AI in Fundamental Physics - Hype or Future?

Tilman Plehn

Universität Heidelberg

PASCOS 2025, Durham

ML in LHC physics

Well-defined questions

- dark matter?
- matter vs antimatter?
- origin of Higgs boson?

Strengths

- fundamental questions
- huge data set
- first-principle, precision simulations

First-principle simulations

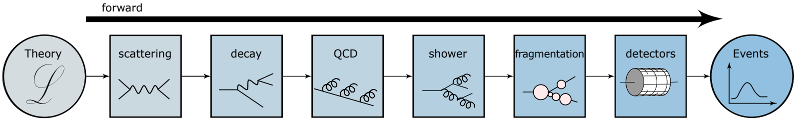
- start with Lagrangian
- calculate scattering
- simulate collisions [Durham...]
- simulate detectors

→ LHC events in virtual worlds

Searches and measurements

- compare simulations and data
- infer underlying theory [SM or BSM]
- publish and re-interpret data

→ Understand LHC data holistically



ML-intro

Similar to fit

- approximate $f_\theta(x) \approx f(x)$
- x phase/feature spac
- f_θ numerical function
- θ latent data representation

All about densities

- regression $x \to f_\theta(x)$
- classification $x \to p_\theta(x) \in [0, 1]$ [Neyman-Pearson]
- generation $r \sim \mathcal{N} \to x \sim p_\theta(x)$
- conditional generation $r \sim \mathcal{N} \to x \sim p_\theta(x|y)$

Fundamental physics

- established challenges and benchmarks
- accuracy, control, uncertainties the goal
- symmetries, constraints known

$\to$ Data complexity + volume

Network training

Learned scalar field $f_\theta(x) \approx f(x)$

- maximize parameter probability given (f_j, σ_j)

$$\theta = \text{argmax}\ p(\theta|x) = \text{argmax}\ \frac{p(x|\theta)\ p(\theta)}{p(x)}$$

$\to$ Gaussian likelihood loss

$$p(x|\theta) \propto \prod_j \exp\left(-\frac{|f_j - f_\theta(x_j)|^2}{2\sigma_j^2}\right)$$

$$\Rightarrow \qquad \mathcal{L} \equiv -\log p(x|\theta) = \sum_j \frac{|f_j - f_\theta(x_j)|^2}{2\sigma_j^2}$$

Network training

Learned scalar field $f_\theta(x) \approx f(x)$

- maximize parameter probability given (f_j, σ_j)

$$\theta = \text{argmax } p(\theta|x) = \text{argmax } \frac{p(x|\theta)\, p(\theta)}{p(x)}$$

→ Gaussian likelihood loss

$$p(x|\theta) \propto \prod_j \exp\left(- \frac{|f_j - f_\theta(x_j)|^2}{2\sigma_j^2} \right)$$

$$\Rightarrow \qquad \mathcal{L} \equiv -\log p(x|\theta) = \sum_j \frac{|f_j - f_\theta(x_j)|^2}{2\sigma_j^2}$$

Learned uncertainties

- loss including normalization

$$\mathcal{L} = \frac{|f(x) - f_\theta(x)|^2}{2\sigma_\theta(x)^2} + \log \sigma_\theta(x) + \cdots$$

- Bayesian ML

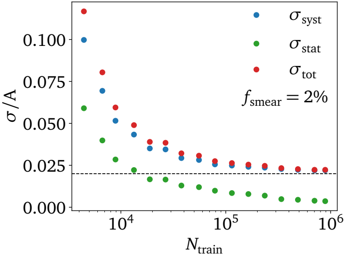
$$\sigma(x)^2 = \int dA \int d\theta \;\; (A - \langle A \rangle)^2 \;\; p(A|\theta)\; q(\theta) \equiv \sigma_{\text{syst}}(x)^2 + \sigma_{\text{stat}}(x)^2$$

→ Learning function and uncertainties

Amplitudes with calibrated uncertainties

Loop amplitude $gg \to \gamma\gamma g(g)$ [Bahl, Elmer, Favaro, Haußmann, TP, Winterhalder]

- systematics: artificial noise
- statistics plateau

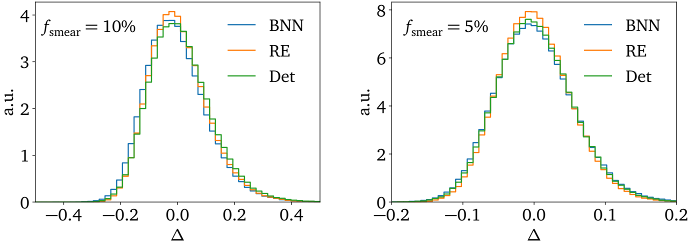


Amplitudes with calibrated uncertainties

Loop amplitude $gg \to \gamma\gamma g(g)$ [Bahl, Elmer, Favaro, Haußmann, TP, Winterhalder]

- systematics: artificial noise
- statistics plateau
- accuracy over phase space

$$\Delta(x) = \frac{A_{\text{NN}}(x) - A_{\text{true}}(x)}{A_{\text{true}}(x)}$$



Amplitudes with calibrated uncertainties

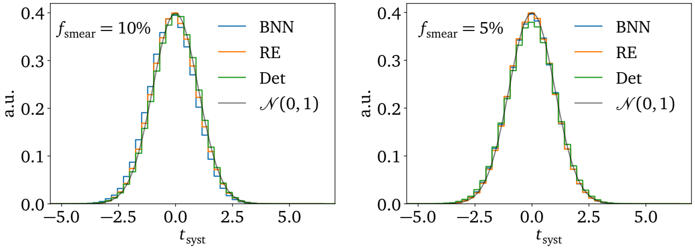
Loop amplitude $gg \to \gamma\gamma g(g)$ [Bahl, Elmer, Favaro, Haußmann, TP, Winterhalder]

- systematics: artificial noise
- statistics plateau
- accuracy over phase space

$$\Delta(x) = \frac{A_{\text{NN}}(x) - A_{\text{true}}(x)}{A_{\text{true}}(x)}$$

- pull over phase space

$$t_{\text{syst}}(x) = \frac{A_{\text{NN}}(x) - A_{\text{true}}(x)}{\sigma_{\text{syst}}(x)}$$



Amplitudes with calibrated uncertainties

Loop amplitude $gg \to \gamma\gamma g(g)$ [Bahl, Elmer, Favaro, Haußmann, TP, Winterhalder]

- systematics: artificial noise
- statistics plateau
- accuracy over phase space

$$\Delta(x) = \frac{A_{\text{NN}}(x) - A_{\text{true}}(x)}{A_{\text{true}}(x)}$$

- pull over phase space

$$t_{\text{syst}}(x) = \frac{A_{\text{NN}}(x) - A_{\text{true}}(x)}{\sigma_{\text{syst}}(x)}$$

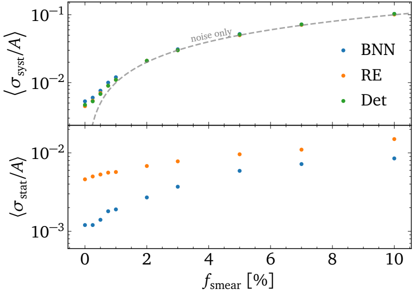
Towards zero noise

- scaling

$$\sigma_{\text{syst}}^2 - \sigma_{\text{syst},0}^2 \approx \sigma_{\text{train}}^2$$

- plateau $\langle \sigma_{\text{syst}}/A \rangle \sim 0.4\%$

$\to$ Limiting factor??



Amplitudes with calibrated uncertainties

Loop amplitude $gg \to \gamma\gamma g(g)$ [Bahl, Elmer, Favaro, Haußmann, TP, Winterhalder]

- systematics: artificial noise
- statistics plateau
- accuracy over phase space

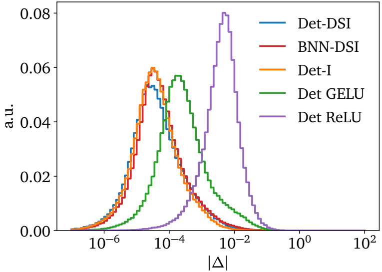
$$\Delta(x) = \frac{A_{\text{NN}}(x) - A_{\text{true}}(x)}{A_{\text{true}}(x)}$$

- pull over phase space

$$t_{\text{syst}}(x) = \frac{A_{\text{NN}}(x) - A_{\text{true}}(x)}{\sigma_{\text{syst}}(x)}$$

Data representation [cf Maitre+Sherpa]

- amplitude from invariants
- learn Minkowski metric
- Deep-sets-invariant network L-GATr transformer



Amplitudes with calibrated uncertainties

Loop amplitude $gg \to \gamma\gamma g(g)$ [Bahl, Elmer, Favaro, Haußmann, TP, Winterhalder]

- systematics: artificial noise
- statistics plateau
- accuracy over phase space

$$\Delta(x) = \frac{A_{\text{NN}}(x) - A_{\text{true}}(x)}{A_{\text{true}}(x)}$$

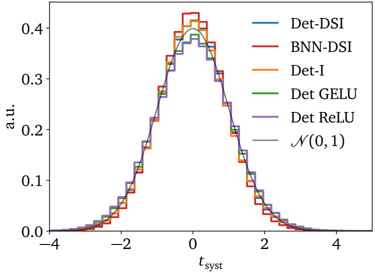
- pull over phase space

$$t_{\text{syst}}(x) = \frac{A_{\text{NN}}(x) - A_{\text{true}}(x)}{\sigma_{\text{syst}}(x)}$$

Data representation [cf Maitre+Sherpa]

- amplitude from invariants
- learn Minkowski metric
- Deep-sets-invariant network
  L-GATr transformer

$\to$ Calibrated systematics



Amplitudes with calibrated uncertainties

Loop amplitude $gg \to \gamma\gamma g(g)$ [Bahl, Elmer, Favaro, Haußmann, TP, Winterhalder]

- systematics: artificial noise
- statistics plateau
- accuracy over phase space

$$\Delta(x) = \frac{A_{\text{NN}}(x) - A_{\text{true}}(x)}{A_{\text{true}}(x)}$$

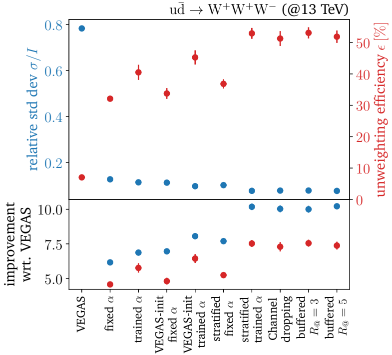
- pull over phase space

$$t_{\text{syst}}(x) = \frac{A_{\text{NN}}(x) - A_{\text{true}}(x)}{\sigma_{\text{syst}}(x)}$$

Madgraph7, Sherpa, MLhad,...

- surrogate amplitudes
  high multiplicities, loops
- multi-channel importance sampling
- end-to-end generators
- tied to detector simulations
- GPU-ready

→ Ultra-fast event generation



ATLAS calibration

Energy calibration with uncertainties [ATLAS + TP, Vogel]

- interpretable calorimeter phase space x
- learned calibration function

$$\mathcal{R}_{\text{NN}}(x) \pm \Delta\mathcal{R}_{\text{NN}}(x) \approx \frac{E^{\text{obs}}(x)}{E^{\text{dep}}(x)}$$

- systematics: noise in data
  network expressivity
  data representation ...

ATLAS calibration

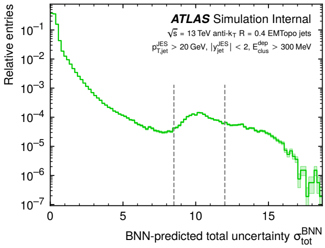
Energy calibration with uncertainties [ATLAS + TP, Vogel]

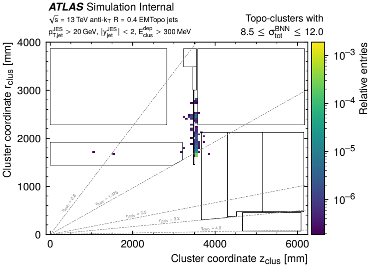
- interpretable calorimeter phase space x
- learned calibration function

$$\mathcal{R}_{\text{NN}}(x) \pm \Delta\mathcal{R}_{\text{NN}}(x) \approx \frac{E^{\text{obs}}(x)}{E^{\text{dep}}(x)}$$

- systematics: noise in data
  network expressivity
  data representation ...

$\to$ Understand (simulated) detector





Controlling generative ML

Generative network for LHC

- Variational Autoencoder [too old] $\longrightarrow$ GAN [2019]
- normalizing flow [2020] $\longrightarrow$ diffusion [2023]
- JetGPT [2023] $\longrightarrow$ vision transformer [2024]

Controlling generative ML

Generative network for LHC

- Variational Autoencoder [too old] $\longrightarrow$ GAN [2019]
- normalizing flow [2020] $\longrightarrow$ diffusion [2023]
- JetGPT [2023] $\longrightarrow$ vision transformer [2024]

Classifier benchmarking [Das, Favaro, Heimel, TP, Shih]

- generation: unsupervised density
- classify training vs generated events $D(x)$
  learned density ratio [Neyman-Pearson]

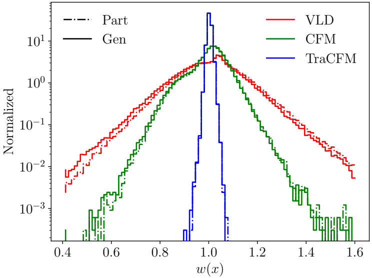
$$w(x_i) = \frac{D(x_i)}{1 - D(x_i)} = \frac{p_\text{data}(x_i)}{p_\text{model}(x_i)}$$

$\rightarrow$ Local quality test

Unfolding generators [Berkeley-Irvine-Heidelberg]

- width of weight distribution
- tails indicating failure mode

$\rightarrow$ Systematic generator benchmarking

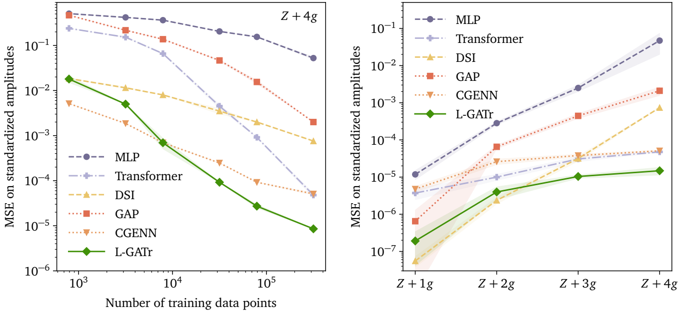


Representation learning

Encode known symmetries [Brehmer, Breso, de Haan, TP, Qu, Spinner, Thaler]

- covariance $[S, f_\theta](x) = 0$
- permutation and Lorentz symmetries
- symmetry breaking learned [Maitre, Ngairangbam, Spannowsky]
- standard amplitude regression

$\rightarrow$ Data efficiency and scaling



Representation learning

Encode known symmetries [Brehmer, Breso, de Haan, TP, Qu, Spinner, Thaler]

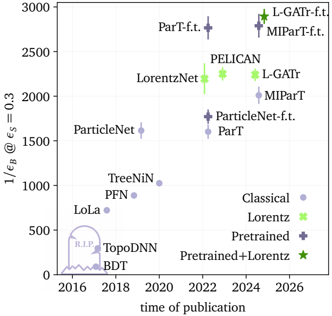
- covariance $[S, f_\theta](x) = 0$
- permutation and Lorentz symmetries
- symmetry breaking learned [Maitre, Ngairangbam, Spannowsky]
- standard amplitude regression

$\rightarrow$ Data efficiency and scaling

General representation

1. top tagging [since 2019]

- graphs vs transformers [Huilin Qu]
- 25$\times$ better rejection from L-GATR



Representation learning

Encode known symmetries [Brehmer, Breso, de Haan, TP, Qu, Spinner, Thaler]

- covariance $[S, f_\theta](x) = 0$
- permutation and Lorentz symmetries
- symmetry breaking learned [Maitre, Ngairangbam, Spannowsky]
- standard amplitude regression

$\to$ Data efficiency and scaling

General representation

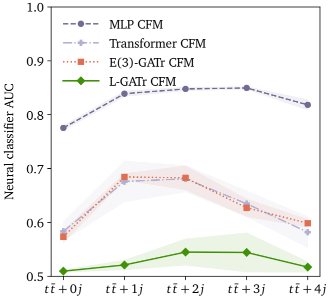
1. top tagging [since 2019]
   - graphs vs transformers [Huilin Qu]
   - 25× better rejection from L-GATR
2. event generation

   CFM with L-GATR

   improved data efficiency

   improved accuracy

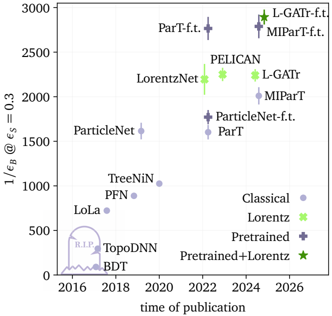
$\to$ LHC data representation



From pre-training to foundation models...

Pre-trained LHC taggers

- pre-train on 100M jets, including tops
- fine-tune on 1.2M jets, QCD & tops
- Benefit from more, very similar data



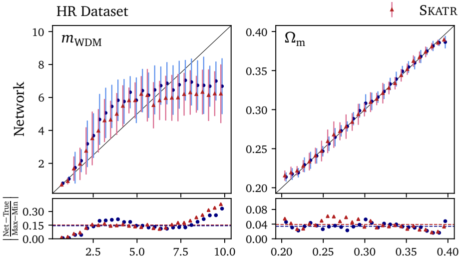
From pre-training to foundation models...

Pre-trained LHC taggers

- pre-train on 100M jets, including tops
- fine-tune on 1.2M jets, QCD & tops
- Benefit from more, very similar data

Pre-trained SKATR inference [Ore, Heneka, TP]

- ViT summary net [like LHC detectors]
- self-supervised training [JEPA]
- improved data efficiency [Duh!]
- Low-resolution pre-training



From pre-training to foundation models...

Pre-trained LHC taggers

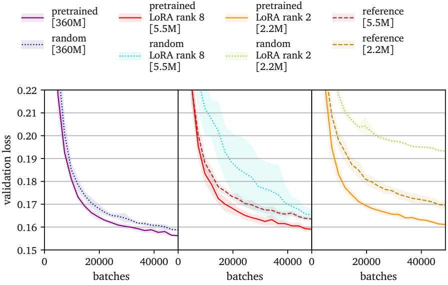
- pre-train on 100M jets, including tops
- fine-tune on 1.2M jets, QCD & tops
- Benefit from more, very similar data

Pre-trained SKATR inference [Ore, Heneka, TP]

- ViT summary net [like LHC detectors]
- self-supervised training [JEPA]
- improved data efficiency [Duh!]
- Low-resolution pre-training

L3M for SKA [Heneka, Nieser, Ore, TP, Schiller]

- very non-physics pre-training
- affine connector networks
- LLM fine-tuned [LoRa]
- foundation model?
  [Berkeley-Hamburg-Geneva-...]
- Whatever...



AI for fundamental physics

Develop AI for the best science

1. just another numerical tool for a numerical field
2. completely transformative new language

- paid for by data science and medical research
- fundamental physics still underperforming
- representation learning the key
- domain knowledge vs pre-training vs foundation models

→ Complexity your friend

→ Training a golden generation

Modern Machine Learning for LHC Physicists

Tilman Plehn[a*], Anja Butter[a,b], Barry Dillon[a],
Theo Heimel[a], Claudius Krause[c], and Ramon Winterhalder[d]

[a] Institut für Theoretische Physik, Universität Heidelberg, Germany
[b] LPNHE, Sorbonne Université, Université Paris Cité, CNRS/IN2P3, Paris, France
[c] HEPHY, Austrian Academy of Sciences. Vienna, Austria
[d] CP3, Université catholique de Louvain, Louvain-la-Neuve, Belgium

March 19, 2024

Abstract

Modern machine learning is transforming particle physics fast, bullying its way into our numerical tool box. For young researchers it is crucial to stay on top of this development, which means applying cutting-edge methods and tools to the full range of LHC physics problems. These lecture notes lead students with basic knowledge of particle physics and significant enthusiasm for machine learning to relevant applications. They start with an LHC-specific motivation and a non-standard introduction to neural networks and then cover classification, unsupervised classification, generative networks, and inverse problems. Two themes defining much of the discussion are well-defined loss functions and uncertainty-aware networks. As part of the applications, the notes include some aspects of theoretical LHC physics. All examples are chosen from particle physics publications of the last few years.[1]