Generative Networks for LHC

Tilman Plehn

Universität Heidelberg

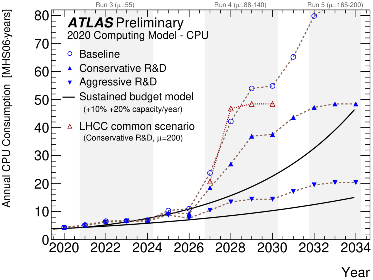
CMS 1/2021

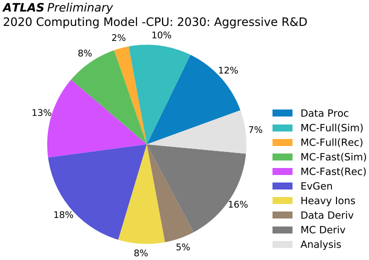
Simulations for future LHC runs

Unique: fundamental understanding of lots of data

- precision theory predictions
- precision simulations
- precision measurements

$\Rightarrow$ What's needed to keep the edge?





Simulations for future LHC runs

Unique: fundamental understanding of lots of data

- precision theory predictions
- precision simulations
- precision measurements

⇒ What's needed to keep the edge?

Event generation towards HL-LHC

- simulated event numbers scaling with the expected events [factor 25]
- general move to NLO/NNLO as standard [5% error]
- higher relevant final-state multiplicities [jet recoil, extra jets, WBF, etc.]
- additional low-rate high-multiplicity backgrounds
- specific precision predictions not available in standard generators [N^3LO in MC?]
- interpretation of measurements with general signal hypothesis [jets+MET]

Simulations for future LHC runs

Unique: fundamental understanding of lots of data

– precision theory predictions
– precision simulations
– precision measurements
⇒ What's needed to keep the edge?

Event generation towards HL-LHC

– simulated event numbers scaling with the expected events [factor 25]
– general move to NLO/NNLO as standard [5% error]
– higher relevant final-state multiplicities [jet recoil, extra jets, WBF, etc.]
– additional low-rate high-multiplicity backgrounds
– specific precision predictions not available in standard generators [N^{3}LO in MC?]
– interpretation of measurements with general signal hypothesis [jets+MET]

Three ways to use ML

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

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]

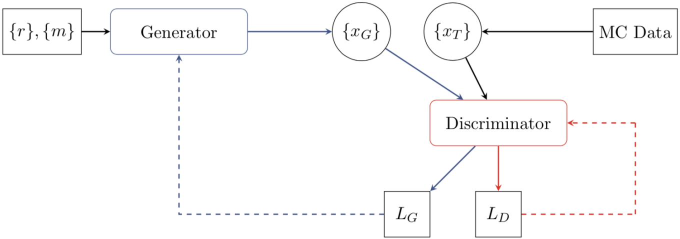
$$L_D = \left\langle -\log D(x) \right\rangle_{x_T} + \left\langle -\log(1 - D(x)) \right\rangle_{x_G}$$

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

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

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

$\Rightarrow$ statistically independent copy of training events

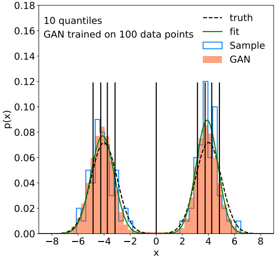
Generative network studies [review 2008.08558]

- 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)]
- Templates for QCD factorization [Lin (2019)]
- EFT models [Erbin (2018)]
- Event subtraction [Butter (2019)]
- Sherpa [Bothmann (2020), Gao (2020)]
- Basics [GANplification (2020), DCTR (2020)]
- Unweighting [Verheyen (2020), Backes (2020)]
- Superresolution [DiBello (2020), Blecher (2020)]

GANplification

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

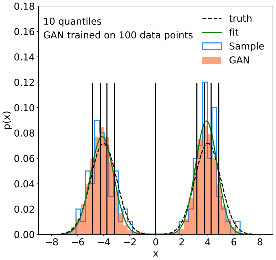
– true function known
  compare GAN vs sampling vs fit
– quantiles with χ^2-values

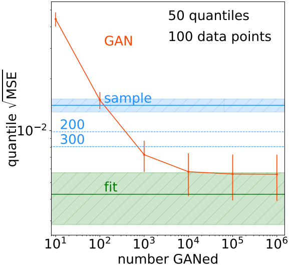


GANplification

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

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



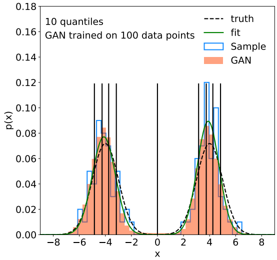


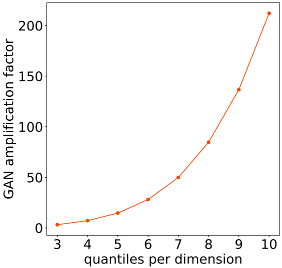
GANplification

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

- true function known
  compare GAN vs sampling vs fit
- quantiles with χ^2-values
- 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





Generative Networks

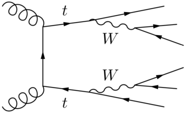
Tilman Plehn

GAN basics

1- Events

2- Inverting

How to GAN LHC events



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

– medium-complex final state $t\bar{t} \to$ 6 jets

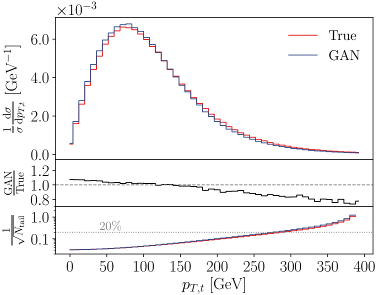
$t/\bar{t}$ and $W^{\pm}$ on-shell with BW $6 \times 4 = 18$ dof

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

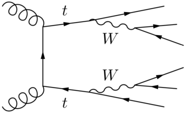
– flat observables flat [phase space coverage okay]

– direct observables with tails [statistical error indicated]

– constructed observables similar

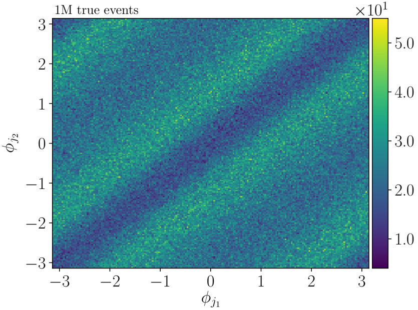


How to GAN LHC events

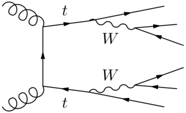


Idea: replace ME for hard process [Butter, TP, Winterhalder]

- 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]

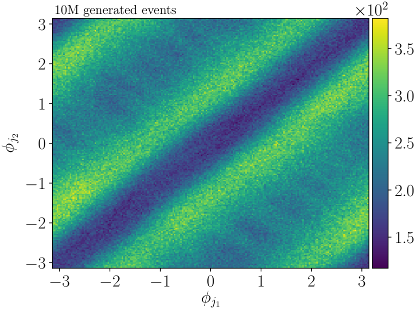


How to GAN LHC events

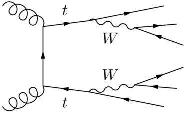


Idea: replace ME for hard process [Butter, TP, Winterhalder]

– 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]

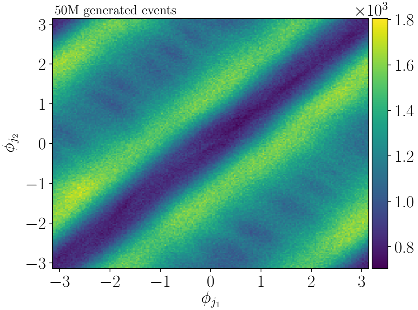


How to GAN LHC events



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

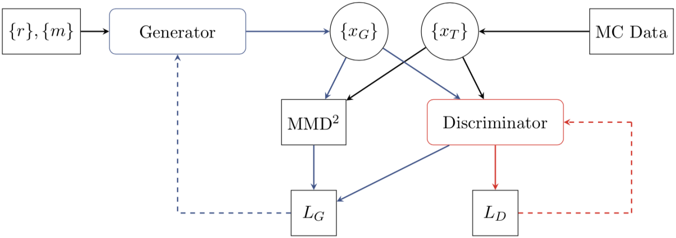
– generally 1D features

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

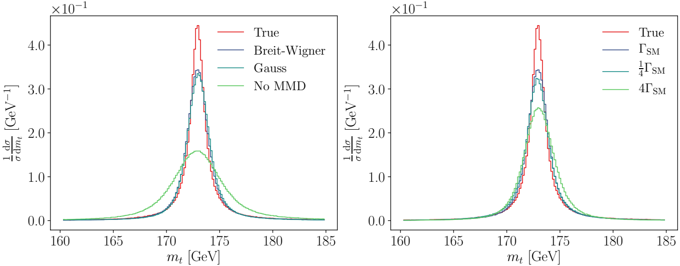
– generally 1D features

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

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

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

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



Unweighting

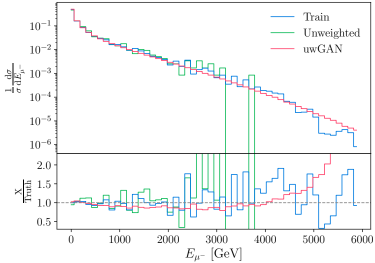
Gaining beyond GANpliflication [Butter, TP, Winterhalder; Clausius' talk]

– 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



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]

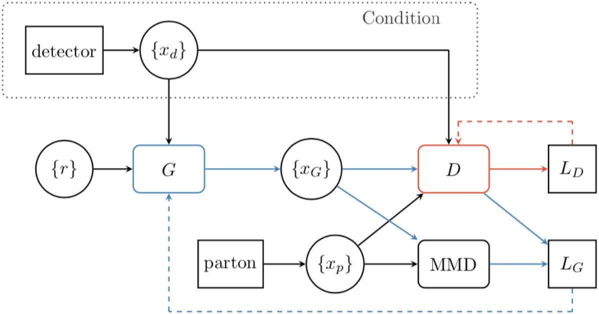
– GAN task

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

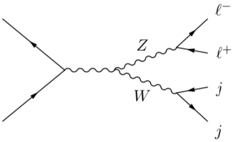
⇒ Full phase space unfolded

Conditional GAN

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



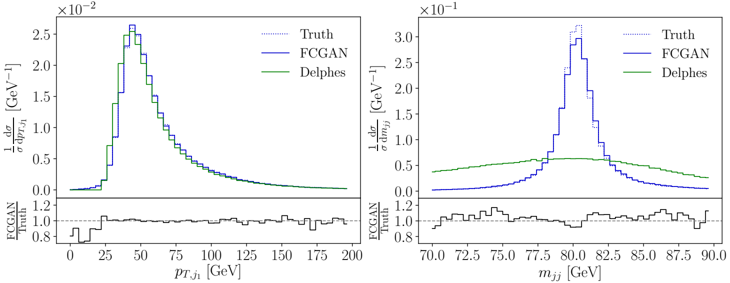
Detector unfolding



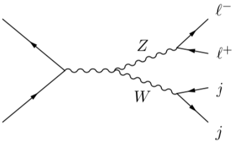
Reference process $pp \to ZW \to (\ell\ell)\ (jj)$

- 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]

Simple application



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

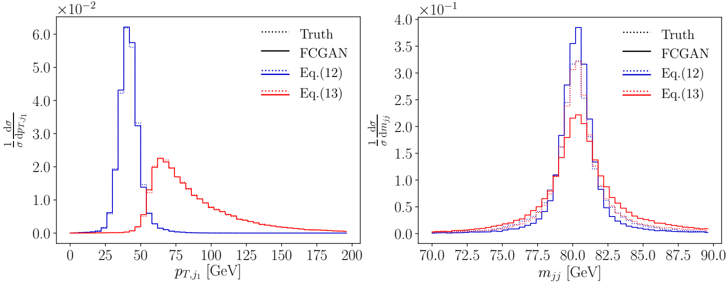
– GAN same as event generation [with MMD]

Simple application

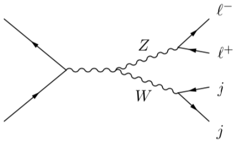
– detector-level cuts [14%, 39% events, no interpolation, MMD not conditional]

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

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



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]

Simple application

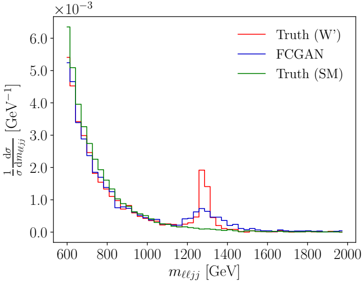
- detector-level cuts [14%, 39% events, no interpolation, MMD not conditional]

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

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

- model dependence of unfolding
- train: SM events
  test: 10% events with W' in s-channel

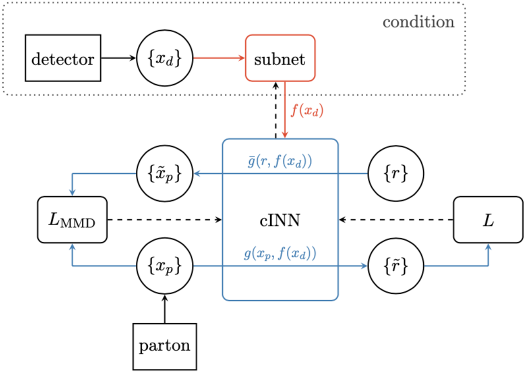
$\Rightarrow$ Working fine, but ill-defined



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 as inverting

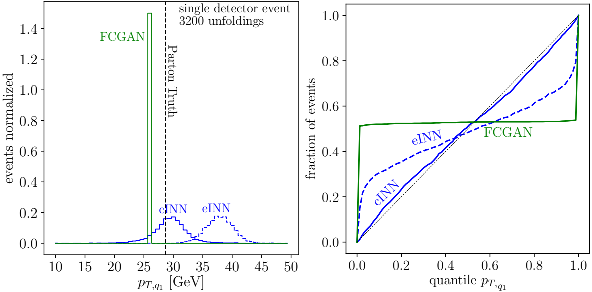
Invertible networks [Bellagente, Butter, Kasieczka, TP, Rousselot, Winterhalder, Ardizzone, Köthe]

- 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 $\{r\}$

Properly defined unfolding [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



Generative Networks

Tilman Plehn

GAN basics

1- Events

2- Inverting

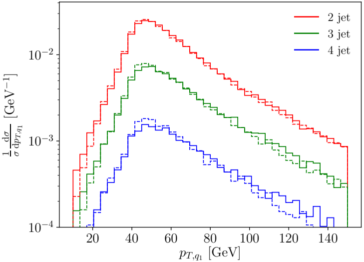
Unfolding as inverting

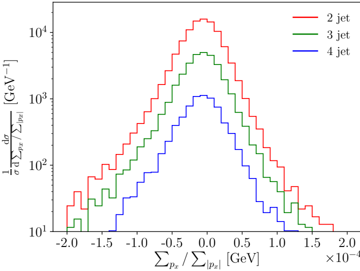
Invertible networks [Bellagente, Butter, Kasieczka, TP, Rousselot, Winterhalder, Ardizzone, Köthe]

- 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 initial-state radiation

- 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]





Unfolding as inverting

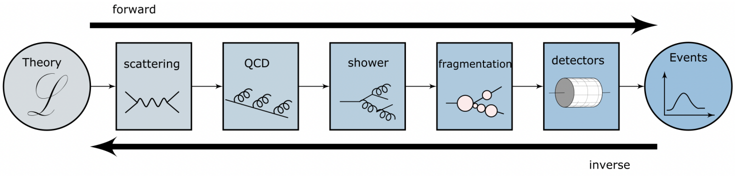
Invertible networks [Bellagente, Butter, Kasieczka, TP, Rousselot, Winterhalder, Ardizzone, Köthe]

– 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 initial-state radiation

– 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?

Backup: How to GAN event subtraction

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

– statistical uncertainty

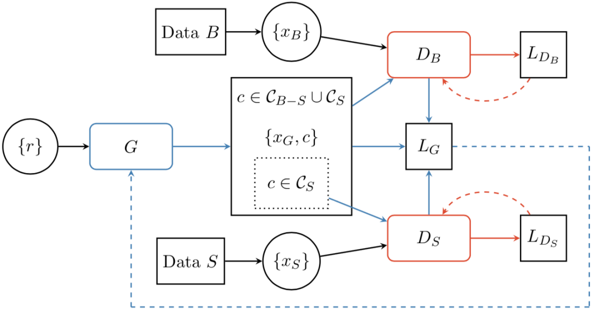
$$\Delta_{B-S} = \sqrt{\Delta_B^2 + \Delta_S^2} > \max(\Delta B, \Delta S)$$

– 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



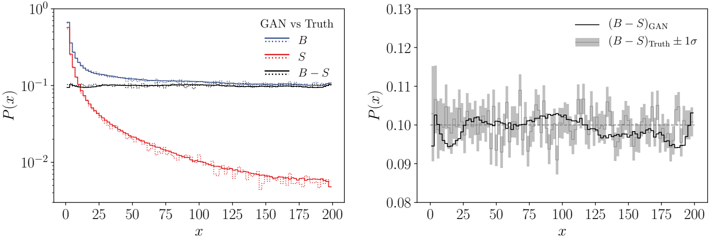
Subtracted events

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

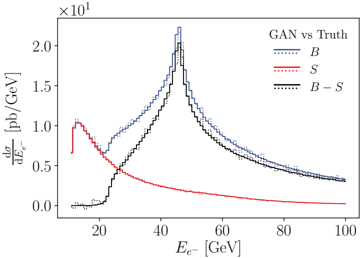
1– 1D toy example

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

– 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

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

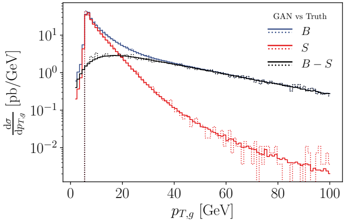
– statistical fluctuations reduced (sic!)

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

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

3– collinear subtraction [assumed non-local]

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



$\Rightarrow$ Applications in theory and analysis