

Machine Learning examples in High Energy Physics

Oleg Filatov¹, Kirill Bukin²

¹Deutsches Elektronen-Synchrotron (DESY)

²Moscow Institute of Physics and Technology



Overview

- ① Making theories
- ② Building detector
- ③ Collecting data
- ④ Reco&ID objects
- ⑤ Simulating physics
- ⑥ Performing analysis
- ⑦ Contemplating future

Overview

- ① Making theories
- ② Building detector
- ③ Collecting data
- ④ Reco&ID objects
- ⑤ Simulating physics
- ⑥ Performing analysis
- ⑦ Contemplating future

Disclaimer: it is impossible to cover all the applications at once. Those which are presented here have been selected solely for the sake of supporting the main narrative of the presentation. I encourage you to follow the links on the slides, explore what's not mentioned and see how far your curiosity may get you!

Note: since the beginning of my research path I've always been a member of CMS collaboration, hence a corresponding bias in presented applications might take place.

Making theories

Making theories

- **Note:** I am extremely “experiment”-biased and don’t really keep track of theory developments
- Here for the sake of completeness I highlight just a few applications which I stumbled upon – but there’s definitely more

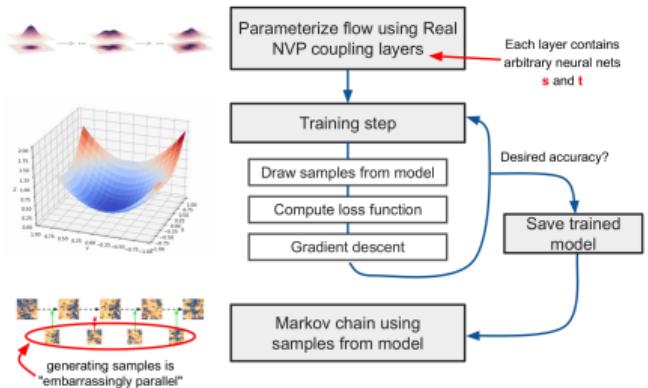
Making theories

- Note: I am extremely "experiment"-biased and don't really keep track of theory developments
- Here for the sake of completeness I highlight just a few applications which I stumbled upon – but there's definitely more
- Lattice QCD: accelerated sampling of field configuration with **gauge-equivariant normalising flows**

Approximate the QCD path integral by Monte Carlo

$$\langle \mathcal{O} \rangle = \frac{1}{Z} \int \mathcal{D}A \mathcal{D}\bar{\psi} \mathcal{D}\psi \mathcal{O}[A, \bar{\psi}\psi] e^{-S[A, \bar{\psi}\psi]} \rightarrow \langle \mathcal{O} \rangle \simeq \frac{1}{N_{\text{conf}}} \sum_i^N \mathcal{O}([U^i])$$

with field configurations U^i distributed according to $e^{-S[U]}$

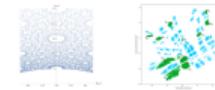


Summary chart: Tej Kanwar

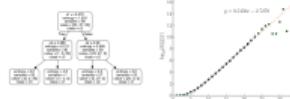
Making theories

- **Note:** I am extremely "experiment"-biased and don't really keep track of theory developments
- Here for the sake of completeness I highlight just a few applications which I stumbled upon – but there's definitely more
- Lattice QCD: accelerated sampling of field configuration with **gauge-equivariant normalising flows**
- String Theory: computation, optimisation, topology

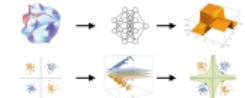
Structure of vacua (Unsupervised ML)
• Clustering, Feature extraction
• Topological data analysis



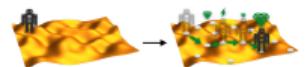
Conjecture generation (Intelligible AI)
• Decision Trees
• Regression



Bypass Computations (Supervised ML)
• Deep neural networks
• Support vector machines



Search the landscape (Semi-supervised ML)
• MC tree searches
• Dynamic programming in MDP
• Reinforcement Learning



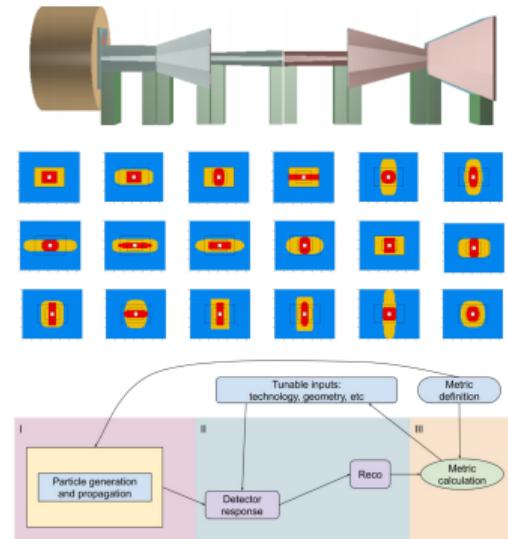
Building detector

Building detector —— o geometry

- It all starts from building a detector - which is tough, **a lot of constraints** need to be taken into account (money vs. physics)
- Can we somehow find the "optimal" geometry?

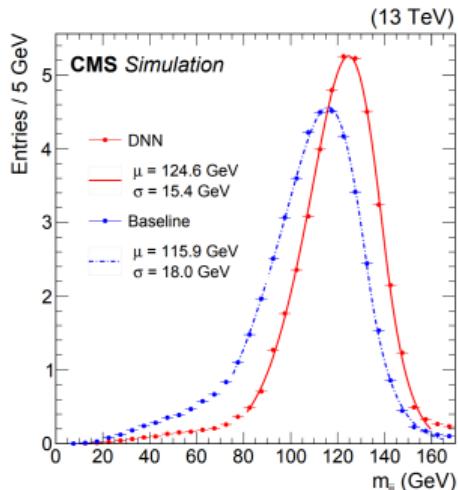
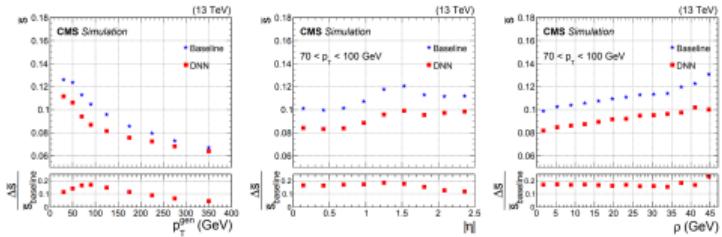
Building detector ——o geometry

- It all starts from building a detector - which is tough, **a lot of constraints** need to be taken into account (money vs. physics)
- Can we somehow find the "optimal" geometry?
- Yes, as it's an optimisation problem
- In fact, "**black box**" optimisation → bayesian optimisation
 - Muon shielding in SHiP experiment
 - LHCb ECal case



Building detector —○ calibration

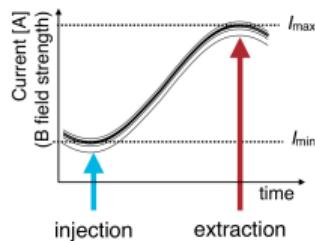
- What is also important then is to setup detector properly → **calibration**
- Take calorimeter and jet mass as observable
- Distribution has asymmetric non-Gaussian tails → MSE fit will have biased result
- Need to simultaneously predict *median* or *mode* of distribution + *quantiles* for jet energy resolution → proper loss needed
- CMS: uses Huber loss
- ATLAS: uses Leaky Gaussian Kernel



Building detector

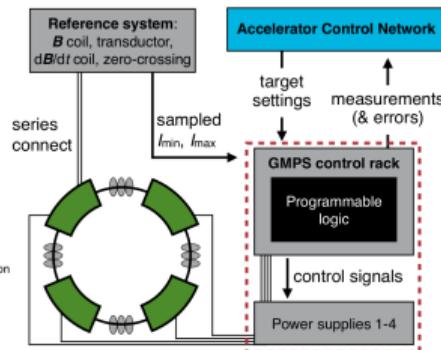
—○ maintenance

ML control system for the Fermilab Booster



- Combination of RF cavities (coil) and bending magnets (coil)
- Bending magnet current ramps in 15hz cycles to maintain the orbit of the accelerating beam

Booster: 400 MeV → 8 GeV



and finally, treat it with love and care

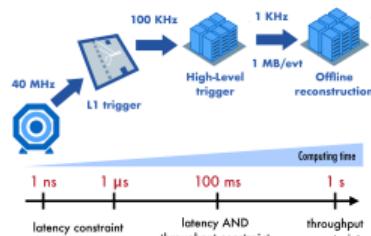
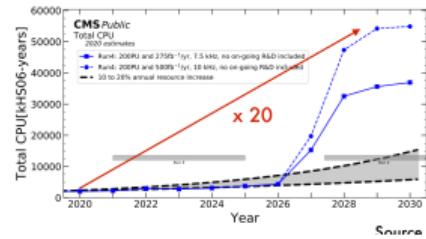
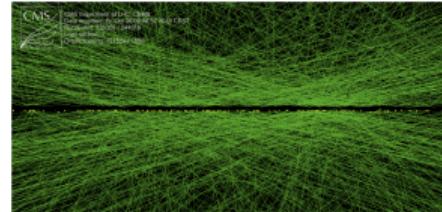
Collecting data

Collecting data

—○ motivation

Ngadiuba's talk @FastML2020

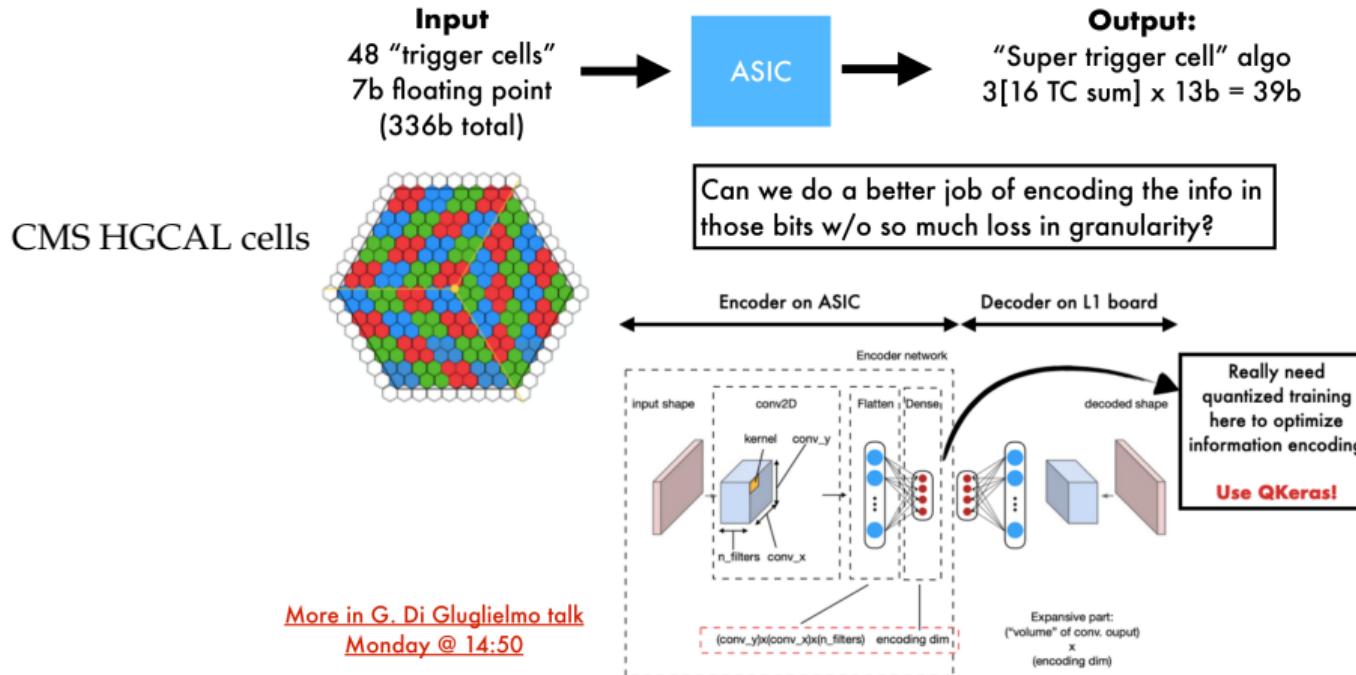
- Once collisions begin, data from the detector needs to be collected
 - Huge in size *but* can't store everything
 - Respond and process new data quickly *but* computational resources limited
- Need to find the "best", select it and do this fast → requires **low latency, large throughput**



Collecting data

—○ DAQ

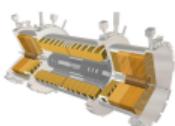
Ngadiuba's talk @FastML2020



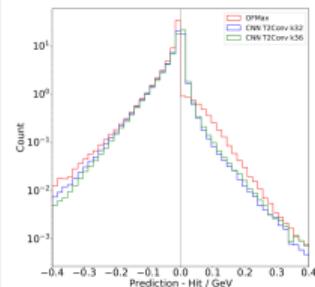
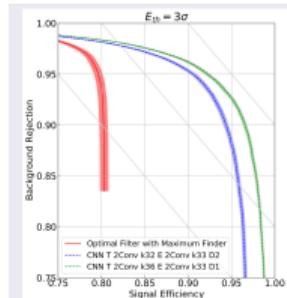
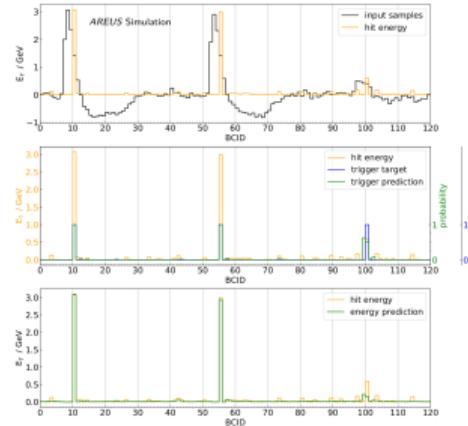
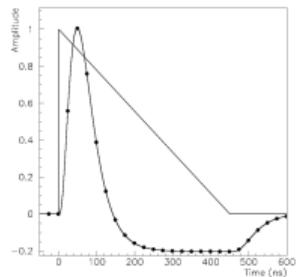
Collecting data

—○ DAQ

Fritzsche's talk @FastML2020



readout per cell →

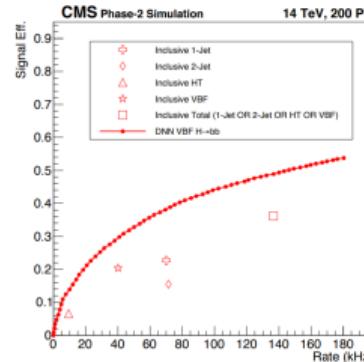


Collecting data

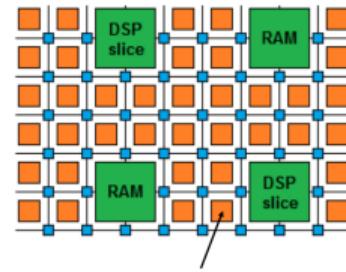
—○ trigger

CMS-TDR-021

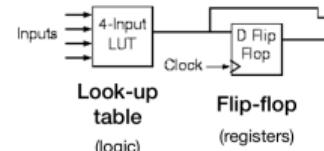
- Once data is digitalised and pre-selected, it needs to be skimmed more prior to be stored → **trigger**
- Conventionally, a set of selection requirements (cuts)
- Not efficient** to select interesting events → **use ML models**
- Need to work *really fast*, be **lightweight** and programmed to fit into hardware (e.g. **FPGA**)



FPGA diagram



Logic cell



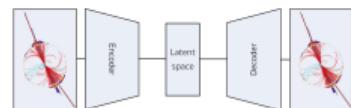
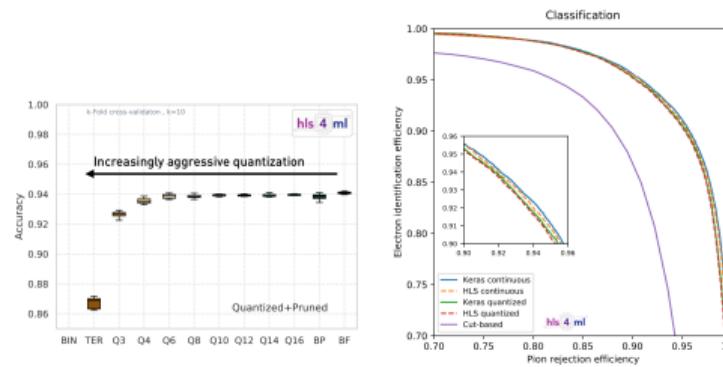
Look-up table (logic)
Flip-flop (registers)

Collecting data

—○ trigger

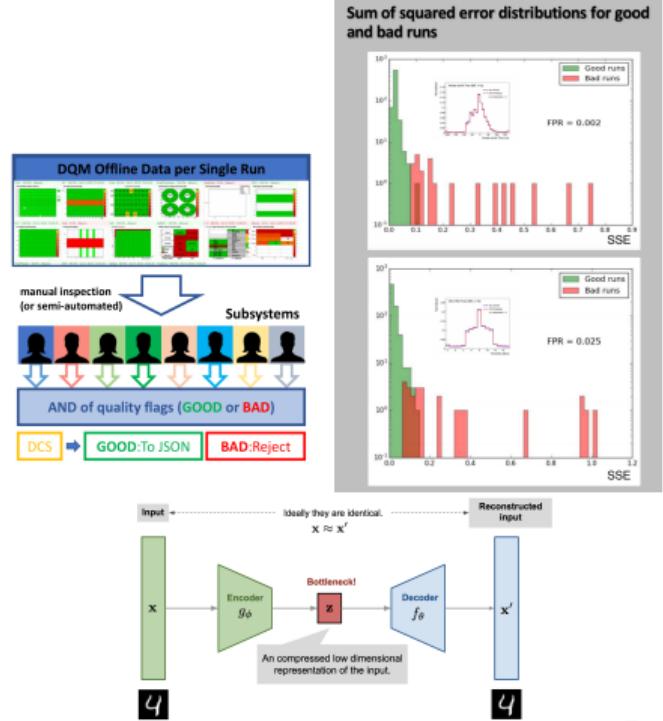
CMS-TDR-021

- Once data is digitalised and pre-selected, it needs to be skimmed more prior to be stored → **trigger**
 - Conventionally, a set of selection requirements (cuts)
 - Not efficient** to select interesting events → **use ML models**
 - Need to work *really fast*, be **lightweight** and programmed to fit into hardware (e.g. **FPGA**)
- **hls4ml project**
- deployment of ML models on FPGA with ease
 - quantization and weight pruning
 - improved object reconstruction (muons, taus, tracks)
 - anomaly detection with signal-agnostic triggering



Collecting data —— o DQM

- After preselection and triggering, we are only interested in "high-quality" data → **data quality monitoring (DQM)**
- can use autoencoder to search for "poor-quality" anomalies



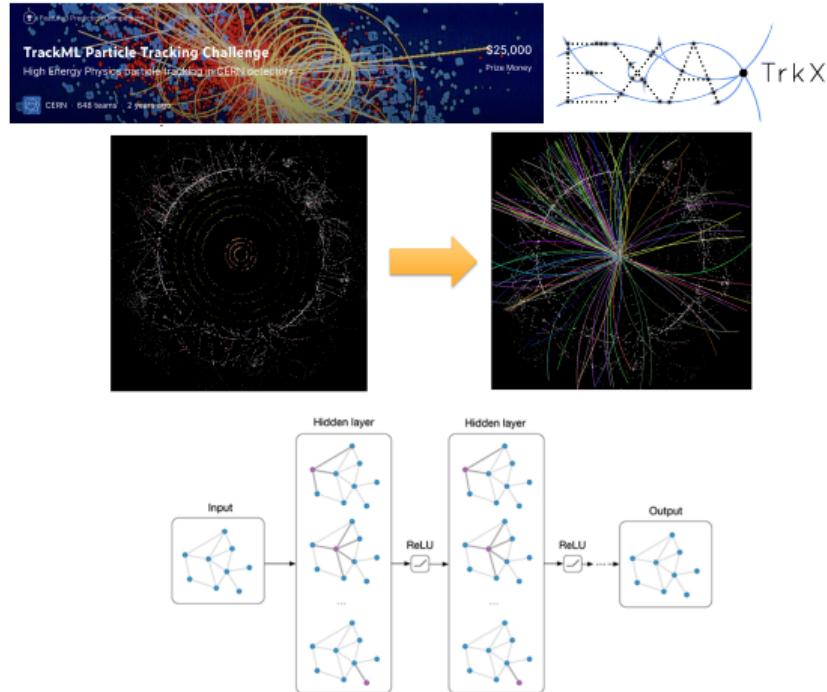
Reconstructing objects

Reconstructing objects

— o tracks

Tracking GNN Walk Through

- Traditional Kalman filter will become inefficient with increased luminosity
- It all (seriously) started from [TrackML hackathon](#) on Kaggle
- Several research projects emerged, e.g. [ACTS](#), [HEP.TrkX](#) and then [Exa.TrkX](#) as its follow-up
- Various methods probed (CNN, RNN), finally converged to **graph networks**
- reached $\sim 95\%$ purity/precision

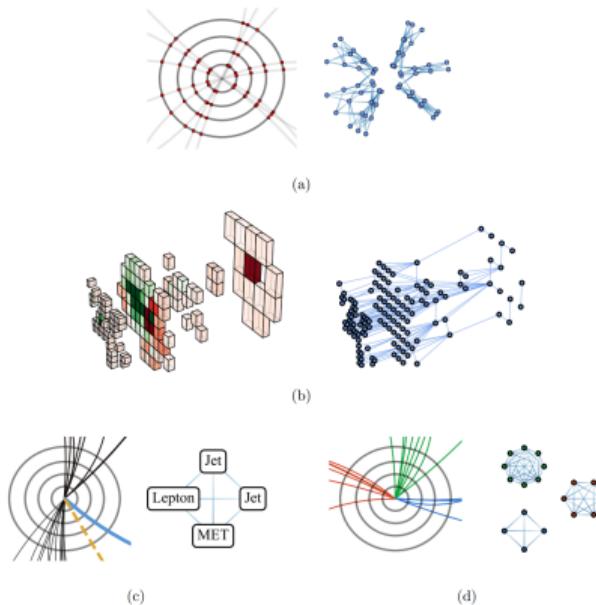


Reconstructing objects

—○ graphs in HEP

GNN in Particle Physics paper

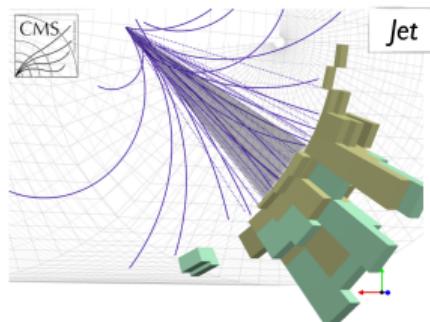
- natural representation
- native treatment of **sparsity**



Reconstructing objects

— o jets: representation

Qu's talk (CMS restricted)

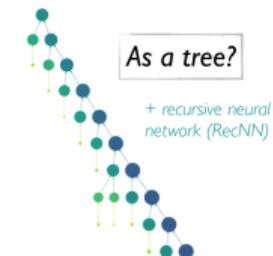


As a sequence?

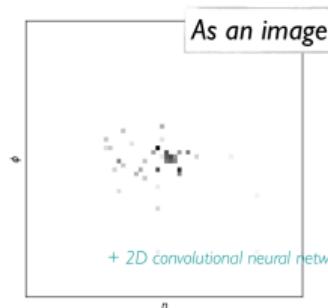
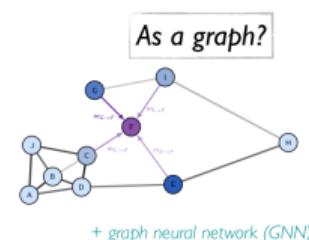


- + recurrent neural network (RNN)
- + 1D CNN
- + etc.

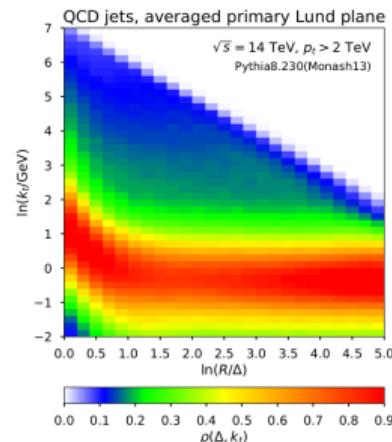
As a tree?



As a graph?



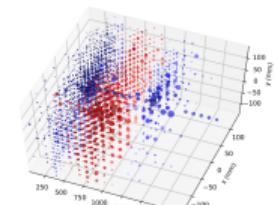
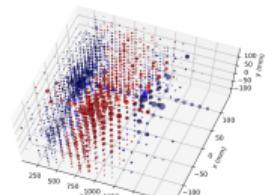
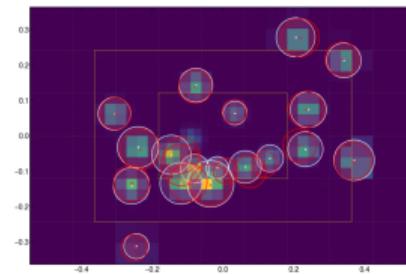
As a Lund plane?



Reconstructing objects

—○ jets: segmentation

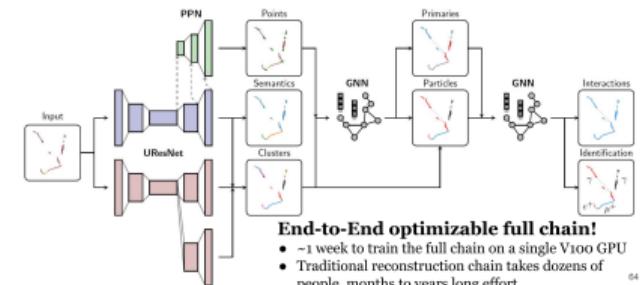
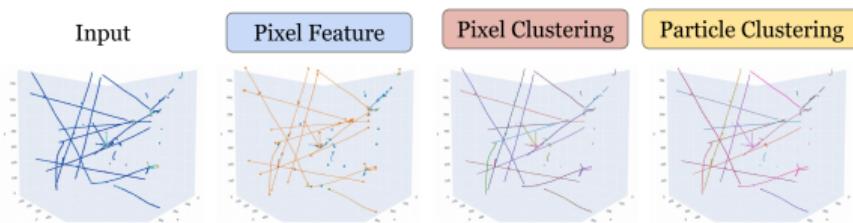
- CV-inspired YOLO + graphs under the hood for cluster reconstruction @LHCb
- GravNet: clustering energy depositions in CMS HGCAL with distance-weighted graph nets
- Object condensation: on top of GravNet to build *one-shot ParticleFlow*



Reconstructing objects

— o bonus: neutrino

Kazuhiro's talk @IML2020

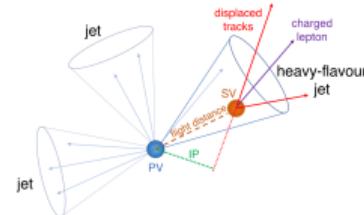


Reconstructing objects

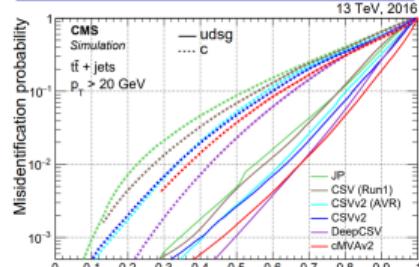
—○ jets: tagging

- **b-tagging** [1], [2]

- crucial to identify jets from b quark (e.g. $H \rightarrow b\bar{b}$)
- ML brings significant gain ($\sim 50\%$) comparing to classical approaches
- solid place in production (e.g. DeepJet @CMS)
- in fact, also used for c/uds/g tagging



| CSVv2 | b vs c | Changed (8 features) | preprocessing | 20 nodes x3 | 1 |
|---------|-------------------------------|--------------------------------|-------------------------------|--------------------------------|---|
| | Secondary Vtx (8 features) | preprocessing | | | 8 |
| | Global variables (7 features) | preprocessing | | | |
| DeepCSV | b vs c | Changed (8 features) | preprocessing | 20 nodes x3 | 2 |
| | Secondary Vtx (5 features) | preprocessing | | | |
| | Global variables (7 features) | preprocessing | | | |
| DeepJet | b vs c | Changed (16 features) x2 | preprocessing | 1x1 core: 64/32/32/8 - RNN 100 | 3 |
| | t/t + jets | Neutral (8 features) x2 | 1x1 core: 32/16/8 - RNN 50 | 200 nodes x3 | |
| | | Secondary Vtx (12 features) x2 | 1x1 core: 64/32/32/8 - RNN 50 | 100 nodes x3 | |
| | | Global variables (17 features) | | | |



Reconstructing objects

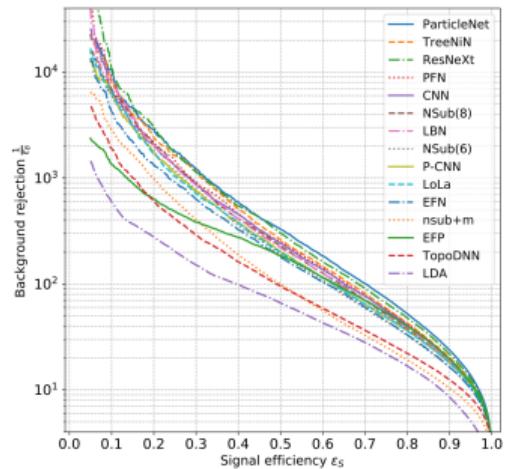
—○ jets: tagging

- **b-tagging** [1] , [2]

- crucial to identify jets from b quark (e.g. $H \rightarrow b\bar{b}$)
- ML brings significant gain ($\sim 50\%$) comparing to classical approaches
- solid place in production (e.g. DeepJet @CMS)
- in fact, also used for c/uds/g tagging

- **t-tagging**

- significant piece in SM puzzle (e.g. $t\bar{t}H$, $t\bar{t}$)
- complex to become benchmarking playground in tagging

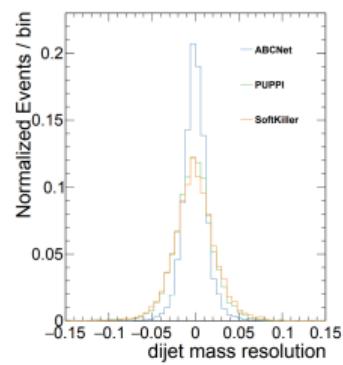
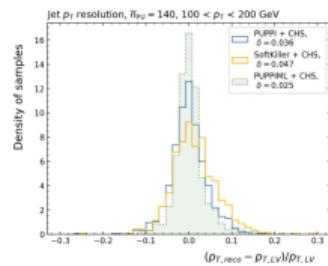
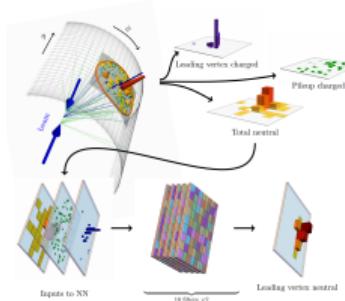


Reconstructing objects

—○ jets: cleansing

• Pile-up mitigation

- remove irrelevant contributions from pile-up
- PUMML: convolutional analysis of LV charged/PU charged/neutral components
- PuppiML: gated graph-based architecture
- ABCNet: graph net with attention



Reconstructing objects

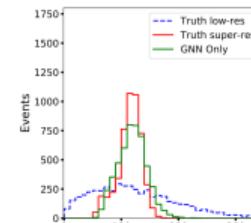
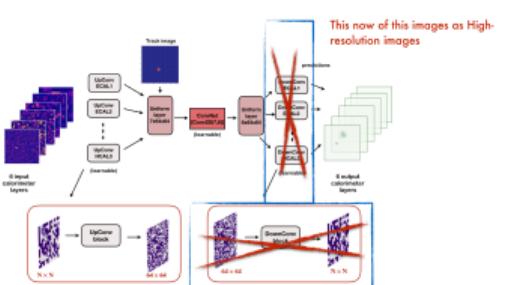
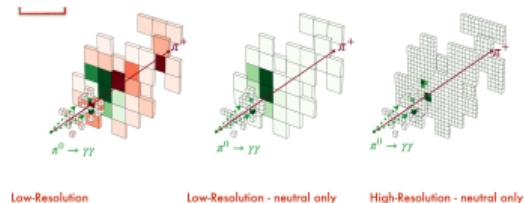
—○ jets: cleansing

- **Pile-up mitigation**

- remove irrelevant contributions from pile-up
- PUMML: convolutional analysis of LV charged/PU charged/neutral components
- PuppiML: gated graph-based architecture
- ABCNet: graph net with attention

- **Super-resolution**

- enhance the calorimeter's granularity with CNN/GNN
- able to nicely resolve photons in $\pi_0 \rightarrow \gamma\gamma$



Reconstructing objects

—○ misc

- **Vertices**

- find ML-reconstructed (secondary) vertices in jets
- to be further passed to tagging algorithms

- **Leptons**

- τ : e.g. ABCNet for low- p_T , or general-purpose DeepTau
- $\mu/e/\gamma$ (were among the first MVA ID)

- **Missing transverse energy (MET)**:

- used to approximate undetected particles' momentum (ν , SUSY)
- e.g. DeepMET

- **Long-lived particles (LLP)**

- if you want to search for some SUSY

- **Bosons**

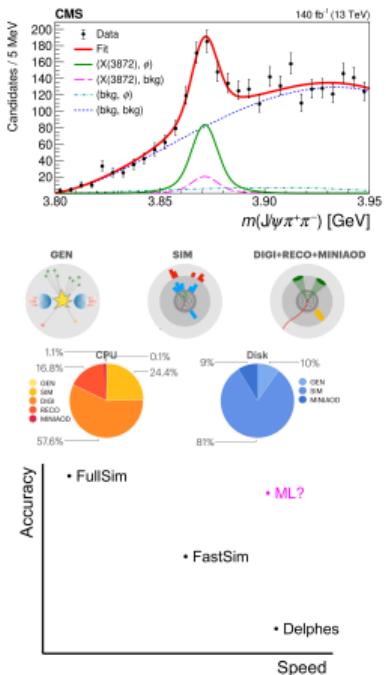
- tag W/Z/H bosons directly
- especially in boosted topologies
- and with a Lund plane

Simulating physics

Simulating physics

—○ motivation

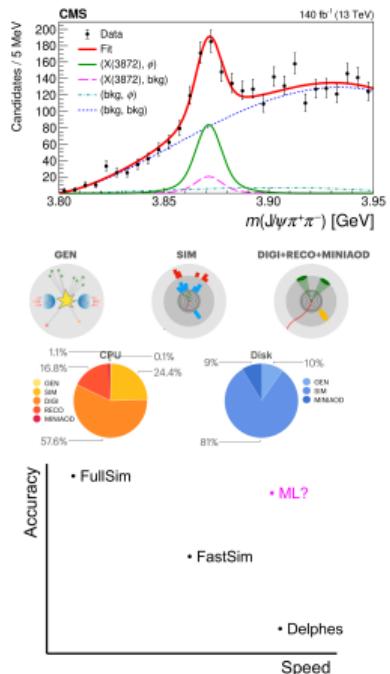
- What we want to know from data is parameters of the model we study (SM/BSM)
- Traditionally, model's behaviour at the level of observables is obtained by (aka **Monte Carlo**) simulation in the form of **templates** to be fitted to observed data
- **Need to simulate A LOT** of data (e.g. to have low statistical uncertainty of model templates)
- Takes A LOT of time and computational resources
- Can we speed it up?



Simulating physics

—○ motivation

- What we want to know from data is parameters of the model we study (SM/BSM)
- Traditionally, model's behaviour at the level of observables is obtained by (aka **Monte Carlo**) simulation in the form of **templates** to be fitted to observed data
- **Need to simulate A LOT** of data (e.g. to have low statistical uncertainty of model templates)
- Takes A LOT of time and computational resources
- Can we speed it up? → **generative models**
 - (unconfirmed) amount of hype in this topic is proportional to amount of data we need to simulate
 - so I will give just a few examples of those



Simulating physics —○ examples

- tried images, features, point clouds used as inputs/outputs
- can be *conditional* → specify properties to be generated
- **GAN**
 - historically first: LAGAN, CaloGAN
 - graph-based, end-to-end mapping GEN→RECO levels and bypassing details
RICH response simulation
- **Autoencoder**
 - e.g. getting high with BIB-AE, or
VAE+GAN @LHCb
- crucial to understand and evaluate **systematic uncertainties**
- reached speed-up of $\lesssim x1000$ comparing to GEANT
- extensive overview [here](#)

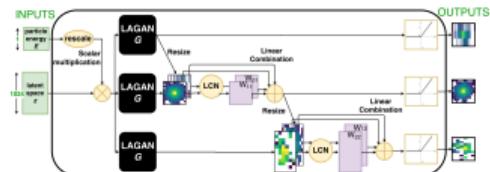
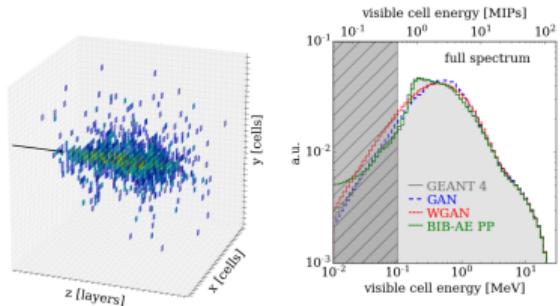


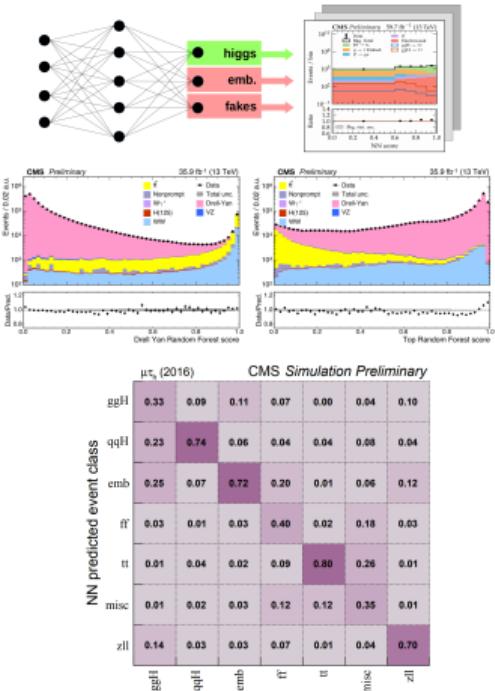
FIG. 4: Composite Generator, illustrating three stream with attentional layer-to-layer dependence.



Performing analysis

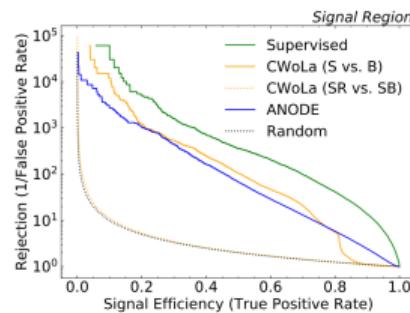
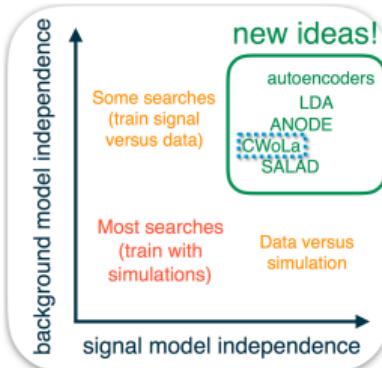
Performing analysis ——○ sig vs. bkgr

- need to classify (several) signal processes from (several) background ones
- afterwards, either apply the cut on model's output and fit observable
- or incorporate it into fitting procedure (e.g. 2D fit)
- **good old classics**, used literally everywhere
- e.g. W^+W^- production (they use Random Forest!!), or $H \rightarrow \tau\tau$ CP study @CMS



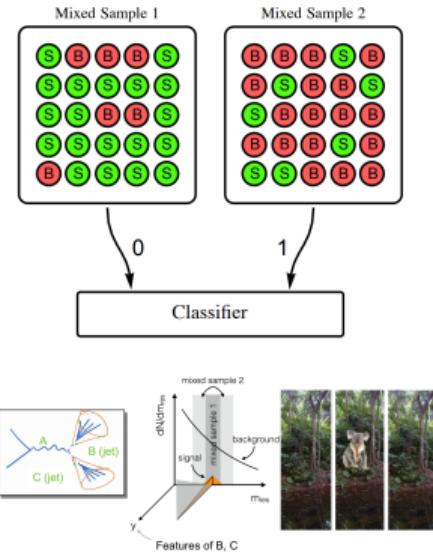
Performing analysis ——○ anomaly detection

- Normally, we search for new physics having some model in mind
- But if one want to do it **model-independently** – try instead to search for *deviations* → **unsupervised learning**
- Also hot topic: ANODE, SALAD



Performing analysis ——○ anomaly detection

- Normally, we search for new physics having some model in mind
- But if one want to do it **model-independently** – try instead to search for *deviations* → **unsupervised learning**
- Also hot topic: [ANODE](#), [SALAD](#)
- [CWoLa](#) (classification without labels) – used in a (serious) analysis [in ATLAS!](#)



Performing analysis ——○ decorrelation & adaptation

- It was observed, that cutting on classifier output can heavily distort the distribution of observable (e.g. invariant mass) → **mass sculpting**
- This is quite dangerous (not noticed, noticed but reduce sensitivity)
- How to reduce dependency of classifier on some particular variable?
- Mode, DisCo, β -TCVAE-Sensitive, uBoost, adversarial methods

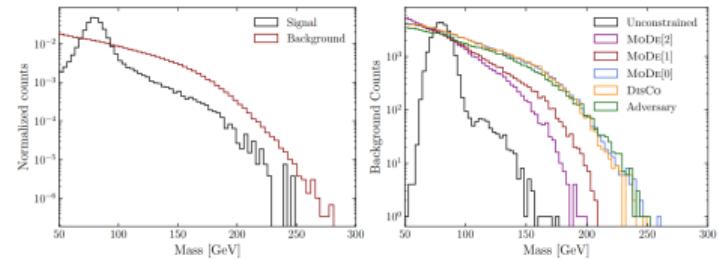
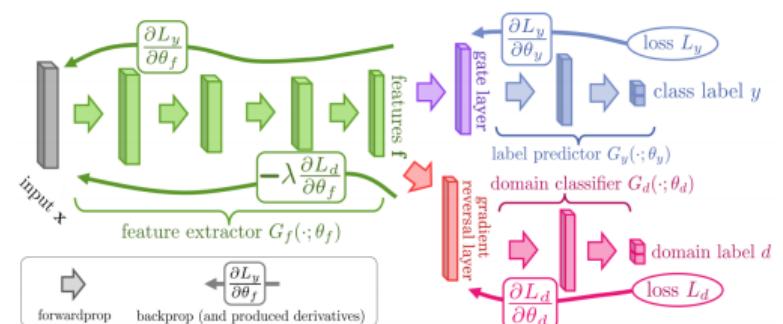


Figure 6. Left: Distributions of signal and background events without selection. Right: Background distributions at 50% signal efficiency (true positive rate) for different classifiers. The unconstrained classifier sculpts a peak at the W -boson mass, while other classifiers do not.

Performing analysis

—○ decorrelation & adaptation

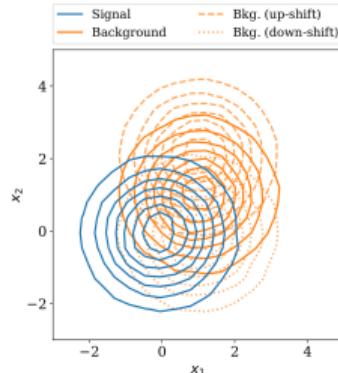
- It was observed, that cutting on classifier output can heavily distort the distribution of observable (e.g. invariant mass) → **mass sculpting**
- This is quite dangerous (not noticed, noticed but reduce sensitivity)
- How to reduce dependency of classifier on some particular variable?
- Mode , DisCo , β -TCVAE-Sensitive , uBoost , adversarial methods
- Furthermore, we can expand this idea to the **meta-level robustness**: MC/background choice, etc. → domain adaptation



Performing analysis

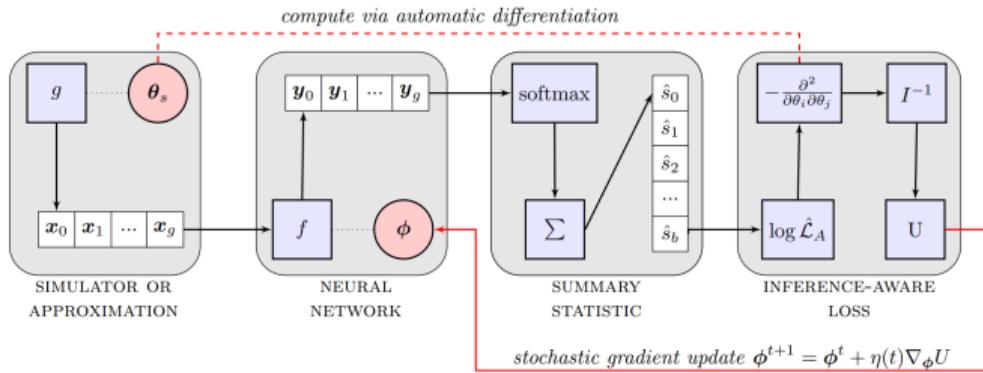
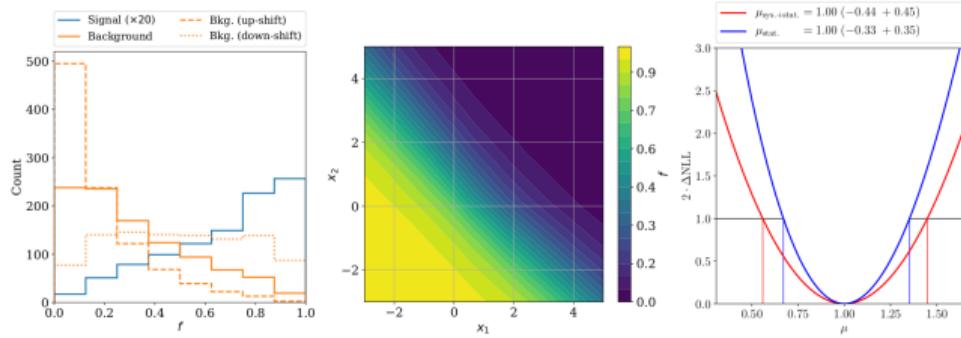
 ——○ nuisances & inference

- Every analysis has statistical/systematic uncertainties, so-called **nuisance parameters**
- Vanilla training of ML models doesn't take them into account
- "Optimised" model in reality can be suboptimal – nuisance variations cause large variations of model prediction, thus **diluting sensitivity**
- Moreover, training with e.g. conventional cross-entropy is also suboptimal – we need to **optimise physical metrics**
- Solutions proposed, e.g.:
 - direct optimisation of Asimov Median Significance (AMS)
 - INFERNO, binned Poisson likelihoods
 - likelihood-free inference
- Complex topic and **one of the challenges of ML in HEP** to date!
- see more in this nice review



Performing analysis

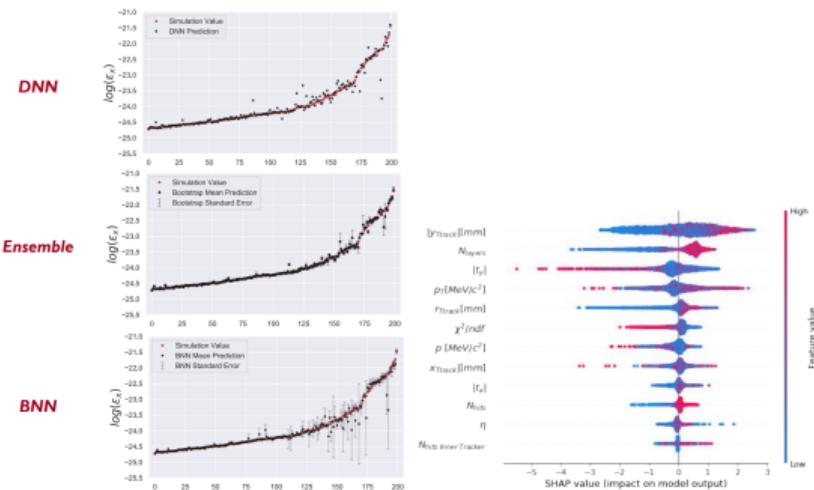
→ nuisances & inference



Performing analysis ——○ interpretability

Interpretable ML book

- why ML works? → hand-wavy answer here
(aka we don't really know)
- but we can try to get insights:
 - bayesian networks to access per-jet uncertainties
 - and more bayesian for accelerators
 - SHAP values for feature importance



Performing analysis

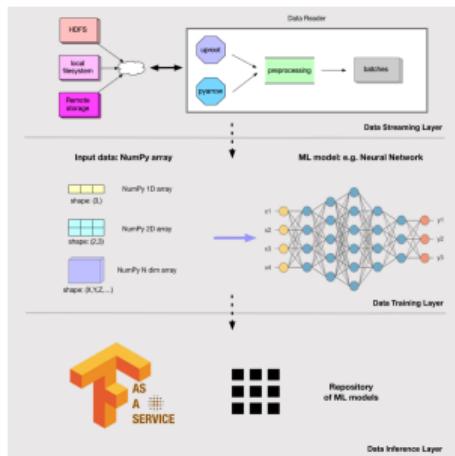
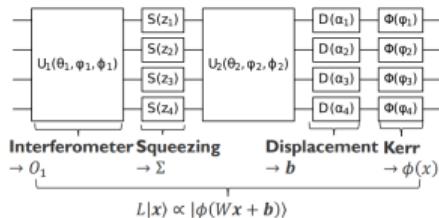
—○ misc.

- **Quantum ML**

- e.g. quantum GANs
- or quantum classifiers

- **Production**

- models on your laptop are generally useless for your colleagues
- needs to be implemented into some framework
- a whole new topic to be aware of



Contemplating
future

Contemplating future — o future directions

thoughts from spring 2019

- **research**

- incorporate physics into the non-physical ML
- make it (physically) interpretable
- get closer to ML community (but also be inventive)

Contemplating future — o future directions

thoughts from spring 2019

- **research**
 - incorporate physics into the non-physical ML
 - make it (physically) interpretable
 - get closer to ML community (but also be inventive)
- **production**
 - deploy more models into frameworks in a centralised manner
 - aggregation of hardware (more GPUs to the crowd!)
 - "make all that cool **research** available for everyone"

Contemplating future — o future directions

thoughts from spring 2019

- **research**
 - incorporate physics into the non-physical ML
 - make it (physically) interpretable
 - get closer to ML community (but also be inventive)
- **production**
 - deploy more models into frameworks in a centralised manner
 - aggregation of hardware (more GPUs to the crowd!)
 - "make all that cool **research** available for everyone"
- **knowledge**
 - structure all that (exponentially growing) papers
 - educate people on how to do ML in their analyses
 - "bookkeeping" of **research** and **production**

Contemplating future —○ about to come

thoughts from spring 2019

- **now**
 - "I heard ML is cool, maybe I should use it?"

Contemplating future —○ about to come

thoughts from spring 2019

- **now**
 - "I heard ML is cool, maybe I should use it?"
- **soon**
 - *opeping a catalogue of models*
 - "today I feel like going for some boosted-graphy-convo-net sprinkled with sparsed variational tagger"

Contemplating future —○ about to come

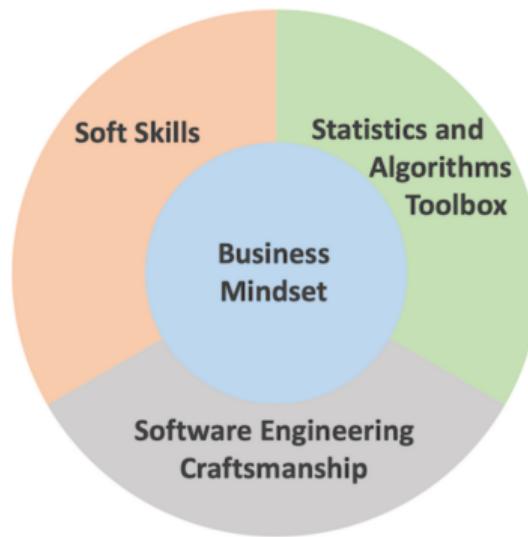
thoughts from spring 2019

- **now**
 - "I heard ML is cool, maybe I should use it?"
- **soon**
 - *opeping a catalogue of models*
 - "today I feel like going for some boosted-graphy-convo-net sprinkled with sparsed variational tagger"
- **future**
 - "sure, but where's new physics?"

amazing talk on ML&HEP future

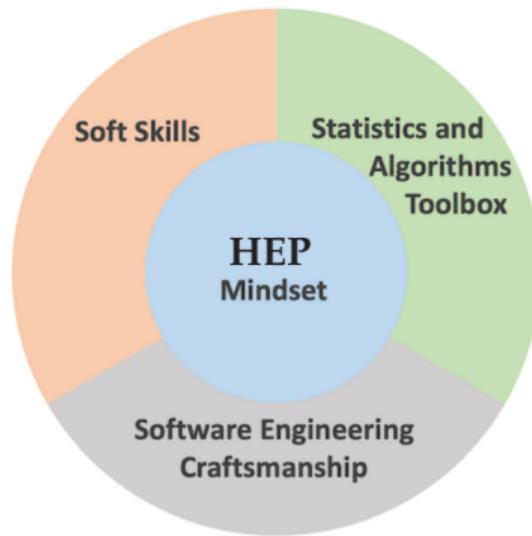
Contemplating future — o third wave data scientist

inspired by



Contemplating future — o third wave data scientist in HEP

inspired by



Summary

- ① Making theories
- ② Building detector
- ③ Collecting data
- ④ Reco&ID objects
- ⑤ Simulating physics
- ⑥ Performing analysis
- ⑦ Contemplating future