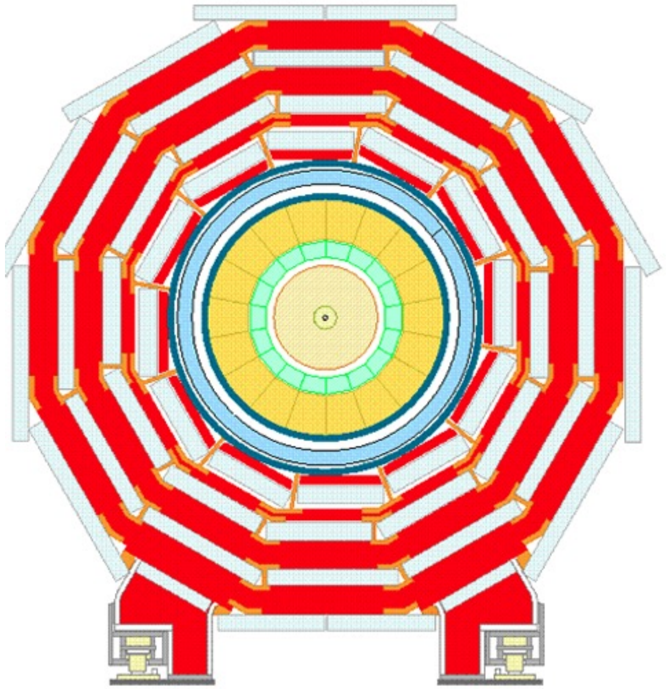
ν-flows for Run 3 ttH(bb)

DESY TOP Reco Meeting



Daina Leyva Pernia
Also for Maria and Matin

01.07.2025

HELMHOLTZ

The ttH(bb) process

- ttH provides the best direct probe of κ_t,
  H → bb largest branching fraction

- Final states:
  - **H → bb** (2 b-jets) + **2 top** (2 b-jets and 2 W bosons)
    - W decays:

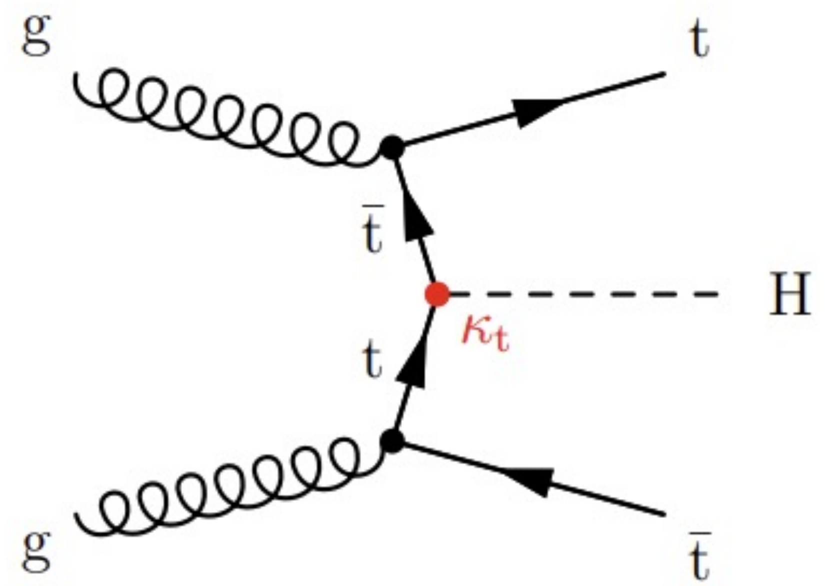
      FH (46%) : 4 jets

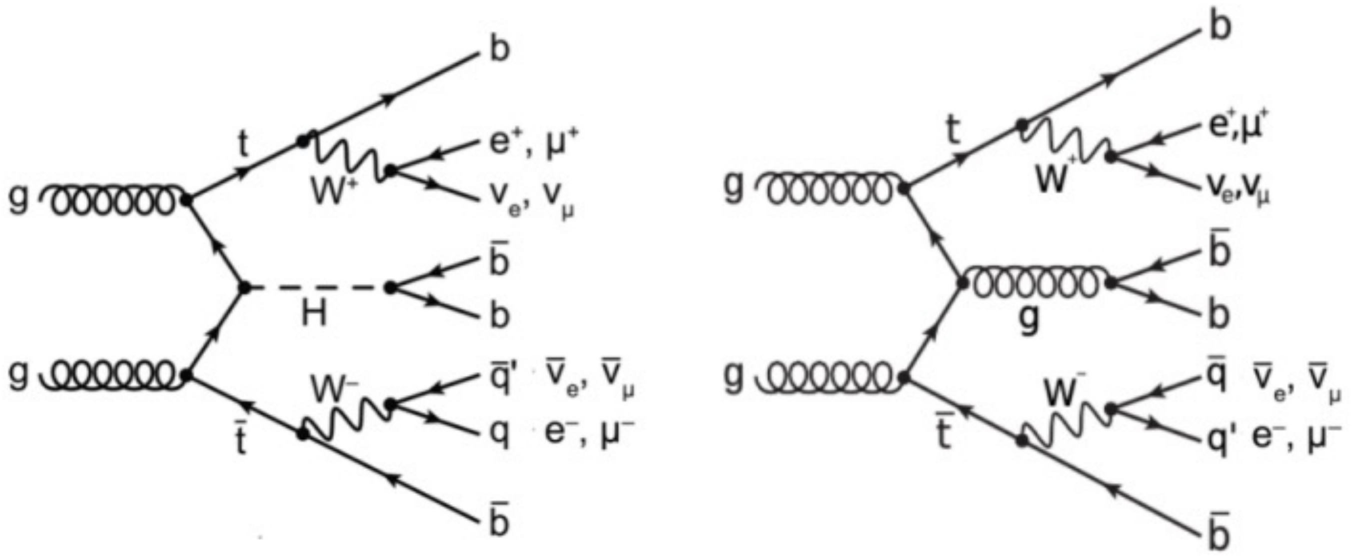
      SL (30%) : 2 jets + 1 lepton (e, μ) + 1 ν

      DL (4%) : 2 leptons + 2 ν

- Challenging process
  - Multiple jets with ambiguous assignment
  - Large irreducible ttbb (+QCD) background in leptonic (FH) final states

  To be addressed during Run 3 with MVA methods + more data-driven approaches (next slides)





Our approach for Run 3

Challenging process

- Multiple jets with ambiguous assignment
- Large irreducible ttbb (+QCD) background in leptonic (FH) final states

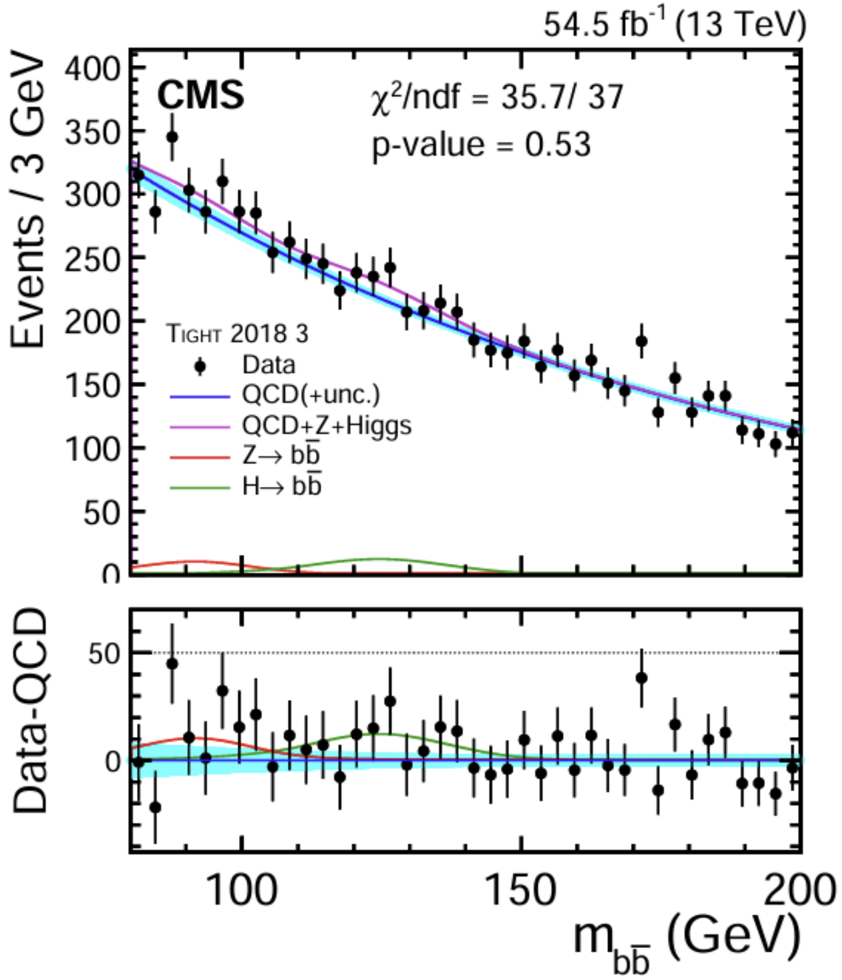
To be addressed with **S**ymmetry **P**reserving **A**ttention **N**etwork (SPANet), an attention based deep learning method developed for particle assignment

Aim for a more (fully?) data-driven background estimation

- Analytic fitting method on physically driven observable(s)
- Higgs invariant mass → strongly motivated observable
- Successfully applied in several analyses (e.g. $H\gamma\gamma$, VBF Hbb)

An approach à la VBF Hbb but facing further challenges

Is a kinematic fit in m_{bb} plausible for ttH(bb)?



JHEP 01 (2024) 173

Our approach for Run 3

Challenging process

- Multiple jets with ambiguous assignment
- Large irreducible ttbb (+QCD) background in leptonic (FH) final states

To be addressed with **S**ymmetry **P**reserving **A**ttention **N**etwork (SPANet), an attention based deep learning method developed for particle assignment

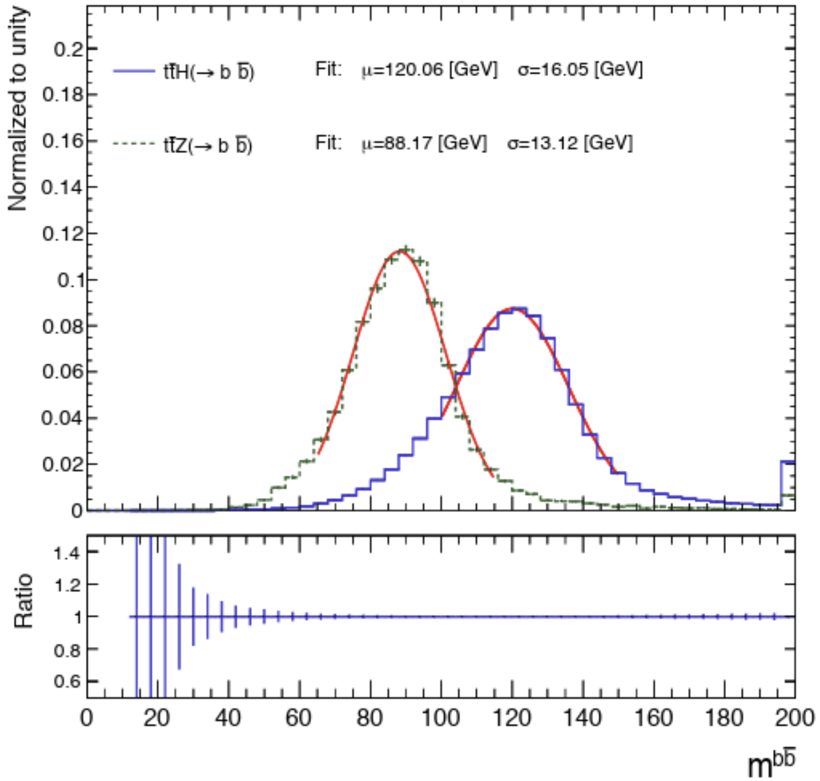
Aim for a more (fully?) data-driven background estimation

- Analytic fitting method on physically driven observable(s)
- Higgs invariant mass → strongly motivated observable
- Successfully applied in several analyses (e.g. H$\gamma\gamma$, VBF Hbb)

An approach à la VBF Hbb but facing further challenges

Is a kinematic fit in m_{bb} plausible for ttH(bb)?

- **YES!** → but it comes down to overcome jet assignment ambiguity

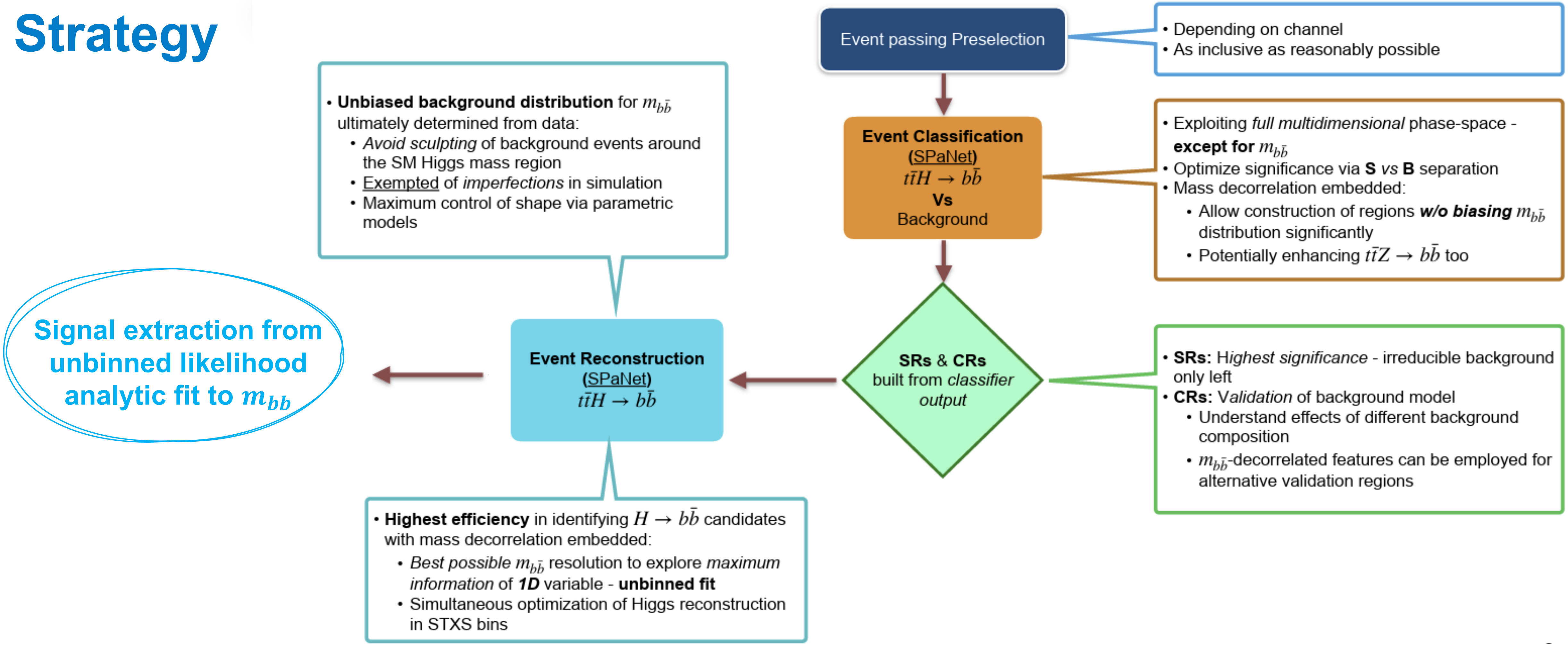


From D. Perez

H/Z invariant mass reconstructed from PF AK4 jets **with known origin**

Resolution comparable with VBF Hbb analysis (HIG-22-009)

Strategy



Event passing Preselection
• Depending on channel
• As inclusive as reasonably possible
Event Classification (**SPaNet**) $t\bar{t}H \to b\bar{b}$ **Vs** Background
• Exploiting *full multidimensional* phase-space - **except for** $m_{b\bar{b}}$
• Optimize significance via **S** *vs* **B** separation
• Mass decorrelation embedded:
• Allow construction of regions ***w/o biasing*** $m_{b\bar{b}}$ distribution significantly
• Potentially enhancing $t\bar{t}Z \to b\bar{b}$ too
SRs & **CRs** built from *classifier output*
• **SRs:** H*ighest significance* - irreducible background only left
• **CRs:** V*alidation* of background model
• Understand effects of different background composition
• $m_{b\bar{b}}$-decorrelated features can be employed for alternative validation regions
Event Reconstruction (**SPaNet**) $t\bar{t}H \to b\bar{b}$
• **Unbiased background distribution** for $m_{b\bar{b}}$ ultimately determined from data:
• *Avoid sculpting* of background events around the SM Higgs mass region
• Exempted of *imperfections* in simulation
• Maximum control of shape via parametric models
• **Highest efficiency** in identifying $H \to b\bar{b}$ candidates with mass decorrelation embedded:
• *Best possible* $m_{b\bar{b}}$ resolution to explore *maximum information* of ***1D*** variable - **unbinned fit**
• Simultaneous optimization of Higgs reconstruction in STXS bins
Signal extraction from unbinned likelihood analytic fit to m_{bb}

Jet-parton assignment with SPANet

(DL channel)

Inputs, assignment targets, regression (under study)

```
INPUTS:
  SEQUENTIAL:
    JetGood:
      pt: log_normalize
      eta: normalize
      mass: log_normalize
      phi: normalize
      btagRobustParTAK4B: none
  GLOBAL:
    PositiveLepton:
      pt: log_normalize
      eta: normalize
      phi: normalize
      mass: log_normalize
    NegativeLepton:
      pt: log_normalize
      eta: normalize
      phi: normalize
      mass: log_normalize
    Neutrinos:
      px: normalize
      py: normalize
      pz: normalize
    anti_Neutrinos:
      px: normalize
      py: normalize
      pz: normalize
```

```
EVENT:
  T1:
    - B
  T2:
    - B
  H:
    - B1
    - B2
PERMUTATIONS:
  H:
    - [ B1, B2 ]
```

```
REGRESSIONS:
  EVENT:
    - Neutrinos_px
    - Neutrinos_py
    - Neutrinos_pz
    - anti_Neutrinos_px
    - anti_Neutrinos_py
    - anti_Neutrinos_pz
```

- Assessing performance through assignment efficiency (eff)

$$eff = \frac{Events\ correctly\ reconstructed}{Number\ of\ possible\ reconstructions}$$

- How distributions are defined:

Prediction: SPANet assignment

Truth: True assignments between the LHE/GEN level particles and the RECO jets

Number of events	
Training	458558
Validation	98261
Test	98266

From M. Torkian

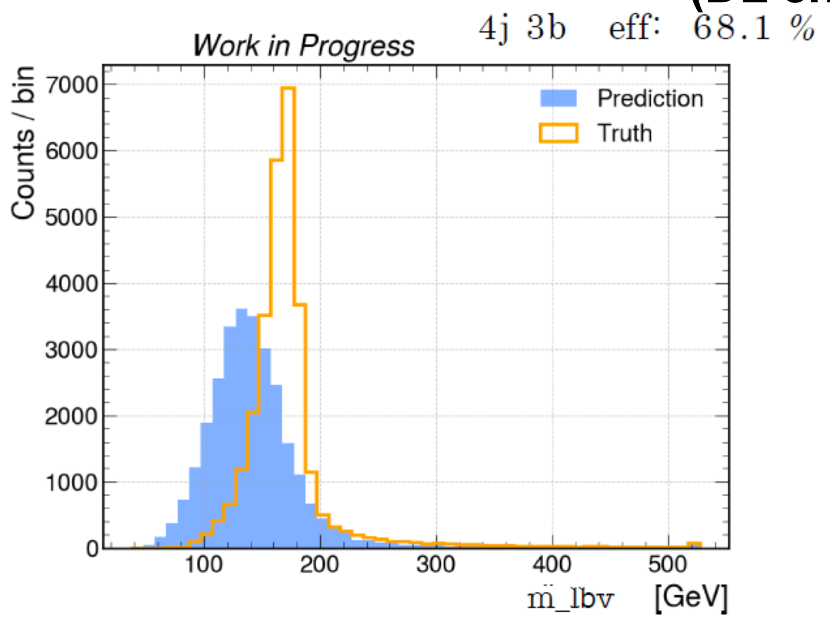
Jet-parton assignment with SPANet

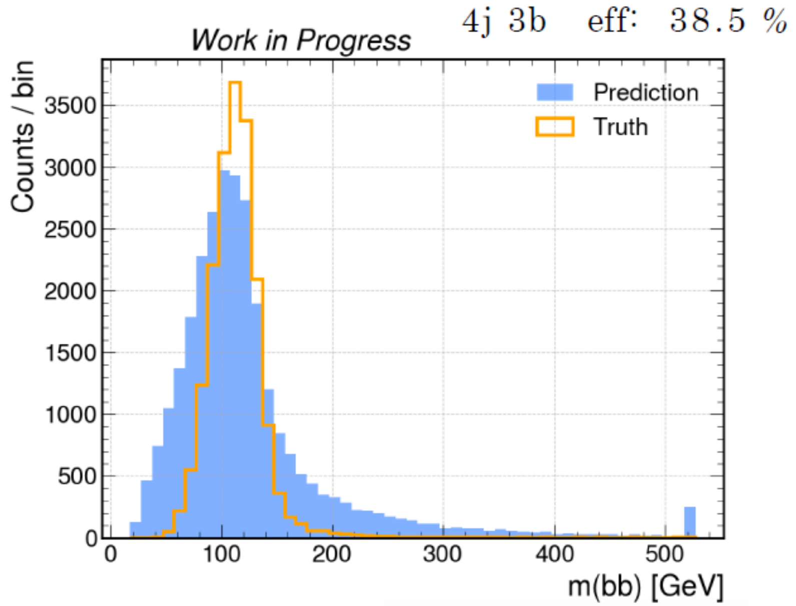
Inputs, assignment targets, regression (under study)

```
INPUTS:
  SEQUENTIAL:
    JetGood:
      pt: log_normalize
      eta: normalize
      mass: log_normalize
      phi: normalize
      btagRobustParTAK4B: none
  GLOBAL:
    PositiveLepton:
      pt: log_normalize
      eta: normalize
      phi: normalize
      mass: log_normalize
    NegativeLepton:
      pt: log_normalize
      eta: normalize
      phi: normalize
      mass: log_normalize
    Neutrinos:
      px: normalize
      py: normalize
      pz: normalize
    anti_Neutrinos:
      px: normalize
      py: normalize
      pz: normalize
```

```
EVENT:
  T1:
    - B
  T2:
    - B
  H:
    - B1
    - B2
PERMUTATIONS:
  H:
    - [ B1, B2 ]
```

```
REGRESSIONS:
  EVENT:
    - Neutrinos_px
    - Neutrinos_py
    - Neutrinos_pz
    - anti_Neutrinos_px
    - anti_Neutrinos_py
    - anti_Neutrinos_pz
```





From M. Torkian

Jet-parton assignment with SPANet

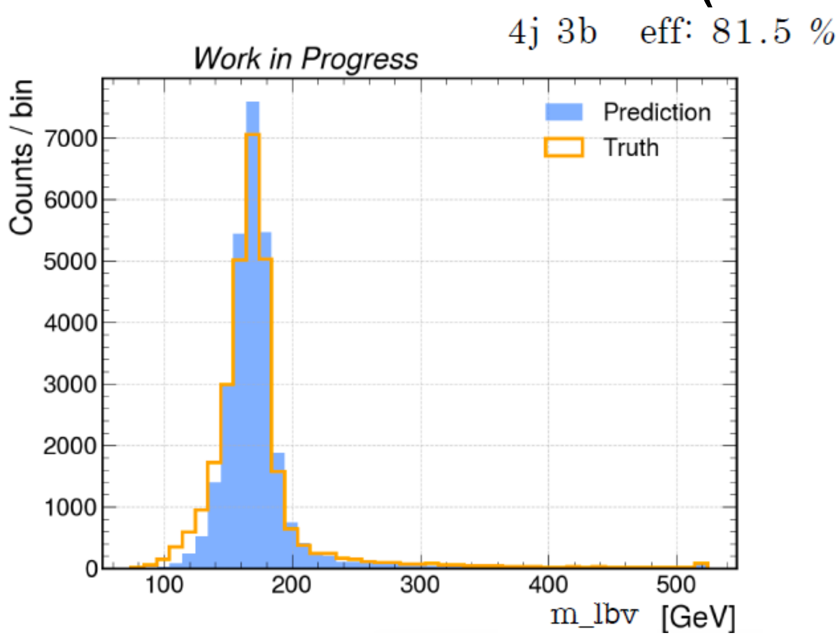
(DL channel)

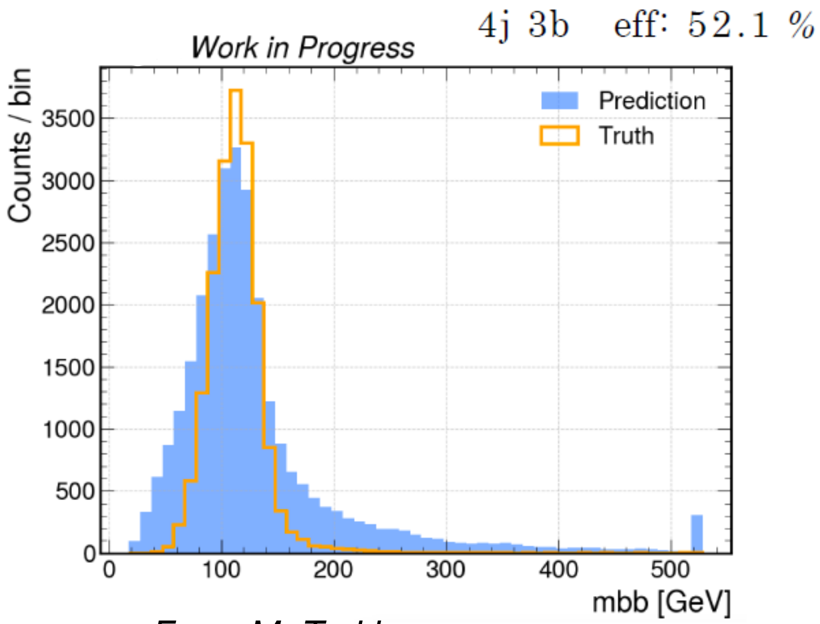
Inputs, assignment targets, regression (under study)

```
INPUTS:
  SEQUENTIAL:
    JetGood:
      pt: log_normalize
      eta: normalize
      mass: log_normalize
      phi: normalize
      btagRobustParTAK4B: none
  GLOBAL:
    PositiveLepton:
      pt: log_normalize
      eta: normalize
      phi: normalize
      mass: log_normalize
    NegativeLepton:
      pt: log_normalize
      eta: normalize
      phi: normalize
      mass: log_normalize
    Neutrinos:
      px: normalize
      py: normalize
      pz: normalize
    anti_Neutrinos:
      px: normalize
      py: normalize
      pz: normalize
```

```
EVENT:
  T1:
    - B
  T2:
    - B
  H:
    - B1
    - B2
PERMUTATIONS:
  H:
    - [ B1, B2 ]
```

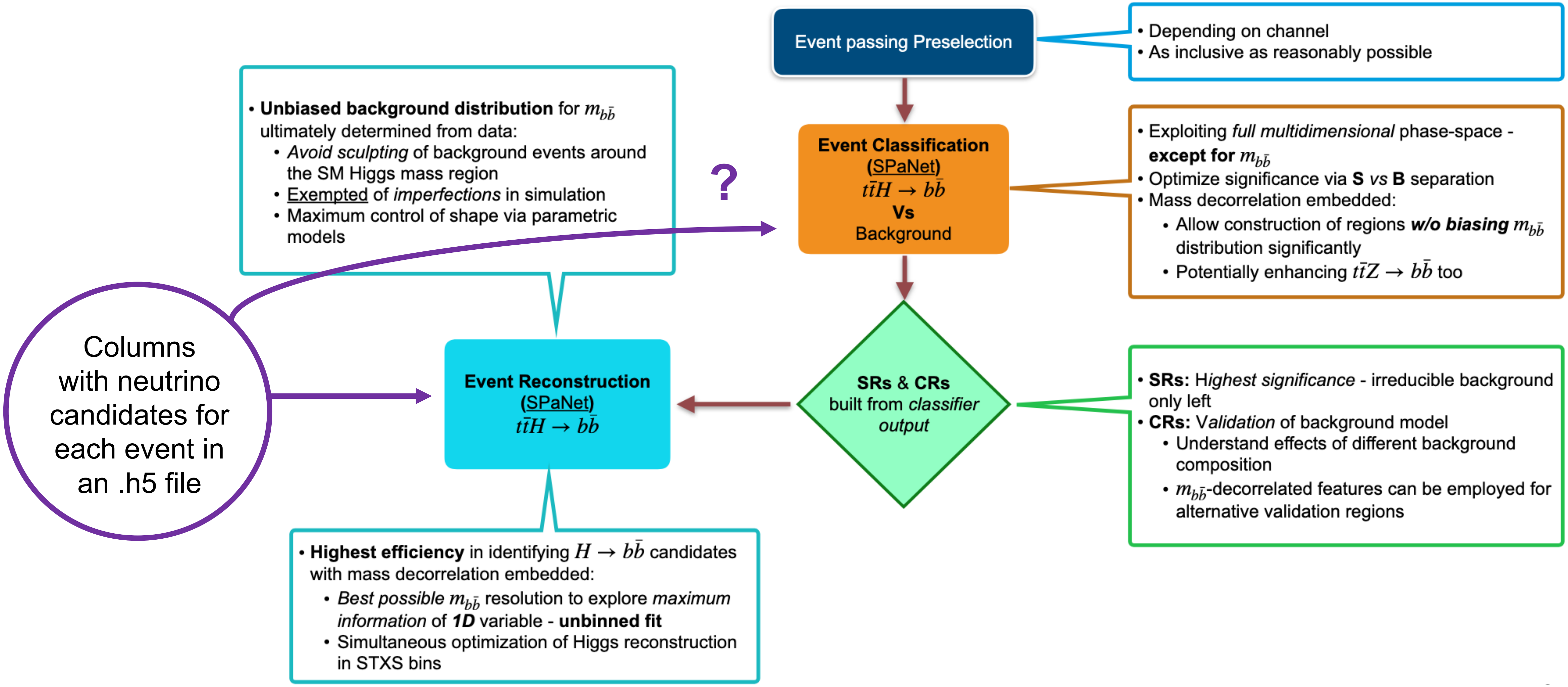
```
REGRESSIONS:
  EVENT:
    - Neutrinos_px
    - Neutrinos_py
    - Neutrinos_pz
    - anti_Neutrinos_px
    - anti_Neutrinos_py
    - anti_Neutrinos_pz
```





From M. Torkian

Neutrino flows for ttH(bb)



Neutrino flows: Conditional Neutrino Regression

The method

arXiv:2207.00664v7, arXiv:2307.02405

- Standard regression methods generally collapse the likelihood into a point estimate

- **ν-flows** is a ML approach to fully reconstruct the neutrinos produced in collisions from the missing transverse momentum and observed event kinematics
  - A **probabilistic approach that provides the likelihood over a range of viable solutions**
  - While many possible neutrino momenta values might be possible, they may not all be equally likely
  - Utilize conditional normalising flows to:
    - Leverage observed event features from the final-state
    - Inductive bias to restrict likelihood over the possible neutrino momentum values
  - **Final momenta are obtained by sampling from the learned conditional likelihood**

- In arXiv:2207.00664v7, direct benefits in a downstream task of jet association is also reported

Neutrino flows for ttH(bb)

Worfklow

Towards a unified framework

- We need both ν and ν^2-flows for SL and DL, respectively
- Aiming to integrate both frameworks via wrappers in a unified plugin structure (right)

To do list

- Starting with ν^2-flows flow for DL
  - Installing framework
  - Running baseline example (training + validation)
  - Produce our inputs with Run 3 CMS simulation and continue testing for our dedicated setup
- Continue with ν-flow for SL, to have a full working sub-projects set
- Integrate into one regression framework for ttH(bb)

```
ttHbb_analysis/
├── config/                  # YAML configs
│   └── model_paths.yaml
├── data/                    # Input/output files
│   ├── coffea_output
│   └── ...
├── models/                  # Model wrappers
│   ├── neutrino_solver.py   # Switch logic (SL vs DL)
│   ├── nu_flow_wrapper.py   # ν-flow interface
│   ├── nu2_flow_wrapper.py  # ν²-flow interface
│   └── __init__.py
├── subprojects/             # External dependencies
│   ├── neutrino_flows/      # Cloned ν-flow repo (SL)
│   └── nu2flows/            # Cloned ν²-flow repo (DL)
├── scripts/                 # Driver scripts
│   └── run_inference_coffea.py
├── utils/                   # Shared functions, e.g., feature prep
│   └── preprocessing.py
├── main.py                  # Optional
└── README.md
```

WE ARE HERE

Inputs for baseline setup

Input datasets (DL ttbar)

- Two sets of MC simulated events for the training and evaluation of neutrino regression models available in zenodo

  - **The nominal (Madgraph)** dataset contains **940605** events split across 4 files for training and one file containing **75496** events for final evaluation. Interfaced with Pythia 8 for PS and hadronization
  - The alternative (Pythia) dataset contains 1248315 events split across 9 files for training and one file containing 138128 events for final evaluation.

File Contents

Each HDF file contains the "delphes" table which holds multiple arrays and structured arrays.

The reconstructed information includes:

- MET: The missing transverse momentum of the event (stored using polar cords)
  - Keys: MET, phi
- leptons: The single reconstructed lepton in the event:
  - Keys: pt, eta, phi, energy, charge, type
- jets: A zero padded table of the leading 10 jets in the event
  - Keys: pt, eta, phi, energy, is_tagged
- njets: Numpy array holding the number of reconstructed jets in the event
- nbjets: Numpy array holding the number of b-tagged jets

The truth/generator level information includes:

- neutrinos: Truth level information of the single neutrino in the event
  - Keys: PDGID, pt, eta, phi, mass
- truth_quarks: Table of information for the two b-quarks in the event
  - Keys: PDGID, pt, eta, phi, mass
- truth_particles: Table of information of all truth level particles in the event
  - Keys: PDGID, pt, eta, phi, mass
- jets_indices: A numpy array containing the identity of each of the jets in the reconstructed jet array (above)
  - 0 = b-jet, 1 = anti b-jet, -1 = other

Simulation settings

- All events generated from simulated pp collisions at $\sqrt{s} = 13$ TeV, $\mathrm{m}_{\mathrm{top}} = 173$ GeV, $\alpha_s(m_Z)$
  - Generator versions outlined in zenodo
- Both samples are interfaced with Delphes for detector simulation with a parametrization that mimics the response of the **ATLAS** detector (FastSim)

[default]

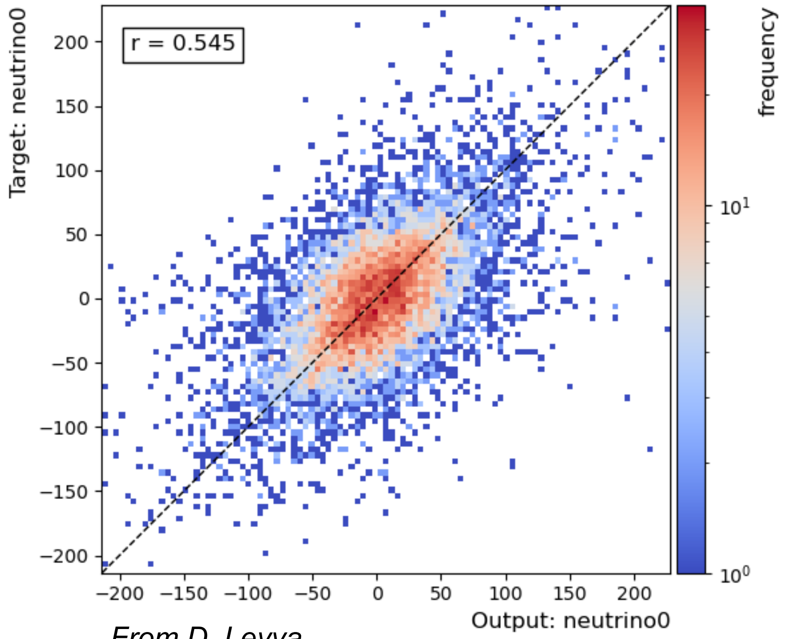
```
train_conf:
  data_dir:  ${paths.data_dir}
  met_kins: px,py
  lep_kins: px,py,pz,log_energy
  jet_kins: px,py,pz,log_energy
  nu_kins: px,py,pz
```

Training and validation

First training attempts with nu2flow setup (from Git linked in arXiv:2307.02405)

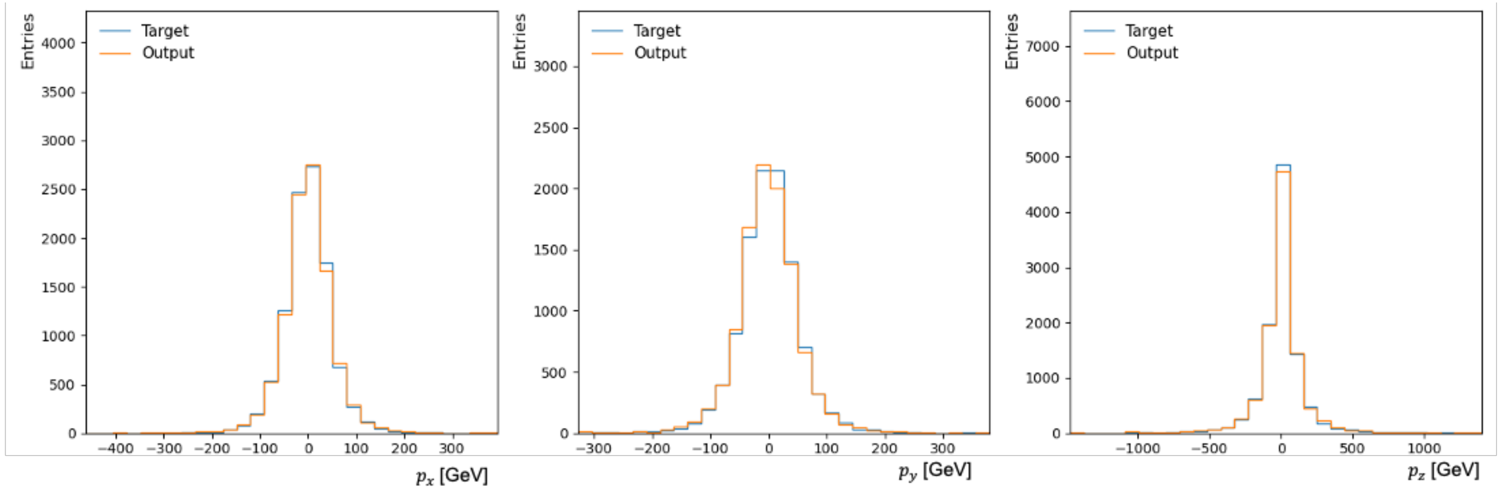
- Not trivial to setup/use: ***several incompatibilities found within the framework!***
  Most problematic: custom *mltools* module not available. Found in a different repository.
  Clean version will be pushed to Git - stay tuned!

Using FastSim ATLAS ttbar dataset in DL channel (DELPHES – from arXiv:2307.02405)



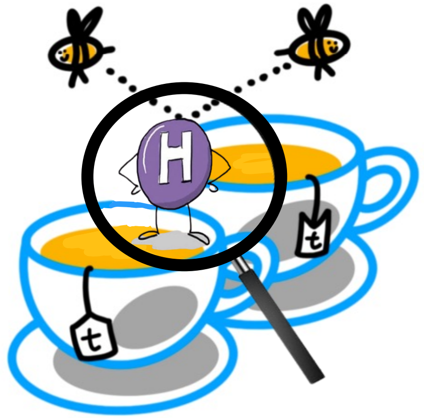
From D. Leyva

- Heatmaps of 2D histograms of truth vs. reconstructed ν momentum (p_x)
- Distributed along the diagonal → more accurate predictions
  - r : correlation coefficient between true and predicted momenta, ideally closer to $r = 1.0$
  - Best so far: $r = 0.5$ (converged after 20-30 epochs) → *trying to improve this!*



- Compatibility between truth and reconstructed seems promising, but..
  To be taken with 1kg of salt (Fast-Sim based)
- ➢ *Next: moving towards integration with CMS Run 3 simulation*

Summary

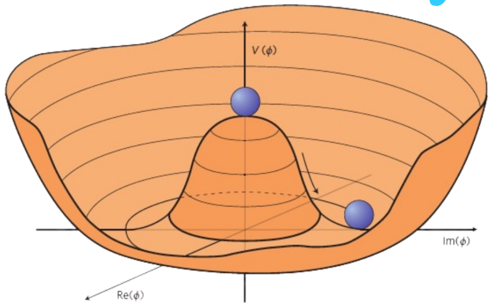


* sketch adapted from ttbb analysis' team

- Discussed ttH(bb) Run 3 analysis strategy
  - Key improvements → data-driven background + SPANet for jet-parton assignment

- Several improvements under the radar:
  - Analyse full 2022 and 2023 datasets and incorporate missing uncertainties
  - Production of Run 3 ttbb sample ongoing
  - **Top reconstruction enhancement with nu-squared flows**
  - b-jet energy regression to improve Higgs mass resolution
  - *Incorporate FH channel!*

More on our DESY-internal web:



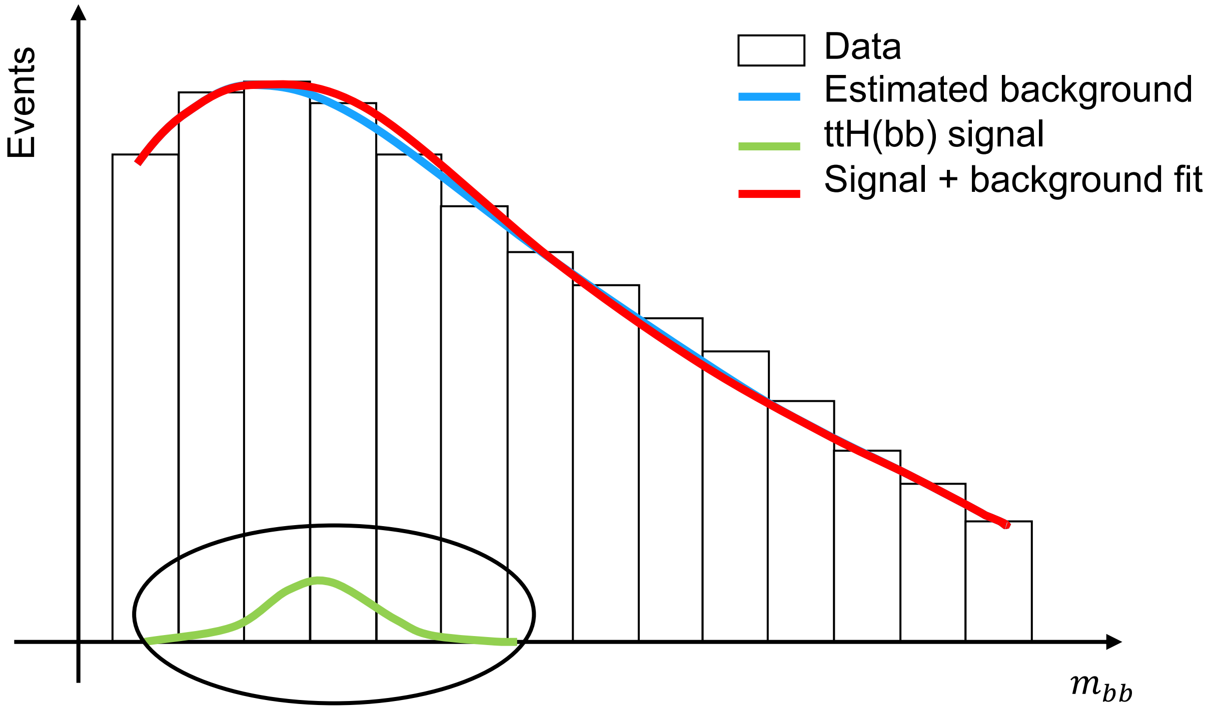
Thank you!

Backup

Strategy

Signal extraction

Signal extraction from unbinned likelihood analytic fit to m_{bb}



Signal description: crystal ball function (under study)

Background description: sigmoid function (turn-on) + polynomial OR exponential decay (under study)

- No dependence of background model to MC-related systematics
- Only systematics: choice of functional form → discrete profiling (arXiv:1408.6865)

The efforts

* up until March this year

Part of the ttHbb team during hackathon at DESY, March 7th

DESY.

+

RWTH AACHEN UNIVERSITY

Di-lepton channel

Di-lepton and
Single-lepton channels

Some bugs/errors/stones found along the way

nu2flows sub-project

- Package required ~20 GB of space
- Missing customized mltools module from repository
  - Intention of the author was to link to another repository, but the link is broken
  - Eventually found correct module within a specific branch of a different repository (here)
- Recommended python version is 3.9
  - Several incompatibilities found (should use 3.10), e.g.:

```
ImportError: Error loading 'src.datamodules.dilepton.H5DataModule':
TypeError("unsupported operand type(s) for |: 'torch._C._TensorMeta' and 'NoneType'")
```

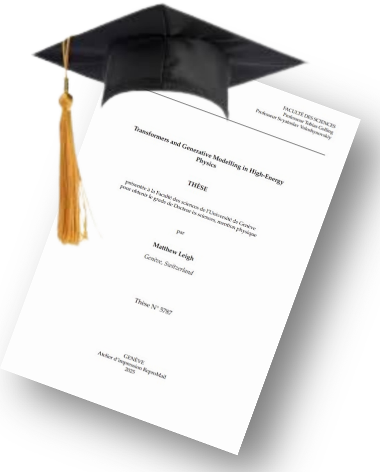
  - Easy to fix but painful to go one-by-one

```
RuntimeError: FlashAttention only supports Ampere GPUs or newer.
```

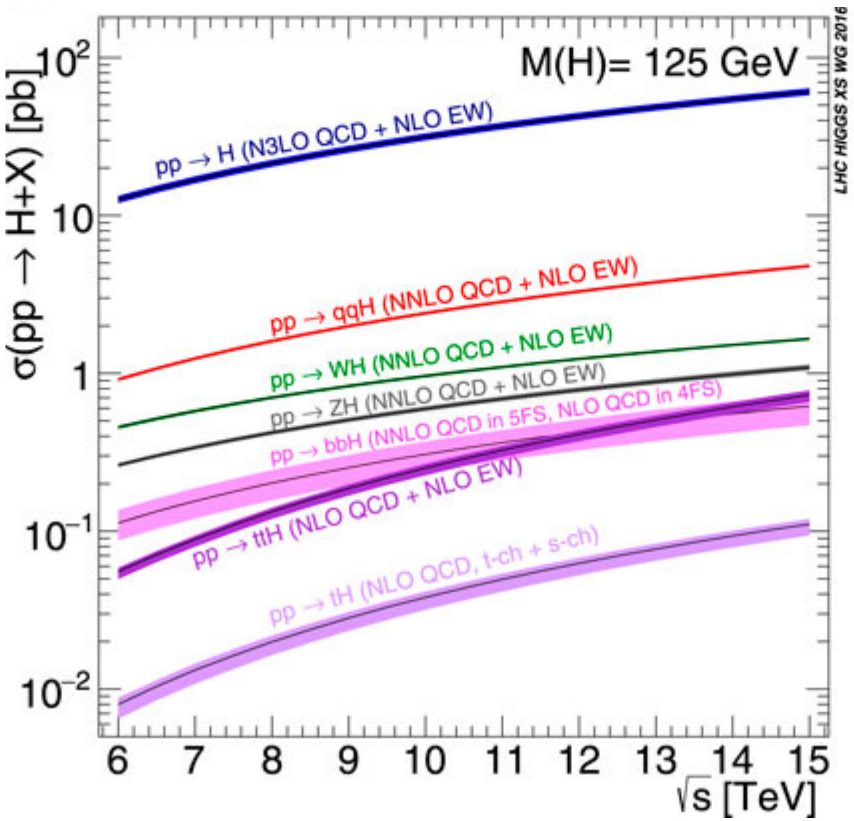
GPU might not be compatible with FlashAttention, which requires NVIDIA Ampere architecture GPUs (RTX 30xx series or newer)

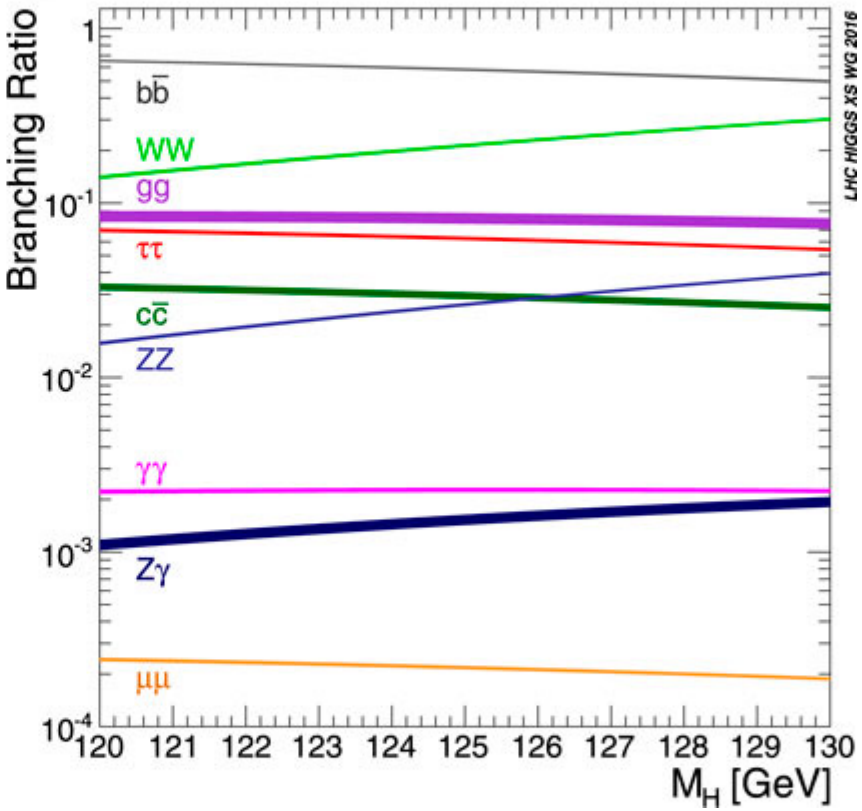
```
transformer_config:
  inpt_dim: 128
  outp_dim: 128
  #do_packed: True
  do_packed: False
```

While debugging/searching through Git, stumbled upon the author's PhD. Thesis "Transformers and Generative modelling in HEP" by M. Leigh (defended ~3 weeks ago)



SM Higgs production cross-section and Branching fraction





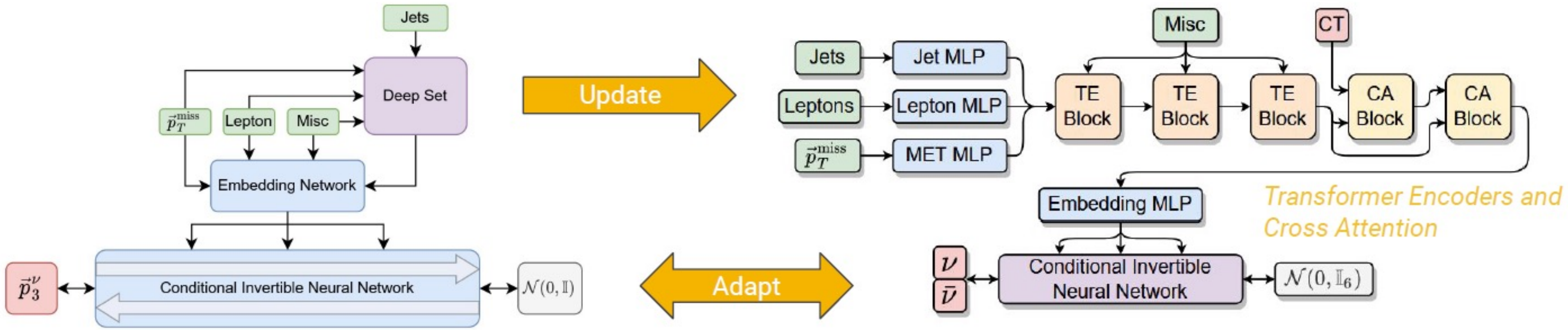
Front. Phys. 11:1230737

Neutrino flows

Comparing ν and ν^2 flows

As before don't want to enforce link between objects - network learns combinatorics

Multiplicity and permutation invariant in jets and leptons
Can easily add any other objects e.g. photons and taus



Normalizing flow remains constant
Scale dimension with number of neutrinos

10

Jet-parton assignment with SPANet

Inputs, assignment targets, regression (under study)

```
INPUTS:
  SEQUENTIAL:
    JetGood:
      pt: log_normalize
      eta: normalize
      mass: log_normalize
      phi: normalize
      btagRobustParTAK4B: none
  GLOBAL:
    PositiveLepton:
      pt: log_normalize
      eta: normalize
      phi: normalize
      mass: log_normalize
    NegativeLepton:
      pt: log_normalize
      eta: normalize
      phi: normalize
      mass: log_normalize
    Neutrinos:
      px: normalize
      py: normalize
      pz: normalize
    anti_Neutrinos:
      px: normalize
      py: normalize
      pz: normalize
```

```
EVENT:
  T1:
    - B
  T2:
    - B
  H:
    - B1
    - B2
PERMUTATIONS:
  H:
    - [ B1, B2 ]
```

```
REGRESSIONS:
  EVENT:
    - Neutrinos_px
    - Neutrinos_py
    - Neutrinos_pz
    - anti_Neutrinos_px
    - anti_Neutrinos_py
    - anti_Neutrinos_pz
```

- Assessing performance through assignment efficiency (eff)

$$eff = \frac{Events\ correctly\ reconstructed}{Number\ of\ possible\ reconstructions}$$

- How distributions are defined:

Prediction: SPANet assignment

Truth: True assignments between the LHE/GEN level particles and the RECO jets

Number of events	
Training	458558
Validation	98261
Test	98266

SPANet

Baseline methods

- Weintroduce a general architecture for jet-parton assignment named SPA-NET: an attention-based neural network
- The high level structure of SPA-NET, visualized in Figure 2, consists of four distinct components

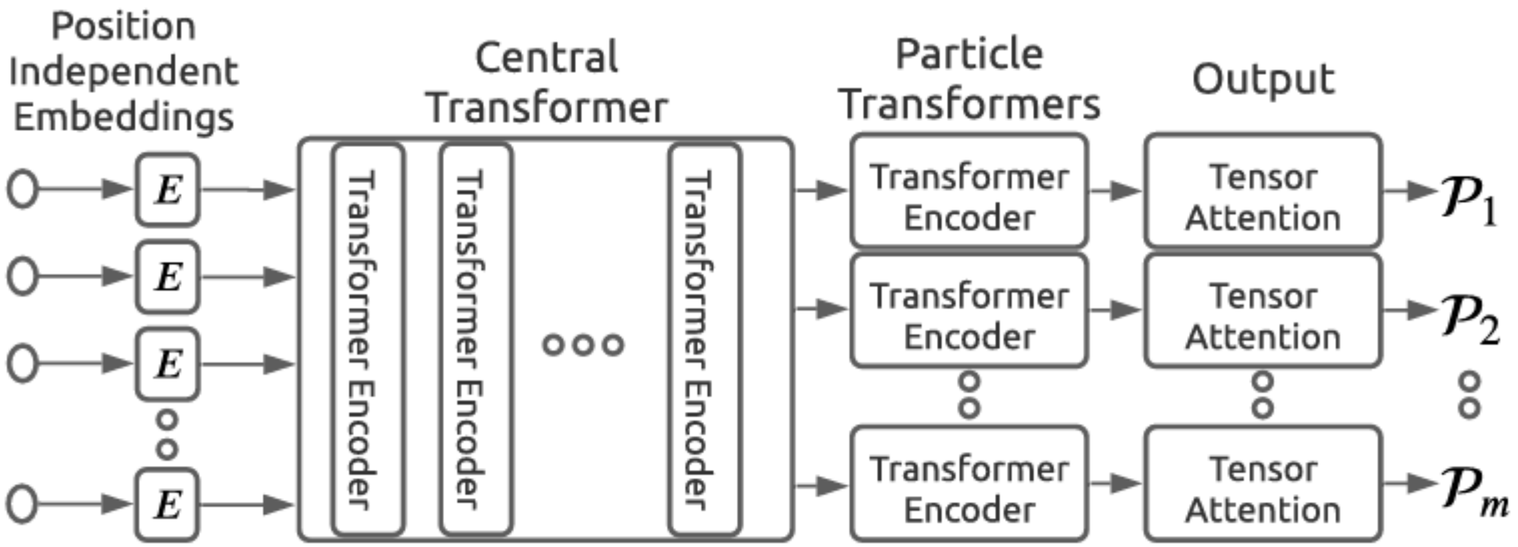


Figure 2: A visualization of the high level structure of SPA-NET.

- SPA-NET improves run-time performance over baseline permutation methods by avoiding having to construct all valid assignment permutations
- Instead, we first partition the jet-parton assignment problem into sub-problems for each resonance particle, as determined by the event Feynman diagram's tree-structure
- Then we proceed in two main steps:

  (1) we solve the jetparton assignment sub-problems within each of these partitions using a novel form of attention which we call Symmetric Tensor Attention

  (2) we combine all the sub-problem solutions into a final jet-parton assignment (Combined Symmetric Loss). This two-step approach also allows us to naturally handle both symmetries described

Contact

Deutsches Elektronen-
Synchrotron DESY

www.desy.de

Daina Leyva Pernia
CMS Higgs Group
daina.leyva.pernia@desy.de
daina.leyva.pernia@cern.de