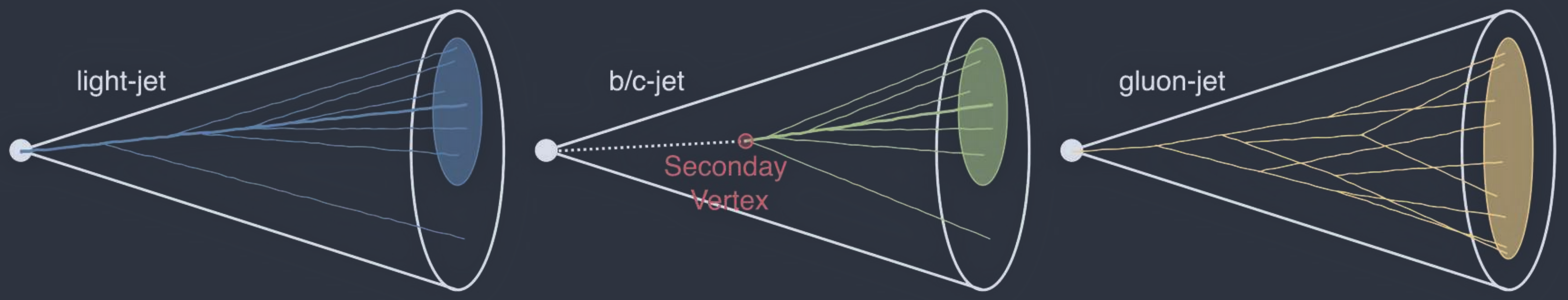
Flavor Tagging using Machine Learning

Masahiro Morinaga

The University of Tokyo
(ICEPP, Beyond AI)

What is Flavor Tagging?



Jet Flavor

- QCD jets are originated from qurak or gluon decay products.
- Tagging jet flavor is important technology to improve analysis sensitivity
- Quark and Gluon:
  - Gluon : A lot of decay products
  - Light flavor : u, d, s-quark
  - Heavy flavor : c, b-quark
- W/Z boson, Higgs, Top tagging :
  - Using large R-jet can identify jet if particle is boosted
  - Only higher p_{T}

Heavy Flavor Tagging

- Rely on secondary vertex from B-meson or D-meson decay
  - b-quark : $b \to B \to D \to K$
  - c-quark : $c \to D \to K$
- ML : BDT, CNN, RNN, DeepSets, Graph NN

Quark/Gluon Tagging

- Separation between light flavor and gluon
- Rely on #of track or similar variables
- ML : Graph or similar specialized model(ParticleNet, Lorentz Group NN)

Strategy

Problem

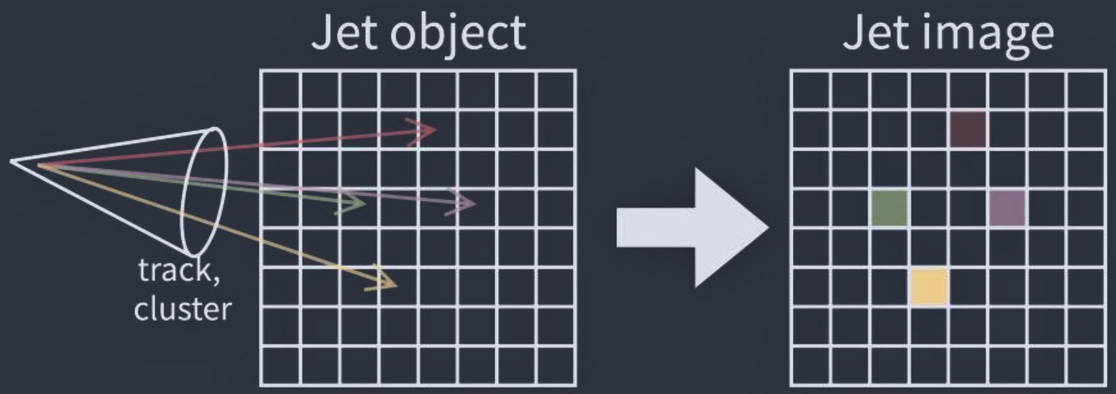
- Specialized model is too difficult to use it in experiment
- Jet kinematics dependency
  - Existing Q/G tagger has performance only at higher p_{T}
  - b-tagging rely on track property
- Q/G tagger and b-tagger are different algorithm.

Aim of this study

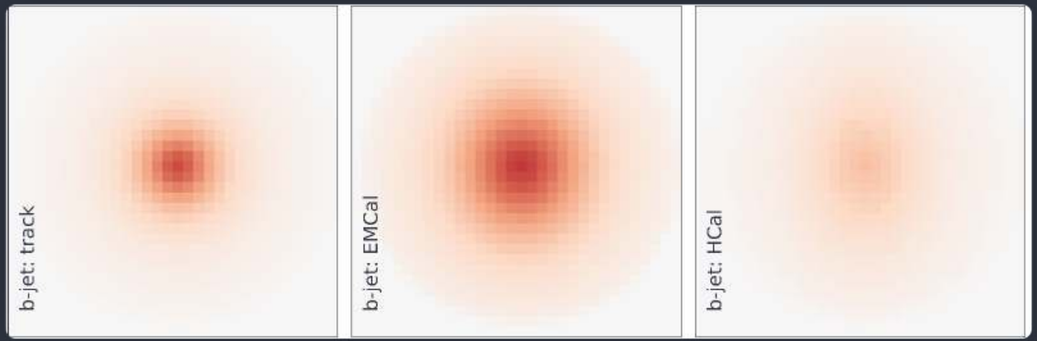
- Developing pratical neural network using model that is popular in ML community
- Less dependency for jet kinematic variables usuing FiLM module
- All-in-one tagger including b, c-tagging and Q/G-tagging
  - Classify light-flavor, gluon-jet, c-jet, b-jet simultaniously
  - Same model or method can be applied for large R-jet tagger(W, Z, t, H)
- Expect better performance at higher p_{T} by using calorimeter information
- Using extreamly high stat sample

Jet Image

Jet Image

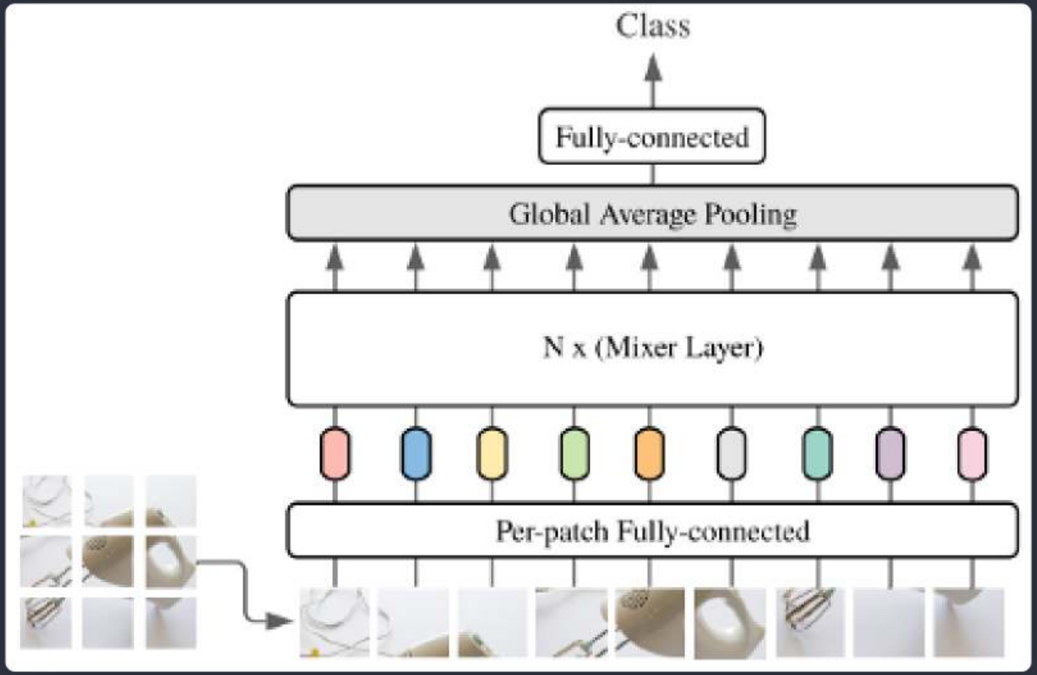


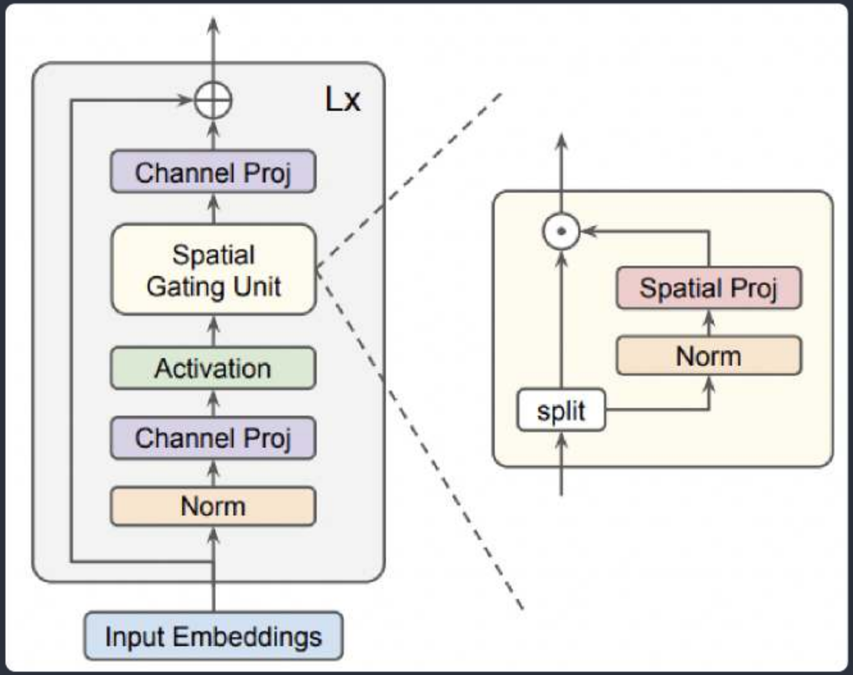
- Making jet image from constituents of jet (track, calo cluster),



- Jet image : tracks and calo clusters inside of jet ($\Delta R = 0.4$)
- Image size is 32 x 32 with five channels:
  - 1st : Count of track
  - 2nd : Count of EM calo cluster
  - 3rd : Count of Had calo cluster
  - 4th : p_{T} of 1-3 layers
  - 5th : sum of d_0 value of track
    - d_0 : transverse impact parameter of track

gMLP : Gated Multi Layer Perceptron

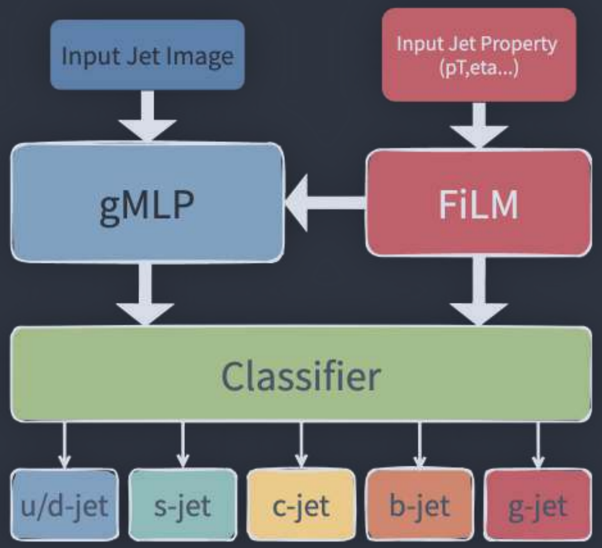




- Recently models which has no convulution are popular and have great performance
  - Vision Transformer(ViT), MLP-like(MLPMixer, gMLP, etc..)
- gMLP is used in this study
  - Similar performance with other models, but gMLP is faster than others.
- Making patch from one image, and pass it to FFC and Spatial Gatin Unit
- Spatial Gating Unit : Gating unit that learn spatial relation among cross-token

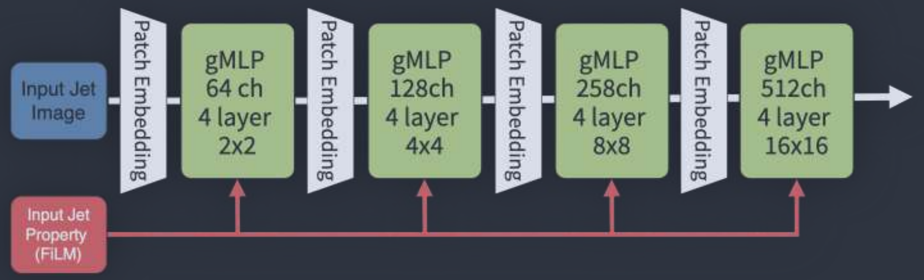
Model Architechture

Model



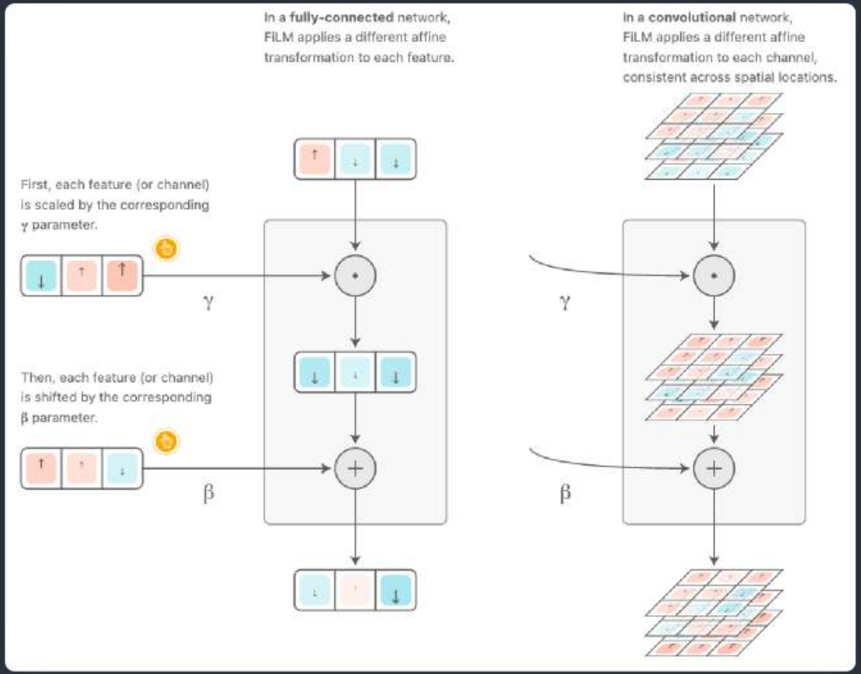
- Feature extractor : main component, which extract unique feature
- FiLM module : General conditional layer
- Classifier : simple MLP to classify all flavors

gMLP as MetaFormer



- Four gMLP Blocks with different patch size
  - Increase patch size 2x2 → 4x4 → 8x8 → 16x16
  - Increase inner feature dimention 64 → 128 → 256 → 512

FiLM : Feature Wise Linear Modulation

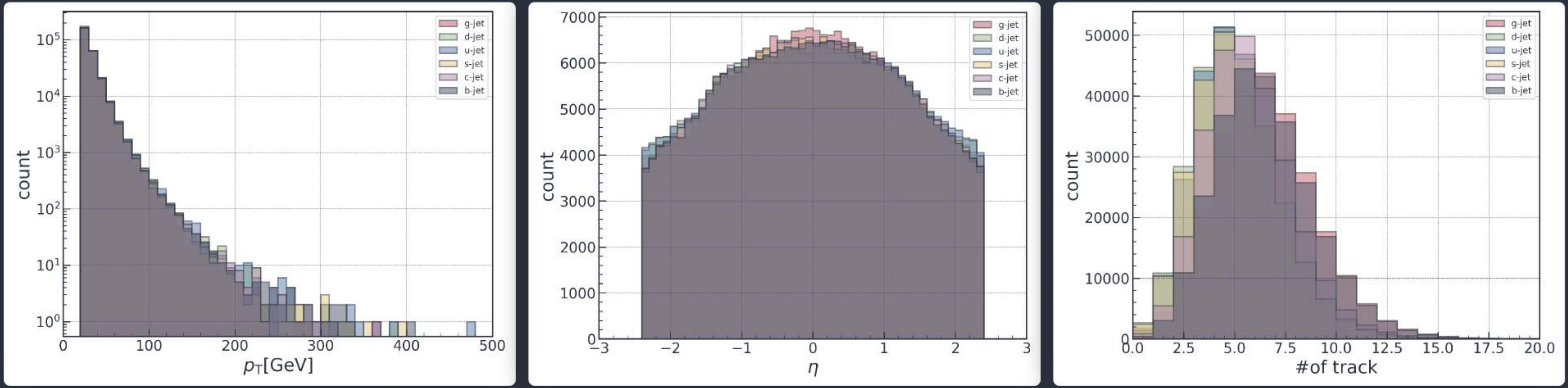


- image from here

Decorrelation using FiLM

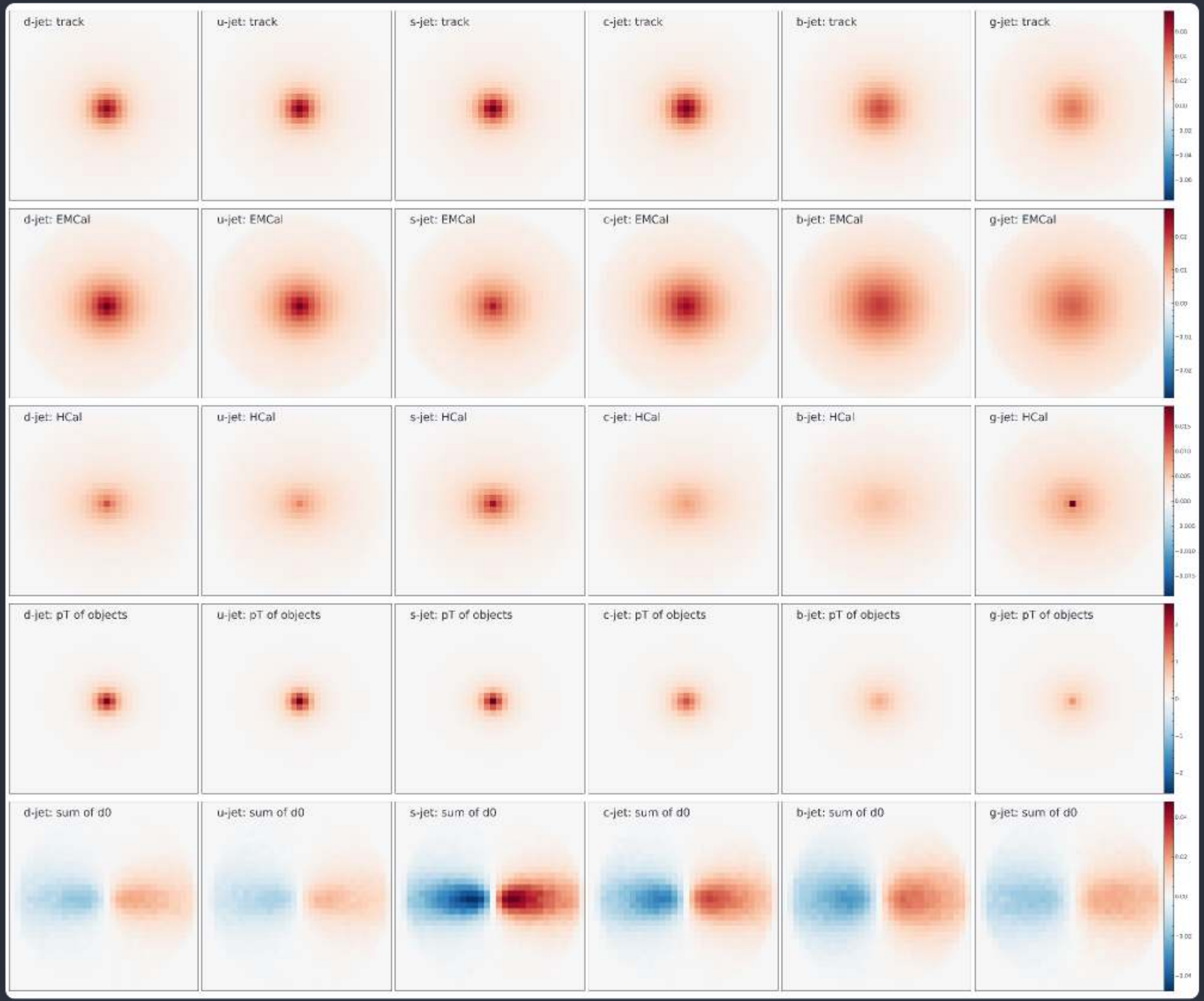
- In order to remove/reduce correlation between output score and jet kinematics(jet property, p_{T}, η, #of track...), FiLM layer is utilized.
- FiLM is General conditional layer that can
- FiLM($F|\gamma, \beta$) = $\gamma * F + \beta$
  - F : output of layer, e.g. convolution, linear...
  - γ, β : affin parameters of FiLM layer, (γ, β は learnable parameters)
- Performance of q/g-tagging, b/c-tagging depends on p_T or #of track
- Removing this correlation is worth to try
  - Just one training for different p_{T} or η region.
  - Expect imporvement of data and background comparison.
- Classification performance might be reduced $\rightarrow$ trade off between

Training Samples



- Delphes are used in this study with ATLAS geometry
  - MG5_aMC of v3.2.0 w/ Pythia8
  - Delphes with ATLAS no pileup card using pflow based jets
- Generate $pp \to gg, u\bar{u}, u\bar{u}, d\bar{d}, s\bar{s}, c\bar{c}, b\bar{b}$
  - Selection : $p_{\mathrm{T}} > 20$ GeV and $|\eta| < 2.4$
  - Truth label : labeled as production mode

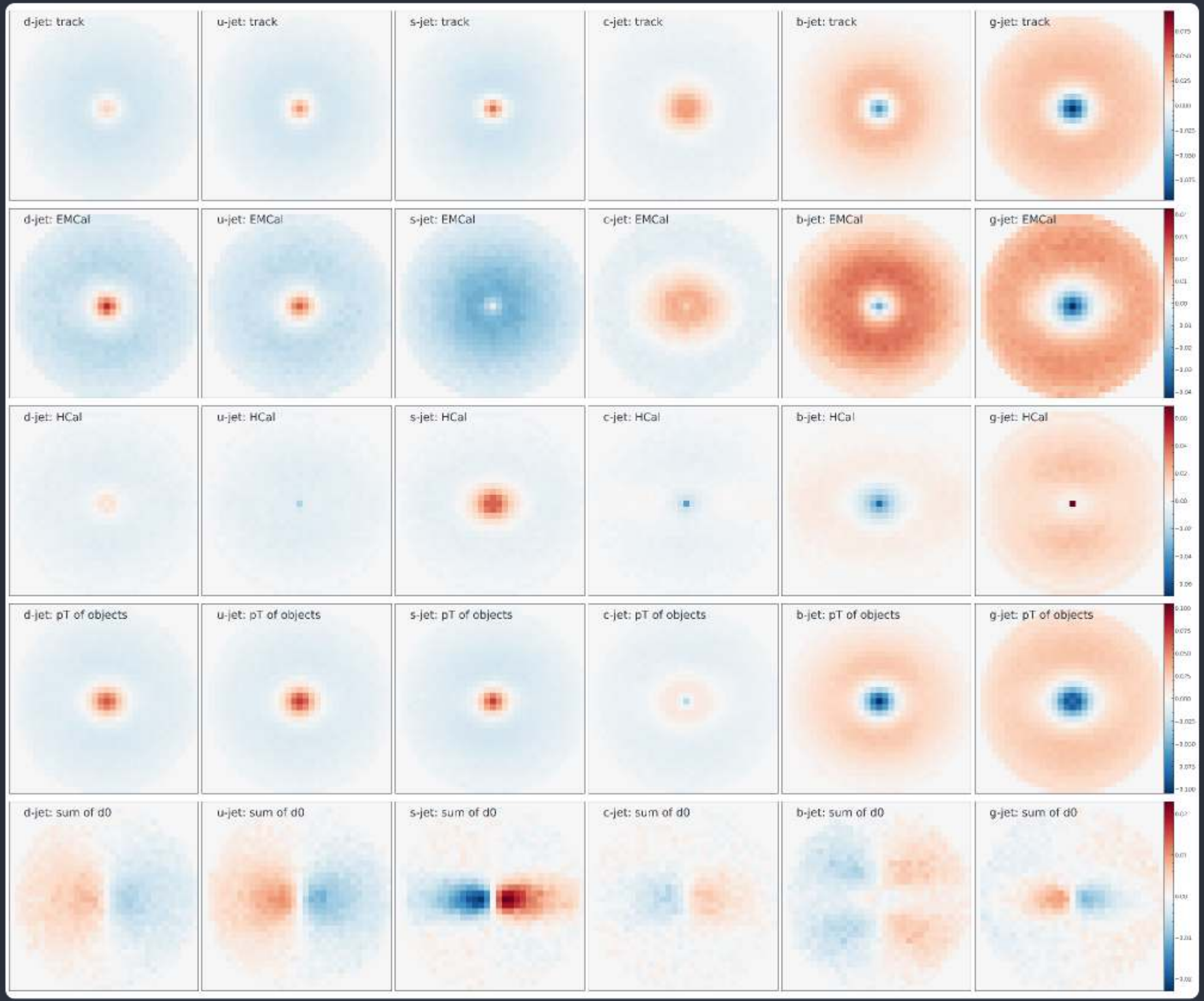
Image Preprocessing



Preprocessing

- Similar preprocessing is applied as this paper
  - Without rotation, just normalize pixels
- Pixel normalization : $I'_{i,j,k}(s) \to (I_{i,j,k}(s) - \mu_{i,j,k})/\sigma_{i,j,k}$
  - μ, σ is mean, standard deviation of each pixel, which is sum over all samples.

Image Preprocessing



Preprocessing

- Same preprocessing is applied
- Pixel normalization : $I'_{i,j,k}(s) \to (I_{i,j,k}(s) - \mu_{i,j,k})/\sigma_{i,j,k}$
  - μ, σ is mean, standard deviation of each pixel, which is sum over all samples.

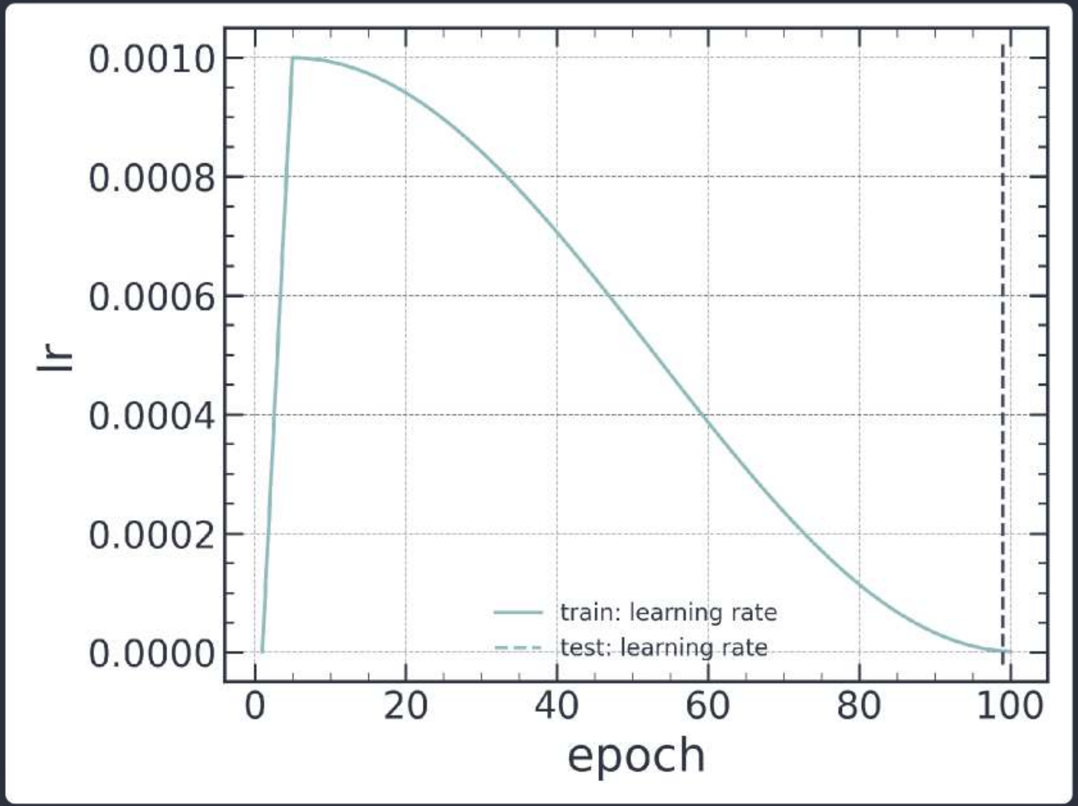
s-jet can be classified?

- s-jet looks not similar u/d-jet $\to$ due to effect of K_S^0 decay
  - K_S^0 is long-lived (~few cm) $\to$ d_0 would be relatively large
  - $K_S^0 \to \pi^\pm \pi^\mp$: More energy depostion at hadron calorimeter.
- If s-jet tagging can be performed then:
  - Measurement of $W \to cs$ decay
  - W/Z-tagging using resolved two jets ($W \to cs/ud$ or $Z \to qq$)
  - $H^+ \to c\bar{s}$?

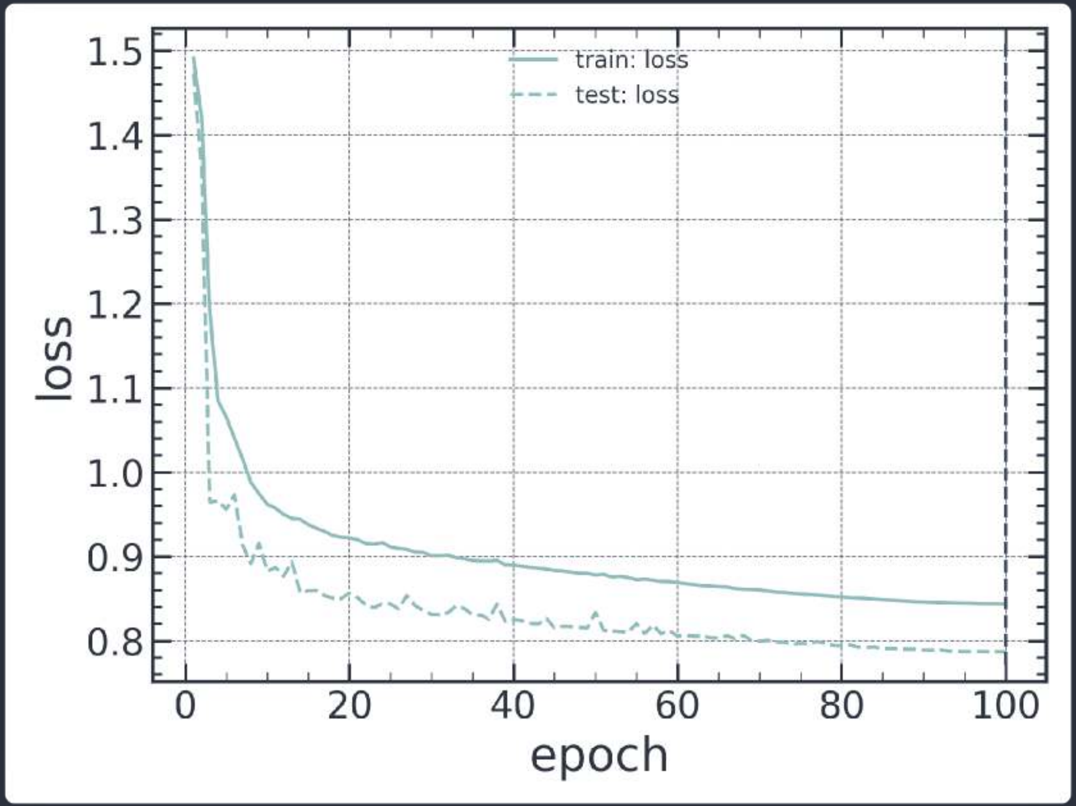
Training Setup

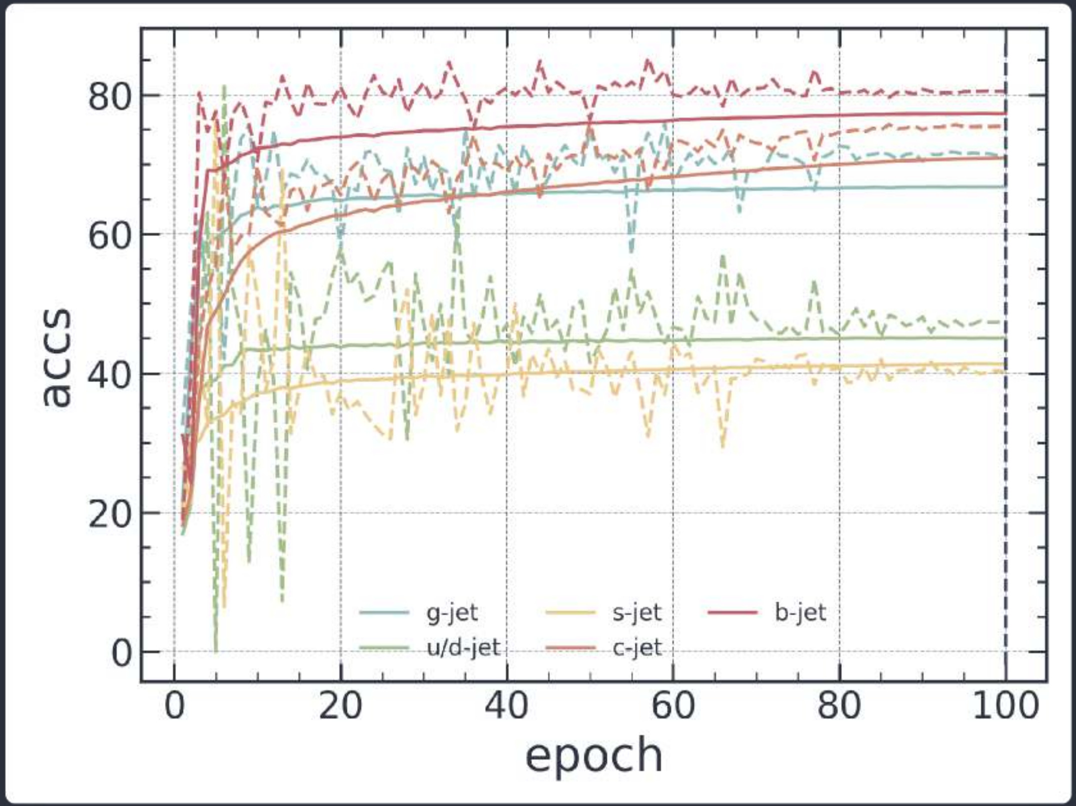
Training setup

- Optimizer : `RAdam`
    - L2 norm : wieght decay = 1.0e-3
- Learning rate : CosineAnnealingWarmupRestarts
    - lr : `1.0e-03` to `1.0e-06` ,
    - wramup : 5 epoch
- Epoch : 100 epoch with early stopping
    - Early stopping : patience = 20
- Loss : Class balanced CE loss, without this correction
    - β : balanced parameter is `0.9999`
- Batch size : default is `2048` .
- Training sample: `2**22 = 4,194,304` sample per jet flavor
- Validation(test) sample : 10% of training samples
- NVidia A100 GPU x 8 parallel distributed training
- Pytorch is used



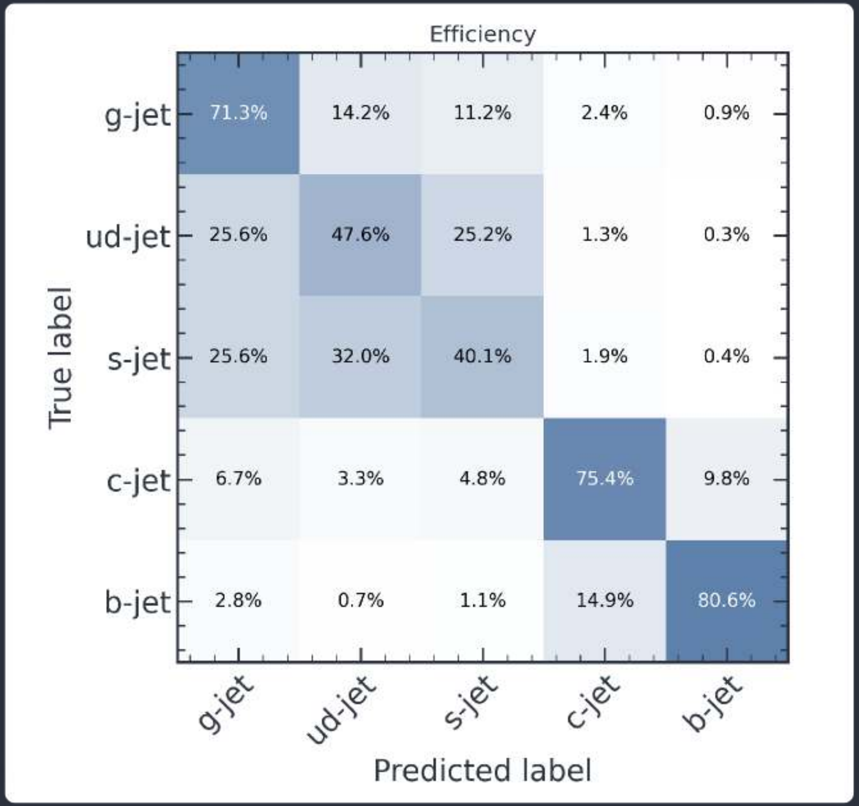
Results

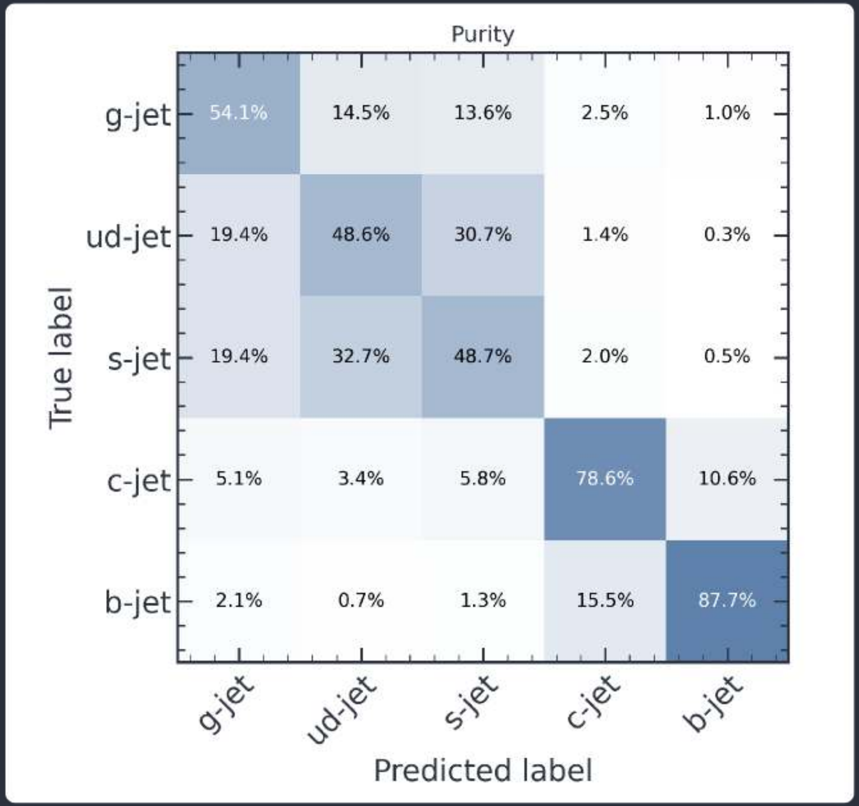




- Left : Loss curve as a function of training epochs
- Right : Accuracies for all flavors as a function of training epochs
- Training looks helthy, no overfitting.
- c-jet accuracy is imporving a lot against training epochs.

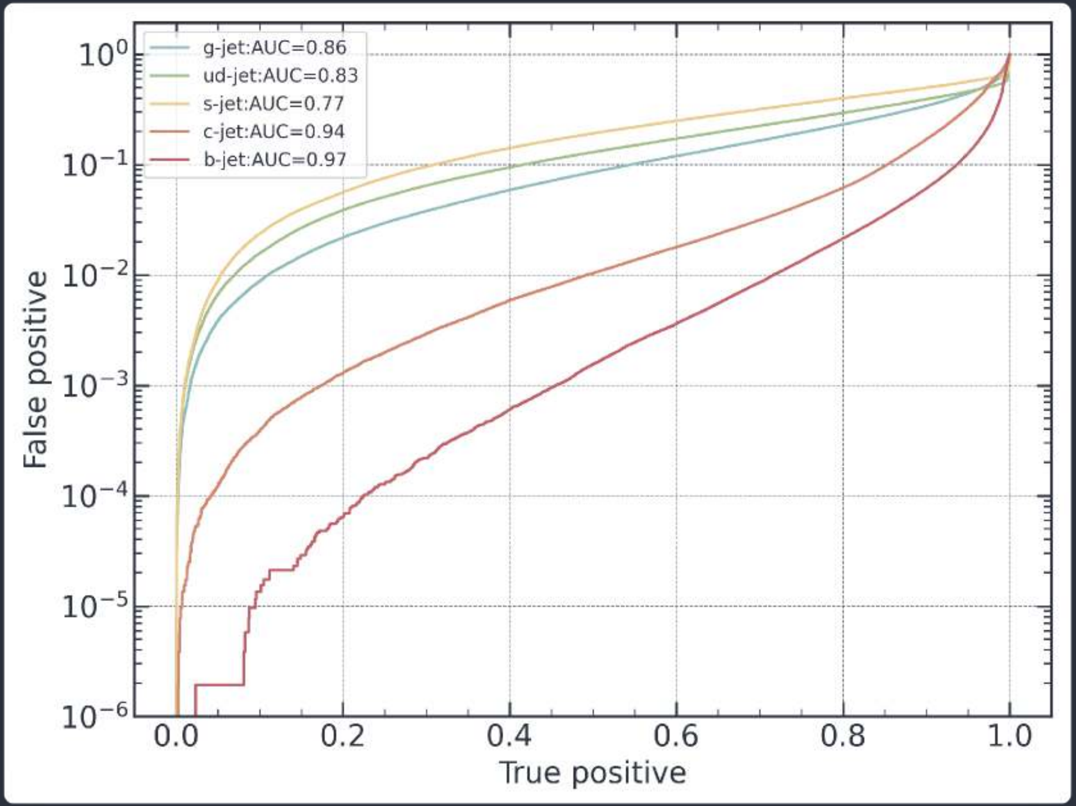
Performance

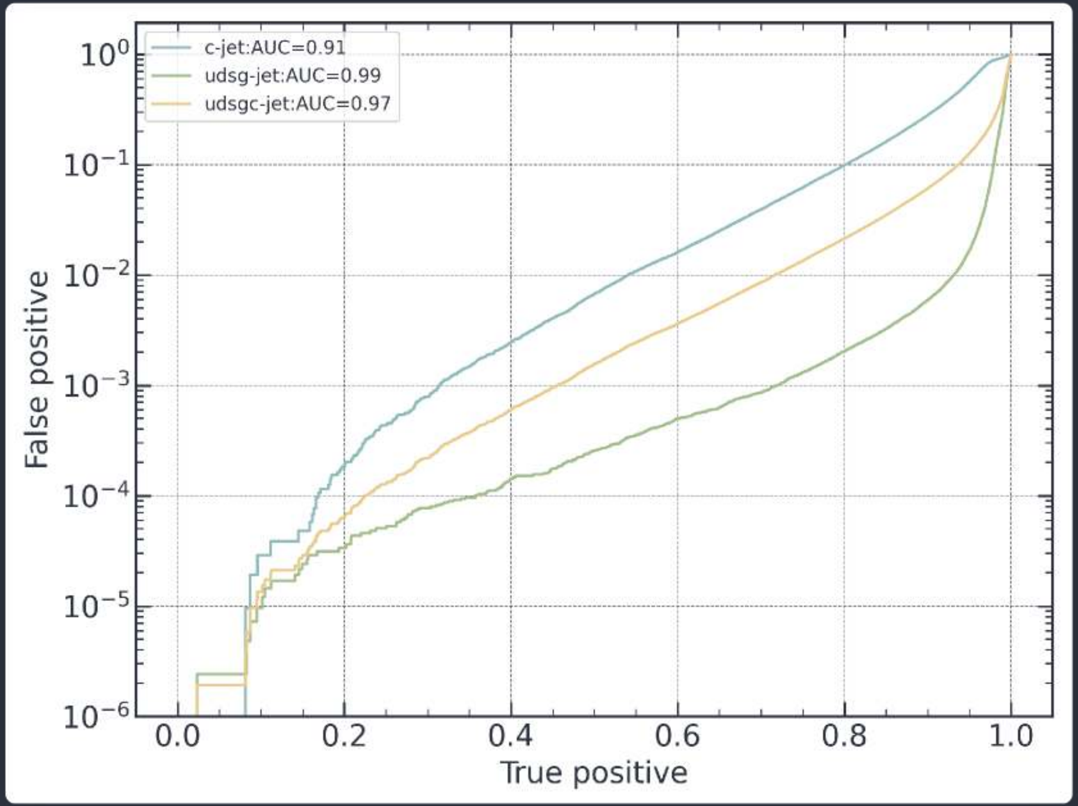




- Confusion matrix:
  - Efficiency : Produced(truth) jet is classified as prediceted label
  - Purity : prurity of predicted label
- b, c-jet has good separation against all other flavors,
- s-jet and u/d-jet are not so separated, but it is worth to try and there are still room for improvements that haven't been tried

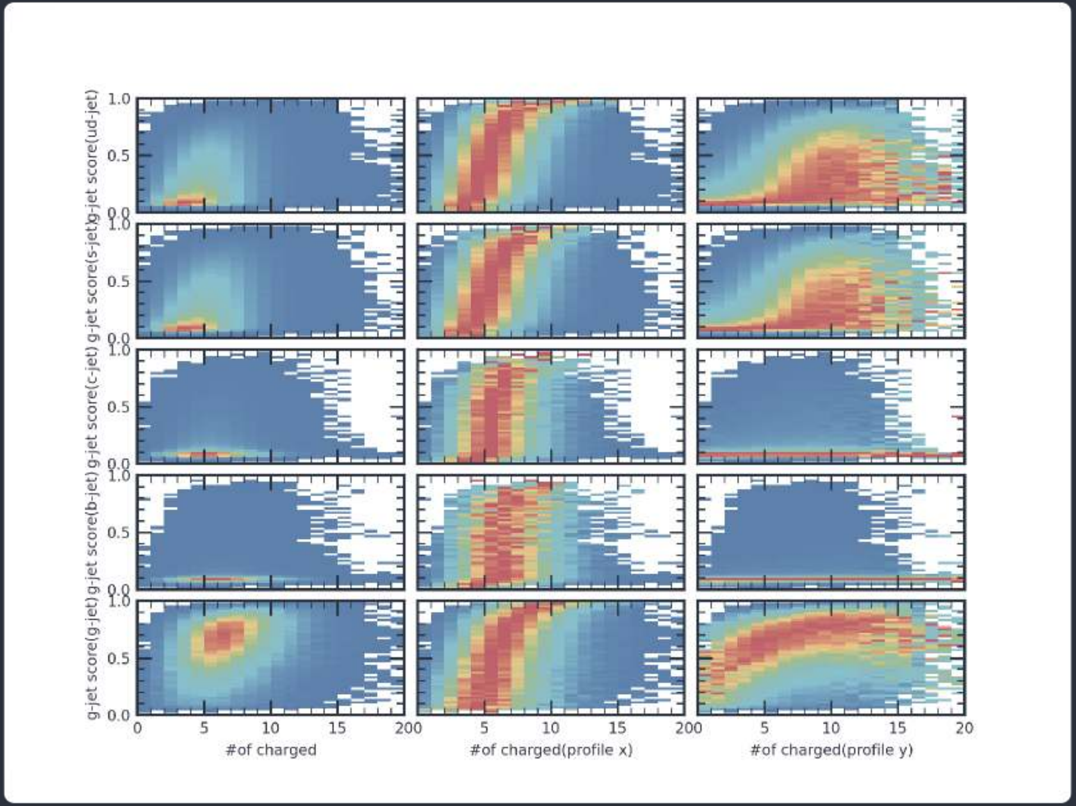
Performance

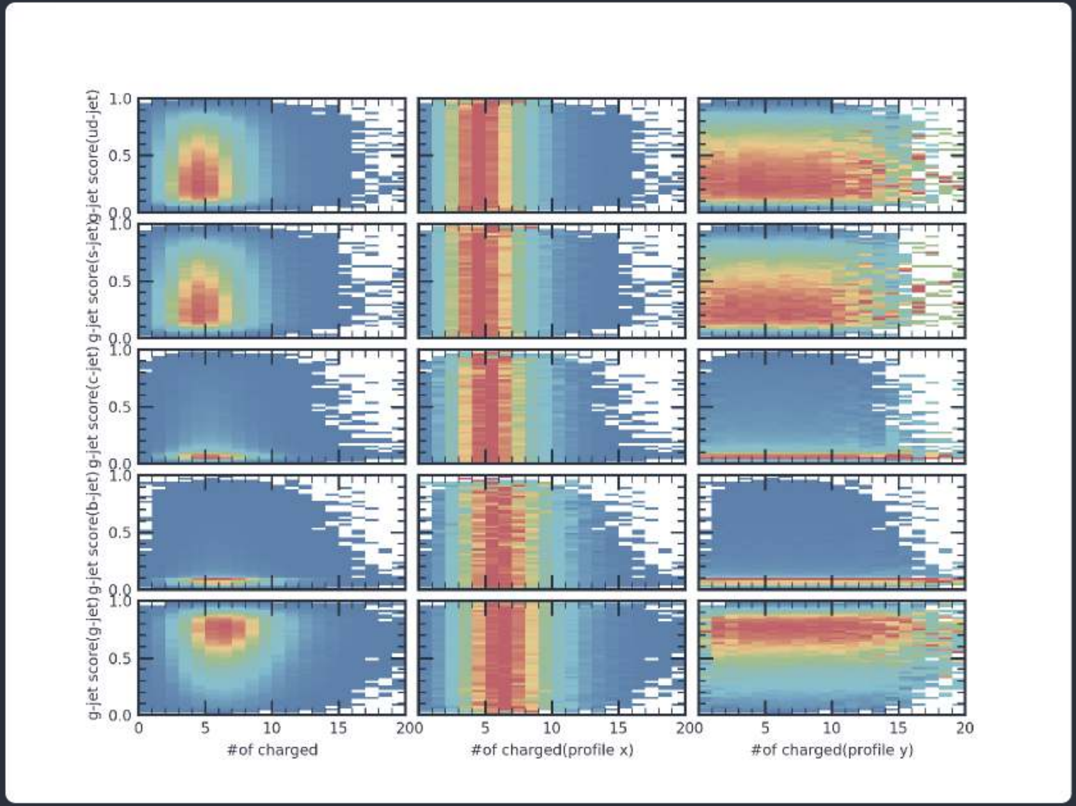




- Left : ROC curve, each flavor against others
- Right : ROC cureve of b-jet against c-jet and udsg-jet, others.

Effect of FiLM Layer





- 2D histogram : x-axis: #of track, y-axis : g-jet score (gluon has strong dependency against the number of tracks)
- Each panel :
  - from top to bottom : ud,s,c,b,g-jet
  - from left to right : nominal 2D, 2D with profile of x-axis, 2D profile of y-axis
- Left : Without FiLM, Right : With FiLM $\rightarrow$ can see clear imporvement

Conclusion and Feature Plan

Conclusion

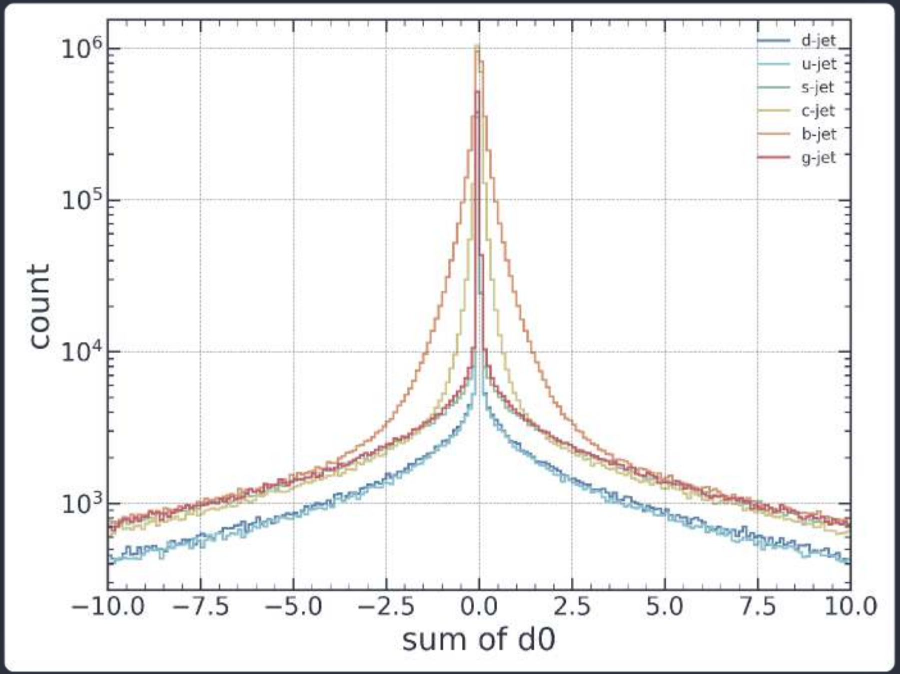
- Flavor tagging with the jet-image was introduced and shown
- Model without convolution(CNN) works well and its result looks pretty promising
- FiLM is worth to utilize in order to decorrelate output score and jet kinematics.
-

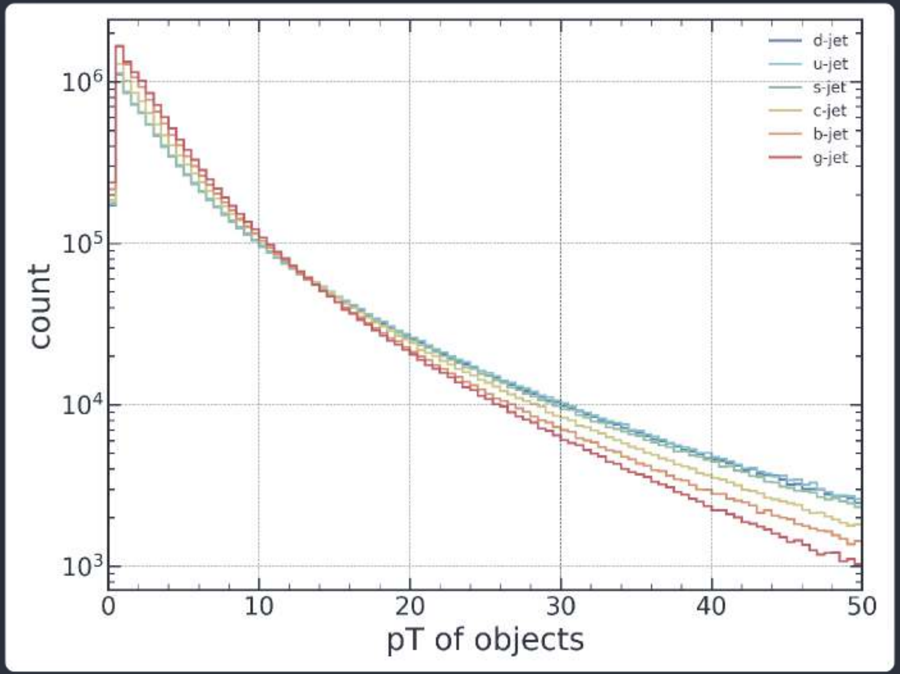
Plans

- Large-R jet tagging : Top, Higgs, W, Z boson tagging
- W, Z-tagging with resolved two jets
- Small-R jet tagging : tau hadronic decay

Backup

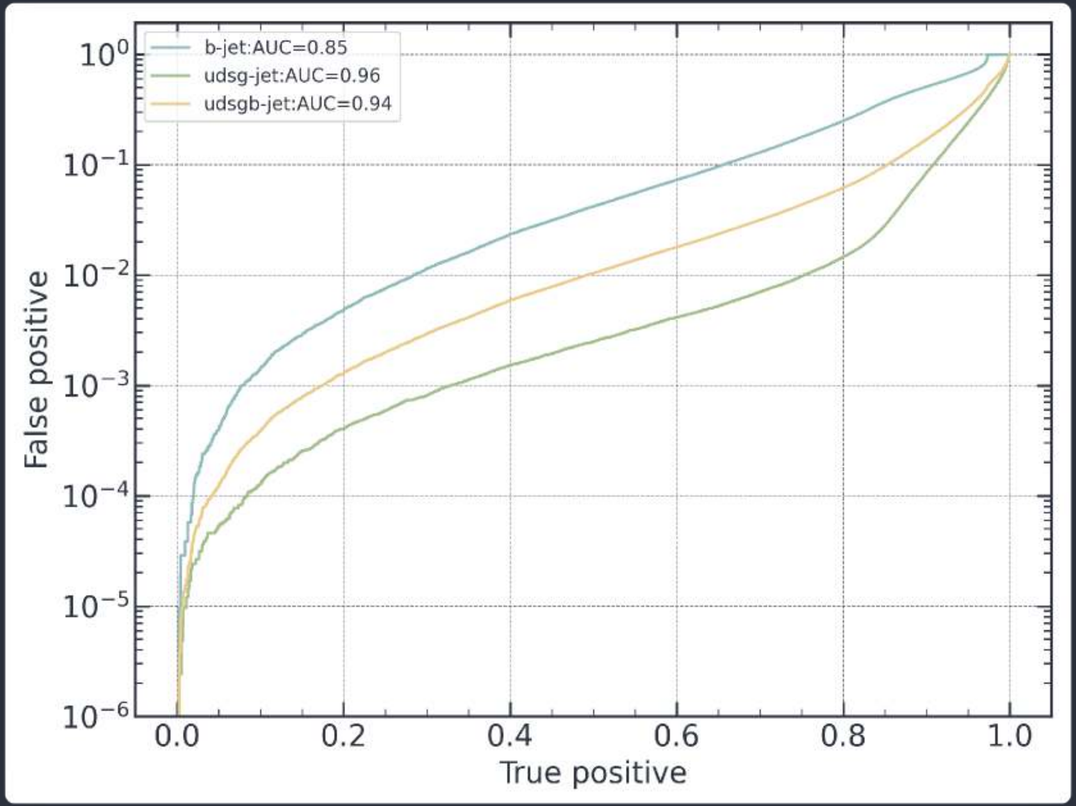
Samples : Constituents

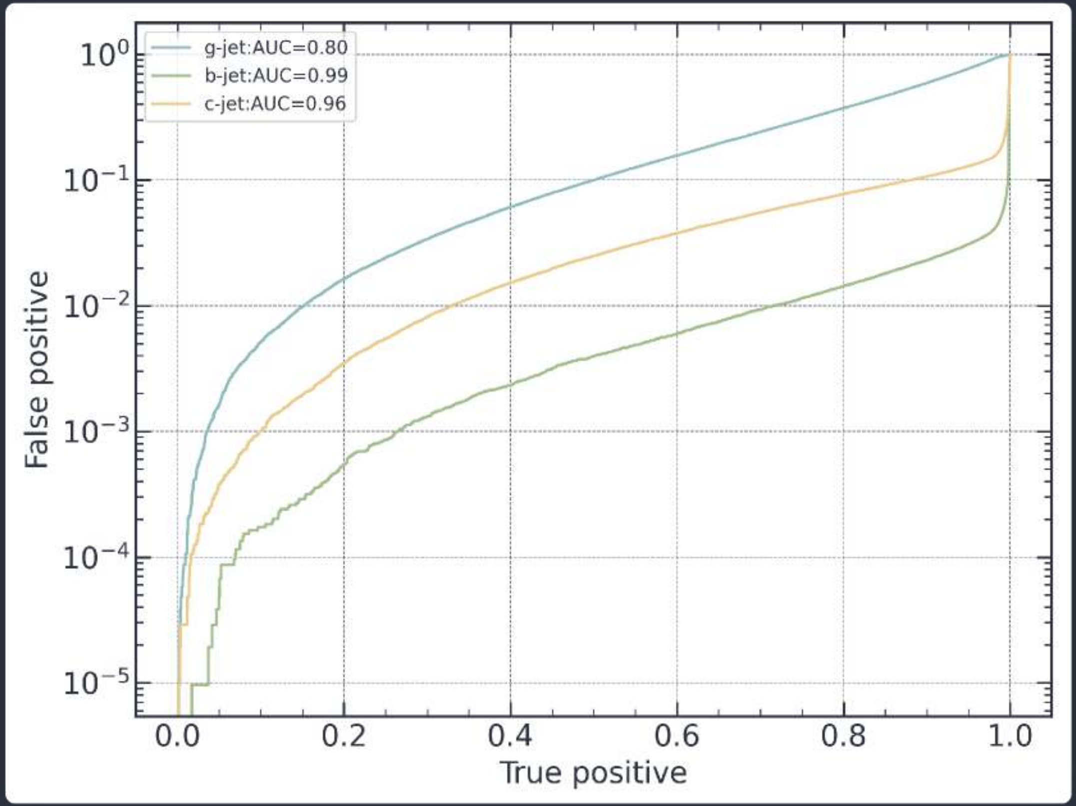




- d_0 distribution for constituents of all flavors:
  - Indeed, b-jet $>$ c-jet $>$ s, g-jet $>$ u, d-jet
- p_{T} distribution for constituents of all flavors:

Performance





- Left : ROC curve, each flavor against others
- Right : ROC cureve of b-jet against c-jet and udsg-jet, others.