Modern ML in LHC Physics

Tilman Plehn

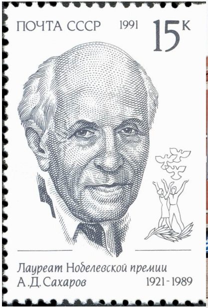
Universität Heidelberg

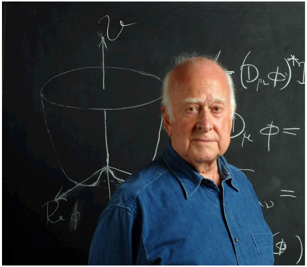
Pisa, September 2024

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

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

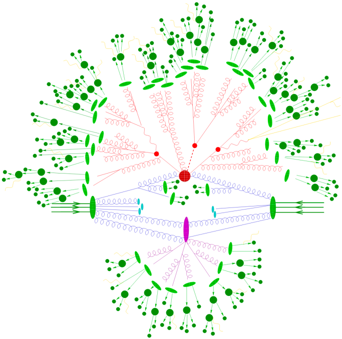
Successful past

- 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



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

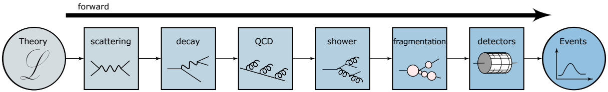
- start with Lagrangian
- calculate scattering using QFT
- simulate collisions
- simulate detectors

$\to$ LHC collisions in virtual worlds

BSM searches

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

$\to$ Lots of data science...



Likelihood loss & uncertainties

Loss to train θ-distributions

- energy measurement for jet j

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

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

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

$\rightarrow$ θ-distributions means Bayesian NN

Likelihood loss & uncertainties

Loss to train θ-distributions

- energy measurement for jet j

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

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

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

$\to$ θ-distributions means 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]

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

Likelihood loss & uncertainties

Loss to train θ-distributions

- energy measurement for jet j

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

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

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

$\to$ θ-distributions means 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]

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

Likelihood loss & uncertainties

Loss to train θ-distributions

- energy measurement for jet j

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

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

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

$\to$ θ-distributions means 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]

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

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

Relation to deterministic networks

Regularization

- BNN loss

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

Relation to deterministic networks

Regularization

- Gaussian prior

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

- deterministic network

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

$\to$ Likelihood with L2-regularization

Relation to deterministic networks

Regularization

- Gaussian prior

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

- 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

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

- likelihood loss

$$\mathcal{L} = -\sum_{x=0,1} \left[\rho^x (1-\rho)^{1-x} \right] \; \log p(T|x\theta_0) \approx -\rho \; \log p(T|\theta_0)$$

- likelihood Gaussian or whatever else...

$\rightarrow$ Regularized likelihood with dropout

Statistics vs systematics

Network evaluation

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

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

- corresponding variance

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

Two uncertainties

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

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

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

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

Generative networks

Unsupervised Bayesian networks

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

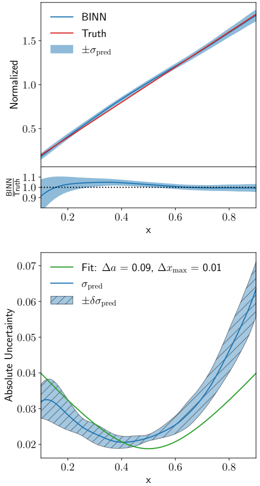
  learn phase space density
  normalizing flow mapping to latent space
  standard distribution in latent space [Gaussian]
  mapping bijective
  sample from latent space
- Bayesian version

  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 $\sigma(x)$



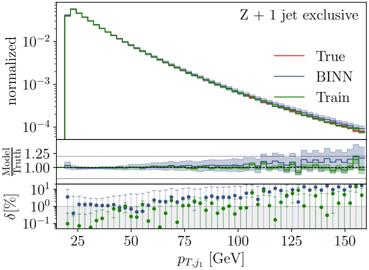
→ INNs, diffusion just (non-parametric) fits

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

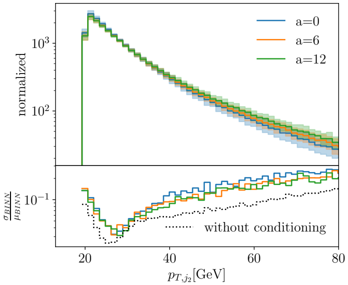
Theory uncertainties

- 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

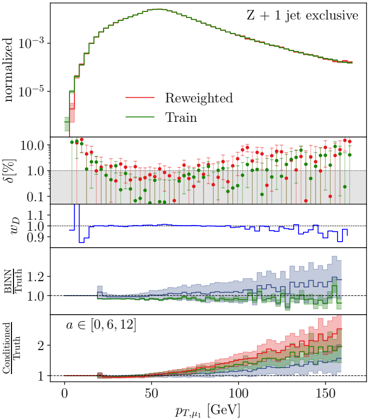


Events with uncertainties

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 [$a = 0 \ldots 30$]
- train conditionally on a
  error bar from sampling a

$\Rightarrow$ Systematic/theory error bars

Transforming LHC physics

Number of searches

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

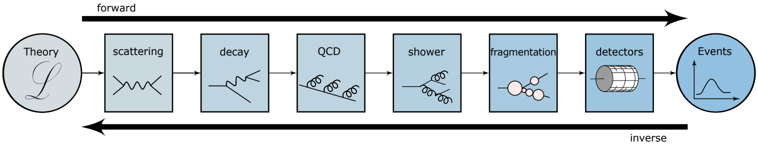
Optimal analyses

- 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

→ Unfold to suitable level [EFT?]



ML-Unfolding

Basic structure

- four phase space distributions

$$\begin{array}{ccc} p_\text{sim}(x_\text{part}) & \xleftrightarrow{\text{unfolding inference}} & p_\text{unfold}(x_\text{part}) \\ \downarrow p(x_\text{reco}|x_\text{part}) & & \uparrow p(x_\text{part}|x_\text{reco}) \\ p_\text{sim}(x_\text{reco}) & \xleftrightarrow{\text{forward inference}} & p_\text{data}(x_\text{reco}) \end{array}$$

- two conditional probabilities

$$p(x_\text{part}|x_\text{reco}) = p(x_\text{reco}|x_\text{part}) \times \frac{p_\text{sim}(x_\text{part})}{p_\text{sim}(x_\text{reco})}$$

- forward and inverse generation symmetric [stochastic]
- learnable from paired events $(x_\text{part}, x_\text{reco})$

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

ML-Unfolding

Basic structure

- four phase space distributions

$p_{\text{sim}}(x_{\text{part}})$ ←unfolding inference→ $p_{\text{unfold}}(x_{\text{part}})$

$p_{\text{sim}}(x_{\text{part}})$ ↓ $p(x_{\text{reco}}|x_{\text{part}})$ ↓ $p_{\text{sim}}(x_{\text{reco}})$

$p_{\text{data}}(x_{\text{reco}})$ ↑ $p(x_{\text{part}}|x_{\text{reco}})$ ↑ $p_{\text{unfold}}(x_{\text{part}})$

$p_{\text{sim}}(x_{\text{reco}})$ ←forward inference→ $p_{\text{data}}(x_{\text{reco}})$

→ 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 $p_{\text{sim}}(x_{\text{part}}) \to p_{\text{unfold}}(x_{\text{part}})$

$p_{\text{sim}}(x_{\text{part}})$ —classifier weights→ $p_{\text{unfold}}(x_{\text{part}})$

$p_{\text{sim}}(x_{\text{part}})$ ↕ pull/push weights ↕ $p_{\text{sim}}(x_{\text{reco}})$

$p_{\text{sim}}(x_{\text{reco}})$ ←classifier weights→ $p_{\text{data}}(x_{\text{reco}})$

- Z+jets in 24D [ATLAS]

→ Driven by (now) established ML-classification

Unfolding by generation

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

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

$p_\text{sim}(x_\text{part})$ ↕ paired data ↕ $p_\text{sim}(x_\text{reco})$

$p_\text{sim}(x_\text{reco})$ ←correspondence→ $p_\text{data}(x_\text{reco})$

$p_\text{data}(x_\text{reco})$ → $p_\text{model}(x_\text{part}|x_\text{reco})$ → $p_\text{unfold}(x_\text{part})$

Improvements crucial

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

$\to$ Driven by generative networks

Unfolding top decays

A 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

$$p_\text{sim}(x_\text{part}|m_s) \qquad\qquad p_\text{unfold}(x_\text{part}|m_s, m_d)$$

$p(x_\text{reco}|x_\text{part})$ ↓ $\qquad\qquad$ ↑ $p_\text{model}(x_\text{part}|x_\text{reco}, m_s)$

$$p_\text{sim}(x_\text{reco}|m_s) \xleftrightarrow{\text{correspondence}} p_\text{data}(x_\text{reco}|m_d)$$

Unfolding top decays

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

- first measure m_t in unfolded data
  then unfold full kinematics
- complete training bias $m_d \to m_s$ [too bad to reweight]

$$p_\text{sim}(x_\text{part} | m_s) \quad \xrightarrow{p(x_\text{reco}|x_\text{part})} \quad p_\text{sim}(x_\text{reco} | m_s)$$

$$p_\text{sim}(x_\text{reco} | m_s) \quad \xleftrightarrow{\text{correspondence}} \quad p_\text{data}(x_\text{reco} | m_d)$$

$$p_\text{data}(x_\text{reco} | m_d) \quad \xrightarrow{p_\text{model}(x_\text{part}|x_\text{reco}, m_s)} \quad p_\text{unfold}(x_\text{part} | m_s, \cancel{m_d})$$

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

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

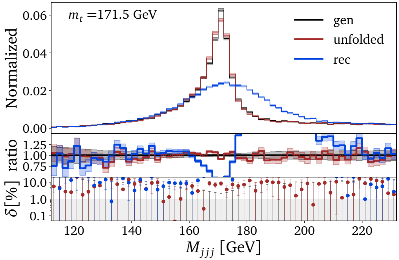
- first measure m_t in unfolded data
  then unfold full kinematics
- complete training bias $m_d \to m_s$ [too bad to reweight]

$$p_\text{sim}(x_\text{part}|m_s) \xrightarrow{p(x_\text{reco}|x_\text{part})} p_\text{sim}(x_\text{reco}|m_s) \xleftrightarrow{\text{correspondence}} p_\text{data}(x_\text{reco}|m_d) \xrightarrow{p_\text{model}(x_\text{part}|x_\text{reco},m_s)} p_\text{unfold}(x_\text{part}|m_s, \not{m_d})$$

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

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

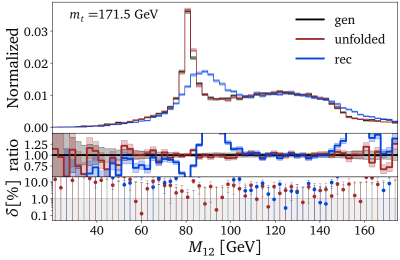
- first measure m_t in unfolded data
  then unfold full kinematics
- complete training bias $m_d \to m_s$ [too bad to reweight]

$$p_{\text{sim}}(x_{\text{part}}|m_s) \xrightarrow{p(x_{\text{reco}}|x_{\text{part}})} p_{\text{sim}}(x_{\text{reco}}|m_s) \xleftrightarrow{\text{correspondence}} p_{\text{data}}(x_{\text{reco}}|m_d) \xrightarrow{p_{\text{model}}(x_{\text{part}}|x_{\text{reco}},m_s)} p_{\text{unfold}}(x_{\text{part}}|m_s, \cancel{m_d})$$

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

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

- first measure m_t in unfolded data
  then unfold full kinematics
- complete training bias $m_d \to m_s$ [too bad to reweight]

$$p_{\text{sim}}(x_{\text{part}} | m_s) \quad \xrightarrow{\;p(x_{\text{reco}}|x_{\text{part}})\;} \quad p_{\text{sim}}(x_{\text{reco}} | m_s)$$

$$p_{\text{sim}}(x_{\text{reco}} | m_s) \quad \xleftrightarrow{\text{correspondence}} \quad p_{\text{data}}(x_{\text{reco}} | m_d)$$

$$p_{\text{data}}(x_{\text{reco}} | m_d) \quad \xrightarrow{\;p_{\text{model}}(x_{\text{part}}|x_{\text{reco}}, m_s)\;} \quad p_{\text{unfold}}(x_{\text{part}} | m_s, \not{m_d})$$

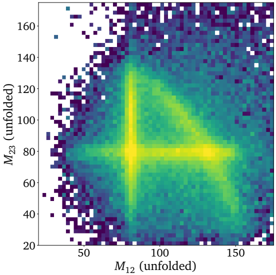
1 weaken bias by training on m_s-range

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
- 1000 Einsteins...

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

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]

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
- 1000 Einsteins...

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

→ 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]