Modern ML for LHC Theory

Tilman Plehn

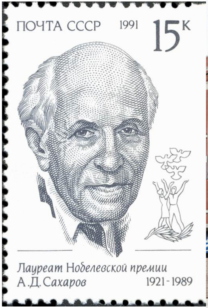
Universität Heidelberg

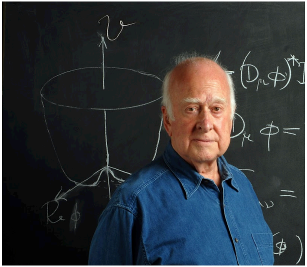
SMASH, Vipava, October 2024

Modern LHC physics

Classic motivation

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





Modern LHC physics

Classic motivation

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

LHC physics

- fundamental questions
- huge data set
- first-principle, precision simulations
- complete uncertainty control

Successful past

- measurements of total rates
- analyses inspired by simulation
- model-driven Higgs discovery

Modern LHC physics

Classic motivation

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

LHC physics

- fundamental questions
- huge data set
- first-principle, precision simulations
- complete uncertainty control

Successful past

- measurements of total rates
- analyses inspired by simulation
- model-driven Higgs discovery

First-principle, precision simulations

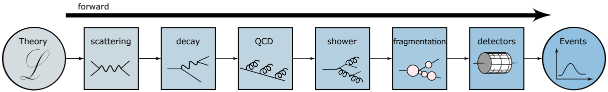
- start with Lagrangian
- calculate scattering using QFT
- simulate collisions
- simulate detectors

$\rightarrow$ LHC collisions in virtual worlds

BSM searches

- compare simulations and data
- understand LHC data systematically
- infer underlying theory [SM or BSM]
- publish useable results

$\rightarrow$ Lots of data science...

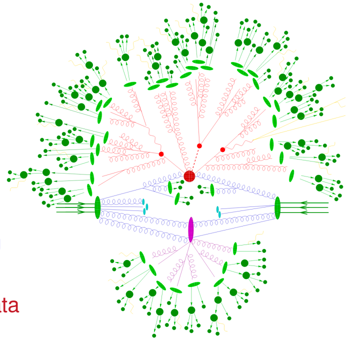


LHC Theory

Turning data into knowledge

- QFT
  start with Lagrangian
  generate Feynman diagrams
- compute hard scattering
  compute decays
  compute jet radiation
- partons inside protons
  hadron-level QCD

$\to$ First-principle simulations, not modeling



HL-LHC: optimal inference with 10$\times$more data

- statistical improvement $\sqrt{10} > 3$
- rate over phase space to $< 0.1\%$
- SBI starts with Simulation $\leftrightarrow$ theory
- speed the key to precision

$\to$ MadNIS & Co

Regression with uncertainties

Calibration function for ATLAS calorimeter

- energy measurement for cluster/jet j

$$\langle E \rangle = \int dE \; E \; p(E)$$

- weighted by reproducing training data $p(\theta|T)$

$$p(E) = \int d\theta \; p(E|\theta) \; p(\theta|T)$$

$\to$ θ-distributions defining Bayesian NN

Regression with uncertainties

Calibration function for ATLAS calorimeter

- energy measurement for cluster/jet j

$$\langle E \rangle = \int dE \; E \; p(E)$$

- weighted by reproducing training data $p(\theta|T)$

$$p(E) = \int d\theta \; p(E|\theta) \; p(\theta|T)$$

$\rightarrow$ θ-distributions defining Bayesian NN

Variational approximation

- definition of training [think $q(\theta)$ as Gaussian with mean and width]

$$p(E) = \int d\theta \; p(E|\theta) \; p(\theta|T) \approx \int d\theta \; p(E|\theta) \; q(\theta)$$

- similarity through minimal KL-divergence [Bayes' theorem to remove unknown posterior]

$$\begin{aligned} D_{\text{KL}}[q(\theta), p(\theta|T)] &= \int d\theta \; q(\theta) \; \log \frac{q(\theta)}{p(\theta|T)} \\ &= \int d\theta \; q(\theta) \; \log \frac{q(\theta)p(T)}{p(T|\theta)p(\theta)} \\ &= D_{\text{KL}}[q(\theta), p(\theta)] - \int d\theta \; q(\theta) \; \log p(T|\theta) + \log p(T) \int d\theta \; q(\theta) \end{aligned}$$

Regression with uncertainties

Calibration function for ATLAS calorimeter

- energy measurement for cluster/jet j

$$\langle E \rangle = \int dE \; E \; p(E)$$

- weighted by reproducing training data $p(\theta|T)$

$$p(E) = \int d\theta \; p(E|\theta) \; p(\theta|T)$$

$\rightarrow$ θ-distributions defining Bayesian NN

Variational approximation

- definition of training [think $q(\theta)$ as Gaussian with mean and width]

$$p(E) = \int d\theta \; p(E|\theta) \; p(\theta|T) \approx \int d\theta \; p(E|\theta) \; q(\theta)$$

- similarity through minimal KL-divergence [Bayes' theorem to remove unknown posterior]

$$\begin{aligned} D_{\text{KL}}[q(\theta), p(\theta|T)] &= \int d\theta \; q(\theta) \; \log \frac{q(\theta)}{p(\theta|T)} \\ &= \int d\theta \; q(\theta) \; \log \frac{q(\theta)p(T)}{p(T|\theta)p(\theta)} \\ &\approx D_{\text{KL}}[q(\theta), p(\theta)] - \int d\theta \; q(\theta) \; \log p(T|\theta) \equiv \mathcal{L} \end{aligned}$$

$\rightarrow$ Two-term loss: likelihood + prior

Relation to deterministic networks

Regularization

- BNN loss

$$\begin{aligned}\mathcal{L} &= -\int d\theta\; q(\theta)\;\log p(T|\theta) + D_\text{KL}[q(\theta), p(\theta)] \\ &= -\int d\theta\; q(\theta)\;\log p(T|\theta) + \frac{\sigma_q^2 - \sigma_p^2 + (\mu_q - \mu_p)^2}{2\sigma_p^2} + \dots\end{aligned}$$

- deterministic network

$$q(\theta) = \delta(\theta - \theta_0) \quad \Rightarrow \quad \mathcal{L} \approx -\log p(T|\theta_0) + \frac{(\theta_0 - \mu_p)^2}{2\sigma_p^2}$$

$\to$ Likelihood with L2-regularization

Relation to deterministic networks

Regularization

- BNN loss

$$\mathcal{L} = -\int d\theta\; q(\theta)\; \log p(T|\theta) + D_\text{KL}[q(\theta), p(\theta)]$$

$$= -\int d\theta\; q(\theta)\; \log p(T|\theta) + \frac{\sigma_q^2 - \sigma_p^2 + (\mu_q - \mu_p)^2}{2\sigma_p^2} + \dots$$

- deterministic network

$$q(\theta) = \delta(\theta - \theta_0) \quad \Rightarrow \quad \mathcal{L} \approx -\log p(T|\theta_0) + \frac{(\theta_0 - \mu_p)^2}{2\sigma_p^2}$$

$\rightarrow$ Likelihood with L2-regularization

Dropout

- Bernoulli weights

$$q(\theta) \rightarrow q(x) = \rho^x (1-\rho)^{1-x}\Big|_{x=0,1} \qquad \text{with} \quad \theta = x\theta_0$$

$\rightarrow$ Regularized likelihood with dropout

Statistics vs systematics

Network evaluation

- expectation value using trained network $q(\theta)$

$$\begin{aligned}\langle E \rangle &= \int dE d\theta \; E \; p(E|\theta) \; q(\theta) \\ &\equiv \int d\theta \; q(\theta) \overline{E}(\theta) \qquad \text{with} \qquad \overline{E}(\theta) = \int dE \; E \; p(E|\theta)\end{aligned}$$

- corresponding variance

$$\begin{aligned}\sigma_\text{tot}^2 &= \int dE d\theta \; \left(E - \langle E \rangle\right)^2 \; p(E|\theta) \; q(\theta) \\ &= \int d\theta \; q(\theta) \left[\overline{E^2}(\theta) - 2\langle E \rangle \overline{E}(\theta) + \langle E \rangle^2 \right] \\ &= \int d\theta \; q(\theta) \left[\overline{E^2}(\theta) - \overline{E}(\theta)^2 + \left(\overline{E}(\theta) - \langle E \rangle \right)^2 \right] \equiv \sigma_\text{syst}^2 + \sigma_\text{stat}^2\end{aligned}$$

Two uncertainties

- statistical — vanishing for perfect training: $q(\theta) \to \delta(\theta - \theta_0)$

$$\sigma_\text{stat}^2 = \int d\theta \; q(\theta) \left[\overline{E}(\theta) - \langle E \rangle \right]^2 = \left[\overline{E}(\theta_0) - \langle E \rangle \right]^2$$

- systematic — vanishing for perfect data: $p(E|\theta) \to \delta(E - E_0)$

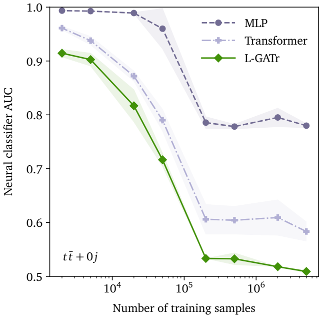
$$\sigma_\text{syst}^2 = \int d\theta \; q(\theta) \left[\overline{E^2}(\theta) - \overline{E}(\theta)^2 \right]$$

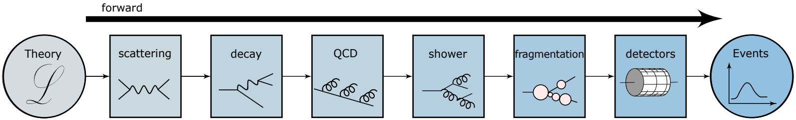
Generative AI

Forward simulations

- learn phase space density
  sample Gaussian $\to$ phase space
- Variational Autoencoder
  $\to$ low-dimensional physics
- Generative Adversarial Network
  $\to$ generator trained by classifier
- Normalizing Flow/Diffusion
  $\to$ (bijective) mapping [INN]
- JetGPT, ViT
  $\to$ non-local structures
- Equivariant L-GATr
  $\to$ guarantee Lorentz symmetry

$\to$ Combinations: equivariant transformer CFM...





Generative Uncertainties

Unsupervised Bayesian networks

- data: event sample [points in 2D space]

  learn phase space density
  standard distribution in latent space [Gaussian]
  sample from latent space

- Bayesian version

  allow weight distributions
  learn uncertainty map

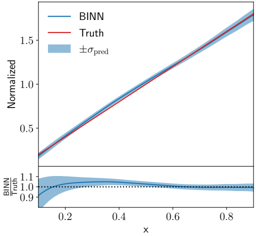
- 2D wedge ramp

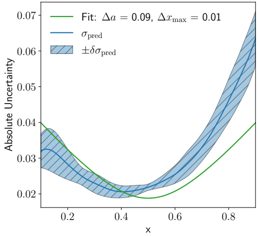
$$p(x) = ax + b = ax + \frac{1 - \frac{a}{2}(x_{\text{max}}^2 - x_{\text{min}}^2)}{x_{\text{max}} - x_{\text{min}}}$$

$$(\Delta p)^2 = \left(x - \frac{1}{2}\right)^2 (\Delta a)^2 + \left(1 + \frac{a}{2}\right)^2 (\Delta x_{\text{max}})^2 + \left(1 - \frac{a}{2}\right)^2 (\Delta x_{\text{min}})^2$$

explaining minimum in $\sigma(x)$

$\to$ INNs, diffusion just (non-parametric) fits



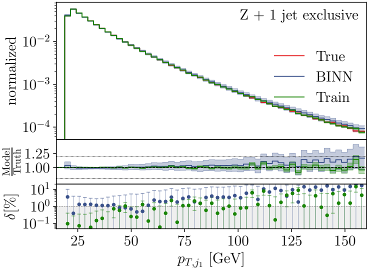


Events with uncertainties

Bayesian network generator

- network with weight distributions [Gal (2016)]
  sample weights [defining error bar]
  frequentist: efficient ensembling

$\Rightarrow$ Training-related error bars



Events with uncertainties

Bayesian network generator

- network with weight distributions [Gal (2016)]
  sample weights [defining error bar]
  frequentist: efficient ensembling

$\Rightarrow$ Training-related error bars

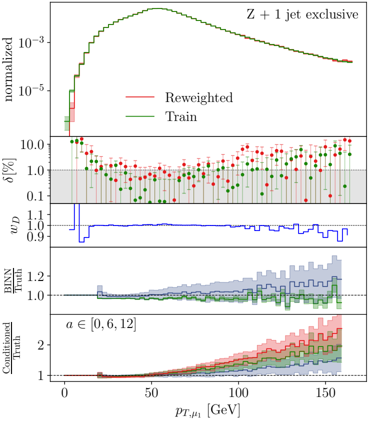
Theory uncertainties

- BNN regression/classification:
  systematics from data augmentation
- systematic uncertainties in tails

$$w = 1 + a\left(\frac{p_{T,j_1} - 15\ \text{GeV}}{100\ \text{GeV}}\right)^2$$

- augment training data [$a = 0 \dots 30$]
- train conditionally on a
  error bar from sampling a

$\Rightarrow$ Systematic/theory error bars



Transforming LHC physics

Number of searches

- SBI: signal and background simulations
- CPU-limitation for many signals?

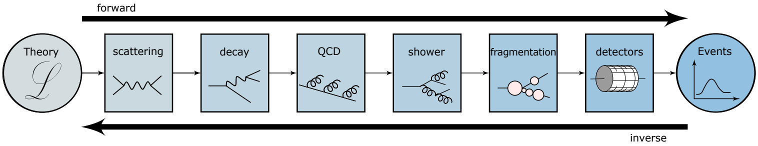
Optimal analyses

- theory limiting many analyses, but continuous progress
- allow for analyses to be updated?

Public LHC data

- common lore:
  LHC data too complicated for amateurs
- in truth:
  hard scattering and decay simulations public
  BSM physics not in hadronization and detector

$\rightarrow$ Unfold to suitable level [EFT?]



ML-Unfolding

Basic structure

- four phase space distributions

$$\begin{tikzcd} p_\text{sim}(x_\text{part}) \arrow[rr, leftrightarrow, "\text{unfolding inference}"] \arrow[d, "p(x_\text{reco}|x_\text{part})"'] & & p_\text{unfold}(x_\text{part}) \\ p_\text{sim}(x_\text{reco}) \arrow[rr, leftrightarrow, "\text{forward inference}"] & & p_\text{data}(x_\text{reco}) \arrow[u, "p(x_\text{part}|x_\text{reco})"'] \end{tikzcd}$$

- forward and inverse generation symmetric [stochastic]

$\to$ ML for unbinned and high-dimensional unfolding?

ML-Unfolding

Basic structure

- four phase space distributions

$$\begin{tikzcd}
p_{\text{sim}}(x_{\text{part}}) \arrow[r, leftrightarrow, "\text{unfolding inference}"] \arrow[d, "p(x_{\text{reco}} | x_{\text{part}})"'] & p_{\text{unfold}}(x_{\text{part}}) \\
p_{\text{sim}}(x_{\text{reco}}) \arrow[r, leftrightarrow, "\text{forward inference}"] & p_{\text{data}}(x_{\text{reco}}) \arrow[u, "p(x_{\text{part}} | x_{\text{reco}})"']
\end{tikzcd}$$

- forward and inverse generation symmetric [stochastic]

$\to$ ML for unbinned and high-dimensional unfolding?

OmniFold [Andreassen, Komiske, Metodiev, Nachman, Thaler]

- learn $p_{\text{sim}}(x_{\text{reco}}) \leftrightarrow p_{\text{data}}(x_{\text{reco}})$ [Neyman-Pearson lemma, CWoLa]
- reweight $p_{\text{sim}}(x_{\text{part}}) \to p_{\text{unfold}}(x_{\text{part}})$

$$\begin{tikzcd}
p_{\text{sim}}(x_{\text{part}}) \arrow[r, "\text{classifier weights}"] \arrow[d, leftrightarrow, "\text{pull/push weights}"'] & p_{\text{unfold}}(x_{\text{part}}) \\
p_{\text{sim}}(x_{\text{reco}}) \arrow[r, leftrightarrow, "\text{classifier weights}"] & p_{\text{data}}(x_{\text{reco}})
\end{tikzcd}$$

$\to$ Driven by established ML-classification

Unfolding by generation

Targeting conditional probability [Butter, TP, Winterhalder,...]

- just like forward ML-generation
- learn inverse conditional probability from $(x_\text{part}, x_\text{reco})$

$$\begin{tikzcd}
p_\text{sim}(x_\text{part}) \arrow[d, leftrightarrow, "\text{paired data}"'] & & p_\text{unfold}(x_\text{part}) \\
p_\text{sim}(x_\text{reco}) \arrow[rr, leftrightarrow, "\text{correspondence}"] & & p_\text{data}(x_\text{reco}) \arrow[u, "p_\text{model}(x_\text{part}|x_\text{reco})"']
\end{tikzcd}$$

Improvements crucial

1. likelihood loss to generate posterior $\rightarrow$ cINN
2. make networks more precise $\rightarrow$ TraCFM
3. remove training prior [Backes, Butter, Dunford, Malaescu]

$\rightarrow$ Driven by generative networks

Unfolding top decays

Tough challenge [Favaro, Kogler, Paasch, Palacios Schweitzer, TP, Schwarz]

- first measure m_t in unfolded data
  then unfold full kinematics
- model dependence: simulation m_s vs data m_d

$$\begin{tikzcd}
p_\text{sim}(x_\text{part} | m_s) \arrow[d, "p(x_\text{reco} | x_\text{part})"'] & p_\text{unfold}(x_\text{part} | m_s, m_d) \\
p_\text{sim}(x_\text{reco} | m_s) \arrow[r, leftrightarrow, "\text{correspondence}"] & p_\text{data}(x_\text{reco} | m_d) \arrow[u, "p_\text{model}(x_\text{part} | x_\text{reco}, m_s)"']
\end{tikzcd}$$

Unfolding top decays

Tough challenge [Favaro, Kogler, Paasch, Palacios Schweitzer, TP, Schwarz]

- first measure m_t in unfolded data
  then unfold full kinematics
- model dependence: simulation m_s vs data m_d

$$\begin{tikzcd}
p_\text{sim}(x_\text{part}|m_s) \arrow[d, "p(x_\text{reco}|x_\text{part})"'] & & p_\text{unfold}(x_\text{part}|m_s, m_d) \\
p_\text{sim}(x_\text{reco}|m_s) \arrow[rr, leftrightarrow, "\text{correspondence}"] & & p_\text{data}(x_\text{reco}|m_d) \arrow[u, "p_\text{model}(x_\text{part}|x_\text{reco}, m_s)"']
\end{tikzcd}$$

1. weaken bias by training on m_s-range
2. strengthen data by including batch-wise $m_d \sim M_{jjj} \in x_\text{reco}$

Unfolding top decays

Tough challenge [Favaro, Kogler, Paasch, Palacios Schweitzer, TP, Schwarz]

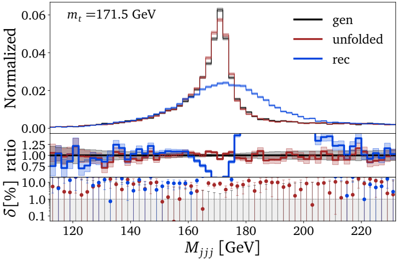
- first measure m_t in unfolded data
  then unfold full kinematics
- model dependence: simulation m_s vs data m_d

$$
\begin{tikzcd}
p_{\text{sim}}(x_{\text{part}} \,|\, m_s) \arrow[d, "p(x_{\text{reco}} \,|\, x_{\text{part}})"'] & & p_{\text{unfold}}(x_{\text{part}} \,|\, m_s, m_d) \\
p_{\text{sim}}(x_{\text{reco}} \,|\, m_s) \arrow[rr, leftrightarrow, "\text{correspondence}"] & & p_{\text{data}}(x_{\text{reco}} \,|\, m_d) \arrow[u, "p_{\text{model}}(x_{\text{part}} \,|\, x_{\text{reco}}, m_s)"']
\end{tikzcd}
$$

1. weaken bias by training on m_s-range
2. strengthen data by including batch-wise $m_d \sim M_{jjj} \in x_{\text{reco}}$

Preliminary unfolding results [TraCFM]

- 4D for calibrated mass measurement



Unfolding top decays

Tough challenge [Favaro, Kogler, Paasch, Palacios Schweitzer, TP, Schwarz]

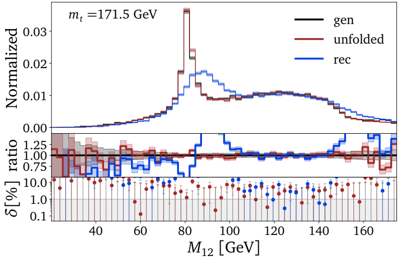
- first measure m_t in unfolded data
  then unfold full kinematics
- model dependence: simulation m_s vs data m_d

$$\begin{tikzcd}
p_\text{sim}(x_\text{part}|m_s) \arrow[d, "p(x_\text{reco}|x_\text{part})"'] & & p_\text{unfold}(x_\text{part}|m_s, m_d) \\
p_\text{sim}(x_\text{reco}|m_s) \arrow[rr, leftrightarrow, "\text{correspondence}"] & & p_\text{data}(x_\text{reco}|m_d) \arrow[u, "p_\text{model}(x_\text{part}|x_\text{reco}, m_s)"']
\end{tikzcd}$$

1. weaken bias by training on m_s-range
2. strengthen data by including batch-wise $m_d \sim M_{jjj} \in x_\text{reco}$

Preliminary unfolding results [TraCFM]

- 4D for calibrated mass measurement



Unfolding top decays

Tough challenge [Favaro, Kogler, Paasch, Palacios Schweitzer, TP, Schwarz]

- first measure m_t in unfolded data
  then unfold full kinematics
- model dependence: simulation m_s vs data m_d

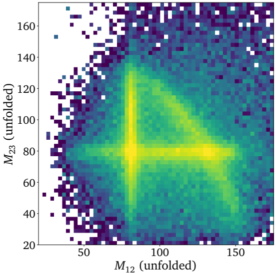
$$
\begin{tikzcd}
p_\text{sim}(x_\text{part} | m_s) \arrow[d, "p(x_\text{reco} | x_\text{part})"'] & & p_\text{unfold}(x_\text{part} | m_s, m_d) \\
p_\text{sim}(x_\text{reco} | m_s) \arrow[rr, leftrightarrow, "\text{correspondence}"] & & p_\text{data}(x_\text{reco} | m_d) \arrow[u, "p_\text{model}(x_\text{part} | x_\text{reco}, m_s)"']
\end{tikzcd}
$$

1. weaken bias by training on m_s-range
2. strengthen data by including batch-wise $m_d \sim M_{jjj} \in x_\text{reco}$

Preliminary unfolding results [TraCFM]

- 4D for calibrated mass measurement
- 12D published data

$\to$ CMS data next



ML for LHC Theory

Developing ML for the best science

- just another numerical tool for a numerical field
- transformative new common language
- driven by money from data science and medical research
- be 10000 Einsteins,

  ...improving established tools
  ...developing new tools for established tasks
  ...transforming through new ideas

$\rightarrow$ You are the 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]