# Prospects of machine learning studies for event reconstruction

T. Suehara (Kyushu U.)

# Disclaimer

- This is just to show prospects.
  So no new results will be shown.
  New results will (hopefully) come from the next meeting.

# Topics

- Deep learning: introduction
- Applications
  - Timing reconstruction
  - Vertex finder update
  - Flavor tagging

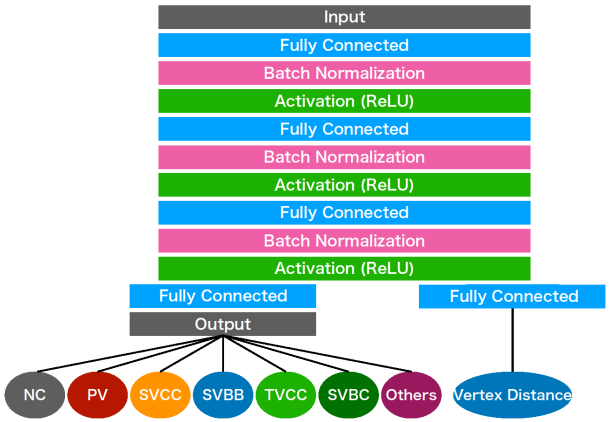
# Introduction of deep learning

- Deep learning is featuring (compared to classical MVA)
  - Many (usually low-level) inputs eg.
    - Pictures, Languages
    - >1000 of input variables possible

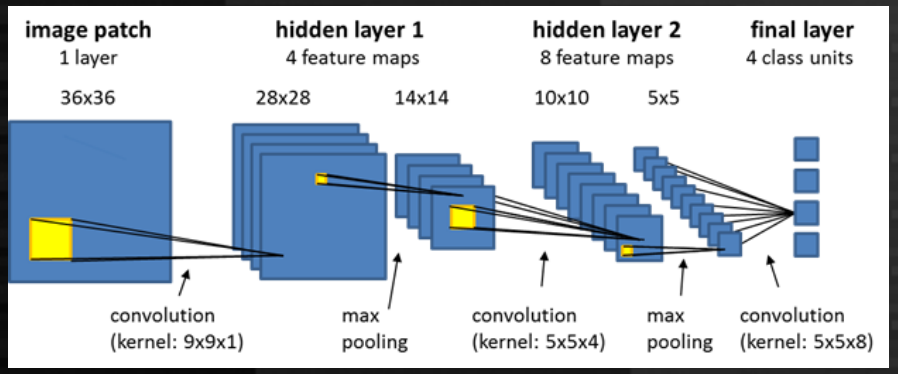
  → Calculation of characteristic values is generally not needed (or favored) any more to avoid loss of information

  - Complicated network structure
    - Fully-connected, convolutional, recurrent, etc.

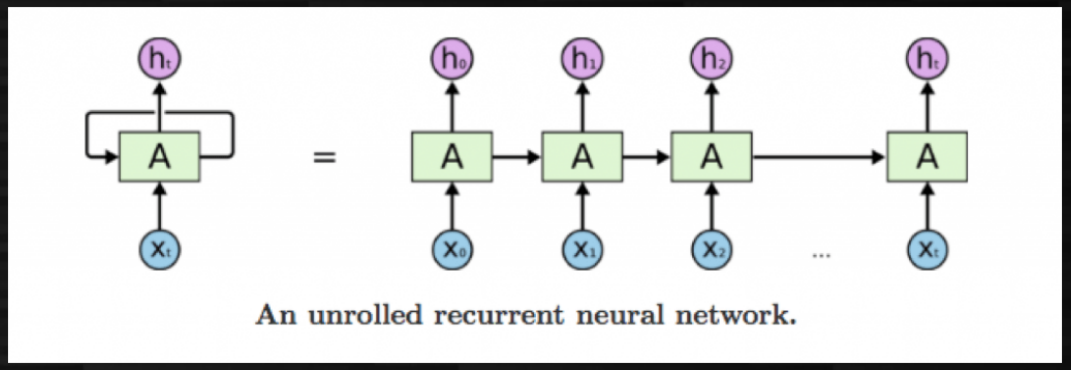
# Network structures



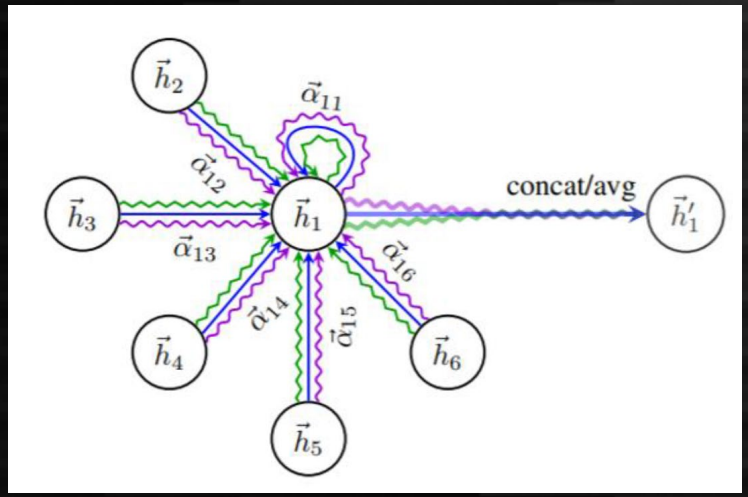

Normal (fully-connected) neutral network




Convolutional Neural Network (CNN)



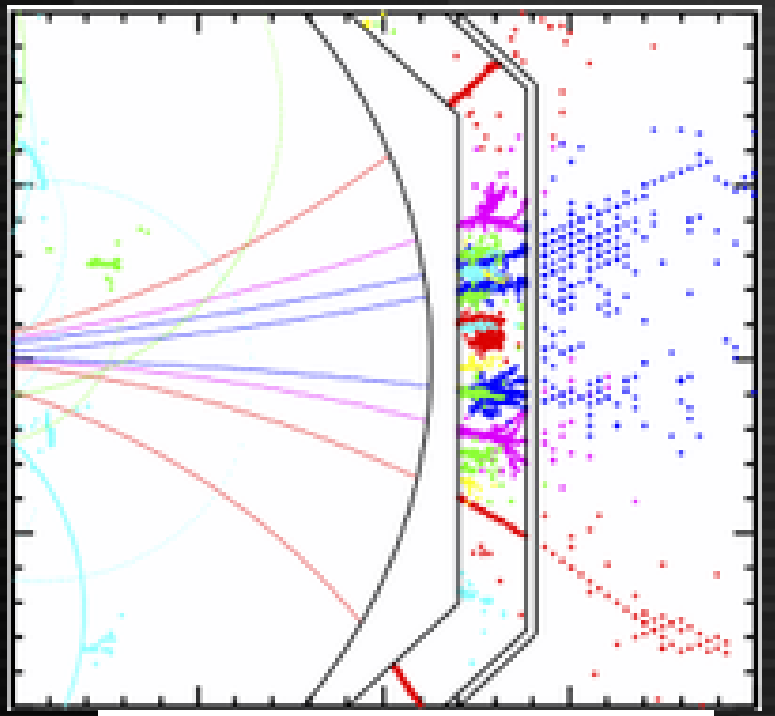

Recurrent Neural Network (RNN)
Many variants like
LSTM, Attention, Transformer

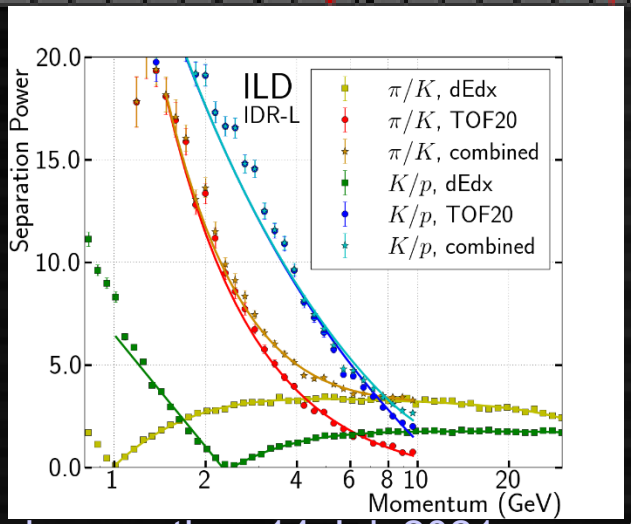



Graph Neural Network
“Nodes + Edges”
GCN or GAN popular

# Application 1: timing reconstruction

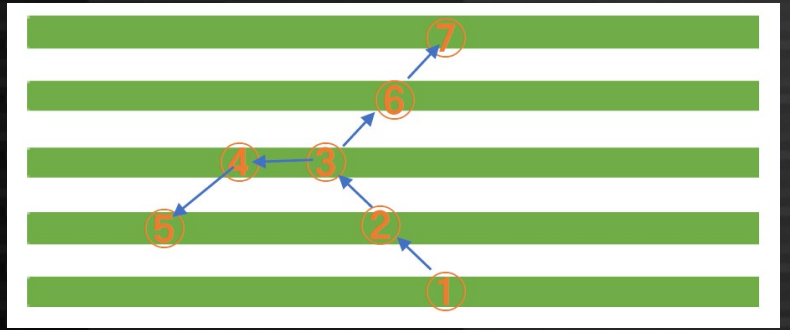
- Time-of-Flight (ToF) is a powerful tool for hadron ID ($\pi$/K/p separation)
  - < 20 psec required
- At calorimeters, hits can be averaged to improve timing resolution of the sensors
- Hadrons @ ECAL
  - Track-like: easy to average
  - Track + 2ndaries
    - Have to identify path inside CAL
  - Showering
    - Separate usable/unusable hits

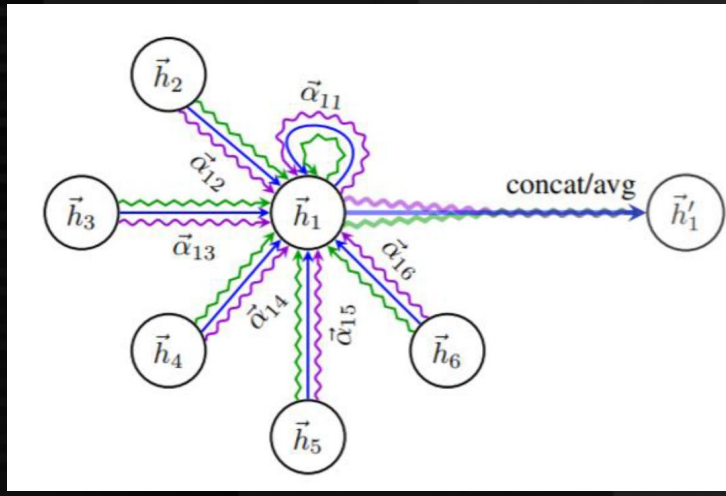




# Timing by Graph Attention Network

- Input variables
  - Position, timing (smeared), energy deposit of each hit
- Output candidates
  - Ordering of hits (parent hit)
  - Averaged time at surface
- Structure
  - Graph attention network
    - Optimize “connection strength” between nodes (hits)
  - Supervising connection (attention weights) or nodes after processing

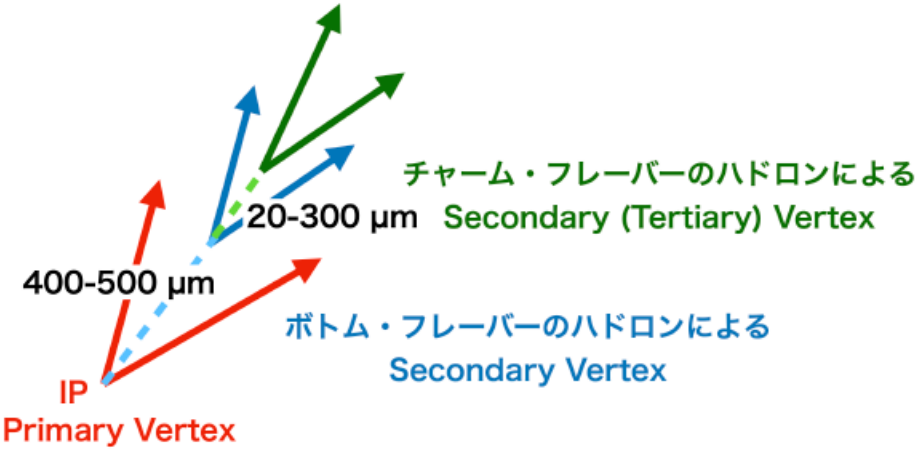


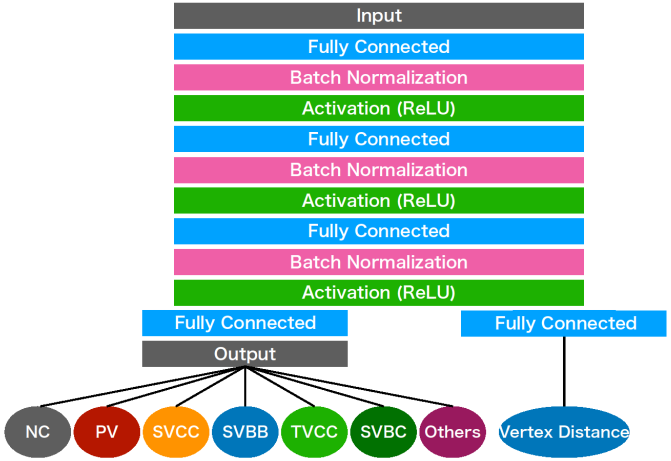



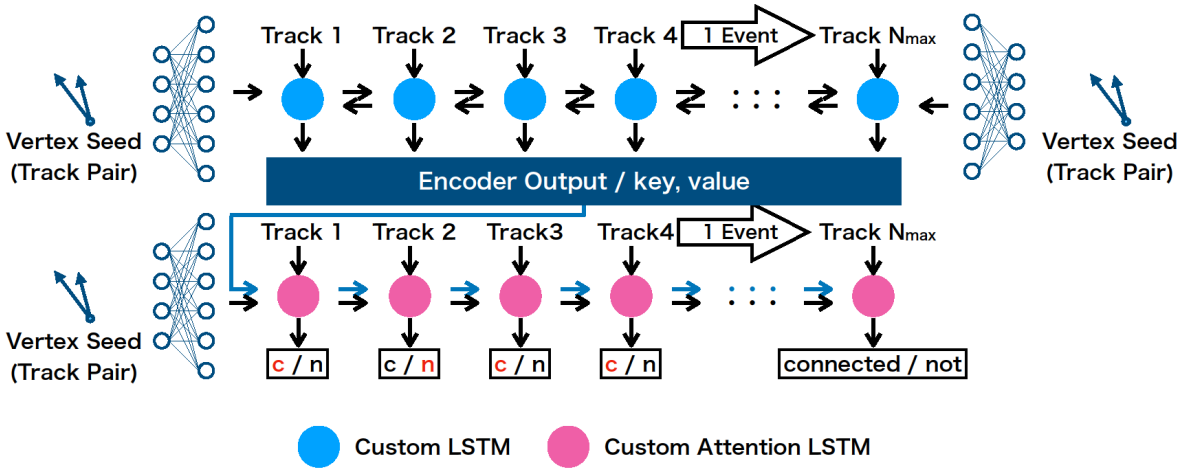
# Application 2: vertex finding

- Used as an input to flavor tagging
  - Current implementation inside LCFIPlus
- First implementation by K. Goto last year
  - Using same strategy as LCFIPlus
    1. Classifying track pairs as “primary”, “secondary” or “not connected”
    2. Try to attach tracks to “primary”/”secondary” pairs
       - Using a kind of “recurrent neural network”
  - Performance similar to LCFIPlus
    - High efficiency but more contamination
  - Need to be improved

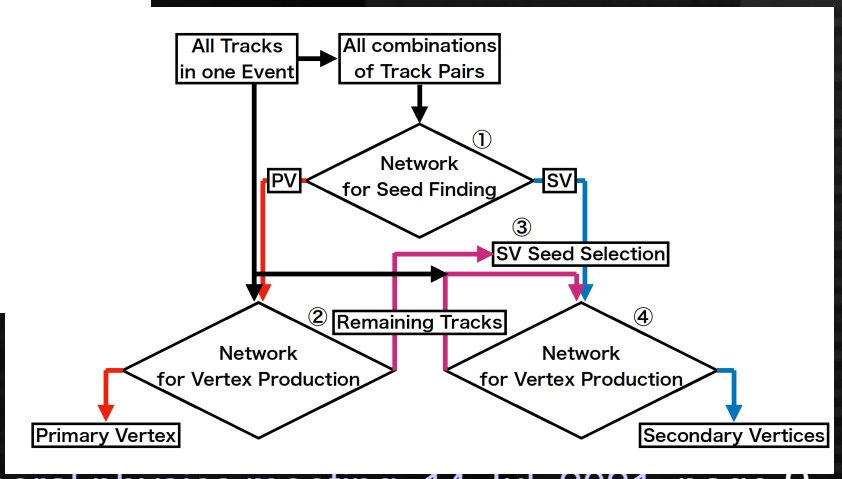
# Vertex Finder by K. Goto




## Seed finding network



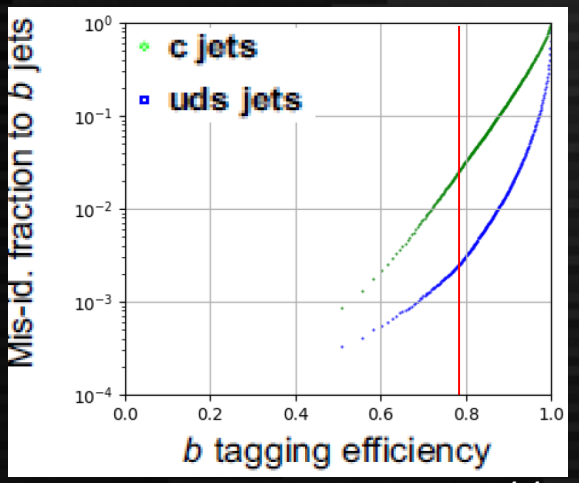

## Vertex production network

# Vertex finder: to be studied

- Paring network
  - Basically geometrical calculation to obtain crossing points of tracks
    - Not an easy job for ML (precision problem)
  - Design network to support finding the crossing points
    - T-parameter notation of the track parameters
    - Matrix of t1-t2 to easily find crossing points
- Vertex production network
  - Try different network (eg. transformer or graph attention)
  - Consider how to pass information to flavor tagging
    - Minimizing information loss
- Combination of above two in network level
  - No practical design yet…

# Application 3: flavor tagging

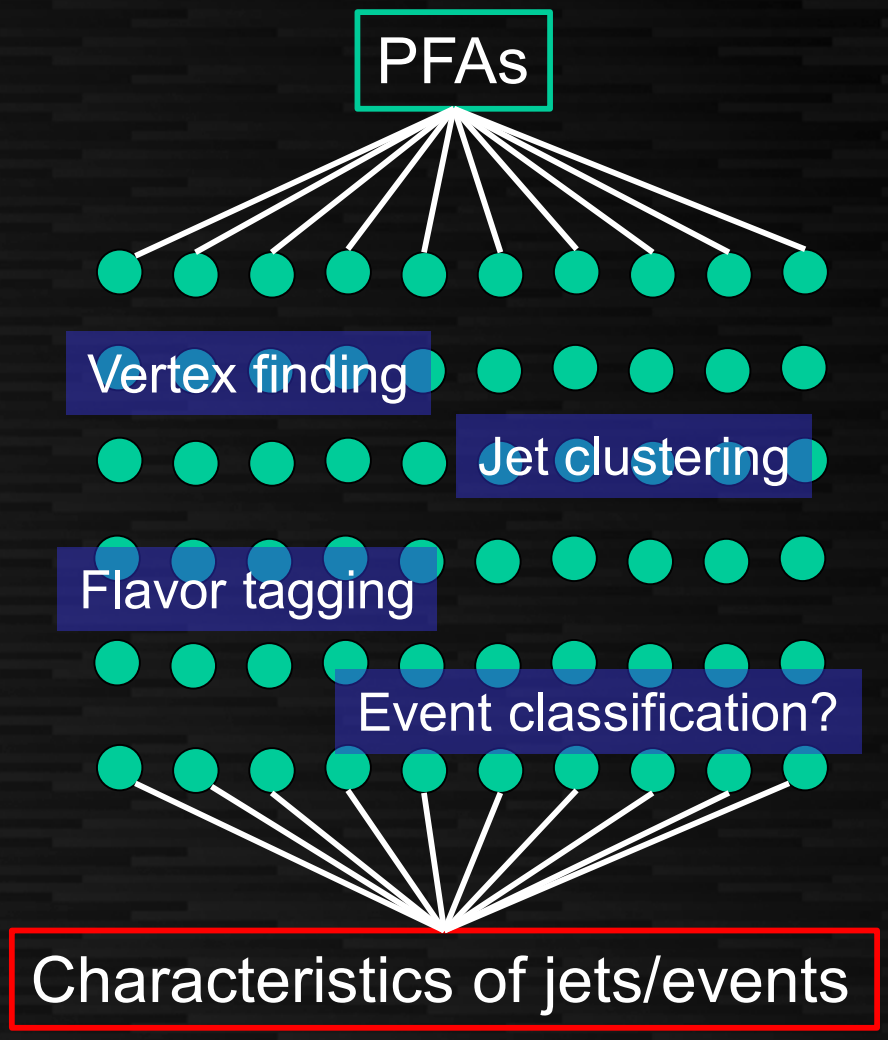
- LCFIPlus uses boosted decision trees (BDT)
  - Using vertices reconstructed in earlier stage
  - Categorization by number of vertices
    - 0, 1, 1+1 (pseudo-vtx), 2
  - 10-30 inputs on each category
  - Output: b-, c-, q- likeness
- Some DNN implementation for FCC study
  - "ParticleNet"-based, graph convolutional network(?)
- Iwasaki-san's results
  - Adding track parameters (without error matrices)
  - Already shows improvement



# Flavor tagging: to be studied

- Adding low-level variables (tracks)
  - How to feed the network error matrices (with connection to parameters)
  - How to combine to high-level variables (vertices)
- Network structure
  - Graph-based? (not well considered yet)
- Check event dependence
  - LCFIPlus variables are carefully selected as quasi boost-invariant
  - Probably we should train with wide category of events
- More outputs
  - Vertex charge, b/c/s/g/ud separtaion?
  - Variables for s/g characterization?

# Future target: total jet reco network?




- "Transfer learning"
  Using pre-trained partial network to solve bigger problems
- How to connect individual networks?
  - Or transferring abstract information to later?
- How to train individual networks?
- How to retrain full network?

- Full event reconstruction by ML (GPU/TPU/FPGA based computing farm for ILC?)