Precision Forward and Inverse Simulations

Tilman Plehn

Universität Heidelberg

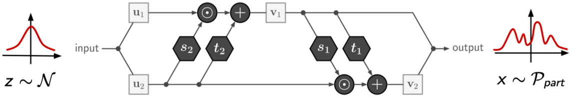
ATLAS, April 2022

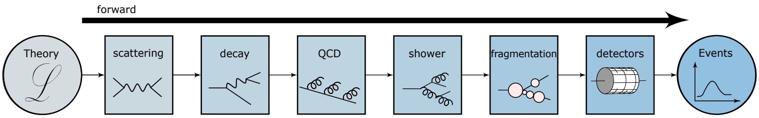
The bestest generative network

Normalizing flows — INN [Ardizzone, Köthe]

- Gaussian latent space
- bijective mapping
- known Jacobian
- log-likelihood loss

$\rightarrow$ Better than VAEs and GANs [for different opinion ask Daniel $\rightarrow$ OTUS]



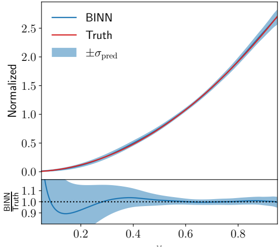


The bestest generative network

Normalizing flows — INN [Ardizzone, Köthe]

- Gaussian latent space
- bijective mapping
- known Jacobian
- log-likelihood loss

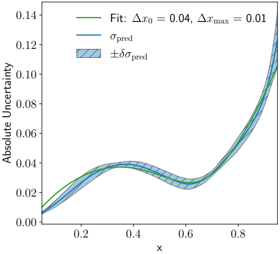
→ Better than VAEs and GANs [for different opinion ask Daniel —



Bayesian INNs [Bellagente, Haußmann, Luchmann, TP]

- network weight distributions [Gal (2016)]
- sample for output [efficient ensembling]
- working for regression, classification
- events with error bars [density & uncertainty maps]
- 2D: wedge ramp, kicker ramp,...

→ INNs just fits



Precision generator

ML-event generators [Butter, Plehn..., CaloFlow: Krause & Shih]

- useful ML-playground
- training from event samples
  no momentum conservation
  no detector effects [Fastsim easy to include]

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 [Butter, Plehn..., CaloFlow: Krause & Shih]

- useful ML-playground
- training from event samples
  no momentum conservation
  no detector effects [Fastsim easy to include]

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]

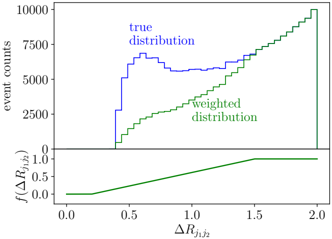
- challenging ΔR_{jj} features
- opposite of importance sampling

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

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

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

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



Precision generator

ML-event generators [Butter, Plehn..., CaloFlow: Krause & Shih]

- useful ML-playground
- training from event samples
  no momentum conservation
  no detector effects [Fastsim easy to include]

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

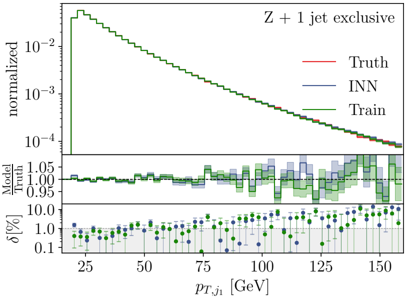
$$w^{(1\text{-jet})} = 1$$

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

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

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

$\to$ Per-cent precision in reach



Precision generator

ML-event generators [Butter, Plehn..., CaloFlow: Krause & Shih]

- useful ML-playground
- training from event samples
  no momentum conservation
  no detector effects [Fastsim easy to include]

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

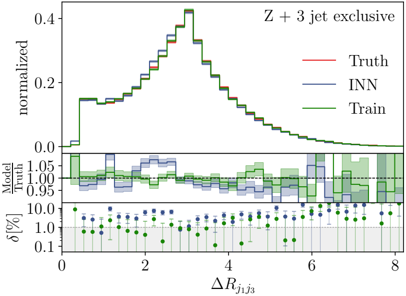
$$w^{(\text{1-jet})} = 1$$

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

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

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

$\to$ Per-cent precision in reach



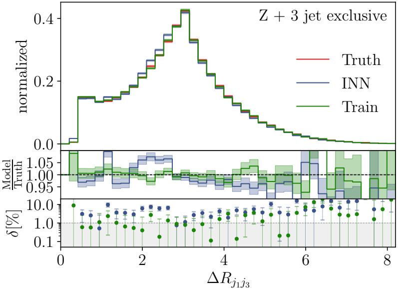
Controlled precision generator

ML-event generators

- useful ML-playground
  transferable to detector simulation
  needed for inverse simulations
- training from event samples
  no detector effects [Fastsim easy to include]

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]



Controlled precision generator

ML-event generators

- useful ML-playground
  transferable to detector simulation
  needed for inverse simulations
- training from event samples
  no detector effects [Fastsim easy to include]

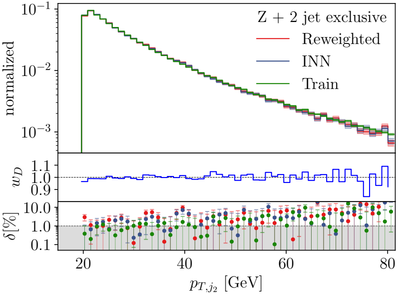
1- top-quark pairs $t\bar{t} \to 6$ jets [resonance peaks]

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

Discriminator: training vs generated

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

$\to$ Control & reweight

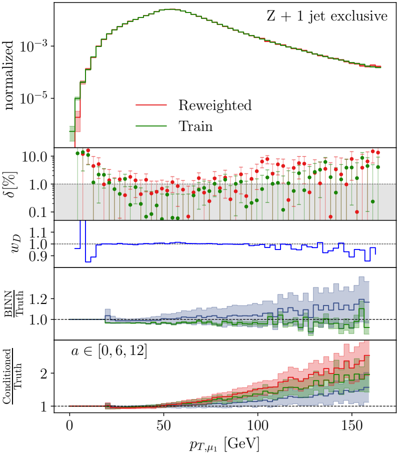


Uncertain precision generator

Bayesian INN generator

- learned uncertainty over phase space
- useful after control step
- low statistics means large uncertainty

→ Training-related error bars

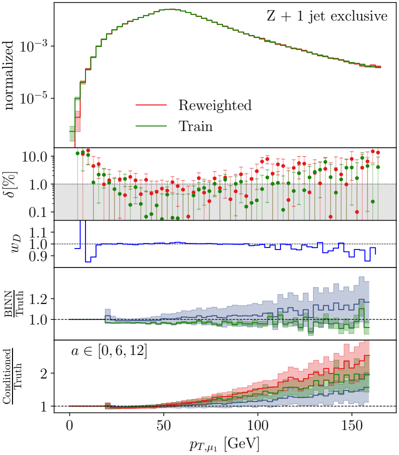


Uncertain precision generator

Bayesian INN generator

- learned uncertainty over phase space
- useful after control step
- low statistics means large uncertainty

$\to$ Training-related error bars



Theory uncertainties

- 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 a
- uncertainty from sampling a

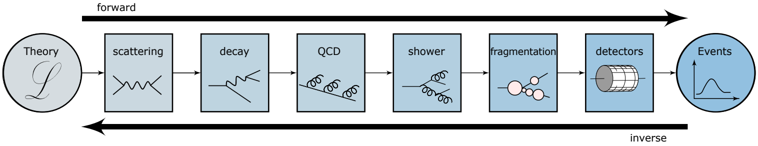
$\to$ Network for LHC standards

Systematic inverse simulation

Invertible ML-simulation [orthogonal technique to Omnifold]

- detector unfolding known problem [needed for global analyses]
- QCD parton from jet algorithm standard
- jet radiation combinatorics challenge
- decays established by top groups [needed for global analyses]
- matrix element method an old dream

→ Free choice of data-theory inference point



Systematic inverse simulation

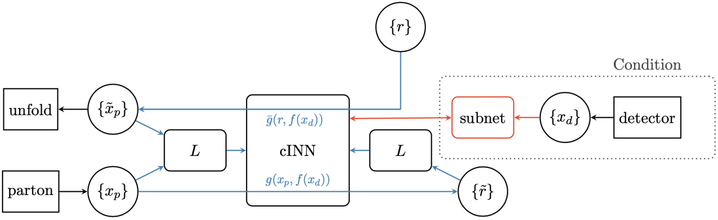
Invertible ML-simulation [orthogonal technique to Omnifold]

- detector unfolding known problem [needed for global analyses]
- QCD parton from jet algorithm standard
- jet radiation combinatorics challenge
- decays established by top groups [needed for global analyses]
- matrix element method an old dream

→ Free choice of data-theory inference point

Conditional INN [Bellagente, Butter, Kasieczka, TP, Rousselot, Winterhalder, Ardizzone, Köthe]

- partonic events from $\{r\}$, given detector event



Systematic inverse simulation

Invertible ML-simulation [orthogonal technique to Omnifold]

- detector unfolding known problem [needed for global analyses]
- QCD parton from jet algorithm standard
- jet radiation combinatorics challenge
- decays established by top groups [needed for global analyses]
- matrix element method an old dream

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

Conditional INN [Bellagente, Butter, Kasieczka, TP, Rousselot, Winterhalder, Ardizzone, Köthe]

- partonic events from $\{r\}$, given detector event
- maximum likelihood loss

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

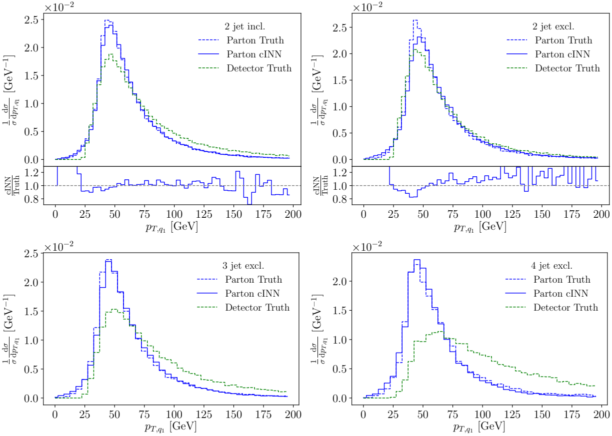
- eventually to be combined with reweighting

$\to$ Stable and statistically calibrated

Inverting to hard process

Undo QCD jet radiation

- nasty jet combinatorics, little information
- detector level: $pp \to ZW$+jets [variable number of objects]
- hard process given, ME vs PS jets from network



Inverting to hard process

Undo QCD jet radiation

- nasty jet combinatorics, little information
- detector level: $pp \to ZW$+jets [variable number of objects]
- hard process given, ME vs PS jets from network

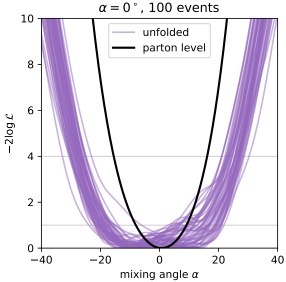
Matrix element method [Butter, Heimel, Martini, Peitzsch, TP (soon)]

- parameter likelihood from parton-level events

$$\mathcal{L}(\theta) = \prod_{i=1}^{N} p(\vec{x}^{(i)}|\theta) = \prod_{i=1}^{N} \frac{1}{\sigma_{\text{fid}}(\theta)} \int d^m z \, \frac{d^m\sigma(\theta)}{dz_1 \dots dz_m} \, T(\vec{x}^{(i)}, \vec{z})$$

$$T(\vec{x}, \vec{z}) = p_{\text{INN}}(\vec{z}|\vec{x})\epsilon(\vec{x}) \qquad \Rightarrow \qquad \mathcal{L}(\theta) = \prod_{i=1}^{N} \frac{\epsilon(\vec{x}^{(i)})}{\sigma_{\text{fid}}(\theta)} \int d^m z \, \frac{d^m\sigma(\theta)}{dz_1 \dots dz_m} \, p_{\text{INN}}(\vec{z}|\vec{x}^{(i)})$$

$$= \prod_{i=1}^{N} \frac{\epsilon(\vec{x}^{(i)})}{\sigma_{\text{fid}}(\theta)} \left\langle \frac{d^m\sigma(\theta)}{dz_1 \dots dz_m} \right\rangle_{\vec{z}\sim p_{\text{INN}}}$$



ML for LHC Theory

ML-applications in LHC physics

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

  ...improve established tasks
  ...develop new tools for established tasks
  ...transform through new ideas
- comprehensive unfolding possible

→ Let's make a difference!

Contents

1	Introduction	4
2	Machine Learning in event generators	5
2.1	Phase space sampling	6
2.2	Scattering Amplitudes	7
2.3	Loop integrals	9
2.4	Parton shower	10
2.5	Parton distribution functions	11
2.6	Fragmentation functions	12
3	End-to-end ML-generators	13
3.1	Fast generative networks	13
3.2	Control and precision	15
4	Inverse simulations and inference	16
4.1	Particle reconstruction	17
4.2	Detector unfolding	17
4.3	Unfolding to parton level	19
4.4	MadMiner	20
4.5	Matrix element method	22
5	Synergies, transparency and reproducibility	23
6	Outlook	24
	References	25

Machine Learning and LHC Event Generation

Anja Butter[1,2], Tilman Plehn[1], Steffen Schumann[3] (Editors),
Simon Badger[4], Sascha Caron[5, 6] Kyle Cranmer[7,8], Francesco Armando Di Bello[9],
Etienne Dreyer[10], Stefano Forte[11], Sanmay Ganguly[12], Dorival Gonçalves[13], Eilam Gross[10],
Theo Heimel[1], Gudrun Heinrich[14], Lukas Heinrich[15], Alexander Held[16], Stefan Höche[17],
Jessica N. Howard[18], Philip Ilten[19], Joshua Isaacson[17], Timo Janßen[3], Stephen Jones[20],
Marumi Kado[9,21], Michael Kagan[22], Gregor Kasieczka[23], Felix Kling[24], Sabine Kraml[25],
Claudius Krause[26], Frank Krauss[20], Kevin Kröninger[27], Rahool Kumar Barman[13],
Michel Luchmann[1], Vitaly Magerya[14], Daniel Maitre[20], Bogdan Malaescu[2],
Fabio Maltoni[28,29], Till Martini[30], Olivier Mattelaer[28], Benjamin Nachman[31,32],
Sebastian Pitz[1], Juan Rojo[33,34], Matthew Schwartz[35], David Shih[25], Frank Siegert[36],
Roy Stegeman[11], Bob Stienen[5], Jesse Thaler[37], Rob Verheyen[38], Daniel Whiteson[18],
Ramon Winterhalder[28], and Jure Zupan[19]

Abstract

First-principle simulations are at the heart of the high-energy physics research program. They link the vast data output of multi-purpose detectors with fundamental theory predictions and interpretation. This review illustrates a wide range of applications of modern machine learning to event generation and simulation-based inference, including conceptional developments driven by the specific requirements of particle physics. New ideas and tools developed at the interface of particle physics and machine learning will improve the speed and precision of forward simulations, handle the complexity of collision data, and enhance inference as an inverse simulation problem.

Submitted to the Proceedings of the US Community Study
on the Future of Particle Physics (Snowmass)

Inverting to QCD

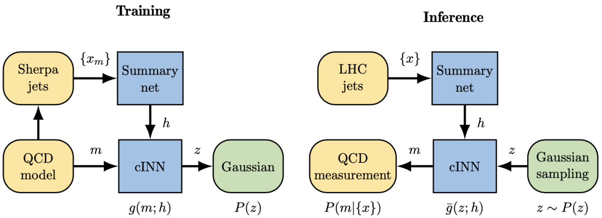
cINN for inference [Bieringer, Butter, Heimel, Höche, Köthe, TP, Radev]

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