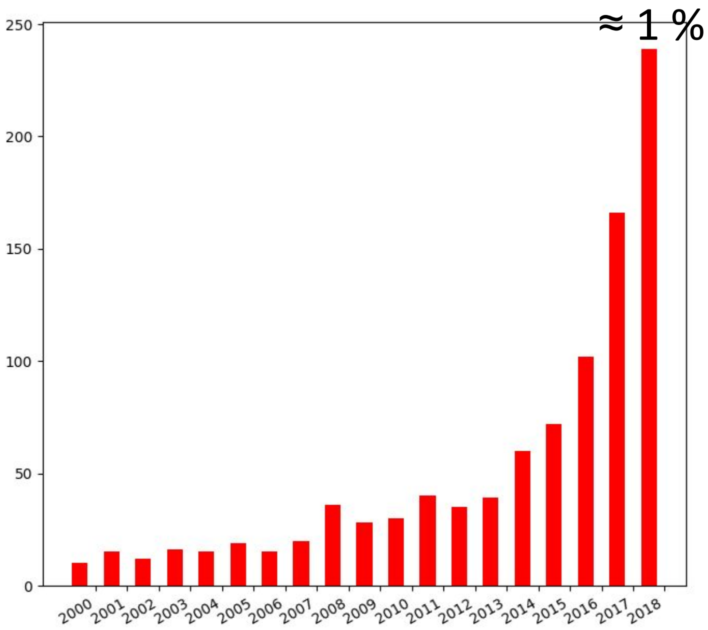
Is Machine Learning ready for HEP?

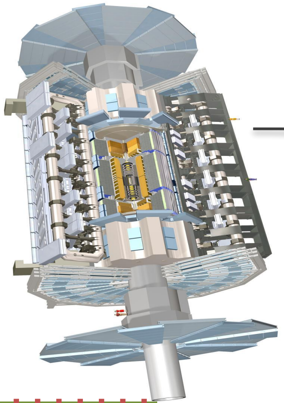
Cécile Germain
Laboratoire de Recherche en Informatique
Université Paris Sud CNRS
INRIA TAU

HEP is in ML

ArXiv physics:astro-ph and physics:hep-ex papers with *machine learning*, *deep learning* or *neural network* in the title or abstract



The discovery pipeline

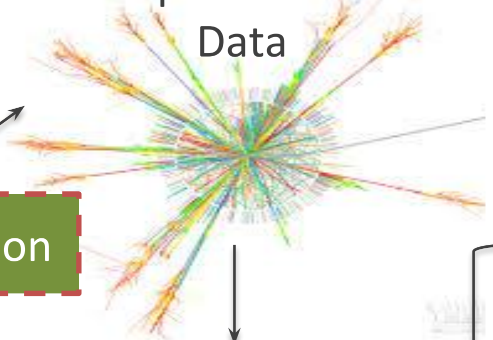


Real time selection

Tracking

Representation

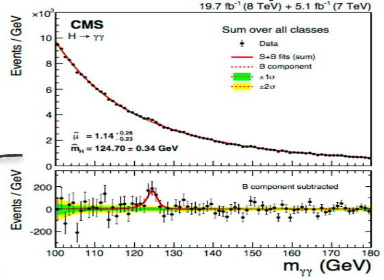
Experimental Data



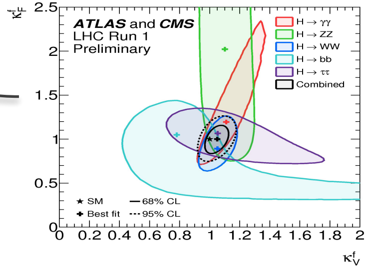
Monitoring

Analysis

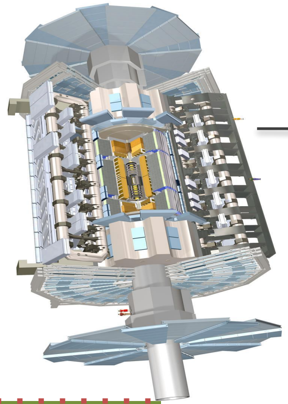
Discovery



Measurement



The discovery pipeline



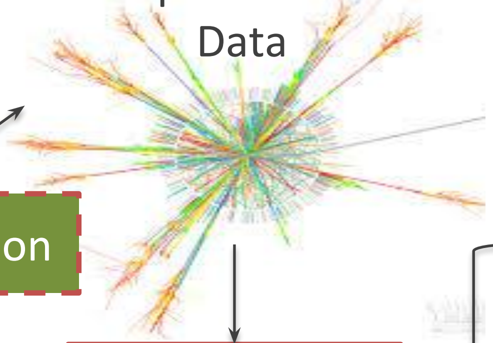
Real time selection

Tracking

Representation

Monitoring

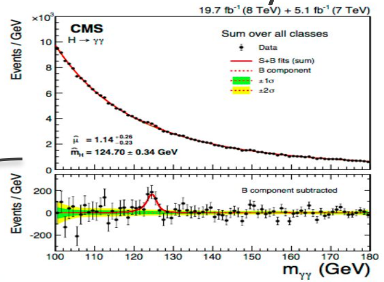
Experimental Data



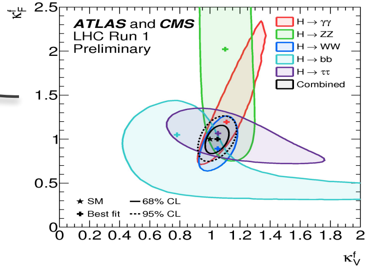
Analysis

Simulated data

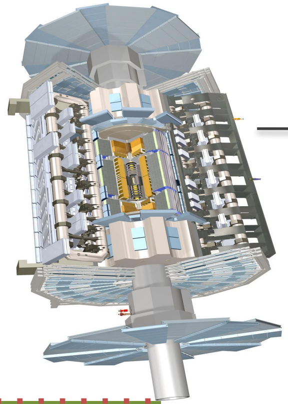
Discovery



Measurement



The discovery pipeline



Real time selection

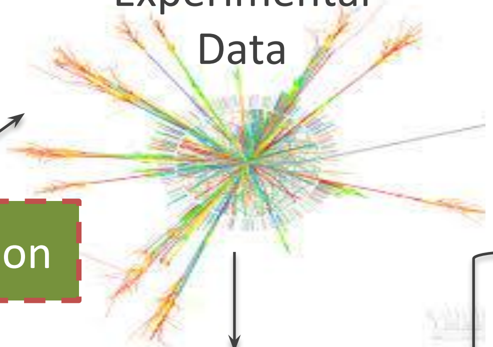
Tracking

Representation

Monitoring

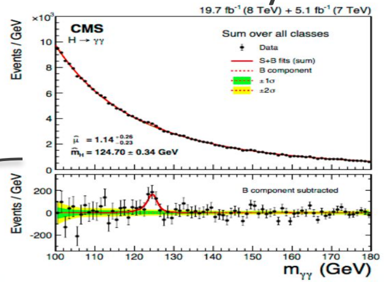
Simulated data

Experimental Data

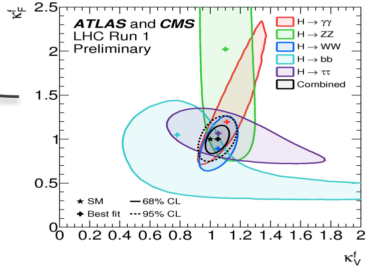


Analysis

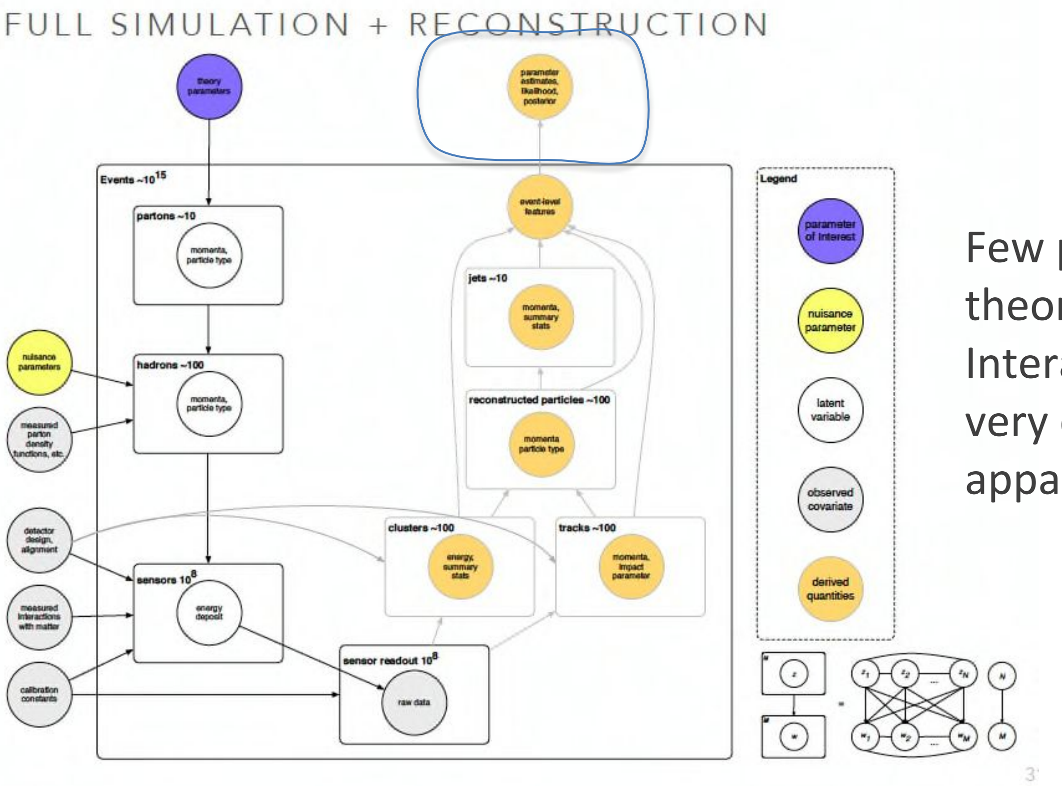
Discovery



Measurement



The simulation pipeline



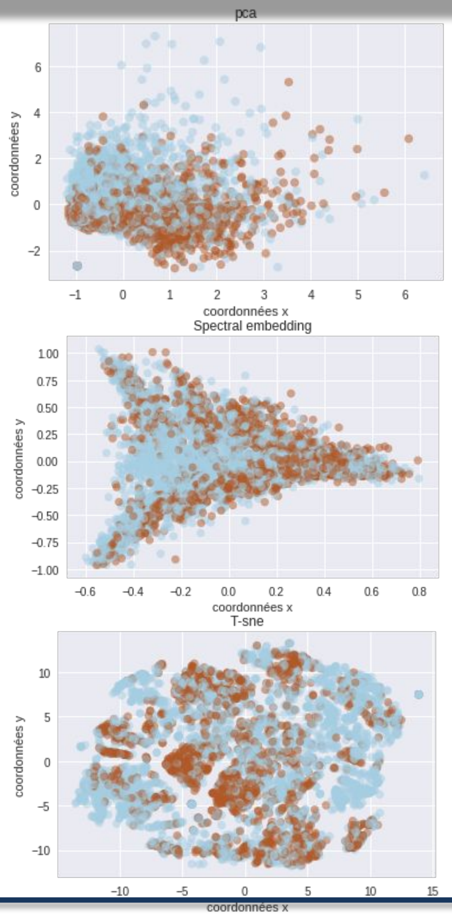
Few parameters from theory

Interaction with a very complex apparatus

[Cranmer NIPS'16]

Analysis: discovery and measurement

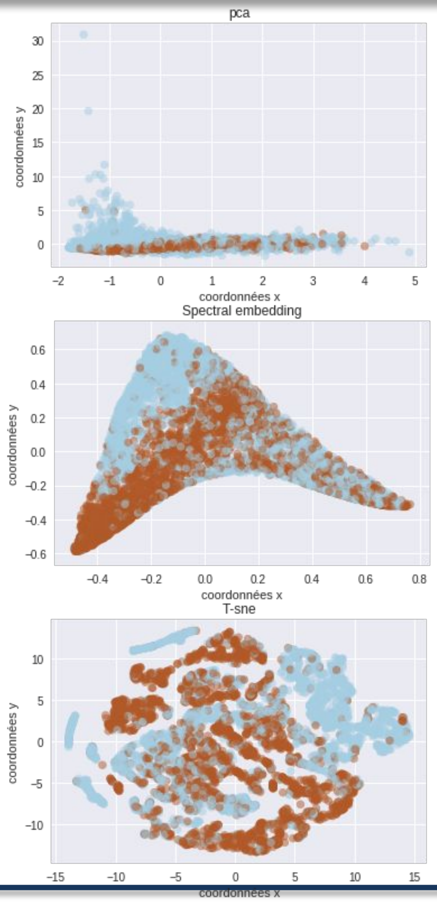
- Likelihood-free inference
    - Likelihood function $p(x|\theta)$ intractable
    - Simulator can generate samples, at a cost
- Workhorse: binary classification
    - Signal vs Background
    - Principled wrt physics objectives



ATLAS full detector simulator

Analysis: discovery and measurement

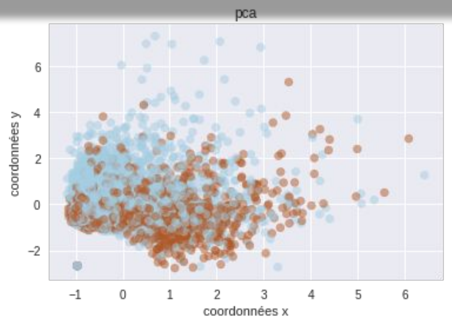
- Likelihood-free inference
  - Likelihood function $p(x|\theta)$ intractable
  - Simulator can generate samples, at a cost
- Workhorse: binary classification
  - Signal vs Background
  - Principled wrt physics objectives

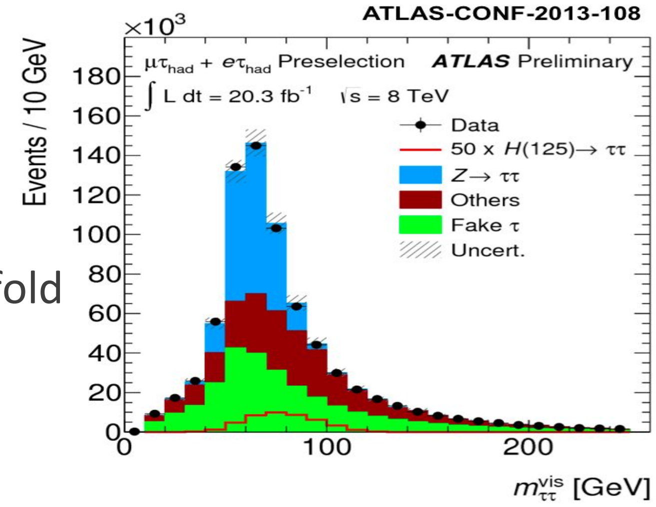


UCI Higgs dataset

Analysis: discovery and measurement

- Likelihood-free inference
  - Likelihood function $p(x|\theta)$ intractable
  - Simulator can generate samples, at a cost
- Workhorse: binary classification
  - Signal vs Background
  - Principled wrt physics objectives
- Surprisingly hard
  - "Dense and full rank": dimension of data manifold = dimension of feature space
  - Needle in a haystack



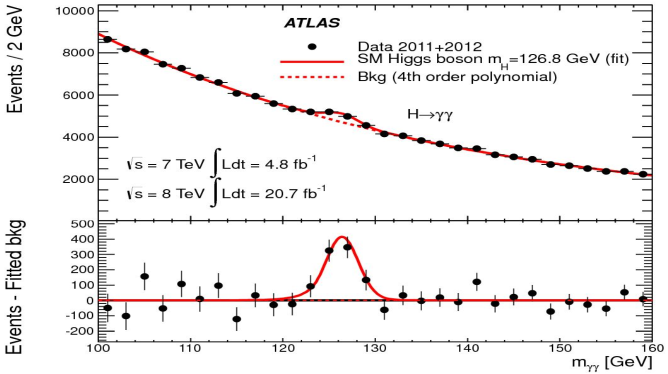


Discovery

Investigate the compliance of the data with the standard model:

statistical testing on a Poisson distribution

- Selection in the feature space: select the events that could be signal and count: N the only observable
- Does this number significantly exceed the expected number of events predicted by a background-only hypothesis?
- Test $\mu = 0$ against $\mu > 0$



$$N \sim \text{Poiss}\,(\mu s + b)$$

s (resp b): expected number of signal (resp background)

Classification for discovery

- Select the could-be signal events: binary classifier $f = (g, t)$
- Balanced dataset, weights w_i as in importance sampling

$$\mathcal{D} = \{(\mathbf{x}_1, y_1, w_1), \ldots, (\mathbf{x}_n, y_n, w_n)\}$$

- Selected signals (resp backgrounds) are True (resp False) Positives

$$s = \sum_{i \in \mathcal{S} \cap \widehat{\mathcal{G}}} w_i \qquad b = \sum_{i \in \mathcal{B} \cap \widehat{\mathcal{G}}} w_i$$

- Optimal decision rule ≈ Neymann-Pea … n
- Classifier performance evaluated on simulations

Performance metric

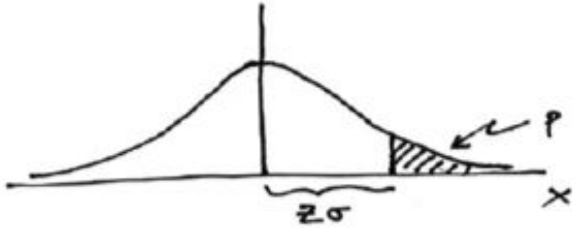
- H0 vs H1: p-value and significance

$$Z = \Phi^{-1}(1-p)$$

- Composite test: $\mu = 0$ against $\mu > 0$
- Approximate Median Significance

$$\text{AMS} = \sqrt{2\left((s+b)\ln\left(1+\frac{s}{b}\right)-s\right)} \approx \frac{s}{\sqrt{b}}$$

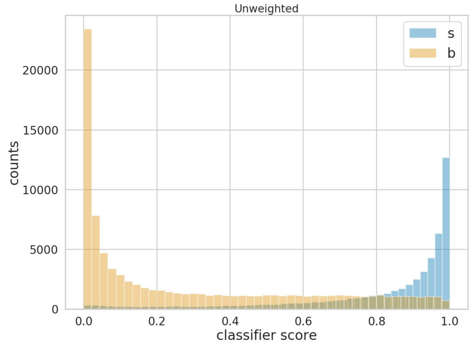
- Expected significance= AUC (Dempster 65)
- Depends only on TP and FP

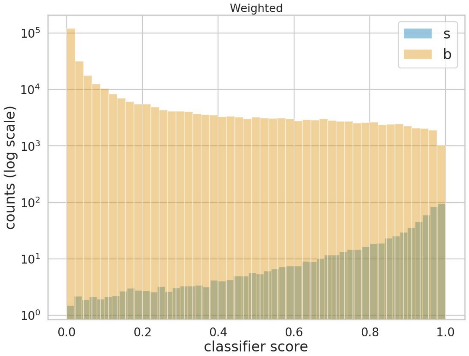


G. Cowan et al. 1007.1727

Classification

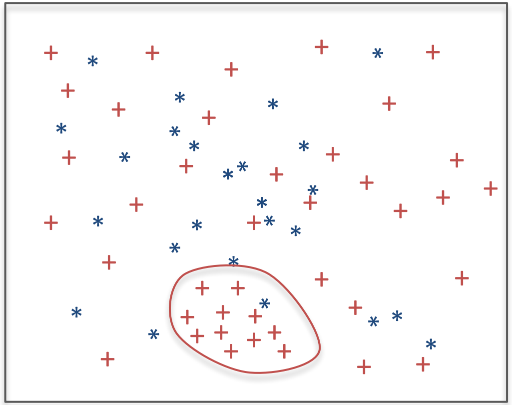
- Accuracy not relevant when the distributions are normalized to their prior probabilities
- Method
  - Consistent classifier (eg cross-entropy)
  - Optimize region threshold on the AMS



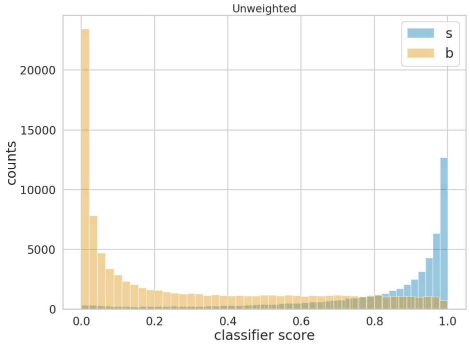


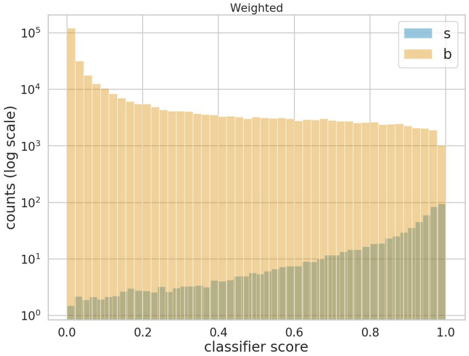
Classification

- Accuracy not relevant when the distributions are normalized to their prior probabilities
- Method
  - Consistent classifier (eg cross-entropy)
  - Optimize region threshold on the AMS

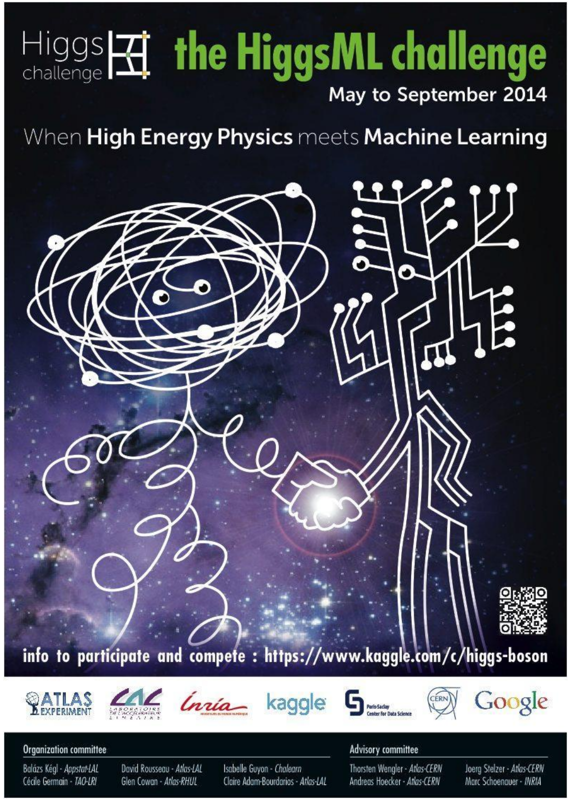


Signal-rich region





Benchmarking

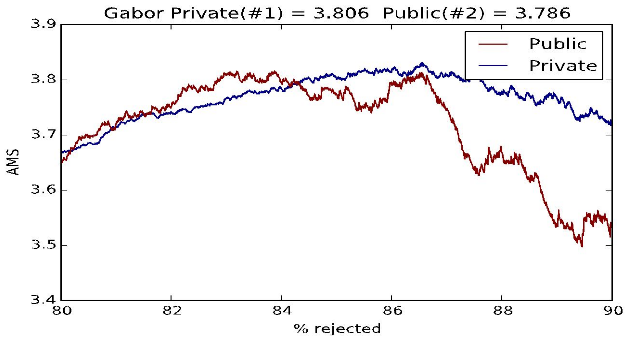


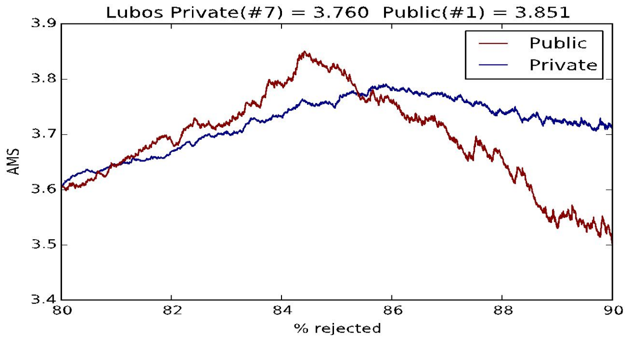
- Dataset typical of real analysis
- 40 features: summary statistics (PRI_) and engineered (DER_)
- 1M instances
- Full simulation
- Evaluated on AMS
- **Available on opendata.cern.ch**

Adam Bourdarios et al, JMLR procs

2014-2019

- No disruptive method emerged
  - Direct optimization of the AMS overfits
  - "D"NN and gradient boosting in the same league
- But ML clear winners
- ML technical expertise critical
  - Feature construction (physics) marginal improvement
  - Efficient cross-validation and bagging decisive
- 2019: technical expertise available in standard process





Systematic errors

- Causes: “known unknowns”
  - Known = can be included in the simulation pipeline, typically as nuisance parameters
  - As opposed to “statistical” error, eg capacity or finite size for the classifier tool
- Decreases sensitivity of analysis to the parameter of interest = wider uncertainty on estimates
- So far, mostly not integrated in selection – upper bound of ML usefulness in LHC analysis

Source of uncertainty		σ_μ
Total		0.39
Statistical		0.24
Systematic		0.31
Experimental uncertainties		
Jets		0.03
E_T^{miss}		0.03
Leptons		0.01
b-tagging	b-jets	0.09
	c-jets	0.04
	light jets	0.04
	extrapolation	0.01
Pile-up		0.01
Luminosity		0.04
Theoretical and modelling uncertainties		
Signal		0.17
Floating normalisations		0.07
Z+jets		0.07
W+jets		0.07
$t\bar{t}$		0.07
Single top-quark		0.08
Diboson		0.02
Multijet		0.02
MC statistical		0.13

ATLAS-CONF-2017-041

Systematics on discovery

- Standard statistical tool: profile likelihood ratio

$$\Lambda(\mu) = \frac{L(\mu, \hat{\hat{\alpha}})}{L(\hat{\mu}, \hat{\alpha})}$$

- Its distribution is asymptotically independent of nuisance parameters α

- Discovery significance for a counting experiment with gaussian uncertainty on background

$$\left[2\left((s+b)\ln\left[\frac{(s+b)(b+\sigma_b^2)}{b^2+(s+b)\sigma_b^2}\right] - \frac{b^2}{\sigma_b^2}\ln\left[1+\frac{\sigma_b^2 s}{b(b+\sigma_b^2)}\right]\right)\right]^{1/2} \cong \frac{s}{\sqrt{b+\sigma_b^2}}$$

Cowan et al. 1007.1727

Elwood et al. 1806.00322, Xia 1810.08387

Systematics and measurement

$N \sim \text{Poiss}(\mu s(\alpha) + b(\alpha))$

Typically $\alpha \sim \prod_i \text{Normal}(m_i, \tau_i)$

- Cross-section μ, normalized to the nominal
- Minimize the relative measurement error:
  - Point-wise classification: estimate mu with error-oriented regularization

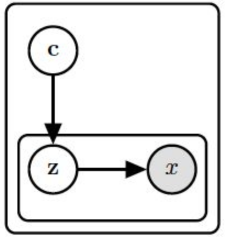
$$\frac{\sigma_\mu}{\mu} = \sqrt{\sigma_{stat}^2 + \sigma_{syst}^2}$$

σ_{stat}: error on the nominal
σ_{syst}:impact of the systematics

  - Or, summary (sufficient) statistics: learn a data-set level representation minimizing the confidence interval

ML Contexts

- Pattern recognition: enforce invariance wrt parameterized known transforms
  - Tangent Prop [simard91], invariance
- The context matters [Edwards17]
  - Topic Model



- Transfer learning
- Fairness [Hardt], with GAN [Louppe], with VAE [Mathieu][Louizos]
  - Demographic parity: independence

$$P(g(X)=v\,|\,z)=P(g(X)=v\,|\,z')$$

  - Equalized odds/opportunity: independent conditional on the class/some class

$$P(g(X)=v\,|\,z,y)=P(g(X)=v\,|\,z',y)$$

Systematics and measurement

$N \sim \text{Poiss}\,(\mu s(\alpha) + b(\alpha))$

Typically $\alpha \sim \prod_i \text{Normal}(m_i, \tau_i)$

- Cross-section μ, normalized to the nominal
- Minimize the relative measurement error:
  - Point-wise classification: estimate mu with error-oriented regularization

$$\frac{\sigma_\mu}{\mu} = \sqrt{\sigma_{stat}^2 + \sigma_{syst}^2} \qquad \sigma_{stat} = \frac{\sqrt{s_0 + b_0}}{s_0} \qquad \sigma_{syst} = \frac{s_z + b_z - s_0 - b_0}{s_0}$$

  - Or, summary (sufficient) statistics: learn a data-set level representation minimizing the confidence interval

Point-wise invariance

Tangent propagation

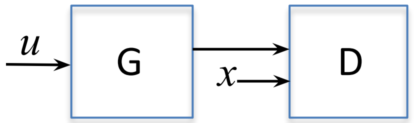
- Output of the model should be invariant according to some known transformation T of the input
- Regularize the derivative of the model according to the parameter of the transformation

$$l(x) = l_{usual}(x) + \lambda \left\| \frac{\partial g(T(x,z))}{\partial z} \right\|_{z=0}^{2}$$

- Data efficient

Pivot Adversarial Network [Louppe et al.]

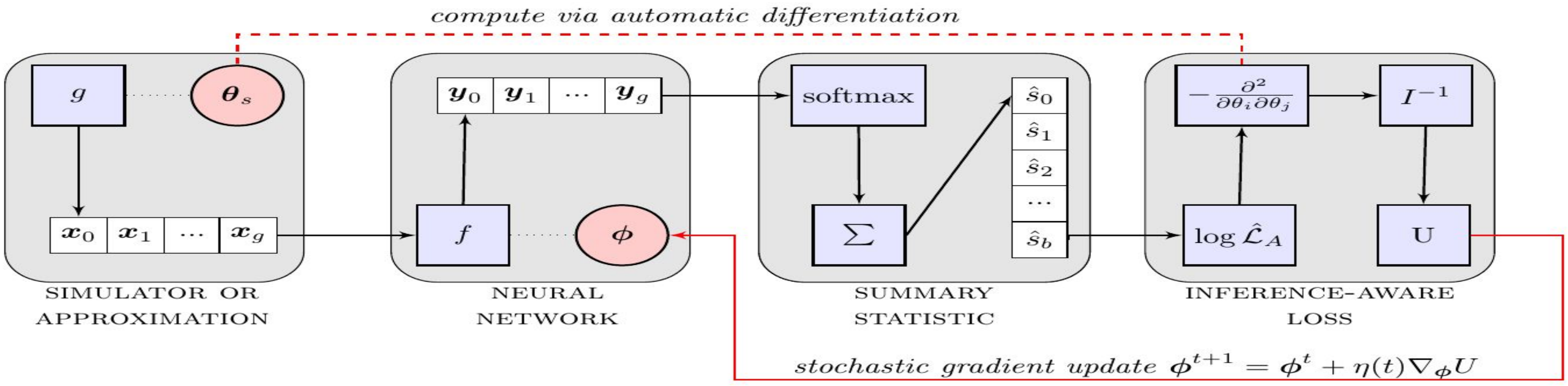
- GAN: learn the (regularized) objective function itself [Goodfellow]



$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

- Generator: distribution of the classification score g
- Discriminator: reconstruct z from g
- Principled, but data intensive

Summary statistics: INFERNO



Learn a representations of the dataset defined by context θ_s

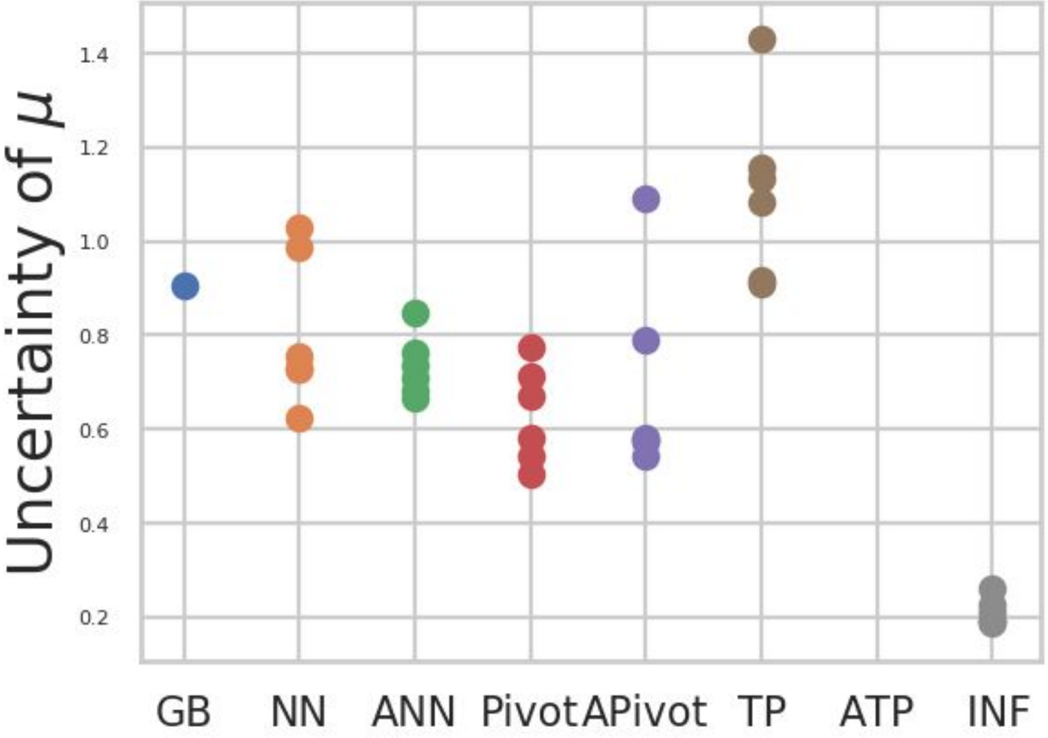
$$L_\phi(\mu,\alpha) = \prod_{i=1}^{b} \mathrm{Poiss}(\hat{s}_i)$$

Loss function $I^{-1}_{kk} \leq \mathrm{var}(\mu)$ accounts for the effect of the nuisance parameters

Laplace approximation

De Castro et al. 1806.04743

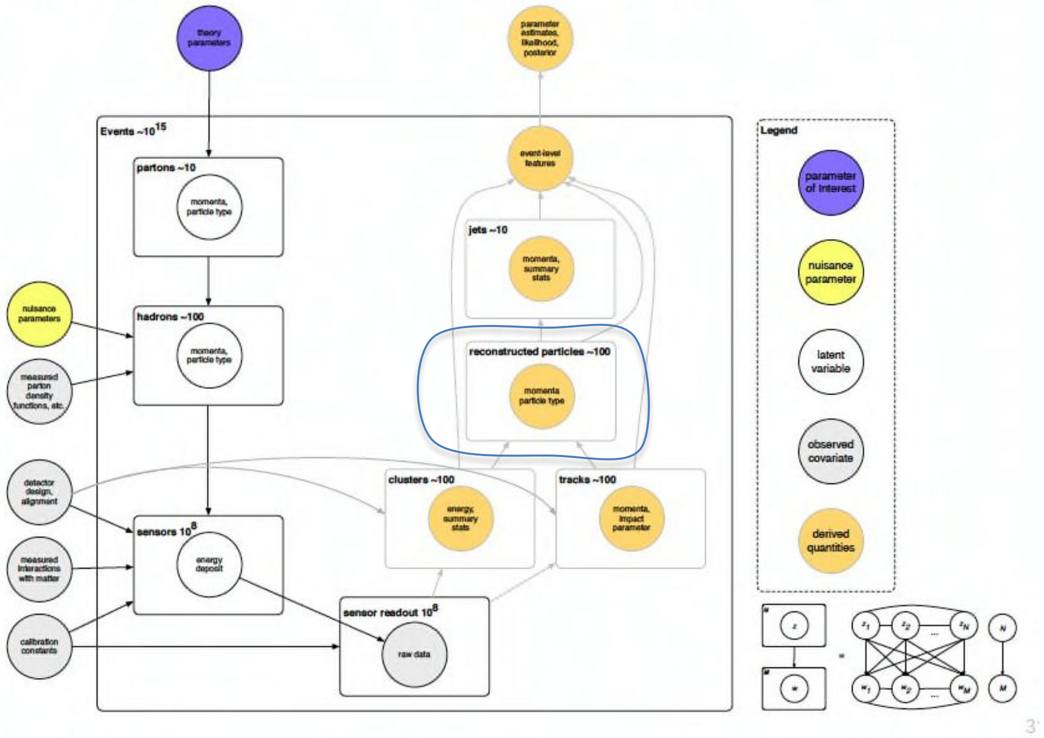
Preliminary results



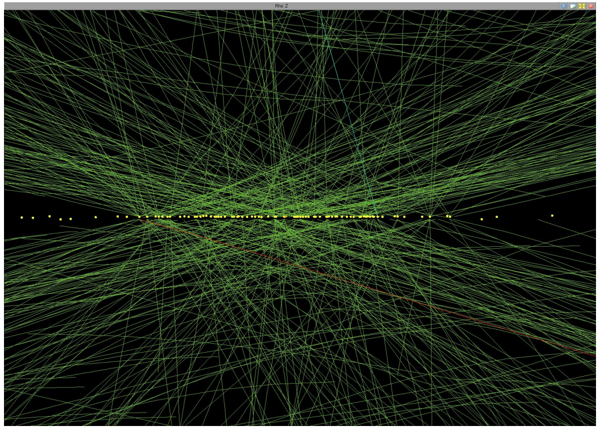
Systematic	mean	std
Tau ES	1.0	0.05
Jet ES	1.0	0.05
Lep ES	1.0	0.01
Soft term	2.7	0.5
Nast Bkg	1.0	0.5

The HEP pipeline

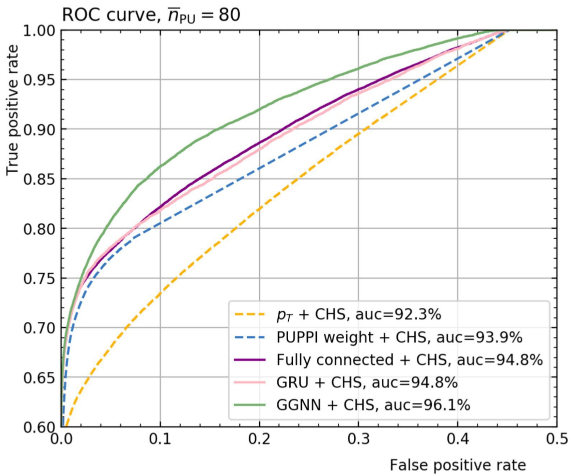
FULL SIMULATION + RECONSTRUCTION

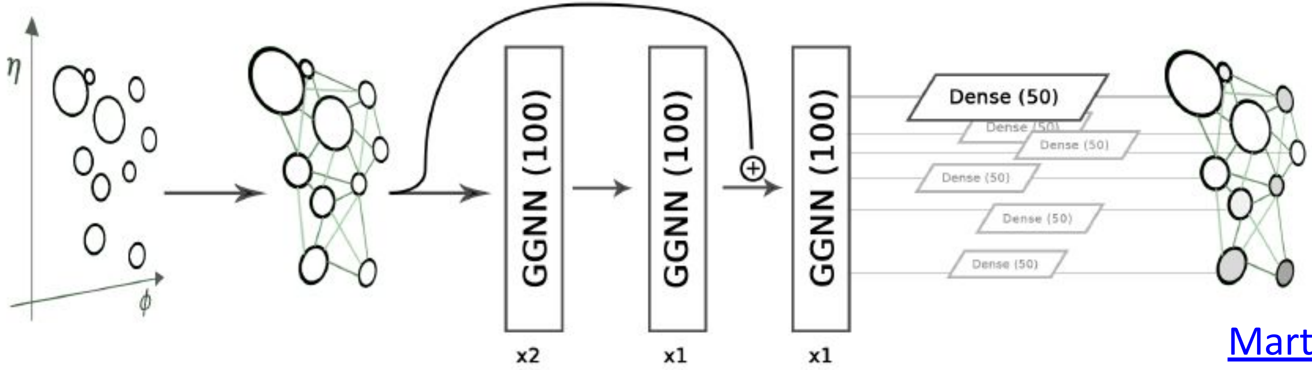


Pile-up mitigation with Graph Neural Networks



Classify particles created by the interesting collision against parasitic ones





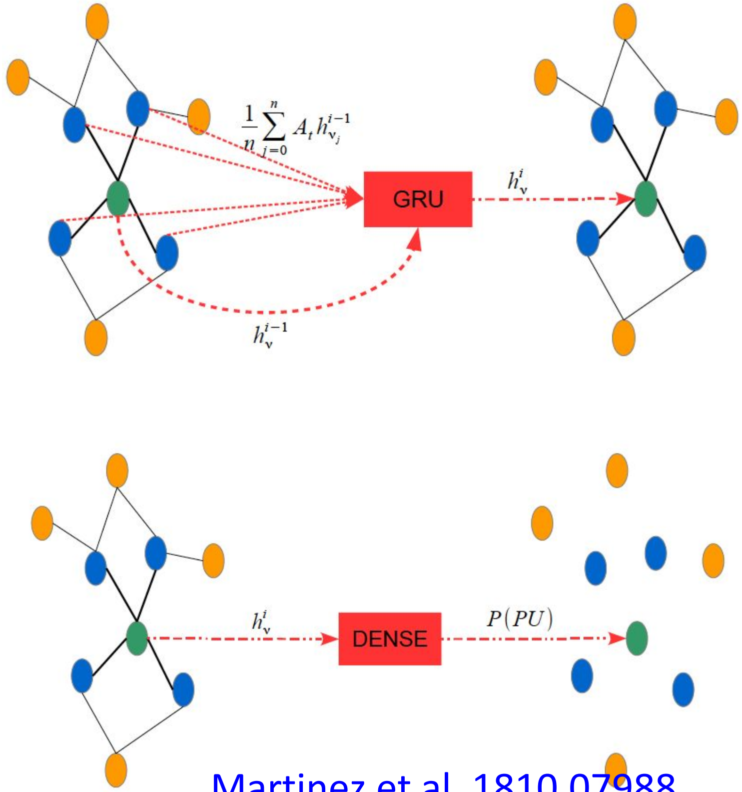
Martinez et al. 1810.07988

Pile-up mitigation with Graph Neural Networks

- Edges defined by distance in the (η,φ) plane
- Message propagation

$$GRU\left(h_{v}^{i-1},\frac{1}{n}\sum_{j=0}^{n}A_{t}h_{v_j}^{i-1}\right)\rightarrow h_{v}^{i}$$

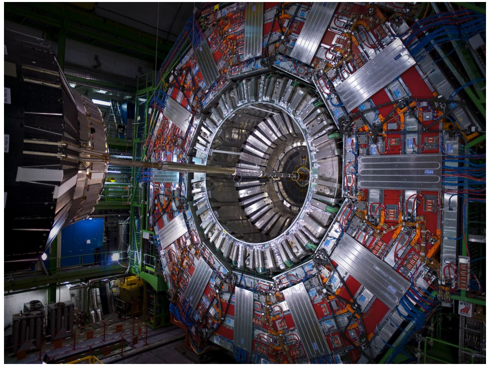
- Classification based on internal representation on each node



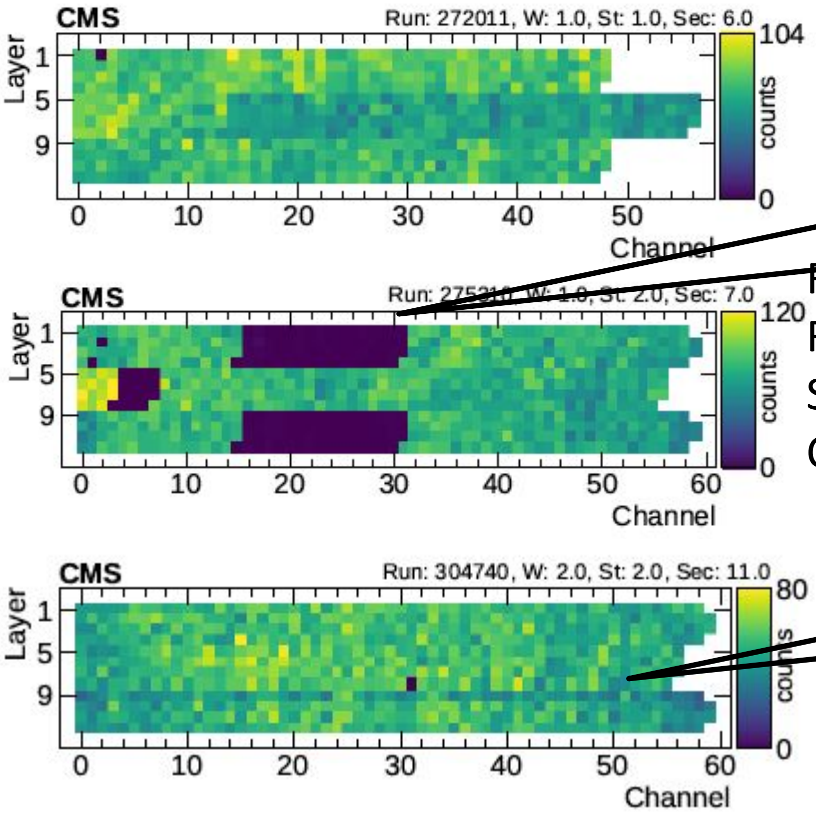
Martinez et al. 1810.07988

Data quality monitoring

- Very summary statistics selected to detect known failure modes.
- Monitored by detector experts, with predetermined validation guide lines
- On-line
  - Subsampled data, raw sensors output
  - Identify failed elements and raise alarm
- Off-line
  - On full data, with physics interpretation
  - Larger granularity



Automating on line DQM



OK

Expected: small variance amongst (6) consecutive channels

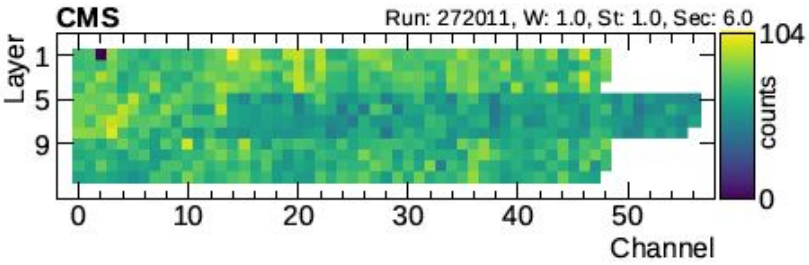
Failed
Frequent
Supervised
CNN

Expected: small variance amongst (6) layers

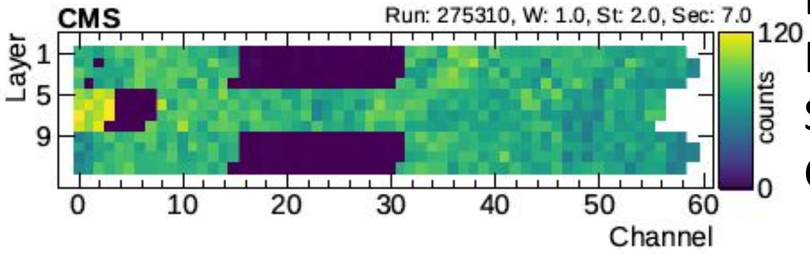
Failed
Rare
Unsupervised
CAE

1808.00911

Automating on line DQM



OK

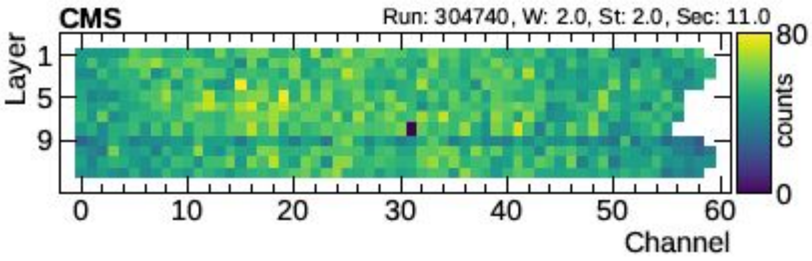


Failed

Frequent

Supervised

CNN

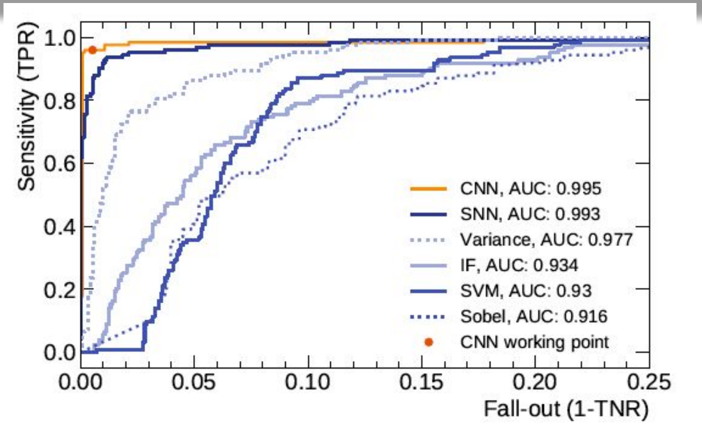


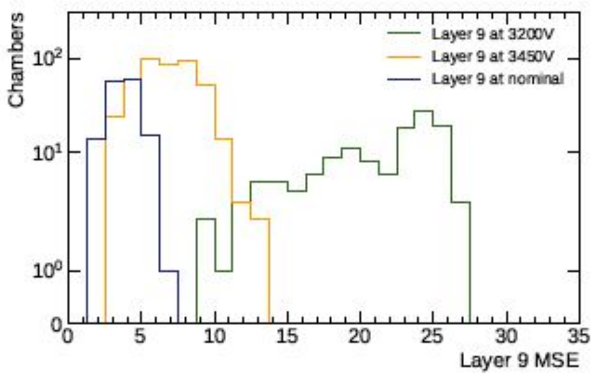
Failed

Rare

Unsupervised

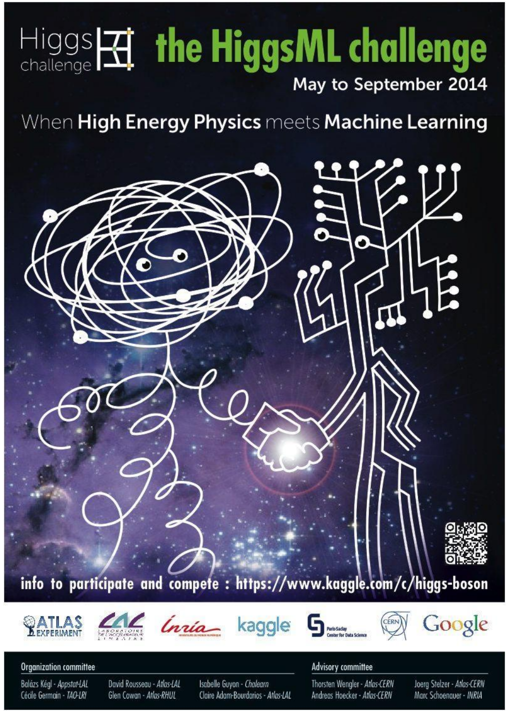
CAE

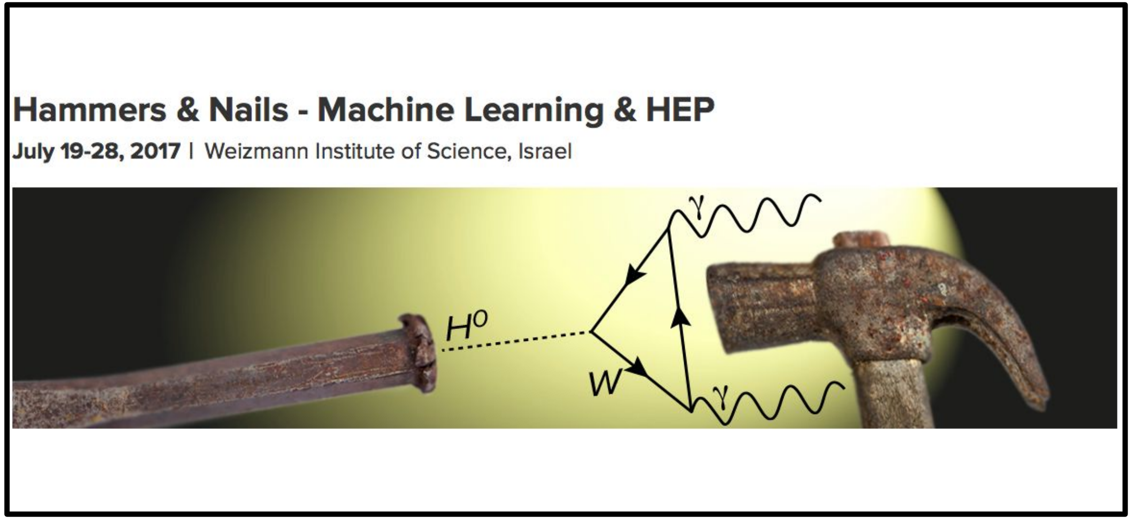




[1808.00911](https://arxiv.org/abs/1808.00911)

Conclusion & Questions





Approximate Median Significance

$$n \sim \mathrm{Poiss}(\mu s + b)$$

$$\mathrm{MLE}: \hat{\mu} = \frac{n-b}{s}$$

$$\text{Profile LR}: \lambda(0) = \frac{L(0)}{L(\hat{\mu})}$$

$$\text{Test statistic } q_0 = -2\ln\lambda(0) \text{ if } n > b, 0 \text{ otherwise}$$

$$= -2(n\ln\frac{b}{n} - n + b)$$

Asymptotically (Wilks) $q_0 \sim$ chi-2.

thus $p = 1 - \Phi(\sqrt{q_0})$ and $Z = \Phi^{-1}(1-p) = \sqrt{q_0}$

AMS=Median$(Z \mid s)$

With $n = s + b$,

$$\mathrm{AMS} = \sqrt{2\left[(s+b)\ln(1+\frac{s}{b}) - s\right]}$$

GNN

$$\mathbf{h}_v = f(\mathbf{x}_v, \mathbf{x}_{co[v]}, \mathbf{h}_{ne[v]}, \mathbf{x}_{ne[v]})$$

- Xv: node v features, idem ne, idem co(incoming edge)
- hv: internal representation
- output depnds on internal state and state

$$\mathbf{o}_v = g(\mathbf{h}_v, \mathbf{x}_v)$$

GNN

$$\mathbf{h}_v = f(\mathbf{x}_v, \mathbf{x}_{co[v]}, \mathbf{h}_{ne[v]}, \mathbf{x}_{ne[v]})$$

- Xv: node v features, idem ne, idem co(incoming edge), hv: internal representation
- Each node is represented by an aggregation of its neighborhood

Algorithm 1: GraphSAGE embedding generation (i.e., forward propagation) algorithm

Input : Graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$; input features $\{\mathbf{x}_v, \forall v \in \mathcal{V}\}$; depth K; weight matrices $\mathbf{W}^k, \forall k \in \{1, ..., K\}$; non-linearity σ; differentiable aggregator functions $\text{AGGREGATE}_k, \forall k \in \{1, ..., K\}$; neighborhood function $\mathcal{N} : v \rightarrow 2^{\mathcal{V}}$

Output : Vector representations $\mathbf{z}_v$ for all $v \in \mathcal{V}$

```
h_v^0 ← x_v, ∀v ∈ V ;
for k = 1...K do
    for v ∈ V do
        h_N(v)^k ← AGGREGATE_k({h_u^{k-1}, ∀u ∈ N(v)});
        h_v^k ← σ(W^k · CONCAT(h_v^{k-1}, h_N(v)^k))
    end
    h_v^k ← h_v^k / ||h_v^k||_2, ∀v ∈ V
end
z_v ← h_v^K, ∀v ∈ V
```