# How to GAN for LHC

Tilman Plehn

Universität Heidelberg

CERN-TH 6/2020

# Machine Learning for LHC

## Fundamental understanding of big data

– LHC and data-driven, but only fundamental with theory

1. simulation from first principles [Pythia, Sherpa, Madgraph]
2. interpretation frameworks [SMEFT, SUSY]
3. best use of the data [using 1, 2, our brains, and ML]

– 1991 visionaries: NN-based quark-gluon tagger

**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.

$\Rightarrow$ not new, not a question *if* experimentalists will use it

# Simple classification done

SciPost Physics    Submission

## The Machine Learning Landscape of Top Taggers

G. Kasieczka (ed)[1], T. Plehn (ed)[2], A. Butter[2], K. Cranmer[3], D. Debnath[4], B. M. Dillon[5], M. Fairbairn[6], D. A. Faroughy[5], W. Fedorko[7], C. Gay[7], L. Gouskos[8], J. F. Kamenik[5,9], P. T. Komiske[10], S. Leiss[1], A. Lister[7], S. Macaluso[3,4], E. M. Metodiev[10], L. Moore[11], B. Nachman,[12,13], K. Nordström[14,15], J. Pearkes[7], H. Qu[8], Y. Rath[16], M. Rieger[16], D. Shih[4], J. M. Thompson[2], and S. Varma[6]

**1** Institut für Experimentalphysik, Universität Hamburg, Germany
**2** Institut für Theoretische Physik, Universität Heidelberg, Germany
**3** Center for Cosmology and Particle Physics and Center for Data Science, NYU, USA
**4** NHECT, Dept. of Physics and Astronomy, Rutgers, The State University of NJ, USA
**5** Jozef Stefan Institute, Ljubljana, Slovenia
**6** Theoretical Particle Physics and Cosmology, King's College London, United Kingdom
**7** Department of Physics and Astronomy, The University of British Columbia, Canada
**8** Department of Physics, University of California, Santa Barbara, USA
**9** Faculty of Mathematics and Physics, University of Ljubljana, Ljubljana, Slovenia
**10** Center for Theoretical Physics, MIT, Cambridge, USA
**11** CP3, Universitéxx Catholique de Louvain, Louvain-la-Neuve, Belgium
**12** Physics Division, Lawrence Berkeley National Laboratory, Berkeley, USA
**13** Simons Inst. for the Theory of Computing, University of California, Berkeley, USA
**14** National Institute for Subatomic Physics (NIKHEF), Amsterdam, Netherlands
**15** LPTHE, CNRS & Sorbonne Université, Paris, France
**16** III. Physics Institute A, RWTH Aachen University, Germany

gregor.kasieczka@uni-hamburg.de
plehn@uni-heidelberg.de

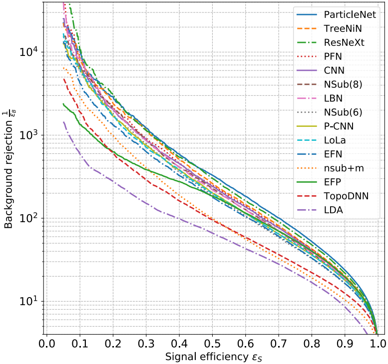
July 24, 2019

### Abstract

**Based on the established task of identifying boosted, hadronically decaying top quarks, we compare a wide range of modern machine learning approaches. Unlike most established methods they rely on low-level input, for instance calorimeter output. While their network architectures are vastly different, their performance is comparatively similar. In general, we find that these new approaches are extremely powerful and great fun.**

**Content**





# (Theory) Networks beyond classification

## Phase space networks

- MC integration [Bendavit (2017)]
- NNVegas [Klimek (2018), Carrazza (2020)]

## Event generation

- parton densities [NNPDF (since 2002)]
- amplitudes [Bishara (2019), Badger (2020)]
- neural importance sampling [Bothmann (2020)]
- i-flow in SHERPA [Gao (2020)]

## Generative networks

- Jet Images [de Oliveira (2017), Carazza (2019)]
- Detectors [Paganini (2017), Musella (2018), Erdmann (2018), Ghosh (2018), Buhmann (2020)]
- Event generation [Otten(2019), Hashemi (2019), Di Sipio (2019), Butter (2019), Martinez (2019), Alanazi (2020)]
- Unfolding [Datta (2018), Bellagente (2019)]
- Templates for QCD factorization [Lin (2019)]
- Models [Erbin (2018), Otten (2018)]
- Event subtraction [Butter (2019)]

# Inspiration from art

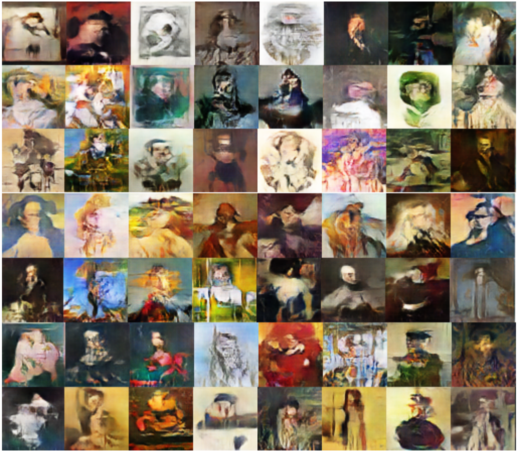
## GANGogh [Bonafilia, Jones, Danyluk (2017)]

– can networks create new pieces of art?
– train on 80,000 pictures [organized by style and genre]
– map noise vector to images
– generate flowers

# Inspiration from art

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

- can networks create new pieces of art?
- train on 80,000 pictures [organized by style and genre]
- map noise vector to images
- generate portraits



# Inspiration from art

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

– can networks create new pieces of art?

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

– map noise vector to images

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

– trained on 15,000 portraits

– sold for $432.500

$\Rightarrow$ all about marketing and sales

# GAN basics

## MC crucial for LHC physics

- goal: data-to-data with fundamental physics input only
- MC challenges

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

  best available interpolation
  structured latent space
  lightning speed, once trained
  inversion solved
  training on MC and/or data, anything goes
- GANs the cool kid

  generator trying to produce best events
  discriminator trying to catch generator
  $\longrightarrow$ competing towards (Nash) equilibrium

# GAN algorithm

## Example: LHC events

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

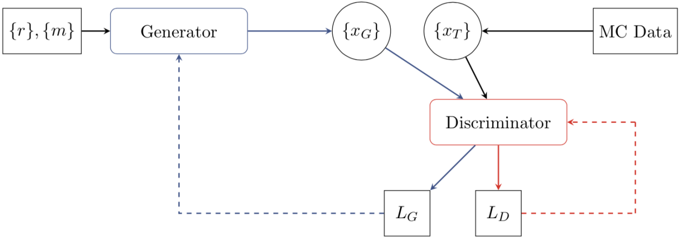
$$L_D = \big\langle -\log D(x) \big\rangle_{x\sim P_T} + \big\langle -\log(1 - D(x)) \big\rangle_{x\sim P_G} \to -2\log 0.5$$

- generator giving events [$D$ needed]

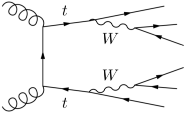
$$L_G = \big\langle -\log D(x) \big\rangle_{x\sim P_G}$$

- loss function evaluated over batch
- noise reduction/stabilization: gradient penalty [alternatively WGAN]

$\Rightarrow$ statistically independent copy of training events



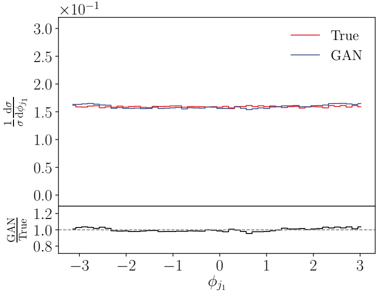
# 1– How to GAN LHC events

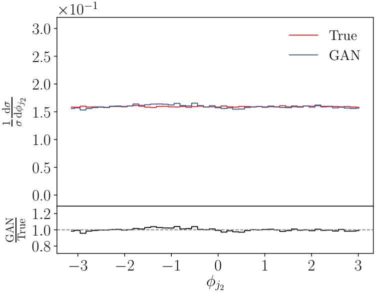



## Idea: replace ME for hard process [Butter, TP, Winterhalder]

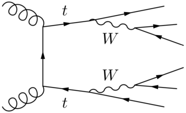
– medium-complex final state $t\bar{t} \to 6$ jets
  $t/\bar{t}$ and $W^{\pm}$ on-shell with BW $6 \times 4 = 18$ dof
  on-shell external states $\to 12$ dof [constants hard to learn]

– flat observables flat [phase space coverage okay]



# 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

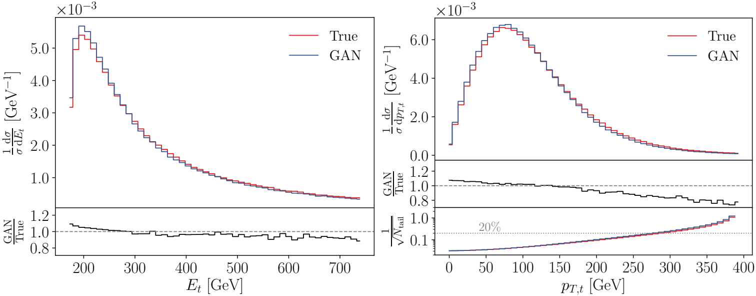
  $t/\bar{t}$ and $W^\pm$ on-shell with BW $6 \times 4 = 18$ dof

  on-shell external states $\to 12$ dof [constants hard to learn]

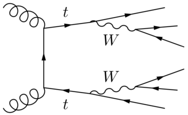
– flat observables flat [phase space coverage okay]

– direct observables with tails [statistical error indicated]

– constructed observables similar



# 1– How to GAN LHC events




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

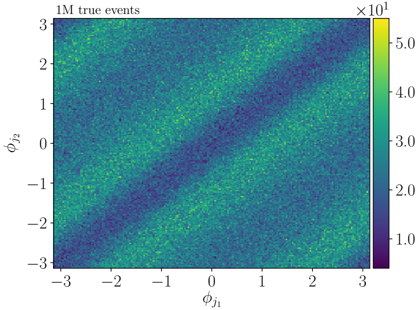
– medium-complex final state $t\bar{t} \to 6$ jets
  $t/\bar{t}$ and $W^\pm$ on-shell with BW $6 \times 4 = 18$ dof
  on-shell external states $\to 12$ dof [constants hard to learn]

– flat observables flat [phase space coverage okay]

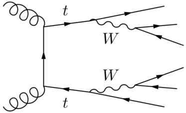
– direct observables with tails [statistical error indicated]

– constructed observables similar

– improved resolution [1M training events]

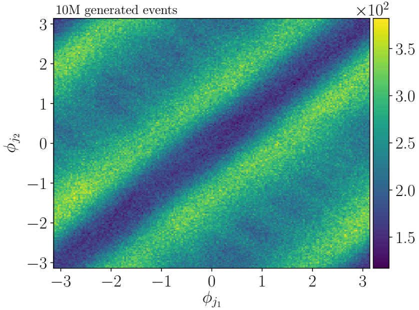


# 1– How to GAN LHC events

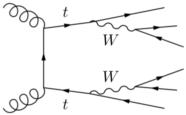



## Idea: replace ME for hard process [Butter, TP, Winterhalder]

- 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]
- flat observables flat [phase space coverage okay]
- direct observables with tails [statistical error indicated]
- constructed observables similar
- improved resolution [10M generated events]

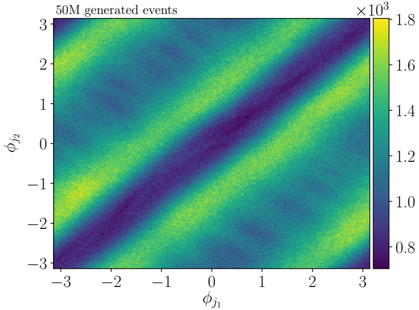


# 1– How to GAN LHC events




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



# Intermediate resonances

## GAN version of adaptive sampling

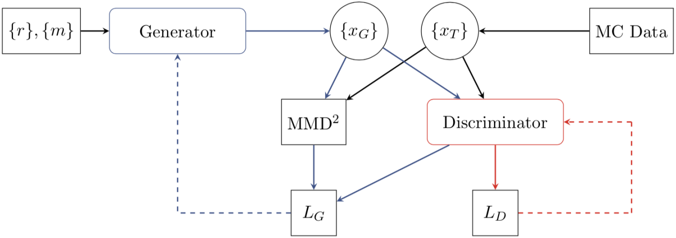
– generally 1D features

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

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

$$\text{MMD}^2 = \langle k(x,x') \rangle_{x,x' \sim P_T} + \langle k(y,y') \rangle_{y,y' \sim P_G} - 2 \langle k(x,y) \rangle_{x \sim P_T, y \sim P_G}$$

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



# Intermediate resonances

## GAN version of adaptive sampling

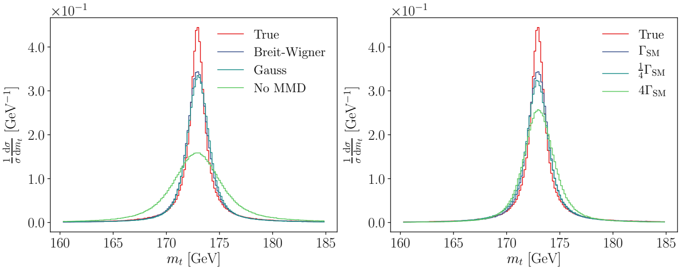
– generally 1D features

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

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

$$\text{MMD}^2 = \big\langle k(x,x') \big\rangle_{x,x' \sim P_T} + \big\langle k(y,y') \big\rangle_{y,y' \sim P_G} - 2 \big\langle k(x,y) \big\rangle_{x \sim P_T, y \sim P_G}$$

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




$\Rightarrow$ minor impact of kernel function and width

# 2– How to GAN event subtraction

## Idea: subtract event samples without bins [Butter, TP, Winterhalder]

– statistical uncertainty

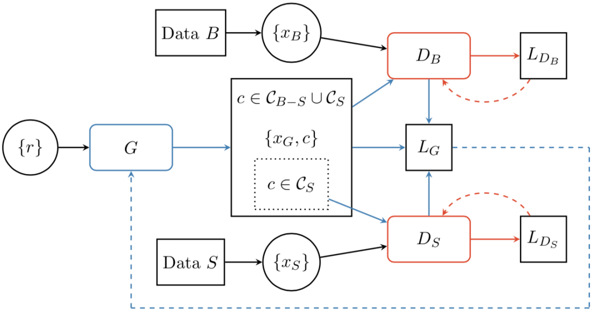
$$\Delta_{B-S} = \sqrt{\Delta_B^2 + \Delta_S^2} \max(\Delta B, \Delta S)$$

– 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



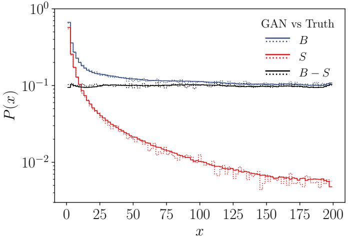
# Subtracted events

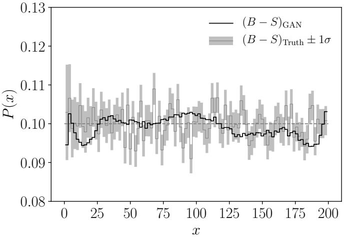
## How to beat statistics by subtracting

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

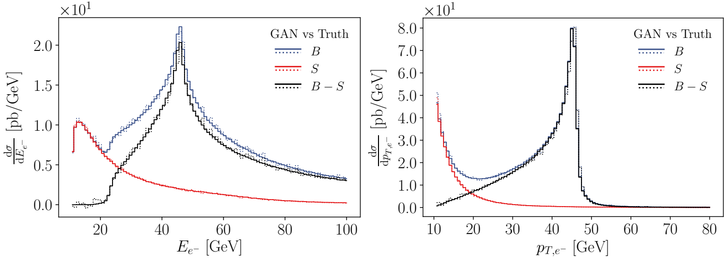
1– 1D toy example

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

– statistical fluctuations reduced (sic!)

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

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



# Subtracted events

## How to beat statistics by subtracting

1– 1D toy example

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

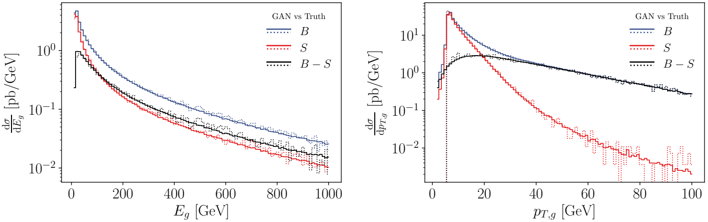
– statistical fluctuations reduced (sic!)

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

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

3– collinear subtraction [assumed non-local]

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




⇒ applications in theory and analysis

# 3– How to GAN away detector effects

## Bottom line from SFitter etc

– total rates lacking information
  STXS model-dependent
  unfolded distributions extremely convenient [$t\bar{t}$ results]

– benefits

  access to hard matrix element/first-principles QCD
  matrix element method

– challenges

  non-invertible detector simulation
  model dependence

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

– detector simulation typical Markov process

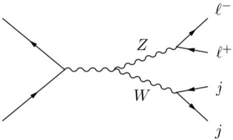
– inversion possible, in principle [entangled convolutions]

– GAN task

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

$\Rightarrow$ full phase space unfolded

# Standard GAN

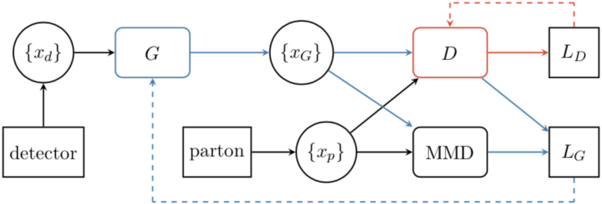



## Reconstructing the parton level

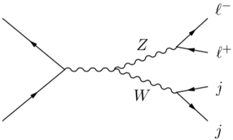
– $pp \to ZW \to (\ell\ell)\ (jj)$

– 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]

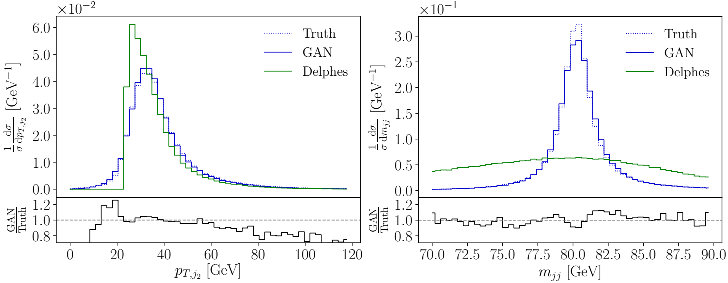


# Standard GAN

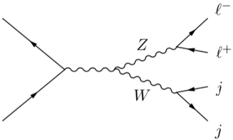



## Reconstructing the parton level

- $pp \to ZW \to (\ell\ell)\ (jj)$
- 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]
- full inversion fine



# Standard GAN

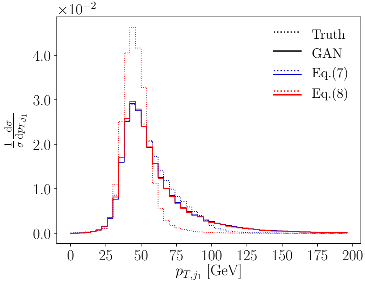



## Reconstructing the parton level

- $pp \to ZW \to (\ell\ell)\,(jj)$
- 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]
- full inversion fine
- problem: kinematics cuts in test data [88%, 38% events]

$$p_{T,j_1} = 30 \ldots 100 \text{ GeV} \tag{7}$$

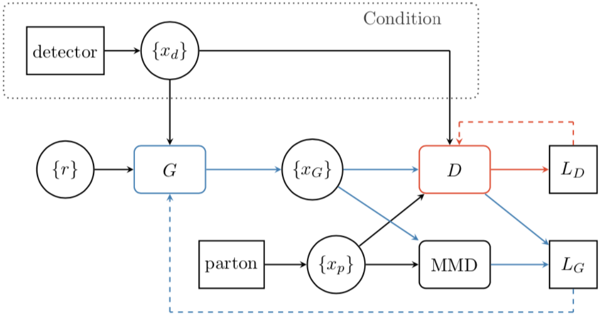
$$p_{T,j_1} = 30 \ldots 60 \text{ GeV} \quad \text{and} \quad p_{T,j_2} = 30 \ldots 50 \text{ GeV} \tag{8}$$



# Fully conditional GAN

## Adding more random sampling to network

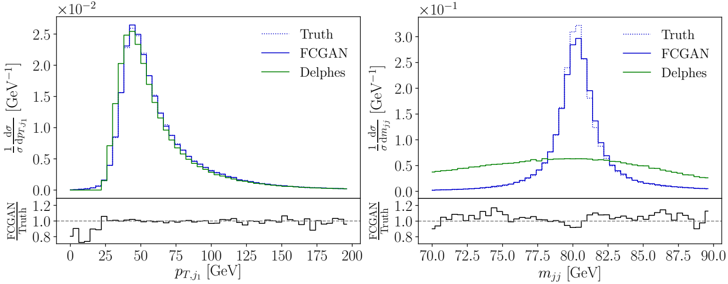
– map random numbers to parton level
  hadron level as condition [matched event pairs]



# Fully conditional GAN

## Adding more random sampling to network

- map random numbers to parton level
  hadron level as condition [matched event pairs]
- full inversion fine [again]



# Fully conditional GAN

## Adding more random sampling to network

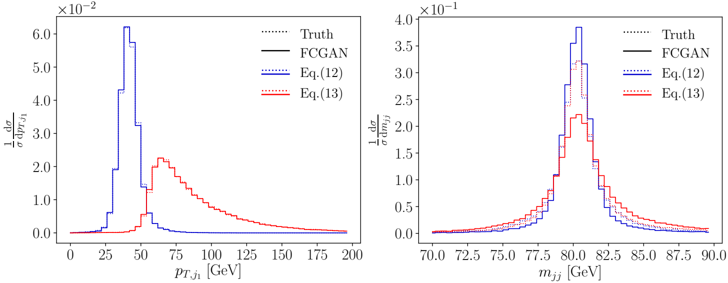
– map random numbers to parton level
  hadron level as condition [matched event pairs]

– full inversion fine [again]

– tougher cuts challenging $m_{jj}$ [14%, 39% events, no interpolation, MMD not conditional]

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

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



# Fully conditional GAN

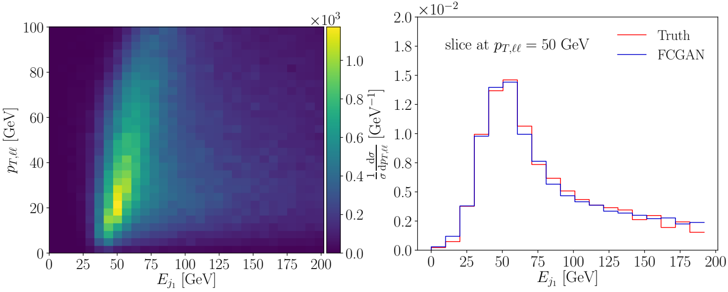
## Adding more random sampling to network

- map random numbers to parton level
  hadron level as condition [matched event pairs]
- full inversion fine [again]
- tougher cuts challenging $m_{jj}$ [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




$\Rightarrow$ 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?

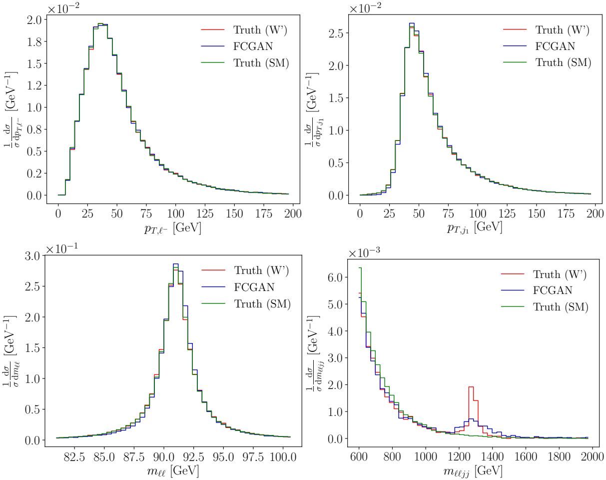
# BSM injection

## 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?



# 4– Unfolding as inverting

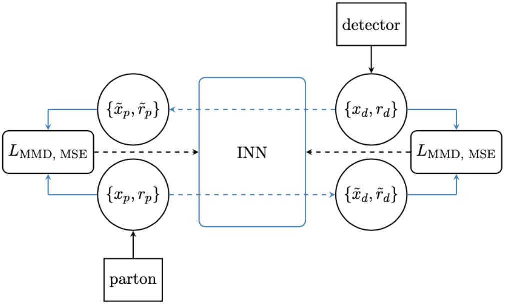
## Invertible networks? [Bellagente, Butter, Kasieczka, TP, Rousselot, Winterhalder (soon)]

- network as bijective transformation — normalizing flow
  Jacobian tractable — normalizing flow
  evaluation in both directions — INN [Ardizzone, Kruse, Rother, Köthe]
- building block: coupling layer

$$x_d \sim g(x_p) \qquad \text{with} \qquad \frac{\partial g(x_p)}{\partial x_p} = \begin{pmatrix} \text{diag } e^{s_2(x_{p,2})} & \text{finite} \\ 0 & \text{diag } e^{s_1(x_{d,1})} \end{pmatrix}$$

- padding by yet more random numbers

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



# 4– Unfolding as inverting

## Invertible networks? [Bellagente, Butter, Kasieczka, TP, Rousselot, Winterhalder (soon)]

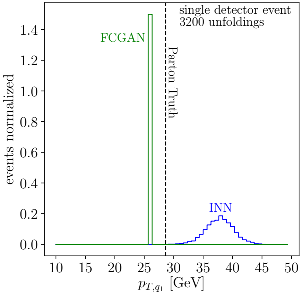
- network as bijective transformation — normalizing flow
  Jacobian tractable — normalizing flow
  evaluation in both directions — INN [Ardizzone, Kruse, Rother, Köthe]
- building block: coupling layer

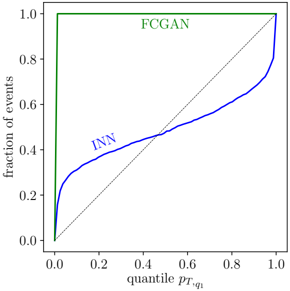
$$x_d \sim g(x_p) \qquad \text{with} \qquad \frac{\partial g(x_p)}{\partial x_p} = \begin{pmatrix} \text{diag } e^{s_2(x_{p,2})} & \text{finite} \\ 0 & \text{diag } e^{s_1(x_{d,1})} \end{pmatrix}$$

- padding by yet more random numbers

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

⇒ proper sampling



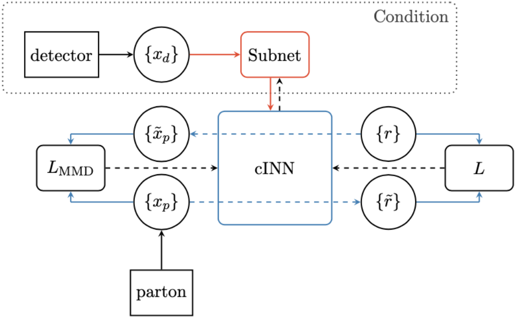



# Conditional INN

## Even more random sampling: conditional network

– same procedure as for GAN

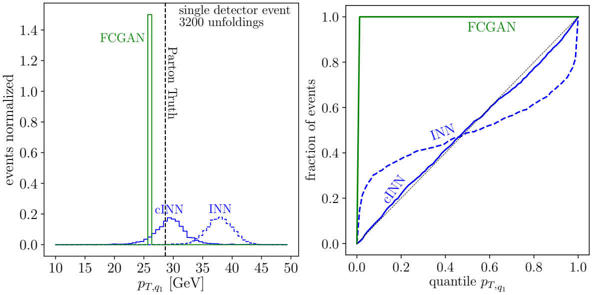
– parton-level events from random numbers



# Conditional INN

## Even more random sampling: conditional network

- same procedure as for GAN
- parton-level events from random numbers
- calibration for statistical unfolding



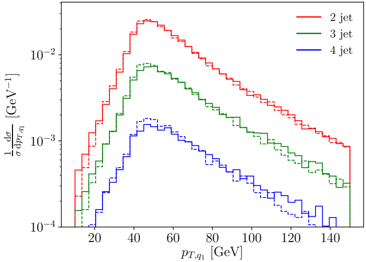
# Conditional INN

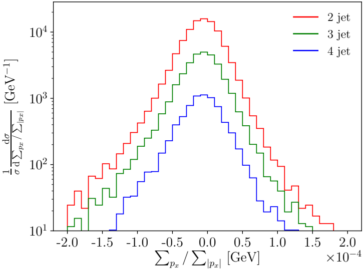
## Even more random sampling: conditional network

– same procedure as for GAN
– parton-level events from random numbers
– calibration for statistical unfolding

## 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]




$\Rightarrow$ proper statistical inversion!

# Outlook

## Machine learning a great tool box

LHC physics really is big data

jet classification was a starting point

generative networks exciting for theory

advantage 1: NN interpolation

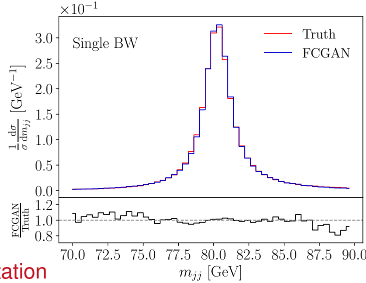
advantage 2: latent space structures

advantage 3: training on MC and/or data

advantage 4: properly invertible

Any ideas for fun applications?

# Dynamic MMD




## Technical side-remark: dynamic MMD

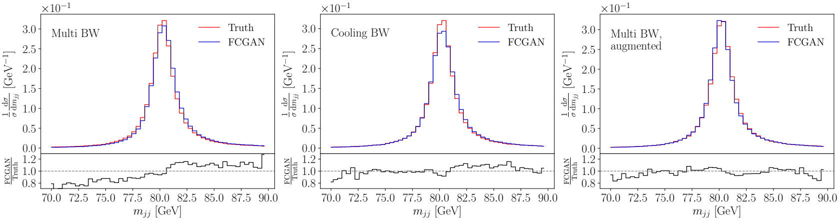
- minimal input

  functional form of correlation $m_{jj}$
  kernel shape (irrelevant) and resolution
- Adaptive resolution?

## Technical side-remark: dynamic MMD implementation

- multiple fixed-width kernels
- multiple kernels for conditional input
- cooling kernel [from SD of generator $m_{jj}$]

⇒ Technical implementation still open...



# Superresolution GANs (preview)

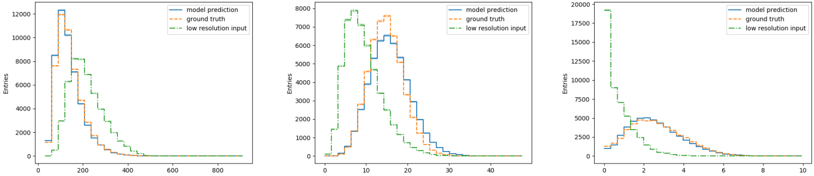
## Getting inspired [Blecher, Butter, Keilbach, TP + Irvine]

– take high-resolution calorimeter images
  down-sample to 1/8th 1D resolution
  GAN inversion

– works because the GAN learn structure [showers are QCD]

– start from low-resolution calorimeter images
  GAN high-resolution images

– energy of constituents no.1,10,30




⇒ GANs are kind of magic