Challenges in Theory and AI/ML

Tilman Plehn

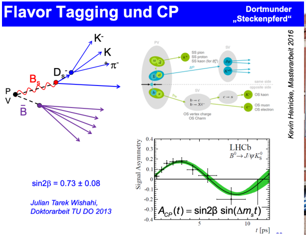
Universität Heidelberg

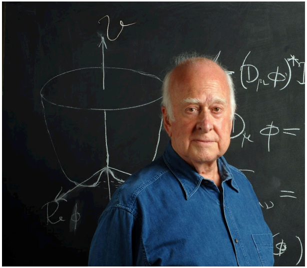
SLAC Summer Institute, August 2023

Modern LHC physics

Classic motivation

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





Modern LHC physics

Classic motivation

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

LHC physics

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

Modern LHC physics

Classic motivation

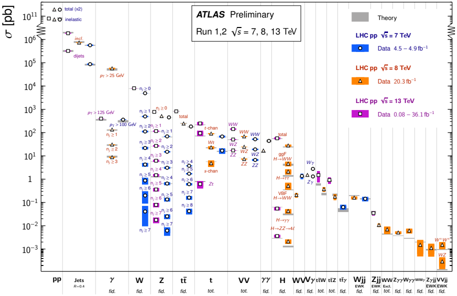
- dark matter?
- baryogenesis?
- origin of Higgs field?

LHC physics

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

Successful past

- measurements of event counts
- analyses inspired by simulation
- model-driven Higgs discovery



Modern LHC physics

Classic motivation

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

LHC physics

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

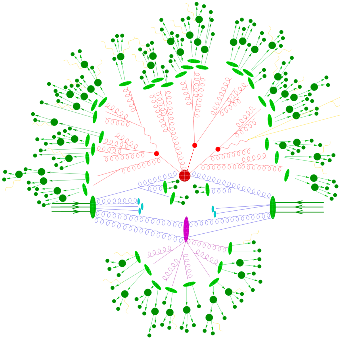
Successful past

- measurements of event counts
- 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?
- baryogenesis?
- origin of Higgs field?

LHC physics

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

Successful past

- measurements of event counts
- analyses inspired by simulation
- model-driven Higgs discovery

First-principle, precision simulations

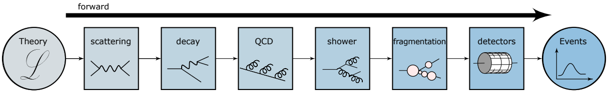
- start with Lagrangian
- calculate scattering using QFT
- simulate collisions
- simulate detectors

$\rightarrow$ LHC collisions in virtual worlds

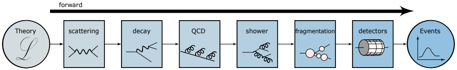
BSM searches

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

$\rightarrow$ Lots of data science...



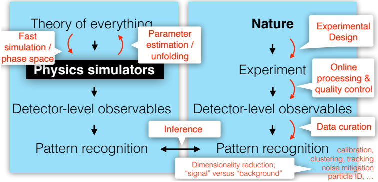
Role of theory



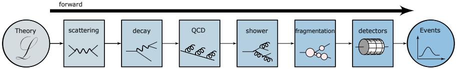
First-principle simulations

- start with Lagrangian
  generate Feynman diagrams
- compute hard scattering amplitudes
  for on-shell, include decays
  add QCD jet radiation [ISR/FSR]
- add parton shower [still QCD]
  push fragmentation towards QCD
- all theory, except for detectors

→ Simulations, not modeling!



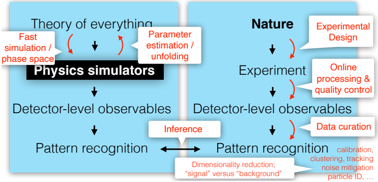
Role of theory



First-principle simulations

- start with Lagrangian
  generate Feynman diagrams
- compute hard scattering amplitudes
  for on-shell, include decays
  add QCD jet radiation [ISR/FSR]
- add parton shower [still QCD]
  push fragmentation towards QCD
- all theory, except for detectors

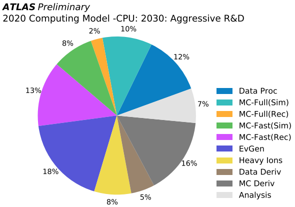
→ Simulations, not modeling!



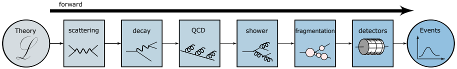
Pythia/Madgraph/Sherpa... for HL-LHC

- factor 25 more expected (= simulated) data
- more complex final states
  higher-orders precision
- parameter coverage for signals
- enable analysis reinterpretation?
  enable global LHC analyses?

→ Theory nightmare



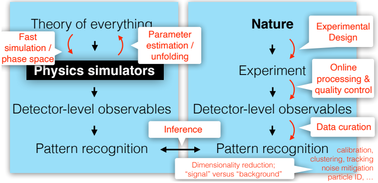
Role of theory



First-principle simulations

- start with Lagrangian
  generate Feynman diagrams
- compute hard scattering amplitudes
  for on-shell, include decays
  add QCD jet radiation [ISR/FSR]
- add parton shower [still QCD]
  push fragmentation towards QCD
- all theory, except for detectors

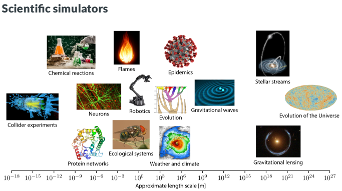
→ Simulations, not modeling!



LHC-specific explainable AI

- SBI conditional on theory simulations
- understanding LHC data is QFT
- computing speed means precision
- control critical
- uncertainties crucial
- phase space interpretable

→ Well-defined, but non-standard AI/ML

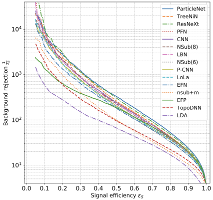


ML-applications in experiment

Top tagging [supervised classification]

- ‘hello world’ of LHC-ML
- end of QCD-taggers
- different NN-architectures

→ Non-NN left in the dust...



SciPost Physics | Submission

The Machine Learning Landscape of Top Taggers

G. Kasieczka (ed)[1], T. Plehn (ed)[2], A. Butter[2], K. Cranmer[3], D. Debnath[4], B. M. Dillon[5], M. Fairbairn[6], D. A. Faroughy[5], W. Fedorko[7], C. Gay[7], L. Gouskos[8], J. F. Kamenik[5,9], P. T. Komiske[10], S. Leiss[1], A. Lister[7], S. Macaluso[3,4], E. M. Metodiev[10], L. Moore[11], B. Nachman,[12,13], K. Nordström[14,15], J. Pearkes[7], H. Qu[8], Y. Rath[16], M. Rieger[16], D. Shih[4], J. M. Thompson[2], and S. Varma[6]

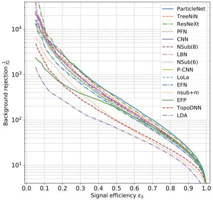
1 Institut für Experimentalphysik, Universität Hamburg, Germany
2 Institut für Theoretische Physik, Universität Heidelberg, Germany
3 Center for Cosmology and Particle Physics and Center for Data Science, NYU, USA
4 NHECT, Dept. of Physics and Astronomy, Rutgers, The State University of NJ, USA
5 Jozef Stefan Institute, Ljubljana, Slovenia
6 Theoretical Particle Physics and Cosmology, King's College London, United Kingdom
7 Department of Physics and Astronomy, The University of British Columbia, Canada
8 Department of Physics, University of California, Santa Barbara, USA
9 Faculty of Mathematics and Physics, University of Ljubljana, Ljubljana, Slovenia
10 Center for Theoretical Physics, MIT, Cambridge, USA
11 CP3, Universitéxx Catholique de Louvain, Louvain-la-Neuve, Belgium
12 Physics Division, Lawrence Berkeley National Laboratory, Berkeley, USA
13 Simons Inst. for the Theory of Computing, University of California, Berkeley, USA
14 National Institute for Subatomic Physics (NIKHEF), Amsterdam, Netherlands
15 LPTHE, CNRS & Sorbonne Université, Paris, France
16 III. Physics Institute A, RWTH Aachen University, Germany

ML-applications in experiment

Top tagging [supervised classification]

- ‘hello world’ of LHC-ML
- end of QCD-taggers
- different NN-architectures

→ Non-NN left in the dust...



SciPost Physics — Submission

The Machine Learning Landscape of Top Taggers

G. Kasieczka (ed)[1], T. Plehn (ed)[2], A. Butter[2], K. Cranmer[3], D. Debnath[4], B. M. Dillon[5], M. Fairbairn[6], D. A. Faroughy[5], W. Fedorko[7], C. Gay[7], L. Gouskos[8], J. F. Kamenik[5,9], P. T. Komiske[10], S. Leiss[1], A. Lister[7], S. Macaluso[3,4], E. M. Metodiev[10], L. Moore[11], B. Nachman,[12,13], K. Nordström[14,15], J. Pearkes[7], H. Qu[8], Y. Rath[16], M. Rieger[16], D. Shih[4], J. M. Thompson[2], and S. Varma[6]

1 Institut für Experimentalphysik, Universität Hamburg, Germany
2 Institut für Theoretische Physik, Universität Heidelberg, Germany
3 Center for Cosmology and Particle Physics and Center for Data Science, NYU, USA
4 NHECT, Dept. of Physics and Astronomy, Rutgers, The State University of NJ, USA
5 Jozef Stefan Institute, Ljubljana, Slovenia
6 Theoretical Particle Physics and Cosmology, King's College London, United Kingdom
7 Department of Physics and Astronomy, The University of British Columbia, Canada
8 Department of Physics, University of California, Santa Barbara, USA
9 Faculty of Mathematics and Physics, University of Ljubljana, Ljubljana, Slovenia
10 Center for Theoretical Physics, MIT, Cambridge, USA
11 CP3, Universitéxx Catholique de Louvain, Louvain-la-Neuve, Belgium
12 Physics Division, Lawrence Berkeley National Laboratory, Berkeley, USA
13 Simons Inst. for the Theory of Computing, University of California, Berkeley, USA
14 National Institute for Subatomic Physics (NIKHEF), Amsterdam, Netherlands
15 LPTHE, CNRS & Sorbonne Université, Paris, France
16 III. Physics Institute A, RWTH Aachen University, Germany

Particle flow [classification, super-resolution]

- mother of jet tools
- combined detector channels
- similar studies in CMS

→ Beyond just concepts

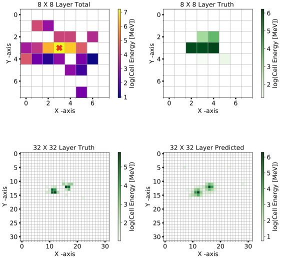


Fig. 7: An event display of total energy shower (within topocluster), as captured by a calorimeter layer of 8×8 granularity, along with the location of the track, denoted by a red cross (left) and the same shower is captured by a calorimeter layer of 32×32 granularity (middle). The bottom right panel shows the corresponding event predicted by the NN. The figure shows that the shower originating from a $\pi^0 \to \gamma\gamma$ is resolved by a 32×32 granularity layer.

Towards a Computer Vision Particle Flow ⋆

Francesco Armando Di Bello[a,3], Sanmay Ganguly[b,1], Eilam Gross[1], Marumi Kado[3,4], Michael Pitt[2], Lorenzo Santi [3], Jonathan Shlomi[1]

[1]Weizmann Institute of Science, Rehovot 76100, Israel
[2]CERN, CH 1211, Geneva 23, Switzerland
[3]Università di Roma Sapienza, Piazza Aldo Moro, 2, 00185 Roma, Italy e INFN, Italy
[4]Université Paris-Saclay, CNRS/IN2P3, IJCLab, 91405, Orsay, France

Jets and parton densities

Anomaly searches [unsupervised training]

- train on QCD-jets, SM-events
- look for non-QCD jets, non-SM events

→ Autoencoders

SciPost Physics | Submission

Better Latent Spaces for Better Autoencoders

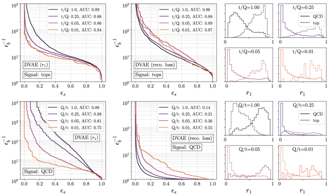
Barry M. Dillon[1], Tilman Plehn[1], Christof Sauer[2], and Peter Sorrenson[3],

1 Institut für Theoretische Physik, Universität Heidelberg, Germany
2 Physikalisches Institut, Universität Heidelberg, Germany
3 Heidelberg Collaboratory for Image Processing, Universität Heidelberg, Germany

April 20, 2021

Abstract

Autoencoders as tools behind anomaly searches at the LHC have the structural problem that they only work in one direction, extracting jets with higher complexity but not the other way around. To address this, we derive classifiers from the latent space of (variational) autoencoders, specifically in Gaussian mixture and Dirichlet latent spaces. In particular, the Dirichlet setup solves the problem and improves both the performance and the interpretability of the networks.



Jets and parton densities

Anomaly searches [unsupervised training]

- train on QCD-jets, SM-events
- look for non-QCD jets, non-SM events

→ Autoencoders

SciPost Physics — Submission

Better Latent Spaces for Better Autoencoders

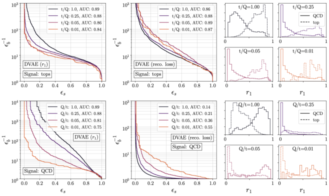
Barry M. Dillon[1], Tilman Plehn[1], Christof Sauer[2], and Peter Sorrenson[3],

1 Institut für Theoretische Physik, Universität Heidelberg, Germany
2 Physikalisches Institut, Universität Heidelberg, Germany
3 Heidelberg Collaboratory for Image Processing, Universität Heidelberg, Germany

April 20, 2021

Abstract

Autoencoders as tools behind anomaly searches at the LHC have the structural problem that they only work in one direction, extracting jets with higher complexity but not the other way around. To address this, we derive classifiers from the latent space of (variational) autoencoders, specifically in Gaussian mixture and Dirichlet latent spaces. In particular, the Dirichlet setup solves the problem and improves both the performance and the interpretability of the networks.



NNPDF/N3PDF parton densities [full blast]

- starting point: pdfs without functional ansatz
- moving on: cutting-edge ML everywhere

→ Leaders in ML-theory

A data-based parametrization of parton distribution functions

Stefano Carrazza[1,2,3], Juan Cruz-Martinez[1], and Roy Stegeman[1]

[1] TIF Lab, Dipartimento di Fisica, Università degli Studi di Milano and INFN Sezione di Milano.
[2] CERN, Theoretical Physics Department, CH-1211 Geneva 23, Switzerland.
[3] Quantum Research Centre, Technology Innovation Institute, Abu Dhabi, UAE.

Received: date / Revised version: date

Abstract. Since the first determination of a structure function many decades ago, all methodologies used to determine structure functions or parton distribution functions (PDFs) have employed a common prefactor as part of the parametrization. The NNPDF collaboration pioneered the use of neural networks to overcome the inherent bias of constraining the space of solution with a fixed functional form while still keeping the same common prefactor as a preprocessing. Over the years various, increasingly sophisticated, techniques have been introduced to counter the effect of the prefactor on the PDF determination. In this paper we present a methodology to remove the prefactor entirely, thereby significantly simplifying the methodology, without a loss of efficiency and finding good agreement with previous results.

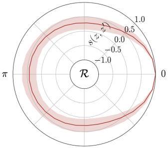
PACS. 12.38.-t Quantum chromodynamics – 12.39.-x Phenomenological quark models – 84.35.+i Neural Networks

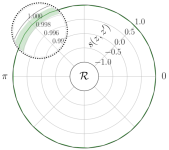
Symmetries

Symmetric networks [contrastive learning, transformer network]

- rotations, translations, permutations, soft splittings, collinear splittings
- learn symmetries/augmentations

→ Symmetric latent representation





SciPost Physics Submission

Symmetries, Safety, and Self-Supervision

Barry M. Dillon[1], Gregor Kasieczka[2], Hans Olischlager[1], Tilman Plehn[1], Peter Sorrenson[3], and Lorenz Vogel[1]

1 Institut für Theoretische Physik, Universität Heidelberg, Germany
2 Institut für Experimentalphysik, Universität Hamburg, Germany
3 Heidelberg Collaboratory for Image Processing, Universität Heidelberg, Germany

August 11, 2021

Abstract

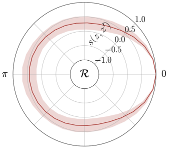
Collider searches face the challenge of defining a representation of high-dimensional data such that physical symmetries are manifest, the discriminating features are retained, and the choice of representation is new-physics agnostic. We introduce JetCLR to solve the mapping from low-level data to optimized observables though self-supervised contrastive learning. As an example, we construct a data representation for top and QCD jets using a permutation-invariant transformer-encoder network and visualize its symmetry properties. We compare the JetCLR representation with alternative representations using linear classifier tests and find it to work quite well.

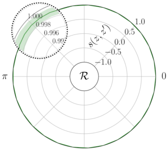
Symmetries

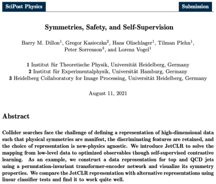
Symmetric networks [contrastive learning, transformer network]

- rotations, translations, permutations, soft splittings, collinear splittings
- learn symmetries/augmentations

→ Symmetric latent representation







SciPost Physics — Submission

Symmetries, Safety, and Self-Supervision

Barry M. Dillon[1], Gregor Kasieczka[2], Hans Olischlager[1], Tilman Plehn[1], Peter Sorrenson[3], and Lorenz Vogel[1]

1 Institut für Theoretische Physik, Universität Heidelberg, Germany
2 Institut für Experimentalphysik, Universität Hamburg, Germany
3 Heidelberg Collaboratory for Image Processing, Universität Heidelberg, Germany

August 11, 2021

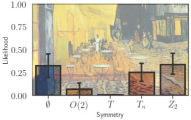
Abstract

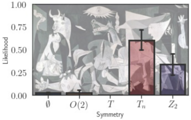
Collider searches face the challenge of defining a representation of high-dimensional data such that physical symmetries are manifest, the discriminating features are retained, and the choice of representation is new-physics agnostic. We introduce JetCLR to solve the mapping from low-level data to optimized observables though self-supervised contrastive learning. As an example, we construct a data representation for top and QCD jets using a permutation-invariant transformer-encoder network and visualize its symmetry properties. We compare the JetCLR representation with alternative representations using linear classifier tests and find it to work quite well.

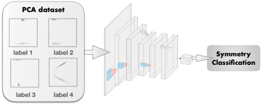
Learning symmetries [representation, visualization]

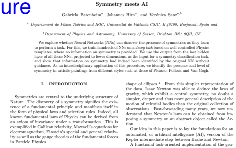
- (particle) physics is all symmetries
- identify symmetries in 2D systems [paintings]

→ Networks representing structure









Symmetry meets AI

Gabriela Barenboim[a], Johannes Hirn[a], and Verónica Sanz[a,b]

[a] Departament de Física Teòrica and IFIC, Universitat de València-CSIC, E-46100, Burjassot, Spain and
[b] Department of Physics and Astronomy, University of Sussex, Brighton BN1 9QH, UK

We explore whether Neural Networks (NNs) can *discover* the presence of symmetries as they learn to perform a task. For this, we train hundreds of NNs on a *decoy task* based on well-controlled Physics templates, where no information on symmetry is provided. We use the output from the last hidden layer of all these NNs, projected to fewer dimensions, as the input for a symmetry classification task, and show that information on symmetry had indeed been identified by the original NN without guidance. As an interdisciplinary application of this procedure, we identify the presence and level of symmetry in artistic paintings from different styles such as those of Picasso, Pollock and Van Gogh.

I. INTRODUCTION

Symmetries are central to the underlying structure of Nature. The discovery of a symmetry signifies the existence of a fundamental principle and manifests itself in the form of physical laws and selection rules. Indeed, all known fundamental laws of Physics can be derived from an axiom of invariance under a transformation. This is exemplified in Galilean relativity, Maxwell's equations for electromagnetism, Einstein's special and general relativity as well as the gauge theories of the fundamental forces in Particle Physics.

shape of ellipses [1]. From this simpler representation of the data, Isaac Newton was able to deduce the laws of gravity, which exhibit a central symmetry, no doubt a simpler, deeper and thus more general description of the motion of celestial bodies than the original collection of observations. Fast-forwarding many years, we now understand that Newton's laws can be obtained from imposing a symmetry on an abstract object called the Action.

Our idea in this paper is to lay the foundations for an automated, or artificial intelligence (AI), version of the Kepler intermediate step between Brahe and Newton.

A functional task-oriented implementation of the gen-

Theory-AI

Tilman Plehn

LHC physics

Examples

Generation

Control

Uncertainty

Testing

Inversion

Integrals and perturbative QFT

Learning integrands and integrals [differentiable activations]

- learn integrand through differiable network
- evalute integral as promitive

$\to$ Novel ML-integrator

boundaries. Analytically, we would compute the primitive F,

$$\frac{d^D F(s;x)}{dx_1 \dots dx_D} = f(s;x) \,, \tag{1.80}$$

and then the integral by evaluating the integration boundaries

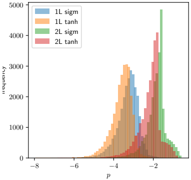
$$\begin{aligned} I(s) &= \int_0^1 dx_1 \cdots \int_0^1 dx_D \, \frac{d^D F(s;x)}{dx_1 \dots dx_D} \\ &= \int_0^1 dx_1 \cdots \int_0^1 dx_{D-1} \, \frac{d^{D-1} F(s;x)}{dx_1 \dots dx_{D-1}} \Bigg|_{x_D=0}^{x_D=1} \\ &= \sum_{x_1,\dots,x_D=0,1} (-1)^{D-\sum x_i} F(s;x) \,. \end{aligned} \tag{1.81}$$

In particle physics we really never know the primitive of a phase space integrand, but we can try to construct it and encode it in a neural network,

$$F_\theta(s;x) \approx F(s;x) \,. \tag{1.82}$$

On the other hand, we do not have data to train a surrogate network for F directly. The idea is to instead train an integrand surrogate, such that its D-th derivative matches f,

$$\mathcal{L}_{\text{MSE}} \left(f(s;x), \frac{dF_\theta(s;x)}{dx_1 \dots dx_D} \right) \,. \tag{1.83}$$



Published for SISSA by Springer
Received: December 6, 2022
Accepted: March 14, 2023
Published: March 28, 2023

Multi-variable integration with a neural network

D. Maître[a,1] and R. Santos-Mateos[b]

[a] Institute for Particle Physics Phenomenology, Physics Department, Durham University, Durham DH1 3LE, U.K.
[b] Department of Electronics and Computing, University of Santiago de Compostela, Santiago de Compostela, Spain

E-mail: daniel.maitre@durham.ac.uk, roi.santos@usc.es

Abstract: In this article we present a method for automatic integration of parametric integrals over the unit hypercube using a neural network. The method fits a neural network to the primitive of the integrand using a loss function designed to minimize the difference between multiple derivatives of the network and the function to be integrated. We apply this method to two example integrals resulting from the sector decomposition of a one-loop and two-loop scalar integrals. Our method can achieve per-mil and percent accuracy for these integrals over a range of invariant values. Once the neural network is fitted, the evaluation of the integral is between 40 and 125 times faster than the usual numerical integration method for our examples, and we expect the speed gain to increase with the complexity of the integrand.

Integrals and perturbative QFT

Learning integrands and integrals [differentiable activations]

- learn integrand through differiable network
- evalute integral as promitive

→ Novel ML-integrator

boundaries. Analytically, we would compute the primitive F,

$$\frac{d^D F(s;x)}{dx_1 \dots dx_D} = f(s;x)\,, \tag{1.80}$$

and then the integral by evaluating the integration boundaries

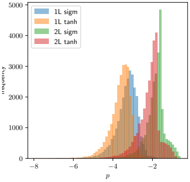
$$\begin{aligned} I(s) &= \int_0^1 dx_1 \cdots \int_0^1 dx_D \, \frac{d^D F(s;x)}{dx_1 \dots dx_D} \\ &= \int_0^1 dx_1 \cdots \int_0^1 dx_{D-1} \, \frac{d^{D-1} F(s;x)}{dx_1 \dots dx_{D-1}} \Bigg|_{x_D=0}^{x_D=1} \\ &= \sum_{x_1,\dots,x_D=0,1} (-1)^{D-\sum x_i} F(s;x)\,. \end{aligned} \tag{1.81}$$

In particle physics we really never know the primitive of a phase space integrand, but we can try to construct it and encode it in a neural network,

$$F_\theta(s;x) \approx F(s;x)\,. \tag{1.82}$$

On the other hand, we do not have data to train a surrogate network for F directly. The idea is to instead train an integrand surrogate, such that its D-th derivative matches f,

$$\mathcal{L}_{\text{MSE}}\left(f(s;x), \frac{dF_\theta(s;x)}{dx_1 \dots dx_D} \right)\,. \tag{1.83}$$



JHEP
Published for SISSA by Springer
Received: December 6, 2022
Accepted: March 14, 2023
Published: March 28, 2023

Multi-variable integration with a neural network

D. Maître[a,1] and R. Santos-Mateos[b]

[a] Institute for Particle Physics Phenomenology, Physics Department, Durham University, Durham DH1 3LE, U.K.

[b] Department of Electronics and Computing, University of Santiago de Compostela, Santiago de Compostela, Spain

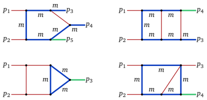
E-mail: daniel.maitre@durham.ac.uk, roi.santos@usc.es

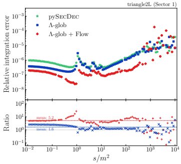
Abstract: In this article we present a method for automatic integration of parametric integrals over the unit hypercube using a neural network. The method fits a neural network to the primitive of the integrand using a loss function designed to minimize the difference between multiple derivatives of the network and the function to be integrated. We apply this method to two example integrals resulting from the sector decomposition of a one-loop and two-loop scalar integrals. Our method can achieve per-mil and percent accuracy for these integrals over a range of invariant values. Once the neural network is fitted, the evaluation of the integral is between 40 and 125 times faster than the usual numerical integration method for our examples, and we expect the speed gain to increase with the complexity of the integrand.

Learning integration paths [invertible networks]

- find optimal integration paths
- learn variable transformation

→ Theory-integrator





SciPost
SciPost Phys. 12, 129 (2022)

Targeting multi-loop integrals with neural networks

Ramon Winterhalder[1,2,3], Vitaly Magerya[4], Emilio Villa[4], Stephen P. Jones[5], Matthias Kerner[4,6], Anja Butter[1,2], Gudrun Heinrich[2,4] and Tilman Plehn[1,2]

1 Institut für Theoretische Physik, Universität Heidelberg, Germany
2 HEiKA - Heidelberg Karlsruhe Strategic Partnership, Heidelberg University, Karlsruhe Institute of Technology (KIT), Germany
3 Centre for Cosmology, Particle Physics and Phenomenology (CP3), Université catholique de Louvain, Belgium
4 Institut für Theoretische Physik, Karlsruher Institut für Technologie, Germany
5 Institute for Particle Physics Phenomenology, Durham University, UK
6 Institut für Astroteilchenphysik, Karlsruher Institut für Technologie, Germany

Abstract

Numerical evaluations of Feynman integrals often proceed via a deformation of the integration contour into the complex plane. While valid contours are easy to construct, the numerical precision for a multi-loop integral can depend critically on the chosen contour. We present methods to optimize this contour using a combination of optimized, global complex shifts and a normalizing flow. They can lead to a significant gain in precision.

Theory-AI

Tilman Plehn

LHC physics

Examples

Generation

Control

Uncertainty

Testing

Inversion

Event generation

Speeding up Sherpa and MadNIS [INNs, sampling]

- precision simulations limiting factor for Runs 3&4
- unweighting critical

→ Phase space sampling

	$gg \to t\bar{t}ggg$	$ug \to t\bar{t}ggu$	$uu \to t\bar{t}guu$	$u\bar{u} \to t\bar{t}gd\bar{d}$
ϵ_{full}	1.1e−2	7.3e−3	6.8e−3	6.6e−4
$\epsilon_{\text{1st,surr}}$	8.7e−3	5.8e−3	4.7e−3	3.6e−4
$\langle t_{\text{full}}\rangle / \langle t_{\text{surr}}\rangle$	39312	2417	199	64
$x^{\text{p.m.}}_{\text{max}}$	52.03	32.52	69.76	326.19
$\epsilon^{\text{p.m.}}_{\text{2nd,surr}}$	2.4e−2	3.8e−2	2.1e−2	5.6e−3
$\alpha^{\text{p.m.}}$	0.9989	0.9984	0.9994	0.9981
$f^{\text{p.m.}}_{\text{eff}}$	2.21	4.89	1.47	0.19
$x^{\text{med}}_{\text{max}}$	30.40	19.14	27.78	25.34
$\epsilon^{\text{med}}_{\text{2nd,surr}}$	4.3e−2	6.4e−2	5.1e−2	7.1e−2
α^{med}	0.9983	0.9966	0.9943	0.9321
$f^{\text{med}}_{\text{eff}}$	3.90	8.26	3.91	2.22

Table 6: Performance measures for partonic channels contributing to $t\bar{t}$+3 jets production at the LHC.

SciPost Physics Submission

MCNET-21-13

Accelerating Monte Carlo event generation – rejection sampling using neural network event-weight estimates

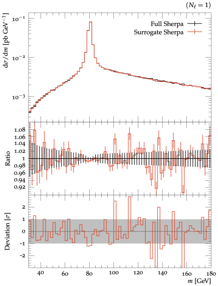
K. Danziger[1], T. Janßen[2], S. Schumann[2], F. Siegert[1]

1 Institut für Kern- und Teilchenphysik, TU Dresden, Dresden, Germany
2 Institut für Theoretische Physik, Georg-August-Universität Göttingen, Göttingen, Germany

September 27, 2021

Abstract

The generation of unit-weight events for complex scattering processes presents a severe challenge to modern Monte Carlo event generators. Even when using sophisticated phase-space sampling techniques adapted to the underlying transition matrix elements, the efficiency for generating unit-weight events from weighted samples can become a limiting factor in practical applications. Here we present a novel two-staged unweighting procedure that makes use of a neural-network surrogate for the full event weight. The algorithm can significantly accelerate the unweighting process, while it still guarantees unbiased sampling from the correct target distribution. We apply, validate and benchmark the new approach in high-multiplicity LHC production processes, including Z/W+4 jets and $t\bar{t}$+3 jets, where we find speed-up factors up to ten.



Event generation

Speeding up Sherpa and MadNIS [INNs, sampling]

- precision simulations limiting factor for Runs 3&4
- unweighting critical

→ Phase space sampling

	$gg \to t\bar{t}ggg$	$ug \to t\bar{t}ggu$	$uu \to t\bar{t}guu$	$u\bar{u} \to t\bar{t}gd\bar{d}$
ϵ_{full}	1.1e−2	7.3e−3	6.8e−3	6.6e−4
$\epsilon_{\text{1st,surr}}$	8.7e−3	5.8e−3	4.7e−3	3.6e−4
$\langle t_{\text{full}}\rangle/\langle t_{\text{surr}}\rangle$	39312	2417	199	64
$x_{\max}^{\text{p.m.}}$	52.03	32.52	69.76	326.19
$\epsilon_{\text{2nd,surr}}^{\text{p.m.}}$	2.4e−2	3.8e−2	2.1e−2	5.6e−3
$\alpha^{\text{p.m.}}$	0.9989	0.9984	0.9994	0.9981
$f_{\text{eff}}^{\text{p.m.}}$	2.21	4.89	1.47	0.19
$x_{\max}^{\text{med}}$	30.40	19.14	27.78	25.34
$\epsilon_{\text{2nd,surr}}^{\text{med}}$	4.3e−2	6.4e−2	5.1e−2	7.1e−2
α^{med}	0.9983	0.9966	0.9943	0.9321
$f_{\text{eff}}^{\text{med}}$	3.90	8.26	3.91	2.22

Table 6: Performance measures for partonic channels contributing to $t\bar{t}$+3 jets production at the LHC.

SciPost Physics Submission

MCNET-21-13

Accelerating Monte Carlo event generation – rejection sampling using neural network event-weight estimates

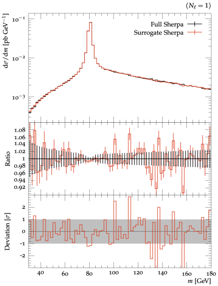
K. Danziger[1], T. Janßen[2], S. Schumann[2], F. Siegert[1]

1 Institut für Kern- und Teilchenphysik, TU Dresden, Dresden, Germany
2 Institut für Theoretische Physik, Georg-August-Universität Göttingen, Göttingen, Germany

September 27, 2021

Abstract

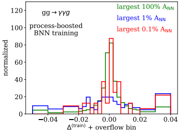
The generation of unit-weight events for complex scattering processes presents a severe challenge to modern Monte Carlo event generators. Even when using sophisticated phase-space sampling techniques adapted to the underlying transition matrix elements, the efficiency for generating unit-weight events from weighted samples can become a limiting factor in practical applications. Here we present a novel two-staged unweighting procedure that makes use of a neural-network surrogate for the full event weight. The algorithm can significantly accelerate the unweighting process, while it still guarantees unbiased sampling from the correct target distribution. We apply, validate and benchmark the new approach in high-multiplicity LHC production processes, including Z/W+4 jets and $t\bar{t}$+3 jets, where we find speed-up factors up to ten.

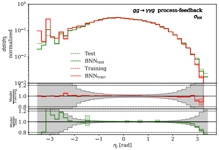


Speeding up amplitudes [precision regression]

- loop-amplitudes expensive
- interpolation standard

→ Precision NN-amplitudes





Prepared for submission to JHEP IPPP/20/116

Optimising simulations for diphoton production at hadron colliders using amplitude neural networks

Joseph Aylett-Bullock[a,b] Simon Badger[c] Ryan Moodie[a]

[a] Institute for Particle Physics Phenomenology, Department of Physics, Durham University, Durham, DH1 3LE, United Kingdom
[b] Institute for Data Science, Durham University, Durham, DH1 3LE, United Kingdom
[c] Dipartimento di Fisica and Arnold-Regge Center, Università di Torino, and INFN, Sezione di Torino, Via P. Giuria 1, I-10125 Torino, Italy

E-mail: j.p.bullock@durham.ac.uk, simondavid.badger@unito.it, ryan.i.moodie@durham.ac.uk

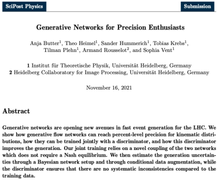
Abstract: Machine learning technology has the potential to dramatically optimise event generation and simulations. We continue to investigate the use of neural networks to approximate matrix elements for high-multiplicity scattering processes. We focus on the case of loop-induced diphoton production through gluon fusion, and develop a realistic simulation method that can be applied to hadron collider observables. Neural networks are trained using the one-loop amplitudes implemented in the NJet C++ library, and interfaced to the Sherpa Monte Carlo event generator, where we perform a detailed study for 2 → 3 and 2 → 4 scattering problems. We also consider how the trained networks perform when varying the kinematic cuts effecting the phase space and the reliability of the neural network simulations.

Invertible event generation

Precision NN-generators [Bayesian generative models]

- control through discriminator [GAN-like]
- uncertainties through Bayesian networks

→ Flow, diffusion, transformer


SciPost Physics — Submission

Generative Networks for Precision Enthusiasts

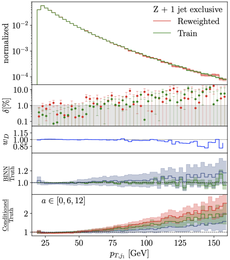
Anja Butter[1], Theo Heimel[1], Sander Hummerich[1], Tobias Krebs[1], Tilman Plehn[1], Armand Rousselot[2], and Sophia Vent[1]

1 Institut für Theoretische Physik, Universität Heidelberg, Germany
2 Heidelberg Collaboratory for Image Processing, Universität Heidelberg, Germany

November 16, 2021

Abstract

Generative networks are opening new avenues in fast event generation for the LHC. We show how generative flow networks can reach percent-level precision for kinematic distributions, how they can be trained jointly with a discriminator, and how this discriminator improves the generation. Our joint training relies on a novel coupling of the two networks which does not require a Nash equilibrium. We then estimate the generation uncertainties through a Bayesian network setup and through conditional data augmentation, while the discriminator ensures that there are no systematic inconsistencies compared to the training data.

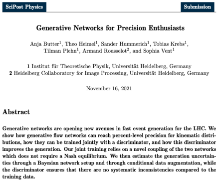


Invertible event generation

Precision NN-generators [Bayesian generative models]

- control through discriminator [GAN-like]
- uncertainties through Bayesian networks

→ Flow, diffusion, transformer


SciPost Physics — Submission

Generative Networks for Precision Enthusiasts

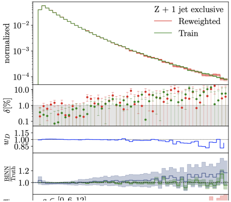
Anja Butter[1], Theo Heimel[1], Sander Hummerich[1], Tobias Krebs[1], Tilman Plehn[1], Armand Rousselot[2], and Sophia Vent[1]

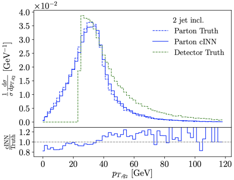
1 Institut für Theoretische Physik, Universität Heidelberg, Germany
2 Heidelberg Collaboration for Image Processing, Universität Heidelberg, Germany

November 16, 2021

Abstract

Generative networks are opening new avenues in fast event generation for the LHC. We show how generative flow networks can reach percent-level precision for kinematic distributions, how they can be trained jointly with a discriminator, and how this discriminator improves the generation. Our joint training relies on a novel coupling of the two networks which does not require a Nash equilibrium. We then estimate the generation uncertainties through a Bayesian network setup and through conditional data augmentation, while the discriminator ensures that there are no systematic inconsistencies compared to the training data.

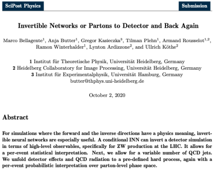




Unfolding and inversion [conditional normalizing flows]

- detector/decays/QCD unfolded
- calibrated inverse sampling

→ Publishing analysis results


SciPost Physics — Submission

Invertible Networks or Partons to Detector and Back Again

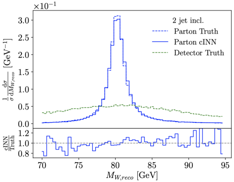
Marco Bellagente[1], Anja Butter[1], Gregor Kasieczka[3], Tilman Plehn[1], Armand Rousselot[1,2], Ramon Winterhalder[1], Lynton Ardizzone[2], and Ullrich Köthe[2]

1 Institut für Theoretische Physik, Universität Heidelberg, Germany
2 Heidelberg Collaboration for Image Processing, Universität Heidelberg, Germany
3 Institut für Experimentalphysik, Universität Hamburg, Germany
butter@thphys.uni-heidelberg.de

October 2, 2020

Abstract

For simulations where the forward and the inverse directions have a physics meaning, invertible neural networks are especially useful. A conditional INN can invert a detector simulation in terms of high-level observables, specifically for ZW production at the LHC. It allows for a per-event statistical interpretation. Next, we allow for a variable number of QCD jets. We unfold detector effects and QCD radiation to a pre-defined hard process, again with a per-event probabilistic interpretation over parton-level phase space.



Proper theory

Navigating string landscape [reinforcement learning]

- searching for viable vacua
- high dimensions, unknown global structure

→ Model space sampling

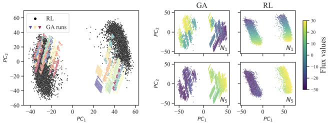


Figure 1: *Left:* Cluster structure in dimensionally reduced flux samples for RL and 25 GA runs (PCA on all samples of GA and RL). The colors indicate individual GA runs. *Right:* Dependence on flux (input) values (N_3 and N_5 respectively) in relation to principal components for a PCA fit of the individual output of GA and RL.

Probing the Structure of String Theory Vacua with Genetic Algorithms and Reinforcement Learning

Alex Cole
University of Amsterdam
a.e.cole@uva.nl

Sven Krippendorf
Arnold Sommerfeld Center for Theoretical Physics
LMU Munich
sven.krippendorf@physik.uni-muenchen.de

Andreas Schachner
Centre for Mathematical Sciences
University of Cambridge
as2673@cam.ac.uk

Gary Shiu
University of Wisconsin-Madison
shiu@physics.wisc.edu

Abstract

Identifying string theory vacua with desired physical properties at low energies requires searching through high-dimensional solution spaces – collectively referred to as the string landscape. We highlight that this search problem is amenable to reinforcement learning and genetic algorithms. In the context of flux vacua, we are able to reveal novel features (suggesting previously unidentified symmetries) in the string theory solutions required for properties such as the string coupling. In order to identify these features robustly, we combine results from both search methods, which we argue is imperative for reducing sampling bias.

Proper theory

Navigating string landscape [reinforcement learning]

- searching for viable vacua
- high dimensions, unknown global structure

→ Model space sampling

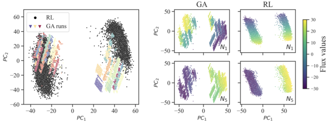


Figure 1: *Left:* Cluster structure in dimensionally reduced flux samples for RL and 25 GA runs (PCA on all samples of GA and RL). The colors indicate individual GA runs. *Right:* Dependence on flux (input) values (N_3 and N_5 respectively) in relation to principal components for a PCA fit of the individual output of GA and RL.

Probing the Structure of String Theory Vacua with Genetic Algorithms and Reinforcement Learning

Alex Cole
University of Amsterdam
a.e.cole@uva.nl

Sven Krippendorf
Arnold Sommerfeld Center for Theoretical Physics
LMU Munich
sven.krippendorf@physik.uni-muenchen.de

Andreas Schachner
Centre for Mathematical Sciences
University of Cambridge
as2673@cam.ac.uk

Gary Shiu
University of Wisconsin-Madison
shiu@physics.wisc.edu

Abstract

Identifying string theory vacua with desired physical properties at low energies requires searching through high-dimensional solution spaces – collectively referred to as the string landscape. We highlight that this search problem is amenable to reinforcement learning and genetic algorithms. In the context of flux vacua, we are able to reveal novel features (suggesting previously unidentified symmetries) in the string theory solutions required for properties such as the string coupling. In order to identify these features robustly, we combine results from both search methods, which we argue is imperative for reducing sampling bias.

Learning formulas [genetic algorithm, symbolic regression]

- approximate numerical function through formula
- example: score/optimal observables

→ PySR

compl	dof	function	MSE
3	1	$a\,\Delta\phi$	$1.30 \cdot 10^{-1}$
4	1	$\sin(a\Delta\phi)$	$2.75 \cdot 10^{-1}$
5	1	$a\Delta\phi x_{p,1}$	$9.93 \cdot 10^{-2}$
6	1	$-x_{p,1}\sin(\Delta\phi + a)$	$1.90 \cdot 10^{-1}$
7	1	$(-x_{p,1} - a)\sin(\sin(\Delta\phi))$	$5.63 \cdot 10^{-2}$
8	1	$(a - x_{p,1})x_{p,2}\sin(\Delta\phi)$	$1.61 \cdot 10^{-2}$
14	2	$x_{p,1}(a\Delta\phi - \sin(\sin(\Delta\phi)))(x_{p,2} + b)$	$1.44 \cdot 10^{-2}$
15	3	$-(x_{p,2}(a\Delta\eta^2 + x_{p,1}) + b)\sin(\Delta\phi + c)$	$1.30 \cdot 10^{-2}$
16	4	$-x_{p,1}(a - b\Delta\eta)(x_{p,2} + c)\sin(\Delta\phi + d)$	$8.50 \cdot 10^{-3}$
28	7	$(x_{p,2} + a)(bx_{p,1}(c - \Delta\phi) - x_{p,1}(d\Delta\eta + ex_{p,2} + f)\sin(\Delta\phi + g))$	$8.18 \cdot 10^{-3}$

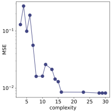


Table 8: Score hall of fame for simplified WBF Higgs production with $f_{\widetilde{WW}} = 0$, including a optimization fit.

SciPost Physics Submission

Back to the Formula — LHC Edition

Anja Butter[1], Tilman Plehn[1], Nathalie Soybelman[1], and Johann Brehmer[2]

1 Institut für Theoretische Physik, Universität Heidelberg, Germany
2 Center for Data Science, New York University, New York, United States
nathalie@soybelman.de

November 16, 2021

Abstract

While neural networks offer an attractive way to numerically encode functions, actual formulas remain the language of theoretical particle physics. We use symbolic regression trained on matrix-element information to extract, for instance, optimal LHC observables. This way we invert the usual simulation paradigm and extract easily interpretable formulas from complex simulated data. We introduce the method using the effect of a dimension-6 coefficient on associated ZH production. We then validate it for the known case of CP-violation in weak-boson-fusion Higgs production, including detector effects.

Generative-network revolution

Generative networks

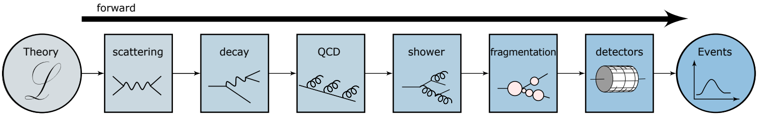
- generate new images, text blocks, LHC events
- encode density in target space
  sample from Gaussian into target space
- reproduce training data, statistically independently
- include uncertainty on estimated density [Bayesian NN]

Generative-network revolution

Generative networks

- generate new images, text blocks, LHC events
- encode density in target space
  sample from Gaussian into target space
- reproduce training data, statistically independently
- include uncertainty on estimated density [Bayesian NN]
- Variational Autoencoder
  $\to$ low-dimensional physics, high-dimensional representation
- Generative Adversarial Network
  $\to$ generator trained by discriminator
- Normalizing Flow/Diffusion Model
  $\to$ stable (bijective) mapping
- Generative Transformer
  $\to$ learning correlations successively

$\to$ Pick model for purpose



Generative-network revolution

Generative networks

- generate new images, text blocks, LHC events
- encode density in target space
  sample from Gaussian into target space
- reproduce training data, statistically independently
- include uncertainty on estimated density [Bayesian NN]
- Variational Autoencoder
  $\rightarrow$ low-dimensional physics, high-dimensional representation
- Generative Adversarial Network
  $\rightarrow$ generator trained by discriminator
- Normalizing Flow/Diffusion Model
  $\rightarrow$ stable (bijective) mapping
- Generative Transformer
  $\rightarrow$ learning correlations successively

$\rightarrow$ Pick model for purpose

Fundamental question: GANplification

- first generated instances reproducing structures
- too many generated instances reproducing noise?

Precision generator

Phase-space generators [typical LHC task]

- training from event samples
  no energy-momentum conservation
- every correlation counts
- $Z_{\mu\mu} + \{1, 2, 3\}$ jets [Z-peak, variable jet number, jet-jet topology]

Precision generator

Phase-space generators [typical LHC task]

- training from event samples
  no energy-momentum conservation
- every correlation counts
- $Z_{\mu\mu} + \{1, 2, 3\}$ jets [Z-peak, variable jet number, jet-jet topology]

INN-generator

- stable bijective mapping

$$\text{latent } r \sim p_\text{latent} \quad \underset{\leftarrow\ \overline{G}_\theta(x)}{\overset{G_\theta(r)\rightarrow}{\longleftrightarrow}} \quad \text{phase space } x \sim p_\text{data}$$

- tractable Jacobian

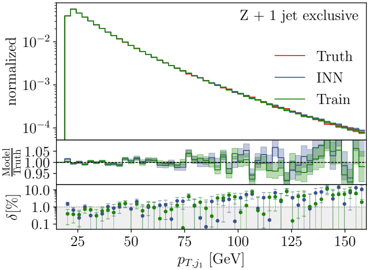
$$dx\ p_\text{model}(x) = dr\ p_\text{latent}(r)$$

$$p_\text{model}(x) = p_\text{latent}\left(\overline{G}_\theta(x)\right) \left| \frac{\partial \overline{G}_\theta(x)}{\partial x} \right|$$

- likelihood loss

$$\mathcal{L}_\text{INN} = -\Big\langle \log p_\text{model}(x) \Big\rangle_{p_\text{data}}$$

⇒ Per-cent precision possible



Controlled precision generator

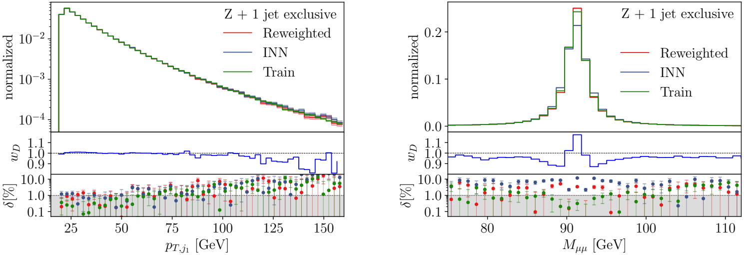
Best of GANs: discriminator

- $D = 0$ (generator) vs $D = 1$ (training)
- NP-optimal discriminator

$$D(x) \to \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_{\text{model}}(x)} \to \frac{1}{2}$$

- learned event weight $w(x) \to \dfrac{D(x)}{1 - D(x)} = \dfrac{p_{\text{data}}(x)}{p_{\text{model}}(x)} \to 1$

$\Rightarrow$ Dual purpose: control and reweight



Controlled precision generator

Best of GANs: discriminator

- $D = 0$ (generator) vs $D = 1$ (training)
- NP-optimal discriminator

$$D(x) \to \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_{\text{model}}(x)} \to \frac{1}{2}$$

- learned event weight

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

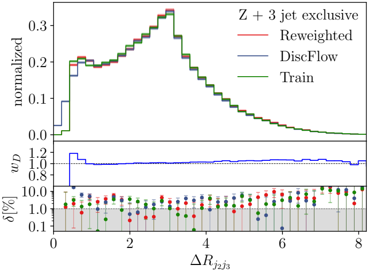
$\Rightarrow$ Dual purpose: control and reweight

Joint training [GAN inspiration]

- GAN-like training unstable [Nash equilibrium??]
- coupling through weights

$$\mathcal{L} = -\int dx \, \frac{p_{\text{data}}^{\alpha+1}(x)}{p_{\text{model}}^{\alpha}(x)} \, \log \frac{p_{\text{model}}(x)}{p_{\text{data}}(x)}$$

$\Rightarrow$ Unweighted, controlled events

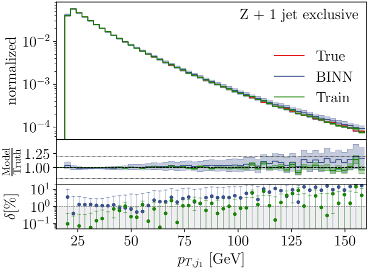


Precision generator with uncertainties

Bayesian network generator

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

⇒ Training-related error bars



Precision generator with uncertainties

Bayesian network generator

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

⇒ Training-related error bars

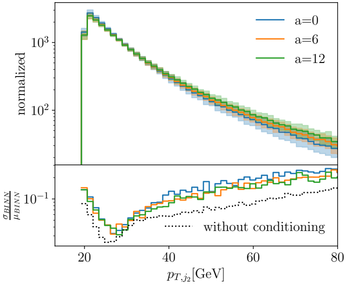
Theory uncertainties

- 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

⇒ Systematic/theory error bars

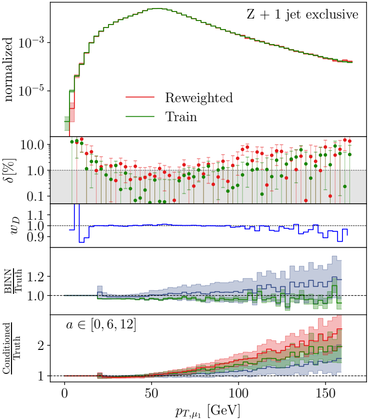


Precision generator with uncertainties

Bayesian network generator

- network with weight distributions [Gal (2016)]
  sample weights [defining error bar]
  working for regression, classification
  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

Testing generative networks

Compare network to training/test data

- supervised: histogram deviation [or pull]
- unsupervised density → histogram discriminator

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

→ Using interpretable phase space

Testing generative networks

Compare network to training/test data

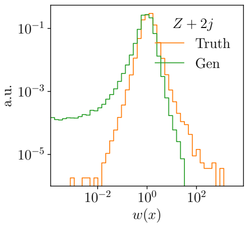
- supervised: histogram deviation [or pull]
- unsupervised density → histogram discriminator

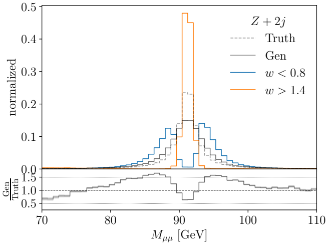
$$w(x_i) = \frac{D(x_i)}{1 - D(x_i)} = \frac{p_{\text{data}}(x_i)}{p_{\text{model}}(x_i)}$$

→ Using interpretable phase space

Applied to event generators [also jets, calorimeter showers]

- shape and width of w-histogram
- pattern in (interpretable) phase space?





Testing generative networks

Compare network to training/test data

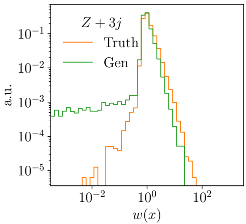
- supervised: histogram deviation [or pull]
- unsupervised density $\rightarrow$ histogram discriminator

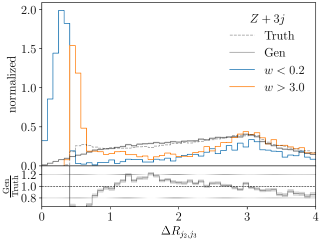
$$w(x_i) = \frac{D(x_i)}{1 - D(x_i)} = \frac{p_{\text{data}}(x_i)}{p_{\text{model}}(x_i)}$$

$\rightarrow$ Using interpretable phase space

Applied to event generators [also jets, calorimeter showers]

- shape and width of w-histogram
- pattern in (interpretable) phase space?



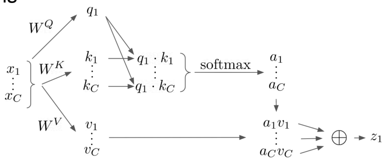


$\rightarrow$ Generative xAI for LHC physicists

JetGPT

Correlations through self-attention

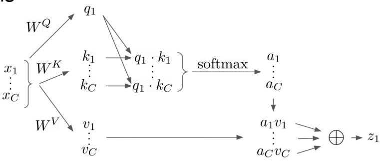
- think of data as bins in phase-space directions
  self-attention: encode relation between bins
  input x, learn relation $x_i \leftrightarrow x_j$
- latent query representation $q = W^Q x$
  latent key representation $k = W^K x$
  define correlation as $A_{ij} = q_i \cdot k_j$
- latent value representation $v = W^V x$
  output $z = A\, v$



JetGPT

Correlations through self-attention

- think of data as bins in phase-space directions
  self-attention: encode relation between bins
  input x, learn relation $x_i \leftrightarrow x_j$
- latent query representation $q = W^Q x$
  latent key representation $k = W^K x$
  define correlation as $A_{ij} = q_i \cdot k_j$
- latent value representation $v = W^V x$
  output $z = A\, v$



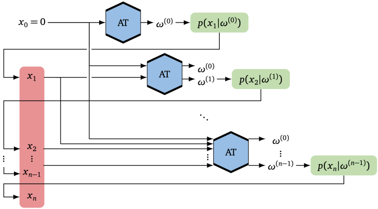
Autoregressive transformer

- factorized density

$$p_{\text{model}}(x|\theta) = \prod_i p(x_i|x_1, \dots, x_{i-1})$$

- bins $\to$ Gaussian mixture model
- autoregressive $A_{ij} = 0$ for $j > i$

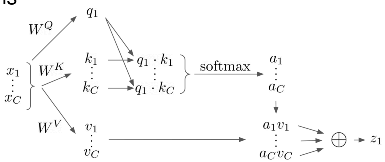
$\to$ Bayesian version for uncertainties



JetGPT

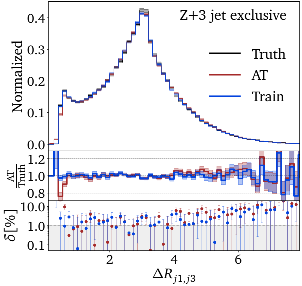
Correlations through self-attention

- think of data as bins in phase-space directions
  self-attention: encode relation between bins
  input x, learn relation $x_i \leftrightarrow x_j$
- latent query representation $q = W^Q x$
  latent key representation $k = W^K x$
  define correlation as $A_{ij} = q_i \cdot k_j$
- latent value representation $v = W^V x$
  output $z = A\, v$



Bayesian JetGPT

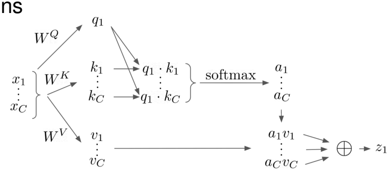
- sometimes you win...



JetGPT

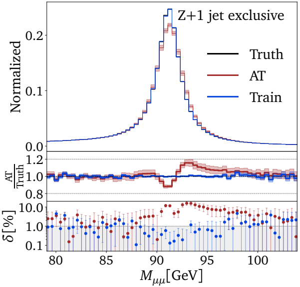
Correlations through self-attention

- think of data as bins in phase-space directions
  self-attention: encode relation between bins
  input x, learn relation $x_i \leftrightarrow x_j$
- latent query representation $q = W^Q x$
  latent key representation $k = W^K x$
  define correlation as $A_{ij} = q_i \cdot k_j$
- latent value representation $v = W^V x$
  output $z = A\, v$



Bayesian JetGPT

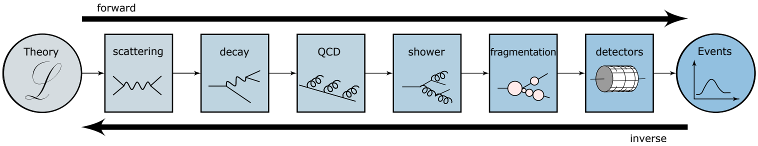
- sometimes you win...
  ...and sometimes there is work to do...



Inverse simulation

Invertible ML-simulation

- forward: $r \to$ events trained on model
- inverse: $r \to$ anything trained on model, conditioned on event



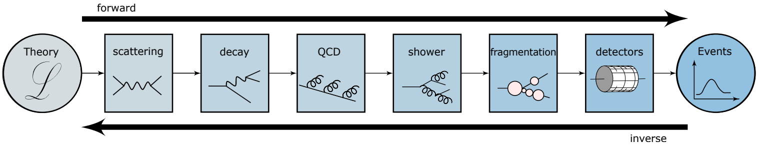
Inverse simulation

Invertible ML-simulation

- forward: $r \to$ events trained on model
- inverse: $r \to$ anything trained on model, conditioned on event
- individual steps known problems

  detector unfolding
  unfolding to QCD parton means jet algorithm
  unfolding jet radiation known combinatorics problem
  unfolding to hard process standard in top groups [needed for global analyses]
  matrix element method an old dream
- improved through coherent ML-method
- free choice of data-theory inference point

$\to$ Transformative progress for HL-LHC



Inverting to QCD

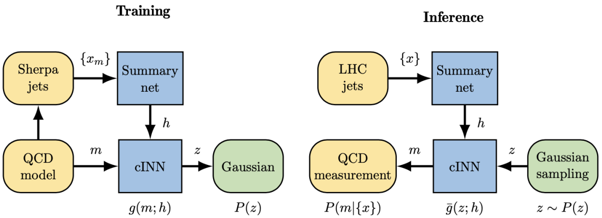
cINN for inference [Bieringer, Butter, Heimel, Höche, Köthe, TP, Radev]

- condition jets with QCD parameters
  train model parameters $\to$ Gaussian latent space
  test Gaussian sampling $\to$ parameter measurement
- beyond C_A vs C_F

$$P_{qq} = C_F \left[D_{qq} \frac{2z(1-y)}{1-z(1-y)} + F_{qq}(1-z) + C_{qq} yz(1-z) \right]$$

$$P_{gg} = 2C_A \left[D_{gg} \left(\frac{z(1-y)}{1-z(1-y)} + \frac{(1-z)(1-y)}{1-(1-z)(1-y)} \right) + F_{gg} z(1-z) + C_{gg} yz(1-z) \right]$$

$$P_{gq} = T_R \left[F_{qq} \left(z^2 + (1-z)^2 \right) + C_{gq} yz(1-z) \right]$$



Inverting to QCD

cINN for inference [Bieringer, Butter, Heimel, Höche, Köthe, TP, Radev]

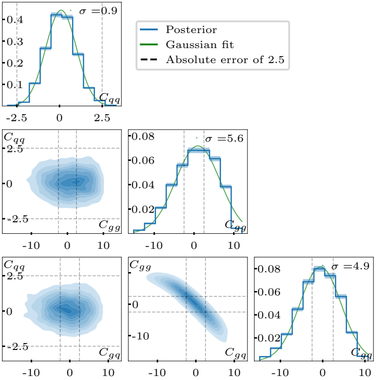
- condition — jets with QCD parameters
  train — model parameters → Gaussian latent space
  test — Gaussian sampling → parameter measurement
- beyond C_A vs C_F

$$P_{qq} = C_F \left[D_{qq} \frac{2z(1-y)}{1-z(1-y)} + F_{qq}(1-z) + C_{qq} yz(1-z) \right]$$

$$P_{gg} = 2C_A \left[D_{gg} \left(\frac{z(1-y)}{1-z(1-y)} + \frac{(1-z)(1-y)}{1-(1-z)(1-y)} \right) + F_{gg} z(1-z) + C_{gg} yz(1-z) \right]$$

$$P_{gq} = T_R \left[F_{qq} \left(z^2 + (1-z)^2 \right) + C_{gq} yz(1-z) \right]$$

- idealized shower [Sherpa]
- More ML-opportunities...



ML for LHC Theory

ML-applications

- just another numerical tool for a numerical field
- driven by money from data science and medical research
- goals are...

  ...improve established tasks
  ...develop new tools for established tasks
  ...transform through new ideas
- xAI through...

  ...precision control
  ...uncertainties
  ...symmetries
  ...formulas

→ Lots of fun with hard LHC problems

Modern Machine Learning for LHC Physicists

Tilman Plehn[a*], Anja Butter[a,b], Barry Dillon[a], Claudius Krause[a,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] NHETC, Dept. of Physics and Astronomy, Rutgers University, Piscataway, USA
[d] CP3, Université Catholique de Louvain, Louvain-la-Neuve, Belgium

July 21, 2023

Abstract

Modern machine learning is transforming particle physics, faster than we can follow, and 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 are meant to lead students with basic knowledge of particle physics and significant enthusiasm for machine learning to relevant applications as fast as possible. 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 reflecting the problem at hand 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. Given that these notes will be outdated already at the time of submission, the week of ML4Jets 2022, they will be updated frequently.

Learning optimal observables

Measure model parameter θ optimally [Butter, TP, Soybelman, Brehmer]

- single-event likelihood

$$p(x|\theta) = \frac{1}{\sigma_{\text{tot}}(\theta)} \frac{d^m\sigma(x|\theta)}{dx^m}$$

- expanded in θ around θ_0, define score

$$\log \frac{p(x|\theta)}{p(x|\theta_0)} \approx (\theta - \theta_0) \, \nabla_\theta \log p(x|\theta)\Big|_{\theta_0} \equiv (\theta - \theta_0) \; t(x|\theta_0) \equiv (\theta - \theta_0) \; \mathcal{O}^{\text{opt}}(x)$$

- to leading order at parton level

$$p(x|\theta) \approx |\mathcal{M}|_0^2 + \theta |\mathcal{M}|_{\text{int}}^2 \qquad \Rightarrow \qquad t(x|\theta_0) \sim \frac{|\mathcal{M}|_{\text{int}}^2}{|\mathcal{M}|_0^2}$$

$\Rightarrow$ And including everything?

Learning optimal observables

Measure model parameter θ optimally [Butter, TP, Soybelman, Brehmer]

- single-event likelihood

$$p(x|\theta) = \frac{1}{\sigma_{\text{tot}}(\theta)} \frac{d^m\sigma(x|\theta)}{dx^m}$$

- expanded in θ around θ_0, define score

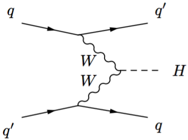
$$\log \frac{p(x|\theta)}{p(x|\theta_0)} \approx (\theta - \theta_0) \, \nabla_\theta \log p(x|\theta)\Big|_{\theta_0} \equiv (\theta - \theta_0) \, t(x|\theta_0) \equiv (\theta - \theta_0) \, \mathcal{O}^{\text{opt}}(x)$$

- to leading order at parton level

$$p(x|\theta) \approx |\mathcal{M}|_0^2 + \theta |\mathcal{M}|_{\text{int}}^2 \qquad \Rightarrow \qquad t(x|\theta_0) \sim \frac{|\mathcal{M}|_{\text{int}}^2}{|\mathcal{M}|_0^2}$$

$\Rightarrow$ And including everything?

CP-violating Higgs production



- unique CP-observable

$$t \propto \epsilon_{\mu\nu\rho\sigma} \, k_1^\mu \, k_2^\nu \, q_1^\rho \, q_2^\sigma \; \text{sign}\left[(k_1 - k_2) \cdot (q_1 - q_2)\right] \xrightarrow{\text{lab frame}} \sin \Delta\phi_{jj}$$

- CP-effect in $\Delta\phi_{jj}$
  D6-effect in $p_{T,j}$

$\Rightarrow$ Established LHC task

Symbolic regression

Symbolic regression of score [PySR (M Cranmer) + final fit]

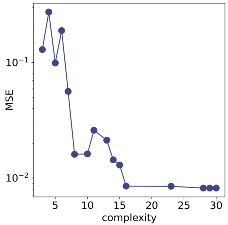
- function to approximate $t(x|\theta)$
- phase space parameters $x_p = p_T/m_H, \Delta\eta, \Delta\phi$ [node]
- operators $\sin x, x^2, x^3, x+y, x-y, x*y, x/y$ [node]
- represent formula as tree [complexity = number of nodes]

⇒ Figures of merit

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}\Big[g_i(x) - t(x, z|\theta)\Big]^2 \to \text{MSE} + \text{parsimony} \cdot \text{complexity}$$

Score around Standard Model

compl	dof	function	MSE
3	1	$a\,\Delta\phi$	$1.30 \cdot 10^{-1}$
4	1	$\sin(a\Delta\phi)$	$2.75 \cdot 10^{-1}$
5	1	$a\Delta\phi x_{p,1}$	$9.93 \cdot 10^{-2}$
6	1	$-x_{p,1}\sin(\Delta\phi + a)$	$1.90 \cdot 10^{-1}$
7	1	$(-x_{p,1} - a)\sin(\sin(\Delta\phi))$	$5.63 \cdot 10^{-2}$
8	1	$(a - x_{p,1})x_{p,2}\sin(\Delta\phi)$	$1.61 \cdot 10^{-2}$
14	2	$x_{p,1}(a\Delta\phi - \sin(\sin(\Delta\phi)))(x_{p,2} + b)$	$1.44 \cdot 10^{-2}$
15	3	$-(x_{p,2}(a\Delta\eta^2 + x_{p,1}) + b)\sin(\Delta\phi + c)$	$1.30 \cdot 10^{-2}$
16	4	$-x_{p,1}(a - b\Delta\eta)(x_{p,2} + c)\sin(\Delta\phi + d)$	$8.50 \cdot 10^{-3}$
28	7	$(x_{p,2} + a)(bx_{p,1}(c - \Delta\phi) - x_{p,1}(d\Delta\eta + ex_{p,2} + f)\sin(\Delta\phi + g))$	$8.18 \cdot 10^{-3}$



Symbolic regression

Symbolic regression of score [PySR (M Cranmer) + final fit]

- function to approximate $t(x|\theta)$
- phase space parameters $x_p = p_T/m_H, \Delta\eta, \Delta\phi$ [node]
- operators $\sin x, x^2, x^3, x+y, x-y, x*y, x/y$ [node]
- represent formula as tree [complexity = number of nodes]

⇒ Figures of merit

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}\Big[g_i(x) - t(x,z|\theta)\Big]^2 \to \text{MSE} + \text{parsimony} \cdot \text{complexity}$$

Score around Standard Model

- expected limits:

  very wrong formula
  wrong formula
- same within statistical limitation:

  right formula
  MadMiner

⇒ Formulas to numerics and back