# Towards the XAI in HEP


<u>Jin Choi</u>


Dec 1. 2023

For HEP and ML workshop

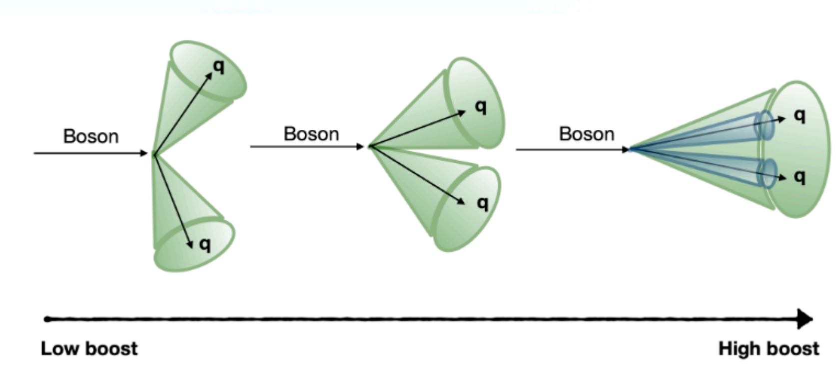
# Introduction

# Introduction

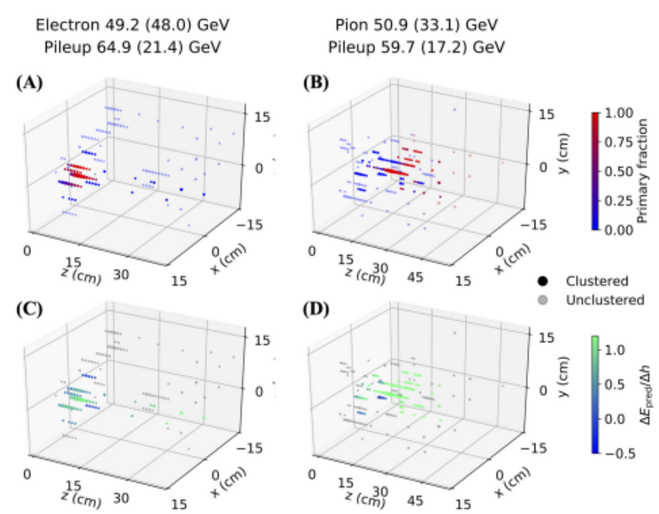
## What is XAI? Why we need it?

**Explosive development of AI in HEP**

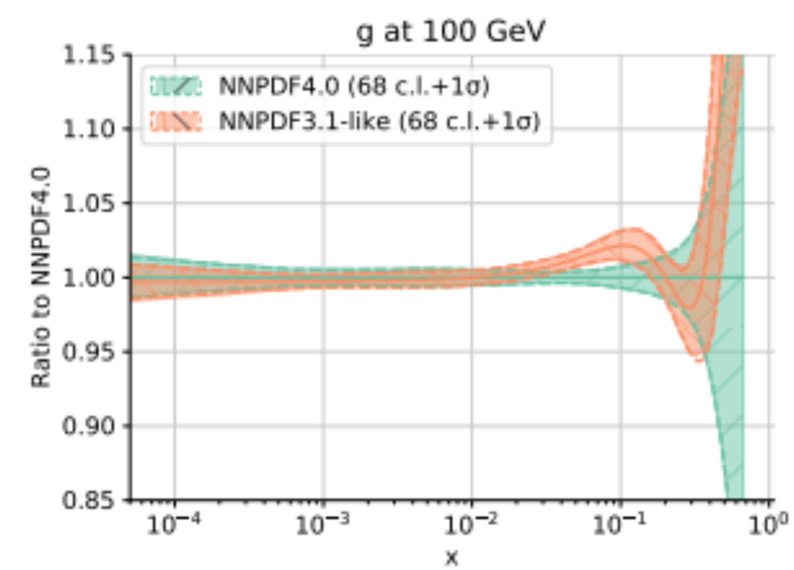
- Neural Networks as parametrized fitting functions - modeling parton distributions and showers
- Jet tagging to identify boosted, heavy quarks
- Implementing pre-trained DNN models on FPGA devices - online reconstruction / triggering
- End to end simulations using generative models
- And more…




[boosted jet tagging]




Particle Identification




[nnpdf 4.0]

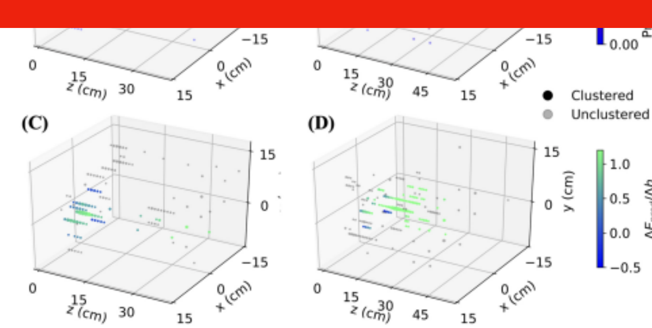
# Introduction

## What is XAI? Why we need it?

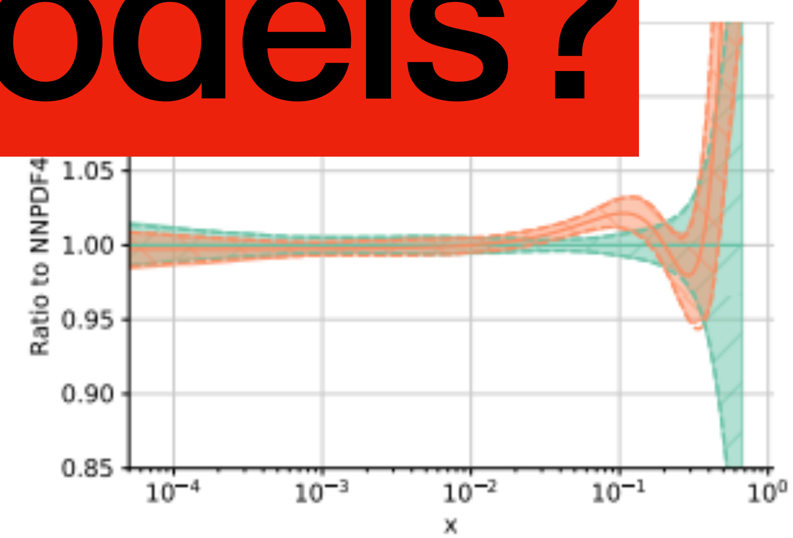
- **Explosive development of AI in HEP**
- Neural Networks as parametrized fitting functions - modeling parton distributions and showers
- Jet t
- Imp ... ring
- End
- And

**But how much we understand about these models?**


[boosted jet tagging]




Particle Identification

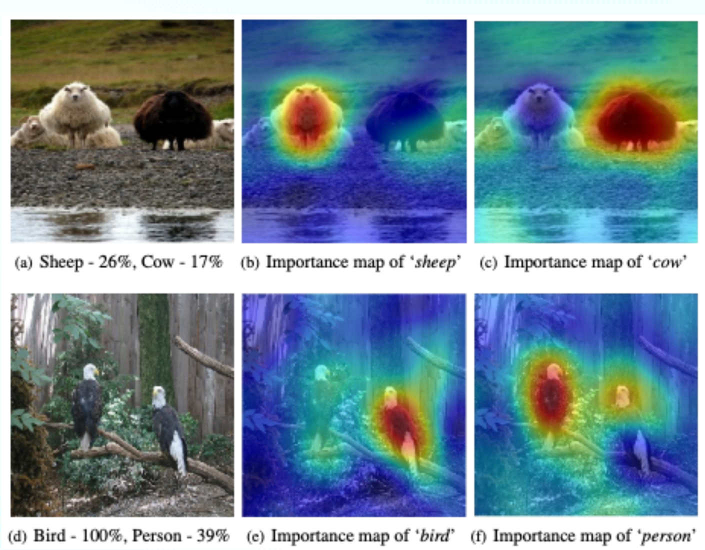



[nnpdf 4.0]

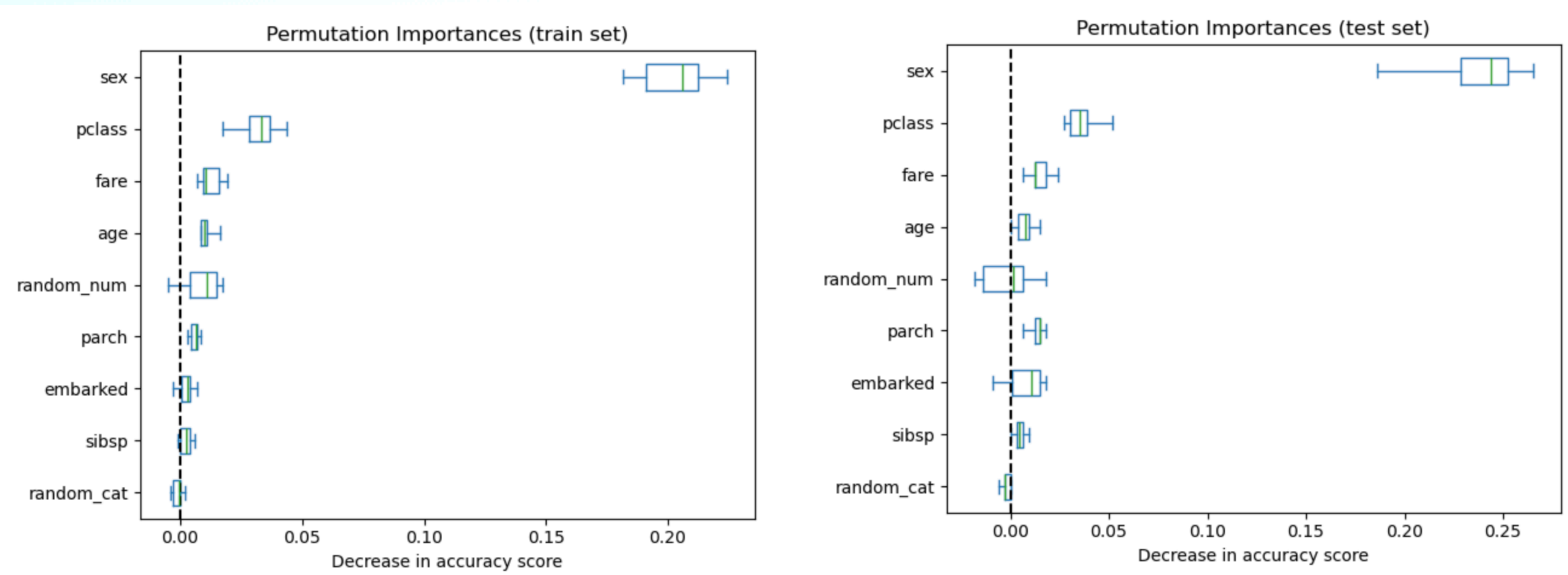
# Introduction

## What is XAI? Why we need it?

- **Correlation vs. Causality** [A Survey]
- The definition of the XAI is still controversial, but mostly concerns:
- **Transparency**: the model should be able to create a human-understandable justification
- **Trustability**: judgement should be based on the knowledge and available explanations
- **Bias understanding** and **Fairness**: XAI helps mitigate biases either from inputs or architectures




[Attributing pixels using RISE method]




[Bias testing using permutation importance]

# Introduction

## What is XAI? Why we need it?

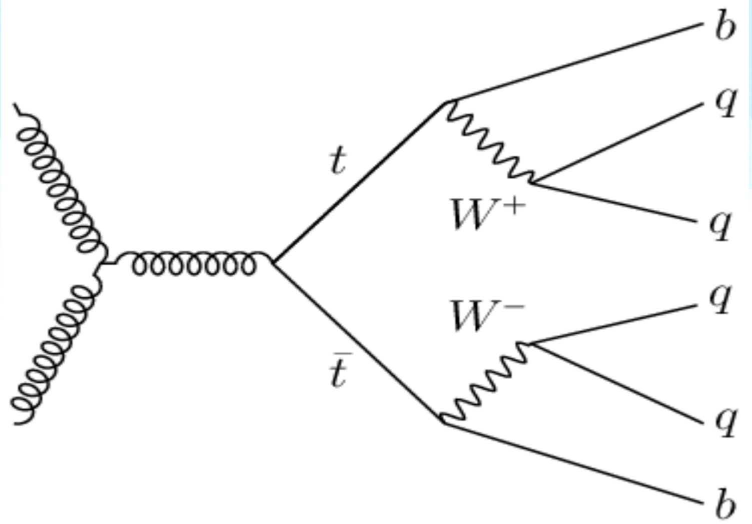
- **Correlation vs. Causality** [A Survey]
  - The definition of the XAI is still controversial, but mostly concerns:
  - **Transparency**: the model should be able to create a human-understandable justification
  - **Trustability**: judgement should be based on the knowledge and available explanations
  - **Bias understanding** and **Fairness**: XAI helps mitigate biases either from inputs or architectures
- **Categorization of the XAI**
  - **Local or Global**: Where is the XAI method focusing on?
  - **Methodology**: What is the algorithmic approach? Input data instances? Model gradients?
  - **Usage**: How is the XAI method developed? Is it intrinsic? Is it model-dependent?
- **In this talk, I will cover the XAI methods and applications for**
  - Simple models with tabular datasets - Decision Tree and DNN based explanation
  - Complex models with graph datasets -
    Graph neural networks and its explanation, surrogated models

# Simple Models

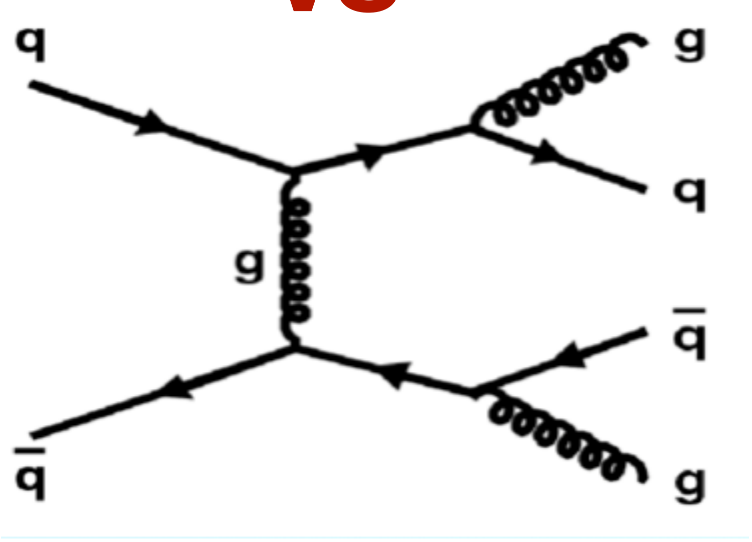
# Simple Models

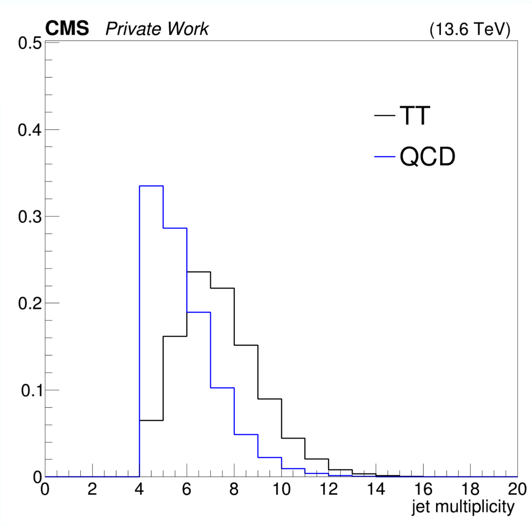
## Example

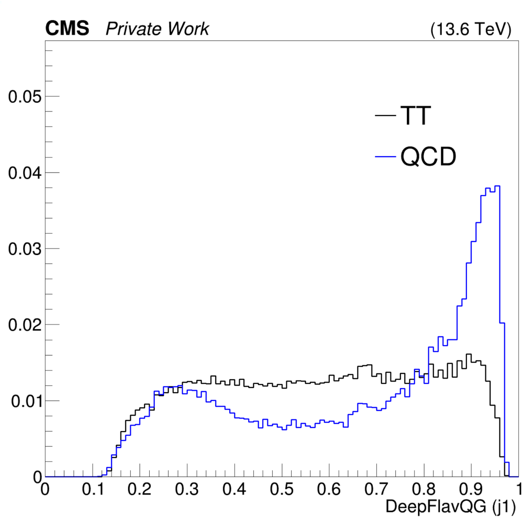
- **Classifying TT hadronic vs. QCD multijets**
- Number of jets / bjets expected to have good discrimination power
- Used features:
  - 4 momentum
  - DeepJet scores for light vs. b / q vs. g
  - EM / Hadron / Muon Energy fractions
  
  Up to 4th leading pt jets
  - Jet multiplicity, HT, average $\Delta R$ between jets
- 300K evts for each TT / QCD, 6:1:3 for train:valid:test split

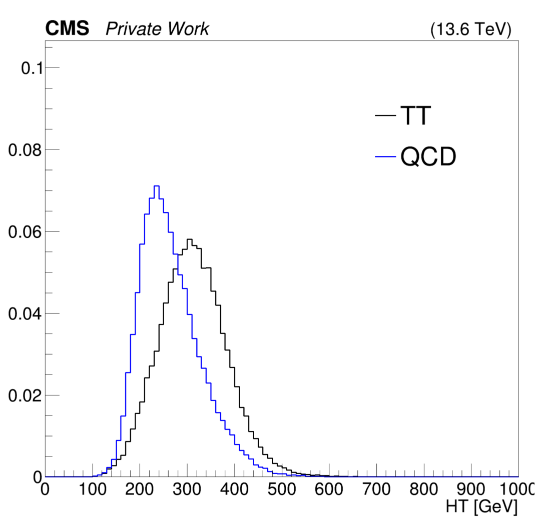



VS



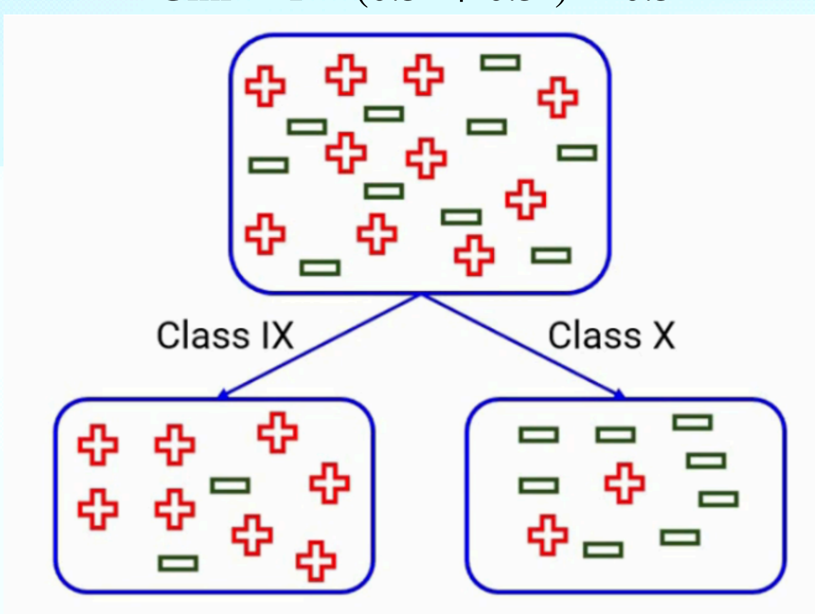
# Simple Models

## Decision Trees

- **First explainable model**
- Decision trees are intrinsically explainable - split of the nodes are based on "impurity"
- Importance of each feature can be mapped by "decrease of impurity after split"

**Example) Gini Index**

$$\mathrm{Gini} = 1 - (0.5^2 + 0.5^2) = 0.5$$




$$\mathrm{Gini} = 1 - (0.8^2 + 0.2^2) = 0.32 \quad \mathrm{Gini} = 1 - (0.2^2 + 0.8^2) = 0.32$$

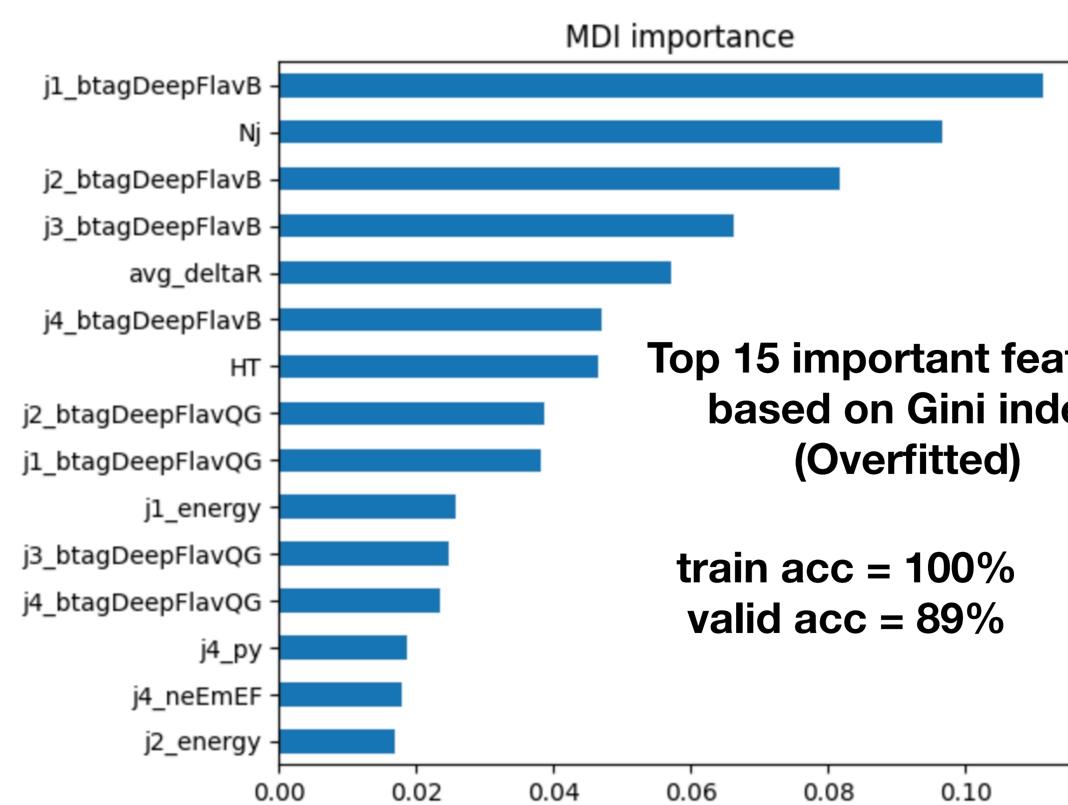
$$\mathrm{Gini_s} = 0.5 \times 0.32 + 0.5 \times 0.32 = 0.32$$

**0.18 decreased by this split!**

**Can be mapped to feature importance**

$$\mathrm{Gini(T)} = 1 - \sum_j p_j^2 \text{ with } j \in C$$

$$\mathrm{Gini_s(T)} = \frac{N_1}{N}\mathrm{Gini(T_1)} + \frac{N_2}{N}\mathrm{Gini(T_2)}$$



# Simple Models

## Decision Trees

- **Permutation Importance**
- Shuffle one of the features from the dataset and observe the decrease in output metric

| Height at age 20 (cm) | Height at age 10 (cm) | ... | Socks owned at age 10 |
|---|---|---|---|
| 182 | 155 | ... | 20 |
| 175 | 147 | ... | 10 |
| ... | ... | ... | ... |
| 156 | 142 | ... | 8 |
| 153 | 130 | ... | 24 |

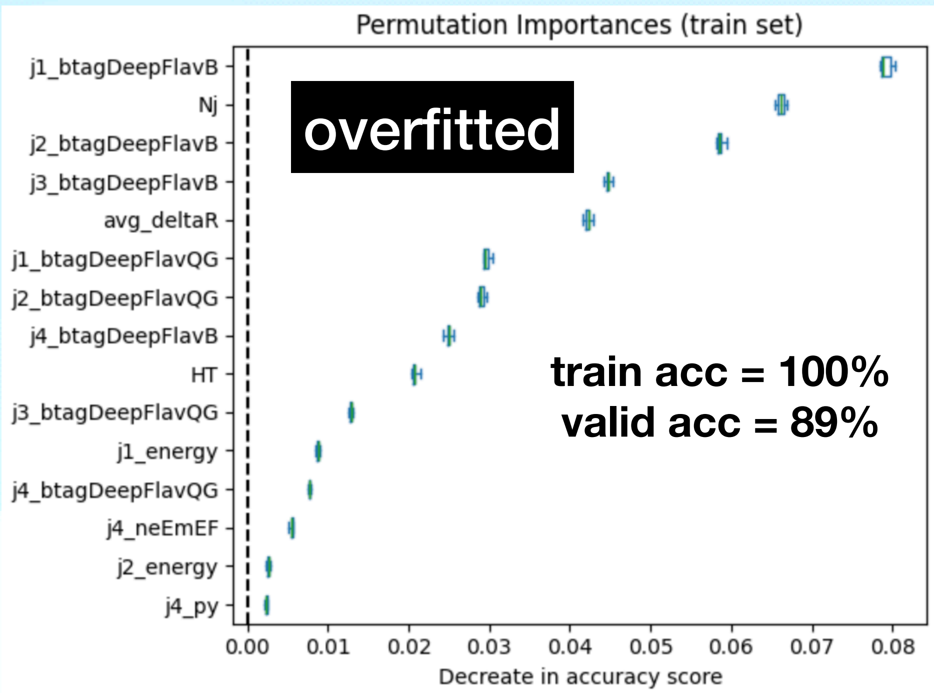
**Large decrease on important features!**

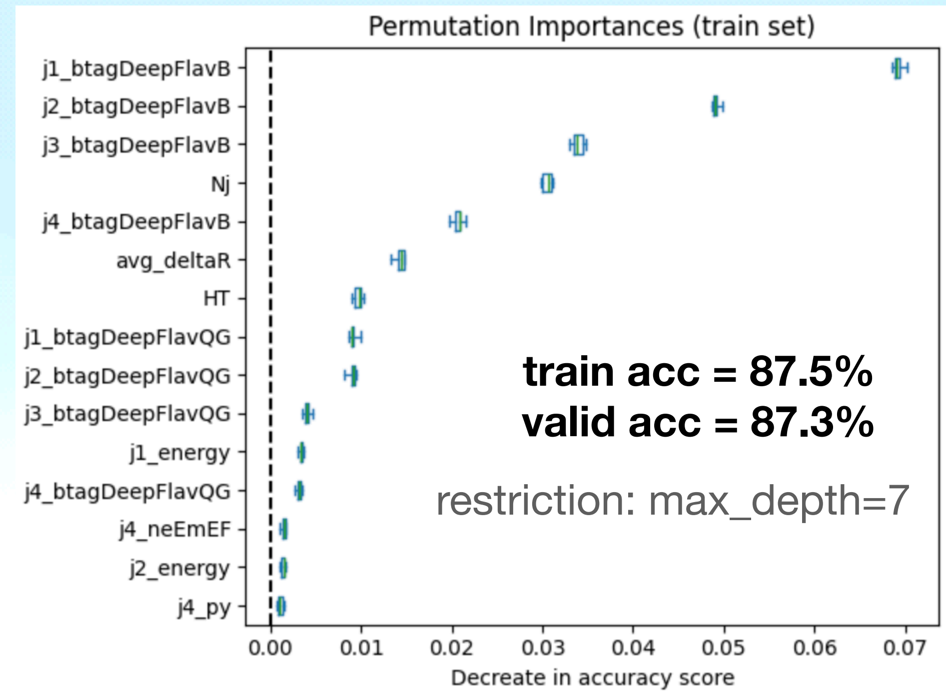
- Model agnostic - feature importances are measured based on the dataset
- Correlation between features naturally considered by shuffling
- **Controlling overfitting using feature importances**
- Overfitting problem occurs because of learning **bias in the train set**
- Result in different feature importances in the train and the validation(or test) set

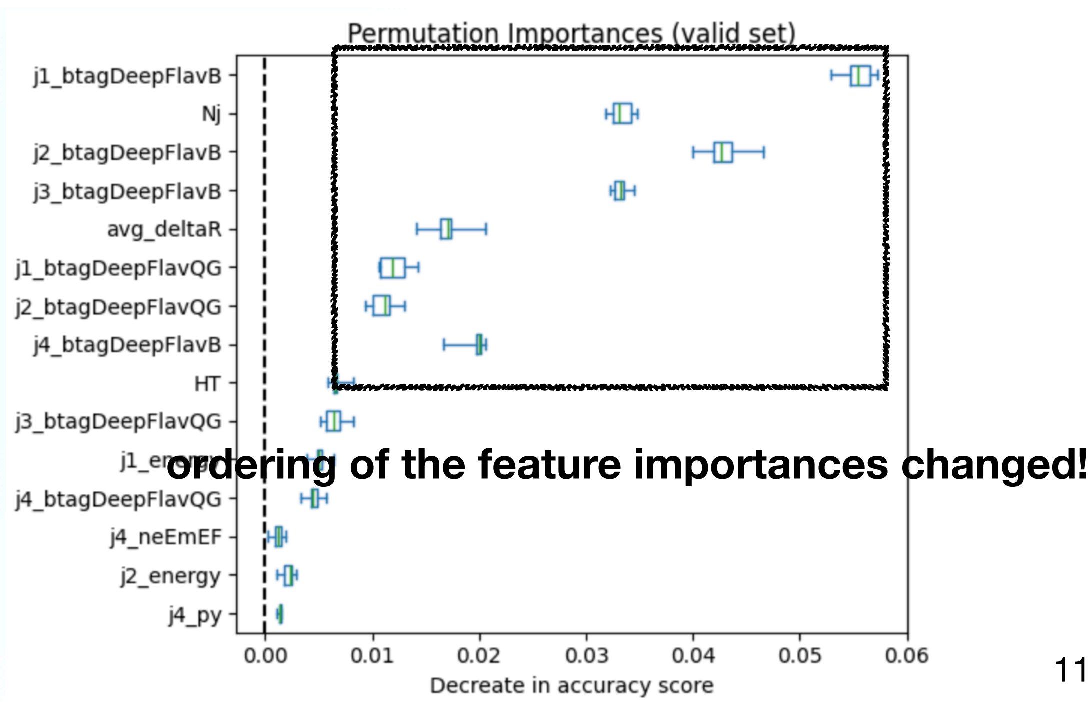
# Simple Models

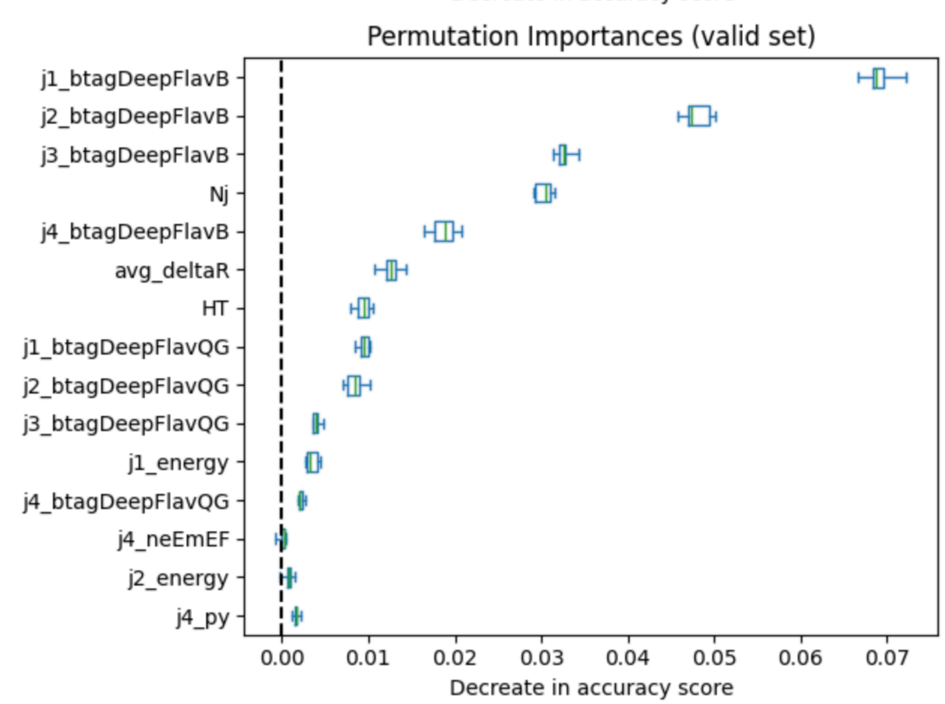
## Decision Trees

- **Controlling overfitting based on permutation importances**



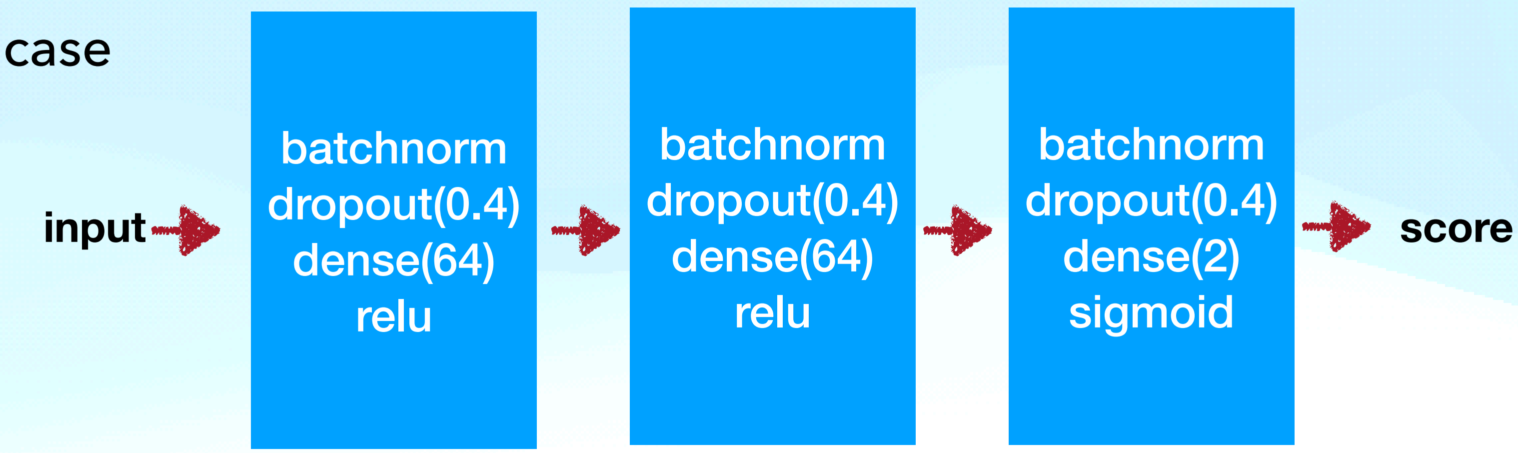



# Simple Models

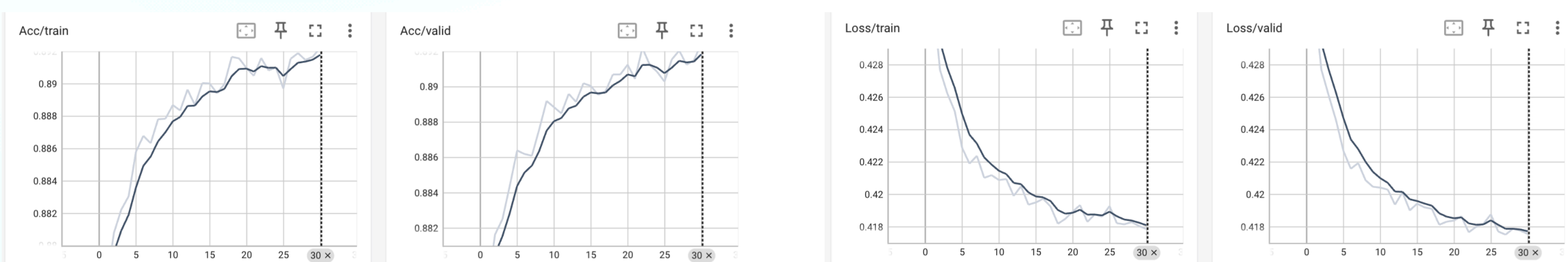
## Deep Neural Networks

### Training

- Used the same inputs as in the BDT case
- Used Adam optimizer, learning for 30 epochs




### Results




- ~ 90% accuracy for both train / valid set
- No specific behaviors to judge overfitting

# Simple Models

## Deep Neural Networks

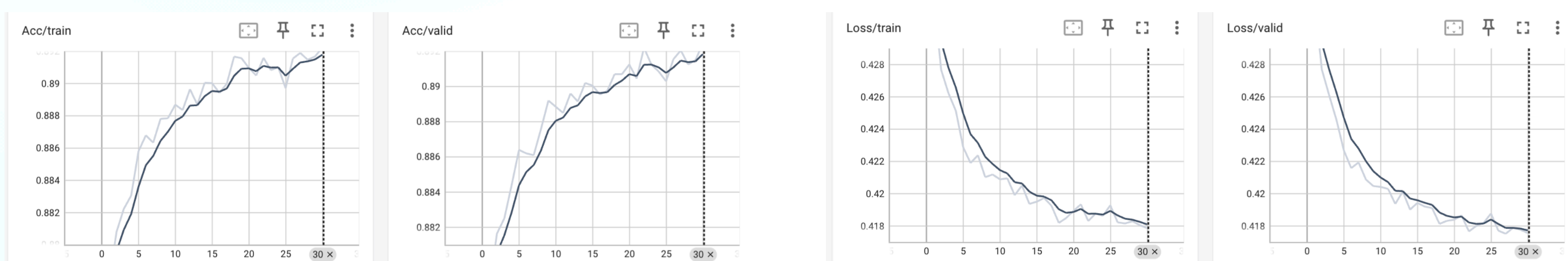
### Training

- Used the same inputs as in the BDT case
- Used Adam optimizer, learning for 30 epochs

input → Black Box? → score

**How can we make explanatory metrics from the DNN?**

### Results




- ~ 90% accuracy for both train / valid set
- No specific behaviors to judge overfitting

# Simple Models

## Deep Neural Networks

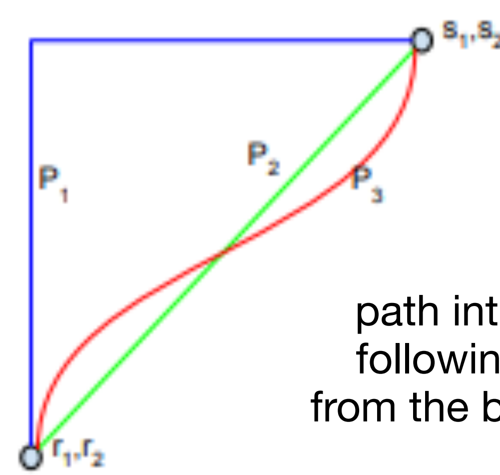
**Attributions**

- Various efforts were made to explain the relationship between input features and outputs
- DNNs are differentiable! Calculation of attributions rely on instantaneous / finite gradients of the models
- e.g.) DeConvNet, Guided back-propagation, DeepLift, LRP, Integrated Gradients…

**Axiomatic approach**

- Attributions should satisfy - **Sensitivity** and **Implementation Invariance**
- Here we focus on **integrated gradients** which satisfy the two axioms:

$$\mathrm{IG}_i(x) \equiv (x_i - x_i') \times \int_{\alpha=0}^{1} \frac{\partial F(x' + \alpha(x - x'))}{\partial x_i} d\alpha$$

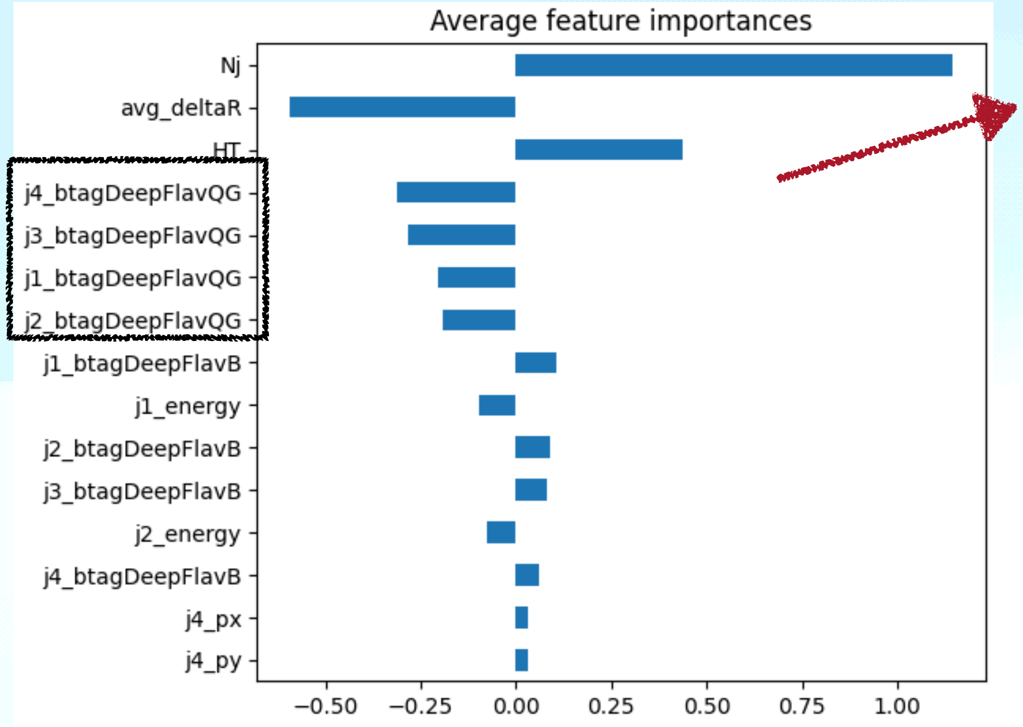



path integral of gradients following the straight line from the baseline x' to input x

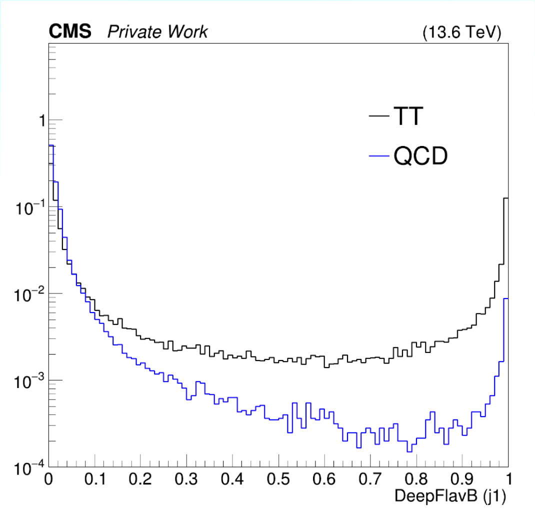
# Simple Models

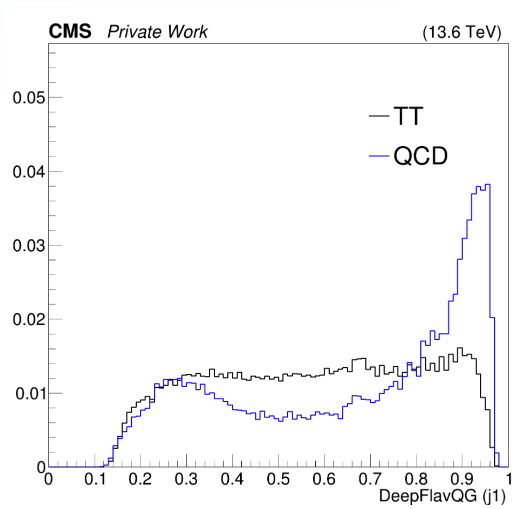
## Deep Neural Networks

- **Extracting Feature Importances from DNN models**
- Calculation of the integrated gradients can be easily done using [captum]




a bit counterintuitive...
the model consider more in QvsG score than light vs b score

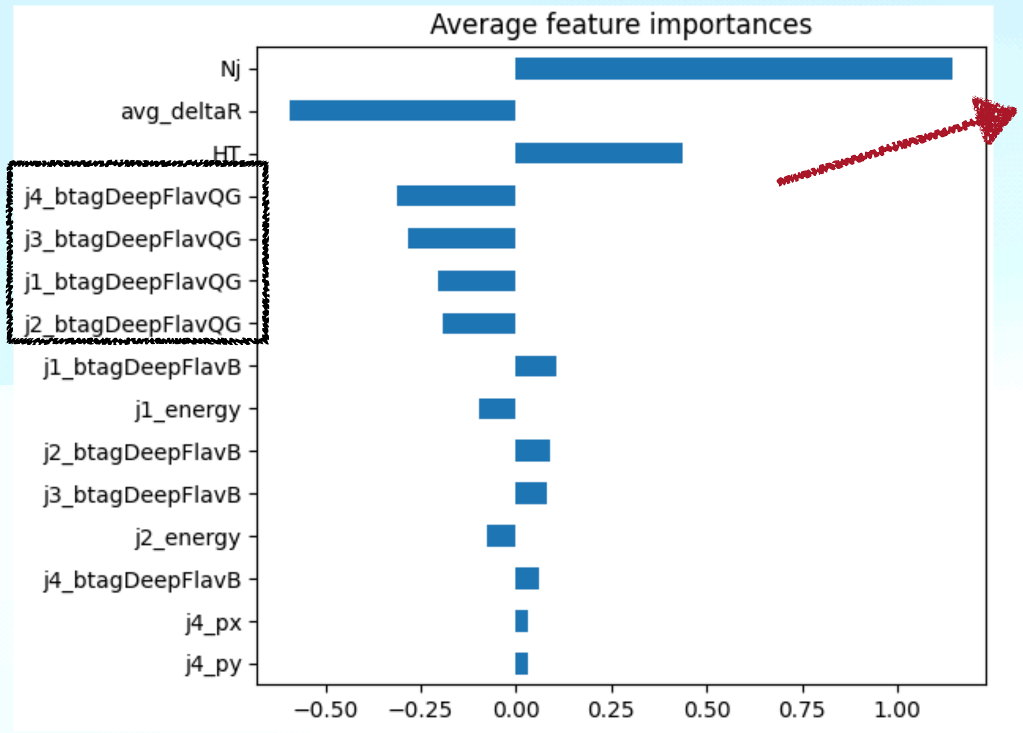


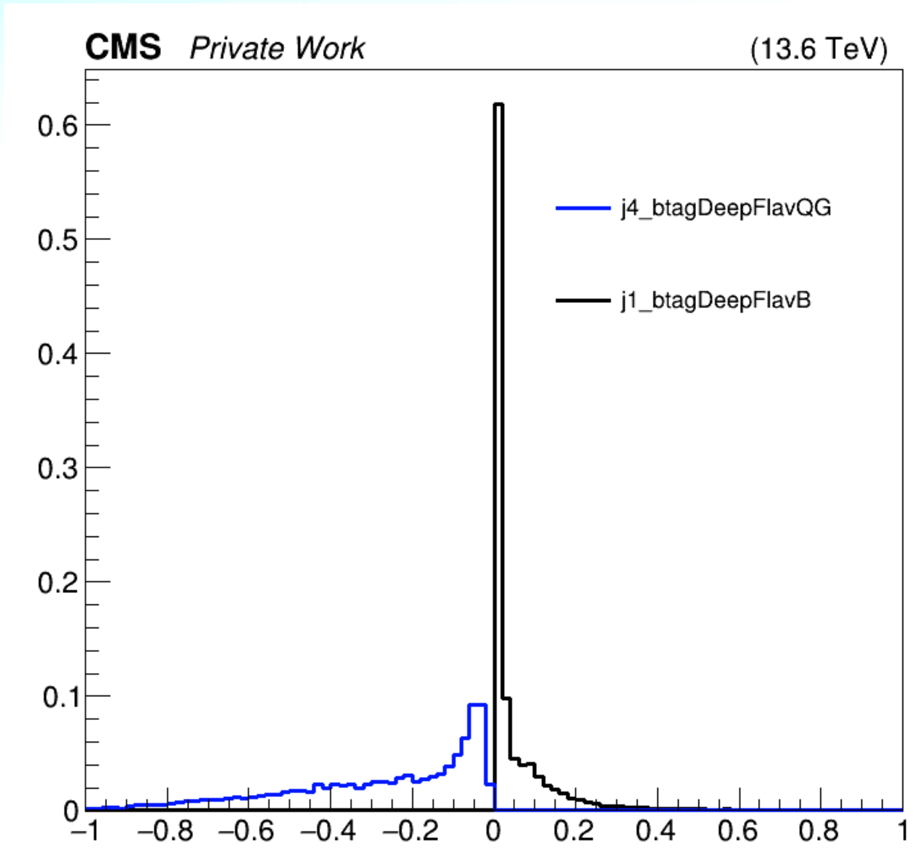
# Simple Models

## Deep Neural Networks

- **Extracting Feature Importances from DNN models**
- Calculation of the integrated gradients can be easily done using [captum]




a bit counterintuitive...
the model consider more in QvsG score than light vs b score



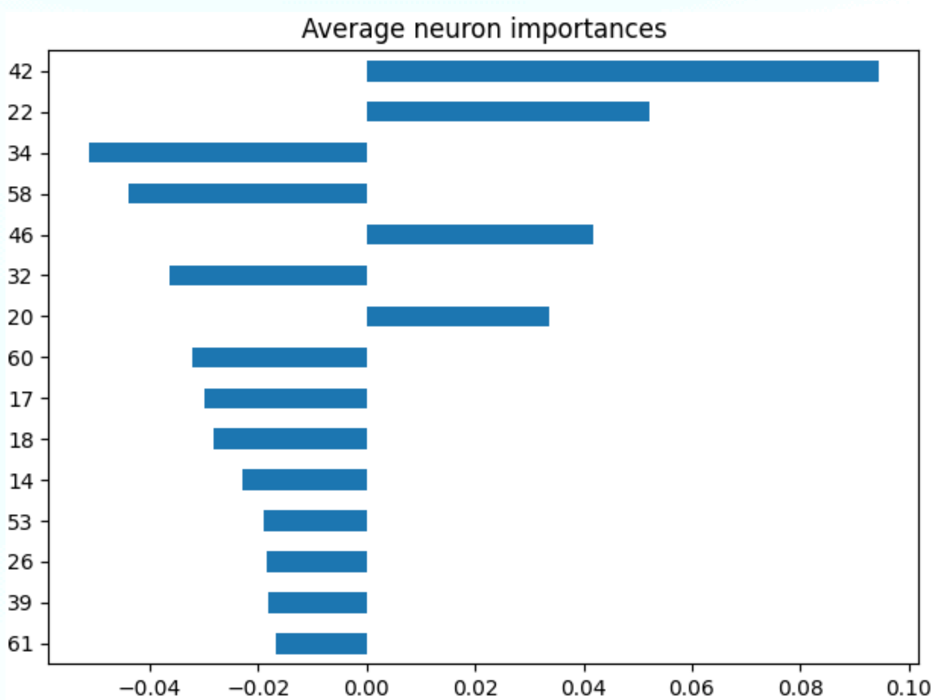

Most of the attributions for j1_btagDeepFlavB assigned to low values

# Simple Models

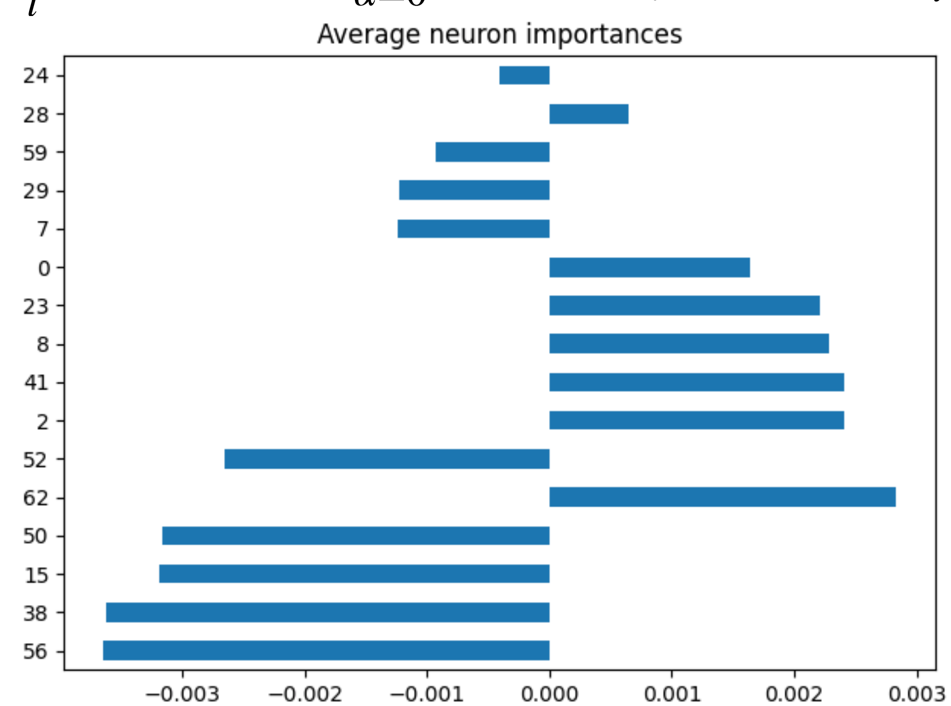
## Deep Neural Networks

**Conductance**

- Not only mapping importance from the features to the outputs, also contribution for each nodes are also possible via chain rules
- Conductance of neuron $y$: $\mathrm{Cond}_i^y(x) \equiv (x_i - x_i') \times \int_{\alpha=0}^{1} \frac{\partial F(x' + \alpha(x - x'))}{\partial y} \frac{\partial y}{\partial x_i} d\alpha$
- Total conductance of neuron $y$: $\mathrm{Cond}^y(x) \equiv \sum_i (x_i - x_i') \times \int_{\alpha=0}^{1} \frac{\partial F(x' + \alpha(x - x'))}{\partial y} \frac{\partial y}{\partial x_i} d\alpha$




Top 15 activated neurons



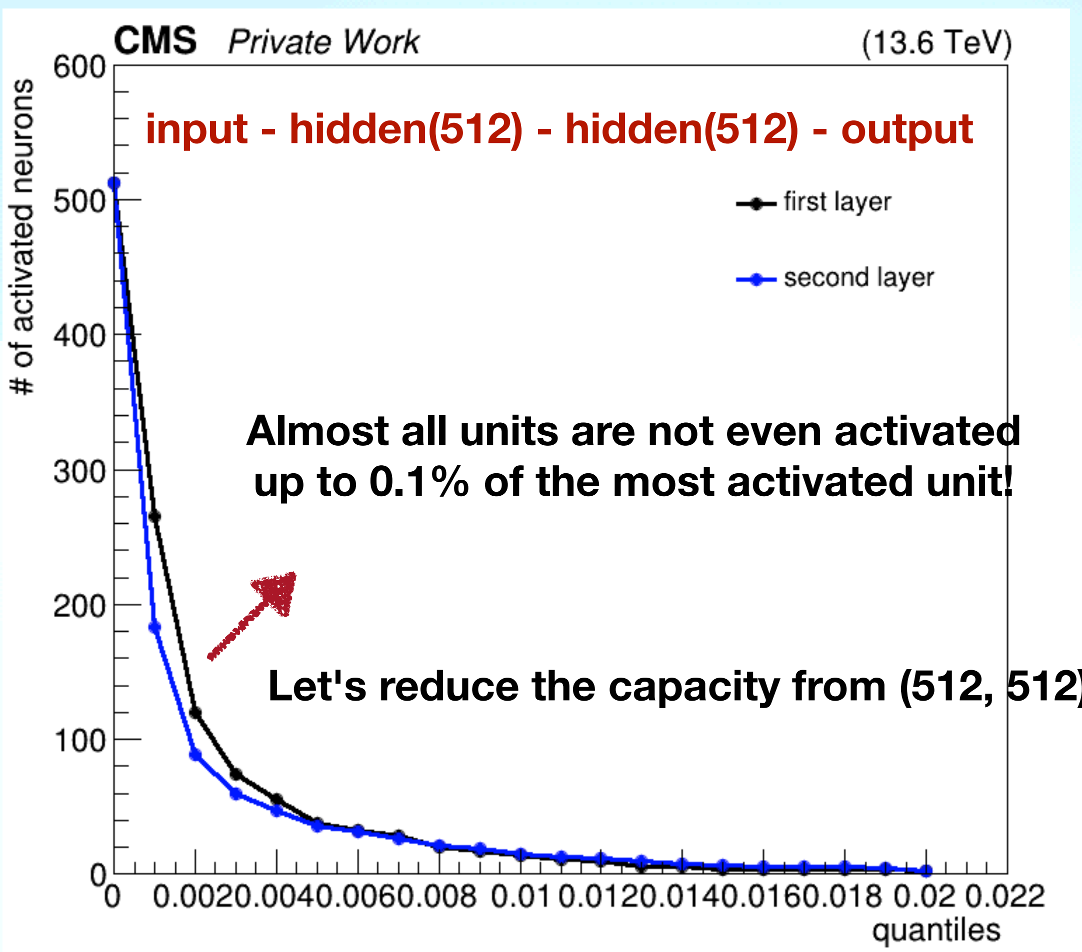

Bottom 15 activated neurons

# Simple Models

## Deep Neural Networks

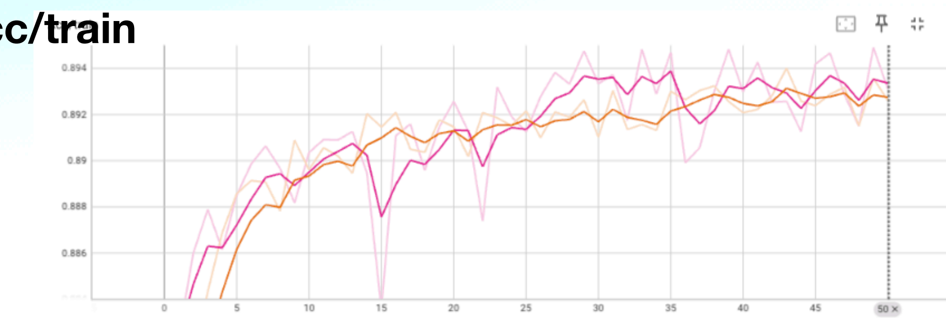
**Optimizing Model Capacity using Conductance**

- Most of the neurons are not activated if the model capacity is too large

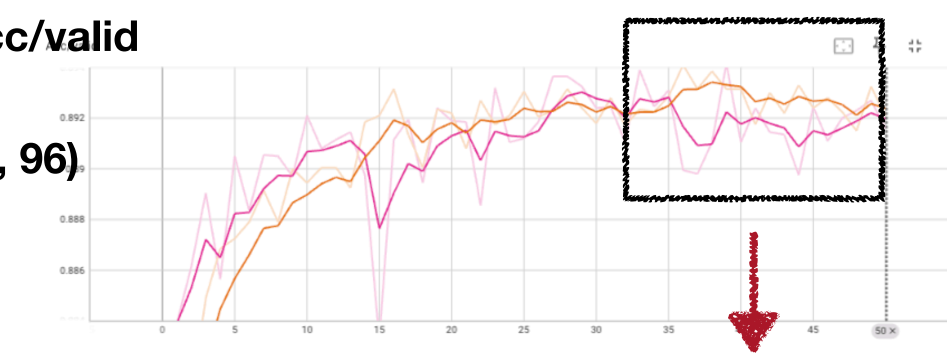



Let's reduce the capacity from (512, 512) to (128, 96)

Acc/train



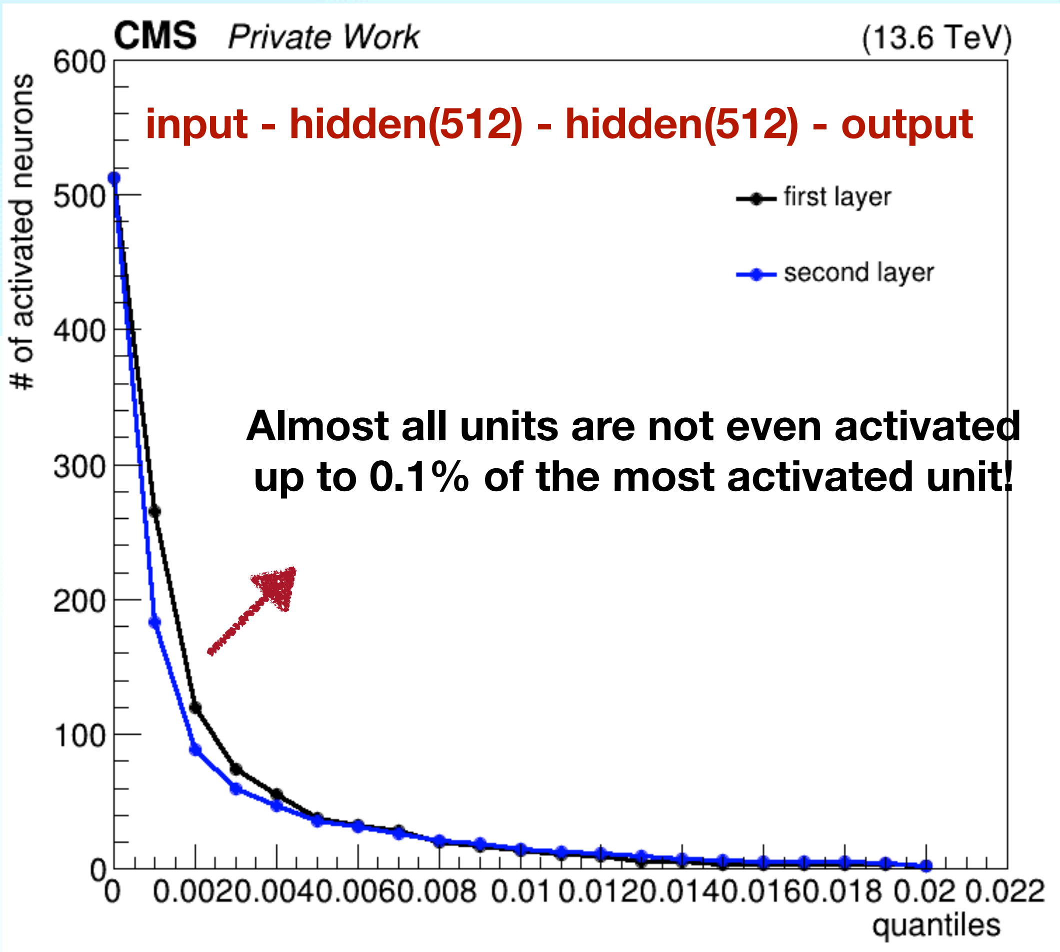
Acc/valid

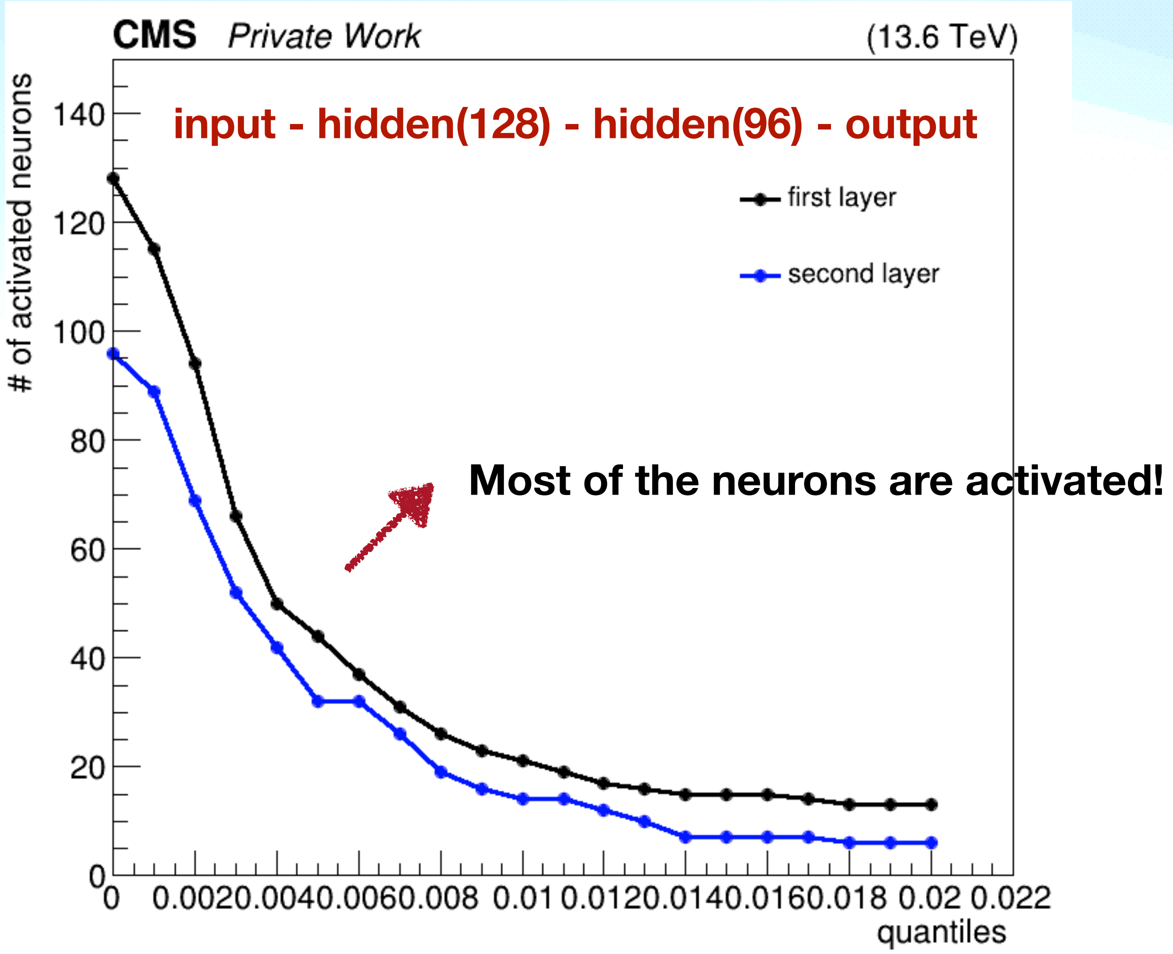


Overfitting behavior reduced

# Simple Models
## Deep Neural Networks

- **Optimizing Model Capacity using Conductance**
- We can see most of the neurons are not activated if the model capacity is too large



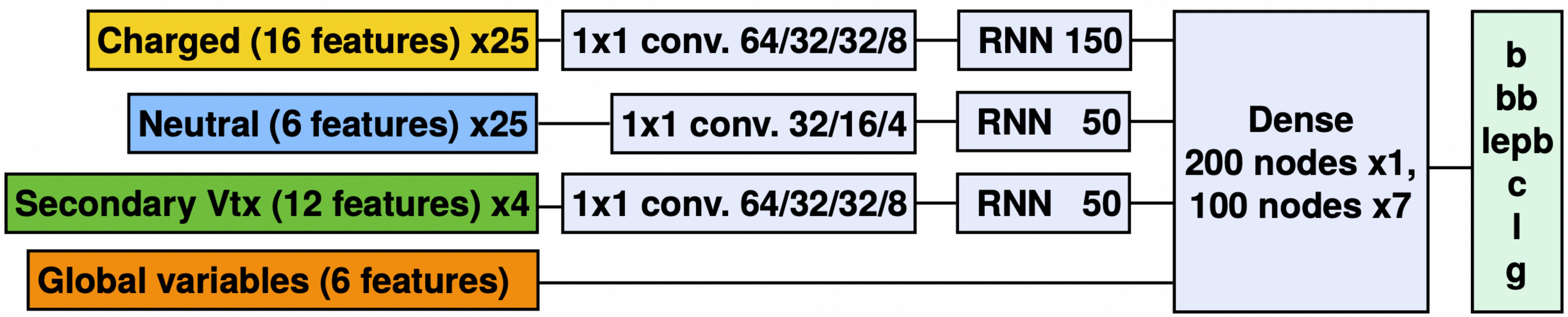



# Complex Models

# Complex Models

## Data Representation

- **What is the most natural representation for HEP events?**
- Ordered lists(tables) / Binary Trees
  → **Manually imposed ordering** might impair the performance
  → The length of the list is **fixed** but the no. of particles in each event is **flexible**



# Complex Models

## Data Representation

❖ **What is the most natural representation for HEP events?**

✓ Ordered lists(tables) / Binary Trees

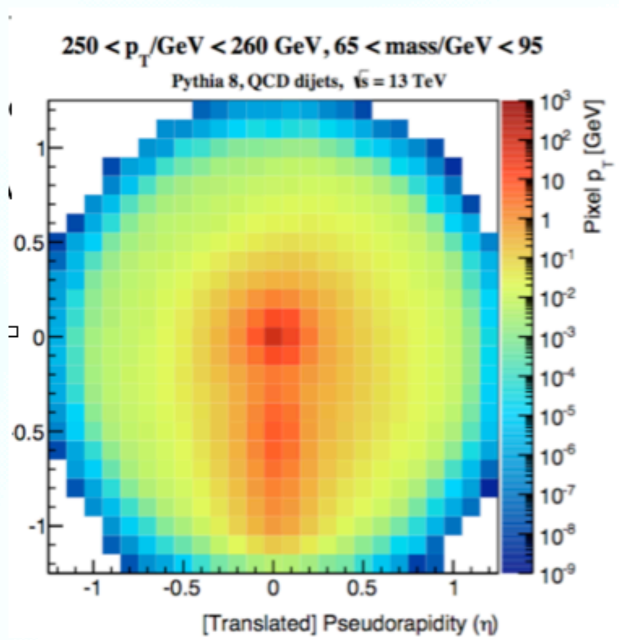
→ Manually imposed ordering might impair the performance

→ The length of the list is fixed but the no. of particles in each event is flexible

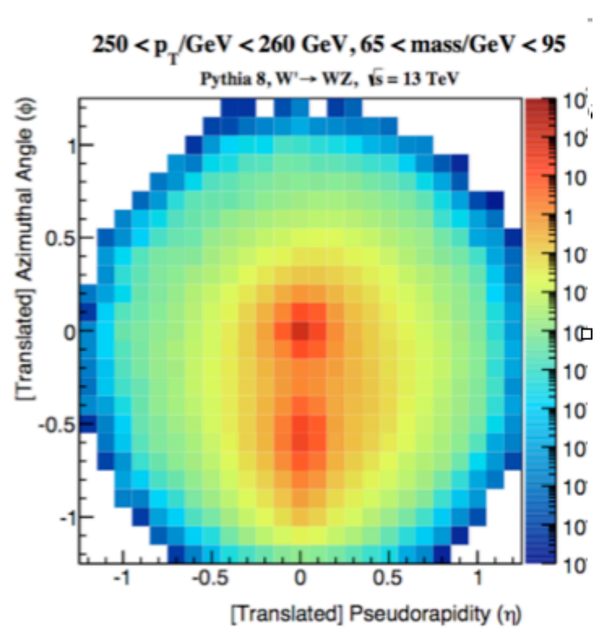
✓ Images: map each pixel of an image with pre-defined intensity

→ Incorporating additional information is not straight-forward

→ **Sparse representation.** $O(1) \sim O(10)$ particles for each event, $O(1000)$ pixels for each image




**g jets**




**light q jet**

# Complex Models

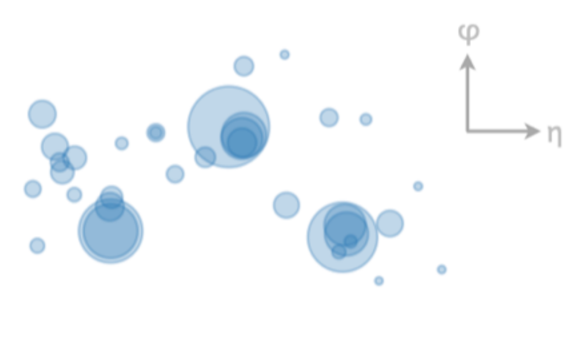
## Data Representation

- **What is the most natural representation for HEP events?**

- Ordered lists(tables) / Binary Trees
  → Manually imposed ordering might impair the performance
  → The length of the list is fixed but the no. of particles in each event is flexible

- Images: map each pixel of an image with pre-defined intensity
  → Incorporating additional information is not straight-forward
  → Sparse representation. $\mathrm{O}(1) \sim \mathrm{O}(10)$ particles for each event, $\mathrm{O}(1000)$ pixels for each image

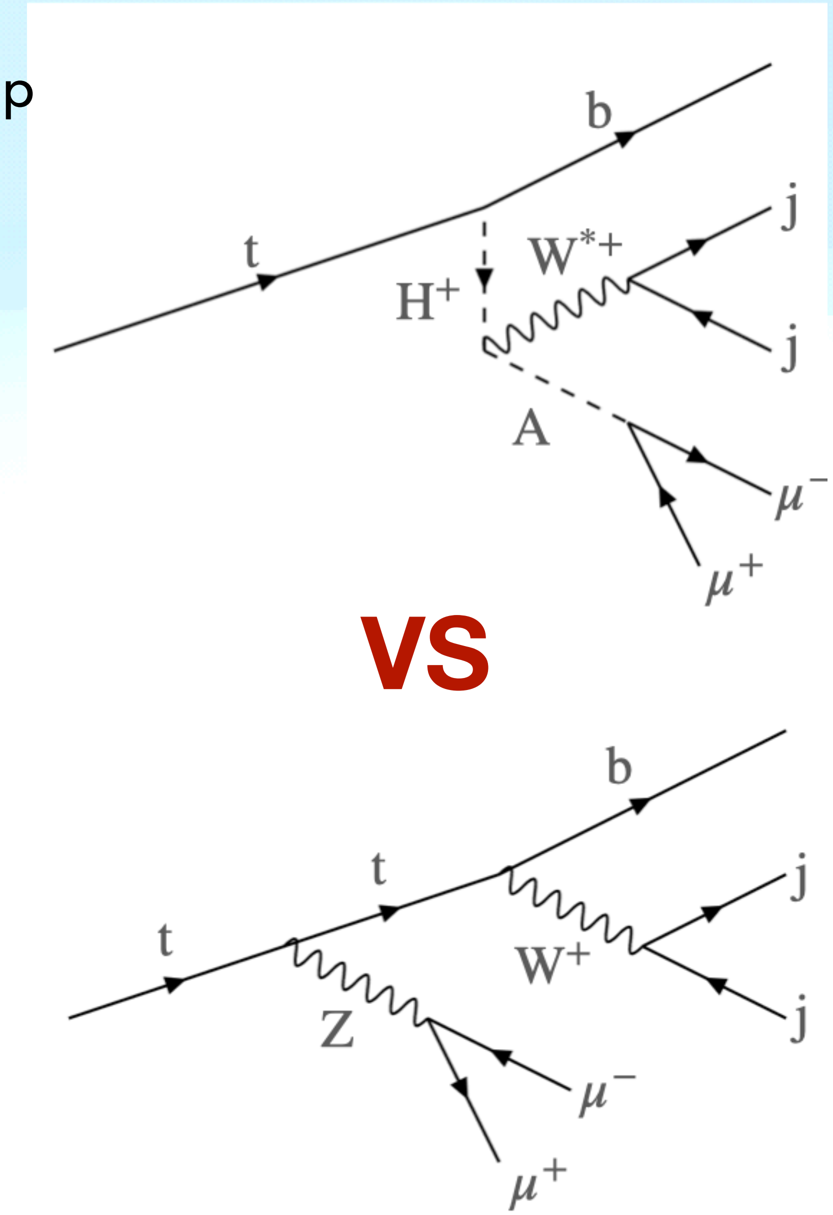
- **Graphs / Particle Clouds(Graphs without edges)**
  → An **unordered**, **permutation invariant** set of particles
  → No need to fix the variable size / No intrinsic ordering
  → Still embedding relationship between 3 nodes are not straight-forward



# Complex Models

## Example

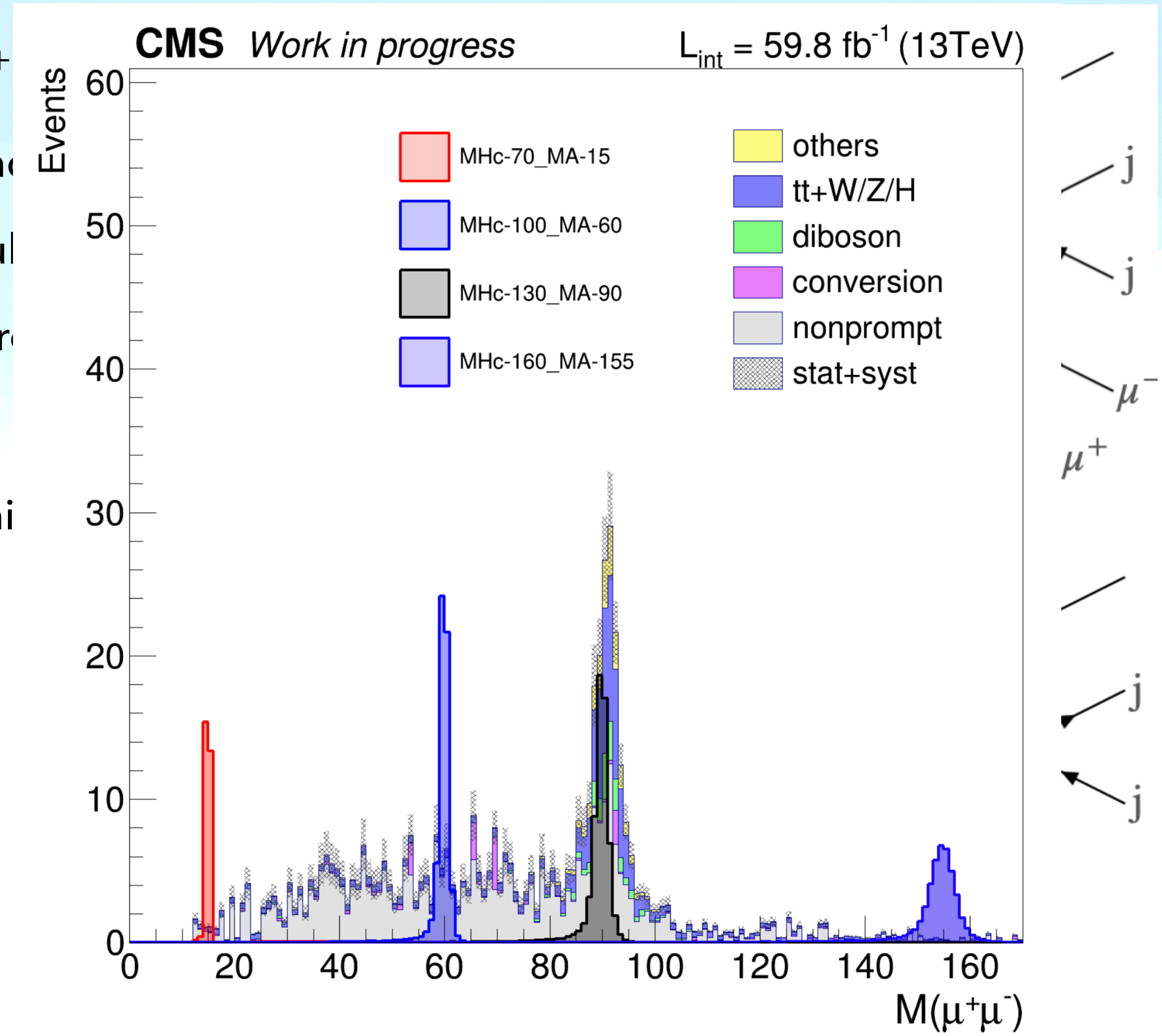
- **Classifying BSM Higgs signal and TT+Z**
- 5 Higgs in 2HDM model - light $\mathrm{H}^{+}$ can be branching from top
- In this study, fix the mass of $\mathrm{H}^{+}$ and $\mathrm{A}$ to be 130 / 90 GeV
- Final state consists of $e\mu^{+}\mu^{-}$ + multi-(b)jets
- $\mathrm{TT}+\mathrm{Z}$ is one of the major backgrounds
- **Remarks in this example**
- $\mathrm{M}\left(\mu^{+}\mu^{-}\right)$ will be the final discrimination variable



# Complex Models

## Example

- **Classifying BSM Higgs signal and TT+Z**
  - 5 Higgs in 2HDM model - light $\mathrm{H}^{+}$
  - In this study, fix the mass of $\mathrm{H}^{+}$ an
  - Final state consists of $e\mu^{+}\mu^{-}$ + mul
  - $\mathrm{TT}+\mathrm{Z}$ is one of the major backgr
- **Remarks in this example**
  - $\mathrm{M}\left(\mu^{+}\mu^{-}\right)$ will be the final discrimi

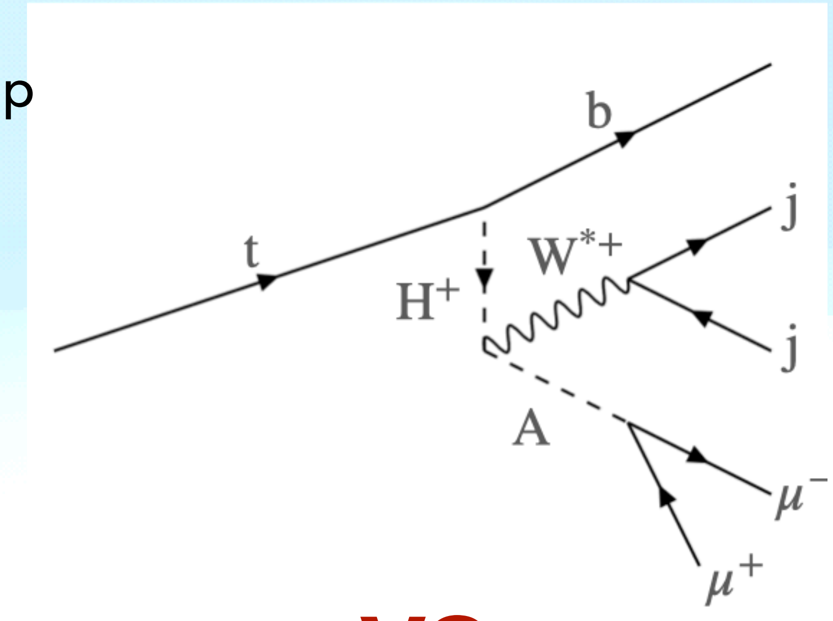


# Complex Models

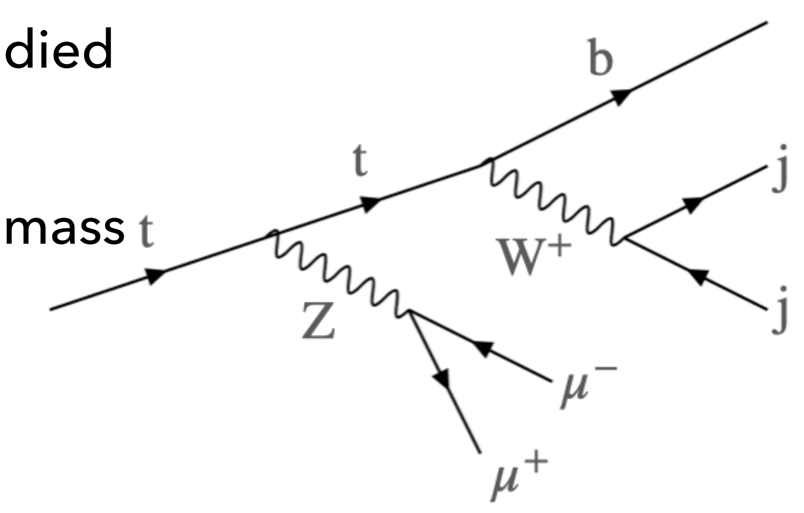
## Example

- **Classifying BSM Higgs signal and TT+Z**
- 5 Higgs in 2HDM model - light $\mathrm{H}^{+}$ can be branching from top
- In this study, fix the mass of $\mathrm{H}^{+}$ and $\mathrm{A}$ to be 130 / 90 GeV
- Final state consists of $e\mu^{+}\mu^{-}$ + multi-(b)jets
- $\mathrm{TT}+\mathrm{Z}$ is one of the major backgrounds

- **Remarks in this example**
- $\mathrm{M}\left(\mu^{+}\mu^{-}\right)$ will be the final discrimination variable
- For further discrimination, Graph Neural Networks will be studied
- Not only the discrimination power,
  we want the model considering features other than di-muon mass



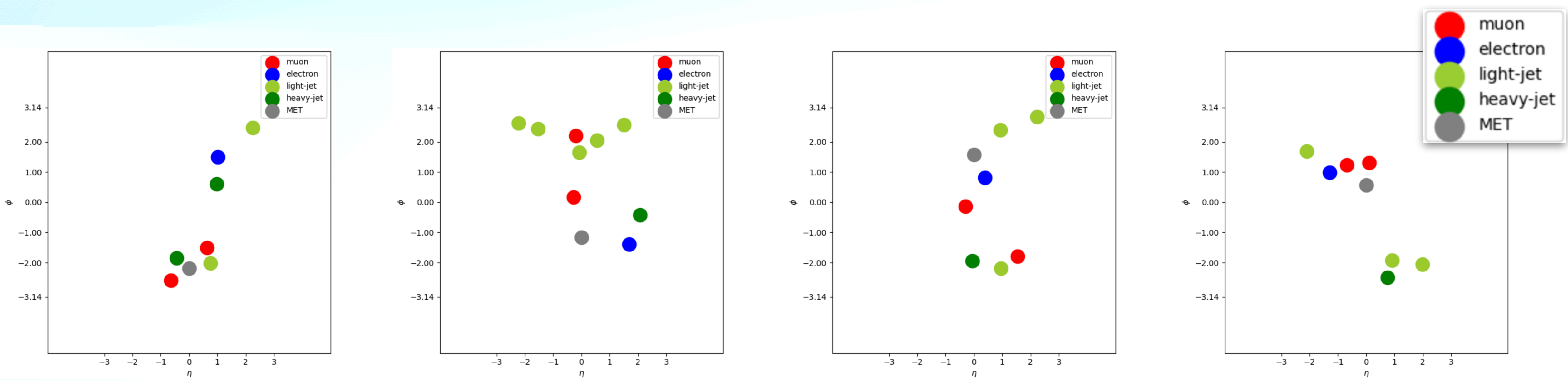

**VS**



# Complex Models

## Data Representation

- **Input features for graph classification**
- Each event is represented as **fully connected undirected graph**
- Node features: 4 momentum of the particle, charge, type of the particle, b-tagging score for jets
- 105K events for the signal and the background, total 210K events, 6:3:1 split for train:valid:test

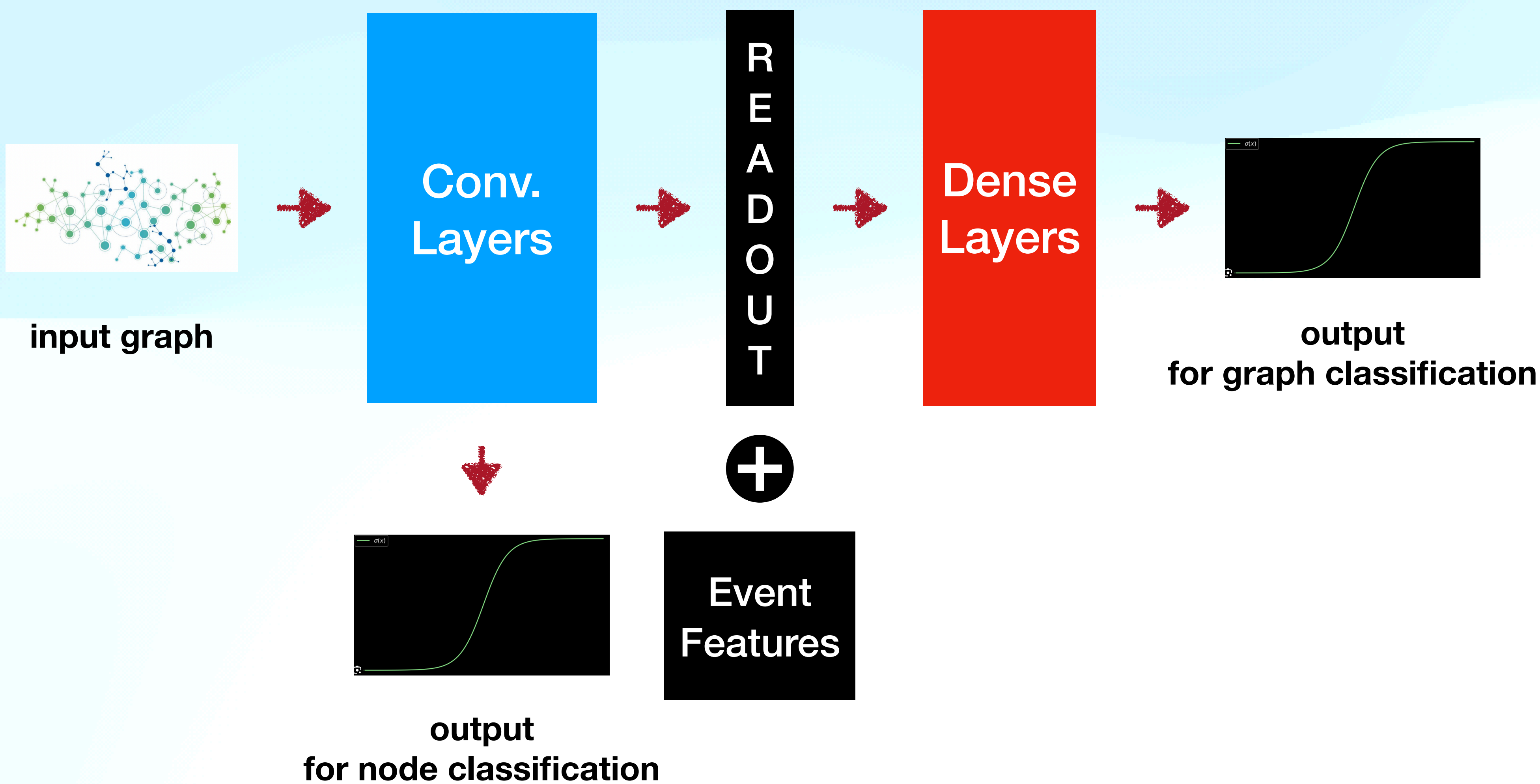



Can you distinguish signal and background events?

# Complex Models

## Graph Neural Networks

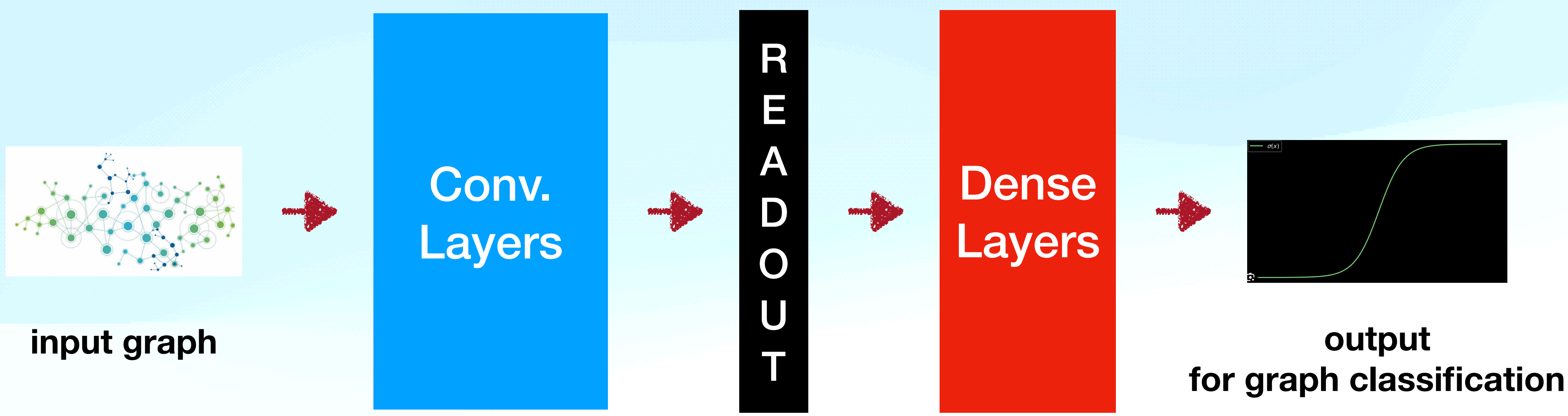
### Basic Structure



# Complex Models

## Graph Neural Networks

**Default Model**




- Conv Layers: TransformerConv(64) → DynamicEdgeConv(64) → DynamicEdgeConv(64)
- Readout: Mean Aggregation for each node features
- Dense Layers: batchnorm → (alpha_dropout(0.4) → dense(64) → SeLU activation)x2 → sigmoid
- In this example, I will test the **dropout rate** of the **connection of the edges**

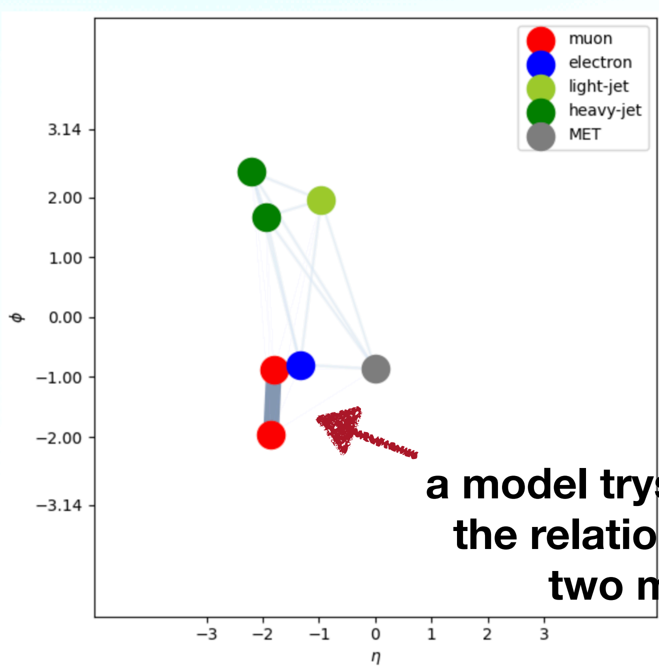
# Complex Models

## Convolution Layers

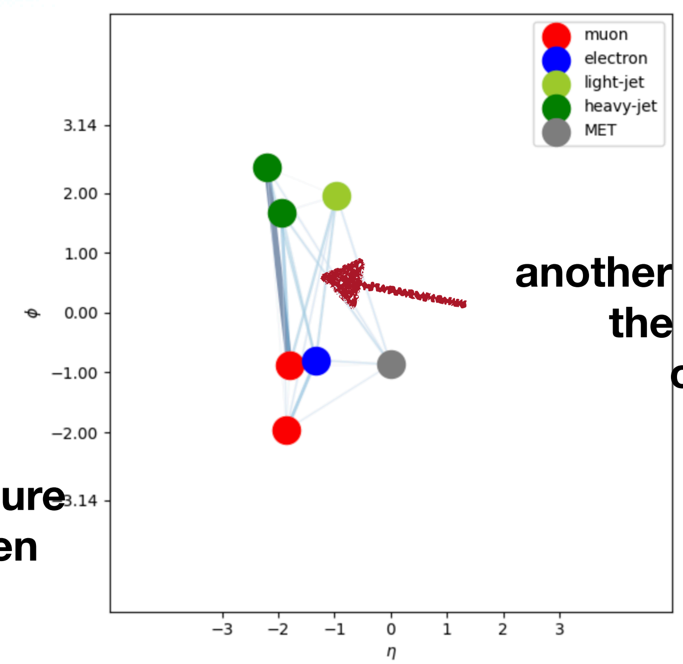
- **Intrinsic Explainability**
- Conv Layers: **TransformerConv(64)** → DynamicEdgeConv(64) → DynamicEdgeConv(64)

$$\mathbf{x}'_i = \mathbf{W}_1 \mathbf{x}_i + \sum_{j \in \mathcal{N}(i)} \boxed{\alpha_{i,j}} \mathbf{W}_2 \mathbf{x}_j, \quad \longrightarrow \quad \alpha_{i,j} = \mathrm{softmax}\left( \frac{(\mathbf{W}_3 \mathbf{x}_i)^\top (\mathbf{W}_4 \mathbf{x}_j)}{\sqrt{d}} \right)$$

- Attention masks already impose the relation between particles!




a model trys to capture
the relation between
two muons




another model trys to capture
the relation between
other particles

# Complex Models

## Convolution Layers

- **Intrinsic Explainability**
- Conv Layers: **TransformerConv(64)** → DynamicEdgeConv(64) → DynamicEdgeConv(64)

$$\mathbf{x}'_i = \mathbf{W}_1 \mathbf{x}_i + \sum_{j \in \mathcal{N}(i)} \boxed{\alpha_{i,j}} \mathbf{W}_2 \mathbf{x}_j, \quad \longrightarrow \quad \alpha_{i,j} = \mathrm{softmax}\left( \frac{(\mathbf{W}_3 \mathbf{x}_i)^\top (\mathbf{W}_4 \mathbf{x}_j)}{\sqrt{d}} \right)$$

- Attention masks already impose the relation between particles!
- Convolution layers supports **dropout_p** which randomly disconnect the edges while training → want to find optimum value of this hyperparameter

# Complex Models

## Hyperparmeter Optimization

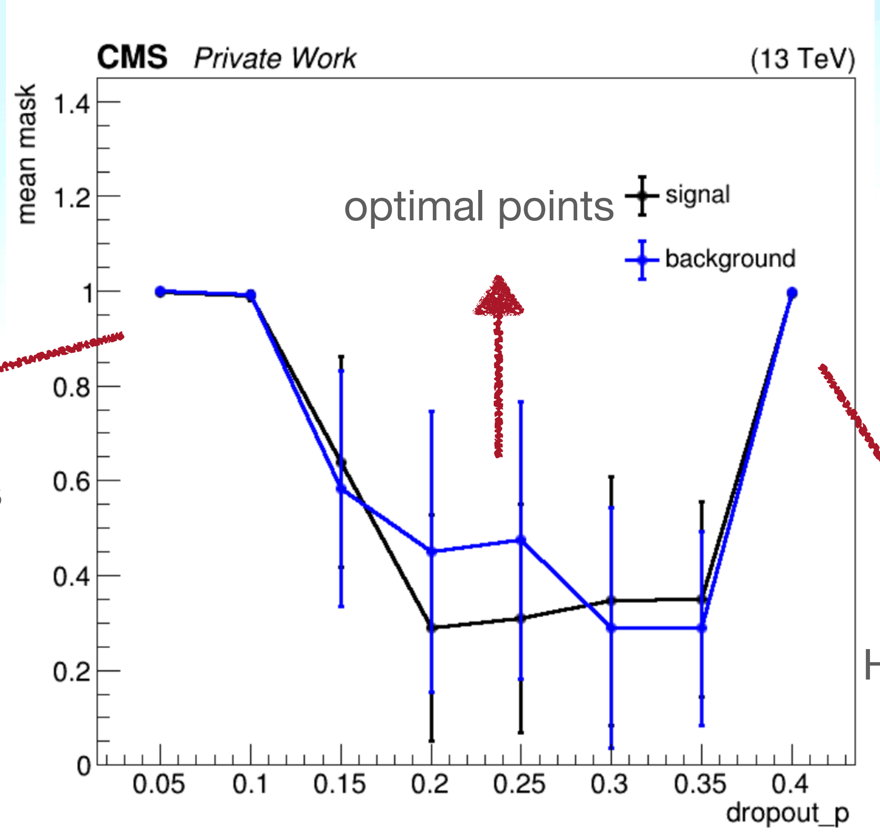
**Graduate Descent Method via XAI**

- Change the dropout_p values and check the distributions of the mask attention of two muons
- Tested [0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4]

**Experimental Settings**
- optimizer: Adam
- learning rate: 0.002
- scheduling: Cyclic LR
- 30 epochs for each model

**Results**
- ~80% accuracy for all models

without edge_dropout,
model primarily considers the relations
between two muons



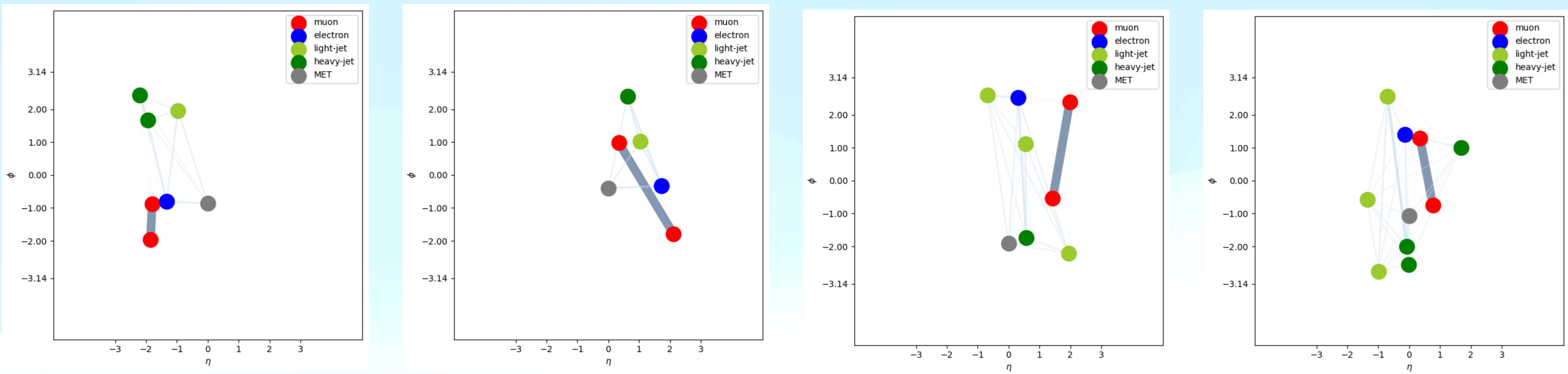

Too large dropout rate
Hard to capture the relations
between other particles

- 0.2~0.3 would be the optimal value!
reduced one dimension for hyperparameter optimization

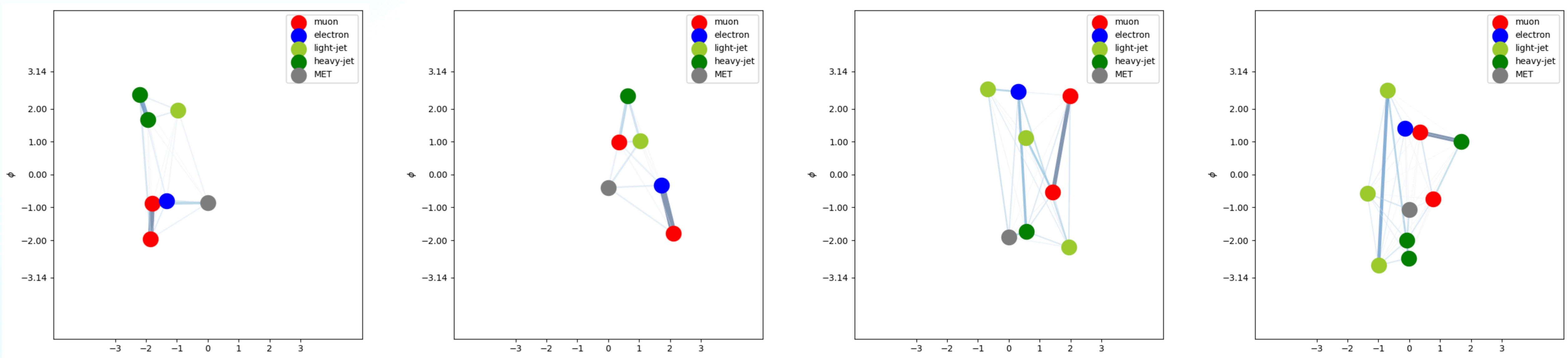
# Complex Models

## Visualization

**dropout_p = 0.05**




**dropout_p = 0.25**




✓ Masks for two muons are isolated from the other particles' graph

# Complex Models

## Surrogated Models

- **Models without intrinsic explanation**
- Edge masks are first order gradients - Can we map from inputs to outputs directly?
- Integrated Gradients for edges → path integral from 0 edge weights to 1

$$\mathrm{IG}_i(w) \sim \int_{\alpha=0}^{1} \frac{\partial F(x' + \alpha(w - w'))}{\partial w_i} d\alpha$$

- Modified ParticleNet does not support edge weights (attention is self-trainable)

- **Surrogated Models**
- We usually do not train the models with intrinsic explanation
- Train another model with intrinsic explanation → Surrogated Models!
- Use the same trainset, re-label the class labels as the model's outputs

# Complex Models

## Surrogated Models

- **GraphNet**
- Conv Layers: **GraphConv(64)** → DynamicEdgeConv(64) → DynamicEdgeConv(64)

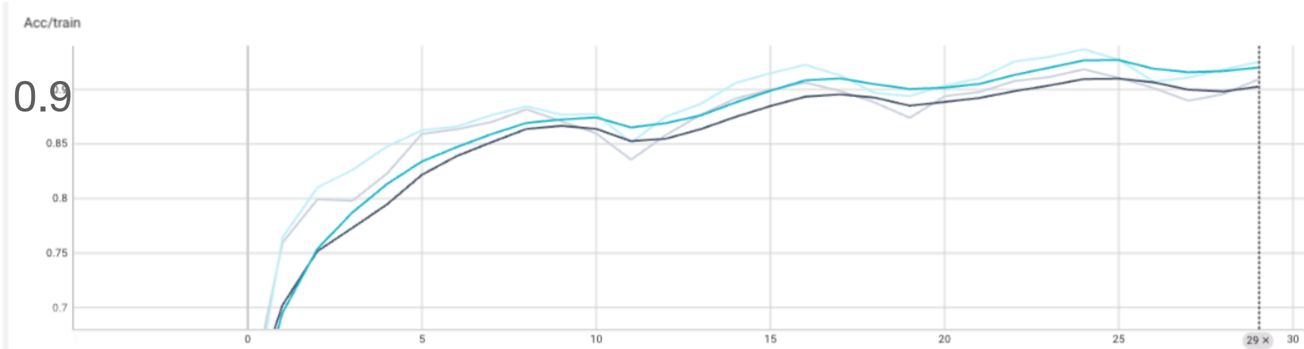
$$\mathbf{x}'_i = \mathbf{W}_1\mathbf{x}_i + \mathbf{W}_2 \sum_{j\in\mathcal{N}(i)} \boxed{e_{j,i}} \cdot \mathbf{x}_j$$

**Integration Variable**

c.f.) TransformerConv

$$\mathbf{x}'_i = \mathbf{W}_1\mathbf{x}_i + \sum_{j\in\mathcal{N}(i)} \alpha_{i,j}\mathbf{W}_2\mathbf{x}_j,$$

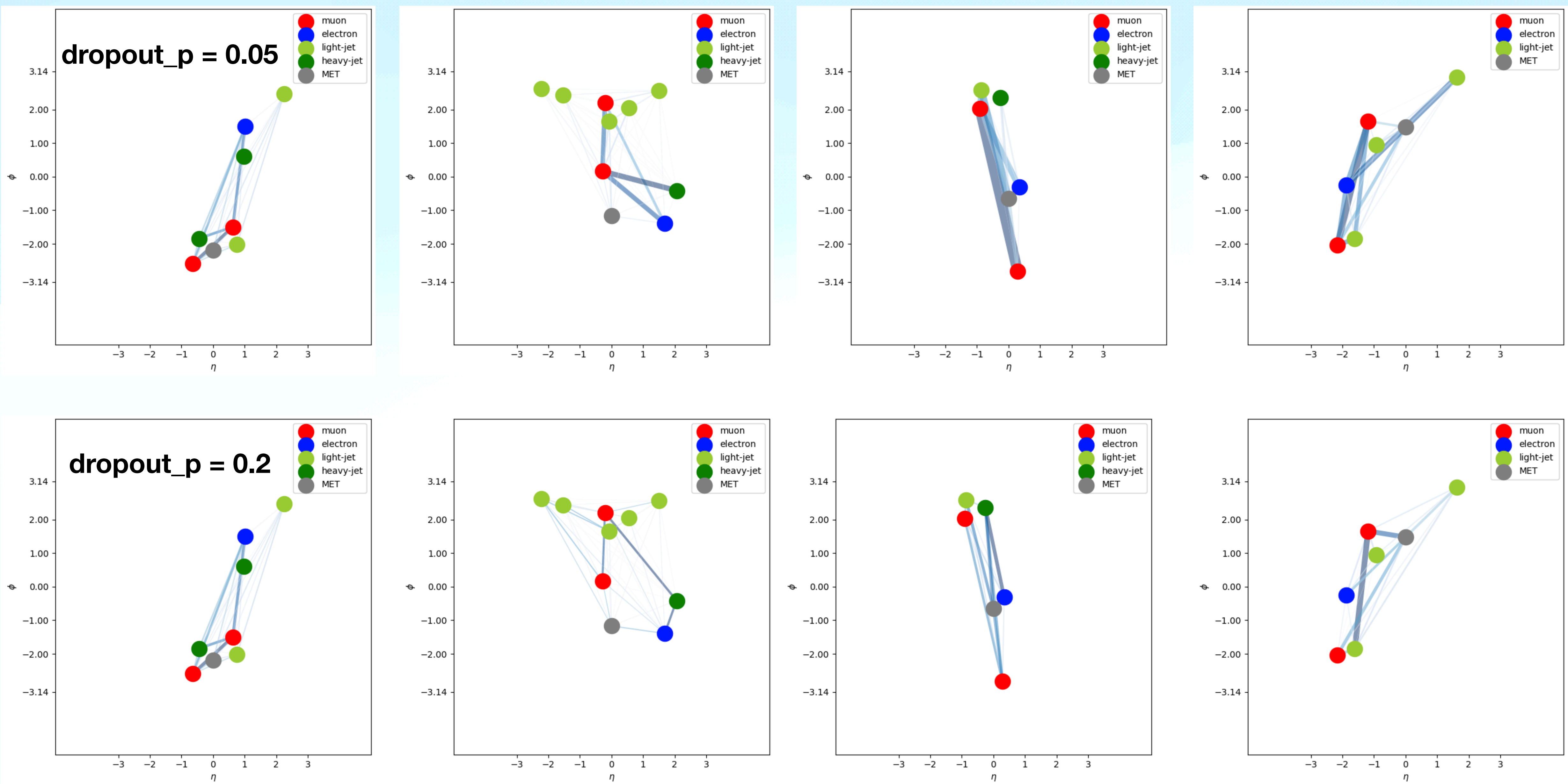
- No edge_dropout applied
- **Training the surrogated model**
- Used the same trainset, re-labelling the class labels with original model's output
- Trained two surrogated models for edge_dropout_p = 0.05 & 0.2
- Both surrogated models showed ~90% accuracy in re-labelled trainset



# Complex Models

## Surrogated Models

### Results



# Conclusion

**Explainability of AI models**

- Achieving explanatory of AI models are task-dep., sample-dep. and model-dep.
  → **No general rule** for achieving explainability!
- Large AI models are perfect for capturing the **correlation** between input features, but lack of **causality** make it hard to interpret
- XAI is a collection of methodologies to make **human-readible causally connected description** of AI models
- Modern **attribution methods** make possible for mapping from input to model output for deep learning models, based on local/global gradients
- Even if your model is not intrinsically explainable, it is possible to train **surrogated models** to achieve explainability