Generative and Invertible Networks for LHC Theory

Tilman Plehn

Universität Heidelberg

Oxford 10/2020

Big data for LHC

LHC: fundamental understanding of lots of data

– what do we really do?

1. theory framework [SMEFT, SUSY]
2. precision predictions [first-principles QFT]
3. compare simulated and measured events

USING NEURAL NETWORKS TO IDENTIFY JETS

Leif LÖNNBLAD*, Carsten PETERSON** and Thorsteinn RÖGNVALDSSON***

Department of Theoretical Physics, University of Lund, Sölvegatan 14A, S-22362 Lund, Sweden

Received 29 June 1990

A neural network method for identifying the ancestor of a hadron jet is presented. The idea is to find an efficient mapping between certain observed hadronic kinematical variables and the quark-gluon identity. This is done with a neuronic expansion in terms of a network of sigmoidal functions using a gradient descent procedure, where the errors are back-propagated through the network. With this method we are able to separate gluon from quark jets originating from Monte Carlo generated e^+e^- events with $\sim 85\%$ approach. The result is independent of the MC model used. This approach for isolating the gluon jet is then used to study the so-called string effect.

In addition, heavy quarks (b and c) in e^+e^- reactions can be identified on the 50% level by just observing the hadrons. In particular we are able to separate b-quarks with an efficiency and purity, which is comparable with what is expected from vertex detectors. We also speculate on how the neural network method can be used to disentangle different hadronization schemes by compressing the dimensionality of the state space of hadrons.

$\Rightarrow$ What can we learn from Google and Facebook?

Generative networks

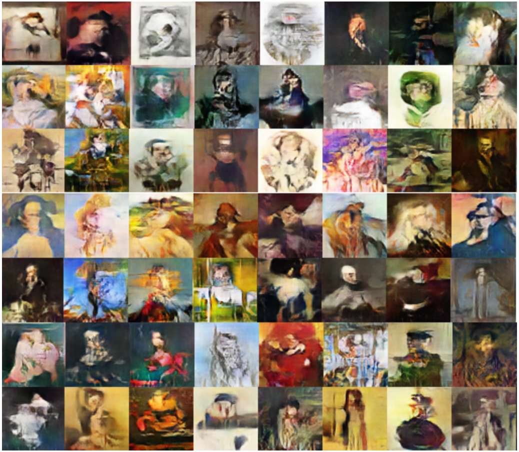
GANGogh [Bonafilia, Jones, Danyluk (2017)]

- 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]

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

$\Rightarrow$ ML all marketing and sales

GAN algorithm

Generating events

- 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]

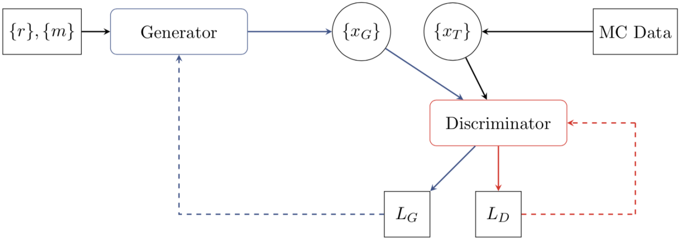
$$L_D = \Big\langle -\log D(x) \Big\rangle_{x_T} + \Big\langle -\log(1 - D(x)) \Big\rangle_{x_G}$$

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

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

- equilibrium $D = 0.5 \;\Rightarrow\; L_D = L_G = 1$

$\Rightarrow$ statistically independent copy of training events



GAN algorithm

Generating events

- 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]

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

$\Rightarrow$ statistically independent copy of training events

Generative network studies

- Jet Images [de Oliveira (2017), Carrazza-Dreyer (2019)]
- Detector simulations [Paganini (2017), Musella (2018), Erdmann (2018), Ghosh (2018), Buhmann (2020)]
- Events [Otten(2019), Hashemi (2019), Di Sipio (2019), Butter (2019), Martinez (2019), Alanazi (2020), Chen (2020)]
- Unfolding [Datta (2018), Omnifold (1911), Bellagente (2019), Bellagente (2020)]
- Templates for QCD factorization [Lin (2019)]
- EFT models [Erbin (2018)]
- Event subtraction [Butter (2019)]
- Sherpa [Bothmann (2020), Gao (2020)]
- Basics [GANplification (2020), DCTR (2020)]

1– Jet generation

GANGogh for jet images [de Oliveira, Paganini, Nachman]

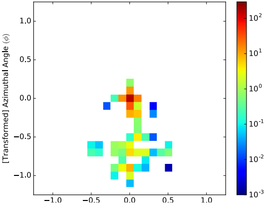
– start with calorimeter or jet images [η vs ϕ]
  sparsity the technical challenge [cf top tagging comparison]

1- reproduce valid jet images from training data

2- organize them by QCD vs W-decay jets

– high-level observables as check

$\Rightarrow$ Generating jets [Carrazza & Dreyer]



GAN questions

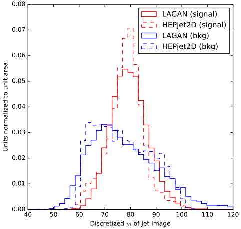
– use cases?

– control?

– uncertainty? [Bayesian networks?]

– gain?

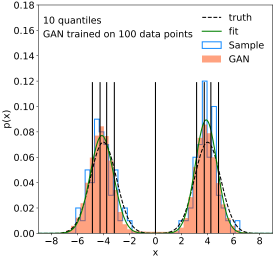
– achievable precision?

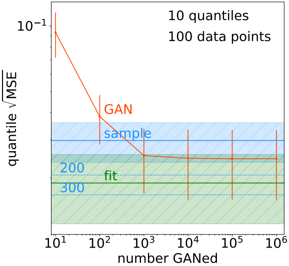


2– GANplification

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

- true function known
  compare GAN vs sampling vs fit
- χ^2-sum of quantiles

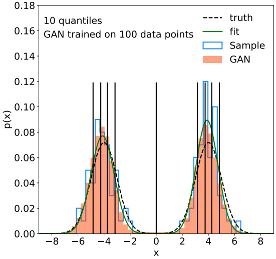


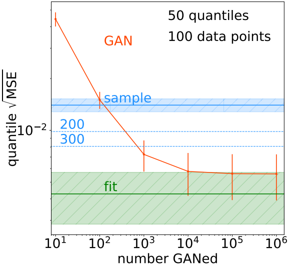


2– GANplification

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

– true function known
  compare GAN vs sampling vs fit
– χ^2-sum of quantiles
– fit like 500-1000 sampled points
  GAN like 500 sampled points [amplifictation factor 5]
  requiring 10,000 GANned events



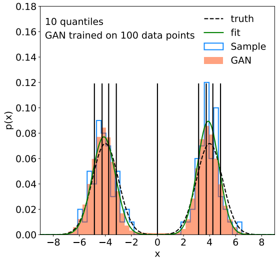


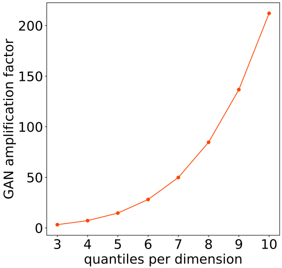
2– GANplification

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

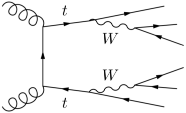
- true function known
  compare GAN vs sampling vs fit
- χ^2-sum of quantiles
- fit like 500-1000 sampled points
  GAN like 500 sampled points [amplifictation factor 5]
  requiring 10,000 GANned events
- 5-dimensional Gaussian shell
  sparsely populated
  amplification vs quantiles
- fit-like additional information
- interpolation and resolution the key [NNPDF]

$\Rightarrow$ GANs enhance training data



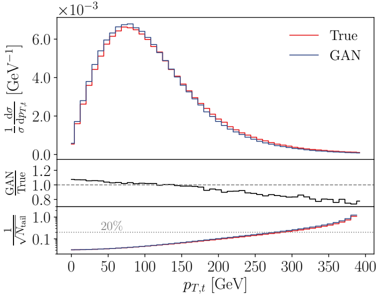


3– How to GAN LHC events

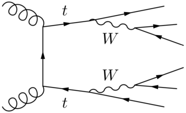


Idea: replace ME for hard process [Butter, TP, Winterhalder]

– 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]
– flat observables flat [phase space coverage okay]
– direct observables with tails [statistical error indicated]
– constructed observables similar



3– How to GAN LHC events



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

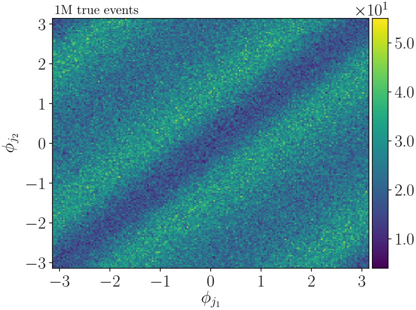
– medium-complex final state $t\bar{t} \to 6$ jets
  $t/\bar{t}$ and $W^\pm$ on-shell with BW $6 \times 4 = 18$ dof
  on-shell external states $\to 12$ dof [constants hard to learn]

– flat observables flat [phase space coverage okay]

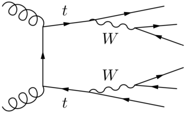
– direct observables with tails [statistical error indicated]

– constructed observables similar

– improved resolution [1M training events]



3– How to GAN LHC events



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

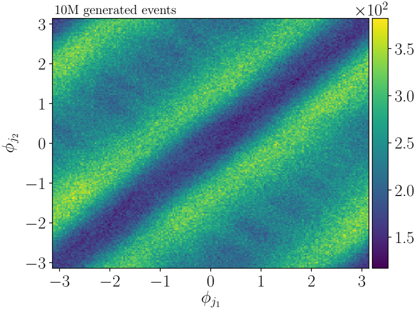
– medium-complex final state $t\bar{t} \to 6$ jets
  $t/\bar{t}$ and $W^\pm$ on-shell with BW $6 \times 4 = 18$ dof
  on-shell external states $\to 12$ dof [constants hard to learn]

– flat observables flat [phase space coverage okay]

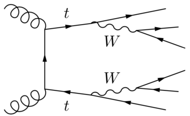
– direct observables with tails [statistical error indicated]

– constructed observables similar

– improved resolution [10M generated events]

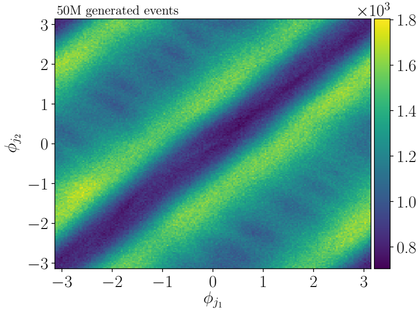


3– How to GAN LHC events



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]
– flat observables flat [phase space coverage okay]
– direct observables with tails [statistical error indicated]
– constructed observables similar
– improved resolution [50M generated events]
– Proof of concept



Chemistry of loss functions

GAN version of adaptive sampling

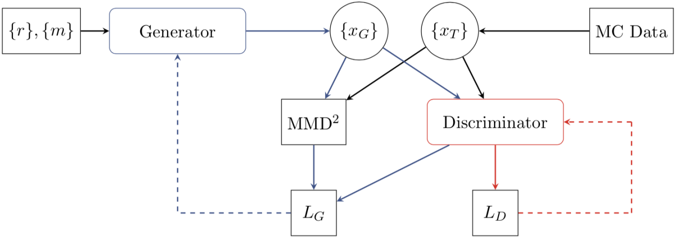
– generally 1D features

  phase space boundaries
  kinematic cuts
  invariant masses [top, W]

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

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

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



Chemistry of loss functions

GAN version of adaptive sampling

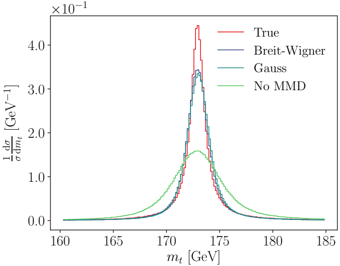
– generally 1D features

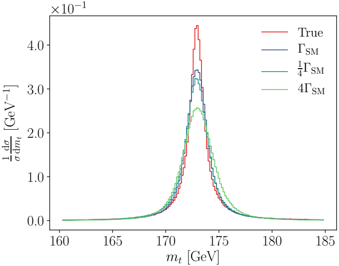
  phase space boundaries
  kinematic cuts
  invariant masses [top, W]

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

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

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

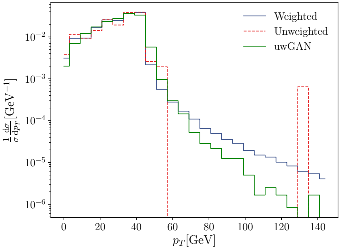




Unfolding

Gaining beyond GANpliflication [Butter, TP, Winterhalder]

- phase space sampling: weighted events [PS weight $\times |\mathcal{M}|^2$]
  events: unweighted events ['just events']
- probabilistic unweighting weak spot of standard MC
- learn phase space from weighted events
  generate unweighted events [(almost)]
- give us a few weeks...



4– How to GAN event subtraction

Idea: subtract samples without binning [Butter, TP, Winterhalder]

– statistical uncertainty

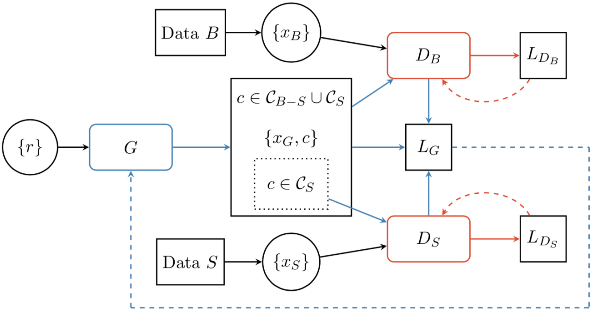
$$\Delta_{B-S} = \sqrt{\Delta_B^2 + \Delta_S^2} > \max(\Delta B, \Delta S)$$

– applications in LHC physics

soft-collinar subtraction, multi-jet merging
on-shell subtraction
background/signal subtraction

– GAN setup

1. differential, steep class label
2. sample normalization



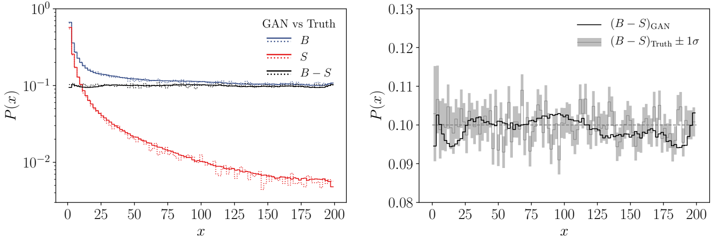
Subtracted events

How to beat statistics by subtracting

1– 1D 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 fluctuations reduced (sic!)



Subtracted events

How to beat statistics by subtracting

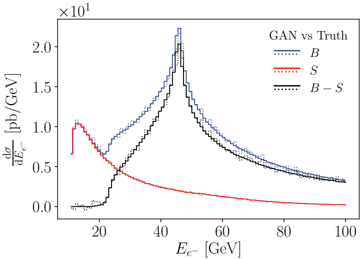
1– 1D toy example

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

– statistical fluctuations reduced (sic!)

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

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



Subtracted events

How to beat statistics by subtracting

1– 1D toy example

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

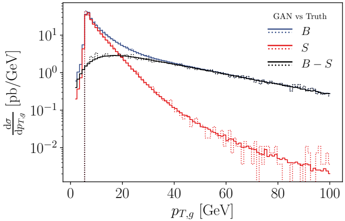
– statistical fluctuations reduced (sic!)

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

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

3– collinear subtraction [assumed non-local]

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



$\Rightarrow$ Applications in theory and analysis

5– How to GAN away detector effects

Goal: invert Markov processes [Bellagente, Butter, Kasiczka, TP, Winterhalder]

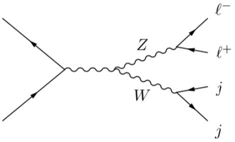
– detector simulation typical Markov process

– inversion possible, in principle [entangled convolutions]

– GAN task

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

$\Rightarrow$ Full phase space unfolded



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

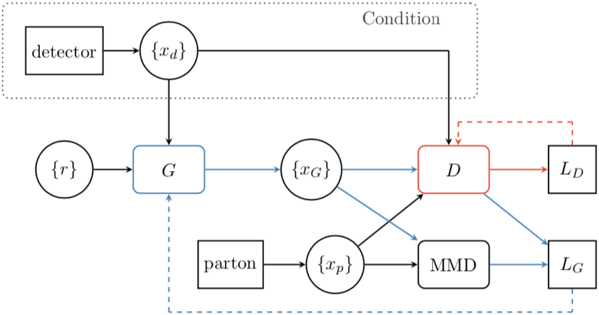
– GAN same as event generation [with MMD]

$\Rightarrow$ Perfect if training and test the same

Fully conditional GAN

Getting random sampling logic right

– map random numbers to parton level
  hadron level as condition [matched event pairs]

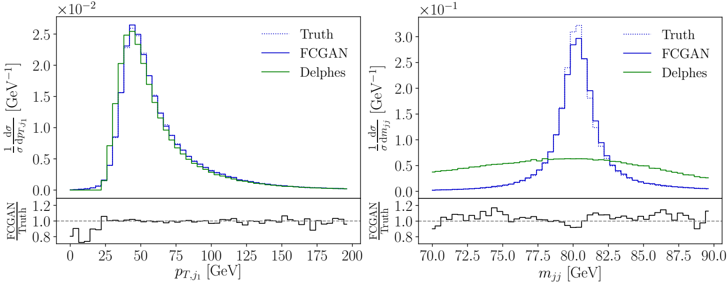


Fully conditional GAN

Getting random sampling logic right

– map random numbers to parton level
  hadron level as condition [matched event pairs]

– full inversion fine



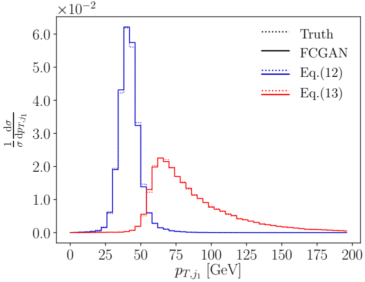
Fully conditional GAN

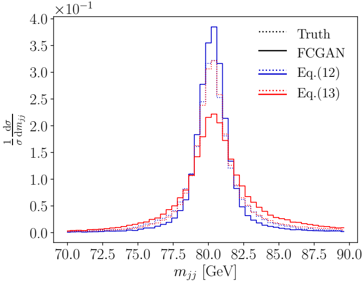
Getting random sampling logic right

- map random numbers to parton level
  hadron level as condition [matched event pairs]
- full inversion fine
- detector-level cuts [14%, 39% events, no interpolation, MMD not conditional]

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

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





Generative Invertible

Tilman Plehn

Fully conditional GAN

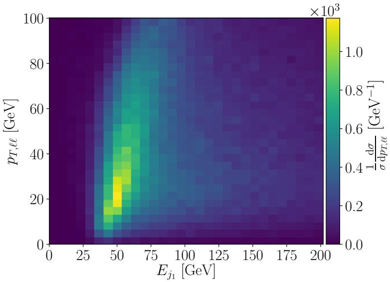
Getting random sampling logic right

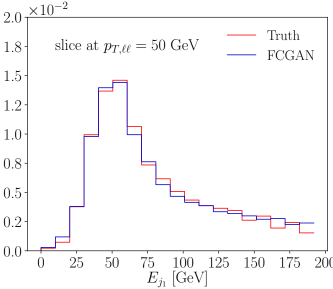
– map random numbers to parton level
  hadron level as condition [matched event pairs]
– full inversion fine
– detector-level cuts [14%, 39% events, no interpolation, MMD not conditional]

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

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

– pretty pictures in 2D





⇒ 1.FCGAN unfolding at work

BSM injection

Different training (MC) and actual data... [not in v1, thank you to Ben Nachman]

...or model dependence of unfolding

...or localization in latent space

- train: SM events
  test: 10% events with W' in s-channel $\Rightarrow$ any guesses?

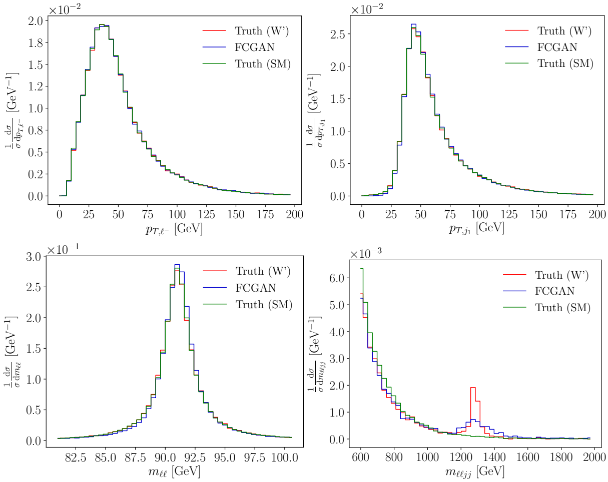
BSM injection

Different training (MC) and actual data... [not in v1, thank you to Ben Nachman]

...or model dependence of unfolding

...or localization in latent space

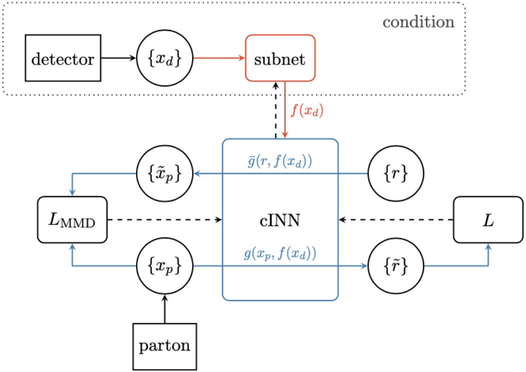
– train: SM events
  test: 10% events with W' in s-channel $\Rightarrow$ any guesses?



6– Unfolding as inverting

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

– network as bijective transformation — normalizing flow
  Jacobian tractable — normalizing flow
  evaluation in both directions — INN [Ardizzone, Rother, Köthe]
– building block: coupling layer
– conditional: parton-level events from $\{r\}$



6– Unfolding as inverting

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

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

– building block: coupling layer

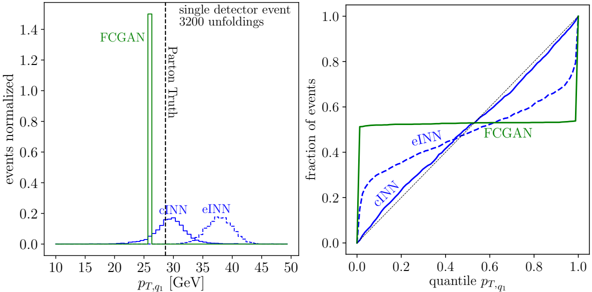
– conditional: parton-level events from $\{r\}$

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

– performance on distributions like FCGAN

– parton-level probability distribution for single detector event

$\Rightarrow$ Proper statistical unfolding



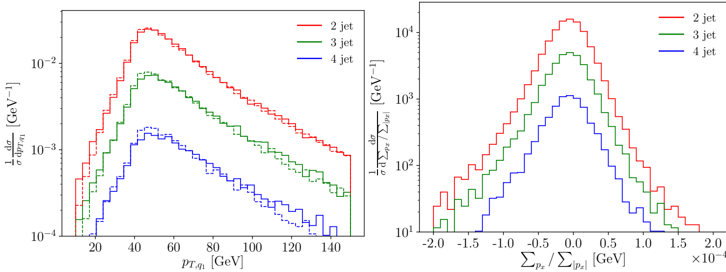
6– Unfolding as inverting

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

- network as bijective transformation — normalizing flow
  Jacobian tractable — normalizing flow
  evaluation in both directions — INN [Ardizzone, Rother, Köthe]
- building block: coupling layer
- conditional: parton-level events from $\{r\}$

Unfolding extra jets

- detector-level process $pp \to ZW$+jets [variable number of objects]
- parton-level hard process chosen $2 \to 2$ [whatever you want]
- ME vs PS jets decided by network [including momentum conservation]



6– Unfolding as inverting

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

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

– building block: coupling layer

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

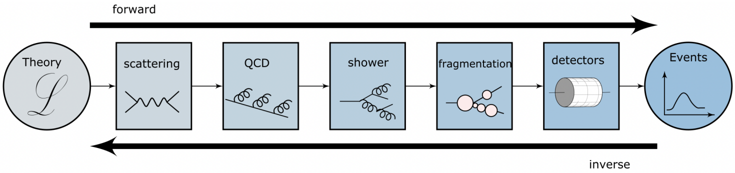
Unfolding extra jets

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

– parton-level hard process chosen $2 \to 2$ [whatever you want]

– ME vs PS jets decided by network [including momentum conservation]

$\Rightarrow$ How systematically can we invert?



Outlook

Machine learning for LHC theory

– goal: data-to-data with fundamental physics input

– MC challenges

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

– neural network benefits

  best available interpolation
  training on MC and/or data, anything goes
  lightning speed, once trained

– GANs the cool kid

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

– INNs the theory hope

  flow networks to control spaces
  invertible network the new tool

  Any ideas?