Machine Learning examples in High Energy Physics

Oleg Filatov[1], Kirill Bukin[2]

[1]Deutsches Elektronen-Synchrotron (DESY)
[2]Moscow Institute of Physics and Technology

Overview

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 curiousity 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.


dont worry about it if you don't
understand

Making theories

Making theories

- Note: I am extremelly ”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

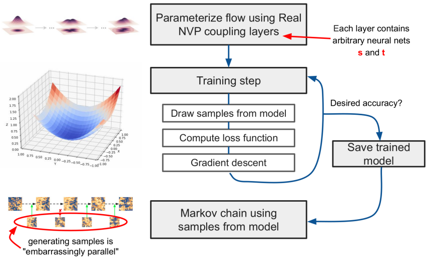
Making theories

- Note: I am extremelly "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**

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

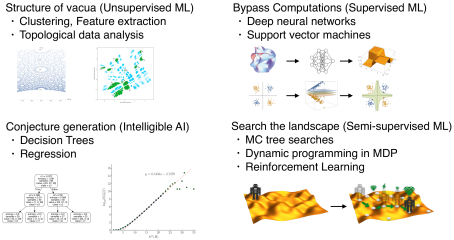
with field configurations U^i distributed according to $e^{-S[U]}$



Summary chart: Tej Kanwar

Making theories

- Note: I am extremelly ”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



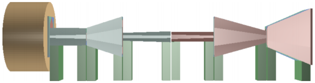
Building detector

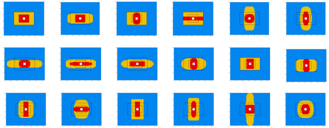
Building detector

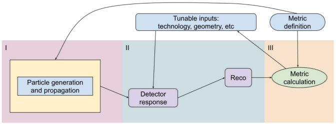
geometry

- 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 → geometry





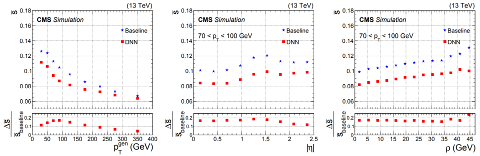


- It all starts from building a detector - which is tough, **a lot of constraints** need to be taken into account (money vs. physics)
- Can we somehow find the "optimal" geometry?
- Yes, as it's an optimisation problem
- In fact, **"black box"** optimisation → bayesian optimisation

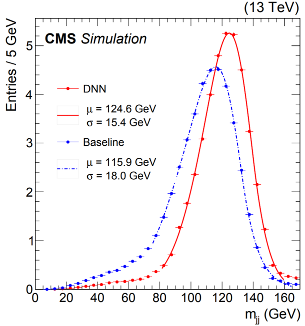
→ Muon shielding in SHiP experiment

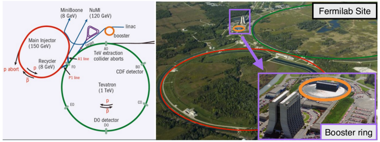
→ LHCb ECal case

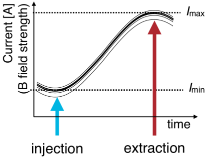
Building detector —○ calibration

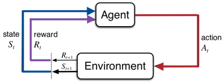


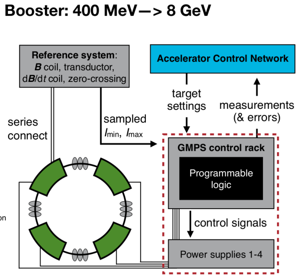
- What is also important then is to setup detector properly → **calibration**
- Take calorimeter and jet mass as observable
- Distribution has asymmetric non-Gaussian tails → MSE fit will have biased result
- Need to simultaneously predict *median* or *mode* of distribution + *quantiles* for jet energy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











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

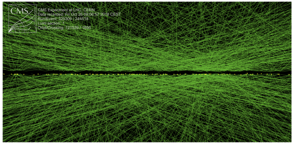
and finally, treat it with love and care

Collecting data

Collecting data

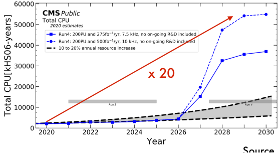
—o motivation

Ngadiuba's talk @FastML2020

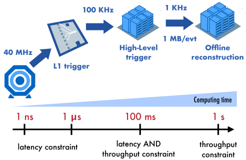


- Once collisions begin, data from the detector needs to be collected
- Huge in size *but* can't store everything
- Respond and process new data quickly *but* computational resources limited

→ Need to find the "best", select it and do this fast → requires **low latency, large throughput**



Source



Collecting data — DAQ

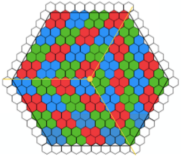
Ngadiuba's talk @FastML2020

Input
48 "trigger cells"
7b floating point
(336b total)

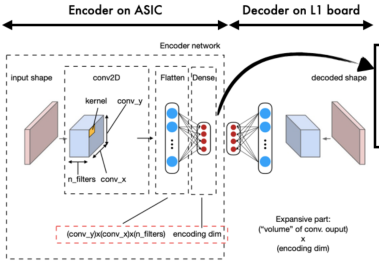
→ ASIC →

Output:
"Super trigger cell" algo
3[16 TC sum] x 13b = 39b

CMS HGCAL cells



Can we do a better job of encoding the info in those bits w/o so much loss in granularity?



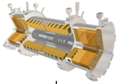
Really need quantized training here to optimize information encoding

Use QKeras!

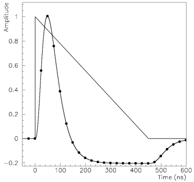
More in G. Di Gluglielmo talk Monday @ 14:50

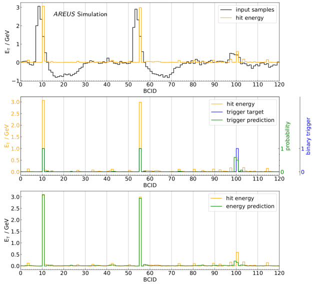
Collecting data —○ DAQ

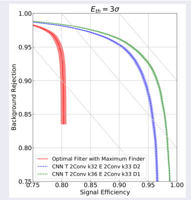
[Fritzsche's talk @FastML2020]

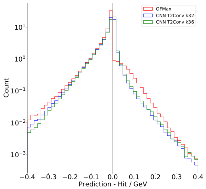


readout per cell







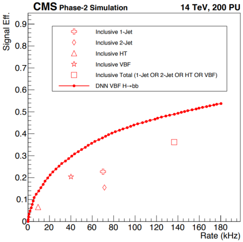


Collecting data

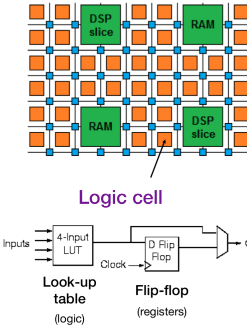
trigger

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



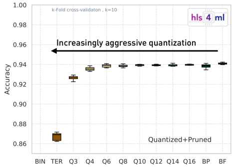
Collecting data

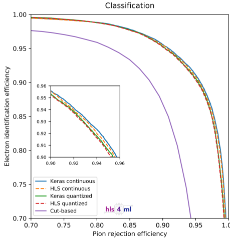
→ trigger

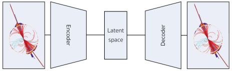
CMS-TDR-021

- 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



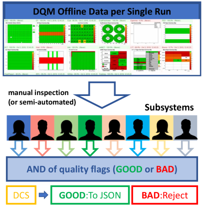


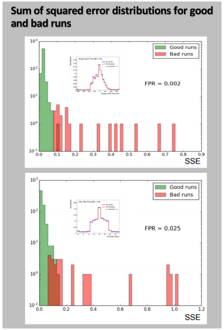


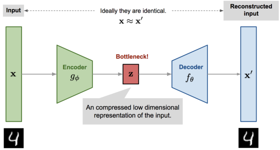
Collecting data

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

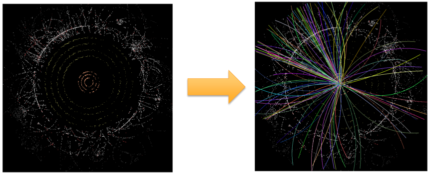




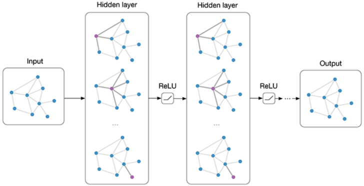


Reconstructing objects

Reconstructing objects → tracks

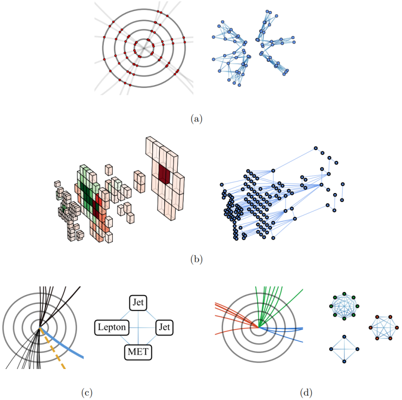


- Traditional Kalman filter will become inefficient with increased luminosity
- It all (seriously) started from TrackML hackathon on Kaggle
- Several research projects emerged, e.g. ACTS, HEP.TrkX and then Exa.TrkX as its follow-up
- Various methods probed (CNN, RNN), finally converged to **graph networks**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



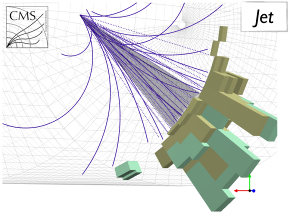
GNN in Particle Physics paper

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

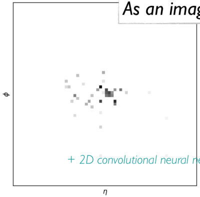


(a) (b) (c) (d)

Reconstructing objects → jets: representation

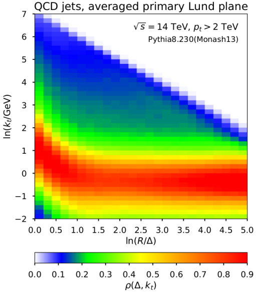


As an image?



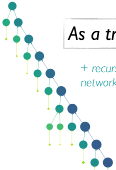
+ 2D convolutional neural network (CNN)

As a Lund plane?



As a sequence?

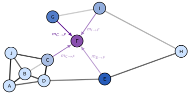
+ recurrent neural network (RNN)
+ 1D CNN
+ etc.



As a tree?

+ recursive neural network (RecNN)

As a graph?

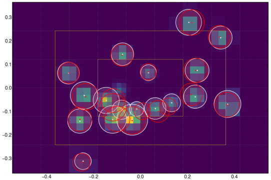


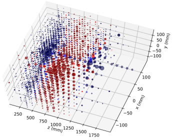
+ graph neural network (GNN)

Reconstructing objects

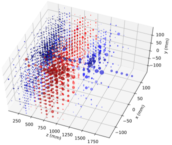
jets: segmentation

- 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 to build *one-shot ParticleFlow*





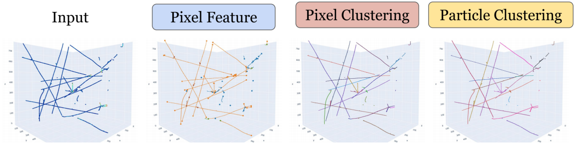
(a) Truth

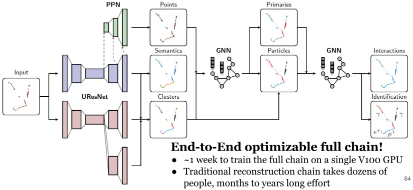


(b) Reconstructed

Reconstructing objects → bonus: neutrino

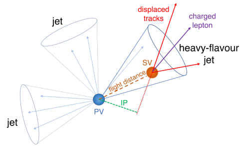
Kazuhiro's talk @IML2020

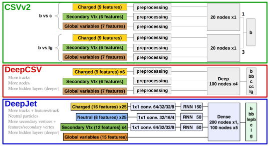


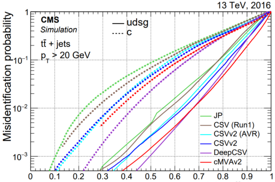


Reconstructing objects — jets: tagging

- **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

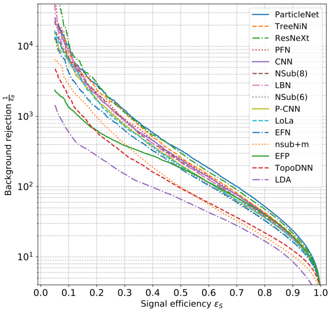




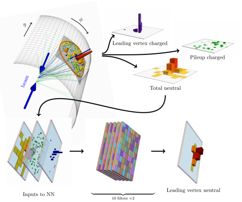


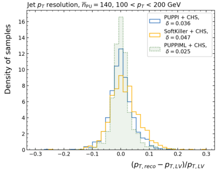
Reconstructing objects → jets: tagging

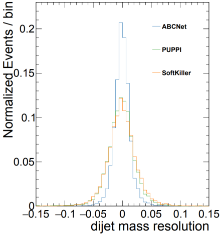
- **b-tagging** [1] , [2]
  - crucial to identify jets from b quark (e.g. $H \rightarrow b\bar{b}$)
  - ML brings significant gain ($\sim 50\%$) comparing to classical approaches
  - solid place in production (e.g. DeepJet @CMS)
  - in fact, also used for c/uds/g tagging
- **t-tagging**
  - significant piece in SM puzzle (e.g. ttH, $t\bar{t}$)
  - complex to become benchmarking playground in tagging

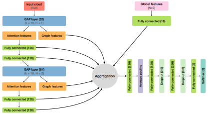


- **Pile-up mitigation**
  - remove irrelevant contributions from pile-up
  - PUMML: convolutional analysis of LV charged/PU charged/neutral components
  - PuppiML: gated graph-based architecture
  - ABCNet: graph net with attention



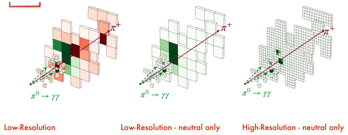


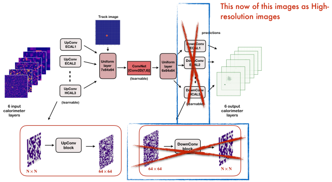


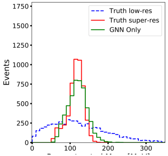


Reconstructing objects — jets: cleansing

- **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 calorimiter's granularity with CNN/GNN
  - able to nicely resolve photons in $\pi_0 \to \gamma\gamma$







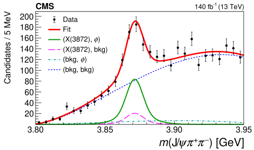
Reconstructing objects — misc

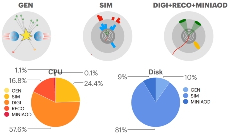
- **Vertices**
  - find ML-reconstructed (secondary) vertices in jets
  - to be further passed to tagging algorithms
- **Leptons**
  - τ: e.g. ABCNet for low-p_T, or general-purpose DeepTau
  - $\mu/e/\gamma$ (were among the first MVA ID)
- **Missing transverse energy** (MET):
  - used to approximate 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

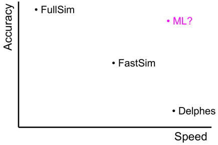
Simulating physics

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?





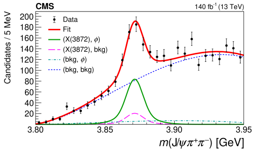


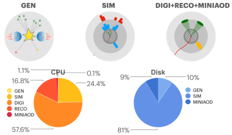
Simulating physics → motivation

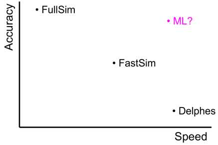
- What we want to know from data is parameters of the model we study (SM/BSM)
- Traditionally, model's behaviour at the level of observables is obtained by (aka **Monte Carlo**) simulation in the form of **templates** to be fitted to observed data
- **Need to simulate A LOT** of data (e.g. to have low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

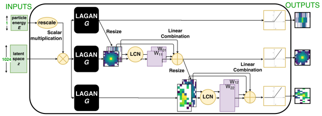
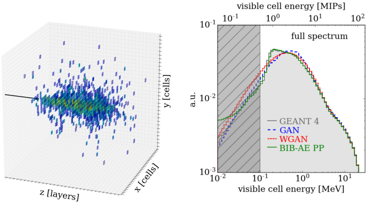


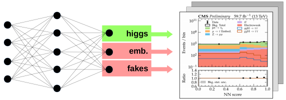
FIG. 4: Composite Generator, illustrating three stream with attentional layer-to-layer dependence.

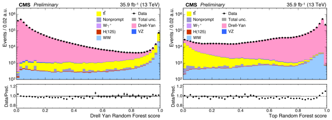


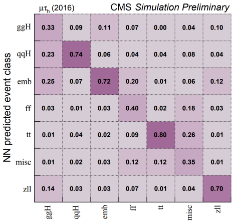
Performing analysis

Performing analysis → sig vs. bkgr

- need to classify (several) signal processes from (several) backround 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 $H \to \tau\tau$ CP study @CMS

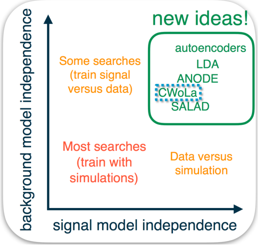


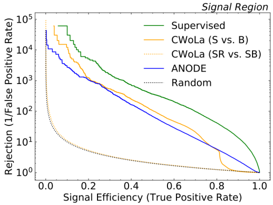




Performing analysis → anomaly detection

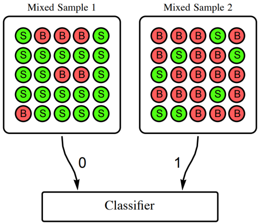
- Normally, we search for new physics having some model in mind
- But if one want to do it **model-independently** – try instead to search for *deviations* → **unsupervised learning**
- Also hot topic: ANODE , SALAD

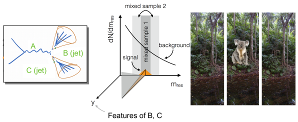




Performing analysis —o anomaly detection

- 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* $\rightarrow$ **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

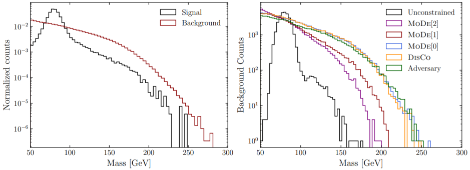
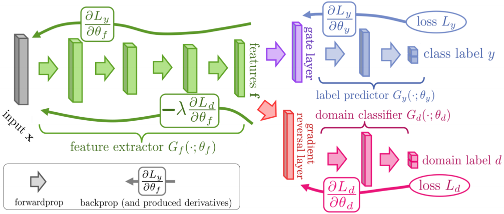


Figure 6. **Left:** Distributions of signal and background events without selection. **Right:** Background distributions at 50% signal efficiency (true positive rate) for different classifiers. The unconstrained classifier sculpts a peak at the W-boson mass, while other classifiers do not.

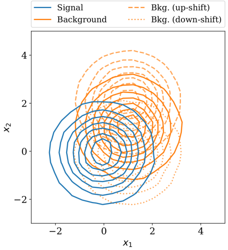
Performing analysis — decorrelation & adaptation

- It was observed, that cutting on classifier output can heavily distort the distribution of observable (e.g. invariant mass) $\rightarrow$ **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. $\rightarrow$ domain adaptation

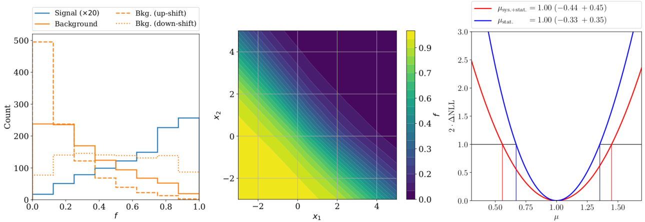


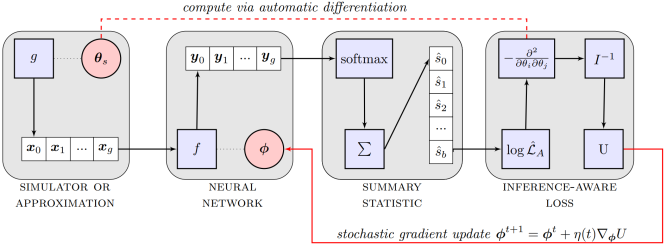
Performing analysis — nuisances & inference

- 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

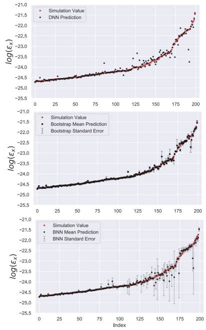
Interpretable ML book

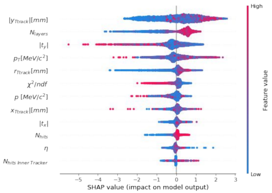
- why ML works? → hand-wavy answer here
  (aka we don't really know)
- but we can try to get insights:
  - bayesian networks to access per-jet uncertainties
  - and more bayesian for accelerators
  - SHAP values for feature importance

DNN

Ensemble

BNN





Performing analysis — misc.

- **Quantum ML**
  - e.g. quantum GANs
  - or quantum classifiers
- **Production**
  - models on your laptop are generally useless for your collegues
  - needs to be implemented into some framework
  - a whole new topic to be aware of

Contemplating future

thoughts from spring 2019

- **interpretability**
  - what is NN learning? physics, perhaps?
  - do we train it properly?
  - can we fully understand it?

Contemplating future — future directions

- **interpretability**
  - what is NN learning? physics, perhaps?
  - do we train it properly?
  - can we fully understand it?
- **incorporating physics** domain knowledge
  - systematics
  - physics laws and symmetries
  - model's place in HEP pipeline

Contemplating future — future directions

- **interpretability**
  - what is NN learning? physics, perhaps?
  - do we train it properly?
  - can we fully understand it?
- **incorporating physics** domain knowledge
  - systematics
  - physics laws and symmetries
  - model's place in HEP pipeline
- close **collaboration with ML community**
  - Kaggle
  - industry experience
  - new architectures and concepts

Contemplating future — about to come

thoughts from spring 2019

- **now**
  - simple s/b classification in analyses
  - particle ID/tagging into production (e.g. CMSSW)
  - why don't you use ML in your analysis?

Contemplating future → about to come

thoughts from spring 2019

- **now**
  - simple s/b classification in analyses
  - particle ID/tagging into production (e.g. CMSSW)
  - why don't you use ML in your analysis?
- **soon**
  - ML at trigger level
  - complex taggers (top, boosted) into production
  - centralized simulation with GANs
  - track/object reconstruction
  - architecture competition and evolution (in fact, now)

Contemplating future → about to come

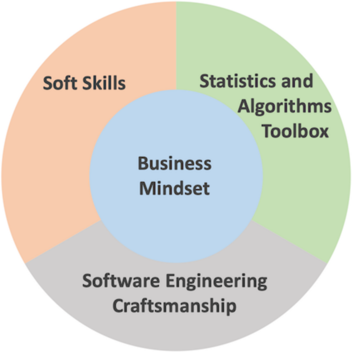
thoughts from spring 2019

- **now**
  - simple s/b classification in analyses
  - particle ID/tagging into production (e.g. CMSSW)
  - why don't you use ML in your analysis?
- **soon**
  - ML at trigger level
  - complex taggers (top, boosted) into production
  - centralized simulation with GANs
  - track/object reconstruction
  - architecture competition and evolution (in fact, now)
- **future**
  - fully inference/systematics-aware learning
  - HEP specific DL libraries
  - end-to-end training
  - continuous assimilation with ML community
  - ...?

amazing talk on ML&HEP future

Contemplating future → third wave data scientist

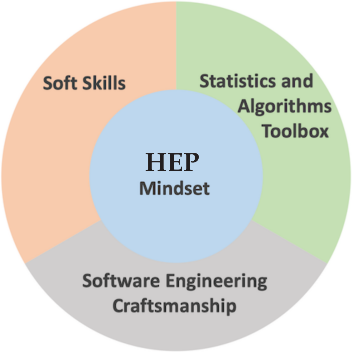
inspired by



Contemplating future

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