Generative Neural Networks for LHC Applications

Anja Butter

ITP, Universität Heidelberg

arXiv:1907.03764, 1912.08824, and 1912.00477

with Marco Bellagente, Gregor Kasieczka, Tilman Plehn, und Ramon Winterhalder

Going beyond simple classification

- Classification is a solved problem

Going beyond simple classification

- Classification is a solved problem
- Building a full toolbox
    - Classification for density estimation
    - Tracking challenge
    - Decorrelating variables
    - Anomaly detection
    - Estimating uncertainties
    - Generative models for event generation and Detector simulation
    - ...

Phase-Space Sampling

Monte Carlo simulations at the heart of any LHC analysis

Phase-Space Sampling

Monte Carlo simulations at the heart of any LHC analysis

Problem: High-dimensionality and rich phase-space structures

Task: Finding an optimal phase-space mapping

$\rightarrow$ Computationally time consuming

Phase-Space Sampling

Monte Carlo simulations at the heart of any LHC analysis

Problem: High-dimensionality and rich phase-space structures

Task: Finding an optimal phase-space mapping

$\rightarrow$ Computationally time consuming

How to generate events more efficiently?
$\rightarrow$ Neural networks!

Neural Networks for Event Generation?

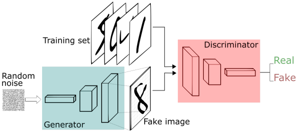
- Input: random numbers
- Output: unweighted events
- Training data:
  - unweighted MC events or real data
  - can include parton showers, hadronization and detector effects

Neural Networks for Event Generation?

- Input: random numbers
- Output: unweighted events
- Training data:
  - unweighted MC events or real data
  - can include parton showers, hadronization and detector effects

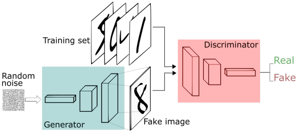
Network architecture? → generative neural network

Generative networks

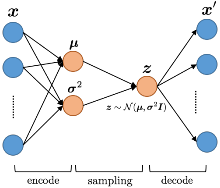


GANs

Generative networks

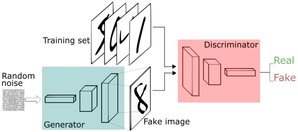


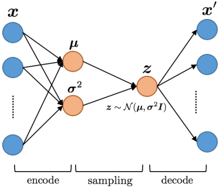
GANs



VAEs

Generative networks



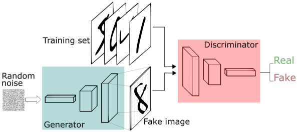


GANs

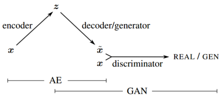
all kinds of hybrids

VAEs

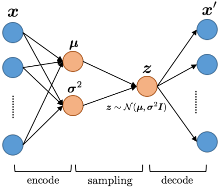
Generative networks



GANs



VAE-GAN



VAEs

Why GANs?

they are hard to train

Why GANs?

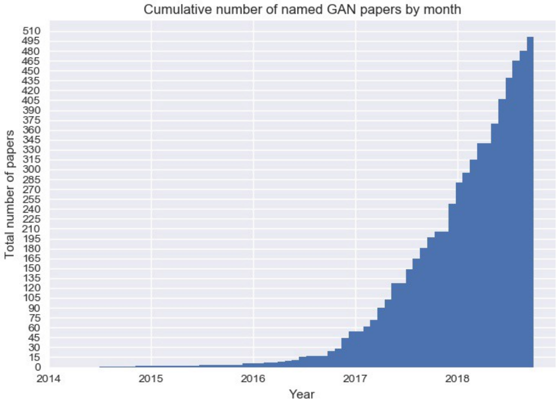
- Many people think they are hard to train

Why GANs?

- Many people think they are hard to train
- Generate better samples than VAE

Why GANs?

- Many people think they are hard to train
- Generate better samples than VAE
- Large community working on GANs



Explosive growth — All the named GAN variants cumulatively since 2014. Credit: Bruno Gavranović

→ Check out the GAN zoo!

Why GANs?

- Many people think they are hard to train
- Generate better samples than VAE
- Large community working on GANs

Why GANs?

- Many people think they are hard to train
- Generate better samples than VAE
- Large community working on GANs
- It really isn't that hard...

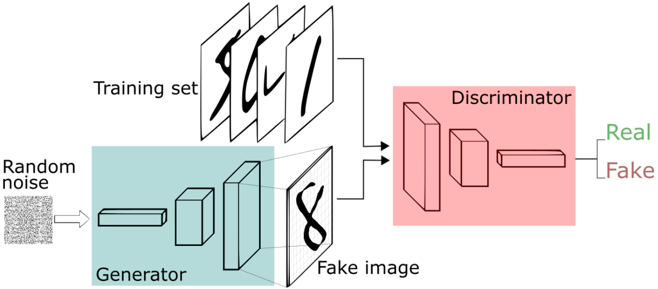
- A lot of experience as a community!
  - Jet Images - de Oliveira et al. [1701.05927], Carazza et al. [1909.01359],
  - Particle shower in Calorimeters - Paganini et al. [CaloGAN, 1705.02355, 1712.10321], Musella et al. [1805.00850], Erdmann et al. [1807.01954], ATLAS [ATL-SOFT-PUB-2018-001, ATL-SOFT-PROC-2019-007]
  - Event generation - Otten et al. [1901.00875], Hashemi et al. [1901.05282], Di Sipio et al. [1903.02433], Butter et al. [1907.03764], Martinez et al. [1912.02748], Alanazi et al. [2001.11103]
  - Unfolding - Datta et al. [1806.00433], Bellagente et al. [1912.0047]
  - Templates for QCD factorization - Lin et al. [1903.02556]
  - EFT models - Erbin et al. [1809.02612]
  - Event subtraction - Butter et al. [1912.08824]
  - ...

Generative Adversarial Networks

GAN: two competing networks → generator and discriminator

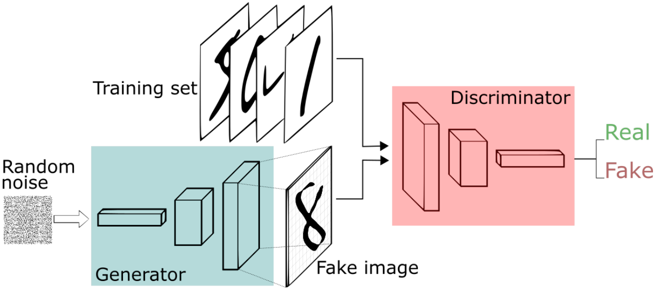
Generative Adversarial Networks

GAN: two competing networks → generator and discriminator



Generative Adversarial Networks

GAN: two competing networks → generator and discriminator

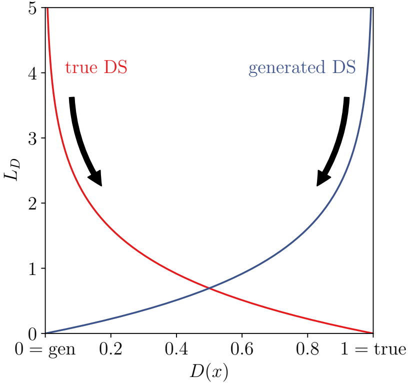


GANs used in many applications like video and image generation and physics.

A real life example

Training the Discriminator

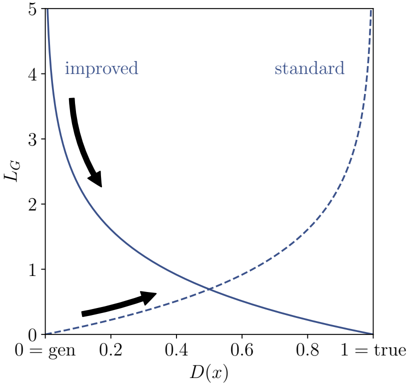
Discriminator loss



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

Training the Generator

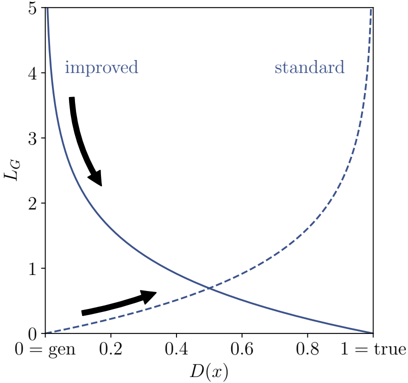
Generator loss



Maximize $$L_G = \left\langle -\log(1 - D(x)) \right\rangle_{x\sim P_G}$$

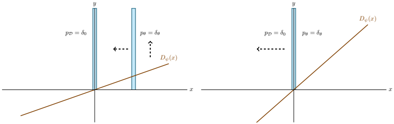
Training the Generator

Generator loss



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

Regularization



(a) $t = t_0$ (b) $t = t_1$

[1801.04406]

Adding gradient penalty

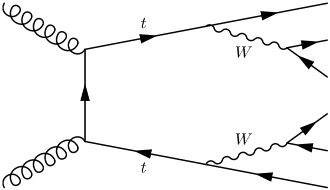
$$\phi(x) = \log \frac{D(x)}{1 - D(x)} \qquad \Rightarrow \qquad \frac{\partial \phi}{\partial x} = \frac{1}{D(x)} \frac{1}{1 - D(x)} \frac{\partial D}{\partial x} \tag{1}$$

$$L_D \to L_D + \lambda_D \left\langle (1 - D(x))^2 \, |\nabla \phi|^2 \right\rangle_{x \sim P_T} + \lambda_D \left\langle D(x)^2 \, |\nabla \phi|^2 \right\rangle_{x \sim P_G} , \tag{2}$$

Top-Pair Production

GAN events for the $2 \to 6$ particle production process

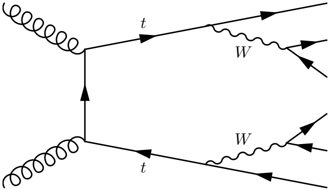
$$pp \to t\bar{t} \to (bW^-)\,(\bar{b}W^+) \to (bq_1\bar{q}_1')\,(\bar{b}q_2\bar{q}_2')\ .$$



Top-Pair Production

GAN events for the $2 \to 6$ particle production process

$$pp \to t\bar{t} \to (bW^-)\,(\bar{b}W^+) \to (bq_1\bar{q}_1')\,(\bar{b}q_2\bar{q}_2')\ .$$

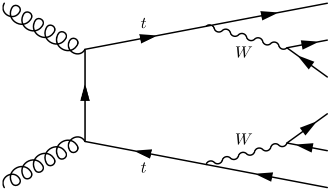


Challenges: 16-dimensional phase-space, 4 resonances, phase-space boundaries, tails

Top-Pair Production

GAN events for the $2 \to 6$ particle production process

$$pp \to t\bar{t} \to (bW^-)\,(\bar{b}W^+) \to (bq_1\bar{q}_1')\,(\bar{b}q_2\bar{q}_2')\ .$$

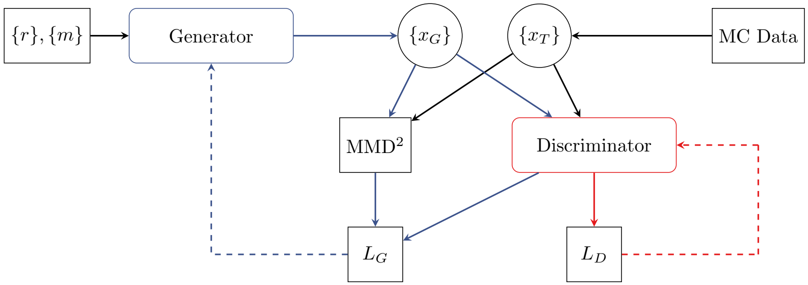


Challenges: 16-dimensional phase-space, 4 resonances, phase-space boundaries, tails

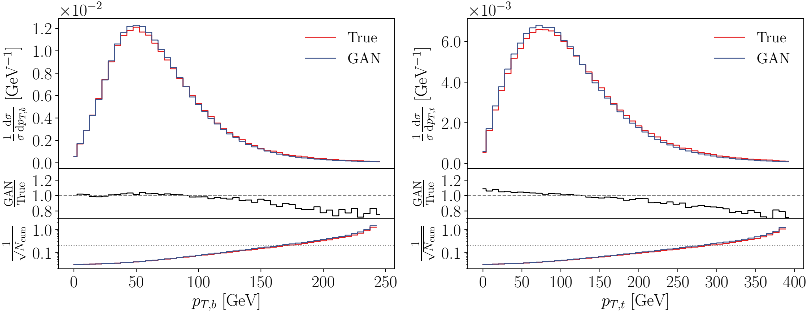
Remarks: fix masses of final state particles
$\to$ generate 18 dim output

additional loss focusing on phase-space structures
$\to$ MMD Loss

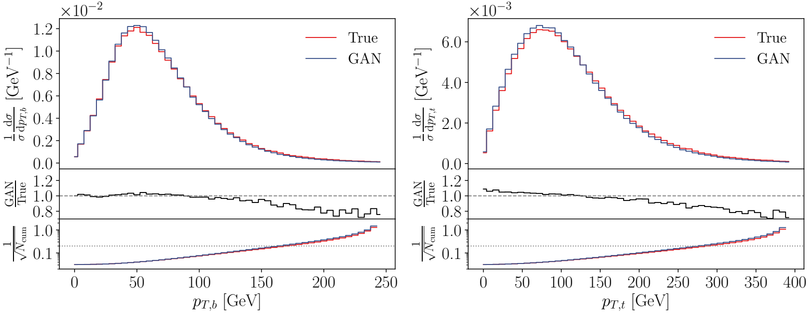
GAN Workflow



Momentum Distributions

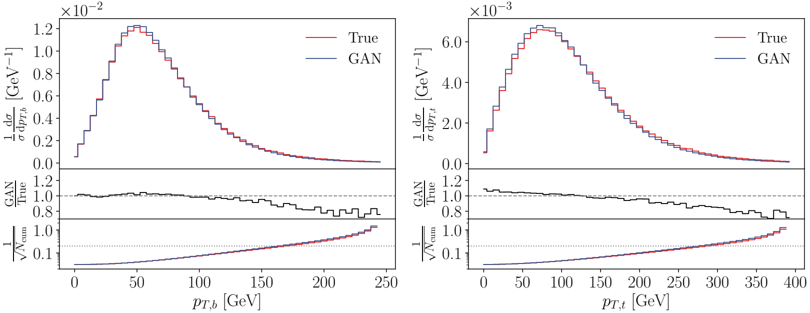


Momentum Distributions



$\rightarrow$ flat distributions easy to learn!

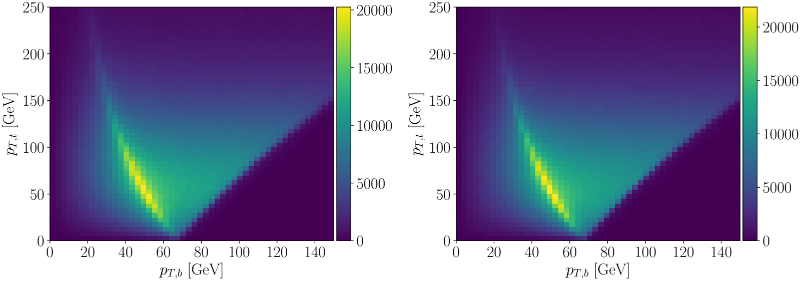
Momentum Distributions



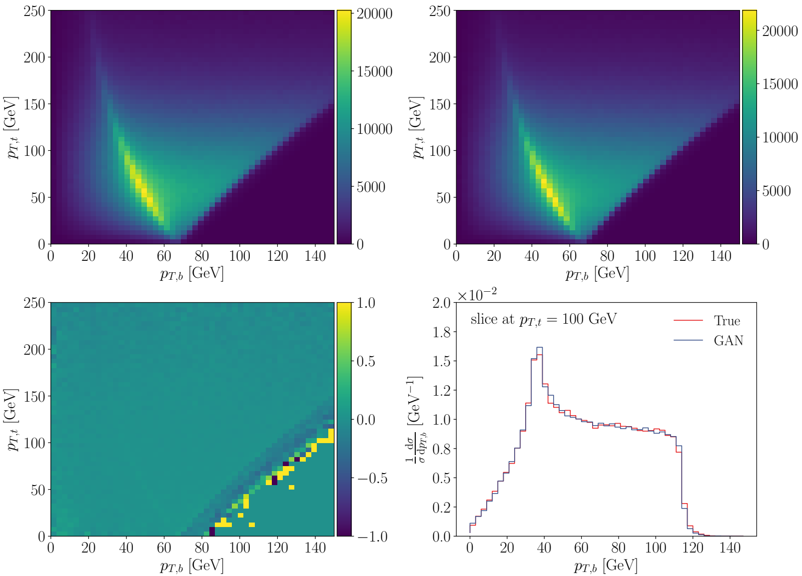
$\rightarrow$ flat distributions easy to learn!

$\rightarrow$ Deviations scale with statistic uncertainty in the tail

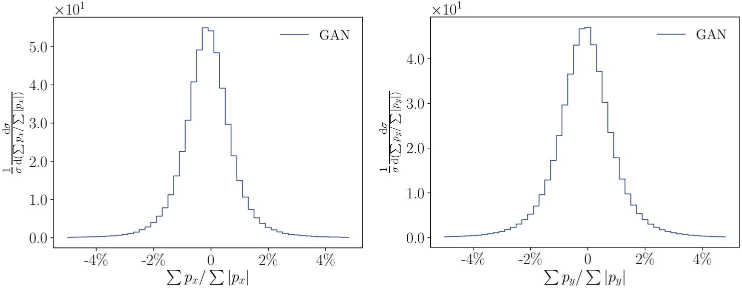
2-dimensional Correlations



2-dimensional Correlations



Momentum Conservation by the Network



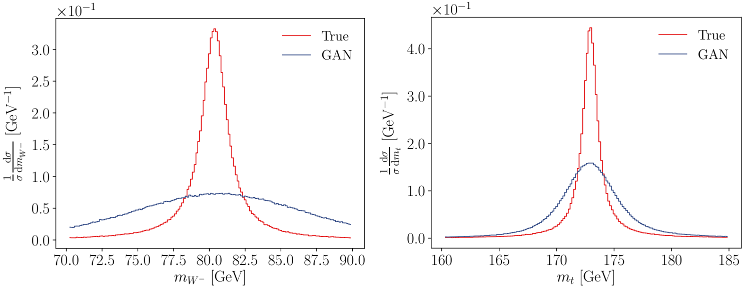
The generator learns to conserve momentum at a 1% level.

Invariant Mass Peaks

What about the resonances?

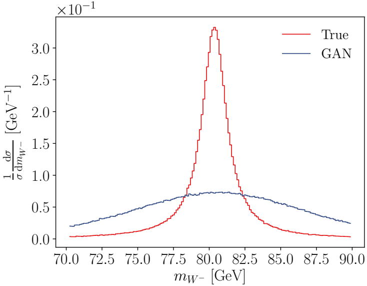
Invariant Mass Peaks

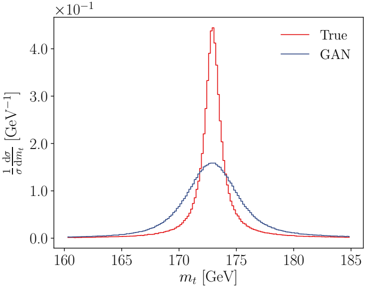
Without the additional loss:



Invariant Mass Peaks

Without the additional loss:

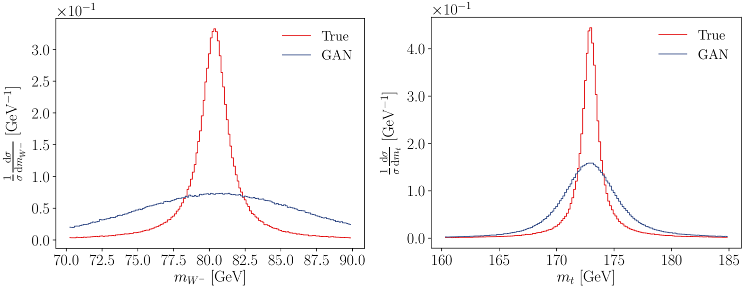




Challenge: resolve the mass peaks

Invariant Mass Peaks

Without the additional loss:



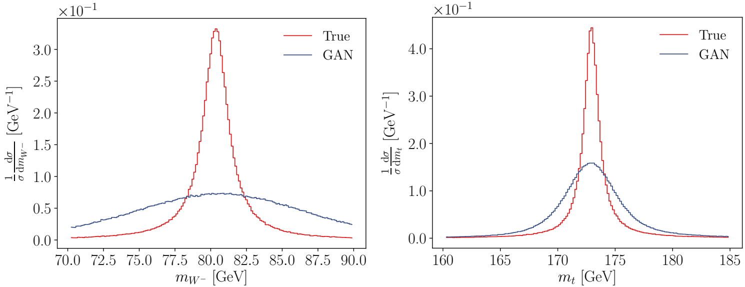
Challenge: resolve the mass peaks

Standard solution: phase-space remapping

$$\int ds \frac{F(s)}{(s-m^2)^2+m^2\Gamma^2} = \frac{1}{m\Gamma}\int dz\ F(s) \quad \text{with} \quad z = \arctan\frac{s-m^2}{m\Gamma}\ .$$

Invariant Mass Peaks

Without the additional loss:



Challenge: resolve the mass peaks

Standard solution: phase-space remapping

$$\int ds \frac{F(s)}{(s-m^2)^2 + m^2\Gamma^2} = \frac{1}{m\Gamma}\int dz\ F(s) \quad \text{with} \quad z = \arctan\frac{s-m^2}{m\Gamma}\ .$$

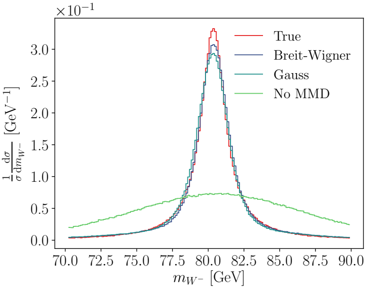
However: knowledge of m and Γ needed

Invariant Mass Peaks

Can we learn it simply from data?

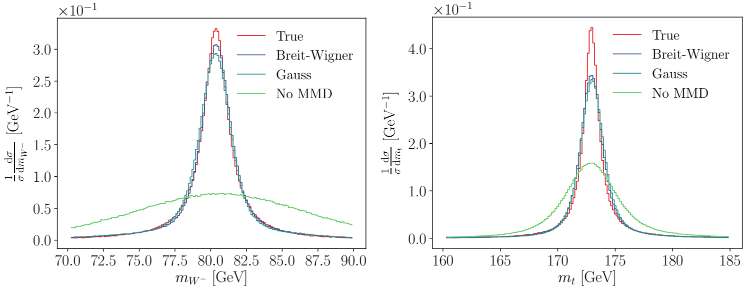
Invariant Mass Peaks

Including the MMD Loss



Invariant Mass Peaks

Including the MMD Loss



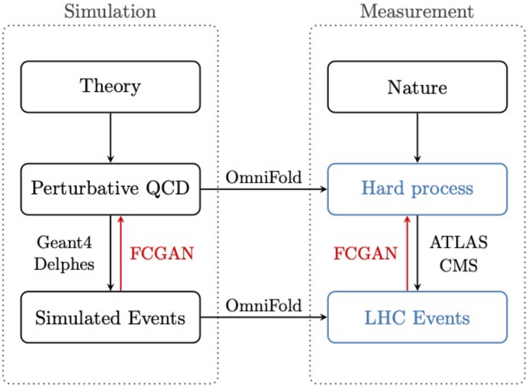
$$\mathrm{MMD}^2(P_T, P_G) = \big\langle k(x,x')\big\rangle_{x,x'\sim P_T} + \big\langle k(y,y')\big\rangle_{y,y'\sim P_G} - 2\big\langle k(x,y)\big\rangle_{x\sim P_T, y\sim P_G}$$

- free kernel choice $\rightarrow$ stable results
- **no** knowledge of m and Γ needed

First conclusion

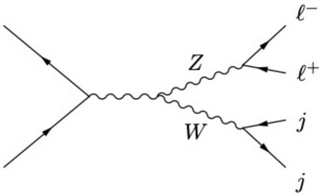
- The GAN is able to reproduce the full phase space structure of a realistic LHC process
- Flat distributions can be reproduced at arbitrary precison, limited only by statistics
- Using the MMD loss, we can even describe rich peaking resonances properly
- The same setup will allow us to generate events from an actual LHC event sample
- The GAN does not require any event unweighting

Unfolding detector effects



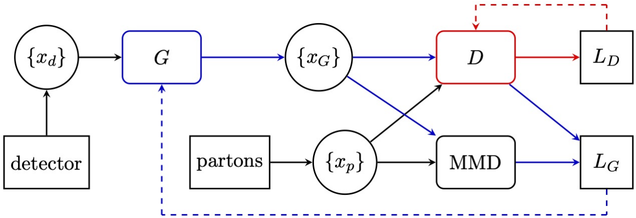
Setup

$$pp \rightarrow ZW^{\pm} \rightarrow (\ell^- \ell^+)\,(jj) \tag{3}$$



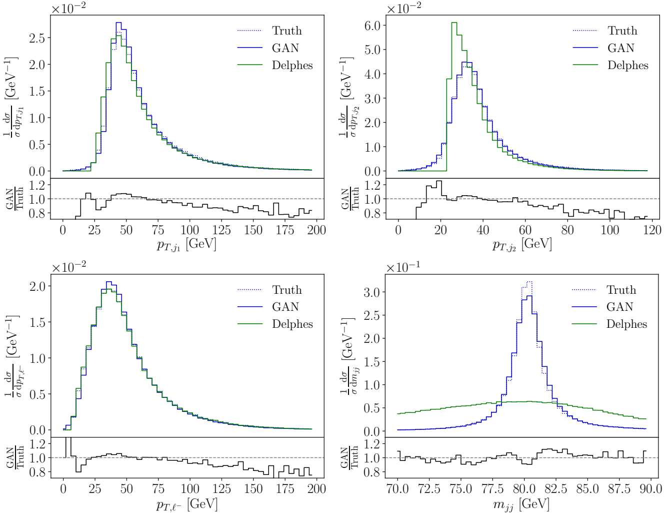
- 300k events using MadGraph+Pythia and Delphes, no ISR
- event selection:
  - exactly 2 jets and a pair of same-flavor opposite-sign leptons.
  - $p_{T,j} > 25$ GeV $\&|\eta_j| < 2.5$ GeV.
- Assign jet to a corresponding parton level object based on ΔR
- Assign leptons based on their charge

GAN setup



- Use GAN to map detector level events to parton level events

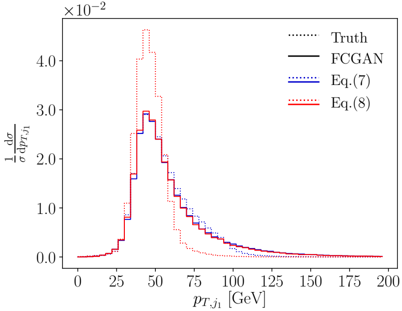
Unfolding the full distribution

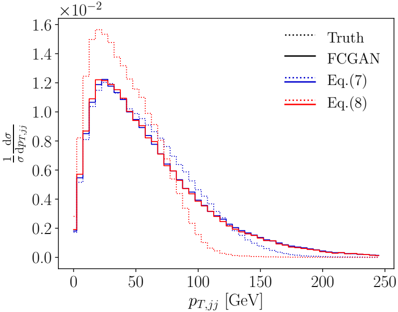


Slicing

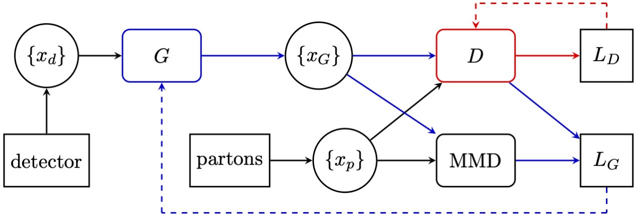
$$\text{Eq.(7)}: \quad p_{T,j_1} = 30 \ldots 100 \text{ GeV}$$

$$\text{Eq.(8)}: \quad p_{T,j_1} = 30 \ldots 60 \text{ GeV} \quad \text{and} \quad p_{T,j_2} = 30 \ldots 50 \text{ GeV}$$

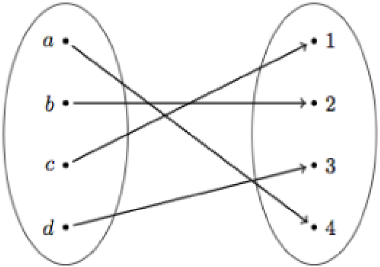




GAN setup



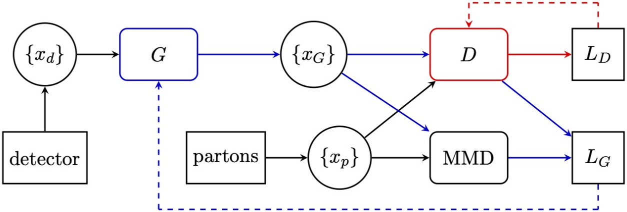
Problems



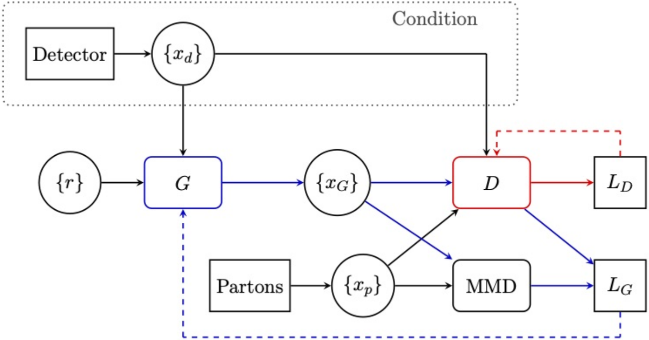
- No use of detector level information
- No concept of locality
- No stochastic mapping

→ Conditional GAN

Conditional GAN I



Conditional GAN I



Conditional GAN II

Adjust loss function

$$L_D = \left\langle -\log D(x) \right\rangle_{x \sim P_p} + \left\langle -\log \left(1 - D(x)\right) \right\rangle_{x \sim P_G}$$

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

Conditional GAN II

Adjust loss function

$$L_D = \langle - \log D(x) \rangle_{x \sim P_p} + \langle - \log (1 - D(x)) \rangle_{x \sim P_G}$$

$$\to L_D^{(\mathrm{FC})} = \langle - \log D(x, y) \rangle_{x \sim P_T, y \sim P_d} + \langle - \log (1 - D(x, y)) \rangle_{x \sim P_G, y \sim P_d}$$

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

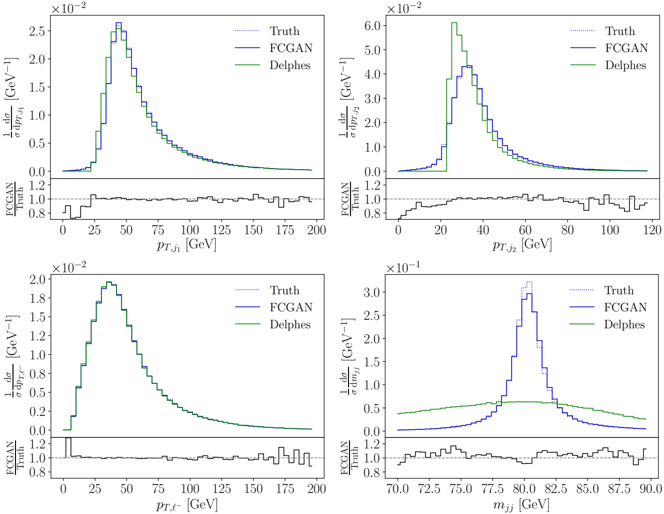
Conditional GAN II

Adjust loss function

$$
\begin{aligned}
L_D &= \langle -\log D(x) \rangle_{x\sim P_p} + \langle -\log (1 - D(x)) \rangle_{x\sim P_G} \\
\rightarrow L_D^{(\mathrm{FC})} &= \langle -\log D(x,y) \rangle_{x\sim P_T, y\sim P_d} + \langle -\log (1 - D(x,y)) \rangle_{x\sim P_G, y\sim P_d}
\end{aligned}
$$

$$
\begin{aligned}
L_G &= \langle -\log D(x) \rangle_{x\sim P_G} \\
\rightarrow L_G^{(\mathrm{FC})} &= \langle -\log D(x,y) \rangle_{x\sim P_G, y\sim P_d}
\end{aligned}
$$

Full distributions

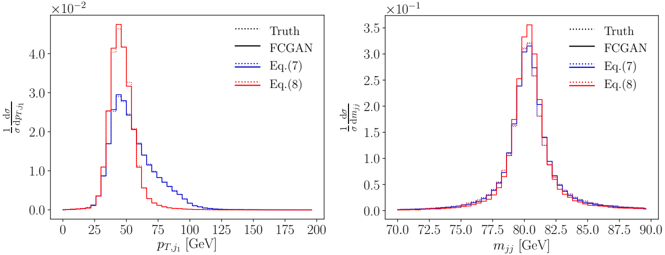


→ Nice by-product: No systematic effect in the tails!

Slicing

$$\text{Eq.(7)}: \quad p_{T,j_1} = 30 \ldots 100 \text{ GeV} \quad (\sim 88\%)$$

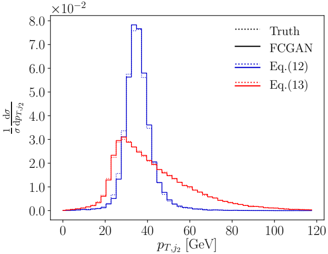
$$\text{Eq.(8)}: \quad p_{T,j_1} = 30 \ldots 60 \text{ GeV} \quad \text{and} \quad p_{T,j_2} = 30 \ldots 50 \text{ GeV} \quad (\sim 38\%)$$

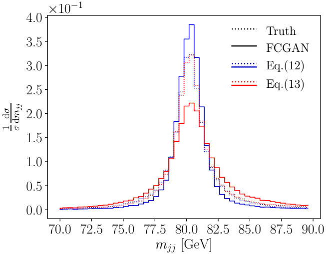


$\rightarrow$

Slicing until it breaks

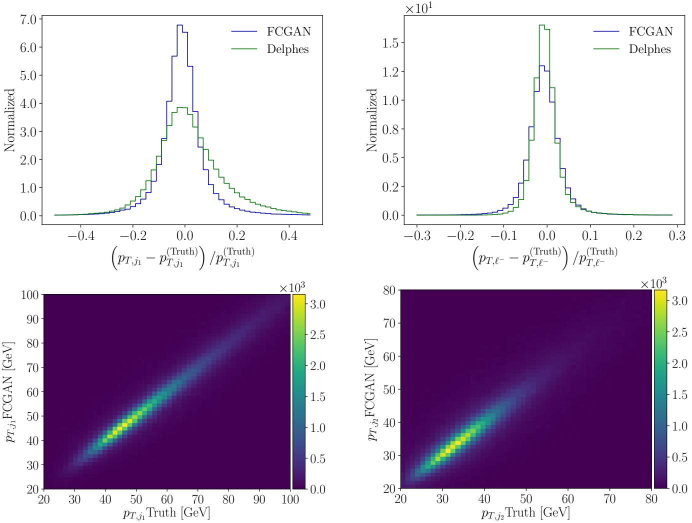
$$\text{Eq.(12)}: \quad p_{T,j_1} = 30 \ldots 50 \text{ GeV} \quad p_{T,j_2} = 30 \ldots 40 \text{ GeV}$$
$$p_{T,\ell^-} = 20 \ldots 50 \text{ GeV} \quad (\sim 14\%)$$
$$\text{Eq.(13)}: \quad p_{T,j_1} > 60 \text{ GeV} \quad (\sim 39\%)$$





→ Requires additional conditioning on the mass

Consistency check - pull & migration matrix



Conclusion Unfolding

- Normal GAN can map full detector level distribution to full parton level distribution
- \- However: No meaningful event by event matching

→ FCGAN introduces stochastic behaviour and notion of locality

\+ More stable predictions for tails

\+ Meaningful slicing

- Only breaks for non conditional invariant mass

- What's next?

Physics case

- Theory uncertainties have become a limiting factor for LHC analyses

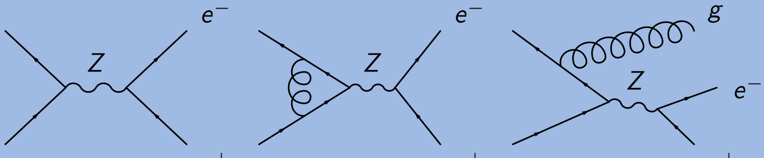
$\rightarrow$ Need for better accuracy

Physics case

- Theory uncertainties have become a limiting factor for LHC analyses

$\rightarrow$ Need for better accuracy

NLO in a nutshell



(a) Born (b) Virtual (c) Real

$$\sigma_{NLO} = \int d\Phi_B (B + V) + \int d\Phi_R R$$

Subtracting divergencies

- Virtual and real corrections diverge individually (eg. IR divergence)
- Sum of divergent contributions is finite

→ Introduce dipoles D_i to cancel divergencies

Dipole subtraction

$$\sigma_{NLO} = \int d\Phi_B \left(B + V + \sum_i d\Phi_{R|B} D_i\right) + \int d\Phi_R \left(R - \sum_i D_i\right)$$

Subtracting divergencies

- Virtual and real corrections diverge individually (eg. IR divergence)
- Sum of divergent contributions is finite

→ Introduce dipoles D_i to cancel divergencies

Dipole subtraction

$$\sigma_{NLO} = \int d\Phi_B \Big(B + V + \sum_i d\Phi_{R|B} D_i\Big) + \int d\Phi_R \Big(R - \sum_i D_i\Big)$$

- Analytic solution only possible for simple processes
- Numeric subtraction of samples:
  - → large statistic uncertainties
  - → limits efficiency
- Other use cases: eg. on-shell subtractions, multi-jet merging

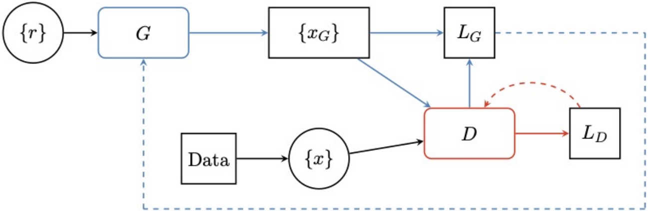
Sample based subtraction of distributions

- Use GAN to subtract distribution P_S (subtract) from P_B (base)
- Distributions represented by samples
- GAN output: samples following P_{B-S}
- Idea:
  - One discriminator per sample distribution
  - Generate label vector c to identify subtraction events
  - $0 \leq c_i \leq 1$, $\sum_i c_i = 1 \rightarrow$ softmax

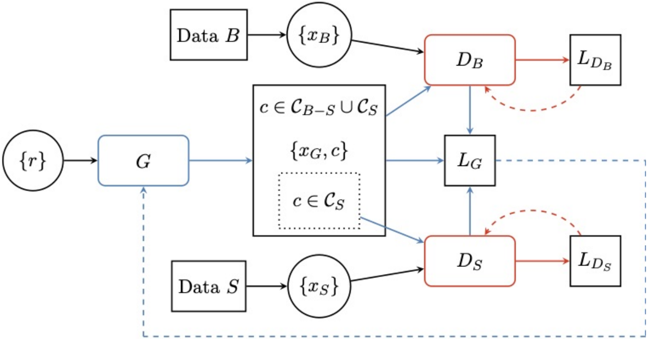
$$c = \begin{pmatrix} \mathcal{C}_S \\ \mathcal{C}_{B-S} \end{pmatrix}$$

	$\mathcal{C}_{B-S}$	$\mathcal{C}_S$
Data B	1	1
Data S	0	1
B-S	1	0

From a standard GAN ...



… to a subtraction GAN



Building the loss function

- Standard GAN loss for each discriminator

Building the loss function

- Standard GAN loss for each discriminator
- Differentiable function to count events of one type

$$f(c) = e^{-\alpha(\max(c)^2-1)^{2\beta}} \in [0,1] \qquad \text{for} \qquad 0 \leq c_i \leq 1 \; .$$

Building the loss function

- Standard GAN loss for each discriminator
- Differentiable function to count events of one type

$$f(c) = e^{-\alpha(\max(c)^2-1)^{2\beta}} \in [0,1] \qquad \text{for} \qquad 0 \leq c_i \leq 1 \,.$$

- Reward clear class assignment

$$L_G^{(\text{class})} = \left(1 - \frac{1}{b} \sum_{c \in batch} f(c)\right)^2$$

Building the loss function

- Standard GAN loss for each discriminator
- Differentiable function to count events of one type

$$f(c) = e^{-\alpha(\max(c)^2-1)^{2\beta}} \in [0,1] \qquad \text{for} \qquad 0 \leq c_i \leq 1\,.$$

- Reward clear class assignment

$$L_G^{(\text{class})} = \left(1 - \frac{1}{b}\sum_{c\in batch} f(c)\right)^2$$

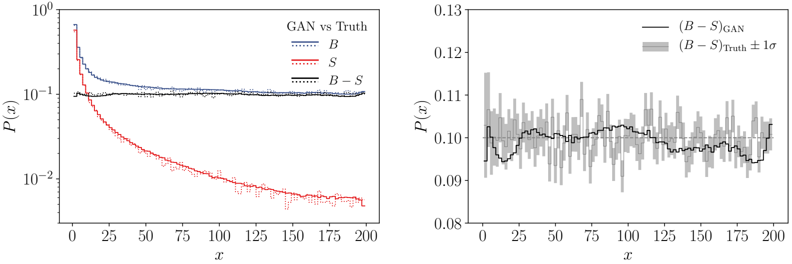
- Fix normalization

$$L_{G_i}^{(\text{norm})} = \left(\frac{\sum_{c\in\mathcal{C}_i} f(c)}{\sum_{c\in\mathcal{C}_B} f(c)} - \frac{\sigma_i}{\sigma_0}\right)^2$$

Toy example

- Toy example:

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



Generalizing the setup

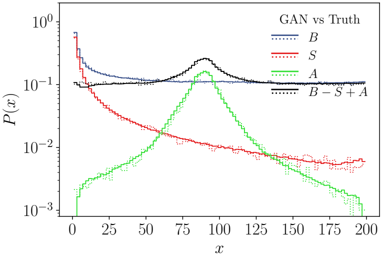
- Include addition

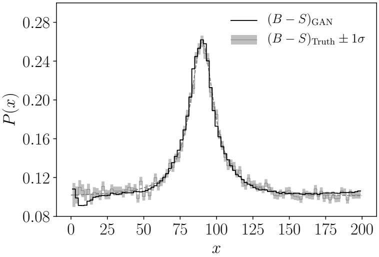
	$\mathcal{C}_{B-S}$	$\mathcal{C}_S$	$\mathcal{C}_A$
Data B	1	1	0
Data S	0	1	0
Data A	0	0	1
B-S+A	1	0	1

- Use case:
  - One distribution is represented by significantly smaller dataset

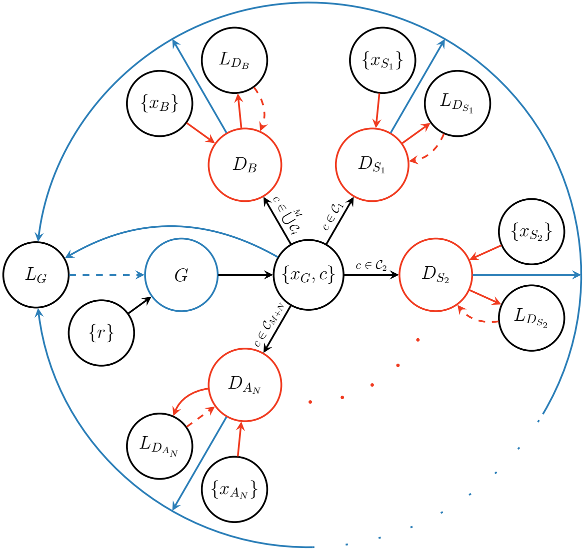
Include addition

$$P_B(x) = \frac{1}{x} + 0.1$$
$$P_S(x) = \frac{1}{x}$$
$$P_A(x) = \frac{5}{\pi}\frac{10}{10^2 + (x-90)^2}$$

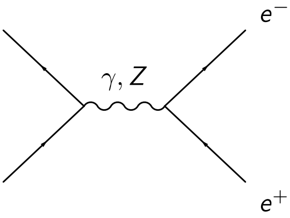




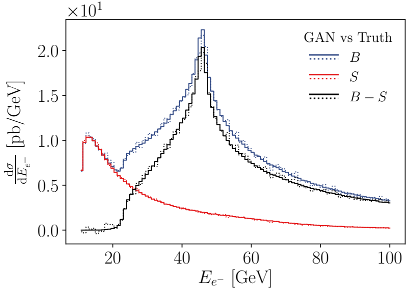
Allowing for more datasets



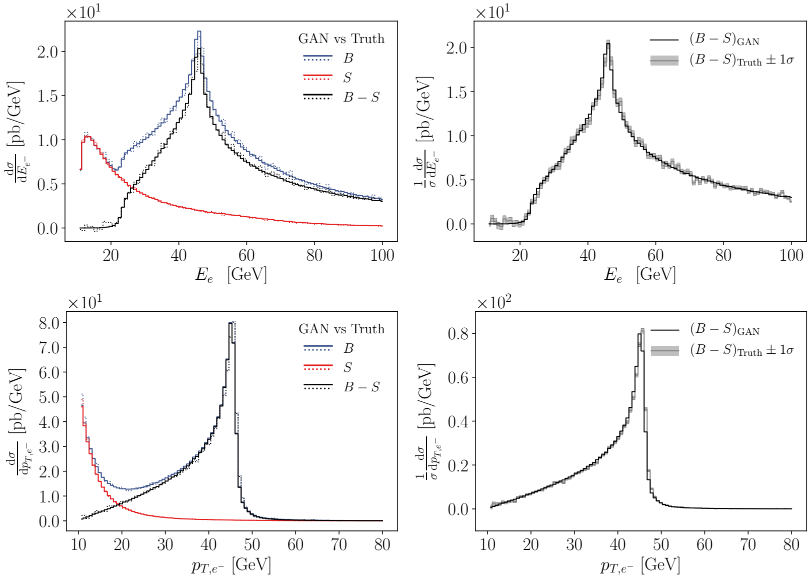
Subtracting LHC events



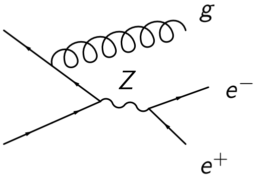
- P_B: $pp \to e^+e^-$
- P_S: $pp \to \gamma \to e^+e^-$
- $p_T > 10$ GeV
- on-shell final state:
  6 dimensional output



Subtracting LHC events

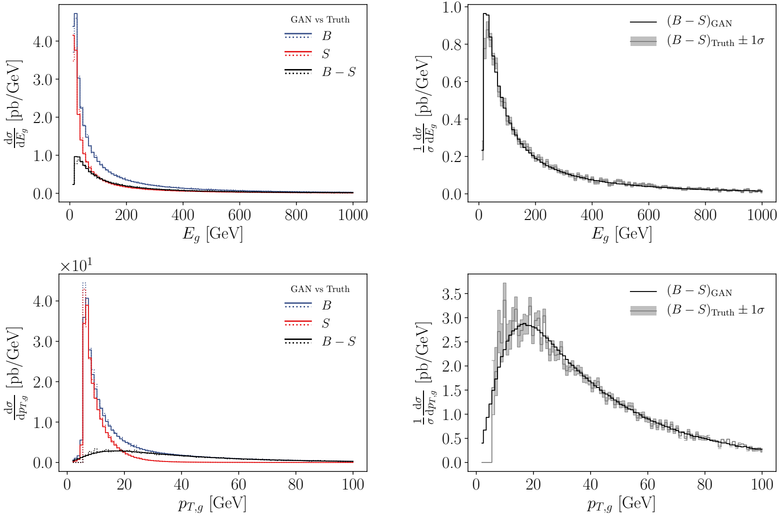


Back to the original problem

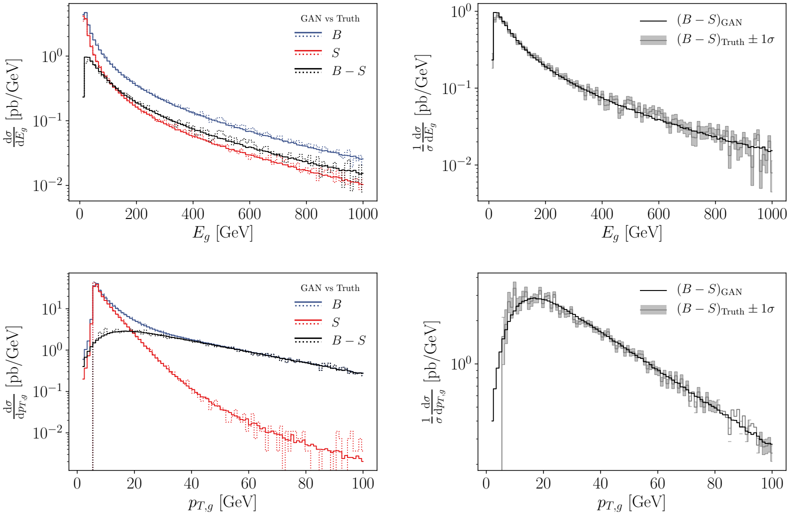


- Subtract the Catany Seymour Dipole from the real emission term
- For proof of concept we use a slightly modifed Catany Seymour kernel $\rightarrow$ increase difference
- Training
  - 10^5 samples per distribution
  - 4-vector representation of Z and g
  - $E_g > 5$ GeV

Results I


GAN vs Truth
B
S
$B-S$
$\frac{d\sigma}{dE_g}$ [pb/GeV]
E_g [GeV]
$(B-S)_{\text{GAN}}$
$(B-S)_{\text{Truth}} \pm 1\sigma$
$\frac{1}{\sigma}\frac{d\sigma}{dE_g}$ [pb/GeV]
$\times 10^1$
$\frac{d\sigma}{dp_{T,g}}$ [pb/GeV]
$p_{T,g}$ [GeV]
$\frac{1}{\sigma}\frac{d\sigma}{dp_{T,g}}$ [pb/GeV]

Results II


GAN vs Truth
B
S
$B-S$
$\frac{d\sigma}{dE_g}$ [pb/GeV]
E_g [GeV]
$(B-S)_{\text{GAN}}$
$(B-S)_{\text{Truth}} \pm 1\sigma$
$\frac{1}{\sigma}\frac{d\sigma}{dE_g}$ [pb/GeV]
$\frac{d\sigma}{dp_{T,g}}$ [pb/GeV]
$p_{T,g}$ [GeV]
$\frac{1}{\sigma}\frac{d\sigma}{dp_{T,g}}$ [pb/GeV]

Conclusion

- HL-LHC results limited by uncertainty on theory prediction
- Need to improve efficiency of computing the subtracted real-emission corrections
- GAN for sample based subtraction
  $\rightarrow$ successful proof of concept!
- Work with Monte Carlo community to test efficiency
- New tool for our ML toolbox
  $\rightarrow$ other use cases?

Summary

- Classification problem solved $\rightarrow$ use ML for new problems

- GANs can learn underlying distributions from event samples
- MMD improves performance for special features

- Generative networks can be used to directly unfold detector level distributions
- Employ FCGAN for notion of locality to enable meaningful slicing

- Successful sample based subtraction implemented
- Test performance for real application

Hyperparameters - Toy1

Parameter	Value
training size	10^5
layers	5
units	128
batch size	1024
learning rate	$3 \cdot 10^{-4}$
decay generator	$5 \cdot 10^{-3}$
decay discriminator	$2 \cdot 10^{-2}$
epochs	4000
discriminator updates	20
α	10
gradient penalty λ_{D_i}	$5 \cdot 10^{-5}$

Hyperparameters - Toy2

Parameter	Value
training size	10^5
layers	7
units	128
batch size	1024
learning rate	$8 \cdot 10^{-4}$
decay generator	$2 \cdot 10^{-2}$
decay discriminator	$2 \cdot 10^{-2}$
epochs	1000
iterations	4
discriminator updates	20
α	5
gradient penalty λ_{D_i}	$5 \cdot 10^{-5}$

Hyperparameters - Resonance

Parameter	Value
training size	10^5
layers	8
G units	160
D units	80
batch size	1024
learning rate	10^{-3}
decay generator	10^{-2}
decay discriminator	10^{-2}
epochs	1000
iterations	5
discriminator updates	2
α	5
gradient penalty λ_{D_i}	10^{-5}

Hyperparameters - Dipole

Parameter	Value
training size	10^5
layers	8
G units	512
D units	256
batch size	1024
learning rate	0.001
decay generator	0.01
decay discriminator	0.01
epochs	20000
iterations	5
discriminator updates	2
α	5
gradient penalty λ_{D_i}	0.001