Invertible Networks

ErUM-Data Community Meeting 1.7.21

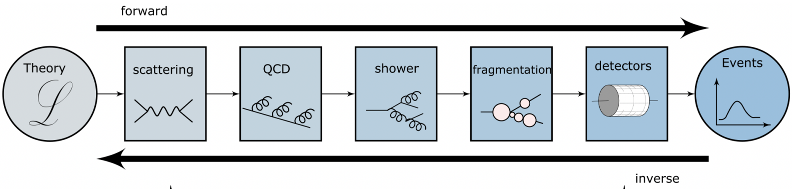
Anja Butter

ITP, Universität Heidelberg

arXiv:2006.06685
with M. Bellagente, G. Kasieczka, T. Plehn, A. Rousselot,

R. Winterhalder, L. Ardizzone, U. Köthe

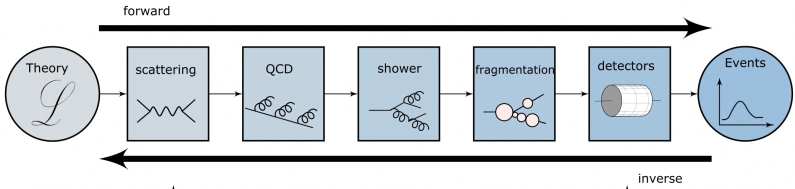
Can we invert the simulation chain?



What we
want to know

What we
measure or simulate

Can we invert the simulation chain?

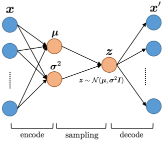


What we
want to know

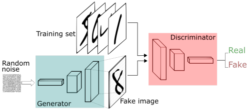
What we
measure or simulate

$\rightarrow$ Get multidimensional probability distribution for each event
$\rightarrow$ Need generative networks

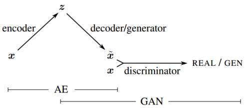
Neural network based generative networks



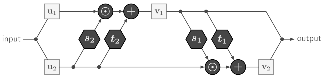
VAE



GAN



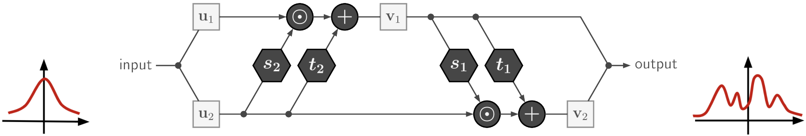
hybrid models



INN/NF

Invertible networks

[1808.04730] L. Ardizzone, J. Kruse, S. Wirkert, D. Rahner, E. W. Pellegrini, R. S. Klessen, L. Maier-Hein, C. Rother, U. Köthe



+ Arbitrary networks s and t
+ Fast evaluation in both directions
+ Simple Jacobian → Control over phase space density

Applications

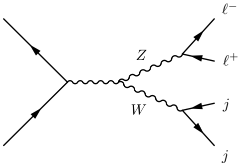
$$r \sim \mathcal{N} \underset{\leftarrow \text{ inversion:} \bar{g}}{\overset{\textsc{Event generator:}g\rightarrow}{\longleftrightarrow}} x \sim \mathcal{M}(r)$$

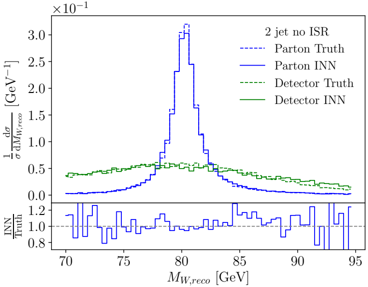
$$\left(x_{\text{parton}}\right) \underset{\leftarrow \text{ inversion:} \bar{g}}{\overset{\textsc{Shower/Detector:}g\rightarrow}{\longleftrightarrow}} \left(x_{\text{detector}}\right)$$

$$\left(x_{\text{model param}}\right) \underset{\leftarrow \text{ inversion:} \bar{g}}{\overset{\textsc{LHC simulation:}g\rightarrow}{\longleftrightarrow}} \left(x_{\text{detector}}\right)$$

Inverting detector effects

- $pp \to ZW \to (ll)(jj)$
- Train: parton $\to$ detector
- Evaluate: parton $\leftarrow$ detector





multi-dimensional ✓ bin independent ✓ only for deterministic process

Including the probabilistic aspect

$$\left(x_{part}\right) \xleftrightarrow[\leftarrow \text{ inversion:}\bar{g}]{\textsc{Pythia},\textsc{Delphes}:g\rightarrow} \left(x_{det}\right)$$

$$\text{with } \mathcal{L} = \mathcal{L}_{part} + \mathcal{L}_{det}$$

Including the probabilistic aspect

$$\begin{pmatrix} x_{part} \\ r_{part} \end{pmatrix} \xleftrightarrow[\leftarrow \text{ inversion:}\bar{g}]{\text{Pythia,Delphes:}g\rightarrow} \begin{pmatrix} x_{det} \\ r_{det} \end{pmatrix}$$

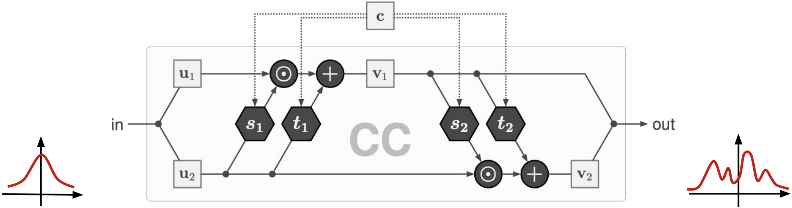
$$\text{with } \mathcal{L} = \mathcal{L}_{part} + \mathcal{L}_{det} + \mathcal{L}_r$$

Arbitrary choice of $\mathcal{L}_{part}, \mathcal{L}_{det}, \mathcal{L}_r$ determines (un)supervised training

Conditional INN

Rephrasing the problem

Given detector level information [$\rightarrow$ condition c]
$\rightarrow$ What is the probability density at parton level?



How to train the network

→ Training: Maximize posterior over model parameters

$$L = -\langle \log p(\theta|y) \rangle_{y\sim data}$$

$$= -\langle \log p(y|\theta) \rangle_{y\sim data} - \log p(\theta) + \text{const.} \quad \leftarrow \text{Bayes}$$

$$= -\left\langle \log p(g_\theta(y)) + \log \left| \frac{\partial g_\theta(y)}{\partial y} \right| \right\rangle - \log p(\theta) \leftarrow \text{change of var}$$

$$= \left\langle 0.5||g(y)||_2^2 - \log \left| \frac{\partial g_\theta(y)}{\partial y} \right| \right\rangle_{y\sim data} - \log p(\theta)$$

Gaussian latent variable
$g(y) \to \mathcal{N}$

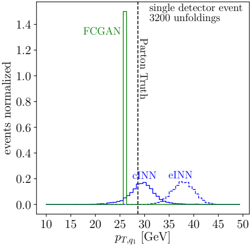
Jacobian

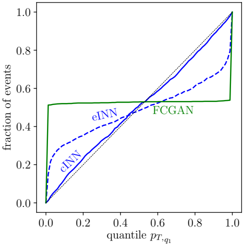
Regularization
$||\theta||^2$

cINN result for calibration

$$x_p \overset{g(x_p, \color{red}{f(x_d)}) \rightarrow}{\underset{\leftarrow \text{ unfolding: } \bar{g}(r, \color{red}{f(x_d)})}{\longleftrightarrow}} r$$

$$\text{Minimizing } L = \left\langle 0.5 ||\bar{g}(x_p, f(x_d)))||_2^2 - \log |J| \right\rangle_{x_p \sim P_p, x_d \sim P_d} - \log p(\theta)$$

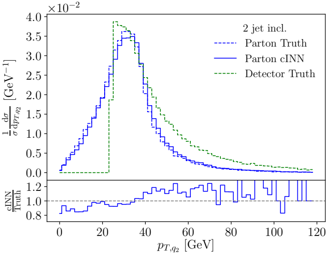




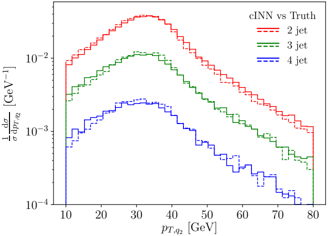
multi-dimensional ✓ bin independent ✓ statistically well defined ✓

$pp > WZ > q\bar{q}l^+l^- + \text{ISR}$

Train on inclusive dataset



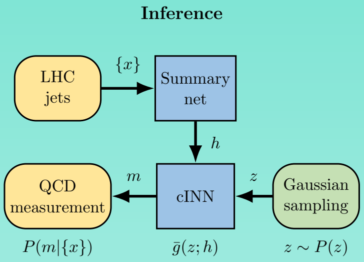
Evaluate
exclusive 2/3/4 jet channels



Going beyond unfolding

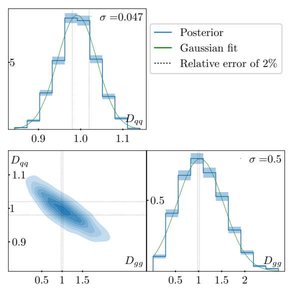
Same principle for inference

Measure parton shower parameters



Infere splitting kernels

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



Invertible networks

- Fast evaluation and tractable Jacobian
- Trainable on samples as well as densities
- Invertible networks for deterministic mapping
- cINN guaranties correct calibration in probabilistic mapping
- Simultaneous unfolding of different exclusive channels