Modern Machine Learning for Jets

Tilman Plehn

Universität Heidelberg

JetMET, Brussels, May 2023

LHC physics vs data scientist

LHC questions

- How to trigger from 3 PB/s to 300 MB/s?

LHC physics vs data scientist

LHC questions

- How to trigger from 3 PB/s to 300 MB/s?

  Data compression [Netflix]

LHC physics vs data scientist

LHC questions

- How to trigger from 3 PB/s to 300 MB/s?

  Data compression [Netflix]
- How to analyze ntuples?

LHC physics vs data scientist

LHC questions

- How to trigger from 3 PB/s to 300 MB/s?

  Data compression [Netflix]
- How to analyze ntuples?

  Graph neural networks [Car cameras]

LHC physics vs data scientist

LHC questions

- How to trigger from 3 PB/s to 300 MB/s?

  Data compression [Netflix]

- How to analyze ntuples?

  Graph neural networks [Car cameras]

- How to incorporate symmetries?

LHC physics vs data scientist

LHC questions

- How to trigger from 3 PB/s to 300 MB/s?

  Data compression [Netflix]
- How to analyze ntuples?

  Graph neural networks [Car cameras]
- How to incorporate symmetries?

  Contrastive learning [Google]

LHC physics vs data scientist

LHC questions

- How to trigger from 3 PB/s to 300 MB/s?

  Data compression [Netflix]
- How to analyze ntuples?

  Graph neural networks [Car cameras]
- How to incorporate symmetries?

  Contrastive learning [Google]
- How to combine tracker and calorimeter?

LHC physics vs data scientist

LHC questions

- How to trigger from 3 PB/s to 300 MB/s?
  Data compression [Netflix]
- How to analyze ntuples?
  Graph neural networks [Car cameras]
- How to incorporate symmetries?
  Contrastive learning [Google]
- How to combine tracker and calorimeter?
  Super-resolution [Gaming]

LHC physics vs data scientist

LHC questions

- How to trigger from 3 PB/s to 300 MB/s?
  Data compression [Netflix]
- How to analyze ntuples?
  Graph neural networks [Car cameras]
- How to incorporate symmetries?
  Contrastive learning [Google]
- How to combine tracker and calorimeter?
  Super-resolution [Gaming]
- How to remove pile-up?

LHC physics vs data scientist

LHC questions

- How to trigger from 3 PB/s to 300 MB/s?
  Data compression [Netflix]
- How to analyze ntuples?
  Graph neural networks [Car cameras]
- How to incorporate symmetries?
  Contrastive learning [Google]
- How to combine tracker and calorimeter?
  Super-resolution [Gaming]
- How to remove pile-up?
  Data denoising [Cars]

LHC physics vs data scientist

LHC questions

- How to trigger from 3 PB/s to 300 MB/s?
  Data compression [Netflix]
- How to analyze ntuples?
  Graph neural networks [Car cameras]
- How to incorporate symmetries?
  Contrastive learning [Google]
- How to combine tracker and calorimeter?
  Super-resolution [Gaming]
- How to remove pile-up?
  Data denoising [Cars]
- How to look for BSM physics?

LHC physics vs data scientist

LHC questions

- How to trigger from 3 PB/s to 300 MB/s?
  Data compression [Netflix]
- How to analyze ntuples?
  Graph neural networks [Car cameras]
- How to incorporate symmetries?
  Contrastive learning [Google]
- How to combine tracker and calorimeter?
  Super-resolution [Gaming]
- How to remove pile-up?
  Data denoising [Cars]
- How to look for BSM physics?
  Autoencoders [SAP]

LHC physics vs data scientist

LHC questions

- How to trigger from 3 PB/s to 300 MB/s?
  Data compression [Netflix]
- How to analyze ntuples?
  Graph neural networks [Car cameras]
- How to incorporate symmetries?
  Contrastive learning [Google]
- How to combine tracker and calorimeter?
  Super-resolution [Gaming]
- How to remove pile-up?
  Data denoising [Cars]
- How to look for BSM physics?
  Autoencoders [SAP]
- How to generate jets/events?

LHC physics vs data scientist

LHC questions

- How to trigger from 3 PB/s to 300 MB/s?
  Data compression [Netflix]
- How to analyze ntuples?
  Graph neural networks [Car cameras]
- How to incorporate symmetries?
  Contrastive learning [Google]
- How to combine tracker and calorimeter?
  Super-resolution [Gaming]
- How to remove pile-up?
  Data denoising [Cars]
- How to look for BSM physics?
  Autoencoders [SAP]
- How to generate jets/events?
  Generative transformer [ChatGPT]

LHC physics vs data scientist

LHC questions

- How to trigger from 3 PB/s to 300 MB/s?

  Data compression [Netflix]
- How to analyze ntuples?

  Graph neural networks [Car cameras]
- How to incorporate symmetries?

  Contrastive learning [Google]
- How to combine tracker and calorimeter?

  Super-resolution [Gaming]
- How to remove pile-up?

  Data denoising [Cars]
- How to look for BSM physics?

  Autoencoders [SAP]
- How to generate jets/events?

  Generative transformer [ChatGPT]
- How to analyse LHC data?

LHC physics vs data scientist

LHC questions

- How to trigger from 3 PB/s to 300 MB/s?
  Data compression [Netflix]
- How to analyze ntuples?
  Graph neural networks [Car cameras]
- How to incorporate symmetries?
  Contrastive learning [Google]
- How to combine tracker and calorimeter?
  Super-resolution [Gaming]
- How to remove pile-up?
  Data denoising [Cars]
- How to look for BSM physics?
  Autoencoders [SAP]
- How to generate jets/events?
  Generative transformer [ChatGPT]
- How to analyse LHC data?
  Simulation-based inference [LHC leading?]

LHC physics vs data scientist

LHC questions

- How to trigger from 3 PB/s to 300 MB/s?
  Data compression [Netflix]
- How to analyze ntuples?
  Graph neural networks [Car cameras]
- How to incorporate symmetries?
  Contrastive learning [Google]
- How to combine tracker and calorimeter?
  Super-resolution [Gaming]
- How to remove pile-up?
  Data denoising [Cars]
- How to look for BSM physics?
  Autoencoders [SAP]
- How to generate jets/events?
  Generative transformer [ChatGPT]
- How to analyse LHC data?
  Simulation-based inference [LHC leading?]
- But how about uncertatinties?

Shortest ML-intro ever

Fit-like approximation [ask NNPDF]

- approximate known $f(x)$ using $f_\theta(x)$
- no parametrization, just very many values θ
- new representation/latent space θ

Construction and contol

- minimize loss to find best θ
- typically, likelihood generalizing fit χ^2
- compare $x \to f_\theta(x)$ for training/test data

LHC applications

- regression $x \to f_\theta(x)$
- classification $x \to f_\theta(x) \in [0, 1]$
- generation $r \sim \mathcal{N} \to f_\theta(r)$
- conditional generation $r \sim \mathcal{N} \to f_\theta(r|x)$
- ...

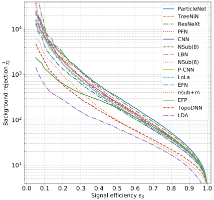
$\to$ Transforming numerical science

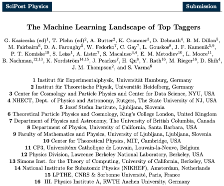
Analysis

Jet tagging [supervised classification]

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

$\to$ ParticleNet & Co established





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]

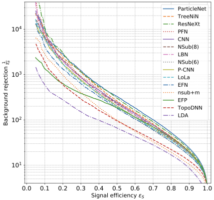
1 Institut für Experimentalphysik, Universität Hamburg, Germany
2 Institut für Theoretische Physik, Universität Heidelberg, Germany
3 Center for Cosmology and Particle Physics and Center for Data Science, NYU, USA
4 NHECT, Dept. of Physics and Astronomy, Rutgers, The State University of NJ, USA
5 Jozef 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

Analysis

Jet tagging [supervised classification]

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

→ ParticleNet & Co established



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

→ Seriously impressive

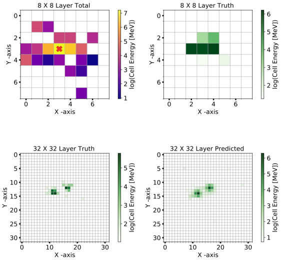


Fig. 7: An event display of total energy shower (within topocluster), as captured by a calorimeter layer of 8×8 granularity, along with the location of the track, denoted by a red cross (left) and the same shower is captured by a calorimeter layer of 32×32 granularity (middle). The bottom right panel 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]

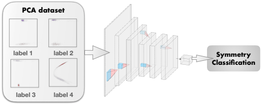
[1]Weizmann Institute of Science, Rehovot 76100, Israel
[2]CERN, CH 1211, Geneva 23, Switzerland
[3]Università di Roma Sapienza, Piazza Aldo Moro, 2, 00185 Roma, Italy e INFN, Italy
[4]Université Paris-Saclay, CNRS/IN2P3, IJCLab, 91405, Orsay, France

Symmetries

Learning symmetries [representation, visualization]

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

→ NN-identified symmetries



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.

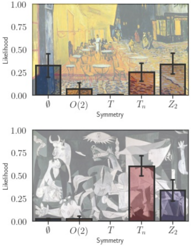
I. INTRODUCTION

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-

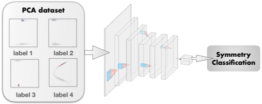


Symmetries

Learning symmetries [representation, visualization]

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

→ NN-identified symmetries



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.

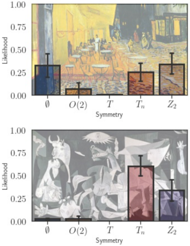
I. INTRODUCTION

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.

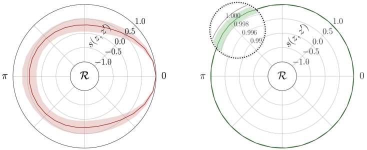
A functional task-oriented implementation of the gen-



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.

Anomalous jets and parton densities

Anomaly searches [unsupervised training]

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

$\rightarrow$ LHC spirit, more later

SciPost Physics Submission

Better Latent Spaces for Better Autoencoders

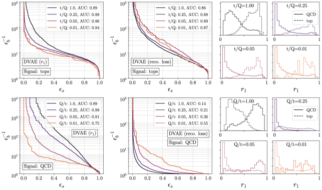
Barry M. Dillon[1], Tilman Plehn[1], Christof Sauer[2], and Peter Sorrenson[3],

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.



Anomalous jets and parton densities

Anomaly searches [unsupervised training]

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

→ LHC spirit, more later

SciPost Physics Submission

Better Latent Spaces for Better Autoencoders

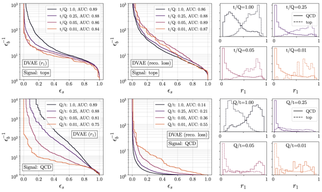
Barry M. Dillon[1], Tilman Plehn[1], Christof Sauer[2], and Peter Sorrenson[3],

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.

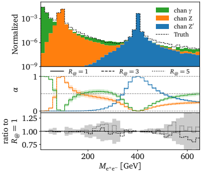
PACS. 12.38.-t Quantum chromodynamics – 12.39.-x Phenomenological quark models – 84.35.+i Neural Networks

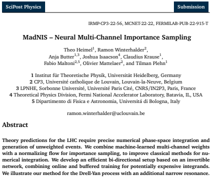
Faster event generators

Speeding up Sherpa and MadNIS [phase space sampling]

- precision simulations limiting factor for Runs 3&4
- fast and efficient sampling key

$\rightarrow$ ML-Multichannel-Vegas





SciPost Physics Submission

IRMP-CP3-22-56, MCNET-22-22, FERMILAB-PUB-22-915-T

MadNIS – Neural Multi-Channel Importance Sampling

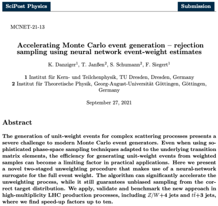
Theo Heimel[1], Ramon Winterhalder[2], Anja Butter[1,3], Joshua Isaacson[4], Claudius Krause[1], Fabio Maltoni[2,5], Olivier Mattelaer[2], and Tilman Plehn[1]

1 Institut für Theoretische Physik, Universität Heidelberg, Germany
2 CP3, Université catholique de Louvain, Louvain-la-Neuve, Belgium
3 LPNHE, Sorbonne Université, Université Paris Cité, CNRS/IN2P3, Paris, France
4 Theoretical Physics Division, Fermi National Accelerator Laboratory, Batavia, IL, USA
5 Dipartimento di Fisica e Astronomia, Universitá di Bologna, Italy

ramon.winterhalder@uclouvain.be

Abstract

Theory predictions for the LHC require precise numerical phase-space integration and generation of unweighted events. We combine machine-learned multi-channel weights with a normalizing flow for importance sampling, to improve classical methods for numerical integration. We develop an efficient bi-directional setup based on an invertible network, combining online and buffered training for potentially expensive integrands. We illustrate our method for the Drell-Yan process with an additional narrow resonance.



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

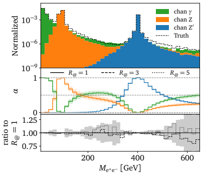
The generation of unit-weight events for complex scattering processes presents a severe challenge to modern Monte Carlo event generators. Even when using sophisticated phase-space sampling techniques adapted to the underlying transition matrix elements, the efficiency for generating unit-weight events 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.

Faster event generators

Speeding up Sherpa and MadNIS [phase space sampling]

- precision simulations limiting factor for Runs 3&4
- fast and efficient sampling key

→ ML-Multichannel-Vegas



SciPost Physics Submission

IRMP-CP3-22-56, MCNET-22-22, FERMILAB-PUB-22-915-T

MadNIS – Neural Multi-Channel Importance Sampling

Theo Heimel[1], Ramon Winterhalder[2],
Anja Butter[1,3], Joshua Isaacson[4], Claudius Krause[1],
Fabio Maltoni[2,5], Olivier Mattelaer[2], and Tilman Plehn[1]

1 Institut für Theoretische Physik, Universität Heidelberg, Germany
2 CP3, Université catholique de Louvain, Louvain-la-Neuve, Belgium
3 LPNHE, Sorbonne Université, Université Paris Cité, CNRS/IN2P3, Paris, France
4 Theoretical Physics Division, Fermi National Accelerator Laboratory, Batavia, IL, USA
5 Dipartimento di Fisica e Astronomia, Universitá di Bologna, Italy

ramon.winterhalder@uclouvain.be

Abstract

Theory predictions for the LHC require precise numerical phase-space integration and generation of unweighted events. We combine machine-learned multi-channel weights with a normalizing flow for importance sampling, to improve classical methods for numerical integration. We develop an efficient bi-directional setup based on an invertible network, combining online and buffered training for potentially expensive integrands. We illustrate our method for the Drell-Yan process with an additional narrow resonance.

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

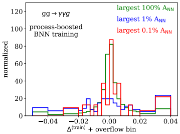
Abstract

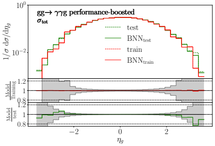
The generation of unit-weight events for complex scattering processes presents a severe challenge to modern Monte Carlo event generators. Even when using sophisticated phase-space sampling techniques adapted to the underlying transition matrix elements, the efficiency for generating unit-weight events 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 [phase space regression]

- loop-amplitudes expensive
- interpolation standard

→ Precision NN-regression, more later





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 $2 \to 3$ and $2 \to 4$ 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.

Forward and inverse simulation

Precision NN-generators [INN + Bayesian discriminator]

- control through discriminator [GAN-like]
- uncertainties through Bayesian networks
- phase space prototypical

$\rightarrow$ Precision & control

SciPost Physics Submission

Generative Networks for Precision Enthusiasts

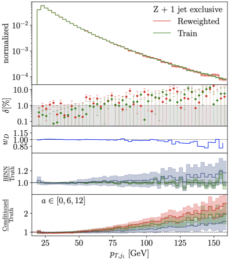
Anja Butter[1], Theo Heimel[1], Sander Hummerich[1], Tobias Krebs[1], Tilman Plehn[1], Armand Rousselot[2], and Sophia Vent[1]

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.



Forward and inverse simulation

Precision NN-generators [INN + Bayesian discriminator]

- control through discriminator [GAN-like]
- uncertainties through Bayesian networks
- phase space prototypical

→ Precision & control

SciPost Physics Submission

Generative Networks for Precision Enthusiasts

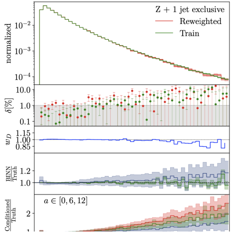
Anja Butter[1], Theo Heimel[1], Sander Hummerich[1], Tobias Krebs[1], Tilman Plehn[1], Armand Rousselot[2], and Sophia Vent[1]

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

November 16, 2021

Abstract

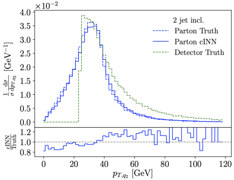
Generative networks are opening new avenues in fast event generation for the LHC. We show how generative flow networks can reach percent-level precision for kinematic distributions, how they can be trained jointly with a discriminator, and how this discriminator improves the generation. Our joint training 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.



Unfolding and inversion [conditional normalizing flows]

- shower/hadronization unfolded by jet algorithm
- detector/decays unfolded e.g. in tops
- calibrated inverse sampling

→ Inverse generation



SciPost Physics Submission

Invertible Networks or Partons to Detector and Back Again

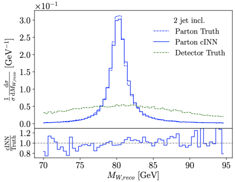
Marco Bellagente[1], Anja Butter[1], Gregor Kasieczka[3], Tilman Plehn[1], Armand Rousselot[1,2], Ramon Winterhalder[1], Lynton Ardizzone[2], and Ullrich Köthe[2]

1 Institut für Theoretische Physik, Universität Heidelberg, Germany
2 Heidelberg Collaboratory 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.



Targeting theory

Navigating string landscape [reinforcement learning]

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

$\rightarrow$ Model space sampling

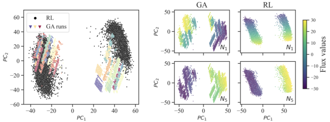


Figure 1: *Left:* Cluster structure in dimensionally reduced flux samples for RL and 25 GA runs (PCA on all samples of GA and RL). The colors indicate individual GA runs. *Right:* Dependence on flux (input) values (N_3 and N_5 respectively) in relation to principal components for a PCA fit of the individual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

Targeting theory

Navigating string landscape [reinforcement learning]

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

→ Model space sampling

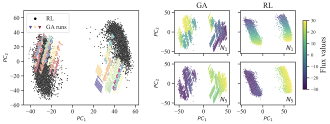


Figure 1: *Left:* Cluster structure in dimensionally reduced flux samples for RL and 25 GA runs (PCA on all samples of GA and RL). The colors indicate individual GA runs. *Right:* Dependence on flux (input) values (N_3 and N_5 respectively) in relation to principal components for a PCA fit of the individual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

→ Useful approximate formulas

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	$(x_{p,2} + a)(bx_{p,1}(c - \Delta\phi) - x_{p,1}(d\Delta\eta + ex_{p,2} + f)\sin(\Delta\phi + g))$	$8.18 \cdot 10^{-3}$

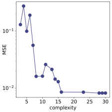


Table 8: Score hall of fame for simplified WBF Higgs production with $f_{W\widetilde{W}} = 0$, including a optimization fit.

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.

Precision regression

Regression as in jet calibration?

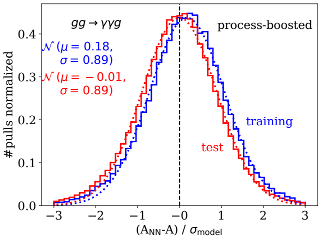
- example: loop amplitudes $gg \to \gamma\gamma g(g)$
- training data $A_j(x)$ exact
- boostable likelihood loss

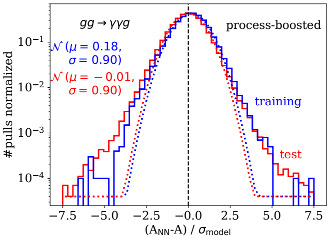
$$L \sim \sum_{\text{points } j} n_j \times \left[\frac{\left| A_j(\omega) - A_j^{\text{truth}} \right|^2}{2\sigma_j(\omega)^2} + \log \sigma_j(\omega) \right] \cdots$$

- pull Gaussian?

$$\frac{A_j(\omega) - A_j^{\text{truth}}}{\sigma_j(\omega)}$$

- NN-fit $\longrightarrow$ NN-interpolation [n_j as function of pull, σ, A,...]





Precision regression

Regression as in jet calibration?

- example: loop amplitudes $gg \to \gamma\gamma g(g)$
- training data $A_j(x)$ exact
- boostable likelihood loss

$$L \sim \sum_{\text{points } j} n_j \times \left[\frac{\left| A_j(\omega) - A_j^{\text{truth}} \right|^2}{2\sigma_j(\omega)^2} + \log \sigma_j(\omega) \right] \cdots$$

- pull Gaussian?

$$\frac{A_j(\omega) - A_j^{\text{truth}}}{\sigma_j(\omega)}$$

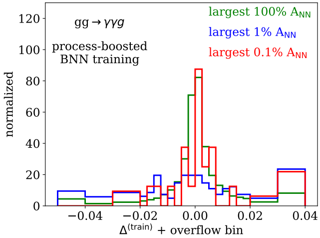
- NN-fit $\longrightarrow$ NN-interpolation [n_j as function of pull, σ, A,...]

Precision

- quality of NN-amplitudes

$$\Delta_j = \frac{\langle A \rangle_j - A_j^{\text{truth}}}{A_j^{\text{truth}}}$$

$\to$ Beyond fit-like regression



Precision regression

Regression as in jet calibration?

- example: loop amplitudes $gg \to \gamma\gamma g(g)$
- training data $A_j(x)$ exact
- boostable likelihood loss

$$L \sim \sum_{\text{points } j} n_j \times \left[\frac{\left| A_j(\omega) - A_j^{\text{truth}} \right|^2}{2\sigma_j(\omega)^2} + \log \sigma_j(\omega) \right] \cdots$$

- pull Gaussian?

$$\frac{A_j(\omega) - A_j^{\text{truth}}}{\sigma_j(\omega)}$$

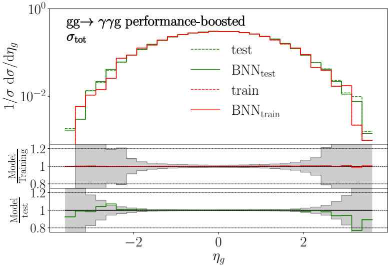
- NN-fit $\longrightarrow$ NN-interpolation [n_j as function of pull, σ, A,...]

Precision

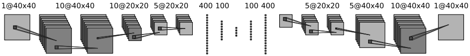
- quality of NN-amplitudes

$$\Delta_j = \frac{\langle A \rangle_j - A_j^{\text{truth}}}{A_j^{\text{truth}}}$$

$\to$ Beyond fit-like regression



Training on QCD only

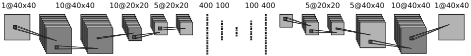


Unsupervised classification

- train on background only
  extract unknown signal from reconstruction error
- reconstruct QCD jets $\rightarrow$ top jets hard to describe
  reconstruct top jets $\rightarrow$ QCD jets just simple top-like jet
- dark-jets complexity: mass drop vs semivisible constituents

$\rightarrow$ Symmetric performance $S \leftrightarrow B$?

Training on QCD only



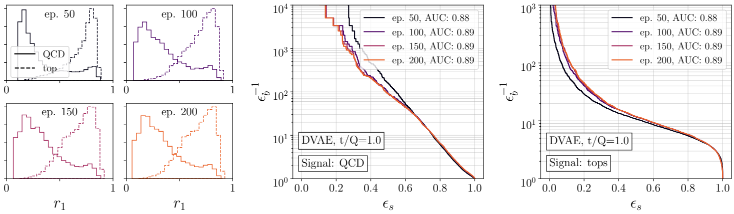
Unsupervised classification

- train on background only
  extract unknown signal from reconstruction error
- reconstruct QCD jets $\rightarrow$ top jets hard to describe
  reconstruct top jets $\rightarrow$ QCD jets just simple top-like jet
- dark-jets complexity: mass drop vs semivisible constituents

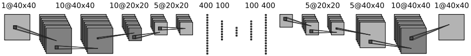
$\rightarrow$ Symmetric performance $S \leftrightarrow B$?

Anomaly score from latent space

- VAE $\rightarrow$ does not work
  GMVAE $\rightarrow$ does not work
  density estimation $\rightarrow$ does not work
  Dirichlet VAE $\rightarrow$ works okay



Training on QCD only



Unsupervised classification

- train on background only
  extract unknown signal from reconstruction error
- reconstruct QCD jets $\rightarrow$ top jets hard to describe
  reconstruct top jets $\rightarrow$ QCD jets just simple top-like jet
- dark-jets complexity: mass drop vs semivisible constituents

$\rightarrow$ Symmetric performance $S \leftrightarrow B$?

Normalized autoencoder [penalize missing features]

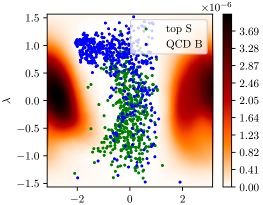
- normalized probability loss
- Boltzmann mapping [E_θ =MSE]

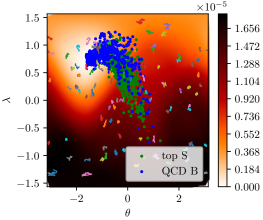
$$p_\theta(x) = \frac{e^{-E_\theta(x)}}{Z_\theta}$$

$$L = -\big\langle \log p_\theta(x) \big\rangle = \big\langle E_\theta(x) + \log Z_\theta \big\rangle$$

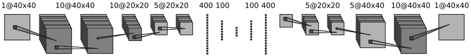
- inducing background metric
- small MSE for data, large MSE for model
- Z_θ from (Langevin) Markov Chain

$\rightarrow$ Proper autoencoder, at last...





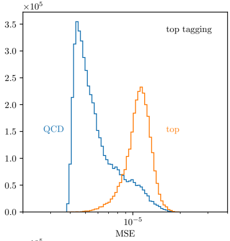
Training on QCD only



Unsupervised classification

- train on background only
  extract unknown signal from reconstruction error
- reconstruct QCD jets $\to$ top jets hard to describe
  reconstruct top jets $\to$ QCD jets just simple top-like jet
- dark-jets complexity: mass drop vs semivisible constituents

$\to$ Symmetric performance $S \leftrightarrow B$?



Normalized autoencoder [penalize missing features]

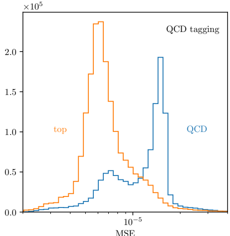
- normalized probability loss
- Boltzmann mapping [E_θ =MSE]

$$p_\theta(x) = \frac{e^{-E_\theta(x)}}{Z_\theta}$$

$$L = -\big\langle \log p_\theta(x) \big\rangle = \big\langle E_\theta(x) + \log Z_\theta \big\rangle$$

- inducing background metric
- small MSE for data, large MSE for model
- Z_θ from (Langevin) Markov Chain

$\to$ Proper autoencoder, at last...



Measuring QCD splitting

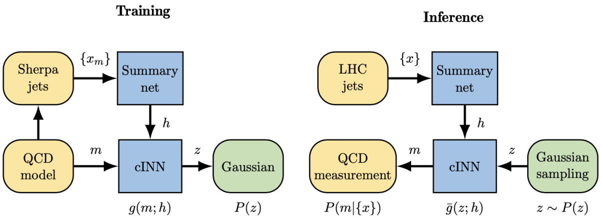
Conditional INN for inference

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

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

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

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



Measuring QCD splitting

Conditional INN for inference

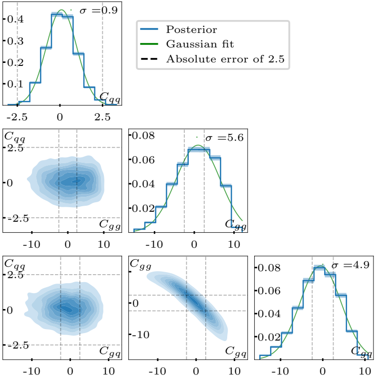
- beyond C_A vs C_F [Kluth etal]

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

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

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

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



ML for the LHC

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

$\rightarrow$ Fun with good old QCD problems

Modern Machine Learning for LHC Physicists

Tilman Plehn[a,*], Anja Butter[a,b], Barry Dillon[a], and Claudius Krause[a,c]

[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

November 2, 2022

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.

Resilient training

Training on simulation, testing on data

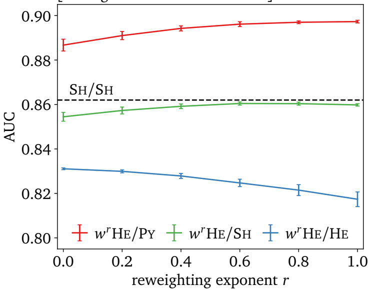
- assume a simulation vs data difference [generalization gap]
  plus, different simulation datasets
- simple question: how train on several datasets?
- adversarial training?
  nuisance parameter?

$\to$ Uncertain feature same as main discriminator??

Constructing an interpolation parameter

- re-weighted samples: Herwig $\xleftrightarrow{0 \leq r \leq 1}$ Pythia
- test data, call it Sherpa
- classify conditionally on r
- 1 use r to define working point
  2. vary r to estimate uncertainty
- best AUC for Pythia training

[reweighted HERWIG → PYTHIA]



Resilient training

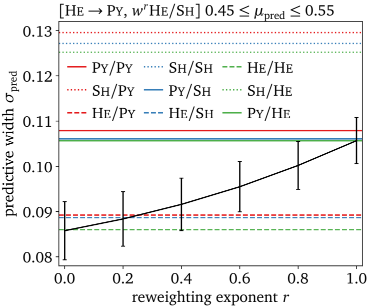
Training on simulation, testing on data

- assume a simulation vs data difference [generalization gap]
  plus, different simulation datasets
- simple question: how train on several datasets?
- adversarial training?
  nuisance parameter?

→ Uncertain feature same as main discriminator??

Constructing an interpolation parameter

- re-weighted samples: Herwig $\xleftrightarrow{0 \leq r \leq 1}$ Pythia
- test data, call it Sherpa
- classify conditionally on r
- 1 use r to define working point
  2. vary r to estimate uncertainty
- best AUC for Pythia training
- lowest uncertainty for Herwig training



Resilient training

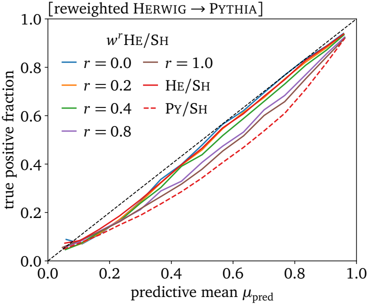
Training on simulation, testing on data

- assume a simulation vs data difference [generalization gap]
  plus, different simulation datasets
- simple question: how train on several datasets?
- adversarial training?
  nuisance parameter?

→ Uncertain feature same as main discriminator??

Constructing an interpolation parameter

- re-weighted samples: Herwig $\xleftrightarrow{0 \leq r \leq 1}$ Pythia
- test data, call it Sherpa
- classify conditionally on r
- 1 use r to define working point
  2. vary r to estimate uncertainty
- best AUC for Pythia training
- lowest uncertainty for Herwig training
- best calibration for Herwig



Resilient training

Training on simulation, testing on data

- assume a simulation vs data difference [generalization gap]
  plus, different simulation datasets
- simple question: how train on several datasets?
- adversarial training?
  nuisance parameter?

$\rightarrow$ Uncertain feature same as main discriminator??

Constructing an interpolation parameter

- re-weighted samples: Herwig $\xleftrightarrow{0 \leq r \leq 1}$ Pythia
- test data, call it Sherpa
- classify conditionally on r
- 1 use r to define working point
  2. vary r to estimate uncertainty
- best AUC for Pythia training
- lowest uncertainty for Herwig training
- best calibration for Herwig
- continuous approach to calibration?

$\rightarrow$ A hammer looking for nails...