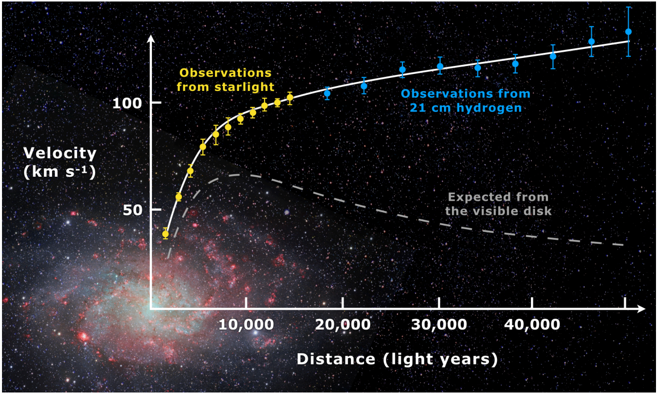
Simulating and unfolding LHC events with generative networks

LPNHE seminar

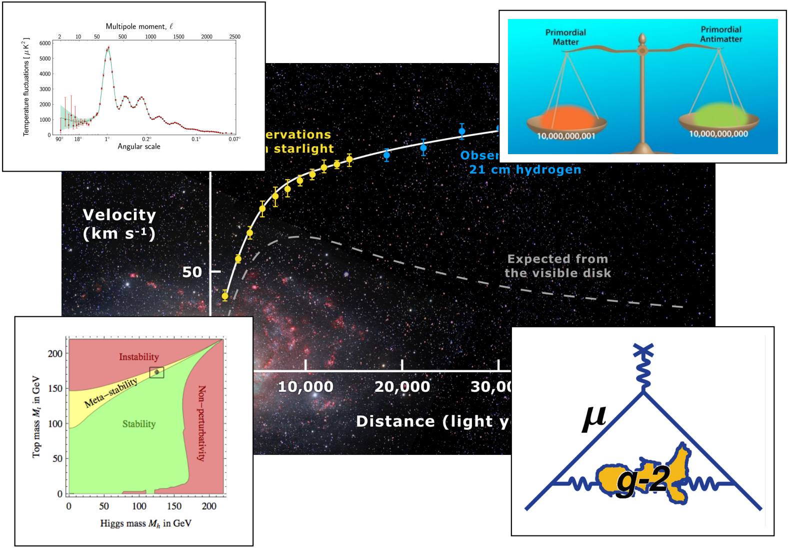
Anja Butter

ITP, Universität Heidelberg

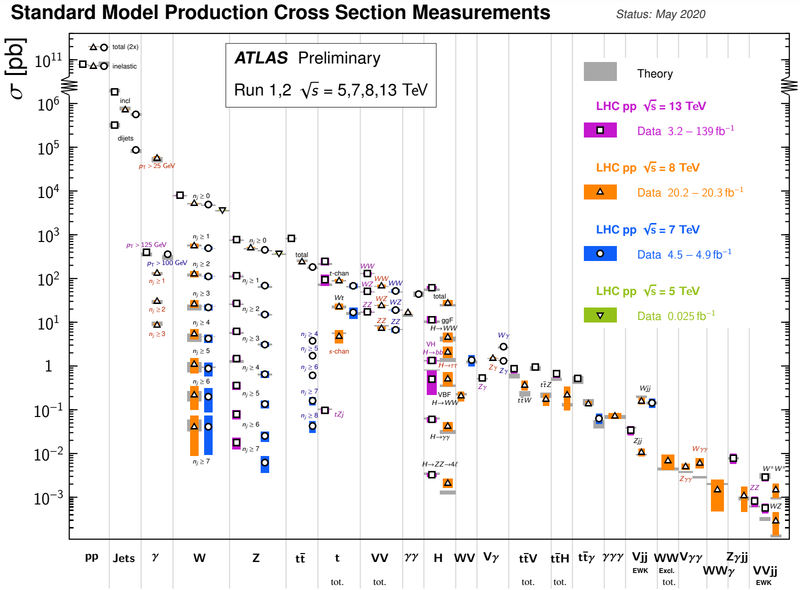
The need for new physics



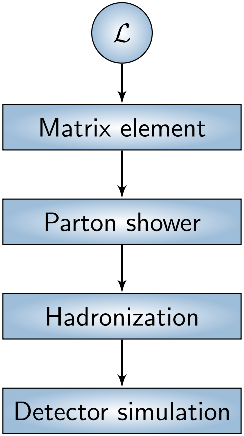
The need for new physics

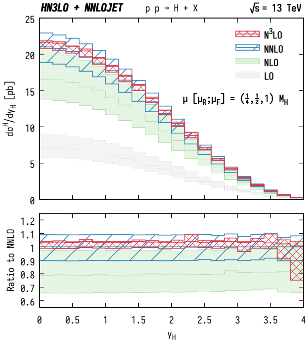


Era of data


Standard Model Production Cross Section Measurements
Status: May 2020
ATLAS Preliminary
Run 1,2 $\sqrt{s}$ = 5,7,8,13 TeV
σ [pb]
Theory
LHC pp $\sqrt{s}$ = 13 TeV
Data 3.2 – 139 fb^{-1}
LHC pp $\sqrt{s}$ = 8 TeV
Data 20.2 – 20.3 fb^{-1}
LHC pp $\sqrt{s}$ = 7 TeV
Data 4.5 – 4.9 fb^{-1}
LHC pp $\sqrt{s}$ = 5 TeV
Data 0.025 fb^{-1}
pp
Jets
γ
W
Z
$t\bar{t}$
t
VV
$\gamma\gamma$
H
WV
Vγ
$t\bar{t}$V
$t\bar{t}$H
$t\bar{t}\gamma$
$\gamma\gamma\gamma$
Vjj EWK
WW Excl.
V$\gamma\gamma$
WWγ
Zγjj
VVjj EWK
tot.

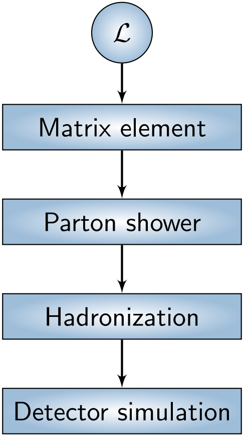
Precision simulations with limited resources

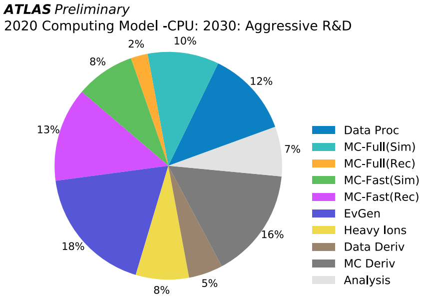




[1807.11501] Cieri, Chen, Gehrmann, Glover, Huss

Precision simulations with limited resources

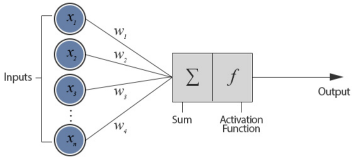




Speed = Precision

How can ML help analyzing data

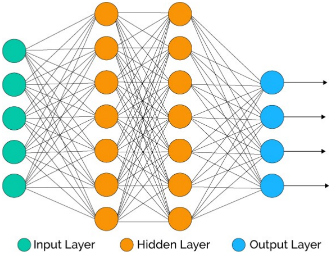
- 1.0 Classification/Regression
  → Label data, eg. Signal vs Background



minimize $L = (y_{true} - y_{output})^2$

How can ML help analyzing data

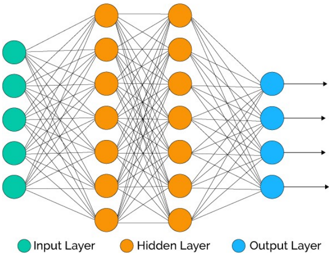
- 1.0 Classification/Regression
  $\rightarrow$ Label data, eg. Signal vs Background



minimize $L = (y_{true} - y_{output})^2$

How can ML help analyzing data

- 1.0 Classification/Regression
  $\rightarrow$ Label data, eg. Signal vs Background

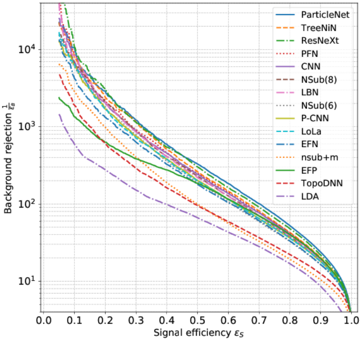


minimize $L = (y_{true} - y_{output})^2$

$+$ low level observables
$+$ efficient training

Why now? $\rightarrow$ GPUs
$\rightarrow$ new algorithms [convolutional networks]

Comparative top tagging study



[1707.08966] G. Kasieczka, et al.

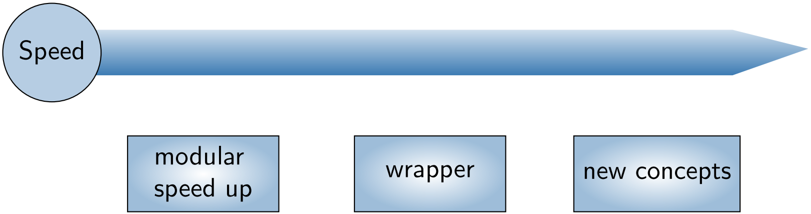
→ Other applications: jet calibration, particle identification, ...

→ Open questions: precision, uncertainties, visualization

How can ML help increasing precision

- ML 2.0 Generative models
  - → Can we simulate new data?

Fast evaluation → more events = higher order → Precision



Boosting standard event generation...

1. Generate phase space points

2. Calculate event weight

$$w_{event} = f(x_1, Q^2) f(x_2, Q^2) \times \mathcal{M}(x_1, x_2, p_1, \ldots p_n) \times J(p_i(r))^{-1}$$

3. Unweighting via importance sampling
   $\rightarrow$ optimal for $w \approx 1$

Boosting standard event generation...

Matrix element

$$w_{event} = f(x_1, Q^2)f(x_2, Q^2) \times \mathcal{M}(x_1, x_2, p_1, \ldots p_n) \times J(p_i(r))^{-1}$$

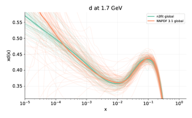
PDF

Phase space mapping

Boosting standard event generation...

Matrix element

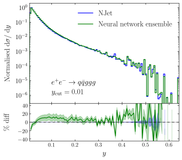
$$w_{event} = f(x_1, Q^2) f(x_2, Q^2) \times \mathcal{M}(x_1, x_2, p_1, \ldots p_n) \times J(p_i(r))^{-1}$$



- NNPDF since 2002(!)
- S. Carrazza, J. Cruz-Martinez [1907.05075]

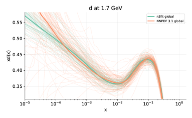
Phase space mapping

Boosting standard event generation...



- Amplitude estimation
- S. Badger, J. Bullock [2002.07516]
- J. Bendavid [1707.00028]

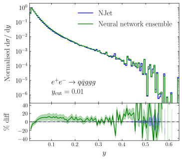
$$w_{event} = f(x_1, Q^2) f(x_2, Q^2) \times \mathcal{M}(x_1, x_2, p_1, \ldots p_n) \times J(p_i(r))^{-1}$$



- NNPDF since 2002(!)
- S. Carrazza, J. Cruz-Martinez [1907.05075]

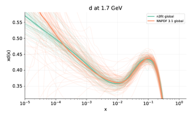
Phase space mapping

Boosting standard event generation...

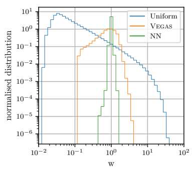


- Amplitude estimation
- S. Badger, J. Bullock [2002.07516]
- J. Bendavid [1707.00028]

$$w_{event} = f(x_1, Q^2) f(x_2, Q^2) \times \mathcal{M}(x_1, x_2, p_1, \dots p_n) \times J(p_i(r))^{-1}$$



- NNPDF since 2002(!)
- S. Carrazza, J. Cruz-Martinez [1907.05075]



- Learn phase space mapping ($\to w \approx 1$)
- Gao et al. [2001.10028]
- Bothmann et al. [2001.05478]

... or training directly on event samples

Event generation

- Generating 4-momenta
- $Z > ll$, $pp > jj$, $pp > t\bar{t}$+decay

[1901.00875] Otten et al. **VAE** & **GAN**

[1901.05282] Hashemi et al. **GAN**

[1903.02433] Di Sipio et al. **GAN**

[1903.02556] Lin et al. **GAN**

[1907.03764, 1912.08824] Butter et al. **GAN**

[1912.02748] Martinez et al. **GAN**

[2001.11103] Alanazi et al. **GAN**

[2011.13445] Stienen et al. **NF**

[2012.07873] Backes et al. **GAN**

[2101.08944] Howard et al. **VAE**

Detector simulation

- Jet images
- Fast calorimeter simulation

[1701.05927] de Oliveira et al. **GAN**

[1705.02355, 1712.10321] Paganini et al. **GAN**

[1802.03325, 1807.01954] Erdmann et al. **GAN**

[1805.00850] Musella et al. **GAN**

[ATL-SOFT-PUB-2018-001, ATLAS-SIM-2019-004, ATL-SOFT-PROC-2019-007] ATLAS **VAE** & **GAN**

[1909.01359] Carazza and Dreyer **GAN**

[1912.06794] Belayneh et al. **GAN**

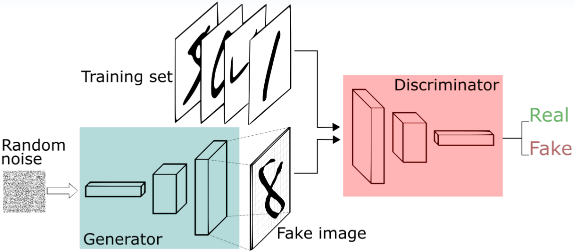
[2005.05334, 2102.12491] Buhmann et al. **VAE**

[2009.03796] Diefenbacher et al. **GAN**

[2009.14017] Lu et al.

NO claim to completeness!

Generative Adversarial Networks



Discriminator $[D(x_T) \to 1,\ D(x_G) \to 0]$

$$L_D = \Big\langle -\log D(x) \Big\rangle_{x\sim P_{Truth}} + \Big\langle -\log(1 - D(x)) \Big\rangle_{x\sim P_{Gen}} \to -2\log 0.5$$

Generator $[D(x_G) \to 1]$

$$L_G = \Big\langle -\log D(x) \Big\rangle_{x\sim P_{Gen}}$$

$\Rightarrow$ **Nash Equilibrium**

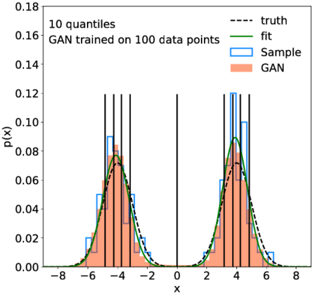
$\Rightarrow$ **New statistically independent samples**

What is the statistical value of GANned events?[2008.06545]

- Camel function
- Sample vs. GAN vs. 5 param.-fit

Evaluation on quantiles:

$$\text{MSE}^* = \sum_{j=1}^{N_\text{quant}} \left(p_j - \frac{1}{N_\text{quant}} \right)^2$$

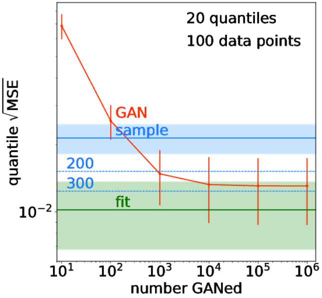


What is the statistical value of GANned events?[2008.06545]

- Camel function
- Sample vs. GAN vs. 5 param.-fit

Evaluation on quantiles:

$$\text{MSE}^* = \sum_{j=1}^{N_\text{quant}} \left(p_j - \frac{1}{N_\text{quant}} \right)^2$$

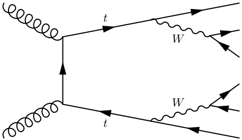


→ Amplification factor 2.5

Sparser data → bigger amplification

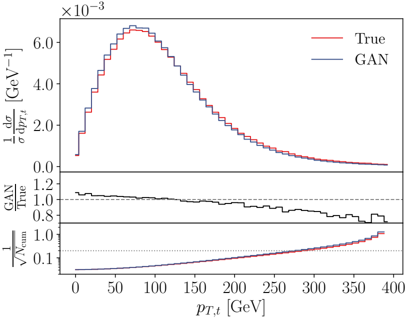
How to GAN LHC events [1907.03764]

- $t\bar{t} \to 6$ quarks
- 18 dim output
  - external masses fixed
  - no momentum conservation

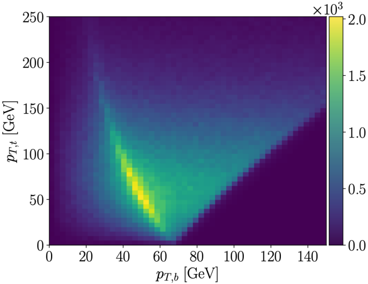


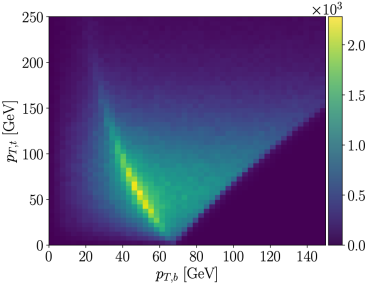
+ Flat observables ✓

– Systematic undershoot in tails [10-20% deviation]

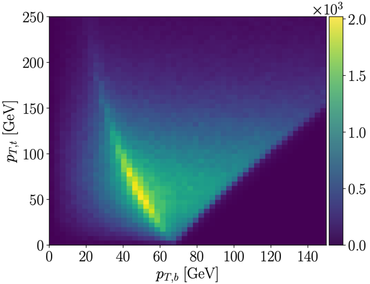


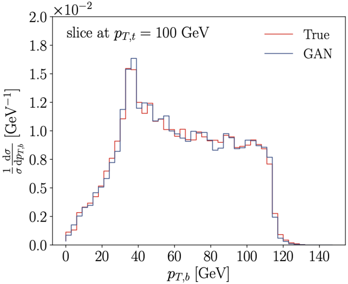
Correlations





Correlations





✓

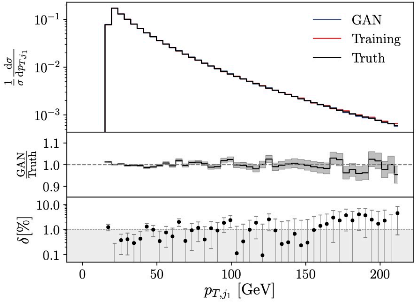
Reaching precision (preliminary)

1. Representation p_T, η, ϕ
2. Momentum conservation
3. Resolve log p_T
4. Regularization: spectral norm
5. Batch information

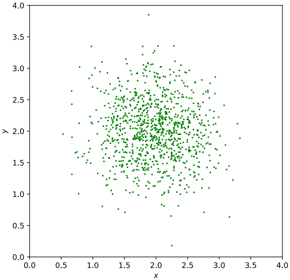
$\rightarrow$ 1% precision ✓

Next step automization

W + 2 jets

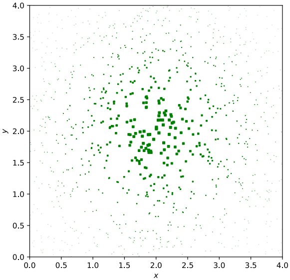


Information in distributions



Information in space distribution
(what we want)

=



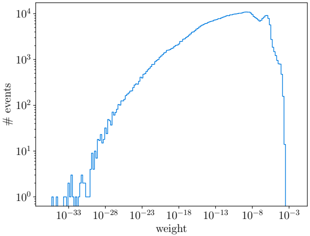
Information in weight
(what we have)

The unweighting bottleneck

- High-multiplicity / higher-order $\rightarrow$ unweighting efficiencies $< 1\%$

$\rightarrow$ Simulate conditions with naive Monte Carlo generator
ME by `Sherpa`, parton densities from `LHAPDF`, Rambo-on-diet

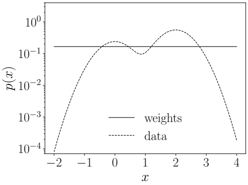
$pp \rightarrow \mu^+\mu^-$ with $m_{\mu\mu} > 50$ GeV



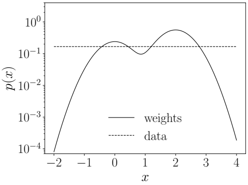
$\rightarrow$ unweighting efficieny 0.2%

Training on weighted events

Information contained in distribution or event weights

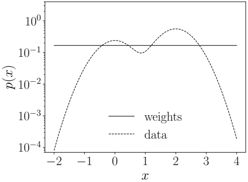


Train on
weighted events



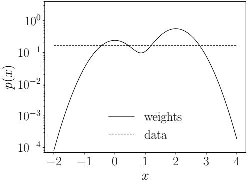
Training on weighted events

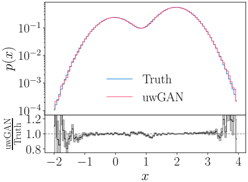
Information contained in distribution or event weights

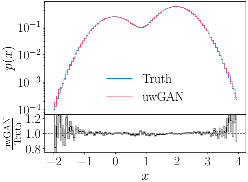


Train on weighted events

Generate unweighted events



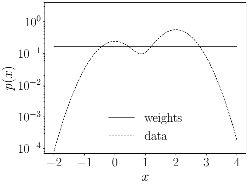


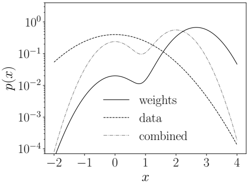


$$L_D = \Big\langle -w \log D(x) \Big\rangle_{x \sim P_{Truth}} + \Big\langle -\log(1 - D(x)) \Big\rangle_{x \sim P_{Gen}}$$

Training on weighted events

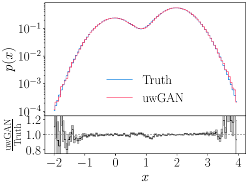
Information contained in distribution or event weights

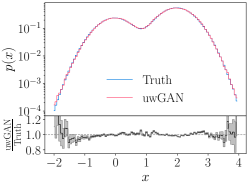




Train on weighted events

Generate unweighted events

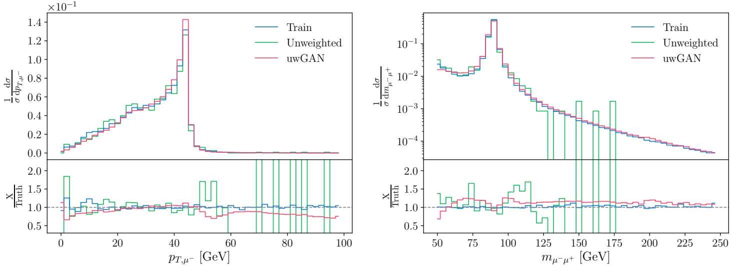




$$L_D = \Big\langle - w \log D(x) \Big\rangle_{x \sim P_{Truth}} + \Big\langle - \log(1 - D(x)) \Big\rangle_{x \sim P_{Gen}}$$

normalizing flow: B. Stienen, R. Verheyen [2011.13445]

uwGAN results



Populates high energy tails

Large amplification wrt. unweighted data!

Short summary

We can ..

→ use GANs to learn event distributions and correlations

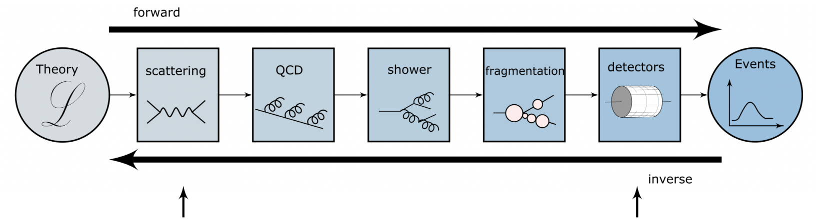
→ amplify underlying statistics

→ achieve precision

→ train directly on weighted events

→ boost precision simulations with generative networks

Can we invert the simulation chain?



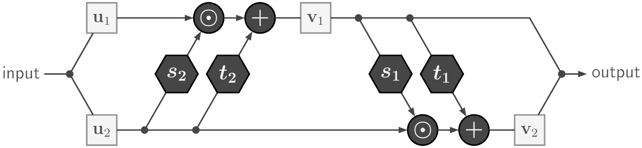
What we
want to know

What we
measure or simulate

wish list: ☐ multi-dimensional
☐ bin independent
☐ statistically well defined

Invertible networks

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



[1808.04730] L. Ardizzone, J. Kruse, S. Wirkert, D. Rahner,

E. W. Pellegrini, R. S. Klessen, L. Maier-Hein, C. Rother, U. Köthe

+ Bijective mapping

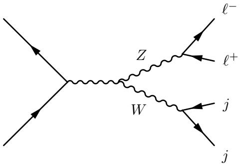
+ Tractable Jacobian

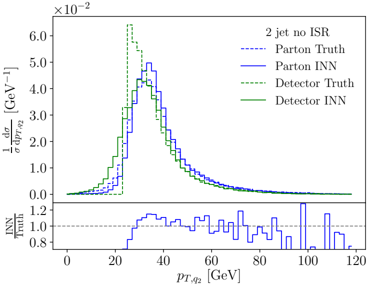
+ Fast evaluation in both directions

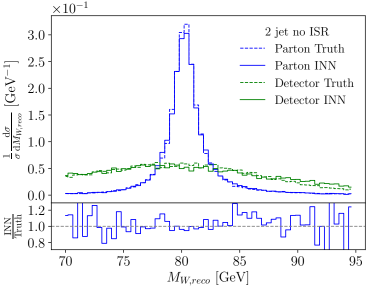
+ Arbitrary networks s and t

Inverting detector effects

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





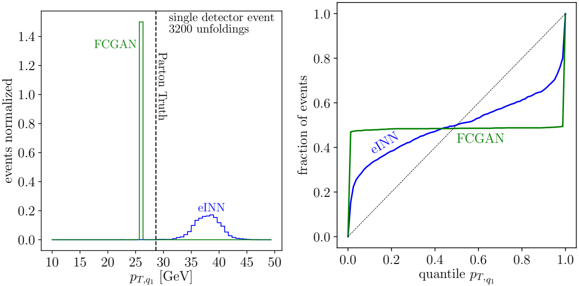


multi-dimensional ✓ bin independent ✓ statistically well defined ?

Including stochastical effects

$$\begin{pmatrix} x_p \\ r_p \end{pmatrix} \xleftrightarrow[\leftarrow \text{ unfolding:}\bar{g}]{\text{PYTHIA,DELPHES:}g\rightarrow} \begin{pmatrix} x_d \\ r_d \end{pmatrix}$$

Sample r_d for fixed detector event
How often is Truth included in distribution quantile?

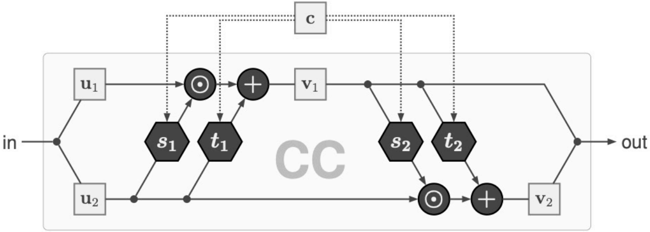


- Problem: arbitrary balance of many loss functions

Taking a different angle

Given an event x_d, what is the probability distribution at parton level?
$\rightarrow$ sample over r, condition on x_d

$$x_p \xleftrightarrow[\leftarrow \text{ unfolding: } \bar{g}(r, f(x_d))]{g(x_p, f(x_d)) \rightarrow} r$$



Taking a different angle

Given an event x_d, what is the probability distribution at parton level?
$\rightarrow$ sample over r, condition on x_d

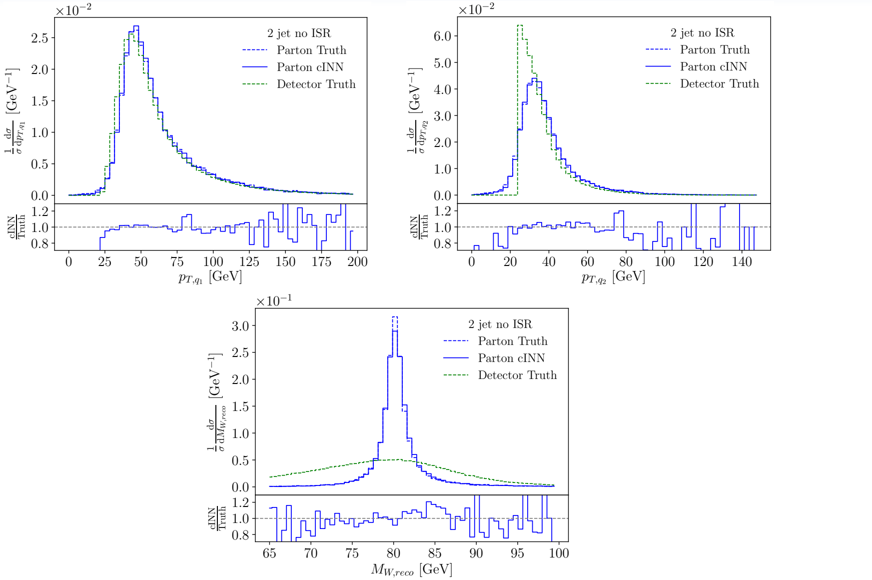
$$x_p \xleftrightarrow[\leftarrow \text{ unfolding: } \bar{g}(r, f(x_d))]{g(x_p, f(x_d)) \rightarrow} r$$

$\rightarrow$ Training: Maximize posterior over model parameters

$$\begin{aligned}
L &= -\left\langle \log p(\theta|x_p, x_d) \right\rangle_{x_p \sim P_p, x_d \sim P_d} \\
&= -\left\langle \log p(x_p|\theta, x_d) \right\rangle_{x_p \sim P_p, x_d \sim P_d} - \log p(\theta) + \text{const.} \quad \leftarrow \text{Bayes} \\
&= -\left\langle \log p(\bar{g}(x_p, x_d)) + \log \left| \frac{\partial \bar{g}(x_p, x_d)}{\partial x_p} \right| \right\rangle - \log p(\theta) \leftarrow \text{change of var} \\
&= \left\langle 0.5 ||\bar{g}(x_p, f(x_d))||_2^2 - \log |J| \right\rangle_{x_p \sim P_p, x_d \sim P_d} - \log p(\theta)
\end{aligned}$$

$\rightarrow$ Jacobian of bijective mapping

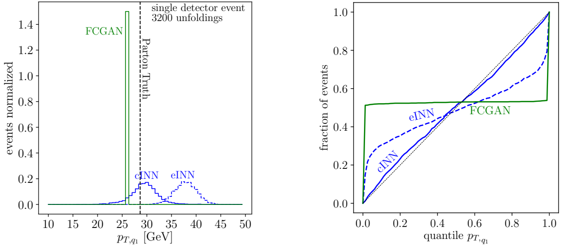
Cross check distributions



Condition INN on detector data [2006.06685]

$$x_p \xleftrightarrow[\leftarrow \text{unfolding: } \bar{g}(r, f(x_d))]{g(x_p, f(x_d)) \rightarrow} r$$

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



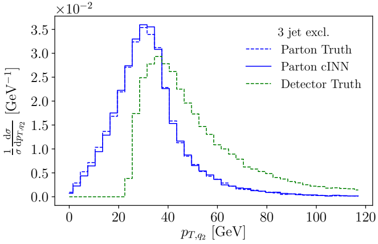
multi-dimensional ✓ bin independent ✓ statistically well defined ✓

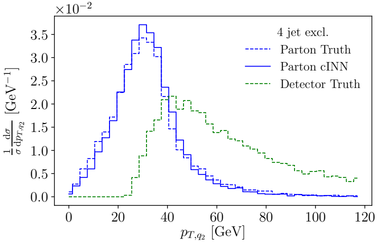
Inverting the full event I

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

$\rightarrow$ ISR leads to large fraction of 2/3/4 jet events

- Train and test on exclusive channels

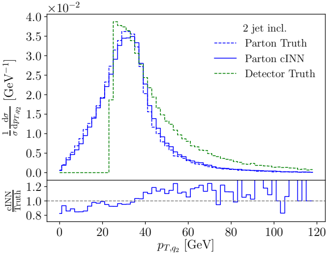




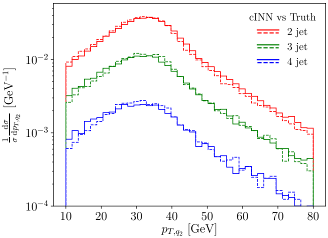
Inverting the full event II

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

Train on inclusive dataset



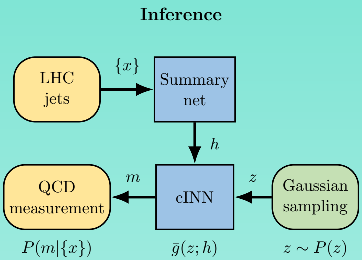
Evaluate
exclusive 2/3/4 jet channels



Going beyond unfolding

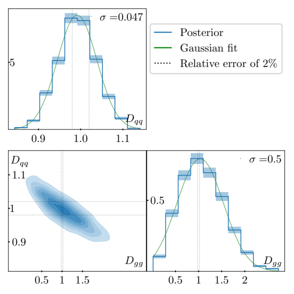
Same principle for inference

Measure parton shower parameters



Infere splitting kernels

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



We can use ML ...

... to enable precision simulations in forward direction

... to turn weighted into unweighted events

... to invert the simulation chain statistically

... for fun and precision :)