon A and the $W^\pm$ and Z bosons. In fact, $W^\pm$ bosons are mass eigenstates while W_μ^3 and B_μ mix to give the two physical bosons A_μ and Z_μ:

$$\begin{pmatrix} Z_\mu^0 \\ A_\mu \end{pmatrix} = \begin{pmatrix} \cos(\theta_W) & -\sin(\theta_W) \\ \sin(\theta_W) & \cos(\theta_W) \end{pmatrix} \begin{pmatrix} W_\mu^3 \\ B_\mu \end{pmatrix} \tag{2.13}$$

with $m_A = 0$ and $m_W = m_Z \cos\theta_W$, where θ_W is called the weak mixing angle and

$$\cos\ \theta_W = \frac{g}{\sqrt{g^2+g'^2}}, \quad \sin\ \theta_W = \frac{g'}{\sqrt{g^2+g'^2}} \tag{2.14}$$

With the same doublet of scalar fields ϕ, we can also generate the fermion masses. Indeed, we can add $SU(2)_L \times U(1)_Y$ gauge-invariant Yukawa interactions between the scalar fields and the fermions which are $SU(2)$ doublets or singlets.

Thus, with the same isodoublet ϕ of scalar fields, we have generated the masses of both the weak vector bosons $W^\pm, Z$ and the fermions, while preserving the gauge symmetry in the lagrangian.

2.2 The Higgs boson

The Lagrangian of the Higgs field can be written as

$$\mathcal{L} = \frac{1}{2}\partial_\mu H \partial\mu H - \lambda v^2 H^2 - \lambda v H^3 - \frac{\lambda}{4} H^4. \tag{2.15}$$

With the mass of the Higgs boson being

$$m_H^2 = 2\lambda v^2 = -2\mu^2 \approx 125.3 \pm 0.4\,(stat.) \pm 0.5\,(syst.)\,GeV/c^2 \tag{2.16}$$

where λ is the Higgs self-coupling parameter. From the previous lagrangian, several Higgs coupling can be derived using Feynman rules:

$$g_{HHH} = 3\frac{m_H^2}{v} \quad , \; g_{HHHH} = 3\frac{m_H^2}{v^2}\;[7]. \tag{2.17}$$

Altough Higgs couplings to fermions and bosons will be mentioned in a latter section, we can mention the couplings to these particles now. These couplings being:

$$g_{Hf\bar{f}} = \frac{m_f}{v} \quad , \; g_{HVV} = -2\frac{m_V^2}{v} \quad , \; g_{HHVV} = -2\frac{m_H^2}{v^2}\;, \tag{2.18}$$

with v being the vacuum expectation value, with the accepted value of

$$v = \sqrt{\frac{\mu^2}{\lambda}} \approx 246.22\,GeV[8]. \tag{2.19}$$

This value can be fixed in terms of the W mass determined by the value of the Fermi constant G_F:

$$m_W = g\frac{v}{2} = \sqrt{\frac{\sqrt{2}g^2}{8G_\mu}} \tag{2.20}$$

This happens in muon decay, which occurs through gauge interactions mediated by W boson exchange, is a particular process through which G_F is measured very accurately.

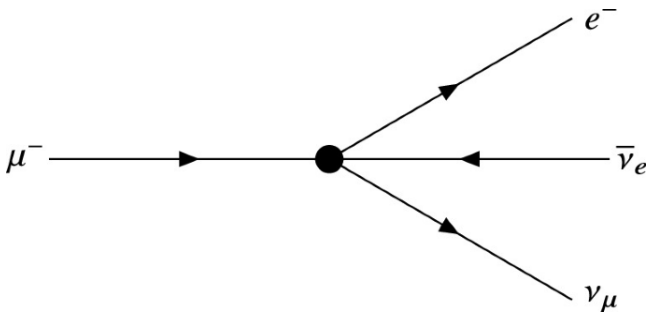


Figure 2.3: Muon decay according to Fermi's theory of weak interaction. The coupling constant is G_F

The Higgs couplings to fermions and bosons are predicted to be proportional to the corresponding particle masses, or squared-masses when it comes to boson masses. So, in Higgs production and decay processes, the dominant mechanisms involve the coupling of the Higgs boson to the heaviest particles available, in other words: $W^\pm$, Z and the third generation of quarks and leptons.

Higgs decay channels

Taking into consideration the previous section, it's no surprise to have the most dominant decays for Higgs boson to be $b\bar{b}$ and WW.

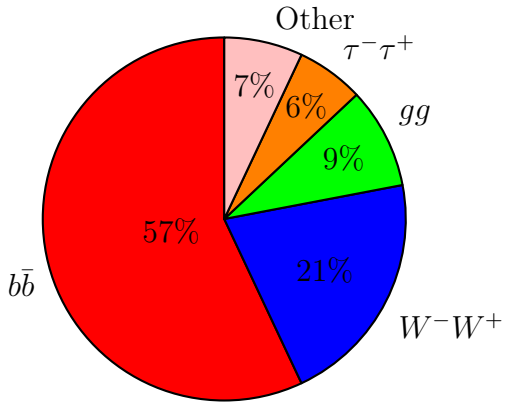


Figure 2.4: Branching ratios of the Higgs boson decays [9].

The category "Other" contains decays such as: ZZ, cc, $\gamma\gamma$, $\mu\mu$ and, theoretically, every massive particles since the Higgs boson couples to all of them.

2.3 Flaws in the SM

As mentioned earlier, the SM isn't an absolute model. There remain some points where it collides with experimental observations performed along the years. Let's go through some of them :

Massive neutrinos

In the SM, neutrinos are considered as massless and only left-handed. However, it has been observed that these particles are, in fact, massive! This has been shown through cosmological experiences [10] that have been able to determine an order of magnitude for neutrinos mass, being sub-eV, way lighter than other particles. Moreover, the concept of neutrino oscillations come from the mixing of the mass eigenstates and the flavour eigenstates. Hence, a neutrino of a specific flavour transitioning into a neutrino of another flavour during free propagation implies that, at least, one of these neutrinos must be massive.

Matter-antimatter (a)symmetry

If the distribution of matter and antimatter was perfectly balanced, the current universe would be empty, each particle of matter would had been annilihated while interacting with a particle of antimatter.
However, since our Universe is not empty, it is definitely not the case! In other words, there must have been an imbalance between matter and antimatter in the early Universe. A necessary condition to this excess of baryonic matter over antibaryonic one, is the baryon number violation. But C-symmetry violation is also needed so

that the interactions which produce more baryons than anti-baryons will not be counterbalanced by interactions which produce more anti-baryons than baryons. CP-symmetry violation is similarly required because otherwise equal numbers of left-handed baryons and right-handed anti-baryons would be produced, as well as equal numbers of left-handed anti-baryons and right-handed baryons. Finally, the interactions must be out of thermal equilibrium, since otherwise CPT symmetry would assure compensation between processes increasing and decreasing the baryon number. The violation of baryon number, of C-symmetry, of CP-symmetry and interactions out of thermal equilibium are called the Sakharov conditions [11].
However, the imbalance described by the SM is not large enough to correspond to our observations.

Dark Matter

Thanks to cosmological observations, researchers have been able to highlight an inconsistence between the expected behaviour of our Galaxy compared to its actual behaviour, especially at great radiuses. Indeed, these regions seem to possess a greater energy density than what our telescopes tend to observe.
The possible explanation introduced to explain such phenomena is the introduction of a so-called *Dark Matter* (DM), which would be a type of matter insensitive to electromagnetical interactions, electrically neutral, very long-lived (or completely stable) and massive at the same time. In other words, its only interactions would be with the Higgs boson and potentially through weak interactions (WIMPs [12]). Moreover, it is know that this dark matter is cold [13], this means that it has a non-relativistic velocity distribution.
There is no sign of such a type of matter in the SM. However, if DM interacts weakly with the SM, it could be produced at the LHC experiments escaping the detector and leaving a large missing transverse momentum as its signature. [14] [15]

Hierarchy problem

The hierarchy problem [16] tends to be the term used by physicists to describe a very important difference between the scale of mass of the electroweak bosons in one hand ($m_{W,Z,H} \approx 100\,GeV$[17]) and on the other, the Planck mass ($m_{Planck} \approx 10^{19}\,GeV$). In the SM, the mass term for the higgs boson can be written as:

$$m^2 H^\dagger H \tag{2.21}$$

it is invariant under gauge and global symmetry on H, meaning that the higgs mass parameter can be modified by radiative corrections. Thus, the Higgs mass is modified by corrective terms from every scale with which it interacts, these terms being proportional to those scales. As mentionned earlier, those scales can go all the way up to the Planck mass, and so, the mass of the Higgs according to quantum field theory expectations is much (much) higher than the experimental result ($m_H \approx 125\,GeV$ [18]). Currently, a "shaky" solution is the numerical cancellation of terms that results in the Higgs mass being reduced to its proper experimental values. However, relying on numerical cancellation is uncomfortable for many physicists.
One of the expected solution to this hierarchy problem is the use of SUper-SYmmetry (SUSY). Indeed, this theory involves the existence of plenty of other particles which

could perform so-called *miraculous cancellation* on the additionnal loops in the Higgs self-energy, solving this problem. Unfortunately, SUSY remains undiscovered as yet at the LHC and at all the other particle accelerator.

2.4 The 2HDM

The two-Higgs-Doublet Model (2HDM) is the most straightforward extension of the SM with one extra scalar doublet which contains more physical neutral and charged Higgs fields. Therefore, this model contains two complex doublets of scalar fields, ϕ_1 and ϕ_2:

$$\phi_i = \begin{pmatrix} \phi_j^+ \\ \phi_j^0 \end{pmatrix} = \begin{pmatrix} \phi_1 + i\phi_2 \\ \phi_3 + i\phi_4 \end{pmatrix} \tag{2.22}$$

with j = 1, 2. Thus, there are now eight degrees of freedom that will be used to give masses to the gauge bosons, instead of only four in the SM. In some cases, after symmetry breaking, three Goldstone bosons provide the longitudinal modes of the bosons $W^\pm$ and Z, that become massive. And there will remain five physical Higgs bosons: three neutral ones H, h, A and two charged ones $H^\pm$. Where H and h are scalar bosons such as $m_H > m_h$ and A is a pseudoscalar boson.
The most general scalar potential contains 14 parameters with CP-violating, CP-conserving and charge-violating minima. However, we can use several simplifying assumptions. First, we assume CP-conservation in Higgs sector, in order to draw a separation between scalars and pseudoscalars. Then, CP is not spontaneously broken. Lastly, discrete symmetries eliminate all quartic terms odd in either of the doublets from the potential, including a term wich softly breaks these symmetries. Under those assumptions, we can now formulate the most general scalar potential for two doublets Φ_1 and Φ_2 with $Y = +1$ as:

$$\begin{aligned} V = {} & m_{11}^2 \Phi_1^\dagger \Phi_1 + m_{22}^2 \Phi_2^\dagger \Phi_2 - m_{12}^2 (\Phi_1^\dagger \Phi_2 + \Phi_2^\dagger \Phi_1) + \frac{\lambda_1}{2} (\Phi_1^\dagger \Phi_1)^2 + \frac{\lambda_2}{2} (\Phi_2^\dagger \Phi_1)^2 \\ & + \lambda_3 \Phi_1^\dagger \Phi_1 \Phi_2^\dagger \Phi_2 + \lambda_4 \Phi_1^\dagger \Phi_2 \Phi_2^\dagger \Phi_1 + \frac{\lambda_5}{2} \left[(\Phi_1^\dagger \Phi_2)^2 + (\Phi_2^\dagger \Phi_1)^2 \right], \end{aligned} \tag{2.23}$$

where all the parameters are real.
As mentionned earlier, with two complex scalar SU(2) doublets there are eight fields:

$$\Phi_a = \begin{pmatrix} \phi_a^+ \\ (v_a + \rho_a + i\eta_a)/\sqrt{2} \end{pmatrix}, \quad a = 1, 2. \tag{2.24}$$

With $\phi^\pm$ corresponding to charged scalar bosons, η to pseudoscalars and ρ to scalars. We can use these parameters to rewrite the scalars and pseudoscalar bosons as:

$$A = \eta_1 \sin\beta - \eta_2 \cos\beta \tag{2.25}$$

$$h = \rho_1 \sin\alpha - \rho_2 \cos\alpha \quad and \quad H = -\rho_1 \cos\alpha + \rho_2 \sin\alpha \tag{2.26}$$

Notice that the standard SM higgs boson would be:

$$\begin{aligned} H^{SM} &= \rho_1 \cos\beta + \rho_2 \sin\beta \\ &= h \sin(\alpha - \beta) - H \cos(\alpha - \beta) \end{aligned} \tag{2.27}$$

The different types of 2HDM

Two-Higgs-doublet models can introduce flavor-changing neutral currents (FCNC) which have not been observed so far and are considered as heavily suprressed by the GIM mechanism [19]. To avoid the prediction of such currents, we require that each group of fermions (up-type quarks: $Q = \frac{2}{3}$: u,c,t; down-type quarks: $Q = \frac{-1}{3}$: d,s,b and charged leptons) couples exactly to one of the two doublets ϕ as formulated here:

$$\mathcal{L}_{Yukawa}^{2HDM} = -Y_d \bar{Q}_L \Phi_d d_R - Y_d \bar{Q}_L \tilde{\Phi}_u u_R - Y_l \bar{L}_L \Phi_l l_R + h.c., \tag{2.28}$$

with $\Phi_{d,u,l}$ corresponding to either ϕ_1 or ϕ_2.
By convention, up-type quarks always couple to ϕ_2.

Depending on which type of fermions couples to which doublet ϕ, one can divide two-Higgs-doublet models into the following classes [20]:

- Type-I: all quarks and charged leptons couple to the same doublet: ϕ_2
- Type-II: only up-type quarks couple to ϕ_2, while down-type quarks and charged lepton couple to ϕ_1
- Type X: all quarks couple to ϕ_2, while charged leptons couple to ϕ_1
- Type Y: up-type quarks and charged leptons couple to ϕ_2, while down-type quarks couple to ϕ_1

The Type-II is the most studied case, since the couplings of the Minimal Super Symmetric Model (MSSM) are a subset of the couplings of Type-II 2HDM.
Another type, called Type-III [21], exists. It relies on the inclusion of tree-level FCNC. Its goal is to be used for large energy scale (multi-TeV and higher), since, in that case, the previous solution used to exclude FCNCs seems unnatural. [22]

$\mathbb{Z}_2$ symmetry

The most commonly used symmetry ensuring the absence of FCNCs is the $\mathbb{Z}_2$ symmetry. In the case of 2HDM, with two doublet Φ_1, Φ_2, a $\mathbb{Z}_2$ symmetry transforms the fields as:

$$\mathbb{Z}_2 \ : \ \Phi_1 \to \Phi_1 \,, \quad \Phi_2 \to -\Phi_2 \,. \tag{2.29}$$

Couplings to fermions

To determine the Yukawa couplings, we can rewrite the Yukawa interaction term as:

$$\begin{aligned} \mathcal{L}_{\text{Yukawa}}^{\text{2HDM}} = &- \sum_{f=u,d,l} \Big(\frac{m_f}{v} \xi_h^f \bar{f} f h + \frac{m_f}{v} \xi_H^f \bar{f} f H - i \frac{m_f}{v} \xi_A^f \Big) \\ &- \left[\frac{\sqrt{2} V_{ud}}{v} \bar{u} (m_u \xi_A^u P_L + m_d \xi_A^d P_R) d H^+ + \frac{\sqrt{2} m_l}{v} \xi_A^l \nu_l l_R H^+ + \text{h.c.} \right] \end{aligned} \tag{2.30}$$

with P_L and P_R are the left and right projection operators, and the factors ξ_H^f, ξ_h^f, ξ_A^f are parameters defined, in the case of Type-II 2HDM, as:

- $\xi_H^u = \frac{sin\alpha}{sin\beta}$, $\xi_H^d = \frac{cos\alpha}{sin\beta}$, $\xi_H^l = \frac{cos\alpha}{sin\beta}$,

- $\xi_h^u = \frac{cos\alpha}{sin\beta}$, $\xi_h^d = \frac{-sin\alpha}{sin\beta}$, $\xi_h^l = \frac{-sin\alpha}{sin\beta}$,
- $\xi_A^u = cot\beta$, $\xi_A^d = tan\beta$, $\xi_A^l = tan\beta$.

2HDM and Super-symmetry

Supersymmetry (SUSY) is a very elegant theory in particle physics which predicts the existence of so-called *super-partners* or *spartners* to each of the pre-existing particles in the SM. Each of these spartners would differ by a half-integer value from the spin of the SM particles, meaning that it's a symmetry transforming fermions to bosons and bosons to fermions. Thus, the SM needs to be extended by adding a new elementary particle for every known particle. Then, two point of view exist when it comes the SUSY, one involve a more simple formulation of the theory, with perfectly unbroken symmetry, meaning that particles and their correspondant spartners would have the same mass. On the other hand, there is also formulations about more complex symmetry, involving spontaneously broken symmetry allowing spartners to differ in mass.

SUSY would provide a very convenient solution to the hierarchy problem and would also be able to provide a DM candidate. Moreover, it unifies the three interactions at the grand unification theory scale.
The simplest supersymmetric extension to the SM is the Minimal Supersymmetric Standard Model (MSSM) [24]. Knowing that this model requires a second Higgs doublet, this specific model of SUSY is directly included in the 2HDM.
However, no experimental proofs of SUSY as been found in high energy experiments.

2.5 Di-Higgs physics

Di-Higgs production

The production of Higgs boson pairs inside particle colliders is very rare, a factor of thousand less likely than for the production of a single Higgs boson. The dominant process to produce a Higgs pair is the gluon-gluon fusion (ggF), where a Higgs pair can emerge as several types of final states, with probability given by branching decays of each boson shown at Fig. (2.4). At leading order (LO), in other words, all vertices in the Feynman diagrams emerge from the lowest-order-Lagrangian, two separate cases appear, represented in the Feynman diagrams at Fig. (2.5). It's important to highlight that the box-like diagram is sensitive only to the Higgs-top quark coupling: κ_t, while the triangle one is sensitive κ_t aswell as to the HHH coupling or Higgs self-coupling: κ_λ. There is a destructive interference between these two diagrams resulting in an overall small cross section of di-Higgs production. The total cross section for this process in the SM is about 33.47 fb. [25]

Despite being the most dominant process, gluon-gluon fusion is not the only production channel. Indeed, the vector-boson fusion channel is able to produce pairs of Higgs bosons, despite a cross section being at least one order of magnitude smaller than ggF. This channel is still interesting to probe since it gives us access to several different couplings, such as the Higgs boson self-coupling, as in ggF, but also to new

couplings like the HHVV coupling: κ_{2V} or the HHV one: κ_V with V being a W or Z boson.

It is also interesting to mention a third di-Higgs boson production channel being via ggF with anomalous Higgs boson coupling. However, this channel won't be discussed in this work.

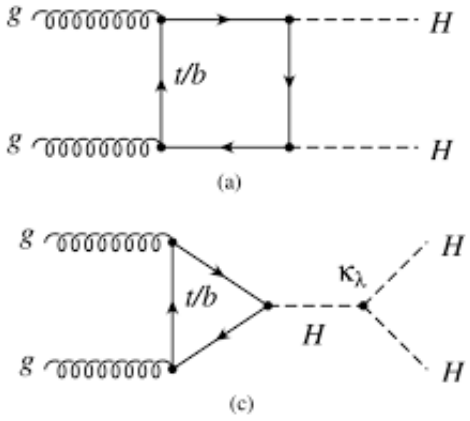


Figure 2.5: Production channels for di-Higgs.

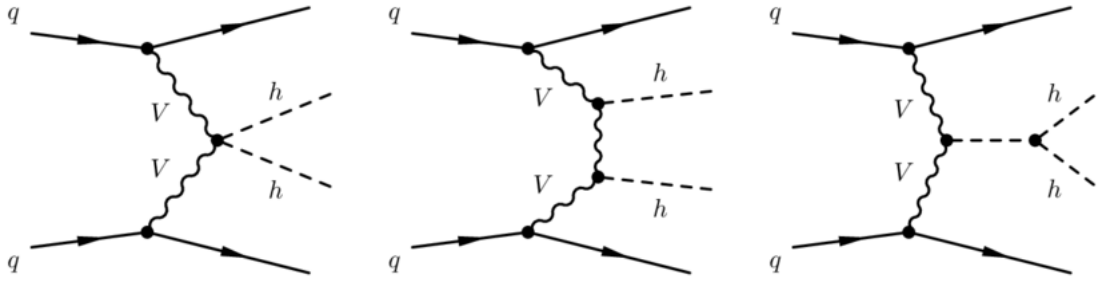


Figure 2.6: Vector Boson Fusion (VBF) production channels for di-Higgs.

2.6 Purpose of this project

As stated earlier, the ability of discriminating signal from background is of crucial importance. This is the framework within which this project is being carried out. The signal we are interested in is the channel $HH \to b\bar{b}\,W^+W^-$, being the second largest decay branching fraction. We consider both W bosons decaying to electrons or muons. The main backgrounds to this channel are the DY process, $t\bar{t}$ production aswell as W + jets events. We are going to use a conditional generative adversarial network trained on a sample of Drell-Yan events to generate background samples of the same process. Allowing future researches to have a better grasp of the charateristics of such a background, in order to filter it out efficiently.

Chapter 3

Di-higgs physics at LHC

Now that we have introduced the concepts used in this work, we can mention the purpose in which this work is carried out: the investigation of di-Higgs production as a potential evidence of new physics. This can be done by observing di-Higgs production (and decay) and monitoring both the coupling values, as κ_t and κ_λ, and the overall cross-section of the process.

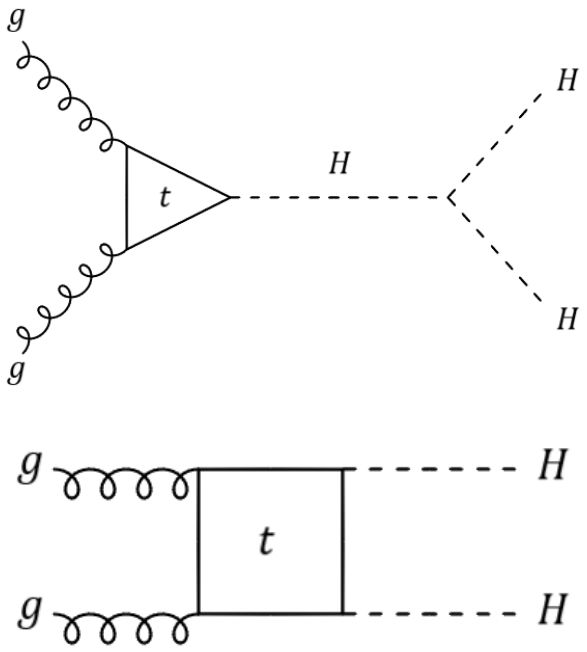


Figure 3.1: Di-Higgs production processes.

Observing this process alone will not directly prove the Two-Higgs-Doublet Model (2HDM) over the Standard Model (SM), as it is also allowed within the SM framework. The key difference between the two models lies in the frequency of this decay. In the SM, detecting this process would require twenty times more data than what is currently available [26] [27]. However, the 2HDM may enhance the di-Higgs production rate for two reasons. First, the main reason would be the existence of new particles, such as heavier Higgs bosons, decaying to HH. This would lead to an increased cross-section of the process, in comparison to the SM. Second, the strength

of the Higgs boson self-interaction might be altered, thus leading to an increased frequency of the di-Higgs decay. If the observation of this specific decay channel of the Higgs boson was observed early enough, it would represent an experimental validation of the 2HDM.

Detecting such a process is a significant challenge. Despite traveling at nearly the speed of light, the Higgs boson's lifetime is extremely short ($\tau_H \approx 1.6 \times 10^{-22}$ second), preventing it from traveling far enough to reach any layer of current particle detectors. Consequently, we must rely on detecting the Higgs boson's decay products, which can eventually reach the detector. As shown in Figure (2.4), there are several decay channels, resulting in a variety of final states for the di-Higgs decay process.

There is no single best choice among these decay channels, as each combination of Higgs boson decays has its own advantages and disadvantages. Two important factors to consider are the branching ratio and the ability to eliminate background noise. The ideal channel would have a high branching ratio and a background that can be effectively filtered out using current techniques or future methods that are within reach. However, channels with high branching ratios often have complex backgrounds, while those with manageable backgrounds tend to have lower branching ratios. Currently, several final state signatures are under investigation, including $\bar{b}b\bar{b}b$, $\bar{b}b\bar{\tau}\tau$, and $\bar{b}b\gamma\gamma$. In this work, we will focus on the $b\bar{b}W^+W^-$ channel.

3.1 Channel of interest: $b\bar{b}W^+W^-$

Since the Higgs boson has several known decays, there is a whole array of possible combinations for the two bosons produced by a di-Higgs decay process. The decay with the largest branching ratio is $HH \to b\bar{b}b\bar{b}$, followed by $HH \to b\bar{b}W^+W^-$, $HH \to b\bar{b}\tau\bar{\tau}$, $HH \to b\bar{b}\gamma\gamma$, ... Despite having the largest branching ratio, $HH \to b\bar{b}b\bar{b}$ is composed of 4 simultaneous b-jets, which represents a complex challenge to identify due to important background. On the other hand, $HH \to b\bar{b}\tau\bar{\tau}$ and $HH \to b\bar{b}\gamma\gamma$ are expected to have much cleaner signature, at the cost of lower branching ratios. Particle physicists believe $HH \to b\bar{b}W^+W^-$ to be a good trade-off between large branching ratio and recognizable final state signature.
We consider at least one W boson decaying to electrons or muons. The main backgrounds contributing to this final state is the top quark pair production $t\bar{t} + jets$, followed by the Drell–Yan process or $W + jets$ processes depending on whether the second W boson decays leptonically or hadronically, respectively.

In the $HH \to b\bar{b}W^+W^-$ case, several signals can be expected. Firstly, depending on the decay channel of the W boson. Indeed, two cases can be considered: the half-leptonic and the fully-leptonic case. In this work, we will main discuss the latter.
The W has several channels of decay:

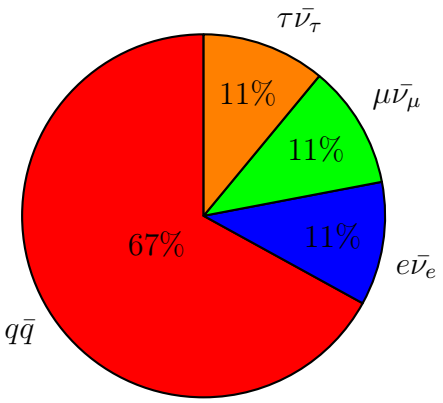


Figure 3.2: Branching ratio of W boson decay.

The half-leptonic case is when only one of the two W-bosons decay into one of the 3 least common channels (leptonic channels), while the other decays into two jets (hadronic channel). The fully-leptonic case is when both W bosons decay through leptonic channels. Thus, in the fully leptonic case, we will have a $bbll\nu\nu$ system, knowing that, in CMS, neutrinos cannot be directly detected. We will then have to rely on the *missing tranverse energy* of an event in order to indirectly detect them. Secondly, in the case of $HH \rightarrow b\bar{b}W^+W^-$, only e's and μ's are taken into account. However, the τ leptons can decay via weak interactions, thus producing an e or a μ alongside two ν's. We can say that the case of a W decaying to a τ is also taken into account.
Thirdly, Higgs bosons decaying to $\tau\bar{\tau}$ themselves decaying to electrons and muons could also be considered as signal, as mentionned in literature. This would lead to a $bbll\nu\nu\nu\nu$ system, since ν's are not identified by any instrument of *CMS* this final state signature can be considered similar to $bbll\nu\nu$.

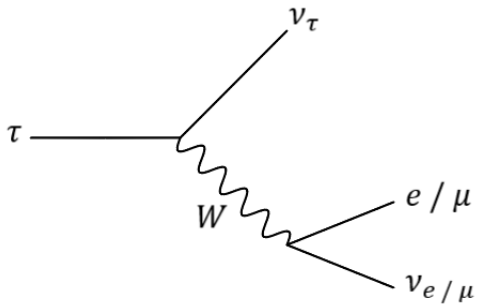
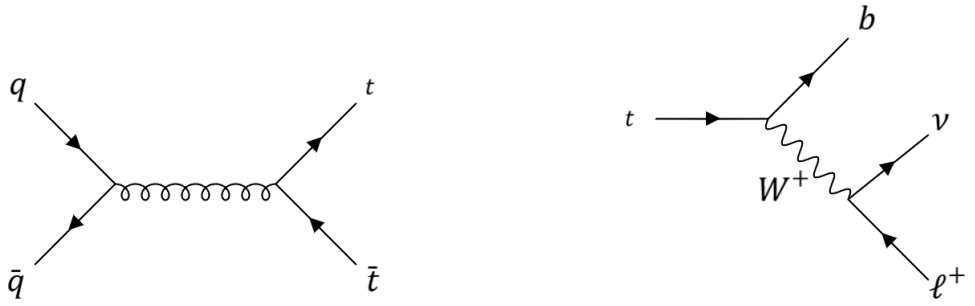


Figure 3.3: Decay of the tau lepton.

3.2 Backgrounds

The main backgrounds to the $b\bar{b}W^+W^-$ state are the production of top quarks pair $t\bar{t} + jets$ Fig.(3.4a), Drell-Yan events Fig.(3.5) and $W + jets$ processes.
The most important one is the $t\bar{t}$ production, it is a common process in hadron-hadron colliders as the LHC. The production of the pair of top quarks does not represent a background in itself, the parasitic signal will come from the decay of

each top quarks[1] shown in Fig.(3.4b). Indeed, each quark will decay into a b-quark and a W boson, thus faking the $bbWW$ signature.
Despite being important, physicists can satisfyingly cope with this process with current background filtering techniques. Indeed, the main difficulty of this process only relies in the $t\bar{t}$ production, since it involves quantum chromo-dynamics (QCD) concepts as *parton distributions functions* (PDFs) that are relatively unpredictable. However, the interesting part of the process, the decay, only depends on electroweak physics, which is totally predictable. Hence, this background can already be filtered out in many high energy physics experiment with current techniques .



(a) Production of $t\bar{t}$.

(b) Decay of the top quark.

Figure 3.4: Background from $t\bar{t}$ events

However, identifying Drell-Yan events is not as straightforward. Indeed, the presence of the quark-antiquark pair being directly involved in the process leads to the implementation of PDFs in the computation. As already stated, these functions are hardly predictable and imply large uncertainties. Moreover, light jets, i.e. non b-jets, arise. In order words, physicists must deal with quarks going through hadronization, another QCD concept, in order to understand a DY event. Then, at next to leading order (NLO), gluon radiations may arise from the quark-antiquark pair as shown in Fig.(3.7). These gluons will eventually split, producing yet another quark-antiquark pair, each one potentially going through hadronization and producing jets.

The source of the uncertainties brought by these QCD concepts are two-fold. Most importantly, this section of high energy physics is still misunderstood so far. Thus, the current impossibility to design accurate techniques computing QCD events. On the other hand, the energy scale at which these processes happen has an influence on the outcome. Indeed, at high energies, where the strong coupling constant α_S is small, perturbative theory can be applied, thus providing relatively accurate results. In the other case, where α_S is large, the expansion series of the perturbative theory will diverge, and thus, cannot be applied anymore. Some other computation techniques exist, such as the QCD lattice [28]. However, the computational requirements are extreme and the result is still victim of important uncertainties.

[1]Conversely to other quarks, the top does not hadronize into a jet. The lifetime of this top quark is sufficiently small to be shorter than the QCD time scale. In other words, the top quark will decay before hadronizing.

Unfortunately, in the DY case, most of the QCD effects belong to the non-perturbative regime.

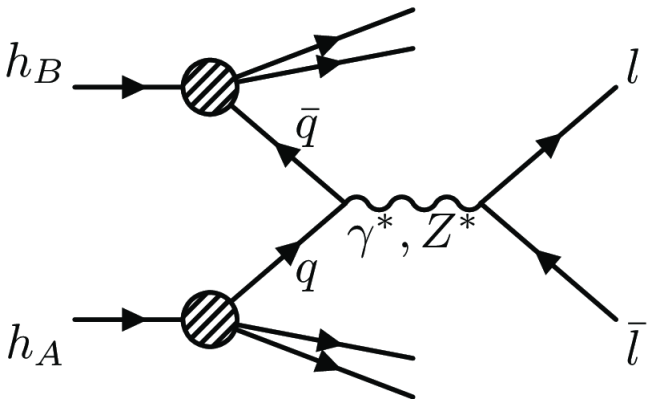


Figure 3.5: Drell-Yan process.

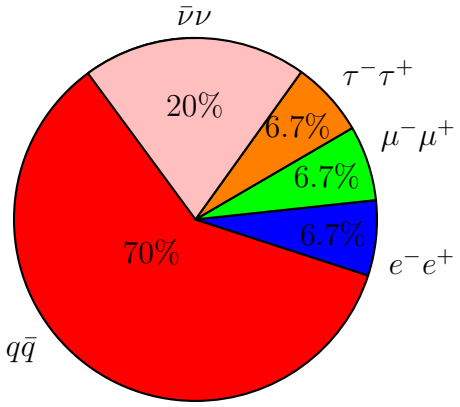


Figure 3.6: Branching ratio of Z boson decay.

Different types of Drell-Yan events

The Fig.(3.5), as already stated, represents the DY process, but an important piece of information was purposefully left out until now, this diagram is at the leading order. However, we could go to NLO, leading to new physical phenomenas. In this case, we are interested in radiative gluons.

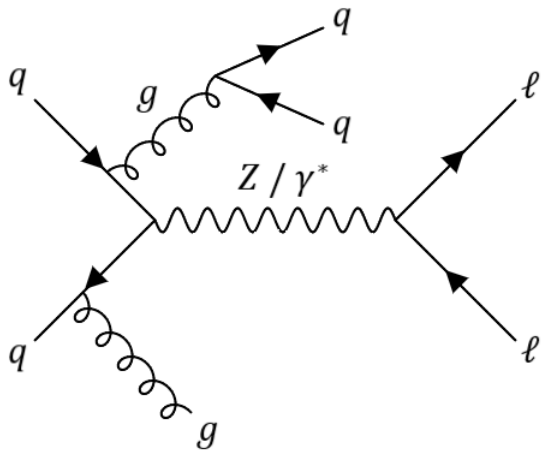


Figure 3.7: NLO Drell-Yan process with radiative gluons.

These radiative gluons can undergo splitting and thus producing a quark-antiquark pair. These quarks have a probability of being b-quarks, thus the event will constitute a background for the $b\bar{b}W^+W^-$ channel.
In the generated training sample, the maximum number of b-jets in a Drell-Yan event is one. The proportion of events with such a jet is approximately 0.5%. This represents one of the main problem of simulating these events. Indeed, the vast majority of the sample will be directly labelled as useless since it mainly contains DY events without the emission of any b-jets, resulting in an important waste of computational power.

Now that we have shown how different processes could produce the same particles as in the $b\bar{b}W^+W^-$ final state, we need to introduce how can the background be discriminated from the signal. In other words, how to recognize a lepton, a b-quark emitted by di-Higgs from a lepton emitted by the decay of a Z and a b-quark produced by a top quark decay?
Well, despite being the same type of particles, they differ at a very important point: kinematic observables.

3.3 CMS and generated variables

Before introducing the different variables selected to be generated by the network, let's briefly discuss the particle detector linked to these research: the *Compact Muon Solenoid* (CMS).

CMS

CMS is one of the particle colliders/detectors at the LHC. As a multipurpose detector, alongside ATLAS, it is equipped with a variety of calorimeters, trackers, taggers, and more. In 2012, it played a crucial role in the discovery of the Higgs boson and continues to be involved in numerous research projects in high-energy physics, such as the search for evidence of the 2HDM.

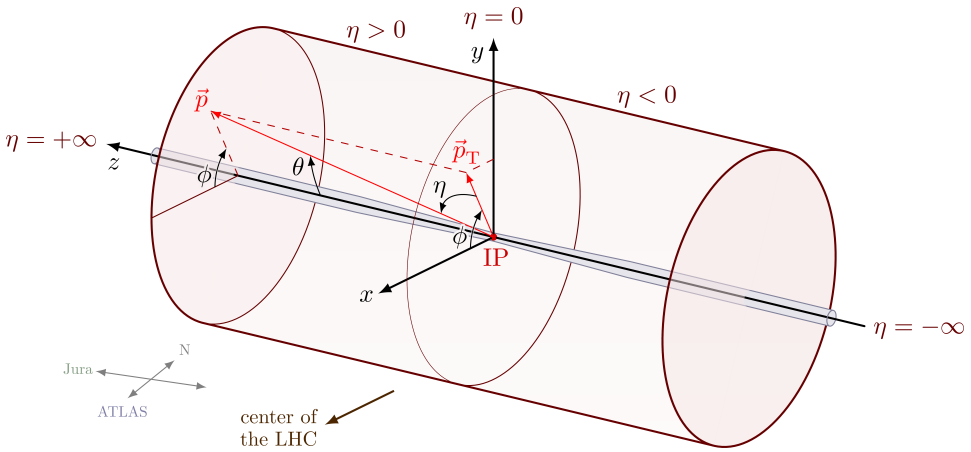
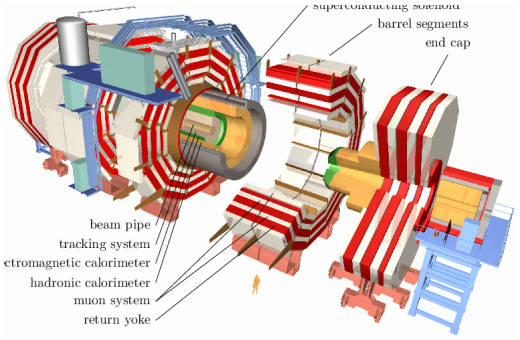
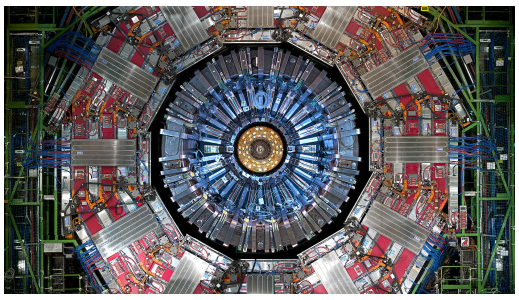


Figure 3.8: Coordinate system used in CMS



(a) Schematic overview of CMS.



(b) Frontview of CMS.

As discussed in the following section, the transverse plane to the particle beam is the main reference used for most studies using CMS. Despite quantum mechanics significantly altering the rules of classical mechanics, several laws remain unchanged. Among them is the conservation of momentum:

$$\sum \vec{p} = 0. \tag{3.1}$$

The transverse plane is particularly convenient since, before the hadron collision, it is completely empty, unlike the longitudinal plane parallel to the beam. Therefore, the transverse plane is a more suitable candidate for observing events inside CMS.

Invariant mass of the di-lepton system

The invariant mass represents the mass of a particle in its rest frame. It's the same in all frames of reference, hence the name *invariant*. It is determined as follows (in natural units):

$$m^2 \;=\; E^2 - ||\mathbf{p}||^2. \tag{3.2}$$

Since the invariant mass is determined from quantities which are conserved during a decay, the invariant mass calculated using the energy and momentum of the decay products of a single particle is equal to the mass of the particle that decayed. The mass of a system of particles can be calculated from the general formula:

$$m^2_{syst} \;=\; \left(\sum E\right)^2 + \left|\left|\sum \mathbf{p}\right|\right|^2, \tag{3.3}$$

with m_{syst} the invariant mass of the system of particles, equal to the mass of the decay particle.
However, in the case of massless or highly relativistic particles, as in our case, the invariant mass is defined by:

$$m^2 = 2p_{T1}p_{T2}(\cosh(\eta_1 - \eta_2) - \cos(\phi_1 - \phi_2)). \tag{3.4}$$

Applying this observable to the research carried out in this report, two leptons originating from a Z boson of mass $\sim 92\ GeV/c^2$ will differ from leptons formed by the decay of a W boson of mass $\sim 80\ GeV/c^2$.
Same goes for a b-quark from a top quark decay of mass $\sim 173\ GeV/c^2$, a b-quark from the sea quark and a bottom from a Higgs boson decay of mass $\sim 125\ GeV/c^2$.

Leading lepton transverse momentum

The transverse momentum $\vec{p}_T$ is the momentum of an object in the transverse plane to the beam. The initial longitudinal momentum in a hadron-hadron collision is unknown, because the partons that make up a hadron share the momentum. Indeed, in the case of protons, each will have three valence quarks and an indetermined number of sea quarks and gluons. However, it is known that the initial transverse momentum is zero, and, if every particles were perfectly detected, the final transverse would also be zero.

Conversely to the other variables, this one has not been picked for its discriminating power. In fact, the momentum of the leading muon was the first variable generated with the 1D GAN in order to assess to performance of the network. Indeed, the *MuonPt* is a very basic, but yet very important, kinematic observable. Thus, if a NN is not able to reproduce it, it might not be able to generate anything else.
Moreover, it has been chosen for its correlation with other observables. This way, we can check the capacity of the cGAN to reproduce distributions and correlations.

It is worth mentionning that this observable could be used in order to discriminate signal from background. Indeed, in the Drell-Yan process the two leptons are emitted by a Z boson. On the other hand, the leptons in the di-Higgs come from the decay of W bosons themselves originated from a Higgs boson decay. The on-shell W will provide a lepton of approximately same momentum than DY, while the off-shell W will emit a lepton with a much lower p_T. However, this can not be applied to this work since only the leading lepton momentum is used.

Missing transverse energy

The transverse energy is an observable tightly bounded to the transverse momentum. Indeed, its mathematical definition is the following:

$$E_T = \sqrt{m^2 + p_T^2} \tag{3.5}$$

As stated in the previous section, both the invariant mass and the transverse momentum are conserved quantity. Meaning that, we would expect the missing transverse

energy to be also conserved. However, it is not detected as such. In fact, some particles escape the detector without being noticed. It can be due to the geometrical constraints of the detector, for instance CMS has limited detection depending on the pseudo-rapidity of particles: $|\eta| < 2.5$. Or it can be caused by the nature of the particle. Indeed, neutrinos have tiny cross-sections, so they will mainly go unnoticed to detectors not specifically designed for neutrino detection.
Though, in particle colliders, the transverse energy is not the most interesting kinematic observable. In fact, physicists will rather look at the missing transverse energy (MET), in order to indirectly detect "invisible" particles in an event. The MET is defined as follows:

$$E_T^{miss} = -\sum_i |\mathbf{p}_T| = -\sum_i p_T(i). \tag{3.6}$$

The missing transverse energy of each event is a powerful information. Indeed, a large quantity of undetected momentum would almost assurely mean the existence of neutrinos within the event. It is the only way to involve these ghostly particles in our analysis. For a Drell-Yan event, there are no neutrinos expected. However, for the di-Higgs decay, two neutrinos should be produced by the decay of the W bosons. In other words, an event with little to no missing transverse momentum has a high probability to be background.

Jets activity

Beside kinematics, the jet activity is also a crucial criterion. Indeed for the signal region, one of the two Higgs bosons is expected to decay into $b\bar{b}$. These bottom quarks subsequently hadronize, forming collimated sprays of particles known as b-jets. Additionally, other partons produced in association with the Higgs bosons can also hadronize, resulting in additional jets in the final state. Therefore, in the di-Higgs process, there is typically significant jet activity, with multiple jets observed in the event, particularly b-jets from the decay of Higgs bosons.
For the Drell-Yan, an intermediate boson (γ or Z) decays into a lepton-antilepton pair. Unlike in the di-Higgs process, where additional jets can arise from the decay of Higgs bosons and associated parton radiation, the Drell-Yan process typically does not involve significant jet activity. This is because the primary interaction involves only two partons (quark and antiquark), and the final-state particles are predominantly the produced leptons, without the hadronization of additional partons.
The number of b-tagged jets is used as the additional label provided to the generator and, at the same time, is used as the blinding variable in training sample.

3.3.1 Quick insight on simulation

For simulating Drell-Yan events, several approaches exist. One of them is shown at Fig.(3.10). The main idea is to draw two separate regions, region A containing non-perturbative QCD physics, i.e. hardly predictable processes, and region B with a pair of quark-antiquark and the electroweak decay of the mediator boson, i.e. predictable processes. This approach focuses on the region B since the physics contained is satisfyingly replicable with our current techniques.

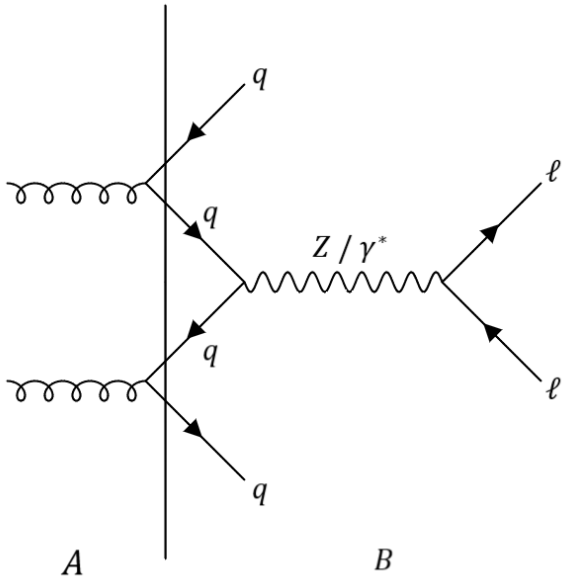


Figure 3.10: Drell-Yan process produced via double gluon splitting. In simulation, region A can be overlooked.

Chapter 4

Numerical methods

As stated many times in this work, the background modeling is a crucial component to the success of an experiment in high energy physics. To achieve this, several strategies are available: data-driven methods, Monte Carlo simulations, parametric and non-parametric methods among others. For basic cases, data-driven background estimation methods are sufficient to estimate the expected number of background events. Unfortunately, these basic cases only represents a minority. In most cases, background modeling relies on direct simulation based on Monte Carlo (MC) event generators or parametric method.
However, for even more intricate situations, these two techniques fall short. The problem is, despite a sufficient background modelling concerning some of its components, it is not enough to accurately predict the characteristics of the total background. That's the problem tackled in this thesis.
Indeed, in this work we address the background modelling of the Drell-Yan process, for di-Higgs physics. The other components of the total background, i.e $t\bar{t}$ production and $W + jets$ events, can be fairly easily modelled. However, this isn't the case for DY due to its varying final state signature. To achieve our goal, we use a *non-parametric data-driven background modelling* method, via a cGAN. Let's briefly breakdown what this expression actually means.

4.1 Parametric and non-parametric methods

Parametric

Parametric methods are statistical techniques relying on specific assumptions about the underlying distribution of the population being studied. These methods typically assume that the data follows a known probability distribution, such as the normal distribution, and estimate the parameters of this distribution using the available data. Parametric methods are those methods for which we a priori know that the population follows a Gaussian distribution [1], or if not then we can easily approximate it using such a distribution. In addition to the Gaussian assumptions, these techniques also assume the independence between observations aswell as a homogeneous variance over the set of events. For normal distributions, the parameters are:

[1] In most cases, a Gaussian distribution is used. However, other distributions can also be taken into account depending on the application. In this work, the *Breit-Wigner* can be an interesting choice.

the mean (μ) and the standard deviation (σ).

The efficency of this category of methods heavily relies on whether or not the assumptions are met. If that's the case, these techniques are very powerful (i.e. able to detect a real effect when it exists), even working on a reduce set of events. However, it makes those a very rigid option, which may not capture complex relationships between variables.

Non-parametric

Conversely to the previous category of methods, non-parametric ones do not rely on specific assumptions of parameters. In fact, they don't depend at all on the population studied. Hence, there are no parameters or distributions needed. However, some assumptions about the data are still required as the independence of observations or the homogeneity of measurements. Conversely to parametric methods, non-parametric ones are widely applicable due to their independence to the studied population, their easy implementation and their robustness to outliers. However, when the assumptions of parametric methods are met, these ones remain more powerful and require smaller sample to achieve the same level of power.

Some examples of non-parametric methods used for LHC data analysis are Kernel Density Estimation (KDE) used to estimate the probability density function of a random variable based on a sample of data. It can be used to visualize the distribution of a specific observable without the assumption of a parametric form for the distribution. Moreover, Random Forest algorithms (an advanced version of decision trees) are also non-parametric approaches. These machine learning algorithms are used to perform classification and regression tasks based on the characteristics of the events/particles probed, without, once again, assuming a specific parametric form for the underlying data distribution.

In this thesis, complex relationships are expected among the variables simulated by our GAN. Hence, we choose to work with non-parametric methods due to their flexibility and low-computational cost.

4.2 Data-driven methods

Data-driven methods are a class of methods that primarily rely on current data collected during the system's/process' lifetime in order to establish relationships between input, internal and output variables. Their aim is to efficiently process and analyze large datasets. Hence their usefulness for the generalization of our cGAN to samples of considerable size.
The term data-driven modeling refers to the use of current data merged with advanced computational techniques, as machine learning, to create models revealing underlying trends and patterns between variables of a same dataset. Moreover, data-driven models can be built with or without detailed knowledge of the underlying processes governing the system behavior, which makes them particularly useful when such knowledge is not in our possession. Hence, data-driven background estimates

are a must in situations where you cannot get a reliable estimate from simulation.

Such methods are extensively used in the scope of data analysis at the LHC. For jet mass reconstruction, some very important variables are determined thanks to data-driven approach. Indeed, both ATLAS and CMS developped tagging algorithms for jets, that includes an array of validation and calibration techniques processed in a data-driven manner.

4.2.1 ABCD method

The ABCD method [34] is a common use to get data-driven background estimation, as seen in [35] one of the main paper this work is based on. The idea behind this method is illustrated in Fig.(4.1). The phase space is divided into four different regions, each defined by variables uncorrelated to each other. The region D is the *signal region*, in other words, the phase space region defined by the triggers and selections used for the signal we are interested in. While the other regions (A, B, C) are the *control regions*, these are obtained by modifying some of the cuts used for the signal selection, in order to obtain similar regions to the signal one, with the important difference that control regions do not contain any signal [2]. Control regions are usually defined over a specific background process, with enough events to insure sufficient statistics. The shape of the background process can then be estimated as a function of one or several variables. These regions can also be referred as *sidebands* in cases where the signal appears as a resonance peak. Signal region is then a specific window and the control regions are on both sides on this windows, hence the name sidebands.

Although the control regions are defined to be as similar as possible to the signal region, some differences in the selection efficiency for the background process may happen between these two regions. Thus, the control region is corrected by deriving additional events weights called *transfer factors*. To determine these factors, we use the two remainings regions: A and B. We assume that the ratio between A and B is defined by the same cuts than the ratio between C and D. Transfer factors are then determined by the change of background from A to B, and they are then applied to C.

4.3 Shortcomings of Monte Carlo simulations

The Monte Carlo (MC) simulations are a broad class of computational algorithms.[3] These simulations are very widespread in many fields, such as high energy physics. Despite their numerous advantages, MC techniques hold several heavy limitations, common to all their applications. These include: high computational cost, especially with complex model; heavy reliance on quality input data and on the different assumptions made; aswell as a difficult result interpretation, especially for cases without a strong statistical background. Moreover, when it comes to high energy

[2] In an ideal case, there is no signal at all. However, some signal might be present in these control regions, but the ratio signal-over-background ratio will remain tiny.

[3] The intrinsic operation of MC simulations won't be addressed here, since it is of no use in this work. For more information, please see [36].

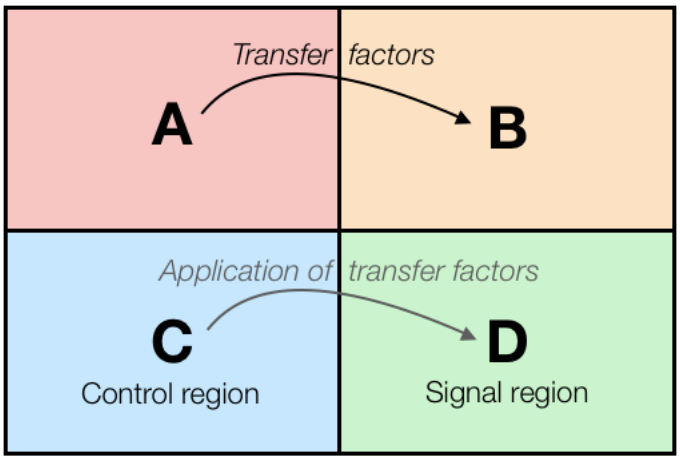


Figure 4.1: Representation of the ABCD method

physics, there also problems coming from imperfect modelling of the detectors and limitations due to fixed-order calculations, meaning that only a finite number of terms in the perturbative expansion are considered.
For all those reasons, we cannot entirely rely on MC simulations. Hence the need of the new approach discussed in this work.

4.4 b-tagging algorithms

The identification (tagging) of the jets coming from the hadronization of heavy flavor quarks, such as bottom quarks, is made possible by the use of data-driven approaches and by distinctive properties of the heavy hadrons. For instance, B-hadrons have a relatively large lifetime ($\mathcal{O}(1.5ps)$) leading them to travel measurable flight length path of a few millimeters before their decay into lighter hadrons. Thus, creating a secondary vertex clearly distinct from the main one.

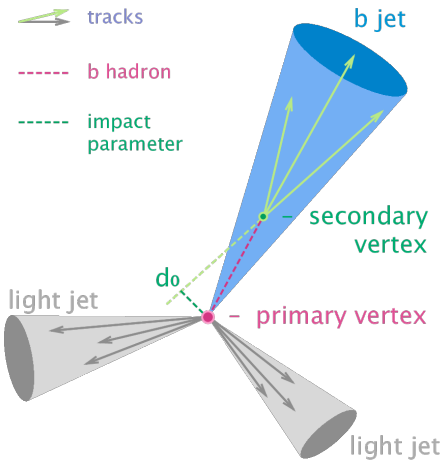


Figure 4.2: Secondary vertex from a b-jet

Moreover, B-hadrons are massive which leads to decay products with larger transverse momentum (relative to the jet axis) in comparison to jets produced by lighter

partons.

B-tagging algorithms have become a crucial components in current data analysis. For instance, for the search of the Higgs boson in the $t\bar{t}H$ channel, these algorithms were used to reduce or even eliminate large backgrounds like $t\bar{t}j\bar{j}$ or $W + jets$. Indeed, the $t\bar{t}j\bar{j}$ background was reduced by two orders of magnitude only using b-jets identification.
When it comes to current experiments, such algorithms remain a crucial component for a successful analysis. Indeed, in the channel probed in this work ($\bar{b}bW^+W^-$), top quarks are decaying into b quarks. The $H \to bb$ decay channel is usually picked in di-Higgs physics experiment since it has the largest branching ratio of all possible decay channels, hence the need of b-tagging algorithms. Moreover, the additional label information used by the cGAN directly depends on b-tagging information since it is a solid criterion to separate the signal region from the background one. Indeed, DY events are not expected to produce two b-jets.

4.5 Morphing

In high energy physics, "morphing" refers to a technique used to interpolate between different simulated events or physical models seamlessly. It allows the exploration of the behaviour of a physical system over a continuous parameter space without the need of plenty of discrete, independent simulations at each point in the parameter space. Morphing techniques typically involve constructing a parameterized mapping between the original and target parameter spaces, often using mathematical functions or interpolation methods. This mapping allows for the generation of simulated events corresponding to intermediate parameter values, providing a more comprehensive understanding of the physics being studied.

In the LHC, the use of morphing techniques for systematic uncertainties is a very common thing. These methods are used to assess the impact of systematic uncertainties on measurements of several parameters as mass, cross-section, ... For example, for uncertainties of the simulation of a physical process, morphing techniques can smoothly interpolate to different simulation in order to estimate the effect of these uncertainties on the final result. [40]

4.6 Application of the cGAN-based approach

In the paper which this thesis is based on, an ATLAS search [41] for Higgs boson decaying to a Z boson and a light hadronically decaying resonance a is taken as case study. In the ATLAS original paper, several of the above techniques are used to perform the analysis. Indeed, a variant of the ABCD method and a MC-based method are used to account for the correlation between several variables. This search is facing two main obstacles: first, for a $\mathcal{O}(100fb^{-1})$ dataset, it is impossible to generate a simulated event sample with a comparable statistical power. Second, the decaying resonance a is identified with a multi-variate methods, requiring a detailed modelling of a several correlations between variables related to kinematics and jets. The sensitivity of the initial search is thus limited by systematic and statistical uncertainties.

Both of these obstacles come from the insufficient size of simulated samples implied by the techniques used originally. However, the cGAN-based approach is able to overcome these. Indeed, with the possibility of generating larger samples, the statistical uncertainties could be suppressed. Thus, allowing a performance improvement, as proved in the case study mentionned. There will still be some remaining uncertainties coming from the training of the cGAN. To mitigate these, we plan to run several networks (5, typically) and combine their results. Moreover, the cGAN is able to target a region (e.g. signal region, background region,...) in which the data will be generated, which is impossible with the current methods. This approach could also be directly based on real data and not MC simulations. Indeed, we could use the official CMS data as a training sample for the network.

This work aims to achieve a similar goal but for a different purpose. As mentionned in [35], for the di-Higgs physics $b\bar{b}W^+W^-$ final state, several backgrounds are considered such as: $t\bar{t} + jets$, single-top production, WW processes as well as Drell-Yan among others. Despite not being generated with the same softwares (*MadGraph5_aMC@NLO* for DY, *POWHEG* [42] for the others mentionned), these backgrounds components are still coming from the same simulation strategy: Monte Carlo. In addition, a very similar approach to the ABCD method is also used to estimate DY events. We want to assess whether or not the cGAN approach is viable alternative to the "MC + ABCD + morphing" combination for the $b\bar{b}W^+W^-$ case.

Chapter 5

Neural networks

In machine learning, a neural network (NN) is a model inspired by the neuronal organization found in the biological neural networks in animal brains.

A NN is made of connected units or nodes called artificial neurons, which loosely model the neurons in a brain. These are connected by edges, which model the synapses in a brain. An artificial neuron receives signals from the previous connected neurons, then processes them and sends a signal to following connected neurons. The "signal" is a real number, and the output of each neuron is computed by some non-linear function of the sum of its inputs, called the activation function. Neurons and edges typically have a weight that adjusts as learning proceeds. The weight increases or decreases the strength of the signal at a connection.

Typically, neurons are aggregated into layers. Different layers may perform different transformations on their inputs. Signals travel from the first layer (the input layer) to the last layer (the output layer), possibly passing through multiple intermediate layers (hidden layers). A network is called *shallow* if it has few layers (3 or 4 in total) or *deep* neural network if it has more than 3 or 4 total layers. In this project, we will focus on deep neural networks (DNN).

Artificial neural networks are used for predictive modeling, adaptive control, and other applications where they can be trained via a dataset. They are also used to solve problems in artificial intelligence. Networks can learn from experience, and can derive conclusions from a complex and seemingly unrelated set of information.

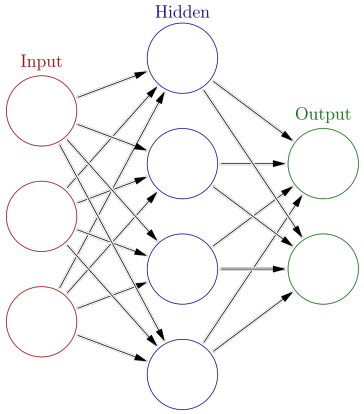


Figure 5.1: Neural network with a single hidden layer

5.1 Basic concepts of Neural Network

Artificial neurons

An artificial neuron is a mathematical model. In most cases, it computes the weighted average of its input and then possibly applies a bias to it, which won't be the case in this work. Afterwards, it passes this result through an activation function. This function is a nonlinear one that accepts a linear input and gives a nonlinear output.
In addition to the connection to other neurons and weights, a threshold can be implemented for every neuron. If the output of any individual node is above the specified threshold value, that node gets activated. In that case, it sends data to the next layer of the network, otherwise, it remains inactive and doesn't transmit any data to the next layer of neurons.

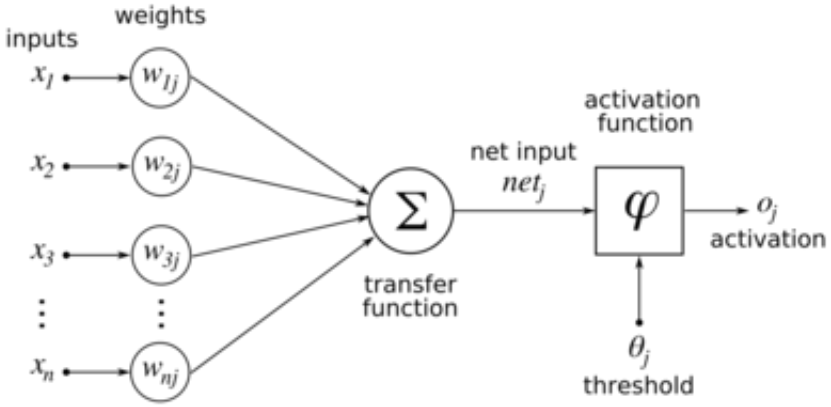


Figure 5.2: Operation of an artificial neuron

Backpropagation

In machine learning, backpropagation refers to a method which computes the gradient of a loss function, i.e. a function representing the price paid for inaccuracy of predictions in classification problems, with respect to the weights of the network. It is computing the gradient one layer at a time, iterating backward from the last layer. It is then used to update the different parameters of the network.

Exploding and vanishing gradients

The cases of vanishing and exploding gradients happen during the backpropagation when the slope of the activation function become progressively smaller or greater as we move backward through the layers of the NN. Obviously, this problem gets worse with DNN. The weight updates becomes either extremely small or extremely large, depending on the case, meaning that it will cause to completely stop the training process of the model.
This problem can be adressed by using specific activation functions, or using *batch normalization* [43] that normalizes the inputs of each layers, reducing the risks of vanishing/exploding gradient.

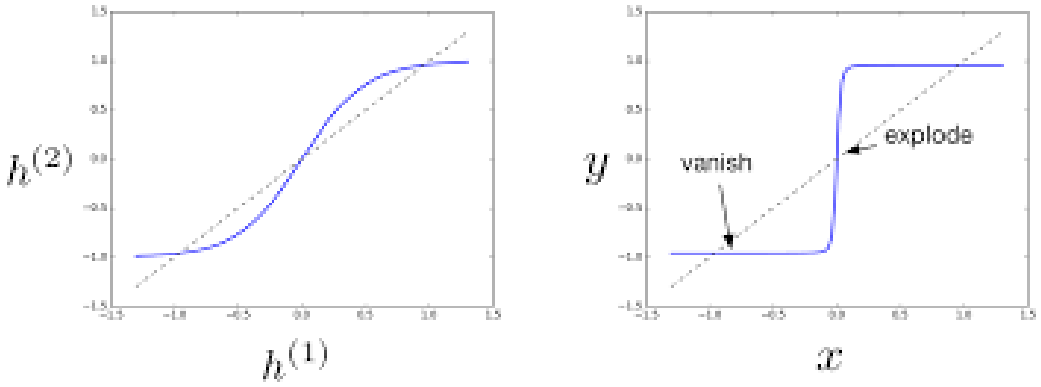


Figure 5.3: Gradients. Left: standard behaviour. Right: Vanishing and exploding gradient

Under and over-fitting

When using NNs for supervised learning, the procedure is usually to divide the data sample in 2 differents subsets: the training set (around 75% of the initial set) and the testing set.[1] However, it can happen that the NN learns too well the training set, as in Fig.(5.4). In this case, the model will perform very accurately on the training set, but once it will be tested on the other set, it will result in mediocre performances. In other words, the NN will learn "by-heart" the training set and won't be able to perform satisfyingly on a different sample, e.g. the testing one. This is called *overfitting*. It represents an important challenge in the construction of a NN. One of the main causes of overfitting is a too complex model. Several tools can be used to avoid the overfitting of a model, we are going to discuss some later on.

On the order hand, there is underfitting, which represents a lack of training of the model leading to a too simplistic model unable to fit properly the data.

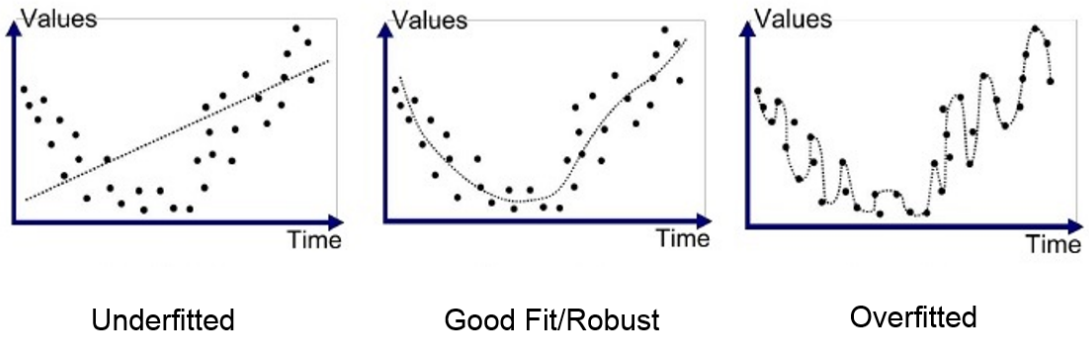


Figure 5.4: Left: underfitting. Middle: fitting. Right: overfitting.

[1]Note that a 3^{rd} set, the validation set, can also be used.

5.2 Generative Adversarial Network (GAN)

A GAN [44] is a specific class of machine learning framework used to approach generative modelling. Generative modelling is an unsupervised task, discovering and learning the patterns within the input data in order to generate new, fake but plausible, examples.
The network is divided in two sub-models, the generator and the discriminator, working in an adversarial way. The generator will create plausible examples based on a latent vector of given dimension. The discriminator will determine whether the example provided is from the input sample (actual data) or if it is generated (fake data). Once the discriminator reaches a validity score of about 50%, the network is producing credible examples.

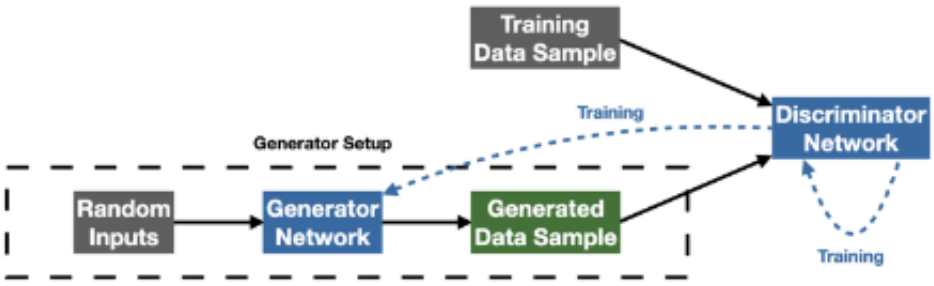


Figure 5.5: Schematical representation of a GAN

The majority of GANs examples available are image-related GANs. The most popular applications are trained on databases such as MNIST [45], a set of hand-written digits or CIFAR-10 [46], which contain many differents datasets, one of them being a set of images of different vehicles such as cars, motorbikes, planes, ... However, in this work, images are not the desired output, we are rather aiming for numerical data. This represents a challenge in comparison to all the documentation available online. Beside the use of GANs in particle physics, application of numerical data GANs are also found in the sector of finances. [47]
During the most part of this work, the input sample data used has been generated via *MadGraph*. However, actual data can be used as input, leading to a more realistic training sample.

In the field of high energy physics, the use of GANs would solve the issue of limited simulated data samples by allowing large generated samples to be produced from much smaller datasets. However, these samples are not perfect, there are intrinsic mismodellings tied to these samples. Thus, it causes some of the largest sources of uncertainty in searches and measurements at the LHC.
The approach to overcome this issue is to slightly modify the network, as discussed in the following section.

5.3 Conditional GAN

A conditional GAN (cGAN) [48] is very similar to a standard GAN except it is able to conditionally generate samples based on an additional piece of information

provided to both the generator and the discriminator. This additional information is called the *label*, allowing the network to return specific outputs. This level of control isn't available with standard GANs.
In the case of this work, this additional label will be the region of the sample, i.e.: signal region or control region.

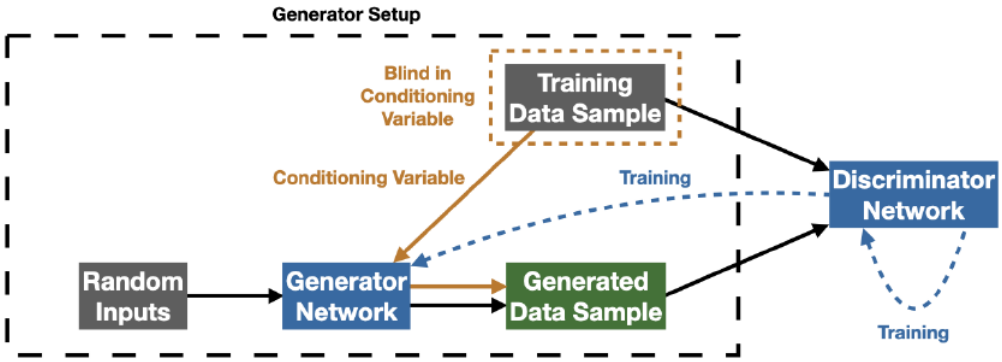


Figure 5.6: Schematical representation of a cGAN

While training a standard GAN using blinded data it falsely informs the GAN that there are no events in the SR, leading to a generative model which predicts an absence of background events in the SR. Conversely, the cGAN learns the distribution of the background features conditioned on the blinding variable and so, despite being given no information about the background in the SR, can extrapolate its prediction into the SR. The cGAN can then be provided with the inclusive distribution of the blinding variable for all data events, and use what it learns in the unblinded data to interpolate the conditional generative model into the signal region, thereby predicting the values of the other variables.
This approach is believed to be unbiased and to be less subject to mismodellings.

Activation function

The activation function of a node in NN is a function that calculates the output of the node based on its individual inputs and their weights. Nontrivial problems can be solved using only a few nodes if the activation function is nonlinear.
There is plenty of activation functions available, we will need to carefully choose the most appropriate to our goal. We can mention some popular functions, such as the sigmoid and the hyperbolic tangeant. They are both nonlinear, which is a crucial criterion in this case. However, the main drawback of these function are their limited sensitivity. Indeed, their nonlinear behaviour only stands in a short interval around 0, decreasing the sensitivity of the network for both large positive and large negative values. Moreover, these are relatively complex function to compute, due to the presence of exponentials in their mathematical formulation.

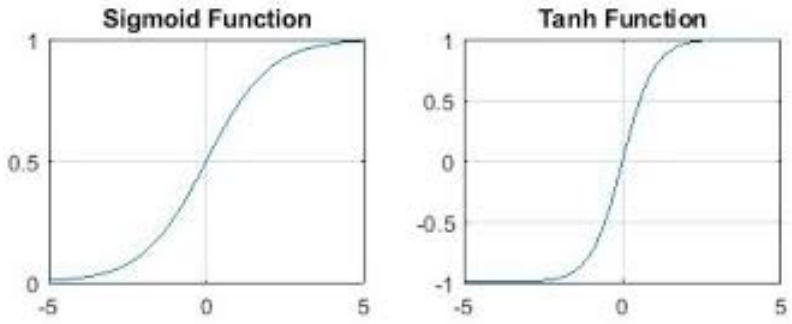


Figure 5.7: Activation functions. Left: sigmoid. Right: hyperbolic tangeant

Now, we need a nonlinear activation function, with a sensitivity to large values and easy to compute. A function checking *almost* all the boxes is the rectified linear unit (ReLU)[49]. However, ReLU is not sensitive large negative value, as one can see on Fig. (5.8) which can cause issues. This leads us to our final choice: leaky rectified liner unit (Leaky ReLU). It stands apart from the standard ReLU thanks to the small gradient in the $]-\infty, 0]$ region. It allows the function to stay active for negative values, avoiding the case of a never activating neuron. Thus, it greatly improves the performance of the network despite adding another hyperparameter (let's call it α) being the slope of the function in the negative region.

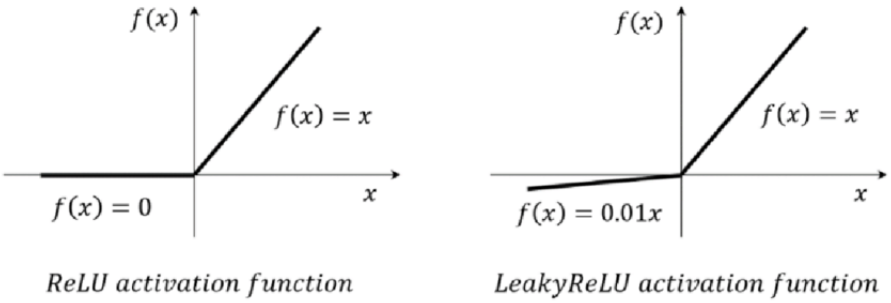


Figure 5.8: Activation functions. Left: ReLU. Right: leaky ReLU

The leaky ReLU is not the only variation of the ReLU function, there is also GELU [50], ELU [51], SELU [52]. Despite being quite recent (less than 10 years), these activation functions are widely used nowadays thanks to their numerous advantages.

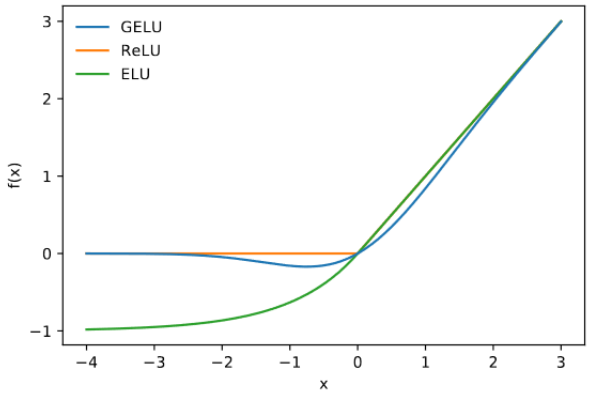


Figure 5.9: GELU and ELU function in comparison to standard ReLU

Despite all the advantages of leaky ReLU over the other activation functions, the former is not used for all the layers of the network. Indeed, the output layers of both the generator and discriminator are using the sigmoid function, as adviced in [47]. One of the advantage is the easier computation of the results, since sigmoid in strictly bounded to $[0, 1]$, which isn't the case of any ReLU variant.

Batch and batch size

Batch size is a crucial component in deep learning training, it represents the number of samples (batches) used in one forward and backward pass through the network and has a direct impact on the accuracy and computational efficiency of the training process. Large batch sizes tends to lead to faster trainings but may result in lower accuracy and overfitting, while smaller batch sizes can provide better accuracy, but can be computationally expensive and time-consuming. The batch size can also affect the convergence of the model, meaning that it can influence the optimization process and the speed at which the model learns. Small batch sizes can be more susceptible to random fluctuations in the training data, while larger batch sizes are more resistant to these fluctuations but may converge more slowly.

Binary cross-entropy

When developping a NN, we need a metric or a function describing the performance of our network, this will help us to optimize our model. If the predictions are close to the actual values, the loss function will be minimum, but if the predictions are far away from the actual values, it will be maximum.
There exist several different loss functions, the best choice depends on the problematic we are facing. In this case, we have a binary classifier (does the example has been generated or does it come from the actual data?) so we choose a loss function called *binary cross-entropy* [54].
The standard cross-entropy is, in short, a tool to measure the difference between two distributions over the same set of events. With the *entropy* being the number of bits required to transmit a randomly selected event from a probability distribution. For example, a skewed distribution has a low entropy while an equal probability distribution has a larger entropy. Another name for *binary cross-entropy* is *log loss*, in that expression it's easy to understand what makes this loss function interesting for

us: the use of logarithms. Indeed, these will penalize heavily incorrect predictions. Here's how binary cross-entropy is computed:

- If the label is 1, the cross-entropy is $-log(p)$ where p is the predicted probability
- but, if the label is 0, the cross-entropy is $-log(1-p)$.

Then, the cross-entropy values are summed up for all examples. It can be written, under mathematical notations as:

$$\text{Log loss} = \frac{1}{N}\sum_{i=1}^{N} -\left(y_i \log(p_i) + (1-y_i)\log(1-p_i)\right), \tag{5.1}$$

where y_i represents the actual class, p_i is the probability of class "1" and $1-p_i$ the probability of class "0".

Stochastic gradient descent optimizer

Optimization algorithms are frequently used in machine learning to identify the best set of parameters that minimize the loss function.

In stochastic gradient descent (SGD) [56], the algorithm quickly learns the direction of steepest descent using a single example of the training set at each time step. While this method has the distinct advantage of being fast, it may never converge to the global minimum. However, it approximates the global minimum closely enough. In practice, SGD is enhanced by gradually reducing the learning rate over time as the algorithm converges. In doing this, we can take advantage of large step sizes to go downhill more quickly and then slow down so as not to miss the global minimum. Due to its speed when dealing with humongous datasets, SGD is a popular choice.

It is useful to mention the existence of another popular optimizer: Adam. It has been proven that Adam outperforms SGD (with Nesterov momentum) on the MNIST data set [57]. However, no performance gap between these two optimizers has been observed in this project.

Learning rate

Learning rate (LR) is a common parameter of optimization algorithms that controls how big a step the gradient descent algorithm takes when tracing its path in the direction of steepest descent in the function space.

If the learning rate is too large, the algorithm takes a large step as it goes downhill. In doing so, gradient descent runs faster, but it has a high chance of missing the global minimum. Conversely, a too small learning rate makes the algorithm slow to converge (i.e., to reach the global minimum), but it is more likely to converge to the global minimum steadily. Empirically, examples of good learning rates are values in the range of 0.001, 0.01, and 0.1. In Fig.(5.10), with a good learning rate, the cost function $C(\theta)$ should decrease after every iteration.

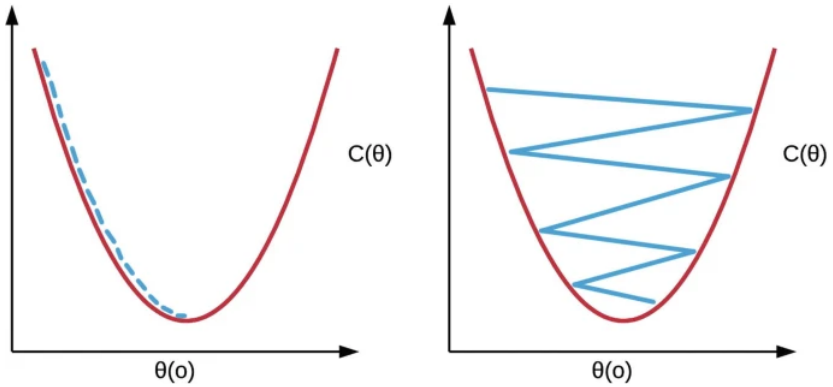


Figure 5.10: Learning rates. Left: Small learning rate. Right: Large learning rate.

In lot of low-level cases, the learning rate can be set at a constant value during the whole process of training. However, it is possible to adjust it dynamically during this process in order to reach better performance of the network. Doing so, two important obstacles can be tackled. In one hand, it allows the network to get out of local minima, and thus to converge to the global minimum with greater ease. On the other hand, as already proven [58], saddle point are also critical points in optimizing paths. The gradients at saddle points tends to be very small and, thus, can slow the learning process. However, tweaking the learning rate allows the rapid traversal of saddle point plateaus.
This will be discussed in greater details in a future section.

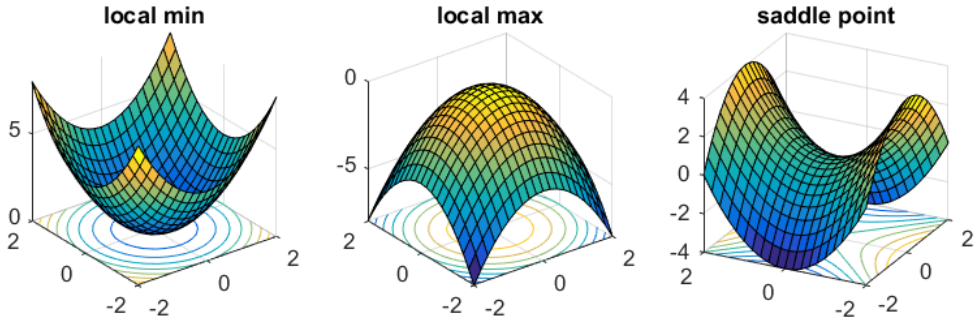


Figure 5.11: Left: local minimum. Middle: local maximum. Right: saddle poiny

Nesterov momentum

One issue with SGD is that it can oscillate and take a long time to converge to a minimum, especially when the loss function has a complex structure or is highly non-convex. To mitigate that issue, we can add another parameter to the optimizer: the momentum. Momentum is a technique that helps to mitigate this issue by adding a momentum term to the update rule.

The momentum term is essentially a weighted average of the past gradients, with the weighting decreasing exponentially as the gradients get further in the past. This helps to smooth out the oscillations, avoid local minima and accelerate convergence by allowing the optimizer to take larger steps in the direction of the minimum.

The Nesterov momentum [59] is a variation aiming to improve the traditional momentum by making a subtle yet powerful change to the update rule. Instead of calculating the gradient at the current position, Nesterov's Momentum calculates the gradient at a position slightly ahead in the direction of the accumulated momentum. This look-ahead step allows the optimizer to correct its course more responsively if it is heading towards a suboptimal direction.[2]

L2 regularization

Regularization is used in machine learning to avoid, as much as possible, overfitting. The idea is to add a penalty to the loss function as the model complexity increases such that the importance given to high order terms will decrease.

$$J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 + \lambda \sum_{j=1}^{n} \theta_j^2\right], \quad \text{with the goal of minimizing } J(\theta). \tag{5.2}$$

For L2 regularization in particular, the penalty term (written in blue) added to the loss function is the *squared magnitude* of coefficient. It encourages smaller, more evenly distributed weights.

Dropout layer

A dropout layer is another type of regularization. The idea is to randomly ignore a subset of neurons of a specific layer during the training session, simulating training multiple neural network architectures to improve generalization.

He initialization

In a NN, when weights are initialized randomly it can pose problem for the convergence of the network. To solve this problem, these weights have to be initialized in a specific way, that depends on the activation used in our model. In this case, we use an appropriate initialization for the (leaky) ReLU function: the *He* weight initialization [60]. The idea behind this concept is to initialize weights in the following range:

$$N\left[\left(-\frac{\sqrt{6}}{\sqrt{n_i(1+\alpha^2)}}, \frac{\sqrt{6}}{\sqrt{n_i(1+\alpha^2)}}\right)\right] \tag{5.3}$$

with N a normal distribution, n_i is the number of incoming network connections in the layer and α is the parameter of leaky ReLU function.
It's also useful to note that initializing weights can help furthermore mitigating exploding and vanishing gradients.

Gradient clipping

Gradient clipping [61] adresses the problem of vanishing and/or exploding gradient by imposing a threshold on the gradients. If the gradients exceed this predefined

[2]For a indepth mathematical formulation, please check: [59]

threshold, they are rescaled to ensure they do not surpass the set limit. This rescaling step helps to keep the gradients within a manageable range, thus preventing drastic updates to the model's parameters that might lead to instability or divergence during training.

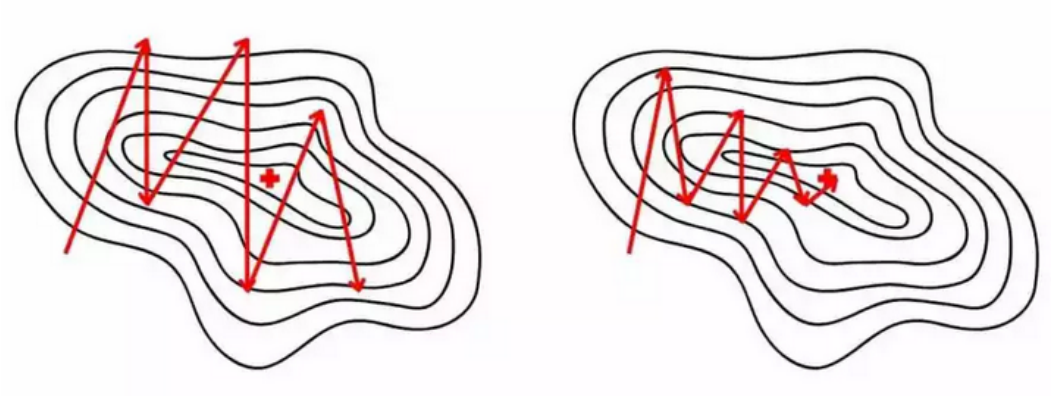


Figure 5.12: Gradient clipping. Left: without gradient clipping. Right: with gradient clipping.

Data preprocessing

Due to the tools used for this project, standardization of datasets is a requirement for many machine learning estimators. Data might behave badly if the individual features do not more or less look like standard normally distributed data. In other words, Gaussian with zero mean and unit variance.
Several techniques exist, one of the most common one being the normalization or standardization of the input distributions in order to have a new distribution with a mean value of 0 ($\mu = 0$) and standard deviation of 1 ($\sigma = 1$). This process is done independently for each feature of the data. Given the distribution of the data, each value in the dataset will have the mean value subtracted, and then divided by the standard deviation of the whole dataset. In mathematical formulation, it represents:

$$z = \frac{x - \mu}{\sigma} \tag{5.4}$$

$$\text{with } \mu = \frac{1}{N}\sum_{i=1}^{N}(x_i), \text{ and: } \sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2}. \tag{5.5}$$

However, the method used in this work is a straightforward rescaling of the inputs. Indeed, since the activation functions used in this project are the sigmoid and leaky ReLU, it's convenient to rescale our data as [0,1], since it's the appropriate range for these functions to operate efficiently. Moreover, it helps to improve the stability of the network, which is crucial with a GAN. Neural networks in general are sensitive to the scale of input features, and having all the features within a similar range can prevent some of them to dominate the learning process.

Now let's consider a network with several hidden layers. As explained, the data preprocessing relates to the input of our network. From the perspective of the second layer, the output of the first layer simply corresponds to its inputs. Hence, it can

be normalized. This concept is called *batch normalization*. [62] However, it depends on a momentum parameter, adding yet another hyperparameter to the list.

5.4 Adaptive learning rate techniques

As stated in the dedicated section, the learning rate can be a complex hyperparameter to correctly tune. Indeed, both a too large and a too small learning rate will cause a poor training of our model. The most straightforward solution would be to determine judiciously a suitable LR for our model. Unfortunately, it's easier said than done.
A better idea is to have recourse to adaptive learning rates techniques. These methods will dynamically modify the value of LR during the training session, depending on the number of epochs already performed or on the performance of one or several specified metric(s).

5.4.1 Learning rate scheduler

This method tends to reduce the value of LR as the training session goes on, following a predefined schedule. The idea is to start with a large LR, in order to get closer quickly to the local minima. Then, the LR will be progressively decreased at each epoch in order to avoid overstepping the targeted minima.
However, the behaviour of this schedule has to be defined by the user, adding yet another pseudo-hyperparameter to the already long list of hyperparameters. Indeed, the scheduler can adopt a linear decrease strategy, exponential, time-based or even a step decay one.
Several variations of learning rate schedulers have been used for this project, but none of them have been retained due to a lack of performance improvement. [63]

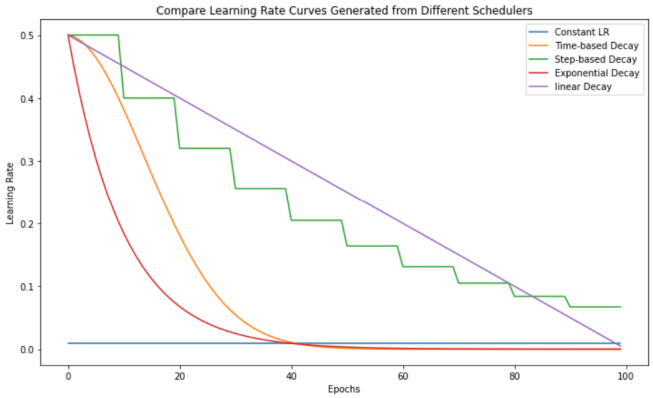
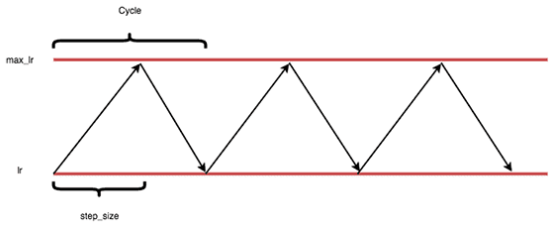


Figure 5.13: Different scheduler behaviours

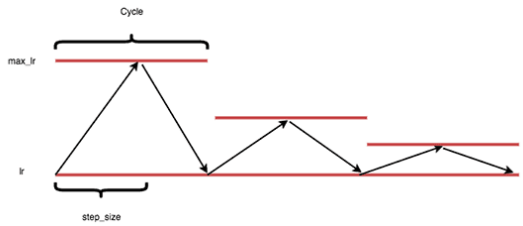
5.4.2 Cyclic learning rate

As it name may suggests, this method implies the concept of cyclical variation of LR. This technique tackles another important concept beside the local minima: the saddle points. As stated in a previous section, saddle points may slow down the learning process. Moreover our network could stay stuck in a local minimum. To prevent these issues, an important LR is required, here's why the cyclic behaviour

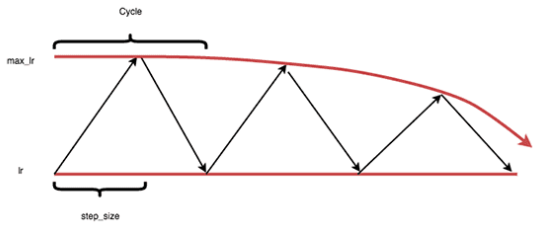
is interesting. As for learning rate scheduler, there are several possible strategies, let's breakdown the three main ones.



(a) Cyclic triangular behaviour



(b) Decreasing cyclic triangular behaviour



(c) Cyclic exponential behaviour

Figure 5.14: Different cyclical behaviours

The (5.14b) and (5.14c) cases mix the concept of "steadily" decreasing LR with cyclical LR. In this work, the decreasing triangular behaviour has been retained due to its better performance with the network.

However, this method isn't included in the *keras.callbacks* package anymore. It could be found in the *tensorflow.addons* package, unfortunately, it is deprecated since no major updates have been made recently. Thus, it may cause package clashes. To address this issue, the method had to be re-implemented with a few improvements.

5.4.3 Reduce learning rate on plateau

This technique is a scheduler variant. However, it is not based on the number of epochs performed but rather on the evolution (or absence of evolution) of one or more metric(s). The user needs to specify the metric to monitor as well as a *patience* parameter, i.e. the number of epoch without any improvement of the metric, once this number exceeded a modification of the LR will be applied. Conversely to the other two methods, the change of LR is applied punctually, creating plateaux in the LR evolution, hence its name. [64]

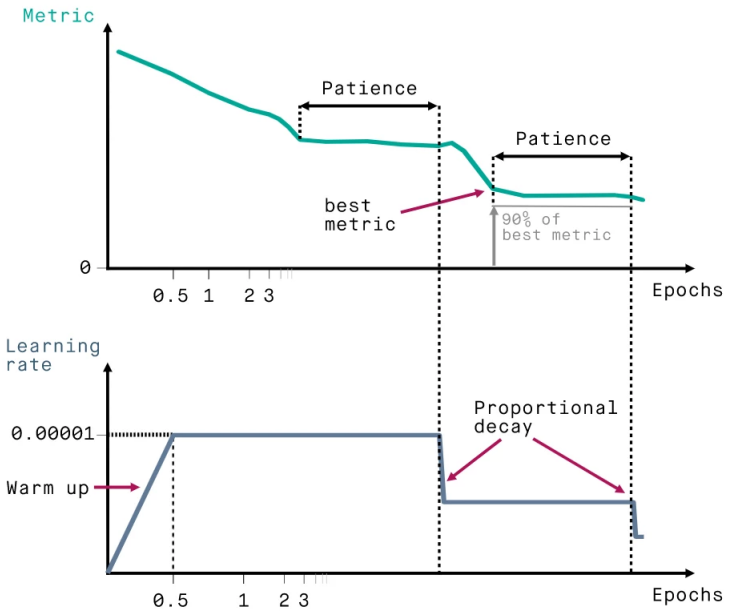


Figure 5.15: Behaviour of the reduce LR on plateau method for both LR and the monitored metric

Despite being trialed, this strategy hasn't been retained.

5.5 Architecture

Here is the composition of the cGAN model used for this work:

- 4 layers for both discriminator and generator with 32 nodes each,
- SGD optimizer with momentum: 0.8 and gradient clipping limited at: 1.0,
- dimension of the latent vector set to 15,
- He initialization,
- leaky ReLU with $\alpha = 0.4$ for all layers except the last one of both networks, sigmoid otherwise,
- batch size set at 800 with batch normalization momentum at 0.8,
- the learning of the discriminator is set to *False* when the generator is training,
- cyclic learning rate with decreasing triangular behaviour with maximum LR: 0.05, minimum LR: 0.00005, step size = 2000.

5.6 Technical details

As mentionned earlier, *MadGraph5* has been used to generate a MC sample. The code has been written in *Python*, using *TensorFlow* [65] and *Keras* [66] packages for the machine learning parts. The *upROOT* [67] package has been used for big data processing.

5.7 Challenges

Although capable of generating very accurate synthetic samples, GANs are also known to be hard to train [68] . Training two networks simultaneously means that when the parameters of one model are updated, the optimization problem changes. This creates a dynamic system that is harder to control. Non convergence is a common issue in GAN training. Deep models are usually trained using an optimization algorithm that looks for the lowest point of a loss function, but in a two-player-non-cooperative-game scenario, instead of reaching an equilibrium, the gradients may conflict and never converge, thus missing the global minimum. In other words, if the generator gets too good too fast, it may fool the discriminator and stop getting meaningful feedback, which in turn will make the generator train on bad feedback, leading to a collapse in output quality. An issue remains in the opposite case, even if these two networks are working in an adversarial way, one cannot outperform the other without compromising the performance of the GAN. These are widely known problems and several attempts were made to improve the stability of GANs. [69]

At some point of this project, these issues were translated as such: the same network with the exact same set of hyperparameters can produce a totally different output distribution from one run to another, making it extremely difficult to fine-tune efficiently hyperparameters as no stable benchmark is available. Several methods were used to address this issue, as referenced previously. All of them improved the performance of the network. However, only the cyclical learning rate strategy was retained due to its better efficiency over other methods.
Plenty of other improving methods were proposed in this paper: [69].

Moreover, a very effective way to adress the stability issue is by adapting our network to the desired goal. Indeed, there exists plenty of variations in GAN architecture, a non-exhaustive but important list can be found in [70].

Chapter 6

Network development history

6.1 Training Sample

For this project, *MadGraph5* [71] was used to generate a sample of 20,000 Drell-Yan events, incorporating the 2HDM extension for *MadGraph5* at a center-of-mass energy of 13.6 TeV. Then, a selection criterion was applied to the sample to retain only events where the di-lepton system consists of a pair of muons. This cut reduces the number of events to approximately 6,000.
An important note must be made here. In the context of the 2HDM, there exist several processes resulting in a pair of leptons from a proton-proton interaction, more than in the SM. Therefore, it is crucial to specify the expected mediator bosons. In our case, these are the Z and γ bosons. However, in the 2HDM, the three neutral Higgs bosons can also serve as mediators.

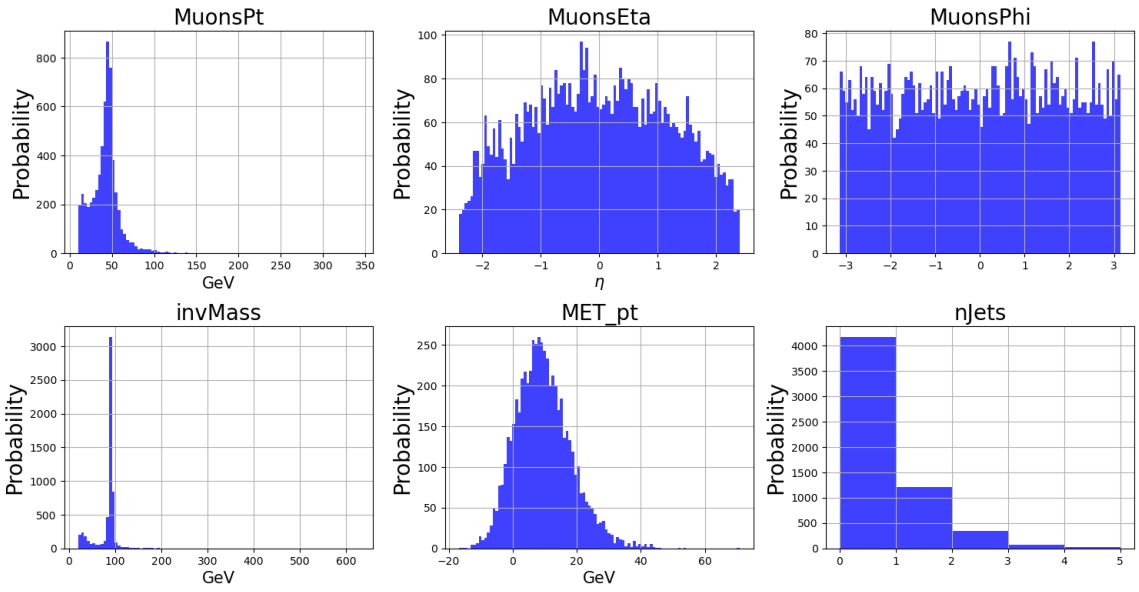


Figure 6.1: Some relevant observables of the DY sample

The proportion of b-jets in the sample are :

DY events	Proportion
No b-jets	99.5%
1 b-jet	0.5%
2 or more b-jets	0%

Table 6.1: Proportion of DY events with x b-jets.

6.2 Adaptation from an image-processing GAN

The original code used as the basis for this work can be found at [72]. Originally designed to train on the MNIST dataset and generate samples of handwritten digits, this type of GAN includes components specifically designed for image processing, such as convolutional layers and maximum/average pooling. Since these are not applicable to our task, they were removed or replaced in our GAN architecture.
Furthermore, the structure of the manipulated objects in image-processing GANs is highly specific. Handwritten digits, for instance, are stored as two-dimensional arrays, with each entry representing a shade of black and white between 0 and 255. In contrast, other GAN examples, like CIFAR-10, store images as three-dimensional arrays since RGB channels are then involved.
After refining the network structure, the initial results can be visualized and analyzed.

6.3 First results

Following the initial adjustments to transition from image-related to data-related tasks, the distributions generated by two different networks, each using the same set of hyperparameters, are presented. The first network is characterized by a single hidden layer with 8 nodes, while the second network consists of 4 hidden layers, each with 256 nodes.

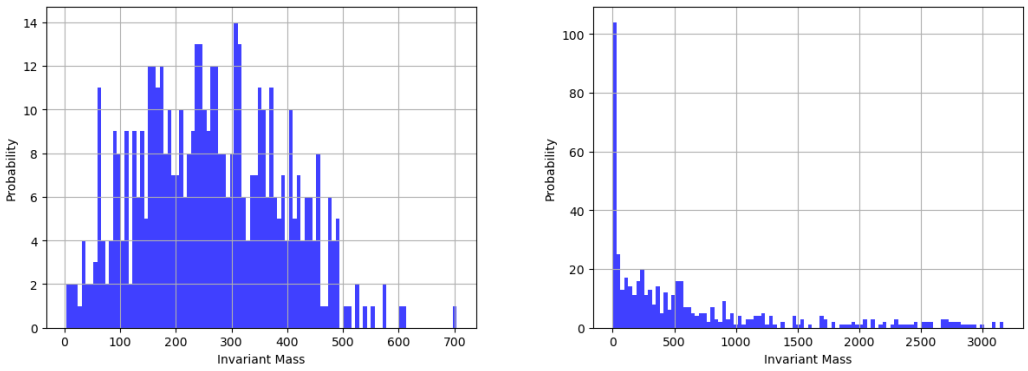


Figure 6.2: Example of output for different networks. Left : simple network. Right : complex network.

6.4 First convergences

Once the model architecture is defined, experimenting with different sets of hyperparameters becomes possible. Initially, the results may appear decent. However, due to factors such as a very limited number of training epochs, the network lacks of stability. As mentioned in a previous section, entirely different output distributions can be generated for the exact same set of hyperparameters. Consequently, fine-tuning the hyperparameters becomes nearly impossible, as there is no consistent benchmark to guide the process.
This instability could be attributed to the loss function becoming trapped in a local minimum or a saddle point, leading to the observed variations in results.

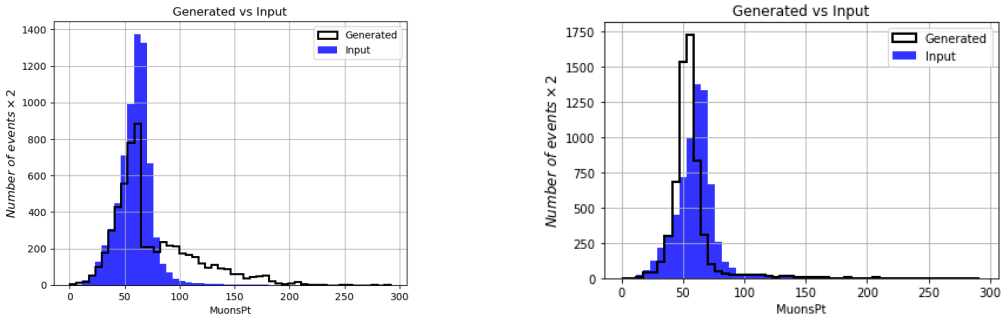


Figure 6.3: Example of output for a same network but with slighlty different learning rate

6.5 Number of epochs and loss functions

The initial approach to overcome local minima was to significantly increase the training duration by a factor of 10-15. Additionally, we monitored the evolution of the loss function for more insights about the training process.
Unfortunately, this strategy resulted in a modification of the output distribution shape, worsening the overall result. This outcome is concerning, as increasing the duration of the training process should ideally guide the network to the expected distribution or, at worst, lead to overfitting. Which is definitely not the case here. Moreover, unexpected behavior in the loss function can reinforce our concerns.
For instance, the decreases in both losses shown in Fig.(6.4c) and Fig.(6.4d) exhibit clear differences. This discrepancy could be explained by the gradient becoming trapped in a local minimum or a saddle point, obvious in Fig.(6.4d) (take care at the vertical axis scale!).
Ultimately, increasing the number of epochs alone isn't a solution to escape local minima and saddle points in our case. We must explore other approaches. However, this adjustment does not necessarily need to be reverted, as it may represent a step in the right direction.

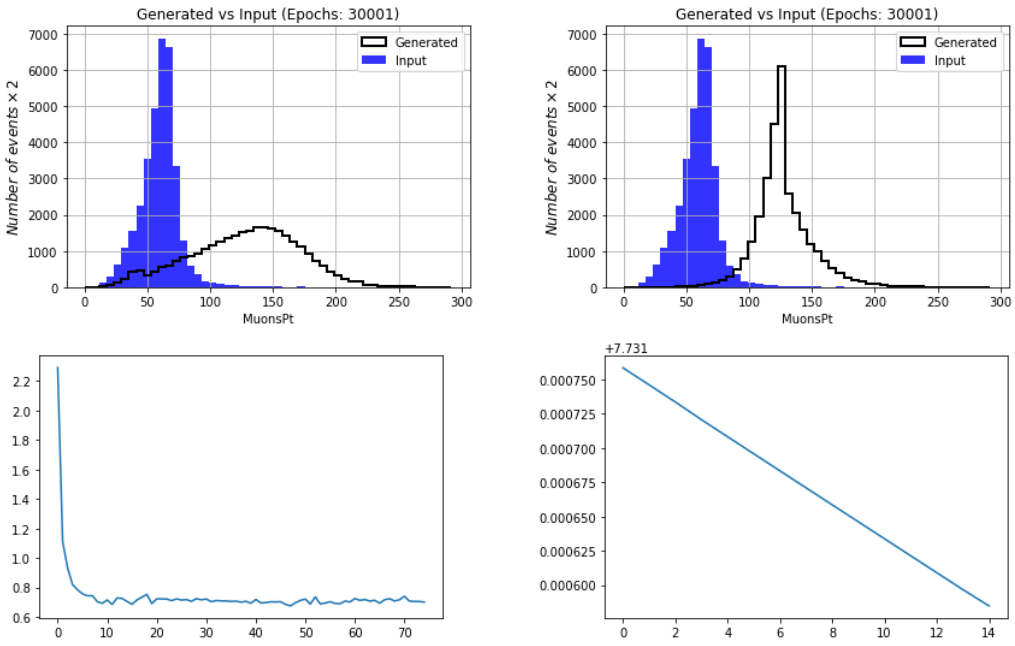


Figure 6.4: Example of output for a same divergent network with the corresponding loss function (binary cross-entropy) evolution.

6.6 Encouraging follow-up?

To tackle the issue of inconsistency, we explored various adaptive learning rate methods, such as cyclic LR, LR scheduler, or reduce LR on plateau. The generated distributions are shown in Fig.(6.5). While the improvements may not be immediately apparent, these adjustments have rendered the network stable. As a result, fine-tuning becomes much more manageable.
However, despite the stability achieved, the network still falls short of generating the expected distribution.
It is worth noting that, from this point, the loss function consistently exhibits a shape similar to Fig.(6.4c).

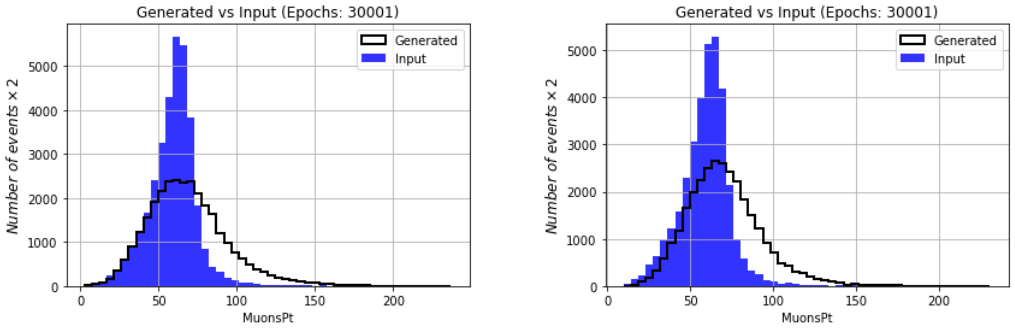


Figure 6.5: Implementation of adaptive LR methods. Left : Cyclic (triangular). Right : Scheduler.

6.7 Encouraging follow-up!

After implementing the *He* weight initializer, batch normalization with a larger batch size, and coupling it with another cyclic learning rate strategy (triangle2), we observe the output of the GAN.
While there is a clear improvement compared to the previous case, the generated distribution still deviates from the input. Notably, the generated simulation seems shifted to the right.

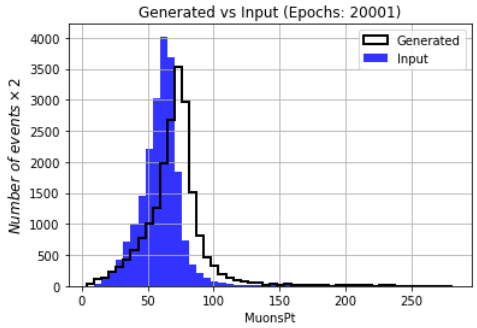


Figure 6.6: Output provided by an almost converging network

6.8 Final result for 1DGAN

After resolving the shifting problem using more adapted rescaling techniques, we obtain the following result for the 1DGAN:

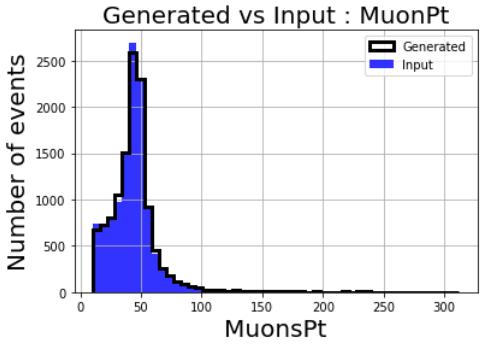


Figure 6.7: Best result obtained so far.

6.9 2-dimensional GAN

As shown in Figure (6.8), the network converges satisfyingly to the input distributions. However, the generated populations struggle to match the peaks of the initial sample.

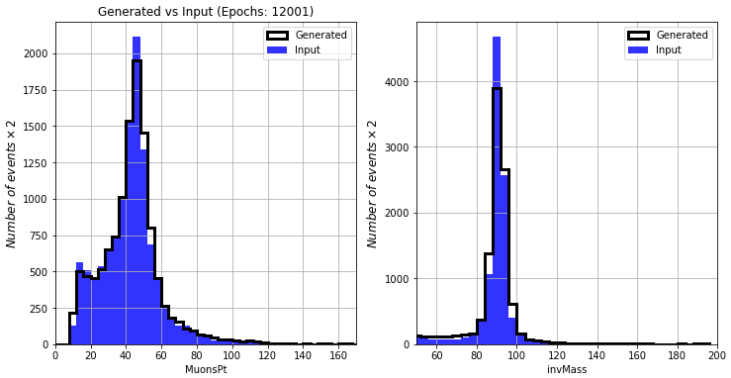


Figure 6.8: Output provided by a 2D GAN. The variables generated are the transverse momentum of muons and the invariant mass of the system

While one-dimensional histograms provide insight into how closely the generated sample matches the initial distribution, our network's primary objective is not only replication, but also the reproduction of correlations between input variables.
To visualize these dependencies, we use two-dimensional histograms, as depicted in Fig.(6.9). The first plot illustrates the correlation between the variables "invMass" and "MuonsPt" for the generated data, while the subsequent plot illustrates the same correlation for the input data. Based on these plots, it can be inferred that the network is satisfyingly reproducing correlations.

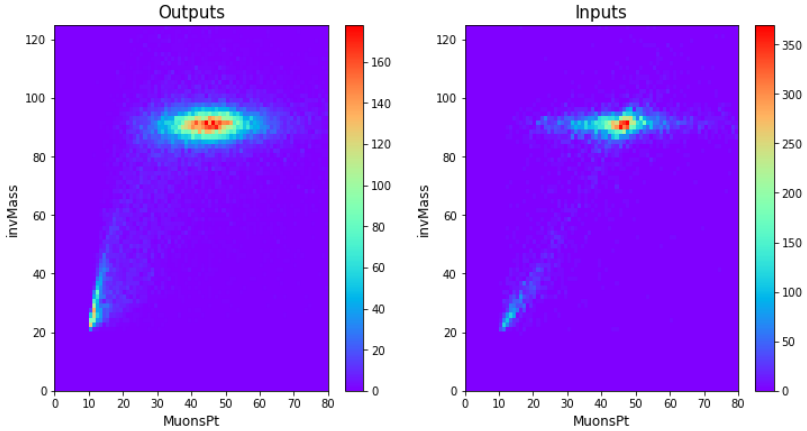


Figure 6.9: Correlations between the two variables. Two zones stand out, the biggest corresponds to Drell-Yan events where a Z is the mediator boson, while the other stands for a γ as mediator boson.

6.10 3-dimensional GAN

Statistical tests

To compare efficiently two distributions, we cannot only rely on visual techniques as one-dimensional histograms, a more objective tool is needed. In the following section, I introduce the *Kolmogorov-Smirnov test* (KS) [73] and the χ^2 *test* to assess whether or not two distributions are considered as sufficiently similar.

Moreover, I use the *mutual information* (MI) [74] to compare two-dimensional histograms used for correlations between variables.
Let's briefly breakdown these tools.

Kolmogorov-Smirnov

This test has several applications. However, in this work, we will use its ability to test whether or not two underlying one-dimensional probability distributions differ. In this case, the Kolmogorov–Smirnov statistic is:

$$\begin{aligned} D_{n,m} &= \sup |F_{1,n}(x) - F_{2,m}(x)|, \\ \text{with } F_{a,b} &= \frac{\text{number of elements in the sample} \leq t}{n} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}_{X_i \leq t}, \end{aligned} \tag{6.1}$$

with n, m the number of events in the distribution, t a fixed parameter and $F_{a,b}$ an empirical distribution function (commonly also called an empirical cumulative distribution function) which can be expressed using a *Bernouilli random variable*: $\mathbf{1}_{X_i \leq t}$. From there, we can compute the p-value, the probability of obtaining test results at least as extreme as the result actually observed. If this value is **greater** than a specified threshold (for instance, 0.05), we conclude a significant association between the two populations. In addition to the p-value, the *scipy* function used also returns the KS statistic. It represents the maximum distance between the two empirical cumulative distribution functions of the two samples.

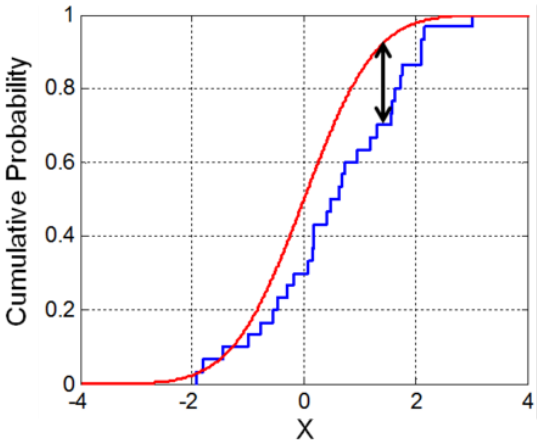


Figure 6.10: KS statistic (in black) between two empirical cumulative distribution functions (red and blue).

χ^2 test

The χ^2 test is a statistical test used to determine whether or not there is a significant association between two distributions. There exist several variant of this test, we will use the most used one: the Pearson variant. It follows the formula:

$$\chi^2 = \sum_{i=1}^{n} = \frac{(O_i - E_i)^2}{E_i}, \tag{6.2}$$

with O_i the number of observations of type i, E_i the expected (theoretical) count of type i. From there, we can compute the p-value. If this value is **less** than a specified threshold (for instance, 0.05), we conclude a significant association between the two populations. Once again, the χ^2 statistic is provided by the *scipy* function, it represents the the discrepancy between the observed and expected frequencies. If this value is large, the two distributions are considered as different.

Mutual Information

The mutual information is a commonly used quantity in information theory that measures the mutual dependence between two random variables. It quantifies the amount of information obtained about one variable by observing the other variable. The MI is tightly bounded to the concept of *entropy*, a notion that quantifies the expected amount of information held in a specific variable.
In this work, I prefer the use of MI over the score of correlation, since the former only measures linear dependency between variables [1].
The mathematical formulation of MI in our case is:

$$I(X;Y) = \sum_x \sum_y P_{(X,Y)}(x,y) \log \left(\frac{P_{(X,Y)(x,y)}}{P_X(x) P_Y(y)} \right), \tag{6.3}$$

with $P_{(X,Y)}$ the joint probability mass function of X and Y, P_X and P_Y the marginal probability mass function of X and Y respectively.

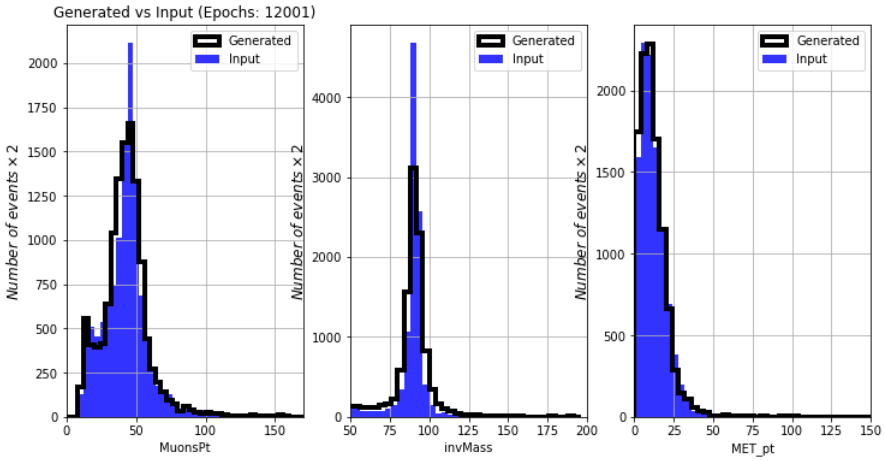


Figure 6.11: Output provided by a 3D GAN. The variables generated are the tranverse momentum of muons, the invariant mass of the system and the missing transverse momentum.

[1] For instance, the correlation score between $\cos x$ and $\sin x$ is tiny, while their dependency to each other is obvious.

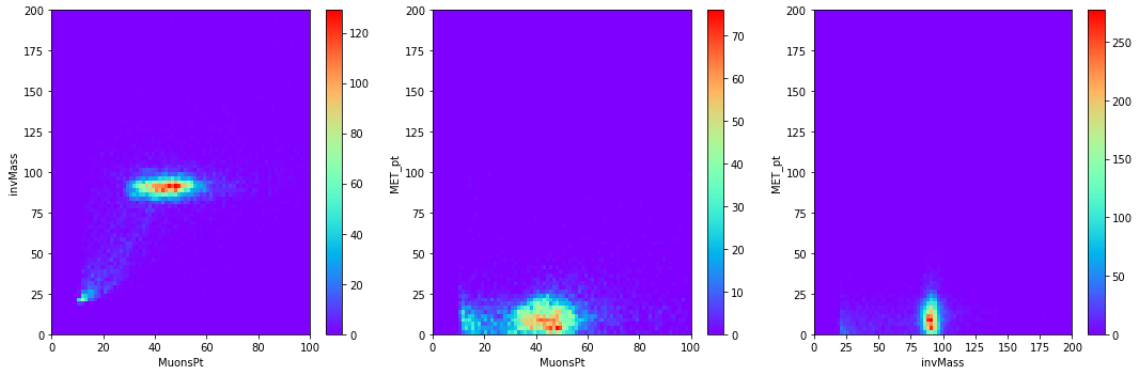


Figure 6.12: Correlations between the two variables generated by the network.

I decide to use the mutual information of two variables to compute the dependency between them. The result obtained is:

Mutual Information score	Outputs	Inputs
MI(MuonPt, invMass)	9.3875	8.6731
MI(invMass, MET_pt)	9.3880	8.6733
MI(MuonPt, MET_pt)	9.3908	8.6731

Table 6.2: Mutual Information Values for a 3-D GAN.

6.11 Conditional GAN

The next step in this work is to transition to a conditional GAN, which involves providing additional information to the generator in the form of a "label". This label indicates the type of data that the network will generate. In this case, we designate the presence of b-tagged jets as the label, creating two classes: samples without b-jets and samples with at least one b-jet. It's worth noting that this variable is blinded in the training sample.

Initially, we assess the performance of the network on each variable.

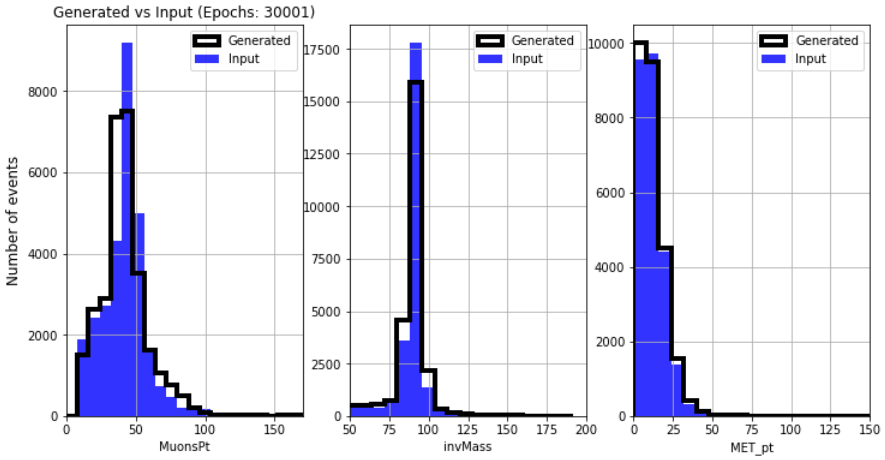


Figure 6.13: Output provided by a 3D cGAN.

In addition to visual tools, we use statistical tests such as the *Kolmogorov-Smirnov* (KS) and the χ^2 to quantitatively assess the similarity or difference between the evaluated distributions. For the KS test, the obtained results are as follows:

KS	**p-values**	**statistics**
MuonPt	0.007	0.433
invMass	0.007	0.433
MET_pt	0.239	0.267

Table 6.3: KS test between inputs and outputs for each observable.

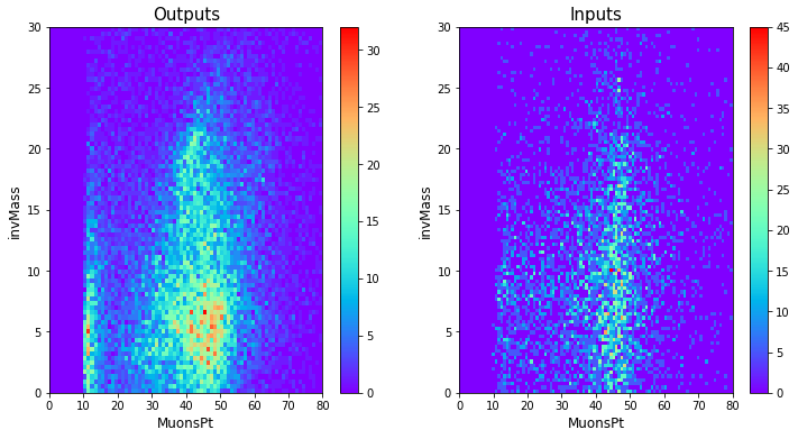
Despite not being immediately apparent with individual examples, it's notable that the p-value for the missing transverse energy (MET_pt) tends to be significantly larger than others in each run. This could be due to the smooth shape of the distribution of this observable.
If we set the significance threshold at 5%, only the MET_pt generated distribution would be considered similar to the corresponding observable in the training sample. And for χ^2:

χ^2	**p-values**	**statistics**
MuonPt	0	668.134
invMass	0.00002	71.200
MET_pt	0.00003	70.594

Table 6.4: χ^2 test between inputs and outputs for each observable.

The results obtained with the χ^2 test are poor and do not provide significant information. Therefore, we will rely on the KS test from now on.

As seen in Fig.(6.16), we can visualize the dependencies between the three selected variables for the generated data and the training sample. Although the zones for outputs are larger, the general behaviour of the dependencies remains similar.



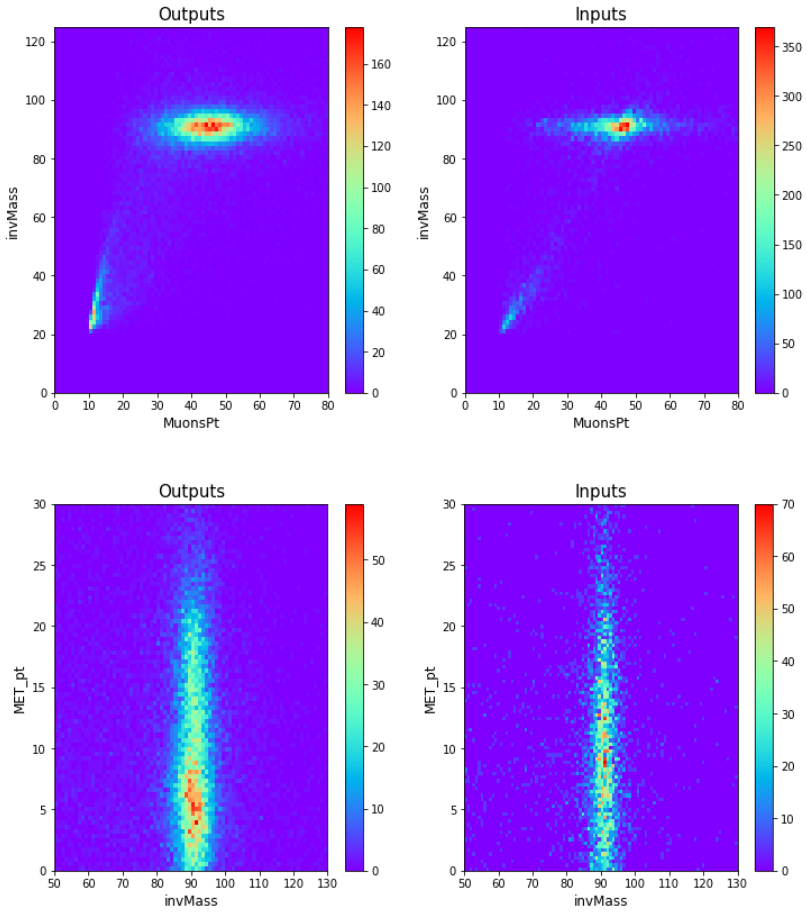


Figure 6.14: Comparison of the input and output correlations between the different variables used. Important note: the color scales are different!

To get numerical values of the dependencies between observables, we can also compute the mutual information values for each pair of variables, the obtained values are:

Mutual Information score	**Outputs**	**Inputs**
MI(MuonPt, invMass)	10.3075	8.6731
MI(invMass, MET_pt)	10.3080	8.6733
MI(MuonPt, MET_pt)	10.2908	8.6731

Table 6.5: Mutual Information Values for a 3D cGAN.

Understanding mutual information scores can indeed be challenging due to their range being $[0, +\infty[$. However, the key is not the exact numerical values, but rather the similarity between the variables of the input and the output. The mutual information scores orbit around similar values in both cases. This suggests that the correlations between observables are close to our desired goal, despite being slightly larger than expected.

Another interesting result to show is the following:

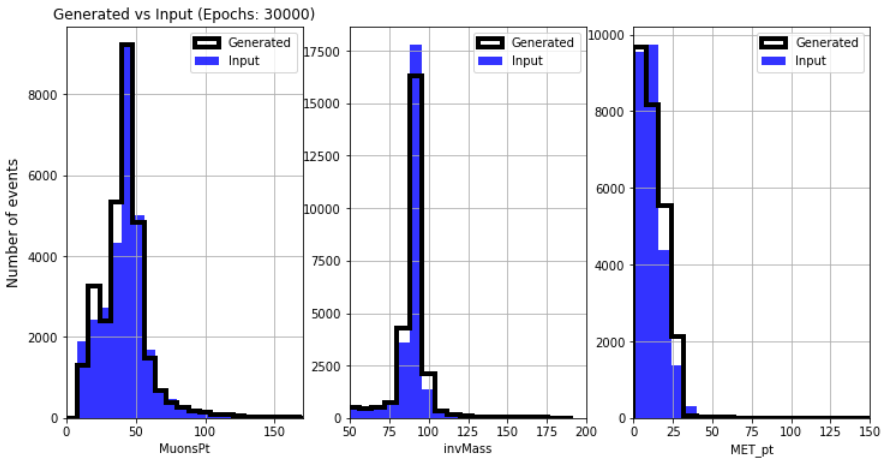


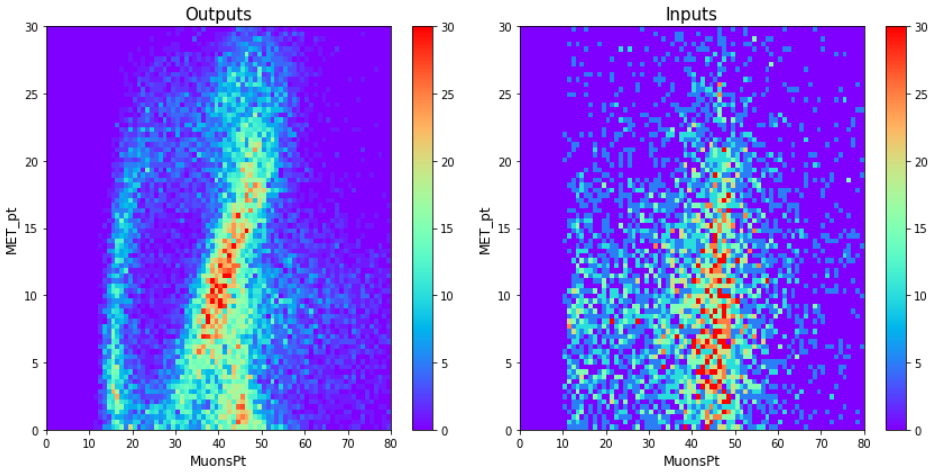
Figure 6.15: Output provided by a 3D cGAN.

In addition to these kind of visual tool, we also use the *Kolmogorov-Smirnov test* (KS) and the χ^2 *test* to assess how similar/different the two distributions evaluated are. For the KS, we obtain:

KS	**p-values**	**statistics**
MuonPt	0.135	0.3
invMass	0.02	0.4
MET_pt	0.135	0.3

Table 6.6: KS test between inputs and outputs for each observable.

For the correlations:



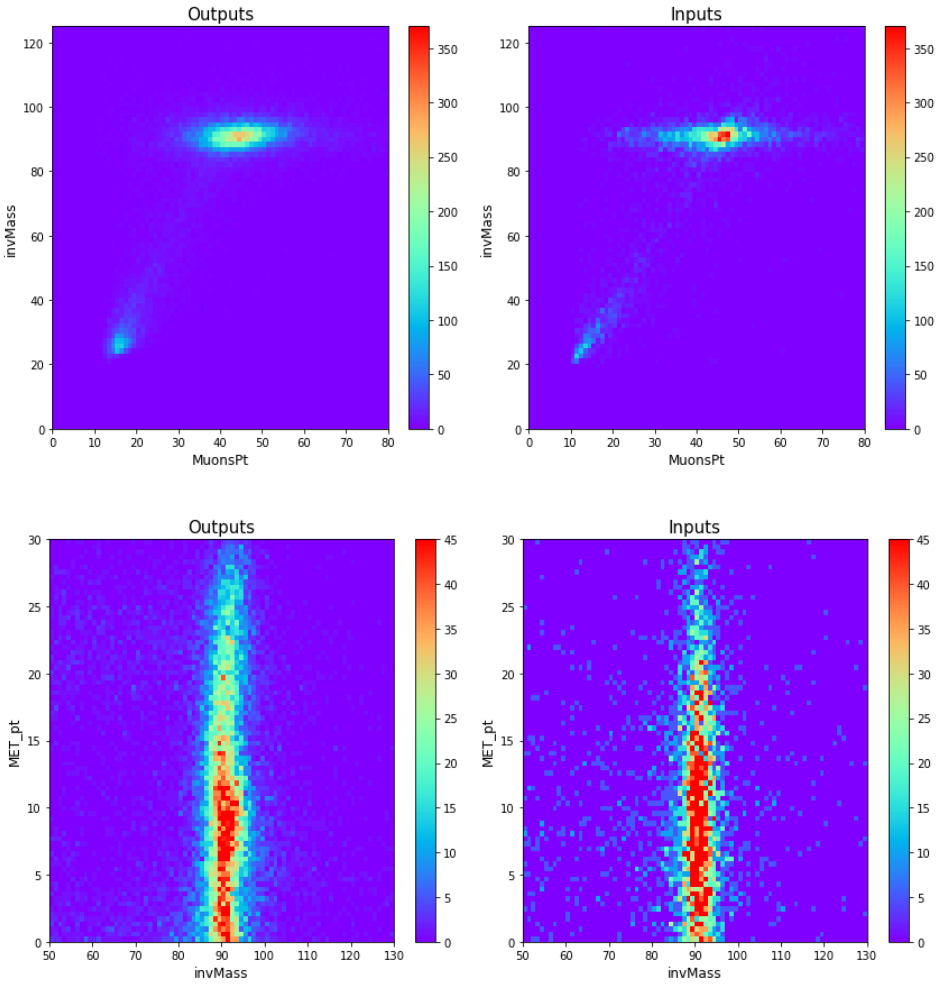


Figure 6.16: Comparison of the input and output correlations between the different variables used.

The mutual information score is similar to the previous example, see table [6.5].

6.12 CMS data

The final phase involves applying the network on real data rather than simulated samples from MC simulations. The dataset used is an nTuple compiled from data collected by CMS in 2022, consisting of around 600,000 events. All the leptons are muons.

Initial sample and data pre-processing

Here are the variables specified in the nTuple:

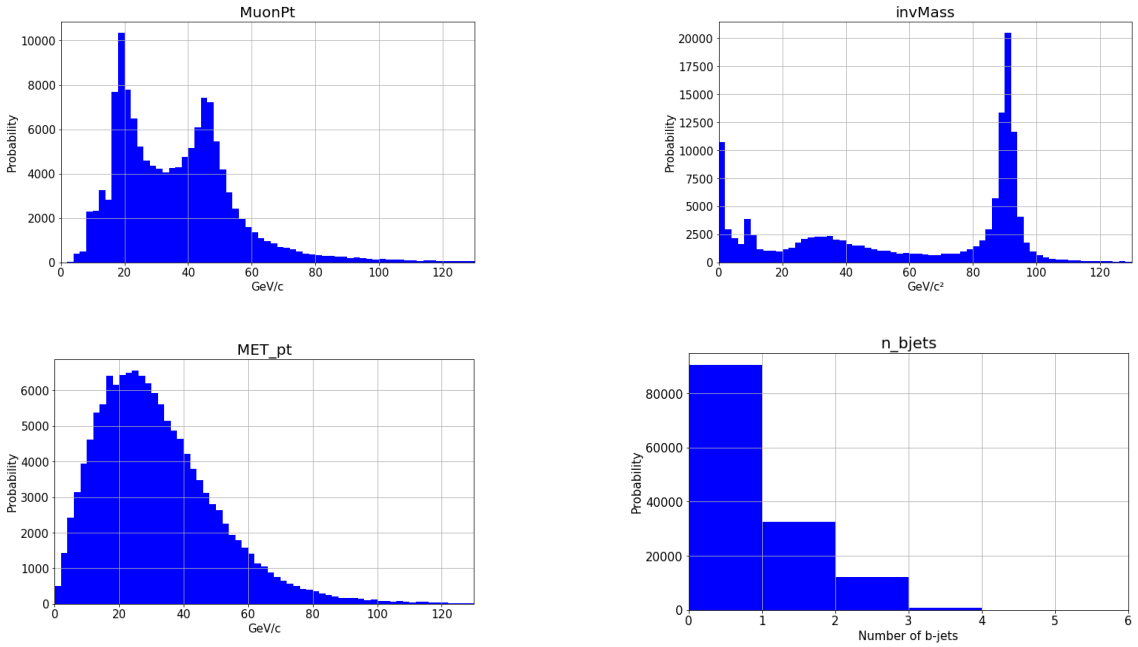


Figure 6.17: Generated variables in the CMS nTuple.

As one can observe, some of the previous distributions differ significantly from the MadGraph samples. These differences arise because the data originates from CMS, where many processes occur at the same time, with DY only being a part of it.
Therefore, the first step is to establish a strict cut on the leading muon transverse momentum, based on the single lepton trigger, and on the number of b-tagged jets. We set:

$$20(\text{GeV}) \leq p_T \ .$$

Afterwards, we want to isolate Drell-Yan from the other processes. To do so, a selection on the invariant mass is made:

$$70(\text{GeV}) \leq m_{ll} \leq 110(\text{GeV}) \ .$$

We then obtain the following distributions:

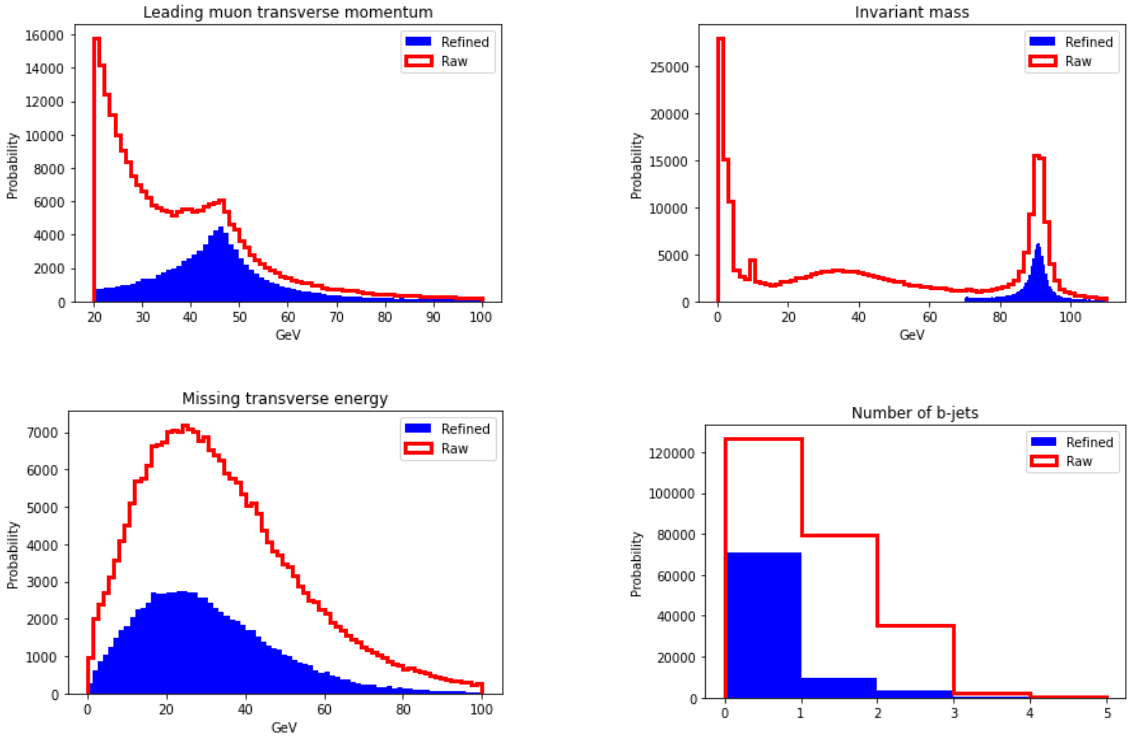


Figure 6.18: Generated variables in the CMS nTuple.

After these cuts, the resulting sample is made of approximately 84,000 events. With the following repartition of b-jets:

DY events	**Proportion**
No b-jets	84.3%
1 b-jet	11.2%
2 b-jets	4.4%
3 or more b-jets	0.1%

Table 6.7: Proportion of DY events with x b-jets. The last case is not considered in this work.

Results using CMS data

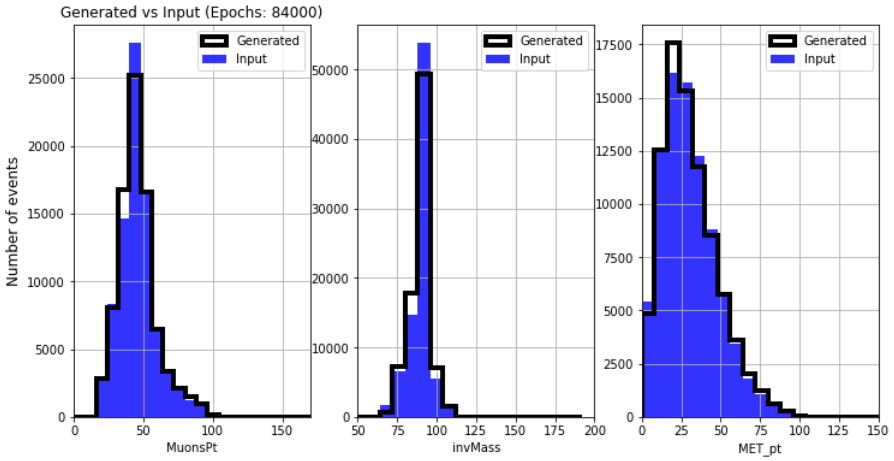


Figure 6.19: Output provided by a 3D cGAN using CMS data as training sample. See Appendix [9] for more detailed plots.

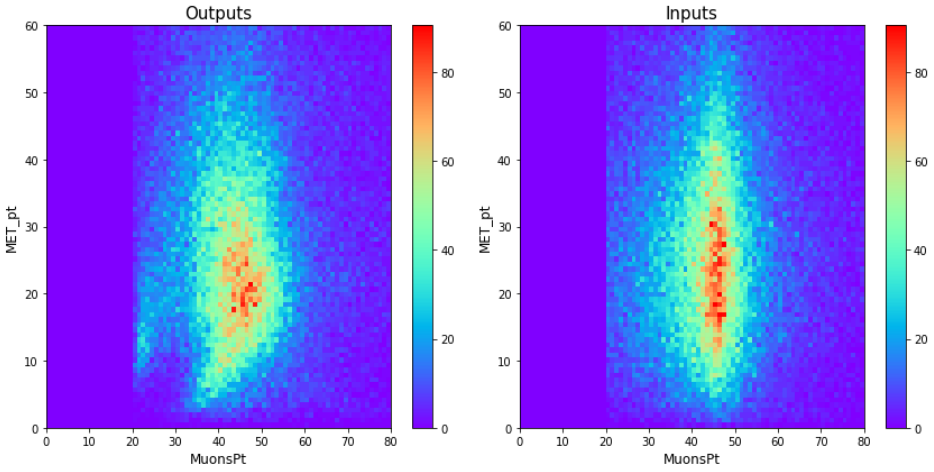
For the KS test, we obtain:

KS	p-values	statistics
MuonPt	0.007	0.433
invMass	0.007	0.433
MET_pt	0.239	0.267

Table 6.8: KS test between inputs and outputs for each observable.

These plots show encouraging results, despite imperfect generated samples.

As seen in Fig.(6.20), we can visualize the dependencies between the three selected variables for the generated data and the training sample. Although the zones for outputs are larger, the general behaviour of the dependencies remains similar.



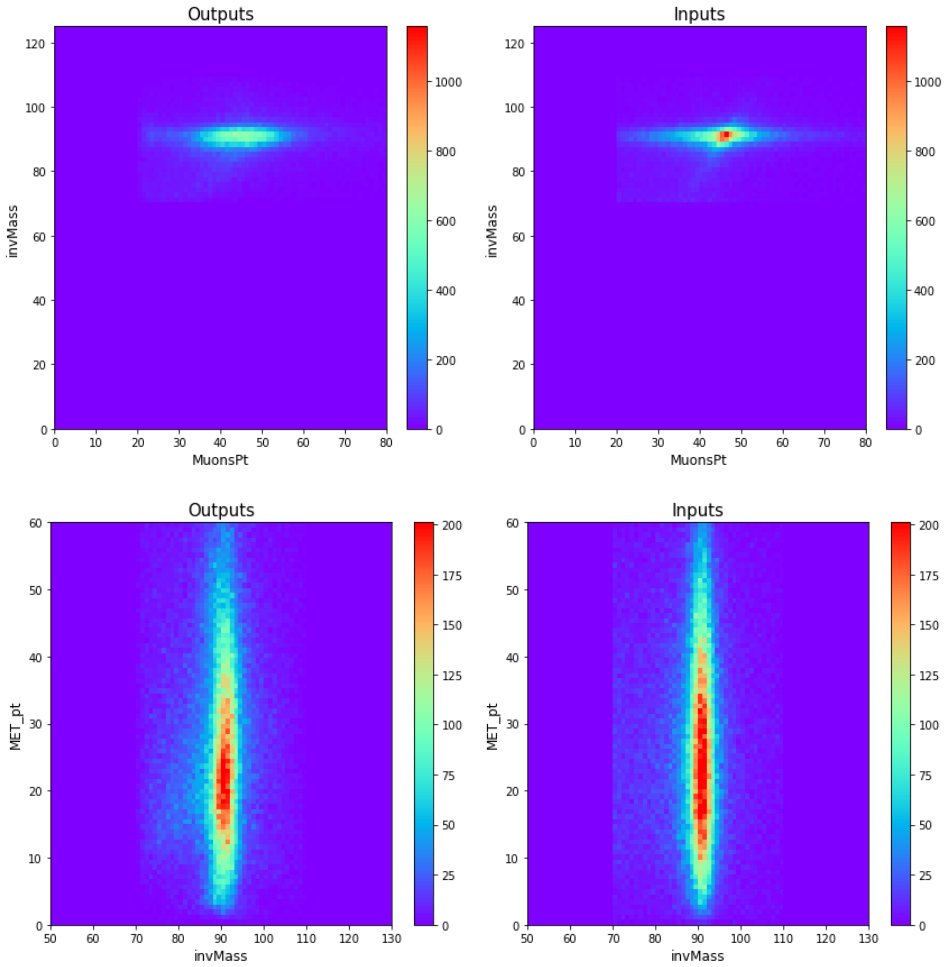


Figure 6.20: Comparison of the input and output correlations between the different variables used.

To get numerical values of the dependencies between observables, we can also compute the mutual information values for each pair of variables, the obtained values are:

Mutual Information score	**Outputs**	**Inputs**
MI(MuonPt, invMass)	11.3258	11.2855
MI(invMass, MET_pt)	11.3265	11.291
MI(MuonPt, MET_pt)	11.3306	11.3246

Table 6.9: Mutual Information Values for a 3D cGAN using CMS data as training sample.

An interesting observation can be made here. The mutual information scores for the inputs are very similar, conversely to previous runs with a GAN trained on the MC sample.

6.13 Signal and control regions

One of the expected task of the cGAN network is to be able to guess the behaviour of generated observables depending on the class assigned, in this case: control and

signal region.
In fact, there exist different ways to define these regions. First, we set the signal region as a set of events without any b-tagged jets, and the control region represents all the events with at least 1 b-tagged jet. Then, we set the signal region as a set of events with 0 or 1 b-tagged jet, and the control region represents all the events with at least 2 b-tagged jets.

Generated samples

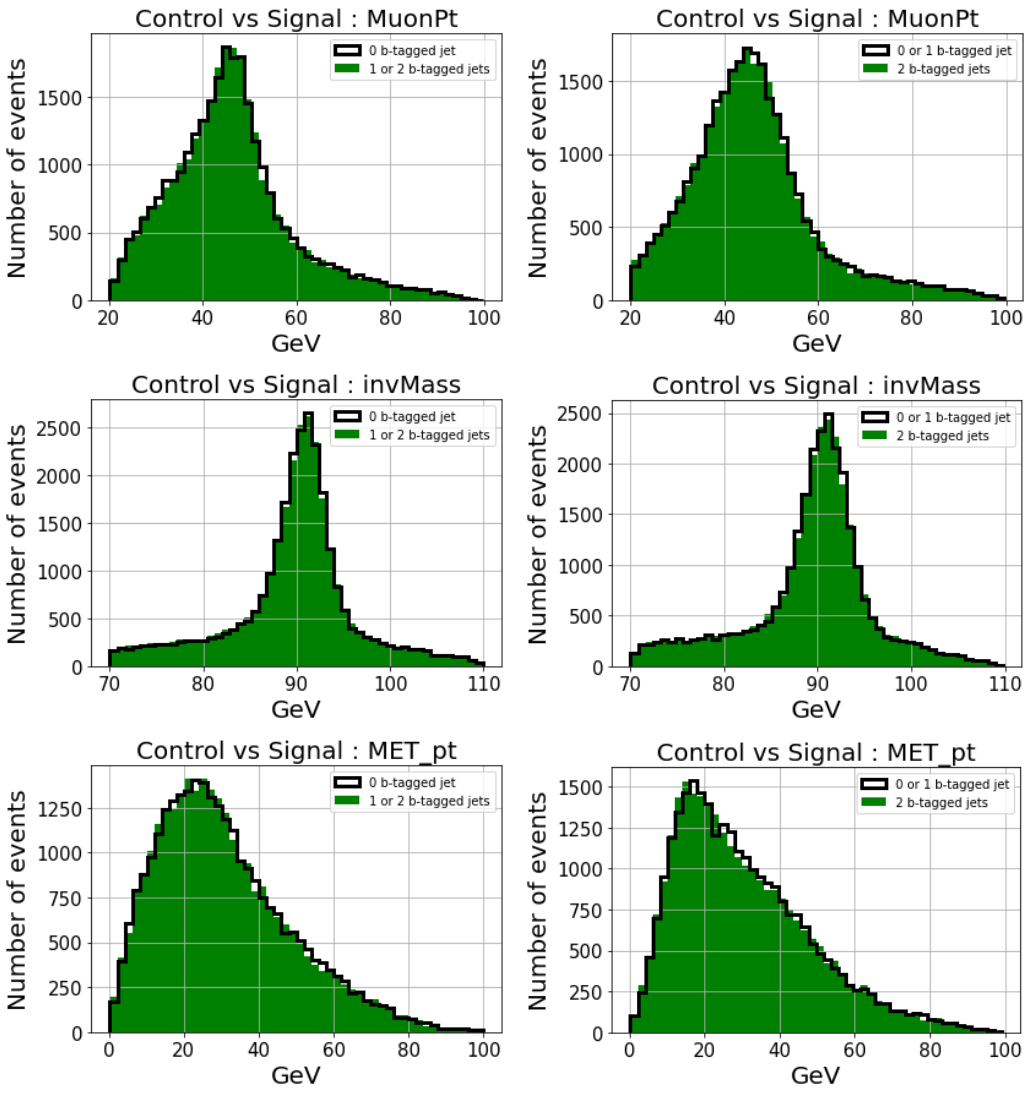


Figure 6.21: Comparison between control and signal region, for two different settings.

No significant deviation is observed between the signal (green distribution) and control events (black lines) for a same observable. However, when we compare the two definitions of SR/CR, the behaviour slightly differs.

CMS data

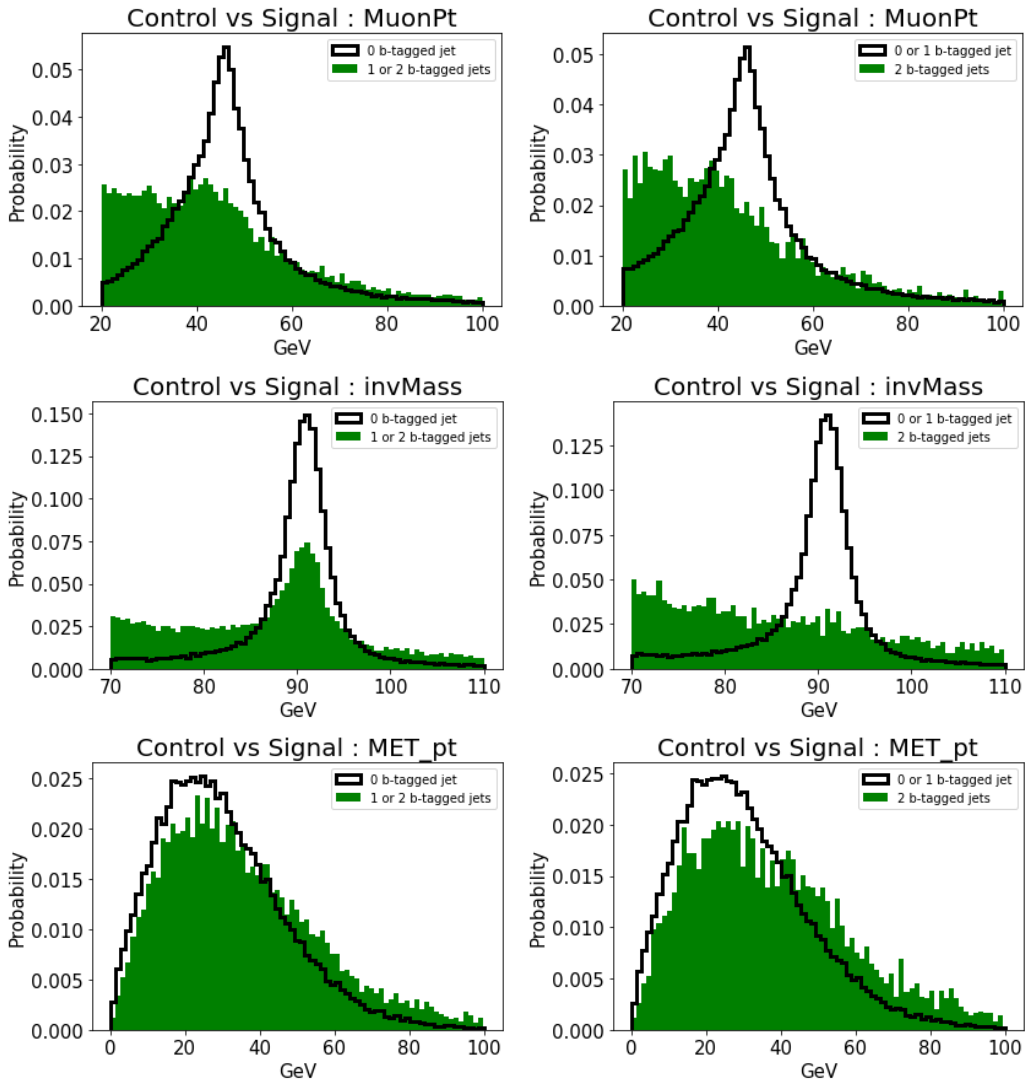


Figure 6.22: Comparison between control and signal region on CMS data, for two different settings.

Firstly, it is essential to note that the signal distributions have significantly fewer statistics compared to the control ones. This is even more pronounced when using the second definition for SR/CR, i.e., the second column of plots. This fact could potentially influence the results. This difference is absent in the previous distributions, using generated events.

The missing transverse energy distributions appear similar for both SR/CR definitions. The differences can likely be attributed to the lower number of events in the signal distributions.
However, for the transverse momentum and invariant mass, there is a notable difference between the two distributions, which cannot be solely explained by the number of events in each sample. In the left column, the shapes are somewhat reproduced, but in the right column, the events with two b-tagged jets seem entirely different from those with 0 or 1 b-jet. The network used in this work fails to capture this

shift in behavior, indicating it is missing crucial information.

To address this issue, another conditioning variable could be selected. The number of b-jets detected is a discrete variable, and the correlation between the three observables and this conditioning variable may be challenging for our network to understand. A straightforward alternative would be the invariant mass as a replacement. Additionally, it is possible to select multiple quantities as conditioning variable. Therefore, we could also choose other observables alongside the number of b-jets to achieve better results.

6.14 Quick discussion about the architectures used

First, the main problem encountered is the lack of stability of the network. After implementing an adaptive learning rate technique, the GAN eventually stabilizes. When transitioning to a cGAN, one could expect the network to gain robustness since an additional piece of information is provided to the network. However, this is not observed in this work. The conditional network seems to be less stable than the standard GAN. The modified version of the cGAN could cause this used here. Here are some of the specifications of the architecture used:

- 4 layers of 32 neurons each,
- latent vector of dimension 15,
- cyclic learning rate, with mode "triangle2"

Second, when the training is done on actual data, the cGAN seems way more stable than previously. This could be explained by the training sample being much larger in this case, thus providing more statistics to the network. On the other hand, the shapes of the distributions of the selected observables are different between the two training samples. When directly retrieving data from the LHC, the distributions are smoother and with broader peaks compared to the MC sample. Both of these influence the performance of the cGAN.
Here are some of the specifications of the architecture used:

- 6 layers of 64 neurons each,
- latent vector of dimension 70,
- cyclic learning rate, with mode "triangle"

Chapter 7

To go further

Throughout this report, we have discussed the applicability of generative adversarial networks (GANs) for background modeling. However, GANs are only one type of generative modeling algorithm among many. Utilizing other algorithms could be beneficial to this field of research. We will review the state of the art with an emphasis on *Normalizing Flows* (NFs) [75]. This specific architecture has already yielded interesting results [76], and physicists start to have some experience with this type of neural network.

Normalizing flows are a class of models used in machine learning for generative modelling. The main idea behind normalizing flows is to transform a simple probability distribution, such as a standard Gaussian distribution, into a more complex distribution that closely matches the true distribution.
Normalizing flows consist of a series of invertible transformations applied to a simple base distribution. These transformations are designed to gradually deform the base distribution in order to converge to the expected result. Each transformation in the flow must be invertible, meaning that you can easily compute both the forward and inverse transformations. By chaining together several of these transformations, normalizing flows can model complex data distributions with intricate patterns and dependencies. The final distribution obtained after applying all transformations is a complex, non-Gaussian distribution which should, as closely as possible, matches the data distribution.

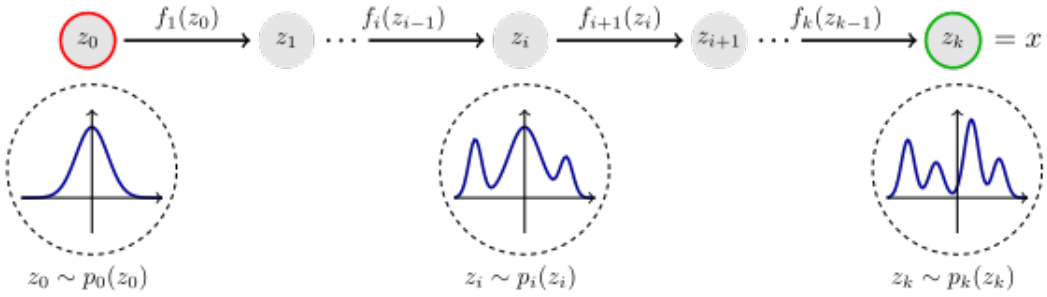


Figure 7.1: Operation of a normalizing flow

In comparison to other generative models, such as GANs and *variational autoencoders* (VAEs) [77] , the training process of NFs is much more stable, it is not

required to thoroughly fine-tune hyperparameters. Moreover, flow-based algorithms tend to converge faster than other models. However, very high-dimensional latent vectors are necessary, which is usually hard to interpret. Let's go through a short state of the art about this strategy.

Recently, several searches have monitored the efficiency of these generative models on similar tasks in order to compare them efficiently.
First, in 2021, an empirical comparison between GANs and NFs was made [78], on different datasets but, with the common point of being low-dimensional data. Both a standard GAN and a WCGAN [79] were used to draw this comparison. Surprisingly, the NF outperforms both type of GANs on several metrics, as kernel density estimation or the very metric on which the WCGAN is based on: the Wasserstein-1 distance. However, this analysis only holds to low-dimensional datasets. There is no certainty that this conclusion could be translated to high-dimensional ones.
Then, in 2022, a comparison between several generative modelling strategies, such as GANs, NFs and VAEs, was performed [80]. The GAN and its variations are presented as the best performing algorithms in terms of training and test speed, parameter efficiency, sample quality, sample diversity, and ability to scale to high resolution data. However, the gap between GANs and the latest version of other approaches was shrinking at the time the paper was published.
Eventually, a search about speech enhancement published in late 2023 [81] takes an interesting approach. Indeed, a new variation of GAN is derived: the SEFGAN. The main idea behind this variant is the use of a NF as generator for the GAN. This article shows that this hybrid approach clearly outperforms a pure NF or pure GAN strategy according to computational metrics and listening experiments done by human listeners.

In addition to the three models already discussed, it is important to mention the existence of *transformers* [82]. Initially designed for natural language processing and computer vision, the use of this architecture has been very recently extended to high energy physics and has provided promising results so far [83] [84]. Hence, transformers might represent an interesting alternative to GANs in background modelling. From a different point of view, a similar method to the SEFGAN can be imagined, i.e., using a transformer as the generator of a GAN. To my knowledge, no papers about such an architecture have been published as of June 2024, whether in high energy physics or other fields. This could be explained by the relative infancy of transformers.

Chapter 8

Conclusion

The goal of this work was the background modellisation of the Drell-Yan process using a completely different approach, generative adversarial networks. In this report, my aim was to briefly introduce the 2HDM, detail the different mechanisms of Higgs boson pair production and decay, and explain how these are relevant in the investigation of new physics, then, to mention the obstacles encountered in validating this theory with experimental evidence.

The results obtained with this new cGAN approach seem promising. Both the one-dimensional distributions and the correlation plots show very similar behavior in comparison to our target, despite some weaknesses, particularly in handling sharp peaks. To overcome the main weaknesses of GAN-like algorithms, various tools were employed to ensure the network converged to a solution.
Moreover, using actual CMS data as the training sample bypasses one of the most significant limitations: the high computational requirement. By adopting a purely data-driven approach, generating a Drell-Yan sample via MC simulation is no longer necessary. The network has also been tested following this approach, and the results are more encouraging than with the generated sample.

In my opinion, one of the most limiting factor for the cGAN is the number of variables generated. Increasing the network's dimension consistently led to a loss in accuracy. Although the effects were not significant for three variables, they could quickly become overwhelming for a larger number. While this work successfully handled three variables, training a 4D or higher cGAN would be beyond my current capabilities. The research team behind the paper this thesis is based on managed to handle five variables with convincing results. Additionally, the algorithm has shown shortcoming when guessing the observables in the signal region. The shift in behaviour was either partially reproduced or completely missed by the cGAN.

One potential improvement would be to replace the DNN used as the cGAN generator with a generative machine learning algorithm, such as a normalizing flow or a transformer. The combination of a normalizing flow and a GAN (SEFGAN) has shown promising results in fields other than particle physics, and transformers are already yielding encouraging results in high-energy physics.
Another conditioning variable could also enhance the network's performance by making the correlation between this variable and the generated observables more

apparent. This, in turn, would lead to more accurate predictions in the signal region.
I also strongly believe that more effective training and a better set of hyperparameters could easily solve this issue. The network training was purposefully kept short due to time constraints. For hyperparameters, a plausible improvement would be to create a hypergrid containing all parameters and evaluate these selections to choose the most adapted set of hyperparameters for the task. However, this would require significant computational power, which was not available to me.

Chapter 9

Appendix

Skills acquired during the project

During this thesis, I had to use an array of new concepts and tools. Here is a summary:

Software

- MadGraph and one of its model, the 2HDM, to generate the initial training sample
- ROOT , to open the different .root files used and to check their content
- Delphes, to simulate a (simplified) detector response, still for the initial sample
- Ubuntu, familiarisation with the OS

Packages

- Keras, for machine learning
- TensorFlow, for machine learning
- UpROOT, for manipulating .root files
- Scipy, for tools as mutual information and KS
- Awkward arrays, type of arrays used in the .root files
- Parquet, pandas-like objects used to process CMS data

Machine learning

- GAN, cGAN and "blinded" cGAN
- brief introduction to the concept of normalizing flows
- plenty of different tools to ease the convergence of a network: dynamic learning rate techniques, *He* initialization, gradient clipping, ...

Soft skills

- improve both active and passive English language skills
- more autonomous learning
- more efficient documentary research

Results plots

The plots shown in this appendix are the exact same results than presented previously, only the binning differs.
For the 3D cGAN trained on CMS data, we have the following generated observables:

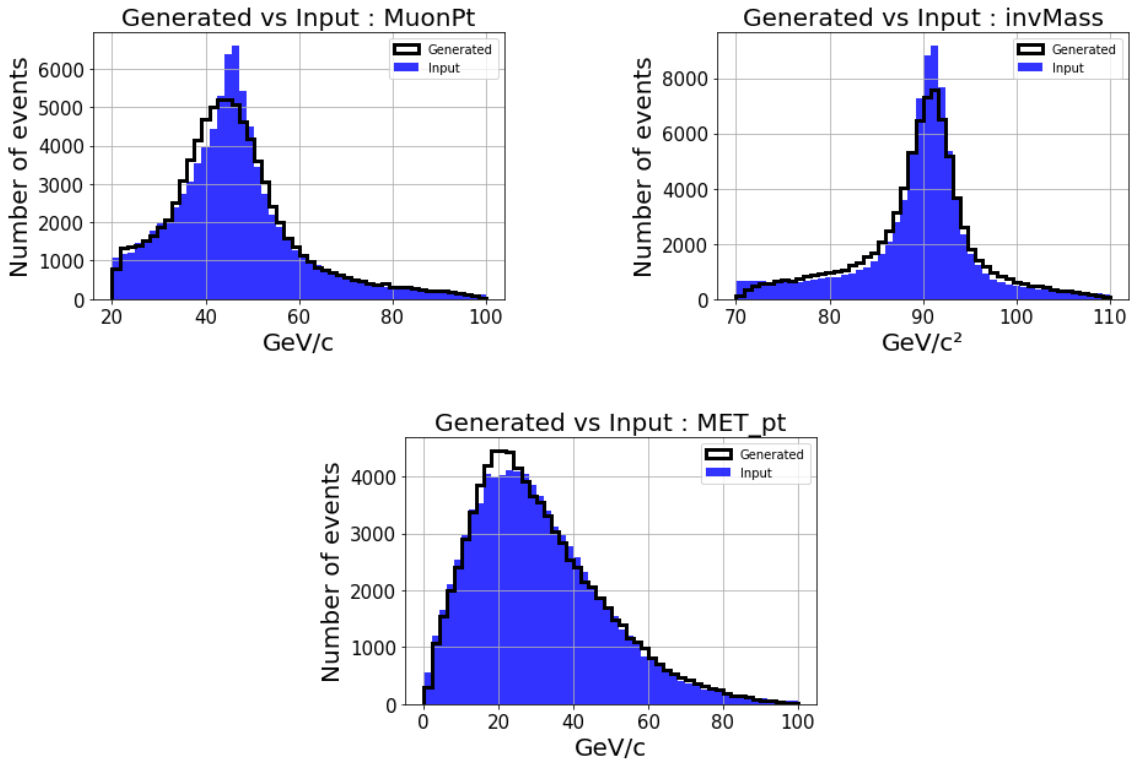


Figure 9.1: Output provided by a 3D cGAN using CMS data as training sample.

Chapter 10

Bibliography

1. Bury, F. *Application of deep learning techniques in CMS data analysis*. Master's thesis, Université catholique de Louvain, Faculté des sciences, École de physique, 2018.

2. Chisholm, A., et al. Non-Parametric Data-Driven Background Modelling using Conditional Probabilities . Journal of High Energy Physics, vol. 2022, no 10, October 2022, p. 1. arXiv.org, https://doi.org/10.1007/JHEP10(2022)001.

3. Maltoni, F. Standard Model and Beyond. Master's course, Université Catholique de Louvain, 2023.

4. Lee, J.; Verleysen, M. Machine Learning: Regression, Deep Networks, and Dimensionality Reduction. Master's course, Université Catholique de Louvain, 2023.

5. Degrande, C.; Lemaître, V.; Delaere, C. Fundamental Interactions and Elementary Particles. Master's course, Université Catholique de Louvain, 2022.

6. Lemaître, V.; Cortina Gil, E. Particle Accelerators and Neutrino Physics. Master's course, Université Catholique de Louvain, 2023.

7. Degée, A. Higgs mechanism in the general Two-Higgs-Doublet Model. Master's thesis, Université de Liège, Faculté des Sciences, Sciences Physiques, 2009. Retrieved from https://orbi.uliege.be/bitstream/2268/68445/1/MmemoireFinal.pdf

8. Amsler, C., et al. Review of Particle Physics. Physics Letters B, vol. 667, no 1-5, September 2008, p. 1-6. DOI.org (Crossref), https://doi.org/10.1016/j.physletb.2008.07.018.

9. InspireHEP. Two-Higgs doublet model and the LHC. Retrieved April 14, 2024, from https://inspirehep.net/files/6dec6d5b7461dfa2693c895eafc4f711

10. Dvorkin, C., et al. Neutrino Mass from Cosmology: Probing Physics Beyond the Standard Model. arXiv:1903.03689, arXiv, March 2019. arXiv.org, https://doi.org/10.48550/arXiv.1903.03689.

11. Sakharov, A. Violation of CP in Variance, C Asymmetry, and Baryon Asymmetry of the Universe . Physics-Uspekhi, vol. 34, no 5, May 1991, p. 392-93, https://ufn.ru/en/articles/1991/5/h/.

12. Garrett, K.. Dark matter: A primer. Advances in Astronomy. 2011 (968283): 1–22. arXiv:1006.2483

13. Biermann, P., et Faustin Munyaneza. The Nature of Dark Matter. AIP Conference Proceedings, vol. 972, 2008, p. 365-73. arXiv.org, https://doi.org/10.1063/1.2870344.

14. Frost, J. Dark Matter Searches at the LHC. 2022. CERN Document Server, https://cds.cern.ch/record/2843045

15. Einasto, J. Dark Matter. arXiv:0901.0632, arXiv, October 2010. arXiv.org, http://arxiv.org/abs/0901.0632.

16. Smith, E. The Hierarchy Problem. The University of Chicago, QFT III Final Paper, 2019, https://homes.psd.uchicago.edu/~sethi/Teaching/P445-S2019/Emily_Smith_QFT_III_Final_Paper.pdf

17. Tanabashi, M., et al. Review of Particle Physics. Physical Review D, vol. 98, August 2018, p. 030001. NASA ADS, https://doi.org/10.1103/PhysRevD.98.030001.

18. CMS mesure la masse du Higgs avec une précision inédite. CERN, April 2024, Here is a long URL: https://home.cern/fr/news/news/physics/cms-measures-higgs-bosons-mass-unprecedented-precision

19. CMS Collaboration. Search for flavor-changing neutral current interactions of the top quark and Higgs boson in final states with two photons in proton-proton collisions at $\sqrt{s} = 13$ TeV. Physical Review Letters, vol. 129, no 3, July 2022, p. 032001. arXiv.org, https://doi.org/10.1103/PhysRevLett.129.032001.

20. Branco, G. C., et al. Theory and phenomenology of two-Higgs-doublet models . Physics Reports, vol. 516, no 1-2, July 2012, p. 1-102. arXiv.org, https://doi.org/10.1016/j.physrep.2012.02.002.

21. Arhrib, A., et al. Two-Higgs-Doublet type-II and -III models and $t \to ch$ at the LHC . The European Physical Journal C, vol. 76, no 6, June 2016, p. 328. arXiv.org, https://doi.org/10.1140/epjc/s10052-016-4167-9.

22. Branco, G. C., et al. Theory and phenomenology of two-Higgs-doublet models . Physics Reports, vol. 516, no 1-2, July 2012, p. 1-102. arXiv.org, https://arxiv.org/pdf/1106.0034.pdf

23. Wikipedia contributors. Supersymmetry. In Wikipedia. Retrieved April 14, 2024, from https://en.wikipedia.org/wiki/Supersymmetry.

24. Csaki, C. The Minimal Supersymmetric Standard Model (MSSM). Modern Physics Letters A, vol. 11, no 08, March 1996, p. 599-613. arXiv.org, https://doi.org/10.1142/S021773239600062X.

25. Spira, M. Effective Multi-Higgs Couplings to Gluons. Journal of High Energy Physics, vol. 2016, no 10, October 2016, p. 26. arXiv.org, https://doi.org/10.1007/JHEP10(2016)026.

26. ATLAS Collaboration. Search for Elusive "Di-Higgs Production" Reaches New Milestone. June 2024, https://atlas.cern/updates/briefing/new-milestone-di-Higgs-search.

27. Tumasyan, A., et al. A Portrait of the Higgs Boson by the CMS Experiment Ten Years after the Discovery. Nature, vol. 607, no 7917, July 2022, p. 60-68. www.nature.com, https://doi.org/10.1038/s41586-022-04892-x.

28. HPQCD and UKQCD Collaborations, et al. High-Precision Lattice QCD Confronts Experiment. Physical Review Letters, vol. 92, no 2, January 2004, p. 022001. APS, https://doi.org/10.1103/PhysRevLett.92.022001.

29. Wikipedia contributors. Neural network (machine learning). In Wikipedia. Retrieved April 14, 2024, from https://en.wikipedia.org/wiki/Neural_network_(machine_learning).

30. Baeldung. Neural Networks and Neurons in Java. Retrieved April 14, 2024, from https://www.baeldung.com/cs/neural-networks-neurons.

31. Bisong, E. Optimization for Machine Learning: Gradient Descent. In Optimization for Machine Learning (pp. 281-303). Retrieved April 14, 2024, from https://link.springer.com/chapter/10.1007/978-1-4842-4470-8_16.

32. « Difference between Parametric and Non-Parametric Methods ». GeeksforGeeks, February 2020, https://www.geeksforgeeks.org/difference-between-parametric-and-non-parametric-methods/.

33. Introduction – Background estimation with the ABCD method. https://cms-opendata-workshop.github.io/workshop-lesson-abcd-method/01-introduction/index.html. Retrieved April 14 2024.

34. CERN. ABCD Method Guide. October 2018. Retrieved from https://twiki.cern.ch/twiki/pub/Main/ABCDMethod/ABCDGuide_draft18Oct18.pdf.

35. CMS Collaboration. Search for HH production in the bbW+W- decay mode in proton-proton collisions at s = 13 TeV. 2024

36. Rummukainen, K. Monte Carlo simulations in physics. Department of Physical Sciences, University of Oulu, https://www.mv.helsinki.fi/home/rummukai/lectures/montecarlo_oulu/lectures/mc_notes1.pdf.

37. Kogler, R., et al. Jet Substructure at the Large Hadron Collider: Experimental Review. September 2019. Retrieved from https://cds.cern.ch/record/2641634/files/1803.06991.pdf

38. Schwartz, M. D. Modern Machine Learning and Particle Physics. Harvard Data Science Review, vol. 3, no 2, April 2021. hdsr.mitpress.mit.edu, https://doi.org/10.1162/99608f92.beeb1183.

39. Lehmacher, M. b-Tagging Algorithms and their Performance at ATLAS. arXiv:0809.4896, arXiv, November 2008. arXiv.org, https://doi.org/10.48550/arXiv.0809.4896.

40. Verkerke, W. Introduction to Morphing. Nikhef, https://indico.cern.ch/event/507948/contributions/2028505/attachments/1262169/1866169/atlas-hcomb-morphwshop-intro-v1.pdf.

41. ATLAS Collaboration. Search for Higgs boson decays into a Z boson and a light hadronically decaying resonance using 13 TeV pp collision data from the ATLAS detector. Physical Review Letters, vol. 125, no 22, November 2020, p. 221802. arXiv.org, https://doi.org/10.1103/PhysRevLett.125.221802.

42. Homepage of the POWHEG BOX. https://powhegbox.mib.infn.it/. Retrieved April 14, 2024.

43. Ioffe, S., Szegedy C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. arXiv:1502.03167, arXiv, March 2015. arXiv.org, http://arxiv.org/abs/1502.03167.

44. Goodfellow, I. J., et al. Generative Adversarial Networks. arXiv:1406.2661, arXiv, June 2014. arXiv.org, https://arxiv.org/abs/1406.2661

45. THE MNIST DATABASE of handwritten digits. Yann LeCun, Courant Institute, NYU Corinna Cortes, Google Labs, New York Christopher J.C. Burges, Microsoft Research, Redmond.

46. CIFAR-10 and CIFAR-100 datasets. https://www.cs.toronto.edu/~kriz/cifar.html. Retrieved April 14, 2024.

47. Eckerli, F., Osterrieder J. Generative Adversarial Networks in finance: an overview. arXiv:2106.06364, arXiv, July 2021. arXiv.org, https://arxiv.org/pdf/2106.06364.pdf

48. Mirza, M., Osindero S. Conditional Generative Adversarial Nets. arXiv:1411.1784, arXiv, November 2014. arXiv.org, https://arxiv.org/abs/1411.1784.

49. Xu, J., Li, Z., Du, B., Zhang, M., Liu, J. (n.d.). Reluplex made more practical: Leaky ReLU. School of Software Engineering, Tongji University, Shanghai, China; Department of Computer Science, University of Warwick, Coventry, United Kingdom; School of Software Engineering, East China Normal University, Shanghai, China, https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9219587

50. Hendrycks, D., Gimpel K. Gaussian Error Linear Units (GELUs). arXiv:1606.08415, arXiv, June 2023. arXiv.org, http://arxiv.org/abs/1606.08415.

51. Clevert, D., et al. Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs). arXiv:1511.07289, arXiv, February 2016. arXiv.org, http://arxiv.org/abs/1511.07289.

52. Klambauer, G., et al. Self-Normalizing Neural Networks. arXiv:1706.02515, arXiv, September 2017. arXiv.org, http://arxiv.org/abs/1706.02515.

53. Epochs, Batch Size, Iterations - How They Are Important. https://www.sabrepc.com/blog/Deep-Learning-and-AI/Epochs-Batch-Size-Iterations. Retrieved April 14, 2024.

54. Ruby, U., Yendapalli, V. (2020). Binary cross entropy with deep learning technique for image classification. International Journal of Advanced Trends in Computer Science and Engineering, 9(4). DOI: 10.30534/ijatcse/2020/175942020.

55. Saxena, S. Binary Cross Entropy/Log Loss for Binary Classification. Analytics Vidhya, March 2021, https://www.analyticsvidhya.com/blog/2021/03/binary-cross-entropy-log-loss-for-binary-classification/.

56. Amari, S. Backpropagation and stochastic gradient descent method. Neurocomputing, vol. 5, no 4, June 1993, p. 185-96. ScienceDirect, https://doi.org/10.1016/0925-2312(93)90006-O.

57. Kingma, D. P., Ba J. Adam: A Method for Stochastic Optimization. arXiv:1412.6980, arXiv, January 2017. arXiv.org, https://arxiv.org/abs/1412.6980

58. Dauphin, Y., et al. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. arXiv:1406.2572, arXiv, June 2014. arXiv.org, https://arxiv.org/pdf/1406.2572.pdf.

59. Xie, X., et al. Adan: Adaptive Nesterov Momentum Algorithm for Faster Optimizing Deep Models. arXiv:2208.06677, arXiv, February 2023. arXiv.org, https://arxiv.org/pdf/2208.06677.pdf

60. He, Kaiming, et al. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. arXiv:1502.01852, arXiv, February 2015. arXiv.org, https://arxiv.org/abs/1502.01852

61. Pascanu, R., et al. On the difficulty of training Recurrent Neural Networks. arXiv:1211.5063, arXiv, February 2013. arXiv.org, https://arxiv.org/abs/1211.5063

62. Ioffe, S., Szegedy C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. arXiv:1502.03167, arXiv, March 2015. arXiv.org, http://arxiv.org/abs/1502.03167.

63. Keras Documentation. LearningRateScheduler. Retrieved April 14, 2024, from https://keras.io/api/callbacks/learning_rate_scheduler/.

64. Keras Documentation. ReduceLROnPlateau. Retrieved April 14, 2024, from https://keras.io/api/callbacks/reduce_lr_on_plateau/.

65. TensorFlow Documentation. Retrieved April 14, 2024, from https://www.tensorflow.org/.

66. Keras Documentation. Retrieved April 14, 2024, from https://keras.io/.

67. PyPI. (n.d.). uproot. Retrieved April 14, 2024, from https://pypi.org/project/uproot/.

68. Wang, Z., et al. Generative Adversarial Networks in Computer Vision: A Survey and Taxonomy. arXiv:1906.01529, arXiv, December 2020. arXiv.org, https://arxiv.org/pdf/1906.01529.pdf.

69. Salimans, T., et al. Improved Techniques for Training GANs. arXiv:1606.03498, arXiv, June 2016. arXiv.org, https://arxiv.org/abs/1606.03498.

70. Wang, Z., et al. Generative Adversarial Networks in Computer Vision: A Survey and Taxonomy. arXiv:1906.01529, arXiv, December 2020. arXiv.org, https://arxiv.org/pdf/1906.01529.pdf.

71. MadGraph Development Team. Particle Content. Retrieved April 14, 2024, from http://madgraph.phys.ucl.ac.be/particles.html.

72. B. N. S. Reenu. Python for Microscopists. GitHub. Retrieved April 14, 2024, from https://github.com/bnsreenu/python_for_microscopists.

73. Massey, F. J. The Kolmogorov-Smirnov Test for Goodness of Fit. Journal of the American Statistical Association, vol. 46, no 253, 1951, p. 68-78. JSTOR, https://doi.org/10.2307/2280095.

74. Batina, L., Gierlichs, B., Prouff, E., Rivain, M., Standaert, F.-X., & Veyrat-Charvillon, N. (2011). Mutual Information Analysis: A Comprehensive Study. *Journal of Cryptology*, 24, 269-291. https://doi.org/10.1007/s00145-010-9084-8.

75. Kobyzev, I., et al. Normalizing Flows: An Introduction and Review of Current Methods. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 43, no 11, November 2021, p. 3964-79. arXiv.org, https://doi.org/10.1109/TPAMI.2020.2992934.

76. Verheyen, R. Event Generation and Density Estimation with Surjective Normalizing Flows. SciPost Physics, vol. 13, no 3, September 2022, p. 047. www.scipost.org, https://doi.org/10.21468/SciPostPhys.13.3.047.

77. Pinheiro Cinelli, L., et al. Variational Autoencoder. Variational Methods for Machine Learning with Applications to Deep Networks. 2021. Springer. pp. 111–149. doi:10.1007/978-3-030-70679-1_5

78. Liu, T., Regier J. An Empirical Comparison of GANs and Normalizing Flows for Density Estimation. arXiv:2006.10175, arXiv, December 2021. arXiv.org, https://doi.org/10.48550/arXiv.2006.10175.

79. Arjovsky, M., et al. Wasserstein GAN. arXiv:1701.07875, arXiv, December 2017. arXiv.org, https://arxiv.org/abs/1701.07875.

80. Bond-Taylor, S., et al. Deep Generative Modelling: A Comparative Review of VAEs, GANs, Normalizing Flows, Energy-Based and Autoregressive Models. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 44, no 11, November 2022, p. 7327-47. arXiv.org, https://doi.org/10.1109/TPAMI.2021.3116668.

81. Strauss, M., et al. SEFGAN: Harvesting the Power of Normalizing Flows and GANs for Efficient High-Quality Speech Enhancement. 2023 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA), 2023, p. 1-5. arXiv.org, https://doi.org/10.1109/WASPAA58266.2023.10248144.

82. Vaswani, A., et al. Attention Is All You Need. 7, arXiv:1706.03762, arXiv, August 2023. arXiv.org, https://doi.org/10.48550/arXiv.1706.03762.

83. Jiang, Z., et al. Ultra Fast Transformers on FPGAs for Particle Physics Experiments. arXiv:2402.01047, arXiv, February 2024. arXiv.org, https://doi.org/10.48550/arXiv.2402.01047.

84. Kim, J. Training toward significance with the decorrelated event classifier transformer neural network. Physical Review D, vol. 109, no 9, May 2024, p. 096035. APS, https://doi.org/10.1103/PhysRevD.109.096035.

UNIVERSITE CATHOLIQUE DE LOUVAIN

Faculté des sciences

Place des sciences, 2 bte L6.06.01, 1348 Louvain-la-Neuve, Belgique | **www.uclouvain.be/sc**