10.1007/JHEP05(2021)208

EXPLAINABLE AI FOR ML JET TAGGERS

Garvita Agarwal, Lauren Hay, Ia Iashvili, Benjamin Mannix, Christine McLean, Margaret Morris, Salvatore Rappoccio, Ulrich Schubert

DPF2021

13th July 2021

University at Buffalo The State University of New York

Motivation

https://doi.org/10.1007/978-3-030-28954-6_10

ML Network
Black Box

Classification:
Castle

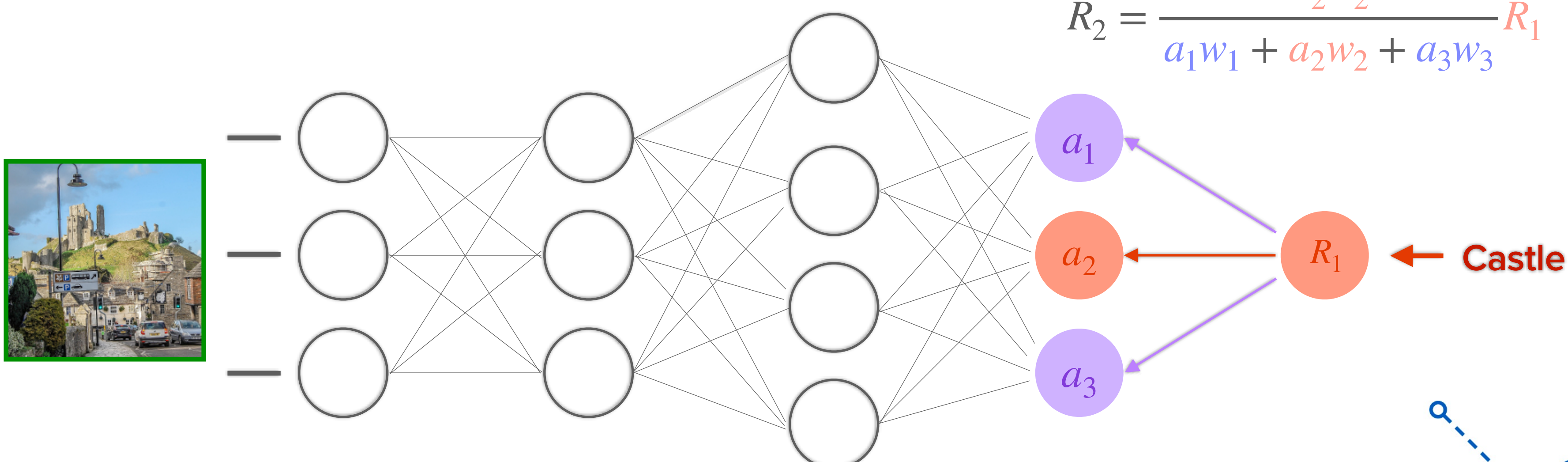
How do we understand the network's decision-making process?

ML explainability with LRP

LRP (layer-wise relevance propagation) propagates a prediction backwards through the network, assigning a relevance to each input

$$R_j = \sum_k \frac{a_j w_{jk}}{\sum_{0,j} a_j w_{jk}} R_k$$

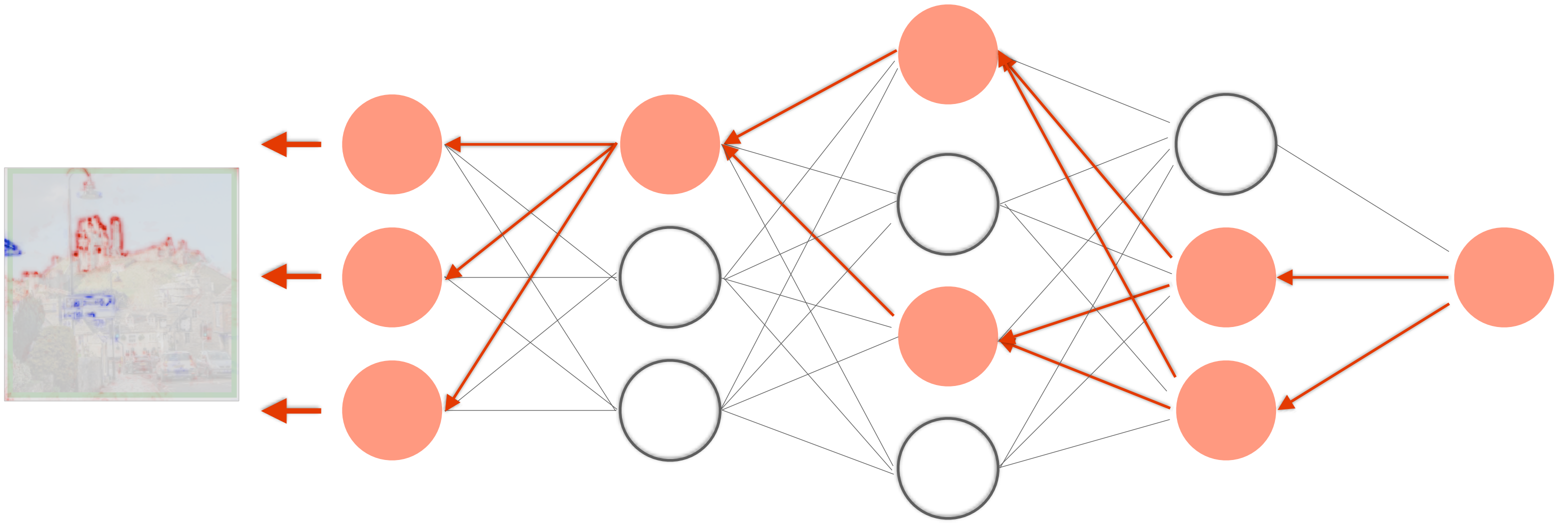
$$R_2 = \frac{a_2 w_2}{a_1 w_1 + a_2 w_2 + a_3 w_3} R_1$$



10.1007/978-3-030-28954-6_10

ML explainability with LRP

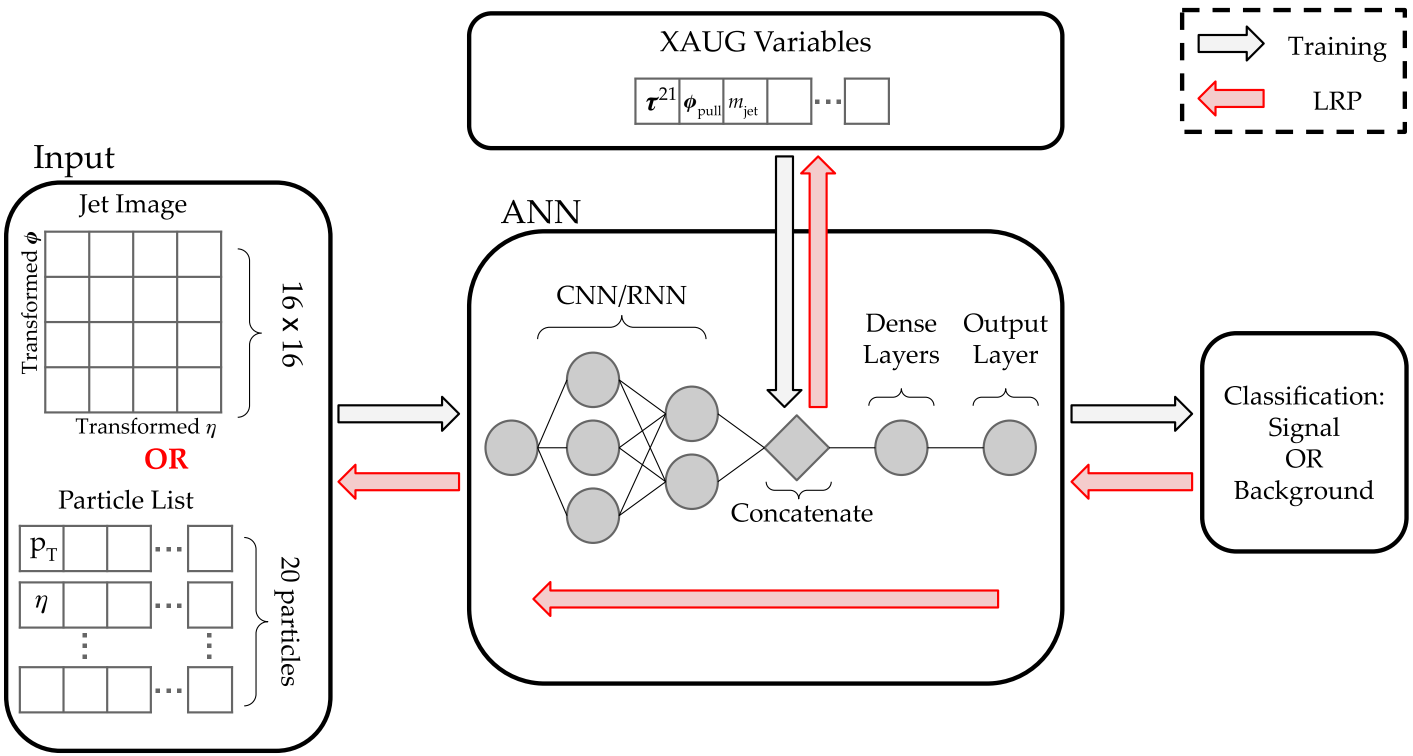
- Relevance is conserved - the backwards propagation process does not alter the prediction
- LRP attributes the entirety of the network's decision to the inputs
  - Visualised as a heat map, in the case of images



10.1007/978-3-030-28954-6_10

ML explainability with XAUG Variables

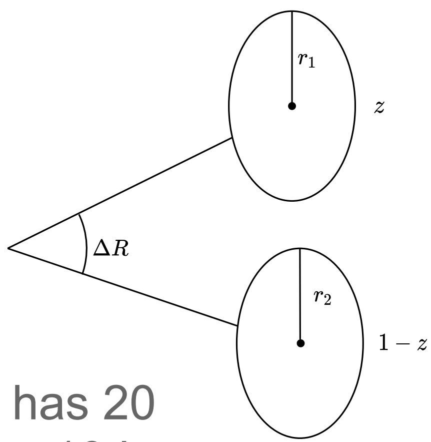
- **Goal:** explain decisions of ML jet classifiers using expert augmented (XAUG) variables
- **Method:** Input XAUGs into jet tagger, analyze network decision with LRP, and compare to network without XAUGs



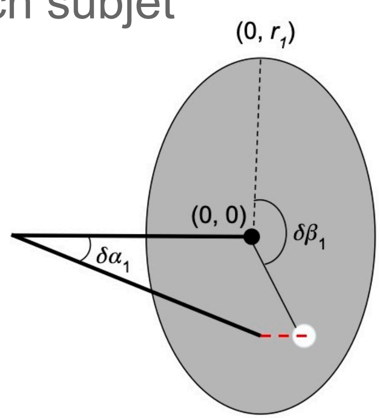
TOY MODEL

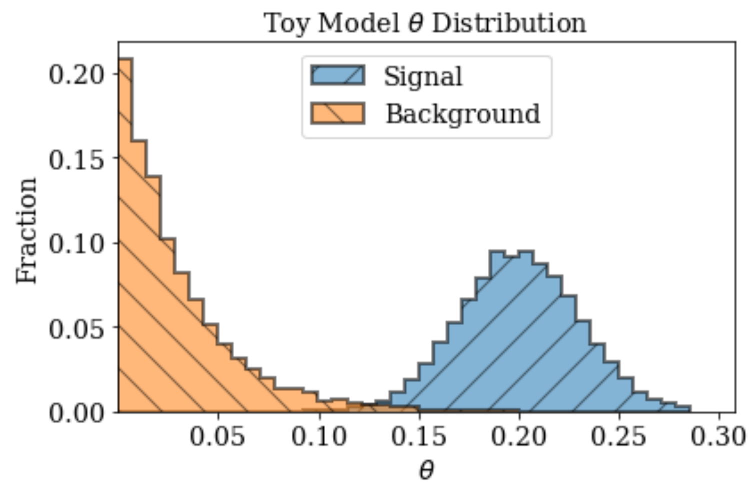
Toy Model

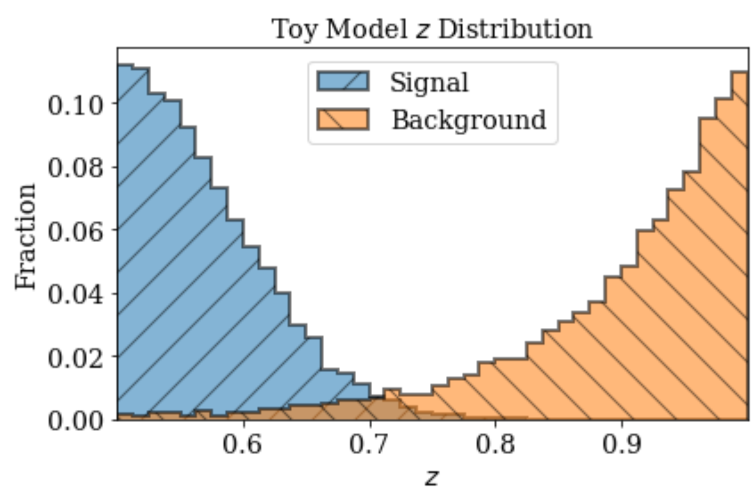
- Toy events simulated to mimic particle-level events
- **Goal:** capture all event information with a few variables



Each "jet" has 20 particles, 10 in each subjet

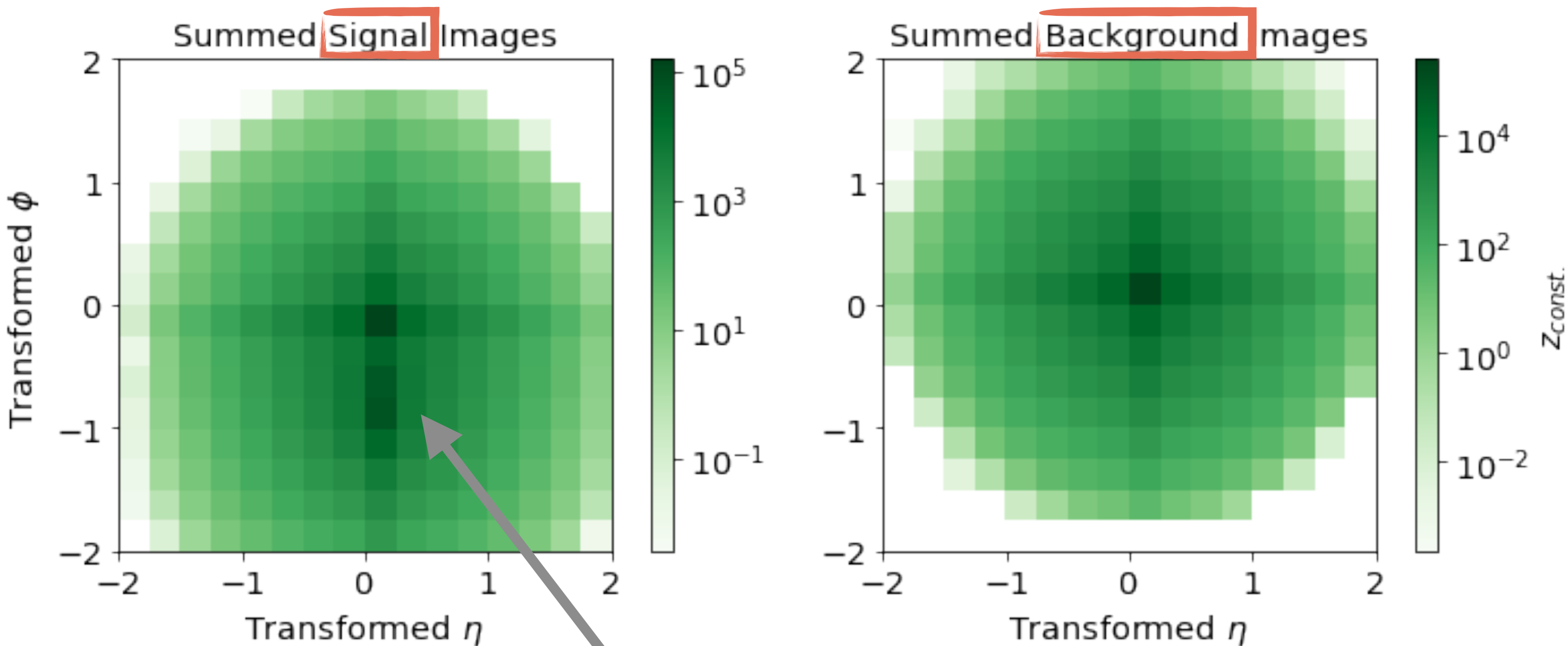






Toy Model

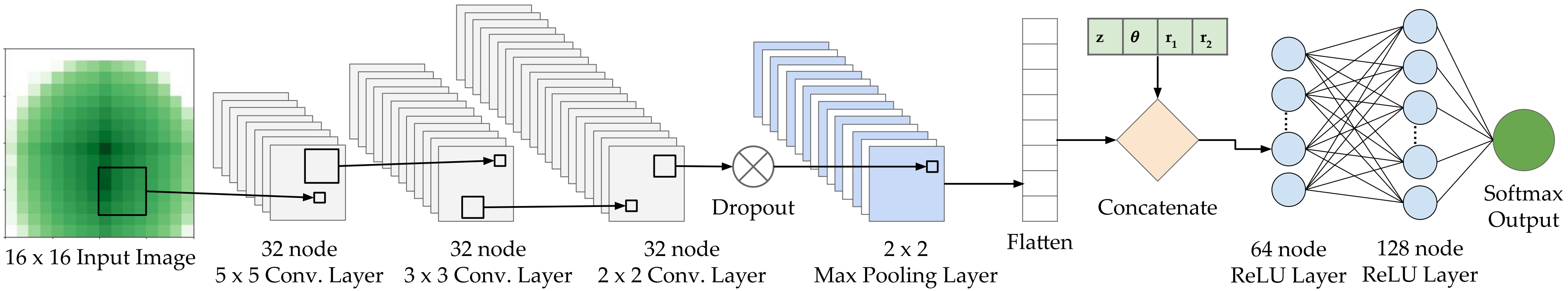
- Image pre-processing
  - Leading-p_T subjet at (0,0), sub-leading at (0,-1)
  - Parity flip



Much more pronounced second subjet

2DCNN

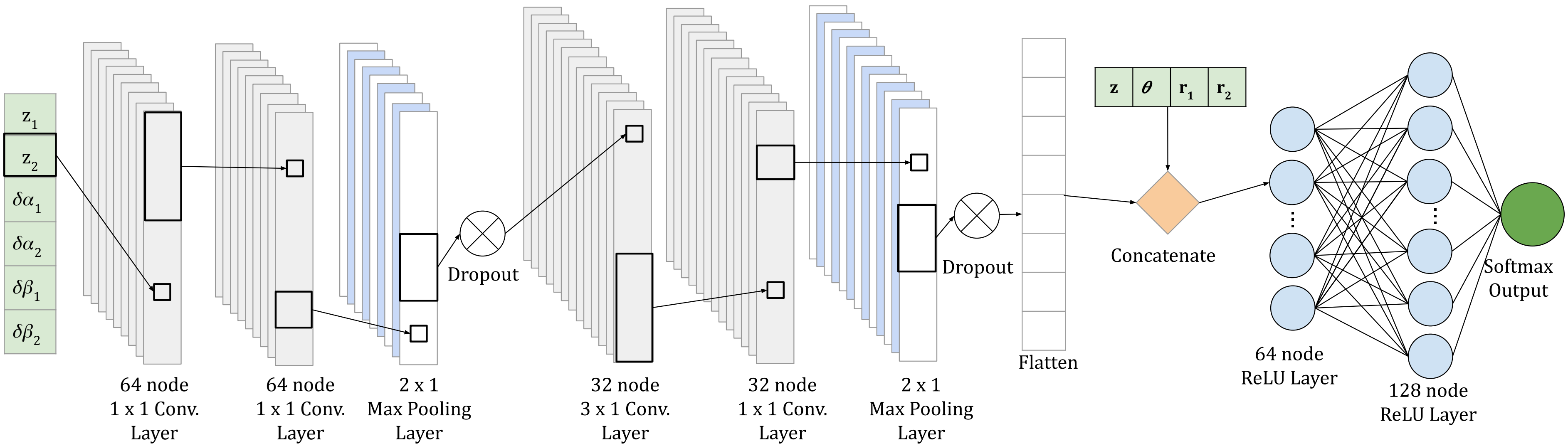
Architecture based on ImageTop network



10.1088/1748-0221/15/06/P06005

1DCNN

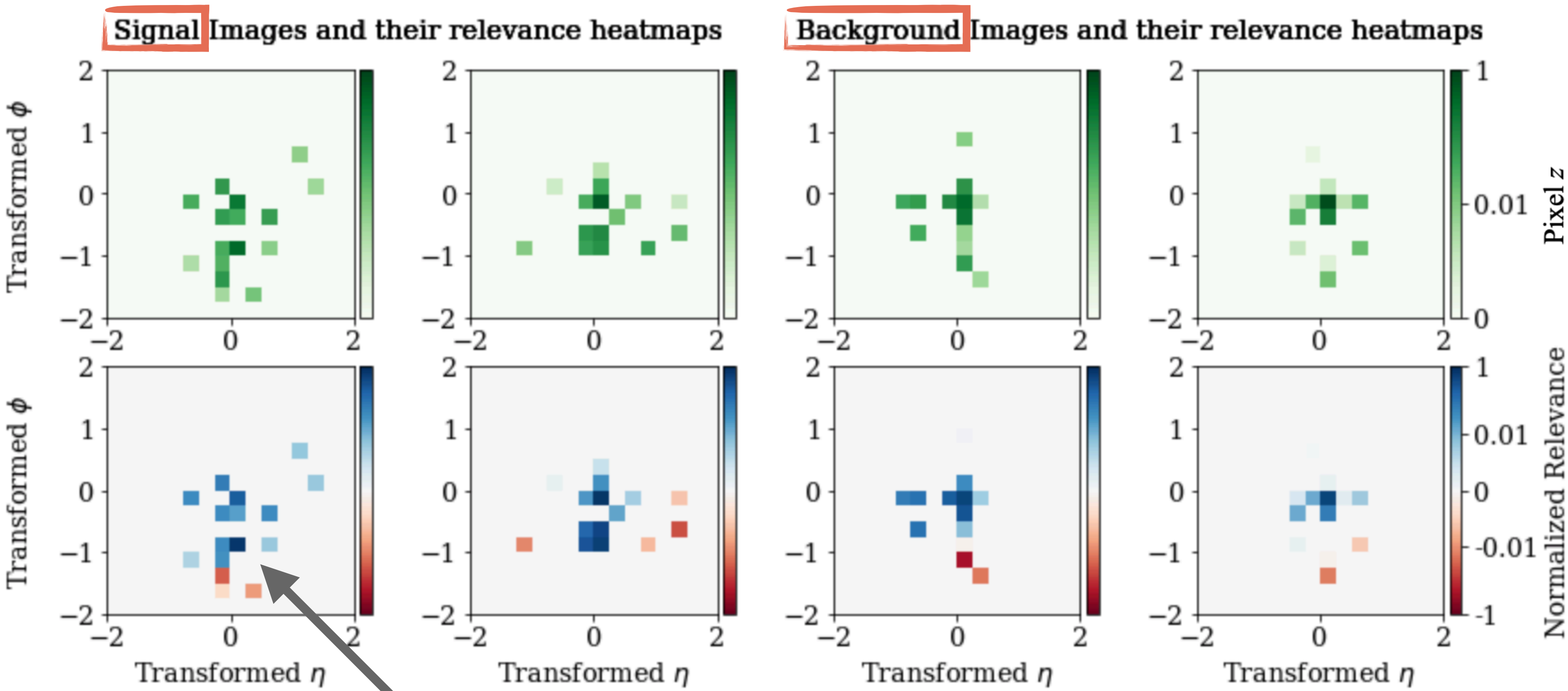
Architecture based on DeepAK8 jet classifier



10.1088/1748-0221/15/06/P06005

Toy 2DCNN LRP Heatmaps

Inputs

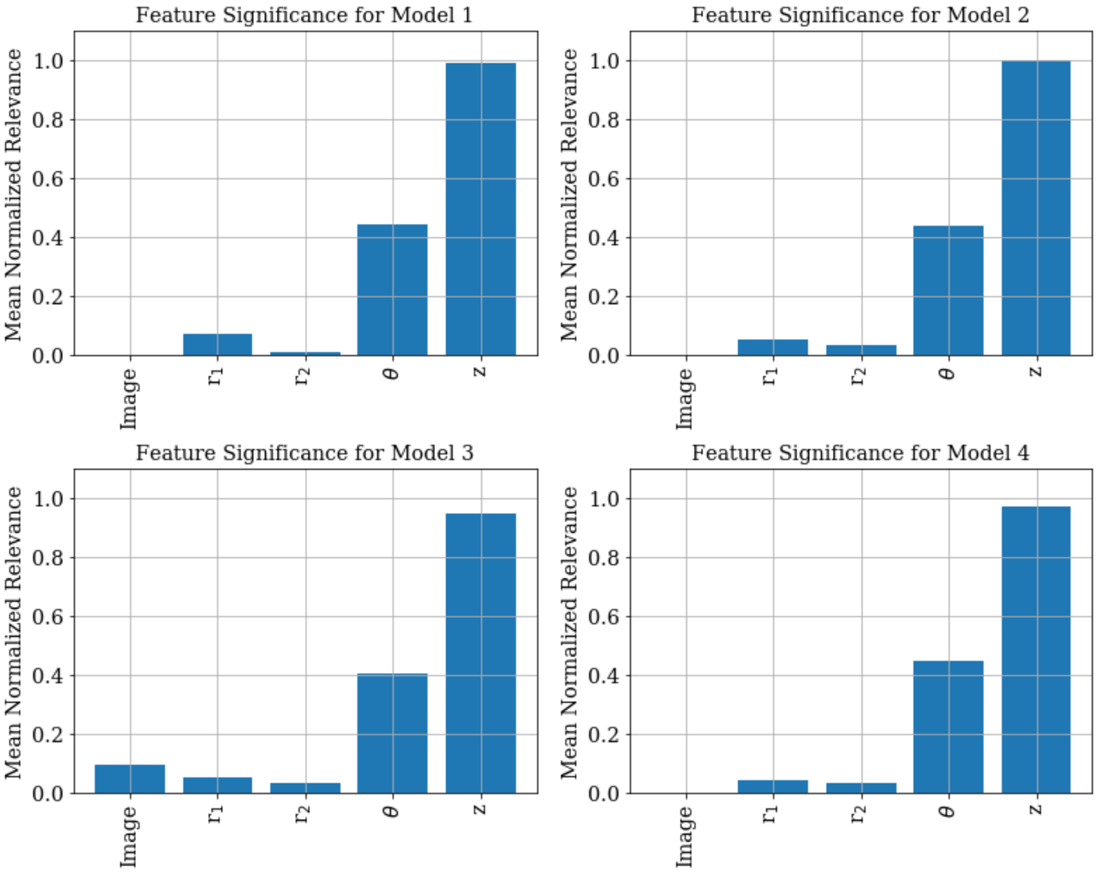


LRP Results

Signal: more relevance along ϕ axis

Toy 2DCNN Results

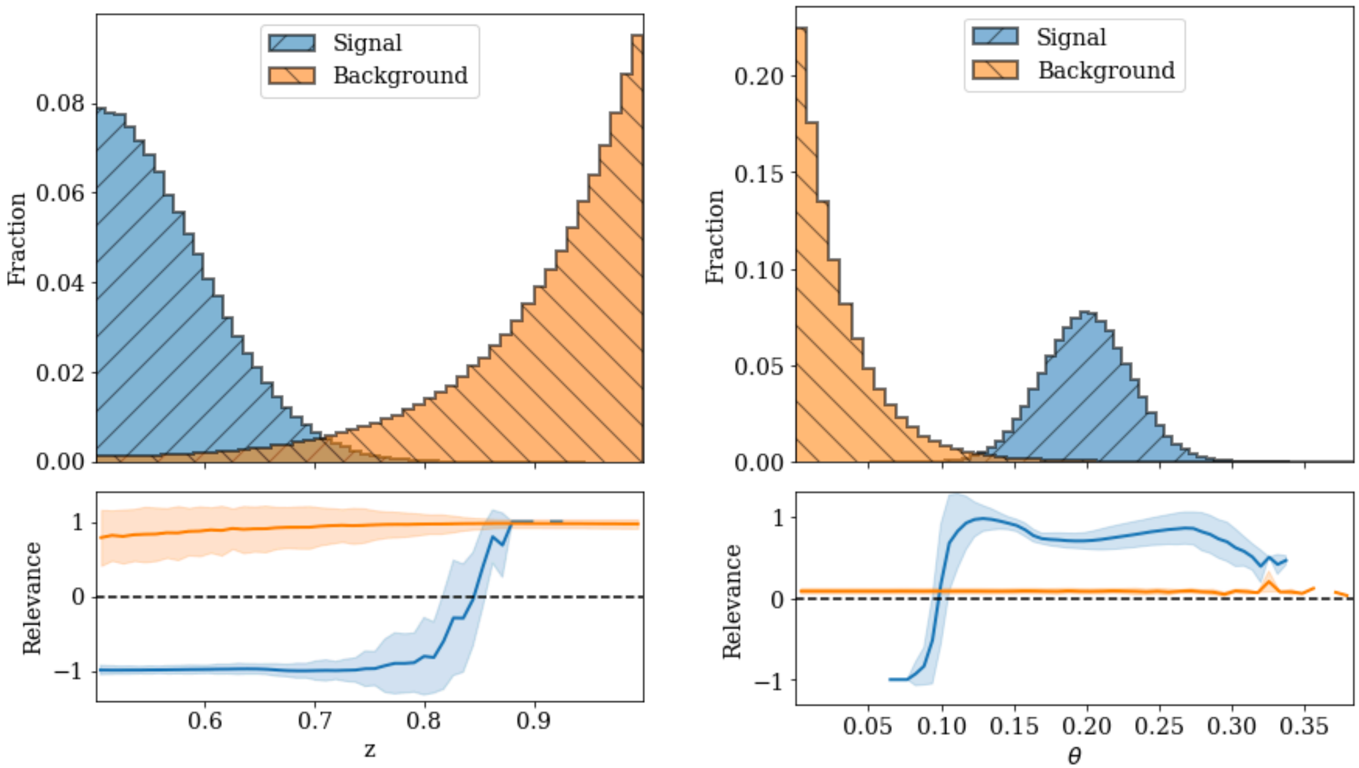
- Mean normalized relevance
  - **For each event:** find feature with max absolute LRP score, divide all scores by this max value
  - **For each image:** sum absolute value of normalized pixels to get a single image LRP score
  - **For each feature:** average normalized relevance scores across all events



Some variation between trainings

Toy 2DCNN Results

- Profile plots: relevance vs corresponding input variable
- For some profiles relevance appears to reflect input distribution, but other don't - networks' decision boundaries live in a higher dimensional space

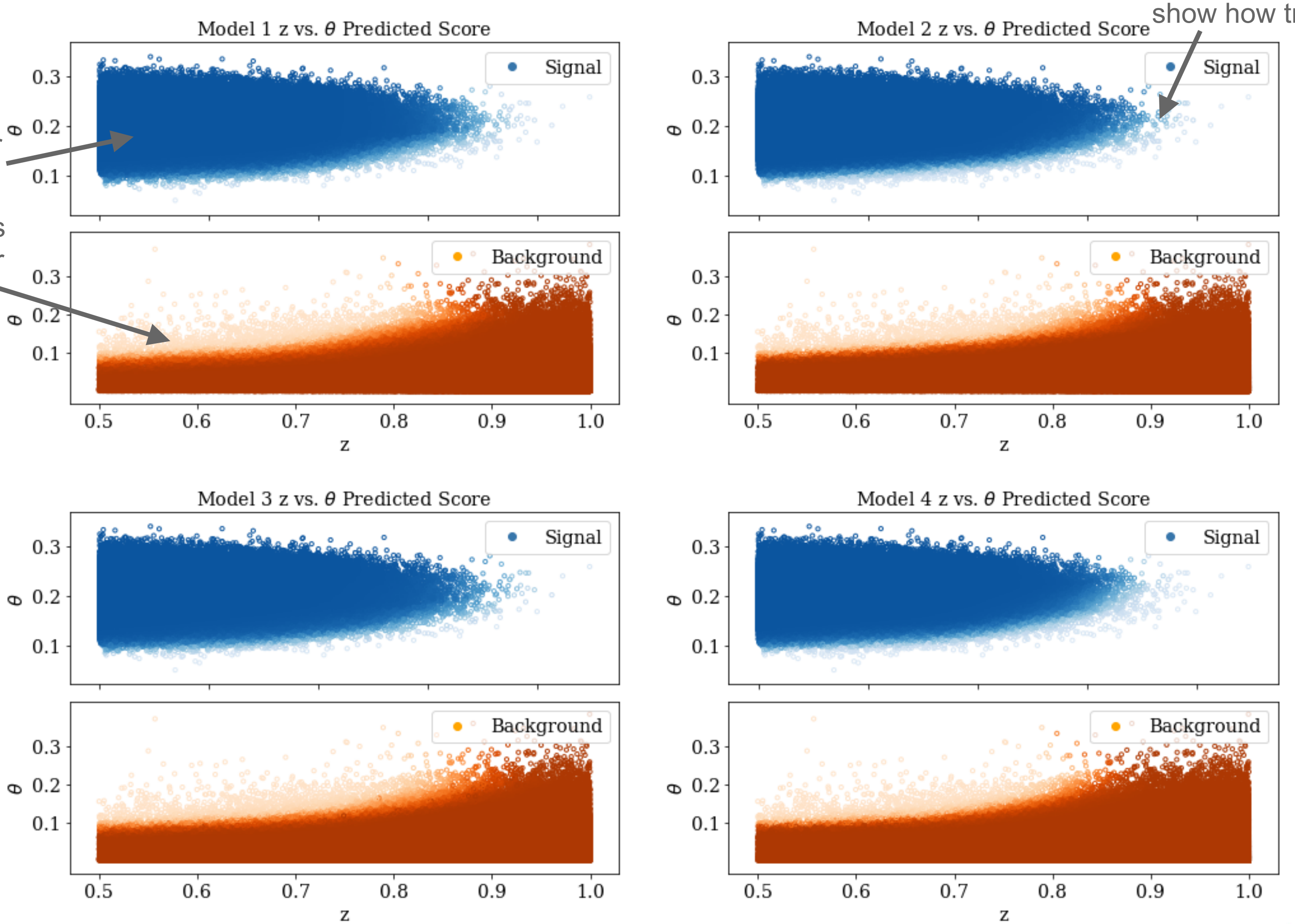


Toy 2DCNN Results

Differences in boundary shapes show how trainings vary

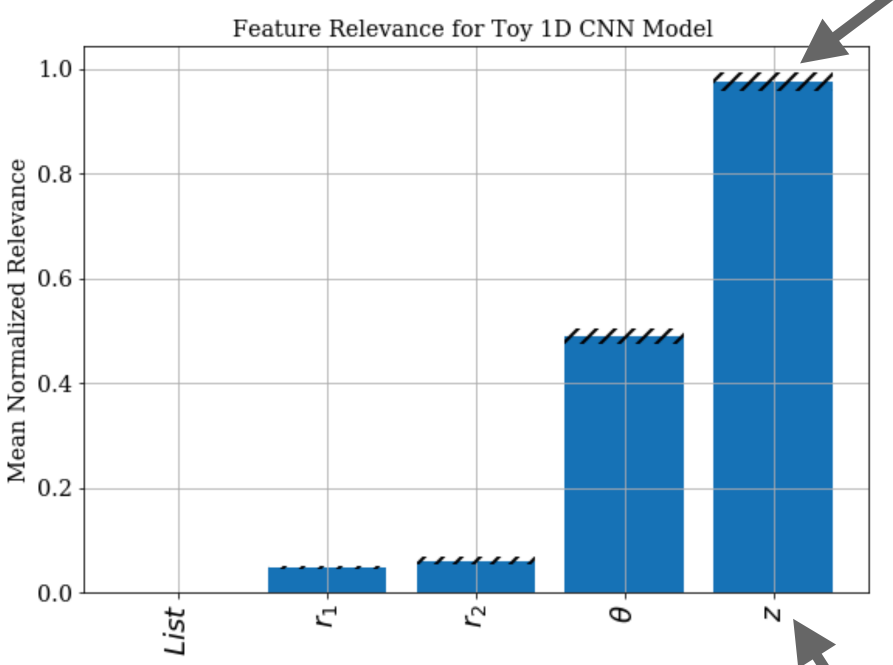
Darker markers corresponds to higher relevance scores.

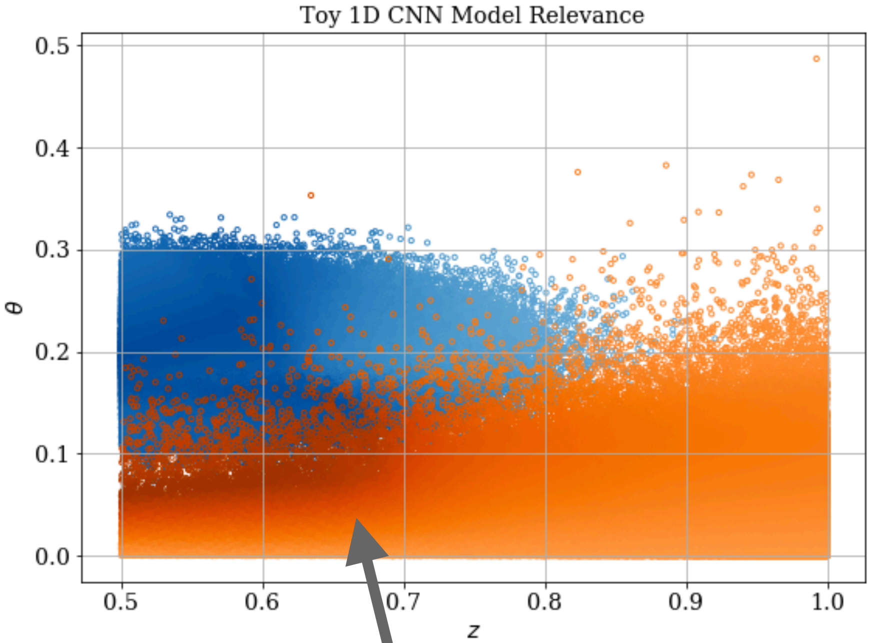
Sharp gradient shows decision boundary for these variables



Toy 1DCNN Results

Error bars show standard deviation of relevance after multiple trainings.





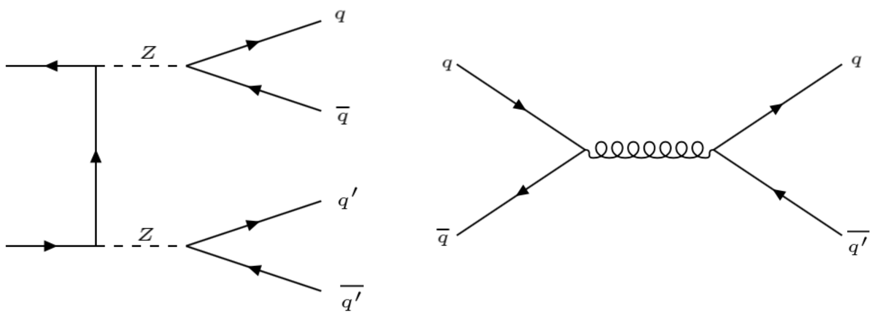
Most relevant features are same as 2DCNN.

More robust "substructure" within relevance of the top two variables.

PYTHIA MODEL

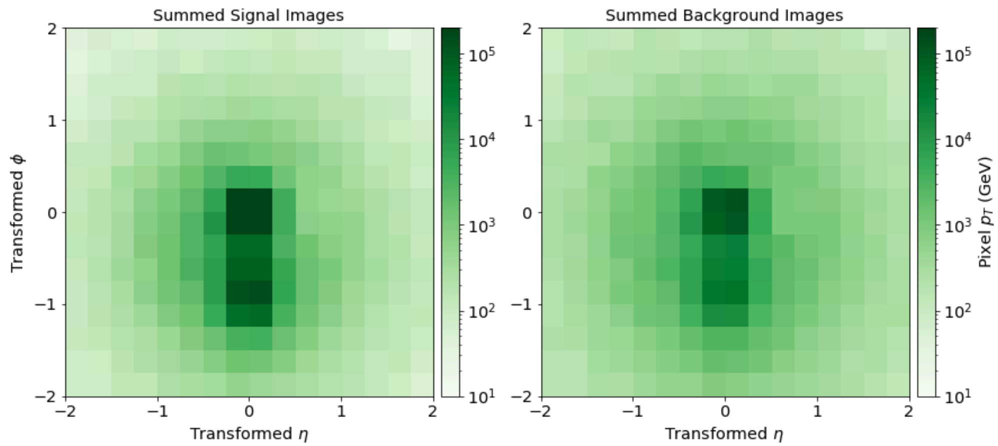
Pythia Model

- **Simulated with Pythia8**
  - Signal: SM ZZ, $Z \rightarrow b\bar{b}$
  - QCD



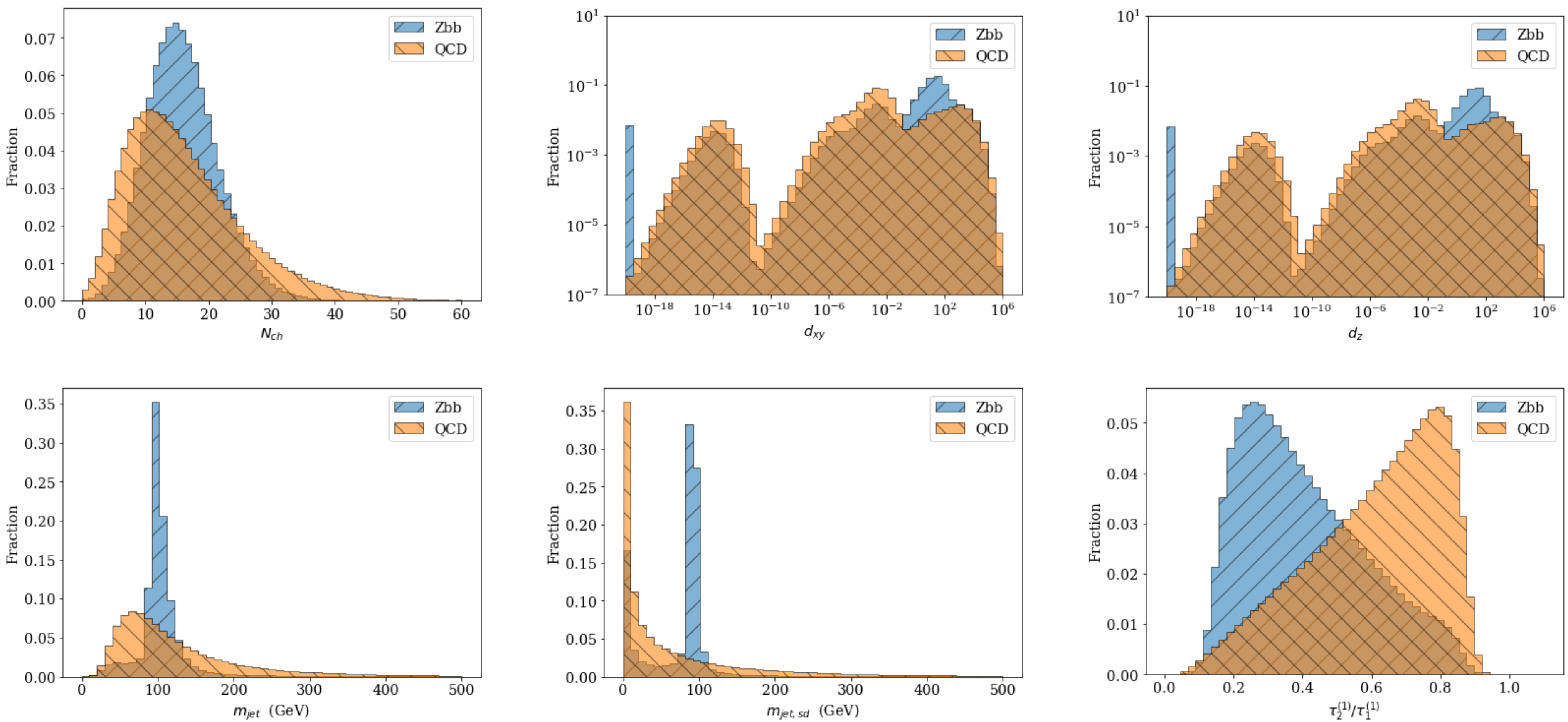
- **Jets**
  - Consider leading AK8 jet
  - $p_T > 200$ GeV
  - mMDT: $z_{cut} = 0.1, \beta = 0$

- **Preprocessing**
  - Rotating and scaling so that lower pT subjet is always at (0,-1), and normalise inputs w.r.t. jet pT, parity flip
  - Same as toy model



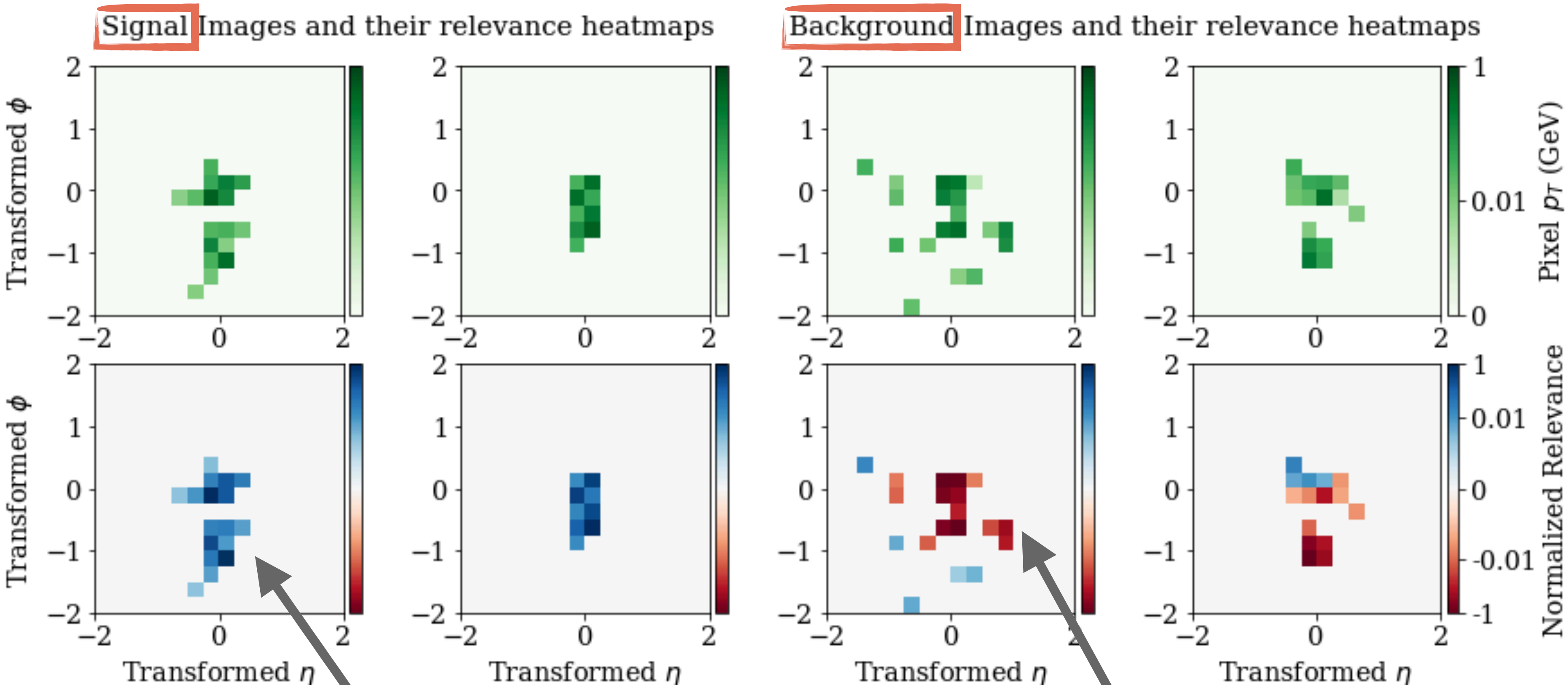
Pythia Model

Use same network structures as Toy Model, replacing inputs with equivalent counterparts.



Pythia LRP Heatmaps

Inputs



LRP Results

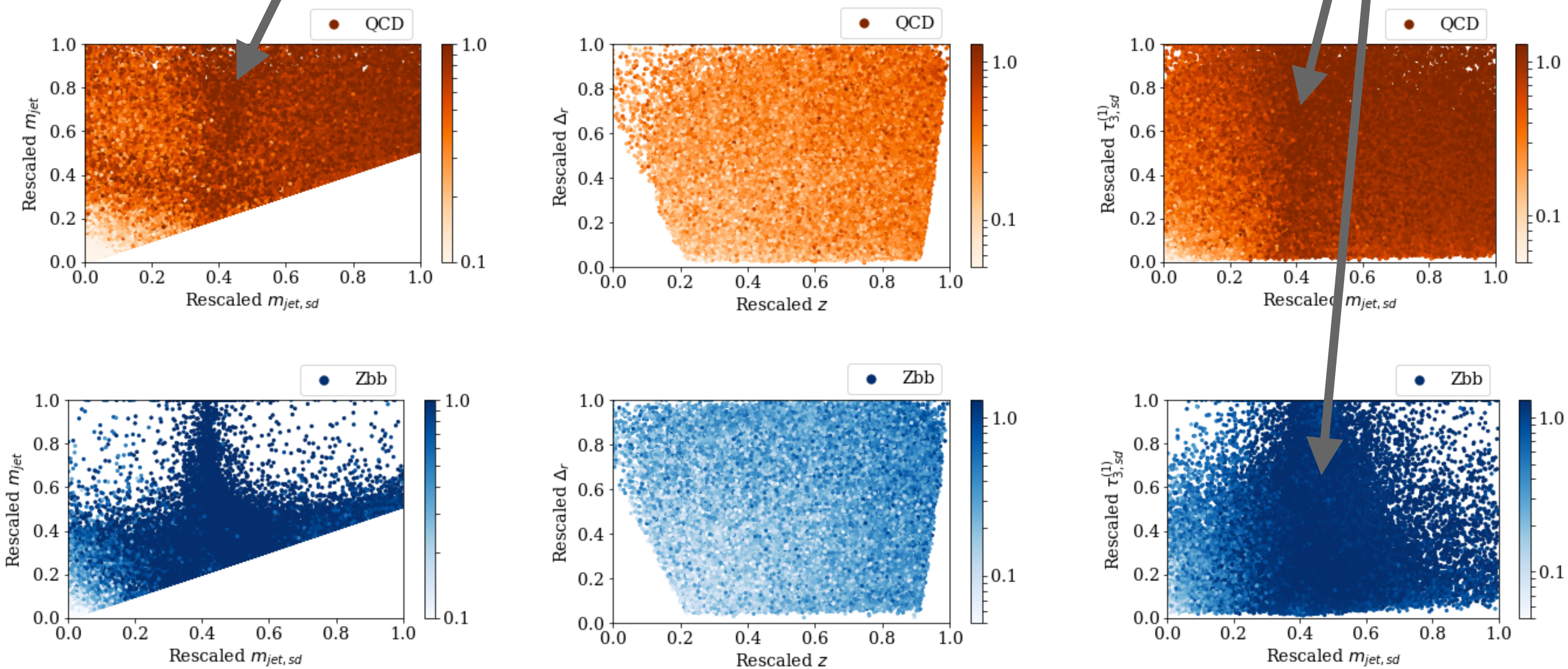
Signal: mostly positive relevance, primarily along ϕ axis

Background: mostly negative relevance, more diffuse

Pythia 1DCNN Results

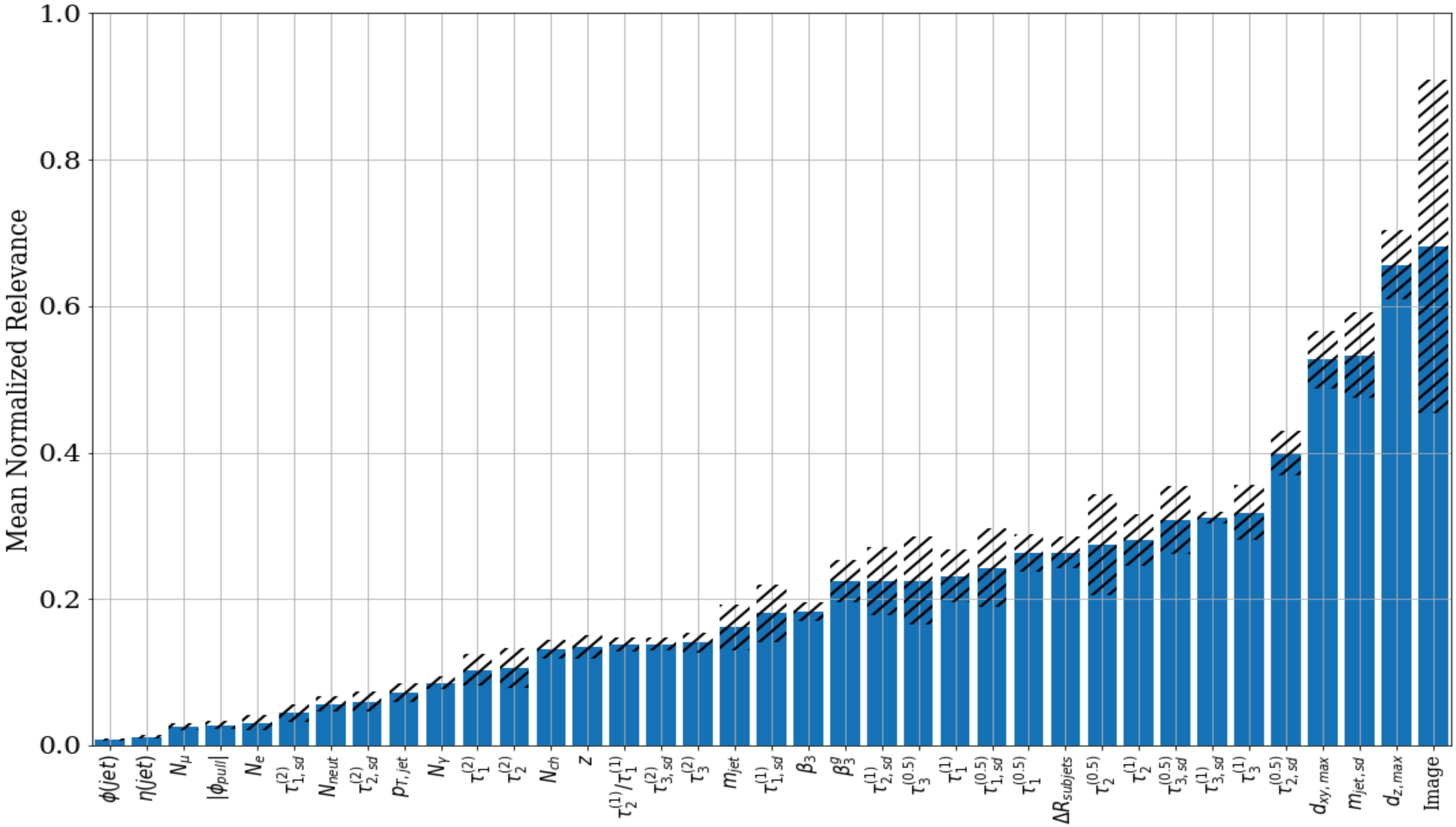
Darker markers: higher absolute relevance score

Decision boundaries: not as clear as for toy model



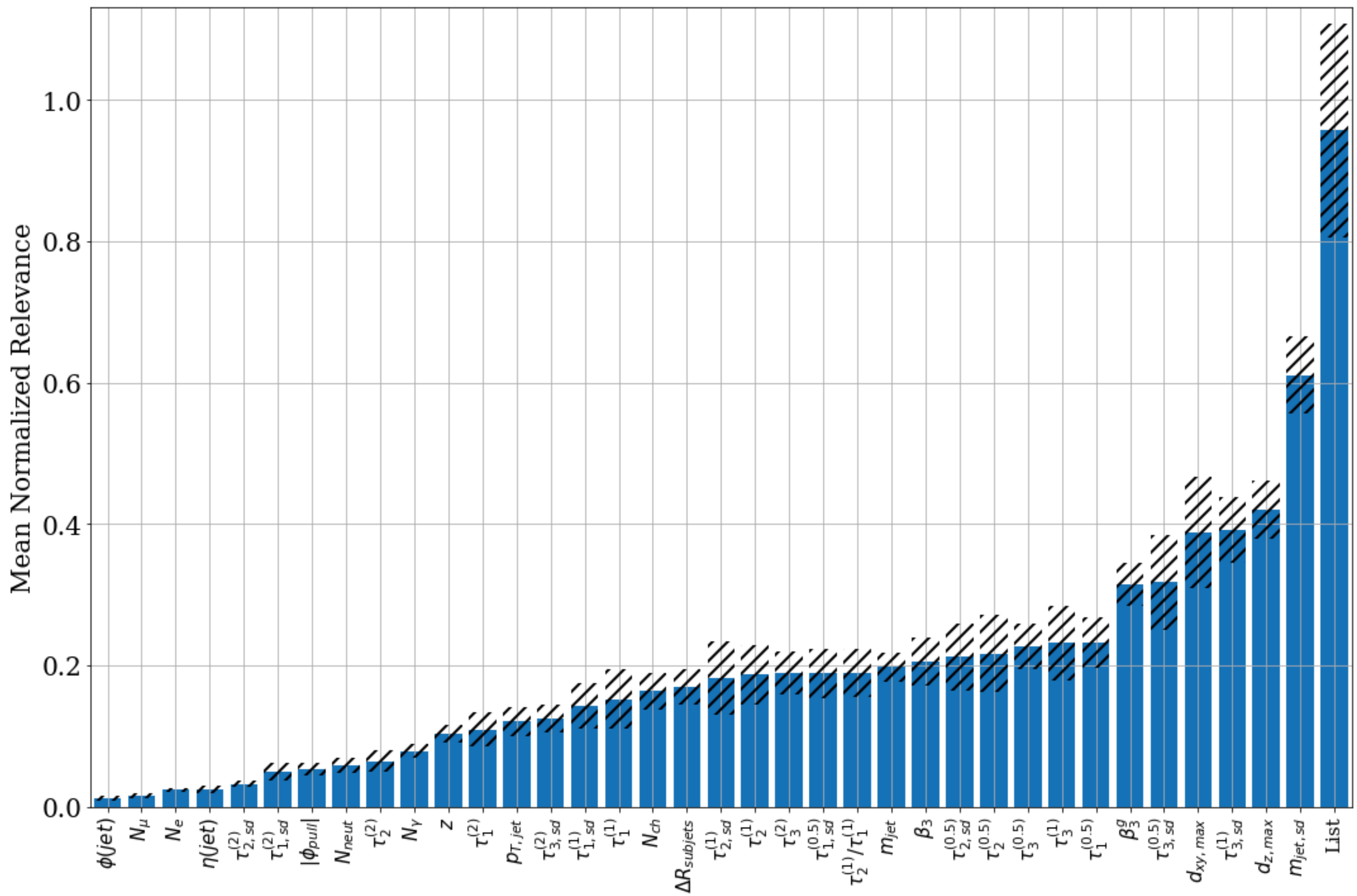
Pythia 2DCNN Results

Image and $d_{z,max}$: highest relevance, depending on the model

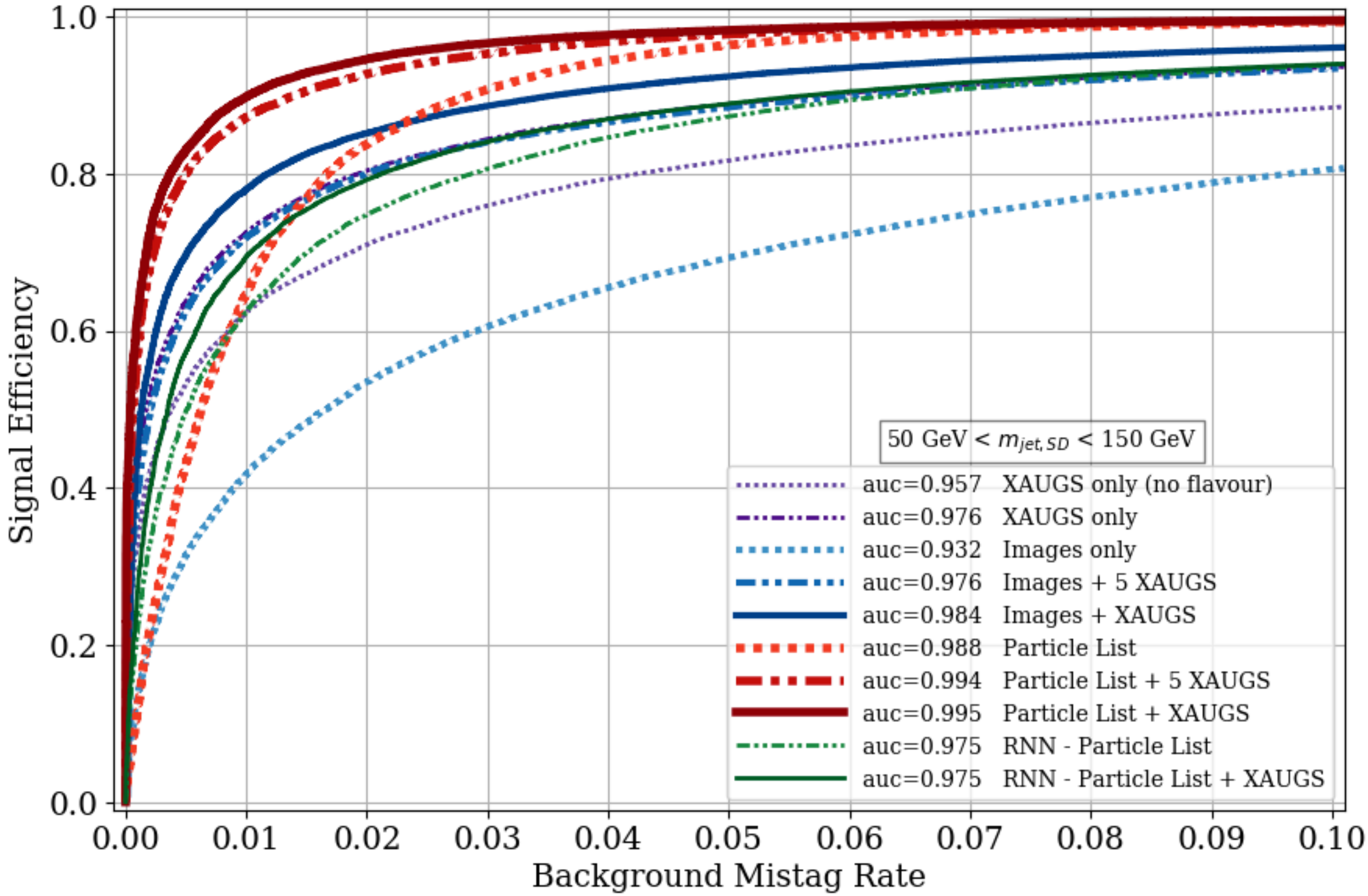


Pythia 1DCNN Results

Particle list: highest relevance for all models



Pythia Results: Model Comparisons



Conclusions

- Introduced novel method for ML tagger explainability: LRP + expert augmented variables
  - Help explain network decisions, and relevant subspaces

- **XAUGs**
  - Can boost classification performance
  - Can entirely capture relevant information of lower-level networks

- **XAUGs + LRP**
  - Can be used to reduce list of network inputs

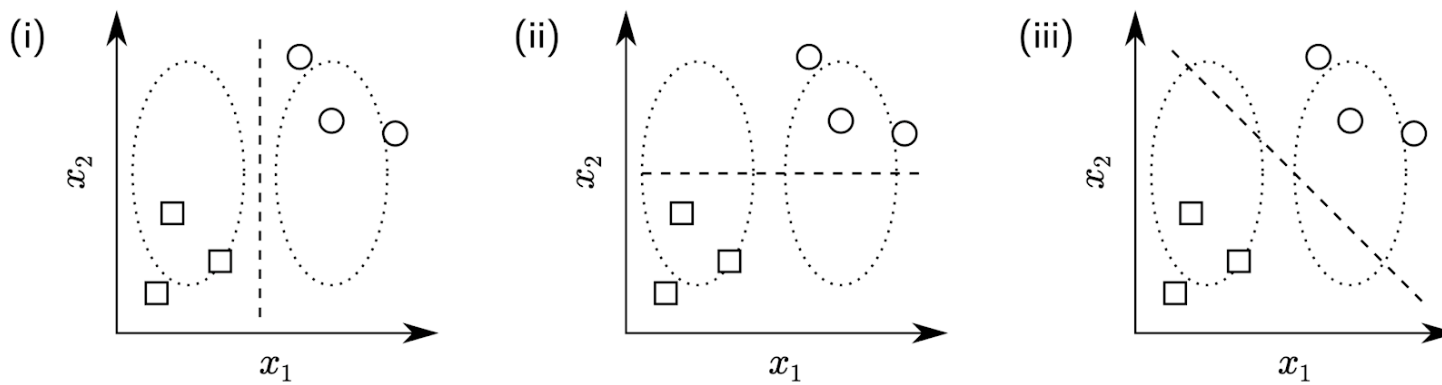
ADDITIONAL MATERIAL

Introduction

ML Black Box

Classification: Castle

- No explanation for the prediction.
- Predictions supported by meaningful patterns in data.
- Model should be able to explain itself → highlight features that support the prediction.



G. Montavon et al.

LRP Propagation Rules

- LRP-z
  - Redistributes the relevance in proportion to the contributions to the neuron activation.
  - Gradient X Input → Noisy

$$R_j = \sum_k \frac{a_j w_{jk}}{\sum_{0,j} a_j w_{jk}} R_k$$

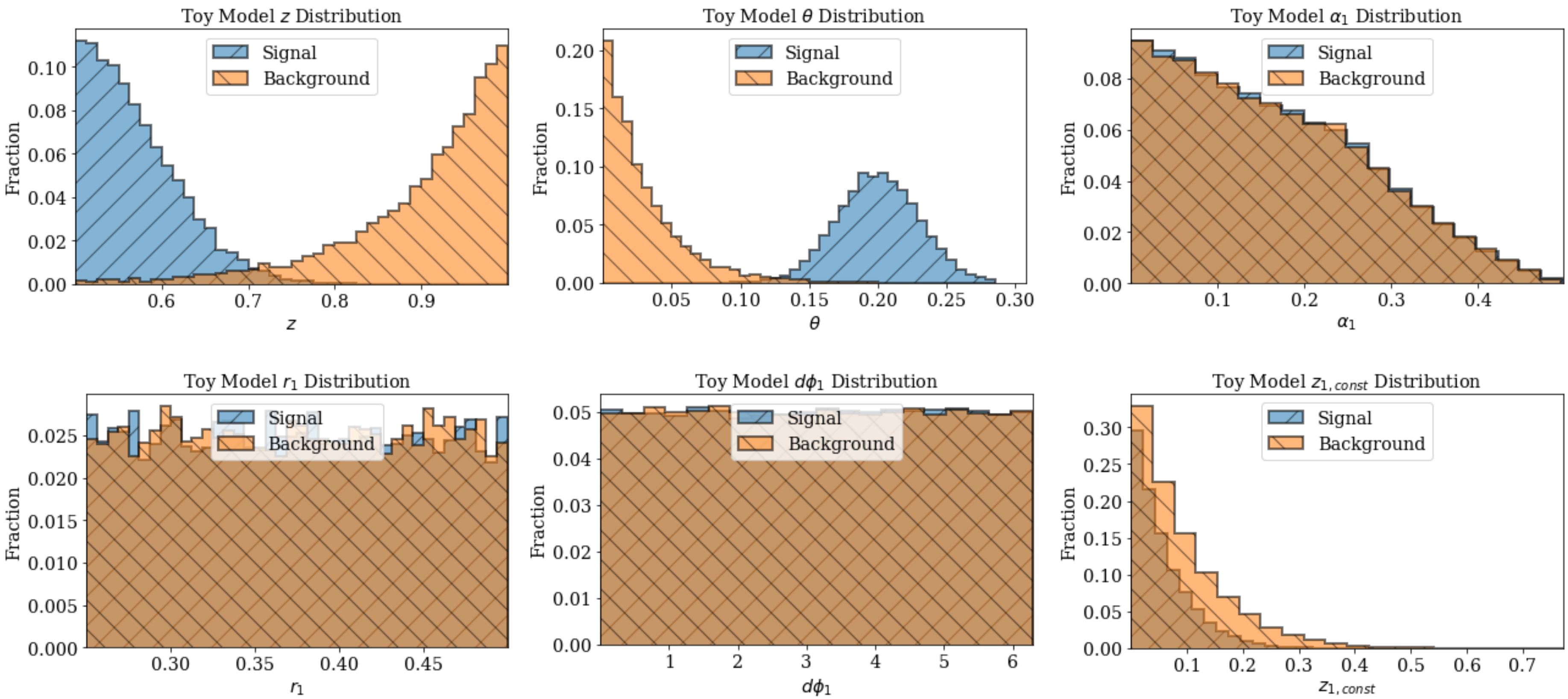
- LRP-ϵ
  - ϵ absorbs some relevance for weak and/or contradictory contributions.
  - For large ϵ only salient explanation factors survive the absorption → Less Noisy
  - Used in our networks' dense layers

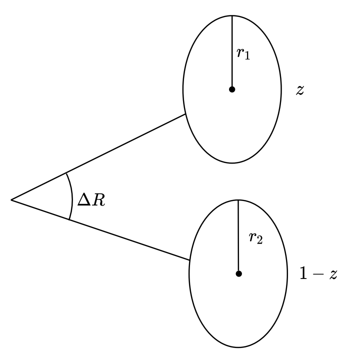
$$R_j = \sum_k \frac{a_j w_{jk}}{\epsilon + \sum_{0,j} a_j w_{jk}} R_k$$

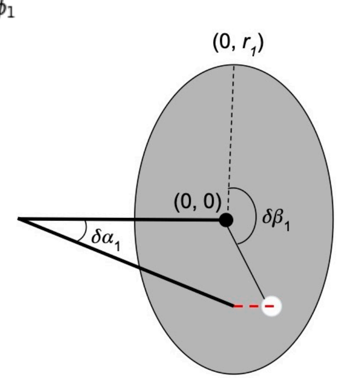
- LRP-$\alpha_1\beta_0$
  - Limiting effect on how large positive and negative relevance can grow → Stable Explanations
  - Used in our networks' convolution layers

$$R_j = \sum_k \left(\alpha \frac{(a_j w_{jk})^+}{\sum_{0,j} (a_j w_{jk})^+} - \beta \frac{(a_j w_{jk})^-}{\sum_{0,j} (a_j w_{jk})^-} \right) R_k$$

Toy Model Inputs





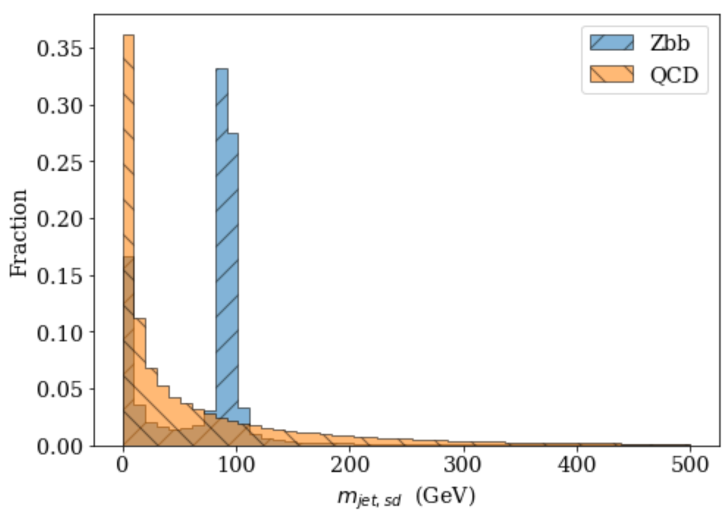


Particle List Inputs

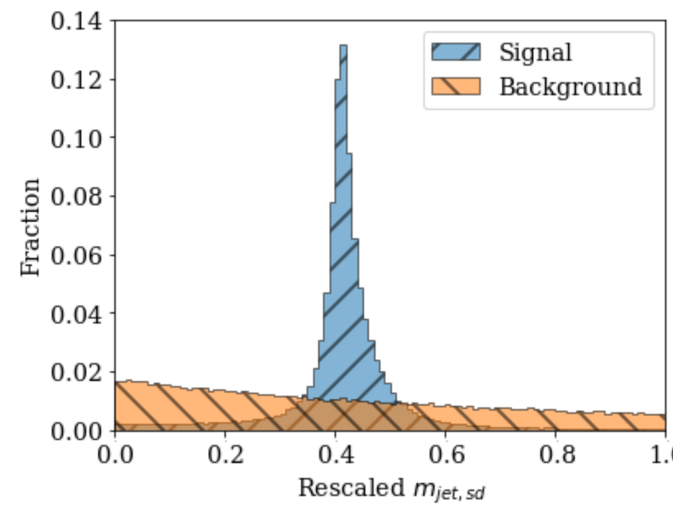
Variable
$log(p_T)$
$log(p_T/p_{T_{jet}})$
$log(E)$
$\|\eta\|$
$\Delta\phi(jet)$
$\Delta\eta(jet)$
$\Delta R(jet)$
$\Delta R(subjet1)$
$\Delta R(subjet2)$
Charge q
isMuon
isElectron
isPhoton
isChargedHadron
isNeutralHadron
d_{xy}
d_z

Pythia Model Preprocessing

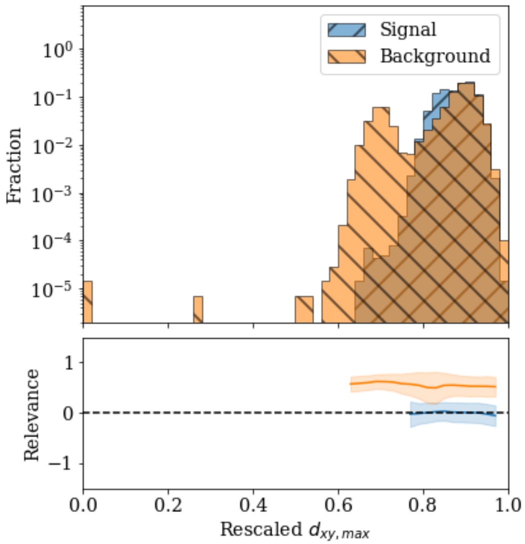
1. Cut on softdrop mass:
   keep jets with m_{SD} 50-150 GeV
2. Numerical rescaling
   1. Rebin outliers to mean + 3(std) and mean - 3(std)
   2. Input distributions are then rescaled from 0 to 1: $\frac{x - x_{min}}{x_{min} - x_{max}}$

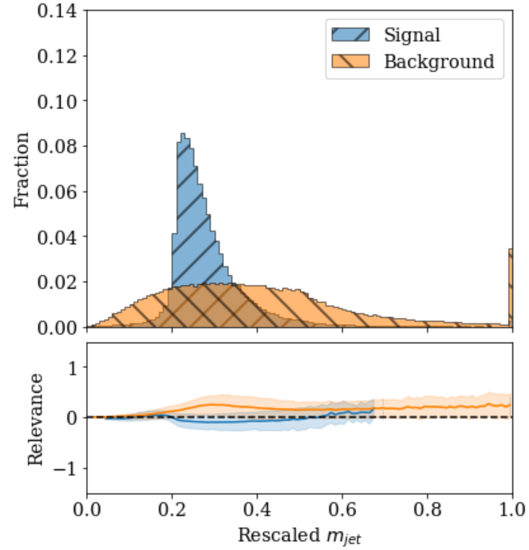


Mass cut + rescaling

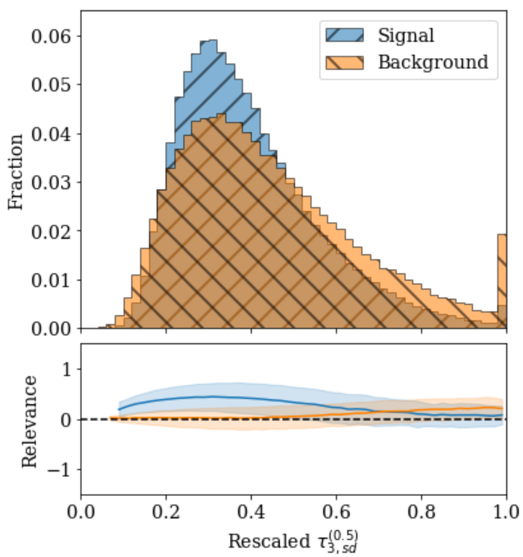


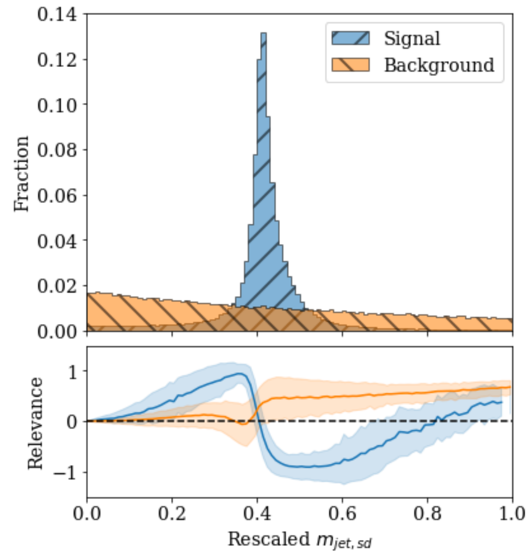
Pythia Model 1D LRP Plots



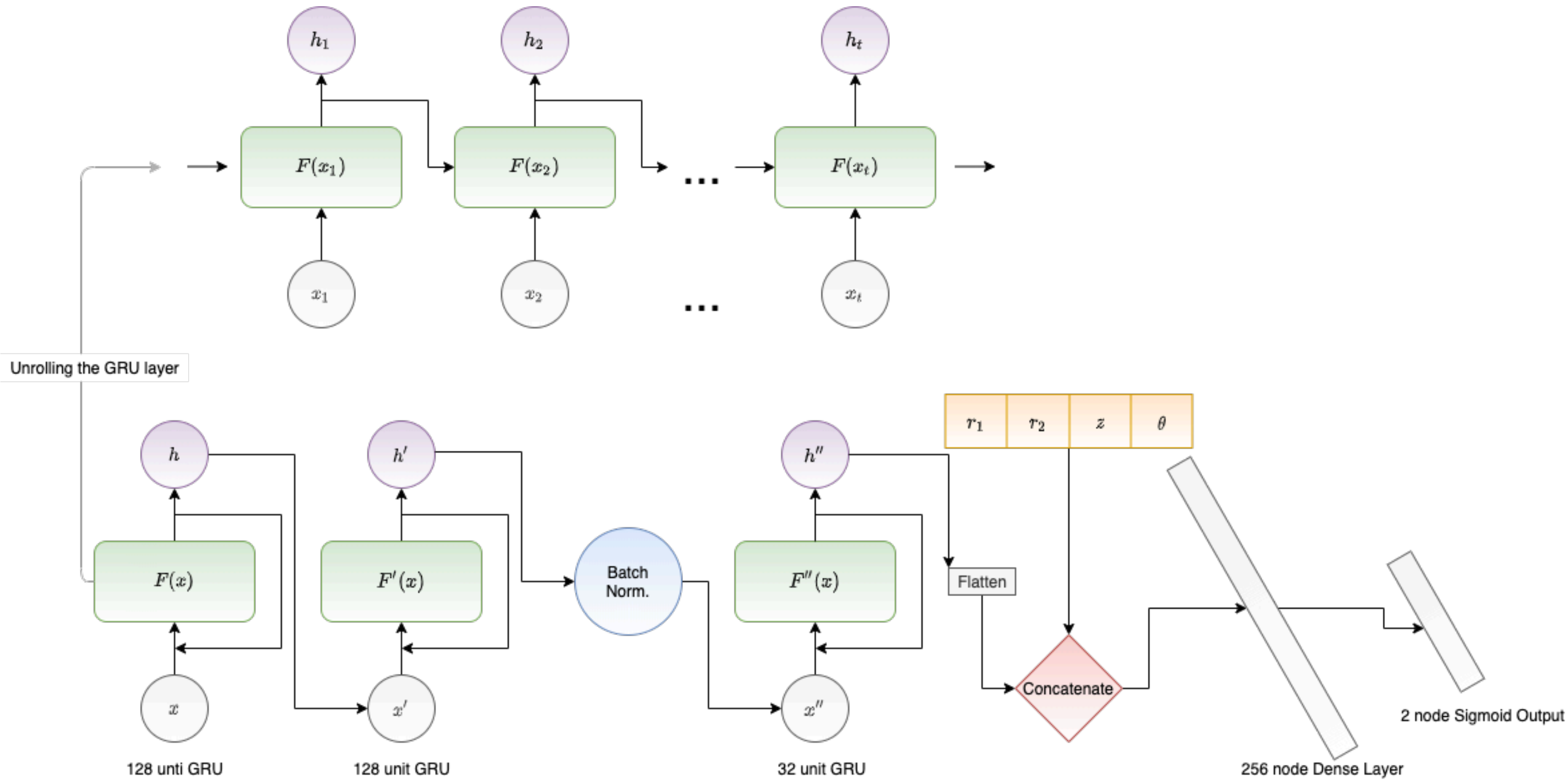


Profiles don't show clear decision boundary - need higher dimensional plots





RNN



h_1
h_2
h_t
$F(x_1)$
$F(x_2)$
...
$F(x_t)$
x_1
x_2
...
x_t
Unrolling the GRU layer
r_1
r_2
z
θ
h
h'
h''
$F(x)$
$F'(x)$
Batch Norm.
$F''(x)$
Flatten
Concatenate
x
x'
x''
128 unti GRU
128 unit GRU
32 unit GRU
2 node Sigmoid Output
256 node Dense Layer