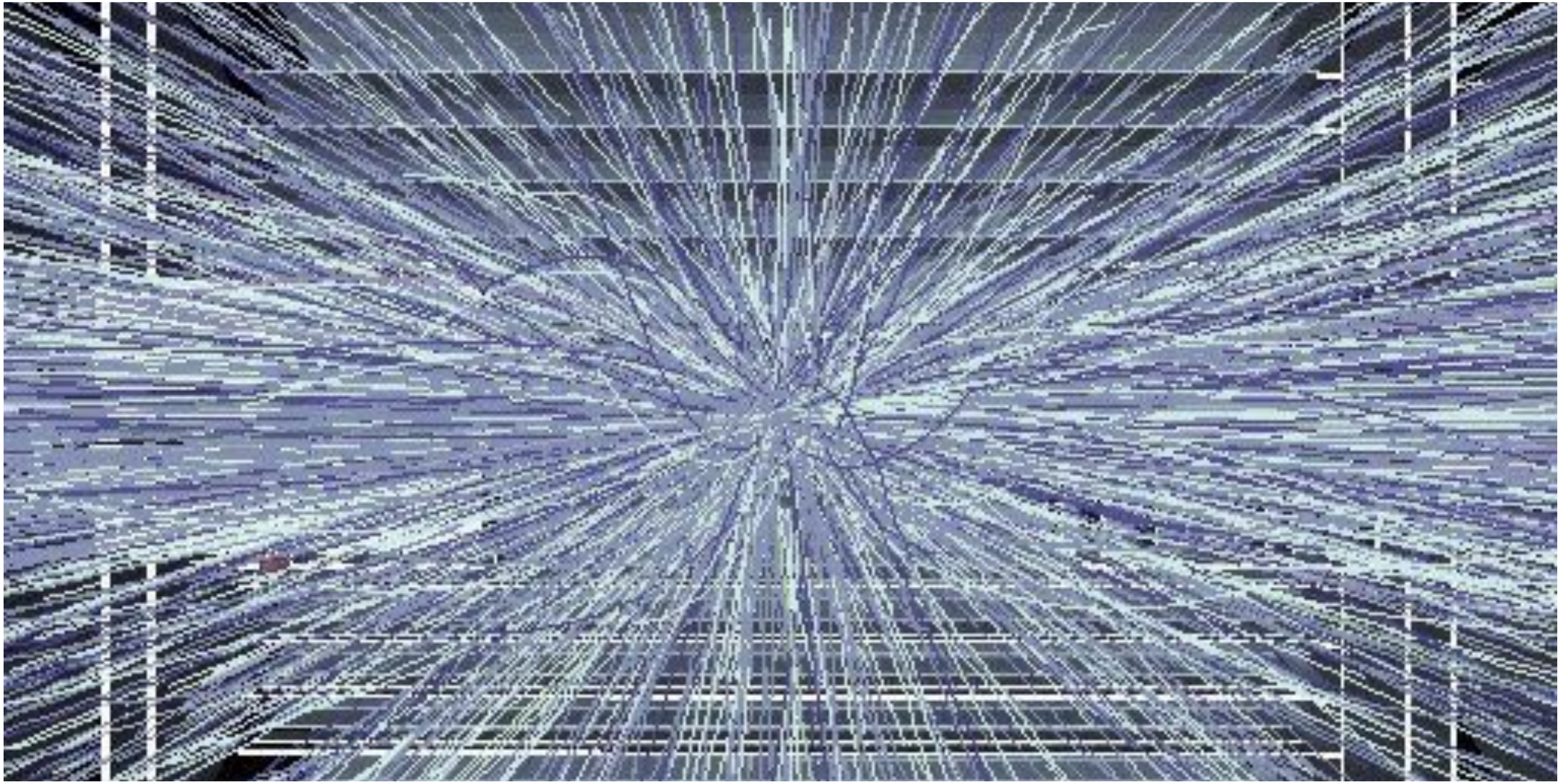
Deep Learning for the future of the Large Hadron Collider

Maurizio Pierini

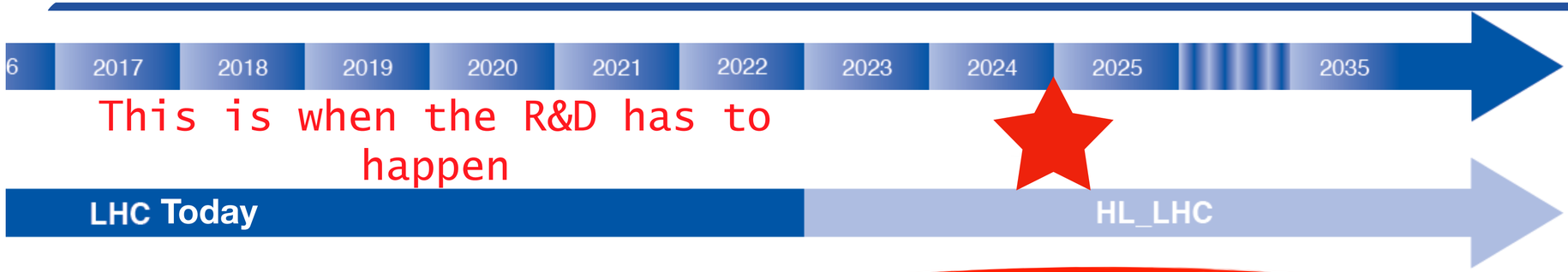
SOSC 2018, Perugia



ML and HEP future challenges

European Research Council

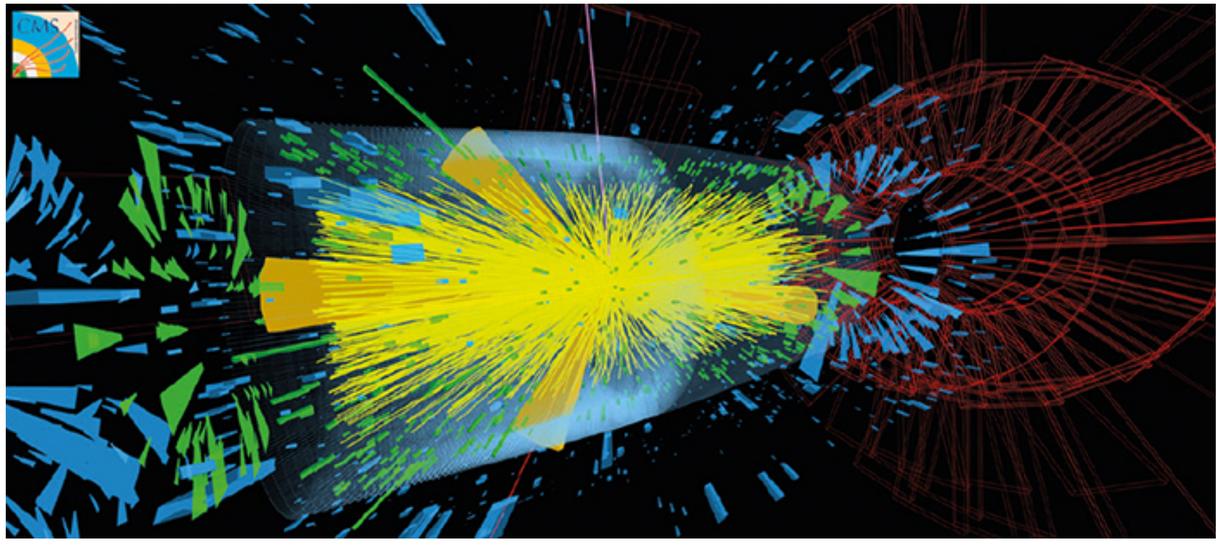
HL-LHC: elephant in the room



- ~40 collisions/event
- ~10 sec/event processing time
- (at best)Same computing resources as today

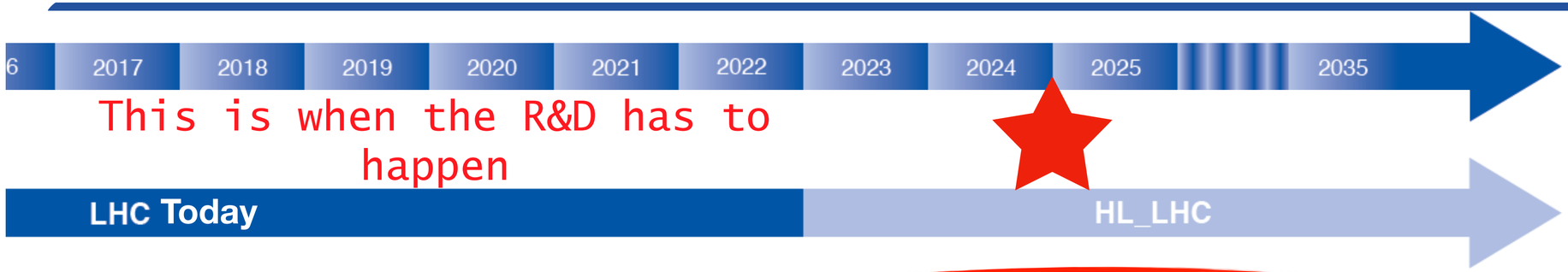
- ~200 collisions/event
- ~minute/event processing time(*)
- (at best)Same computing resources as today

- *Flat budget vs. more needs = **current rule-based reconstruction algorithms will not be sustainable***
- ***Adopted solution:*** *more granular and complex detectors → more computing resources needed → more problems*
- ***Modern Machine Learning might be the way out***



(*)With nowadays software development

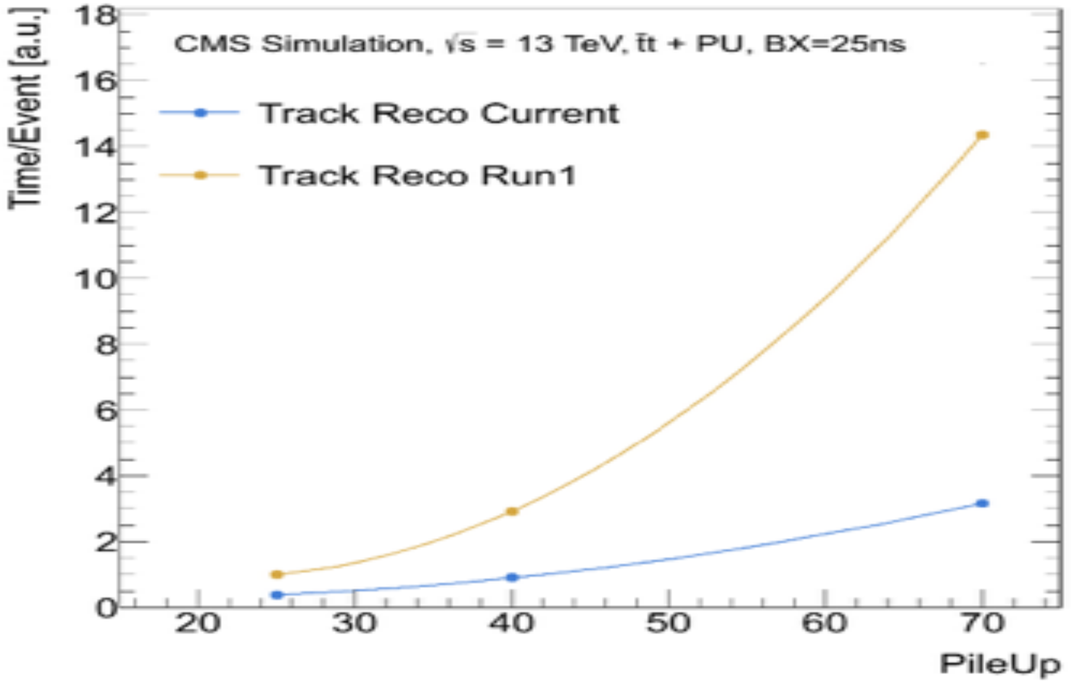
HL-LHC: elephant in the room



- ~40 collisions/event
- ~10 sec/event processing time
- (at best)Same computing resources as today

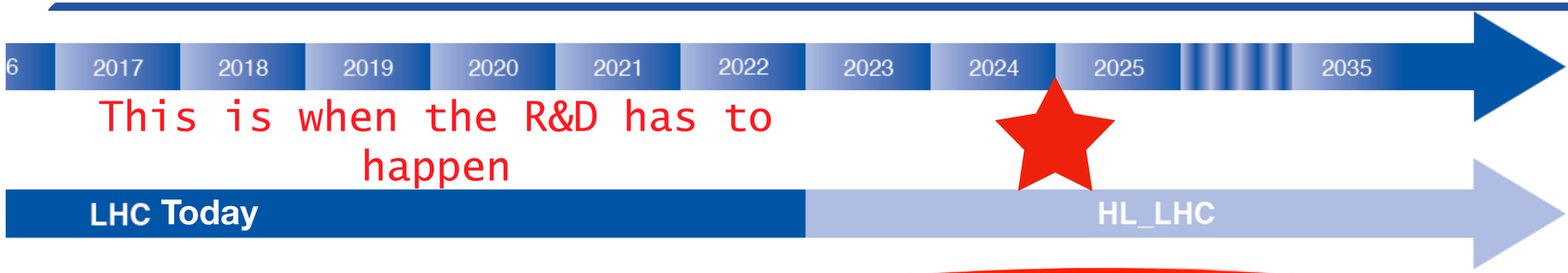
- ~200 collisions/event
- ~minute/event processing time(*)
- (at best)Same computing resources as

- *Flat budget vs. more needs = **current rule-based reconstruction algorithms will not be sustainable***
- ***Adopted solution:** more granular and complex detectors → more computing resources needed → more problems*
- ***Modern Machine Learning might be the way out***



(*)With nowadays software development

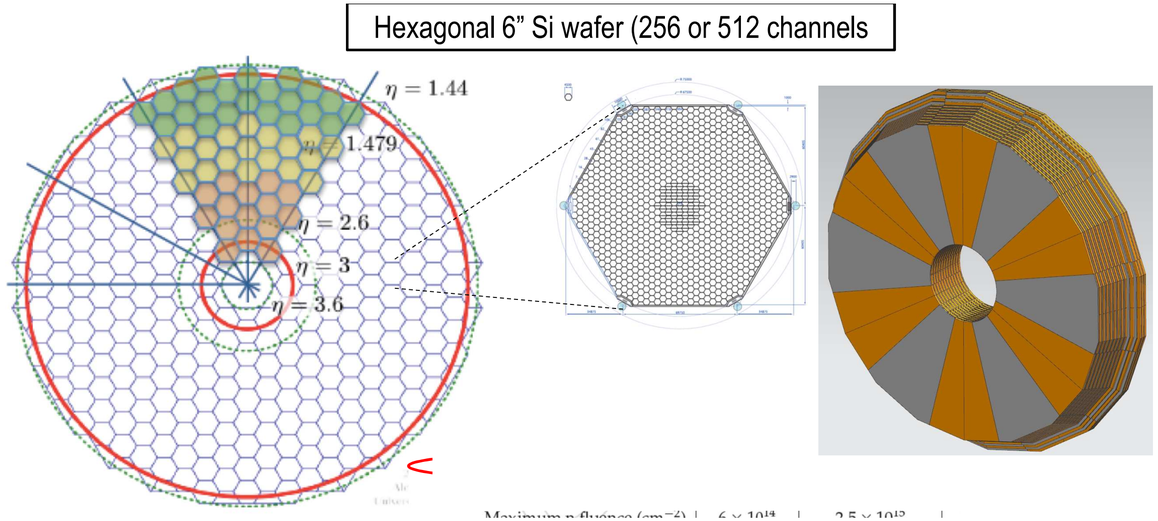
HL-LHC: elephant in the room



- ~40 collisions/event
- ~10 sec/event processing time
- (at best)Same computing resources as today

- ~200 collisions/event
- ~minute/event processing time(*)
- (at best)Same computing resources as today

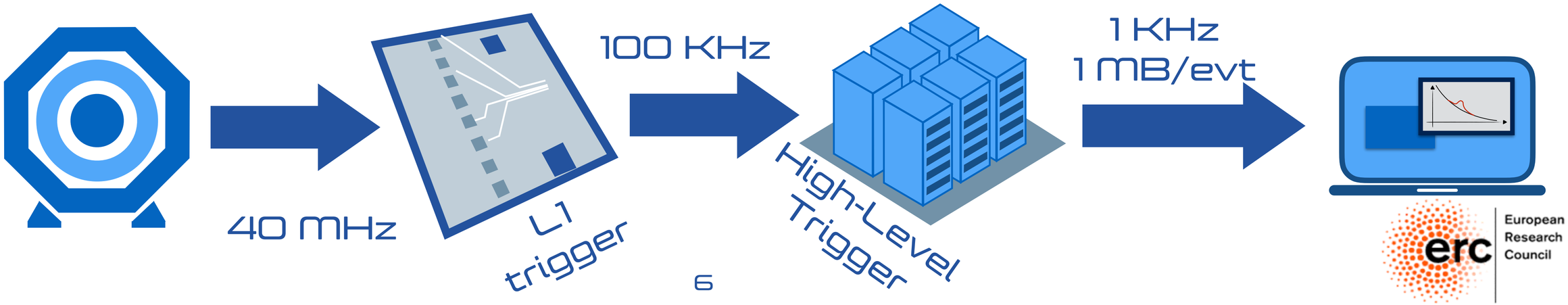
- *Flat budget vs. more needs =* ***current rule-based reconstruction algorithms will not be sustainable***
- ***Adopted solution:*** *more granular and complex detectors → more computing resources needed → more problems*
- ***Modern Machine Learning might be the way out***



(*)With nowadays software development

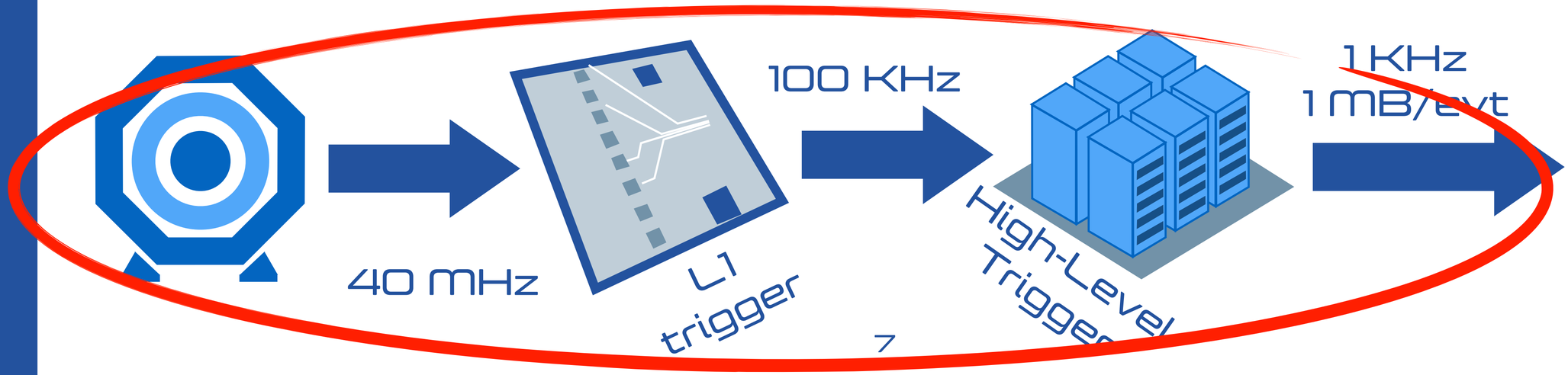
The LHC Big Data Problem

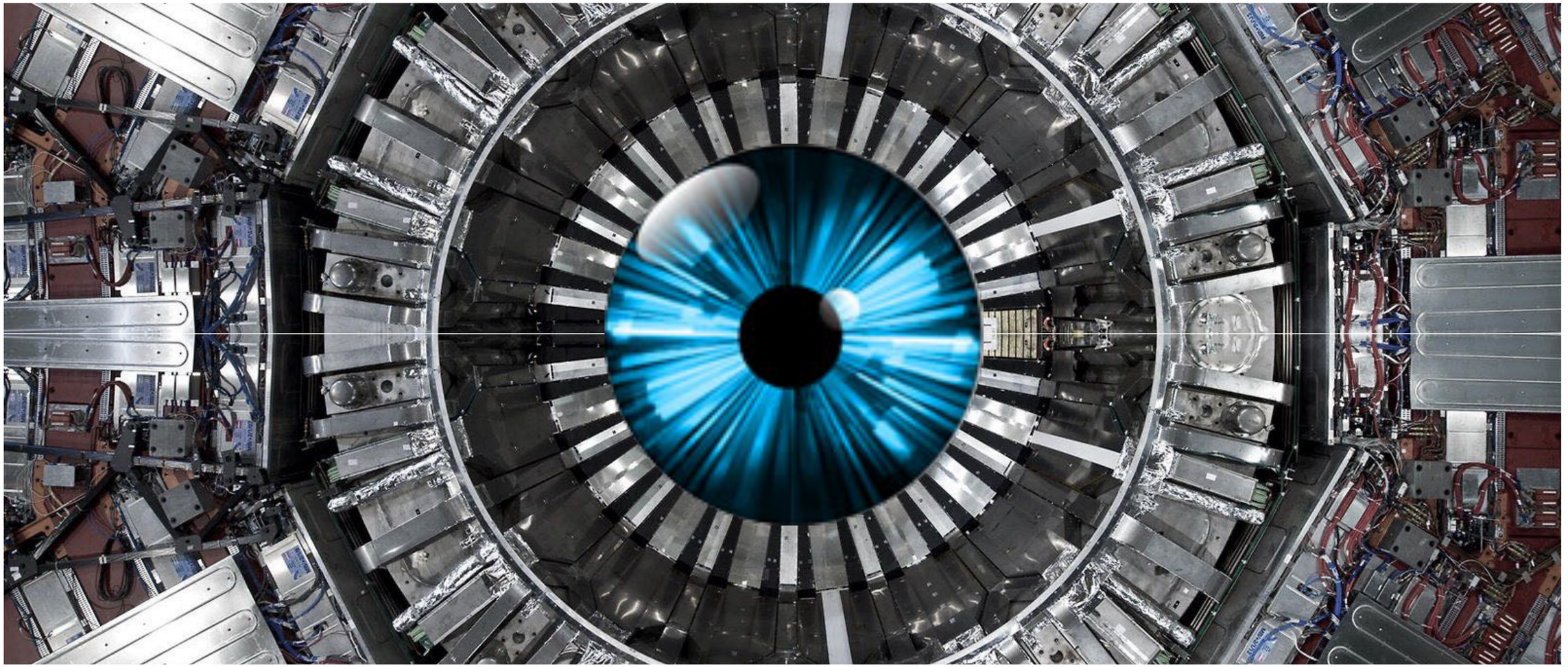
- *Too many data, too large data -> need to filter online*
- *Filters based on theoretical bias: **we might be loosing good events***
  - *L1 trigger: local, hardware based, on FPGA, @experiment site*
  - *HLT: local/global, software based, on CPU, @experiment site*
  - *Offline: global, software based, on CPU, @CERN T0*
  - *Analysis: user-specific applications running on the grid*



The LHC Big Data Problem

- *We are not seeing new physics: just re-doing what we do today with x10 more data WILL NOT be enough.*
- *The solution to the HL-LHC problem: modern Machine Learning to be faster and better in what we do today, freeing resources for new ideas*
- *This ML deployment need to happen in between collisions and data analysis (trigger, reconstruction, …), where freeing resources will make a difference*



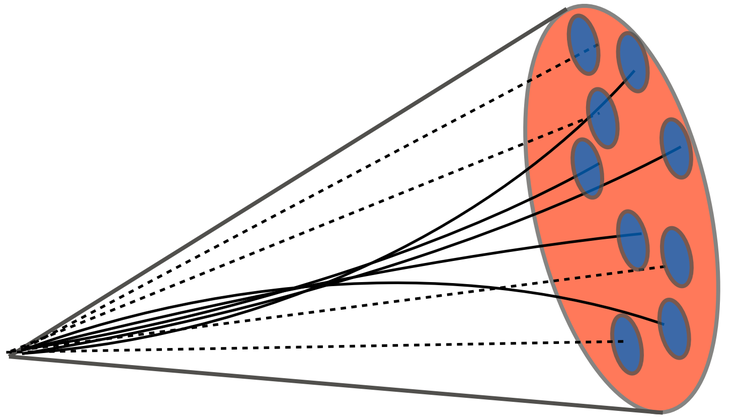


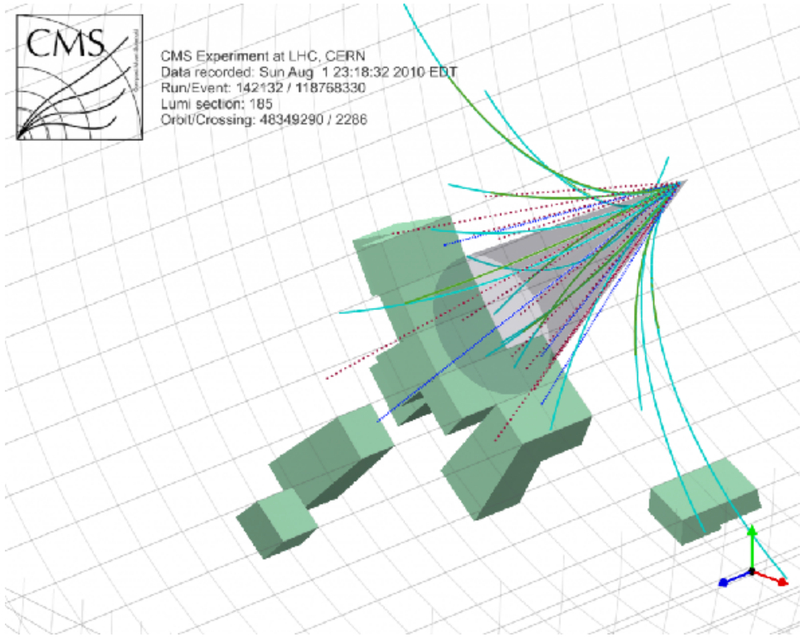
Faster Particle Reconstruction With Computer Vision

European Research Council

Faster/better Jet Reco

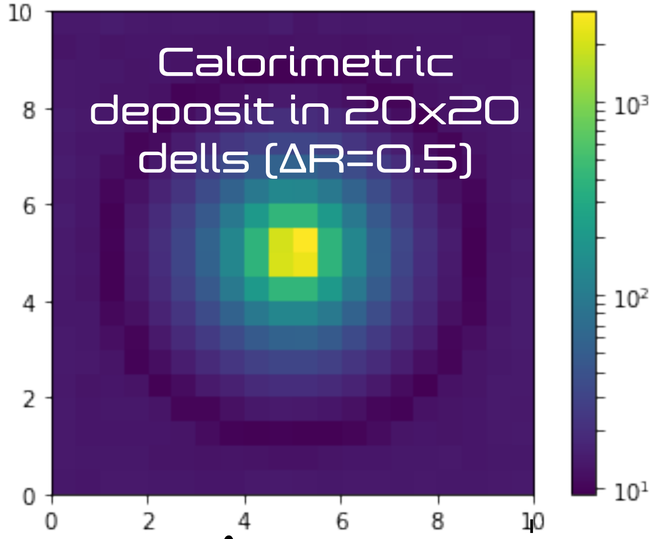
- *Jets are cone-like showers of quarks and gluons that produce tens of particles, all close to each other*
- *Jets can be reconstructed from calorimetric deposits, tracks, or full particles*
- *Depending on the quality of the ingredients, reconstruction can be more or less precise*
  - *coarse at trigger, when using fast calorimeter reconstructions*
  - *accurate offline, using all information*

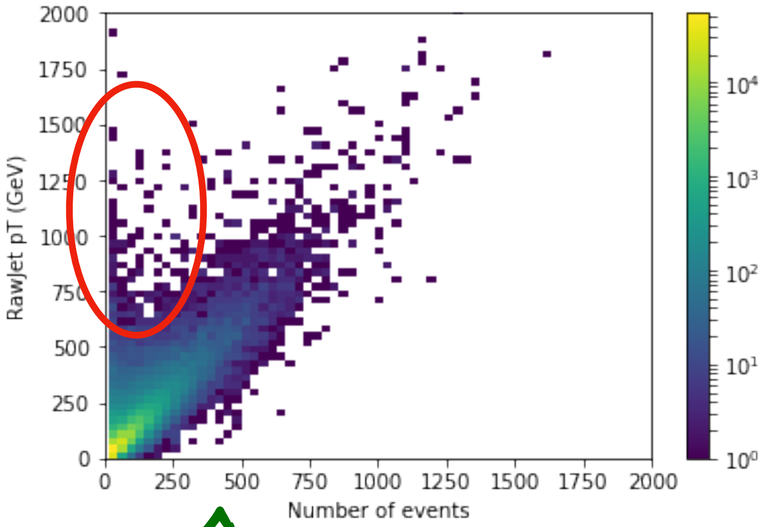


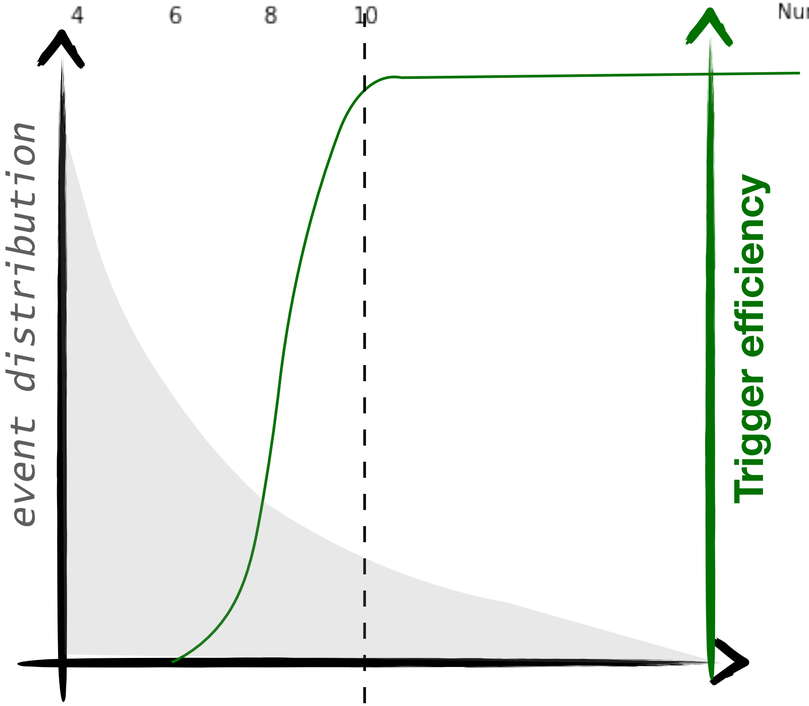


Faster/better Jet Reco

- *At L1/HLT, raw data objects are used*
  - *map of energy deposit on calorimeter*
  - *tracks from local/less accurate tracking*
- *Low-energy jets are promoted to large energy and not discarded*
- *So, triggers accept more than what should, or (at fixed budget) one is forced to apply tighter selection*

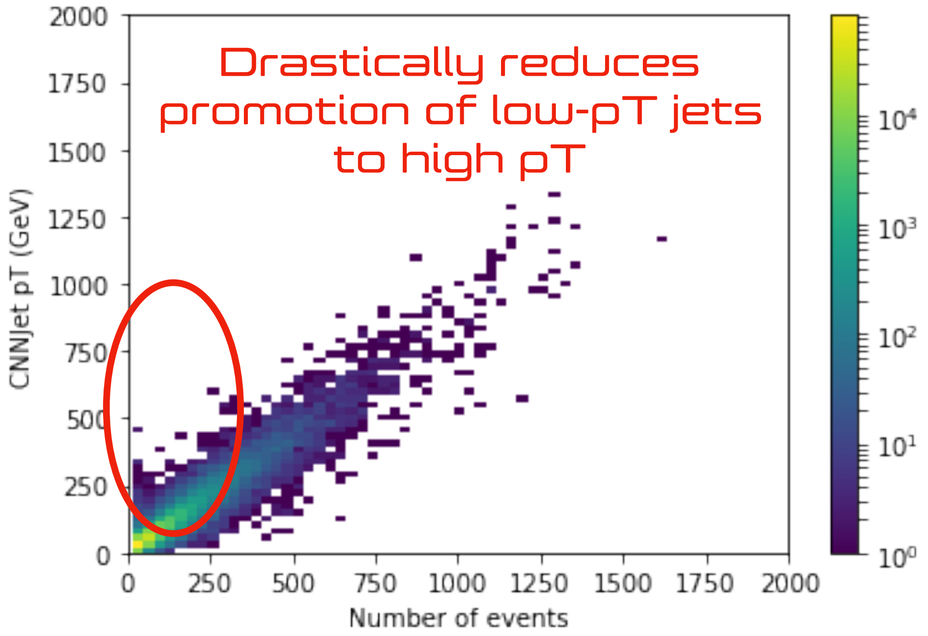


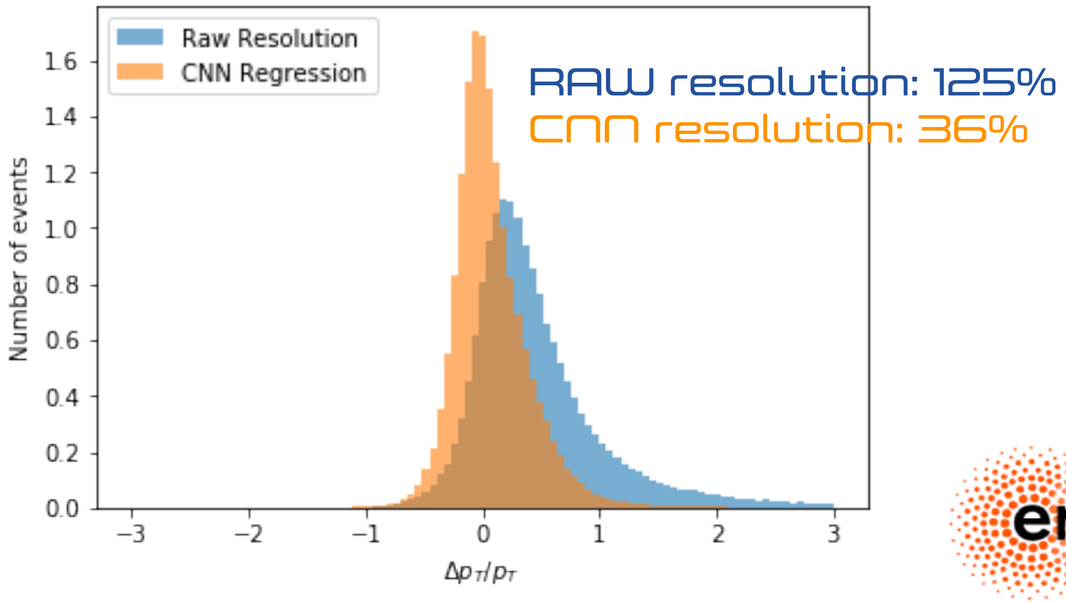




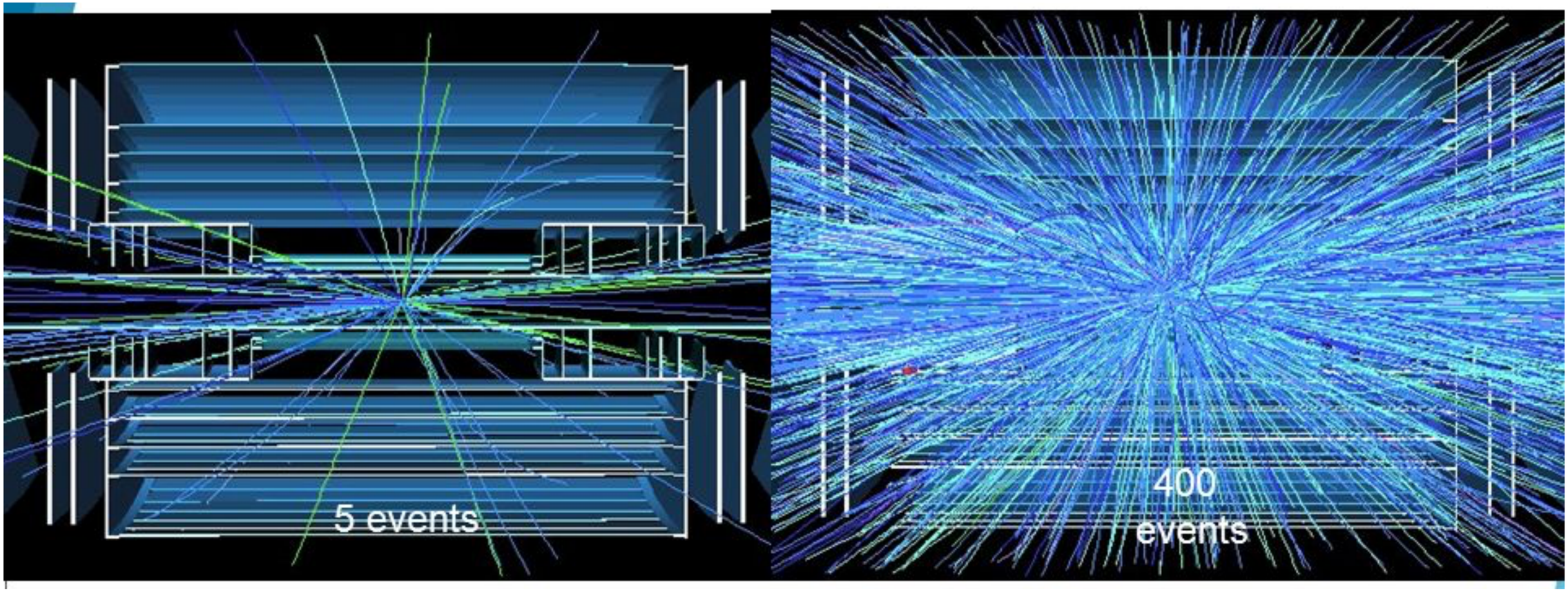
Faster/better Jet Reco

- *One could use ML to "guess" the offline jet energy from the online one + additional*
- *With a simple CNN2D network, substantial improvement observed*
- *Consequences downstream: better select which events to write /sec*
  - *We can do the same physics with less resources*
  - *We can do more physics with same resources*





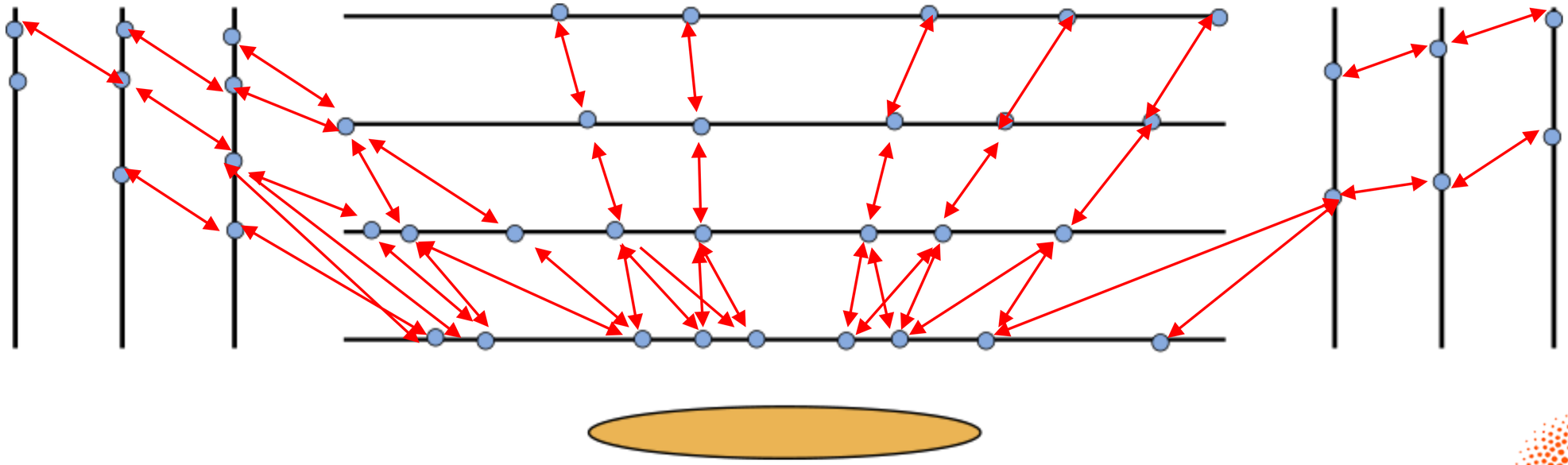
Combinatorics Reduction



- *Tracking is the most expensive workflow we have in RECO*
- *The more tracks we have, the more problematic it becomes (non-linearity due to combinatoric when connecting dots*

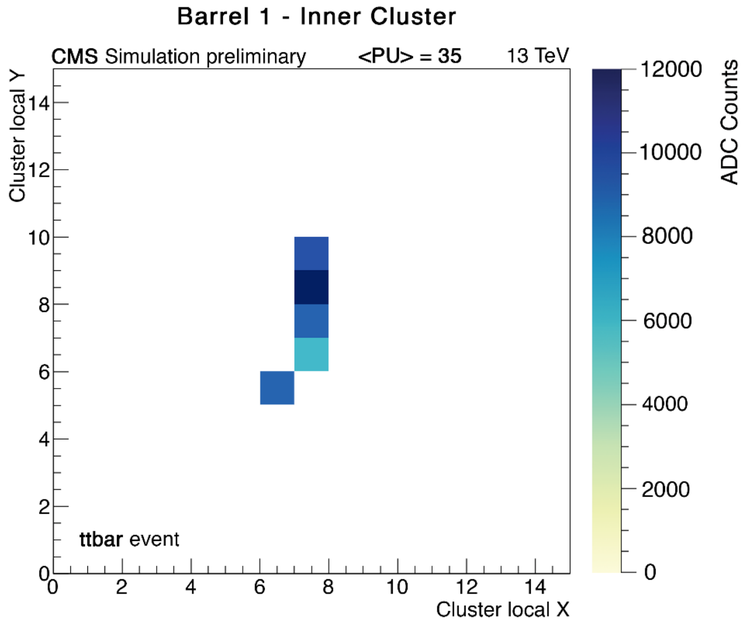
Combinatorics Reduction

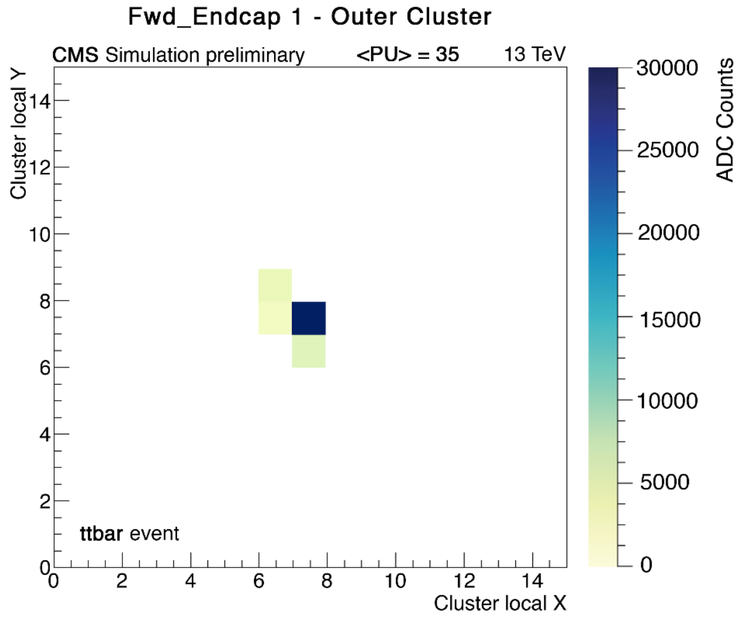
- *Track reco starts with a see in the inner layers (triplet of hits)*
- *Finding seeds is not the most CPU intensive aspect*
- *But its outcome determines the complexity of the following steps*
  - *one can speed up tracking by reducing the number of fake seeds*
- *We tried to solve this problem using a ConvNN*



Seeds as images

- *The detector sees the charge deposited by the crossing particle: a hit*
- *A hit is a window of sensors (16x16 here) with its deposited charge. This can be seen as a sparse digital image.*
- *Given two images, one can train a network to decide if a pair of hits is a good or bad match*

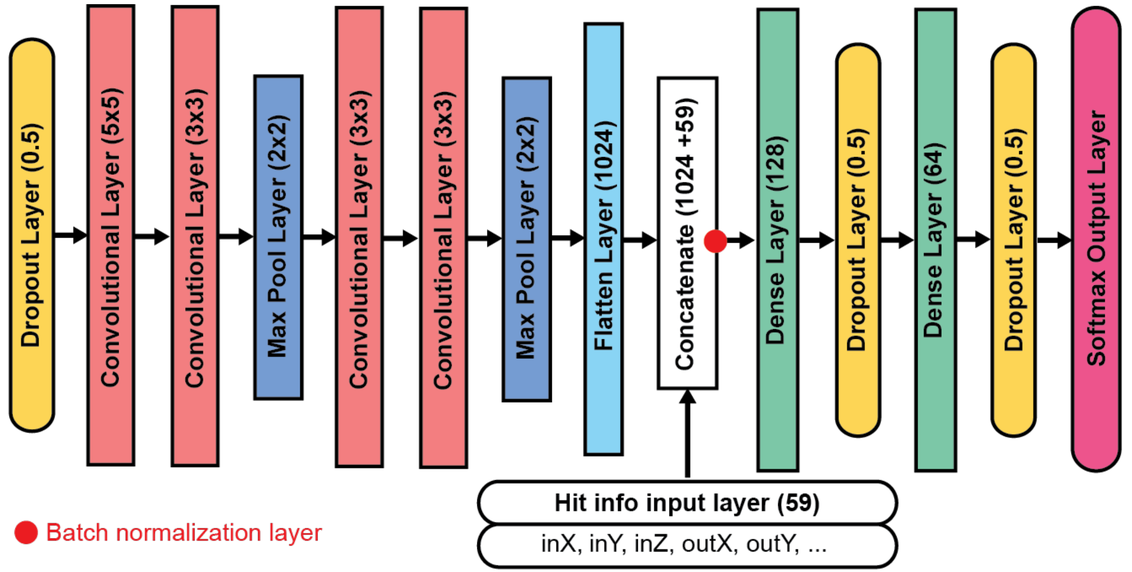




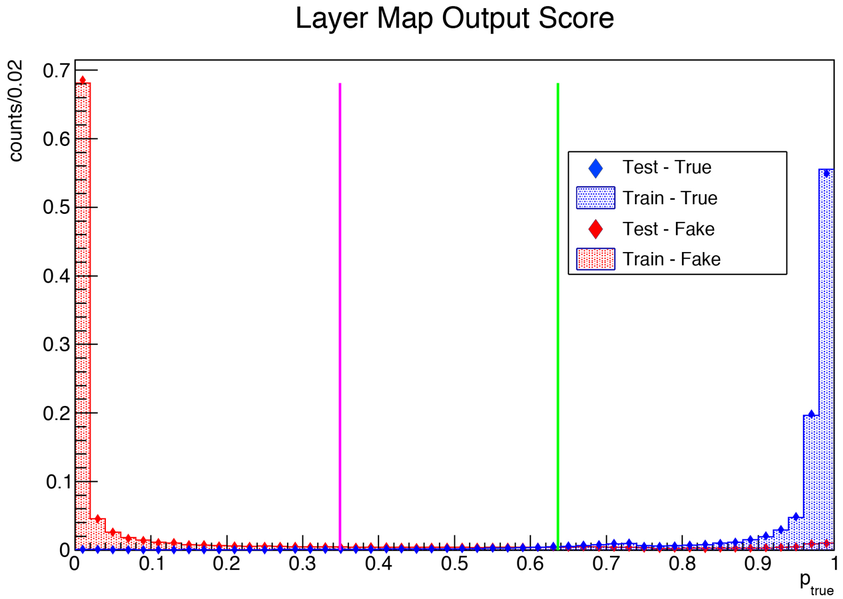
PixelSeed ConvNN

- *The final model uses two sets of inputs:*
  - *the hit images*

  - *a set of expert features (e.g., position of the hits in the detector) to help the learning process*



- *The trained model shows a good separation of true vs fake seeds*



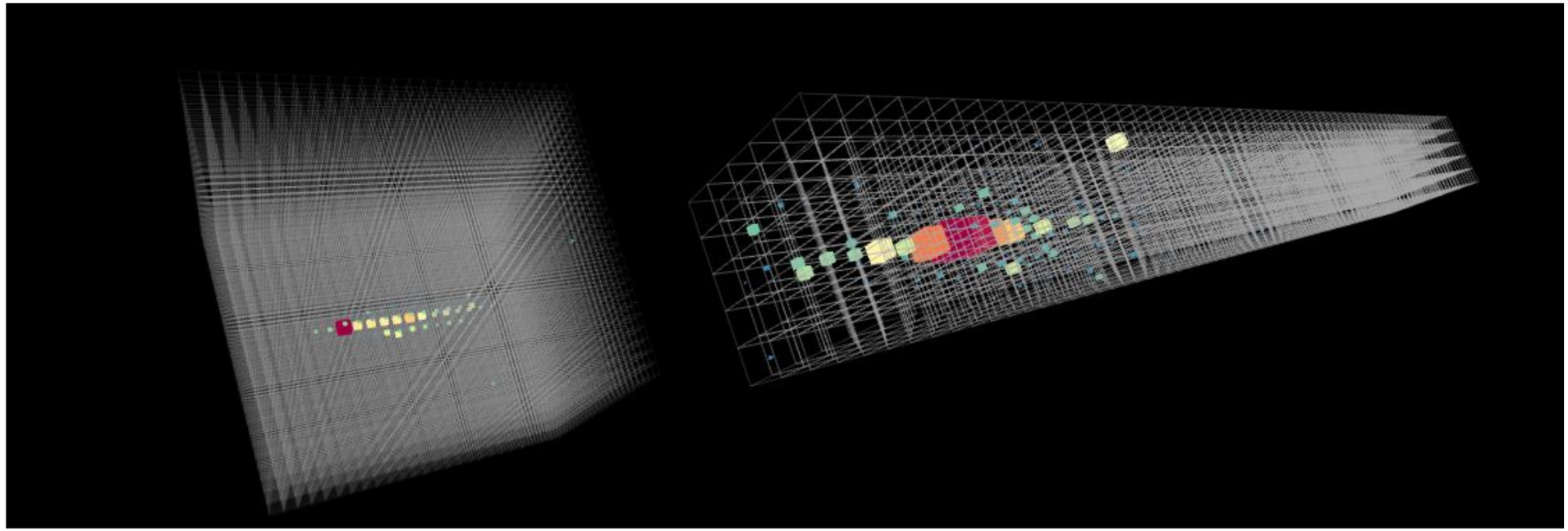
- ***One can reduce the fake rate by one order of magnitude with a few % loss in efficiency***

```
Efficiency (tpr) @ fake rejection
tpr @ rej 50%: 0.998996700259
tpr @ rej 75%: 0.990524391331
tpr @ rej 90%: 0.922210826719
tpr @ rej 99%: 0.338669401587
```


European Research Council

Calorimetry & Computer Vision

- *(next generation) digital calorimeters: 3D arrays of sensors with more regular geometry*
- *Ideal configuration to apply Convolutional Neural Network*
  - *speed up reconstruction at similar performances*
  - *and possibly improve performances*

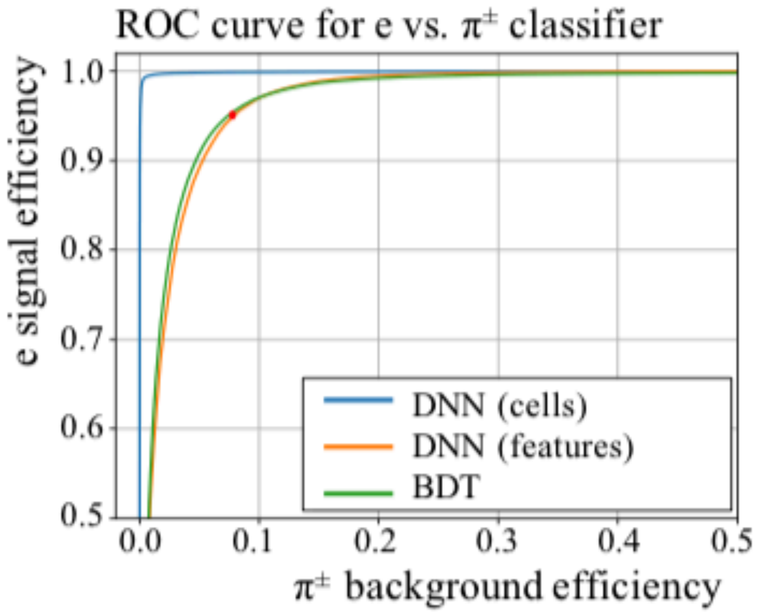


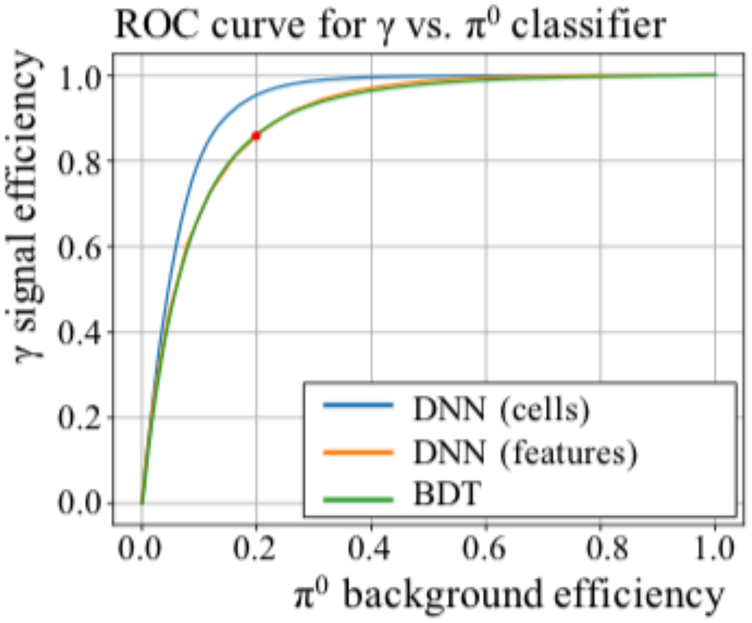
See contribution to NIPS workshop

European Research Council

Proof of Principle: Particle ID

- *We tried particle ID on a sample of simulated events*
  - *one particle/event (e, γ, π^0, π)*
- *Different event representations*
  - *high-level features related to event shape (moments of X,Y, and Z projections, etc)*
  - *raw data (energy recorded in each cell)*
- *Pre-filtered pion events to select the nasty ones and make the problem harder*

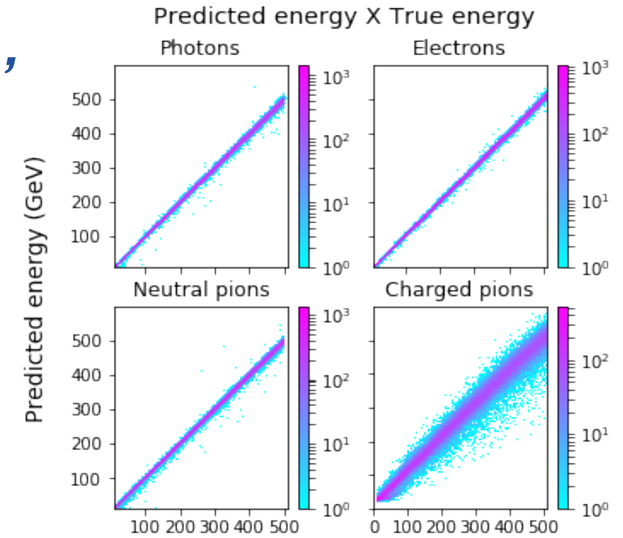


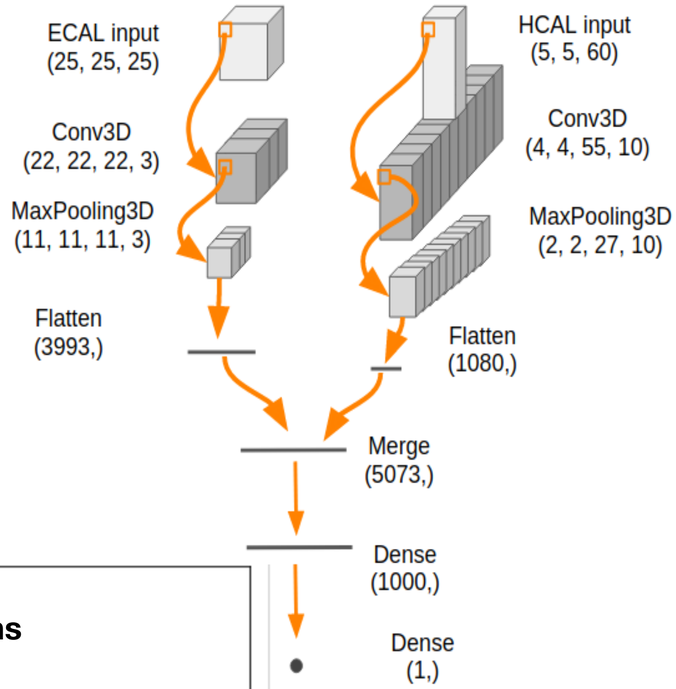


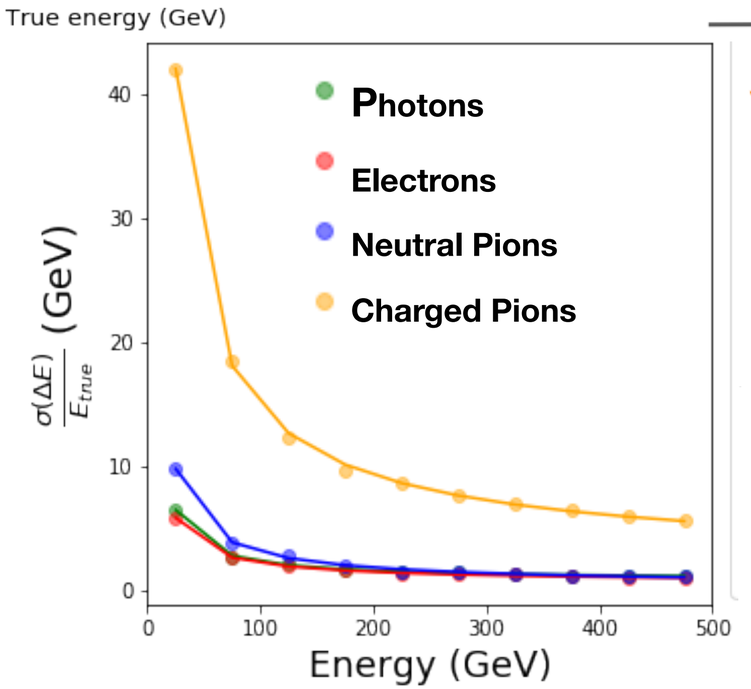
See contribution to NIPS workshop

Proof of Principle: Energy Regression

- *Correctly reconstruct energy, with physics meaningful performances*
  - *ECAL performances better than HCAL (as expected)*
  - *π^0 resolution ~ $\sqrt{2}$ γ resolution (as expected)*
- *FAST: used only RAW data as inputs -> no pre-processing*
  - *Processing time reduced by 10^3 wrt traditional approaches*
  - *Potentially usable both online and offline*



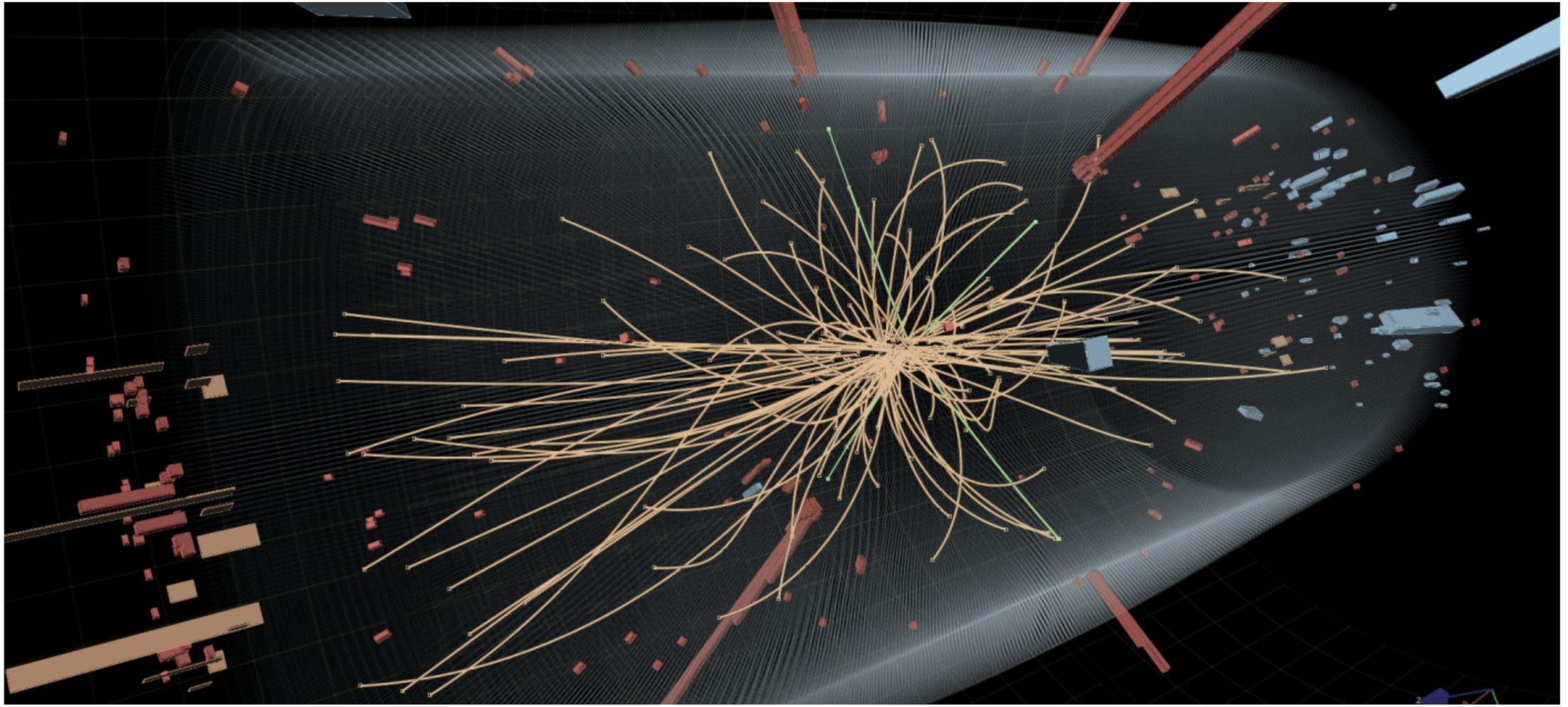




$\frac{\sigma(\Delta E)}{E_{true}}$ (GeV)

See contribution to NIPS workshop

European Research Council

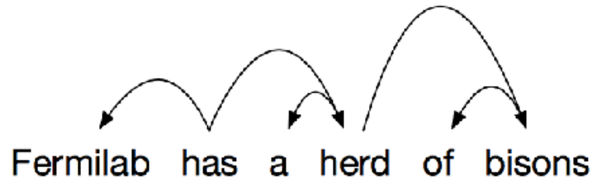


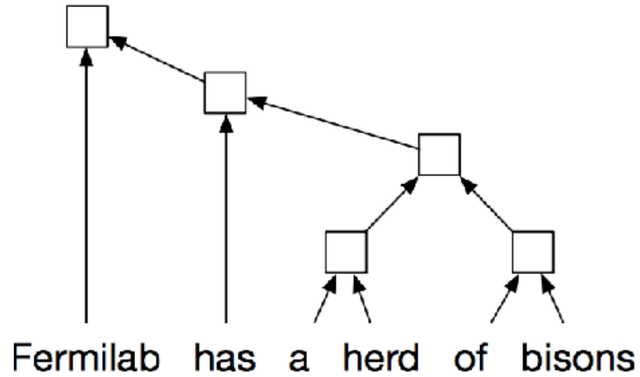
HEP & Language processing networks

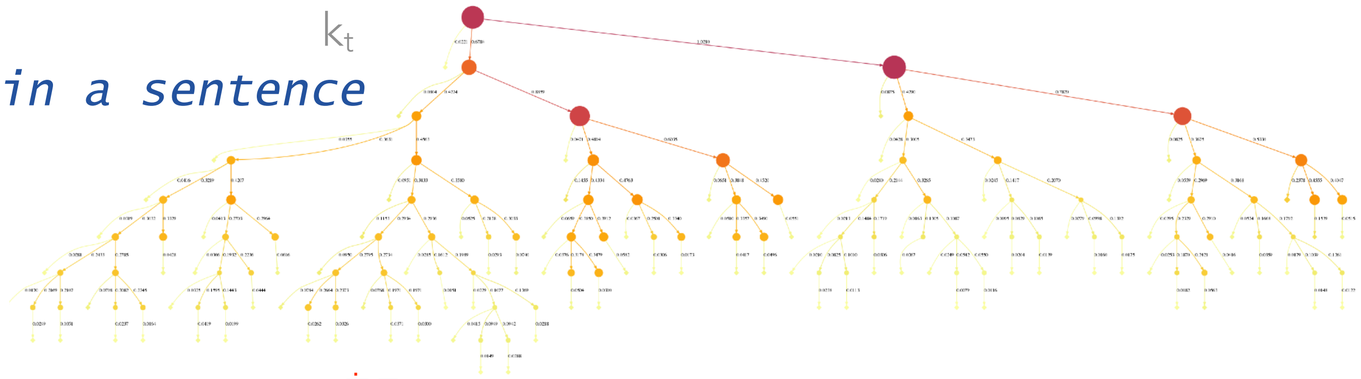
European Research Council

Particle Flow & language processing

- *CMS uses particle flow for event reconstruction:*
  - *At some point in the central processing, collision images are turned into a list of particles.*
  - *From these particles, complex objects (e.g., jets) are formed*
- *In this framework, Computing vision approaches are not necessarily ideal*
- *One can instead use language-processing approaches (e.g., recurrent neural networks*
  - *particles are words in a sentence*
  - *QCD is the grammar*

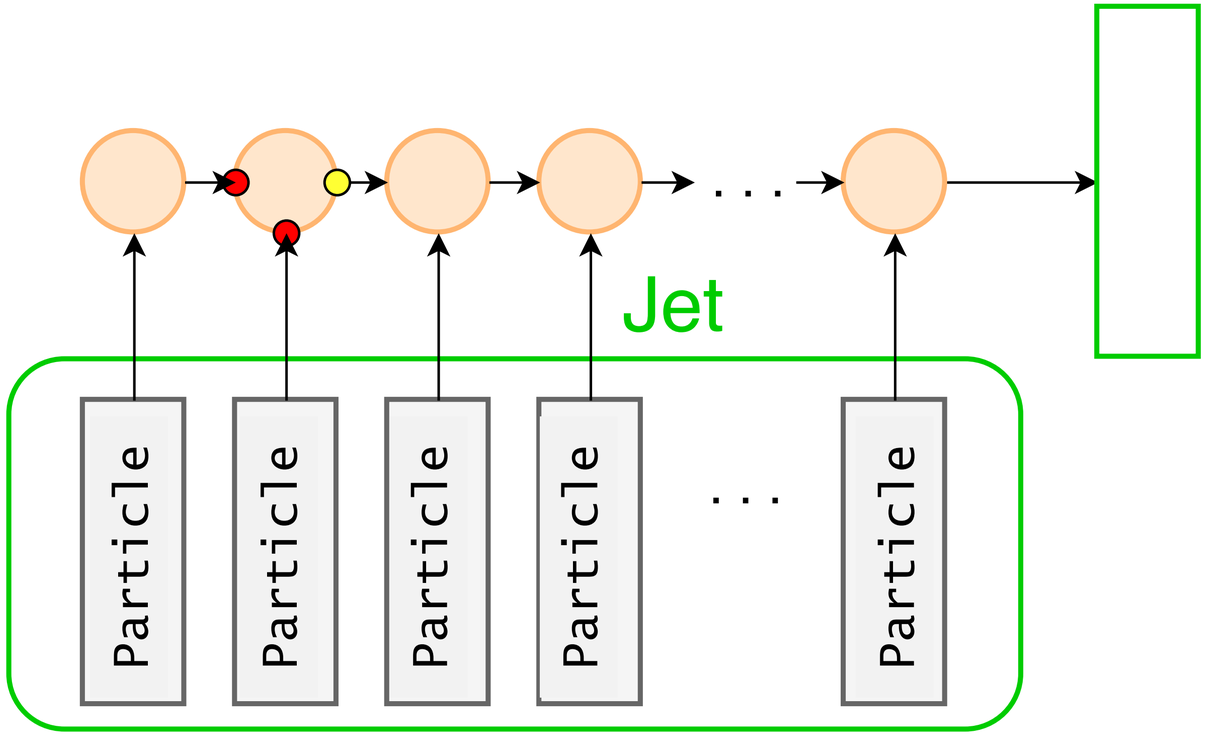






Recurrent Neural Networks

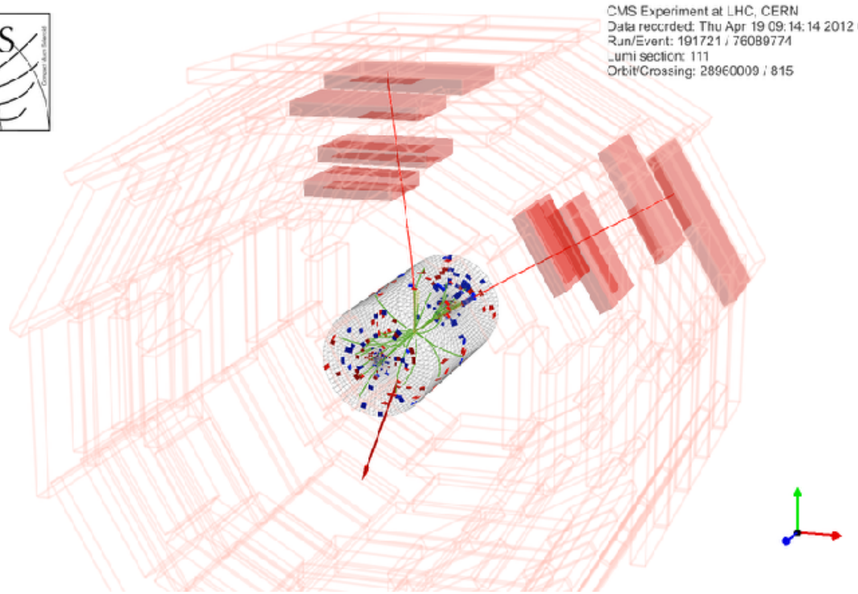
- A network architecture suitable to process an ordered sequence of inputs
  - words in text processing
  - a time series
  - particles in a list
- Could be used for a single jet or the full event
- Next step: graph networks (active research direction)

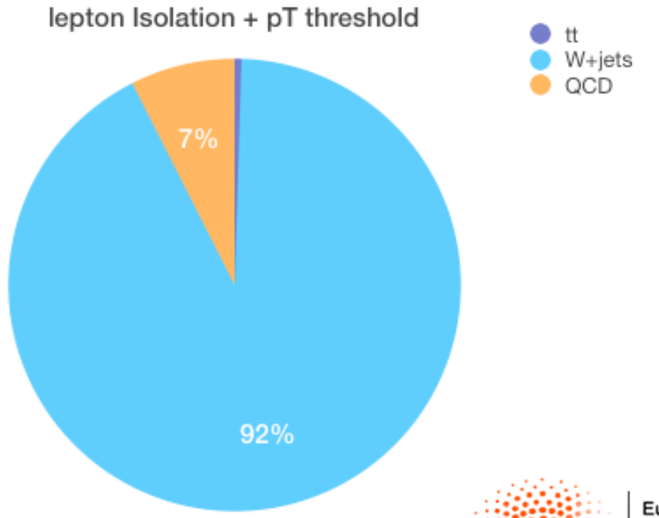


A Topology Classifier

A typical example: leptonic triggers

- *at the LHC, producing an isolated electron or muon is very rare. Typical smoking gun that something interesting happened (Z,W,top,H production)-> **TAKE THEM!***
- *Triggers like those are very central to ATLAS/CMS physics*
- ***The sample selected is enriched in interesting events, but still contaminated by non-interesting ones***
- *Can we clean this up w/o biasing the physics? yes, with ML*

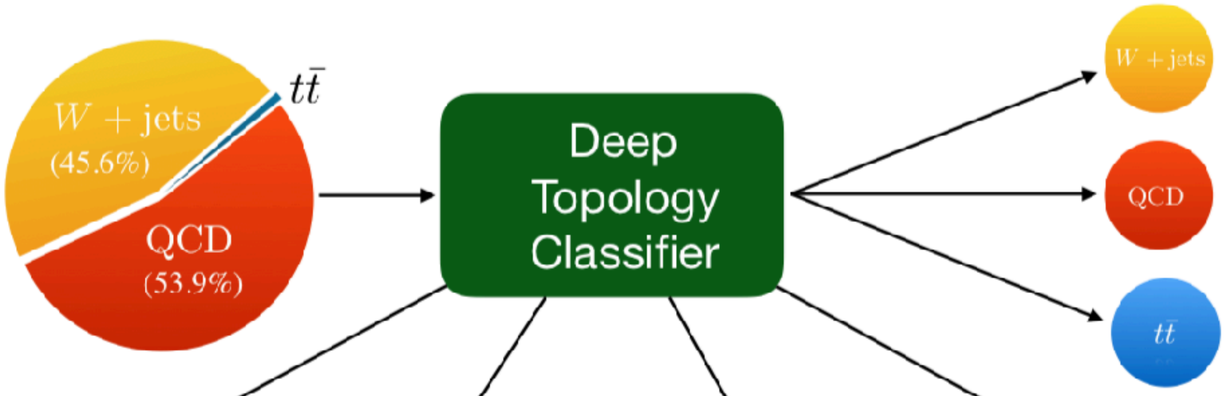




See contribution to NIPS workshop

European Research Council

A Topology Classifier



Calo Image Classifier

Raw images of the calorimetry hits fed to a convolutional NN.

Particle Sequence Classifier

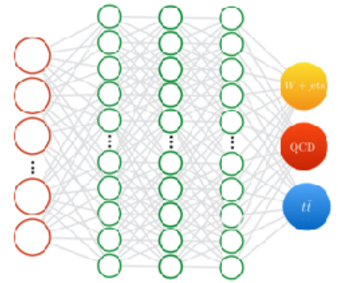
A sequence of particles taken as input to a recurrent NN.

Abstract Image Classifier

Based on an abstract representation of the reconstructed particles as an image to feed to a convolutional NN.

High-level Feature Classifier

Use high-level features as inputs to a fully connected NN.



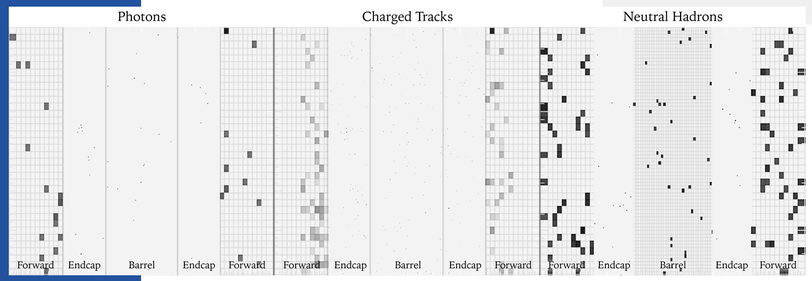
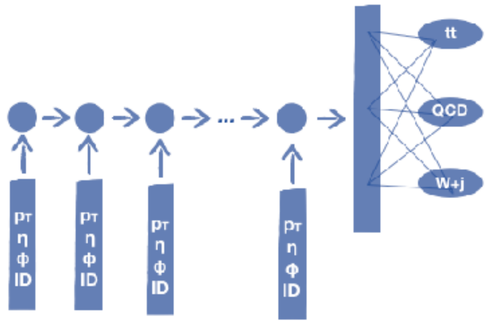
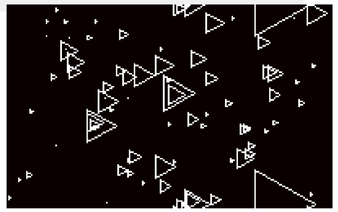
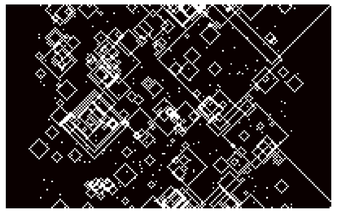


Figure 2: An example of a $t\bar{t}$ event as the input of the raw-image classifier.

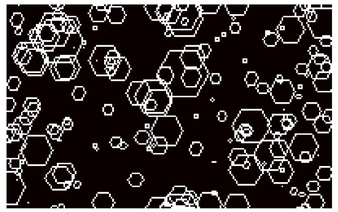




(a) Photons

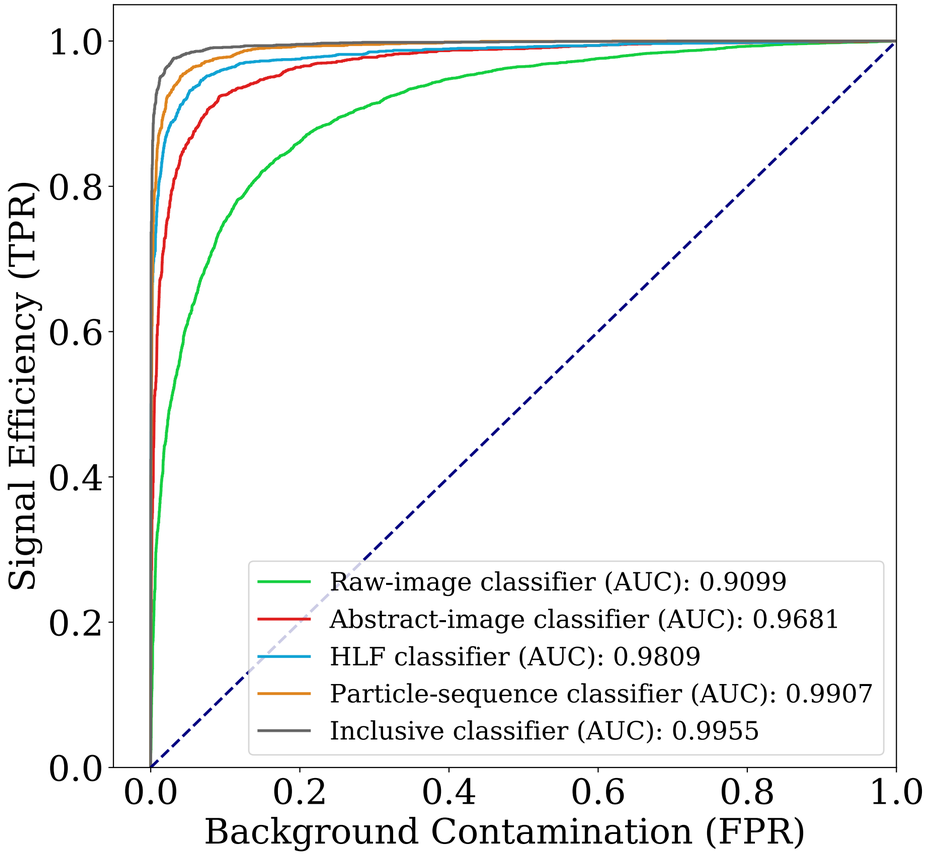


(b) Charged Particles

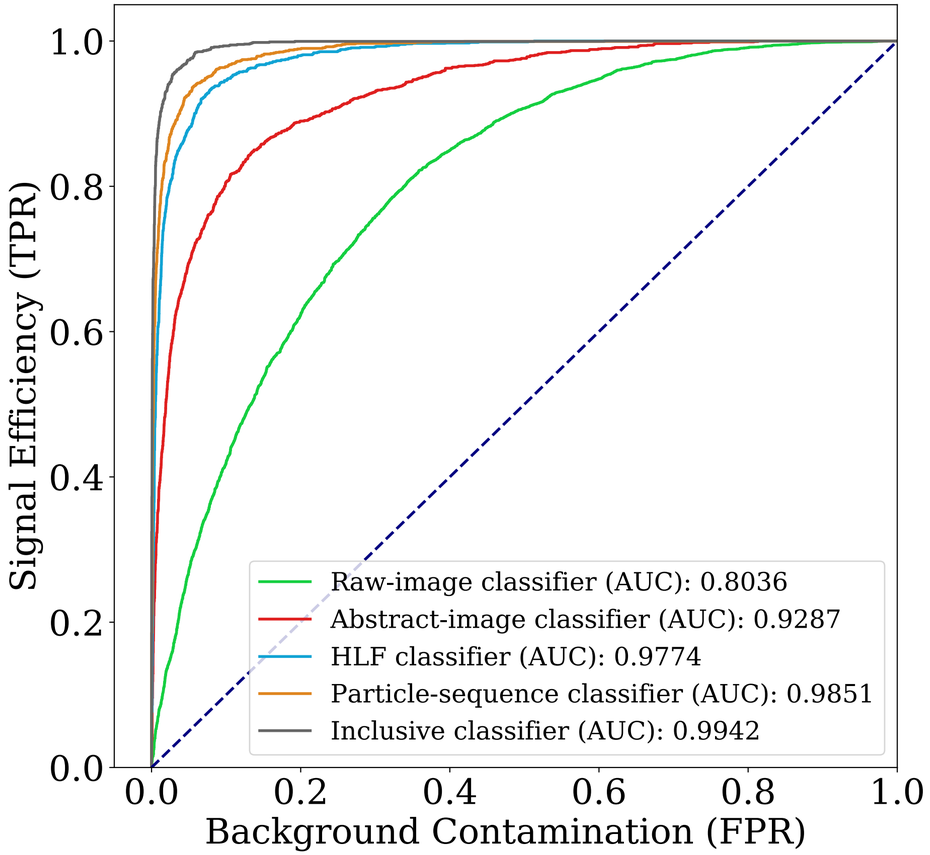


(c) Neutral Hadrons

Selection performances



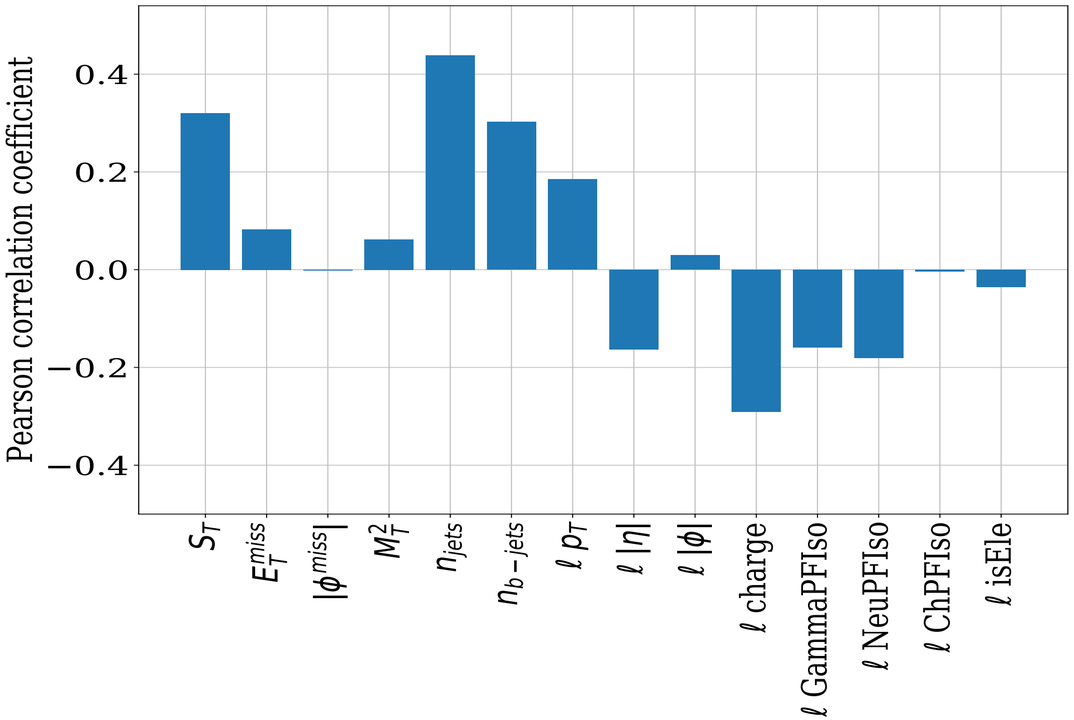
(a) $t\bar{t}$ selector

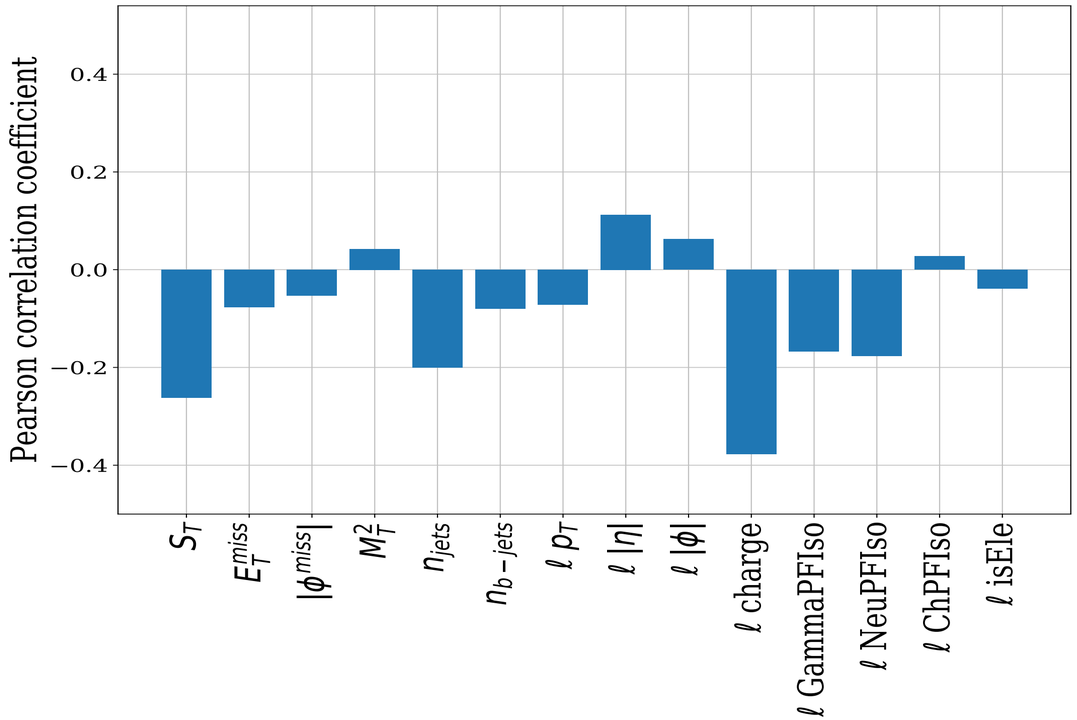


(b) W selector

Can select 99% of the top events and reduce the fraction of written events by a factor ~ 7

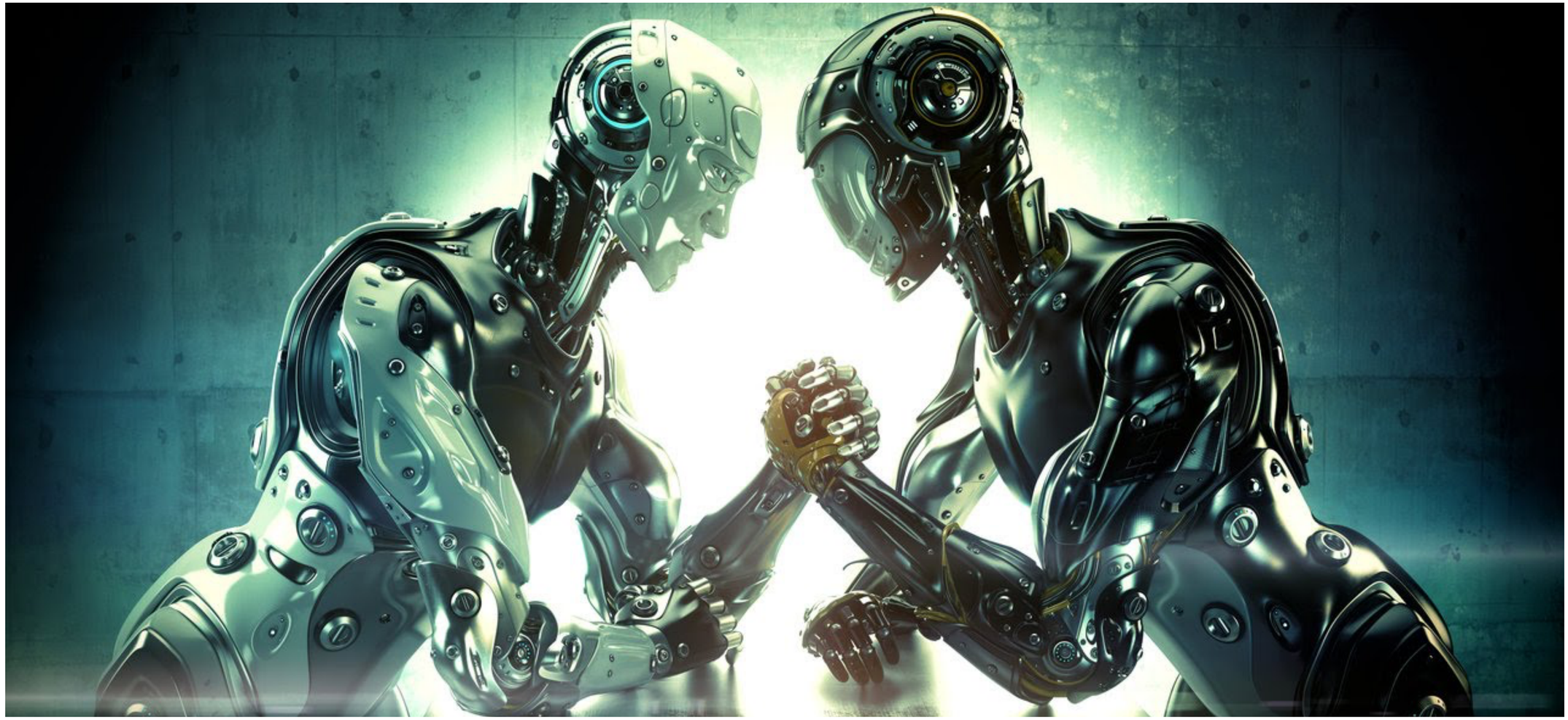
Selection performances





What is the network learning?

- tt events are more crowded that W events
- leptons in W and tt events are isolated from other particles

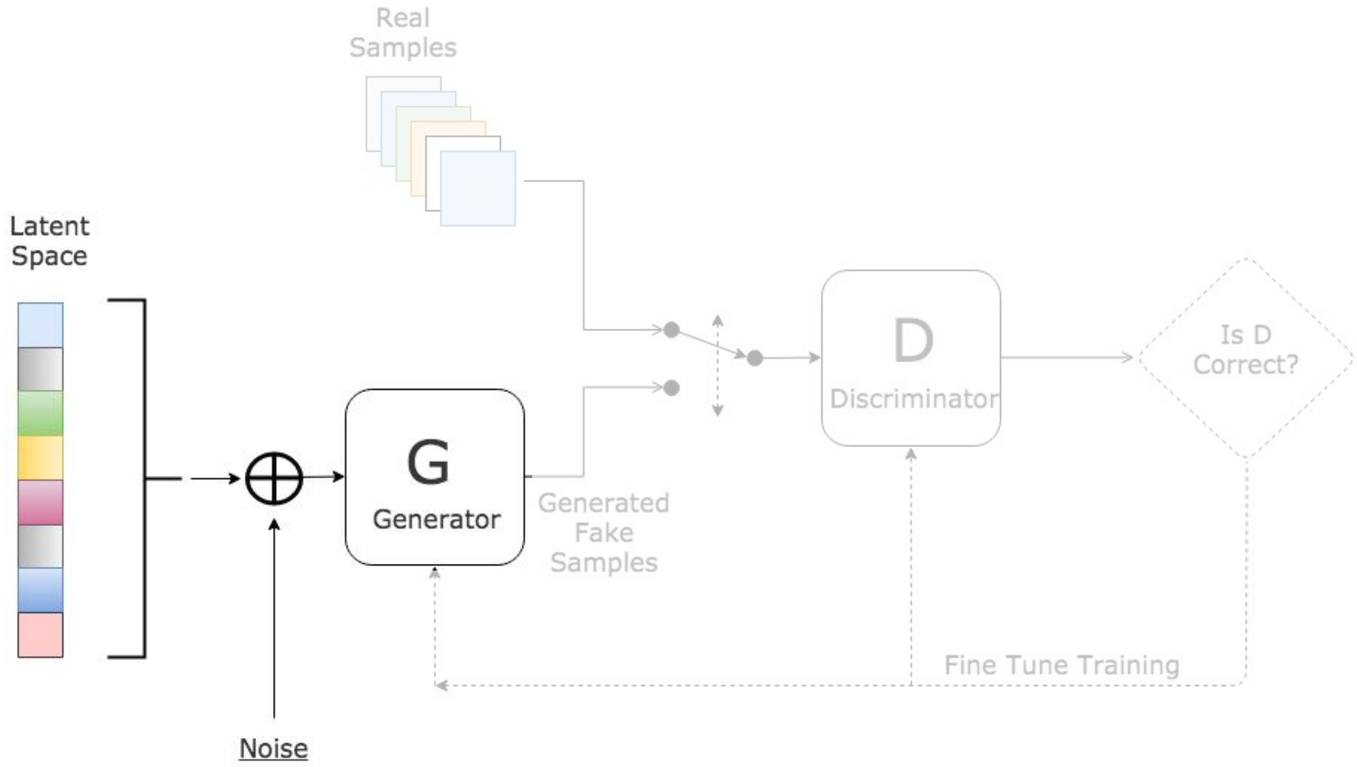


Generative Adversarial Networks

European Research Council

Generative Adversarial Training

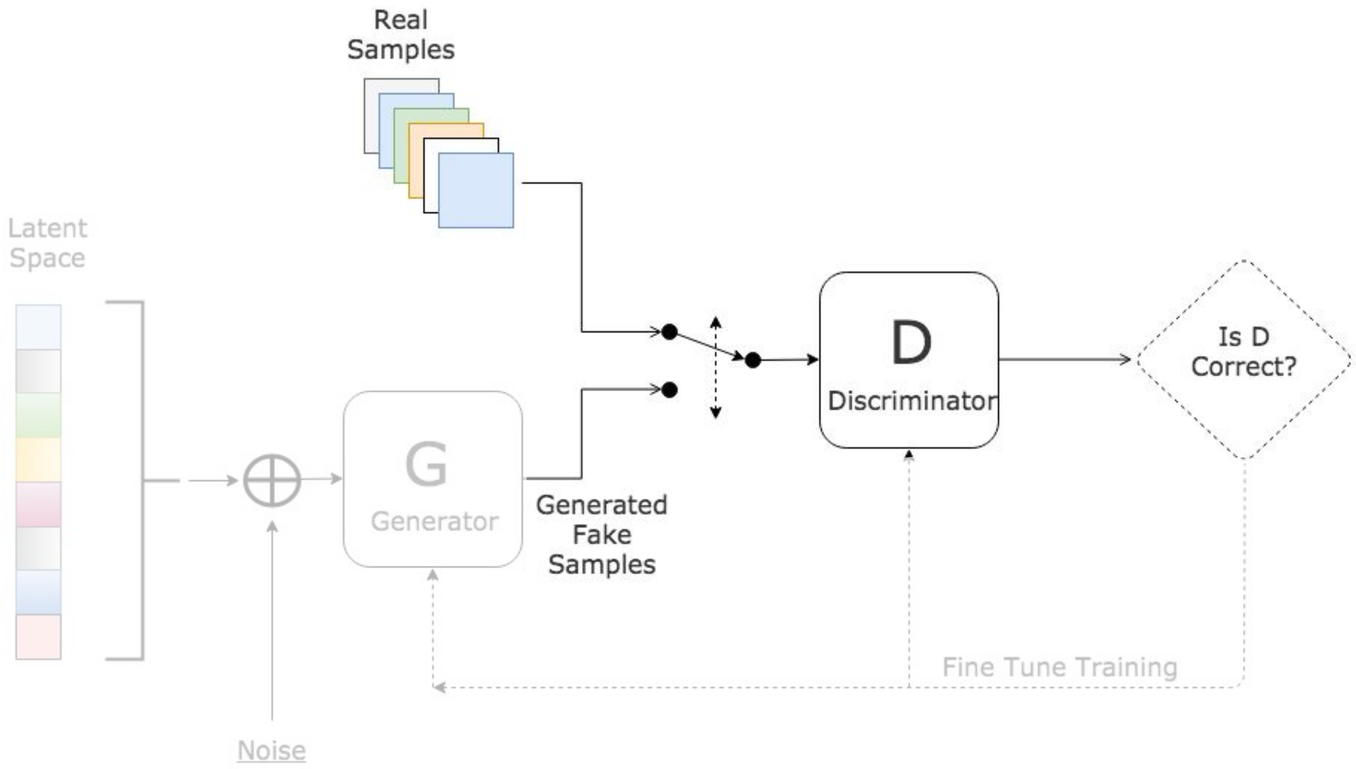
- *Two networks trained in competition*
  - ***Generator: creates images starting from random noise*** *(and optionally some other information to transform)*
  - *Discriminator: tries to distinguish true from generator-created images*



- *The loss function to minimise is written as Loss(Gen)-Loss(Disc)*
  - *Goes up is discriminator improves*
  - *Goes down if the generator improves*
- *The generator learns to make images like a given set it never sees, simply training itself to fool the generator*

Generative Adversarial Training

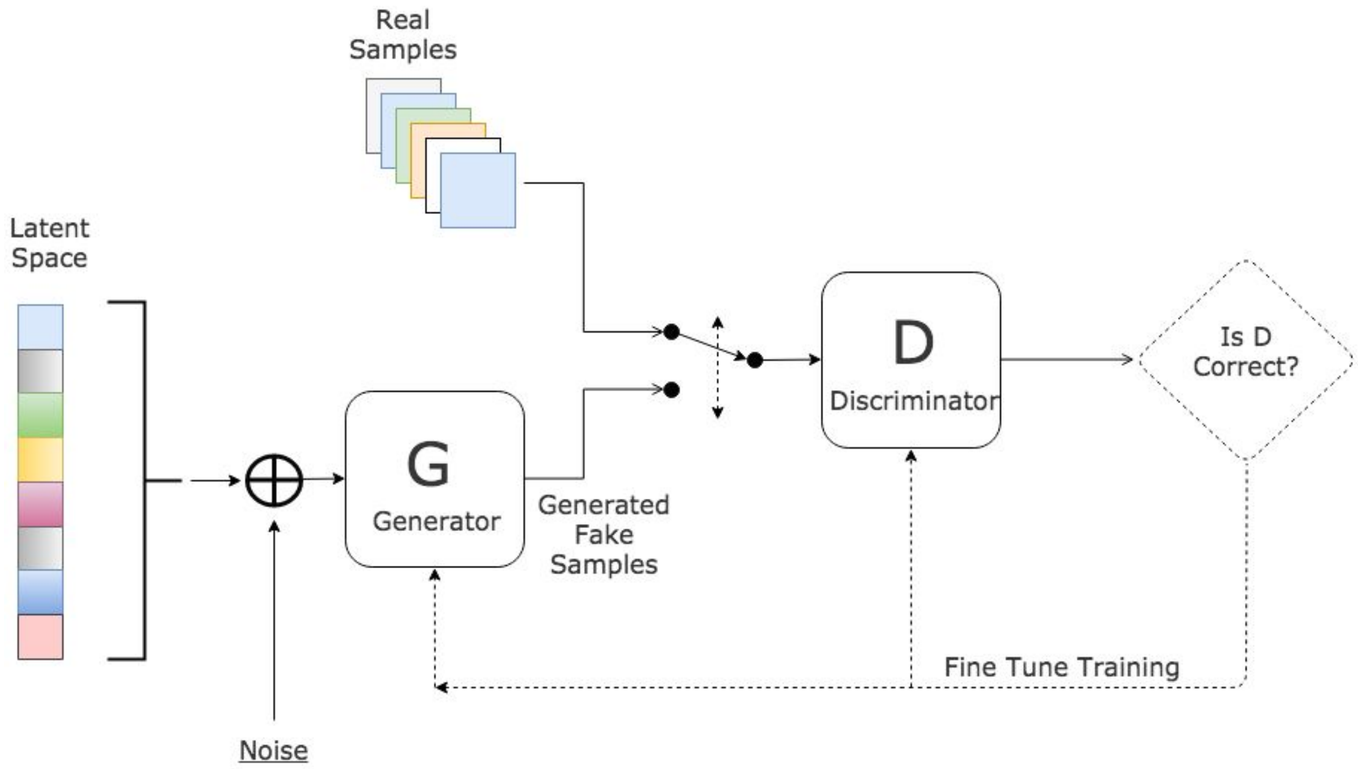
- *Two networks trained in competition*
  - *Generator: creates images starting from random noise (and optionally some other information to transform)*
  - ***Discriminator: tries to distinguish true from generator-created images***



- *The loss function to minimise is written as Loss(Gen)-Loss(Disc)*
  - *Goes up is discriminator improves*
  - *Goes down if the generator improves*
- *The generator learns to make images like a given set it never sees, simply training itself to fool the generator*

Generative Adversarial Training

- *Two networks trained in competition*
  - *Generator: creates images starting from random noise (and optionally some other information to transform)*
  - *Discriminator: tries to distinguish true from generator-created images*



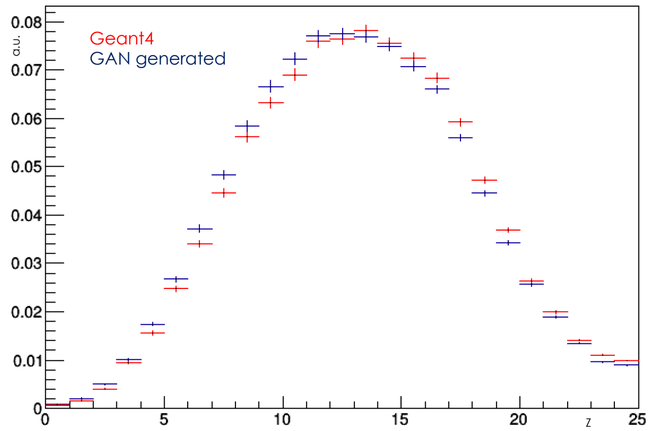
- ***The loss function to minimise is written as Loss(Gen)-Loss(Disc)***
  - ***Goes up is discriminator improves***
  - ***Goes down if the generator improves***
- ***The generator learns to make images like a given set it never sees, simply training itself to fool the generator***

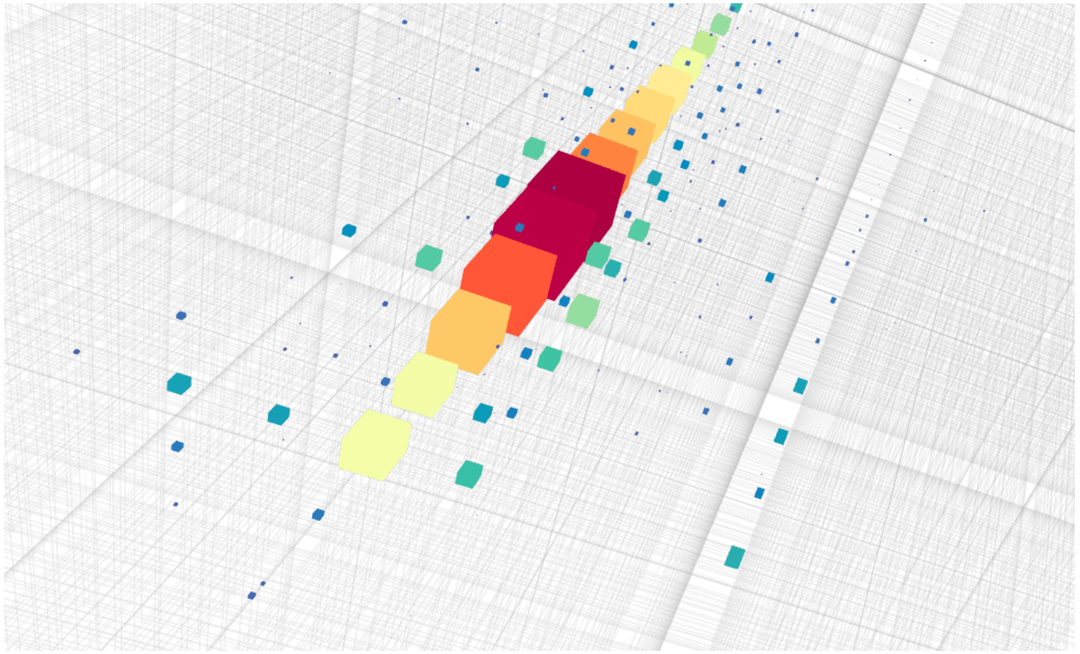
Adversarial training in azione

Image generation

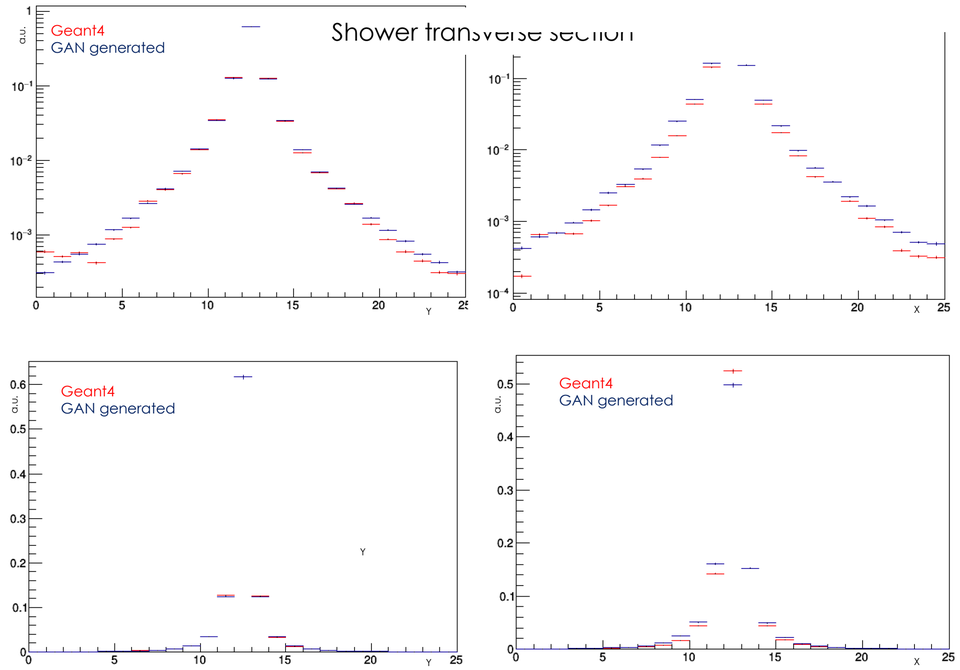
- Start from random noise
- Works very well with images
  - Applied to electron showers in digital calorimeters as a replacement of GEANT

Shower longitudinal section





Shower transverse section

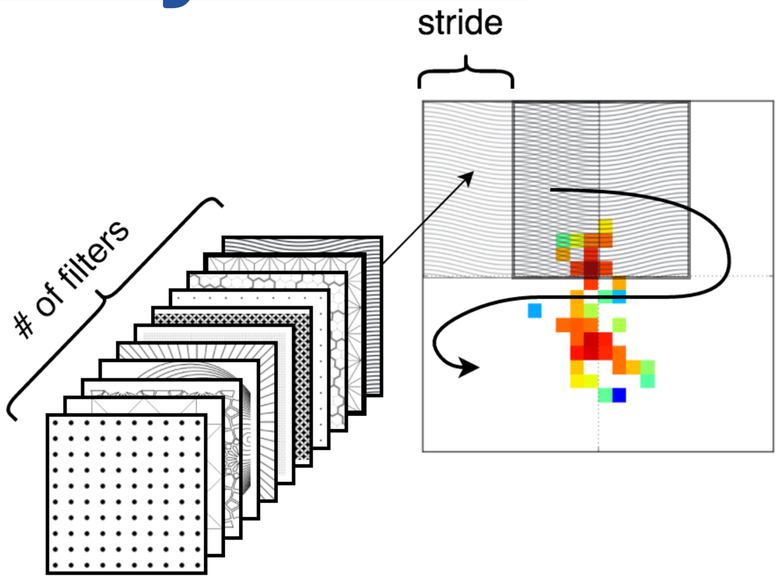


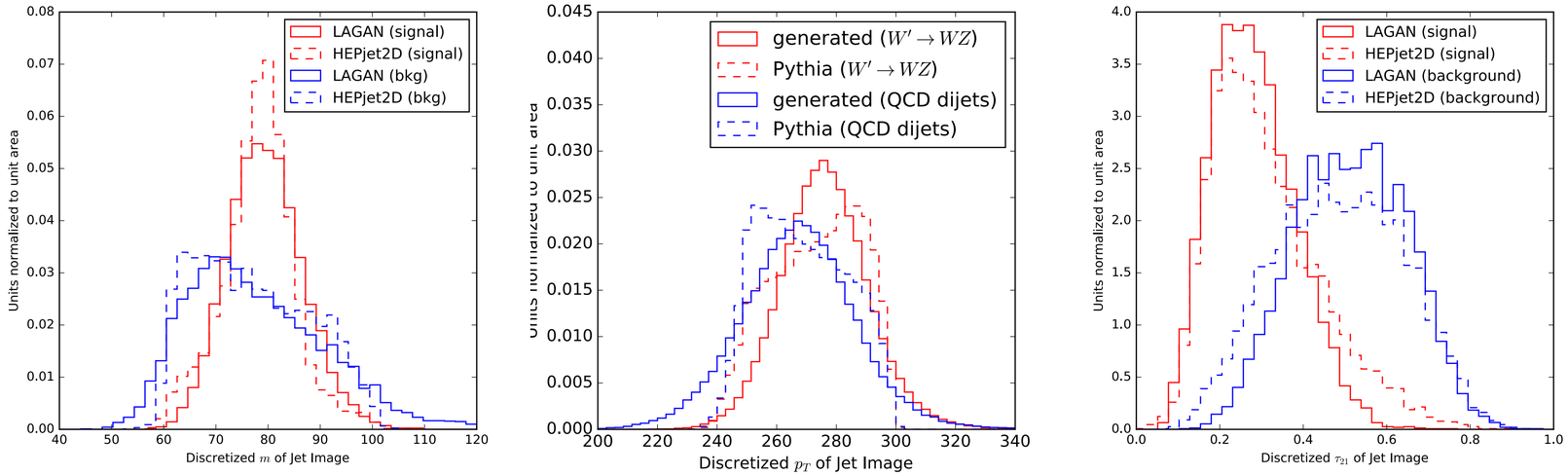
See contribution to NIPS workshop

European Research Council

Generating full jets

- Start from random noise
- Works very well with images
  - Applied to electron showers in digital calorimeters as a replacement of GEANT



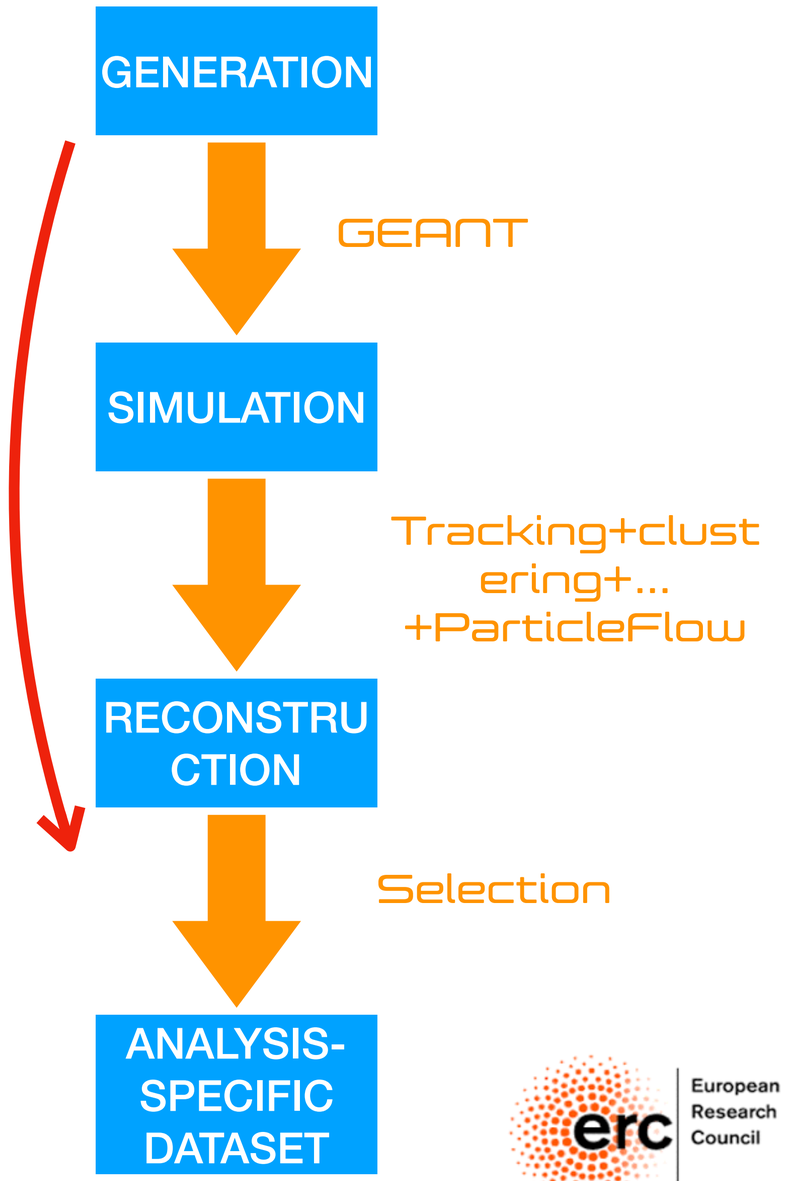


de Olivera, Paganini, and Nachman
https://arxiv.org/pdf/1701.05927.pdf

Figure 6: The distributions of image mass $m(I)$, transverse momentum $p_{\mathrm{T}}(I)$, and n-subjettiness $\tau_{21}(I)$. See the text for definitions.

Simulation is half of the problem

- *Reconstruction involves more than one detector (e.g., tracker + calorimeter) and produces (at least in CMS) a list of particles*
- *GANs were proved to be useful to emulate the SIM+RECO step in one goal*
  - *jets out of full particle reconstruction emulated in a GAN*
  - *trained on actual SIM+RECO synthetic data by CMS*



Simulation is half of the problem

- *Reconstruction involves more than one detector (e.g., tracker + calorimeter) and produces (at least in CMS) a list of particles*
- *GANs were proved to be useful to emulate the SIM+RECO step in one goal*
  - *jets out of full particle reconstruction emulated in a GAN*
  - *trained on actual SIM+RECO synthetic data by CMS*

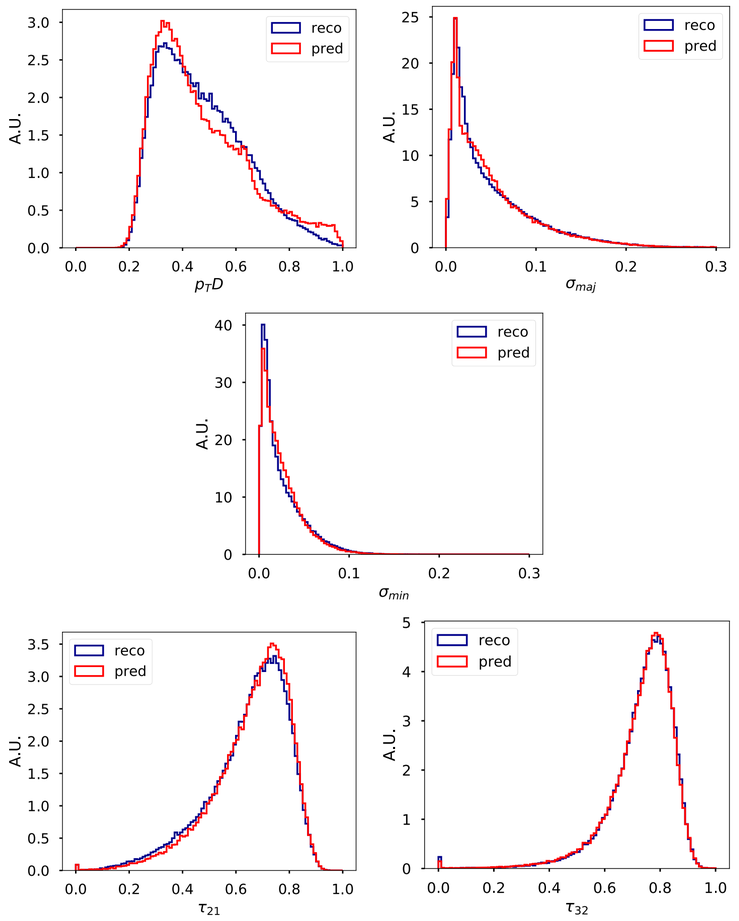
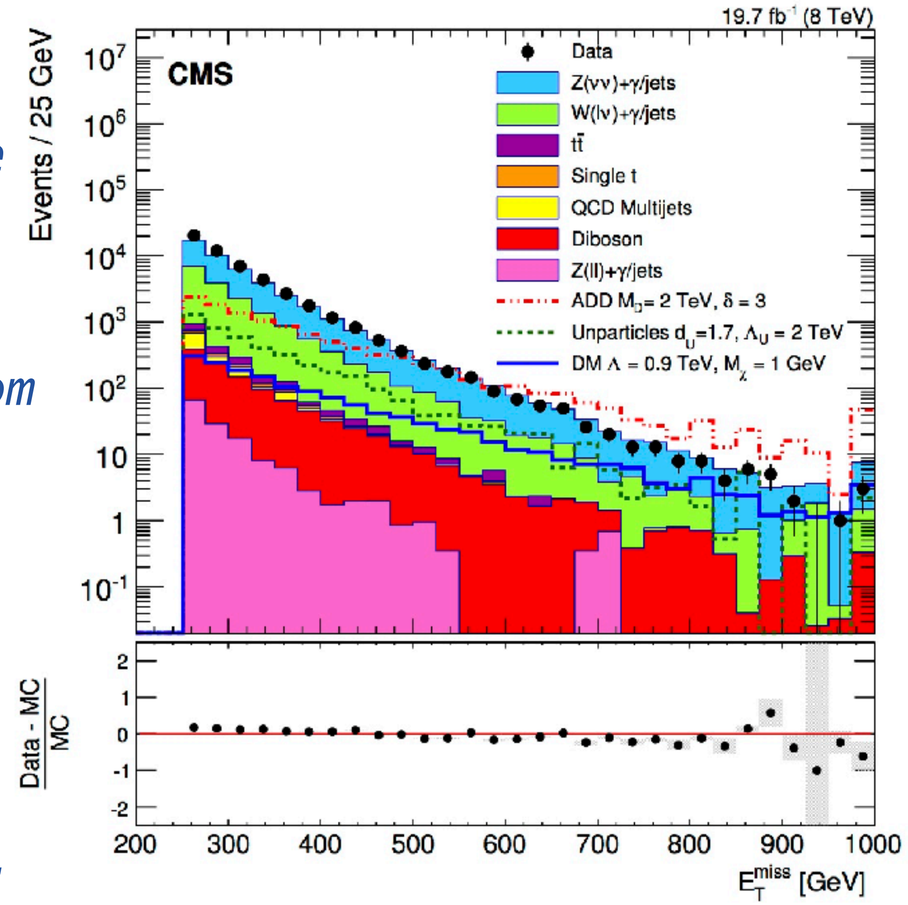


Fig. 6 Distribution of high level variables used for quark/gluon discrimination (first two rows) and merged jets tagging (last row). Blue histograms are obtained from the input data, while red ones are obtained using the generative model.

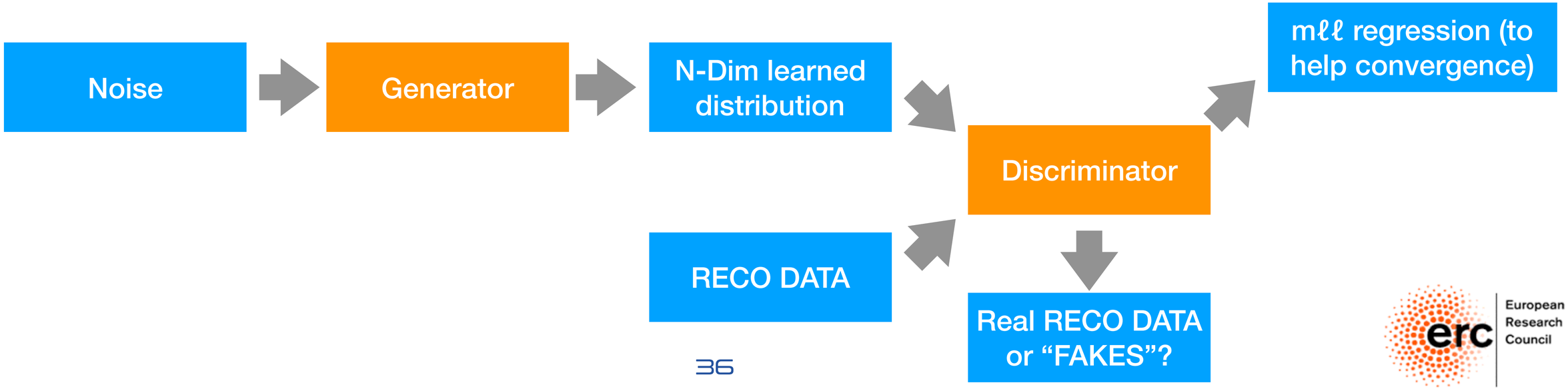
An application-specific approach

- *CPU is only part of the problem. Storage is the other big one (biggest at the moment)*
- *A typical LHC event takes 1 MB of disk / tape. Typically, multiple copies of the same event are stored in different sites worldwide*
- *After all processing, a typical analysis uses a few KB of data: high-level features computed from the 1 MB of raw information (e.g. CMS nanoAOD)*
- *Can we use a generative model to go straight to this dataset?*
  - *PRO: big save both on disk & CPU*
  - *CONS: the training setup is analysis specific. Several Generative Models will be needed to cover all use cases*



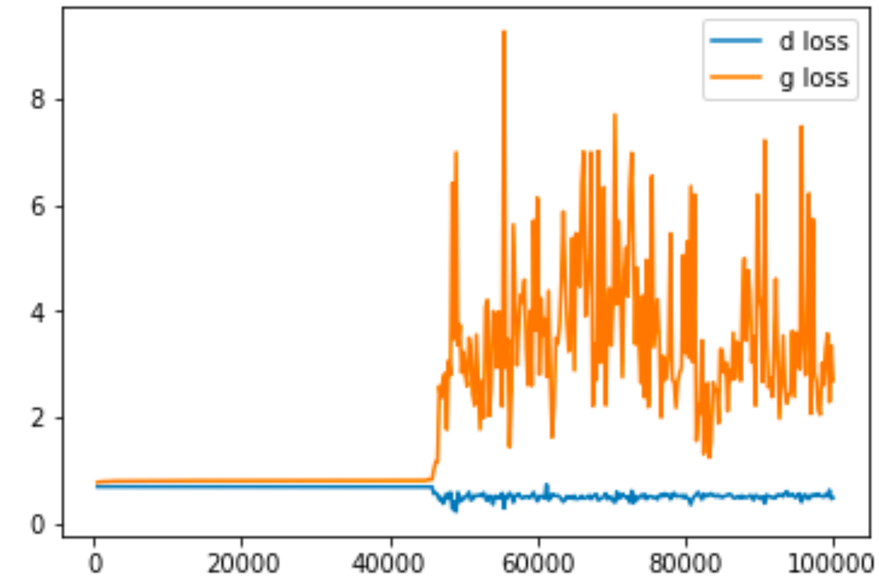
RECO→ RECO GAN

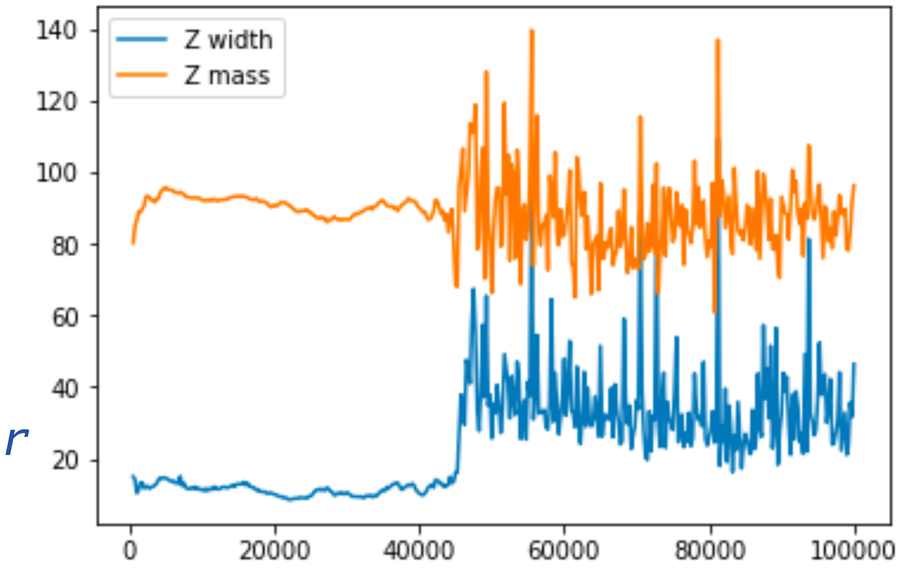
- *We consider a classic GAN setup as baseline (also tried wGAN)*
  - *Train a generator an discriminator in an adversarial fashion*
  - *Add regression of $m_{\ell\ell}$ to the generator cost function, in order to stabilize the training*
- *Implemented in keras+TF*
- *Running on server mounting GTX1080 cards + CSCS Piz Daint (project cn01)*



Training

- *Loss function:* ***cross entropy + c · mse($m_{\ell\ell}$)***
  - *fixed c = 0.01*
- *Dataset size: 2M events*
- *100K epochs / 512 events per batch*
- *Multiple trainings of the same model with randomized starting point, to minimise dependency on initial conditions*
- ***(as normal with GANs) training quite unstable and wGAN didn't really help***
  - *Cannot use the loss function itself as a guide to the best model*
  - *instead, work on defining a generator quality assessment based on statistics tests*





Generator quality

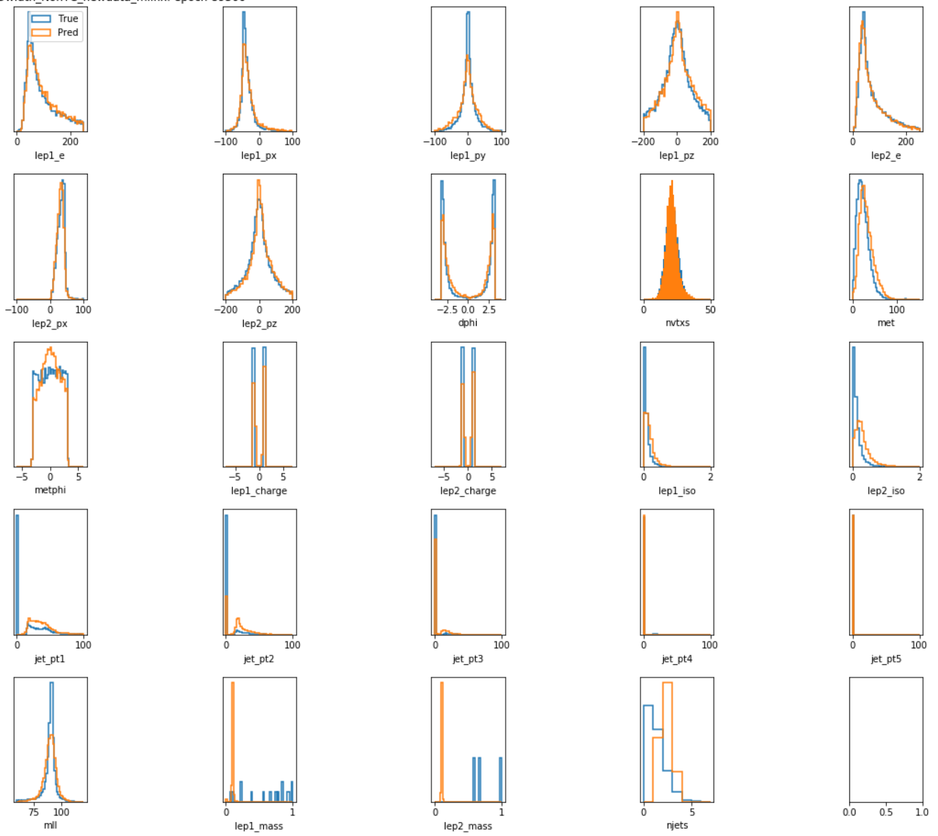
- *Define global- and feature-specific quality tests*
  - Statistics Score (SS) $= \frac{1}{19^2}\sum_{i,j}^{19}\frac{\Delta\sigma_{ij}}{\sigma_{ij}^{\text{real}}} + \frac{1}{19}\sum_{i}^{19}\frac{\Delta\mu_i}{\mu_i^{\text{real}}}$.
    - Roughly the normalized sum of the difference between covariances and averages.
    - $\Delta\sigma_{ij}$ = difference between the covariance matrix elements in the real and generated data
    - $\sigma_{ij}^{\text{real}}$ = covariance matrix element for real data
    - $\Delta\mu_i$ = difference between the average value of the ith variable in the real and generated data
    - μ_i^{real} = average value of the ith variable in the real data
  - MLLKS = Kolmogorov-Smirnov test statistic for the MLL
  - MetPhiKS = Kolmogorov-Smirnov test statistic for the MET-ϕ
  - LepIsoKS = Kolmogorov-Smirnov test statistic for the leading lepton isolation
- *Work in progress: investigating usage of standard pdf-distance definitions (KL divergence, earth-mover distance, etc)*

Results

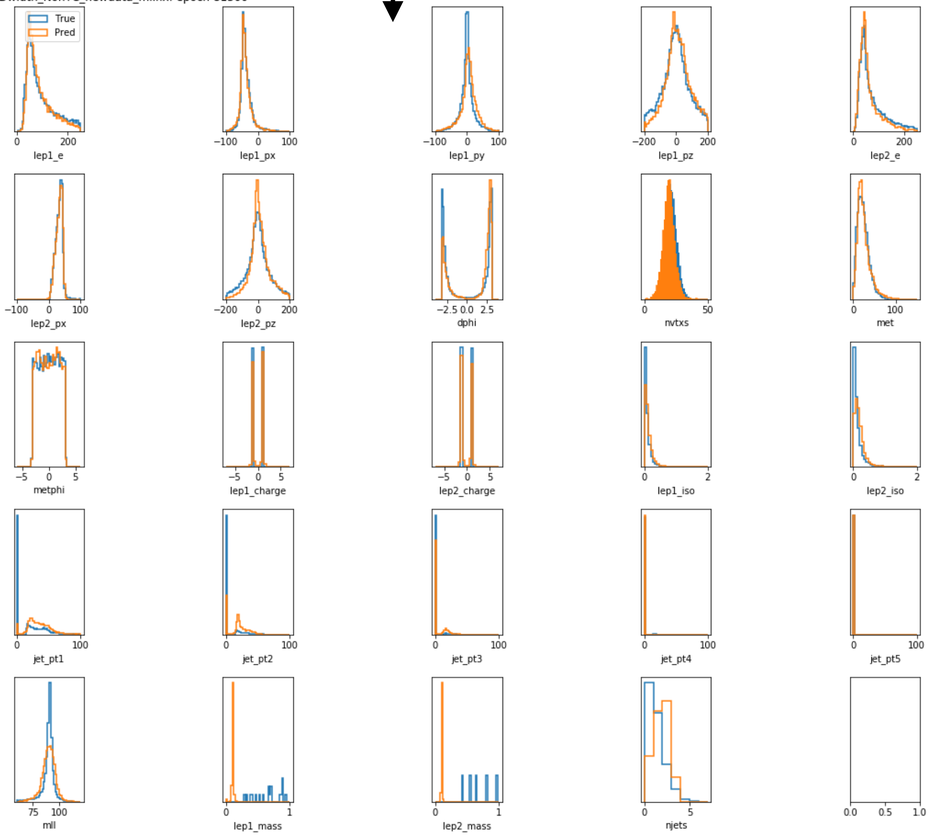
Trial	Epoch	LepIsoRank	MetPhiRank	MLLRank	ScoreRank	StatsScore	MLLKS	MetPhiKS	LepIsoKS	SortKey
12	69500	160	298	59	43	27.826635	0.085580	0.032840	0.236480	560.000000
12	99000	34	166	107	392	47.069988	0.096820	0.024580	0.188500	699.000000
6	29000	144	309	79	228	40.682663	0.090900	0.033440	0.232180	760.000000
12	53000	571	139	23	67	30.737935	0.075540	0.022300	0.305880	800.000000
7	89500	85	448	88	181	38.685347	0.092740	0.041520	0.214840	802.000000
12	81500	284	74	306	154	37.152292	0.122060	0.013740	0.263080	818.000000

Probably Best Overall

tch512_bgbd_mllANDwidth_NonTC_newdata_mllfix: epoch 89500

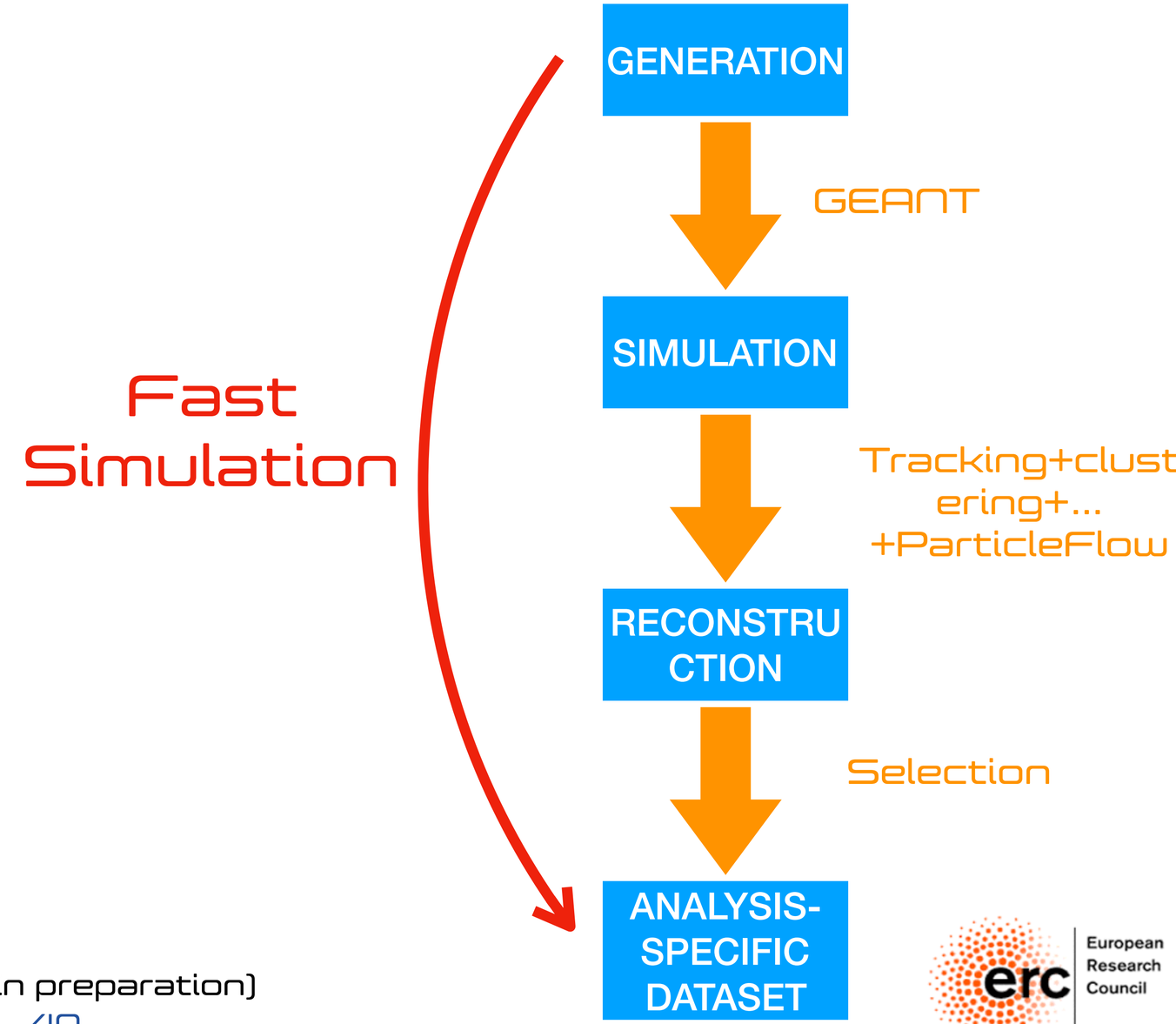


atch512_bgbd_mllANDwidth_NonTC_newdata_mllfix: epoch 81500



GEN→ RECO GAN

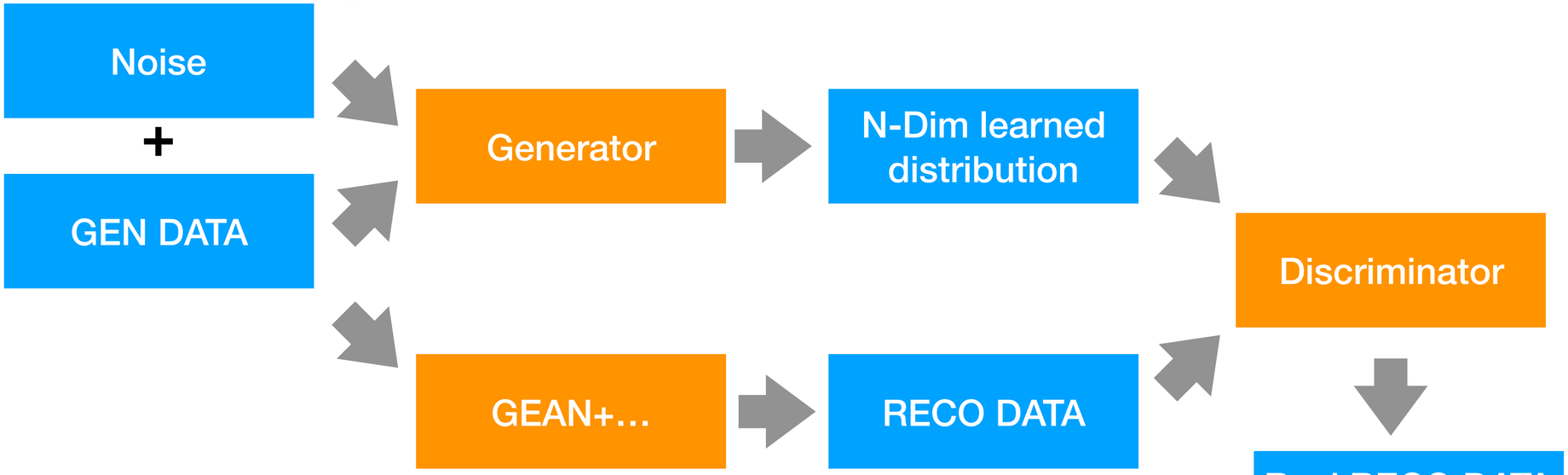
- In view of large statistics needs, one can use generative models as statistics augmentation tools
- For instance, could generate expert-feature quantities used in an analysis (muon four-momenta, jet momenta, etc.)
- Like sampling from histogram with two main advantages
  - no need to bin
  - generalizes to multi-dimensional problems



with K. Dutta, N. Amin, B. Hashemi and D. Olivito (in preparation)

An example

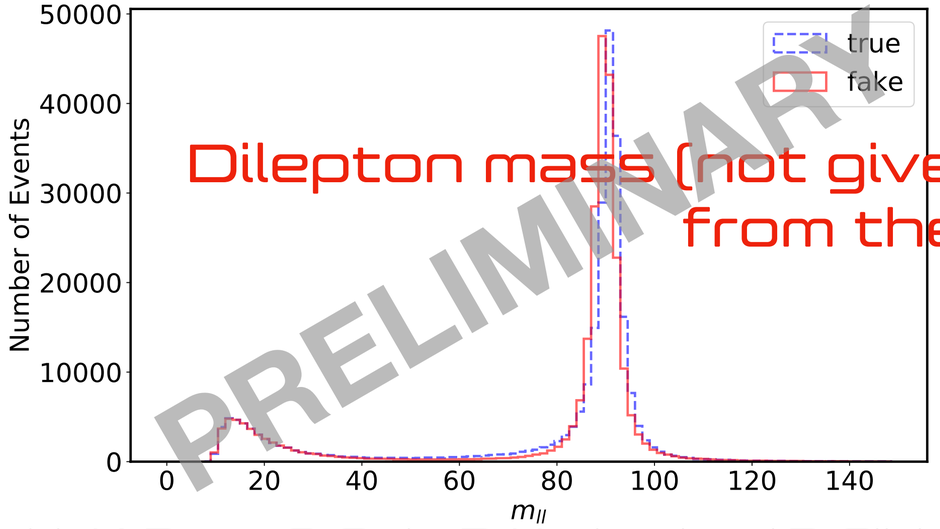
- *Dimuon events at LHC*
- *Typical analysis would use a few handful of quantities (muon momenta, isolation, jet pTs, etc)*
- *Can learn the N-dim distribution of these quantities with GAN setup*
- *Can use the generator network as a fastsim tool*

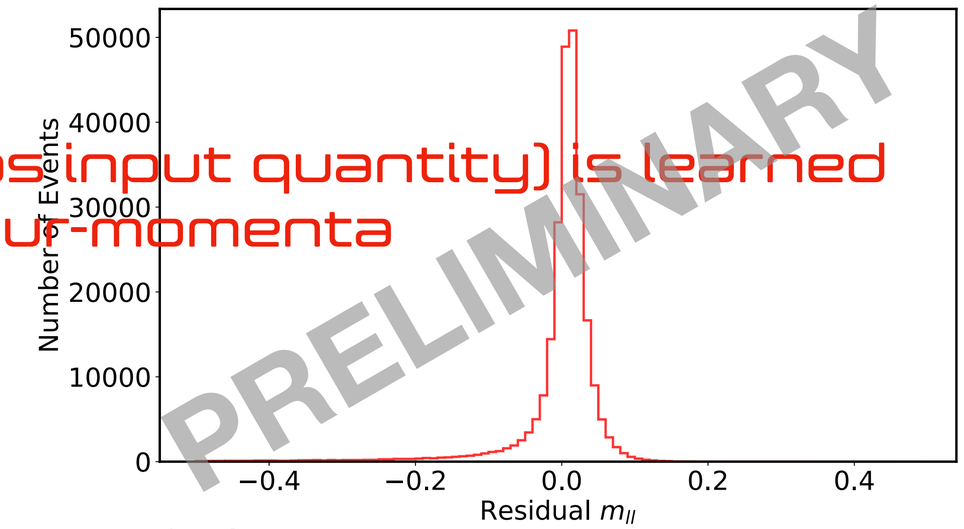


with K. Dutta, N. Amin, B. Hashemi and D. Olivito (in preparation)

An example

- *Dimuon events at LHC*
- *Typical analysis would use a few handful of quantities (muon momenta, isolation, jet pTs, etc)*
- *Can learn the N-dim distribution of these quantities with GAN setup*
- *Can use the generator network as a fastsim tool*





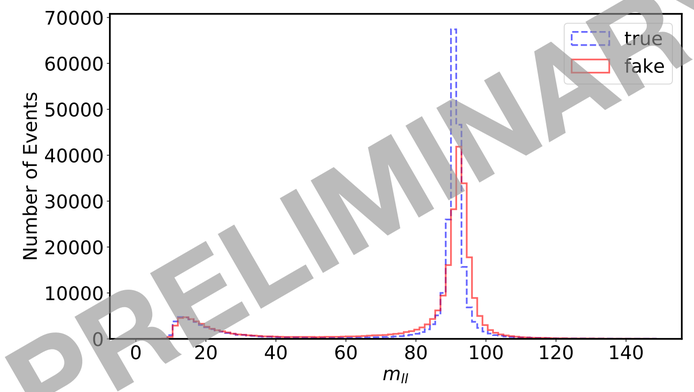
Dilepton mass (not given as input quantity) is learned from the four-momenta

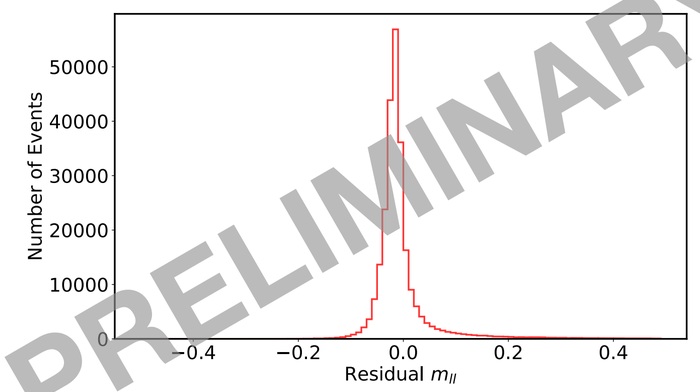
with K. Dutta, N. Amin, B. Hashemi and D. Olivito (in preparation)

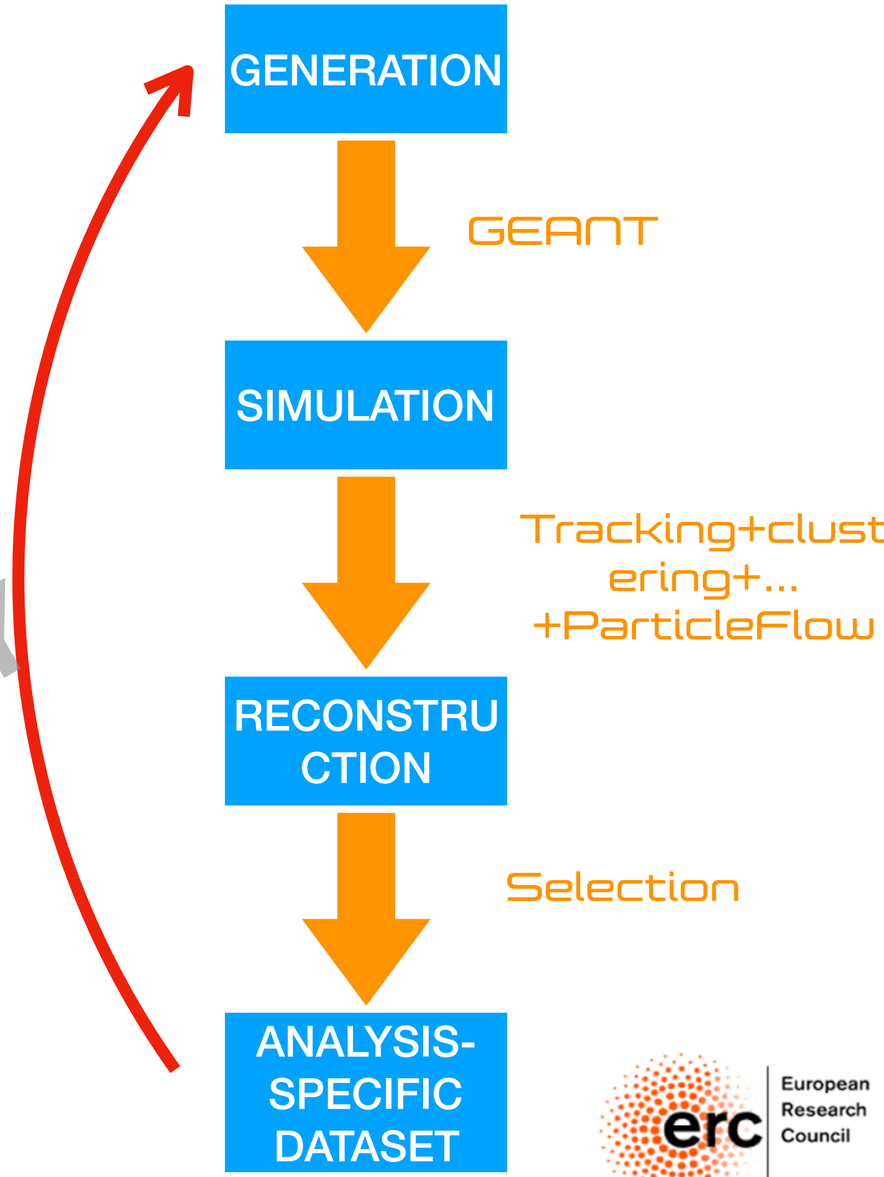
Analysis-specific unfolding

- One could invert the process and learn to predict the GEN features from the RECO ones
- Formally the same procedure
- Invert the role of the target and the input datasets

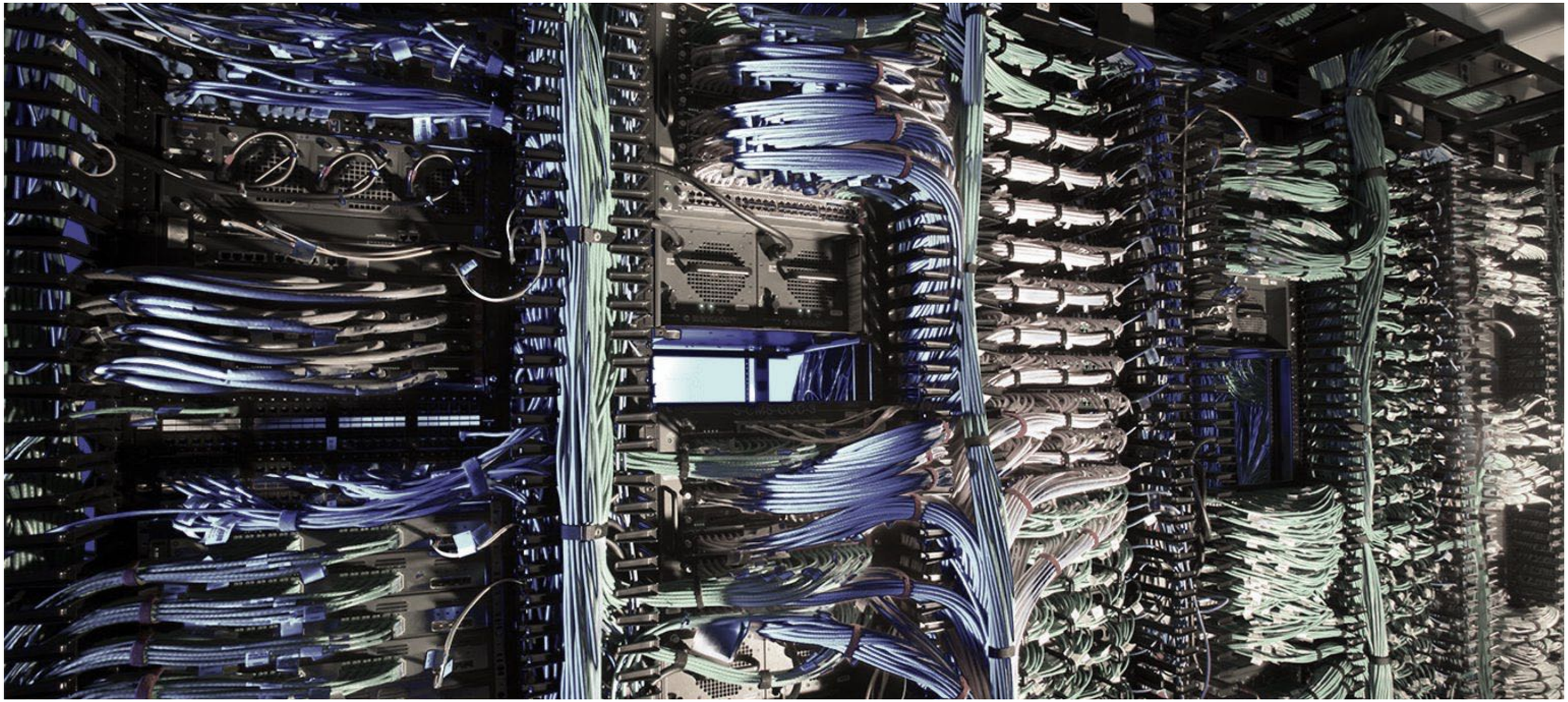
Unfolding







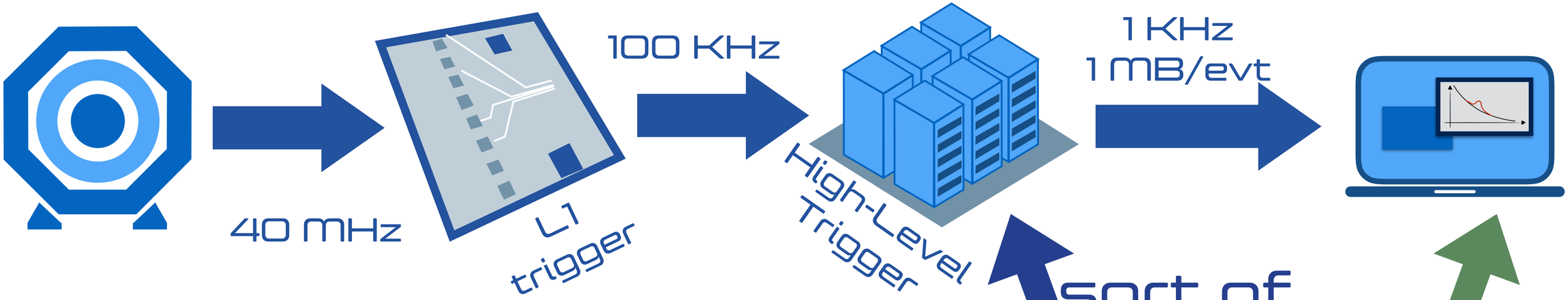
with K. Dutta, N. Amin, B. Hashemi and D. Olivito (in preparation)



Fast Decision Taking

European Research Council

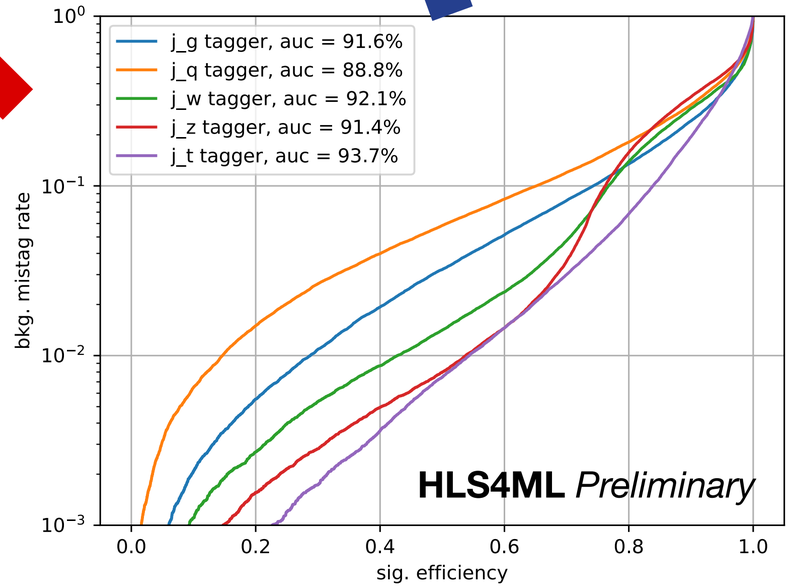
The LHC Big Data Problem

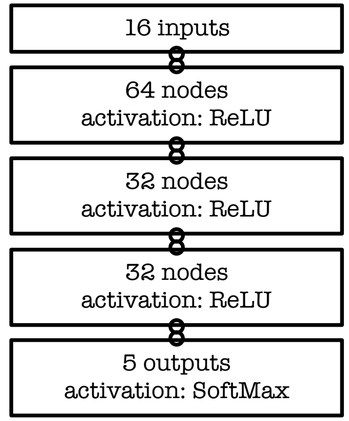


sort of OK

Easy

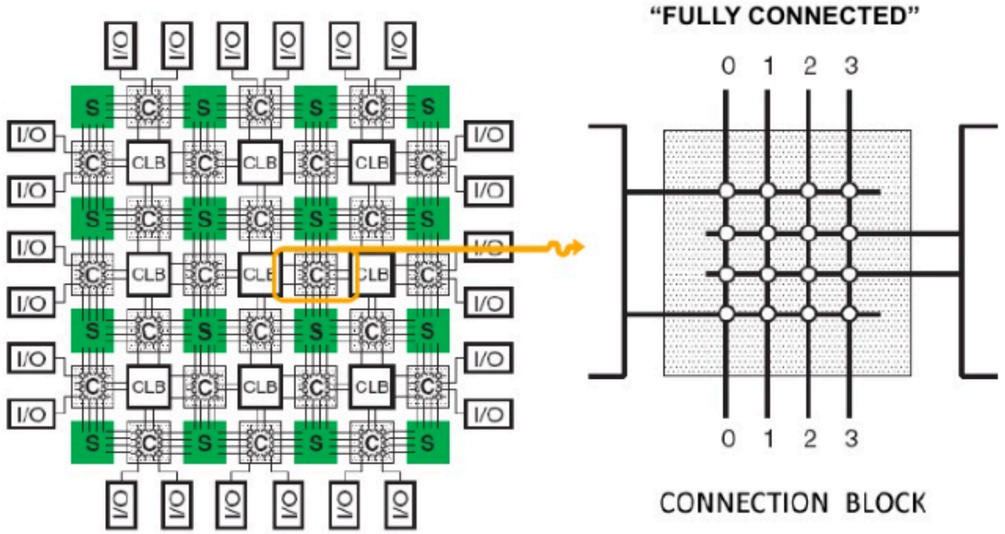
Complicated: need to be fast (10 ms) and with very small resources



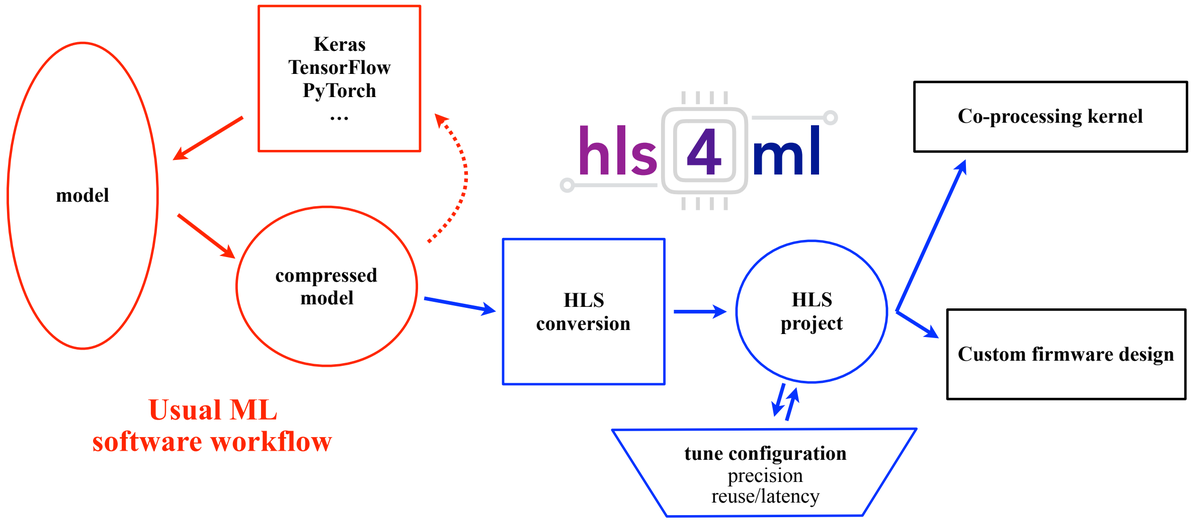


Bring DL to L1

- *The L1 trigger is a complicated environment*
  - *decision to be taken in ~10 µsec*
  - *only access to local portions of the detector*
  - *processing on Xilinx FPGA, with limited memory resources*
- *Some ML already running @L1*
  - *CMS has BDT-based regressions coded as look-up tables*
- *Working to facilitate DL solutions @L1 with dedicated library*

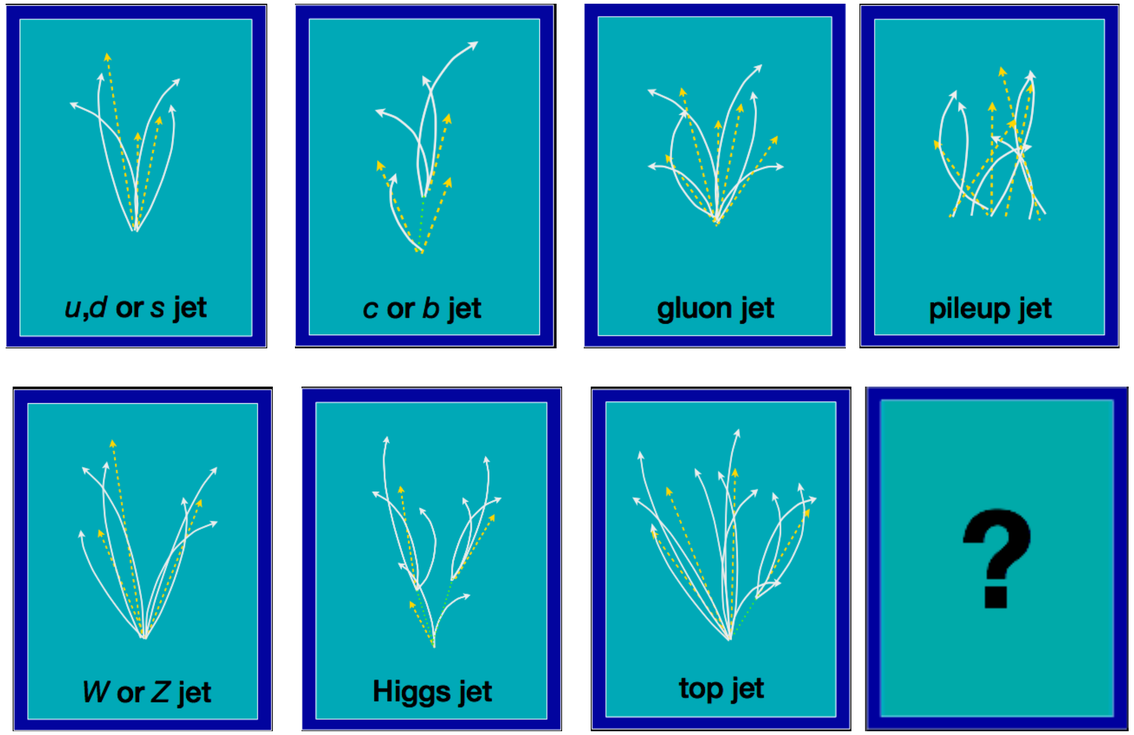


HLS4ML: CERN/FNAL/MIT collaboration



Example: jet tagging

- *You have a jet at LHC: spray of hadrons coming from a "shower" initiated by a fundamental particle of some kind (quark, gluon, W/Z/H bosons, top quark)*
- *You have a set of jet features whose distribution depends on the nature of the initial particle*
- *You can train a network to start from the values of these quantities and guess the nature of your jet*
- *To do this you need a sample for which you know the answer*



Example: jet tagging

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

16 inputs

↓

64 nodes
activation: ReLU

↓

32 nodes
activation: ReLU

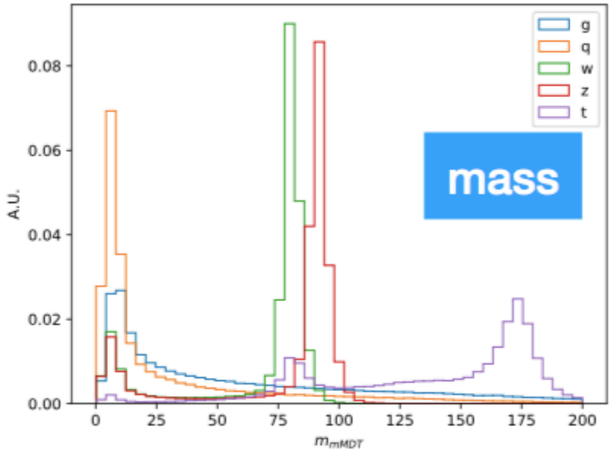
↓

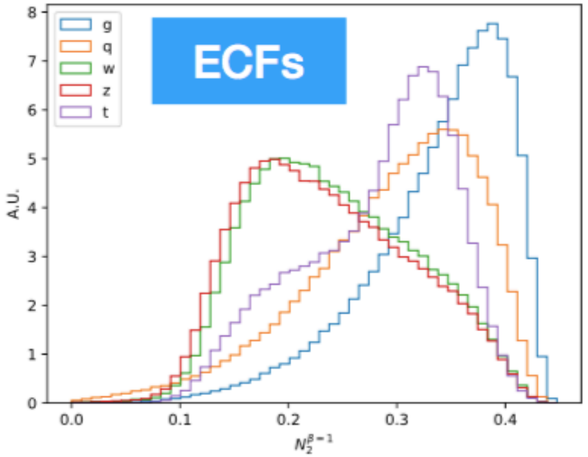
32 nodes
activation: ReLU

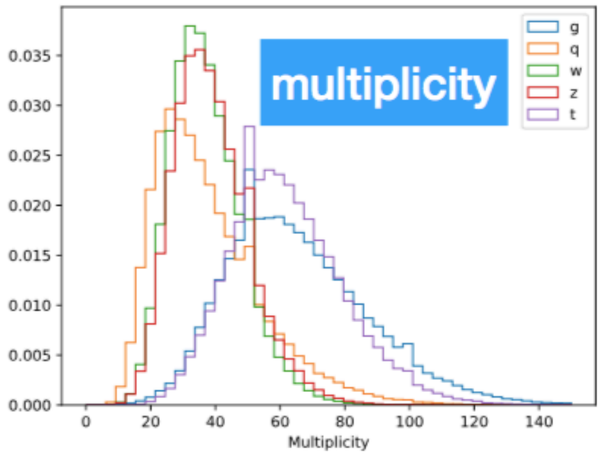
↓

5 outputs
activation: SoftMax

Each input layer transmits a feature to the following hidden layer

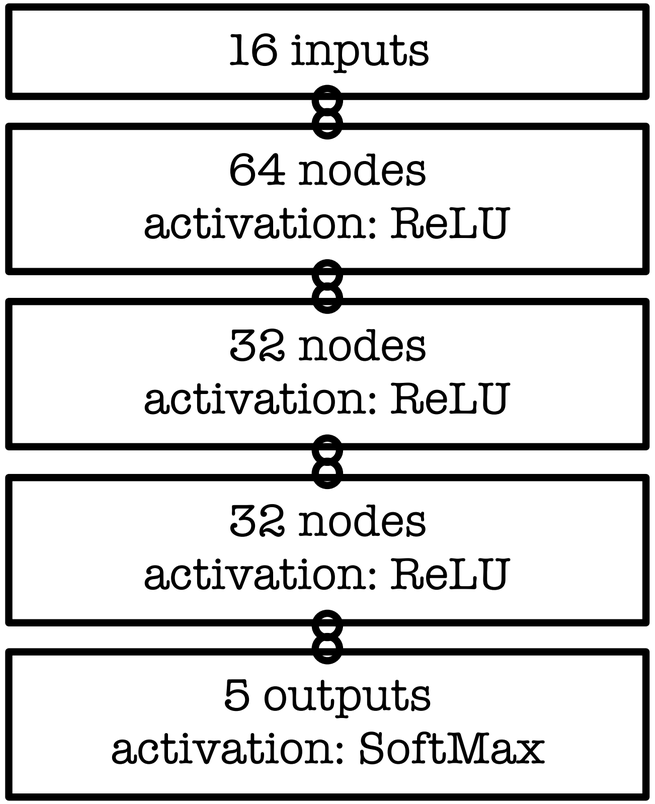


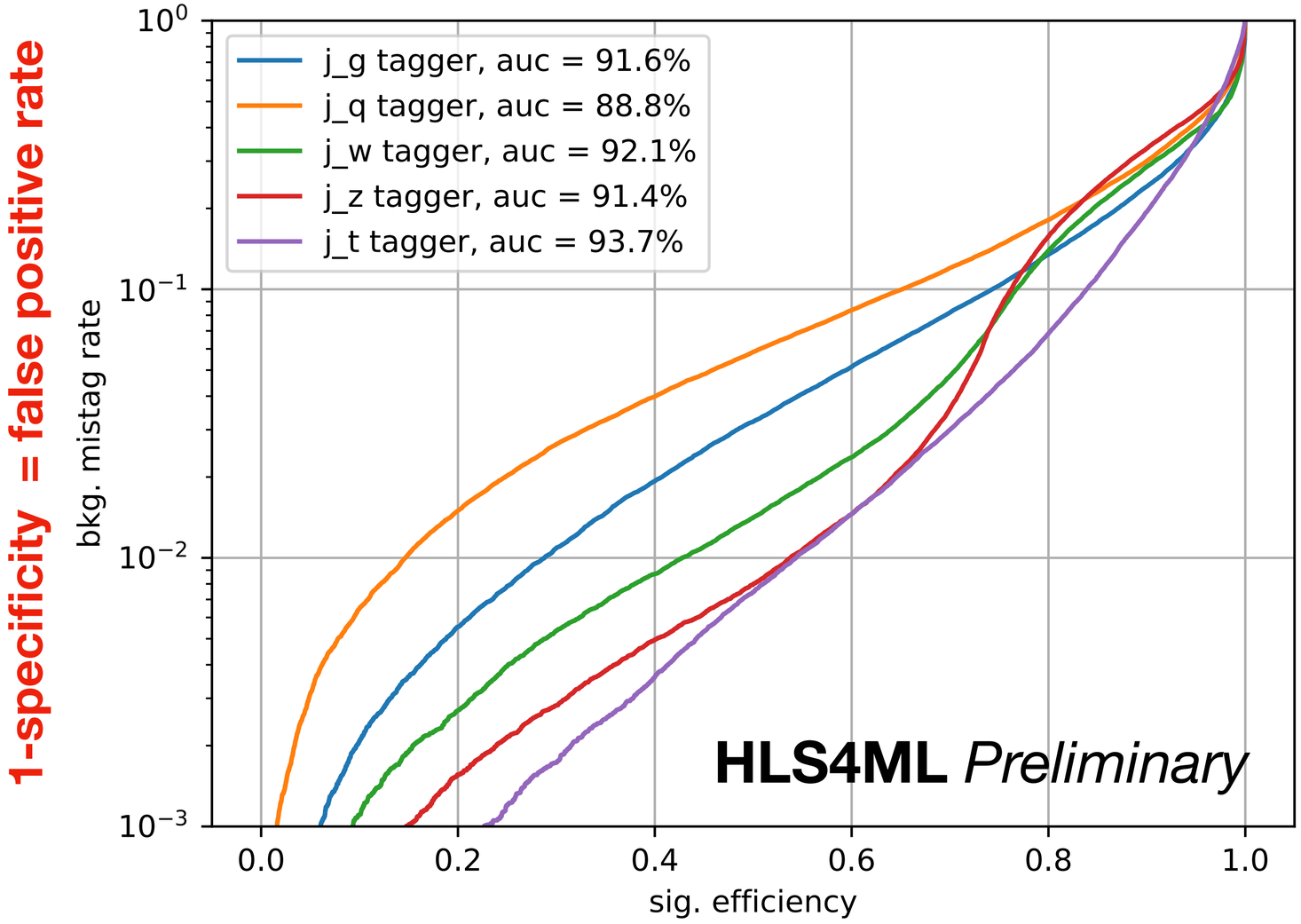




Example: jet tagging

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



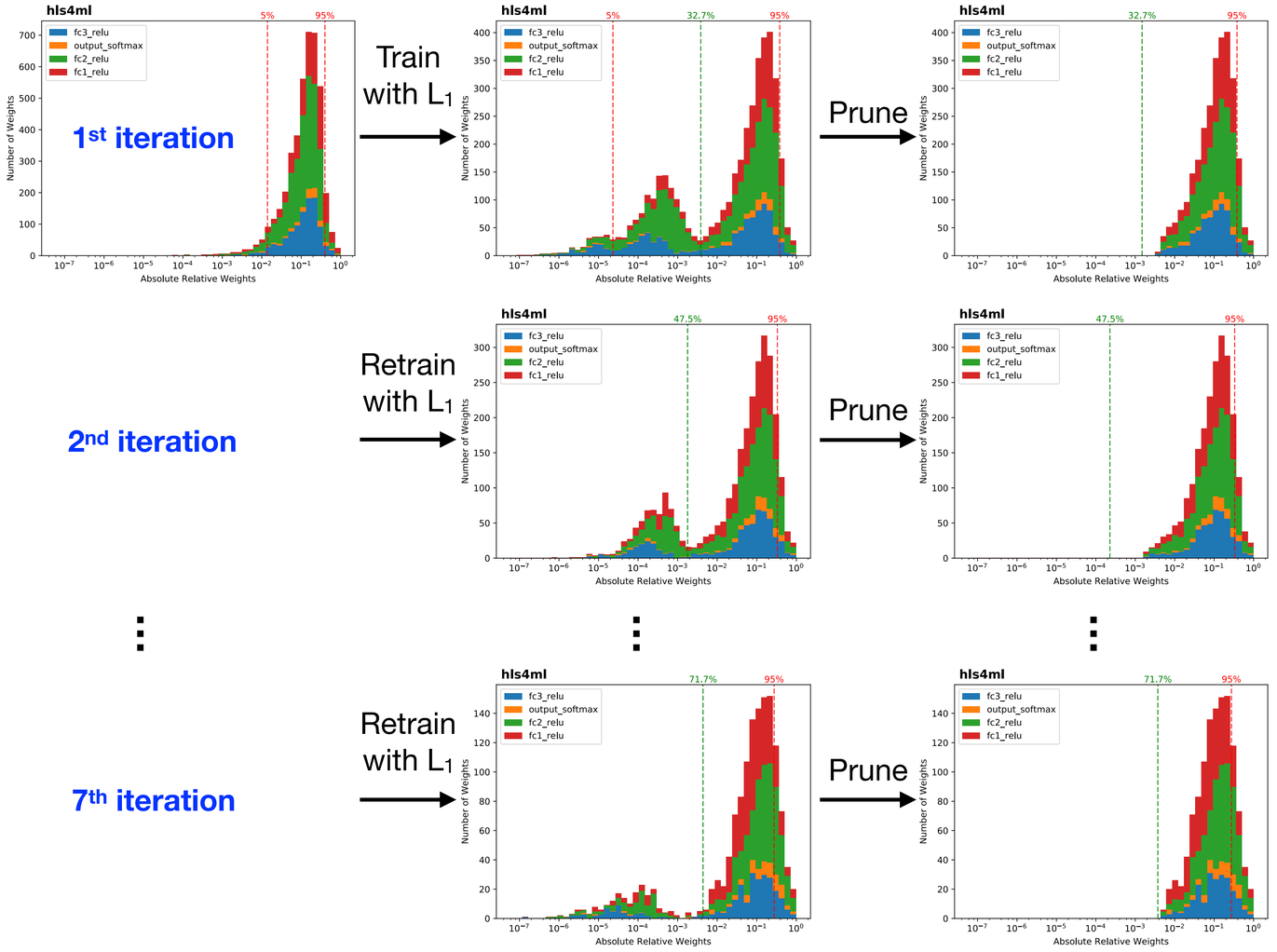


Make the model cheaper

- ***Pruning:*** *remove parameters that don't really contribute to performances*
  - *force parameters to be as small as possible (regularization)*

$$L_\lambda(\vec{w}) = L(\vec{w}) + \lambda||\vec{w}_1||$$

  - *Remove the small parameters*
  - *Retrain*

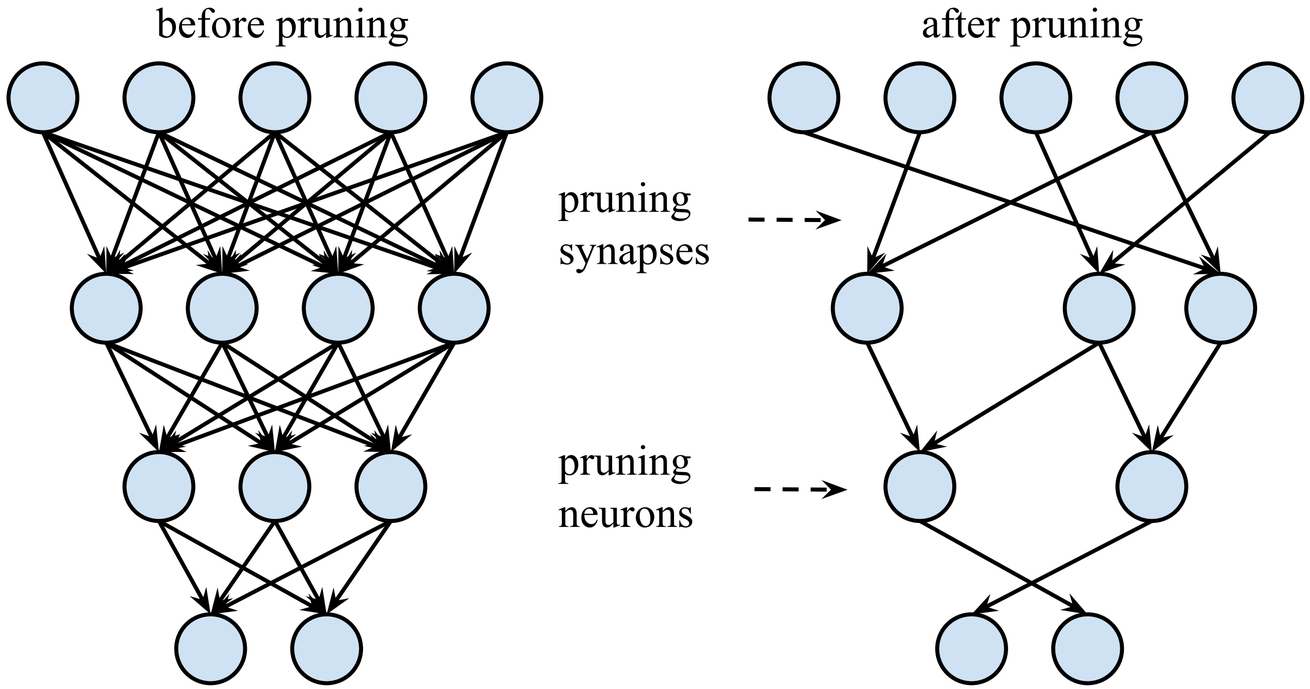


Make the model cheaper

- ***Pruning:*** *remove parameters that don't really contribute to performances*
  - *force parameters to be as small as possible (regularization)*

$$L_\lambda(\vec{w}) = L(\vec{w}) + \lambda||\vec{w}_1||$$

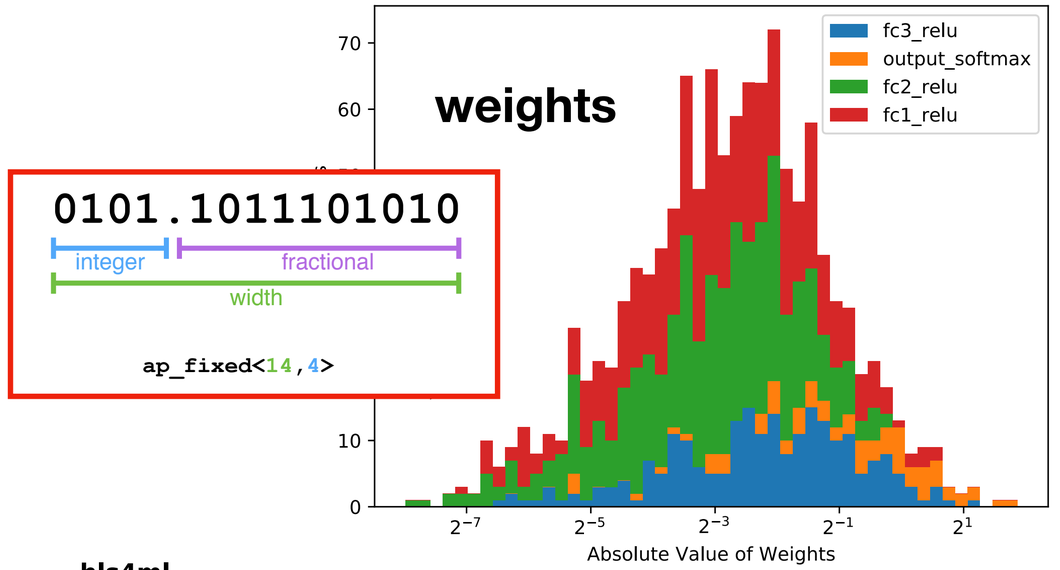
  - *Remove the small parameters*
  - *Retrain*

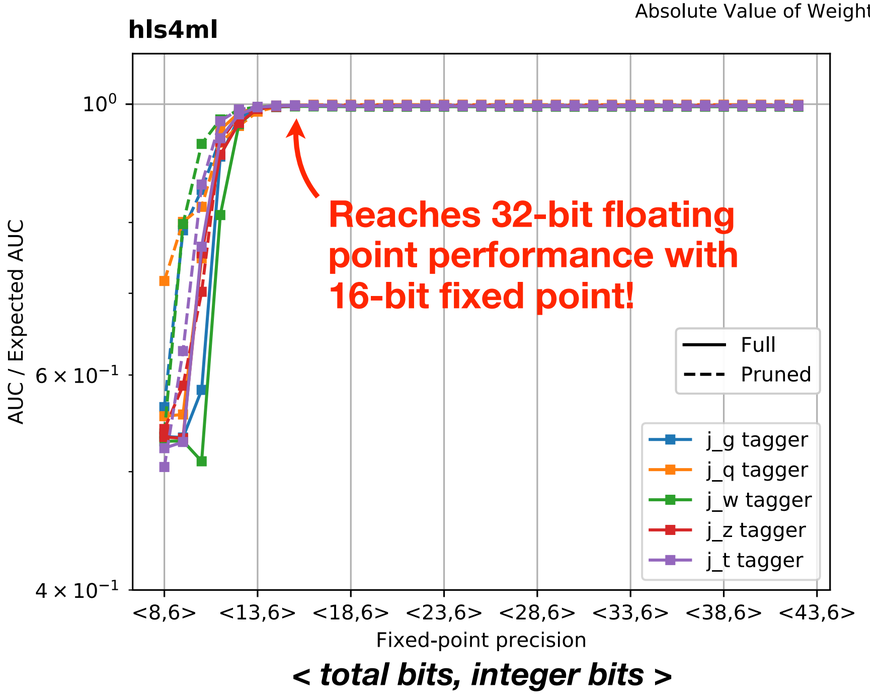


→ 70% reduction of weights and multiplications w/o performance loss

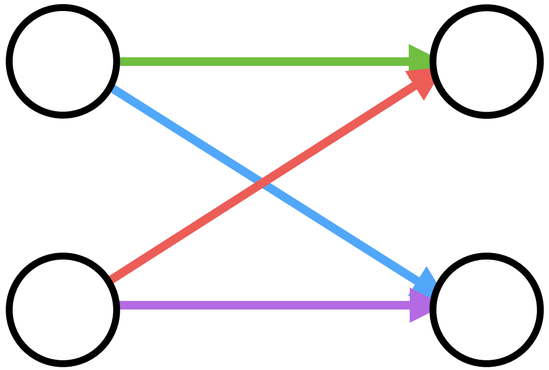
Make the model cheaper

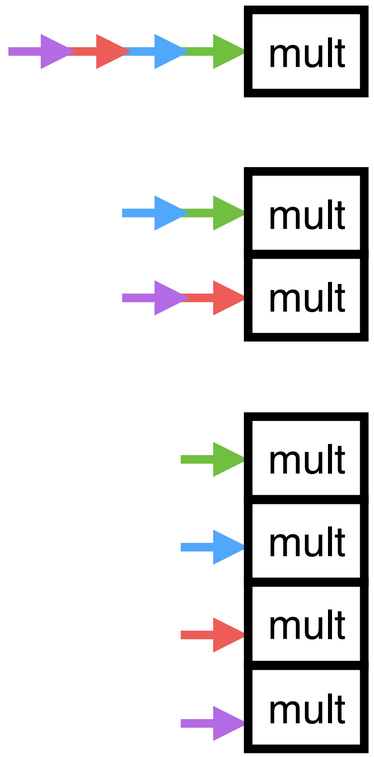
- ***Quantization***: *reduce the number of bits used to represent numbers (i.e., reduce used memory)*
  - *models are usually trained at 64 or 32 bits*
  - *this is not necessarily needed in real life*
- *In our case, we could reduce to 16 bits w/o loosing precision*
- *Beyond that, one would have to accept some performance loss*





Speed vs Memory





Fully serial

reuse = 4
use 1 multiplier 4 times

reuse = 2
use 2 multipliers 2 times each

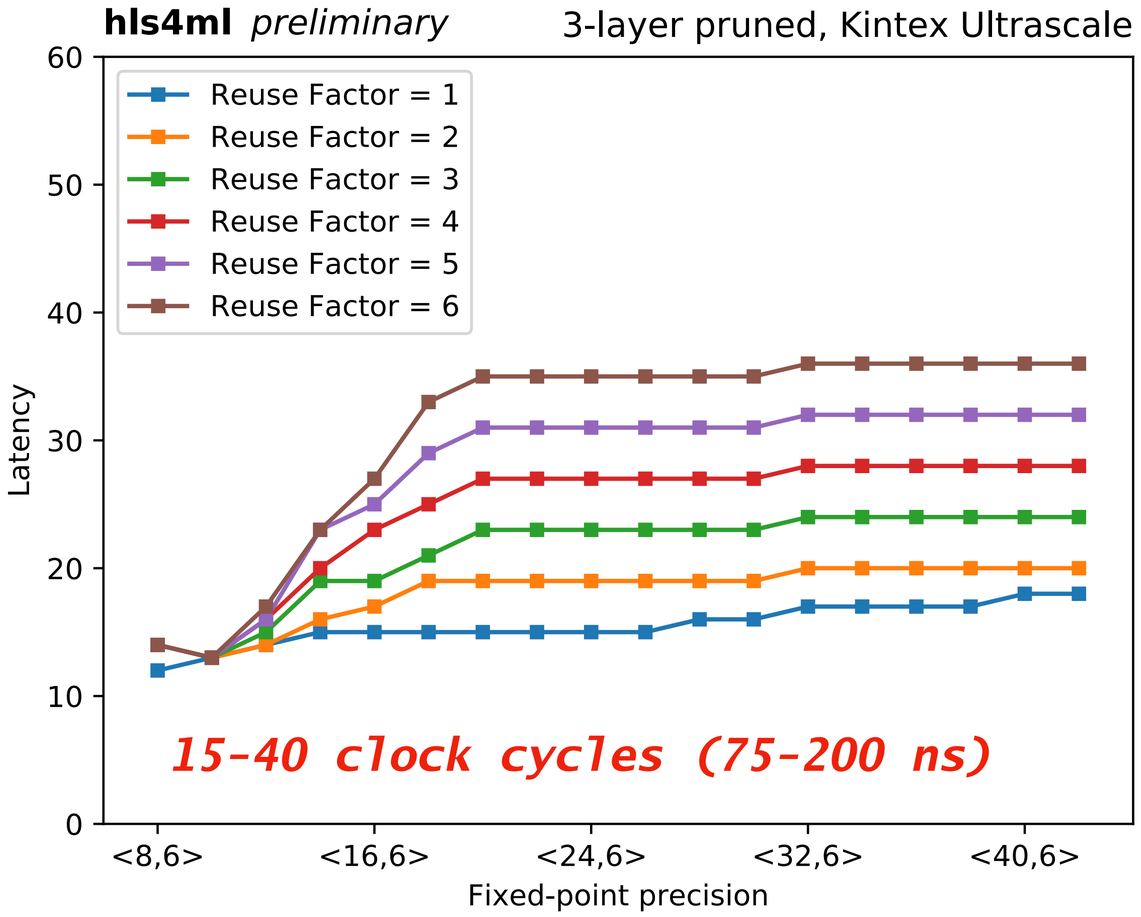
reuse = 1
use 4 multipliers 1 time each

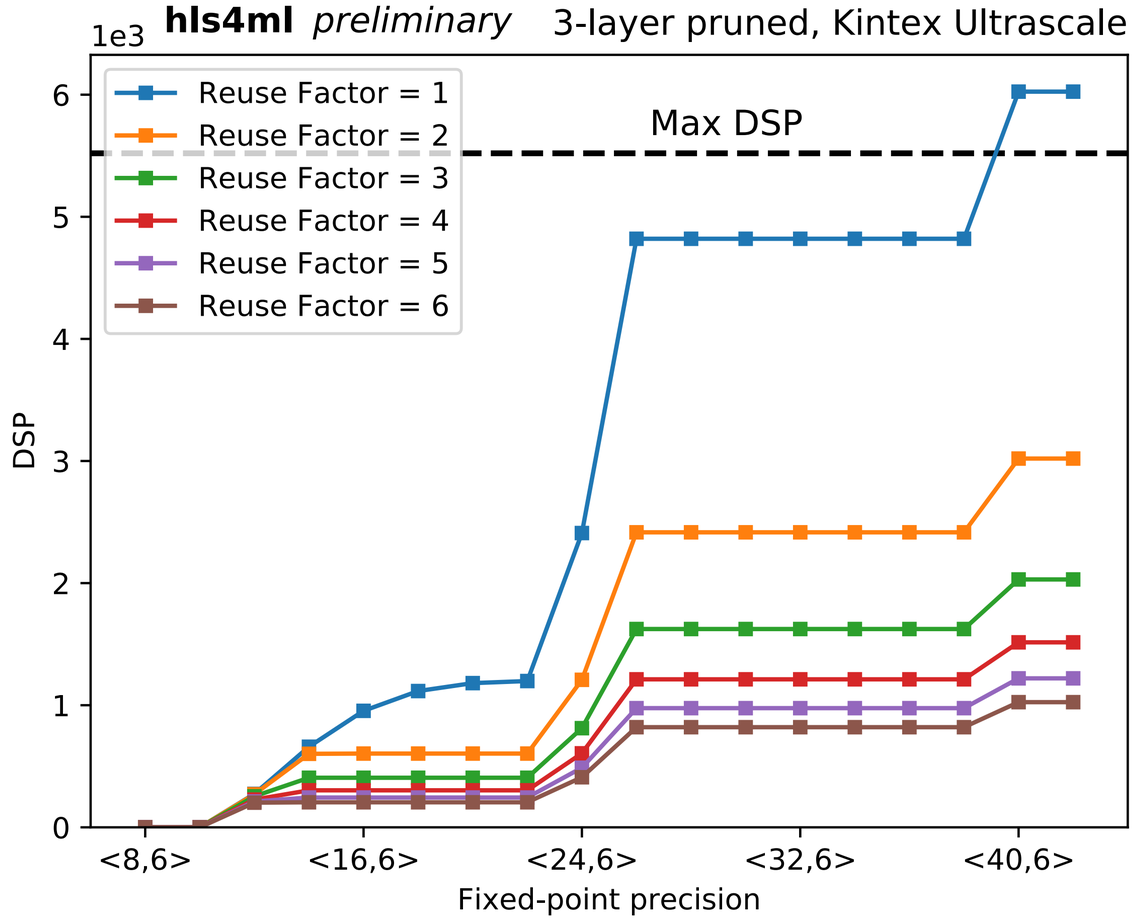
Fully parallel

Less resources/
Less throughput

Reuse factor: how much to parallelize operations in a hidden layer

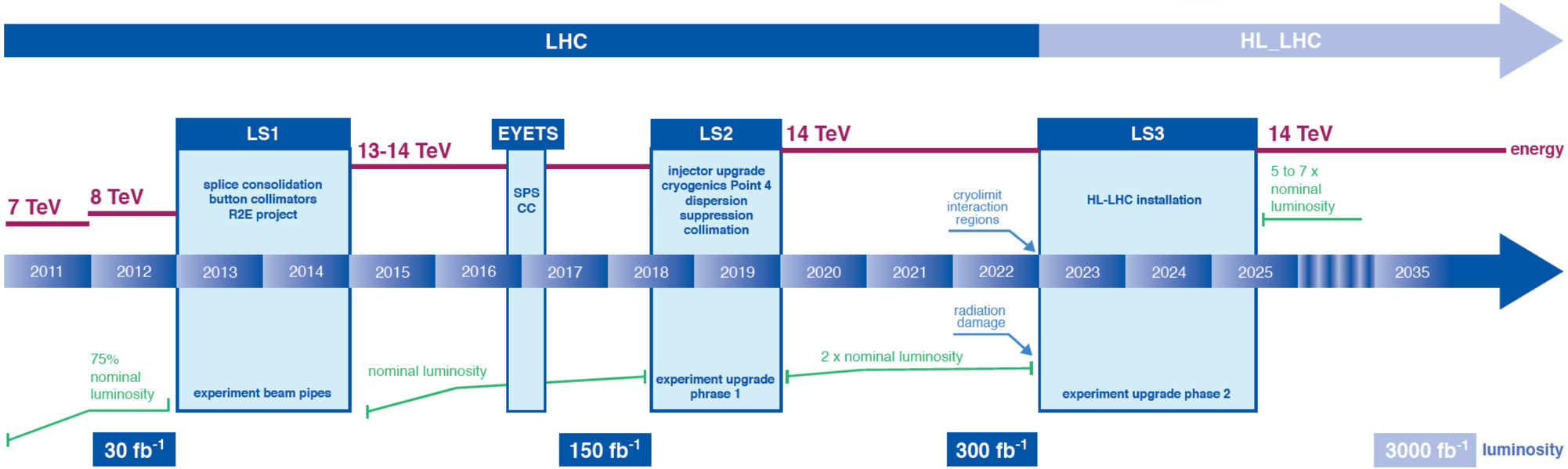
Performances





NB: FPGA emulator over-estimates resource needs by a factor 2-4 (tested our emulation vs actual deployment)

A roadmap



- ***We need to be ready by 2025 (High-Luminosity LHC)***
- *LHC Run 3 (2020-2022) is the ultimate demonstration opportunity*
  - *for model building, deployment and commissioning*
- ***Strong synergy with other research lines in HEP**, when Deep Learning is even easier to apply*
  - *Dark Matter underground experiments*
  - *Neutrino experiments*

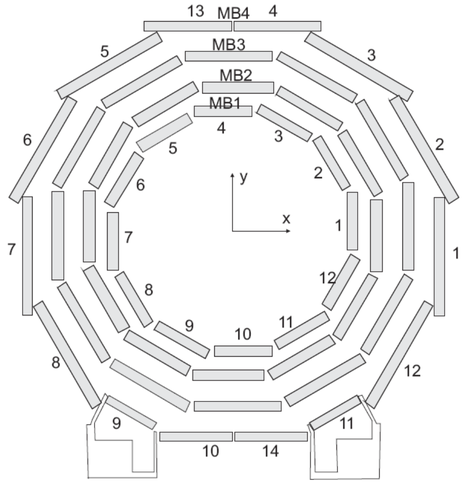
Backup

Detector Monitoring

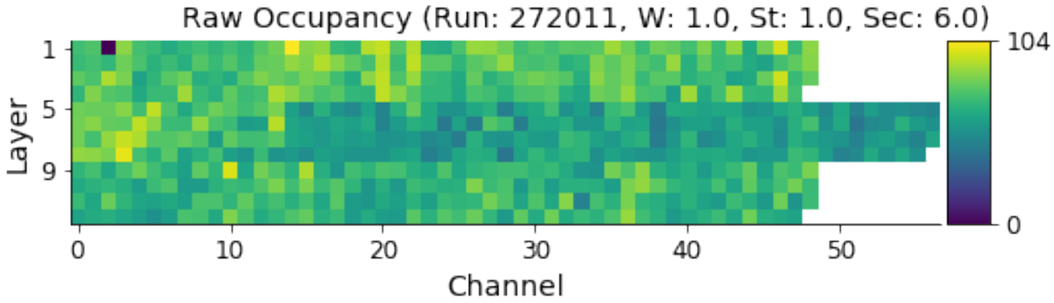
European Research Council

Data Quality Monitoring

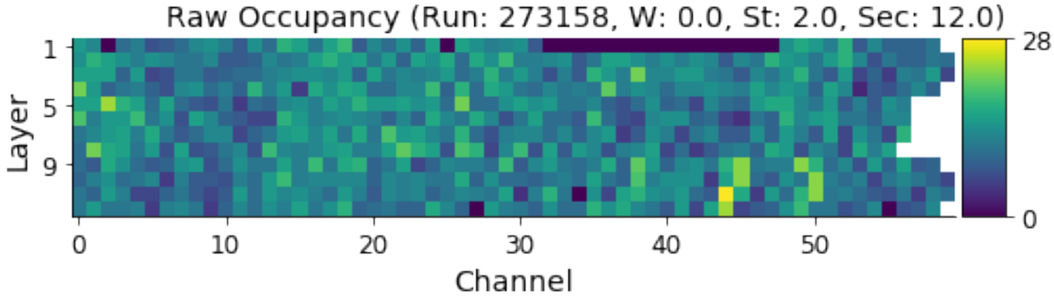
- *When taking data, >1 person watches for anomalies in the detector 24/7*
- *At this stage no global processing of the event*
- *Instead, local information from detector components available (e.g., detector occupancy in a certain time window)*



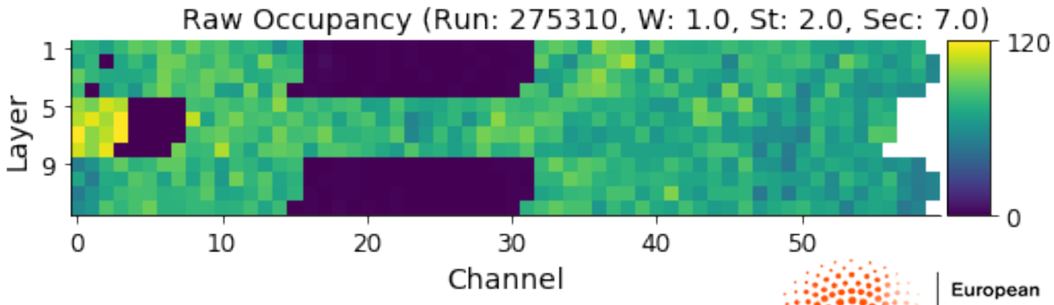
A



B



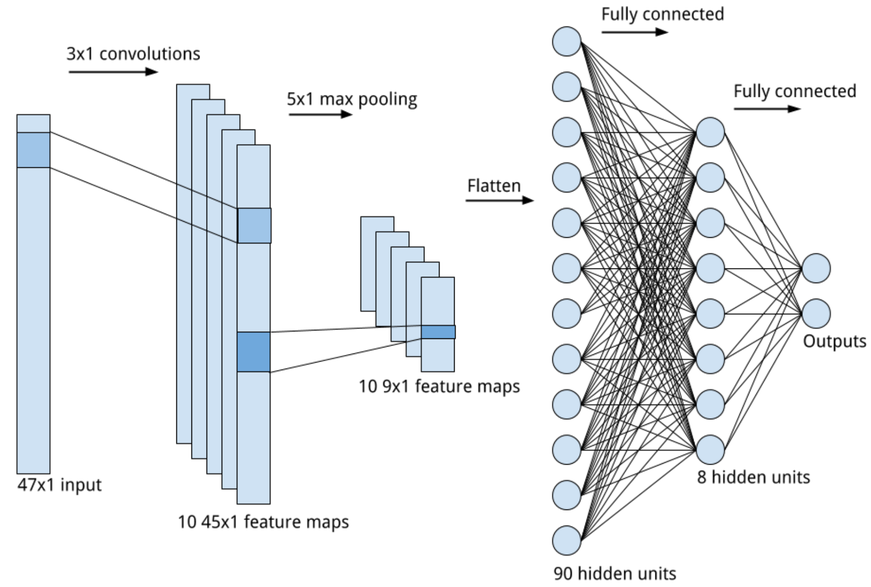
C

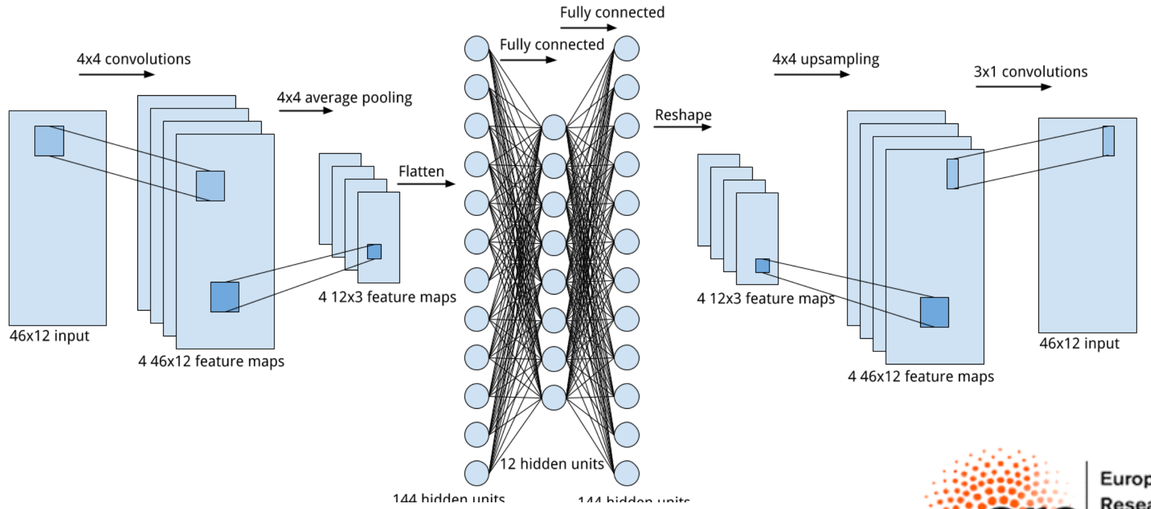


Two approaches

- *Given the nature of these data, ConvNN are a natural analysis tool. Two approaches pursued*
  - *Classify good vs bad data. Works if failure mode is known*
  - *Use autoencoders to assess data "typicality". Generalises to unknown failure modes*

A. Pol et al., to appear soon

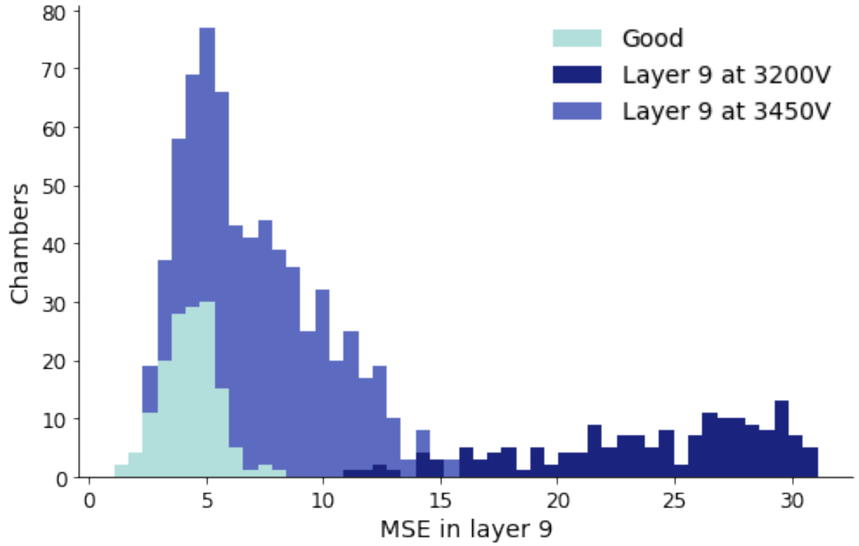


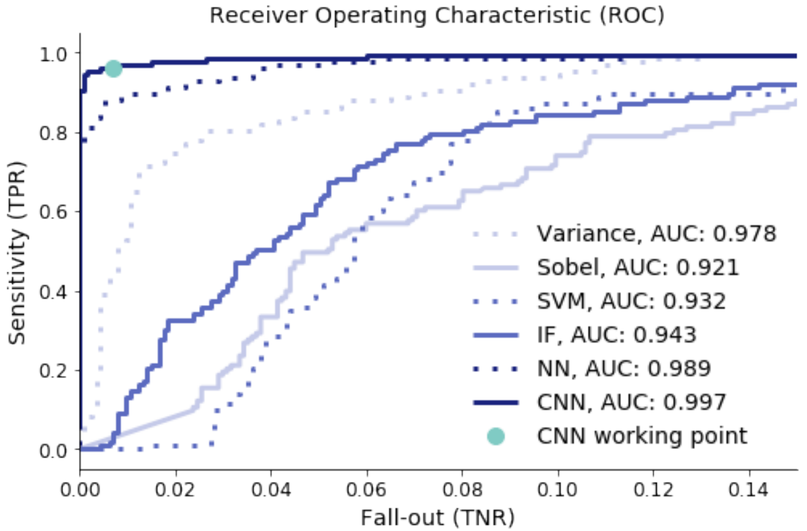


Two approaches

- *Given the nature of these data, ConvNN are a natural analysis tool. Two approaches pursued*
  - *Classify good vs bad data. Works if failure mode is known*
  - *Use autoencoders to assess data "typicality". Generalises to unknown failure modes*

A. Pol et al., to appear soon

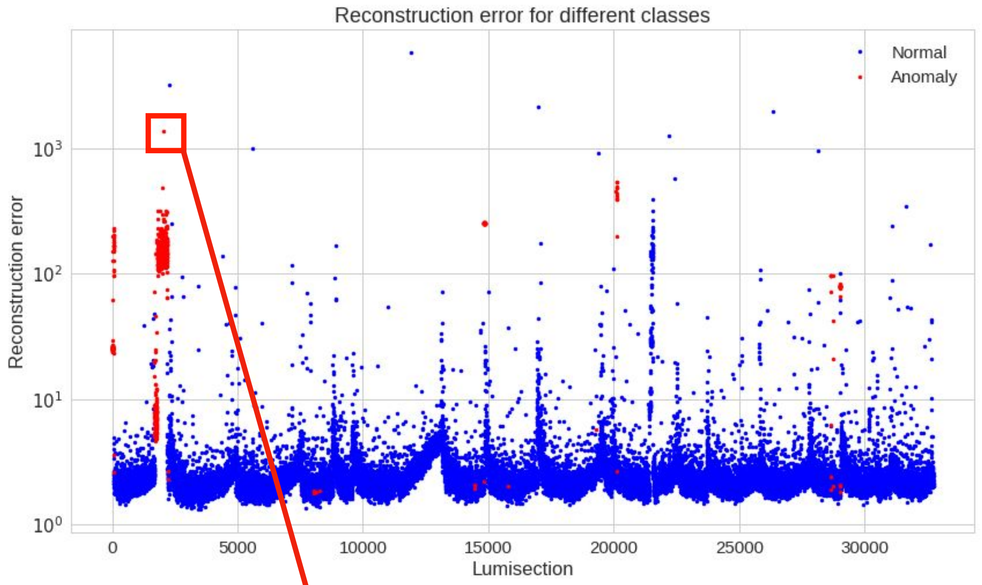


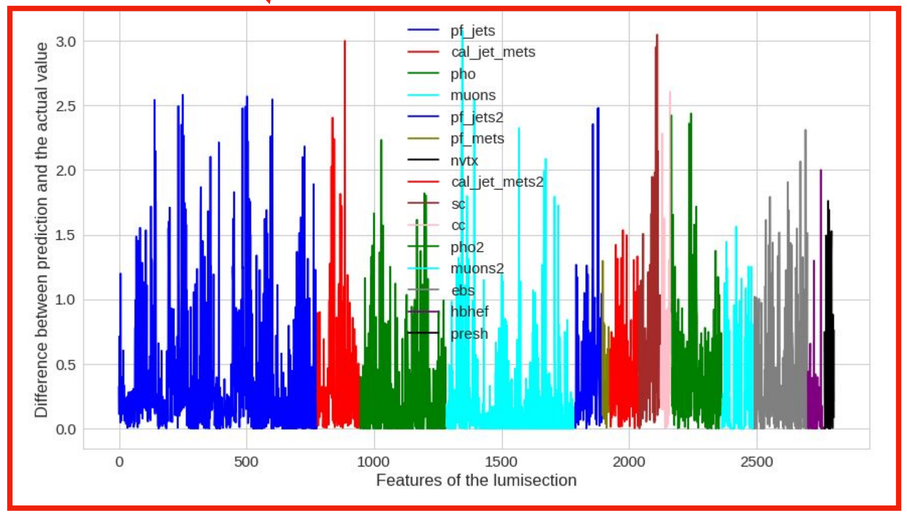


European Research Council

Data Quality Certification

- *Autoencoder-based 1-class approach generalises to later stages of quality assessment*
  - *after reconstruction of the events, event reconstruction allows a global assessment (w.g., looking at electrons, muons, etc rather than hits in the detector)*
  - *A global autoencoder can spot all these features*
  - *Monitoring individual contributions to loss function (e.g., MSE) one can track the problem back to a specific physics object/detector component*





F. Široký et al., to appear sooner or later

HL4ML: FPGA details

Xilinx Vivado 2017.2

Results are slightly different in other versions of Vivado

e.g. 2016.4 optimization is less performant for Xilinx ultrascale FPGAs

Clock frequency: 200 MHz

Latency results can vary (~10%) with different clock choices

FPGA: Xilinx Kintex Ultrascale (XCKU115-FLVB2104)

Results are slightly different in other FPGAs

e.g. Virtex-7 FPGAs are slightly differently optimized

European Research Council

Why Deep Learning

- *Neural network can model non linear functions*
  - *the more complex is the network, the more functions it can approximate*
- *Neural network are faster to evaluate (inference) than typical reco algorithm.*
  - *This is the speed up we need*
- *Neural Networks (unlike other kind of ML algorithms) are very good with raw (non-preprocessed) data (the recorded hits in the event)*
  - *could use them directly on the detector inputs*

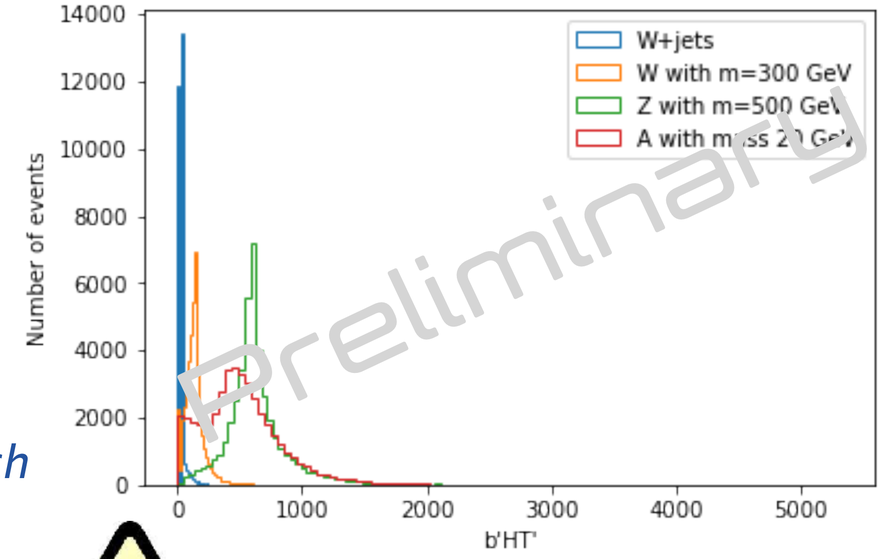
$$(pT, \eta, \phi, E)_{OFFLINE} = f\left((pT, \eta, \phi, E)_{ONLINE}\right)$$

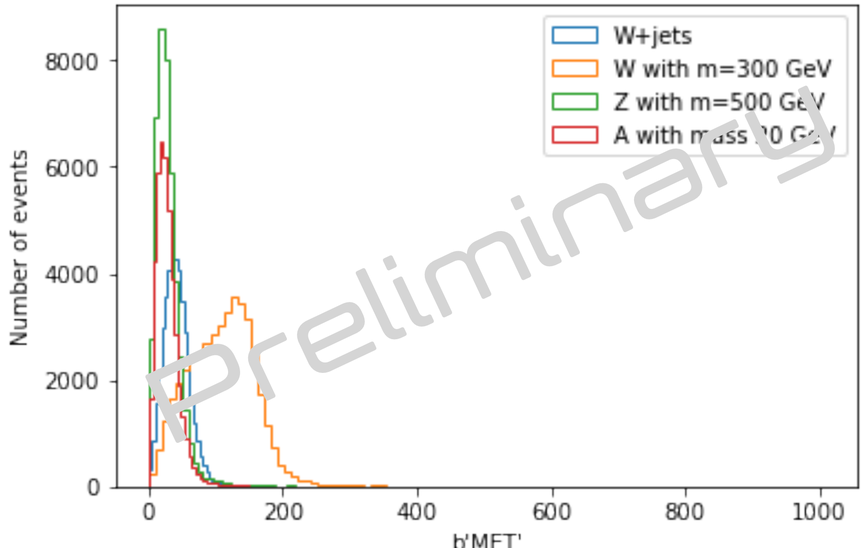
$$(pT, \eta, \phi, E)_{OFFLINE} = g\left(\text{Event hits}\right)$$

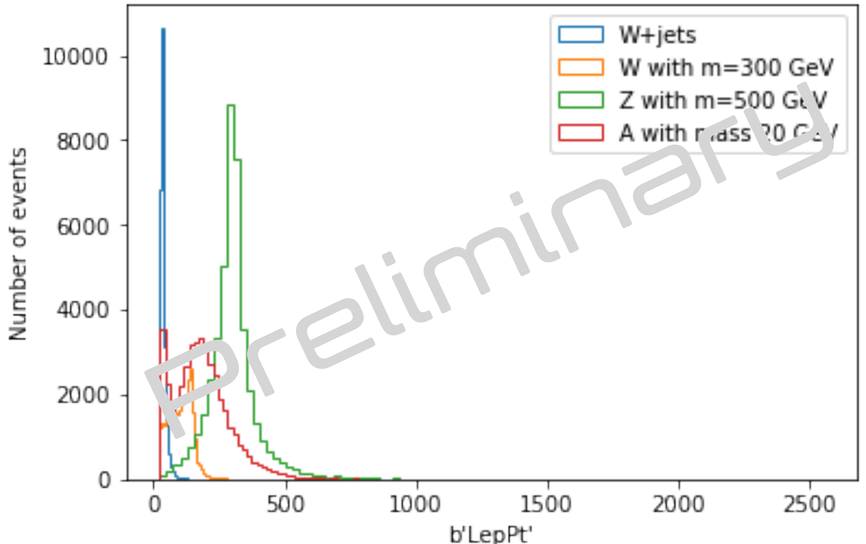
One would have to learn f and g to evaluate them at trigger. Online processing is replaced by offline training

Beyond the toy-model

- *Approach works in principle*
  - *Can identity easily 2 of the 3 models*
  - *With enough statistics, could see the third*
- *Might not work in absolute*
  - *encoder based on physics motivate quantities which are not model-agnostic*
  - *Use deep:learning: train on raw data directly.* ***To be done next***



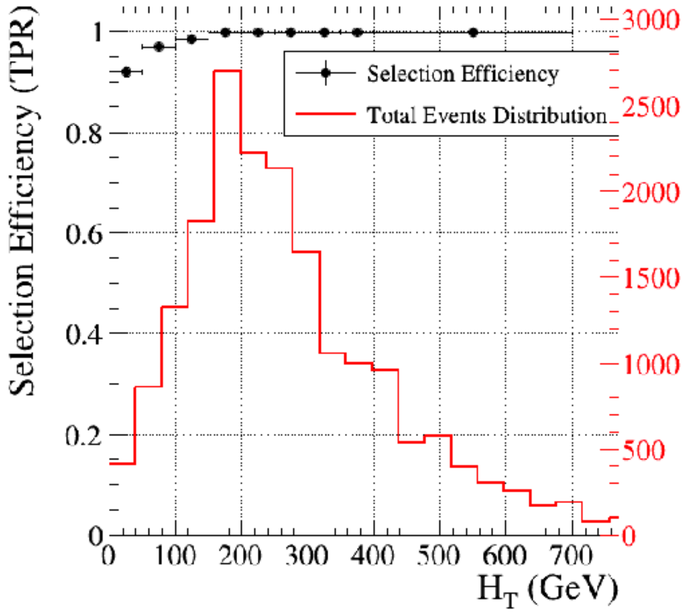


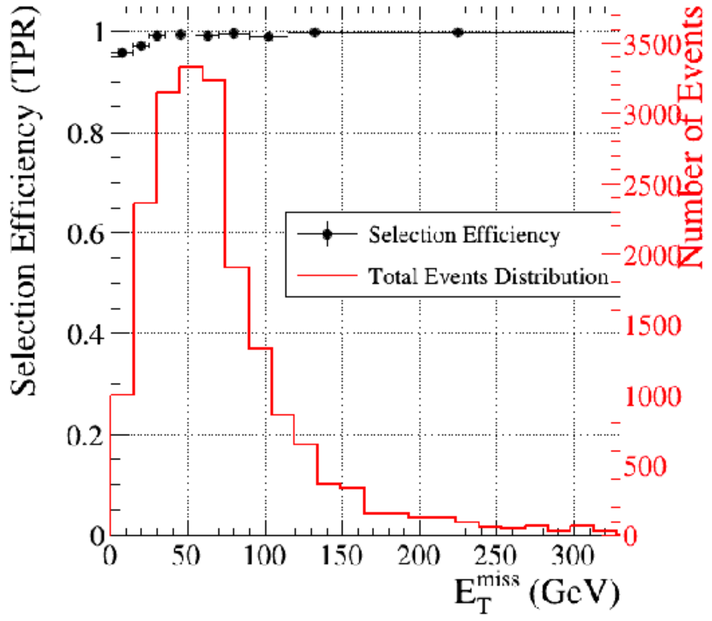


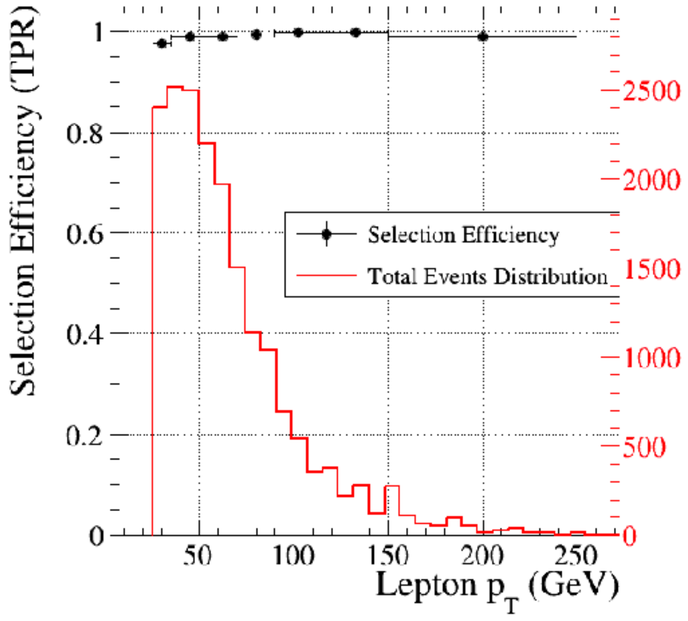
European Research Council

Kinematic Bias?

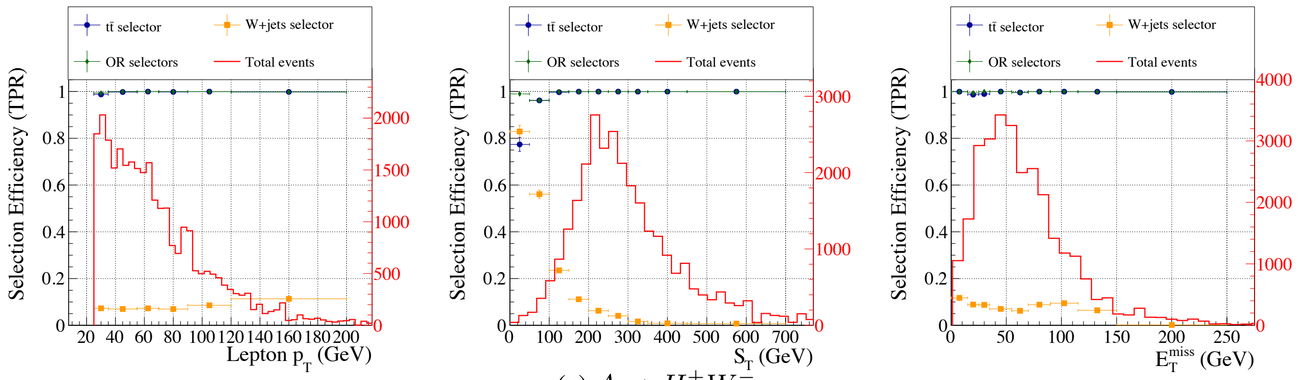
- *With 99% signal efficiency, bias on kinematic variables within the uncertainty of a trigger-efficiency measurement*



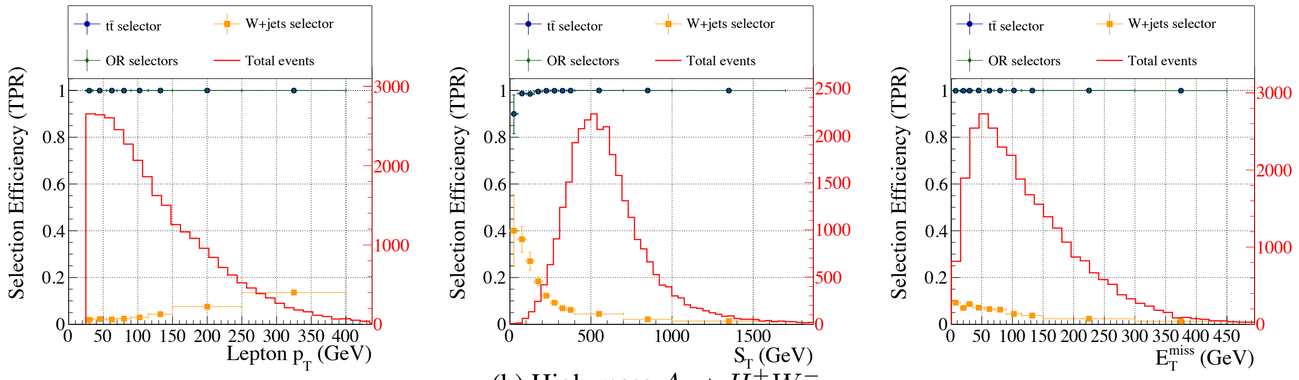




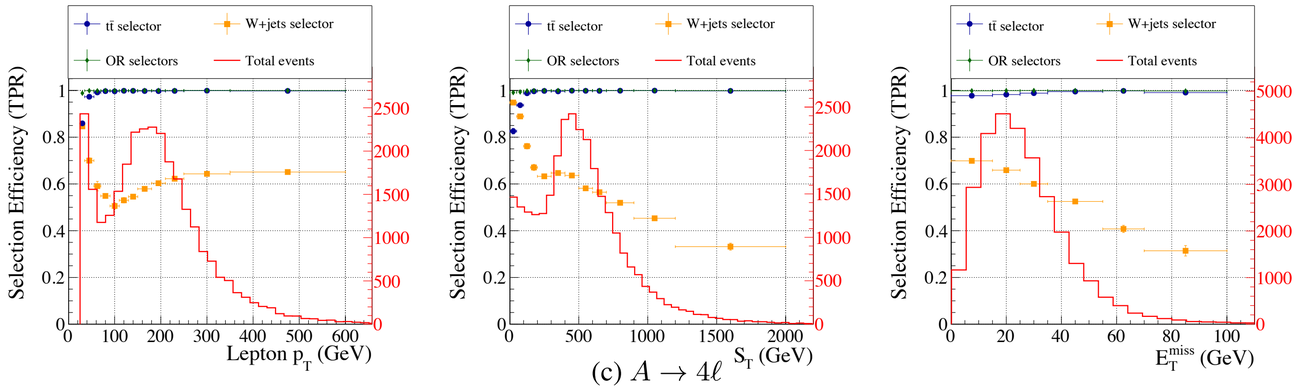
TOPCLASS: do we kill New Physics?



(a) $A \to H^+W^-$



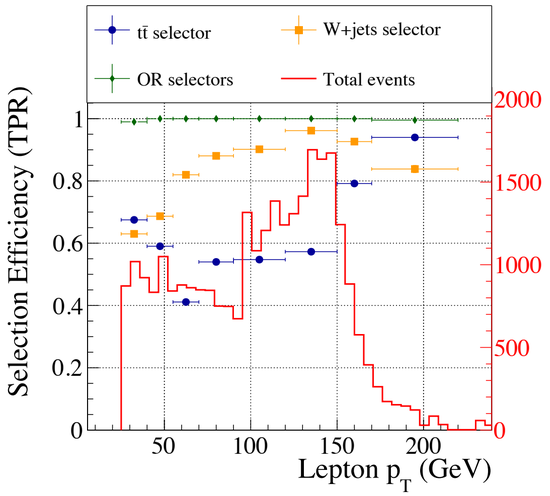
(b) High-mass $A \to H^+W^-$

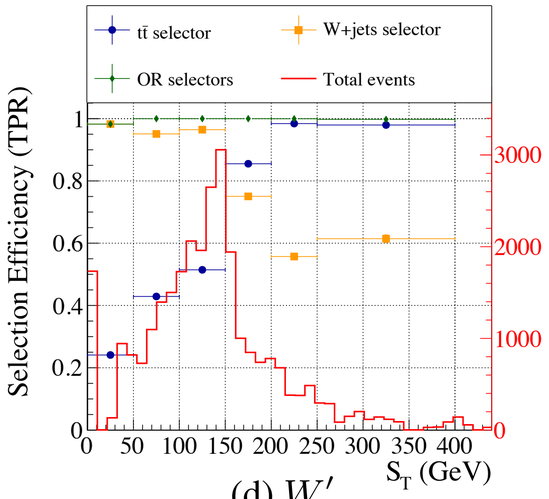


(c) $A \to 4\ell$

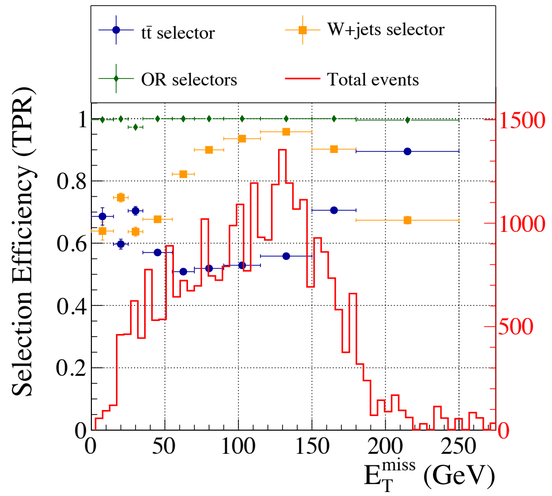
European Research Council

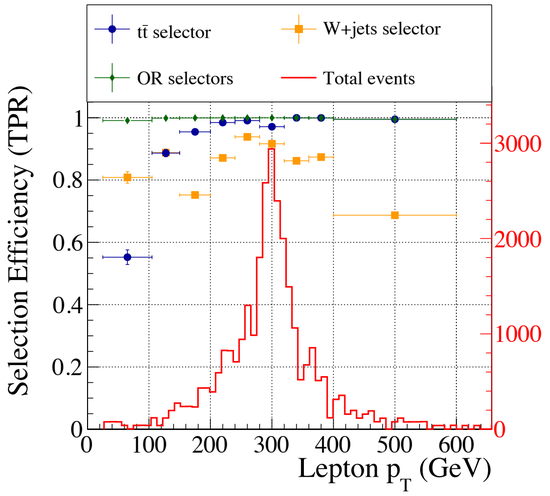
TOPCLASS: do we kill New Physics?

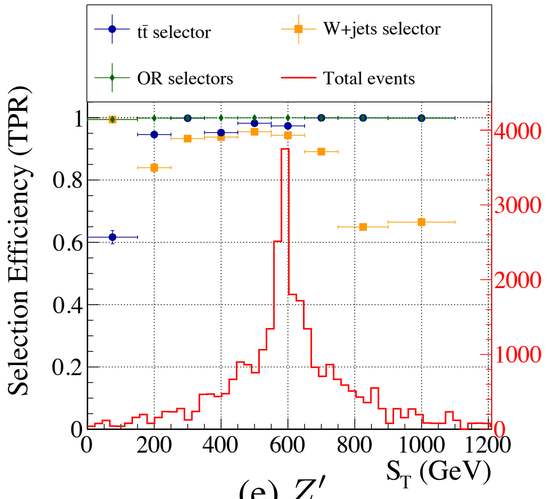




(d) W'







(e) Z'