Automatic Colorization for Jet Clustering

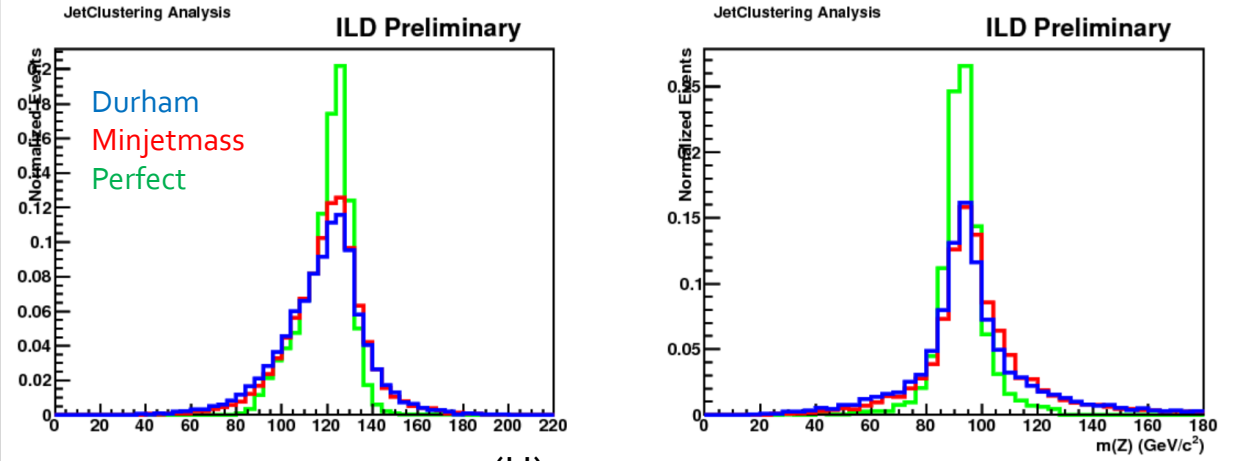
Masakazu Kurata

General Meeting

02/09/2018

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@500GeV(ZHH): ～40% improvement if perfect!

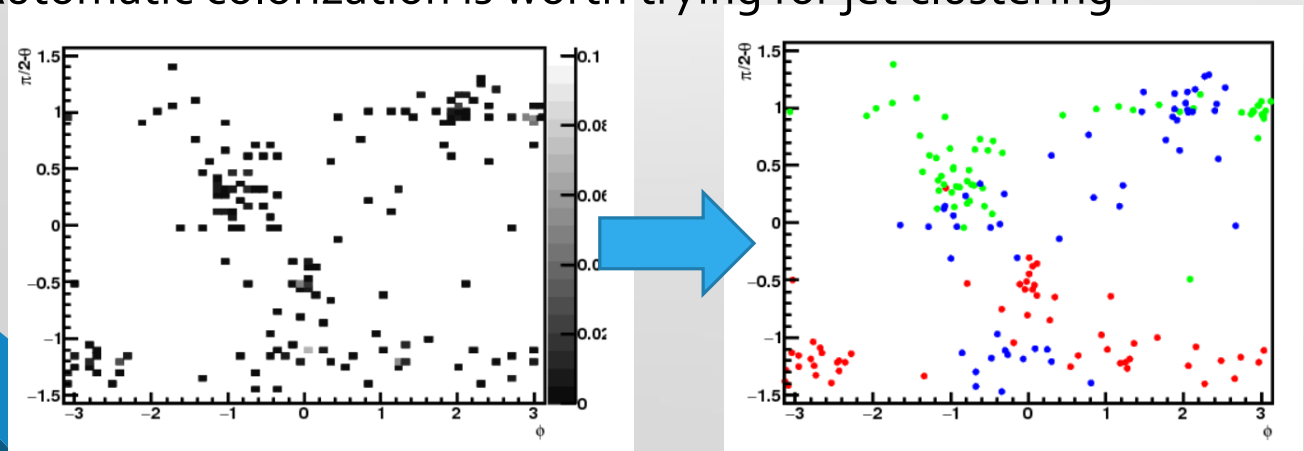


m(H)

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

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 Convolutional Neural Network(CNN), we will extract both of the features
  - Encorder-Decorder type CNN is used (calls as u-network)
  - Already ～30 layers in CNN! → DEEP LEARNING
- Clustering is equivalent to "colorize" each particle in the same cluster
  - Grey scale ⇒ color
  - So, Automatic colorization is worth trying for jet clustering



Example

Grey scale

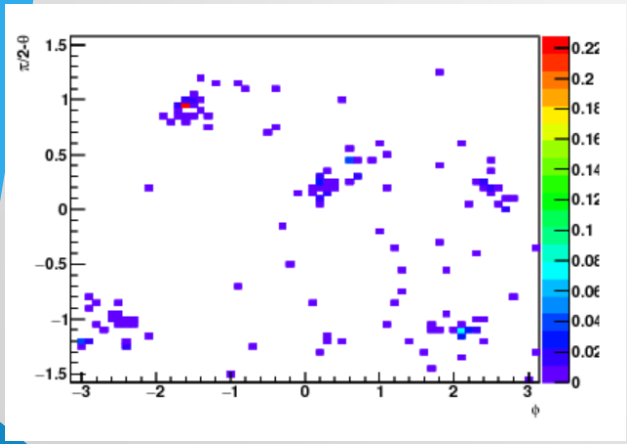
Expectation

Truth

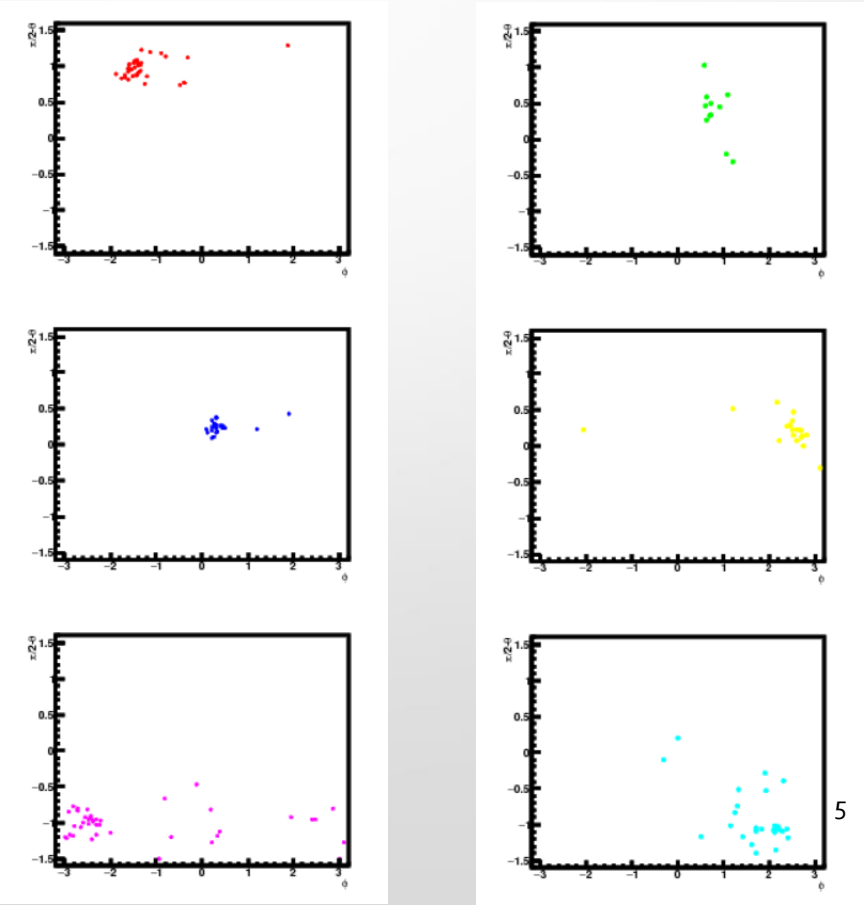
Trial

- Using a certain map(s) of each event, estimate color of each track
  - Do not consider color-singlet state

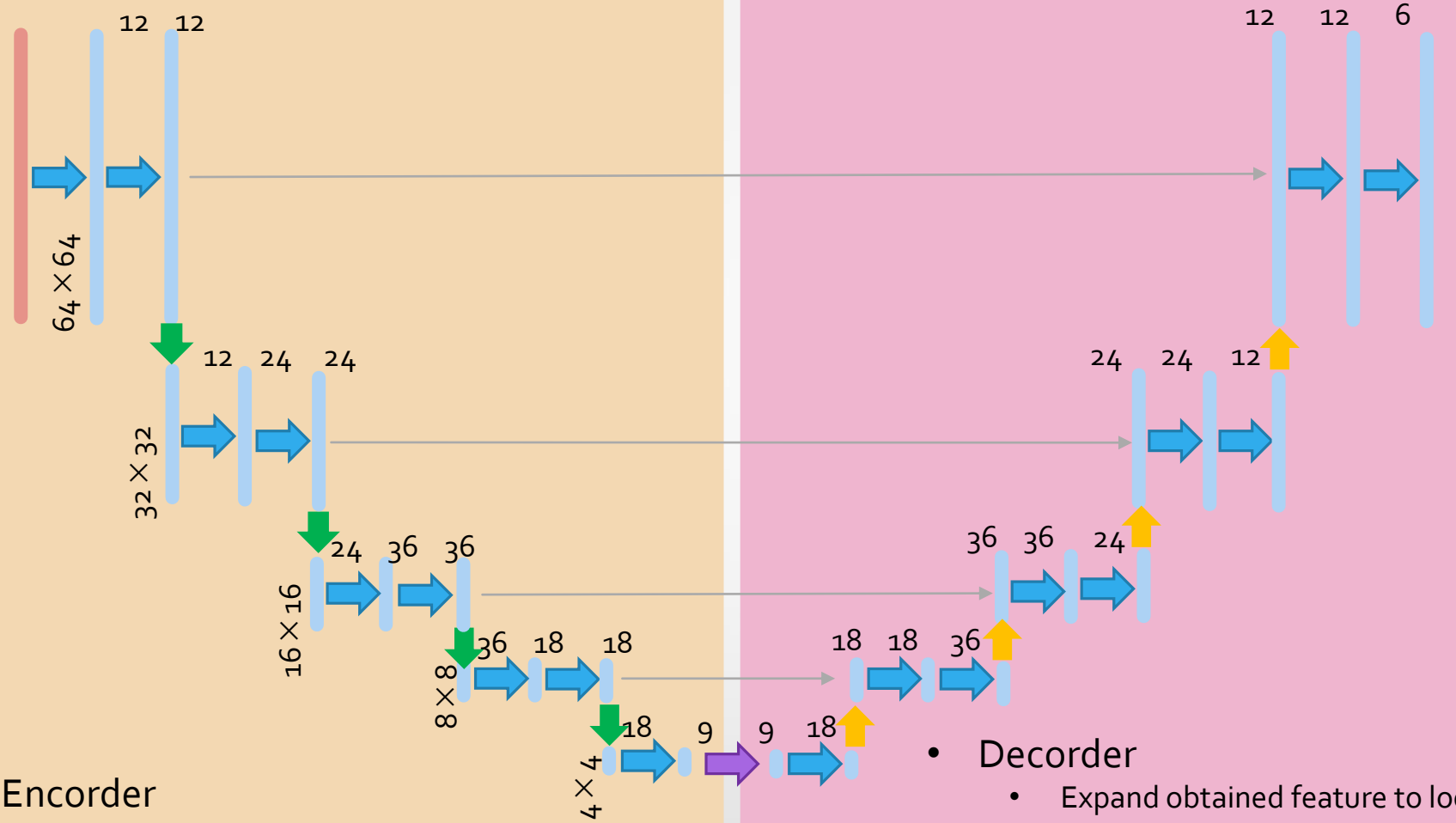
Input(64×64pixel figure)
e.g.) energy map



Output(64×64pixel figure)



U-network

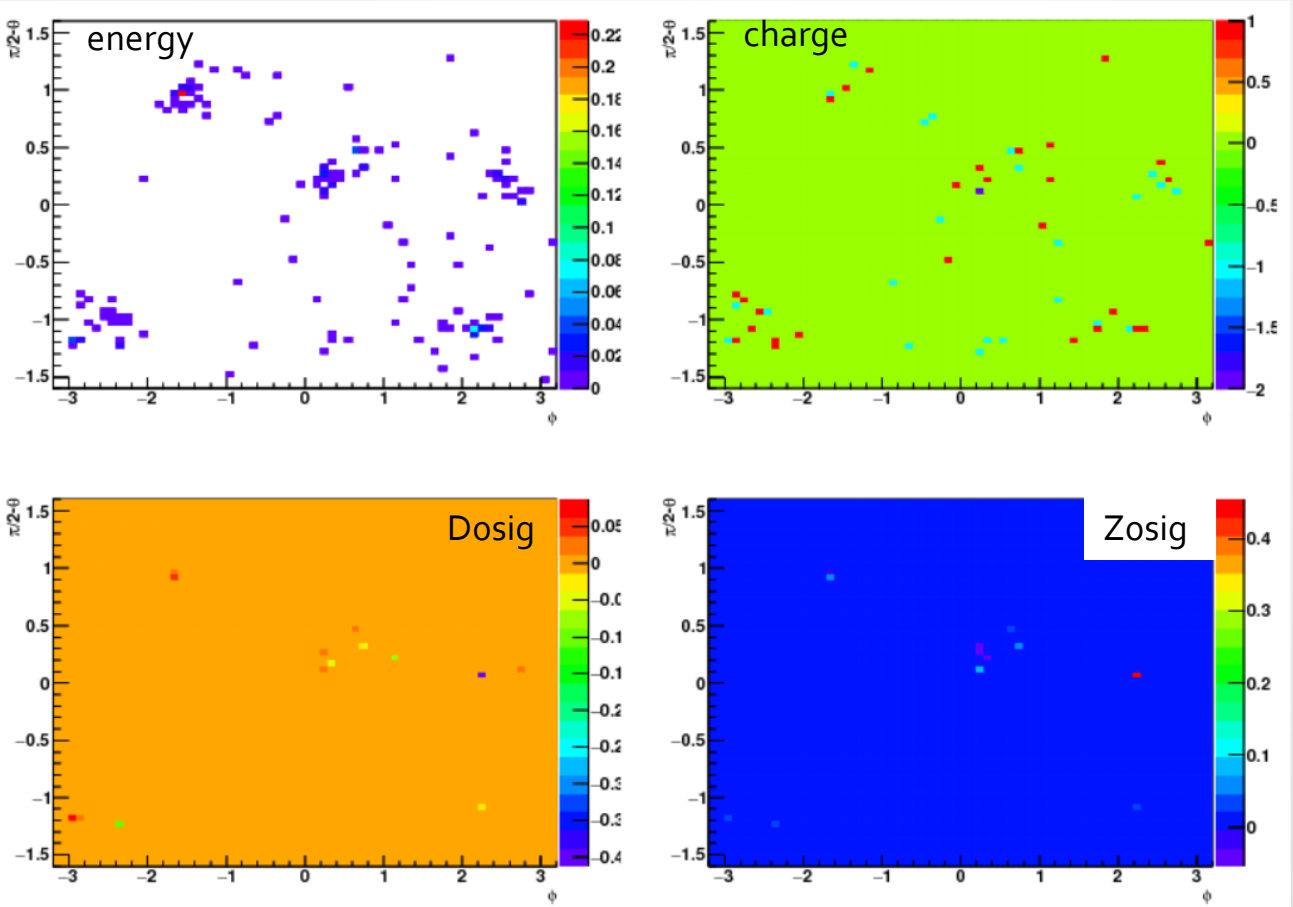


- Encorder
  - Extract global & higher order feature
  - Downsample to make network robust for distortion & shift effect
  - Lost position information
- Decorder
  - Expand obtained feature to local
  - Upsample to recover position information
  - Merge encoder nodes to get precise position information

- Consider periodic condition in ϕ direction

Multiple input

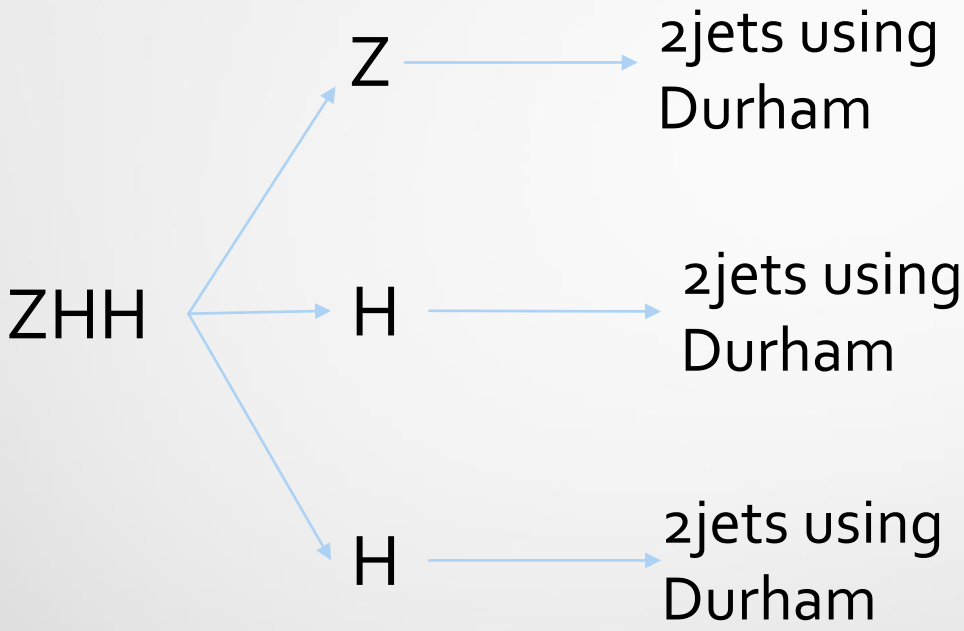
- Several variables are used for input image
  - Energy map
  - Charge map
  - Dosig map
  - Zosig map



- Training goes to quicker convergence than that of energy only
  - Not guaranteed good input variable set: need much time to check…
- Trying to include Ecal and Hcal maps: so far, almost same performance…
- dE/dx??

Create answer

- Supervised learning - Create "answer" jets: perfect Durham jet clustering

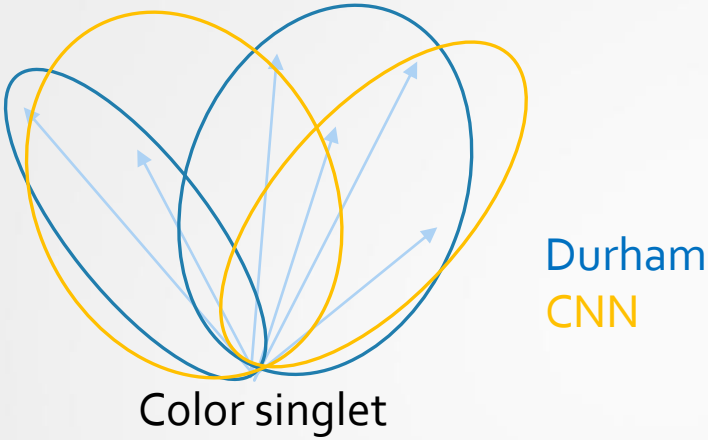


So far, do not consider color singlet state: number of jets is 6

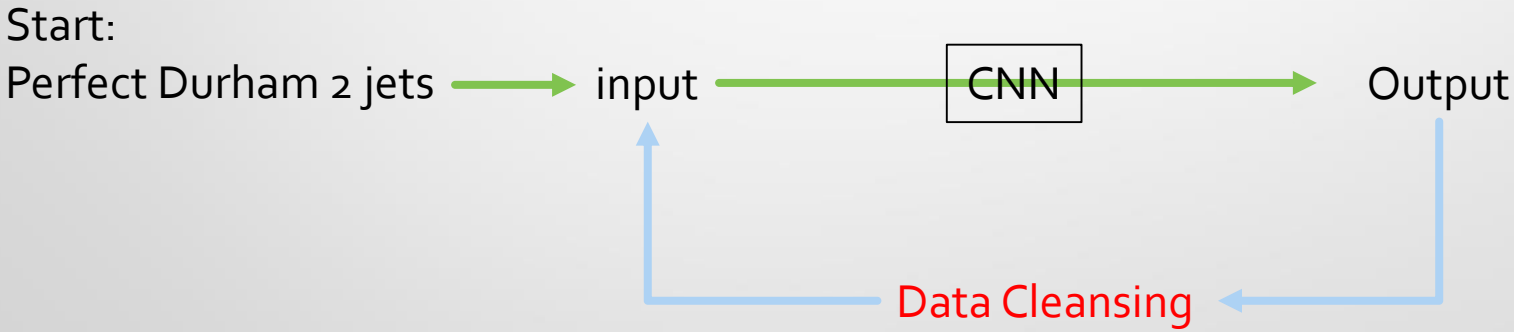
ZHH→(qq)(bb)(bb)→6jets

Data Cleansing

- Perfect Durham clustering is not always the best clustering into 2 jets for CNN



- By using the preliminary training weights, clustering into 2 jets is performed



Clustering particles to make loss function minimum

Iterate some times

status

- Use ZHH→(qq)(bb)(bb): 6jets clustering
- Use 104000 events for training
- Don't consider color singlet state for network training
  - But use the freedom of color singlet state: Data cleansing for better performance
- Input: 4 images output: 6 images
- Structure: u-network + CRF(don't mention today)

Over fit check

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

$$L = -\frac{1}{N}\sum_{jet}\sum_{track} f\left(\frac{E_{track}}{E_{jet}}\right) \cdot \mathrm{Log}(y_{track})$$

- Cross entropy is weighted by the function of track energy to introduce the importance
  - Energetic tracks should be colored correctly
  - Comparison between train & test

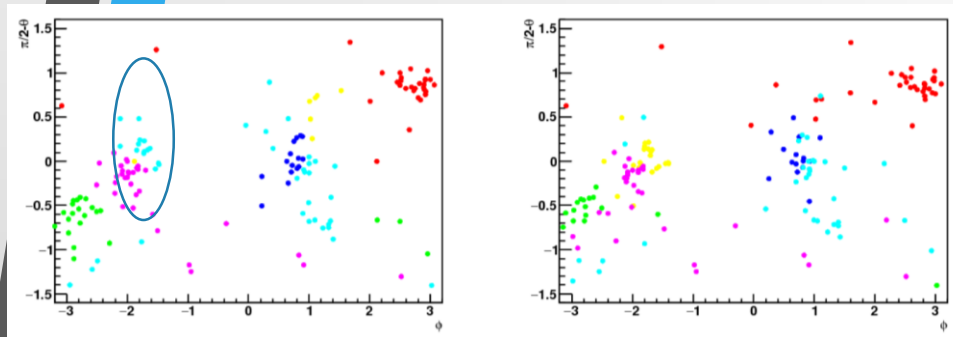
Num. of training events	L_{Train}	L_{Test}
900	0.308	1.25
11500	0.485	0.642
50000	0.374	0.386
56000	0.283	0.305
104000	0.271	0.312

- Performance seems going better gradually
  - But overfitting again… (though still under training)
  - need more!!

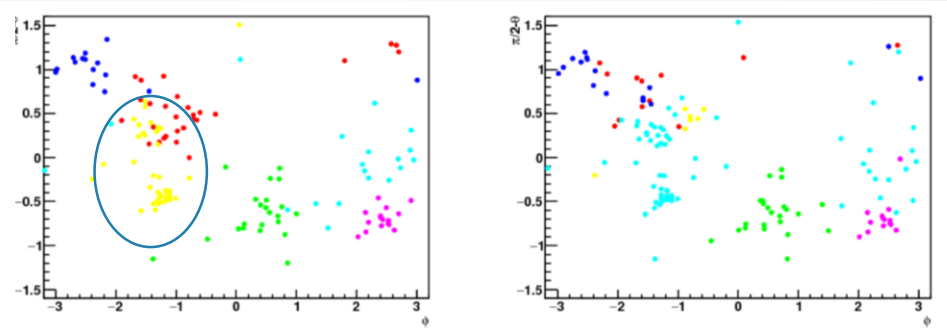
Examples(bad)

- Using test samples

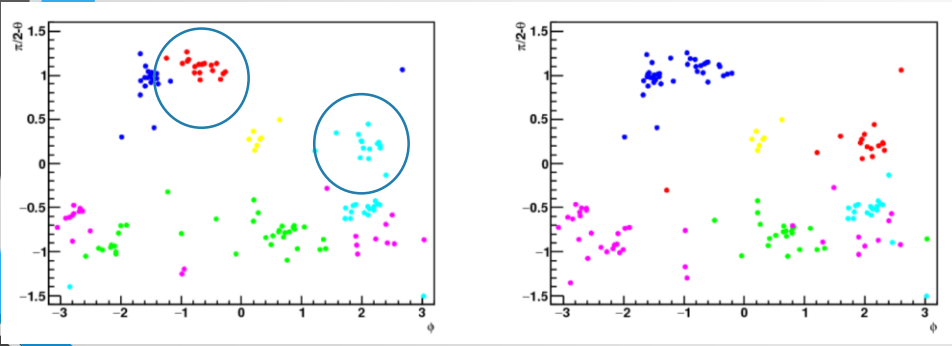
Example 1 Estimation Answer



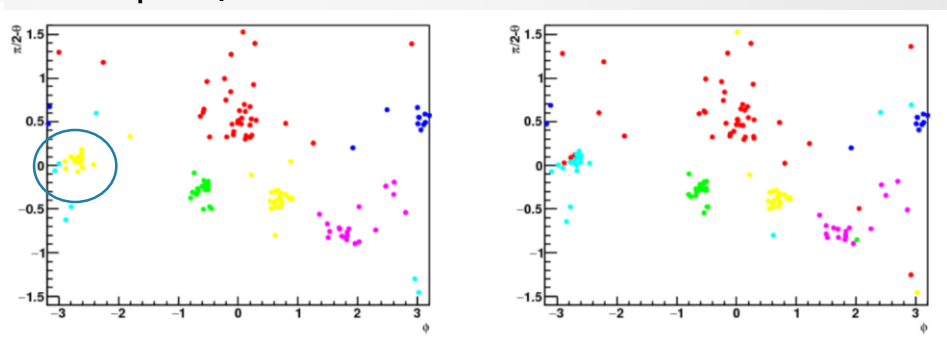
Example 2 Estimation Answer



Example 3 Estimation Answer



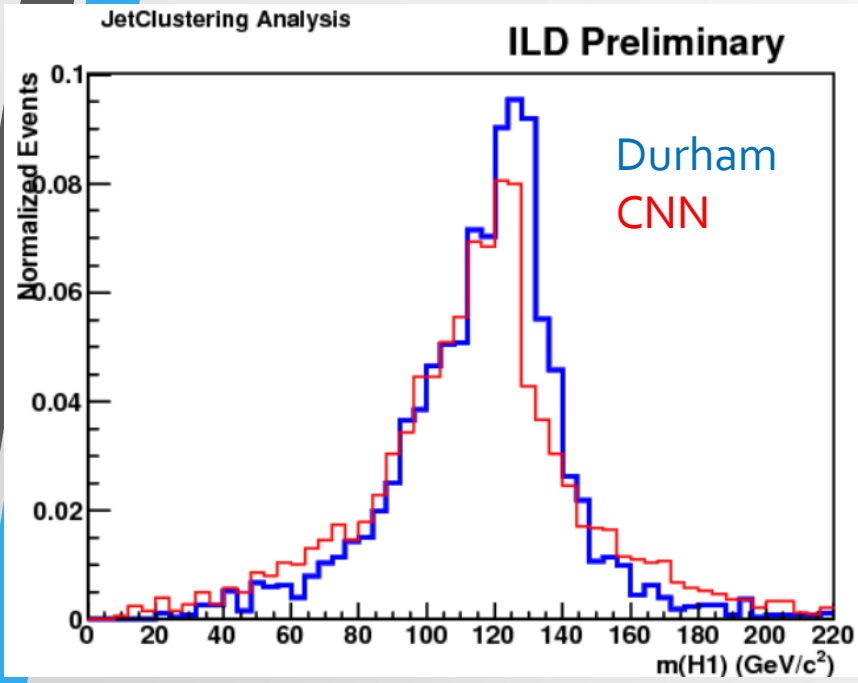
Example 4 Estimation Answer

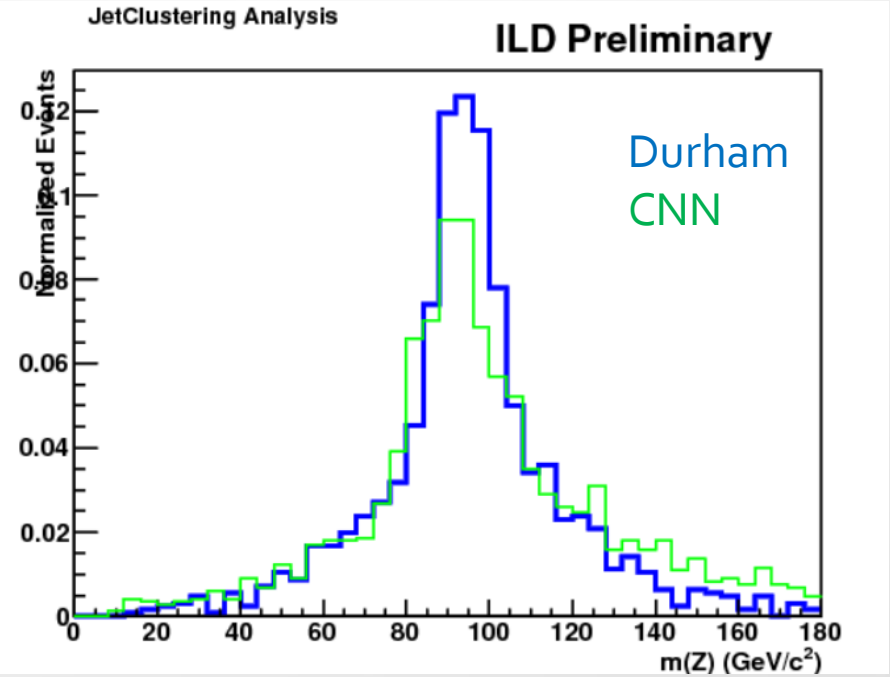


- Need to investigate mis-colorize
- Split cluster(Gluon splitting?) is difficult to colorize…
- Of course some events are very difficult to colorize correctly…

- Lets go to reconstructing mass!
  - I know performance does not reach even Durham level so far…
  - How much far from Durham
- Assume color information is known

Mass distribution





- 1500 evts. test samples
- Still something like that…

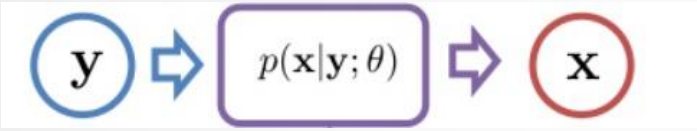
Summary and Outlook

- Continue to investigate
- There are many ideas which are likely to improve, but no time to try now…

backups

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
- 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 (because inspired by Ising model):

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

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

Preliminary architecture

- CRF is used for post-processing of CNN

$$E(\mathbf{x}) = \underbrace{\sum_{(i,j)\in\mathcal{E}} f_{ij}(x_i, x_j)}_{\text{Constraint for pairwise tracks}} + \underbrace{\sum_{i=1}^{N} f_i(x_i)}_{\text{Output of u-network for each track}}$$

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

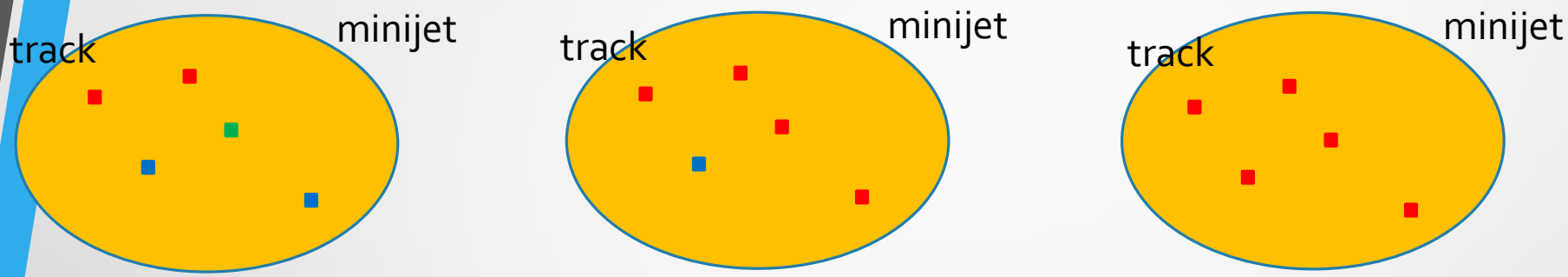
$$f_{ij} = \omega \cdot \exp\left(-\alpha \frac{2Min\left(E_i^2, E_j^2\right)}{Evis^2}(1 - cos\theta)\right)$$

  - 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?

Including physics effect naturally

- Include minijet effect
- Like spin correlation of Ferromagnetism, members inside minijets should be same status to make energy minimum



High energy state | Middle energy state | Low energy state

- Energy of Boltzmann probability:

$$E(x) = \underbrace{\sum_{i} f_i(x_i)}_{\text{External field (output of CNN)}} + \underbrace{\sum_{(i,j)} f_{ij}(x_i, x_j)}_{\text{Pairwise effect}} + \underbrace{\sum_{minijet} f_{minijet}(\boldsymbol{x_{trks}})}_{\text{Status inside each minijet}}$$

- Can include vertex constraint with same way
- Now: no reason, but 30 minijets created