Precison Simulations Using Machine Learning

Tilman Plehn

Universität Heidelberg

Glasgow, April 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 events with 4-vectors?

LHC physics vs data scientist

LHC questions

- How to trigger from 3 PB/s to 300 MB/s?
  Data compression [Netflix]
- How to analyze events with 4-vectors?
  Graph neural networks [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 events with 4-vectors?
  Graph neural networks [Cars]
- 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 events with 4-vectors?
  Graph neural networks [Cars]
- 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 events with 4-vectors?
  Graph neural networks [Cars]
- 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 events with 4-vectors?

  Graph neural networks [Cars]
- 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 events with 4-vectors?
  Graph neural networks [Cars]
- 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 events with 4-vectors?

  Graph neural networks [Cars]
- 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 events with 4-vectors?
  Graph neural networks [Cars]
- 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 events with 4-vectors?
  Graph neural networks [Cars]
- 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 events with 4-vectors?
  Graph neural networks [Cars]
- 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 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 events with 4-vectors?
  Graph neural networks [Cars]
- 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 analyse LHC data?
  Simulation-based inference [Edinburgh Ultra-Mini 2007]

LHC physics vs data scientist

LHC questions

- How to trigger from 3 PB/s to 300 MB/s?
  Data compression [Netflix]
- How to analyze events with 4-vectors?
  Graph neural networks [Cars]
- 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 analyse LHC data?
  Simulation-based inference [Edinburgh Ultra-Mini 2007]
- How to treat 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

- define loss function
- minimize loss to find best θ
- 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

Networks with error bar

Training-related uncertainties

- different trainings
  different initalizations
  different data sets
- histogram network output: $f_\theta(x) \pm \Delta f(x)$

→ Bayesian network: $\Delta f_\theta(x)$ from $\Delta\theta$ [Yarin Gal (2016)]

Energy measurement with NN

- expectation value from probability distribution

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

- energy $p(E|\theta)$ encoded in network parameters
  parameters $p(\theta|T)$ trained on data T

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

→ Prediction by sampling weights

$$\langle E \rangle = \int dE \; d\theta \; E \; p(E|\theta) \; p(\theta|T) = \int dE \; d\theta \; E \; p(E|\theta) \; q(\theta)$$

Constructing the loss function

Training means encoding $p(\theta|T)$

- so-called variational approximation [think $q(\theta)$ as Gaussian with mean and width]

$$p(E) = \int d\theta \; p(E|\theta) \; p(\theta|T) \stackrel{!}{=} \int d\theta \; p(E|\theta) \; q(\theta)$$

- similarity through minimized KL-divergence

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

Constructing the loss function

Training means encoding $p(\theta|T)$

- so-called variational approximation [think $q(\theta)$ as Gaussian with mean and width]

$$p(E) = \int d\theta\; p(E|\theta)\; p(\theta|T) \stackrel{!}{=} \int d\theta\; p(E|\theta)\; q(\theta)$$

- similarity through minimized KL-divergence

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

- Bayes' theorem to replace $p(\theta|T)$

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

- normalize distributions, ignore irrelevant terms, so minimize

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

Constructing the loss function

Training means encoding $p(\theta|T)$

- so-called variational approximation [think $q(\theta)$ as Gaussian with mean and width]

$$p(E) = \int d\theta \; p(E|\theta) \; p(\theta|T) \stackrel{!}{=} \int d\theta \; p(E|\theta) \; q(\theta)$$

- similarity through minimized KL-divergence

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

- Bayes' theorem to replace $p(\theta|T)$

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

- normalize distributions, ignore irrelevant terms, so minimize

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

→ Loss combining likelihood and regularization

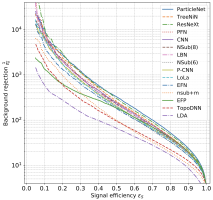
$$L = - \int d\theta \; q(\theta) \; \log p(T|\theta) + D_\text{KL}[q(\theta), p(\theta)]$$

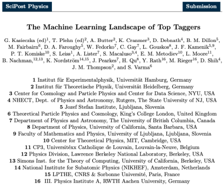
ML-applications for analysis

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]

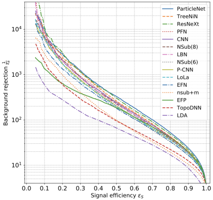
1 Institut für Experimentalphysik, Universität Hamburg, Germany
2 Institut für Theoretische Physik, Universität Heidelberg, Germany
3 Center for Cosmology and Particle Physics and Center for Data Science, NYU, USA
4 NHECT, Dept. of Physics and Astronomy, Rutgers, The State University of NJ, USA
5 Jozef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

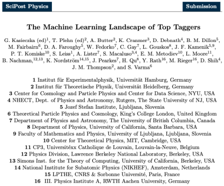
ML-applications for analysis

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

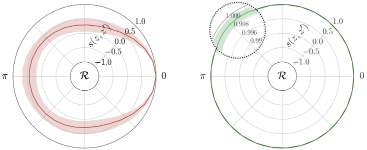
G. Kasieczka (ed)[1], T. Plehn (ed)[2], A. Butter[2], K. Cranmer[3], D. Debnath[4], B. M. Dillon[5], M. Fairbairn[6], D. A. Faroughy[5], W. Fedorko[7], C. Gay[7], L. Gouskos[8], J. F. Kamenik[5,9], P. T. Komiske[10], S. Leiss[1], A. Lister[7], S. Macaluso[3,4], E. M. Metodiev[10], L. Moore[11], B. Nachman,[12,13], 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]

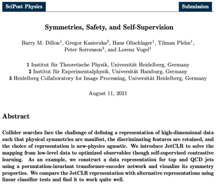
1 Institut für Experimentalphysik, Universität Hamburg, Germany
2 Institut für Theoretische Physik, Universität Heidelberg, Germany
3 Center for Cosmology and Particle Physics and Center for Data Science, NYU, USA
4 NHECT, Dept. of Physics and Astronomy, Rutgers, The State University of NJ, USA
5 Jozef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

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.

Events and amplitudes

Speeding up Sherpa and MadNIS [sampling]

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

$\rightarrow$ 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
$x_{\text{max}}^{\text{p.m.}}$	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
$x_{\text{max}}^{\text{med}}$	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

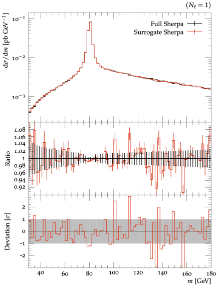
K. Danziger[1], T. Janßen[2], S. Schumann[2], F. Siegert[1]

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.



Events and amplitudes

Speeding up Sherpa and MadNIS [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
$x_{\text{max}}^{\text{p.m.}}$	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
$x_{\text{max}}^{\text{med}}$	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

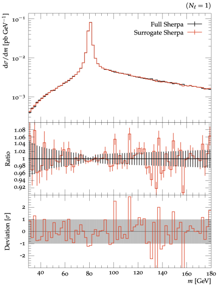
K. Danziger[1], T. Janßen[2], S. Schumann[2], F. Siegert[1]

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

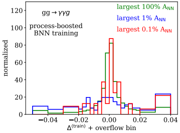
The generation of unit-weight events for complex scattering processes presents a severe challenge to modern Monte Carlo event generators. Even when using sophisticated phase-space sampling techniques adapted to the underlying transition matrix elements, the efficiency for generating unit-weight events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

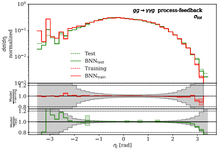


Speeding up amplitudes [precision regression]

- 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

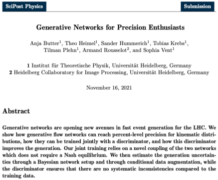
Abstract: Machine learning technology has the potential to dramatically optimise event generation and simulations. We continue to investigate the use of neural networks to approximate matrix elements for high-multiplicity scattering processes. We focus on the case of loop-induced diphoton production 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 $2 \to 3$ 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.

Invertible event generation

Precision NN-generators [Bayesian discriminator-flows]

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

$\rightarrow$ Discussed later



SciPost Physics Submission

Generative Networks for Precision Enthusiasts

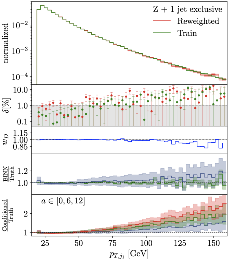
Anja Butter[1], Theo Heimel[1], Sander Hummerich[1], Tobias Krebs[1], Tilman Plehn[1], Armand Rousselot[2], and Sophia Vent[1]

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.

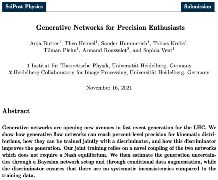


Invertible event generation

Precision NN-generators [Bayesian discriminator-flows]

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

→ Discussed later


SciPost Physics Submission

Generative Networks for Precision Enthusiasts

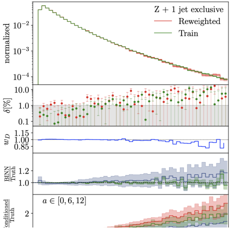
Anja Butter[1], Theo Heimel[1], Sander Hummerich[1], Tobias Krebs[1], Tilman Plehn[1], Armand Rousselot[2], and Sophia Vent[1]

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

November 16, 2021

Abstract

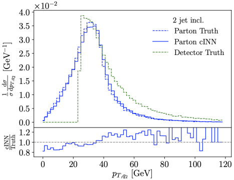
Generative networks are opening new avenues in fast event generation for the LHC. We show how generative flow networks can reach percent-level precision for kinematic distributions, how they can be trained jointly with a discriminator, and how this discriminator improves the generation. Our joint training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

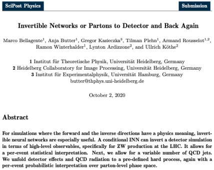


Unfolding and inversion [conditional normalizing flows]

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

→ Discussed later




SciPost Physics Submission

Invertible Networks or Partons to Detector and Back Again

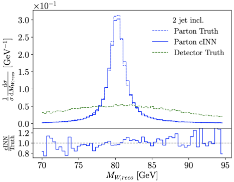
Marco Bellagente[1], Anja Butter[1], Gregor Kasieczka[3], Tilman Plehn[1], Armand Rousselot[1,2], Ramon Winterhalder[1], Lynton Ardizzone[2], and Ullrich Köthe[2]

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.



Modern generative networks

Generative networks

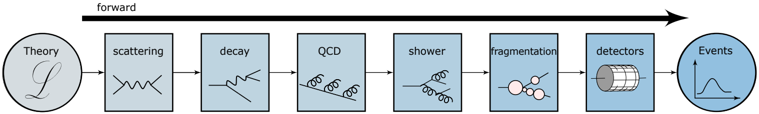
- generate new images, text blocks, etc
- encode density in target space
  sample from Gaussian into target space
- reproduce training data, statistically independently

Modern generative networks

Generative networks

- generate new images, text blocks, etc
- encode density in target space
  sample from Gaussian into target space
- reproduce training data, statistically independently
- Variational Autoencoder
  → low-dimensional physics, high-dimensional objects
- Generative Adversarial Network
  → generator trained by classifier
- Normalizing Flow/Diffusion Model
  → stable bijective mapping
- Generative Pre-trained Transformer
  → learning all structures

→ Pick best model for purpose



Modern generative networks

Generative networks

- generate new images, text blocks, etc
- encode density in target space
  sample from Gaussian into target space
- reproduce training data, statistically independently
- Variational Autoencoder
  $\to$ low-dimensional physics, high-dimensional objects
- Generative Adversarial Network
  $\to$ generator trained by classifier
- Normalizing Flow/Diffusion Model
  $\to$ stable bijective mapping
- Generative Pre-trained Transformer
  $\to$ learning all structures

$\to$ Pick best model for purpose

Fundamental question: GANplification

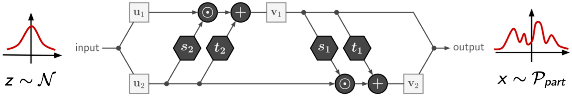
- first generated instances reproducing structures
- too many generated instances reproducing noise?

Modern generative networks

Normalizing flows — INN

- Gaussian latent space
- bijective mapping
- known Jacobian
- likelihood loss
- variety of coupling layers

→ Perfect for speed and precision

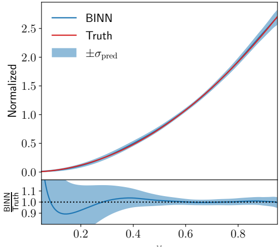


Modern generative networks

Normalizing flows — INN

- Gaussian latent space
- bijective mapping
- known Jacobian
- likelihood loss
- variety of coupling layers

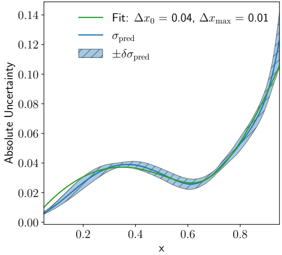
→ Perfect for speed and precision



INNs with uncertainties

- Bayesian NN for density estimation
- events with error bars
- density & uncertainty maps cross-talking

→ Bayesian INNs just fits with error bars



Precision generator

ML-event generators

- useful ML-playground
- training from event samples
  no momentum conservation
  no detector effects [sharper structures]

1- top-quark pairs $t\bar{t} \to 6$ jets [resonance peaks]

2- $Z_{\mu\mu} + \{1, 2, 3\}$ jets [Z-peak, variable jet number, jet-jet topology]

Precision generator

ML-event generators

- useful ML-playground
- training from event samples
  no momentum conservation
  no detector effects [sharper structures]

1- top-quark pairs $t\bar{t} \to 6$ jets [resonance peaks]

2- $Z_{\mu\mu} + \{1, 2, 3\}$ jets [Z-peak, variable jet number, jet-jet topology]

INN-generator [Butter, Heimel, Hummerich, Krebs, TP, Rousselot, Vent]

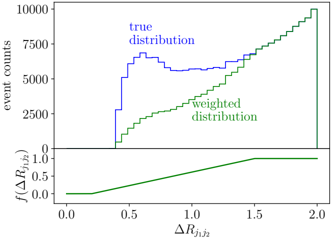
- challenging ΔR_{jj} features
- opposite of importance sampling

$$w^{(\text{1-jet})} = 1$$

$$w^{(\text{2-jet})} = f(\Delta R_{j_1, j_2})$$

$$w^{(\text{3-jet})} = f(\Delta R_{j_1, j_2}) f(\Delta R_{j_2, j_3}) f(\Delta R_{j_1, j_3})$$

$$f(\Delta R) = \frac{\Delta R - R_-}{R_+ - R_-} \qquad (\Delta R \in [R_-, R_+])$$



Precision generator

ML-event generators

- useful ML-playground
- training from event samples
  no momentum conservation
  no detector effects [sharper structures]

1- top-quark pairs $t\bar{t} \to 6$ jets [resonance peaks]

2- $Z_{\mu\mu} + \{1, 2, 3\}$ jets [Z-peak, variable jet number, jet-jet topology]

INN-generator [Butter, Heimel, Hummerich, Krebs, TP, Rousselot, Vent]

- challenging ΔR_{jj} features
- opposite of importance sampling

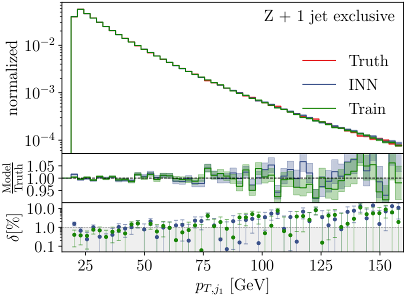
$$w^{(1\text{-jet})} = 1$$

$$w^{(2\text{-jet})} = f(\Delta R_{j_1, j_2})$$

$$w^{(3\text{-jet})} = f(\Delta R_{j_1, j_2}) f(\Delta R_{j_2, j_3}) f(\Delta R_{j_1, j_3})$$

$$f(\Delta R) = \frac{\Delta R - R_-}{R_+ - R_-} \qquad (\Delta R \in [R_-, R_+])$$

$\to$ Per-cent precision in reach

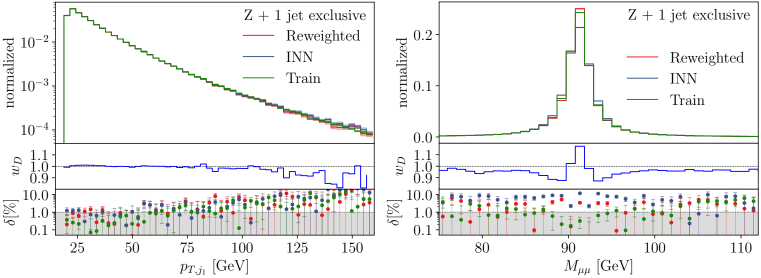


Controlled precision generator

Discriminator: training vs generated

- probability output $D = 0(\text{generator}), 1(\text{truth})$
- decent generator $D \approx 0.5$
- additional event weight $w_D = D/(1-D)$

→ Dual use — control & reweight



Controlled precision generator

Discriminator: training vs generated

- probability output $D = 0(\text{generator}), 1(\text{truth})$
- decent generator $D \approx 0.5$
- additional event weight $w_D = D/(1-D)$

$\to$ Dual use — control & reweight

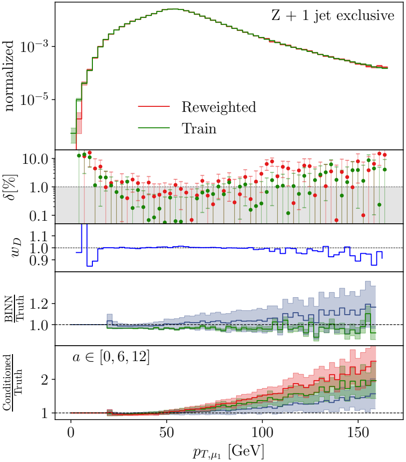
Uncertainties

- training uncertainties from BINN
- low statistics challenging
- systematics from data augmentation
- adjust data in tails $[a = 0 \dots 30]$

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

- train conditionally on smeared a
- error bar from sampling a

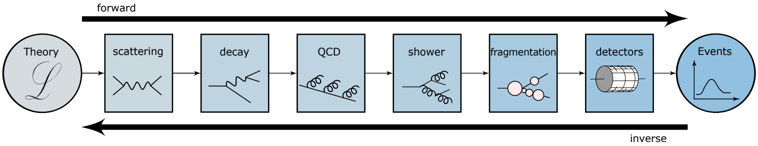
$\to$ INNs for LHC standards



Inverse simulation

Invertible ML-simulation

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



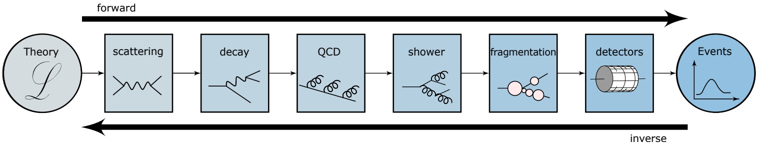
Inverse simulation

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

$\to$ Free choice of data-theory inference point



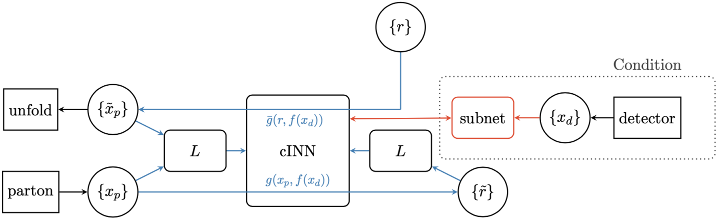
Inverting to hard process

Conditional INN

- partonic events x_p from $\{r\}$, given reco-event x_r
- loss based on likelihood

$$\begin{aligned} L &= -\left\langle \log p(\theta|x_p,x_r)\right\rangle_{x_p,x_r} \\ &= -\left\langle \log p(x_p|x_r,\theta) + \log p(\theta|x_r) - \log p(x_p|x_r)\right\rangle_{x_p,x_r} \\ &= -\left\langle \log p(x_p|x_r,\theta)\right\rangle_{x_p,x_r} - \log p(\theta) + \text{const.} \\ &= -\left\langle \log p(g(x_p|x_r)) + \log\left|\frac{\partial g(x_p|x_r)}{\partial x_p}\right|\right\rangle_{x_p,x_r} - \log p(\theta) + \text{const.} \end{aligned}$$

$\rightarrow$ Stable and statistically calibrated



Inverting to hard process

Conditional INN

- partonic events x_p from $\{r\}$, given reco-event x_r
- loss based on likelihood

$$L = -\left\langle \log p(\theta|x_p, x_r) \right\rangle_{x_p, x_r}$$

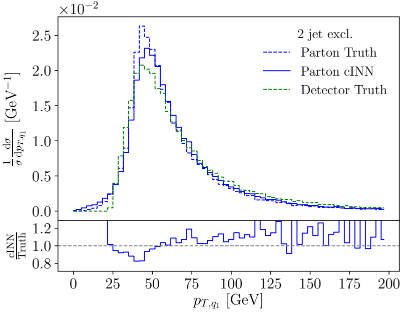
$$= -\left\langle \log p(g(x_p|x_r)) + \log \left| \frac{\partial g(x_p|x_r)}{\partial x_p} \right| \right\rangle_{x_p, x_r} - \log p(\theta) + \text{const.}$$

$\to$ Stable and statistically calibrated

Undo detector and QCD jet radiation in $pp \to ZW$+jets

- hard process given
- detector and reconstruction universal
- jet radiation (approximately) universal
- model-independence: Butter-Malaescu

$\to$ Stable and statistically calibrated



Inverting to hard process

Conditional INN

- partonic events x_p from $\{r\}$, given reco-event x_r
- loss based on likelihood

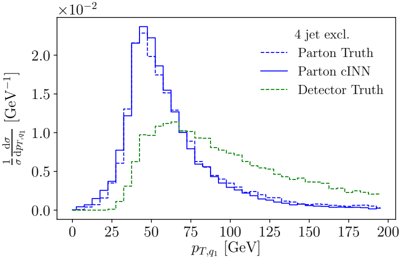
$$L = -\left\langle \log p(\theta|x_p, x_r) \right\rangle_{x_p,x_r}$$
$$= -\left\langle \log p(g(x_p|x_r)) + \log \left| \frac{\partial g(x_p|x_r)}{\partial x_p} \right| \right\rangle_{x_p,x_r} - \log p(\theta) + \text{const.}$$

$\to$ Stable and statistically calibrated

Undo detector and QCD jet radiation in $pp \to ZW$+jets

- hard process given
- detector and reconstruction universal
- jet radiation (approximately) universal
- model-independence: Butter-Malaescu

$\to$ Stable and statistically calibrated



Inverting to hard process

Conditional INN

- partonic events x_p from $\{r\}$, given reco-event x_r
- loss based on likelihood

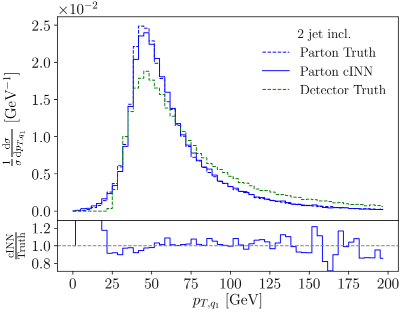
$$L = -\left\langle \log p(\theta|x_p, x_r) \right\rangle_{x_p,x_r}$$
$$= -\left\langle \log p(g(x_p|x_r)) + \log \left| \frac{\partial g(x_p|x_r)}{\partial x_p} \right| \right\rangle_{x_p,x_r} - \log p(\theta) + \text{const.}$$

$\to$ Stable and statistically calibrated

Undo detector and QCD jet radiation in $pp \to ZW$+jets

- hard process given
- detector and reconstruction universal
- jet radiation (approximately) universal
- model-independence: Butter-Malaescu

$\to$ Stable and statistically calibrated



Optimal observables

Measure model parameter θ optimally

- 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

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

- leading order parton level

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

Optimal observables

Measure model parameter θ optimally

- single-event likelihood

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

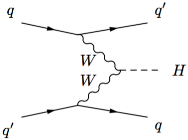
- expanded in θ around θ_0, define score

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

- leading order parton level

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

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

PySR

Analytic formula for score

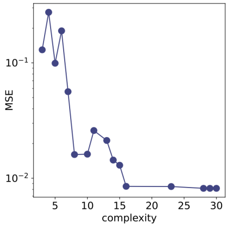
- function to approximate $t(x|\theta)$
- phase space parameters $x_p = p_T/m_H, \Delta\eta, \Delta\phi$ [node]
- operators $\sin x, x^2, x^3, x+y, x-y, x*y, x/y$ [node]
- represent formula as tree [complexity = number of nodes]

⇒ Figures of merit

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

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



PySR

Analytic formula for score

- 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

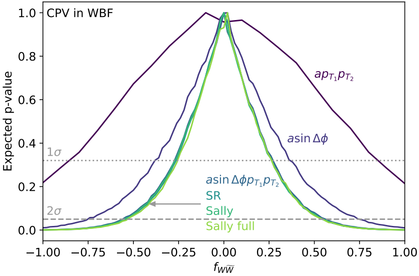
$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left[g_i(x) - t(x,z|\theta)\right]^2 \to \text{MSE} + \text{parsimony} \cdot \text{complexity}$$

Score around Standard Model

- expected limits:

  very wrong formula
  wrong formula
  right formula
  MadMiner
- same within statistical limitation

⇒ New optimal observables next



ML for LHC Theory

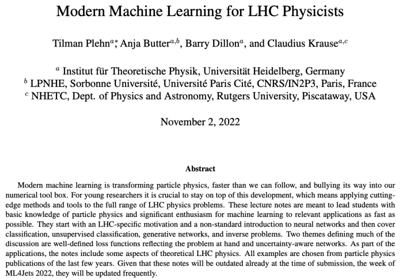
ML-applications

- 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

$\rightarrow$ Fun with good old LHC 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.

Inverting to QCD

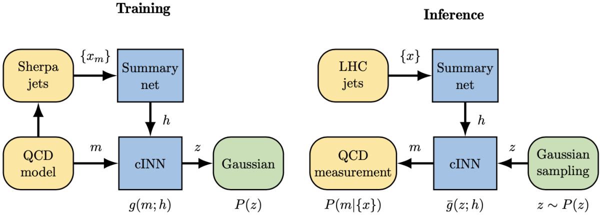
cINN for inference [Bieringer, Butter, Heimel, Höche, Köthe, TP, Radev]

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

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

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

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



Inverting to QCD

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

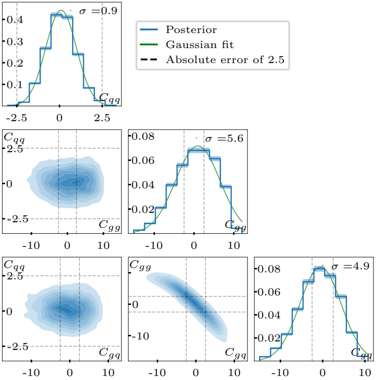
- condition jets with QCD parameters
  train model parameters → Gaussian latent space
  test Gaussian sampling → parameter measurement
- beyond C_A vs C_F [Kluth etal]

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

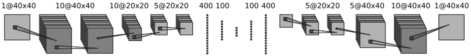
$$P_{gg} = 2C_A \left[D_{gg} \left(\frac{z(1-y)}{1-z(1-y)} + \frac{(1-z)(1-y)}{1-(1-z)(1-y)} \right) + F_{gg} z(1-z) + C_{gg} yz(1-z) \right]$$

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

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



Learning background only

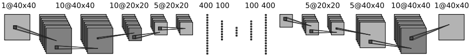


Unsupervised classification

- 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

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

Learning background only



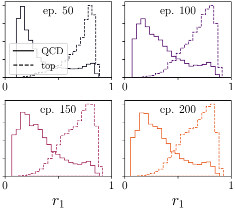
Unsupervised classification

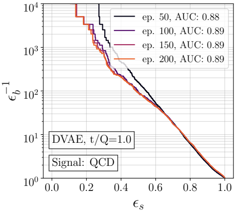
- train on background only
  extract unknown signal from reconstruction error
- reconstruct QCD jets $\to$ top jets hard to describe
- reconstruct top jets $\to$ QCD jets just simple top-like jet

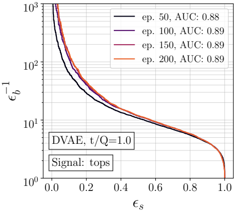
$\to$ Symmetric performance $S \leftrightarrow B$?

Moving to latent space

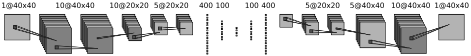
- anomaly score from latent space?
- VAE $\to$ does not work
  GMVAE $\to$ does not work
  Dirichlet VAE $\to$ works okay
  density estimation $\to$ does not work







Learning background 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

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

Normalized autoencoder [penalize missing features]

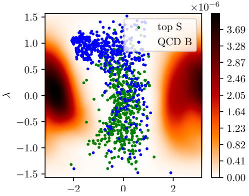
- normalized probability loss
- Boltzmann mapping [E_θ =MSE]

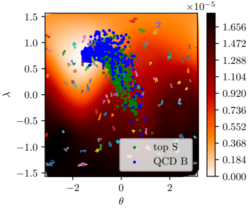
$$p_\theta(x) = \frac{e^{-E_\theta(x)}}{Z_\theta}$$

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

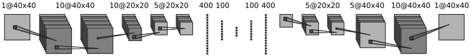
- inducing background metric
- small MSE for data, large MSE for model
- Z_θ from (Langevin) Markov Chain

$\rightarrow$ Symmetric autoencoder, at last





Learning background 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

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

Normalized autoencoder [penalize missing features]

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

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

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

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

$\rightarrow$ Symmetric autoencoder, at last