Modern Machine Learning for LHC Theory

Tilman Plehn

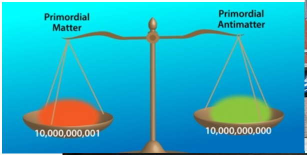
Universität Heidelberg

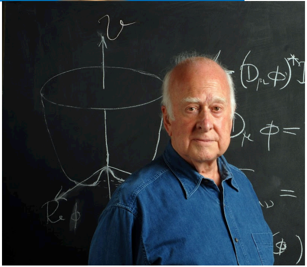
IAIFI, August 2024

Modern LHC physics

Classic motivation

- dark matter?
- baryogenesis?
- origin of Higgs field?





Modern LHC physics

Classic motivation

- dark matter?
- baryogenesis?
- origin of Higgs field?

Unique LHC setting

- first-principle simulations
- huge data set
- uncertainty control

Modern LHC physics

Classic motivation

- dark matter?
- baryogenesis?
- origin of Higgs field?

Unique LHC setting

- first-principle simulations
- huge data set
- uncertainty control

Glorious past

- measurements of event counts
- model-driven Higgs discovery
- lots of exclusions [plus LHCb hadrons]

Modern LHC physics

Classic motivation

- dark matter?
- baryogenesis?
- origin of Higgs field?

Unique LHC setting

- first-principle simulations
- huge data set
- uncertainty control

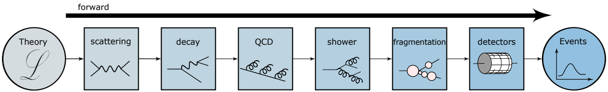
Glorious past

- measurements of event counts
- model-driven Higgs discovery
- lots of exclusions [plus LHCb hadrons]

First-principle simulations

- start with Lagrangian
- then quantum field theory
- finally detectors
- symmetries crucial

$\rightarrow$ LHC collisions in virtual worlds



Modern LHC physics

Classic motivation

- dark matter?
- baryogenesis?
- origin of Higgs field?

Unique LHC setting

- first-principle simulations
- huge data set
- uncertainty control

Glorious past

- measurements of event counts
- model-driven Higgs discovery
- lots of exclusions [plus LHCb hadrons]

First-principle simulations

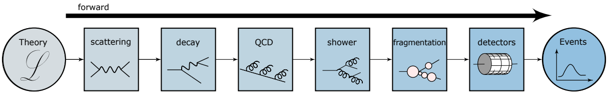
- start with Lagrangian
- then quantum field theory
- finally detectors
- symmetries crucial

$\to$ LHC collisions in virtual worlds

Modern inference

- simulations vs data $\to$ SBI [this morning]
- understand all of LHC data
- discover BSM physics?

$\to$ ML-case obvious



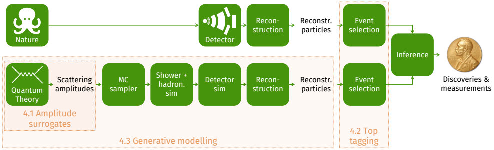
LHC Theory

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

$\rightarrow$ First-principle simulations, not modeling

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 20$\times$more data

- statistical improvement $\sqrt{20} = 4.5$
- rate over phase space to $< 0.1\%$
- SBI starts with simulation $\to$ theory
- speed the key [also to precision]
- module-wise improvements

$\to$ MadNIS & Co

ML-Theory

Tilman Plehn

LHC theory

Generative AI

Unfolding

Top decays

Bayesian NNs

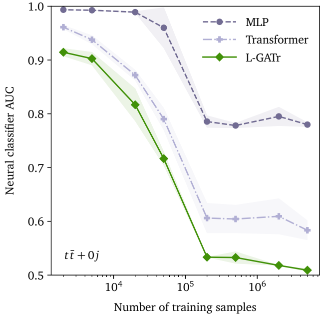
Calibration

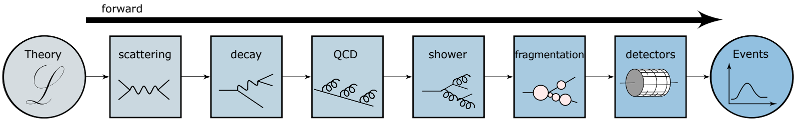
Generative AI

Forward simulations [Favaro, Heimel, Hütsch, Palacios Schweitzer, Spinner, Villadamigo, Winterhalder...]

- 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 [with Jesse and QualComm AI]
  $\to$ guarantee Lorentz symmetry

$\to$ Combine models: Transfermer, TraCFM,...





Generative AI

Forward simulations [Favaro, Heimel, Hütsch, Palacios Schweitzer, Spinner, Villadamigo, Winterhalder...]

- learn phase space density [with error bar]
  sample Gaussian $\to$ phase space
- Variational Autoencoder
- Generative Adversarial Network
- Normalizing Flow/INN/Diffusion
- JetGPT, ViT
- Equivariant L-GATr

$\to$ Combine models: Transfermer, TraCFM,...

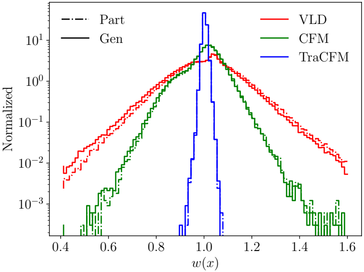
Quality control [Das, Favaro, Heimel, Krause, TP, Shih]

- classifier easier to train
  training vs generated

$$w(x_i) = \frac{D(x_i)}{1 - D(x_i)} = \frac{p_{\text{train}}(x_i)}{p_{\text{gen}}(x_i)}$$

- performance from width of distribution
- $w(x_i) \gg 1$ missing feature
  $w(x_i) \ll 1$ missing cut

$\to$ Systematic performance test



Transforming LHC physics

Number of searches

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

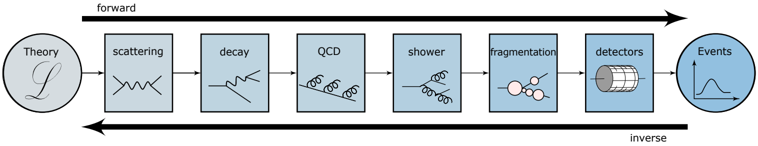
Optimal analyses

- 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

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

- 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 $(x_\text{part}, x_\text{reco})$

→ ML for unbinned and high-dimensional unfolding?

ML-Unfolding

Basic structure

- four phase space distributions

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

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

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

- Z+jets in 24D [ATLAS]

$\to$ 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 $(x_{\text{part}}, x_{\text{reco}})$

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

Improvements crucial

1. likelihood loss to generate posterior $\to$ 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

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

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]

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

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

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

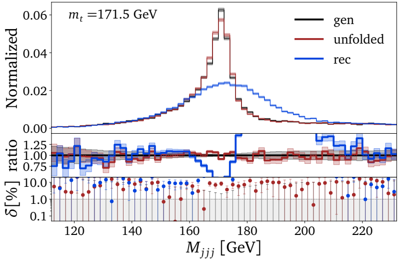
- first measure m_t in unfolded data
  then unfold full kinematics
- complete training bias $m_d \to m_s$ [too bad to reweight]

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

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

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

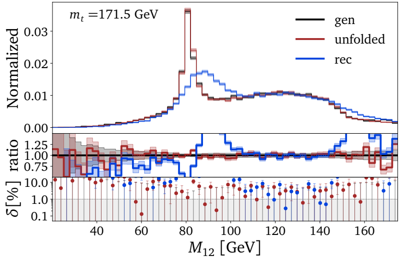
- first measure m_t in unfolded data
  then unfold full kinematics
- complete training bias $m_d \to m_s$ [too bad to reweight]

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

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]

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

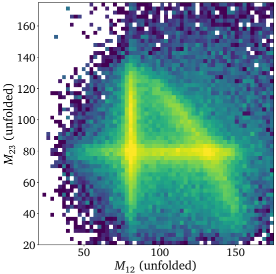
1 weaken bias by training on m_s-range

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

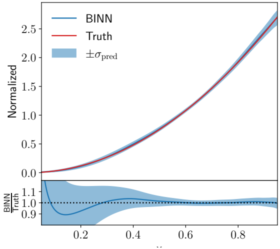


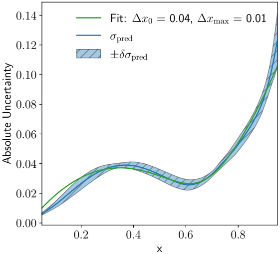
Generative networks with uncertainties

Bayesian generative networks [Bellagente, Haussmann, Luchmann, TP]

- network weight distributions for density
- sampling phase space
  events with error bars on weights
- learned density & uncertainty
  reflecting network learning?

→ INNs like fitted functions





Generative networks with uncertainties

Bayesian generative networks [Bellagente, Haussmann, Luchmann, TP]

- network weight distributions for density
- sampling phase space
  events with error bars on weights
- learned density & uncertainty
  reflecting network learning?

→ INNs like fitted functions

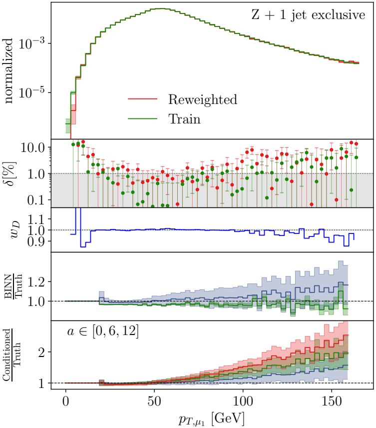
LHC events with uncertainties [Heimel, Vent...]

- classifier weight for improvement [DCTR]
- statistical training limitation
  encoded in Bayesian generator
- systematic training limitation

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

sampled through conditional generator

→ Comprehensive uncertainty control



Understandable calorimeter calibration

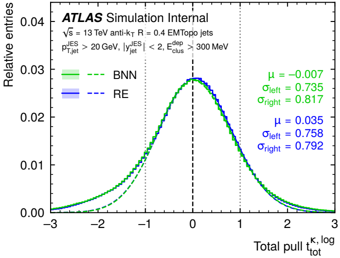
Calibration with uncertainties [Vogel, Loch, TP,...]

- interpretable topo-cluster phase space x
- learned calibration

$$\mathcal{R}^{\text{BNN}}(x) = \mathcal{R}(x) = \frac{E^{\text{EM}}(x)}{E^{\text{dep}}(x)}$$

- learned uncertainties $\Delta\mathcal{R}(x)$ [Nathan's talk]
  Bayesian neural networks
  repulsive ensembles

→ check error vs data spread using pull



Understandable calorimeter calibration

Calibration with uncertainties [Vogel, Loch, TP,...]

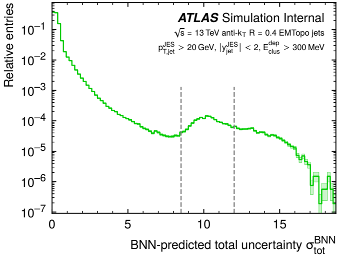
- interpretable topo-cluster phase space x
- learned calibration

$$\mathcal{R}^{\text{BNN}}(x) = \mathcal{R}(x) = \frac{E^{\text{EM}}(x)}{E^{\text{dep}}(x)}$$

- learned uncertainties $\Delta\mathcal{R}(x)$ [Nathan's talk]
  Bayesian neural networks
  repulsive ensembles

$\to$ check error vs data spread using pull

$\to$ Understand data using uncertainties



Understandable calorimeter calibration

Calibration with uncertainties [Vogel, Loch, TP,...]

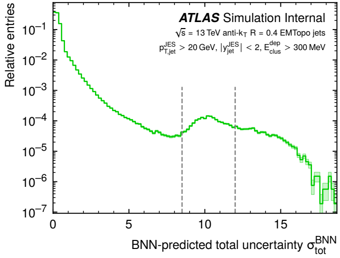
- interpretable topo-cluster phase space x
- learned calibration

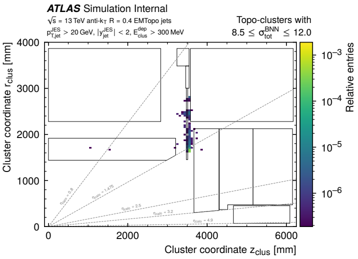
$$\mathcal{R}^{\text{BNN}}(x) = \mathcal{R}(x) = \frac{E^{\text{EM}}(x)}{E^{\text{dep}}(x)}$$

- learned uncertainties $\Delta\mathcal{R}(x)$ [Nathan's talk]
  Bayesian neural networks
  repulsive ensembles

$\to$ check error vs data spread using pull

$\to$ Understand data using uncertainties





ML for LHC Theory

Developing ML for the best science

- just another numerical tool for a numerical field
- transformative new common language

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

$\rightarrow$ 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]