

Search for exotic Higgs boson decays with CMS and fast machine learning solutions for the LHC

by

Ho Fung Tsoi

A dissertation submitted in partial fulfillment of
the requirements for the degree of

Doctor of Philosophy

(Physics)

at the

UNIVERSITY OF WISCONSIN–MADISON

2024

Date of final oral examination: July 30, 2024

The dissertation is approved by the following members of the Final Oral Committee:

Sridhara Dasu, Professor, Physics

Tulika Bose, Professor, Physics

Ellen Zweibel, Professor, Astronomy and Physics

Yu Hen Hu, Professor, Electrical and Computer Engineering



Search for exotic Higgs boson decays with CMS and fast machine learning solutions for the LHC

Ho Fung Tsoi

Abstract

There exists potential for discoveries beyond the Standard Model in scalar sector, which could manifest as Higgs boson exotic decays into new light pseudoscalars. This search targets such decays, focusing on pseudoscalar masses ranging from 12 to 60 GeV, in final states where one pseudoscalar decays into two b quarks and the other into two τ leptons or two muons. The analysis is based on a dataset of proton-proton collisions at $\sqrt{s} = 13$ TeV, collected by the CMS detector during LHC Run 2, with an integrated luminosity of 138 fb^{-1} . Dedicated neural networks are used to distinguish between signal and background, significantly enhancing sensitivity. The results are presented as exclusion limits at 95% confidence level on the model-independent branching ratio and are interpreted within two-Higgs doublet models augmented by a singlet.

The second part of this thesis presents machine learning methods to enhance overall sensitivity in the low-latency domain for the LHC experiments. A novel machine learning-based trigger algorithm is developed, using anomaly detection to search for new physics in a model-agnostic manner as close to the raw collision data as possible. This anomaly detection trigger is sensitive to a wide range of both conventional and unconventional physics signatures and has an inference latency of $\mathcal{O}(100)$ ns on an FPGA. It is deployed during Run 3 in the CMS Level-1 trigger system, which processes the first round of real-time event selection from collision data at a rate of 40 MHz. Additionally, a novel model compression method using symbolic regression is developed to accelerate machine learning inference to nanosecond speeds on FPGAs. This method demonstrates potential to significantly reduce the computational costs of machine learning algorithms while maintaining performance comparable to that of neural networks. These advancements are crucial for meeting the sensitivity and computational demands of resource-constrained environments such as the LHC experiments.

To my family.

Acknowledgements

First, I would like to express my deep gratitude to my advisor, Sridhara Dasu, for offering me the opportunity to work in the Wisconsin CMS group and for guiding me in making this thesis possible. He is professional, experienced, resilient, open-minded, easy-going, and caring. I was very fortunate to have such a great advisor along the journey.

I was very lucky to have the invaluable experience of residing at CERN for two years, during which I met many great people.

I thank Cecile Caillol for guiding me through the physics analysis and for teaching me a lot about physics and CMS. She was a great mentor to me. I also thank Pieter Everaerts for guiding me through the analysis and for being helpful whenever I had doubts. Their presence gave me a sense of confidence and reassurance.

I thank Vladimir Loncar for his guidance in the field of machine learning. I enjoyed sharing ideas with him, and it always led to productive discussions. His informative advice has also helped me avoid numerous traps. I truly appreciate all his support.

I thank Adrian Alan Pol for his productive feedback during our collaboration on the machine learning projects. I am very grateful to Isobel Ojalvo for her trust in me to take on the development of anomaly detection trigger. I also thank Alexander Savin and Varun Sharma for their supervision of the CaloLayer-1 trigger work.

I appreciate everyone I have met around Madison and CERN. Special thanks go to my officemates, Piyush Kumar, Abhishikth Mallampalli, Ganesh Parida, and Victor Shang, with whom I spent many usual hours, as well as to everyone in the Wisconsin CMS group.

Finally, I thank my family for raising me and for their support throughout my life.

Contents

0	Introduction	1
I	Search for exotic Higgs boson decays with CMS	4
1	Theoretical motivations	5
1.1	The Standard Model of particle physics	5
1.1.1	Gauge symmetry and Lagrangian	6
1.1.2	Electroweak symmetry breaking and Higgs mechanism	10
1.1.3	Higgs boson	15
1.2	New physics beyond the Standard Model	21
1.2.1	The Standard Model is incomplete	21
1.2.2	Motivations for exotic Higgs boson decays	22
1.2.3	Two-Higgs doublet models augmented by a scalar singlet	23
2	Experiment overview	28
2.1	The Large Hadron Collider	28
2.2	The Compact Muon Solenoid detector	31
2.2.1	Tracker	34
2.2.2	Electromagnetic calorimeter	35
2.2.3	Hadron calorimeter	36
2.2.4	Muon system	38
2.2.5	Trigger system	38
2.3	Reconstruction and identification of physics objects	41
2.3.1	Electrons	43
2.3.2	Muons	44
2.3.3	Jets and b-tagging	46
2.3.4	Hadronic taus	48
3	Search for Higgs boson decaying to a pair of pseudoscalars $h \rightarrow aa$	51
3.1	Search for $h \rightarrow aa \rightarrow \tau\tau b\bar{b}$	51
3.1.1	Analysis strategy	52
3.1.2	Simulation	55
3.1.3	Trigger selection	57
3.1.4	Object selection	57
3.1.5	Background estimation	59

3.1.6	Corrections to simulation	63
3.1.7	Optimization using neural networks	64
3.1.8	Systematic uncertainties	84
3.1.9	Signal extraction	88
3.1.10	Exclusion limits and model interpretation	96
3.2	Combination of the $\tau\tau bb$ and $\mu\mu bb$ final states	102
3.2.1	Analysis overview of $h \rightarrow aa \rightarrow \mu\mu bb$	102
3.2.2	Combination strategy	106
3.2.3	Exclusion limits and model interpretation	107
3.3	Future prospects	112
4	Summary of Part I	115
II	Fast machine learning solutions for the LHC	117
5	Motivations and fundamentals	118
5.1	Machine learning	119
5.1.1	Fully-connected neural network	121
5.1.2	Convolutional neural network	124
5.2	Field-programmable gate array (FPGA)	126
5.2.1	Basics	126
5.2.2	Machine learning inference on FPGAs with <code>hls4ml</code>	128
6	Anomaly detection in the Level-1 trigger for model-agnostic new physics searches	131
6.1	Motivations	132
6.2	Anomaly detection	135
6.3	Datasets	136
6.4	Model architecture and training	139
6.4.1	Autoencoder	139
6.4.2	Model compression	143
6.5	Performance	145
6.6	Summary and future prospects	154
7	Symbolic regression for hardware-efficient inference	155
7.1	Symbolic regression	156
7.1.1	Methods	157
7.2	Efficient implementation on FPGAs	160
7.3	Dataset	162
7.4	Experiments	162
7.5	Results	164
7.6	Summary and future prospects	172
8	Summary of Part II	173
9	Conclusion and future prospects	175

A	Published results and personal contributions	176
----------	---	------------

List of Figures

1.1	The shape of the Higgs potential where the minimum is at a non-zero field value.	12
1.2	The SM prediction for the cross sections of the 125-GeV Higgs boson in different production modes, plotted as functions of the LHC energy. The figure is taken from Ref. [1]	17
1.3	The three dominant Higgs production channels at the LHC.	18
1.4	The SM prediction for the branching ratios of the different Higgs decay channels, assuming a Higgs boson mass of 125 GeV. The values are quoted from Ref. [1]	19
1.5	CMS measurements of the coupling modifiers of the Higgs boson to fermions and heavy gauge bosons, plotted as functions of particle mass, which are consistent with the SM prediction. The figure is taken from Ref. [2].	20
2.1	The CERN accelerator complex. The injection sequence goes as: Linac 4 \rightarrow PS Booster \rightarrow PS \rightarrow SPS \rightarrow LHC. The figure is taken from Ref. [3]. . .	29
2.2	Summary of CMS measurements of the SM cross sections. The figure is taken from Ref. [4].	31
2.3	Data taking schedule of the LHC and the HL-LHC. The figure is taken from Ref. [5].	32
2.4	Schematic view of the CMS detector. The figure is taken from Ref. [6]. . . .	33
2.5	Schematic view of the Phase-1 CMS tracker, showing one quadrant in the $r - z$ plane. The pixel detector is shown in green, while the strip detector is shown in red and blue. The figure is taken from Ref. [7].	34
2.6	Schematic view of the CMS ECAL, showing one quadrant in the $r - z$ plane. The figure is taken from Ref. [8].	36
2.7	Schematic view of the CMS HCAL during Run 2 operation, showing one quadrant in the $r - z$ plane. The figure is taken from Ref. [9].	37
2.8	Schematic view of the CMS detector during Run 2 operation, showing one quadrant in the $r - z$ plane, with the different muon stations indicated. The figure is taken from Ref. [10].	39
2.9	The CMS L1 trigger during Run 2 operation, with each trigger subsystem indicated. The figure is taken from Ref. [11].	40
2.10	Schematic view in a transverse slice of CMS, showing the signature of each of the five PF candidates. The figure is taken from Ref. [12].	42

2.11	Characteristics of a jet originating from a b or c quark, which consists of tracks originated from a secondary vertex (SV) displaced from the primary vertex (PV) with a large impact parameter (IP). The figure is taken from Ref. [13].	48
3.1	Schematic view of an event of the production of a Higgs boson in association with a pair of top quarks (tth), simulated by a MC generator. The two incident protons are represented by the big ovals in green. The big circle at the center represents the hard scattering process that produces a top-quark pair and a Higgs boson, with their subsequent decays represented by the small circles in red. The oval in purple represents the secondary interaction called the underlying event. Parton showering is followed by hadronization, represented by the small ovals in green. Finally, the hadron decays are represented by the circles in dark green. The figure is taken from Ref. [14].	55
3.2	The 15 input features for the ‘1 b tag’ category in the $\mu\tau_h$ final state. The signal (background) class is represented by the distributions in blue (orange).	70
3.3	The first 15 input features for the ‘> 1 b tag’ category in the $\mu\tau_h$ final state. The signal (background) class is represented by the distributions in blue (orange).	71
3.4	The last 15 input features for the ‘> 1 b tag’ category in the $\mu\tau_h$ final state. The signal (background) class is represented by the distributions in blue (orange).	72
3.5	Correlation matrices for the 15 input features for the ‘1 b tag’ category in the $\mu\tau_h$ final state, evaluated with the signal and background events, respectively.	73
3.6	Correlation matrices for the 30 input features for the ‘> 1 b tag’ category in the $\mu\tau_h$ final state, evaluated with the signal and background events, respectively.	74
3.7	In each category, the loss as a function of the training epoch is plotted on the left, while the accuracy as a function of the training epoch is plotted on the right. The blue (orange) curves represent evaluation on the training (validation) set.	78
3.8	DNN score distributions in the six categories. The box histograms in red (green) represent the signal (background) class evaluated on the training and validation sets combined. The filled histograms in orange (blue) represent the signal (background) class evaluated on the test set.	79
3.9	ROC curves in the six categories, evaluated on the test set. The area under the curve (AUC) is indicated. The dotted lines represent a classifier that does not learn and makes predictions by random chance.	80
3.10	2D correlation plots of $m_{\tau\tau}$ vs. DNN score, for the $\mu\tau_h$ ‘1 b tag’ category. Signal, background, and observed data are plotted separately.	81
3.11	Pre-fit distributions of the DNN score with the category indicated, comparing the weighted background events and the observed data. The total background uncertainty is shown in grey. The lower panel shows the ratio of the observed data to the expected background yields. The figures are taken from Ref. [15].	83

3.12	Post-fit distributions of $m_{\tau\tau}$ for the $\mu\tau_h$ ‘1 b tag’ category, with the subregion indicated. The signal (35 GeV) is shown as a box histogram in red, assuming the branching ratio $B(h \rightarrow aa \rightarrow \tau\tau bb) = 10\%$. The figures are taken from Ref. [15].	90
3.13	Post-fit distributions of $m_{\tau\tau}$ for the $\mu\tau_h$ ‘> 1 b tag’ category, with the subregion indicated. The signal (35 GeV) is shown as a box histogram in red, assuming the branching ratio $B(h \rightarrow aa \rightarrow \tau\tau bb) = 10\%$. The figures are taken from Ref. [15].	91
3.14	Post-fit distributions of $m_{\tau\tau}$ for the $e\tau_h$ ‘1 b tag’ category, with the subregion indicated. The signal (35 GeV) is shown as a box histogram in red, assuming the branching ratio $B(h \rightarrow aa \rightarrow \tau\tau bb) = 10\%$. The figures are taken from Ref. [15].	92
3.15	Post-fit distributions of $m_{\tau\tau}$ for the $e\tau_h$ ‘> 1 b tag’ category, with the subregion indicated. The signal (35 GeV) is shown as a box histogram in red, assuming the branching ratio $B(h \rightarrow aa \rightarrow \tau\tau bb) = 10\%$. The figures are taken from Ref. [15].	93
3.16	Post-fit distributions of $m_{\tau\tau}$ for the $e\mu$ ‘1 b tag’ category, with the subregion indicated. The signal (35 GeV) is shown as a box histogram in red, assuming the branching ratio $B(h \rightarrow aa \rightarrow \tau\tau bb) = 10\%$. The figures are taken from Ref. [15].	94
3.17	Post-fit distributions of $m_{\tau\tau}$ for the $e\mu$ ‘> 1 b tag’ category, with the subregion indicated. The signal (35 GeV) is shown as a box histogram in red, assuming the branching ratio $B(h \rightarrow aa \rightarrow \tau\tau bb) = 10\%$. The figures are taken from Ref. [15].	95
3.18	Observed and expected model-independent exclusion limits at 95% CL on the branching ratio $B(h \rightarrow aa \rightarrow \tau\tau bb)$ (%). The three final states are shown: $\mu\tau_h$ (upper left), $e\tau_h$ (upper right), and $e\mu$ (lower left), as well as the combination (lower right). The figures are taken from Ref. [15].	97
3.19	Branching ratio $B(aa \rightarrow bb\tau\tau)$ as a function of $\tan\beta$ predicted in 2HDM+S for a pseudoscalar mass of 40 GeV. The figure is taken from Ref. [16]. . . .	98
3.20	Observed exclusion limits at 95% CL on the branching ratio $B(h \rightarrow aa)$ (%). The results are obtained from the combination of all final states, interpreted in 2HDM+S Type I (independent of $\tan\beta$), Type II ($\tan\beta = 2$), Type III ($\tan\beta = 2$), and Type IV ($\tan\beta = 0.6$), respectively. The figure is taken from Ref. [15].	100
3.21	Observed exclusion limits at 95% CL on the branching ratio $B(h \rightarrow aa)$ (%), plotted in the plane of $\tan\beta$ vs. signal mass. The results are obtained from the combination of all final states, interpreted in 2HDM+S Type III (upper) and Type IV (lower), respectively. The figures are taken from Ref. [15]. . .	101
3.22	Fitted background models in $m_{\mu\mu}$ in the five categories. The figures are taken from Ref. [15].	104
3.23	Observed and expected model-independent exclusion limits at 95% CL on the branching ratio $B(h \rightarrow aa \rightarrow \mu\mu bb)$. The figure is taken from Ref. [15].	105
3.24	Observed and expected exclusion limits at 95% CL on the branching ratio $B(h \rightarrow aa \rightarrow llbb)$ (%) within 2HDM+S, where $l = \mu, \tau$. The figure is taken from Ref. [15].	108

3.25	Observed exclusion limits at 95% CL on the branching ratio $B(h \rightarrow aa)$ (%). The results are obtained from the combination of the $\tau\tau bb$ and $\mu\mu bb$ final states, interpreted in 2HDM+S Type I (independent of $\tan\beta$), Type II ($\tan\beta = 2$), Type III ($\tan\beta = 2$), and Type IV ($\tan\beta = 0.6$), respectively. The figure is taken from Ref. [15].	109
3.26	Observed exclusion limits at 95% CL on the branching ratio $B(h \rightarrow aa)$ (%), plotted in the plane of $\tan\beta$ vs. signal mass. The results are obtained from the combination of the $\tau\tau bb$ and $\mu\mu bb$ final states, interpreted in 2HDM+S Type III (upper) and Type IV (lower), respectively. The figures are taken from Ref. [15].	110
3.27	Summary of observed and expected exclusion limits at 95% CL on the branching ratio $B(h \rightarrow aa)$ in 2HDM+S Type II at $\tan\beta = 2$. The figure is taken from Ref. [17].	111
3.28	Phase-2 trigger study for the search channel $h \rightarrow aa \rightarrow bb\tau\tau$. Top: trigger efficiency of the isolated single- μ trigger ($p_T > 24 GeV$) as a function of the generated muon p_T . Bottom: expected exclusion limits at 95% CL on the branching ratio $B(h \rightarrow aa \rightarrow bb\tau\tau)$ (%), with and without the trigger efficiency corrections. The figures are taken from Ref. [18].	113
3.29	Event yield of the $h \rightarrow aa \rightarrow \tau\tau bb$ signal as a function of the muon and τ_h p_T in the $\mu\tau_h$ final state, prior to applying trigger p_T requirements.	114
5.1	An example fully-connected NN with two hidden layers is shown, with the first hidden layer consisting of 12 neurons and the second hidden layer consisting of 8 neurons. The NN has 6 input features and produces 1 output. The figure is generated using the NN-SVG tool [19].	122
5.2	An example of a convolution operation of a 3×3 filter applied to a 5×5 input, with valid padding and a stride size of 1. The output value 2 in blue corresponds to the dot product of the input subgrid in red and the filter in green: $(-2) \times (1) + (1) \times (1) + (2) \times (1) + (3) \times (0) + (-4) \times (0) + (0) \times (0) + (1) \times (-1) + (-3) \times (-1) + (1) \times (-1) = 2$. The output dimensions are calculated by: $\lfloor \frac{H-M+2P}{S} + 1 \rfloor \times \lfloor \frac{W-N+2P}{S} + 1 \rfloor = \lfloor \frac{5-3+2(0)}{1} + 1 \rfloor \times \lfloor \frac{5-3+2(0)}{1} + 1 \rfloor = 3 \times 3$	124
5.3	An example CNN with a 2D convolutional layer and a pooling layer followed by a fully-connected NN. The convolutional layer has 2 filters of dimensions 9×9 , operating with valid padding and a stride size of 1. The pooling layer has a filter of dimensions 5×5 , operating with valid padding and stride dimensions the same as the pool dimensions. The pooling output is flattened into a vector fed as inputs to a fully-connected NN, which has a hidden layer with 128 neurons and an output layer of 1 neuron. The figure is generated using the NN-SVG tool [19].	126
5.4	hls4ml workflow of converting an ML model into an FPGA implementation. The figure is taken from Ref. [20].	129
6.1	Comparison of anomaly detection trigger and traditional trigger in terms of rate control and model independence. The plot is for illustration purposes and is not drawn to scale.	133

6.2	Illustration of anomaly detection. Anomalous data (rare occurrences: non-rectangular shapes, in red) are identified as deviations from normal data (abundant occurrences: rectangular shapes, in blue). In the context of the LHC, normal events correspond to abundant SM processes with large cross sections, such as soft QCD productions. Anomalous data, on the other hand, correspond to rare SM processes or BSM processes with signatures deviating from normal data.	136
6.3	Input features to CICADA: energy deposits at the trigger tower level in the calorimeter layer-1 trigger subsystem, received from the ECAL and HCAL. Each input event consists of energy deposits in the $18(i\phi) \times 14(i\eta) = 252$ trigger regions, where each region is a 4×4 trigger towers. Here, $i\phi$ and $i\eta$ refer to special trigger coordinates, which take integer values to label the discrete region locations. The heat map displays an example of a ZeroBias data event, with the energy deposit values indicated by color intensity. The correspondence of a trigger region is marked by red squares in both the input heat map (left) and the trigger tower map (right). The figure is adapted from Ref. [21]	137
6.4	Illustration of the autoencoder that compresses and then reconstructs the regional energy deposits: input \rightarrow CNN encoder \rightarrow latent space \rightarrow CNN decoder \rightarrow output. The figure is generated using the PlotNeuralNet tool [22].	139
6.5	Configuration of the autoencoder model for CICADA implemented in Keras [23].	141
6.6	Comparison of inputs and reconstruction outputs from the autoencoder. In the ZeroBias event, the reconstruction is fairly good, with the major high-energy regions well-reconstructed in both intensity and location (anomaly score = 0.81). In the signal event, which is simulated SUEP, the model fails to reconstruct the high-energy regions in the lower part (anomaly score = 14.21). The figures are taken from Ref. [21].	142
6.7	Illustration of knowledge distillation: the smaller model, called the student model, learns to directly regress the anomaly score produced by the larger autoencoder model, called the teacher model. The figure is generated using the PlotNeuralNet [22] and the NN-SVG [19] tools.	144
6.8	Configuration of the student model for CICADA implemented in QKeras [24], which is the final model to be converted into an FPGA implementation. . .	145
6.9	Anomaly score distributions comparing ZeroBias data and various signals. .	146
6.10	ROC curves plotted for various signals vs. ZeroBias data. The trigger rate on the horizontal axis refers to the rate of triggering on ZeroBias events in MHz, with the dashed line indicating a trigger rate of 3 kHz. Practically, the anomaly detection trigger is expected to operate at a trigger rate around and below this line due to the limited bandwidth shared by many different physics groups.	147
6.11	ROC curves in the region of low background efficiency, comparing CICADA (red) and the HT trigger (black). Several practical CICADA rates are indicated by the vertical dashed lines.	148

6.12	Signatures that CICADA prefers to trigger on. Top: trigger-level jet multiplicity profiles before and after applying the CICADA trigger. Bottom: jet p_T profiles before and after applying the CICADA trigger. The histograms in purple show the ZeroBias profiles where no threshold cut is applied on the CICADA score. The histograms in other colors show the profiles at various threshold cuts on the CICADA score.	151
6.13	Illustration of rate overlap between CICADA and the existing trigger menu. The CICADA pure rate as a function of the CICADA overall rate. The pure rate refers to the rate attributed to CICADA only, without any other existing trigger paths being fired. The figure is taken from Ref. [21].	152
6.14	Illustration of the rate stability of CICADA across different 2023 runs. Top: CICADA score distributions. Bottom: CICADA rate as a function of the score threshold. The figures are taken from Ref. [21].	153
7.1	Illustration of genetic programming approach to SR, where expressions are represented as trees. Mutation: a tree node is randomly chosen, and its content is changed (top). Crossover: subtrees from two expression trees are swapped (bottom).	158
7.2	Illustration of NN approach to SR. Unary operator (left): activation functions can be generalized to more different mathematical operations, such as $\sin(x)$, which activate one neuron. Binary operator (right): activation functions can also be generalized to operations involving two inputs, such as $x \times y$, which activate two neurons simultaneously.	159
7.3	Comparison of the original mathematical function and the LUT-based approximation, which is discretized within a finite range by a pre-computed table of input-output pairs. Two functions are illustrated: \sin (left) and \tan (right). Two table definitions with different granularity are compared, with the precision of $\langle 12, 6 \rangle$. The table definition is denoted as $\text{LUT}(x_{\min}, x_{\max}, \text{table granularity})$. The figures are taken from Ref. [25].	161
7.4	Comparison of ROC curves of the baseline quantization-aware trained (QAT) NN and the trigonometric model ($c_{\max} = 80$) with and without LUT-based approximations applied. The numbers in brackets are the corresponding ROC AUC values. The figure is taken from Ref. [25].	166
7.5	Accuracy of the symbolic models relative to the baseline NN. Three different model sizes are considered: $c_{\max} = 20, 40$, and 80 . Symbolic models with and without LUT-based approximations are compared. The figures are taken from Ref. [25].	167
7.6	DSP utilization on an FPAG. Three different model sizes are considered: $c_{\max} = 20, 40$, and 80 . Symbolic models with and without LUT-based approximations are compared with the baseline NN. The figures are taken from Ref. [25].	168
7.7	LUT utilization on an FPAG. Three different model sizes are considered: $c_{\max} = 20, 40$, and 80 . Symbolic models with and without LUT-based approximations are compared with the baseline NN. The figures are taken from Ref. [25].	169

7.8	Inference latency on an FPGA. Three different model sizes are considered: $c_{\max}=20, 40$, and 80 . Symbolic models with and without LUT-based approximations are compared with the baseline NN. The figures are taken from Ref. [25].	170
7.9	Symbolic models with and without latency-aware training (LAT) are compared in terms of accuracy relative to the baseline NN (top) and latency (bottom). Seven different model sizes are considered: $c_{\max}=20, 30, 40, 50, 60, 70$, and 80 . Two bit widths are considered: $\langle 18, 8 \rangle$ and $\langle 16, 6 \rangle$. The figures are taken from Ref. [25].	171

List of Tables

1.1	Field content of the SM. The bold numbers denote the fundamental representations of the non-Abelian gauge groups: 3 in $SU(3)_C$ and 2 in $SU(2)_L$, while 1 denotes a singlet, indicating the field is not charged under that group. The numbers under $U(1)_Y$ denote the hypercharge Y : a field with $Y = 0$ is not charged under $U(1)_Y$. The representations and charges are the same across the three generations of fermions.	10
1.2	Particle content of the SM. The numbers are quoted from Ref. [26].	15
1.3	Four coupling types of 2HDMs (and of 2HDM+S) that forbid flavor-changing neutral currents at tree level [27, 28].	25
3.1	Summary of trigger p_T thresholds in the three data-taking years [15]. The p_T requirements apply to the corresponding HLT-reconstructed objects. In each final state per year, events are selected if they pass any of the available triggers.	57
3.2	Composition ratios of the three leading background processes in the background class.	65
3.3	Numbers of training events in the signal and background classes. All events are required to pass the baseline selection.	66
3.4	Summary of the DNN architecture. All DNNs have two hidden layers, each followed by a dropout layer. The numbers of nodes in the input and hidden layers, as well as the dropout rates, are shown.	77
3.5	Classification accuracy (with a score threshold of 0.5) evaluated on the test set.	78
3.6	Definition of the subregions for $m_{\tau\tau}$ by threshold cuts on the DNN score.	85
7.1	Expressions of the trigonometric model ($c_{\max} = 20$), with ROC AUC shown. The table is taken from Ref. [25].	165
7.2	Expressions for the top quark class ($c_{\max} = 40$), with ROC AUC shown. The table is taken from Ref. [25].	165

Chapter 0

Introduction

Exotic and rare decays of the recently discovered Higgs boson [29–31] present an avenue for searches of new physics phenomena. However, searches for extended Higgs sectors with light partners of the 125-GeV Higgs boson present an experimental challenge in reconstructing their signatures in profuse backgrounds in proton-proton collisions at 13 TeV. The challenge begins early in the data acquisition process itself. Machine learning (ML) techniques are among the most promising options for selecting these rare events and analyzing them. This thesis presents several interconnected results, starting with an ML-enabled study of existing data, followed by the implementation of ML-enabled trigger electronics systems for ongoing data acquisition, and potential improvements for the future.

Part I of this thesis presents a physics analysis. It explores the search for new physics beyond the Standard Model (BSM) through the exotic decays of the Higgs boson, analyzing proton-proton collision data collected by the Compact Muon Solenoid (CMS) detector at the Large Hadron Collider (LHC). This analysis indicated no deviations from the Standard Model (SM) predictions with the current data and analysis strategies.

Despite this null result, exclusion limits can be derived. These limits indicate that any signal production rate, or coupling constant associated with the interaction, exceeding a certain value is ruled out with a certain level of statistical confidence. This suggests

either the non-existence of the signal or that the current experimental framework lacks the sensitivity to detect the signal if it exists. These constraints can also be interpreted within the parameter phase space of theoretical models that predict these new physics phenomena. These results enable experimentalists to refine their analysis techniques to probe previously unexplored phase space while inspiring theorists to adjust their current models or develop new ones.

Analyses searching for new physics often face the challenge of limited statistics in their data samples. While it is possible to optimize existing analysis techniques and collect more data for future analyses, this process is typically lengthy. For instance, the discovery of the Higgs boson was the result of many years of persistent efforts.

However, we cannot simply wait for more data to be collected and rely solely on incremental improvements to current analysis techniques. Instead, we must continuously pursue innovative and unconventional methods to improve our existing approaches. ML has proven to be effective in advancing various scientific and technological sectors. There is no doubt that ML is transforming, and will continue to transform, the way we do experimental high-energy physics (HEP), potentially revolutionizing the field.

Part II of this thesis introduces ML methods to address the limitations of current strategies in the search for new physics at the LHC, aiming for more fundamental advancements.

For example, the analysis presented in Part I faces a significant challenge: the target signature includes soft (low-momentum) final-state objects, resulting in over 90% of the signal being filtered out by the Level-1 trigger before reaching the analysis phase. This occurs because the analysis relies on a traditional trigger strategy that implements a simple momentum cut on the physics objects for event selections. Such a trigger strategy typically requires a relatively high momentum threshold for selecting objects to maintain a manageable output rate, given the extreme collision rate of 40 MHz at the input. As a result, the decay channel with the highest branching ratio (two hadronically decaying tau leptons) is completely discarded from the analysis due to the high tau momentum

threshold required by the existing trigger path. This is a common issue in many new physics searches at the LHC.

To address this, a novel trigger strategy is developed and introduced in CMS, using an ML-based anomaly detection technique to search for new physics in a model-agnostic manner. This strategy aims to improve overall sensitivity at a lower level, specifically at the Level-1 trigger. The Level-1 trigger is crucial as it processes the first round of event selections immediately after the collider produces raw collision events. Events that do not pass the trigger selections are permanently discarded, making it essential to preserve as many potential events as possible at the Level-1 trigger. The anomaly detection trigger can identify a wide range of rare SM and BSM signals without prior knowledge of any physics models, outperforming traditional triggers in certain areas. This novel trigger strategy is particularly effective for challenging signatures that traditional triggers would typically reject.

The development of the anomaly detection trigger faces challenges due to the so-called Level-1 constraints, which arise from the scarcity of computational resources and the need for low-latency decisions. The current Level-1 trigger system requires a latency below $\mathcal{O}(1)$ μs , and the algorithm must fit on a single Field-Programmable Gate Array (FPGA) board with limited resources. To address these challenges, various techniques are employed to compress the size of the model while maintaining competitive performance.

In response to the increasing demands for fast ML algorithms in resource-constrained environments such as the LHC experiments, a novel model compression method using symbolic regression (SR) is developed. SR is shown to be effective in a benchmark LHC dataset for physics object identification, achieving nanosecond-latency inference speeds on an FPGA with significantly lower resource consumption compared to traditional modeling approaches. It is demonstrated that SR has potential as a promising alternative to neural network-based models in various fields.

Part I

Search for exotic Higgs boson decays with CMS

Chapter 1

Theoretical motivations

Before discussing the analysis of the search for exotic Higgs boson decays, it is essential to introduce the Standard Model (SM) of particle physics, as presented in Sec. 1.1. The SM currently provides the most accurate theoretical description of elementary particles and their fundamental interactions. Yet, it is incomplete, leaving many questions about the universe unresolved. This motivates the development of new physics models beyond the Standard Model (BSM). Among the various BSM theories, two-Higgs doublet models are a straightforward extension of the SM scalar sector. These models predict that the 125-GeV Higgs boson can decay exotically into new particles not predicted by the SM. Such models have not yet been ruled out by experimental results published so far, thus motivating further exploration. The motivations and phenomenology of these models are discussed in Sec. 1.2.

1.1 The Standard Model of particle physics

The SM is a theoretical framework describing the fundamental laws of nature that govern elementary particles and their interactions [32–35]. It predicts the existence of elementary matter particles, namely quarks and leptons, and accounts for three of the four known fundamental interactions: electromagnetic, weak, and strong forces. The SM is formulated as a quantum field theory, with its dynamics described by a Lagrangian. In this theory,

quantum fields are the fundamental entities defined in spacetime, and particles emerge as the excited states of the underlying fields.

Matter particles are represented by spin- $\frac{1}{2}$ fermionic fields. Gauge symmetries give rise to spin-1 bosonic fields, where gauge bosons mediate the forces between fermions. There is a special spin-0 bosonic field, called the Higgs field, which is responsible for electroweak symmetry breaking, giving mass to some of the gauge bosons via the Brout–Englert–Higgs mechanism [36–41]. Matter particles acquire mass through their Yukawa coupling with the Higgs field [33]. The quanta of the Higgs field correspond to the Higgs boson, which acquires mass through self-interaction.

All particles predicted by the SM, including the Higgs boson, have been experimentally observed, and the model parameters have been precisely measured to be consistent within the SM framework [2, 29–31, 42].

The construction of the particle content and their interactions are given in Sec. 1.1.1. The electroweak symmetry breaking and the Higgs mechanism are discussed in Sec. 1.1.2. The Higgs boson in the context of the LHC is discussed in Sec. 1.1.3.

1.1.1 Gauge symmetry and Lagrangian

The physics of the SM [32–35] is encoded in the Lagrangian (density), which takes the form

$$\mathcal{L}_{\text{SM}} = \mathcal{L}_{\text{gauge}} + \mathcal{L}_{\text{fermions}} + \mathcal{L}_{\text{Yukawa}} + \mathcal{L}_{\text{Higgs}}. \quad (1.1)$$

This Lagrangian respects the global Poincare symmetry and the internal gauge symmetry $\text{SU}(3)_C \times \text{SU}(2)_L \times \text{U}(1)_Y$.

Poincare symmetry, the symmetry of spacetime in special relativity, dictates that physical laws remain invariant under spatial and temporal translations, spatial rotations, and Lorentz boosts. These symmetries are global, meaning that transformations apply uniformly across spacetime points (i.e., a spatial translation of a system applies the same translation to every point within the system). According to Noether’s theorem [43], each of these symmetries corresponds to a conservation law: the conservation of momentum and

energy, angular momentum, and center-of-mass velocity, respectively. Therefore, Poincare symmetry establishes the fundamental constraints for the spacetime framework in which quantum fields are embedded.

In gauge theory, when quantum fields undergo local gauge transformations (i.e., phase changes that depend on spacetime points), additional terms arise from the kinetic terms in the Lagrangian because the transformations vary from point to point in spacetime, inducing dynamics. These additional kinetic terms, which vary with spacetime coordinates, are nontrivial for the Lagrangian and can alter the underlying physics if not properly treated. To counteract these terms and maintain gauge invariance, new gauge fields that transform under the same local gauge transformations are introduced. This ensures that the Lagrangian remains gauge invariant, preserving the physical laws.

The introduction of gauge fields leads to their dynamic behavior and interactions with other fields that are coupled to them. Consequently, this results in the dynamic mediation of forces between fermions through the exchange of gauge bosons, with each force arising from a specific gauge symmetry.

The $SU(3)_C$ subgroup generates strong interactions, where C stands for color charge. There are three color charges, namely red, blue, and green (purely a naming convention, as they could be called anything else without changing the physics). Quarks are charged under this group and form color triplets, where a triplet represents the three color states of a quark field. The $SU(3)_C$ transformations on a color triplet can be thought of as rotations within its color space. The gauge symmetry ensures that all color states are dynamically equivalent, with distinctions arising only from the rules of color exchange to maintain charge conservation during interactions.

There are eight generators of $SU(3)_C$, corresponding to eight independent transformations, which generate eight types of gauge bosons called gluons. According to Yang-Mills theory [44], a gauge theory based on a non-Abelian symmetry group, these gluons are massless spin-1 bosons that can also self-interact. The theory of strong interactions is called quantum chromodynamics (QCD).

The product group $SU(2)_L \times U(1)_Y$ generates electroweak interactions, where L stands for left-handedness and Y stands for hypercharge. The theory is chiral in nature, as left-handed fermionic fields are charged under $SU(2)_L$ to form doublets, while right-handed ones are neutral and form singlets. The three generators of $SU(2)_L$ are called weak isospin, and the generator of $U(1)_Y$ is called weak hypercharge, resulting in four massless gauge bosons in total. However, the bosons enabling weak decays must be massive to account for their short lifetimes.

After electroweak symmetry breaking, the $SU(2)_L \times U(1)_Y$ symmetry is broken and reduced to $U(1)_{EM}$, splitting the electroweak force into the observed weak force and electromagnetism. Through the Higgs mechanism, three weak bosons acquire mass and are identified as the observed massive W^\pm and Z^0 bosons that mediate the weak force, with the $SU(2)_L$ symmetry being spontaneously broken. The remaining weak boson remains massless and is identified as the photon γ , which mediates the electromagnetic force, corresponding to the $U(1)_{EM}$ symmetry. The electroweak symmetry breaking and the Higgs mechanism will be discussed in Sec. 1.1.2.

The SM Lagrangian in Eq. 1.1 is decomposed to reflect the different sectors and physical manifestations.

- $\mathcal{L}_{\text{gauge}}$ represents the kinetic terms for the gauge fields:

$$\mathcal{L}_{\text{gauge}} = -\frac{1}{4}G_{\mu\nu}^a G^{a\mu\nu} - \frac{1}{4}W_{\mu\nu}^a W^{a\mu\nu} - \frac{1}{4}B_{\mu\nu} B^{\mu\nu}, \quad (1.2)$$

where $G_{\mu\nu}^{a=1,\dots,8}$, $W_{\mu\nu}^{a=1,2,3}$, and $B_{\mu\nu}$ are the $SU(3)_C$, $SU(2)_L$, and $U(1)_Y$ gauge field tensors, respectively. These tensors encode the spacetime propagation of the gauge fields and their self-interaction strength. Only the non-Abelian gauge fields G_μ^a and W_μ^a are self-interacting, whereas the Abelian gauge field B_μ propagates freely until it couples with other fields.

- $\mathcal{L}_{\text{fermions}}$ represents the kinetic terms for quarks and leptons, which take the generic

form for each fermionic field ψ :

$$\mathcal{L}_\psi = i\bar{\psi}\gamma^\mu D_\mu\psi, \quad (1.3)$$

where D_μ is the covariant derivative that incorporates gauge field interactions, and γ^μ are the gamma matrices, which are essential for forming the basis of fermions in relativistic quantum mechanics. Together, these elements ensure the gauge and Lorentz invariance of the theory. This generic form encodes how fermions propagate in spacetime and the forces they experience through their interactions with gauge bosons. For the first generation of matter, the left-handed quarks and left-handed leptons form $SU(2)_L$ doublets, namely $q_L = \begin{pmatrix} u_L \\ d_L \end{pmatrix}$ and $l_L = \begin{pmatrix} e_L \\ \nu_L \end{pmatrix}$, while the right-handed ones form singlets: u_R , d_R , and e_R . The exact form of D_μ depends on the gauge group under which the fermion is charged:

$$\begin{aligned} D_\mu &= \left(\partial_\mu - ig_s G_\mu^a T^a - ig W_\mu^a \tau^a - ig' B_\mu Y \right), & \text{for } q_L; \\ D_\mu &= \left(\partial_\mu - ig_s G_\mu^a T^a - ig' B_\mu Y \right), & \text{for } u_R, d_R; \\ D_\mu &= \left(\partial_\mu - ig W_\mu^a \tau^a - ig' B_\mu Y \right), & \text{for } l_L; \\ D_\mu &= \left(\partial_\mu - ig' B_\mu Y \right), & \text{for } e_R. \end{aligned} \quad (1.4)$$

Here, $T^{a=1,\dots,8}$ (3×3 matrices acting on color triplets), $\tau^{a=1,2,3}$ (2×2 matrices acting on weak isospin doublets), and Y (a constant that takes different values among fermions) are the generators of $SU(3)_C$, $SU(2)_L$, and $U(1)_Y$, respectively, with gauge couplings g_s , g , and g' . These extend similarly to the other two generations of fermions.

- $\mathcal{L}_{\text{Yukawa}}$ represents the Yukawa coupling terms connecting different fermions via a scalar field ϕ , which take the generic form:

$$y_{ij}\bar{\psi}_i\phi\psi_j, \quad (1.5)$$

Table 1.1: Field content of the SM. The bold numbers denote the fundamental representations of the non-Abelian gauge groups: **3** in $SU(3)_C$ and **2** in $SU(2)_L$, while **1** denotes a singlet, indicating the field is not charged under that group. The numbers under $U(1)_Y$ denote the hypercharge Y : a field with $Y = 0$ is not charged under $U(1)_Y$. The representations and charges are the same across the three generations of fermions.

Field	$SU(3)_C$	$SU(2)_L$	$U(1)_Y$
Spin-1/2 matter field			
q_L	3	2	1/6
u_R	3	1	2/3
d_R	3	1	-1/3
l_L	1	2	-1/2
e_R	1	1	-1
Spin-1 gauge field			
$G^{a=1,\dots,8}$	8	1	0
$W^{a=1,2,3}$	1	3	0
B	1	1	0
Spin-0 scalar field			
Φ	1	2	1/2

where y_{ij} are the Yukawa coupling constants specific to different pairs of fermions represented by the indices i and j . Fermion masses are generated through these couplings only after electroweak symmetry breaking, with each fermion mass being directly proportional to its corresponding Yukawa coupling constant.

- $\mathcal{L}_{\text{Higgs}}$ represents the Higgs sector:

$$\mathcal{L}_{\text{Higgs}} = |D_\mu \Phi|^2 - V(\Phi), \quad (1.6)$$

where $V(\Phi)$ is the potential of the Higgs field Φ , responsible for electroweak symmetry breaking. This part will be described in Sec. 1.1.2.

The field content of the SM is summarized in Tab. 1.1.

1.1.2 Electroweak symmetry breaking and Higgs mechanism

An important observation about the SM Lagrangian in Eq. 1.1 is that it does not include mass terms in order to preserve gauge invariance. If a massive gauge field A_μ existed,

there would be a mass term of the form $m^2 A_\mu A^\mu$ in the Lagrangian. However, such a term depends on the choice of gauge, thus spoiling gauge invariance. The same principle applies to fermions, where a mass term of the form $m^2 \bar{\psi}\psi = m^2(\bar{\psi}_L\psi_R + \bar{\psi}_R\psi_L)$ breaks the $SU(2)_L$ symmetry, which is chiral in nature. Therefore, all fermions and gauge bosons in the SM must be initially massless. However, the observed W/Z bosons and fermions are massive, indicating that the original gauge symmetry must be broken. This phenomenon is explained by the Brout–Englert–Higgs mechanism [36–41] as the electroweak symmetry is spontaneously broken, which consequentially gives mass to the W/Z bosons. Concurrently, fermions acquire mass through their Yukawa couplings with the Higgs field [33].

The Higgs Lagrangian reads:

$$\mathcal{L}_{\text{Higgs}} = D_\mu \Phi^\dagger D^\mu \Phi - V(\Phi), \quad (1.7)$$

where the Higgs field is a complex scalar doublet charged under the electroweak $SU(2)_L \times U(1)_Y$:

$$\Phi = \begin{pmatrix} \phi^+ \\ \phi^0 \end{pmatrix}, \quad (1.8)$$

and therefore

$$D_\mu \Phi = \left(\partial_\mu - ig W_\mu^a \tau^a - i \frac{g'}{2} B_\mu \right) \Phi. \quad (1.9)$$

The Higgs potential takes the form:

$$V(\Phi) = -\mu^2 \Phi^\dagger \Phi + \lambda (\Phi^\dagger \Phi)^2, \quad (1.10)$$

where μ^2 and λ are real numbers to ensure unitarity. The parameter λ controls vacuum stability and must be a positive number to avoid minima at $-\infty$. The parameter μ^2 , if positive, triggers electroweak symmetry breaking. This can be seen by plotting the potential when $\mu^2 > 0$ and $\lambda > 0$, which takes the shape of a “Mexican hat”, as shown in Fig. 1.1. At zero field value, the potential is at a local maximum (top of the hat). Instead, the minimum of the potential is located at a non-zero field value, indicating that the Higgs

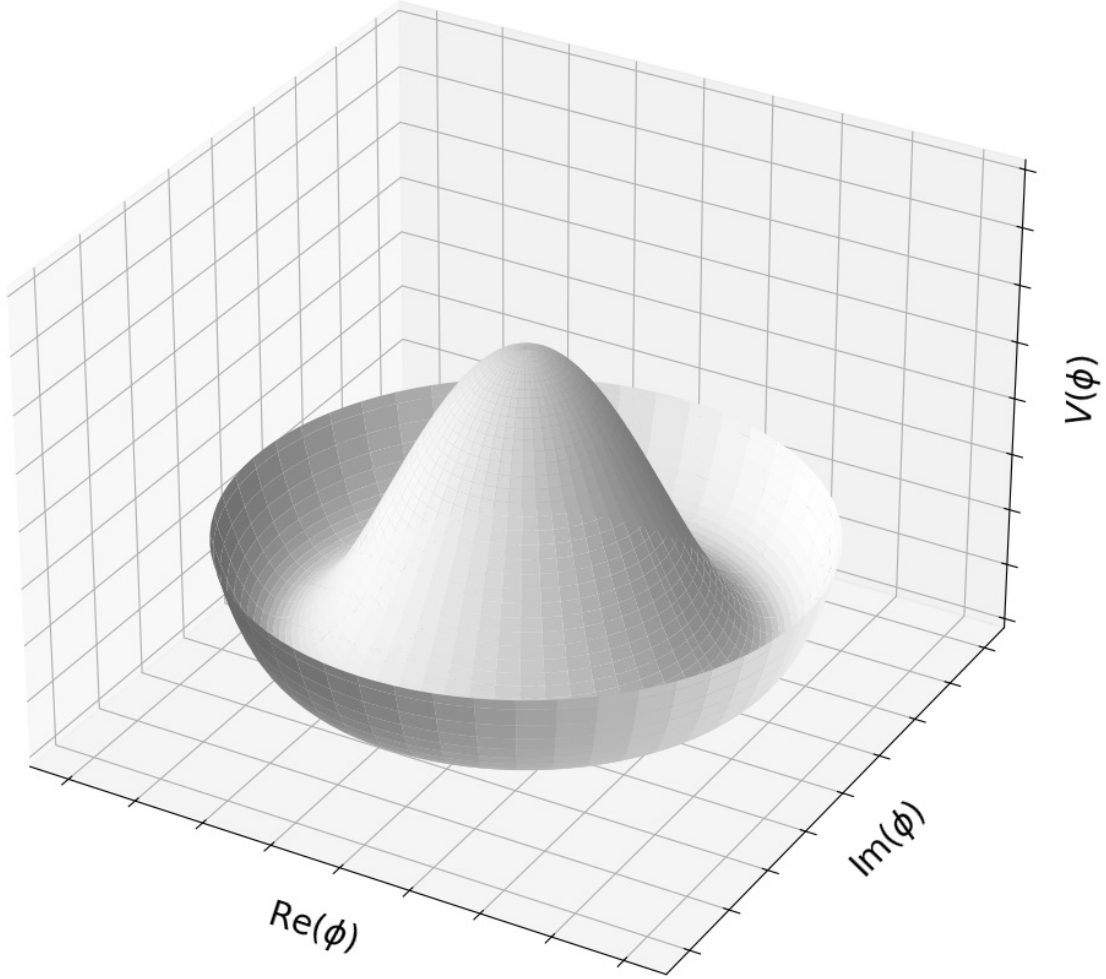


Figure 1.1: The shape of the Higgs potential where the minimum is at a non-zero field value.

field attains a non-zero vacuum expectation value (vev):

$$\langle \Phi \rangle_0 = \begin{pmatrix} 0 \\ \frac{\nu}{\sqrt{2}} \end{pmatrix} \quad \text{with} \quad \nu = \sqrt{\frac{\mu^2}{\lambda}}. \quad (1.11)$$

In fact, there exist infinitely many solutions that are degenerate to Eq. 1.11 as one can rotate it by a $U(1)$ transformation while the potential remains at its minimum value. How-

ever, when the system condenses by choosing one of these gauge choices for the vacuum state, this configuration no longer respects the original gauge symmetry. It should be emphasized that only the specific configuration breaks the gauge symmetry once the vacuum state is chosen, whereas the Lagrangian itself remains gauge invariant. This phenomenon is referred to as spontaneous electroweak symmetry breaking: $SU(2)_L \times U(1)_Y \rightarrow U(1)_{EM}$.

The consequences of electroweak symmetry breaking include the generation of masses for the gauge bosons through the Higgs mechanism, the existence of the Higgs boson, and the generation of masses for the fermions via Yukawa couplings. These are realized by first expanding the Higgs field around its vev in the gauge from Eq. 1.11:

$$\Phi = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 \\ \nu + h(x) \end{pmatrix}, \quad (1.12)$$

where $h(x)$ is a real scalar field representing the fluctuations around the vacuum state.

Substituting Eq. 1.12 into the Higgs Lagrangian in Eq. 1.7, we get:

$$\begin{aligned} |D_\mu \Phi|^2 - V(\Phi) = & \frac{(\nu + h)^2}{8} \left(g^2 (W_\mu^1)^2 + g^2 (W_\mu^2)^2 + (gW_\mu^3 - g'B_\mu)^2 \right) \\ & + \frac{1}{2} (\partial_\mu h)^2 + \nu^2 \lambda h^2 + \nu \lambda h^3 + \frac{\lambda}{4} h^4, \end{aligned} \quad (1.13)$$

where the second line gives rise to the Higgs boson, which has cubic and quartic self-couplings and has a mass at tree level:

$$m_h = \sqrt{2\lambda\nu}. \quad (1.14)$$

To read off the masses of the gauge bosons, one can rewrite the first line of Eq. 1.13 in terms of the physical fields:

$$\begin{aligned} & m_W^2 W_\mu^- W^{+\mu} + \frac{1}{2} m_Z^2 Z_\mu Z^\mu + \frac{1}{2} m_\gamma^2 A_\mu A^\mu \\ & + \frac{2m_W^2}{\nu} h W_\mu^- W^{+\mu} + \frac{m_W^2}{\nu^2} h^2 W_\mu^- W^{+\mu} + \frac{m_Z^2}{\nu} h Z_\mu Z^\mu + \frac{m_Z^2}{2\nu^2} h^2 Z_\mu Z^\mu, \end{aligned} \quad (1.15)$$

where $W_\mu^\pm = (W_\mu^1 \mp iW_\mu^2)/\sqrt{2}$ correspond to the W^\pm bosons, $Z_\mu = (gW_\mu^3 - g'B_\mu)/\sqrt{g^2 + g'^2}$ corresponds to the Z boson, and $A_\mu = (g'W_\mu^3 + gB_\mu)/\sqrt{g^2 + g'^2}$ is orthogonal to Z_μ and corresponds to the photon γ . The masses read:

$$m_W = \frac{\nu g}{2}, \quad m_Z = \frac{\nu \sqrt{g^2 + g'^2}}{2}, \quad m_\gamma = 0. \quad (1.16)$$

The second line of Eq. 1.15 gives rise to couplings between the Higgs boson and the heavy gauge bosons with coupling strengths:

$$g_{hWW} = \frac{2m_W^2}{\nu}, \quad g_{hZZ} = \frac{m_Z^2}{\nu}, \quad g_{hhWW} = \frac{m_W^2}{\nu^2}, \quad g_{hhZZ} = \frac{m_Z^2}{2\nu^2}. \quad (1.17)$$

On the fermion side, the Yukawa terms for the first generation read:

$$\mathcal{L}_{\text{Yukawa}} = -y_u \bar{q}_L \tilde{\Phi} u_R - y_d \bar{q}_L \Phi d_R - y_e \bar{l}_L \tilde{\Phi} e_R + \text{h.c.}, \quad (1.18)$$

where $\tilde{\Phi} = i\sigma_2 \Phi^*$ is required to ensure the correct coupling pairs, and y_f is the Yukawa coupling strength for the corresponding flavor. After electroweak symmetry breaking, Eq. 1.18 becomes:

$$\begin{aligned} \mathcal{L}_{\text{Yukawa}} = & -m_u \bar{u}_L u_R - m_d \bar{d}_L d_R - m_e \bar{e}_L e_R \\ & - \frac{m_u}{\nu} h \bar{u}_L u_R - \frac{m_d}{\nu} h \bar{d}_L d_R - \frac{m_e}{\nu} h \bar{e}_L e_R + \text{h.c.}, \end{aligned} \quad (1.19)$$

where the fermion masses take the generic form:

$$m_f = \frac{y_f \nu}{\sqrt{2}}, \quad (1.20)$$

and the Higgs boson couples to a pair of same-flavor fermions at a strength proportional to the fermion mass:

$$g_{hff} = \frac{m_f}{\nu}. \quad (1.21)$$

The same coupling structure applies to all three generations of fermions.

Table 1.2: Particle content of the SM. The numbers are quoted from Ref. [26].

Particle	Symbol	Mass	Charge [q_e]	Forces
Spin-1/2 quark (matter)				
Up quark	u	2.16 MeV	+2/3	Strong, Weak, EM
Down quark	d	4.67 MeV	-1/3	Strong, Weak, EM
Strange quark	s	93.4 MeV	-1/3	Strong, Weak, EM
Charm quark	c	1.27 GeV	+2/3	Strong, Weak, EM
Bottom quark	b	4.18 GeV	-1/3	Strong, Weak, EM
Top quark	t	172.7 GeV	+2/3	Strong, Weak, EM
Spin-1/2 lepton (matter)				
Electron	e	0.511 MeV	-1	Weak, EM
Muon	μ	105.7 MeV	-1	Weak, EM
Tau	τ	1.777 GeV	-1	Weak, EM
Electron neutrino	ν_e	$\simeq 0$	0	Weak
Muon neutrino	ν_μ	$\simeq 0$	0	Weak
Tau neutrino	ν_τ	$\simeq 0$	0	Weak
Spin-1 gauge boson (force carrier)				
Gluon	g	0	0	Strong
W boson	W^\pm	80.4 GeV	± 1	Weak, EM
Z boson	Z	91.2 GeV	0	Weak
Photon	γ	0	0	EM
Spin-0 Higgs boson (special scalar)				
Higgs boson	h	125 GeV	0	—

The particle content of the SM is summarized in Tab. 1.2.

1.1.3 Higgs boson

In this section, we discuss the Higgs boson in the context of the LHC [2, 42]. Previously, we saw in Eq. 1.13 the existence of the Higgs boson as the excitation around the vev of the Higgs field. The Higgs boson couples to itself (see Eq. 1.13), to the heavy gauge bosons (see Eq. 1.15), and to the fermions (see Eq. 1.19). This coupling structure leads to a rich phenomenology, including various production mechanisms of the Higgs boson at particle colliders and multiple decay channels that can be explored to validate the SM.

Before discussing the production mechanisms of the Higgs boson and its production rate at the LHC, it is important to define the expected production rate of a particle as the product of the cross section and the instantaneous luminosity.

In particle physics, the cross section is a measure of the probability that a specific interaction will occur when two particles collide. It is typically expressed in units of barn (b), where 1 b is equivalent to 10^{-28} m^2 due to the small scale in high energy physics. Analogously, the cross section can be thought of as an effective area of the “target” particle that appears to the “projectile” particle during a collision. Thus, the smaller the cross section, the higher the chance for the “projectile” particle to miss the “target” particle during each pass. Theoretical models, such as the SM, allow for the prediction of the cross section for specific interactions, which can then be compared to the experimental data. On the other hand, the instantaneous luminosity is defined as the rate at which particles cross per unit area at a given point in the accelerator, measured in $\text{cm}^{-2}\text{s}^{-1}$.

Fig. 1.2 shows the SM prediction for the production cross sections of the Higgs boson at a mass of 125 GeV resulting from proton-proton collisions at the LHC. The total Higgs production cross section at a center-of-mass energy $\sqrt{s} = 13 \text{ TeV}$ is $55.4 \pm 2.5 \text{ pb}$ [1], which is around $5 \times 10^{-35} \text{ cm}^{-2}$. At a typical instantaneous luminosity of $2 \times 10^{34} \text{ cm}^{-2}\text{s}^{-1}$, achieved during LHC Run 2, it is expected to produce one Higgs boson per second. In the Run 2 data collected from 2016 to 2018, corresponding to an integrated luminosity of around 138 fb^{-1} , it is expected that nearly 8×10^6 Higgs bosons were produced.

The three dominant production channels are described below.

- **Gluon-gluon fusion (ggF).** This is the dominant channel where a pair of gluons, each from one of the protons, fuses via a virtual quark loop, and the Higgs boson is produced at the hqq vertex. The leading-order Feynman diagram of this process is depicted in Fig. 1.3a. Despite the fact that the leading order already involves loops, which are generally considered as small corrections in perturbative quantum field theory, this channel still accounts for around 87% of the total Higgs production cross section. The dominance is due to the hqq coupling strength being proportional to the quark mass (see Eq. 1.21), leading to the dominance from the top quark, the heaviest particle in the SM. The experimental signature of this channel is that the Higgs boson is typically isolated without nearby accompanying particles.

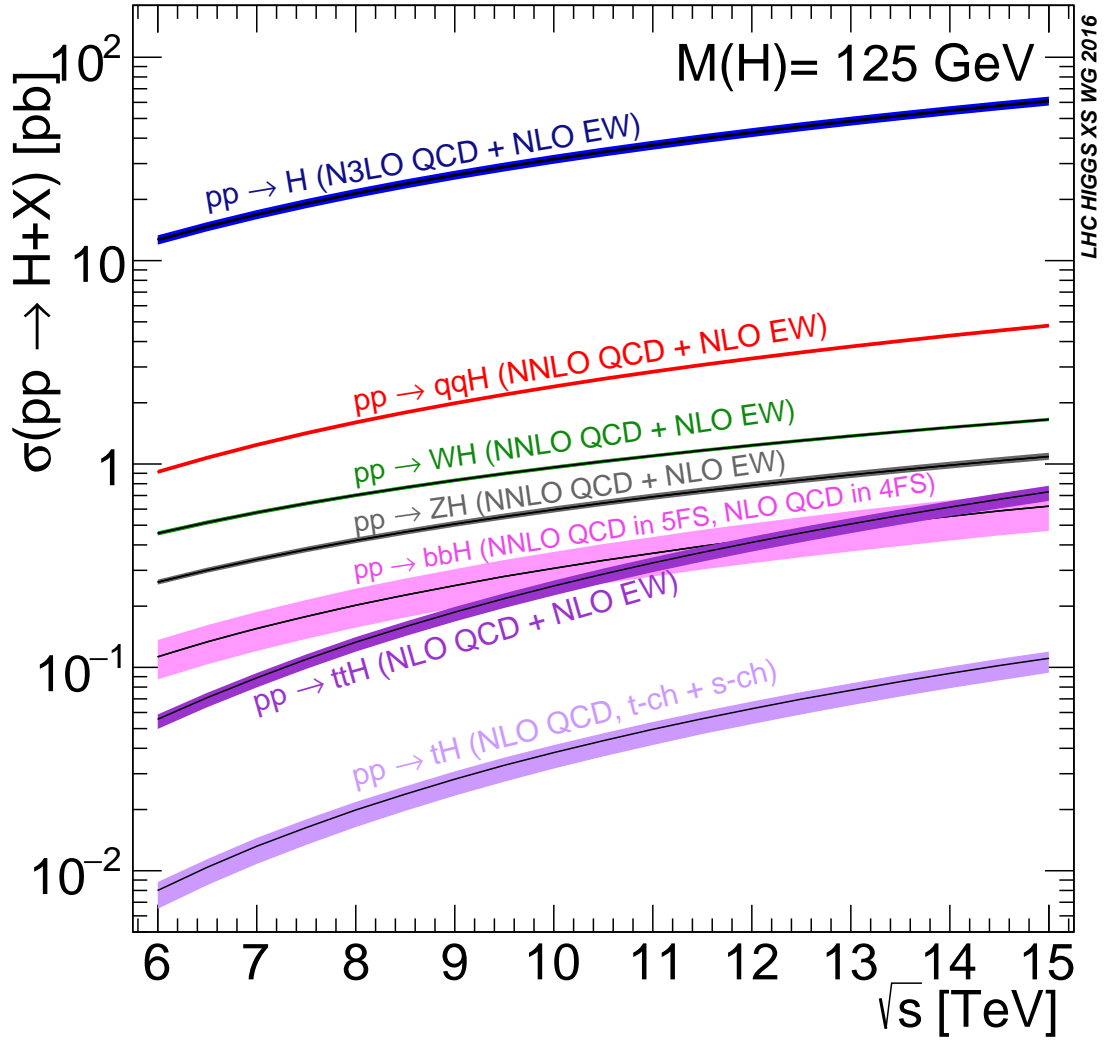


Figure 1.2: The SM prediction for the cross sections of the 125-GeV Higgs boson in different production modes, plotted as functions of the LHC energy. The figure is taken from Ref. [1]

- **Vector boson fusion (VBF).** This is the second most dominant channel, where each quark from one of the protons emits a vector boson $V = W/Z$, then the vector boson pair fuses to produce a Higgs boson at the hVV vertex, as depicted in Fig. 1.3b. This channel accounts for around 7% of the total Higgs production cross section. The two quarks that emitted the vector bosons remain in the final state as two high-energy jets in the forward and backward regions of the detector. This results in the signature of the Higgs boson accompanied by a back-to-back jet pair near the beam

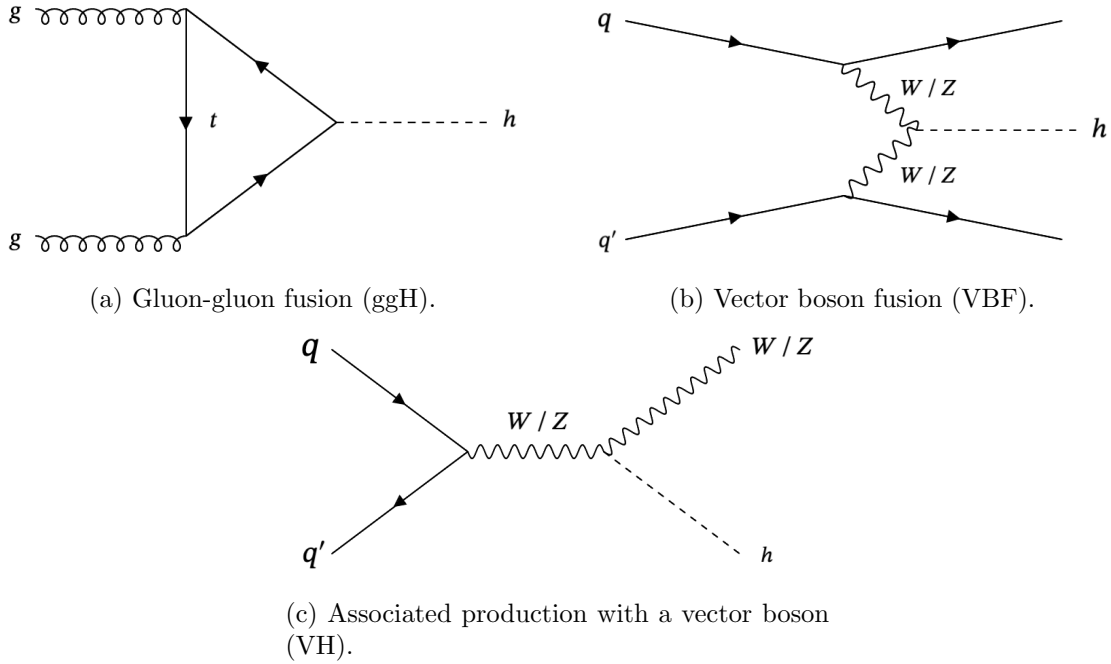


Figure 1.3: The three dominant Higgs production channels at the LHC.

axis with a large invariant mass.

- **Associated production with a vector boson (VH).** This channel involves a quark from a proton annihilating with an anti-quark from another proton to produce a W/Z boson, which then emits a Higgs boson, as depicted in Fig. 1.3c. This results in the Higgs boson being accompanied by a W/Z boson in the final state. Although the cross section is smaller than that of the VBF channel, the decay products from the associated V can be used to distinguish this process from other SM processes.

The Higgs boson may decay into different SM particles when allowed by kinematics and the coupling structure. Fig. 1.4 shows the SM prediction for the branching ratios of the different Higgs decay channels, where the branching ratio is defined as the probability of a particle decaying through a particular channel among all possible channels.

In general, the Higgs boson couples more strongly to heavier particles, meaning it decays more preferentially to heavier fermions than lighter fermions. The decay of the Higgs boson into bottom quarks is the most dominant channel, accounting for more than

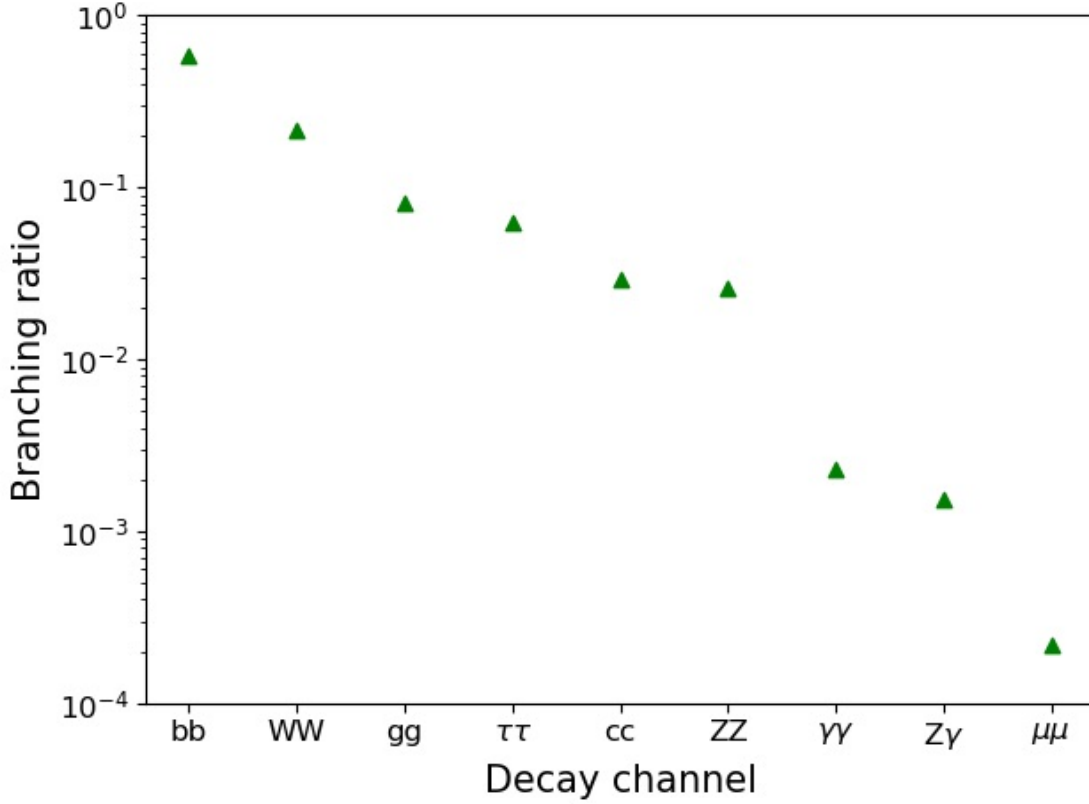


Figure 1.4: The SM prediction for the branching ratios of the different Higgs decay channels, assuming a Higgs boson mass of 125 GeV. The values are quoted from Ref. [1]

half of all decay occurrences. For a generic Higgs decay $h \rightarrow X + Y$, the condition $m_X + m_Y \leq m_h \approx 125$ GeV must be satisfied for energy conservation. The decay channels $h \rightarrow WW/ZZ$, even though $2m_{W/Z} > m_h$, are possible when one of the heavy vector bosons is off-shell, referring to a virtual particle with a mass different from its nominal value, as allowed by the uncertainty principle. Although the Higgs boson does not directly couple to massless particles, decay channels such as $h \rightarrow \gamma\gamma/gg$ are possible through indirect couplings in higher-order processes.

In 2012, the ATLAS [45] and CMS [46] Collaborations announced the observation of a new scalar boson at a mass of around 125 GeV [29–31] from proton-proton collision data at the LHC. Since then, more data have become available, allowing for a more precise characterization of the particle. So far, the observed boson is consistent with the Higgs

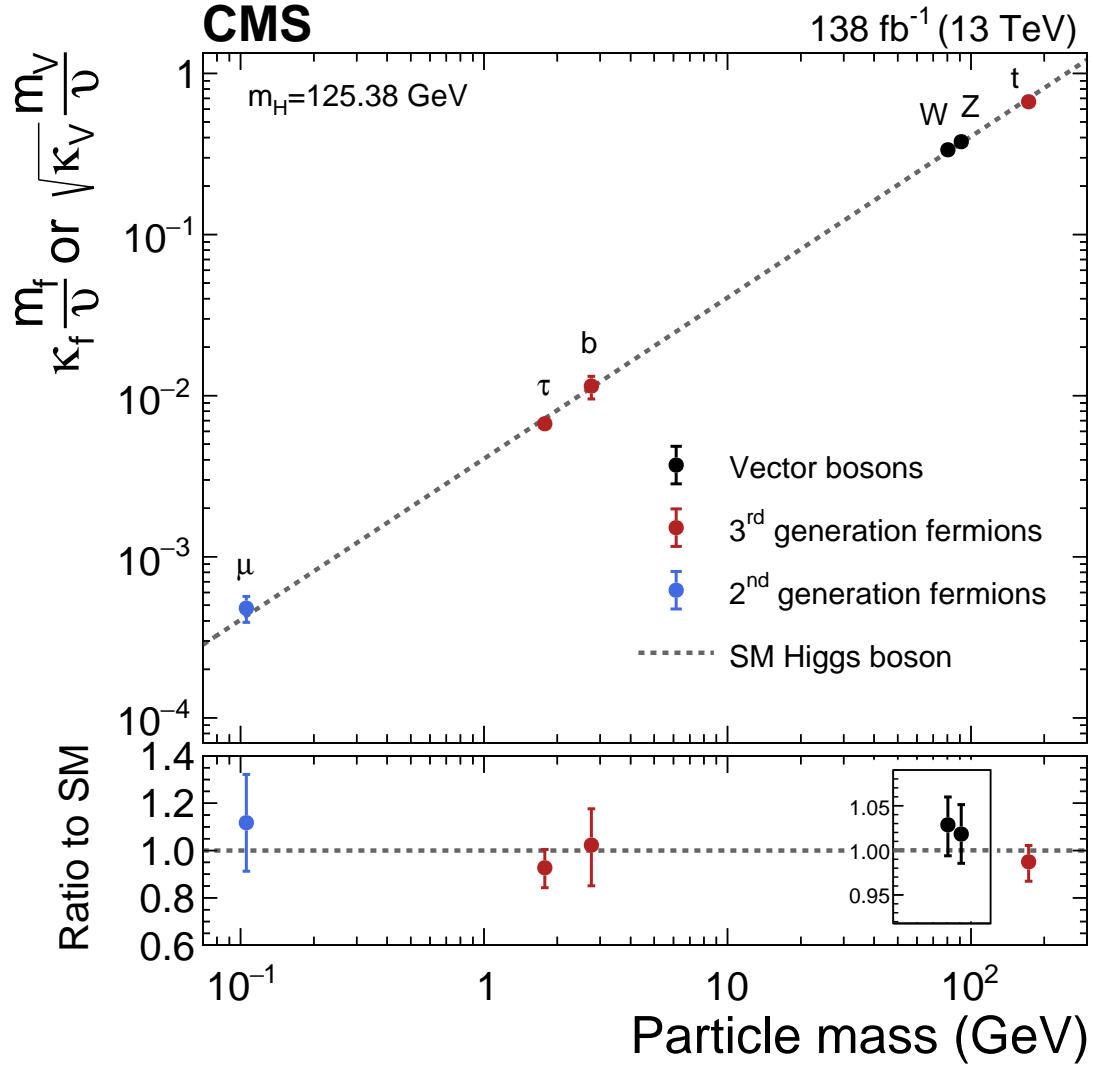


Figure 1.5: CMS measurements of the coupling modifiers of the Higgs boson to fermions and heavy gauge bosons, plotted as functions of particle mass, which are consistent with the SM prediction. The figure is taken from Ref. [2].

boson predicted by the SM within the experimental uncertainties [2, 42]. Fig. 1.5 shows the consistency of the CMS measurements with the theoretical prediction of the Higgs boson couplings with SM particles.

1.2 New physics beyond the Standard Model

1.2.1 The Standard Model is incomplete

Although the SM has been experimentally verified with unprecedented precision and accuracy, it is considered an effective field theory valid only up to a certain energy scale and is expected to fail beyond that point. There are several unresolved puzzles in the universe that the SM is not capable of addressing, motivating the exploration of new physics beyond the SM.

For instance, the SM accounts for only three of the four known fundamental forces, excluding the gravitational force at the quantum level. Incorporating gravity into the same theoretical framework as the other three forces results in a non-renormalizable quantum field theory [47], which predicts observable quantities with unrealistic infinite values.

Another missing piece is the absence of a candidate for dark matter in the SM. Astrophysical observations provide strong evidence for the existence of dark matter [48]. These observations indicate a new form of matter that is electromagnetically neutral and thus invisible, yet its existence is inferred from its substantial gravitational effects on large-scale objects such as galaxies. The nature of dark matter remains unknown today.

Arguably, the most profound problem related to the SM scalar sector is the huge energy gap between the electroweak scale and the Planck scale, known as the hierarchy problem [49]. The electroweak scale, set by the vev of the SM Higgs field, is measured to be $\nu \approx 246$ GeV. The Planck scale, on the order of 10^{19} GeV, refers to the energy scale at which the effects of quantum gravity become significant. Since the SM is an effective field theory valid up to a certain energy scale Λ , the Higgs mass squared receives leading-order quantum corrections through Yukawa coupling to the top quark (heaviest SM particle), which scales quadratically with Λ :

$$m_h^2 = m_{h,0}^2 + \Delta m_h^2, \quad \Delta m_h^2 \sim \mathcal{O}(\Lambda^2). \quad (1.22)$$

This shows that the Higgs mass is sensitive to new physics through Λ . Assuming that

the SM holds all the way up to the Planck scale, we have $\Lambda \sim 10^{19}$ GeV. This implies that extreme fine-tuning is necessary to almost perfectly offset this quadratic divergence, thereby justifying the surprisingly small observed Higgs mass compared to this scale. This issue refers to the naturalness of the Higgs mass and suggests the likelihood of the existence of new physics below the Planck scale.

Besides these issues, there are several more unresolved problems not mentioned here, all pointing to the fact that the SM is incomplete, prompting the formulation of BSM theories.

1.2.2 Motivations for exotic Higgs boson decays

The ATLAS and CMS experiments have put upper bounds of 12 and 16%, respectively, on the branching ratio of the Higgs boson to undetected particles at 95% confidence level, using the Run 2 data at $\sqrt{s} = 13$ TeV [2, 42]. This leaves room for exotic decays of the Higgs boson to BSM particles. Theoretically, extensions of the SM scalar sector are well-motivated, and exploring exotic Higgs decays can serve as a direct probe into these BSM theories [28, 50, 51].

First, the scalar sector is theoretically favorable for serving as a bridge connecting the hidden sector to the SM, known as the Higgs portal [52]. Hidden-sector matter, by definition, is not charged under the SM gauge group and thus does not have direct couplings with the SM fermions and gauge bosons. However, the SM scalar sector can be minimally extended to mediate interactions between SM particles and hidden-sector particles. For example, dark matter can be introduced into a theory through the Higgs portal [53].

Second, the scalar sector remains the least understood sector within the SM. The exact nature of the observed scalar boson—whether it belongs to the SM, a minimal framework to realize it, or a broader and more fundamental theory—remains an open question. Many BSM theories propose an extended scalar sector that can predict multiple scalar bosons, with the potential to adjust one to align with the SM predictions [28]. More importantly,

these theories offer potential solutions to some of the previously mentioned problems. Such BSM theories can manifest as exotic decays of the Higgs boson.

Additionally, the narrow decay width of the SM Higgs boson could potentially enhance the branching ratios for decays to BSM particles. The decay width, which is the sum of partial widths from all possible decay channels, measures the instability of a particle and is inversely proportional to its lifetime according to the uncertainty principle. The SM Higgs boson has a width of around 4 MeV [1], which is only 0.003% of its mass (125 GeV). This small Higgs width is mainly due to the small Yukawa couplings to fermions. For comparison, the W boson, with a mass of 80 GeV, has a width of 2 GeV [26]. The tiny width of the Higgs boson in the SM implies that even a small coupling of the Higgs boson to a new light BSM particle could lead to a significant enhancement of the total width, resulting in a non-negligible exotic decay channel.

1.2.3 Two-Higgs doublet models augmented by a scalar singlet

There exists an interesting class of exotic decays in which the Higgs boson decays into a pair of light pseudoscalar particles, which then decay into SM particles. Such decays are allowed in various theories, such as two-Higgs doublet models augmented by a scalar singlet (2HDM+S) [28, 50].

These models are well-motivated for several reasons. First, the scalar sector of the next-to-minimal supersymmetric extension of the SM (NMSSM) is a specific case of 2HDM+S, which, as a supersymmetric theory, offers a solution to the hierarchy problem [49]. In general, 2HDMs provide additional sources of charge-parity (CP) violation that can potentially explain the baryon asymmetry in the universe [54]. As one of the simplest possible extensions of the SM, these models have not yet been excluded by experimental data.

We first discuss the generic class of 2HDMs. In the SM, the scalar sector minimally consists of a complex scalar doublet, subject to the potential in Eq. 1.10. In 2HDMs, two doublets, Φ_1 and Φ_2 , are proposed, giving rise to a more complicated potential and richer phenomenology.

Under the assumptions, for simplification in most phenomenological studies, that CP is conserved in the scalar sector (allowing distinction between predicted scalars and pseudoscalars) and that there are no quartic scalar potential terms odd in either of the doublets due to a discrete Z_2 symmetry (preventing flavor-changing neutral currents at tree level), the most general scalar potential takes the form [28]:

$$V = m_{11}^2 \Phi_1^\dagger \Phi_1 + m_{22}^2 \Phi_2^\dagger \Phi_2 - m_{12}^2 (\Phi_1^\dagger \Phi_2 + \Phi_2^\dagger \Phi_1) + \frac{\lambda_1}{2} (\Phi_1^\dagger \Phi_1)^2 + \frac{\lambda_2}{2} (\Phi_2^\dagger \Phi_2)^2 + \lambda_3 \Phi_1^\dagger \Phi_1 \Phi_2^\dagger \Phi_2 + \lambda_4 \Phi_1^\dagger \Phi_2 \Phi_2^\dagger \Phi_1 + \frac{\lambda_5}{2} [(\Phi_1^\dagger \Phi_2)^2 + (\Phi_2^\dagger \Phi_1)^2], \quad (1.23)$$

where all parameters are real-valued.

Minimizing the potential, the two doublets attain their vevs:

$$\langle \Phi_1 \rangle_0 = \begin{pmatrix} 0 \\ \frac{\nu_1}{\sqrt{2}} \end{pmatrix}, \quad \langle \Phi_2 \rangle_0 = \begin{pmatrix} 0 \\ \frac{\nu_2}{\sqrt{2}} \end{pmatrix}, \quad (1.24)$$

where $v = \sqrt{\nu_1^2 + \nu_2^2} \approx 246$ GeV for electroweak symmetry breaking. Expanding around the vacuum state and substituting into the two-Higgs Lagrangian, there are eight degrees of freedom for the two complex scalar doublets.

Defining two important angle parameters: α and β . The ratio of the vevs can be parameterized by β :

$$\tan \beta = \frac{\nu_2}{\nu_1}. \quad (1.25)$$

The angle β diagonalizes the squared mass matrix of the charged scalars. One massless charged field is absorbed to give mass to the W^\pm boson, while the other one, with a non-zero mass eigenvalue, results in the physical charged scalar H^\pm . The angle β also diagonalizes the squared mass matrix of the pseudoscalars, where one of them is massless and is absorbed to give mass to the Z boson, while the massive one results in the physical pseudoscalar A . The angle α diagonalizes the squared mass matrix of the neutral scalars, resulting in two physical neutral scalars H and h . In total, there are five physical Higgs bosons: H^\pm , A , H , and h .

Table 1.3: Four coupling types of 2HDMs (and of 2HDM+S) that forbid flavor-changing neutral currents at tree level [27, 28].

	Type I	Type II	Type III	Type IV
Up-type quark	Φ_1	Φ_1	Φ_1	Φ_1
Down-type quark	Φ_1	Φ_2	Φ_1	Φ_2
Charged lepton	Φ_1	Φ_2	Φ_2	Φ_1

In a widely assumed scenario known as the decoupling limit [50], the lightest neutral scalar h is identified as the observed 125-GeV Higgs boson. In this scenario, all other Higgs bosons are much heavier than h , leading to SM-like couplings for h . This limit also corresponds to $(\beta - \alpha) \approx \pi/2$, where the h state tends to align with the full vev and is principally responsible for electroweak symmetry breaking.

In general, 2HDMs can predict scalar-mediated flavor-changing neutral currents at tree level, which are forbidden in the SM and strongly constrained by experimental data. These can be avoided by ensuring that fermions with the same quantum numbers couple only to one of the two doublets [27]. Four coupling types can satisfy this condition, as listed in Tab. 1.3.

Adding an additional complex scalar singlet as a next-order extension of 2HDMs, known as 2HDM+S, significantly enriches the phenomenology of exotic decays of the 125-GeV Higgs boson [50]. Considering that the parameter space for 2HDMs regarding these exotic decays is tightly constrained by current experimental data, 2HDM+S introduce a broader spectrum that remains unexplored.

The singlet has only small couplings to the doublets in the potential to preserve the SM-like couplings for h . This extension gives rise to two new physical states that are mostly singlet-like: a scalar (s) and a pseudoscalar (a). These states acquire their couplings to the SM particles only indirectly through mixing with the doublets. If the new states are light enough, exotic Higgs decays to these light states $h \rightarrow ss/aa/Za$, which then decay to SM particles, are possible.

In particular, we investigate $h \rightarrow aa$, which is the search performed for this thesis work and presented in Ch. 3. This decay channel has a simpler coupling structure and

therefore offers easier interpretation than a similar topology $h \rightarrow ss$. This is because the singlet-like pseudoscalar, a , has mixing with just one pseudoscalar from the doublets, while the singlet-like scalar, s , has mixing with two scalars from the doublets, introducing an additional degree of freedom from the extra mixing parameter. Therefore, the focus is typically on the pseudoscalar case rather than the scalar case when searching for such a decay topology.

This channel also allows probes of a wider mass range $m_a \leq m_h/2 \approx 62.6$ GeV compared to $h \rightarrow Za$, which can only probe $m_a \leq m_h - m_Z \approx 17$ GeV. Different types of 2HDM+S have different preferences for the final states in which the pseudoscalars decay, depending on the parameter $\tan \beta = \langle \Phi_2 \rangle_0 / \langle \Phi_1 \rangle_0$, as described below. Here, we assume m_a is above the bb -threshold so that $a \rightarrow bb$ is possible.

- **Type I: all fermions couple to the same doublet.** The branching ratios of the pseudoscalar to fermions do not depend on $\tan \beta$ and are proportional to those of the SM Higgs boson to fermions. In this case, decays to heavier fermions are always more favorable than lighter ones, when kinematically allowed.
- **Type II: up-type quarks couple to Φ_1 , while leptons and down-type quarks couple to Φ_2 .** The branching ratios to down-type quarks and leptons (up-type quarks) are enhanced (suppressed) when $\tan \beta > 1$, and vice versa. For example, $h \rightarrow aa \rightarrow bbbb/\tau\tau bb$ are the dominant channels for large $\tan \beta > 1$, while $a \rightarrow cc$ can be comparable to $a \rightarrow bb$ for $\tan \beta < 1$, even though c is lighter than b . Also, Type II of 2HDM+S is essentially the scalar sector of the NMSSM.
- **Type III: both up-type and down-type quarks couple to Φ_1 , while leptons couple to Φ_2 .** The branching ratios to leptons (up-type and down-type quarks) are enhanced (suppressed) when $\tan \beta > 1$, and vice versa. For example, $h \rightarrow aa \rightarrow \tau\tau\tau\tau$ is the dominant channel for large $\tan \beta > 1$, while $h \rightarrow aa \rightarrow bbbb$ is the dominant channel for $\tan \beta < 1$.
- **Type IV: up-type quarks and leptons couple to Φ_1 , while down-type**

quarks couple to Φ_2 . The branching ratios to down-type quarks (up-type quarks and leptons) are enhanced (suppressed) when $\tan \beta > 1$, and vice versa. For example, $h \rightarrow aa \rightarrow bbbb$ is the dominant channel for large $\tan \beta > 1$, while $a \rightarrow bb/cc/\tau\tau$ can be comparable for $\tan \beta < 1$.

Chapter 2

Experiment overview

This chapter provides a brief overview of the experimental setup. Sec. 2.1 introduces the Large Hadron Collider (LHC). Sec. 2.2 introduces the Compact Muon Solenoid (CMS) detector located on the LHC ring, describing its subdetectors that capture different aspects of collision events and the trigger system that handles initial event selections. Sec. 2.3 covers the reconstruction of physics objects relevant to the analysis in this thesis.

2.1 The Large Hadron Collider

The LHC [55] is the largest and highest-energy particle collider in the world. It features a 27-km synchrotron ring composed of superconducting magnets and is situated about 100 m underground on the Franco-Swiss border near Geneva, Switzerland. Built by CERN (the European Organization for Nuclear Research), the primary objectives of the LHC are to search for the Higgs boson and new physics beyond the Standard Model, such as supersymmetry and dark matter.

The performance of a particle collider is defined by three main factors: the type of particles it collides, the energy of the colliding beams, and the luminosity. The LHC is designed to collide protons at a maximum center-of-mass energy of 14 TeV with an instantaneous luminosity of up to $10^{34} \text{ cm}^{-2}\text{s}^{-1}$. Protons are chosen due to their stability and their ability to achieve the highest possible collision energy, as they lose less energy

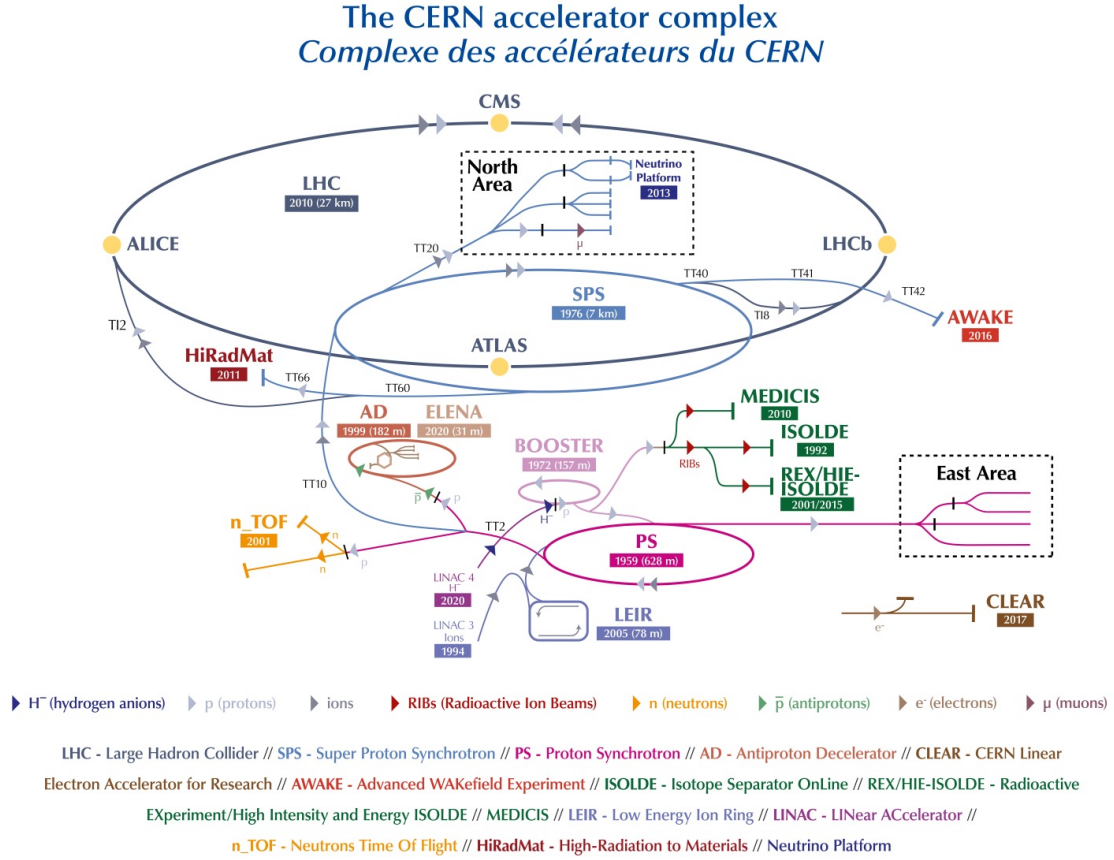


Figure 2.1: The CERN accelerator complex. The injection sequence goes as: Linac 4 \rightarrow PS Booster \rightarrow PS \rightarrow SPS \rightarrow LHC. The figure is taken from Ref. [3].

through synchrotron radiation in magnetic fields compared to lighter subatomic particles. The TeV energy scale is crucial for testing electroweak physics and searching for new phenomena predicted at this energy level.

The CERN accelerator complex includes several machines that sequentially increase the energy of a proton beam, with the LHC being the final stage, as shown in Fig. 2.1. Initially, protons are generated by stripping electrons from hydrogen ions using electric fields and then accelerated to 160 MeV in the Linear Accelerator 4 (Linac 4). The proton beam is then sequentially injected into the Proton Synchrotron Booster (PSB), the Proton Synchrotron (PS), and the Super Proton Synchrotron (SPS), where it is accelerated to 2, 26, and 450 GeV, respectively.

After these stages, the protons are divided and injected into two parallel beam pipes in the LHC, where they circulate in opposite directions. It takes about 4 minutes to fill each beam pipe and an additional 20 minutes to accelerate the protons to their maximum energy. During Run 2, each proton beam reached an energy of 6.5 TeV, roughly 7000 times the proton's rest mass, traveling at 99.999999% the speed of light and orbiting the 27-km LHC ring at a frequency of 11 kHz. The two beams are then directed to collide at four interaction points along the LHC ring.

The proton beam consists of bunches rather than a continuous stream, allowing interactions to be separated from each other at discrete intervals (25 ns, or 40 MHz). The machine is expected to produce N events:

$$N = \sigma \mathcal{L}, \quad (2.1)$$

where σ is the cross section for a specific interaction or a collection of interactions, and $\mathcal{L} = \int L(t)dt$ represents the integrated luminosity over a period of time.

The instantaneous luminosity is characterized by several machine factors:

$$L = \frac{N_b^2 n_b f_{\text{rev}} \gamma}{4\pi \epsilon_n \beta^*} F. \quad (2.2)$$

Here, $N_b \approx 10^{11}$ is the number of protons per bunch, $n_b = 2808$ is the number of bunches per beam, $f_{\text{rev}} \approx 11$ kHz is the revolution frequency, γ is the relativistic Lorentz factor, ϵ_n is the normalized transverse beam emittance, β^* is the beam beta function at the collision point, and F is the reduction factor due to the crossing angle at the collision point. Together, these factors result in the nominal value of $L = 10^{34} \text{ cm}^{-2}\text{s}^{-1}$.

The CMS measurements of the SM cross sections are shown in Fig. 2.2. The LHC is scheduled to collect data with an integrated luminosity of 450 fb^{-1} by the end of Phase-1, it will then upgrade to the Phase-2 high-luminosity LHC (HL-LHC), where 3000 fb^{-1} of data will be collected by its end, as illustrated in Fig. 2.3.

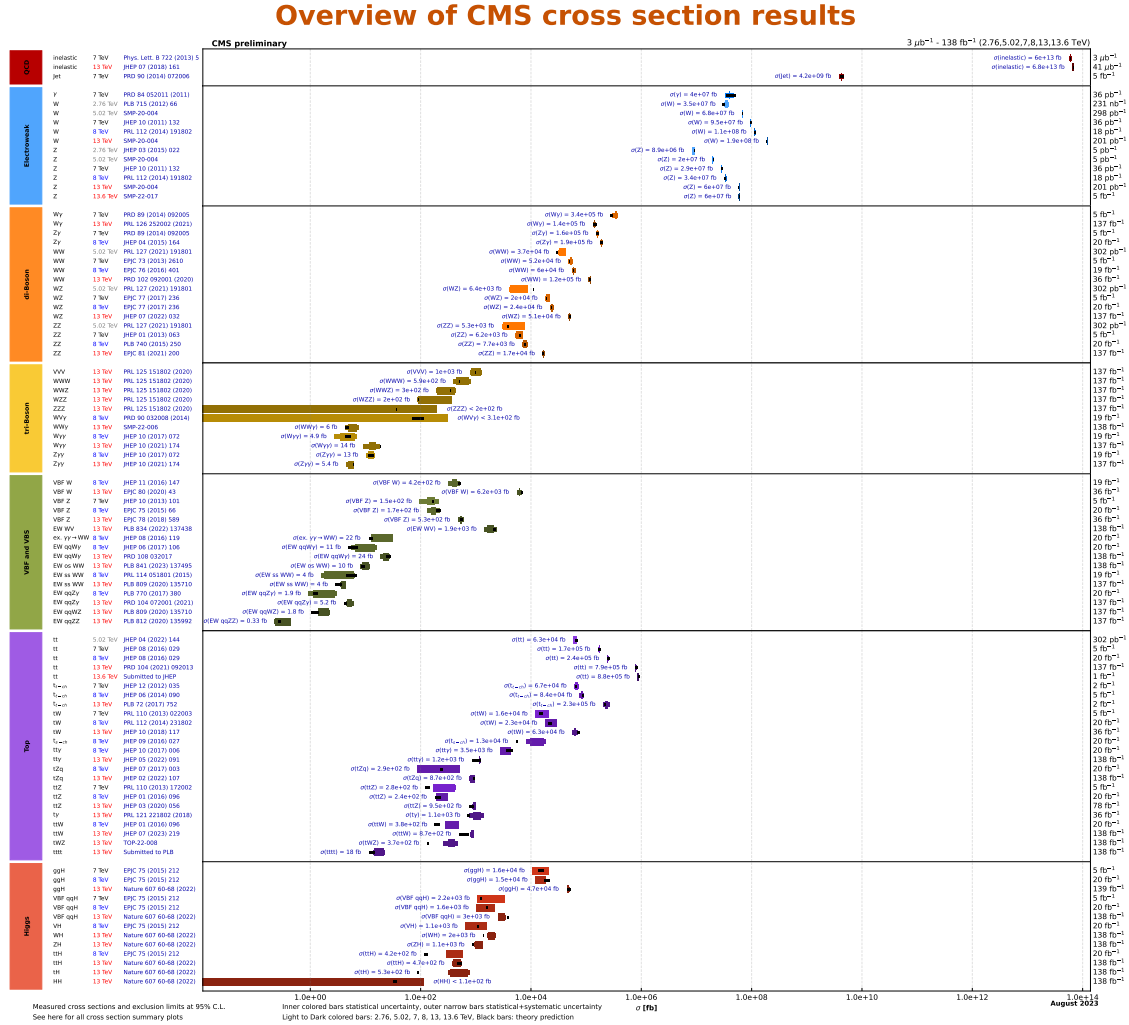


Figure 2.2: Summary of CMS measurements of the SM cross sections. The figure is taken from Ref. [4].

2.2 The Compact Muon Solenoid detector

The LHC ring hosts four detectors where collisions take place, with CMS [6, 46] being one of them. CMS is a general-purpose detector capable of a wide variety of missions, including studying the nature of electroweak symmetry breaking, precision measurements of elementary particles, and searching for new physics beyond the SM. When a collision occurs, many particles are produced, some stable and some unstable. The unstable particles quickly decay into stable particles that can be detected. Essentially, CMS acts as a high-speed camera, capturing snapshots of these particles from collisions every 25 ns.



Figure 2.3: Data taking schedule of the LHC and the HL-LHC. The figure is taken from Ref. [5].

A schematic view of CMS is shown in Fig. 2.4. CMS has a compact cylindrical shape, measuring 15 m in diameter and 28.7 m in length, and weighs 14,000 tons. CMS adopts a coordinate system with the origin at the nominal collision point. The x -axis points radially inward toward the center of the LHC, the y -axis points vertically upward, and the z -axis points along the beam direction in a right-handed manner.

Since the detector is cylindrical, the azimuthal angle ϕ is defined in the x - y plane and measured from the x -axis, while the polar angle θ is measured from the z -axis. Since the rest frame of the detector is not necessarily the rest frame of the colliding particles, quantities invariant under Lorentz boosts along the beam direction are defined. These include the transverse component of momentum (p_T), energy (E_T), and missing energy from undetected particles (MET). Additionally, pseudorapidity (η) is defined to replace the polar angle:

$$\eta = -\ln \tan(\theta/2), \quad (2.3)$$

where the difference in η is Lorentz invariant along the beam direction.

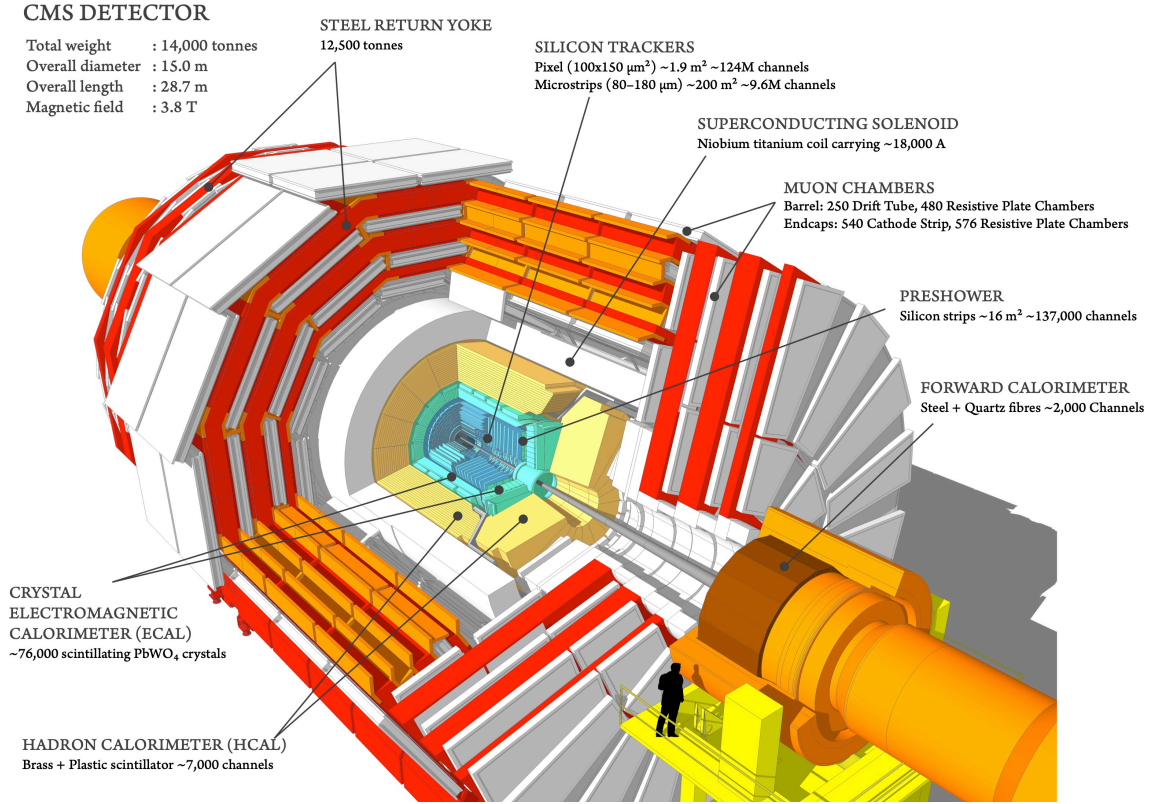


Figure 2.4: Schematic view of the CMS detector. The figure is taken from Ref. [6].

CMS consists of several concentric layers of subdetectors that capture different properties of the particles produced in a collision. First, it has a superconducting solenoid magnet with 6 m internal diameter, generating a 3.8 T magnetic field to bend the trajectories of charged particles. The charge of a charged particle can be determined by the bending direction in the magnetic field. The transverse momentum of a charged particle can be inferred from the curvature of the trajectory: higher momentum results in smaller curvature.

The innermost layer within the magnet is a tracker that records these trajectories traced by charged particles, as described in Sec. 2.2.1. Outside the inner tracker are the calorimeters that record the energy of various particles, as described in Sec. 2.2.2 and Sec. 2.2.3. Outside the magnet, the muon system is dedicated to detecting muons, as described in Sec. 2.2.4. There is a two-tier trigger system that promptly filters and processes collision data, as described in Sec. 2.2.5.

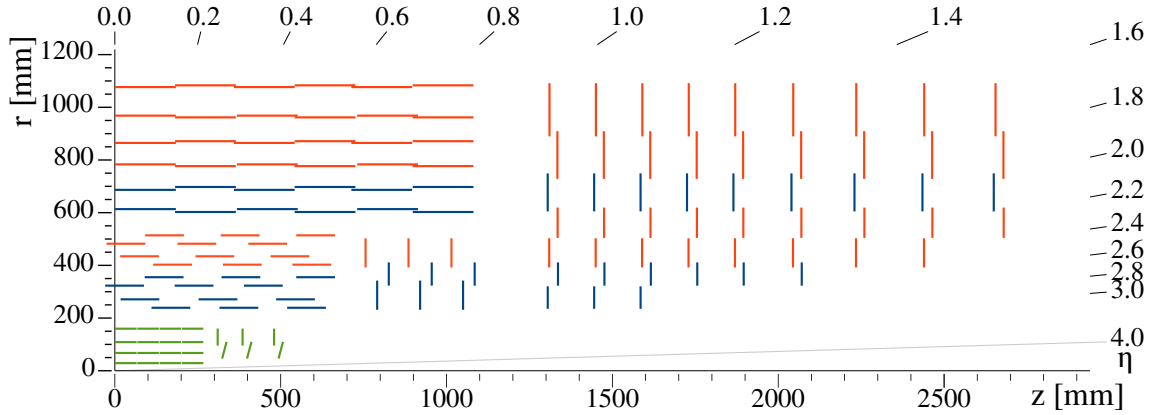


Figure 2.5: Schematic view of the Phase-1 CMS tracker, showing one quadrant in the $r-z$ plane. The pixel detector is shown in green, while the strip detector is shown in red and blue. The figure is taken from Ref. [7].

2.2.1 Tracker

The innermost part of CMS, closest to the collision point, is the tracking system [6, 46]. The tracker is designed to provide high-precision measurements of the trajectories of charged particles and to reconstruct primary vertices for promptly produced particles and secondary vertices for long-lived particles. When charged particles enter the tracker and interact electromagnetically with the silicon sensors, signals are generated and recorded as hits. These hits are then used as input for track reconstruction algorithms to identify particle tracks.

Positioned closest to the collision point, the tracker aims to capture as many particles from a collision as possible, providing more accurate and complete track measurements and enhancing the reconstruction of interaction vertices. However, each collision produces a large number of particles, averaging around 1000 particles per bunch crossing under design conditions, and occurs every 25 ns. This requires the tracker to have high spatial granularity and fast response time to resolve the tracks and correctly assign them to the bunch crossing. Additionally, a high radiation tolerance is required to withstand exposure to a high particle flux for a long period of time. Therefore, the tracker is made entirely of silicon to handle these challenging operating conditions.

The tracker consists of a pixel detector and a strip detector, as shown in Fig. 2.5.

The pixel detector, positioned closest to the collision point and before the strip detector, primarily provides resolution for measuring the impact parameter. It has a configuration of four barrel layers and three forward disks on each end to provide a four-hit coverage for tracks, with a total of 124 million readout channels.

The strip detector has a longitudinal length of 5 m and a diameter of 2.5 m, consisting of ten barrel layers and nine forward disks on each end. There are 9.3 million silicon microstrips in total, providing a hit resolution of 20 μm for perpendicular crossing.

In summary, the tracker can detect charged particles with $p_{\text{T}} > 50$ MeV within $|\eta| < 2.5$, providing a p_{T} resolution of around 1% and an impact parameter resolution of around 10 μm for charged particles at $p_{\text{T}} = 100$ GeV in the central region of the detector.

2.2.2 Electromagnetic calorimeter

The next layer is the electromagnetic calorimeter (ECAL) [6, 8, 46], which is designed specifically to measure the energy of electrons (e) and photons (γ).

A calorimeter is a detector that measures the kinetic energy of particles. When a high-energy particle enters the calorimeter, it interacts with the detector materials, which are chosen for their high absorbing power. These interactions induce the production of secondary particles, resulting in a cascade shower until the particles lose most of their kinetic energy. The energy deposited by these secondary particles is converted into signals that can be collected and used to infer the energy of the incident particle.

A schematic view of the ECAL is shown in Fig. 2.6. The ECAL consists of a barrel and two endcap sections, made up of 75848 lead tungstate (PbWO_4) crystals. When a high-energy electron enters the ECAL crystals, it interacts electromagnetically with the material, primarily losing energy through bremsstrahlung ($e \rightarrow e\gamma$), radiating high-energy photons. These photons then undergo pair production, creating electron-positron pairs ($\gamma \rightarrow e^+e^-$). These processes repeat, creating a cascade of decays where more particles are produced with decreasing energy until the secondary particles have a mean energy below the process thresholds. The PbWO_4 crystal is chosen for its ability to

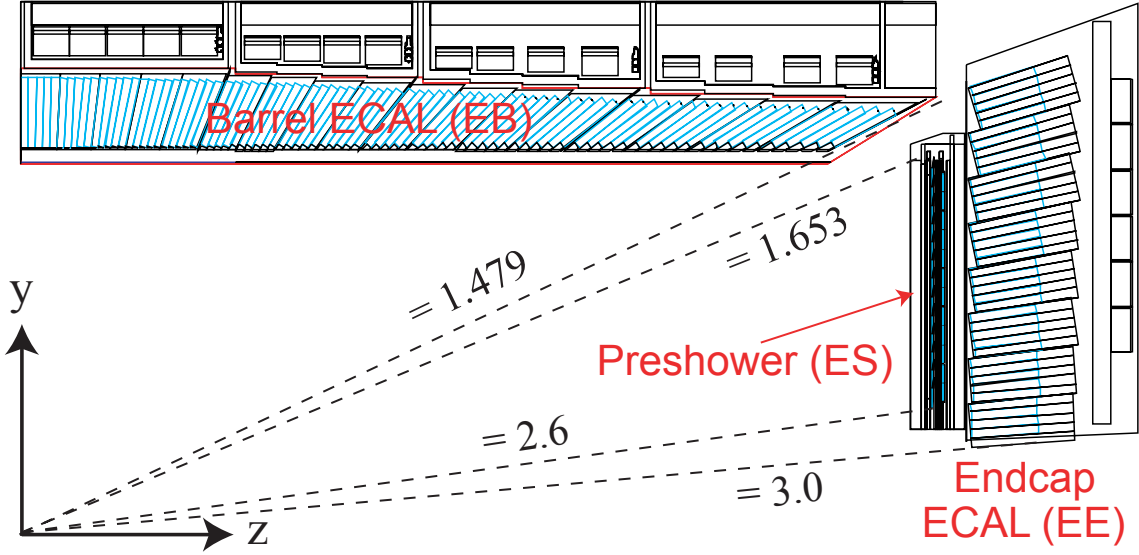


Figure 2.6: Schematic view of the CMS ECAL, showing one quadrant in the $r - z$ plane. The figure is taken from Ref. [8].

effectively contain these electromagnetic showers. Each PbWO_4 crystal has a length of 23 cm, corresponding to around 25 radiation lengths, where one radiation length is defined as the mean length traveled by an electron such that its energy is reduced by a factor of e^1 through electromagnetic interactions with the material.

The ECAL has a barrel energy resolution for electrons [56]:

$$\frac{\sigma}{E} = \frac{2.8\%}{\sqrt{E/\text{GeV}}} \oplus \frac{12\%}{E/\text{GeV}} \oplus 0.3\% \quad (2.4)$$

where the three sources correspond to the stochastic, noise, and constant terms, respectively.

2.2.3 Hadron calorimeter

Surrounding the ECAL is the hadron calorimeter (HCAL) [6, 9, 46], which is designed to measure the energy of hadrons—particles made of quarks and gluons. This is crucial for reconstructing hadronic jets and MET. When high-energy hadrons enter the HCAL, they interact strongly with nuclei in the medium, producing secondary particles that interact further downstream, resulting in hadronic showers.

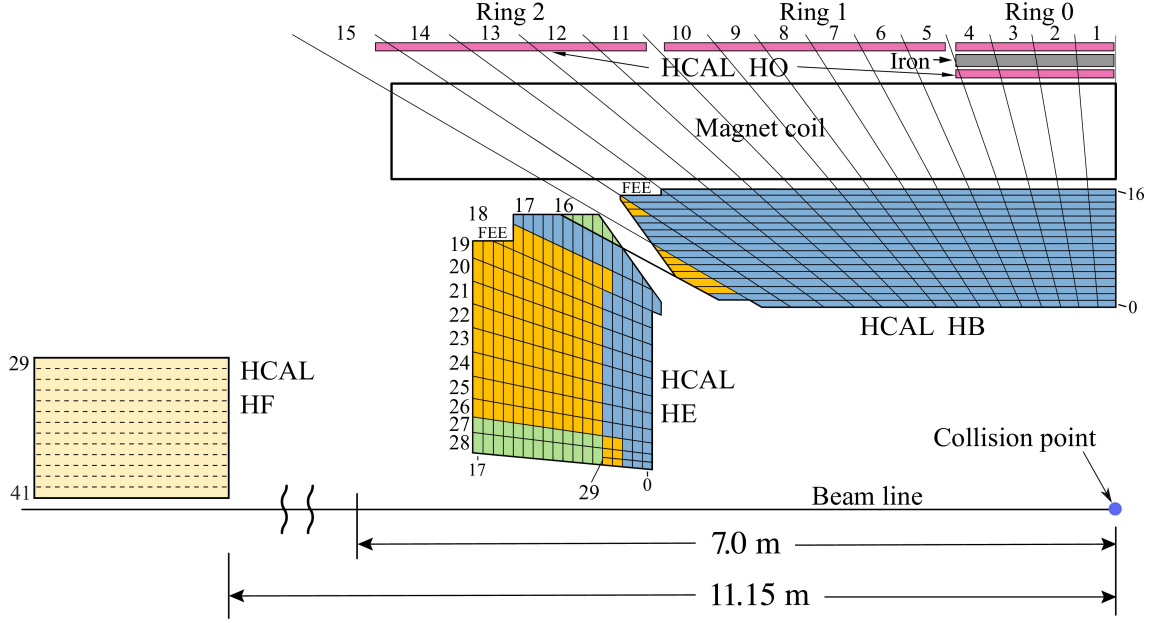


Figure 2.7: Schematic view of the CMS HCAL during Run 2 operation, showing one quadrant in the $r - z$ plane. The figure is taken from Ref. [9].

Hadronic showers typically spread wider and over a longer distance than electromagnetic showers due to more complex hadronic interactions and a longer nuclear interaction length, which is defined as the mean free path for a hadron traversing the medium before undergoing a nuclear interaction. This means that a larger HCAL volume is required to allow hadronic showers to fully develop and be contained. Therefore, the HCAL is positioned farther away from the collision point than the ECAL to ensure that the more compact electromagnetic showers are contained first and do not mix with the larger hadronic showers.

The HCAL consists of four subdetectors optimized for different η ranges: the hadron barrel (HB) for $|\eta| < 1.39$, the hadron endcap (HE) for $1.3 < |\eta| < 3$, the hadron outer (HO) for $|\eta| < 1.26$, and the hadron forward (HF) for $3 < |\eta| < 5.2$, as shown in Fig. 2.7. The HB and HE consist of alternating layers of brass absorber plates and scintillator tiles. The HO is located outside the magnet coil and provides additional coverage for the tails of hadronic showers when the ECAL barrel and HB cannot fully contain them. The HF consists of steel absorbers with quartz fibers to withstand the harsh environment in the

high pseudorapidity regions, where the particle flux is highest.

The combined ECAL and HCAL in the barrel and endcap regions provide an energy resolution for charged pions [6]:

$$\frac{\sigma}{E} = \frac{84.7\%}{\sqrt{E/\text{GeV}}} \oplus 7.6\%. \quad (2.5)$$

2.2.4 Muon system

The last layer is the muon system [6, 10, 46], a dedicated detector that provides precise muon measurements.

High-energy electrons are stopped in ECAL through bremsstrahlung, which occurs at a rate inversely proportional to the square of the mass. Muons, which are heavier, have a bremsstrahlung rate suppressed by a factor of $(m_e/m_\mu)^2 \approx 10^{-5}$ compared to electrons. For muons at the GeV scale, energy loss occurs primarily through ionization. As a result, muons are not stopped by calorimeters and can escape the detector, necessitating the muon system in the outermost layer to trace their trajectories. The main feature of muons in the detector is their long trail of ionization: they produce tracks in the tracker, deposit small fractions of energy in calorimeters, and produce tracks in the muon system.

The muon system is positioned outside the solenoid magnet, consisting of three types of gas ionization chambers as of Run 2: drift tubes (DTs) for $|\eta| < 1.2$, cathode strip chambers (CSCs) for $0.9 < |\eta| < 2.4$, and resistive plate chambers (RPCs) for $|\eta| < 1.9$, as shown in Fig. 2.8. The muon p_T measurement is improved when combined with information from the tracker, achieving a p_T resolution of 1% in the barrel and 3% in the endcaps for p_T up to around 100 GeV [10].

2.2.5 Trigger system

Due to the extremely high data rate of proton-proton collisions at the LHC, where the beams cross every 25 ns (40 MHz), producing $\mathcal{O}(100)$ terabytes of data per second, it is impossible to process and store all data. Therefore, a trigger system is used to reduce

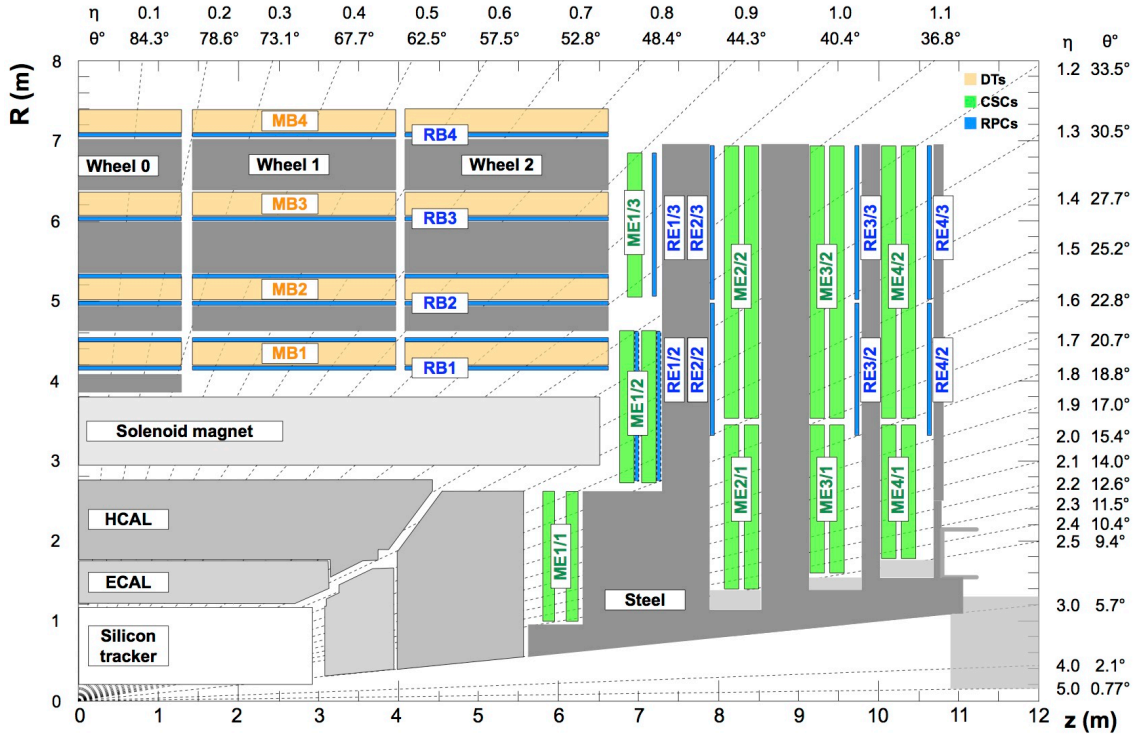


Figure 2.8: Schematic view of the CMS detector during Run 2 operation, showing one quadrant in the $r - z$ plane, with the different muon stations indicated. The figure is taken from Ref. [10].

the rate in real time to a manageable level in two steps: the Level-1 (L1) trigger and the High-Level trigger (HLT) [6, 11, 46]. A trigger decision determines whether to store a given event immediately after the collision for further analysis, based on a set of event selection criteria related to the physics signatures presented in an event.

The L1 trigger is implemented in custom hardware processors and is designed to reduce the input rate from 40 MHz down to 100 kHz. The trigger decision at L1 is based on coarse-grained trigger-level information to maintain an overall latency below $4 \mu\text{s}$, fixed by the event processing time and the capacity of the buffer system. The architecture consists of several trigger subsystems, as shown in Fig. 2.9.

The calorimeter trigger receives inputs of local energy deposits from the calorimeters, calibrates them in Layer 1, and then reconstructs from them to form physics objects such as electrons, photons, tau leptons, jets, and energy sums in Layer 2. The muon trigger

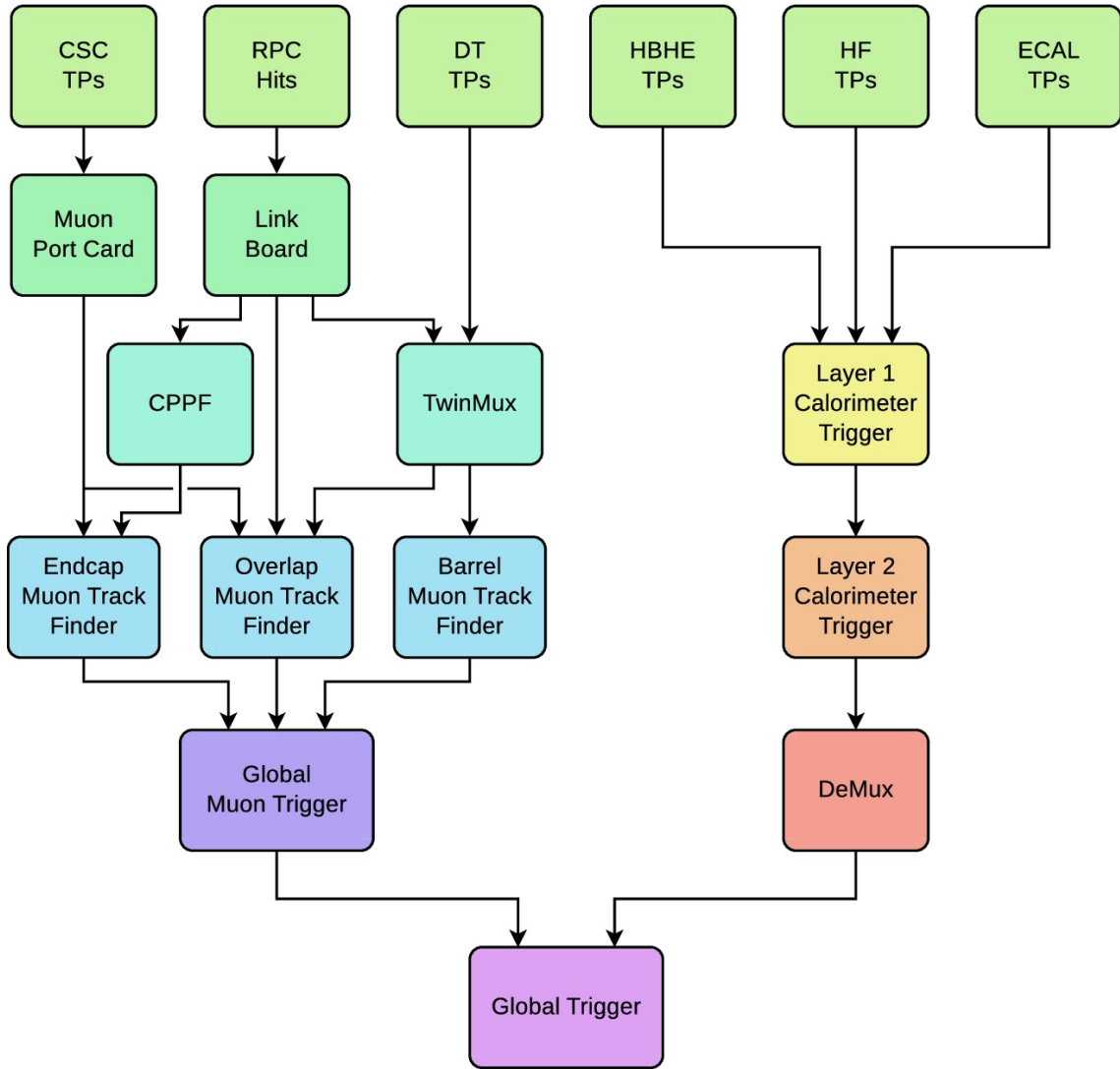


Figure 2.9: The CMS L1 trigger during Run 2 operation, with each trigger subsystem indicated. The figure is taken from Ref. [11].

receives inputs of muon information from the three muon stations, each with muon track finders for muon reconstruction in the respective η range. These are then combined in the global muon trigger for final muon selection. The global trigger combines calorimeter objects and muons to make the final trigger decision. Ensuring the quality of the selection criteria in L1 is essential since it performs the first round of event selection at the LHC, and events that are not triggered are permanently discarded.

The HLT is implemented in software running on CPUs and GPUs from a farm of

commercial processors and is designed to reduce the rate from L1 of 100 kHz to 1 kHz for permanent storage for offline analysis. The HLT has access to the full event information and can reconstruct finer-grained physics objects to apply more sophisticated selection criteria than L1.

The HLT processes and filters data in parallel sequences based on specific physics requirements, called HLT paths. These HLT paths consist of sequences of object reconstruction and selections that progress from simpler to more complex along the sequence. These HLT paths are then grouped into datasets that share similar physics objects, such as single muon and di-tau datasets, for offline analysis.

2.3 Reconstruction and identification of physics objects

This section discusses the reconstruction and identification of physics objects in CMS. The idea of the particle-flow (PF) algorithm is discussed in the following. The specific objects most relevant to the analysis presented in Ch. 3 include electrons (Sec. 2.3.1), muons (Sec. 2.3.2), hadronic tau (Sec. 2.3.4), and b jets (Sec. 2.3.3).

The PF algorithm [12] aims to construct a global event description by integrating information from all subdetectors to optimally reconstruct and identify particles produced in collision events. Five individual particles, called PF candidates, are first reconstructed using their unique signature produced in the detector, as shown in Fig. 2.10 and described in the following.

- **Photons** are neutral and do not produce tracks in the tracker. They enter the ECAL, create electromagnetic showers, and are contained within the ECAL. These showers result in deposits of energy in clusters of neighboring ECAL cells. The corresponding ECAL clusters are not matched to any tracks in the tracker.
- **Electrons** are charged and produce tracks in the tracker. Similar to photons, they create electromagnetic showers and are contained within the ECAL. The electron tracks can be extrapolated to the corresponding ECAL clusters to form electron

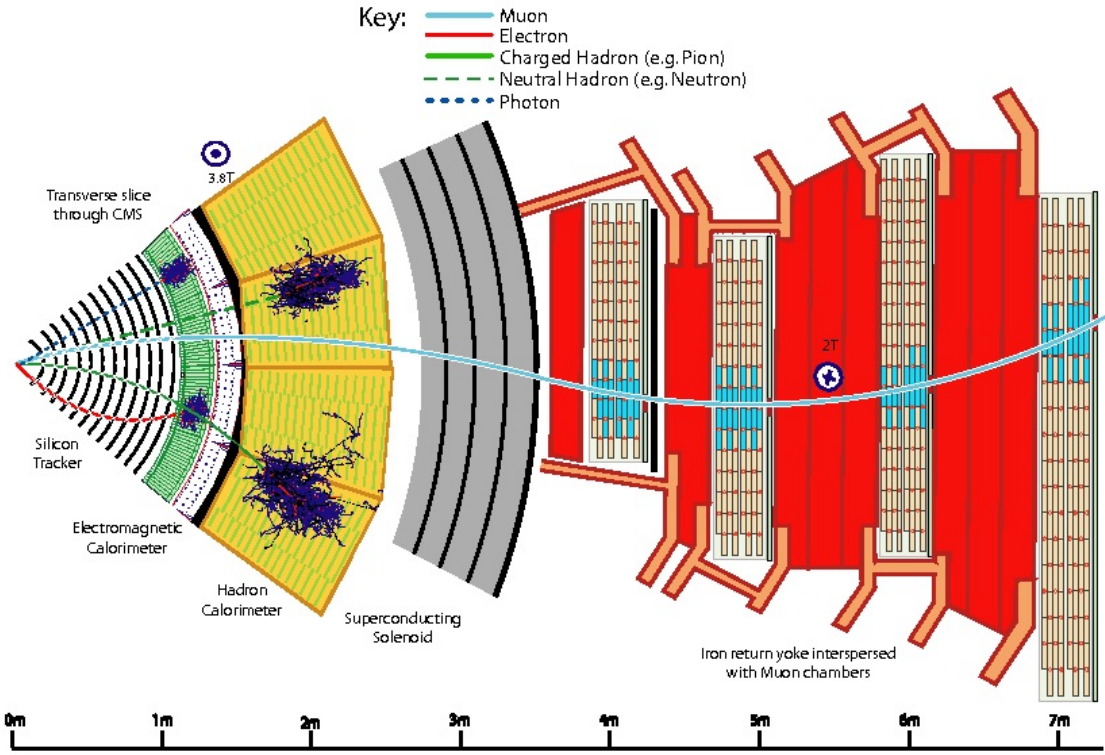


Figure 2.10: Schematic view in a transverse slice of CMS, showing the signature of each of the five PF candidates. The figure is taken from Ref. [12].

trajectory. Their energy is determined by combining the information of the track momentum and the ECAL energy clusters.

- **Muons** produce tracks in the tracker, traverse the calorimeters leaving almost no trails, and produce hits in the muon chambers before escaping the detector. They lose energy primarily through ionization, resulting in relatively clear trajectories with long trails across the detector. Their energy is determined from the curvature of the tracks.
- **Charged hadrons** produce tracks in the tracker, traverse the ECAL, create hadronic showers in the HCAL, and are contained within the HCAL. The hadronic shower shape is typically wider and longer than electromagnetic showers due to the more complex nuclear interactions and a longer nuclear interaction length. Their trajec-

tories are determined by extrapolating the tracker tracks to the compatible HCAL clusters. Their energy is measured by combining the information of the track momentum and the ECAL and HCAL energy clusters.

- **Neutral hadrons** are reconstructed in a similar way to charged hadrons, except that they are not matched to any tracks in the tracker.

These PF candidates are then used for downstream reconstruction and identification algorithms to further improve the respective performance and to form higher-level objects, such as missing transverse energy, hadronic taus, and jets with flavor tagging.

2.3.1 Electrons

Electrons are reconstructed using information from the tracker and the ECAL and identified using a multivariate technique [57].

The first step of electron reconstruction is to find ECAL clusters. A cluster is formed by grouping neighboring cells that have energy higher than a certain threshold, typically a few times higher than the expected electronic noise. Clusters with a local maximum energy greater than 1 GeV are used to seed the next steps, assuming they correspond to an incident particle. Since electrons originating from the primary vertex may undergo showering in the tracker before reaching the ECAL, there could be multiple secondary electrons and photons from bremsstrahlung and pair production surrounding the primary electrons, thus producing multiple clusters around the primary one. Therefore, these neighboring ECAL clusters are combined to form superclusters that include all the possible energy deposits from the primary electrons and their radiations.

Since electrons lose energy by radiating bremsstrahlung photons along the way, which can perturb the trajectories, a dedicated tracking algorithm is needed to take this factor into account when reconstructing electron tracks and estimating the track parameters. The electron tracks are then combined with the ECAL superclusters to form electron candidates based on the kinematics and quality of the tracks, the shape of the superclusters, and the compatibility between the tracks and the superclusters. The superclusters are

also refined by using tracking information to include additional clusters from photon conversion and bremsstrahlung that were missed when forming superclusters using only the ECAL information. The electron reconstruction efficiency, defined as the ratio between the number of reconstructed superclusters matched to reconstructed electrons and the total number of reconstructed superclusters, is measured from the data to be above 95% for $p_T > 10$ GeV using $Z \rightarrow ee$ events.

For electrons promptly produced at the primary vertex, various background sources can lead to misidentification. These include secondary electrons from b or c quark decays, hadrons with high ECAL deposits that fake electrons, and electrons originating from photon conversions. A multivariate discriminant is constructed to identify electrons. The discriminating variables are mainly based on the sums of isolation energy around the direction of the electron, the geometric shape of the ECAL shower, the quality and hit pattern of the electron track, and the compatibility between the track and the ECAL clusters. Three working points, corresponding to three threshold cuts in the multivariate discriminant, are defined with identification efficiencies of 70, 80, and 90%, respectively.

2.3.2 Muons

Muons are reconstructed using information from the tracker and the muon system and identified based on sequential selection requirements [10].

Muons are essentially ionization tracks without complex patterns as other particles. The first step involves the local reconstruction of hits of the muon trajectory in muon chambers. When muons traverse a muon chamber, they ionize gas molecules in the cells to produce free electrons along their trajectories. These free electrons are then guided by the applied electric field to arrive at the cell boundary, producing electric signals that encode the positions and time arrivals of the muon trajectory within the cell. These measurements are calibrated in each of the three muon subdetectors for local reconstruction of the muon hits. Furthermore, each of the CSC and DT chambers consists of multiple layers, and the muon hits are interpolated to form straight-line segments to be used as inputs for muon

track reconstruction.

The reconstruction of muon tracks is based on the tracks independently reconstructed in the muon system (standalone-muon tracks) and in the tracker, respectively. There are three types of reconstructed muons.

- **Tracker muons** are reconstructed starting from the tracker tracks. These tracks are extrapolated outward to the muon system with the loose requirement that at least one CSC or DT segment is matched to the extrapolated track. This loose requirement is efficient in covering low- p_T muons as they typically penetrate through fewer muon chambers. However, this can also lead to a higher fake rate and lower momentum resolution.
- **Global muons** are reconstructed starting from the standalone-muon tracks. These tracks are extrapolated inward to match compatible tracker tracks. In contrast to tracker muons, additionally requiring the presence of standalone-muon tracks leads to higher efficiency for high- p_T muons and better momentum resolution.
- **Standalone muons** are reconstructed only using standalone-muon tracks without compatible reconstructed tracker tracks. They are mainly cosmic muons which enter the muon system from outside the detector and are not energetic enough to penetrate to the inner tracker.

These reconstructed muons are then identified into different categories based on a set of selection criteria on various variables. The main variables include the quality of the tracks, the compatibility in the track matching, and the compatibility with the primary vertex. The so-called medium muon identification is a working point defined to optimally select prompt muons and muons from heavy flavor decays, with an overall efficiency of 99.5% for identifying muons from simulated W/Z decays. A tighter working point, called tight muon identification, is defined to further suppress muon misidentification and has efficiencies ranging from 95 and 99%.

2.3.3 Jets and b-tagging

Quarks and gluons can be produced from high-energy proton-proton collisions. Due to QCD confinement, these quarks and gluons cannot exist freely and will undergo parton showering and hadronization, in which hadrons are formed out of them. These hadrons tend to move closely along the same direction as the original quark or gluon, resulting in a collimated shower of particles called a jet. Individual particles of the jet can be grouped to reconstruct the jet. Identifying the original particle of the jet is called jet tagging.

Jets are reconstructed by clustering PF candidates with the anti- k_T algorithm [58, 59]. The algorithm iteratively combines particles into a jet using the distance measures:

$$d_{ij} = \min(p_{T,i}^{-2}, p_{T,j}^{-2}) \frac{\Delta R_{ij}^2}{R^2}, \quad d_{iB} = p_{T,i}^{-2}. \quad (2.6)$$

Here, d_{ij} is the distance between the i -particle and the j -particle, which have transverse momenta $p_{T,i}$ and $p_{T,j}$, respectively. $\Delta R_{ij}^2 = (\eta_i - \eta_j)^2 + (\phi_i - \phi_j)^2$ is the angular separation, which is Lorentz invariant along the beam direction. The parameter R defines the size of the jet cone for the clustering. d_{iB} is the distance between the i -particle and the beam. The power of -2 to the momenta ensures that the jets are preferentially seeded from the most energetic particles. During Run 2, CMS set $R = 0.4$ as the standard jet size, called AK4 jets.

The iteration proceeds as follows. The first step is to collect the list of particles in the event and compute the distances in Eq. 2.6 for all particles. Then, find the smallest value among all d_{ij} and d_{iB} . If the smallest value is of type d_{ij} , a new particle is formed by combining the i -particle and the j -particle by summing their 4-momenta; the i -particle and the j -particle are then removed from the list, and the new particle is added to the list. If the smallest value is of type d_{iB} , then the i -particle is considered a jet and removed from the list. After updating the list of particles, the distances are recalculated. These steps are repeated until the list is exhausted.

Since protons are bunched in a beam, many collisions occur simultaneously in a single

bunch crossing. For each bunch crossing, the collision with the highest $\sum p_T^2$, where the sum runs over the tracks of the primary vertex, is identified as the event of interest, since it potentially contains the hardest scattering process. Other primary vertices are considered as additional background collisions, called pileup.

Particles from the pileup vertices can overlap with particles from the vertex of interest and confuse the jet reconstruction. The number of pileup vertices in Run 2 was on average 23 in the 2016 data-taking period, and 32 in 2017 and 2018 [60]. These pileup effects are mitigated using the standard technique called charged-hadron subtraction (CHS) [12], where charged particles with reconstructed tracks associated with pileup vertices are identified and excluded before the jet clustering step. Neutral particles from pileup are treated by applying an event-by-event energy correction to the reconstructed jets.

The reconstructed jet energy is successively corrected for several factors [61]. First, there is a pileup offset correction, which estimates and subtracts the additional energy incorrectly included in the jet reconstruction from pileup. This term helps to remove pileup contributions from photons and neutral hadrons after the CHS. Then, corrections are applied to adjust the non-uniformity of the jet energy response and remove its dependence on the jet p_T and η due to detector effects. The residual difference between data and simulation is corrected.

Jets originating from b quarks are called b jets. B jets contain b hadrons from the hadronization of b quarks. The lifetime of b hadrons is on the order of $\tau \sim 1.5$ ps or $c\tau \sim 0.45$ mm, which is relatively longer than that of light-flavor hadrons. This leads to displaced tracks originating from a secondary vertex (SV) due to the decay of b hadrons, as illustrated in Fig. 2.11. Therefore, reconstruction of secondary vertices is crucial in b-tagging.

A jet flavor classification algorithm called **DeepJet** is used to tag b jets [13,62]. **DeepJet** is a neural network trained to classify the flavor of a given jet. The input features to the neural network include properties of the SV and all constituent particles of the jet, as well as jet-level information such as the jet kinematics and the number of tracks. Three

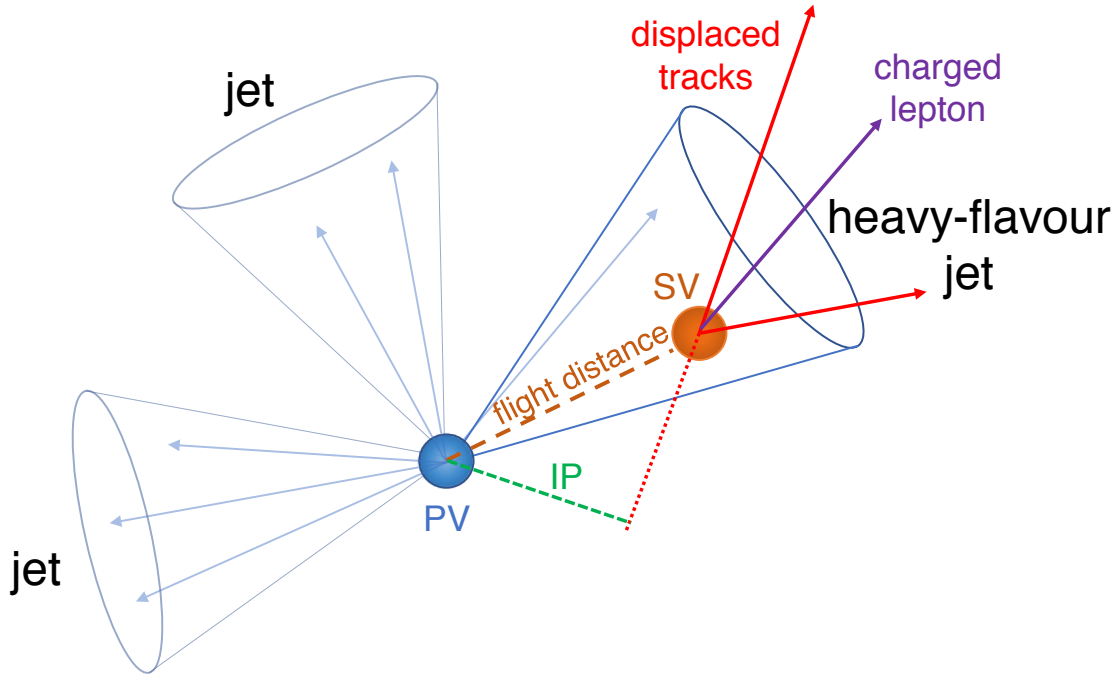


Figure 2.11: Characteristics of a jet originating from a b or c quark, which consists of tracks originated from a secondary vertex (SV) displaced from the primary vertex (PV) with a large impact parameter (IP). The figure is taken from Ref. [13].

working points are defined, corresponding to the b-tagging efficiencies of 65, 80, and 95%, respectively, measured in simulated top-quark pair events, with misidentification rates as light-flavor jets of 0.1, 1, and 10%, respectively, measured in simulated QCD multijet events [63].

2.3.4 Hadronic taus

The τ lepton has a mass of 1.777 GeV and has the following leading decay modes [26].

- Leptonic decays (τ_μ, τ_e)
 - $\tau^- \rightarrow \mu^- \bar{\nu}_\mu \nu_\tau$ 17.4%
 - $\tau^- \rightarrow e^- \bar{\nu}_e \nu_\tau$ 17.8%
- Hadronic decays (τ_h)
 - $\tau^- \rightarrow h^- \nu_\tau$ 11.5%

$-\tau^- \rightarrow h^- \pi^0 \nu_\tau$	25.9%
$-\tau^- \rightarrow h^- \pi^0 \pi^0 \nu_\tau$	9.5%
$-\tau^- \rightarrow h^- h^- h^+ \nu_\tau$	9.8%
$-\tau^- \rightarrow h^- h^- h^+ \pi^0 \nu_\tau$	4.8%

Here, h^\pm denotes a charged hadron such as a charged pion or a charged kaon, and the numbers shown are the measured branching ratios. The electrons and muons in the leptonic final states are reconstructed as previously described in Sec. 2.3.1 and Sec. 2.3.2 respectively. About two-thirds of the time, tau leptons decay hadronically into final states that contain either one or three charged hadrons, accompanied by a neutrino and sometimes neutral hadrons.

Reconstruction of τ_h is done using the hadron-plus-strips (HPS) algorithm [64]. A τ_h manifests as a narrow jet with a low multiplicity of jet constituents compared to gluon/quark jets. The final states are characterized by the number of charged hadrons and the number of π^0 .

The charged hadrons are reconstructed as tracks in the tracker with HCAL clusters. A π^0 promptly decays into a pair of photons that are likely to be converted into e^-e^+ pairs. The e^-e^+ pair will radiate bremsstrahlung photons and will be separated by the magnetic field, with a larger separation in ϕ than in η due to the direction of the field.

A π^0 is reconstructed by iteratively clustering electrons and photons within a $\Delta\eta \times \Delta\phi$ window called a strip. The strip size varies in $0.05 < \Delta\eta < 0.15$ and $0.05 < \Delta\phi < 0.3$, and is dynamically determined from the p_T of the electrons and photons being clustered. This dynamic adjustment is necessary because for boosted τ_h , the decay products are more collimated. The strip size can dynamically shrink to fully contain the potential candidates of electrons and photons, minimizing background contributions. However, if a larger strip size is used, it can also contain the candidates but will increase background contributions and contaminate the strip reconstruction. These strips are then combined with charged hadrons to form the τ_h candidates, with requirements on the reconstructed mass to be compatible with different decay modes.

Identification of τ_h from background objects is done using the **DeepTau** algorithm [65]. Background objects include electrons, muons, and gluon/quark jets, which can potentially be misidentified as τ_h . **DeepTau** is a neural network trained to simultaneously discriminate genuine τ_h against these three classes of background objects. The input features to the neural network include properties of all reconstructed individual particles near the τ_h candidate and properties of the τ_h candidate itself. Various working points are defined for each of the three discriminators by setting the target identification efficiencies measured from $h \rightarrow \tau\tau$ events with $\tau_h p_T \in [30, 70]$ GeV. The identification efficiencies range from 40 to 98%, 99.5 to 99.95%, and 60 to 99.5% for discriminators against jets, muons, and electrons, respectively.

Chapter 3

Search for Higgs boson decaying to a pair of pseudoscalars $h \rightarrow aa$

This chapter presents two analyses, and their statistical combination, of the search for exotic decays of the Higgs boson to a pair of light pseudoscalars, using proton-proton collision data at $\sqrt{s} = 13$ TeV, collected by the CMS detector during LHC Run 2 (three data-taking years 2016-2018), with an integrated luminosity of 138 fb^{-1} . Sec. 3.1 presents the analysis in the final state with two b quarks and two tau leptons ($h \rightarrow aa \rightarrow \tau\tau bb$). Sec. 3.2 presents the combination analysis between the $\tau\tau bb$ and the $\mu\mu bb$ final states. Sec. 3.3 discusses some future aspects.

3.1 Search for $h \rightarrow aa \rightarrow \tau\tau bb$

This section details the full Run 2 analysis of the search for $h \rightarrow aa \rightarrow \tau\tau bb$ [15].

Sec. 3.1.1 outlines the strategy of the analysis and improvements compared to a previous effort. Data and simulation samples are described in Sec. 3.1.2. Trigger and object selections are outlined in Sec. 3.1.3 and Sec. 3.1.4, respectively. Sec. 3.1.5 presents the background estimation. Sec. 3.1.6 presents the correction factors applied to simulation. Sec. 3.1.7 presents the dedicated neural networks used for optimization and the event categorization. Sec. 3.1.8 presents the systematic uncertainties. Sec. 3.1.9 presents the

signal extraction method. Sec. 3.1.10 presents the results as exclusion limits and model interpretation.

3.1.1 Analysis strategy

For the Higgs boson with a mass of 125 GeV, the decay $h \rightarrow aa$ is possible if $m_a \leq m_h/2 = 62.5$ GeV. The analysis probes the range of m_a from 12 to 60 GeV, where the final state-objects are resolved. For masses around 10 GeV and below, the decay products begin to merge, and dedicated reconstruction techniques will be needed.

The final state has two b quarks and two τ leptons. The b quarks undergo parton showering and hadronize to form b jets that are identified using the b-tagging algorithm discussed in Sec. 2.3.3. As discussed in Sec. 2.3.4, a τ lepton can decay hadronically (τ_h), into an electron (e), or a muon (μ), with accompanying neutrinos.

The branching ratios of $\tau\tau$ decays are: $\tau\tau \rightarrow \tau_h\tau_h$ ($\approx 42\%$), $\tau\tau \rightarrow \mu\tau_h$ ($\approx 23\%$), $\tau\tau \rightarrow e\tau_h$ ($\approx 23\%$), $\tau\tau \rightarrow e\mu$ ($\approx 6\%$), $\tau\tau \rightarrow ee$ ($\approx 3\%$), and $\tau\tau \rightarrow \mu\mu$ ($\approx 3\%$). Although the branching ratio is the largest in the $\tau_h\tau_h$ final state, it is not considered in the analysis due to the too high threshold on $\tau_h p_T$ in the existing trigger path, which is not efficient in selecting low-momentum τ_h originating from the light pseudoscalar. The ee and $\mu\mu$ final states are not considered due to their low branching ratios and the large Drell-Yan (DY) background ($Z \rightarrow ee/\mu\mu$). Therefore, only three $\tau\tau$ final states are analyzed: $\mu\tau_h$, $e\tau_h$, and $e\mu$.

The three leading background processes are top-quark pair production with additional jets (tt+jets), DY Z boson production with additional jets (Z+jets), and events with jets misidentified as one or more of the final-state objects (fakes). Other minor background processes include single top quark (ST), massive vector boson pair (VV), W boson with additional jets (W+jets), and SM Higgs boson. The $Z \rightarrow \tau\tau$ +jets background is estimated from a combination of data and simulation, the fake background is estimated using data-driven methods, and the rest are estimated from simulation.

Events are selected by requiring the presence of the $\tau\tau$ final-state objects and at least

one b-tagged jet. The selected events are then divided into two categories based on the number of b-tagged jets: events with exactly one b-tagged jet fall into a category called ‘1 b tag’, while events with more than one b-tagged jet fall into another category called ‘> 1 b tag’.

Each category is further divided into several subregions to optimize the signal sensitivity. For each of the two categories in each of the three $\tau\tau$ final states, a dedicated neural network is trained as a binary classifier to distinguish between the $h \rightarrow aa \rightarrow \tau\tau bb$ signal and various background processes. The input features to the neural networks include p_T , angular variables, and invariant masses constructed from combinations of the four final-state objects. The classifier output is interpreted as signal probability and used as a discriminant, with subregion threshold values defined by optimizing the overall sensitivity.

The invariant mass of the two τ leptons, $m_{\tau\tau}$, is used as the observable for the final likelihood fit. The signal hypothesis appears as a resonance in the $m_{\tau\tau}$ distribution, with different mass points peaking at their corresponding values. This allows for the identification of the signal mass potentially responsible for any excess in the data. Finally, a binned maximum likelihood fit is performed simultaneously on the $m_{\tau\tau}$ distributions in all subregions.

Compared to a previous effort [16], which was the first LHC search in this final state done by CMS using partial Run 2 data collected in 2016 with an integrated luminosity of 35.9 fb^{-1} (referred to as the 2016-only analysis), the full Run 2 analysis has the following improvements.

1. **Optimization using machine learning.** The 2016-only analysis performed sequential optimization selection (cut-based), applying hard-threshold cuts on several discriminating variables for optimization. In contrast, the full Run 2 analysis employs dedicated neural networks trained to distinguish between signal and background events. These neural networks utilize more discriminating variables, allowing for the machine-learning of correlations between them. This approach results in additional sensitivity improvement beyond what is achieved by the increased data size

alone.

2. **Advanced object reconstruction techniques.** Central algorithms for object selection in CMS have been improved since the 2016-only analysis. Advancements in algorithms for τ_h and b-tagging, for instance, lead to better reconstruction and identification efficiencies.
3. **Advanced background estimation techniques.** One of the main background processes, $Z \rightarrow \tau\tau$, was estimated entirely from simulation in the 2016-only analysis. In the full Run 2 analysis, it is estimated from a combination of data and simulation, reducing sources of systematic uncertainties.
4. **Advanced signal mass reconstruction technique.** In the 2016-only analysis, the observable $m_{\tau\tau}$ was reconstructed from only the visible parts of the τ leptons without accounting for neutrinos. In the full Run 2 analysis, a dedicated algorithm is used to account for missing energy, providing a more accurate reconstruction and a better mass resolution.
5. **Finer-grained selection.** The 2016-only analysis selected events with only one bin in the b jet multiplicity—requiring at least one b jet. The full Run 2 analysis separates events into categories with exactly one b jet and at least two b jets, leading to better sensitivity.
6. **Increased low-mass sensitivity.** The selection in the $e\mu$ final state is modified to improve sensitivity for low mass signals, down to 12 GeV, which was not possible in the 2016-only analysis.
7. **Increased statistics.** The statistics of the data samples used increases from 35.9 fb^{-1} in 2016 to 138 fb^{-1} combining all three years (2016-2018) in Run 2. This leads to an improvement in the expected exclusion limits by a factor of 2 due to luminosity scaling.

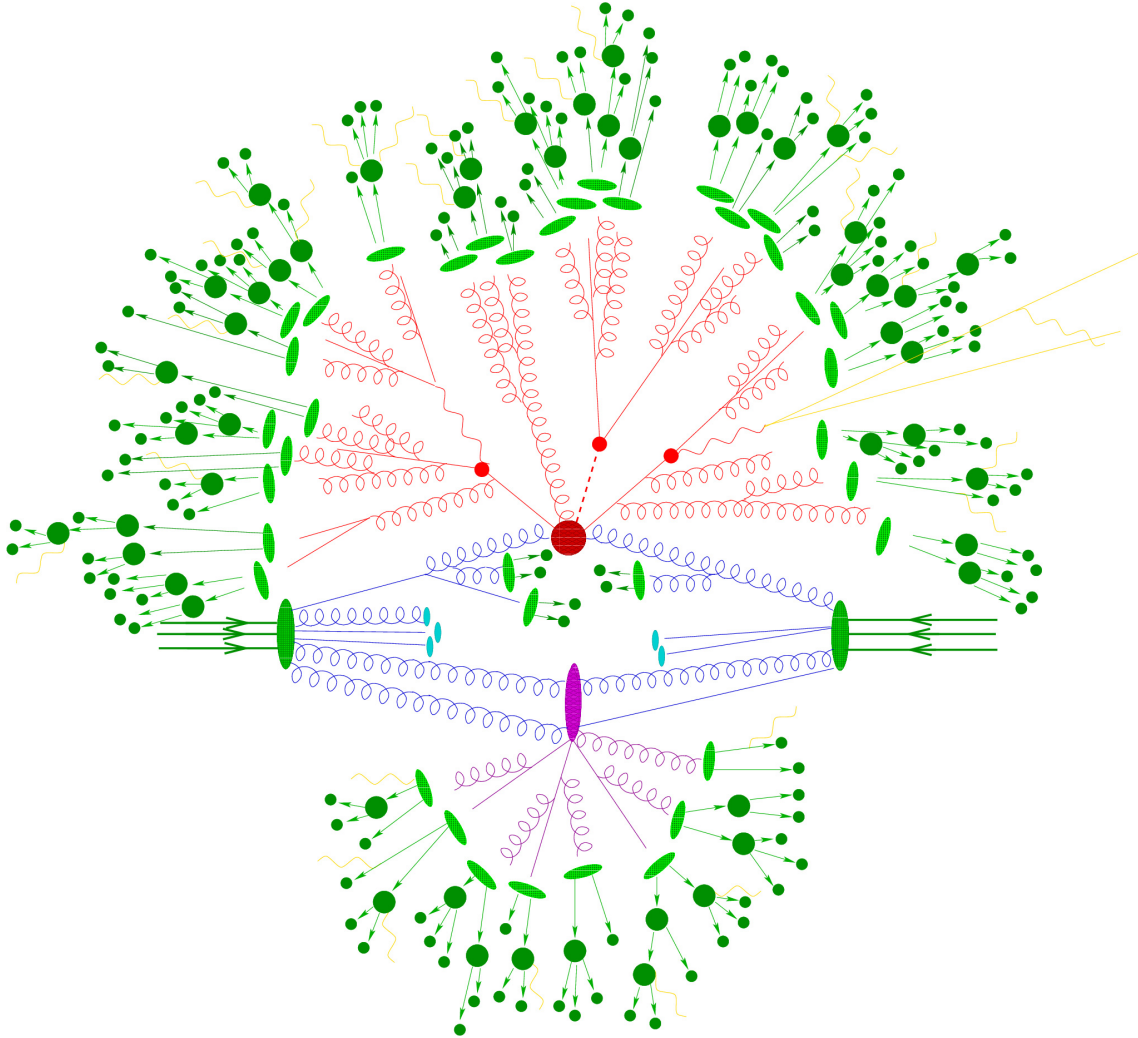


Figure 3.1: Schematic view of an event of the production of a Higgs boson in association with a pair of top quarks ($t\bar{t}h$), simulated by a MC generator. The two incident protons are represented by the big ovals in green. The big circle at the center represents the hard scattering process that produces a top-quark pair and a Higgs boson, with their subsequent decays represented by the small circles in red. The oval in purple represents the secondary interaction called the underlying event. Parton showering is followed by hadronization, represented by the small ovals in green. Finally, the hadron decays are represented by the circles in dark green. The figure is taken from Ref. [14].

3.1.2 Simulation

Simulation is necessary to provide expected background estimation and signal modeling, which are compared with the observed data for signal extraction. Fig. 3.1 illustrates different processes involved in a typical proton-proton collision event. When protons collide

at high energy, hard scattering processes occur, which are described by perturbative QCD and are model-dependent (SM or BSM). Since protons are composite particles, secondary interactions can also occur, called underlying events, which are not the event of interest. Additional QCD radiation is produced, creating cascades of radiation called parton (quarks and gluons) showers, and the average parton energy decreases during the process. Below the QCD scale, partons start to hadronize to form hadrons, which then further decay into stable particles to be detected by the detector.

These processes can be factorized and simulated by Monte Carlo (MC) generators. Several independent generators can provide matrix element calculations for hard scattering processes, depending on the physics models. The DY Z+jets, W+jets, and VV events are generated using the `MadGraph5_aMC@NLO` package [66]. The tt+jets and ST events are generated using the `POWHEG` package [67–70]. The SM Higgs boson events are generated using the `POWHEG` package [71–75].

The signal $h \rightarrow aa \rightarrow \tau\tau bb$ events are generated by the `MadGraph5_aMC@NLO` package using the NMSSMHET model [50]. Mass points from 15 to 60 GeV in steps of 5 GeV, and an additional mass point of 12 GeV, are generated. The two leading production channels, gluon-gluon fusion (ggF) and vector boson fusion (VBF) (see Sec 1.1.3), are considered for the signal generation.

The parton distribution functions (PDFs) are described by the NNPDF3.1 set [76]. Parton showering and hadronization are simulated using the `PYTHIA` package [77]. The underlying events for the 2016 (2017-2018) data are described by the CUETP8M1 [78] (CP5 [79]) tune. Jets are matched between the matrix element level and the showering level [80–82]. Furthermore, pileup interactions are simulated and superimposed onto the simulated hard scattering events, which are reweighted to match the pileup distribution in data. The CMS detector simulation is performed using the `GEANT4` package [83, 84].

Table 3.1: Summary of trigger p_T thresholds in the three data-taking years [15]. The p_T requirements apply to the corresponding HLT-reconstructed objects. In each final state per year, events are selected if they pass any of the available triggers.

$\mu\tau_h$			
	2018	2017	2016
single- μ $p_T(\mu)$	24, 27	24, 27	22
cross- $\mu\tau_h$ ($p_T(\mu), p_T(\tau_h)$)	(20, 27)	(20, 27)	(19, 20)
$e\tau_h$			
	2018	2017	2016
single-e $p_T(e)$	32, 35	27, 32	25
cross- $e\tau_h$ ($p_T(e), p_T(\tau_h)$)	(24, 30)	(24, 30)	—
$e\mu$			
	2018	2017	2016
cross- $e\mu$ ($p_T(e), p_T(\mu)$)	(23, 8), (12, 23)	(23, 8), (12, 23)	(23, 8), (12, 23)

3.1.3 Trigger selection

The trigger (online) selection is based on the final-state objects from the two τ . Depending on the data-taking period and the $\tau\tau$ final state, there are different trigger paths that select on a single lepton or both objects. The $\mu\tau_h$ final state uses single- μ and cross- $\mu\tau_h$ triggers; the $e\tau_h$ final state uses single-e and cross- $e\tau_h$ triggers; the $e\mu$ final state uses cross- $e\mu$ triggers. These triggers require the corresponding electron or muon reconstructed at the HLT to be isolated. The p_T thresholds for the triggers are tabulated in Tab. 3.1.

3.1.4 Object selection

In the $\mu\tau_h$, $e\tau_h$, and $e\mu$ final states, the presence of a pair of $\mu\tau_h$, $e\tau_h$, and $e\mu$ with opposite electric charge is required, respectively. Events with extra electrons or muons are rejected to suppress background contributions from DY and multilepton events.

Muons are required to be within $|\eta| < 2.4$. They must pass the medium identification working point, which has an overall efficiency of 99.5% [10] (see Sec. 2.3.2). In the $\mu\tau_h$ final state, the offline selection requires the muon p_T threshold to be 1 GeV higher than the corresponding trigger threshold listed in Tab. 3.1, where the trigger becomes fully efficient with the efficiency independent of the p_T . For the cross- $e\mu$ triggers in the $e\mu$ final

state, the muon p_T threshold is required to be 1 GeV higher when the online threshold is 23 GeV, for the same reason above. However, it is required to be 5 GeV higher when the online threshold is 8 GeV, as a too low p_T threshold would lead to a high misidentification rate.

Electrons are required to be within $|\eta| < 2.4$. They must pass the multivariate identification at a working point that has an efficiency of 90% [57] (see Sec. 2.3.1). In the $e\tau_h$ and $e\mu$ final states, the offline selection requires the electron p_T threshold to be 1 GeV higher than the corresponding trigger threshold listed in Tab. 3.1.

To suppress misidentification rate of a prompt lepton from non-prompt or fake leptons, it is useful to calculate the energy sum from other particles surrounding the lepton. This is done by defining the relative lepton isolation [10, 57]:

$$I_{\text{rel}}^{l=\{e,\mu\}} = \frac{\sum_{\text{charged}} p_T + \max(0, \sum_{\text{neutral}} p_T - p_T^{\text{pileup(neutral)}})}{p_T^{l=\{e,\mu\}}}. \quad (3.1)$$

Here, $\sum_{\text{charged}} p_T$ sums over the charged particles with tracks originating from the primary vertex, while $\sum_{\text{neutral}} p_T$ sums over the neutral particles, and the lepton of interest is excluded from these sums. All the sums are calculated within a cone centered around the lepton direction with a radius of $\Delta R = \sqrt{(\Delta\phi)^2 + (\Delta\eta)^2}$. Since neutral particles do not have tracking information, the neutral particle sum includes contributions from both the primary vertex and pileup vertices. Therefore, the term $p_T^{\text{pileup(neutral)}}$ represents the neutral particles from pileup and is subtracted from the neutral particle sum. For muon isolation, $p_T^{\text{pileup(neutral)}}$ is estimated from charged particles associated with pileup vertices, using a known ratio of neutral to charged hadron production in inelastic proton-proton collisions, which is estimated from simulation. For electron isolation, it is estimated from the pileup energy flow density.

The size of the isolation cone is set to be $\Delta R = 0.4$ for muons and $\Delta R = 0.3$ for electrons. Muons and electrons are required to be isolated: $I_{\text{rel}}^\mu < 0.15$ and $I_{\text{rel}}^e < 0.1$. In the $e\mu$ final state, the electron and the muon are required to be separated by $\Delta R > 0.3$.

In the $\mu\tau_h$ and $e\tau_h$ final states, τ_h candidates are selected using the HPS algorithm [64]

(see Sec. 2.3.4). The τ_h candidates are required to be within $|\eta| < 2.1$ (2.3) if the events (do not) pass the cross triggers. The τ_h p_T threshold in offline selection is required to be 5 GeV higher than the corresponding trigger threshold listed in Tab. 3.1. For events that pass only the single lepton triggers, the τ_h candidates are required to have $p_T > 20$ GeV. They are also required to be separated from the selected electron or muon by $\Delta R > 0.4$.

To reduce the τ_h misidentification rates against jets, electrons, and muons, the **DeepTau** algorithm is applied [65] (see Sec. 2.3.4). The medium working point is chosen for the discriminator against jets in both final states, which corresponds to an efficiency of 70%. In the $\mu\tau_h$ final state, the tight working point is chosen for the discriminator against muons, and the very-loose working point is chosen for the discriminator against electrons, effectively suppressing the large $Z \rightarrow \mu\mu$ background. Conversely, in the $e\tau_h$ final state, the tight working point is chosen for the discriminator against electrons, and the very-loose working point is chosen for the discriminator against muons, effectively suppressing the large $Z \rightarrow ee$ background.

Additionally, at least one b-tagged jet is required in all final states. First, at least one untagged jet within $|\eta| < 2.4$ and with $p_T > 20$ GeV is selected, which must be separated from any of the selected electron, muon, or τ_h by $\Delta R > 0.5$. Then, b jets are tagged using the **DeepJet** algorithm [13, 62] (see Sec. 2.3.3). The medium working point is chosen, corresponding to a b-tagging efficiency of 80% and a misidentification rate of 1% against light-flavor jets.

3.1.5 Background estimation

Some background processes are estimated directly from simulation, some are estimated entirely from data, and some are estimated from a combination of data and simulation. The estimation methods are described as follows.

- $Z \rightarrow \tau\tau$ is estimated using the embedded technique [85]. In this method, $Z \rightarrow \mu\mu$ events are selected from data as the baseline events, as they have a clean signature with relatively low background contamination. Then, the muon pairs are removed

and replaced by simulated τ pairs that have the same kinematic properties. This results in the $Z \rightarrow \tau\tau$ events estimated from data, except for the decays of the τ leptons being simulated, reducing the number of corrections needed and systematic uncertainties due to simulation. In fact, due to the lepton universality in the SM except in the scalar sector, the estimation also includes all other SM non-Higgs processes with two τ leptons. In principal, the selected data can include $h \rightarrow \mu\mu$ events, but since the Yukawa coupling is directly proportional to the fermion mass, so the embedded $h \rightarrow \tau\tau$ is negligible compared to its actual rate. Therefore, these embedded $\tau\tau$ events have to be excluded from all other SM background processes to avoid double counting, except for the Higgs boson processes, which need to be estimated separately.

- **jets $\rightarrow \tau_h$ fake** accounts for events inclusively with a jet misidentified as the τ_h candidate in the $\mu\tau_h$ and $e\tau_h$ final states. This background is estimated from data in a sideband region using the fake rate method with the following steps.
 1. The first step is to measure the fake rate, f , which is defined as the probability for a jet reconstructed as a τ_h that passes the τ_h identification against jets. This fake rate is a function of the $\tau_h p_T$ and is measured from the $Z \rightarrow \mu\mu + \text{jets}$ events selected from data. This rate is also measured separately for the $\mu\tau_h$ and $e\tau_h$ final states by setting the corresponding working points of the τ_h identification against electrons and muons.
 2. Then, an orthogonal sideband region is defined with the same baseline selection, except that the τ_h candidate fails the τ_h identification, making it likely to be a jet fake.
 3. In the sideband region, start from the data and subtract from it all the MC simulated events where the τ_h candidate is either a genuine τ_h or a lepton fake. This step removes the non-jet fakes from the estimation. Essentially, these data events represent the inclusive background events with a jet reconstructed as the τ_h candidate which fails the τ_h identification.

4. Finally, scale these data events by the factor $\frac{f}{1-f}$, which represents the ratio between the number of jets passing the τ_h identification and the number of jets failing the τ_h identification. These scaled data now represent the inclusive background events that have a jet reconstructed as the τ_h candidate which passes the τ_h identification.

Since this method inclusively estimates all background events with a jet faking the τ_h , these events should be excluded from all the simulated events to avoid double counting.

- **QCD multijet** accounts for QCD events with jets misidentified as one or both leptons in the $e\mu$ final state. In the following, OS (SS) denotes opposite-sign (same-sign) electric charge of the $e\mu$ pair. This background is estimated from data in different sideband regions with the following steps.

1. An orthogonal sideband region, called the SS region, is defined with the same baseline selection except with a SS $e\mu$ pair. Conversely, the baseline region is called the OS region.
2. In the SS region, start from the data and subtract all the MC simulated events. This removes all background contributions except the QCD multijet process, as it is not estimated from any simulated samples. These data events, with both leptons being isolated, are denoted as $\text{QCD}_{\text{SS}}^{\text{iso-e, iso-}\mu}$.
3. There are three multiplicative scale factors needed to extrapolate these data events from the SS region to the OS region using other sideband regions by inverting the isolation of the two leptons. The first one is the OS-to-SS scale factor, which is the ratio of the number of QCD events between the OS and SS regions: $\text{QCD}_{\text{OS}}^{\text{iso-e, anti-}\mu} / \text{QCD}_{\text{SS}}^{\text{iso-e, anti-}\mu}$. This is parameterized by the separation between the leptons, $\Delta R(e, \mu)$, measured from another orthogonal region with an isolated electron and an anti-isolated muon.
4. The second scale factor is a 2D correction factor to account for the lepton p_T

dependence of the OS-to-SS scale factor. This is measured similarly to the OS-to-SS scale factor but parameterized by the p_T of both leptons and serves as a closure correction to the OS-to-SS scale factor.

5. The third scale factor is a 2D correction factor to adjust the OS-to-SS scale factor from having an anti-isolated muon to an isolated muon. This factor is parameterized by the p_T of both leptons. It is measured by taking the ratio between the OS-to-SS scale factor measured with an anti-isolated electron and an isolated muon, and the OS-to-SS scale factor measured with both leptons being anti-isolated: $\frac{\text{QCD}_{\text{OS}}^{\text{anti-e, iso-}\mu} / \text{QCD}_{\text{SS}}^{\text{anti-e, iso-}\mu}}{\text{QCD}_{\text{OS}}^{\text{anti-e, anti-}\mu} / \text{QCD}_{\text{SS}}^{\text{anti-e, anti-}\mu}}$.

Since this method accounts exclusively for the QCD multijet process, there is no double counting with other background processes.

- **tt+jets** is estimated from MC simulation. It includes three possible decay modes of the W bosons from the top quarks: 1) both W bosons decay hadronically, 2) one W boson decays leptonically and the other hadronically, and 3) both W bosons decay leptonically. To avoid double counting with the embedded $\tau\tau$ events, the simulated tt+jets events that contain two genuine τ leptons are excluded. To avoid double counting with the jet $\rightarrow \tau_h$ events in the $\mu\tau_h$ and $e\tau_h$ final states, the simulated tt+jets events that contain the τ_h candidate originating from a jet are excluded. These exclusions are applied to all other simulated processes.
- **$Z \rightarrow ee/\mu\mu$, ST, VV, SM Higgs boson** are the remaining background processes and have small contributions. These are all estimated from the MC simulation.

For each simulated process with a cross section σ and an arbitrary number of simulated events $N_{\text{generated}}$, all simulated events are scaled by a factor of $\sigma\mathcal{L}/N_{\text{generated}}$ to produce the yield expected for the integrated luminosity of \mathcal{L} .

3.1.6 Corrections to simulation

Simulation samples are used to predict the expected yields of the signal and background processes under study. They are based on the best-known theoretical predictions and experimental conditions, aiming to accurately represent the underlying physics processes for interpreting of the real data.

However, there can be various sources of mis-modeling or imperfections in the simulation, causing disagreements between data and simulation and affecting the reliability of the simulated events. For example, the actual detector response and resolution may not be perfectly modeled in simulation, leading to simulated object energy being systematically lower or higher than the measured energy. Therefore, scale factors are derived and applied to the simulated quantities to align them with the data. These corrections are necessary to ensure proper agreement with real data, providing a more accurate description and reducing systematic uncertainties.

The main corrections applied to simulation in the analysis are listed below.

- **Trigger efficiencies** are corrected by scaling the corresponding triggered events to account for the differences in trigger performance between simulation and data.
- **Pileup reweighting** is applied to correct the pileup profile in simulation to match that observed in data.
- **Lepton reconstruction and identification efficiencies** are corrected separately for electrons and muons.
- **Electron energy** is corrected for the scale and resolution.
- **τ_h reconstruction and identification efficiencies** are corrected separately for genuine and misidentified τ_h candidates.
- **τ_h energy** is corrected separately for genuine and misidentified τ_h candidates.
- **Jet energy** is corrected by applying multiplicative scale factors that account for various mis-measurement effects, adjusting the scale of the jet energy distribution.

Jet energy resolution is then corrected by smearing, which adjusts the shape of the jet energy distribution.

- **Missing transverse energy** is corrected for resolution and response by applying recoil corrections to W+jets, Z+jets, and Higgs boson samples. For samples without recoil corrections, correction is applied to account for the energy measurements not clustered into jets.
- **B-tagging efficiencies** are corrected as a function of the b-tagging discriminant, jet flavor, p_T , and η .
- **Top quark p_T spectrum** is corrected in the tt+jets samples as a function of the top p_T to account for the missing higher-order calculations at the matrix element level in simulation.
- **Z p_T spectrum** is corrected in the DY $Z \rightarrow e\bar{e}/\mu\bar{\mu}$ samples as a function of the Z p_T to account for the missing higher-order calculations at the matrix element level in simulation.

3.1.7 Optimization using neural networks

In the following, we first discuss the signal mass reconstruction and then the machine learning-based optimization.

The SVFit algorithm [86] is used to reconstruct the invariant mass of the two τ leptons, $m_{\tau\tau}$, which serves as the observable for signal extraction. Instead of reconstructing the mass from only the visible τ decay products, $m_{\tau\tau}^{\text{vis}}$, the algorithm reconstructs the full mass, by taking into account the missing transverse energy from the invisible neutrinos. Compared to $m_{\tau\tau}^{\text{vis}}$, the $m_{\tau\tau}$ distribution of the signal $h \rightarrow a\bar{a} \rightarrow \tau\tau b\bar{b}$ has a more accurate resonance peak centered around the corresponding mass hypothesis and has an improved resolution.

After the baseline selection, events with two τ leptons and at least one b-tagged jet passing various object quality tests are selected and categorized into the ‘1 b tag’ and ‘>

Table 3.2: Composition ratios of the three leading background processes in the background class.

	tt+jets (%)	$Z \rightarrow \tau\tau$ (%)	fakes (%)
$\mu\tau_h$ ‘1 b tag’	42.4	17.4	40.2
$\mu\tau_h$ ‘> 1 b tag’	68.5	6.2	25.3
$e\tau_h$ ‘1 b tag’	51.3	12.5	36.2
$e\tau_h$ ‘> 1 b tag’	73	4.8	22.2
$e\mu$ ‘1 b tag’	92.5	3.1	4.4
$e\mu$ ‘> 1 b tag’	98.5	0.7	0.8

1 b tag’ categories. The categories are further divided into subregions to optimize the overall sensitivity (signal-to-background ratio) using deep neural networks (DNN).

One DNN is dedicated to each category in each $\tau\tau$ final state, resulting in a total of $(2 \text{ categories}) \times (3 \text{ } \tau\tau \text{ final states}) = 6$ DNNs. Each DNN is trained as a binary classifier to distinguish the signal class from the background class. Details of the DNN optimization are presented below.

Training data preparation

The training data is split into two classes: signal and background. The signal class is prepared by combining simulated signal events of all 11 generated mass points. The background class is prepared by combining only the three leading background processes: tt+jets, $Z \rightarrow \tau\tau$, and fakes. The tt+jets events are taken from the simulated samples. The $Z \rightarrow \tau\tau$ events are taken from the embedded samples. In the $\mu\tau_h$ and $e\tau_h$ final states, fakes refers to the $\text{jet} \rightarrow \tau_h$ events estimated from the data-driven fake rate method; in the $e\mu$ final state, fakes refers to the QCD multijet events estimated from the data-driven OS/SS method.

The three background processes are combined into a single background class without applying event weights, using composition ratios according to the weighted $m_{\tau\tau}$ distribution after baseline selection. This approximation represents the true background composition and simplifies training by avoiding the need to weight the training data. The background composition ratios are listed in Tab. 3.2.

To ensure unbiased training and classification, the number of training events in the

Table 3.3: Numbers of training events in the signal and background classes. All events are required to pass the baseline selection.

	signal	tt+jets	$Z \rightarrow \tau\tau$	fakes
$\mu\tau_h$ ‘1 b tag’	158664	67273	27608	63783
$\mu\tau_h$ ‘> 1 b tag’	28721	19674	1781	7266
$e\tau_h$ ‘1 b tag’	51753	26549	6469	18735
$e\tau_h$ ‘> 1 b tag’	10046	7334	482	2230
$e\mu$ ‘1 b tag’	110020	101768	3411	4841
$e\mu$ ‘> 1 b tag’	16733	16482	117	134

signal class should be equal to that in the background class. The number of training events in each class is set by the number of available simulated signal events, which is much less than the available background events. In the background class, the number of training events for the three background processes is allocated according to the composition ratios in Tab. 3.2. Additionally, all training events are required to pass the baseline selection to represent the actual distributions in the analysis. All three years of data are combined for maximum statistics, and the DNNs are trained without differentiating between the years. The numbers of training events are tabulated in Tab. 3.3.

Input features

Essentially, a DNN classifier learns the correlation of the input features that distinguish between different classes. The input features are chosen based on the principle that the signal class and the background class should be differently distributed in these variables to maximize discrimination power.

An important point to note is that the $m_{\tau\tau}$ variable should be explicitly excluded from the list of input features. This is because $m_{\tau\tau}$ is the observable for signal extraction in the analysis. The DNN discriminant is used to define subregions of $m_{\tau\tau}$ that maximize the overall signal-to-background ratio. If the DNN discriminant is highly correlated with $m_{\tau\tau}$, defining subregions by cutting on the discriminant value would introduce shaping effects on the signal resonant distribution in $m_{\tau\tau}$, biasing and invalidating the optimization. Therefore, the $m_{\tau\tau}$ variable is not considered as an input to the DNNs, and the correlation between the DNN output and $m_{\tau\tau}$ is checked after training.

The input features include lower-level kinematic properties of the final-state objects and higher-level variables such as angular separation and invariant masses. In the ‘1 b tag’ category, these variables are constructed from the two τ objects and the leading b-tagged jet. In the ‘> 1 b tag’ category, there are more variables due to the additional sub-leading b-tagged jet.

All the input features are listed below.

- $\mu\tau_h$ ‘1 b tag’: $p_T(\mu), p_T(b_1), p_T(\tau\tau), \eta(\tau\tau), m_{b_1\mu}, m_{b_1\tau_h}, m_{b_1\tau\tau}, \Delta R(\mu, \tau_h), \Delta R(b_1, \mu), \Delta R(b_1, \tau_h), \Delta R(b_1, \tau\tau), m_T(\mu, p_T^{\text{miss}}), m_T(\tau_h, p_T^{\text{miss}}), m_T(b_1, p_T^{\text{miss}}), D_\zeta$.
- $\mu\tau_h$ ‘> 1 b tag’: $p_T(\mu), p_T(b_1), p_T(b_2), m_{b_1\mu}, m_{b_1\tau_h}, m_{b_2\mu}, m_{b_2\tau_h}, m_{bb}, m_{b_1\tau\tau}, m_{b_2\tau\tau}, m_{bb\mu}, m_{bb\tau_h}, m_{bb\tau\tau}, \Delta R(\mu, \tau_h), \Delta R(b_1, \mu), \Delta R(b_1, \tau_h), \Delta R(b_2, \mu), \Delta R(b_2, \tau_h), \Delta R(b_1, b_2), \Delta R(b_1, \tau\tau), \Delta R(b_2, \tau\tau), \Delta R(bb, \mu), \Delta R(bb, \tau_h), \Delta R(bb, \tau\tau), m_T(\mu, p_T^{\text{miss}}), m_T(\tau_h, p_T^{\text{miss}}), m_T(b_1, p_T^{\text{miss}}), m_T(b_2, p_T^{\text{miss}}), \Delta m_a, D_\zeta$.
- $e\tau_h$ ‘1 b tag’: $p_T(e), p_T(b_1), p_T(\tau\tau), \eta(\tau\tau), m_{b_1e}, m_{b_1\tau_h}, m_{b_1\tau\tau}, \Delta R(e, \tau_h), \Delta R(b_1, e), \Delta R(b_1, \tau_h), \Delta R(b_1, \tau\tau), m_T(e, p_T^{\text{miss}}), m_T(\tau_h, p_T^{\text{miss}}), m_T(b_1, p_T^{\text{miss}}), D_\zeta$.
- $e\tau_h$ ‘> 1 b tag’: $p_T(e), p_T(b_1), p_T(b_2), m_{b_1e}, m_{b_1\tau_h}, m_{b_2e}, m_{b_2\tau_h}, m_{bb}, m_{b_1\tau\tau}, m_{b_2\tau\tau}, m_{bbe}, m_{bb\tau_h}, m_{bb\tau\tau}, \Delta R(e, \tau_h), \Delta R(b_1, e), \Delta R(b_1, \tau_h), \Delta R(b_2, e), \Delta R(b_2, \tau_h), \Delta R(b_1, b_2), \Delta R(b_1, \tau\tau), \Delta R(b_2, \tau\tau), \Delta R(bb, e), \Delta R(bb, \tau_h), \Delta R(bb, \tau\tau), m_T(e, p_T^{\text{miss}}), m_T(\tau_h, p_T^{\text{miss}}), m_T(b_1, p_T^{\text{miss}}), m_T(b_2, p_T^{\text{miss}}), \Delta m_a, D_\zeta$.
- $e\mu$ ‘1 b tag’: $p_T(e), p_T(\mu), p_T(b_1), p_T(\tau\tau), \eta(\tau\tau), m_{b_1e}, m_{b_1\mu}, m_{b_1\tau\tau}, \Delta R(e, \mu), \Delta R(b_1, e), \Delta R(b_1, \mu), \Delta R(b_1, \tau\tau), m_T(e, p_T^{\text{miss}}), m_T(\mu, p_T^{\text{miss}}), m_T(b_1, p_T^{\text{miss}}), D_\zeta$.
- $e\mu$ ‘> 1 b tag’: $p_T(e), p_T(\mu), p_T(b_1), p_T(b_2), m_{b_1e}, m_{b_1\mu}, m_{b_2e}, m_{b_2\mu}, m_{bb}, m_{b_1\tau\tau}, m_{b_2\tau\tau}, m_{bbe}, m_{bb\mu}, m_{bb\tau\tau}, \Delta R(e, \mu), \Delta R(b_1, e), \Delta R(b_1, \mu), \Delta R(b_2, e), \Delta R(b_2, \mu), \Delta R(b_1, b_2), \Delta R(b_1, \tau\tau), \Delta R(b_2, \tau\tau), \Delta R(bb, e), \Delta R(bb, \mu), \Delta R(bb, \tau\tau), m_T(e, p_T^{\text{miss}}), m_T(\mu, p_T^{\text{miss}}), m_T(b_1, p_T^{\text{miss}}), m_T(b_2, p_T^{\text{miss}}), \Delta m_a, D_\zeta$.

Here, b_1 and b_2 refer to the leading and sub-leading b-tagged jets, respectively. $\tau\tau$ represents the reconstructed di- τ system using the SVFit algorithm. p_T^{miss} is the magnitude of

the missing transverse momentum vector, \vec{p}_T^{miss} , calculated as the negative vector sum of the transverse momenta of all the PF candidates in an event [87].

The transverse mass of an object $X = \{e, \mu, \tau_T, b_1, b_2\}$ and the missing momentum is defined as:

$$m_T(X, p_T^{\text{miss}}) \equiv \sqrt{2p_T^X p_T^{\text{miss}} (1 - \cos(\Delta\phi))}, \quad (3.2)$$

where p_T^X is the transverse momentum of the object X , and $\Delta\phi$ is the azimuthal angle between the X momentum and \vec{p}_T^{miss} . Signals typically have a small value of m_T , while backgrounds such as W +jets and $t\bar{t}$ +jets tend to have larger values of m_T due to a higher p_T^{miss} .

The D_ζ variable is defined as [88]:

$$D_\zeta \equiv p_\zeta - 0.85p_\zeta^{\text{vis}}, \quad (3.3)$$

where ζ denotes the axis along the bisector of the transverse momenta of the two τ candidates, p_ζ is the projection of \vec{p}_T^{miss} onto to the ζ axis, and p_ζ^{vis} is the projection of the visible τ decay products onto the ζ axis.

In the ‘> 1 b tag’ category, the presence of all four final-state objects allows for the reconstruction of both pseudoscalar masses and their difference:

$$\Delta m_a \equiv \frac{m_{b\bar{b}} - m_{\tau\tau}}{m_{\tau\tau}}. \quad (3.4)$$

Δm_a is typically narrowly peaked around 0 for the signal while widely distributed for the background.

Moreover, the signal typically has lower- p_T final-state objects since they are the decay products of the light pseudoscalars, while the background has a wider p_T spectrum. Similarly, the signal typically has more narrowly peaked invariant masses than the background. The final-state objects of the signal have a more correlated angular separation between each other, restricted by the unique decay topology, while the angular separation is more randomly distributed for the background. The signal and background distributions of the

input features for the ‘1 b tag’ category are shown in Fig. 3.2, and those for the ‘> 1 b tag’ category are shown in Fig. 3.3 and Fig. 3.4, in the $\mu\tau_h$ final state as an illustration.

The correlations between the input features are shown in Fig. 3.5 for the ‘1 b tag’ category and in Fig. 3.6 for the ‘> 1 b tag’ category, using the $\mu\tau_h$ final state as an illustration. In signal events, some inputs can be more correlated than others because they are of the same kind but only constructed from different combinations of the final-state objects ($\tau\tau b\bar{b}$), all originating from the two pseudoscalar decays that originated from the same Higgs boson decay. Highly correlated inputs include invariant masses and ΔR . In comparison, the inputs are less correlated for background events, since the final-state objects from different background processes are not necessarily from the same resonant decays and may even be completely independent of one another.

Having some highly correlated input features is not a concern for the analysis as long as there is no overfitting of the model and the model output is not correlated with the observable $m_{\tau\tau}$, which is used for signal extraction. These will be checked after the models are trained.

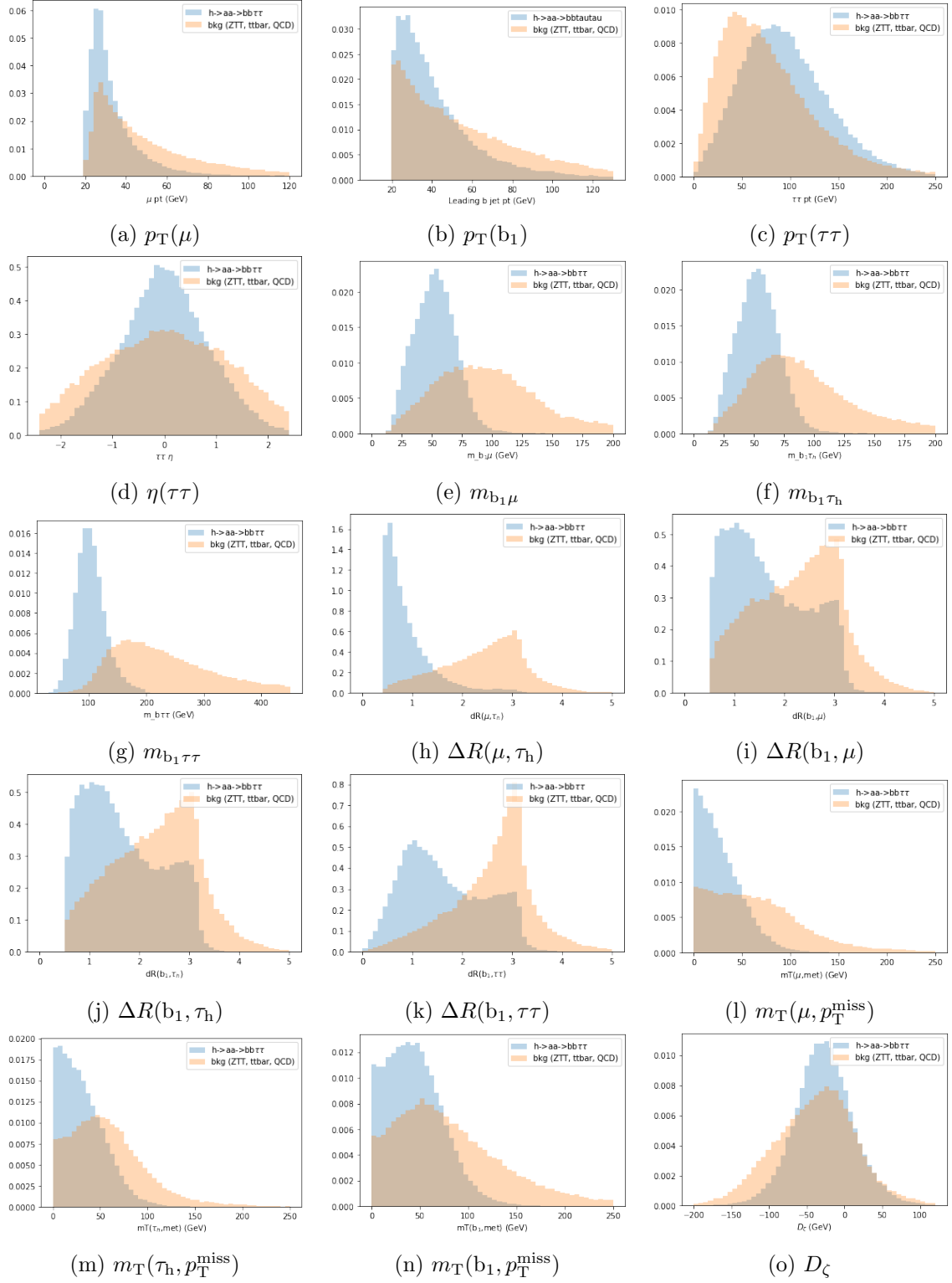


Figure 3.2: The 15 input features for the ‘1 b tag’ category in the $\mu\tau_h$ final state. The signal (background) class is represented by the distributions in blue (orange).

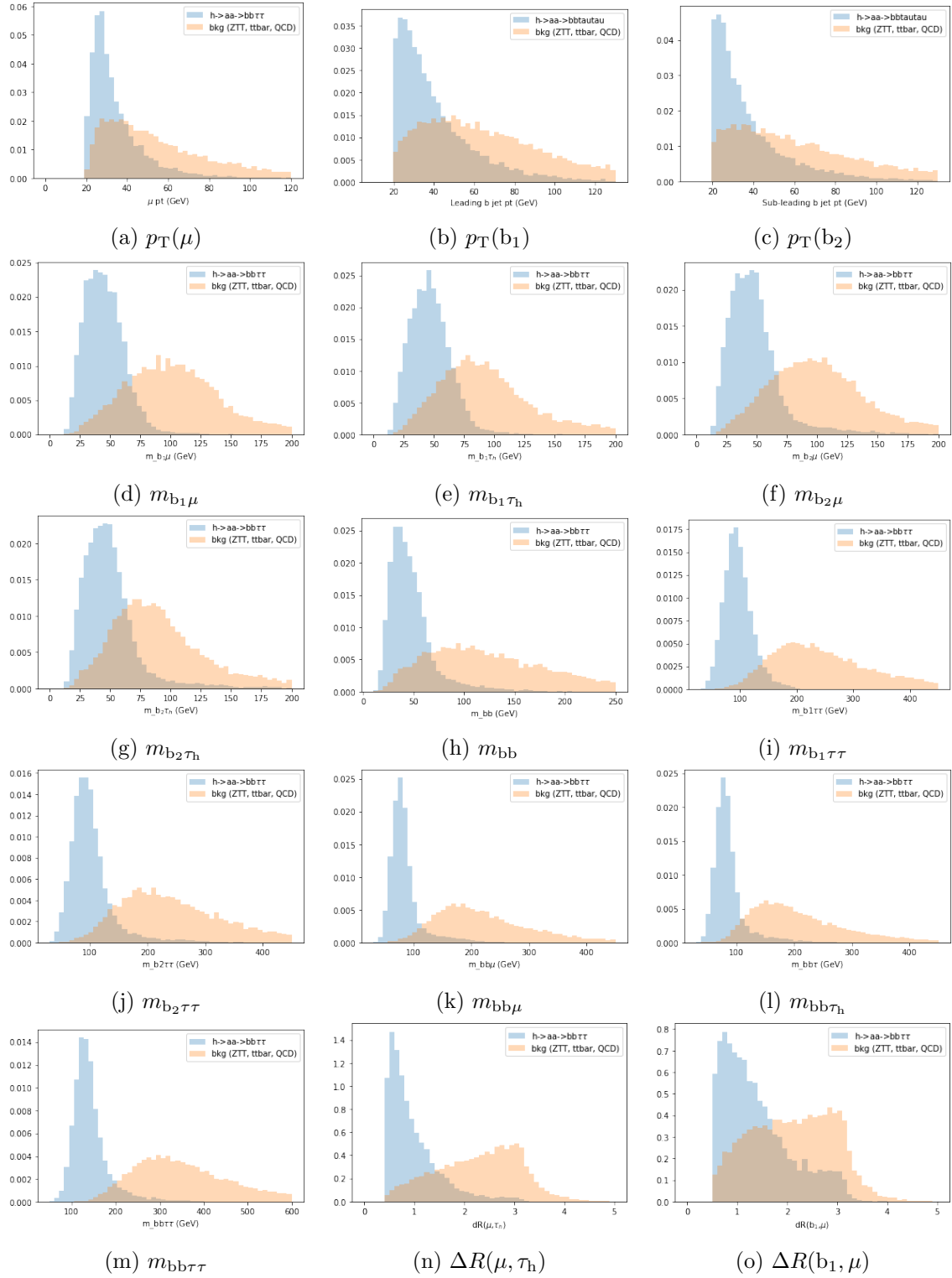


Figure 3.3: The first 15 input features for the '> 1 b tag' category in the $\mu\tau_h$ final state. The signal (background) class is represented by the distributions in blue (orange).

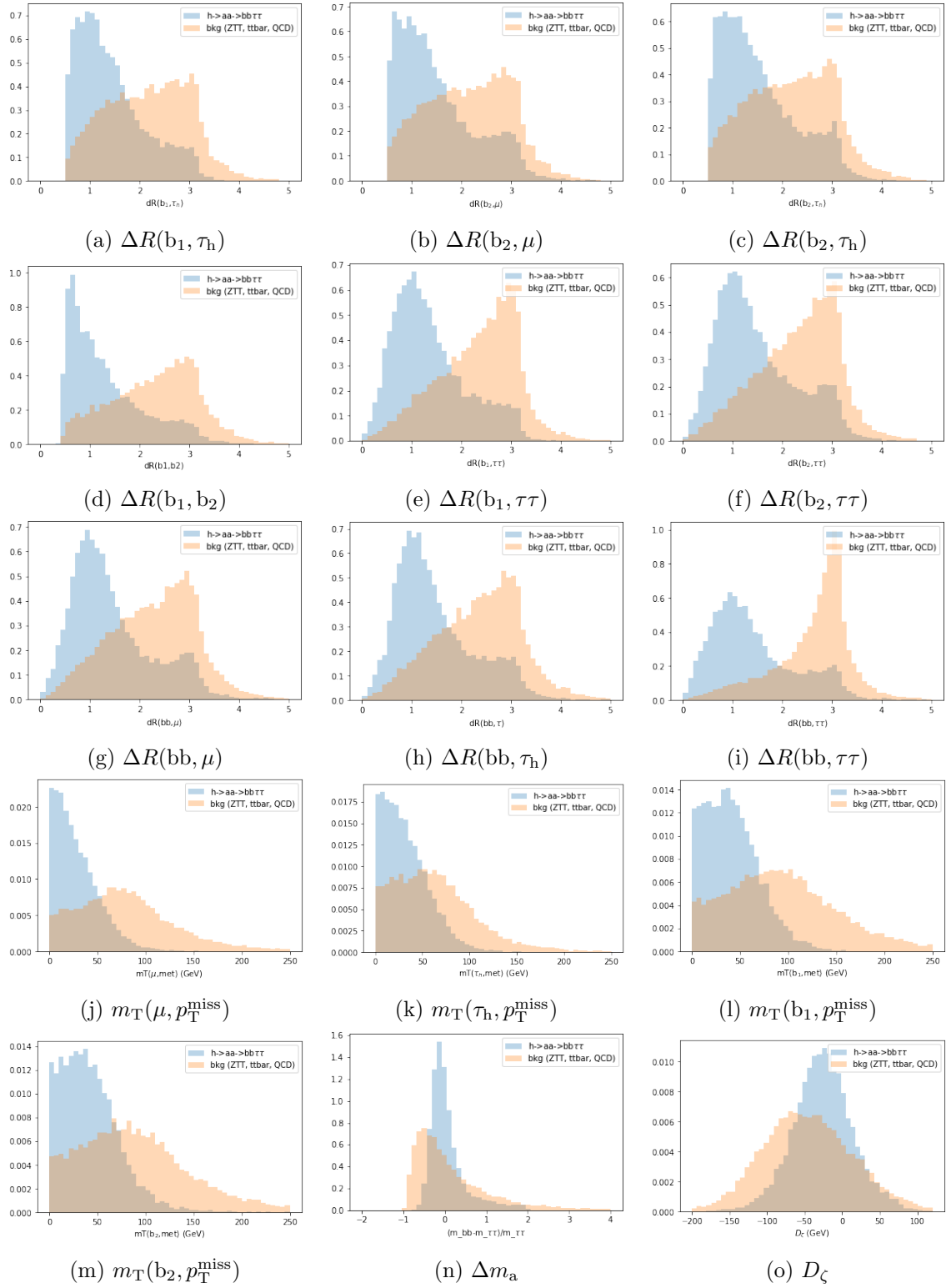
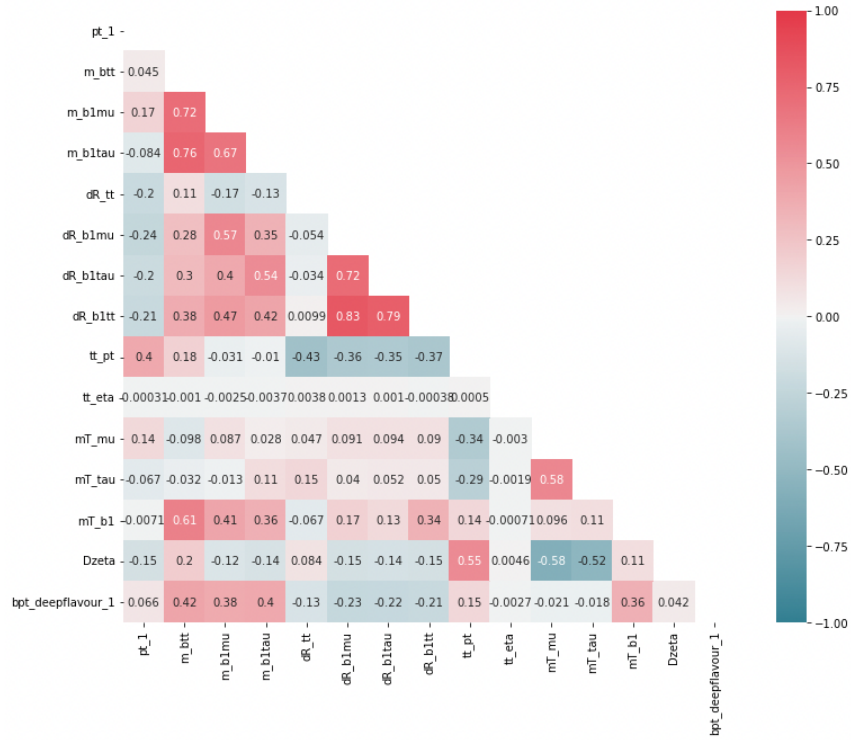
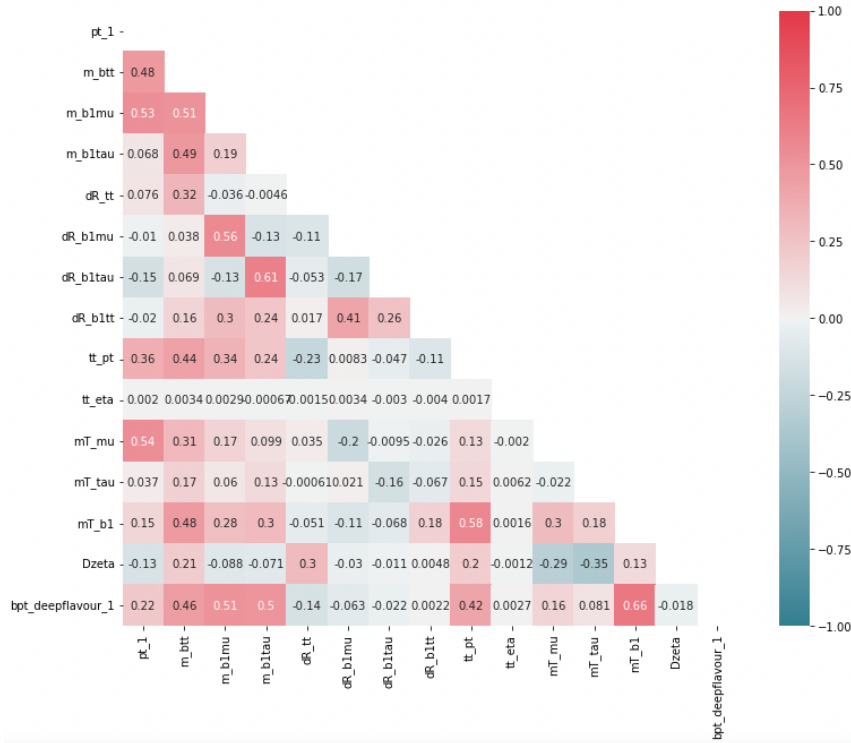


Figure 3.4: The last 15 input features for the '> 1 b tag' category in the $\mu\tau_h$ final state. The signal (background) class is represented by the distributions in blue (orange).

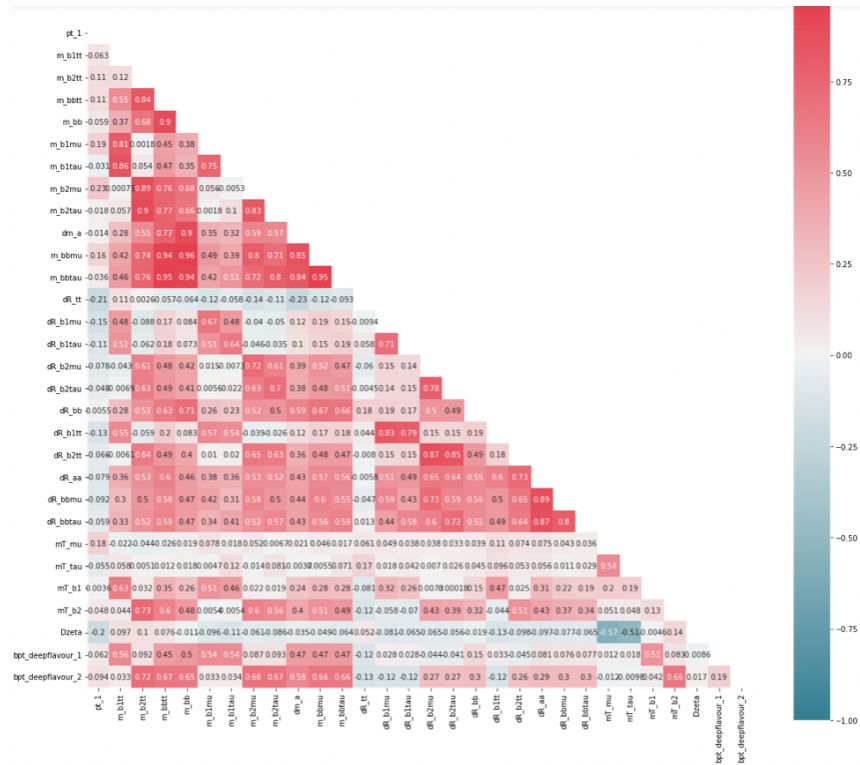


(a) Input correlation in signal events.

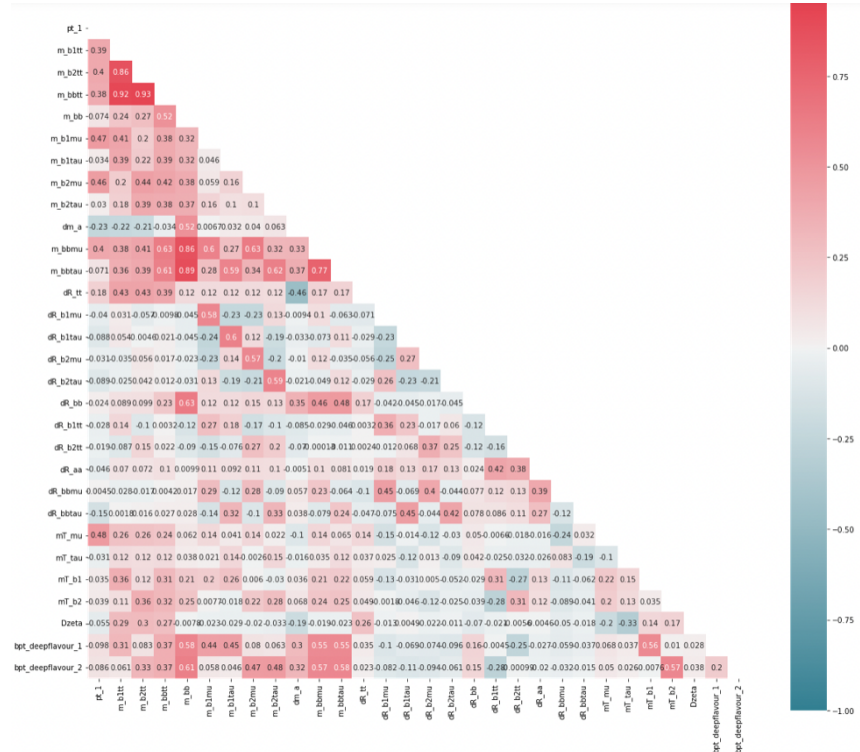


(b) Input correlation in background events.

Figure 3.5: Correlation matrices for the 15 input features for the ‘1 b tag’ category in the $\mu\tau_h$ final state, evaluated with the signal and background events, respectively.



(a) Input correlation in signal events.



(b) Input correlation in background events.

Figure 3.6: Correlation matrices for the 30 input features for the ‘> 1 b tag’ category in the $\mu\tau_h$ final state, evaluated with the signal and background events, respectively.

Model and training

The DNN models are implemented and trained using the `Keras` library [23]. The fundamentals of neural networks are discussed in more details in Part. II. Essentially, a fully-connected neural network [89–93] consists of three components: 1) input layer, 2) hidden layers, and 3) output layer. The number of nodes in the input layer equals the number of input features. The number of hidden layers and the number of nodes per hidden layer are hyperparameters that can be varied and optimized for different problems. Nodes between adjacent layers are connected with model weights, which are used to perform matrix multiplications when data flow from one layer to the next, followed by nonlinear activation at each node. The rectified linear unit (ReLU) [94] function is used as activation for nodes in the hidden layers:

$$\text{ReLU}(x) = \max(0, x), \quad (3.5)$$

which is nonlinear while not being computationally intensive. The output layer consists of one node for binary classification, which can be constrained to output a probability-like value between 0 and 1 by using the sigmoid function as activation:

$$\sigma(x) = \frac{1}{1 + e^{-x}}. \quad (3.6)$$

For a binary classifier trained with supervised learning, training data are labeled: the signal class is labeled as $y = 1$ and the background class is labeled as $y = 0$. Given an event with an input vector \mathbf{x} , a DNN can be constructed as a function that predicts the probability for the event being a signal: $\hat{y} = f(\mathbf{x}; \mathbf{w})$, where \mathbf{w} denotes the model weights. The model weights \mathbf{w} are learned from the labeled training data by minimizing an objective function called the loss function $l(\hat{y}, y)$, which typically measures the distance between the model prediction \hat{y} and the label y . A gradient-based optimization algorithm called `Adam` [95] is used to find the optimal model weights for the training dataset $\{(\mathbf{x}_i, y_i)\}_{i=1, \dots, n}$.

with inputs \mathbf{x}_i and labels y_i :

$$\mathbf{w}^* = \arg \min_{\mathbf{w}} \sum_{i=1}^n l(f(\mathbf{x}_i; \mathbf{w}), y_i), \quad (3.7)$$

where the binary cross-entropy loss [93] is used:

$$l(\hat{y}, y) = -\left(y \log \hat{y} + (1 - y) \log(1 - \hat{y})\right). \quad (3.8)$$

If a training event is a signal ($y = 1$), the loss becomes $l = -\log \hat{y}$, which is zero if the model makes a perfectly correct prediction ($\hat{y} = 1$), and penalizes with positive infinity loss if the model makes a perfectly wrong prediction ($\hat{y} = 0$). Conversely, for a background event ($y = 0$), the loss is $-\log(1 - \hat{y})$. Therefore, the model learns to output a value as close to 1 (0) as possible for a signal (background) event, effectively computing the signal probability for a given event.

The input features are standardized so that each of them has a zero mean and unit variance. This ensures all inputs are on the same scale for faster convergence in the optimization process. The training data are randomly split into three sets: the training set ($\approx 60\%$), the validation set ($\approx 20\%$), and the test set ($\approx 20\%$). The training set is used for training, where the model weights are optimized with respect to it. The validation set is used for independent evaluation concurrently at train time for monitoring purposes. The test set is used for independent evaluation after the model is trained.

If the model size is too large, it is prone to overfitting. Overfitting occurs when the model is too expressive and fits to noisy fluctuations and outliers in the training dataset, similar to fitting a 10-order polynomial to parabolic data. During training, overfitting is indicated by a much lower training loss compared to validation loss. Overfitting prevents the model from generalizing to unseen data, such as the test set. Dropout [96] is a common technique to reduce overfitting, which randomly selects nodes from a layer and sets them to zero during training, ensuring no specific nodes become overly important. A dropout layer is applied to each hidden layer in the models, with the dropout rate being

Table 3.4: Summary of the DNN architecture. All DNNs have two hidden layers, each followed by a dropout layer. The numbers of nodes in the input and hidden layers, as well as the dropout rates, are shown.

	Input dim.	Layer-1 dim.	Layer-2 dim.	Dropout rate for layer-1	Dropout rate for layer-2
$\mu\tau_h$ ‘1 b tag’	15	52	36	0.2	0.2
$\mu\tau_h$ ‘> 1 b tag’	30	62	32	0.3	0.4
$e\tau_h$ ‘1 b tag’	15	52	36	0.2	0.2
$e\tau_h$ ‘> 1 b tag’	30	58	34	0.3	0.4
$e\mu$ ‘1 b tag’	16	52	34	0.4	0.3
$e\mu$ ‘> 1 b tag’	31	56	38	0.2	0.3

a hyperparameter. Conversely, if the model size is too small, it is prone to underfitting, failing to learn useful information from the data.

To find the optimal model architecture, the **Hyperband** algorithm [97] is used in the **KerasTuner** framework [98]. This algorithm employs an adaptive resource allocation strategy to efficiently perform hyperparameter search across a large set of random model configurations. The hyperparameters include the number of hidden layers, the number of nodes per hidden layer, and the dropout rate per hidden layer. The optimized hyperparameters are summarized in Tab. 3.4 for the DNNs in the six categories.

The training curves are shown in Fig. 3.7. Training is set to stop early when there is no significant gain or when the performance on the training set becomes better than on the validation set.

Performance and Validation

The distributions of the DNN output (score) are shown in Fig. 3.8. As expected, the signal events tend to have score values close to one, while the background events tend to have score values close to zero. The distributions for the test set are consistent with those for the training and validation sets, suggesting that the model is not over-trained and can generalize its performance to unseen data. The receiver operating characteristic (ROC) curves are shown in Fig. 3.9.

A ROC curve measures the true positive rate (signal efficiency) as a function of the false positive rate (background efficiency). Every point on the ROC curve corresponds to a threshold value on the score: for a given event, if its score is higher than the threshold, it is predicted as signal; otherwise, it is predicted as background. The area under the curve

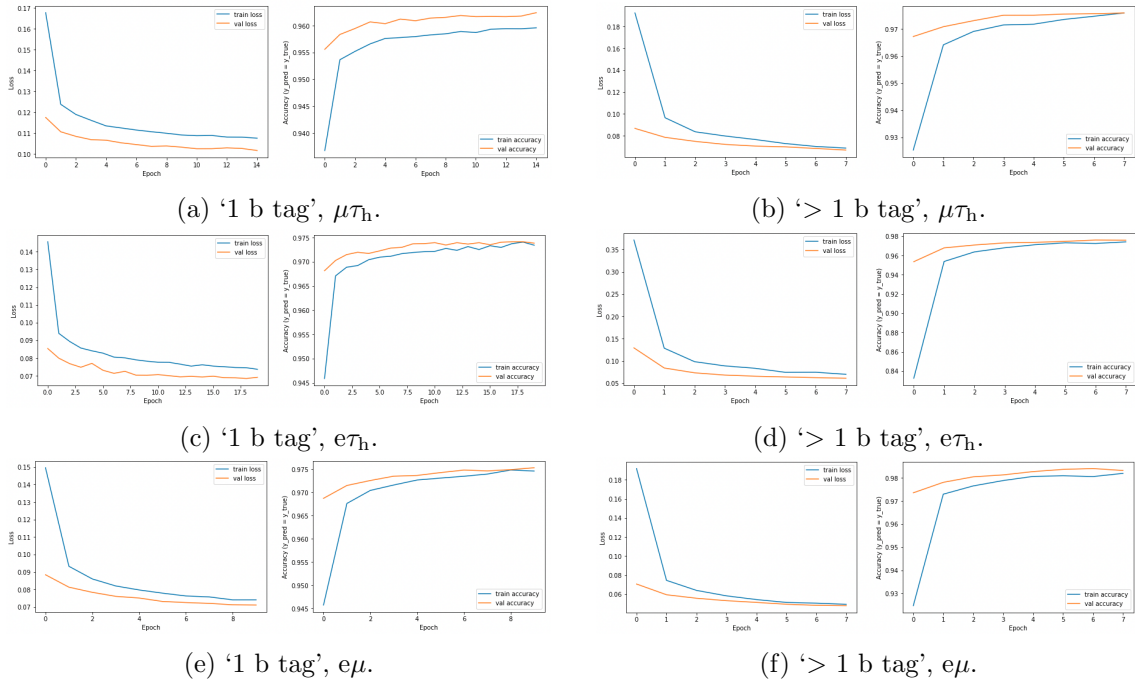


Figure 3.7: In each category, the loss as a function of the training epoch is plotted on the left, while the accuracy as a function of the training epoch is plotted on the right. The blue (orange) curves represent evaluation on the training (validation) set.

Table 3.5: Classification accuracy (with a score threshold of 0.5) evaluated on the test set.

	Accuracy (%)
$\mu\tau_h$ '1 b tag'	96.2
$\mu\tau_h$ '> 1 b tag'	97.5
$e\tau_h$ '1 b tag'	97.4
$e\tau_h$ '> 1 b tag'	97.4
$e\mu$ '1 b tag'	97.6
$e\mu$ '> 1 b tag'	98.1

(AUC) quantifies the ROC curve. The AUC is higher than 0.99 in all six categories.

Tab. 3.5 lists the classification accuracy, measured as the probability of the classifier making the correct classification. In this example, the decision threshold is set at a score value of 0.5.

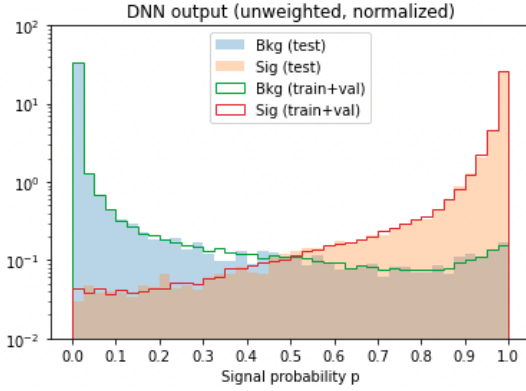
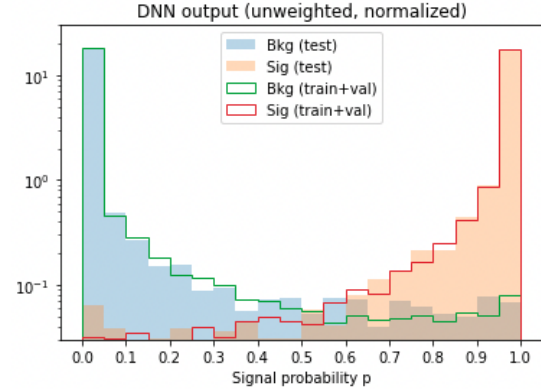
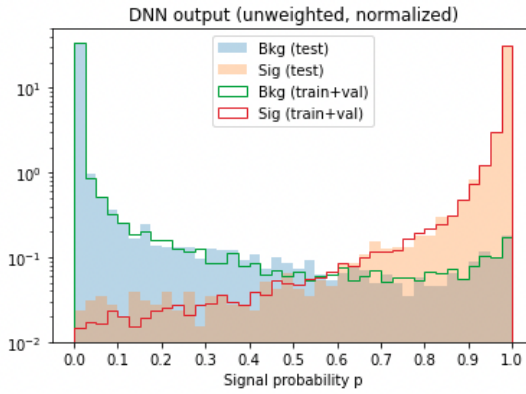
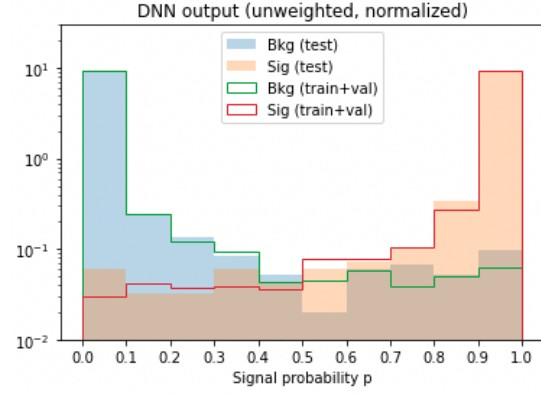
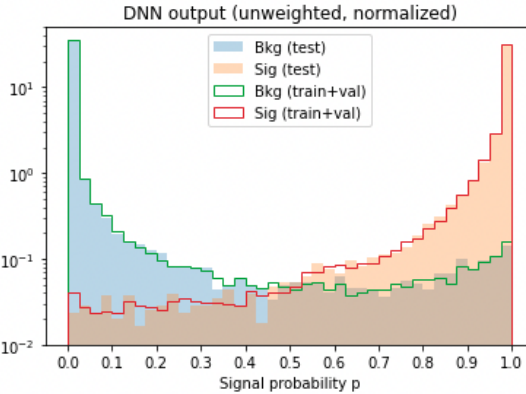
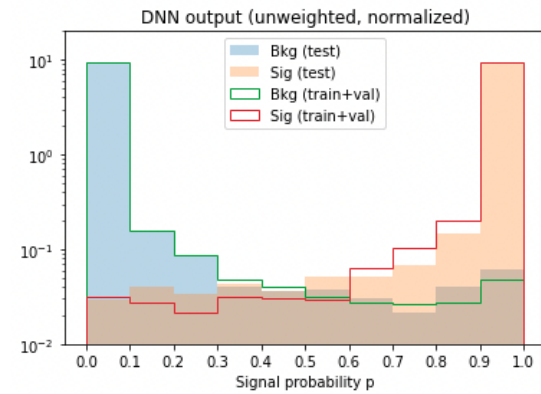
(a) '1 b tag', $\mu\tau_h$.(b) '> 1 b tag', $\mu\tau_h$.(c) '1 b tag', $e\tau_h$.(d) '> 1 b tag', $e\tau_h$.(e) '1 b tag', $e\mu$.(f) '> 1 b tag', $e\mu$.

Figure 3.8: DNN score distributions in the six categories. The box histograms in red (green) represent the signal (background) class evaluated on the training and validation sets combined. The filled histograms in orange (blue) represent the signal (background) class evaluated on the test set.

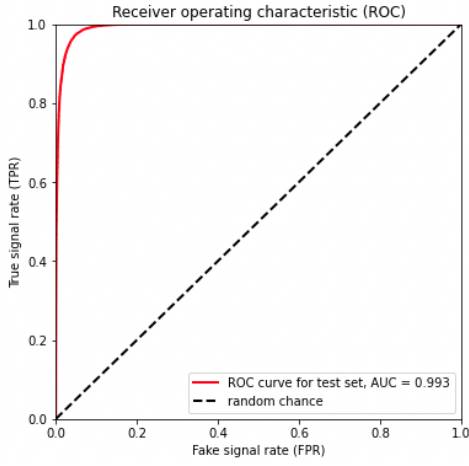
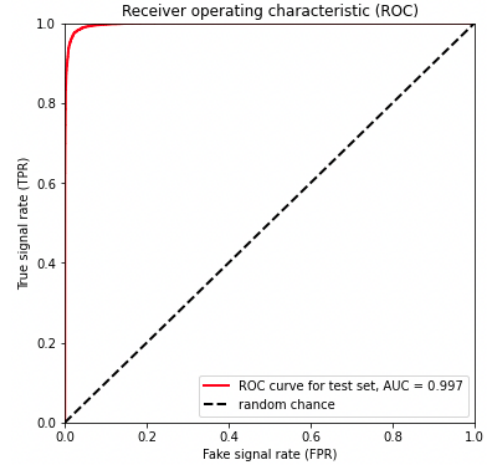
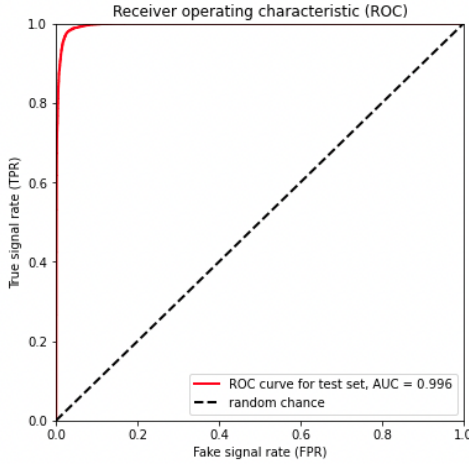
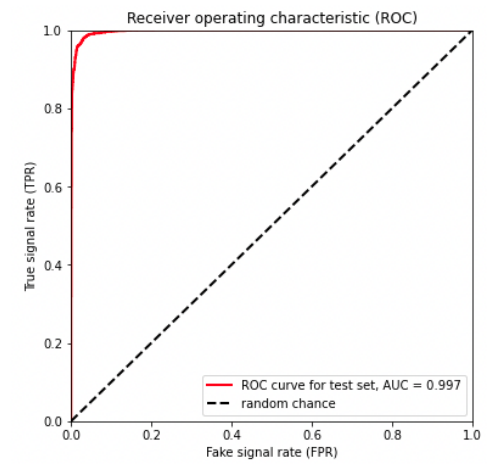
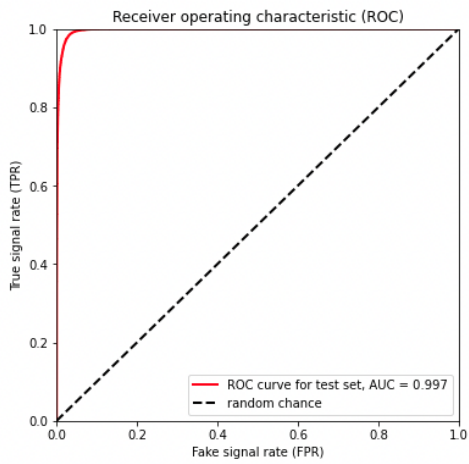
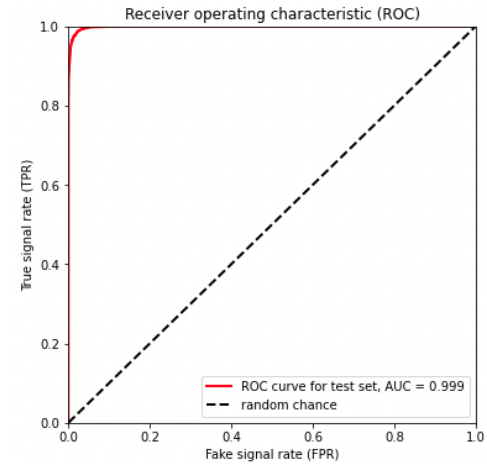
(a) '1 b tag', $\mu\tau_h$.(b) '> 1 b tag', $\mu\tau_h$.(c) '1 b tag', $e\tau_h$.(d) '> 1 b tag', $e\tau_h$.(e) '1 b tag', $e\mu$.(f) '> 1 b tag', $e\mu$.

Figure 3.9: ROC curves in the six categories, evaluated on the test set. The area under the curve (AUC) is indicated. The dotted lines represent a classifier that does not learn and makes predictions by random chance.

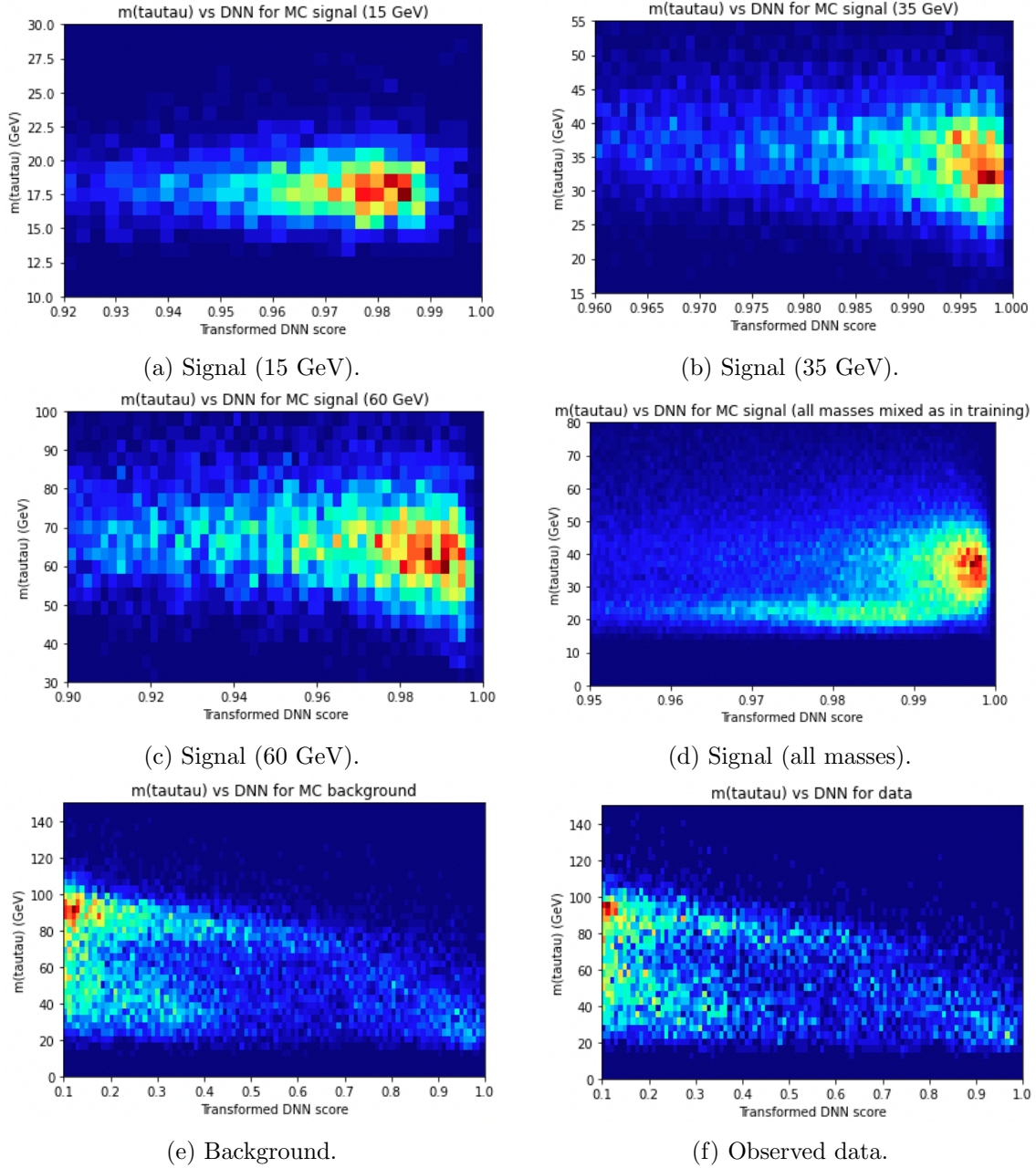


Figure 3.10: 2D correlation plots of $m_{\tau\tau}$ vs. DNN score, for the $\mu\tau_h$ ‘1 b tag’ category. Signal, background, and observed data are plotted separately.

The DNN score and $m_{\tau\tau}$ are validated to be uncorrelated, as shown in Fig. 3.10.

Categorization

The DNN scores are used to define subregions for $m_{\tau\tau}$ in each of the six categories. Each category typically has several signal regions (SRs) with high signal-to-background

ratios to enhance sensitivity, and one control region (CR) that is signal-depleted and background-enriched to validate the background estimation. The SRs (CRs) are defined by events with high (low) DNN score.

However, as shown in Fig. 3.8, the score distributions are highly concentrated around the score boundaries: signal events are piled up at the rightmost bin while background events are piled up at the leftmost bin. This concentration can potentially lead to impractical threshold definitions. For instance, the overall sensitivity could change dramatically when the SRs are redefined with a score threshold changing from 0.99999 to 0.99998.

Therefore, the DNN score is first transformed to dilute the density of the high score region, using the property of the tanh function:

$$\tilde{p}_k = \tanh^{-1}(p \times \tanh(k))/k, \quad (3.9)$$

where $\tilde{p}_k \in (0, 1)$ is the transformation of the raw DNN score $p \in (0, 1)$, and $k > 0$ is a parameter that controls the dilution. The parameter is set to $k = 1.5$ for the $\mu\tau_h$ and $e\tau_h$ final states, and $k = 2.5$ for the $e\mu$ final state.

After the transformation, the DNN scores are plotted, comparing the observed data with the expected signal and background distributions with event weights applied, as shown in Fig. 3.11.

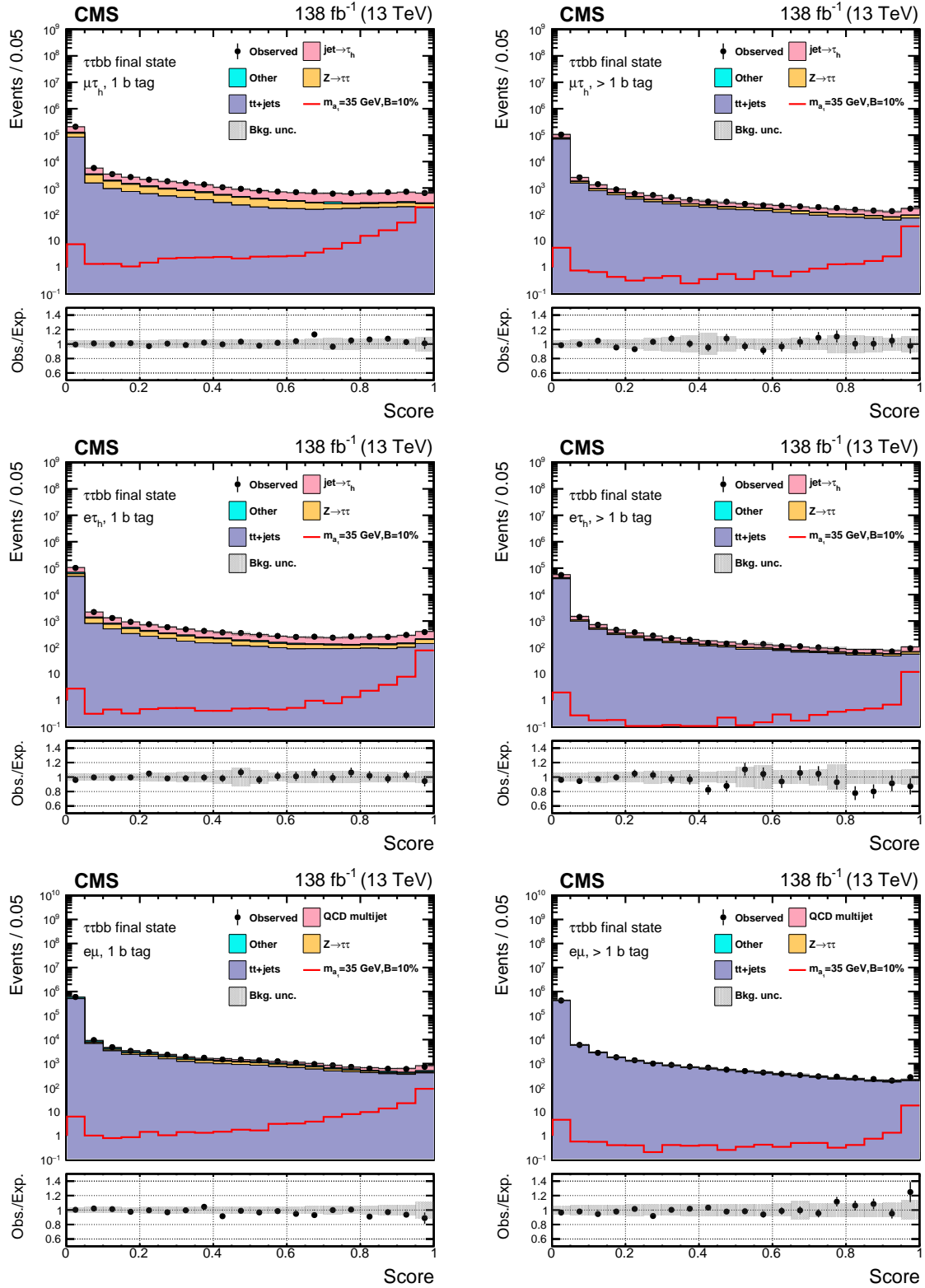


Figure 3.11: Pre-fit distributions of the DNN score with the category indicated, comparing the weighted background events and the observed data. The total background uncertainty is shown in grey. The lower panel shows the ratio of the observed data to the expected background yields. The figures are taken from Ref. [15].

These scores are used to define various subregions using a grid-scan strategy described in the following steps, performed separately in each category per data-taking year.

1. Starts from the first threshold and insert it in the high score region, say a score value of 0.99. This defines two subregions: the first SR called SR1 (score > 0.99), and the CR (score < 0.99).
2. Compute the expected exclusion limit (see Sec. 3.1.9) combining SR1 and CR, which quantifies the sensitivity of the signal extraction.
3. Change the threshold to a lower score value in small grid points and repeat the computation of the expected exclusion limit. This generates the expected exclusion limit as a function of the score threshold.
4. Fix the threshold to the score value that gives the minimum expected exclusion limit among the grid points.
5. Insert a new threshold and repeat the scanning procedures while keeping the first threshold fixed. This creates three subregions: SR1, SR2, and CR.
6. Keep inserting new thresholds until there is no additional gain in sensitivity.

This strategy is an approximation to search from all the possible combinations of thresholds, as only one threshold is varied at a time while keeping others fixed. To validate the thresholds obtained from the above steps, some thresholds are scanned again but in a reverse order. Thresholds are updated if the sensitivity improves. The final subregion definitions are summarized in Tab. 3.6. Threshold values can differ between different data-taking periods because of varying trigger paths, causing slight differences in the respective distributions.

3.1.8 Systematic uncertainties

In analyses, uncertainties quantify the reliability and precision of predictions and measurements. There are two types of uncertainties: statistical and systematic. Statistical

Table 3.6: Definition of the subregions for $m_{\tau\tau}$ by threshold cuts on the DNN score.

	‘1 b tag’				‘> 1 b tag’		
	SR1	SR2	SR3	CR	SR1	SR2	CR
$\mu\tau_h$ 2018	> 0.98	$\in (0.95, 0.98)$	$\in (0.9, 0.95)$	< 0.9	> 0.99	$\in (0.96, 0.99)$	< 0.96
$\mu\tau_h$ 2017	> 0.97	$\in (0.94, 0.97)$	$\in (0.9, 0.94)$	< 0.9	> 0.98	$\in (0.94, 0.98)$	< 0.94
$\mu\tau_h$ 2016	> 0.97	$\in (0.94, 0.97)$	$\in (0.89, 0.94)$	< 0.89	> 0.97	$\in (0.93, 0.97)$	< 0.93
$e\tau_h$ 2018	> 0.97	$\in (0.945, 0.97)$	$\in (0.9, 0.945)$	< 0.9	> 0.96	—	< 0.96
$e\tau_h$ 2017	> 0.985	$\in (0.965, 0.985)$	$\in (0.93, 0.965)$	< 0.93	> 0.985	—	< 0.985
$e\tau_h$ 2016	> 0.985	$\in (0.965, 0.985)$	$\in (0.93, 0.965)$	< 0.93	> 0.96	—	< 0.96
$e\mu$ 2018	> 0.99	$\in (0.95, 0.99)$	$\in (0.85, 0.95)$	< 0.85	> 0.98	$\in (0.94, 0.98)$	< 0.94
$e\mu$ 2017	> 0.985	$\in (0.95, 0.985)$	$\in (0.85, 0.95)$	< 0.85	> 0.97	$\in (0.93, 0.97)$	< 0.93
$e\mu$ 2016	> 0.99	$\in (0.95, 0.99)$	$\in (0.85, 0.95)$	< 0.85	> 0.98	$\in (0.94, 0.98)$	< 0.94

uncertainties refer to the random fluctuations in a measurement process arising from the limited sample size, which can be improved by increasing the sample size. Systematic uncertainties refer to those that consistently impact the result and cannot be improved by increasing the sample size. These can arise from sources such as theoretical modeling, detector effects, background estimation methodologies, and selection criteria.

The systematic uncertainties are listed in the following.

- **Trigger efficiencies.** An uncertainty of 2% is associated with triggers without a τ_h requirement, and 5-10% for triggers with a τ_h requirement. These are shape uncertainties applied to events according to their triggered paths.
- **Luminosity.** Uncertainties associated with luminosity measurements are applied as normalization uncertainties to simulated events, amounting to 1, 2, and 1.5% for the 2016, 2017, and 2018 data-taking periods, respectively, along with additional correlated uncertainties between the periods [99–101].
- **Lepton reconstruction and identification efficiencies.** A normalization uncertainty of 2% is associated with electron [57] and muon [102], respectively.
- **Electron energy.** Uncertainties associated with the electron energy scale depend on the electron p_T and η [103]. These are shape uncertainties, and as energy variations, they can cause event migration between subregions.

- **Muon energy.** Uncertainties associated with the muon energy scale depend on the muon p_T and η , varying between 0.4-2.7% as shape uncertainties [10].
- **τ_h reconstruction and identification efficiencies.** Uncertainties associated with these efficiencies depend on the τ_h p_T and η , varying between 3-5% [64]. They are applied as shape uncertainties separately on τ_h matched to a genuine τ_h , electron, and muon, respectively, at the generator level.
- **τ_h energy** Uncertainties associated with the τ_h energy scale depend on the τ_h p_T and decay mode, varying between 0.2-1.1% [64]. They are applied as shape uncertainties separately on τ_h matched to a genuine τ_h , electron, and muon, respectively, at the generator level.
- **Jet energy.** Uncertainties associated with the jet energy corrections are applied as a function of the jet p_T and η [61]. They cause shape variations ranging from 10-15%. Uncertainties associated with the jet energy resolution are also applied.
- **Missing transverse energy** Shape uncertainties are associated with the recoil corrections and the unclustered energy measurements.
- **B-tagging efficiencies.** There are several sources of uncertainty that contribute to the efficiency measurements: jet energy scale uncertainties, contamination of light-flavor jets in heavy-flavor jet regions, contamination of heavy-flavor jets in light-flavor jet regions, and statistical uncertainties [13]. These uncertainties are applied according to the simulated jet properties and are taken as shape uncertainties.
- **$t\bar{t}$ +jets background estimation.** There is a normalization uncertainty of 4.2% on the production cross section [104]. Shape uncertainties are associated with the parton shower uncertainties and the variations of the renormalization and factorization scales.
- **Embedded $Z \rightarrow \tau\tau$ background estimation.** A normalization uncertainty of 4% is applied [85]. Shape uncertainties are associated with the tau tracking efficiencies

and the event contamination from non-DY and non-Higgs processes.

- **Fake background estimation.** For the QCD multijet background in the $e\mu$ final state, a normalization uncertainty of 20% accounts for the limited statistics in the estimation from the sideband regions. The scale factors obtained from fits have associated shape uncertainties. For the $\text{jet} \rightarrow \tau_h$ background in the $\mu\tau_h$ and $e\tau_h$ final states, since the fake rate is measured only from the $Z \rightarrow \mu\mu + \text{jets}$ events but generalized to all other possible background sources such as $W + \text{jets}$ and QCD multijet, a normalization uncertainty of 20% is applied for that. There are also shape uncertainties associated with the fake rate measurements.
- **Signal modeling.** Uncertainties from the perturbative QCD calculations, PDFs, and the strong coupling constant (α_s) contribute to the SM Higgs boson production cross sections [1]. These uncertainties affect the overall signal yields with negligible shape effects and are accounted for by a normalization uncertainty of 3.6% on the sum of the ggF and VBF Higgs boson production cross sections.
- **Other background estimations.** In $Z \rightarrow ee/\mu\mu$ samples, an uncertainty of 10% is associated with the Z p_T reweighting factor, taken as a shape uncertainty. Normalization uncertainties on the production cross sections amount to 5, 5, and 2% for diboson, single top, and DY samples, respectively [104]. Normalization uncertainties on the production cross sections and the branching ratios for the SM Higgs boson background are also accounted for [1].

The normalization uncertainties refer to uncertainties that change the overall event yields while keeping the shapes of the $m_{\tau\tau}$ distributions unchanged, whereas the shape uncertainties change the shapes. Uncertainties that affect any of the input features to the DNNs are propagated to the DNN scores correspondingly.

3.1.9 Signal extraction

With a signal and background model that predicts the expected event yields along with uncertainty descriptions, parameters producing the most compatible model with the observed data are obtained using a statistical method called a maximum likelihood fit [105]. A binned maximum likelihood fit is performed based on the $m_{\tau\tau}$ distributions in all sub-regions and final states simultaneously.

Starting from the binned likelihood function that describes the compatibility between the observed data and the predictions:

$$\mathcal{L}(\text{data}|\mu, \boldsymbol{\theta}) = \prod_i \text{Poisson}(n_i|\mu s_i(\boldsymbol{\theta}) + b_i(\boldsymbol{\theta}))p(\tilde{\boldsymbol{\theta}}|\boldsymbol{\theta}). \quad (3.10)$$

Here, the product runs over all the bins of the distributions of the fit observable, $m_{\tau\tau}$, where the i -th bin follows a Poisson distribution with observed data n_i and expected yields from signal μs_i and background b_i in that bin:

$$\text{Poisson}(n_i|\mu s_i(\boldsymbol{\theta}) + b_i(\boldsymbol{\theta})) = \frac{e^{-(\mu s_i(\boldsymbol{\theta}) + b_i(\boldsymbol{\theta}))} (\mu s_i(\boldsymbol{\theta}) + b_i(\boldsymbol{\theta}))^{n_i}}{n_i!} \quad (3.11)$$

The signal strength is controlled by the parameter μ . Both signal and background yields are parameterized by the nuisance parameters $\boldsymbol{\theta}$, which control the variations in the systematic uncertainties. The nuisance parameters are constrained through the probability density functions $p(\boldsymbol{\theta}|\tilde{\boldsymbol{\theta}}) \propto p(\tilde{\boldsymbol{\theta}}|\boldsymbol{\theta})$, where $\tilde{\boldsymbol{\theta}}$ represent the nominal values.

According to the nature and properties of the uncertainty, a probability density function is assigned to each nuisance parameter. Nuisance parameters representing normalization uncertainties are assigned a log-normal probability density function. Nuisance parameters representing shape uncertainties are assigned a Gaussian distribution, requiring two shape templates corresponding to ± 1 standard deviation of variations as inputs for the shape interpolation. The statistical uncertainties due to limited sample size are accounted for by a Poisson nuisance parameter per bin [106].

A binned maximum likelihood fit is performed in the background-only hypothesis ($\mu =$

0) using the CMS statistical analysis and combination tool called **COMBINE** [107]. The fitted $m_{\tau\tau}$ distributions are shown: $\mu\tau_h$ ‘1 b tag’ in Fig. 3.12, $\mu\tau_h$ ‘> 1 b tag’ in Fig. 3.13, $e\tau_h$ ‘1 b tag’ in Fig. 3.14, $e\tau_h$ ‘> 1 b tag’ in Fig. 3.15, $e\mu$ ‘1 b tag’ in Fig. 3.16, and $e\mu$ ‘> 1 b tag’ in Fig. 3.17.

The statistical uncertainties dominate over the systematic uncertainties in the analysis due to the small sample size available. Among the systematic uncertainties, the dominant ones include the signal modeling, the normalization uncertainties in the fake background yields, and the uncertainties in the $t\bar{t}$ +jets cross section. Overall, the observed data is consistent with the SM predictions in all subregions and final states.

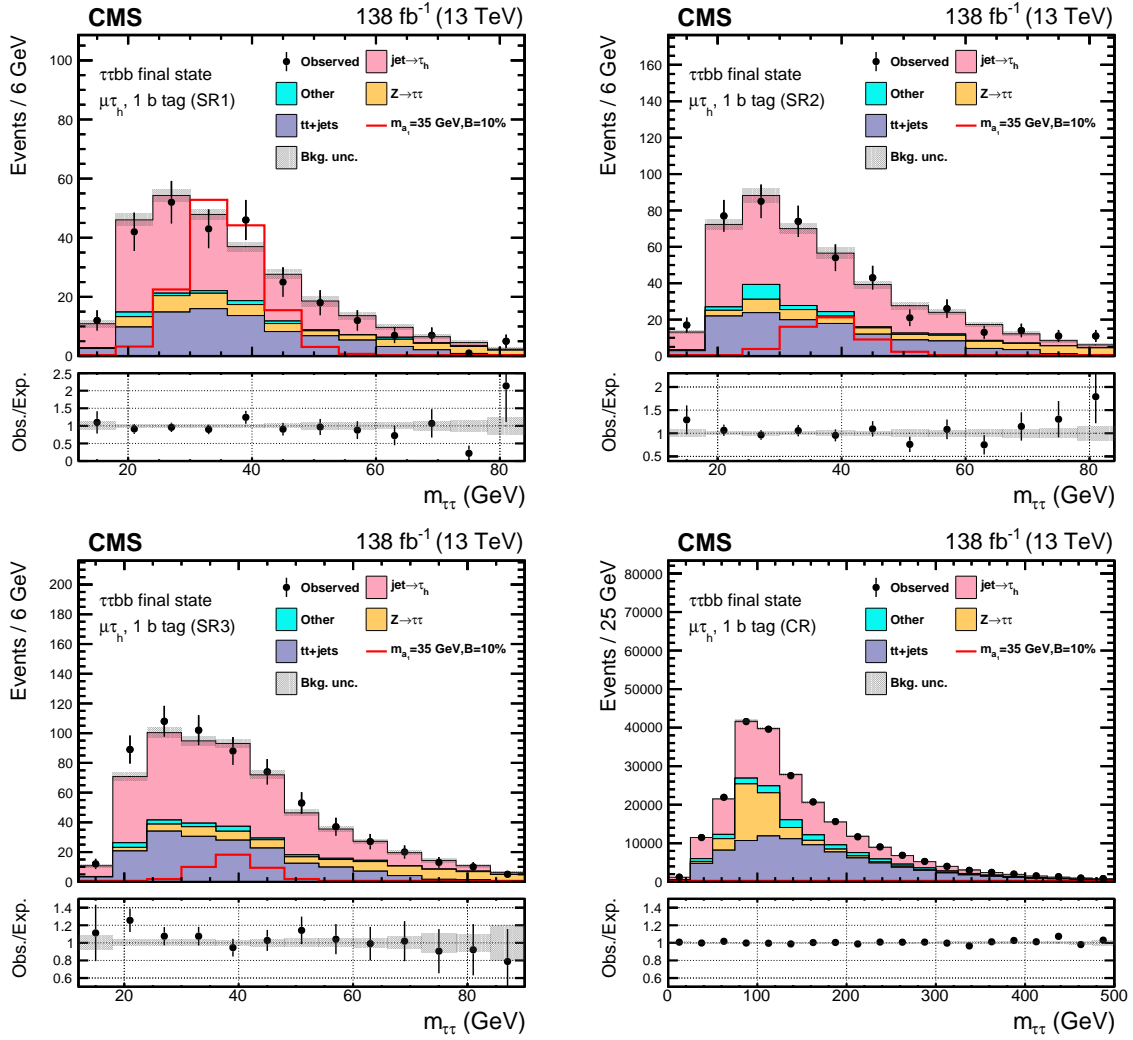


Figure 3.12: Post-fit distributions of $m_{\tau\tau}$ for the $\mu\tau_h$ '1 b tag' category, with the subregion indicated. The signal (35 GeV) is shown as a box histogram in red, assuming the branching ratio $B(h \rightarrow aa \rightarrow \tau\tau b\bar{b}) = 10\%$. The figures are taken from Ref. [15].

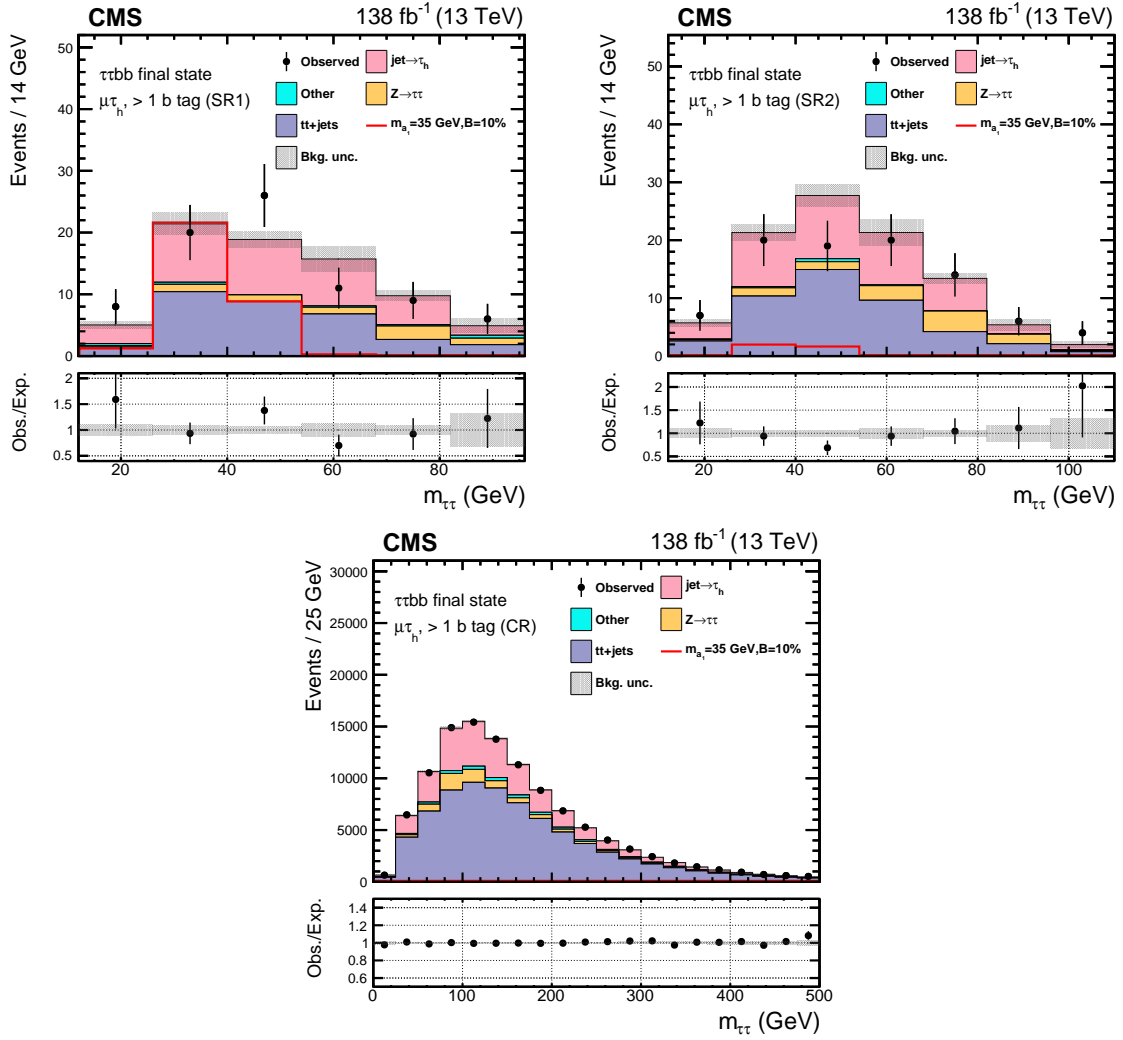


Figure 3.13: Post-fit distributions of $m_{\tau\tau}$ for the $\mu\tau_h$ ' > 1 b tag' category, with the subregion indicated. The signal (35 GeV) is shown as a box histogram in red, assuming the branching ratio $B(h \rightarrow aa \rightarrow \tau\tau bb) = 10\%$. The figures are taken from Ref. [15].

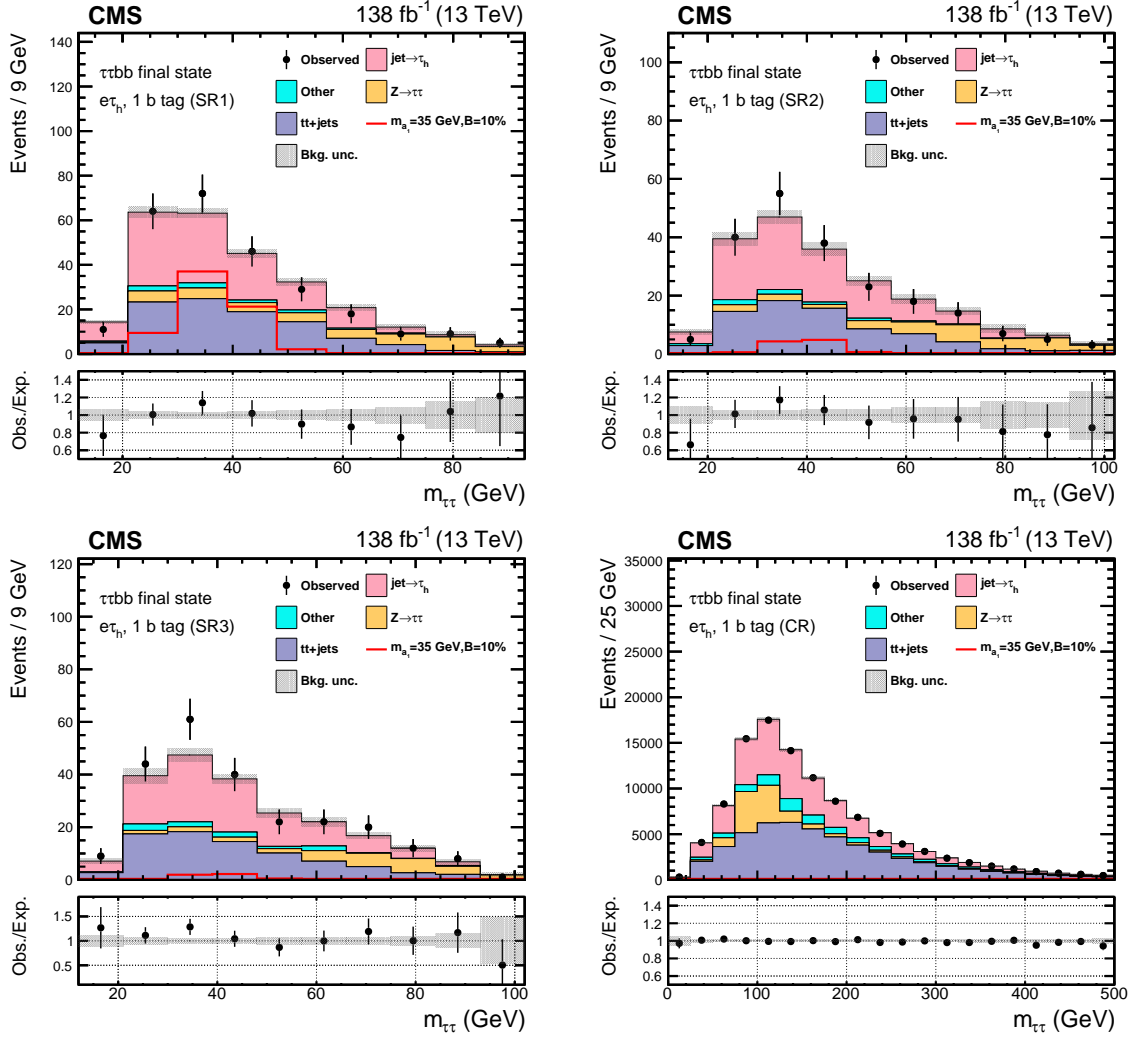


Figure 3.14: Post-fit distributions of $m_{\tau\tau}$ for the $e\tau_h$ '1 b tag' category, with the subregion indicated. The signal (35 GeV) is shown as a box histogram in red, assuming the branching ratio $B(h \rightarrow aa \rightarrow \tau\tau b\bar{b}) = 10\%$. The figures are taken from Ref. [15].

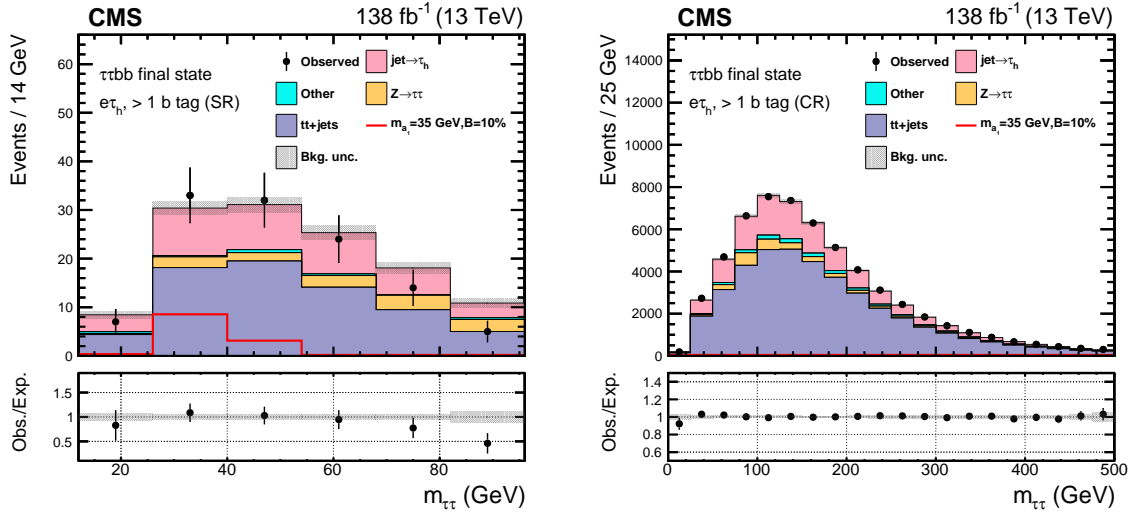


Figure 3.15: Post-fit distributions of $m_{\tau\tau}$ for the $e\tau_h$ '> 1 b tag' category, with the sub-region indicated. The signal (35 GeV) is shown as a box histogram in red, assuming the branching ratio $B(h \rightarrow a\bar{a} \rightarrow \tau\tau b\bar{b}) = 10\%$. The figures are taken from Ref. [15].

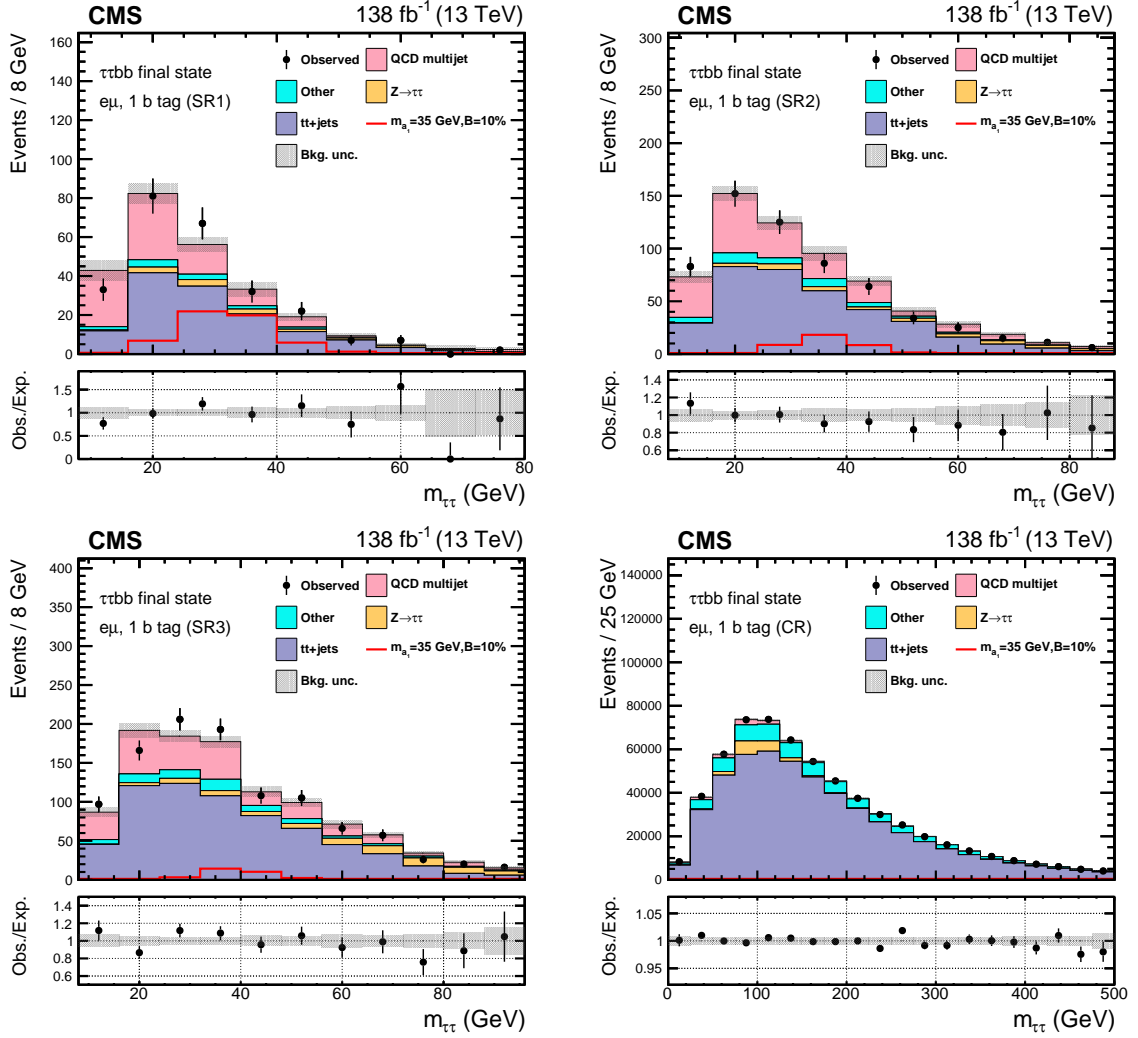


Figure 3.16: Post-fit distributions of $m_{\tau\tau}$ for the $e\mu$ '1 b tag' category, with the subregion indicated. The signal (35 GeV) is shown as a box histogram in red, assuming the branching ratio $B(h \rightarrow aa \rightarrow \tau\tau bb) = 10\%$. The figures are taken from Ref. [15].

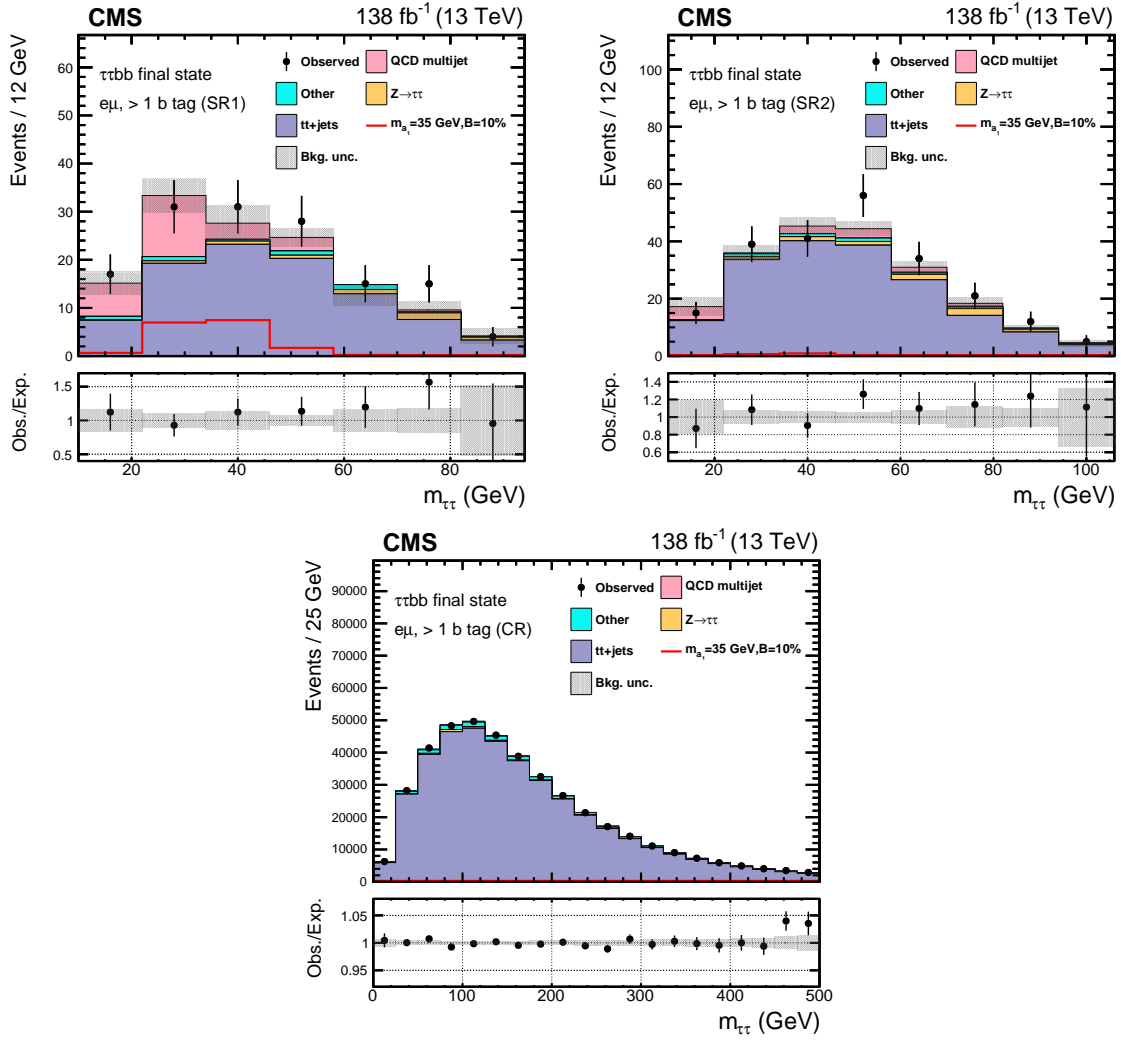


Figure 3.17: Post-fit distributions of $m_{\tau\tau}$ for the $e\mu$ ' > 1 b tag ' category, with the subregion indicated. The signal (35 GeV) is shown as a box histogram in red, assuming the branching ratio $B(h \rightarrow aa \rightarrow \tau\tau b\bar{b}) = 10\%$. The figures are taken from Ref. [15].

3.1.10 Exclusion limits and model interpretation

Since no excess of events over the SM predictions are observed, an upper bound can be set to exclude higher signal strength values statistically incompatible with the observation, called exclusion limits, using the modified frequentist CL_s technique in the asymptotic approximation [105, 108–110].

The signal-plus-background hypothesis (s+b) and the background-only (b-only) hypothesis are compared using the test statistic:

$$q_\mu = -2 \ln \frac{\mathcal{L}(\text{data}|\mu, \hat{\boldsymbol{\theta}}_\mu)}{\mathcal{L}(\text{data}|\hat{\mu}, \hat{\boldsymbol{\theta}})}, \quad (3.12)$$

where $\hat{\mu}$ and $\hat{\boldsymbol{\theta}}$ globally maximize \mathcal{L} , $\hat{\boldsymbol{\theta}}_\mu$ maximize \mathcal{L} conditional in μ at which q_μ is computed, and $0 \leq \hat{\mu} \leq \mu$ is required to ensure a non-negative signal strength and a one-sided confidence interval. To derive exclusion limits, the CL_s criterion is used:

$$\text{CL}_s = \frac{\text{CL}_{s+b}}{\text{CL}_b} = \frac{p(q_\mu > q_\mu^{\text{obs}}|s+b)}{p(q_\mu > q_\mu^{\text{obs}}|b\text{-only})} = \alpha, \quad (3.13)$$

where $p(q_\mu > q_\mu^{\text{obs}}|s+b)$ and $p(q_\mu > q_\mu^{\text{obs}}|b)$ are the probabilities of getting a q_μ value larger than the observed value q_μ^{obs} under the two hypotheses, respectively, and α sets the confidence level (CL) at $1 - \alpha$ for excluding the signal strength value larger than μ . Typically, exclusion limits are set at 95% CL ($\alpha = 0.05$) for LHC results.

The observed and expected exclusion limits at 95% CL on the signal branching ratio $\text{B}(h \rightarrow aa \rightarrow \tau\tau b\bar{b})$ are shown in Fig. 3.18 for the three $\tau\tau$ final states and their combination. Limits are higher for high-mass pseudoscalars because the signal has more overlap with the background in the $m_{\tau\tau}$ distributions. In the $\mu\tau_h$ and $e\tau_h$ final states, limits are higher for low-mass pseudoscalars because they become boosted with their decay products starting to merge and overlap with each other, therefore the signal acceptance is lower due to the minimum ΔR requirement. Only the $e\mu$ final state is sensitive to the lowest mass point of 12 GeV, because it has a looser ΔR requirement than other final states. $\mu\tau_h$ is the most sensitive final state, setting observed (expected) limits as low as around 1.8%

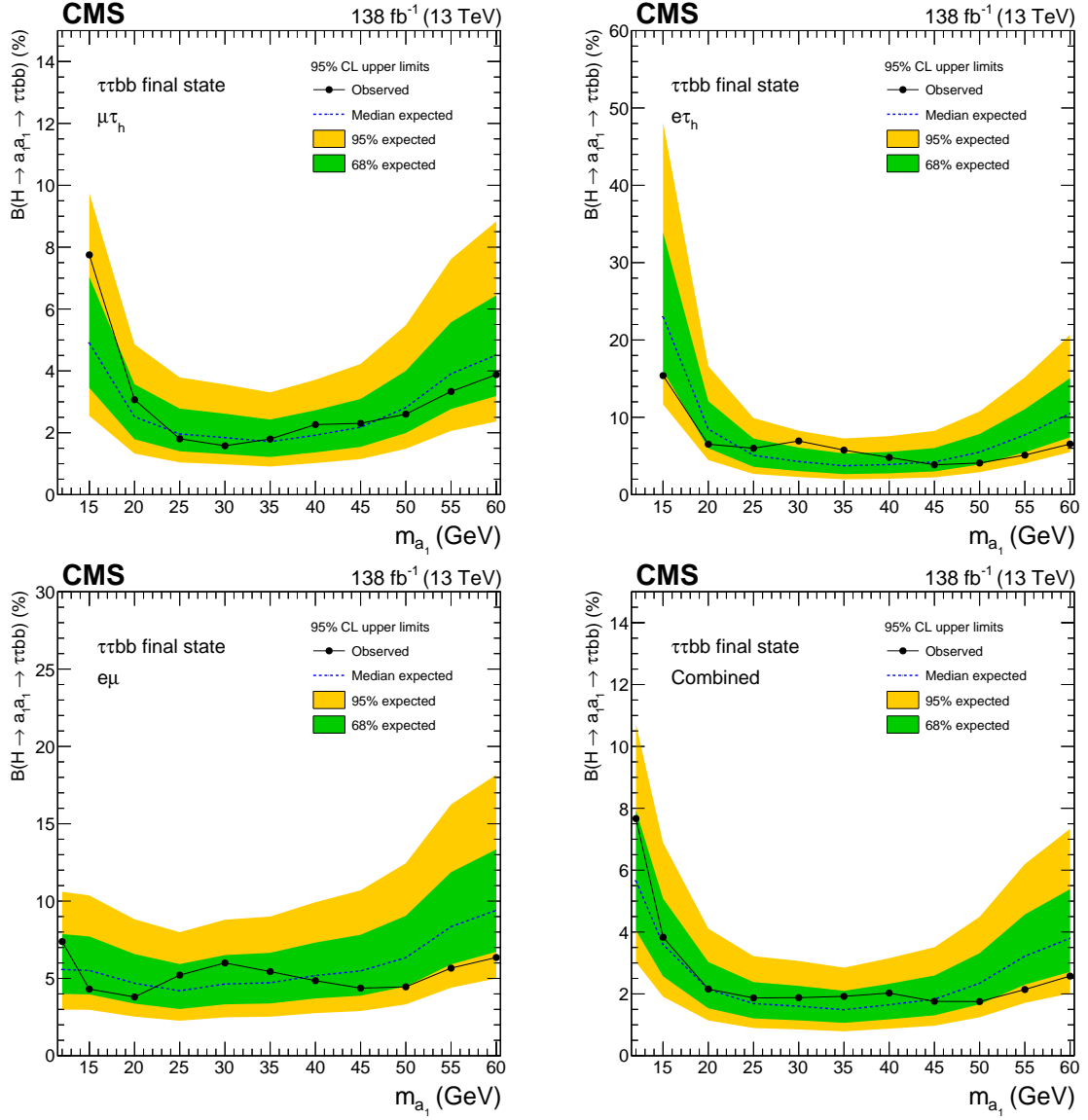


Figure 3.18: Observed and expected model-independent exclusion limits at 95% CL on the branching ratio $B(h \rightarrow aa \rightarrow \tau\tau bb)$ (%). The three final states are shown: $\mu\tau_h$ (upper left), $e\tau_h$ (upper right), and $e\mu$ (lower left), as well as the combination (lower right). The figures are taken from Ref. [15].

(1.7%) at 35 GeV. Although $B(\tau\tau \rightarrow \mu\tau_h) = B(\tau\tau \rightarrow e\tau_h)$, but $e\tau_h$ is less sensitive due to the higher- p_T requirements from the triggers. The combination of the three $\tau\tau$ final states sets observed (expected) limits between 1.7-7.7% (1.5-5.7%) for a pseudoscalar mass between 12 and 60 GeV.

The model-independent limits on $B(h \rightarrow aa \rightarrow \tau\tau bb)$ are converted to limits on $B(h \rightarrow$

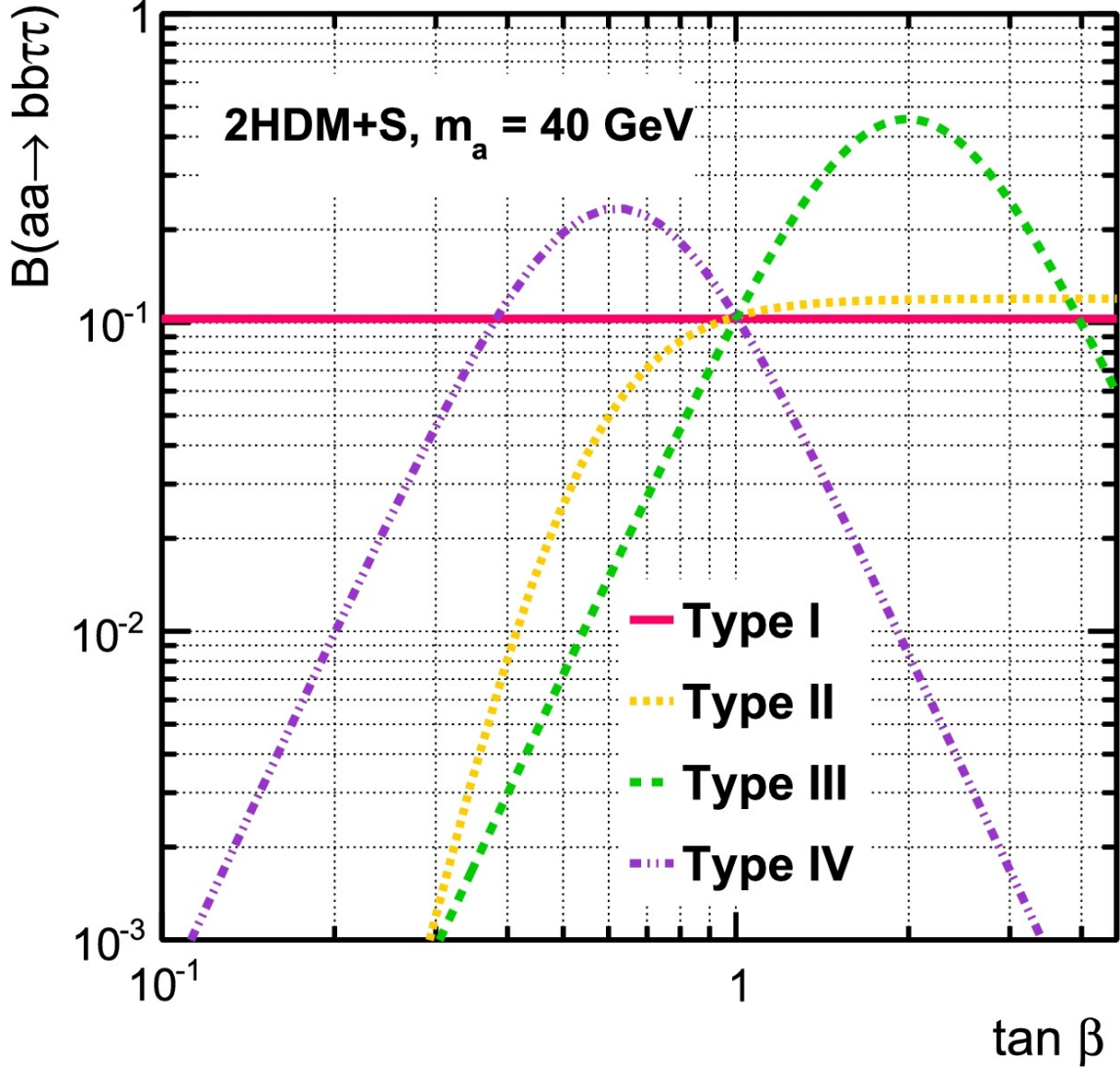


Figure 3.19: Branching ratio $B(aa \rightarrow bb\tau\tau)$ as a function of $\tan\beta$ predicted in 2HDM+S for a pseudoscalar mass of 40 GeV. The figure is taken from Ref. [16].

aa) interpreted in 2HDM+S scenarios. This requires theoretical predictions on $B(aa \rightarrow \tau\tau bb)$ [50, 111]. Fig. 3.19 shows $B(aa \rightarrow \tau\tau bb)$ as a function of $\tan\beta$ in the four types of 2HDM+S for a single mass point. As discussed in Sec. 1.2.3, the pseudoscalar decay in Type I is independent of $\tan\beta$. The branching ratio in Type II increases monotonically with $\tan\beta$, starting to converge around $\tan\beta = 2$. The branching ratios in Type III and IV are at maximum at around $\tan\beta = 0.6$ and $\tan\beta = 2$, respectively.

Fig. 3.20 shows the observed exclusion limits on $B(h \rightarrow aa)$ in the four types of

2HDM+S, with $\tan\beta$ around the values corresponding to the highest $B(aa \rightarrow \tau\tau bb)$. The observed exclusion limits on $B(h \rightarrow aa)$ plotted in the plane of $\tan\beta$ vs. signal mass for Type III and IV are shown in Fig. 3.21. The phase space enclosed by the contour line of $B(h \rightarrow aa) = 16\%$ is where the analysis is sensitive enough to exclude more than the constraint from the CMS measurements, which sets a limit of 16% on the branching ratio of the Higgs boson to undetected particles [2].

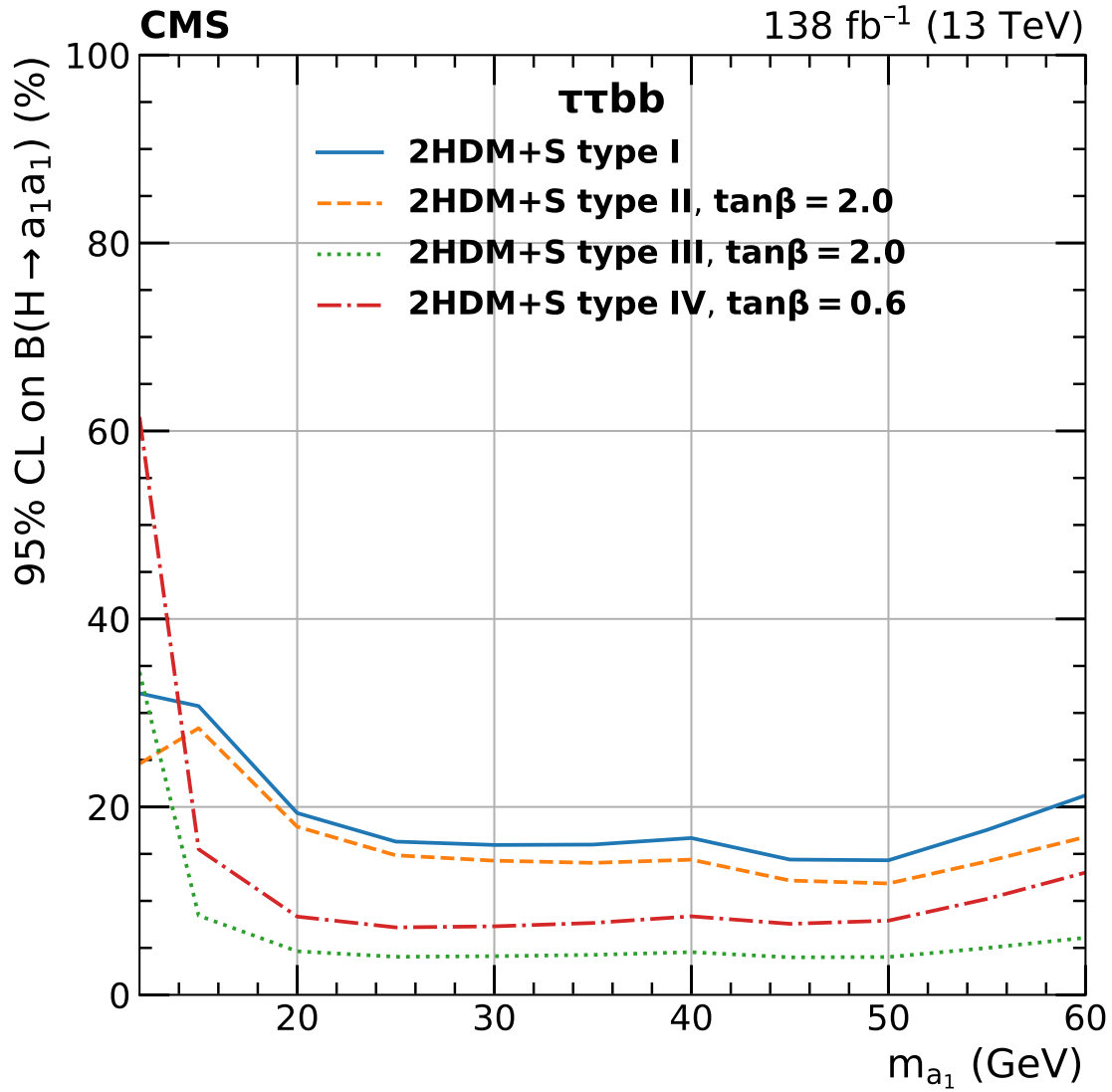


Figure 3.20: Observed exclusion limits at 95% CL on the branching ratio $B(h \rightarrow aa)$ (%). The results are obtained from the combination of all final states, interpreted in 2HDM+S Type I (independent of $\tan\beta$), Type II ($\tan\beta = 2$), Type III ($\tan\beta = 2$), and Type IV ($\tan\beta = 0.6$), respectively. The figure is taken from Ref. [15].

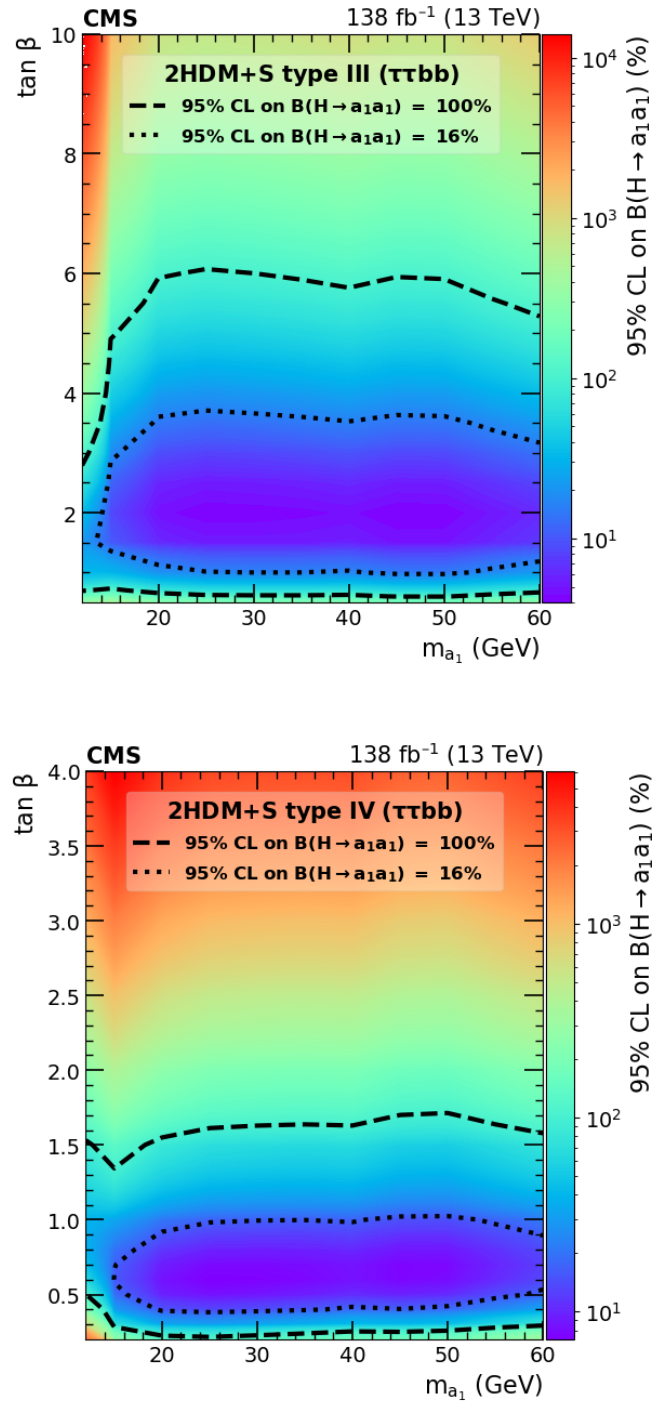


Figure 3.21: Observed exclusion limits at 95% CL on the branching ratio $B(h \rightarrow aa)$ (%), plotted in the plane of $\tan \beta$ vs. signal mass. The results are obtained from the combination of all final states, interpreted in 2HDM+S Type III (upper) and Type IV (lower), respectively. The figures are taken from Ref. [15].

3.2 Combination of the $\tau\tau bb$ and $\mu\mu bb$ final states

This section presents the combination of the two channels $h \rightarrow aa \rightarrow \tau\tau bb$ and $h \rightarrow aa \rightarrow \mu\mu bb$ [15]. Sec. 3.2.1 provides an overview of the $h \rightarrow aa \rightarrow \mu\mu bb$ analysis. Sec. 3.2.2 outlines the combination strategy. Sec. 3.2.3 presents the combination results. Finally, Sec. 3.3 discusses future aspects.

3.2.1 Analysis overview of $h \rightarrow aa \rightarrow \mu\mu bb$

The $\mu\mu bb$ channel has a lower branching ratio predicted in 2HDM+S compared to the $\tau\tau bb$ channel due to the mass difference between the muon and τ lepton. For example, in 2HDM+S Type III, $B(aa \rightarrow \tau\tau bb)$ can reach around 50% at $\tan\beta = 2$, while $B(aa \rightarrow \mu\mu bb)$ is only around 0.2% in the same scenario. However, this channel benefits from the high muon performance and the high di-muon mass resolution in CMS, making it competitive despite the low predicted branching ratio. Unlike the presence of τ leptons in the $\tau\tau bb$ channel, this analysis has only one final state, $\mu\mu bb$, and probes a similar mass range between 15 and 62.5 GeV.

Events are required to have two prompt muons with opposite electric charge and separated by $\Delta R > 0.4$. The leading and sub-leading muons must have a p_T of at least 17 and 15 GeV, respectively. The muons are required to pass the tight muon identification (see Sec. 2.3.2) and be isolated with $I_{\text{rel}}^\mu < 0.25$ (see Eq. 3.1).

Additionally, events are required to have at least two jets with $p_T > 15$ GeV, separated from muons by $\Delta R > 0.4$. As in the $\tau\tau bb$ channel, **DeepJet** is used to tag b jets [62, 63]. At least two b-tagged jets are required: one must pass the tight working point with an efficiency of 65% and a misidentification rate of 0.1%, and the other must pass the loose working point with an efficiency of 95% and a misidentification rate of 10% (see Sec. 2.3.3).

The di-muon mass, $m_{\mu\mu}$, is used as the observable for signal extraction due to its high resolution and its resonant distribution for the pseudoscalar mass. The two leading background processes are $DY\ Z \rightarrow \mu\mu$ and $t\bar{t} + \text{jets}$. The total background events are collectively estimated by fitting parametric polynomials to the data. The signal is described

by parametric models fitted to the simulated signal samples and interpolated between the limited number of generated mass points.

To suppress background, events are required to have $m_{\mu\mu}$ between 14 and 70 GeV and $p_T^{\text{miss}} < 60$ GeV. Optimization is done based on the variable $\chi_d^2 = \chi_{H,d}^2 + \chi_{bb,d}^2$, where $\chi_{H,d}$ and $\chi_{bb,d}$ are the decorrelated variables that measure the compatibility of the reconstructed masses with the signal hypothesis:

$$\chi_H = \frac{m_{\mu\mu bb}/\text{GeV} - 125}{\sigma_H}, \quad \chi_{bb} = \frac{m_{bb} - m_{\mu\mu}}{\sigma_{bb}}. \quad (3.14)$$

σ_H and σ_{bb} are the mass resolutions obtained by Gaussian fits to $m_{\mu\mu bb}$ and $m_{\mu\mu}$ in simulated events, respectively. Events are required to have $\chi_d^2 < 1.5$.

The selected events are then divided into five categories. The first category, called ‘Low p_T ’, selects events with at least one b-tagged jet with low $p_T < 20$ GeV, improving sensitivity for low signal masses. Another category, called ‘VBF’, targets the VBF production channel of the Higgs boson by selecting events with two jets that have a large invariant mass $m_{jj} > 250$ GeV. The other three categories are based on whether the b jet, which passes the loose working point at baseline, passes at most the loose (TL), medium (TM), or tight (TT) working points, respectively.

Signal is estimated from simulation with associated systematic uncertainties. Background is estimated by fitting parametric models to data, with uncertainties in both the parameters and the functional form. An unbinned maximum likelihood fit is performed on the $m_{\mu\mu}$ distributions. The fitted background models are consistent with data in all five categories, as shown in Fig. 3.22, with no excess of data events observed. Therefore, exclusion limits at 95% CL on the signal branching ratio $B(h \rightarrow aa \rightarrow \mu\mu bb)$ are obtained, as shown in Fig. 3.23.

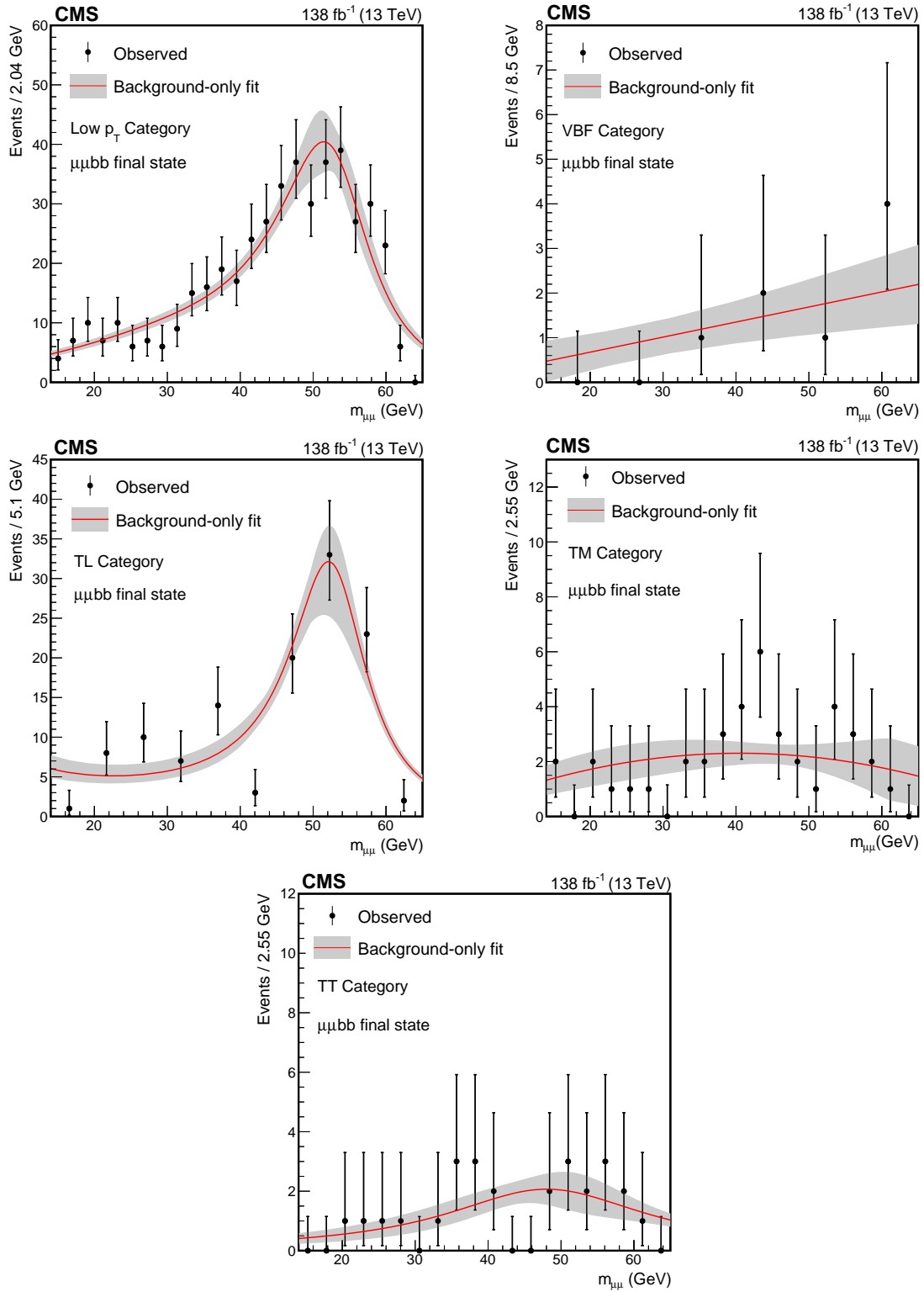


Figure 3.22: Fitted background models in $m_{\mu\mu}$ in the five categories. The figures are taken from Ref. [15].

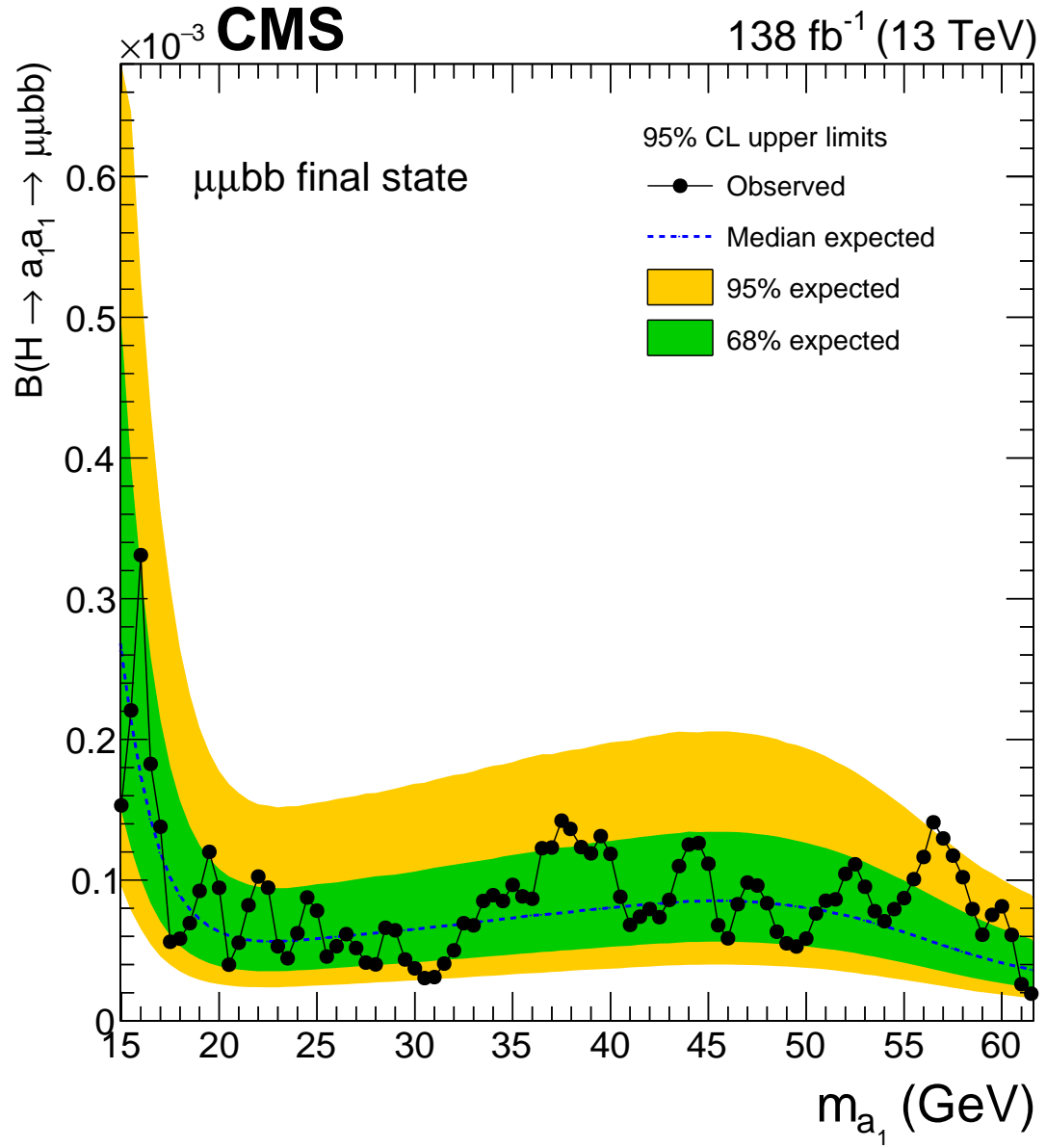


Figure 3.23: Observed and expected model-independent exclusion limits at 95% CL on the branching ratio $B(h \rightarrow aa \rightarrow \mu\mu b\bar{b})$. The figure is taken from Ref. [15].

3.2.2 Combination strategy

In the combination, systematic uncertainties common between the two analyses are correlated. These include theoretical uncertainties related to signal modeling and experimental uncertainties related to jet energy, b-tagging, and muon performance. All other uncertainties are included but treated as uncorrelated between the two channels.

To combine the two channels, the branching ratios of the pseudoscalars to the fermions are required. To obtain the combined limits on $B(h \rightarrow aa \rightarrow llbb) = B(h \rightarrow aa \rightarrow \mu\mu bb) + B(h \rightarrow aa \rightarrow \tau\tau bb)$, the yields of the signals $h \rightarrow aa \rightarrow \mu\mu bb$ and $h \rightarrow aa \rightarrow \tau\tau bb$ are scaled by $B(aa \rightarrow \mu\mu bb)/B(aa \rightarrow llbb)$ and $B(aa \rightarrow \tau\tau bb)/B(aa \rightarrow llbb)$, respectively, in the combined fit. The relative branching ratios can be computed from the exact decay width predicted in 2HDM+S [50, 111]:

$$\Gamma(a \rightarrow ll) = \frac{(\xi_l^M)^2 m_l^2 m_a}{8\pi\nu^2} \sqrt{1 - \frac{4m_l^2}{m_a^2}}, \quad (3.15)$$

where ξ_l^M depends on the 2HDM+S types and the coupling parameters, $\nu \approx 246$ GeV, m_l is the lepton mass, and m_a is the pseudoscalar mass. Within the same 2HDM+S scenario, the ratios become:

$$\frac{B(aa \rightarrow \mu\mu bb)}{B(aa \rightarrow llbb)} \approx \left(1 + \frac{m_\tau^2}{m_\mu^2}\right)^{-1}, \quad \frac{B(aa \rightarrow \tau\tau bb)}{B(aa \rightarrow llbb)} \approx \left(1 + \frac{m_\mu^2}{m_\tau^2}\right)^{-1}, \quad (3.16)$$

for $m_a \gg m_l$. Here, the dependence on the 2HDM+S type and parameters are canceled in the ratios. Consequently, the signal combined in this way and the combined limits on $B(h \rightarrow aa \rightarrow llbb)$ are independent of the 2HDM+S type and $\tan\beta$. However, these limits are not model-independent as they are still interpreted within the 2HDM+S framework, where the relative signal strengths are predicted.

The combined limits on $B(h \rightarrow aa)$ can then be directly derived by dividing the combined limits on $B(h \rightarrow aa \rightarrow llbb)$ by the predicted $B(aa \rightarrow llbb)$, which depends on the 2HDM+S type and $\tan\beta$.

3.2.3 Exclusion limits and model interpretation

The combined analysis shows statistical uncertainties dominating over systematic uncertainties. Systematic uncertainties contribute around 6% of the total uncertainty, primarily associated with the theoretical modeling of the signal, jet energy scale in the $\mu\mu bb$ channel, and fake background estimation in the $\tau\tau bb$ channel.

Fig. 3.24 shows the exclusion limits at 95% CL on $B(h \rightarrow aa \rightarrow llbb)$ combining the $\mu\mu bb$ and $\tau\tau bb$ channels. These limits range from 0.6-7.7% (observed) and 0.8-5.7% (expected). Although interpreted within the 2HDM+S framework, these limits are independent of the model type and parameters, as discussed in Sec. 3.2.2.

Fig. 3.25 shows the observed combined limits at 95% CL on $B(h \rightarrow aa)$ in the four types of 2HDM+S. In Type II, which is a particular case of producing the scalar sector of the NMSSM, the limits are observed to be in the range of 5-23% for $\tan\beta > 1$. The observed combined limits on $B(h \rightarrow aa)$ plotted in the plane of $\tan\beta$ vs. signal mass for Type III and IV are shown in Fig. 3.26. Additionally, the combination result is added to the CMS plots that summarize the results of all $h \rightarrow aa$ searches in different channels. Fig. 3.27 shows the summary for the 2HDM+S Type II ($\tan\beta = 2$) scenario. In the pseudoscalar mass range between 12 and 60 GeV, the limits from this combination are the most stringent among the analyzed channels.

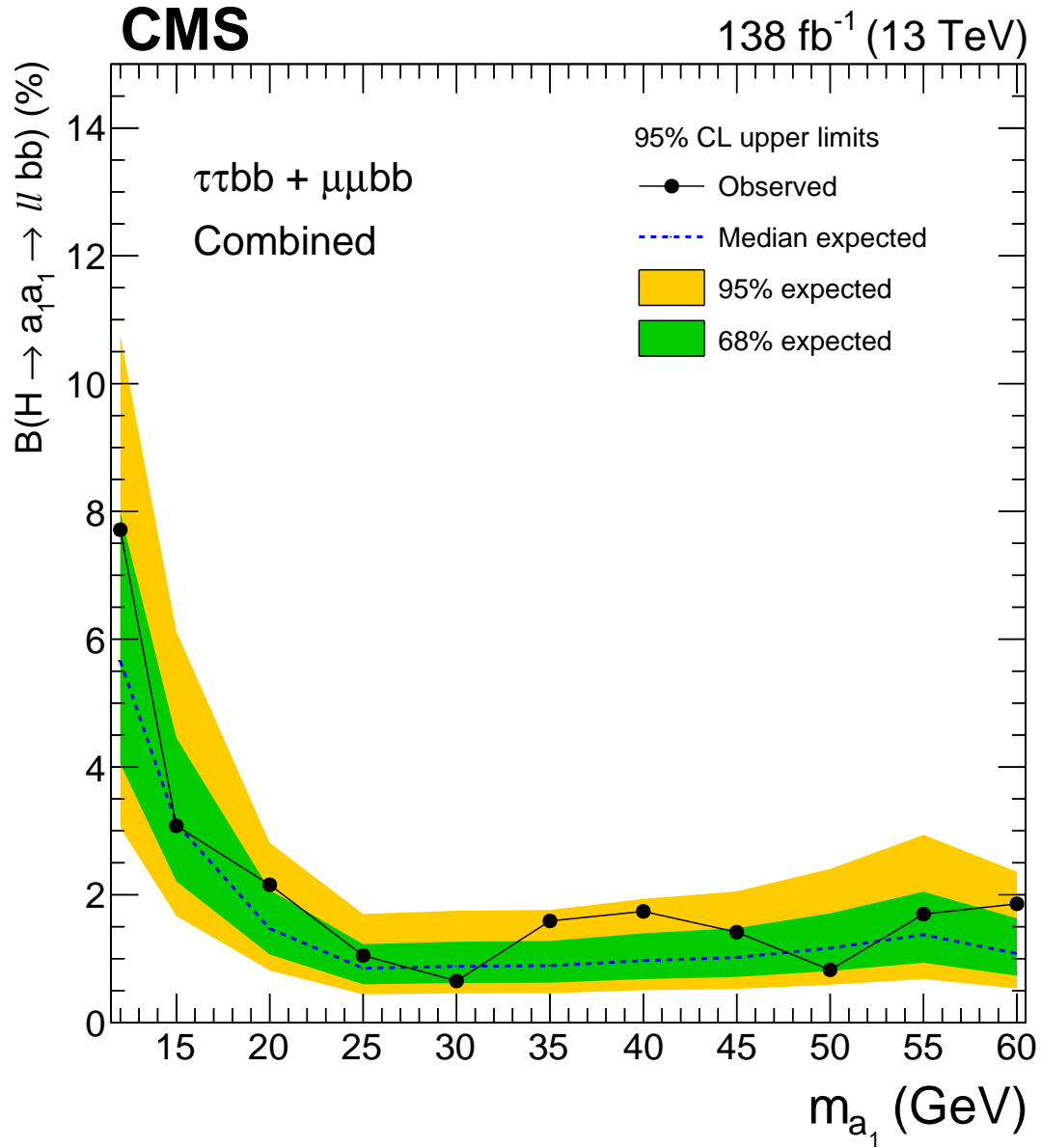


Figure 3.24: Observed and expected exclusion limits at 95% CL on the branching ratio $B(h \rightarrow aa \rightarrow llbb)$ (%) within 2HDM+S, where $l = \mu, \tau$. The figure is taken from Ref. [15].

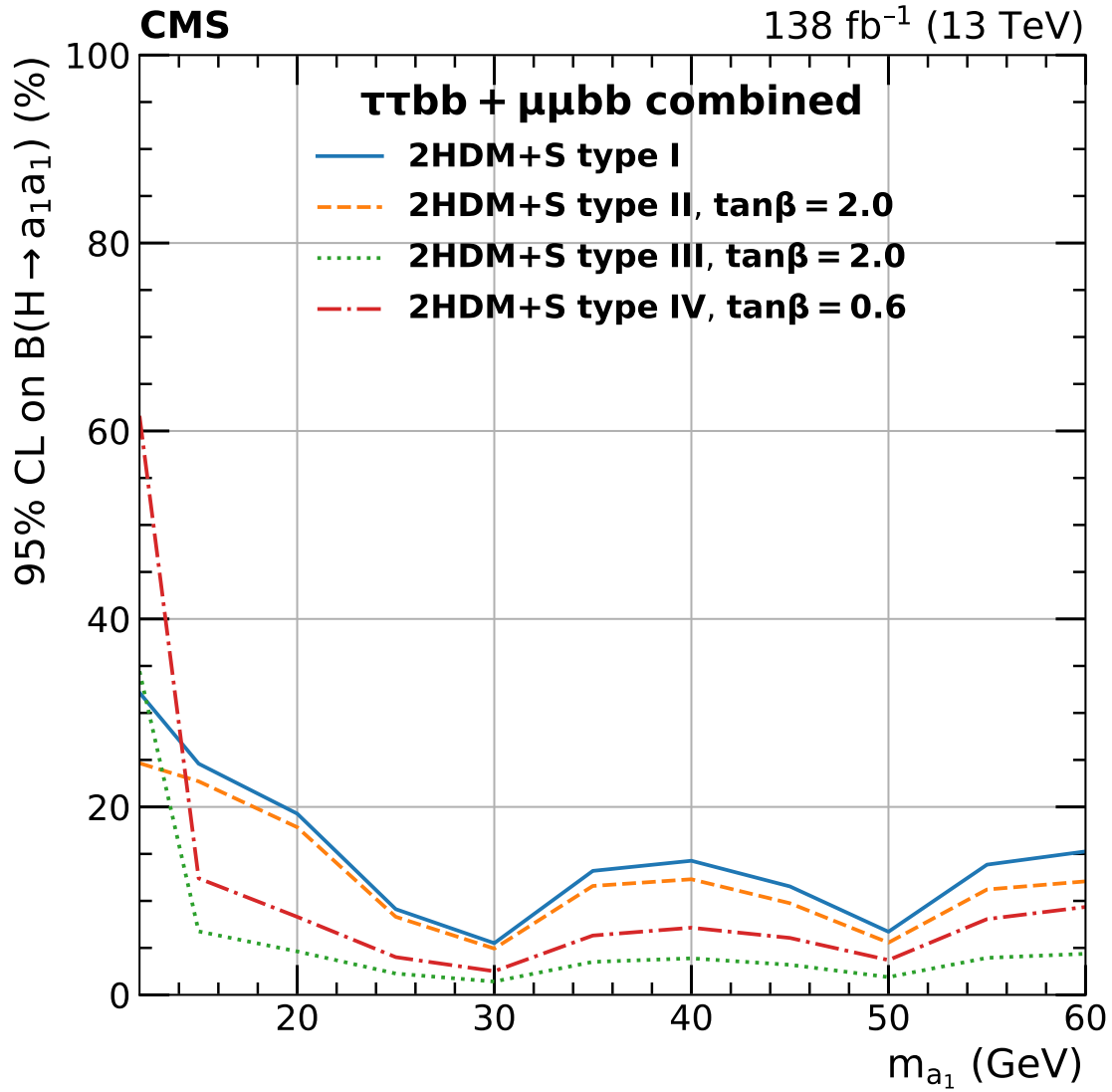


Figure 3.25: Observed exclusion limits at 95% CL on the branching ratio $B(h \rightarrow aa)$ (%). The results are obtained from the combination of the $\tau\tau b\bar{b}$ and $\mu\mu b\bar{b}$ final states, interpreted in 2HDM+S Type I (independent of $\tan\beta$), Type II ($\tan\beta = 2$), Type III ($\tan\beta = 2$), and Type IV ($\tan\beta = 0.6$), respectively. The figure is taken from Ref. [15].

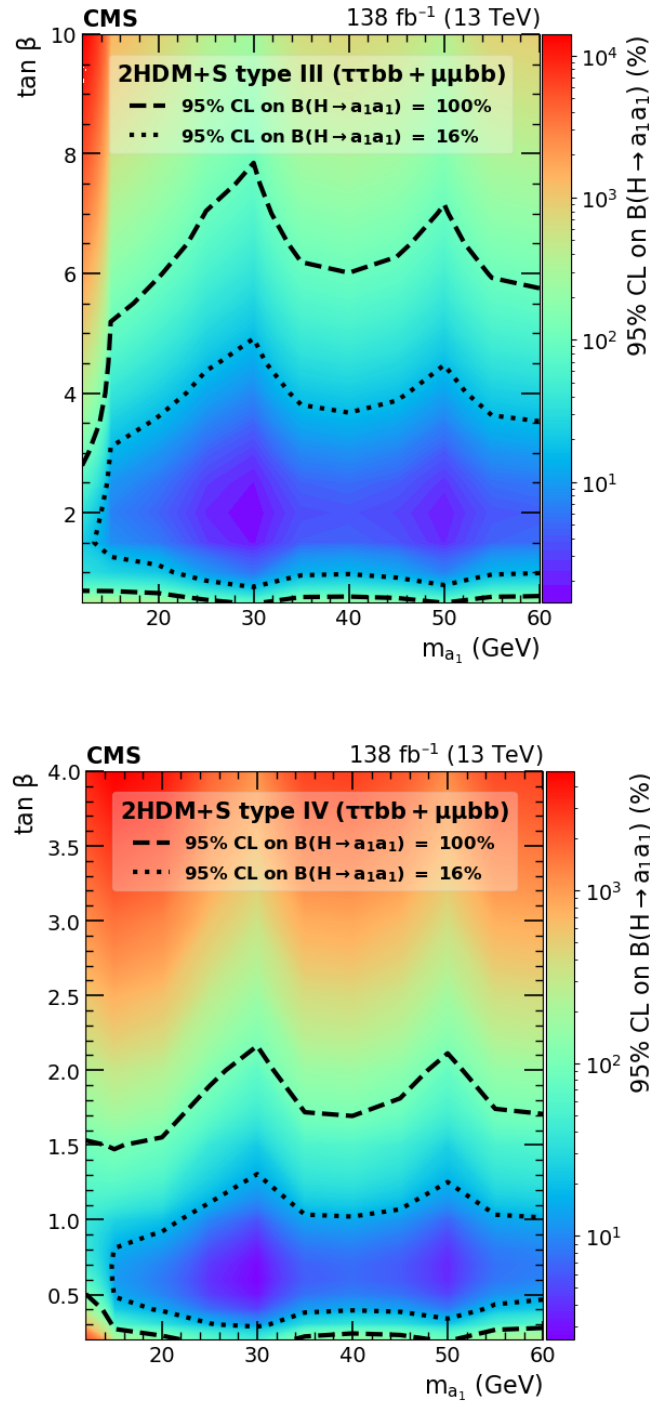


Figure 3.26: Observed exclusion limits at 95% CL on the branching ratio $B(h \rightarrow aa)$ (%), plotted in the plane of $\tan \beta$ vs. signal mass. The results are obtained from the combination of the $\tau\tau bb$ and $\mu\mu bb$ final states, interpreted in 2HDM+S Type III (upper) and Type IV (lower), respectively. The figures are taken from Ref. [15].

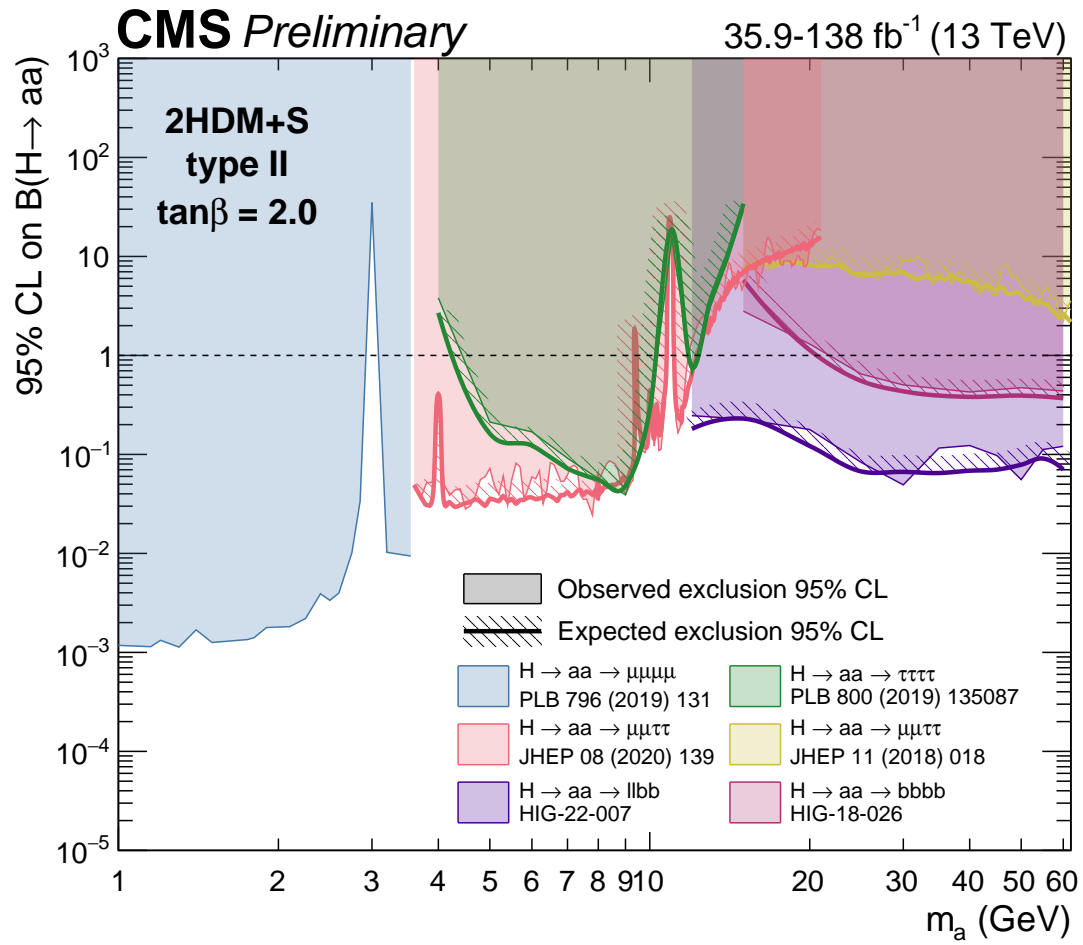


Figure 3.27: Summary of observed and expected exclusion limits at 95% CL on the branching ratio $B(h \rightarrow aa)$ in 2HDM+S Type II at $\tan\beta = 2$. The figure is taken from Ref. [17].

3.3 Future prospects

The searches for $h \rightarrow aa$ will continue as more data will be collected in the ongoing Run 3 and the upcoming Phase-2 HL-LHC. To examine the expected sensitivity gain in the future, a projection study based on the 2016-only analysis results to the HL-LHC is performed in the $h \rightarrow aa \rightarrow bb\tau\tau$ channel [18].

In Phase-2 HL-LHC, the center-of-mass energy is expected to reach 14 TeV with a total integrated luminosity of 3000 fb^{-1} . In this study, the Phase-2 single- μ trigger requiring an isolated muon with $p_T > 24 \text{ GeV}$ is simulated, and its efficiency per event for the $\mu\tau_h$ final state is shown in Fig. 3.28 (top). The expected exclusion limits are obtained by projecting the results to 3000 fb^{-1} . As shown in Fig. 3.28 (bottom), the expected exclusion limits on the branching ratio $B(h \rightarrow aa \rightarrow bb\tau\tau)$ reach as low as around 0.4-0.5% in the intermediate mass range, compared to the lowest 1.5% at 35 GeV in the full Run 2 analysis shown in Fig. 3.18.

Despite the anticipated sensitivity improvement due to increased data statistics, a significant challenge in all searches for $h \rightarrow aa$ remains unaddressed. The main issue is that the final-state objects are typically too soft because they originate from light pseudoscalar decays, causing most of the signal events to be rejected by triggers that set relatively high p_T thresholds on these final-state objects.

For example, both the 2016-only and the full Run 2 analyses of the search for $h \rightarrow aa \rightarrow bb\tau\tau$ did not consider the $\tau_h\tau_h$ final state due to the low signal acceptance caused by the high $\tau_h p_T$ threshold at trigger, despite it having the highest branching ratio among all $\tau\tau$ final states. Moreover, in the analyzable $\mu\tau_h$ final state, Fig. 3.29 shows that most of the final-state $\mu\tau_h$ have low momenta, and existing single- μ and cross- $\mu\tau_h$ triggers already remove over 90% of the signal yields.

The issue remains unaddressed if the same trigger strategy continues in the future, regardless of increased data statistics. Even more stringent p_T thresholds would likely be implemented to manage trigger rates during high-luminosity collision runs. This challenge will persist in all LHC analyses targeting signatures with low- p_T final-state objects. In

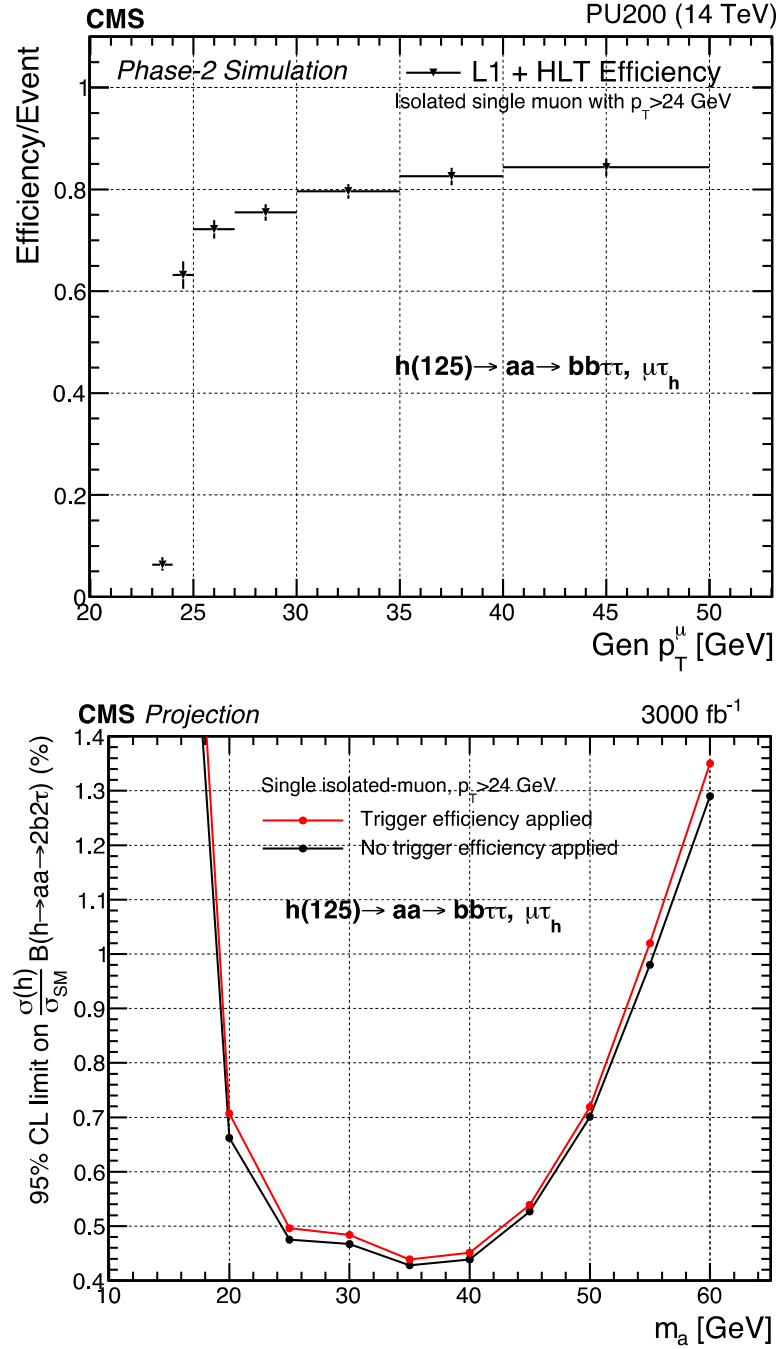


Figure 3.28: Phase-2 trigger study for the search channel $h \rightarrow aa \rightarrow bb\tau\tau$. Top: trigger efficiency of the isolated single- μ trigger ($p_T > 24$ GeV) as a function of the generated muon p_T . Bottom: expected exclusion limits at 95% CL on the branching ratio $B(h \rightarrow aa \rightarrow bb\tau\tau)$ (%), with and without the trigger efficiency corrections. The figures are taken from Ref. [18].

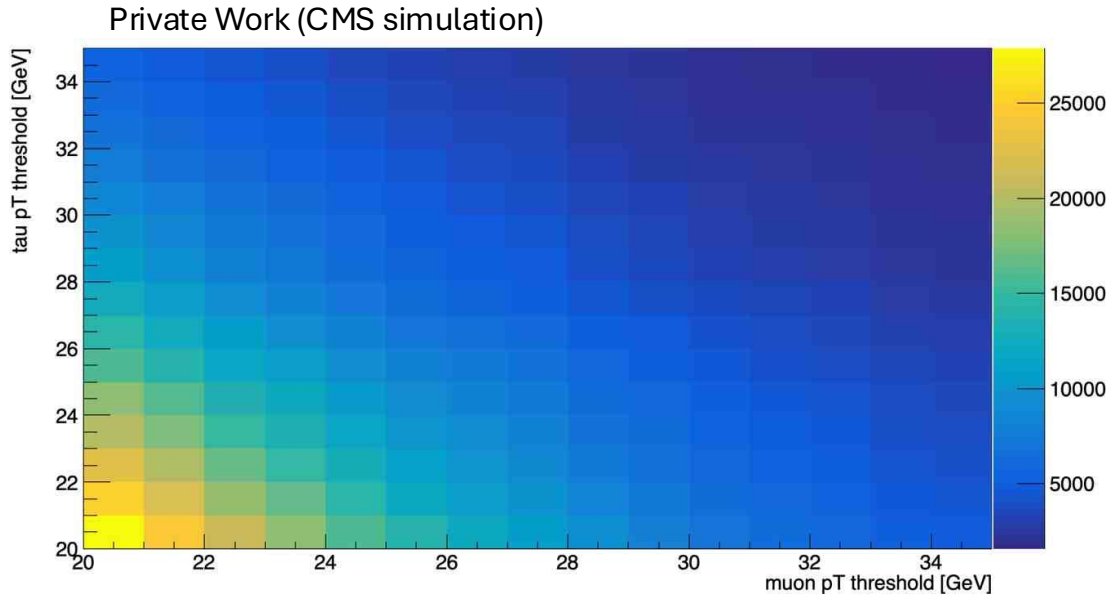


Figure 3.29: Event yield of the $h \rightarrow aa \rightarrow \tau\tau bb$ signal as a function of the muon and τ_h p_T in the $\mu\tau_h$ final state, prior to applying trigger p_T requirements.

Part II, potential solutions are proposed to address this challenge at a lower level using novel machine learning methods at the trigger and in the low-latency domain.

Chapter 4

Summary of Part I

A search for new physics beyond the Standard Model with the CMS experiment is presented, targeting an exotic decay of the Higgs boson to a pair of light pseudoscalars, which subsequently decay to a pair of b quarks and a pair of τ leptons: $h \rightarrow aa \rightarrow \tau\tau bb$. The analysis is performed using the full Run 2 dataset of proton-proton collisions at $\sqrt{s} = 13$ TeV, collected by the CMS detector, corresponding to an integrated luminosity of 138 fb^{-1} . No excess of events over the Standard Model predictions is observed. Exclusion limits at 95% confidence level are set on the branching ratio $B(h \rightarrow aa \rightarrow \tau\tau bb)$, ranging from 1.7-7.7% (observed) and 1.5-5.7% (expected), for a pseudoscalar mass between 12 and 60 GeV.

The analysis is combined with the $h \rightarrow aa \rightarrow \mu\mu bb$ analysis, which sets exclusion limits at 95% confidence level on the branching ratio $B(h \rightarrow aa \rightarrow \mu\mu bb)$, ranging from $(0.17-3.3) \times 10^{-4}$ (observed) and $(0.35-2.6) \times 10^{-4}$ (expected), for a pseudoscalar mass between 15 to 62.5 GeV.

The combined analysis sets exclusion limits at 95% confidence level on the branching ratio $B(h \rightarrow aa \rightarrow llbb)$, where $l = \mu, \tau$, ranging from 0.6-7.7% (observed) and 0.8-5.7% (expected), interpreted within the 2HDM+S framework.

Interpreting in 2HDM+S, observed exclusion limits at 95% confidence level on the branching ratio $B(h \rightarrow aa)$ are set at around 23% for most of the Type II scenarios, and

as low as 1 and 3% for Types III ($\tan \beta = 2$) and IV ($\tan \beta = 0.5$), respectively.

The main challenge for these analyses targeting signatures with low-momentum final-state objects is the harsh trigger selection that requires high-momentum thresholds on the objects. The issue persists if the trigger strategy remains unchanged, regardless of increased data statistics and improved offline analysis techniques. This problem brings up the next part of the thesis, which introduces novel machine learning methods to address the issue at a lower level.

Part II

Fast machine learning solutions for the LHC

Chapter 5

Motivations and fundamentals

Machine learning (ML) has proven its huge success in most, if not all, scientific and technological domains [112]. The particle physics research program at the Large Hadron Collider (LHC) is no exception. The demand for ML applications is expected to grow rapidly due to their potential of surpass the performance limitations of traditional methods and algorithms across various fields.

In particular, protons are collided every 25 nanoseconds during LHC runs, corresponding to an extreme rate of 40 MHz, producing hundreds of terabytes of raw data per second. Due to the limited bandwidth and storage space, and because potential physics processes of interest occur at low probabilities, the raw data must be processed and filtered at the edge of the detectors. This processing is handled by the trigger system [11, 113–115], which performs the first round of data selection to decide if an event should be stored for later analysis or discarded permanently. Additionally, because this processing is run in real time concurrently with the 40 MHz collisions, the trigger system is constrained to make decisions within 4 microseconds (4×10^{-6} s). Therefore, algorithms capable of selecting potential physics events out of the vast and noisy background while requiring minimal computational resources and operating extremely quickly are essential for this critical task.

This motivates fast ML in the context of the LHC—running physics-sensitive ML algo-

rithms efficiently on custom hardware such as Field-Programmable Gate Arrays (FPGAs) with a low latency typically on the order of microseconds and below. This combination leverages the capability of ML algorithms to generalize accurate predictions by automatically learning from data, often outperforming traditional algorithms limited by explicit instructions. FPGAs can be customized for a given algorithm to optimize resource allocation and running speed, which is ideal for resource-constrained environments such as the LHC experiments. They can potentially run models of reasonable size within sub-microseconds.

Before delving into the anomaly detection trigger algorithm in Ch. 6 and the symbolic regression for efficient model compression in Ch. 7, this chapter briefly introduces the necessary fundamentals. The fundamentals of ML are presented in Sec. 5.1. The fundamentals of FPGAs and the implementation of ML on FPGAs are presented in Sec. 5.2.

5.1 Machine learning

In ML, problems are defined and solved by leveraging data. Without explicitly programmed logic, ML models automatically learn and recognize patterns from data and make predictions [93, 112].

Input features are variables that quantify the properties of data and are used for ML models to learn patterns. The choice of features depends on the problem at hand, as some features can be more relevant in solving one problem but add noise in another. Features can be of different types.

For example, the features used in the analysis presented in Sec. 3.1.7 are numerical with continuous values, such as the momenta and invariant masses of the final-state objects. However, discrete values, such as the number of jets or the discriminant working points, could also be potential features. For image problems, images are the raw data, where each image can be converted into a 2D array of pixel intensity values or a 3D array with an additional dimension representing the RGB colors. For language problems, texts are the raw data, where each word is extracted and embedded into a vector space representation

that can be processed by ML models.

Once a set of features is chosen, they are used to train the ML model for a specific goal, such as classifying the categories of the data or making numerical predictions. The process of creating and optimizing the set of input features for a given problem is called feature engineering.

Two types of learning are focused in the following chapters: supervised and unsupervised. In supervised learning, training data are labeled to guide the model to learn the definite answers, which requires pre-specified input-output pairs. For example, the training data in the analysis presented in Sec. 3.1.7 have each event labeled as either signal or background, which is a classification problem. For a regression problem, such as predicting the energy of a given jet from a set of features representing the corresponding detector response, the true energy value for each input jet needs to be specified in the training data.

In contrast, unsupervised learning does not require labeled data and allows models to discover patterns inherently present in the data. Anomaly detection [116] is an example, which is employed to build the trigger algorithm presented in Ch. 6. This approach identifies rare or unexpected deviations from the training data without knowing the exact forms of the deviations.

Anomaly detection is particularly valuable for new physics searches at the LHC since the space of possible theoretical physics models is vast and it is inefficient to target one specific model at a time in each analysis. This approach is well-suited for situations where theoretical models are not known or well-defined, allowing for the detection of any unusual signatures deviating from the Standard Model (SM) predictions that could potentially indicate new physics, without relying on specific physics models.

ML models have adjustable parameters that parameterize their predictions, optimized by an objective function called a loss function. The loss function typically measures the distance between the predicted outputs and the expected outputs, depending on the problem. Minimizing this loss function trains the model to perform the desired task. Regularization

terms can be added to the loss function to penalize certain undesired behaviors, such as excessively large model weights.

Since ML models can involve numerous tunable weights, finding the optimal set of weights is a complex multivariate optimization problem typically approached by numerical methods. Gradient descent is a class of iterative methods to solve this optimization problem numerically. In its simplest form, gradient descent initializes a set of model weights $\boldsymbol{\theta}$ to $\boldsymbol{\theta}_0$ and updates iteratively in the direction of the negative gradient of the loss function J :

$$\boldsymbol{\theta}_{i+1} = \boldsymbol{\theta}_i - \alpha \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}_i), \quad (5.1)$$

where the values at step $(i + 1)$ are updated based on the previous step i , and α is the learning rate that controls the size of the steps. The gradient terms can be efficiently computed with backpropagation [117], which uses the chain rule to propagate errors backward through the neural network (NN), avoiding the direct computation of gradients that can be computationally infeasible, especially for large models.

In principle, the loss function J is computed by averaging individual losses across the entire training dataset. However, this can be computationally intensive for large datasets or models. Instead, the loss function can be estimated from a small subset of the dataset, called a mini-batch, and the model weights are updated once for each mini-batch instead of once for the entire dataset. This is referred to as mini-batch stochastic gradient descent. There are more sophisticated gradient descent variants, such as Adam [95], which adaptively adjust the learning rate for each weight, ensuring stable and efficient convergence.

Sec. 5.1.1 and Sec. 5.1.2 describe the fully-connected NN and convolutional NN (CNN), respectively, which are building blocks of the anomaly detection models presented in Ch. 6.

5.1.1 Fully-connected neural network

An NN in ML is a class of artificial models that process data and make predictions similarly to how biological neurons function in the human brain [89–93]. Constructing an NN involves three essential components: the input layer, hidden layers, and output layer.

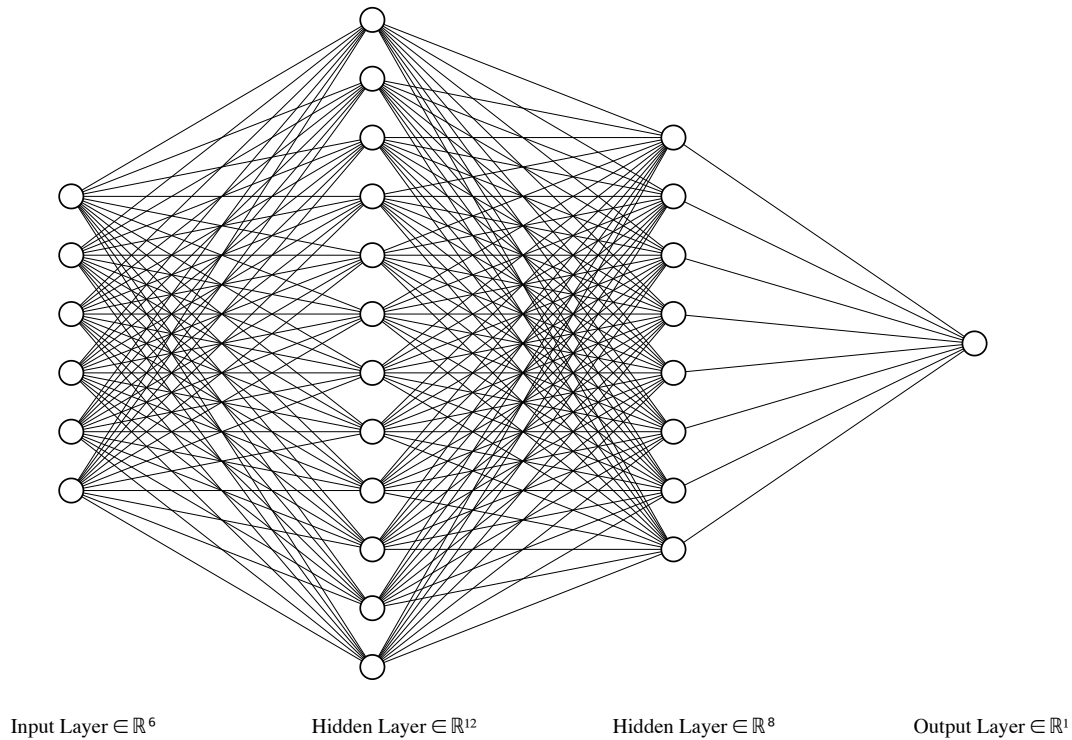


Figure 5.1: An example fully-connected NN with two hidden layers is shown, with the first hidden layer consisting of 12 neurons and the second hidden layer consisting of 8 neurons. The NN has 6 input features and produces 1 output. The figure is generated using the NN-SVG tool [19].

The input layer receives the input features, while the output layer produces the model outputs. The hidden layers, situated between the input and output layers, consist of nodes (neurons) that receive, weigh, and combine inputs from neurons in the previous layer, apply activation functions, and then send the processed data to the next layer. An NN is fully-connected if each neuron in one layer is connected to all neurons in adjacent layers. Fig. 5.1 shows an example of a fully-connected NN with two hidden layers.

Starting from the input layer, denoted as $\mathbf{x}^{(0)}$, which receives a vector representing the n_0 input features: $\mathbf{x}^{(0)} = \mathbf{x} \in \mathbb{R}^{n_0}$. The input features are then passed through a stack of hidden layers, where a linear transformation followed by a nonlinear activation is performed in each passage from one layer to the next. The hidden layer $l > 0$, which has a dimension of n_l , receives the input vector $\mathbf{x}^{(l-1)} \in \mathbb{R}^{n_{l-1}}$ from the previous layer $l - 1$

and outputs the transformed vector $\mathbf{x}^{(l)} \in \mathbb{R}^{n_l}$, which takes the generic form:

$$\mathbf{x}^{(l)} = g^{(l)}(\mathbf{w}^{(l)}\mathbf{x}^{(l-1)} + \mathbf{b}^{(l)}). \quad (5.2)$$

Here, $\mathbf{w}^{(l)}\mathbf{x}^{(l-1)} + \mathbf{b}^{(l)}$ is a linear transformation which weighs each of the neuron values from the previous layer by the weight matrix $\mathbf{w}^{(l)} \in \mathbb{R}^{n_l \times n_{l-1}}$ and adds the bias terms $\mathbf{b}^{(l)} \in \mathbb{R}^{n_l}$. Then, each of the neurons is activated by a nonlinear function $g^{(l)}(\cdot)$, which is essential as nonlinearity ensures the NN can learn complicated patterns from data.

Conversely, if all layers perform only linear transformations, these serial linear transformations are equivalent to a single linear transformation regardless of the number of layers stacked, limiting the NN to linear data. In fact, the universal approximation theorem [91] guarantees that any continuous function can be approximated to an arbitrary level of accuracy by a fully-connected NN with at least one hidden layer and a sufficient number of neurons.

Combining all operations, a fully-connected NN with L layers, excluding the input layer, is essentially a function $\hat{y} = \phi(\mathbf{x}; \mathbf{w}, \mathbf{b})$, where $\phi : \mathbb{R}^{n_0} \rightarrow \mathbb{R}^{n_L}$ integrates all linear transformations and nonlinear activations from all layers, with \mathbf{w} and \mathbf{b} representing all the weights and biases.

The weights \mathbf{w} and biases \mathbf{b} are trainable parameters determined by minimizing the objective function with respect to a training dataset $\{(\mathbf{x}_i, y_i)\}_{i=1, \dots, N}$. Finding the optimal \mathbf{w} and \mathbf{b} involves solving the following optimization problem:

$$\mathbf{w}^*, \mathbf{b}^* = \arg \min_{\mathbf{w}, \mathbf{b}} \sum_{i=1}^N l(\phi(\mathbf{x}_i; \mathbf{w}, \mathbf{b}), y_i), \quad (5.3)$$

where $l(\cdot)$ is the individual loss function. This optimization problem can be solved numerically using gradient-based methods.

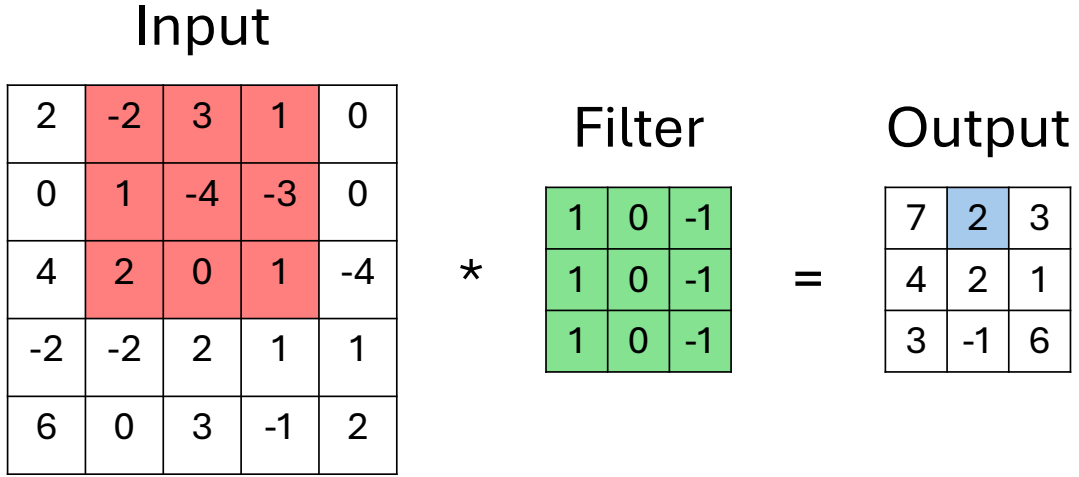


Figure 5.2: An example of a convolution operation of a 3×3 filter applied to a 5×5 input, with valid padding and a stride size of 1. The output value 2 in blue corresponds to the dot product of the input subgrid in red and the filter in green: $(-2) \times (1) + (1) \times (1) + (2) \times (1) + (3) \times (0) + (-4) \times (0) + (0) \times (0) + (1) \times (-1) + (-3) \times (-1) + (1) \times (-1) = 2$. The output dimensions are calculated by: $\lfloor \frac{H-M+2P}{S} + 1 \rfloor \times \lfloor \frac{W-N+2P}{S} + 1 \rfloor = \lfloor \frac{5-3+2(0)}{1} + 1 \rfloor \times \lfloor \frac{5-3+2(0)}{1} + 1 \rfloor = 3 \times 3$.

5.1.2 Convolutional neural network

CNN [118, 119] is a type of NN where neurons are connected in specific orders, specifically designed for processing data with a fixed-grid topology, such as images.

The key feature of a CNN is the convolution operation, which involves small matrices of weights, called filters, sliding over the input grids and performing inner products to detect patterns. For an input matrix I of dimensions $H \times W$, a 2D convolution operation with a filter F of dimensions $M \times N$ ($M \leq H$ and $N \leq W$) is given by:

$$(I * F)[i, j] = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} I[i+m, j+n] F[m, n], \quad (5.4)$$

where valid padding and a stride size of 1 are assumed. Padding is a technique to preserve information at the grid boundary and control the output dimension by adding extra grid points surrounding the original input grid. There are two common padding options: valid padding and same padding. Valid padding refers to no extra grid points being added.

Same padding refers to adding extra grid points such that the output dimension remains the same as the input. Stride size is a parameter that controls the number of grid points to move at a time when sliding the filters across the input grid. The output dimension for a convolution operation is given by $\lfloor \frac{H-M+2P}{S} + 1 \rfloor \times \lfloor \frac{W-N+2P}{S} + 1 \rfloor$, where P and S denote the padding and the stride size, respectively. The filter weights are trainable parameters. An example of a 2D convolution operation is illustrated in Fig. 5.2.

It is straightforward to generalize the 2D convolution operation to volume inputs, such as an image with RGB colors, which can be represented by a tensor of dimensions $H \times W \times 3$, as opposed to a grayscale image represented by a tensor of dimensions $H \times W \times 1$. For an input tensor I of dimensions $H \times W \times L$, where L denotes the number of channels, a 2D convolution operation with K filters of dimensions $M \times N \times L$, denoted collectively by the F tensor of dimensions $M \times N \times L \times K$, is given by:

$$(I * F)[i, j, k] = \sum_{l=0}^{L-1} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} I[i+m, j+n, l] F[m, n, l, k], \quad (5.5)$$

where valid padding and a stride size of 1 are assumed. The number of filters K in the convolution operation becomes the number of channels in the output tensor.

Additionally, pooling layers can be used to perform dimensionality reduction. Similar to convolutional layers, pooling is done by sliding filters to aggregate information but without trainable weights. There are two primary kinds of pooling: max pooling and average pooling. Max pooling sets the output value as the maximum value within the subgrid, while average pooling uses the average value calculated within the subgrid.

As the outputs of convolutional and pooling layers are still in tensor form with multiple dimensions, they can be flattened into a vector followed by a fully-connected NN to produce the output layer for the desired goal, such as classification or regression. An example CNN is illustrated in Fig. 5.3.

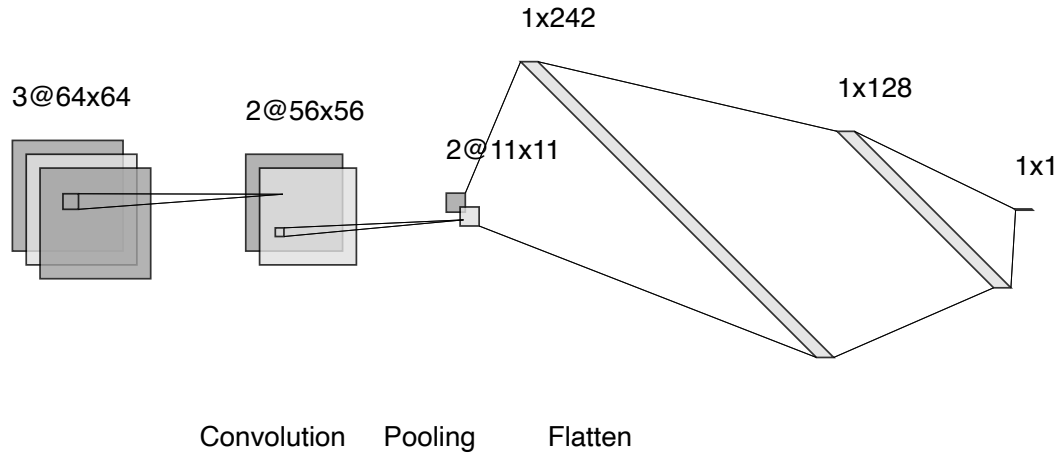


Figure 5.3: An example CNN with a 2D convolutional layer and a pooling layer followed by a fully-connected NN. The convolutional layer has 2 filters of dimensions 9×9 , operating with valid padding and a stride size of 1. The pooling layer has a filter of dimensions 5×5 , operating with valid padding and stride dimensions the same as the pool dimensions. The pooling output is flattened into a vector fed as inputs to a fully-connected NN, which has a hidden layer with 128 neurons and an output layer of 1 neuron. The figure is generated using the NN-SVG tool [19].

5.2 Field-programmable gate array (FPGA)

5.2.1 Basics

A field-programmable gate array (FPGA) is a type of configurable integrated circuit that can be re-programmed multiple times for different algorithms, even after the board is manufactured. An FPGA board consists of various building blocks with different computational functionalities, interconnected as needed to perform specific algorithmic computations. The main advantages of FPGAs are their efficient performance leveraging parallel programming and their flexibility as customizable hardware [120].

Comparing to CPUs and GPUs, which are more general-purpose processors, FPGAs can be configured to have direct hardware implementations for a specific algorithm, reducing unnecessary operations and optimizing implementation efficiency. ASICs, on the other hand, can be more optimized for a specific algorithm for maximum performance and efficiency but cannot be changed once manufactured, making them much less flexible than

FPGAs.

These features make FPGAs particularly attractive for the LHC trigger environment, which requires low-latency algorithms to process and filter collision data concurrently at an input rate of 40 MHz. Additionally, the re-programmable nature of FPGAs grants flexibility for algorithm testing, modifications, and upgrades, which are frequently needed for the trigger systems as the detector conditions and the LHC running conditions are constantly evolving.

Some terminologies and essential building blocks of FPGAs are described below [20, 120].

- **Digital signal processors (DSPs)** are a type of FPGA resources. They are specialized for performing arithmetic operations such as additions and multiplications. They are often more scarce in ML model implementations, which involve many arithmetic operations from matrix computations.
- **Lookup tables (LUTs)** are a type of FPGA resources. They can be used to perform logic functions and can be configured as pre-defined tables that map inputs to outputs via simple memory lookup, potentially saving intensive runtime calculations for complex functions.
- **Flip-flops (FFs)** are a type of FPGA resources. They control data flow using the clock signal to ensure data synchronization and enable proper sequential operations by retaining past data states.
- **Block RAMs (BRAMs)** are a type of FPGA resources. They provide fast data storage and retrieval, essential for executing complex algorithms with low latency.
- **Fixed-point precision** refers to the representation of real numbers with a fixed number of bits for the integer part (I) and a fixed number of bits for the fractional part (F), denoted as $\langle I + F, I \rangle$. Compared to floating-point representation, fixed-point representation allows for reducing the resources needed to represent numbers and perform arithmetic operations.

- **Clock cycle** is the basic unit of time defined by the oscillation period of a clock signal, used to synchronize sequential operations and coordinate the timing of data flow. In the LHC trigger environment, a typical clock frequency of 200 MHz is needed, corresponding to a period of 5 ns for a complete clock cycle.
- **Initiation interval (II)** is the number of clock cycles required to initiate consecutive operations. For highly parallelizable operations, a lower II is desirable, allowing new data to be processed more frequently and reducing latency. For complex operations with high data dependencies, a higher II is necessary to ensure current operations are completed before the next ones can be initiated.
- **Latency** measures the time required for an algorithm to run from input to output, which is crucial for applications with strict time constraints.
- **Reuse factor** determines how many times a resource is reused for multiple operations. A higher reuse factor results in more serial operations, requiring fewer resources but leading to higher latency and lower throughput. Conversely, a lower reuse factor allows for more parallel operations, resulting in lower latency and higher throughput, but requiring more resources.
- **High-level synthesis (HLS)** refers to a design process that allows hardware functionality to be written and manipulated using high-level languages such as C++, greatly simplifying the design and optimization process. HLS tools then convert these high-level descriptions into hardware description languages (HDLs) that describe electronic circuits, which can be synthesized into FPGA configurations.

5.2.2 Machine learning inference on FPGAs with `hls4ml`

`hls4ml` [20, 121] (high-level synthesis for machine learning) is a package that converts ML models written within common ML library frameworks into FPGA implementations using HLS. `hls4ml` is a python package that, combined with HLS tools, reduces the need

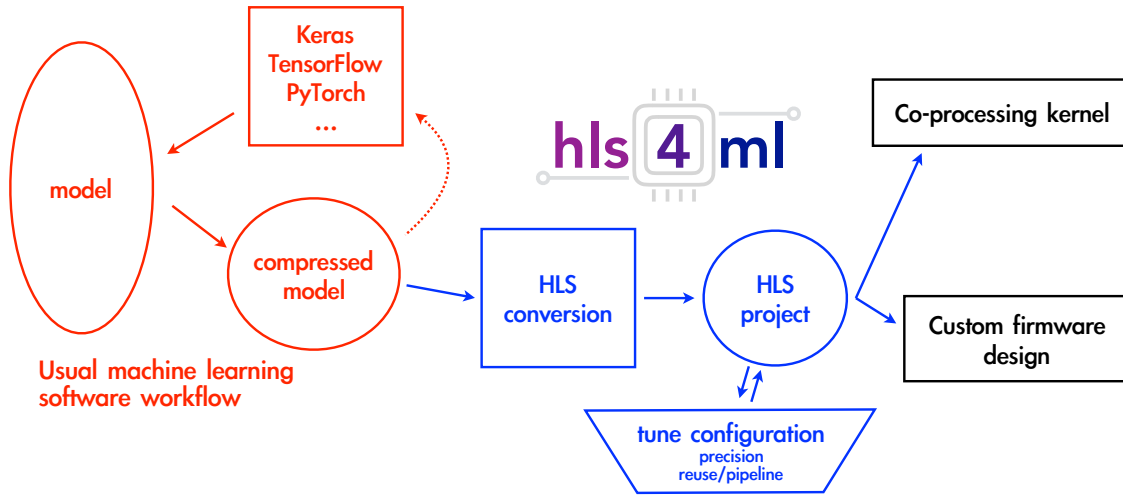


Figure 5.4: `hls4ml` workflow of converting an ML model into an FPGA implementation. The figure is taken from Ref. [20].

for extensive experience in specialized hardware languages. Its easy-to-use feature benefits users from a wide range of scientific and technological areas, including the LHC community.

The workflow is illustrated in Fig. 5.4. ML models are first built and trained using standard ML libraries such as `Keras` [23] and `PyTorch` [122]. `hls4ml` supports automatic HLS conversion of different types of ML architectures, including boosted decision trees [123], fully-connected NNs [20], CNNs [124], and symbolic expressions [25]. Using the `hls4ml` converter, the trained models are then converted into HLS project where the implementation configuration can be tuned and optimized for specific applications. The HLS project can then be synthesized for implementation on FPGAs.

There are three general techniques to optimize for efficient ML inference on FPGAs: compression, quantization, and parallelization [20]. These techniques are essential for environments with scarce resources and tight latency constraints. For instance, pruning is a specific method of model compression to reduce the size of the model [125–127]. ML models can have a large number of parameters, some of the parameters may be redundant and removable without downgrading performance. Unimportant weights can be identified by their small values below a certain threshold, which can be removed. Zeroing these weights effectively reduces the number of arithmetic operations while maintaining model

performance since these pruned weights have minimal impact on the output.

Another general technique is quantization, which involves reducing the precision of the numeric representation and calculations [24, 126, 128]. In most standard ML libraries, 32-bit floating point precision is assumed for representation and calculations. However, this high precision often introduces redundancy, as it is typically not necessary for ML models to achieve optimal performance. Thus, precision can be quantized to fixed precision, taking a fixed and smaller bit width as desired. A technique called quantization-aware training [24] is beneficial for training pre-quantized NNs, as opposed to post-quantization which can lead to significant performance downgrades since the quantization is done without fine-tuning.

Moreover, leveraging parallelization for computations and tuning for the optimal configuration for a specific use case is very useful [20]. This can be controlled by adjusting the reuse factor for operators such as multipliers, which consume DSPs that are often scarce for NN models. For instance, full parallelization can be achieved by setting all multipliers to be reused as minimally as possible so that the multiplications are performed simultaneously. This allows for higher throughput and lower latency but requires more resources. Conversely, full serialization can be achieved by setting all multipliers to be reused as much as possible so that the multiplications are performed sequentially. This requires fewer resources but results in lower throughput and higher latency. Therefore, there is a trade-off between resource utilization and throughput, which can be tuned and optimized for specific problems depending on the performance goal and environmental constraints.

Chapter 6

Anomaly detection in the Level-1 trigger for model-agnostic new physics searches

This chapter presents a novel ML-based algorithm implemented on an FPGA for the CMS Level-1 (L1) trigger system: Calorimeter Image Convolutional Anomaly Detection Algorithm (CICADA) [21]. CICADA uses the anomaly detection technique based on low-level calorimeter information to search for new physics in a model-agnostic way at the trigger level. Alongside the Anomaly eXtraction Online Level-1 Trigger Lightweight (AXOL1TL) [129] algorithm, which follows the same principle but is based on global trigger information, this marks the first time CMS has implemented such a new trigger strategy, distinct from the existing methods. The anomaly detection triggers have shown potential in detecting both rare and new physics signals in a completely different manner, capable of capturing unconventional experimental signatures that might escape traditional trigger selection. Both CICADA and AXOL1TL are deployed during the ongoing Run 3 in CMS, initiating a novel approach to hunting for new physics at the LHC.

The motivation for the anomaly detection trigger is discussed in Sec. 6.1. Sec. 6.2 introduces the concept of anomaly detection in ML and its applications at the LHC.

Then, CICADA is introduced. Sec. 6.3 describes the datasets used. Sec. 6.4 details the model architecture and training process. Sec. 6.5 presents the model performance. Sec. 6.6 provides a summary and discusses future aspects.

6.1 Motivations

Although a significant milestone was achieved in 2012 by ATLAS and CMS with the discovery of the Higgs boson [29–31], the LHC has yet to discover any new physics beyond the Standard Model (BSM).

There are three possibilities: 1) the new physics occurs at a scale beyond the current LHC energy scale, 2) new physics is present in the data but there are not enough data samples collected to reveal it statistically, or 3) new physics is present in the data but we have been looking in the wrong places or using the wrong event selections. Both 1) and 2) can be addressed by increasing the collision energy and collecting more data for future analysis. However, 3) is a more systematic problem, which, if not properly addressed, would hinder our ability to discover new physics regardless of how high collision energy is or how much more data is collected.

This systematic problem is particularly critical at the L1 trigger. The L1 trigger performs the first round of event selection immediately after collisions occur. The trigger algorithms determine whether the 40 MHz collision events contain potential physics signature and should be stored for later analysis or discarded immediately. If the trigger algorithms are not capable of detecting new physics and discard those events, they will be lost permanently and never be available for offline analysis, no matter how capable the offline analysis techniques are. Therefore, the anomaly detection algorithms are developed and targeted for the L1 trigger system as a front guard to preserve as many potential new physics events as possible [21, 129].

With the targeted system identified, the ML-based anomaly detection technique [116] is chosen to address the issue for several reasons. First, ML is data-driven, allowing models to make decisions based on patterns present in the data, thus minimizing human

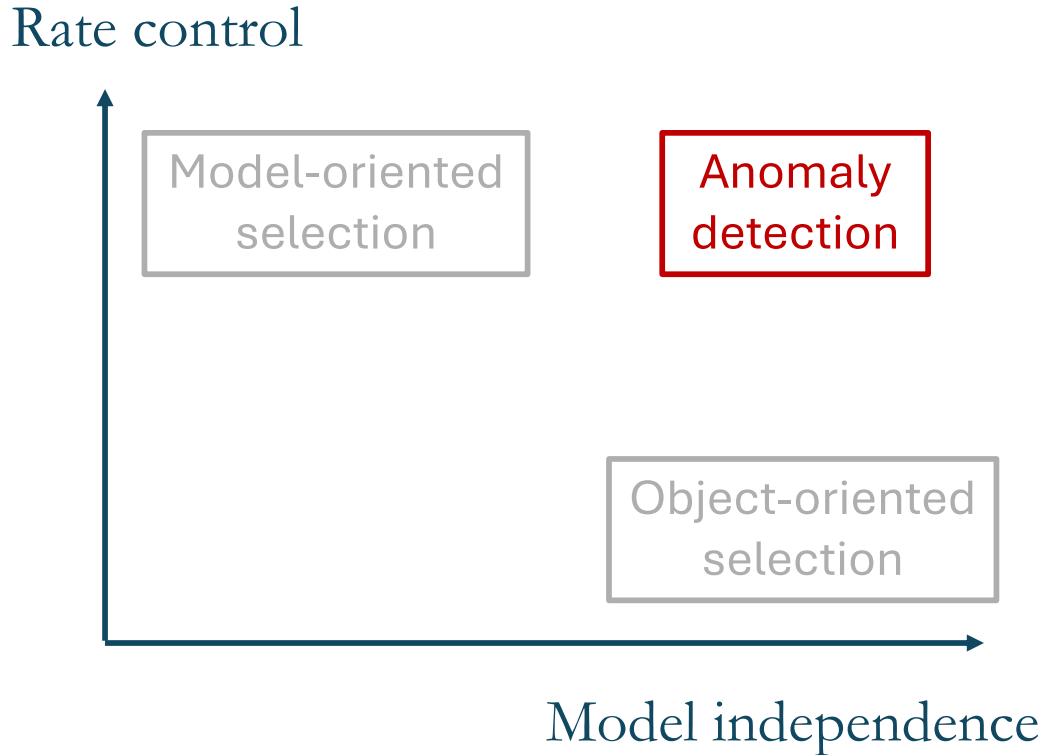


Figure 6.1: Comparison of anomaly detection trigger and traditional trigger in terms of rate control and model independence. The plot is for illustration purposes and is not drawn to scale.

interventions and reducing bias. ML models excel at learning unknown and complex correlation and have already proven very successful in tackling problems with complicated data structures [112]. This contrasts with traditional triggers that rely on simple cuts on object momentum. The anomaly detection technique does not require targeting a specific physics model, making it a model-agnostic method capable of identifying all possible new physics signatures that deviate from the abundant standard physics processes.

Traditional triggers have mainly relied on two strategies: model-oriented selection and object-oriented selection [11]. The pros and cons of the two strategies, along with the new anomaly detection strategy, are compared in terms of trigger rate control and model independence, as shown in Fig. 6.1, and discussed below.

- **Model-oriented selection** targets a specific known physics model or a class of

models sharing a well-defined physics signature in a collision event. For instance, the dedicated vector boson fusion (VBF) trigger requires the presence of a pair of forward-backward jets with a large invariant mass. This trigger specifically targets VBF-like topologies and is highly efficient for signals from specific models, such as Higgs boson production via VBF. The precise and narrow phase space results in a low trigger rate, minimizing the burden on the L1 trigger system, which is limited to a maximum output rate of 100 kHz. However, this type of trigger only works within a dedicated phase space, which is often very limited and cannot be generalized to other analyses. In the context of new physics searches, where models may exhibit a wide range of different topologies or where the true models are unknown, this trigger strategy can become inefficient.

- **Object-oriented selection** targets specific physics objects or combinations of physics objects and selects them based on their kinematic properties. For example, the analysis presented in Ch. 3 relies entirely on such triggers, such as single- μ and cross- $\mu\tau_h$ triggers in the $\mu\tau_h$ final state (see Sec. 3.1.3). These triggers are more general-purpose and model-independent, as they can capture a wide range of signals when the relevant objects are present in the final state and pass simple kinematic cuts, such as minimum- p_T thresholds. However, these triggers, which can select a wide range of events, typically require high thresholds on the object p_T to prevent the trigger rate from exceeding the L1 output limit. This significantly lowers the signal acceptance, especially for analyses involving low- p_T final-state objects. The analysis presented in Ch. 3 exemplifies this issue, as it involves low- p_T final-state objects originating from light pseudoscalar decays. The most sensitive $\tau_h\tau_h$ final state is completely discarded due to the excessively high p_T thresholds in τ_h -based triggers, and over 90% of signal events are removed by the triggers in the other analyzable final states (see Fig. 3.29).
- **Anomaly detection** can achieve both effective rate control and model independence while maintaining physics sensitivity, thus addressing the issues associated

with the two traditional trigger strategies mentioned above. This approach does not rely on any specific physics models as inputs when building the trigger, nor is it constrained to select a particular set of physics objects, making it model independent. Additionally, by leveraging the power of ML to automatically identify hidden correlations in data, anomaly detection triggers exhibit high physics sensitivity compared to simple cut-based selections. As a result, signal acceptance for a wide range of signals can potentially remain high even at a relatively low trigger rate, avoiding significant compromise on challenging signals, such as those with low- p_T objects, to suppress the trigger rate. The concept of anomaly detection will be explained in the next section.

6.2 Anomaly detection

Anomaly detection [116] is a data analysis technique used to identify rare or unexpected patterns within a large set of similarly distributed events, as illustrated in Fig. 6.2.

There are two classes of events in anomaly detection: normal and anomalous. It is assumed that the majority of events in a dataset follow the same or similar distributions, referred to as normal events. Outliers, which deviate significantly from the majority, indicate critical incidents and are referred to as anomalous events. For example, fraud in a credit card account can be detected through unusual spending patterns, such as a high-value purchase in foreign currency or multiple high-value purchases within a short period.

In the context of the LHC, collisions are dominated by uninteresting background processes with large cross sections (see Fig. 2.2 for cross section measurements of SM processes), such as soft QCD, constituting the normal events. Interesting physics processes with unconventional signatures occur at low probabilities and can naturally be treated as anomalies to be detected, these include rare SM processes such as Higgs boson production, and BSM processes such as dark matter production with large missing energy.

The unsupervised anomaly detection technique is of particular valuable for new physics

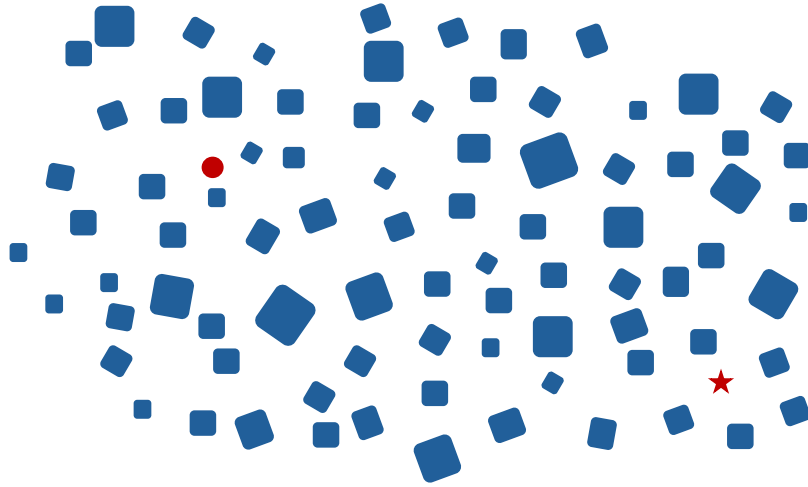


Figure 6.2: Illustration of anomaly detection. Anomalous data (rare occurrences: non-rectangular shapes, in red) are identified as deviations from normal data (abundant occurrences: rectangular shapes, in blue). In the context of the LHC, normal events correspond to abundant SM processes with large cross sections, such as soft QCD productions. Anomalous data, on the other hand, correspond to rare SM processes or BSM processes with signatures deviating from normal data.

searches at the LHC. In this approach, ML models are trained on unlabeled and unfiltered collision data to learn patterns from the abundant normal events. In theory, these ML models can detect anomalous events from all rare or new physics processes simultaneously, as long as their signatures deviate from the normal events, since no specific physics models are assumed during training. This method allows for model-agnostic new physics searches by capturing as many potential events as possible for offline analysis, reducing the risk of permanently discarding significant events through traditional triggers.

6.3 Datasets

The input features to CICADA [21] are sourced from the calorimeter layer-1 trigger subsystem in CMS, abbreviated as CaloLayer-1. CaloLayer-1 receives low-level information, such as energy deposits at the trigger tower level, from the ECAL and HCAL (see Fig. 2.9

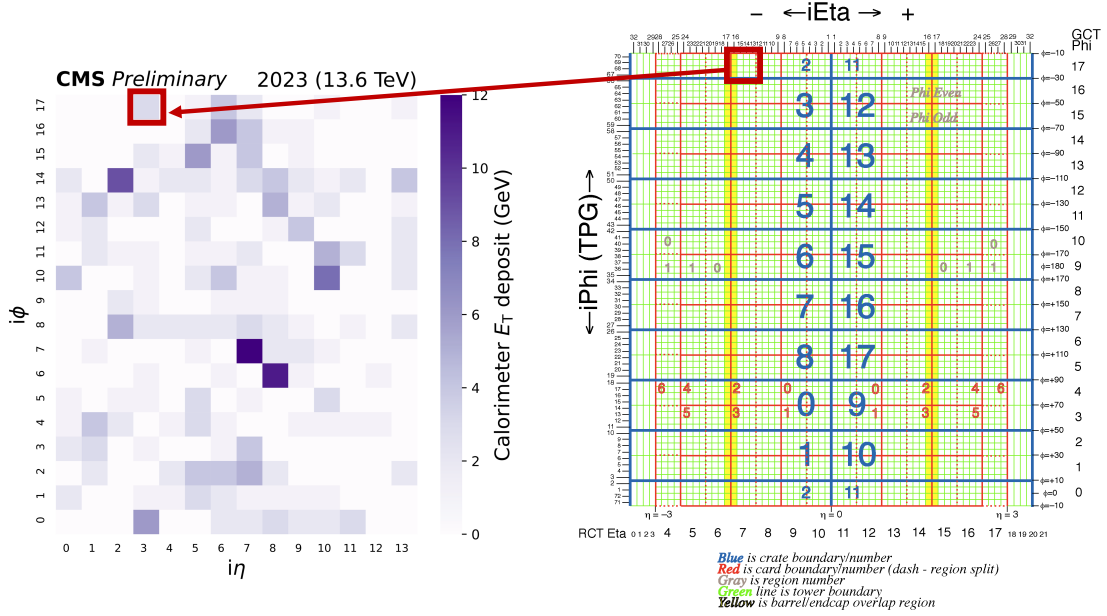


Figure 6.3: Input features to CICADA: energy deposits at the trigger tower level in the calorimeter layer-1 trigger subsystem, received from the ECAL and HCAL. Each input event consists of energy deposits in the $18(i\phi) \times 14(i\eta) = 252$ trigger regions, where each region is a 4×4 trigger towers. Here, $i\phi$ and $i\eta$ refer to special trigger coordinates, which take integer values to label the discrete region locations. The heat map displays an example of a ZeroBias data event, with the energy deposit values indicated by color intensity. The correspondence of a trigger region is marked by red squares in both the input heat map (left) and the trigger tower map (right). The figure is adapted from Ref. [21]

for the L1 trigger subsystems). Fig. 6.3 (right) illustrates the geometric coordinate system for CaloLayer-1 in $i\phi$ and $i\eta$, which take integer values to represent discrete trigger regions. Each trigger region encompasses 4 trigger towers. There are $18(i\phi) \times 14(i\eta) = 252$ trigger regions in total, with each region containing the summed energy deposits from both ECAL and HCAL. These 252 energy deposits serve as the input features per event for CICADA.

For unsupervised anomaly detection, the training data consist entirely of ZeroBias data, which are unfiltered collision events, removing any bias toward specific physics signatures that might be introduced by trigger selection. ZeroBias data are real collision events instead of simulated ones, meaning their true underlying physics processes are unknown and cannot be correctly labeled. However, the expected occurrence of different known processes can be roughly estimated from SM cross section measurements (see

Fig 2.2), which are predominantly QCD processes. Although ZeroBias data also include rare SM processes, such as Higgs signals that are potential events of interest, or may even contain tiny and unknown signals from new physics processes, these can still be detected as anomalies during inference. This is because, during training, their contribution is negligible compared to the vast number of QCD events, which have orders of magnitude higher cross sections.

The ZeroBias datasets collected during Run 3 in 2023 are used for training. The energy deposits in the inputs represent low-level information that does not involve any high-level reconstruction entities, such as electron or jet features, minimizing potential bias introduced by other algorithms. An example input feature map from a ZeroBias event is shown in Fig. 6.3 (left).

Background events, such as soft QCD processes, form the bulk of ZeroBias data, typically resulting in low-energy deposits across many trigger regions due to multiple soft jets. On average, ZeroBias events exhibit low-energy and noisy patterns in the regional map. In contrast, interesting events generally involve a high momentum transfer in the hard scatter process, resulting in more distinct deposit patterns due to the high-energy objects or high-multiplicity objects in the final state.

For evaluation, five simulated signals are chosen to represent a broad range of different experimental signatures, demonstrating the generalization capability of the anomaly detection trigger. These signals include both rare SM and BSM processes: 1) Higgs boson decays into two long-lived particles, which then decay into four b quarks (HTo2LongLivedTo4b), 2) soft unclustered energy patterns (SUEP), 3) BSM Higgs boson production in association with two b quarks, where the Higgs boson decays into two b quarks (SUSYGluGluToBB-HtoBB), 4) top quark pair production (TT), and 5) Higgs boson production via VBF, where the Higgs boson decays into two c quarks (VBFHto2C).

The SUEP signal, in particular, presents an unconventional experimental signature, involving multiple low-energy jets depositing energies in the calorimeters that are typically too soft to be clustered into objects due to the energy threshold in the clustering algo-

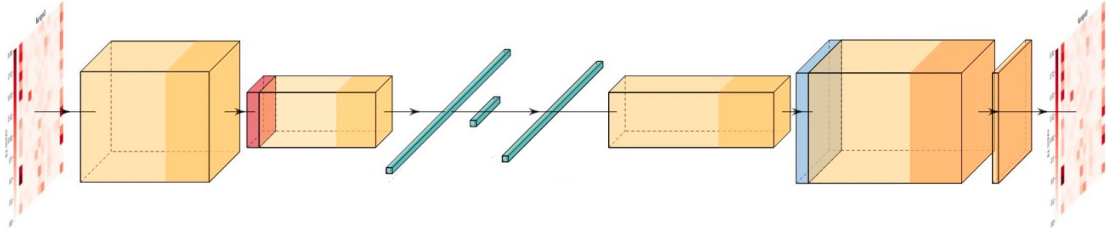


Figure 6.4: Illustration of the autoencoder that compresses and then reconstructs the regional energy deposits: input \rightarrow CNN encoder \rightarrow latent space \rightarrow CNN decoder \rightarrow output. The figure is generated using the PlotNeuralNet tool [22].

rhythms [130, 131]. This type of signal is easily rejected by traditional triggers since there are no reconstructed objects to be selected, leaving only faint patterns in the regional map. Therefore, it serves as an excellent example to demonstrate the performance of the anomaly detection trigger.

6.4 Model architecture and training

6.4.1 Autoencoder

The autoencoder architecture [132] has the potential to perform unsupervised new physics detection at the LHC [133]. An autoencoder is a type of NN architecture that consists of an encoder and a decoder. The encoder is an NN that transforms the input into a compressed representation called latent space, which encodes the essential information of the input. The decoder is an NN that mirrors the encoder to decompress the latent representation and reconstruct the input. Essentially, the autoencoder is designed to learn how to reconstruct its input. The size of the latent space is typically smaller than the input size to ensure effective compression; otherwise, the autoencoder might learn a trivial identity function.

Since the CICADA inputs are image-like data (see Fig. 6.3), CNNs are used to form both the encoder and decoder. Each input is represented by a tensor with dimensions of $18 \times 14 \times 1$, retaining the regional map geometry of $18(i\phi) \times 14(i\eta)$, with one channel representing the energy deposits per region. Fig. 6.4 illustrates the autoencoder model

used to reconstruct the regional energy information.

Fig. 6.5 shows the configuration of the autoencoder implemented in Keras [23]. The CNN encoder starts with a convolutional layer containing 20 3×3 filters, with same padding and a stride size of 1, followed by ReLU activation (see Eq. 3.5). An average pooling layer with 2×2 filters is then applied to reduce the dimensions of the convolution output by a factor 2. Another convolutional layer with 30 3×3 filters is applied, followed by ReLU activation. The convolution output is then flattened into a 1D array and fed into a dense layer with 80 neurons, activated by the ReLU function. The output layer, which forms the latent space, has a dimension of 80, smaller than the regional input with $18 \times 14 \times 1 = 252$ features. The decoder mirrors the encoder architecture, using an upsampling layer to increase the image dimensions through interpolative resizing. The autoencoder combines the encoder and decoder networks, with the output retaining the same dimensions as the input.

There are more than 300,000 trainable parameters, which are the filter weights and neuron connection weights. These weights are optimized using the Adam optimizer [95] to minimize the reconstruction loss, computed by the mean squared error (MSE) between the inputs and outputs:

$$l(\mathbf{x}, \hat{\mathbf{y}}) = \frac{1}{n} \sum_i (x_i - \hat{y}_i)^2, \quad (6.1)$$

where \mathbf{x} and $\hat{\mathbf{y}}$ represent the input and output tensors, respectively, and the sum runs over the $n = 252$ trigger regions per event. Approximately one million ZeroBias data events collected in Run 3 during 2023 are used for the training.

The reconstruction performance is quantified by the MSE, which is used as the anomaly score for detecting anomalous events. ZeroBias events are expected to have low anomaly scores because the autoencoder is trained to reconstruct these events well. In contrast, rare SM or BSM events are expected to have high anomaly scores because they are assumed to be effectively absent in the training, making the autoencoder less capable of accurately reconstructing their unfamiliar patterns.

Fig. 6.6 shows the reconstruction performance of the autoencoder after training. For

Layer (type)	Output Shape	Param #
input (InputLayer)	[(None, 18, 14, 1)]	0
conv2d_1 (Conv2D)	(None, 18, 14, 20)	200
relu_1 (Activation)	(None, 18, 14, 20)	0
pool_1 (AveragePooling2D)	(None, 9, 7, 20)	0
conv2d_2 (Conv2D)	(None, 9, 7, 30)	5430
relu_2 (Activation)	(None, 9, 7, 30)	0
flatten (Flatten)	(None, 1890)	0
latent (Dense)	(None, 80)	151280
dense (Dense)	(None, 1890)	153090
reshape2 (Reshape)	(None, 9, 7, 30)	0
relu_3 (Activation)	(None, 9, 7, 30)	0
conv2d_3 (Conv2D)	(None, 9, 7, 30)	8130
relu_4 (Activation)	(None, 9, 7, 30)	0
upsampling (UpSampling2D)	(None, 18, 14, 30)	0
conv2d_4 (Conv2D)	(None, 18, 14, 20)	5420
relu_5 (Activation)	(None, 18, 14, 20)	0
output (Conv2D)	(None, 18, 14, 1)	181
=====		
Total params: 323,731		
Trainable params: 323,731		
Non-trainable params: 0		

Figure 6.5: Configuration of the autoencoder model for CICADA implemented in Keras [23].

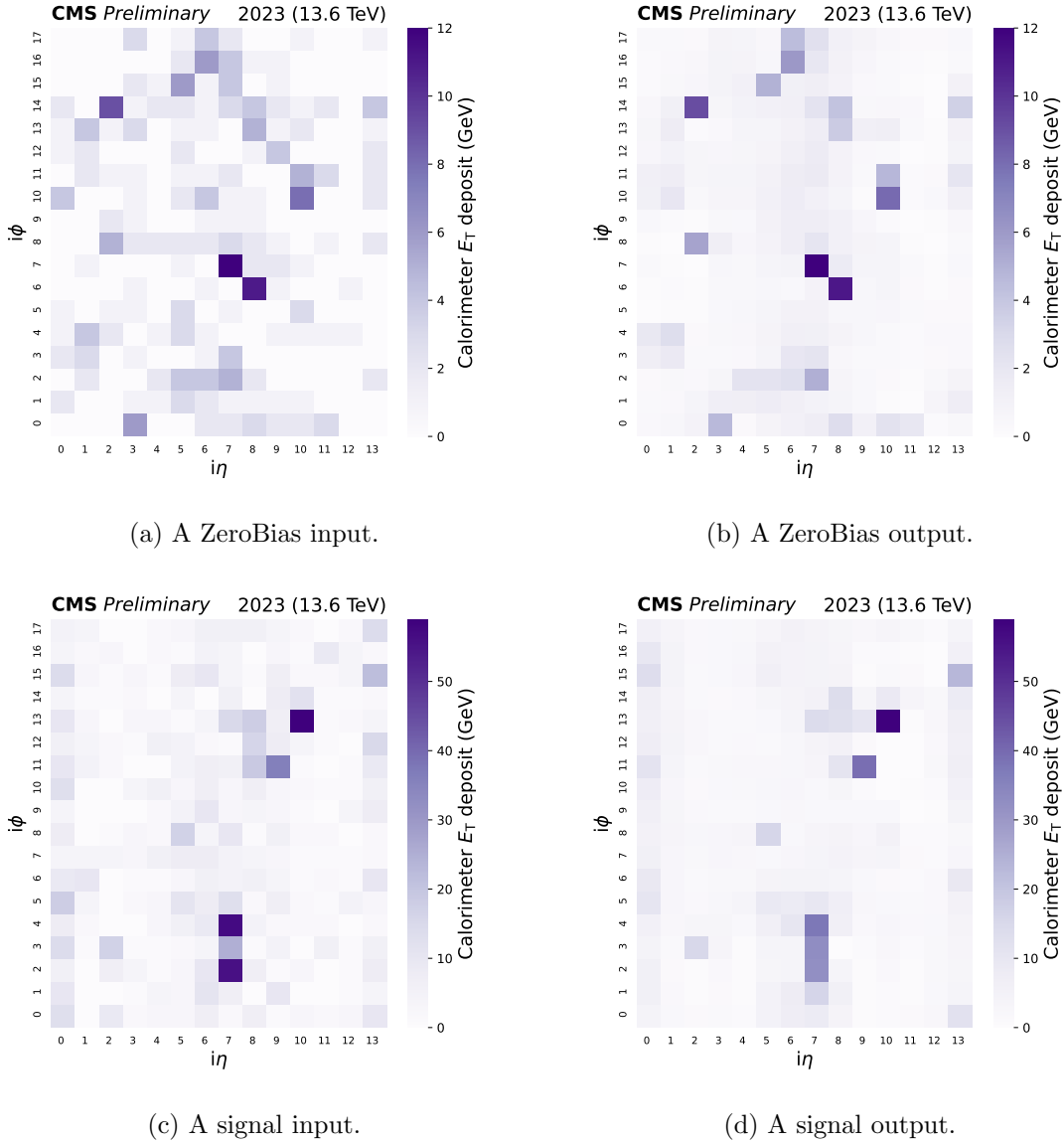


Figure 6.6: Comparison of inputs and reconstruction outputs from the autoencoder. In the ZeroBias event, the reconstruction is fairly good, with the major high-energy regions well-reconstructed in both intensity and location (anomaly score = 0.81). In the signal event, which is simulated SUEP, the model fails to reconstruct the high-energy regions in the lower part (anomaly score = 14.21). The figures are taken from Ref. [21].

the ZeroBias event, the model captures and reconstructs the major high-energy deposits in both intensity and location, resulting in an anomaly score of 0.81. Conversely, an example BSM event with an unconventional experimental signature, known as SUEP, shows poor reconstruction, where some major high-energy deposits are diluted and blurred, resulting

in a higher anomaly score of 14.21.

6.4.2 Model compression

In principle, the anomaly score computed from the autoencoder outputs can be used as a discriminant to trigger anomalous events at runtime. However, the autoencoder, with over 300,000 parameters, is too large to be implemented on an FPGA board with limited resources and to run within the tight sub-microsecond latency constraint. Additionally, computing the MSE involves summing the differences across the 252 regions and squaring them, which adds further computation time to the algorithm. Therefore, the model needs to be compressed to a much smaller size to be deployable.

First, a model compression technique called knowledge distillation [134, 135] is employed. Instead of implementing the oversized autoencoder, referred to as the teacher model, its knowledge is transferred to a smaller model, referred to as the student model, as illustrated in Fig. 6.7. In this setup, the teacher model, which has already been trained, is capable of learning complex patterns from the data due to its high expressivity and large number of trainable parameters. The student model takes the same input tensors as the teacher model but follows with much fewer layers of operations, ultimately outputting a single number. The anomaly score (MSE) is calculated for each training event using the outputs of the teacher model and is used as a label when training the student model, which learns to directly output the anomaly score, bypassing the MSE calculation. Thus, the student model is trained under the supervision of the teacher model to directly regress the anomaly score.

Next, the student model is further compressed by quantization [24, 126, 128]. The teacher model does not need to be quantized since it is not intended for deployment. However, quantization is essential for the student model to be deployed, as it limits the numerical representation and calculations to shorter and fixed bit widths. This conserves FPGA resources, such as DSPs, which are typically scarce for NNs. The bit widths need to be carefully determined: a too short bit width can prevent the model from learning useful

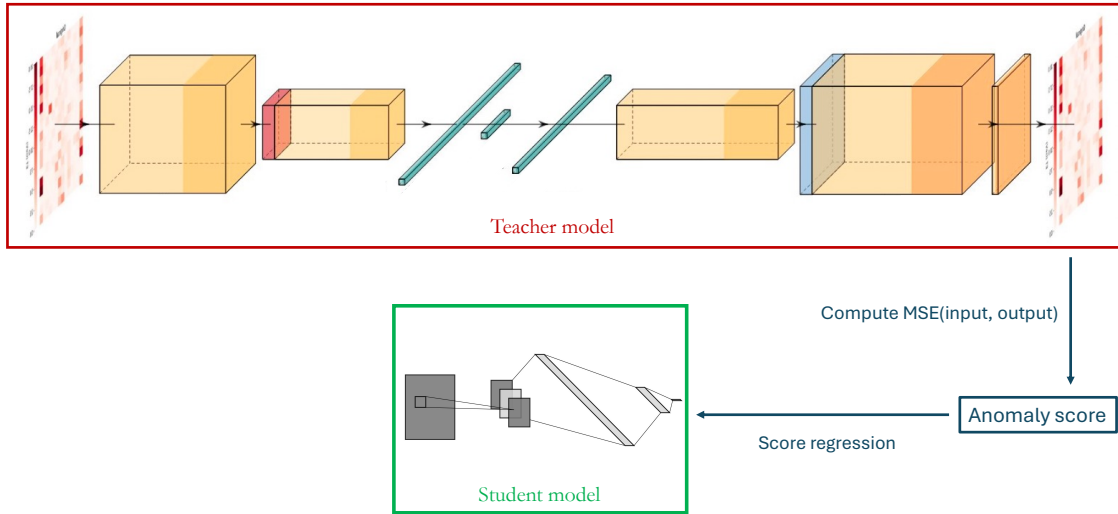


Figure 6.7: Illustration of knowledge distillation: the smaller model, called the student model, learns to directly regress the anomaly score produced by the larger autoencoder model, called the teacher model. The figure is generated using the PlotNeuralNet [22] and the NN-SVG [19] tools.

patters, while a too long bit width can lead to high resource utilization. The regional energy received at CaloLayer-1 is 10-bit long, taking integer values between 0 and 1024.

The optimal architecture of the student model is determined using the **AutoQKeras** framework [24], which treats quantization configuration as hyperparameters in the hyperparameter search in **KerasTuner** [98]. The student model is trained using **Adam** [95] to minimize the MSE between the anomaly score from the teacher model and the output of the student model.

Fig. 6.8 shows the configuration of the final student model implemented in **QKeras** [24]. The input layer is a vector with a length of 252, matching the data format received from hardware at CaloLayer-1. The input vector is first reshaped to a tensor with dimensions of $18 \times 14 \times 1$ to be fed into a convolutional layer with 3×3 filters, assuming valid padding and a stride size of 2, followed by ReLU activation. This output is then flattened and passed to a dense layer with 20 neurons, also followed by ReLU activation. The final output layer has one neuron. In both the convolutional and dense layers, the bias terms are removed to reduce the number of parameters in the model. All filter weights, neuron

Layer (type)	Output Shape	Param #
In (InputLayer)	[(None, 252)]	0
reshape (Reshape)	(None, 18, 14, 1)	0
conv (QConv2D)	(None, 8, 6, 3)	27
relu1 (QActivation)	(None, 8, 6, 3)	0
flatten (Flatten)	(None, 144)	0
dense1 (QDense)	(None, 20)	2880
relu2 (QActivation)	(None, 20)	0
output (QDense)	(None, 1)	20
=====		
Total params: 2,927		
Trainable params: 2,927		
Non-trainable params: 0		

Figure 6.8: Configuration of the student model for CICADA implemented in QKeras [24], which is the final model to be converted into an FPGA implementation.

connection weights, and the ReLU function are quantized to 16 bits, with 4 bits for the integer part. The weights connecting to the output are quantized to 16 bits, with 2 bits for the integer part.

6.5 Performance

The final student model of CICADA is converted into HLS using `hls4ml` [20, 121] and synthesized with Vivado HLS (2020.1) [136] on a Xilinx Virtex-7 FPGA. The resource utilization and latency (in the order of 100 ns) are well within the available budget, meeting the stringent L1 constraints and allowing room for further modifications and improvements in the future.

Evaluation is conducted on the test set of ZeroBias data and various simulated signal

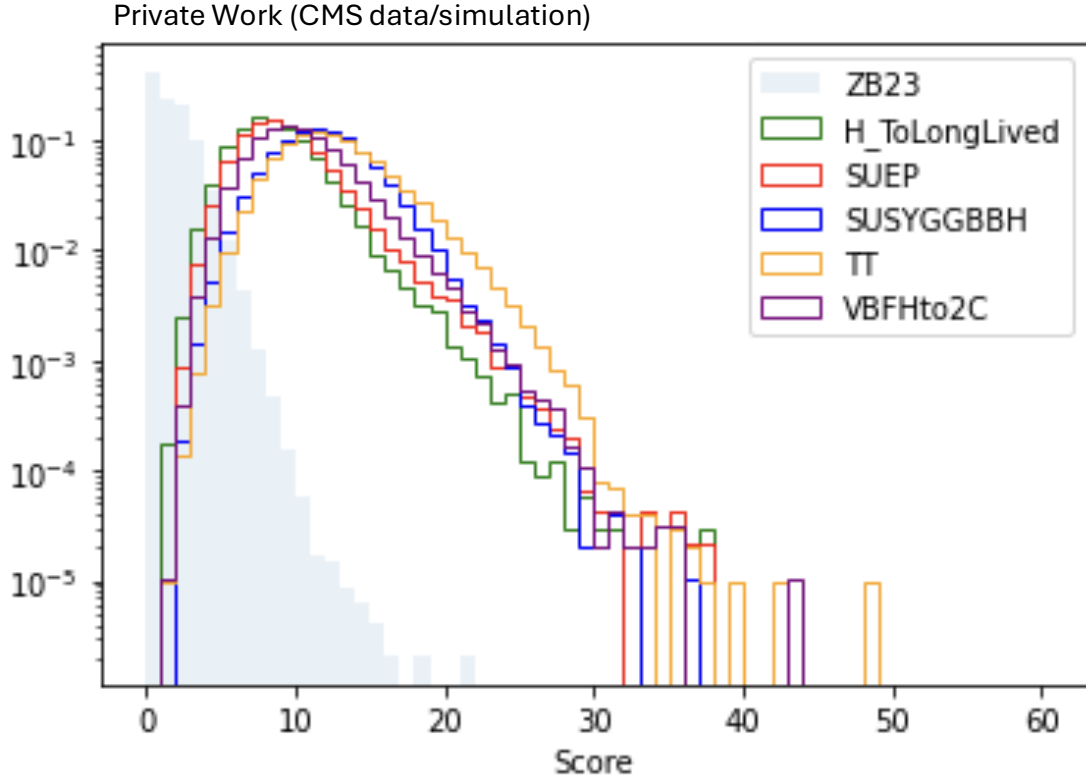


Figure 6.9: Anomaly score distributions comparing ZeroBias data and various signals.

samples mentioned in Sec. 6.3. Fig. 6.9 shows the score distributions. As expected, ZeroBias events have low scores close to 0, while all signal events peak at higher score values and have a long tail in the high score region. This anomaly score serves as a discriminant, allowing various working points to be defined for the trigger.

Fig. 6.10 shows the ROC curves, which plot signal efficiency against background efficiency, with background referring to ZeroBias data. On the horizontal axis, the background efficiency is converted into trigger rate in MHz, which is useful for defining working points for the trigger.

The entire L1 trigger system includes numerous trigger algorithms targeting different experimental signatures, such as muons and jets, so the total bandwidth is shared by many individual trigger paths [11]. Therefore, the new trigger path cannot operate at a high trigger rate in order to comply with the L1 constraints. The vertical line in Fig. 6.10 indicates an output rate of 3 kHz for CICADA. Below this line is the region of interest

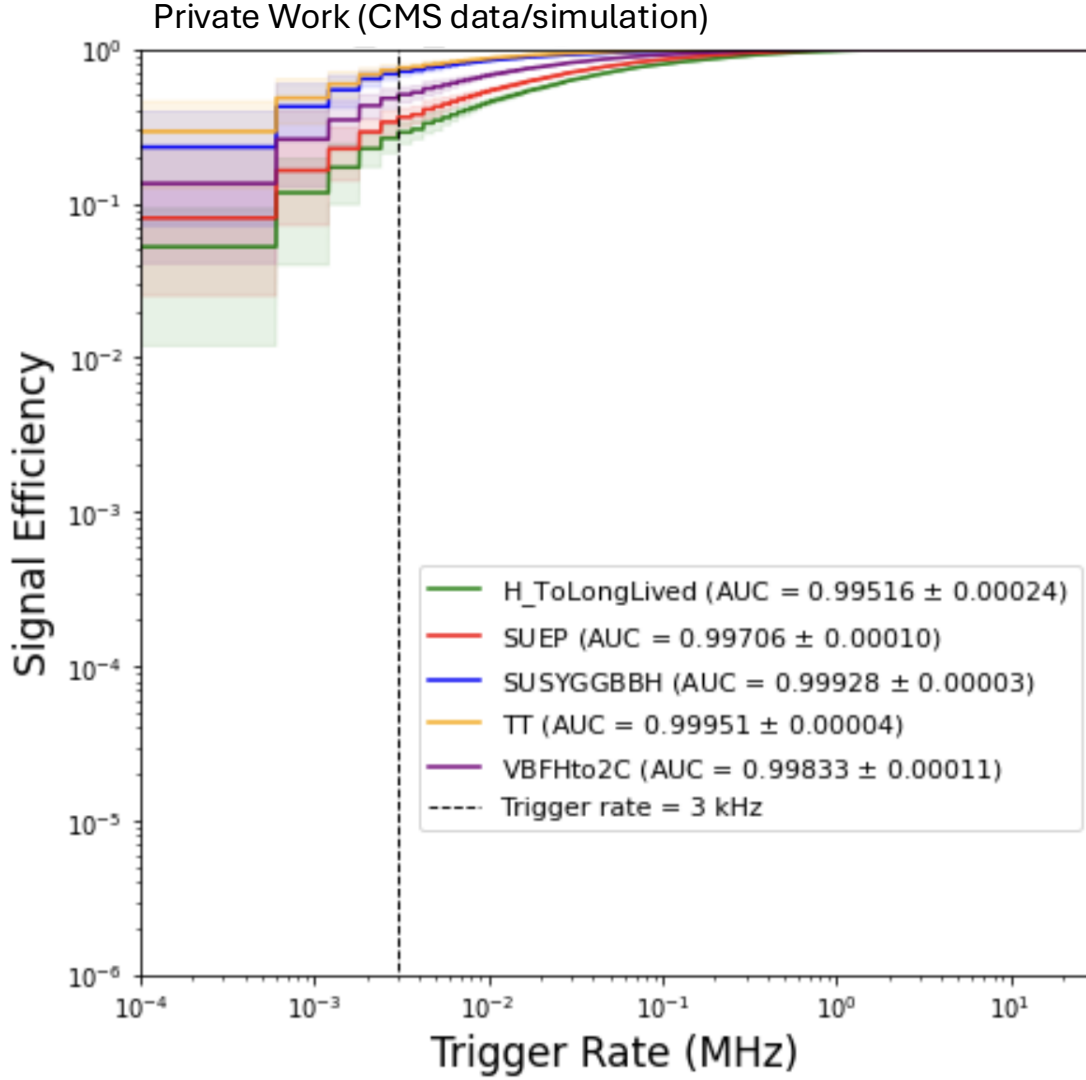


Figure 6.10: ROC curves plotted for various signals vs. ZeroBias data. The trigger rate on the horizontal axis refers to the rate of triggering on ZeroBias events in MHz, with the dashed line indicating a trigger rate of 3 kHz. Practically, the anomaly detection trigger is expected to operate at a trigger rate around and below this line due to the limited bandwidth shared by many different physics groups.

where working points can be defined. It is evident that CICADA is very sensitive to all the signals, maintaining signal efficiencies above 10% even at a low output rate of 1 kHz from an input rate of around 25 MHz.

The HT trigger, which selects events based on the scalar sum of p_T of jets [11], is one of the common traditional triggers and is chosen for comparison with CICADA. Fig. 6.11

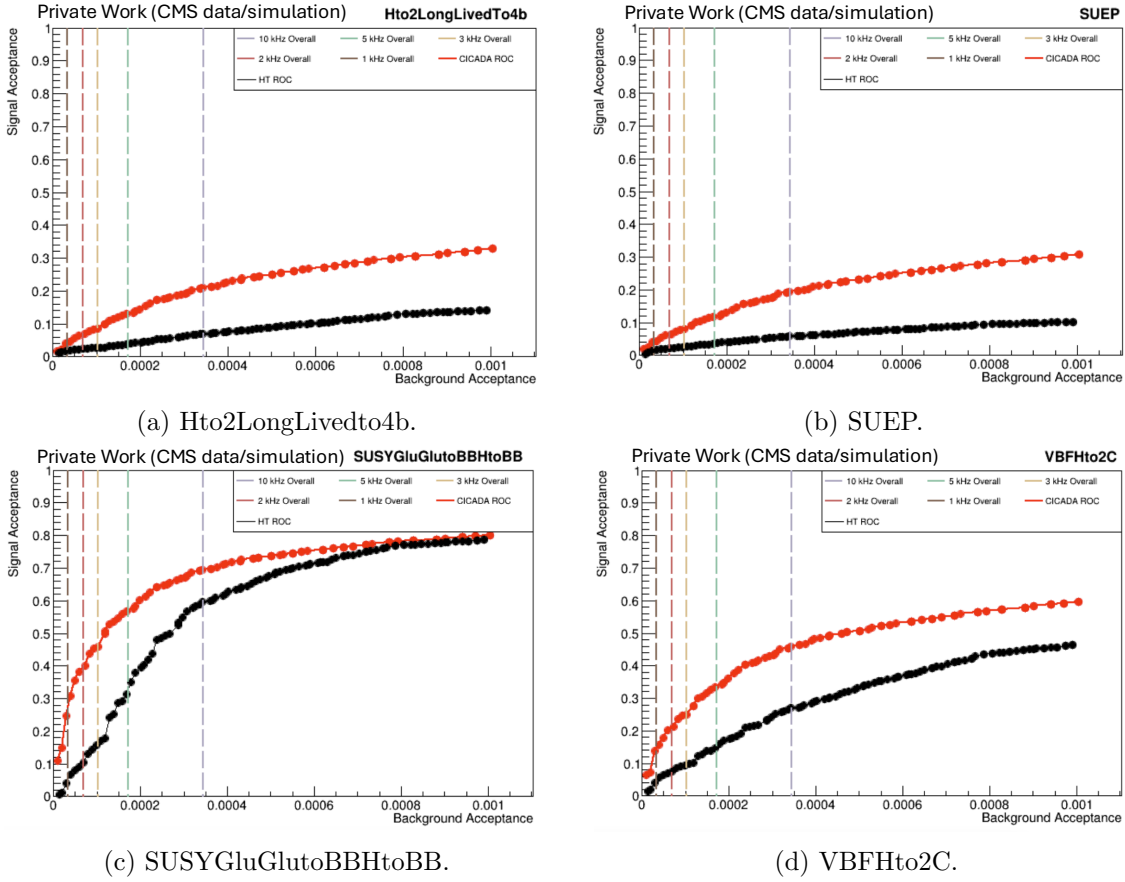


Figure 6.11: ROC curves in the region of low background efficiency, comparing CICADA (red) and the HT trigger (black). Several practical CICADA rates are indicated by the vertical dashed lines.

shows the ROC curves focusing on the region of low background efficiency. CICADA achieves significantly higher signal efficiencies across different signals with varying signatures, especially for signals with low-energy jets in the final states, where the HT trigger is not efficient. This is because CICADA identifies anomalies based on the deposit energy and the deposit pattern. The additional gain in CICADA compared to the HT trigger can be attributed to detecting anomalies in deposit patterns, which the HT trigger is not sensitive to. Therefore, the HT trigger is more comparable to CICADA for signals with energy anomalies dominating over pattern anomalies, such as high mass BSM Higgs boson decays (SUSYGluGluBBHtoBB). However, CICADA shows a significant advantage for signals with low-energy deposits that have pattern anomalies, such as SUEP.

It is also interesting to investigate what particular signatures CICADA prefers to capture as anomalies. Fig. 6.12 shows the distributions of the L1 jet multiplicity and p_T before and after applying the CICADA trigger. CICADA prefers a high multiplicity of jets as well as jets with high p_T in the final state. However, unlike traditional triggers, which typically apply hard threshold cuts on the object p_T resulting in zero acceptance below the threshold, CICADA remains sensitive to the entire spectrum of jet p_T , including regions with very low jet p_T . This suggests that CICADA can detect signals with very low- p_T jets that would have been rejected by traditional cut-based triggers. These conclusions also apply to other L1 objects such as electrons/photons and taus, but not muons since they do not deposit energy in the calorimeters. Essentially, CICADA prefers final states with a high multiplicity of objects and objects with high p_T , but it does not lose significant acceptance to low- p_T objects.

As the trigger system involves many different trigger paths that capture different signatures, it is crucial to understand the overlap between CICADA and all existing triggers. Fig. 6.13 shows the CICADA pure rate as a function of the overall rate. The pure rate corresponds to events triggered only by CICADA and not by any other existing trigger paths, representing the purity gain by adding CICADA to the trigger system. The overall rate refers to the total events triggered by CICADA regardless of whether other trigger paths are also fired. A few benchmark points are indicated: an overall rate of 5 kHz includes a pure rate of 3 kHz.

As a trigger algorithm, the stability of CICADA needs to be demonstrated to ensure that it does not produce unexpected behaviors, such as a sudden rate hike during a collision run. As seen in Fig. 6.14, the CICADA score distributions remain stable across three different runs in 2023. Runs B, C, and D refer to data-taking periods from April-May, May-June, and June-August in 2023, respectively, where running conditions can change during and across these periods. If the CICADA score threshold is defined in one run, the output rate would remain almost constant in other runs, indicating that CICADA would not undergo significant rate changes under typical running conditions. This stability is

crucial for maintaining a stable total rate below the limit for the entire trigger system. The stable CICADA rate also allows for pre-defined working points to adapt different running conditions, proving flexibility.

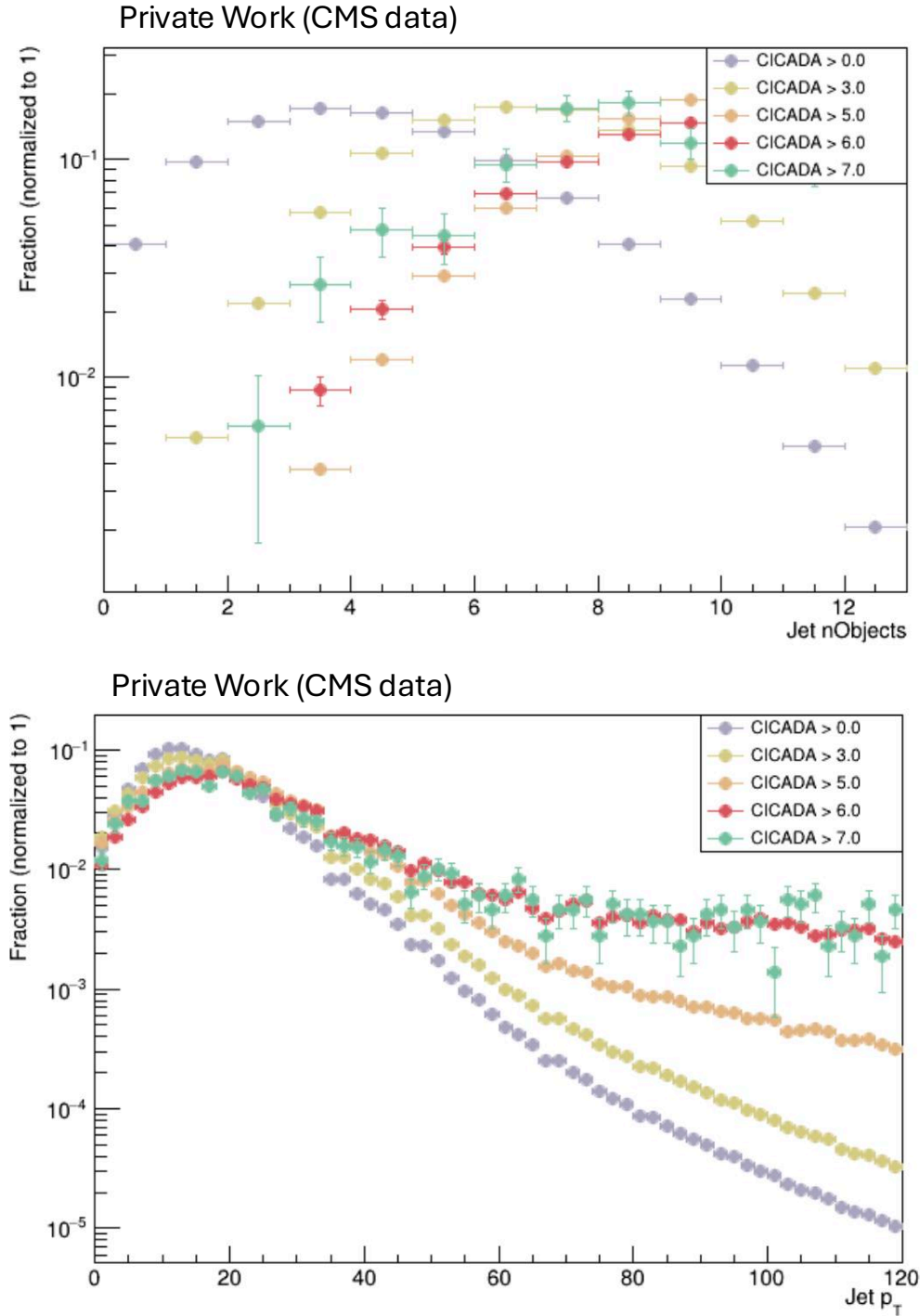


Figure 6.12: Signatures that CICADA prefers to trigger on. Top: trigger-level jet multiplicity profiles before and after applying the CICADA trigger. Bottom: jet p_T profiles before and after applying the CICADA trigger. The histograms in purple show the Zero-Bias profiles where no threshold cut is applied on the CICADA score. The histograms in other colors show the profiles at various threshold cuts on the CICADA score.

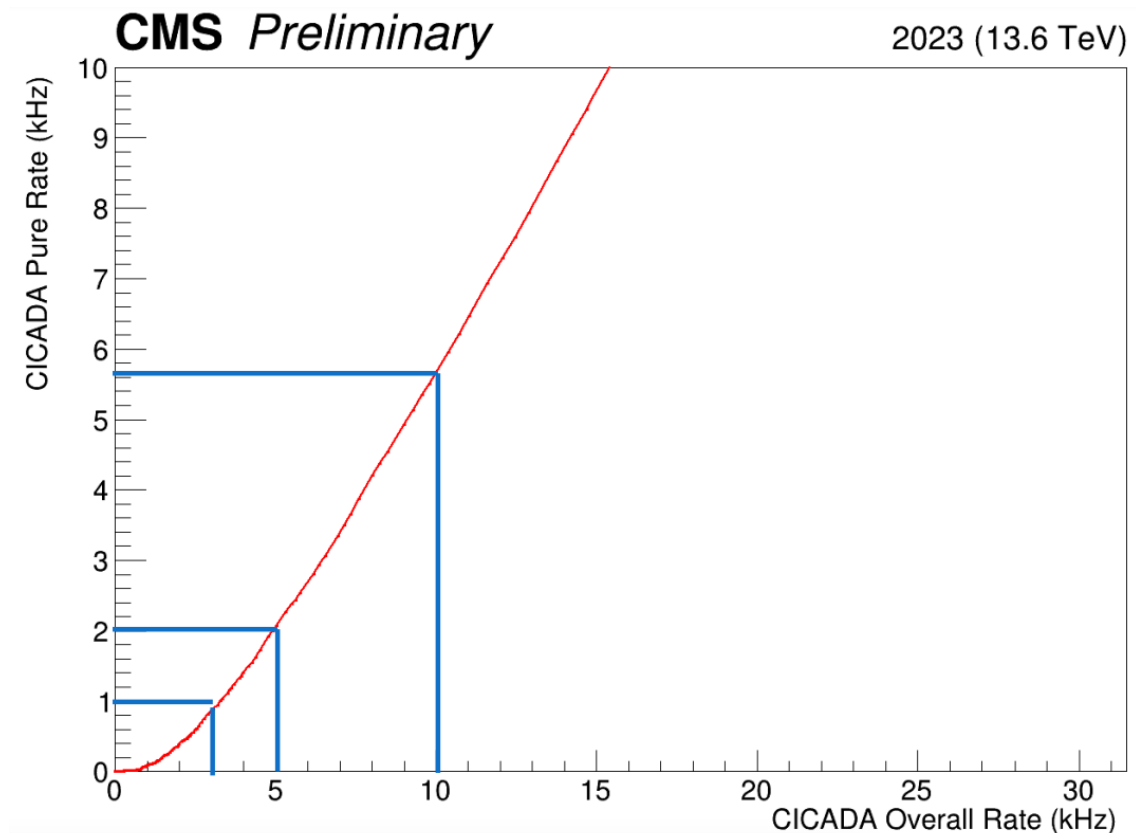


Figure 6.13: Illustration of rate overlap between CICADA and the existing trigger menu. The CICADA pure rate as a function of the CICADA overall rate. The pure rate refers to the rate attributed to CICADA only, without any other existing trigger paths being fired. The figure is taken from Ref. [21].

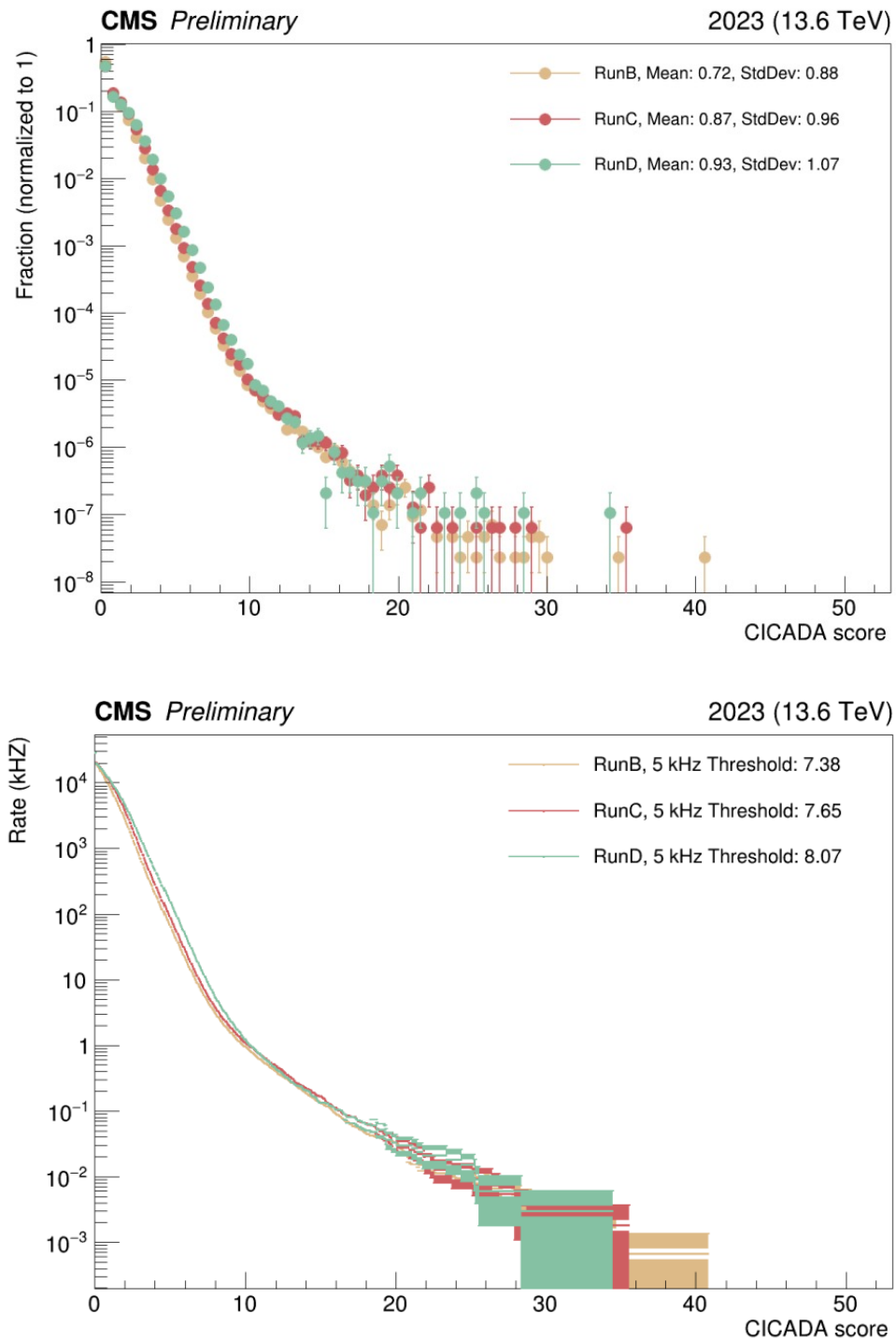


Figure 6.14: Illustration of the rate stability of CICADA across different 2023 runs. Top: CICADA score distributions. Bottom: CICADA rate as a function of the score threshold. The figures are taken from Ref. [21].

6.6 Summary and future prospects

In summary, an anomaly detection trigger algorithm called Calorimeter Image Convolutional Anomaly Detection Algorithm (CICADA) has been developed for Run 3. CICADA leverages machine learning to perform model-agnostic new physics searches using low-level calorimeter information at the Level-1 trigger in CMS. CICADA has demonstrated its ability to capture both conventional and unconventional experimental signatures from a wide range of rare and new physics signals. It prefers to select final states with high object multiplicity and high-energy objects and is also sensitive to objects with very low momentum, which would typically be rejected by traditional triggers based on hard threshold cuts on object momentum.

It is the first development of an anomaly detection algorithm in the trigger system, opening new avenues for physics research at the LHC. For instance, it potentially enables new physics searches in challenging and previously unexplored phase spaces, such as soft unclustered energy patterns and highly-compressed scenarios in supersymmetry models, which feature very soft objects in the final states. Additionally, CICADA can be combined with offline analyses that use anomaly detection techniques to perform truly model-agnostic searches throughout the entire selection process.

The main challenge in developing CICADA is to meet the stringent resource and latency constraints at the Level-1 trigger. Although model compression techniques are employed to ensure that the final model fits within an FPGA's budget and runs with a latency on the order of 100 ns, there is always a trade-off with the inference accuracy. These model compression techniques may not always achieve the desired performance while staying within tight constraints for every problem. Therefore, it is imperative to explore new techniques that allow for higher flexibility in developing future algorithms in resource-constrained environments like the LHC experiments. One promising technique is symbolic regression, which can effectively compress machine learning models and be efficiently implemented on FPGAs. This will be presented in the next chapter.

Chapter 7

Symbolic regression for hardware-efficient inference

In this chapter, a novel model compression method using symbolic regression (SR) in the context of FPGA is presented. This approach addresses the limitations in developing efficient ML algorithms in resource-constrained environments, such as the extremely tight resource and latency budgets at the Level-1 (L1) trigger encountered when developing the anomaly detection algorithms in the previous chapter. This chapter provides a lower-level improvement method, aiming to add more flexibility in future ML modeling to achieve effective and efficient inference on FPGAs for solving critical tasks.

A brief introduction to SR is given in Sec. 7.1. Then, a study of SR for fast machine learning inference on FPGAs is presented [25]. Sec. 7.2 presents strategies for efficient implementation of SR on FPGAs. Sec. 7.3 describes a benchmark LHC jet tagging dataset used for the study. Sec. 7.4 outlines the configurations for expression searching. Sec. 7.5 presents the results of model performance and FPGA resource utilization. Sec. 7.6 discusses future aspects.

7.1 Symbolic regression

SR is an ML technique that searches for models in the form of mathematical expressions to fit a dataset [137–139]. Unlike neural network (NN) models, which often involve a large number of parameters and function as black-box machines, symbolic models are typically more compact and allow for human interpretation since they are presented as analytic equations.

This naturally prompts an investigation into whether symbolic models can perform as well as common ML models, such as NNs, while consuming fewer computational resources. This is particularly relevant for the L1 trigger system in LHC experiments, which often require compromises on model performance to stay within resource budgets and meet the extreme sub-microsecond latency requirement. Therefore, the study of SR presented here focuses on the primary objective of model compression for fast inference on custom hardware such as FPGA.

In traditional regression techniques such as polynomial regression, the functional form of the model is fixed, for instance, by the highest degree in the polynomial. The coefficients are then adjusted by fitting the model to data. These traditional methods require human input to guess the exact functional form that could describe the shape of the data distributions. Their effectiveness is greatly limited and highly dependent on specific problems, especially for those with arbitrarily complicated shapes without underlying true functions. This requires humans to handcraft custom equation forms iteratively, which can be time-consuming and does not guarantee finding the desired solutions.

In contrast, SR is a regression method that leverages the power of ML to find the best-fit equations without prior human guesses of the final functional form. SR can be performed using genetic programming or NNs, as described in Sec. 7.1.1, where the functional forms themselves are ‘trainable parameters’, automatically determined from data in a supervised manner. In these approaches, only the building blocks of the expressions, instead of the full expressions, are needed. The building blocks include mathematical operators that can make up the expression candidates, such as unary operators (e.g., $\sin(x)$),

$\exp(-(x)^2)$) and binary operators (e.g., $+$, \times , $\sin(x) \cos(y)$).

Even with only a few mathematical operators, the equation search space in SR can grow exponentially, making it capable of fitting a wide range of different distributions. However, the downside is that SR is a complex combinatorial problem that is NP-hard [140]. As a result, searching for the best possible expressions typically involves high computational intensity. On the other hand, constraining the final expressions to be in a compact form is essential for human interpretability and efficient deployment in resource-constrained environments. This must be considered in SR algorithms to balance the trade-off between model performance and the practicality of the expressions.

7.1.1 Methods

There are two main computational approaches to SR: genetic programming and deep learning.

Genetic programming, first proposed in Ref. [137], formulates populations of programs and evolves them to solve particular computational problems in a manner inspired by biological evolution. In the context of SR, mathematical expressions are represented in a tree-like structure, where variables, constants, and operators are represented by tree nodes connected according to their algebraic relations. As shown in Fig. 7.1, new expressions can be created by mutation, where a tree node is randomly chosen to change its content, or by crossover, where subtrees from two expression trees are swapped.

PySR [141] is one of the SR libraries based on genetic programming, developed for discovering interpretable expressions for sciences. The algorithm creates and evolves expressions in multi-populations and filters candidates based on a tournament selection. It has been demonstrated that SR can potentially be used to extract underlying relations from data of physical systems, such as rediscovering Newton’s force laws from a system of interacting particles [142].

While the genetic programming approach is effective in finding accurate expressions to fit data, it is not computationally efficient due to its discrete nature of constructing and

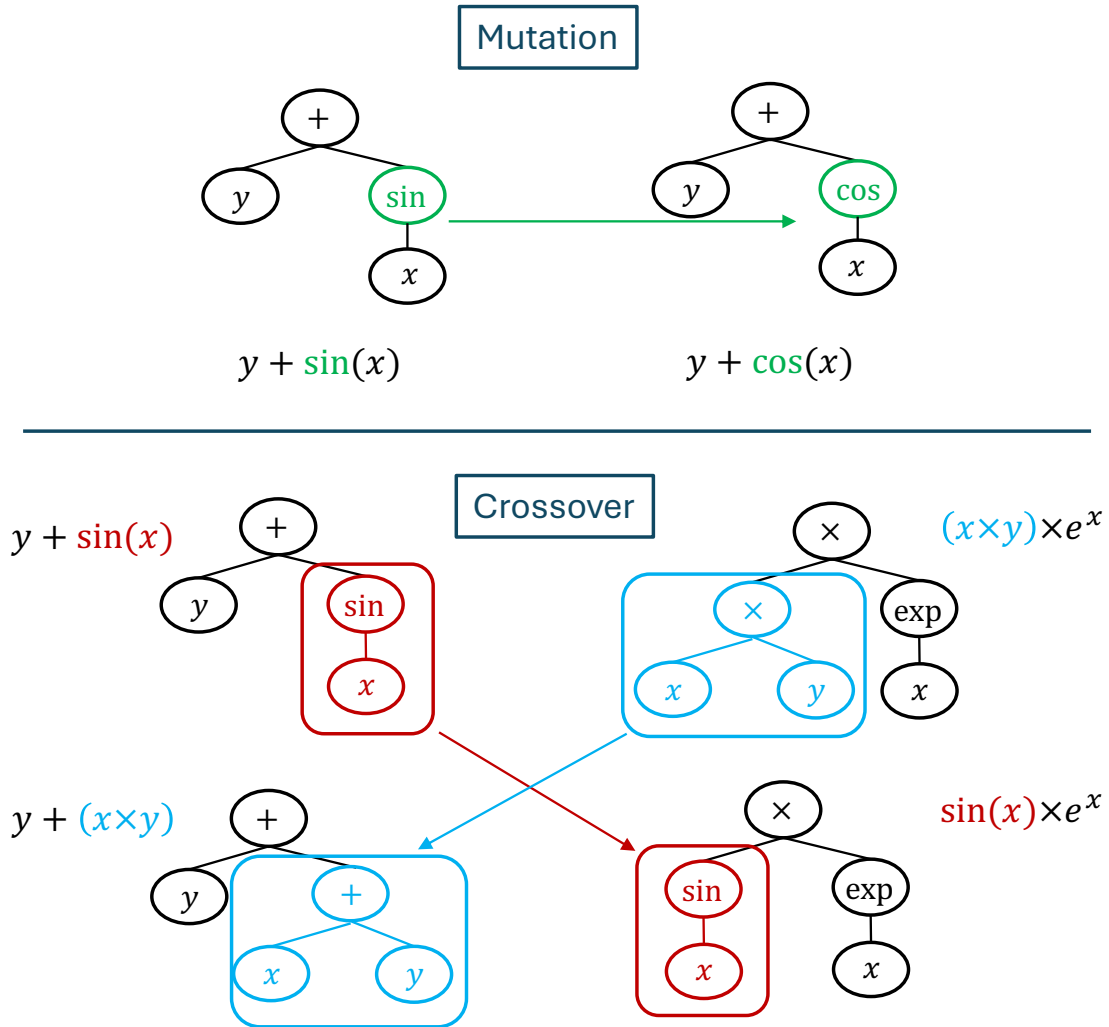


Figure 7.1: Illustration of genetic programming approach to SR, where expressions are represented as trees. Mutation: a tree node is randomly chosen, and its content is changed (top). Crossover: subtrees from two expression trees are swapped (bottom).

evolving the candidates. This inefficiency is exacerbated for large and high-dimensional datasets.

On the other hand, SR can be performed using deep learning, particularly by training NNs to find the best-fit equations. This approach allows for faster convergence due to gradient-based optimization and the possibility of scaling SR to larger and more complex datasets. Equation learner [143–145] is one of the earliest approaches using NNs to perform SR. It uses a fully-connected NN and generalizes the activation functions to custom

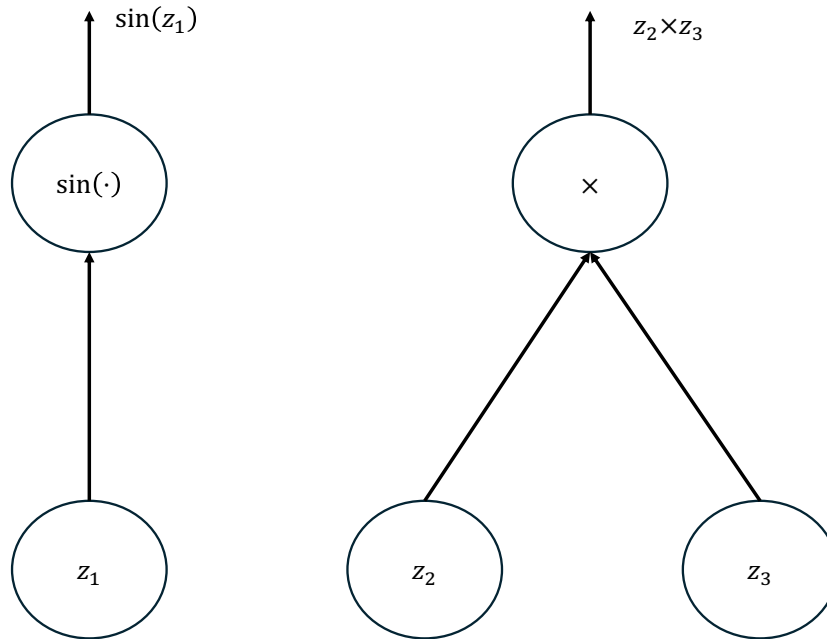


Figure 7.2: Illustration of NN approach to SR. Unary operator (left): activation functions can be generalized to more different mathematical operations, such as $\sin(x)$, which activate one neuron. Binary operator (right): activation functions can also be generalized to operations involving two inputs, such as $x \times y$, which activate two neurons simultaneously.

mathematical operators, as shown in Fig. 7.2. Unary operators, such as \sin , can activate one neuron, while binary operators, such as \times , can activate two neurons simultaneously, allowing for a greater variety of expression candidates.

Finding compact expressions requires NNs to have a highly sparse connection structure, which traditionally has relied on multistage pruning methods, such as in Ref. [146], to encourage sparsity. These methods involve a one-time pruning to remove all parameters with values below a certain threshold, often resulting in a significant compromise in accuracy. **SymbolNet** [147] extends the idea and improves it with a SR-dedicated pruning framework inspired by the dynamic sparse training [148]. This allows SR to scale to data with higher input dimensions and to find symbolic models that are as compact as possible while maintaining competitive performance.

7.2 Efficient implementation on FPGAs

After symbolic expressions are found, they can be converted into FPGA firmware with `hls4ml` [20, 25, 121]. Strategies are developed to adapt and optimize symbolic models for the FPGA environment.

One such functionality in `hls4ml` allows for approximating mathematical functions with lookup tables (LUTs), called LUT-based functions. This is similar to the quantization of NNs discussed in Sec. 6.4.2, where model weights are represented in fixed and short bit widths. The LUT-based approximation replaces mathematical functions, such as \sin , with a fixed-size table of pre-computed input-output pairs. The table range and size can be customized; for example, a LUT-based \sin function can be defined for $-8 < x < 8$ with 256 steps for granularity.

Fig. 7.3 compares \sin and \tan functions with their LUT-based approximations. These approximations avoid the runtime computation of mathematical functions, which can be very computationally intensive. For example, a \tan function can take over 40 clock cycles to compute on an FPGA, whereas a LUT-function operation is a simple memory lookup that takes only one clock cycle, regardless of the underlying mathematical function. Replacing mathematical functions with LUT-based counterparts significantly reduces FPGA resource usage and speeds up inference, while maintaining flexibility in the table definition to adapt to specific problems at hand.

If function accuracy cannot be compromised by LUT-based approximations, the original mathematical functions can be retained, and an alternative strategy to search for efficient symbolic models for FPGA implementation can be used. The method is called latency-aware training [25], which adds penalty weights to mathematical operators during model training according to their individual computation latency.

To quantify the model size for symbolic expressions, a metric called complexity [141] can be defined by counting the total number of tree nodes in the expression tree. This can be computed using the `preorder_traversal` function in `Sympy` [149]. For example, the expressions $y + \sin(x)$ and $(x \times y) \times e^x$ in Fig. 7.1 have complexities of 4 and 6, respectively.

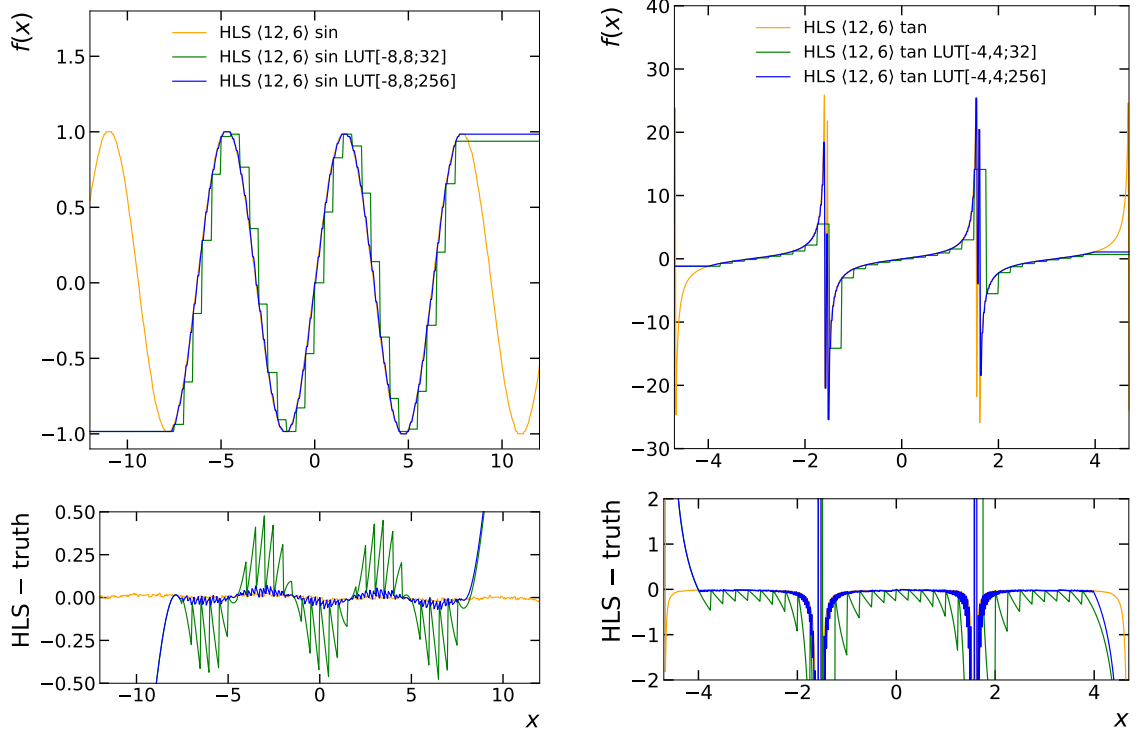


Figure 7.3: Comparison of the original mathematical function and the LUT-based approximation, which is discretized within a finite range by a pre-computed table of input-output pairs. Two functions are illustrated: sin (left) and tan (right). Two table definitions with different granularity are compared, with the precision of $\langle 12, 6 \rangle$. The table definition is denoted as $\text{LUT}(x_{\min}, x_{\max}, \text{table granularity})$. The figures are taken from Ref. [25].

There is a simple assumption that every node is equally weighted. However, this does not hold true if complexity is used as a proxy for approximating computational resource consumption, as computing an exponential function is more intensive than performing an arithmetic addition, for instance.

In latency-aware training, the latency of each operator, measured in clock cycles, is assigned as the complexity of the operator. The total complexity of an expression is the sum of complexities of all its building blocks. By including complexity as one of the constraint terms in training, operators that require more clock cycles to compute on an FPGA are penalized more, ensuring that the final expressions obtained are optimized in terms of resource utilization and latency simultaneously.

7.3 Dataset

To demonstrate the effectiveness of SR as a model compression technique, the LHC jet tagging dataset [150] is chosen for the study, designed to benchmark the performance of various fast ML algorithms on FPGAs. The dataset consists of simulated jets produced from high-energy proton-proton collisions at the LHC. Each jet can contain up to 150 particles and originates from either a light quark (q), gluon (g), W boson (W), Z boson (Z), or top quark (t).

The anti- k_T algorithm [58, 59] is used to cluster the jets with a cone radius of $R = 0.8$ (see Sec. 2.3.3). A minimal selection based on the jet p_T is applied to remove extreme outliers [20]. The task is to identify a given jet as one of the five classes from 16 high-level input features: $\sum z \log z$, $C_1^{\beta=0,1,2}$, $C_2^{\beta=1,2}$, $D_2^{\beta=1,2}$, $D_2^{(\alpha,\beta)=(1,1),(1,2)}$, $M_2^{\beta=1,2}$, $N_2^{\beta=1,2}$, m_{mMDT} , and Multiplicity. These input features are built from various jet reconstruction algorithms and are standard variables for studying jet substructure. More details on these features are described in Refs. [20, 151, 152]. For training, the inputs are standardized to have a zero mean and unit variance.

7.4 Experiments

First, a baseline model is needed to evaluate what SR can improve upon conventional algorithms. The baseline is a small-sized fully-connected NN consisting of three hidden layers with 64, 32, and 32 neurons, respectively, each followed by a ReLU activation [20]. The input layer has 16 neurons, and the output layer has 5 neurons followed by a softmax activation [93]:

$$\text{softmax}(\mathbf{x})_i = \frac{\exp(x_i)}{\sum_j \exp(x_j)}, \quad (7.1)$$

where the denominator sums over the 5 classes. The softmax outputs, $\text{softmax}(\mathbf{x})_i$, give the probabilities of the input jet originating from class i . The jet is identified according to the class that has the highest probability output. This baseline is chosen for its small model size, making it practical for FPGA deployment while maintaining reasonable jet

tagging accuracy. The baseline model is implemented in `QKeras` for quantization-aware training [24].

There are two experiments for the two strategies outlined in Sec. 7.2: one for studying the LUT-based functions and another for studying the latency-aware training [25]. Similar to the baseline, each symbolic model consists of five expressions, with the five outputs corresponding to the probabilities of being one of the five jet classes.

In the experiments, SR is performed using `PySR` [141], which is highly configurable to fit symbolic expressions under different constraints. In `PySR`, the model selection strategy is set to be based on the highest accuracy for a given maximum expression complexity. The configurable parts include the choice of operators allowed in expressions, the complexity assignment to each operator, and the various constraints on the operators, such as function nesting. These constraints will be set differently for the two experiments.

In general, the training attempts to solve the following optimization problem:

$$f_{\{q,g,W,Z,t\}}^* = \arg \min_{f_{\{q,g,W,Z,t\}} \in \mathcal{S}} \sum_i \sum_{\text{jet} \in \{q,g,W,Z,t\}} l(f_{\text{jet}}(\mathbf{x}^i), y_{\text{jet}}^i). \quad (7.2)$$

Here, $f_{\{q,g,W,Z,t\}}$ are the five expressions that take the 16 input features as variables \mathbf{x} . \mathcal{S} is the equation space containing all possible equations with complexity up to a maximum value $c \leq c_{\max}$ and satisfying the constraints defined in the training configuration. The first sum runs over the training dataset $\{\mathbf{x}^i, \mathbf{y}^i\}$, where \mathbf{y}^i are the labels. The second sum runs over the five jet classes. l is the individual loss function, taken as the L2 margin loss: $l(\hat{y}, y) = (1 - \hat{y}y)^2$, with labels set to +1 for the true class and -1 otherwise. The jet class corresponding to the expression that outputs the highest score is identified. To constrain the size of the symbolic models and ensure faster convergence, 6 out of the 16 features are selected using a random forest regressor in `scikit-learn` [153] for the training.

For the study of LUT-based approximations, four types of expressions are considered: polynomial, trigonometric, exponential, and logarithmic. These models are compared in terms of jet tagging accuracy and FPGA resource utilization, with and without the LUT-based approximations applied to the functions. To search for the polynomial expressions,

the choice of operators are limited to arithmetic: $+$ and \times . To search for other types of expressions, additional operators are allowed: $\sin(\cdot)$ for trigonometric, $\exp(-(\cdot)^2)$ for exponential, and $\log(|\cdot|)$ for logarithmic. A complexity of 1 is assigned indistinguishably to all operators, since the latency is equivalent in all lookup table operations. Constraints include excluding any kind of function nesting, such as $\sin(\sin(\cdot))$, and setting the maximum expression complexity c_{\max} to 20, 40, and 80, respectively, to obtain different sets of expressions. For each symbolic model, there are two sets of identical expressions: one using the original functions and another using the LUT-based functions.

For the study of latency-aware training, no LUT-based functions are employed, so different operators will be computed at runtime on an FPGA with varying latencies. To account for this, the latency for each operator, obtained from synthesis results, is measured in clock cycles and assigned as operator complexity in the search. The choice of operators include $+$ [1], $-$ [1], \times [1], $\sin(\cdot)$ [8], $\tan(\cdot)$ [48], $\exp(\cdot)$ [3], $\sinh(\cdot)$ [9], $\cosh(\cdot)$ [8], and $\log(|\cdot|)$ [4], where the numbers in square brackets refer to the corresponding number of clock cycles assigned as operator complexity, for the precision $\langle 16, 6 \rangle$, for example. The constraints include excluding any kind of function nesting and setting the maximum expression complexity c_{\max} to 20, 30, 40, 50, 60, 70, and 80, respectively, to obtain different sets of expressions.

All models are converted into FPGA implementation using `hls4ml` [20, 25, 121] and synthesized with Vivado HLS (2020.1) [136] on a Xilinx Virtex-9 FPGA. The FPGA clock frequency is set to 200 MHz, corresponding to a clock period of 5 ns. The SR models are compared with the baseline in terms of jet tagging accuracy, FPGA resource utilization (DSPs and LUTs), and latency.

7.5 Results

Tab. 7.1 shows the five expressions of the trigonometric model obtained at $c_{\max} = 20$, and Tab. 7.2 shows expressions for the top quark class in each of the four models. These expressions demonstrate the possibility of obtaining very compact expressions that pro-

Class	Trigonometric model ($c_{\max} = 20$)	AUC
g	$\sin(-2C_1^{\beta=1} + 0.31C_1^{\beta=2} + m_{\text{mMDT}} + \text{Multiplicity} - 0.09\text{Multiplicity}^2 - 0.79)$	0.897
q	$-0.33(\sin(m_{\text{mMDT}}) - 1.54)(\sin(-C_1^{\beta=1} + C_1^{\beta=2} + \text{Multiplicity}) - 0.81)\sin(m_{\text{mMDT}}) - 0.81$	0.853
t	$\sin(C_1^{\beta=1} + C_1^{\beta=2} - m_{\text{mMDT}} + 0.22(C_1^{\beta=2} - 0.29)(-C_1^{\beta=2} + C_2^{\beta=1} - \text{Multiplicity}) - 0.68)$	0.920
W	$-0.31(\text{Multiplicity} + (2.09 - \text{Multiplicity})\sin(8.02C_1^{\beta=2} + 0.98)) - 0.5$	0.877
Z	$(\sin(4.84m_{\text{mMDT}}) + 0.59)\sin(m_{\text{mMDT}} + 1.14)\sin(C_1^{\beta=2} + 4.84m_{\text{mMDT}}) - 0.94$	0.866

Table 7.1: Expressions of the trigonometric model ($c_{\max} = 20$), with ROC AUC shown. The table is taken from Ref. [25].

vide reasonable performance. Fig. 7.4 shows the ROC curves comparing the baseline NN model and the trigonometric model with and without LUT-based approximations. In this example, the expressions with LUT-based approximations do not compromise accuracy compared to the same expressions using the original functions. Their performance is very close the that of the baseline NN, which has a much larger model size.

Model	Expression for the top quark class ($c_{\max} = 40$)	AUC
Polynomial	$C_1^{\beta=2} + 0.09m_{\text{mMDT}}(2C_1^{\beta=1} + M_2^{\beta=2} - m_{\text{mMDT}} - \text{Multiplicity} - (1.82C_1^{\beta=1} - M_2^{\beta=2})(C_1^{\beta=2} - 0.49m_{\text{mMDT}}) - 3.22) - 0.53$	0.914
Trigonometric	$\sin(0.06(\sum z \log z)M_2^{\beta=2} - 0.25C_1^{\beta=2}(-C_1^{\beta=1} + 2C_1^{\beta=2} - M_2^{\beta=2} + \text{Multiplicity} - 8.86) - m_{\text{mMDT}} + 0.06\text{Multiplicity} - 0.4)$	0.925
Exponential	$0.23C_1^{\beta=1}(-m_{\text{mMDT}} + \text{Gauss}(0.63\text{Multiplicity}) + 1) - \text{Gauss}(C_1^{\beta=1}) + 0.45C_1^{\beta=2} - 0.23m_{\text{mMDT}} + 0.23\text{Gauss}((4.24 - 1.19C_2^{\beta=1})(C_1^{\beta=2} - m_{\text{mMDT}})) + 0.15$	0.920
Logarithmic	$C_1^{\beta=2} - 0.1m_{\text{mMDT}}(\text{Multiplicity} \times \log(\text{abs}(\text{Multiplicity})) + 2.2) - 0.02\log(\text{abs}(\text{Multiplicity})) - 0.1(C_1^{\beta=2}(C_1^{\beta=1} - 1.6M_2^{\beta=2} + m_{\text{mMDT}} + 1.28) - m_{\text{mMDT}} - 0.48)\log(\text{abs}(C_1^{\beta=2})) - 0.42$	0.923

Table 7.2: Expressions for the top quark class ($c_{\max} = 40$), with ROC AUC shown. The table is taken from Ref. [25].

Fig. 7.5 shows the accuracy of the symbolic models relative to the baseline. Three different model sizes are examined for the symbolic models: $c_{\max} = 20$, 40, and 80. All models are implemented for a range of fixed-point precision. The polynomial models are the simplest type as they do not have any mathematical function in the expressions, but their accuracy at $c_{\max} = 80$ can already achieve above 90% relative to the baseline, which is a fully-connected NN with a few thousands parameters. For other types of models, the relative accuracy can achieve over 90% even for very compact expressions with $c_{\max} = 20$. The LUT-based approximations lead to minimal drops in accuracy in some cases, but these are not significant.

Fig. 7.6 and Fig. 7.7 show the DSP and LUT usage of the models, respectively. The baseline NN consumes the most resources, while symbolic models consume orders of mag-

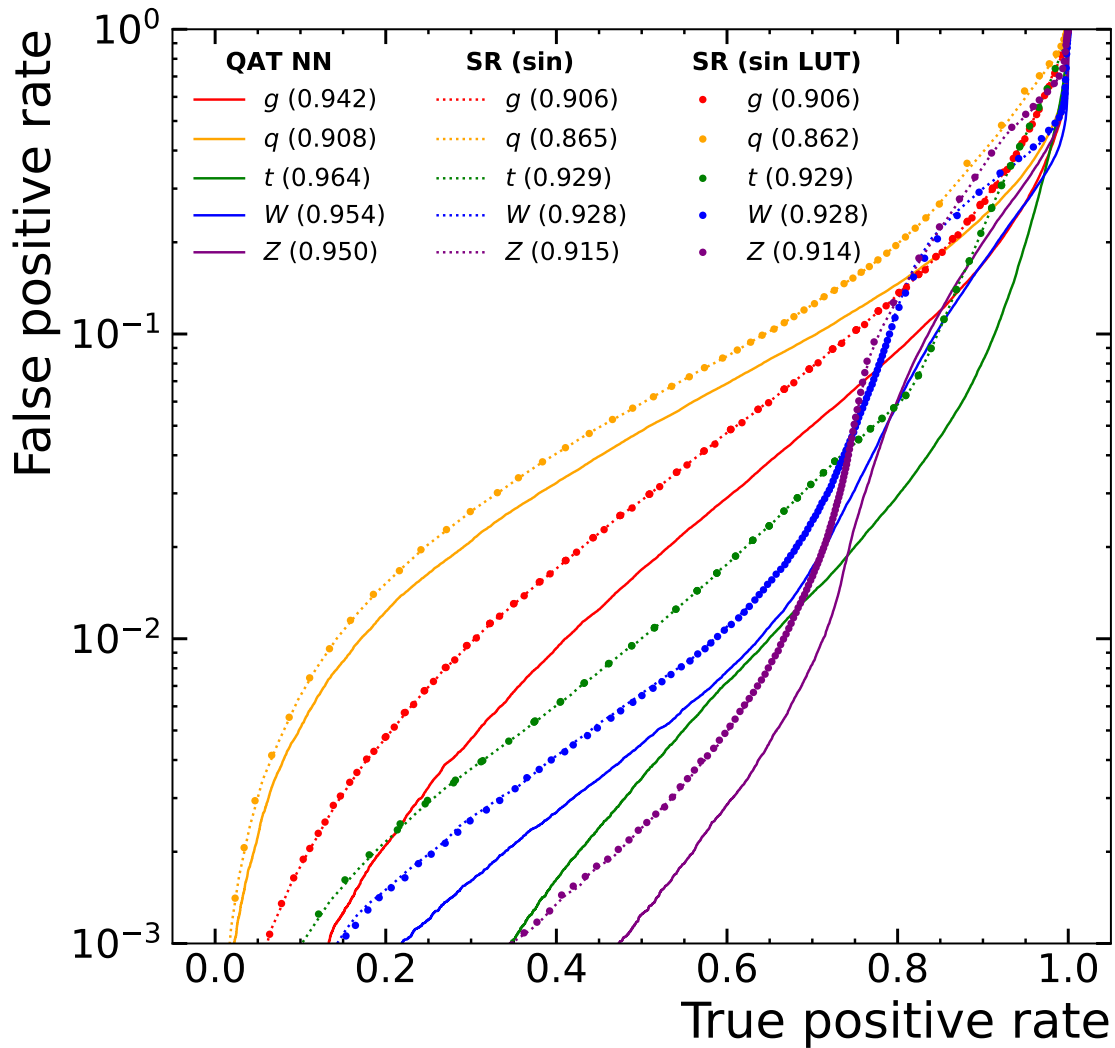


Figure 7.4: Comparison of ROC curves of the baseline quantization-aware trained (QAT) NN and the trigonometric model ($c_{\max} = 80$) with and without LUT-based approximations applied. The numbers in brackets are the corresponding ROC AUC values. The figure is taken from Ref. [25].

nitude less in general. This significant drop in resource utilization is expected since each symbolic model consists of only five lines of equations involving a very limited numbers of operations. In contrast, the baseline NN involves large matrix multiplications with thousands of model parameters. The same symbolic models with LUT-based approximations lead to significant improvements in resource consumption. This is expected since the original mathematical functions are computed at runtime, while the lookup table operations

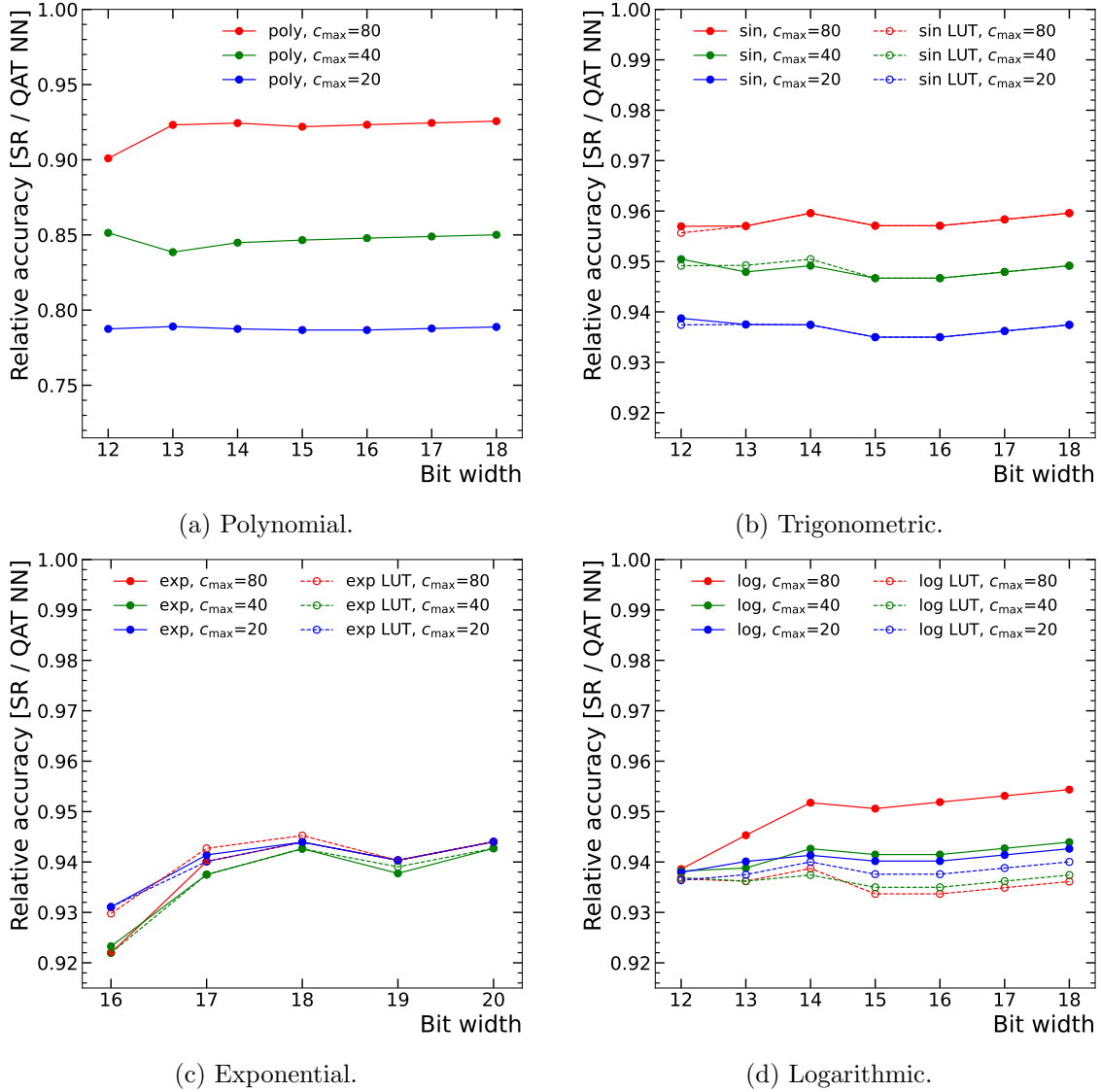


Figure 7.5: Accuracy of the symbolic models relative to the baseline NN. Three different model sizes are considered: $c_{\max} = 20, 40$, and 80 . Symbolic models with and without LUT-based approximations are compared. The figures are taken from Ref. [25].

involve only simple memory retrieval. There is an upward trend in resource consumption with the increase in bit width, as it takes more resources to represent numbers and perform the calculations.

Fig. 7.8 shows the latency of the models. The baseline NN requires a constant latency of 65 ns to compute in all cases. Most symbolic models require much less latency, especially those with LUT-based approximations. In particular, the logarithmic models at $c_{\max} = 20$

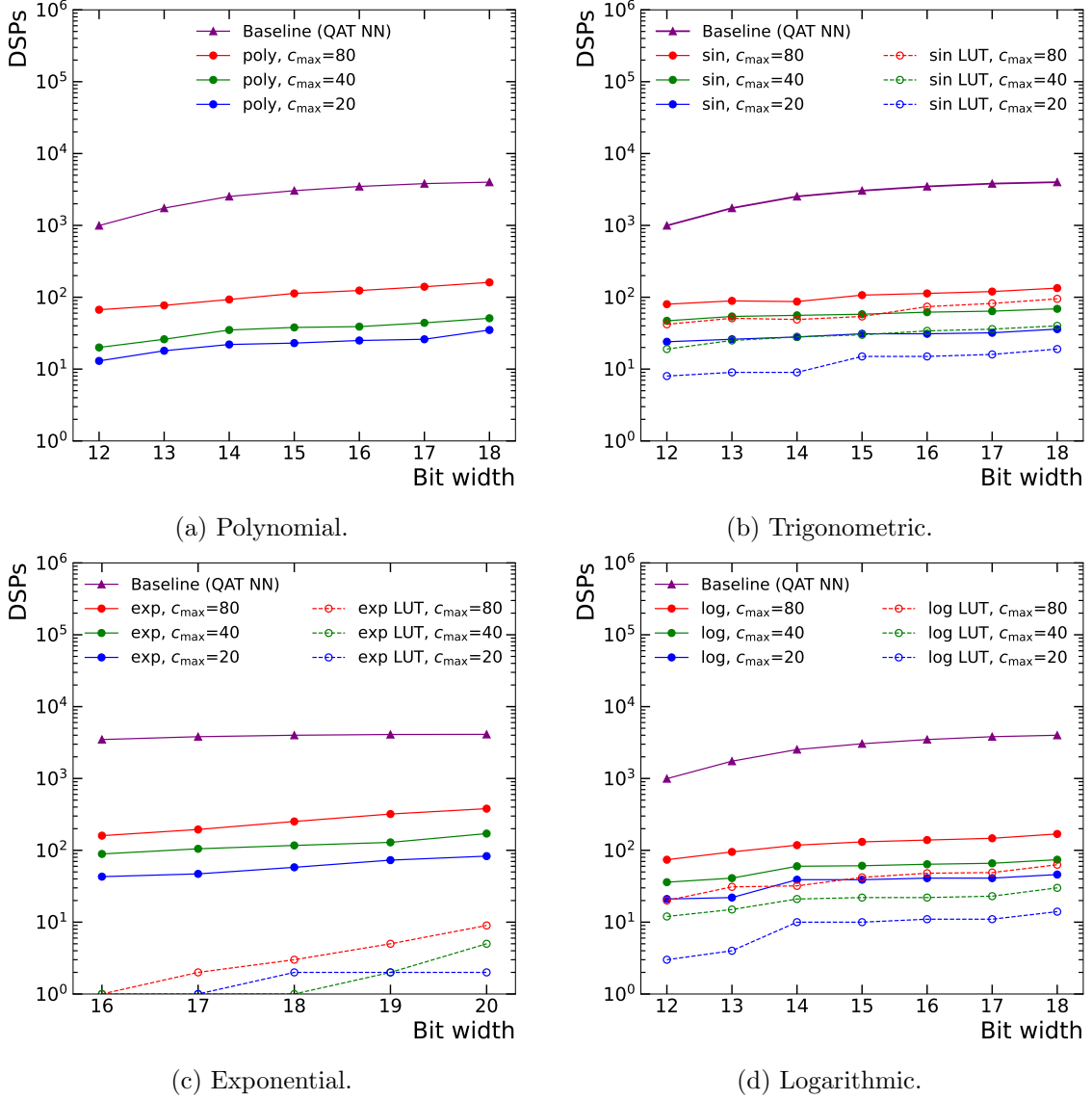


Figure 7.6: DSP utilization on an FPG. Three different model sizes are considered: $c_{\max} = 20, 40$, and 80 . Symbolic models with and without LUT-based approximations are compared with the baseline NN. The figures are taken from Ref. [25].

at the lowest two bit widths (12 and 13) require only one clock cycle to compute, which is 5 ns. These models achieve an accuracy of around 93.5% relative to the baseline NN while reducing latency by a factor of 13. This demonstrates the potential of SR to perform competitively while requiring significantly fewer resources and lower latency to run.

For the latency-aware training, where LUT-based approximations are disabled, the symbolic models obtained using the same configuration with and without setting the op-

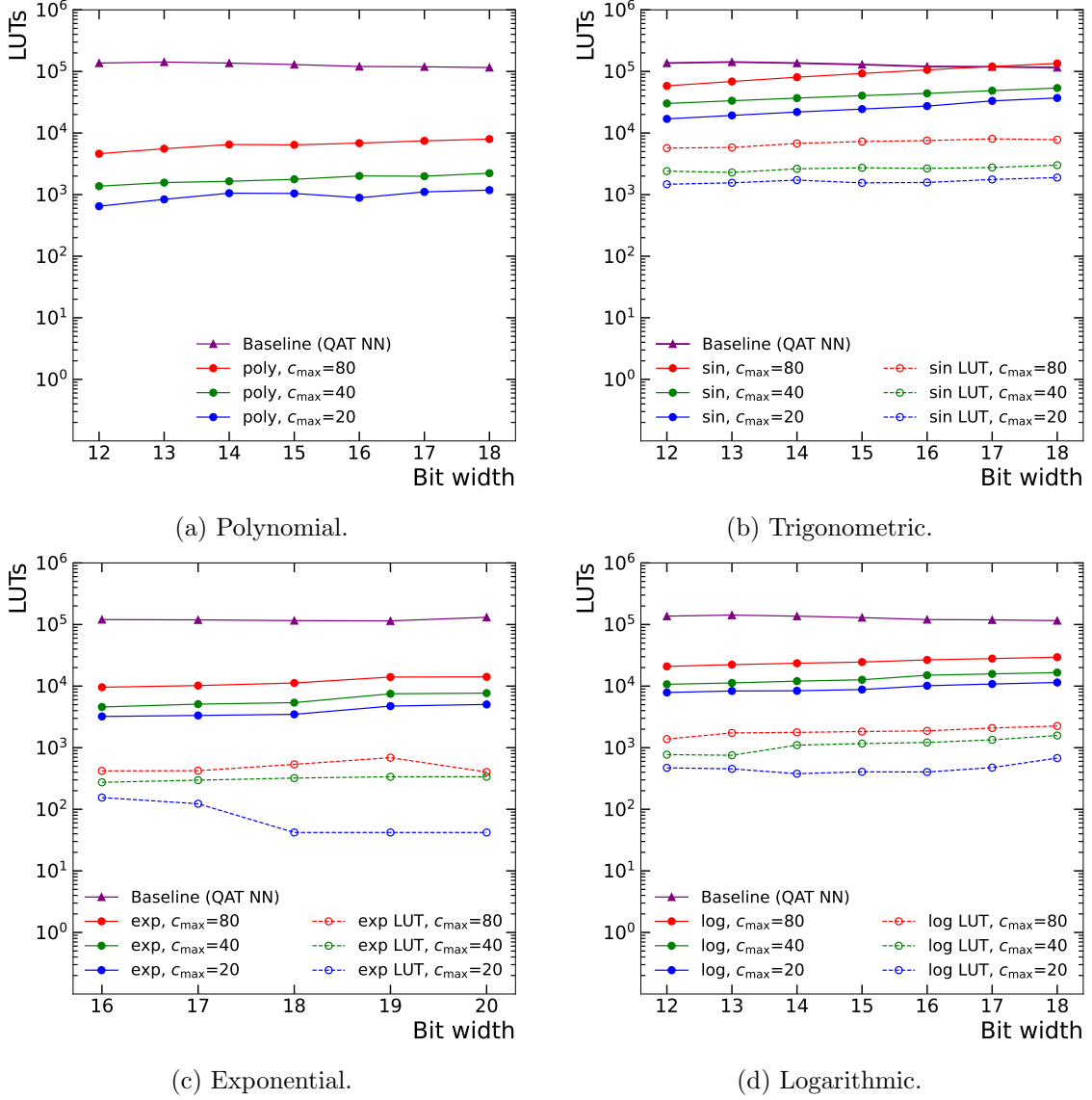


Figure 7.7: LUT utilization on an FPG. Three different model sizes are considered: $c_{\max} = 20, 40$, and 80 . Symbolic models with and without LUT-based approximations are compared with the baseline NN. The figures are taken from Ref. [25].

erator complexity to the corresponding latency are compared in Fig. 7.9. Models are trained repeatedly in the range of c_{\max} between 20 and 80, and two bit widths are considered: $\langle 18, 8 \rangle$, $\langle 16, 6 \rangle$. The difference in relative accuracy is mostly within 1% between models with and without latency-aware training. However, as expected, the latency is systematically lower for models obtained from latency-aware training. This demonstrates the effectiveness of integrating hardware constraints into the training process, leading to

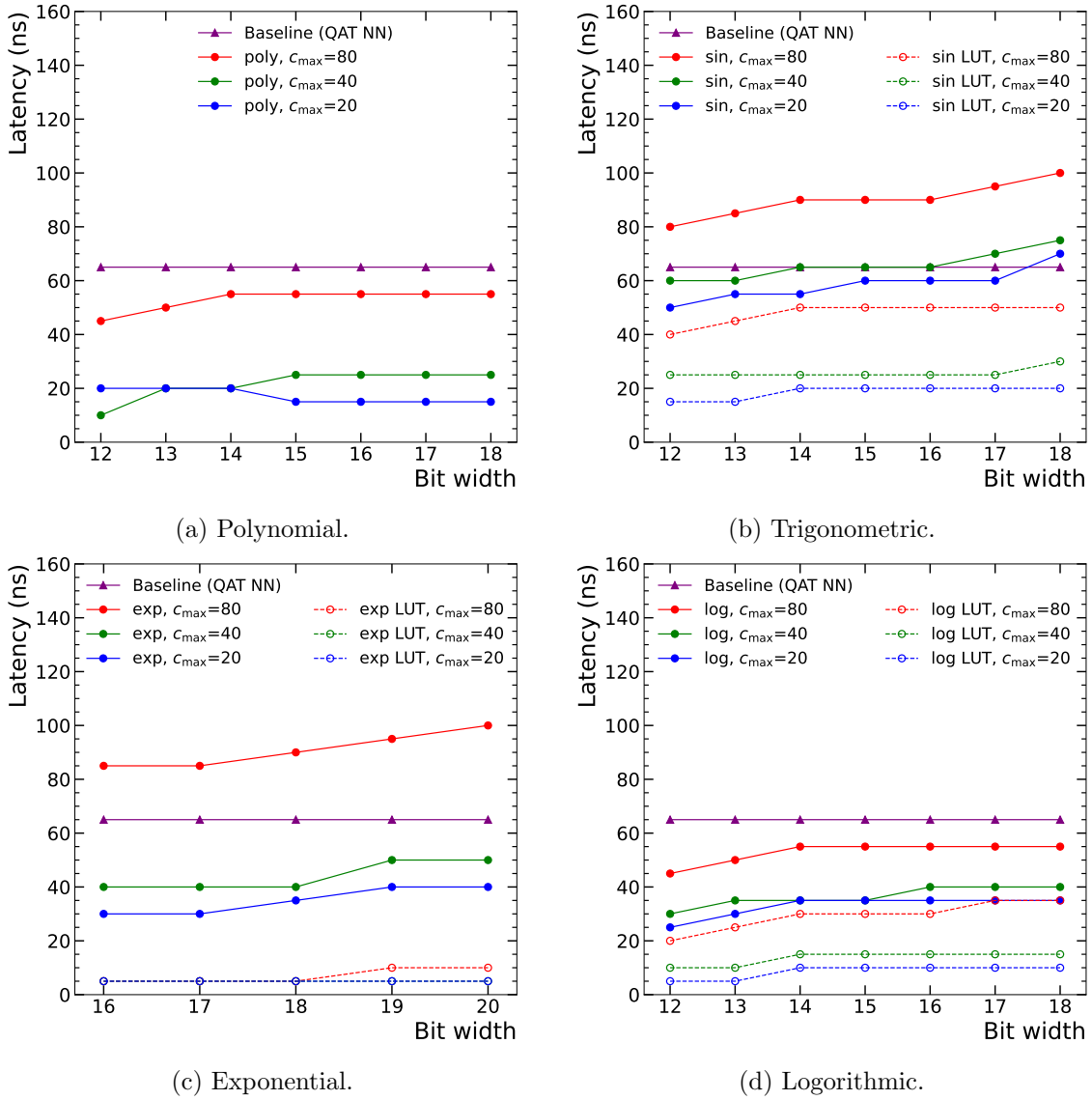


Figure 7.8: Inference latency on an FPGA. Three different model sizes are considered: $c_{\max}=20, 40$, and 80 . Symbolic models with and without LUT-based approximations are compared with the baseline NN. The figures are taken from Ref. [25].

more hardware-efficient models to be discovered.

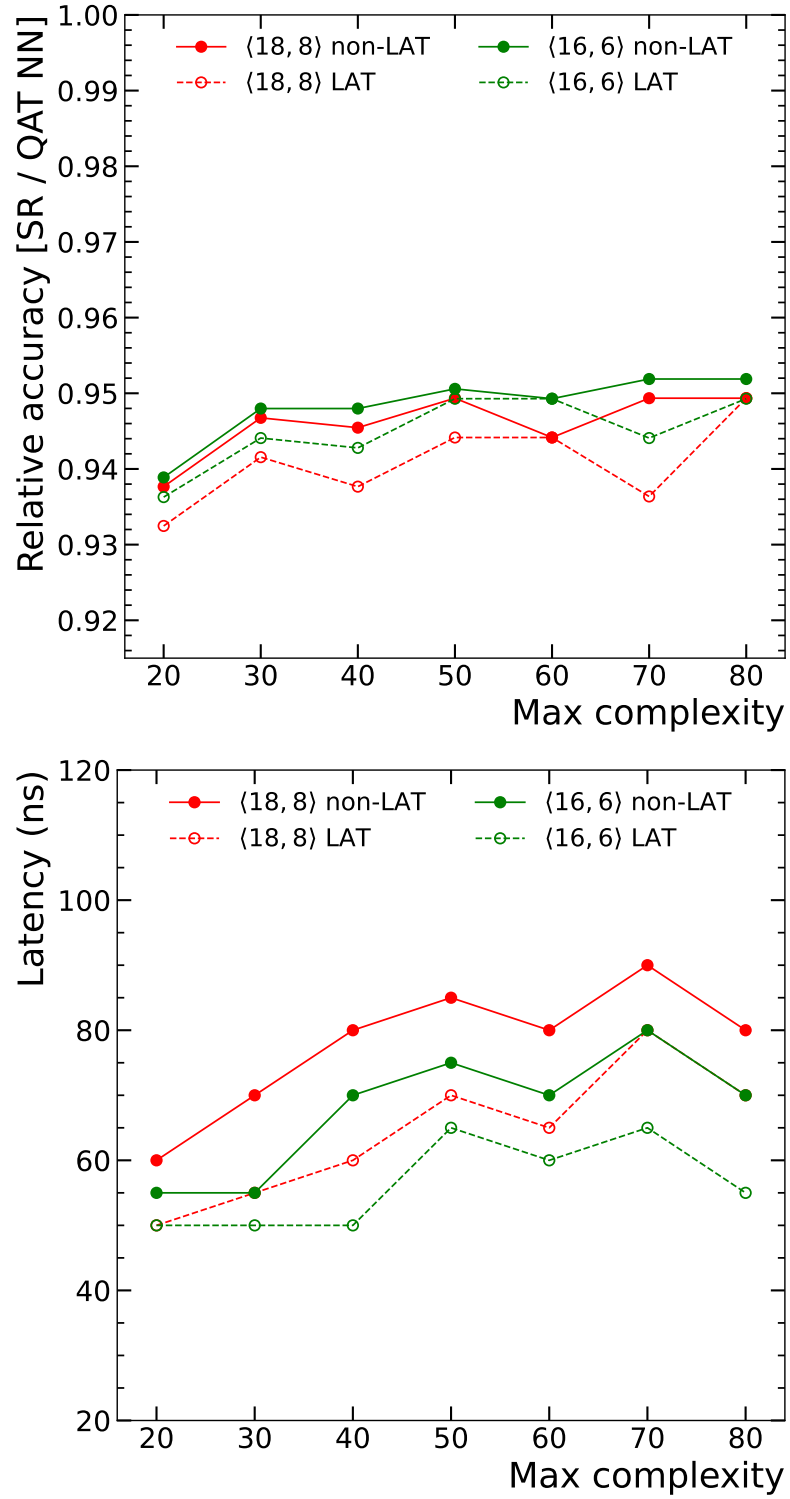


Figure 7.9: Symbolic models with and without latency-aware training (LAT) are compared in terms of accuracy relative to the baseline NN (top) and latency (bottom). Seven different model sizes are considered: $c_{\max} = 20, 30, 40, 50, 60, 70$, and 80 . Two bit widths are considered: $\langle 18, 8 \rangle$ and $\langle 16, 6 \rangle$. The figures are taken from Ref. [25].

7.6 Summary and future prospects

In summary, this study demonstrates that symbolic regression can be an effective model compression technique for developing lightweight models on FPGAs, achieving comparable performance to conventional models like neural networks while requiring significantly fewer FPGA resources and lower latency. In particular, symbolic models performing a benchmark LHC jet tagging task achieve inference latency as low as 5 nanoseconds and overall accuracy over 90% relative to the baseline neural network, which requires 65 nanoseconds to run. This provides an alternative modeling method with greater flexibility and improvements for developing machine learning algorithms to perform critical tasks in resource-constrained environments such as the Level-1 trigger in LHC experiments.

High-energy physics problems at the LHC typically involve many input features. For instance, the anomaly detection trigger presented in Ch. 6 has 252 input features, and some other problems involving detector readout signals can easily exceed such level. However, most symbolic regression methods, such as those based on genetic programming, cannot efficiently handle datasets with more than $\mathcal{O}(10)$ inputs, greatly limiting their potential applications. A recent study in Ref. [147] provides a solution by training neural networks in a pruning framework dedicated to symbolic regression, which can efficiently scale to datasets with up to $\mathcal{O}(1000)$ inputs. This opens up the possibility of deploying more symbolic models on FPGAs to perform critical physics tasks while running fast enough to meet the harsh constraints imposed by the extreme data rate of 40 MHz at the LHC.

Chapter 8

Summary of Part II

The second part of this thesis partially addresses the challenge faced in the analysis presented in the first part: low signal acceptance due to low-momentum objects in the final states being rejected by traditional trigger selection based on hard threshold cuts on object momenta. This problem is common across similar analyses at the LHC. A novel machine learning trigger algorithm called Calorimeter Image Convolutional Anomaly Detection Algorithm (CICADA) has been developed, which has the potential solve this issue.

CICADA uses anomaly detection techniques to perform model-agnostic new physics searches at the trigger level. It has demonstrated high signal sensitivity to both conventional and unconventional experimental signatures from a wide range of signals, including those with signatures similar to the analysis presented in this thesis. In particular, CICADA is sensitive to challenging signatures such as soft unclustered energy patterns, which would likely be rejected by traditional triggers. This demonstrates the potential of anomaly detection triggers to broadly enhance the physics program at the LHC.

Developing CICADA is challenging due to its deployment on FPGAs in the Level-1 trigger system, which has stringent requirements on resource budgets and latency limits below microseconds. Model compression techniques are in high demand to address these challenges for future developments. Another study investigates the use of symbolic regression as a new modeling method for fast machine learning inference on FPGAs. This

method has demonstrated competitive performance comparable to conventional models such as neural networks, while requiring significantly fewer FPGA resources and lower latency. This provides an alternative method with improvements and higher flexibility for future algorithms that need to meet stringent computational constraints.

Chapter 9

Conclusion and future prospects

The main challenge for searches such as the $h \rightarrow aa$ described in Part I is the ability to target signatures with low-momentum final-state objects in the harsh trigger selection that requires high-momentum thresholds on the objects. This issue persists if the trigger strategy remains unchanged, regardless of increased luminosity and improved offline analysis techniques. This problem brings up Part II, which introduces novel machine learning methods to address the issue at a lower level.

The studies in Part II provide new solutions to solve physics problems for the LHC, especially for the low-mass searches for new physics beyond the Standard Model, while addressing the challenging and unavoidable computational constraints. The anomaly detection triggers are currently deployed in CMS during Run 3, and it is expected that interesting anomaly datasets will soon be collected and analyzed to look for new physics. New opportunities are present in the ongoing Run 3, and they also pave the way for the upcoming HL-LHC era.

Appendix A

Published results and personal contributions

This thesis is primarily based on the following published results, with personal contributions indicated.

- CMS Collaboration, “**Search for exotic decays of the Higgs boson to a pair of pseudoscalars in the $\mu\mu b\bar{b}$ and $\tau\tau b\bar{b}$ final states**”, *Eur. Phys. J. C* **84**, 493 (2024), doi: <https://doi.org/10.1140/epjc/s10052-024-12727-4>, arXiv: [2402.13358](https://arxiv.org/abs/2402.13358). [15]
 - Was the lead analyst for the $\tau\tau b\bar{b}$ final state, and led the combination analysis of the two final states.
- CMS Collaboration, “**Level-1 Trigger Calorimeter Image Convolutional Anomaly Detection Algorithm**”, CMS Detector Performance Summaries (public document): [CMS-DP-2023-086](#) (2023) [21].
 - Developed the model of the Calorimeter Image Convolutional Anomaly Detection Algorithm (CICADA): model architecture design, training, and evaluation.
- CMS Collaboration, “**The Phase-2 Upgrade of the CMS Data Acquisition and High Level Trigger**”, CERN-LHCC-2021-007, CMS-TDR-022 [Technical Design Report](#) (2021). [18]
 - Performed the sensitivity projection study for the Phase-2 trigger in the analysis channel $h \rightarrow aa \rightarrow \tau\tau b\bar{b}$ [Sec.10.3.2.5, pp.240-243 in the TDR].

- H.F. Tsoi et al., “**Symbolic regression on FPGAs for fast machine learning inference**”, *EPJ Web Conf.* **295**, 09036 (2024), 26th International Conference on Computing in High Energy and Nuclear Physics (CHEP 2023),
doi: <https://doi.org/10.1051/epjconf/202429509036>, arXiv: [2305.04099](#). [25]
 - Validated the HLS implementation and performed the experiments.
- H.F. Tsoi et al., “**SymbolNet: Neural Symbolic Regression with Adaptive Dynamic Pruning**”, arXiv: [2401.09949](#). [147]
 - Proposed the model and performed the experiments.

Bibliography

- [1] de Florian, D. *et al.* Handbook of LHC Higgs Cross Sections: 4. Deciphering the Nature of the Higgs Sector **2/2017** (2016). CERN-2017-002-M, CERN-2017-002, [1610.07922](#).
- [2] Tumasyan, A. *et al.* A portrait of the Higgs boson by the CMS experiment ten years after the discovery. *Nature* **607**, 60–68 (2022). [Erratum: *Nature* 623, (2023)], [2207.00043](#).
- [3] Lopienska, E. The CERN accelerator complex, layout in 2022. Complexe des accélérateurs du CERN en janvier 2022 (2022). URL <https://cds.cern.ch/record/2800984>. General Photo.
- [4] CMS Collaboration. CMS Standard Model summary plots. <https://twiki.cern.ch/twiki/bin/view/CMSPublic/PhysicsResultsCombined>. Summary of the cross-section measurements of Standard Model processes (August 2023).
- [5] The HL-LHC project. <https://hilumilhc.web.cern.ch/content/hl-lhc-project>. LHC/HL-LHC Plan (last update February 2022).
- [6] Hayrapetyan, A. *et al.* Development of the CMS detector for the CERN LHC Run 3. *JINST* **19**, P05064 (2024). [2309.05466](#).
- [7] Tumasyan, A. *et al.* The Phase-2 Upgrade of the CMS Tracker (2017). URL <https://doi.org/10.17181/CERN.QZ28.FLHW>. CERN-LHCC-2017-009, CMS-TDR-014.
- [8] Bayatian, G. L. *et al.* CMS Physics: Technical Design Report Volume 1: Detector Performance and Software (2006). URL <https://cds.cern.ch/record/000922757>. CERN-LHCC-2006-001, CMS-TDR-8-1.
- [9] Sirunyan, A. M. *et al.* Calibration of the CMS hadron calorimeters using proton-proton collision data at $\sqrt{s} = 13$ TeV. *JINST* **15**, P05002 (2020). [1910.00079](#).
- [10] Sirunyan, A. M. *et al.* Performance of the CMS muon detector and muon reconstruction with proton-proton collisions at $\sqrt{s} = 13$ TeV. *JINST* **13**, P06015 (2018). [1804.04528](#).
- [11] Sirunyan, A. M. *et al.* Performance of the CMS Level-1 trigger in proton-proton collisions at $\sqrt{s} = 13$ TeV. *JINST* **15**, P10017 (2020). [2006.10165](#).

- [12] Sirunyan, A. M. *et al.* Particle-flow reconstruction and global event description with the CMS detector. *JINST* **12**, P10003 (2017). [1706.04965](#).
- [13] Sirunyan, A. M. *et al.* Identification of heavy-flavour jets with the CMS detector in pp collisions at 13 TeV. *JINST* **13**, P05011 (2018). [1712.07158](#).
- [14] Gleisberg, T. *et al.* Event generation with SHERPA 1.1. *JHEP* **02**, 007 (2009). [0811.4622](#).
- [15] Hayrapetyan, A. *et al.* Search for exotic decays of the Higgs boson to a pair of pseudoscalars in the $\mu\mu b\bar{b}$ and $\tau\tau b\bar{b}$ final states. *Eur. Phys. J. C* **84**, 493 (2024). [2402.13358](#).
- [16] Sirunyan, A. M. *et al.* Search for an exotic decay of the Higgs boson to a pair of light pseudoscalars in the final state with two b quarks and two τ leptons in proton-proton collisions at $\sqrt{s} = 13$ TeV. *Phys. Lett. B* **785**, 462 (2018). [1805.10191](#).
- [17] CMS Collaboration. Summary of 2HDM+S searches at 13 TeV (Run 2). <https://twiki.cern.ch/twiki/bin/view/CMSPublic/Summary2HDMSRun2>. (November 2023).
- [18] CMS Collaboration. The Phase-2 Upgrade of the CMS Data Acquisition and High Level Trigger. Tech. Rep., CERN, Geneva (2021). URL <https://cds.cern.ch/record/2759072>. CERN-LHCC-2021-007, CMS-TDR-022.
- [19] LeNail, A. NN-SVG: Publication-Ready Neural Network Architecture Schematics. *Journal of Open Source Software* **4**, 747 (2019). URL <https://doi.org/10.21105/joss.00747>.
- [20] Duarte, J. *et al.* Fast inference of deep neural networks in FPGAs for particle physics. *JINST* **13**, P07027 (2018). [1804.06913](#).
- [21] CMS Collaboration. Level-1 Trigger Calorimeter Image Convolutional Anomaly Detection Algorithm (2023). URL <https://cds.cern.ch/record/2879816>. CERN-CMS-DP-2023-086.
- [22] Iqbal, H. HarisIqbal88/PlotNeuralNet v1.0.0 (2018). URL <https://doi.org/10.5281/zenodo.2526396>.
- [23] Chollet, F. *et al.* Keras. <https://keras.io> (2015).
- [24] Coelho, C. N. *et al.* Automatic heterogeneous quantization of deep neural networks for low-latency inference on the edge for particle detectors. *Nature Mach. Intell.* **3**, 675–686 (2021). [2006.10159](#).
- [25] Tsoi, H. F. *et al.* Symbolic Regression on FPGAs for Fast Machine Learning Inference. *EPJ Web Conf.* **295**, 09036 (2024). [2305.04099](#).
- [26] Workman, R. L. *et al.* Review of Particle Physics. *PTEP* **2022**, 083C01 (2022). URL <https://doi.org/10.1093/ptep/ptac097>.

- [27] Glashow, S. L. & Weinberg, S. Natural Conservation Laws for Neutral Currents. *Phys. Rev. D* **15**, 1958 (1977).
- [28] Branco, G. C. *et al.* Theory and phenomenology of two-Higgs-doublet models. *Phys. Rept.* **516**, 1–102 (2012). [1106.0034](#).
- [29] Aad, G. *et al.* Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC. *Phys. Lett. B* **716**, 1–29 (2012). [1207.7214](#).
- [30] Chatrchyan, S. *et al.* Observation of a New Boson at a Mass of 125 GeV with the CMS Experiment at the LHC. *Phys. Lett. B* **716**, 30–61 (2012). [1207.7235](#).
- [31] Chatrchyan, S. *et al.* Observation of a New Boson with Mass Near 125 GeV in pp Collisions at $\sqrt{s} = 7$ and 8 TeV. *JHEP* **06**, 081 (2013). [1303.4571](#).
- [32] Glashow, S. L. Partial Symmetries of Weak Interactions. *Nucl. Phys.* **22**, 579–588 (1961).
- [33] Weinberg, S. A Model of Leptons. *Phys. Rev. Lett.* **19**, 1264–1266 (1967).
- [34] Salam, A. Weak and Electromagnetic Interactions. *Conf. Proc. C* **680519**, 367–377 (1968).
- [35] 't Hooft, G. & Veltman, M. J. G. Regularization and Renormalization of Gauge Fields. *Nucl. Phys. B* **44**, 189–213 (1972).
- [36] Englert, F. & Brout, R. Broken symmetry and the mass of gauge vector mesons. *Phys. Rev. Lett.* **13**, 321–323 (1964). URL <https://link.aps.org/doi/10.1103/PhysRevLett.13.321>.
- [37] Higgs, P. W. Broken symmetries, massless particles and gauge fields. *Phys. Lett.* **12**, 132–133 (1964). URL [https://doi.org/10.1016/0031-9163\(64\)91136-9](https://doi.org/10.1016/0031-9163(64)91136-9).
- [38] Higgs, P. W. Broken symmetries and the masses of gauge bosons. *Phys. Rev. Lett.* **13**, 508–509 (1964). URL <https://link.aps.org/doi/10.1103/PhysRevLett.13.508>.
- [39] Guralnik, G. S., Hagen, C. R. & Kibble, T. W. B. Global conservation laws and massless particles. *Phys. Rev. Lett.* **13**, 585–587 (1964). URL <https://link.aps.org/doi/10.1103/PhysRevLett.13.585>.
- [40] Higgs, P. W. Spontaneous symmetry breakdown without massless bosons. *Phys. Rev.* **145**, 1156–1163 (1966). URL <https://link.aps.org/doi/10.1103/PhysRev.145.1156>.
- [41] Kibble, T. W. B. Symmetry breaking in non-abelian gauge theories. *Phys. Rev.* **155**, 1554–1561 (1967). URL <https://link.aps.org/doi/10.1103/PhysRev.155.1554>.
- [42] Aad, G. *et al.* A detailed map of Higgs boson interactions by the ATLAS experiment ten years after the discovery. *Nature* **607**, 52–59 (2022). [Erratum: *Nature* 612, E24 (2022)], [2207.00092](#).

- [43] Noether, E. Invariante variationsprobleme. *Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen, Mathematisch-Physikalische Klasse* **1918**, 235–257 (1918). URL <http://eudml.org/doc/59024>.
- [44] Yang, C. N. & Mills, R. L. Conservation of isotopic spin and isotopic gauge invariance. *Phys. Rev.* **96**, 191–195 (1954). URL <https://link.aps.org/doi/10.1103/PhysRev.96.191>.
- [45] Aad, G. *et al.* The ATLAS Experiment at the CERN Large Hadron Collider. *JINST* **3**, S08003 (2008).
- [46] Chatrchyan, S. *et al.* The CMS Experiment at the CERN LHC. *JINST* **3**, S08004 (2008).
- [47] Zee, A. *Quantum Field Theory in a Nutshell* (Princeton University Press, 2010).
- [48] Bertone, G., Hooper, D. & Silk, J. Particle dark matter: Evidence, candidates and constraints. *Phys. Rept.* **405**, 279–390 (2005). [hep-ph/0404175](#).
- [49] Martin, S. P. A Supersymmetry primer. *Adv. Ser. Direct. High Energy Phys.* **18**, 1–98 (1998). [hep-ph/9709356](#).
- [50] Curtin, D. *et al.* Exotic decays of the 125 GeV Higgs boson. *Phys. Rev. D* **90**, 075004 (2014). [1312.4992](#).
- [51] Cepeda, M., Gori, S., Outschoorn, V. M. & Shelton, J. Exotic Higgs Decays. *Annual Review of Nuclear and Particle Science* **72**, 119–149 (2021). URL <https://doi.org/10.1146/annurev-nucl-102319-024147>. [2111.12751](#).
- [52] Englert, C., Plehn, T., Zerwas, D. & Zerwas, P. M. Exploring the Higgs portal. *Phys. Lett. B* **703**, 298–305 (2011). [1106.3097](#).
- [53] Arcadi, G., Djouadi, A. & Raidal, M. Dark Matter through the Higgs portal. *Phys. Rept.* **842**, 1–180 (2020). [1903.03616](#).
- [54] Trodden, M. Electroweak baryogenesis: A Brief review. In *33rd Rencontres de Moriond: Electroweak Interactions and Unified Theories*, 471–480 (1998). [hep-ph/9805252](#).
- [55] Evans, L. & Bryant, P. LHC Machine. *JINST* **3**, S08001 (2008).
- [56] Adzic, P. *et al.* Energy resolution of the barrel of the CMS electromagnetic calorimeter. *JINST* **2**, P04004 (2007).
- [57] Sirunyan, A. M. *et al.* Electron and photon reconstruction and identification with the CMS experiment at the CERN LHC. *JINST* **16**, P05014 (2021). [2012.06888](#).
- [58] Cacciari, M., Salam, G. P. & Soyez, G. The anti- k_t jet clustering algorithm. *JHEP* **04**, 063 (2008). [0802.1189](#).
- [59] Cacciari, M., Salam, G. P. & Soyez, G. FastJet User Manual. *Eur. Phys. J. C* **72**, 1896 (2012). [1111.6097](#).

- [60] Sirunyan, A. M. *et al.* Pileup mitigation at CMS in 13 TeV data. *JINST* **15**, P09018 (2020). [2003.00503](#).
- [61] Khachatryan, V. *et al.* Jet energy scale and resolution in the CMS experiment in pp collisions at 8 TeV. *JINST* **12**, P02014 (2017). [1607.03663](#).
- [62] Bols, E., Kieseler, J., Verzetti, M., Stoye, M. & Stakia, A. Jet Flavour Classification Using DeepJet. *JINST* **15**, P12012 (2020). [2008.10519](#).
- [63] CMS Collaboration. Performance of the DeepJet b tagging algorithm using 41.9/fb of data from proton-proton collisions at 13TeV with Phase 1 CMS detector (2018). URL <https://cds.cern.ch/record/2646773>. CERN-CMS-DP-2018-058.
- [64] Sirunyan, A. M. *et al.* Performance of reconstruction and identification of τ leptons decaying to hadrons and ν_τ in pp collisions at $\sqrt{s} = 13$ TeV. *JINST* **13**, P10005 (2018). [1809.02816](#).
- [65] Tumasyan, A. *et al.* Identification of hadronic tau lepton decays using a deep neural network. *JINST* **17**, P07023 (2022). [2201.08458](#).
- [66] Alwall, J. *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**, 079 (2014). [1405.0301](#).
- [67] Nason, P. A New method for combining NLO QCD with shower Monte Carlo algorithms. *JHEP* **11**, 040 (2004). [hep-ph/0409146](#).
- [68] Frixione, S., Nason, P. & Oleari, C. Matching NLO QCD computations with Parton Shower simulations: the POWHEG method. *JHEP* **11**, 070 (2007). [0709.2092](#).
- [69] Alioli, S., Nason, P., Oleari, C. & Re, E. A general framework for implementing NLO calculations in shower Monte Carlo programs: the POWHEG BOX. *JHEP* **06**, 043 (2010). [1002.2581](#).
- [70] Alioli, S., Hamilton, K., Nason, P., Oleari, C. & Re, E. Jet pair production in POWHEG. *JHEP* **04**, 081 (2011). [1012.3380](#).
- [71] Alioli, S., Nason, P., Oleari, C. & Re, E. NLO Higgs boson production via gluon fusion matched with shower in POWHEG. *JHEP* **04**, 002 (2009). [0812.0578](#).
- [72] Bagnaschi, E., Degrandi, G., Slavich, P. & Vicini, A. Higgs production via gluon fusion in the POWHEG approach in the SM and in the MSSM. *JHEP* **02**, 088 (2012). [1111.2854](#).
- [73] Nason, P. & Oleari, C. NLO Higgs boson production via vector-boson fusion matched with shower in POWHEG. *JHEP* **02**, 037 (2010). [0911.5299](#).
- [74] Luisoni, G., Nason, P., Oleari, C. & Tramontano, F. $HW^\pm/HZ + 0$ and 1 jet at NLO with the POWHEG BOX interfaced to GoSam and their merging within MiNLO. *JHEP* **10**, 083 (2013). [1306.2542](#).

- [75] Hartanto, H. B., Jager, B., Reina, L. & Wackerroth, D. Higgs boson production in association with top quarks in the POWHEG BOX. *Phys. Rev. D* **91**, 094003 (2015). [1501.04498](#).
- [76] Ball, R. D. *et al.* Parton distributions from high-precision collider data. *Eur. Phys. J. C* **77**, 663 (2017). [1706.00428](#).
- [77] Sjöstrand, T. *et al.* An introduction to PYTHIA 8.2. *Comput. Phys. Commun.* **191**, 159–177 (2015). [1410.3012](#).
- [78] Skands, P., Carrazza, S. & Rojo, J. Tuning PYTHIA 8.1: the Monash 2013 Tune. *Eur. Phys. J. C* **74**, 3024 (2014). [1404.5630](#).
- [79] Sirunyan, A. M. *et al.* Extraction and validation of a new set of CMS PYTHIA8 tunes from underlying-event measurements. *Eur. Phys. J. C* **80**, 4 (2020). [1903.12179](#).
- [80] Alwall, J. *et al.* Comparative study of various algorithms for the merging of parton showers and matrix elements in hadronic collisions. *Eur. Phys. J. C* **53**, 473–500 (2008). [0706.2569](#).
- [81] Alwall, J., de Visscher, S. & Maltoni, F. QCD radiation in the production of heavy colored particles at the LHC. *JHEP* **02**, 017 (2009). [0810.5350](#).
- [82] Frederix, R. & Frixione, S. Merging meets matching in MC@NLO. *JHEP* **12**, 061 (2012). [1209.6215](#).
- [83] Agostinelli, S. *et al.* GEANT4—a simulation toolkit. *Nucl. Instrum. Meth. A* **506**, 250–303 (2003).
- [84] Allison, J. *et al.* Geant4 developments and applications. *IEEE Trans. Nucl. Sci.* **53**, 270 (2006).
- [85] Sirunyan, A. M. *et al.* An embedding technique to determine $\tau\tau$ backgrounds in proton-proton collision data. *JINST* **14**, P06032 (2019). [1903.01216](#).
- [86] Bianchini, L. *et al.* Reconstruction of the Higgs mass in events with Higgs bosons decaying into a pair of τ leptons using matrix element techniques. *Nucl. Instrum. Meth. A* **862**, 54–84 (2017). [1603.05910](#).
- [87] Sirunyan, A. M. *et al.* Performance of missing transverse momentum reconstruction in proton-proton collisions at $\sqrt{s} = 13$ TeV using the CMS detector. *JINST* **14**, P07004 (2019). [1903.06078](#).
- [88] Khachatryan, V. *et al.* Search for neutral MSSM Higgs bosons decaying to a pair of tau leptons in pp collisions. *JHEP* **10**, 160 (2014). [1408.3316](#).
- [89] Rosenblatt, F. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review* **65** **6**, 386–408 (1958).
- [90] Cybenko, G. V. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems* **2**, 303–314 (1989).

- [91] Hornik, K., Stinchcombe, M. B. & White, H. L. Multilayer feedforward networks are universal approximators. *Neural Networks* **2**, 359–366 (1989).
- [92] Haykin, S. *Neural networks: a comprehensive foundation* (Prentice Hall PTR, 1994).
- [93] Goodfellow, I., Bengio, Y. & Courville, A. *Deep Learning* (MIT Press, 2016). <http://www.deeplearningbook.org>.
- [94] Nair, V. & Hinton, G. E. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning, ICML'10*, 807–814 (Omnipress, Madison, WI, USA, 2010).
- [95] Kingma, D. & Ba, J. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)* (San Diego, CA, USA, 2015). **1412.6980**.
- [96] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* **15**, 1929–1958 (2014). URL <https://jmlr.org/papers/v15/srivastava14a.html>.
- [97] Li, L., Jamieson, K., DeSalvo, G., Rostamizadeh, A. & Talwalkar, A. Hyperband: A novel bandit-based approach to hyperparameter optimization. *Journal of Machine Learning Research* **18**, 1–52 (2018). URL <https://jmlr.org/papers/v18/16-558.html>.
- [98] O'Malley, T. *et al.* KerasTuner. <https://github.com/keras-team/keras-tuner> (2019).
- [99] Sirunyan, A. M. *et al.* Precision luminosity measurement in proton-proton collisions at $\sqrt{s} = 13$ TeV in 2015 and 2016 at CMS. *Eur. Phys. J. C* **81**, 800 (2021). **2104.01927**.
- [100] CMS Collaboration. CMS luminosity measurement for the 2017 data-taking period at $\sqrt{s} = 13$ TeV (2018). URL <https://cds.cern.ch/record/2621960>. CMS-PAS-LUM-17-004.
- [101] CMS Collaboration. CMS luminosity measurement for the 2018 data-taking period at $\sqrt{s} = 13$ TeV (2019). URL <https://cds.cern.ch/record/2676164>. CMS-PAS-LUM-18-002.
- [102] Tumasyan, A. *et al.* Measurement of the inclusive and differential Higgs boson production cross sections in the decay mode to a pair of τ leptons in pp collisions at $\sqrt{s} = 13$ TeV. *Phys. Rev. Lett.* **128**, 081805 (2022). **2107.11486**.
- [103] Khachatryan, V. *et al.* Performance of Electron Reconstruction and Selection with the CMS Detector in Proton-Proton Collisions at $\sqrt{s} = 8$ TeV. *JINST* **10**, P06005 (2015). **1502.02701**.

- [104] Butterworth, J. *et al.* PDF4LHC recommendations for LHC Run II. *J. Phys. G* **43**, 023001 (2016). [1510.03865](#).
- [105] ATLAS, CMS, LHC Higgs Combination Group. Procedure for the LHC Higgs boson search combination in Summer 2011 (2011). URL <https://cds.cern.ch/record/1379837>. CMS-NOTE-2011-005, ATL-PHYS-PUB-2011-011.
- [106] Barlow, R. J. & Beeston, C. Fitting using finite Monte Carlo samples. *Comput. Phys. Commun.* **77**, 219–228 (1993).
- [107] Hayrapetyan, A. *et al.* The CMS Statistical Analysis and Combination Tool: COMBINE (2024). CMS-CAT-23-001, CERN-EP-2024-078, [2404.06614](#).
- [108] Junk, T. Confidence level computation for combining searches with small statistics. *Nucl. Instrum. Meth. A* **434**, 435–443 (1999). [hep-ex/9902006](#).
- [109] Read, A. L. Presentation of search results: The CL_s technique. *J. Phys. G* **28**, 2693–2704 (2002).
- [110] Cowan, G., Cranmer, K., Gross, E. & Vitells, O. Asymptotic formulae for likelihood-based tests of new physics. *Eur. Phys. J. C* **71**, 1554 (2011). [Erratum: *Eur. Phys. J. C* **73**, 2501 (2013)], [1007.1727](#).
- [111] Haisch, U., Kamenik, J. F., Malinauskas, A. & Spira, M. Collider constraints on light pseudoscalars. *JHEP* **03**, 178 (2018). [1802.02156](#).
- [112] LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436–444 (2015).
- [113] Aad, G. *et al.* Operation of the ATLAS trigger system in Run 2. *JINST* **15**, P10004 (2020). [2007.12539](#).
- [114] ATLAS Collaboration. Technical Design Report for the Phase-II Upgrade of the ATLAS TDAQ System (2017). URL <https://doi.org/10.17181/CERN.2LBB.4IAL>. CERN-LHCC-2017-020, ATLAS-TDR-029.
- [115] Zabi, A., Berryhill, J. W., Perez, E. & Tapper, A. D. The Phase-2 Upgrade of the CMS Level-1 Trigger (2020). URL <https://cds.cern.ch/record/2714892>. CERN-LHCC-2020-004, CMS-TDR-021.
- [116] Chandola, V., Banerjee, A. & Kumar, V. Anomaly detection: A survey. *ACM Comput. Surv.* **41** (2009). URL <https://doi.org/10.1145/1541880.1541882>.
- [117] Rumelhart, D. E., Hinton, G. E. & Williams, R. J. Learning representations by back-propagating errors. *Nature* **323**, 533–536 (1986).
- [118] LeCun, Y. *et al.* Handwritten digit recognition with a back-propagation network. In Touretzky, D. (ed.) *Advances in Neural Information Processing Systems*, vol. 2 (Morgan-Kaufmann, 1989). URL https://proceedings.neurips.cc/paper_files/paper/1989/file/53c3bce66e43be4f209556518c2fcb54-Paper.pdf.
- [119] Lecun, Y., Bottou, L., Bengio, Y. & Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* **86**, 2278–2324 (1998).

- [120] Kastner, R., Matai, J. & Neuendorffer, S. Parallel Programming for FPGAs. *ArXiv e-prints* (2018). [1805.03648](#).
- [121] FastML Team. fastmachinelearning/hls4ml. <https://doi.org/10.5281/zenodo.1201549> (2023). URL <https://github.com/fastmachinelearning/hls4ml>.
- [122] Paszke, A. *et al.* Pytorch: An imperative style, high-performance deep learning library. In Wallach, H. *et al.* (eds.) *Advances in Neural Information Processing Systems*, vol. 32 (Curran Associates, Inc., 2019). URL https://proceedings.neurips.cc/paper_files/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf.
- [123] Summers, S. *et al.* Fast inference of Boosted Decision Trees in FPGAs for particle physics. *JINST* **15**, P05026 (2020). [2002.02534](#).
- [124] Aarrestad, T. *et al.* Fast convolutional neural networks on FPGAs with hls4ml. *Mach. Learn. Sci. Tech.* **2**, 045015 (2021). [2101.05108](#).
- [125] LeCun, Y., Denker, J. & Solla, S. Optimal brain damage. In Touretzky, D. (ed.) *Advances in Neural Information Processing Systems*, vol. 2 (Morgan-Kaufmann, 1989). URL https://proceedings.neurips.cc/paper_files/paper/1989/file/6c9882bbac1c7093bd25041881277658-Paper.pdf.
- [126] Han, S., Mao, H. & Dally, W. J. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. In *International Conference on Learning Representations (ICLR)* (2016). [1510.00149](#).
- [127] Louizos, C., Welling, M. & Kingma, D. P. Learning sparse neural networks through l_0 regularization. In *International Conference on Learning Representations* (2018). [1712.01312](#).
- [128] Gupta, S., Agrawal, A., Gopalakrishnan, K. & Narayanan, P. Deep learning with limited numerical precision. In Bach, F. & Blei, D. (eds.) *Proceedings of the 32nd International Conference on Machine Learning*, vol. 37 of *Proceedings of Machine Learning Research*, 1737–1746 (PMLR, Lille, France, 2015).
- [129] CMS Collaboration. Anomaly Detection in the CMS Global Trigger Test Crate for Run 3 (2023). URL <https://cds.cern.ch/record/2876546>. CERN-CMS-DP-2023-079.
- [130] Strassler, M. J. Why Unparticle Models with Mass Gaps are Examples of Hidden Valleys (2008). [0801.0629](#).
- [131] Knapen, S., Pagan Griso, S., Papucci, M. & Robinson, D. J. Triggering Soft Bombs at the LHC. *JHEP* **08**, 076 (2017). [1612.00850](#).
- [132] Rumelhart, D. E., Hinton, G. E. & Williams, R. J. Learning internal representations by error propagation. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1: Foundations*, 318–362 (MIT Press, Cambridge, MA, USA, 1986).

- [133] Govorkova, E. *et al.* Autoencoders on field-programmable gate arrays for real-time, unsupervised new physics detection at 40 MHz at the Large Hadron Collider. *Nature Mach. Intell.* **4**, 154–161 (2022). [2108.03986](#).
- [134] Buciluundefined, C., Caruana, R. & Niculescu-Mizil, A. Model compression. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, 535–541 (Association for Computing Machinery, New York, NY, USA, 2006). URL <https://doi.org/10.1145/1150402.1150464>.
- [135] Hinton, G., Vinyals, O. & Dean, J. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop* (2015). [1503.02531](#).
- [136] Xilinx. Vivado Design Suite User Guide: High-Level Synthesis. https://www.xilinx.com/support/documentation/sw_manuals/xilinx2020_1/ug902-vivado-high-level-synthesis.pdf (2020).
- [137] Koza, J. Genetic programming as a means for programming computers by natural selection. *Statistics and Computing* **4**, 87–112 (1994).
- [138] Schmidt, M. & Lipson, H. Distilling free-form natural laws from experimental data. *Science* **324**, 81–85 (2009).
- [139] Cava, W. L. *et al.* Contemporary symbolic regression methods and their relative performance. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)* (2021).
- [140] Virgolin, M. & Pissis, S. P. Symbolic regression is NP-hard. *Transactions on Machine Learning Research* (2022). [2207.01018](#).
- [141] Cranmer, M. Interpretable Machine Learning for Science with PySR and SymbolicRegression.jl (2023). URL <https://github.com/MilesCranmer/PySR>. [2305.01582](#).
- [142] Cranmer, M. *et al.* Discovering symbolic models from deep learning with inductive biases. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. & Lin, H. (eds.) *Advances in Neural Information Processing Systems*, vol. 33, 17429–17442 (Curran Associates, Inc., 2020). URL https://proceedings.neurips.cc/paper_files/paper/2020/file/c9f2f917078bd2db12f23c3b413d9cba-Paper.pdf.
- [143] Martius, G. & Lampert, C. H. Extrapolation and learning equations (2016). [1610.02995](#).
- [144] Sahoo, S. S., Lampert, C. H. & Martius, G. Learning equations for extrapolation and control (2018). [1806.07259](#).
- [145] Werner, M., Junginger, A., Hennig, P. & Martius, G. Informed equation learning (2021). [2105.06331](#).
- [146] Kim, S. *et al.* Integration of neural network-based symbolic regression in deep learning for scientific discovery. *IEEE Transactions on Neural Networks and Learning Systems* **32**, 4166–4177 (2021).

- [147] Tsoi, H. F., Loncar, V., Dasu, S. & Harris, P. SymbolNet: Neural Symbolic Regression with Adaptive Dynamic Pruning (2024). **2401.09949**.
- [148] LIU, J., XU, Z., SHI, R., Cheung, R. C. C. & So, H. K. Dynamic sparse training: Find efficient sparse network from scratch with trainable masked layers. In *International Conference on Learning Representations* (2020). **2005.06870**.
- [149] Meurer, A. *et al.* SymPy: symbolic computing in Python. *PeerJ Computer Science* **3**, e103 (2017). URL <https://doi.org/10.7717/peerj-cs.103>.
- [150] Pierini, M., Duarte, J. M., Tran, N. & Freytsis, M. HLS4ML LHC Jet dataset (150 particles) (2020). URL <https://doi.org/10.5281/zenodo.3602260>.
- [151] Coleman, E. *et al.* The importance of calorimetry for highly-boosted jet substructure. *JINST* **13**, T01003 (2018). **1709.08705**.
- [152] Moreno, E. A. *et al.* JEDI-net: a jet identification algorithm based on interaction networks. *Eur. Phys. J. C* **80**, 58 (2020). **1908.05318**.
- [153] Pedregosa, F. *et al.* Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* **12**, 2825–2830 (2011).