

Higgs Maxwell workshop 2025

# Machine Learning applications to Higgs analyses in ATLAS

Giuseppe Callea



University  
of Glasgow

19/02/2025

# Outline

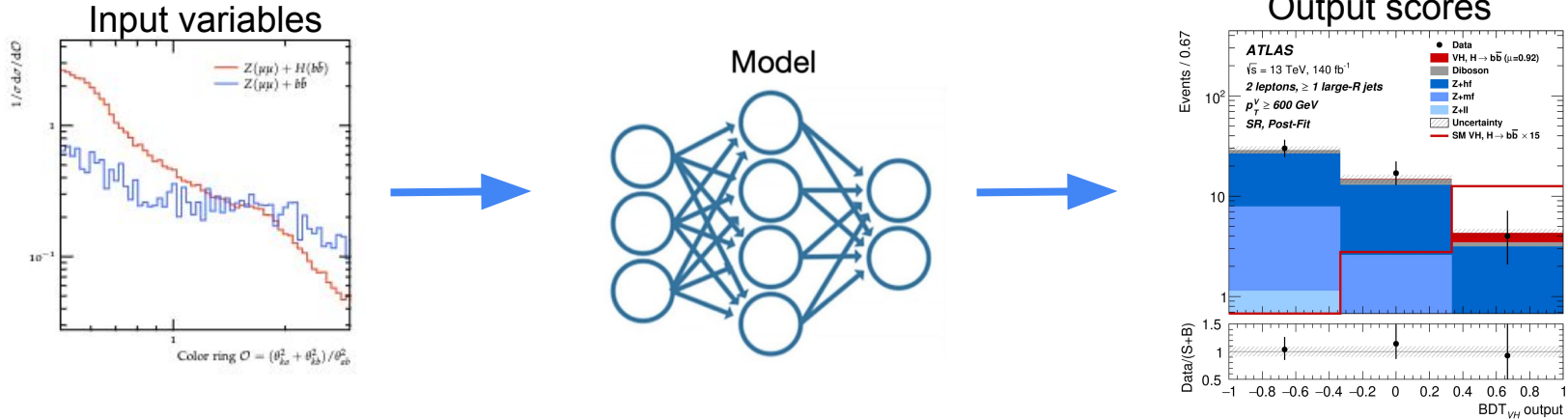
- Overview of Machine Learning (ML) techniques in ATLAS
- Historical facts and ML evolution throughout the years
- Examples in particle reconstruction and simulation
- Selected analyses
- Preservation
- Summary and outlook

# Disclaimers

- I'm more an "experienced" user of ML techniques rather than an expert
- AI/ML is sprouting in the HEP field, today's talk based on a selection of architectures and techniques



# How is Machine Learning used in ATLAS?



Two main general use cases:

- **Classification:** Separating different classes of data (e.g. signal vs background), outputs are classification scores
- **Regression:** Compute underlying variables, outputs are estimations of those variables values



# Assessing the results

Different metrics for assessing model's performance

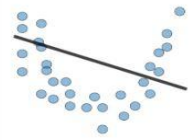

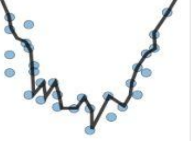
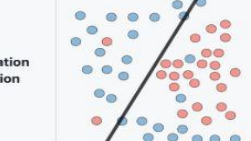
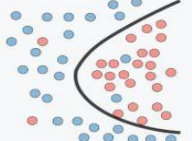
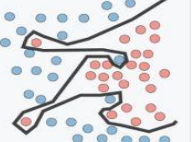


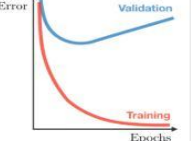
- Classification:
  - Most common looking at ROC Curves (Receiver Operating Characteristics)
  - Also Precision and Recall which can be more useful for imbalanced classes
- Regression: Mean Absolute error, Root Mean Square error etc.

# What can go wrong: Over-fitting

When it models the training data too well, e.g., learns the detail and noise in the training data to the extent that it negatively impacts the performance on new data

This means that the noise or random fluctuations in the training data is picked up and learned as concepts by the model

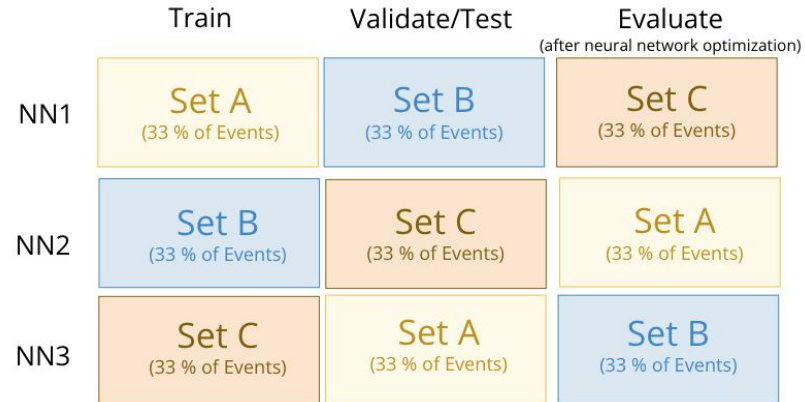
Not an exact agreement for what level of over-fitting is acceptable (Usually expect training performance to be a bit better than test set)

	Underfitting	Just right	Overfitting
Symptoms	<ul style="list-style-type: none"><li>• High training error</li><li>• Training error close to test error</li><li>• High bias</li></ul>	<ul style="list-style-type: none"><li>• Training error slightly lower than test error</li></ul>	<ul style="list-style-type: none"><li>• Very low training error</li><li>• Training error much lower than test error</li><li>• High variance</li></ul>
Regression illustration			
Classification illustration			
Deep learning illustration			
Possible remedies	<ul style="list-style-type: none"><li>• Complexify model</li><li>• Add more features</li><li>• Train longer</li></ul>		<ul style="list-style-type: none"><li>• Perform regularization</li><li>• Get more data</li></ul>

# Cross validation

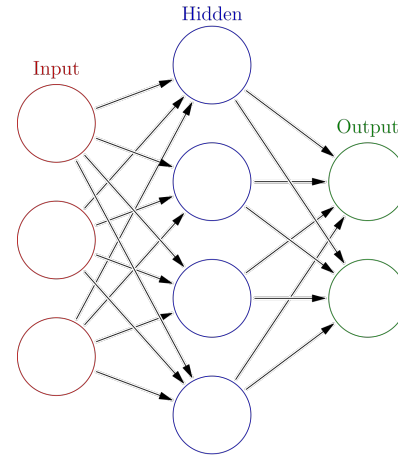
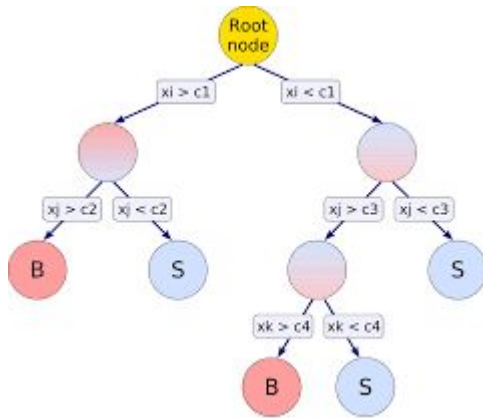
When actually using the model, ensure that a given data/MC event is not being evaluated with a model for which that event was part of the training data (otherwise will have a bias)

- Common way to avoid this is to use a cross validation setup - split data into folds and train multiple models (each with same hyperparameters, input features etc)
- For final analysis events evaluated using a model for which it was not part of the training
- This ensures the performance of the different models is consistent (otherwise possible indication of problematic over-fitting)



# Which Machine Learning tools are used in ATLAS?

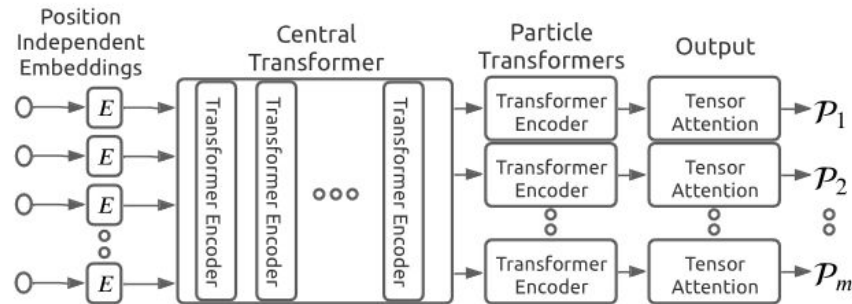
- Boosted Decision Trees (BDTs): Series of individual decision trees making cuts on input values to produce high-purity ‘leaves’. Subsequent trees are trained on the residuals of the previous tree (Boosted). This is controlled by the “learning rate” parameter
- Neural Networks: Deep (DNN), Recurrent (RNN), Parametric (PNN), Transformer, Graph (GNN)



# A word on Transformers

[Attention is all you need](#)

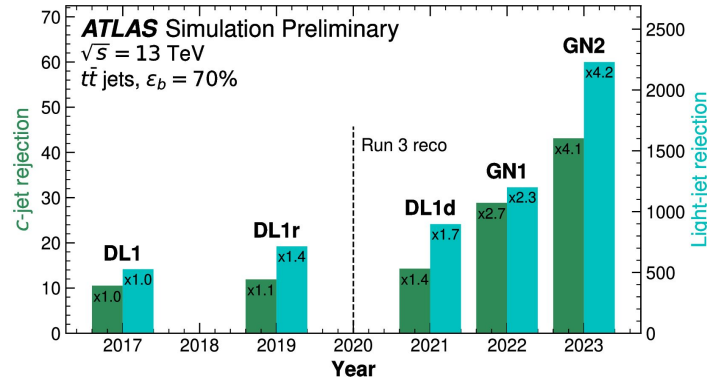
- “Attention-based ”Transformer architectures are now state of the art on many tasks in HEP and in natural language processing
- Invariant with respect to the order of the input sequence (e.g. jets 4-vectors + flavour tagging information)
- Very effective at modeling variable-length sets because they can learn combinatorial relationships between set elements with a polynomial run-time
- Usually trained to predict the probability of an event to be from a signal or from a background enriched region



# How is Machine Learning used in ATLAS?

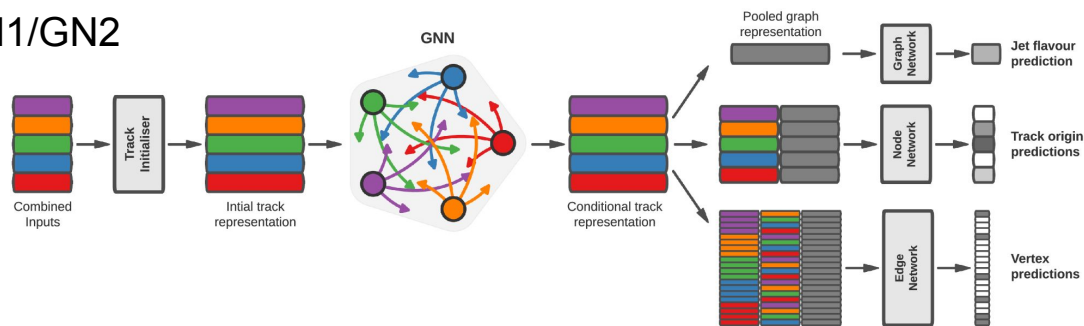
Vast number of applications, including object reconstruction and identification!

- Simulation -> Improve CPU performances against Geant4
- Jet flavour tagging using Graph Neural Networks (GNN) to discriminate heavy flavour jets (b- or c-jets) from light flavours
- Other object reconstruction (e.g. electrons, photons, taus) to improve their performances
- Re-analysis of ATLAS data -> providing tighter constraints on the analysis parameters



# Jet Flavour Tagging leading the revolution

- 1992: Multi Layered Perceptron at [LEP](#)
- 2006: First ML flavour tagging tool at a hadron collider at [D0](#)
- 2007: NNs for FTAG at [CDF](#)
- 2012-2017: [ATLAS](#) developing BDT based MV1 and MV2
- 2017-2019: [CMS](#) uses deep learning with DeepCSV
- 2017-2021: [ATLAS](#) is back to RNN and DNN DL1r and DL1d
- 2019-: [CMS](#) used ParticleNet
- 2021-: [ATLAS](#) moved to GN1/GN2

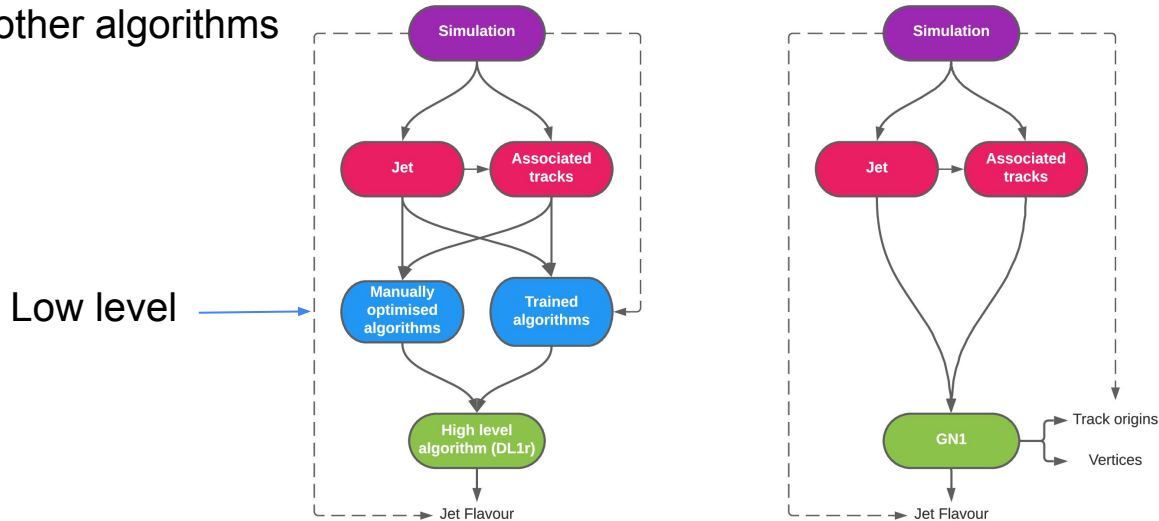


# Jet Flavour Tagging leading the revolution

[GN1 paper](#)

DL1r (left): Uses “**Low-level**” algorithms using tracks to reconstruct a particular aspect of the experimental signature of heavy flavour jets

GN1 (right): single neural network, directly taking tracks and some jet information as input, without depending on other algorithms





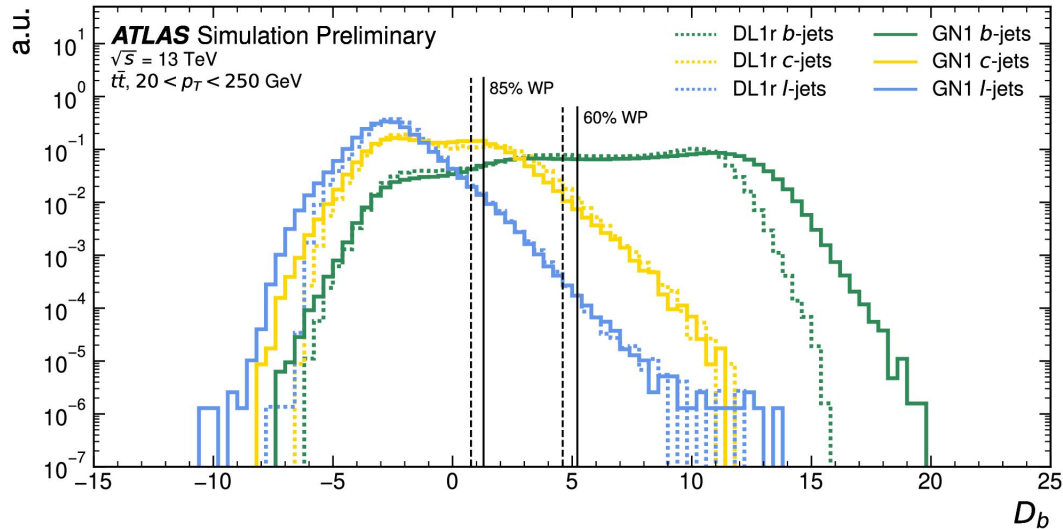
# Constructing the FTAG discriminants

[GN1 paper](#)

The high-level algorithms calculate the probabilities for each flavour class:  $p_b$  (b-jets),  $p_c$  (c-jets) and  $p_l$  (light-jets)

$$D_b = \log \frac{p_b}{(1 - f_c)p_l + f_c p_c}$$

$$D_c = \log \frac{p_c}{(1 - f_b)p_l + f_b p_b}$$



Similar shapes for  $b$ -,  $c$ - and light-jets. GN1 model shifts the  $b$ -jet distribution to higher values of  $D_b$  (regions with the best discrimination)

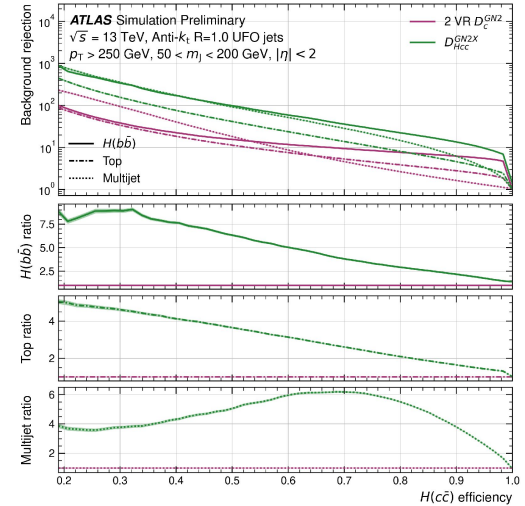
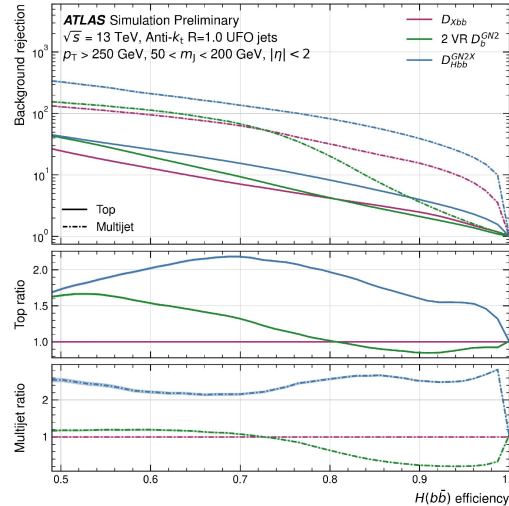
# The “boosted” variant

[Xbb/Xcc tagger paper](#)

A similar architecture can be exploited for Lorentz boosted H->bb tagging  
 GN2X achieves a background rejection factor of 40 for jets from top-quark decays and 300 for multijet event

$$D_{Hbb}^{GN2X} = \ln \left( \frac{P_{Hbb}}{f_{Hcc} \cdot P_{Hcc} + f_{top} \cdot P_{top} + (1 - f_{Hcc} - f_{top}) \cdot P_{QCD}} \right)$$

$f_{Hcc} = 0.02$  and  $f_{top} = 0.25$ , following an optimisation procedure to maximise the rejection for a given efficiency



# Fast Simulation

[ATLAS Run3 software paper](#)

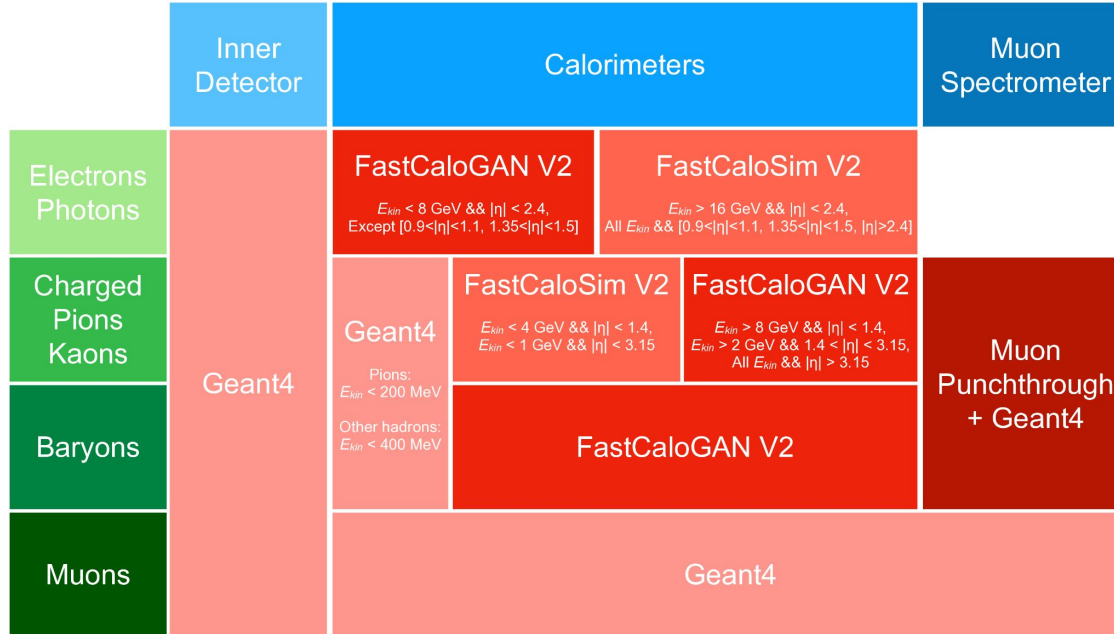
Geant4 “Full detector Simulation” requires considerable CPU resources

Faster simulation methods (AtI Fast3) replaces the calorimeter shower simulation (most CPU intensive step) tackling the slow propagation and interactions of incident particles with the direct generation of energy deposits in the calorimeters:

- the simulation of hadrons, photons and electrons in the calorimeters is handled by a combination of two fast simulations tools; FastCaloSimV2, using a parametric approach, and FastCaloGAN, which uses generative adversarial networks (GANs)
- FastCaloGAN is among the first tools based on generative models used for production in a large HEP experiment

# Fast Simulation

AtlFast3 requires only 20% of the CPU of the full simulation

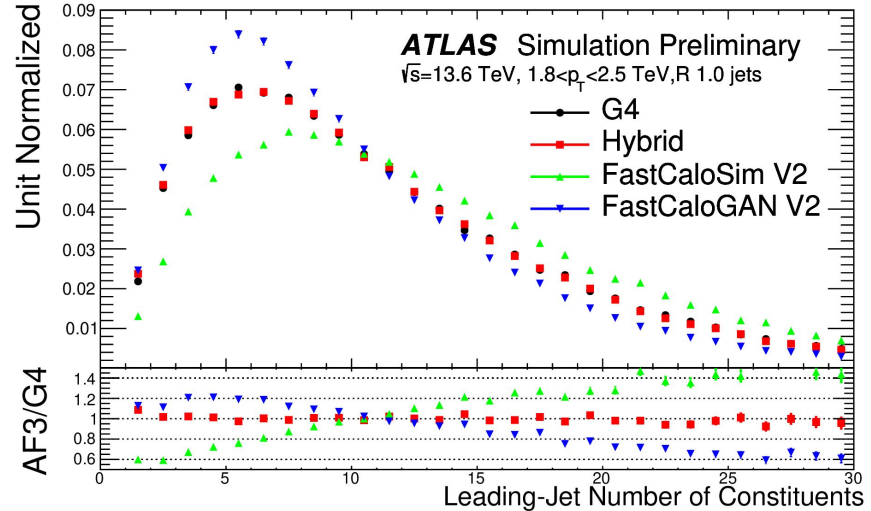
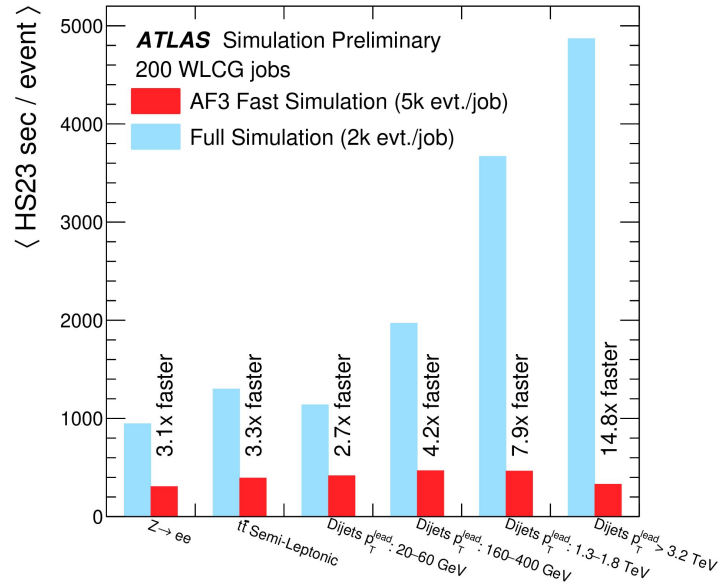


# Fast Simulation in Higgs analyses and beyond

[ATLAS Run3 software paper](#)

[AtlFast3 performance paper](#)

Fast Simulation tools are very useful when simulating complicated signals (e.g. di-Higgs, X->SH, BSM processes)



# Analysis Showcase

- VH (H->bb) legacy [arXiv:2410.19611](https://arxiv.org/abs/2410.19611)
- ttH (H->bb) legacy [arXiv:2407.10904](https://arxiv.org/abs/2407.10904)
- Resonant X->SH->bbyy [JHEP 11 \(2024\) 047](https://arxiv.org/abs/2407.10904)

Recent publications which involve interesting ML tools

Many interesting ATLAS result available here:

<https://twiki.cern.ch/twiki/bin/view/AtlasPublic>

## legacy analysis

[legəsi ə'nælısıs] adjective

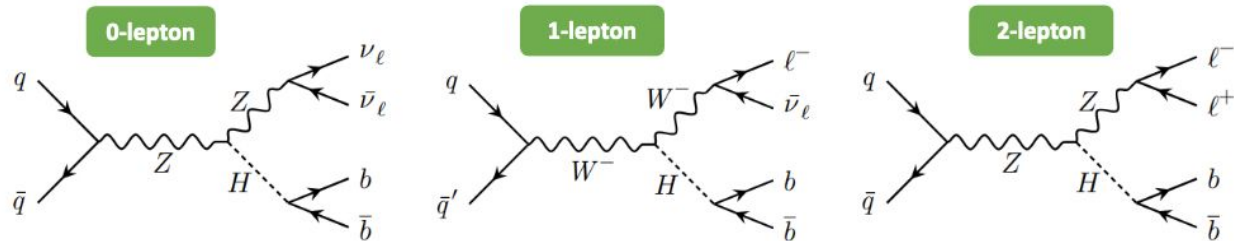
---

Re-analysis of a previous publication with improved object reconstruction and novel analysis techniques to provide improved sensitivity.

# VHbb/cc legacy analysis

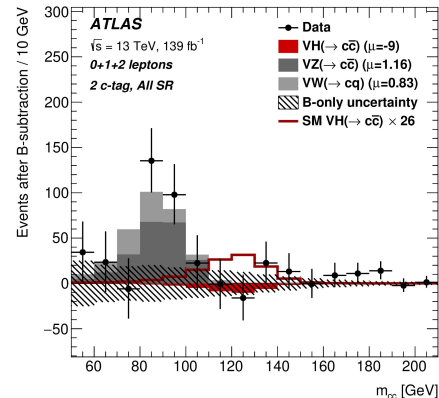
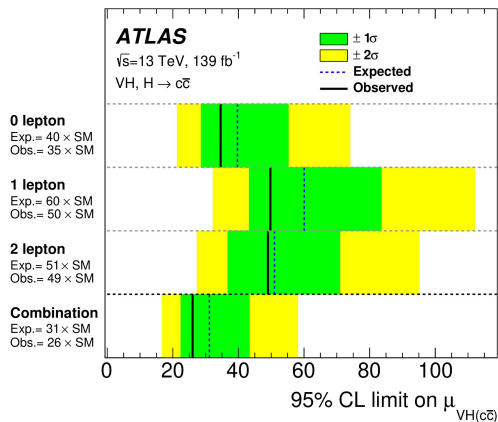
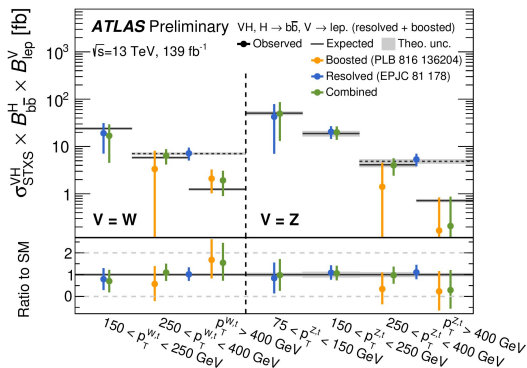
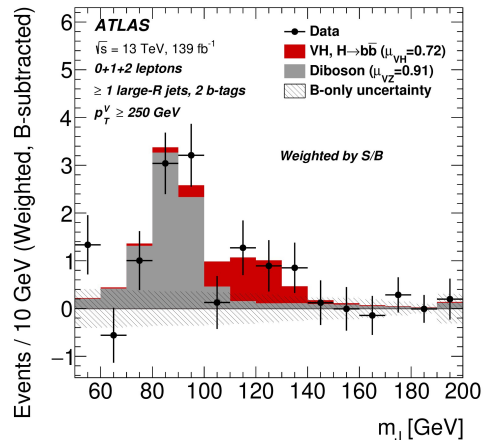
- Probing the Yukawa coupling in the quark sector is a cornerstone of the LHC programme
- Measuring the coupling to the b-quark is reaching the precision era!
- Growing interest in the coupling towards the 2nd generation quarks: accessible by the LHC?

Study associate production of  $H \rightarrow bb$  and  $H \rightarrow cc$  and a leptonically decaying vector boson ( $V=W, Z$ ) to avoid vast hadronic jets background: three lepton final states (0-, 1-, 2-leptons)



# Previous VHbb/cc publications

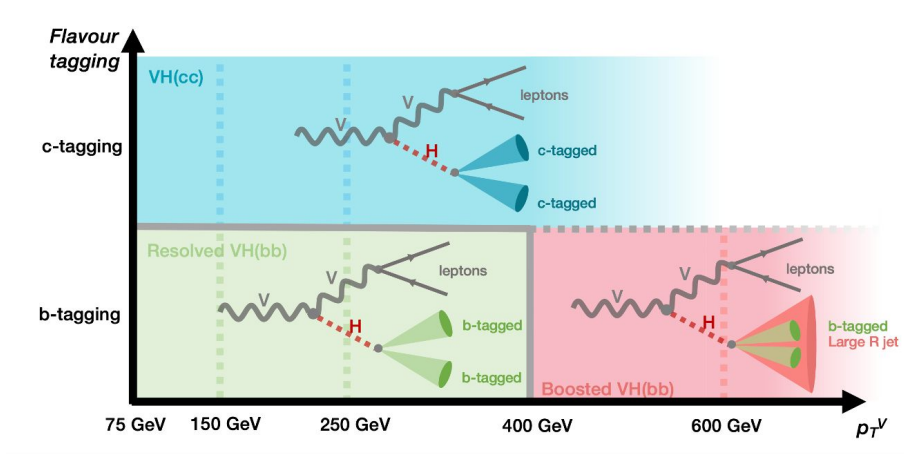
- Resolved VH(bb) analysis [link](#)
- Boosted VH(bb) analysis [link](#)
- VH(cc) analysis & VH(cc) + resolved VH(bb) combination [link](#)
- VHbb resolved + boosted combination [link](#)





# VHbb/cc legacy analysis

“Resolved”:  
R=0.4 jets for  
flavour tagging



“Boosted”  
R=1 jet and check  
the variable-R (VR)  
track-jets for  
flavour tagging

Flavour Tagging and the vector boson transverse momenta used to characterise the analysis

# VHbb/cc Legacy improvements

The legacy analysis makes use of the strategies from the previous publications but also includes some R&D:

- Particle flow jet collections (combining tracking and calorimeter information)
  - Improved energy and angular resolution of jets compared to techniques that only use the calorimeter in the central region of the detector
- Introduced a **BDT-based MVA technique** in the boosted VHbb and VHcc regimes
- Switch to **DL1r** flavour tagging algorithm (MV2c10 before)
  - Switching from BDT to RNN
  - New variables to enhance the signal-background discrimination
- Novel techniques to estimate the modelling uncertainties

# Flavour tagging (resolved VHbb + VHcc)

Fundamental ingredient for this analysis

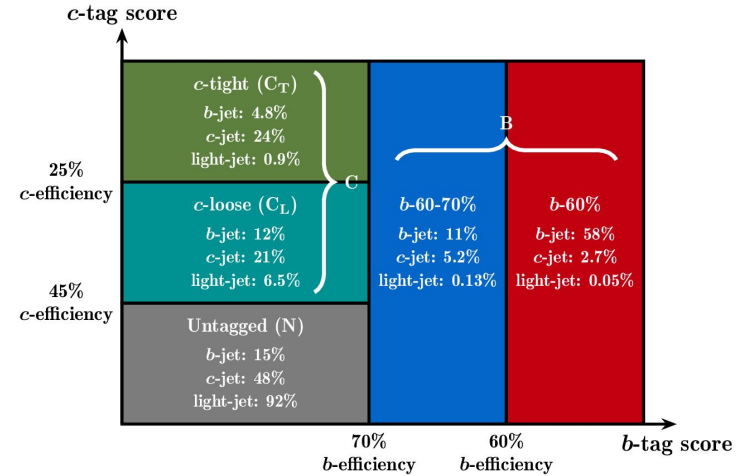
2D pseudo-continuous flavour tagging scheme

$$DL1r_b = \ln \left( \frac{p_b}{f_c p_c + (1 - f_c) p_u} \right)$$

$$DL1r_c = \ln \left( \frac{p_c}{f_b p_b + (1 - f_b) p_u} \right)$$

$p_x$  is the flavour probability given by the DL1r algorithm

$f_y$  is the effective y-jet fraction in the background hypothesis ( $f_b = 0.018$  and  $f_c = 0.3$ )



# Flavour tagging (boosted VHbb)

- Using VR track-jets as b-tagging input, using the standard Pseudo Continuous b-tagging (PCBT) calibration scheme
- Moved to DL1r at 85% WP; It was MV2c10 at 70% before (better c-/light-jet rejection, so to cover more signal events in the low statistics boosted region without increasing too much background events)
- Switch from  $m_{BB}$  to MVA score as final discriminant provides better signal and background separation power (improving the significance by ~50%)

# The taggings

**Direct tagging (DT)**: Cut-based method based on the jet score. Causing a lack in statistics in some phase space regions, so they cannot model the background effectively, leading to a large MC statistical uncertainty

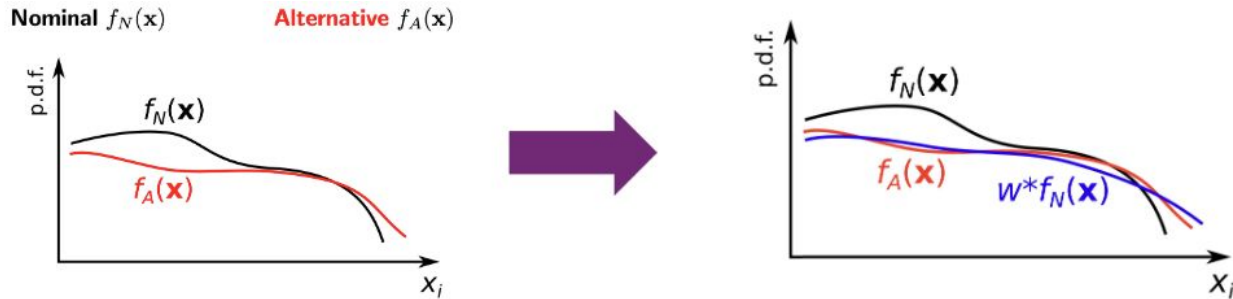
**Truth tagging (TT)**: reweight events based on their probability to pass tagging requirement. Showed a reasonable good closure with the direct tagged distribution -> **estimated with a GNN**

**Hybrid tagging (HT)**: DT b-jets and then TT the other flavours. Chosen to maximise the TT benefits and reduce the potential impact from the observed slightly non-closure

	VHbb resolved	VHbb boosted	VHcc
Hybrid tagging	Yes (b-jets are DT'd)	No (fully TT'd)	Yes (b-jets are DT'd)
Truth tag WP	70% $b$ & 70% $b$	85% $b$ & 85% $b$	c-tight & c-tight
MC stat. % for TT regions	100%	100%	8%
V+jets	hybrid tagged	truth tagged	hybrid tagged
single-top	hybrid tagged	truth tagged	hybrid tagged
$t\bar{t}$	direct tagged	truth tagged	direct tagged
diboson	direct tagged	truth tagged	direct tagged
signal	direct tagged	truth tagged	direct tagged

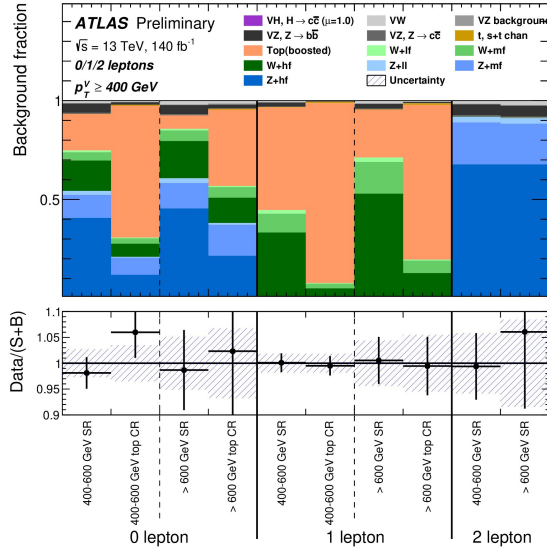
# Shape uncertainties: CARL

- Shape systematic uncertainties quantify the effects on the shape of the final fitting discriminant when comparing different MC samples (e.g. nominal and alternative)
- Calibrated Likelihood Ratio Estimator (CARL): a **DNN** to reweigh the nominal sample to make it look like the alternative sample to have shape uncertainties that are less susceptible to statistical fluctuations. This replaces BDT reweighting
- Inputs: all MVA variables + a few additions
- The reweighted nominal sample is then used as the CARL shape uncertainty

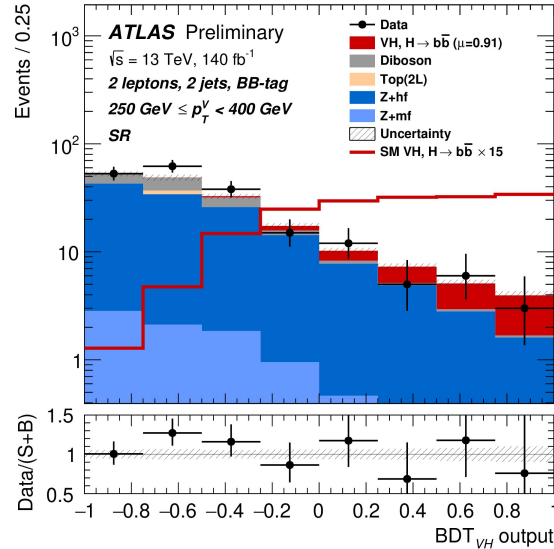


# VHbb/cc legacy analysis - Results

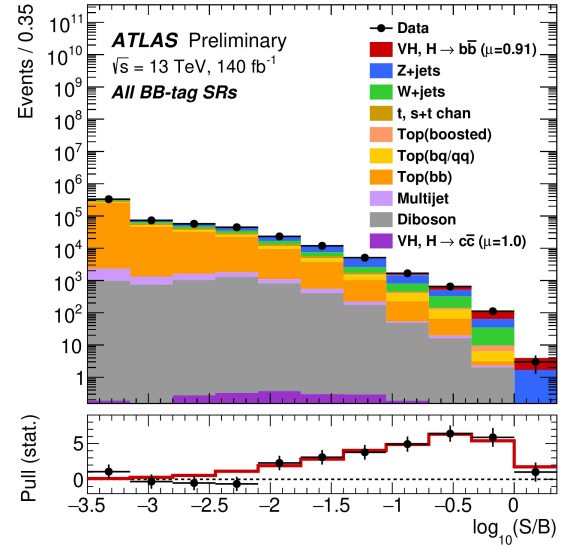
## Background composition



## BDT output for 2L boosted VHbb

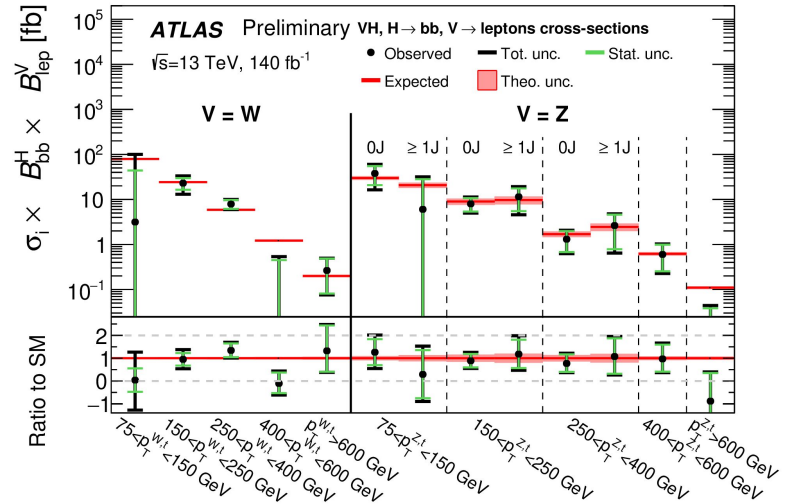
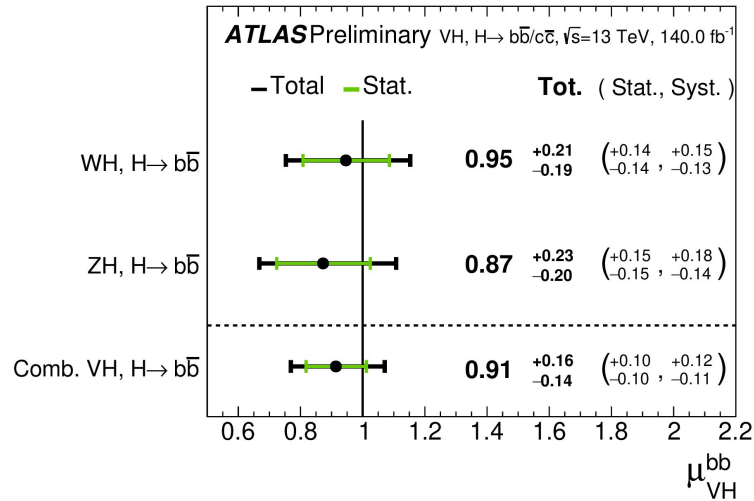


## Yields as a function of log(S/B)



# VHbb/cc legacy analysis - Results

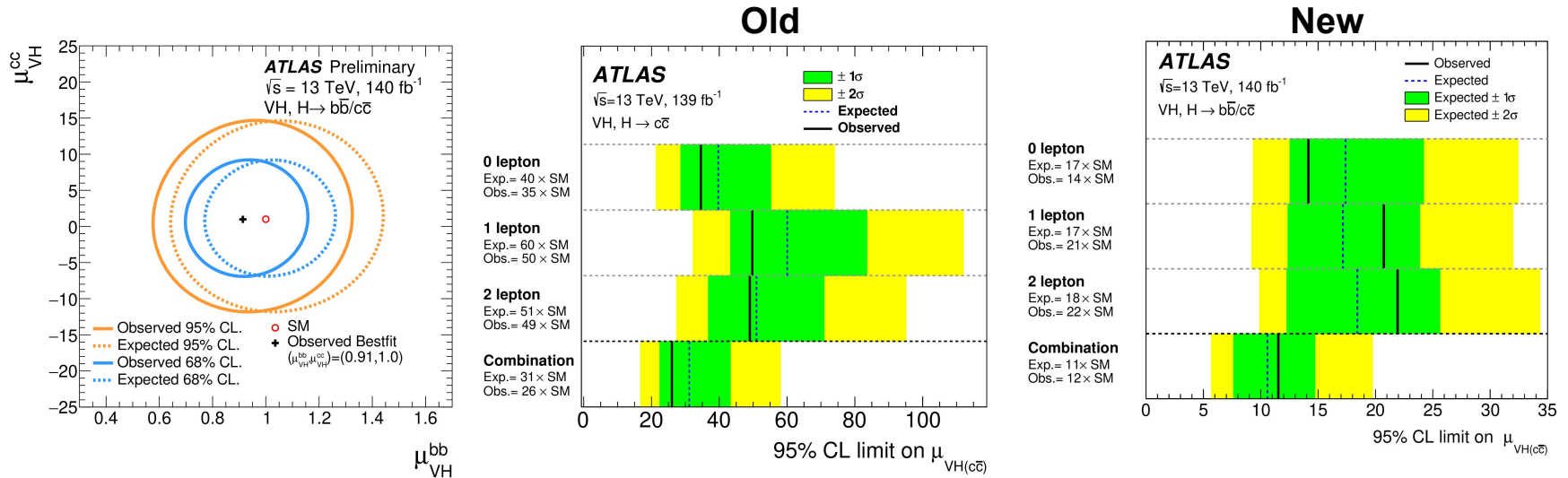
Good agreement with the SM. Individual production of WH and ZH established with observed (expected) significances of 5.3 (5.5) and 4.9 (5.7). Observation of WH(H→bb) production!





# VHbb/cc legacy analysis - Signal strength

$\mu_{bb} = 0.91 \pm 0.10$  (stat.)  $\pm 0.12$  (syst);  $\mu_{cc} = 1 \pm 4$  (stat.)  $\pm 3.6$  (syst), corresponding to an observed (expected) upper limit of 11.3 time the SM predictions at the 95% CL



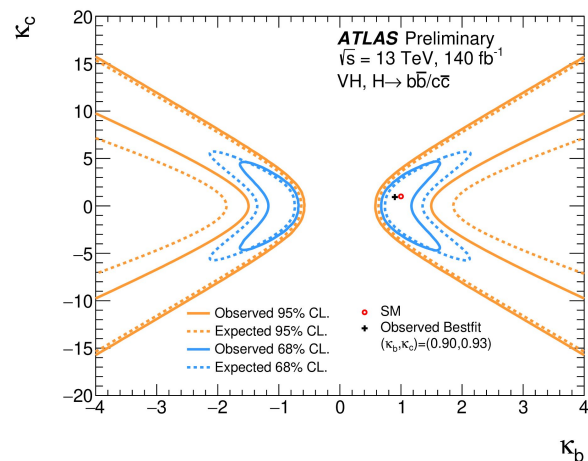
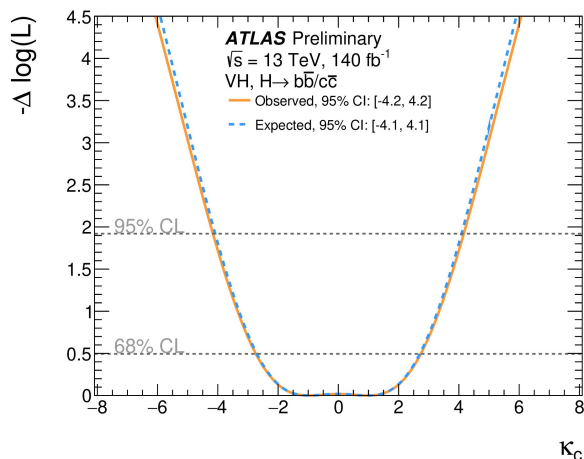
More than a factor 2 improvement

# kappa-framework

Best fit values interpreted in the kappa-framework as coupling modifiers  $\kappa_b$  and  $\kappa_c$

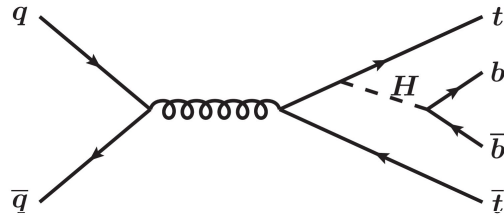
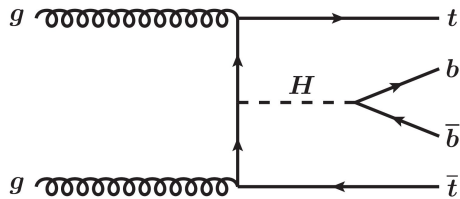
$$\mu_{VH}^{cc} = \frac{\kappa_c^2}{1 + B_{hbb}^{SM}(\kappa_b^2 - 1) + B_{hcc}^{SM}(\kappa_c^2 - 1)}$$

$$\mu_{VH}^{bb} = \frac{\kappa_b^2}{1 + B_{hbb}^{SM}(\kappa_b^2 - 1) + B_{hcc}^{SM}(\kappa_c^2 - 1)}$$



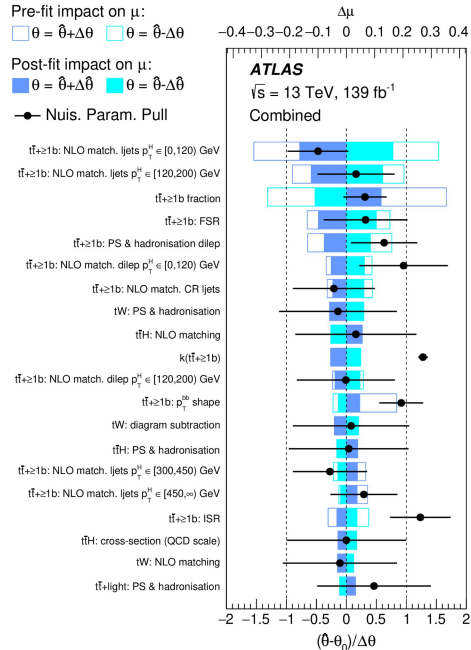
# ttH (H->bb) legacy analysis

- Provides direct access to the top Yukawa coupling (Only ~1% of Higgs production XS)  
Semi-leptonic decay of one (two) tops offer distinctive final signature in the single (dilepton) channel, avoiding multi-jet QCD background
- Studied in the single- and di-lepton channels (based on the W-boson decays)

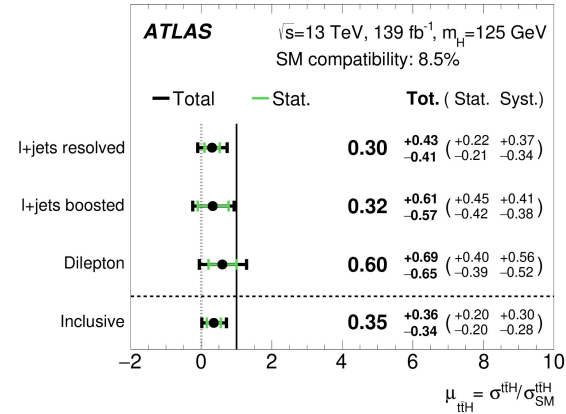


# ttH (H->bb) legacy analysis

- Exposed to irreducible tt+2b background
- First Run 2 results had a low overall inclusive signal strength. Dominated by modelling



Observed (expected) significance: 1.0 (2.7) $\sigma$



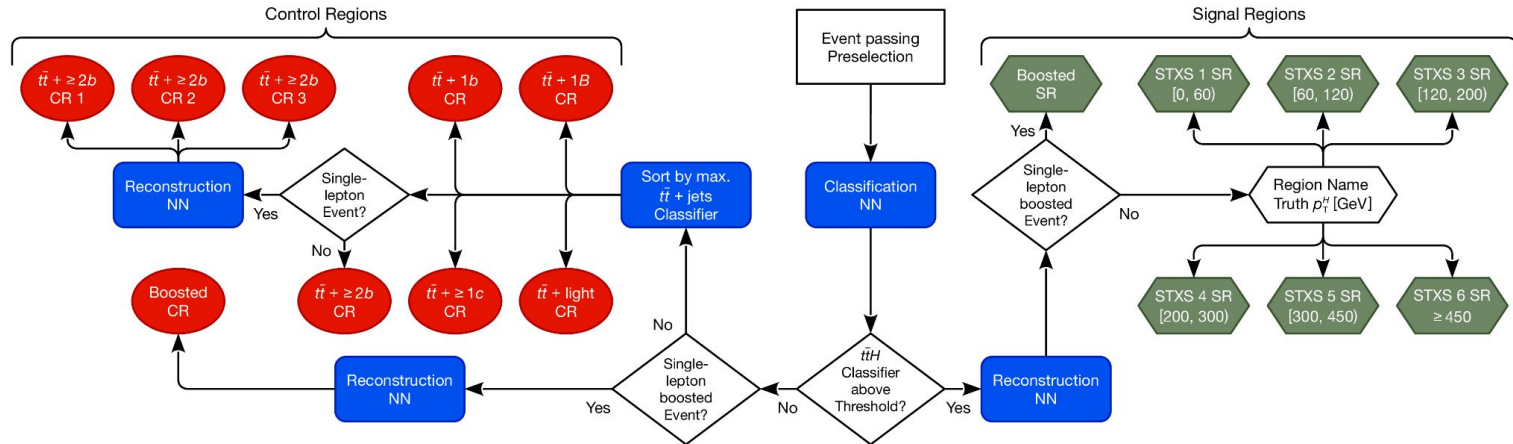
# ttH (H->bb) legacy improvements

- PFlow jets and DL1r b-tagger
- Loosened pre-selection for increase stats
- Higgs reco and classification done with Transformer NN with particle 4-vector inputs
- Additional background CRs
- New tt+bb nominal and systematics model developed in 4FS and a new 5FS systematic model for tt+c and tt+light

# $t\bar{t}H$ ( $H \rightarrow b\bar{b}$ ) legacy analysis

Higgs reco and classification done with attention based transformers

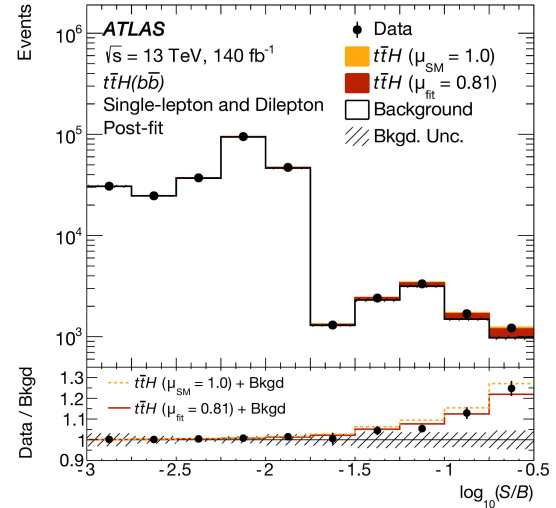
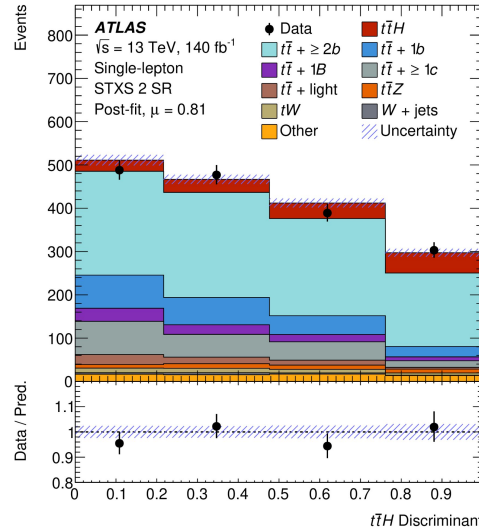
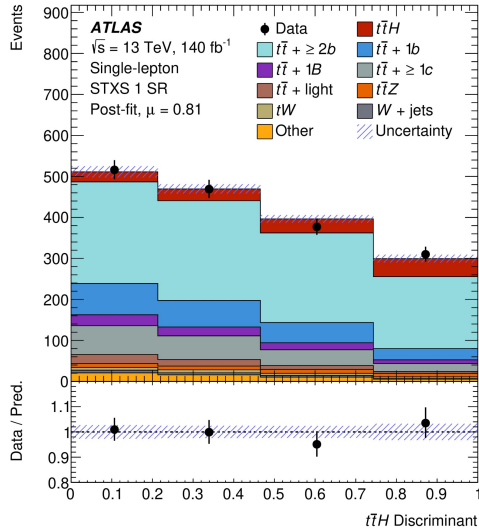
- Classification transformer to predict the probability for an event to be signal or background
- Reconstruction transformer to predict the ID of jets originating from the Higgs: predict which 2 jets most likely from H decay and gets the Higgs  $p_T$  from combining jet four-vectors



# $t\bar{t}H$ ( $H \rightarrow b\bar{b}$ ) legacy analysis

$$d_i = \frac{p_i}{\sum_{i \neq j} p_j \cdot \hat{N}_{ij}}$$

Classification transformer: Probability  $p_i$  of a network class  $i$  is converted into a discriminant  $d_i$  in order to maximise the separation between the class and all the other classes ( $i \neq j$ ), to yield a similar number of events in the control regions, and to maximise sensitivity.

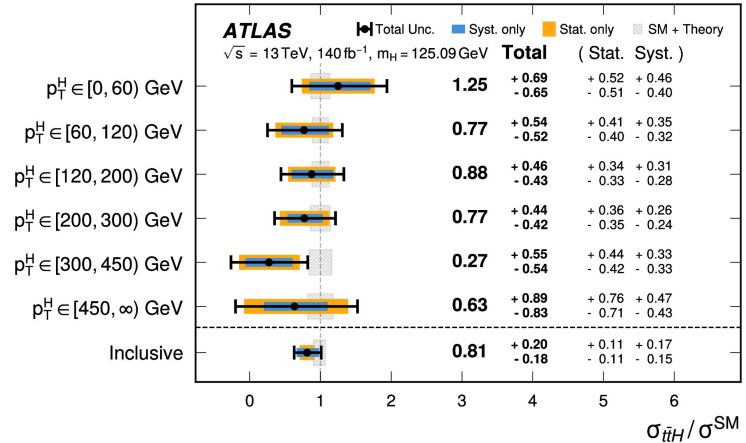
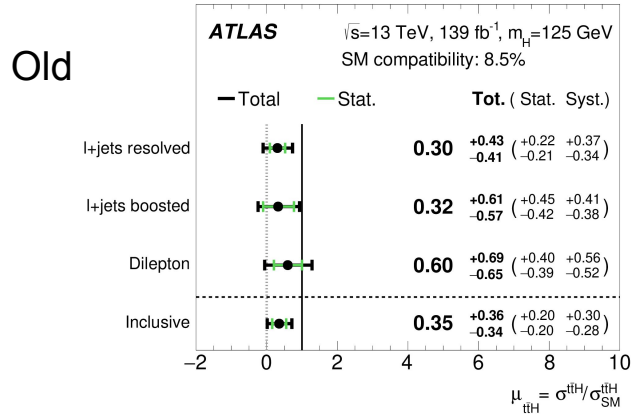


# ttH (H->bb) legacy analysis

Observed (expected) significance: 4.6 (5.4) $\sigma$  (was 1.0 (2.7)  $\sigma$  in the previous Run 2 analysis)

- Factor 2 improvement in expected sensitivity w.r.t. previous analysis
- Limited by systematic uncertainties
- STXS measurement compatible with SM p-value of 89%

Most precise single channel XS measurement for both inclusive and differential!



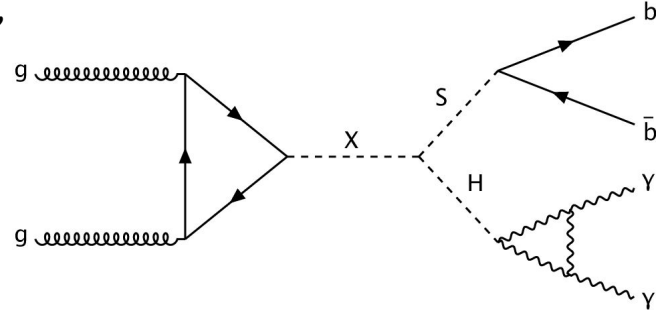


# X->SH->bbyy analysis

A search for the resonant production of a heavy scalar  $X$  decaying into a Higgs boson and a new lighter scalar  $S$ , through  $X \rightarrow S(\rightarrow bb)H (\rightarrow \gamma\gamma)$

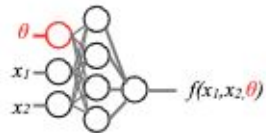
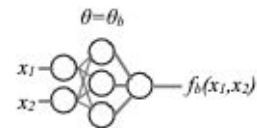
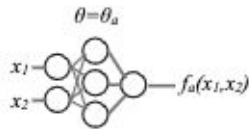
161 signal mass points are generated in the range  $170 \text{ GeV} \leq m_X \leq 1000 \text{ GeV}$  and  $15 \text{ GeV} \leq m_S \leq 500 \text{ GeV}$ .

The number of  $b$ -tagged jets is used to categorise events in two regions, requiring 1 or 2  $b$ -tagged jets. The signal events contain the characteristic  $H \rightarrow \gamma\gamma$  decay with the  $m_{\gamma\gamma}$  distribution peaking around the Higgs boson mass at  $\sim 125 \text{ GeV}$ .



# X->SH->bby analysis

- Parameterised neural networks (PNNs), which take input a vector of event characteristics and a vector of phase space parameters ( $\theta$ ). Yields to a function that is parameterised in  $\theta$
- Provides a unique discriminant for each signal hypothesis, separating the targeted signal events from background events
  - each value of  $\theta = (m_S, m_\chi)$ , the PNN( $\theta$ ) is effectively a different observables
- The PNNs provide sensitivity over the considered mass range and allow interpolation to values of  $\theta$  not explicitly included in the training

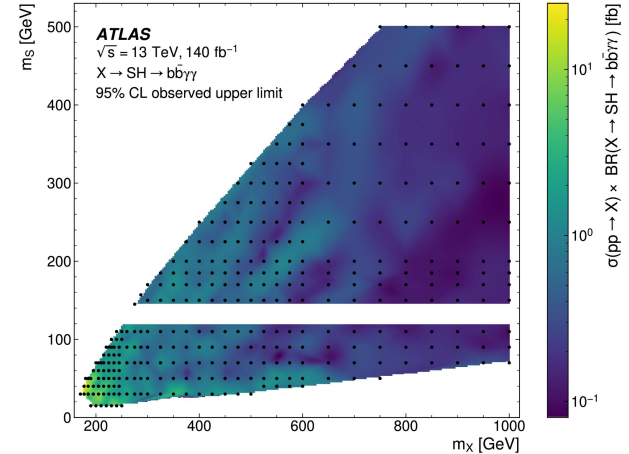
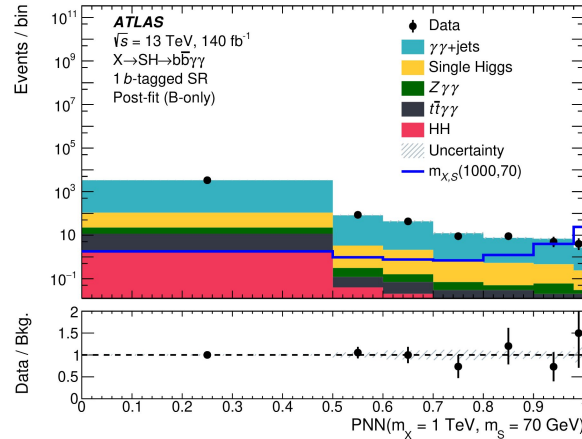
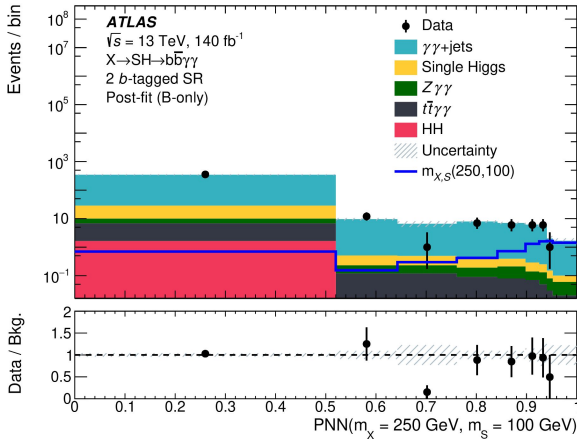


Instead of having 121 NNs you only have **one** parametric NN

# X->SH->bbyy analysis

No significant excess with respect to the SM background is found. 95% CL upper limits are set in the ranges  $170 \text{ GeV} \leq m_X \leq 1000 \text{ GeV}$  and  $15 \text{ GeV} \leq m_S \leq 500 \text{ GeV}$

The largest deviation from the background-only expectation occurs for  $(m_X, m_S) = (575, 200) \text{ GeV}$  with a local (global) significance of 3.5 (2.0) standard deviations



# Neural Network preservation

[Les Houches guide to reusable ML models in LHC analyses](#)

With the increasing usage of machine-learning in high-energy physics analyses, the publication of the trained models in a reusable form has become a crucial question for analysis preservation and reuse the ML model defines the analysis output, so that re-interpretability of the analysis directly depends on the reusability of the ML model

Open Neural Network Exchange (ONNX) as best tool for NN preservation, but not every framework provides ONNX files, in some cases the conversion is non-trivial

## **Les Houches guide to reusable ML models in LHC analyses**

*Jack Y. Araz<sup>1</sup>, Andy Buckley<sup>2</sup>, Gregor Kasieczka<sup>3</sup>, Jan Kieseler<sup>4</sup>, Sabine Kraml<sup>5</sup>, Anders Kvellestad<sup>6</sup>, Andre Lessa<sup>7</sup>, Tomasz Procter<sup>2</sup>, Are Raklev<sup>6</sup>, Humberto Reyes-Gonzalez<sup>8,9,10</sup>, Krzysztof Rolbiecki<sup>11</sup>, Sezen Sekmen<sup>12</sup>, Gokhan Unel<sup>13</sup>*

# Neural Network preservation

[Les Houches guide to reusable ML models in LHC analyses](#)

The long-term stability of the preservation format needs to be addressed. It is useful to advertise the exact software versions used to produce, save and run the neural network

On the reinterpretation tools side, all the major frameworks (CheckMATE, GAMBIT's Collider-Bit, MadAnalysis 5, Rivet, and ADL/CutLang) have developed interfaces for using published ML models. This was extensively discussed at the last two workshops of the Reinterpretation

# Summary and outlook

Many analyses, simulation and reconstruction activities are using complex ML methods using low-level event information (tracks, vertices, etc.)

Tools are now easier to use/more streamlined than ever before, with several tutorials available

However, ML tools are vulnerable to biases/bugs than more conventional methods

Still a lot to gain from studying Run-2 data! “Legacy” measurements showed significant improvements, mostly coming from updated ML techniques

**Thanks for your attention!**

# Backup

# Sequential vS Functional NNs

## **Sequential Neural Networks**

A sequential neural network is a linear stack of layers, organized in a sequence. Each layer has exactly one input tensor and one output tensor. This type of model is quite straightforward and ideal for tasks where the data flows in a single direction from input to output without any need for more complex connections.

## **Functional Neural Networks**

The functional API, on the other hand, allows for the creation of complex architectures, like multi-input/output models, shared layers, and models with non-linear topology. It's much more flexible and can handle more sophisticated designs.



# Attention is all you need

Dynamically weights the importance of different input parts, allowing the model to focus on the most relevant sections when making predictions.

1. Input Representation: The model processes input data (e.g., a sentence or an image) and generates a set of intermediate representations (often called "keys").
2. Query: For each part of the input sequence that the model is currently processing, it generates a "query" vector.
3. Weight Calculation: The query is compared with each key using a similarity function (like dot product). This generates a set of weights (or attention scores) that indicate the importance of each input part.
4. Weighted Sum: The weights are then used to compute a weighted sum of the input representations, emphasizing the most important parts.
5. Output: The weighted sum is then used as the input for the next step in the model.

In machine translation, an attention mechanism allows the model to focus on different parts of the input sentence while generating each word of the output sentence. This improves the quality of translations, especially for longer sentences.

# Attention is all you need

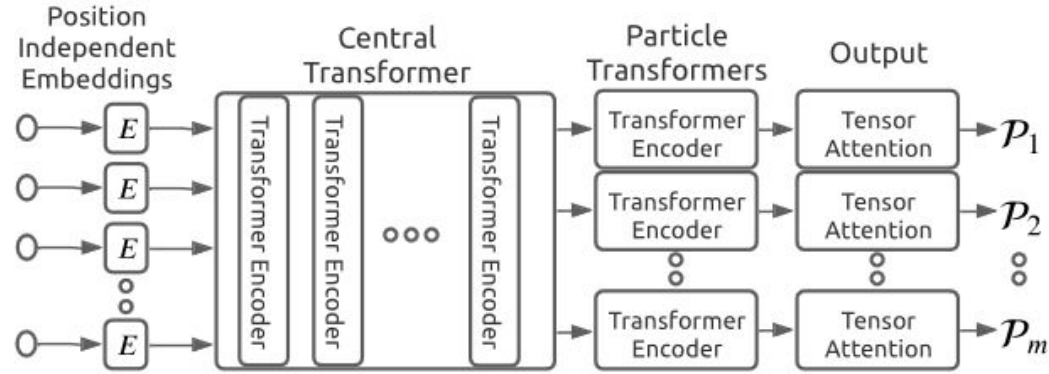
## Types of Attention Mechanisms

- Self-Attention: Used within the same sequence to compute the relationships between different positions. It's the backbone of the Transformer architecture.
- Cross-Attention: Computes relationships between different sequences, often used in tasks where the input and output sequences are different, like in translation.

## Benefits

- Handling Long Sequences: Attention helps models manage long sequences without the issues of vanishing gradients.
- Improved Interpretability: By visualizing attention weights, one can often interpret which parts of the input the model is focusing on.
- Flexibility: It can be applied in various domains like text, images, and even audio.

# A word on Transformers



1. independent jet embeddings to produce latent space representations for each jet
2. a central stack of transformer encoders
3. additional transformer encoders for each particle
4. a novel tensor-attention to produce the jet-parton assignment distributions.