Machine Learning applications for Future e^+e^- Colliders

Jason Lee
University of Seoul

KPS Fall Meeting 2020

Introduction

- Machine Learning doesn't need an intro anymore
- Already used in pretty much all classification problems
  - Most easy gains already gained
- Current tends -> efficient use of resources
  - simulation
  - permutations - track finding
  - triggering
- Wide applications
  - creative uses
  - field has exploded: hard to cover much in 20 mins

- Theory
- Accelerator
- Detector Monitoring
- Triggering
- Detector Simulation
- Reconstruction
- Identification
- Physics Analysis
- Computing

Recent HEP ML Workshop

4th Inter-experiment Machine Learning Workshop

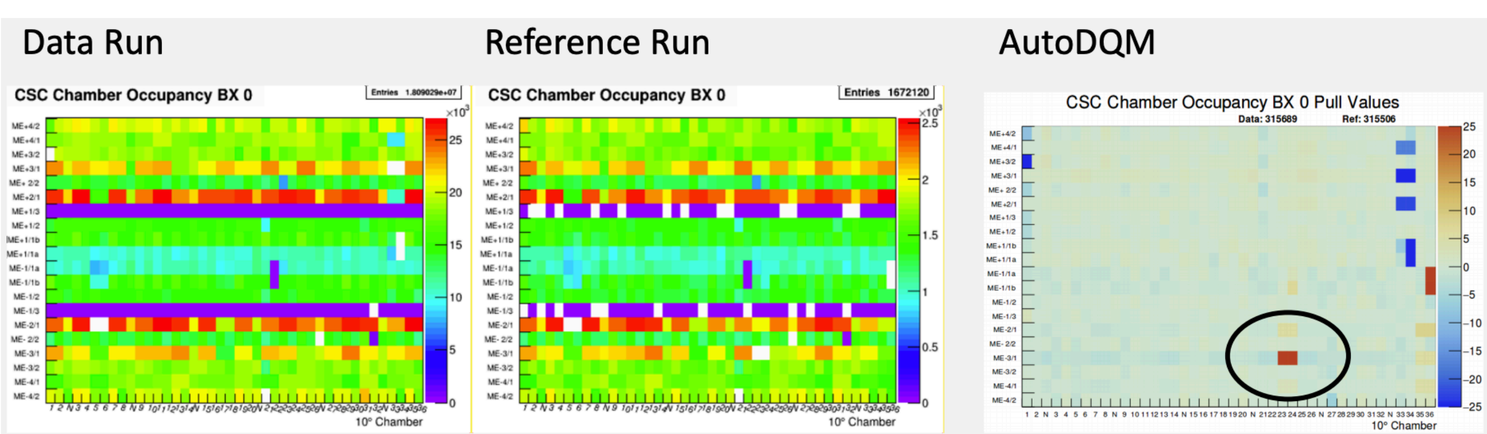
https://indico.cern.ch/event/852553

- 874 registered participants
- 88 contributions (tutorials, plenary, workshop)
- Lots of collaborations with industry (deepmind, facebook) and other disciplines
- 15 talks on Graph Neural Networks (GNNs)
  - ideal for structured data
- 11 talks on Generative Adversarial Networks (GANs)
- wide range of talks
  - lots of new ways to solve problems
  - use of machine learning as a scientific advisor

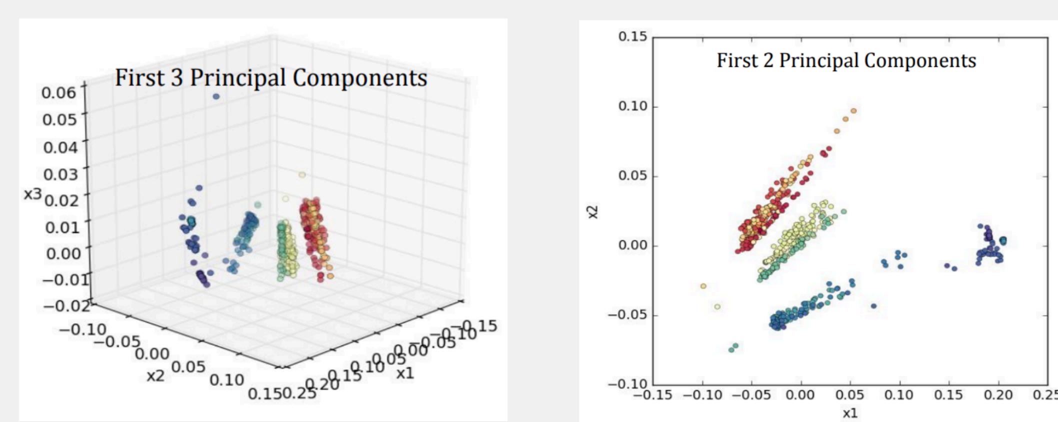
Detector Monitoring

https://indico.cern.ch/event/852553/contributions/4057666/attachments/2127765/3582662/AutoDQM_IML.pdf

- ➡ Data Quality Monitoring (DQM) shifters monitor detector performance during data taking to certify good "data"
  - ➡ time consuming and prone to human error
- ➡ AutoDQM created to semi-automate data certification based on statistical tests and ML techniques
  - ➡ 1D histograms: Kolmogorov-Smirnov Test
  - ➡ 2D histograms: Pull Values/Chi-Squared Test
  - ➡ all plots: Principal Component Analysis
    - ➡ bad runs are not described well by the principal components
  - ➡ Looking into Autoencoders, Clustering algorithms (DBSCAN, k-means)
  - ➡ CMS implementations: muon dets (CSC, DT, RPCs), ECAL, Muon and Calo triggers
- ➡ Future
  - ➡ Full detector monitoring including conditions, using other sub-detectors for reference



Color represents run number (i.e. time). Clusters correspond to different groups of runs with differences in configuration



X_1 corresponds to instantaneous luminosity

Triggering

- At reconstruction level machine learning is not yet implemented.
  - while in identifying and classifying, machine learning is very widely used, b-tagging, quark tagging, egamma id, tau id
- At trigger level, CMS uses BDTs at L1 for muon pt assignment, soon to be updated with NN.
- ML is very interesting for triggering since inference is fast and faster with GPUs or FPGAs

Muon triggering

❑ Motivation

- Trigger decision can be treated as image segmentation problem in Computer Vision, and Convolutional Neural Network(CNN) is a powerful tool in segmentation problem.

- Our plan is to increase trigger accuracy with CNN based segmentation, while reducing latency by FPGA to enable online use.

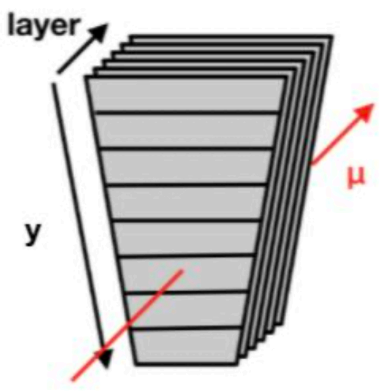


Fig1. Detector we aim for

❑ Dataset Info.

- We aim for generic muon detectors. In this study, the chamber has 6 layers, with layer consisting of 384 strips in the x axis and segmented in 8 partitions in the y axis.

- A chamber image made up of 6*8*384 voxels, and each voxel is binary (0 or 1). When the current flows on a strip, a voxel turned 1 from 0.

- Muons are ranging from p_T 5~100 GeV.

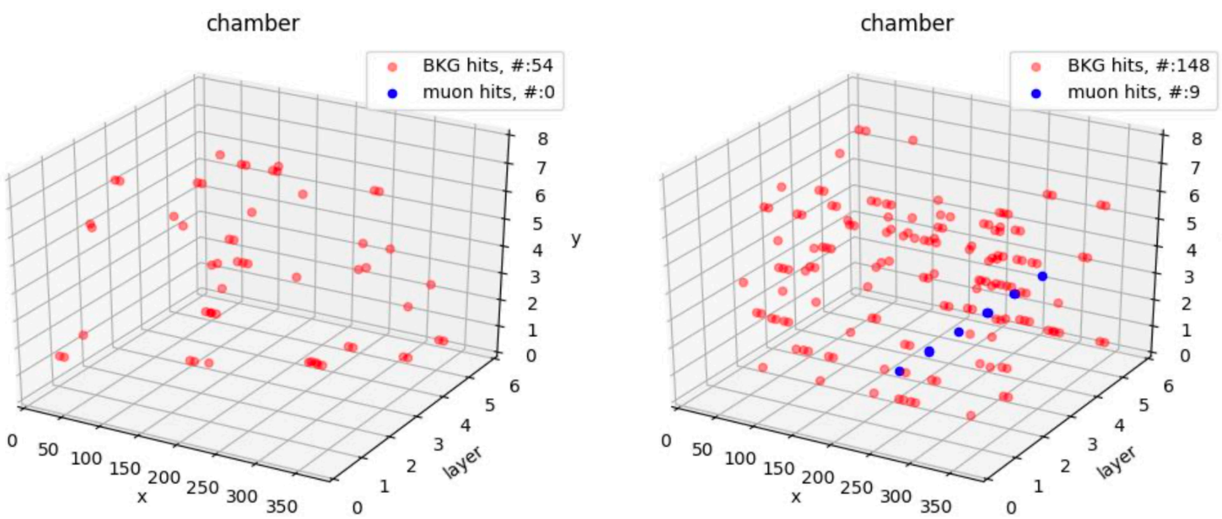


Fig2. Visualization of chambers with hits

Muon triggering

❑ Training CNN

- Convolution kernels extract the visual features of muon hits so that model output close to ground truth.
- Loss - weighted binary cross entropy loss

$$\mathrm{WCE}\left(p,\hat{p}\right) = -\left(\beta p \log\left(\hat{p}\right) + (1-p)\log\left(1-\hat{p}\right)\right)$$

- Keras with tensorflow 1.14 for the backend.

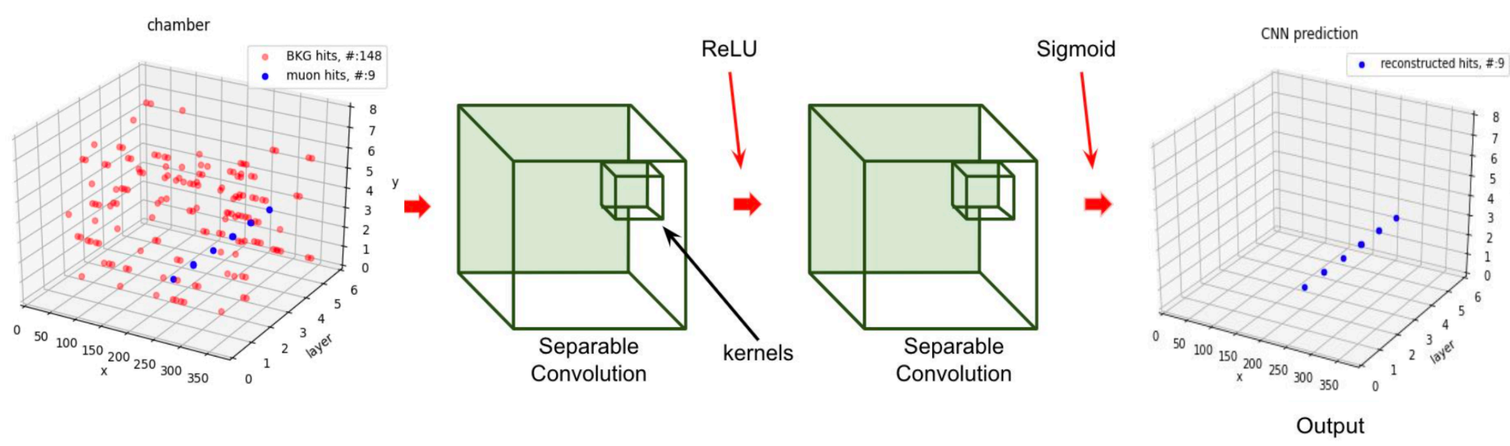


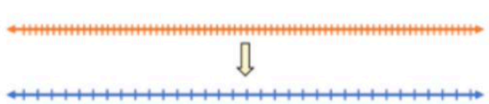
Fig3. CNN based track segment reconstruction

❑ Deploying on FPGA

- Quantization for smaller model size and faster computation.

Quantization

- Changing float 32→int 8 for lighter model



Layer optimization

- Deleting batch normalization layer and sigmoid layer

  - Batch normalization layer is used for training, not inferencing.
  - Sigmoid layer can be replaced by some appropriate threshold.

Compilation & Deploying

- To deploy the CNN in FPGA, we used the compilation tool provided by Vitis-AI (Xilinx AI framework).

Muon triggering - results

❑ Efficiency & Fake rate

- About 300,000 images are used for test.
- If True Positive Rate (TP/P) of a image is over 0.6, we define it as a “matched”.
- In efficiency plot, 0~5GeV is empty, so the number of samples in first bin is half of others.

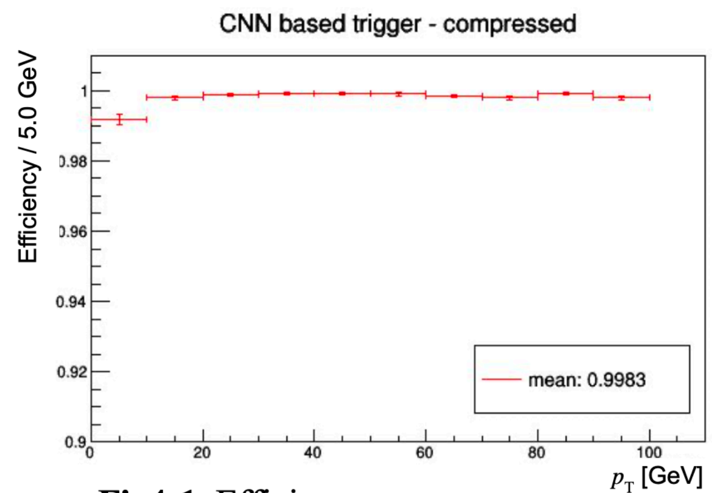


Fig4-1. Efficiency

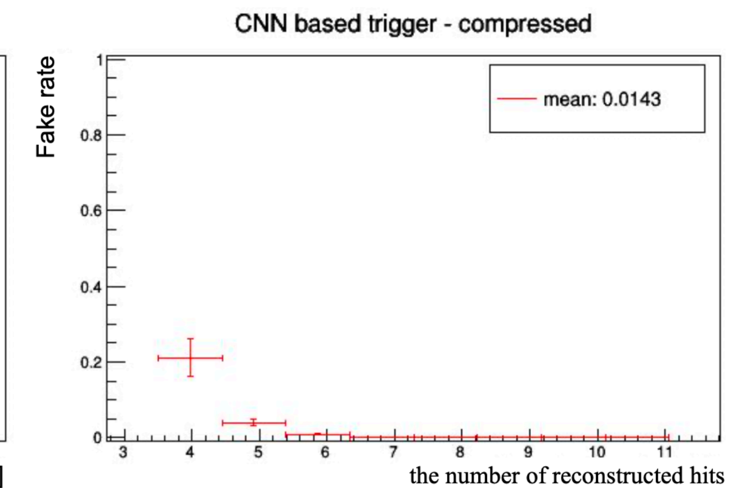


Fig4-2. Fake rate

※ *Definitions of efficiency and fake rate are on postsers.*

❑ Throughput

- Throughput is the rate at which something is processed.
- On Fig5, throughput was measured in processed images per second (FPS), and the buffer sizes are 1, 2, 4, ··· , 2048 from left.
- Only 1~6 buffer size is allowed for FPGA now.

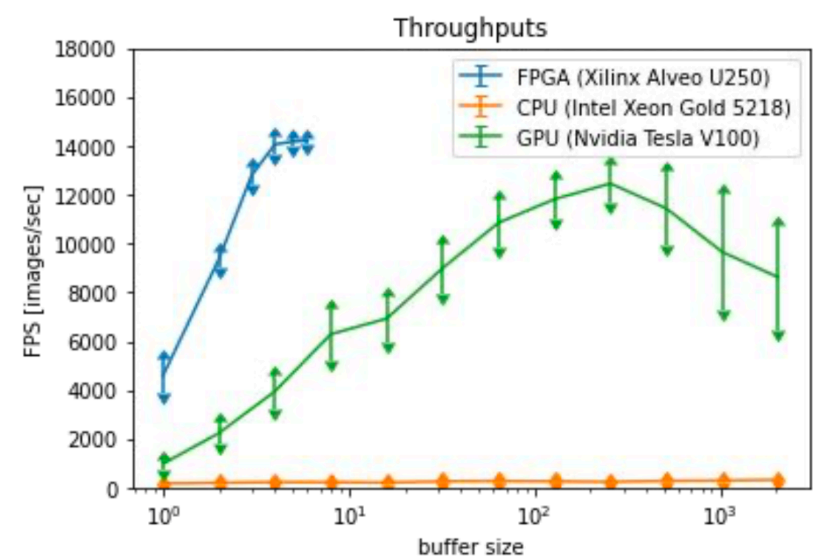


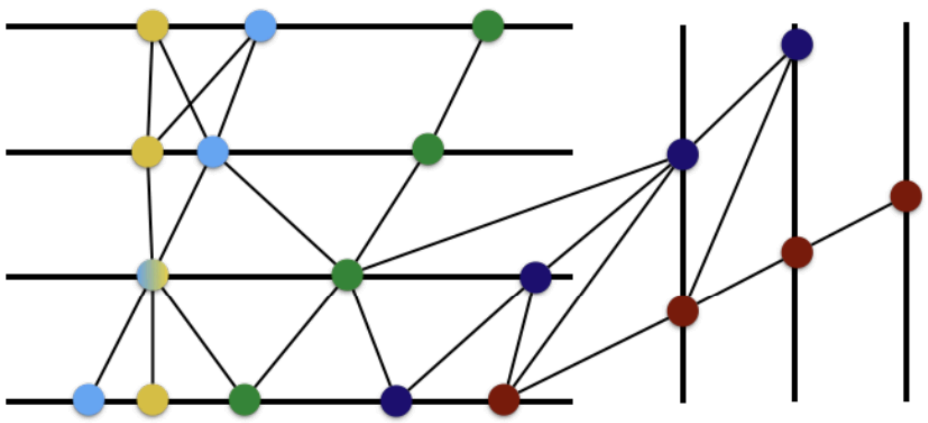
Fig5. Throughput according to processors

➡ shows very promising results.

➡ Almost reco level performance at L1

➡ FPGA not fully optimised for speed yet!

Track building with GNN



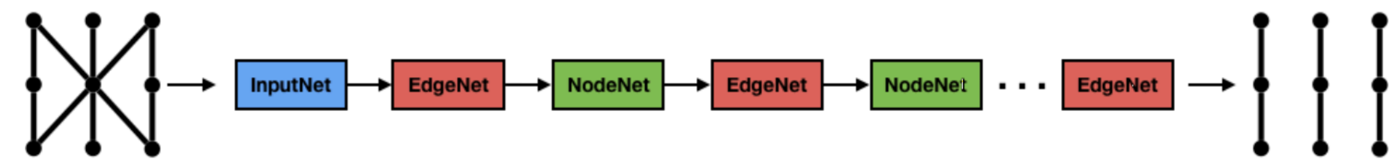
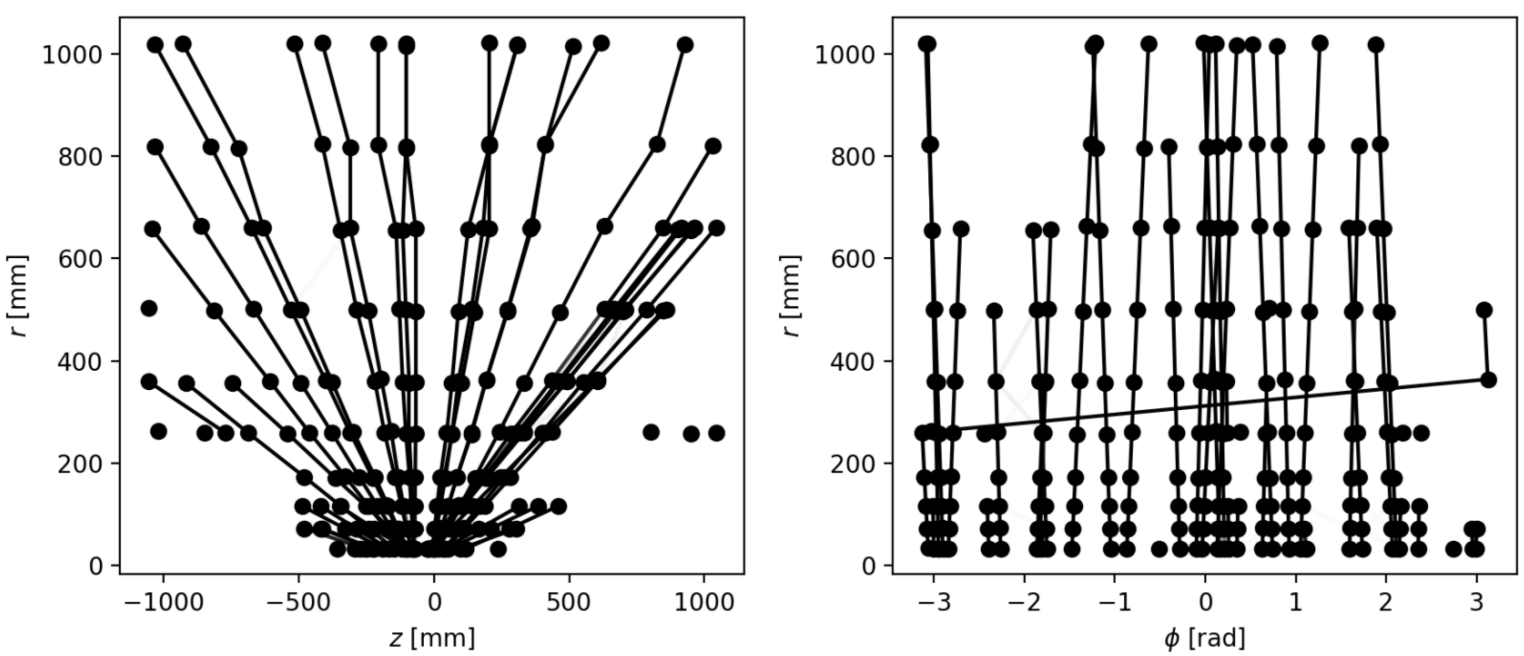


Figure 9. Diagram of the Graph Neural Network model which begins with an input transformation layer and has a number of recurrent iterations of alternating EdgeNetwork and NodeNetwork components. In this case, the final output layer is the EdgeNetwork, making this a segment classifier model.

- ➡ EdgeNetwork computes edge weights using the features of the start and end nodes
- ➡ NodeNetwork computes new features for every node using the edge weight



- ➡ purity 99.2%, efficiency 97.9%, accuracy 99.4%
- ➡ reduce network size (pruning, precision) for implementation at trigger level

https://arxiv.org/abs/1810.06111

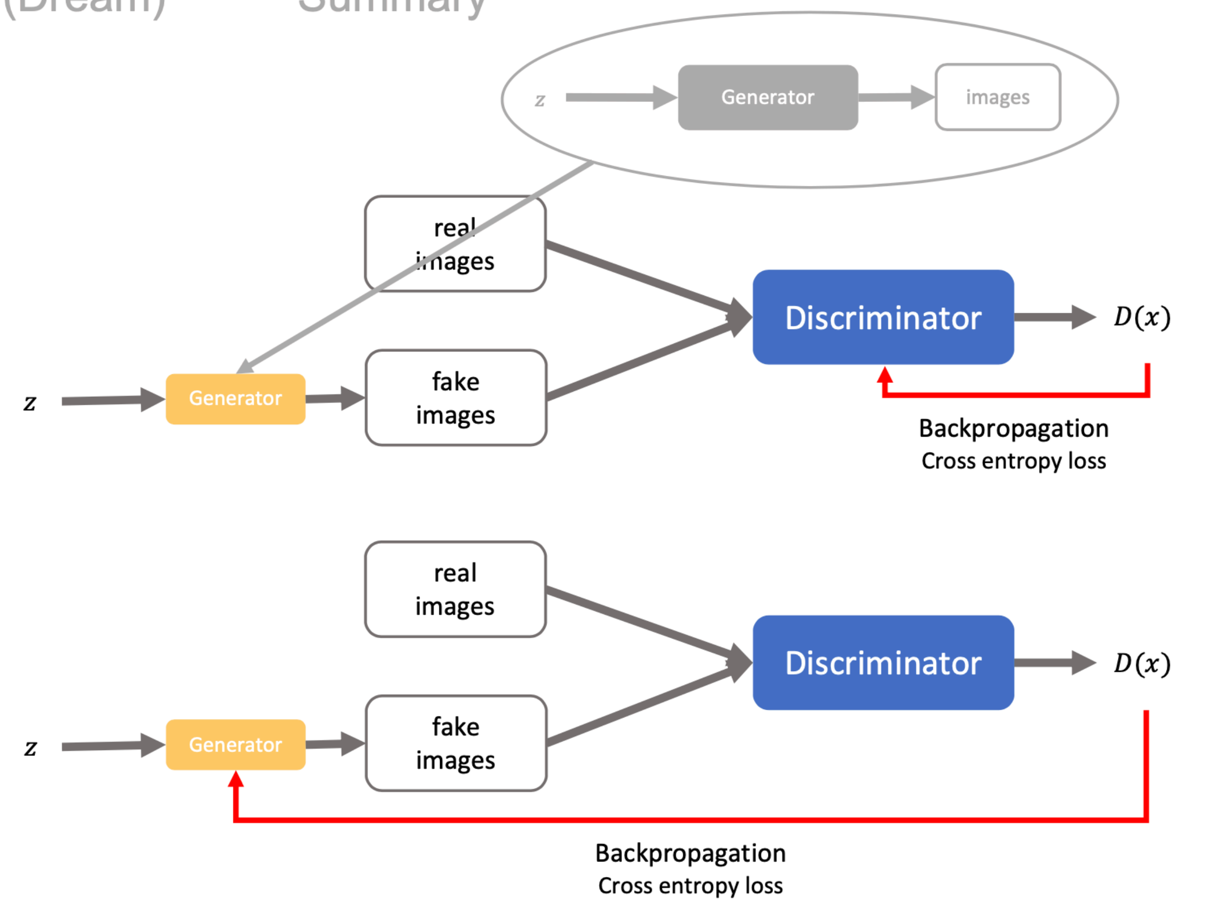
Detector Simulation

- Complex modern detectors take a long time -> calorimeters
  - many ideas have been proposed and in use
    - frozen showers, fast sim, DELPHES etc.
- Generative Adversarial Networks (GANs)
  - Calorimeters - CaloGAN (Paganini et al.)
  - WGAN, DCGAN
- Variational autoencoders
  - Calorimeters - Bounded-Information-Bottleneck autoencoder (Buhmann et al.)

GAN (Generative Adversarial Network)

Generator and Discriminator

- Generator
  - Generator make images from noise. (similar way to VAE)
  - If generator is trained well, we can generate many images similar real images.
- Discriminator Network
  - Discriminator get real images and fake images.
  - If discriminator get real images, $D(x_{real})$ should be 1
  - If discriminator get fake images, $D(x_{fake})$ should be 0
- Adversarial Network
  - If discriminator get real images, $D(x_{real})$ should be 0
  - If discriminator get fake images, $D(x_{fake})$ should be 1
  - Adversarial relationship : discriminatorNets vs AdversarialNets
- minimax algorithm
  - GAN solve this function by training.



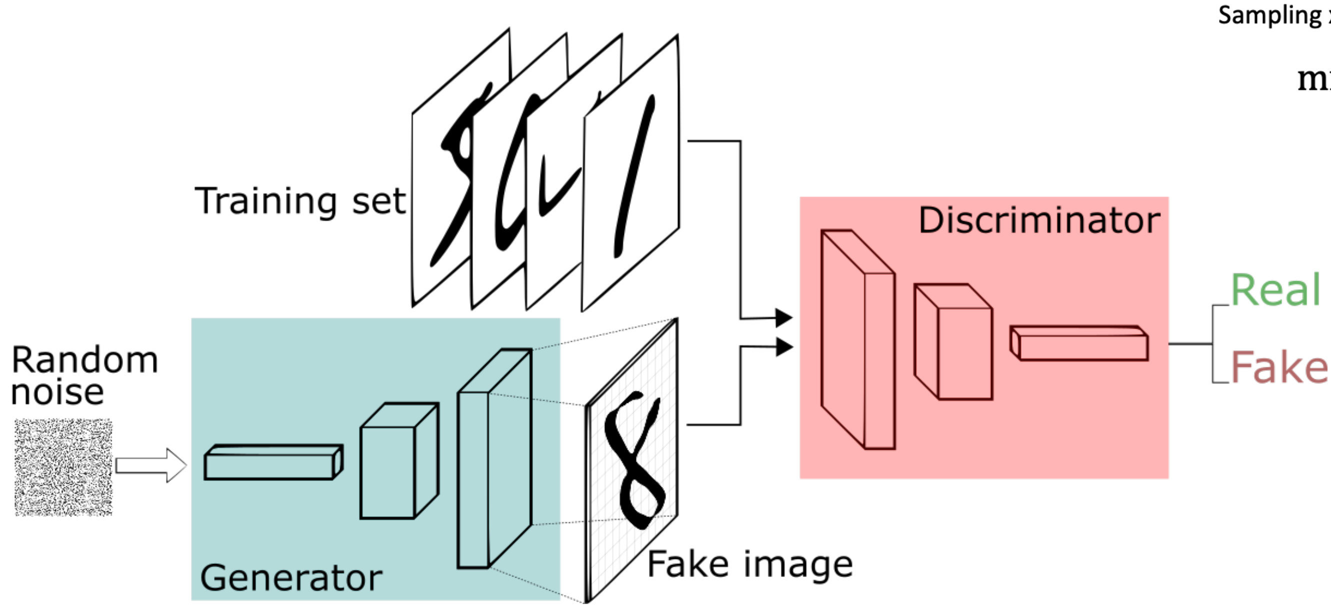
Sampling z from noise

Sampling x from real data

$$\text{minmax}\, V(D,G) = E_{x\sim p_{data}(x)}[logD(x)] + E_{z\sim p_z(z)}[\log(1 - D(G(z))]$$

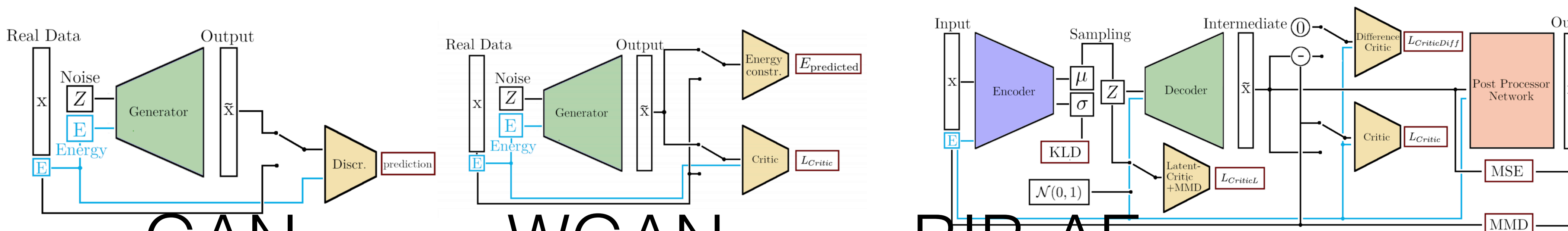
Probability determining real about real data

Probability determining real about fake data



Generative Calo Sims

https://arxiv.org/pdf/2005.05334.pdf



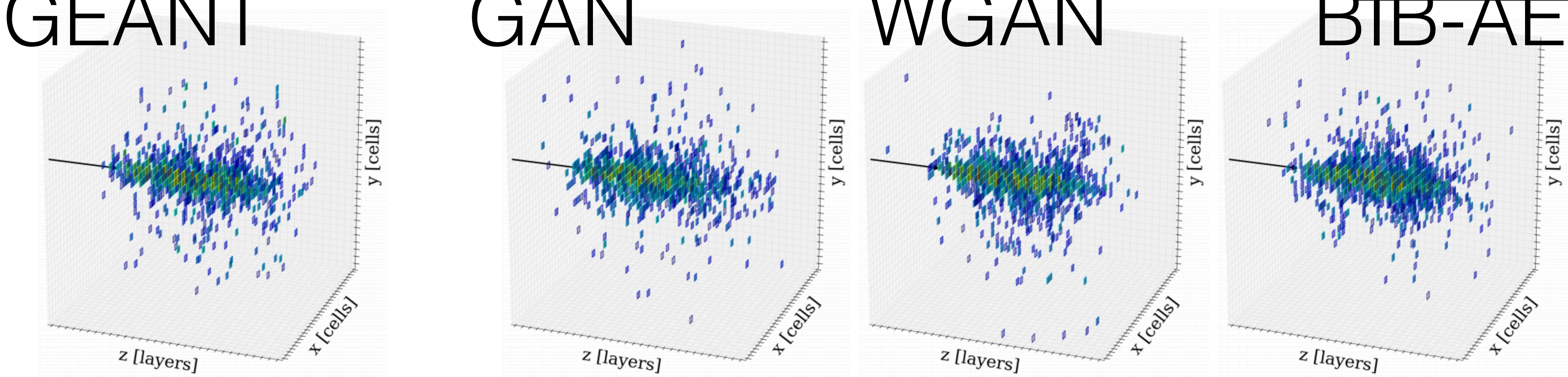
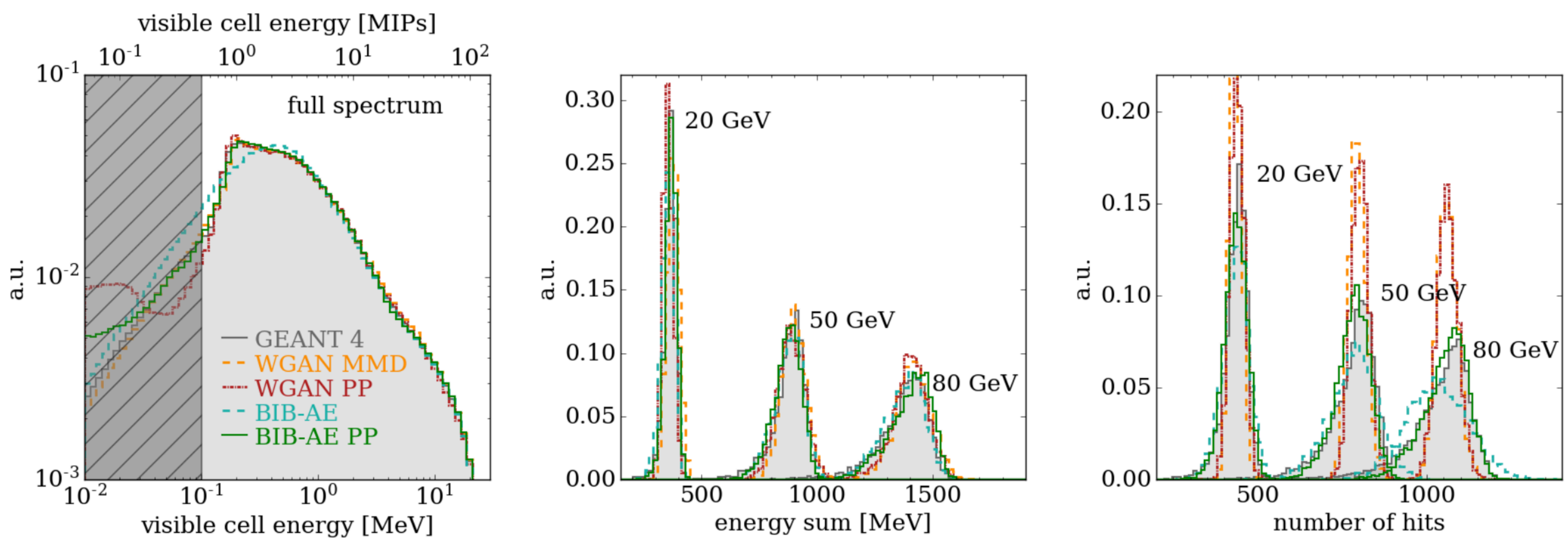


Fig. 5 Examples of individual 50 GeV photon showers generated by GEANT4 (left), the GAN (center left), WGAN (center right), and BIB-AE (right) architectures. Colors encode the deposited energy per cell.



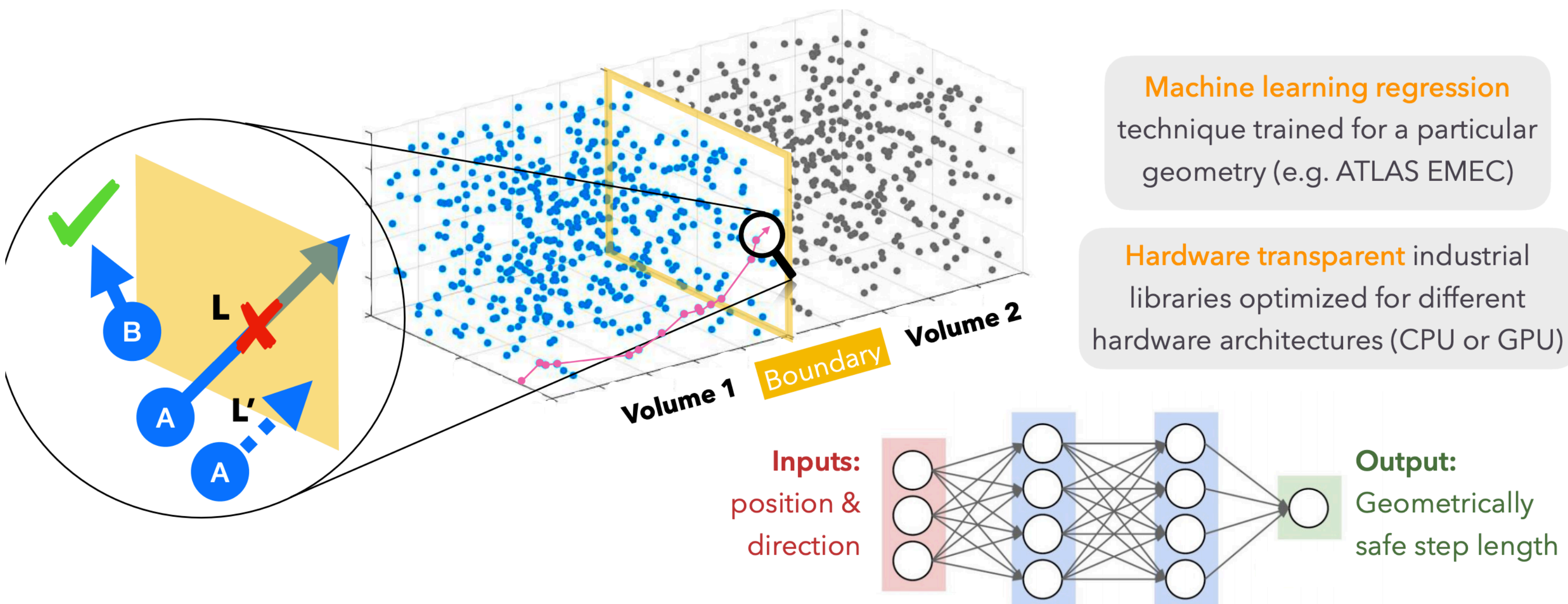
- ➡ almost there...
  - ➡ already much better than fast sim
  - ➡ lots of effort from many experiments
- ➡ Future
  - ➡ training with real data
  - ➡ digitalised/ reconstructed
  - ➡ tracking detectors
  - ➡ pile-up

Pre-Learning Det Geometry

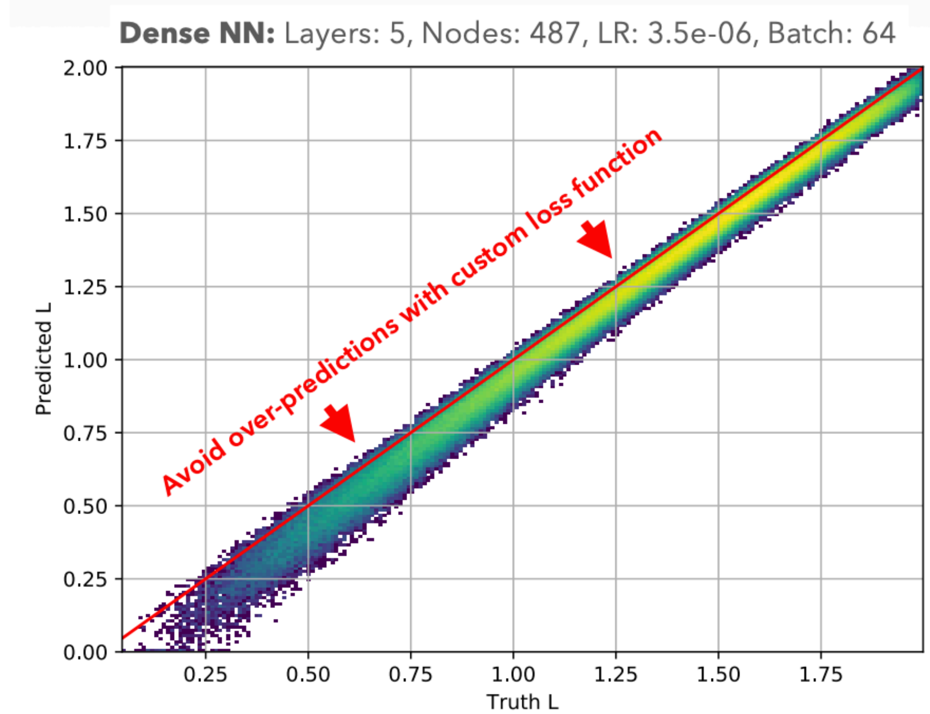
https://indico.cern.ch/event/852553/contributions/4059291/attachments/2128524/3584069/IML2020.pdf

The Idea

Surrogate modeling within Geant4: *Could we speed-up the geometry exploration by using a* pre-defined/learned map instead of algorithmic calculations *in each step?*



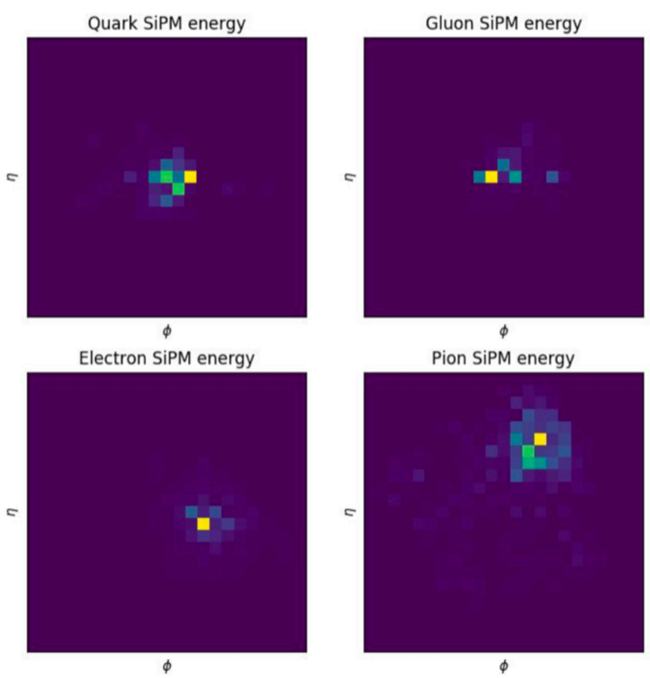
	Geometry	Time [μs]
Geant4	ATLAS EMEC	~ 5
	(nested) Twisted Trapezoids	50 - 100
Dense NN evaluation	-	~ 1000

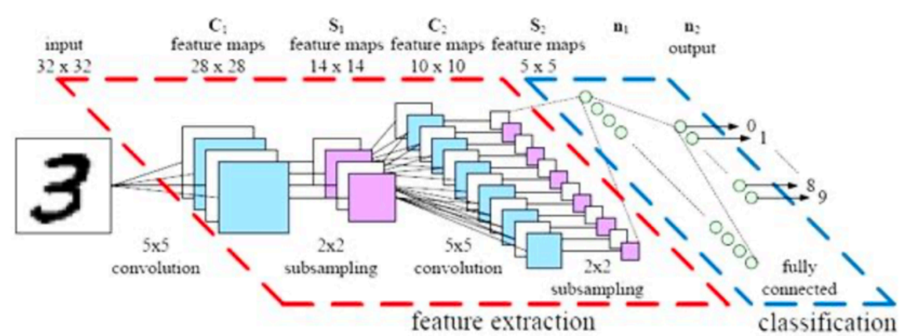


➡ single thread is slow currently, no optimisations

➡ very parallelable -> can gain in GPUs

e/γ/pion classification in calorimeters





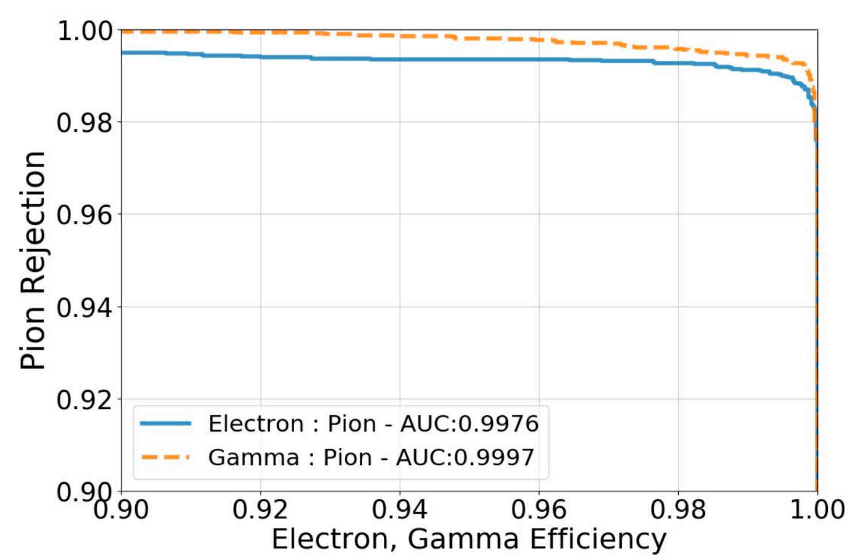
Concept of CNN

(http://parse.ele.tue.nl/cluster/2/CNNArchitecture.jpg)

Electron, gamma and pion discrimination

Gamma is more discriminated with pion, while pion can be rejected over 97% with any efficiency of both electron and gamma.

Few pion images which is very similar to electron and gamma are wrongly predicted.



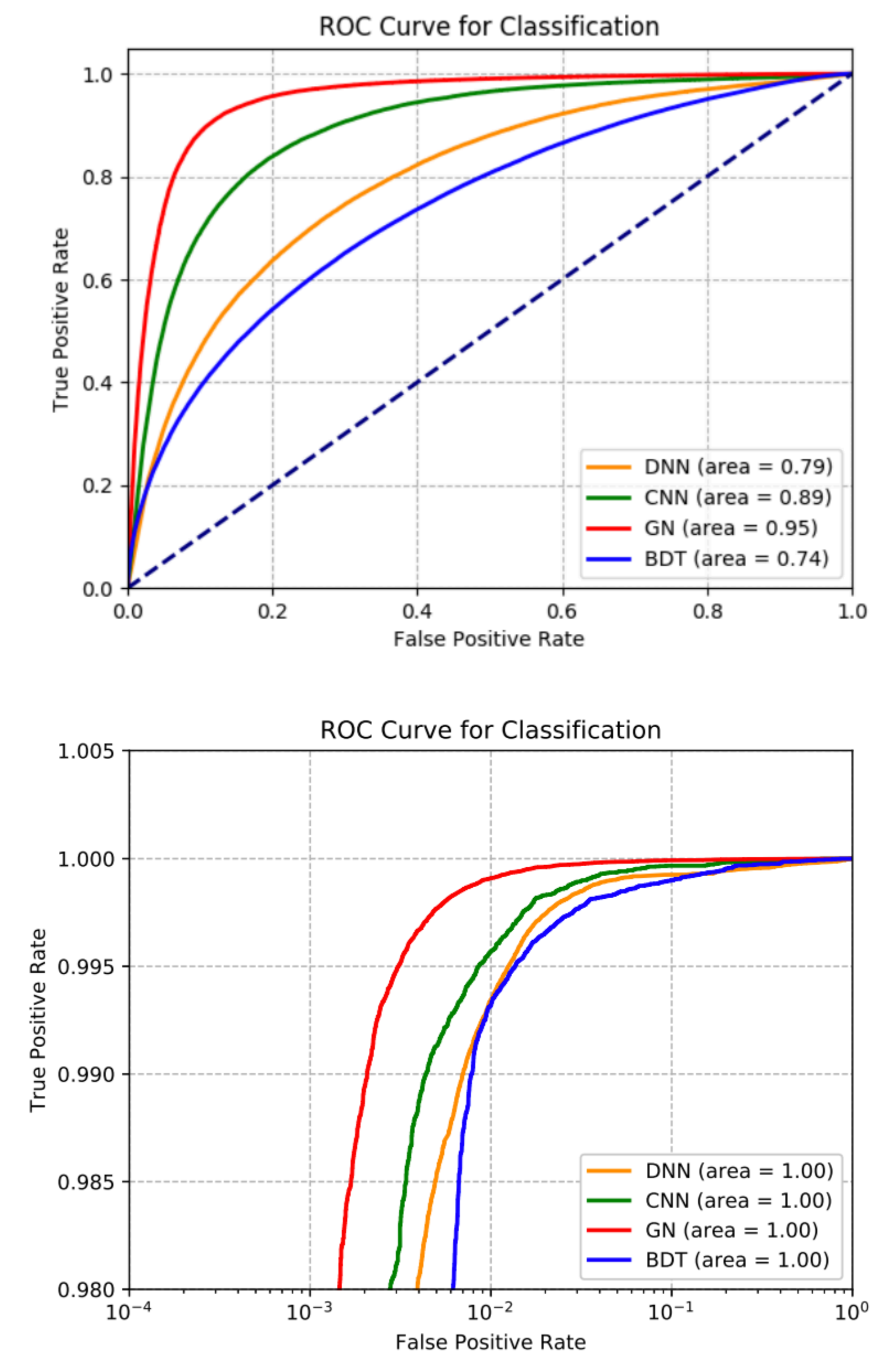


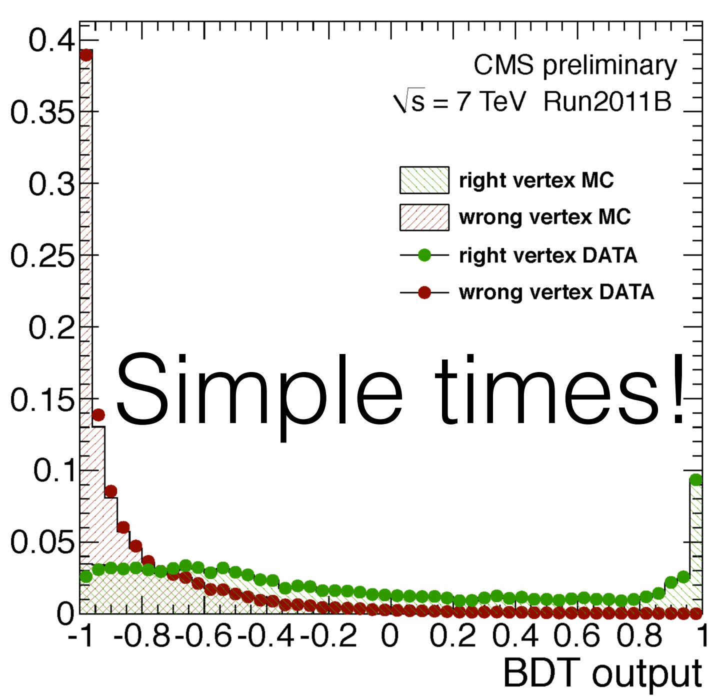
Fig. 13. ROC curve comparisons for γ vs. π^0 (top) and e vs. $\pi^\pm$ (bottom) classification using different neural network architectures. Samples include particle energies from 10 to 510 GeV, and an inclusive η range.

G2.08* : Deep learning implementation for Dual-Readout calorimeter - Yunjae Lee

https://arxiv.org/abs/1912.06794

Physics

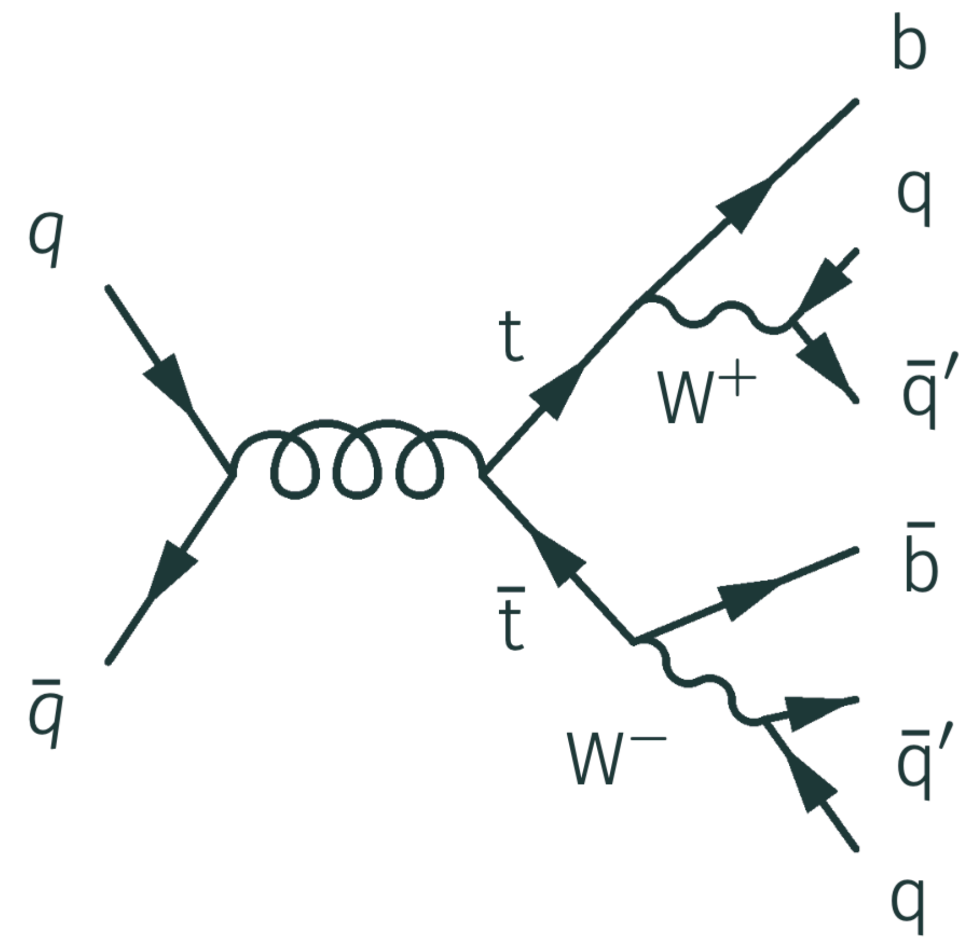
- ML is also widely used in final physics analysis
  - BDT has been a standard for awhile now
  - mostly used as a simple binary classifier to separate signal from background
  - now seeing very wide range of ML uses
    - CWoLa - Classification without labels (https://arxiv.org/abs/1708.02949)
      - data driven
      - anomaly detection, training classifiers with data
    - Unfolding with ML (https://indico.cern.ch/event/852553/contributions/4058157/attachments/2127229/3581617/IML_AButter.pdf)
    - use of machine learning as a scientific advisor (https://indico.cern.ch/event/852553/contributions/4059762/attachments/2126919/3581625/IML_SMenary_201021.pdf)
    - top tagging (https://indico.cern.ch/event/852553/contributions/4059759/attachments/2126774/3580825/LGN.pdf)



	AUC	Acc	$1/\epsilon_B$ ($\epsilon_S = 0.3$) single	mean	median	#Param
CNN	0.981	0.930	914±14	995±15	975±18	610k
ResNeXt	0.984	0.936	1122±47	1270±28	1286±31	1.46M
TopoDNN	0.972	0.916	295±5	382± 5	378 ± 8	59k
Multi-body N-subjettiness 6	0.979	0.922	792±18	798±12	808±13	57k
Multi-body N-subjettiness 8	0.981	0.929	867±15	918±20	926±18	58k
TreeNiN	0.982	0.933	1025±11	1202±23	1188±24	34k
P-CNN	0.980	0.930	732±24	845±13	834±14	348k
ParticleNet	0.985	0.938	1298±46	1412±45	1393±41	498k
LBN	0.981	0.931	836±17	859±67	966±20	705k
LoLa	0.980	0.929	722±17	768±11	765±11	127k
LDA	0.955	0.892	151±0.4	151.5±0.5	151.7±0.4	184k
Energy Flow Polynomials	0.980	0.932	384			1k
Energy Flow Network	0.979	0.927	633±31	729±13	726±11	82k
Particle Flow Network	0.982	0.932	891±18	1063±21	1052±29	82k
LGN	0.964	0.929	435			4.5k

ttbar full hadronic jet association

- The reconstruction of events with intermediate states decaying to jets requires a technique to assign jets to partons.
- The number of jets can be greater than the number of partons because of additional QCD radiation. It makes the assignment harder.
- We introduce **the self-attention for jet assignment (SAJA) network without requiring jet permutations**. We apply SAJA to find jet-parton assignments of fully-hadronic $t\bar{t}$ events to test the performance.



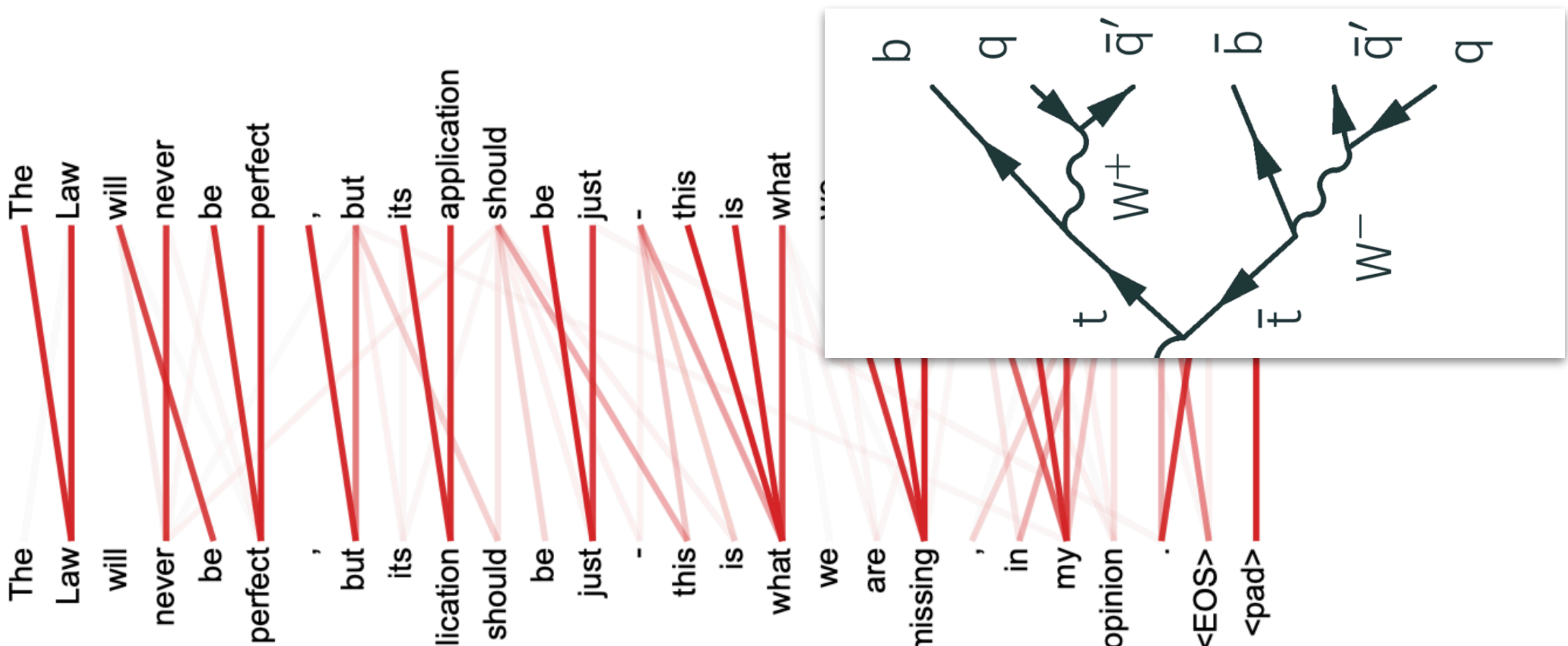
https://indico.cern.ch/event/852553/contributions/4059757/attachments/2127372/3582061/IML2020_Zero-Permutation-Jet-Parton-Assignment.pdf

ttbar full hadronic jet association

- The model implementation is based on TRANSFORMER, which is the neural machine translation model and features **self-attention**. [A. Vaswani, arXiv:1706.03762]

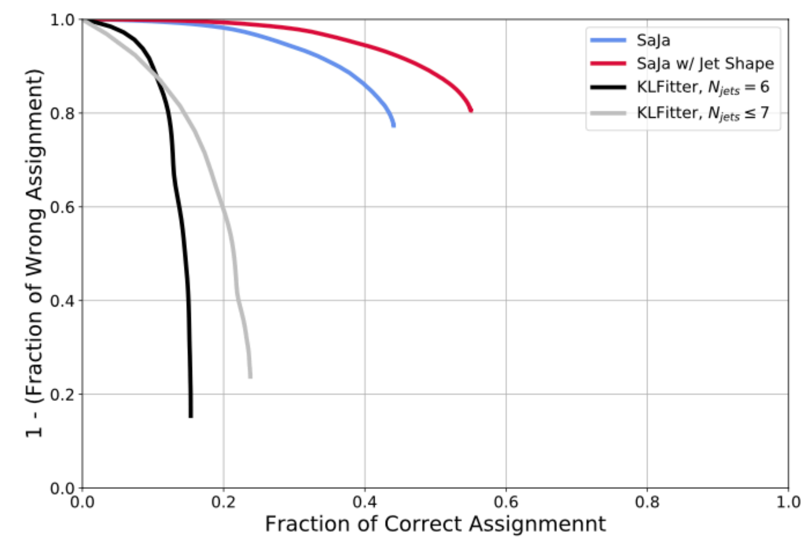
- Self-attention is a weight sum of the elements of the input set, where the weight matrix is also computed from the elements.

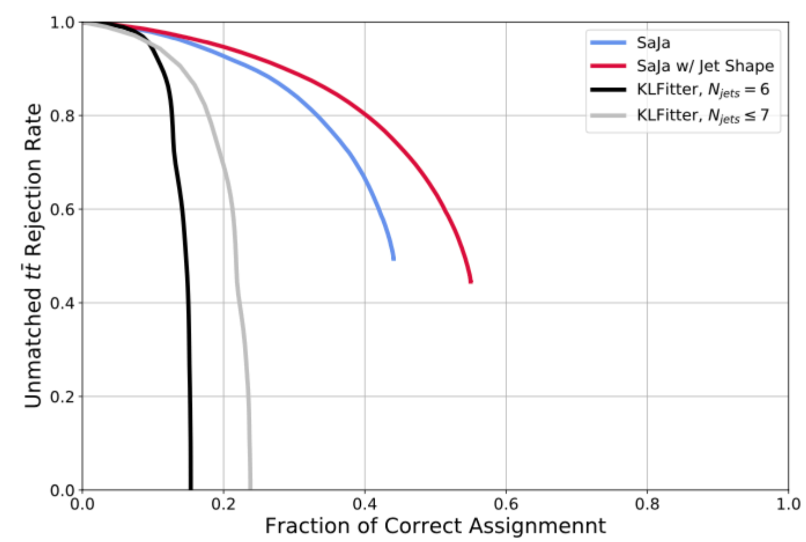
- Self-attention based model can learn the dependency between elements.

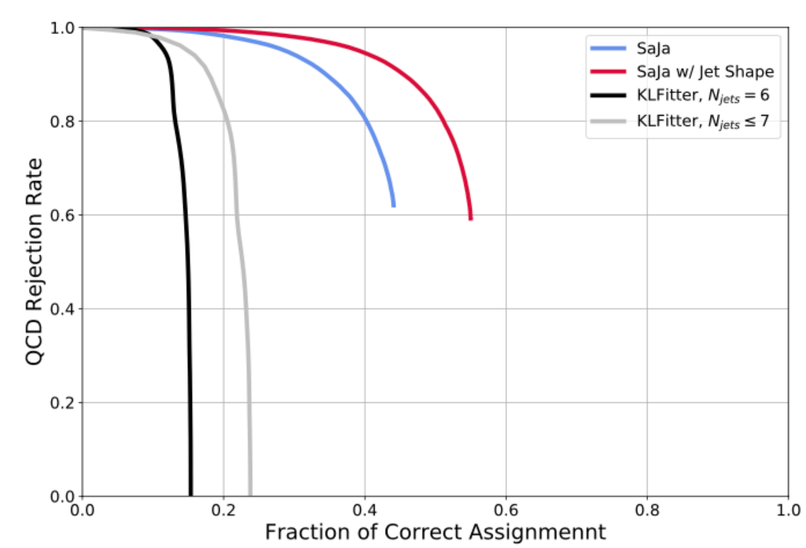


ttbar full hadronic jet association

- The assignment performance is visualized as ROC-like curves drawn by varying the threshold value for the predictive entropy of SAJA or the likelihood of KLFITTER.
- SAJA shows more powerful performance than KLFITTER.
- Predictive entropy not only reduces poor jet-parton assignments but also helps reduce unmatched $t\bar{t}$ and QCD multijet events without additional training process.
- Jet shape increases the fraction of correct assignment.



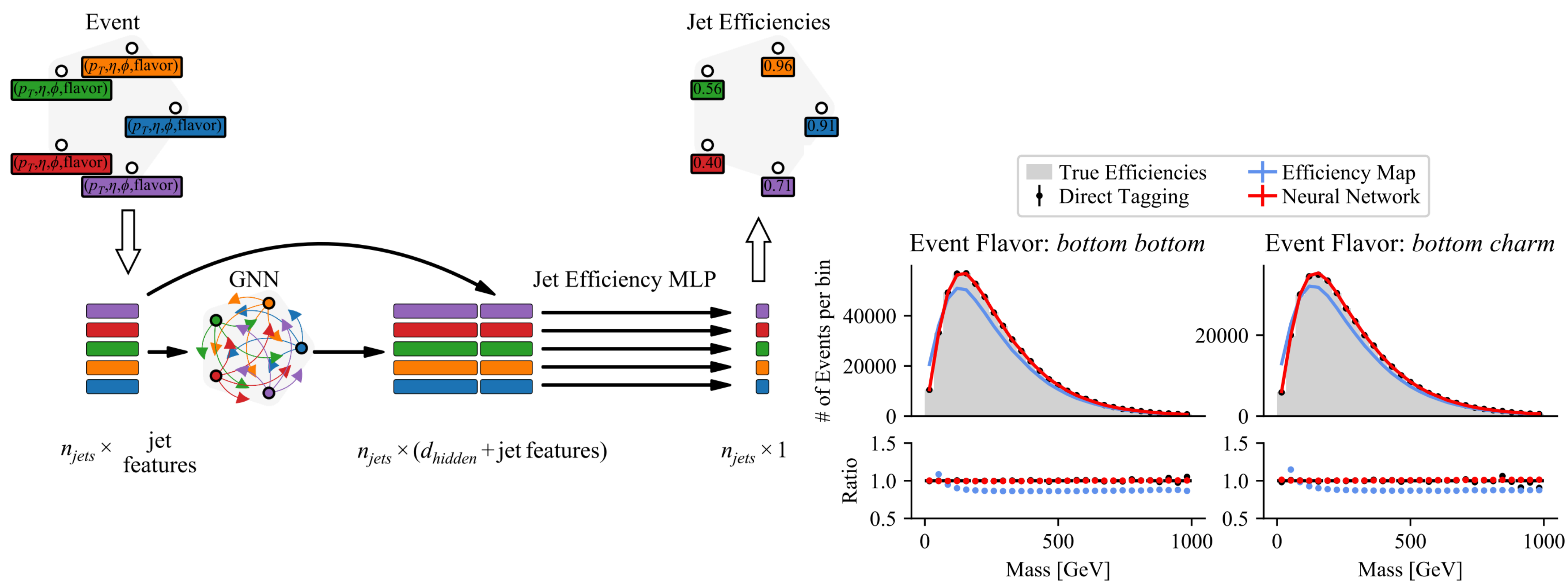




Efficiency Parameterisation with Neural Network

https://arxiv.org/abs/2004.02665v2

- ➡ Efficiency Map in p_T - η bins, used in b-tagging, lepton reco/id/trig etc
- ➡ Limited due to MC stats
- ➡ Use GNN to learn efficiencies as function parameters



- ➡ more input parameters -> generalise well -> better efficiency estimations
- ➡ Learns the jet-jet dependency -> important jets close together

Summary

- ML has overtaken particle physics
  - Standard tool for HEP
  - Integrated into core CMS SW
- Detector Monitoring to Identification should be production ready by future experiments
- New networks, training techniques updated
  - new imaginative ways to implement in HEP!

- Theory
- Accelerator
- Detector Monitoring
- Triggering
- Detector Simulation
- Reconstruction
- Identification
- Physics Analysis
- Computing

The EnD

➡ ML is overtaking particle physics