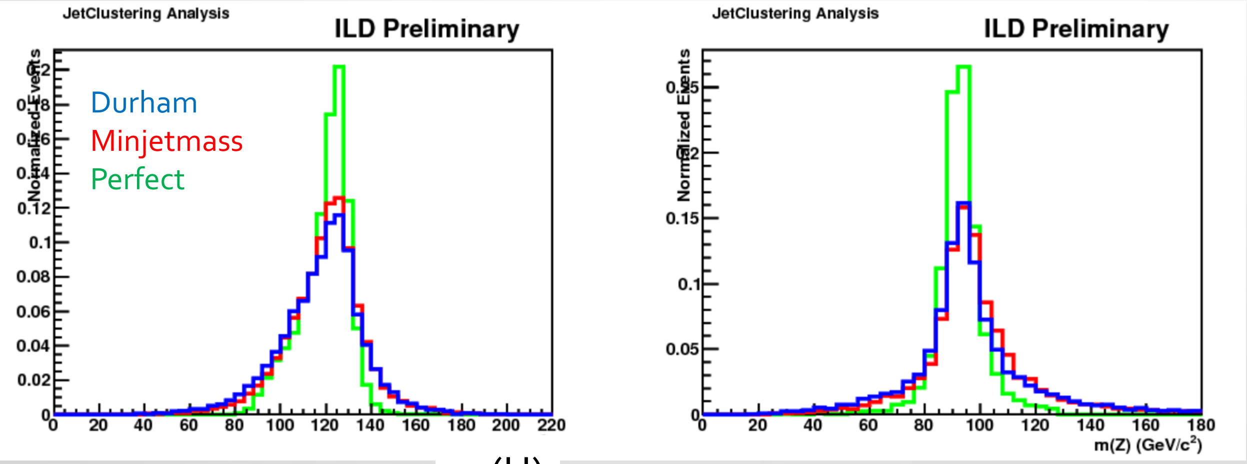
Automatic Colorization for Jet Clustering

Masakazu Kurata

04/14/2018

Introduction

- Jet clustering is one of the main key to obtain better physics results
  - Physics results are strongly limited by mis-clustering
  - To obtain correct jets leads to improve the mass resolution of the resonances
- Present jet clustering is far from good tool for reconstructing jets
  - e.g. Higgs self-coupling: ～40% improvement if perfect!

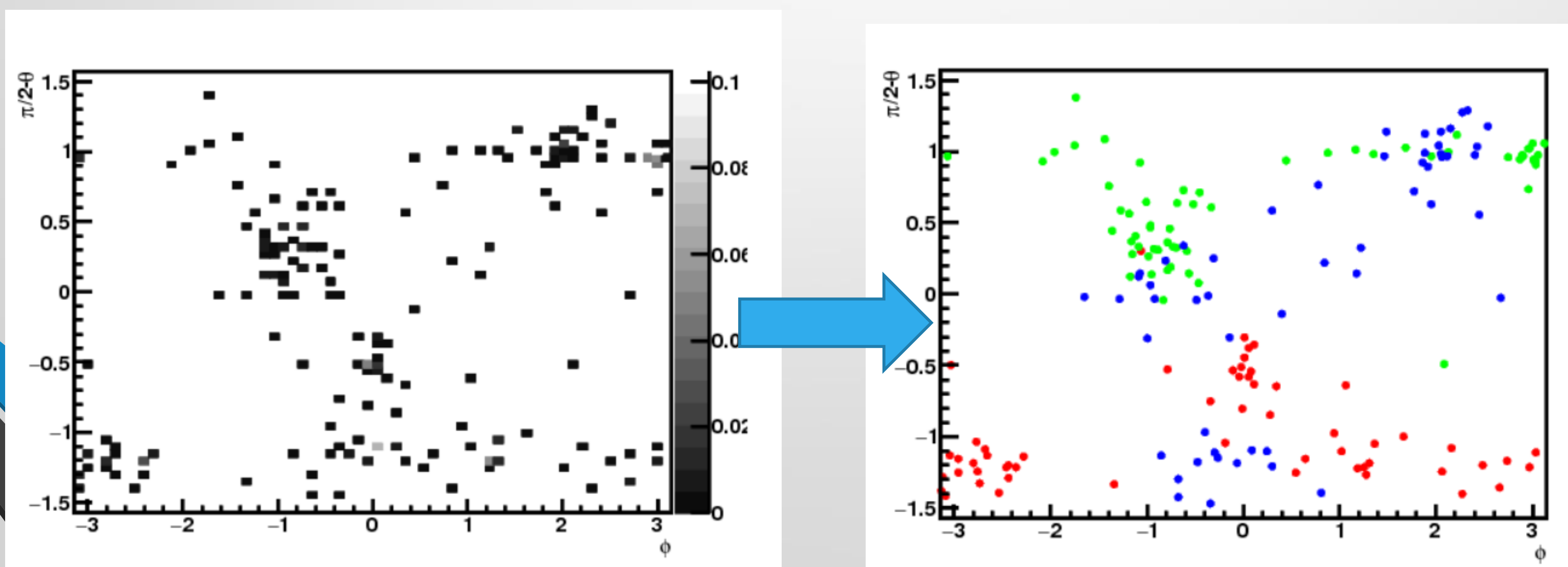


m(H)

- Staging: even at 250GeV, clustering is very important
  - Separation of ZH/ZZ/WW in hadronic events

Make the most of CNN

- Tried supervised learning(Feedforward neural network)
- One of the problem is how we can absorb the difference between events
  - Needs very high track(minijet) assignment efficiency to improve mass resolution
  - For very high efficiency, from NN view, all the events look "exception"…→infinite number of nodes & infinite number of events is necessary?
- CNN can relatively absorb position shift & distortion of shape
  - So, CNN meets this?
- One idea: "Automatic Colorization" using CNN
  - Gray scale → Color

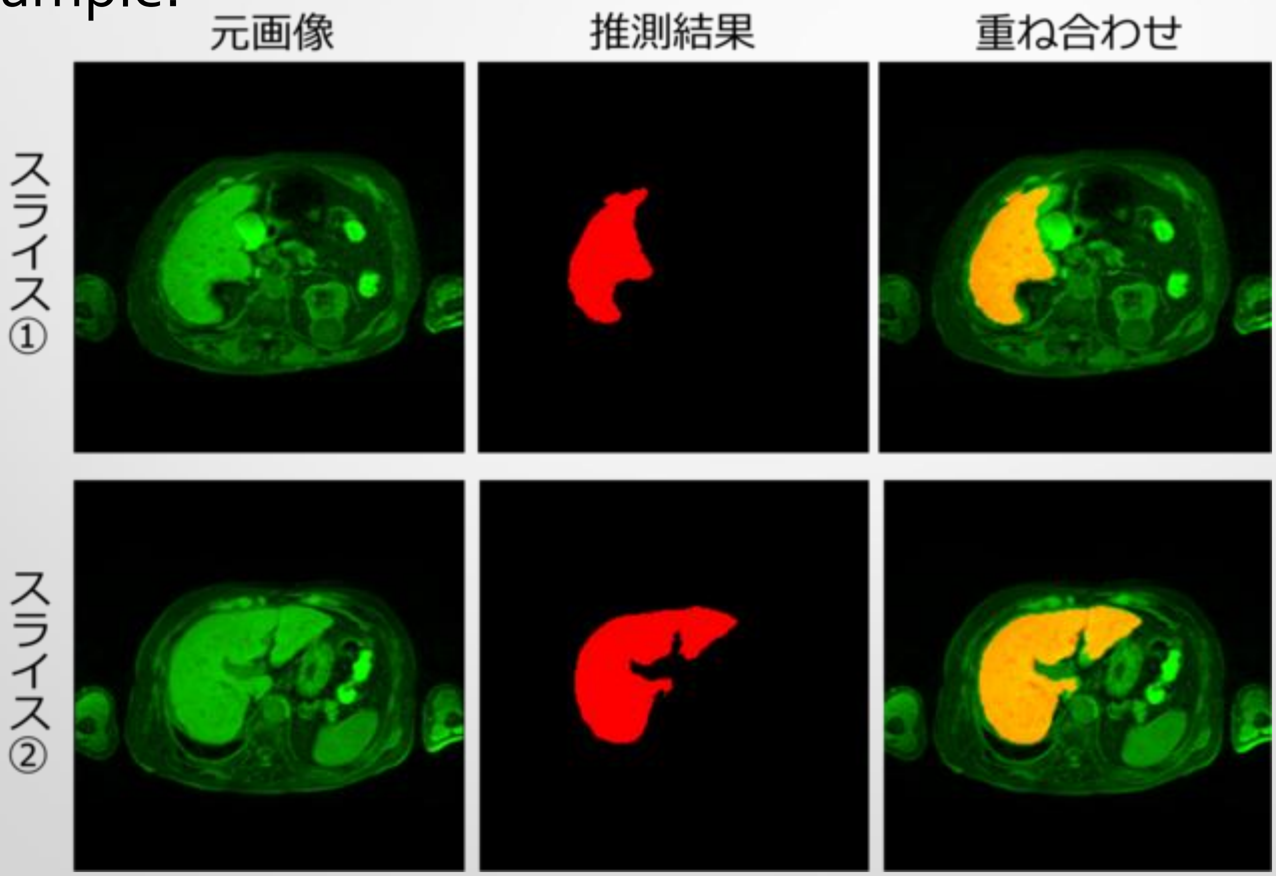


Example

Grey scale Expectation Truth

or

- We can estimate the region of what we want to know
- This calls “semantic segmentation”
- Example:



- Can we apply these kinds of techniques for jet clustering?

Use CNN for automatic colorization

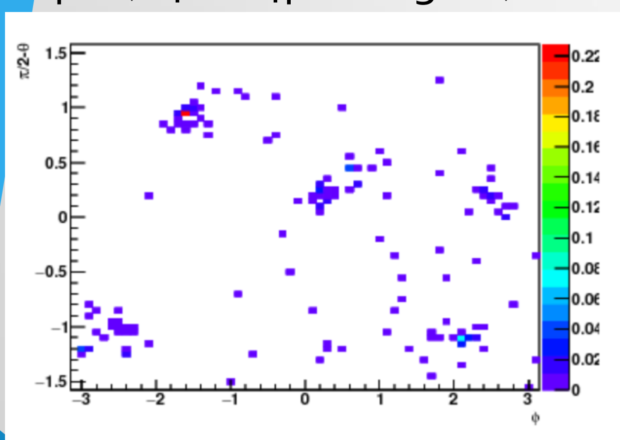
- For jet clustering, we need the global and local information for each event
  - Global: Where is the large energy located?
  - Local: Correlation between neighbors or large energy area?
- Using CNN, we will extract both of the features
  - Encorder-Decorder type CNN is used (calls as u-network)
  - Already ～30 layers in CNN!
- Add Conditional Random Fields for improvement
  - 1-2% improvement can be seen in semantic segmentation
  - How about this?

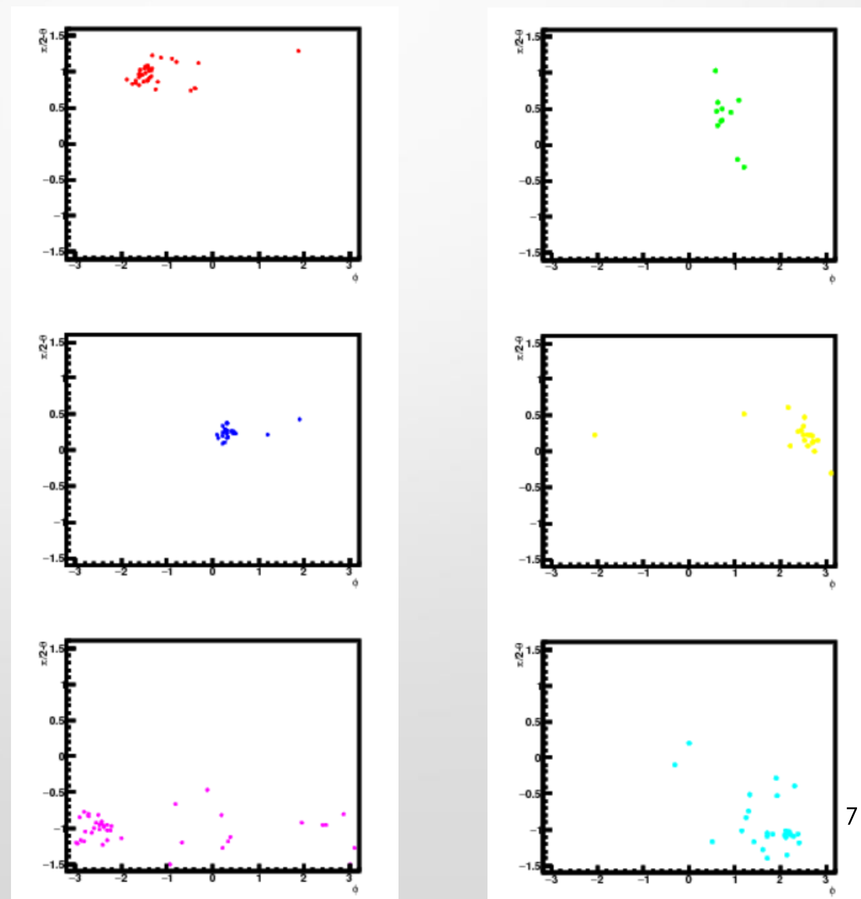
trial

- Using energy map of each event, estimate color of each track
  - ZHH→(qq)(bb)(bb)→6jets
  - Do not consider color-singlet state

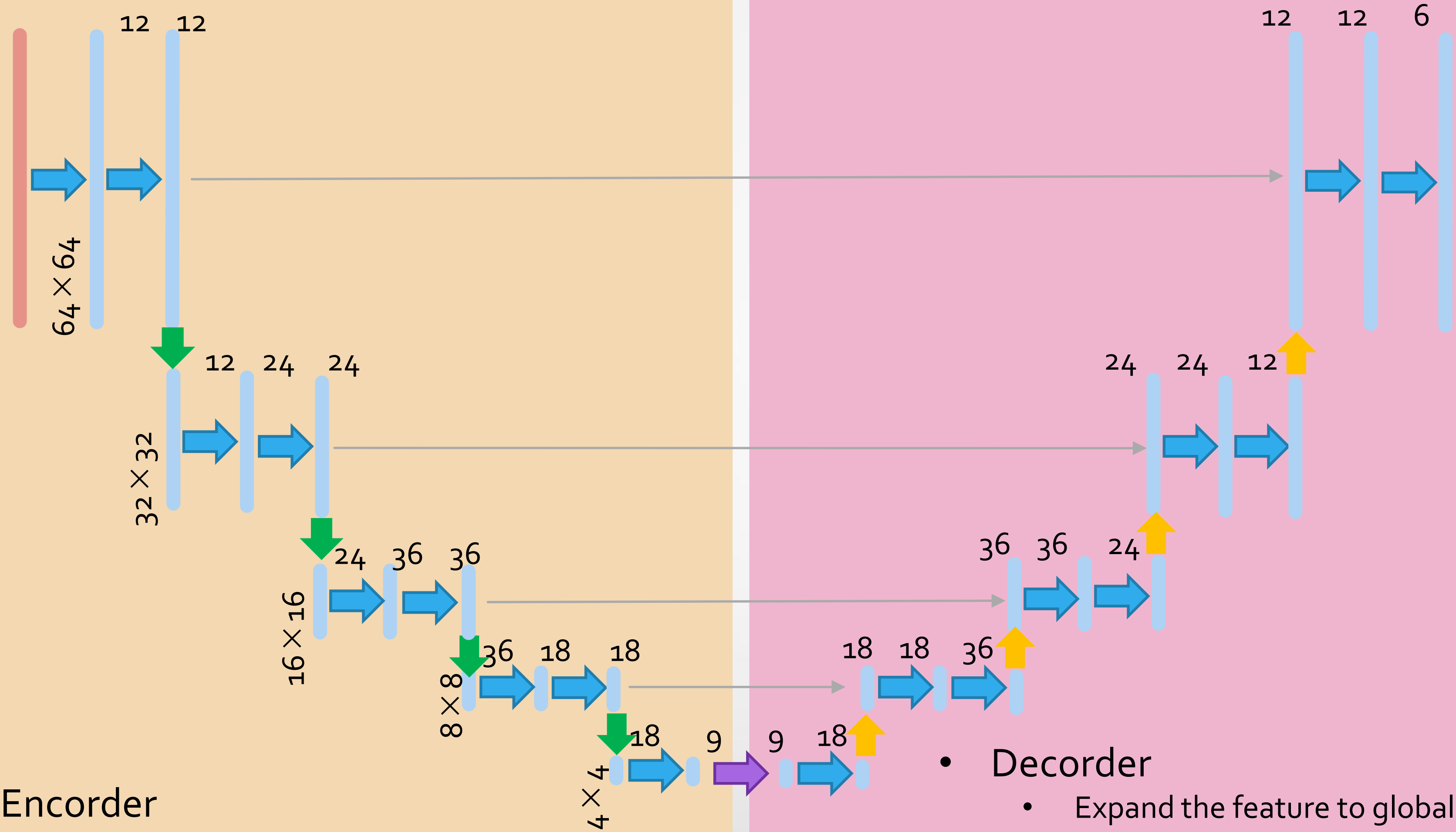
Output(64×64pixel figure)

Input(64×64pixel figure)





U-network

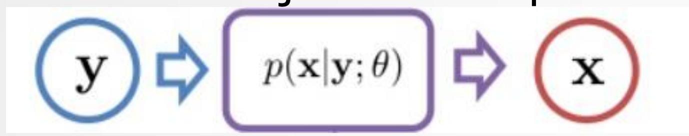


- Encorder
  - Extract local & higher order feature
  - Down sample to make network robust for distortion & shift effect
  - Lost position information

- Decorder
  - Expand the feature to global
  - Up sample to recover position information
  - Merge encoder nodes to get precise position information

Conditional Random Fields

- Inference of a latent variable from measured variables
  - x: measured variables → energy, momentum, charge, etc.
  - y: latent variable → the jet which a particle is coming from



  - We have to estimate good conditional probability distribution

- So, use Conditional Random Fields:
  - Estimate parameters to maximize the Boltzmann probability:

$$p(x|y;\theta) = \frac{1}{Z}\exp(-E(x))$$

$$E(\mathbf{x}) = \sum_{(i,j)\in\mathcal{E}} f_{ij}(x_i, x_j) + \sum_{i=1}^{N} f_i(x_i)$$

- Same structure as Ising model:

$$H(\sigma) = -J \sum_{i=1,...,L} \sigma_i \sigma_{i+1} - h \sum_i \sigma_i$$

- Optimization procedure is very similar to Ising model
  - Mean field approximation, Gibbs sampling, etc.

Preliminary architecture

- CRF is used for post-processing of CNN

$$E(\mathbf{x}) = \sum_{(i,j)\in\mathcal{E}} f_{ij}(x_i, x_j) + \sum_{i=1}^{N} f_i(x_i)$$

Constraint for pairwise tracks

Output of u-network for each track

- In first term, we will be able to impose physics constraints
  - Now, simplest case: impose Durham distance measure:

$$f_{ij} = \omega \cdot \exp\left(-\alpha \frac{2Min\left(E_i^2, E_j^2\right)}{Evis^2}(1 - cos\theta)\right)$$

  - Based on the fact that jet products will fly colinearly
- We can impose any physics constraint
  - I don't know what is good...
  - Vertex constraint?
  - Other distance measure(anti-kt?)
  - Something else?

Over fit check

- This is still test stage, so cannot check overfitting well
- Just estimate using loss function(small is better):

$$L = -\frac{1}{N}\sum_{jet}\sum_{track}\frac{E_{track}}{E_{jet}} Log y_{track}$$

- If no overfitting, L is almost same between test and train

Without CRF Num. of training events	Loss Train	Loss Test
140	0.185	1.78
4000	0.464	0.725
9000	0.571	0.654

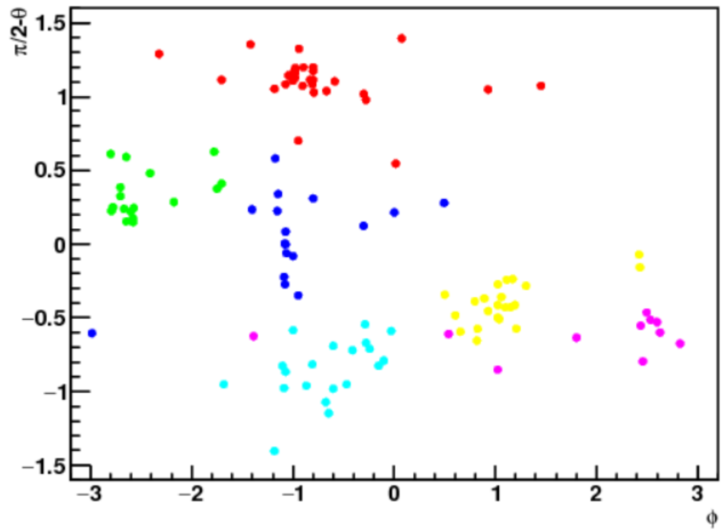
With CRF Num. of training events	Loss Train	Loss Test
900	0.308	1.25
1200	0.393	0.976
6000	0.418	0.728
7800	0.473	0.688
10000	0.500	0.666

- CRF looks better performance?
- Over fitting will vanish if num. of training events is O(10000)
  - Reaches 10000 evts., but still overfitting…
  - Performance gradually degrades, still poor performance
  - So need to optimize the network size to recover the performance

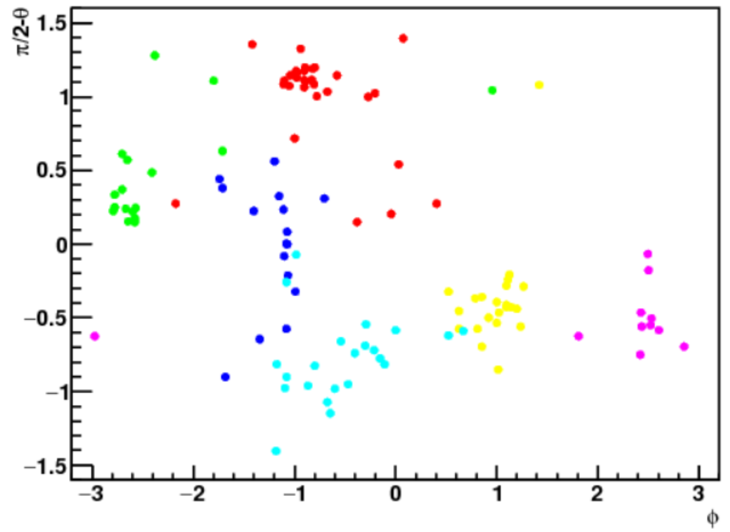
Examples(good?)

- Example 1:

Estimation

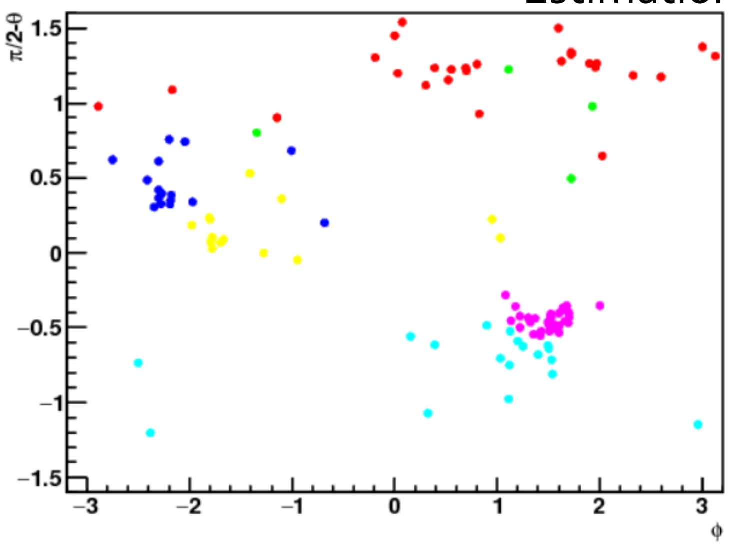


Answer

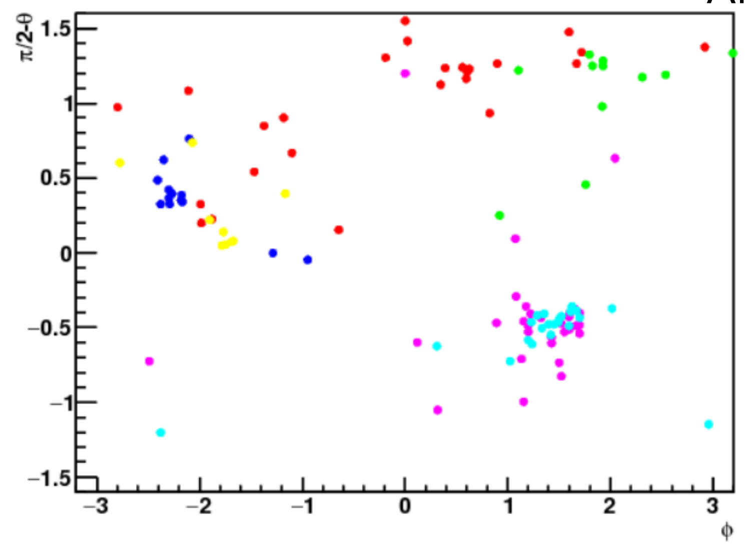


- Example 2:

Estimation



Answer

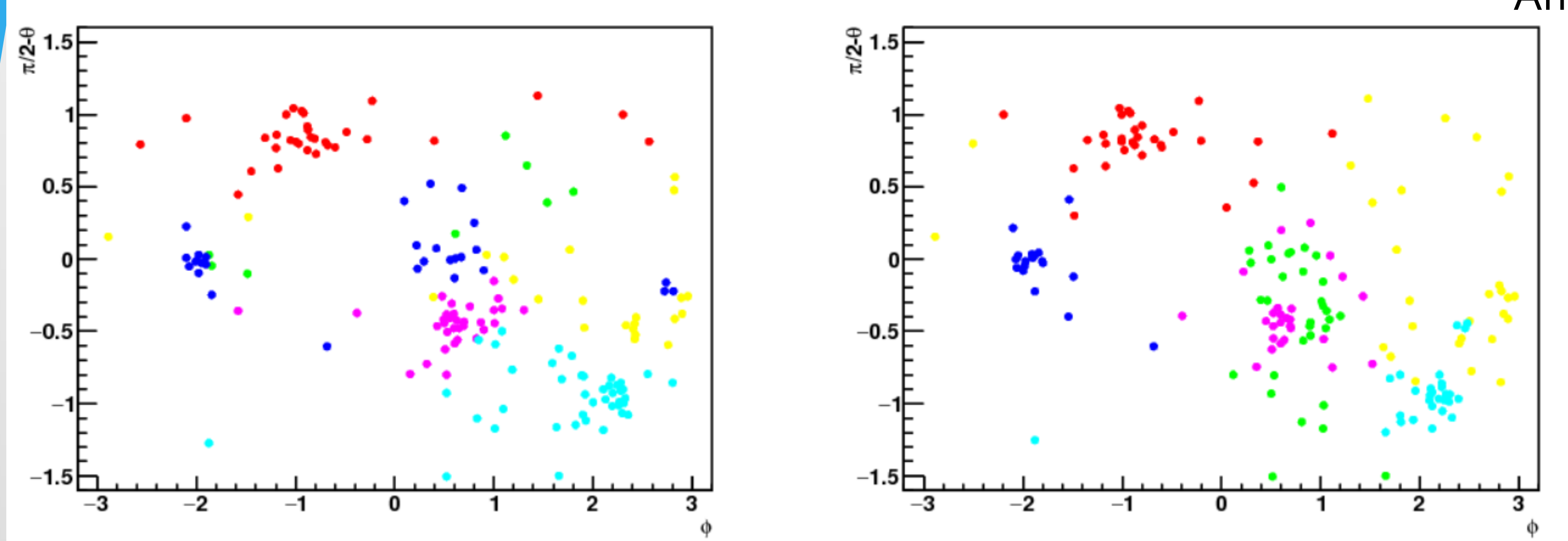


Examples(bad?)

- example 1:

Estimation

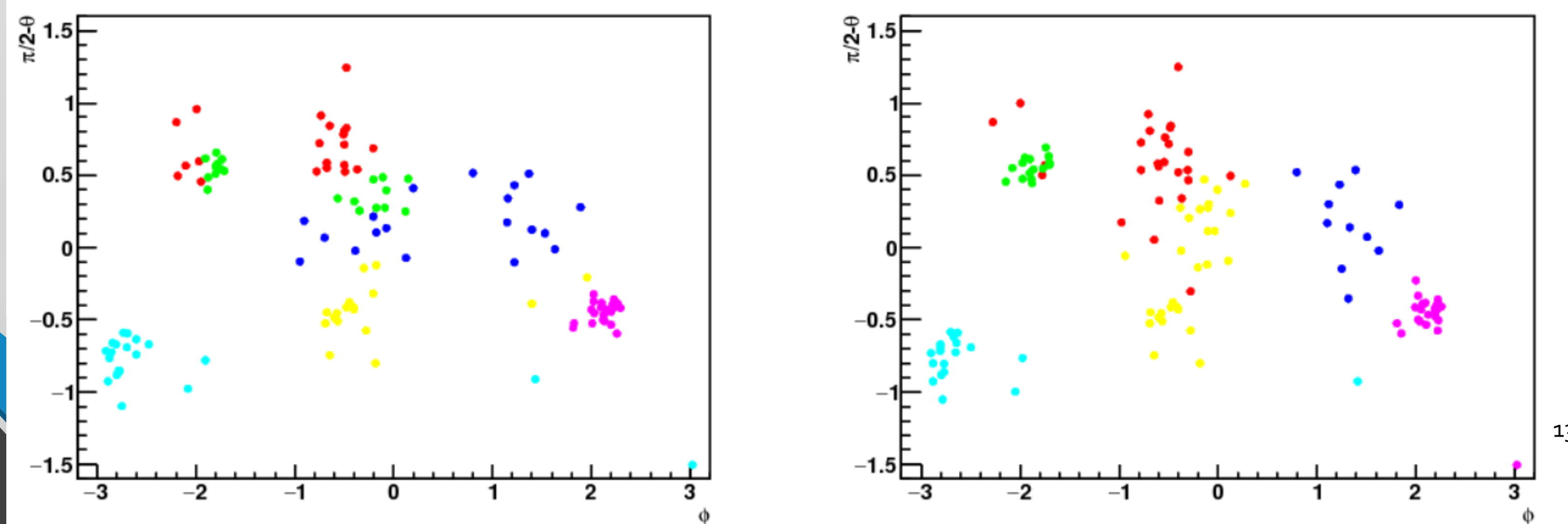
Answer



- Example 2:

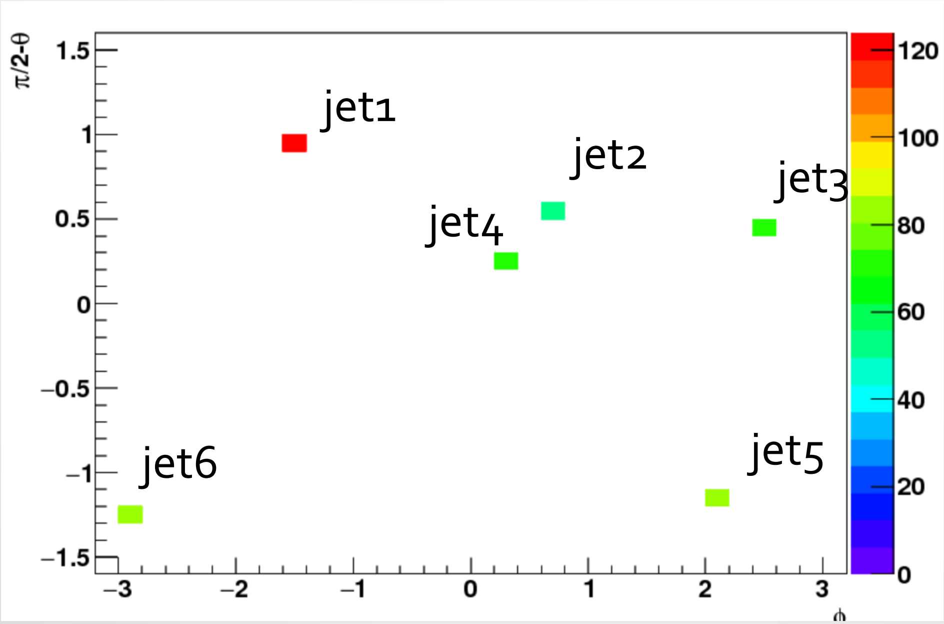
Estimation

Answer



Cause of bad colorization

- Coming from Jet numbering
  - So far, jets are ordered by θ(π/2-θ) value
  - Ordering from top to bottom of the image



- So, CNN is insensitive for ϕ direction…
  - Different energetic area with almost same θ value is regarded as same jet…
  - I tried some patterns of jet numbering rule, only this works well…

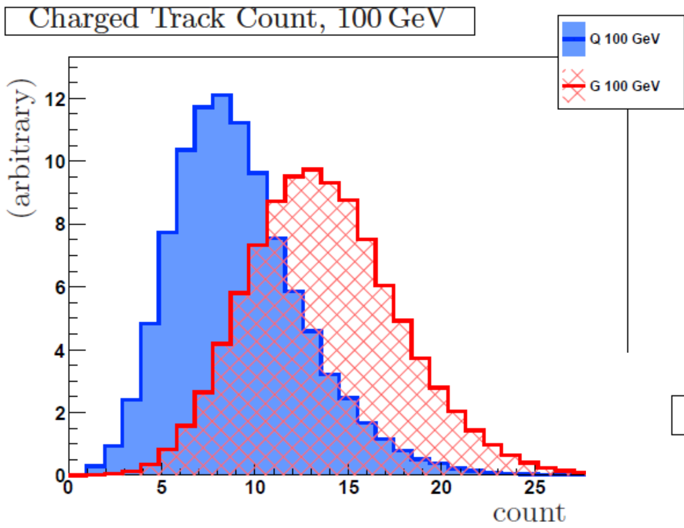
Problems

- So, need to learn the background(empty pixel)?
  - Start to try it
  - Training failed…
- CRF needs iteration to make energy function stable (& minimum)
  - 5-10 iteration necessary…
  - Now 4 iteration steps: I can adjust iteration and avoid memory problem!
- How about multi-input?
  - Apparently, only energy map is very difficult to make better jet clustering!
  - Charge, PID(=track mass), D0, Z0, etc…
  - e.g.) number of charged tracks in a jet has some power to separate quark/gluon jets

    ⇒charge map useful?

    Start to try it

arXiv: 1211.7038



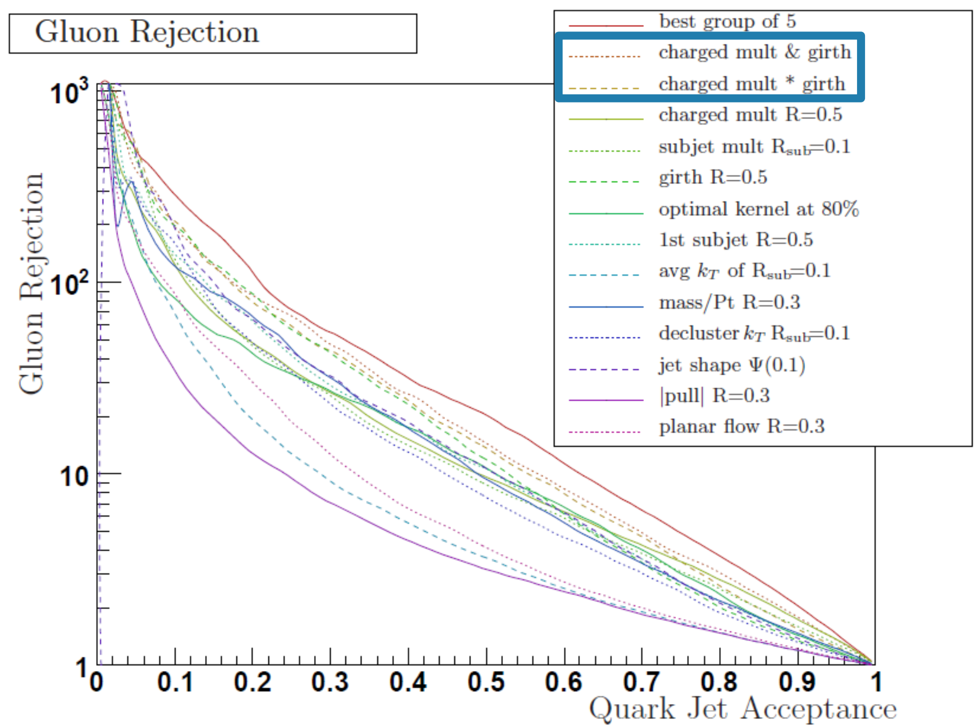
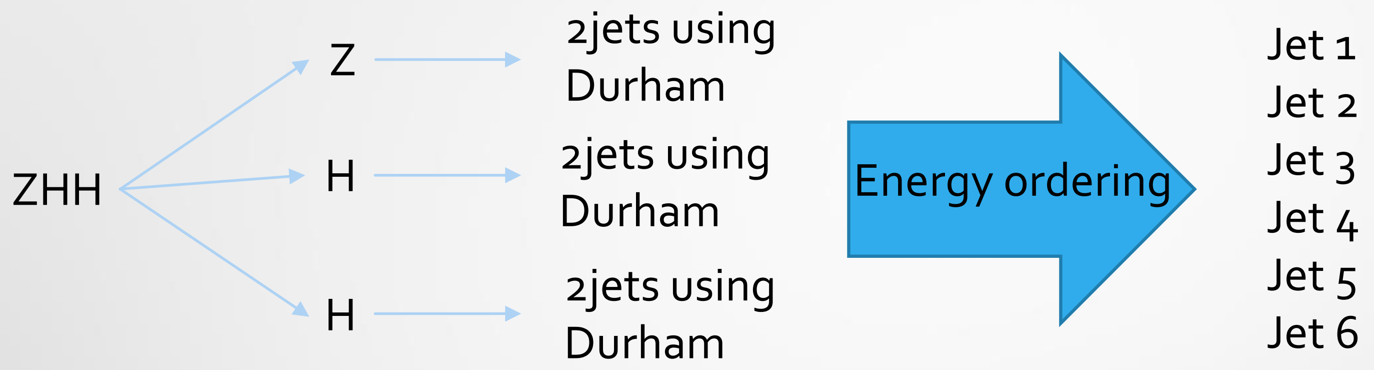


Figure 22. ROC curves for 100 GeV PYTHIA8 jets for selected variables. These curves show the background (gluon jet) rejection efficiency ($1/\varepsilon_B$) as a function of the signal (quark jet) acceptance efficiency (ε_S).

backups

notation

- Create "answer" jets: perfect Durham jet clustering



- Numbering jets
  - Simply, energy ordering of the jets
  - So far, there seems no dependence of the jet direction!
    - Thanks to CNN (c.f. CNN can absorb position shift effect)