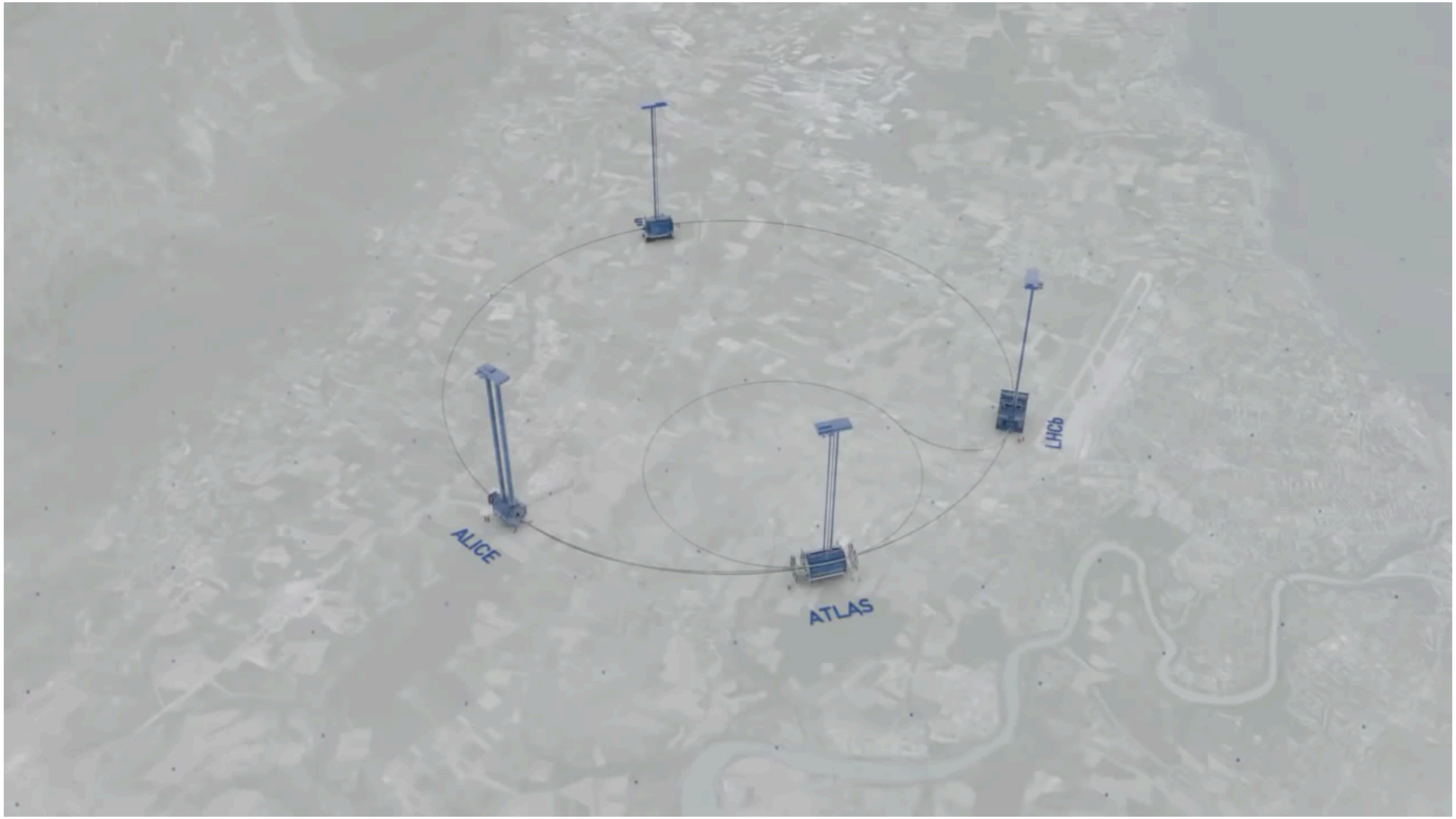
Deep Learning Applications for LHC experiments

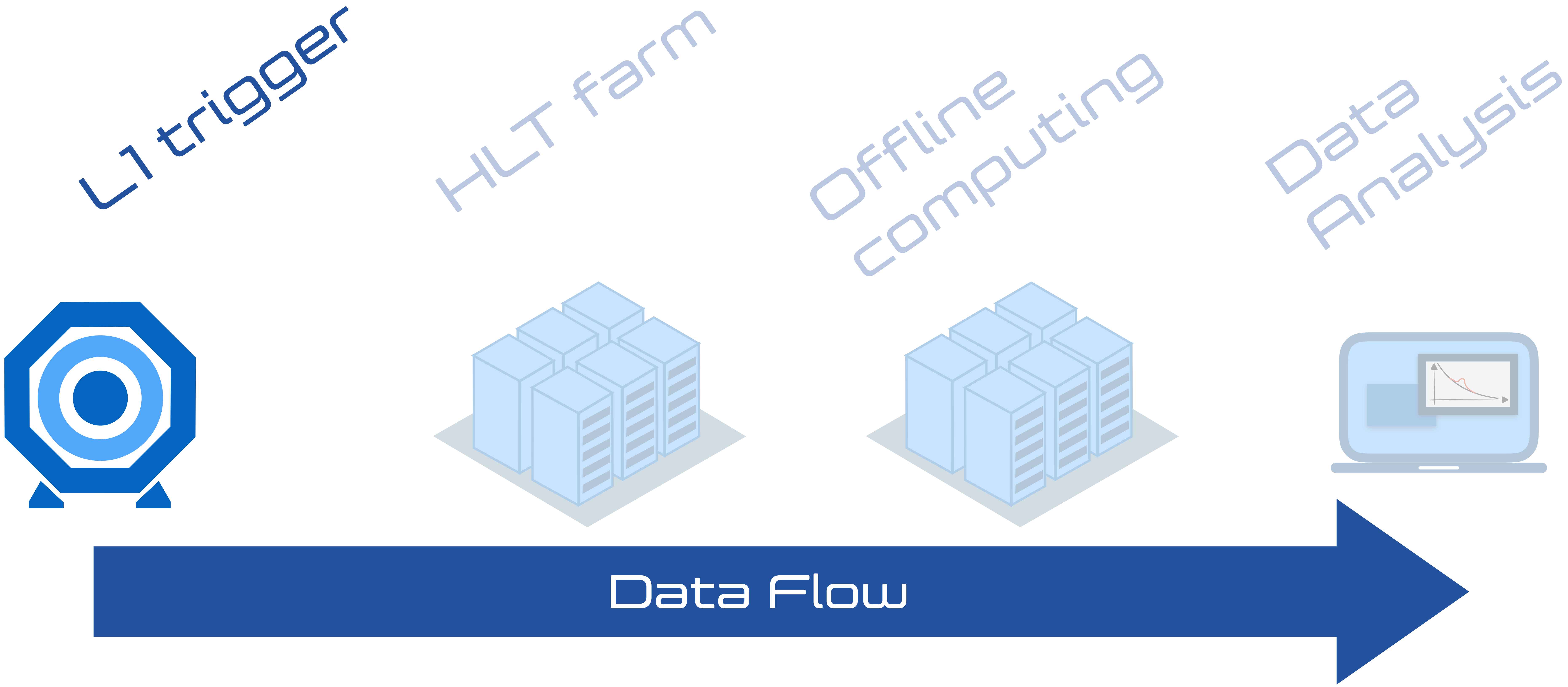
Maurizio Pierini

Oxford, 27/11/2018



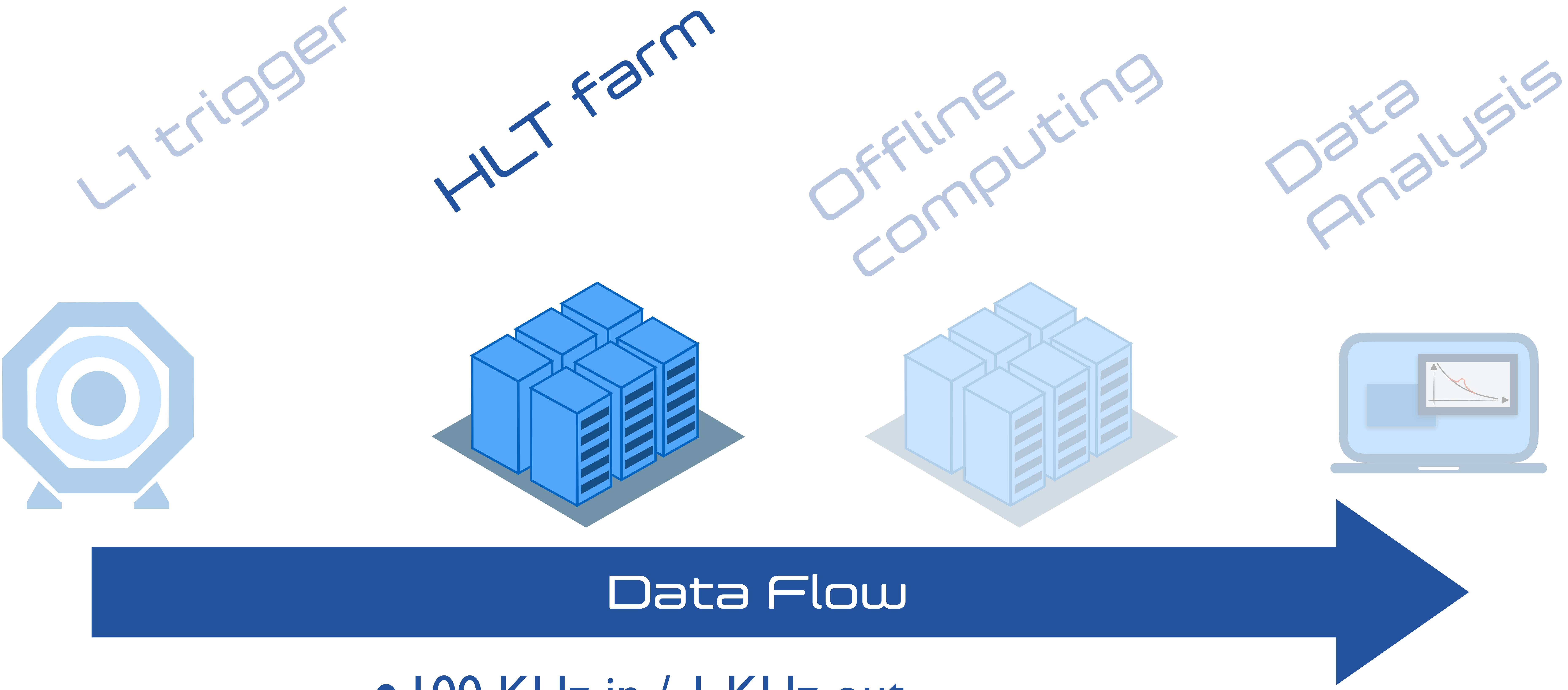
https://www.youtube.com/watch?v=jDC3-QSiLB4

The LHC Big Data problem



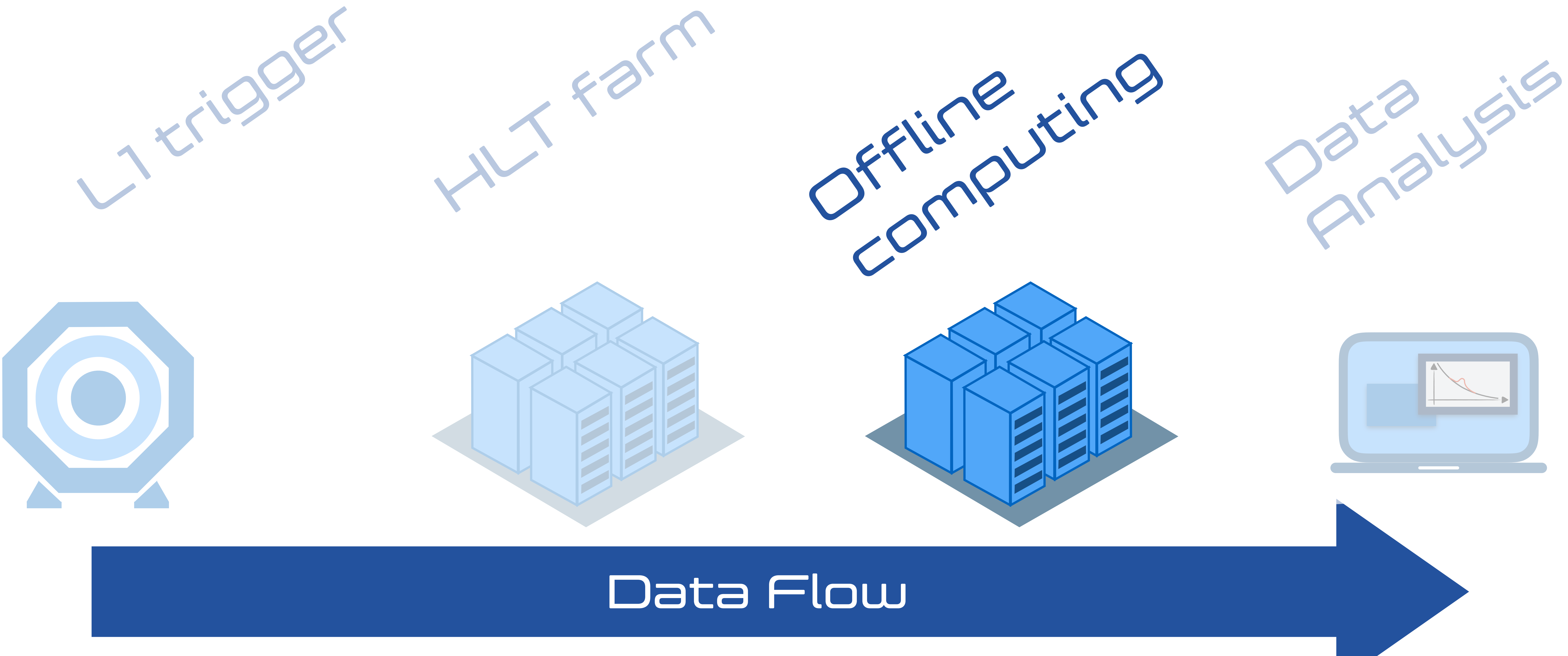
- 40 MHz in / 100 KHz out
- ~ 500 KB / event
- Processing time: ~10 µs
- Based on coarse local reconstructions
- FPGAs / Hardware implemented

The LHC Big Data problem



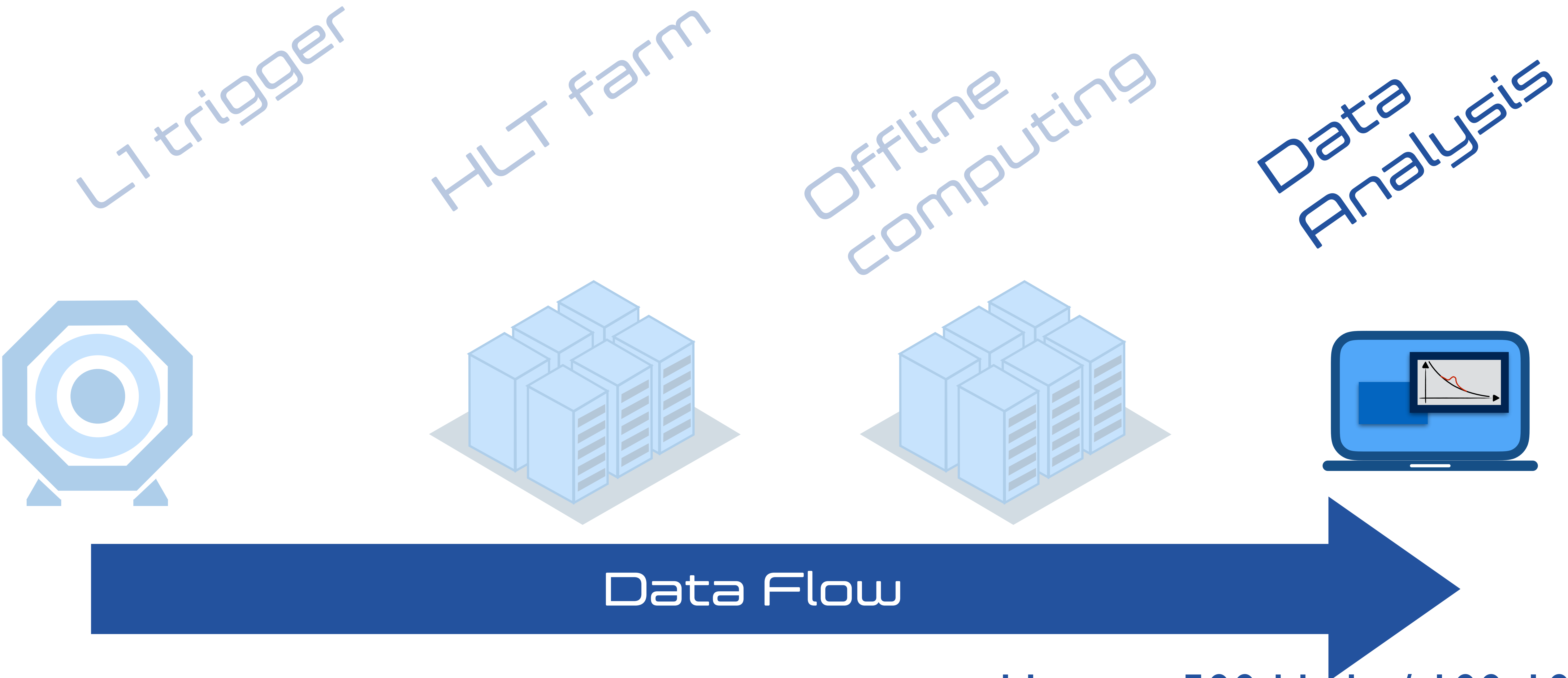
- 100 KHz in / 1 KHz out
- ~ 500 KB / event
- Processing time: ~100 ms
- Based on simplified global reconstructions
- Software implemented on CPUs

The LHC Big Data problem



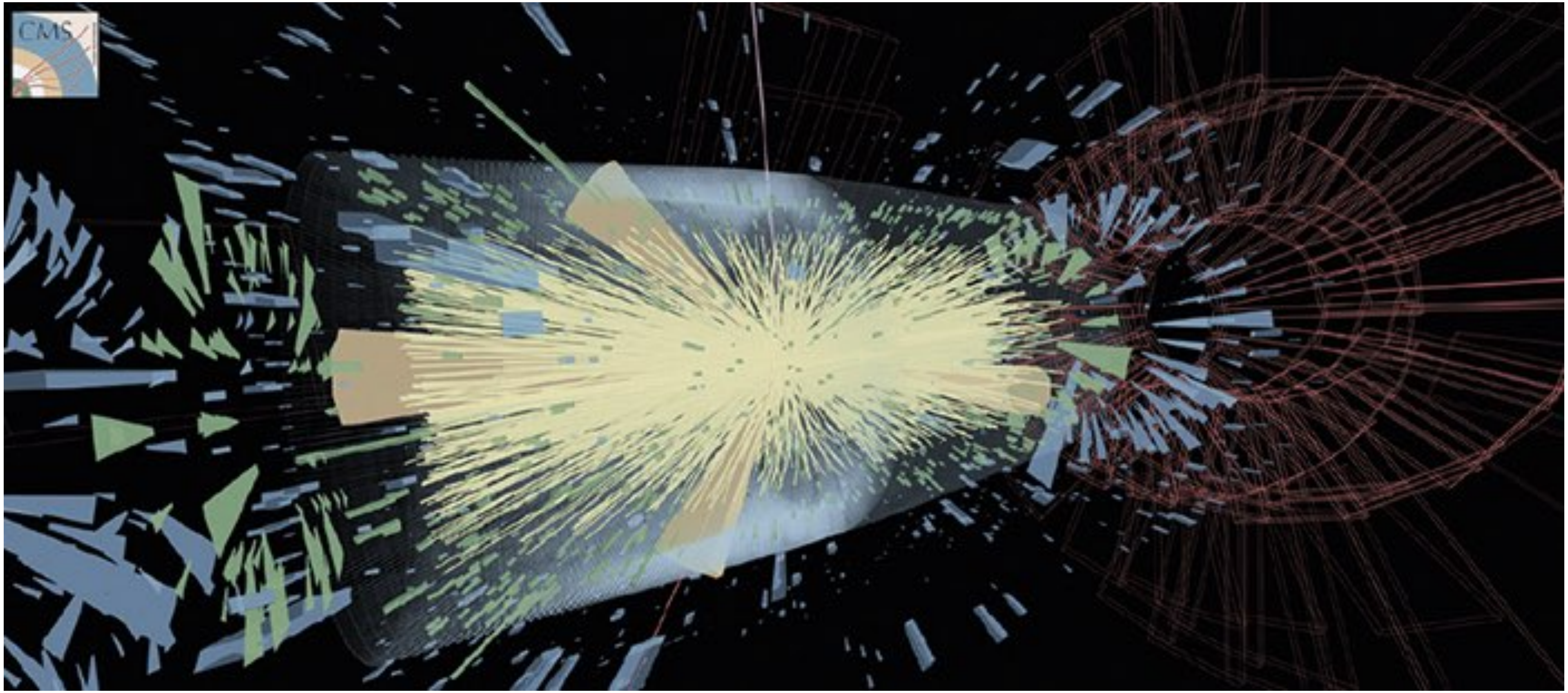
- 1 KHz in / 1.2 kHz out
- ~ 1 MB / 200 kB / 30 kB per event
- Processing time: ~20 s
- Based on accurate global reconstructions
- Software implemented on CPUs

The LHC Big Data problem



- Up to ~ 500 Hz In / 100-1000 events out

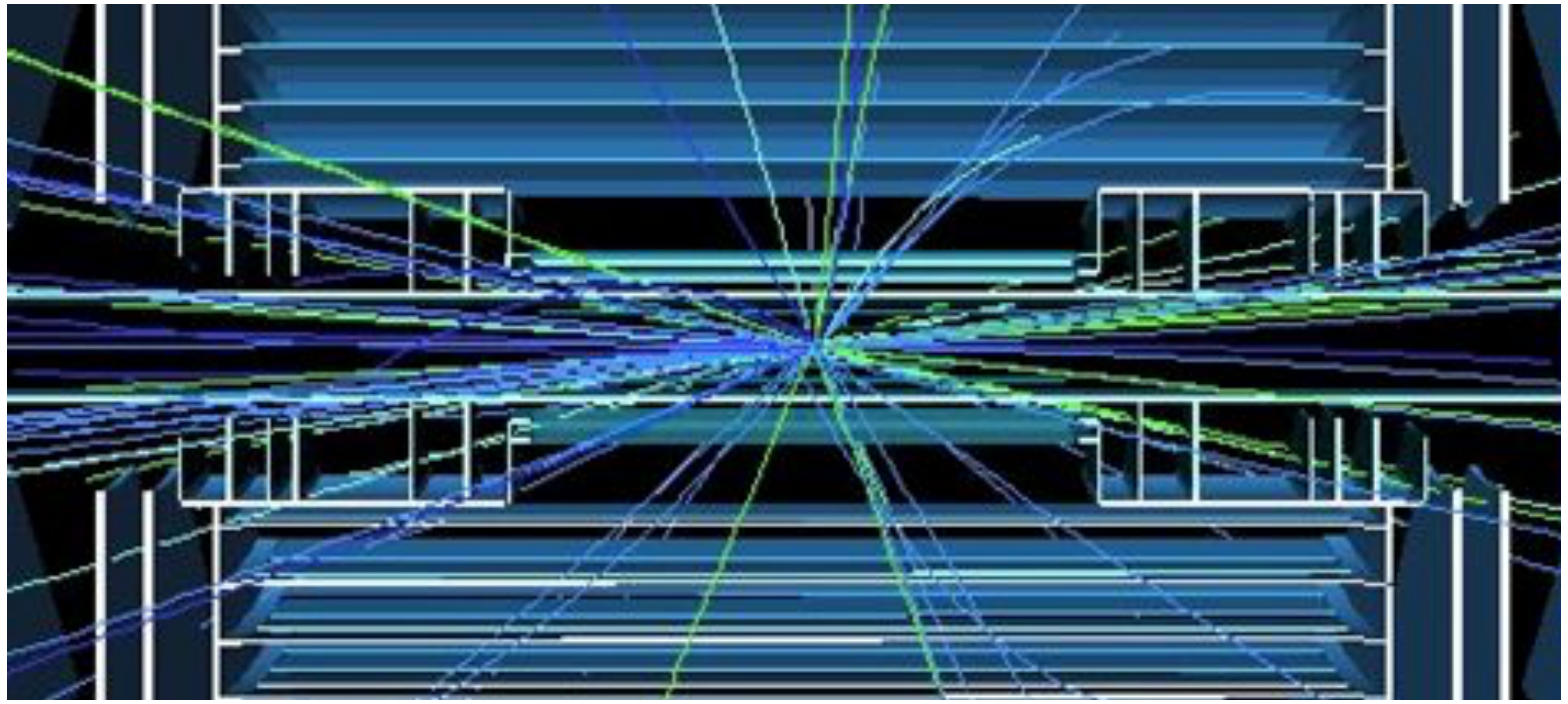
- <30 KB per event
- Processing time irrelevant
- User-written code + centrally produced selection algorithms



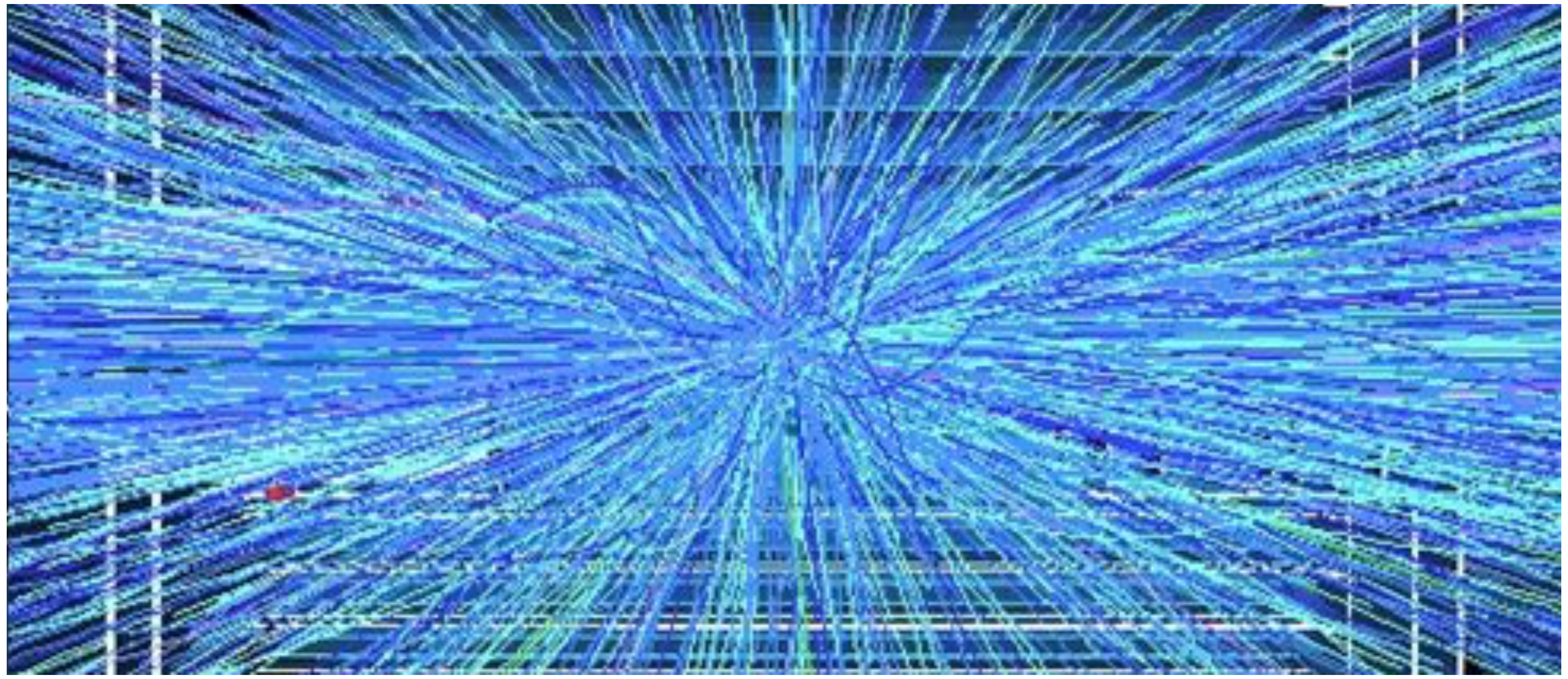
LHC future and its big challenge

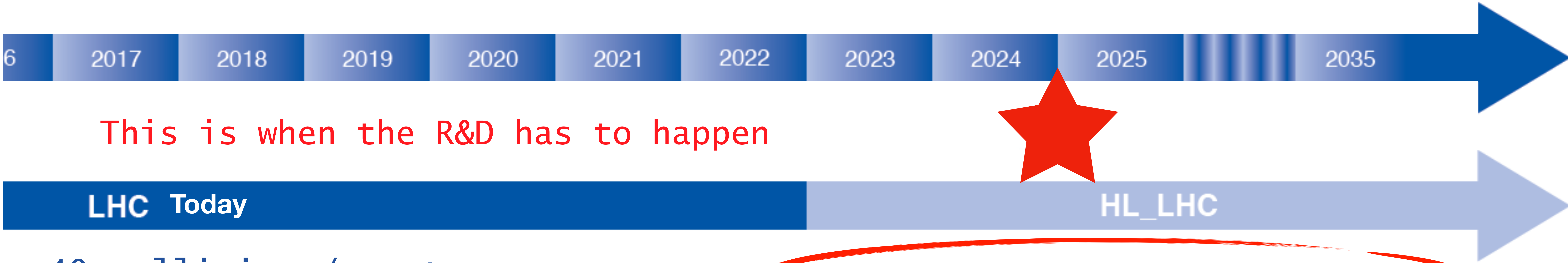
HL-LHC: elephant in the room

5 interactions/beam cross



400 interactions/beam cross





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

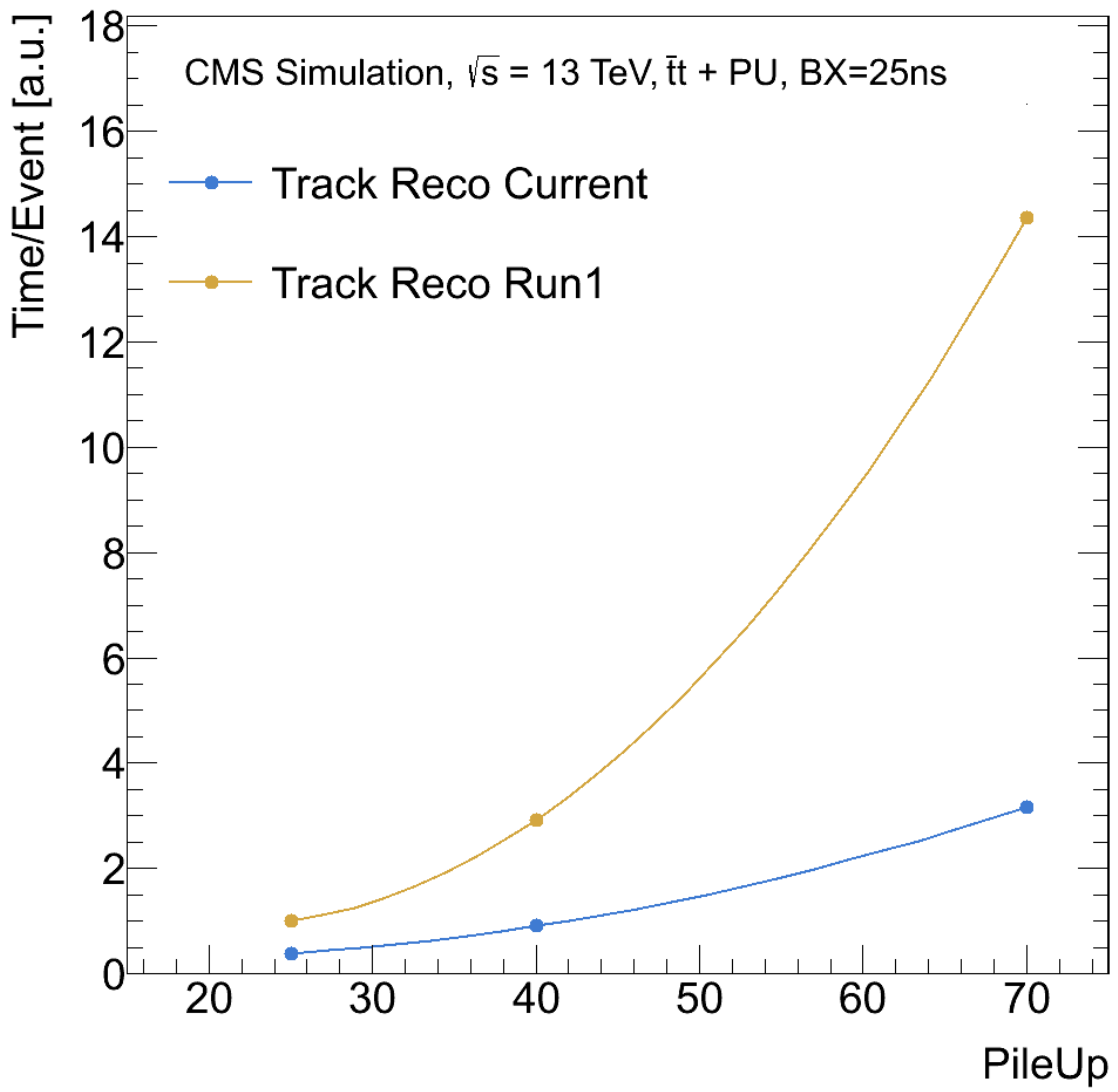
- ~200 collisions/event
- ~minute/event processing time
- (at best)Same computing resources as today

HL-LHC: elephant in the room

- *Flat budget vs. more needs = **<u>current rule-based reconstruction algorithms will not be sustainable</u>***

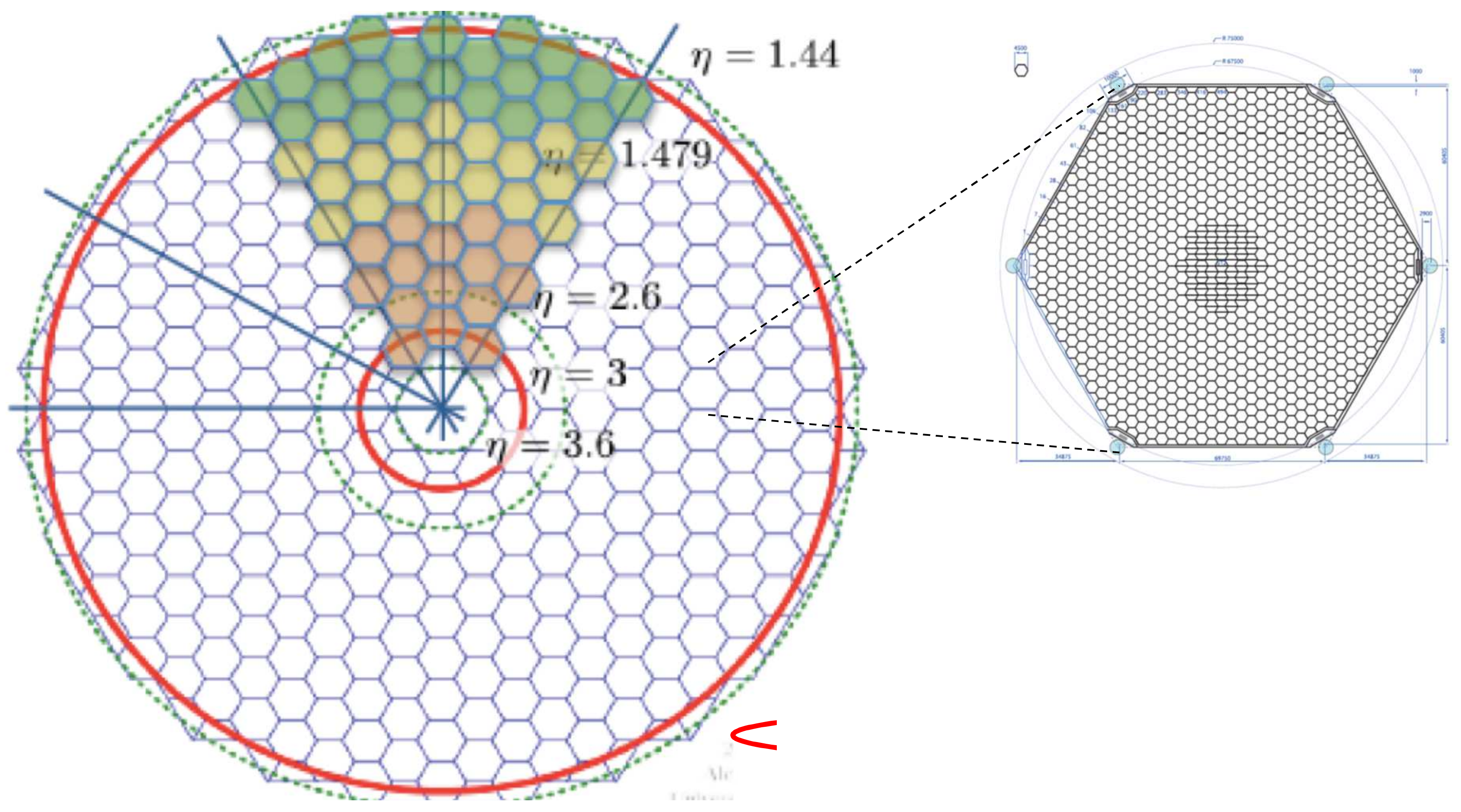
- ***<u>Adopted solution:</u>*** *more granular and complex detectors → more computing resources needed → more problems*

- ***Modern Machine Learning might be the way out***

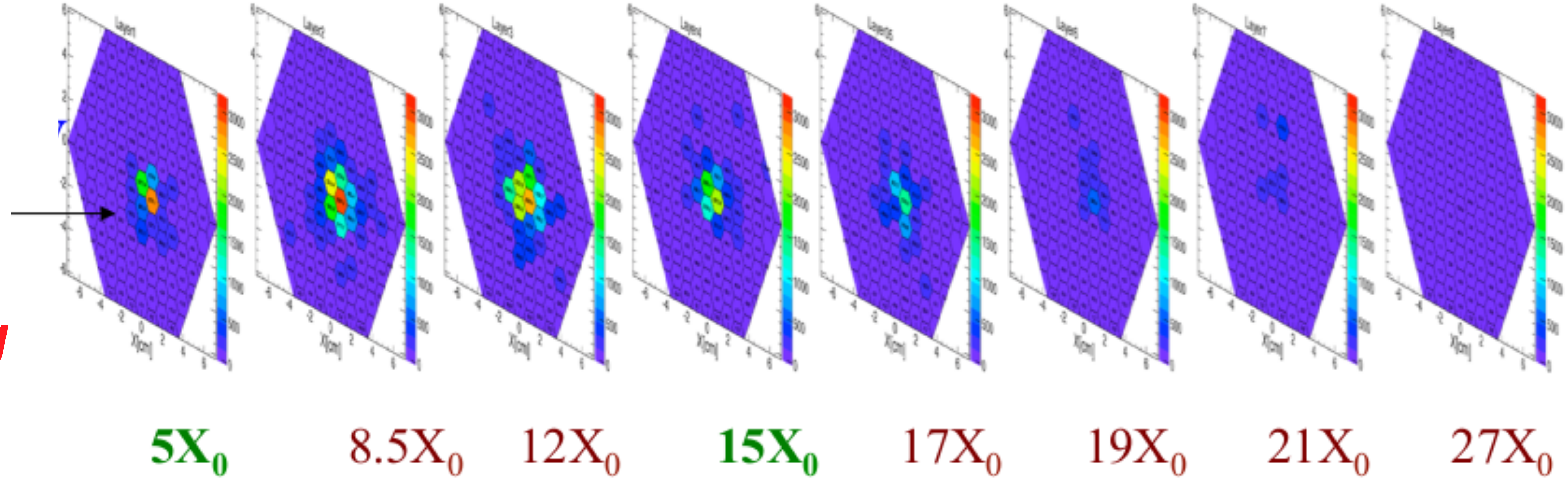


HL-LHC: elephant in the room

- 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***



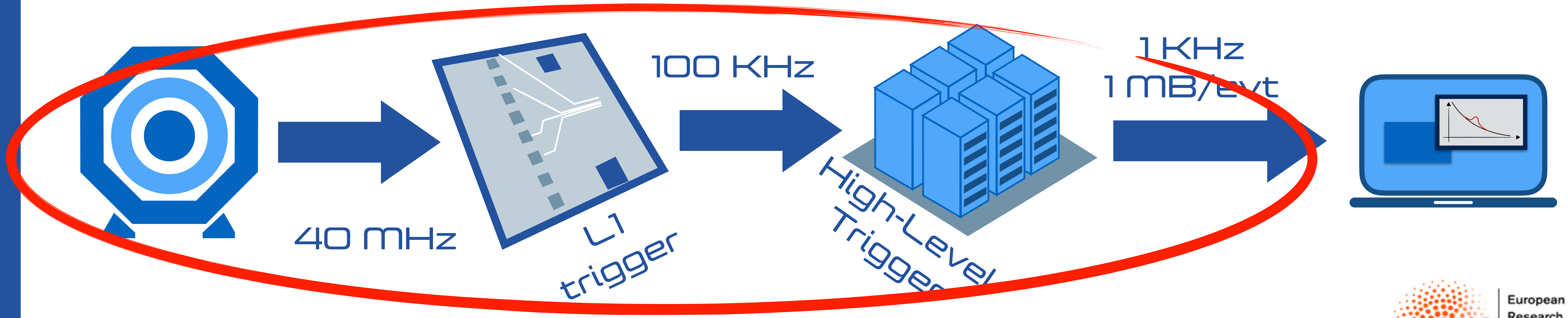
250 GeV electron passing through 8 layers (27 X_0)



Deep Learning and LHC Big Data

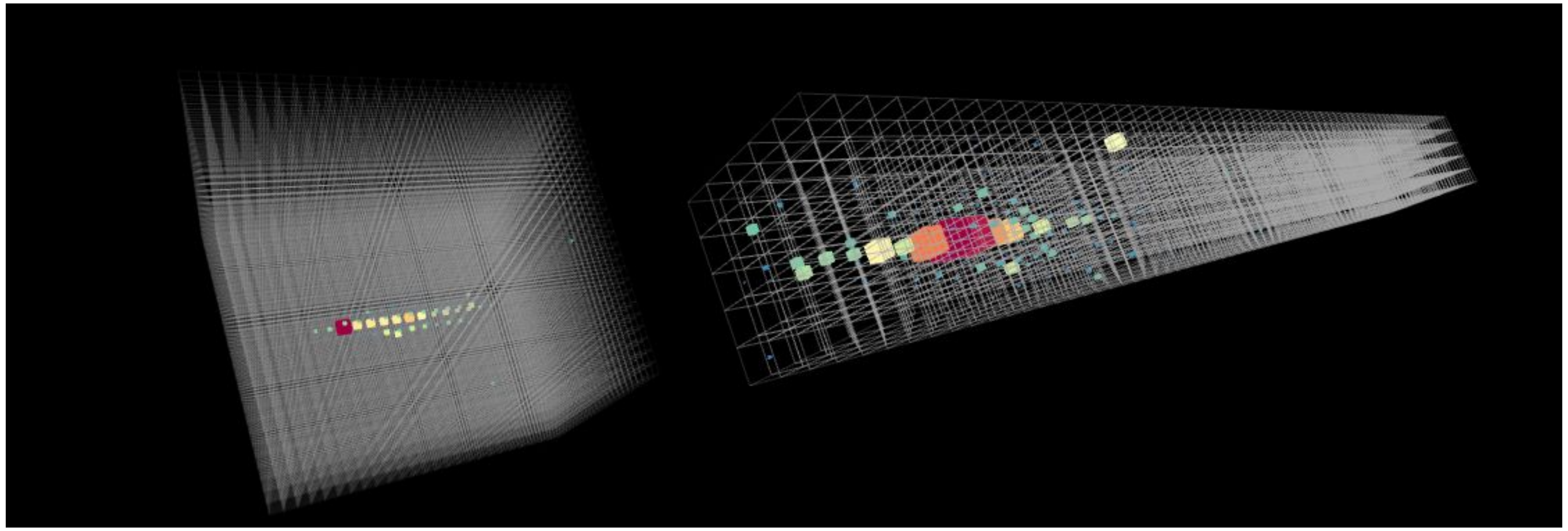
- *Possible solution to the HL-LHC problem: Deep Learning to be <u>faster</u> and <u>better</u> in what we do today, freeing resources for new ideas*

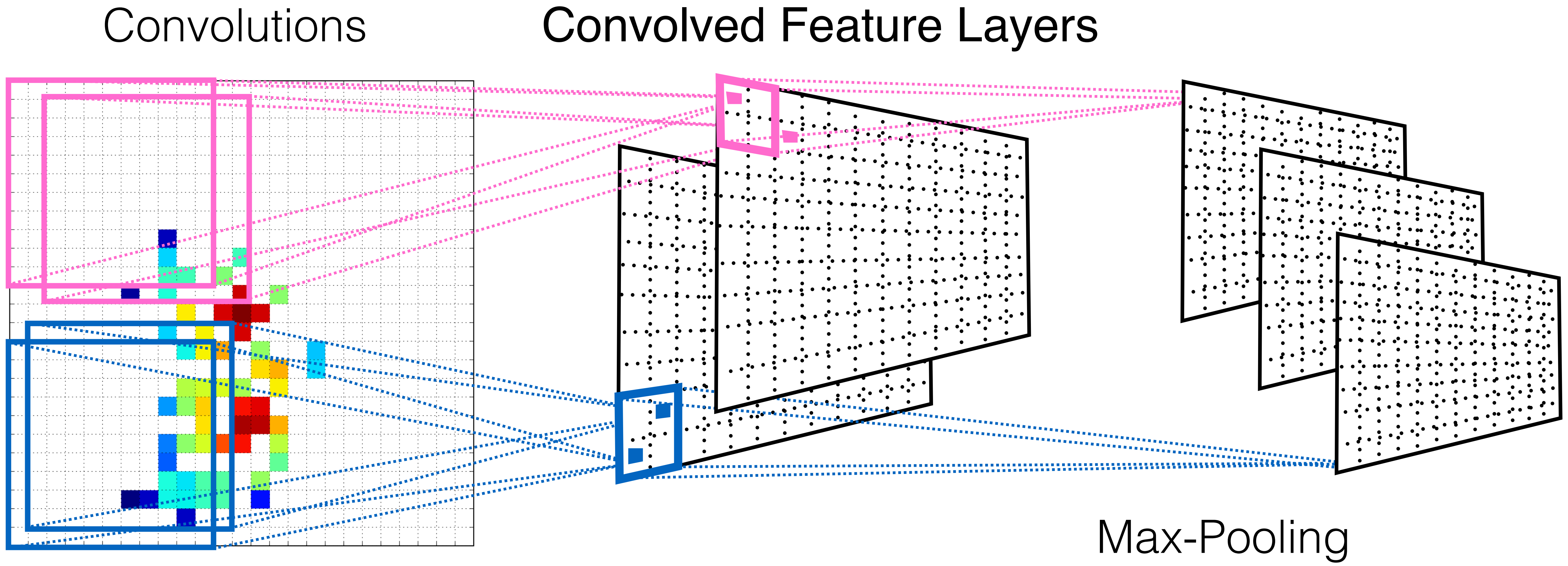
- *DL deployment needs to happen in between collisions and data analysis (trigger, reconstruction, …), where freeing resources will make a difference*



Particle reconstruction as image detection

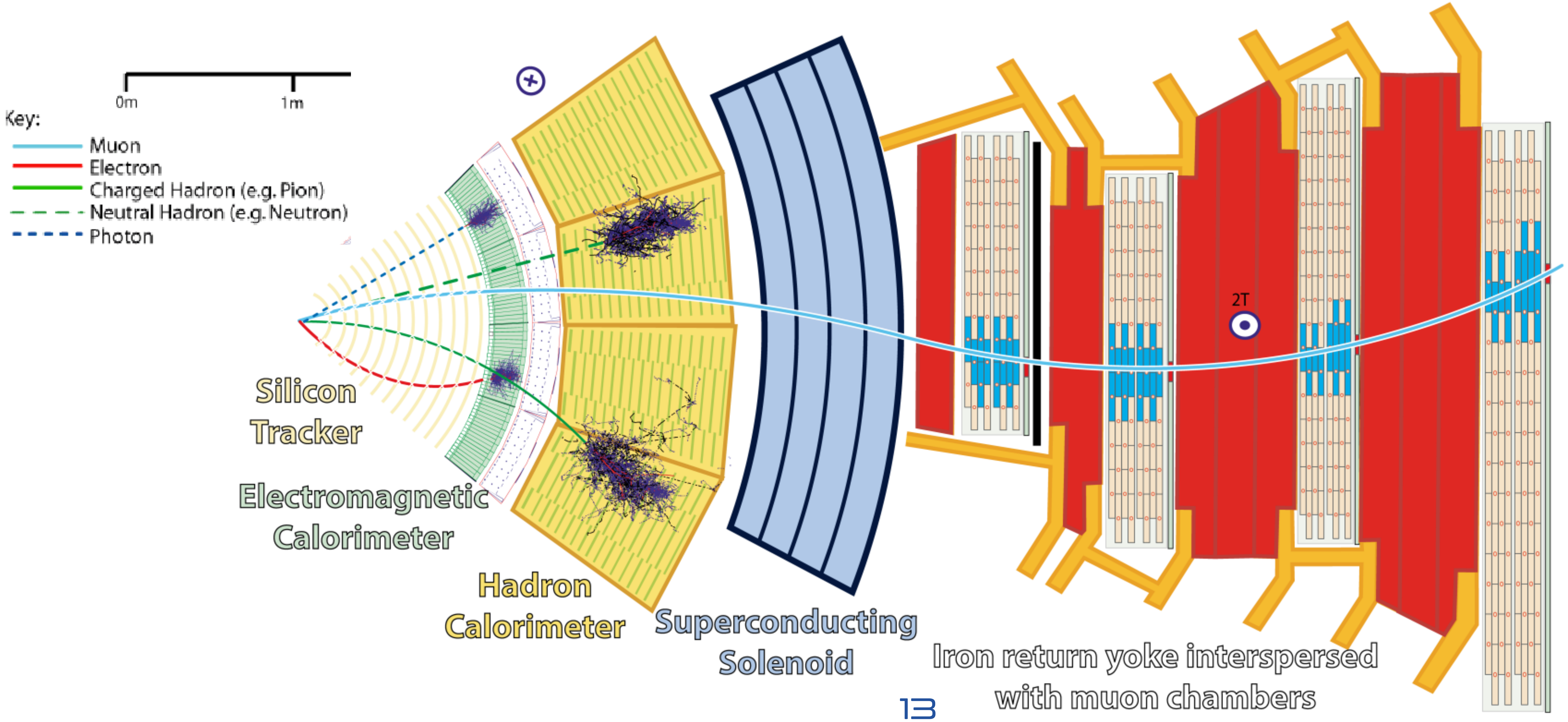
- Future detectors will be 3D arrays of sensors with regular geometry
- Ideal configuration to apply Convolutional Neural Network
  - speed up reconstruction at similar performances
  - and possibly improve performances





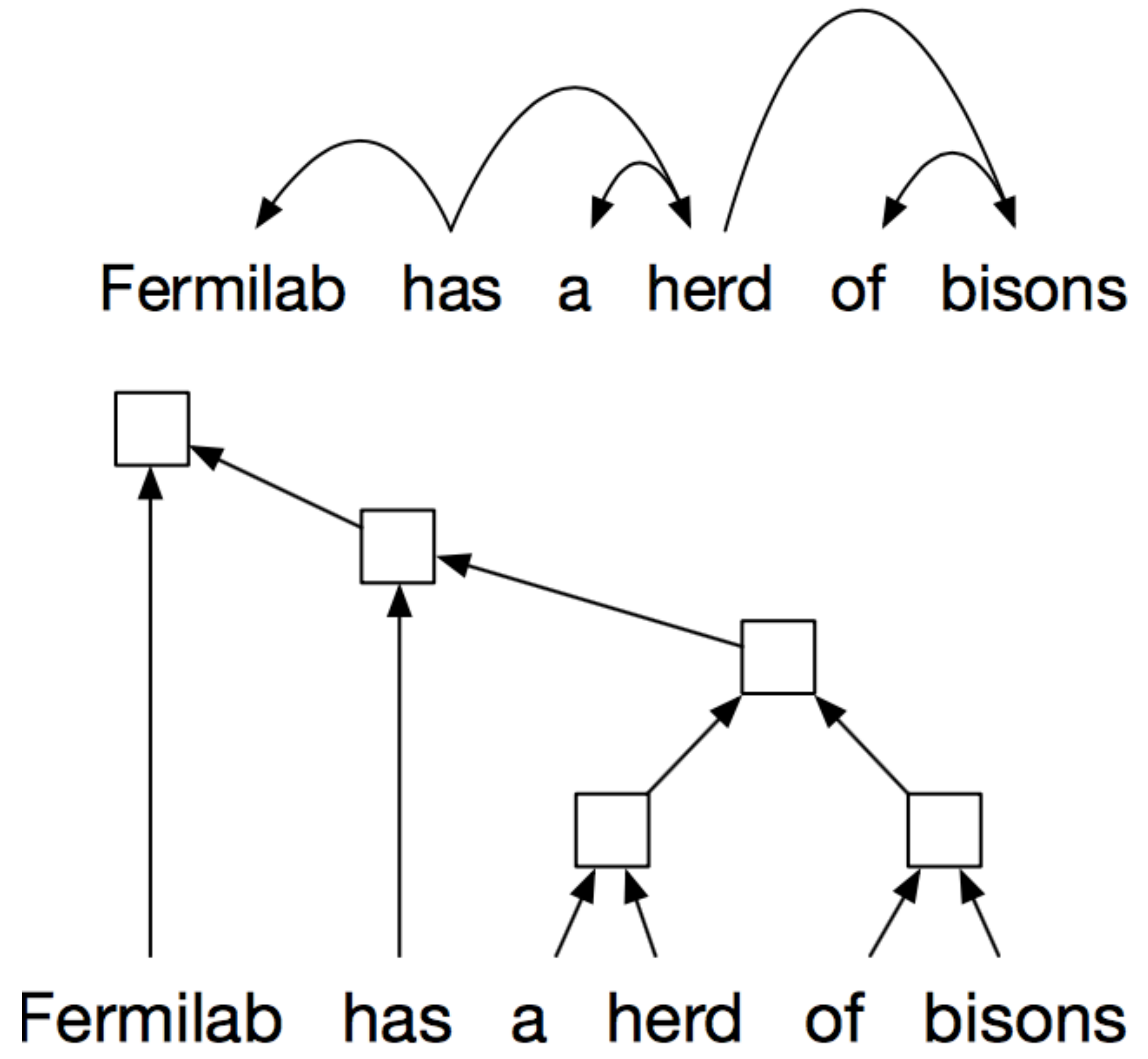
Particle Flow

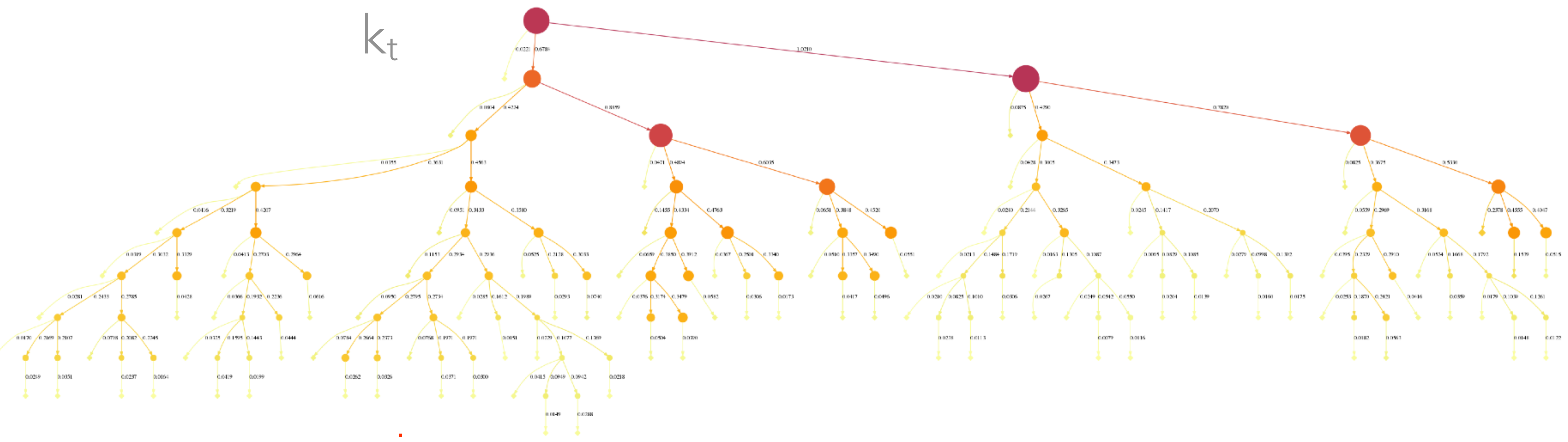
- CMS uses PF to combine sub-detector information and produce a list of reconstructed particles
- Anything (jets, MET, resonances, etc) is reconstructed from these particles
- One could generalise the VAE new-physics-detection algorithm and make it PF compliant
  - integrated in the reconstruction flow @HLT
  - can abstract from model dependence inherited by any physics-motivated HLF choice



LHC events & language processing

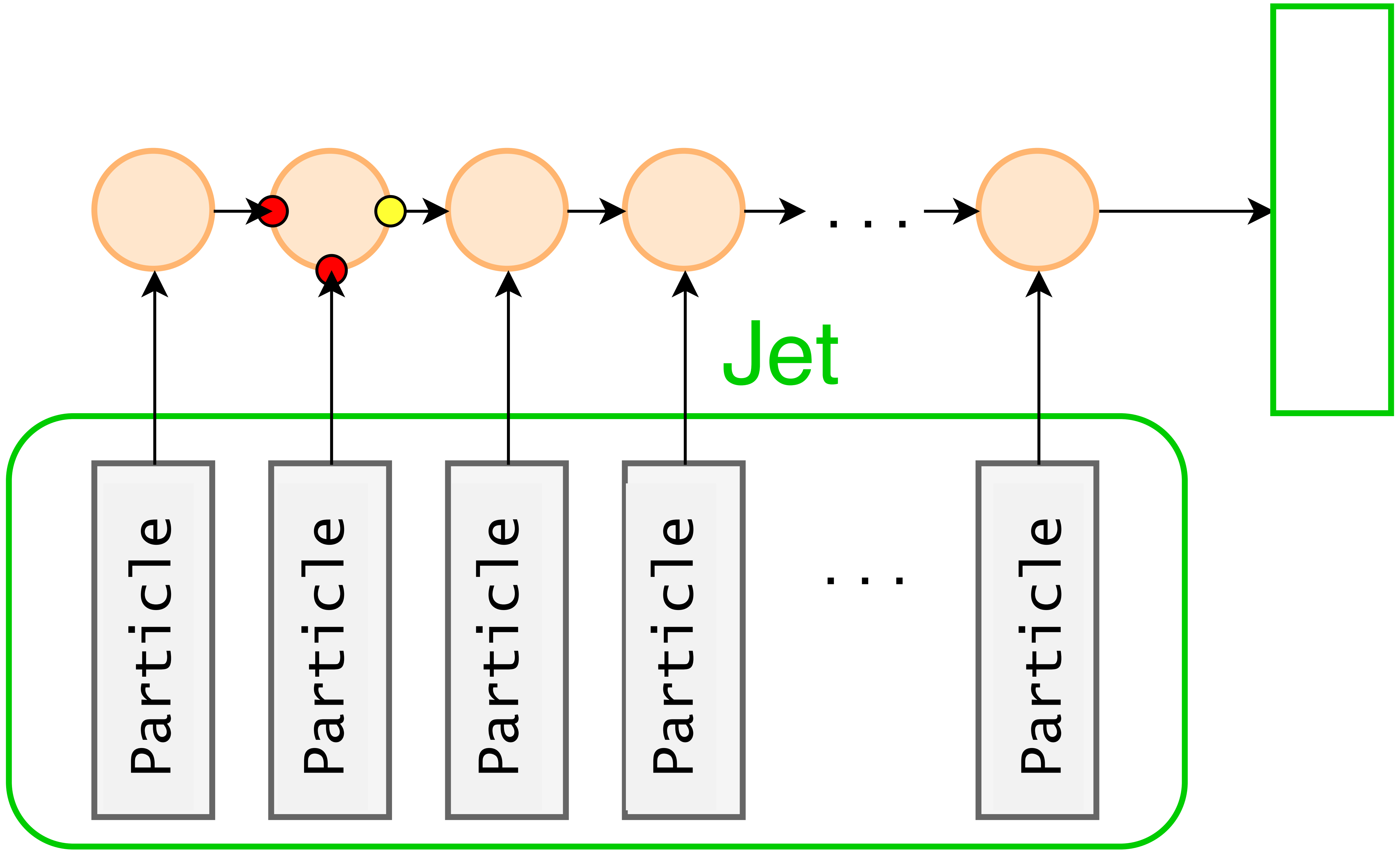
- *PF reco is not the best match for computing vision techniques (e.g., convolutional neural networks) don't work*
  - *one would have to convert the particles to a pixelated images, loosing resolution*
- *Instead, list of particles can be processed by Deep Learning architectures designed for natural language processing (RNN, LSTMs, GRUs, …)*
  - *particles as words in a sentence*
  - *QCD is the grammar*





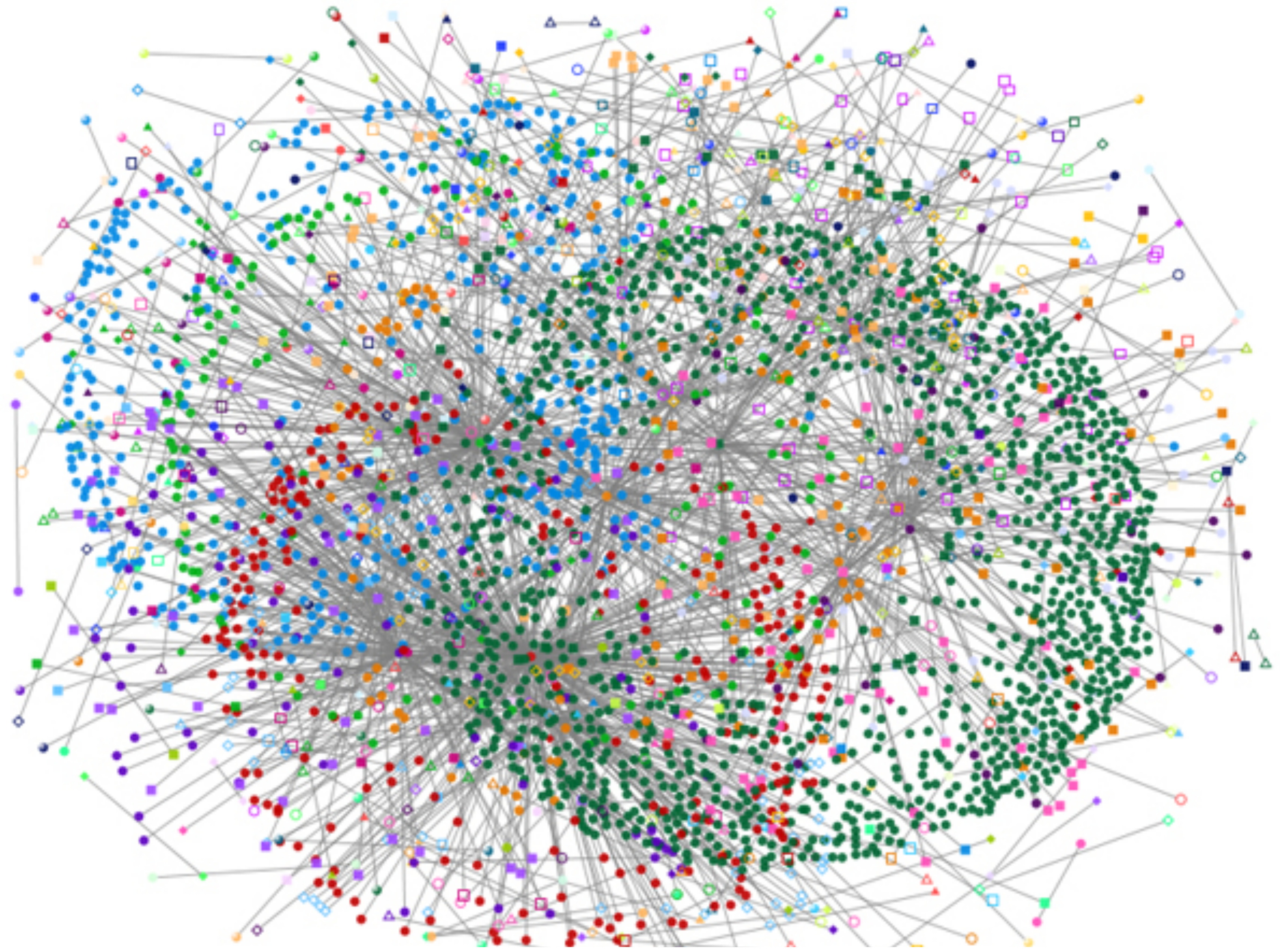
Recurrent Neural Networks

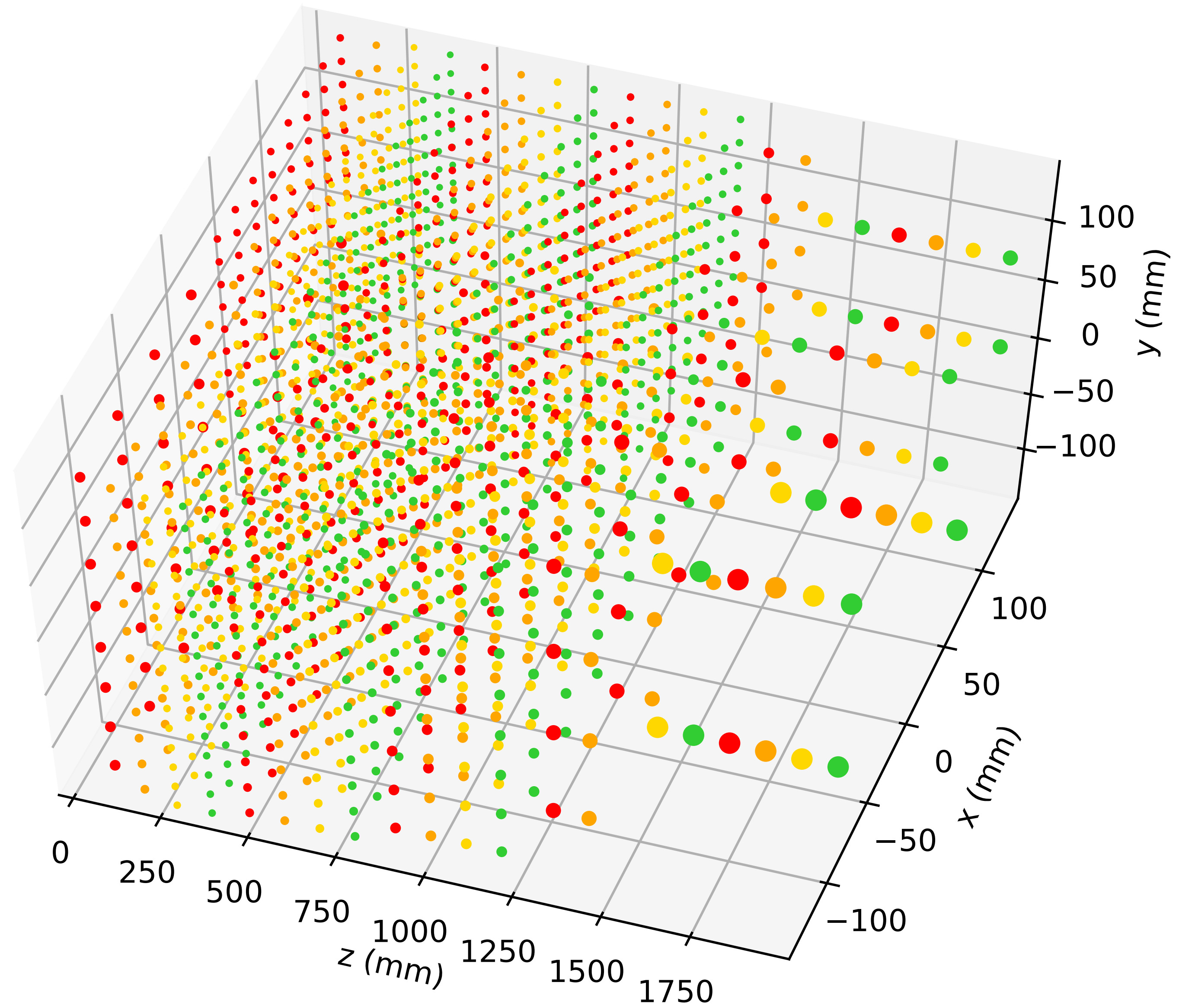
- *A network architecture suitable to process an ordered sequence of inputs*
  - *words in text processing*
  - *a time series*
  - *particles in a list*
- *Could be used for a single jet or the full event*
- *Next step: graph networks (active research direction)*



Graph Networks for real detectors

- CNNs assume that our detectors are regular arrays of sensors
- Our detectors are not
  - different components with different technologies
  - some particle visible only to some part of the detector
- CNNs don't really fit the sparsity of the collision data
- Instead, we think graph networks can work better

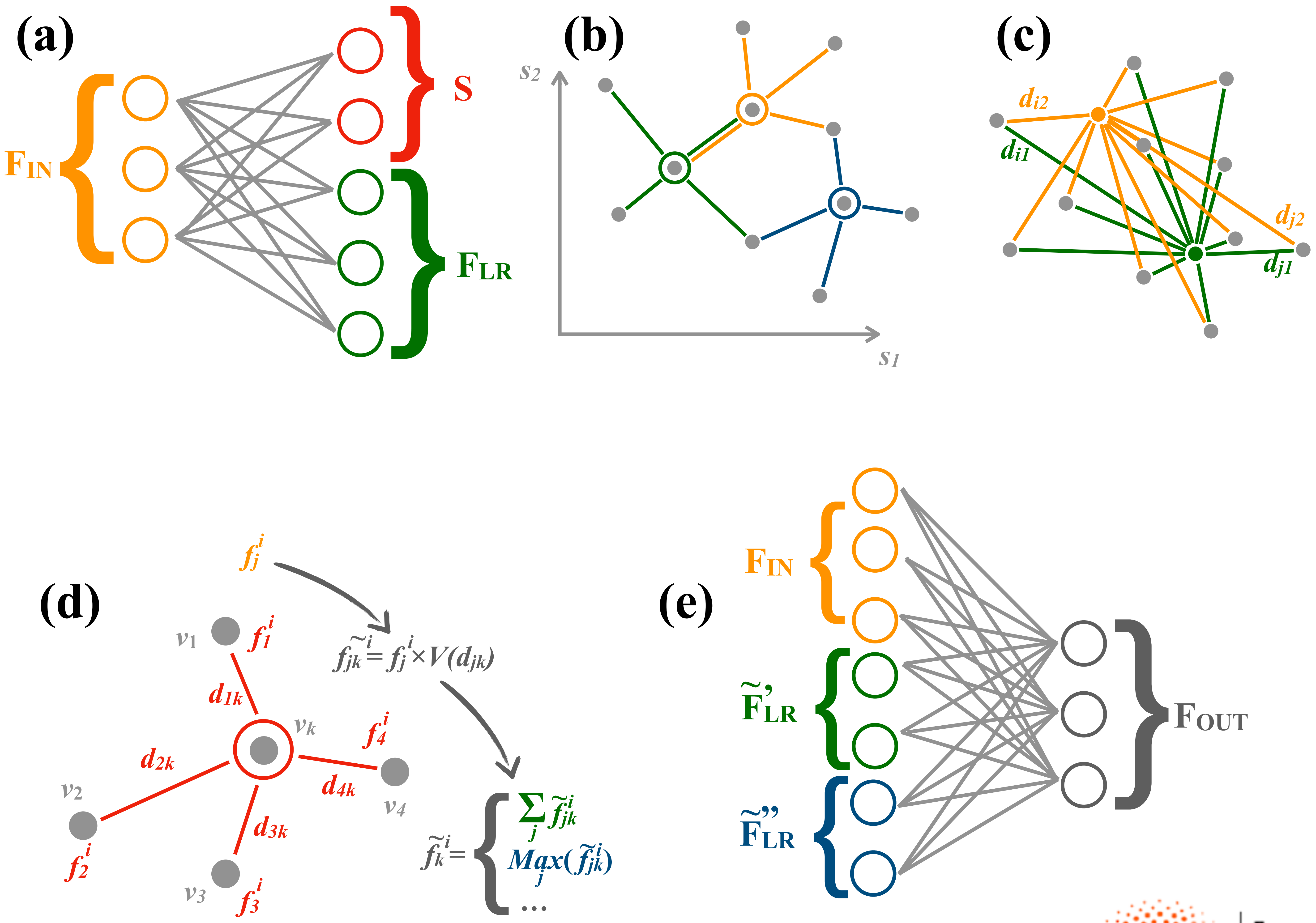




Learning representation

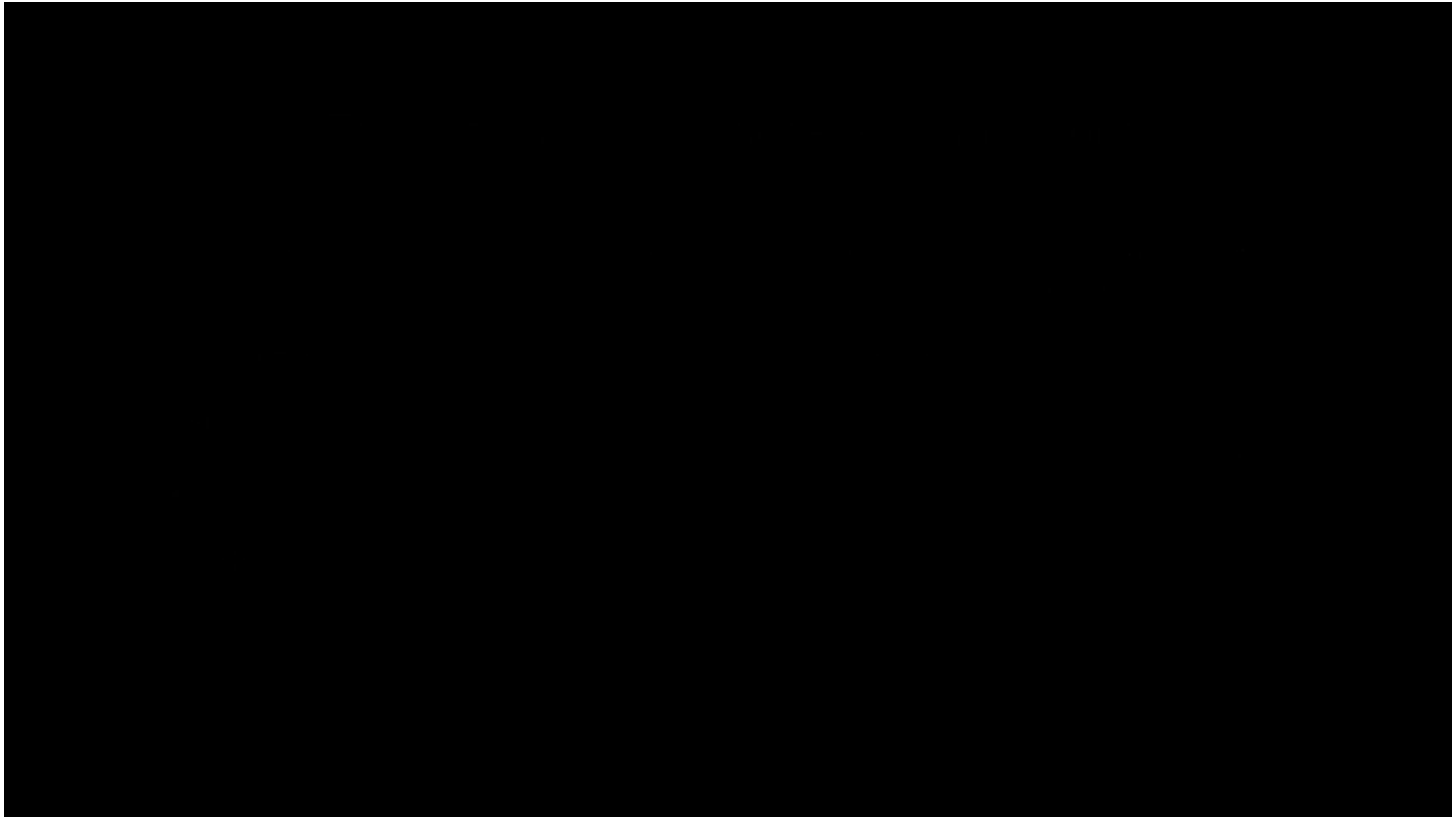
- What a graph network does is projecting a set of point into some other space where the concept of nearby (related to affinity) is learned

- Such a concept allows to abstract from the geometry of a detector and design experiment-independent architectures

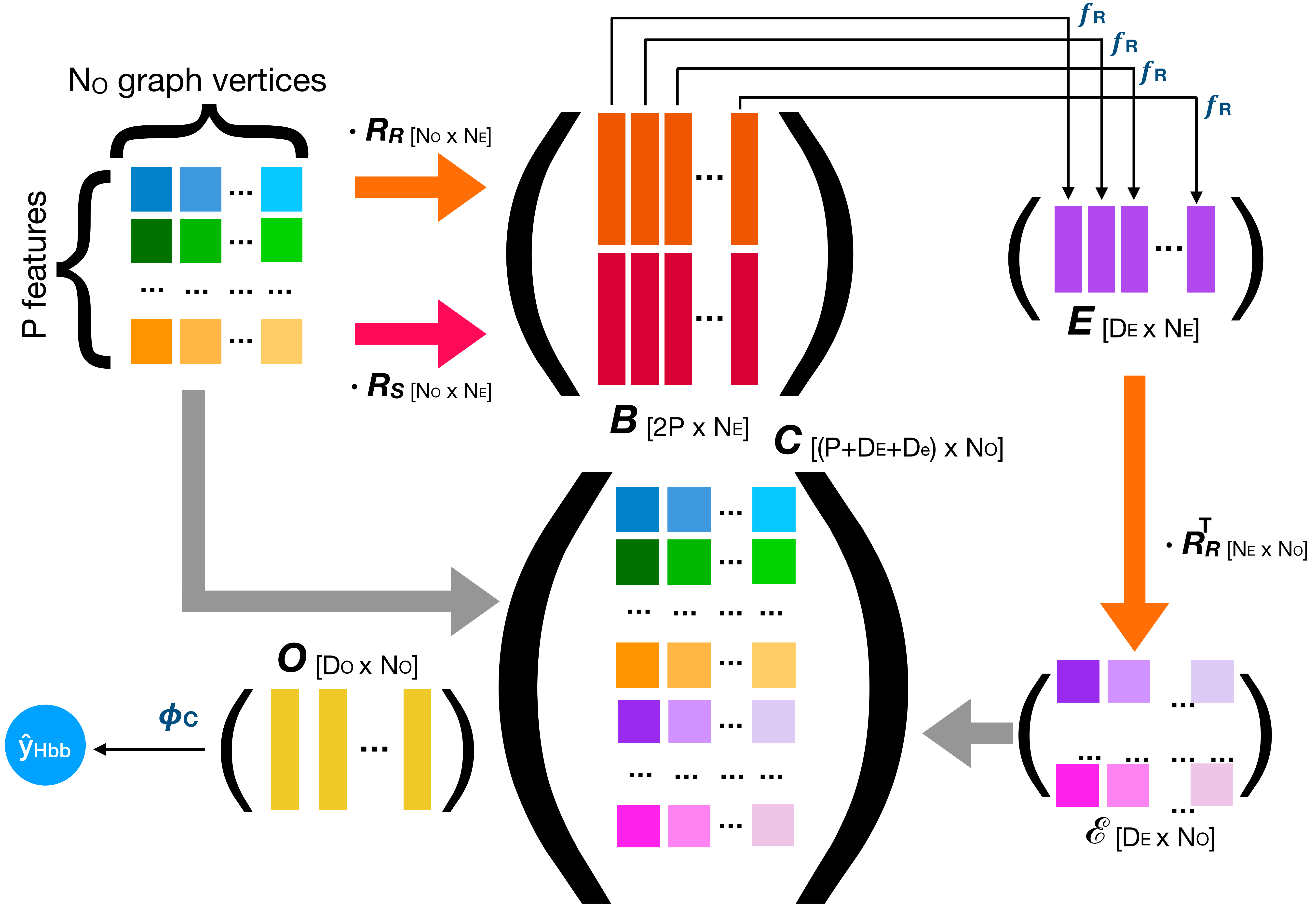


https://arxiv.org/abs/1902.07987

Interaction Networks

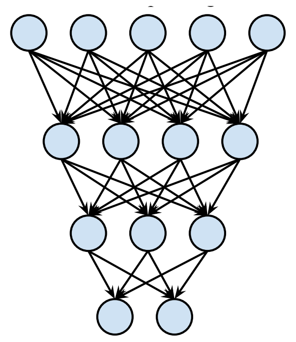


Interaction Networks

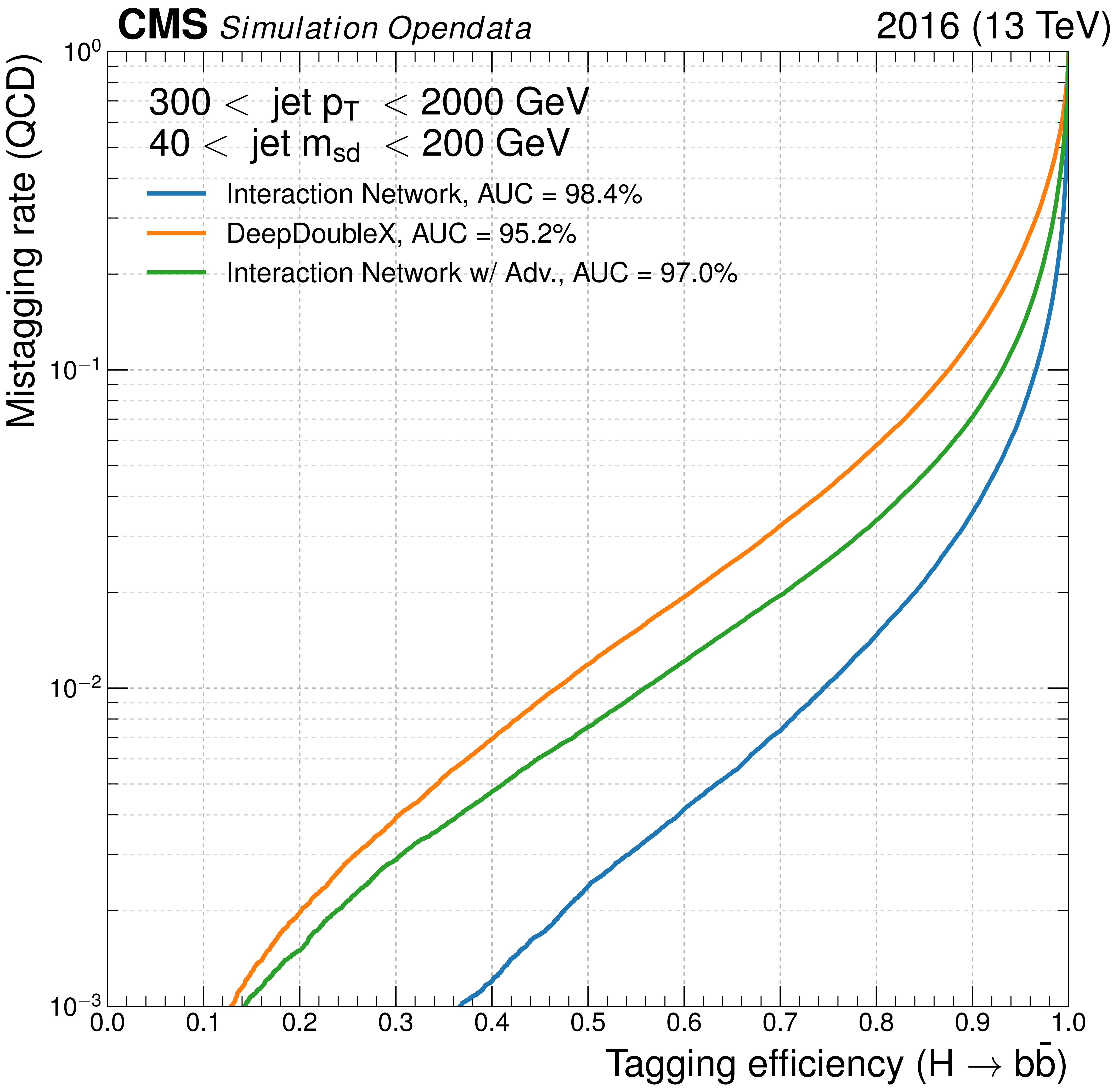


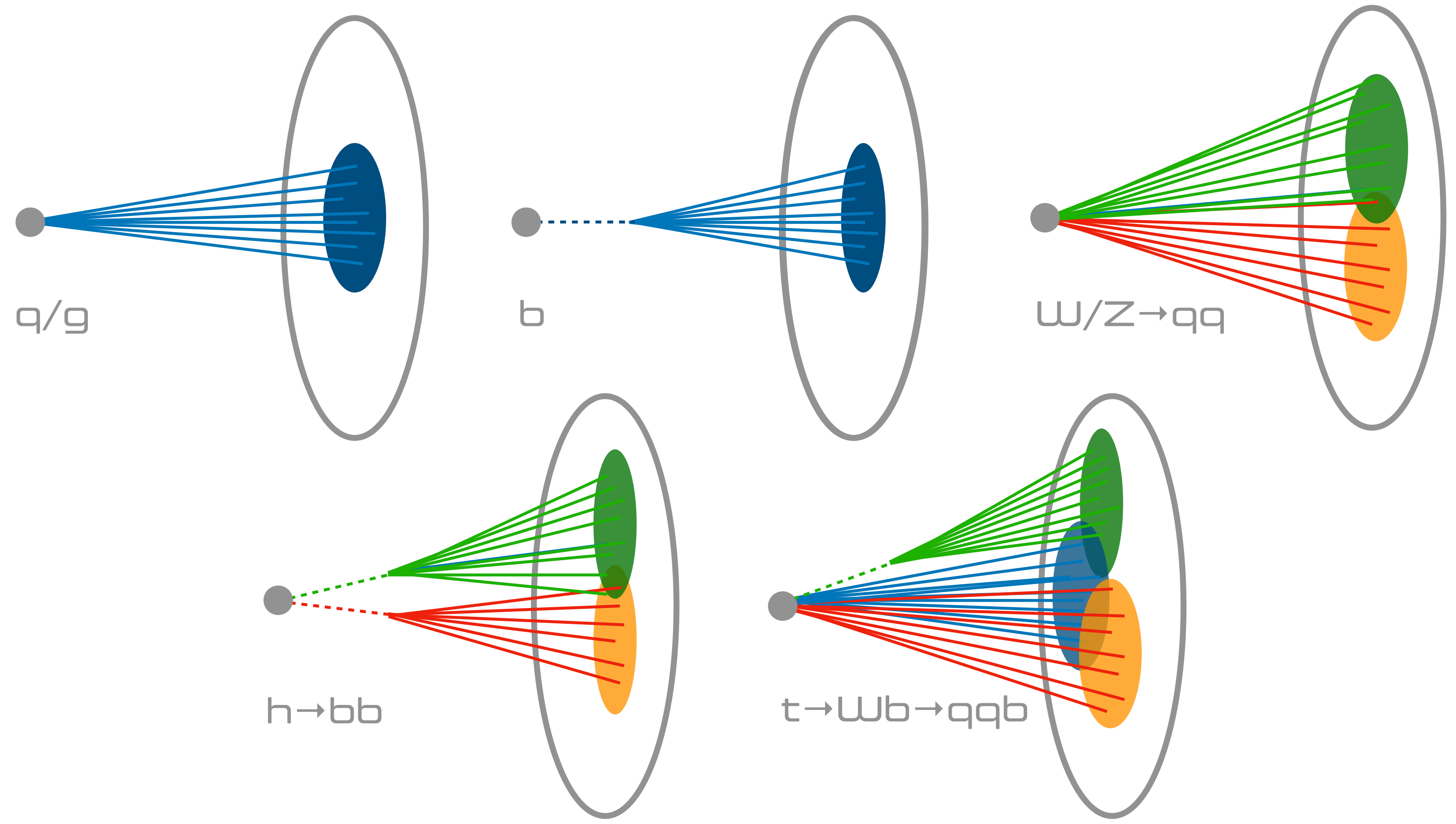
N_O: # of constituents
P: # of features
$N_E = N_O(N_O-1)$: # of edges
D_E: size of internal representations
D_O: size of post-interaction internal representation

ϕ_C, f_O, f_R parameterized as neural networks



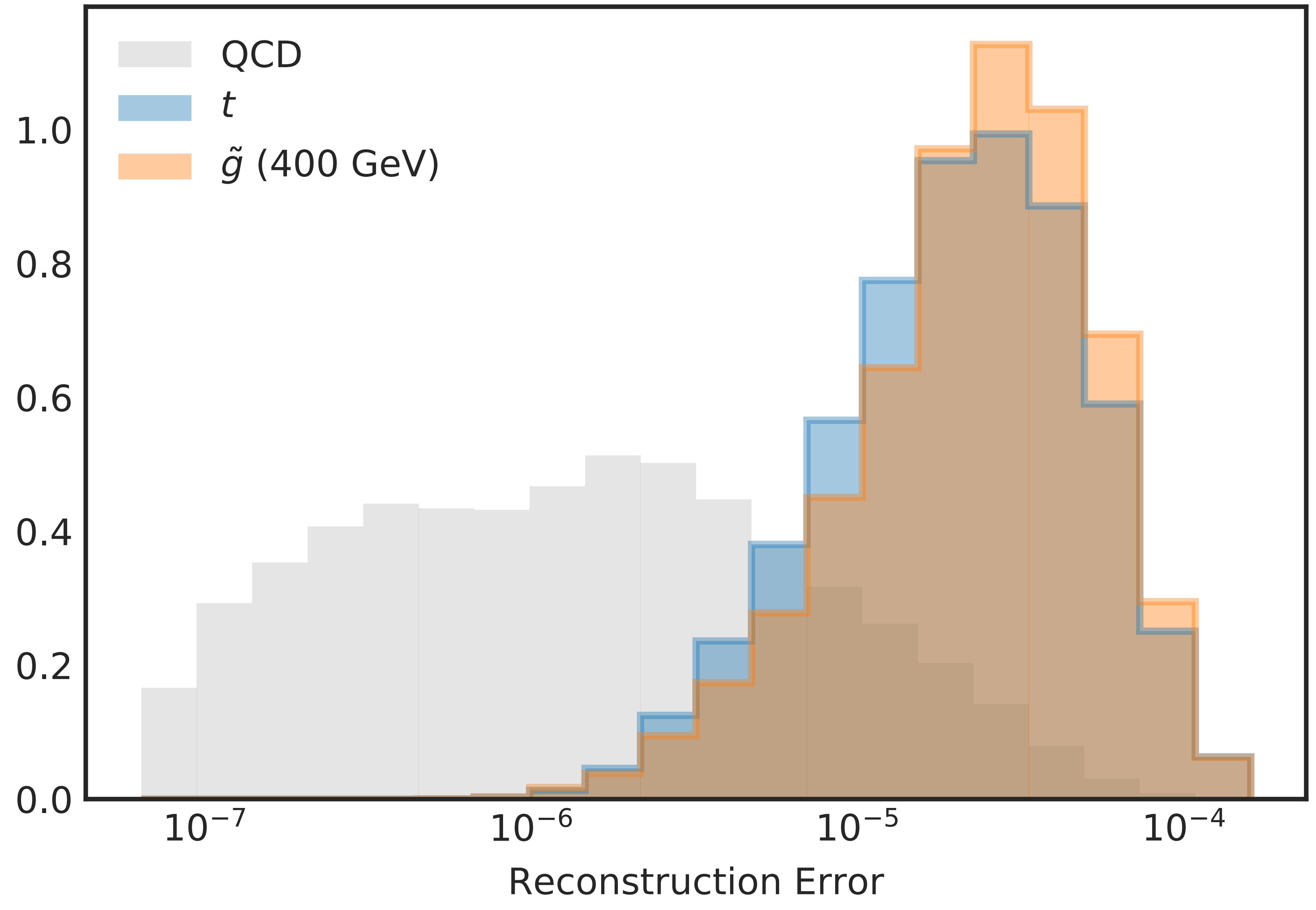
Interaction Networks

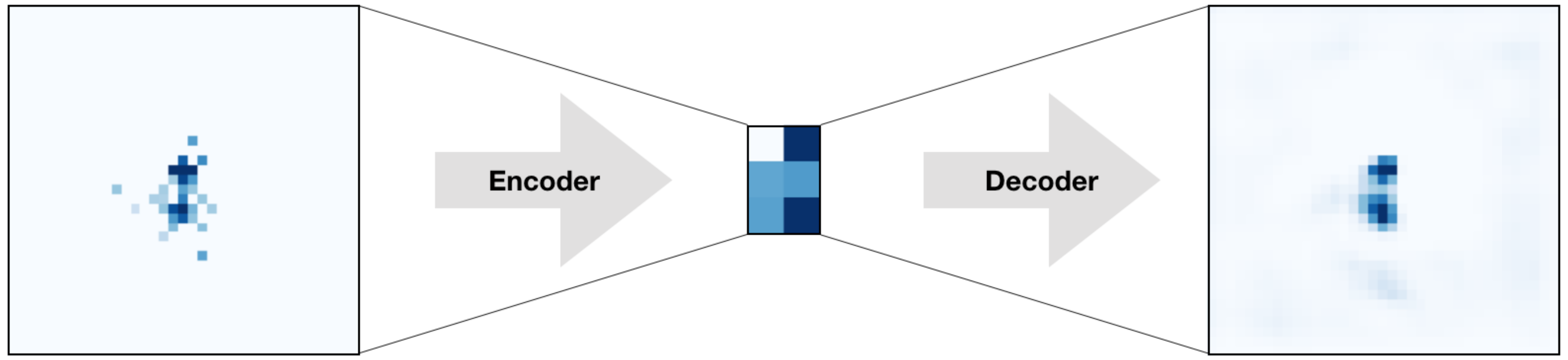


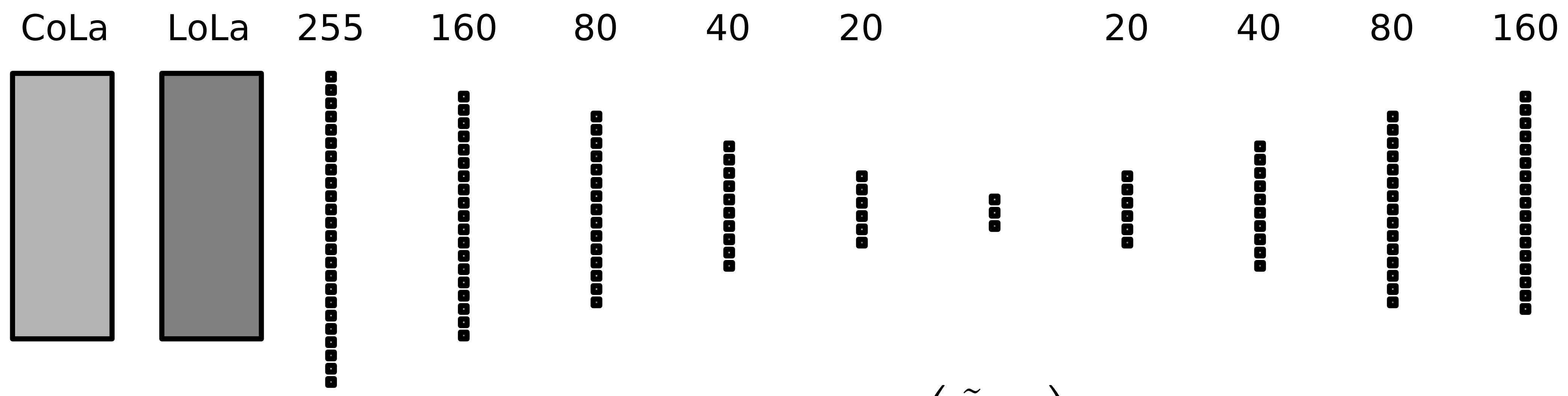


Autoencoders

- Use autoencoders to learn standard Physics
- Find new physics as "distant" events
- Based on image and physics-inspired representations of jets







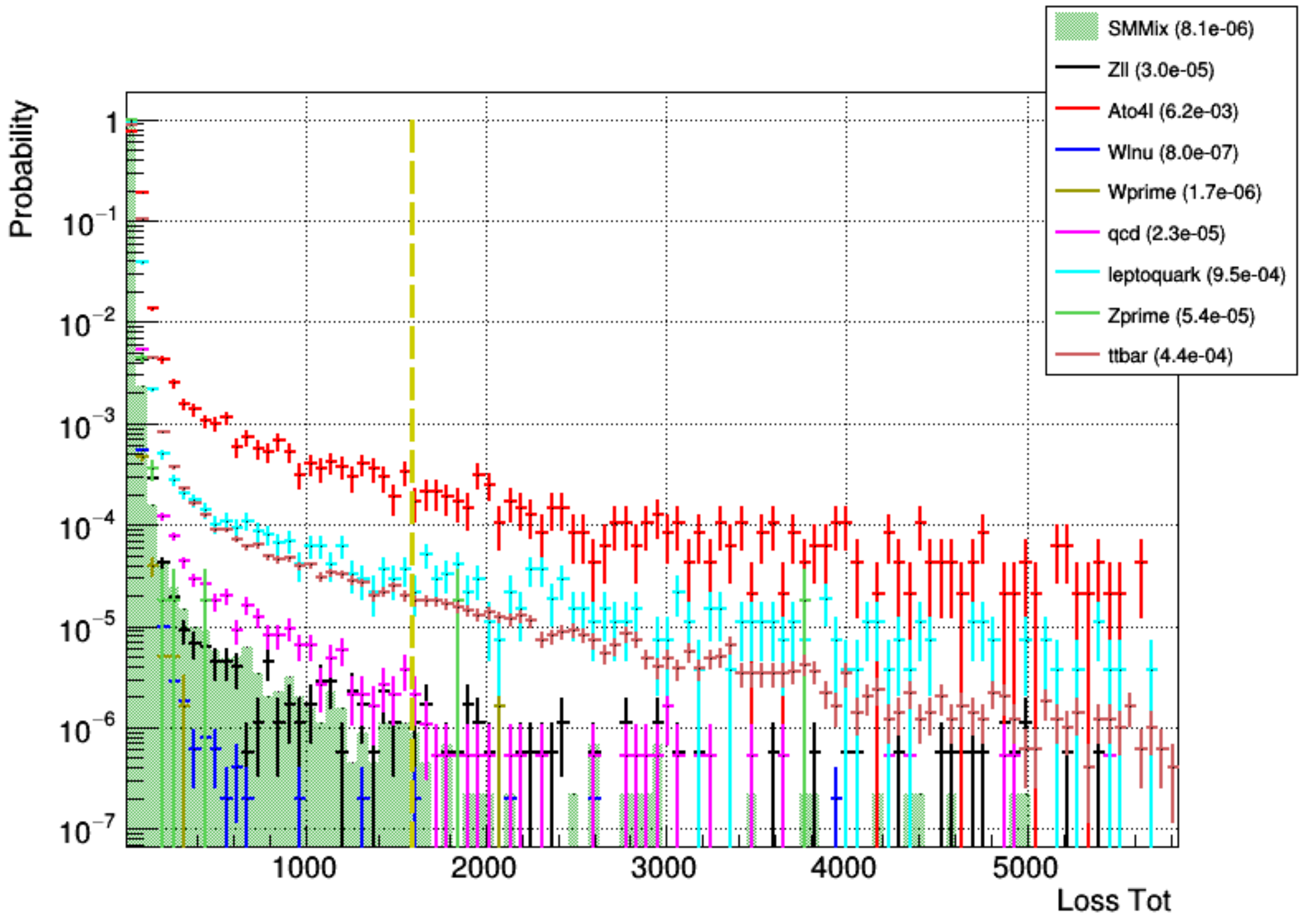
$$\tilde{k}_j = \begin{pmatrix} \tilde{k}_{0,j} \\ \tilde{k}_{1,j} \\ \tilde{k}_{2,j} \\ \tilde{k}_{3,j} \end{pmatrix} \xrightarrow{\text{LoLa}} \begin{pmatrix} \tilde{k}_{0,j} \\ \tilde{k}_{1,j} \\ \tilde{k}_{2,j} \\ \tilde{k}_{3,j} \\ \sqrt{\tilde{k}_j^2} \end{pmatrix} .$$

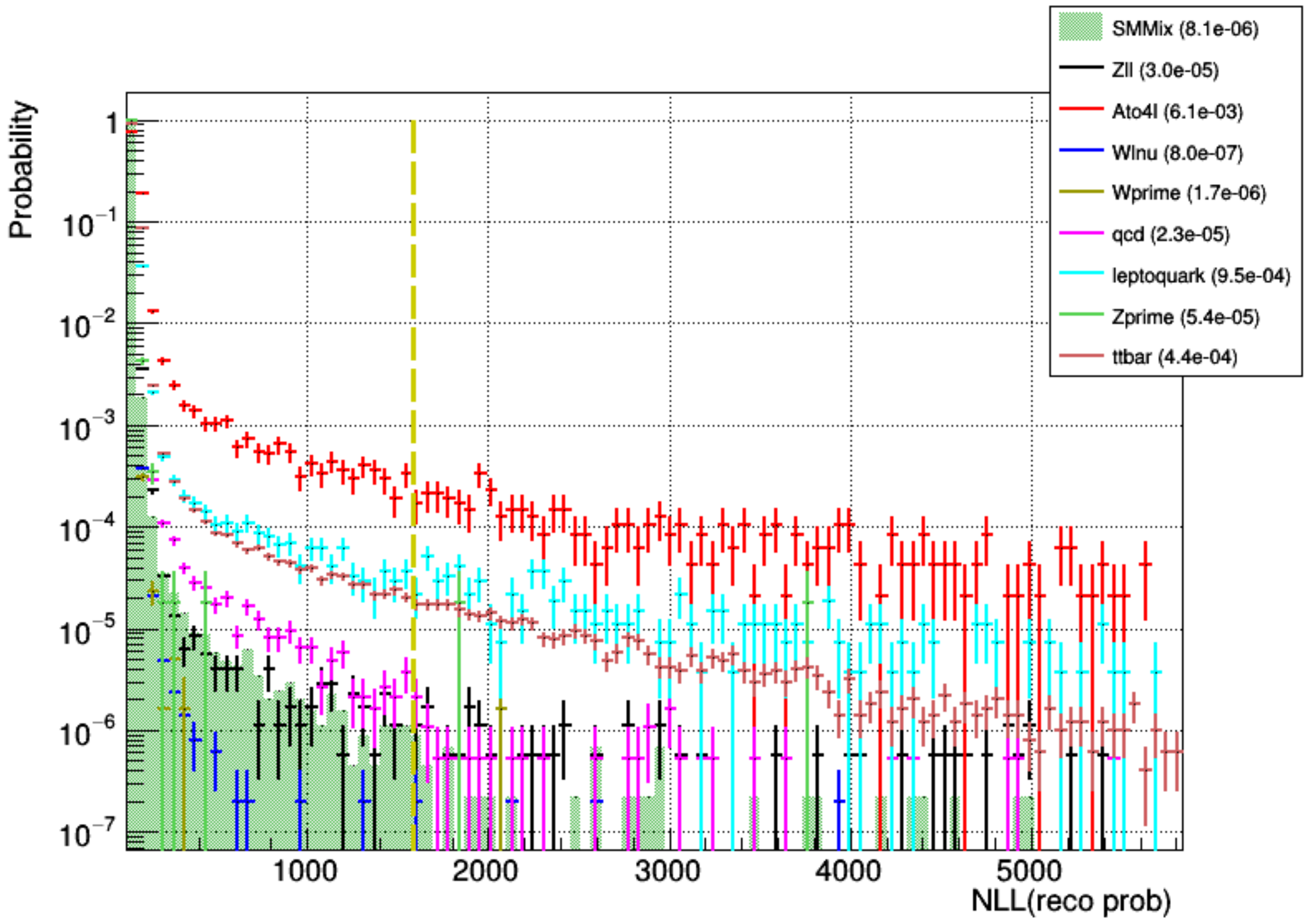
[Farina et al., arXiv:1808.08992](#)

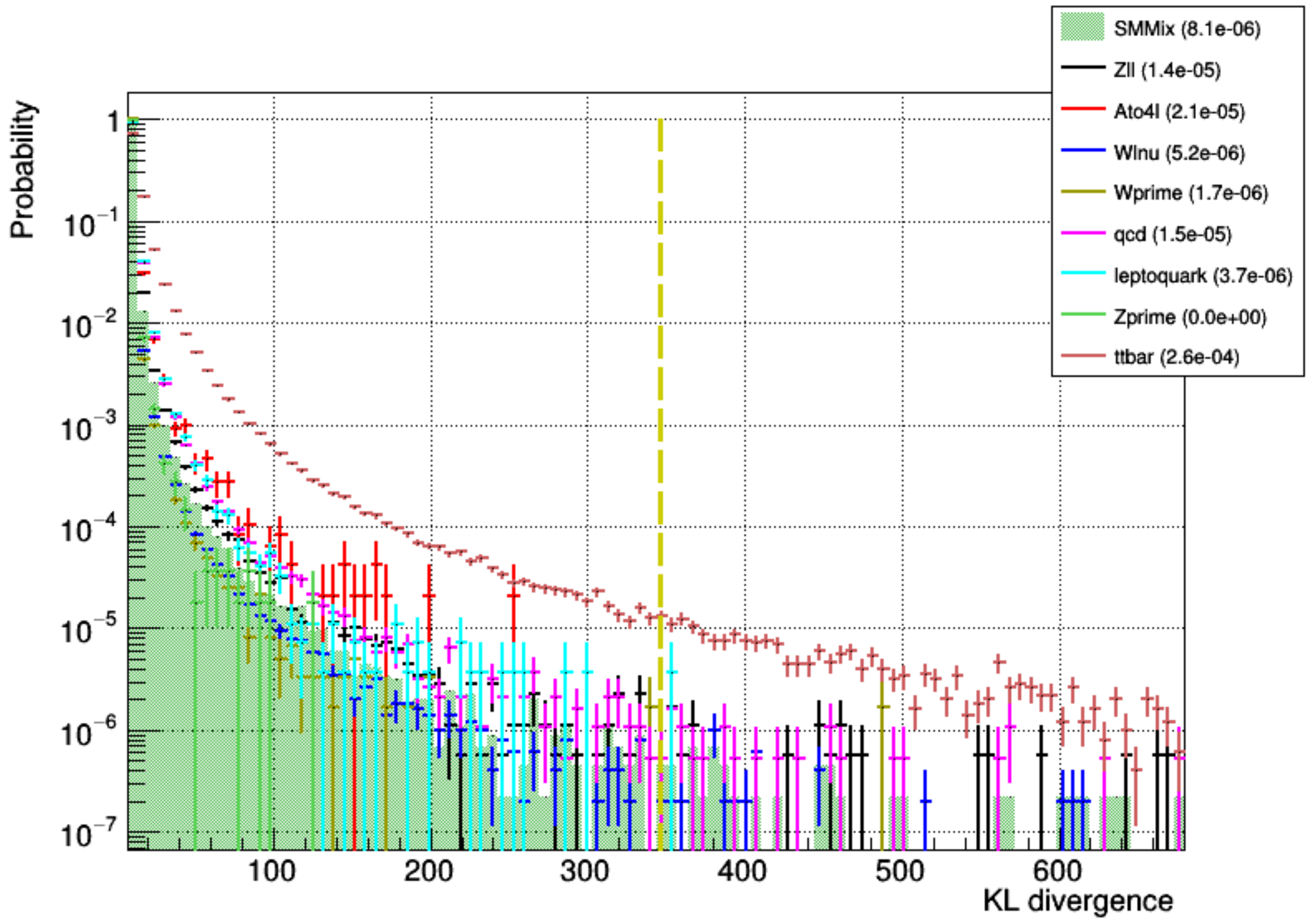
[Heimel et al., arXiv:1808.08979](#)

Anomaly Detection

- Anomaly defined as a p-value threshold on a given test statistics
  - Loss function an obvious choice
  - Some part of a loss could be more sensitive than others
  - We tested different options and found the total loss to behave better

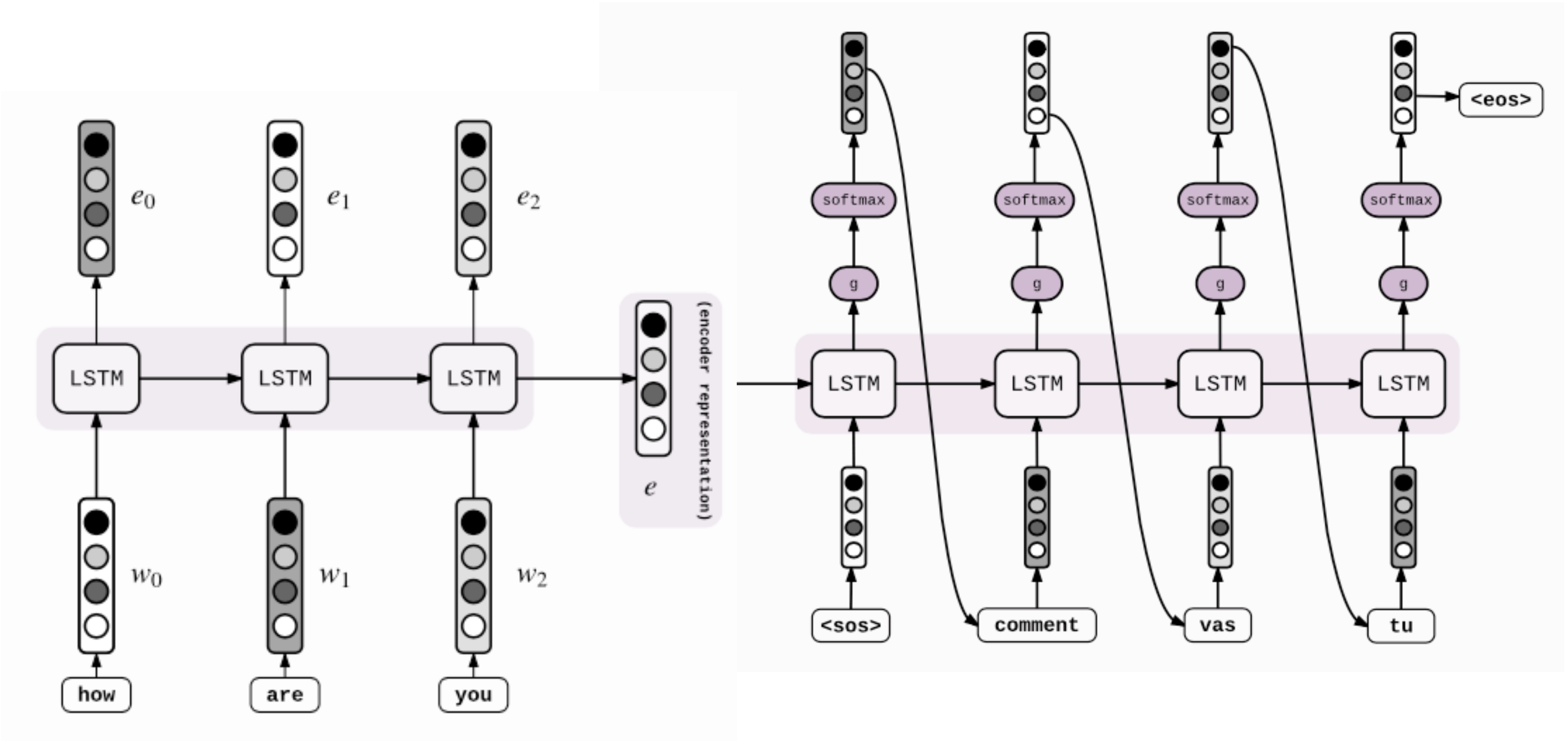






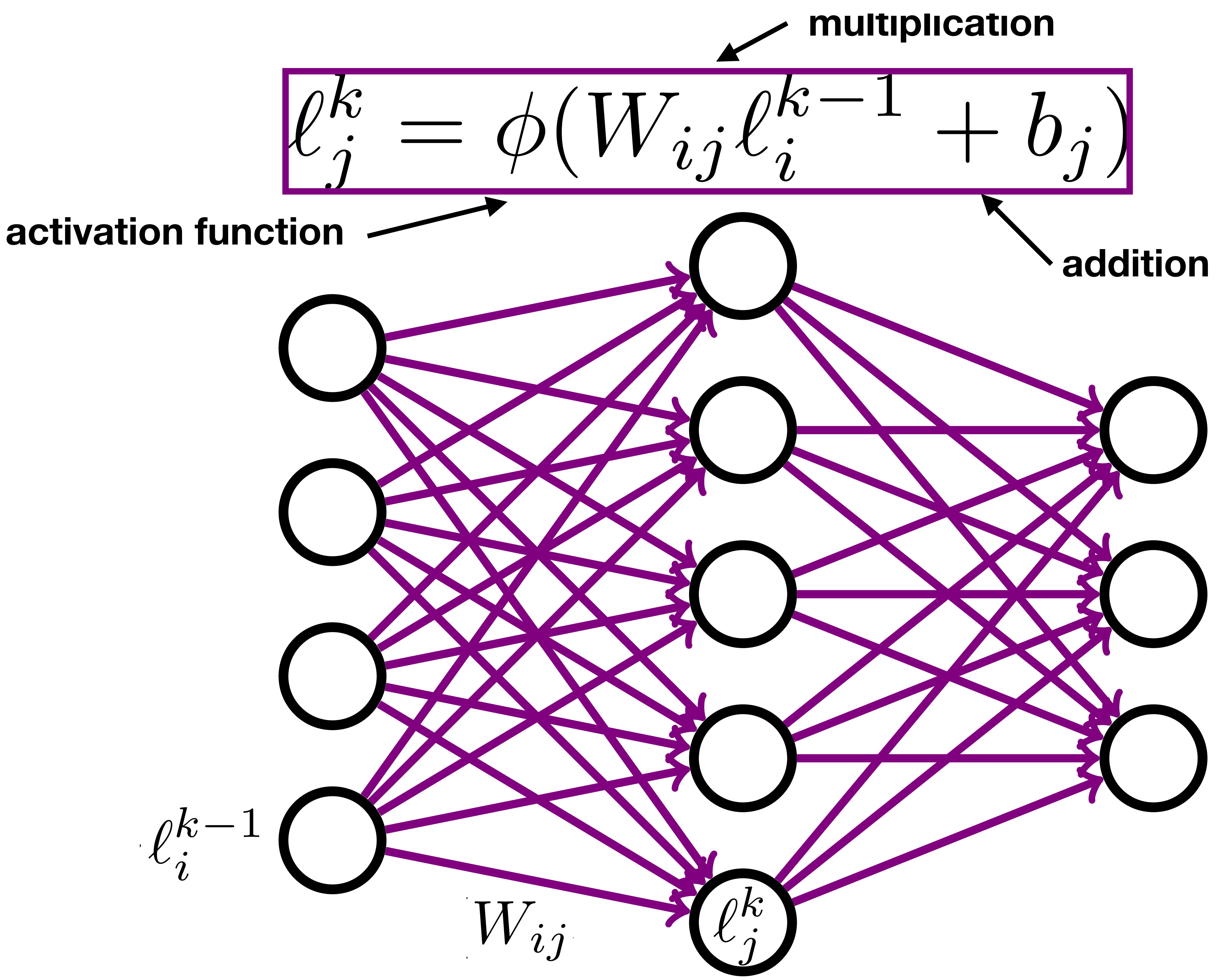
VAE with PF particles

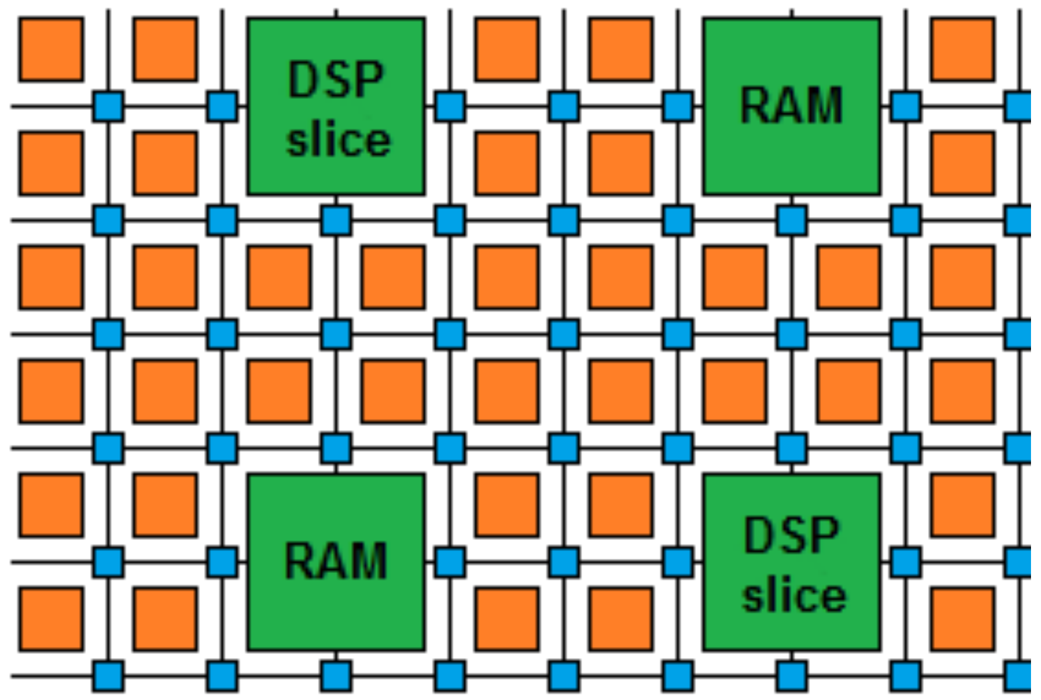
- Issues:
  - variable number of particles/event as input
  - need to return particles as output
- Networks used for translation
  - start from a sentence in language
  - code its meaning in some latent space z
  - translate to some other language, generating words from z



Network Operations

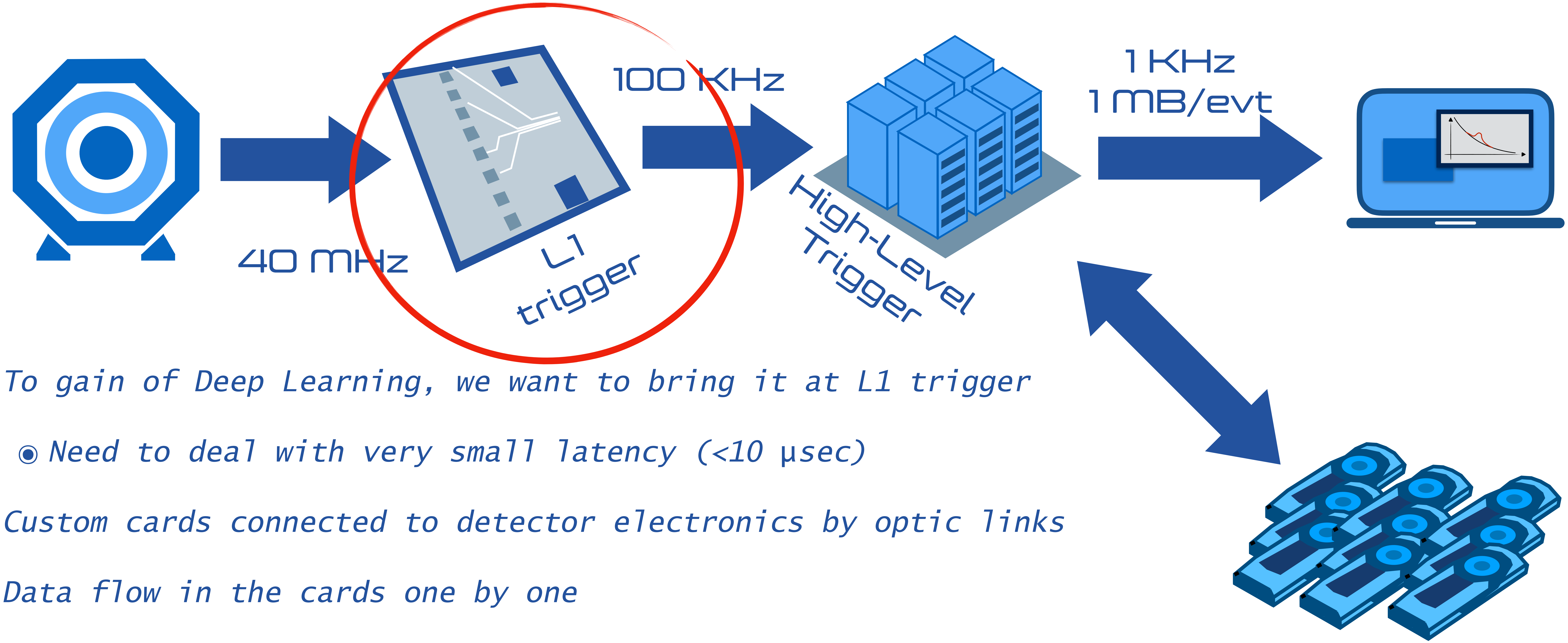
- *A classic Dense NN manipulate the inputs in three ways*
  - *multiplying by weights*
  - *adding biases*
  - *applying activation functions*
- *All these operations map nicely into an FPGA*
  - *high IO, DSPs, LUTs, tunable precision*





European Research Council

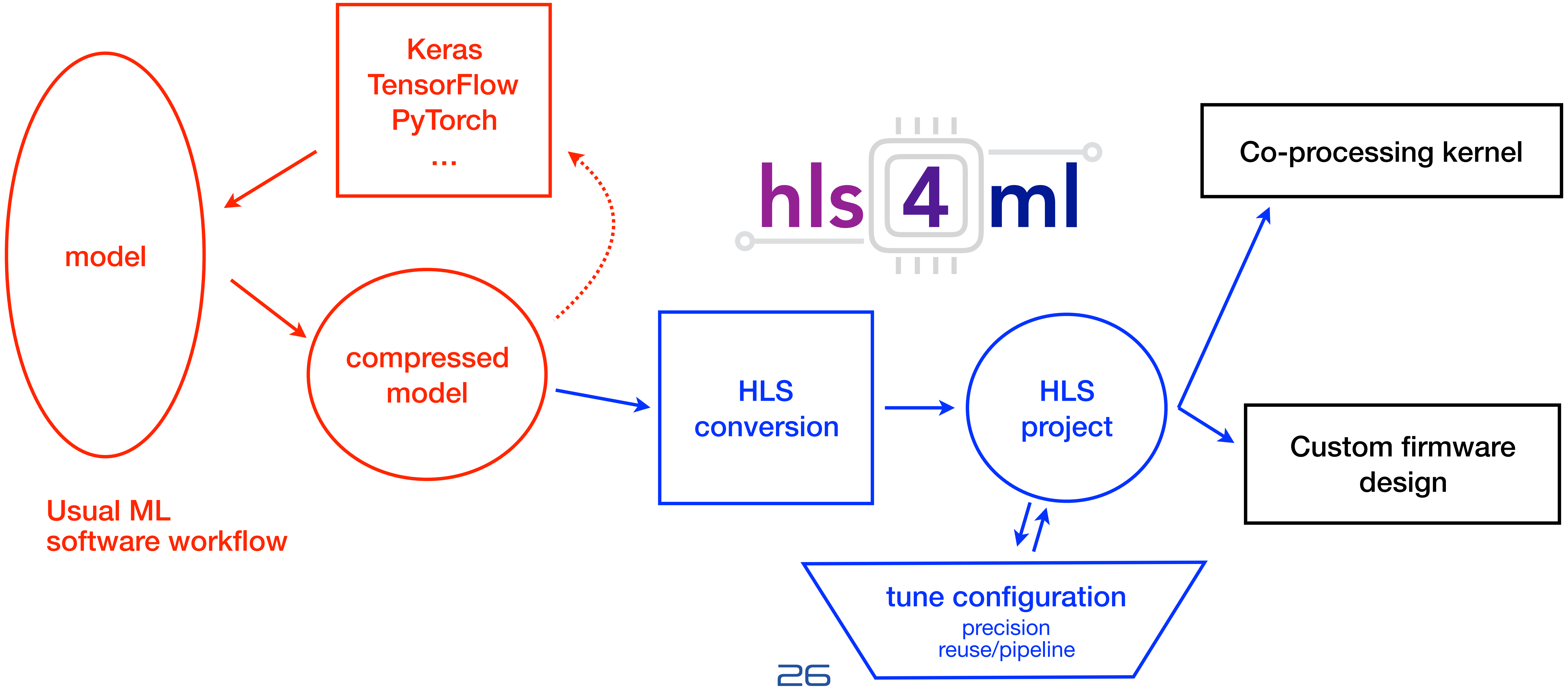
Deep Learning at L1



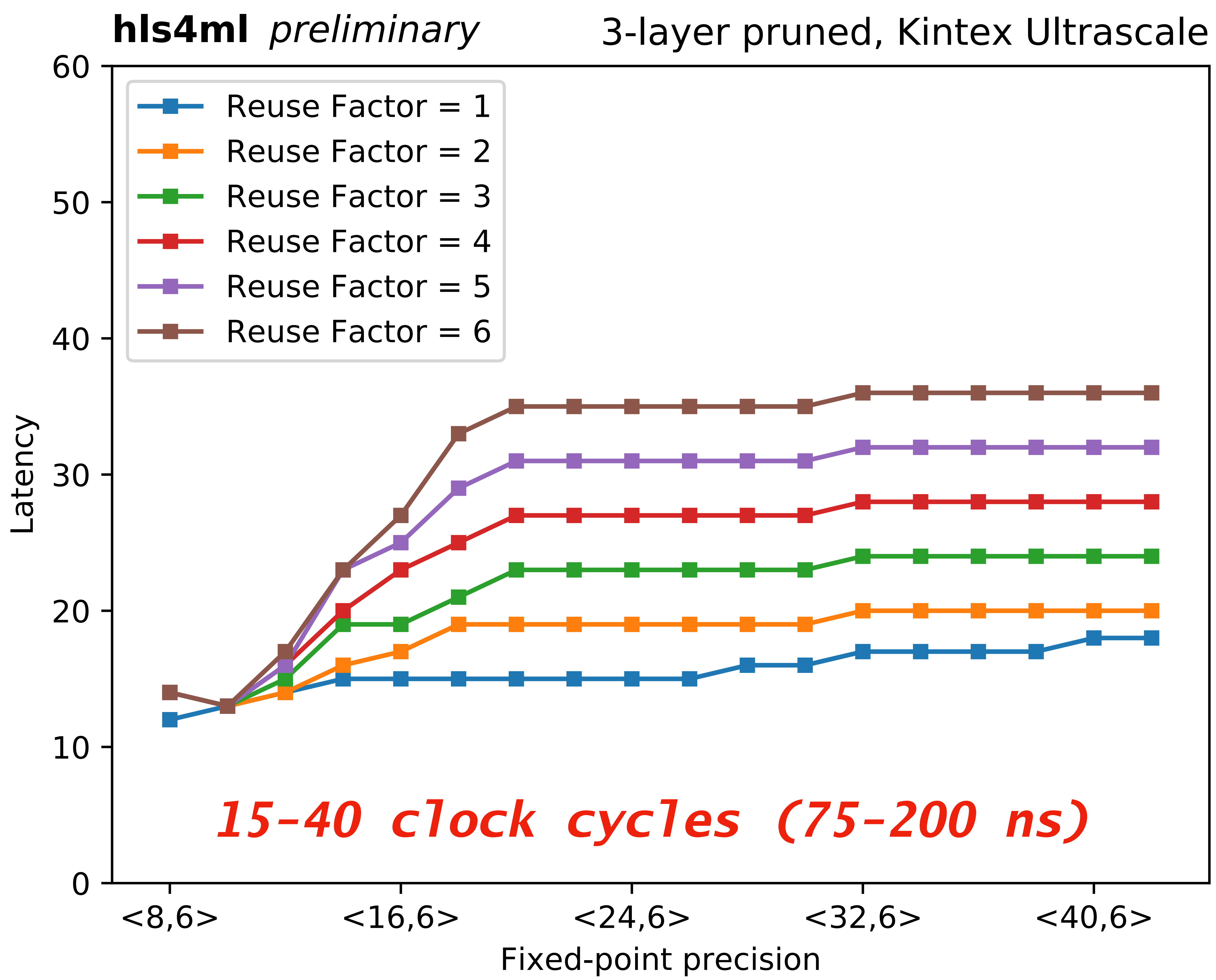
- To gain of Deep Learning, we want to bring it at L1 trigger
  - Need to deal with very small latency (<10 μsec)
- Custom cards connected to detector electronics by optic links
- Data flow in the cards one by one
- Networks need to be implemented in FPGA firmware
  - advanced design by expert engineers (not common resource in HEP)
  - automatic translation tools doing the job

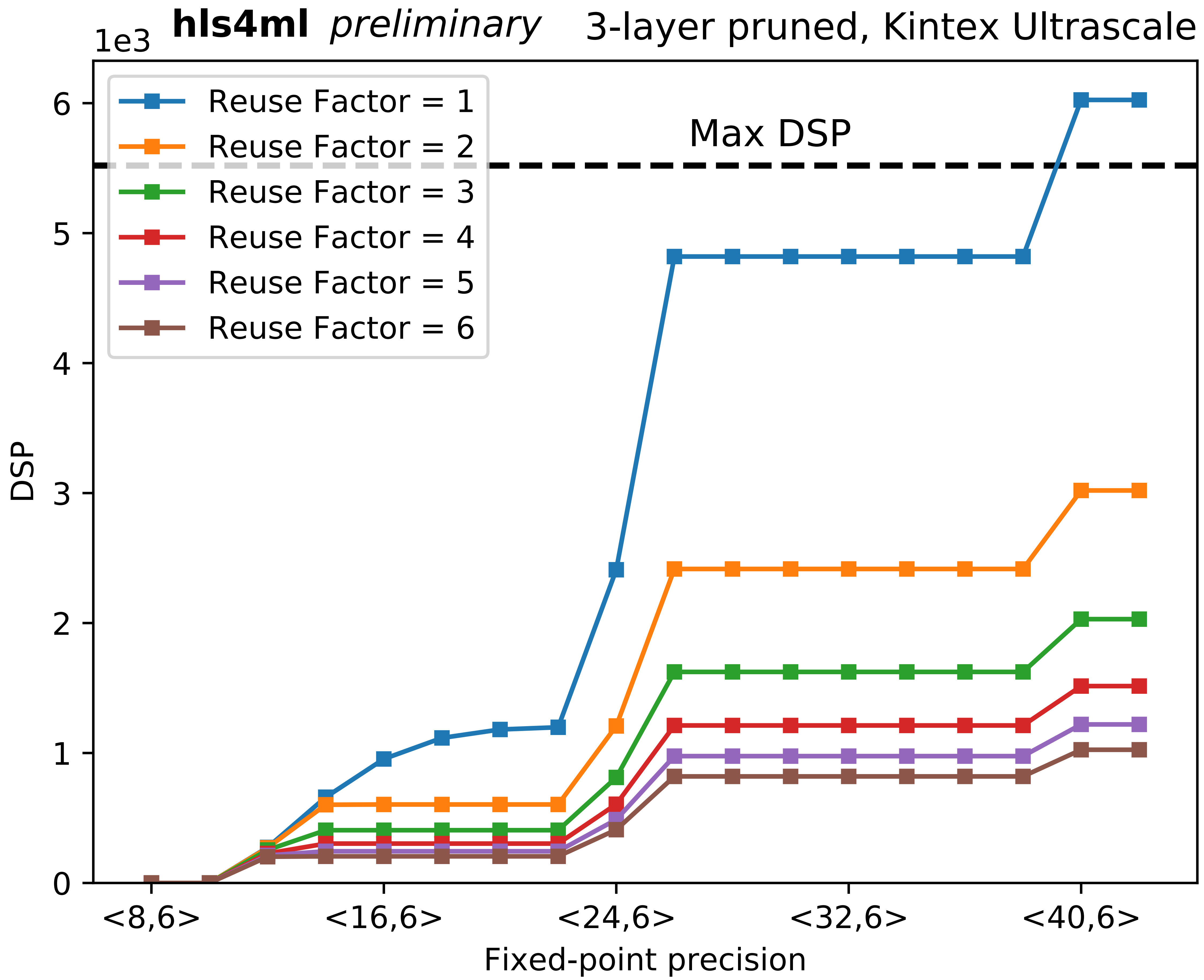
HLS4ML

- *HLS4ML aims to be this automatic tool*
  - *reads as input models trained on standard DeepLearning libraries*
  - *comes with implementation of common ingredients (layers, activation functions, etc)*
  - *Uses HLS softwares to provide a firmware implementation of a given network*
  - *Could also be used to create co-processing kernels for HLT environments*
- *It turns a neural network into an electronic circuit, emulated on the FPGA*



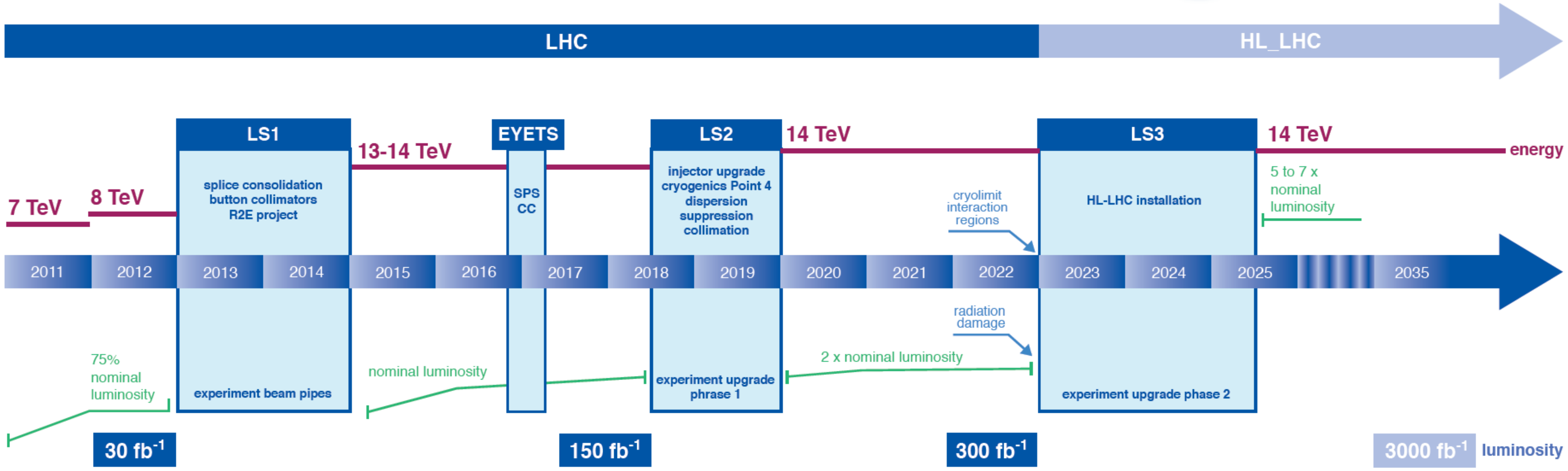
Parallelisation





Foreseen architecture (FPGAs) will handle these networks
Inference-optimized GPUs could break the current paradigm
Looking forward to R&D projects with nVidia & E4 on this

Deep Learning 4 HEP: 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***
  - *Dark Matter underground experiments*
  - *Neutrino experiments*
- *Collaboration with nVidia to find optimal (performances & speed) solutions*