ML-Unfolding — Case, Ideas, Progress, and News

Tilman Plehn

Universität Heidelberg

PHYSTAT Workshop, Paris, June 2024

Case for (ML-)Unfolding

Number of analyses

- optimal forward inference:
  full signal and background simulations
  high-dimensional, unbinned SBI
- CPU-limitation for many signals

→ Unfold detectors once

Optimal analyses

- theory limiting many LHC analyses
  make best use of continuous progress
- allow for analyses to be updated

→ Unfold detectors/soft QCD and save data

Public LHC data

- common lore:
  LHC data too complicated for amateurs
  no way to even try to publish LHC data
- in truth:
  hard scattering and decay simulations easy
  BSM physics not in hadronization and detector

→ Unfold to hard scattering

High-dimensional and unbinned

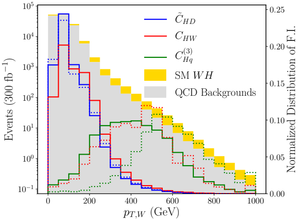
Simple process $pp \to W_\ell H_{bb}$ [Brehmer, Dawson, Homiller, Kling, TP, long time ago]

- example operators [wf vs vertex structure vs 4-point]

$$\widetilde{\mathcal{O}}_{HD} = (\phi^\dagger \phi)\Box(\phi^\dagger \phi) - \frac{1}{4}(\phi^\dagger D^\mu \phi)^* (\phi^\dagger D_\mu \phi)$$

$$\mathcal{O}_{HW} = \phi^\dagger \phi W^a_{\mu\nu} W^{\mu\nu a}$$

$$\mathcal{O}^{(3)}_{Hq} = (\phi^\dagger i\overleftrightarrow{D}^a_\mu \phi)(\overline{Q}_L \sigma^a \gamma^\mu Q_L)$$



High-dimensional and unbinned

Simple process $pp \to W_\ell H_{bb}$ [Brehmer, Dawson, Homiller, Kling, TP, long time ago]

- example operators [wf vs vertex structure vs 4-point]

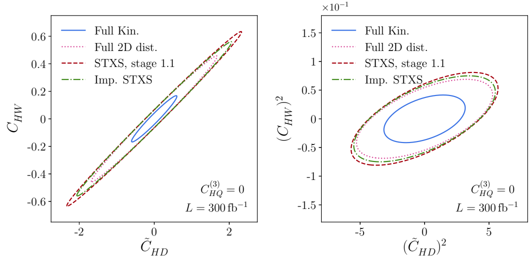
$$\widetilde{\mathcal{O}}_{HD} = (\phi^\dagger \phi)\Box(\phi^\dagger \phi) - \frac{1}{4}(\phi^\dagger D^\mu \phi)^*(\phi^\dagger D_\mu \phi)$$

$$\mathcal{O}_{HW} = \phi^\dagger \phi W^a_{\mu\nu} W^{\mu\nu a}$$

$$\mathcal{O}^{(3)}_{Hq} = (\phi^\dagger i\overleftrightarrow{D}^a_\mu \phi)(\overline{Q}_L \sigma^a \gamma^\mu Q_L)$$

Full kinematics vs $p_{T,W} - m_{T,\text{tot}}$

- bulk operators



High-dimensional and unbinned

Simple process $pp \to W_\ell H_{bb}$ [Brehmer, Dawson, Homiller, Kling, TP, long time ago]

- example operators [wf vs vertex structure vs 4-point]

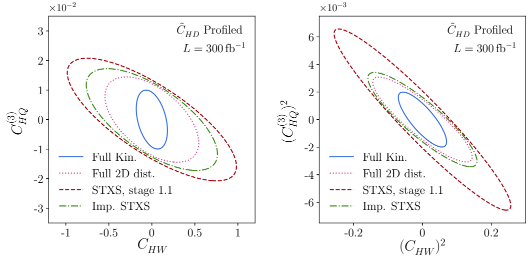
$$\widetilde{\mathcal{O}}_{HD} = (\phi^\dagger \phi)\Box(\phi^\dagger \phi) - \frac{1}{4}(\phi^\dagger D^\mu \phi)^* (\phi^\dagger D_\mu \phi)$$

$$\mathcal{O}_{HW} = \phi^\dagger \phi W^a_{\mu\nu} W^{\mu\nu a}$$

$$\mathcal{O}^{(3)}_{Hq} = (\phi^\dagger i \overleftrightarrow{D}^a_\mu \phi)(\overline{Q}_L \sigma^a \gamma^\mu Q_L)$$

Full kinematics vs $p_{T,W} - m_{T,\text{tot}}$

- bulk operators
- tail operator



$\to$ Full, unbinned kinematics the key [top groups doing better]

Unfolding without and with ML

Basic idea

- 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

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

Unfolding without and with ML

Basic idea

- four phase space distributions

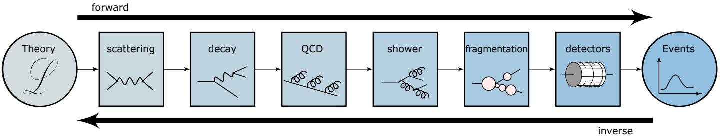
$$\begin{tikzcd}[column sep=large] p_\text{sim}(x_\text{part}) \arrow[r, leftrightarrow, "\text{unfolding inference}"] \arrow[d, "p(x_\text{reco}|x_\text{part})"'] & p_\text{unfold}(x_\text{part}) \\ p_\text{sim}(x_\text{reco}) \arrow[r, leftrightarrow, "\text{forward inference}"] & p_\text{data}(x_\text{reco})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

- two conditional probabilities

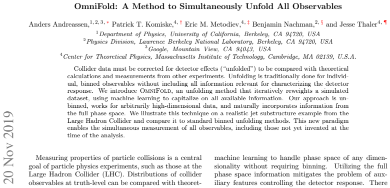
$$p(x_\text{part}|x_\text{reco}) = p(x_\text{reco}|x_\text{part}) \, \frac{p_\text{sim}(x_\text{part})}{p_\text{sim}(x_\text{reco})}$$

LHC simulations

- paired
- stochastic, usually single-mode [nothing LHC is deterministic]
- following energy scale/resolution
- starting from fundamental parameters



Unfolding by reweighting



OmniFold

- use paired events $(x_\text{part}, x_\text{reco})$
  learn $p_\text{sim}(x_\text{reco}) \leftrightarrow p_\text{data}(x_\text{reco})$
  reweight $p_\text{sim}(x_\text{part}) \to p_\text{unfold}(x_\text{part})$

$$\begin{tikzcd}
p_\text{sim}(x_\text{part}) \arrow[r, "\text{classifier weights (3)}"] \arrow[d, leftrightarrow, "\text{pull (2)/push weights(4)}"'] & p_\text{unfold}(x_\text{part}) \\
p_\text{sim}(x_\text{reco}) \arrow[r, leftrightarrow, "\text{classifier weights (1)}"] & p_\text{data}(x_\text{reco})
\end{tikzcd}$$

- unbinned classifier weight [Neyman-Pearson lemma, CWoLa]

$$w_D(x_i) = \frac{D(x_i)}{1 - D(x_i)} \to \frac{p_1(x_i)}{p_2(x_i)}$$

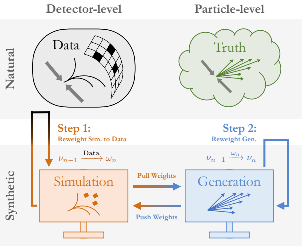
- high-dimensional classification, like jet tagging

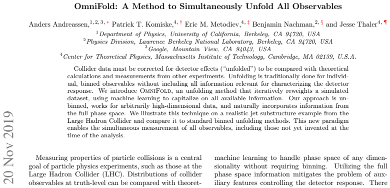
$\to$ Driven by (now) established ML-classification

Unfolding by reweighting

OmniFold

- use paired events $(x_{\text{part}}, x_{\text{reco}})$
  learn $p_{\text{sim}}(x_{\text{reco}}) \leftrightarrow p_{\text{data}}(x_{\text{reco}})$
  reweight $p_{\text{sim}}(x_{\text{part}}) \to p_{\text{unfold}}(x_{\text{part}})$





OmniFold: A Method to Simultaneously Unfold All Observables

Anders Andreassen,[1,2,3,*] Patrick T. Komiske,[4,†] Eric M. Metodiev,[4,‡] Benjamin Nachman,[2,§] and Jesse Thaler[4,¶]

[1] *Department of Physics, University of California, Berkeley, CA 94720, USA*
[2] *Physics Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA*
[3] *Google, Mountain View, CA 94043, USA*
[4] *Center for Theoretical Physics, Massachusetts Institute of Technology, Cambridge, MA 02139, U.S.A.*

Collider data must be corrected for detector effects ("unfolded") to be compared with theoretical calculations and measurements from other experiments. Unfolding is traditionally done for individual, binned observables without including all information relevant for characterizing the detector response. We introduce OmniFold, an unfolding method that iteratively reweights a simulated dataset, using machine learning to capitalize on all available information. Our approach is unbinned, works for arbitrarily high-dimensional data, and naturally incorporates information from the full phase space. We illustrate this technique on a realistic jet substructure example from the Large Hadron Collider and compare it to standard binned unfolding methods. This new paradigm enables the simultaneous measurement of all observables, including those not yet invented at the time of the analysis.

20 Nov 2019

Measuring properties of particle collisions is a central goal of particle physics experiments, such as those at the Large Hadron Collider (LHC). Distributions of collider observables at truth-level can be compared with theoret- machine learning to handle phase space of any dimensionality without requiring binning. Utilizing the full phase space information mitigates the problem of auxiliary features controlling the detector response. There

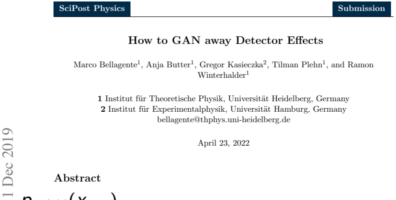
- unbinned classifier weight [Neyman-Pearson lemma, CWoLa]

$$w_D(x_i) = \frac{D(x_i)}{1 - D(x_i)} \quad \to \quad \frac{p_1(x_i)}{p_2(x_i)}$$

- high-dimensional classification, like jet tagging

$\to$ Driven by (now) established ML-classification

Unfolding by generation


SciPost Physics Submission

How to GAN away Detector Effects

Marco Bellagente[1], Anja Butter[1], Gregor Kasieczka[2], Tilman Plehn[1], and Ramon Winterhalder[1]

1 Institut für Theoretische Physik, Universität Heidelberg, Germany
2 Institut für Experimentalphysik, Universität Hamburg, Germany
bellagente@thphys.uni-heidelberg.de

April 23, 2022

1 Dec 2019

Abstract

Sampling conditional probability

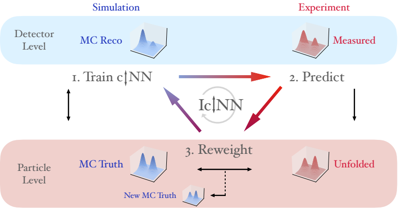
- just like forward ML-generation
- learn inverse conditional probability also from paired events $(x_{\text{part}}, x_{\text{reco}})$

$$
\begin{array}{ccc}
p_{\text{sim}}(x_{\text{part}}) & & p_{\text{unfold}}(x_{\text{part}}) \\
\text{paired data} \updownarrow & & \uparrow p_{\text{model}}(x_{\text{part}} | x_{\text{reco}}) \\
p_{\text{sim}}(x_{\text{reco}}) & \xleftrightarrow{\text{correspondence}} & p_{\text{data}}(x_{\text{reco}})
\end{array}
$$

Two improvements needed [taking some time]

1. likelihood loss to generate posterior $\rightarrow$ cINN
2. remove training prior $\rightarrow$ IcINN [Backes, Butter, Dunford, Malaescu]

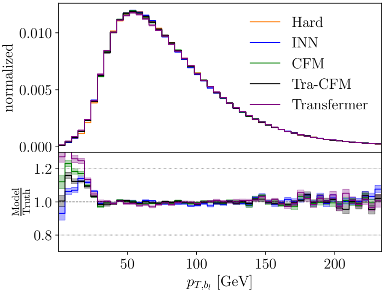
$\rightarrow$ Driven by generative networks



Further improvements from generative AI

Generative networks for the LHC

- phase space integration
  event generation
  calorimeter shower simulation
  MEM inference
  unfolding
  generative inference [astro/cosmo/GW]
- built-in smoothness [regularization]
- since 2019
  GAN $\rightarrow$ INN $\rightarrow$ CFM
- combinatorics $\rightarrow$ TraCFM
- features: learned classifiers
  uncertainties: Bayesian networks
  precision: classifier weights
- phase space parametrization important

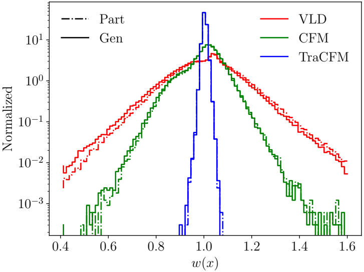


Further improvements from generative AI

Generative networks for the LHC

- phase space integration
  event generation
  calorimeter shower simulation
  MEM inference
  unfolding
  generative inference [astro/cosmo/GW]
- built-in smoothness [regularization]
- since 2019
  GAN $\to$ INN $\to$ CFM
- combinatorics $\to$ TraCFM
- features: learned classifiers
  uncertainties: Bayesian networks
  precision: classifier weights
- phase space parametrization important

$\to$ ML-progress, see Nathan & Javier



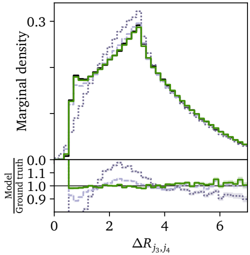
Further improvements from generative AI

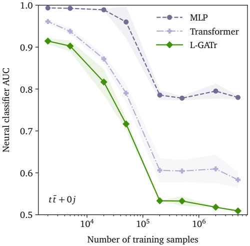
Generative networks for the LHC

- phase space integration
  event generation
  calorimeter shower simulation
  MEM inference
  unfolding
  generative inference [astro/cosmo/GW]
- built-in smoothness [regularization]
- since 2019
  GAN $\to$ INN $\to$ CFM
- combinatorics $\to$ TraCFM
- features: learned classifiers
  uncertainties: Bayesian networks
  precision: classifier weights
- phase space parametrization important

$\to$ ML-progress, see Nathan & Javier

$\to$ further improvements coming
Lorentz-covariant GATr-CFM [$t\bar{t} + 4j$]





Unfolding top decays

Enjoying a technical challenge [Favaro, Kogler, Paasch, Palacios Schweitzer, TP, Schwarz]

- Terascale statistics school:

TP	'So what would nobody ever use unfolding for?'
experts	'mass measurements, clearly'
TP	(becoming very quiet)

Unfolding top decays

Enjoying a technical challenge [Favaro, Kogler, Paasch, Palacios Schweitzer, TP, Schwarz]

- first measure m_t in unfolded boosted decays
  then unfold kinematics of 3 subjets
- model dependence m_s vs 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

Enjoying a technical challenge [Favaro, Kogler, Paasch, Palacios Schweitzer, TP, Schwarz]

- first measure m_t in unfolded boosted decays
  then unfold kinematics of 3 subjets
- 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 range of m_s-values
2. strengthen data by including batch-wise $m_d \sim M_{jjj} \in x_\text{reco}$

Unfolding top decays

Enjoying a technical challenge [Favaro, Kogler, Paasch, Palacios Schweitzer, TP, Schwarz]

- first measure m_t in unfolded boosted decays
  then unfold kinematics of 3 subjets
- complete training bias $m_d \to m_s$ [too bad to reweight]

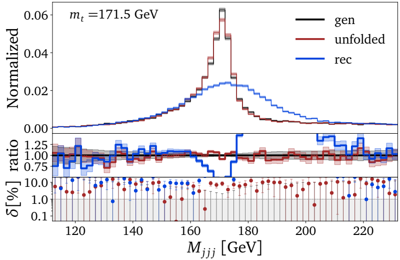
$$\begin{tikzcd}
p_\text{sim}(x_\text{part}|m_s) \arrow[d, "p(x_\text{reco}|x_\text{part})"'] & & p_\text{unfold}(x_\text{part}|m_s, \cancel{m_d}) \\
p_\text{sim}(x_\text{reco}|m_s) \arrow[rr, leftrightarrow, "\text{correspondence}"] & & p_\text{data}(x_\text{reco}|m_d) \arrow[u, "p_\text{model}(x_\text{part}|x_\text{reco},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

1 weaken bias by training on range of m_s-values

2 strengthen data by including batch-wise $m_d \sim M_{jjj} \in x_\text{reco}$

Preliminary unfolding results [TraCFM]

- 4D-masses for in-situ calibration



Unfolding top decays

Enjoying a technical challenge [Favaro, Kogler, Paasch, Palacios Schweitzer, TP, Schwarz]

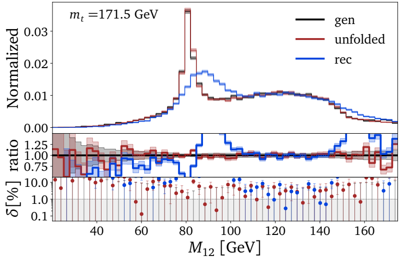
- first measure m_t in unfolded boosted decays
  then unfold kinematics of 3 subjets
- complete training bias $m_d \to m_s$ [too bad to reweight]

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

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

Preliminary unfolding results [TraCFM]

- 4D-masses for in-situ calibration



Unfolding top decays

Enjoying a technical challenge [Favaro, Kogler, Paasch, Palacios Schweitzer, TP, Schwarz]

- first measure m_t in unfolded boosted decays
  then unfold kinematics of 3 subjets
- complete training bias $m_d \to m_s$ [too bad to reweight]

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

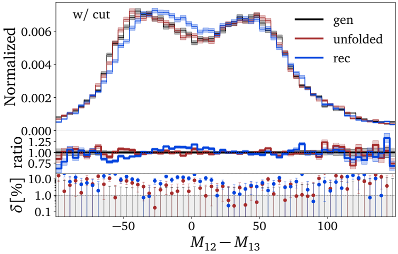
1 weaken bias by training on range of m_s-values

2 strengthen data by including batch-wise $m_d \sim M_{jjj} \in x_\text{reco}$

Preliminary unfolding results [TraCFM]

- 4D-masses for in-situ calibration
- full 12D for publication

$\to$ CMS data next



Unfolding top decays

Enjoying a technical challenge [Favaro, Kogler, Paasch, Palacios Schweitzer, TP, Schwarz]

- first measure m_t in unfolded boosted decays
  then unfold kinematics of 3 subjets
- complete training bias $m_d \to m_s$ [too bad to reweight]

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

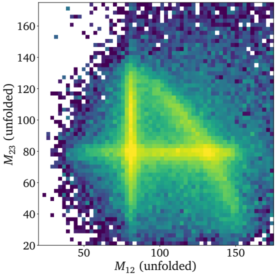
1 weaken bias by training on range of m_s-values

2 strengthen data by including batch-wise $m_d \sim M_{jjj} \in x_\text{reco}$

Preliminary unfolding results [TraCFM]

- 4D-masses for in-situ calibration
- full 12D for publication

$\to$ CMS data next

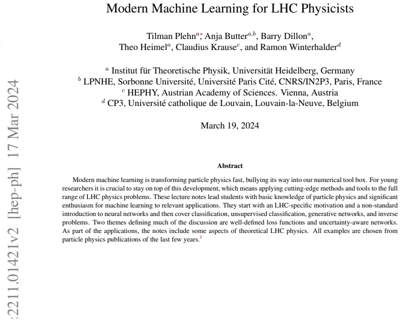


Outlook

Unfolding LHC data

- efficient analyses
  optimal updated analyses
  public LHC data
- my personal dream
- LHC-inverse problem
  unbinned & high-dimensional
- ML (just) the transformative tool

→ reweighting + conditional 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]