# Generative and Invertible Networks for LHC Theory

Tilman Plehn

Universität Heidelberg

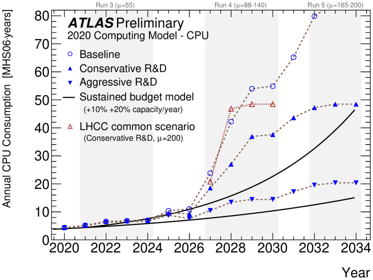
Physics Meets ML 3/2021

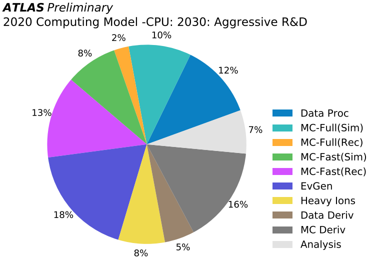
# Challenges towards HL-LHC

Paradigm shift: model searches ⟶ fundamental understanding of data

– precision QCD
– precision simulations
– precision measurements

⇒ Nothing fundamental without simulations [not even unsupervised...]



# Challenges towards HL-LHC

## Paradigm shift: model searches $\longrightarrow$ fundamental understanding of data

– precision QCD
– precision simulations
– precision measurements
$\Rightarrow$ Nothing fundamental without simulations [not even unsupervised...]

## 10-year HL-LHC requirements

– simulated event numbers $\sim$ expected events [factor 25 for HL-LHC]
– general move to NLO/NNLO [1%-2% error]
– higher relevant multiplicities [jet recoil, extra jets, WBF, etc.]
– new low-rate high-multiplicity backgrounds
– cutting-edge predictions not through generators [N$^3$LO in Pythia?]

# Challenges towards HL-LHC

## Paradigm shift: model searches $\longrightarrow$ fundamental understanding of data

– precision QCD
– precision simulations
– precision measurements
$\Rightarrow$ Nothing fundamental without simulations [not even unsupervised...]

## 10-year HL-LHC requirements

– simulated event numbers $\sim$ expected events [factor 25 for HL-LHC]
– general move to NLO/NNLO [1%-2% error]
– higher relevant multiplicities [jet recoil, extra jets, WBF, etc.]
– new low-rate high-multiplicity backgrounds
– cutting-edge predictions not through generators [N$^3$LO in Pythia?]

## Three ways to use ML

– improve current tools: iSherpa, ML-MadGraph, etc
– new tools: ML-generator-networks
– conceptual ideas in theory simulations and analyses

# Generative networks

## GANGogh [Bonafilia, Jones, Danyluk (2017)]

– neural network: learned function $f(x)$ [regression, classification]

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

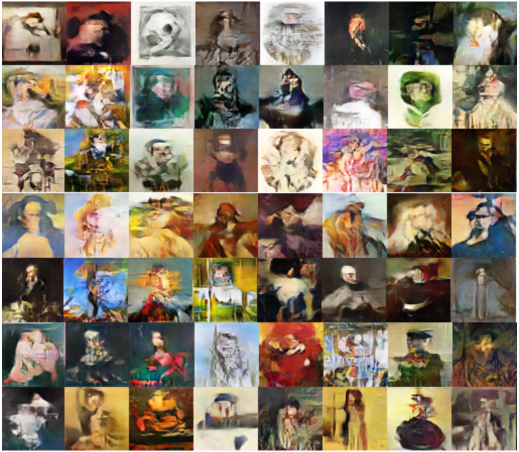
– train on 80,000 pictures [organized by style and genre]

– generate flowers

# Generative networks

## GANGogh [Bonafilia, Jones, Danyluk (2017)]

– neural network: learned function $f(x)$ [regression, classification]
– 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 networks

## GANGogh [Bonafilia, Jones, Danyluk (2017)]

– neural network: learned function $f(x)$ [regression, classification]
– can networks create new pieces of art?
  map random numbers to image pixels?
– train on 80,000 pictures [organized by style and genre]

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

– trained on 15,000 portraits
– sold for $432.500
⇒ ML all marketing and sales

# Generative networks

## GANGogh [Bonafilia, Jones, Danyluk (2017)]

– neural network: learned function $f(x)$ [regression, classification]

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

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

## 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)]

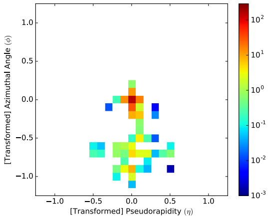
– calorimeter or jet images
sparsity the technical challenge

1- reproduce valid jet images from training data

2- organize them by QCD vs $W$-decay jets

– high-level observables $m, \tau_{21}$ as check

$\Rightarrow$ GANs generating QCD jets



# GAN algorithm

## Generating events [phase space positions, possibly with weights]

- training: true events $\{x_T\}$
  output: generated events $\{r\} \to \{x_G\}$
- discriminator constructing $D(x)$ by minimizing [classifier $D(x) = 1, 0$ true/generator]

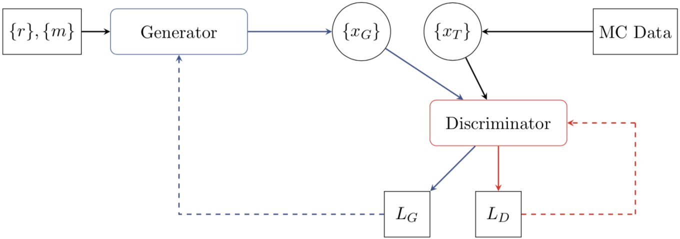
$$L_D = \Big\langle -\log D(x) \Big\rangle_{x_T} + \Big\langle -\log(1 - D(x)) \Big\rangle_{x_G}$$

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

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

- equilibrium $D = 0.5 \;\Rightarrow\; L_D = L_G = -\log 0.5$

$\Rightarrow$ statistically independent copy of training events



# GAN algorithm

## Generating events [phase space positions, possibly with weights]

- training: true events $\{x_T\}$
  output: generated events $\{r\} \to \{x_G\}$
- discriminator constructing $D(x)$ by minimizing [classifier $D(x) = 1, 0$ true/generator]
- generator constructing $r \to x_G$ by minimizing [$D$ needed]

$\Rightarrow$ statistically independent copy of training events

## Generative network studies

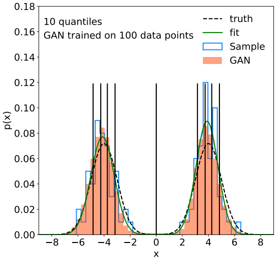
- Jets [de Oliveira (2017), Carrazza-Dreyer (2019)]
- Detector simulations [Paganini (2017), Musella (2018), Erdmann (2018), Ghosh (2018), Buhmann (2020)]
- Events [Otten (2019), Hashemi, DiSipio, Butter (2019), Martinez (2019), Alanazi (2020), Chen (2020), Kansal (2020)]
- Unfolding [Datta (2018), Omnifold (2019), Bellagente (2019), Bellagente (2020), Vandegar (2020), Howard (2020)]
- Templates for QCD factorization [Lin (2019)]
- EFT models [Erbin (2018)]
- Event subtraction [Butter (2019)]
- Phase space [Bothmann (2020), Gao (2020), Klimek (2020)]
- Basics [GANplification (2020), DCTR (2020)]
- Unweighting [Verheyen (2020), Backes (2020)]
- Superresolution [DiBello (2020), Baldi (2020)]

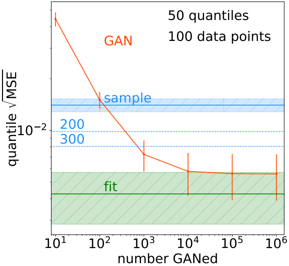
# GANplification

## Gain beyond training data [Butter, Diefenbacher, Kasieczka, Nachman, TP]

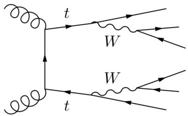
- true function known
  compare GAN vs sampling vs fit
- quantiles with $\chi^2$-values
- fit like 500-1000 sampled points
  GAN like 500 sampled points [amplifictation factor 5]
  requiring 10,000 GANned events
- interpolation and resolution the key [NNPDF]

$\Rightarrow$ GANs beyond training data



# 1– How to GAN LHC events

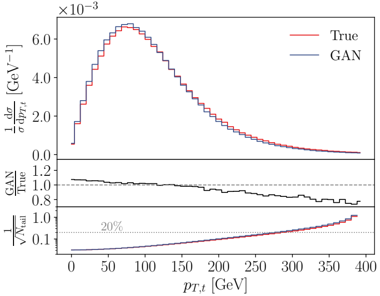



## Idea: replace ME for hard process [Butter, TP, Winterhalder]

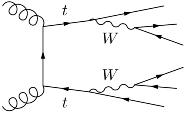
– medium-complex final state $t\bar{t} \to 6$ jets
  $t/\bar{t}$ and $W^\pm$ on-shell with BW $6 \times 4 = 18$ dof
  on-shell external states $\to$ 12 dof [constants hard to learn]
  parton level, because it is harder

– flat observables flat [phase space coverage okay]

– standard observables with tails [statistical error indicated]

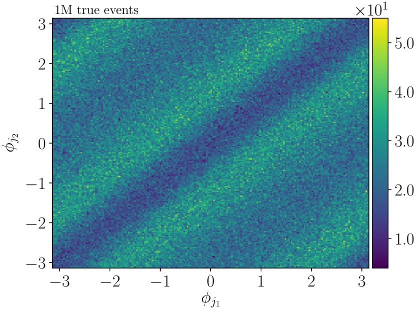


# 1– How to GAN LHC events

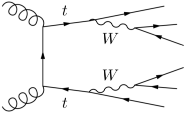



## Idea: replace ME for hard process [Butter, TP, Winterhalder]

- medium-complex final state $t\bar{t} \to 6$ jets
  $t/\bar{t}$ and $W^\pm$ on-shell with BW $6 \times 4 = 18$ dof
  on-shell external states $\to 12$ dof [constants hard to learn]
  parton level, because it is harder
- flat observables flat [phase space coverage okay]
- standard observables with tails [statistical error indicated]
- improved resolution [1M training events]



# 1– How to GAN LHC events



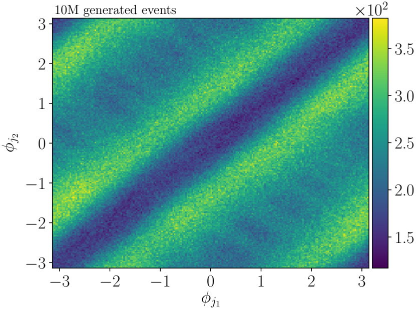

## Idea: replace ME for hard process [Butter, TP, Winterhalder]

– medium-complex final state $t\bar{t} \to 6$ jets
$t/\bar{t}$ and $W^{\pm}$ on-shell with BW $6 \times 4 = 18$ dof
on-shell external states $\to 12$ dof [constants hard to learn]
parton level, because it is harder

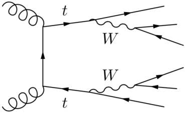
– flat observables flat [phase space coverage okay]

– standard observables with tails [statistical error indicated]

– improved resolution [10M generated events]



# 1– How to GAN LHC events




## Idea: replace ME for hard process [Butter, TP, Winterhalder]

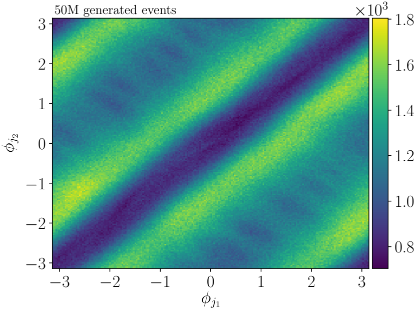
– medium-complex final state $t\bar{t} \to 6$ jets
  $t/\bar{t}$ and $W^{\pm}$ on-shell with BW $6 \times 4 = 18$ dof
  on-shell external states $\to$ 12 dof [constants hard to learn]
  parton level, because it is harder

– flat observables flat [phase space coverage okay]

– standard observables with tails [statistical error indicated]

– improved resolution [50M generated events]

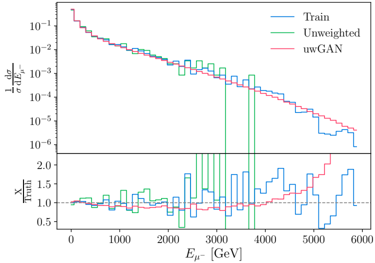
– Forward simulation working



# Statistical bonus: unweighting

## Gaining beyond GANpliflication [Butter, TP, Winterhalder]

- phase space sampling: weighted events [PS weight $\times |\mathcal{M}|^2$]
  events: constant weights
- probabilistic unweighting weak spot of standard MC
- learn phase space patterns [density estimation]
  generate unweighted events [through loss function]
- compare training, GAN, classic unweighting



# Statistical bonus: subtraction

## Subtract samples without binning [Butter, TP, Winterhalder]

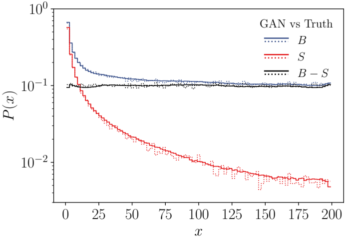
– statistical uncertainty

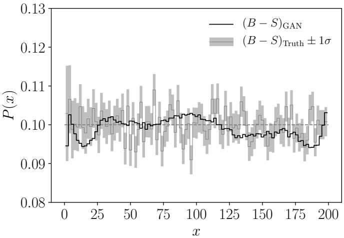
$$\Delta_{B-S} = \sqrt{\Delta_B^2 + \Delta_S^2} > \max(\Delta B, \Delta S)$$

– 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 [Butter, TP, Winterhalder]

- statistical uncertainty

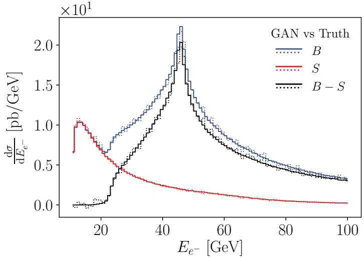
$$\Delta_{B-S} = \sqrt{\Delta_B^2 + \Delta_S^2} > \max(\Delta B, \Delta S)$$

- 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 [Butter, TP, Winterhalder]

– 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

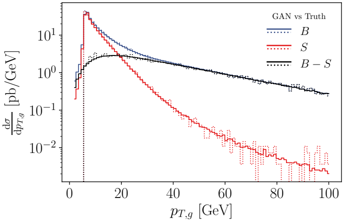
$$P_B(x) = \frac{1}{x} + 0.1 \qquad P_S(x) = \frac{1}{x} \qquad \Rightarrow \qquad P_{B-S} = 0.1$$

– 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)



# 2– How to GAN away detector effects

## Goal: invert Monte Carlo [Bellagente, Butter, Kasiczka, TP, Winterhalder]

– parton shower, detector simulation typical examples [drawing random numbers]

– inversion possible, in principle [entangled convolutions, model assumed]

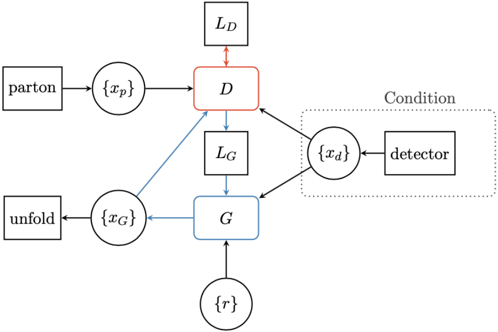
– GAN task

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

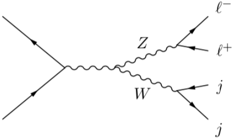
⇒ Full phase space unfolded

## Conditional GAN

– random numbers $\longrightarrow$ parton level
hadron level as condition
matched event pairs



# Detector unfolding

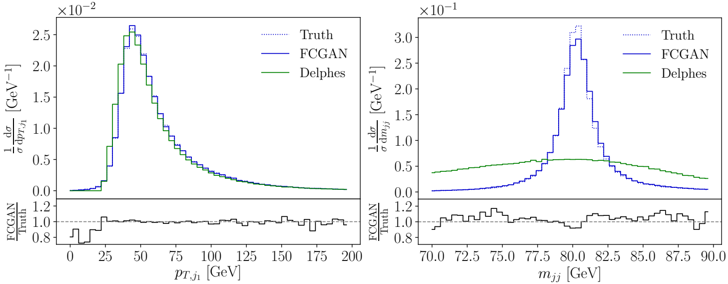



## Reference process $pp \to ZW \to (\ell\ell)\ (jj)$

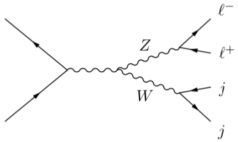
– broad $jj$ mass peak
  narrow $\ell\ell$ mass peak
  modified $2 \to 2$ kinematics
  fun phase space boundaries

– GAN same as event generation [with MMD]

## Model (in)dependence



# Detector unfolding




## Reference process $pp \to ZW \to (\ell\ell)\ (jj)$

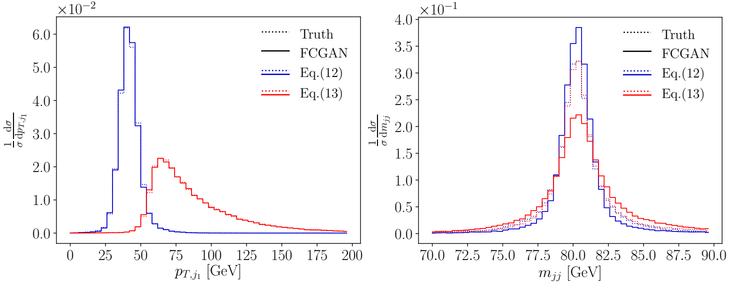
- broad $jj$ mass peak
  narrow $\ell\ell$ mass peak
  modified $2 \to 2$ kinematics
  fun phase space boundaries
- GAN same as event generation [with MMD]

## Model (in)dependence

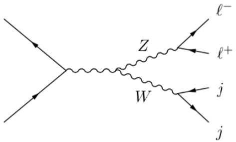
- detector-level cuts [14%, 39% events, no interpolation, MMD not conditional]

$$p_{T,j_1} = 30 \dots 50\ \text{GeV} \quad p_{T,j_2} = 30 \dots 40\ \text{GeV} \quad p_{T,\ell^-} = 20 \dots 50\ \text{GeV} \tag{12}$$

$$p_{T,j_1} > 60\ \text{GeV} \tag{13}$$



# Detector unfolding




## Reference process $pp \to ZW \to (\ell\ell)\,(jj)$

– broad $jj$ mass peak
  narrow $\ell\ell$ mass peak
  modified $2 \to 2$ kinematics
  fun phase space boundaries

– GAN same as event generation [with MMD]

## Model (in)dependence

– detector-level cuts [14%, 39% events, no interpolation, MMD not conditional]

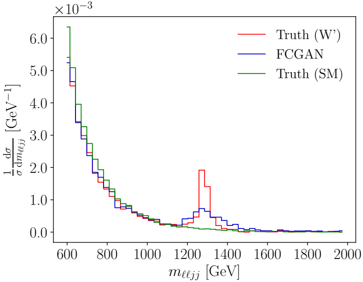
$$p_{T,j_1} = 30 \dots 50\ \text{GeV} \quad p_{T,j_2} = 30 \dots 40\ \text{GeV} \quad p_{T,\ell^-} = 20 \dots 50\ \text{GeV} \qquad (12)$$

$$p_{T,j_1} > 60\ \text{GeV} \qquad (13)$$

– model dependence [Thank you to BenN]

– train: SM events
  test: 10% events with $W'$ in $s$-channel

$\Rightarrow$ Working fine, but ill-defined



# 3– Proper inverting

## Invertible networks [Bellagente, Butter, Kasieczka, TP, Rousselot, Winterhalder, Ardizzone, Köthe]

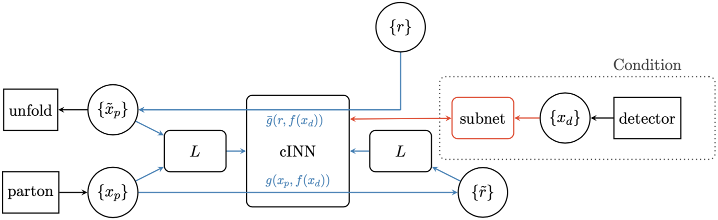
– network as bijective transformation — normalizing flow
  Jacobian tractable [specifically: coupling layer]
  evaluation in both directions — INN [Ardizzone, Rother, Köthe]

– standard setup, random-number-padded working like FCGAN

– conditional: parton-level events from $\{r\}$

– maximum likelihood loss

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



# 3– Proper inverting

## Invertible networks [Bellagente, Butter, Kasieczka, TP, Rousselot, Winterhalder, Ardizzone, Köthe]

– network as bijective transformation — normalizing flow
Jacobian tractable [specifically: coupling layer]
evaluation in both directions — INN [Ardizzone, Rother, Köthe]

– standard setup, random-number-padded working like FCGAN

– conditional: parton-level events from $\{r\}$

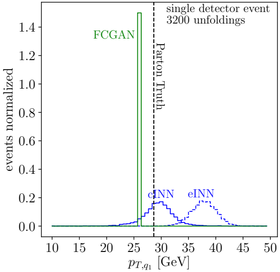
– maximum likelihood loss

## Again $pp \to ZW \to (\ell\ell)\ (jj)$

– performance on distributions like FCGAN

– parton-level probability distribution for single detector event

$\Rightarrow$ Well-defined statistical inversion



# 3– Proper inverting

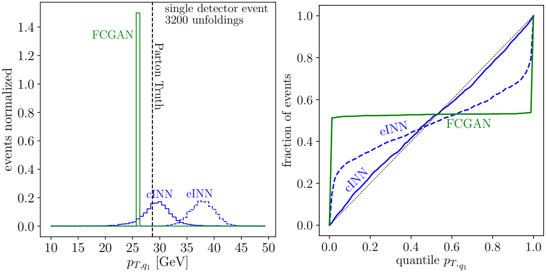
## Invertible networks [Bellagente, Butter, Kasieczka, TP, Rousselot, Winterhalder, Ardizzone, Köthe]

- network as bijective transformation — normalizing flow
  Jacobian tractable [specifically: coupling layer]
  evaluation in both directions — INN [Ardizzone, Rother, Köthe]
- standard setup, random-number-padded working like FCGAN
- conditional: parton-level events from $\{r\}$
- maximum likelihood loss

## Again $pp \to ZW \to (\ell\ell)\ (jj)$

- performance on distributions like FCGAN
- parton-level probability distribution for single detector event

$\Rightarrow$ Well-defined statistical inversion



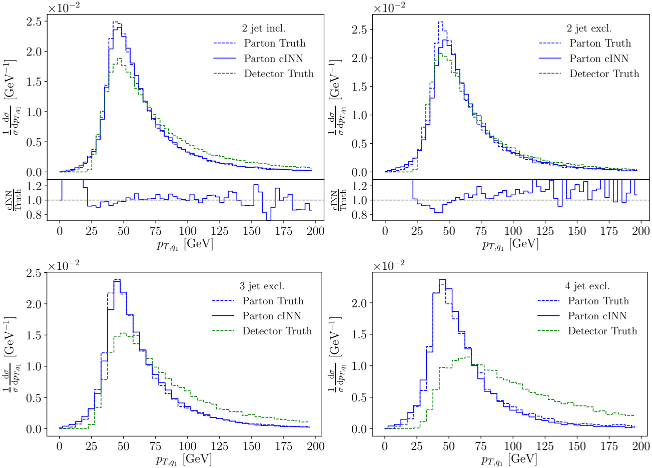
# 4– Inverting to hard process

## What theorists want: undo ISR

– detector-level process $pp \to ZW$+jets [variable number of objects]

– ME vs PS jets decided by network

– training jet-inclusively or jet-exclusively
parton-level hard process chosen $2 \to 2$



# 4– Inverting to hard process

## What theorists want: undo ISR

– detector-level process $pp \to ZW$+jets [variable number of objects]

– ME vs PS jets decided by network

– training jet-inclusively or jet-exclusively
parton-level hard process chosen $2 \to 2$

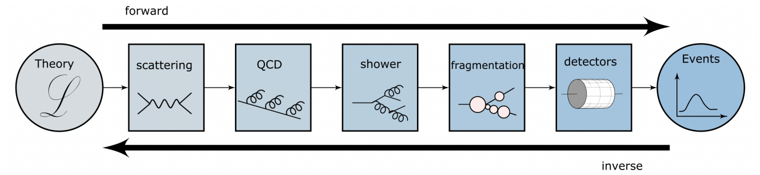
## Towards systematic inversion

– detector unfolding known problem

– QCD parton from jet algorithm standard

– jet radiation possible

$\Rightarrow$ Invertible simulation in reach



# 5– Inverting to measure

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

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

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

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

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

# 5– Inverting to measure

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

– condition jets with QCD parameters
  train model parameters $\longrightarrow$ Gaussian latent space
  test Gaussian sampling $\longrightarrow$ QCD parameter measurement

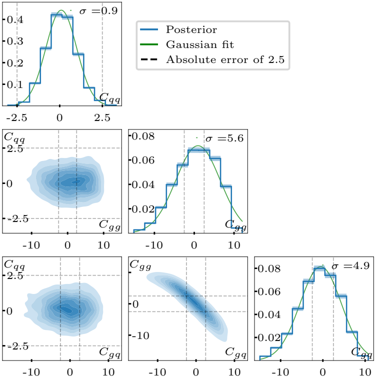
– going beyond $C_A$ vs $C_F$ [Kluth etal]

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

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

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

– idealized shower [Sherpa]



# 5– Inverting to measure

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

– condition jets with QCD parameters
  train model parameters $\longrightarrow$ Gaussian latent space
  test Gaussian sampling $\longrightarrow$ QCD parameter measurement

– going beyond $C_A$ vs $C_F$ [Kluth etal]

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

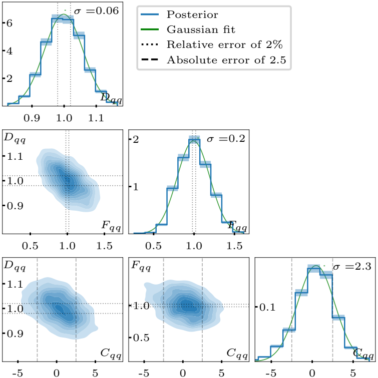
$$P_{gg} = 2C_A \left[ D_{gg} \left( \frac{z(1-y)}{1-z(1-y)} + \frac{(1-z)(1-y)}{1-(1-z)(1-y)} \right) + F_{gg} z(1-z) + C_{gg} yz(1-z) \right]$$

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

– idealized shower [Sherpa]

– reality hitting...

– More ML-opportunities...



# Machine learning for LHC theory

## HL-LHC data vs fundamental physics

- MC challenges

  higher-order precision in bulk
  coverage of tails
  unfolding to access fundamental QCD

- GANs the cool kid

  generator trying to produce best events
  discriminator trying to catch generator,

- INNs my theory hope

  flow networks for control
  condition for inversion
  Bayes for uncertainties

- Progress needs Professionals!