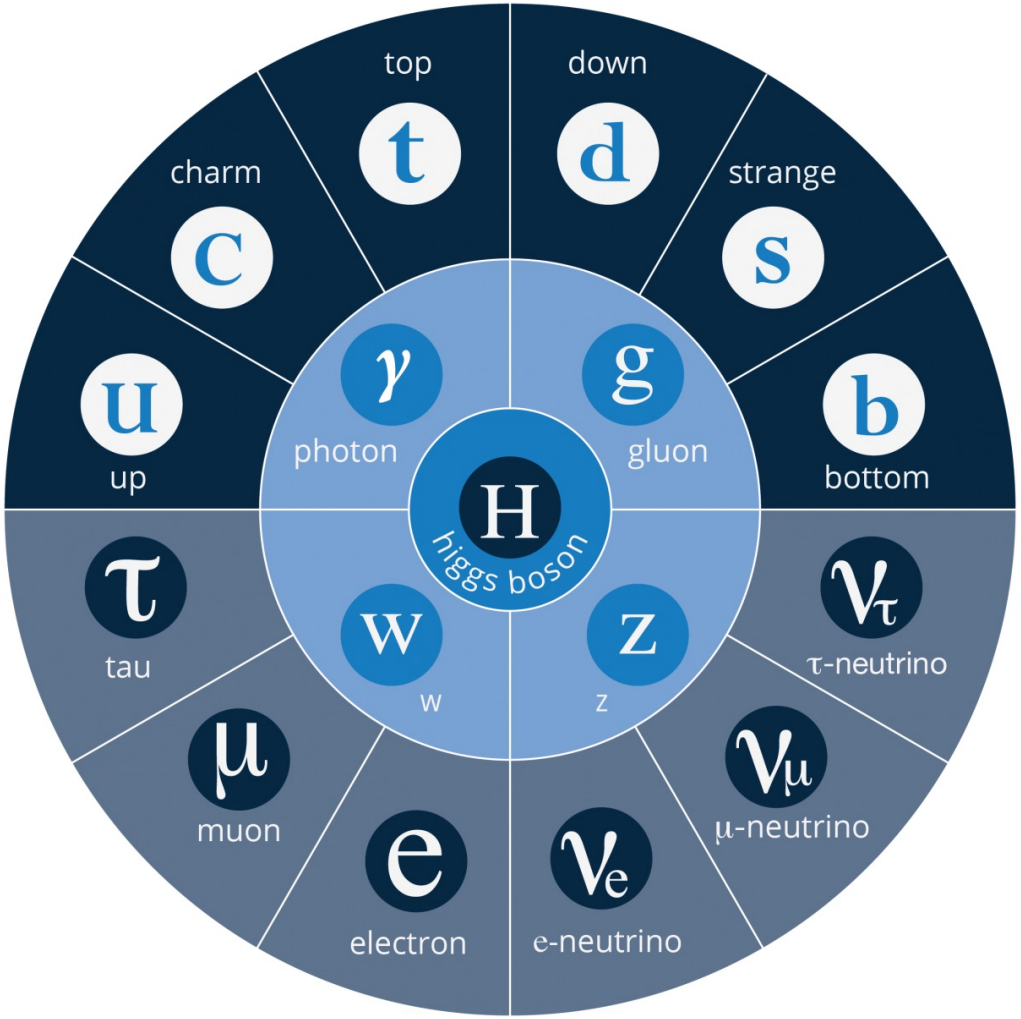
# Machine Learning in Particle Physics

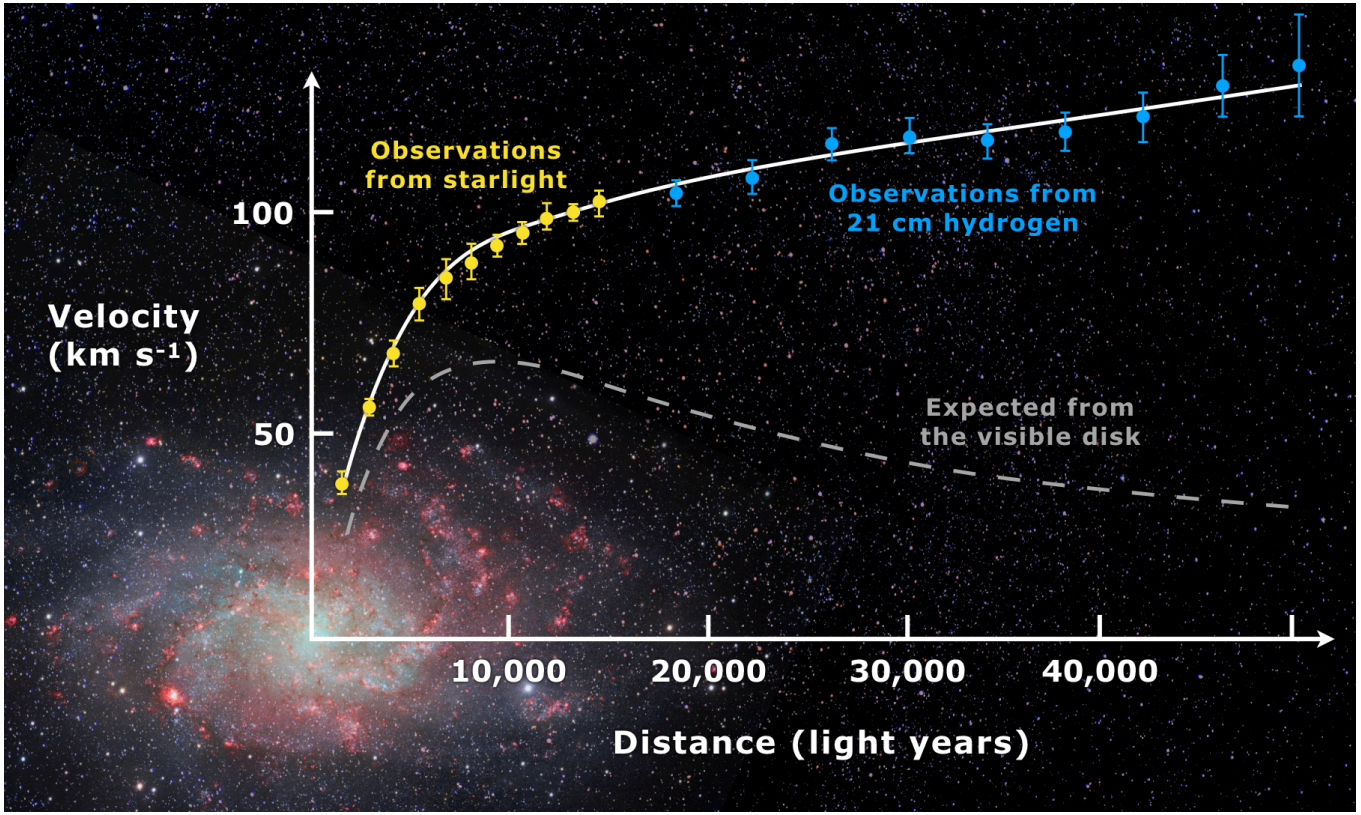
Anja Butter

ITP, Universität Heidelberg

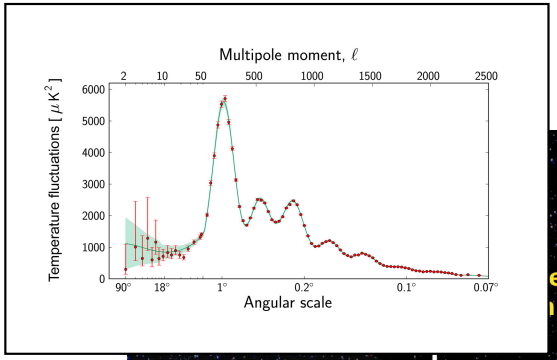
# A structurally complete theory

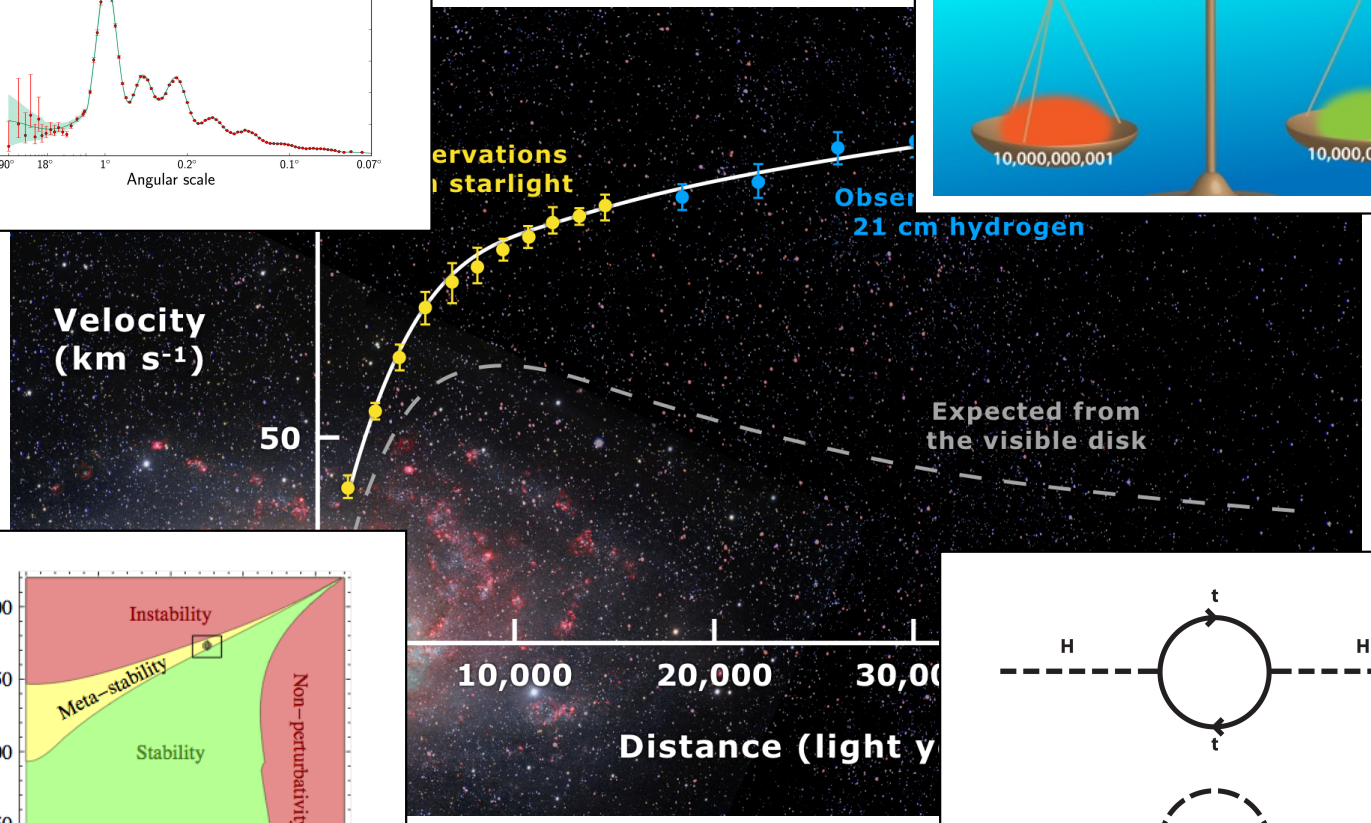


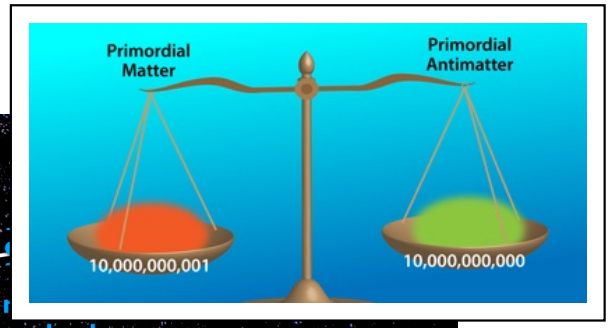
# The need for new physics

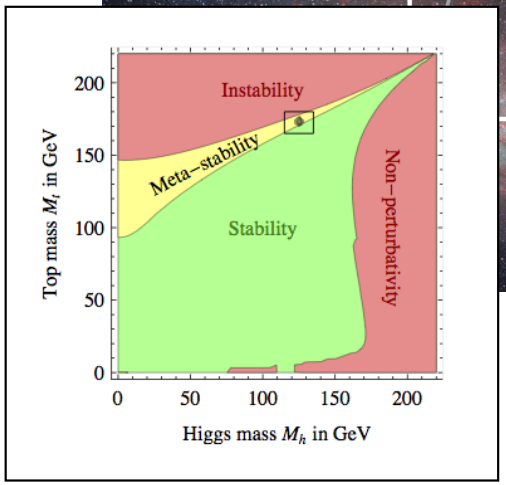


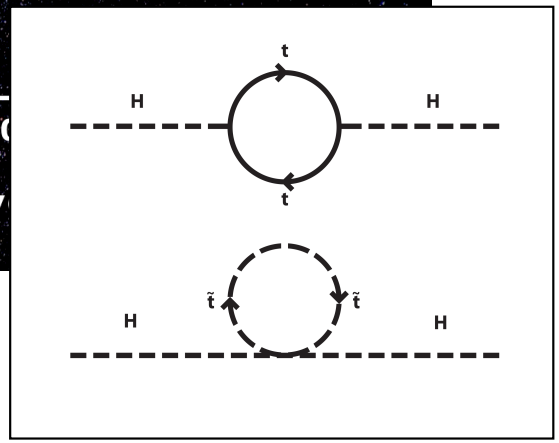
# The need for new physics



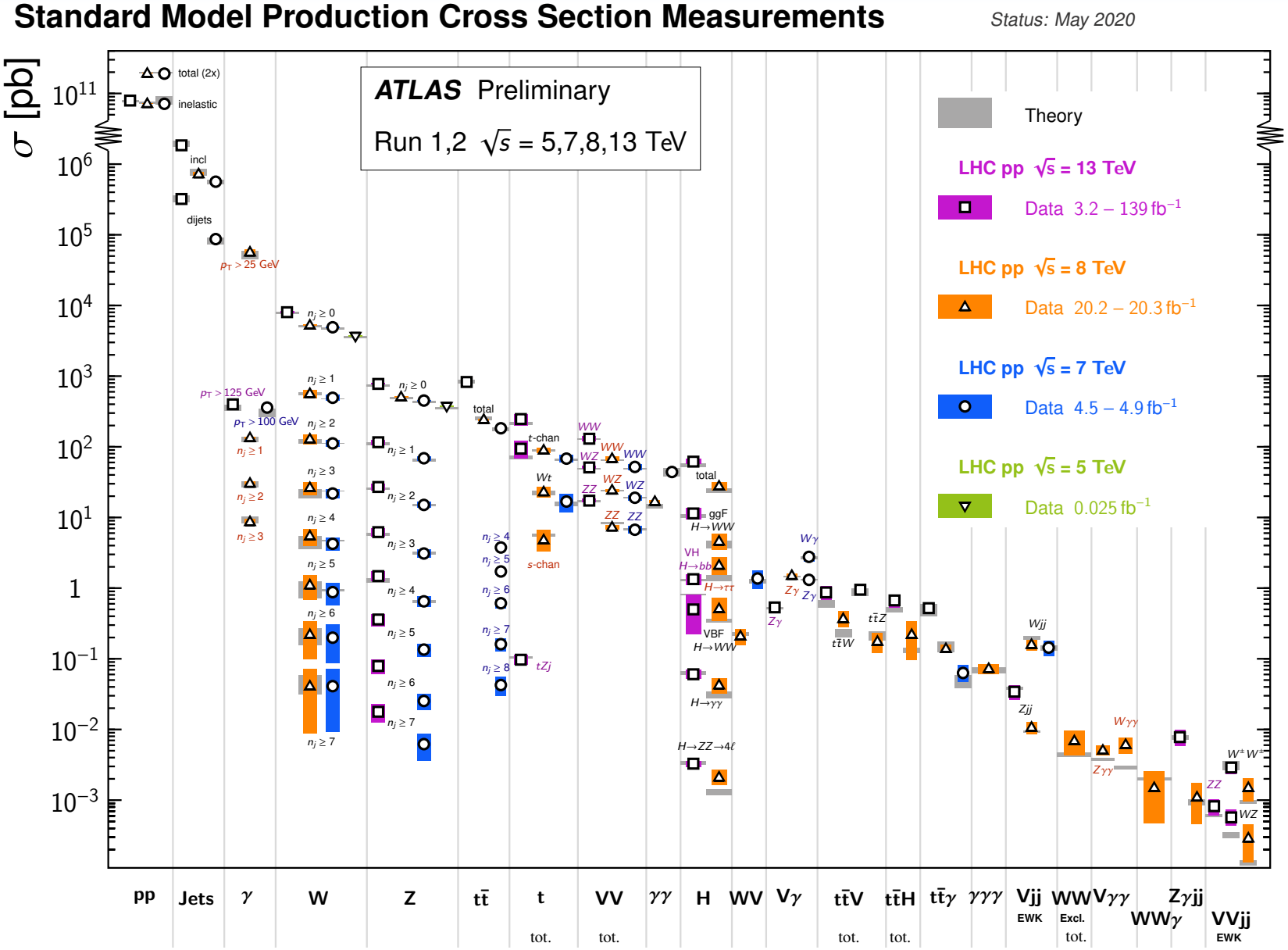



# Era of data

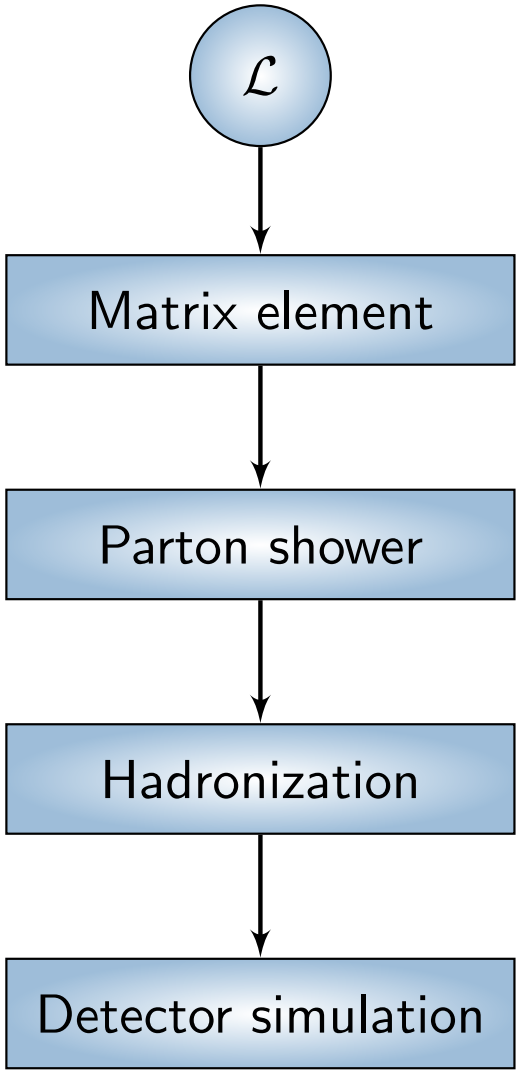

Standard Model Production Cross Section Measurements
Status: May 2020
ATLAS Preliminary
Run 1,2 √s = 5,7,8,13 TeV
σ [pb]
Theory
LHC pp √s = 13 TeV
Data 3.2 – 139 fb⁻¹
LHC pp √s = 8 TeV
Data 20.2 – 20.3 fb⁻¹
LHC pp √s = 7 TeV
Data 4.5 – 4.9 fb⁻¹
LHC pp √s = 5 TeV
Data 0.025 fb⁻¹
pp
Jets
γ
W
Z
tt̄
t
VV
γγ
H
WV
Vγ
tt̄V
tt̄H
tt̄γ
γγγ
Vjj
WWVγγ
Zγjj
WWγ
VVjj

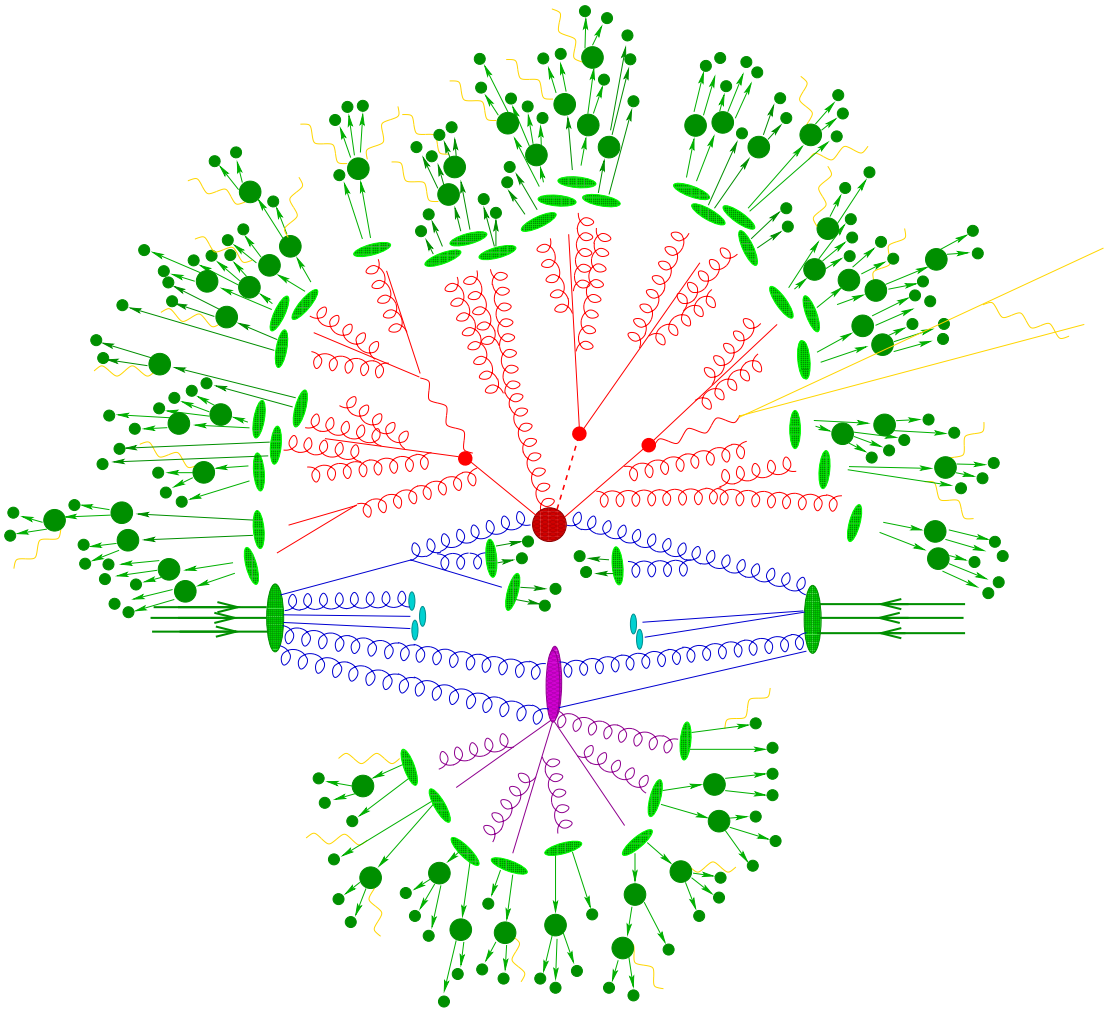
# First principle based event generation

Choose arbitrary Lagrangian like

$$
\begin{aligned}
\mathcal{L}_{\text{eff}} = \mathcal{L}_{\text{SM}} & \\
& - \frac{\alpha_s}{8\pi}\frac{f_{GG}}{\Lambda^2}\phi^\dagger\phi\; G^a_{\mu\nu}G^{a\mu\nu} + \frac{f_{BB}}{\Lambda^2}\phi^\dagger\hat{B}_{\mu\nu}\hat{B}^{\mu\nu}\phi + \frac{f_{WW}}{\Lambda^2}\phi^\dagger\hat{W}_{\mu\nu}\hat{W}^{\mu\nu}\phi \\
& + \frac{f_{\phi,2}}{\Lambda^2}\frac{1}{2}\partial^\mu\left(\phi^\dagger\phi\right)\partial_\mu\left(\phi^\dagger\phi\right) + \frac{f_{WWW}}{\Lambda^2}\,\text{Tr}\left(\hat{W}_{\mu\nu}\hat{W}^{\nu\rho}\hat{W}^{\mu}_{\rho}\right) \\
& + \frac{f_B}{\Lambda^2}(D_\mu\phi)^\dagger\hat{B}^{\mu\nu}(D_\nu\phi) + \frac{f_W}{\Lambda^2}(D_\mu\phi)^\dagger\hat{W}^{\mu\nu}(D_\nu\phi) \\
& + \frac{f_\tau m_\tau}{v\Lambda^2}(\phi^\dagger\phi)(\bar{L}_3\phi e_{R,3}) + \frac{f_b m_b}{v\Lambda^2}(\phi^\dagger\phi)(\bar{Q}_3\phi d_{R,3}) + \frac{f_t m_t}{v\Lambda^2}(\phi^\dagger\phi)(\bar{Q}_3\tilde{\phi} u_{R,3})
\end{aligned}
$$

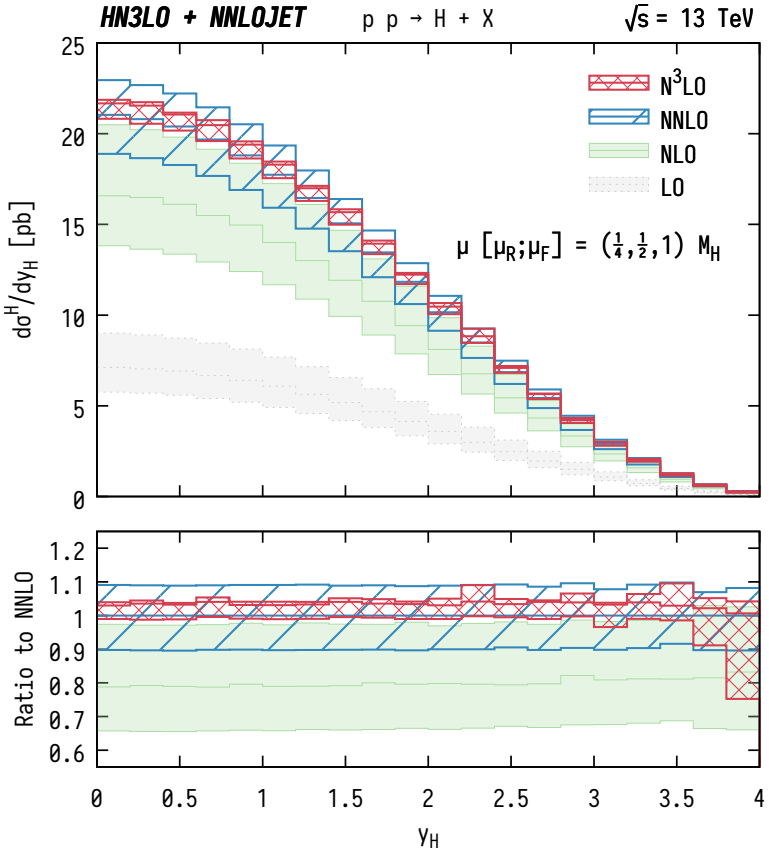
# First principle based event generation



a sherpa artist

# Precision simulations




[1807.11501] Cieri, Chen, Gehrmann, Glover, Huss

# New physics is hidden

Big data
HL-LHC = 25×Run2

+

First-principle
precision simulation

↓

Make best use of both!

↓

Big data techniques

# Ahead of time

**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**

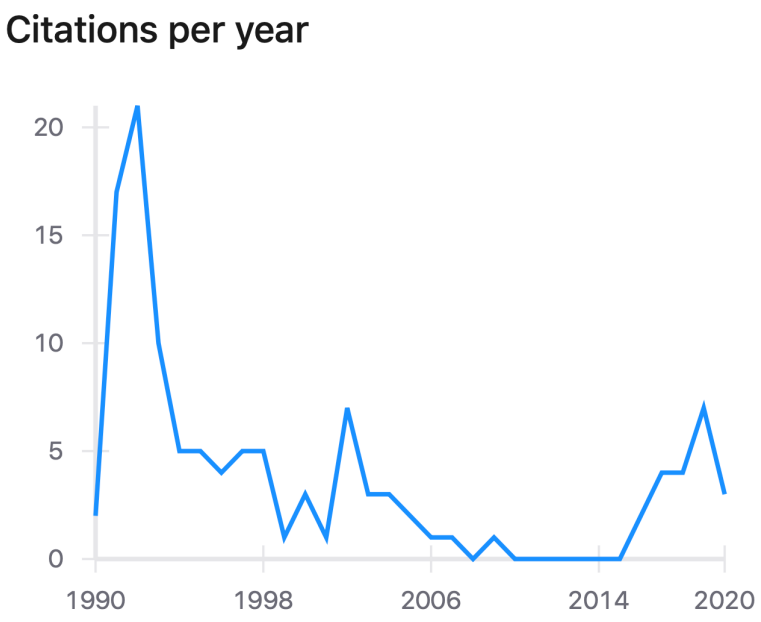
# Ahead of time

**USING NEURAL NETWORKS TO IDENTIFY JETS**

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

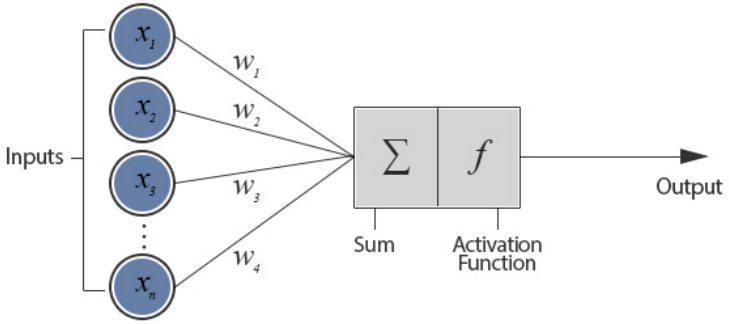
*Department of Theoretical Physics, University of Lund, Sölvegatan 14A, S-22362 Lund, Sweden*

**Received 29 June 1990**



# How can ML help increasing precision

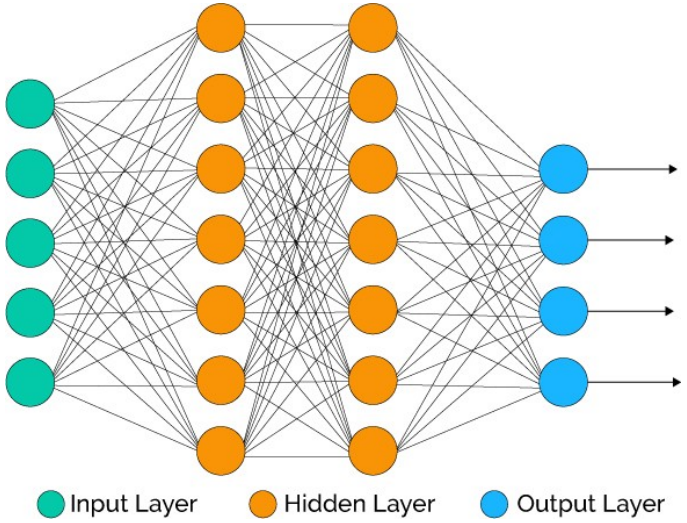
- 1.0 Classification/Regression
  $\rightarrow$ Label data




minimize $L = (y_{true} - y_{output})^2$

# How can ML help increasing precision

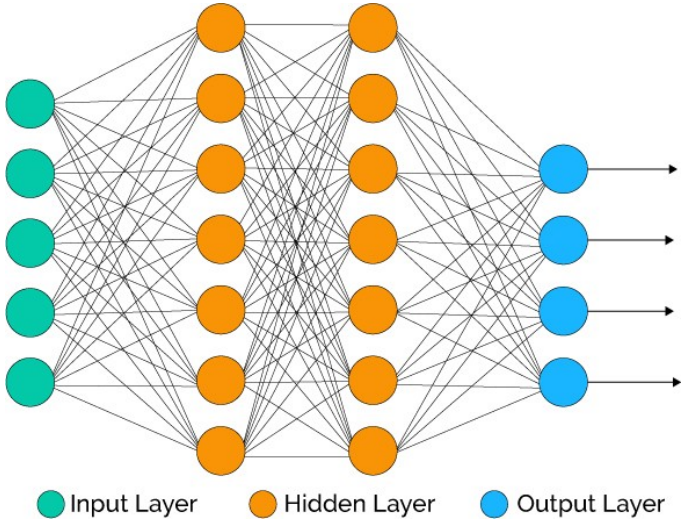
- 1.0 Classification/Regression
  $\rightarrow$ Label data




minimize $L = (y_{true} - y_{output})^2$

# How can ML help increasing precision

- 1.0 Classification/Regression
  $\rightarrow$ Label data




minimize $L = (y_{true} - y_{output})^2$

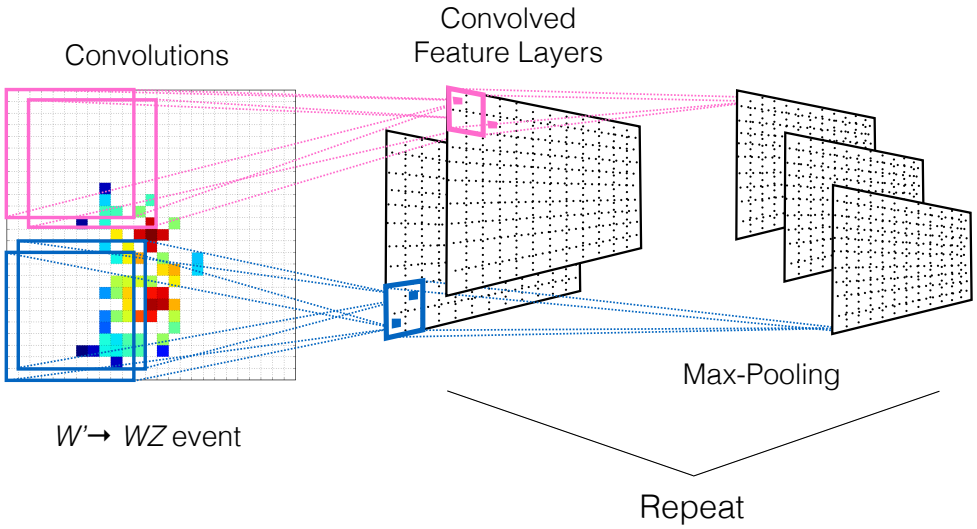
$+$ low level observables
$+$ efficient training

Why now? $\rightarrow$ GPUs
$\rightarrow$ new algorithms [convolutional networks]

# First application - jet tagging

Convolutional network on W/QCD jet images

+ Physics: theoretical and experimental control
+ Straight forward from ML developments




[1511.05190] L. Oliveira, M. Kagan, L. Mackey, B. Nachman, A. Schwartzman

# Top tagging with physics networks

- W $\to$ top jets
- Lo(rentz) La(yer):
  Lorentz vectors $\to$ physics motivated objects

[1707.08966] AB, G. Kasieczka, T. Plehn, M. Russell

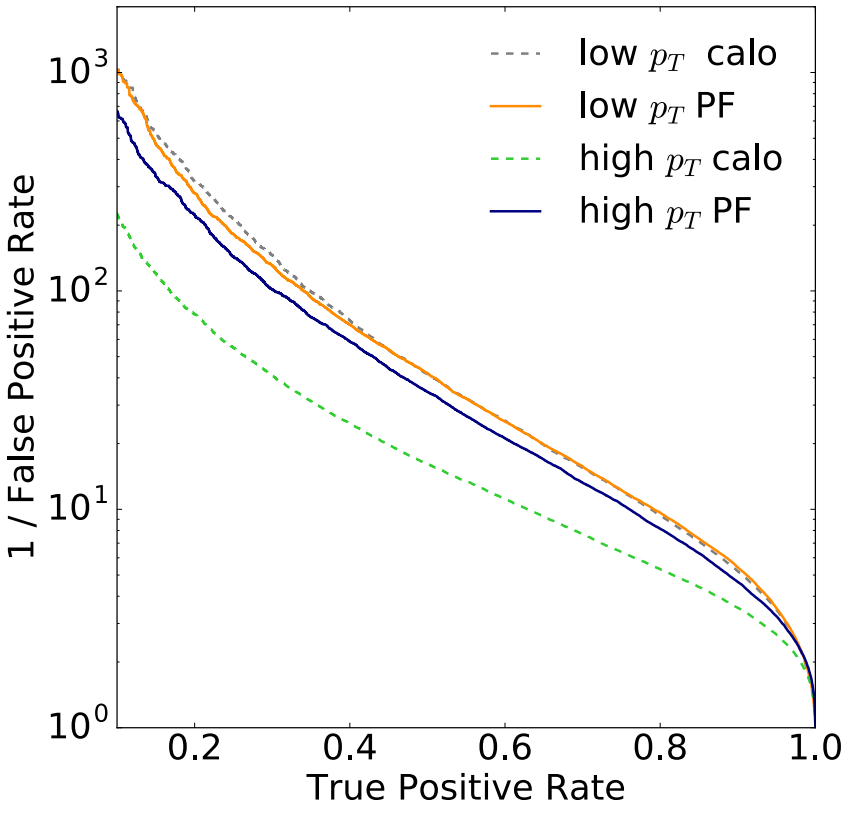
$$\tilde{k}_j \xrightarrow{\text{LoLa}} \hat{k}_j = \begin{pmatrix} m^2(\tilde{k}_j) = \tilde{k}_{j,\mu}\, \eta^{\mu\nu}\, \tilde{k}_{j,\nu} \\ p_T(\tilde{k}_j) \\ w_{jm}^{(E)}\, E(\tilde{k}_m) \\ w_{jm}^{(d)}\, d_{jm}^2 \end{pmatrix}$$

$$\text{with} \qquad d_{jm}^2 = (\tilde{k}_j - \tilde{k}_m)_\mu\, \eta^{\mu\nu}\, (\tilde{k}_j - \tilde{k}_m)_\nu$$

Training yields:

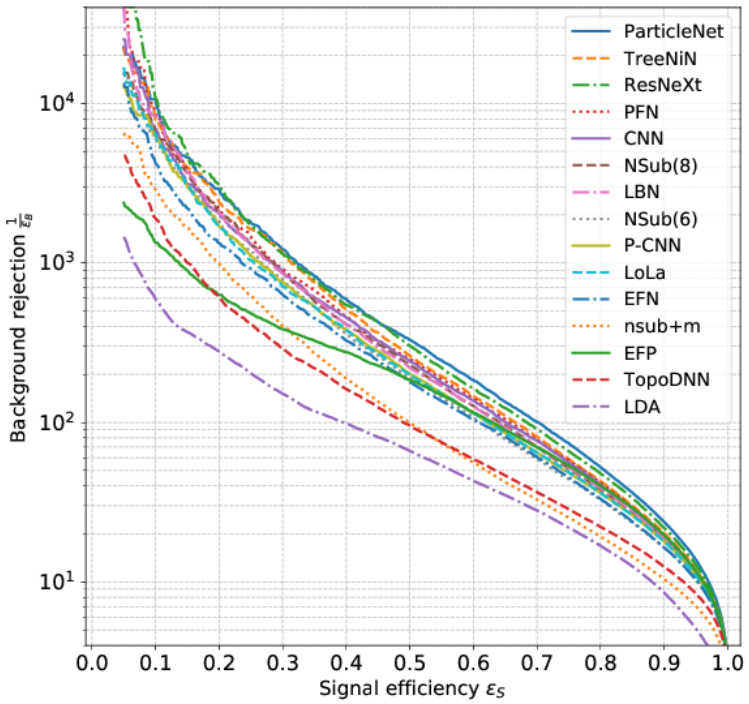
$$\eta = \text{diag}(\quad 0.99 \pm 0.02, -1.01 \pm 0.01, -1.01 \pm 0.02, -0.99 \pm 0.02)$$

# LoLa vs Image




- Combine tracking & calorimeter information
- Improved performance for boosted jets
- Recent network: ParticleNet [1902.08570] H. Qu, L. Gouskos

# Comparative top tagging study




[1707.08966] G. Kasieczka, et al.

→ Other applications: jet calibration, particle identification, ...

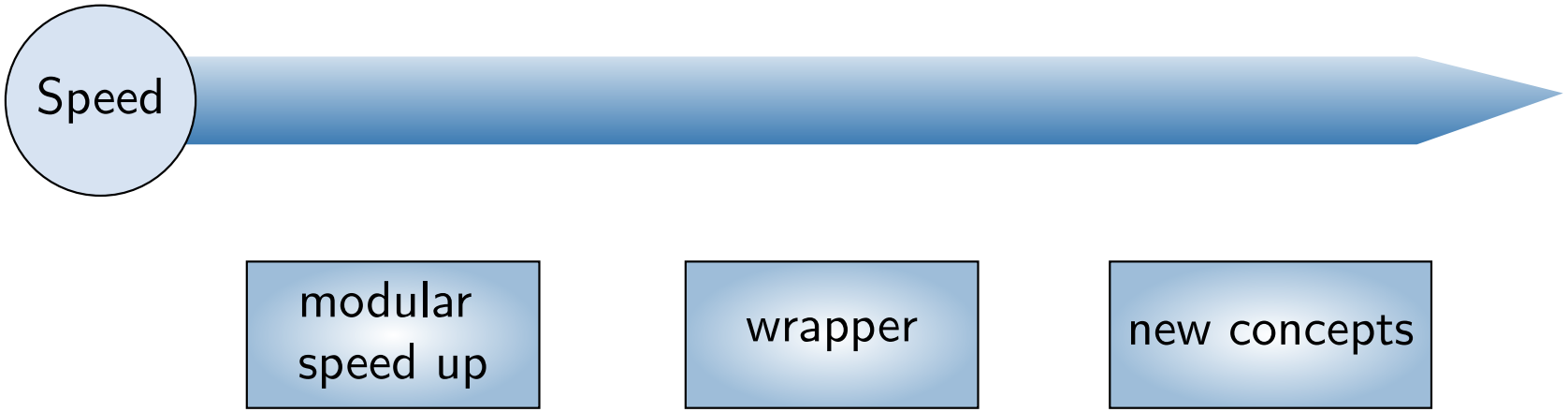
→ Open questions: precision, uncertainties, visualization

# Precision in forward simulations

- ML 2.0 Generative models
  - → Can we simulate new data?

Speed → more events / higher order → Precision

more events = higher order



# Boosting standard event generation...

1. Generate phase space points

2. Calculate event weight

$$w_{event} = f(x_1, Q^2) f(x_2, Q^2) \times \mathcal{M}(x_1, x_2, p_1, \ldots p_n) \times J(p_i(r))^{-1}$$

3. Unweighting via importance sampling
$\rightarrow$ optimal for $w \approx 1$

# Boosting standard event generation...

Matrix element

$$w_{event} = f(x_1, Q^2) f(x_2, Q^2) \times \mathcal{M}(x_1, x_2, p_1, \ldots p_n) \times J(p_i(r))^{-1}$$

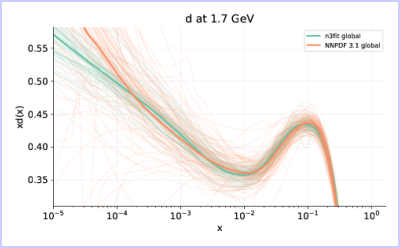
PDF

Phase space mapping

# Boosting standard event generation...

Matrix element

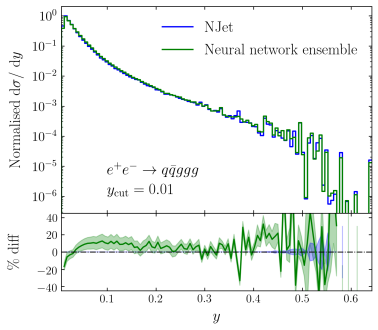
$$w_{event} = f(x_1, Q^2) f(x_2, Q^2) \times \mathcal{M}(x_1, x_2, p_1, \dots p_n) \times J(p_i(r))^{-1}$$




- NNPDF since 2002(!)
- S. Carrazza, J. Cruz-Martinez [1907.05075]

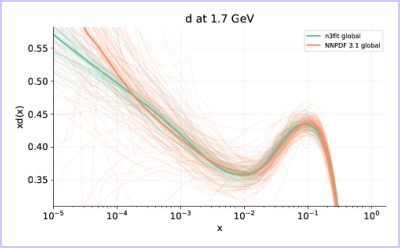
Phase space mapping

# Boosting standard event generation...




- Amplitude estimation
- S. Badger, J. Bullock [2002.07516]
- J. Bendavid [1707.00028]

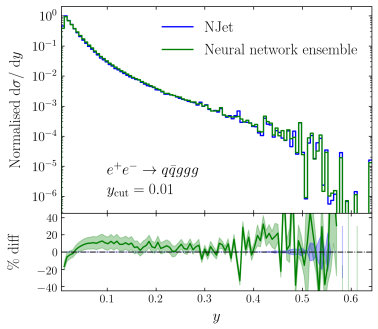
$$w_{event} = f(x_1, Q^2) f(x_2, Q^2) \times \mathcal{M}(x_1, x_2, p_1, \dots p_n) \times J(p_i(r))^{-1}$$




- NNPDF since 2002(!)
- S. Carrazza, J. Cruz-Martinez [1907.05075]

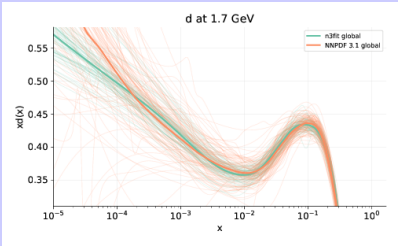
Phase space mapping

# Boosting standard event generation...

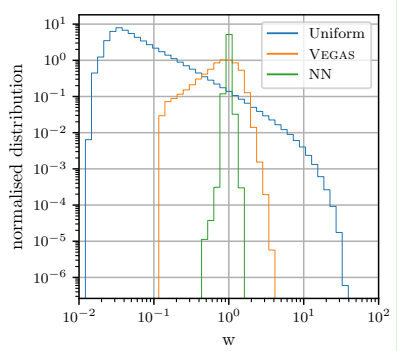



- Amplitude estimation
- S. Badger, J. Bullock [2002.07516]
- J. Bendavid [1707.00028]

$$w_{event} = f(x_1, Q^2) f(x_2, Q^2) \times \mathcal{M}(x_1, x_2, p_1, \dots p_n) \times J(p_i(r))^{-1}$$




- NNPDF since 2002(!)
- S. Carrazza, J. Cruz-Martinez [1907.05075]




- Learn phase space mapping ($\to w \approx 1$)
- Gao et al. [2001.10028]
- Bothmann et al. [2001.05478]

# ... or training directly on event samples

## Event generation

- Generating 4-momenta
- $Z > ll$, $pp > jj$, $pp > t\bar{t}$+decay

[1901.00875] Otten et al. **VAE** & **GAN**

[1901.05282] Hashemi et al. **GAN**

[1903.02433] Di Sipio et al. **GAN**

[1903.02556] Lin et al. **GAN**

[1907.03764, 1912.08824] Butter et al. **GAN**

[1912.02748] Martinez et al. **GAN**

[2001.11103] Alanazi et al. **GAN**

## Detector simulation

- Jet images
- Fast shower simulation in calorimeters

[1701.05927] de Oliveira et al. **GAN**

[1705.02355, 1712.10321] Paganini et al. **GAN**

[1802.03325, 1807.01954] Erdmann et al. **GAN**

[1805.00850] Musella et al. **GAN**

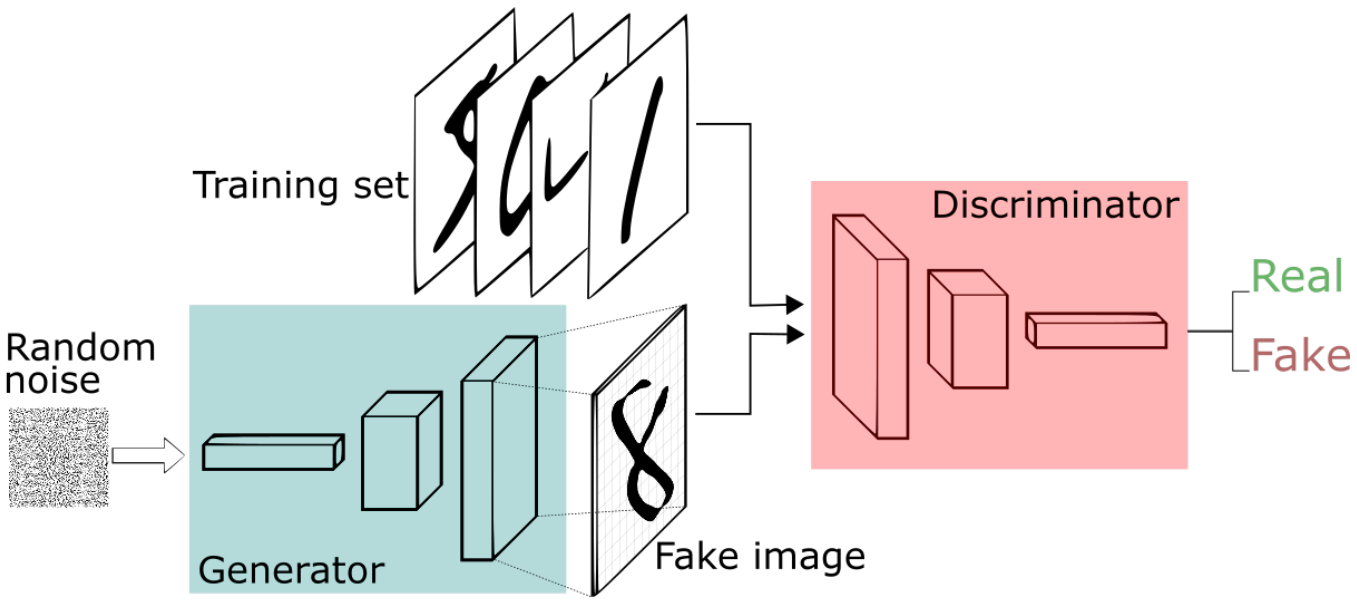
[ATL-SOFT-PUB-2018-001, ATLAS-SIM-2019-004, ATL-SOFT-PROC-2019-007] ATLAS **VAE** & **GAN**

[1909.01359] Carazza and Dreyer **GAN**

[2005.05334] Buhmann et al. **VAE**

NO claim to completeness!

# Generative Adversarial Networks




**Discriminator**

$$L_D = \langle -\log D(x) \rangle_{x \sim P_{Truth}} + \langle -\log(1 - D(x)) \rangle_{x \sim P_{Gen}}$$

**Generator**

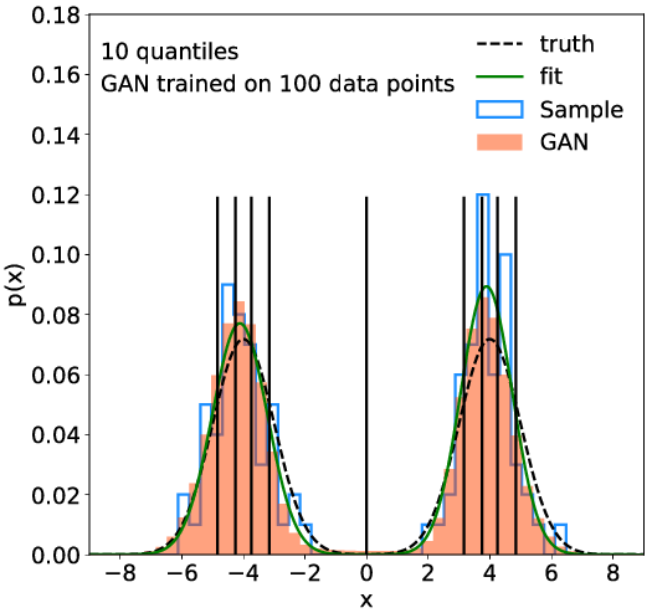
$$L_G = \langle -\log D(x) \rangle_{x \sim P_{Gen}}$$

# What is the statistical value of GANned events?

- Camel function
- Sample vs. GAN vs. 5 param.-fit

Evaluation on quantiles:

$$\mathrm{MSE}^* = \sum_{j=1}^{N_{\mathrm{quant}}} \left( p_j - \frac{1}{N_{\mathrm{quant}}} \right)^2$$

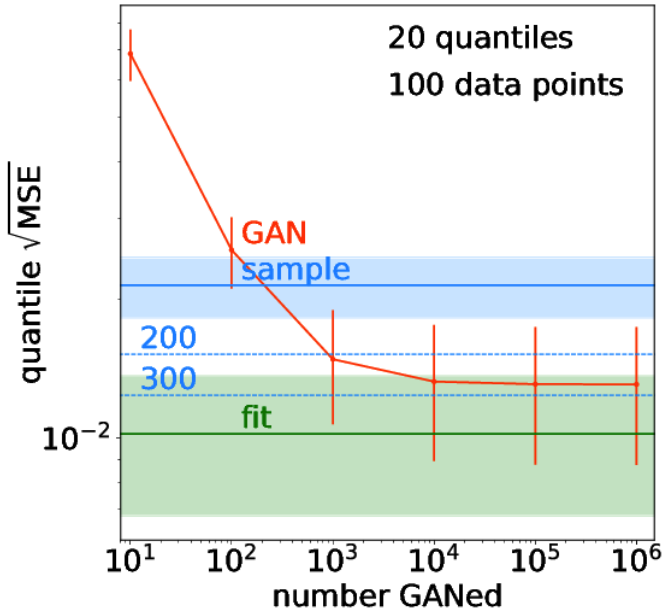


# What is the statistical value of GANned events?

- Camel function
- Sample vs. GAN vs. 5 param.-fit

Evaluation on quantiles:

$$\mathrm{MSE}^* = \sum_{j=1}^{N_{\mathrm{quant}}} \left( p_j - \frac{1}{N_{\mathrm{quant}}} \right)^2$$

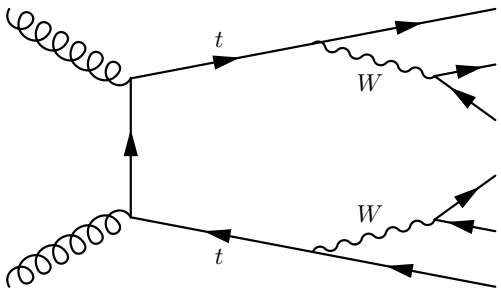


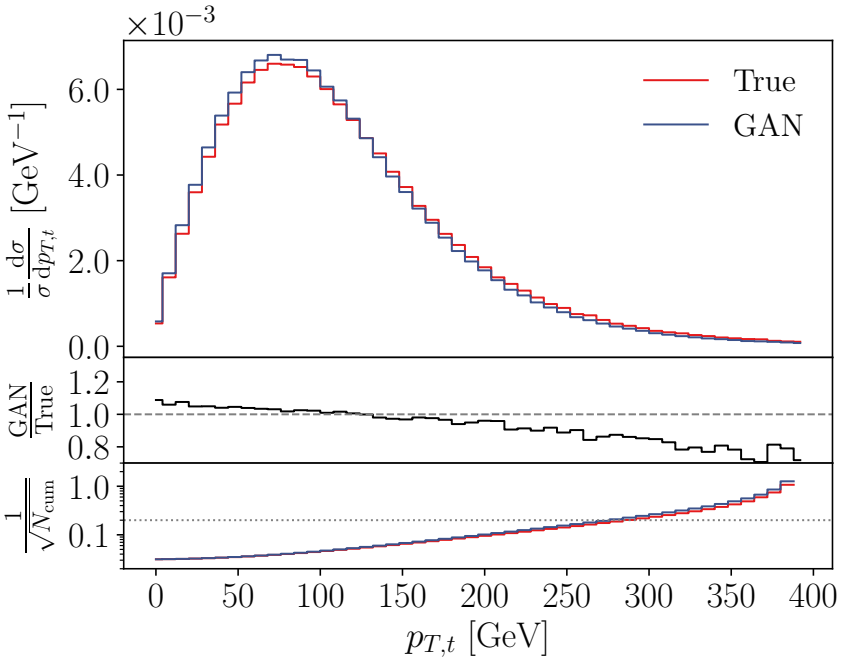

$\rightarrow$ Amplification factor 2.5

Sparser data $\rightarrow$ bigger amplification

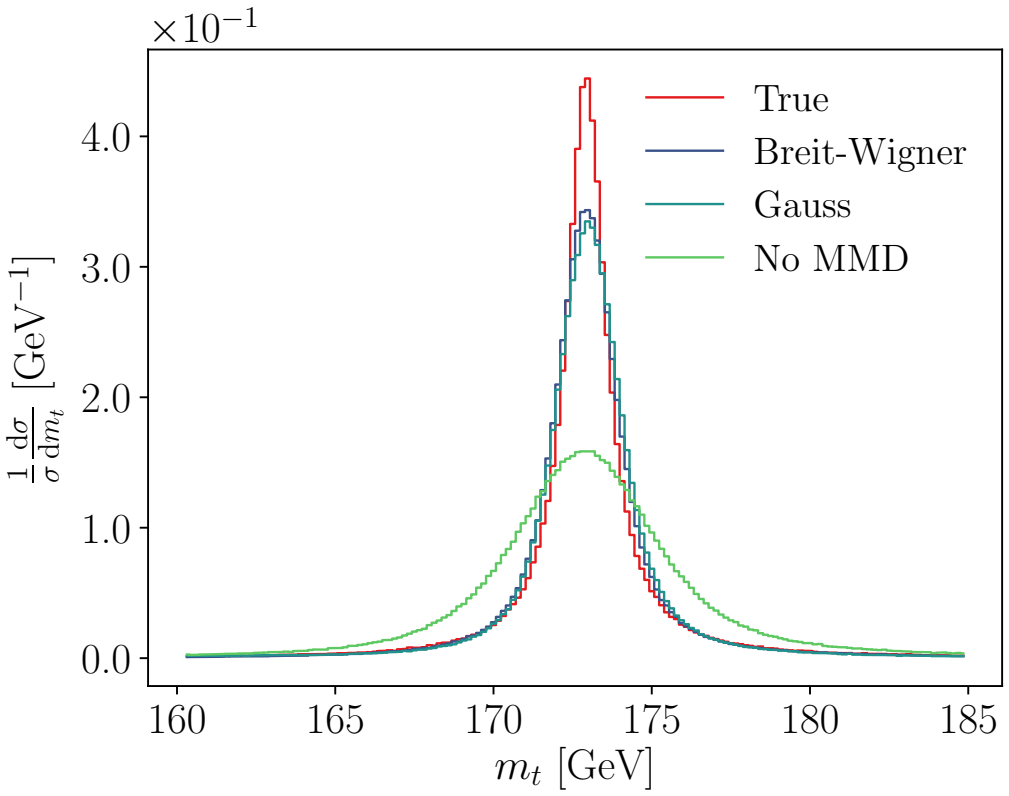
# How to GAN LHC events

- $t\bar{t} \to 6$ quarks
+ Flat observables ✓
+ Narrow structures $\to$ kernel loss ✓
– Systematic undershoot in tails [10-20% deviation]



# Special features




**Solution: MMD kernel**

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

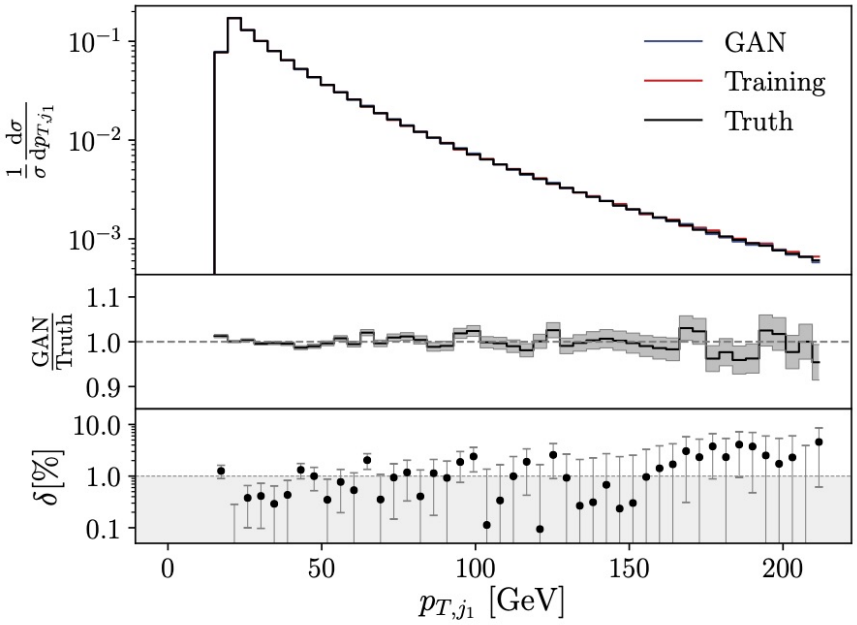
# Reaching precision (preliminary)

1. Representation $p_T, \eta, \phi$
2. Momentum conservation
3. Resolve $\log p_T$
4. Regularization: spectral norm
5. Batch information

$\rightarrow$ 1% precision ✓

Automization?

W + 2 jets

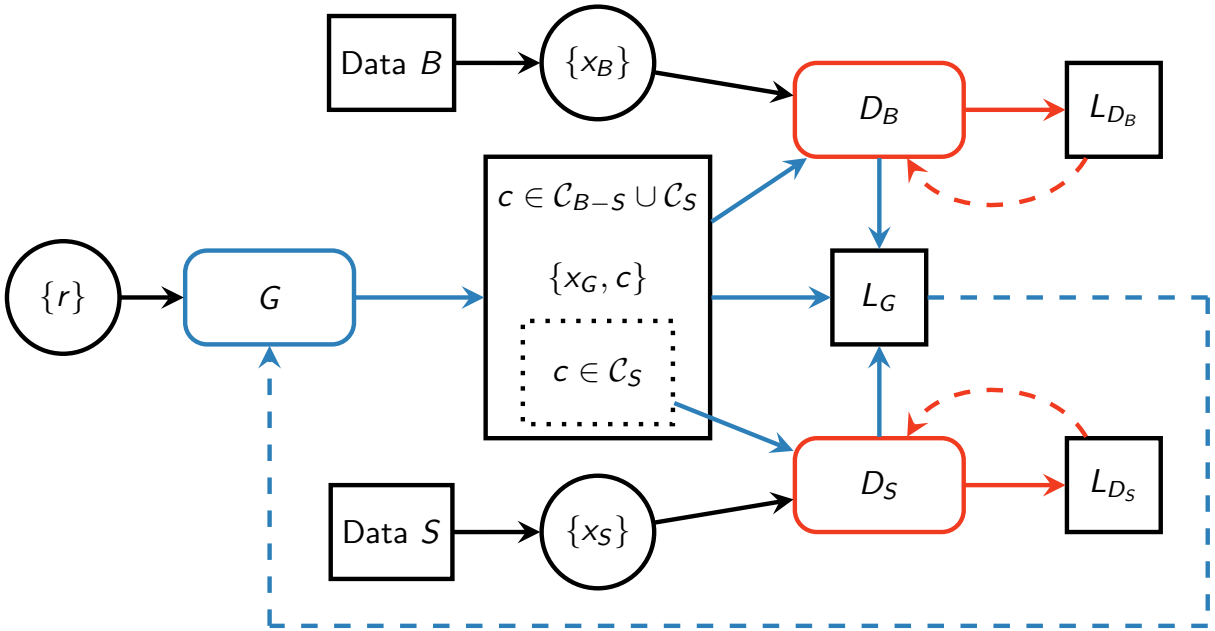


# Negative events

- Generate high-dim. difference distributions
- Beat bin-induced uncertainty

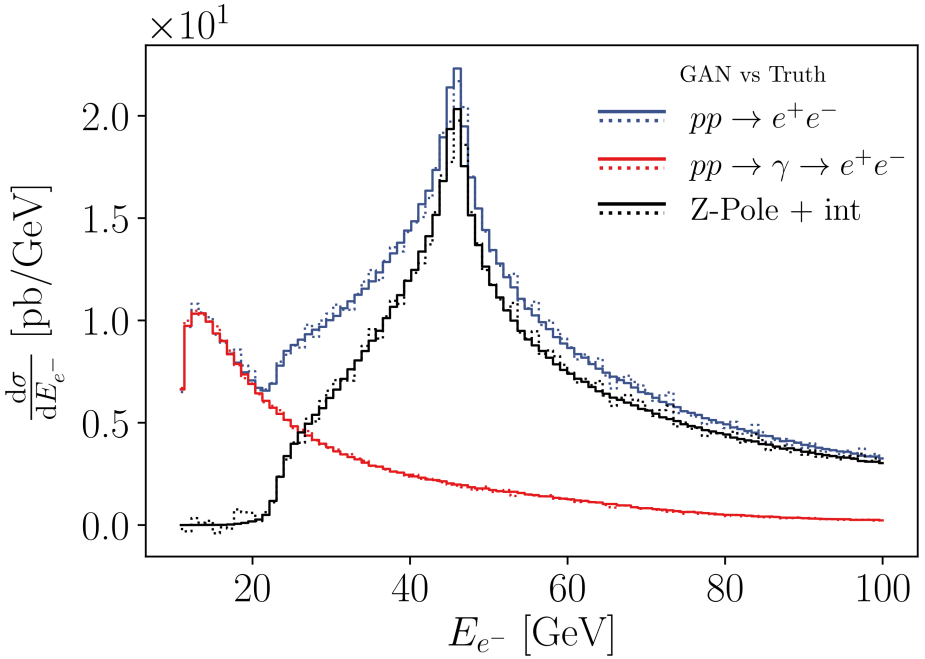
$$\Delta_{B-S} > \max(\Delta_B, \Delta_S)$$

- Applications:
  - Background subtraction, soft-collinear subtraction, ...



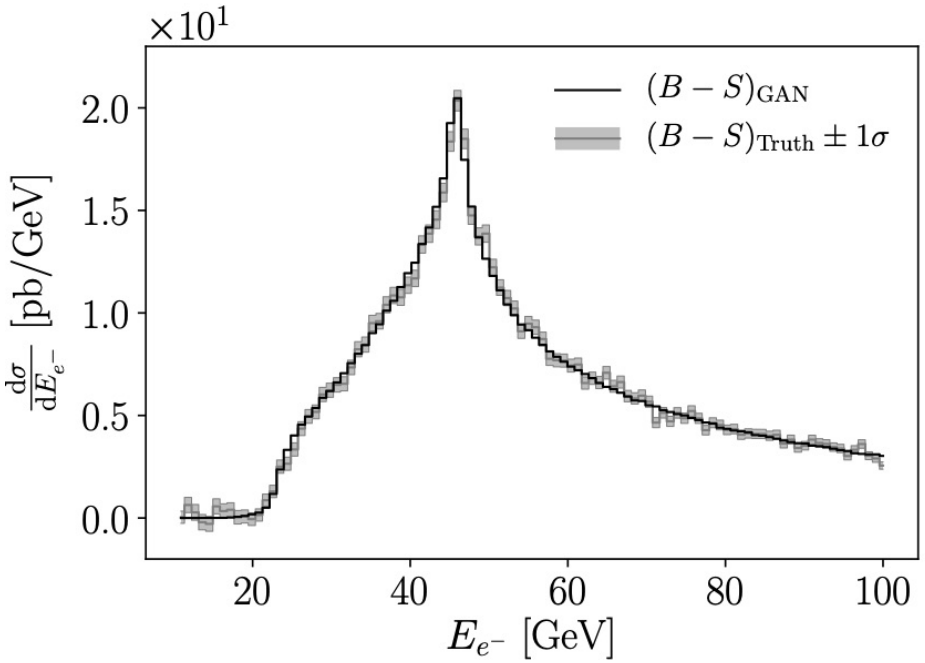
# Generative background subtraction

- Training data:
  - $pp \to e^+e^-$
  - $pp \to \gamma \to e^+e^-$
- Generated events: Z-Pole + interference

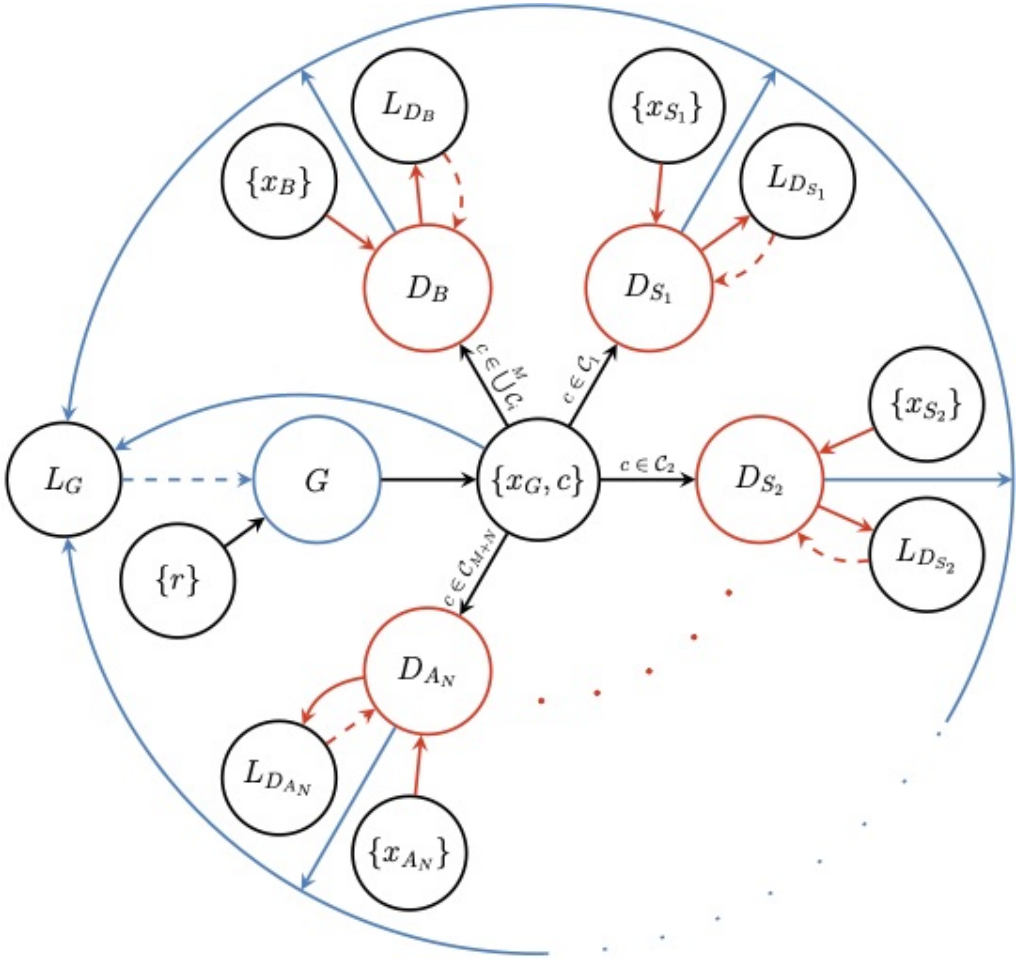


# Generative background subtraction

- Training data:
  - $pp \to e^+e^-$
  - $pp \to \gamma \to e^+e^-$
- Generated events: Z-Pole + interference

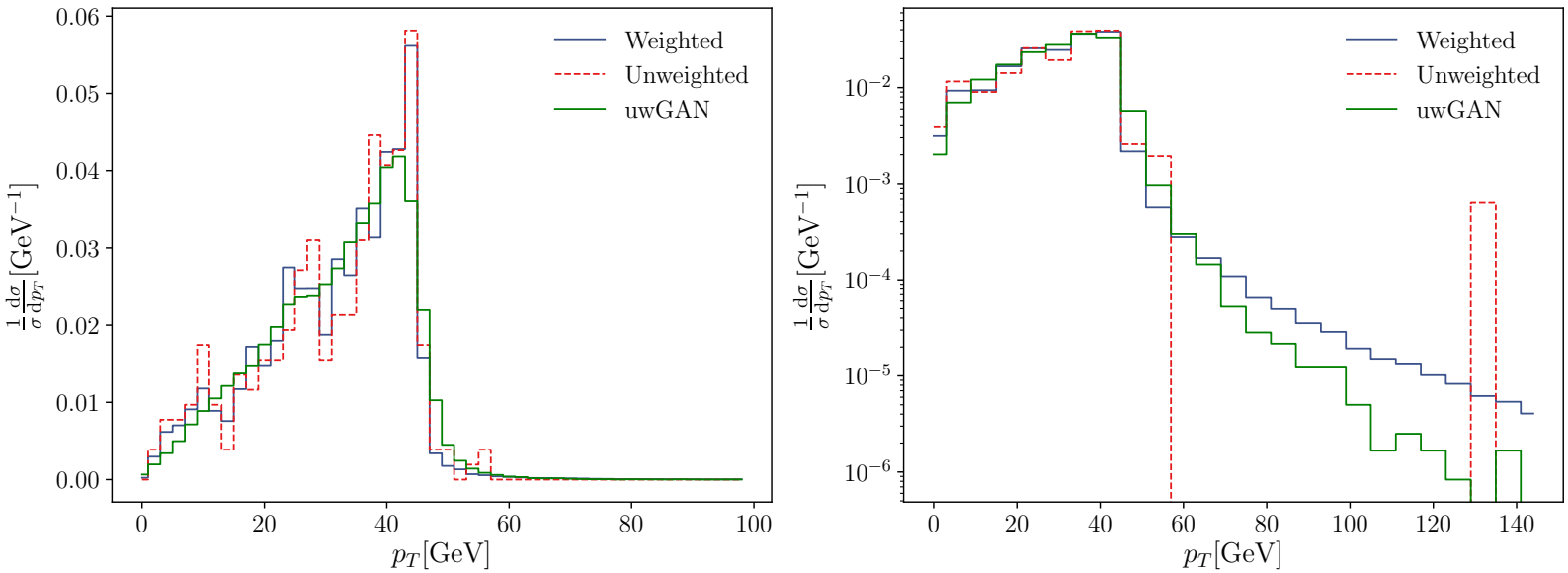


# General Subtraction GAN



# Training on weighted events (preliminary)

$$L_D = \left\langle -w \log D(x) \right\rangle_{x \sim P_{Truth}} + \left\langle -\log(1 - D(x)) \right\rangle_{x \sim P_{Gen}}$$



# Inverting a Markov process

$$x_{part} \xleftrightarrow[\leftarrow \text{ inversion}:\bar{g}]{\text{PYTHIA,DELPHES}:g\rightarrow} x_{det}$$

$$\downarrow$$

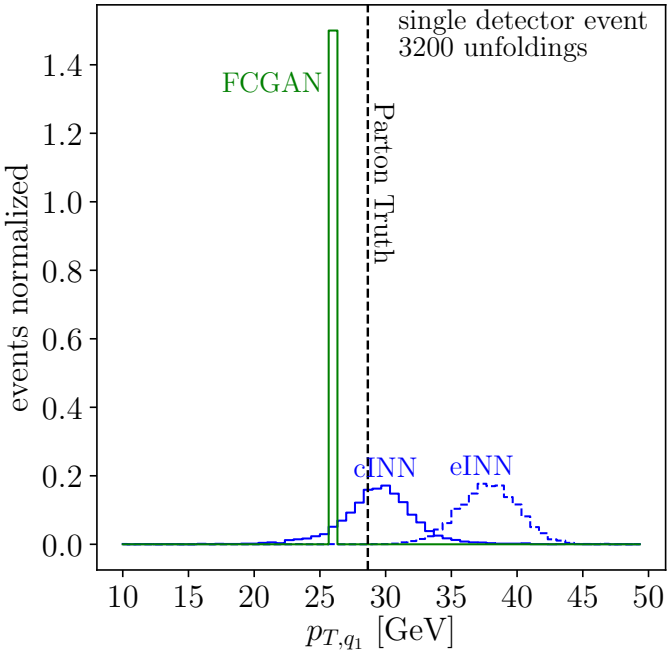
$$L = \min_{\theta} - \langle \log p(\theta | x_{part}, x_{det}) \rangle_{x_{part} \sim P_p, x_{det} \sim P_d}$$

$$\downarrow$$

conditional invertible network $g$

$$\downarrow$$

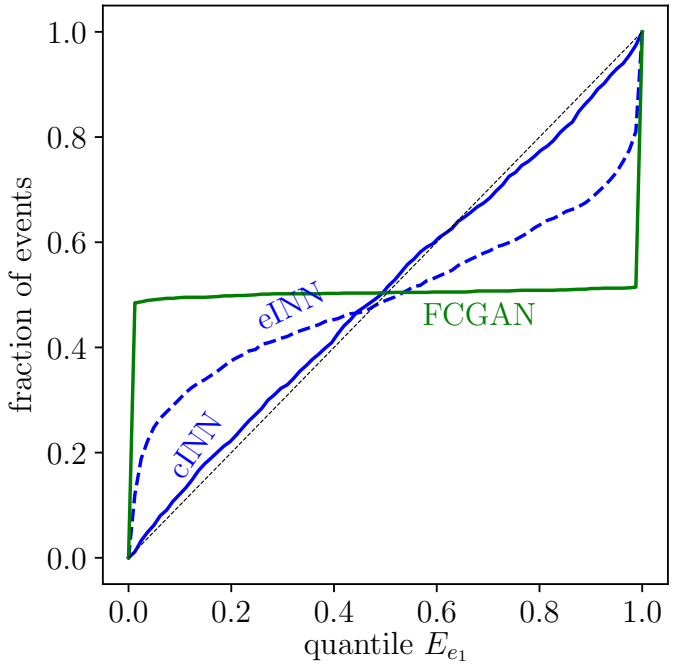
$$x_p \xleftrightarrow[\leftarrow \text{ inversion: } \bar{g}(r, f(x_d))]{g(x_p, f(x_d)) \rightarrow} r$$



# Calibration

Unfold single event many times $\rightarrow$ distribution

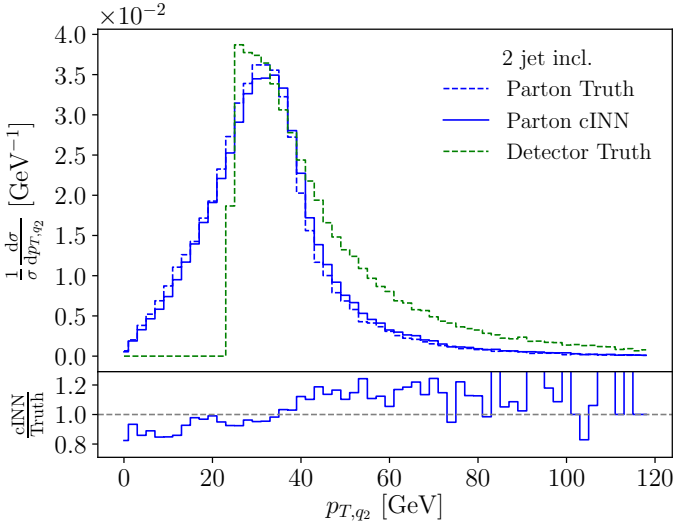
Fraction of events in quantile of unfolded distribution



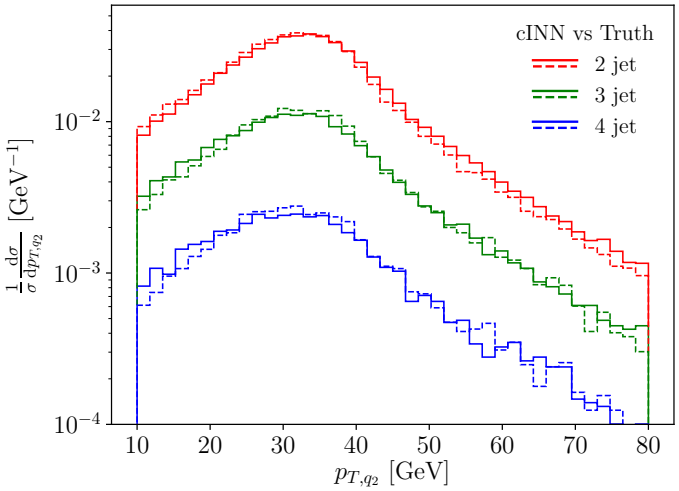
# Inverting the full event

$pp > WZ > q\bar{q}l^+l^- + \text{ISR}$

Train on full dataset




Evaluate
exclusive 2/3/4 jet channels



# We can use ML to ...

... exploit data in best possible way

... explore new opportunities in HEP

... improve analyses with optimized S vs B classification

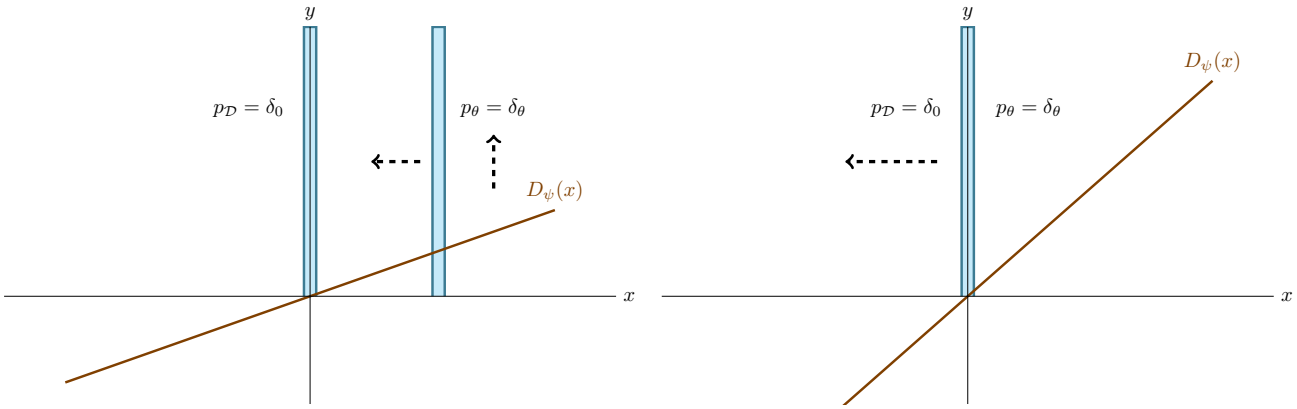
... enable precision simulations in forward direction

... unfold high dimensions

... learn more about particle physics!

# BACK UP

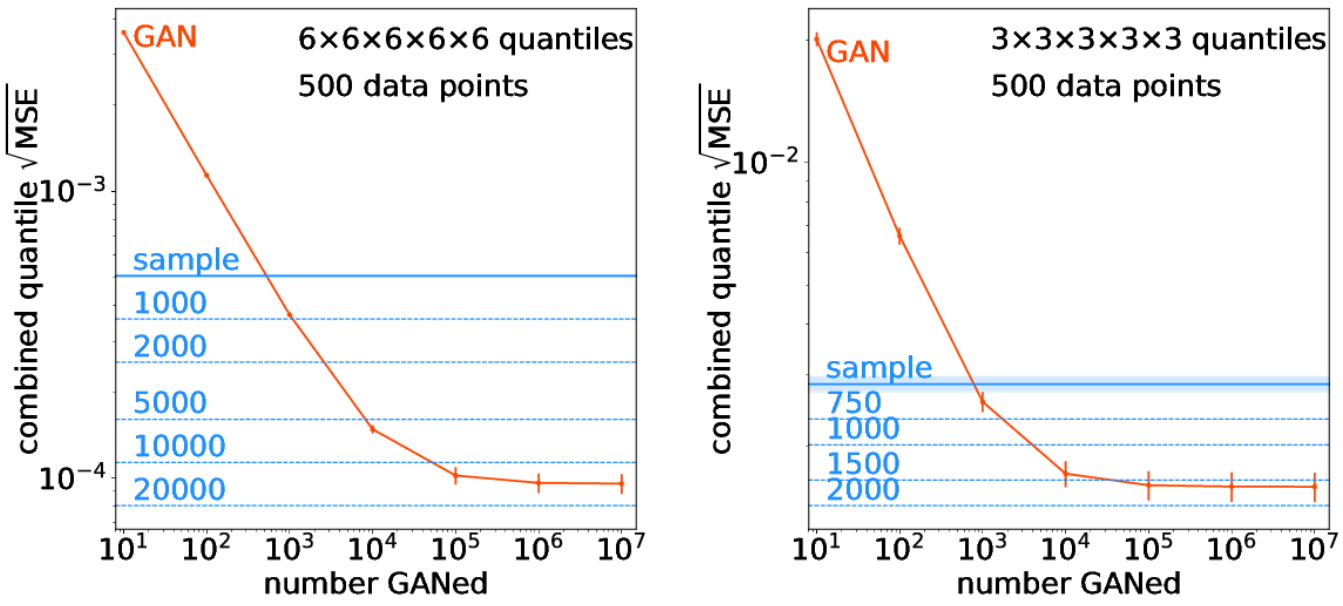
# The GAN challenge
# or
# Why do we need regularization?



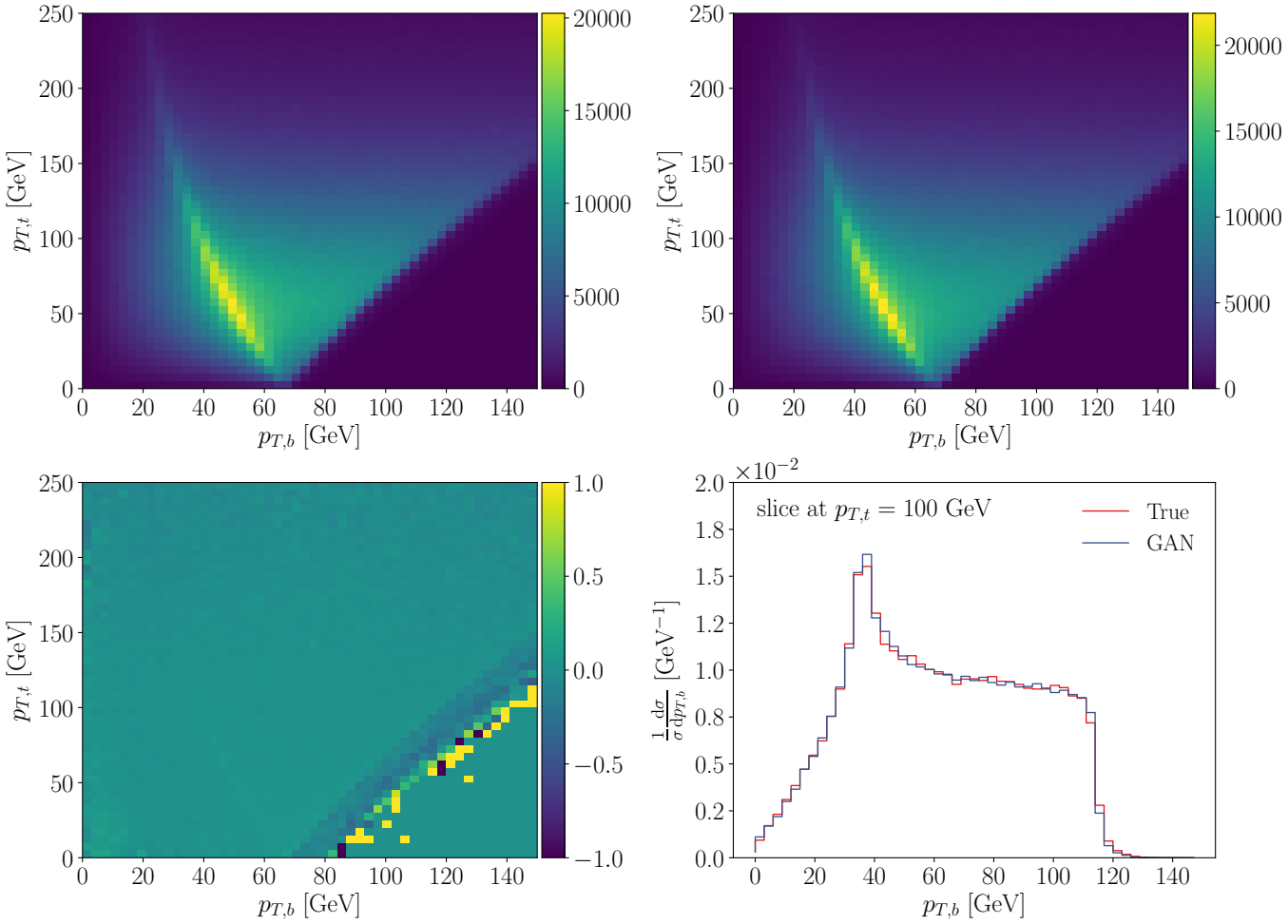

(a) $t = t_0$ (b) $t = t_1$

Solutions:
Additional loss or restricted network parameters

# Amplification



# Correlations



# The subtraction 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

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