Geometric Deep Learning algorithms for tau lepton identification in the ATLAS experiment

Davide Fiacco, Alessandra Betti, Stefano Giagu

Sapienza Università di Roma

Introduction: Tau ID in ATLAS

In 2019 the Recurrent Neural Network (RNN) algorithm was introduced in ATLAS for the tau lepton identification.
Two specific networks were trained, for the 1-prong and the 3-prong cases.

Based on the presentation of Tau ID of the Tau CP Workshop (20/02/2023), I retrained 4 models:
-the RNN model;
- The DeepSet model, one of proposed upgrades;
- Two new models, based on graphs: Edge-Conv NN and a Transformer.

The ntuples came from the MxAOD:
/eos/atlas/atlascerngroupdisk/perf-tau/MxAODs/R22/Run2repro/TauID/

R22 Variables

These are the variables used in this work, labeled as R22.

1p Variables	1p Variables	1p Variables
"TauJets/centFrac"	"TauTracks/pt_log"	"TauClusters/et_log"
"TauJets/etOverPtLeadTrk"	"TauTracks/pt_tau_log"	"TauClusters/pt_tau_log"
"TauJets/dRmax"	"TauTracks/dEta"	"TauClusters/dEta"
"TauJets/SumPtTrkFrac"	"TauTracks/dPhi"	"TauClusters/dPhi"
"TauJets/EMPOverTrkSysP"	"TauTracks/nInnermostPixelHits"	"TauClusters/SECOND_R"
"TauJets/ptRatioEflowApprox"	"TauTracks/nPixelHits"	"TauClusters/SECOND_LAMBDA"
"TauJets/mEflowApprox"	"TauTracks/nSCTHits"	"TauClusters/CENTER_LAMBDA"
"TauJets/pt_tau_log"	"TauTracks/z0sinthetaTJVA"	
Additional Variables for 3p case	"TauTracks/z0sinthetaSigTJVA"	
"TauJets/trFlightPathSig"	"TauTracks/d0TJVA"	
"TauJets/massTrkSys"	"TauTracks/d0SigTJVA"	

R22 Variables

These are the variables used in this work, labeled as R22.

1p Variables	1p Variables	1p Variables
"TauJets/centFrac"	"TauTracks/pt_log"	"TauClusters/et_log"
"TauJets/etOverPtLeadTrk"	"TauTracks/pt_tau_log"	"TauClusters/pt_tau_log"
"TauJets/dRmax"	"TauTracks/dEta"	"TauClusters/dEta"
"TauJets/SumPtTrkFrac"	"TauTracks/dPhi"	"TauClusters/dPhi"
"TauJets/EMPOverTrkSysP"	"TauTracks/nInnermostPixelHits"	"TauClusters/SECOND_R"
"TauJets/ptRatioEflowApprox"	"TauTracks/nPixelHits"	"TauClusters/SECOND_LAMBDA"
"TauJets/mEflowApprox"	"TauTracks/nSCTHits"	"TauClusters/CENTER_LAMBDA"
“TauJets/pt_tau_log”	"TauTracks/z0sinthetaTJVA"	
Additional Variables for 3p case	"TauTracks/z0sinthetaSigTJVA”	
“TauJets/trFlightPathSig”	"TauTracks/d0TJVA"	
"TauJets/massTrkSys”	"TauTracks/d0SigTJVA"	

These gives global information of the tau candidate, so also called as ‘Global’ variables

R22 Variables

These are the variables used in this work, labeled as R22.

1p Variables	1p Variables	1p Variables
"TauJets/centFrac"	"TauTracks/pt_log"	"TauClusters/et_log"
"TauJets/etOverPtLeadTrk"	"TauTracks/pt_tau_log"	"TauClusters/pt_tau_log"
"TauJets/dRmax"	"TauTracks/dEta"	"TauClusters/dEta"
"TauJets/SumPtTrkFrac"	"TauTracks/dPhi"	"TauClusters/dPhi"
"TauJets/EMPOverTrkSysP"	"TauTracks/nInnermostPixelHits"	"TauClusters/SECOND_R"
"TauJets/ptRatioEflowApprox"	"TauTracks/nPixelHits"	"TauClusters/SECOND_LAMBDA"
"TauJets/mEflowApprox"	"TauTracks/nSCTHits"	"TauClusters/CENTER_LAMBDA"
“TauJets/pt_tau_log”	"TauTracks/z0sinthetaTJVA"	
Additional Variables for 3p case	"TauTracks/z0sinthetaSigTJVA”	
“TauJets/trFlightPathSig”	"TauTracks/d0TJVA"	
"TauJets/massTrkSys”	"TauTracks/d0SigTJVA"	

They describe each track of the tau candidate, so also called as ‘Track’ variables

R22 Variables

These are the variables used in this work, labeled as R22.

1p Variables	1p Variables	1p Variables
"TauJets/centFrac"	"TauTracks/pt_log"	"TauClusters/et_log"
"TauJets/etOverPtLeadTrk"	"TauTracks/pt_tau_log"	"TauClusters/pt_tau_log"
"TauJets/dRmax"	"TauTracks/dEta"	"TauClusters/dEta"
"TauJets/SumPtTrkFrac"	"TauTracks/dPhi"	"TauClusters/dPhi"
"TauJets/EMPOverTrkSysP"	"TauTracks/nInnermostPixelHits"	"TauClusters/SECOND_R"
"TauJets/ptRatioEflowApprox"	"TauTracks/nPixelHits"	"TauClusters/SECOND_LAMBDA"
"TauJets/mEflowApprox"	"TauTracks/nSCTHits"	"TauClusters/CENTER_LAMBDA"
“TauJets/pt_tau_log”	"TauTracks/z0sinthetaTJVA"	
Additional Variables for 3p case	"TauTracks/z0sinthetaSigTJVA”	
“TauJets/trFlightPathSig”	"TauTracks/d0TJVA"	
"TauJets/massTrkSys”	"TauTracks/d0SigTJVA"	

They describe each cluster of the tau candidate, so also called as ‘Cluster’ variables

Dataset

The used dataset is composed of:

		Training	Testing
1-prong	signal	2.8 x 10^6	5.5 x 10^6
	background	4.9 x 10^6	12.3 x 10^6
3-prong	signal	1.5 x 10^6	1.5 x 10^6
	background	5.9 x 10^6	18.5 x 10^6

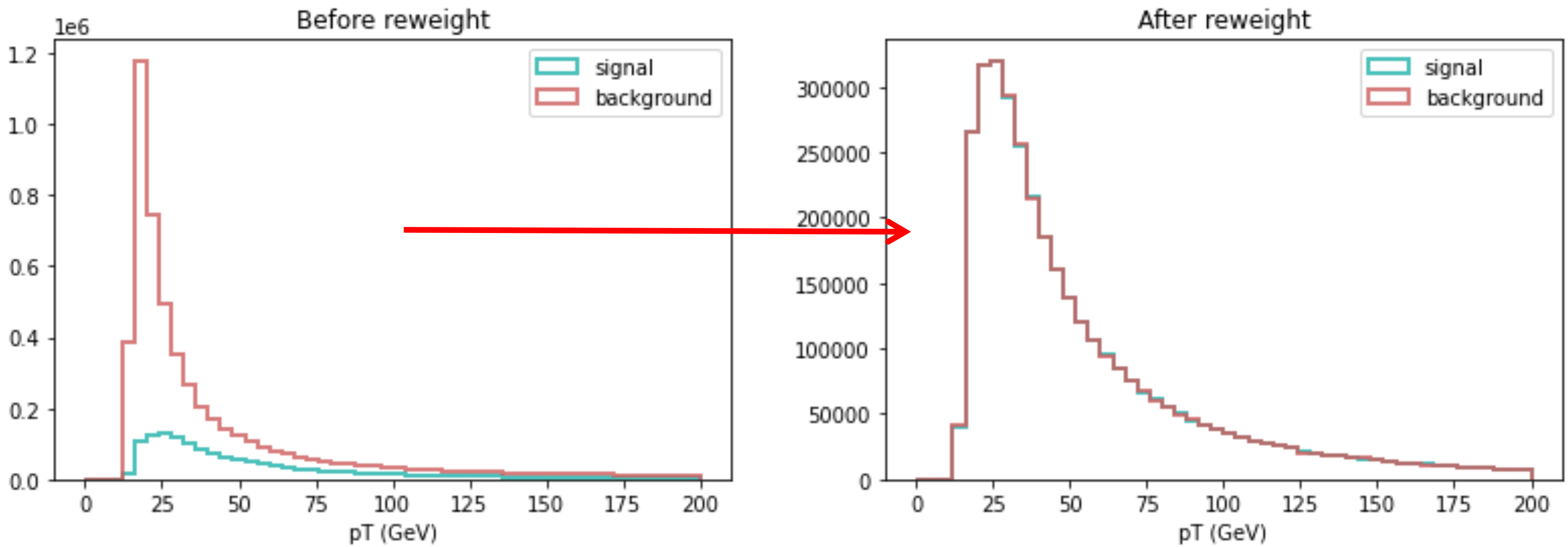
The 15% of training events is used for the Validation step.

Preprocessing and reweight

Each Tau-jet can have a max of 10 tracks and 6 clusters, ordered by energy.

I executed the same preprocessing code used for RNN and DeepSet.

Jet weight of training events is obtained by pt reweight * beamSpotWeight variable, and because the two classes are not balanced, are also computed class_weights.

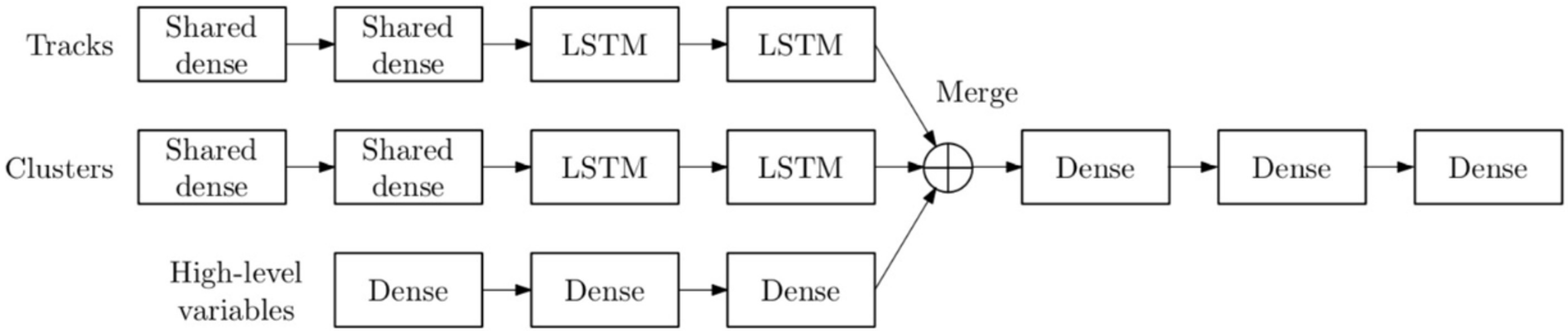


In the following slides they will be presented the 4 studied models of this work.

Recurrent Neural Network

State-of-the-art in ATLAS is given by the **Recurrent Neural Network** (**RNN**) (ATL-PHYS-PUB-2022-044).

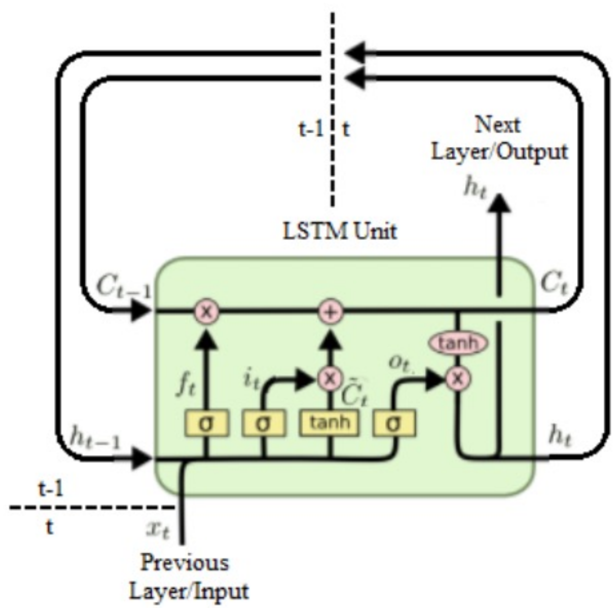
It processes the three inputs separately, and merge them togheter.



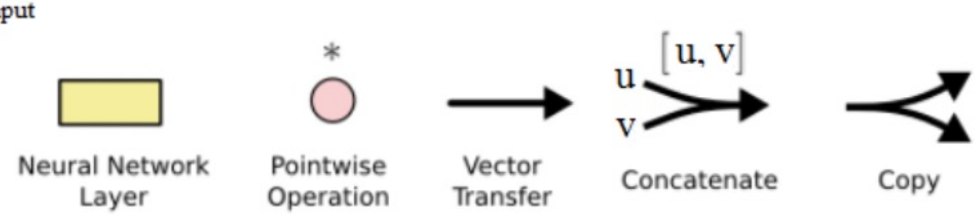
LSTM layer

An LSTM layer is an RNN layer that learns long-term dependencies between time steps in time series and sequence data.

Studying the correlations between tracks and clusters
Their correlation the network learns how to select signal.



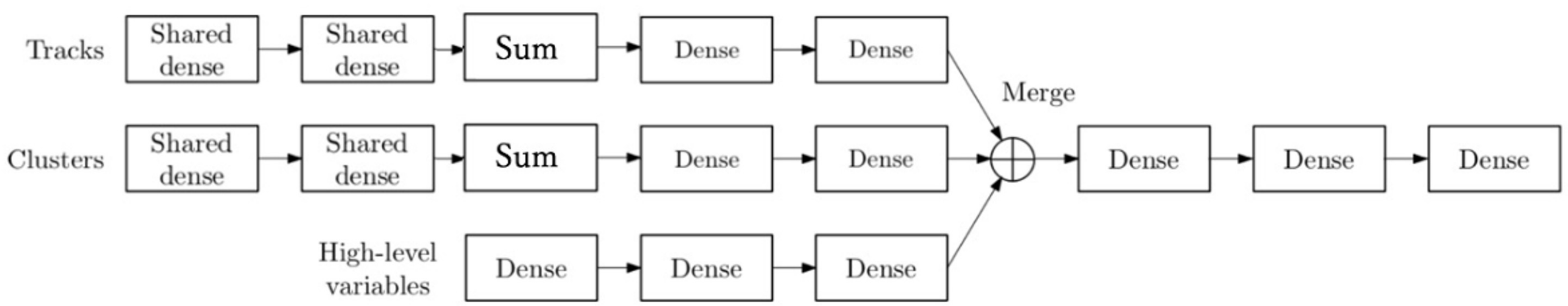
$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \;+\; b_f\right)$$
$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \;+\; b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \;+\; b_C)$$
$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$
$$o_t = \sigma\left(W_o \; [h_{t-1}, x_t] \;+\; b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$



DeepSet

It is an extremely faster and lighter model, which is one of the main possible upgrades for the tau identification in ATLAS.

It processes the three inputs separately, and merge them togheter.



Geometric Deep Learning models

Geometric ML models are **NN that take graphs as input**.

They **exploit the geometric structure** of the input to extract powerful features.

Geometric Deep Learning models

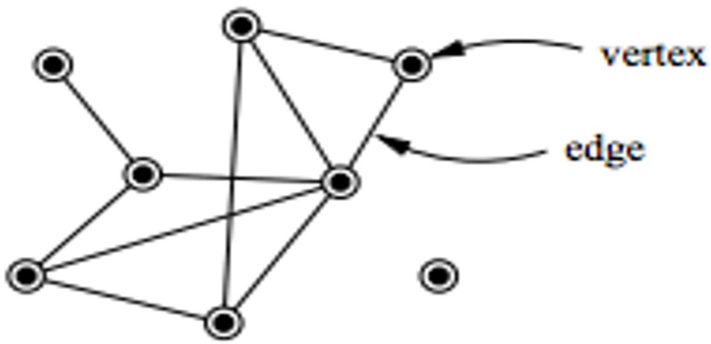
Geometric ML models are **NN that take graphs as input**.

They **exploit the geometric structure** of the input to extract powerful features.

How is the graph computed?

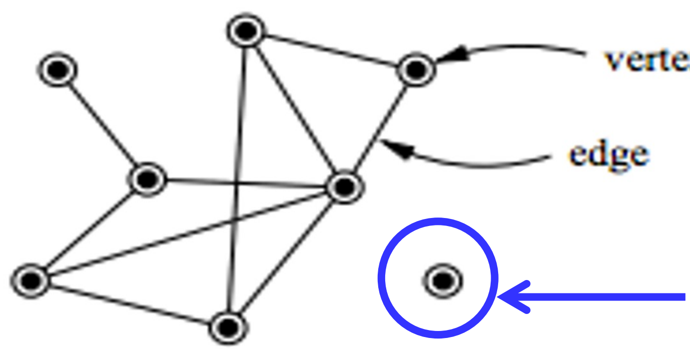
The Dataset: Definition of the Graph

Given a set of N points, the **graph** $\boldsymbol{G = (V, E)}$ is given by **vertices** $\boldsymbol{V = \{1,\ldots,n\}}$ and **edges** $\boldsymbol{E \subseteq V \times V}$



The Dataset: Definition of the Graph

Given a set of N points, the **graph *G* = (*V*, *E*)** is given by **vertices *V* = {1,...,n}** and **edges $E \subseteq V \times V$**

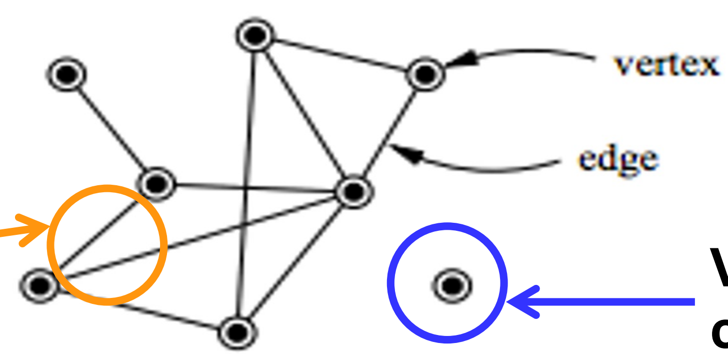


Vertices are tracks and clusters;

Their position is given by angular coordinates (η,ϕ)

The Dataset: Definition of the Graph

Given a set of N points, the **graph G = (V, E)** is given by **vertices V = {1,...,n}** and **edges $E \subseteq V \times V$**



Edges couple each node with its k nearest neighbours;

Vertices are tracks and clusters;

Their position is given by angular coordinates (η,ϕ)

Distance Function: $\Delta R = \sqrt{(\Delta\eta)^2 + (\Delta\phi)^2}$

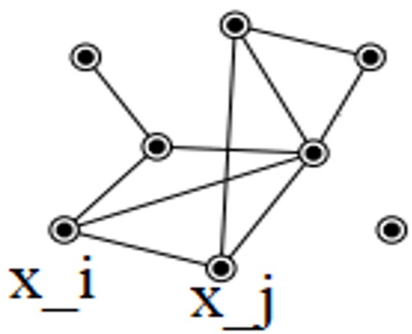
k is given by user (e.g. 3,8,..)

First proposed model:The EdgeConv-NN

It takes the graph and the global variables

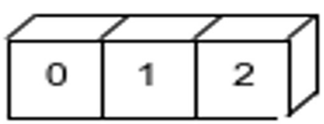
Layers that identify t

Local variables as a graph:
-i runs on nodes;
-j runs on k-nn of each node;



Output ϵ [0,1]

Global variables

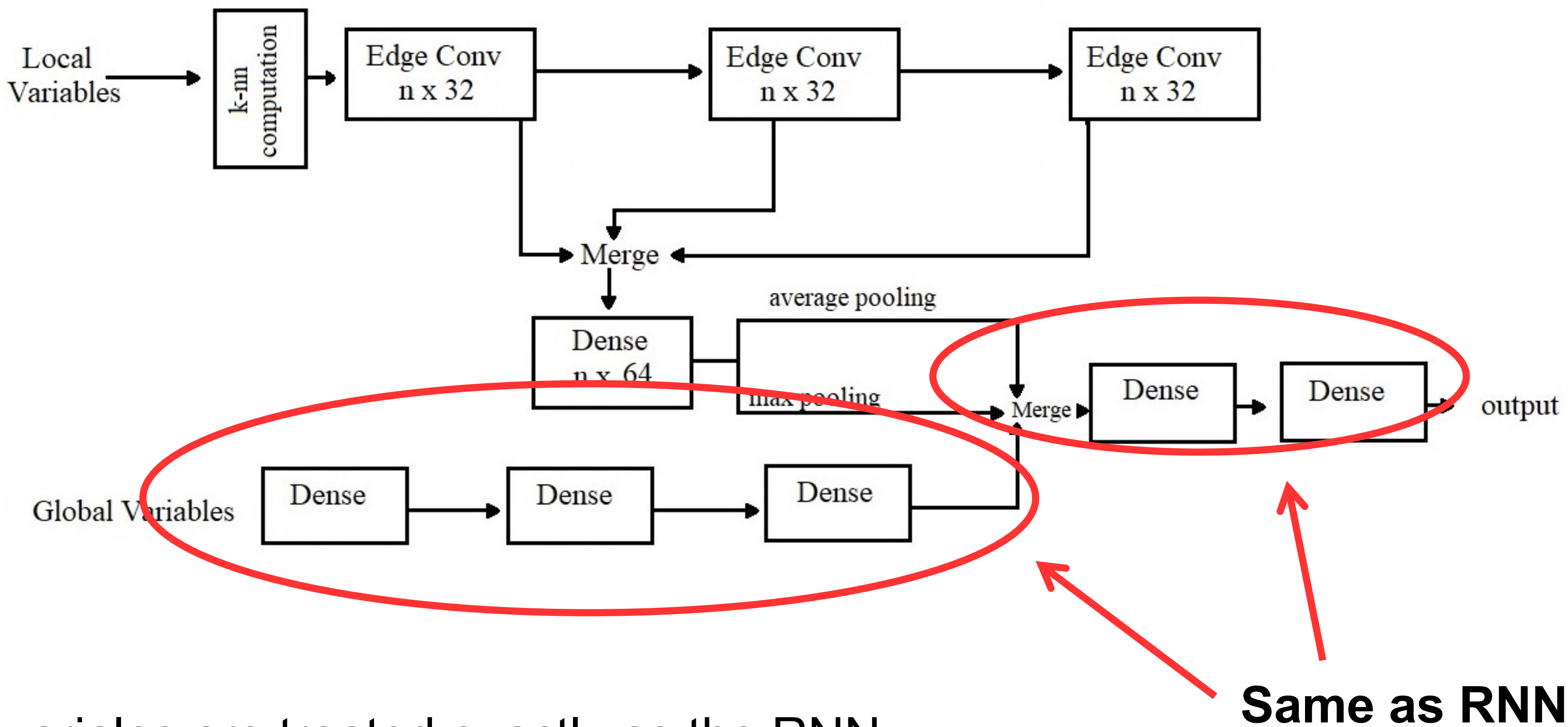


Same as RNN

First proposed model:The EdgeConv-NN

It takes the graph and the global variables.

The Local variales are the feature matrix of the nodes.
After the computation of k-nn, on the edges it is applied the edge-convolution operation.

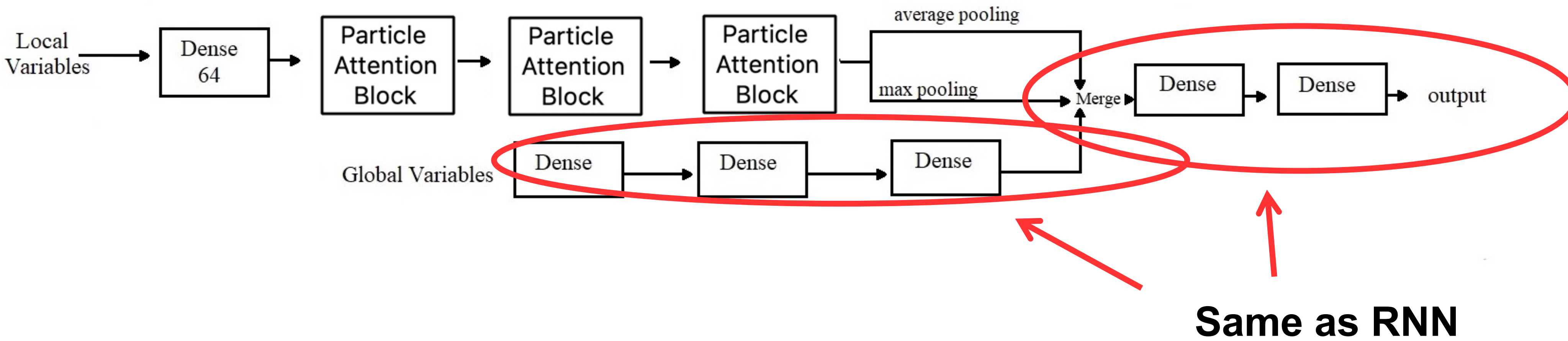


The Global variales are treated exactly as the RNN

Second Proposed Model: The Transformer

It takes the graph and the global variables

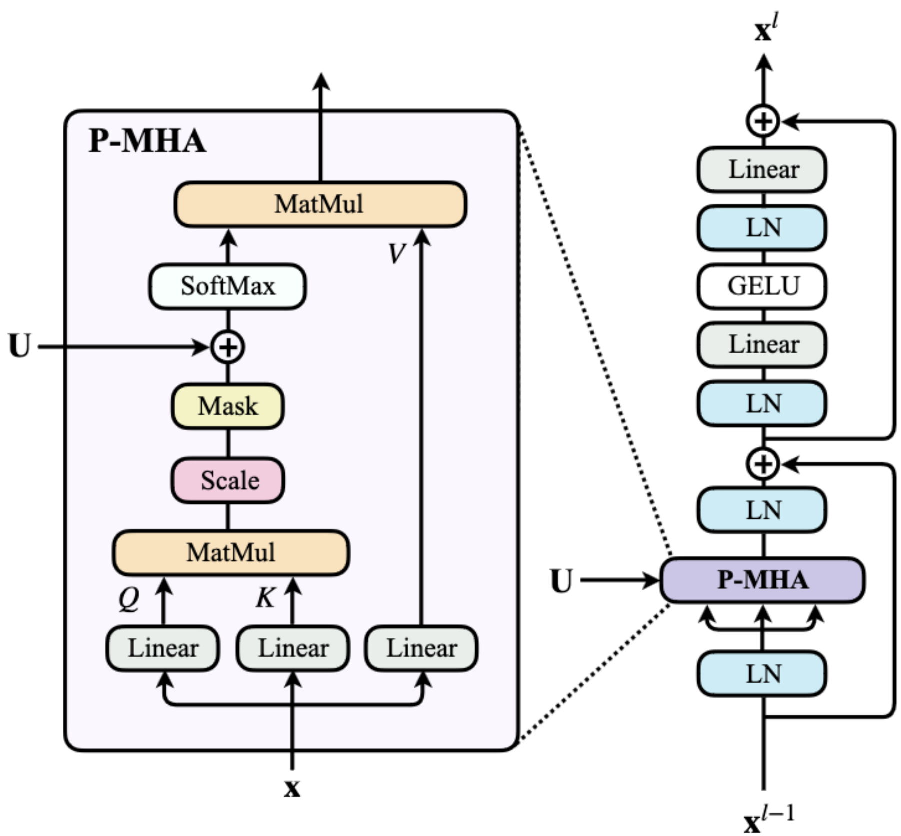
The Local variables are the feature matrix of the nodes.
They are treated as a fully connected graph, on which it is applied the Particle Attention Blocks.



The Global variales are treated exactly as the RNN

Particle Attention Block

I implemented the Particle Transformer Block (Figure a), without the application of the mask and the Particle-Multi Head Attention layer (Figure b).



(b) Particle Attention Block

All studied networks

In this work four different models have been trained with same dataset.
Their number of parameters is shown by the table:

	Number of parameters
RNN	56k
DeepSet	34k
Edge-Conv NN	41k
Transformer	307k

Performance: The execution time

-Trained on **simulated tau candidates** and **QCD background**

<u>All models are trained with the same batchsize</u>

*Single epoch time can vary a lot during trainings, total time is more stable.

	R22 1-prong		R22 3-prong	
	*Time of epoch	Tot training time (Num epochs)	*Time of epoch	Tot training time (Num epochs)
RNN	418 s	5.4 h (44 n. ep.)	404 s	3.7 h (27 n. ep.)
DeepSet	108 s	43 m (14 n. ep.)	103 s	1 h (21 n. ep.)
Edge-conv NN	224 s	1.8 h (25 n. ep.)	203 s	1.4 h (20 n. ep.)
Transformer	246 s	2.3 h (30 n. ep.)	230 s	2.3 h (31 n. ep.)

Performance: The execution time

Simulation time for process 6Million signal events:

	Events processed per sec.	Minutes
RNN	85k	1.45
DeepSet	300k	0.70
Edge-Conv NN	70k *	1.80
Transformer	100k	1.43

*In the EdgeConv-NN algorithm there is also the construction of the k-NN graph

The ROC curve

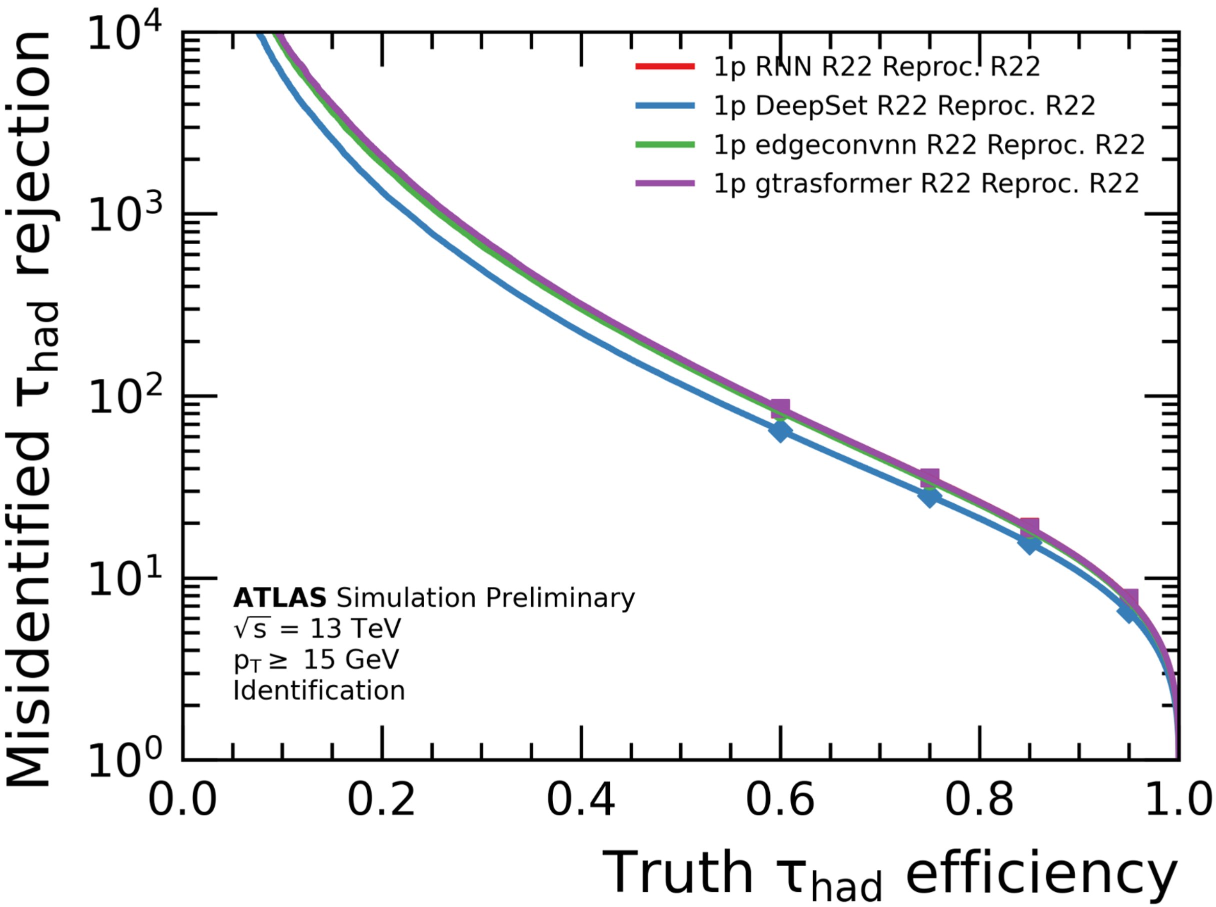
The output of the networks is a variable distributed between 0 and 1, from which it has to be extracted the label of the event.

Varying the threshold on this score, it's possible to study the efficiency on the signal and the rejection power of the background.

$$\text{Rejection Power} = \frac{1}{\text{background selection efficiency}}$$

R22 Performance: ROC curve

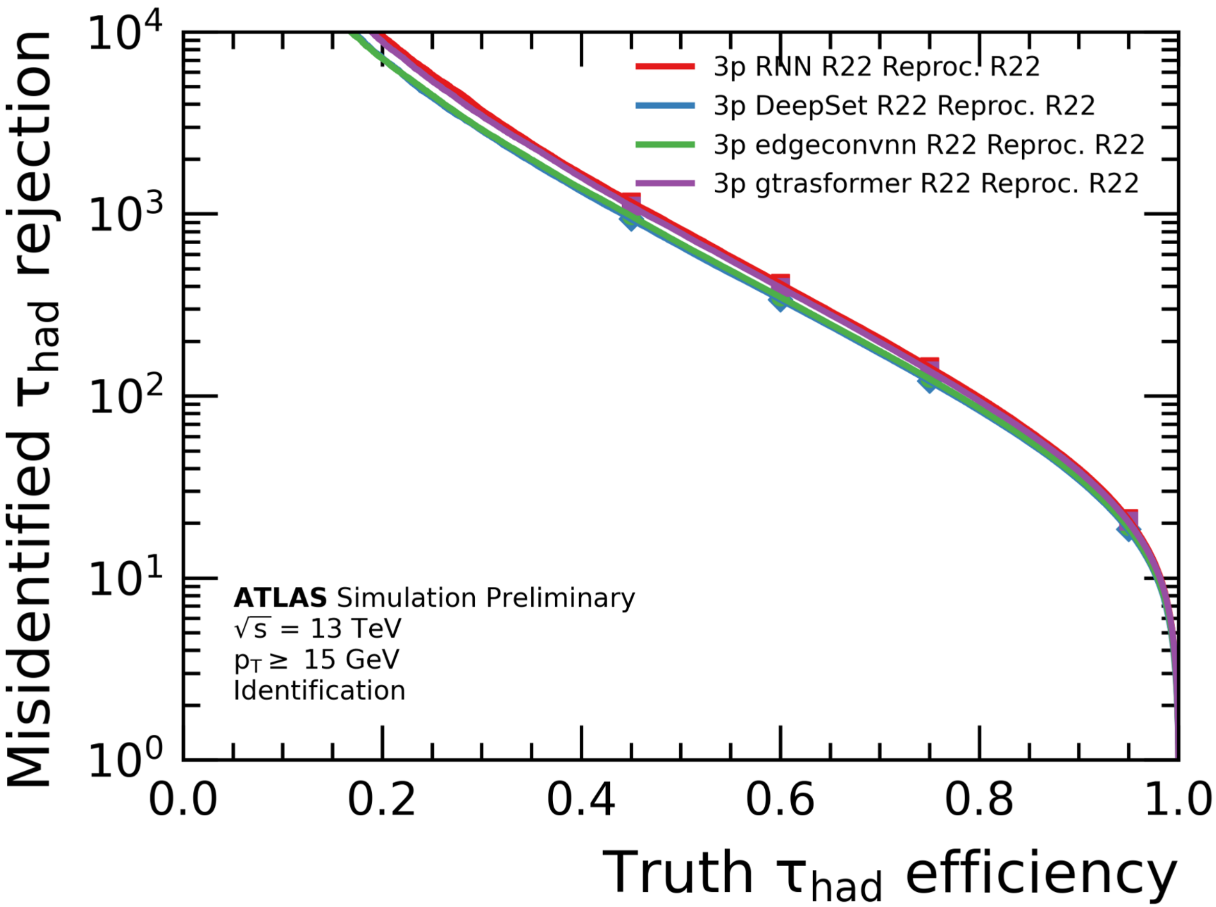
Very similar rejection for Transformer, Edge-convNN, RNN.



WP	0.6	0.75	0.85	0.95
RNN	85	35	19	7.7
DPS	65	28	16	6.6
E-C NN	82	34	18	7.5
Tran.	85	35	19	7.7

R22 Performance: ROC curve

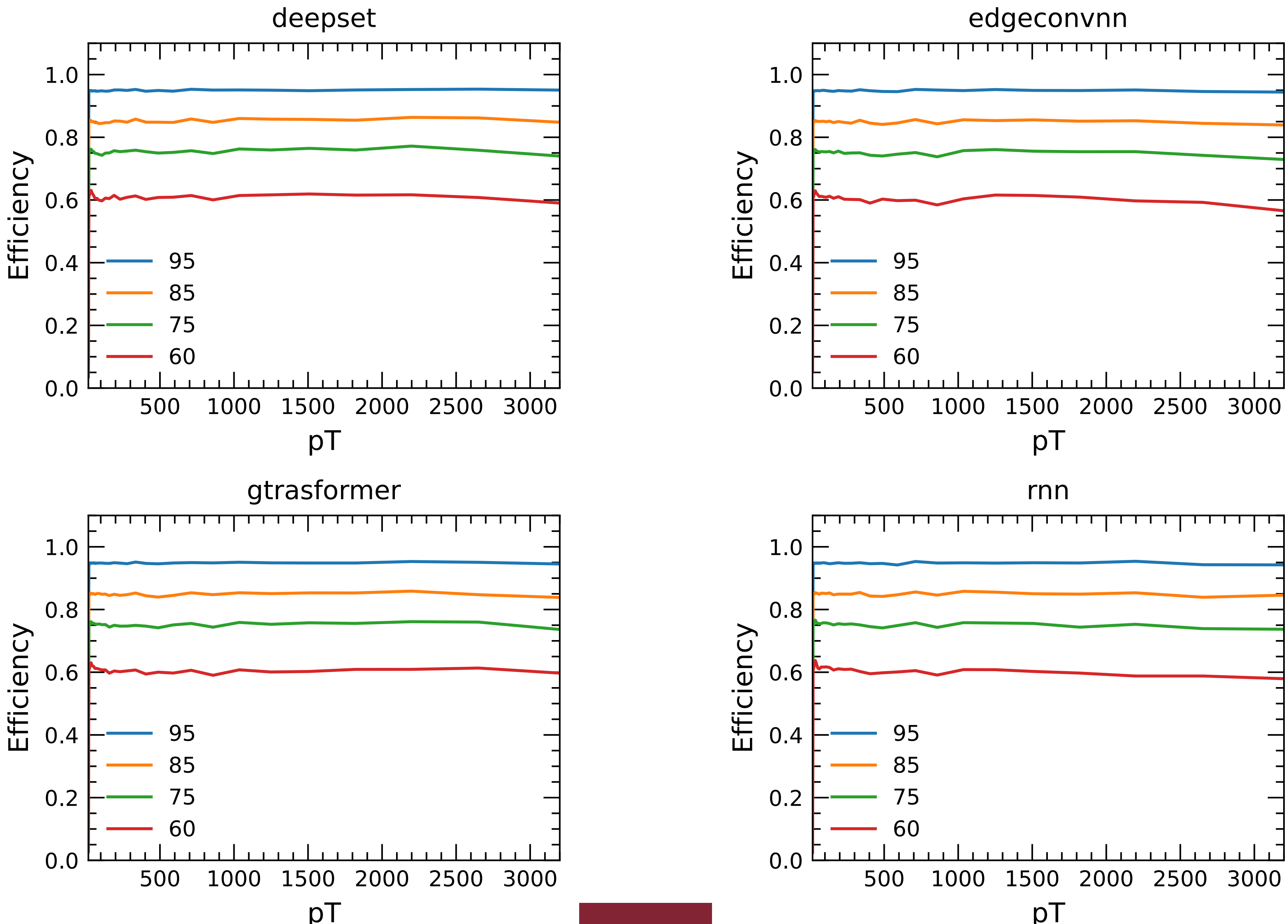
Best rejection is given by RNN, than Transformer, EdgeConvNN and DeepSet.



WP	0.45	0.6	0.75	0.95
RNN	1168	416	144	21
DPS	938	339	121	18
E-C NN	978	350	124	19
Tran.	1105	396	138	20

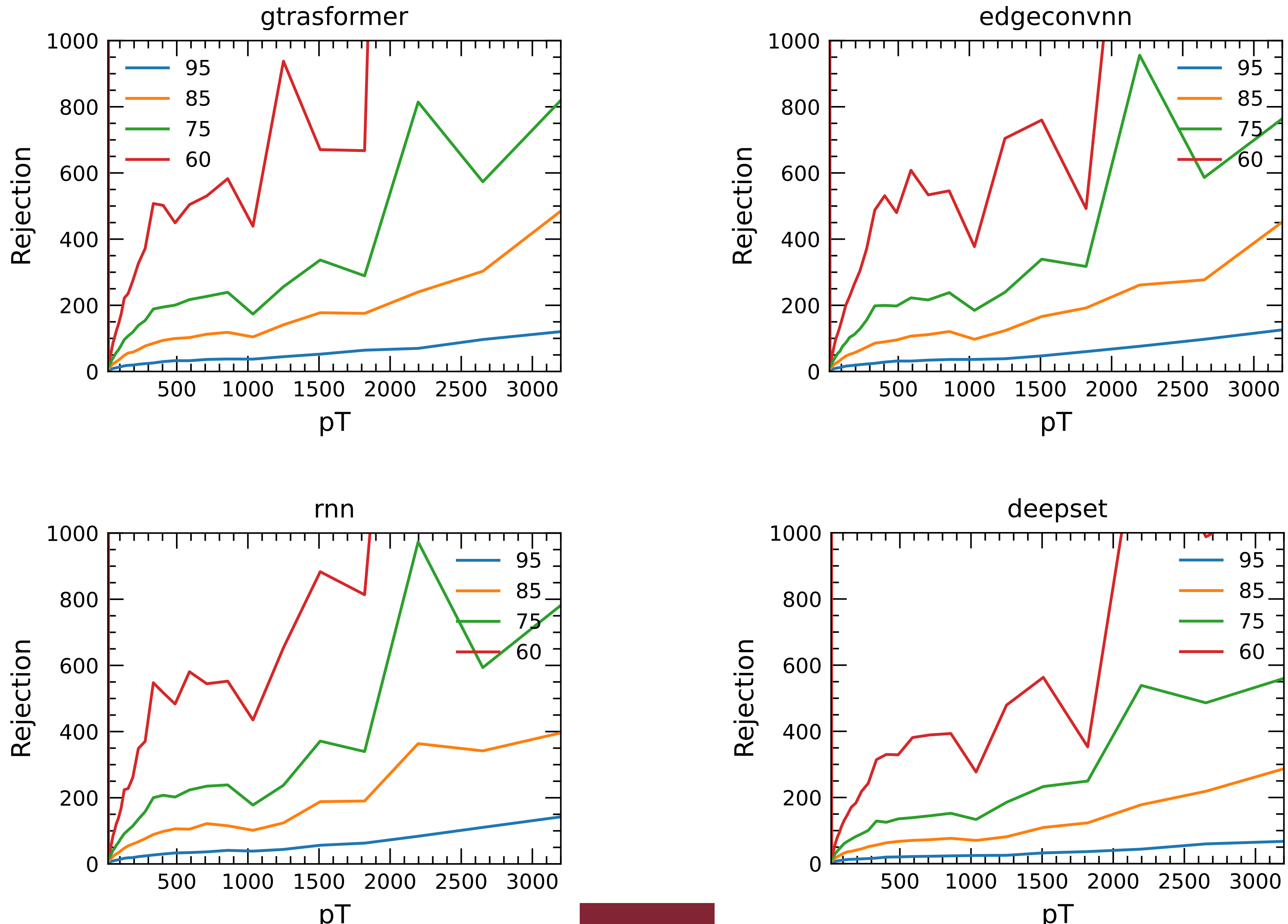
Efficiency for 1-prong case

These efficiencies are estimated from a flattened sample of events.



Efficiency for 1-prong case

These efficiencies are estimated from a flattened sample of events.



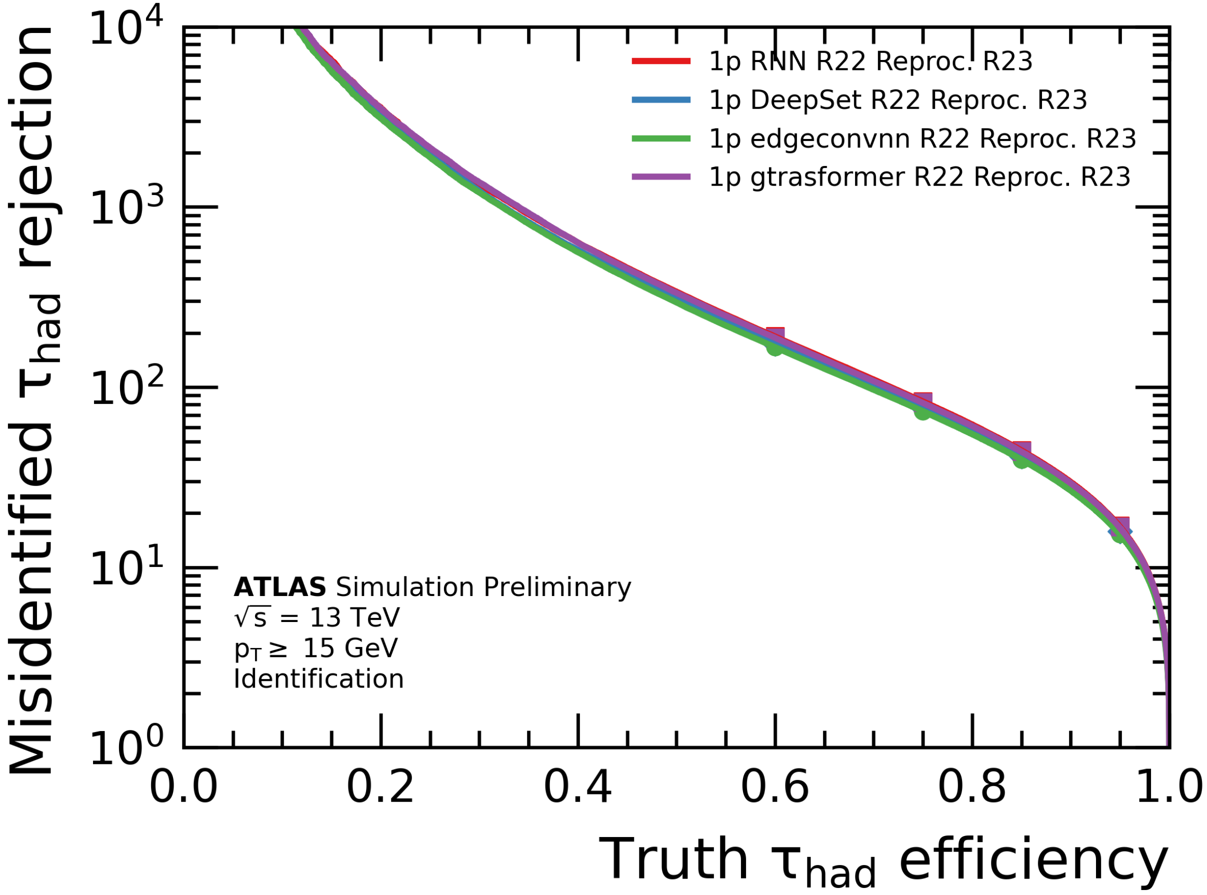
R23 Variables

The R23 variables contain all R22 ones togheter with the output given by the Track Classification of the RNN.
The new variables will be added as Track variables, and are the following:
(both for 1 and 3 prong)

Track RNN Variables
"TauTracks/chargedScoreRNN"
"TauTracks/isolationScoreRNN"
"TauTracks/conversionScoreRNN"

R23 Performance: ROC curve

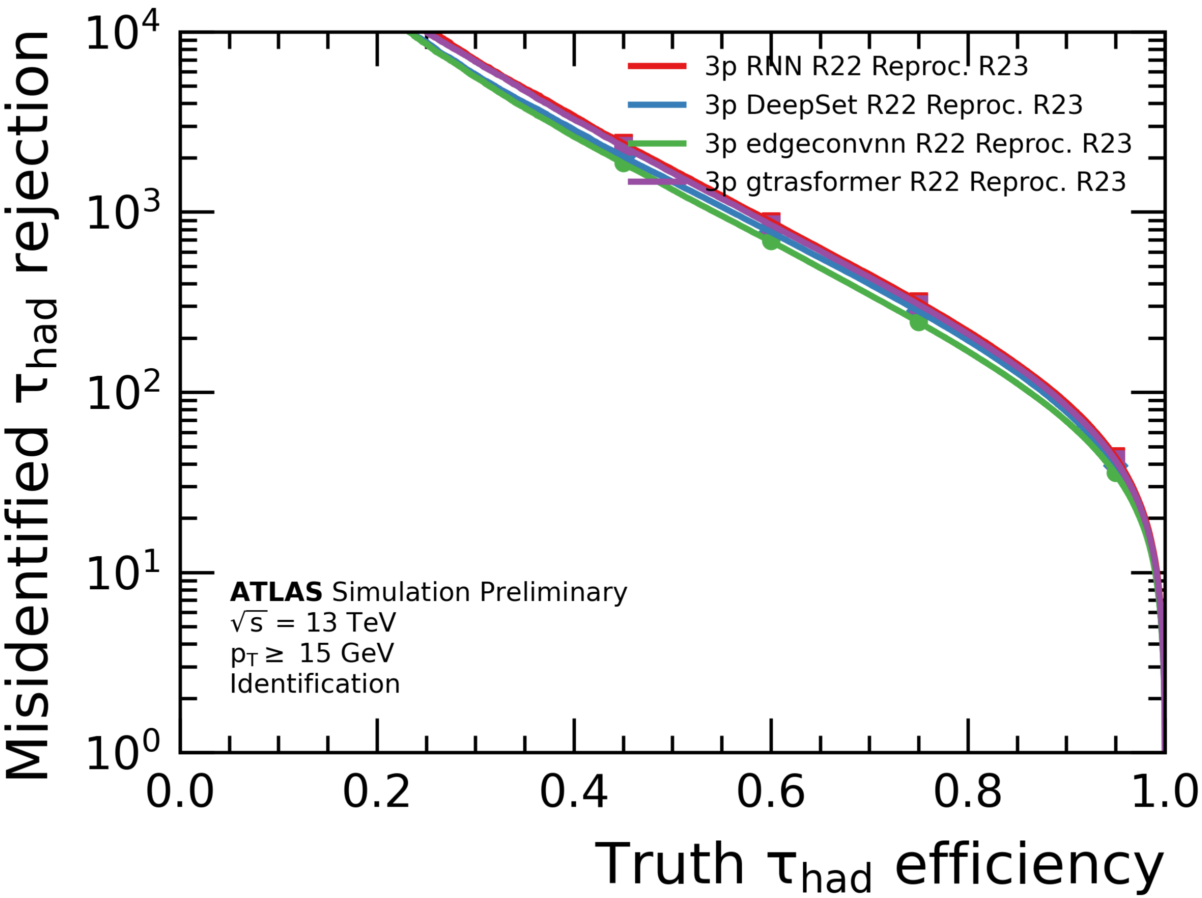
The RNN Track Classification variables have a central role.
All models have an improvement.
Similar rejection for Transformer and RNN, and for DeepSet and EdgeConvNN.



WP	0.45	0.6	0.75	0.95
RNN	191	83	45	17
DPS	174	77	41	16
E-C NN	167	74	40	15
Tran.	188	82	44	17

R23 Performance: ROC curve

In this case the best model is the RNN, than the Transformer, DeepSet and the EdgeConvNN.



WP	0.45	0.6	0.75	0.95
RNN	2410	888	319	44
DPS	2037	777	283	39
E-C NN	1868	690	246	36
Tran.	2318	851	307	42

Conclusions...

- **Graph rapresentation** can contain **information** in a more efficient way;

- Geometric deep learning models can be good candidates to improve tau identification in ATLAS;

- Proposed two models: **EdgeConv-NN** and a **Transformer.**

… and Future studies

- A deeper study of input variables is needed, to search some new discriminant information;

- Apply these new models on other tasks, as Track Classification or Tau Decay Classification, to test their bigger expressive power;

- Develop a Fast Track Pre-selector;

Tau Track Classification

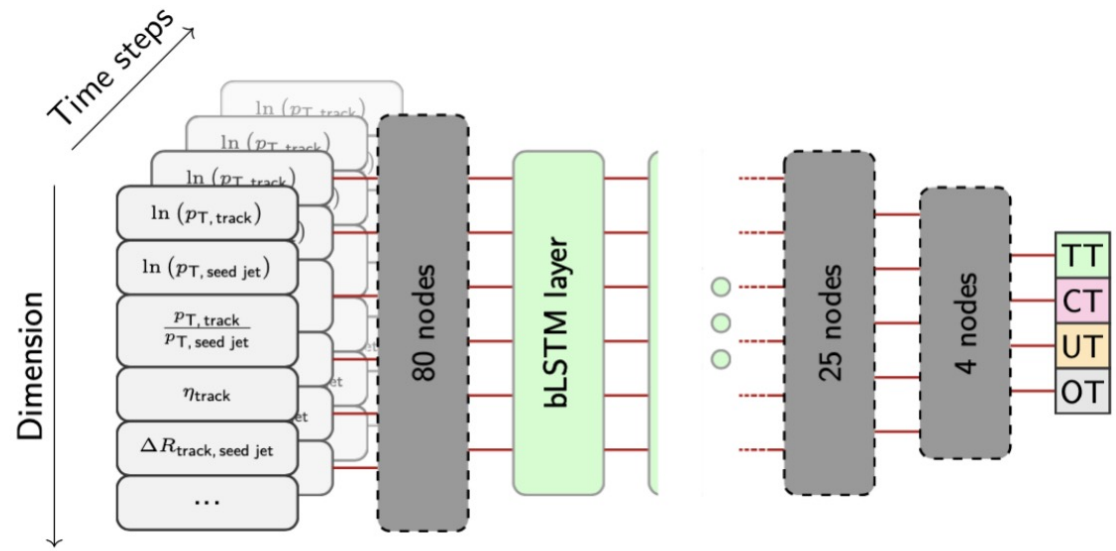
This step is performed before of the Tau reconstruction, to Estimate the prongness of tau candidate;

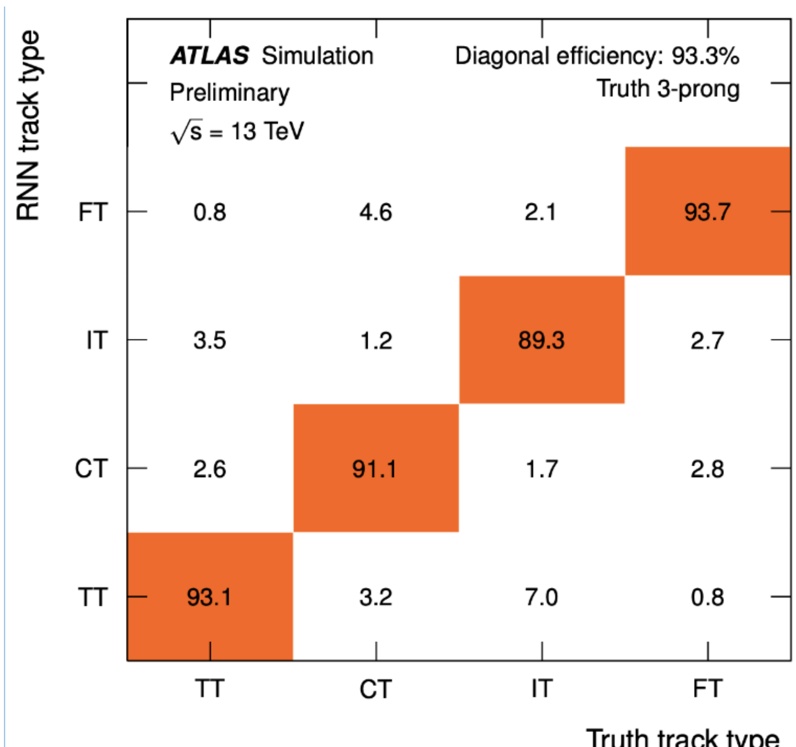
4 track labels:

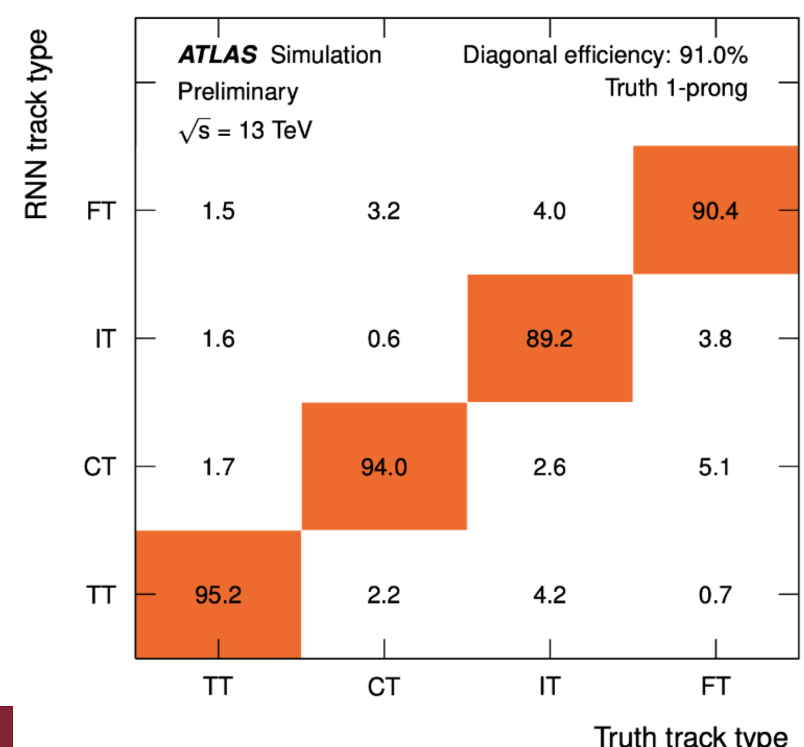
- Tau Tracks: (charged pions), to determine prongness;
- Isolation Tracks, originating from remnants of the hard scattering interaction;
- Conversion Track, from photon conversion in the detector;
- Fake Tracks, from pile-up or mis-reconstructed.

Tau Track Classification

Currently based on another RNN







Fast Track Preselector

Recent developments in b-tag group.

Last update on similar approach for Tau group is an algorithm proposed in 2017, where the fast tracking is executed in an RoI centred on the barycentre of the Tau candidate jet seed;

Proposal: dedicated low precision filter for flavour tagging using ML (much faster than tracking).

Geometric Deep Learning algorithms for tau lepton identification in the ATLAS experiment

SAPIENZA
UNIVERSITÀ DI ROMA

ATLAS
EXPERIMENT

R22 Variables

These are the variables used in this work, labeled as R22.

1p Variables	1p Variables	1p Variables
"TauJets/centFrac"	"TauTracks/pt_log"	"TauClusters/et_log"
"TauJets/etOverPtLeadTrk"	"TauTracks/pt_tau_log"	"TauClusters/pt_tau_log"
"TauJets/dRmax"	"TauTracks/dEta"	"TauClusters/dEta"
"TauJets/SumPtTrkFrac"	"TauTracks/dPhi"	"TauClusters/dPhi"
"TauJets/EMPOverTrkSysP"	"TauTracks/nInnermostPixelHits"	"TauClusters/SECOND_R"
"TauJets/ptRatioEflowApprox"	"TauTracks/nPixelHits"	"TauClusters/SECOND_LAMBDA"
"TauJets/mEflowApprox"	"TauTracks/nSCTHits"	"TauClusters/CENTER_LAMBDA"
“TauJets/pt_tau_log”	"TauTracks/z0sinthetaTJVA"	
Additional Variables for 3p case	"TauTracks/z0sinthetaSigTJVA”	
“TauJets/trFlightPathSig”	"TauTracks/d0TJVA"	
"TauJets/massTrkSys”	"TauTracks/d0SigTJVA"	

Feature Matrix

It is given by (N_nodes, N_tot_features) where:
N_tot_features = N_fts_tracks + N_fts_clusters

The `-2` is given by the fact that angular coordinates are shared features of both tracks and clusters, while all the others are assumed to be independent.

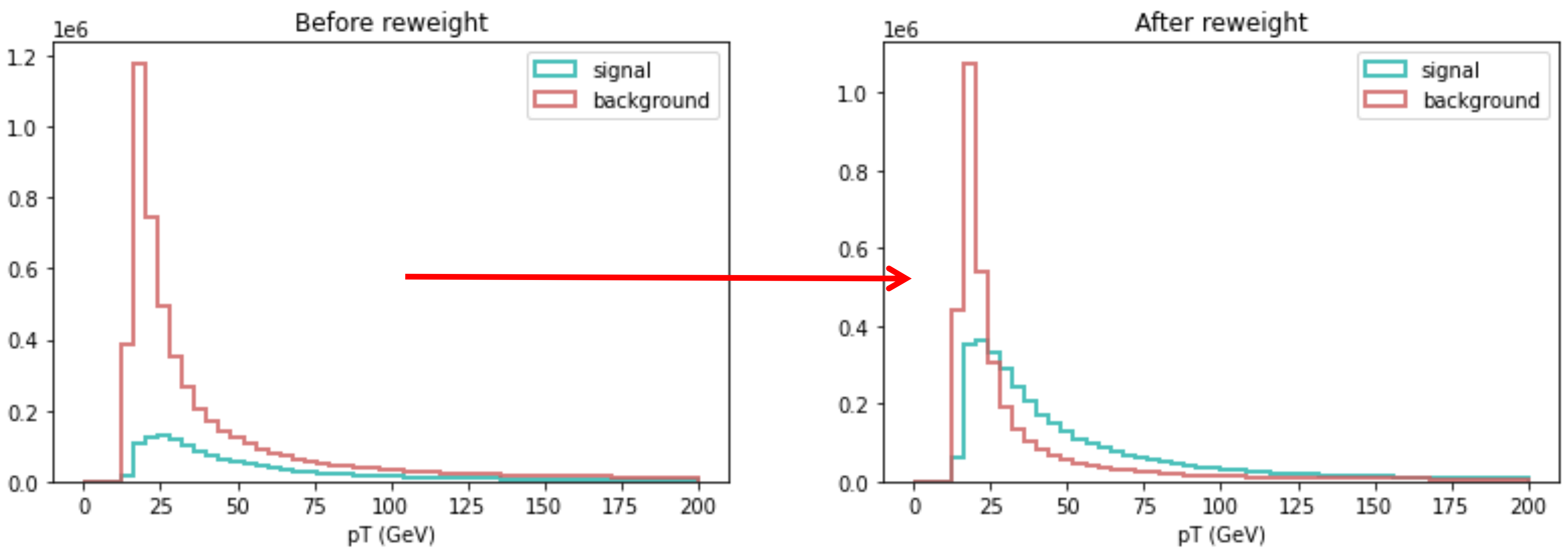
So a single row is like:
[η, ϕ, other_track_fts, other_cluster_fts]

Discussion

Why a model as the Transformer, which seems to overcome RNN performances in many fields, doesn‘t give similar results here?

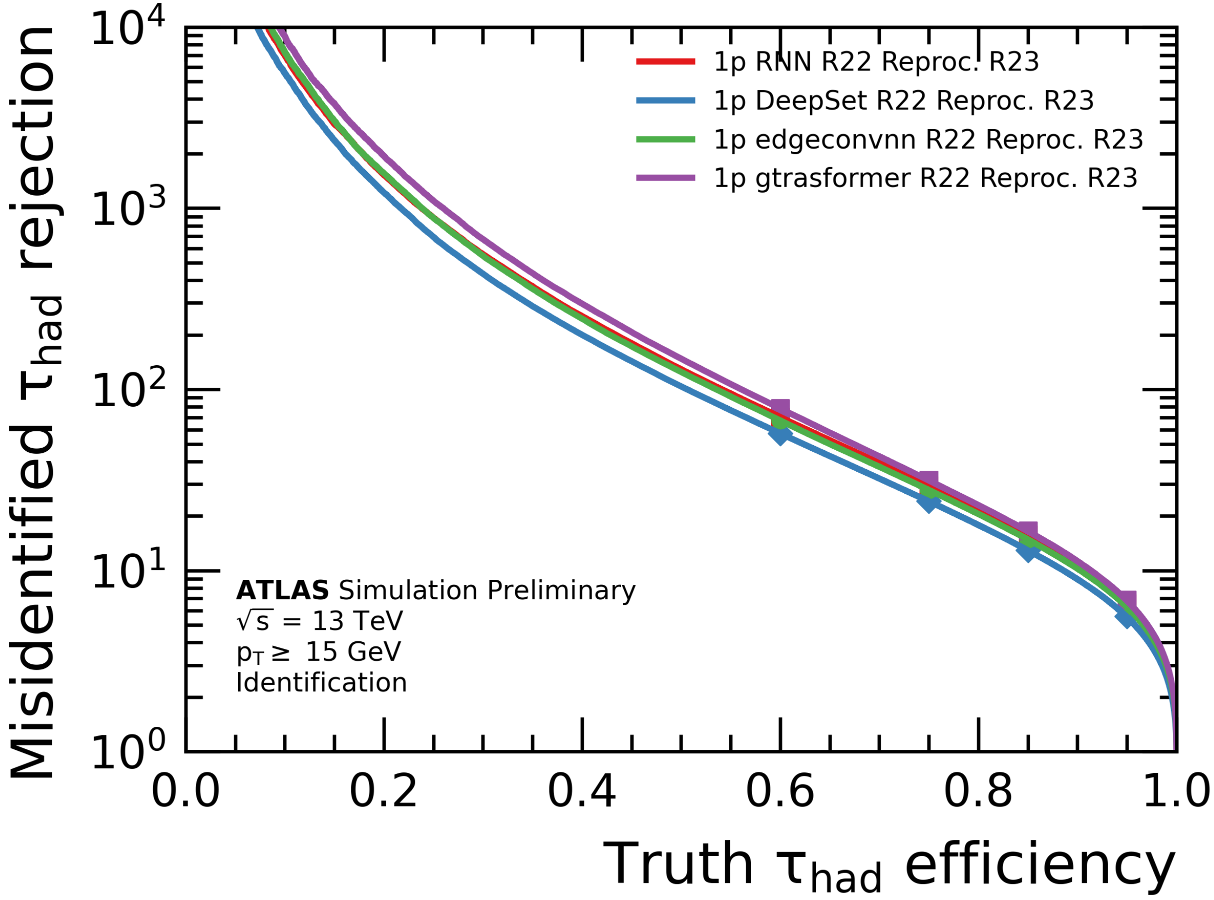
<u>A central role seems to be given by the pt reweight</u>.

I retrained these models with R22 variables using only class_weight



R22 Performance (no RW): ROC curve

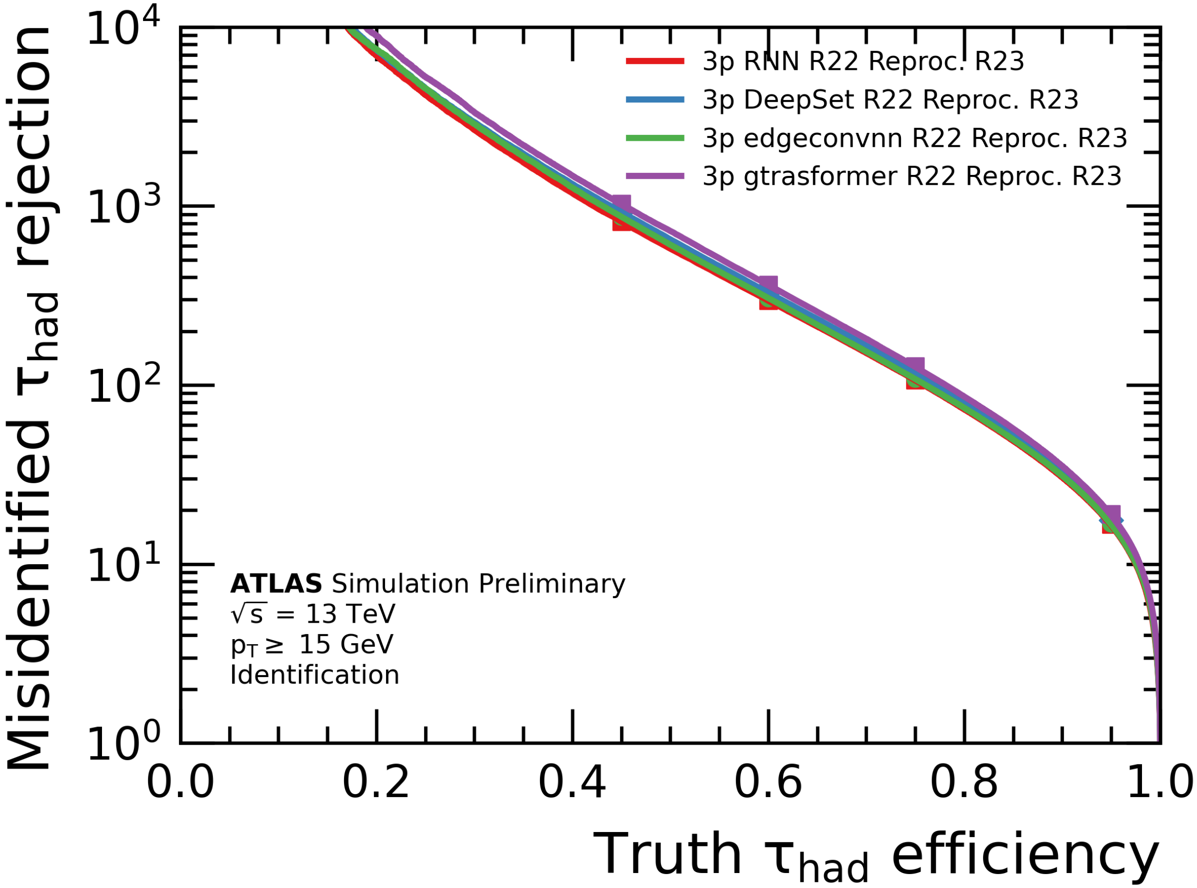
Best performances given by the Transformer, with RNN similar at high efficiency.



WP	0.6	0.75	0.85	0.95
RNN	70	29	16	6.6
DPS	57	24	13	5.6
E-C NN	67	28	15	6.4
Tran.	79	31	16	6.9

R22 Performance (no RW): ROC curve

Best rejection is given by Transformer.
But the rejections values are smaller respect the ones with the reweight.



WP	0.45	0.6	0.75	0.95
RNN	821	297	107	17
DPS	927	330	117	18
E-C NN	873	307	108	17
Tran.	1027	363	127	19