Machine Learning and AI in HEP experiments
(with focus on LHC/colliders)

Andrea Rizzi -University/INFN of Pisa

AI@INFN, Bologna, May 3rd 2022

Istituto Nazionale di Fisica Nucleare

UNIVERSITÀ DI PISA

Outline / disclaimers

Istituto Nazionale di Fisica Nucleare

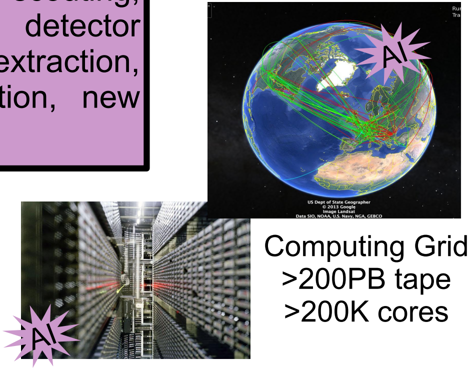
UNIVERSITÀ DI PISA

- Current usage of ML/AI in High Energy Physics and HEP data analyses
  - A few example
- Can we go beyond current usage of ML/AI ?
- Data representation
- Interesting recent developments

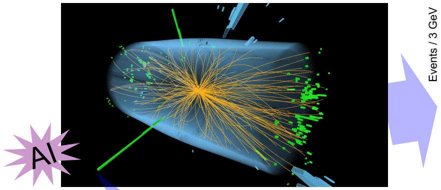
- Disclaimers
  - My experience is mostly limited to CMS/LHC, the presentation has hence a strong bias
  - Credits for some of the material goes to CMS ML conveners and their recent presentation at Jet/MET Workshop in florence

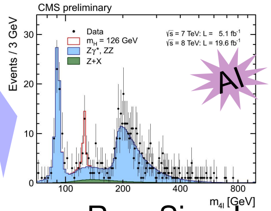
AI in HEP

Role of AI: accelerator control, data acquisition, event triggering, anomaly detection, new physics scouting, event reconstruction, event generation, detector simulation, LHC grid control, analytics, signal extraction, likelihood free inference, background rejection, new physics searches, ...

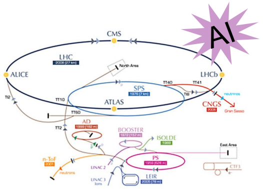

AI
AI
US Dept of State Geographer
© 2013 Google
Image Landsat
Data SIO, NOAA, U.S. Navy, NGA, GEBCO

Computing Grid
>200PB tape
>200K cores

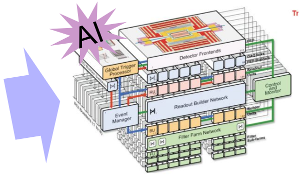




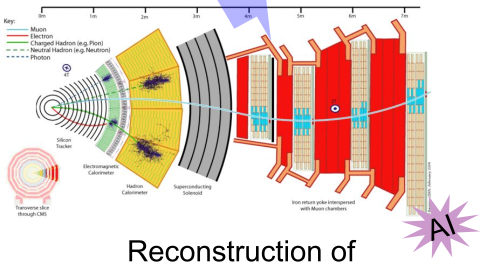
Rare Signal Measurement
~1 out of 10^6



Large Hadron Collider
40 MHz of collision



CMS L1 &
High-Level Triggers
50k cores, 1kHz

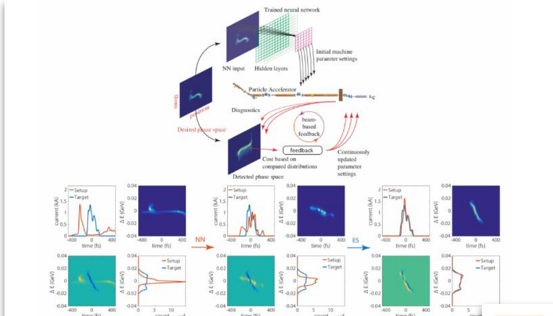

0m
1m
2m
3m
4m
5m
6m
7m
Key:
Muon
Electron
Charged Hadron (e.g. Pion)
Neutral Hadron (e.g. Neutron)
Photon
4T
Silicon Tracker
Electromagnetic Calorimeter
Hadron Calorimeter
Superconducting Solenoid
Iron return yoke interspersed with Muon chambers
Transverse slice through CMS
AI

Reconstruction of different particles and "physics objects"

Applications in HEP

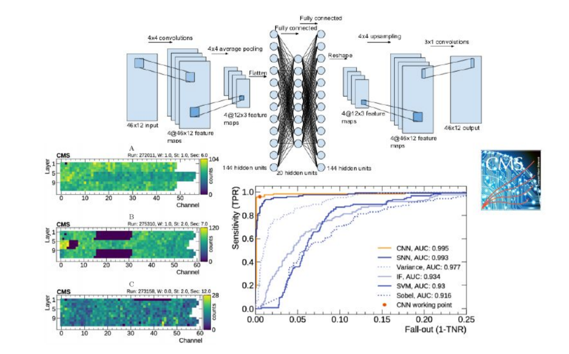
- New ideas discussed in experiments dedicated groups, conferences, workshops
  - A nice “review” https://iml-wg.github.io/HEPML-LivingReview/ (~600 ML in HEP papers listed)

Beam control

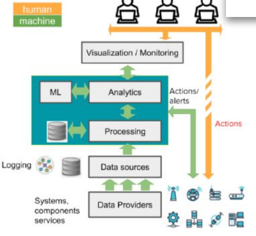


A. Scheinker, C. Emma, A.L. Edelen, S. Gessner [2001.05461]

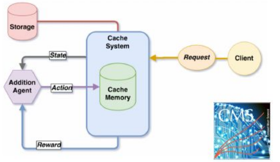
Data Quality Monitoring



A.A. Pol, G. Cerminara, C. Germain, M. Pierini, A. Seth [doi:10.1007/s41781-018-0020-1]



Operational Intelligence [cds:2709338]

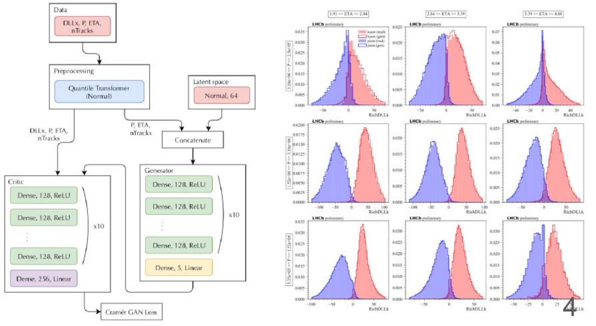


Cache Type	Throughput	Cost	Read on hit ratio	Band sat.	CPU Eff.
SCDL	**79.43%**	**50.68%**	21.22%	58.94%	58.75%
LFU	65.01%	104.73%	**33.29%**	**51.00%**	**60.92%**
Size Big	49.02%	111.73%	28.55%	54.40%	60.41%
LRU	47.15%	112.84%	27.64%	54.93%	59.90%
Size Small	46.71%	113.01%	27.39%	55.01%	59.73%

Caching suggestions using Reinforcement Learning
LOD 2020, in proceedings

Data Management and caching on the GRID

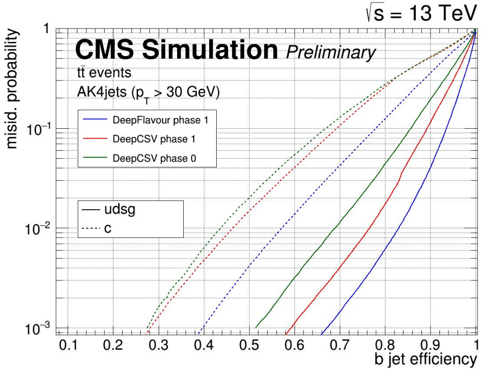
Data Simulation

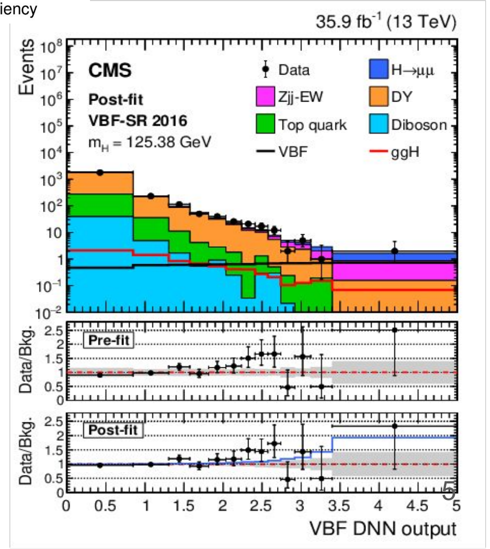


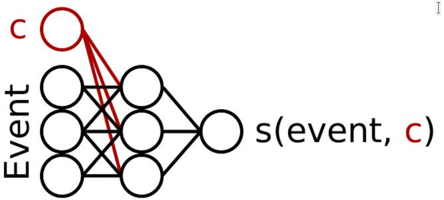
Generative Adversarial Networks for LHCb Fast Simulation [2003.09762]

Usage of ML for **physics analyses**

- “Objects” reconstruction
  - Electrons, Muon, Jets, Missing Energy, etc…
  - Better ID: b-tag, tau tag, quark/gluon
  - Better energy/momentum measurement with “regression” techniques
- Signal vs Background discrimination
  - Bread and butter S vs B separation (MVA)
  - Multi-class discriminators, Parametric DNN, Mass decorrelation, anomaly detection with VAEs
- In the past 5 years we mostly replaced BDTs with DNN at reco/analysis level





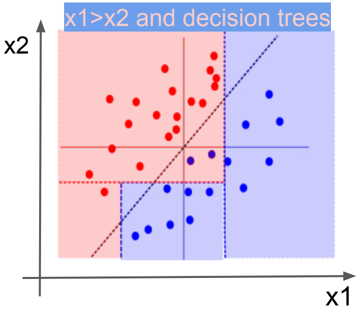


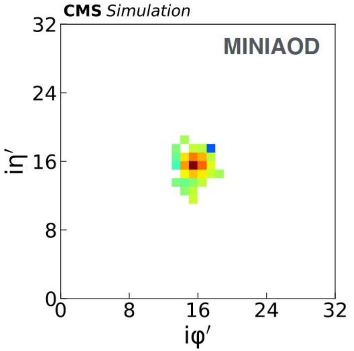
Do we want more ML? Yes

- When should we **not** use ML?
  - When something has a clear, well known analytical/algorithmic solution
  - When we do not have a good data representation for AI to work on (see later slides) and we would be forced to recast our data in some inefficient format (e.g. jet images)
  - When we would not be able to perform some of the tasks needed to do a proper scientific statement on the problem we are studying (a.k.a. explainability matters)
  - When we would need to write a loss function that would properly take into account all the content of a Particle Physics textbook (i.e. our bias of prior knowledge of HEP matters)
- When should we use ML ?
  - Pretty much for everything else
  - Each time our “classical” approach has
    - A large number of inputs/outputs
    - A good amount of arbitrariness in the algorithm parameters
    - A complexity we give up on trying to understand

Is “end-to-end” AI the final goal ?

- “ML mantra”: do not engineer features, give AI the raw information
- Should we really try to go from “detector hits” to “is there an Higgs boson?”
  - Short answer: No
- Long answer…
  - Plenty of analytical steps (e.g. Lorentz vector algebra)
  - Factorization of steps is a key of HEP analysis
    - Calibration and alignment
    - Explainability
    - Intermediate cross-checks, judgement calls, paper credibility



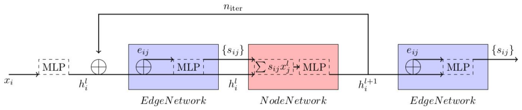


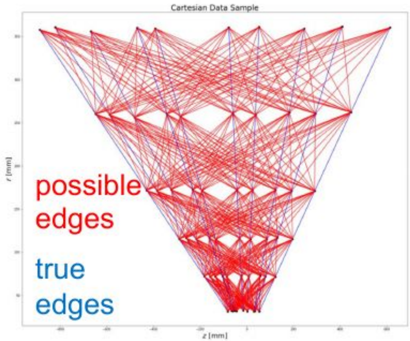
- … but “skip some steps” is good!!
  - *Central limit theorem*: we may not need to simulate the details of Bethe-Bloch energy loss of every single secondary particle to properly go from a “Generator level/MC truth Jet” to a “Reconstructed Jet”
    - We already kill some delta ray, approximate the calo showers, even in our GEANT4 full simulations!
  - Some intermediate observables are **more or less arbitrary**, introduced only because we are not able to handle more complexity (but ML could do it):
    - Shower shapes (CMS at some point had more calorimetric shower shape variables than hits in the calo cluster)
    - Scalar isolation variables (in fact we are inventing PU corrections, shrinking cones, etc…)
    - Jet algorithms “QCD motivated” metrics (in fact, we have p= -1 , 0, +1)

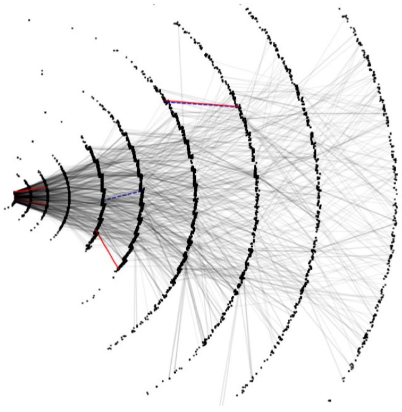
$$d_{ij} = \min\left(k_{Ti}^{2p}, k_{Tj}^{2p}\right)\frac{\Delta_{ij}}{D}$$

Reconstruction

- Which reconstruction steps improvements will bring more benefits to analyses?
  - **Jets:** substructure and tagging, boosted regime, energy regression => very active field!
  - **Energy regression** (not only jets): energy/momentum enters invariant mass calculation => peak resolution => analysis sensitivity
  - Challenging environments, for example:
    - TeV objects in detectors optimized for 100 GeV physics
    - High pile-up scenarios (e.g. HL-LHC)
    - Pattern recognition with diverging combinatorics
    - Showering detectors
- Holy Grail at HL-LHC: Tracking
  - At PU 140 / 200 pattern recognition complexity explodes
  - ML approach
    - Graph Networks?
  - How about secondary vertexing?
    - Number of pairs of tracks diverges even more!



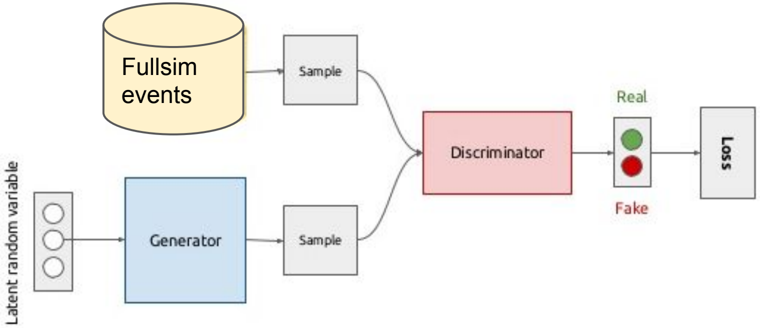


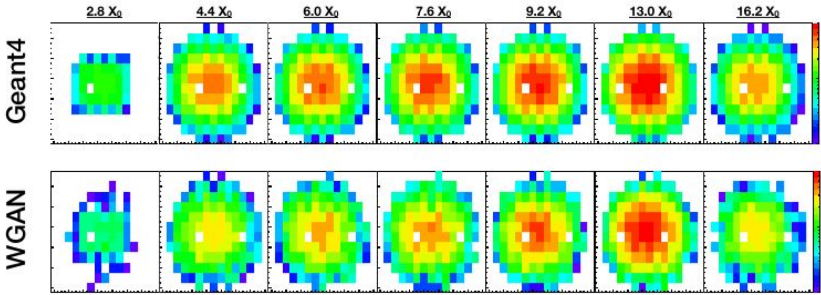


GNN applied to charged particle tracking
[2007.00149]

Simulation

- Full-simulation (GEANT4 based)
  - Speed up some slow parts (e.g. shower models)
- Fast-simulation
  - Various versions of “fastsim” exists (w/o ML)
    - Trying to re-produce a low-level data-tier is hard
      - Do not attempt to simulate “tracking hits” and just short cut to “tracks”
    - Only produce high level (analysis) data-tier (e.g. Delphes simulation)
  - Many inputs, arbitrary in the algorithms, high complexity => ideal AI field!
- Ideally ML may allows to have a simulation that is almost as accurate as Fullsim for analysis purpose but could run as fast as Delphes
  - Having a simplified (invertible?) model to go from Generator level to Analysis level could also help in data interpretation

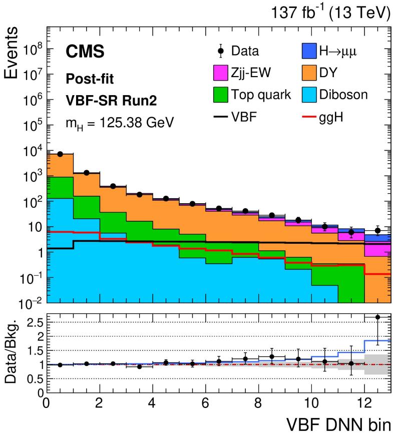




Background models

LHC analyses can be broadly split in **two categories** in terms of how the backgrounds are predicted

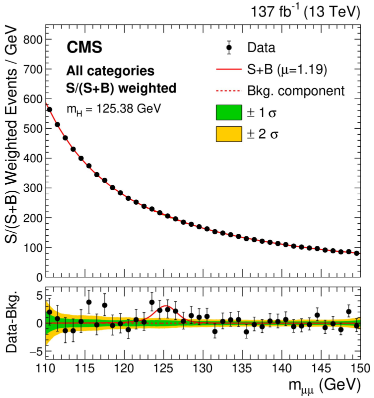
- MC based
  - with possibly some corrections originating from control regions
- Fully “data-driven”, or better “smoothness driven”: i.e. arbitrary functions chosen as background exhibiting no signal-like features (typically peaking structures)



- Trust MC
- High S/B
- MC stat uncert

Can AI help or provide something intermediate?

- Very fast simulation could be used to improve MC based
  - generate >100B events
- Can we use ML to augment our final background ntuples?
  - e.g. smoothing of non trivial distributions
- ML could be used to generate background functions with the needed features (we could use AI to better express our bias of “how the function should be”)



- Limited to smooth observable
- Need categories
- Bias studies

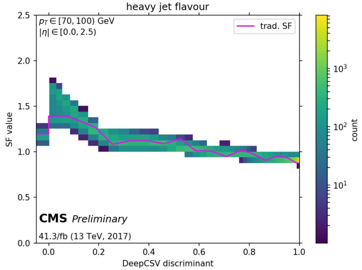
Calibrations, systematics, optimizations

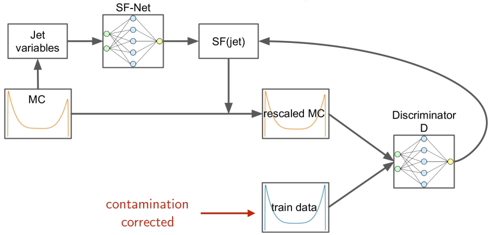
Typically there are two ways of handling calibrations/corrections and/or systematics in MC simulated samples

1. Apply some event by **event weight** (efficiency corrections, PDF weights, etc..)
2. **Re-run the full analysis** with different inputs (energy corrections, generator settings)

Can we move more of the type “2” to “1” (much cheaper) ?

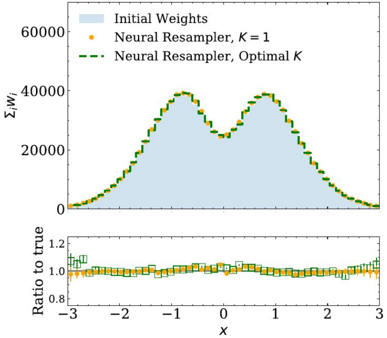
- Jet energy corrections as a weight? (save ~ x40 event looping)
- Generator tuning/settings as a weight? (a single full sim pass)
- LO to NNLO reweighting? (avoiding negative weights!)

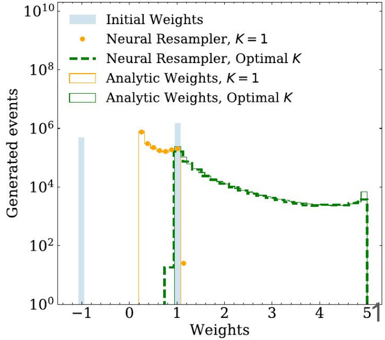




Typically a multidimensional reweight problem with correlations, too hard for simple approaches such as “I make the ratio of two histograms”

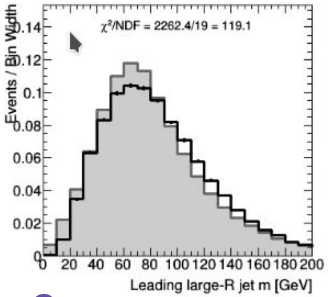
- Machine Learning can be a game changer

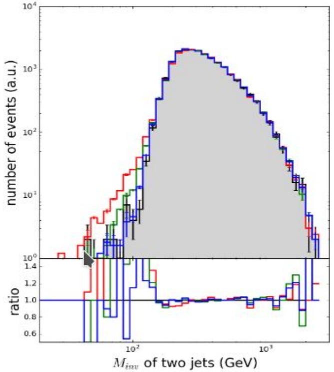




MC tuning and AI

- We have a lot of (real) data in the “soft” phase space that is often used for tuning of generators (parton shower, hadronization, etc…)
- We tune some (more or less) QCD motivated effective models to predict some observables
  - Underlying event properties
  - Track multiplicity
  - Pile-up
  - Jet showering

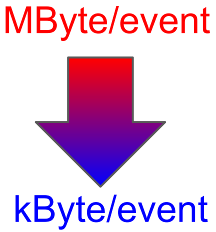




- Would AI be able to better model these processes?
  - Complexity we give up understanding, many inputs/outputs, many arbitrary parameters/algorithms
  - Can we create unsupervised generative models? How do we establish later the connection to the Matrix Element level observables?
- NB: State of the art Parton Density Functions (PDFs) are already using Neural Networks (NNPDF 3.1 current default at LHC)

Data tiers in High Energy Physics

- Event data is typically represented with multiple “data formats”
  - RAW format with low level detector information
  - RECO detailed information of reconstructed particles
  - AOD detailed format for analysis
  - MiniAOD compressed format for analysis
  - NanoAOD/Ntuples highest level representation



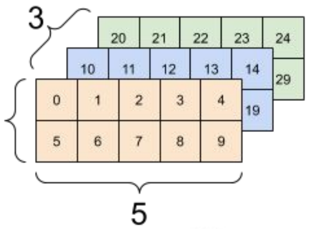
MByte/event

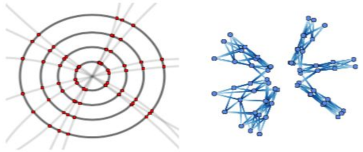
kByte/event

- Lightest formats capture our bias of “what is relevant for analysis”
- Should we rather ask AI what is relevant for analysis?
  - Anything that is allowing to better learn the event probability distribution is “relevant for analysis”
  - We have plenty of real data and plenty of example signals
- Concrete examples, starting small aiming big
  - Lossy compression: compression of track covariant matrix (CMS MINIAOD using a human designed compression)
  - Train DNN (VAE) to just classify LHC events that can later be exploited in analyses
  - How about an AI developed PICOAOD ?
    - Target **100 bytes/event** ?
    - Should the 100 bytes be still “explainable” ? (as opposite of just a 100 nodes VAE middle layer)
    - A full HL-LHC analysis may require the analysis of 100 billions events
  - Anomaly detection as a way to find new physics (“anomaly” with respect to what?)

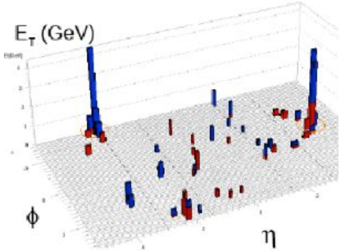
Data representation: graph networks

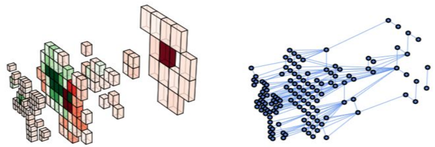
- Problem: a lot of AI tools are designed with images, videos, text, speech applications
  - Nicely represented with multidimensional tensors
  - Invariance under space/time translation easily represented (CNN and RNN are successful as they exploit this)
- High Energy physics data …
  - are **sparse** (we do zero suppression already at hit level)
  - have **variable length** and are rich of **relations**
  - have different invariance than translations in space/time
- Graph networks/Geometric deep learning
  - **represent data on a graph!!**
- First applications already being the new state of the art (e.g. in tagging algorithms)
- Drawbacks/limitations
  - Lack of standard tools / some python tools not easily integrated in C++ reconstruction software
  - Acceleration not as straightforward as for regular tensors



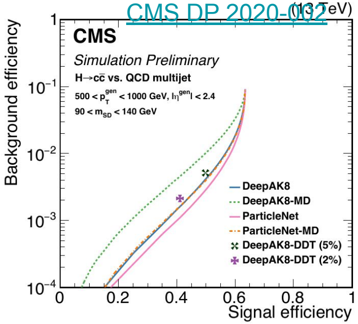


Tracker hits as a graph

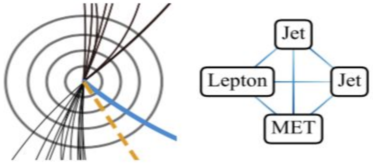




Calo hits as a graph



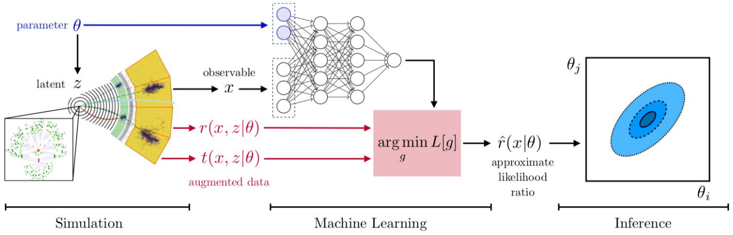
CMS DP 2020-002

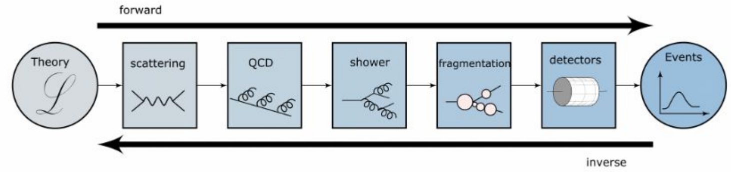


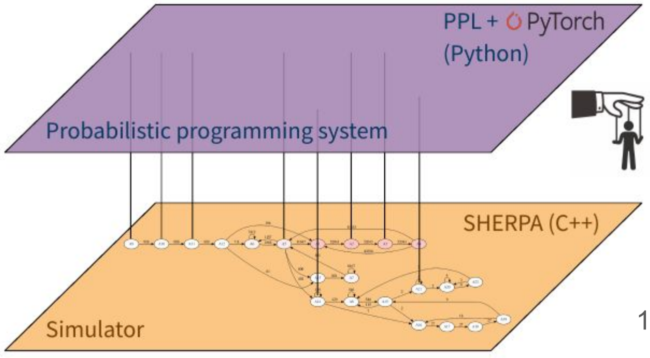
Objects in event as a graph

Fitting, publishing, uncertainties, likelihoods

- We know how to do Theory => Prediction
  - But we are unsatisfied with Measurement => Theory
- The way experiments fit and publish results is changing
  - Common tools for fitting (for cross-experiment combinations)
  - Combination across different analyses
  - Interpretations in large parameter spaces
  - Effective Field Theories common language
  - Unfolding to parton/particle level
- Can ML play a role?
  - The fit observables are often DNN
    - How can we optimize them for inference? (e.g. aware of systematic uncertainties)
  - Simulation-based inference
  - Can we just “invert” some end-to-end algorithms?

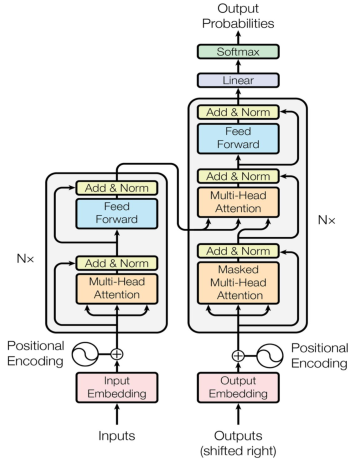


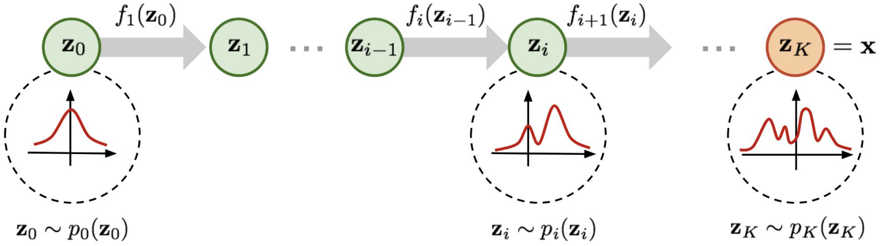




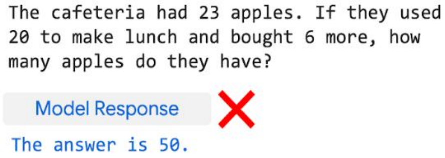
AI technologies we are not yet exploiting

- AI Transformers
  - Recntly proposed architecture that allows AI/ML to scale to multi-billions parameters model (see e.g. Google AI blog)
  - Many (impressive) applications in language processing
  - A “recurrent” architecture with the concept of “attention” to learn context
- Normalizing flows
  - Models using (learnable) bijections to transform from a multi dimensional-gaussian distributed space to a complex multi-dimensional probability





A couple of years ago

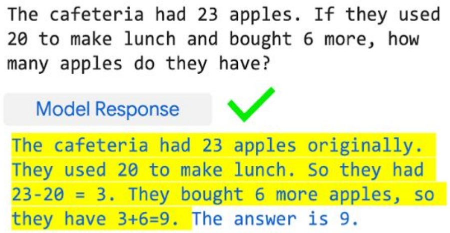


The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Response ✗

The answer is 50.

Recent results



The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Response ✓

The cafeteria had 23 apples originally. They used 20 to make lunch. So they had 23-20 = 3. They bought 6 more apples, so they have 3+6=9. The answer is 9.

Conclusions

- Hundreds of papers in the context of HEP and AI/ML
- Multiple concrete applications already deployed
- R&D extremely active, some trade-off to balance
  - **Latest tools and ideas** from computer scientists
  - Need to **deploy stable tools** for data-taking and analysis
- AI/ML can be a game-changer for future experiments
  - Optimization and automatization of operations (and design?) of accelerators and experiments
  - Better, faster simulations
  - More performant reconstruction algorithms
  - Better interplay with theory and interpretation