arXiv:2305.13781v2 [hep-ph] 31 May 2023

Sharpening the $A \to Z^{(*)}h$ Signature of the Type-II 2HDM at the LHC through Advanced Machine Learning

W. Esmail[a] , A. Hammad[b] and S. Moretti[c,d]

[a] GSI Helmholtzzentrum für Schwerionenforschung GmbH, 64291 Darmstadt, Germany.
[b] Institute of Convergence Fundamental Studies, Seoul National University of Science and Technology, 232 Gongneung-ro, Nowon-gu, Seoul, 01811, Korea.
[c] School of Physics and Astronomy, University of Southampton, Highfield, Southampton, UK.
[d] Department of Physics & Astronomy, Uppsala University, Box 516, SE-751 20 Uppsala, Sweden.

Abstract

The $A \to Z^{(*)}h$ decay signature has been highlighted as possibly being the first testable probe of the Standard Model (SM) Higgs boson discovered in 2012 (h) interacting with Higgs companion states, such as those existing in a 2-Higgs Doublet Model (2HDM), chiefly, a CP-odd one (A). The production mechanism of the latter at the Large Hadron Collider (LHC) takes place via $b\bar{b}$-annihilation and/or gg-fusion, depending on the 2HDM parameters, in turn dictated by the Yukawa structure of this Beyond the SM (BSM) scenario. Among the possible incarnations of the 2HDM, we test here the so-called Type-II, for a twofold reason. On the one hand, it intriguingly offers two very distinct parameter regions compliant with the SM-like Higgs measurements, i.e., where the so-called 'SM limit' of the 2HDM can be achieved. On the other hand, in both configurations, the AZh coupling is generally small, hence the signal is strongly polluted by backgrounds, so that the exploitation of Machine Learning (ML) techniques becomes extremely useful. Ours approach in this respect is a three-prong one. Firstly, we adjust ML models to analyze all possible High Energy Physics (HEP) data types, so as to maximize the amount of input information. Secondly, unlike most 'black-box' ML approaches currently in use in the HEP community, we exploit a (linear) Centered Kernel Alignment (CKA) similarity metric to analyze the learned representations in the hidden layers, thereby enabling an interpretative element of our results. Thirdly, we emphasise that the proposed ML models are generic and can thus be adopted in other physics problems. Concerning the one at hand, by using such advanced ML implementations, we ultimately show that the sensitivity of LHC searches in the $l^+l^-b\bar{b}$ ($l = e, \mu$) final state can significantly be improved with respect to traditional cut-and-count analyses and/or simpler ML algorithms. This is true for all distinctive kinematical configurations involving the $A \to Z^{(*)}h$ decay, i.e., below threshold ($m_A < m_Z + m_h$), at its maximum ($m_Z + m_h < m_A < 2m_t$) and near the onset of $t\bar{t}$ pair production ($m_A \approx 2m_t$), for which we propose Benchmark Points (BPs) for future phenomenological analyses.

Contents

1 Introduction **2**

2 The 2HDM **4**
2.1 The Higgs potential 4
2.2 Constraining the 2HDM free parameters 6
2.3 Assessment of the parameter space 6

3 Analysis strategy **7**
3.1 Current ATLAS and CMS results 8
3.2 Kinematical analysis 8

4 DNN **9**
4.1 MLP 11
4.2 CNN 15
4.3 SNN 17
4.4 HDNN 18
4.5 GNN 19
4.5.1 GCN 20
4.5.2 DGCNN 21
4.5.3 GraphSAGE 22
4.5.4 GAT 24

5 Similarity of DNN hidden layers representations **25**

6 Results **27**

7 Conclusions **29**

1 Introduction

The Higgs boson discovered at the LHC in 2012 by both the ATLAS and CMS collaborations [1, 2] is very much consistent with the one embedded in the SM, when it comes to the innumerable measurements performed of its properties (i.e., mass, Yukawa and gauge couplings, spin, CP quantum numbers). Yet, the so-called SM limit, whereby a Higgs boson state of an enlarged Higgs sector can play the role of the SM one, exists in a variety of BSM scenarios.

Amongst the latter, we concentrate here on the 2HDM [3], being the simplest BSM realization of the Higgs mechanism of Electro-Weak Symmetry Breaking (EWSB) employing the only Higgs field multiplet structure revealed by Nature so far, i.e., the doublet one. Such a BSM scenario is rather varied in its Higgs sector as, after EWSB, it contains five physical Higgs states. These are the A (massive, neutral and CP-odd), the $H^\pm$ (massive, charged and with mixed CP) states alongside two massive neutral CP-even ones, h and H (with, conventionally, $m_H > m_h$). Either of the latter two can be the aforementioned SM-like

Higgs state with a mass of 125 GeV or so, i.e., the h (in which case one speaks of a 'normal mass hierarchy' scenario) or the H (in which case one speaks of an 'inverted mass hierarchy' scenario). Herein, we assume the first configuration, such that $m_h = 125$ GeV (with all h couplings being SM-like). A 2HDM is phenomenologically appealing also for another reason, it can easily comply with the strong limits from EW Precision Observables (EWPOs), as it suffices to set the $H^\pm$ mass somewhat degenerate with those of the A and/or h/H states. Finally, the 2HDM can dispense of large Flavour Changing Neutral Currents (FCNCs) by simply invoking a $\mathbb{Z}_2$ symmetry between the two Higgs doublet fields, which can prevent these from occurring at tree-level. In turn, this implies well-defined Yukawa structures (which we will describe in detail below), depending on how the two Higgs doublets couple to fermions, which go under the name of Type-I, -II, lepton-specific and flipped [3].

Amongst all these, we concentrate here on the Type-II case. We do so as this realization of the 2HDM is the most challenging one phenomenologically. In fact, it implies a lower bound on the charged Higgs mass around 600 GeV, as per constraints coming from $b \to s\gamma$ transitions [4]. As mentioned, the EWPOs then require also the A and/or H states to be rather heavy. In fact, in the 2HDM Type-II, two different regions over the $(\cos(\beta - \alpha), \tan\beta)$ plane[1] can realize the aforementioned SM-like configuration (see, e.g., Refs. [5–8]). According to the analysis of Refs. [9, 10] (see also Ref. [6]), in the first one, the so-called 'alignment limit', whereby $\cos(\beta-\alpha) \to 0$ (and the couplings of the h state to u- and d-type quarks have the same sign as those in the SM) the CP-odd Higgs state is required to be rather heavy ($m_A \geq 350$ GeV) while, in the second one, the so-called 'wrong-sign scenario', whereby $\cos(\beta-\alpha)$ can reach 0.4 or so (and the couplings of the h state to $u(d)$-type quarks have the same(opposite) sign as(to) those in the SM) the A mass can be as light as 200 GeV or so.

Thus, the 2HDM Type-II offers the possibility to LHC searches of establishing sensitivity to the presence of the A state over a wide mass range. The latter is most copiously produced via $b\bar{b}, gg \to A$ and can, in particular, decay via $A \to Z^{(*)}h$, which can then altogether be elevated to a new A search channel, alongside the traditional ones in $\tau^+\tau^-$ (for $m_A < 2m_t$) and $t\bar{t}$ (for $m_A > 2m_t$) final states, since the h mass is now known rather precisely. The importance of the $A \to Z^{(*)}h$ decay channel has been repeatedly emphasised in literature, as it would simultaneously allow one to establish the presence of an extended Higgs sector as well as the gauge structure of the theory embedding it.

There are several searches that have been carried out at the LHC looking for the A state via the $A \to Zh$ decay, i.e., with the Z on-shell (which we will describe in a forthcoming section), typically done by using $Z \to l^+l^-$ ($l = e, \mu$) and $h \to b\bar{b}$ decays. However, these all concentrated on an A mass range starting from $m_Z + m_h \approx 215$ GeV, i.e., assuming decays of the CP-odd Higgs boson into Z and h particles both being on-shell. While this assumption is fully justified in the case of the Higgs boson, which has a typical width of order 10 MeV at most (according to latest h measurements at the LHC), it is less so for the gauge boson, for which the width-to-mass ratio is of order 3%. Off-shell effects involving the Z boson are therefore not negligible, hence searching for the CP-odd Higgs boson decaying into Z^*h is of phenomenological interest, as recently advocated in, e.g., [11, 12].

It is the purpose of our paper the one of proposing new searches for the $b\bar{b}, gg \to A \to Z^{(*)}h \to l^+l^-b\bar{b}$ channel at the LHC over an extended m_A range, from values both below m_A+m_Z (where the neutral weak gauge boson is off-shell, Z^*) and (far) above it (where the neutral weak gauge boson is on-shell, Z). Furthermore, in the light of the fact that the AZh vertex is suppressed in the 2HDM Type-II so that SM backgrounds to the aforementioned $l^+l^-b\bar{b}$ signature are significant, in order to establish sensitivity to this BSM scenario too,

[1] Here, α is the mixing angle between the h and H states and $\tan\beta$ is the ratio of the Vacuum Expectation Values (VEVs) of the two doublets.

we deploy here advanced Machine Learning (ML) methods that could well be adopted by ATLAS and CMS, as they surpass the state-of-the-art therein in this respect.

Specifically, we carry out our search by utilizing a set of Deep Neural Networks (DNNs) that span all data types at the LHC, e.g., kinematical distributions, energy deposit of charged hadrons and (reconstructed) four-momenta of final state particles. A Multi-Layer Perceptron (MLP) network that analyzes the constructed kinematical distributions of the final state particles is also used. Then, a Convolution Neural Network (CNN), which analyzes jet images that can be constructed by visualising the pT (transverse momentum) distributions of the final state jets is exploited. Furthermore, we adopt a new method for a Siamese Neural Network (SNN) which is a twin encoder model with two training stages. The model maps the high dimensional input feature space to lower dimensional space (latent space) such that the Euclidean distance between images from different classes is maximal. For this purpose the SNN minimizes a modified contrastive loss function in the first training stage, while in the second training stage it minimizes an entropy loss function. Also, we adjust a Hybrid Deep Neural Network (HDNN), which is a two streams input network that can analyze the kinematical distributions and constructed jet images at the same time. Finally, to tackle the issue of sparse pixels in jet images, we utilize a suite of Graph Neural Networks (GNNs) to examine the graphs developed from the four-momenta of the final state particles. In this scenario, we employ four distinct GNNs: a Dynamic Graph Convolution Neural Network (DGCNN), a Graph Convolution Network (GCN), a Graph Attention (GAT) network, and a Graph Sample and AggreGate (GraphSAGE) network.

In order to study the influence of each network individually, we utilize a linear CKA to evaluate the similarity among hidden layer representations. This approach is necessary as Deep Learning (DL) models are typically considered as black boxes without innate explanations for their results. Instead, we leverage CKA to analyze the information learned by the hidden layers of each model, providing a robust explanation for each model's individual classification accuracy. Despite the linear CKA's innate ability to explain the reported model accuracy by scrutinizing the representation pattern within a model's hidden layers, we also use the CKA to compare the classification accuracy among different used models.

Our paper is organised as follows. In the next section, we describe the 2HDM, in particular, its Type-II realization. We then illustrate our overall analysis strategy, followed by a detailed description of the various ML methods that we advocate. Then we provide an analysis of the learned representations of the hidden layers of each model thereby offering a robust explanation of the reported accuracy of our results. After which, we present the latter and finally conclude.

2 The 2HDM

In this section we first give a brief review of the 2HDM with type-II Yukawa couplings focusing on the aspects of it which are relevant to our analysis. We then describe theoretical and experimental constraints applicable to it. We finally scan over its parameter space to extract interesting BPs to be used in our numerical analysis.

2.1 The Higgs potential

The 2HDM is an extension of the SM through a second $SU(2)_L$ Higgs doublet with the same quantum numbers under the SM symmetry gauge group [3, 13]. The two $(SU)_L$ doublet

fields, ϕ_1 and ϕ_2, are defined as

$$\phi_1 = \begin{pmatrix} \eta_1^+ \\ (v_1 + h_1 + ih_3)/\sqrt{2} \end{pmatrix}, \qquad \phi_2 = \begin{pmatrix} \eta_2^+ \\ (v_2 + h_2 + ih_4)/\sqrt{2} \end{pmatrix}, \tag{1}$$

in terms of four (pseudo)real scalar fields h_i, with $i = 1, \ldots, 4$, two complex charged fields η_i^+, with $i = 1, 2$, and two Vacuum Expectation Values (VEVs) v_i, with $i = 1, 2$. The Lagrangian density of the model can be decomposed as

$$\mathcal{L}_{\rm 2HDM} = \mathcal{L}_{\rm SM} + \mathcal{L}_\phi + V_\phi + Y_\phi \,, \tag{2}$$

where $\mathcal{L}_{\rm SM}$ contains the kinetic terms for the SM gauge fields and fermions, $\mathcal{L}_\phi$ contains those of the two Higgs doublet fields, V_ϕ denotes the Higgs potential of the two doublet fields and Y_ϕ is the Yukawa part which gives rise to the couplings between the Higgs fields and SM fermions. The most general 2HDM Higgs potential is given by

$$\begin{aligned} V_\phi = {} & m_{11}^2(\phi_1^\dagger\phi_1) + m_{22}^2(\phi_2^\dagger\phi_2) - \left[m_{12}^2(\phi_1^\dagger\phi_2) + \text{h.c.}\right] \\ & + \lambda_1(\phi_1^\dagger\phi_1)^2 + \lambda_2(\phi_2^\dagger\phi_2)^2 + \lambda_3(\phi_1^\dagger\phi_1)(\phi_2^\dagger\phi_2) + \lambda_4(\phi_1^\dagger\phi_2)(\phi_2^\dagger\phi_1) \\ & + \frac{1}{2}\left[\lambda_5(\phi_1^\dagger\phi_2)^2 + \left[\lambda_6(\phi_1^\dagger\phi_1) + \lambda_7(\phi_2^\dagger\phi_2)\right](\phi_1^\dagger\phi_2) + \text{H.c.}\right] . \end{aligned} \tag{3}$$

Such a potential allows for Flavor Changing Neutral Currents (FCNCs) at tree level, though, which are strongly constrained by experimental measurements. Adding a global Z_2 symmetry to the potential, with $(\phi_1, \phi_2) \to (\phi_1, -\phi_2)$ transformations, prevents the existence of FCNC sources in it [14]. However, the most general Yukawa interaction violates such a Z_2 symmetry, thus leading again to potentially FCNCs at tree level [15]. Thus, to tame the latter, only specific Yukawa structures, known as the aforementioned Types [3], are allowed. However, to enable EWSB compliant with the measured particle spectrum of the SM, a softly broken Z_2 symmetry should be enabled, by requiring a small but non-vanishing mass $m_{12}^2(\phi_1^\dagger\phi_2)$ and setting $\lambda_6 = \lambda_7 = 0$. (Herein, softly means that the model still respects the Z_2 symmetry at small distances in all order of perturbation theory.) The 'soft' mass m_{12}^2 and λ_5 are in general complex, though, with two phases $m_{12}^2 = |m_{12}^2|e^{i\eta(m_{12}^2)}$ and $\lambda_5 = |\lambda_5|e^{i\eta(\lambda_5)}$ [16, 17]. In the following, we will consider a real potential that preserves the CP symmetry, thus with vanishing complex phases, $\eta(m_{12}^2) = \eta(\lambda_5) = 0$. In such a configuration of the 2HDM, then 7 independent parameters remain, which are λ_i, with $i = 1, \ldots 5$, $\tan\beta = v_2/v_1$ and m_{12}^2, from which the physical parameters, i.e., Higgs boson masses and couplings, are obtained, with the constraint that one of the former must be set to 125 GeV or so (which in our case is the one of the h field). Finally, as mentioned already, amongst the possible Yukawa structures, we restrict our study to the Type-II only.

The tree level mass matrix squared for the Higgs fields can be obtained as

$$\left(\mathcal{M}^2\right)_{ij} = \left.\frac{\partial V_\phi}{\partial h_i \partial h_j}\right|_{h_{i,j}=0} , \tag{4}$$

where the h_i's ($i = 1, \ldots, 4$) are the four components of the complex doublet fields. Upon EWSB, three physical neutral scalars are obtained after diagonalizing the corresponding mass matrices, two CP-even (scalar) ones (h, H) and a CP-odd (pseudoscalar) one (A), with masses given by

$$m_{h,H}^2 = \frac{1}{2}\left[\chi_{11}^2 + \chi_{22}^2 \mp \sqrt{(\chi_{11}^2 - \chi_{22}^2)^2 + 4(\chi_{12}^2)^2}\right] , \tag{5}$$

$$m_A^2 = \frac{2m_{12}^2}{\sin 2\beta} - \lambda_5 v^2 \,, \tag{6}$$

with

$$\chi^2_{11} = m^2_{12} \tan\beta + 2\lambda_1 v^2 \cos^2\beta\,, \tag{7}$$

$$\chi^2_{22} = m^2_{12} \cot\beta + 2\lambda_2 v^2 \sin^2\beta\,, \tag{8}$$

$$\chi^2_{12} = -m^2_{12} + \frac{1}{2}(\lambda_3+\lambda_4+\lambda_5) v^2 \sin 2\beta\,, \tag{9}$$

where the VEVs satisfy the relation $v = \sqrt{v_1 + v_2}$ (with v being the SM one)[2]. As intimated, in the following, we will consider h as the SM-like Higgs boson discovered at the LHC in 2012.

To stay with the neutral Higgs sector, the imposed CP conservation only allows for tree level couplings between two massive gauge bosons and the CP-even Higgs states while the CP-odd Higgs state can only couple to a gauge boson and a CP-even Higgs one. Furthermore, all neutral Higgs states can couple to fermions. The couplings strength of the neutral Higgs bosons to both matter and forces are parameterized in terms of $\tan\beta$ and another parameter, α, which is the mixing angle between the CP-even Higgs states [3]. Specifically, the coupling strength of the AZh vertex is proportional to $\cos(\beta-\alpha)$.

2.2 Constraining the 2HDM free parameters

The 2HDM free parameters are constrained from various theoretical considerations and experimental observations. In order to account for the perturbativity of the Higgs potential, the magnitude of the couplings in the Higgs potential is constrained to $|\lambda_i| \leq 4\pi$ ($i = 1,\ldots 5$). The stability of the model vacuum constrains a combination of these couplings, as follows [18]

$$\lambda_1, \lambda_2 > 0, \quad \lambda_3 + \sqrt{\lambda_1\lambda_2} > 0, \quad \lambda_3 + \lambda_4 - \lambda_5 + \sqrt{\lambda_1\lambda_2} > 0\,. \tag{10}$$

The contribution of the 2HDM particles to EW Precision Observables (EWPOs) at the loop level affects the measured oblique parameters, which are constrained from global fits to be [19]

$$S = 0.03 \pm 0.10, \quad T = 0.05 \pm 0.12, \quad U = 0.03 \pm 0.10\,, \tag{11}$$

so that we account for these limits too. The precise measurements of the SM Higgs mass and coupling strengths by the ATLAS and CMS experiments add extra bounds on the properties of the SM-like Higgs, h [20–23]. Furthermore, the other neutral and charged Higgs states undergo constraints from null resonance searches at various colliders, see, e.g., [24–27]. The contribution of the charged Higgs boson to B meson decays sets severe bounds on the $(m_{H^\pm}, \tan\beta)$ plane, as mentioned. The dominant bounds come from the following Branching ratios (Br) measurements: $\mathrm{Br}(B^+ \to \tau^+\nu) = (1.06 \pm 0.19) \times 10^{-4}$ and $Br(B \to S\gamma)_{E_\gamma \geq 1.6\mathrm{GeV}}) = (3.32 \pm 0.15) \times 10^{-4}$ [28]. Specifically, for large $\tan\beta$ the charged Higgs boson mass is constrained to be $m_{H^\pm} \geq 600$ GeV or so while for $\tan\beta \leq 10$ such a mass bound is significantly relaxed [29, 31].

2.3 Assessment of the parameter space

In order to find viable parameter space points that satisfy all the mentioned constraints we scan over the aforementioned Higgs potential free parameters. For fast convergence we use the ML assisted scanner package of Ref. [29] to scan over the following ranges:

$$\begin{aligned} &0 \leq \lambda_1 \leq 10, \qquad 0 \leq \lambda_2 \leq 0.2, \qquad -10 \leq \lambda_3 \leq 10, \qquad -10 \leq \lambda_4 \leq 10, \\ &-10 \leq \lambda_5 \leq 10, \qquad 1 \leq \tan\beta \leq 45, \qquad -6000\ \mathrm{GeV}^2 \leq m^2_{12} \leq 0\ \mathrm{GeV}^2\,. \end{aligned} \tag{12}$$

[2]The other two Higgs states emerging from the 2HDM after EWSB are charged and are denoted by $H^\pm$.

(The narrow range of λ_2 is to keep the SM-like Higgs h as the lightest neutral Higgs state.) As a result, we obtain 300 000 points that satisfy all constraints, which are shown in Fig. 1.

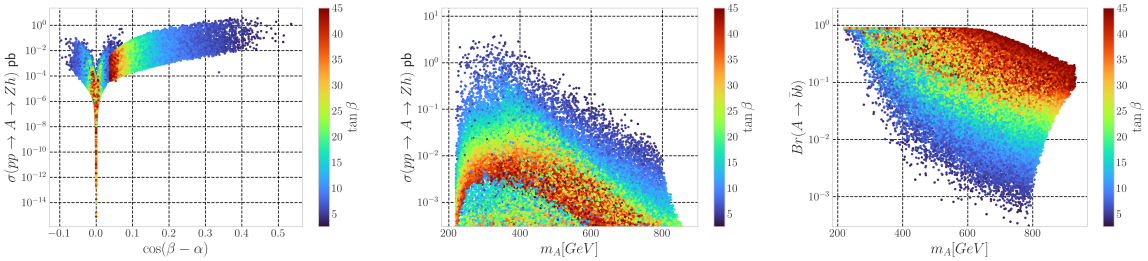

Figure 1: The scan output points that satisfy all constrains. (Left) Total cross section of the process $pp \to A \to Zh$ at $\sqrt{s} = 14$ GeV versus $\cos(\beta - \alpha)$. (Middle) The same versus m_A. (Right) The decay rate $Br(A \to b\bar{b})$ versus m_A. The color bar represents the corresponding $\tan\beta$ value in all plots.

The left plot shows the total cross section of the process $pp \to A \to Zh$ at $\sqrt{s} = 14$ GeV versus $\cos(\beta - \alpha)$, with the color bar representing the corresponding $\tan\beta$ value. The overall coupling strength is proportional to some function of $\tan\beta$, depending upon the relative size of the $ht\bar{t}$ and $hb\bar{b}$ couplings at production level times $\cos(\beta - \alpha)$ at decay level, the latter modulated by the functional form of the total width in terms of α and β. The middle plot shows the same data points mapped against m_A. By combining these two plots, it is clear that the production times decay cross section can be up to $\sim \mathcal{O}(1$ pb) for $m_A \leq 400$ GeV and/or $\tan\beta < 10$. The right plot shows the $Br(A \to b\bar{b})$, as this is the dominant one over the mass range of interest here, i.e., $m_A < 600$ GeV or so, while for larger m_A the dominant decay modes are $A \to Z^{(*)}H$ and, mostly, $A \to t\bar{t}$. Indeed, the $A \to Z^{(*)}h$ mode pursued here is never dominant, although it is maximised in the region between $m_Z + m_h$ and $2m_t$.

As for the separate dynamics of production and decay, it is worth emphasizing the following. On the one hand, for smaller $\tan\beta$, the main contribution to the production cross section comes from $gg \to A$ (i.e., gg-fusion) while, for larger $\tan\beta$, the dominant one is $b\bar{b} \to A$ (i.e., $b\bar{b}$-annihilation). On the other hand, for the $A \to Z^{(*)}$ decay rate, the dependence on $\tan\beta$ is less straightforward. As for the further two transitions, $Z^{(*)} \to l^+l^-$ ($l = e, \mu$) and $h \to b\bar{b}$, these (essentially) are SM processes. Finally, it is worth mentioning that, when the top quark loop (entering gg-fusion) exhibits an imaginary part for $m_A > 2m_t$, there occurs a destructive interference of our signal with the $pp \to Z^{(*)}b\bar{b}$ process, which yields a small reduction of the total cross section [32, 33], which we neglect here.

3 Analysis strategy

In this section, we numerically investigate our chosen signature of the CP-odd Higgs boson of the 2HDM Type-II at Run 3 of the LHC and HL-LHC using different recent ML models. Thus, we concentrate on the process $pp \to A \to Zh$ with $\sqrt{s} = 14$ TeV and an integrated luminosity L_{int} of 300 and 3000 fb^{-1}. The subprocesses of interest are initiated by $b\bar{b}$-annihilation and gg-fusion, eventually yielding a lepton l^+l^- ($l = e, \mu$) and b-jet pair, as shown in Fig. 2. We first review experimental results on this process obtained by ATLAS and CMS, we then borrow the most essential elements of one of their (kinematical) analyses to introduce our own approach, before moving on to describe the ML part of it.

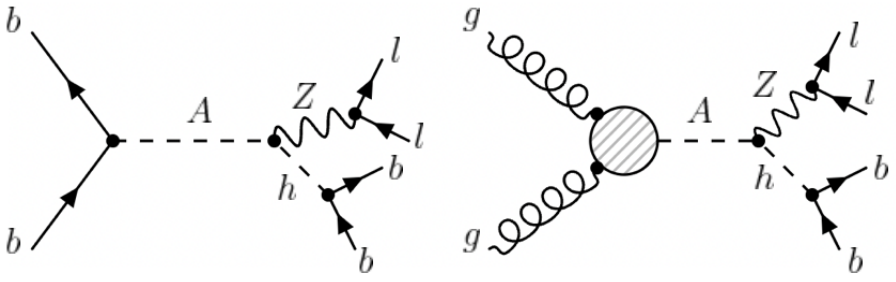

Figure 2: Feynman diagrams for the considered signal subprocesses.

3.1 Current ATLAS and CMS results

Up-to-date searches for $pp \to A \to Zh$ signals at the LHC, in a variety of final states, were recently reviewed in great detail in another paper of one of us [12], to which we refer the reader, including their interpretation in two different 2HDM Types. Here, we limit ourselves to the case of the Type-II considered (in normal mass hierarchy) and refers specifically to the $Zh \to l^+l^-b\bar{b}$ ($l = e, \mu$) signature.

The decay $A \to Zh$ has been searched for at the LHC by both the ATLAS and CMS collaborations assuming the case of normal mass hierarchy (i.e., $m_h = 125$ GeV) and an on-shell Z boson, i.e., specifically assuming $m_A \geq m_Z + m_h$. Just like here, in such searches, the A state is assumed to be produced via $b\bar{b}, gg \to A$[3] and the decay rates of the h state to fermions are given by the measurements of the Br's of the 125 GeV boson, thus $Br(h \to b\bar{b}) \approx 57\%$.

In the CMS analysis of Ref. [34] (see also Ref. [35]), with $\sqrt{s} = 13$ TeV and 35.9 fb^{-1}, targeting the $m_A > 225$ GeV range, separate searches are carried out for the decays $Z \to e^+e^-$ and $Z \to \mu^+\mu^-$. In each case, the signal is separated into categories with 1, 2 and 3 b-jets. In general, the selection efficiencies are similar for the two production mechanisms in the 1 and 2 b-jet categories and they increase slightly with increasing m_A while in the 3 b-jet category the selection efficiency for $gg \to Ab\bar{b}$ is considerably larger (due to the presence of more b-jets in the signal) than that for $gg \to A$, being almost an order of magnitude greater for $m_A < 300$ GeV. Furthermore, the SM backgrounds to the $l^+l^-b\bar{b}$ signature are largest for the 1 b-jet category and smallest for the 3 b-jet one. In the 2 b-jet category (that we are assume here), the dominant backgrounds are found to be $pp \to Zb\bar{b}$ and $pp \to t\bar{t}$, which we shall adopt here too. Searches for the above signature by the ATLAS collaboration in Refs. [36] (with 36.1 fb^{-1} of luminosity) and [37] (with 139 fb^{-1} of luminosity) have similar strategies and derive comparable limits on the total cross section, over the ranges $m_A > 220$ GeV and $m_A > 280$ GeV, respectively.

In carrying out our ML driven analysis, we will compare our results with some of those from the aforementioned ATLAS and CMS analyses, as well as an earlier ATLAS study, the one of Ref. [38], based on 3.2 fb^{-1} of data and starting from $m_A = 220$ GeV, which is, in fact, the one offering the simplest kinematical analysis, upon which we will model ours.

3.2 Kinematical analysis

Following the analysis in [38], we require events to have two isolated leptons, these being either electrons or muons, and two isolated b-tagged jets. The reconstructed events satisfy the following requirements in transverse momentum, pseudorapidity and separa-

[3] Although the emulation of the first subprocess via $gg \to Ab\bar{b}$ is sometimes used [39].

tion: $pT(l) > 20$ GeV, $pT(j) > 25$ GeV and $|\eta(l, \text{jet}, b)| \leq 2.5$. Furthermore, our selection cuts on the invariant masses of the hadronic and leptonic systems are as follows: 75 GeV $< m_{bb} <$ 145 GeV and 70 GeV $< m_{ll} <$ 110 GeV. Jets are reconstructed with the anti-k_T algorithm [40] but our results proved stable against a change of clustering algorithm to the Cambridge-Aachen one [41, 42].

Reconstructed events are required to have at least two isolated b-jets with cone radius $R = 0.4$ using a flat b-tagging efficiency of 70%. For the mistagging rate of gluon and light quark jets as b ones, we adopt a flat rate of 10^{-3} while, for c-jets, we use 10^{-2}.

The dominant background contributions come from the W^+W^- leptonic decays in the $pp \to t\bar{t}$ process and from $pp \to Zb\bar{b}$, where the Z boson decays leptonically. Other background processes like single-top production, di-gauge boson production, associated production of a gauge boson with the SM Higgs and $pp \to W^{\pm}\bar{b}b$ are not considered as they can be removed by the basic cuts applied here [38].

Table 1: Input parameters for our four BPs. The last column shows the total cross section for the process depicted in Fig. 2.

m_A[GeV]	λ_1	λ_2	λ_3	λ_4	λ_5	$\tan\beta$	m_{12}^2 GeV^2	$\cos(\beta-\alpha)$	σ_{tot} [fb]
200	6.81	0.14	1.86	−0.12	−0.31	5.02	−4260	0.37	65.8
250	6.12	0.14	1.86	−0.11	−0.81	5.01	−4270	0.38	86.49
300	5.22	0.13	4.00	−1.82	−1.44	4.68	−4530	0.35	109.42
350	4.69	0.14	3.85	−1.35	−1.57	4.38	−5440	0.34	95.68

We carry out the analysis for four BPs, with $m_A = 200, 250,$ 300 and 350 GeV. For the first BP, the Z boson is produced off-shell while in other three cases is on-shell. The four BPs are chosen from the output of the scanned points mentioned in section 2.3, all of which satisfy all relevant theoretical and experimental bounds. Tab. 1 shows the input parameters for the four BPs, alongside their production times decay cross sections, down to the final state $l^+l^-b\bar{b}$. We notice that all our BPs belong to the 'wrong-sign scenario', i.e., the right-arm region of Refs. [9, 10], typically offering larger total cross sections than in the 'alignment limit', thereby making these particularly amenable to experimental analyses.

Simulation of the signal and background events proceeds through a chain of sequentially automated steps. For events generation and cross section calculation we use MadGraph [43] with its standard generation level cuts (which do not bias our detector level results). For gg-fusion, the loop implemented in MadGraph is an effective vertex as described in [44]. SPheno [45, 46] is used to compute the numerical value of such an effective vertex at the Leading Order (LO) in perturbation theory. PYTHIA [47] is exploited for parton showering, hadronization, heavy flavor decays and for adding the soft underlying event, multi-particle scatterings, etc. FastJet [48] is used for jet clustering. The fast simulation of the ATLAS detector was done with the DELPHES package [49]. Finally, the standard ATLAS card is modified to be able to simulate the tracks and energy deposit from the charged hadrons.

4 DNN

After event simulation, we adopt different types of ML models to analyze different categories of events, kinematical distributions, energy deposits of charged hadrons and (reconstructed) four-momenta of the final state particles.

Starting with high-level kinematical distributions, we adopt a MLP model to optimize the separation between the signal and background distributions. The constructed distributions have unique information about the global structure of the signal and background

events, thus the structure of the MLP network, with fully-connected layers, is able to analyze the global features ending up with large classification power between the signal and background events. Although the MLP can achieve high classification performance, the fact that some background distributions have similar kinematical structure to signal ones hinders the overall classification power. However, one can improve the classification performance by applying initial cuts that maximize the signal-to-background yield before feeding the distributions to the MLP. Furthermore, the constructed kinematical spectra exhibit a large correlation among each other and applying a cut on any distribution will, in some cases, affect the structure of all others, aspect which then continues to hinder the classification performance of the MLP. In order to control the global impact of the initial cuts, one has then to decorrelate such a dependence across the kinematical variables via the square-root of the covariance matrix or Gaussian transformation of variables as described in [50]. In the end, although the initial cuts may increase the classification performance, we opted not to apply any thus allowing full freedom to the MLP in finding the optimal classification boundaries.

A second approach is to analyze the charged hadrons by exploiting the fact that, in an unbroken $SU(3)_C$, color is conserved in the QCD interaction and provides different color flow structures for different processes. The structure of the color flow depends on the color nature of mediating particles, e.g., the radiation pattern within and around b-(anti)quark pairs from Higgs boson decays is expected to be different from the radiation pattern of the same from tt production or Zbb processes. In order to exploit the color flow properties to classify signal and background events, one can think of the LHC detector as a giant camera and the streams of hadrons as images. The constructed images are two dimensional arrays in the $(\eta - \phi)$ plane while the pixels size is adjusted to be within the detector response and the pixels are weighted by the sum of the total transverse momentum deposited in the corresponding part in the detector [51–54]. We adopt a CNN model to analyze the constructed jet images and output the classification probability for signal and background events. The CNN is constructed by combining two different sets of hidden layers, convolution ones and fully-connected ones. Convolution layers are constructed from filters (kernels) that share their weights locally and hence they are able to capture local information stored into the images while the fully-connected layers are handcrafted to analyze the captured local information ending up with global information about the image by adjusting different structures of the neurons for signal and background images.

Although the CNN is designed to capture local information of the jet images, there is no guarantee it can capture some hidden information, e.g., similar or dissimilar local information for images from the same or different classes, respectively. For this purpose, we introduce a SNN [55, 56]. This is a two-step training network with twin encoders that share their weights. As a twin convolution encoder model, it processes the images in pairs from the same or different classes. In the first training stage the model learns similar features shared amongst images from the same class, e.g., pairs of signal or background images, by minimizing the latent space Euclidean distance between them and maximizing the distance between images from different classes, i.e., pairs from signal and background images. Once the latent space is shaped by separating the Euclidean distance between signal and background images, the second training stage starts by freezing the optimized weights for one encoder and adding a fully-connected layer and one output layer with two output neurons to identify the signal and background events. The construction of the SNN enables it to learn hidden features that are shared among each class.

To incorporate the different data structure as inputs to a neural network, a dual-input HDNN is then constructed [57–60]. The first stream consists of fully-connected layers that process the reconstructed kinematical variables (see below). The second stream consists of

two-dimensional convolution layers and pooling layers that process the jet images. The two streams are then concatenated into one flatten layer, then, for better expressivity, a fully-connected layer is added before the final output layer. The HDNN model with dual inputs has the advantage to combine the global information captured by the fully-connected layers acting on the kinematical distributions and the local information captured by the kernels in the convolution layers acting on the jet images. To analyze the combined information, global and local one, exalts the model expressivity in terms of signal and background events, which in turn enhances the overall classification performance.

CNNs are specifically tailored to process grid-like data structures, such as images, where local information is paramount. By using predefined filters, CNNs capture local patterns effectively. However, this design inherently carries significant inductive biases [61]. As the constructed jet images are sparse, inductive biases confuse the model, which ends up with lower classification performance. The challenge here stems from the model inability to adequately process sparse, non-grid-like data, which is a significant limitation for CNNs. To address these issues, we propose the use of GNNs instead. Unlike CNNs, GNNs can process input data that naturally form a graph structure, with entities represented as nodes and relationships as edges. This makes GNNs adept at handling sparse and/or irregular data. In the case of jet physics, for instance, the four-momenta of the final state particles can be seen as graph nodes, while the graph edges can be weighted by the angular distance between the particles. GNNs have an inherent ability to handle both local and global information in the data. They propagate information across the graph, allowing each node to be influenced by its neighbor information and iteratively capture long-range dependencies. This makes GNNs better performing so as to overcome the limitations of CNNs in this context.

As a generic set-up for all the proposed DNNs, we require all models to have an output layer with two neurons and a softmax function. The loss function is the categorical cross entropy defined as

$$\text{Loss} = -\sum_i Y_i \log(\hat{Y}_i)\,, \tag{13}$$

with $i = 1, 0$ for signal and background classes, respectively, and $Y_i, \hat{Y}_i$ are the true and predicted labels from each class. The dimension of the final output probability, $\hat{Y}$, is 1×2, $(\mathcal{P}_{sig}, \mathcal{P}_{bkg})$, with $\mathcal{P}$ ranging between $[0, 1]$. If $\mathcal{P}_{sig} > 0.5$ ($\mathcal{P}_{bkg} < 0.5$), the corresponding event is classified as most likely being a signal event and if $\mathcal{P}_{sig} < 0.5$ ($\mathcal{P}_{bkg} > 0.5$) the corresponding event is classified as most likely being a background event. An AdamW optimizer [62] is used to optimize the minimization of the loss function with learning rate 10^{-3}, weight decay 4×10^{-3} and exponential decay rate 0.9. The size of the input data is $200,000$ in all models, divided into 70% for training and 30% for testing the model accuracy. The DNNs are trained and tested on equal size data sets for signal and background events.

it's worth noting that we haven't fine-tuned the proposed networks, given the substantial computational resources that would be necessary. Indeed, conducting a grid search of the hyper-parameters could potentially improve the accuracy of the classification results we've reported.

4.1 MLP

A MLP is the basic type of a feed-forward DNN which consists of fully-connected hidden layers of different length. Given the nature of the fully-connected layers, a MLP is designed to learn the global information in the reconstructed kinematical distributions. This can be achieved by firing specific neurons in each hidden layer corresponding to the signal or background distributions. After training, a MLP exhibits specific structures of the fired neurons in case of signal or background events. We point out that the full connection of the

MLP hidden layers enables the model to propagate all event information among all hidden layers and thus its ability to learn global information about the event is increased [4].

For optimal classification performance, we select distributions with high discrimination power between signal and background events. To select the highly ranked distributions we follow a Sequential Backward Selection (SBS) feature [63] by first constructing all possible kinematical distributions and 'greedily'[5] removing one feature after another, in order to find the one that maximizes a cross-validated score when an estimator is trained on this single feature. The feature selection method indicates highly ranked nine kinematical distributions as shown in Fig. 3, e.g., for the signal BP with $m_A = 300$ GeV. Although the kinematical distributions herein are for a specific signal point, we found that other signal BPs have similar discriminative power. The selected kinematical distributions can be chosen as follows.

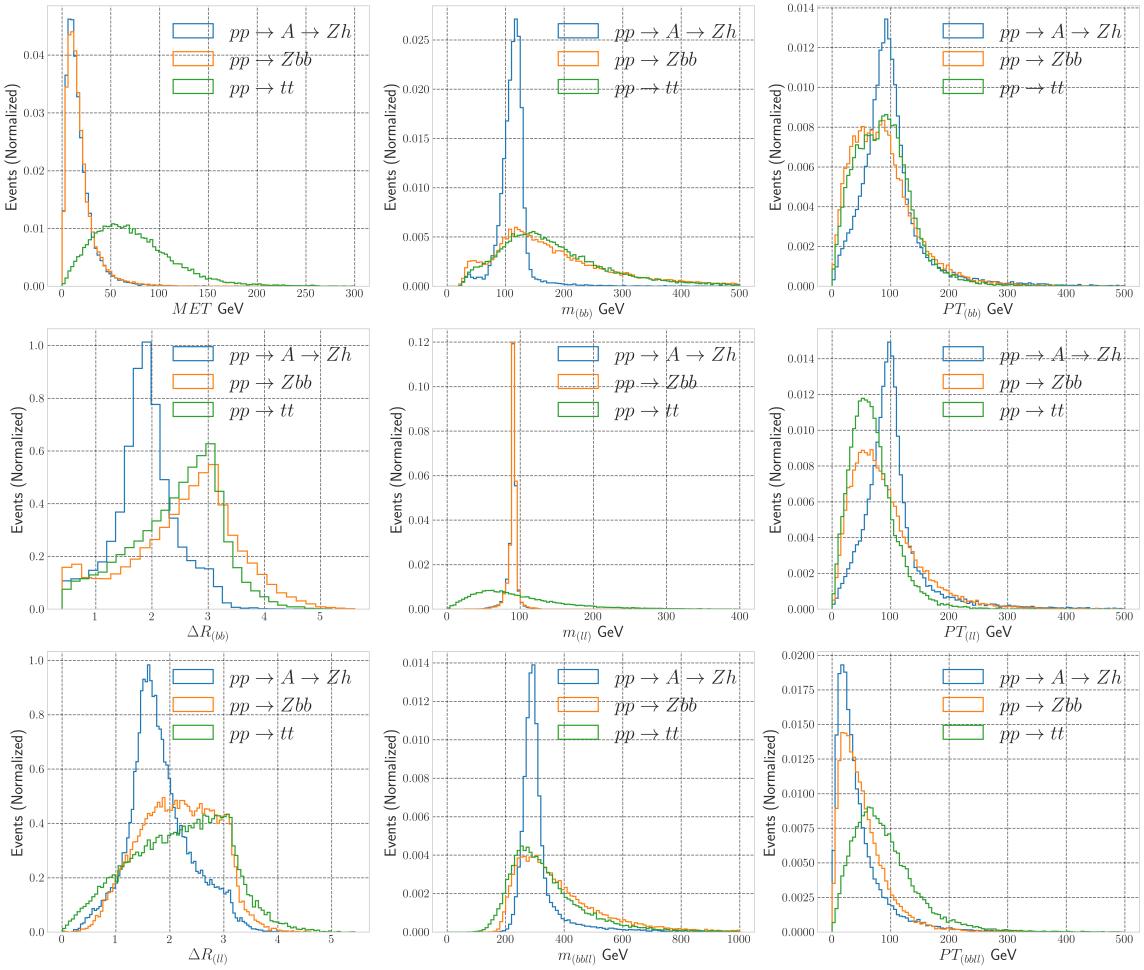

Figure 3: Kinematical distributions for signal (BP with $m_A = 300$ GeV) and background events superimposed and normalized to 1 before applying the pre-selection cuts. The color codes hold for all distributions as follows: signal (blue), $pp \to t\bar{t}$ (green) and $pp \to Zb\bar{b}$ (orange).

[4]Obviously, a MLP cannot be used for jet image analysis as the nature of the fully-connected layers makes the model depend on the spatial position of the energy deposits into the image. In contrast, a CNN with local weight sharing among its kernels makes the model independent of such a spatial position.

[5]That is, by using a greedy algorithm that follows the problem-solving heuristic of making the locally optimal choice at each stage.

- MET: Missing Transverse Energy (MET), defined as $MET = |-\sum_{v_i} \vec{pT}(v_i)|$, which is the sum of the transverse momenta of the visible particles. This is very similar for the signal and $pp \to Zb\bar{b}$, with $pp \to t\bar{t}$ set apart.

- $m_{(bb)}$: Invariant mass of the b-jet pair. Signal events show a narrow peak around the rest mass of the SM-like Higgs boson while background events show a broader peak as they are initially produced from QCD radiation in the case of $pp \to Zb\bar{b}$ and from top decays in the case of $pp \to t\bar{t}$.

- $PT_{(bb)}$: Transverse momentum of the b-jet pair reproducing m_h, which exemplifies the Higgs boson boost in signal events (and is different for other BPs). This distribution has a strong degree of similarity with the background ones.

- $\Delta R_{(bb)}$: Angular distance separation between the two b-jets reconstructing the Higgs boson, with $\Delta R_{(bb)} = \sqrt{(\Delta\eta_{(bb)})^2 + (\Delta\phi_{(bb)})^2}$. For the $pp \to Zb\bar{b}$ background, the two b-jets recoil against the associated Z when the latter is produced near its mass shell, thus they have a small boost factor and fly back-to-back with angular distance around π. A similar behavior also applies to the b-jets emerging from top-(anti)quark leptonic decays, for which $\Delta R_{(bb)}$ peaks again around π. Signal events show instead a narrow peak around 1.6 (for a heavier A, e.g., $m_A = 350$ GeV, the b-jets receive extra an boost and $\Delta R_{(bb)}$ peaks around 1.

- $m_{(ll)}$: Invariant mass of the lepton pair. Reconstructed events from the signal and $pp \to Zb\bar{b}$ processes offer a tight reconstruction of the Z boson mass by showing a narrow peak around m_Z while, for $pp \to t\bar{t}$ events, the final state leptons emerge from a pair of $W^{\pm}$ boson decays, thereby missing such a distinctive feature. (In the case of signal events from the BP with $m_A = 200$ GeV, the Z^* boson is produced off-shell and thus the invariant mass peak for the signal is very similar to that of $pp \to t\bar{t}$.)

- $PT_{(ll)}$: Transverse momentum of the two leptons reproducing m_Z which exemplifies the $Z^{(*)}$ boson boost in signal events (again, the latter and $pp \to t\bar{t}$ events have similar distributions which are in turn different from that of the background process $pp \to Zb\bar{b}$).

- $\Delta R_{(ll)}$: Angular distance separation between the two leptons reconstructing the $Z^{(*)}$, which exhibits a similar behavior as the angular separation between the two b-jets in all cases.

- $m_{(bbll)}$: Invariant mass of the b-jet and lepton pairs which reconstruct the masses of the h and $Z^{(*)}$ boson, respectively, in turn reconstructing the mass of the A state. (For the BP with $m_A = 200$ GeV, the reconstructed A mass peak is broader as the final leptons from the off-shell Z^* boson decay are soft and can be missed.) There is a clear difference here between signal and background events.

- $pT_{(bbll)}$: Transverse momentum of the b-jet and lepton pairs which reconstruct the masses of the h and $Z^{(*)}$ boson, respectively, in turn exemplifying the A state boost in signal events, which overlap significantly with that of background ones.

After reconstruction of the kinematic distributions we stack all backgrounds and signal events separately such that each data set has dimensions $d_{\text{distribution}} = (9, N)$ with N being the total number of events. As a supervised classification problem, we assign a numeric label $Y = 1$ to the signal events and $Y = 0$ to (the whole of) the background events.

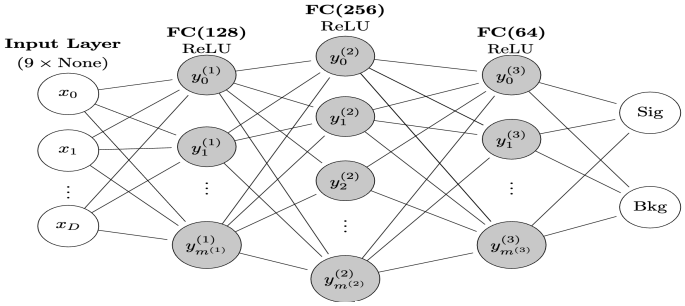

Figure 4: A schematic architecture for the used MLP model. Here, we visualize a Fully-Connected (FC) NN layer.

Having the input distributions and their labels adjusted, we use MLP with an input layer of the same dimension as the inputs. Fig. 4 shows a schematic architecture of the used MLP model which consists of three pairs of fully-connected hidden layers with Rectified Linear Unit (ReLU) activation function and an output layer with two neurons and softmax activation function. The number of neurons in the first pair is 256, in the second is 128 and in the third is 64. To avoid over-training, each hidden layer pair is followed by a dropout layer. During the training process the model tries to minimize the difference between its

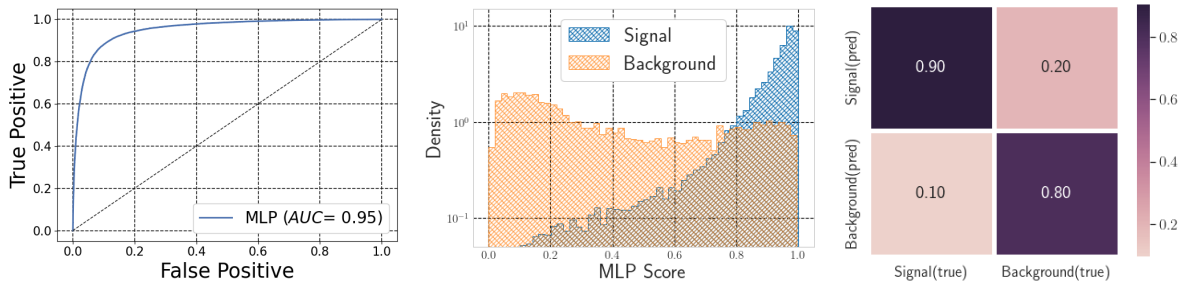

Figure 5: MLP test results when trained on kinematic distributions for signal events (BP with $m_A = 300$ GeV). The ROC curve (left), MLP output score (middle) and confusion matrix (right) are shown.

predictions and the assigned labels. To measure the model ability to generalize to new unseen data, we test the model accuracy to unseen test set.

Fig. 5 shows the MLP results from the test sample for the signal BP with $m_A = 300$ GeV. To quantify the classification power of the model, the left plot shows the Receiver Operator Characteristic (ROC) curve[6] The middle plot shows the output score of the model for the signal events (blue) and background events (orange). The right plot shows the CM when a symmetric threshold value at 0.5 is used on the model output.

[6]The ROC curve is an evaluation metric for binary classification problems: it is a probability curve that plots the true positive rate against the false positive rate at various threshold values and essentially separates the 'signal' from the 'noise' caused by misclassifying the background. In other words, it shows the performance of the model to identify the signal events at all classification thresholds. The Area Under the Curve (AUC) is the measure of the ability of a binary classifier to distinguish between classes and is used as a summary of the ROC curve. As the ROC curve quantifies the relation between true and false positive rates, which indicates the model ability to identify the signal events, the Confusion Matrix (CM) reports the values for both positive and negative hypotheses. Accordingly, one can clearly estimate the model response to identify the signal and background events.

4.2 CNN

As mentioned above, the global structure of the color flow can be seen at the LHC as a color string from the soft hadrons that stretch between the two colored connected jets. The different color structure for different processes originates from the color nature of the parent particle, which can provide the event with an observable to aid the search for new physics. The two b-quarks from the decay of the Higgs boson form a color dipole whose radiation pattern is contained primarily within a pair of cones around the two b-quarks, with a tendency for more radiation to occur in the region between the two. In contrast, the two b-quarks in $pp \to Zb\bar{b}$ and $pp \to t\bar{t}$ events come from colored particles and are thus not directly connected, forming two isolated cones with less radiation in the region between the two b-quarks that in the signal. To effectively identify the different radiation patterns, we construct the jet images as a squared array in the (η, ϕ) plane with each pixel given by the total hadron pT deposited in the associated region of the calorimeter. In Fig. 4.2 we show normalized pT distributions for $50,000$ events for signal BP with $m_A = 300$ GeV (left), $pp \to Zbb$ (middle) and pp (right). To ensure that the CNN is not learning space-time

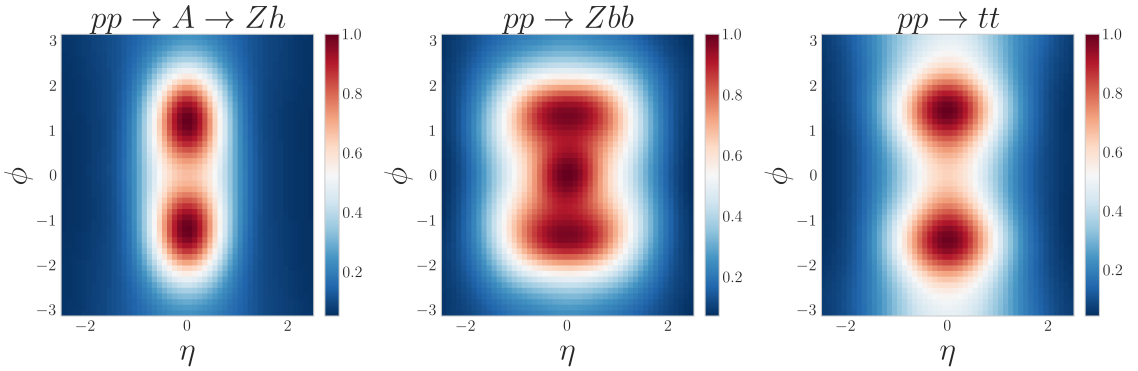

Figure 6: Normalized pT distribution for accumulated $50,000$ events after pre-processing steps for signal (BP with $m_A = 300$ GeV) and backgrounds events.

symmetries and can be generalized to new unseen data at different locations, the jet images are pre-processed as follow:

1. Image cleansing Images are constructed only from hadrons which have track information while at the same time we remove leptons (and photons).

2. Pixelization The region in the (η, ϕ) plane is discretized into a 50×50 grid with each pixel weighted by the sum of the transverse momentum in it.

3. Centering We center all particles in an image by shifting $(\frac{(\eta_b+\eta_{\bar{b}})}{2}, \frac{(\phi_b+\phi_{\bar{b}})}{2})$ to $(0, 0)$, which assists the independence of the model classification from the spatial location of the radiated hadrons.

4. Momentum smearing Constructed images are mostly sparse, which hinders the classification performance of the CNN model. To reduce the number of the sparse pixels, we smear the transverse momentum using a Gaussian function within 3 Standard Deviations (SDs) or σ's [64].

5. Normalization We normalize the pixel intensity by dividing each pixel in an image by the maximum pixel intensity value, which helps the model to converge to the global minimum of the loss function.

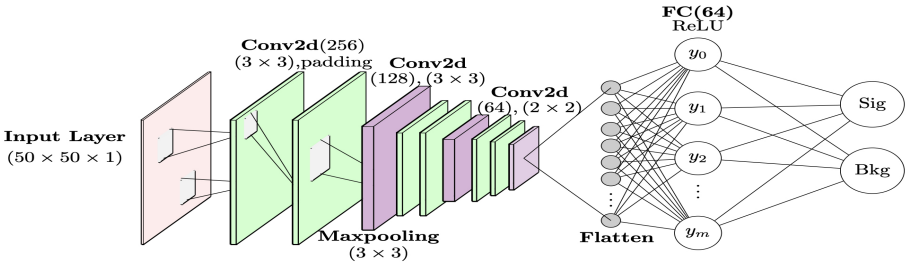

Figure 7: A schematic architecture for the used CNN model. Here, 'Conv2d' represents a two-dimensional CNN layer.

After the pre-processing steps, the constructed image has the dimension of $50 \times 50 \times 1$. In principle, one can add more information to the images, e.g., leptons, MET, etc. properties [57]. That information can be incorporated into an image by expanding the last dimension of it, i.e., the image depth. Although having more information into an image allows the model to learn more characteristic features of the events, we found that including leptons and MET information to our images does not increase the classification performance very much. Instead, we opted not to include such an information in order to reduce the computational costs.

To analyze the constructed jet images we adopt a CNN model with the structure depicted in Fig. 7. The model consists of four convolution layers, one fully-connected layer and one output layer. The first and second convolution layers have 256 kernels with kernel size 3, ReLU activation function and stride length of 1. To keep the dimensions of the original input images, we use a padding layer. Third and fourth convolution layers have 128 kernels with kernel size 3 and ReLU activation function. Fifth and sixth convolution layers have 64 kernels with kernel size 2 and ReLU activation function. After the second and the fourth convolution layers we use max pooling layers with size 2×2. After these, we use a dropout layer with 30% dropout rate. Output from the last convolution layer is flattened and projected to one fully-connected layer and dropout layer with 64 neurons and ReLU activation function.

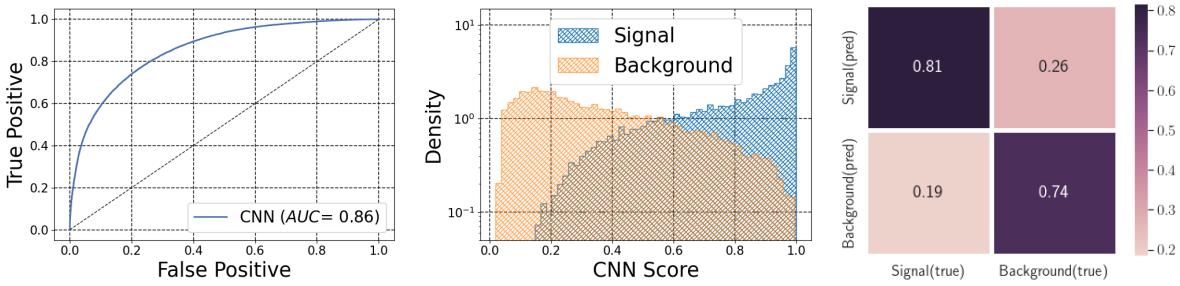

Figure 8: CNN test results when trained on jet images for signal events (BP with $m_A = 300$ GeV). The ROC curve (left), CNN output score (middle) and confusion matrix (right) are shown.

The results of the CNN analysis are shown in Fig. 8. The classification performance is here quantified by the reported AUC metric with value 0.86. We finally notice that the classification performance of the CNN analysis of the jet images can be enhanced with extra pre-processing steps, e.g., by constructing the Lund plane [65] or Riemannian mapping [66].

4.3 SNN

The SNN was first introduced as an algorithm for handwritten signature verification [67]. The main power of the SNN is that it maps input features into a latent space such that a simple distance in it, the Euclidean distance, approximates the characteristic features in the original one. It consists of two identical convolution encoders sharing the same set of weights in order to compare a pair of feature vectors in terms of their similarity or dissimilarity. It realizes a non-linear embedding of the data with the objective of bringing together similar examples and to move apart dissimilar examples. To measure the similarity or dissimilarity of the input pairs we use the Euclidean distance as a similarity metric learning given by

$$D = \sqrt{\sum_{i}^{n} \left(x_i^1 - x_i^2\right)^2}\,, \tag{14}$$

where x^1 and x^2 are the latent outputs from the two encoders and n is the latent space dimension. More precisely, given a pair of input images, the two encoders extract the features in each image and map these onto the latent space as vectors (x^1, x^2). The SNN then minimizes the Euclidean distance between x^1 and x^2 if they belong to the same class, e.g., the signal or background class, while it maximizes the Euclidean distance between x^1 and x^2 if they belong to different classes, e.g., signal and background classes. To do so, the SNN has to be trained in two stages. Firstly, the model computes the similarity or dissimilarity by minimizing a modified contrastive loss function as

$$\mathcal{L}(y, D) = \alpha(1-y) * D^2 + y * \left[\mathrm{Max}(\beta - D, 0)\right]^2\,, \tag{15}$$

with y, D being the true and predicted distance, respectively, while α, β are the margin parameters for learning the similarity and dissimilarity, respectively. Both parameters are hyper-parameters to be tuned (in our study we fix both to 1). Also, we adjust the true distance between the negative pair to be 1 and 0 for the positive pair. We would also like to mention that, in the self-supervised contrastive learning, data augmentation is used for learning similarity and dissimilarity [68–70]. In this case, the classification performance depends on the impact of the data augmentations [71]. Moreover, strong augmentations and implicit regularization may cause dimensional collapse of the projected data into the latent space [72]. We stress that, for our SNN with supervised contrastive learning, the mentioned problems no longer exist.

Once the model is trained to minimize the contrastive loss function, we start the second learning stage by freezing the weights of one of the encoders and add two fully-connected layers and one output layer with two neurons and softmax function. For the training in the second stage, signal images are labelled with 1 while background images are labelled with 0 and we use a categorical cross entropy loss function. A schematic architecture of the used SNN model is shown in Fig. 9, which consists of two identical convolution encoders, each of these having the same structure as the discussed CNN without the output layer.

The results of the SNN are shown in Fig. 10, which shows a larger classification performance over the used CNN with $AUC = 0.95$. Although the SNN processes the same jet images as the CNN network, it shows an improved classification accuracy over the CNN. This enhancement is, obviously, due to the minimization of the contrastive loss function, which in turn enables the model to learn more information from the jet images. (A detailed discussion about the learned representations by the hidden layers and its impact on the classification performance is presented in section 5.)

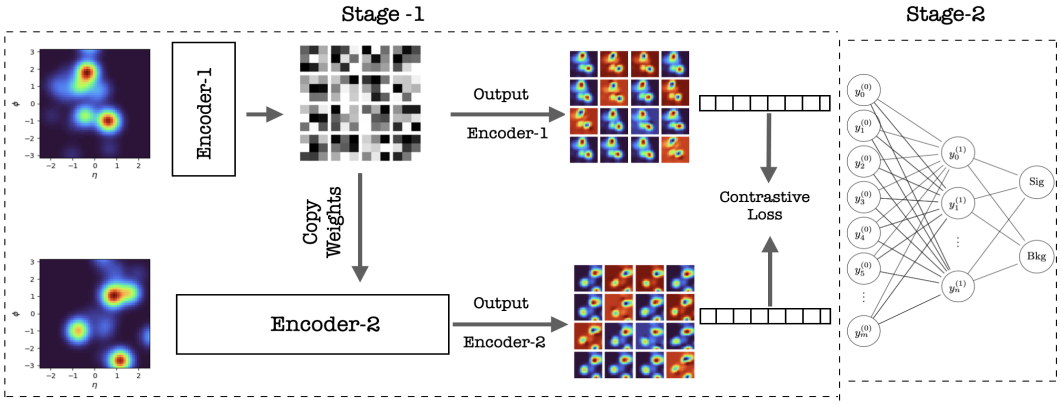

Figure 9: A schematic architecture for the used SNN model. Here, Encoder-1 and Encoder-2 are identical and copy their weights during the first training stage. Input images can be a positive pair, i.e., both images are either signal or background, or a negative pair, i.e., one signal image and one background image.

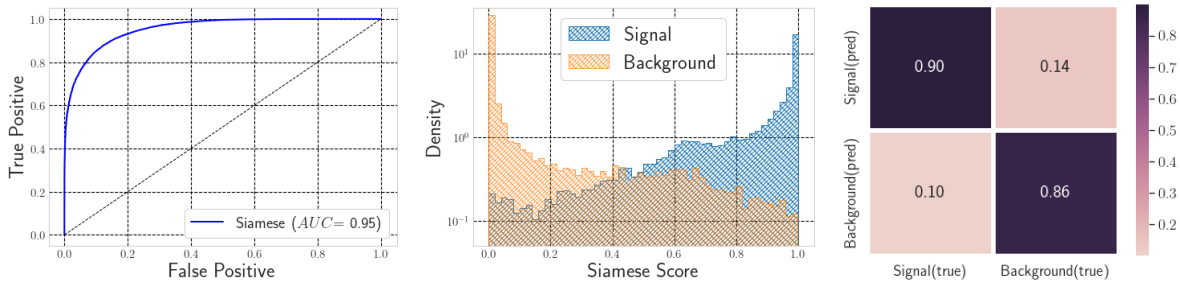

Figure 10: SNN test results when trained on jet images for signal events (BP with $m_A = 300$ GeV). The ROC curve (left), SNN output score (middle) and confusion matrix (right) are shown.

4.4 HDNN

To improve the expressivity of the ML model of signal and background events one can incorporate different information into the discussed models, e.g., by adding the lepton information to the reconstructed images to be analyzed by the CNN or encoding the hadrons information as distributions to be analyzed by the MLP. In both cases one can find a slight improvement in the classification performance of each model individually, as the latter is still able to learn specific types of event information, local or global. Furthermore, concatenating a MLP and CNN into a two stream HDNN model can improve the classification performance [57–60]. The first stream, which processes the input jet images, consists of convolution, max pooling and drop-out layers plus one flattened layer. The second stream, which processes the kinematic distributions, consists of fully-connected and drop-out layers. Both streams are then concatenated to one fully-connected layer and one output layer with two neurons for predictions, a HDNN. The two streams map the high dimensional information onto their own latent space (a lower dimensions space), e.g., the CNN and MLP map the local and global high dimensional information onto lower dimensional space individually. Concatenating the decomposed information in each latent space into one flatten layer expresses all characteristic features of the signal and background events. A schematic architecture of the HDNN is shown in Fig. 11.

The HDNN is constructed by combining the above CNN convolution layers and MLP

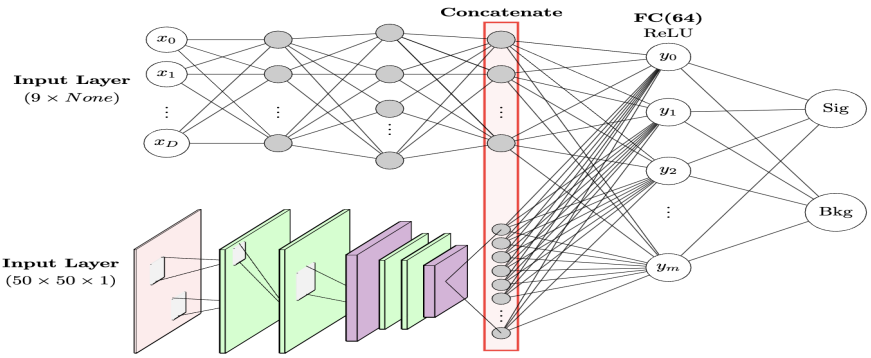

Figure 11: A schematic architecture for the used HDNN model.

without the output layer. A concatenation layer is used to connect the last layer of the two models. A fully-connected layer with 128 neurons is added with ReLU activation function and a dropout layer with a 30% dropout rate. The last output layer consists of two neurons and a softmax activation function.

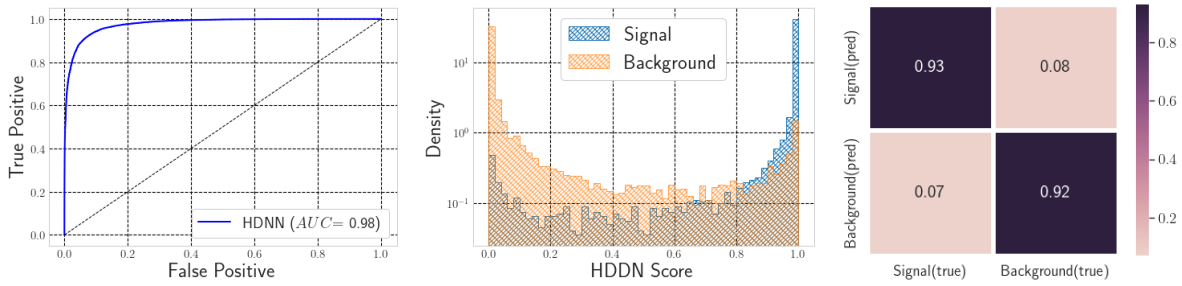

Figure 12: HDNN test results when trained on kinematic distributions and jet images for signal events (BP with $m_A = 300$ GeV). The ROC curve (left), HDNN output score (middle) and confusion matrix (right) are shown.

The results of the HDNN analysis are shown in Fig. 12. The ROC curve shows a large improvement in the HDNN classification performance over the MLP or CNN individually, with $AUC = 0.98$. Moreover, incorporating the different data types into a HDNN with two streams enables the model to learn the intrinsic features of the signal and background events with equal rates as reported by the confusion matrix (right plot).

4.5 GNN

One way to avoid the sparsity issue of image-based NNs is to utilize a graphical structure consisting of nodes and edges to encode particle information. GNNs can then be employed to incorporate the topological relationships among the nodes and edges and learn graph-structured data. Each reconstructed final state object is represented by a single node in this approach. This study, in alignment with the methodology described in [58], represents each node i in the input layer as a feature vector $\mathbf{x} = (I_l, I_b, m, pT, E)$ that collects the properties of the corresponding particle, where, e.g., m, pT and E denote the invariant mass, transverse momentum and energy of a particle system, respectively. The initial values for I_l and I_b are set to 0. The hardest lepton and b-jet in an event assign a value of 1 to I_l and I_b, respectively, while, the second hardest lepton and b-jet in the event assign a value of -1 to I_l and I_b, respectively. The angular correlation between two nodes i and j is

represented by an edge vector $e_{i,j}$, which consists of a single component that is defined by the angular distance $\Delta R_{(x_i,x_j)}$ between the particles in nodes i and j.

In the course of our comprehensive study, various architectures were tested to identify the optimal model capacity. Subsequent to extensive trials and analysis, our observations revealed that the maximum performance across all tested GNN models was achieved with a configuration of three hidden layers.

This preference can be rationalized by considering the nature of the graphs involved, typically containing a limited number of nodes, that is, ranging between 4 and 25. Each layer in a GNN, by design, corresponds to the aggregation of information from the neighboring nodes, which is one edge away (1-hop). With small-scale graphs, only a few layers are often sufficient to incorporate the entirety of the graph data. Conversely, incorporating an excessive number of layers in a GNN can potentially lead to an undesirable effect known as oversmoothing. This is a circumstance in which the features of all nodes become overly homogeneous, subsequently impairing the performance of the model.

In terms of activation function, we employed the Leaky Rectified Linear Unit (LeakyReLU) following graph convolution layers. This was then followed by the utilization of a max pooling layer to aggregate the node embeddings, subsequently applying a final linear layer. For the optimization process, a learning rate of 6.4×10^{-6} and a weight decay parameter of 1×10^{-6} were employed. These values were chosen to ensure efficient learning without compromising the stability of the model. All of the developed models in our study were constructed utilizing the 'PyTorch Geometric' framework [73], a powerful and efficient library designed to facilitate the implementation of graph-based DL models.

4.5.1 GCN

GCNs have gained significant attention in recent years due to their ability to learn representations of graph-structured data that are invariant with the input graph size and topology [74]. GCNs have been successfully applied to various domains, including social network analysis, molecular biology, recommendation (or recommender) systems and natural language processing. The goal of a GCN is to learn a function that maps the input features to a new representation, capturing the relationships among the vertices in the graph.

The core idea behind GCNs is to generalize the convolution operation from regular grids to irregular graphs. A graph convolution operation can be thought of as a local averaging of features from neighboring vertices, which captures both the local structure of the graph and the features associated with each vertex.

Given an input graph $G = (V, E)$, the graph convolution operation is defined as

$$H^{(l+1)} = \sigma\left(\hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right),$$

where $H^{(l)} \in \mathbb{R}^{N\times F_l}$ is the feature matrix at layer l, with N being the number of vertices in the graph, F_l the dimension of the feature space at layer l and $W^{(l)} \in \mathbb{R}^{F_l\times F_{l+1}}$ the learnable weight matrix at layer l. Furthermore, $\sigma(\cdot)$ denotes the activation function.

The matrix $\hat{A} \in \mathbb{R}^{N\times N}$ is the adjacency matrix of the input graph with added self-connections, defined as $\hat{A} = A + I_N$, where A is the adjacency matrix of G and I_N is the identity matrix of size N. The matrix $\hat{D} \in \mathbb{R}^{N\times N}$ is a diagonal matrix with $\hat{D}_i = \sum_j \hat{A}_{ij}$, representing the degree of vertex i in the graph with added self-connections.

The graph convolution operation can be interpreted as a message-passing mechanism, where each vertex aggregates information from its neighbors and updates its features according to the learned weights. This process is repeated for a number of layers, allowing the model to capture higher-order relationships between vertices in the graph.

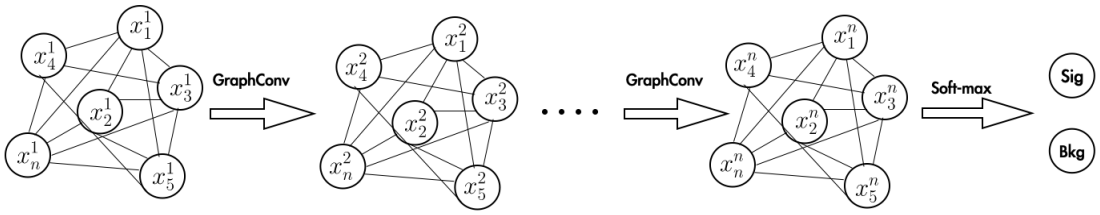

Figure 13: A schematic architecture for the used GCN model.

The results from our GCN analysis are presented in Fig. 14. The leftmost plot displays the ROC curve for our trained GCN model, showcasing an AUC value of 0.84. The middle plot demonstrates the model scores for both signal and background. On the rightmost plot, the CM is displayed where the signal and background diagonal entries are 0.83 and 0.76, respectively.

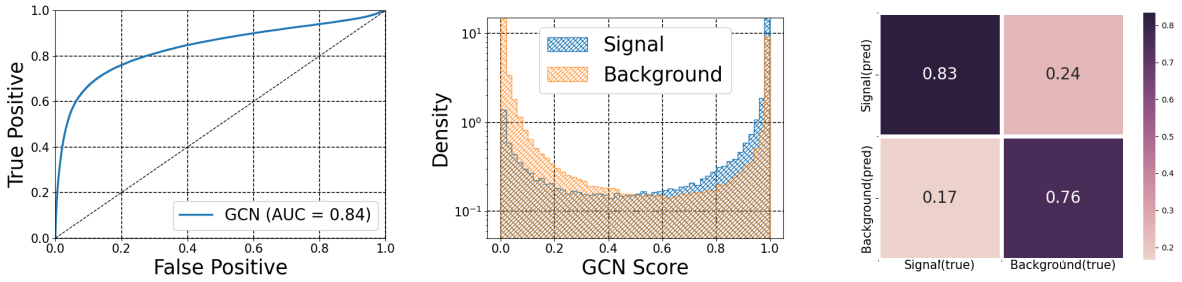

Figure 14: GCN test results when trained on kinematic distributions and jet images for signal events (BP with $m_A = 300$ GeV). The ROC curve (left), GCN output score (middle) and confusion matrix (right) are shown.

Despite the demonstrated effectiveness of the GCN model, it isn't without its constraints. These limitations have encouraged the exploration and establishment of multiple variants and extensions. One major constraint of the original GCN model is its incapability to handle inductive learning tasks. This inability to generalize to unseen graph structures or vertices is due to the GCN's reliance on the explicit adjacency matrix of the input graph, which makes adapting the model to new data a challenge.

Additionally, the model lacks flexibility when capturing a wide array of graph structures and applying filters of different spatial sizes, primarily because the conventional GCN model utilizes a fixed neighborhood aggregation scheme. It's also worth noting that GCNs treat all neighboring vertices with equal importance during the feature aggregation phase, which could be sub-optimal when some neighbors provide more significant information than others.

These shortcomings led to the development of diverse GCN variants, including GraphSAGE, Graph Attention Networks (GAT), and Dynamic Graph Convolutional Neural Networks (DGCNN). These adaptations address the challenges by incorporating inductive learning capabilities, applying spectral techniques, and/or introducing attention mechanisms, thereby extending the usability and enhancing the performance of graph-based deep learning models.

4.5.2 DGCNN

Unlike GCNs, which often assume that graphs are static in nature and hence fail to capture the dynamics of edges, in a DGCNN [84], the EdgeConv operation serves as the fundamental

building block. It considers edges as basic units of information propagation instead of nodes, which is especially beneficial in capturing local geometric structures and dealing with unordered point sets. DGCNNs extend the idea of CNNs to graphs, where each layer of the network operates on the nodes and edges of the graph. What makes DGCNNs unique is their ability to learn the importance of edges dynamically during training. Instead of relying on pre-defined edge weights (ΔR in our case), a DGCNN learns to assign weights to the edges of the graph based on their importance for the task at hand. This allows the network to adapt to different graphs and tasks more effectively.

For an edge e_{ij}, the EdgeConv operation is:

$$\mathbf{h}_{ij} = \Phi(\mathbf{v}_i, \mathbf{v}_j - \mathbf{v}_i).$$

In this equation, $\mathbf{v}i$ is the feature vector of the node i, Φ is a shared MLP applied to the concatenation of the feature vector of node i and the difference between the feature vectors of nodes j and i. The new feature of node i is then computed by aggregating the transformed features of all neighboring nodes:

$$\mathbf{v}'_i = \rho(\{\mathbf{h}_{ij} | \forall j \in N(i)\}),$$

where $N(i)$ denotes the neighborhood of node i and ρ is a symmetric function, such as max pooling or average pooling. The edge features are recomputed in every layer, allowing the graph structure to dynamically evolve based on the learned node features. This dynamic nature is a key advantage of the EdgeConv operation in DGCNNs.

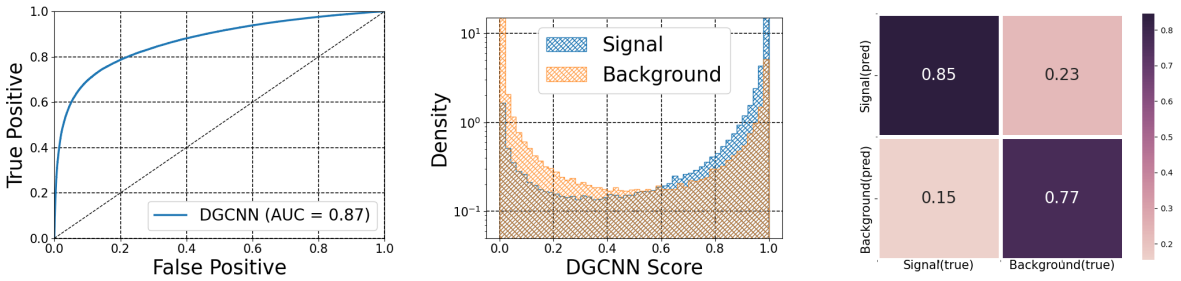

Figure 15: DGCNN test results when trained on kinematic distributions and jet images for signal events (BP with $m_A = 300$ GeV). The ROC curve (left), DGCNN output score (middle) and confusion matrix (right) are shown.

The results of the DGCNN analysis is shown in Fig. 15. This showcases an enhancement from the previous GCN model, with a reported AUC metric value of 0.87, signifying improved classification performance.

4.5.3 GraphSAGE

GraphSAGE is an inductive learning framework for graph-structured data that allows GCN to generalize to unseen graph structures or vertices [75]. Unlike traditional GCNs, which are transductive and rely on the explicit adjacency matrix of the entire input graph, a GraphSAGE structure learns to generate embeddings for individual vertices by sampling and aggregating features from their local neighborhoods.

The core idea behind GraphSAGE is to learn a function that generates vertex embeddings by aggregating features from a fixed-size local neighborhood, irrespective of the graph size or structure. To achieve this, GraphSAGE employs a two-step procedure: sampling and aggregation.

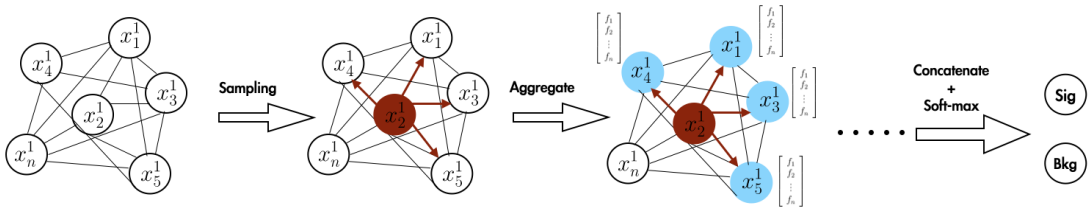

Figure 16: A schematic architecture for the used GraphSAGE model (for the node x_2). Here, dots indicates repeated sampling and aggregation for all graph nodes.

Sampling For each vertex v in the graph, GraphSAGE samples a fixed-size set of neighbors at different search depths. The sampling procedure is carried out for K iterations, where K is the number of layers in the model. In the k-th iteration, a fixed-size set of S_k neighbors is sampled uniformly at random from the k-hop neighborhood of v.

Aggregation After sampling the neighbors, GraphSAGE aggregates the features from the sampled neighborhood to generate the embeddings for each vertex. The aggregation function can be any differentiable and permutation-invariant function, such as the element-wise mean, Long Short-Term Memory (LSTM) or max pooling. The aggregation process is carried out in a hierarchical manner, starting from the outermost layer and moving towards the target vertex.

Given a vertex v, let $\mathcal{N}_k(v)$ denote the set of k-hop neighbors of v. The feature aggregation process in GraphSAGE can be formally defined as follows:

$$\mathbf{f}_v^{(k)} = \text{AGGREGATE}_k\left(\mathbf{f}_u^{(k-1)} : u \in \mathcal{N}_k(v)\right), \tag{16}$$

where $\mathbf{f}_v^{(k)}$ is the feature vector of vertex v at the k-th layer and $\text{AGGREGATE}_k(\cdot)$ is the aggregation function at layer k. After aggregating the features from all layers, the final embedding for vertex v is computed by concatenating the original feature vector $\mathbf{f}_v^{(0)}$ and the aggregated feature vector from the last layer $\mathbf{f}_v^{(K)}$:

$$\mathbf{f}'_v = \text{CONCAT}\left(\mathbf{f}_v^{(0)}, \mathbf{f}_v^{(K)}\right). \tag{17}$$

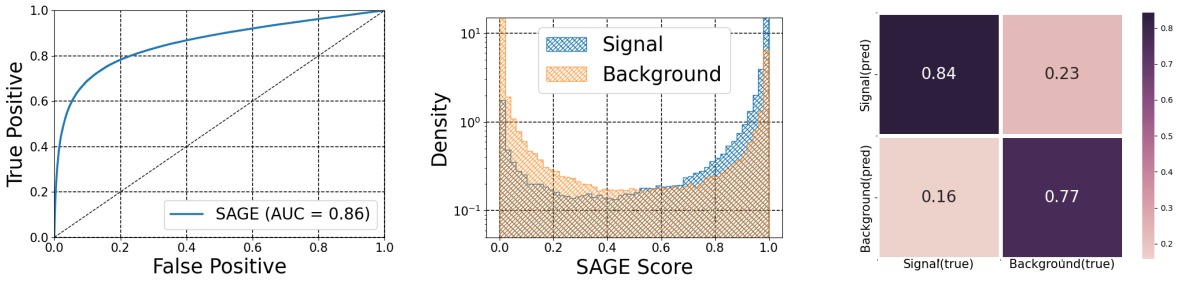

Figure 17: GraphSAGE test results when trained on kinematic distributions and jet images for signal events (BP with $m_A = 300$ GeV). The ROC curve (left), GraphSAGE output score (middle) and confusion matrix (right) are shown.

The outcomes from the GraphSAGE analysis are displayed in Fig. 17. It reveals a classification performance, measured by the ROC AUC of 0.86. While GraphSAGE is predominantly designed for larger graphs, and our dataset comprises mainly smaller graphs. Despite this, there's still an observed improvement in comparison to the GCN model.

The observed improvement with GraphSAGE on smaller graphs could possibly be attributed to its distinctive feature aggregation method. Unlike GCN, which heavily relies on the graph's global structure, GraphSAGE employs a more flexible, inductive approach, aggregating features from the local neighborhood of each node. As a result, even with smaller graphs, GraphSAGE can extract meaningful, context-rich features leading to enhanced performance. Also, GraphSAGE's sample-based training method helps to capture and generalize even subtle patterns present in smaller graphs.

4.5.4 GAT

A GAT is a variant of the GCN that incorporate the attention mechanism to adaptively weigh the importance of neighboring vertices during the feature aggregation step [76]. By assigning different weights to neighbors based on their relative importance, GATs are able to learn more expressive and flexible graph representations compared to standard GCNs.

The attention mechanism in GAT is designed to compute a pair-wise attention coefficient between any two connected vertices, which is used to weigh the contribution of neighboring features during the aggregation step. Formally, the attention coefficients for a vertex i and its neighbor j can be defined as:

$$e_{ij} = \text{LeakyReLU}\left(\mathbf{a}^\top \left[\mathbf{W}\mathbf{h}_i \oplus \mathbf{W}\mathbf{h}_j\right]\right), \tag{18}$$

where $\mathbf{h}_i$ and $\mathbf{h}_j$ are the feature vectors of vertices i and j, respectively, $\mathbf{W} \in \mathbb{R}^{F' \times F}$ is a shared weight matrix that projects the input features onto a higher-dimensional space, $\oplus$ denotes concatenation, $\mathbf{a} \in \mathbb{R}^{2F'}$ is a learnable attention vector and LeakyReLU is used as the activation function. To ensure that the attention coefficients are invariant to the order of vertices, the attention mechanism is made symmetric by considering the concatenation of the transformed feature vectors for both vertices. The attention coefficients are then normalized using the softmax function to obtain the final attention weights:

$$\alpha_{ij} = \text{softmax}j(e_{ij}) = \frac{\exp(e_{ij})}{\sum_{k \in \mathcal{N}_i} \exp(e_{ik})}, \tag{19}$$

where $\mathcal{N}_i$ is the set of neighboring vertices of vertex i.

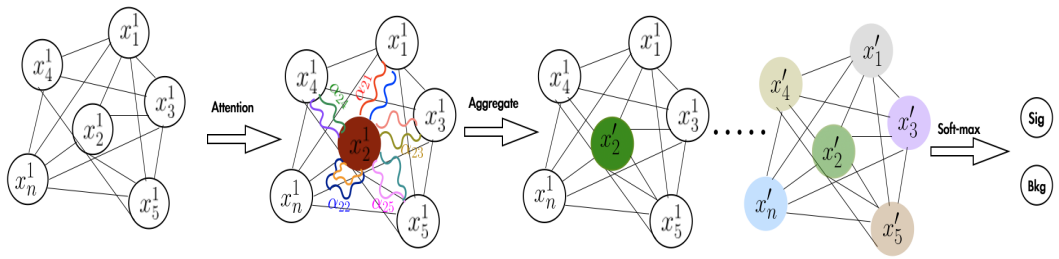

Figure 18: A schematic architecture for the used GAT model (for the node x_2). Here, wavy lines illustrate the self and neighbours attention corresponding to the node while different colors denote independent attention computations (attention heads). The aggregated features from each head are averaged to obtain x'_i. Also, α_{ij} are the attention weights.

With the attention weights computed, the next step in GAT is to aggregate the features from neighboring vertices. The feature aggregation can be expressed as a linear combination of the transformed features of the neighbors, weighted by the attention coefficients:

$$\mathbf{h}i' = \sigma\left(\sum j \in \mathcal{N}_i \alpha_{ij} \mathbf{W}\mathbf{h}_j\right), \tag{20}$$

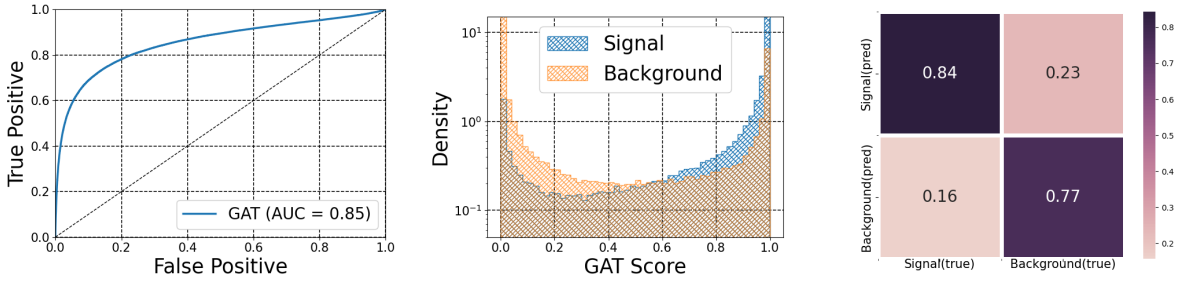

Figure 19: GAT test results when trained on kinematic distributions and jet images for signal events (BP with $m_A = 300$ GeV). The ROC curve (left), GAT output score (middle) and confusion matrix (right) are shown.

where $\sigma(\cdot)$ is the activation function.

The analysis of the GAT is depicted in Fig. 19, with the classification performance, as expressed through AUC metric, returning a value of 0.85. In our training process, we employed four attention heads. This setup allows the model to capture different types of relationships and multi-dimensional information from neighboring nodes, enhancing its representation learning capability. GAT's improvement over GCN could be attributed to its capability to weigh the importance of neighboring nodes differently when aggregating information.

5 Similarity of DNN hidden layers representations

DL models are treated as black boxes which predictions are very hard to explain according to the learned information in the hidden layers. Recently, there have been proposed interesting methods trying to explain the predictions of DL models [77, 78]. They assume that the contribution of a feature can be determined by measuring how the prediction score changes when the feature is altered. Although the proposed method can explain the change in the prediction score among different types of input features, it does not give a clear insight about what is the learned representation for each hidden layer. The challenge in analyzing the hidden layers representations of NNs is that features are distributed across a large number of neurons. Moreover, hidden layers do not have fixed size of neurons. Linear CKA [79] addresses these challenges, enabling quantitative comparisons of representations within and across networks.

To compute the similarity of the hidden layers representations, CKA takes as an input the hidden layers activation matrices as $X \in \mathbb{R}^{d\times P_1}$ and $Y \in \mathbb{R}^{d\times P_2}$ with P_1 and P_2 being the neurons in the different hidden layers evaluated on the same input set with size equals to d. The CKA similarity is defined as

$$\mathrm{CKA}(M,N) = \frac{\mathrm{HSIC}(M,N)}{\sqrt{\mathrm{HSIC}(M,M)\mathrm{HSIC}(N,N)}}\,, \tag{21}$$

where $M = XX^\top$ and $N = YY^\top$[7] denote the Gram matrices for the two hidden layers. The main advantage of having a Gram matrix is that we can compute the similarity of hidden layers representations with different number of neurons as the Gram matrix always has the dimension of $d \times d$. Moreover, one can compute the CKA similarity for hidden layers from different DNNs.

[7] As we compute the linear CKA then M is simply $XX^\top$ while, for kernel CKA, M can be computed as $\Phi(X,X)$ with Φ being the used kernel function.

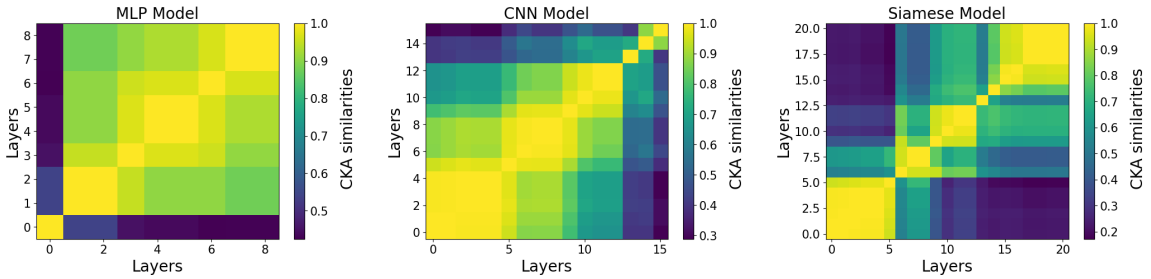

Figure 20: CKA similarities for the MLP (left), CNN (middle) and SNN (right) models for the signal (BP with $m_A = 300$ GeV) using 1000 test events are shown. Layers include the input, fully-connected, convolution, pooling and dropout layers but exclude the output layer.

The Hilbert Schmidt Independent Criterion (HSIC) is defined as

$$\text{HSIC}(M, N) = \frac{1}{(d-1)^2}\, tr(MHNH)\,, \tag{22}$$

with H is a centering matrix and M, N are defined above. It is worth mentioning that the HSIC is not invariant to random scaling of the input features, but it can be made invariant through a normalization as introduced in the CKA formula. The value of a CKA similarity ranges between $[0, 1]$ and indicates the similarity of the learned representations by each hidden layer. Layers with small CKA values do not share the same representations and they learn different information from the input data which improves the model classification performance. A larger CKA value indicates that layers learn the same information about the input features resulting in no improvement of the classification accuracy of the model. In this case one can truncate those layers which share the same information to reduce the model complexity with no impact on the classification performance. For illustration of the relation between the CKA similarity of the hidden layers and the classification accuracy we point out to Fig. 3 in [79]. Here, Fig. 20 shows the CKA similarity for three DNN models: MLP (left plot), CNN (middle plot) and SNN (right plot). The models are trained on the signal point with $m_A = 300$ GeV and to compute the CKA we adopt 1000 test samples for each model. The CKA value is then computed for all layers of each model, e.g., Conv2d, FC, dropout, pooling and input layers, but we do not include the final output layer (the two neurons layer with softmax activation). The MLP model shows a uniform similarity distribution among all layers except the input layer. The uniform similarity in the MLP model indicates that the model is able to capture global information only. In fact, such a behavior of the MLP layers is expected as the input features are high-level kinematic distributions which encode the global characteristic features of the signal and background events. Moreover, the uniform similarity among the MLP hidden layers indicates that one can reduce the number of the used layers with no significant reduction of the classification accuracy.

The CNN model shows a large CKA similarity among each convolution layer pair. The first two convolution layers and the pooling layer have large CKA similarities. Also, the second and third pair of the convolution layers as well as the pooling layers have large CKA similarities among themselves. The last layers, $14^{\text{th}} - 16^{\text{th}}$, are the fully-connected layers which have similar representation among themselves but are different from the convolution layers. In general, all convolution layers share a CKA similarity ~ 0.8 among each other which indicates that they all capture specific local information encoded into the jet images.

As for the SNN model, when tested on the same CNN input, we notice that, while the

first two convolution layers and the pooling layers have large CKA value as in the CNN model, the other convolution layers have small CKA similarity to the first convolution layers. The last layers, $15^{\text{th}} - 20^{\text{th}}$, are the fully-connected and dropout layers. The fact that the first couple of convolution layers do not share the same similarity to the later convolution layers indicates that convolution layers in the SNN capture different information from the input jet images. Indeed, the additional information captured by SNN layers is the reason for the enhanced classification accuracy over the CNN one. Overall, the CKA similarity for SNN hidden layers assures that the convolution layers do not capture only local information (similar to the CNN layers) but also capture different types of information, similarity and dissimilarity of the input images.

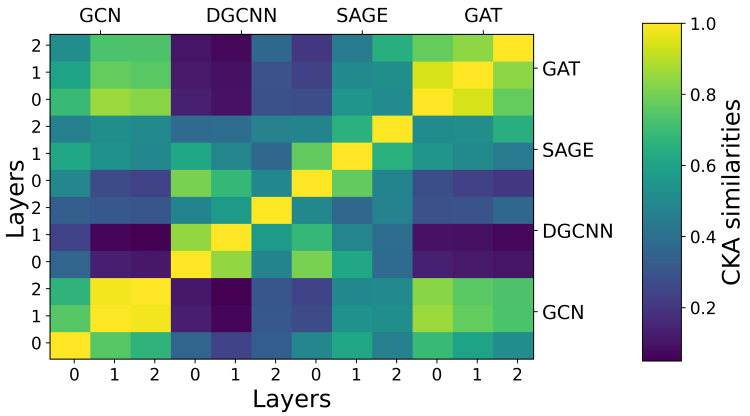

Figure 21: CKA Similarity for GNN models. The CKA values are computed for the layers of each model as well as layers from different models.

The power of the CKA approach lies in its ability to compute the similarity between hidden layers from different models. As seen in Fig. 21, we demonstrate the CKA similarity across all hidden layers of the GNN models. These similarities are drawn both within the hidden layers of each individual model and between the hidden layers from different GNN models.

An interesting observation is that the hidden layers from the DGCNN and GraphSAGE models demonstrate a relatively high similarity, approximately 0.7, but are distinctly different from the GCN and GAT models. This similarity reflects the classification accuracy values, with DGCNN and GraphSAGE displaying nearly identical classification accuracy across all mass points, as shown in Fig. 22.

6 Results

We now apply the different ML models to probe the $l^+l^-b\bar{b}$ ($l = e, \mu$) signature of the $pp \to A \to Z^{(*)}h$ process at Run 3 of the LHC (with an integrated luminosity of 300 fb^{-1}) and the HL-LHC (with an integrated luminosity of 3000 fb^{-1}). The discrimination power of each of the networks measures how well the signal and background features are recognized, which is quantified by the ROC curve. The better discrimination performance between signal and backgrounds, the higher the true positive rate than the false positive rate in the ROC curve. Detailed information about the remaining number of signal and background events after optimizing the cuts on the DNN output for all the considered signal points can be found in Tab. 2.

The expected upper limit on the total cross section can be constructed by computing

the probability of finding the expected data incompatible with the prediction of the various ML models. The expectation value of having a certain number of events in the i^{th} m_A bin in the DNN output score distribution is [80]

$$E = \mu S_i + B_i \,, \tag{23}$$

where S_i and B_i are the number of signal and background events, respectively, and μ is the signal strength parameter. The signal strength parameter defines the type of statistic measure, so that $\mu = 1$ is rejecting a signal discovery hypothesis and defining the upper limit on the total cross section. Such a limit can be calculated from the optimization of the signal-to-background cut on the DNN output and this has been done using the following significance formula [81–83]:

$$\sigma_{sys} = \left[2 \left((N_s + N_b) \ln \frac{(N_s + N_b)(N_b + \sigma_b^2)}{N_b^2 + (N_s + N_b)\sigma_b^2} - \frac{N_b^2}{\sigma_b^2} \ln(1 + \frac{\sigma_b^2 N_s}{N_b(N_b + \sigma_b^2)}) \right) \right]^{1/2} , \tag{24}$$

with N_s and N_b being the number of signal and background events, respectively, and where σ_b is the total uncertainty in the background events. For a 95% Confidence Level (C.L.) upper limit on the total cross section for $\sigma(pp \to A \to Z^{(*)}h) \times Br(h \to \bar{b}b)$, we require the signal significance to be $\sigma_{sys} \leq 2$ [81]. The corresponding results for all considered ML models are shown in Fig. 22. The CMS and ATLAS bounds extracted from [34] (Fig. 5) and [38] (Fig. 11), respectively, linearly scaled to the considered integrated luminosities, are also presented[8].

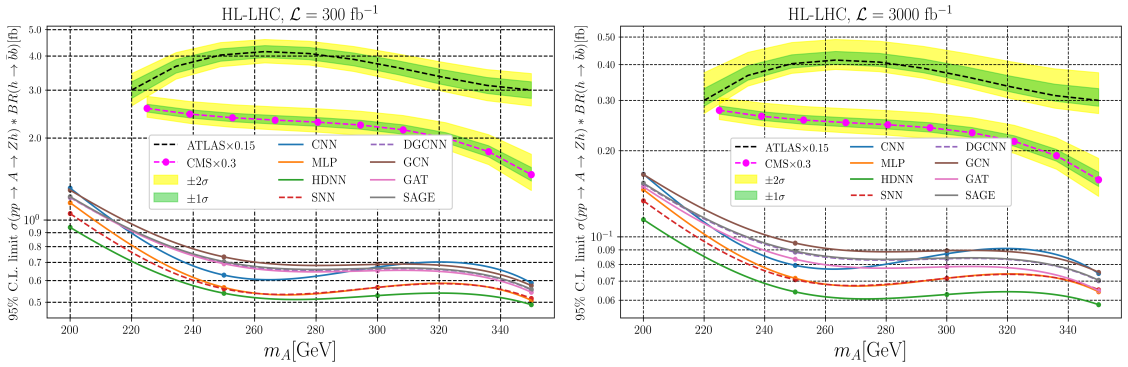

Figure 22: 95% C.L. upper limit on the total cross section for the process $\sigma(pp \to A \to Zh) \times Br(h \to \bar{b}b)$ at Run 3 of the LHC with $L_{\text{int}} = 300$ fb^{-1} (left) and the HL-LHC with $L_{\text{int}} = 3000$ fb^{-1} (right), both with $\sqrt{s} = 14$ TeV and assuming a systematic uncertainty $\sigma_b = 10\%$ on the background. The expected CMS and ATLAS limits extracted from [34] (Fig. 5) and [38] (Fig. 11), respectively, and linearly scaled to the two luminosities are also given. Bullet points on the ML curves indicate the four BPs considered.

The overarching observation here is that the results obtained from all our ML approaches systematically outperform the experimental results for both LHC configurations. Amongst the former, as obviously seen, the HDNN with two stream inputs has the most stringent limit amongst all networks. Indeed, this is expected as merging both global and local information by concatenating an MLP and CNN enables the model to access all needed information

[8] It is worth mentioning here that the ATLAS analysis considers a combined limit between the channels of two isolated leptons plus no leptons. If, for consistency with our analysis, one considered only the ATLAS results from the channel with two isolated leptons channel, the experimental limit in Fig. 22 will be relaxed.

about the events which in turns enhances the classification performance. Interestingly, the SNN has an equal, and even better, classification performance than the MLP network. Although the SNN is trained on jet images, which encode only local information about the event, i.e., radiation patterns of the charged hadrons, it has more stringent limits than the MLP network, which is trained on kinematic distributions encoding only global information. We interpret this as follows: that learning the similarity among events from the same class can secretly provide "unseen" global information to the SNN model. This can be achieved by mapping events from different classes, signal and background, into different locations in the Euclidean latent space of the model during the first training stage. The newly learned "unseen" global information can be clearly interpreted from the CKA of SNN where the early convolution layers capture information different to the ones captured by the late convolution ones. Moreover, GNNs (DGCNN, GCN, GAT and GraphSAGE), produce a slightly worse performance than CNNs. This is because they analyze low level data, i.e., (reconstructed) four-momenta of the final state particles. To increase the classification performance of GNN models, we would need to add more inductive biases to be tuned. In fact, the size of the constructed graphs is small, which is due to the small number of (reconstructed) final state particles in each event. Accordingly, if we increased the number of the inductive biases, e.g., by increasing the number of GNN layers, the GNN model would overfit.

7 Conclusions

In summary, in this paper, we have used a variety of advanced ML techniques, most of which had never been applied previously to the study of collider processes, to prove that they can offer a significant improvement with respect to traditional LHC analyses, using either a cut-and-count approach or else based on traditional ML approaches, such as (shallow) NNs. Our methodology, in fact, exploits instead DNNs, the latter covering MLP, CNN, SNN and HDNN algorithms and a variety of GNNs (DGCNNs, GCNs, GAT and GraphSAGE networks), including an interpretable ML element that gives us confidence in the accuracy of our predictions.

In order to exemplify the scope of this multi-prong ML approach, we have targeted a BSM process to which current experimental analyses by ATLAS and CMS have moderate sensitivity, i.e., $b\bar{b}, gg \to A \to Z^{(*)}h \to l^+l^-b\bar{b}$ ($l = e, \mu$), involving the production and decay of a CP-odd Higgs state of a 2HDM Type-II (A) and the SM-like Higgs discovered in 2012 (h). The CERN machine configurations adopted included Run 3 of the LHC as well as the HL-LHC, wherein $\sqrt{s} = 14$ TeV for both and $L_{\rm int} = 300$ and 3000 fb^{-1}, respectively. This process is rather intriguing, as it is a potential BSM signal that would prove simultaneously the presence of an extended Higgs sector (with a different CP nature) with respect to the SM one and its underlying gauge structure. However, such a BSM signal suffers from overwhelming backgrounds, so that it is a significant challenge to extract it. Furthermore, depending on the A mass, whether it is such that $m_A < m_Z + m_h$, $m_Z + m_h < m_A < 2m_t$ or $m_A \approx 2m_t$, both the size of the signal (via the $Br(A \to Z^{(*)})$, with the gauge boson being either off- or on-shell as m_A increases) and the composition of the background samples vary significantly, so that different kinematical selections are generally required to optimise the sensitivity of the various searches therein.

By adopting standard acceptance cuts on the final state objects (leptons and hadrons) and a rather bland selection around the $Z^{(*)}$ and h masses, so long that these are supplemented by a combination of the aforementioned ML tools, we were able to improve, in comparison to the very latest ATLAS and CMS results, the sensitivity to the cross section

of the signal process by at least a factor of 4(2) over the $m_Z + m_h < m_A < 2m_t (m_A \approx 2m_t)$ region while at the same time proving that there can be sensitivity also over the so-far unexplored $m_A < m_Z + m_h$ interval. Finally, we find that CNN approaches generally outperform GNN ones.

Table 2: Analysis summary for all DNN models considered (column 1) for all signal BPs from Tab. 1, identified by the A mass (column 2). Column 3 shows the area under the ROC curve in each case. Columns 5 and 6 show the number of remaining signal and background events, respectively, after maximizing the cut on the DNN output. Column 6 shows the final significance. For illustrative purposes, event rates and significances are here computed at the luminosity mid point of 1000 fb^{-1}.

	BPs	AUC	Signal (S)	Background (B)	σ
MLP	$m_A = 200$ GeV	0.90	6538	486	156
	$m_A = 250$ GeV	0.92	46894	1422	496
	$m_A = 300$ GeV	0.95	63060	1287	614
	$m_A = 350$ GeV	0.96	69496	1004	678
CNN	$m_A = 200$ GeV	0.87	9253	2007	142
	$m_A = 250$ GeV	0.89	47122	2675	443
	$m_A = 300$ GeV	0.86	60114	4417	485
	$m_A = 350$ GeV	0.90	66320	2364	574
SNN	$m_A = 200$ GeV	0.92	7267	441	171
	$m_A = 250$ GeV	0.94	45822	1135	507
	$m_A = 300$ GeV	0.95	61940	1210	612
	$m_A = 350$ GeV	0.96	69894	2843	672
HDNN	$m_A = 200$ GeV	0.93	7810	328	191
	$m_A = 250$ GeV	0.95	44033	767	525
	$m_A = 300$ GeV	0.98	63243	784	661
	$m_A = 350$ GeV	0.97	69798	956	685
DGCNN	$m_A = 200$ GeV	0.86	6973	722	150
	$m_A = 250$ GeV	0.83	39126	1958	414
	$m_A = 300$ GeV	0.87	51436	1362	532
	$m_A = 350$ GeV	0.91	63463	1131	608
GCN	$m_A = 200$ GeV	0.84	7256	992	143
	$m_A = 250$ GeV	0.78	38242	2083	403
	$m_A = 300$ GeV	0.84	50154	1416	520
	$m_A = 350$ GeV	0.89	61510	1051	620
GAT	$m_A = 200$ GeV	0.85	6904	745	147
	$m_A = 250$ GeV	0.82	38459	1894	412
	$m_A = 300$ GeV	0.85	51237	1407	528
	$m_A = 350$ GeV	0.9	63607	1223	622
GraphSAGE	$m_A = 200$ GeV	0.85	6915	723	148
	$m_A = 250$ GeV	0.82	38993	1994	412
	$m_A = 300$ GeV	0.86	51284	1392	529
	$m_A = 350$ GeV	0.9	63655	1207	624

Acknowledgments

AH thanks Mihoko Nojiri for the fruitful discussion about the SSN. AH is funded by the grant NRF-2021R1A2C4002551. SM is supported in part through the NExT Institute and the STFC Consolidated Grant No. ST/L000296/1.

References

[1] ATLAS Collaboration, Phys. Lett. B **716** (2012) 1.

[2] CMS Collaboration, Phys. Lett. B **716** (2012) 30.

[3] G. C. Branco, P. M. Ferreira, L. Lavoura, M. N. Rebelo, M. Sher and J. P. Silva, Phys. Rept. **516** (2012), 1-102 [arXiv:1106.0034 [hep-ph]].

[4] M. Misiak and M. Steinhauser, Eur. Phys. J. C **77**, no.3, 201 (2017) [arXiv:1702.04571 [hep-ph]].

[5] P. M. Ferreira, J. F. Gunion, H. E. Haber and R. Santos, Phys. Rev. D **89**, no.11, 115003 (2014) [arXiv:1403.4736 [hep-ph]].

[6] J. Bernon, J. F. Gunion, H. E. Haber, Y. Jiang and S. Kraml, Phys. Rev. D **93**, 035027 (2016).

[7] P. Basler, P. M. Ferreira, M. Mühlleitner and R. Santos, Phys. Rev. D **97**, 095024 (2018).

[8] P. M. Ferreira, S. Liebler and J. Wittbrodt, Phys. Rev. D **97**, 055008 (2018).

[9] E. Accomando, C. Byers, D. Englert, J. Hays and S. Moretti, Phys. Rev. D **105**, no.11, 115004 (2022)

[10] E. Accomando, D. Englert, C. Byers, J. Hays and S. Moretti, [arXiv:1905.07313 [hep-ph]].

[11] E. Accomando, M. Chapman, A. Maury and S. Moretti, Phys. Lett. B **818**, 136342 (2021) [arXiv:2002.07038 [hep-ph]].

[12] A. G. Akeroyd, S. Alanazi and S. Moretti, [arXiv:2301.00728 [hep-ph]].

[13] T. D. Lee, Phys. Rev. D **8** (1973), 1226-1239

[14] S. L. Glashow and S. Weinberg, Phys. Rev. D **15** (1977), 1958

[15] I. F. Ginzburg and M. Krawczyk, Phys. Rev. D **72** (2005), 115013 [arXiv:hep-ph/0408011 [hep-ph]].

[16] S. Antusch, O. Fischer, A. Hammad and C. Scherb, JHEP **03** (2021), 200 [arXiv:2011.10388 [hep-ph]].

[17] S. Antusch, O. Fischer, A. Hammad and C. Scherb, JHEP **08** (2022), 224 [arXiv:2112.00921 [hep-ph]].

[18] I. P. Ivanov and J. P. Silva, Phys. Rev. D **92** (2015) no.5, 055017 [arXiv:1507.05100 [hep-ph]].

[19] M. Baak, M. Goebel, J. Haller, A. Hoecker, D. Kennedy, R. Kogler, K. Moenig, M. Schott and J. Stelzer, Eur. Phys. J. C **72** (2012), 2205 [arXiv:1209.2716 [hep-ph]].

[20] S. Chatrchyan *et al.* [CMS], Phys. Lett. B **716** (2012), 30-61 [arXiv:1207.7235 [hep-ex]].

[21] G. Aad *et al.* [ATLAS and CMS], JHEP **08** (2016), 045 [arXiv:1606.02266 [hep-ex]].

[22] G. Aad *et al.* [ATLAS and CMS], Phys. Rev. Lett. **114** (2015), 191803 [arXiv:1503.07589 [hep-ex]].

[23] G. Aad *et al.* [ATLAS], Phys. Lett. B **716** (2012), 1-29 [arXiv:1207.7214 [hep-ex]].

[24] [LEP Higgs Working Group for Higgs boson searches], [arXiv:hep-ex/0107034 [hep-ex]].

[25] T. Aaltonen *et al.* [CDF], Phys. Rev. Lett. **102** (2009), 021802 [arXiv:0809.3930 [hep-ex]].

[26] G. Aad *et al.* [ATLAS], Phys. Lett. B **801** (2020), 135148 [arXiv:1909.10235 [hep-ex]].

[27] A. M. Sirunyan *et al.* [CMS], JHEP **03** (2020), 034 [arXiv:1912.01594 [hep-ex]].

[28] Y. S. Amhis *et al.* [HFLAV], Eur. Phys. J. C **81** (2021) no.3, 226 [arXiv:1909.12524 [hep-ex]].

[29] A. Hammad, M. Park, R. Ramos and P. Saha, [arXiv:2207.09959 [hep-ph]].

[30] M. Gustafsson, PoS **CHARGED2010** (2010), 030 [arXiv:1106.1719 [hep-ph]].

[31] T. Enomoto and R. Watanabe, JHEP **05** (2016), 002 [arXiv:1511.05066 [hep-ph]].

[32] A. Djouadi, J. Ellis, A. Popov and J. Quevillon, JHEP **03** (2019), 119 [arXiv:1901.03417 [hep-ph]].

[33] S. Jung, J. Song and Y. W. Yoon, JHEP **05** (2016), 009 [arXiv:1601.00006 [hep-ph]].

[34] A. M. Sirunyan *et al.* [CMS Collaboration], Eur. Phys. J. C **79** (2019) no.7, 564 [arXiv:1903.00941 [hep-ex]].

[35] A. M. Sirunyan *et al.* [CMS Collaboration], JHEP **03** (2020), 065 [arXiv:1910.11634 [hep-ex]].

[36] M. Aaboud *et al.* [ATLAS Collaboration], JHEP **03** (2018), 174 [erratum: JHEP **11** (2018), 051] [arXiv:1712.06518 [hep-ex]].

[37] M. Aaboud *et al.* [ATLAS Collaboration], ATLAS-CONF-2020-043.

[38] M. Aaboud *et al.* [ATLAS Collaboration], ATLAS-CONF-2016-015.

[39] F. Maltoni, K. Paul, T. Stelzer and S. Willenbrock, Phys. Rev. D64 (2001), 094023 [arXiv:hep-ph/0106293 [hep-ph]].

[40] M. Cacciari, G. P. Salam and G. Soyez, JHEP **04** (2008), 063 doi:10.1088/1126-6708/2008/04/063 [arXiv:0802.1189 [hep-ph]].

[41] Y. L. Dokshitzer, G. D. Leder, S. Moretti and B. R. Webber, JHEP **08** (1997), 001 doi:10.1088/1126-6708/1997/08/001 [arXiv:hep-ph/9707323 [hep-ph]].

[42] M. Wobisch and T. Wengler, [arXiv:hep-ph/9907280 [hep-ph]].

[43] J. Alwall, R. Frederix, S. Frixione, V. Hirschi, F. Maltoni, O. Mattelaer, H. S. Shao, T. Stelzer, P. Torrielli and M. Zaro, JHEP **07** (2014), 079 [arXiv:1405.0301 [hep-ph]].

[44] A. Belyaev, N. D. Christensen and A. Pukhov, Comput. Phys. Commun. **184** (2013), 1729-1769 [arXiv:1207.6082 [hep-ph]].

[45] W. Porod and F. Staub, Comput. Phys. Commun. **183** (2012), 2458-2469 [arXiv:1104.1573 [hep-ph]].

[46] W. Porod, Comput. Phys. Commun. **153** (2003), 275-315 [arXiv:hep-ph/0301101 [hep-ph]].

[47] T. Sjostrand, S. Mrenna and P. Z. Skands, JHEP **05** (2006), 026 [arXiv:hep-ph/0603175 [hep-ph]].

[48] M. Cacciari, G. P. Salam and G. Soyez, Eur. Phys. J. C **72** (2012), 1896 [arXiv:1111.6097 [hep-ph]].

[49] J. de Favereau *et al.* [DELPHES 3], JHEP **02** (2014), 057 [arXiv:1307.6346 [hep-ex]].

[50] A. Hocker, P. Speckmayer, J. Stelzer, J. Therhaag, E. von Toerne, H. Voss, M. Backes, T. Carli, O. Cohen and A. Christov, *et al.* [arXiv:physics/0703039 [physics.data-an]].

[51] P. T. Komiske, E. M. Metodiev and M. D. Schwartz, JHEP **01** (2017), 110 [arXiv:1612.01551 [hep-ph]].

[52] K. Fraser and M. D. Schwartz, JHEP **10** (2018), 093 [arXiv:1803.08066 [hep-ph]].

[53] J. Cogan, M. Kagan, E. Strauss and A. Schwarztman, JHEP **02** (2015), 118 [arXiv:1407.5675 [hep-ph]].

[54] L. de Oliveira, M. Kagan, L. Mackey, B. Nachman and A. Schwartzman, JHEP **07** (2016), 069 [arXiv:1511.05190 [hep-ph]].

[55] Sumit Chopra, Raia Hadsell and Yann LeCun, 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05).

[56] W. Blokland, P. Ramuhalli, C. Peters, Y. Yucesan, A. Zhukov, M. Schram, K. Rajput and T. Jeske, Phys. Rev. Accel. Beams **25** (2022) no.12, 122802 [arXiv:2110.12006 [physics.acc-ph]].

[57] J. H. Kim, M. Kim, K. Kong, K. T. Matchev and M. Park, JHEP **09** (2019), 047 [arXiv:1904.08549 [hep-ph]].

[58] L. Huang, S. b. Kang, J. H. Kim, K. Kong and J. S. Pi, JHEP **08** (2022), 114 [arXiv:2203.11951 [hep-ph]].

[59] T. Flacke, J. H. Kim, M. Kunkel, P. Ko, J. S. Pi, W. Porod and L. Schwarze, [arXiv:2304.09195 [hep-ph]].

[60] A. Hammad, S. Khalil and S. Moretti, Phys. Rev. D **107** (2023) no.7, 075027 [arXiv:2208.10133 [hep-ph]].

[61] Maithra Raghu, Thomas Unterthiner, Simon Kornblith, Chiyuan Zhang and Alexey, [arXiv:2108.08810 [cs.CV]].

[62] Ilya Loshchilov and Frank Hutter, [arXiv:1711.05101 [cs.LG]].

[63] Mangal, Ankita and Holm, Elizabeth A, arXiv:1804.09604 [stat.ML].

[64] T. Buss, B. M. Dillon, T. Finke, M. Krämer, A. Morandini, A. Mück, I. Oleksiyuk and T. Plehn, [arXiv:2202.00686 [hep-ph]].

[65] C. K. Khosa and S. Marzani, Phys. Rev. D **104** (2021) no.5, 055043 [arXiv:2105.03989 [hep-ph]].

[66] A. Hammad and M. Park, [arXiv:2209.03898 [hep-ph]].

[67] Jane Bromley, James W. Bentz and etal, International journal of pattern recognition and artificial intelligence,1993

[68] B. M. Dillon, G. Kasieczka, H. Olischlager, T. Plehn, P. Sorrenson and L. Vogel, SciPost Phys. **12** (2022) no.6, 188 [arXiv:2108.04253 [hep-ph]].

[69] B. M. Dillon, L. Favaro, F. Feiden, T. Modak and T. Plehn, [arXiv:2301.04660 [hep-ph]].

[70] Khosla Prannay , Teterwak Piotr, Wang Chen, Sarna Aaron and etal, Advances in neural information processing systems, arXiv:2004.11362 [cs.LG]

[71] Zhang Junbo and Ma Kaisheng, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, arXiv:2206.00227 [cs.CV]

[72] Jing Li, Vincent Pascal, LeCun Yann and Tian Yuandong, arXiv preprint arXiv:2110.09348

[73] Matthias Fey and Jan Eric Lenssen. Fast graph representation learning with PyTorch Geometric. *arXiv preprint arXiv:1903.02428*, 2019.

[74] Thomas N. Kipf and Max Welling, *Semi-Supervised Classification with Graph Convolutional Networks*, Proceedings of the International Conference on Learning Representations (ICLR), 2017.

[75] William L. Hamilton, Rex Ying, and Jure Leskovec, *Inductive Representation Learning on Large Graphs*, Advances in Neural Information Processing Systems (NeurIPS), pp. 1024-1034, 2017.

[76] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio, *Graph Attention Networks*, Proceedings of the International Conference on Learning Representations (ICLR), 2018.

[77] Barda A.J., Horvat, Hochheiser https://doi.org/10.1186/s12911-020-01276-x

[78] Rudin and Cynthia, Nature machine intelligence, arXiv:1811.10154 [stat.ML]

[79] Kornblith Simon, Norouzi Mohammad, Lee Honglak and Hinton Geoffrey, International Conference on Machine Learning arXiv:1905.00414 [cs.LG]

[80] G. Cowan, K. Cranmer, E. Gross and O. Vitells, Eur. Phys. J. C **71** (2011), 1554 [erratum: Eur. Phys. J. C **73** (2013), 2501] [arXiv:1007.1727 [physics.data-an]].

[81] T. Abe *et al.* [LHC Dark Matter Working Group], Phys. Dark Univ. **27** (2020), 100351 [arXiv:1810.09420 [hep-ex]].

[82] S. Antusch, E. Cazzato, O. Fischer, A. Hammad and K. Wang, JHEP **10** (2018), 067 [arXiv:1805.11400 [hep-ph]].

[83] S. Antusch, A. Hammad and A. Rashed, Phys. Lett. B **810** (2020), 135796 [arXiv:2003.11091 [hep-ph]].

[84] Wang, Yue and Sun, Yongbin and Liu, Ziwei and Sarma, Sanjay E. and Bronstein, Michael M. and Solomon, Justin M., *Dynamic Graph CNN for Learning on Point Clouds*, ACM Trans. Graph., vol. 38, no. 5, Article 146, October 2019.