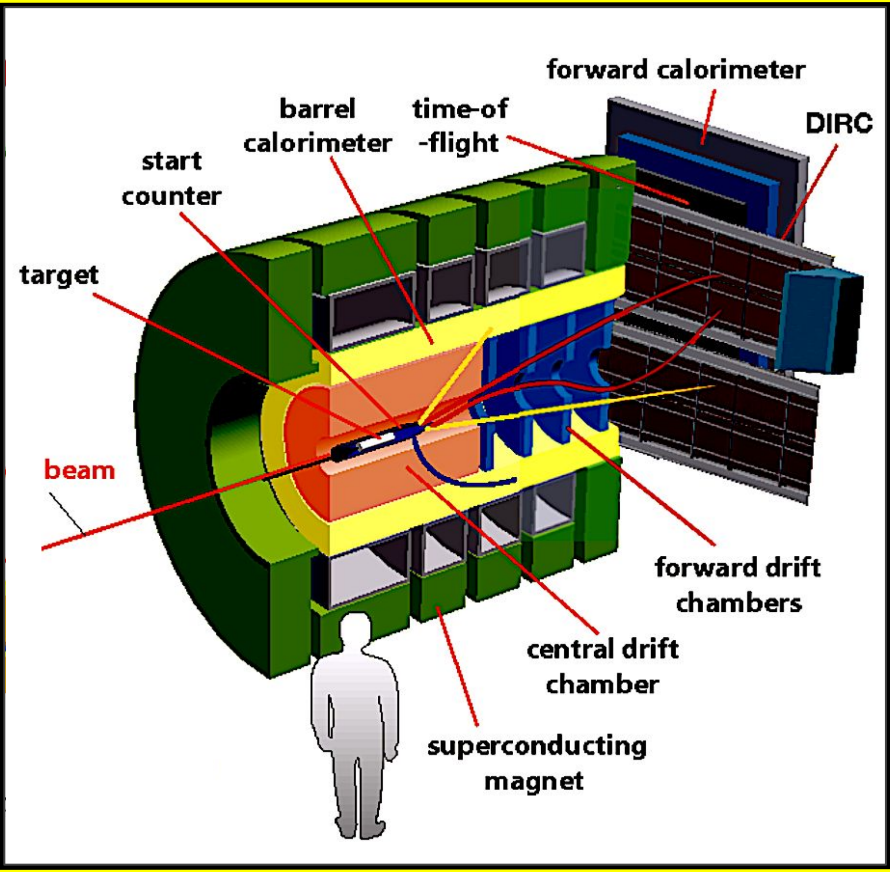
A deep learning approach to PID and alignment of the GlueX DIRC

C. Fanelli

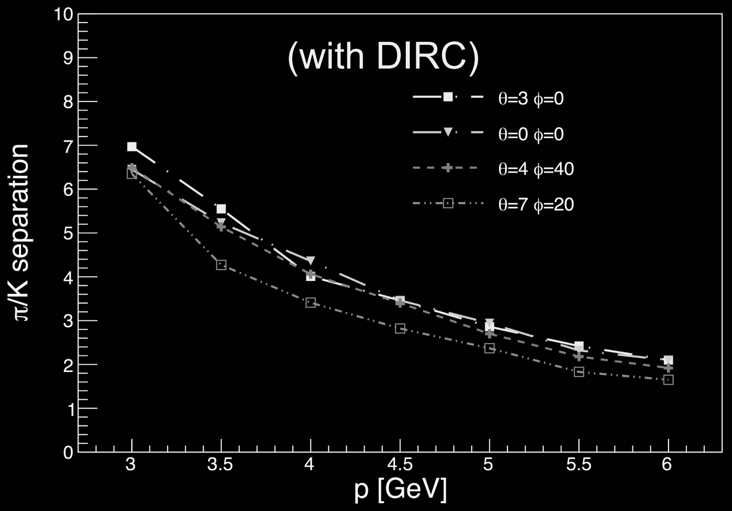
User Group Annual Meeting at JLab, June 2018

The GlueX detector in Hall D and the DIRC

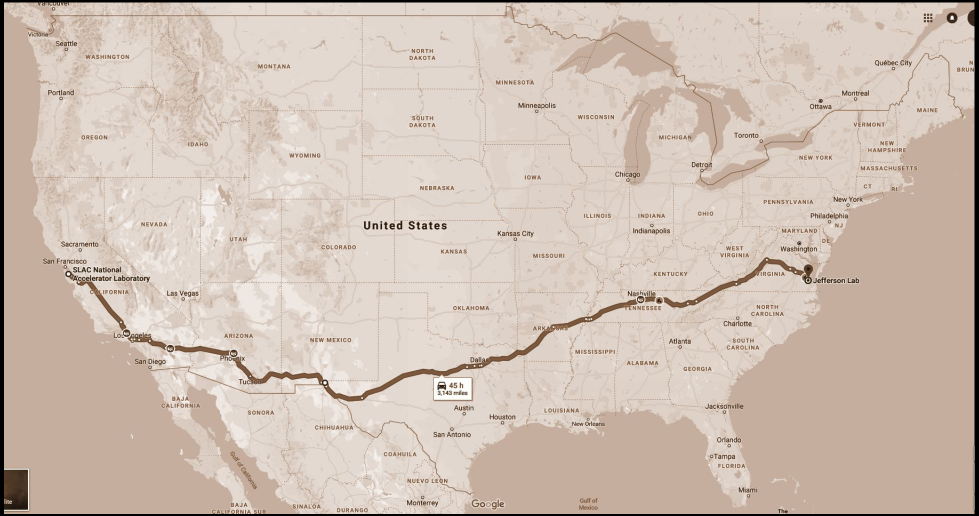


DIRC will improve GlueX PID capabilities

(current π/K separation limited to 2 GeV/c)



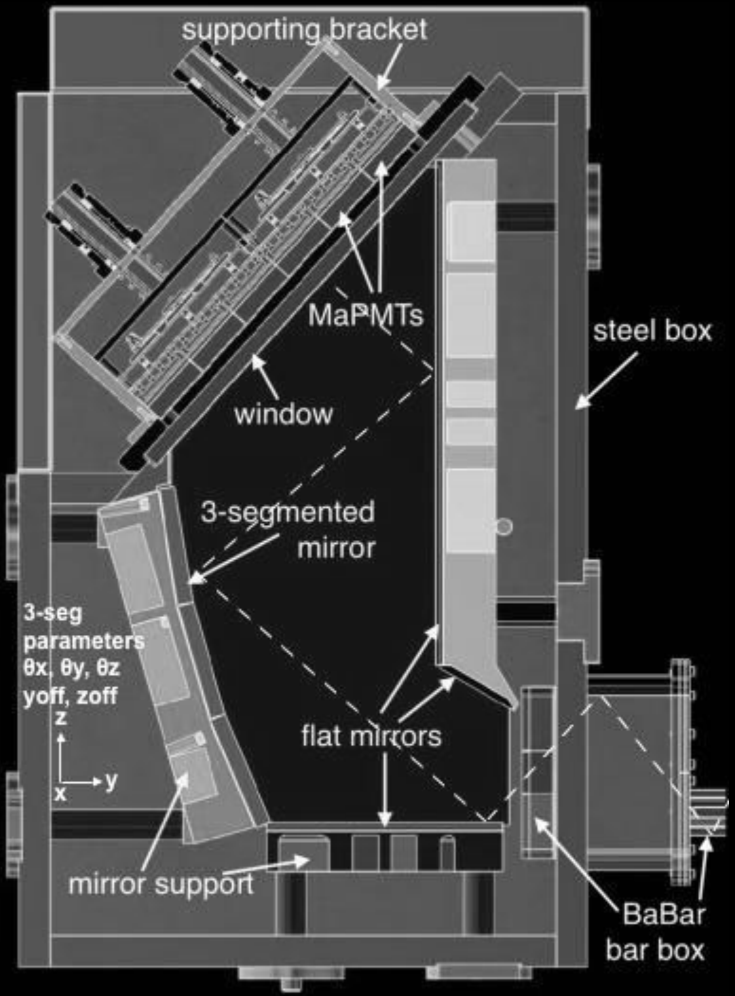
Transportation and installation

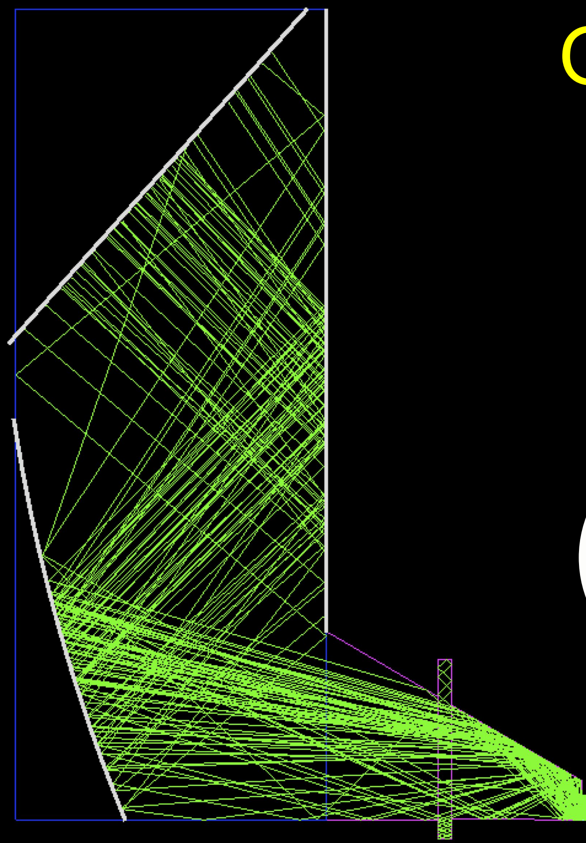


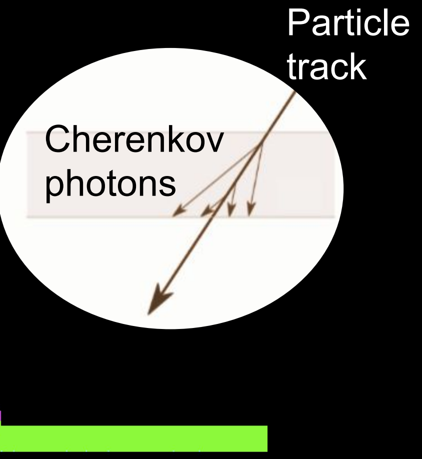
Milestone:

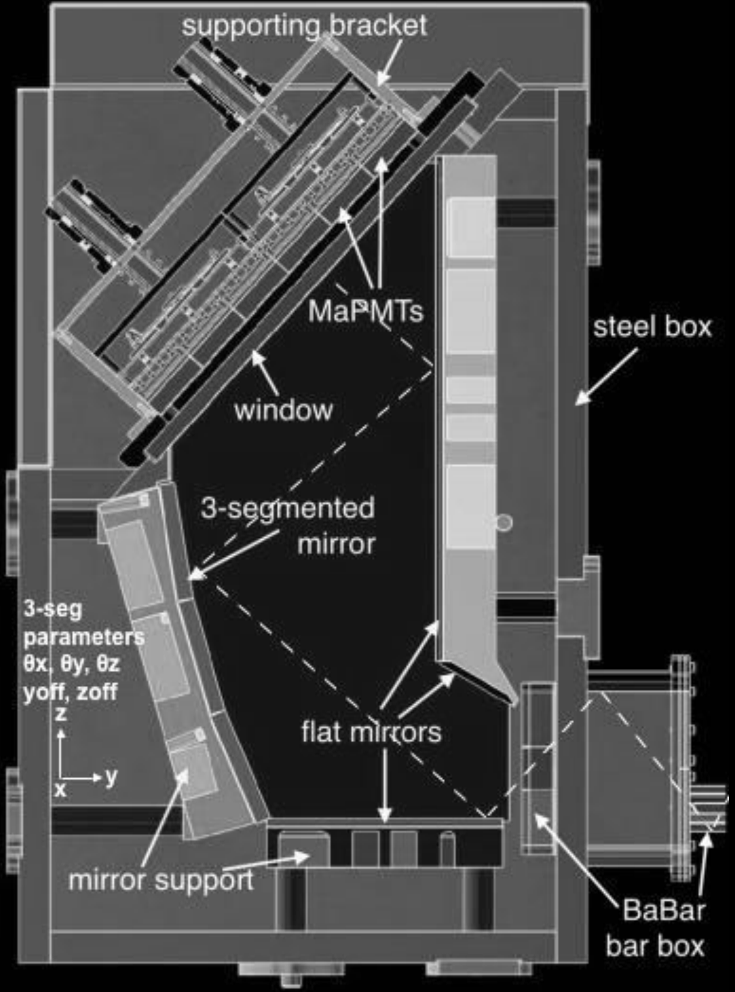
- 6/5/2018 all of the DIRC bar boxes in Hall D
- No issues mechanically and optically
- 2 installed in the lower part

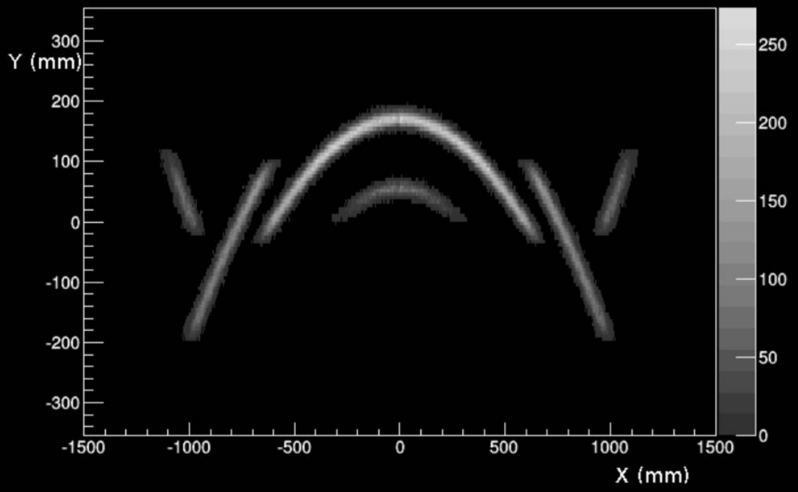
Optical Box


supporting bracket
MaPMTs
steel box
window
3-segmented mirror
3-seg parameters θx, θy, θz yoff, zoff
z
y
x
flat mirrors
mirror support
BaBar bar box



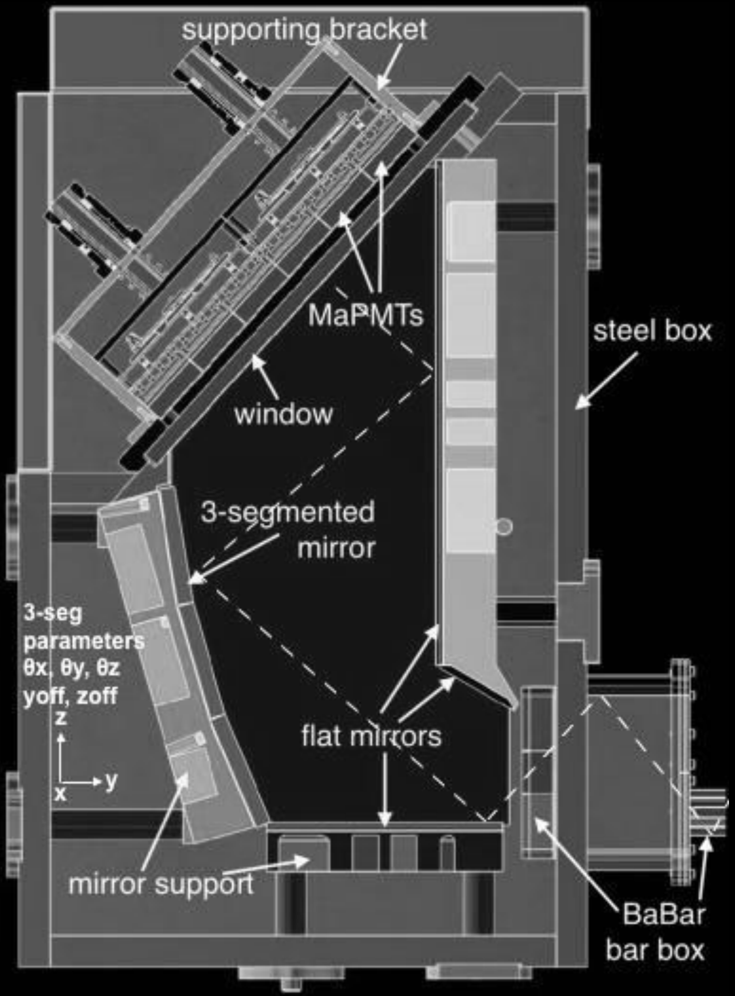

Particle track
Cherenkov photons





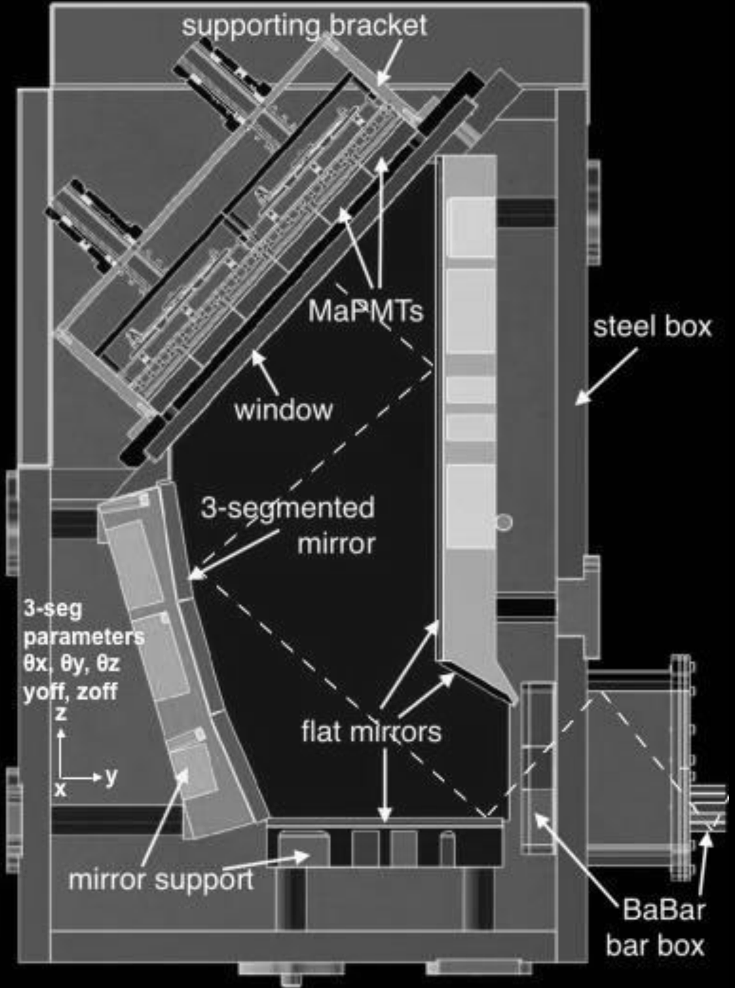
Cherenkov Photon "Ring" in PMT plane

On average about ~30 photons detected per particle

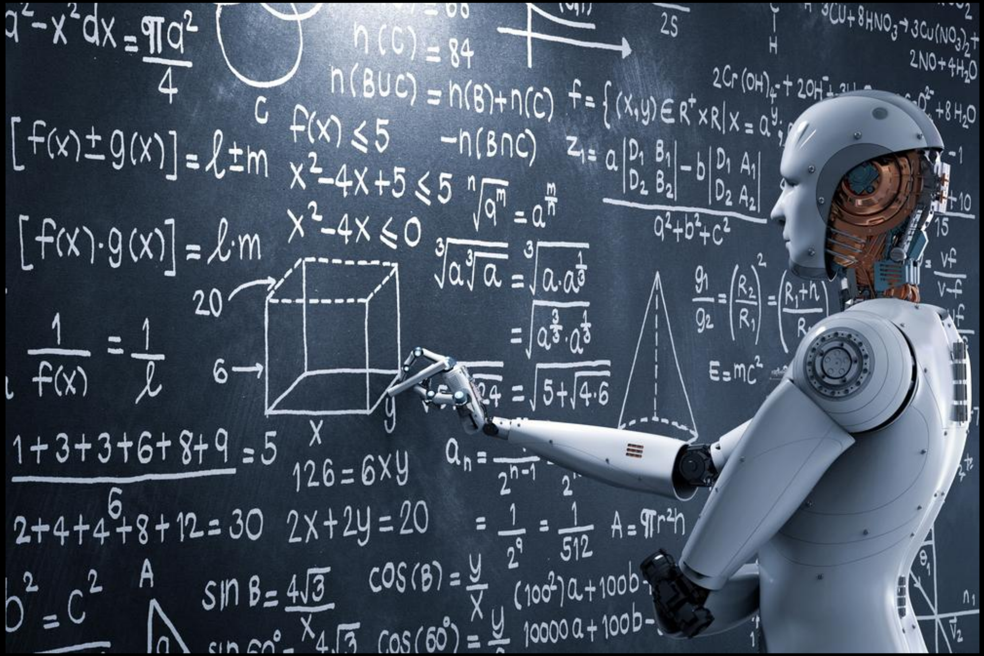


Misalignment

- After installation the optical box will be filled by distilled water (refraction index close to bars).
- Optical box made by several components, system for calibration.
- During data-taking this becomes a black-box problem with many non-differentiable terms.
    - relative alignment of the tracking system with the location and angle of the bars
    - mirrors shifts cause parts of the image change
    - other offsets
- These aspects make seemingly impossible to analytically understand the change in PMT pattern

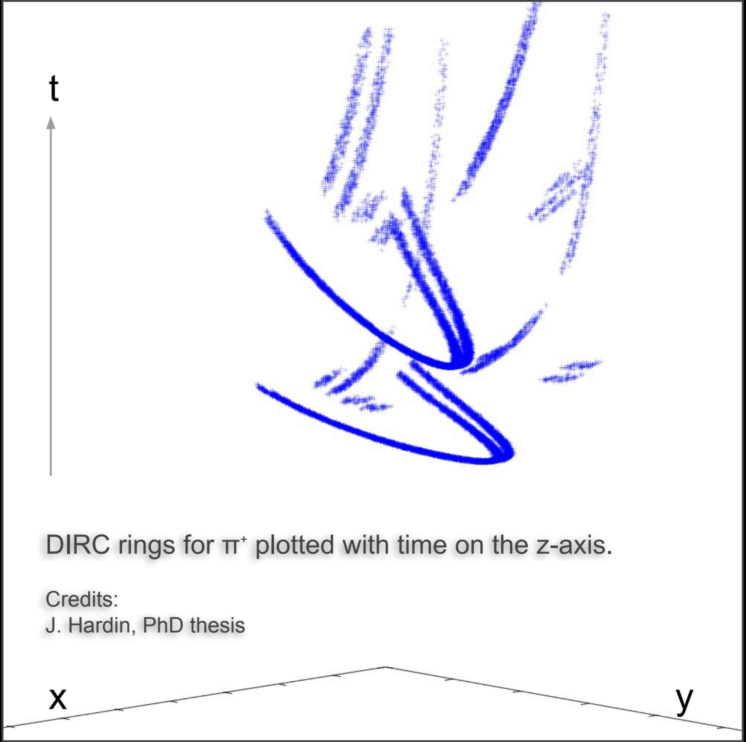


A complex inverse problem



- A grid search with high dimensionality seems less suitable.
- Need of dedicated algorithms for alignment optimization using pure samples of pions from Ks

Hit Patterns



DIRC rings for π^+ plotted with time on the z-axis.

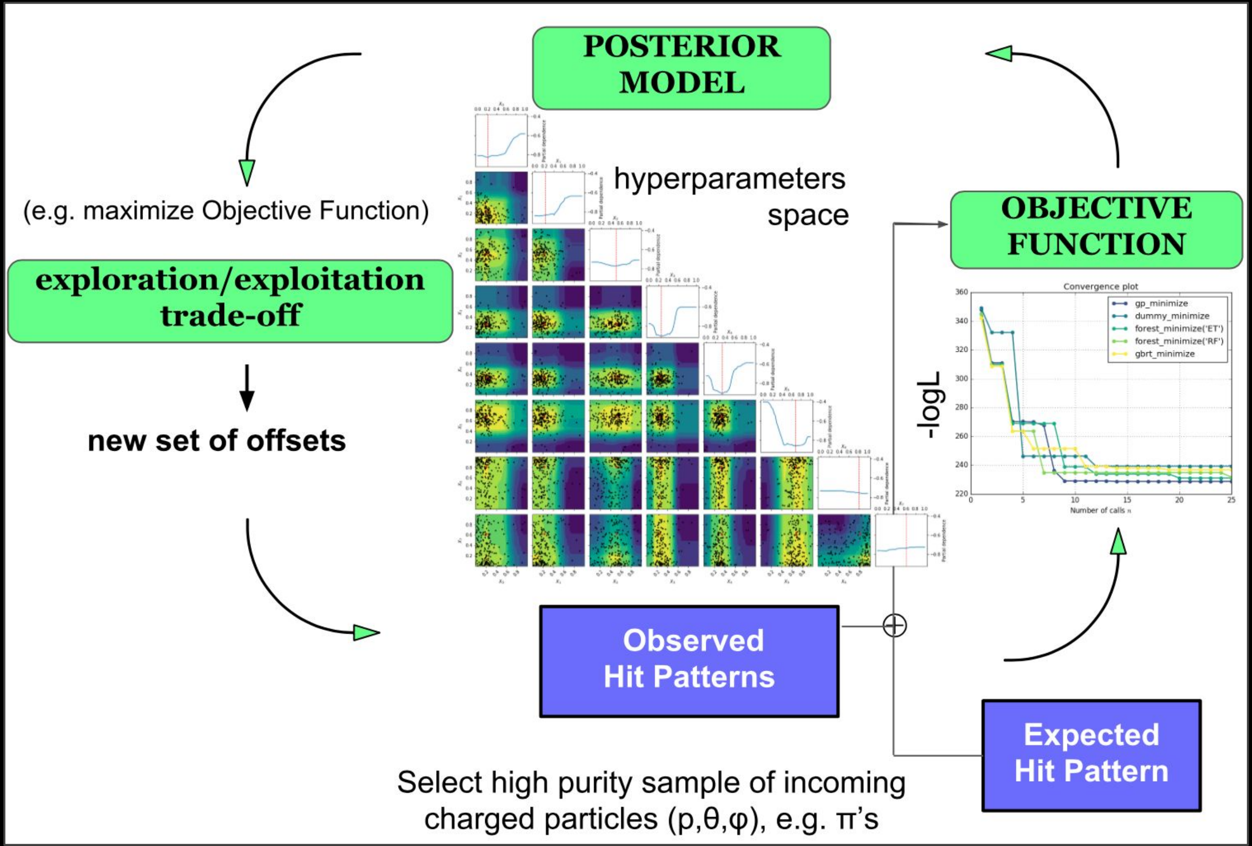
Credits:
J. Hardin, PhD thesis

- 3D (x,y,t) readout and this allows to separate spatial overlaps.
- Patterns take up significant fractions of the PMT in x,y and are read out over 50-100 ns due to propagation time in bars.
- H12700 PMTs have a time resolution of O(200 ps) and read-out electronics giving time information in 1 ns buckets.

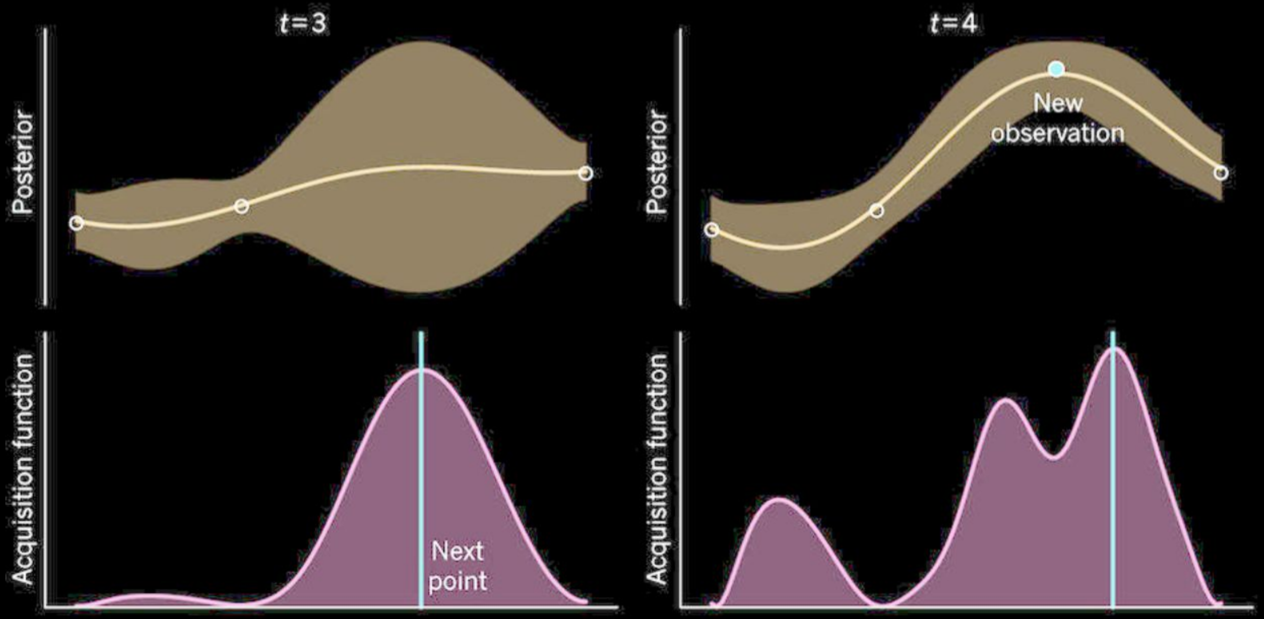
J. Hardin and M. Williams, JINST 11.10 (2016)

Bayesian Optimization

- BO is a strategy for global optimization of **black-box functions**.
- After gathering evaluations BO builds a posterior distribution used to construct an **acquisition function.**
- This determines what is **next query point**.
- BO is agnostic to what is optimized.



Bayesian Optimization



- BO worked amazingly well for tuning MC (Ilten, Williams, Yang [1610.08328]).
- BO could align the DIRC detector leveraging current reconstruction algorithms.

Bayesian Optimization

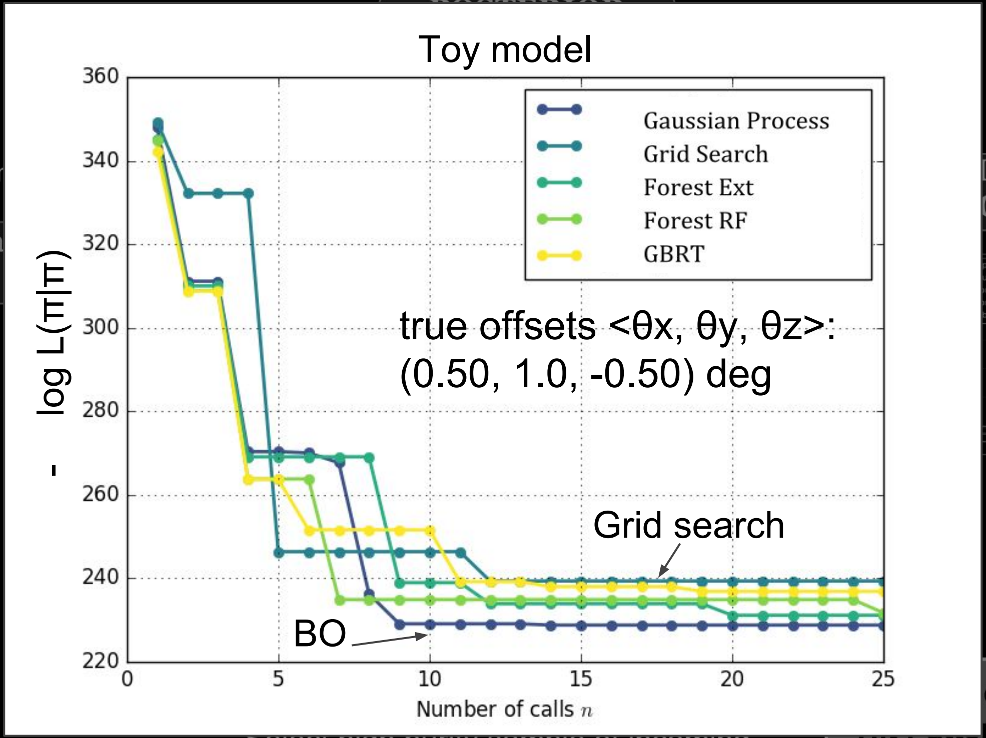
Toy model

- 3 parameters:
- 3-seg mirror θx, θy, θz

Each call based on high purity sample of (only) 100 pions

true:
(0.50, 1.0, -0.50) deg

found:
(0.48, 0.9, -0.44) deg

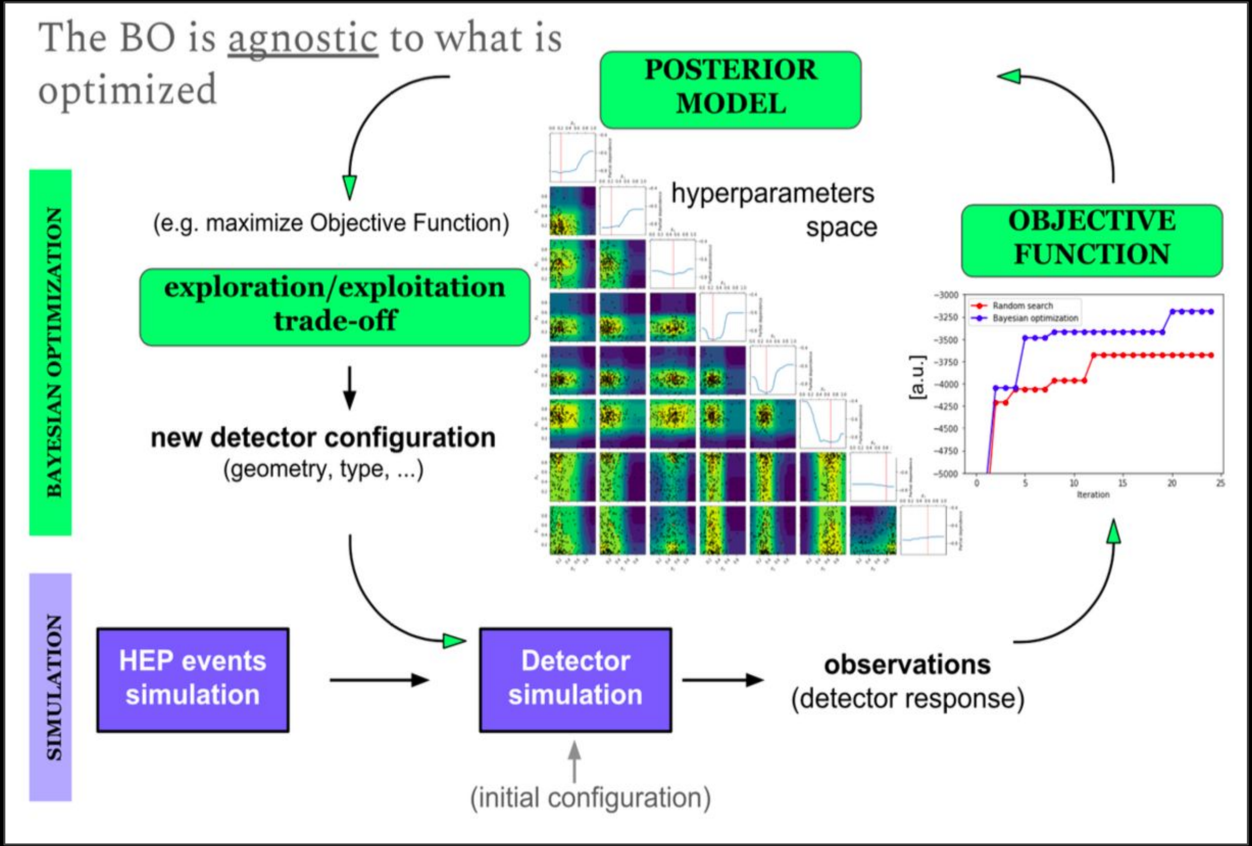


Detector Optimization

- Optimization of detector design is quite complex problem that can be accomplished with BO

- Multi-purpose detector like EIC requires large-scale simulations of the main processes to make decision

- Goal: satisfy detector requirements and minimize cost R&D



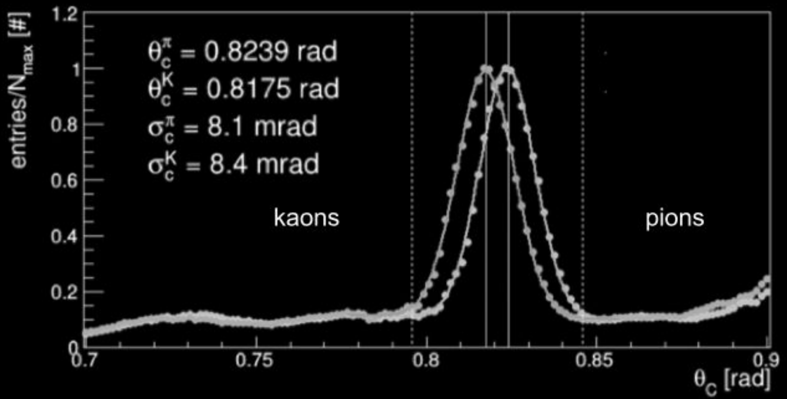
Reconstruction Algorithms and PID

LUT-based geometrical

R. Dzhygadlo et al. Nucl. Instr. And Meth. A, 766:263 (2014)

1. Creation of the LUT: store directions at the end of the radiator for each hit pixel

2. Direction from the LUT for the hit pixels are combined with the track directions (from tracking)

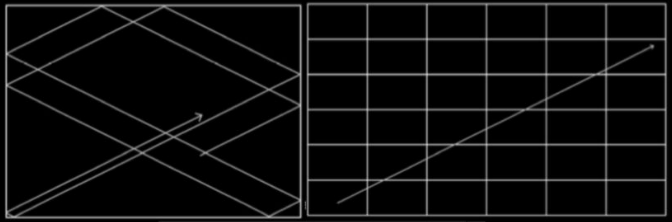


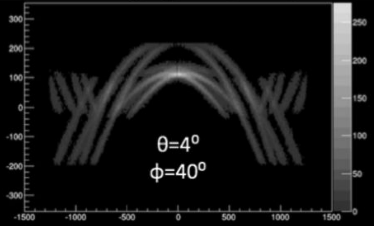
KDE-based

J. Hardin and M. Williams, JINST 11.10 (2016)

Fast tracing mapping straight lines through a tiled plane

1. Generation - 2. Traces through bars - 3. Traces through expansion volume





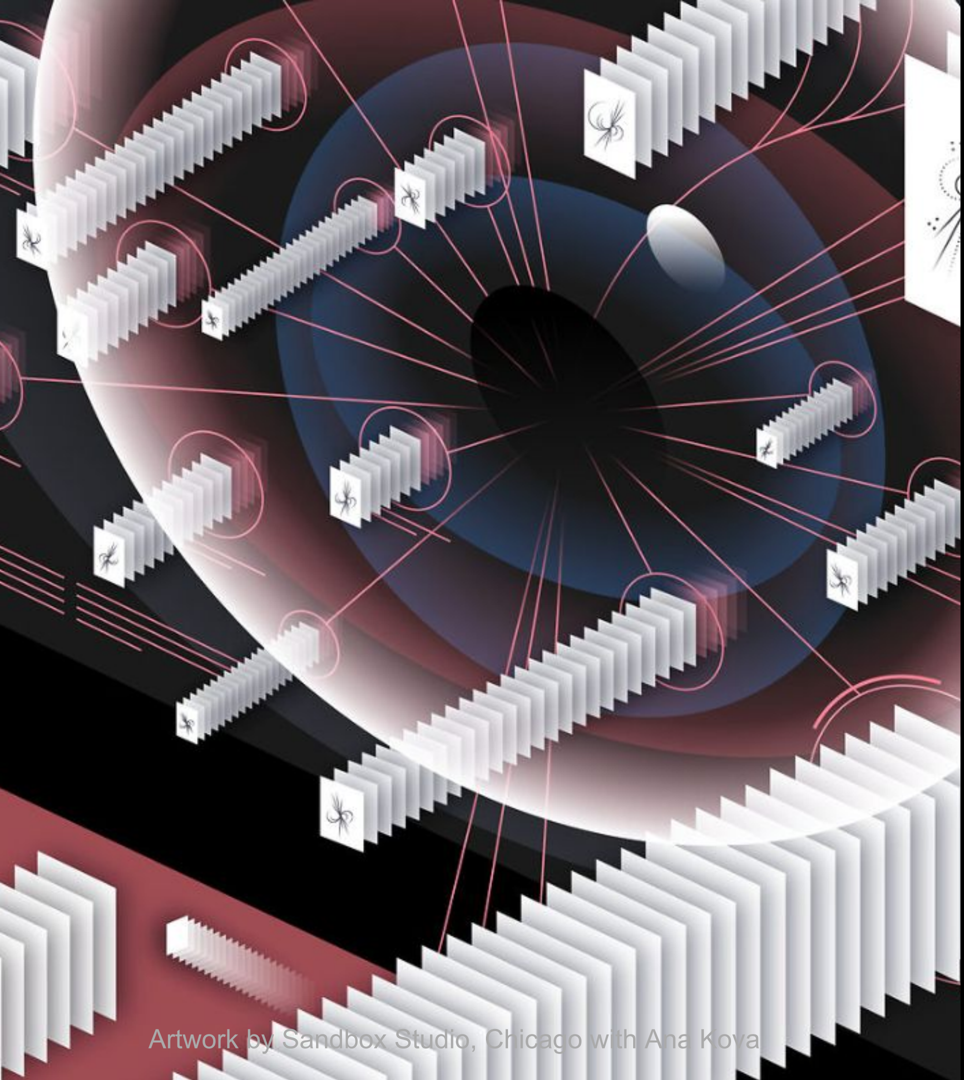
$$P(x) \approx \sum_{i}^{n} K(x - s_i)$$

https://github.com/jmhardin/FasDIRC

basically a trade-off memory/CPU usage

faster reconstruction/hit pattern

better resolution in regions with high overlap

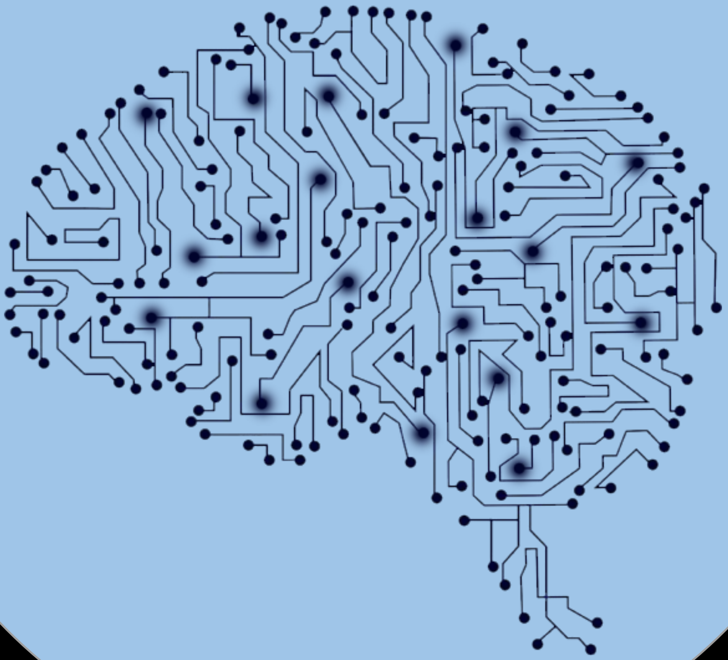

Artwork by Sandbox Studio, Chicago with Ana Kova

Can we use the same technology that facebook uses to recognize faces for doing particle ID?

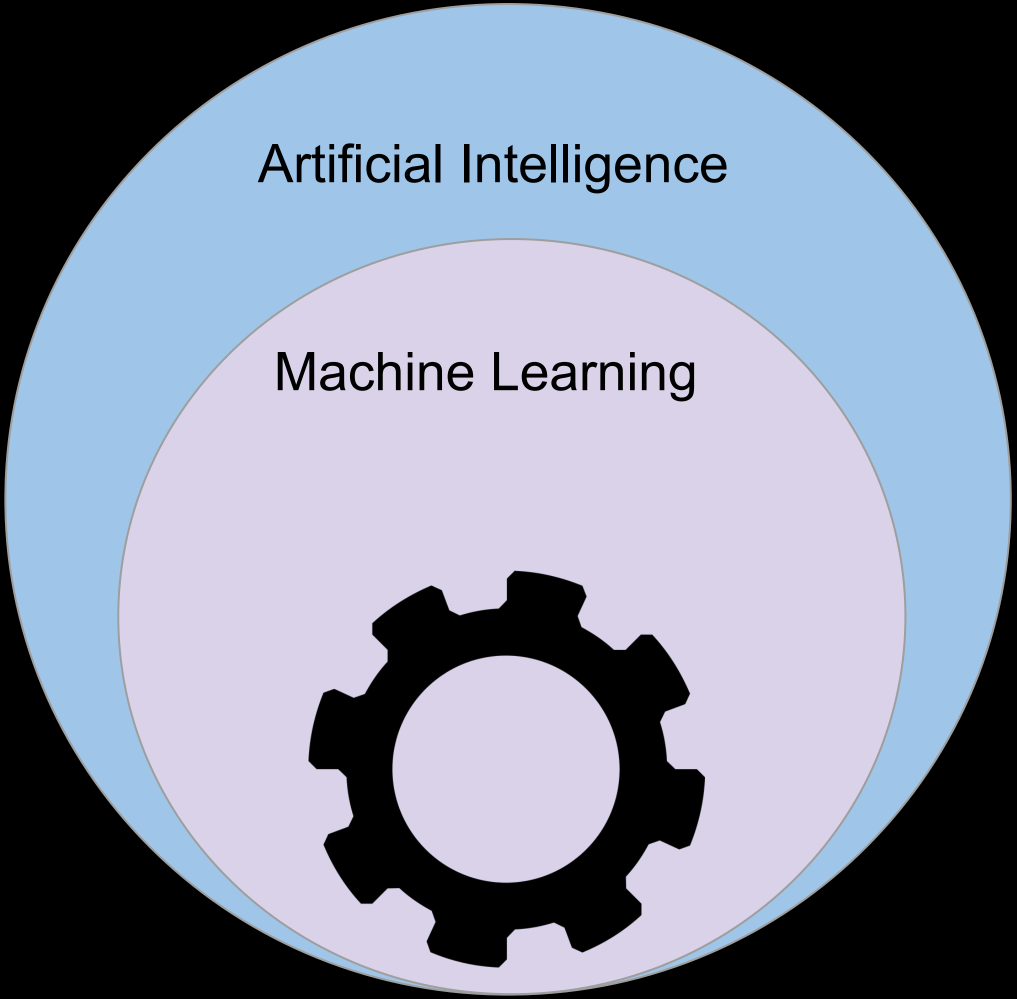
slide format suggested by a ML algorithm

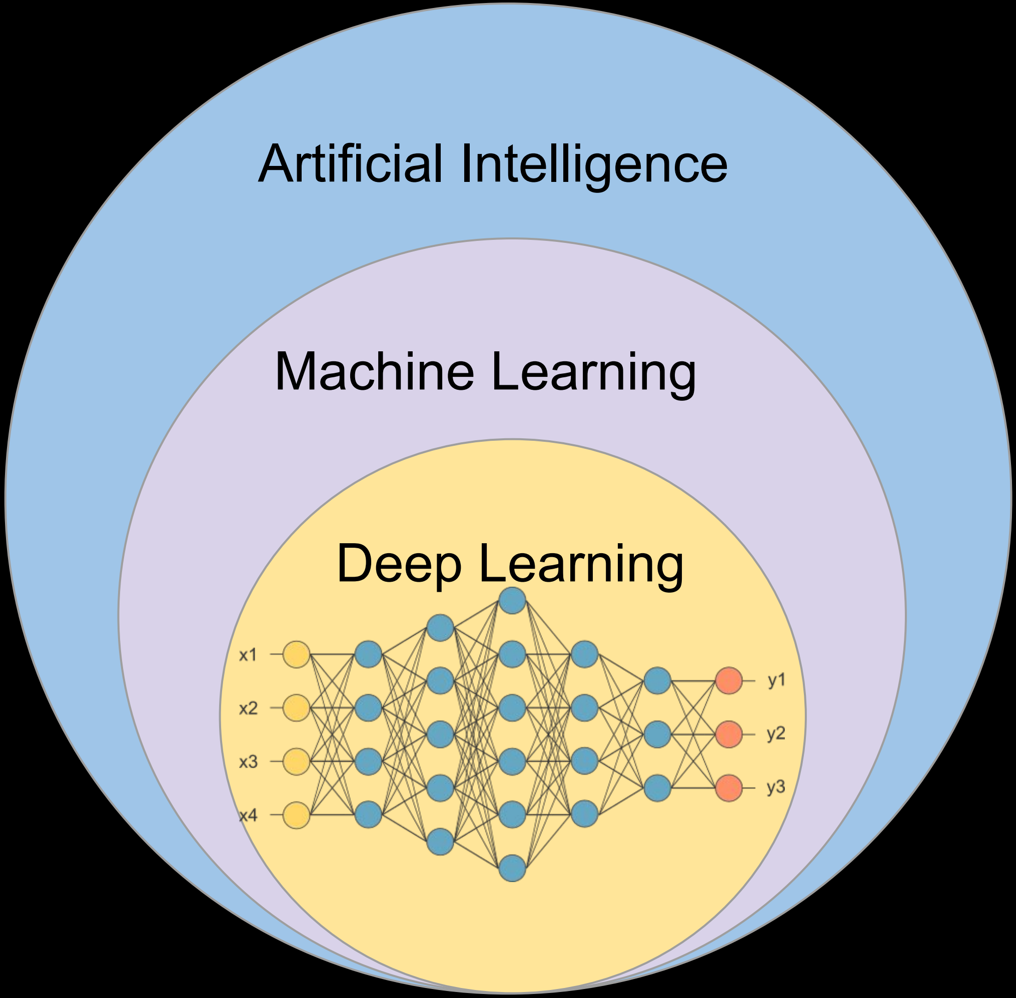
Any technique enabling computers to mimic human behaviour

Artificial Intelligence



A subset of AI based on statistical methods to enable machines to improve with experiences

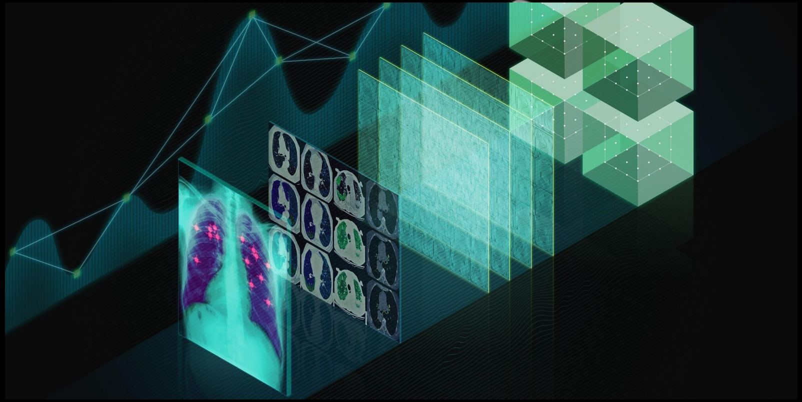




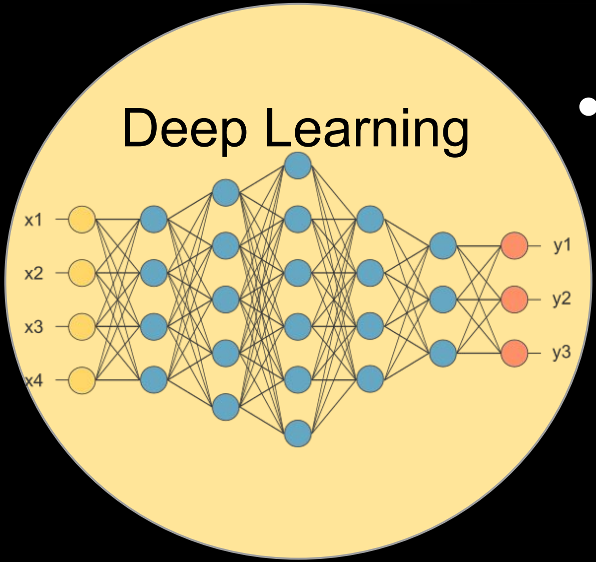
A subset of ML which makes the computation of multi-layer Neural Network feasible

when applied to massive datasets (as particle physics experiments) and giving massive computer power it outperforms all other models most of the time

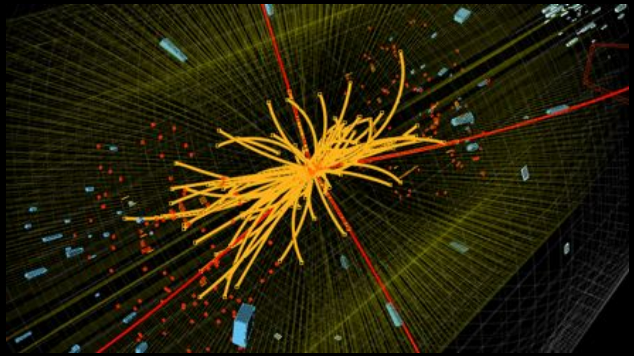
- “Hottest” field in AI, and it’s everywhere



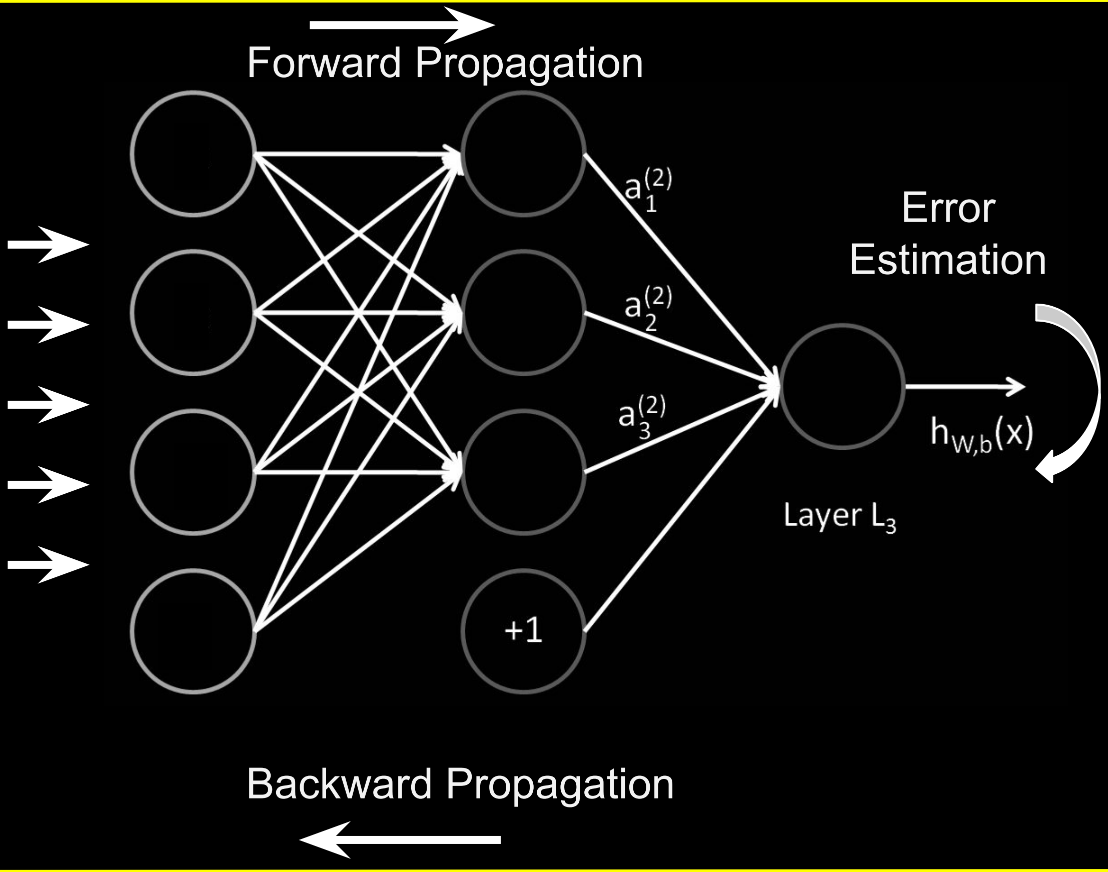
Google



- use of ML (and DL) in HEP is becoming ubiquitous

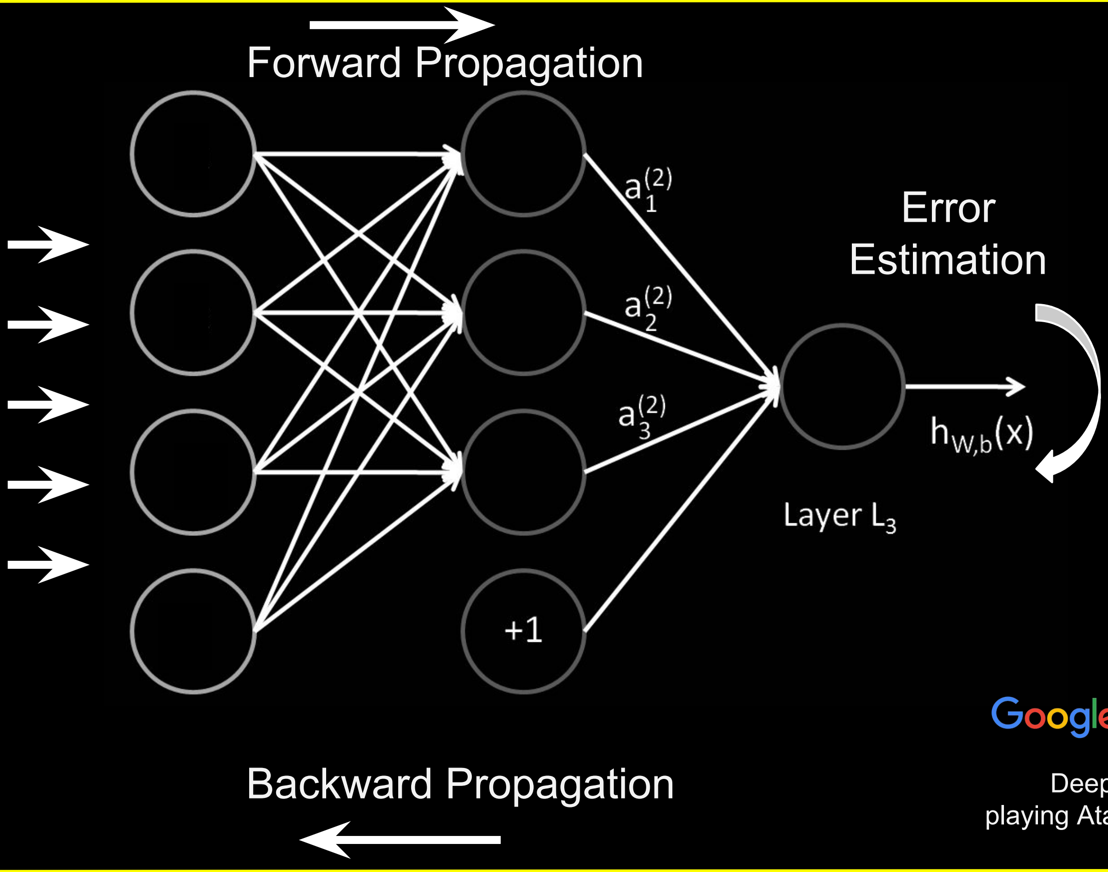


How does it work?



- The real magic about NN is the result of an optimization technique: back-propagation (how a NN works to improve its output over time)
- DL (more hidden) nets are good in learning non-linear functions (heavy processing tasks)
- Based on old school NN revitalized by augmented capabilities (e.g. GPU) and a plethora of new architectures (RNN, CNN, autoencoders, GAN, etc.)

Why does it work?



- Debatable and may seem a dark art (e.g. pruning/dropout neurons, transfer learning)
- No doubts it works…

Google DeepMind

Deep Q-learning playing Atari Breakout

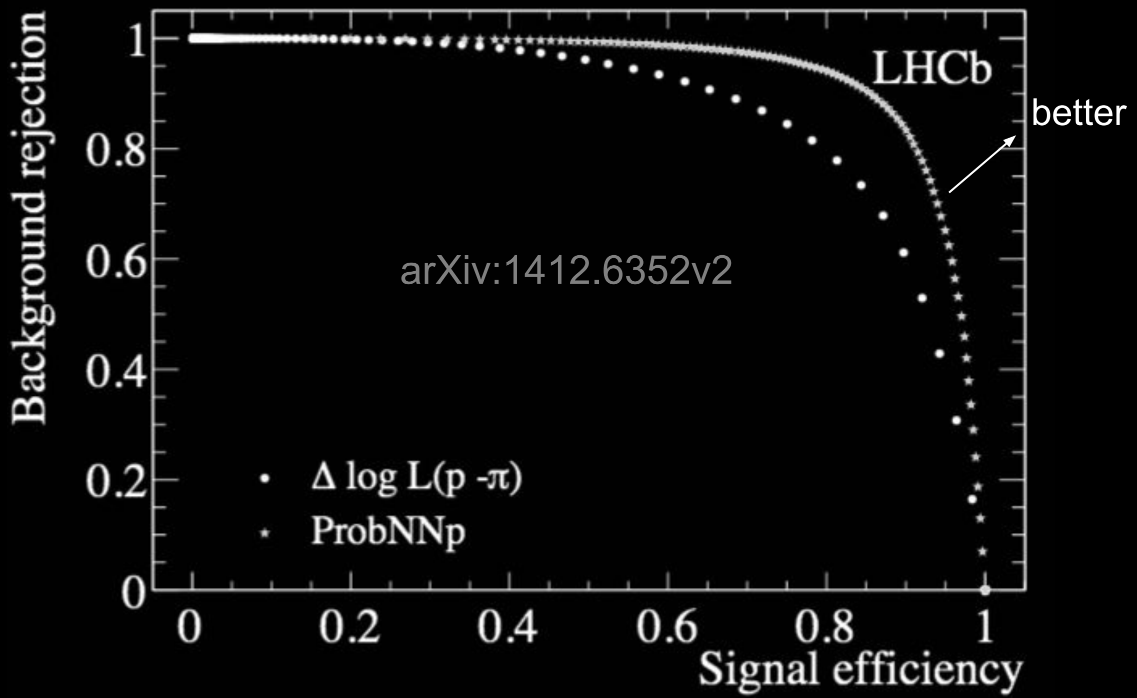
Mnih et al, 1312.5602

Nature, 518.7540 (2015)

Example PID: NN vs Likelihood Approach

- LHCb uses NNs trained on 32 features from all subsystems each of which is trained to identify a specific particle type

- Standard candles are used as calibration samples to characterize performance of NN

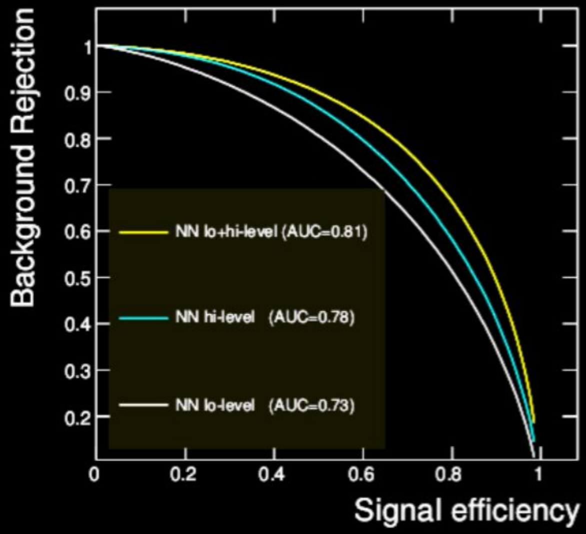


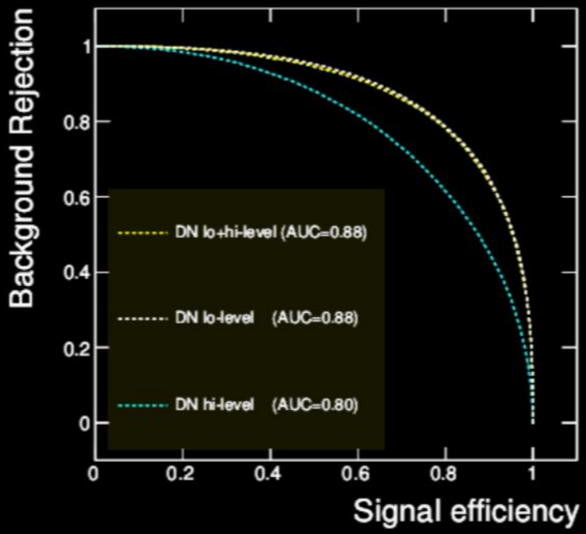
Typically getting ~3x less misID background per particle.

Deep Learning

- In particle physics, our goal is often (if not always) that of distinguishing signal from background.
- We need to build high-level features (e.g. invariant masses) to accomplish this task.
- DNN does not need our “help” and can learn from low-level features.

Baldi, Sadowski, Whiteson arXiv:1402.4735

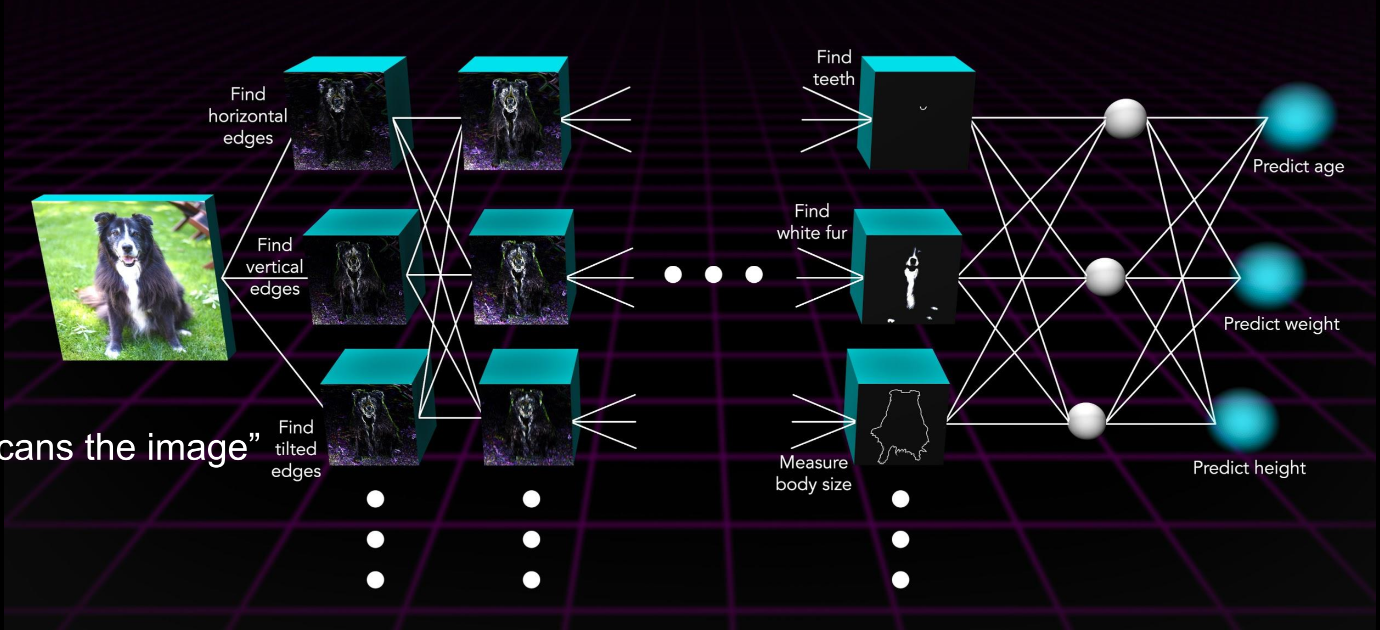




The DNN is able to learn all that it needs in this case, as providing high-level features results in no gains. DNN using low-level features outperforms any selection based only on high-level features.

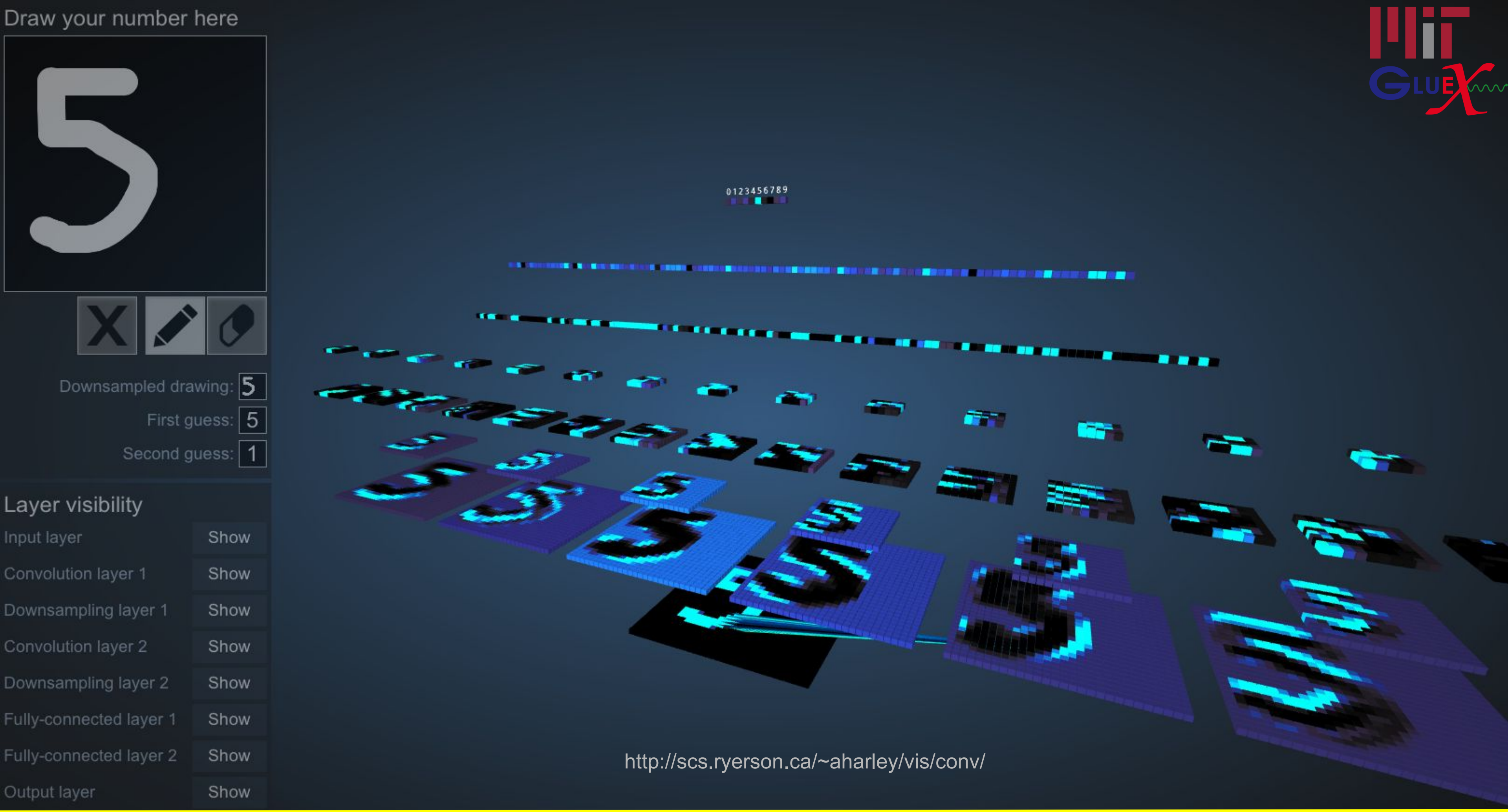
Convolutional Neural Network

- CNN architecture is inspired by human visual cortex. An image can be thought as a group of numbers each describing an intensity value. This can be input in a NN for classification in output.



CNN “scans the image”

- The neurons in a CNN look for local examples of translationally invariant features. This is done using convolutional filters to locate patterns producing maps of simple features, then build complex features using many layers of simple feature maps.

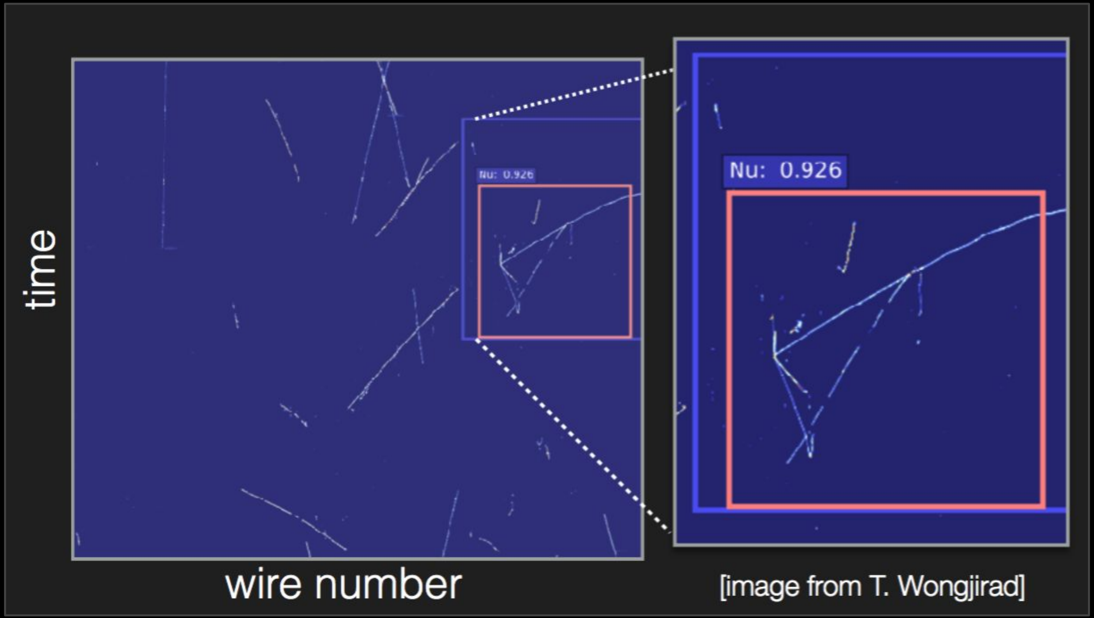


http://scs.ryerson.ca/~aharley/vis/conv/

CNNs for neutrinos

- MicroBoone has managed to train CNNs that can locate neutrino interactions within an event (draw bounding boxes), identify objects and assign pixels to them arXiv 1611.05531

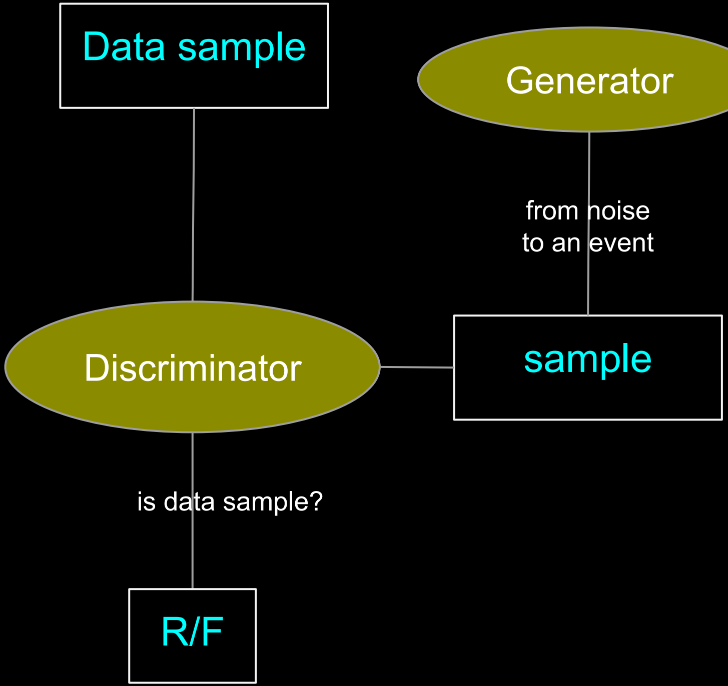
Similar work ongoing at other ν-experiments (see, e.g. , NOvA [1604.01444]), and also at colliders in the area of jet physics [1511.05190], [1603.09349], …)



DNN at NOvA led to an impressive improvement for ν_e-detection
(equivalent to ~ 30% more in exposure than previous PID techniques: $$$)

Generative Adversarial Network

arXiv:1406.2661



Fast Simulations

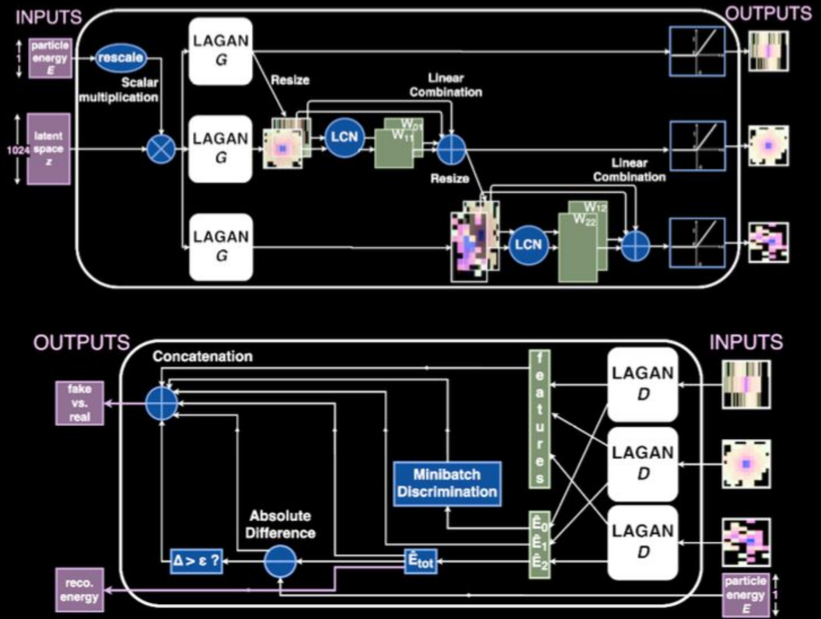
- Detailed simulation of detector response is provided by amazing tools like Geant, which is slow and often prohibitive for generating large enough samples.
- Cutting-edge application of deep learning uses GAN for fast simulation.
- 2-NN game, one model maps noise to images, the other classifies the images if real or fake.
- The goal is to confuse the discriminator.

- CALOGAN: Paganini, de Oliveira, Nachman 1705.02355
- jet images production: 1701.05927

CALOGAN can generate the reconstructed CALO image using random noise, skipping the GEANT and RECO steps

Generative Adversarial Network

arXiv:1406.2661



Fast Simulations

- Detailed simulation of detector response is provided by amazing tools like Geant, which is slow and often prohibitive for generating large enough samples.
- Cutting-edge application of deep learning uses GAN for fast simulation.
- 2-NN game, one model maps noise to images, the other classifies the images if real or fake.
- <u>The goal is to confuse the discriminator</u>.

- CALOGAN: Paganini, de Oliveira, Nachman 1705.02355
- jet images production: 1701.05927

CALOGAN can generate the reconstructed CALO image using random noise, skipping the GEANT and RECO steps

Deliverables and timeline

- DIRC alignment with BO. ~ ½ year
- Explore deep nets architectures for GlueX DIRC/PID; determination of best approaches enhanced by BO hyperparameter tuning (on a longer term ~ 1 year).
- Collaboration: MIT/DIRC group.
- Resources: deep learning workstation.

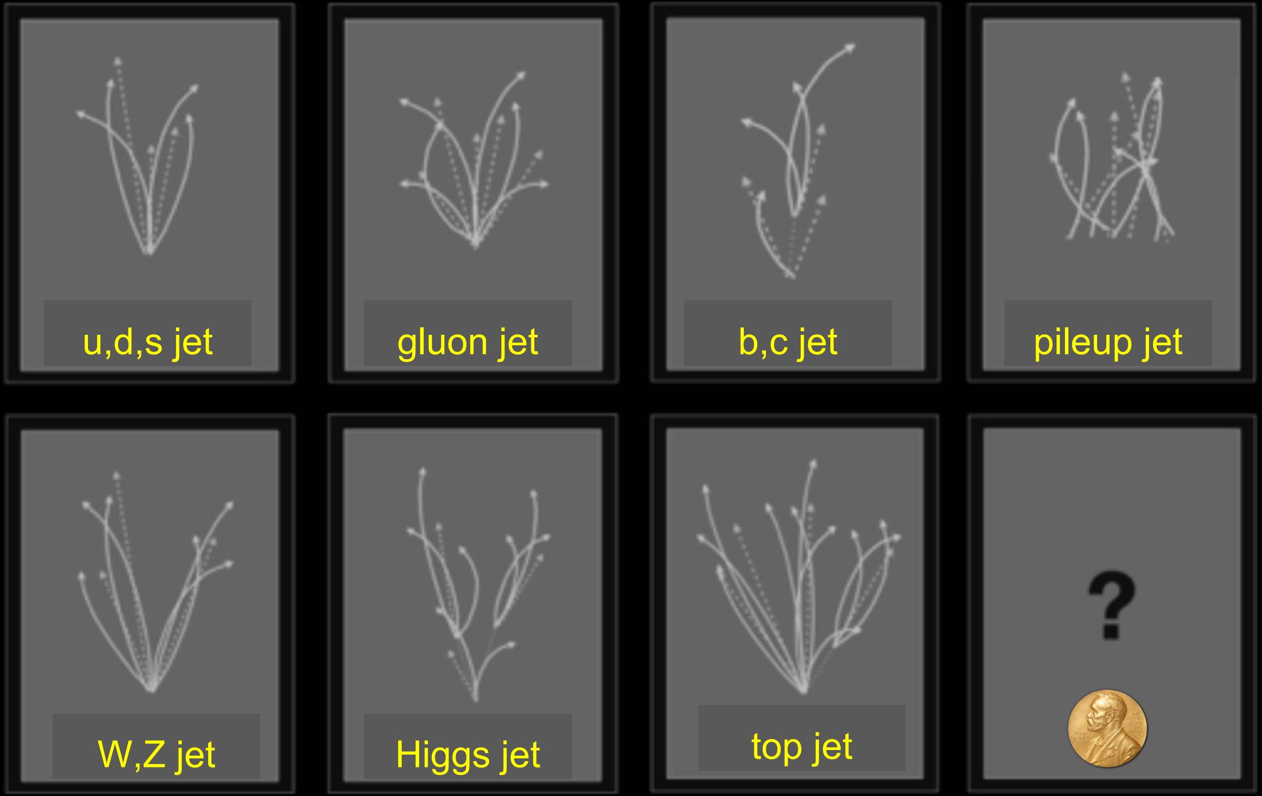
Summary

- Use of ML became ubiquitous in particle physics. Deep Learning is starting to make an impact.
- A lot of exciting cutting-edge work not discussed, e.g., DKF, DNNs on FPGAs, automatic anomaly detection, compression using autoencoders, etc.
- Systematics are vital in our field: we are developing systematics aware ML algorithms and we are characterizing “black boxes” (e.g. DIRC optical box, EIC detector design).
- Beyond the issue of systematics, our data have other interesting features from a CS perspective: sparse data, irregularities in detector geometries, heterogeneous information, physical symmetries and conservation laws (e.g. recursive NN), etc.

We just began to scratch the surface when it comes to using tools like BO and DL and recent achievements in our field (e.g. NOvA, LHCb, ...) suggest it’s worth exploring these strategies.

BACKUP

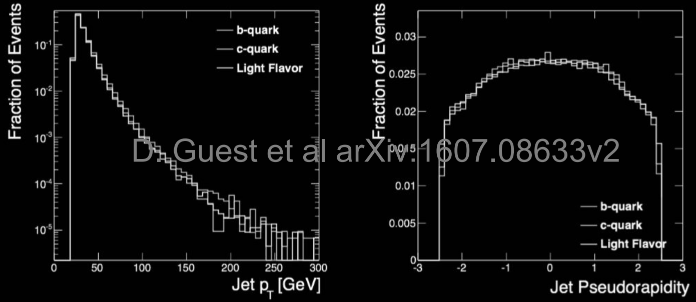
Deep jet tagging



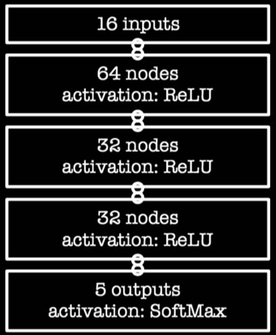
- A jet in LHC is a spray of hadrons from shower initiated by some fundamental particle
- Define set of features with distributions depending on the jet nature
- Train NN using a sample of jets whose nature is known

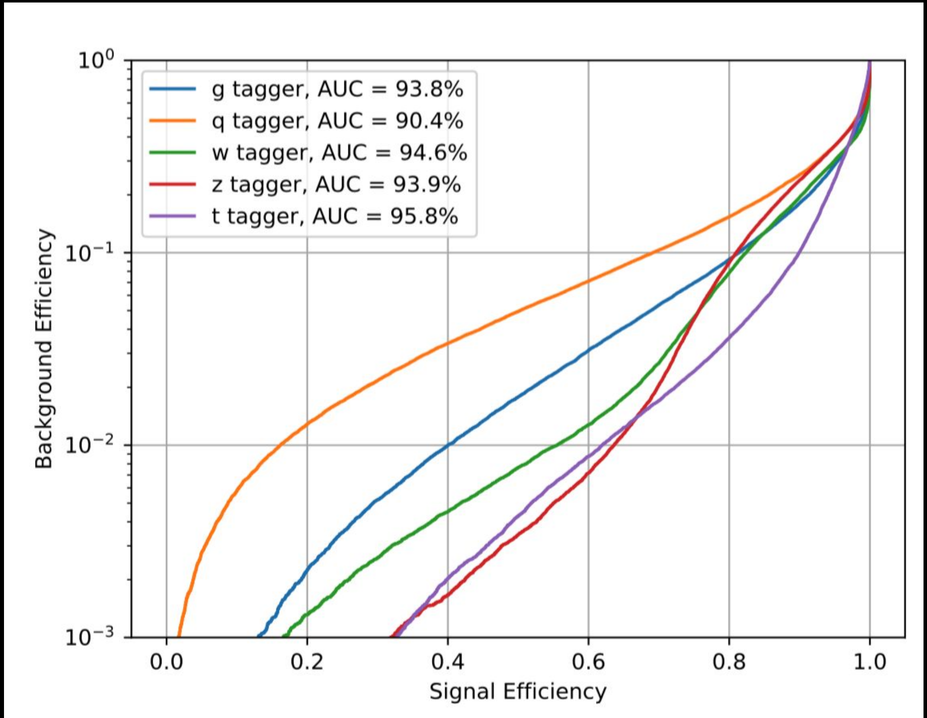
Deep jet tagging

- DNN based on high-level features (jet masses, multiplicities, energy correlation functions, etc.)



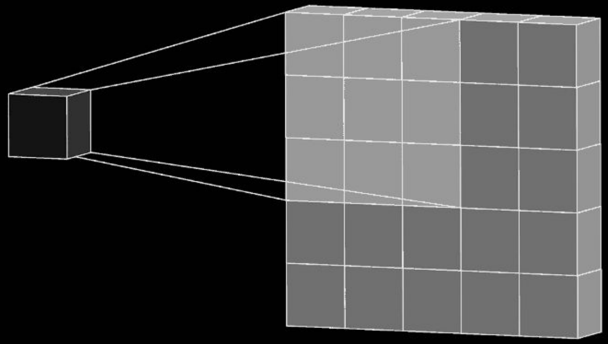
D. Guest et al arXiv:1607.08633v2





J. Duarte et al arXiv:1804.06913v2

Intuitively understanding CNN

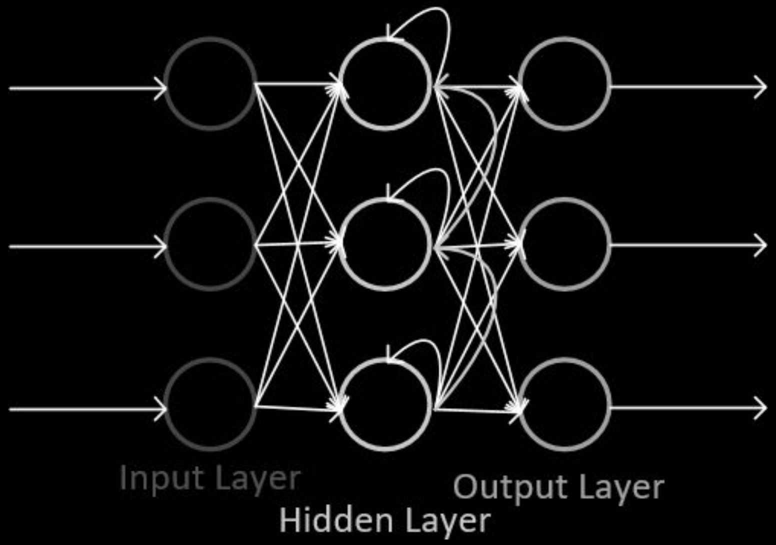


3_0	3_1	2_2	1	0
0_2	0_2	1_0	3	1
3_0	1_1	2_2	2	3
2	0	0	2	2
2	0	0	0	1

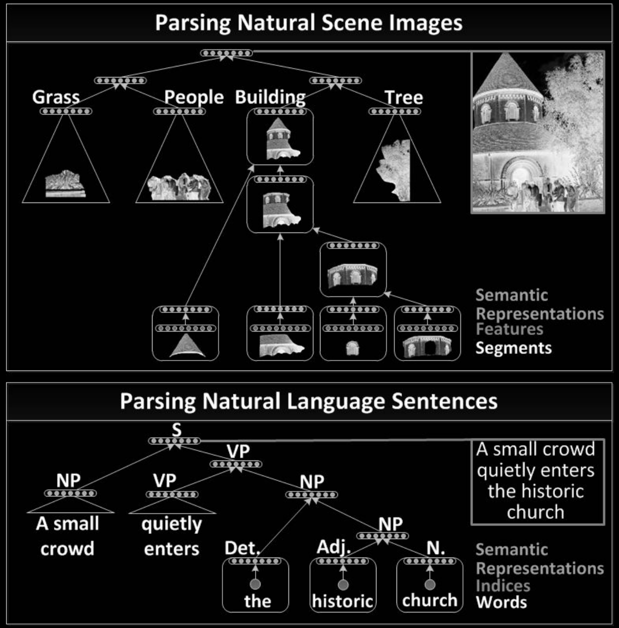
12.0	12.0	17.0
10.0	17.0	19.0
9.0	6.0	14.0

Other Architectures: Recurrent & Recursive NN

Recurrent Neural Network



Recursive Neural Network

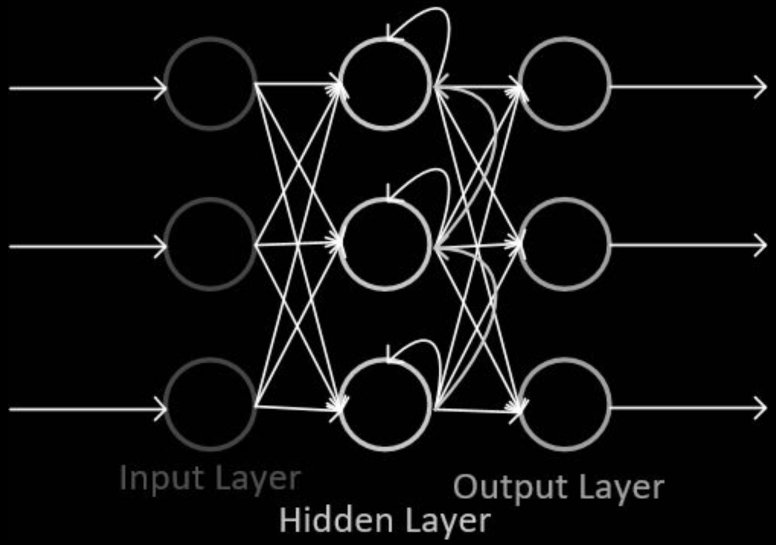


- Example: jet physics.
- More efficient than image-based networks.

G. Louppe, K. Cho, C. Becot, and K. Cranmer 1702.00748v1

Other Architectures: Recurrent & Recursive NN

Recurrent Neural Network



Recursive Neural Network

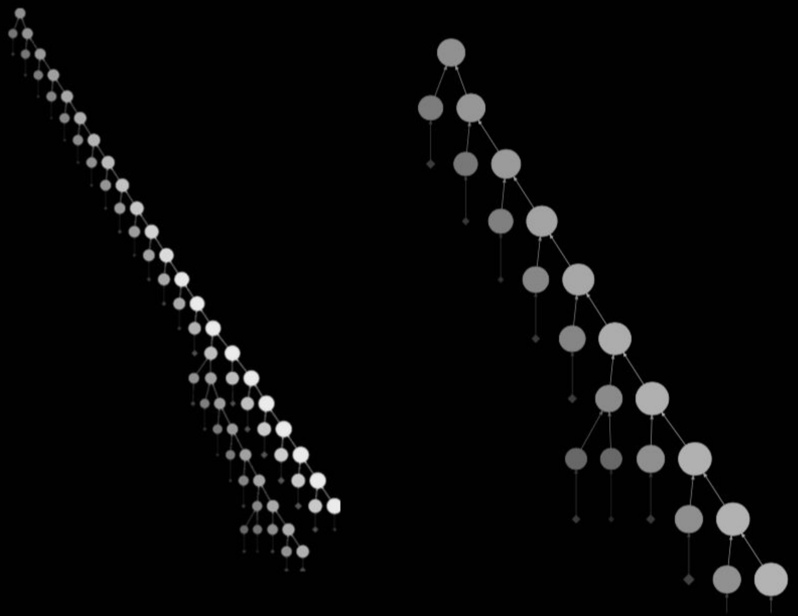
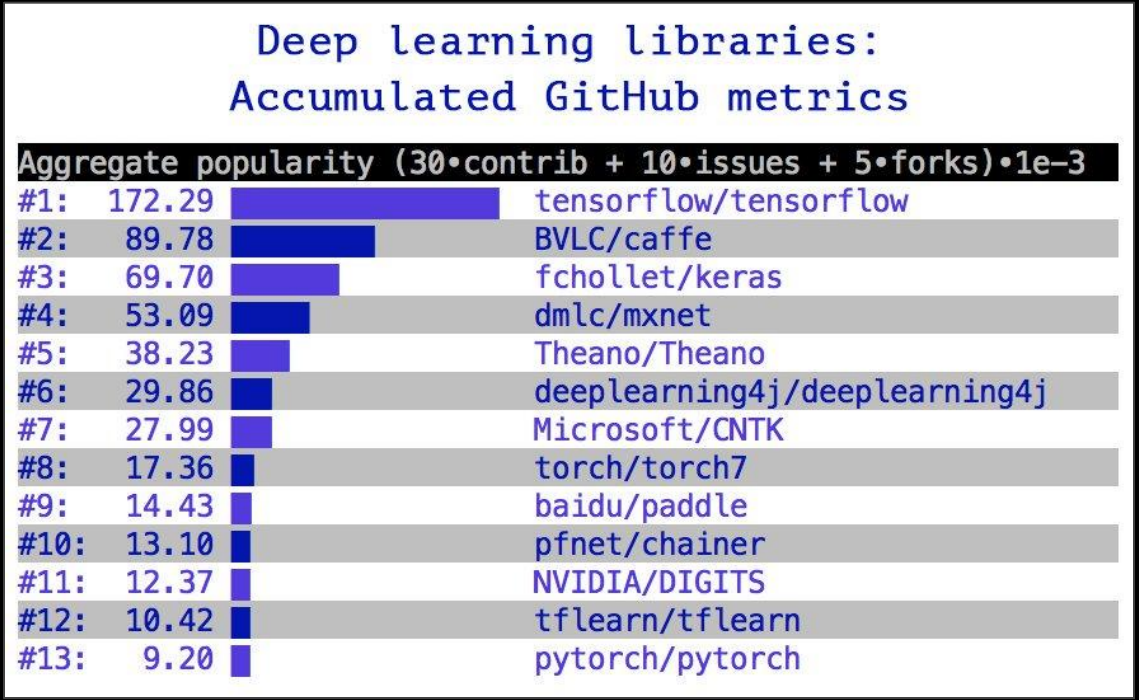


Figure 2. Typical tree structures for 1 TeV gluon jet (left) and quark jet(right).

T. Cheng [1711.02633v1]

Frameworks



Deep learning libraries:
Accumulated GitHub metrics
Aggregate popularity (30•contrib + 10•issues + 5•forks)•1e-3
#1: 172.29 tensorflow/tensorflow
#2: 89.78 BVLC/caffe
#3: 69.70 fchollet/keras
#4: 53.09 dmlc/mxnet
#5: 38.23 Theano/Theano
#6: 29.86 deeplearning4j/deeplearning4j
#7: 27.99 Microsoft/CNTK
#8: 17.36 torch/torch7
#9: 14.43 baidu/paddle
#10: 13.10 pfnet/chainer
#11: 12.37 NVIDIA/DIGITS
#12: 10.42 tflearn/tflearn
#13: 9.20 pytorch/pytorch

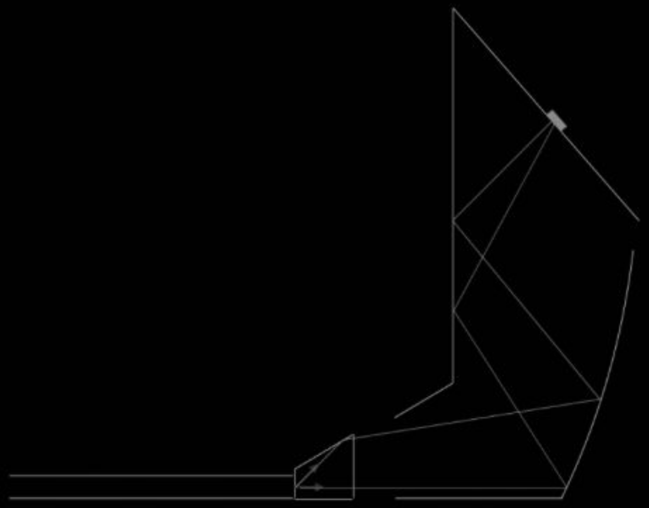
Geometrical Reconstruction

See *R. Dzhygadlo et al. Nucl. Instr. And Meth. A, 766:263, 2014*

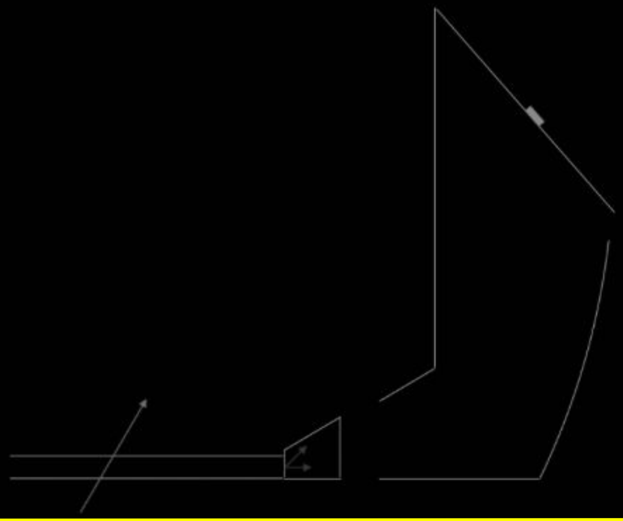
BABAR-like

Has two stages:

1. Creation of the look-up table (LUT): store directions at the end of the radiator for each hit pixel



2. Directions from the LUT for the hit pixels are combined with the track direction (from the tracking system)



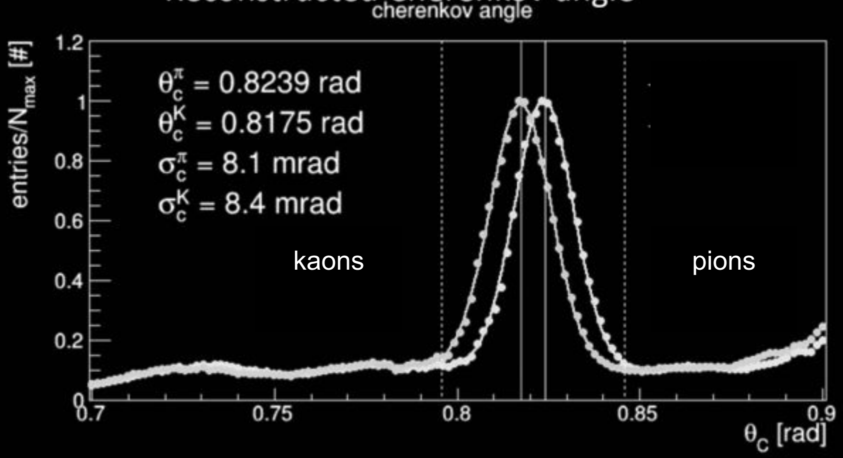
Geometrical Reconstruction

Kaons and pions with p = 4 GeV/c, theta = 11°, phi = 90°

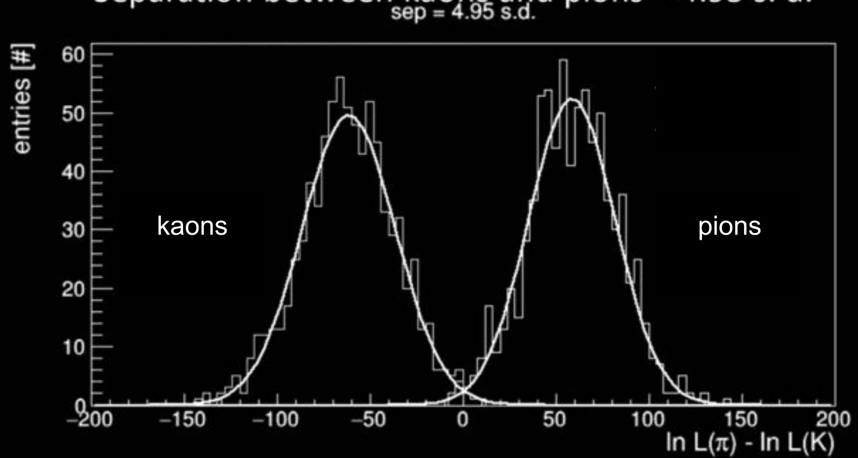
Log likelihood is based on the Cherenkov angle and the number of detected photons

Timing information is used to cut out some solutions

Reconstructed Cherenkov angle



Separation between kaons and pions = 4.95 s. d.



$$\sigma_c^2 = \sigma_{tr}^2 + \frac{\sigma^2}{N}$$

Design goal: >= 3 s.d. between pions and kaons for momenta up to 4 GeV/c

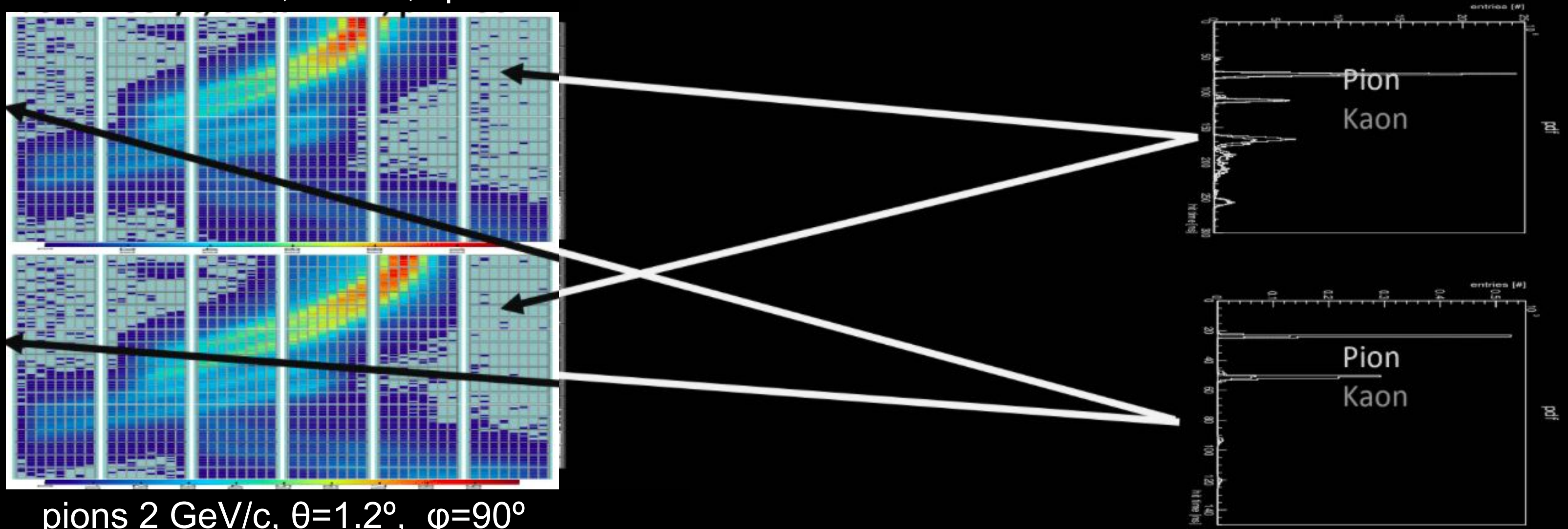
Time-based imaging

Calculate log likelihoods for each particle hypothesis directly from the time spectrum in each hit pixel

In advance need time spectra for each particle type and the given track configuration (momentum, direction)

Such time spectra can be **simulated or calculated analytically**

kaons 2 GeV/c, θ=1.2º, φ=90º

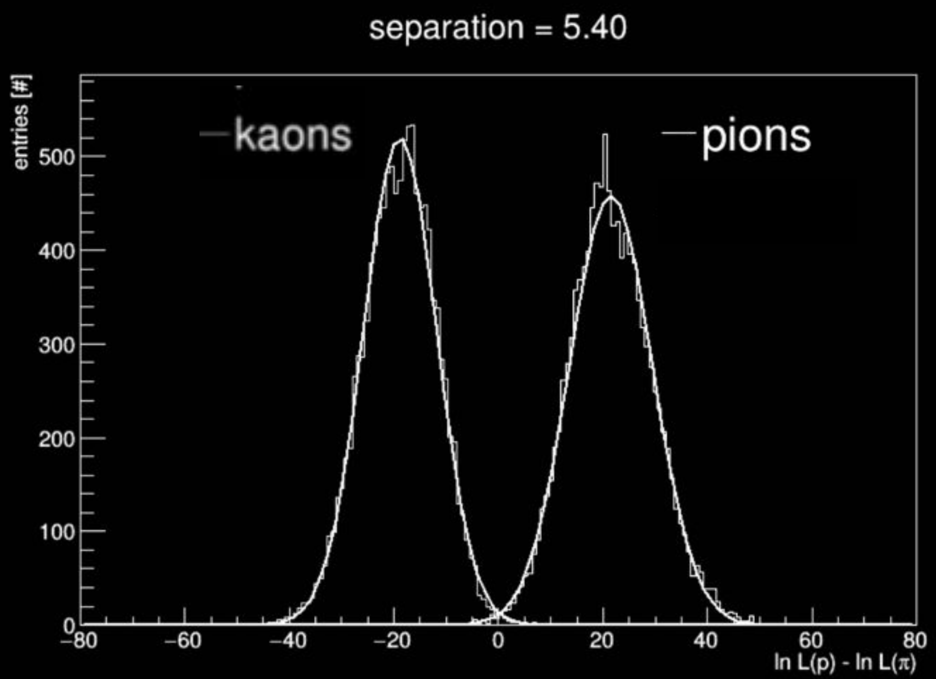


pions 2 GeV/c, θ=1.2º, φ=90º

Time-based imaging

4 GeV/c pions and kaons, theta = 1.2°, phi = 90°

$$\log\mathcal{L}_h = \sum_{i=1}^{N} \log\left(\frac{S_h(x_i,y_i,t_i)+B(x_i,y_i,t_i)}{N_e}\right) + \log P_N(N_e)$$



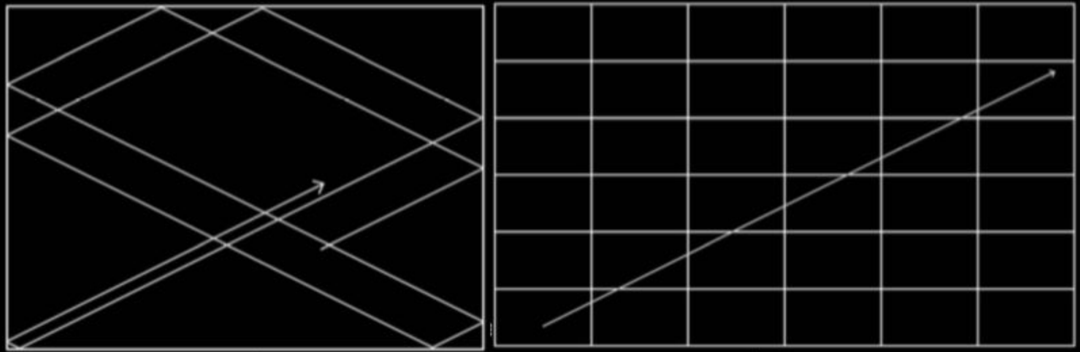
Right now we simulate PDFs

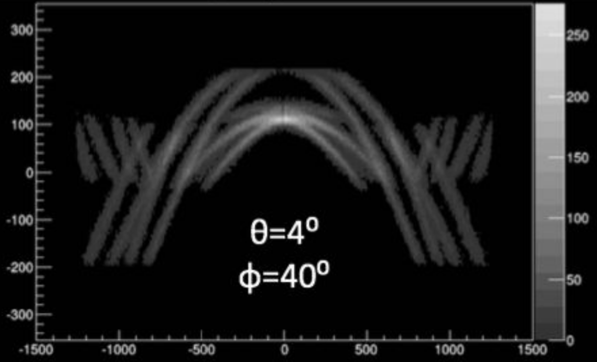
Belle II TOP uses **analytical PDFs**, and we plan to try this method

FastDIRC - KDE based Reconstruction

- Use Kernel Density Estimation to compare particle hypothesis patterns
- Use fast tracing to implement KDE
  - O(1) speed vs O(100) bounces
- Compare Log Likelihoods

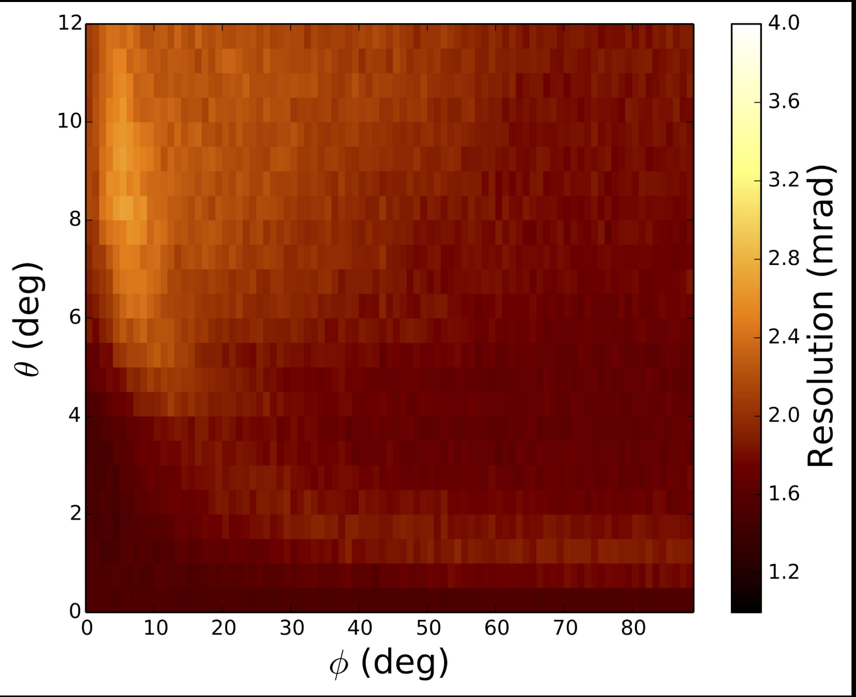
$$P(x) \approx \sum_{i}^{n} K(x - s_i)$$

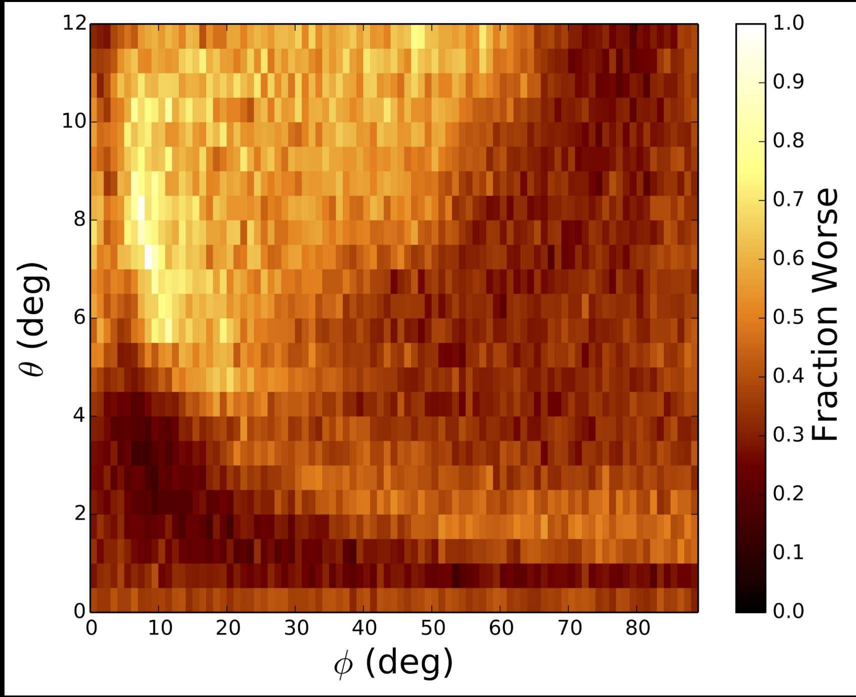




See the FastDIRC paper (J Hardin, M Williams 1608.01180)
Github: https://github.com/jmhardin/FastDIRC

FastDIRC - KDE based Reconstruction





FastDIRC

Fraction Worse = (LUTres - FastDIRCres)/LUTres