This Report is part of a project that has received funding from the **European Union's Horizon 2020 research and innovation programme under grant agreement Nº675440**

AMVA4NewPhysics ITN

Work Package 1 - Deliverable 1.1

Multivariate Analysis Methods for Higgs Boson Searches At The Large Hadron Collider

AMVA4NewPhysics authors

February 27, 2017

Abstract

This document describes the studies performed to select the most performant tools for the classification and regression problems arising in the search for $H \to b\bar{b}$ and $H \to \tau\tau$ decays in Higgs pair production processes at the LHC.

Contents

1 Introduction

The 2012 discovery of the Higgs boson by the ATLAS and CMS collaborations opens up a new era for particle physics. The characterisation of that particle, the comparison of its measured properties to theory predictions, and the search for non-standard-model effects involving its phenomenology are now the highest priority of the LHC experiments.

Among the most important open questions, the measurement of the Higgs coupling parameters is of paramount importance. Only through a detailed comparison of predicted and measured couplings of the new-found particle to all matter and interaction fields can we ascertain its true nature, and verify whether new physics hides in the newly opened Higgs sector.

One of the Higgs coupling parameters, the "self-coupling" λ parameter, can best be measured by studying the process of Higgs boson pair-production. We focus our attention to that process in the studies within AMVA4NewPhysics. In particular, members from INFN and Oxford concentrate on the $hh \to b\,\bar{b}\,b\,\bar{b}$ decay mode, and members of LIP concentrate on the mixed $hh \to \tau\,\bar{\tau}\,b\,\bar{b}$ decay mode. This document describes the studies of multivariate analysis techniques aimed at selecting those signals amidst the very large backgrounds, as well as regressing measured signal parameters such that their estimate can be improved as much as possible. A different final state of interest of the AMVA4NewPhysics network, the $t\bar{t}h$ associated production process of single Higgs bosons and top quark pairs, is under study by CERN, Louvain, and Oviedo members; although it is not discussed in this document, the techniques described here will eventually be tested on that final state, too.

It should be borne in mind that the preliminary studies documented here aimed to examine what the potential is for machine learning in the abstract problem of discrimination of Higgs pair production from backgrounds and related variable regression problems, without reference to direct experimental detail, and as such our aim was to develop the most performant algorithms possible. This means that less consideration was paid to the number of input features used, and other items which in practice might propagate systematic uncertainties into the algorithms' outputs. Having established an upper limit of performance in the abstract setup, it will of course be necessary to balance the benefits of including more input features against the increased uncertainties that they might bring in a real data analysis.

2 Datasets

A range of Monte Carlo (MC) samples were produced within the AMVA4NewPhysics ITN. The studies performed here only consider standard model (SM) di-Higgs production and the dominant backgrounds for each decay channel investigated. These are fully-leptonic $t\,\bar{t}$ and $b\,\bar{b}\,b\,\bar{b}$ QCD, for $h\,h \to \tau\,\bar{\tau}\,b\,\bar{b}$ and $h\,h \to b\,\bar{b}\,b\,\bar{b}$, respectively. It should be noted that here "fully-leptonic" is defined according to the decays of the W bosons, which may decay only to leptons, $\ell \in \{e, \mu, \tau\}$. The semi-leptonic decay mode is expected to be a significant source of background for the $h\,h \to \tau\,\bar{\tau}\,b\,\bar{b}$ search, and this will be considered in further investigations.

2.1 Signal

A sample of 10^7 events was generated with Matrix-Elements (MEs) at leading order using MadGraph 5 [1], for the SM process $p\,p \to h\,h$ at $\sqrt{s} = 13\,\text{TeV}$. The four-flavour scheme was used, with incoming partons being sampled from the `nn23lo1` [2] parton density function (PDF). The parton showering, hadronisation, and decays were handled by Pythia 8 [3]. Pythia 8 is applied twice, separately, to produce $h\,h \to \tau\,\bar{\tau}\,b\,\bar{b}$ and $h\,h \to b\,\bar{b}\,b\,\bar{b}$ samples. In the $b\,\bar{b}\,b\,\bar{b}$ case the decays channels for Higgs bosons are restricted to $h \to b\,\bar{b}$. In the $\tau\,\bar{\tau}\,b\,\bar{b}$ case, the $h \to \tau\,\bar{\tau}$ channel is added, with a branching ratio forced to be equal to that of the $h \to b\,\bar{b}$. A subsequent MC-truth cut is then used to select events in which the Higgs-boson-pair decays to the $\tau\,\bar{\tau}\,b\,\bar{b}$ final state (half of the events).

The following ME requirements are modified from the MadGraph 5.2.4.2 default values (at the parton level):

- b-jets: $p_T \geq 20\,\mathrm{GeV}$
- jets and b-jets: $|\eta| \leq 3$
- distance between two jets: $\Delta R_{jj} \geq 0.1$
- distance between two b-jets: $\Delta R_{bb} \geq 0.1$

The ME production cross section reported by MadGraph 5 is $14.518 \pm 0.001\,\mathrm{fb}$. Using next-to-next-to-leading-log (NNLL) calculations matched to next-to-next-to-leading-order (NNLO) and accounting for top-quark mass effects to next-to-leading-order (NLO), the LHC Higgs Cross-Section Working Group [4] calculates that the production cross section for $g\,g \rightarrow h\,h$ at $\sqrt{s} = 13\,\mathrm{TeV}$ is $33.5^{+1.8}_{-2.3}\mathrm{fb}$, for $m_h = 125\,\mathrm{GeV}$, and that the branching ratios for $h \rightarrow b\,\bar{b}$ and $h \rightarrow \tau\,\bar{\tau}$ are $0.5824^{+0.0072}_{-0.0074}$ and $0.062\,72 \pm 0.0010$, respectively. The theoretical production cross sections are therefore $2.45^{+0.14}_{-0.17}\mathrm{fb}$ and $11.37^{+0.64}_{-0.80}\mathrm{fb}$ for the $h\,h \rightarrow \tau\,\bar{\tau}\,b\,\bar{b}$ and $h\,h \rightarrow b\,\bar{b}\,b\,\bar{b}$ samples, respectively.

2.2 QCD background

A sample of 10^7 events were generated with leading-order MEs using MadGraph 5, for the SM process $p\,p \rightarrow b\,\bar{b}\,b\,\bar{b}$. The four-flavour scheme was used, with incoming partons being sampled from the `nn23lo1` PDF. The parton showering, hadronisation, and decays were handled by Pythia 8.

The ME requirements are the same as those used for the production of the signal sample.

The production cross section reported by MadGraph 5 is $1.7471 \pm 0.0001\,\mathrm{nb}$. Applying a LO→NLO rescaling factor calculated from Ref. [1] of $1.73^{+1.13}_{-0.73}$, the theoretical cross section for $p\,p \rightarrow b\,\bar{b}\,b\,\bar{b}$ is $3.0^{+2.0}_{-1.3}\mathrm{nb}$. It should mentioned that the rescaling factor was calculated without consideration of generation phase-space, and its use here is purely to indicate the differences in scale of production cross section between signal and background; we recommend that a more accurate value be calculated for analyses sensitive to its value.

2.3 Fully-leptonic $t\,\bar{t}$ background

The production of background events follows the prescription and settings specified in Ref. [5]. A sample of 10^7 events with NLO MEs was generated in Powheg Box 2 [6, 7, 8] using the `hvq` process [9] for the SM process $p\,p \rightarrow t\,\bar{t}$ at $\sqrt{s} = 13\,\mathrm{TeV}$, in which the $t\,\bar{t}$ pair is forced to decay to a di-leptonic final state, including the τ lepton. Incoming partons are sampled from the `CT10` PDF [10], included via Lhapdf 5 [11]. Parton showering, hadronisation, and unstable decays were handled by Pythia 8.

The production cross section reported by Powheg Box is $73.04 \pm 0.06\,\mathrm{pb}$. Using NNLL matched to NNLO calculations and assuming a top-quark mass of $173.2\,\mathrm{GeV}$, the production cross section for $p\,p \rightarrow t\,\bar{t}$ at $\sqrt{s} = 13\,\mathrm{TeV}$ is $816.0^{+39.5}_{-44.7}\mathrm{pb}$ [12]. The branching ratio for $W \rightarrow \ell\nu$ is 0.3272 ± 0.0030 [12], for $\ell \in e, \mu, \tau$. The theoretical production cross section for fully-leptonic $t\,\bar{t}$ is therefore $87.4^{+4.4}_{-4.9}\mathrm{pb}$.

2.4 Detector simulation

The simulation applied to both signal and background samples using Delphes [13, 14, 15] was configured to produce a response in between the ATLAS [16] and CMS [17] detectors. This choice was dictated by the need to allow researchers that belong to the two collaborations to profit equally from those studies. A middle-ground between CMS and ATLAS also allows results to be obtained which are not too dependent on experimental detail.

Delphes uses parametrised responses to allow the quick simulation of a real detector-environment by reconstructing final-state objects with given efficiencies, applying resolution effects, and simulating pile-up contributions. Whilst it is not expected to provide a simulation as accurate as that of Geant 4 [18, 19], it is expected to be sufficiently accurate to validate the proofs-of-concepts addressed, and the comparisons made, in this document.

3 Event selection and feature definition

3.1 $\tau\,\bar{\tau}\,b\,\bar{b}$ selection

3.1.1 Selection and categorisation

The samples are analysed in Root [20]. Sequential attempts are made to accept events into exclusive final state categories in the following order: $\mu\,\tau_h\,b\,b$, $e\,\tau_h\,b\,b$, $\tau_h\,\tau_h\,b\,b$, $\mu\,\mu\,b\,b$, $e\,e\,b\,b$, and $e\,\mu\,b\,b$, where τ_h indicates a resolved τ-tagged jet, and b indicates a resolved b-tagged jet.

Tables 1, 2, and 3 detail the selection requirements for light leptons, τ jets, and b jets, respectively. These cuts aim to reflect an example of the acceptance that would be achievable at a particle detector like CMS. The values themselves were based on those used in Ref. [21], which describes a previous investigation by CMS into $h\,h \to \tau\,\bar{\tau}\,b\,\bar{b}$. Jets are reconstructed using the anti-k_t algorithm [22] with an R parameter of 0.5. In case of ambiguity in the selection, the following choices are made:

- When the selected event contains more than the required number of acceptable τ-jets, the hardest τ jets which satisfy the charge requirements are chosen.
- When the selected event contains more than two acceptable b-jets, the pair whose invariant mass is closest to 125 GeV (i.e. the Higgs-boson mass) is chosen.

Signal events passing the final selection cuts undergo an additional check using information from the MC generator (MC-truth) by setting a flag according to whether the selected final

	Primary lepton		Secondary lepton	
Lepton type	e	μ	e	μ
$p_T >$	24 GeV	19 GeV	10 GeV	10 GeV
$\|\eta\| <$	2.1	2.1	2.4	2.5
$I_{rel} <$	0.1	0.1	0.3	0.3
Lepton multiplicity				
$\ell\,\tau_h\,b\,b$	1		0	
$\tau\,\bar{\tau}\,b\,\bar{b}$	0		0	
$\ell\,\ell\,b\,b$	2 of opposite charge		0	

Table 1: Selection requirements on light leptons ($\ell \in e, \mu$). Lepton-multiplicity requirements are exclusive, e.g. in the case of the $\ell\,\tau_h\,b\,b$ final state if a secondary e or μ were present, the event would be rejected.

Final state	$\ell\,\tau_h\,b\,b$	$\tau\,\bar{\tau}\,b\,\bar{b}$	$\ell\,\ell\,b\,b$
	at least 1 OS	at least 1 OS pair	none
$p_T >$	20 GeV	45 GeV	10 GeV
$\|\eta\| <$	2.3	2.1	2.5

Table 2: Selection requirements on τ jets. Delphes *uses a parametrised tagging algorithm with a Boolean output. In the $\ell\,\tau_h\,b\,b$ category, the τ jet must have a charge opposite to that of the primary lepton, while for $\tau\,\bar{\tau}\,b\,\bar{b}$ candidate events must contain an oppositely charged pair of τ jets. In the case of $\ell\,\ell\,b\,b$ final state, the event is rejected if any τ jets with $p_T > 10\,\mathrm{GeV}$ and $|\eta| < 2.5$ are present.*

	At least 2 b-jets
$p_T >$	30 GeV
$\|\eta\| <$	2.4

Table 3: Selection requirements on b jets. Delphes *uses a parametrised tagging algorithm with a Boolean output.*

states correspond to the true decay products of both Higgs bosons. For the b-jets, this requires the generator-level b-quarks produced by the decay of a generator-level Higgs boson to be within 0.5 rad of the selected b-jets. For hadronically decaying τ leptons, the generator-level τ lepton coming from the decay of a Higgs boson must be within 0.5 rad of the selected τ-jet. For leptonically decaying τ leptons, the generator-level light-lepton, corresponding to the reconstruction-level selected light-lepton, must have been produced by the decay of a generator-level τ-lepton which was produced by the decay of a generator-level Higgs boson. If all four final states are correctly selected, the event is flagged as passing the MC-truth match.

3.1.2 Acceptance

Table 4 gives the acceptance values for signal and background events into the six final state categories. Table 5 details the acceptance of signal and background events into the six final state categories times the production cross section of the sample. “Signal MM” indicates the mismatched events, i.e. the accepted events from the signal sample which failed the MC-truth match.

	Acceptance [%]		
Sample	$\mu\,\tau_h\,b\,b$	$e\,\tau_h\,b\,b$	$\tau_h\,\tau_h\,b\,b$
Signal	1.228 ± 0.005	1.044 ± 0.005	1.150 ± 0.005
Signal MM	0.204 ± 0.002	0.174 ± 0.002	0.218 ± 0.002
Bkg.	1.037 ± 0.003	0.925 ± 0.003	0.0550 ± 0.0007
	$e\,\mu\,b\,b$	$\mu\,\mu\,b\,b$	$e\,e\,b\,b$
Signal	0.250 ± 0.002	0.131 ± 0.002	0.124 ± 0.002
Signal MM	0.0373 ± 0.0009	0.0196 ± 0.0006	0.0156 ± 0.0006
Bkg.	2.225 ± 0.005	1.169 ± 0.003	1.058 ± 0.003

Table 4: Acceptance of events (in %) for each final state category. “Signal MM” indicates events from the signal sample which failed the MC-truth match.

	Acceptance × cross-section [pb]		
Sample	$\mu\,\tau_h\,b\,b$	$e\,\tau_h\,b\,b$	$\tau_h\,\tau_h\,b\,b$
Signal	$(3.0^{+0.2}_{-0.2}) \times 10^{-5}$	$(2.6^{+0.1}_{-0.2}) \times 10^{-5}$	$(2.8^{+0.2}_{-0.2}) \times 10^{-5}$
Signal MM	$(5.0^{+0.3}_{-0.4}) \times 10^{-6}$	$(4.3^{+0.2}_{-0.3}) \times 10^{-6}$	$(5.3^{+0.3}_{-0.4}) \times 10^{-6}$
Bkg.	$0.91^{+0.05}_{-0.05}$	$0.81^{+0.04}_{-0.05}$	$0.048^{+0.002}_{-0.003}$
	$e\,\mu\,b\,b$	$\mu\,\mu\,b\,b$	$e\,e\,b\,b$
Signal	$(3.2^{+0.2}_{-0.2}) \times 10^{-6}$	$(6.1^{+0.4}_{-0.4}) \times 10^{-6}$	$(3.0^{+0.2}_{-0.2}) \times 10^{-6}$
Signal MM	$(4.8^{+0.3}_{-0.4}) \times 10^{-7}$	$(9.1^{+0.6}_{-0.7}) \times 10^{-7}$	$(3.8^{+0.3}_{-0.3}) \times 10^{-7}$
Bkg.	$1.02^{+0.05}_{-0.06}$	$1.9^{+0.1}_{-0.1}$	$0.92^{+0.05}_{-0.05}$

Table 5: Acceptance times production cross-section for each final-state category. "Signal MM" indicates events from the signal sample which failed the MC truth match.

3.1.3 Event reconstruction

Once the event is selected, the two Higgs bosons are reconstructed from the selected final states.

First, the two τ-leptons are defined: In the case of a hadronically decaying τ lepton, the 4-momentum of the τ-tagged jet is used; for leptonically decaying τ leptons, the 4-momentum of the light lepton is used. The 4-momentum of the parent Higgs-boson ($h_{\tau\,\bar{\tau}}$) is then reconstructed from the vectorial sum of the 4-momenta of the two τ leptons and the vector of missing momentum projected in the plane transverse to the beam axis. Next, the 4-momentum of the Higgs boson which decays to $b\,\bar{b}$ ($h_{b\,\bar{b}}$) is reconstructed from the vectorial sum of the 4-momenta of the two selected b-tagged jets. Finally, the combined 4-momentum of the two Higgs bosons is calculated (the di-Higgs).

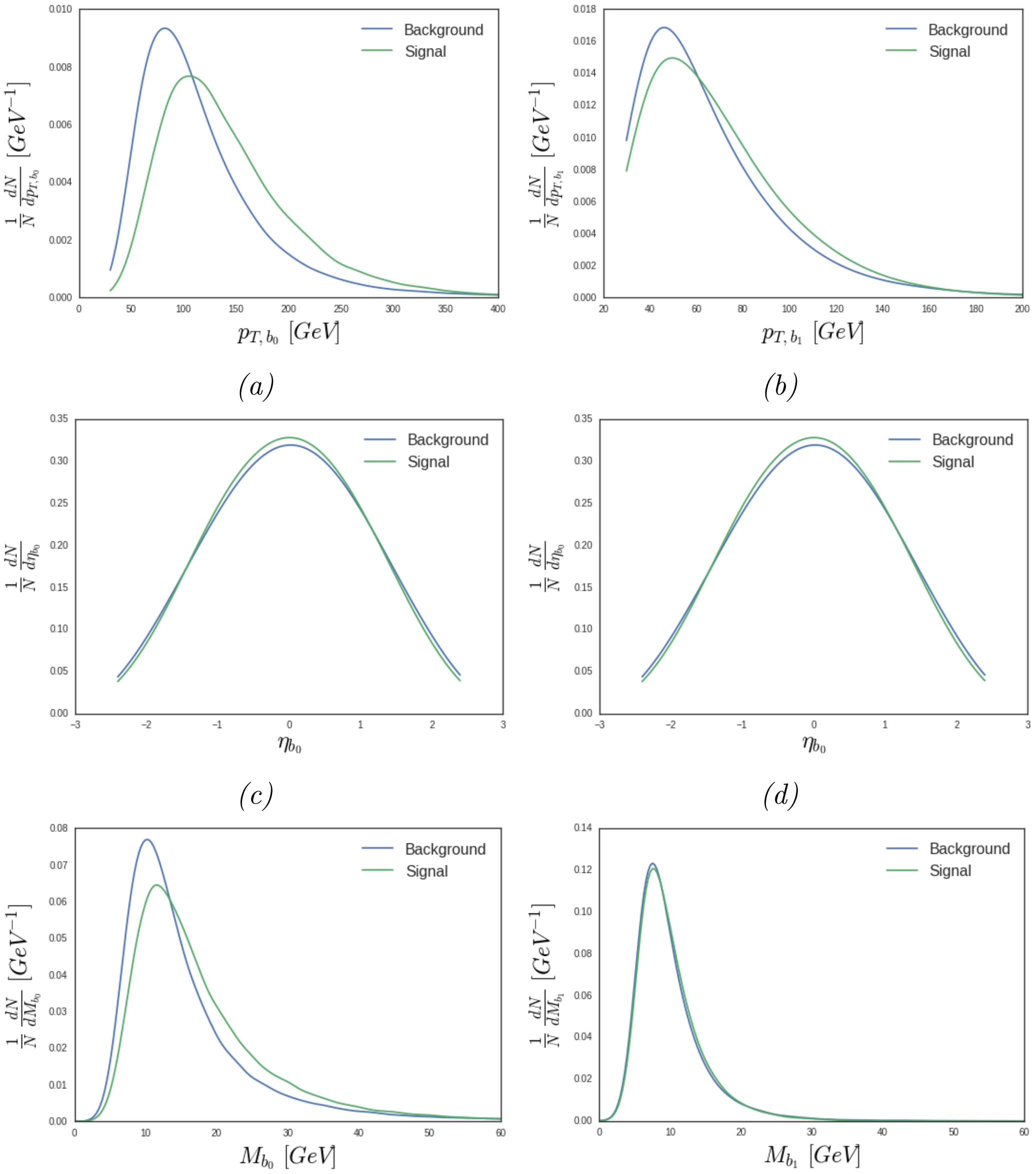
3.1.4 Final-state features

18 low-level final state features are calculated during the event selection process: p_T, η, and ϕ for each selected final state, the magnitude of missing transverse momentum ($\overrightarrow{p}_T^{\text{miss}}$), and the azimuthal angle of the vector of the missing momentum ($\phi_{\overrightarrow{p}_T^{\text{miss}}}$). We label the harder of the two b-jets as '0' and the other as '1'. The τ leptons are labelled thusly: in the $\tau_h\,\tau_h\,b\,b$ and $\ell\,\ell\,b\,b$ categories the harder of the two τ leptons is labelled '0' and the other as '1'; in the $\ell\,\tau_h\,b\,b$ categories the '0' indicates the τ-jet and '1' indicates the lepton.

Twelve reconstructed features are also calculated: p_T, η, ϕ, and the invariant masses of the three systems which correspond to: the Higgs boson which decays to $b\,\bar{b}$ ($h_{b\,\bar{b}}$); the Higgs boson which decays to $\tau\,\bar{\tau}$ ($h_{\tau\,\bar{\tau}}$); and the di-Higgs-boson system ($h\,h$). The transverse mass (m_T) is also calculated in the case of $\ell\,\tau_h\,b\,b$ final states, according to Eq. 1:

$$m_T = \sqrt{2p_{T,\ell} \times \overrightarrow{p}_T^{\text{miss}} \times \left(1 - \cos\Delta\phi_{\ell,\overrightarrow{p}_T^{\text{miss}}}\right)}. \tag{1}$$

Example distributions of these features are shown in Figs. 1 to 5 in the form of kernel density-estimations [23, 24] (KDE). Similar to histograms, these are an estimation of the probability-density function for a random variable, however they demonstrate quicker convergence to the true density [25]. They also have the advantage of providing a smooth estimation of densities. Rather than binning data and letting each data point contribute a given area to its bin, the KDE method centres a kernel function (in our case a Gaussian function) at each data point, sums their contributions, and then normalises the area to one. The variance of the Gaussian kernel is a free parameter which can either be set by hand, or estimated by methods such as *Silverman's rule of thumb* [26].

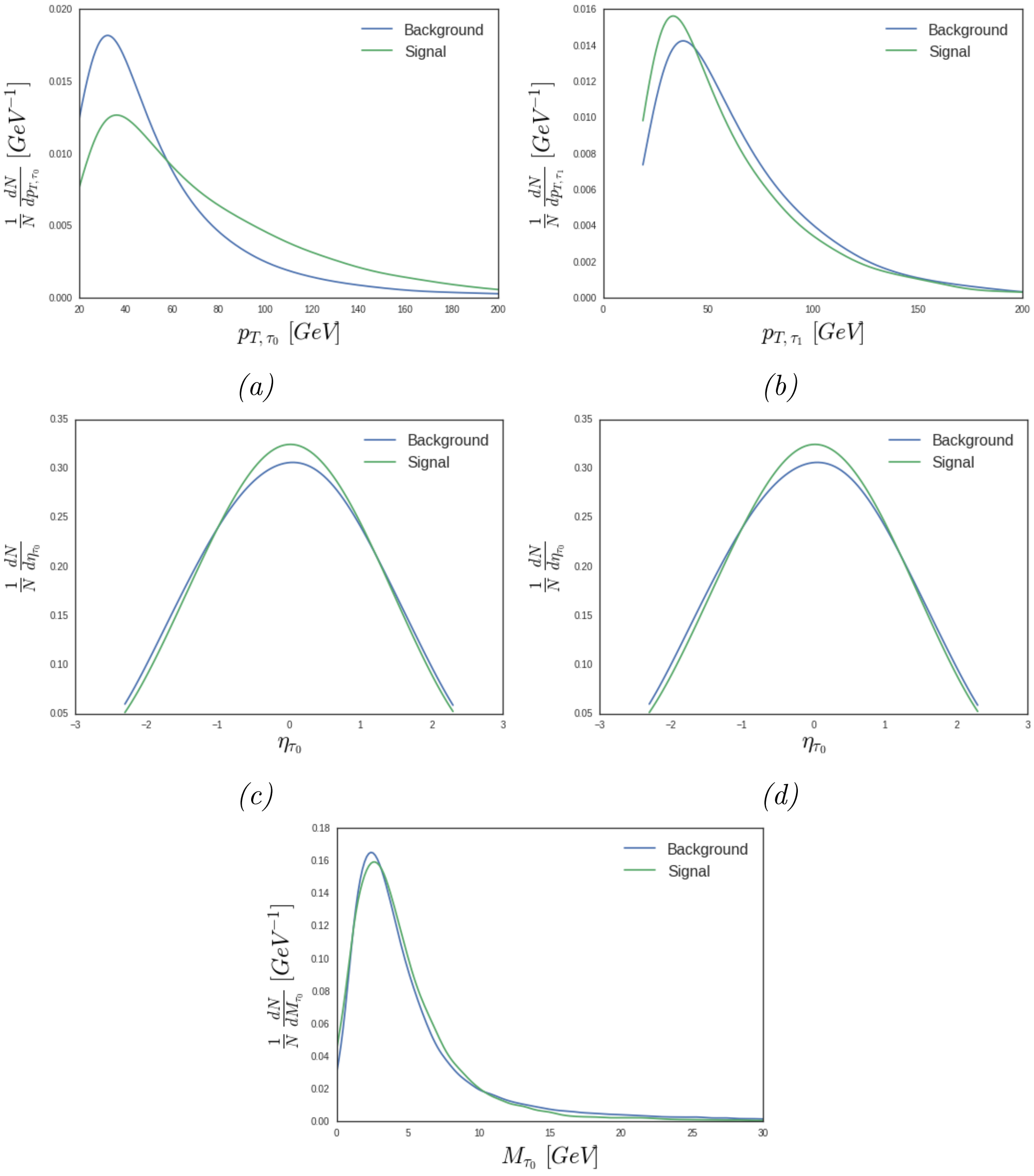


(a) *(b)*

(c) *(d)*

(e) *(f)*

Figure 1: Feature distributions for selected b-jets for signal and background events in the $\mu\,\tau_h\,b\,b$ *category.* b_0 *is the harder of the two jets. Both samples are normalised to one*

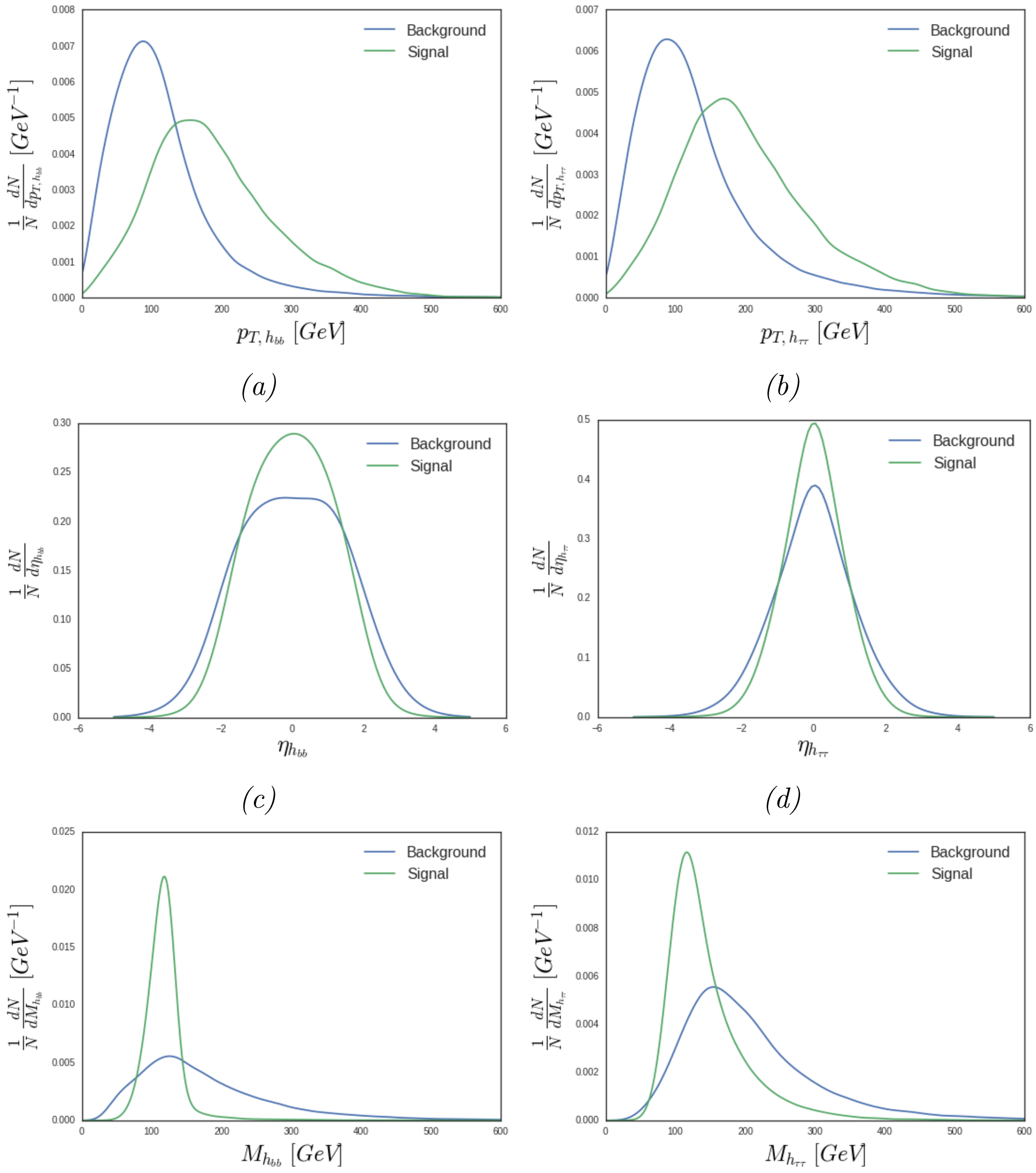


(a) *(b)*

(c) *(d)*

(e)

Figure 2: Feature distributions for selected τ-leptons for signal and background events in the $\mu\,\tau_h\,b\,b$ category. τ_0 corresponds to the τ_h and τ_1 corresponds to the muon. Both samples are normalised to one

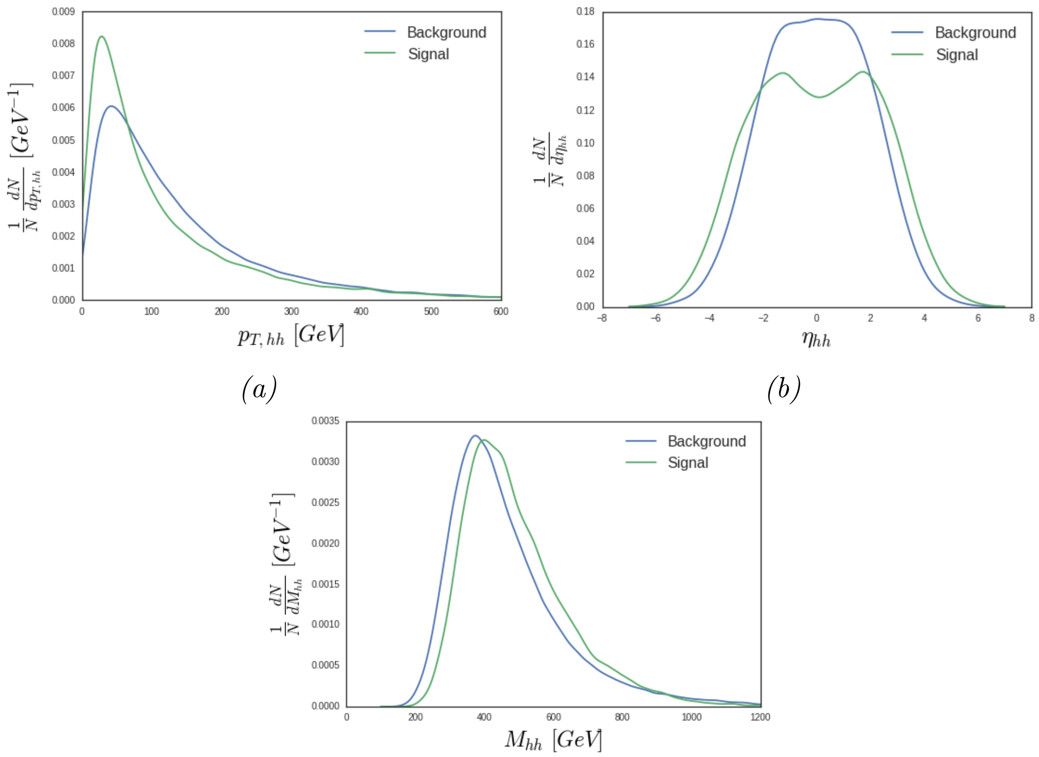


(a) *(b)*

(c) *(d)*

(e) *(f)*

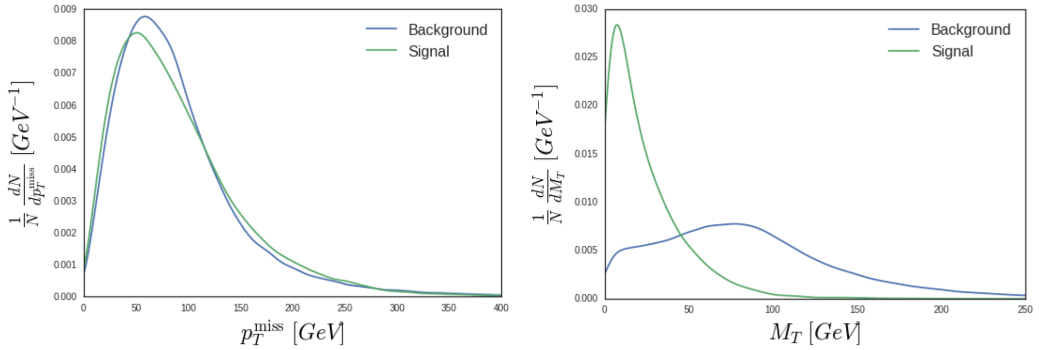
Figure 3: Feature distributions for $h_{b\bar{b}}$ and $h_{\tau\bar{\tau}}$ systems for signal and background events in the $\mu\,\tau_h\,b\,b$ category. Both samples are normalised to one



(a) *(b)*

(c)

Figure 4: Feature distributions for di-Higgs system for signal and background events in the $\mu\,\tau_h\,b\,b$ category. Both samples are normalised to one



(a) *(b)*

Figure 5: Feature distributions for the magnitude of missing transverse momentum and M_T for signal and background events in the $\mu\,\tau_h\,b\,b$ category. Both samples are normalised to one

3.2 $b\bar{b}b\bar{b}$ selection

3.2.1 Selection and reconstruction

The data sample is analysed in ROOT [20]. The selection begins with the requirement that the events contain at least 4 b-tagged jets with $p_T > 30$ GeV, located in the central rapidity region with $|\eta| < 2.5$. The b jets are tagged with the anti-k_T reconstruction algorithm [27] and with a jet size of R = 0.5. As mentioned in Sec. 2, at the matrix element level all the final state jets must also be separated by $\Delta R > 0.1$. The loose selection criteria applied to the $b\bar{b}b\bar{b}$ final state events are summarized in Tab. 6.

Table 6: Selection requirements on $b\bar{b}b\bar{b}$ final state.

	At least 4 b-jets
p_T	> 30 GeV
$\lvert\eta\rvert$	< 2.5
ΔR	> 0.1

Once the cut based selection has been applied, the four jets with the highest p_T are paired to construct two Higgs boson candidates. All possible jet pair combinations are considered and then the configuration that minimises the relative difference of di-jet masses is chosen. This combination represents the choice most consistent with the decays of two particles of equal mass. The *leading* Higgs boson candidate is defined to be the candidate with the highest p_T. Consequently, the remaining Higgs candidate is called *sub-leading*.

Once the Higgs boson candidates have been identified, their invariant mass is required to be close to the nominal Higgs boson mass of $m_h = 125$ GeV. In particular we require the following condition:

$$|m_h - 125\,\text{GeV}| < 40\,\text{GeV}. \tag{2}$$

Here, m_h is the invariant mass of each of the two Higgs candidates. For the signal process we expect a clear enhancement in the mass distribution of the reconstructed Higgs boson around the nominal value of the mass (125 GeV), while we do not expect any particular structure for the background. Indeed, with this cut we retain the 60% of the signal candidates which have passed the previous loose cuts, while we keep just the 26% of the main background considered. The previous consideration can be inferred from Tab. 7.

3.2.2 Acceptance

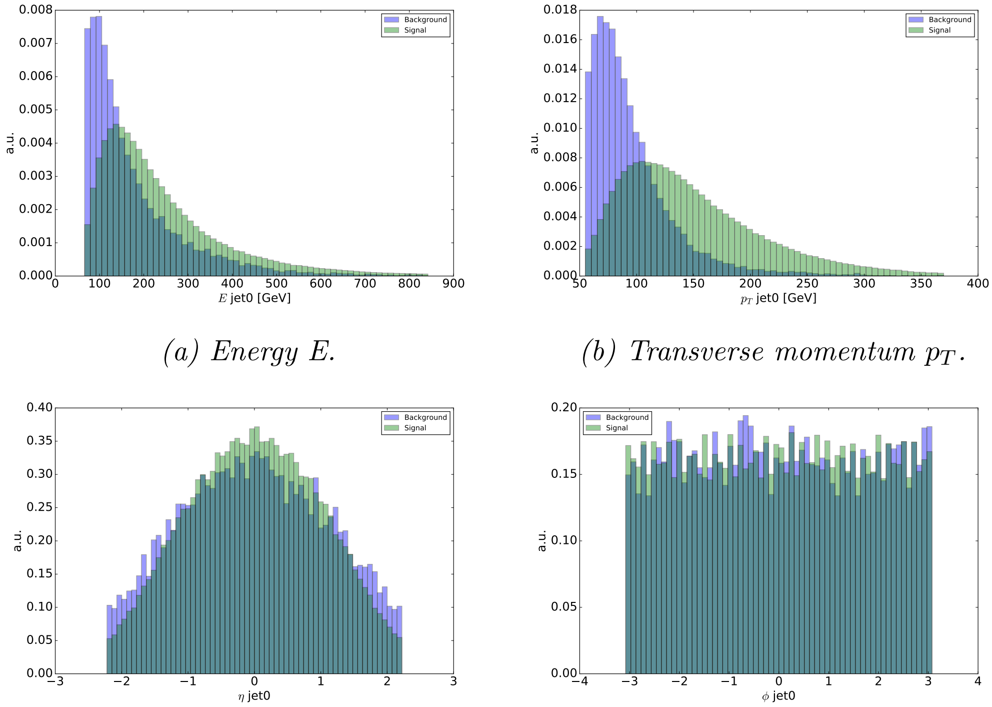
In Tab. 7 the acceptance values for signal and background events before and after applying the cut defined in Eq. 2 are reported. Note that the acceptance values are given here as a percentage of the total number of generated events (10^7 for each sample as described in Sec. 2). The values obtained for the acceptance of the background samples justify the choice to keep only the most prominent background $pp \to b\bar{b}b\bar{b}$ in the following studies of classification and regression.

3.2.3 Final-state features

Table 7: Acceptance in percentage for each signal and background process before (second column) and after (third column) the requirement in Eq. 2.

Process	Acceptance before m_h cut [%]	Acceptance after m_h cut [%]
$hh \to b\bar{b}b\bar{b}$	13.142 ± 0.012	7.909 ± 0.009
$pp \to b\bar{b}b\bar{b}$	0.762 ± 0.003	0.203 ± 0.001
$pp \to jjjj$	0.0008 ± 0.0001	0.0002 ± 0.0001
$pp \to bbjj$	0.0278 ± 0.0005	0.0059 ± 0.0002
$pp \to t\bar{t} \to b\bar{b}jjjj$	0.519 ± 0.002	0.204 ± 0.001

The event selection returns 16 kinematic variables describing the final state event. For each of the four selected b-jets, the components of the four-vector are given: the transverse momentum p_T, the energy E and the angular variables η and ϕ. These variables represent the low level features used as input to the classification of signal versus background process. The low level features of the b-jet with highest momentum in the leading Higgs candidate are shown in Fig. 6.



(a) Energy E.

(b) Transverse momentum p_T.

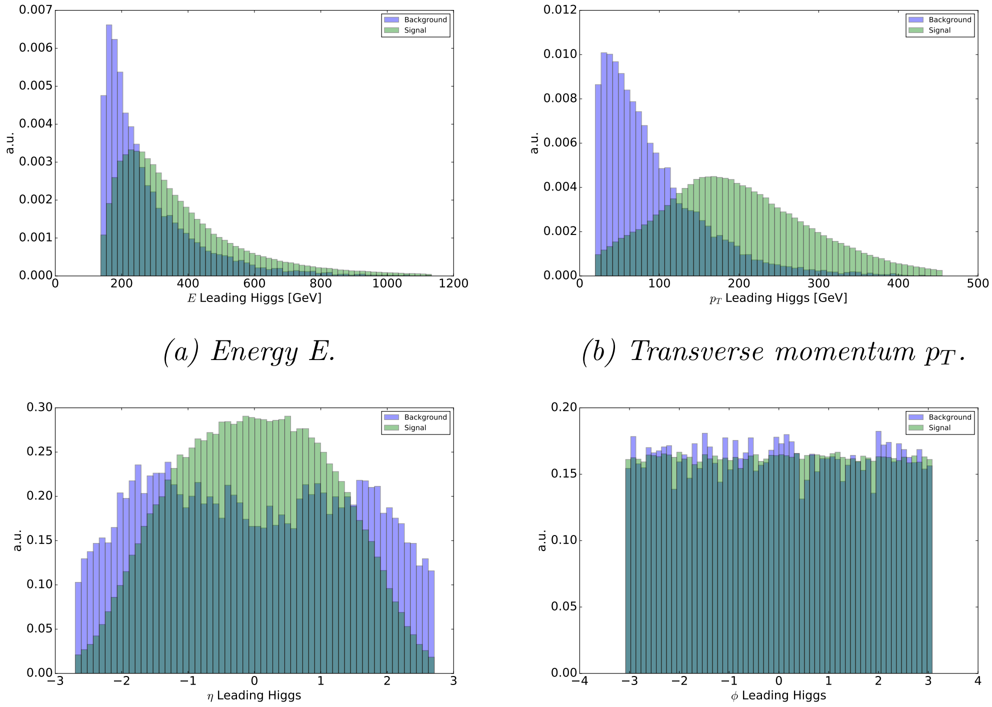
(c) Pseudorapidity η.

(d) Angular variable ϕ.

Figure 6: Low level features associated to the b-jet with the highest momentum, coming from the leading Higgs.

We observe that the distributions of the jet p_T and E fall off more rapidly for background events than the di-Higgs signal. The pseudorapidity distribution shown in Fig. 6c is slightly different in the two cases, suggesting that the production is more central in the case of signal events. As expected, background and signal events display a uniform distribution for the jet ϕ.

In order to improve the classification performance, we calculate the following high level features: p_T, E, η and ϕ for each of the two Higgs boson candidates. The relative distributions for the leading Higgs candidate are illustrated in Fig. 7. We observe a trend similar to the one noticed in Fig. 6, but for the signal of the reconstructed Higgs boson we obtain, as expected, a harder p_T spectrum and a stronger difference in the η distribution with respect to the background.



(a) Energy E.

(b) Transverse momentum p_T.

(c) Pseudorapidity η.

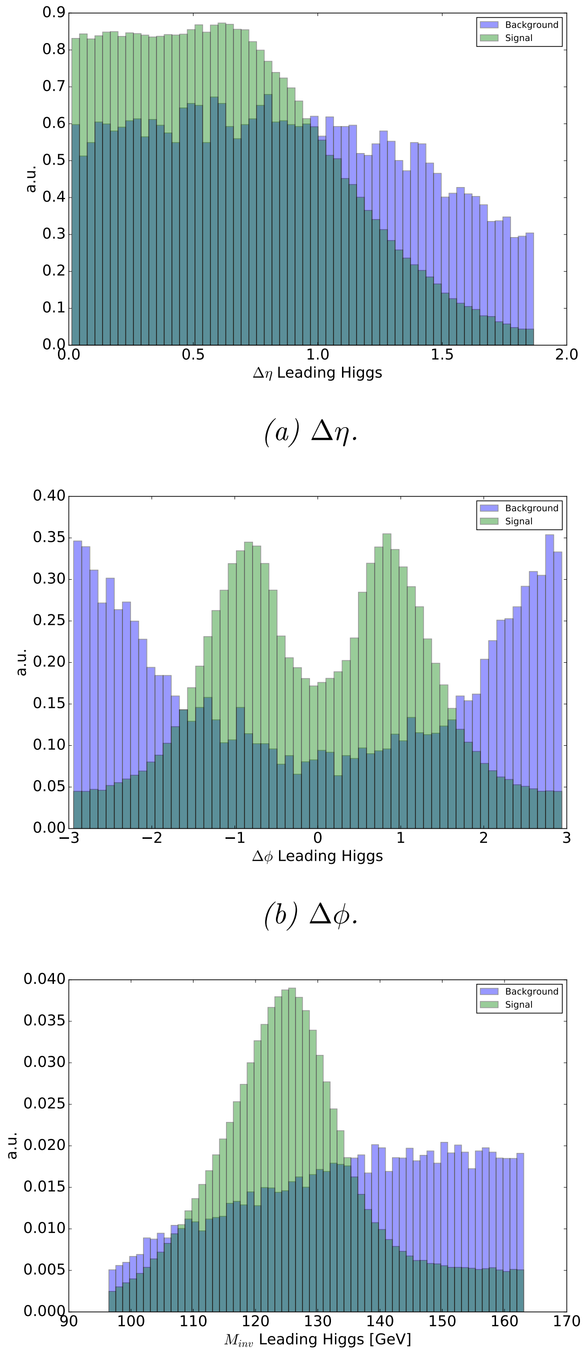
(d) Angular variable ϕ.

Figure 7: Low level features associated to the Higgs boson candidate with the highest momentum.

In addition, we have studied the variables $\Delta\eta$ and $\Delta\phi$, *i.e.* the angular distance between the two jets coming from the same Higgs candidate, and the invariant mass of the two Higgs candidates, defined as

$$M_{inv} = \sqrt{E^2 - p_T^2(1 + \sinh\eta^2)}. \tag{3}$$

In summary we have studied 30 features that can be used to classify signal and background events. The high level features $\Delta\eta$, $\Delta\phi$ and M_{inv} of the leading Higgs candidate are sketched in Fig. 8. We notice that these distributions are the most discriminating between signal and background. In particular, while the invariant mass presents the expected enhancement at the nominal Higgs mass (125 GeV), the background distribution does not present a specific structure.



(a) $\Delta\eta$.

(b) $\Delta\phi$.

(c) Invariant mass M_{inv}.

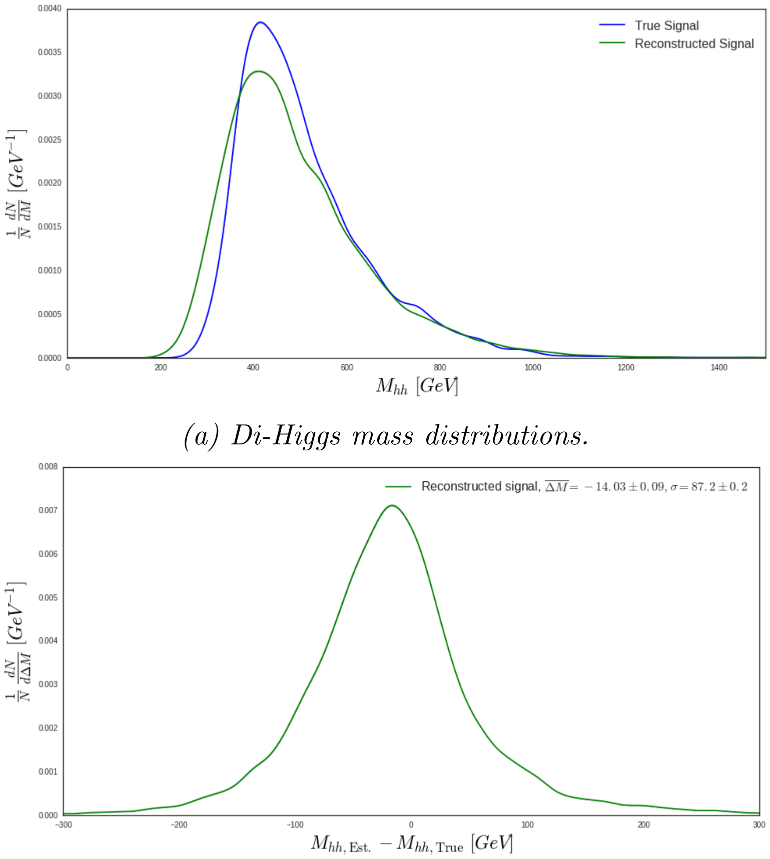
Figure 8: High level features associated to the leading Higgs boson candidate.

4 Feature regression in $\tau\,\bar{\tau}\,b\,\bar{b}$

4.1 Initial state

Figure 9 shows a comparison between the true distribution of the di-Higgs mass and that which is obtained from the reconstructed final states in the $\mu\,\tau_h\,b\,b$ category.

From these results, we can see that the reconstructed masses overestimate the low mass region, resulting in large, negative means in the delta distribution. Additionally, the delta distribution is very wide, indicating that there is not just a systematic offset in the estimation,



(a) Di-Higgs mass distributions.

(b) Delta distribution of di-Higgs mass.

Figure 9: Comparison between distributions of "true" and "reconstructed" di-Higgs mass for the $\mu\,\tau_h\,b\,b$ category.

but that the estimation process itself carries a lot of imprecision. Most likely, the inaccuracy and imprecision are due to the energies carried by the neutrinos in the τ-lepton decays not being correctly accounted for.

By using some regression tool, the accuracy and resolution of the di-Higgs mass estimation might be improved.

4.2 The regressor

Multi-layer feed-forward artificial neural-networks (NNs) are used for the regression, with the same basic layout being adopted for all applications.

Keras [28] is used to implement a NN which consists of three hidden layers, containing 150, 120, and 100 nodes, respectively. All layers are initialised according to He-normal initialisation [29] to help propagate gradients. All nodes (except output nodes) use parametrised, rectified linear-unit activation functions [29], which prevent gradients becoming saturated in the network. Output nodes use linear activation, since the regressor target-features are in the region $[-\infty, \infty]$. All layers (except the output) are batch-normalised [30] to account for differences in scale between input features, and to speed up training.

During training, drop-out is applied to nodes (except output nodes) with a probability of 0.1 [31]. In some architectures, Gaussian noise is applied throughout the network (except in output layer) in order to corrupt signals and reduce the possibility of over-fitting by forcing the network to generalise to the training data.

Each regressor is trained in batches of 32 events of signal data, for a maximum limit of 10 000 epochs. The Nesterov-Adam optimiser [32] is used to minimise the mean squared-error (MSE) of predictions (the loss function). Training can finish early if the MSE does not decrease for ten epochs and, invariably, the upper limit of 10 000 is never reached. The mean square-error is calculated according to Eq. 4:

$$\mathrm{MSE} = \frac{1}{N}\sum_{n=1}^{N}\left(\hat{y}_n - y_n\right)^2 , \tag{4}$$

where N is the number of predictions, $\hat{y}_n$ is a vector of predictions, and y_n is a vector of true values. The MSE quantises the difference between the regressor's predictions and the true momenta, i.e. lower values indicate better performance.

For development, five-fold cross-validation is used and architectures are compared by their mean final MSE values, i.e. the arithmetic mean of the MSE values of each of the five regressors is calculated. The method of k-fold cross-validation involves splitting the data into five equal-sized portions. The regressor is then trained and tested five times from scratch, each time using a different portion for testing and the remaining four for training. This method shows the general response of the regressor, rather than how it happened to respond to a particular set of data.

For training for application, data are split into 'training' and 'validation' samples, with 80 % of the data being used for training and the remaining 20 % being reserved for validation.

Regressors are then trained and tested ten times on the entirety of the training data, and the most performant NNs are selected. For regressors aiming to output the di-Higgs mass, these are:

1. The one with the lowest absolute mean pull on the di-Higgs mass;
2. the one with the lowest standard deviation for the pull on the di-Higgs mass;
3. and the one with the lowest MSE for the di-Higgs mass.

The selected NNs are combined into an ensemble by equally-weighting their outputs to produce a mean response, which is then used to calculate the regressed feature(s). The response of the regressor ensemble is then confirmed on the validation data.

Note that the validation sample is not used to make any selection or comparison of architectures or training cycles; it is a pure hold-out sample. The same split into training and validation samples is made for each regressor. Whilst this may lead to regressors acting on regressed inputs showing greater response on the training data than they show on the validation data, it is assumed that the response on the validation data is still greater than it would be if the data were instead split more times such that no data were reused, due to the larger data samples.

Also note that hyperparameter optimisation is not the focus of this preliminary study, so only two models will be considered:

- Model 0: The basic layout described above with no Gaussian noise;
- Model 1: The basic layout described above with Gaussian noise at $\sigma = 0.5$ for the first hidden layer, and $\sigma = 0.2$ for the other two hidden layers.

Model 1 aims to test whether corrupting network signals forces the network to better generalise to the data. Model 0 will show what the 'natural' response of the network is. If Model 1 demonstrates the best performance, then it is possible Model 0 over fits to the data. If Model 0 demonstrates the best performance, then the Gaussian noise was not necessary and was impeding the function of the network.

Since this is a preliminary investigation aiming to validate the use of neural-networks for regression tasks in particle physics, only the $\mu\,\tau_h\,b\,b$ final-state category is investigated. This was chosen because it contains both hadronic and leptonic decays for the τ leptons, and has a higher acceptance than the $e\,\tau_h\,b\,b$ category.

4.3 Single-stage regression

Initially, a single-stage regression approach is employed. Here, input features are used to regress directly to the di-Higgs mass.

4.3.1 Feature sets

The following input features are used: Reconstructed 3-momenta and mass of both b-quarks, both τ leptons, both Higgs bosons and their vectorial sum (di-Higgs vector); p_T and ϕ of the missing momentum vector; and m_T. The total number of features included is thus 31.

4.3.2 Development results

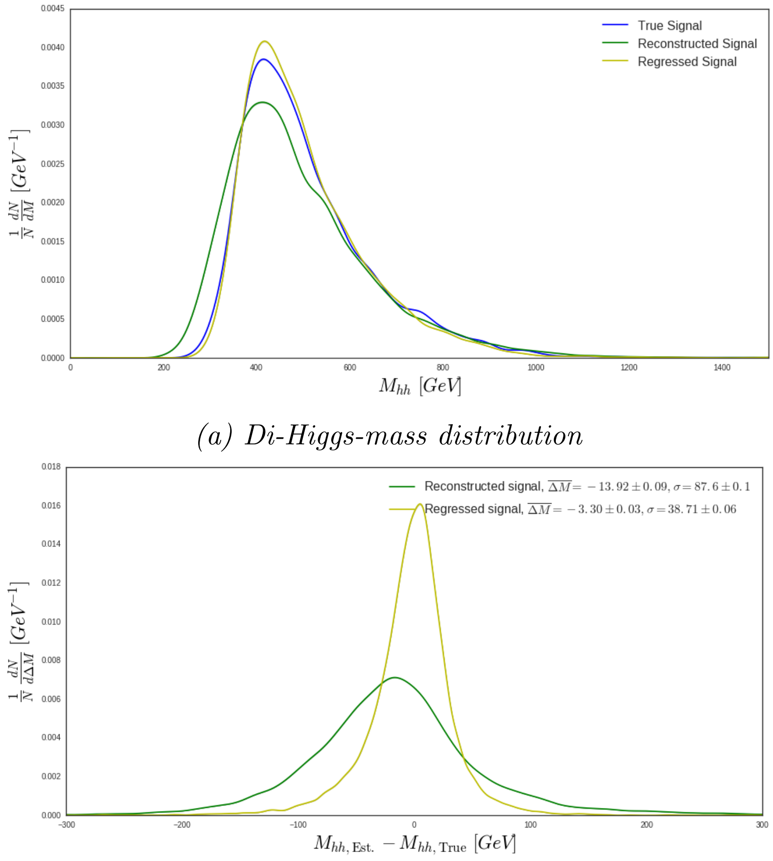
The results of training the two models on the development dataset are shown in Tab. 8. We can see that Model 0 demonstrates the best performance, by having the lowest mean squared-error, and so will be trained for application. The fact that Model 0 demonstrates the best performance indicates that the Gaussian noise in Mode 1 is not required.

Architecture	Mean MSE [GeV^2]	N_{Features}
Model 0	1870 ± 70	31
Model 1	2000 ± 200	31

Table 8: Summary of the mean MSE values for single-stage regression to the di-Higgs mass, for various architectures during development. Lower values are better.

4.3.3 Application of single-stage di-Higgs regression

Model 0 is trained for application according to the methodology in Sec. 4.2. Fig. 10 illustrates the response of the single-stage regressor on the di-Higgs mass. Fig. 10a shows that the regressor removes the low-mass overestimation of the reconstructed mass in favour of a closer match to the true (generator-level) mass-distribution. Fig. 10b re-illustrates this and quantifies the improvement: a 76 % reduction in the mean of the delta distribution, and a 56 % reduction in its width.



(a) Di-Higgs-mass distribution

(b) Delta distribution of di-Higgs mass.

Figure 10: Comparison between distributions of "true", "reconstructed", and "single-stage" regression di-Higgs mass for the $\mu\,\tau_h\,b\,b$ final state.

4.4 b-quark regression

A possible way of improving regression performance would be to incorporate a greater amount of generator-level information into input features. This can be achieved by first regressing to the 4-momenta of the decay products of the two Higgs-bosons ($h_{b\,\bar{b}}$ and $h_{\tau\,\bar{\tau}}$), and then using the regressed features as inputs to the di-Higgs-mass regressor. Due to the missing energy in τ lepton decays, it is likely that the reconstructed 4-momenta of the b-jets is closer to the true 4-momenta of the b-quarks than the reconstructed 4-momenta of the τ lepton decay products is to the 4-momenta of the τ leptons, therefore it makes sense to perform regression on the b-jets first in order to aid the τ regressor as much as possible.

In order to regress the b-jets, the generator-level 3-momenta of the b-quarks resulting from the $h_{b\,\bar{b}}$ decays are used as target features.

4.4.1 Feature sets

It was quickly found that regressing to the azimuthal angle is non-trivial, due to its closed boundaries ($-\pi = +\pi$). Several attempts were made to map the feature into a space in which the regressor could interpret it correctly, however it proved easier to move instead the features into a Cartesian coordinate system (i.e. p_x, p_y, p_z).

Four sets of features are considered:

- Set 0: Cartesian 3-momenta and mass of both b-jets and both τ-leptons, plus p_x and p_y of missing momentum vector; 18 features in total.

- Set 1: Set 0 plus magnitude of 3-momenta and energy of both b-jets and both τ-leptons; 26 features in total.
- Set 2: Set 0 plus Cartesian 3-momenta and mass of both reconstructed Higgs bosons and their vectorial sum (di-Higgs vector), plus m_T in the case of $\ell\,\tau_h\,b\,b$ final states; 31 features in total.
- Set 3: Set 0 plus Set 1 plus Set 2; 45 features in total.

4.4.2 Development results

The development results for the various combinations of models and feature sets are recorded in Tab. 9. We see that the architecture consisting of Set 3 and Model 0 demonstrates the best performance (has the lowest MSE).

Architecture	Mean MSE [GeV^2]	N_{Features}
Set 0 Model 0	750 ± 60	18
Set 0 Model 1	670 ± 20	18
Set 1 Model 0	650 ± 20	26
Set 1 Model 1	660 ± 10	26
Set 2 Model 0	700 ± 30	31
Set 2 Model 1	620 ± 10	31
Set 3 Model 0	580 ± 20	45
Set 3 Model 1	590 ± 20	45

Table 9: Summary of the mean MSE values for b-regression response, for various architectures during development. Lower values are better.

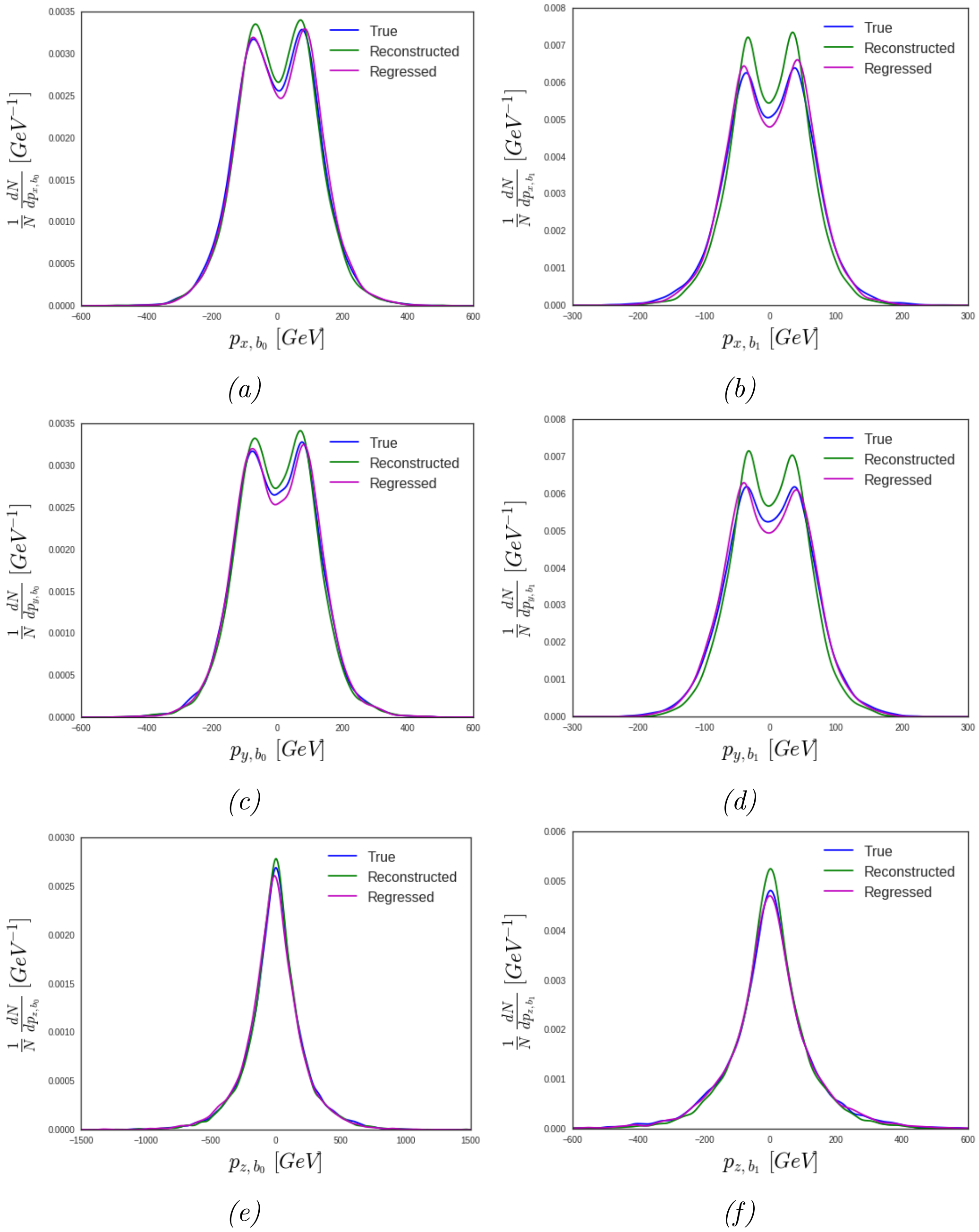
4.4.3 Application of the b-quark regression

The architecture consisting of Set 3 and Model 0 is trained for application, and Fig. 11, Fig. 12 and Fig. 13 illustrate the distributions of true and regressed b-quark momenta, the delta distributions of the b regressor on the b-quark momenta, and the reconstructed and b-regressed Higgs-mass distributions, respectively, for the $\mu\,\tau_h\,b\,b$ final state.

We can see from Fig. 11 that the reconstructed distributions tend to overestimate the regions of low absolute-momenta. Applying the b regressor moves the distributions to more closely match the true distributions in the regions of higher absolute-momenta, though at the expense of now slightly underestimating the regions of low absolute-momenta. From the delta distributions in Fig. 12 we observe that applying the b regressor increases the absolute value of the mean, indicating a decrease in estimation accuracy, but it also serves to reduce the width of the distribution, meaning that the precision of the estimation is increased.

Since the plots in Fig. 12 focus on the low to medium delta values, it might be difficult to see how the regressor increases the precision of the momenta estimates, when the plots show that the reconstructed distributions all peak at higher values than the regressed ones. Checking the extreme values of the delta distributions we find that the reconstructed estimates occasionally produce very high values, which the regressor helps to improve, at the cost of reduced precision in the low-delta regions. Effectively: the reconstruction method either functions very well (narrow peak in low delta-regions) or very poorly (delta distributions extend out to high values); the regressor aims for balanced precision by focussing on improving the poor estimates and concentrating less on the good estimates.

Indeed, by checking the invariant-mass distributions for the b-jet pairs in signal (Fig. 13), we see that the b regressor causes a decrease in the low-mass region, and a centring and symmetrising of the distribution about a mass close to that of the Higgs boson. For signal, this distribution should be a Delta function at the Higgs mass, so the decrease in the width of the distribution is encouraging. This also means that regardless of the reduced precision in the low-delta regions of Fig. 12, applying the regressor considerably improves the mass estimation for $h_{b\bar{b}}$.



(a)

(b)

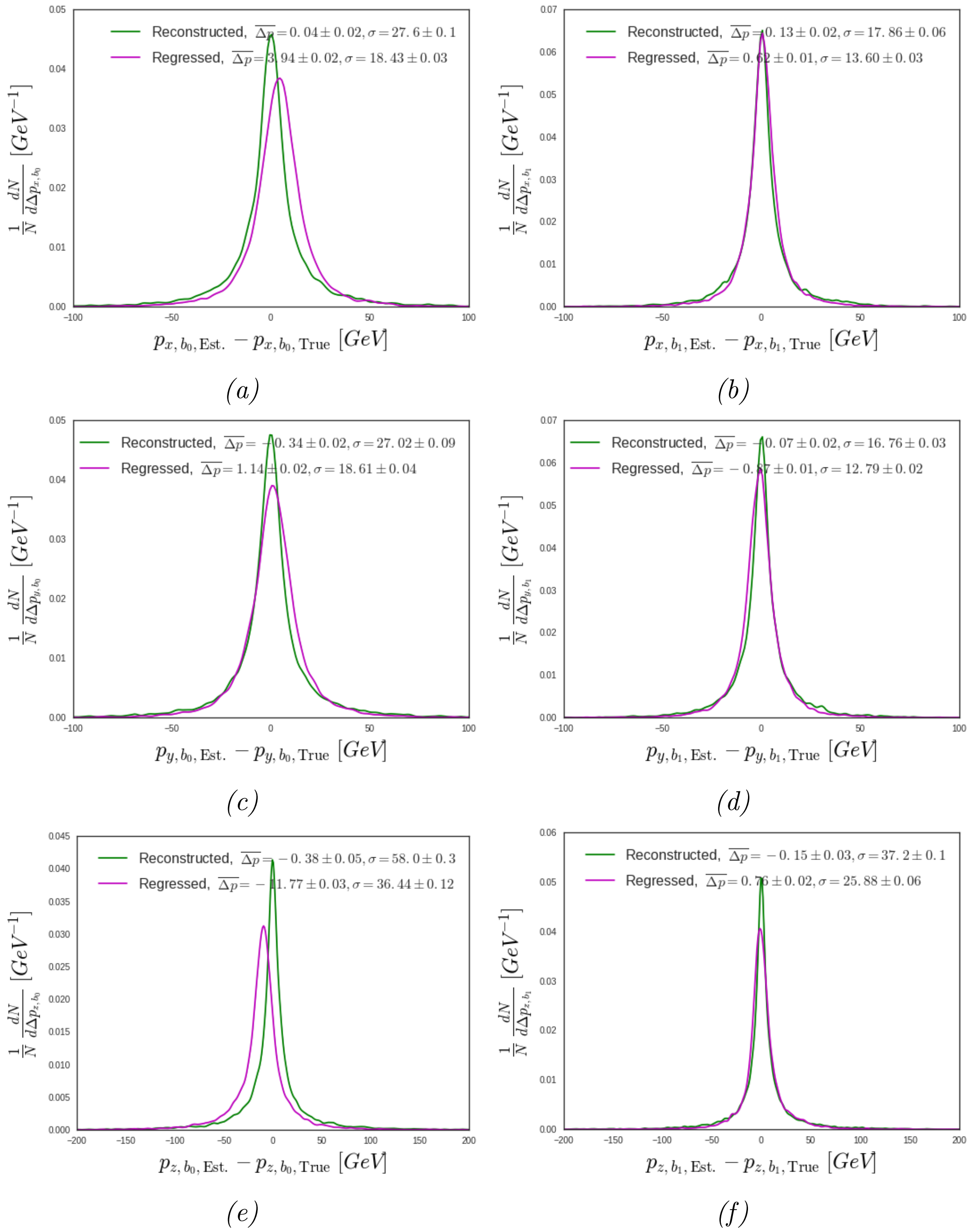
(c)

(d)

(e)

(f)

Figure 11: True and regressed b-quark momenta distributions for signal events in the $\mu\,\tau_h\,b\,b$ category.



(a)

(b)

(c)

(d)

(e)

(f)

Figure 12: Delta distributions for b-regressor response on b-quark momenta for signal events in the $\mu\,\tau_h\,b\,b$ category.

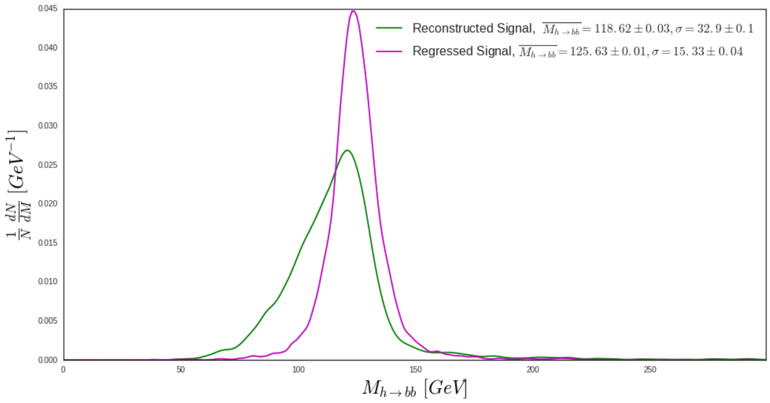


Figure 13: Reconstructed and regressed b-jet pair invariant-mass distributions for signal events in the $\mu\,\tau_h\,b\,b$ final state.

4.5 τ-lepton regression

Next, we apply regression to the τ-lepton momenta.

4.5.1 Feature sets

Eight sets of features are considered:

- Set 0: Reconstructed Cartesian 3-momenta and mass of both b-jets and both τ-leptons, plus p_x and p_y of the missing momentum vector; 18 features in total.
- Set 1: Set 0 plus magnitude of reconstructed 3-momenta and energy of both b-jets and both τ-leptons, using 26 features;
- Set 2: Set 0 plus reconstructed Cartesian 3-momenta and mass of both reconstructed Higgs bosons and their vectorial sum (di-Higgs vector), plus m_T in the case of $\ell\,\tau_h\,b\,b$ final states; 31 features in total;
- Set 3: Set 0 plus Set 1 plus Set 2; 45 features in total.
- Set 4: Set 0, but using b-regressed momenta for both b-jets and using Pythia 8's b-quark mass instead of the reconstructed b-jet masses; 18 features in total.
- Set 5: Set 1, but using b-regressed momenta and energy for both b-jets and using Pythia 8's b-quark mass instead of the reconstructed b-jet masses; 26 features in total;
- Set 6: Set 2, but using b-regressed momenta for both b-jets, using Pythia 8's b-quark mass instead of the reconstructed b-jet masses, and b-regressed momenta and massed for $h_{b\,\bar{b}}$ and di-Higgs; 31 features in total.
- Set 7: Set 4 plus Set 5 plus Set 6, using 45 features.

4.5.2 Development results

Development results for the eight feature-sets are shown in Tab. 10. We see that the architecture of Set 7 and Model 1 demonstrates the lowest MSE. Here we see that including Gaussian noise allows the regressor to better generalise to the data.

Architecture	Mean MSE [GeV^2]	N_{Features}
Set 0 Model 0	860 ± 20	18
Set 0 Model 1	980 ± 30	18
Set 1 Model 0	980 ± 20	26
Set 1 Model 1	910 ± 30	26
Set 2 Model 0	870 ± 50	31
Set 2 Model 1	860 ± 20	31
Set 3 Model 0	900 ± 70	45
Set 3 Model 1	860 ± 10	45
Set 4 Model 0	1000 ± 30	18
Set 4 Model 1	950 ± 40	18
Set 5 Model 0	960 ± 20	26
Set 5 Model 1	930 ± 20	26
Set 6 Model 0	910 ± 60	31
Set 6 Model 1	880 ± 20	31
Set 7 Model 0	860 ± 20	45
Set 7 Model 1	840 ± 30	45

Table 10: Summary of the mean MSE values for τ-regression response for various architectures during development. Lower values are better.

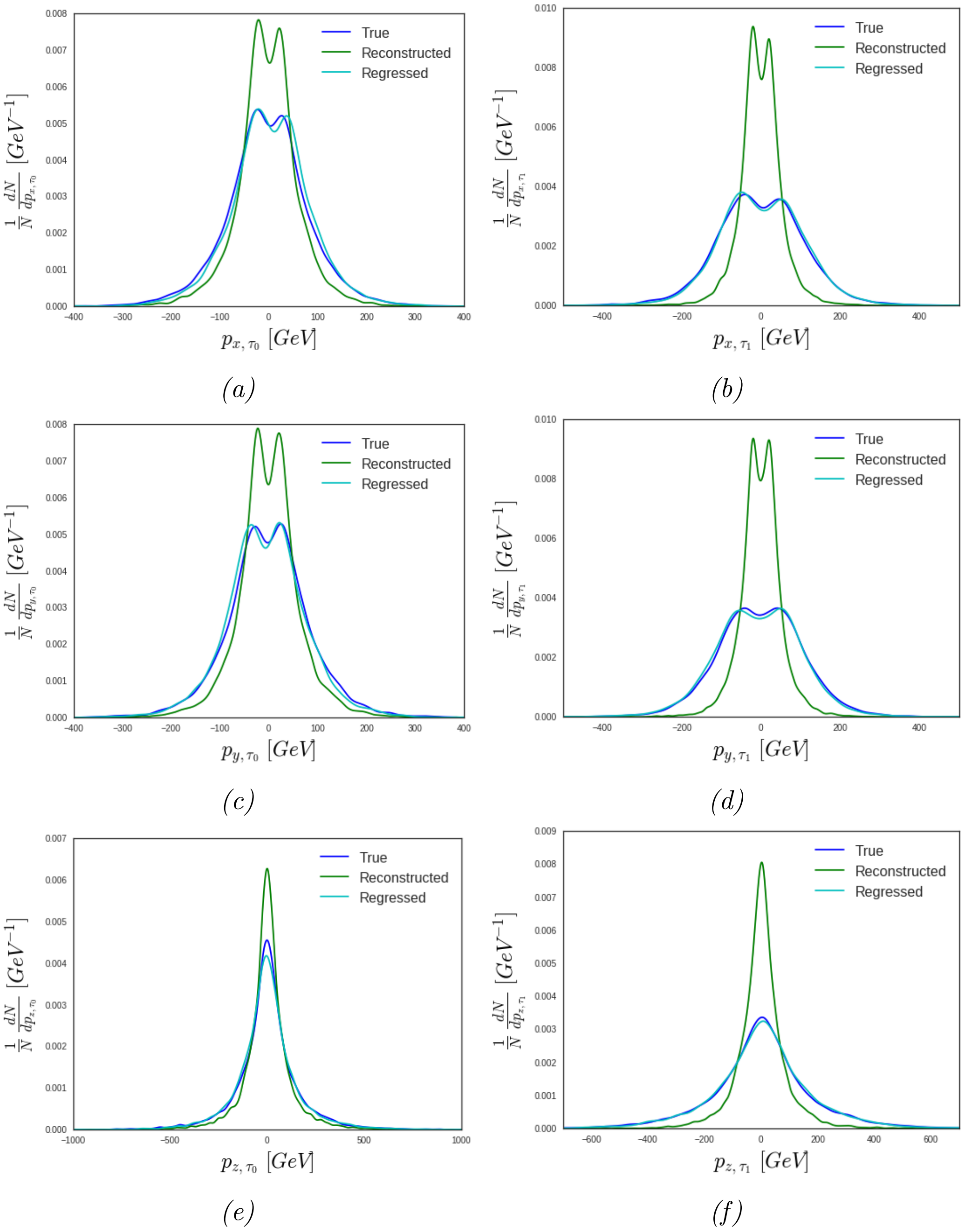
4.5.3 Application of τ-lepton regression

The architecture consisting of Set 7 and Model 1 is trained for application, and Fig. 14, Fig. 16, and Fig. 15 illustrate the distributions of true and regressed τ-lepton momenta, the delta distributions of the τ regressor on the τ-lepton momenta, and the reconstructed and τ-regressed Higgs-mass distributions, respectively, for the $\mu\,\tau_h\,b\,b$ final state.

The results mirror those seen for the b regression in Sec. 4.4, except that in Fig. 14 we see an even more extreme overestimation of the low absolute-momenta in the reconstructed distributions. This can be explained by the fact that the neutrinos resulting from the decays of that τ leptons are not accounted for, so it is to be expected that the reconstructed momenta are lower than the true values; indeed τ_1 is actually just a muon in this final-state category (see Sec. 3.1.4). As before, however, we see that applying the regressor moves the distributions to more closely match the true ones.

In Fig. 15 we see that, as before, the regressor causes increases in the absolute values of the means of the delta distributions, but large decreases in their widths.

From the estimations of the di-τ-lepton invariant-masses we see that the regressed distribution is centred close to the Higgs mass and has a smaller width than the reconstructed distribution. It should be remembered from Sec. 3.1.3 that the reconstructed distribution is the invariant mass of the τ jet, the muon, and the missing transverse-momentum. We can see from the overestimation of the high-mass region that including all of the missing energy in the τ leptons is too crude an approximation, and some of this should have been included in the b-jet pair, which exhibited an underestimation of the Higgs mass. The regressors, however, provide an increase in both the accuracy and the precision of the Higgs-mass estimations without requiring us to correctly distribute the missing energy amongst the final-states.



(a)

(b)

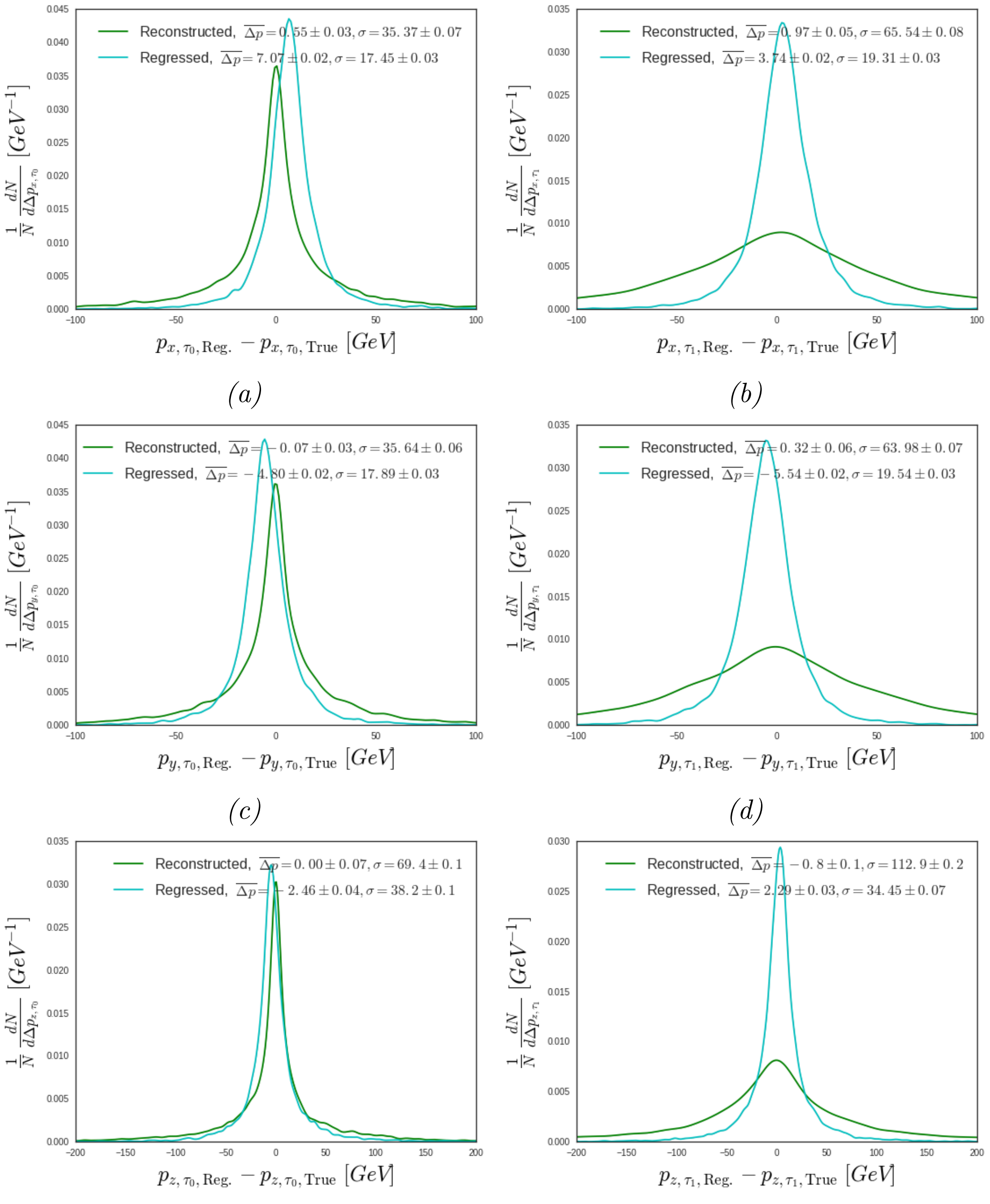
(c)

(d)

(e)

(f)

Figure 14: True and regressed τ-lepton momenta distributions for the $\mu\,\tau_h\,b\,b$ category.



(a) *(b)*

(c) *(d)*

(e) *(f)*

Figure 15: Delta distributions for τ-regressor response on τ-lepton momenta for the $\mu\,\tau_h\,b\,b$ category.

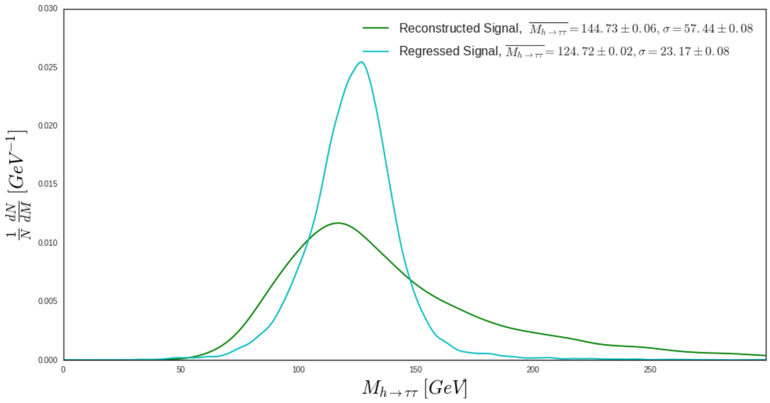


Figure 16: Reconstructed and τ-regressed distributions for the $h_{\tau\bar{\tau}}$ mass for the $\mu\,\tau_h\,b\,b$ final state.

4.6 Di-Higgs regression

Having regressed the 3-momenta of both the b jets and both the τ leptons, we can try feeding in the regressed features into a regressor to the di-Higgs mass, and check whether this three-stage regression offers any advantage over the single-stage regression performed in Sec. 4.3.

4.6.1 Feature sets

Two sets of features are considered:

- Set 0: b-regressed Cartesian 3-momenta, energy, and absolute momenta of both b jets, plus PYTHIA 8's b-quark mass, plus b-regressed Cartesian 3-momentum, energy, mass, and absolute momenta of $h_{b\bar{b}}$, plus, τ-regressed Cartesian 3-momenta, energy, and absolute momenta of both τ leptons, plus the PDG τ-lepton mass of 1.776 86 GeV [12], plus m_T, plus τ-regressed Cartesian 3-momenta, energy, mass, and absolute momenta of $h_{\tau\bar{\tau}}$ and di-Higgs, plus p_x and p_y of the missing momentum vector; 45 features in total.

- Set 1: Set 0, but using reconstruction-level features for all features: 45 features in total.

Although Set 1 contains no regressed features (similar to the single-stage regression performed earlier), it does contain extra features which were not previously used. This should make a comparison between three- and single-stage regression focus only on the effect of pre-regression by accounting for the fact that the three-stage regression uses more input features.

4.6.2 Development results

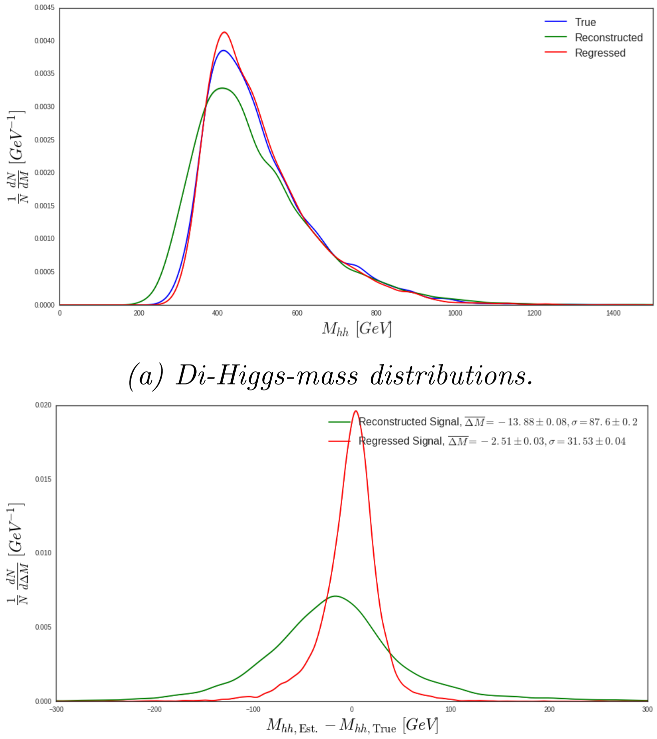
Tab. 11 shows the results for development testing of three-stage regression to the di-Higgs mass. We see that the architecture consisting of Set 0 and Model 1 demonstrates the best performance.

Architecture	Mean MSE [GeV^2]	N_{Features}
Set 0 Model 0	1070 ± 20	45
Set 0 Model 1	1040 ± 30	45
Set 1 Model 0	1730 ± 70	45
Set 1 Model 1	1760 ± 70	45

Table 11: Summary of the mean MSE values for three-stage regression to the di-Higgs mass for various architectures during development. Lower values are better.

4.6.3 Application of three-stage di-Higgs regression

Having trained the di-Higgs regressor for application, we can see from the results shown in Fig. 17a, that applying the regressor improves the estimation of the di-Higgs mass by correcting the overestimation of the low-mass region, which was presumably caused by mistreatment of the missing energy. From the delta distribution in Fig. 17b, we see that the regressor results in improvements to both the mean and the width of the distribution.



(a) Di-Higgs-mass distributions.

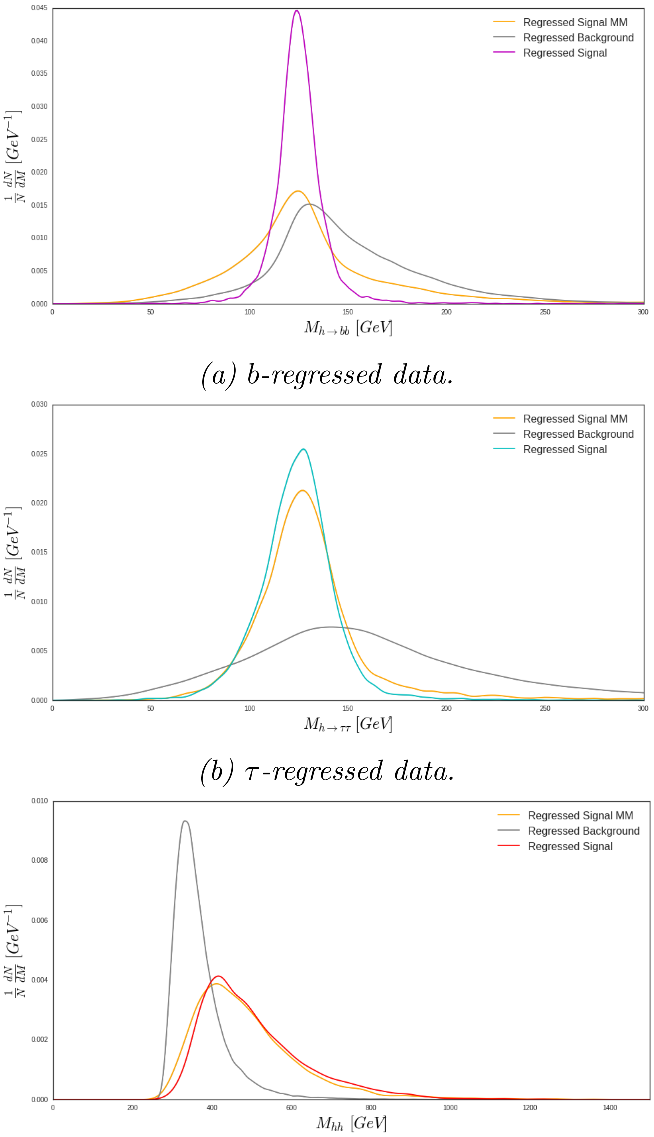
(b) Delta distribution of di-Higgs mass.

Figure 17: Comparison between distributions of true, reconstructed, and three-stage (3S) regression di-Higgs mass for the $\mu\,\tau_h\,b\,b$ final state.

4.7 Background response

Having developed regressors for signal, it is important to check their responses when fed non-signal data, in order to make sure that the responses are different, in other words, that the regressed features are still usable for classification tasks. Fig. 18 compares the responses of the three regressors on signal, mismatched signal (signal events where incorrect final-states were selected), and background data.

In Fig. 18a we see that both the b-regressed mismatched signal (orange) and the background (grey) distributions peak close to the Higgs mass, but show asymmetric distributions and suppressed peaks. Fig. 18b shows a smaller difference between signal and mismatched signal for τ-regressed data. This is because it is normally the miss-selection of b jets which causes signal events to fail the MC-truth check. The background distribution is seen to have a very wide peak around 150 GeV. In Fig. 18c we see that the three-stage di-Higgs-mass regressor has a similar response for both signal and mismatched signal data, but background data is concentrated in a region of lower mass than the signal data, thereby providing separation between signal and background.



(a) b-regressed data.

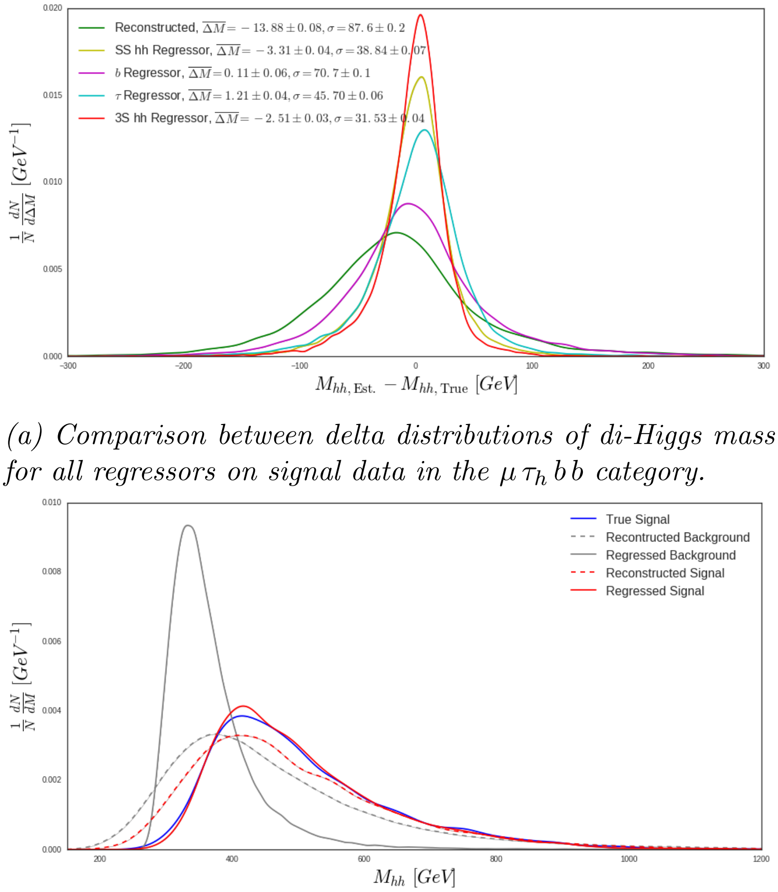
(b) τ-regressed data.

(c) Three-stage-regressed di-Higgs invariant-mass distribution.

Figure 18: Comparison between responses of regressors on signal, mismatched (MM) signal, and background for the $\mu\,\tau_h\,b\,b$ final state.

4.8 Summary of $hh \to \tau\,\bar{\tau}\,b\,\bar{b}$ regression studies

Fig. 19 concludes the regression study in $\tau\,\bar{\tau}\,b\,\bar{b}$ events. From Fig. 19a we see that each the three-stage di-Higgs mass regressor provides the most precise response and great improvements in both accuracy and precision over the use of reconstructed features. In Fig. 19b we see that applying the three-stage regressor di-Higgs mass regressor not only improves the estimate of di-Higgs mass, but also increases the separation between the signal and background distributions, which should increase the discriminating power of the feature in classification.



(a) Comparison between delta distributions of di-Higgs mass for all regressors on signal data in the $\mu\,\tau_h\,b\,b$ category.

(b) Response of the three-stage di-Higgs-mass regressor on signal and background data compared to the generator-level di-Higgs mass, in the $\mu\,\tau_h\,b\,b$ category.

Figure 19: Overall performance of regression in $h\,h \to \tau\,\bar{\tau}\,b\,\bar{b}$ events.

5 Classification of $h\,h$ signal in $\tau\,\bar{\tau}\,b\,\bar{b}$

5.1 The classifier

A multi-layer feed-forward artificial neural-network (NN) is used for the classification, with the same basic layout being adopted for all applications:

Keras [28] is used to implement a NN which consists of seven hidden layers, each containing 100 nodes. All layers are initialised according to He-normal initialisation [29] to help propagate gradients. All nodes (except output nodes) use parametrised, rectified linear-unit activation functions [29], which prevent gradients becoming saturated in the network. Output nodes use sigmoid activation, since the classification targets are either zero (background) or one (signal).

All layers (except the output) are batch-normalised [30] to account for differences in scale between input features, and to speed up training.

During training, drop-out is applied to nodes (except output nodes) with a probability of 0.2 [31]. In some architectures, Gaussian noise is applied throughout the network (except in output layer) in order to corrupt signals and reduce the possibility of over-fitting by forcing the network to generalise to the training data.

Each classifier is trained in batches of 32 events of signal data, for a maximum limit of 10 000 epochs. The data are weighted such that the sum of weights for both signal and background are equal. The Nesterov-Adam optimiser [32] is used to minimise the binary cross-entropy (BCE) of predictions (the loss function). Training can finish early if the BCE does not decrease for ten epochs and, invariably, the upper limit of 10 000 is never reached. The binary cross-entropy is calculated according to Eq. 5:

$$\mathrm{BCE} = -\frac{1}{N}\sum_{n=1}^{N}\left[y_n \log \hat{y}_n + (1-y_n)\log\left(1-\hat{y}_n\right)\right], \tag{5}$$

where N is the number of predictions, $\hat{y}_n$ is a vector of predictions (in our case the prediction of event class, $\hat{y} \in [0,1]$), and y_n is a vector of true values (here, the actual event class, $y \in \{0,1\}$). It quantises the difference between the classifier's prediction and the true class, i.e. lower values indicate better performance.

All formal comparisons between architectures will be made using their BCE values, but the accuracy of the classifiers will also be reported in order to provide a performance metric which is more easily understood by humans. The accuracy is calculated according to Eq. 6:

$$\mathrm{ACC} = \frac{N_{\mathrm{TP}} + N_{\mathrm{TN}}}{N}, \tag{6}$$

where N_{TP} is the number of true-positive predictions (correctly identified signal events), N_{TN} is the number of true-negative predictions (correctly identified background events), and N is the number of predictions; it is the fraction of correct predictions. The classifier categorises events with an output value greater than or equal to 0.5 as signal and those with a value less than 0.5 as background, e.g a signal event with a value of 0.7 would be a true-positive result and a background even with a value of 0.1 would be a true-negative result.

For development, stratified five-fold cross-validation is used and architectures are compared by their mean final BCE values, i.e. the arithmetic mean of the BCE values of each of the five classifiers is calculated. Stratified, k-fold cross-validation is similar to standard k-fold cross-validation (Sec. 4.2) except that each portion of the data set contains approximately the same fraction of classes as the full data set. This ensures that balance between classes is propagated to each training set such that each classifier might train optimally.

For training for application, data are split into 'training' and 'validation samples', with 80 % of the data being used for training and the remaining 20 % being reserved for validation. Since signal events which failed the MC-match check (referred to 'signalMM' in Sec. 4.2) are effectively another source of background to the search, these events will not be considered at this stage and are removed from the training and validation data.

Classifiers are then trained and tested ten times on the entirety of the training data, and the most performant NNs are selected. The response of the classifier is then confirmed on the validation data.

Note that the validation sample is not used to make any selection or comparison of architectures or training cycles; it is a pure hold-out sample. The same split into training and validation samples is made for each classifier, and for signal data it is the same split as was used for the regressor validation in Sec. 4.2.

Also note that hyperparameter optimisation is not the focus of this preliminary study, so only two models will be considered:

- Model 0: The basic layout described above with no Gaussian noise;
- Model 1: The basic layout described above with Gaussian noise with $\sigma = 0.5$ for applied at the first hidden layer.

Model 1 aims to test whether corrupting network signals forces the network to better generalise to the data. Model 0 will show what the ‘natural’ response of the network is. If Model 1 demonstrates the best performance, then it is possible Model 0 over fits to the data. If Model 0 demonstrates the best performance, then the Gaussian noise was not necessary and was impeding the function of the network.

Since this is a preliminary investigation aiming to validate the use of neural-networks for regression tasks in particle physics, only the $\mu\,\tau_h\,b\,b$ final-state category is investigated. This was chosen because it contains both hadronic and leptonic decays for the τ leptons, and has a higher acceptance than the $e\,\tau_h\,b\,b$ category.

5.1.1 Feature definitions

In order to train a classifier, we consider sets of features which describe the events and the final-states. For ease of reading, we collectively refer to the b jets and τ leptons as ‘the low-level objects’, and the Higgs bosons and their vector sum (the di-Higgs vector) as ‘the high-level objects’. The sets of features are defined below.

Low-level features 17 features which characterise the basic final-states in the events.

- The momenta of the low-level objects, which may be considered in the reconstructed and regressed regimes using the regressors developed in Sec. 4.2; 12 features.
- The reconstructed p_x and p_y of the missing momentum vector; 2 features.
- The reconstructed masses of both b-jets and the τ-jet; 3 features.

High-level features 13 features which characterise the high-level objects in the events.

- The momenta of the high-level objects, which may be considered in the reconstructed and regressed regimes; 9 features.
- The masses of the high-level objects, which may be considered in the reconstructed and regressed regimes; 3 features.
- m_T; 1 feature.

Momenta and energy features 14 features which offer more compact information on the kinematics of objects in the events. All of these may be considered in both the reconstructed and regressed regimes.

- The absolute momenta ($|p|$) of the low-level objects; 4 features.
- The absolute momenta ($|p|$) of the high-level objects; 3 features.
- The absolute energy (E) of the low-level objects; 4 features.
- The absolute energy (E) of the high-level objects; 3 features.

Final-state momenta-difference features 50 features which relate the momenta of objects to one another. We consider the addition of the element-wise difference in pairs of objects' 3-momenta and the twist of pairs of objects. The twist between objects i and j is calculated according to Eq. 7:

$$\tau_{i,j} = \tan^{-1} \frac{\Delta\phi_{i,j}}{\Delta\eta_{i,j}}, \tag{7}$$

where $\Delta\phi_{i,j}$ and $\Delta\eta_{i,j}$ are the difference in ϕ and η between objects i and j, respectively [33]. Again, these may be considered in both the reconstructed and regressed regimes.

- The difference in momenta for all combinations of low-level objects and $\overrightarrow{p}_T^{\text{miss}}$; 26 features.
- The difference in momenta for all combinations of high-level objects and $\overrightarrow{p}_T^{\text{miss}}$; 15 features.
- The twist for all combinations of low-level objects; 6 features.
- The twist for all combinations of high-level objects; 3 features.

Multiplicity features 4 features which help characterise global event by returning the multiplicity of physics objects.

- N_{jets}, the number of jets in the event.
- $N_{b-\text{jets}}$, the number of b-tagged jets in the event.
- $N_{\tau-\text{jets}}$, the number of τ-tagged jets in the event.
- N_{γ}, the number of photons in the event.

Global kinematic-features 13 features which help characterise global event by returning the kinematics of physics objects.

- $\text{Min}\,(p_T)$, $\overline{p_T}$, and $\text{Max}\,(p_T)$, the event-wise minimum, mean, and maximum p_T of jets, respectively; 3 features.
- $\text{Min}\,(\eta)$, $\overline{\eta}$, and $\text{Max}\,(\eta)$, the event-wise minimum, mean, and maximum η of jets, respectively; 3 features.
- $\text{Min}\,(M)$, $\overline{M}$, and $\text{Max}\,(M)$, the event-wise minimum, mean, and maximum invariant mass of jets, respectively; 3 features.
- H_T, the scalar sum of the transverse energy of all jets.
- s_T, the scalar sum of $\overrightarrow{p}_T^{\text{miss}}$, light-lepton p_T, photon p_T, and H_T.
- E_{vis}, the sum of visible energy.
- Centrality, the scalar sum of the p_T of all objects divided by E_{vis}.

Event-shape features Up to 12 features which characterise the shape of events, defined using the eigenvalues of the following tensors:

$$\text{sphericity tensor} = \frac{1}{\sum_i |\overrightarrow{p_i}|^2} \sum_i \begin{bmatrix} p_{i,x}p_{i,x} & p_{i,x}p_{i,y} & p_{i,x}p_{i,z} \\ p_{i,y}p_{i,x} & p_{i,y}p_{i,y} & p_{i,y}p_{i,z} \\ p_{i,z}p_{i,x} & p_{i,z}p_{i,y} & p_{i,z}p_{i,z} \end{bmatrix}, \tag{8}$$

and

$$\text{spherocity tensor} = \frac{1}{\sum_i |\overrightarrow{p_i}|} \sum_i \left[\frac{1}{|\overrightarrow{p_i}|} \begin{bmatrix} p_{i,x}p_{i,x} & p_{i,x}p_{i,y} & p_{i,x}p_{i,z} \\ p_{i,y}p_{i,x} & p_{i,y}p_{i,y} & p_{i,y}p_{i,z} \\ p_{i,z}p_{i,x} & p_{i,z}p_{i,y} & p_{i,z}p_{i,z} \end{bmatrix} \right], \tag{9}$$

where $\sum_i$ is a summation over either all objects in the event or just the low-level objects. The eigenvalues of the tensors are then calculated, ordered, and normalised such that: $\lambda_1 \geq \lambda_2 \geq \lambda_3$ and $\lambda_1 + \lambda_2 + \lambda_3 = 1$. The following features are then calculated according to Ref. [34]:

- Sphericity and spherocity: $S = \frac{3}{2}(\lambda_2 + \lambda_3)$.
- Aplanarity and aplanority: $A = \frac{3}{2}\lambda_3$.
- Υ of sphericity: $\Upsilon = \frac{\sqrt{3}}{2}(\lambda_2 - \lambda_3)$.
- Shape of spherocity: $D = 27\lambda_1\lambda_2\lambda_3$

5.2 Feature selection

In this section we aim to decide on which features from Sec. 5.1.1 should be used as inputs to our classifier in order to get the greatest performance for the fewest number of input features. Tab. 12 lists the results for all development tests, and in the remainder of the section we aim to lead readers through the tests which were performed. Since these tests involve a large number of features, there is the possibility that the classifier might over fit to the data. Because of this we initially only consider Model 1, since the Gaussian noise should force the network to generalise to the data. Having selected the smallest, most performant feature set we will then test the response of Model 0, to see whether the Gaussian noise is necessary.

In order to help highlight the significance of results, we perform hypothesis tests and report the p-values. The exact statement of the hypotheses varies from stage to stage, and will be described in the documentation of each stage.

Architecture	Mean BCE	Mean Accuracy [%]	N_{features}	P-value
First stage: Coordinate system				
Set 0	0.199 ± 0.004	91.9 ± 0.2	17	-
Set 1	0.1371 ± 0.0009	94.71 ± 0.07	17	-
Second stage: All features				
Set 2	0.107 ± 0.001	95.81 ± 0.06	123	-
Set 3	0.1123 ± 0.0007	95.60 ± 0.04	123	-
Set 4	0.119 ± 0.002	95.2 ± 0.1	123	-
Set 5	0.111 ± 0.0011	95.64 ± 0.05	123	-
Third stage: Removal of momenta and energy features				
Set 6	0.115 ± 0.002	95.52 ± 0.07	116	0.00
Set 7	0.110 ± 0.001	95.72 ± 0.06	113	0.13
Fourth stage: Removal of final-state momenta-difference features				
Set 8	0.1087 ± 0.0009	95.71 ± 0.04	97	0.21
Set 9	0.108 ± 0.001	95.76 ± 0.08	108	0.35
Set 10	0.1097 ± 0.0006	95.72 ± 0.04	117	0.06
Set 11	0.107 ± 0.001	95.82 ± 0.05	120	-
Fifth stage: Removal of global and shape features				
Set 12	0.110 ± 0.002	95.66 ± 0.06	114	0.12
Set 13	0.1094 ± 0.0004	95.70 ± 0.04	114	0.10
Set 14	0.1088 ± 0.0003	95.75 ± 0.05	107	0.18
Set 15	0.109 ± 0.001	95.70 ± 0.03	116	0.17
Final stage: Model selection				
Set 11 Model 0	0.119 ± 0.003	95.4 ± 0.1	120	-
Set 11 Model 1	0.107 ± 0.001	95.82 ± 0.05	120	-

Table 12: Summary of classifier-development tests.

5.2.1 First stage: Coordinate system

In Sec. 4 the 3-momenta were transformed into a Cartesian coordinate system. These first tests aim to decide on whether to use 3-momenta in terms of p_T, η, and ϕ or p_x, p_y, and p_z. In order to accentuate the effect of altering the coordinate system, we will just consider low-level features.

With this in mind, we consider two sets of features:

- Set 0: Reconstructed p_T, η, and ϕ of the low-level objects, reconstructed p_T and ϕ of the missing momentum vector, and reconstructed masses of both b-jets and the τ-jet; 17 features in total.
- Set 1: Set 0, but using Cartesian coordinates for 3-momenta (p_x, p_y, and p_z); 17 features in total.

Due to both coordinate systems being equally complex, no justification for picking one over the other is required other than performance, so we will not perform hypothesis tests at this point.

From the results in Tab. 12, we see that Set 1 (Cartesian coordinates) outperforms Set 0 (p_T, η, and ϕ), so we will proceed to use momenta in the Cartesian coordinate system.

5.2.2 Second stage: All features

The classification power of neural networks stems from their ability to discover high-dimensional patterns in data. In order to ensure that the classifier is a powerful as possible, we will feed in all available features and then proceed to remove sets of features if it is shown that doing so does not significantly reduce the classifier's performance. We consider the follow sets of features:

- Set 2: All features, working in the Cartesian system with regressed momenta and reconstructed masses; 123 features in total.
- Set 3: All features, working in the Cartesian system with regressed momenta and regressed masses; 123 features in total.
- Set 4: All features, working in the Cartesian system with reconstructed momenta and reconstructed masses; 123 features in total.
- Set 5: All features, working in the Cartesian system with reconstructed momenta and regressed masses; 123 features in total.

From Tab. 12 we see that Set 2 (regressed momenta and reconstructed masses) demonstrates the greatest classification power (has the lowest BCE).

5.2.3 Third stage: Removal of momenta and energy features

We will now proceed to attempt to remove sets of features from Set 2 in order to report the smallest, most performant set of features. Feature sets will be removed if it is shown that doing so results in no significant drop in performance. We define 'significant' by performing a one-tailed t-test at a significance level of 0.9 for the null hypothesis that the loss of the proposed set is greater than the loss of Set 2, against the alternative hypothesis that the loss of Set 2 and the proposed set are equal. In the case that the loss of the proposed set is less than the loss of Set 2, we will proceed to accept that as the null set. Whilst this test induces a high probability of type-I errors (retention of features which do not aid classification), we believe that the occurrence of a type-II error (removal of discriminant features) is a risk worth taking measures to avoid.

- Set 6: Set 2 minus $|p|$ of the regressed low-level objects, and the regressed high-level objects; 116 features in total.
- Set 7: Set 2 minus E of the regressed low-level objects, and the regressed high-level objects; 113 features in total.

From Tab. 12 we see that removal of the features results in a significant drop in performance, therefore we will continue to use them.

5.2.4 Fourth stage: Removal of final-state momenta-difference features

Continuing our attempts to reduce the number of features in the most performant set, we now examine the removal of the momenta difference and twist features.

- Set 8: Set 2 minus the difference in momenta for all combinations of regressed low-level objects and $\overrightarrow{p}_T^{\text{miss}}$; 97 features in total.

- Set 9: Set 2 minus the difference in momenta for all combinations of regressed high-level objects and $\overrightarrow{p}_T^{\text{miss}}$; 108 features in total.

- Set 10: Set 2 minus the twist for all combinations of regressed low-level objects; 117 features in total.

- Set 11: Set 2 minus the twist for all combinations of regressed high-level objects; 120 features in total.

We see, in Tab. 12, that Sets 8, 9 and 10 demonstrate significant drops in performance, however Set 11 demonstrated better performance than Set 2, and contains fewer features, therefore we proceed to accept Set 11 as the most performant set.

5.2.5 Fifth stage: Removal of global and shape features

In the final attempt to reduce the number of features in the most performant set, we now consider the removal of the global and shape features. We update our hypothesis test to use Set 11 as the null set.

- Set 12: Set 11 minus the shape features for the entire event; 114 features in total.
- Set 13: Set 11 minus the shape features for the low-level objects; 114 features in total.
- Set 14: Set 11 minus the global kinematic features; 107 features in total.
- Set 15: Set 11 minus the multiplicity features; 116 features in total.

Tab. 12 shows that all the new sets demonstrate significant drops in performance, therefore we proceed to keep Set 11 as the most performant set.

5.2.6 Final stage: Model selection

Having established the smallest, most performant set of features, we proceed to choose between the two models: Model 1 the model tested so far, which applies Gaussian noise to help force the classifier to generalise to the data; and Model 0 the basic model, which does not apply Gaussian noise. From the comparison in Tab. 12, we see that Model 1 outperforms Model 0, therefore the final architecture for application is Model 1 and Set 11.

5.3 Application

It should be reported that in the development section (Sec. 5.2), the intention had been to balance the training samples such that the summed weights of signal and background were equal. This was to be achieved by weighting each background event by the reciprocal of the number of background events, and by weighting each signal event by the reciprocal of the number of signal events. By accident, however, each sample was weighted by the reciprocal of the size of the other, increasing the imbalance between classes in the data. It is assumed that conclusion of Sec. 5.2 (that Model 1 Set 11 is the optimal architecture) would remain unchanged if the intended weighting had been implemented, and that the classifiers would simply have received a slight increase in performance. Indeed, by checking a weighting-independent metric (the *area under ROC curve*) for Model 1 Set 11 we find that both schemes result in similar performance: 0.9928 ± 0.0001 (intended scheme) versus 0.9927 ± 0.0002 (development scheme). The authors feel that due to this similarity, the development tests in Sec. 5.2 do not need to be repeated. In training the classifier for application we will move to using the intended weighting scheme.

As described Sec. 5.1, the architecture consisting of Model 1 and Set 11 is trained and tested ten times on the training data. We then form an ensemble of the three networks which demonstrated the lowest BCE values on the training data. The outputs are weighted according to one minus their BCE values. The ensemble prediction is then the weighted sum of each networks' output divided by the sum of weights. This method of ensembling allows the most performant network to have the strongest influence over the final prediction, but to be supported by less-performant networks in hard-to-classify events. Indeed, by comparing the ROC integrals on training data for each classifier and the ensembled classifier in Tab. 13 we see that the ensemble outperforms any of the its sub-components.

Classifier	ROC integral
1st classifier	0.9954
2nd classifier	0.9952
3rd classifier	0.9951
Ensembled classifier	0.9958

Table 13: Performance of the top three most performant classifiers on training data, and their weighted ensemble. Performance measured by the ROC integral; values closest to one are better.

The ensembled classifier is then applied to the validation data in order to predict the class of each event; signal or background. The distribution of the classifier output is shown in Fig. 20. We observe that signal and background are well clustered towards one and zero, respectively, indicating that the classifier is able to separate well the classes of events. In Fig. 21 we plot the receiver operating characteristic (ROC) curve. This illustrates the trade-off between signal acceptance against background acceptance. The area of under the ROC curve should be as close to one as possible; this would correspond to perfect separation between signal and background. We report an area of $0.993\,66 \pm 0.000\,03$ on the validation data.

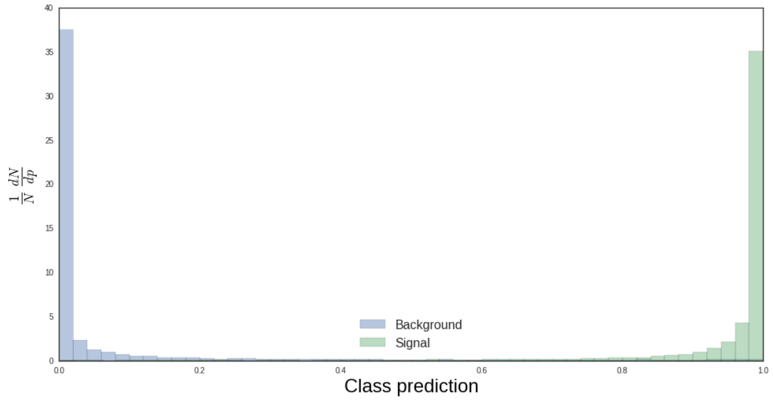


Figure 20: Class predictions of the classifier on validation data. 0 indicates 'background-like' and 1 indicates 'signal-like'. Both signal and background are normalised to one.

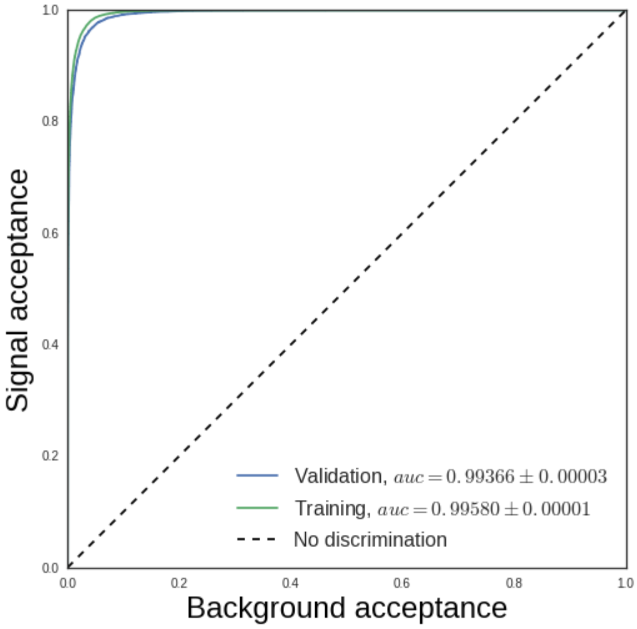


Figure 21: ROC curve for the classifier on both training and validation data.

5.4 Summary of $hh \to \tau\,\bar{\tau}\,b\,\bar{b}$ classification studies

From the studies performed in this section, we find that neural networks are highly applicable to the separation of signal ($h\,h \to \tau\,\bar{\tau}\,b\,\bar{b}$) and background (fully-leptonic $t\,\bar{t}$) contributions in the $\mu\,\tau_h\,b\,b$ final state. Taking Fig. 20 and normalising signal and background to their cross-section times event acceptance, we find in Fig. 22 that the background still contributes heavily at even high values of the class prediction. An important process, which was not considered in this preliminary study, is hyper-parameter optimisation (number of layers, number of nodes, *et cetera*). Perhaps by adjusting these in a dedicated study, the separation between signal and background might be further improved and the number of required input-features reduced.

Since the fine-tuning of the classifier parameters is likely to be extremely sensitive to the accuracy of the simulated data, these studies should be performed on data which has been passed through a full detector-simulation, such as Geant 4 [18, 19]. The switch to fully simulated data is expected to cause a drop in the baseline performance of the classifier This, however, will leave room for measurable improvements in classification power since, as seen from the results here, the ROC curve is already close to saturation.

It is important to remember that fully-leptonic $t\,\bar{t}$ is not the only source of background in our search, and that other sources will contribute significantly. These must also be investigated.

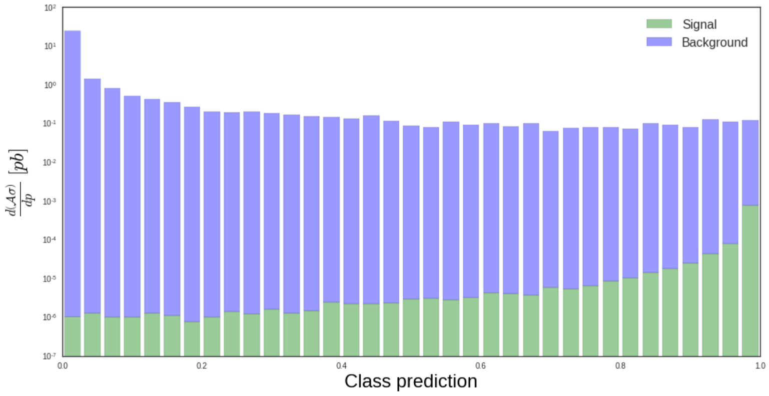


Figure 22: Class predictions of the classifier on validation data. 0 indicates 'background-like' and 1 indicates 'signal-like'. Both signal and background are normalised to their respective values of cross-section times event acceptance. The background distribution is stacked on top of the signal distribution.

6 Classification of $h\,h$ signal in $b\,\bar{b}\,b\,\bar{b}$

In the following, the study of different multivariate analyses (MVA), used for the classification of the $hh \to b\bar{b}b\bar{b}$ signal among the most prominent background, is reviewed.

6.1 Deep Neural Network classifier (SGD)

The first attempt at classifying the signal process versus background is performed by applying a multi-layer feed-forward artificial neural-network (NN). In particular, this is a deep NN which uses a stochastic gradient descent (SGD) algorithm to minimise the objective function $J(\theta)$ parameterized by a model's parameters, θ, by updating the parameters in the opposite direction of the gradient of the objective function $\nabla_\theta J(\theta)$ with respect to the parameters. The learning rate η determines the size of the steps we take to reach a local minimum. The NN is implemented with Keras [28], a high-level neural networks library written in Python.

The NN uses the 30 features identified in Sec. 3.2.1 as input. For the sake of clarity, these features are listed below.

For each of the four selected b-jets:

- the transverse momentum p_T;
- the energy E;
- the angular variables η and ϕ.

For each of the two Higgs boson candidates:

- the transverse momentum p_T;
- the energy E;
- the angular variables η and ϕ;

- $\Delta\eta$ and $\Delta\phi$, the angular distances between the two jets forming the reconstructed Higgs boson;
- the invariant mass M_{inv}.

The dataset is split into training and validation samples, composed by 80% and 20% of total events, respectively. The first step is the standardisation of the features, in such way that they are centered around zero with a standard deviation of one. This procedure is important for the comparison of measurements which have different units, and it is also a general requirement for many machine learning algorithms which might behave incorrectly if the individual features are not normally distributed. Once the features have been standardised, a statistical procedure called principal component analysis (PCA) is applied with the aim of detecting the correlation between variables. The PCA uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables.

We define the NN using a *sequential* model: a linear stack of five hidden layers, each containing 100 nodes. The initialization defines the way to set the initial random weights of KERAS layers. For all the nodes, with the exception of the output node, we use a Parametric Rectified Linear Unit (PReLU) [29] as the activation function, which adaptively learns the parameters of the rectifiers and prevents the saturation of the gradients in the network. The output nodes use the logistic sigmoid function which allows binary outputs, 0 for the background and 1 for the signal. The activations are normalised throughout *Batch Normalisation* which is a transformation that maintains the mean activation close to zero and the activation standard deviation close to one. We also apply an additive, zero-centered Gaussian noise with standard deviation 0.5 to the input. This procedure is useful to mitigate overfitting. Dropout is also applied to nodes (except output nodes) with a probability of 0.2 [31]. Dropout is a popular regularisation technique with deep networks, where network units are randomly masked during training.

The NN is trained for 50 epochs in batches of 32 events. Because of sample unbalance, the background events have been weighted with the number of signal events in order to have an equal sum of weights in both cases. Finally, we use the logarithmic loss function or *binary cross-entropy* (BCE) during training, the preferred loss function for binary classification problems defined in Eq. 10. The model also uses the efficient Nesterov-Adam optimisation algorithm to minimise the BCE and the accuracy metrics will be collected when the model is trained.

$$\mathrm{BCE} = -\frac{1}{N}\sum_{i=1}^{N}\left[y_n \log \hat{y}_n + (1-y_n)\log(1-\hat{y}_n)\right], \tag{10}$$

In Eq. 10, N is the number of samples, $\hat{y}_n$ is a vector of predictions, and y_n is a vector of true values. In Fig. 23 the output scores of the NN run over the validation data are reported. Intuitively, the more differentiated the distribution of the NN output is for signal and background events, the more efficient the discrimination will be.

We observe a good discrimination between signal and background obtained with the NN. The result can be quantified by making use of the ROC (Receiver Operating Characteristic) curve in Fig. 24. The ROC curve illustrates the performance of a binary classifier system as its discrimination threshold is varied. The curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings. The best way to summarize ROC performance in a single number is its Area Under the Curve (AUC). Hypothetically, a unitary value of the ROC AUC would represent a perfect discrimination of the signal among background. The ROC AUC obtained with the NN is:

$$\mathrm{ROC\ AUC} = 0.92130. \tag{11}$$

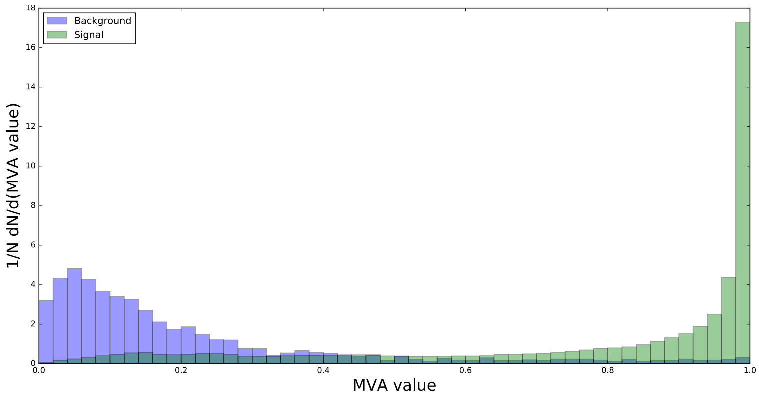


Figure 23: Output scores of the NN classification over the validation data.

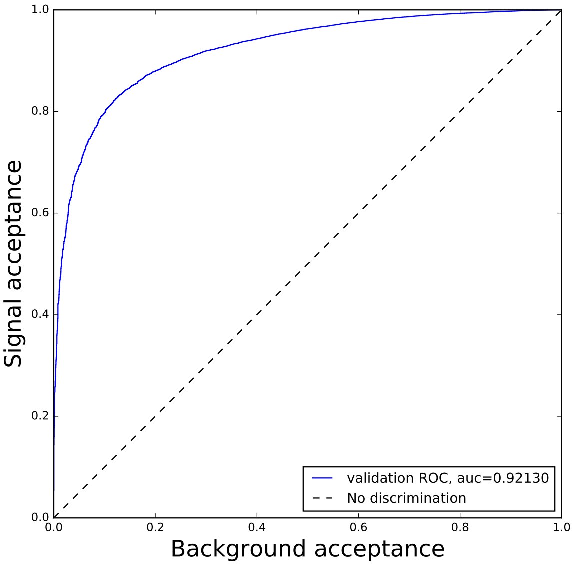


Figure 24: The ROC curve summarising the performance of the NN.

6.2 Boosted Decision Tree classifier

To benchmark the result obtained with the NN we also trained a XGBoost (eXtreme Gradient Boosting [35]) algorithm, which is an implementation of gradient-boosted decision-trees in Python. We run the XGBoost over the same features selected for the NN and defined in Sec. 3.2.1 and Sec. 6.1. The dataset is split into a training set (80%) and a validation set (20%). Before training the XGBoost, we standardise and decorrelate the features following the same procedure used in Sec. 6.1. We then tune some fundamental parameters which define the gradient-boosted decision-tree model. We set the number of boosting stages to perform the XGBoost to 1000 and the learning rate of the algorithm to 0.1. Gradient boosting is fairly robust to over-fitting so a large number usually results in better performance. The maximum depth limits the number of nodes in the tree, the best value depends on the interaction of the input variables. We set this parameter to 5. We specify the learning task and the corresponding learning objective as "binary:logistic". This means that the output predictions are probability confidence scores in the [0,1] interval, corresponding to the probability that the event originates from the signal process.

In Fig. 25 the result of the XGBoost in terms of the multivariate analysis (MVA) output is illustrated.

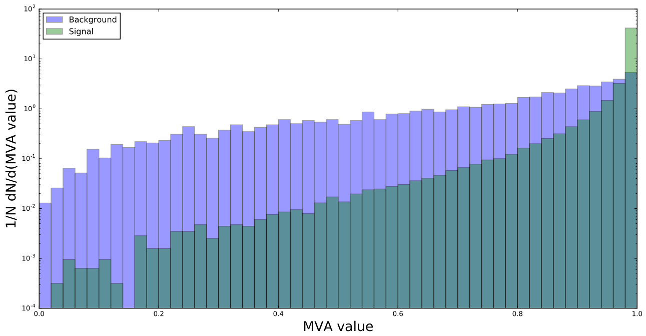


Figure 25: Output scores of the XGBoost classification over the validation data.

As in the case of the NN we evaluate the performance of the XGBoost classifier by plotting the ROC reported in Fig. 25.

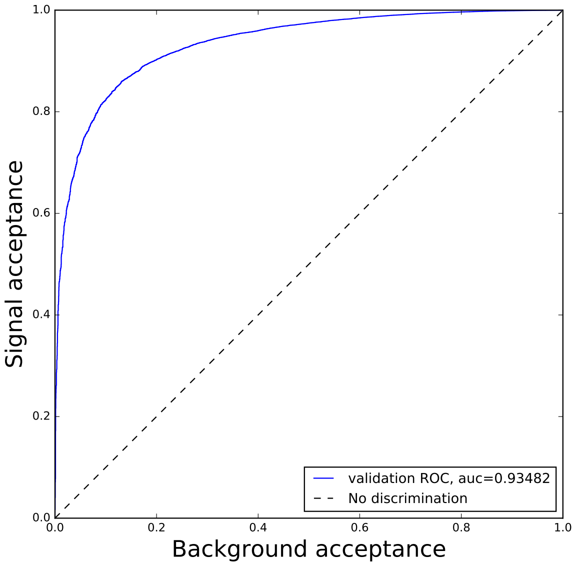


Figure 26: The ROC curve summarising the performance of the XGBoost.

The value of the ROC AUC is now

$$\text{ROC AUC} = 0.93482. \tag{12}$$

6.3 Deep Neural Network classifier (Genetic Algorithm)

A second neural network (NN) has been evaluated to discriminate between the signal and the background. In this NN the minimisation of the objective function is achieved using a Genetic Algorithm (GA), which is a non-deterministic minimisation strategy suitable for the solution of complex optimization problems, for instance when a very large number of quasi-equivalent minima are present [36]. The neural network is characterized by 17 input variables and two hidden layers of 5 and 3 nodes respectively. In every node, a sigmoid activation function evaluates the inputs of the previous layer and provides an output in the range $[0, 1]$ that can be interpreted as the probability of having a signal event. The input variables of the neural network are:

- the transverse momenta of the reconstructed Higgs candidates;
- the transverse momentum of the reconstructed Higgs pair.
- the invariant masses of the reconstructed Higgs candidates;
- the invariant mass of the reconstructed Higgs pair;
- the energies of the reconstructed Higgs candidates;
- the separation in the $\phi-\eta$ plane between the two reconstructed Higgs: $\Delta R = \sqrt{\Delta\eta^2 + \Delta\phi^2}$;
- the transverse momenta of the four b-jets coming from the decays of the two Higgs;
- the separation in η ($\Delta\eta$) between the two b-jets inside every reconstructed Higgs;
- the separation in ϕ ($\Delta\phi$) between the two b-jets inside every reconstructed Higgs.

The training of the neural network is performed on half of the MC samples that have passed the selection described in Sec. 3.2.1. The other half of the MC samples have been used to cross-validate the training in order to avoid over-training. The algorithm exploited for the training of the neural network is a Genetic Algorithm (GA), which minimises a cross-entropy function. This type of algorithm is based on the theory of evolution and natural selection [37].
In order to have an equal sum of weights, the number of background events have been rescaled to the number of signal events. The scaled distribution of signal and background events as function of the NN response is reported in Fig. 27.

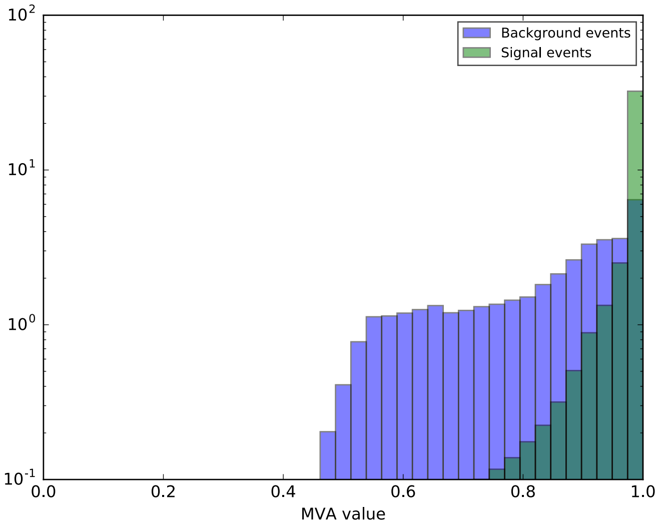


Figure 27: Output scores of the NN classification over the validation data.

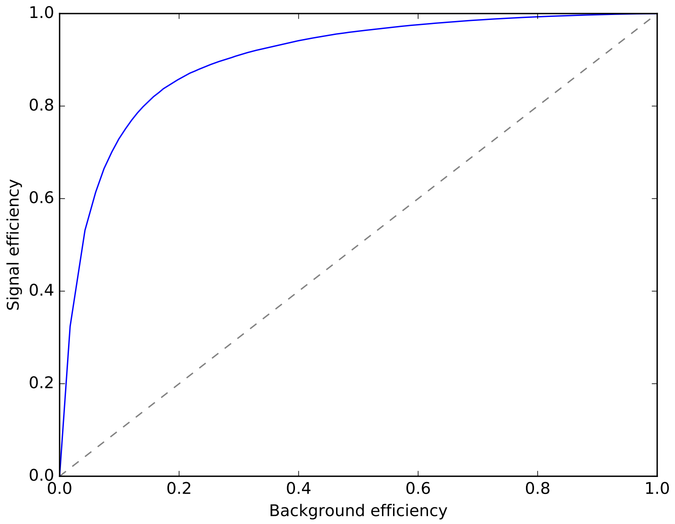


Figure 28: The ROC curve summarising the performance of the neural network (GA).

The ROC curve providing the graphical representation of the performance of the NN is shown in Fig. 28. For this NN, the area under the ROC curve is:

$$\text{ROC AUC} = 0.90696 \tag{13}$$

6.4 Summary of $hh \to b\bar{b}b\bar{b}$ classification studies

By looking at the previous results in terms of ROC AUC, we can compare the performances of the classifiers. In general, all the reviewed MVA provide good results, with the BDT outperforming the two neural networks.

The GA-based NNs are more robust with respect to the potential issue of getting stuck in local minima but they cannot be applied to minimisation problems with a large number of parameters, as the deep neural network model considered in Sec. 6.1. In particular, the SGD NN uses $O(10^5)$ free parameters, which makes it more complex than the GA which only uses $O(10^2)$ free parameters.

Tab. 14 summarises the results provided by the three classifiers used for our $h\,h \to b\,\bar{b}\,b\,\bar{b}$ studies, in terms of ROC AUC value and complexity. For the NNs the complexity is defined by the number of free parameters, while for the BDT classifier the complexity is specified by the number of boosting stages (number of trees) needed to perform it and the number of nodes in the trees (max depth).

Table 14: MVA results in terms of ROC AUC (second column). In the table is also indicated the complexity of the MVA architecture, in terms of free parameters for the NNs and .

Classifier	ROC AUC	Complexity
SGD NN	0.92130	10^5 free parameters
BDT XGBoost	0.93482	1000 estimators, depth 5
GA NN	0.90696	10^2 free parameters

7 Feature regression in $b\,\bar{b}\,b\,\bar{b}$

Motivated by the results obtained when regressing generator level observables for the $\tau\,\bar{\tau}\,b\,\bar{b}$ final state, here we extend those studies to the $hh \to b\,\bar{b}\,b\,\bar{b}$ process. The main goal will be to verify whether the regressed variables can outperform those estimations obtained by classical event reconstruction techniques.

Given the great similarities between this problem and one discussed in Sec. 4, instead of repeating the studies comparing different strategies and models, only the best performing approach and model for that scenario will be considered here. Therefore, a three-stage regression procedure will also be carried out, analogous to the best performing approach for $hh \to \tau\,\bar{\tau}\,b\,\bar{b}$ regression. However, is relevant to point out that in this case the two initial stages both correspond to $b\,\bar{b}$ regression, for the b-quark pairs decaying from each of the Higgs bosons.

In the following subsection, a more detailed description of the research problem that we are trying to solve and the proposed approach which will be followed are included. Then, the procedure and results obtained for each stage and the effect of these techniques on background events will be discussed before summarizing the conclusions obtained and possible future work in Sec. 7.7.

7.1 Research problem and proposed solution

Modern general purpose collider experiments, such as CMS and ATLAS at the LHC, have in place a sequence of complex algorithms in order to analyze the low-level detector response for each event acquired and generate a compact but much higher level representation through a process referred to as event reconstruction. At the end of this hierarchical procedure, all event information is reduced to a set of physics objects (i.e. data structures) of different types: electrons, muons, taus, jets, photons and MET. The variables associated with each of these physics objects include a four-vector (p_T, η, ϕ and E in hadron collider coordinates) and possibly some type-dependent features (e.g. charge sign or number of tracks). This event representation is preferred for carrying out data analyses because it represents directly the final state particles produced in the event and therefore is much easier to map to the underlying theoretical physical processes.

However, some useful information can potentially be lost and a few reconstructed objects can be artificially created (i.e. fake, not matching the true particles) through the mentioned data reduction algorithms. In addition, a large calibration effort, based on simulation and verified with collider data, is usually required in the experiments to adjust the reconstructed-object variables such that they are as close as possible to the generator-level quantities. Then, the whole event reconstruction process is the combination of some physically motivated algorithms with extensive parametric calibration. Therefore, when only the simulation-based calibration is considered, the event reconstruction problem can also be thought of as a supervised machine-learning regression-problem for predicting the high-level generator object variables given the lower-level detector information.

The advantage of using flexible machine-learning regression-techniques, such as gradient boosting or multilayer neural-networks, instead of simpler parametric-fitting techniques is that given enough training data they could use non-linear relationships between input variables to improve the prediction accuracy. In particular, deep neural networks (DNN) seem to be a good family of regression models for this problem, given the outperforming results observed in a diverse range of regression tasks during the last few years. The usefulness of these kinds of models for high-energy-physics data stems from their ability to learn high-level representations directly from low-level features [38] and that they can also deal with heterogeneous and variably sized input data (e.g. tracks, calorimeter deposits) [39, 40].

Because in the studies presented for this report neither experiment data nor complete detector simulation is used, but a simplified simulation framework as described in Sec. 2.4, detailed detector-level information cannot be produced. Until we can openly use full-detector simulations for publicly accessible studies, we have decided to simplify our research problem to the regression of generator level variables using relevant reconstructed event variables. By using not only the reconstructed variable corresponding to the generator-level variable to be modelled but also additional variables potentially correlated, we aim to obtain regressed variables which can more accurately represent generator-level behaviour and likely be used to perform more powerful statistical inference.

We want to study the applicability of these techniques to the $hh \rightarrow b\,\bar{b}\,b\,\bar{b}$ analysis, therefore we are going to use as a training sample the signal dataset presented in Sec. 2. The selection is analogous to the one described in Sec. 3.2.1, the only differences being that not only the combination of the four highest p_T jets but all of those that are b-tagged are used in the Higgs-pairing algorithm and that the Higgs-candidate invariant-mass cut is removed. Because the generator-level object variables of the b-quark and derived variables will be the target of the regressions, they have to be uniquely specified and in a consistent order. This is done by first matching (i.e. $\Delta R \leq 0.5$) reconstructed jets with the partons coming from the Higgs decays (before PYTHIA hadronization), then an event will only be used for training and testing if each pair of jets from each Higgs candidate is uniquely matched with their gen-level hadrons and the rest of events are filtered out.

At the end of the selection, pairing, and matching mentioned before, each event will be composed of two pairs of jets at reconstruction level and two pair of b-quarks at generator level, one pair from each Higgs boson in both cases. Generator level and reconstruction level for each Higgs-boson candidate can be then easily computed and compared. The same applies to the kinematic observables of the four-body (i.e. $h\,h$) object. The Higgs candidates' invariant masses and the $h\,h$ invariant mass are three simple variables that can provide, by themselves, good signal and background separation for $h\,h$ and are thoroughly used in most published CMS and ATLAS analyses to date [41, 42]. However, especially when the $h\,h$ final state includes jets or missing energy, reconstruction-level distributions of those variables are smeared out and those differences are due to detector-resolution effects and imperfect calibration.

In this section, the regression of the three variables mentioned before will be studied. This will be done in three stages, somewhat in line with the best result in Sec. 4. Firstly, a multivariate regression of the kinematic variables of the b-quarks originating from the decay of the leading Higgs-boson (i.e. highest reconstructed p_T) will be carried out. The advantage of regressing the b-quark kinematics instead of the mass directly is that the generator-level mass is a very peaky feature (delta-like) which causes complications when training, while the b-quark features follow a smooth distribution. The second stage will consist on a similar approach but for the b-quarks from the trailing Higgs-boson. The regressed variables can be used to have an alternate estimation of the Higgs invariant-mass which will only depend on reconstruction-level information. Finally, the $h\,h$ invariant-mass will be directly regressed, re-using all the features engineered in the previous steps.

7.2 Regression model and training details

A multi-layer feed-forward neural-network, implemented using KERAS an with a layout analogous to the one described in Sec. 4.2 (three hidden layers of 150, 130 and 100 units with He-normal inilialization and PReLU activation functions [29]) will be used for all regression problems addressed in this section. The input and output node size depends on the number of input features and regression targets.

Batch normalization [30] is added to every layer to speed up training. The loss function to be minimized using Nesterov-Adam optimizer [32] is the mean squared error (MSE) for a mini-batch of samples of size M:

$$\text{MSE} = \frac{1}{M}\sum_{m=1}^{M} \left(\hat{y}_m - y_m\right)^2 , \tag{14}$$

where $\hat{y}$ is a vector of predictions and y is a vector of true values. After experimentation with stochastic mini-batches of different sizes, larger batch sizes where found to achieve better performance and be more stable during training, so $m = 256$ was used for all the models shown in this work.

A total of 1M events from the original sample were available after selection and generator-level matching, which were randomly split in a training set with 80% of the samples and test set including the remaining. Given the large amount of training data, it is possible that larger models (more layers or units) could achieve better performance at a higher computational cost, which could be studied as an extension of this work.

7.3 1$^{\text{st}}$ step: leading $h \to b\,\bar{b}$ kinematic regression

For this regression, the generator-level 3-momenta in Cartesian coordinates of each the b-quarks which are the decay products of the leading Higgs-boson are used as target features. A total of 44 input features were used, composed of the following reconstructed event-variables:

- The 3-momenta in Cartesian coordinates, energy, mass, and momenta modulus of every jet selected after pairing (4 jets, 24 features).
- The 3-momenta in Cartesian coordinates, energy, mass, and momenta modulus of each reconstructed Higgs candidate (2 Higgs candidates, 12 features).
- The 3-momenta in Cartesian coordinates, energy, mass, and momenta modulus of the reconstructed $h\,h$ candidate (6 features).
- The transverse momenta in Cartesian coordinates of the missing transverse energy of the event (2 features).

After training the regression model described in 7.2 for 200 epochs in mini-batches of 256 samples, the distributions of the regressed targets are shown together with the generator-level truth and the corresponding reconstructed features for the test set in Fig. 29. The target feature behaviour is well captured by the regressed features. For the second b-quark, ordered according to the p_T of the matched reconstructed object so it can be applied to data, the probability density is clearly better reproduced by the regressed kinematic variables.

For easier comparison between the regressed and reconstructed features, the distributions of their differences with respect to the generator-level target are shown in Fig. 30. The coordinates p_x and pz of both b-quarks are well modelled, outperforming the corresponding reconstructed variables. However small, but significant, biases are observed for p_z, especially for the first b-quark. This might indicate that the model chosen is not flexible enough or that it could potentially converge to a better minimum. While this will likely not represent a problem for the study presented here, it should be examined in greater detail in future work.
Using the regressed p_x, p_y, and p_z and using the b-quark invariant-mass for each jet we can compute all relevant kinematics for the regressed Higgs-boson candidate and compare against its reconstructed equivalent. The regressed and reconstructed invariant-mass obtained with this procedure is shown at Fig. 31. The generator-level mass cannot be displayed displayed in the same representation because is effectively a delta-function at 125 GeV (the total width of

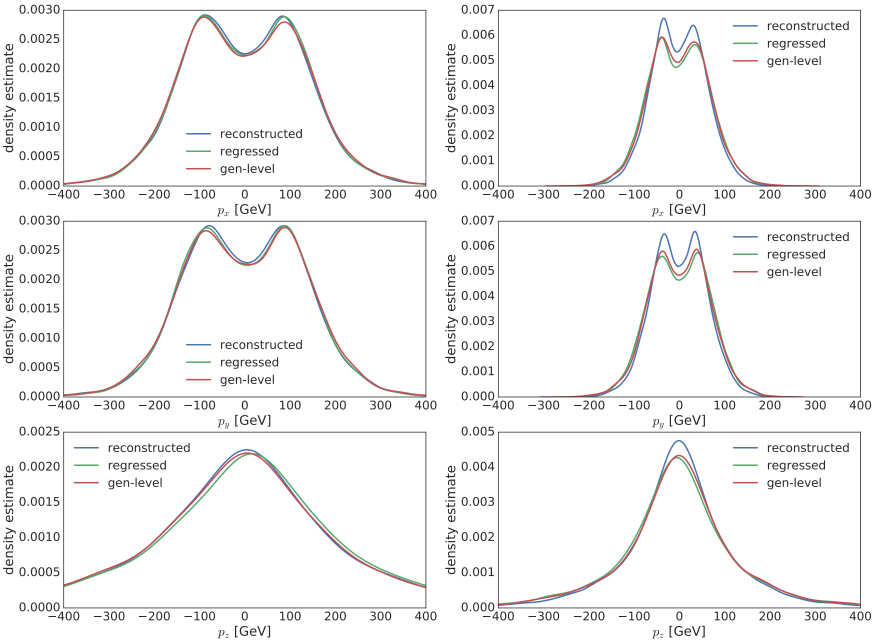


Figure 29: Reconstructed, regressed, and gen-level b-quark momenta from the decay of the leading Higgs-boson for $hh \to b\bar{b}b\bar{b}$ test events.

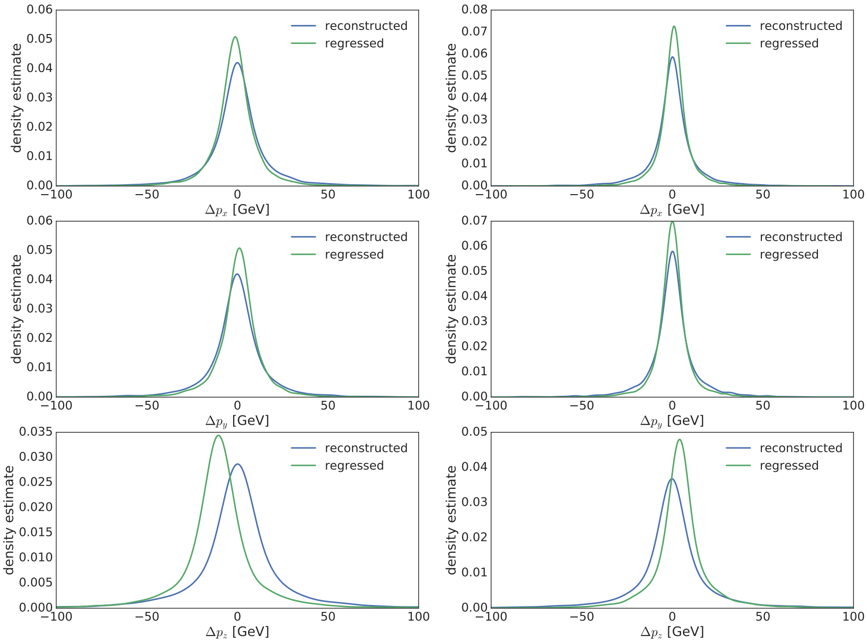


Figure 30: Differences between reconstructed and regressed momenta and the gen-level b-quark momenta from the decay of the leading Higgs-boson for $hh \to b\bar{b}b\bar{b}$ test events.

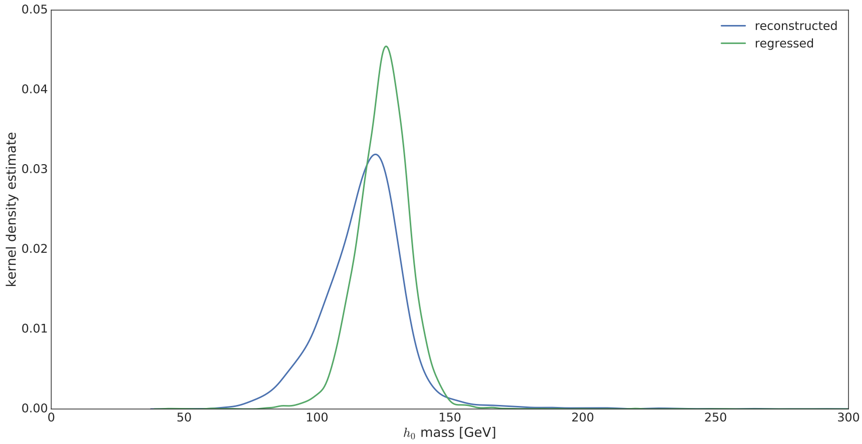


Figure 31: Reconstructed and regression-based invariant mass of the leading Higgs-boson for $hh \rightarrow b\bar{b}b\bar{b}$ events. The gen-level corresponding variable is not displayed but consists of a delta-like peak at 125 GeV.

Variable	$\bar{m}$ [GeV]	σ_m [GeV]
Reconstructed	118.62 ± 0.04	18.5 ± 0.6
Regressed	124.81 ± 0.02	10.8 ± 0.1

Table 15: Summary of the mean and standard deviation for the reconstructed and regressed leading Higgs boson candidate invariant mass.

the Higgs boson is 4 MeV in the standard model). While the reconstructed leading Higgs-boson invariant mass is smeared out and heavy tailed at lower masses, the regressed-mass distribution is more peaked and symmetrical. This can also be seen by computing the mean and standard deviation for each variable, which are summarized in Tab. 15. The regressed Higgs-boson mass is much closer to the expected generator-level value and its standard deviation is greatly reduced. While naively one could say that will likely translate to better differentiation from background it is still to be seen the effect of the regressor on background events, which will studied in Sec. 7.6.

7.4 2$^{\text{nd}}$ step: trailing $h \rightarrow b\bar{b}$ kinematic regression

Similarly to the previous step, the generator-level 3-momenta in Cartesian coordinates of each the b-quarks which are the decay products of the trailing Higgs-boson are used as target features. A total of 55 input features are used, composed by the same 44 variables described in the first step, plus the following 11 reconstructed event variables:

- The 3-momenta in Cartesian coordinates of the regressed b-quarks from the decay of the leading Higgs-candidate from the previous regression stage (6 features).
- The 3-momenta in Cartesian coordinates, energy, and mass of the leading Higgs-candidate created using the regressed b-quarks from the decay of the leading Higgs-candidate from the previous regression stage (5 features).

After training the regression model described in 7.2 for 50 epochs in mini-batches of 256 samples, the distributions of the regressed targets are shown together with the generator-level truth and

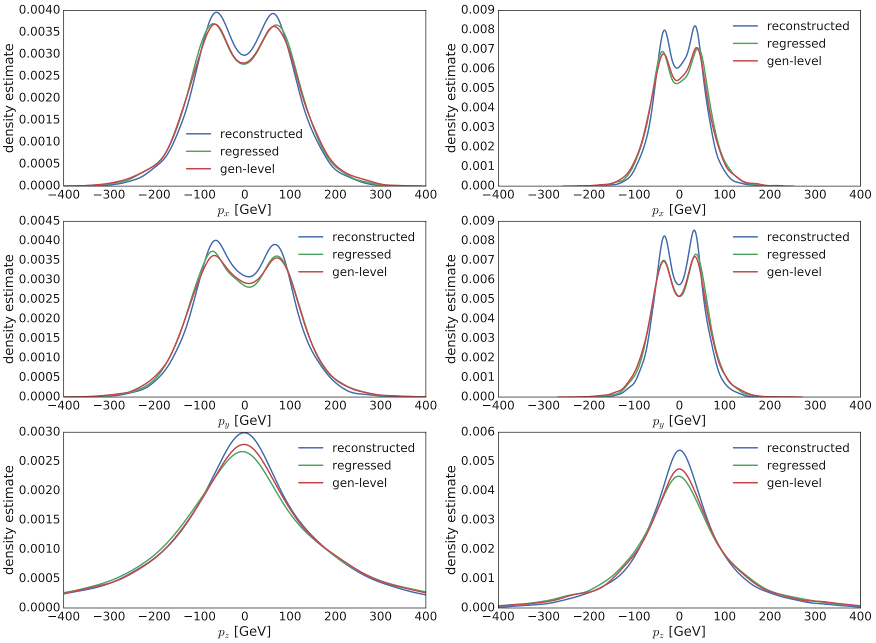


Figure 32: Reconstructed, regressed and gen-level b-quark momenta from the decay of the trailing Higgs candidate for $hh \to b\,\bar{b}\,b\,\bar{b}$ test events.

Variable	$\bar{m}$ [GeV]	σ_m [GeV]
Reconstructed	114.17 ± 0.04	20.6 ± 0.9
Regressed	122.57 ± 0.02	9.35 ± 0.04

Table 16: Summary of the mean and standard deviation for the reconstructed and regressed trailing Higgs-boson invariant-mass.

the corresponding reconstructed features in Fig. 32 for the test set. The target feature behaviour is better captured by the corresponding reconstructed features. This can be confirmed by studying the distribution of the per jet differences with respect to the generator-level provided in Fig. 33. The coordinates p_x and p_z of both b-quarks are very well modelled, outperforming the corresponding reconstructed variables. Small biases are observed for the p_z, especially for the first b-quark, which were also seen in Sec. 7.3, where their origin and importance are discussed.

As in the previous step, using the regressed p_x, p_y and p_z along with the b-quark invariant mass for each jet we can compute all relevant kinematics for the regressed Higgs-boson candidate and compare against its reconstructed equivalent. The regressed and and reconstructed invariant-masses obtained with this procedure is shown at Fig. 34. Again, we observe that the regressed distribution is more peaked and symmetrical than the reconstructed distribution of the trailing Higgs invariant mass. A summary of the results is provided in Tab. 16, evidencing that again the regressed Higgs-boson mass is closer to the generator-level value and its standard deviation is greatly reduced. The same caveat discussed *supra* applies here: the improvement does not directly traslate in a better separation of signal and background, which is discussed in Sec. 7.6.

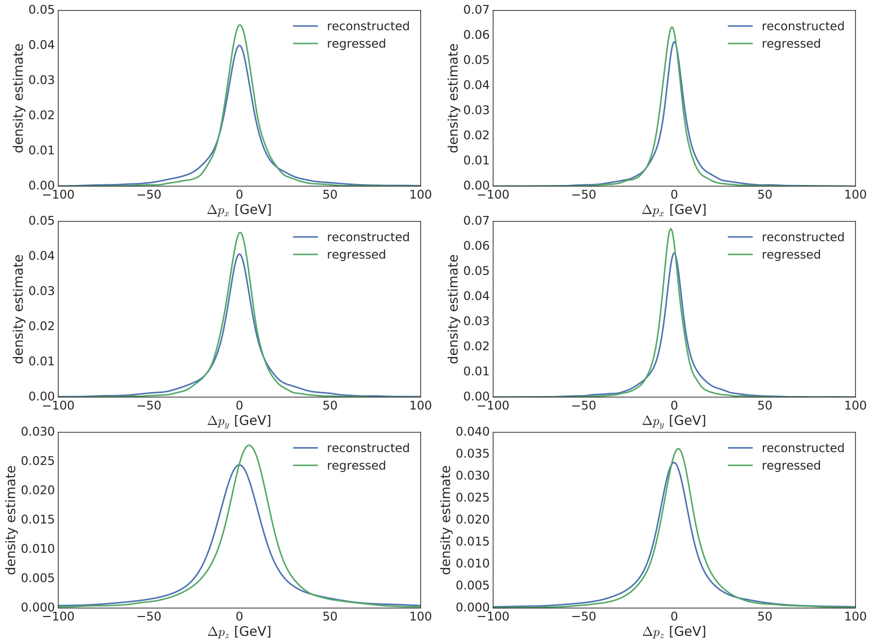


Figure 33: Differences between reconstructed and regressed momenta and the gen-level b-quark momenta from the decay of the trailing Higgs candidate for $hh \to b\bar{b}b\bar{b}$ test events.

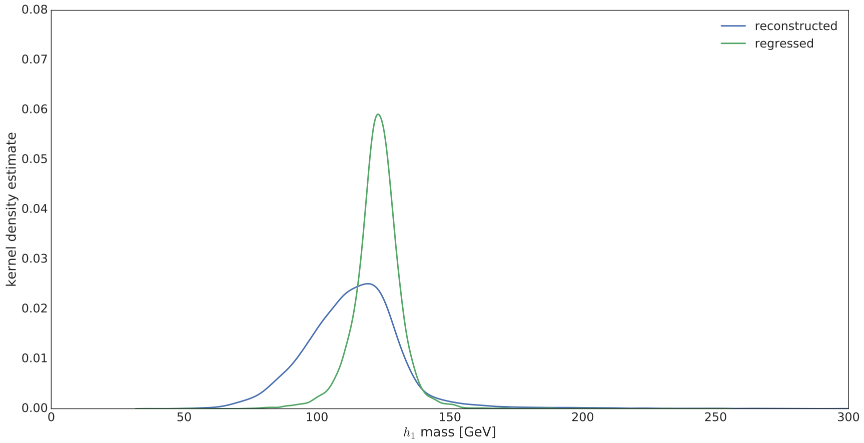


Figure 34: Reconstructed and regression-based invariant mass of the trailing Higgs-boson for $hh \to b\bar{b}b\bar{b}$ events. The corresponding gen-level variable is not displayed but consists of a delta-like peak at 125 GeV.

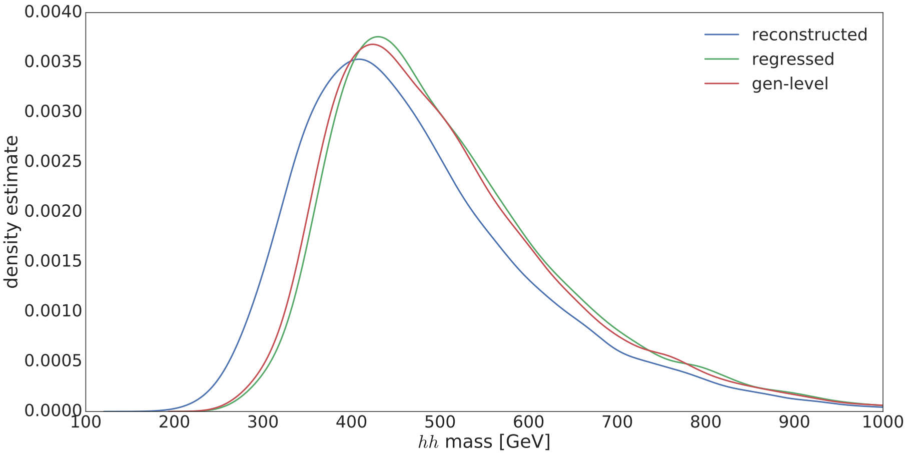


Figure 35: Reconstructed, regressed, and gen-level $h\,h$ system invariant-mass.

7.5 3$^{\text{rd}}$ step: $h\,h$ mass regression

For this regression, the problem is simplified to univariate regression because only the generator-level $h\,h$ invariant mass is the target. A total of 71 input features were used, composed of the following reconstructed event variables:

- The 3-momenta in Cartesian coordinates, energy, mass, and momenta modulus of every jet selected after pairing (4 jets, 24 features).
- The 3-momenta in Cartesian coordinates, energy, mass, and momenta modulus of each reconstructed Higgs-candidate (2 Higgs candidates, 12 features).
- The 3-momenta in Cartesian coordinates, energy, mass, and momenta modulus of the reconstructed $h\,h$-candidate (6 features).
- The transverse momenta in Cartesian coordinates of the missing transverse energy of the event (2 features).
- The 3-momenta in Cartesian coordinates of the regressed b-quarks from the decay of the Higgs candidate from the previous two regression stages (12 features).
- The 3-momenta in Cartesian coordinates, energy, and mass of the Higgs candidate created using the regressed b-quarks variables from the two previous regression stages (10 features).
- The 3-momenta in Cartesian coordinates, energy, and mass of the $h\,h$ candidate created using the regressed b-quarks variables from the two previous regression stages (5 features).

After training the regression model described in 7.2 for 50 epochs in mini-batches of 256 samples, the distribution of the regressed target is shown together with the generator-level truth and the corresponding reconstructed feature in Fig. 35 for the test set. The target-feature behaviour is much better captured by the regressed feature than the corresponding reconstructed feature. This can be confirmed by studying the $h\,h$ invariant-mass differences with respect to the generator-level in Fig. 36.

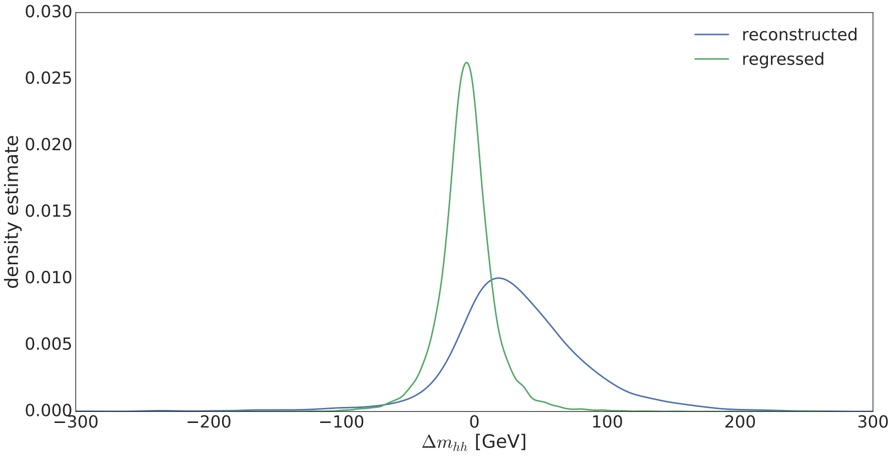


Figure 36: Differences of the reconstructed and regressed $h\,h$ invariant-masses with respect to the generator-level value.

7.6 Background regression response

For training and testing the regressor in the previous studies we have only dealt with generator-matched events from the $hh \rightarrow b\,\bar{b}\,b\,\bar{b}$ process. However, if this regressors where going to be applied to experimental data it is important to check the regressor response over background events. In this section, the regressed distributions for the main analysis-background, $b\,\bar{b}\,b\,\bar{b}$ via QCD, will be studied and compared with the reconstructed variables.

The reconstructed and regressed Higgs invariant masses for the $hh \rightarrow b\,\bar{b}\,b\,\bar{b}$ and $b\,\bar{b}\,b\,\bar{b}$ QCD background are shown in Fig. 37 and Fig. 38, for the leading and trailing candidate respectively. In both cases, the reconstructed background density is soft and with very long tails. Instead, the distribution of the regressed invariant masses are strongly altered, behaving similarly to the $hh \rightarrow b\,\bar{b}\,b\,\bar{b}$ events by peaking around 125 GeV. This would affect the performance if we were to use directly the regressed variables instead the reconstructed for doing statistical inference based solely on the shape of their distributions. Nevertheless, these variables could potentially be useful as features of a machine learning classifier which discriminates signal from different backgrounds, as demonstrated for $\tau\,\bar{\tau}\,b\,\bar{b}$ in Sec. 5.

The extreme morphing of the background invariant-masses could be explained by the fact that the model has only be trained with a sample for which the generator-level Higgs-boson invariant-masses were constant at 125 GeV. Even though the p_x, p_y, and p_z of the b-quarks were the regression targets, rather than the Higgs-boson invariant-mass, it is very likely that an approximation of the implicit relation:

$$(E^0 + E^1)^2 - (p_x^0 + p_x^1)^2 - (p_y^0 + p_y^1)^2 - (p_z^0 + p_z^1)^2 = m_h^2, \tag{15}$$

where the 0 and 1 super-indices correspond to each b-quark coming from the decay of the Higgs boson, was learned during the training process. If an approximate internal representation as such has been learned, that would help improving the performance but could result in the observed effect when computing the Higgs-candidate invariant-mass. In a sense, the model has learned a physics-based relation between the variables present in the training data which is assumed and applied during evaluation even if the data is different from the one used for training.

The regressed and reconstructed $h\,h$ invariant-mass using the model trained in Sec. 7.5 was evaluated for background events and the resulting distribution are shown in Fig. 39. The

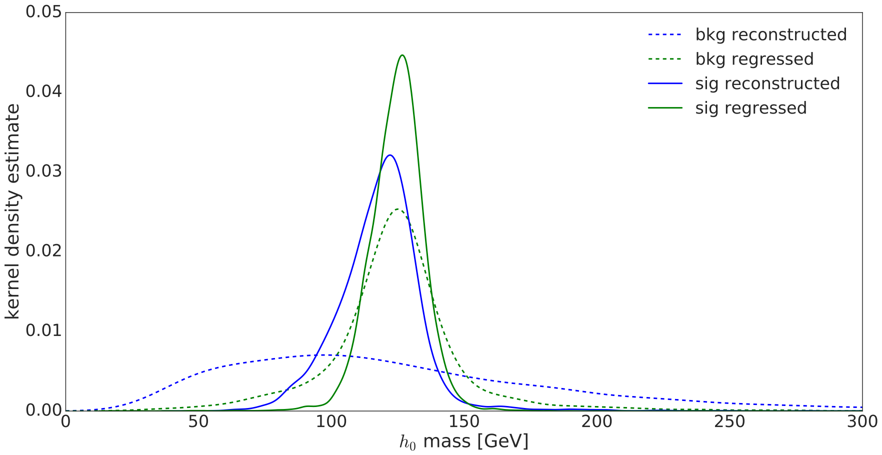


Figure 37: Reconstructed and regression-based invariant mass of the leading Higgs-candidate for $hh \to b\,\bar{b}\,b\,\bar{b}$ and background $b\,\bar{b}\,b\,\bar{b}$ events.

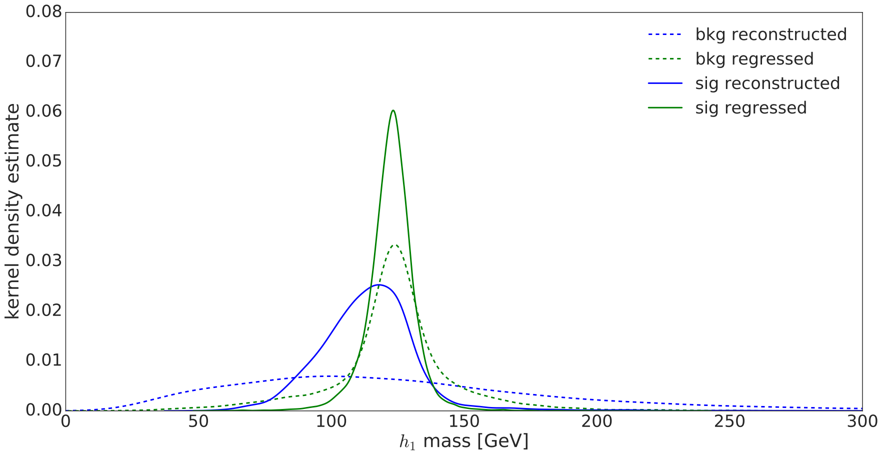


Figure 38: Reconstructed and regression-based invariant mass of the trailing Higgs-candidate for $hh \to b\,\bar{b}\,b\,\bar{b}$ and background $b\,\bar{b}\,b\,\bar{b}$ events.

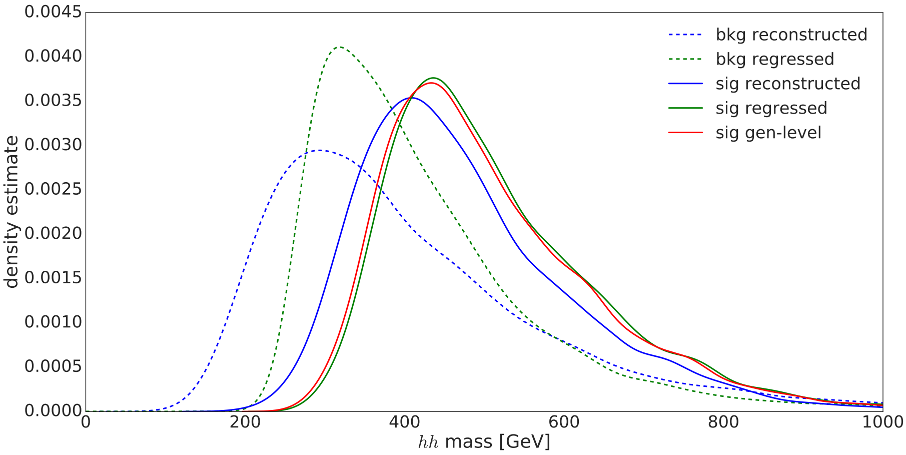


Figure 39: Reconstructed and regression-based invariant mass of $h\,h$ system for $hh \rightarrow b\,\bar{b}\,b\,\bar{b}$ and background $b\,\bar{b}\,b\,\bar{b}$ events. The generator-level distribution for signal is also shown.

regressed-background distribution also differs importantly from the reconstructed, the low-mass tail being greatly reduced and the main peak increased and shifted towards higher energies. Separation of signal from background is not clearly improved, but nevertheless the regressed variables might also be useful for future classification-studies.

7.7 Summary of $hh \rightarrow b\,\bar{b}\,b\,\bar{b}$ regression studies

In this study the generator-level variables for the b-quarks and $h\,h$ candidates have been regressed using deep neural-networks trained on a large, fast-simulation data-sample of events for the $hh \rightarrow b\,\bar{b}\,b\,\bar{b}$ decay channel. While the regression was able to model very accurately $hh \rightarrow b\,\bar{b}\,b\,\bar{b}$ events, the regressed features will likely not help to distinguish signal and background by themselves but could be of used as inputs for another machine-learning classifier. One interesting challenge for the future would be to process the event reconstruction problem in a process-independent way from a supervised machine-learning perspective. One of the still unsolved difficulties is how to successfully deal with type heterogeneity for both input (tracks, calorimeter towers and other variable sized low-level detector information) and targets (type and measurable properties of all final state particles, which is also of a variable size set).

8 Conclusions

In this document we present the results of studies performed with advanced multi-variate algorithms applied to the problem of improving our sensitivity to the rare process of Higgs-boson pair-production at the LHC. The studies concentrated on two decay channels of interest to the AMVA4NewPhysics institutions: the one involving two tau-leptons and two bottom-quark jets, and the one involving four bottom-quark jets.

In both cases we studied the regression of quantities derived from detector measurements to target features of relevance to the analyses: the kinematic variables of the Higgs-boson decay-products and the invariant-mass of the $h\,h$ system. Significant improvements were found in both considered decay channels: mass-resolution increases of 50 % and 60 % for the Higgs boson in the $b\,\bar{b}$ and $\tau\,\bar{\tau}$ decay channels, respectively; and di-Higgs-mass resolution increases of 60 % in both the $b\,\bar{b}\,b\,\bar{b}$ and $\tau\,\bar{\tau}\,b\,\bar{b}$ decay channels.

We also studied the classification problem of distinguishing di-Higgs events from the main background, obtaining very promising performances with the complex deep neural-networks employed in the $\tau\,\bar{\tau}\,b\,\bar{b}$ study, and with an advanced BDT implementation in the $b\,\bar{b}\,b\,\bar{b}$ study, which was found to outperform an earlier attempt with a deep neural-network.

In the case of the $\tau\,\bar{\tau}\,b\,\bar{b}$ decay channel we also examined the benefits of employing the regressed variables in the classification problem. Doing so, we were able to achieve a 10% percent improvement in classifier performance.

Due to the fact that the generation of the datasets made use of a quick, but potentially inaccurate, detector simulation (Delphes), and that the detector geometry corresponded to something in between the ATLAS and CMS detectors, the studies presented in this document cannot be used in their current forms for experimental searches at ATLAS or CMS. However, they provide a strong foundation for the application of advanced machine-learning technology to fully simulated datasets, and eventually to real collider data. We intend to now proceed in that direction but will be unable to document in public reports the results of those further investigations, due to the restrictive rules of the experimental collaborations.

References

[1] J. Alwall *et al.*, *The automated computation of tree-level and next-to-leading order differential cross sections, and their matching to parton shower simulations*, JHEP **07** (2014) 079, `arXiv:1405.0301 [hep-ph]`.

[2] R. D. Ball *et al.*, *Parton distributions with LHC data*, Nucl. Phys. **B867** (2013)244–289, `arXiv:1207.1303 [hep-ph]`.

[3] T. Sjöstrand *et al.*, *An Introduction to PYTHIA 8.2*, Comput. Phys. Commun. **191** (2015)159–177, `arXiv:1410.3012 [hep-ph]`.

[4] **LHC Higgs Cross Section Working Group** Collaboration, D. de Florian *et al.*, *Handbook of LHC Higgs Cross Sections: 4. Deciphering the Nature of the Higgs Sector*, `arXiv:1610.07922 [hep-ph]`.

[5] **CMS Collaboration** Collaboration, *Comparisons of Theory Predictions for the ttbar Process with Data from pp Collisions at sqrt(s)= 8 TeV*, Tech. Rep. CMS-PAS-TOP-15-011, CERN, Geneva, 2015. `https://cds.cern.ch/record/2110635`.

[6] P. Nason, *A New method for combining NLO QCD with shower Monte Carlo algorithms*, JHEP **11** (2004) 040, `arXiv:hep-ph/0409146 [hep-ph]`.

[7] S. Frixione, P. Nason, and C. Oleari, *Matching NLO QCD computations with Parton Shower simulations: the POWHEG method*, JHEP **11** (2007) 070, `arXiv:0709.2092 [hep-ph]`.

[8] S. Alioli, P. Nason, C. Oleari, and E. Re, *A general framework for implementing NLO calculations in shower Monte Carlo programs: the POWHEG BOX*, JHEP **06** (2010) 043, `arXiv:1002.2581 [hep-ph]`.

[9] K. Hamilton, P. Richardson, and J. Tully, *A Positive-Weight Next-to-Leading Order Monte Carlo Simulation for Higgs Boson Production*, JHEP **04** (2009) 116, `arXiv:0903.4345 [hep-ph]`.

[10] M. Guzzi *et al.*, *CT10 parton distributions and other developments in the global QCD analysis*, `arXiv:1101.0561 [hep-ph]`.

[11] M. R. Whalley, D. Bourilkov, and R. C. Group, *The Les Houches accord PDFs (LHAPDF) and LHAGLUE*, in *HERA and the LHC: A Workshop on the implications of HERA for LHC*

physics. Proceedings, Part B, pp. 575–581. 2005. `arXiv:hep-ph/0508110 [hep-ph]`. `http://lhapdf.hepforge.org`.

[12] **Particle Data Group** Collaboration, C. Patrignani *et al.*, *Review of Particle Physics*, Chin. Phys. **C40** no. 10, (2016) 100001.

[13] **DELPHES 3** Collaboration, J. de Favereau *et al.*, *DELPHES 3, A modular framework for fast simulation of a generic collider experiment*, JHEP **02** (2014) 057, `arXiv:1307.6346 [hep-ex]`.

[14] M. Selvaggi, *DELPHES 3: A modular framework for fast-simulation of generic collider experiments*, J. Phys. Conf. Ser. **523** (2014) 012033.

[15] A. Mertens, *New features in Delphes 3*, J. Phys. Conf. Ser. **608** no. 1, (2015) 012045.

[16] **ATLAS Collaboration** Collaboration, G. Aad *et al.*, *The ATLAS Experiment at the CERN Large Hadron Collider*, J. Instrum. **3** (2008) S08003. 437. Also published by CERN Geneva in 2010.

[17] **CMS** Collaboration, S. Chatrchyan *et al.*, *The CMS experiment at the CERN LHC*, JINST **3** (2008) S08004.

[18] J. Allison *et al.*, *Geant4 developments and applications*, IEEE Transactions on Nuclear Science **53** no. 1, (Feb, 2006)270–278.

[19] S. Agostinelli *et al.*, *Geant4—a simulation toolkit*, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment **506** no. 3, (2003)250 – 303.
`http://www.sciencedirect.com/science/article/pii/S0168900203013688`.

[20] R. Brun and F. Rademakers, *ROOT - An Object Oriented Data Analysis Framework*, Nucl. Inst. & Meth. in Phys. Res. A **389** (1997)81–86. Proceedings AIHENP'96 Workshop, Lausanne, Sep. 1996 See also http://root.cern.ch/.

[21] **CMS** Collaboration, C. Collaboration, *Search for non-resonant Higgs boson pair production in the* $\mathrm{b\overline{b}}\tau^{+}\tau^{-}$ *final state*,.

[22] M. Cacciari, G. P. Salam, and G. Soyez, *The Anti-k(t) jet clustering algorithm*, JHEP **04** (2008) 063, `arXiv:0802.1189 [hep-ph]`.

[23] M. Rosenblatt, *Remarks on Some Nonparametric Estimates of a Density Function*, Ann. Math. Statist. **27** no. 3, (09, 1956)832–837. `http://dx.doi.org/10.1214/aoms/1177728190`.

[24] E. Parzen, *On Estimation of a Probability Density Function and Mode*, Ann. Math. Statist. **33** no. 3, (09, 1962)1065–1076. `http://dx.doi.org/10.1214/aoms/1177704472`.

[25] D. W. Scott, *On Optimal and Data-Based Histograms*, Biometrika **66** no. 3, (1979)605–610. `http://www.jstor.org/stable/2335182`.

[26] B. W. Silverman, *Density Estimation for Statistics and Data Analysis.* Chapman & Hall, London, 1986.

[27] G. P. S. M. Cacciari and G. Soyez, *The anti-kt jet clustering algorithm*, JHEP (2008). `arXiv:0802.1189`.

[28] F. Chollet, *Keras*, GitHub (2016). `https://github.com/fchollet/keras`.

[29] K. He, X. Zhang, S. Ren, and J. Sun, *Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification*, CoRR (2015), `arXiv:1502.01852 [cs.CV]`.

[30] S. Ioffe and C. Szegedy, *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*, " Stanford University, Tech. Rep. (2015). `http://jmlr.org/proceedings/papers/v37/ioffe15.pdf`.

[31] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, *Dropout: A Simple Way to Prevent Neural Networks from Overfitting*, Journal of Machine Learning Research **15** (2014)1929–1958. `http://jmlr.org/papers/v15/srivastava14a.html`.

[32] T. Dozat, *Incorporating Nesterov momentum into Adam*, JMLR Workshop and Conference Proceedings **9** (2010). `http://cs229.stanford.edu/proj2015/054_report.pdf`.

[33] **CMS** Collaboration, V. Khachatryan *et al.*, *Search for a pseudoscalar boson decaying into a Z boson and the 125 GeV Higgs boson in $\ell^+\ell^- b\bar{b}$ final states*, Phys. Lett. **B748** (2015)221–243, `arXiv:1504.04710 [hep-ex]`.

[34] J. Gallicchio *et al.*, *Multivariate discrimination and the Higgs + W/Z search*, JHEP **04** (2011) 069, `arXiv:1010.3698 [hep-ph]`.

[35] C. T. Chen, *XGBoost*,. `arXiv:1603.02754`.

[36] J. K. B. et al., *Boosting Higgs pair production in the $b\bar{b}b\bar{b}$ final state with multivariate techniques*,. `arXiv:1512.08928`.

[37] B. C. A. et al., *Genetic algorithms and experimental discrimination of susy models*, JHEP (2004). `arXiv:hep-ph/0406277`.

[38] P. Baldi, P. Sadowski, and D. Whiteson, *Searching for Exotic Particles in High-Energy Physics with Deep Learning*, Nature Commun. **5** (2014) 4308, `arXiv:1402.4735 [hep-ph]`.

[39] D. Guest, J. Collado, P. Baldi, S.-C. Hsu, G. Urban, and D. Whiteson, *Jet Flavor Classification in High-Energy Physics with Deep Neural Networks*, Phys. Rev. **D94** no. 11, (2016) 112002, `arXiv:1607.08633 [hep-ex]`.

[40] G. Louppe, K. Cho, C. Becot, and K. Cranmer, *QCD-Aware Recursive Neural Networks for Jet Physics*, `arXiv:1702.00748 [hep-ph]`.

[41] **ATLAS Collaboration** Collaboration, *Search for pair production of Higgs bosons in the $b\bar{b}b\bar{b}$ final state using proton–proton collisions at $\sqrt{s} = 13$ TeV with the ATLAS detector*, Tech. Rep. ATLAS-CONF-2016-049, CERN, Geneva, Aug, 2016. `http://cds.cern.ch/record/2206131`.

[42] **CMS Collaboration** Collaboration, *Search for non-resonant pair production of Higgs bosons in the* $\mathrm{b\bar{b}b\bar{b}}$ *final state with 13 TeV CMS data*, Tech. Rep. CMS-PAS-HIG-16-026, CERN, Geneva, 2016. `https://cds.cern.ch/record/2209572`.

[43] M. Cacciari, G. P. Salam, and G. Soyez, *FastJet User Manual*, Eur. Phys. J. **C72** (2012) 1896, `arXiv:1111.6097 [hep-ph]`.

[44] M. Brucher *et al.*, *Scikit-learn: Machine Learning in Python*, Journal of Machine Learning Research **12** (2011)2825–2830.

[45] J. D. Hunter, *Matplotlib: A 2D graphics environment*, Computing In Science & Engineering **9** no. 3, (2007)90–95.

[46] T. Augspurger *et al.*, *seaborn: v0.7.1 (June 2016)*, June, 2016. `https://doi.org/10.5281/zenodo.54844`.

[47] Theano Development Team, *Theano: A Python framework for fast computation of mathematical expressions*, arXiv e-prints **abs/1605.02688** (May, 2016). `http://arxiv.org/abs/1605.02688`.

[48] S. van der Walt, S. C. Colbert, and G. Varoquaux, *The NumPy Array: A Structure for Efficient Numerical Computation*, Computing in Science Engineering **13** no. 2, (March, 2011)22–30.

[49] W. McKinney, *Data Structures for Statistical Computing in Python*, in *Proceedings of the 9th Python in Science Conference*, S. van der Walt and J. Millman, eds., pp. 51 – 56. 2010.

[50] S. Seabold and J. Perktold, *Statsmodels: econometric and statistical modeling with Python*, in *Proceedings of the 9th Python in Science Conference*, pp. 57–61. 2010. `http://statsmodels.sourceforge.net/stable/`.

A Software details

Software	Version	References	Use/Notes
Keras	1.05	[28]	Implementing neural networks
Root	6.06/04	[20]	Analysis of Monte Carlo data
MadGraph	5.2.4.2	[1]	MC generation
Pythia	8.219	[3]	MC generation
Powheg Box	2	[6, 7, 8, 9]	MC generation
Lhapdf	5.9.1	[11]	PDFs for MC generation
Delphes	3.3.2	[13, 14, 15]	Detector simulation
FastJet	3.1.3	[43]	Jet clustering
Scikit-Learn	various	[44]	Cross-validation
Matplotlib	various	[45]	Plot production
Seaborn	various	[46]	Plot production
Theano	various	[47]	Keras back-end
NumPy	various	[48]	Data analysis and computation
Pandas	various	[49]	Data analysis and computation
Statsmodels	0.6.1	[50]	Kernel density-estimation