Construct Deep Jet Clustering

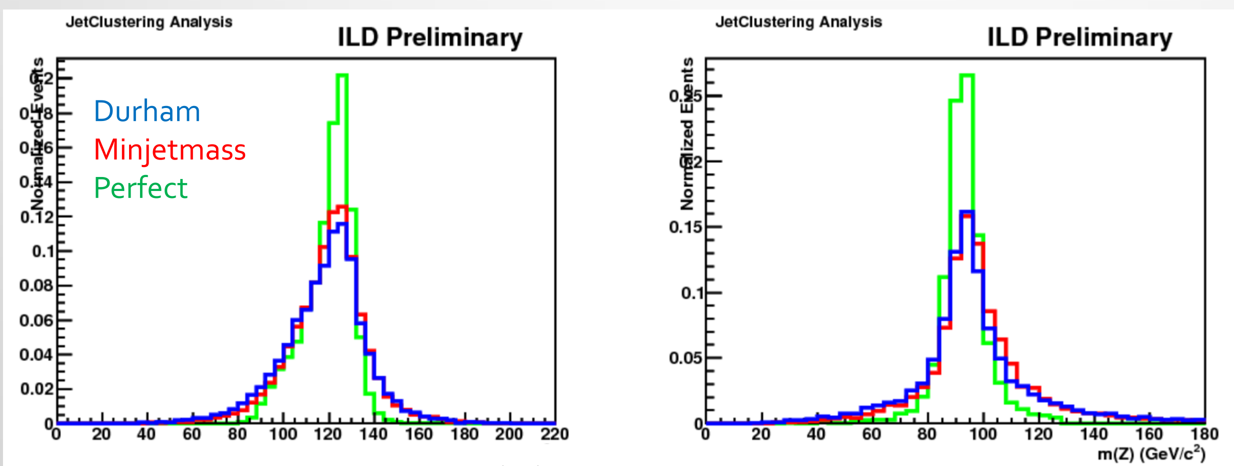
Masakazu Kurata

LCWS2019

10/29/2019

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!

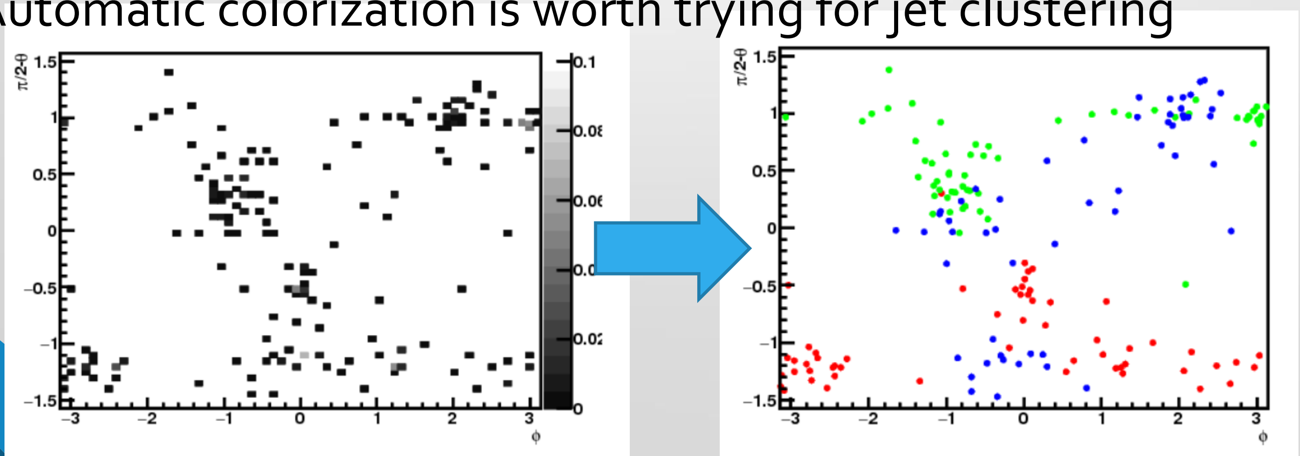


m(H)

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

Use CNN for automatic colorization

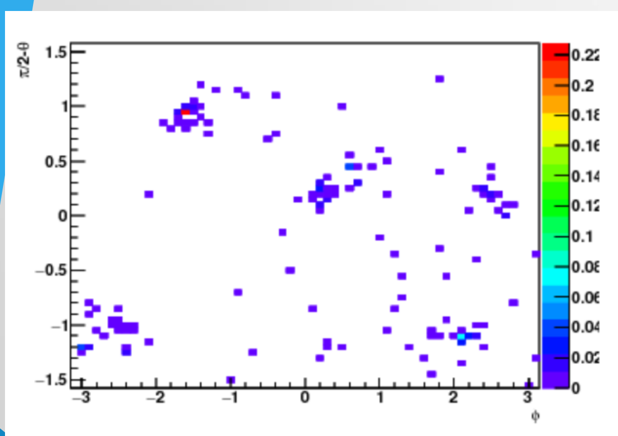
- For jet clustering, we need the global and local information for each event
  - Global: Where is the large energy located?
  - Local: Correlation between neighbors or large energy area?
- Using Convolutional Neural Network(CNN), we will extract both features
  - Encorder-Decorder type CNN is used (calls as u-network, mention later)
- Clustering is equivalent to "colorize" each particle in the same cluster
  - Grey scale ⇒ color
  - So, Automatic colorization is worth trying for jet clustering



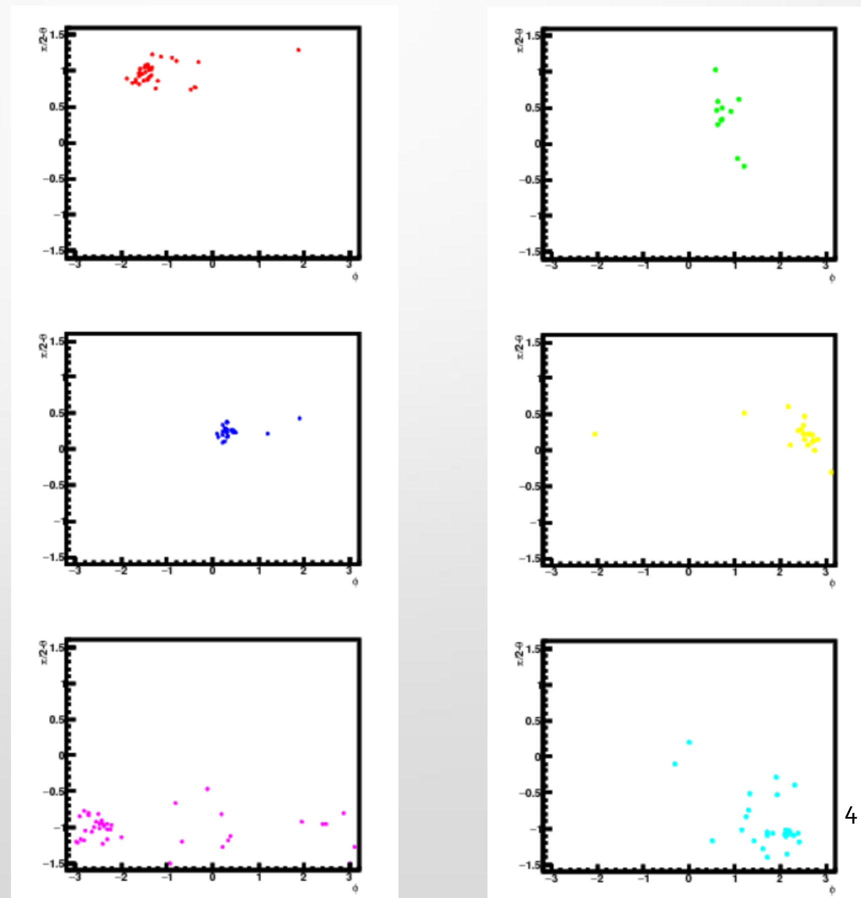
Trial

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

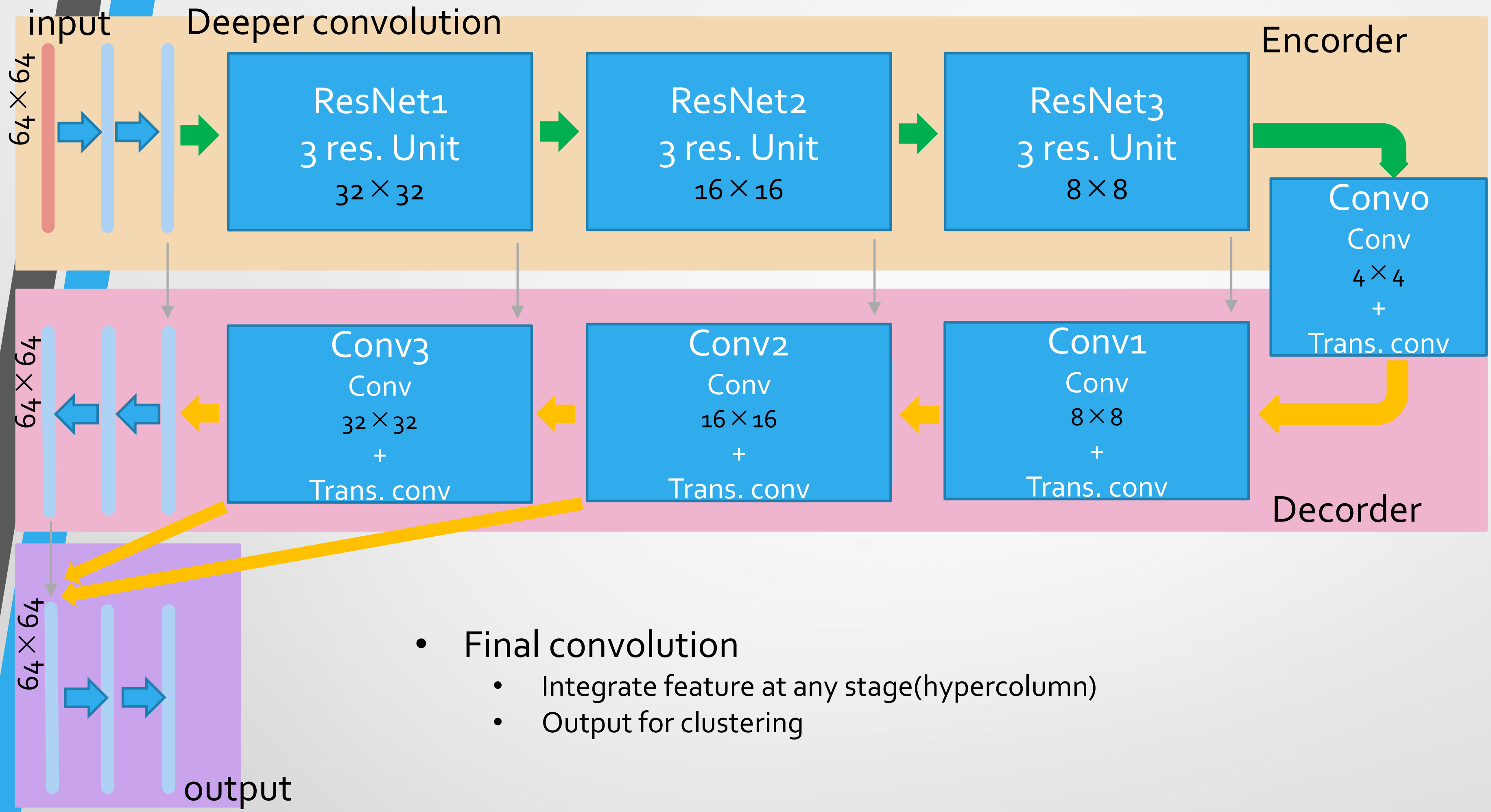
Input(64×64pixel figure)
e.g.) energy map



Output(64×64pixel figure)



Network Architechture



- Final convolution
  - Integrate feature at any stage(hypercolumn)
  - Output for clustering

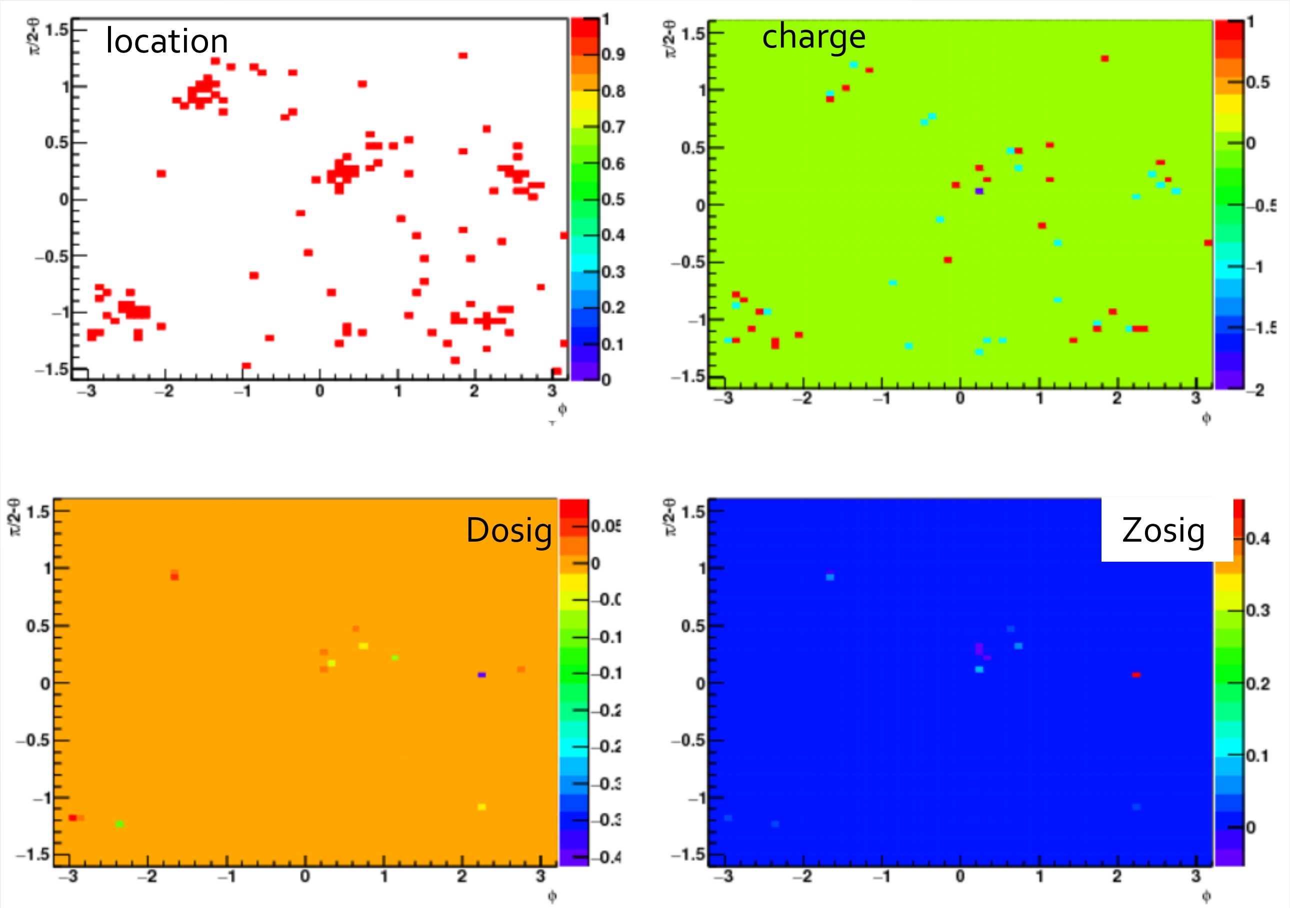
- Encorder
  - Extract global & higher order feature
  - Downsample to make network robust for distortion & shift effect
  - Lost position information

- Decorder
  - Expand obtained feature to local
  - Upsample to recover position information
  - Merge encoder nodes to get precise position information

Multiple input

- Several variables are used for input image
  - Location map
  - Charge map
  - Dosig map
  - Zosig map
  - Ecal map
  - Hcal map
  - Energy:

    use different way



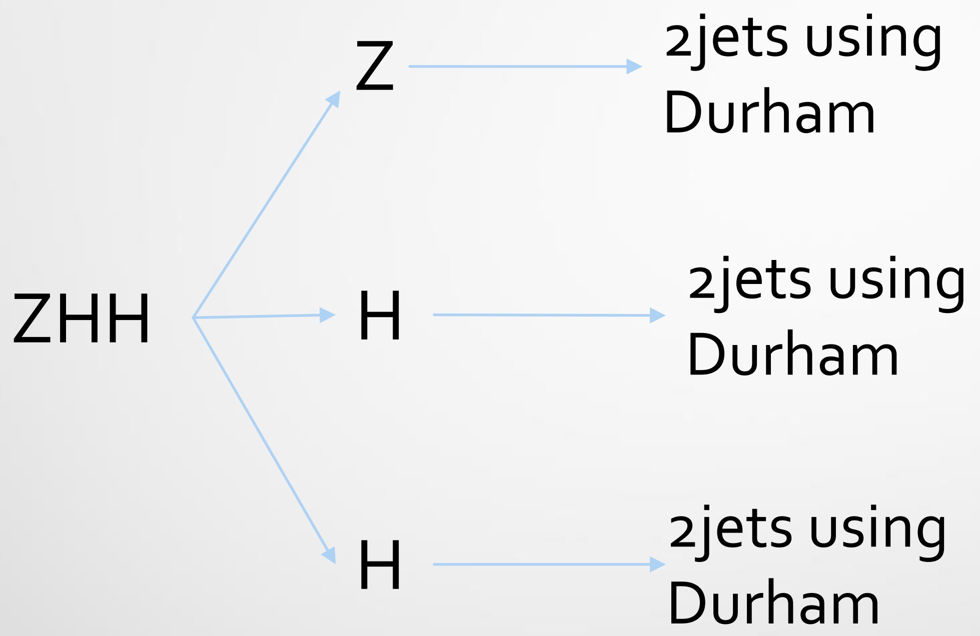
- Training goes to quicker convergence than that of energy only

  Not guaranteed good input variable set: need much time to check…

- dE/dx??

Create answer

- Supervised learning - Create “answer” jets: perfect Durham jet clustering

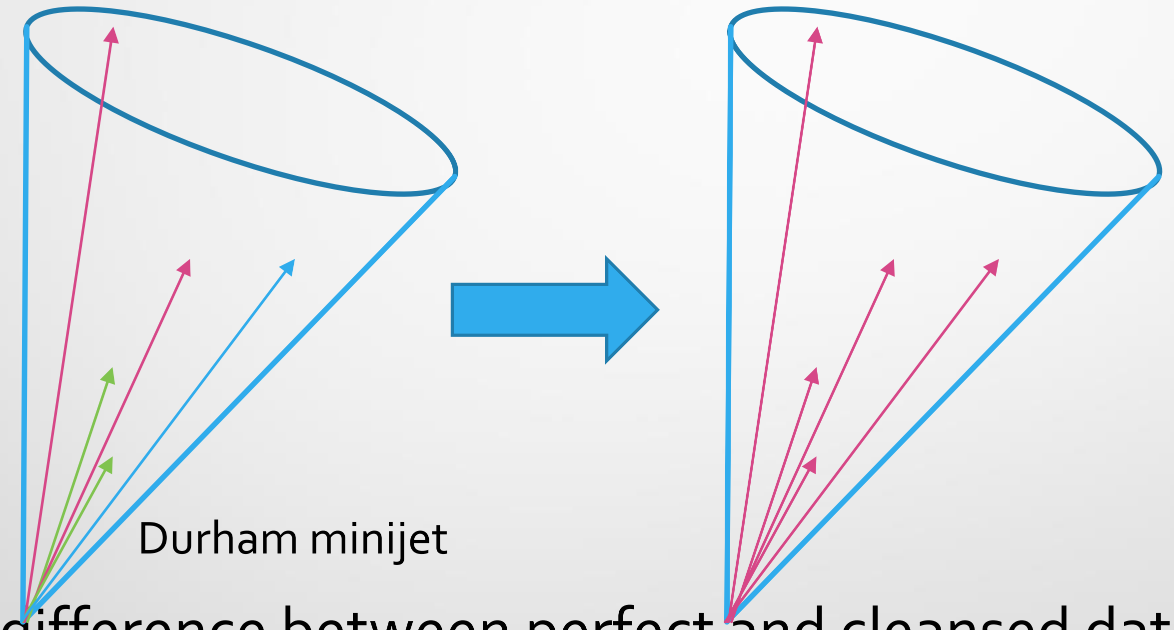


So far, do not consider color singlet state: number of jets is 6

ZHH→(qq)(bb)(bb)→6jets

Data Cleansing1

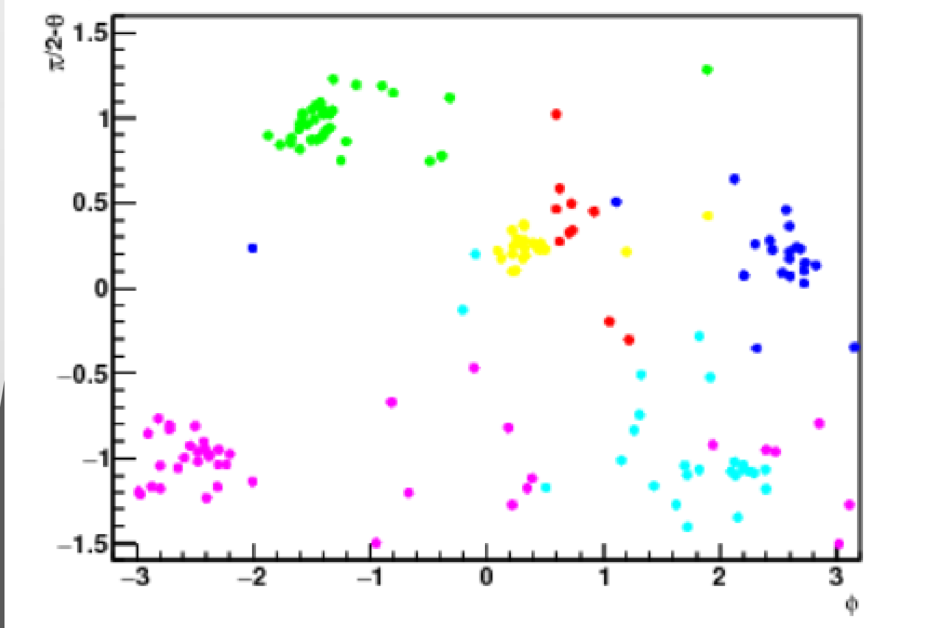
- So far, power of expression is not enough to assign jets perfectly
  - Overlapping of jets creates very complicated boundary shape
  - Very complicate separation is difficult: it looks "noise" for CNN
- Using Durham jet clustering, bundle particles into some minijets



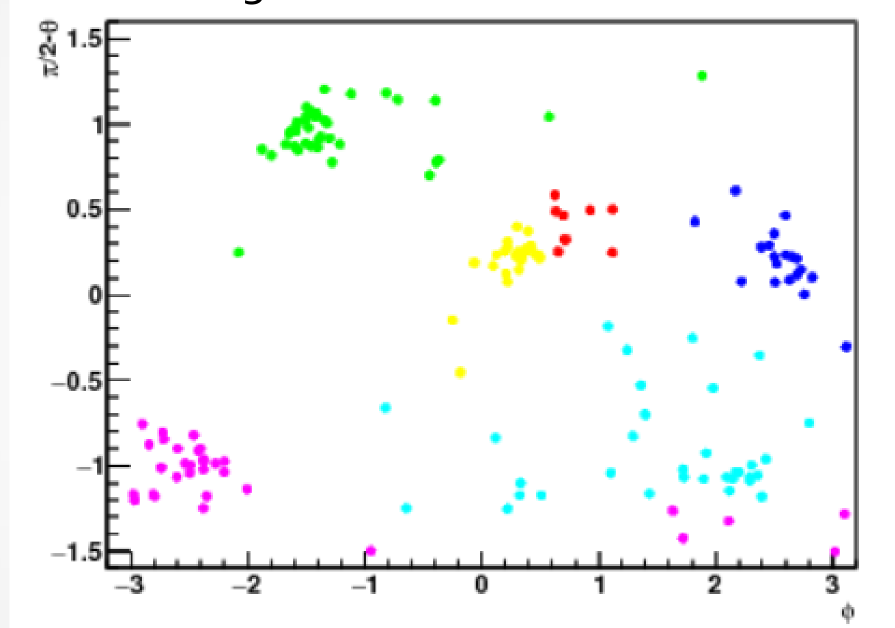
- 10% mass difference between perfect and cleansed data is allowed
  - Bundling particles into as small number of minijets as possible
  - Num. of minijets is different event by event

Effect on clusters

perfect



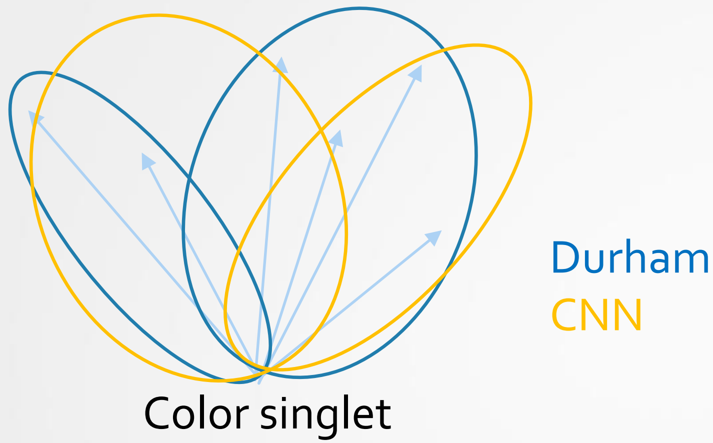
Cleansing



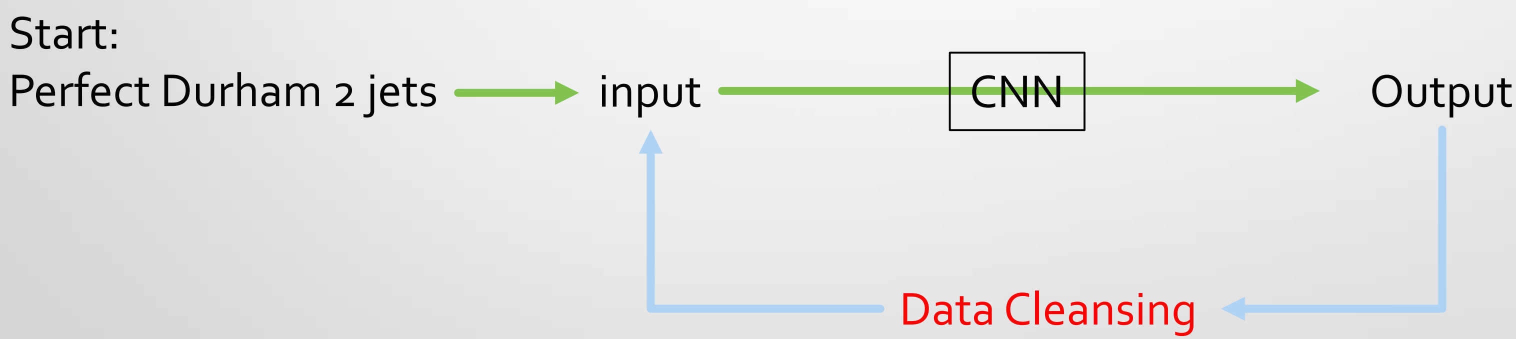
- Particles in jet can be gathered same region
- Clearer boundary between jets

Data Cleansing2

- 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



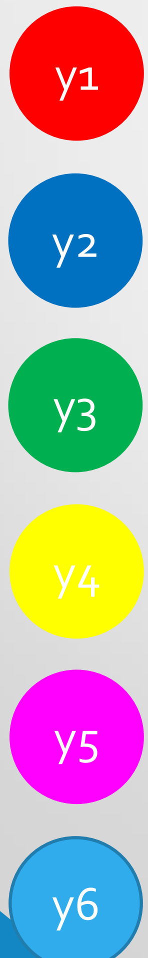
Clustering particles to make loss function minimum

Iterate a few times

About output

- Output is obtained as the probability of assignment to each color
- Color with max probability is assigned

output



$$\sum_i y_i = 1.0$$

$$color = \max y_i$$

status

- Use ZHH→(qq)(bb)(bb): 6jets clustering
  - q: uds or c
- Use 112000 events for training
- Don't consider color singlet state for network training
  - But, as mentioned, use the freedom of color singlet state: Data cleansing for better performance
- Input: 6 images output: 6 images
- Structure: mentioned above(resnet + hyper column)

Over fit check

- Performance comparison
  - Using cross entropy loss(small is better performance)

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

- Using energy, importance is defined
  - Larger energy particles should be assigned more correctly than lower energy particles
  - So, larger energy particles have larger importance on loss function

- Loss function will be almost same value if no over fitting:

Cross-entropy loss	Train	Test
L	0.329	0.355

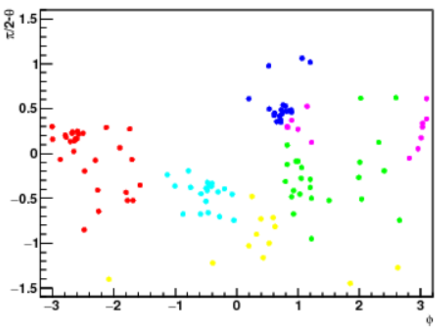
  - Almost same – over fitting will be small
  - More training necessary…

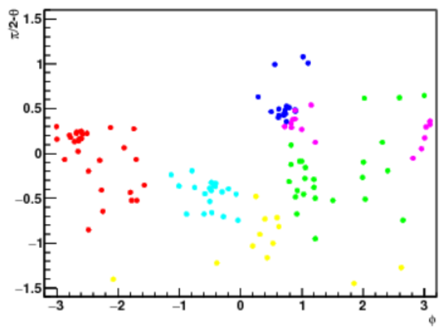
- Should be lower for better performance

Examples

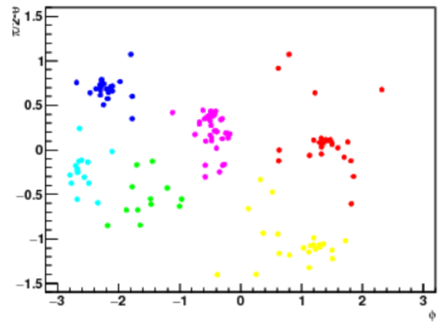
- Using test samples

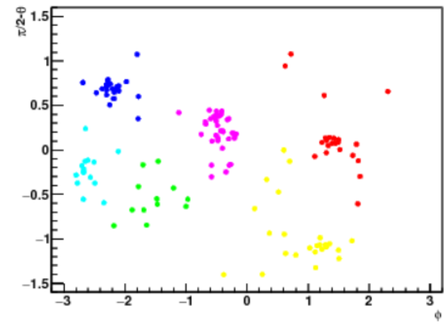
Example 1 Estimation Answer



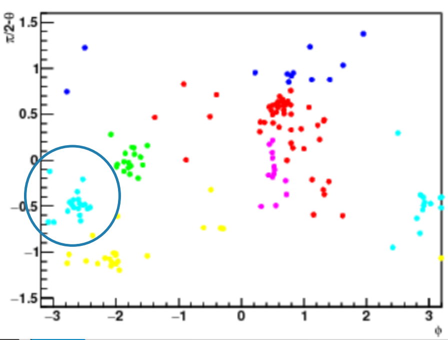


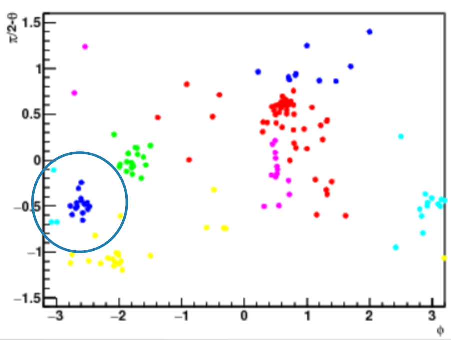
Example 2 Estimation Answer



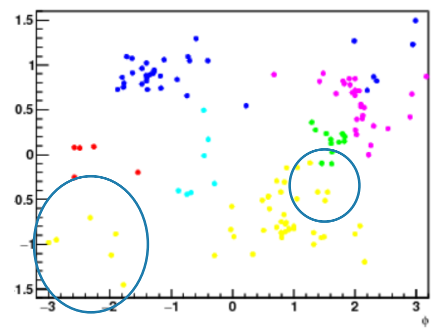


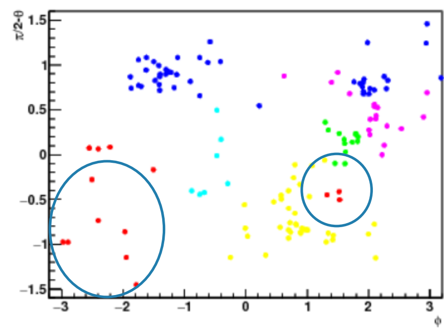
Example 3 Estimation Answer





Example 4 Estimation Answer

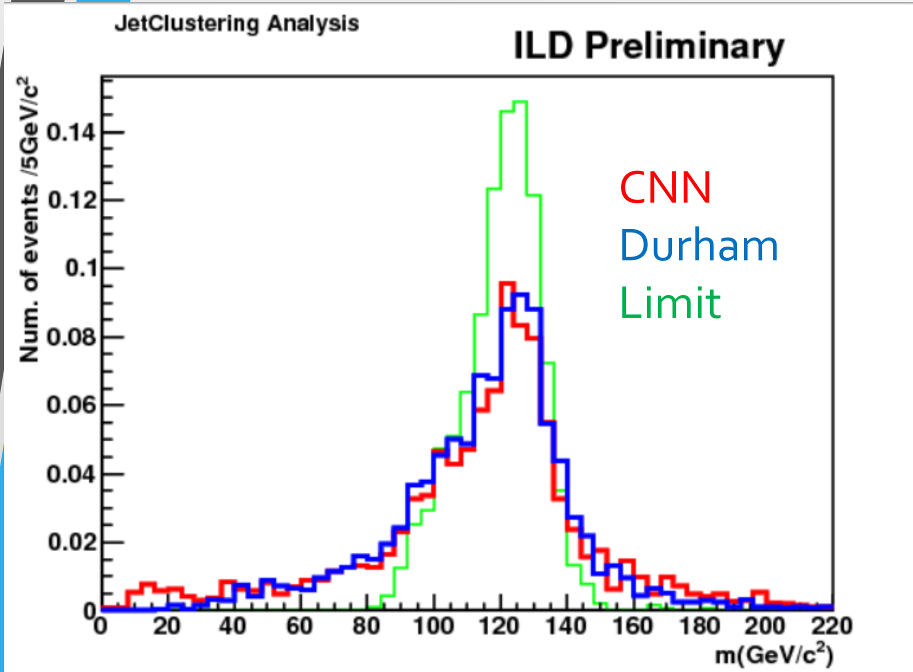


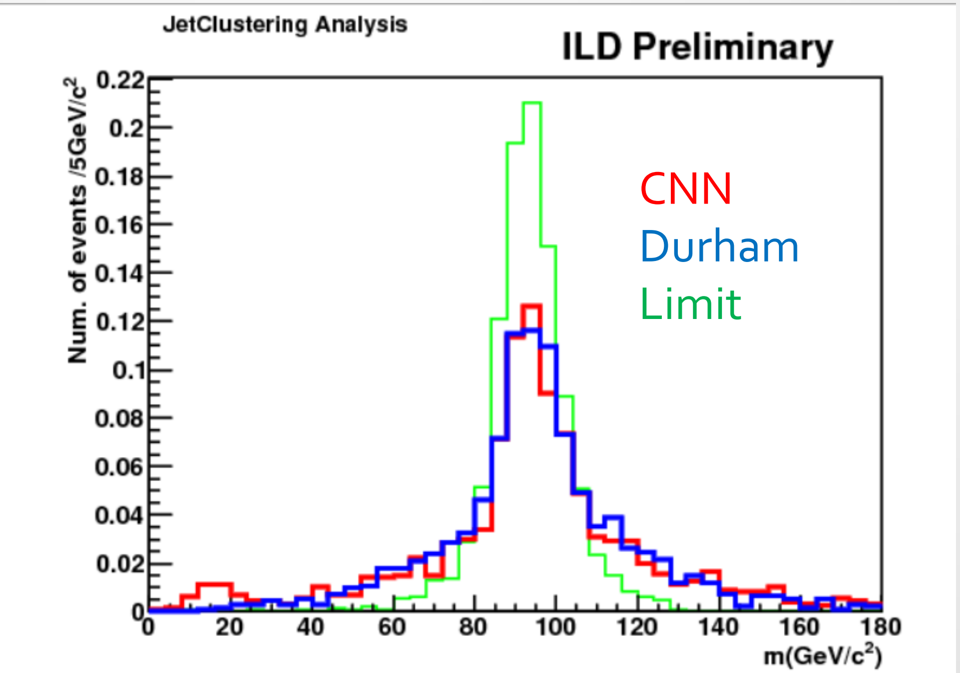


- Need to investigate mis-colorize
- Split cluster(Gluon splitting?) is difficult to colorize…
- Of course some events are very difficult to colorize correctly…

- Let’s go to check mass reconstruction
  - Check how much far from Durham
- Assume color information is known
  - Jet pairing is solved

Mass distribution





- 1500 evts. test samples
- Reach Durham level
  - Still far from limit…

Summary and Outlook

- Mass resolution reaches Durham level
- Input data quality is very important to final results
  - Suppressing "noise" leads to good result
  - So, have to explore better input data correction
  - Of course, Higgs & Z mass resolution should be kept well
- Continue to investigate:
  - Gluon splitting
- Performance is going better, need to explore better performance!
- Next steps:
  - Check bias of uds & c jets – how about (bb)(bb)(bb) case?
  - Bias of process - Other process??
  - Lead to better separation between signal & backgrounds?

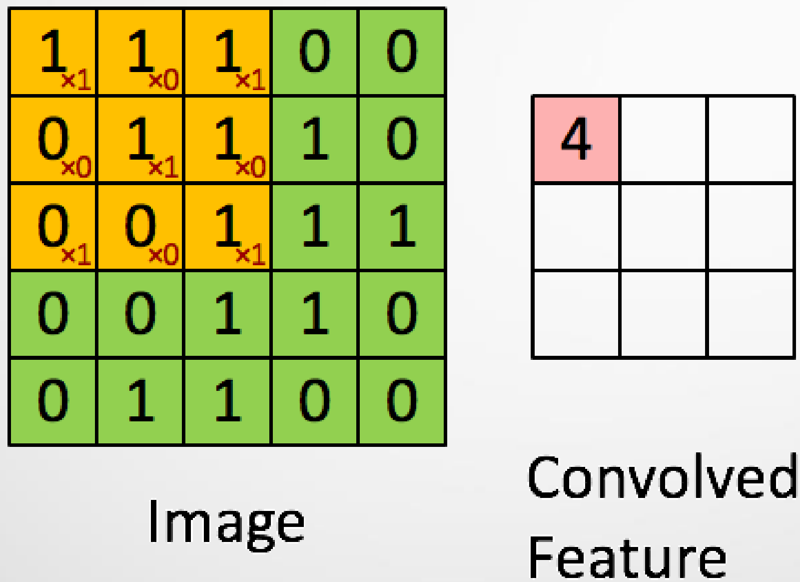
backups

Basics: convolution

- Convolution: Apply the filters to extract the feature
  - Sum of the product of each pixel and filter weights:

    $$y_{kl} = \sum_{i,j} w_{ij} \cdot x_{(k+i)(l+j)} \; (+b)$$

  - Slide filters over all the pixels



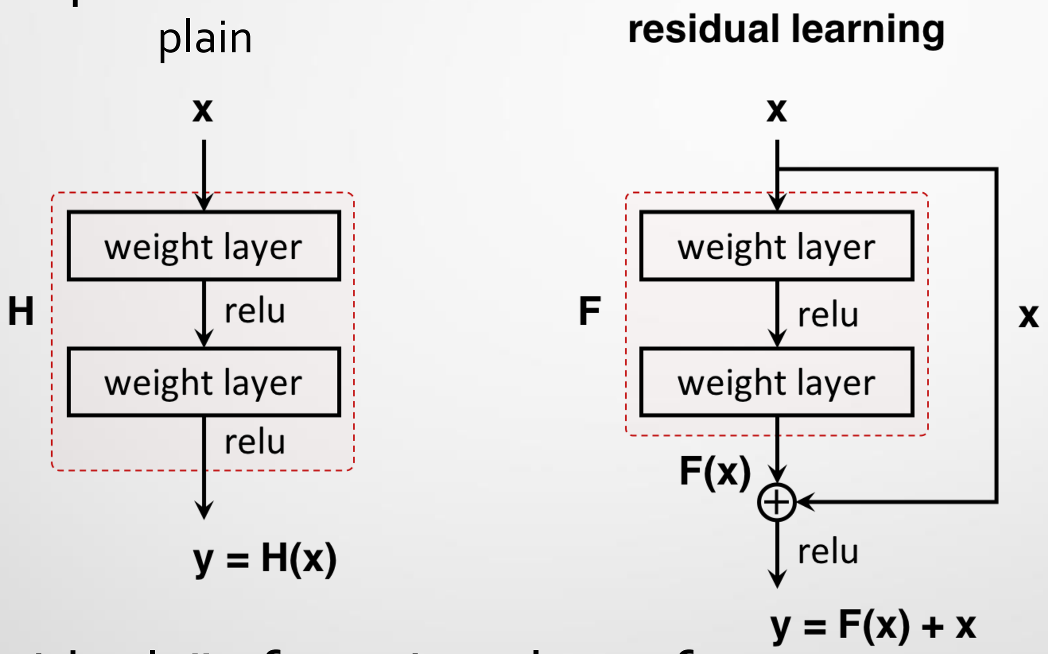
  - Filters are parameters: CNN can obtain them automatically
- After the convolutional operation, apply non-linear transform

  $$z_{kl} = \sigma(y_{kl})$$

  - “Non-linear” is important to get good expression
- Stack these operations

Basics: Residual convolution

- Stream is divided into 2 paths:
  - Path with convolution
  - Path without any operation
- Sum up these 2 path in downstream



- Can learn "Residuals" of previous layer features
- Can construct very deep network
  - >100 layers can be constructed
  - Deeper will be better performance

Basics: Transposed convolution

- Reverse operation of convolution
  - After adding padding, do convolution
  - Use for upsampling