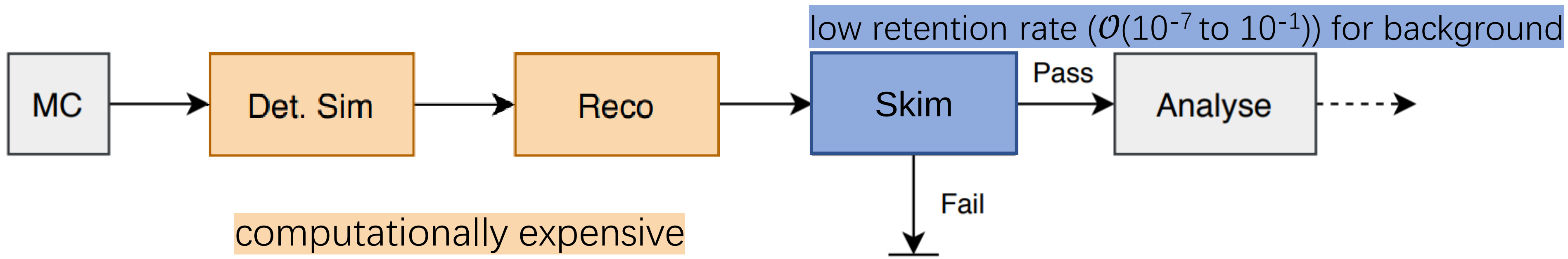
Sampling Methods of SmartBKG

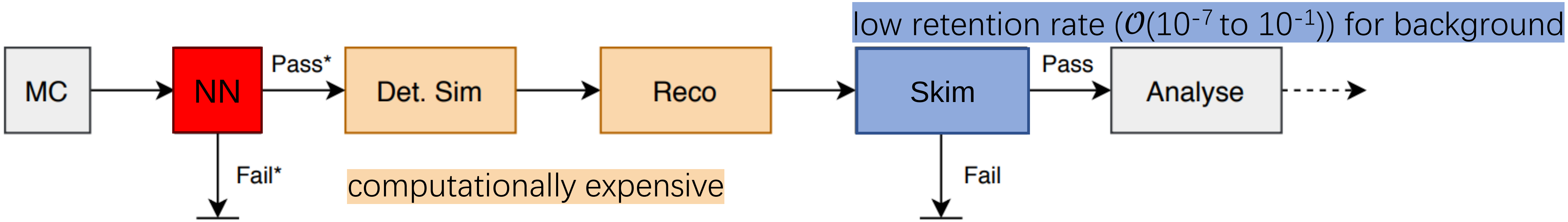
Boyang Yu[1] , Nikolai Hartmann[1] , Thomas Kuhr[1]
[1] *Ludwig-Maximilians-Universität München*
Sampling Techniques Hackathon, May 6th, 2024

Motivation

Normal Monte Carlo Simulation data flow



Simulation with SmartBKG selection

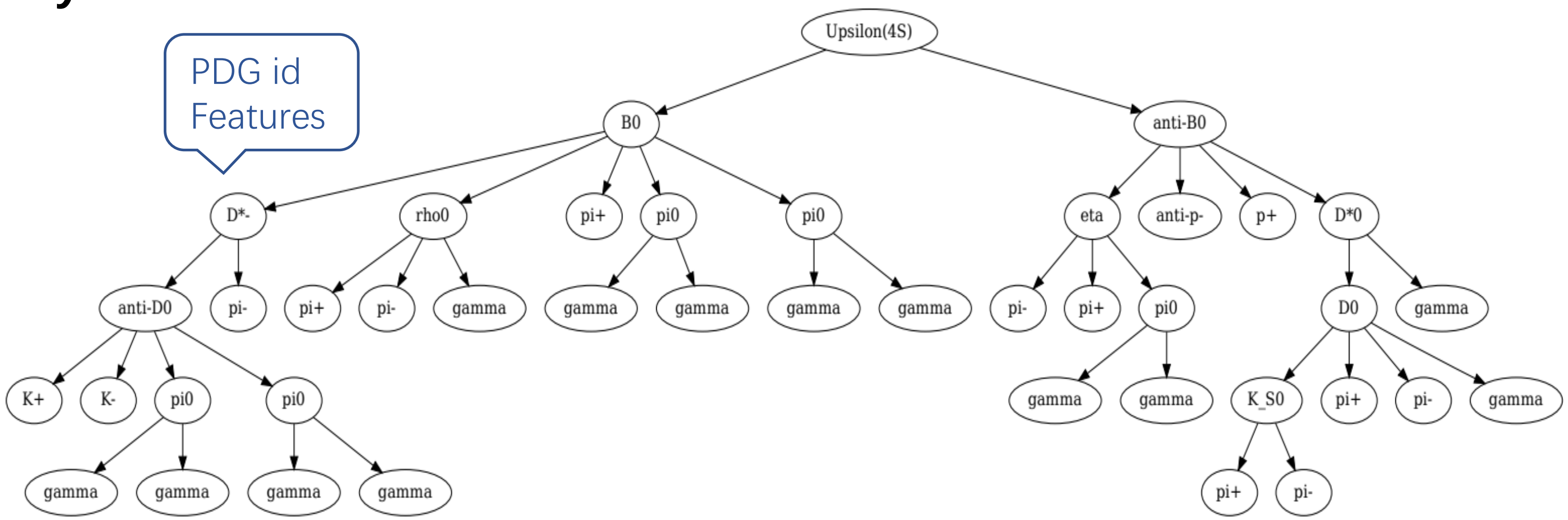


Introduction

Tree Structures of Particle Decay

⇅

Graph Neural Network

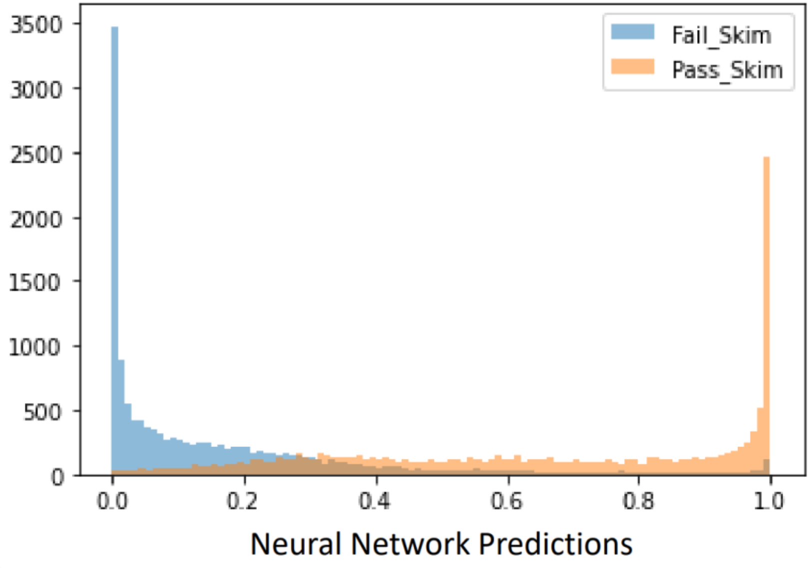


Dataset:

- Each event (each **Graph**):
  - ➢ Decay of $\Upsilon(4S) \rightarrow B^0\bar{B}^0$
  - ➢ Particles (**Nodes**)
  - ➢ Mother/Daughter relations (two way **Edges**) + self loops
  - ☐ Each particle (each **Node**)
    - ➢ **PDG** id
    - ➢ 8 **Features**: **Production time**, **Energy**, **Position** (3d), **Momentum** (3d)
- **Label** per event: Pass/Fail after the skims
  ⋆ FEI Hadronic B0, retention rate 4.25%
- Other event level **attributions** for the studies of sampling methods: e.g. $\boldsymbol{M_{bc}}$, $\boldsymbol{\Delta E}$ etc.

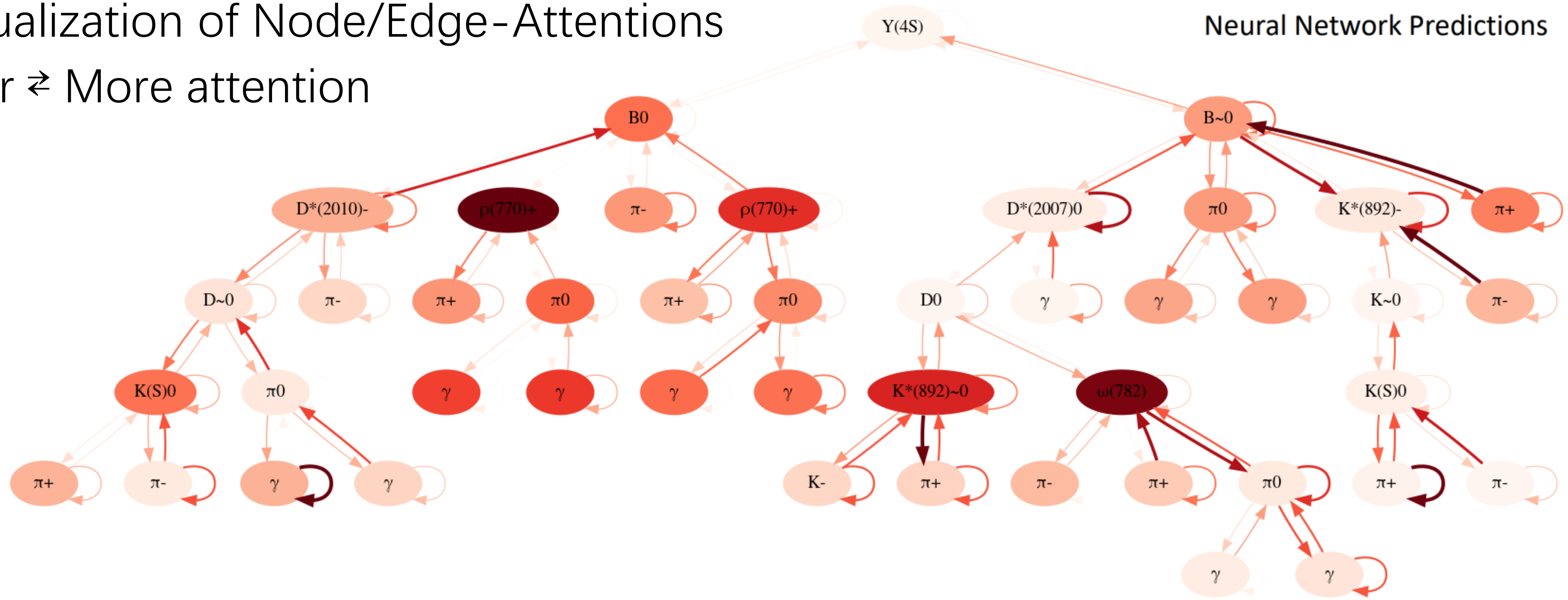
NN Performance

- NN output distribution

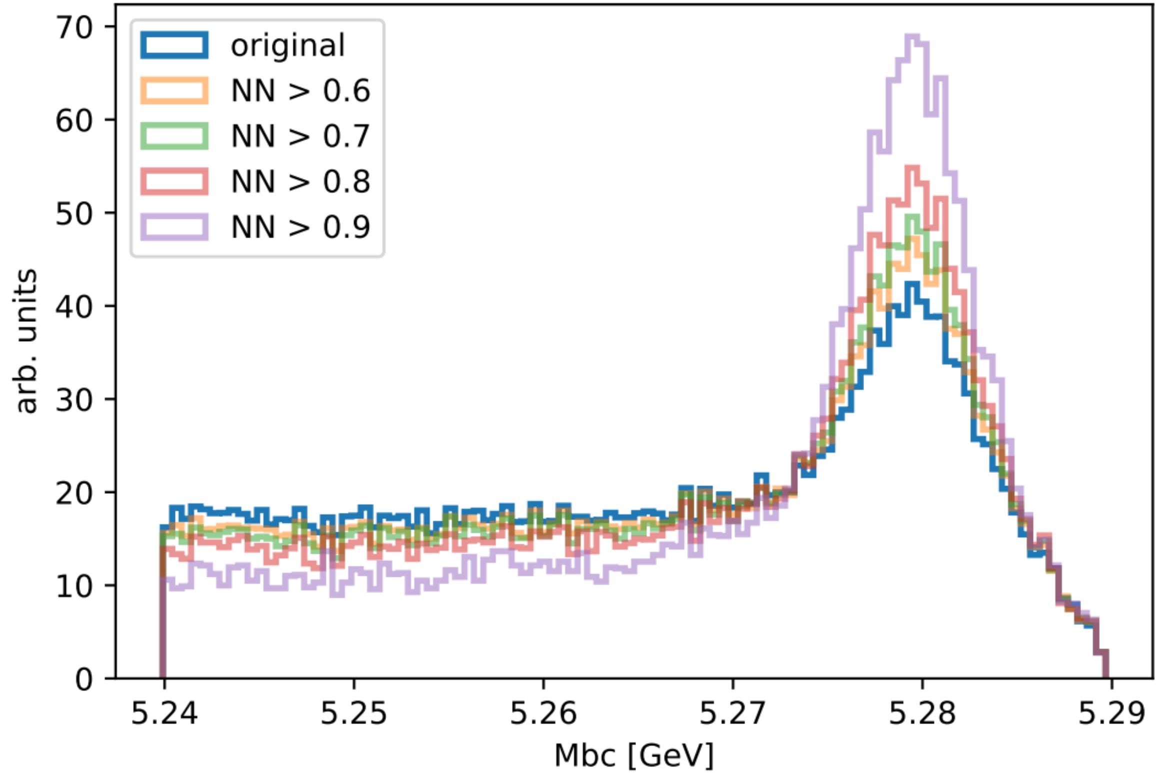


- Visualization of Node/Edge-Attentions

Darker ⇄ More attention



Bias due to False-Negatives with Naive Filtering



Skim \ NN	Positive	Negative
Pass	True-Positive (TP)	**False-Negative (FN)**
Fail	False-Positive (FP)	True-Negative (TN)

How to correct the bias brought by False-Negatives

- Reject events more carefully and cleverly-> sampling methods
  - Reduce **False-Negatives** but also **True-Negatives**, lower speedup
- Simulate **Pass** distribution with **True-Positive** events -> weights
  - Hard to have perfect simulation, still biased
- Studied:
  - With random sampling:
    - With weights: Importance sampling
    - Without weights: Rejection sampling
  - Without random sampling:
    - With weights: Reweighing

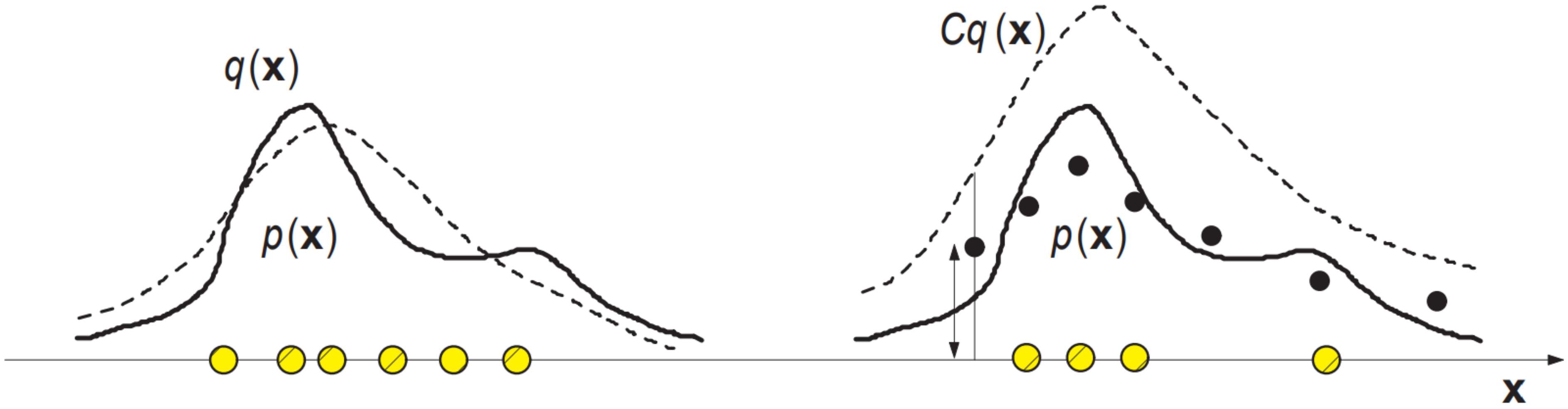
Speedup: $s = \frac{t_{no\_filter}}{t_{filter}}$

Effective Sample Size: $N_{eff} = \frac{(\sum \omega_i)^2}{\sum \omega_i^2}$

Skim \ NN	Positive	Negative
Pass	True-Positive (TP)	**False-Negative (FN)**
Fail	False-Positive (FP)	True-Negative (TN)

Sampling methods

Importance Sampling | **Rejection Sampling**



Statistics lecture

Importance Sampling:

- Sample x from q(x)
- Reweight with factor p(x)/q(x)
- Get simulated p(x)

Rejection Sampling:

- Sample x from Cq(x), u from unitary distribution
- Accept if Cq(x) < up(x)
- Get simulated p(x)

Our modeling

Importance Sampling:

- Predictions on all events to get q distribution
- Sample x from unitary distribution u(x)
- Sample q(x) from u(x)
- Reweight with factor 1/q(x)
- Get simulated p(x)=u(x)

⇔ Keep all events

Rejection Sampling:

- Predictions on all events NN(x)
- Build q distribution and find best C
  - from binned NN(x)
  - from manual function to simulate NN(x)
- Build p distribution
  - from binned NN(x) of **Pass** events

⇔ Keep all **Pass** events

Sampling methods

Importance Sampling
- NN output directly as probability
- Equally binned NN output as probability

Reweighting
- GBDT classifier for True-Positive / False Positive, trained on selected event-level attributions
- Weight from CLF output
- Weight from equally/quantile binned CLF output

Rejection Sampling
- Binned NN(x) as Cq(x), binned NN(x_**Pass**) as p(x)
- Simulated function Cq(x), binned NN(x) as p(x)
  - q(x) simulating binned NN(x)
- Simulated function Cq(x_**Pass**) as Cq(x), binned NN(x_**Pass**) as p(x)
  - q(x_**Pass**) simulating binned NN(x_**Pass**)
- ps. Unitary function as Cq(x), binned NN(x) as p(x) ⇔ Importance sampling
  - q(x) simulating NN(x), therefore unitary

Sampling methods

Comparisons

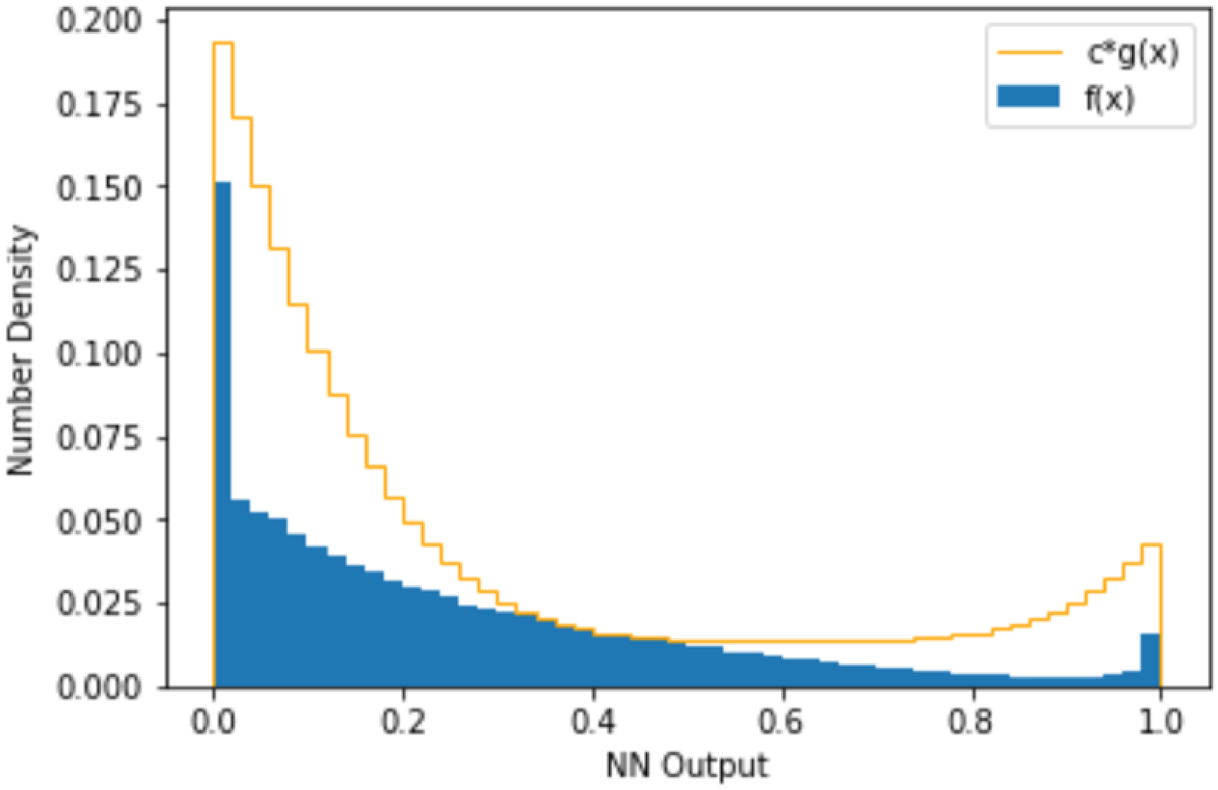
	Importance Sampling	Reweighting	Rejection Sampling
Use of NN output	As selection probability	As selection criteria	As input to selection probability
Prior information	**None**	Pretrained CLF, Selection threshold	Proposal distribution
Weight	Inversed NN output	From CLF output	**None**
Loss to train NN	Speedup	Binary cross entropy	Binary cross entropy
Speedup	2.0	**6.5**	2.6
Bias	**None**	None for CLF variables Small for others	Small

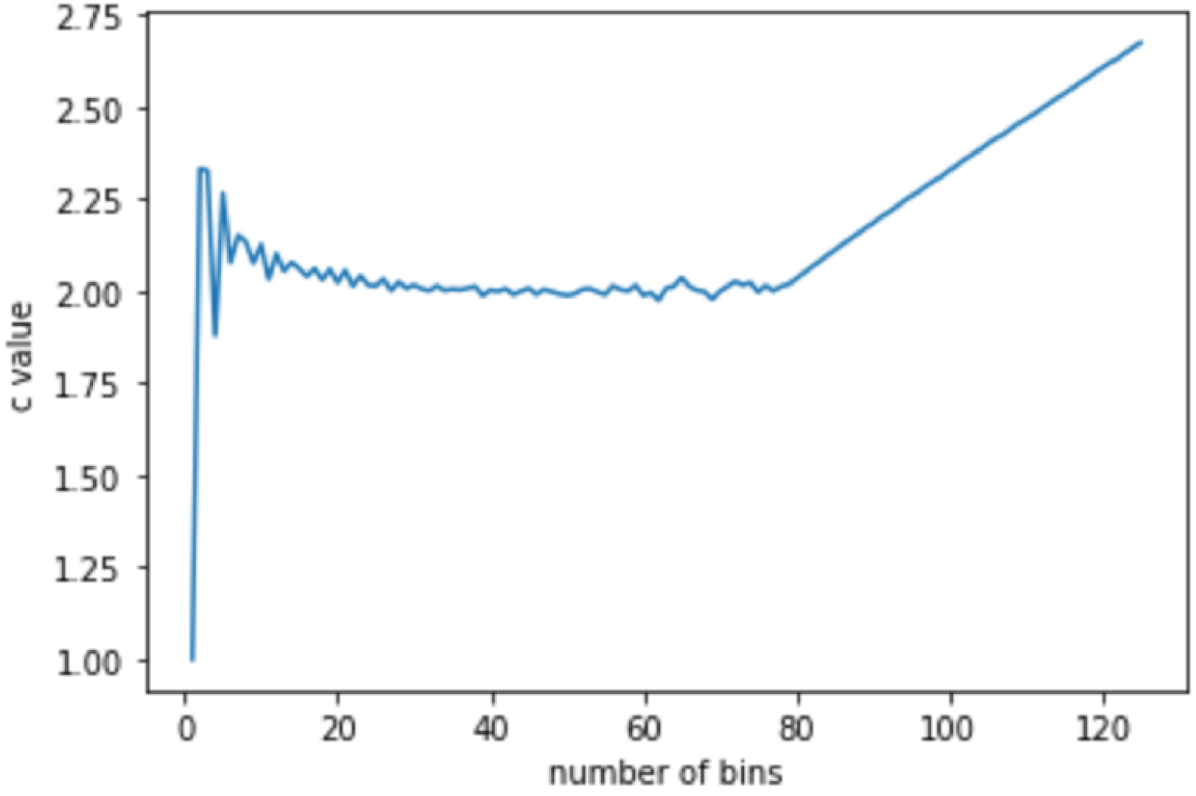
LUDWIG-
MAXIMILIANS-
UNIVERSITÄT
MÜNCHEN

Thank You for your Attention

Boyang Yu[1] , Nikolai Hartmann[1] , Thomas Kuhr[1]
[1] *Ludwig-Maximilians-Universität München*
Sampling Techniques Hackathon, May 6th, 2024

Backup





$$g(x) = \frac{(x-0.6)^4 + 0.01}{\sum_0^1 \Delta x\left[(x-0.6)^4 + 0.01\right]}$$

Best:

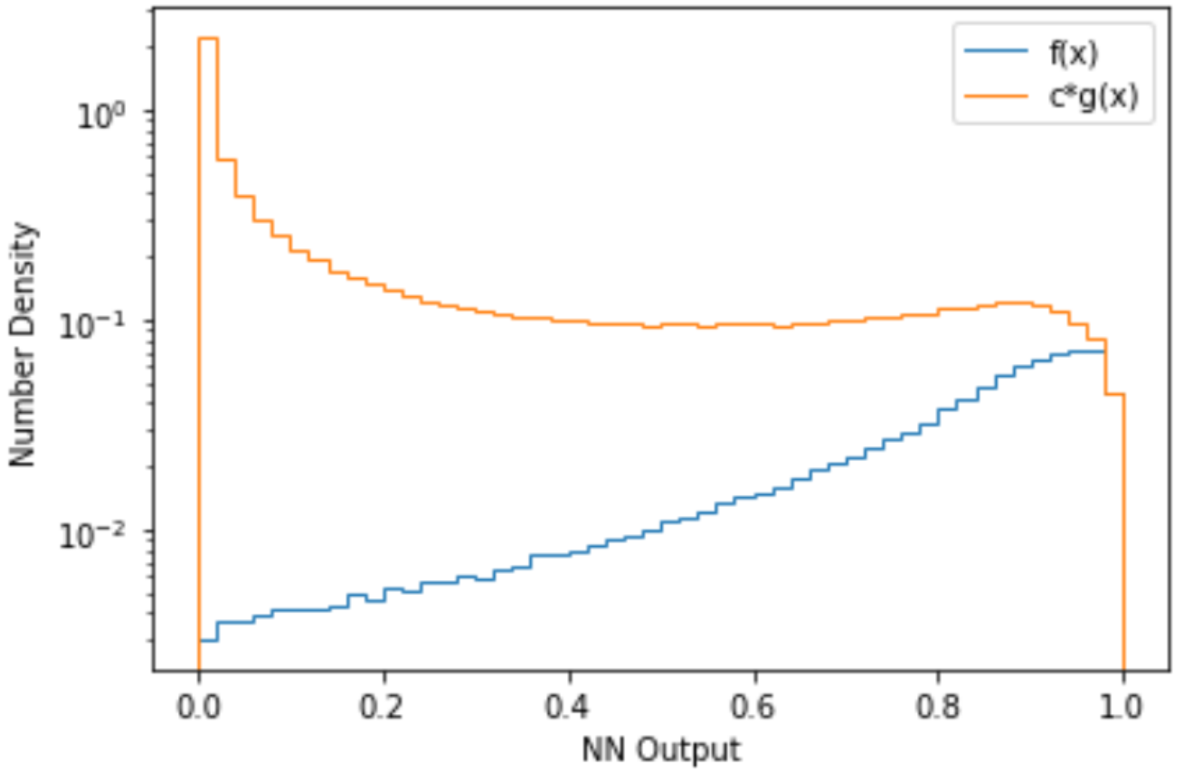
bins = 65

c = 1.98

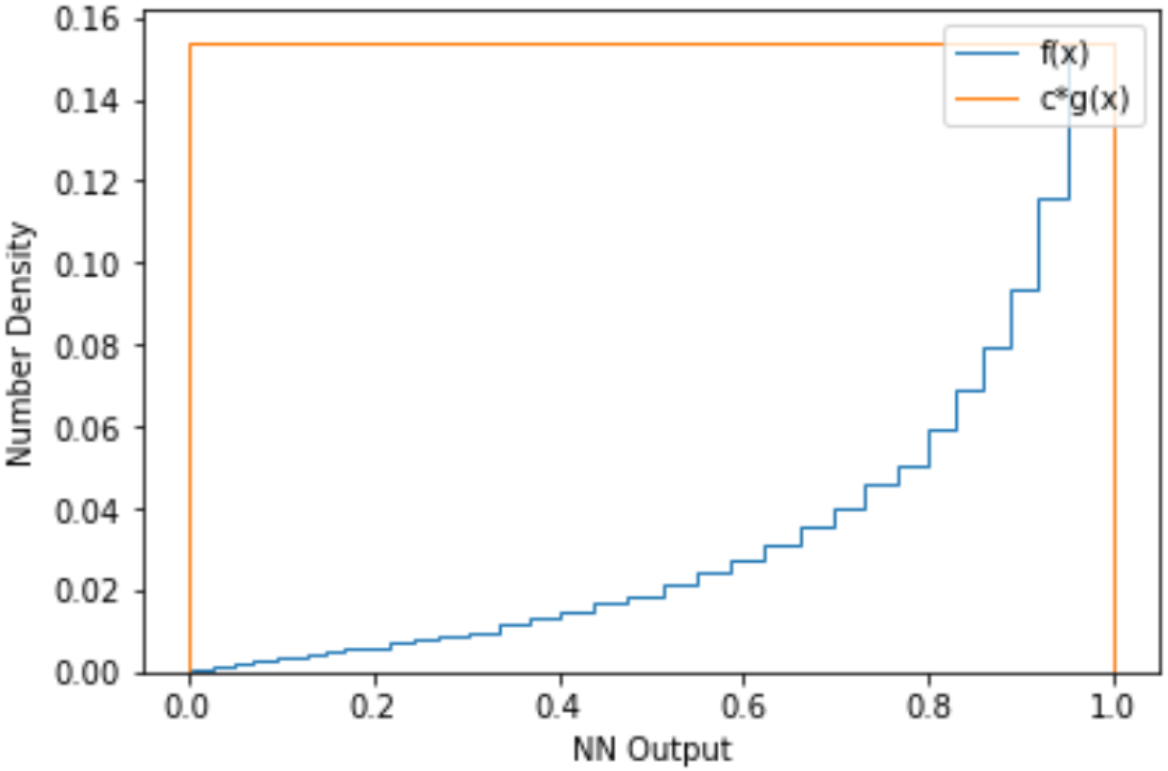
Chosen:

bins = 50

c = 1.99

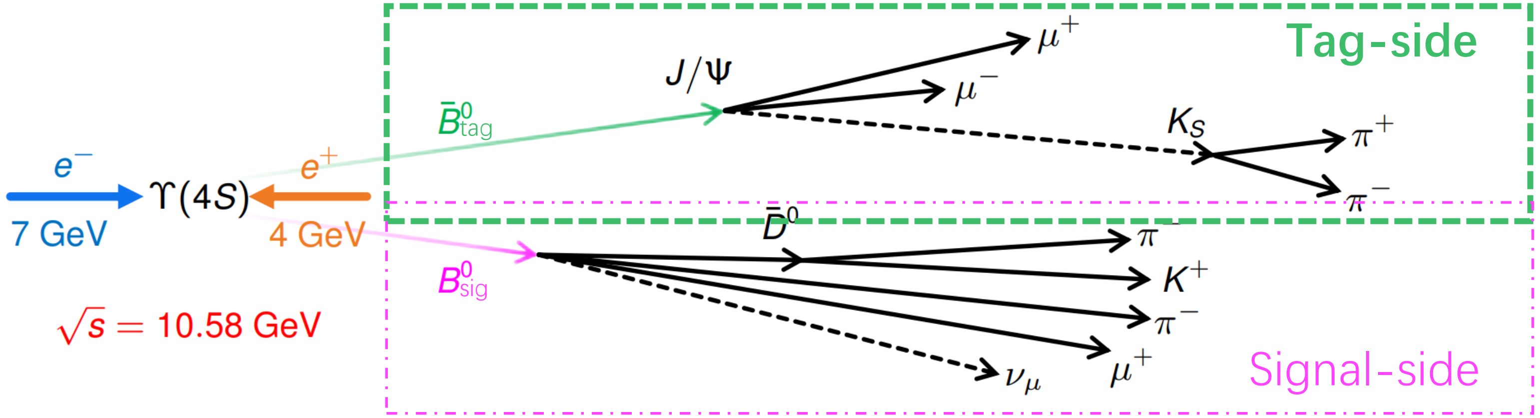


Equal
c=8.8



Quantiled
c=7.7

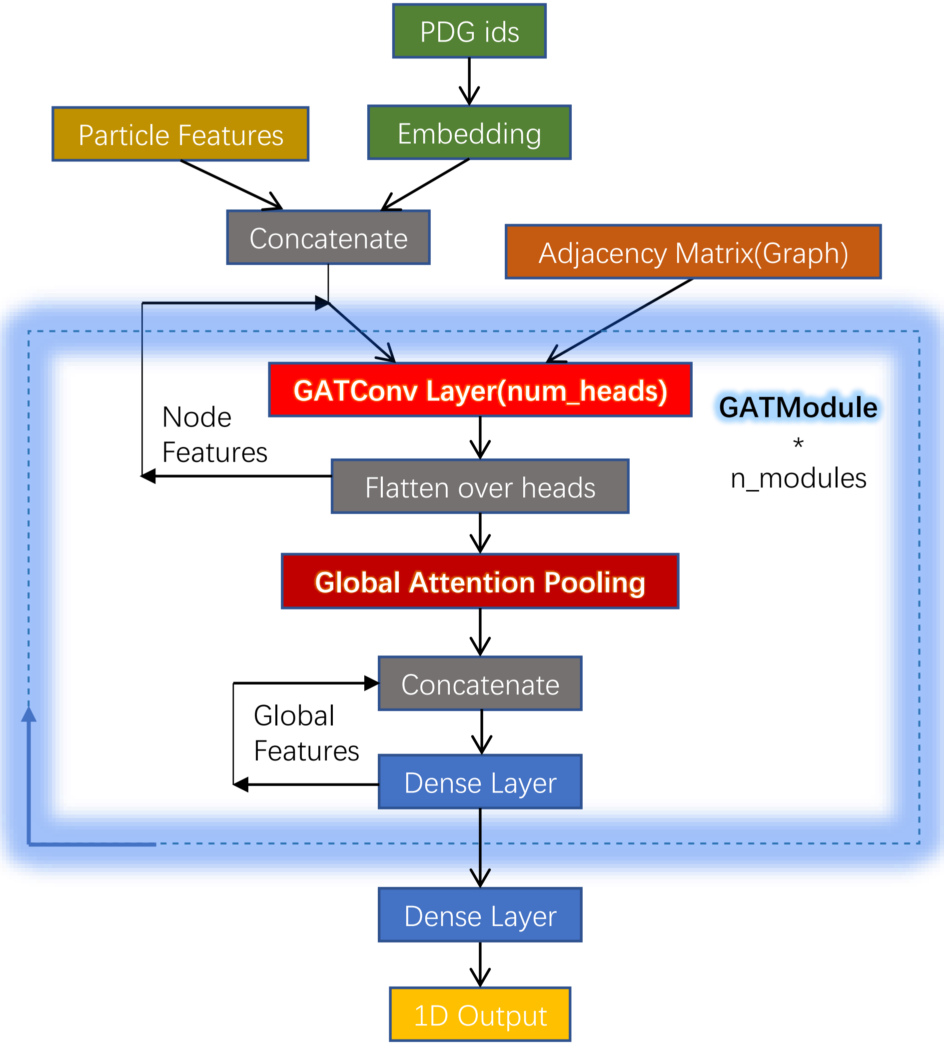
Tagging method:

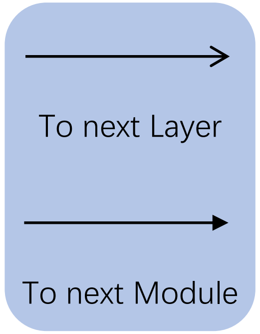


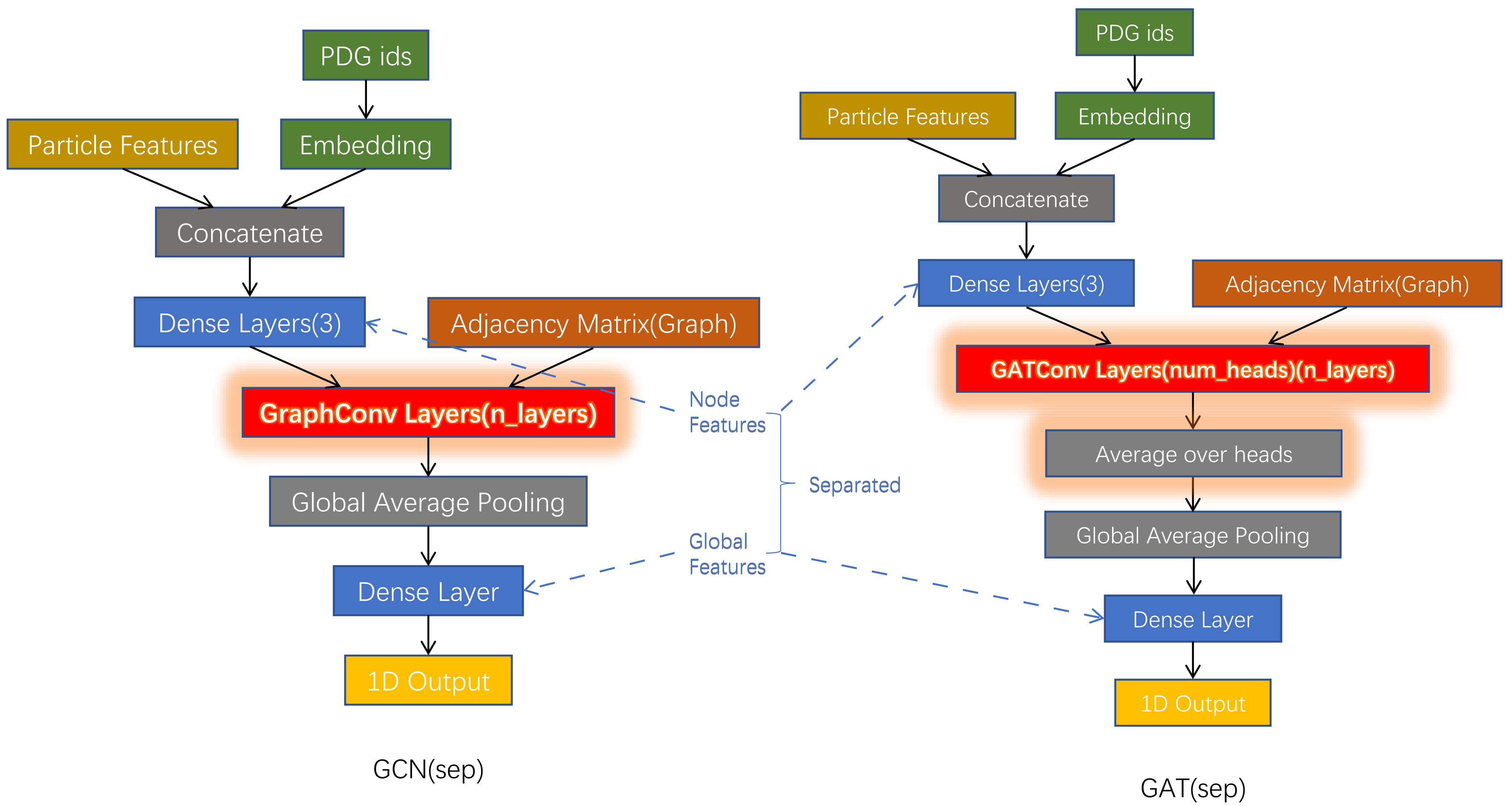
Retention rate after reconstruction and selection of tag-side B candidate:

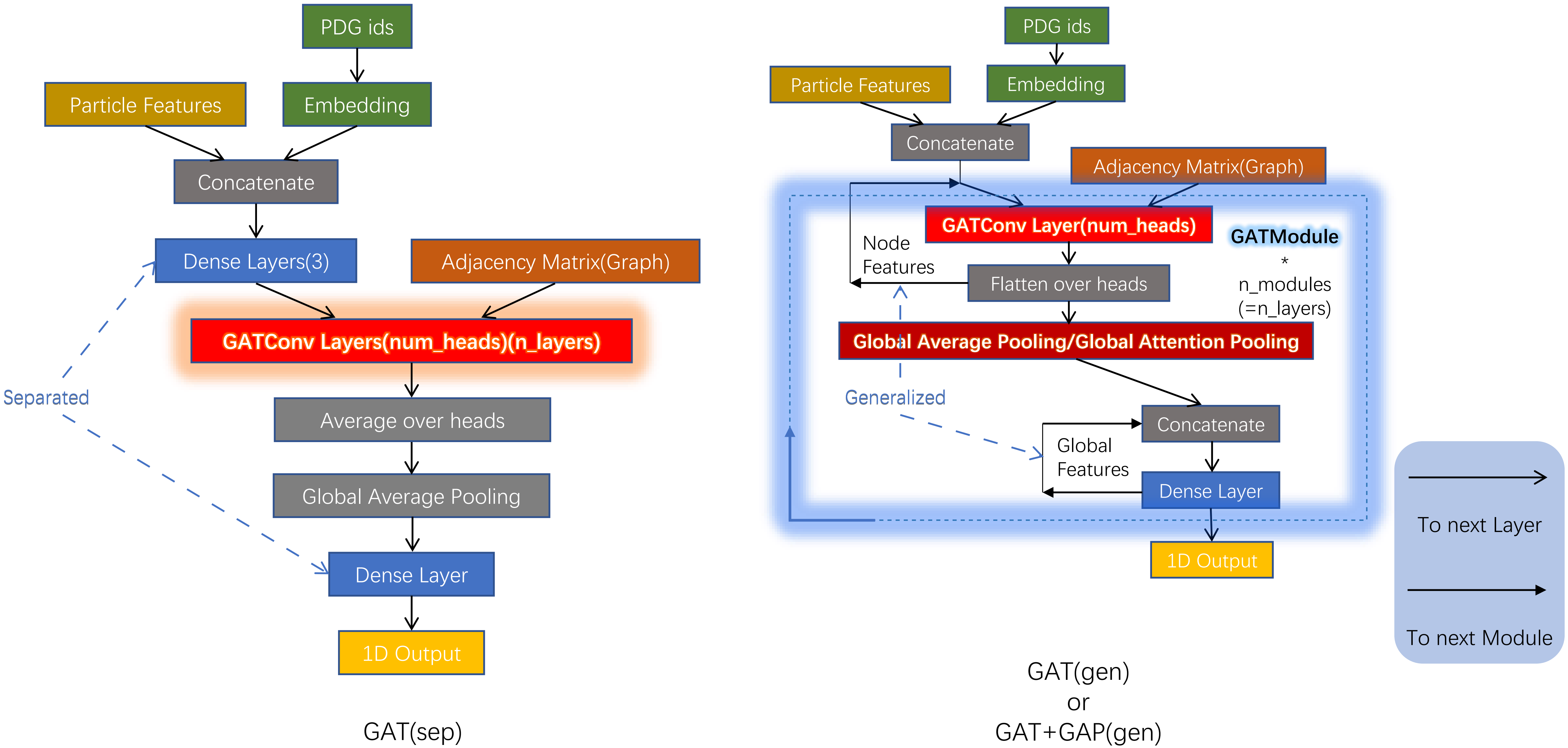
FEI Skim	Hadronic B^+	Hadronic B^0
Mixed ($\Upsilon(4s) \to B^0\bar{B}^0$)	5.62%	4.25%

Final Architecture:
GAT+GAP

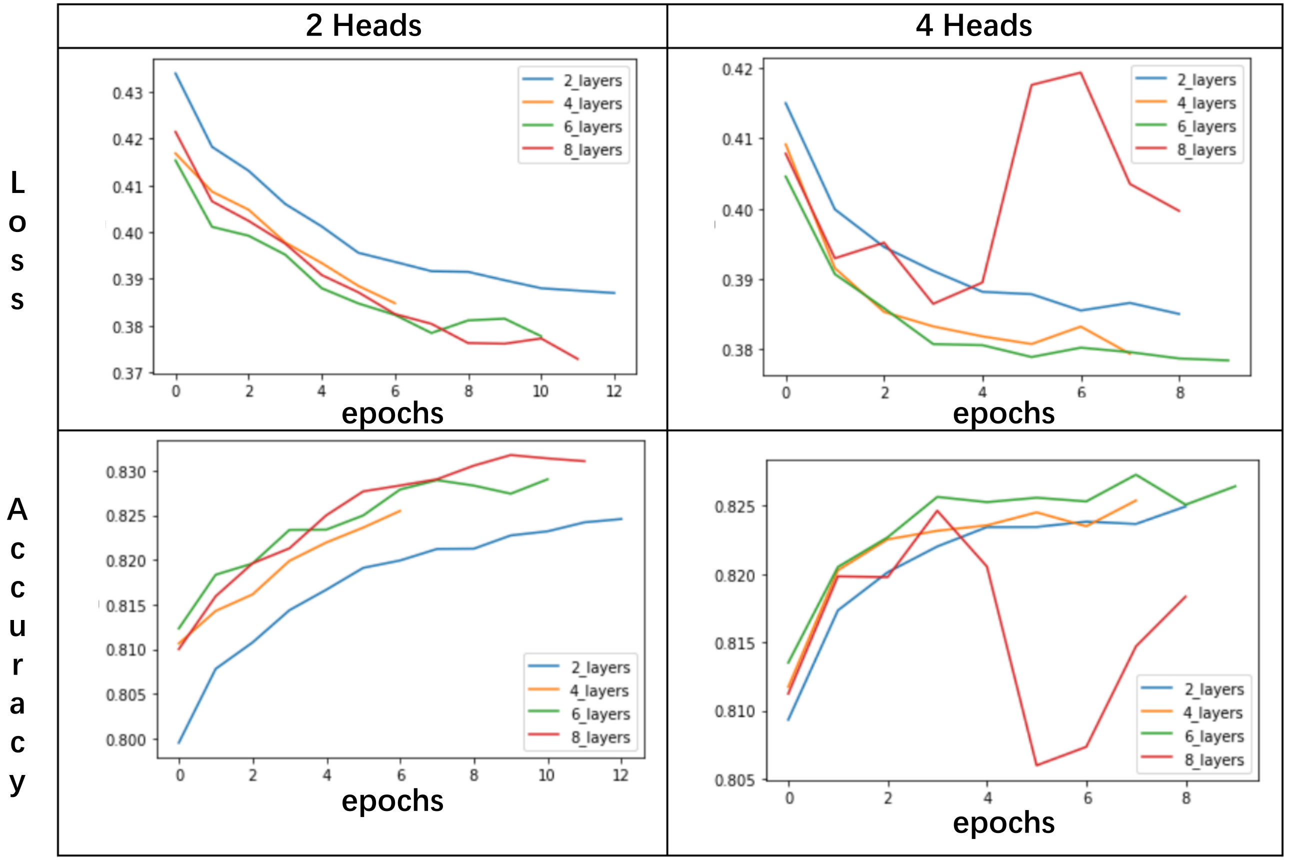


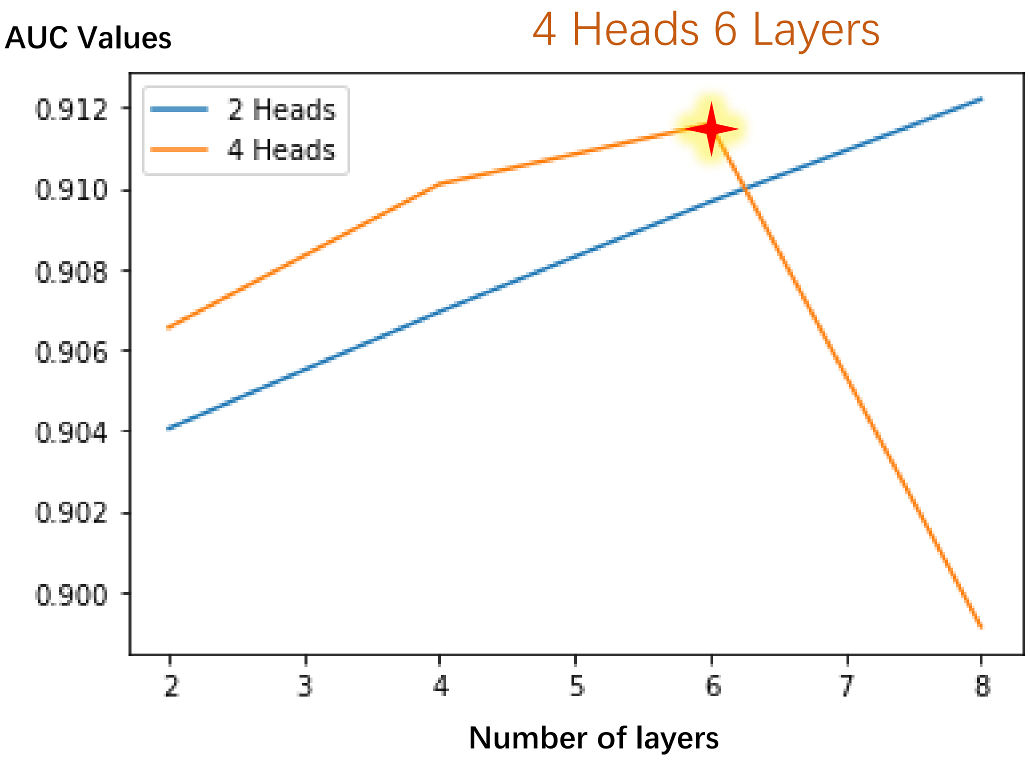



PDG ids
Particle Features
Embedding
Concatenate
Dense Layers(3)
Adjacency Matrix(Graph)
GraphConv Layers(n_layers)
Global Average Pooling
Dense Layer
1D Output
GCN(sep)
Node Features
Separated
Global Features
PDG ids
Particle Features
Embedding
Concatenate
Dense Layers(3)
Adjacency Matrix(Graph)
GATConv Layers(num_heads)(n_layers)
Average over heads
Global Average Pooling
Dense Layer
1D Output
GAT(sep)


PDG ids
Particle Features
Embedding
Concatenate
Dense Layers(3)
Adjacency Matrix(Graph)
GATConv Layers(num_heads)(n_layers)
Separated
Average over heads
Global Average Pooling
Dense Layer
1D Output
GAT(sep)
PDG ids
Particle Features
Embedding
Concatenate
Adjacency Matrix(Graph)
GATConv Layer(num_heads)
GATModule
*
n_modules
(=n_layers)
Node
Features
Flatten over heads
Global Average Pooling/Global Attention Pooling
Generalized
Concatenate
Global
Features
Dense Layer
1D Output
GAT(gen)
or
GAT+GAP(gen)
To next Layer
To next Module

Quantitative Studies





Comparison

Parameters:

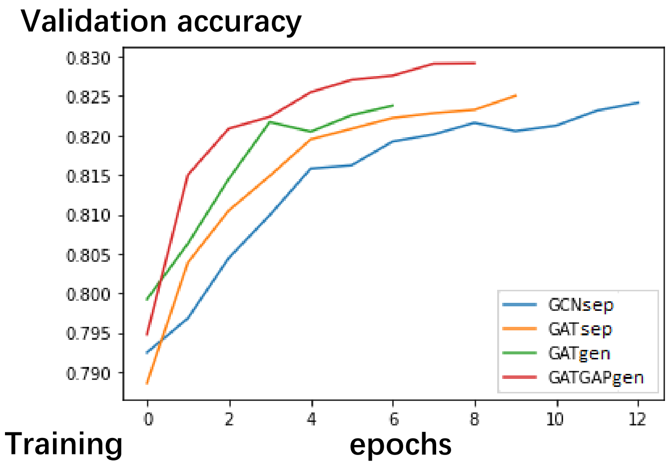
- n_heads = 4
- n_layers = 6
- n_units = 128
- batch_size = 128
- n_train = 0.9M
- n_val = 0.1M
- n_test = 0.5M

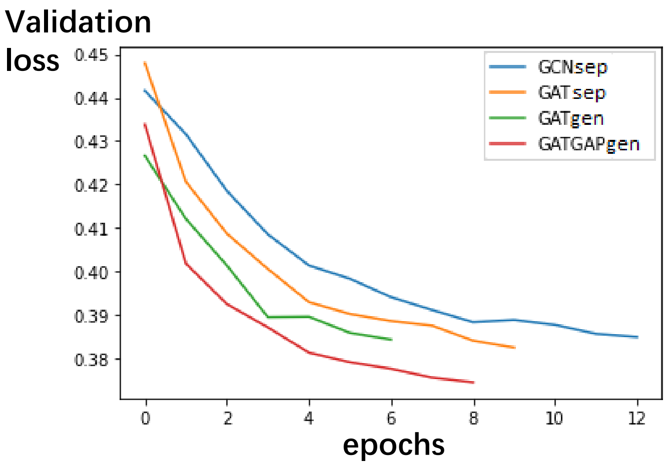
Loss:

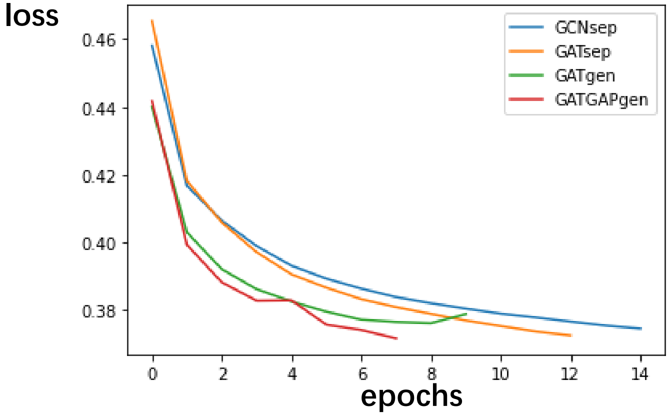
- Entropy

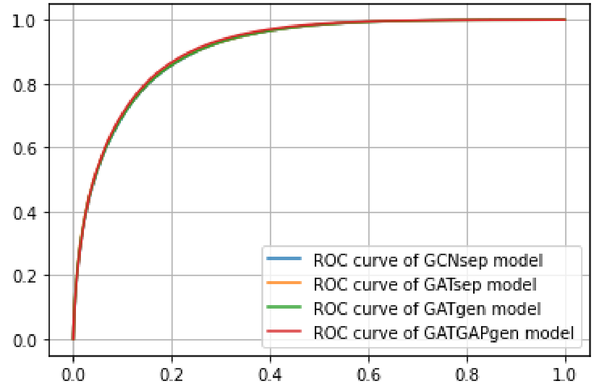
EarlyStopping:

- patience = 3
- delta = 1e-5





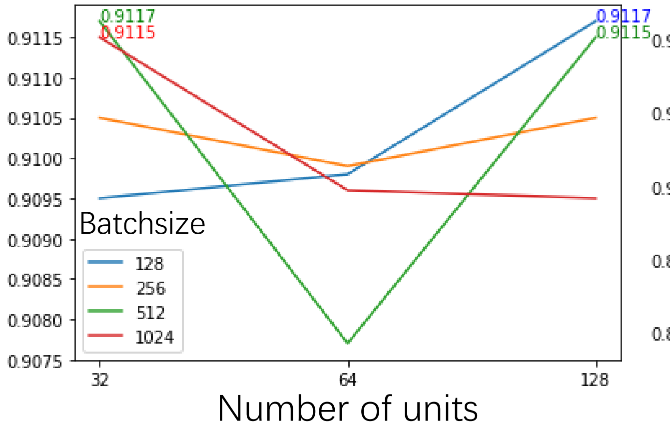




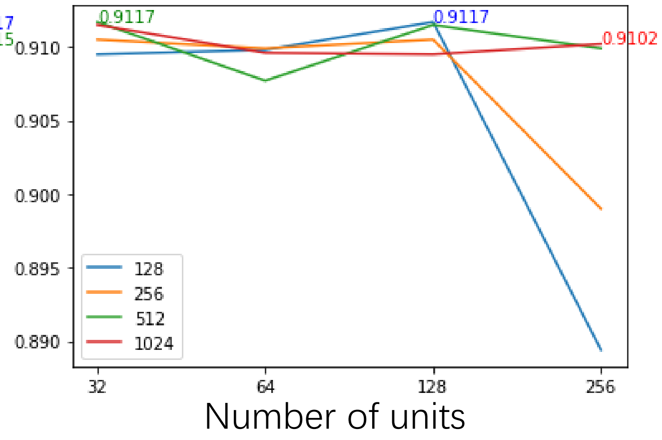
	GCN(sep)	GAT(sep)	GAT(gen)	GAT+GAP(gen)
TrainingTime	3619.46s	4047.47s	3471.48s	5049.81s
AUCValues	0.90831	0.90937	0.90891	0.91216

Grid Search

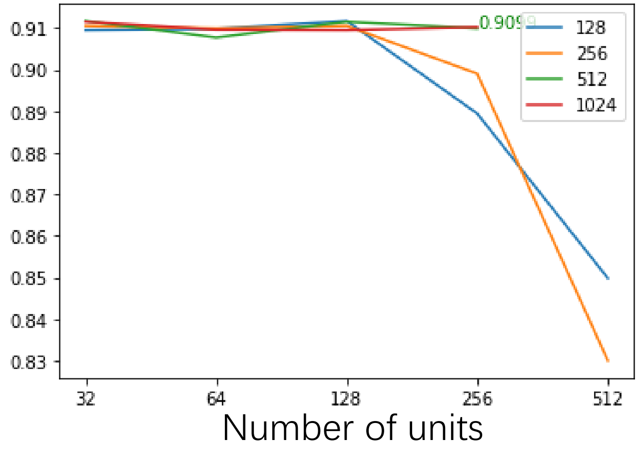
AUC



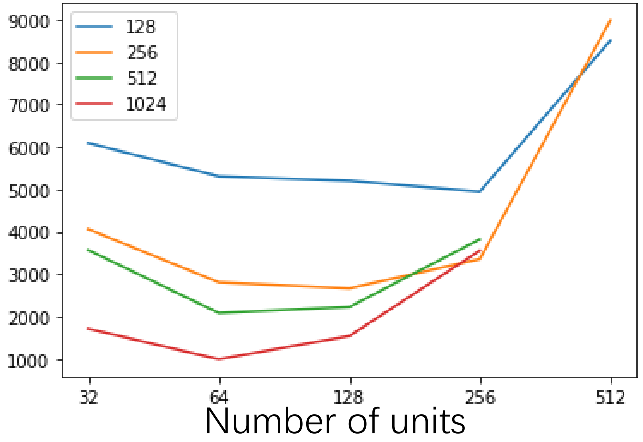
AUC



AUC



Training Time



Best Combinations

Batch-size	Number of units	AUC	Training Time
128	128	0.9117	5205
256	32	0.9105	4061
256	128	0.9105	2666
512	32	0.9117	3568
512	128	0.9115	2228
1024	32	0.9115	1716
1024	256	0.9102	3556

Network Sizes

# Units	# Parameters
32	120,527
64	459,951
128	1,808,495
256	7,184,367
512	28,651,247

Hyperparameter Optimization

Model	AUC
GCN(sep)	0.908
GAT(sep)	0.909
GAT(gen)	0.909
GATGAP(gen)	0.912

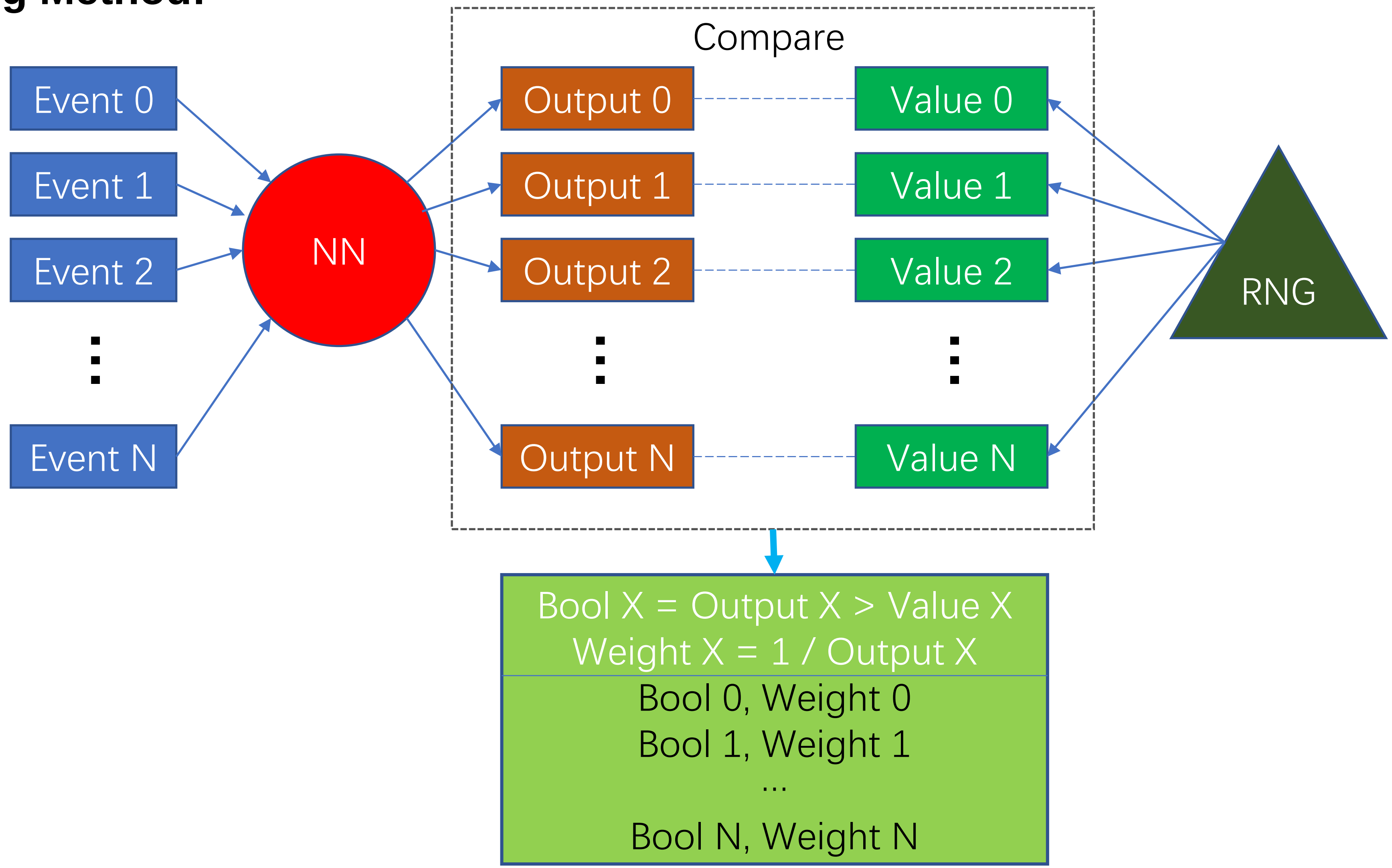
Batch Size	Number of Units	AUC	Training Time in s
128	16	0.9131	10940
512	32	0.9117	3568
128	128	0.9117	5205
1024	32	0.9115	1716
512	128	0.9115	2228
256	128	0.9115	2666
256	32	0.9115	4061

Number of Units	Number of Parameters
16	34,911
32	120,527
64	459,951
128	1,808,495

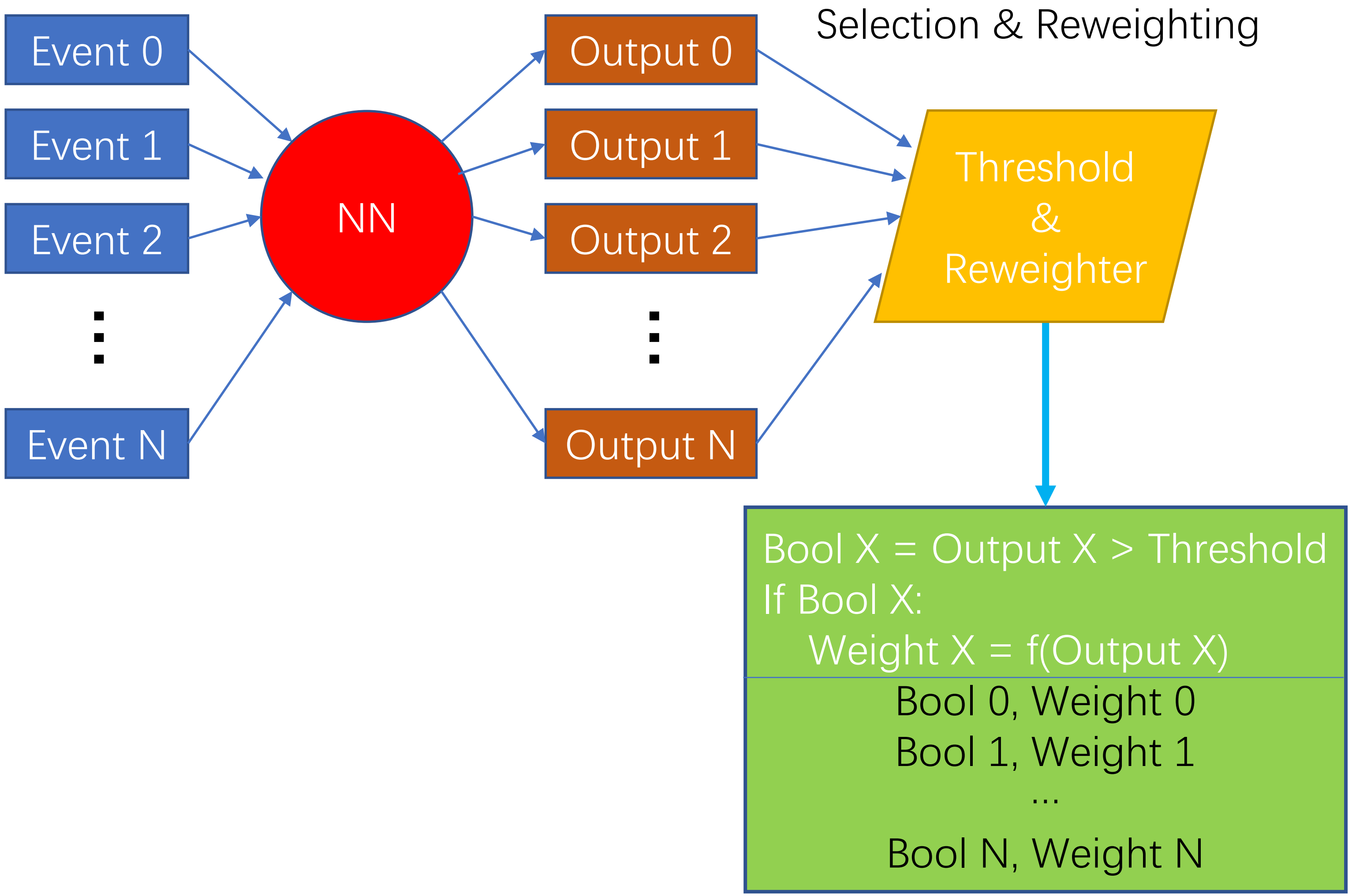
Final Configuration:

- GATGAP Model using PyTorch + Deep Graph Library (DGL)
- 6 layers with 4 attention heads each and 32 units for GAT output & global features
  -> ≈ 120k parameters
- Batch size 1024 (GPU training)

Sampling Method:



Reweighting Method:



Studied reweighters:

- GBDT Reweighting
- Histogram Reweighting

Reweighting Method:

- Train a Gradient Boosting Decision Tree (GBDT) classifier with some event level variables to distinguish between True-Positve events and False-Negative events
- GBDT Reweighting: use the outputs of the classifier directly:

$$w = \frac{1}{p_{clf}} = \frac{1}{p_{TP}/p_{TP+FN}} = \frac{p_{pass\_skim}}{p_{TP}}$$

- Histogram Reweighting: compare the score histogram of all the events that can pass the skim (True-Positive + False-Negative) with the score histogram of True-Positives to give each bin of score a scaling factor:

$$w = w_{bin_i}|_{p_{clf}\in bin_i} = \frac{H_{pass\_skim,i}}{H_{TP,i}}|_{p_{clf}\in bin_i}$$

Skim \ NN	Positive	Negative
Pass	True-Positive (TP)	False-Negative (FN)
Fail	False-Positive (FP)	True-Negative (TN)

Relative statistical uncertainty and effective sample size

Variable	Formula	Remark
NN outputs / Probabilities to pass	$\{p_i\}$	'i' refers to each event in the whole sample (batch)
Weights	$\{\omega_i\} = \left\{\frac{1}{p_i}\right\}$	Infinities (at $p_i = 0$) are excluded and set to 0 Avoid the bias by construction
Relative statistical uncertainty	$S = \frac{\sqrt{\sum \omega_i^2 p_i}}{\sum \omega_i p_i}$	$\sum \omega_i^2 p_i = \sum \omega_i$ $\sum \omega_i p_i = N$ Here consider only passed events (label = 1)
Effective sample size	$N_{eff} = \frac{1}{S^2}$	Number of events needed to reach **the same statistical uncertainty without sampling**

Speedup rate

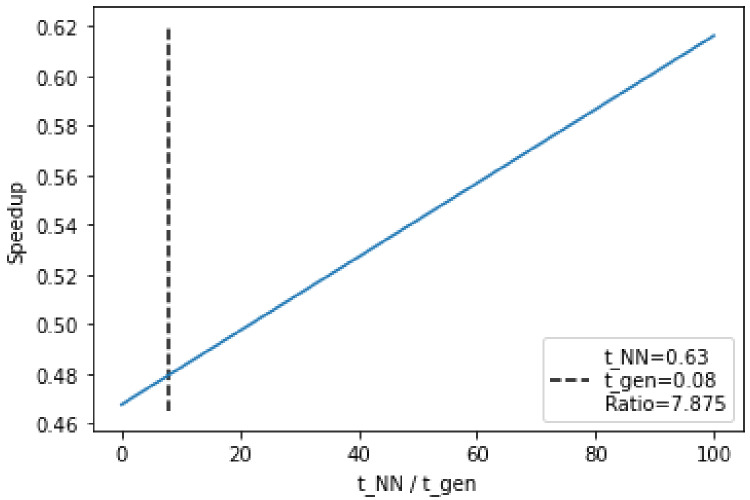
Variable	Formula	Remark
Skim retention rate	$r = 0.05$	Probability to pass the skim process
Times of different phases in ms	$t_{gen} = 0.08$ $t_{NN} = 0.63$ $t_{SR} = 97.04$	Taken from previous studies
Effective number of events after sampling	$n_+ = \sum p_i$ $n_- = \sum (1 - p_i)$	$\{p_i\}$ will be devided into two subsets where the events will/won't pass the skim process
Time consuming with NN filter	$t_+ = [n_{TP} r + n_{FP}(1-r)](t_{gen} + t_{NN} + t_{SR})$ $t_- = [n_{FN} r + n_{TN}(1-r)](t_{gen} + t_{NN})$	Positive/Negative: Result of sampling True/False: Result of sampling == skim process
Time consuming without NN	$t_0 = N_{eff}(t_{gen} + t_{NN})$	To reach the same statistical uncertainty
(Inverse) Speedup rate	$R = \frac{t_+ + t_-}{t_0}$	The lower the better

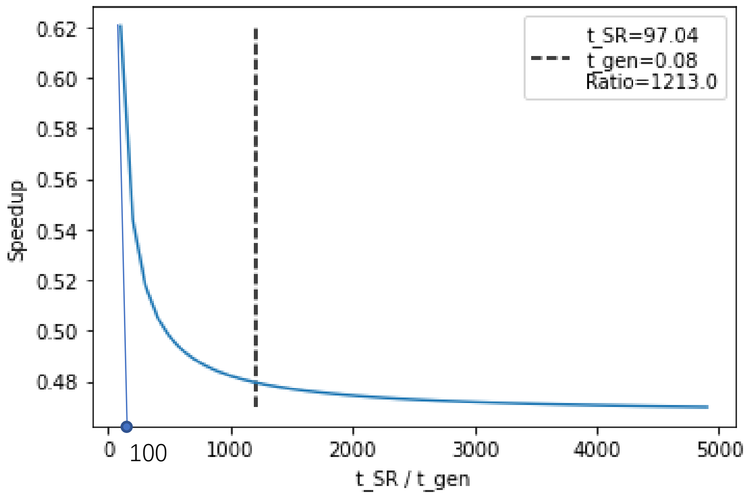
Robustness:

Weak dependency of **Speedup** on $\boldsymbol{t_{NN}}$ and $\boldsymbol{t_{SR}}$

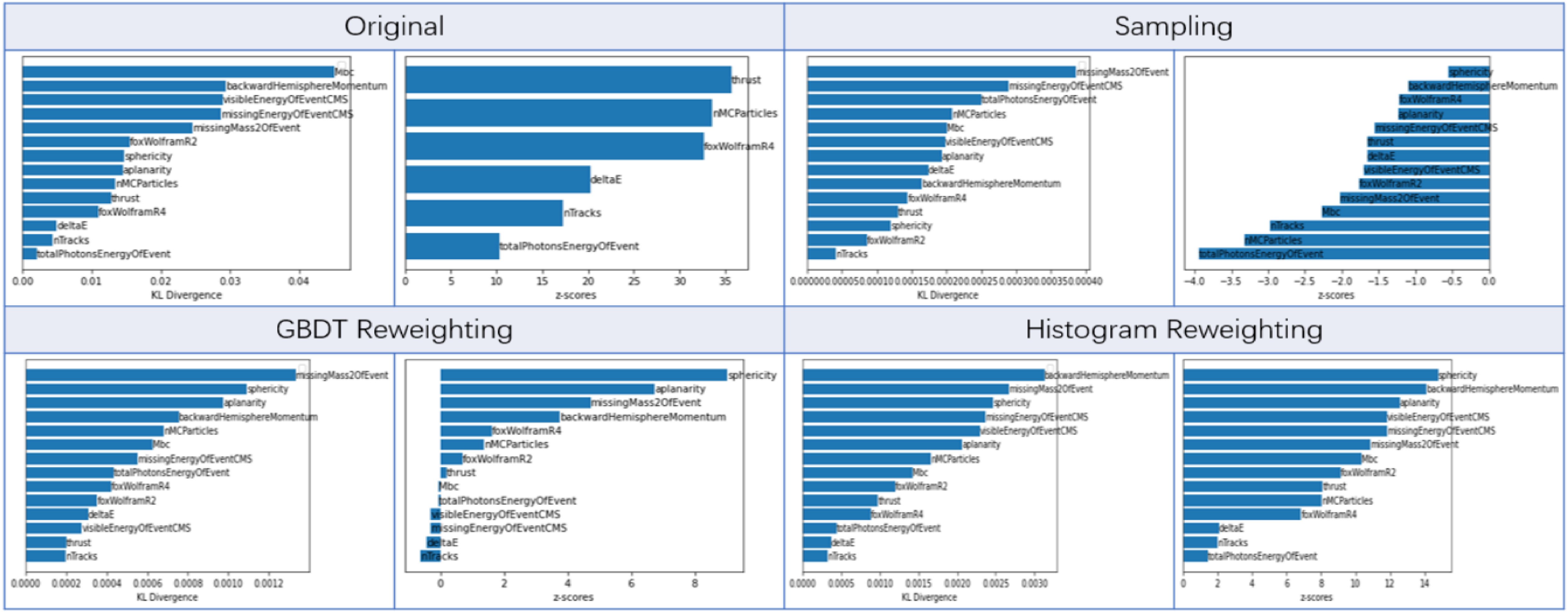
↓

Safe to generalize

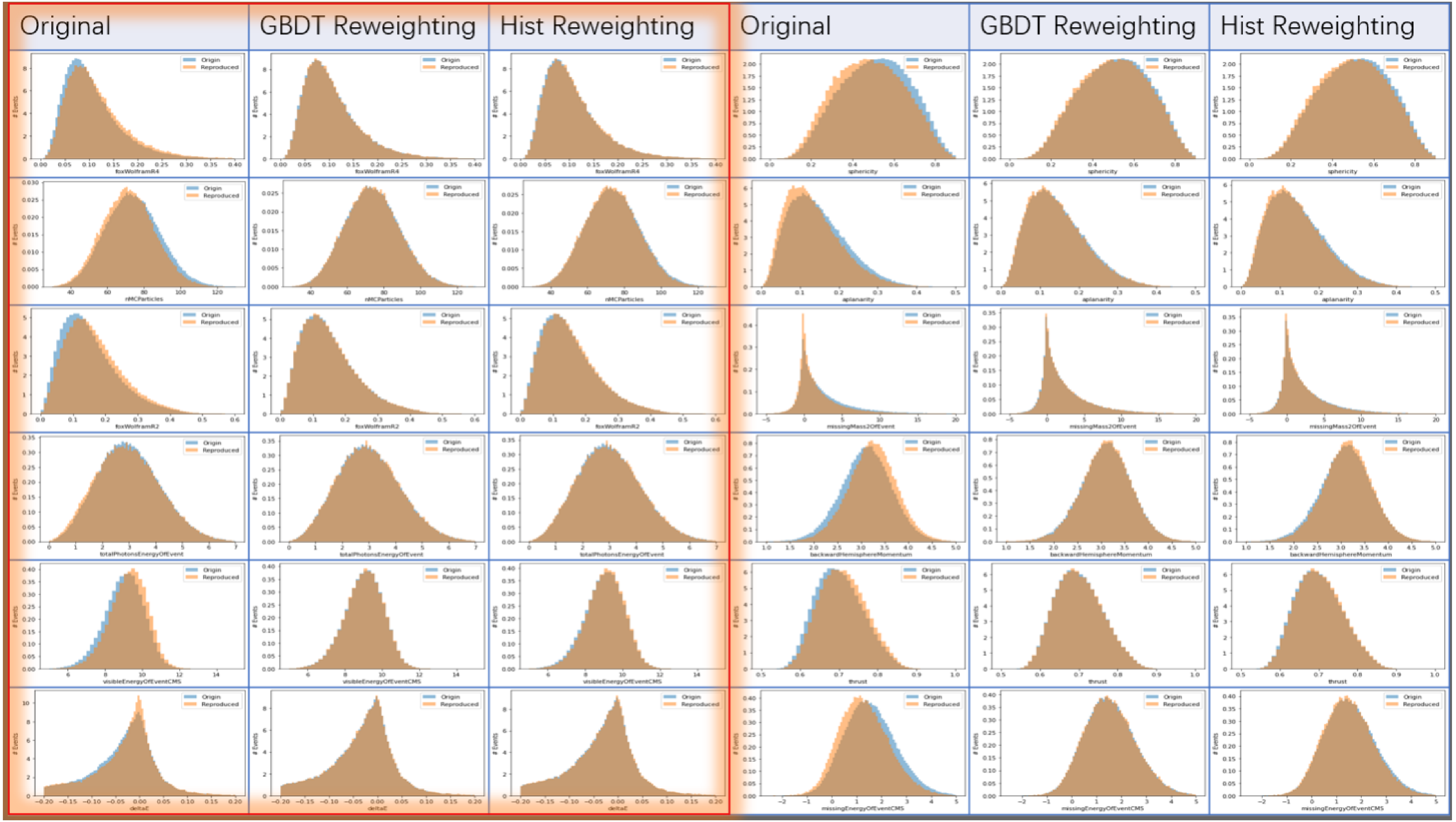


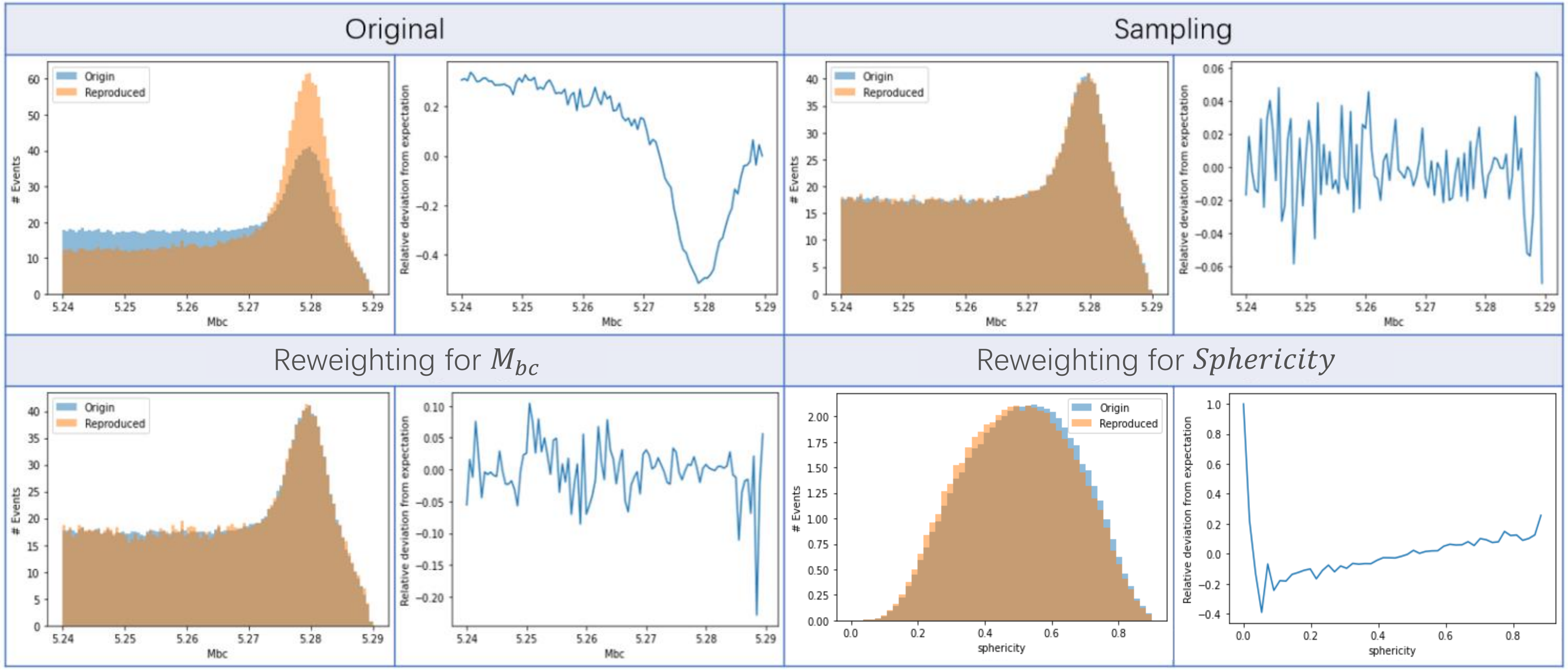


KS-Test



Original
Sampling
GBDT Reweighting
Histogram Reweighting
KL Divergence
z-scores
Mbc
backwardHemisphereMomentum
visibleEnergyOfEventCMS
missingEnergyOfEventCMS
missingMass2OfEvent
foxWolframR2
sphericity
aplanarity
nMCParticles
thrust
foxWolframR4
deltaE
nTracks
totalPhotonsEnergyOfEvent


Original
GBDT Reweighting
Hist Reweighting
Original
GBDT Reweighting
Hist Reweighting
Origin
Reproduced
Events
foxWolframR4
nMCParticles
foxWolframR2
totalPhotonsEnergyOfEvent
visibleEnergyOfEventCMS
deltaE
sphericity
aplanarity
missingMass2OfEvent
backwardHemisphereMomentum
thrust
missingEnergyOfEventCMS



Original
Sampling
Reweighting for M_{bc}
Reweighting for $Sphericity$
Origin
Reproduced
Events
Relative deviation from expectation
Mbc
sphericity

skim.WGs.ewp.inclusiveBplusToKplusNuNu

- Track cleanup:
  - p_t > 0.1
  - thetaInCDCAcceptance
  - dr<0.5 and abs(dz)<3.0
- Event cleanup:
  - 3 < nCleanedTracks < 11
- Kaon pre-cuts:
  - track cleanup + event cleanup + nPXDHits > 0
- **K+ reconstruction**
- Kaon cuts:
  - p_t rank=1
  - kaonID>0.01
- **B+ reconstruction**
- B+ cut:
  - mva_identifier: MVAFastBDT_InclusiveBplusToKplusNuNu_Skim > 0.5