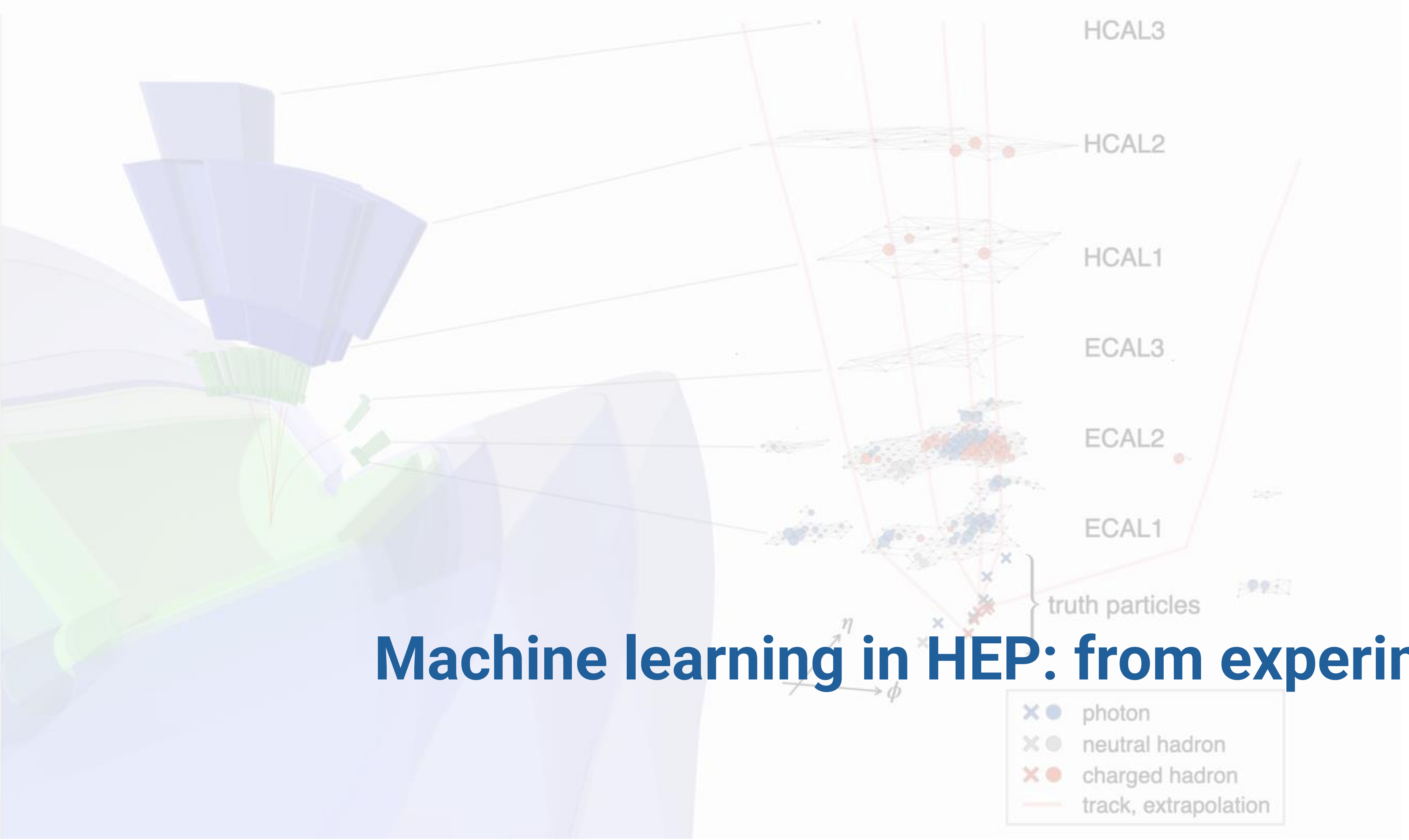


Machine learning in HEP: from experiment to theory

Università di Genova

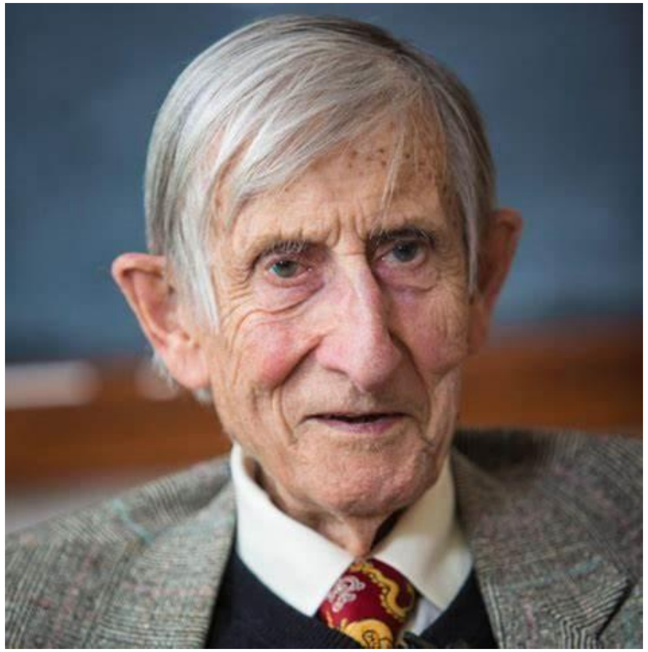
Francesco Armando Di Bello, U. and INFN Genova. 04/10/24

INFN

Introduction

- I will describe the usage of ML from the prespective of an experimentalist.
- A selection of topic that I find the most relevant and promising will be discussed.
- ML is having a strong impact to the HL-LHC projections and beyond, I will summarize the state-of-the art techniques at LHC experiments, and then move to a more speculative part

"New directions in science are launched by new tools much more often than by new concepts."
F. Dyson

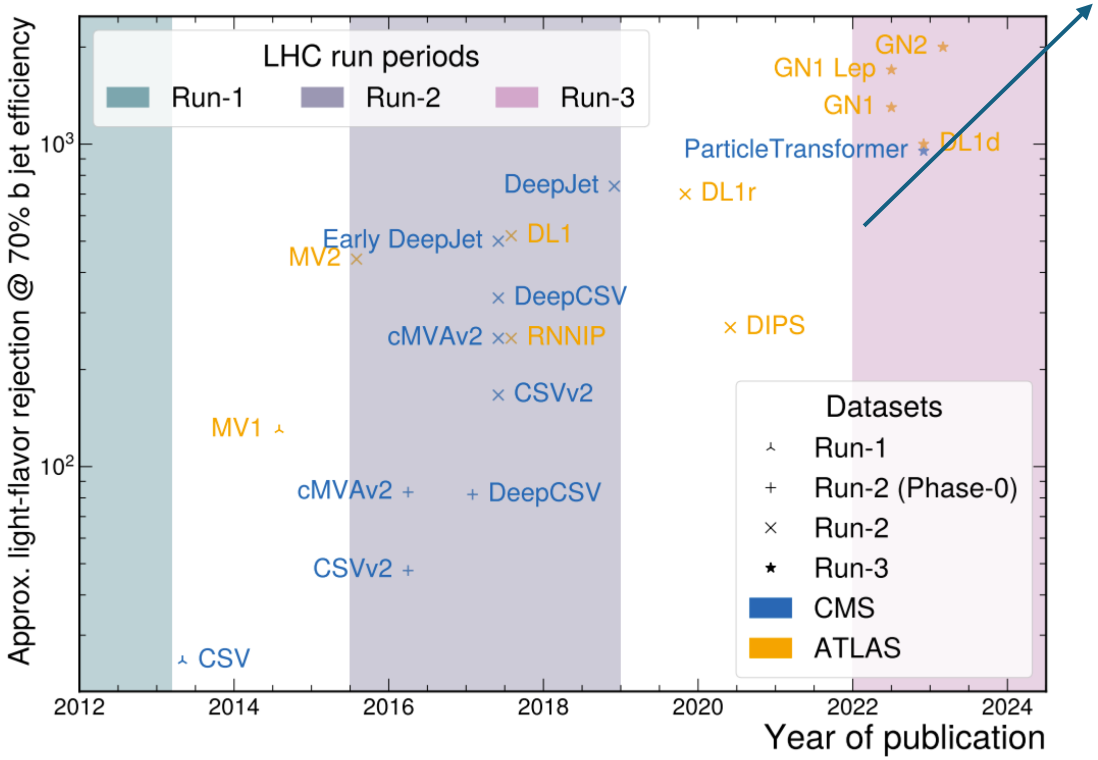


Thanks to L. Henrich for the inspiration

The raise of ML in HEP

Will use b-jet identification as a benchmark to showcase improvements (similar in other kind jet-tagging)

Are we reaching a plateau?

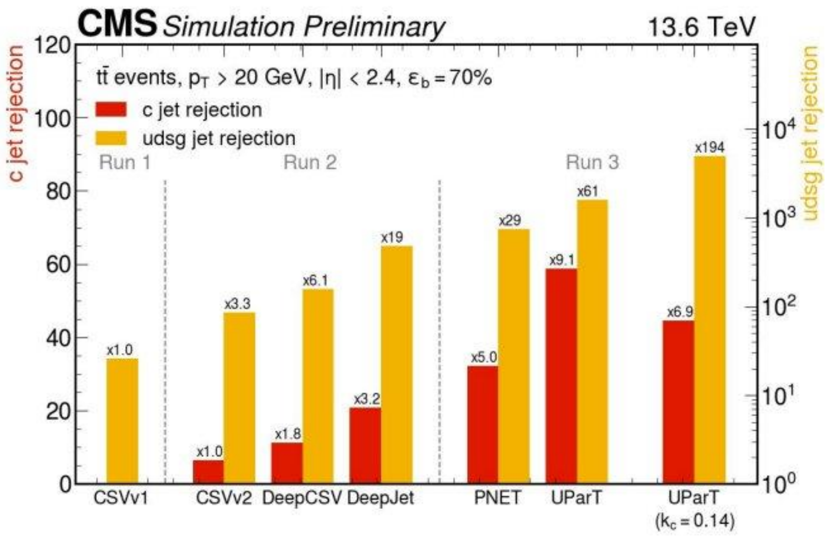


Attention Is All You Need

Nowadays, transformers are the state-of-the-art in HEP experiments

1706.03762 (2017)

From ML to physics

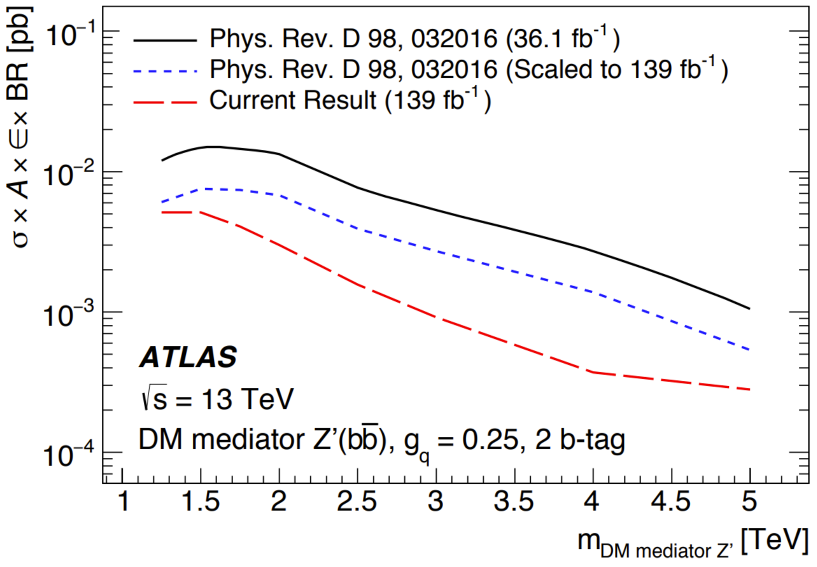


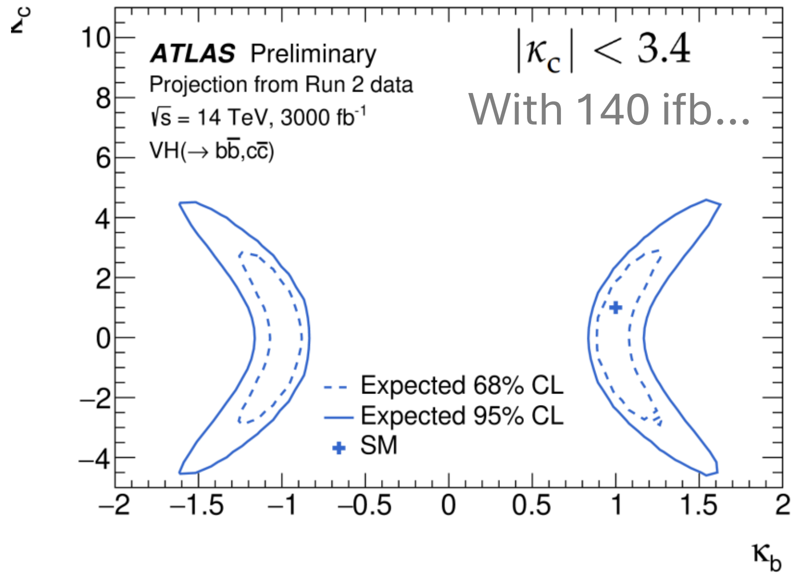
The usage of state-of-the-art ML techniques in experiments is dominant, and in certain cases, even opening new frontiers!

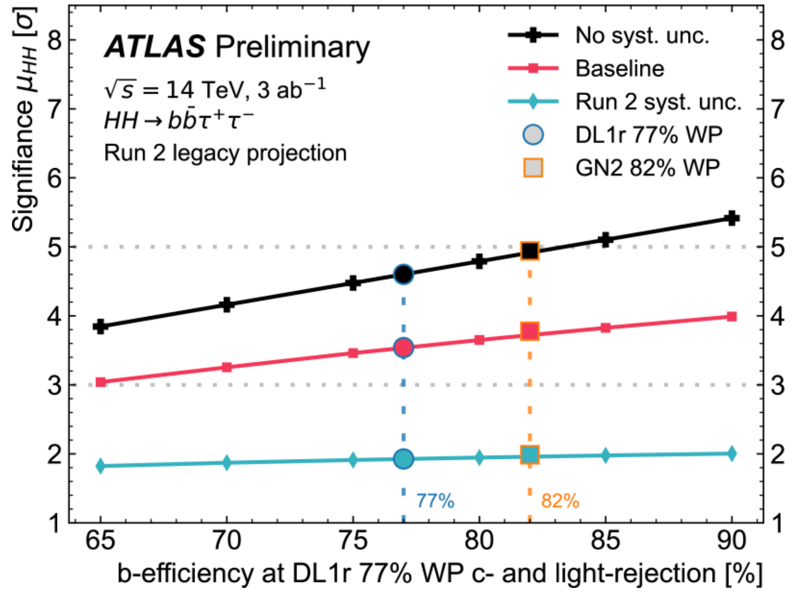
High mass searches

Direct charm Yukawa

Self-coupling







high-mass searches

ATLAS extrapolation
charm Yukawa CMS

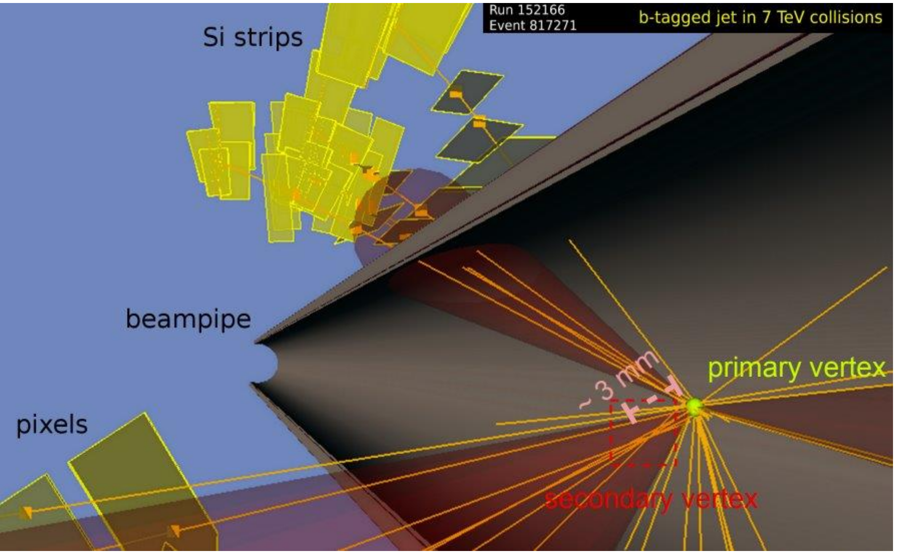
ATL-PHYS-PUB-2024-016

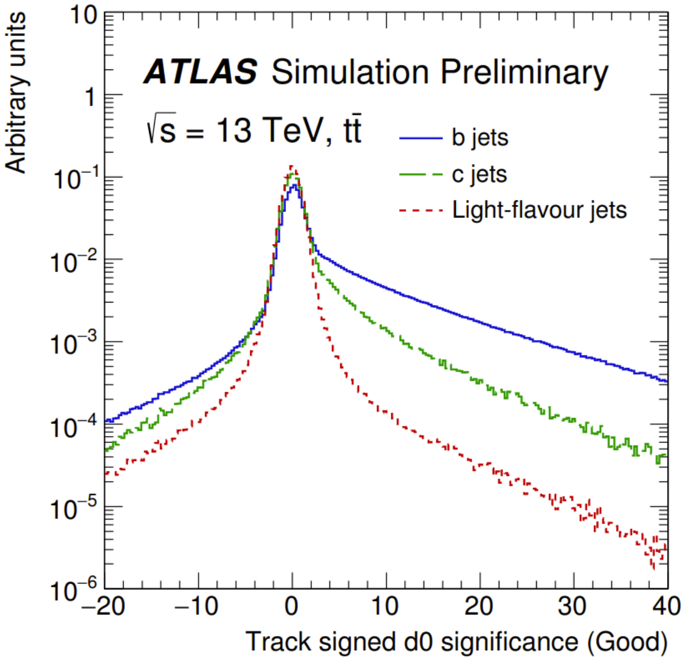
Why is ML useful for HEP?

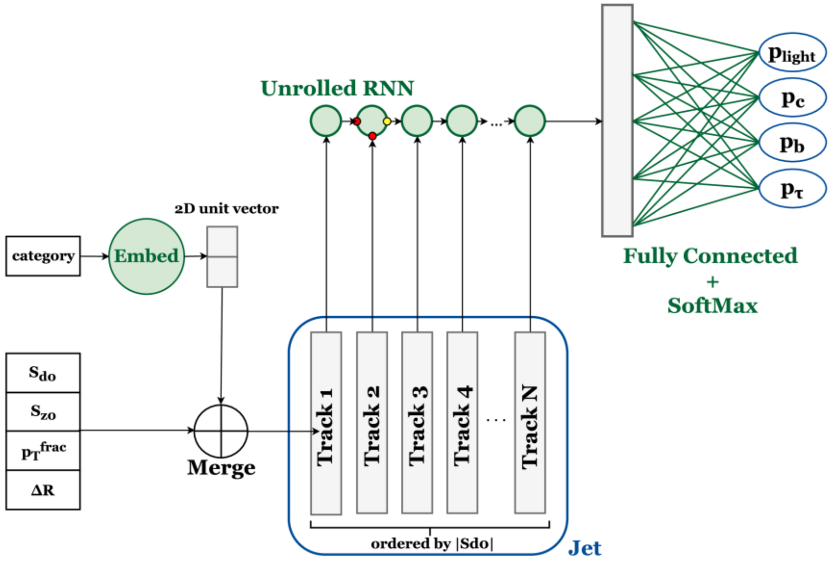
Mostly due to the ability of neural networks to naturally cope with high dimensions

Let's use an example…

$$\sum_{i=1}^{N} \log\left(\frac{p_b}{p_u}\right)$$







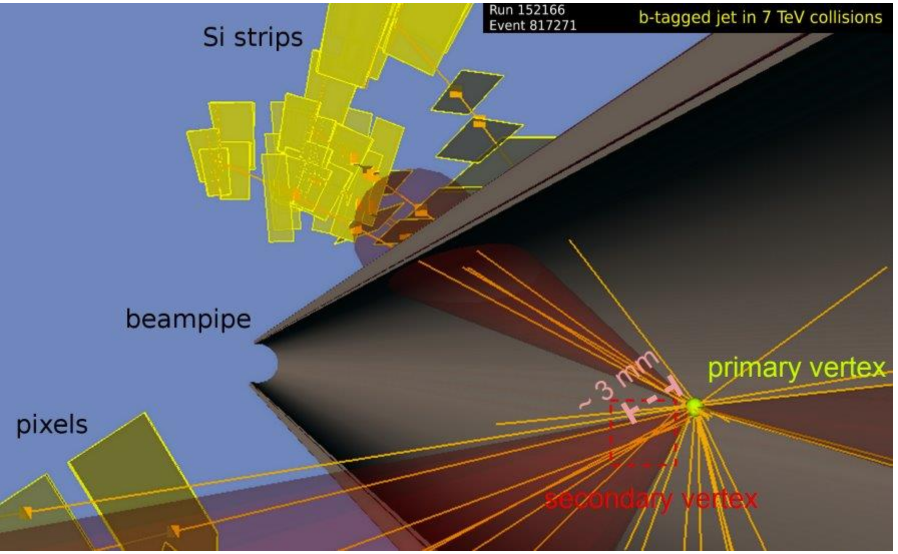
For a jet with N tracks, compute N time 1D probability

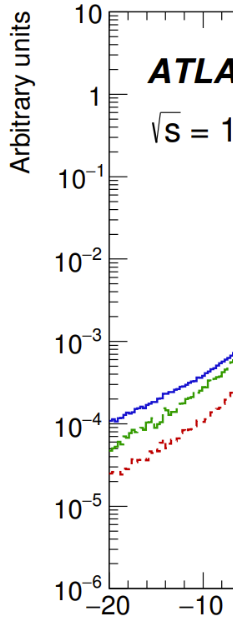
A NN is just a multi-dimensional template. Run 1 time, in N dimesions

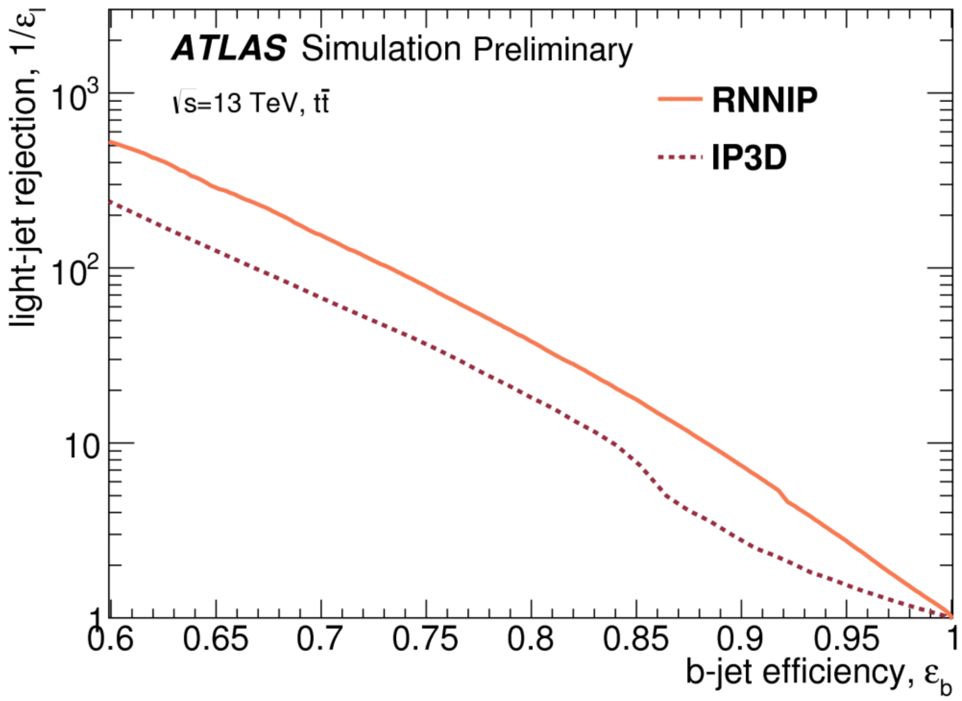
Why is ML useful for HEP?

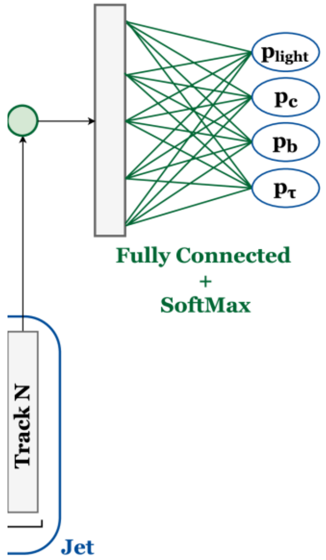
Mostly due to the ability of neural networks to naturally cope with high dimensions

Let’s use an example…









For a jet with N tracks, compute N time 1D probability

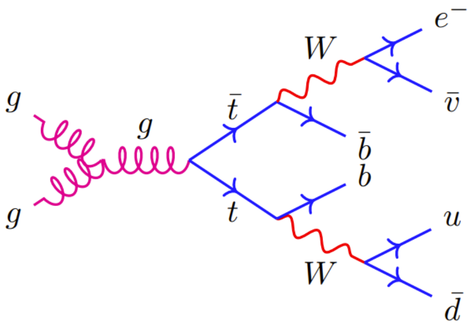
A NN is just a multi-dimensional template. Run 1 time, in N dimesions

From experiment to theory

We use complex, untractable, chains to get p(theta|x), with theta being the theory POI to be measured, and x are the experimental data from our detectors.

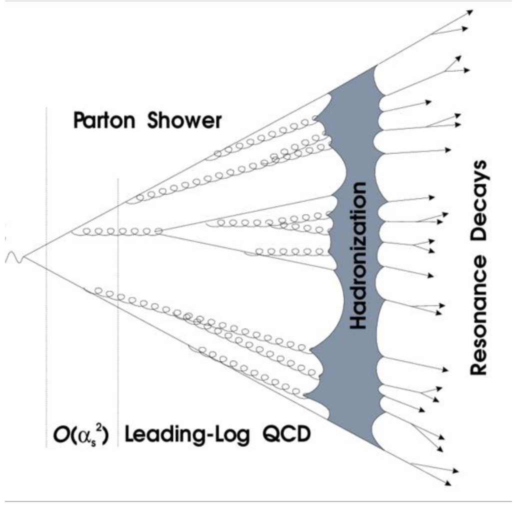
ML applications are proposed, for different purposes, in each step of this chain

Generation:
matrix element



$p(z_p|\text{theta})$

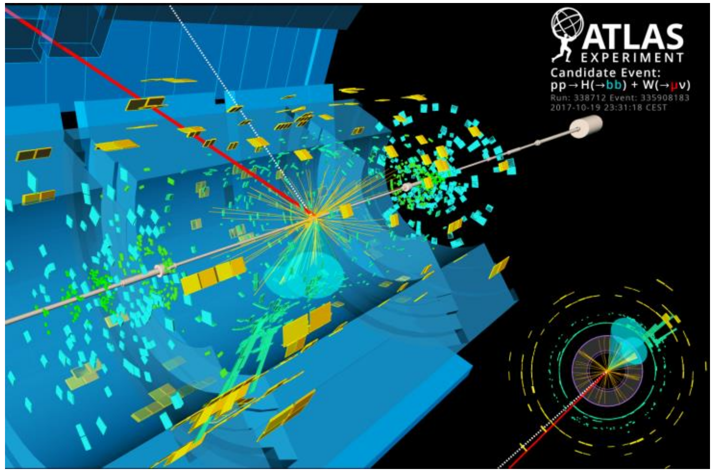
Hadronization,
parton shower



$p(z_r|\ z_p)$

Simulation, reconstruction

$$p(x|\theta) = \int dz \;\; p(x|z_h)\; p(z_h|z_p)\; p(z_p|\theta)$$

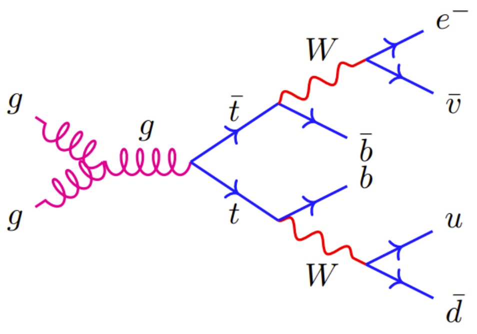


$$x \sim p(x|\theta)$$

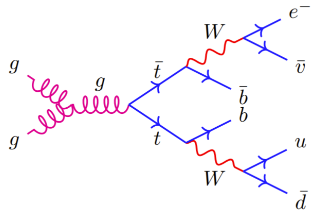
Event generation with ML

Matrix element generation is the first step. ME simulations, especially higher orders, are computationally expensive, can ML be of help?

Generation:
matrix
element

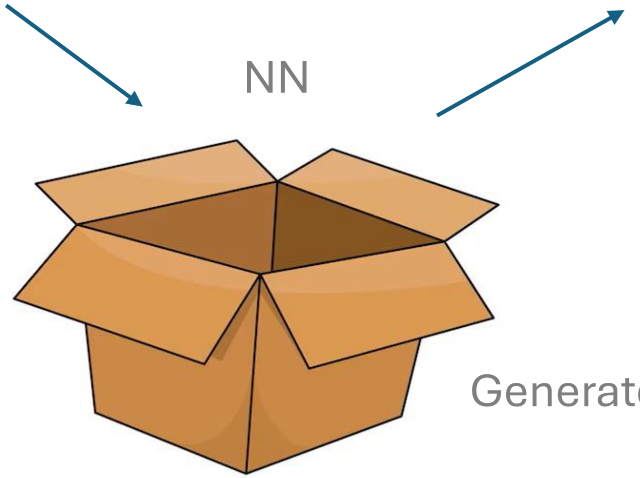


$p(z_e|theta)$

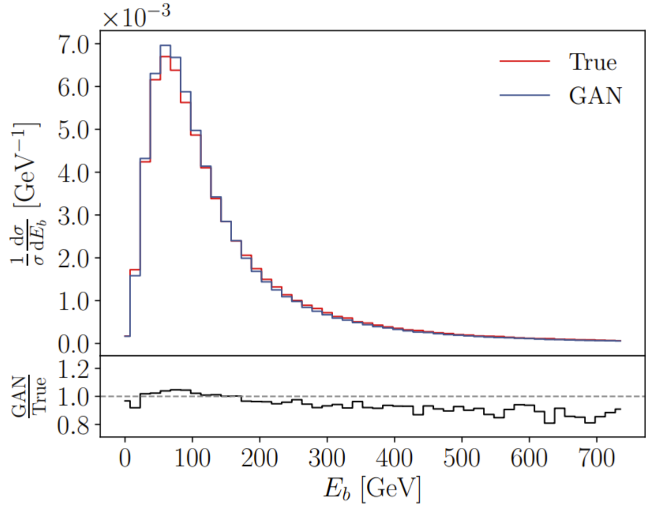


NN

Train



Generate

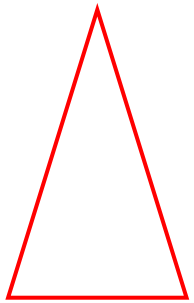


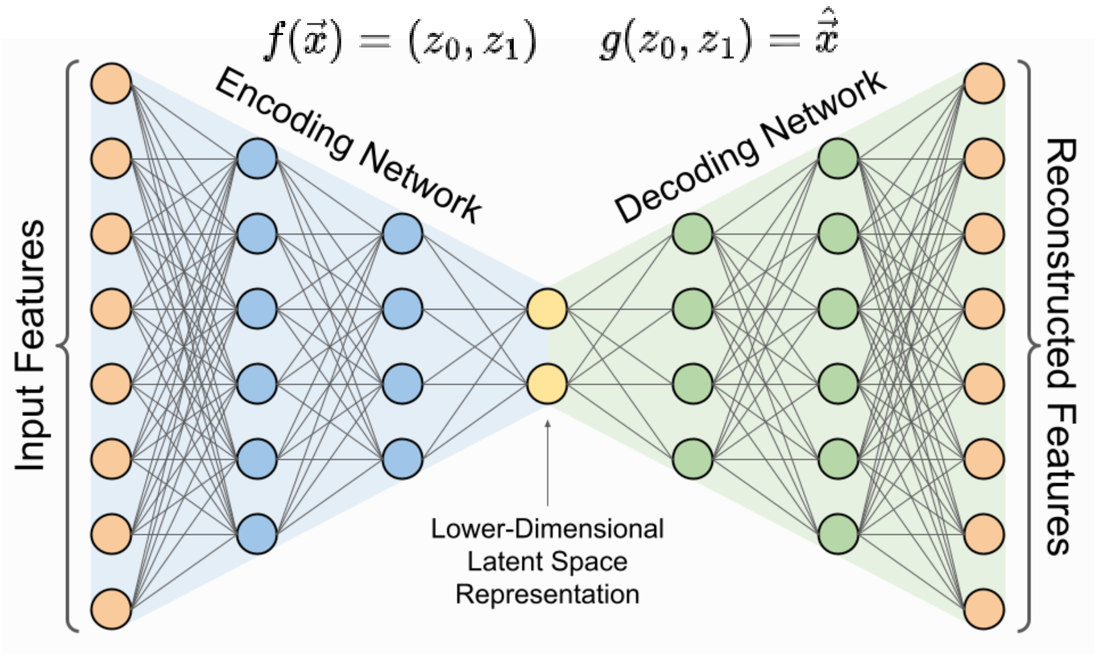
A NN as a high-dimensional look-up table

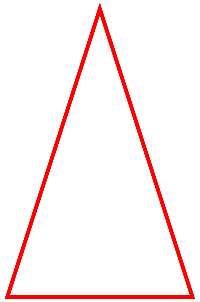
The idea of generative models

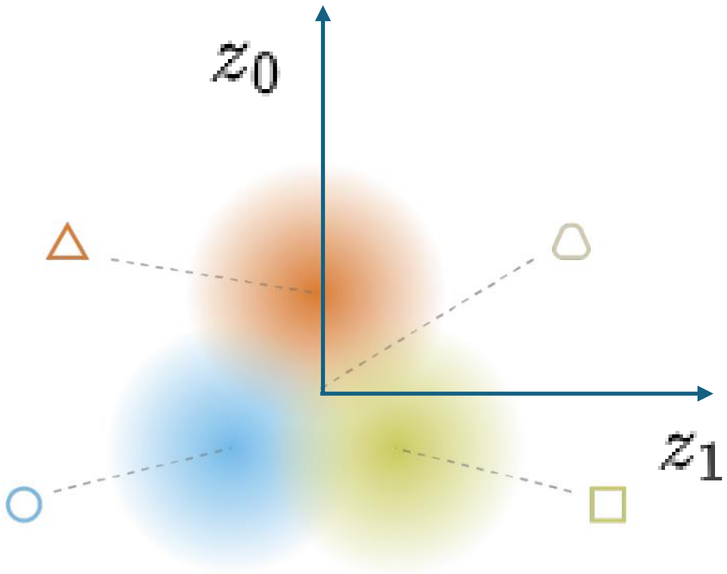
Many models nowadays in the market: variational-autoeconders (VAE), generative adversial networks, diffusion models, normalizing flows…

The most intuitive way to understand generation with NN is with VAE









Trained hidden space

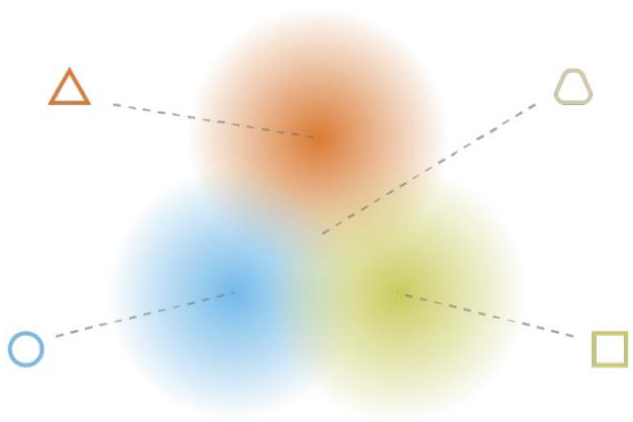
Basically a mapping into a latent, or hidden, state

The idea of generative models

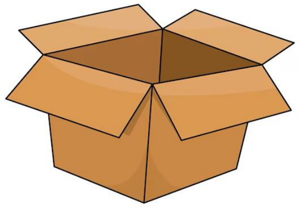
Many models nowadays in the market: variational-autoeconders (VAE), generative adversial networks, diffusion models, normalizing flows…

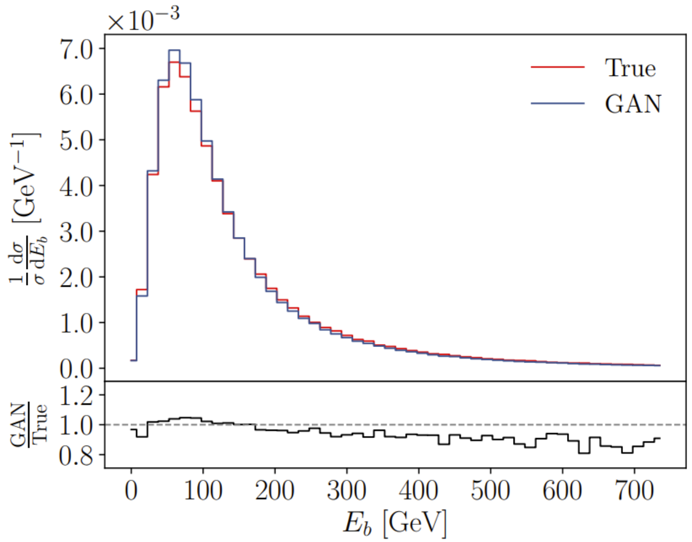
The most intuitive way to understand generation with NN is with VAE

Sample a pair from here



$$g(z_0, z_1) = \hat{\vec{x}}$$





Open questions:

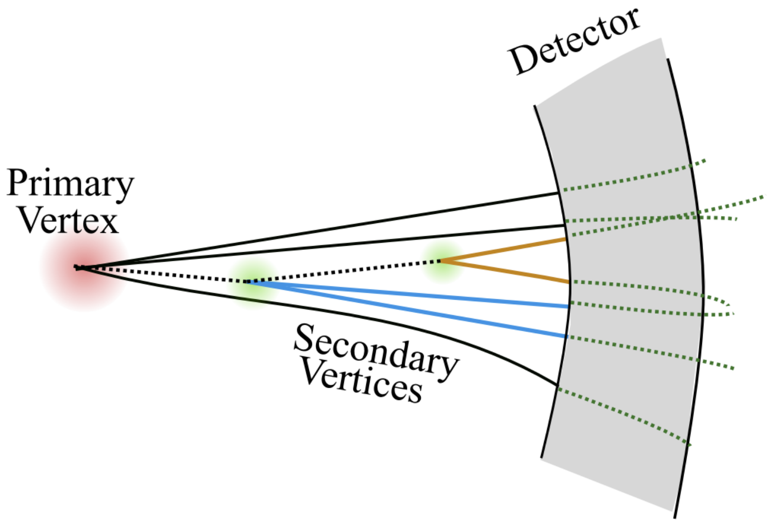
1. If I trained with N events, up to which M can I realible generate? [2409.16336]
2. How can I define a metric that control the accuracy of the generation models in high dimensions?

Improving object reconstruction with ML tools

Arguably, where the impact of ML techniques are having the largest impact

Identification of jets taken as an example, we want to solve 3 problems:

1. Given a jet, what is the probability that the jet originates from a b-quark? Set-to-Float



Probability

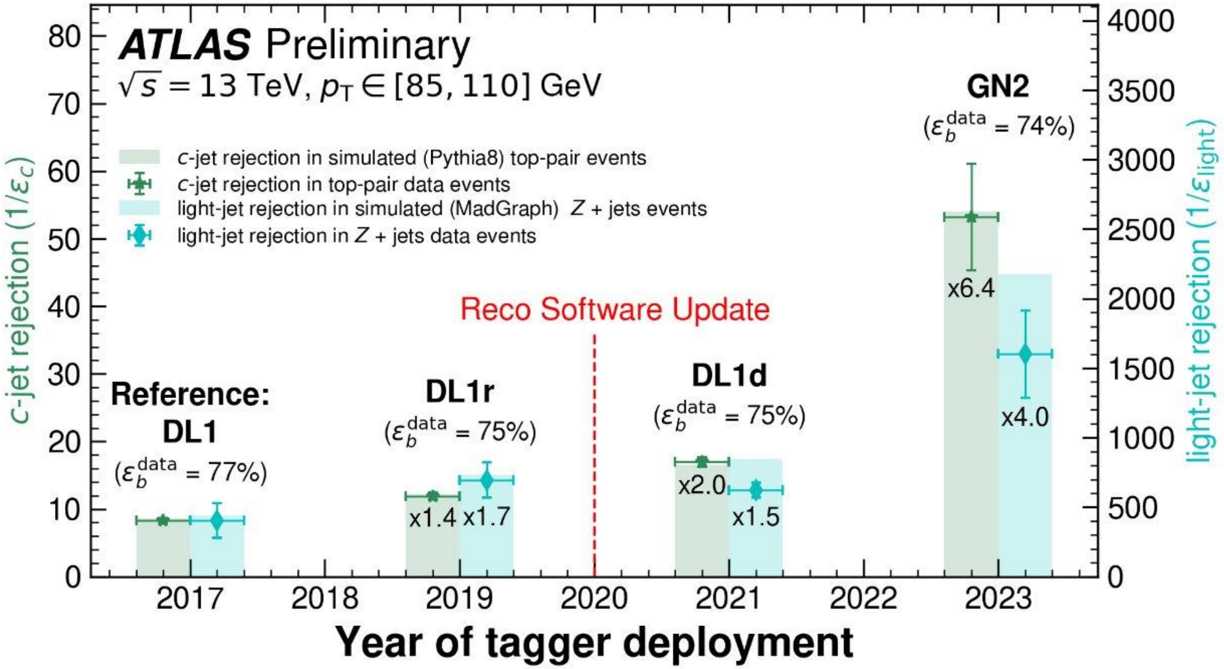
2008.02831 (arxiv.org)

Improving object reconstruction with ML tools

Arguably, where the impact of ML techniques are having the largest impact

Identification of jets taken as an example, we want to solve 3 problems:

1. Given a jet, what is the probability that the jet originates from a b-quark? Set-to-Float



The main point is about the performance generalize to data

There is a priori no insurance that in data it will work well. It is working so far... but we are trying to improve

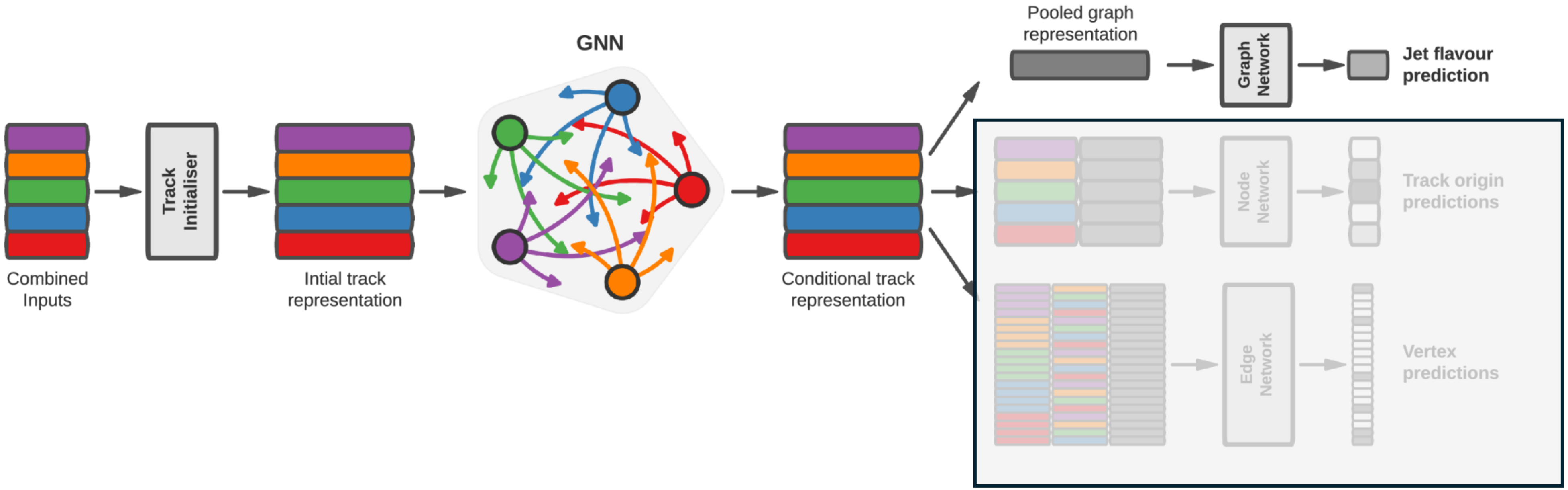
ML to improve reconstruction of physics objects

Arguably, where the impact of ML techniques are shaping the field

Identification of jets taken as an example, we want to solve 3 problems:

1. Given a jet, what is the probability that the jet originates from a b-quark? Set-to-Float

What is it a transfomer? It is a stack of standard NN, similar to building an electronic circuits...



Graph Neural Network Jet Flavour Tagging with the ATLAS Detector (cern.ch)

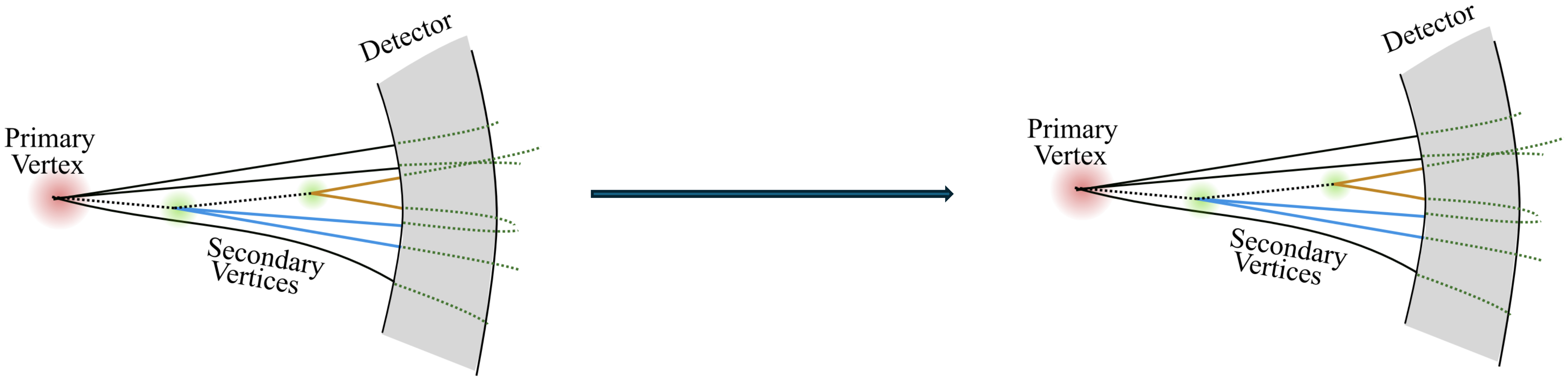
ML to improve reconstruction of physics objects

Arguably, where the impact of ML techniques are shaping the field

Identification of jets taken as an example, we want to solve 3 problems:

1. Given a jet, what is the probability that the jet originates from a b-quark? Set-to-Float

2. Given the jet constituents, what is the probability that each of them comes from a B hadron weakly decaying? Set-to-Vector



Probability that each single track is associated to a common vertex

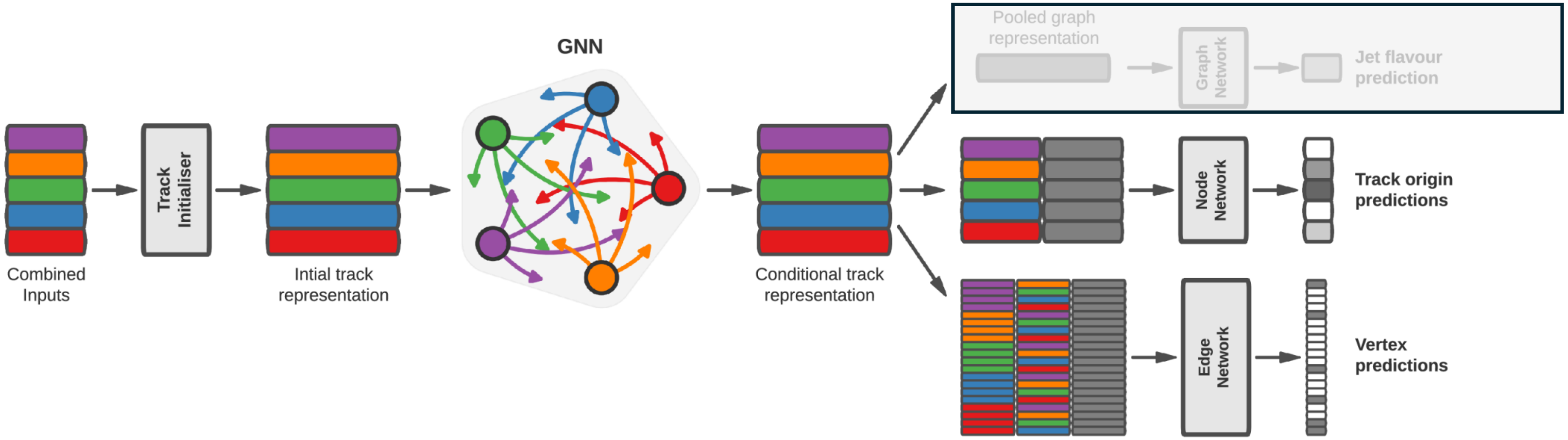
ML to improve reconstruction of physics objects

Arguably, where the impact of ML techniques are shaping the field

Identification of jets taken as an example, we want to solve 3 problems:

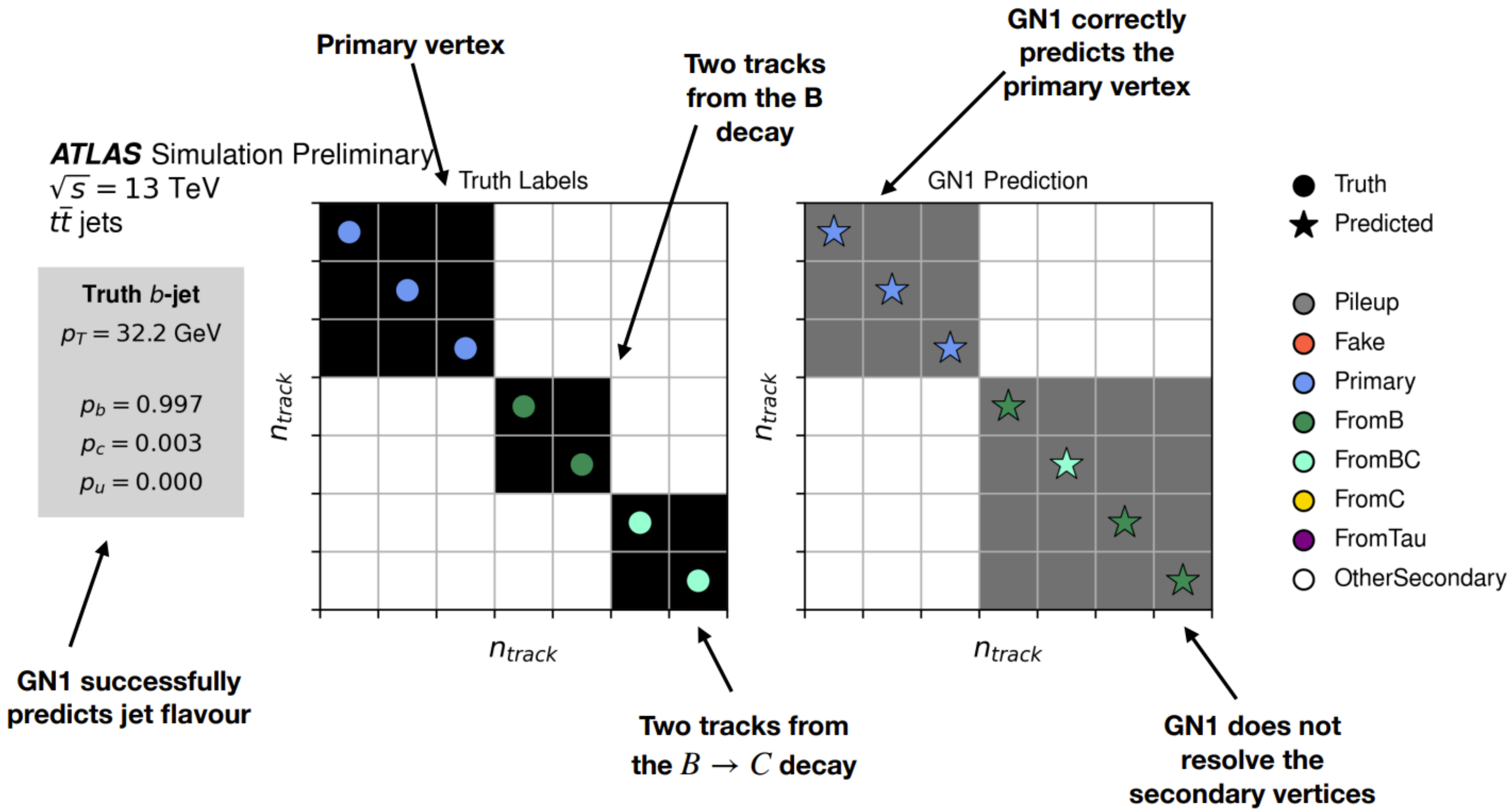
1. Given a jet, what is the probability that the jet originates from a b-quark? Set-to-Float

2. Given the jet constituents, what is the probability that each of them comes from a B hadron weakly decaying? Set-to-Vector



Some (personal) thoughts on interpretability

Interpretability of a ML tool can mean many things (explainable AI etc...). But I want to focus solely on one question: can we build a model that help us understand what has being learned (up to some limit)?

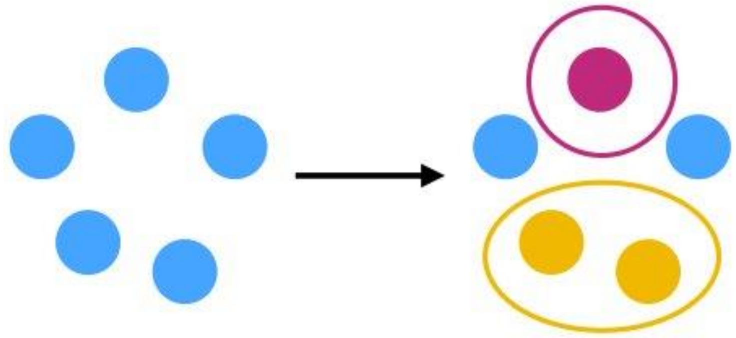


ML to improve reconstruction of physics objects

Arguably, where the impact of ML techniques are shaping the field

Identification of jets taken as an example, we want to solve 3 problems:

3. Given the two points above, can I reconstruct the vertices inside the jet? Set-to-Set



This is an area of very active developements…
Two complementary approaches:

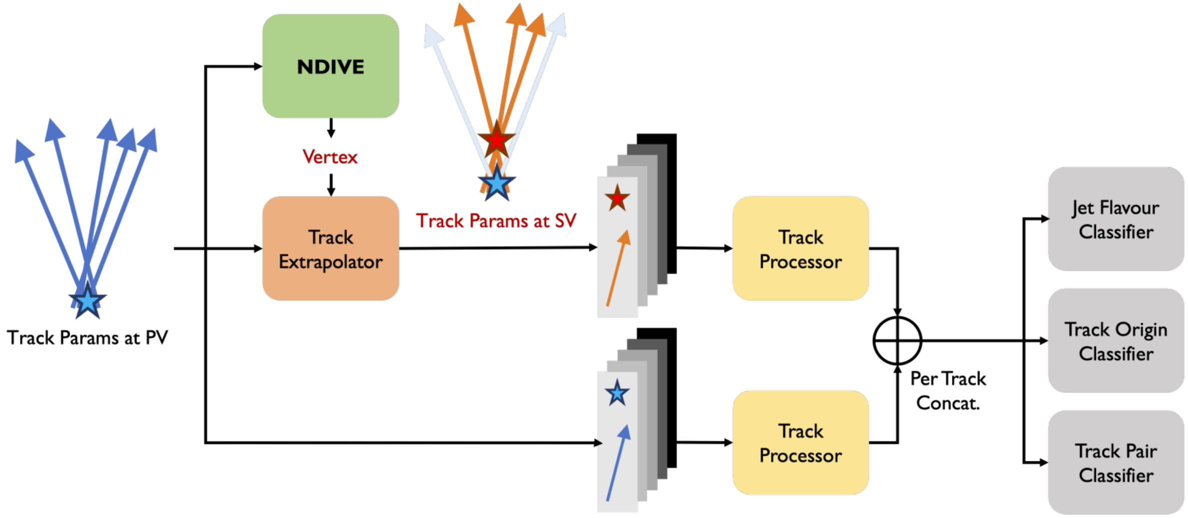
1. Include physics information in the training: differentiable programming
2. Physics-informed neural network

Differentiable programming as a tool to inject physics information

The first example of this is actually from theory: NNPDF
Building on the same principle, physics information can be included directly into NNs

$$\mathcal{L} = Classification + \mathcal{S}$$

$$\mathcal{S} = \sum_{i=1}^{N} w_i (\mathbf{q}_i - \mathbf{h}_i(\mathbf{v}, \mathbf{p}_i))^T \mathbf{V}_i^{-1} (\mathbf{q}_i - \mathbf{h}_i(\mathbf{v}, \mathbf{p}_i)),$$



Introduce explicitely physics knowledge into the network

Helps interpretability, convergence, performance and robustness.

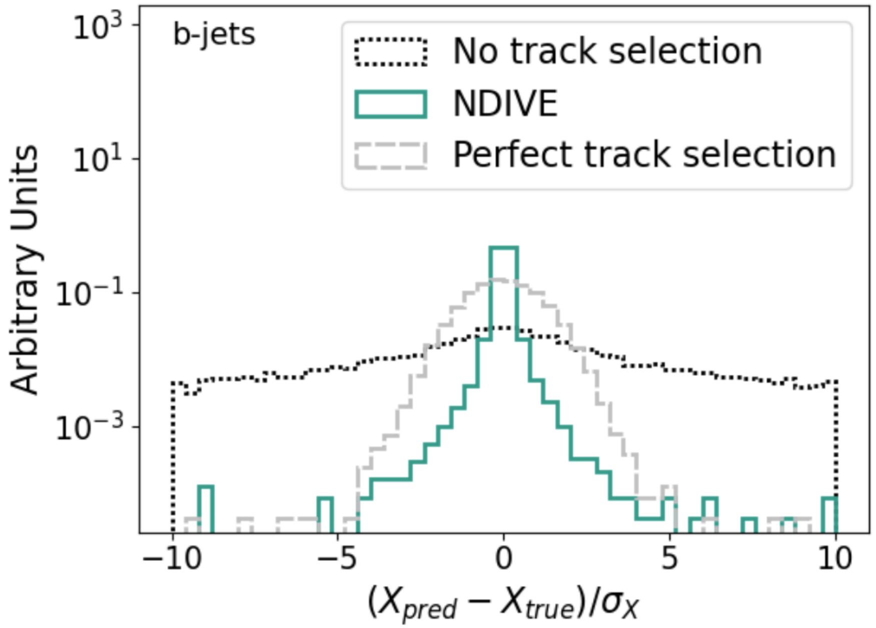
Open question: at some point the network can learn this anyway, what is this limit?

Differentiable programming as a tool to inject physics information

The first example of this is actually from theory: NNPDF
Building on the same principle, physics information can be included directly into NNs

$$\mathcal{L} = Classification + \mathcal{S}$$

$$\mathcal{S} = \sum_{i=1}^{N} w_i (\mathbf{q}_i - \mathbf{h}_i(\mathbf{v}, \mathbf{p}_i))^T \mathbf{V}_i^{-1} (\mathbf{q}_i - \mathbf{h}_i(\mathbf{v}, \mathbf{p}_i)),$$

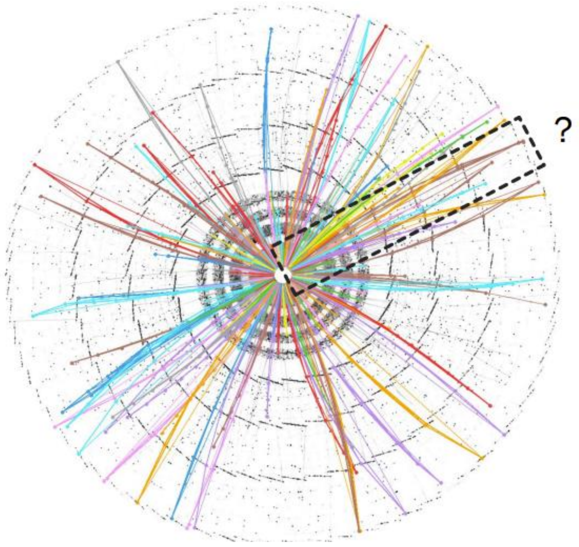


Post-tagging, one can look at the secondary vertex, and its covariance!

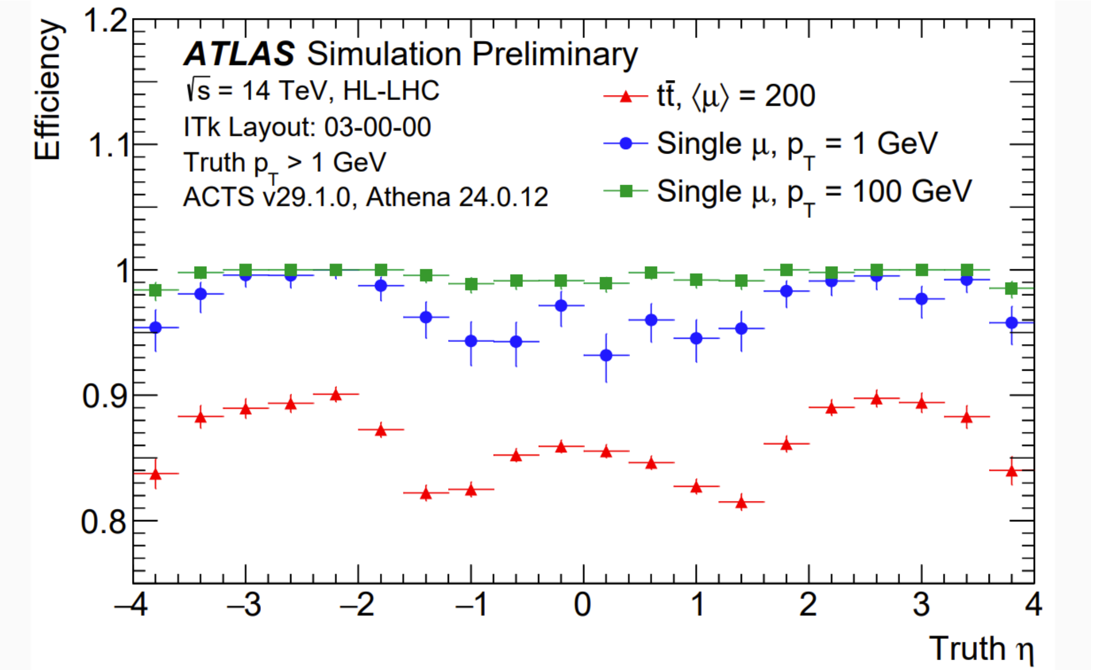
More on set-to-set problems

Reconstruction, due to efficiencies, mistags, resolutions of the detector is effectively a set-to-set task.
Major efforts are under: tracking and global particle flow-reconstruction

Tracking: from detector hits to tracks



Work in progress to compare to standard performance



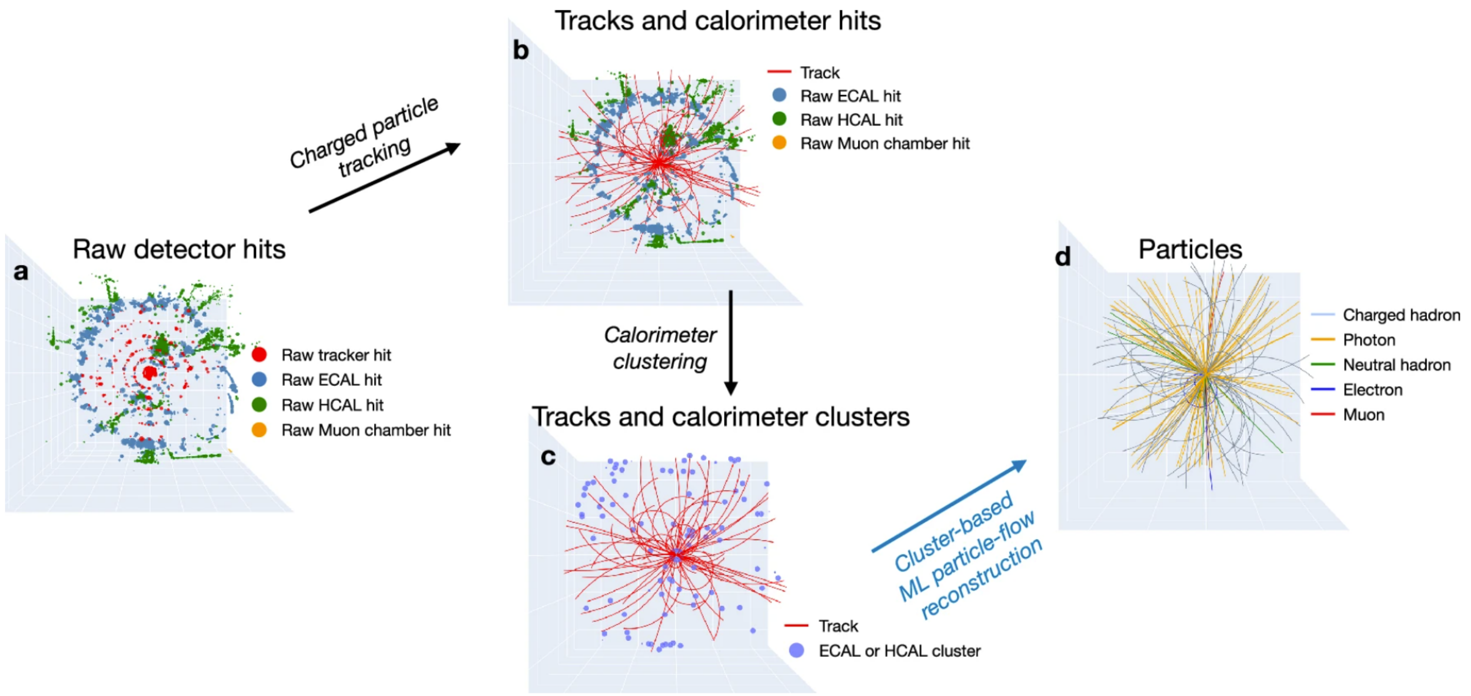
[Finding and Fitting Public.pdf (cern.ch)]

[ATL-ITK-PROC-2022-006.pdf (cern.ch)]

[Status of the ACTS Integration for Phase-II ATLAS Track Reconstruction (cern.ch)]

A closer look at particle-flow

Global particle flow algorithms: given the set of detector hits, reconstruct the final state particles (pions, kaons, etc…)



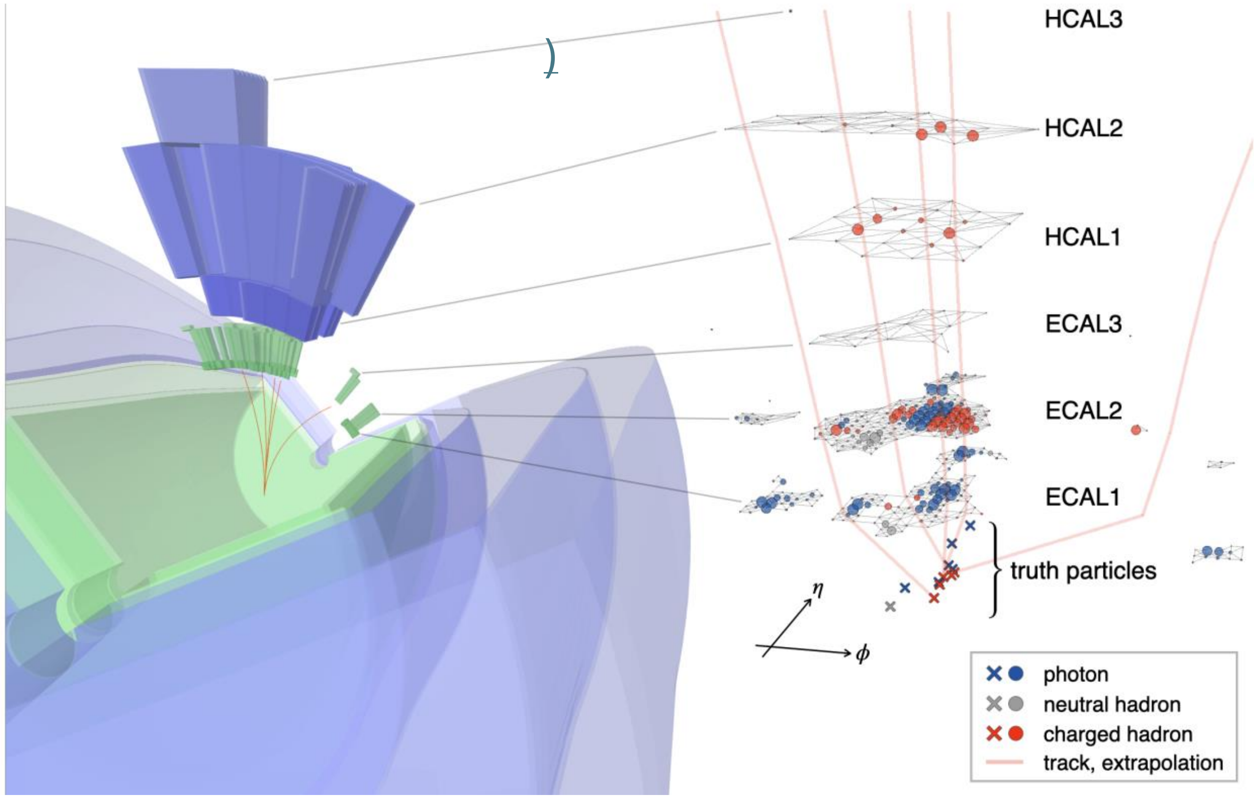
[Improved particle-flow event reconstruction with scalable neural networks for current and future particle detectors | Communications Physics (nature.com)]

A closer look at particle-flow

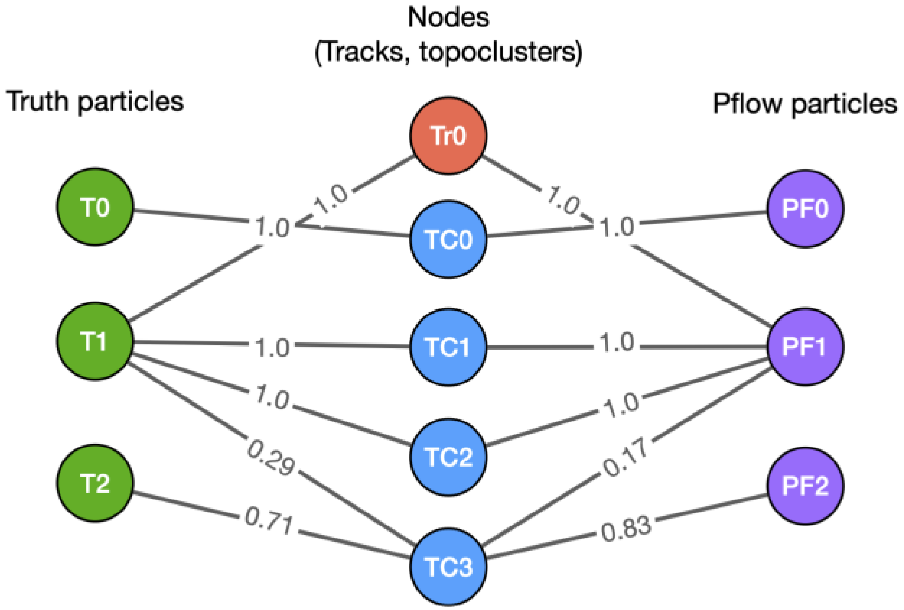
hypergraphs

Global particle flow algorithms: given the set of detector hits, reconstruct the final state particles (pions, kaons, etc...)

From detector hits, to graphs



Reconstructng matrix elements



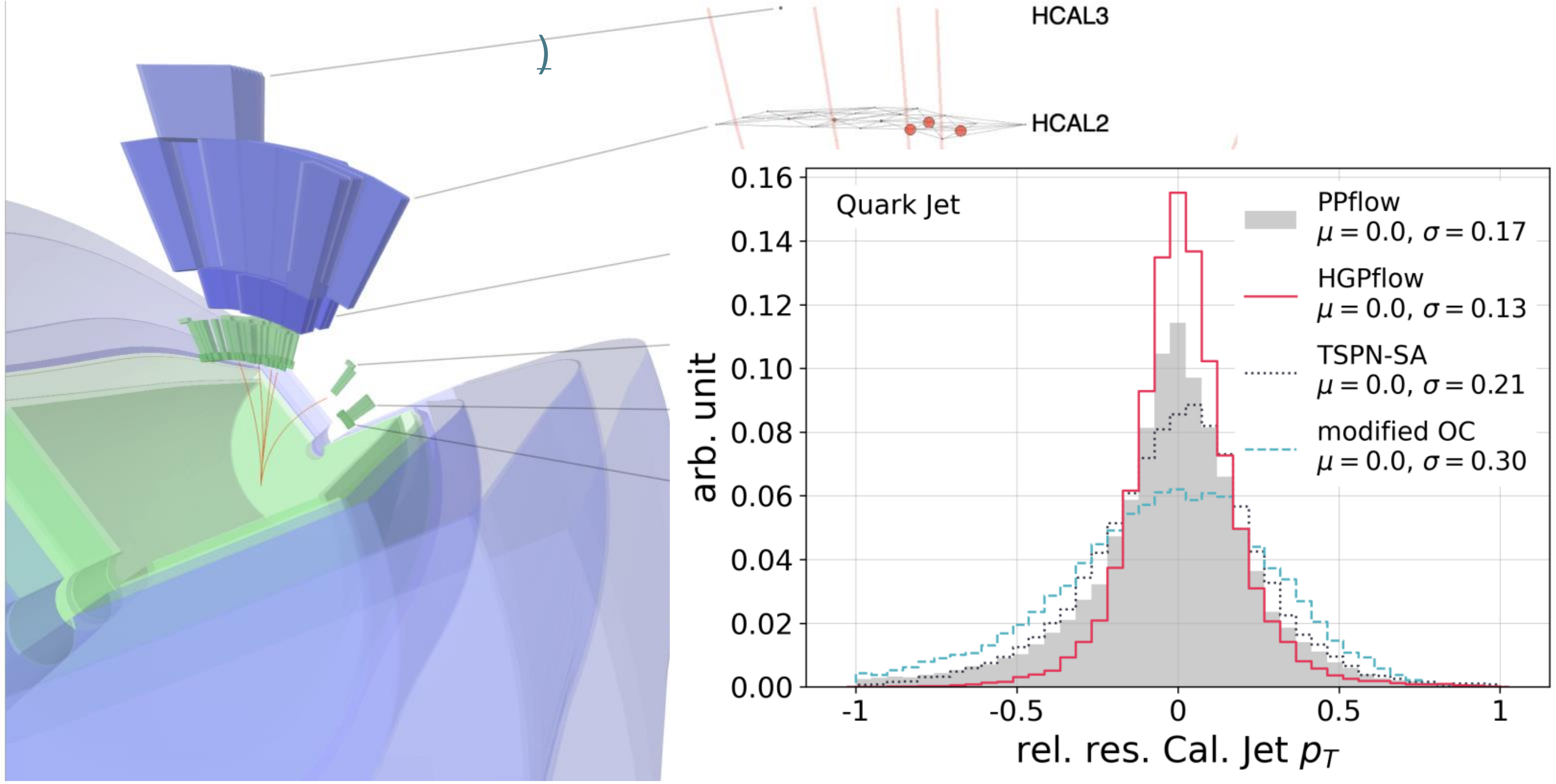
A closer look at particle-flow

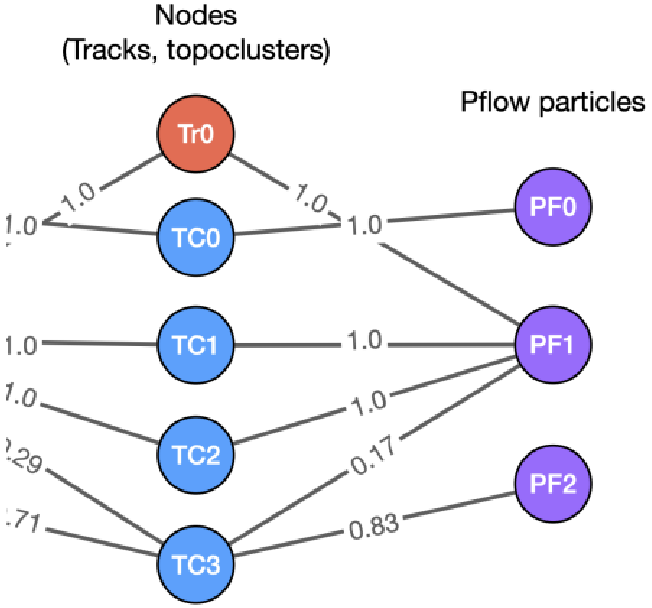
hypergraphs

Global particle flow algorithms: given the set of detector hits, reconstruct the final state particles (pions, kaons, etc...)

From detector hits, to graphs

Reconstructng matrix elements



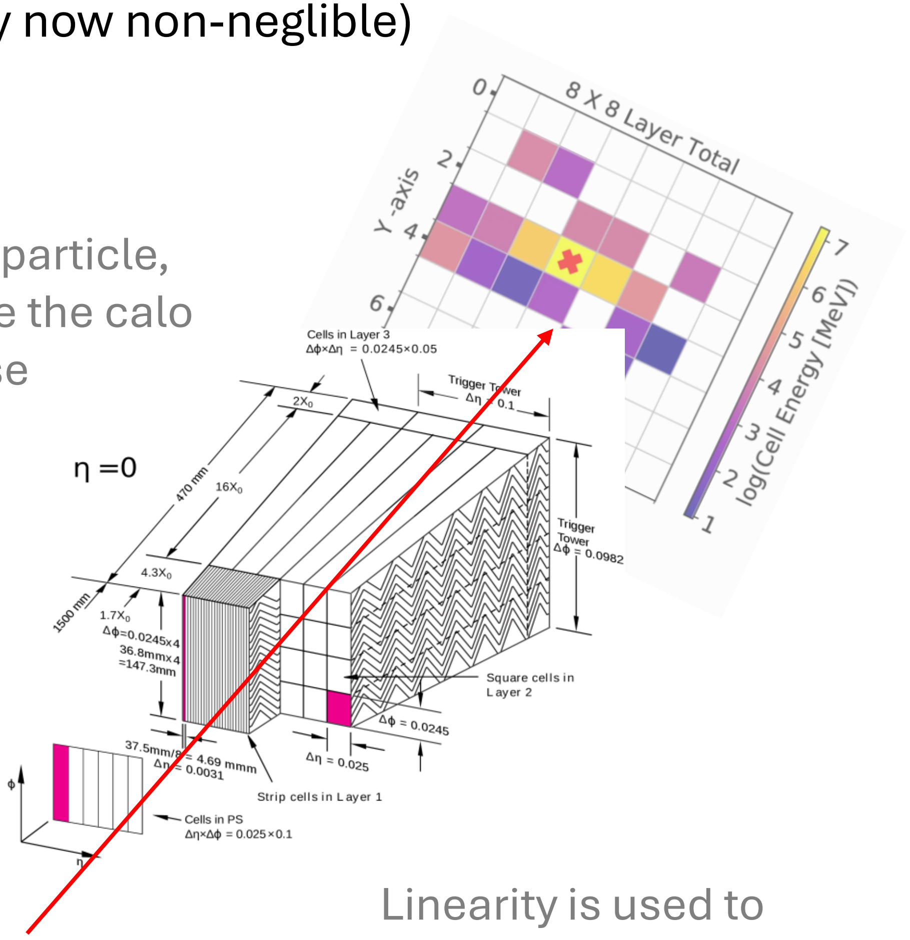


Event generation at reconstruction: conditional event generation

Calorimeter simulation is the most CPU expensive task in experiments, a real concern at HL-LHC

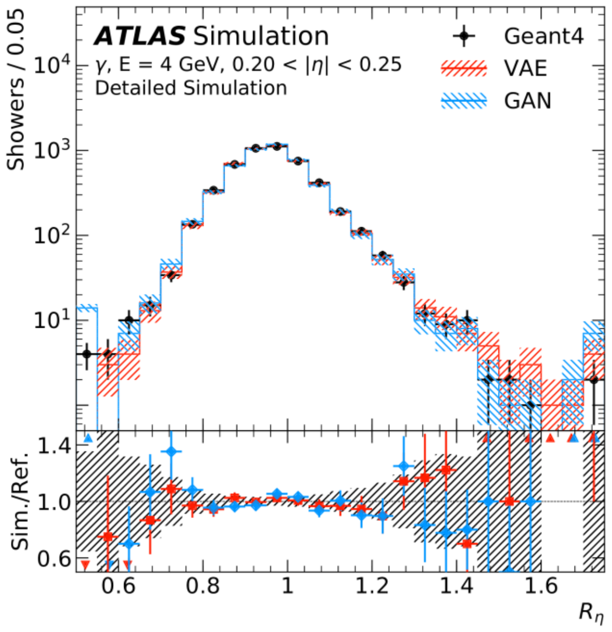
We need high statistics to reduce our systematics on the avilable MC stat templates (that often case are already now non-neglible)

Given a particle, generate the calo response



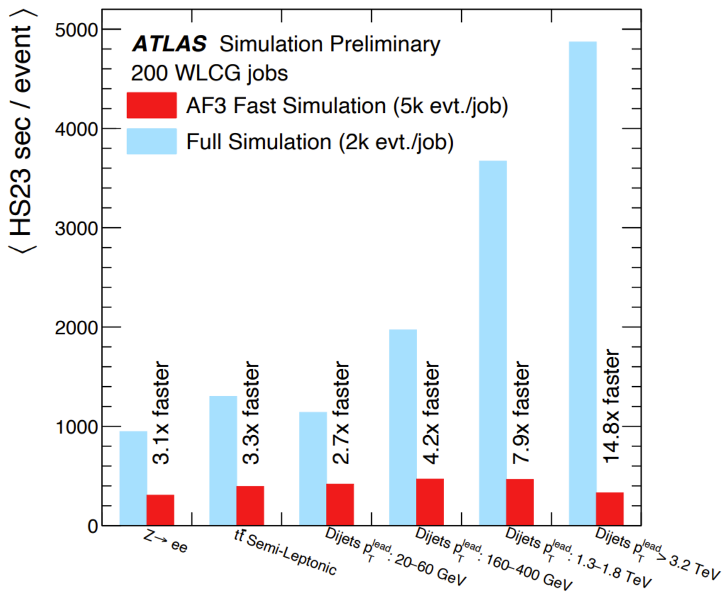
Incoming particle

Linearity is used to get the full event



Energy ratio of calo layers

SIM-2023-005



CPU performance gain

Comput Softw Big Sci 8, 7 (2024)

A look into the future, towards foundation models?

Foundation models: one take it all

ChatGBT and similar… generate text, images, it traslates, many tasks into a single network. To what this tralsate in terms of physics application?

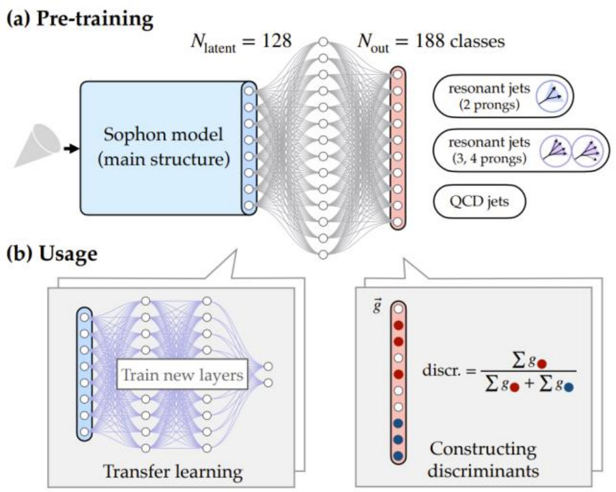


TABLE I. Summary of the 188 jet labels in the JetClass-II dataset.

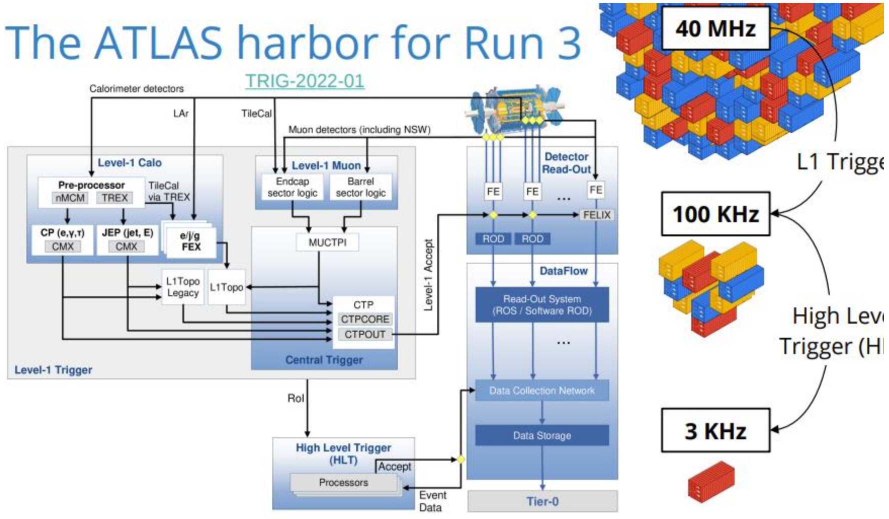
Major types	Index range	Label names
Resonant jets: $X \to 2$ prong	0–14	bb, cc, ss, qq, bc, cs, bq, cq, sq, gg, ee, $\mu\mu$, $\tau_\mathrm{h}\tau_e$, $\tau_\mathrm{h}\tau_\mu$, $\tau_\mathrm{h}\tau_\mathrm{h}$
Resonant jets: $X \to 3$ or 4 prong	15–160	$bbbb$, $bbcc$, $bbss$, $bbqq$, $bbgg$, $bbee$, $bb\mu\mu$, $bb\tau_\mathrm{h}\tau_e$, $bb\tau_\mathrm{h}\tau_\mu$, $bb\tau_\mathrm{h}\tau_\mathrm{h}$, bbb, bbc, bbs, bbq, bbg, bbe, $bb\mu$, $cccc$, $ccss$, $ccqq$, $ccgg$, $ccee$, $cc\mu\mu$, $cc\tau_\mathrm{h}\tau_e$, $cc\tau_\mathrm{h}\tau_\mu$, $cc\tau_\mathrm{h}\tau_\mathrm{h}$, ccb, ccc, ccs, ccq, ccg, cce, $cc\mu$, $ssss$, $ssqq$, $ssgg$, $ssee$, $ss\mu\mu$, $ss\tau_\mathrm{h}\tau_e$, $ss\tau_\mathrm{h}\tau_\mu$, $ss\tau_\mathrm{h}\tau_\mathrm{h}$, ssb, ssc, sss, ssq, ssg, sse, $ss\mu$, $qqqq$, $qqgg$, $qqee$, $qq\mu\mu$, $qq\tau_\mathrm{h}\tau_e$, $qq\tau_\mathrm{h}\tau_\mu$, $qq\tau_\mathrm{h}\tau_\mathrm{h}$, qqb, qqc, qqs, qqq, qqg, qqe, $qq\mu$, $gggg$, $ggee$, $gg\mu\mu$, $gg\tau_\mathrm{h}\tau_e$, $gg\tau_\mathrm{h}\tau_\mu$, $gg\tau_\mathrm{h}\tau_\mathrm{h}$, ggb, ggc, ggs, ggq, ggg, gge, $gg\mu$, bee, cee, see, qee, gee, $b\mu\mu$, $c\mu\mu$, $s\mu\mu$, $q\mu\mu$, $g\mu\mu$, $b\tau_\mathrm{h}\tau_e$, $c\tau_\mathrm{h}\tau_e$, $s\tau_\mathrm{h}\tau_e$, $q\tau_\mathrm{h}\tau_e$, $g\tau_\mathrm{h}\tau_e$, $b\tau_\mathrm{h}\tau_\mu$, $c\tau_\mathrm{h}\tau_\mu$, $s\tau_\mathrm{h}\tau_\mu$, $q\tau_\mathrm{h}\tau_\mu$, $g\tau_\mathrm{h}\tau_\mu$, $b\tau_\mathrm{h}\tau_\mathrm{h}$, $c\tau_\mathrm{h}\tau_\mathrm{h}$, $s\tau_\mathrm{h}\tau_\mathrm{h}$, $q\tau_\mathrm{h}\tau_\mathrm{h}$, $g\tau_\mathrm{h}\tau_\mathrm{h}$, $qqqb$, $qqqc$, $qqqs$, $bbcq$, $ccbs$, $ccbq$, $ccsq$, $sscq$, $qqbc$, $qqbs$, $qqcs$, $bcsq$, bcs, bcq, bsq, csq, $bce\nu$, $cse\nu$, $bqe\nu$, $cqe\nu$, $sqe\nu$, $qqe\nu$, $bc\mu\nu$, $cs\mu\nu$, $bq\mu\nu$, $cq\mu\nu$, $sq\mu\nu$, $qq\mu\nu$, $bc\tau_e\nu$, $cs\tau_e\nu$, $bq\tau_e\nu$, $cq\tau_e\nu$, $sq\tau_e\nu$, $qq\tau_e\nu$, $bc\tau_\mu\nu$, $cs\tau_\mu\nu$, $bq\tau_\mu\nu$, $cq\tau_\mu\nu$, $sq\tau_\mu\nu$, $qq\tau_\mu\nu$, $bc\tau_\mathrm{h}\nu$, $cs\tau_\mathrm{h}\nu$, $bq\tau_\mathrm{h}\nu$, $cq\tau_\mathrm{h}\nu$, $sq\tau_\mathrm{h}\nu$, $qq\tau_\mathrm{h}\nu$
QCD jets	161–187	$bbccss$, $bbccs$, $bbcc$, $bbcss$, $bbcs$, bbc, $bbss$, bbs, bb, $bccss$, $bccs$, bcc, $bcss$, bcs, bc, bss, bs, b, $ccss$, ccs, cc, css, cs, c, ss, s, others

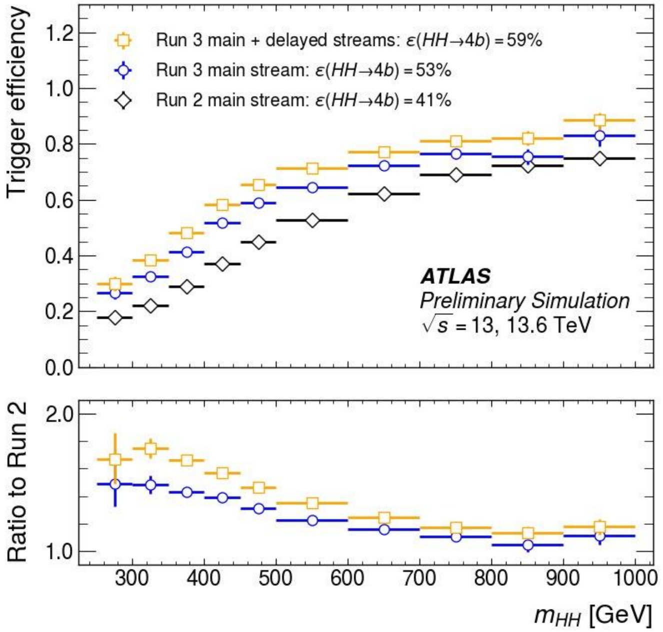
FTAG_ML_20240911_H_Qu (cern.ch)

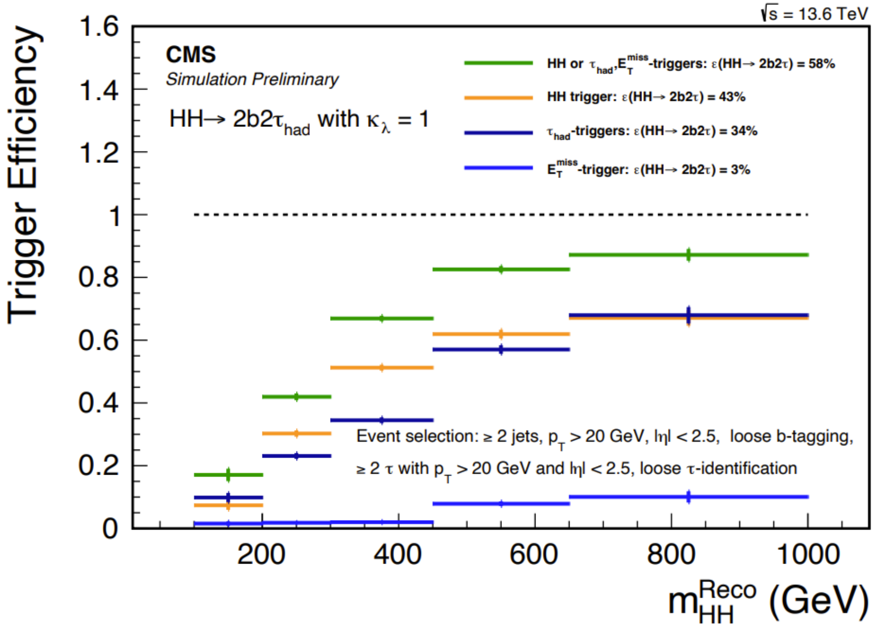
Accelerating Resonance Searches via Signature-Oriented Pre-training (2024)

All of this… but faster, much faster!

Machine learning improvements propagating also at trigger level (crucial for HL-LHC)







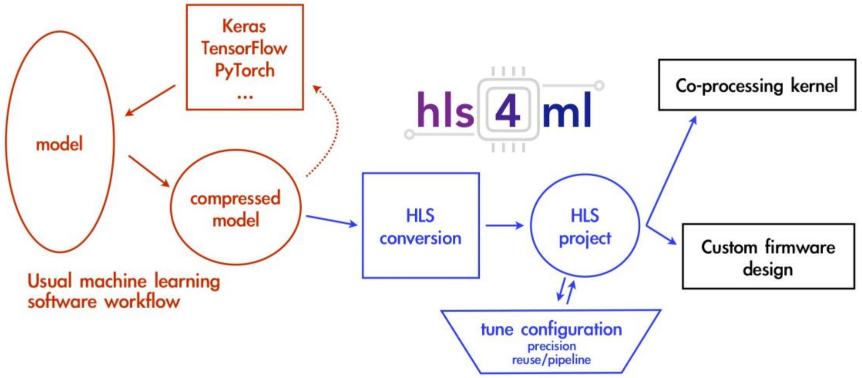
[trigger]

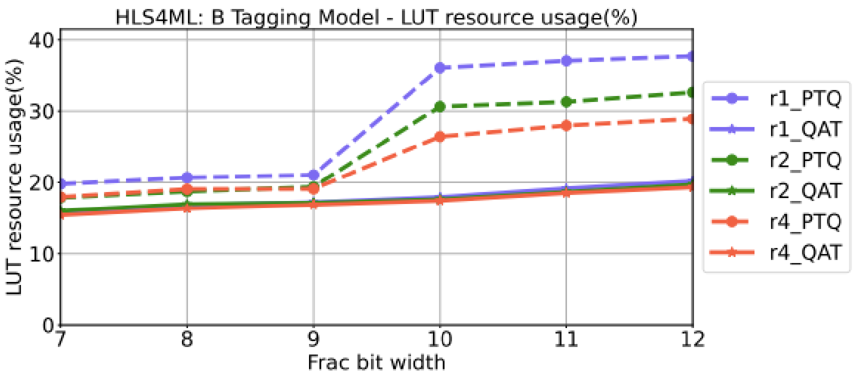
[CMS trigger]

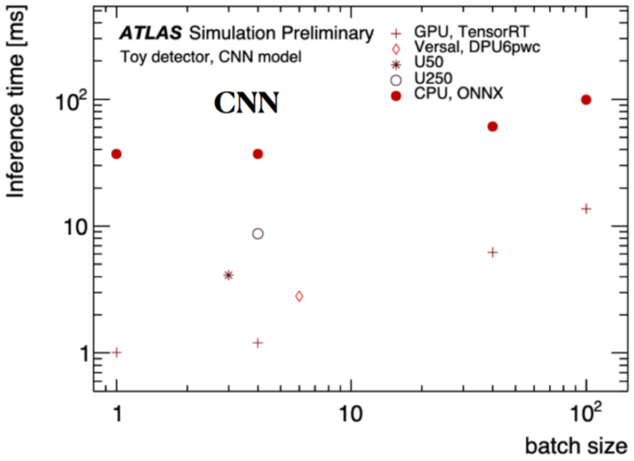
All of this… but faster, much faster!

At lvl0 need an inference time O(10 micro-seconds), need heterogenous farm…

HLS4ML 2409.05207 (arxiv.org) muons ICHEP muons FPGA





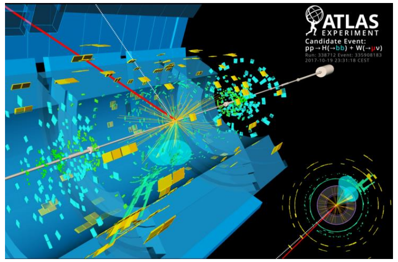


We have a measurement, let's go back to theory: Unfolding

- ML can help producing an unbinned, multidimensional unfolded measurements
- Large literature available: invertible neural networks etc…
- Literature is also growing in trying to tackle the uncertainty related to the usage of ML methods

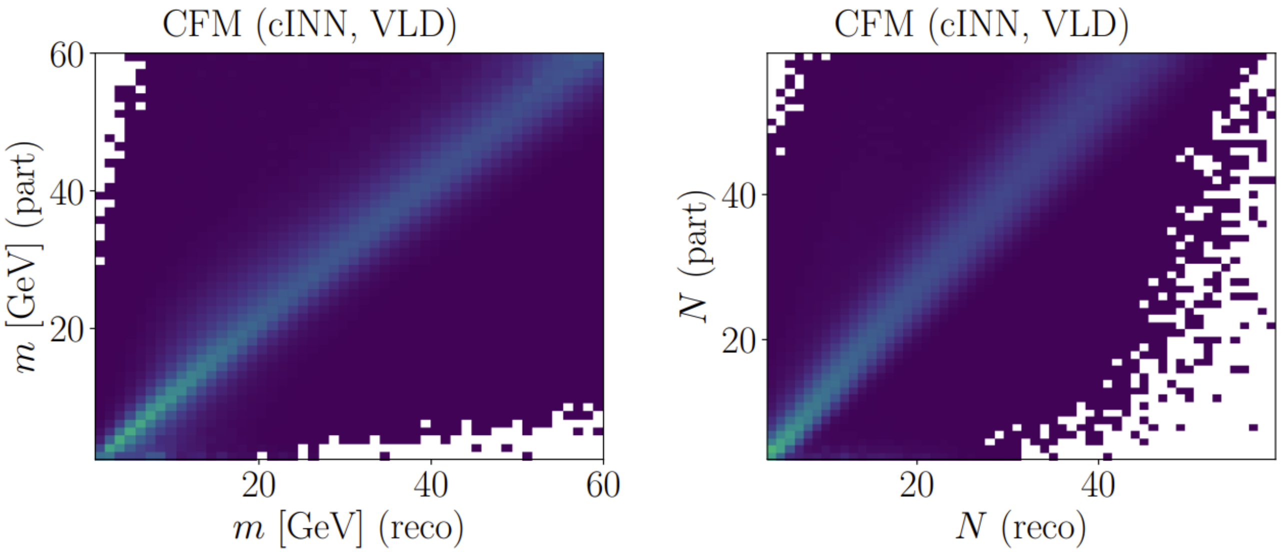
Reconstruction+data analysis

p(theta)



p(theta|x)

Unfolding



Reco-truth unfolding matrices: projections

[Modern Machine Learning Tools for Unfolding (2024)]

Conclusions

- Our job is to make the most of the data we collected (and will collect)
- ML has proven to be of great help in handling optimally our data, in few years, we have improved by several factors our performance, not easy to estimate an extrapolation to HL-LHC time and beyond, but I am optimistic
- I also skipped many topics, optimal transport, SBI, Nflows, likelihood free inference…
- The trend nowadays is to explore low-level information, such as tracker hits etc… stay tuned!
- In parallel to keep improving performance, I think we shall spend effort in using more explicitely real data in NN trainings, and make an effort to provide semi-interpretable algorithms

Thank you