Lecture 6:
Keras
and / or
Deep Learning in HEP

Amir Farbin

Keras

https://keras.io/

Why go Deep?

- ***Better Algorithms***
    - DNN-based classification/regression generally **out perform** hand crafted algorithms.
    - In some cases, it may provide a ***solution*** where ***algorithm approach doesn't exist or fails***.
    - ***Unsupervised learning***: make sense of complicated data that we don't understand or expect.
- ***Easier Algorithm Development***: ***Feature Learning*** instead of *Feature Engineering*
    - Reduce time physicists spend writing developing algorithms, ***saving time and cost***. (e.g. ATLAS > $250M spent software)
    - Quickly perform performance ***optimization*** or ***systematic studies***.
- ***Faster Algorithms***
    - After training, DNN inference is often *faster* than sophisticated algorithmic approach.
    - DNN can ***encapsulate expensive computations***, e.g. Matrix Element Method.
    - ***Generative Models*** enable fast simulations.
    - ***Already parallelized*** and optimized for GPUs/HPCs.
    - ***Neuromorphic*** processors.

Where is ML needed?

- Traditionally ML Techniques in HEP
  - Applied to Particle/Object Identification
  - Signal/Background separation
  - Here, ML maximizes reach of existing data/detector… equivalent to additional integral luminosity.
  - There is lots of interesting work here… and potential for big impact.
- Now we hope ML can help address looming computing problems of the next decade:
  - ***Reconstruction***
    1. Intensity Frontier- ***LArTPC*** Automatic Algorithmic ***Reconstruction*** still struggling
    2. Energy Frontier- ***HL-LHC Tracking***- Pattern Recognition blows up due to combinatorics
  - ***Simulation***
    3. LHC Calorimetry- Large Fraction of ATLAS CPU goes into ***shower simulation***.

Problems

Data Analysis

- Objectives:
  - ***Searches*** (hypothesis testing): Likelihood Ratio Test (Neyman-Pearson lemma)
  - ***Measurements***: Maximum Likelihood Estimate
  - ***Limits*** (confidence intervals): Also based on Likelihood

$$\frac{P(x|H_1)}{P(x|H_0)} > k_\alpha$$

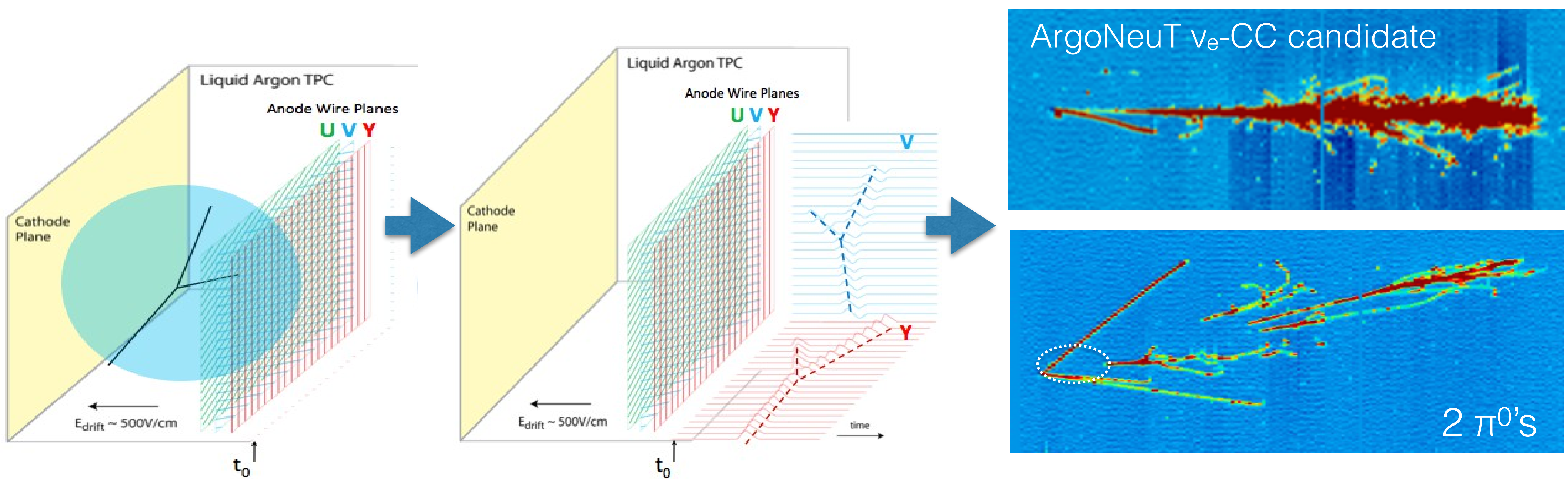
- ***Likelihood***

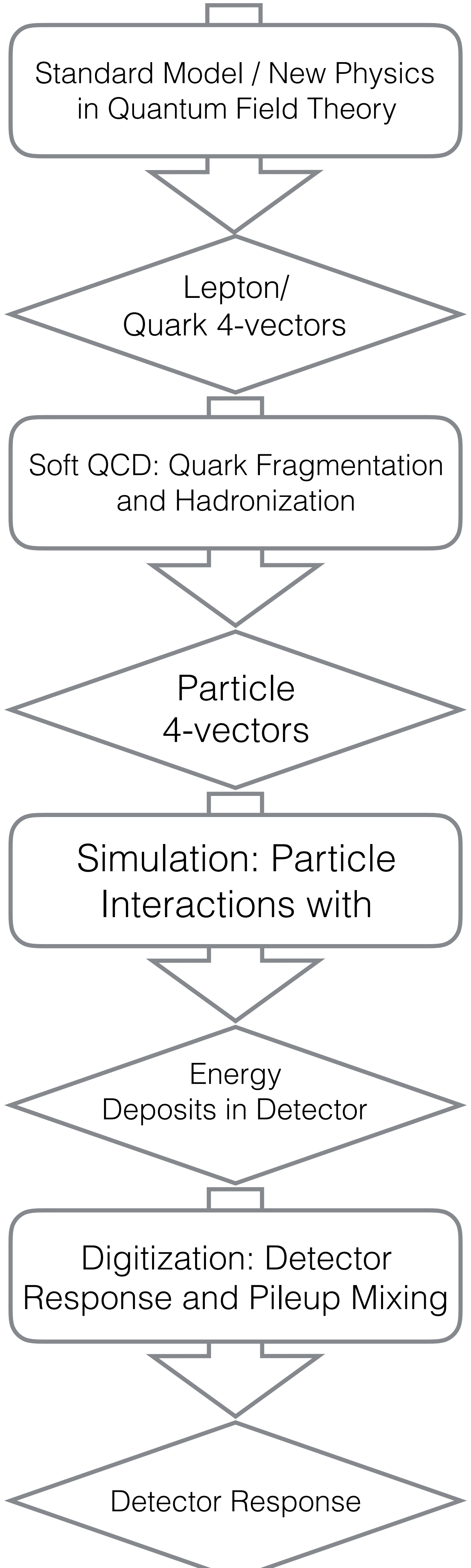
$$p(\{x\}|\theta) = \mathrm{Pois}(n|\nu(\theta)) \prod_{e=1}^{n} p(x_e|\theta)$$

  - *n* Independent Events (*e*) with Identically Distributed Observables ({*x*})
  - Significant part of Data Analysis is ***approximating the likelihood*** as best as we can.
- We dedicate huge amount of resources to use Monte Carlo simulation to effectively estimate these likelihoods.

Neutrino Detectors

- ***Need large mass/volume*** to maximize chance of neutrino interaction.
- Technologies:
  - Water/Oil Cherenkov
  - Segmented Scintillators
  - ***Liquid Argon Time Projection Chamber: promises ~ 2x detection efficiency.***
    - ***Provides tracking, calorimetry, and ID all in same detector***.
    - Chosen technology for US's flagship LBNF/DUNE program.
    - Usually 2D read-out… 3D inferred.
- After many years of trying, ***good automatic reconstruction still not demonstrated.***



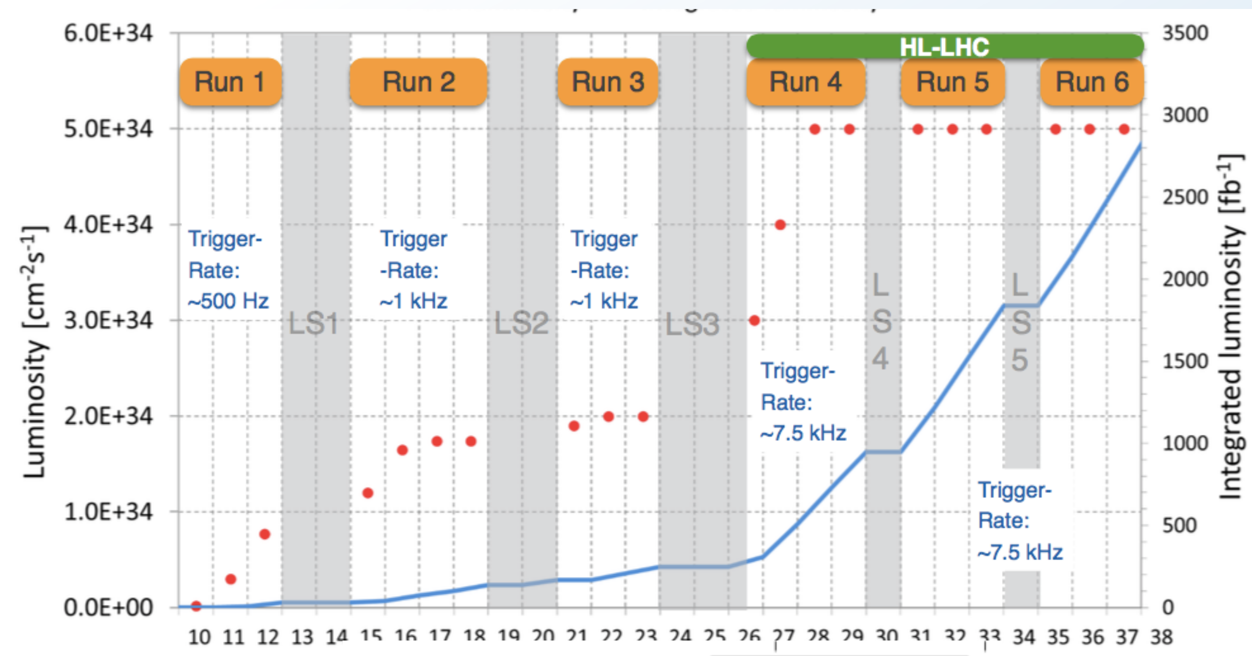


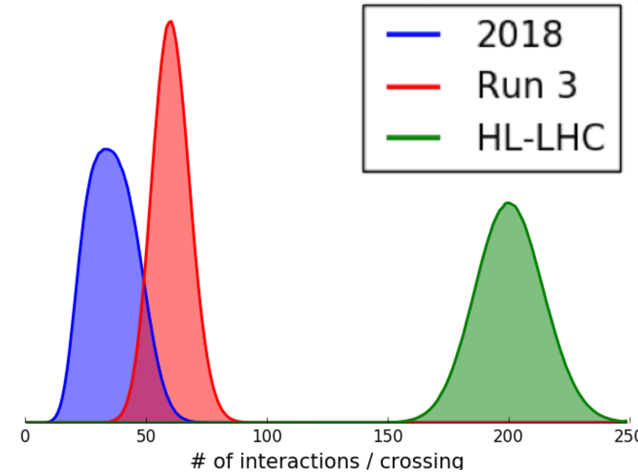
Simulation

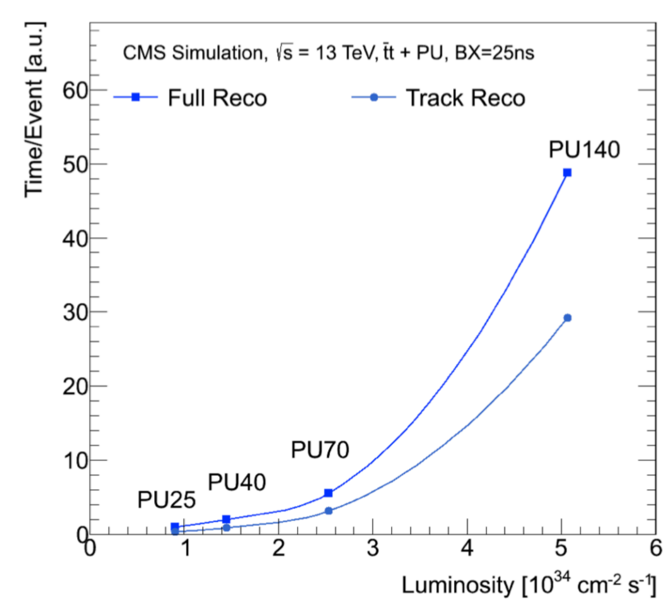
- Simulation in HEP is a multi-step process…
- Two steps are irreversible.
  - ***Hadronization***: Quarks turn to jets of particles via Quantum Chromodynamics (QCD) at energies where theory is too strong to compute perturbatively.
    - Use semi-empirical models tuned to Data.
  - ***Simulation***: Particles interact with the Detector via stochastic processes
    - Use detailed Monte Carlo integration over the "micro-physics"
- Therefore we cannot formally evaluate the likelihoods.
- Rely on Monte Carlo Method to perform Probability Density Estimation
- The simulation step is extremely time consuming… O(1 hr) / collision… LHC produces 40 million/sec
  - ATLAS simulation takes O(50%) of ATLAS resource
    - Lager fraction than CMS because of calorimeter
- For HL-LHC, NLO and NNLO generation will become even more relevant… these can be time consuming too.

HL-LHC

- ***Higher Granularity + High Trigger Rates***
  - ~10x higher input rates.
  - Trigger Needs:
    - Better Calorimetry
    - Tracking
- ***Low New Physics x-sections***, need:
  - Detail Physics: NLO / NNLO
  - Faithful Simulation: Geant
- ***High Pileup***: O(200) proton collision / crossing
  - Tracking Pattern Recognition



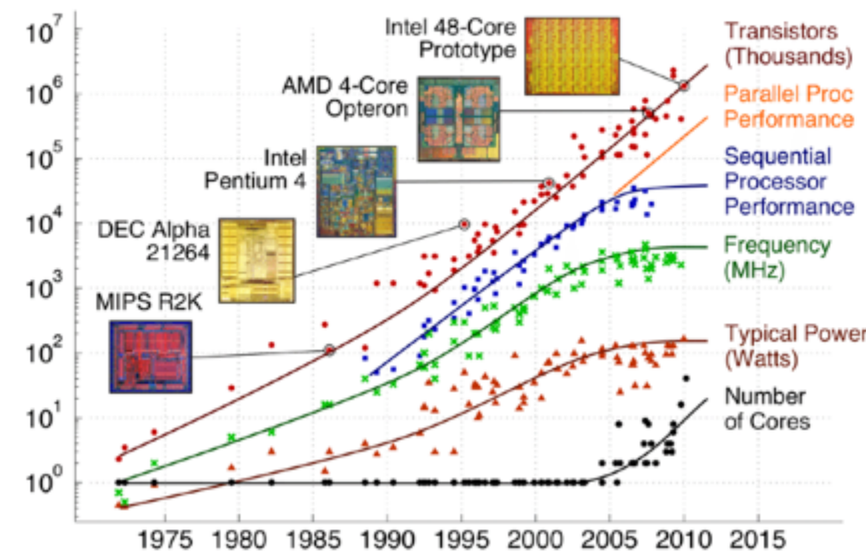


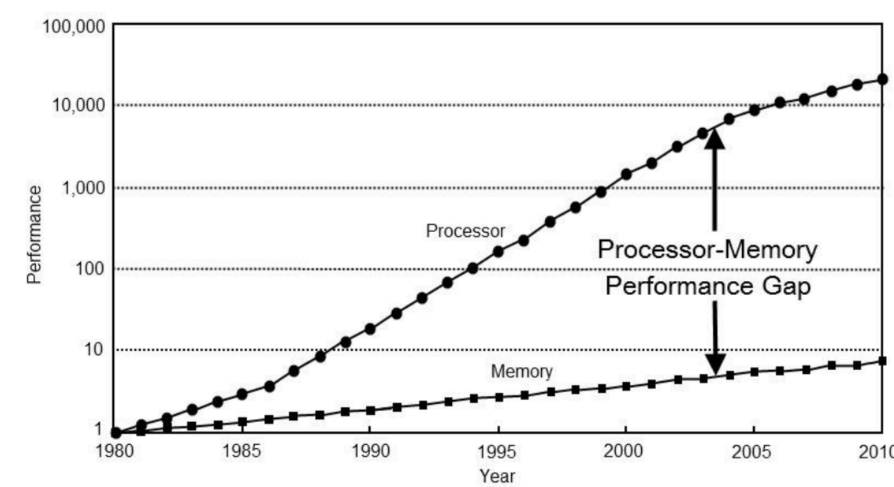


https://indico.cern.ch/event/681549/contributions/2930935/attachments/1664434/2667677/lhcp_lange_2018.pdf

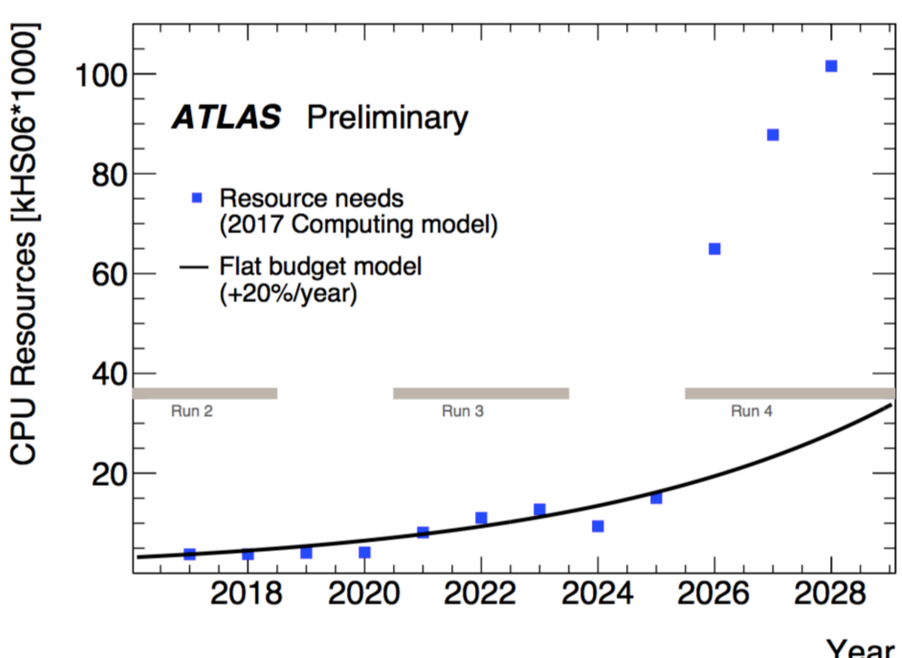
Computing

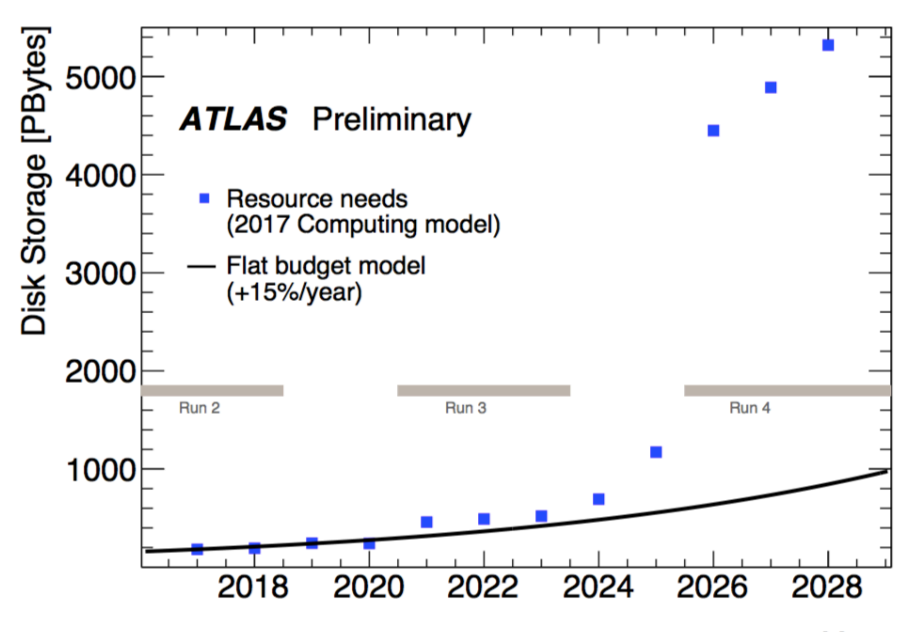
- ***HEP Reco*** is ***Embarrassingly parallel problem*** → Single threaded and memory-heavy software
    - Past few decades: scaling via ever faster / denser commodity linux boxes
- ***Moore's law has stalled***:
    - Cost of adding more transistors/silicon area no longer decreasing.
    - Trend towards more cores and slower memory access.
    - Co-processors: MiC, GPUs, FPGA, …
- Storage Scaling also a problem…
- ***HL-LHC computing requires budget many times larger than LHC***.





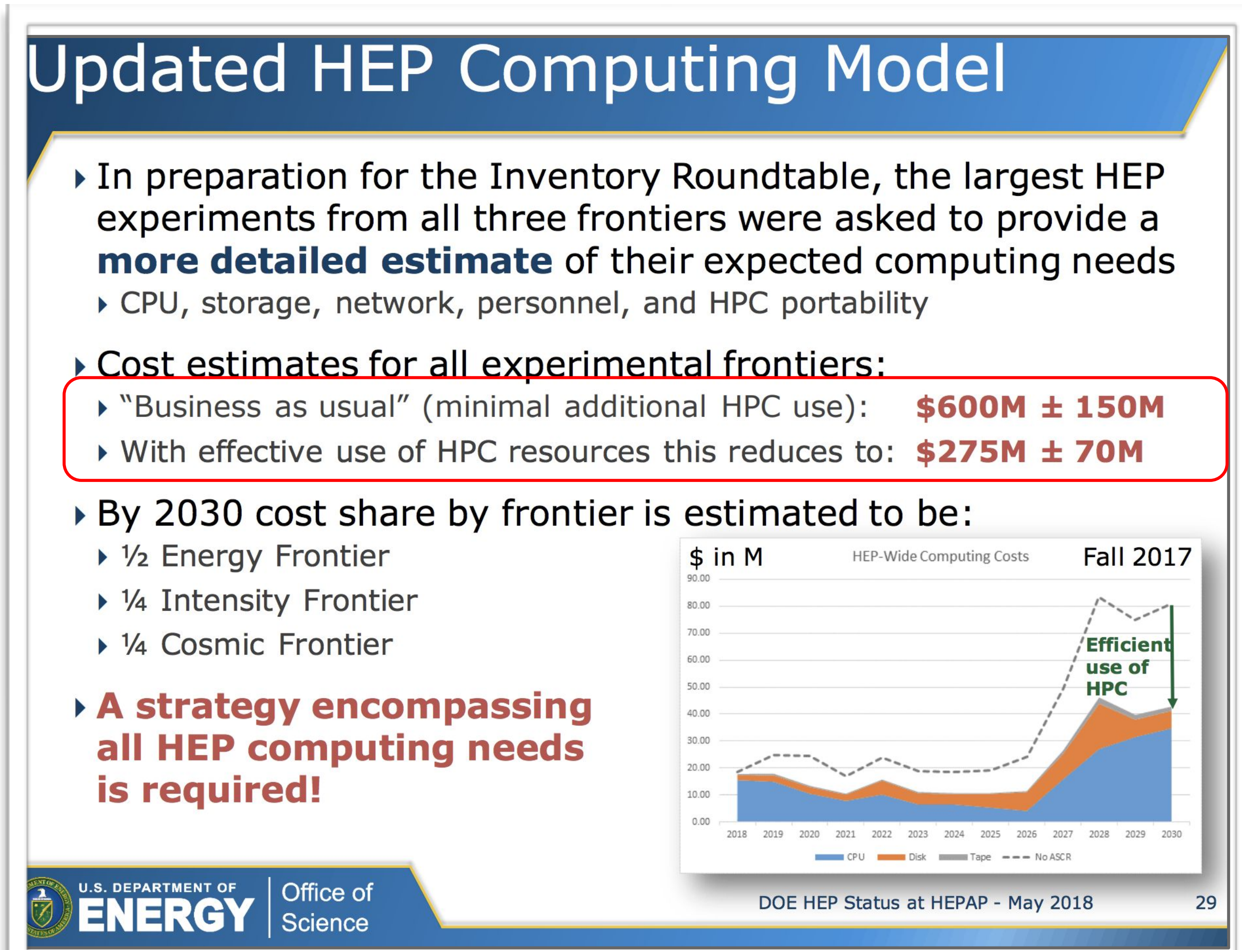
Processor-Memory speed evolution





Computing Solutions

- ***Highly parallel processors*** (e.g. GPUs) are already > 10x CPUs for certain computations.
    - Trend is away from x86 towards ***specialized hardware*** (e.g. GPUs, Mics, FPGAs, Custom DL Chips)
    - Unfortunately parallelization (i.e. Multi-core/GPU) has been extremely difficult for HEP.
- ***Leverage opportunistic resources and HPC***
    - most computation power in highly parallel processors
- Replace trigger, reconstruction, and simulation algorithms with better, faster, and easier ***Deep Learning*** algorithms.
    - These algorithms not only run on newest accelerators, but are the driving force in processor evolution.



[Jim Siegrist, HEPAP meeting, May 2018]

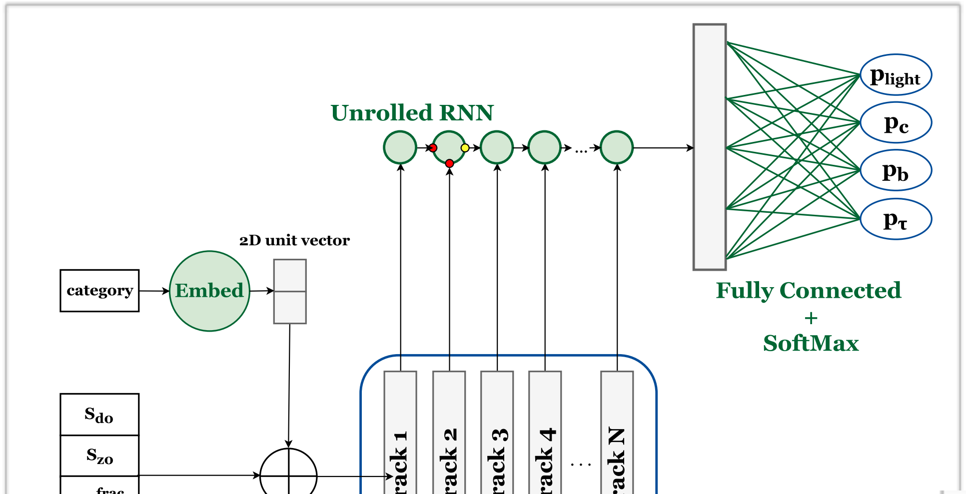
Deep Learning in HEP

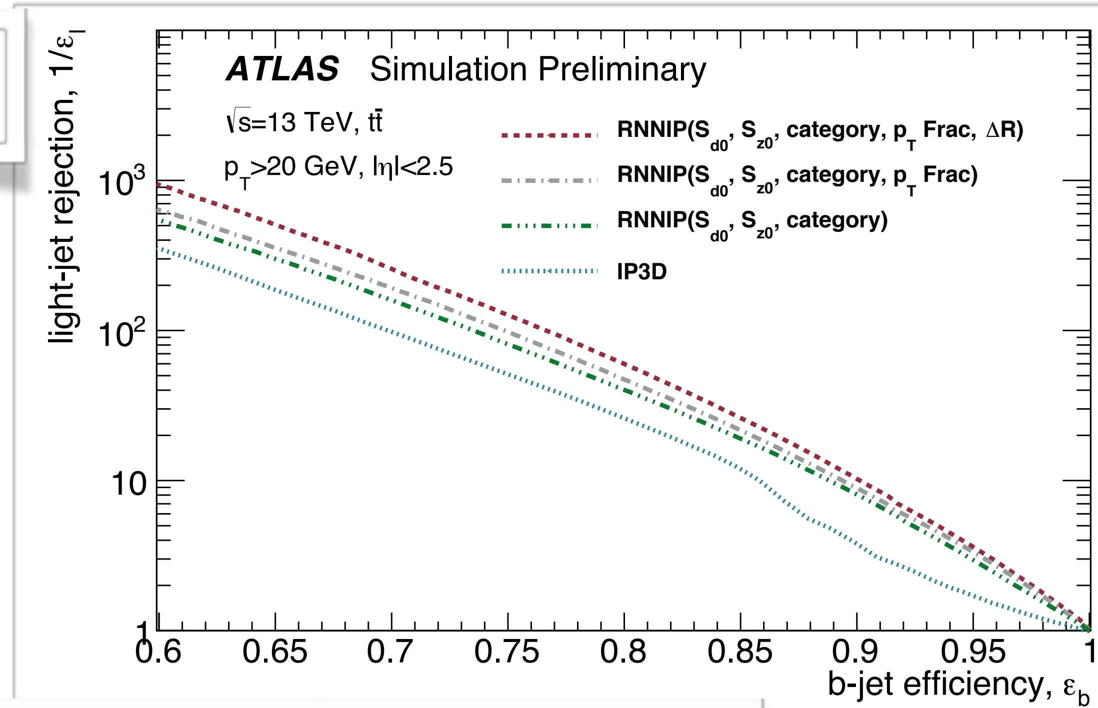
- Few simple cases already deployed. e.g. feature based b-tagging
- Lots of promising studies of more sophisticated approaches and applications.
- Basic framework integration… difficult and subtle concerns on horizon
    - e.g. DL framework integration. Model storage and book keeping. Model Memory management.
- We have a long program of research before suitability and ability to do real work.
- Example path:
    - ***Feasibility*** studies demonstrating potential.
        - Simplified datasets, idealized formulations. e.g. work by Micky, Luke, & Ben
    - ***Growing realism***: Tackling real detector. e.g. ATLAS GAN.
    - ***Systematic studies***: e.g. CaloDNN
    - ***Physics Application***: Target high impact potential physics, and work it through.
    - ***Build***: Democratize by integrating into framework.
    - ***Integrate***: Production workflow.

Identification of Jets Containing b-Hadrons with Recurrent Neural Networks at the ATLAS Experiment

ATL-PHYS-PUB-2017-003

Dan Guest
ATLAS Collaboration

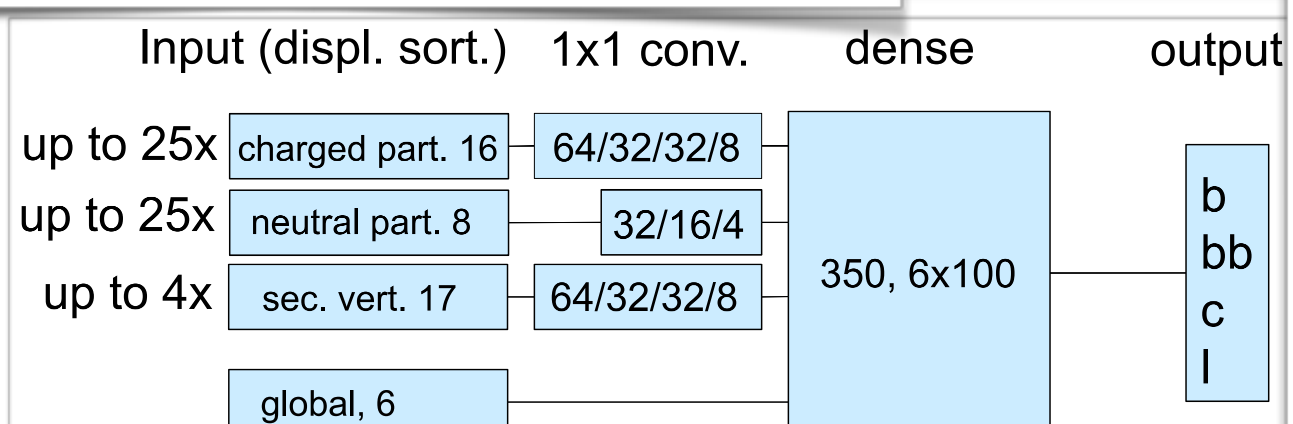


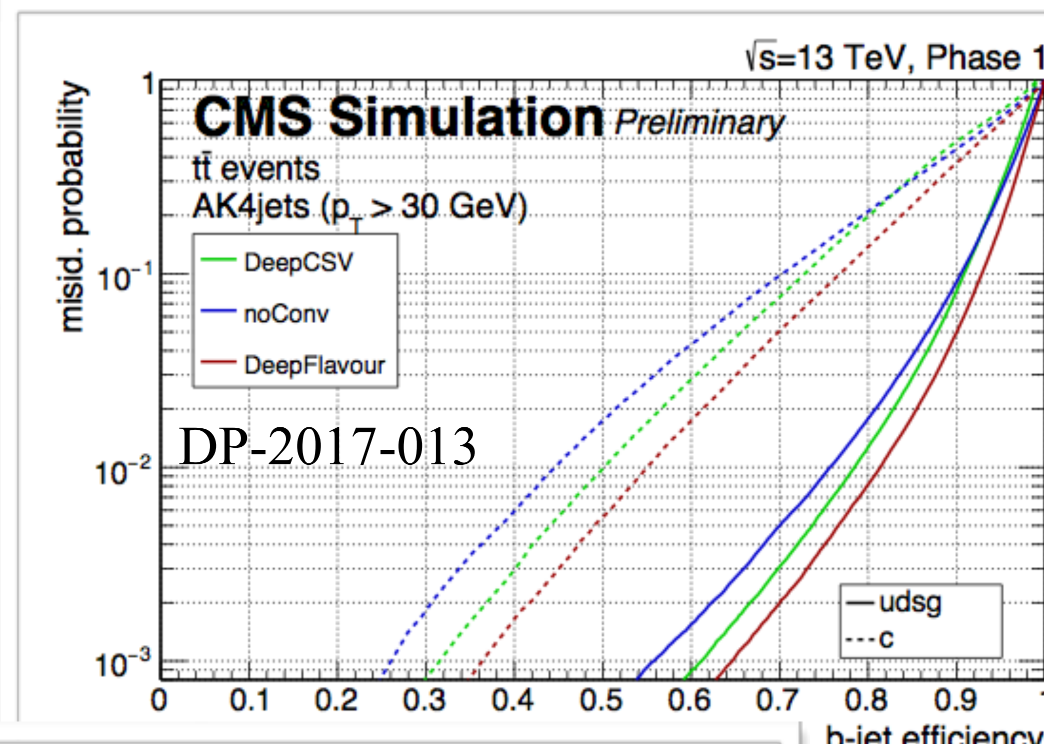


RNN with IP3D inputs improves light rejection by 1.7
With $\Delta R(\text{track}, \text{jet})$ and p_T^{frac}, improves light rejection by 2.5

Convolutional NN for flavour-tagging

Markus Stoye[1,2] for the CMS collaboration
CERN[1], ITN aMVA4newphysics[2]





- E.g. last row, 15% more background and up to ~50% more signal
- Significantly improved limit (150 GeV in Higgsino mass)

Example: Calorimetry with Deep Learning

Calorimetry with Deep Learning: Particle Identification and Simulation for Collider Physics

Dawit Belayneh[1], Federico Carminati[2], Amir Farbin[3], Benjamin Hooberman[4], Gulrukh Khattak[2,5], Miaoyuan Liu[6], Junze Liu[4], Dominick Olivito[7], Vitória Barin Pacela[8], Maurizio Pierini[2], Alexander Schwing[4], Maria Spiropulu[9], Sofia Vallecorsa[2], Jean-Roch Vlimant[9], Wei Wei[4], and Matt Zhang[a4]

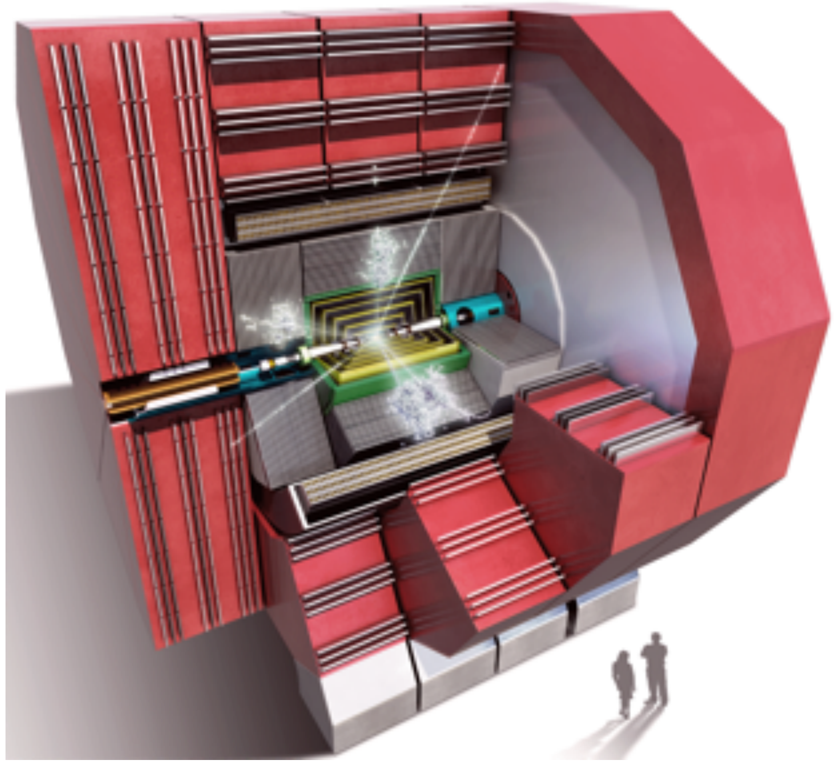
[1] Univ. of Chicago
[2] CERN
[3] Univ. of Texas Arlington
[4] Univ. of Illinois at Urbana-Champaign
[5] UET Peshawar
[6] Fermi National Accelerator Laboratory
[7] Univ. of California, San Diego
[8] Univ. of Helsinki
[9] California Institute of Technology

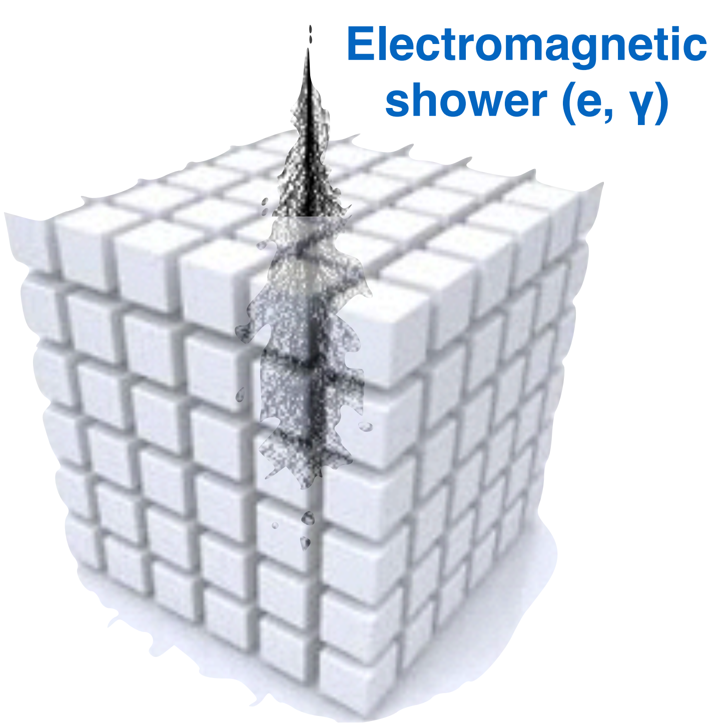
the date of receipt and acceptance should be inserted later

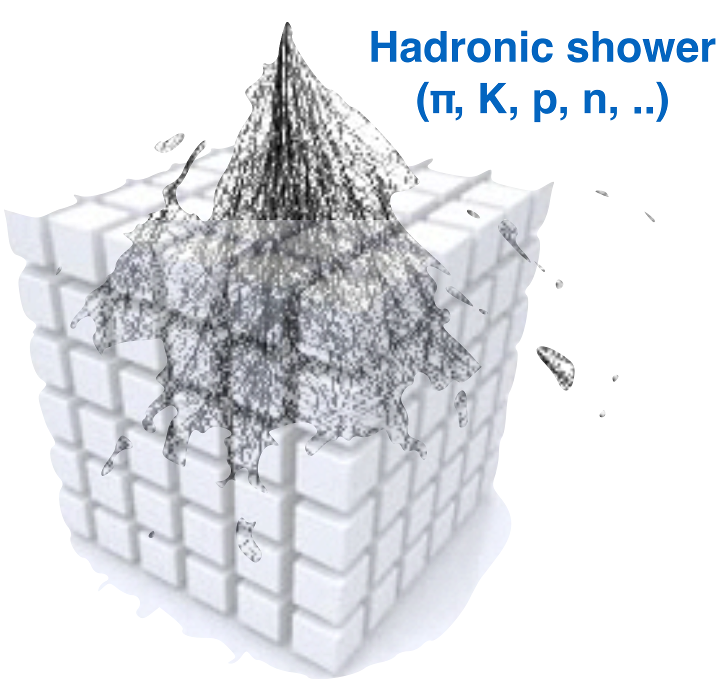
Abstract. Using calorimeter data from particle detectors, we apply various machine learning techniques to tasks involving the identification and simulation of particles produced in high-energy particle collisions. We train neural networks on raw calorimeter-cell-level information, and show that these provide significant improvements in performance for particle classification and energy regression as compared to methods which rely on traditional algorithms, such as feature-based neural nets and boosted decision trees. We compare various neural architectures, and perform hyperparameter scans to study the optimal configurations of these nets for the classification and regression problems. Furthermore, we demonstrate the applicability of these nets to other detector geometries, specifically ATLAS-like and CMS-like geometries. In addition, we train a generative adversarial network that provides reasonable modeling of shower features for different particle types at various angles and energies. This network could serve as a fast and computationally light method of simulating particle showers in generated collision events.

Calorimeter Dataset

- CLIC is a proposed CERN project for a linear accelerator of electrons and positrons to TeV energies (~ LHC for protons)
    - LCD is a detector concept.
    - Not a real experiment yet, so we could simulate data and make it public.
- The LCD calorimeter is an array of absorber material and silicon sensors comprising the most granular calorimeter design available
    - Data is essentially a 3D image
- With at effective eta/phi resolution of 0.003x0.003, we can down sample to get ~ ATLAS granularity: 0.025x0.1 (pre-sampler) to 0.2x0.1 Tile D.

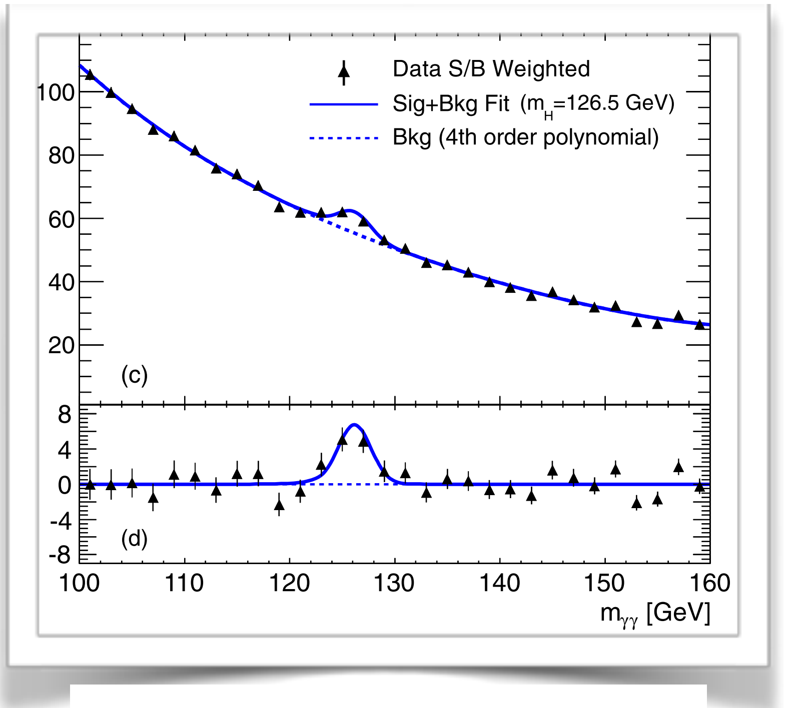


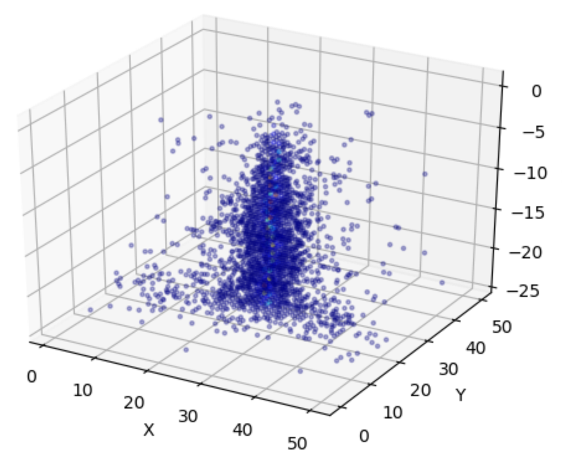


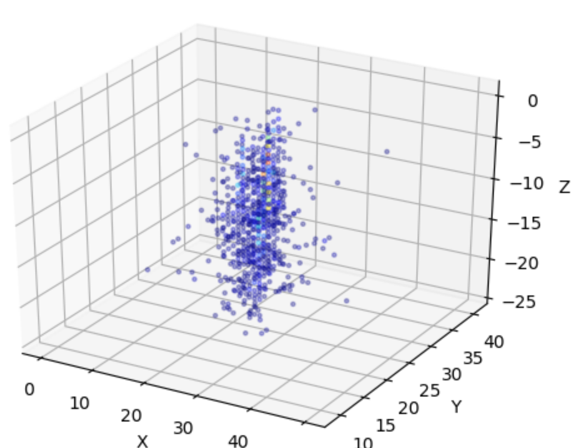


The Project

- 3 Parts to calorimetry:
  - ***Classification***: ID'ing the type of particle.
    - Better performance, less background under peak
  - ***Regression***: Measuring the Energy
    - Better performance, skinner peak, less background under peak.
  - ***Simulation***: Necessary for every step… very computationally expensive.
    - Faster … save money.
- Primary goal: while all of these have been demonstrated to be feasible, move toward realism, deep investigation, and implementation.
- Sub-Goals:
  - Demonstrate improvement over traditional techniques
  - Hyper-parameter studies
  - Project to different real detectors







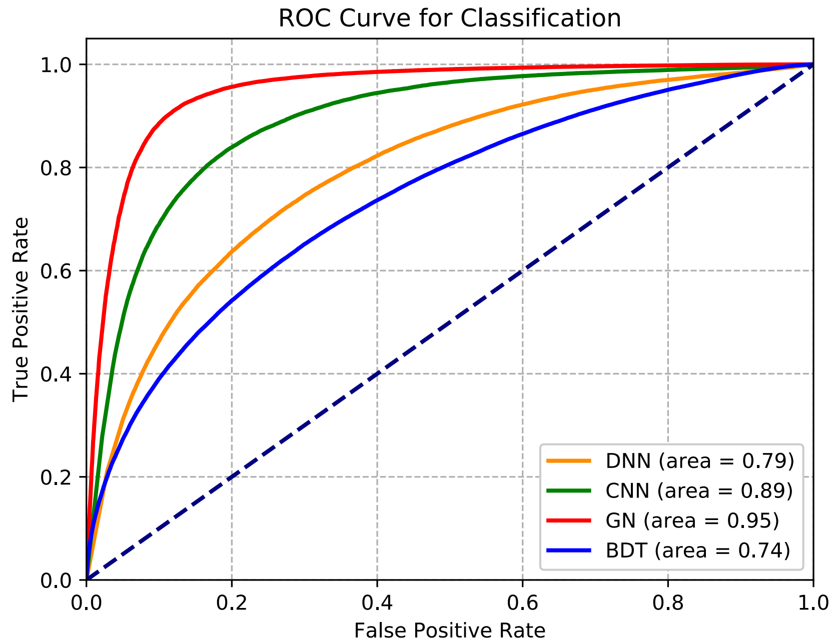
Classification

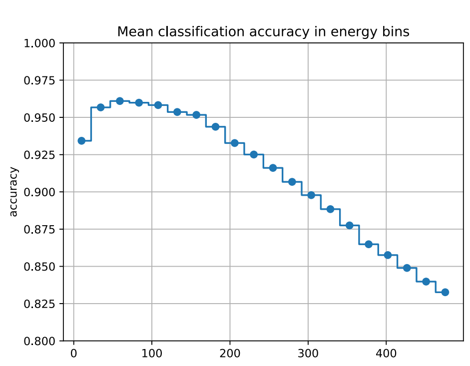
Compare BDT trained Features to DNNs trained on Images.

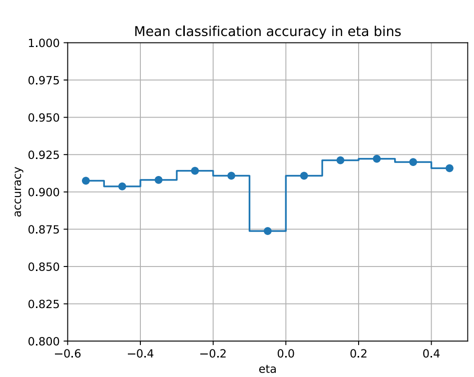
Add realism…

γ vs π_0

π_0 $\Delta\theta_{\gamma\gamma} < 0.01$

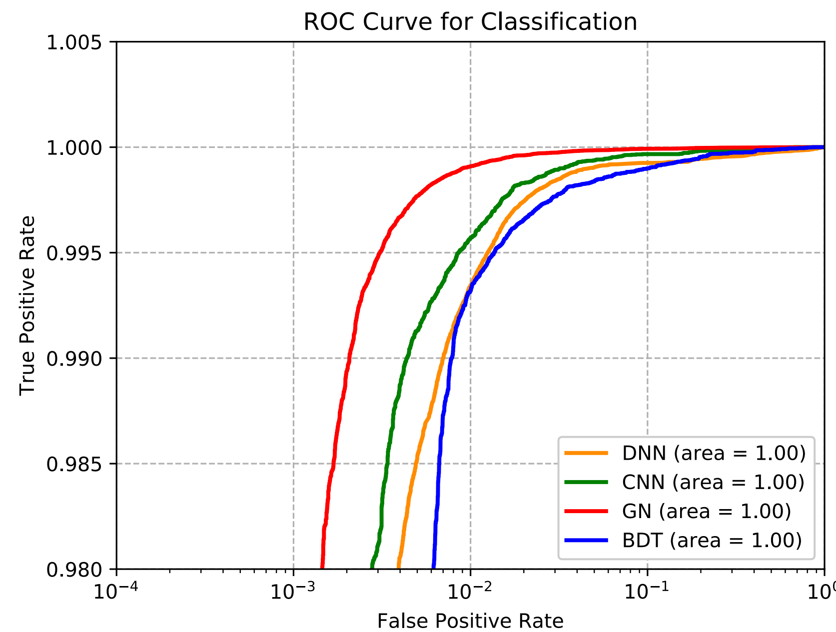


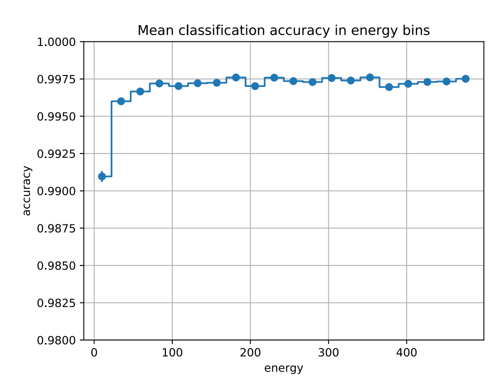


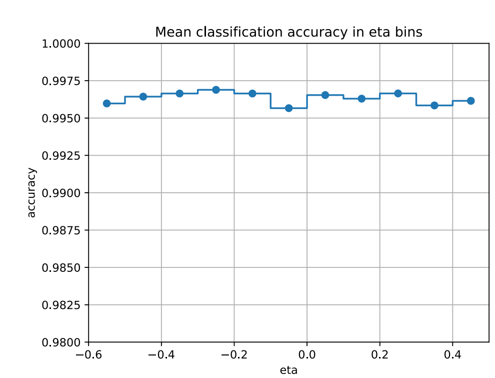


$e^\pm$ vs $\pi^\pm$

$E_{HCAL}/E_{ECAL} < 0.1$



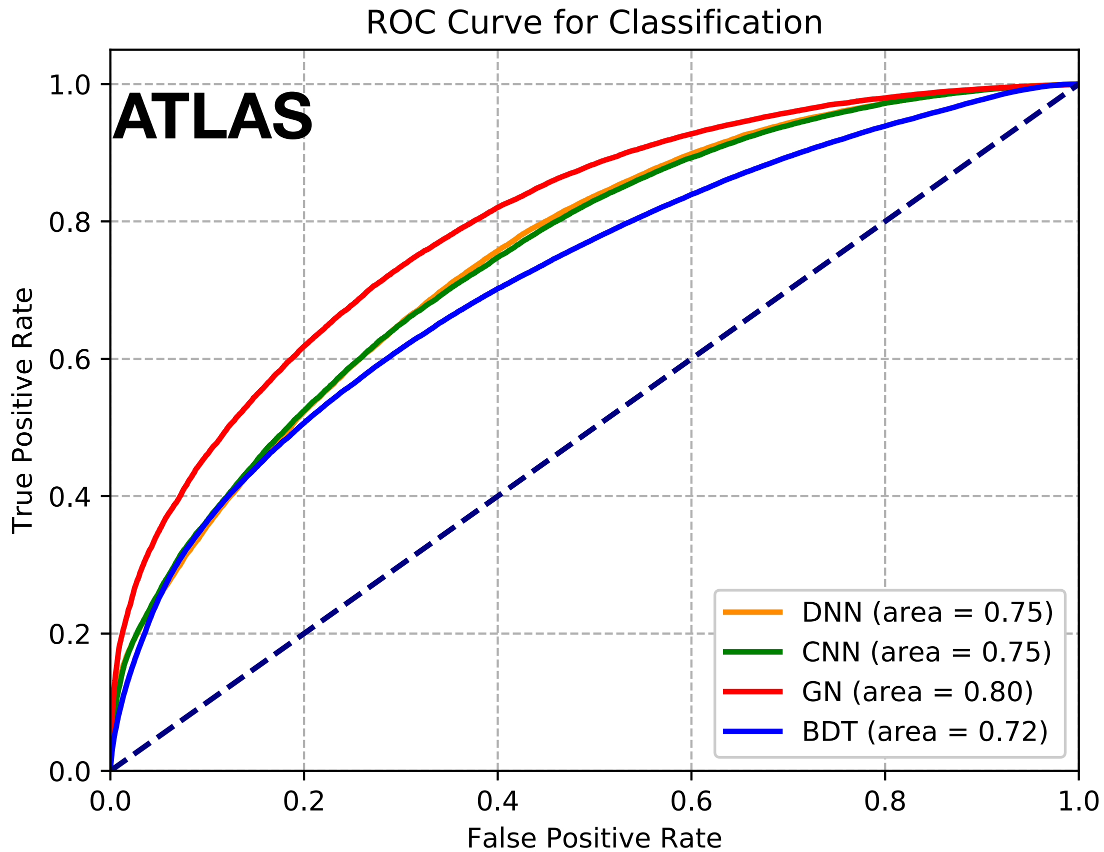


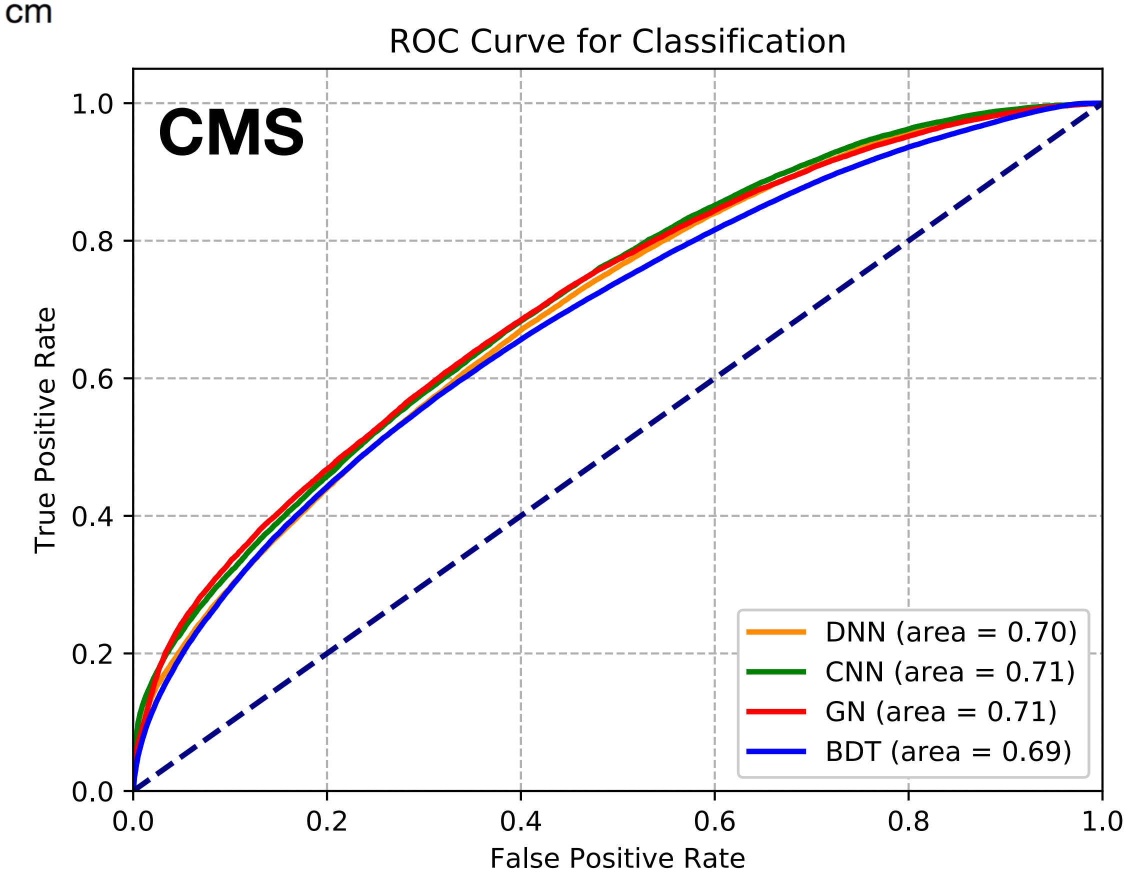


ATLAS / CMS

Detector Geometry:

CLIC:
$\Delta\eta \times \Delta\phi$: 0.003 * 0.003
Image size: 25 * 25* 25
Material: Tungsten
Radiation length: 0.3504 cm
Moliere radius: 0.9327 cm





ATLAS :
1st layer:
$\Delta\eta \times \Delta\phi$: 0.025/8 * 0.1
Image size: 25 * 1
2nd layer:
$\Delta\eta \times \Delta\phi$: 0.025 * 0.025
Image size: 3 * 3
3rd layer:
$\Delta\eta \times \Delta\phi$: 0.5 * 0.025
Image size: 2 * 3
Material: Liquid argon
Radiation length: 14 cm
Moliere radius: 9.043 cm

CMS:
$\Delta\eta \times \Delta\phi$: 0.0175 * 0.0175
Image size: 5 * 5
Material: Lead tungstate
Radiation length: 0.8903 cm
Moliere radius: 1.959 cm

Hyper-parameter

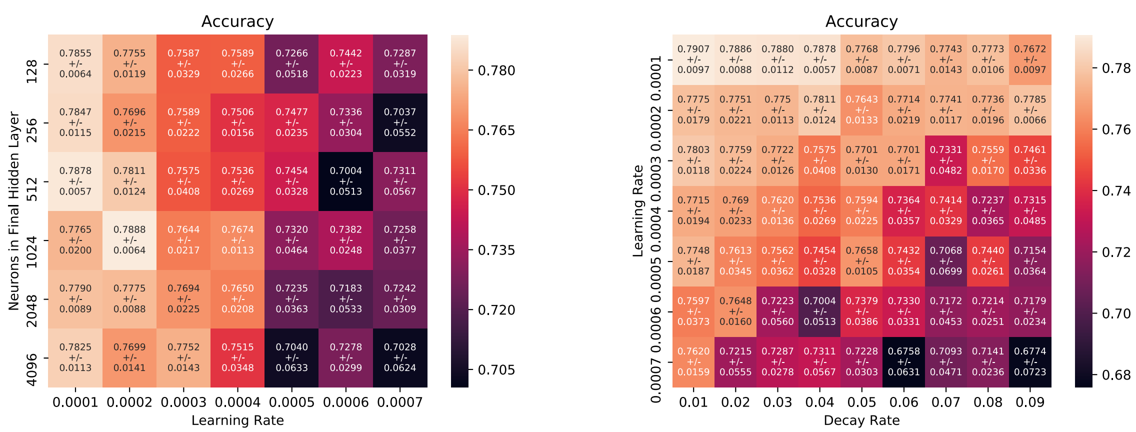


Figure 3: Hyperparameter scan results for GN. Scanning over number of hidden layers vs. learning rate (left) and learning rate vs. decay rate (right).

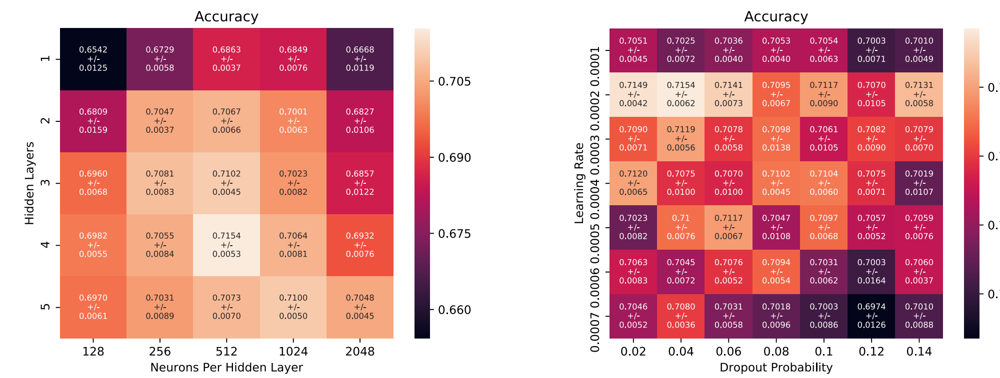


Figure 1: Hyperparameter scan results for DNN. Scanning over number of hidden layers vs. number of neurons per hidden layer (left) and learning rate vs. dropout probability (right).

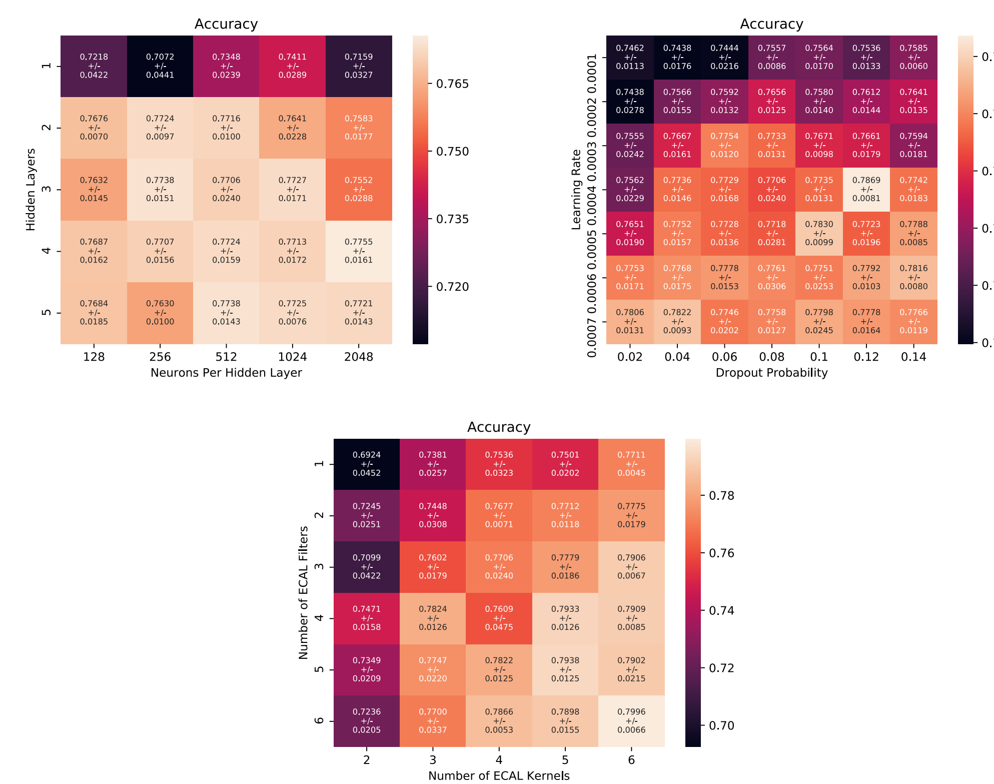


Figure 2: Hyperparameter scan results for CNN. Scanning over number of hidden layers vs. number of neurons per hidden layer (top left), learning rate vs. dropout probability (top right), and number of ECAL filters vs. number of ECAL kernels (bottom).

- Ran on Bluewaters and my UTA DL Cluster

Regression

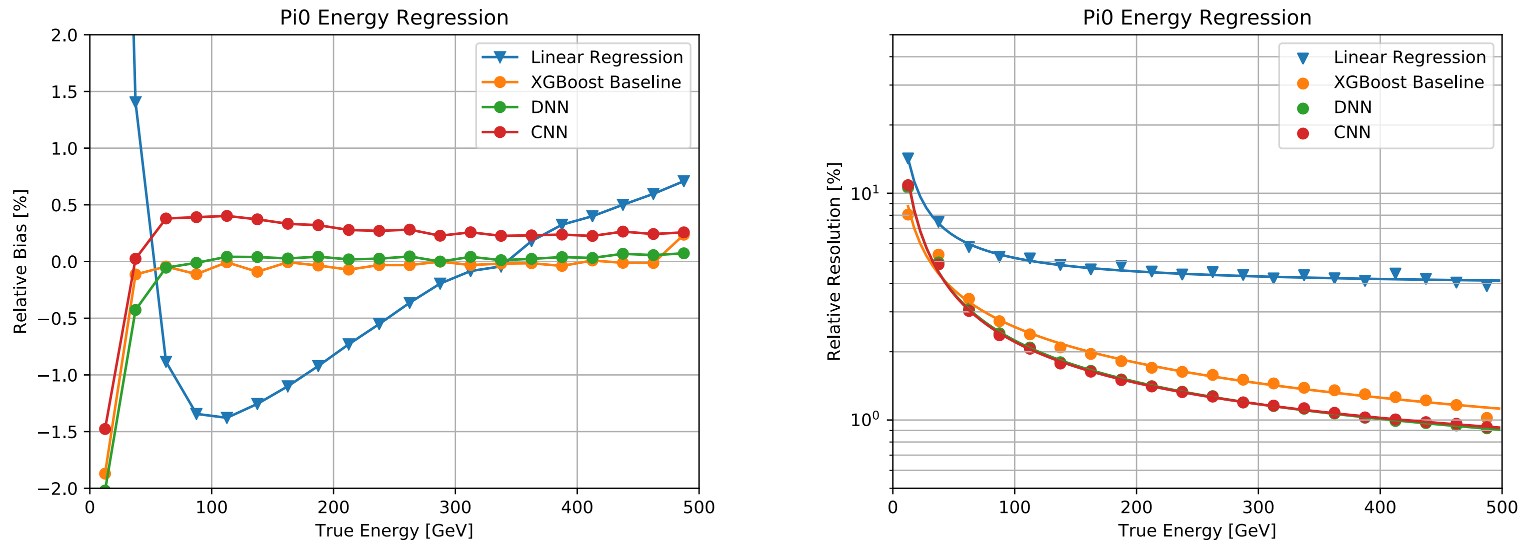


Figure 15: (Left) Bias and (right) resolution as a function of true energy for energy predictions for π^0, on fixed-angle samples.

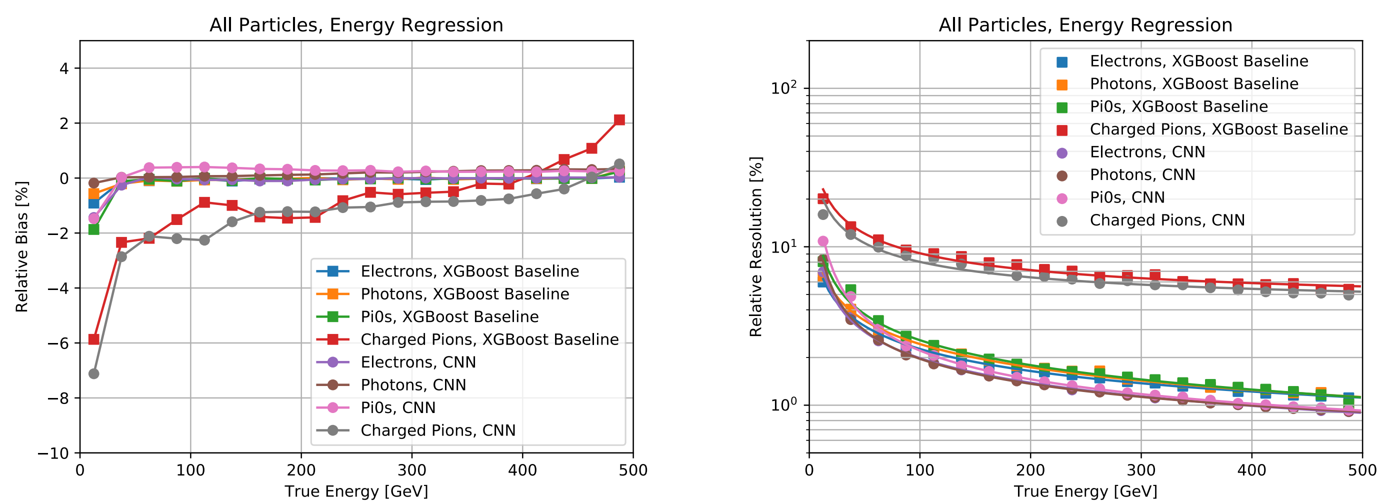


Figure 16: (Left) Bias and (right) resolution as a function of true energy for energy predictions for all particles, comparing the XGBoost baseline with the best CNN model, on fixed-angle samples.

Simulation

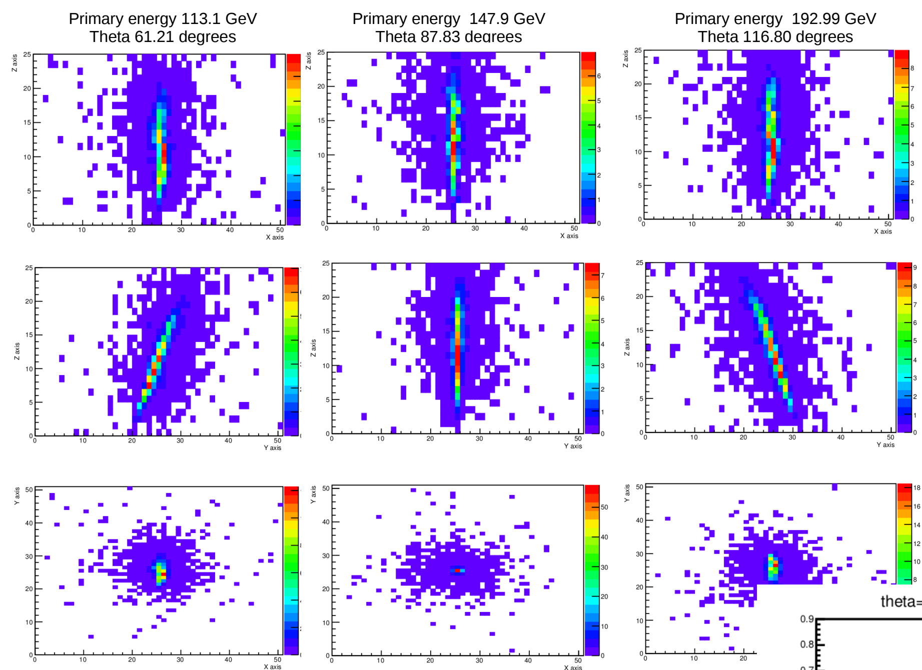
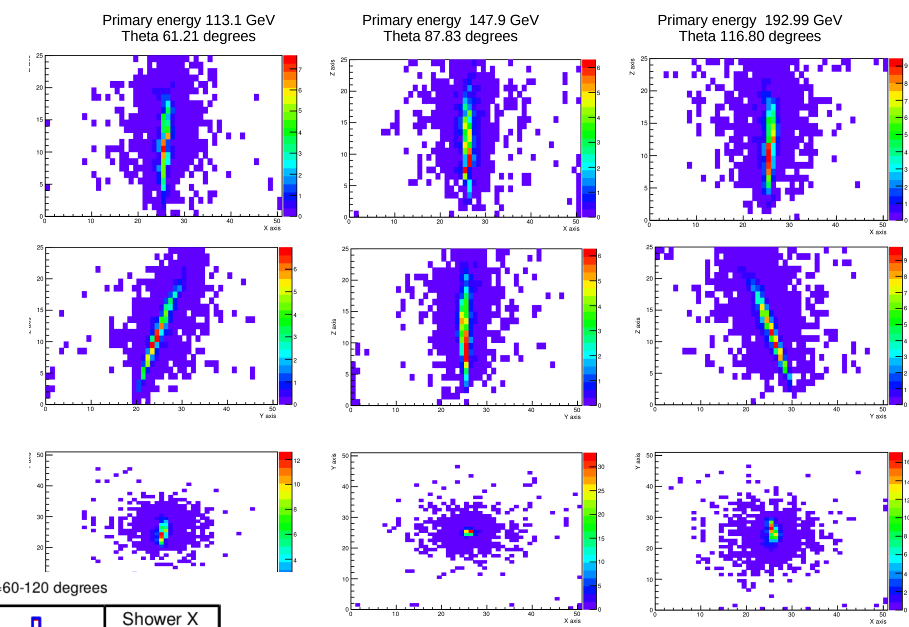


Figure 17: GEANT4 events with different primary particle energ



ted events with similar primary energies and angles.

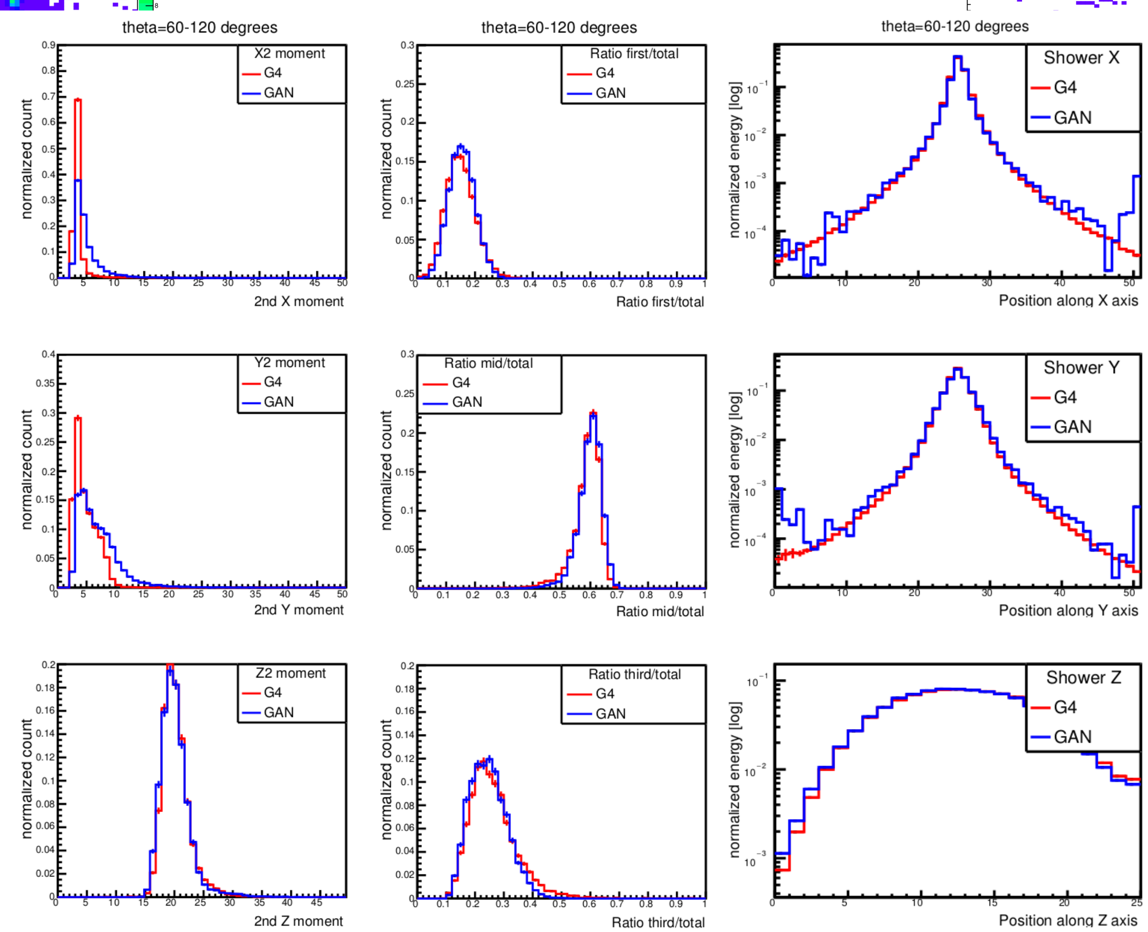
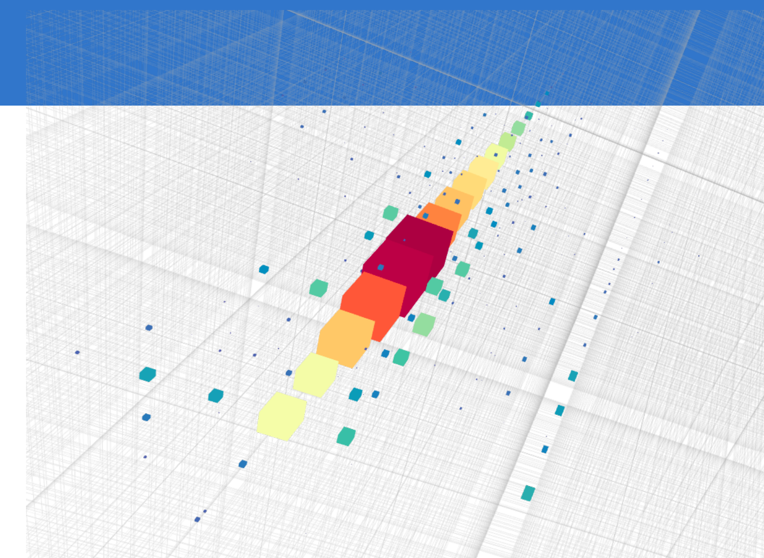


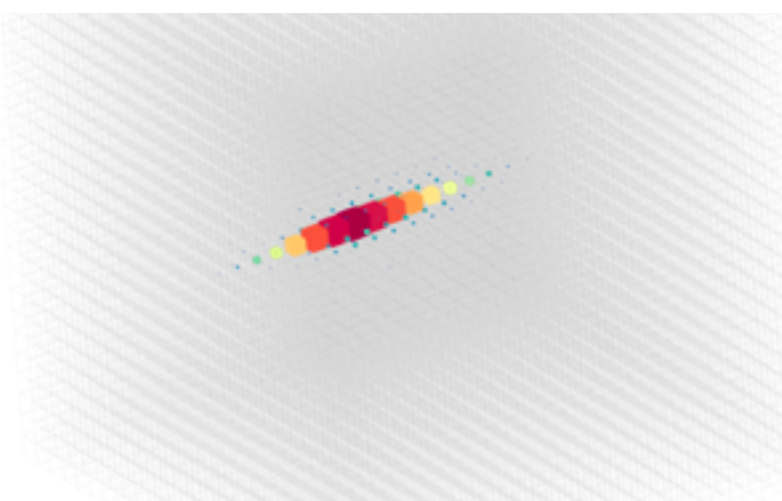
Figure 20: Geant4 vs. GAN comparison for shower width (second moment) in x,y,z, ratio of energy deposited in parts along direction of particle traversal to total energy and shower shapes along x,y,z axis in log scale for 100-200 GeV primary particle energies and 60-120 degrees theta

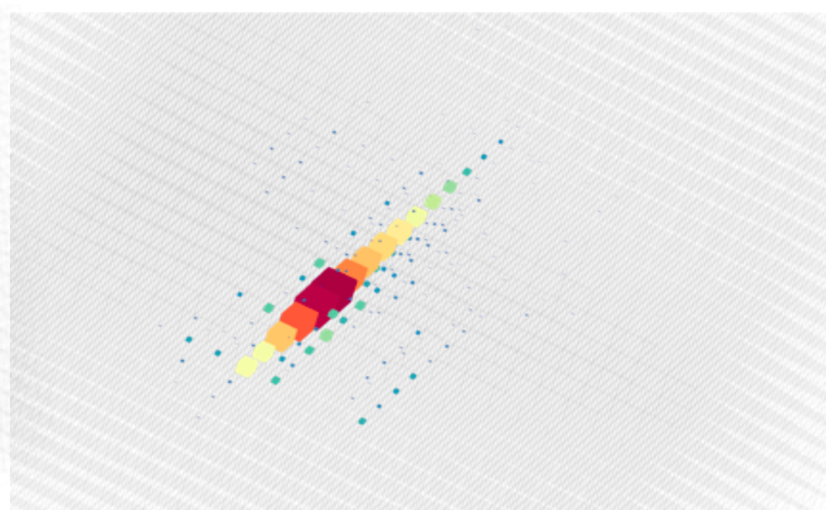
Preliminary

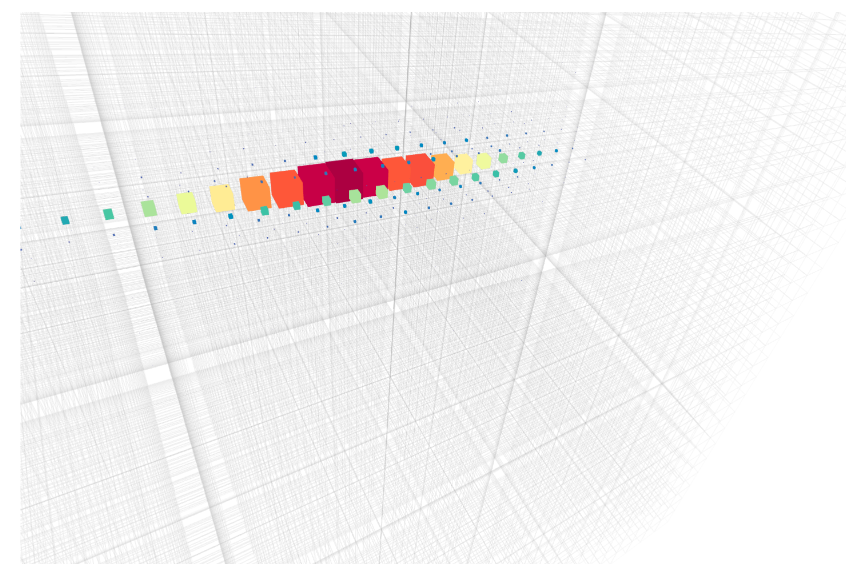
Some images



- Slice energy spectrum
- Start with photons & electrons







Training and HP Optimization on Titan and Summit in progress

Anomaly Detection

Variational Autoencoders for New Physics Mining a
the Large Hadron Collider

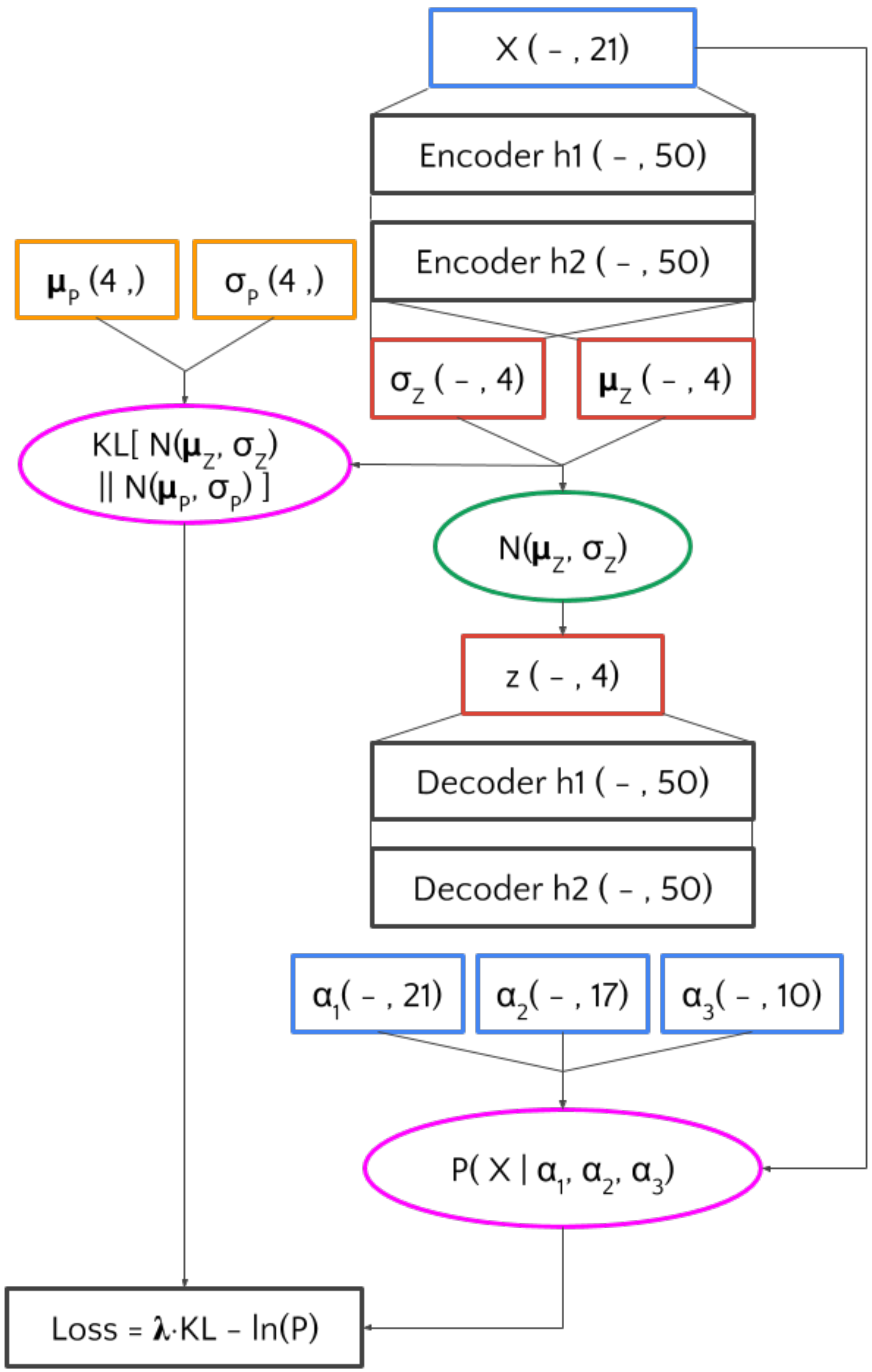
Olmo Cerri[1], Thong Q. Nguyen[1], Maurizio Pierini[2], Maria Spiropulu[1], a
Jean-Roch Vlimant[1]

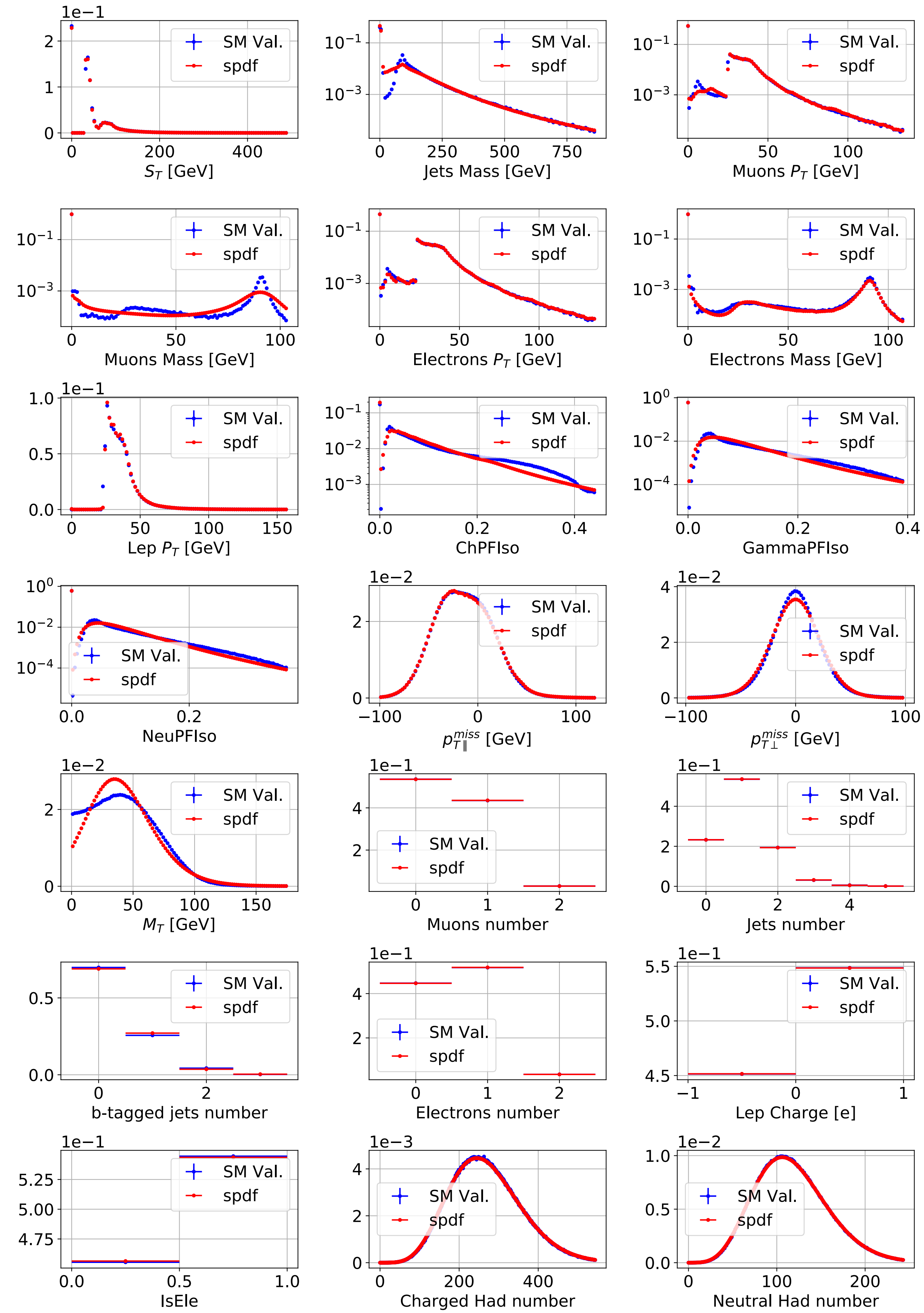
[1]California Institute of Technology, 1200 E. California Blvd, Pasadena, CA 91
[2]CERN, CH-1211 Geneva, Switzerland

Abstract

Using variational autoencoders trained on known physics processes, we develop a one-side p-value test to isolate previously unseen processes as outlier events. Since the autoencoder training does not depend on any specific new physics signature, the proposed procedure has a weak dependence on underlying assumptions about the nature of new physics. An event selection based on this algorithm would be complementary to classic LHC searches, typically based on model-dependent hypothesis testing. Such an algorithm would deliver a list of anomalous events, that the experimental collaborations could further scrutinize and even release as a catalog, similarly to what is typically done in other scientific domains. Repeated patterns in this dataset could motivate new scenarios for beyond-the-standard-model physics and inspire new searches, to be performed on future data with traditional supervised approaches. Running in the trigger system of the LHC experiments, such an application could identify anomalous events that would be otherwise lost, extending the scientific reach of the LHC.

https://arxiv.org/pdf/1811.10276.pdf


X (– , 21)
Encoder h1 (– , 50)
Encoder h2 (– , 50)
μ_p (4 ,)
σ_p (4 ,)
σ_z (– , 4)
μ_z (– , 4)
KL[$N(\mu_z, \sigma_z)$ || $N(\mu_p, \sigma_p)$]
$N(\mu_z, \sigma_z)$
z (– , 4)
Decoder h1 (– , 50)
Decoder h2 (– , 50)
α_1(– , 21)
α_2(– , 17)
α_3(– , 10)
P(X | α_1, α_2, α_3)
Loss = λ·KL – ln(P)


SM Val.
spdf
S_T [GeV]
Jets Mass [GeV]
Muons P_T [GeV]
Muons Mass [GeV]
Electrons P_T [GeV]
Electrons Mass [GeV]
Lep P_T [GeV]
ChPFIso
GammaPFIso
NeuPFIso
$p_{T\parallel}^{miss}$ [GeV]
$p_{T\perp}^{miss}$ [GeV]
M_T [GeV]
Muons number
Jets number
b-tagged jets number
Electrons number
Lep Charge [e]
IsEle
Charged Had number
Neutral Had number

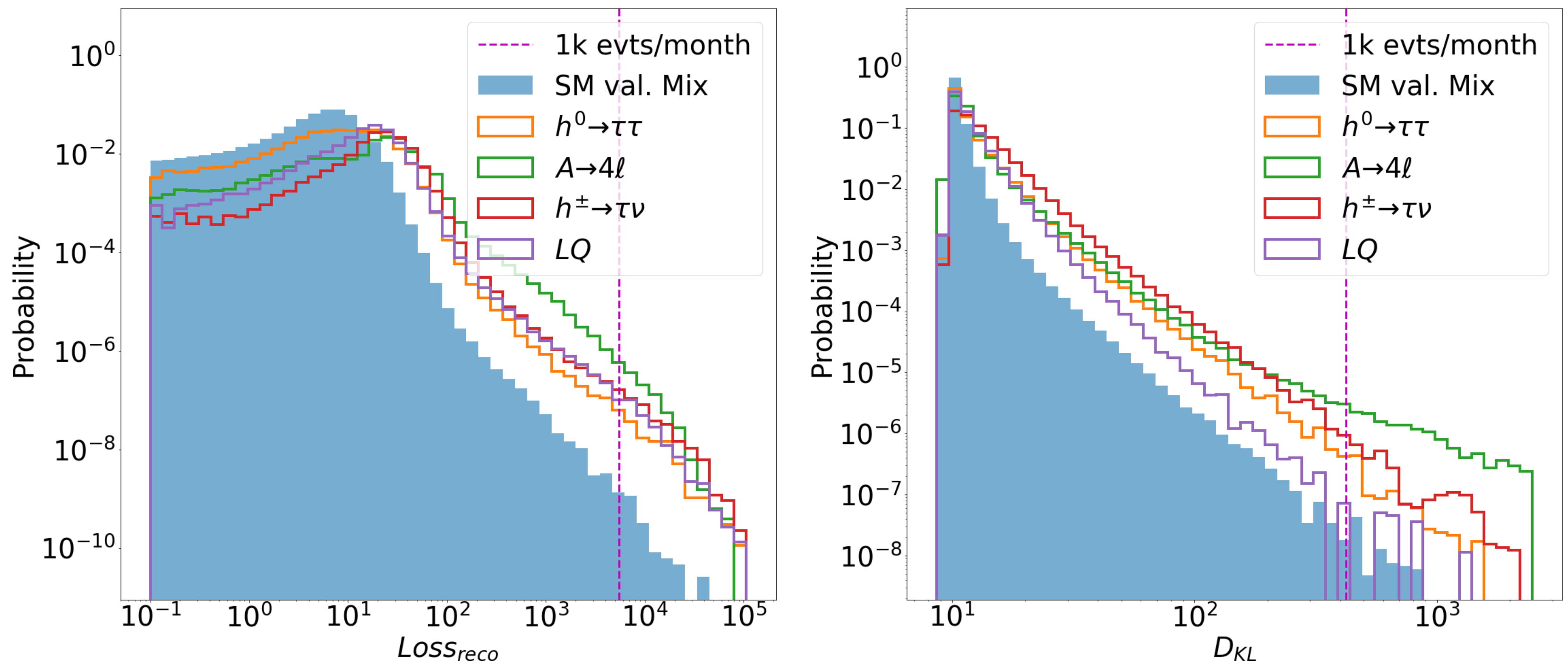


Figure 7: Distribution of the loss components: $Loss_{reco}$ (left) and D_{KL} (right) for the validation dataset. For comparison, the corresponding distribution for the SM processes and the four benchmark BSM models are shown. The vertical line represents a lower threshold such that $5.4 \cdot 10^{-6}$ of the SM events would be retained, equivalent to ~ 1500 expected SM events per month.

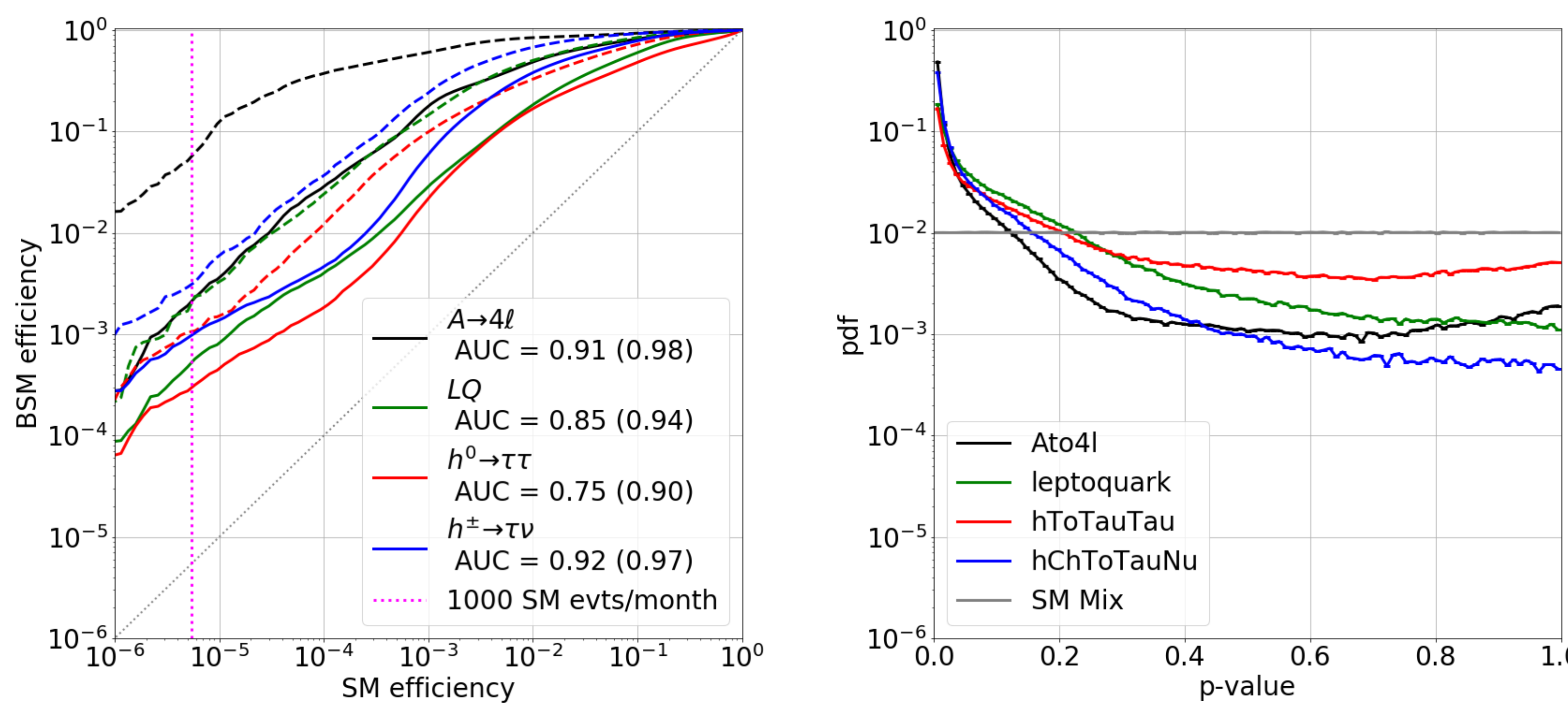


Figure 8: Left:ROC curves for the VAE trained only on SM mix (solid), compared to the corresponding curves for the four supervised BDT models (dashed) described in Section 4.2. Right: p-value distribution for the SM cocktail events and the four BSM benchmark processes.

Event Generation

Event Generation and Statistical Sampling with Deep Generative Models and a Density Information Buffer

Sydney Otten,[1,2,*] Sascha Caron,[1,3,†] Wieske de Swart,[1] Melissa van Beekveld,[1] Luc Hendriks,[1] Caspar van Leeuwen,[4] Damian Podareanu,[4] Roberto Ruiz de Austri,[5] and Rob Verheyen[1]

[1]*Institute for Mathematics, Astro- and Particle Physics IMAPP*
Radboud Universiteit, Nijmegen, The Netherlands
[2]*GRAPPA, University of Amsterdam, The Netherlands*
[3]*Nikhef, Amsterdam, The Netherlands*
[4]*SURFsara, Amsterdam, The Netherlands*
[5]*Instituto de Fisica Corpuscular, IFIC-UV/CSIC*
University of Valencia, Spain

(Dated: March 14, 2019)

We present a study for the generation of events from a physical process with generative deep learning. To simulate physical processes it is not only important to produce physical events, but also to produce the events with the right frequency of occurrence (density). We investigate the feasibility to learn the event generation and the frequency of occurrence with Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs) to produce events like Monte Carlo generators. We study three toy models from high energy physics, i.e. a simple two-body decay, the processes $e^+e^- \to Z \to l^+l^-$ and $pp \to t\bar{t}$ including the decay of the top quarks and a simulation of the detector response. We show that GANs and the standard VAE do not produce the right distributions. By buffering density information of Monte Carlo events in latent space given the encoder of a VAE we are able to construct a prior for the sampling of new events from the decoder that yields distributions that are in very good agreement with real Monte Carlo events and are generated $\mathcal{O}(10^8)$ times faster. Applications of this work include generic density estimation and sampling, targeted event generation via a principal component analysis of encoded events in the latent space and the possibility to generate better random numbers for importance sampling, e.g. for the phase space integration of matrix elements in quantum perturbation theories. The method also allows to build event generators directly from real data events.

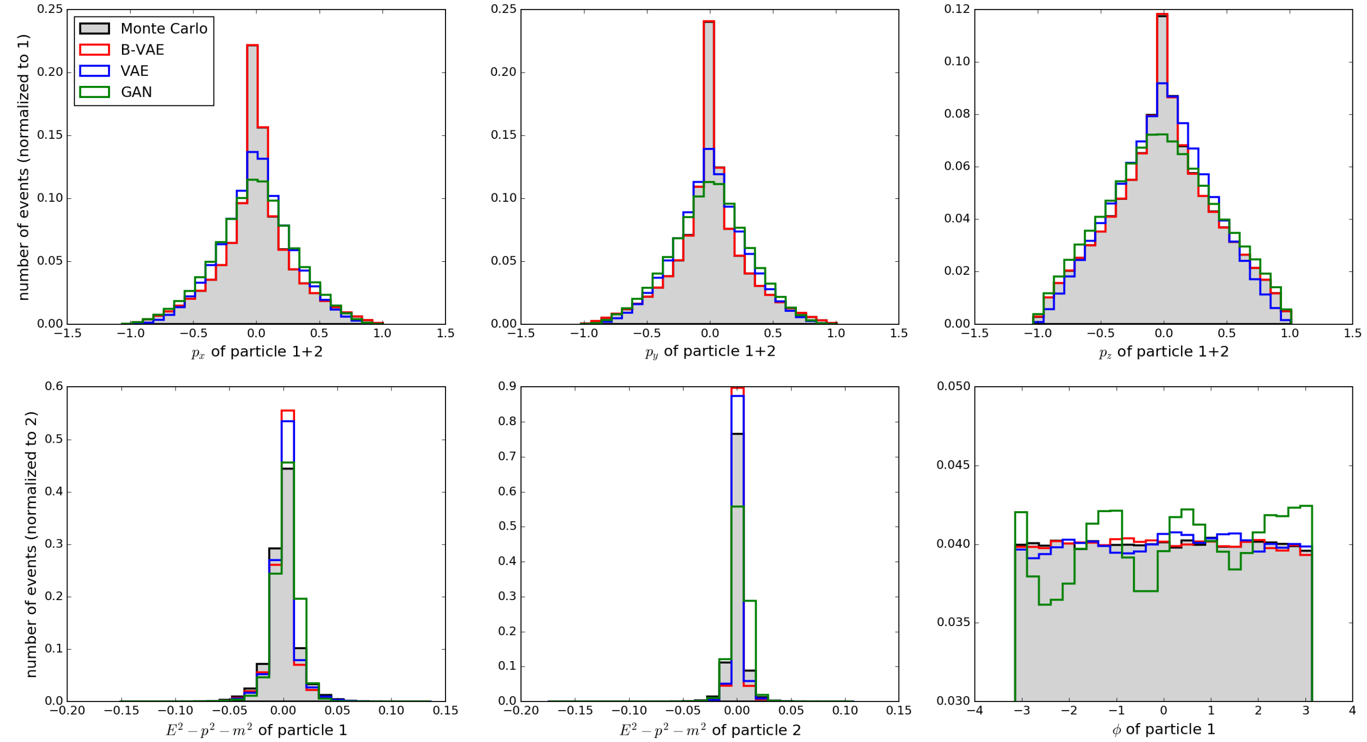


FIG. 1. Events that are generated by a Monte Carlo generator for a toy two-body decay (gray), by the VAE with a standard normal prior (blue line), by the B-VAE with a buffering of density information in the latent space (red line) and by the GAN (green line). The top line shows the distributions for p_x, p_y, p_z of particle 1 + 2. The bottom line shows $E^2 - p^2 - m^2$ for particle 1 and 2 and the distribution for the azimuthal angle ϕ of particle 1.

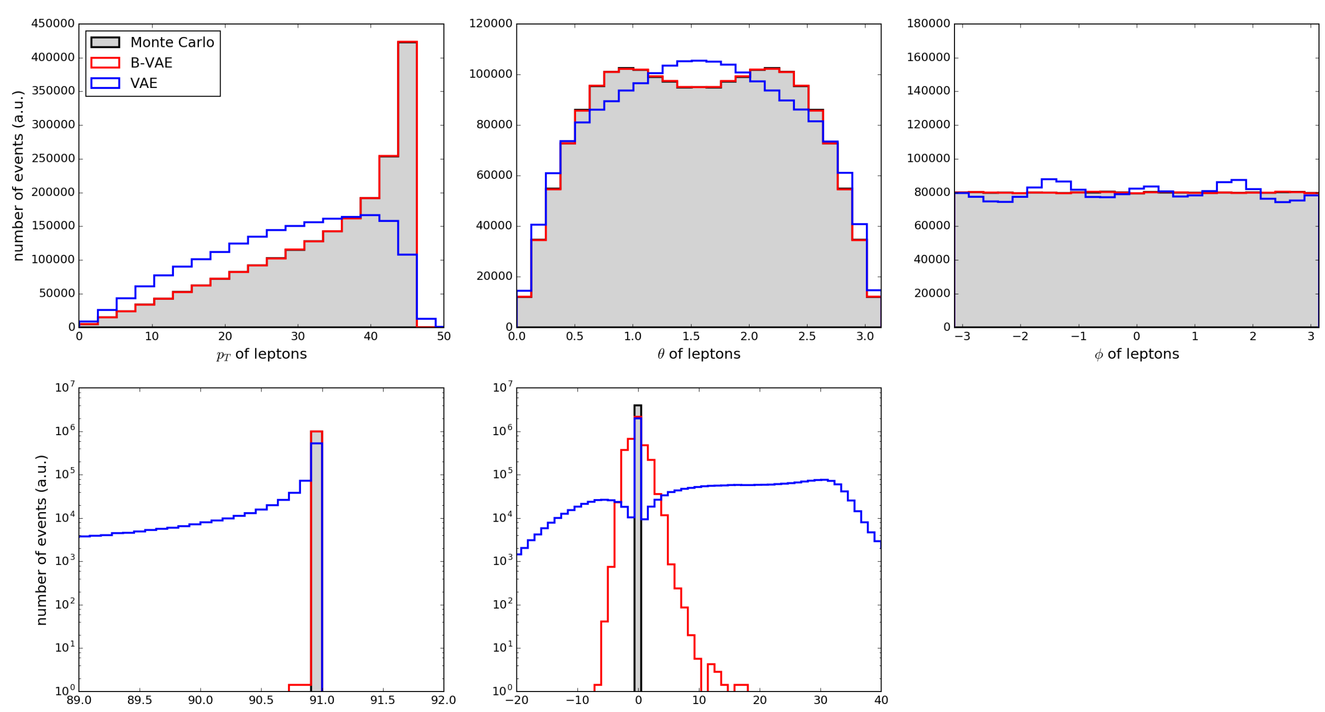
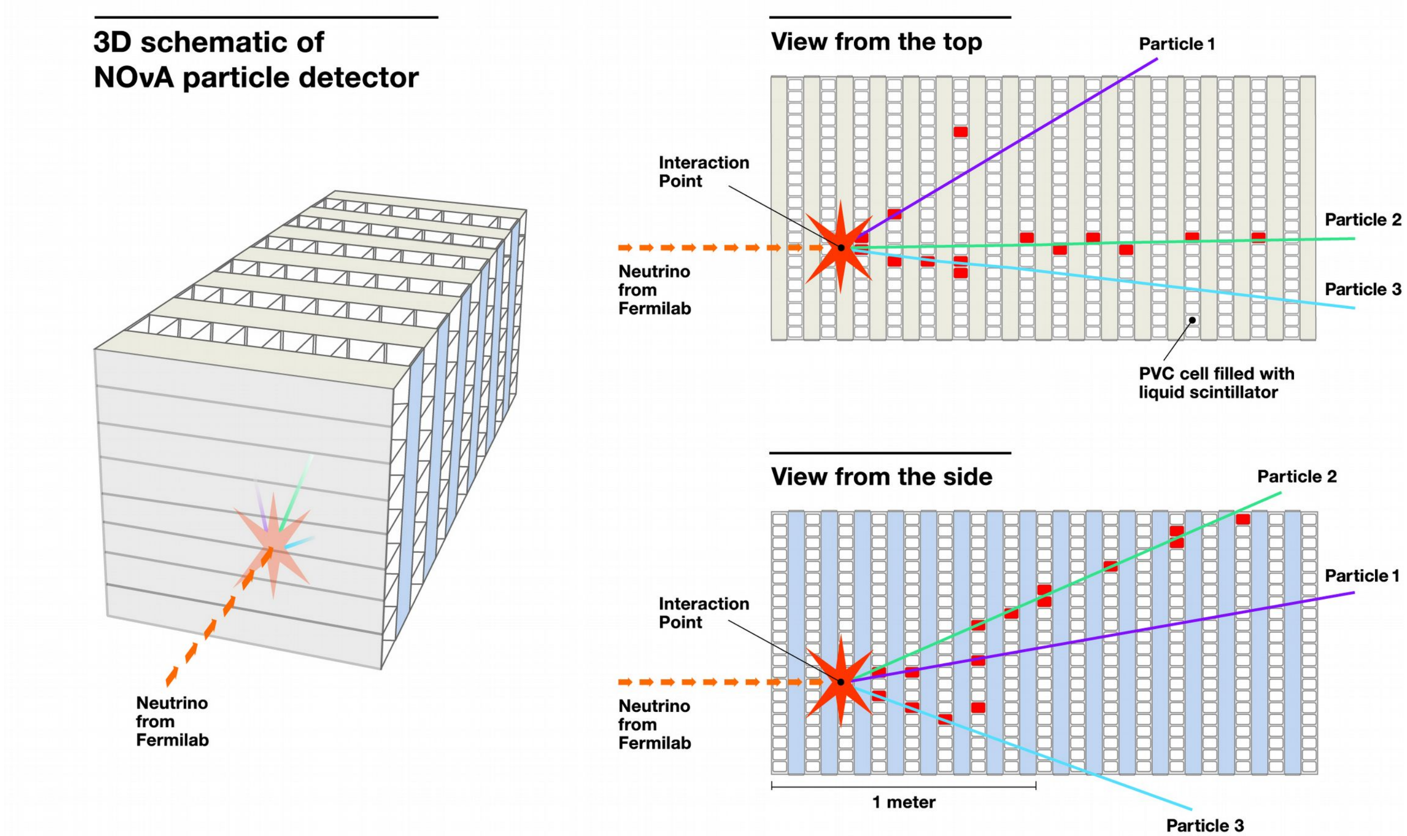


FIG. 3. Events that are generated by a Monte Carlo generator for the $e^+e^- \to Z \to l^+l^-$ process (gray), by the VAE with a standard normal prior (blue line) and by the B-VAE with a buffering of density information in the latent space (red line). The top line shows the lepton p_T, θ and ϕ. The p_T is shown in GeV. The bottom line shows the invariant mass of the lepton pair (which should be the mass of the Z-boson) and the invariant mass of the leptons themselves (which should be 0 GeV, the mass of the leptons during generation). The number of events is normalized to the number of generated Monte Carlo events.

Neutrino Physics

Neutrino Physics

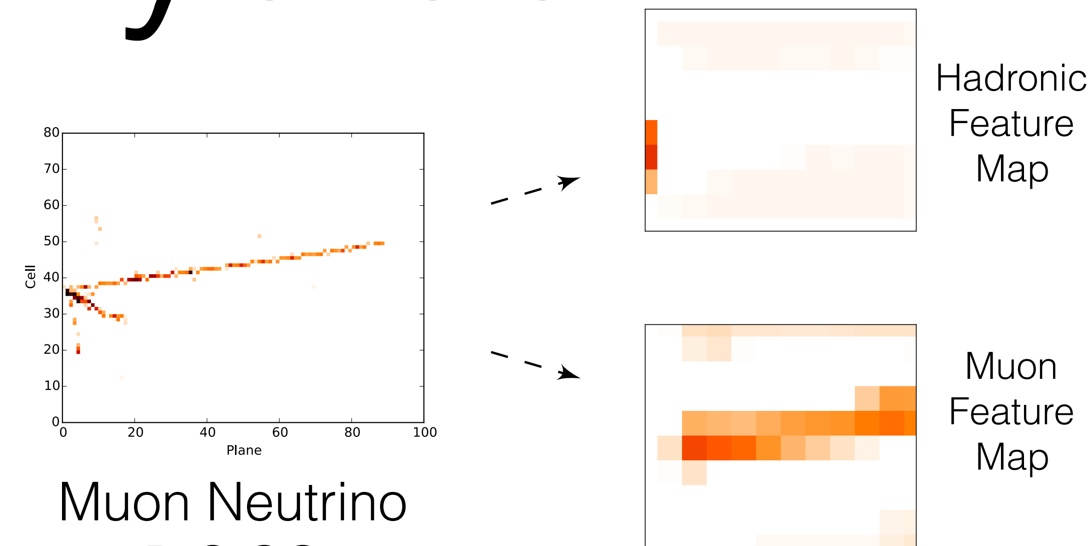
- Core Physics requires just measuring ***neutrino flavor and energy***.
- Generally clean (low multiplicity) and high granularity.
- ***First HEP CNN application: Nova*** using Siamese Inception CNN.



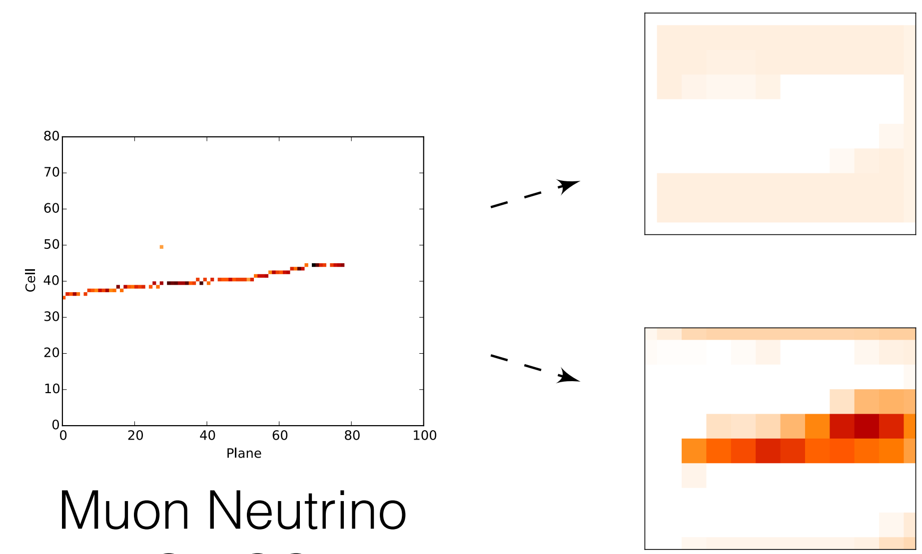
	CVN Selection Value	ν_e sig	Tot bkg	NC	ν_μ CC	Beam ν_e	Signal Efficiency	Purity
Contained Events	–	88.4	509.0	344.8	132.1	32.1	–	14.8%
$s/\sqrt{b}$ opt	0.94	43.4	6.7	2.1	0.4	4.3	49.1%	86.6%
$s/\sqrt{s+b}$ opt	0.72	58.8	18.6	10.3	2.1	6.1	66.4%	76.0%

	CVN Selection Value	ν_μ sig	Tot bkg	NC	Appeared ν_e	Beam ν_e	Signal Efficiency	Purity
Contained Events	–	355.5	1269.8	1099.7	135.7	34.4	–	21.9%
$s/\sqrt{b}$ opt	0.99	61.8	0.1	0.1	0.0	0.0	17.4%	99.9%
$s/\sqrt{s+b}$ opt	0.45	206.8	7.6	6.8	0.7	0.1	58.2%	96.4%

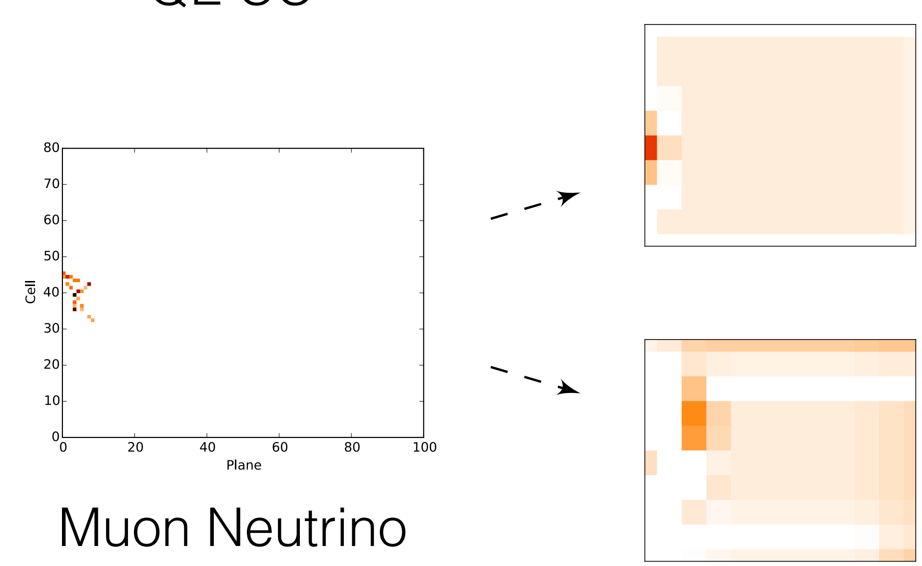
- ***40% Better Electron Efficiency for same background.***



Muon Neutrino
DIS CC



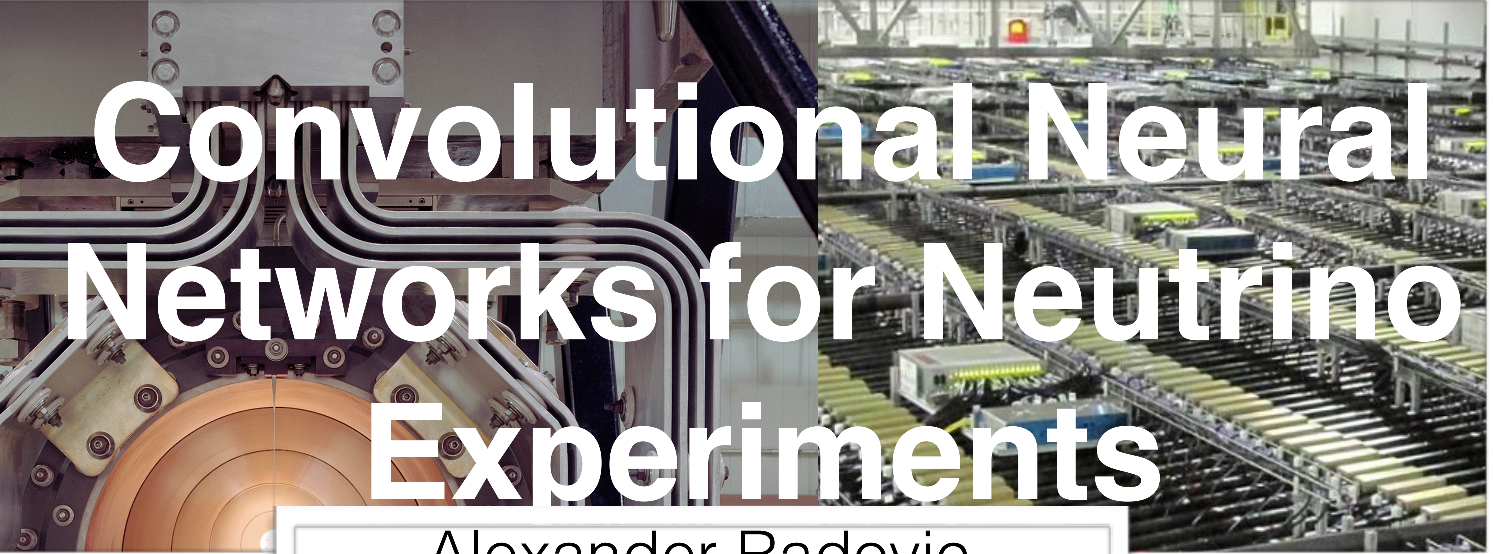
Muon Neutrino
QE CC



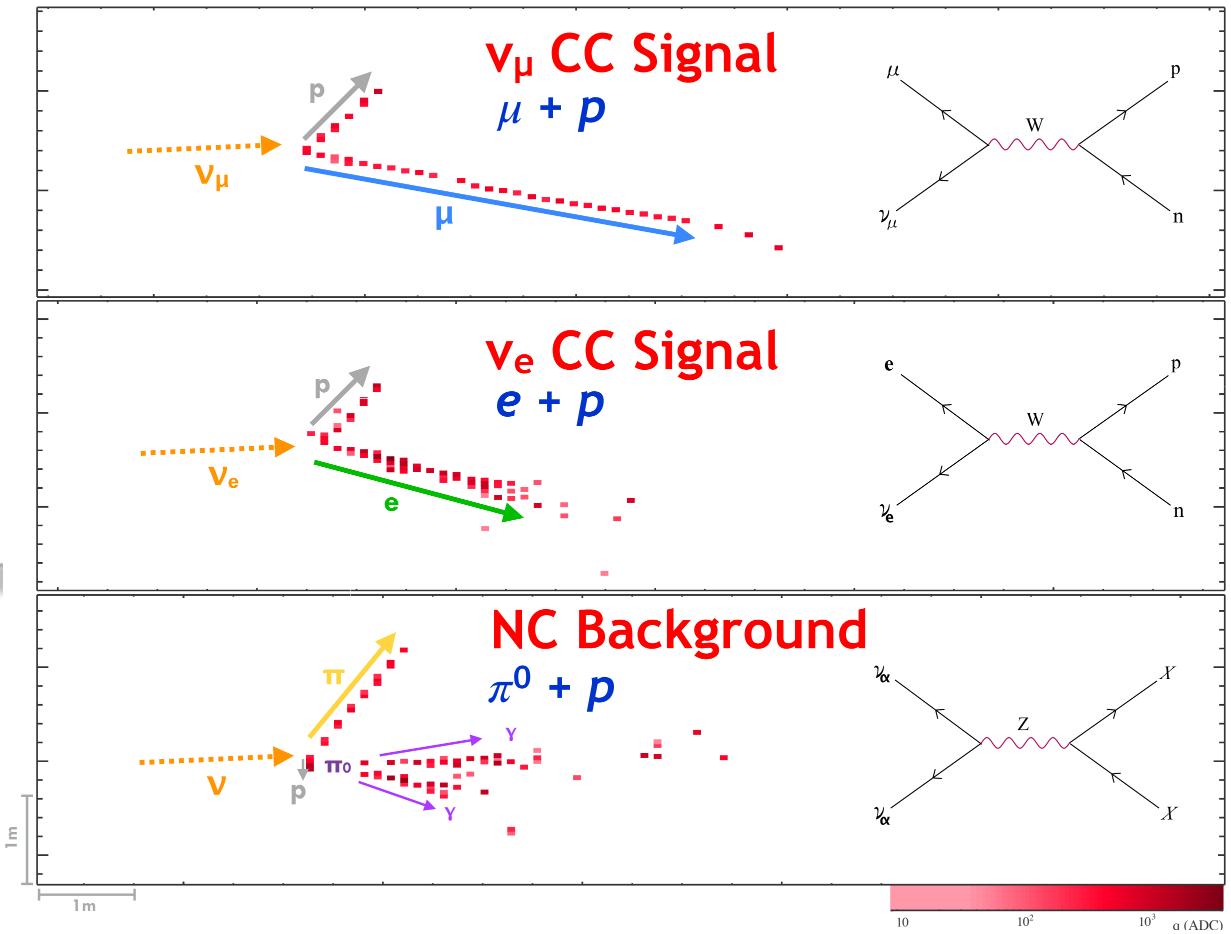
Muon Neutrino
NC

http://arxiv.org/pdf/1604.01444.pdf

Convolutional Neural Networks for Neutrino Experiments



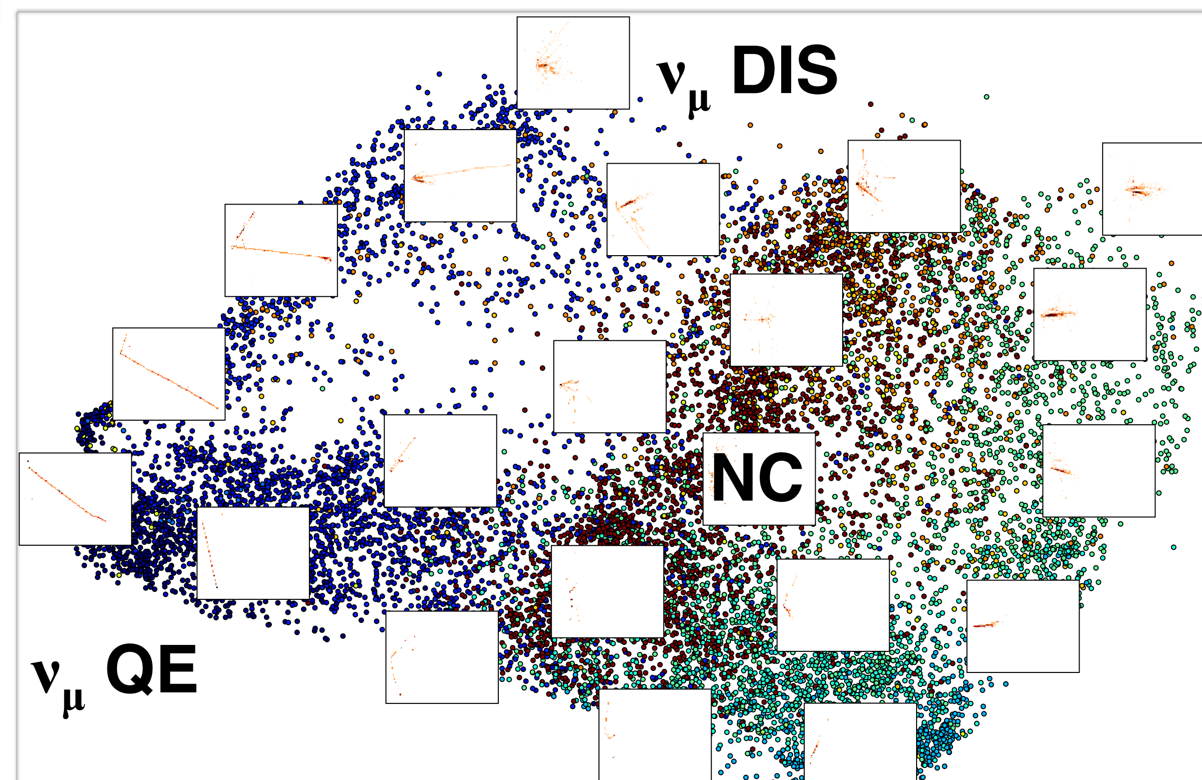
Alexander Radovic
College of William and Mary



Why Convolutional Neural Networks?

- That means that any oscillation analysis can benefit fror precise identification of the interaction in two ways:
  - Estimating the lepton flavor of the incoming neutrino.
  - Correctly identifying the type of neutrino interaction, t better estimate the neutrino energy, aka is it a quasi elastic event or a resonance event?

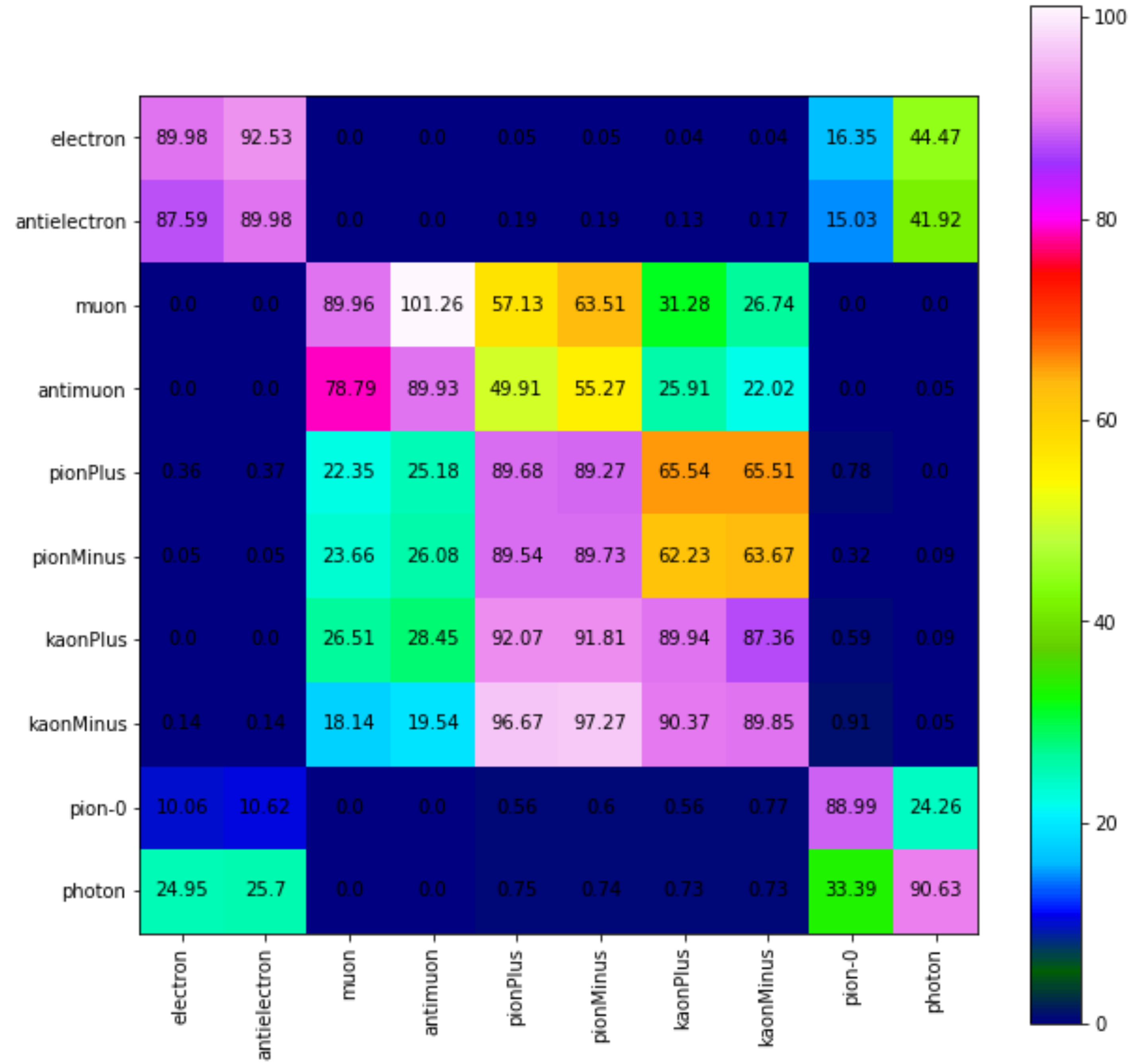
- Our detectors are also often the perfect domain:
  - Large ~uniform volumes where spatially invariant response is a benefit.
  - Usually only one or two detector systems.



t-SNE projection of final features to 2D.

However our CNN achieves **73%** efficiency and **76%** purity on **ν_e** selection at the $s/\sqrt{s+b}$ optimized cut.
Equivalent to **30%** more exposure with the old PIDs.

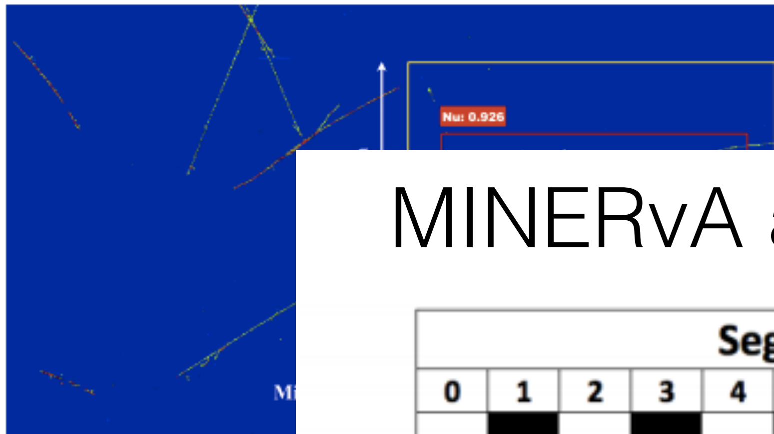
LArIAT: DNN vs Alg

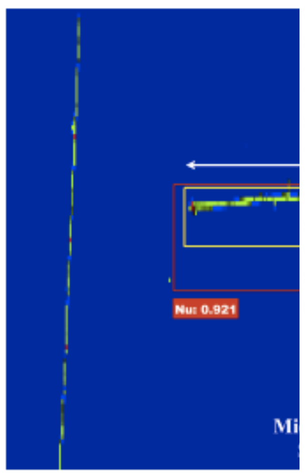


	$\pi+$	$\kappa+$	$\mu+$	$e+$	γ
DNN	74.42%	40.67%	6.37%	0.12%	0%
LArIAT	74.5%	68.8%	88.4%	6.8%	2.4%

	$\pi-$	$\kappa-$	$\mu-$	$e-$	γ
DNN	78.68%	54.47%	13.54%	0.11%	0.25%
LArIAT	78.7%	73.4%	91.0%	7.5%	2.4%

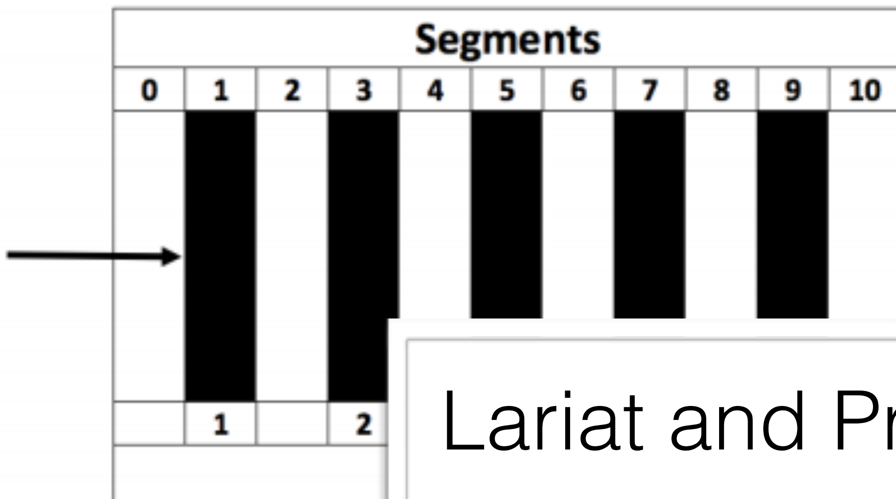
MicroBooNE

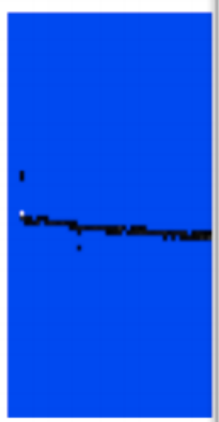




JINST 12 (2017) r

MINERvA and MENNDL

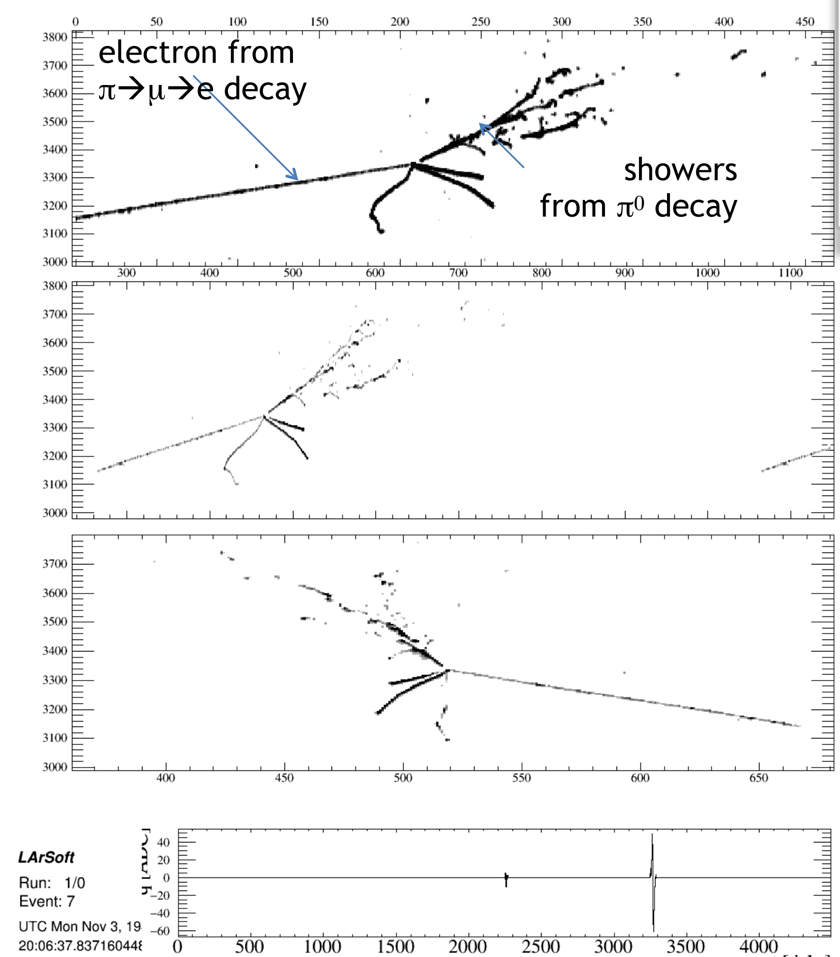




PhyStat-nu Fermila

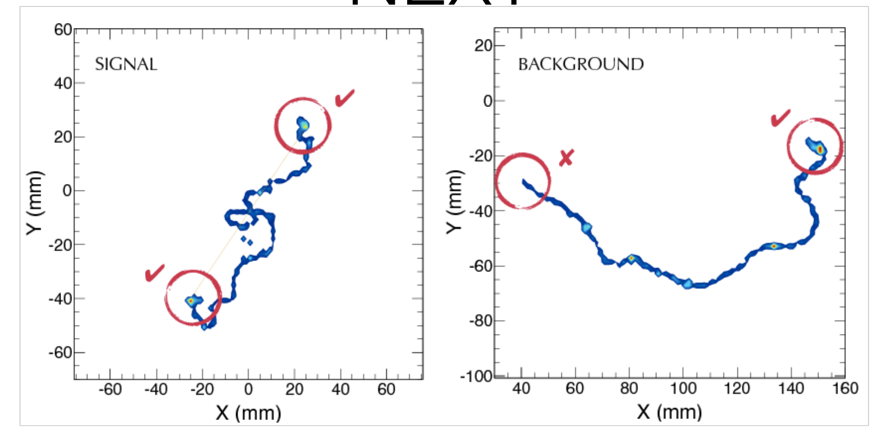
Lariat and ProtoDUNE

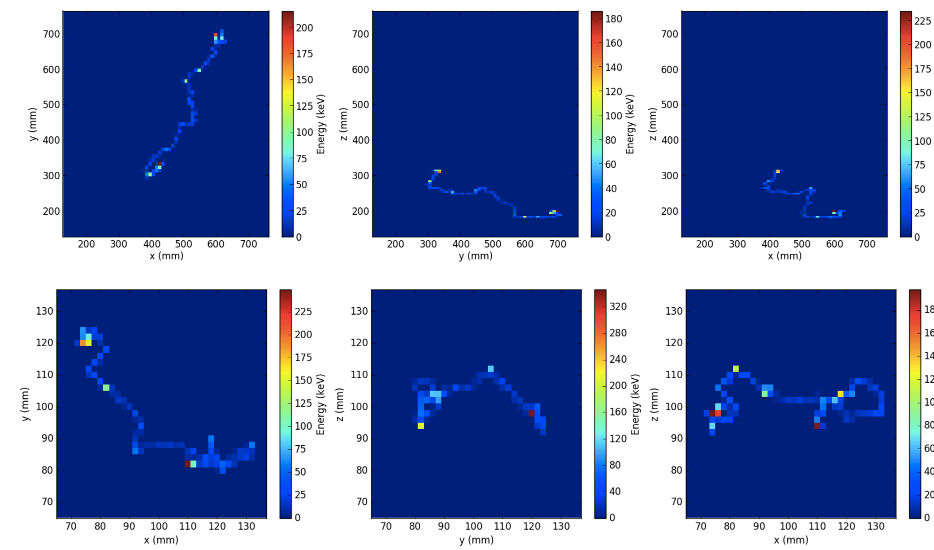
EM / hadronic component discrimination



Private communication, Robert Sutlej

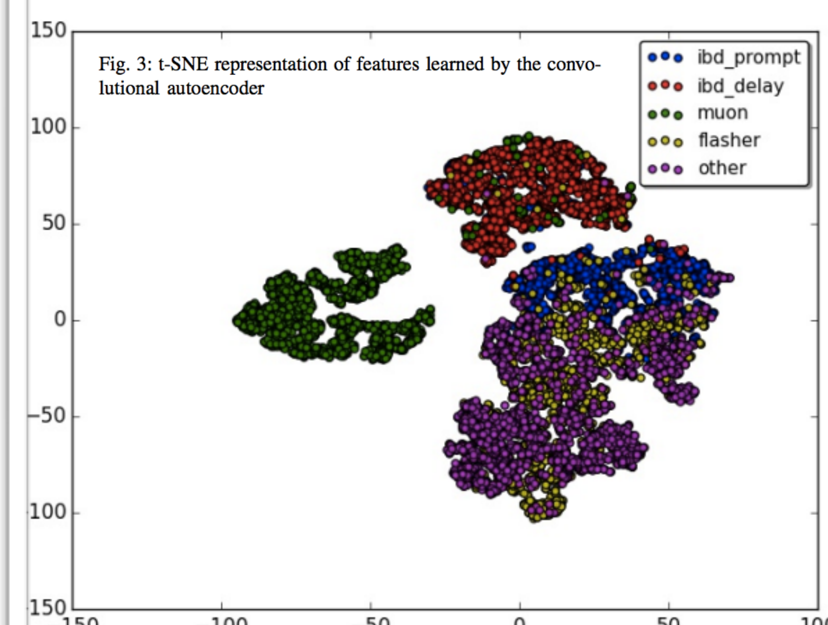
NEXT





JINST 12 (2017) no.01, T01004

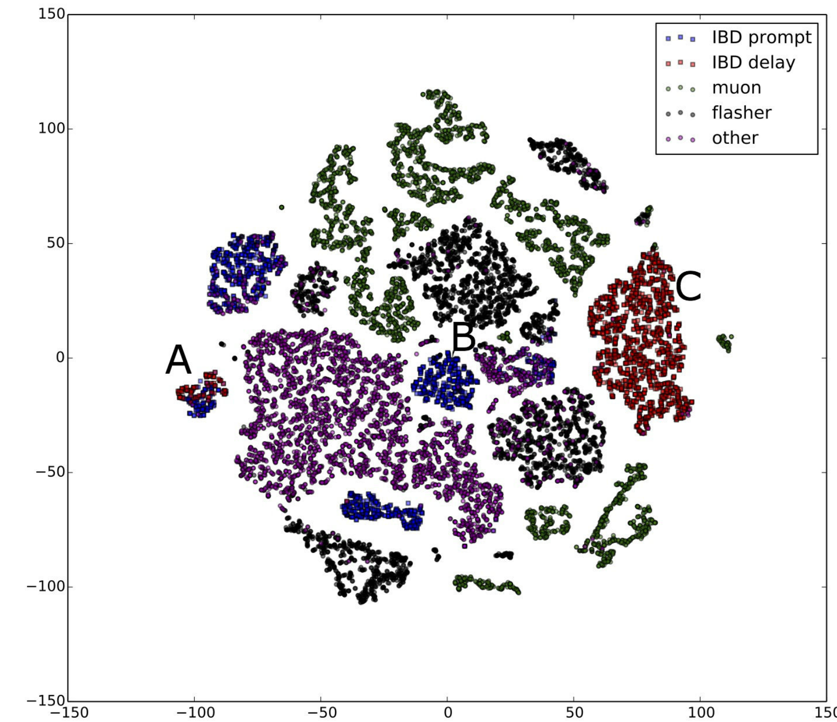
Daya Bay



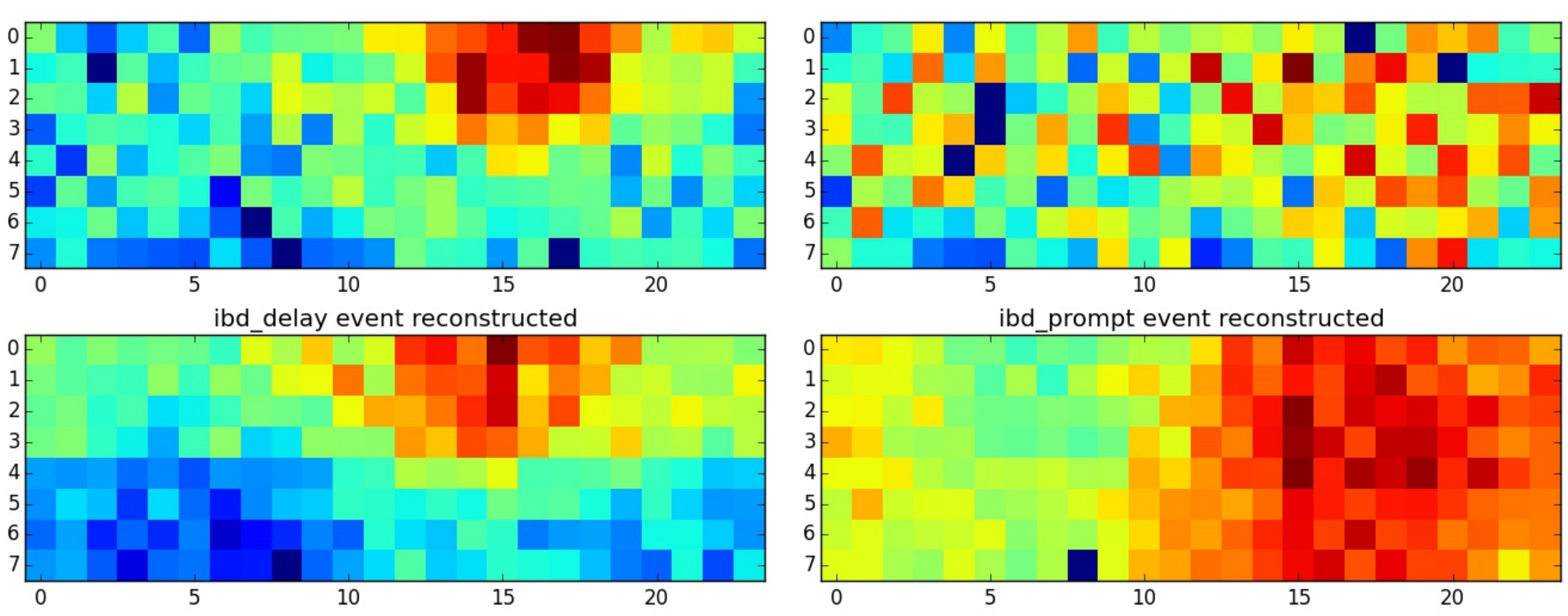
arXiv:1601.07621

Learning Representations

- Example: ***Daya Bay Experiment*** (*Evan Racah, et al*)
- Input: 8 x 24 PMT unrolled cylinder. ***Real Data (no simulation)***
- 2 Studies:
  - ***Supervised CNN Classifier***
    - Labels from standard analysis: Prompt/Delayed Inverse Beta Decay, Muon, Flasher, Other.
  - ***Convolutional Auto-encoder*** (semi-supervised)
    - Clearly separates muon and IBD delay ***without any physics knowledge***.
    - Potentially could have ID'ed problematic data (e.g. flashers) much earlier.

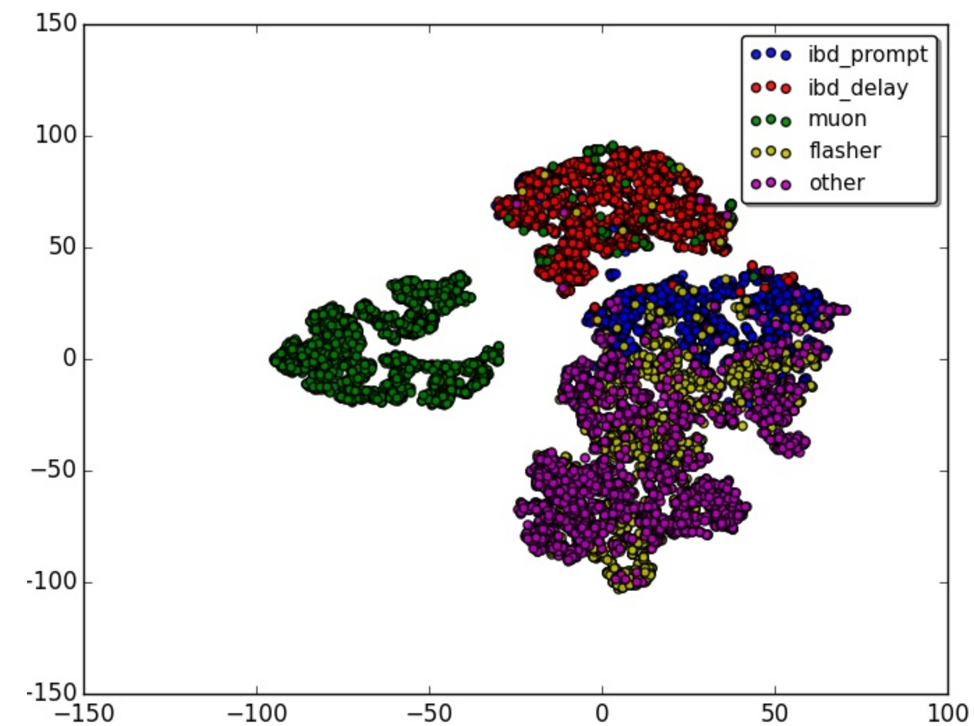


t-SNE reduction of 26-dim representation of the last fully connected layer.



(a) Example of an "IBD delay" event

(b) Example of an "IBD prompt" event



t-SNE reduction of 10 parameter latent representation.

Jet Physics with Deep Learning

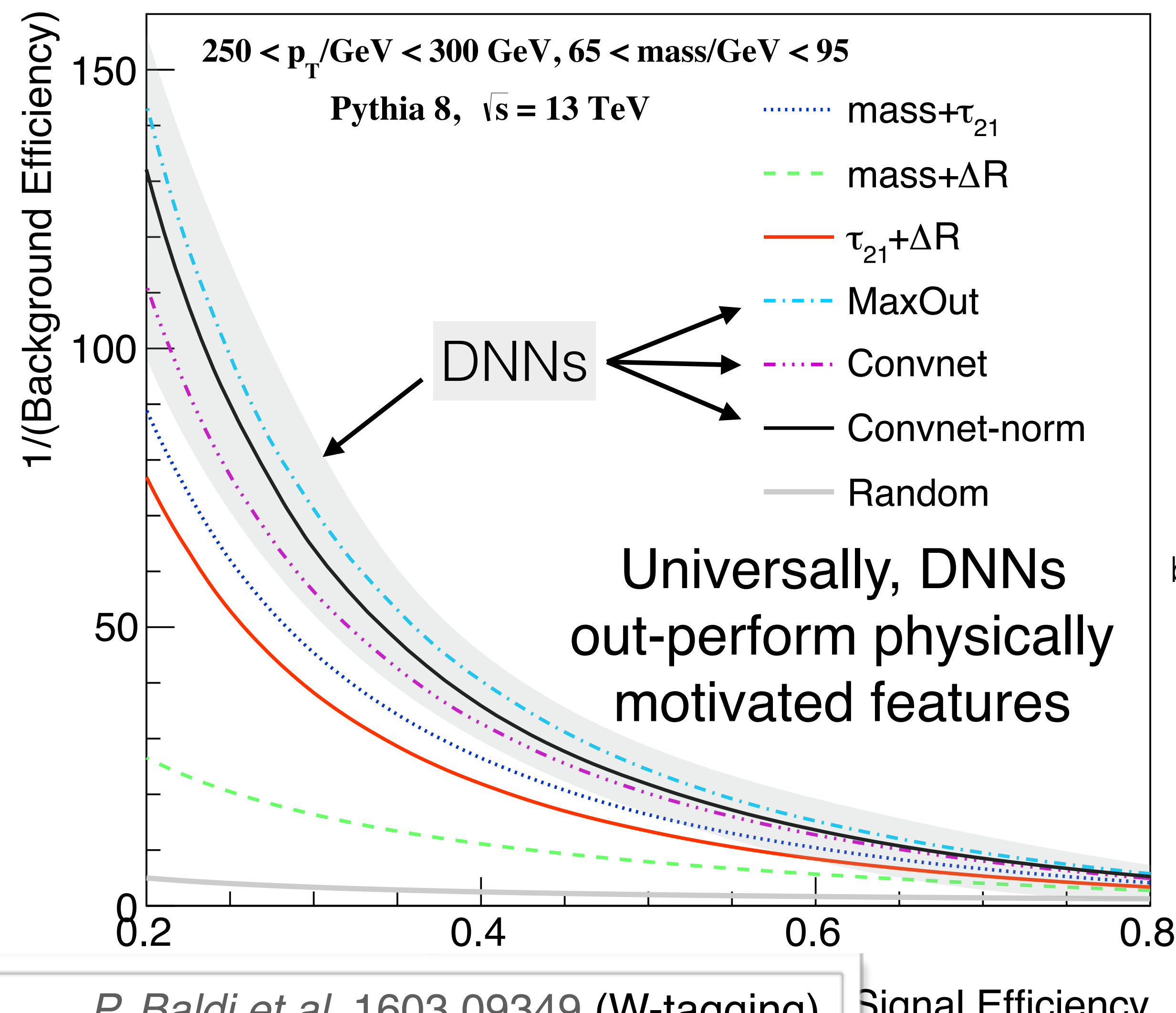
The Jet Image

Benjamin Nachman

Lawrence Berkeley National Laboratory

BERKELEY LAB

Modern Deep NN's for Classification



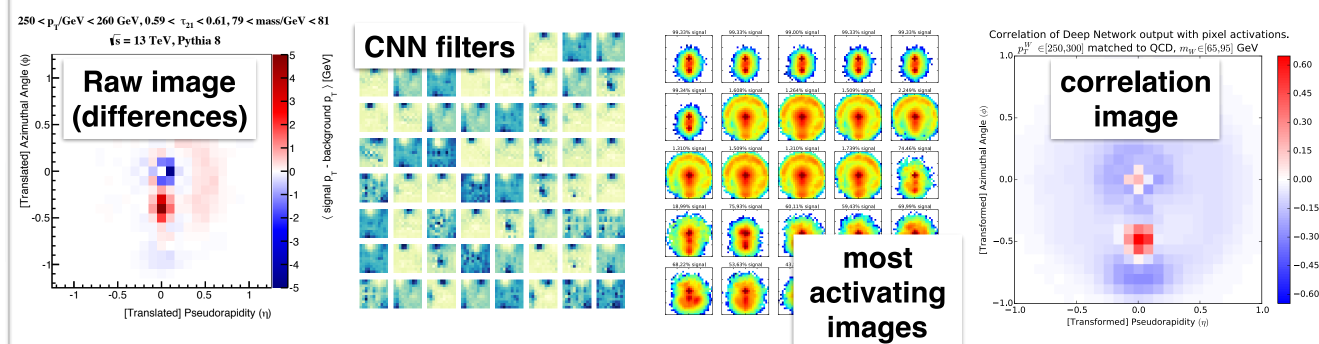
Universally, DNNs out-perform physically motivated features

P. Baldi et al. 1603.09349 (W-tagging)
J. Barnard et al. 1609.00607 (W-tagging)
P. Komiske et al. 1612.01551 (q/g-tagging)
G. Kasieczka et al. 1701.08784 (top-tagging)

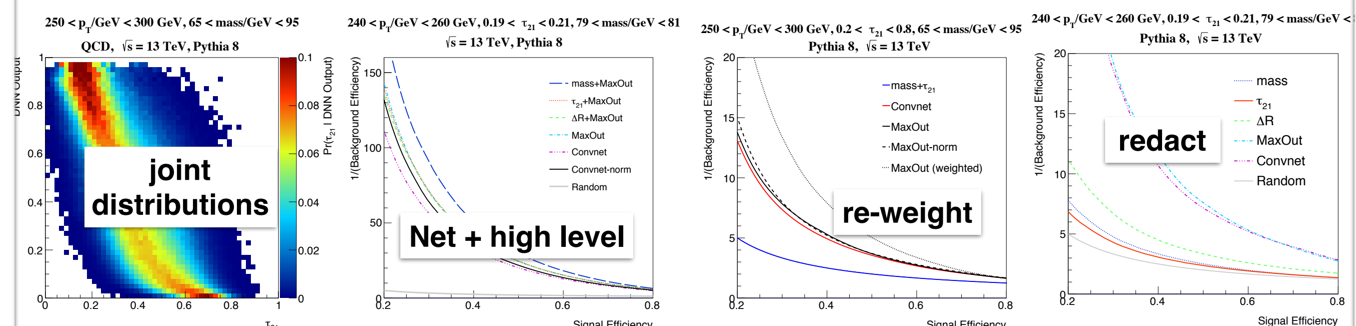
...by how much varies by study; also no two groups have the same setup or metrics…

Learning about Learning

Jet images afford a lot of natural visualization



as a community, we have also developed many techniques



The key to robustness is to study what is being learned; this may even help us to learn something new about QCD!

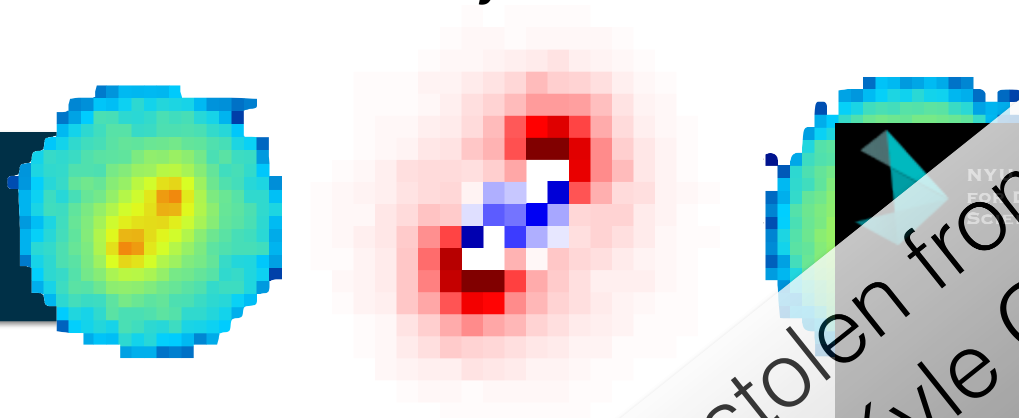
Modern Machine Learning

for Classification, Regression, and Generation in Jet Physics

BERKELEY LAB

Benjamin Nachman

Lawrence Berkeley National Laboratory



CERN Data Science Seminar, Novem

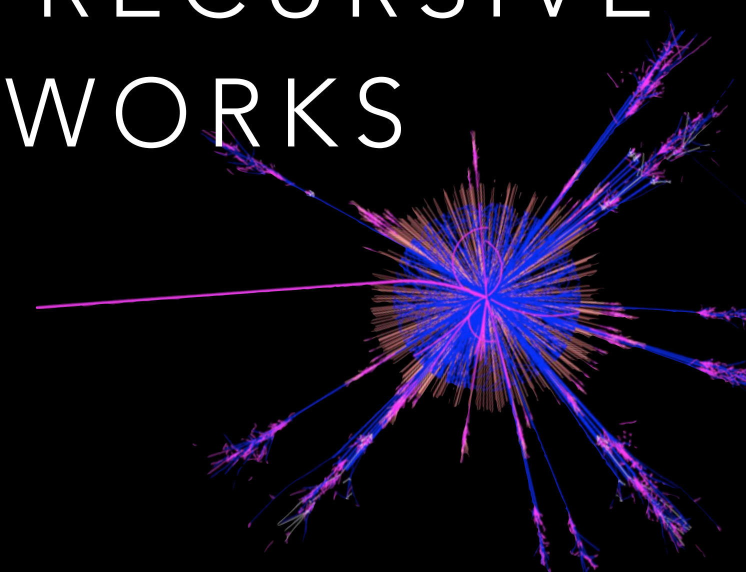
Next slides stolen from Ben Nachmans
and Kyle Cranmer's
Excellent CERN Seminars

NYU CENTER FOR DATA SCIENCE

CENTER FOR COSMOLOGY AND PARTICLE PHYSICS

ARXIV:1702.00748

QCD-AWARE RECURSIVE NEURAL NETWORKS

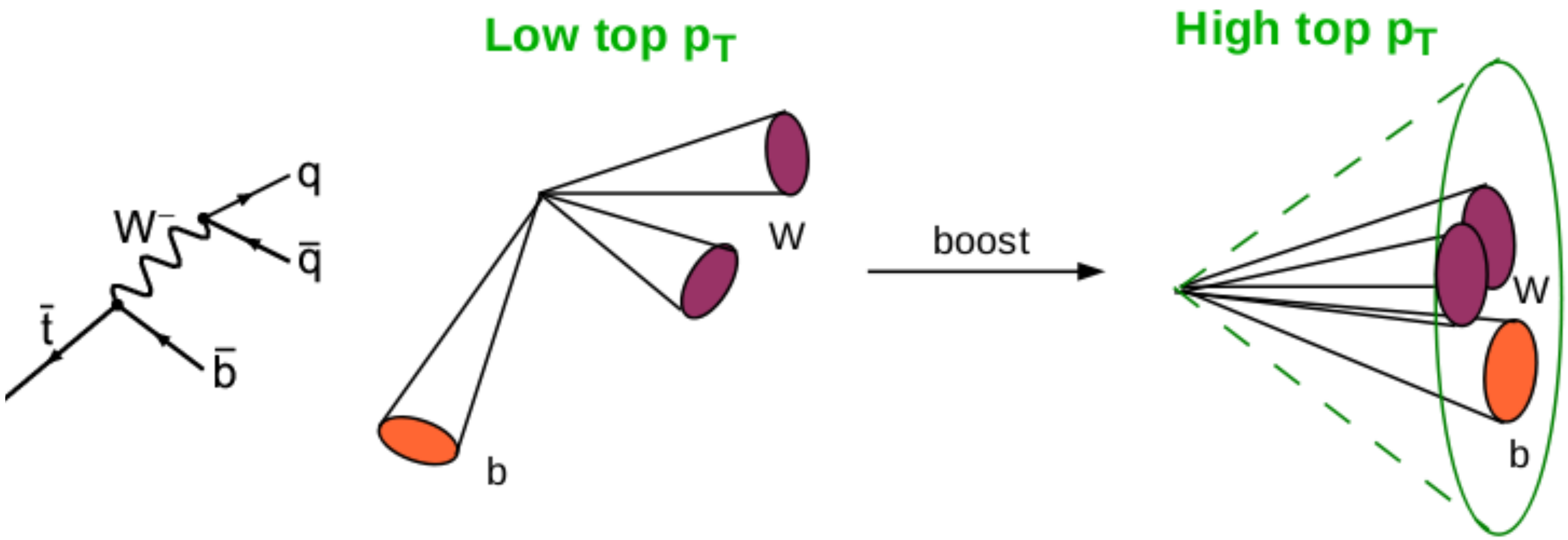


@KyleCranmer
New York University
Department of Physics
Center for Data Science

with:
Gilles Louppe
Kyunghyun Cho
Joan Bruna
Cyril Becot

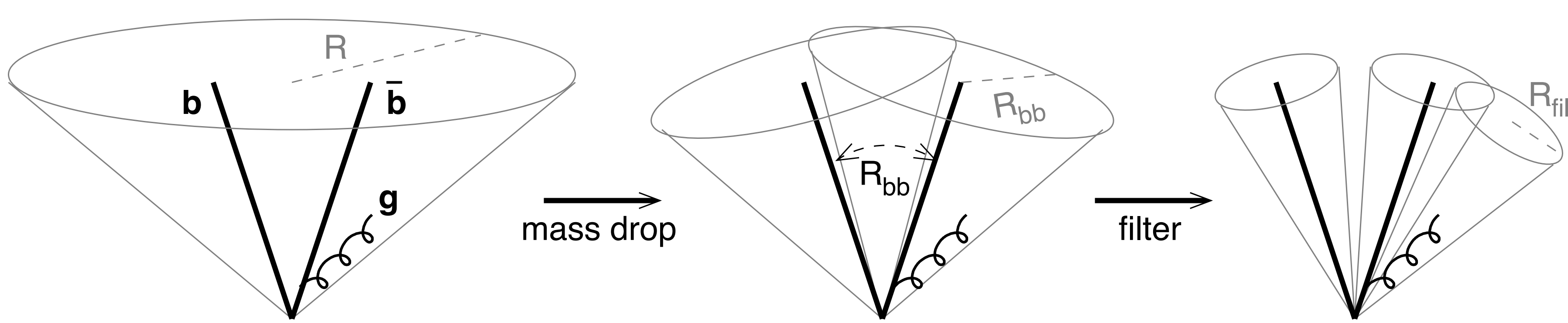
JET SUBSTRUCTURE

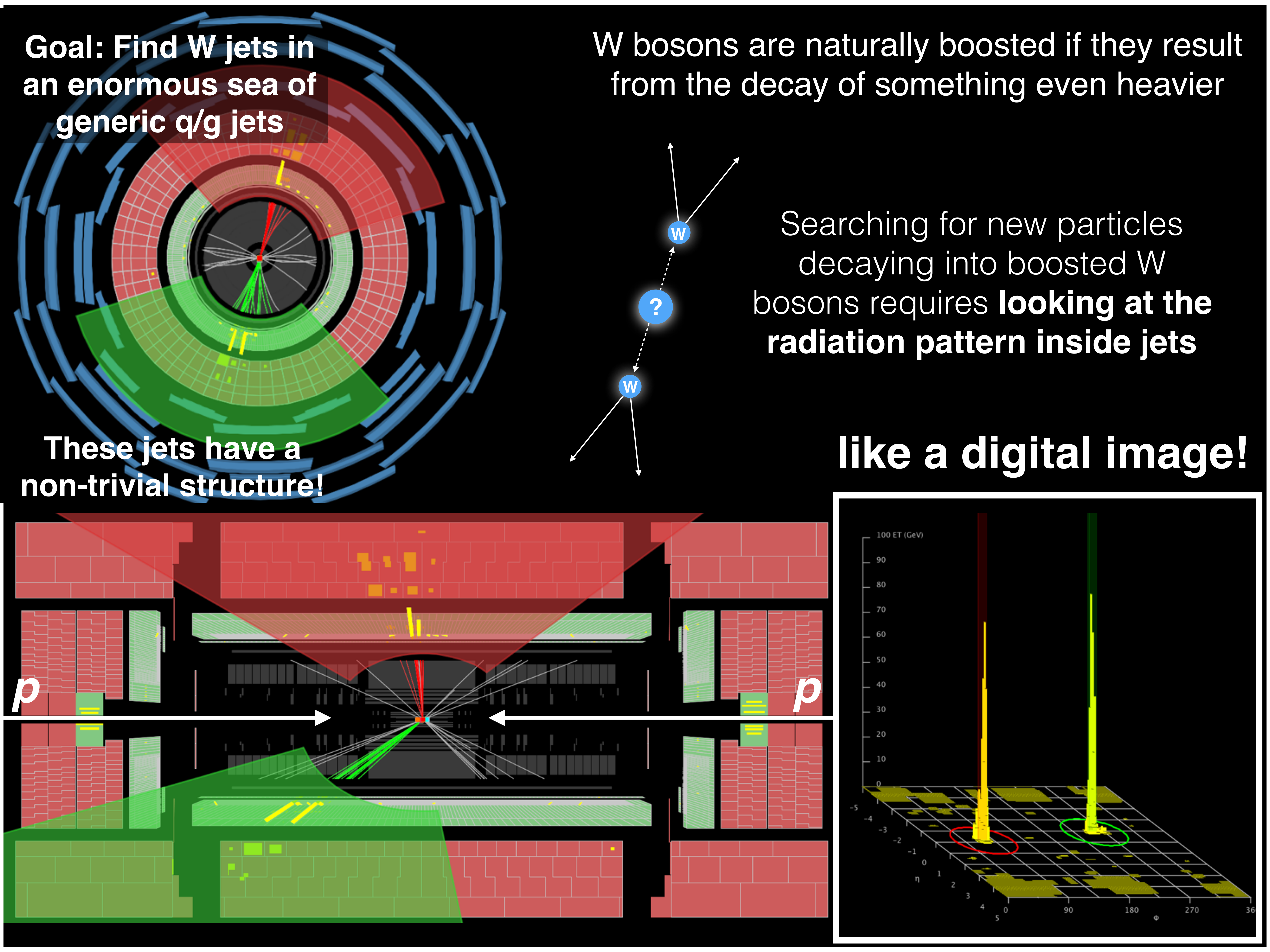
Many scenarios for physics Beyond the Standard Model include highly boosted W, Z, H bosons or top quarks



Identifying these rests on subtle substructure inside jets

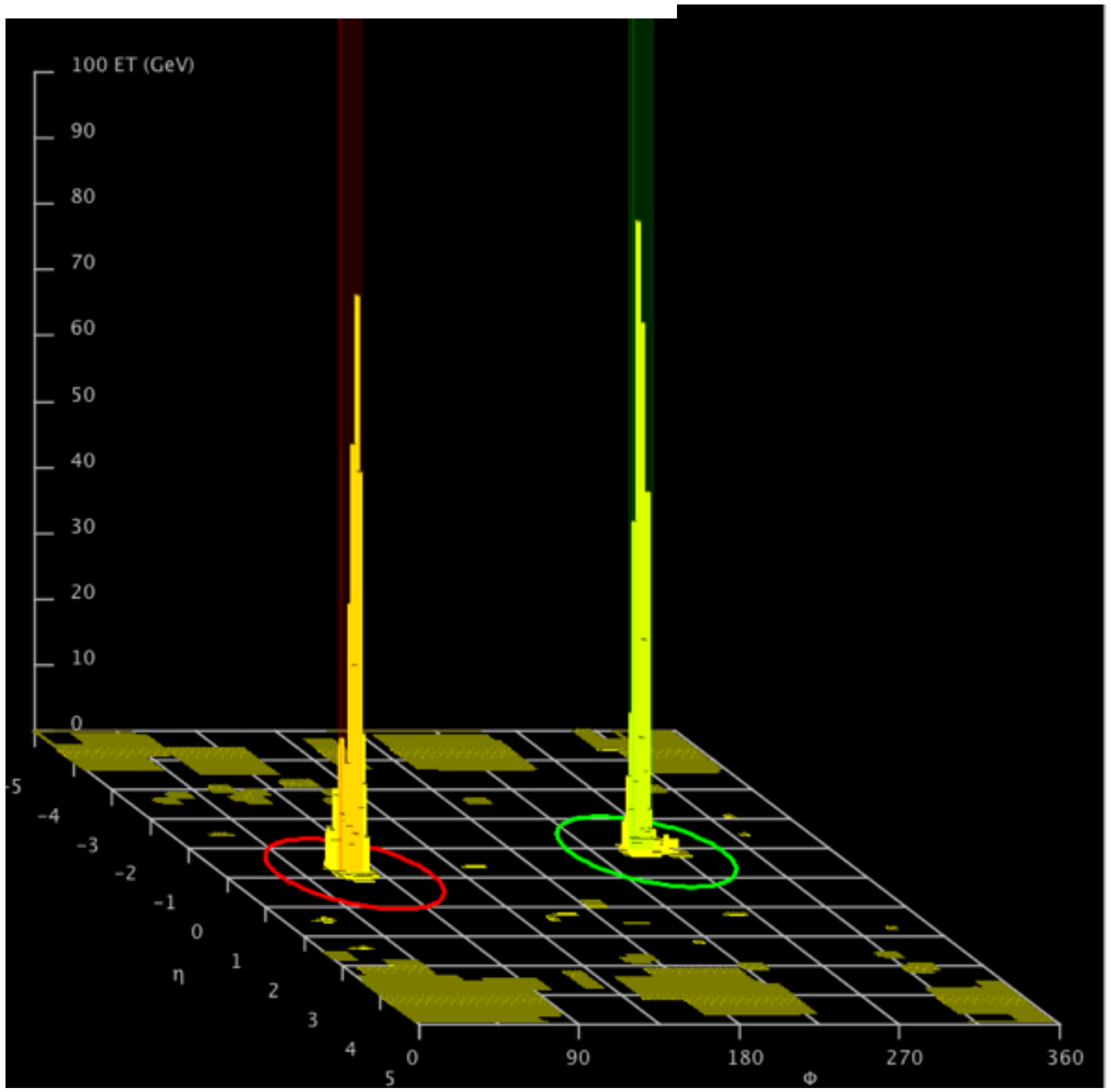
- an enormous number of theoretical effort in developing observables and techniques to tag jets like this

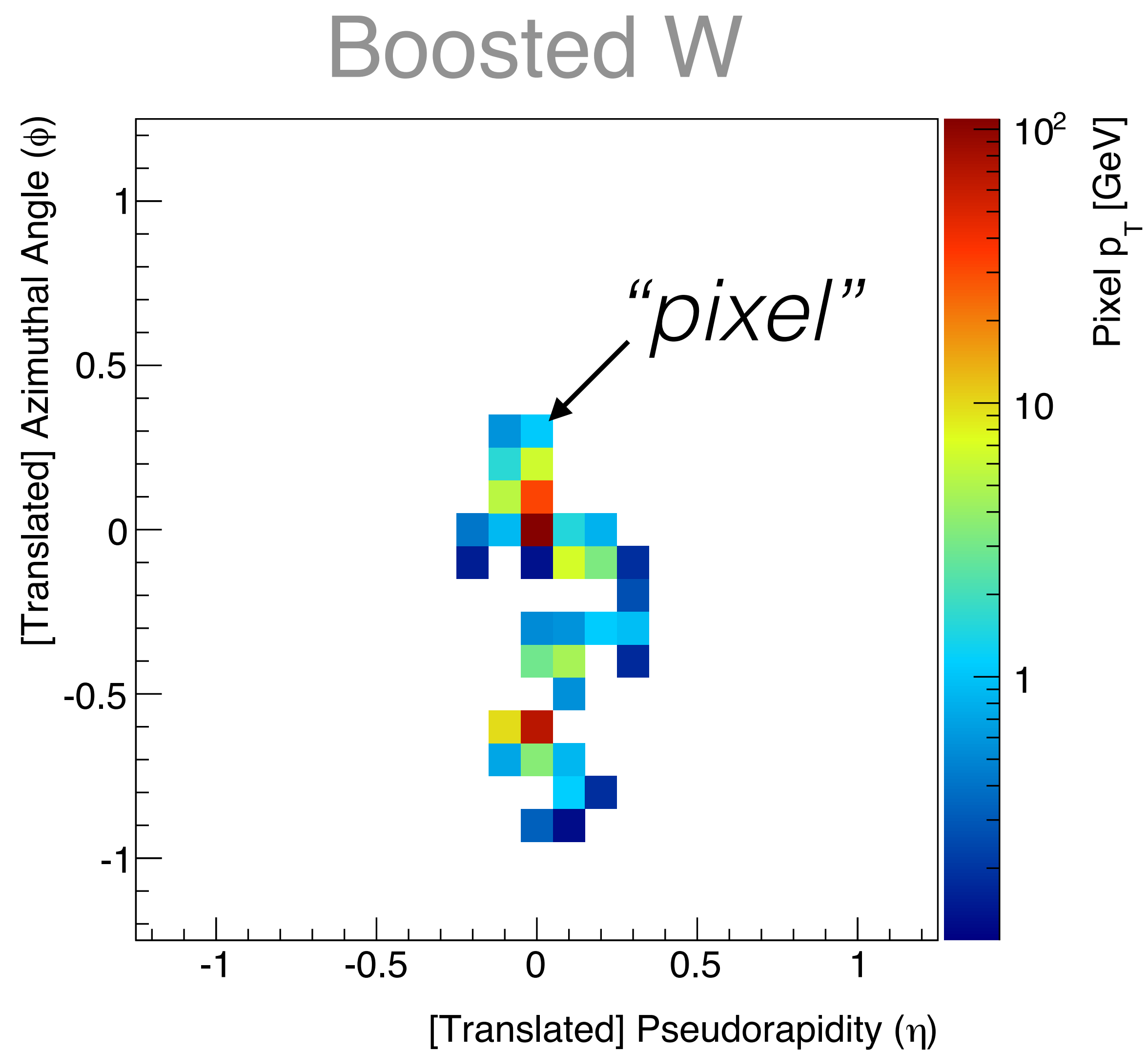



Goal: Find W jets in an enormous sea of generic q/g jets
W bosons are naturally boosted if they result from the decay of something even heavier
W
?
W
Searching for new particles decaying into boosted W bosons requires **looking at the radiation pattern inside jets**
These jets have a non-trivial structure!
like a digital image!
p
p
100 ET (GeV)
90
80
70
60
50
40
30
20
10
0
-5
-4
-3
-2
-1
0
1
2
3
4
5
η
0
90
180
270
Φ

the Jet Image

J. Cogan et al. JHEP 02 (2015) 118

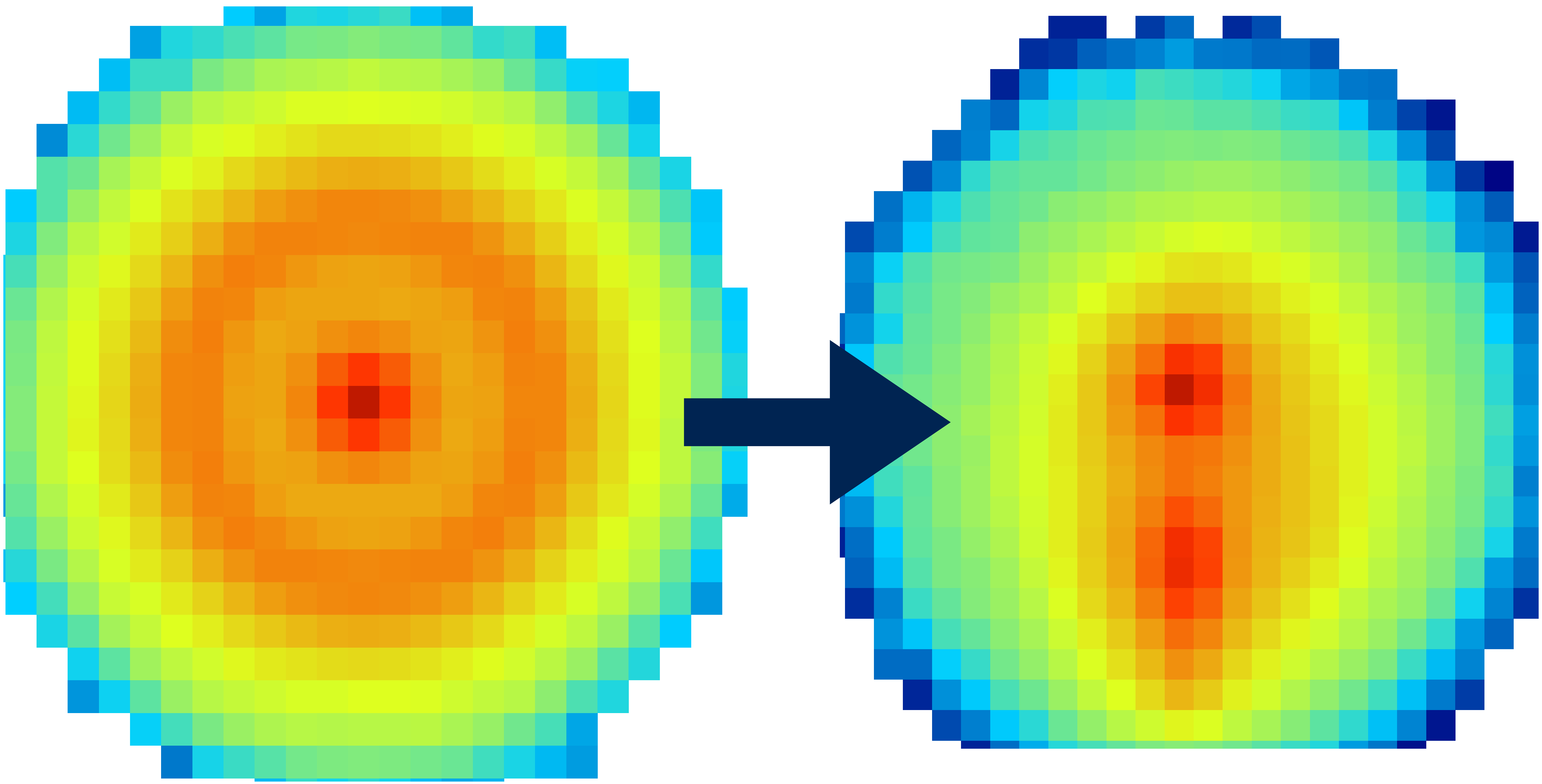




no smooth edges, clear features, low occupancy (number of hit pixels)

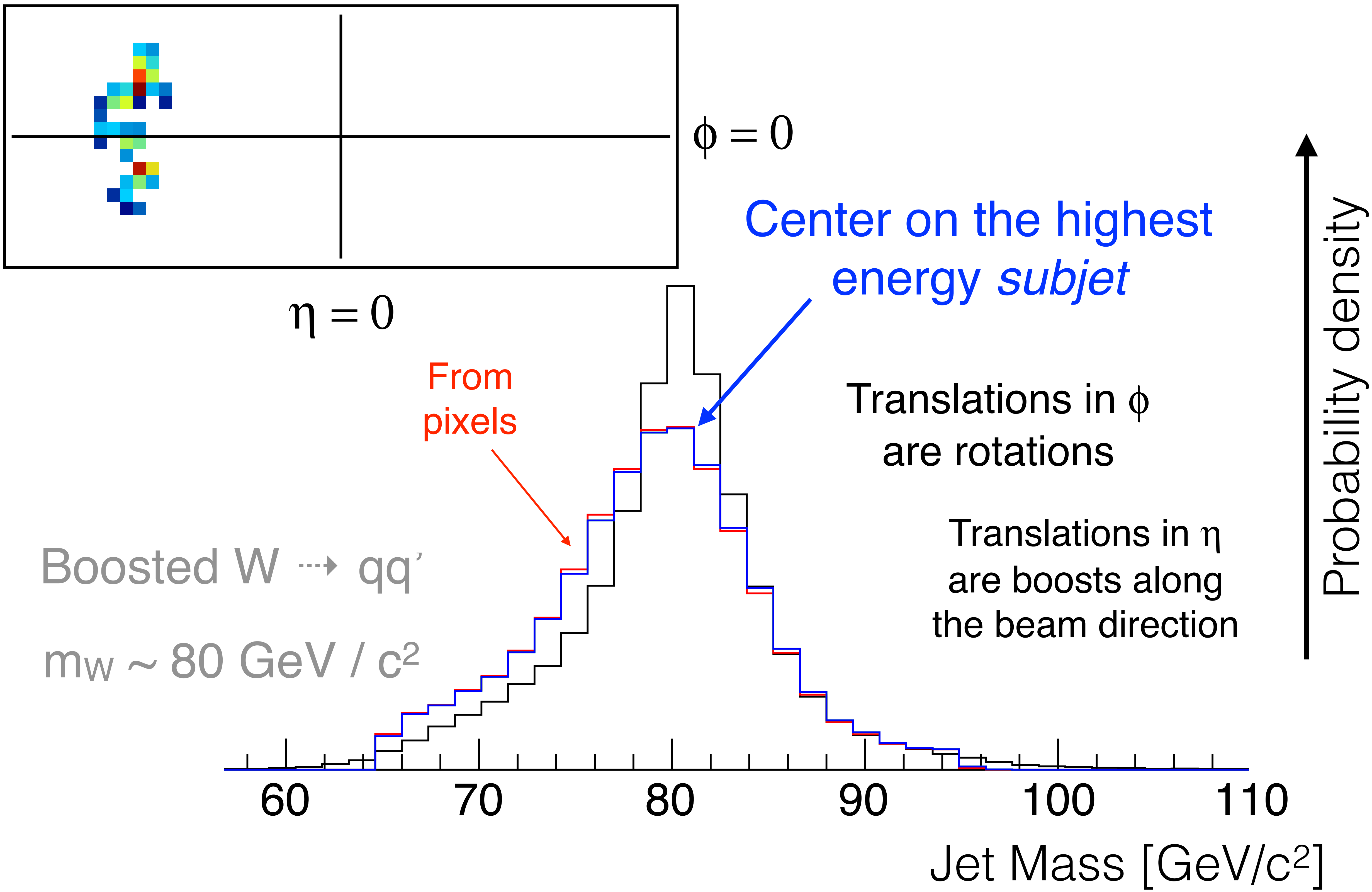
Pre-processing & spacetime symmetries

One of the first typical steps is pre-processing



Can help to learn faster & smarter; but must be careful!

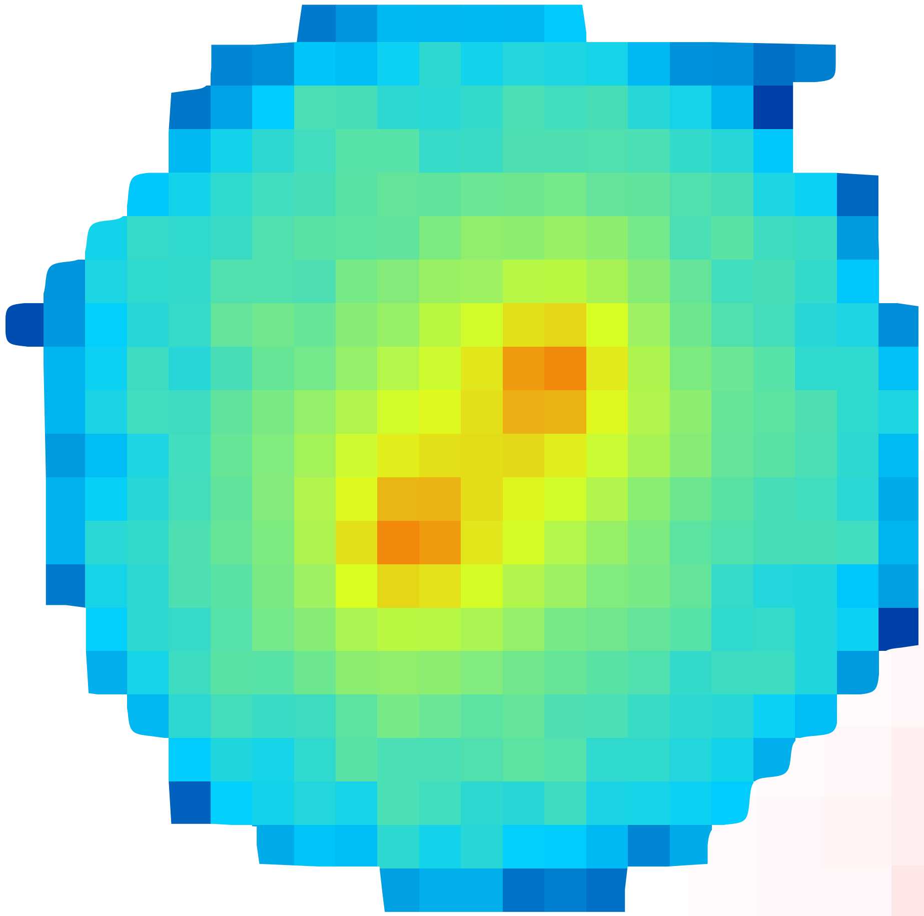
One of the most useful physics-inspired features is the ***jet mass***



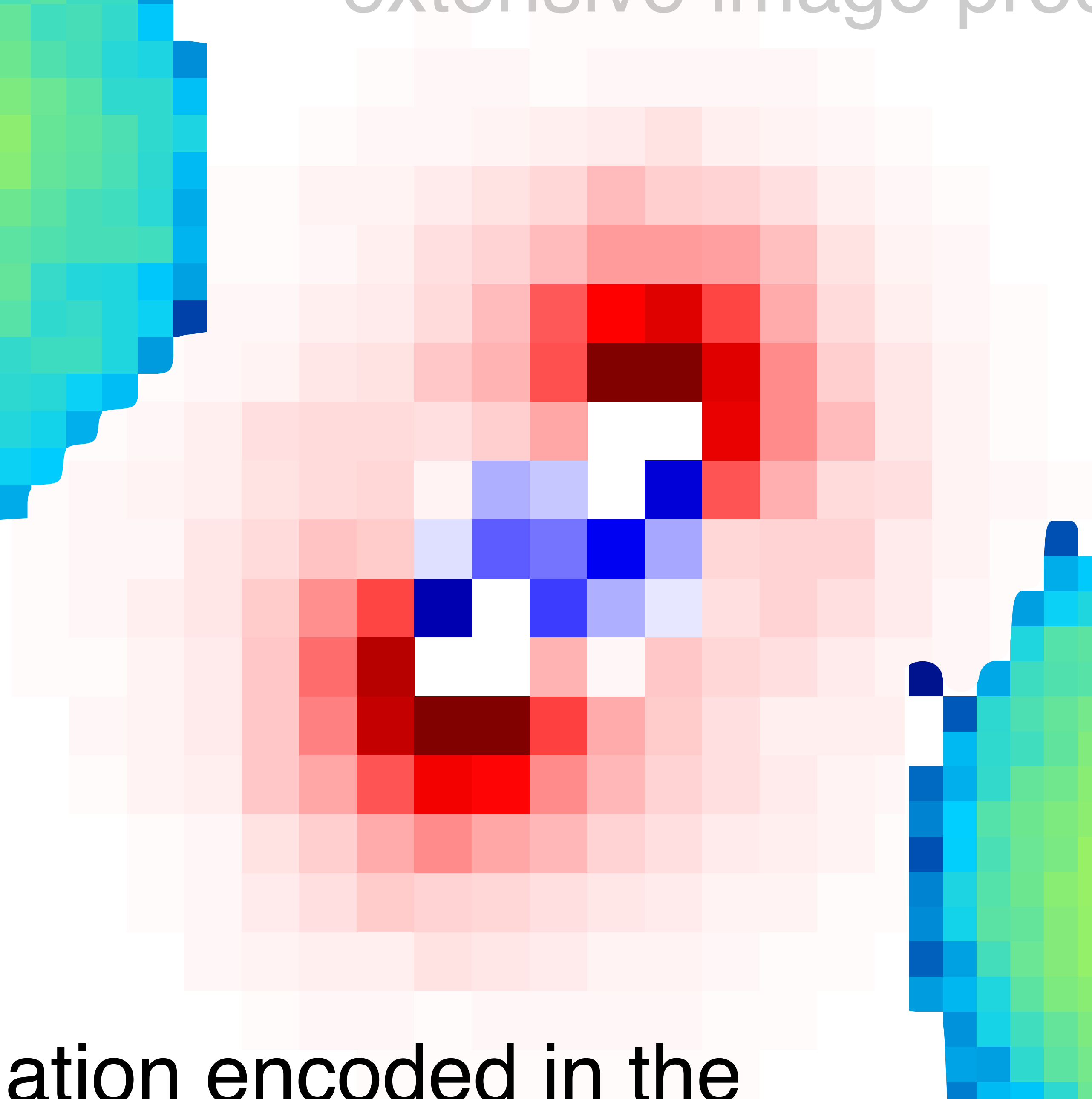
Why images?

Can directly visualize physics

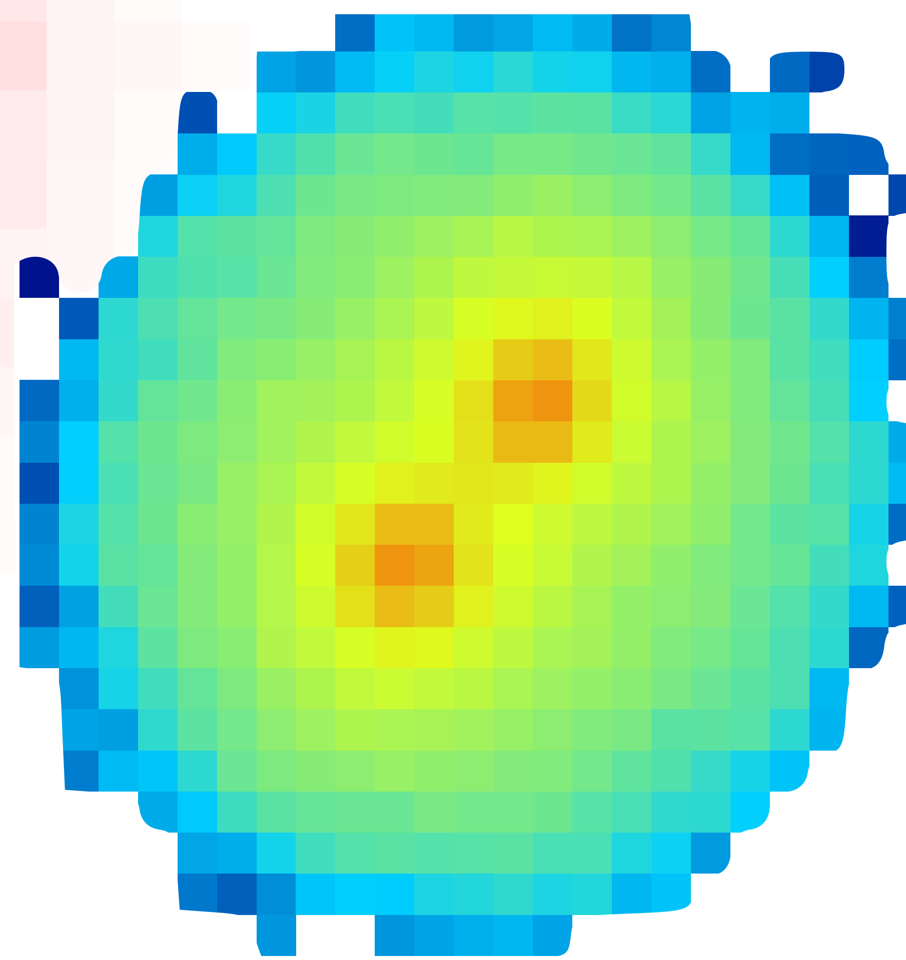
and we can benefit from the extensive image processing literature



$\mathbf{W} \dashrightarrow q\bar{q}$

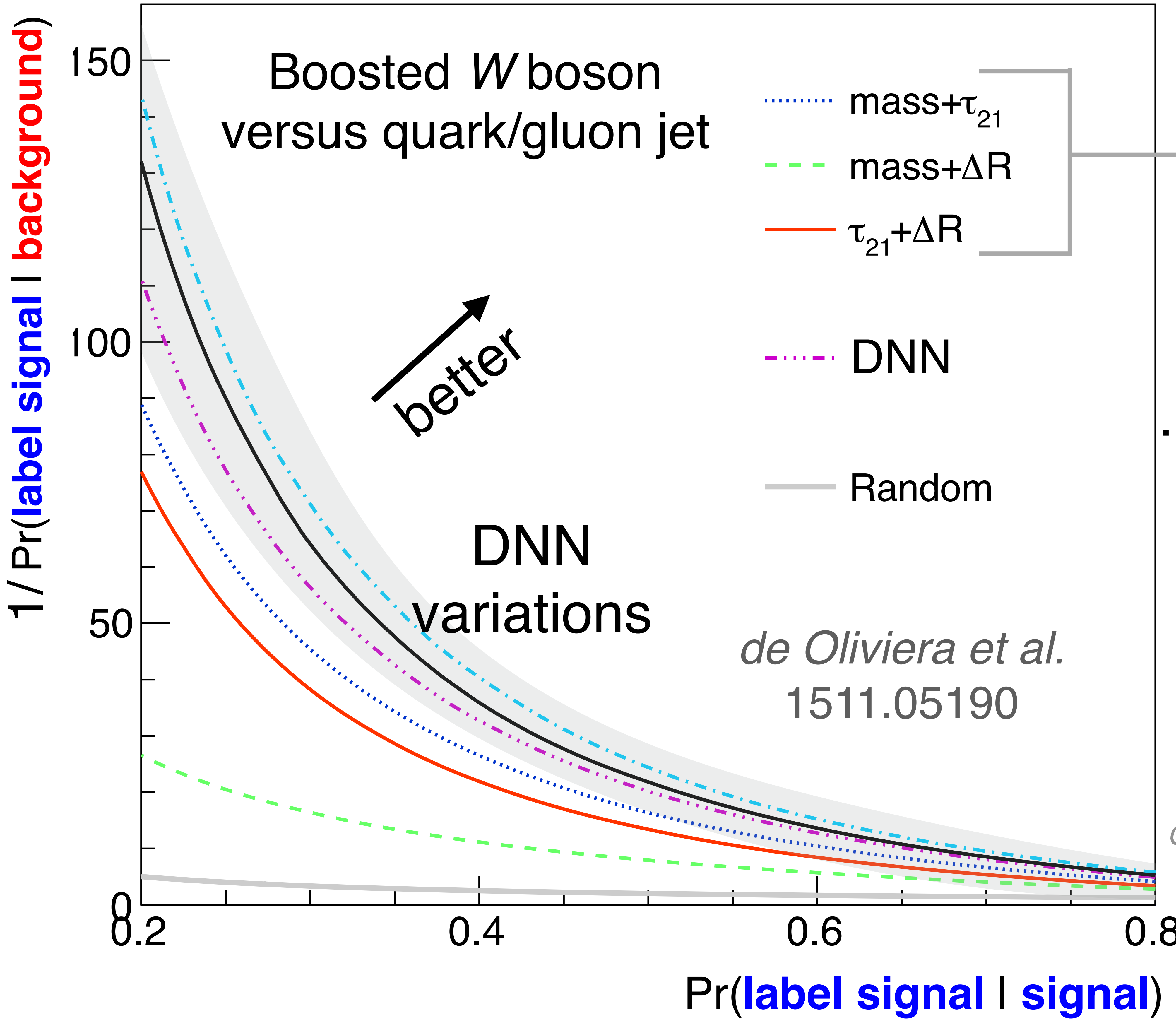


$\mathbf{g} \dashrightarrow q\bar{q}$



there is information encoded in the physical distance between pixels

Modern Deep NN's for Classification



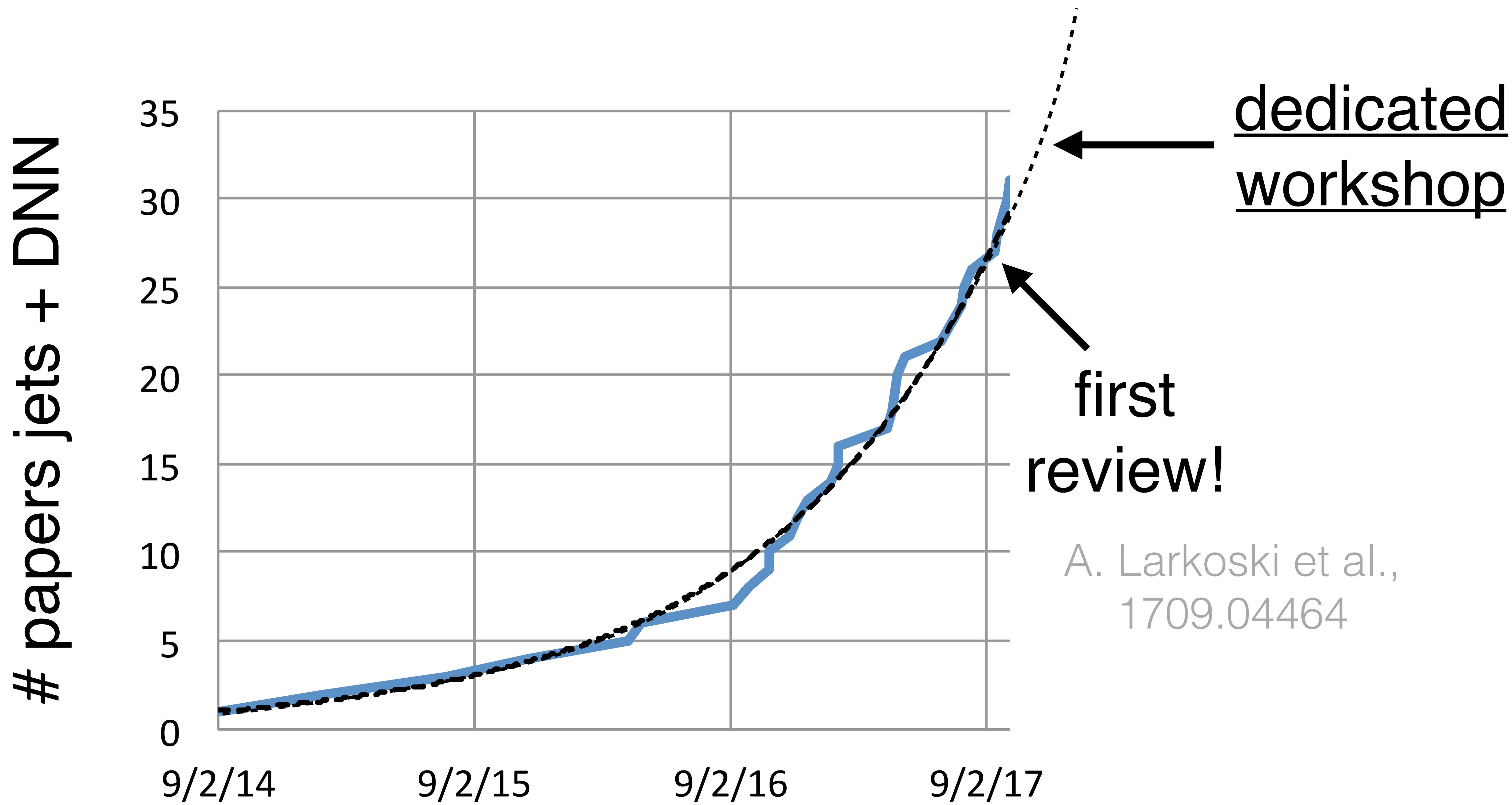
mass, τ_{21}, ΔR are all simple functions of the image

…what the DNN is learning is active R&D!

See also
L. Almeida et al. 1501.05968
Baldi et al. 1603.09349
J. Barnard et al. 1609.00607
P. Komiske et al. 1612.01551
G. Kasieczka et al. 1701.08784
W. Bhimji et al. 1711.03573

Exciting New Directions

So far only scratches the surface
…this is a very active field of research!



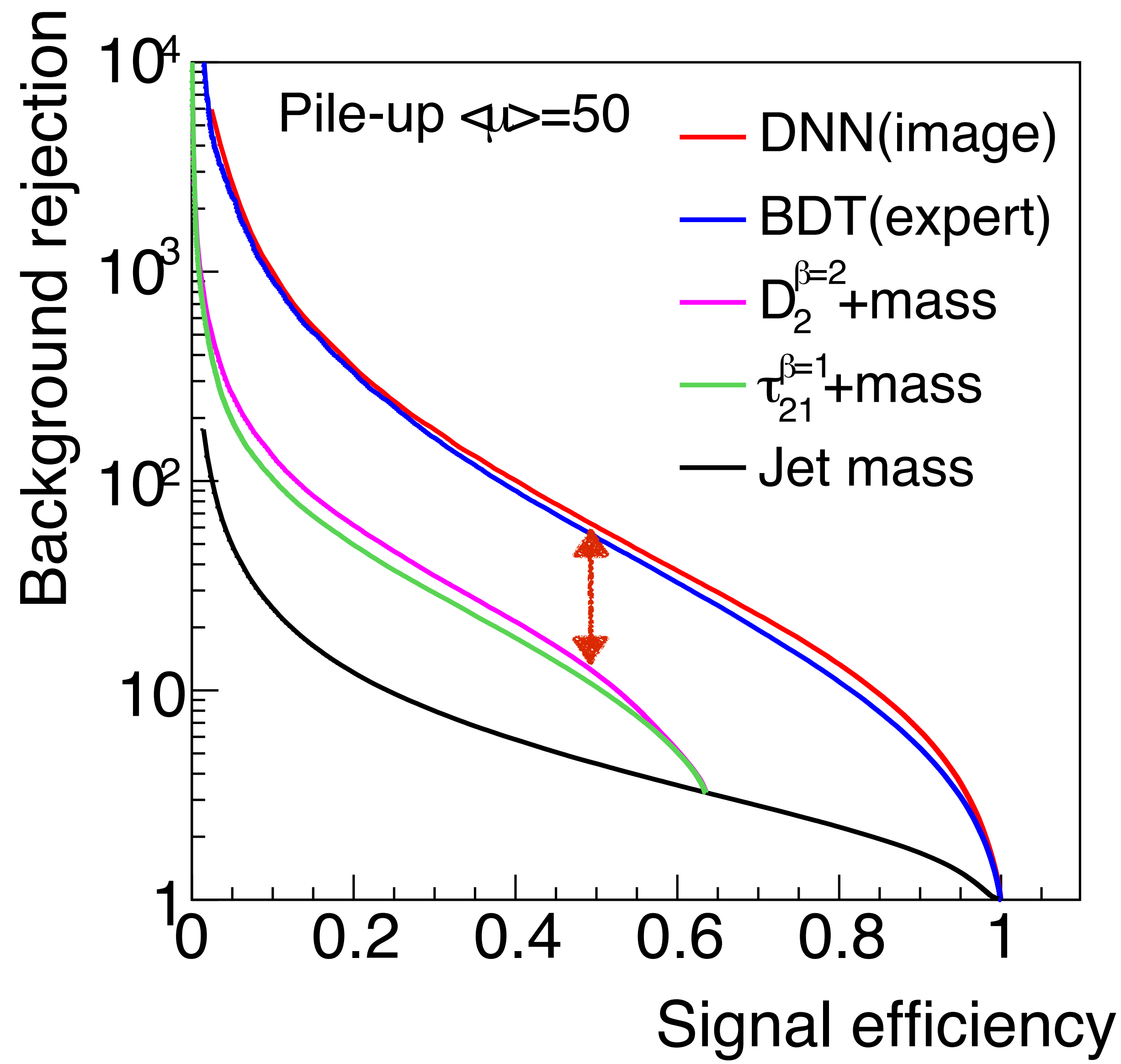
A. Larkoski et al., 1709.04464

Oliveira, et. al arXiv:1511.05190
Whiteson, et al arXiv:1603.09349
Barnard, et al arXiv:1609.00607

While the DNN shows a significant improvement with respect to the jet mass combined with single theory inspired variable (eg. τ_{21}, D_2), only a small improvement with respect to a BDT using several theory-inspired variables

Other Problems:

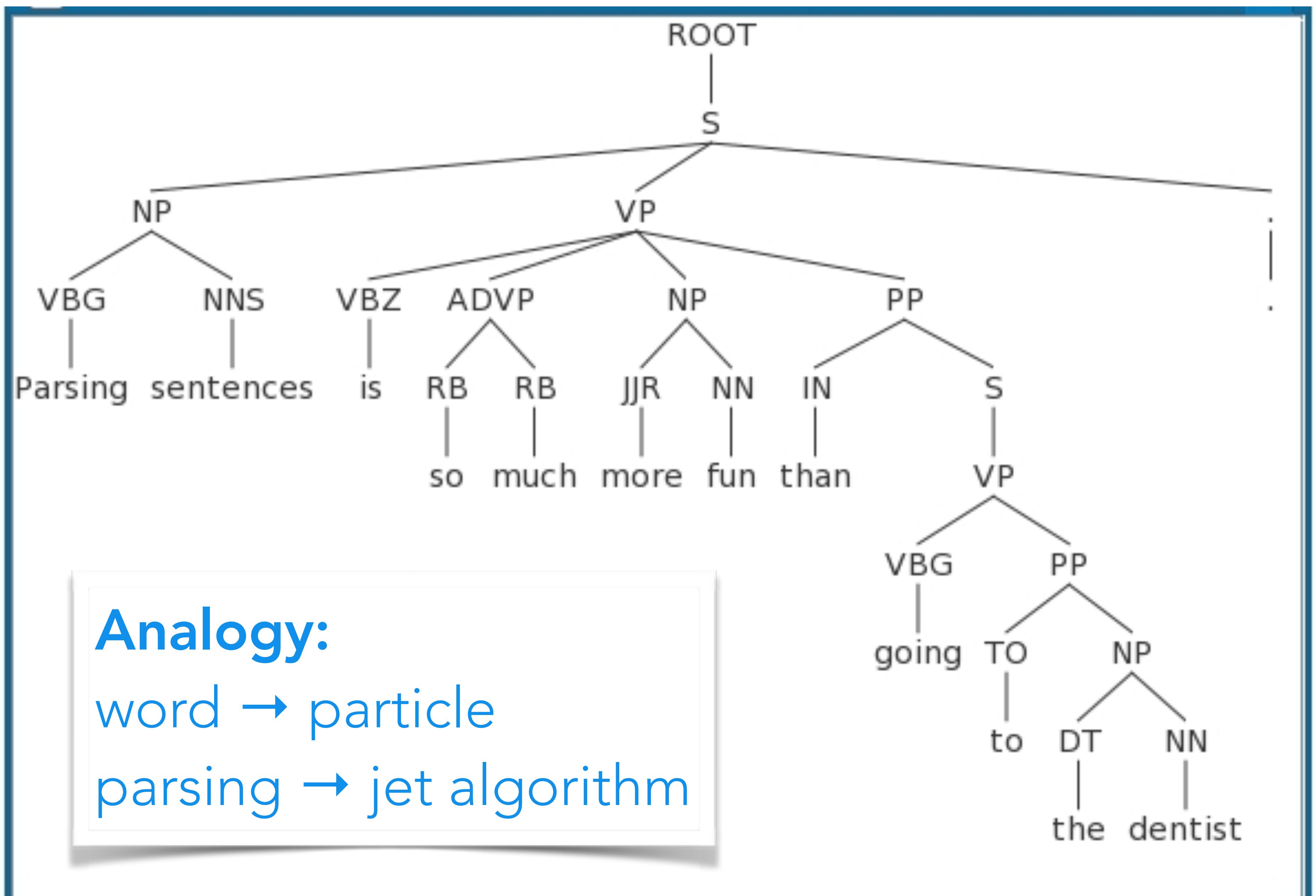
- image-based approach not easily generalized to non-uniform calorimeters
- not easy to extend to tracks, projecting into towers looses information
- theory inspired variables work on set of 4-vectors & have important theoretical properties



FROM IMAGES TO SENTENCES

Recursive Neural Networks showing great performance for Natural Language Processing tasks

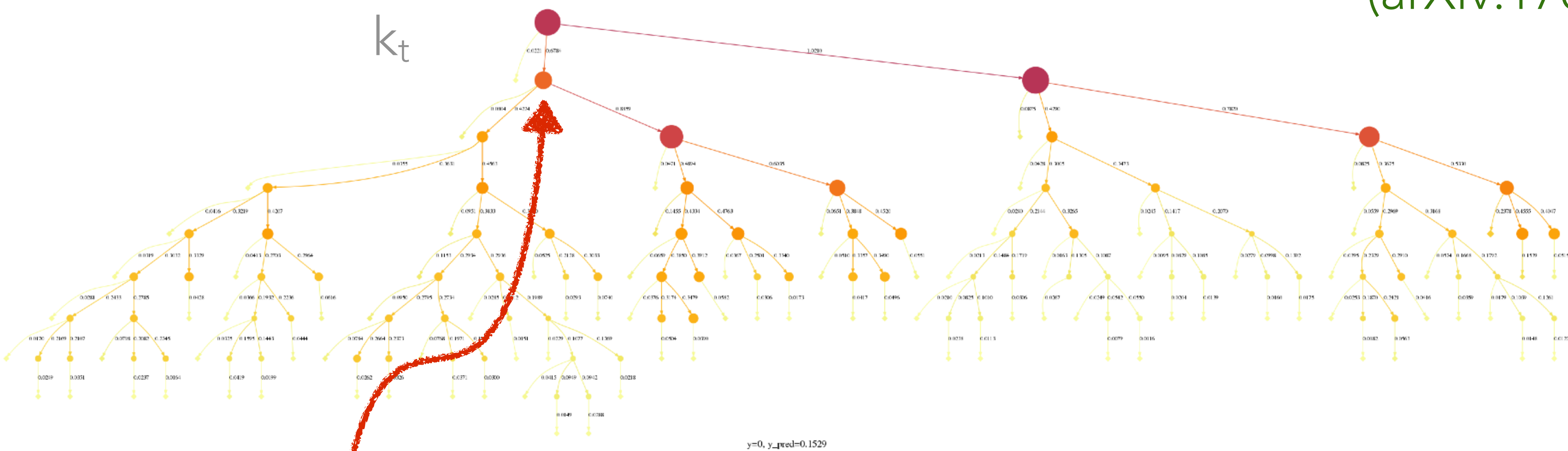
- neural network's topology given by parsing of sentence!



Analogy:
word → particle
parsing → jet algorithm

QCD-INSPIRED RECURSIVE NEURAL NETWORKS

(arXiv:1702.00748)



$$\mathbf{h}_k^{\text{jet}} = \begin{cases} \mathbf{u}_k & \text{if } k \text{ is a leaf} \\ \mathbf{z}_H \odot \tilde{\mathbf{h}}_k^{\text{jet}} + \mathbf{z}_L \odot \mathbf{h}_{k_L}^{\text{jet}} + & \text{otherwise} \\ \hookrightarrow \mathbf{z}_R \odot \mathbf{h}_{k_R}^{\text{jet}} + \mathbf{z}_N \odot \mathbf{u}_k & \end{cases}$$

$$\mathbf{u}_k = \sigma\left(W_u g(\mathbf{o}_k) + b_u\right)$$

$$\mathbf{o}_k = \begin{cases} \mathbf{v}_{i(k)} & \text{if } k \text{ is a leaf} \\ \mathbf{o}_{k_L} + \mathbf{o}_{k_R} & \text{otherwise} \end{cases}$$

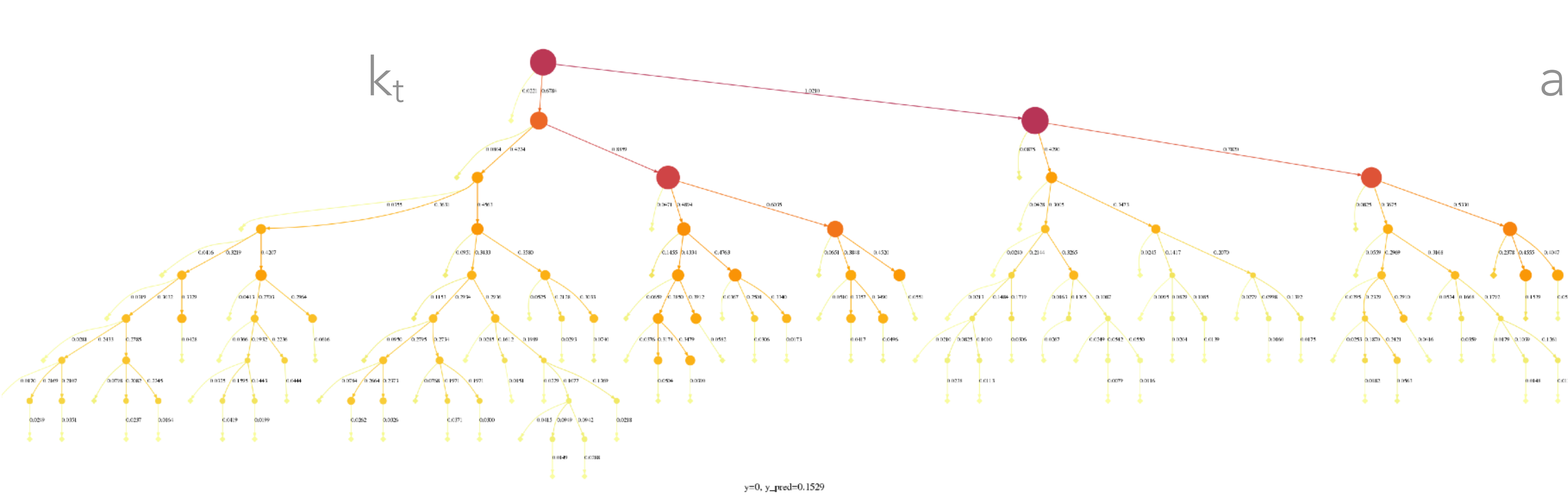
$$\tilde{\mathbf{h}}_k^{\text{jet}} = \sigma\left(W_{\tilde{h}} \begin{bmatrix} \mathbf{r}_L \odot \mathbf{h}_{k_L}^{\text{jet}} \\ \mathbf{r}_R \odot \mathbf{h}_{k_R}^{\text{jet}} \\ \mathbf{r}_N \odot \mathbf{u}_k \end{bmatrix} + b_{\tilde{h}}\right)$$

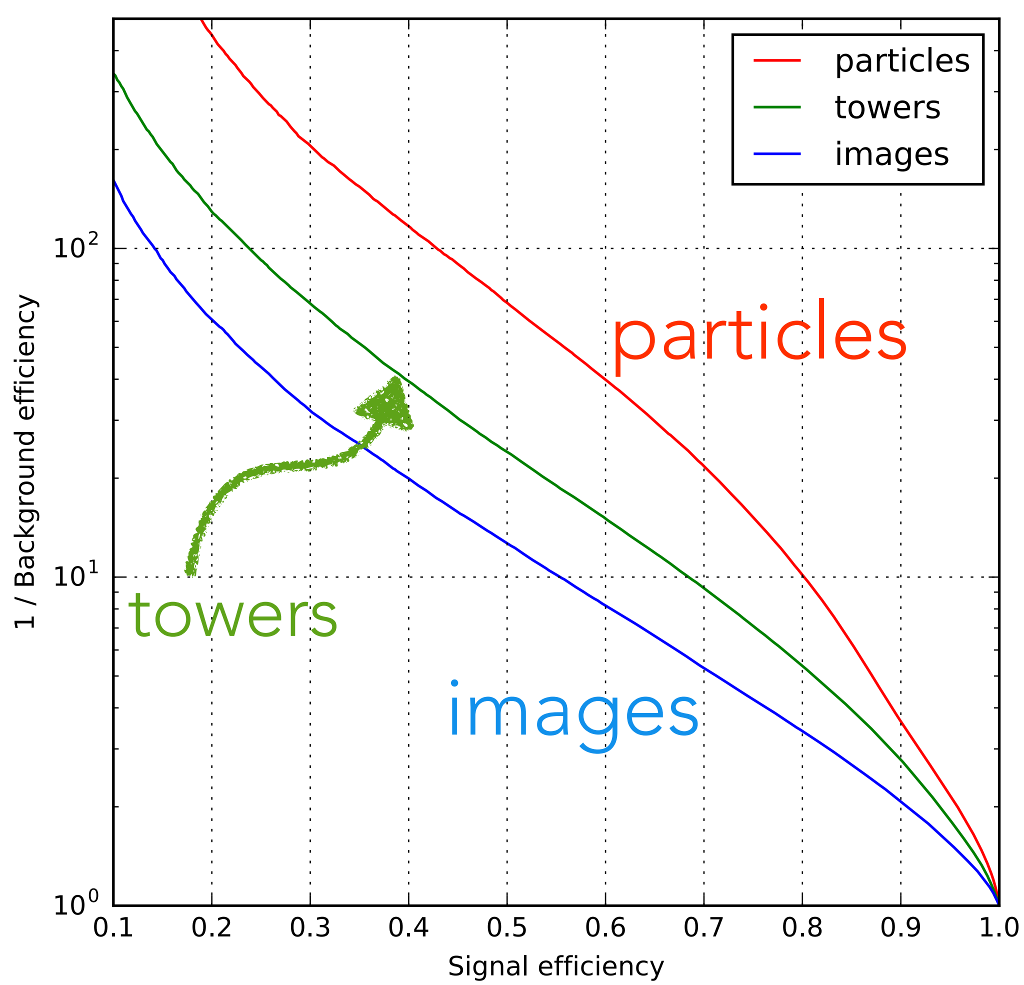
$$\begin{bmatrix} \mathbf{z}_H \\ \mathbf{z}_L \\ \mathbf{z}_R \\ \mathbf{z}_N \end{bmatrix} = \text{softmax}\left(W_z \begin{bmatrix} \tilde{\mathbf{h}}_k^{\text{jet}} \\ \mathbf{h}_{k_L}^{\text{jet}} \\ \mathbf{h}_{k_R}^{\text{jet}} \\ \mathbf{u}_k \end{bmatrix} + b_z\right)$$

$$\begin{bmatrix} \mathbf{r}_L \\ \mathbf{r}_R \\ \mathbf{r}_N \end{bmatrix} = \text{sigmoid}\left(W_r \begin{bmatrix} \mathbf{h}_{k_L}^{\text{jet}} \\ \mathbf{h}_{k_R}^{\text{jet}} \\ \mathbf{u}_k \end{bmatrix} + b_r\right)$$

- Each node combines 4-momentum in (E-scheme recombination of o_k) and a non-linear transformation of hidden state of children h_{kL}, $h_{kR} \in \mathbb{R}^{40}$
- Recursively applied (shared weights, Markov)
- "gating" allows for weighting of information of L/R children and for to flow directly along one branch

QCD-INSPIRED RECURSIVE NEURAL NETWORKS





- W-jet tagging example using data from Dawe, et al arXiv:1609.00607
- down-sampling by projecting into images looses information
- RNN needs much less data to train!

Neural Message Passing for Jet Physics

Isaac Henrion, Johann Brehmer, Joan Bruna, Kyunghun Cho, Kyle Cranmer
Center for Data Science
New York University
New York, NY 10012
{henrion*, johann.brehmer, bruna, kyunghyun, kyle.cranmer*}@nyu.edu

Gilles Louppe
Department of Computer Science
University of Liège
Belgium
g.louppe@ulg.ac.be

Gaspar Rochette
Department of Computer Science
École Normale Supérieure
Paris, France
gaspar.rochette@ens.fr

Abstract

Supervised learning has incredible potential for particle physics, and one application that has received a great deal of attention involves collimated sprays of particles called jets. Recent progress for jet physics has leveraged machine learning techniques based on computer vision and natural language processing. In this work, we consider message passing on a graph where the nodes are the particles in a jet. We design variants of a message-passing neural network (MPNN); (1) with a learnable adjacency matrix, (2) with a learnable symmetric adjacency matrix, and (3) with a set2set aggregated hidden state and MPNN with an identity adjacency matrix. We compare these against the previously proposed recursive neural network with a fixed tree structure and show that the MPNN with a learnable adjacency matrix and two message-passing iterations outperforms all the others.

Table 1: Summary of classification performance for several approaches.

Network	Iterations	ROC AUC	$R_{\epsilon=50\%}$
RecNN-k_t (without gating) [10]	1	0.9185 ± 0.0006	68.3 ± 1.8
RecNN-k_t (with gating) [10]	1	0.9195 ± 0.0009	74.3 ± 2.4
RecNN-desc-p_T (without gating) [10]	1	0.9189 ± 0.0009	70.4 ± 3.6
RecNN-desc-p_T (with gating) [10]	1	0.9212 ± 0.0005	$\mathbf{83.3 \pm 3.1}$
RelNet	1	0.9161 ± 0.0029	67.69 ± 6.80
MPNN (directed)	1	0.9196 ± 0.0015	89.35 ± 3.54
MPNN (directed)	2	0.9223 ± 0.0008	$\mathbf{98.26 \pm 4.28}$
MPNN (directed)	3	0.9188 ± 0.0031	85.93 ± 8.50
MPNN (undirected)	1	0.9193 ± 0.0015	86.41 ± 3.80
MPNN (undirected)	2	0.8949 ± 0.1004	$\mathbf{97.27 \pm 5.02}$
MPNN (undirected)	3	0.9185 ± 0.0036	84.53 ± 8.64
MPNN (set, directed)	1	0.9189 ± 0.0017	88.23 ± 4.53
MPNN (set, directed)	2	0.9191 ± 0.0046	87.46 ± 14.14
MPNN (set, directed)	3	0.9176 ± 0.0049	$\mathbf{88.33 \pm 9.84}$
MPNN (set, undirected)	1	0.9196 ± 0.0014	85.65 ± 4.48
MPNN (set, undirected)	2	0.9220 ± 0.0007	$\mathbf{94.70 \pm 2.95}$
MPNN (set, undirected)	3	0.9158 ± 0.0054	75.94 ± 12.54
MPNN (id)	1	0.9169 ± 0.0013	74.75 ± 2.65
MPNN (id)	2	0.9162 ± 0.0020	74.41 ± 3.50
MPNN (id)	3	0.9158 ± 0.0029	74.51 ± 5.20