Generative Networks in Particle Physics

Tilman Plehn

Universität Heidelberg

Hamburg, 2/2022

Particle physics goals

Particle physics defined by

- fundamental questions
- lot of data
- first-principles predictions
- precision analysis

Fundamental questions

- particle nature of dark matter?
- origin of the Higgs mechanism?
- matter-antimatter asymmetry?
- Standard Model all there is?

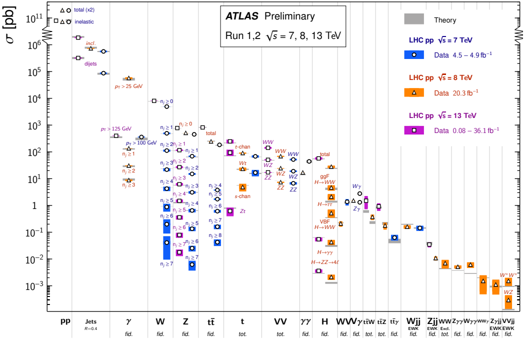
Particle physics goals

Particle physics defined by

- fundamental questions
- lot of data
- first-principles predictions
- precision analysis

Rate measurements

- many processes
- vastly different rates
- high precision
- predicted by theory



Particle physics goals

Particle physics defined by

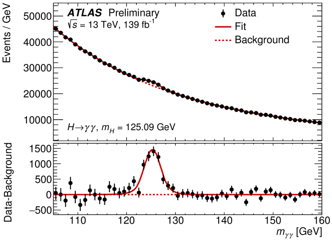
– fundamental questions
– lot of data
– first-principles predictions
– precision analysis

Rate measurements

– many processes
– vastly different rates
– high precision
– predicted by theory

Rates not interesting

– new physics rare and heavy
– phase space vast
$\Rightarrow$ bumps, tails, kinematics instead



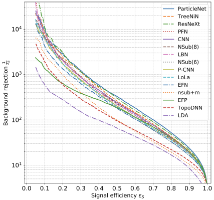
Experimental ML-applications

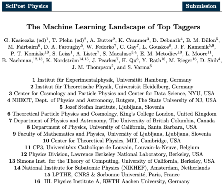
Top tagging [supervised classification]

– different NN-architectures

– tagger comparison

⇒ Just do it right...




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

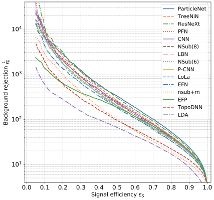
Experimental ML-applications

Top tagging [supervised classification]

– different NN-architectures

– tagger comparison

⇒ Just do it right...



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]

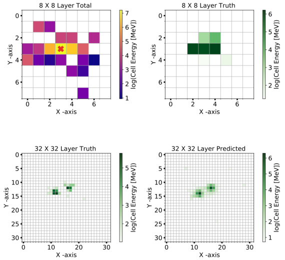
1 Institut für Experimentalphysik, Universität Hamburg, Germany
2 Institut für Theoretische Physik, Universität Heidelberg, Germany
3 Center for Cosmology and Particle Physics and Center for Data Science, NYU, USA
4 NHECT, Dept. of Physics and Astronomy, Rutgers, The State University of NJ, USA
5 Jozef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

⇒ Seriously impressive



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

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.

Jets, QCD, symmetries

Lund plane representation [input preprocessing]

– QCD-inspired input with cutting-edge networks

– angular separation vs transverse momentum

$\Rightarrow$ Understanding data helps

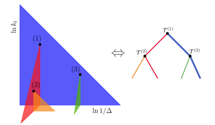
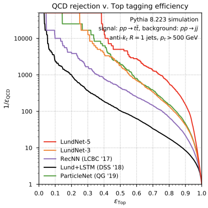


Figure 1. The Lund plane representation of a jet (left) where each emission is positioned according to its Δ and k_t coordinates, and the corresponding mapping to a binary Lund tree of tuples (right). The thick blue line represents the primary sequence of tuples $\mathcal{L}_{\mathrm{primary}}$.



Prepared for submission to JHEP OUTP-20-15P

Jet tagging in the Lund plane with graph networks

Frédéric A. Dreyer,[a] Huilin Qu[b]

[a] Rudolf Peierls Centre for Theoretical Physics, Clarendon Laboratory, Parks Road, Oxford OX1 3PU, UK

[b] CERN, EP Department, CH-1211 Geneva 23, Switzerland

Abstract: The identification of boosted heavy particles such as top quarks or vector bosons is one of the key problems arising in experimental studies at the Large Hadron Collider. In this article, we introduce LundNet, a novel jet tagging method which relies on graph neural networks and an efficient description of the radiation patterns within a jet to optimally disentangle signatures of boosted objects from background events. We apply this framework to a number of different benchmarks, showing significantly improved performance for top tagging compared to existing state-of-the-art algorithms. We study the robustness of the LundNet taggers to non-perturbative and detector effects, and show how kinematic cuts in the Lund plane can mitigate overfitting of the neural network to model-dependent contributions. Finally, we consider the computational complexity of this method and its scaling as a function of kinematic Lund plane cuts, showing an order of magnitude improvement in speed over previous graph-based taggers.

Jets, QCD, symmetries

Lund plane representation [input preprocessing]

- QCD-inspired input with cutting-edge networks
- angular separation vs transverse momentum

⇒ Understanding data helps

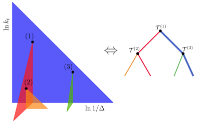
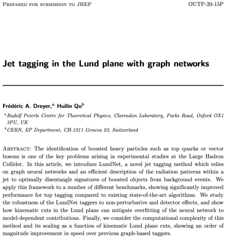


Figure 1. The Lund plane representation of a jet (left) where each emission is positioned according to its Δ and k_t coordinates, and the corresponding mapping to a binary Lund tree of tuples (right). The thick blue line represents the primary sequence of tuples $\mathcal{L}_{\text{primary}}$.


Prepared for submission to JHEP OUTP-20-15P

Jet tagging in the Lund plane with graph networks

Frédéric A. Dreyer,[a] Huilin Qu[b]

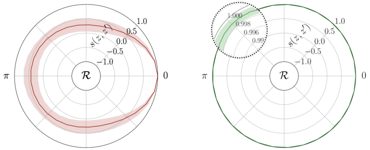
[a] Rudolf Peierls Centre for Theoretical Physics, Clarendon Laboratory, Parks Road, Oxford OX1 3PU, UK
[b] CERN, EP Department, CH-1211 Geneva 23, Switzerland

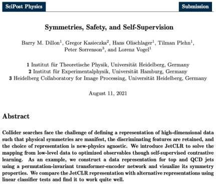
Abstract: The identification of boosted heavy particles such as top quarks or vector bosons is one of the key problems arising in experimental studies at the Large Hadron Collider. In this article, we introduce LundNet, a novel jet tagging method which relies on graph neural networks and an efficient description of the radiation patterns within a jet to optimally disentangle signatures of boosted objects from background events. We apply this framework to a number of different benchmarks, showing significantly improved performance for top tagging compared to existing state-of-the-art algorithms. We study the robustness of the LundNet taggers to non-perturbative and detector effects, and show how kinematic cuts in the Lund plane can mitigate overfitting of the neural network to model-dependent contributions. Finally, we consider the computational complexity of this method and its scaling as a function of kinematic Lund plane cuts, showing an order of magnitude improvement in speed over previous graph-based taggers.

Self-supervised training [contrastive learning, transformer network]

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

⇒ Symmetry-aware latent space




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.

Non-QCD and parton densities

Anomaly searches [unsupervised training]

- look for non-QCD jets, non-SM events
- idea of BSM searches, trigger

⇒ Latent density?

SciPost Physics | Submission

Better Latent Spaces for Better Autoencoders

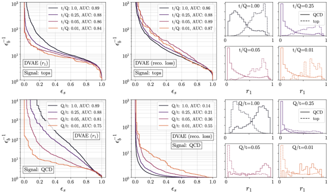
Barry M. Dillon[1], Tilman Plehn[1], Christof Sauer[2], and Peter Sorrenson[3],

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.



Non-QCD and parton densities

Anomaly searches [unsupervised training]

- look for non-QCD jets, non-SM events
- idea of BSM searches, trigger

⇒ Latent density?

SciPost Physics Submission

Better Latent Spaces for Better Autoencoders

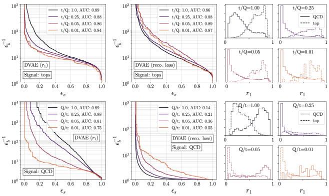
Barry M. Dillon[1], Tilman Plehn[1], Christof Sauer[2], and Peter Sorrenson[3],

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

Events and amplitudes

Speeding up event generation [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

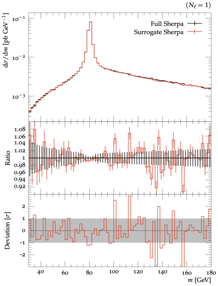
K. Danziger[1], T. Janßen[2], S. Schumann[2], F. Siegert[1]

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 event generation [sampling]

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

⇒ Phase space sampling

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

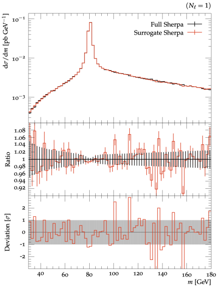
September 27, 2021

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.

	$gg \to t\bar{t}ggg$	$ug \to t\bar{t}ggu$	$uu \to t\bar{t}guu$	$u\bar{u} \to t\bar{t}gd\bar{d}$
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

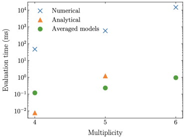
Table 6: Performance measures for partonic channels contributing to $t\bar{t}$+3 jets production at the LHC.

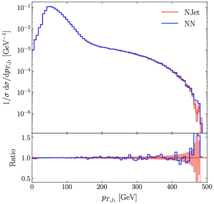


Speeding up amplitudes [regression]

- loop-amplitudes expensive
- interpolation standard

⇒ Network 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.

String landscape and learned formulas

Navigating string landscape [reinforcement learning]

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

⇒ Model space sampling

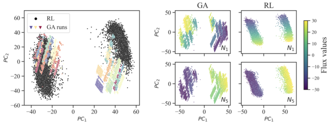


Figure 1: *Left:* Cluster structure in dimensionally reduced flux samples for RL and 25 GA runs (PCA on all samples of GA and RL). The colors indicate individual GA runs. *Right:* Dependence on flux (input) values (N_3 and N_5 respectively) in relation to principal components for a PCA fit of the individual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

String landscape and learned formulas

Navigating string landscape [reinforcement learning]

– searching for viable vacua
– high dimensions, unknown global structure
$\Rightarrow$ Model space sampling

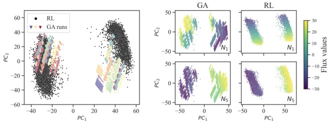


Figure 1: *Left:* Cluster structure in dimensionally reduced flux samples for RL and 25 GA runs (PCA on all samples of GA and RL). The colors indicate individual GA runs. *Right:* Dependence on flux (input) values (N_3 and N_5 respectively) in relation to principal components for a PCA fit of the individual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

Learning formulas [genetic algorithm, symbolic regression]

– approximate numerical function through formula
– example: score/optimal observables
$\Rightarrow$ Useful approximate formulas

compl	dof	function	MSE
3	1	$a\,\Delta\phi$	$1.30\cdot10^{-1}$
4	1	$\sin(a\Delta\phi)$	$2.75\cdot10^{-1}$
5	1	$a\Delta\phi x_{p,1}$	$9.93\cdot10^{-2}$
6	1	$-x_{p,1}\sin(\Delta\phi+a)$	$1.90\cdot10^{-1}$
7	1	$(-x_{p,1}-a)\sin(\sin(\Delta\phi))$	$5.63\cdot10^{-2}$
8	1	$(a-x_{p,1})x_{p,2}\sin(\Delta\phi)$	$1.61\cdot10^{-2}$
14	2	$x_{p,1}(a\Delta\phi-\sin(\sin(\Delta\phi)))(x_{p,2}+b)$	$1.44\cdot10^{-2}$
15	3	$-(x_{p,2}(a\Delta\eta^2+x_{p,1})+b)\sin(\Delta\phi+c)$	$1.30\cdot10^{-2}$
16	4	$-x_{p,1}(a-b\Delta\eta)(x_{p,2}+c)\sin(\Delta\phi+d)$	$8.50\cdot10^{-3}$
28	7	$(x_{p,2}+a)(bx_{p,1}(c-\Delta\phi)$ $-x_{p,1}(d\Delta\eta+ex_{p,2}+f)\sin(\Delta\phi+g))$	$8.18\cdot10^{-3}$

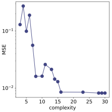


Table 8: Score hall of fame for simplified WBF Higgs production with $f_{W\widetilde{W}}=0$, including a optimization fit.

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

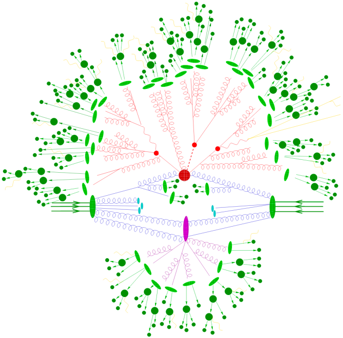
While neural networks offer an attractive way to numerically encode functions, actual formulas remain the language of theoretical particle physics. We use symbolic regression trained on matrix-element information to extract, for instance, optimal LHC observables. This way we invert the usual simulation 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.

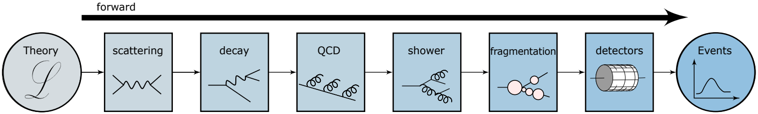
LHC simulations

Simulation-based inference

– start with Lagrangian
– calculate scattering in perturbative QFT
– simulate events [theory: Sherpa, Madgraph, Pythia]
– simulate detectors [experiment: ATLAS, CMS, Delphes]

⇒ LHC events in virtual worlds





LHC simulations

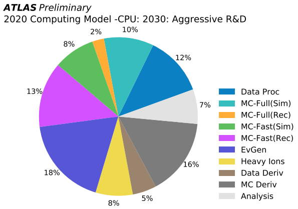
Simulation-based inference

- start with Lagrangian
- calculate scattering in perturbative QFT
- simulate events [theory: Sherpa, Madgraph, Pythia]
- simulate detectors [experiment: ATLAS, CMS, Delphes]

$\Rightarrow$ LHC events in virtual worlds

HL-LHC: preparing for 25-fold data set

- simulated event numbers $\sim$ expected events
- statistics requiring 1%-2% uncertainty
- flexible signal hypotheses [time-dependent]
- low-rate high-multiplicity backgrounds



LHC simulations

Simulation-based inference

– start with Lagrangian

– calculate scattering in perturbative QFT

– simulate events [theory: Sherpa, Madgraph, Pythia]

– simulate detectors [experiment: ATLAS, CMS, Delphes]

$\Rightarrow$ LHC events in virtual worlds

HL-LHC: preparing for 25-fold data set

– simulated event numbers $\sim$ expected events

– statistics requiring 1%-2% uncertainty

– flexible signal hypotheses [time-dependent]

– low-rate high-multiplicity backgrounds

Three ways to use ML

– improve current tools

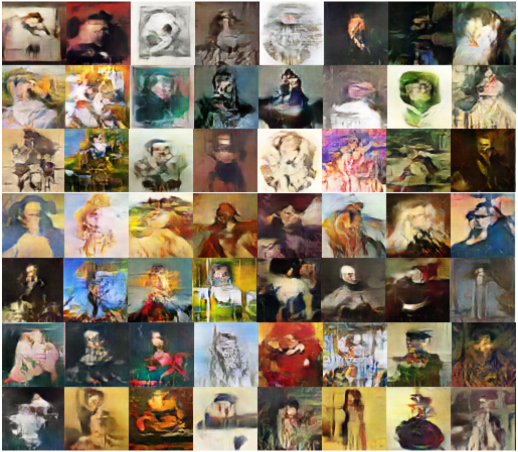
– new ML-tools

– conceptually new ideas

Generative neural networks

GANGogh [Bonafilia, Jones, Danyluk (2017)]

– can networks create new pieces of art?
  map random numbers to image pixels
– train on 80,000 pictures [organized by style and genre]
– generate portraits



Generative neural networks

GANGogh [Bonafilia, Jones, Danyluk (2017)]

– can networks create new pieces of art?
map random numbers to image pixels
– train on 80,000 pictures [organized by style and genre]
– generate portraits

Edmond de Belamy [Caselles-Dupre, Fautrel, Vernier (2018)]

– trained on 15,000 portraits
– sold for $432.500

$\Rightarrow$ ML all marketing and sales

Generative neural networks

GANGogh [Bonafilia, Jones, Danyluk (2017)]

– can networks create new pieces of art?
  map random numbers to image pixels

– train on 80,000 pictures [organized by style and genre]

– generate portraits

Edmond de Belamy [Caselles-Dupre, Fautrel, Vernier (2018)]

– trained on 15,000 portraits

– sold for $432.500

$\Rightarrow$ ML all marketing and sales

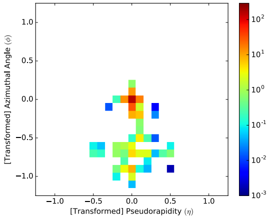
Jet portraits [de Oliveira, Paganini, Nachman (2017)]

– calorimeter or jet images

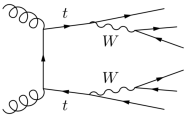
– reproduce valid jet images from training data

– organize them by QCD vs W-decay jets

$\Rightarrow$ Generative networks useful for particle physics



How to GAN



Adversarial training

- training: true events $\{x_T\}$
  output: generated events $\{r\} \to \{x_G\}$
- discriminator classifier function $D(x)$ from minimizing [$D(x) = 1$(T), 0(G)]

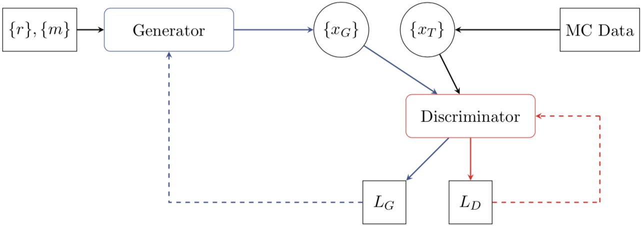
$$L_D = \big\langle -\log D(x) \big\rangle_{x_T} + \big\langle -\log(1 - D(x)) \big\rangle_{x_G}$$

- generator mapping $r \to x_G$ by minimizing [D needed]

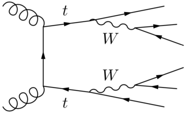
$$L_G = \big\langle -\log D(x) \big\rangle_{x_G}$$

- Nash equilibrium $D = 0.5$

$\Rightarrow$ statistically independent copy of training events



How to GAN



Adversarial training

- training: true events $\{x_T\}$
  output: generated events $\{r\} \to \{x_G\}$
- discriminator classifier function $D(x)$ from minimizing [$D(x) = 1$(T), 0(G)]

$$L_D = \Big\langle -\log D(x) \Big\rangle_{x_T} + \Big\langle -\log(1 - D(x)) \Big\rangle_{x_G}$$

- generator mapping $r \to x_G$ by minimizing [D needed]

$$L_G = \Big\langle -\log D(x) \Big\rangle_{x_G}$$

- Nash equilibrium $D = 0.5$

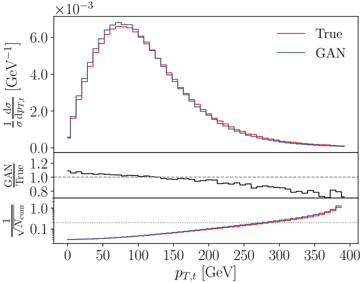
$\Rightarrow$ statistically independent copy of training events

GAN LHC events

- typical process $t\bar{t} \to 6$ quarks [18D final state]
- observables with tails

$\Rightarrow$ two big LHC questions:

How precisely can we GAN?
What is their uncertainty?



Chemistry of loss functions

Pointing GANs to specific features

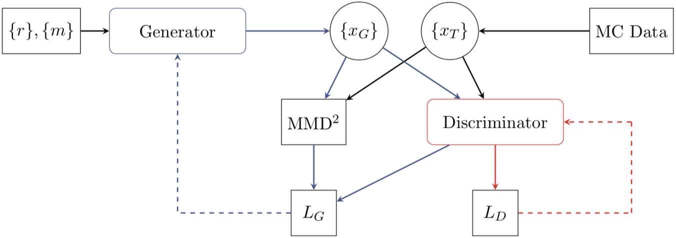
– low-dimensional sharp features

phase space boundaries
kinematic cuts
invariant masses

– batch-wise comparison of distributions, MMD loss with kernel k

$$\text{MMD}^2(P_T, P_G) = \langle k(x, x') \rangle_{x_T, x'_T} + \langle k(y, y') \rangle_{y_G, y'_G} - 2 \langle k(x, y) \rangle_{x_T, y_G}$$

$$L_G \to L_G + \lambda_G \, \text{MMD}^2$$



Chemistry of loss functions

Pointing GANs to specific features

– low-dimensional sharp features

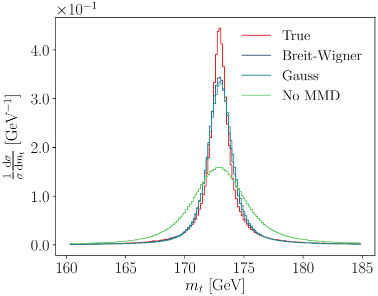
phase space boundaries
kinematic cuts
invariant masses

– batch-wise comparison of distributions, MMD loss with kernel k

$$\text{MMD}^2(P_T, P_G) = \langle k(x,x')\rangle_{x_T,x'_T} + \langle k(y,y')\rangle_{y_G,y'_G} - 2\langle k(x,y)\rangle_{x_T,y_G}$$

$$L_G \to L_G + \lambda_G \,\text{MMD}^2$$

⇒ It works...



GANplification

Gain beyond training data

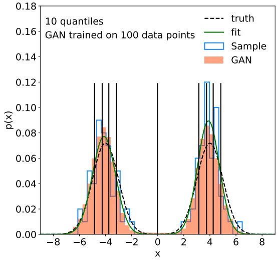
– true function known
compare sampling vs GAN vs fit

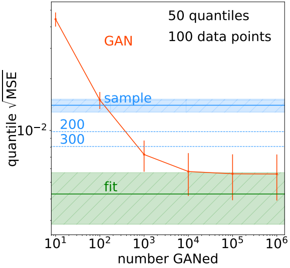
– quantiles with χ^2-values

– start with 100 sampled points
fit like 700 sampled points
GAN like 500 sampled points ...
... but requiring 10,000 GANned events

– interpolation and resolution key [implicit bias]

$\Rightarrow$ Generative networks beyond training data





Statistical bonus: unweighting

Gaining beyond GANpliflication

- phase space sampling: PS weight $\times |\mathcal{M}|^2$
  density information in weights [for uniform grid]
- experiment: observed configurations
  density information in density

$\Rightarrow$ information in mix of density and weights

Statistical bonus: unweighting

Gaining beyond GANplification

– phase space sampling: PS weight $\times |\mathcal{M}|^2$
density information in weights [for uniform grid]

– experiment: observed configurations
density information in density

$\Rightarrow$ information in mix of density and weights

– weak spot: hit-and-miss unweighting

relative event weights $w_j / w_{\max} \in [0, 1]$
random number $r \in [0, 1] < w_j / w_{\max}$ means keep event

Statistical bonus: unweighting

Gaining beyond GANpliflication

– phase space sampling: PS weight $\times |\mathcal{M}|^2$
density information in weights [for uniform grid]

– experiment: observed configurations
density information in density

$\Rightarrow$ information in mix of density and weights

– weak spot: hit-and-miss unweighting

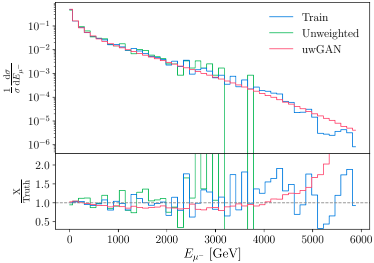
relative event weights $w_j / w_{\text{max}} \in [0, 1]$
random number $r \in [0, 1] < w_j / w_{\text{max}}$ means keep event

– learn from weighted, generate unweighted events

$$L_D = \frac{\big\langle -w(x) \log D(x) \big\rangle_{x_T}}{\big\langle w(x) \big\rangle_{x_T}} + \big\langle -\log(1 - D(x)) \big\rangle_{x_G}$$

$$L_G = \big\langle -\log D(x) \big\rangle_{x_G}$$

$\Rightarrow$ GANs can unweight



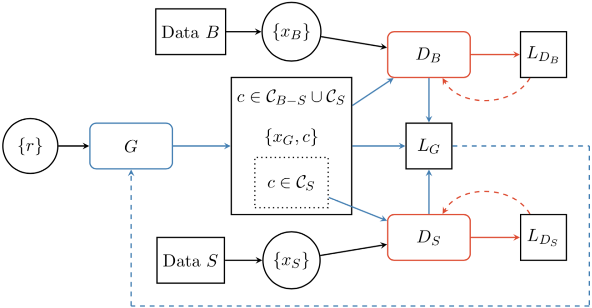
Statistical bonus: subtraction

Subtract samples without binning

– statistical uncertainty

$$\Delta_{B-S} = \sqrt{\Delta_B^2 + \Delta_S^2} > \max(\Delta B, \Delta S)$$

– GAN setup: differential class label, sample normalization



Statistical bonus: subtraction

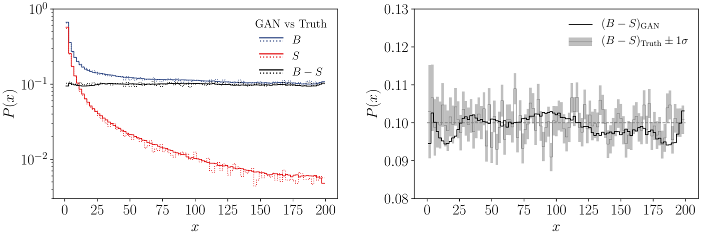
Subtract samples without binning

- statistical uncertainty

$$\Delta_{B-S} = \sqrt{\Delta_B^2 + \Delta_S^2} > \max(\Delta B, \Delta S)$$

- GAN setup: differential class label, sample normalization
- toy example

$$P_B(x) = \frac{1}{x} + 0.1 \qquad P_S(x) = \frac{1}{x} \qquad \Rightarrow \qquad P_{B-S} = 0.1$$



Statistical bonus: subtraction

Subtract samples without binning

– statistical uncertainty

$$\Delta_{B-S} = \sqrt{\Delta_B^2 + \Delta_S^2} > \max(\Delta B, \Delta S)$$

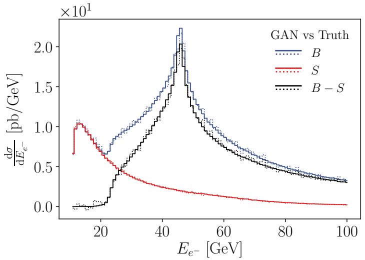
– GAN setup: differential class label, sample normalization

– toy example

$$P_B(x) = \frac{1}{x} + 0.1 \qquad P_S(x) = \frac{1}{x} \qquad \Rightarrow \qquad P_{B-S} = 0.1$$

– event-based background subtraction [weird notation, sorry]

$$pp \to e^+e^- \quad \text{(B)} \qquad pp \to \gamma \to e^+e^- \quad \text{(S)} \qquad \Rightarrow \qquad pp \to Z \to e^+e^- \quad \text{(B-S)}$$



Statistical bonus: subtraction

Subtract samples without binning

– statistical uncertainty

$$\Delta_{B-S} = \sqrt{\Delta_B^2 + \Delta_S^2} > \max(\Delta B, \Delta S)$$

– GAN setup: differential class label, sample normalization

– toy example

$$P_B(x) = \frac{1}{x} + 0.1 \qquad P_S(x) = \frac{1}{x} \qquad \Rightarrow \qquad P_{B-S} = 0.1$$

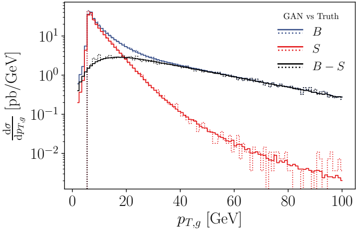
– event-based background subtraction [weird notation, sorry]

$$pp \to e^+e^- \quad \text{(B)} \qquad pp \to \gamma \to e^+e^- \quad \text{(S)} \qquad \Rightarrow \qquad pp \to Z \to e^+e^- \quad \text{(B-S)}$$

– collinear subtraction [assumed non-local]

$pp \to Zg$ (B: matrix element, S: collinear approximation)

$\Rightarrow$ GANs can subtract samples



Variational autoencoder

Alternative generative architecture

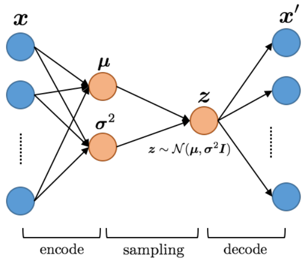
– reconstruction loss [like autoencoder]

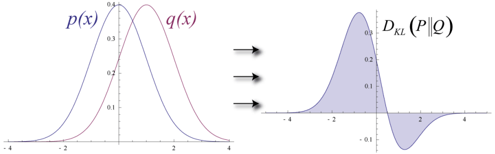
$$L_{\text{VAE}} = \sum |x - x'|^2 + \beta D_{\text{KL}}$$

– Gaussian latent space via KL-divergence

$$D_{\text{KL}}(p; q) = \int dx \; p(x) \log \frac{p(x)}{q(x)}$$

$$D_{\text{KL}}(\mathcal{N}_{\mu,\sigma}; \mathcal{N}_{0,1}) = \frac{1 + \log \sigma^2 - \mu^2 - \sigma^2}{2}$$





Variational autoencoder

Alternative generative architecture

– reconstruction loss [like autoencoder]

$$L_{\text{VAE}} = \sum |x - x'|^2 + \beta D_{\text{KL}}$$

– Gaussian latent space via KL-divergence

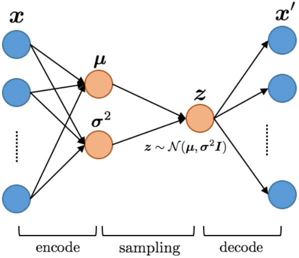
$$D_{\text{KL}}(p; q) = \int dx\, p(x) \log \frac{p(x)}{q(x)}$$

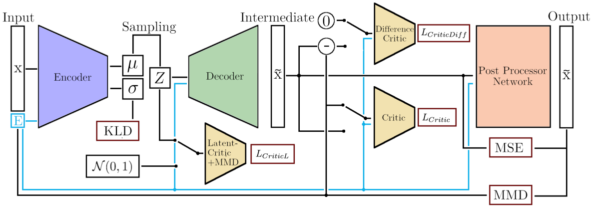
$$D_{\text{KL}}(\mathcal{N}_{\mu,\sigma}; \mathcal{N}_{0,1}) = \frac{1 + \log \sigma^2 - \mu^2 - \sigma^2}{2}$$

– VAE-GAN replacing reconstruction loss

$$L_{\text{VAE-GAN}} = L_{\text{GAN}} + \beta D_{\text{KL}}$$

– application to detector simulations [ask Gregor]





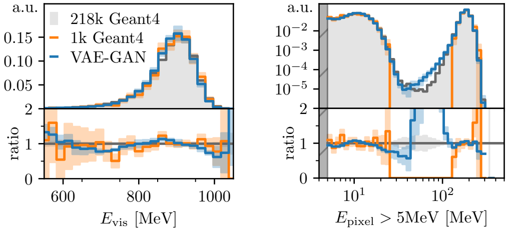
Calomplification

Gain for fast detector simulation

- photon shower in 3D-calorimeter
  energy deposition in 30^3 cells

  1k showers for training, 218k showers as truth
  downsized VAE-GAN architecture

$\Rightarrow$ how many generated events make sense?



Calomplification

Gain for fast detector simulation

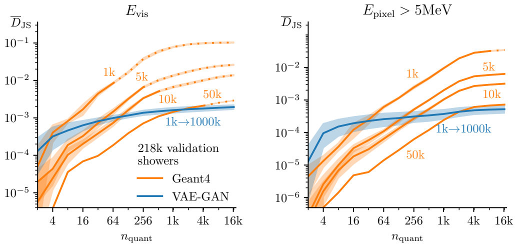
– photon shower in 3D-calorimeter
  energy deposition in 30^3 cells

  1k showers for training, 218k showers as truth
  downsized VAE-GAN architecture

⇒ how many generated events make sense?

– benchmarking as function of quantiles [bin resolution]
  comparison using $D_{JS}(p;q) = D_{\text{KL}}(p;q) + D_{\text{KL}}(q;p)$

⇒ Generative networks really amplify data sets



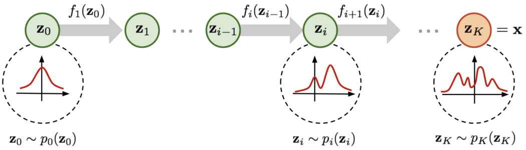
Normalizing flows — invertible networks

Looking for stable networks

- mapping physics space $\longleftrightarrow$ latent space
- INN: bijective mapping
  symmetric training and evaluation
  Gaussian latent space
- structural element: coupling block [affine, spline]
- log-likelihood loss [moved into Gaussian latent space]

$$L_{\text{INN}} = -\left\langle \log \frac{P_G(x)}{P_T(x)} \right\rangle_{x_T}$$

$$= -\left\langle \frac{\psi(x)^2}{2} - \log J(x) - \log P_T(x) \right\rangle_{x_T}$$



Generative networks with error bars

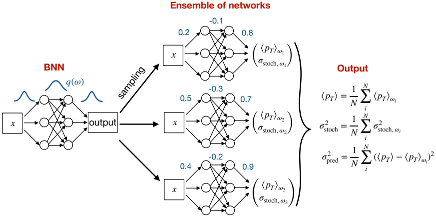
Bayesian generative network

– data: event sample [points in 2D space]

learn phase space density
Gaussian in latent space
mapping bijective
sample from latent space

– Bayesian version

allow weight distributions
learn uncertainty map



Generative networks with error bars

Bayesian generative network

– data: event sample [points in 2D space]

learn phase space density
Gaussian in latent space
mapping bijective
sample from latent space

– Bayesian version

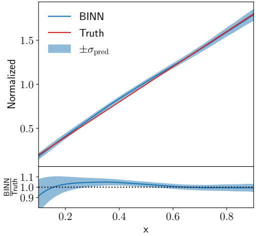
allow weight distributions
learn uncertainty map

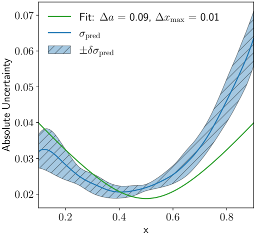
– 2D wedge ramp

$$p(x) = ax + b = ax + \frac{1 - \frac{a}{2}(x_{\text{max}}^2 - x_{\text{min}}^2)}{x_{\text{max}} - x_{\text{min}}}$$

$$(\Delta p)^2 = \left(x - \frac{1}{2}\right)^2 (\Delta a)^2 + \left(1 + \frac{a}{2}\right)^2 (\Delta x_{\text{max}})^2 + \left(1 - \frac{a}{2}\right)^2 (\Delta x_{\text{min}})^2$$

explaining minimum in $\sigma_{\text{pred}}(x)$





Generative networks with error bars

Bayesian generative network

– data: event sample [points in 2D space]

learn phase space density
Gaussian in latent space
mapping bijective
sample from latent space

– Bayesian version

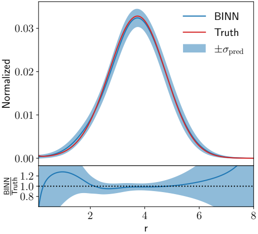
allow weight distributions
learn uncertainty map

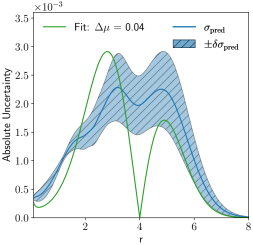
– Gaussian ring [$\mu = 4,\ w = 1$]

$$\Delta p = \left| \frac{G(r)}{r}\ \frac{\mu - r}{w^2} \right|^2 (\Delta\mu)^2 + \left| \frac{(r-\mu)^2}{w^3} - \frac{1}{w} \right|^2 (\Delta w)^2$$

explaining dip in $\sigma_{\text{pred}}(x)$

$\Rightarrow$ Generative networks just (non-parametric) fits



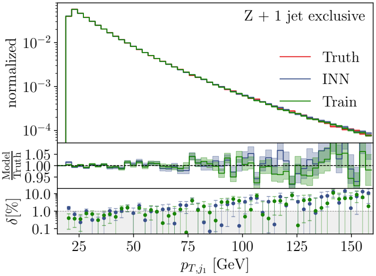


Precision generator

Challenging an INN-generator

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

– training on 5.4M Z+jets events
truth defined as high-stats training data
goal: 1% precision relative to truth



Precision generator

Challenging an INN-generator

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

– training on 5.4M Z+jets events
truth defined as high-stats training data
goal: 1% precision relative to truth

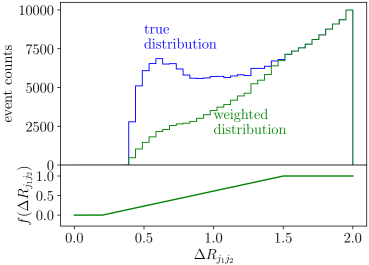
– holes in geometric distance ΔR_{jj} [QFT problem :)]

– magic transformation
monotouous function with weights [opposite of importance sampling]

$$w^{(1\text{-jet})} = 1$$
$$w^{(2\text{-jet})} = f(\Delta R_{j_1,j_2})$$
$$w^{(3\text{-jet})} = f(\Delta R_{j_1,j_2}) f(\Delta R_{j_2,j_3}) f(\Delta R_{j_1,j_3})$$

with

$$f(\Delta R) = \begin{cases} 0 & \text{for } \Delta R < R_- \\ \dfrac{\Delta R - R_-}{R_+ - R_-} & \text{for } \Delta R \in [R_-, R_+] \\ 1 & \text{for } \Delta R > R_+ \end{cases}$$



Precision generator

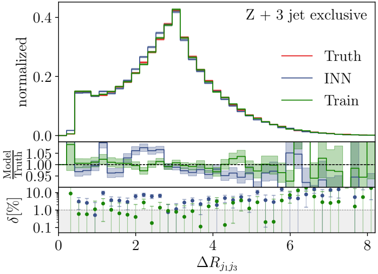
Challenging an INN-generator

- $Z_{\mu\mu} + \{1,2,3\}$ jets [Z-peak, variable jet number, jet-jet topology]
- training on 5.4M Z+jets events
  truth defined as high-stats training data
  goal: 1% precision relative to truth
- holes in geometric distance ΔR_{jj} [QFT problem :)]
- magic transformation
  monotouous function with weights [opposite of importance sampling]

$$w^{(1\text{-jet})} = 1$$
$$w^{(2\text{-jet})} = f(\Delta R_{j_1,j_2})$$
$$w^{(3\text{-jet})} = f(\Delta R_{j_1,j_2}) f(\Delta R_{j_2,j_3}) f(\Delta R_{j_1,j_3})$$

with

$$f(\Delta R) = \begin{cases} 0 & \text{for } \Delta R < R_- \\ \dfrac{\Delta R - R_-}{R_+ - R_-} & \text{for } \Delta R \in [R_-, R_+] \\ 1 & \text{for } \Delta R > R_+ \end{cases}$$



⇒ Per-cent precision possible

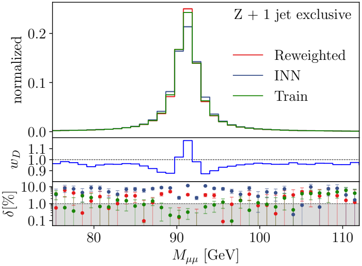
Control & precision

Additional discriminator: training vs generated

- input $\{p_T, \eta, \phi, M, M_{\mu\mu}, \Delta R\}$
  output $D = 0$G), 1T $\rightarrow 0.5$
- decent generator training $D \approx 0.5$
- additional event weight

$$w_D(x) = \frac{D(x)}{1 - D(x)}$$

$\Rightarrow$ 1. control and 2. reweight



Generative Networks

Tilman Plehn

Control & precision

Additional discriminator: training vs generated

- input $\{p_T, \eta, \phi, M, M_{\mu\mu}, \Delta R\}$
  output $D = 0\text{G}), 1\text{T} \to 0.5$
- decent generator training $D \approx 0.5$
- additional event weight

$$w_D(x) = \frac{D(x)}{1 - D(x)}$$

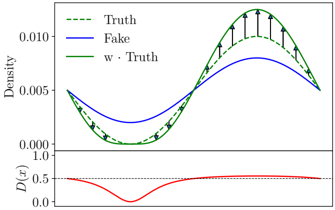
$\Rightarrow$ 1. control and 2. reweight

Joint DiscFlow training [GAN-inspired]

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

$$L_{\text{DiscFlow}} = -\sum_{i=1}^{B} w_D(x_i)^\alpha \; \log \frac{P(x_i)}{P_{\text{ref}}(x_i)}$$

$$\approx -\int dx \; \frac{P_{\text{ref}}^{\alpha+1}(x)}{P^\alpha(x)} \; \log \frac{P(x)}{P_{\text{ref}}(x)}$$



Control & precision

Additional discriminator: training vs generated

- input $\{p_T, \eta, \phi, M, M_{\mu\mu}, \Delta R\}$
  output $D = 0$G), 1T $\to 0.5$
- decent generator training $D \approx 0.5$
- additional event weight

$$w_D(x) = \frac{D(x)}{1 - D(x)}$$

$\Rightarrow$ 1. control and 2. reweight

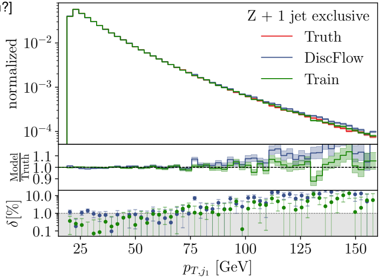
Joint DiscFlow training [GAN-inspired]

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

$$L_{\text{DiscFlow}} = - \sum_{i=1}^{B} w_D(x_i)^\alpha \log \frac{P(x_i)}{P_{\text{ref}}(x_i)}$$

$$\approx - \int dx \, \frac{P_{\text{ref}}^{\alpha+1}(x)}{P^\alpha(x)} \log \frac{P(x)}{P_{\text{ref}}(x)}$$

$\Rightarrow$ Controlled unweighted events



Control & precision

Additional discriminator: training vs generated

– input $\{p_T, \eta, \phi, M, M_{\mu\mu}, \Delta R\}$
  output $D = 0\text{G}), 1\text{T} \to 0.5$
– decent generator training $D \approx 0.5$
– additional event weight

$$w_D(x) = \frac{D(x)}{1 - D(x)}$$

$\Rightarrow$ 1. control and 2. reweight

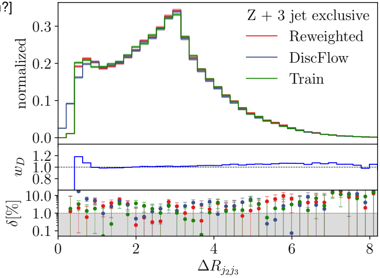
Joint DiscFlow training [GAN-inspired]

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

$$L_{\text{DiscFlow}} = -\sum_{i=1}^{B} w_D(x_i)^\alpha \log \frac{P(x_i)}{P_{\text{ref}}(x_i)}$$

$$\approx -\int dx \frac{P_{\text{ref}}^{\alpha+1}(x)}{P^\alpha(x)} \log \frac{P(x)}{P_{\text{ref}}(x)}$$

$\Rightarrow$ Reweighted precision events

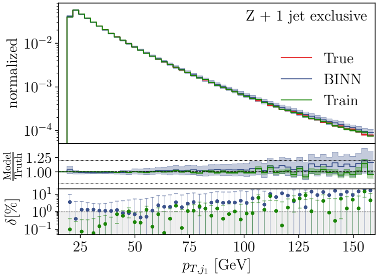


Uncertainties

Bayesian INN generator

- uncertainty over phase space
- training statistics leading source

$\Rightarrow$ Training-related error bars



Uncertainties

Bayesian INN generator

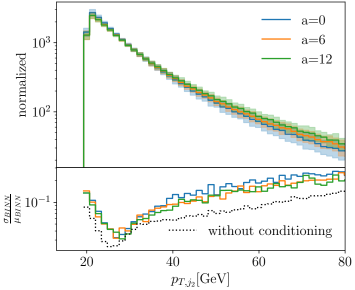
– uncertainty over phase space
– training statistics leading source
⇒ Training-related error bars

Theory uncertainties

– BNN regression/classification: systematics from data augmentation
– systematic uncertainties in tails

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

– augment training data [$a = 0 \dots 30$]
– train conditionally on smeared a
error bar from sampling a
⇒ Systematic/theory error bars



Uncertainties

Bayesian INN generator

- uncertainty over phase space
- training statistics leading source

$\Rightarrow$ Training-related error bars

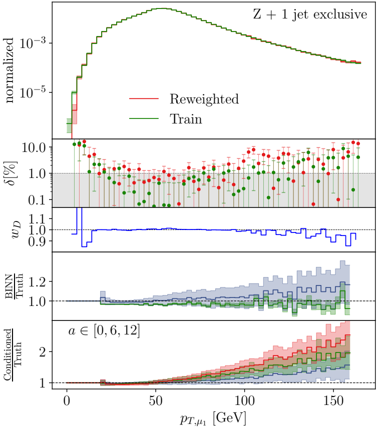
Theory uncertainties

- BNN regression/classification: systematics from data augmentation
- systematic uncertainties in tails

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

- augment training data [$a = 0 \dots 30$]
- train conditionally on smeared a error bar from sampling a

$\Rightarrow$ Systematic/theory error bars

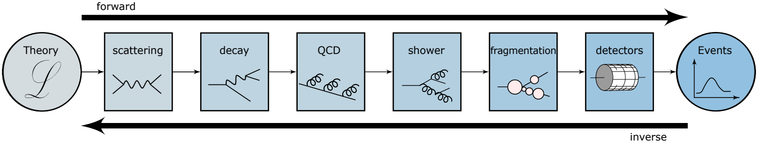


Inverting event simulations

Inverting LHC simulations

– unfolding QCD-shower to hard parton standard [jet algorithm]
unfolding detector common
unfolding top-quark decays useful
matrix element method complete unfolding

⇒ systematic approach through generative network



Conditional GAN

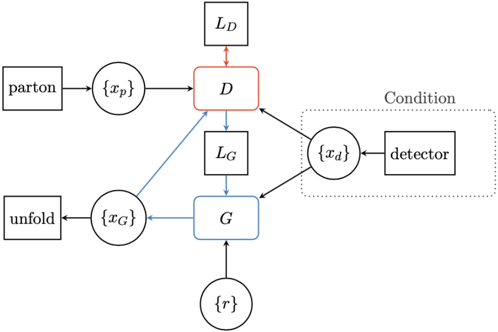
Goal: invert standard simulation

– detector simulation typical Monte Carlo, random-number-driven
– inversion possible, in principle [but entangled convolutions]
– generative network task

$$\text{partons} \xrightarrow{\text{DELPHES}} \text{detector} \xrightarrow{\text{GAN}} \text{partons}$$

Conditional generative networks

– random numbers to parton level
hadron level as condition
training on matched event pairs
– FCGAN the first example



Conditional INN

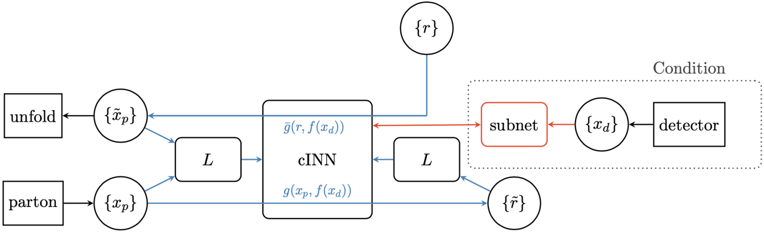
Statistical inversion

– task: construct parton-level pdf for (single) detector-level event

1- generative network: parton-level events from $\{r\}$

2- maximum likelihood loss

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



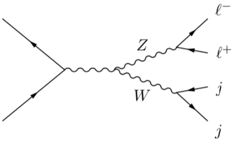
Conditional INN

Statistical inversion

– task: construct parton-level pdf for (single) detector-level event

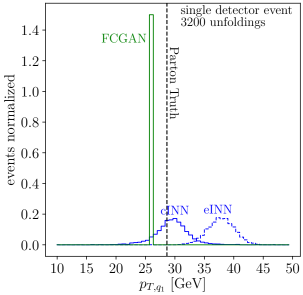
1- generative network: parton-level events from $\{r\}$

2- maximum likelihood loss



This time $pp \to ZW \to (\ell\ell)\ (jj)$

– distribution: single pair (x_p, x_d), 3200 unfoldings



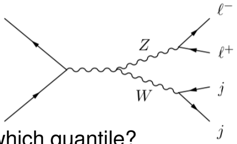
Conditional INN

Statistical inversion

– task: construct parton-level pdf for (single) detector-level event

1- generative network: parton-level events from $\{r\}$

2- maximum likelihood loss

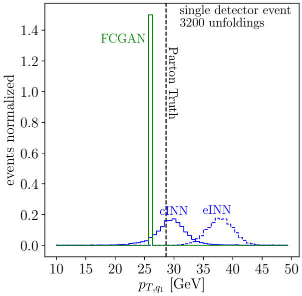


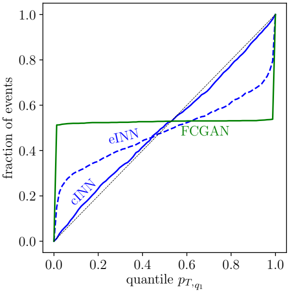
This time $pp \to ZW \to (\ell\ell)\ (jj)$

– distribution: single pair (x_p, x_d), 3200 unfoldings

– calibration: 1500 pairs (x_p, x_d), 60 unfoldings, truth in which quantile?

$\Rightarrow$ Conditional INNs solve inverse problems statistically





Machine learning for LHC theory

Machine learning for the LHC

– Classification/regression standard

  uncertainties?
  symmetries?
  experimental realities?

– GANs the cool kid

  generator producing best events
  discriminator checking generator
  limited in precision and uncertainty control

– INNs my work horse

  flow networks for control and precision
  Bayesian for error bars
  condition for inversion

– All results from 3 years, clearly a field for young people!