ML4Jets2022 @ Rutgers Univ
Nov 3, 2022

VBF vs. GGF Higgs with Full-Event Deep Learning — Towards a Decay-Agnostic Tagger

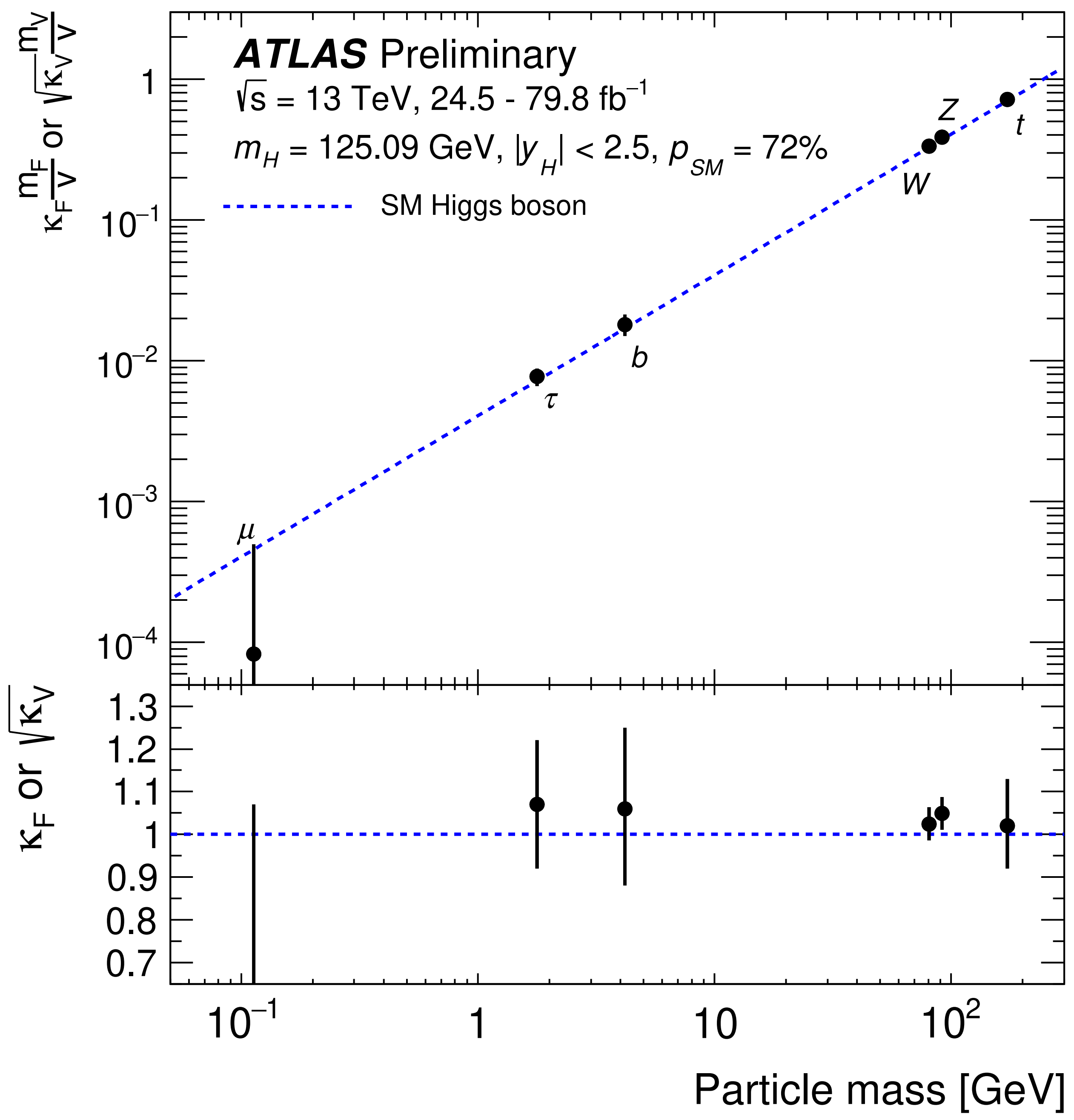
Cheng-Wei Chiang
National Taiwan University
National Center for Theoretical Sciences

Based on: CWC, David Shih and Shang-Fu Wei, 2209.05518 [hep-ph]

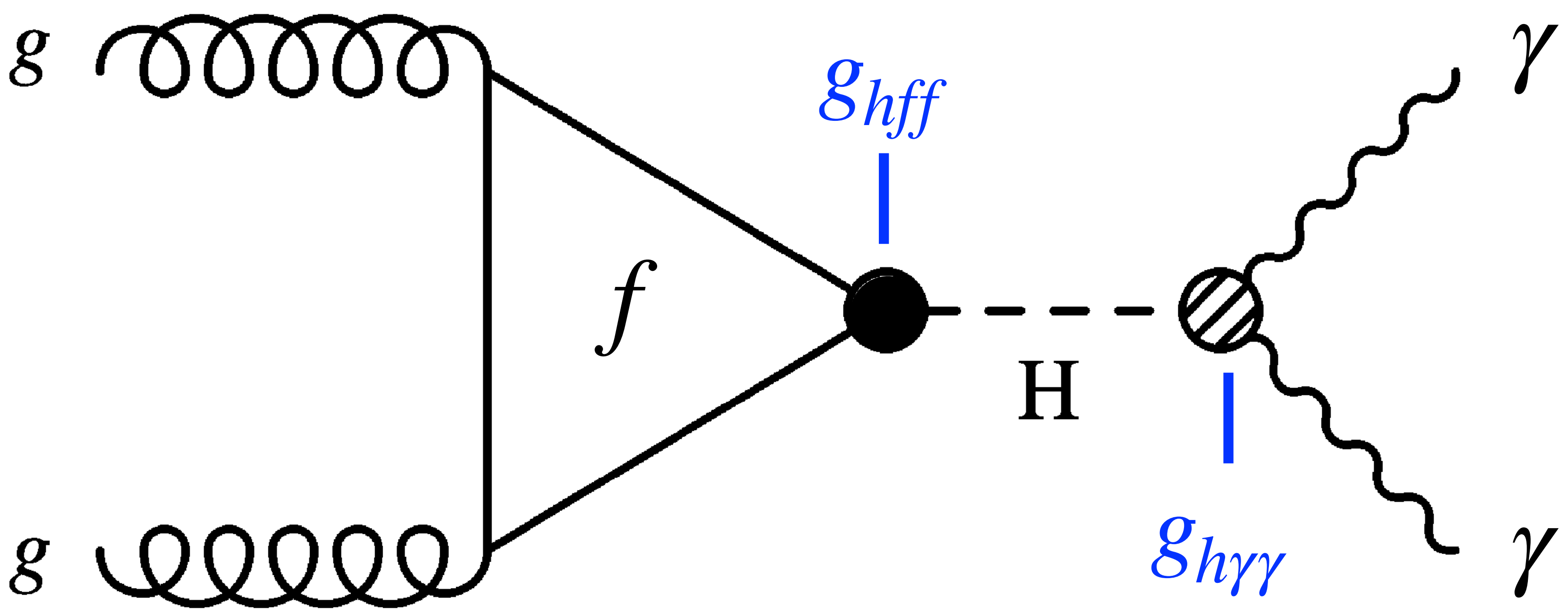
Higgs Physics Program

- After the Higgs boson discovery, an urgent physics program is to determine all the Higgs couplings precisely.
  ➠ look for any significant deviations
  ➠ hints of new physics

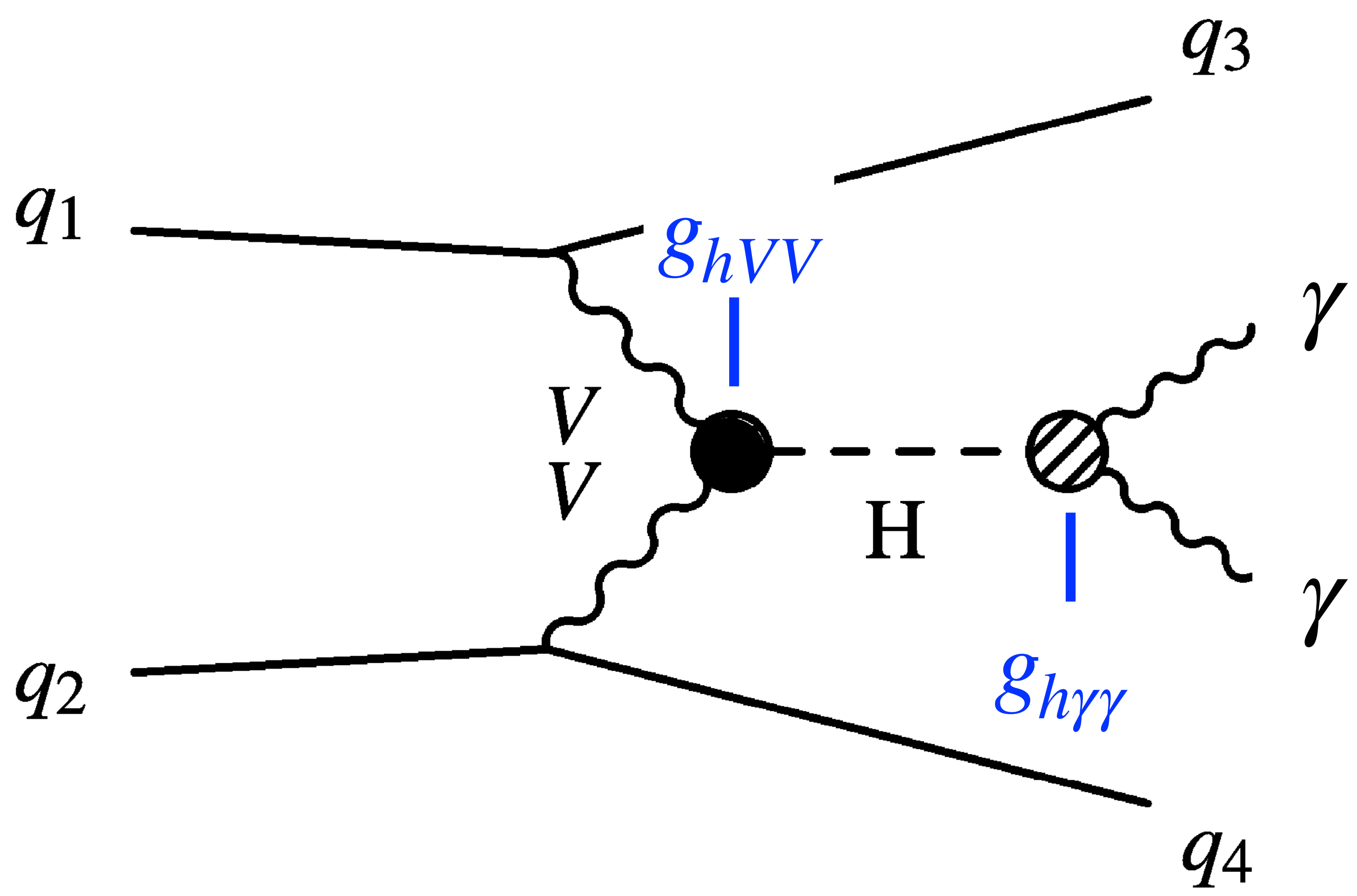
- This requires the ability to discriminate the two dominant production channels (others being even smaller).
  ➠ pinpoint the sources of deviations (production or decay part or both)



ATLAS 2019



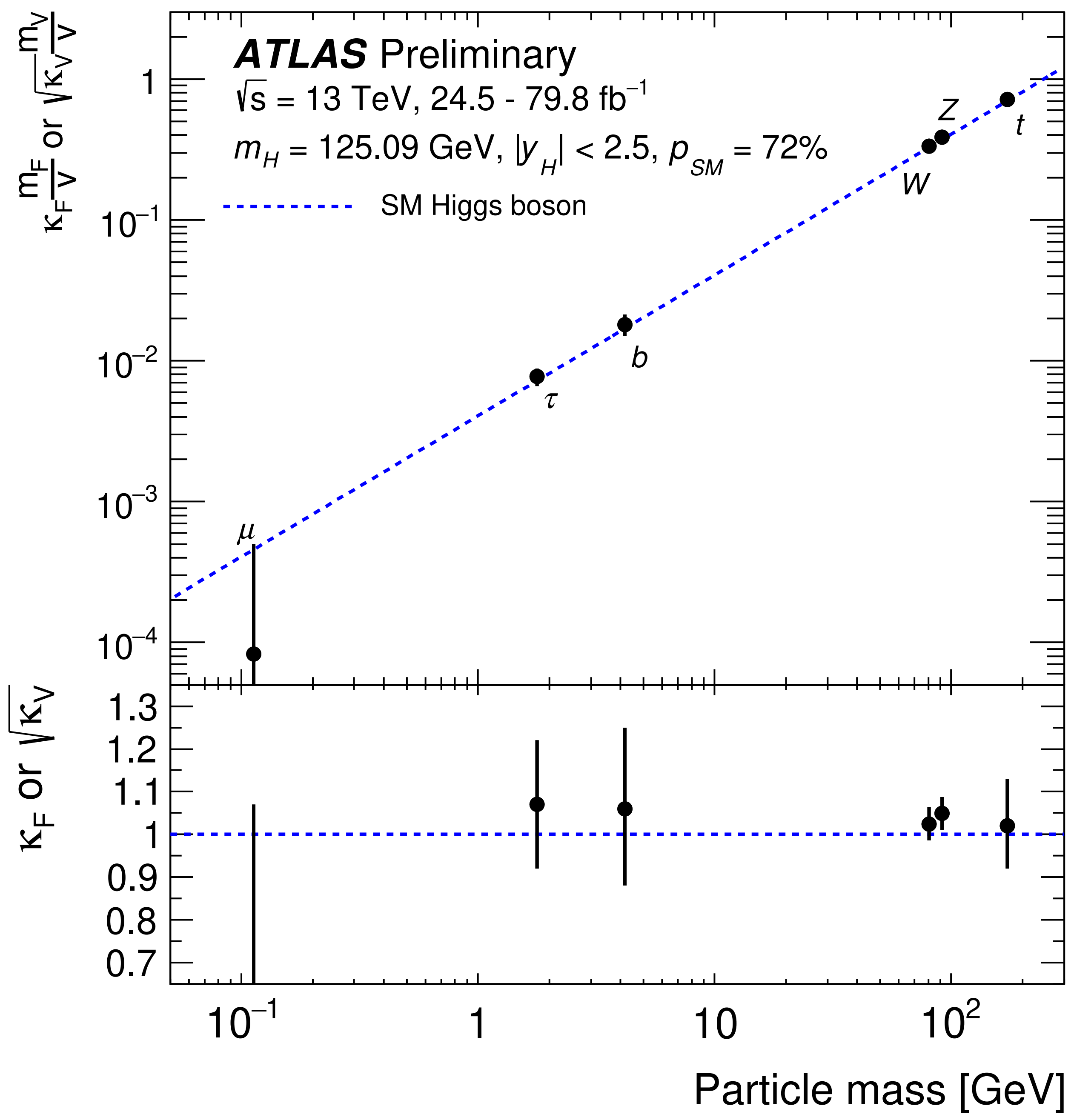
(a) ggF production



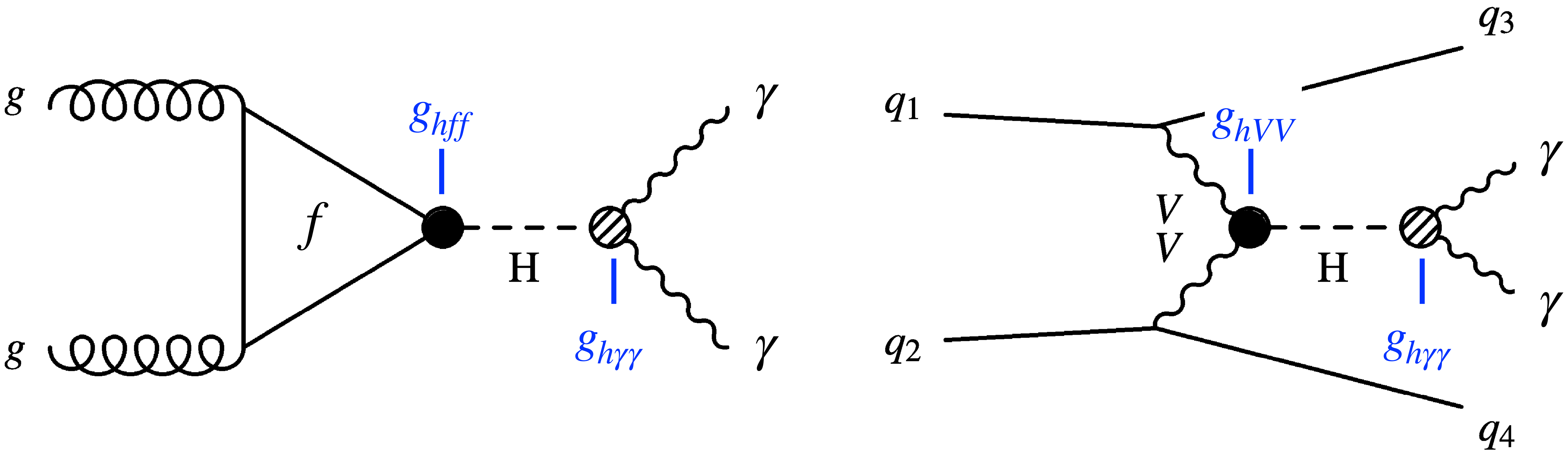
(b) VBF production

Higgs Physics Program

- **VBF** or the g_{hVV} coupling is essential for studying the role of the Higgs boson in the EWSB.
- Questions:
  - For any Higgs channel, how can we *efficiently* and *correctly* discriminate/label the two mechanisms?
  - Can it be independent of how the Higgs decays?



ATLAS 2019



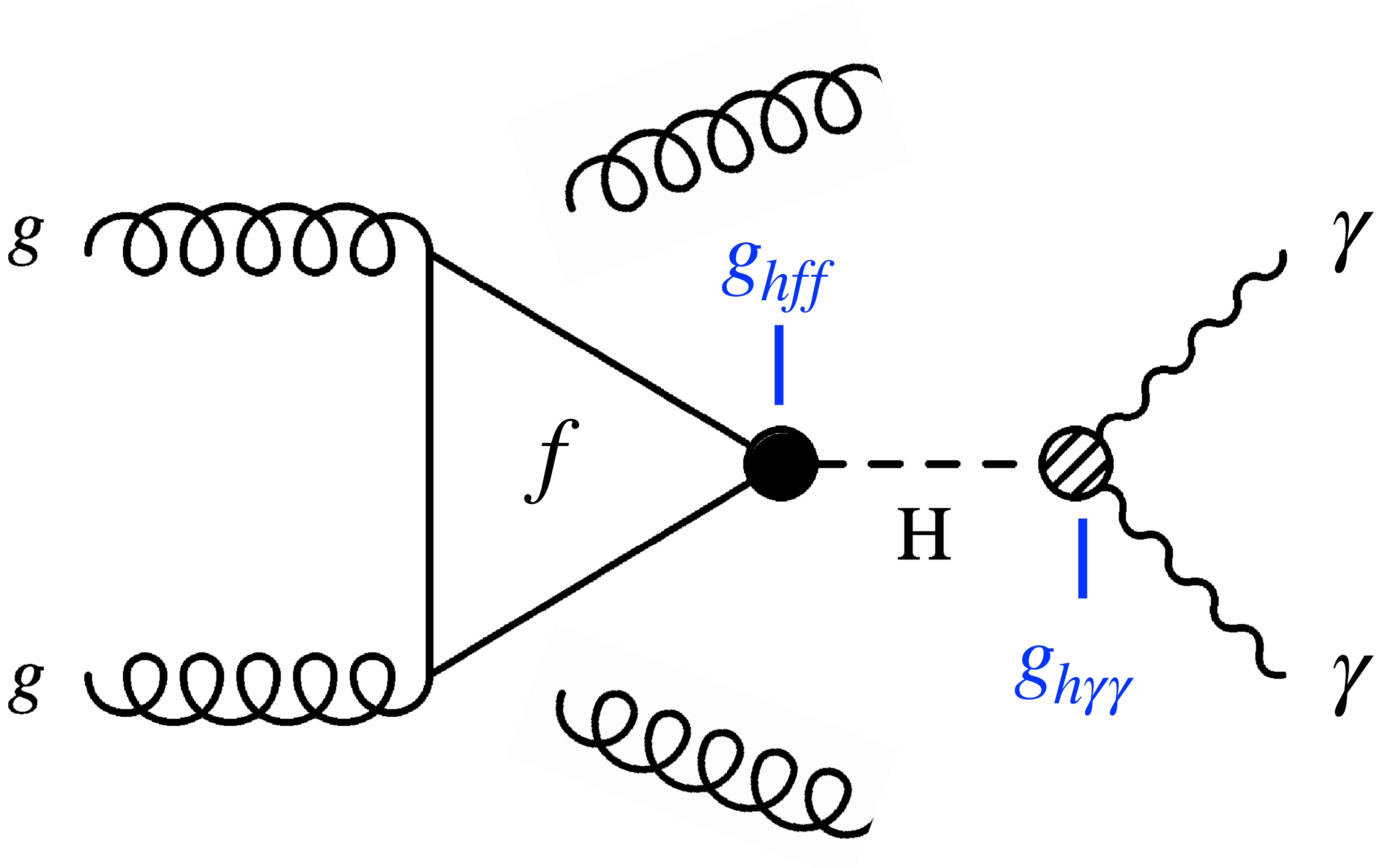
(a) ggF production

(b) VBF production

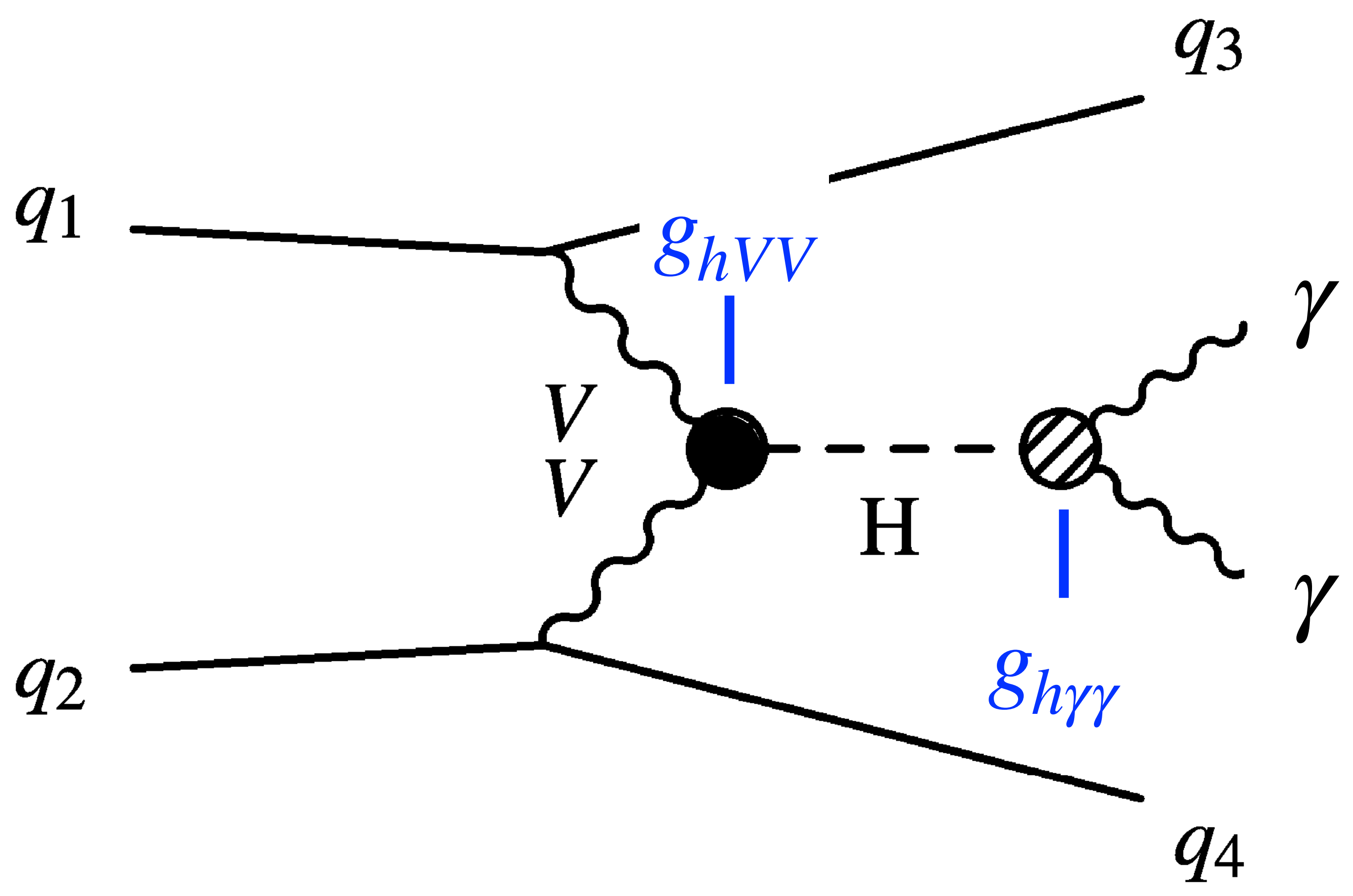
Higgs Physics Program

- VBF events come with two **forward quark-initiated jets** from the hard process, while GGF jets tend to be **gluon-initiated ISR**.
  ➠ different jet distributions

- Since the Higgs is a **color singlet scalar**, the Higgs decay should be *factorizable* from the VBF or GGF initial state jets, especially for electroweak final states.
  ➠ Higgs decay independent



(a) ggF production



(b) VBF production

Previous Studies

- Machine learning methods had been previously applied to the VBF vs GGF classification problem, mostly using *high-level* observables.
  - **Boosted decision trees** (**BDTs**) trained on *high-level* physics variables (e.g., invariant jet mass, rapidity difference of the leading jets, various jet shape variables, etc) were studied *separately* (using *different* cuts, etc) for $H \to \gamma\gamma$ and $H \to WW^*$ final states. Chan, Cheung, Chung, and Hsu 2017
  - The *multiclass* classification of multiple Higgs production modes (including VBF and GGF), with **BDTs** trained on *high-level* features and a specialized **two-stream CNN** on event images of *low-level* inputs, was studied specifically for the boosted $H \to bb$ regime. Chung, Hsu and Nachman 2020
  - Experimental studies have also used BDTs, DNNs or RNNs on a variety of Higgs decay modes to discriminate VBF from GGF events, taking the *high-level* features as input. several refs of ATLAS and CMS 2020—2022

Our Classifiers

- We construct a BDT trained on *high-level features* defined from the leading two jets and the Higgs decay products (the latter to be taken away eventually) as the **baseline** characterizing the prior art.
- Beyond it, we consider the following methods:
    - Train a **jet-level CNN** to distinguish the leading two jets (quark vs gluon), and add the jet-CNN scores to the inputs of the BDT for improvement.
    - Train an **event-level CNN** to distinguish full VBF vs GGF events, using full-event images out of the energy deposits of all the reconstructed particles in the event.
    - Train an event-level network based on the **self-attention** model, by converting the input event into a sequence that directly records the detector-level information.

Lin, Feng, dos Santos, Yu, Xiang, Zhou and Bengio 2017
Vaswani, Shazeer, Parmar, Uszkoreit, Jones, Gomez, Kaiser, and Polosukhin 2017

Event Generation

- We generate events with a Higgs plus **up to three jets**, with the Higgs decaying into **a pair of photons**, for 14-TeV LHC.

parton-level events

MG5aMC@NLO2.7.3
PDFs: CT10
jet matching: MLM with xqcut = 30 GeV and qcut = 45 GeV.

parton showering / hadronization

Pythia8.245

detector simulation

Delphes3.4.2 with default ATLAS card FastJet3.3.2 for jet clustering with the anti-kT algorithm with $R = 0.4$

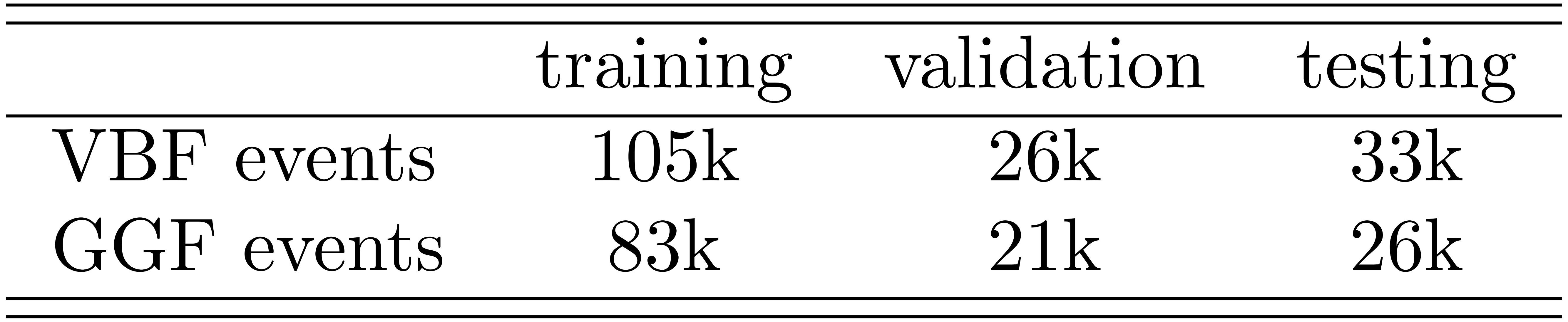
- tree-level MG5 for VBF
- effective vertex generated by FeynRules2.3.3 for GGF

- local dipole recoil toggled on for VBF events to better model the emission of additional jets

- jets required to have $p_T > 25$ GeV.
- using EFlow objects instead of the default Tower objects as inputs of the jet cluster module

VBF Pre-Selection

- Consider **VBF** events as the **signal** and **GGF** events as the **background**.
- Use the **pre-selection cuts**: $N_\gamma \geq 2$, $120 \leq M_{\gamma\gamma} \leq 130$ GeV, $N_j \geq 2$, and $\Delta\eta_{jj} \geq 2$, with the jets having $p_T > 30$ GeV.
- Generate 500k events each for the VBF and GGF samples.
  ➠ after the pre-selection, left with 164k events for VBF and 131k for GGF (jet samples being twice the numbers)
  ➠ the training scheme listed as follows:



	training	validation	testing
VBF events	105k	26k	33k
GGF events	83k	21k	26k

Models

- Consider the following types of NNs:
  - **BDT**-type (using `XGBoost1.5.0`) ➠ taking mostly kinematic variables as inputs
  - **CNN**-type (`TensorFlow2.0.0` with `Keras` API) ➠ taking jet/full-event images as inputs
  - **Self-Attention** (`TensorFlow2.5.0` with `Keras` API) ➠ taking particle 4-vectors as inputs

BDT hyperparameters

Max depth	3
Learning rate	0.1
Objective	binary logistic
Early stop	10 epochs
Evaluation metric	binary logistic

NN hyperparameters

Optimizer	Adam
Loss function	categorical cross entropy
Early stopping	20 epochs – CNN 50 epochs – self-attention
Batch size	1024

BDT Input Features

- Baseline high-level features (kinematic and jet shape variables) used in BDTs:

Higgs decay product-related

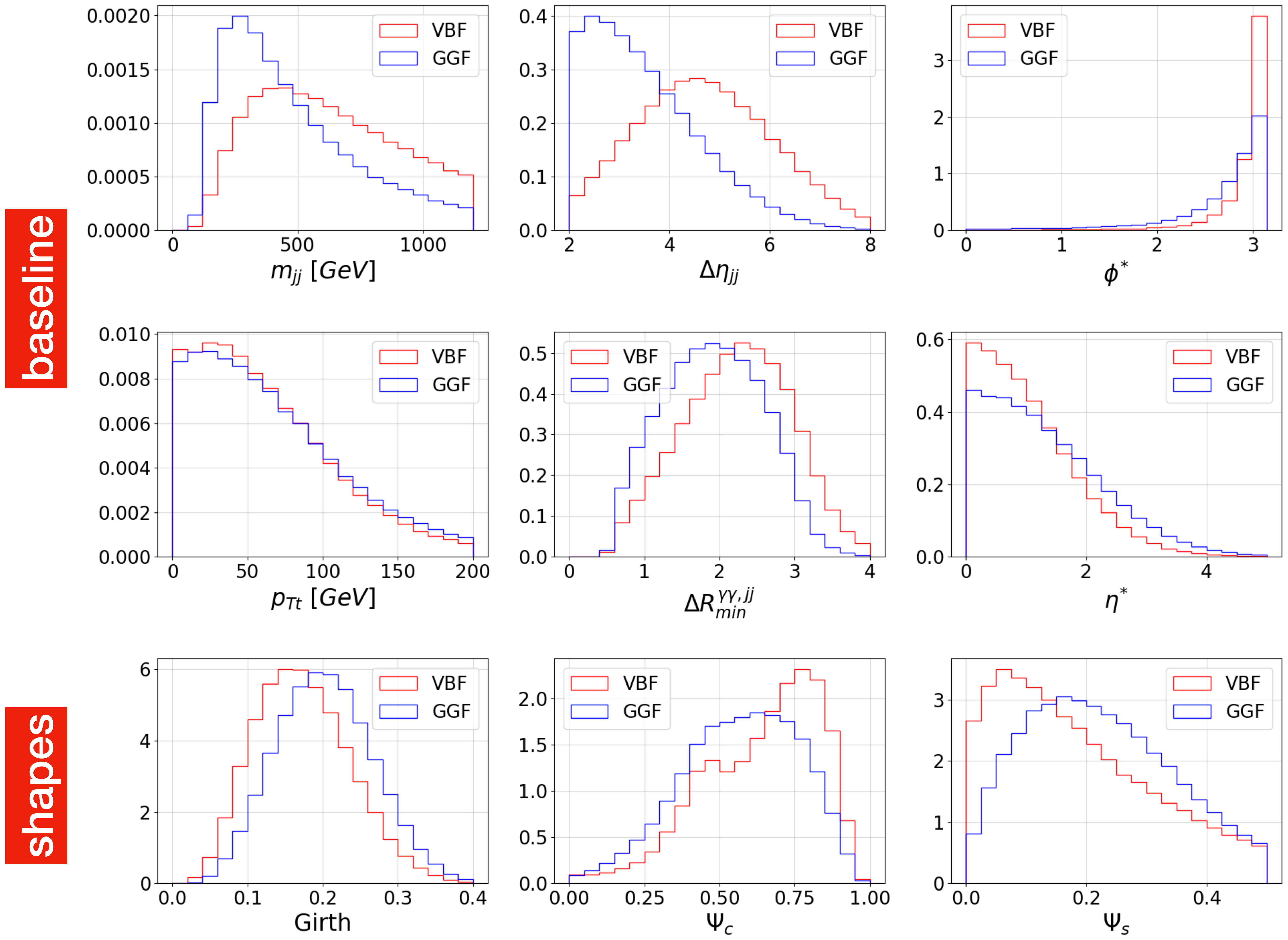
baseline ATLAS 2018	1. m_{jj}, the invariant mass of j_1 and j_2
	2. $\Delta\eta_{jj}$, the absolute difference of the pseudo-rapidities of j_1 and j_2
	3. ϕ^*, defined by the ϕ-difference between the leading di-photon and di-jet
	4. $p_{Tt}^{\gamma\gamma}$, defined by $\left\|(\mathbf{p}_T^{\gamma_1}+\mathbf{p}_T^{\gamma_2})\times\hat{t}\right\|$, where $\hat{t}=(\mathbf{p}_T^{\gamma_1}-\mathbf{p}_T^{\gamma_2})/\|\mathbf{p}_T^{\gamma_1}-\mathbf{p}_T^{\gamma_2}\|$
	5. $\Delta R_{\gamma j}^{\min}$, defined by the minimum η-ϕ separation between γ_1/γ_2 and j_1/j_2
	6. η^*, defined by $\|\eta_{\gamma_1\gamma_2}-(\eta_{j_1}+\eta_{j_2})/2\|$, where $\eta_{\gamma_1\gamma_2}$ is the pseudo-rapidity of the leading di-photon
shape Shelton 2013	7. the girth summed over the two leading jets $\sum_{j=1}^{2} g_j=\sum_{j=1}^{2}\sum_{i\in J^j}^{N} p_{T,i}^{j} r_i^j/p_T^j$
	8. the central integrated jet shape $\Psi_c=\sum_{j=1}^{2}\sum_{i\in J^j}^{N} p_{T,i}^{j}(0<r_i^j<0.1)/(2p_T^j)$
	9. the sided integrated jet shape $\Psi_s=\sum_{j=1}^{2}\sum_{i\in J^j}^{N} p_{T,i}^{j}(0.1<r_i^j<0.2)/(2p_T^j)$

constituent label

distance between the constituent and the jet axis

Distributions of BDT Input Variables

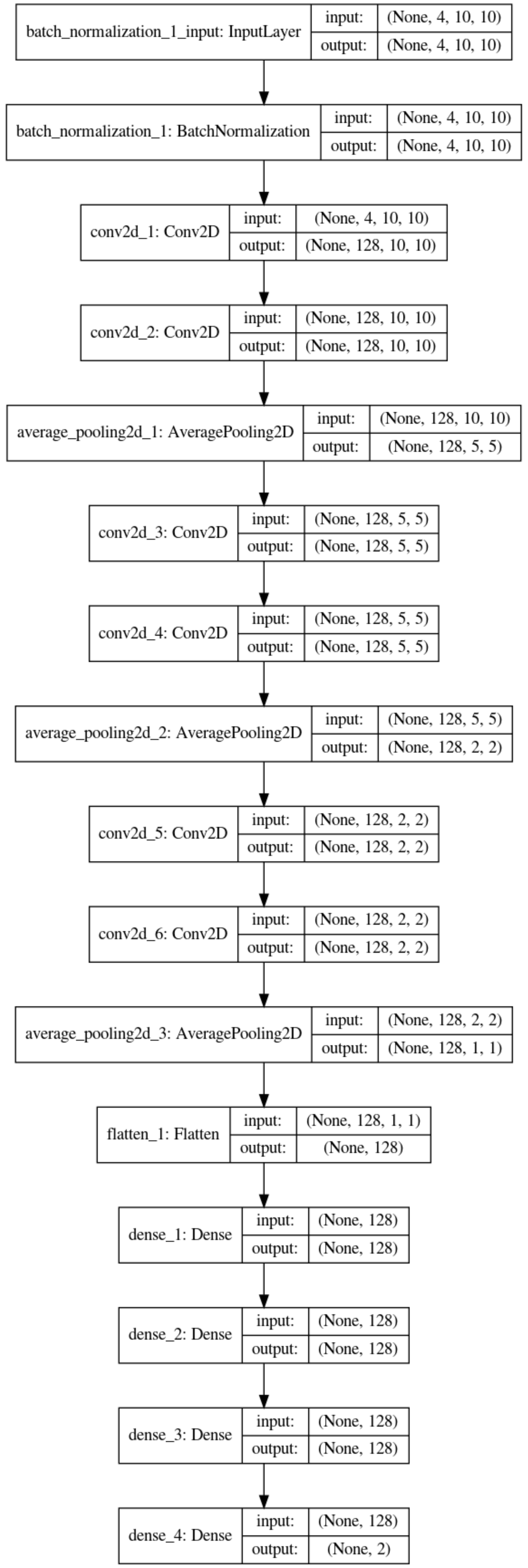
- All histograms are normalized.
- GGF events tend to have more jet activities (*gluon-initiated from ISR*) than VBF events (*forward quark-initiated from the hard process*) — an important feature for CNN.
- **BDT: baseline**: using baseline variables only
- **BDT: baseline + shape**: using baseline and shape variables together
- **BDT: baseline + jet-CNN**: using baseline variables and jet-CNN (see next slide) scores



Jet-CNN

- It is trained on jet images formed out of the *leading two jets* from the VBF and GGF events.
- Input jet image manipulation:
  - **Pre-processing**: standard centralization, rotation, and flipping.
  - **Pixelation**: from detector responses into 10×10 pixels.
  - **4 channels**: Tower E_T, Tower hits, Track E_T, and Track hits.

- Our jet-CNN takes a jet image as its input and outputs a score ranging from 0 (GGF-jet) to 1 (VBF-jet).
- The scores of leading/subleading jets can be useful features for subsequent event-by-event classification.



batch_normalization_1_input: InputLayer — input: (None, 4, 10, 10) — output: (None, 4, 10, 10)

batch_normalization_1: BatchNormalization — input: (None, 4, 10, 10) — output: (None, 4, 10, 10)

conv2d_1: Conv2D — input: (None, 4, 10, 10) — output: (None, 128, 10, 10)

conv2d_2: Conv2D — input: (None, 128, 10, 10) — output: (None, 128, 10, 10)

average_pooling2d_1: AveragePooling2D — input: (None, 128, 10, 10) — output: (None, 128, 5, 5)

conv2d_3: Conv2D — input: (None, 128, 5, 5) — output: (None, 128, 5, 5)

conv2d_4: Conv2D — input: (None, 128, 5, 5) — output: (None, 128, 5, 5)

average_pooling2d_2: AveragePooling2D — input: (None, 128, 5, 5) — output: (None, 128, 2, 2)

conv2d_5: Conv2D — input: (None, 128, 2, 2) — output: (None, 128, 2, 2)

conv2d_6: Conv2D — input: (None, 128, 2, 2) — output: (None, 128, 2, 2)

average_pooling2d_3: AveragePooling2D — input: (None, 128, 2, 2) — output: (None, 128, 1, 1)

flatten_1: Flatten — input: (None, 128, 1, 1) — output: (None, 128)

dense_1: Dense — input: (None, 128) — output: (None, 128)

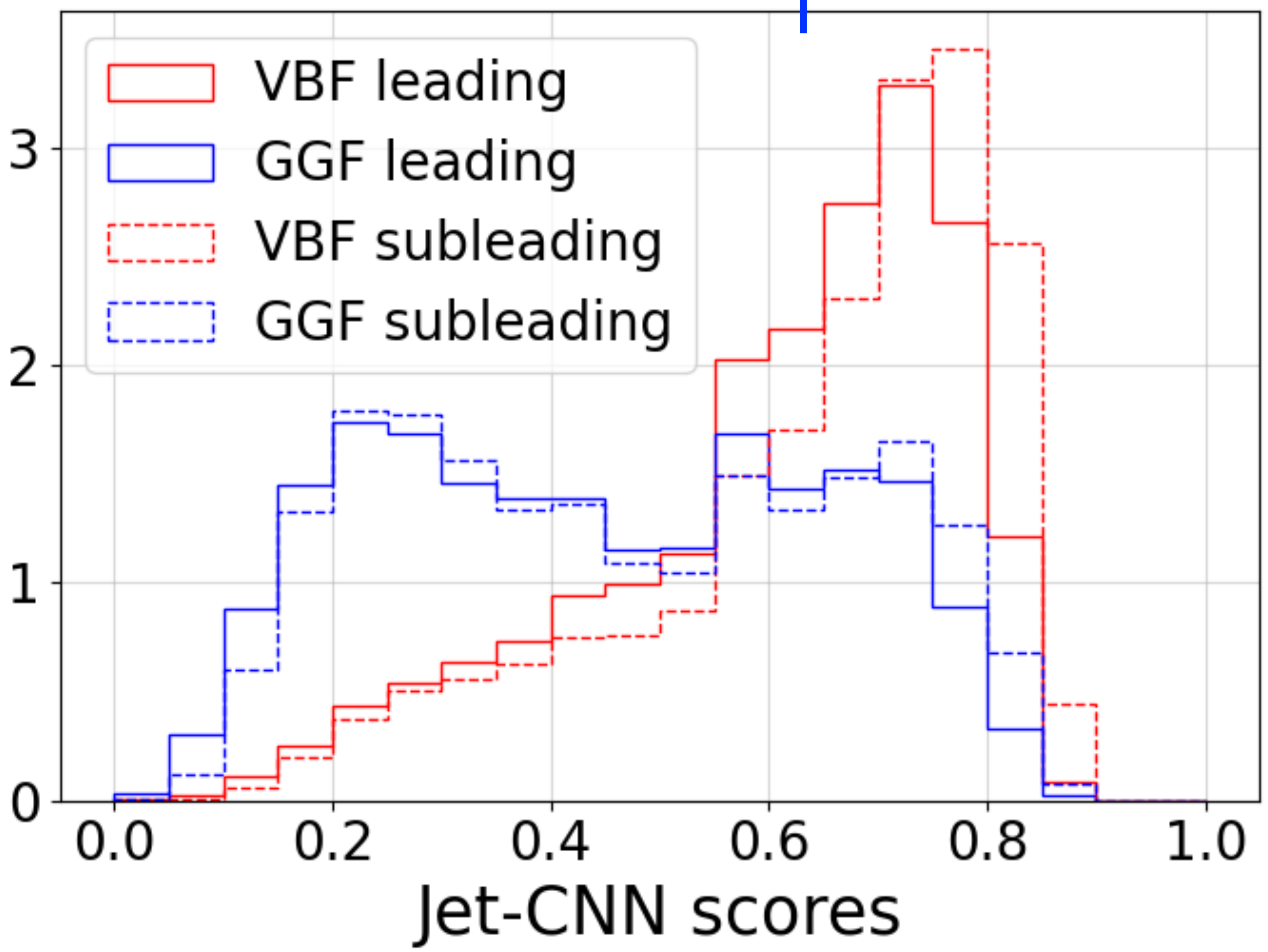
dense_2: Dense — input: (None, 128) — output: (None, 128)

dense_3: Dense — input: (None, 128) — output: (None, 128)

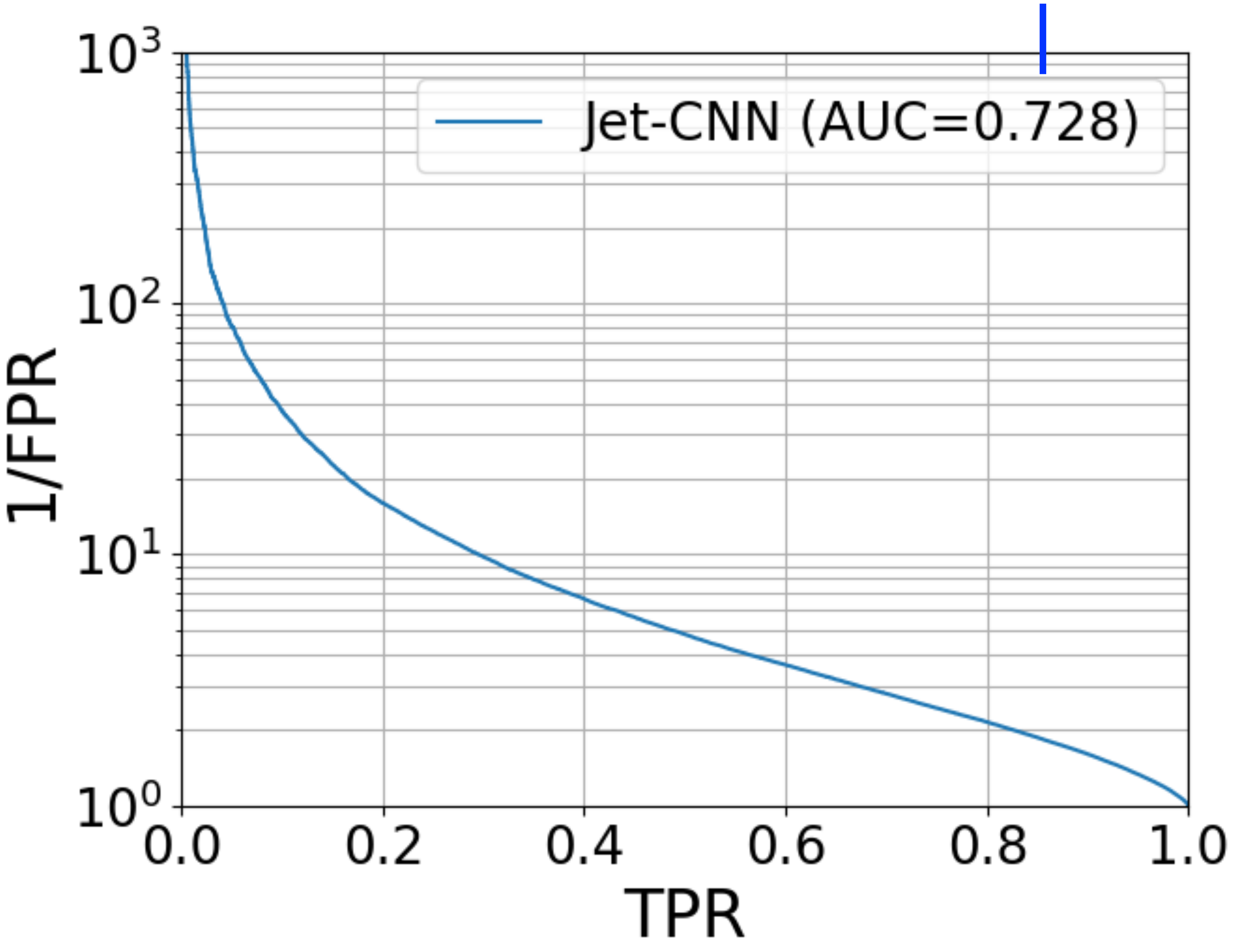
dense_4: Dense — input: (None, 128) — output: (None, 2)

Performance of Jet-CNN

all distributions being normalized



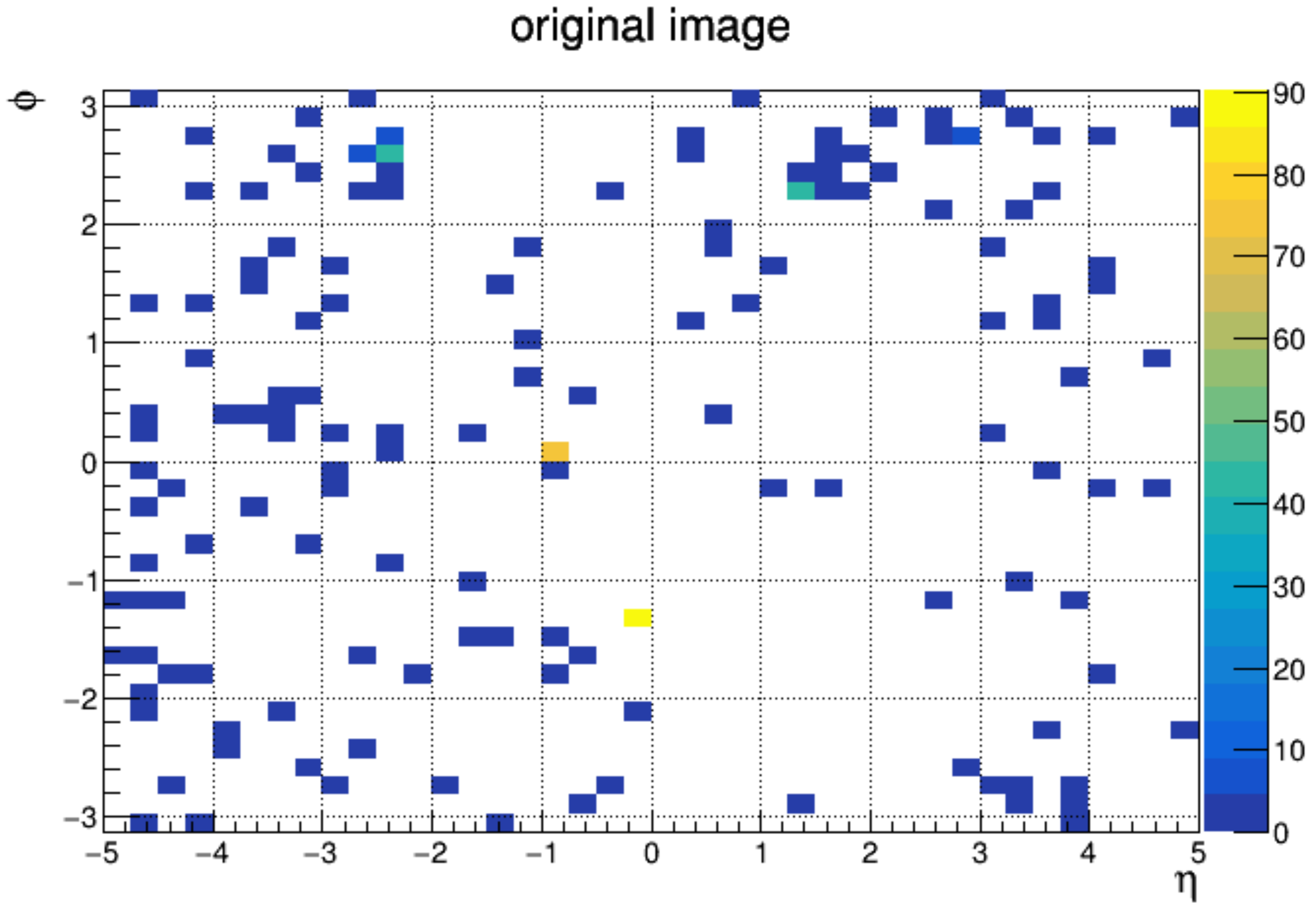
though not very efficient, yet useful for subsequent event-level classification

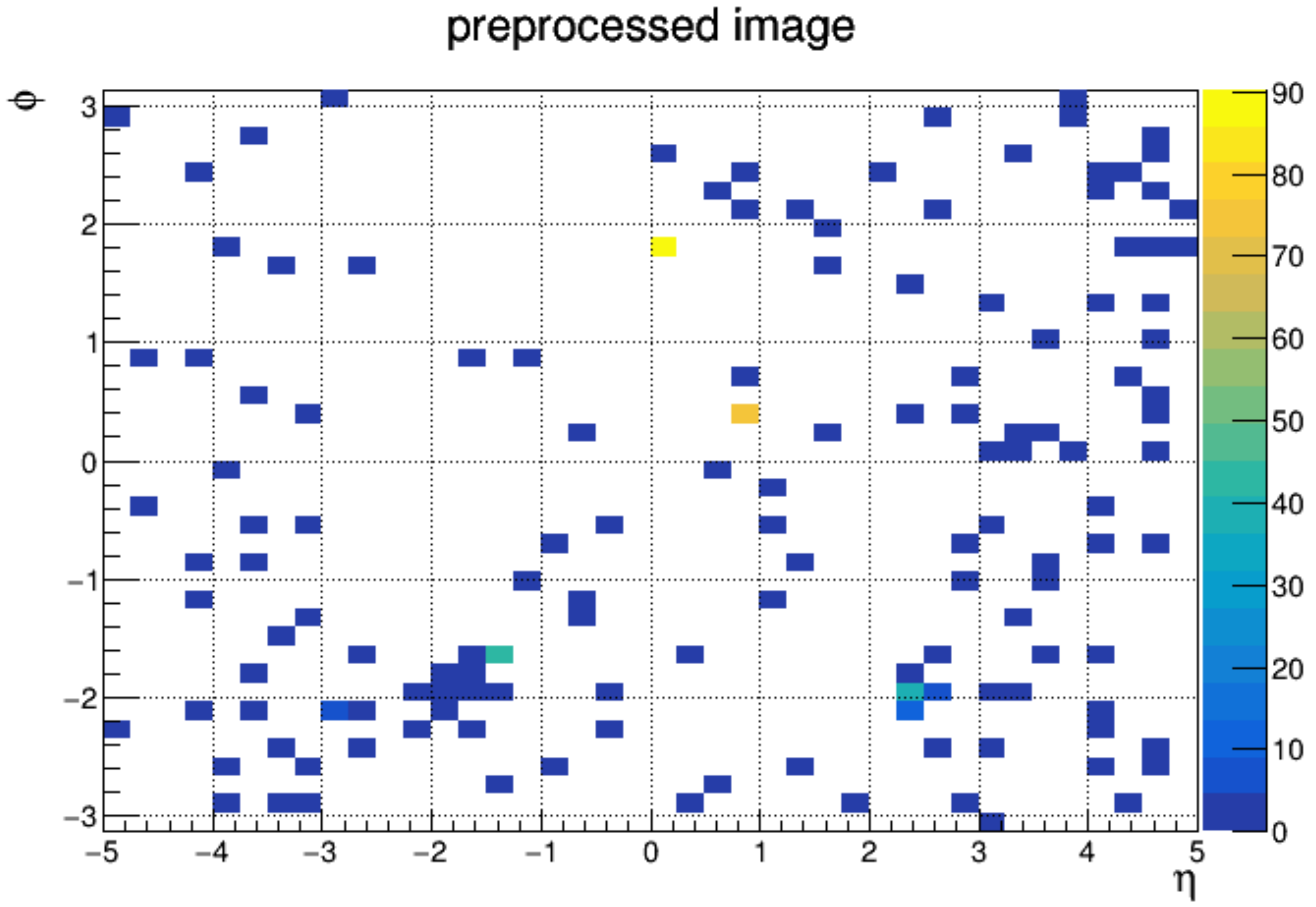


one tagger trained on mixed samples of leading and subleading jets

Event Image Preparation

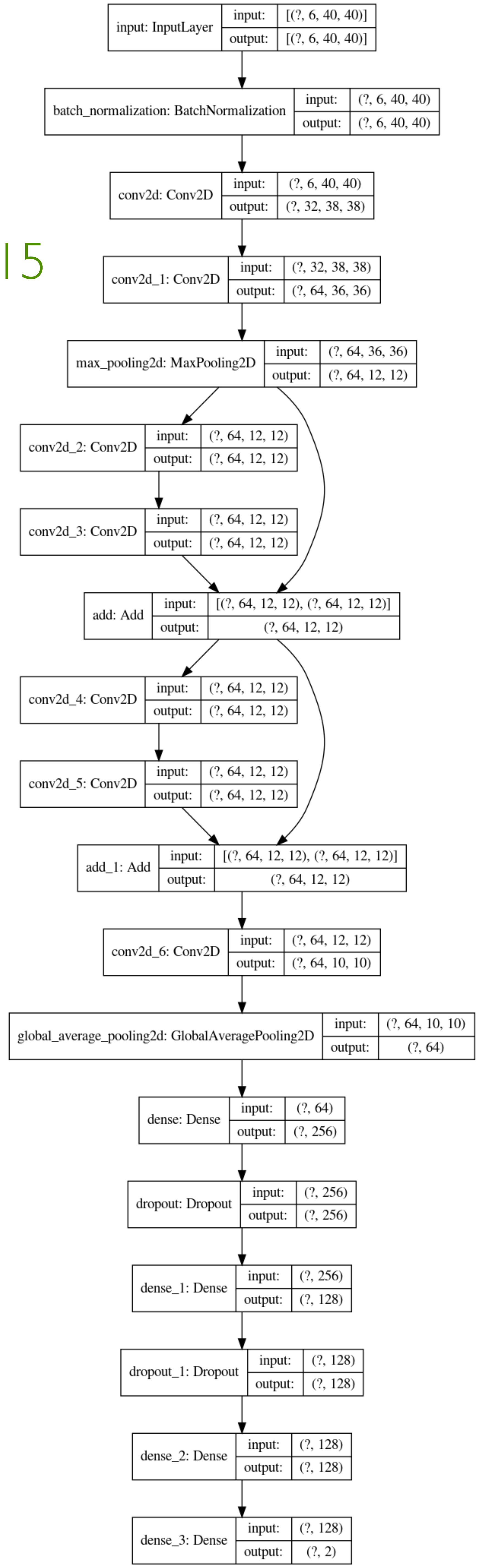
- **Pre-processing**: move the weighted center to the origin along the ϕ direction, and flip the image vertically or horizontally to make the upper-right quadrant more energetic than all the others
- **Pixelation**: from detector responses into 40×40 pixels
- **6 channels**: Tower E_T, Tower hits, Track E_T, Track hits, Photon E_T, and Photon hits





Event-CNN

- We employ a toy ResNet model in our event-CNN. He, Zhang, Ren, and Sun 2015
- Two Convolution Layers form a *residual block* in ResNet.
- There are *shortcuts* connecting the residual blocks, enabling us to deepen our model without suffering from the degradation problem.
- The sizes of filters in the Convolution Layers and pools in the Pooling Layers are all 3×3.



input: InputLayer | input: [(?, 6, 40, 40)] | output: [(?, 6, 40, 40)]

batch_normalization: BatchNormalization | input: (?, 6, 40, 40) | output: (?, 6, 40, 40)

conv2d: Conv2D | input: (?, 6, 40, 40) | output: (?, 32, 38, 38)

conv2d_1: Conv2D | input: (?, 32, 38, 38) | output: (?, 64, 36, 36)

max_pooling2d: MaxPooling2D | input: (?, 64, 36, 36) | output: (?, 64, 12, 12)

conv2d_2: Conv2D | input: (?, 64, 12, 12) | output: (?, 64, 12, 12)

conv2d_3: Conv2D | input: (?, 64, 12, 12) | output: (?, 64, 12, 12)

add: Add | input: [(?, 64, 12, 12), (?, 64, 12, 12)] | output: (?, 64, 12, 12)

conv2d_4: Conv2D | input: (?, 64, 12, 12) | output: (?, 64, 12, 12)

conv2d_5: Conv2D | input: (?, 64, 12, 12) | output: (?, 64, 12, 12)

add_1: Add | input: [(?, 64, 12, 12), (?, 64, 12, 12)] | output: (?, 64, 12, 12)

conv2d_6: Conv2D | input: (?, 64, 12, 12) | output: (?, 64, 10, 10)

global_average_pooling2d: GlobalAveragePooling2D | input: (?, 64, 10, 10) | output: (?, 64)

dense: Dense | input: (?, 64) | output: (?, 256)

dropout: Dropout | input: (?, 256) | output: (?, 256)

dense_1: Dense | input: (?, 256) | output: (?, 128)

dropout_1: Dropout | input: (?, 128) | output: (?, 128)

dense_2: Dense | input: (?, 128) | output: (?, 128)

dense_3: Dense | input: (?, 128) | output: (?, 2)

Self-Attention Model

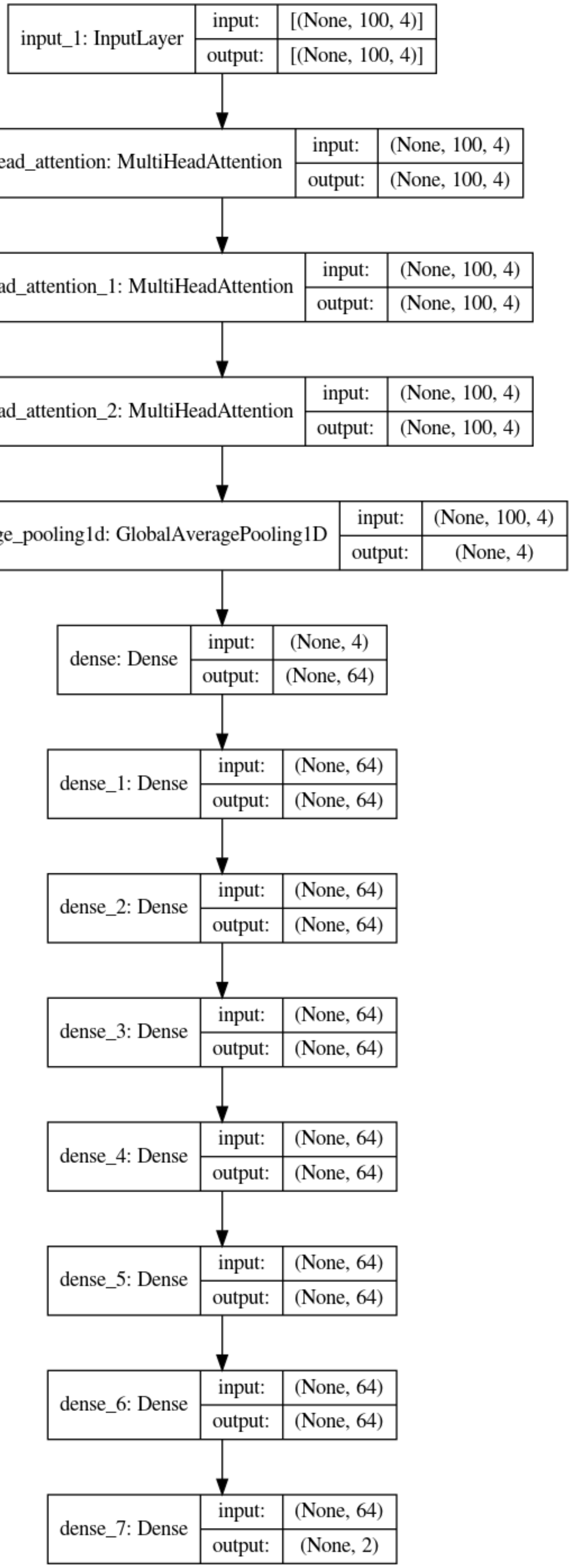
- Consider as an alternative the **self-attention** technique, which is used in the famous Transformer model dealing with sequence-to-sequence tasks.

Lin, Feng, dos Santos, Yu, Xiang, Zhou, and Bengio 2017
Vaswani, Shazeer, Parmar, Uszkoreit, Jones, Gomez, Kaiser, and Polosukhin 2017

- Instead of representing an event as an image, view the event as a *sequence* of the p_T, η, ϕ, and Q of the 100 highest-p_T reconstructed particles in the event (with zero padding for events with fewer than 100 particles).

- The self-attention network could be advantageous over event-level images because it is not subject to the information loss induced by pixelation.

- A nice property of the self-attention mechanism is that it preserves the *permutation invariance* of the inputs (as CNN).



input_1: InputLayer — input: [(None, 100, 4)] — output: [(None, 100, 4)]

multi_head_attention: MultiHeadAttention — input: (None, 100, 4) — output: (None, 100, 4)

multi_head_attention_1: MultiHeadAttention — input: (None, 100, 4) — output: (None, 100, 4)

multi_head_attention_2: MultiHeadAttention — input: (None, 100, 4) — output: (None, 100, 4)

global_average_pooling1d: GlobalAveragePooling1D — input: (None, 100, 4) — output: (None, 4)

dense: Dense — input: (None, 4) — output: (None, 64)

dense_1: Dense — input: (None, 64) — output: (None, 64)

dense_2: Dense — input: (None, 64) — output: (None, 64)

dense_3: Dense — input: (None, 64) — output: (None, 64)

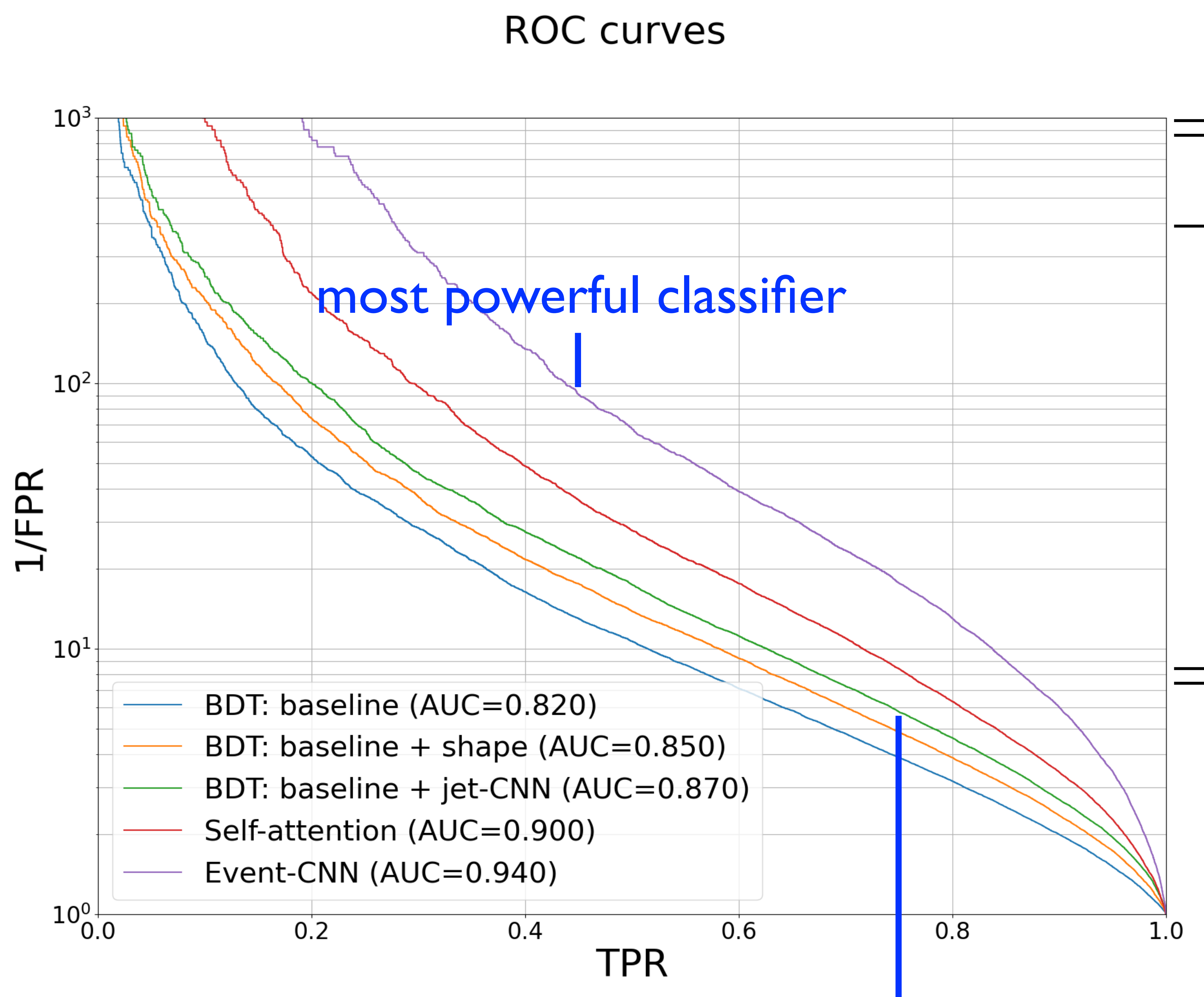
dense_4: Dense — input: (None, 64) — output: (None, 64)

dense_5: Dense — input: (None, 64) — output: (None, 64)

dense_6: Dense — input: (None, 64) — output: (None, 64)

dense_7: Dense — input: (None, 64) — output: (None, 2)

Comparison of Models



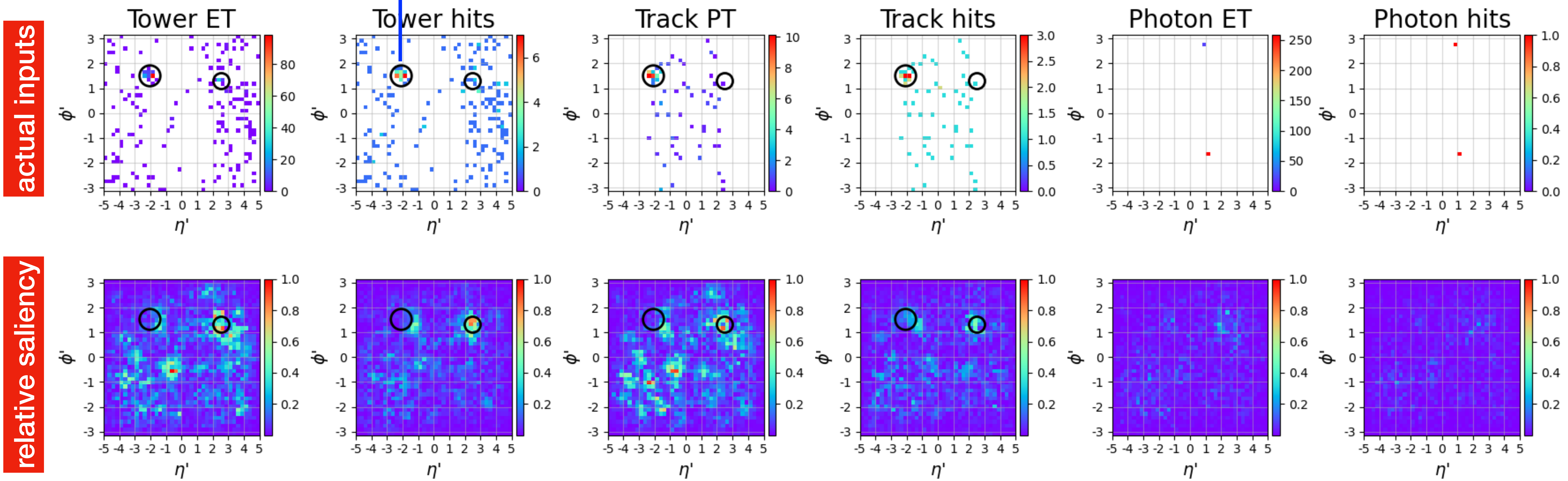
	FPR	AUC
BDT: baseline	0.035	0.820
BDT: baseline + shape	0.027	0.850
BDT: baseline + jet-CNN	0.022	0.870
Self-attention	0.010	0.900
Event-CNN	0.003	0.940

Performance comparison at TPR = 0.3

- Our jet-CNN score is more useful than jet shapes.
- Combining jet shapes and jet-CNN scores tried, but does not make any improvement
  ➠ jet-CNN has learned the information contained in the human-engineered jet shapes

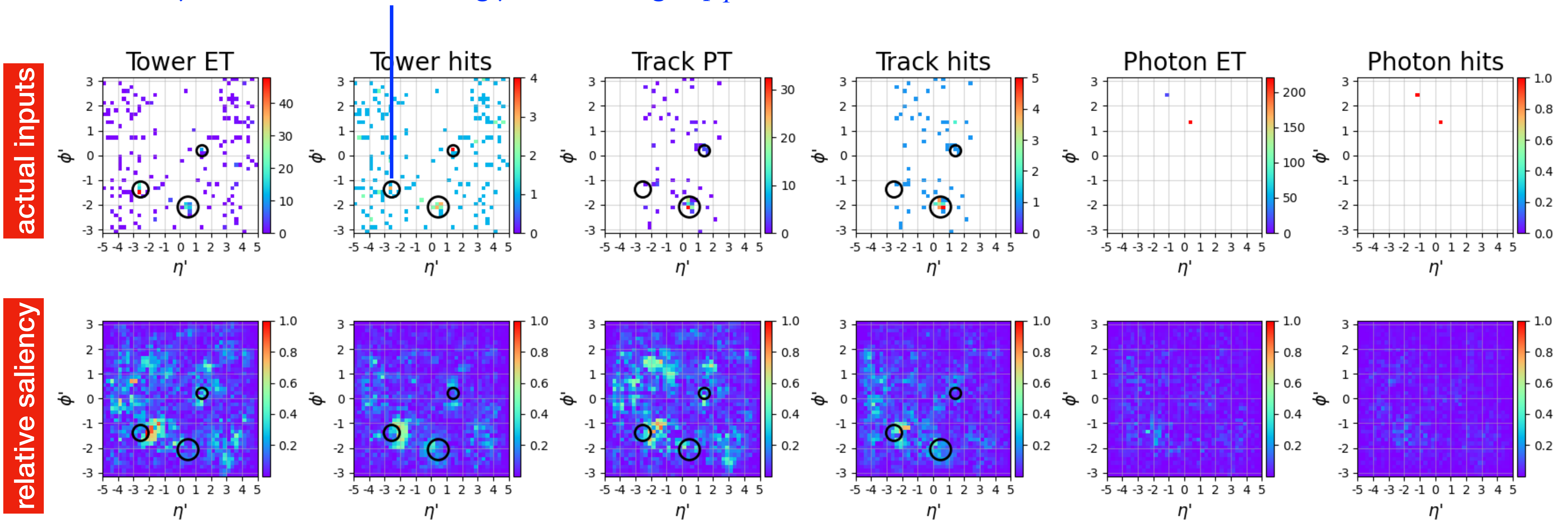
Saliency Map of A VBF Event

clustered jets, with sizes indicating jet's ordering in p_T



- CNN generally focuses on the locations with more hadronic activities.
- CNN makes use of lower p_T jets and hadronic activity that falls below the jet p_T threshold (30 GeV).
- CNN is much more focused on where jets are than the locations of photons.

Saliency Map of A GGF Event



clustered jets, with sizes indicating jet's ordering in p_T

- CNN generally focuses on the locations with more hadronic activities.
- CNN makes use of lower p_T jets and hadronic activity that falls below the jet p_T threshold (30 GeV).
- CNN is much more focused on where jets are than the locations of photons.

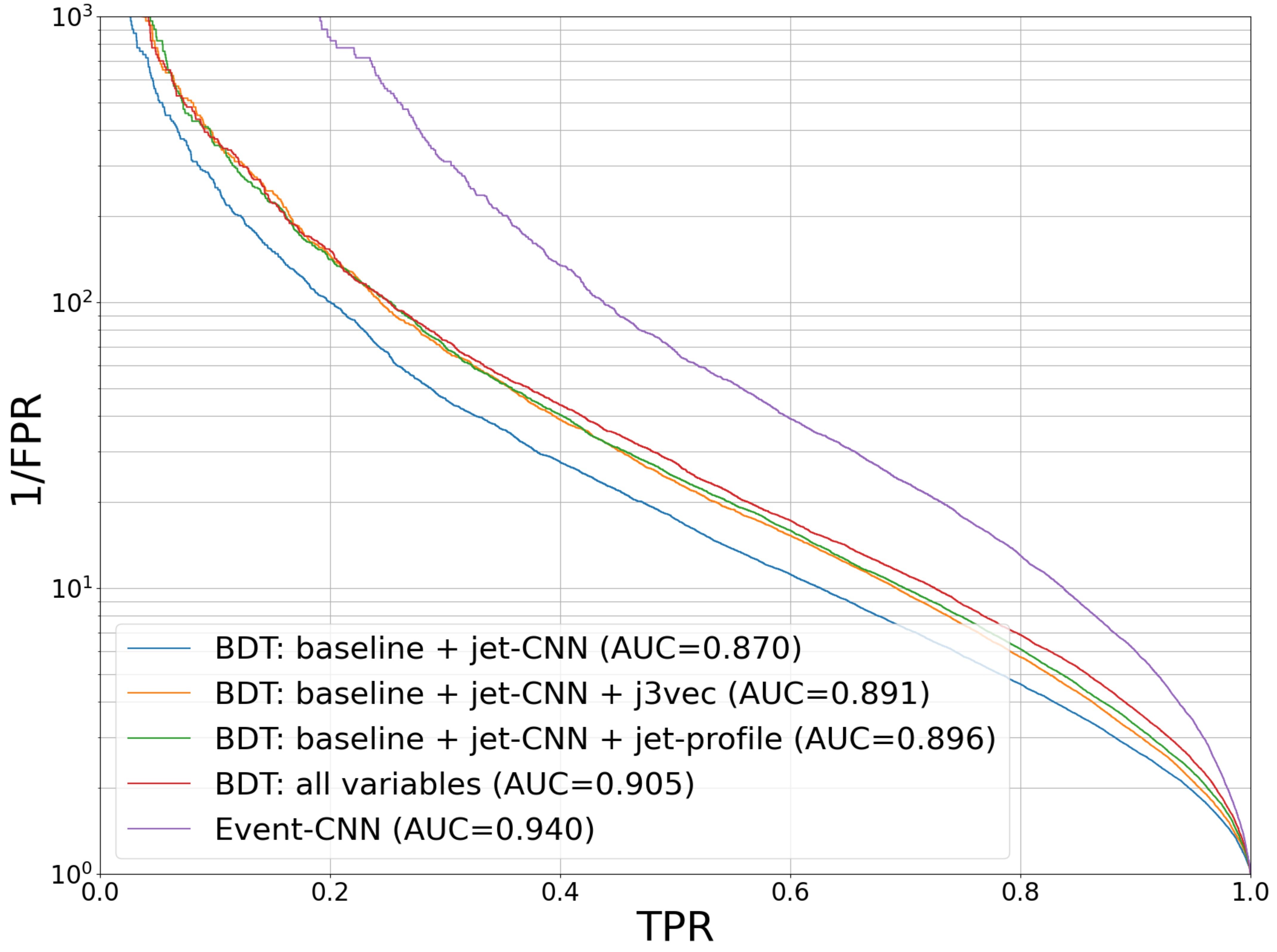
Improvements of BDTs

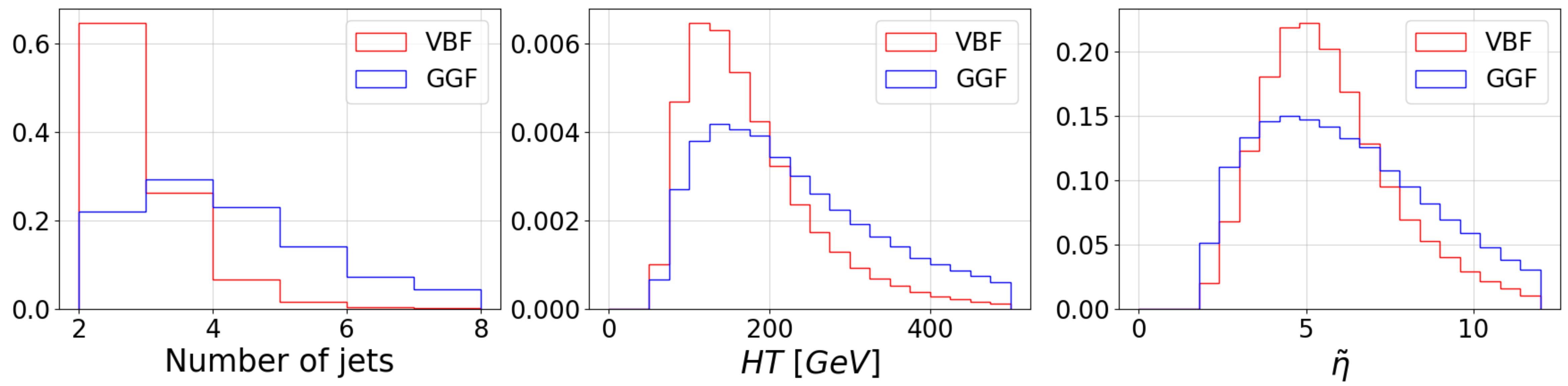
- The study of the saliency maps suggests considering information about **additional hadronic activity** in the event beyond the leading two jets:
- Including the 4-momentum of the third hardest jet, as well as inclusive kinematic variables that take all jets into account:
    - 4-momentum of the third jet in p_T ordering, denoted as “**j3vec**;”
    - “**jet-profile**” that includes:
        - $HT = \sum_{j \in \text{jets}} p_T^j$, characterizing the p_T distribution of the jets;
        - $\tilde{\eta} = \sum_{j \in \text{jets}} \left| \eta^j \right|$, characterizing the positional distribution of the jets; and
        - the number of jets.

Results of Improved BDTs

- Add the above new inputs to **BDT: baseline + jet-CNN**.

ROC curves

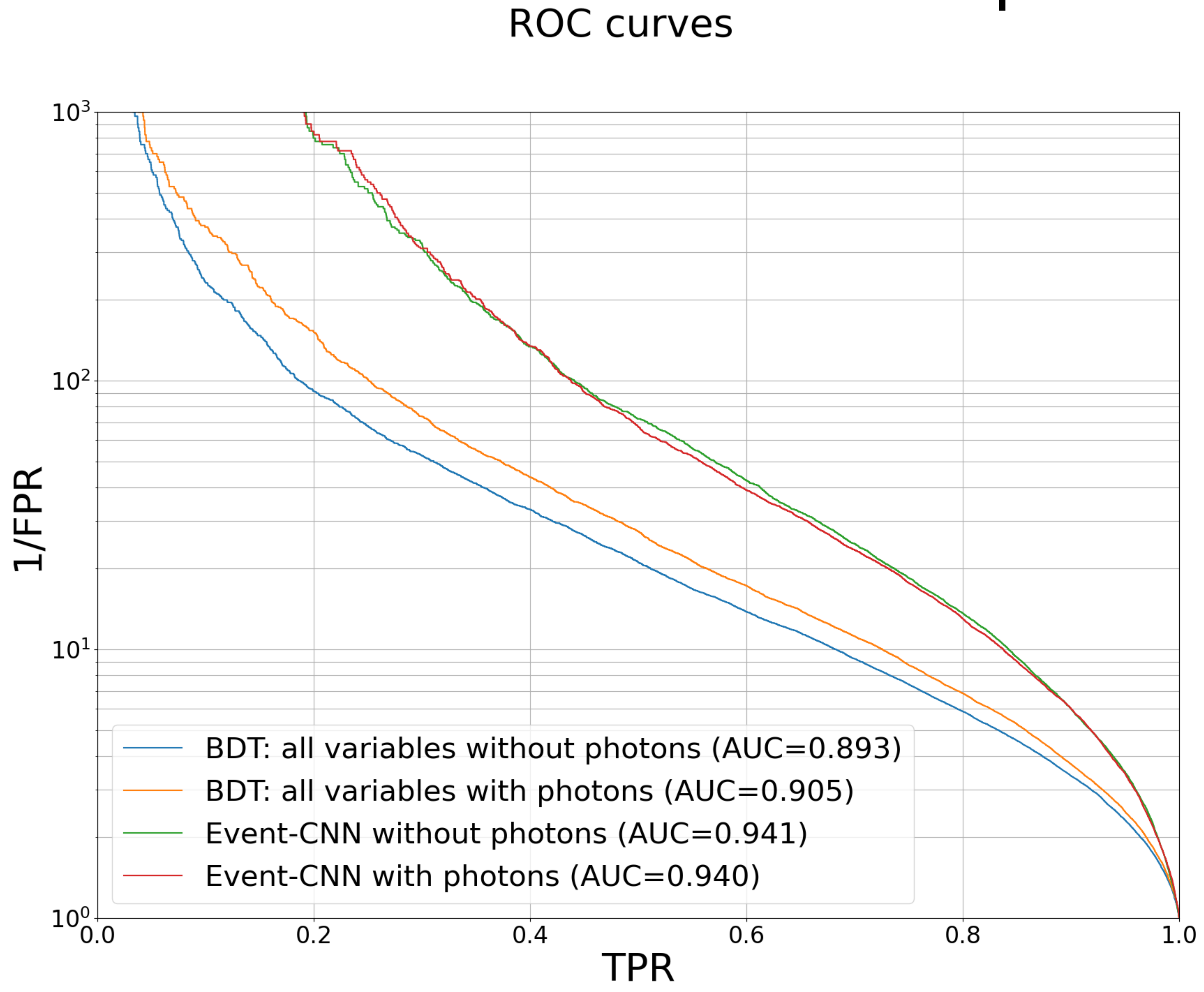




- Both 4-momentum of the third jet and the jet-profile have comparable improvements.
  ➠ they provide equivalent info in the sense that combining them does not improve
- GGF tends to have more than two jets.
  ➠ the existence of the third jet is crucial info
- The best BDT, including all 12 variables, has an AUC topping at 0.905.

Removal of Photon Information

- Using the diphoton mode as an explicit example, we show that the information of the two photons does not affect the performance of the classifier.
- A comparison of performance for **BDT: all variables** and **event-CNN** with and without the information of the photon pair.



- Could train a single VBF vs. GGF classifier that is agnostic to the Higgs decay mode.
- Could be applied to a variety of Higgs decay channels in a uniform way.

Summary

- We have proposed an event-level classifier for VBF vs GGF Higgs production channels.

- Full-event deep learning classifiers (CNN, self-attention model) that utilize low-level inputs (full-event images, particle 4-momentum sequence) significantly outperform classifiers based on high-level features (kinematic and jet shape variables).

- Through saliency maps, we have observed that additional jets beyond the leading two and unclustered hadronic activity help the CNN classification as well as the BDTs.

- We have shown the possibility of a VBF vs GGF classifier that is agnostic to the Higgs decay mode, with the performance unchanged after removing the diphoton information.

- Future directions: including high-order QCD corrections; generalizing to a multi-class classifier by including more production modes; checking decay-agnosticism for other decay modes; exploring other networks (e.g., GNN); etc.

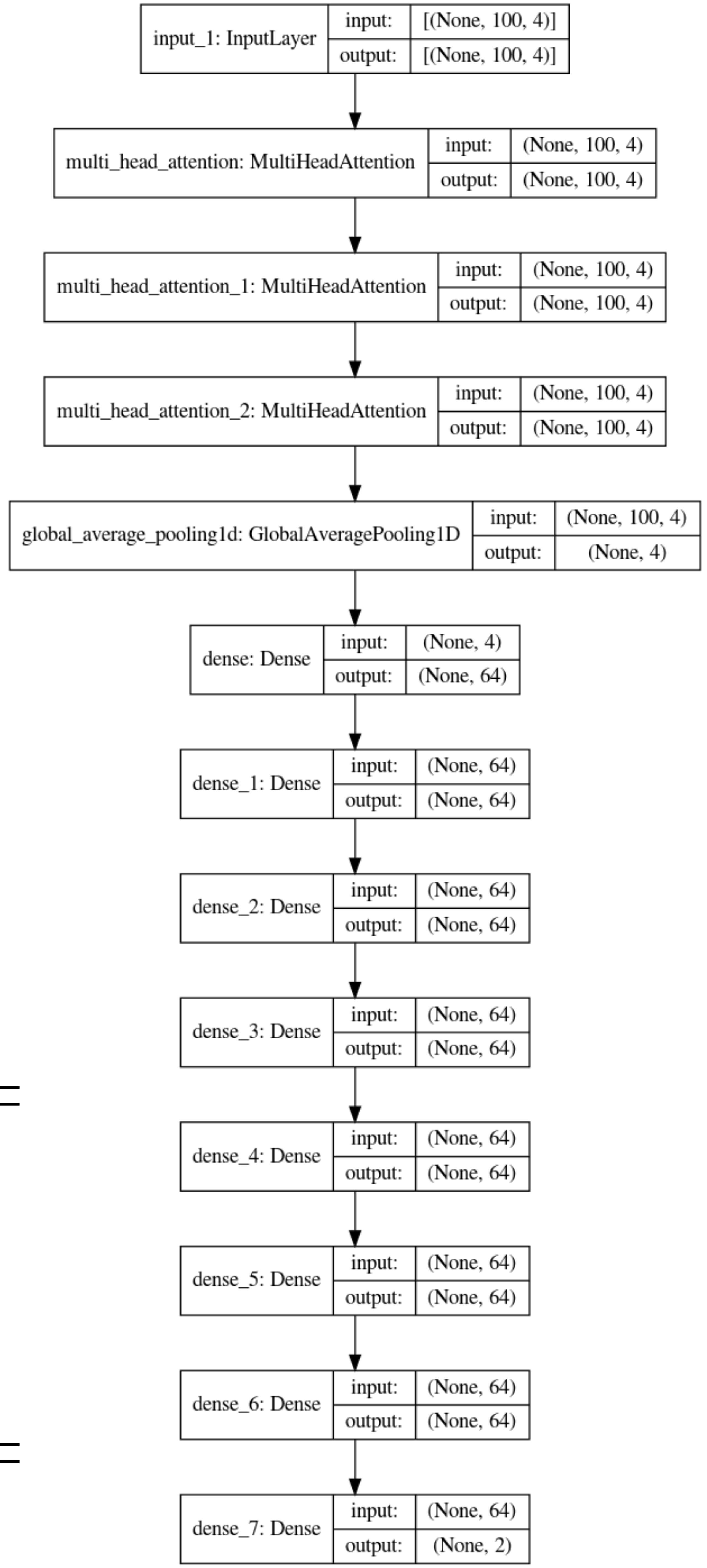
Thank You!

Backup Slides

Self-Attention Model

- The self-attention model is implemented on `TensorFlow2.5.0` and `Keras`.
- There are three five-head attention layers at the beginning, followed by a Global Average Pooling (GAP) Layer, which converts the sequence of detector responses into a single vector by taking the element-wise average, before sending to seven Dense Layers to keep permutation invariance of the input sequence.
- Hyperparameter of the model are summarized as follows:

Optimizer	Adam
Loss function	categorical crossentropy
Early stopping	50 epochs
Batch size	1024



input_1: InputLayer | input: [(None, 100, 4)] | output: [(None, 100, 4)]

multi_head_attention: MultiHeadAttention | input: (None, 100, 4) | output: (None, 100, 4)

multi_head_attention_1: MultiHeadAttention | input: (None, 100, 4) | output: (None, 100, 4)

multi_head_attention_2: MultiHeadAttention | input: (None, 100, 4) | output: (None, 100, 4)

global_average_pooling1d: GlobalAveragePooling1D | input: (None, 100, 4) | output: (None, 4)

dense: Dense | input: (None, 4) | output: (None, 64)

dense_1: Dense | input: (None, 64) | output: (None, 64)

dense_2: Dense | input: (None, 64) | output: (None, 64)

dense_3: Dense | input: (None, 64) | output: (None, 64)

dense_4: Dense | input: (None, 64) | output: (None, 64)

dense_5: Dense | input: (None, 64) | output: (None, 64)

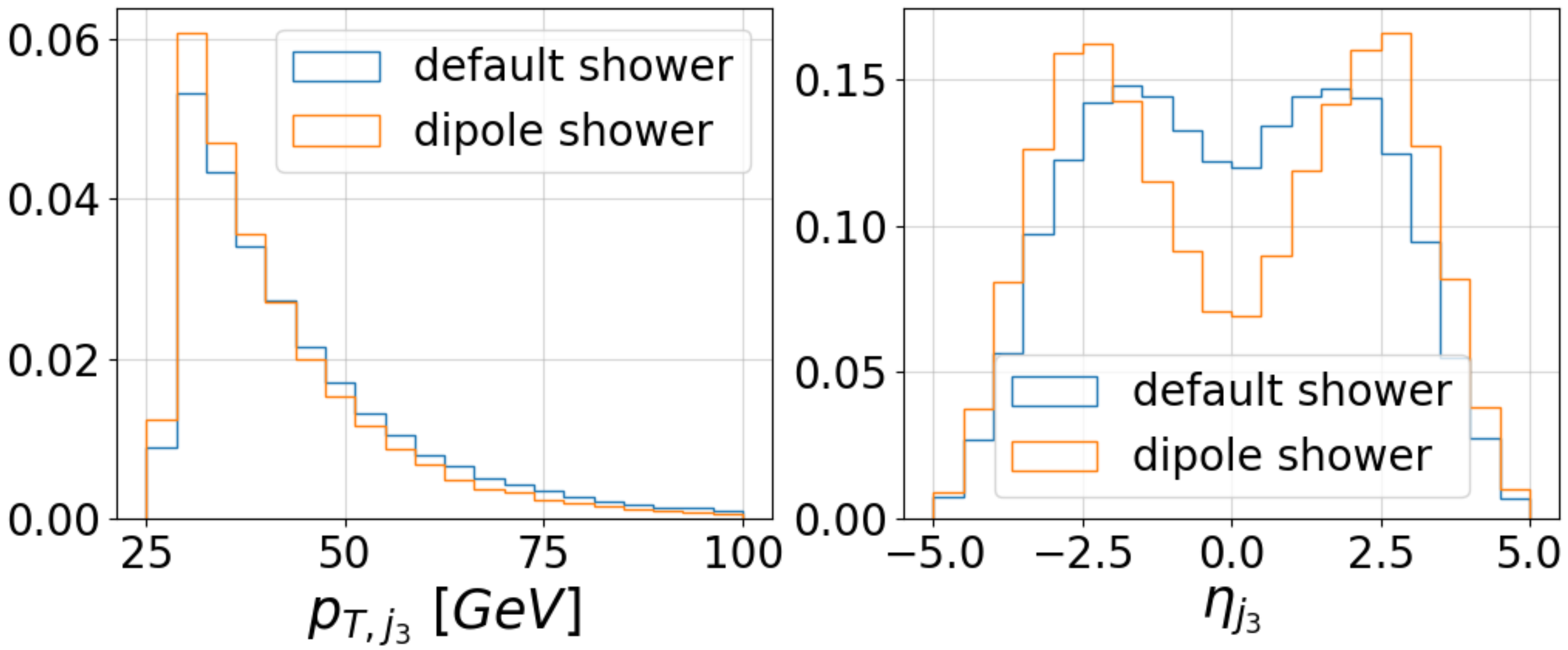
dense_6: Dense | input: (None, 64) | output: (None, 64)

dense_7: Dense | input: (None, 64) | output: (None, 2)

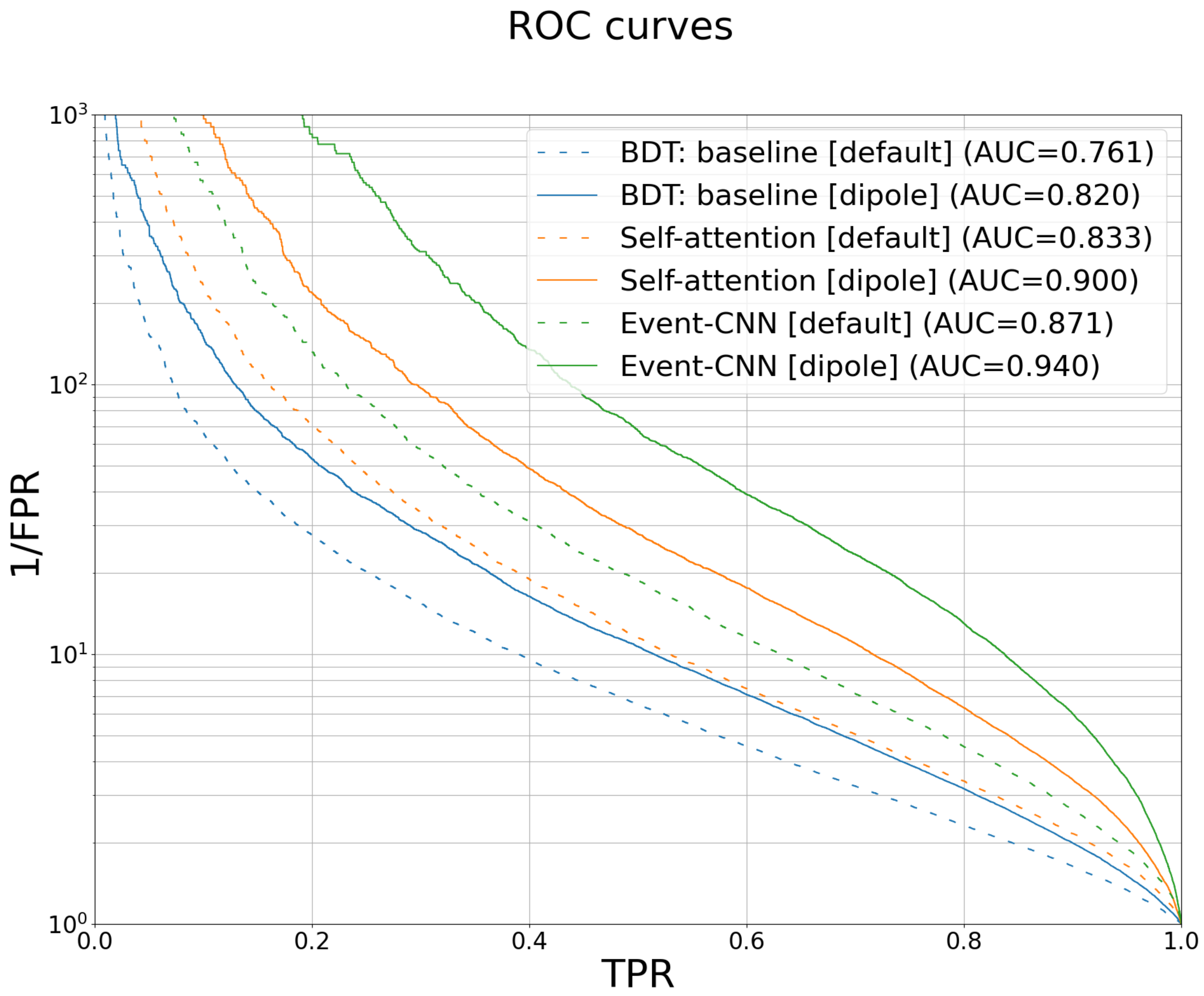
Effects of The Local Dipole Recoil Option

- The default Pythia shower depicts the emission of additional jets in VBF poorly in the central region.

Höche, Mrenna, Payne, Preuss, Skands 2022
Jäger, Karlberg, Plätzer, Scheller 2020
Konar, Ngairangbam 2022



- Comparison of using the local dipole recoil scheme for the VBF process and using the default shower scheme in Pythia.


ROC curves
10^3
10^2
10^1
10^0
1/FPR
TPR
BDT: baseline [default] (AUC=0.761)
BDT: baseline [dipole] (AUC=0.820)
Self-attention [default] (AUC=0.833)
Self-attention [dipole] (AUC=0.900)
Event-CNN [default] (AUC=0.871)
Event-CNN [dipole] (AUC=0.940)