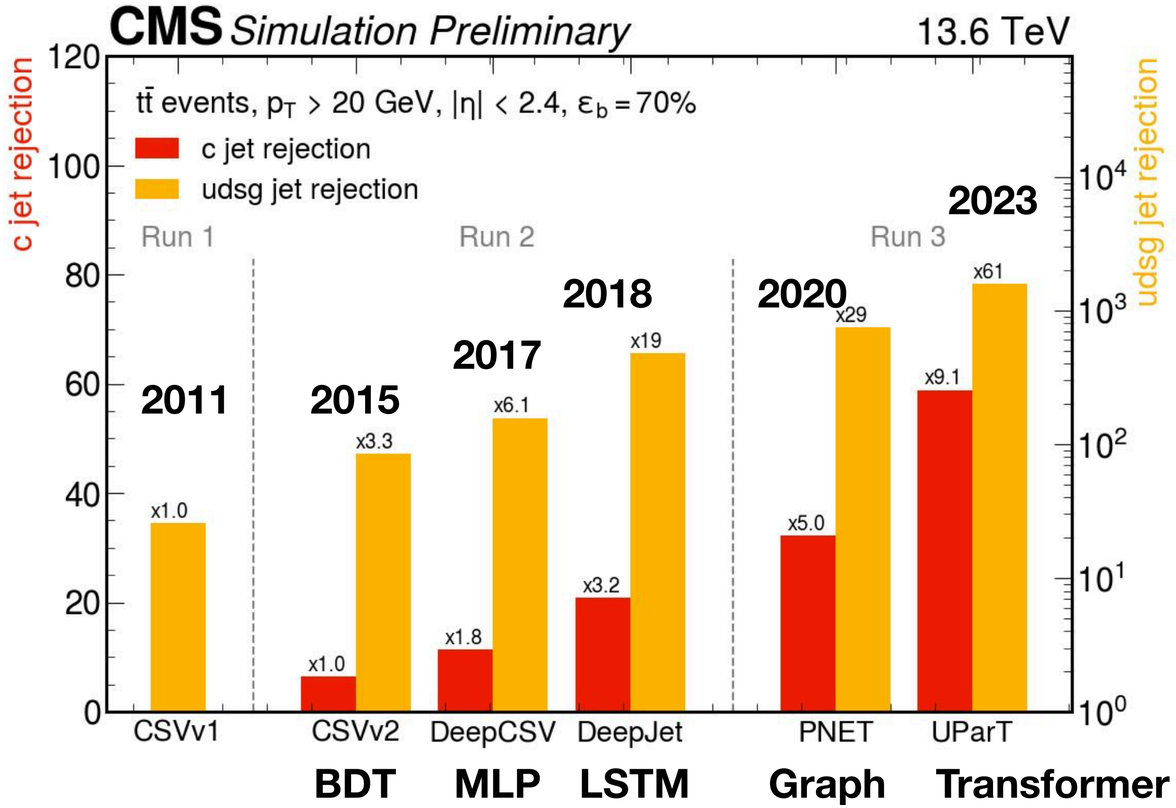
CERN and Graph Neural Networks

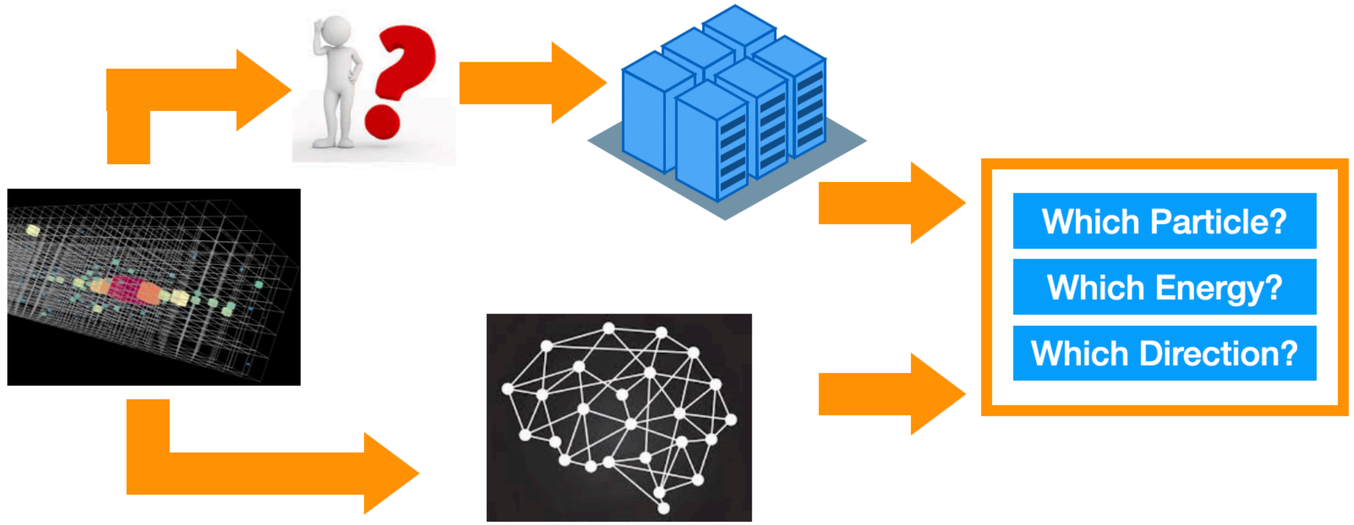
A brief History

- *Serious activity on Deep Learning applications at LHC started around 2015*
  - *Series of workshops to reach out AI experts outside HEP and start collaborations*
  - *Several proof-of-principle studies*
- *Heavy R&D activity since then, with young and enthusiastic community growing year by year*
  - *By now, DL is established in our data processing and analysis*



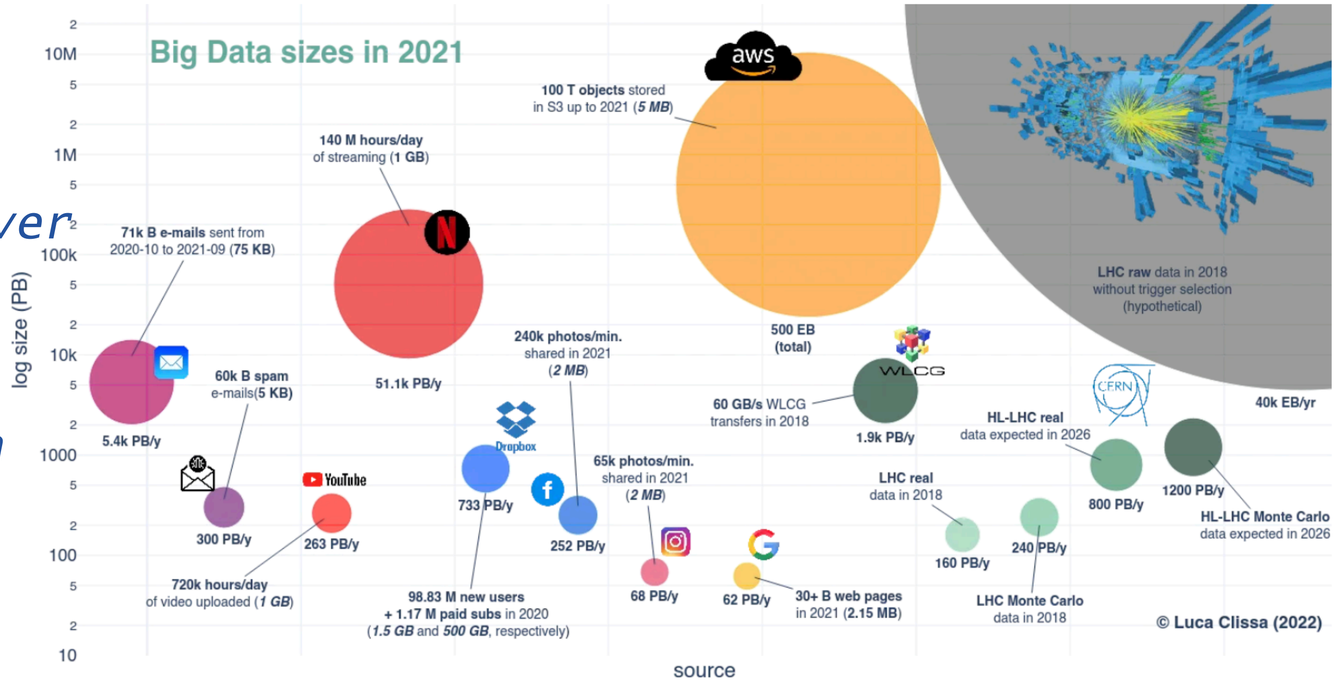
What Do we Use DL for

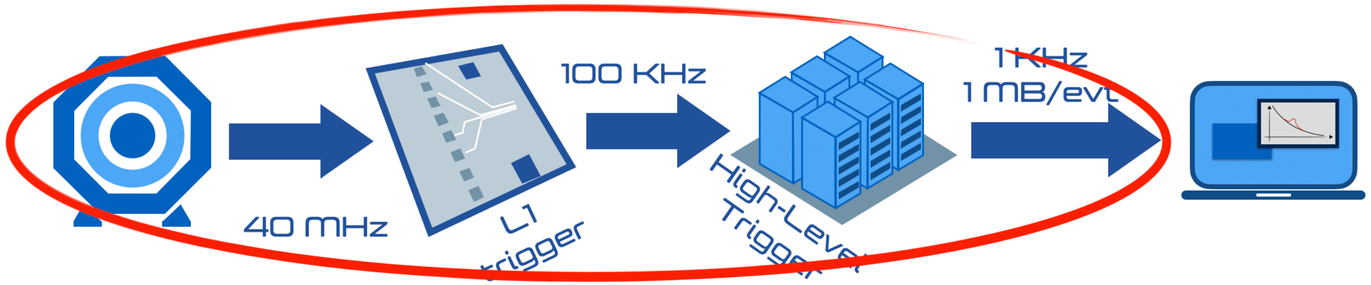
- *Better solutions: in typical "offline environment" (aka data analysis), DL is used to improve signal-to-background discrimination*
  - *Regression, Classification, Anomaly Detection*
- *Faster solutions: in most CPU-heavy processes (pattern recognition, clustering, generation) and downstream tasks, we have robust solutions to our problem. DL is mostly used as an approximation of these solutions, which could run faster*



Where DL could help the most

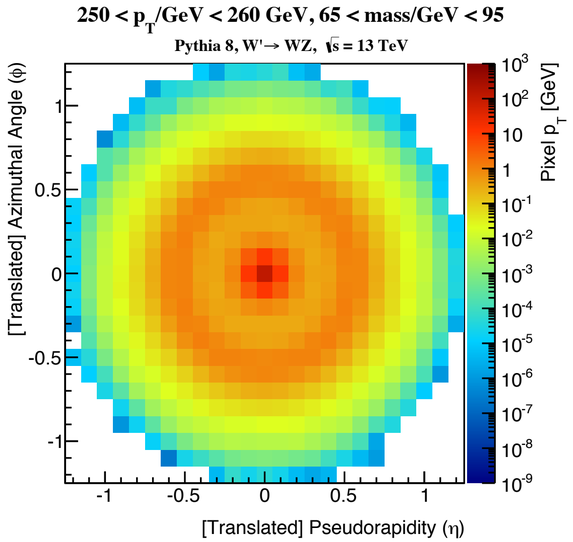
- *LHC produces O(10)M collisions/sec*
- *Each collision is O(1) MB*
  - *The largest data stream humankind ever handled*
- *Events selected in a two-tier data processing system, which selects which are to be taken*
  - *First layer on custom electronic (FPGAs), responds within O(1) μs*
  - *Second layer on CPU farm (GPU accelerated in certain experiments), responds within O(100) ms*
- *Bringing DL closer to the detector provides a fast-to-execute high-complexity solution for a better decision*

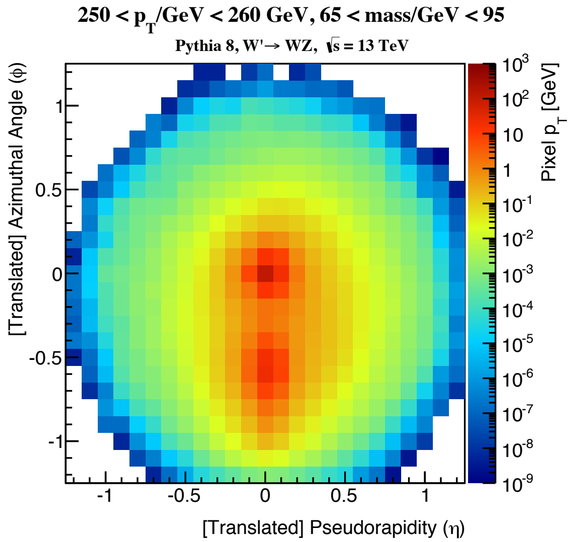


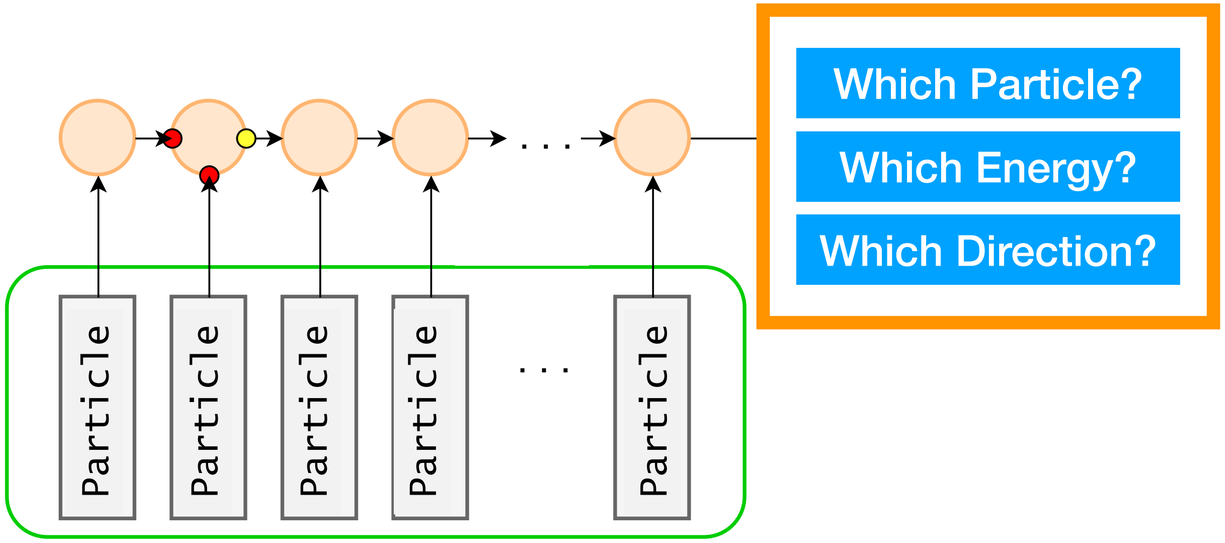


It started with DNN/CNN

- *The very first proof of concepts focused on DNNs*
  - *Replaced BDTs with more powerful NNs acting on high-level features. Not always big improvements (BDTs good enough often)*
- *Moved to more RAW data representations*
  - *formatting our data as images -> CNN*
  - *formatting our data as sequences -> RNN/LMST/GRU*
- *Seen improvements, but still limited by the need to hammer our data into a data representation that was not natural*
  - *Lost information in the process*

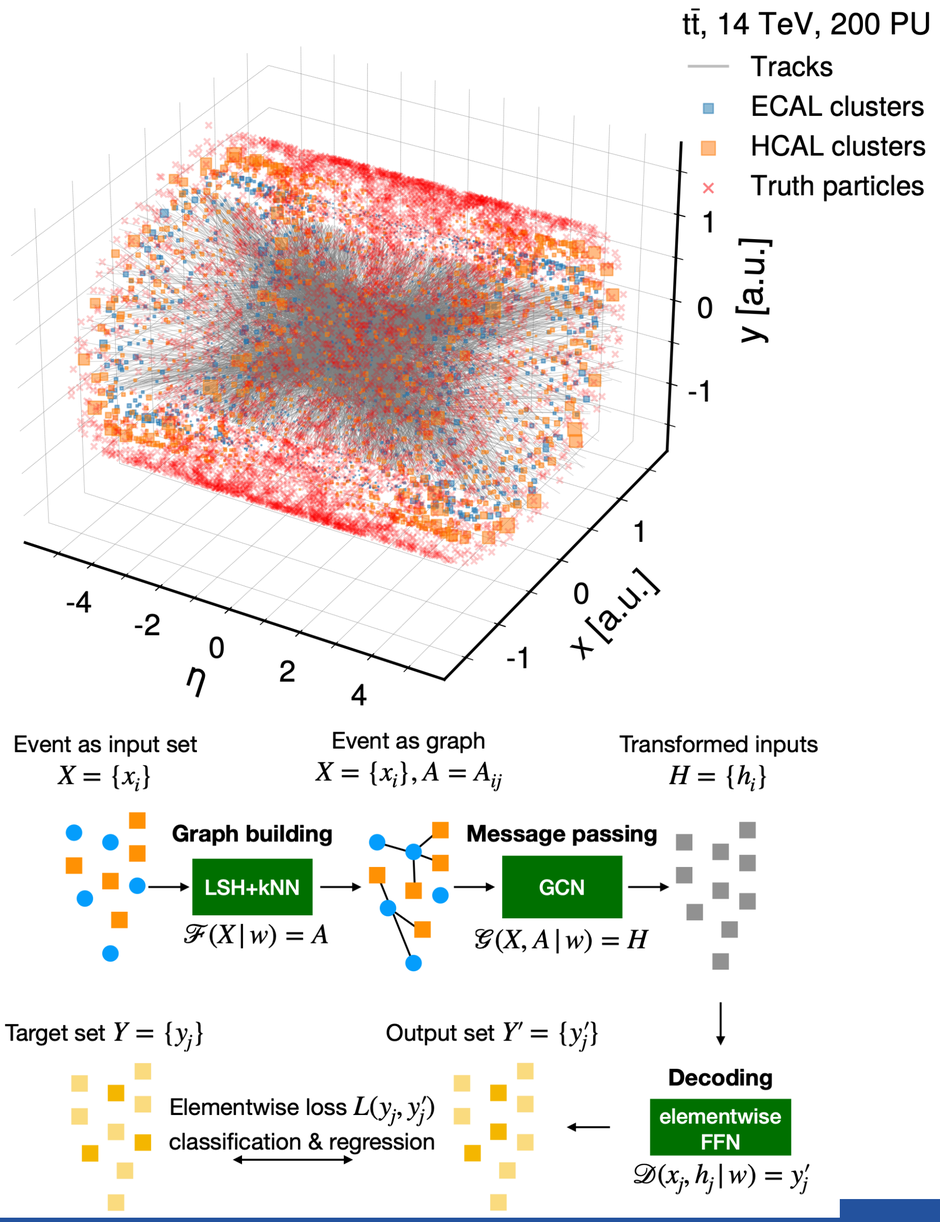






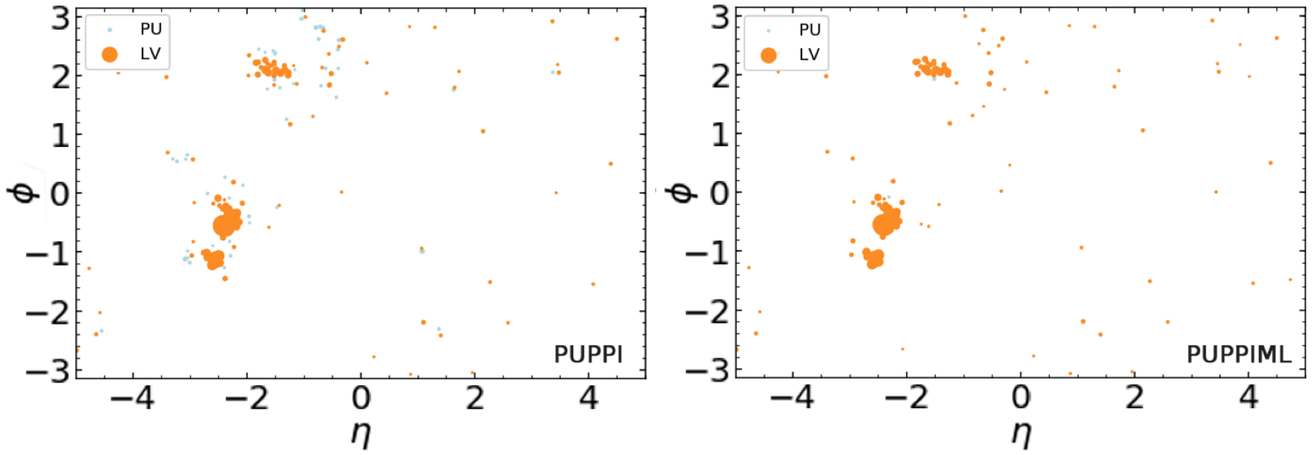
Our Data Are Point Clouds

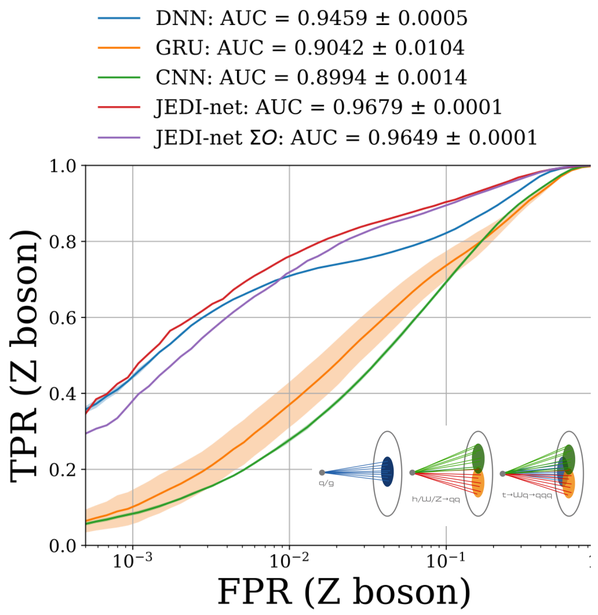
- *Around 2017, thanks to conversation with DeepMind researchers, it became clear that we had to move to graph neural networks*
  - *Our data are effectively point clouds*
- *Since then, several works showed how graph nets could provide superior solution wrt other DL architectures*
- *Now, Graph Nets are the paradigm*

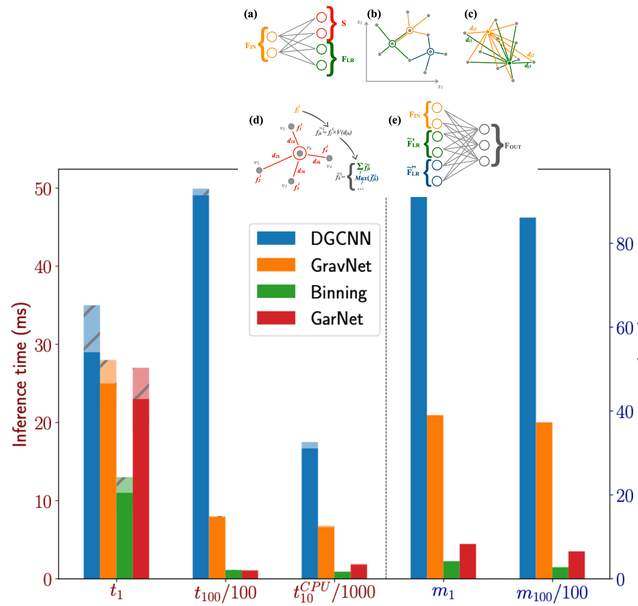


Our Data Are Point Clouds

- *Around 2017, thanks to conversation with DeepMind researchers, it became clear that we had to move to graph neural networks*
  - *Our data are effectively point clouds*
- *Since then, several works showed how graph nets could provide superior solution wrt other DL architectures*
- *Now, Graph Nets are the paradigm*

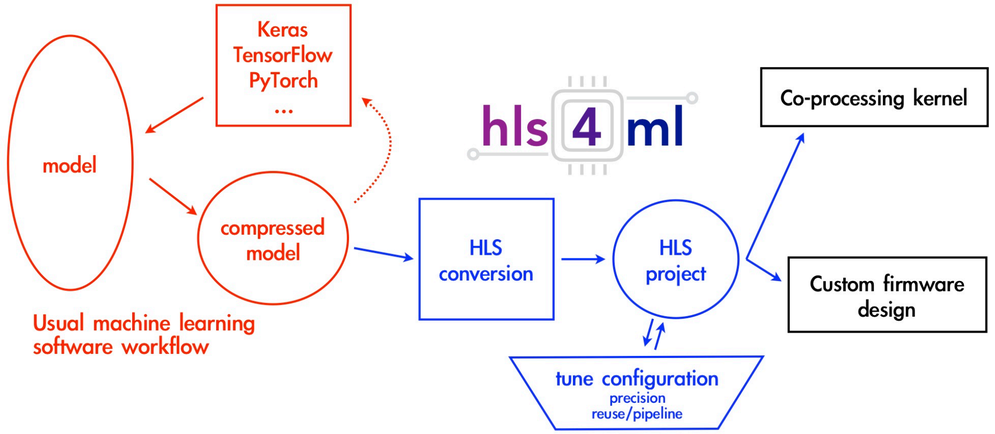


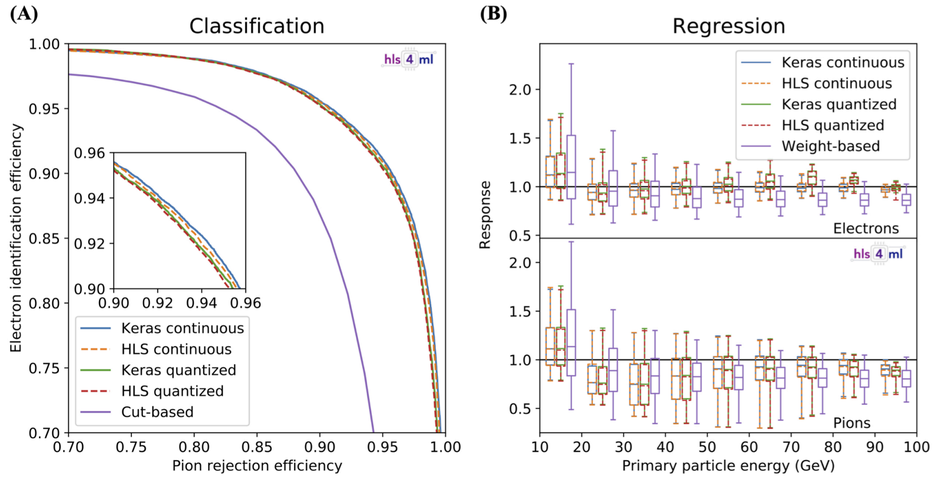




Accelerating DL inference

- *For us, it is crucial to be able to bring DL as close as possible to the detector*
  - *Only in CMS, ~ 20 projects of DNNs for first-tier hardware trigger for CMS upgrade*
- *In lack of solutions compatible with our ultrafast latency constraints, we developed an in-house solution*
  - *We managed to speed NN inference to O(100) sec for various architectures, including some graph networks*
  - *Plans for a more generic support of Graph Nets (might require heavy library rewriting)*
  - *Very interested to explore deployment on AI-specific hardware (we started a conversation with GraphCore at some point, killed by Covid and a former colleague of us/your, who then left your company), both to replace FPGAs at L1 and as an accelerator device at HLT*





Model	V_{max}	R_{reuse}	Latency (cycles)	Interval (cycles)	DSP (10^3)	LUT (10^3)	FF (10^3)	BRAM (Mb)	ROC AUC	Response RMS
Continuous	128	32	155	55	3.1 [56%]	57 [9%]	39 [2.9%]	1.8 [2.3%]	0.98	0.23
Quantized	128	32	148	50	1.6 [29%]	70 [11%]	41 [3.1%]	1.9 [2.4%]	0.98	0.24
Quantized	64	16	99	34	1.6 [29%]	63 [9%]	38 [2.9%]	1.8 [2.3%]	0.96	0.24
Quantized	32	8	75	26	1.4 [25%]	52 [8%]	33 [2.5%]	1.8 [2.3%]	0.86	0.37
Quantized	16	4	63	22	1.5 [27%]	57 [9%]	37 [2.8%]	1.8 [2.3%]	0.64	0.36

50 nsec/cycle