Construct Deep Jet Clustering

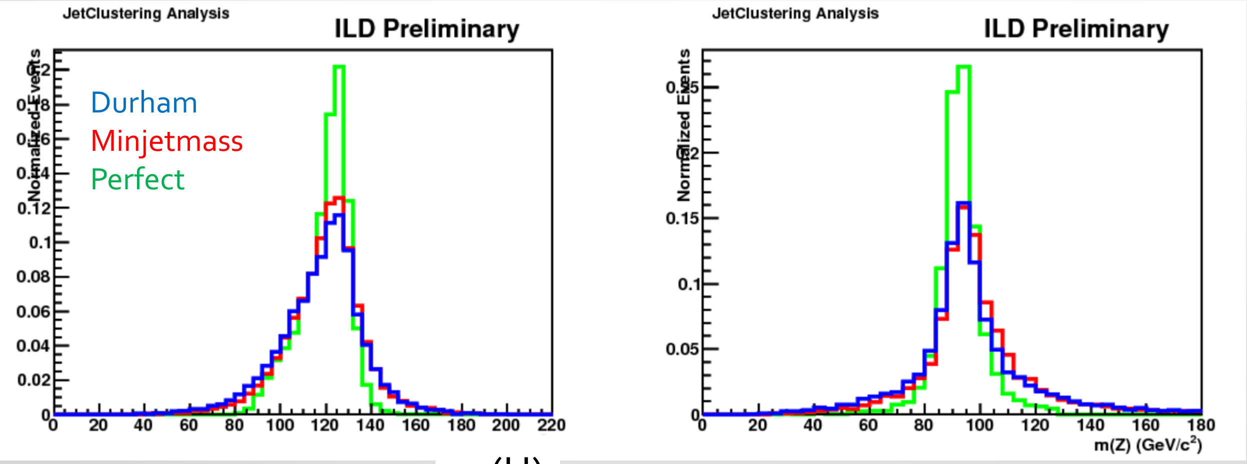
Masakazu Kurata

General Physics Meeting

11/11/2020

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!

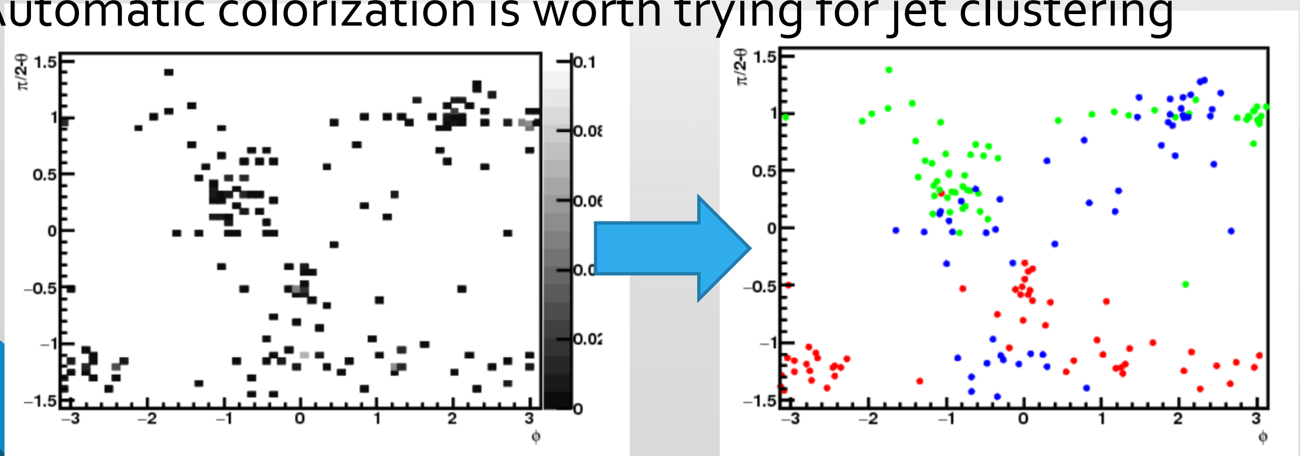


m(H)

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

Use CNN for automatic colorization

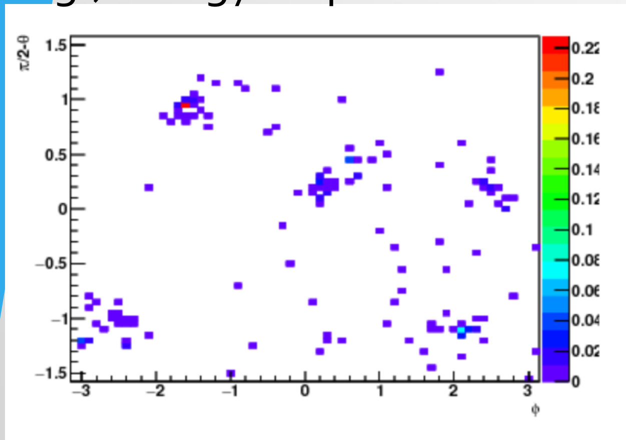
- For jet clustering, we need the global and local information for each event
  - Global: Where is the large energy located?
  - Local: Correlation between neighbors or large energy area?
- Using Convolutional Neural Network(CNN), we will extract both features
  - Encorder-Decorder type CNN is used (calls 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



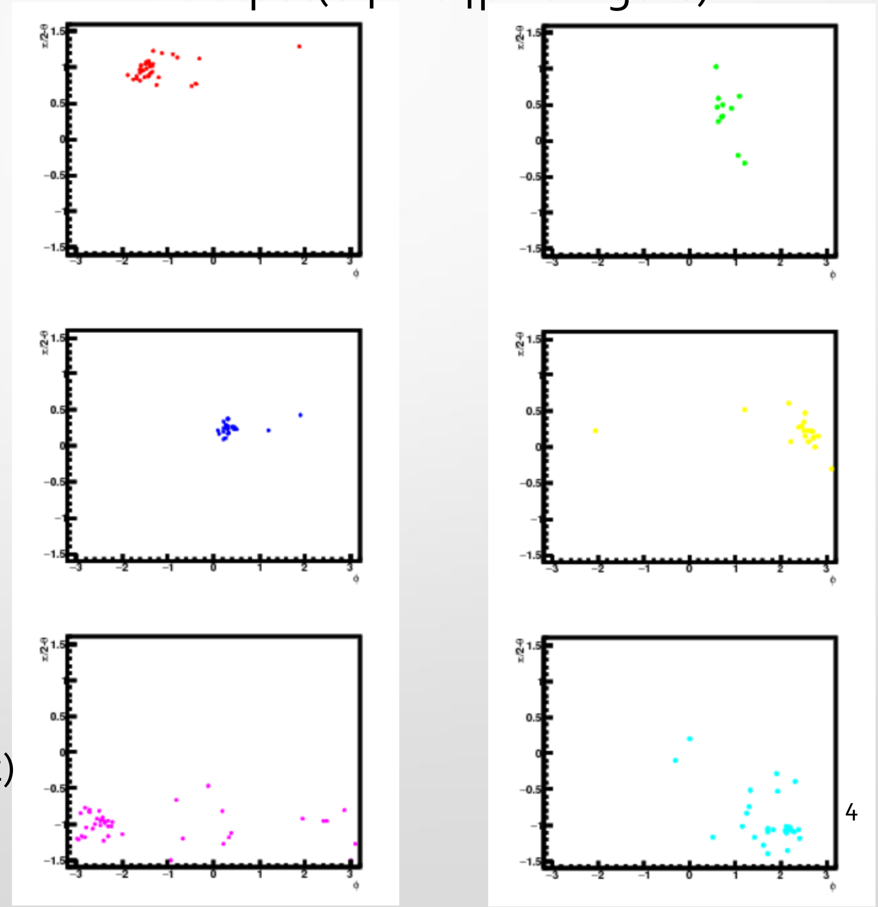
Trial

- Use Keras & tensorflow backend
- 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



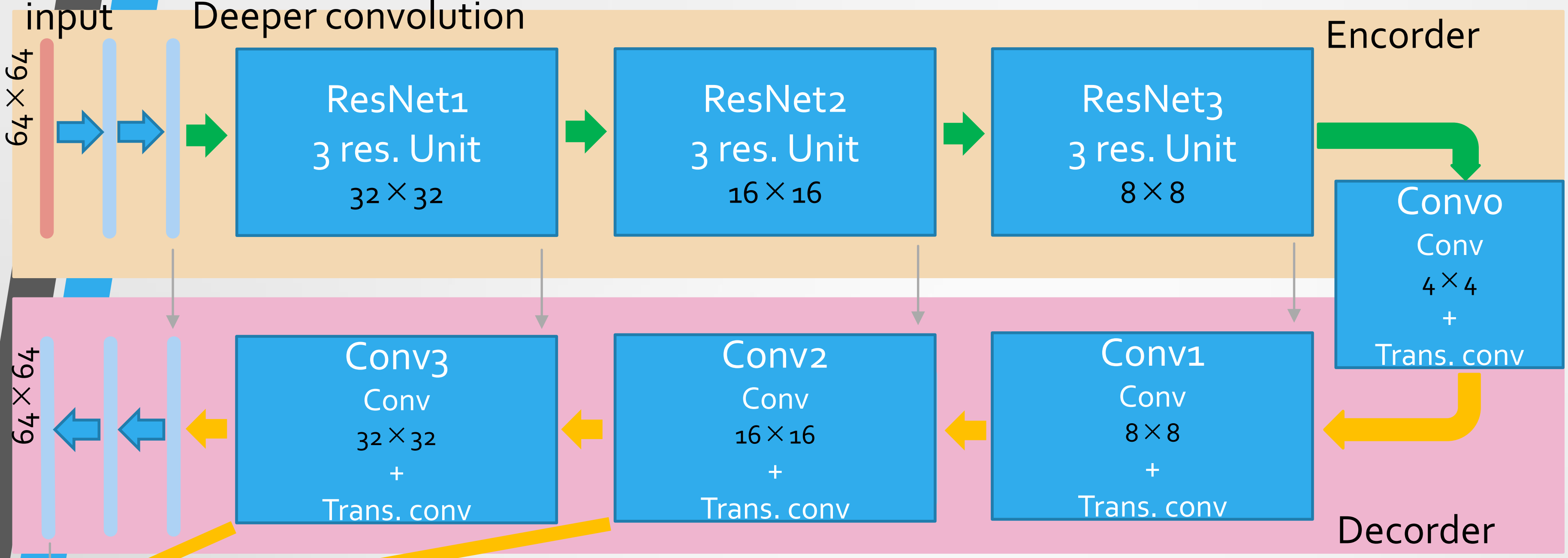
Output(64×64pixel figure)

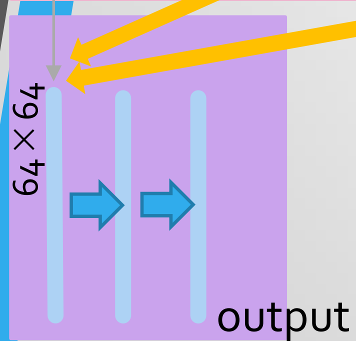


9 input images
Energy map
Charge map
Dosig map + direction vector(x, y, z)
Zosig map
Ecal map
Hcal map

+ no particle

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

Create answer

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

ZHH

Z → 2jets using Durham

H → 2jets using Durham

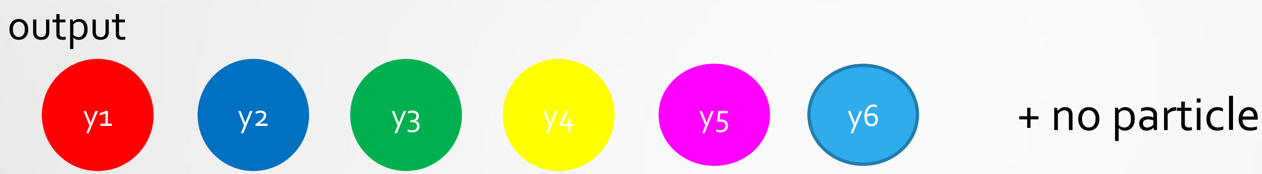
H → 2jets using Durham

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

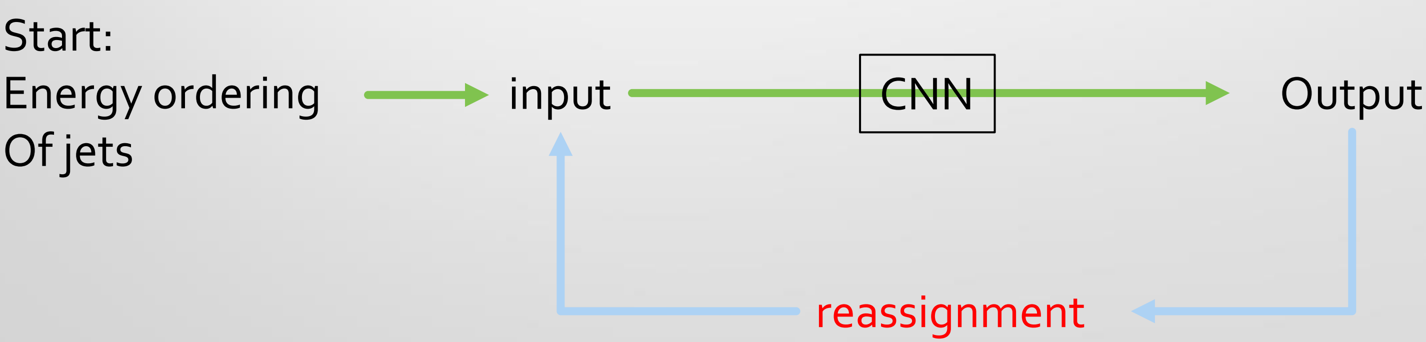
ZHH→(qq)(bb)(bb)→6jets

Pseudo-labelling

- Output: inference of the probability of the color to be assigned
  - $\sum y_i = 1.0$



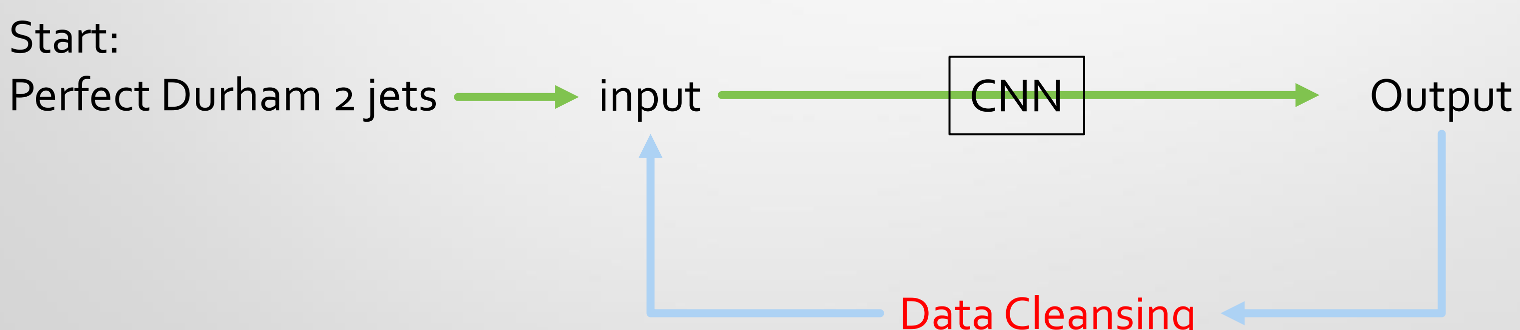
- The combination of color assignments is arbitrary, so assign them so that the loss function is minimized.
  - Using preliminary results after a training, re-assign the color combination
  - Minimize cross entropy $L = \frac{1}{n}\sum y_i \log p_j$



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



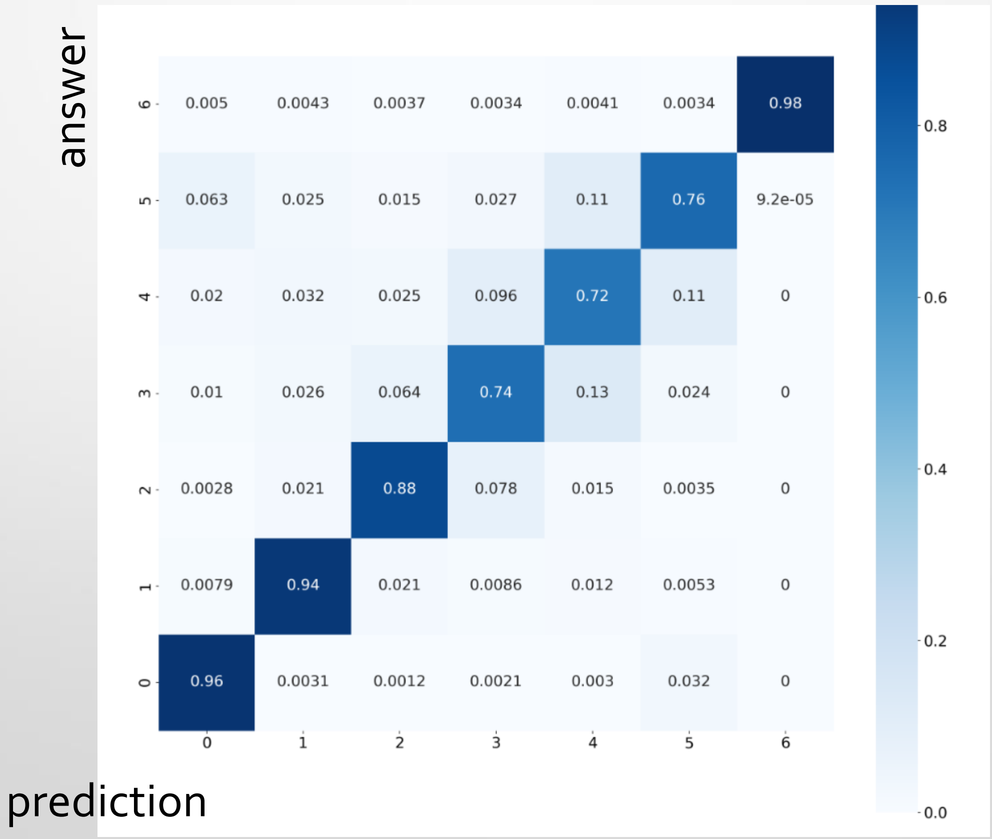
- Clustering particles to make loss function minimum
- First pseudo-labeling. After that, data cleansing
  - Is there better way?

status

- Use ZHH→(qq)(bb)(bb): 6jets clustering
  - q: uds + c + b   ratio: 9:9:5   need to consider the effect of flavor ratio
- Use 230000 events for training(207000 train, 23000 validation)
  - Very weak or no over fitting can be seen between train vs. validation
- Don't consider color singlet state for network training
- Input: 6 + 3 images   output: 6 + 1 images

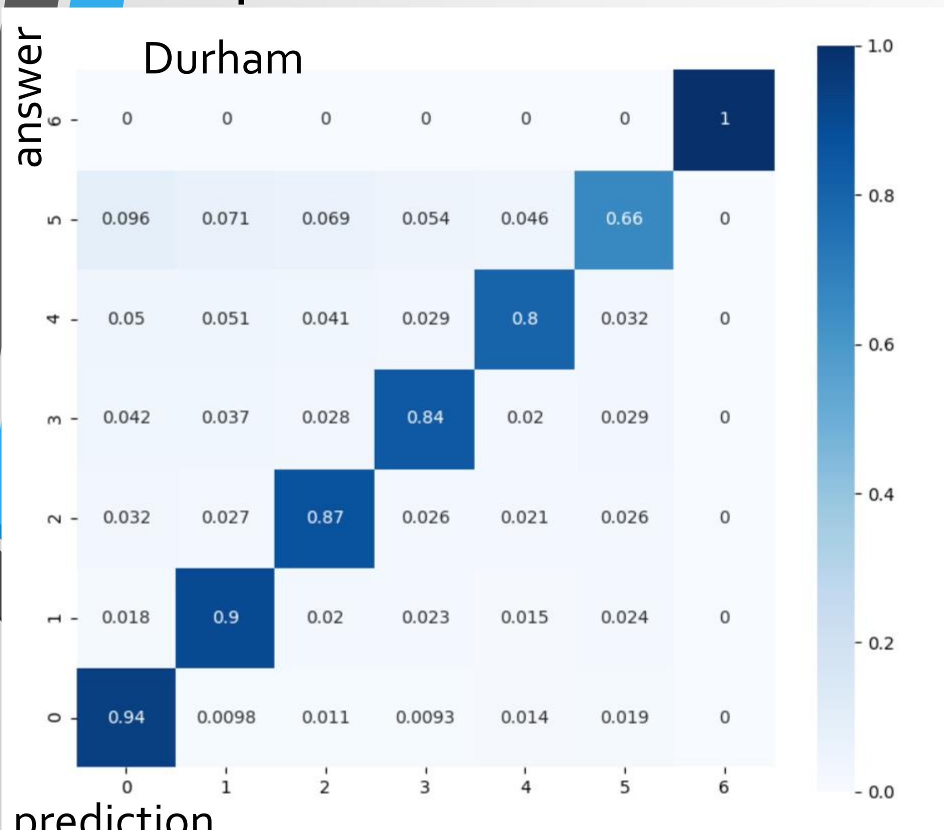
Preliminary results

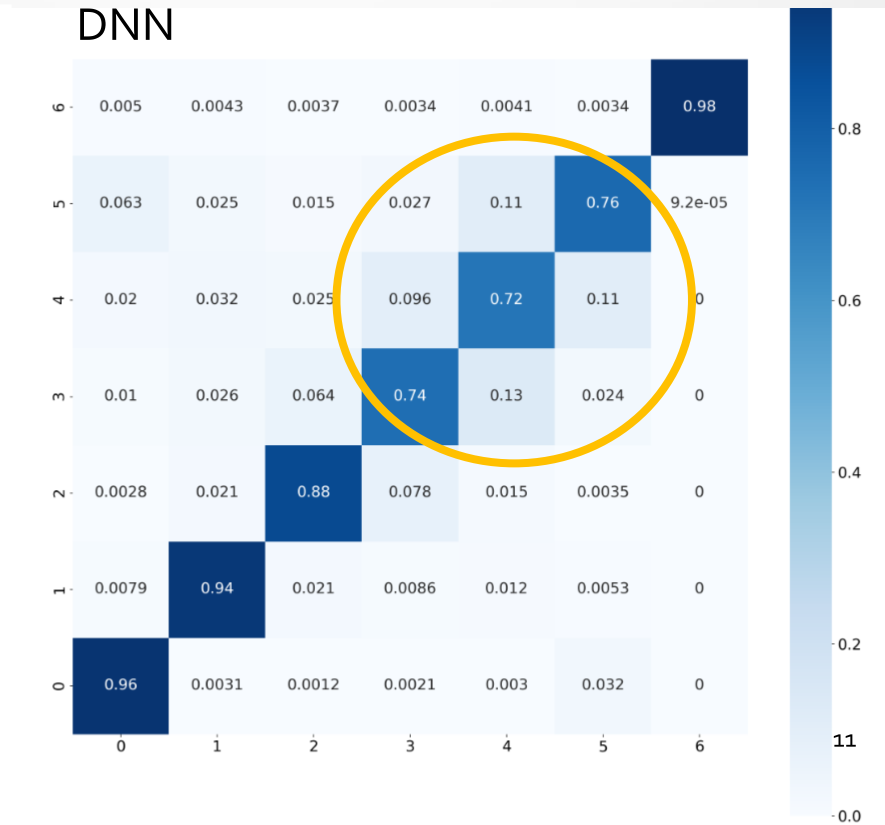
- Confusion matrix
  - 1000 events
  - Num. 0-5: jet number   6: no particle
  - Off-diagonal means mis-assignment
  - Some particles locate at adjacent jets of their answer jet: should be reduce this correlation
  - Now under investigation
- Need to check resolution of Higgs mass
  - Need to improve



Comparison with Durham

- First 3 jets: start to improve
- Last 3 jets: a bit worse
  - Worst efficiency is improved, but need to improve more
- Correlation between adjacent jets, and asymmetric
- Need to improve circle efficiency to get better performance





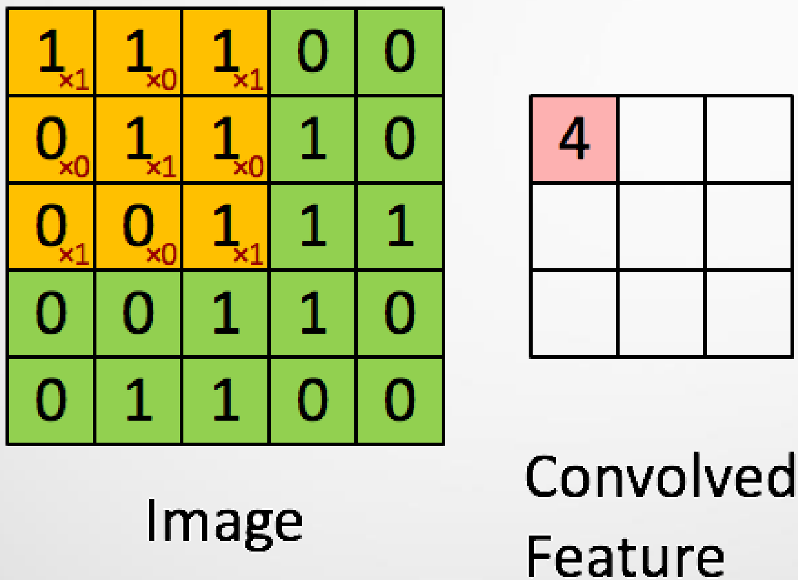
backups

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



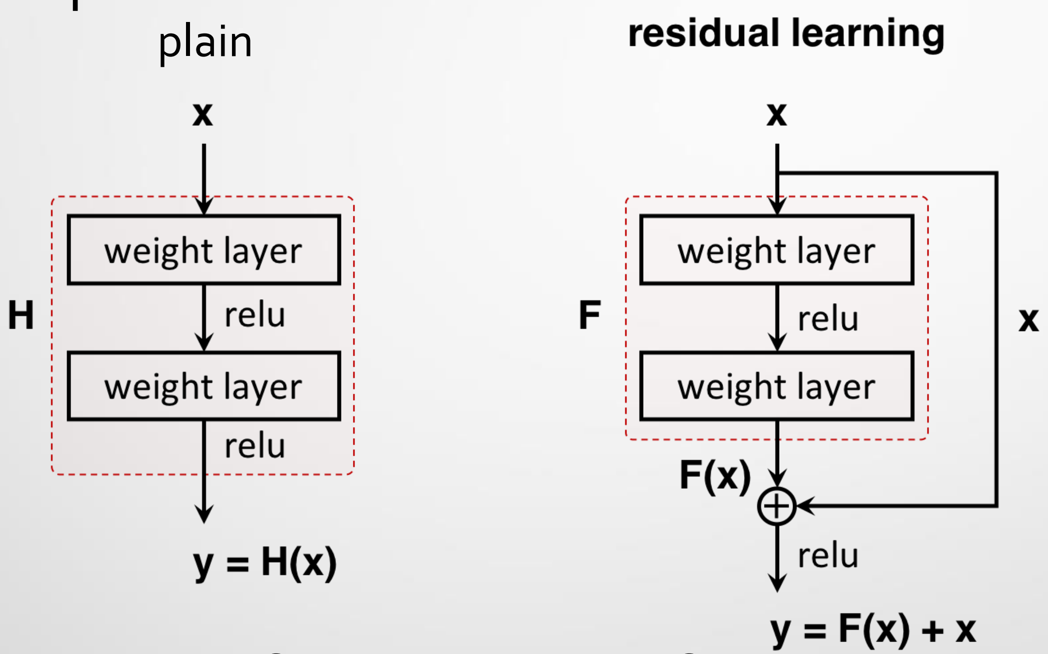
  - Filters are parameters: CNN can obtain them automatically
- After the convolutional operation, apply non-linear transform

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

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

Basics: Residual convolution

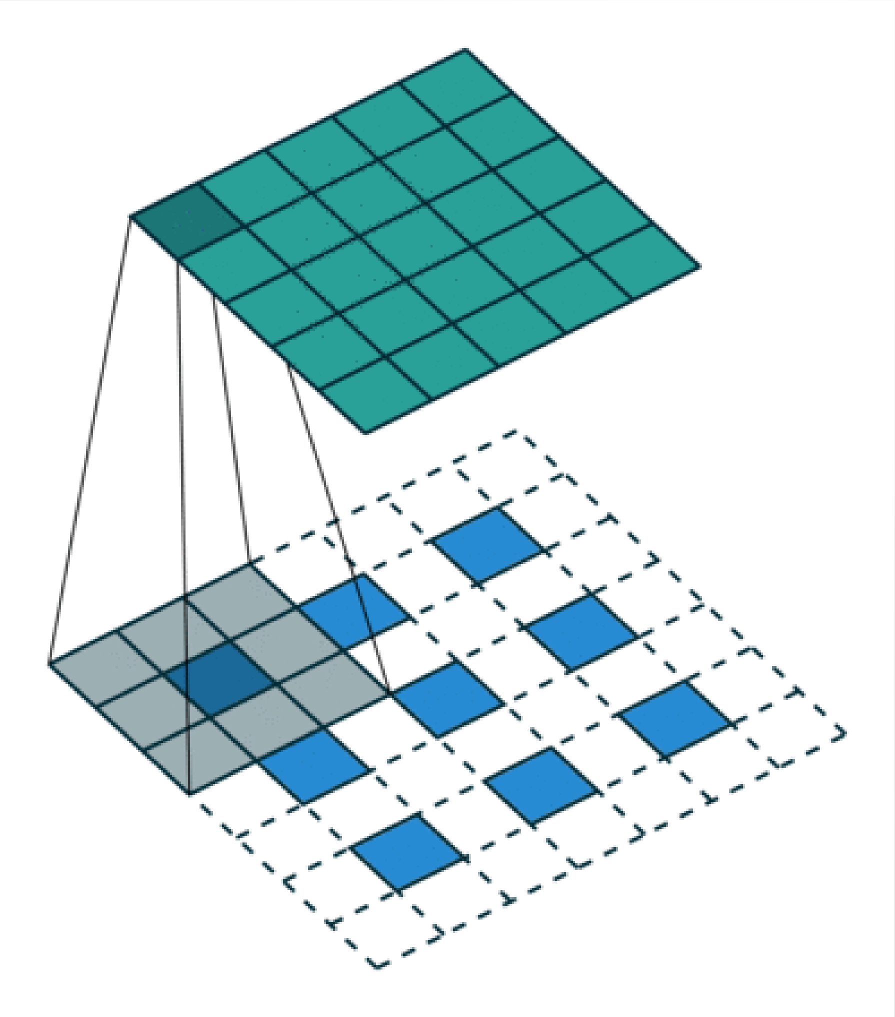
- Stream is divided into 2 paths:
  - Path with convolution
  - Path without any operation
- Sum up these 2 path in downstream



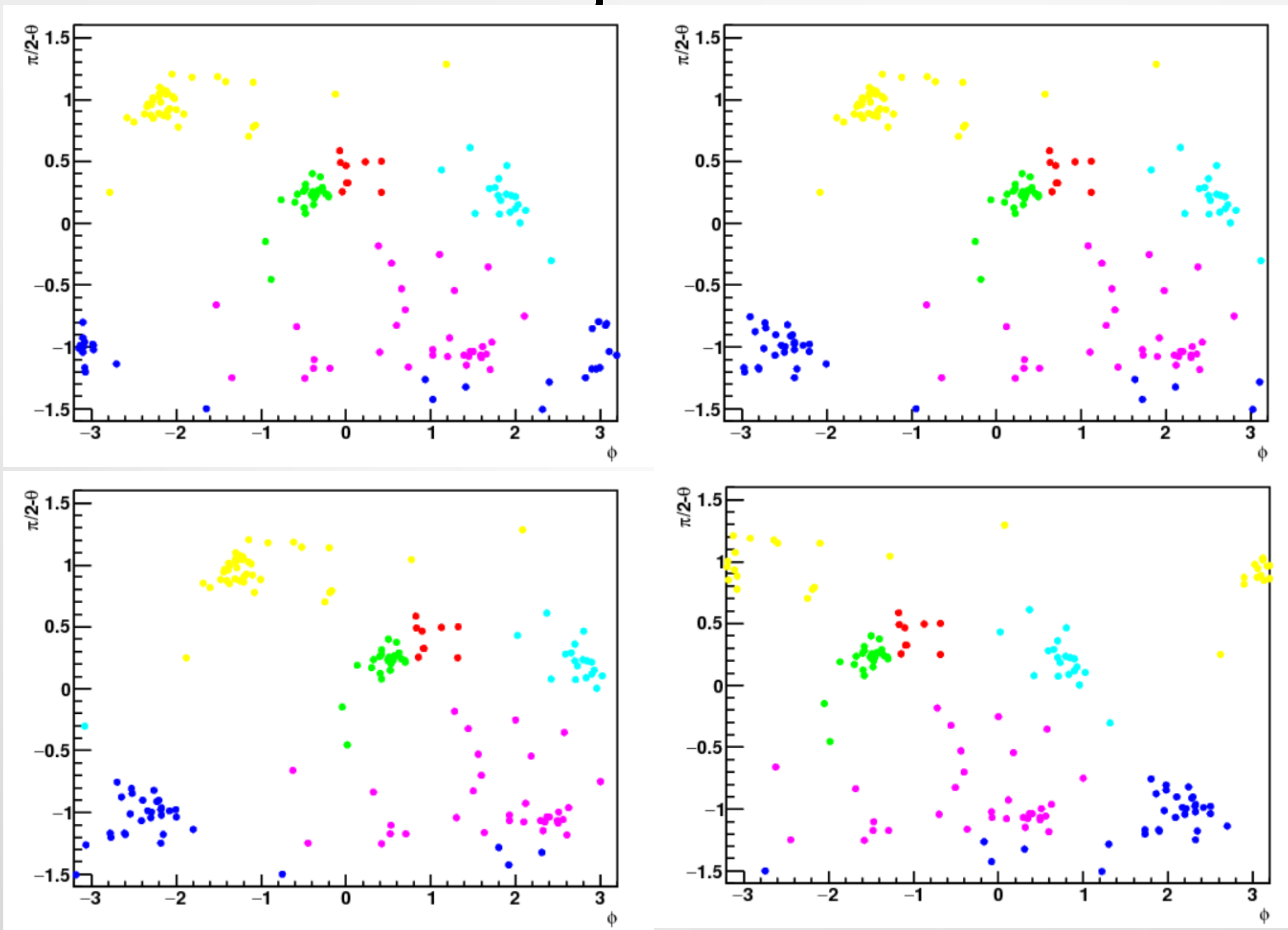
- Can learn "Residuals" of previous layer features
- Can construct very deep network
  - >100 layers can be constructed
  - Deeper will be better performance

Basics: Transposed convolution

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



Data Augmentation



- Random shift for x axis
  - Considering periodic condition of ϕ angle (f(Φ+2π) = f(Φ))
  - To suppress over fitting
- Add random y-flip (I think not good from physics point of view, but suppress over-fitting is important)