Generative Networks for the LHC

Tilman Plehn

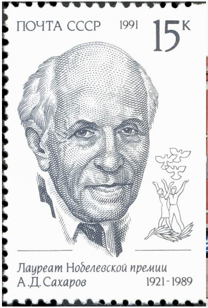
Universität Heidelberg

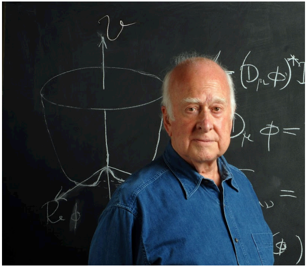
ELLIS Heidelberg, June 2024

Modern LHC physics

Classic motivation

- dark matter?
- matter vs antimatter?
- origin of Higgs boson?





Modern LHC physics

Classic motivation

- dark matter?
- matter vs antimatter?
- origin of Higgs boson?

LHC physics

- fundamental questions
- huge data set
- first-principle, precision simulations
- complete uncertainty control

Modern LHC physics

Classic motivation

- dark matter?
- matter vs antimatter?
- origin of Higgs boson?

LHC physics

- fundamental questions
- huge data set
- first-principle, precision simulations
- complete uncertainty control

Successful past

- measurements of total rates
- analyses inspired by simulation
- model-driven Higgs discovery

Modern LHC physics

Classic motivation

- dark matter?
- matter vs antimatter?
- origin of Higgs boson?

LHC physics

- fundamental questions
- huge data set
- first-principle, precision simulations
- complete uncertainty control

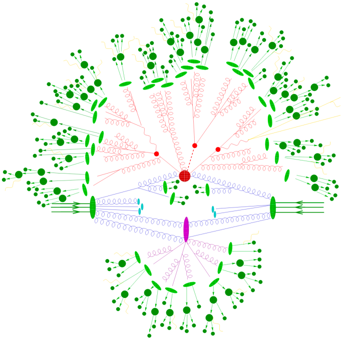
Successful past

- measurements of total rates
- analyses inspired by simulation
- model-driven Higgs discovery

First-principle, precision simulations

- start with Lagrangian
- calculate scattering using QFT
- simulate collisions
- simulate detectors

$\rightarrow$ LHC collisions in virtual worlds



Modern LHC physics

Classic motivation

- dark matter?
- matter vs antimatter?
- origin of Higgs boson?

LHC physics

- fundamental questions
- huge data set
- first-principle, precision simulations
- complete uncertainty control

Successful past

- measurements of total rates
- analyses inspired by simulation
- model-driven Higgs discovery

First-principle, precision simulations

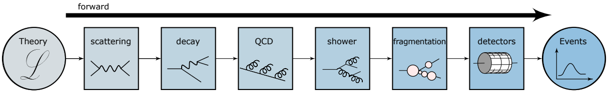
- start with Lagrangian
- calculate scattering using QFT
- simulate collisions
- simulate detectors

$\rightarrow$ LHC collisions in virtual worlds

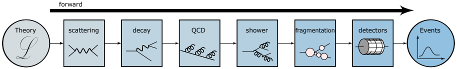
BSM searches

- compare simulations and data
- understand LHC data systematically
- infer underlying theory [SM or BSM]
- publish useable results

$\rightarrow$ Lots of data science...



Role of theory

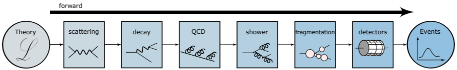


First-principle simulations

- start with Lagrangian
  generate Feynman diagrams
- compute hard scattering amplitudes
  for on-shell, include decays
  add QCD jet radiation [ISR/FSR]
- add parton shower [still QCD]
  push fragmentation towards QCD
- all theory, except for detectors

$\to$ Simulations, not modeling

Role of theory



First-principle simulations

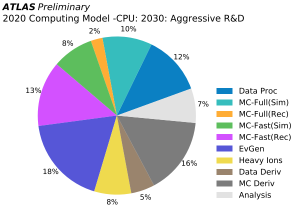
- start with Lagrangian
  generate Feynman diagrams
- compute hard scattering amplitudes
  for on-shell, include decays
  add QCD jet radiation [ISR/FSR]
- add parton shower [still QCD]
  push fragmentation towards QCD
- all theory, except for detectors

→ Simulations, not modeling

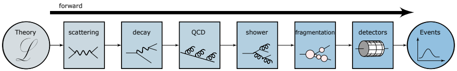
Pythia/Madgraph/Sherpa... for HL-LHC

- factor 25 more expected (= simulated) data
- more complex final states
  higher-orders precision
- parameter coverage for signals
- enable analysis reinterpretation?
  enable global LHC analyses?

→ Theory nightmare



Role of theory



First-principle simulations

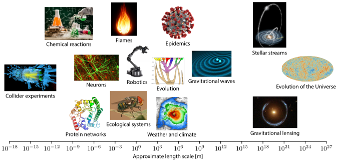
- start with Lagrangian
  generate Feynman diagrams
- compute hard scattering amplitudes
  for on-shell, include decays
  add QCD jet radiation [ISR/FSR]
- add parton shower [still QCD]
  push fragmentation towards QCD
- all theory, except for detectors

→ Simulations, not modeling

LHC-specific explainable AI

- SBI conditional on theory simulations
- understanding LHC data is QFT
- computing speed means precision
- control critical
- uncertainties crucial
- phase space interpretable

→ Lots to talk about...



Generative-network (r)evolution

Same problems, better networks

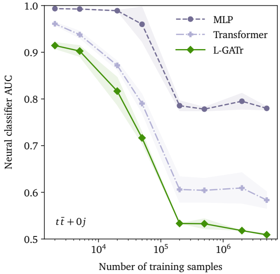
- encode density in target space
  sample from Gaussian into target space
- reproduce training data, statistically independently

Generative-network (r)evolution

Same problems, better networks

- encode density in target space
  sample from Gaussian into target space
- reproduce training data, statistically independently
- VAE [not good]
- GAN [2019]
- normalizing flow/INN [2020/2021]
- diffusion [2023]
- diffusion with attention [2023]
- autoregressive transformer [2023/2024]
- covariant diffusion generator [2024]

$\to$ Bayesianize for uncertainty on estimated density



B-INN as starting point

LHC event generation

- n-particle phase space $\quad n \times 4$ d.o.f. [training on events]
- conceptual playgound for

  MadNIS: phase space sampling [similar to Sherpa]
  inference: unfolding, matrix element method, Bayesian inference
  efficient event shipping
- $Z_{\mu\mu} + \{1, 2, 3\}$ jets [Z-peak, variable jet number, jet-jet topology]

B-INN as starting point

LHC event generation

- n-particle phase space $\quad n \times 4$ d.o.f. [training on events]
- conceptual playgound for
  MadNIS: phase space sampling [similar to Sherpa]
  inference: unfolding, matrix element method, Bayesian inference
  efficient event shipping
- $Z_{\mu\mu} + \{1, 2, 3\}$ jets [Z-peak, variable jet number, jet-jet topology]

INN-generator [2110.13632]

- stable bijective mapping

$$\text{latent } r \sim p_\text{latent} \quad \underset{\leftarrow \overline{G}_\theta(x)}{\overset{G_\theta(r) \rightarrow}{\longleftrightarrow}} \quad \text{phase space } x \sim p_\text{data}$$

- tractable Jacobian

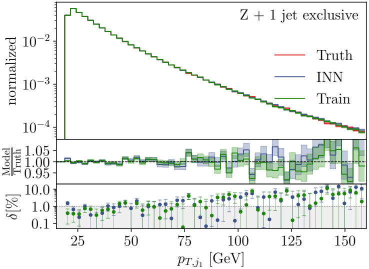
$$dx \; p_\text{model}(x) = dr \; p_\text{latent}(r)$$

$$p_\text{model}(x) = p_\text{latent}\left(\overline{G}_\theta(x)\right) \left| \frac{\partial \overline{G}_\theta(x)}{\partial x} \right|$$

- likelihood loss

$$\mathcal{L}_\text{INN} = -\Big\langle \log p_\text{model}(x) \Big\rangle_{p_\text{data}}$$

$\Rightarrow$ Per-cent precision possible

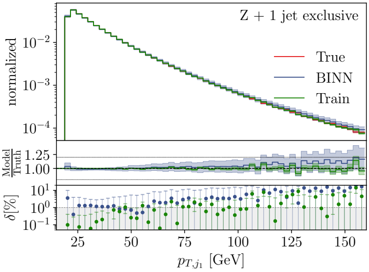


Precision generator with uncertainties

Bayesian network generator

- network with weight distributions [Gal (2016)]
  sample weights [defining error bar]
  working for regression, classification
  frequentist: efficient ensembling

⇒ Training-related error bars



Precision generator with uncertainties

Bayesian network generator

- network with weight distributions [Gal (2016)]
  sample weights [defining error bar]
  working for regression, classification
  frequentist: efficient ensembling

$\Rightarrow$ Training-related error bars

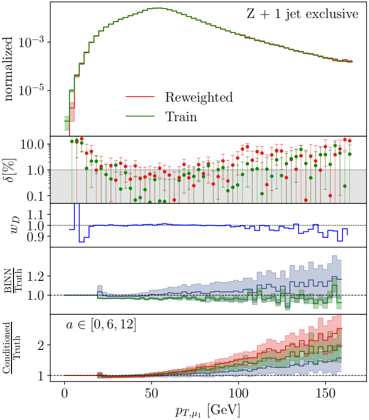
Theory uncertainties

- BNN regression/classification:
  systematics from data augmentation
- systematic uncertainties in tails

$$w = 1 + a \left(\frac{p_{T,j_1} - 15 \ \text{GeV}}{100 \ \text{GeV}} \right)^2$$

- augment training data [$a = 0 \dots 30$]
- train conditionally on a
  error bar from sampling a

$\Rightarrow$ Controlled per-cent precision

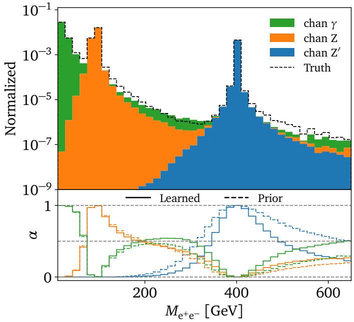


Madgraph, Madevent, MadNIS

INNs as correlated VEGAS

- phase space integration and generation
- VEGAS grids nothing but Jacobians
  INNs are better Jacobians
- learn together with channel weights
- mixted online and buffered training

→ mainstream INN-application

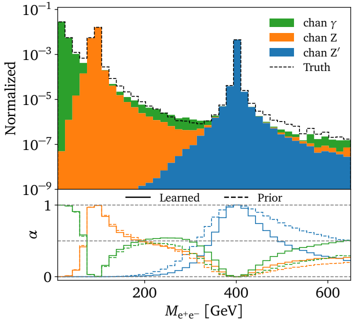


Madgraph, Madevent, MadNIS

INNs as correlated VEGAS

- phase space integration and generation
- VEGAS grids nothing but Jacobians
  INNs are better Jacobians
- learn together with channel weights
- mixted online and buffered training

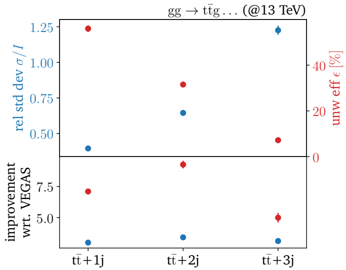
→ mainstream INN-application



Ultra-fast event generation

- applied to proper processes
  up to 1200 Feynman diagrams
  almost 1000 channels
- combination with Madgraph
- goal: 10x improvement

→ Getting implemented right now...



Controlling generative networks

Compare generated with training data

- easy for regression $\Delta = (A_\text{data} - A_\text{model})/A_\text{data}$
- unsupervised density $\to$ supervised density ratio

$$w(x_i) = \frac{D(x_i)}{1 - D(x_i)} = \frac{p_\text{data}(x_i)}{p_\text{model}(x_i)}$$

- classifier more precise and reliable

$\to$ Weight ratio over interpretable phase space

Controlling generative networks

Compare generated with training data

- easy for regression $\Delta = (A_\text{data} - A_\text{model})/A_\text{data}$
- unsupervised density $\to$ supervised density ratio

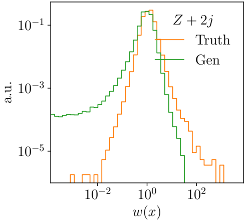
$$w(x_i) = \frac{D(x_i)}{1 - D(x_i)} = \frac{p_\text{data}(x_i)}{p_\text{model}(x_i)}$$

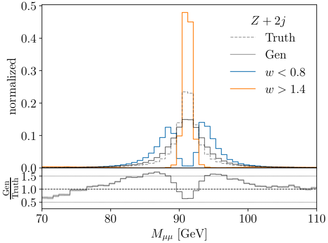
- classifier more precise and reliable

$\to$ Weight ratio over interpretable phase space

Event generators [same for jets, calorimeter showers]

- shapes of w-histogram vs phase space
- shifted weights indicating poor resolution





Controlling generative networks

Compare generated with training data

- easy for regression $\Delta = (A_\text{data} - A_\text{model})/A_\text{data}$
- unsupervised density $\to$ supervised density ratio

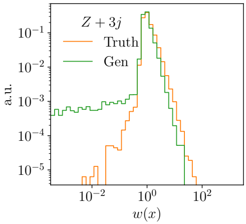
$$w(x_i) = \frac{D(x_i)}{1 - D(x_i)} = \frac{p_\text{data}(x_i)}{p_\text{model}(x_i)}$$

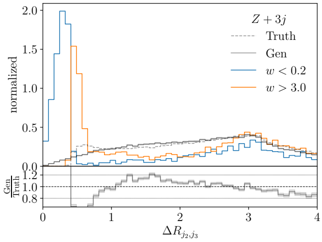
- classifier more precise and reliable

$\to$ Weight ratio over interpretable phase space

Event generators [same for jets, calorimeter showers]

- shapes of w-histogram vs phase space
- shifted weights indicating poor resolution
- small weights indicating missing feature





Conditional flow matching

Diffusion, better than flows

- denoising as generative model

$$p(x,t) \to \begin{cases} p_{\text{data}}(x) & t \to 0 \\ p_{\text{latent}}(x) = \mathcal{N}(x;0,1) & t \to 1 \end{cases}$$

- encode density in velocity [continuity equation]

$$\frac{\partial p(x,t)}{\partial t} + \nabla_x \left[p(x,t) v(x,t) \right] = 0$$

generate from velocity [using ODE solvers]

$$\frac{\partial p(x,t)}{\partial t} + \nabla_x \left[p(x,t) v(x,t) \right] = 0 \qquad \Leftrightarrow \qquad \frac{dx(t)}{dt} = v(x(t),t)$$

Conditional flow matching

Diffusion, better than flows

- denoising as generative model

$$p(x,t) \rightarrow \begin{cases} p_{\text{data}}(x) & t \rightarrow 0 \\ p_{\text{latent}}(x) = \mathcal{N}(x;0,1) & t \rightarrow 1 \end{cases}$$

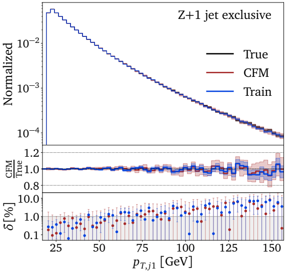
- encode density in velocity [continuity equation]

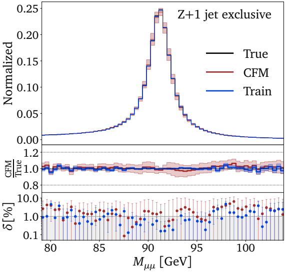
$$\frac{\partial p(x,t)}{\partial t} + \nabla_x \left[p(x,t) v(x,t) \right] = 0$$

generate from velocity [using ODE solvers]

$$\frac{\partial p(x,t)}{\partial t} + \nabla_x \left[p(x,t) v(x,t) \right] = 0 \qquad \Leftrightarrow \qquad \frac{dx(t)}{dt} = v(x(t),t)$$

$\rightarrow$ Sub-percent precision





Direct diffusion

Structural advantage of CFM model

- sample from one distribution into another
  avoid learning some features
- example: off-shell top decays from on-shell top decays

$$x \sim p_\text{on}(x) \quad \longleftrightarrow \quad x \sim p_\text{model}(x) \sim p_\text{off}(x)$$

- standard CFM with boundary conditions

$$p(x,t) \to \begin{cases} p_\text{off}(x) & t \to 0 \\ p_\text{on}(x) & t \to 1 \end{cases}$$

Direct diffusion

Structural advantage of CFM model

- sample from one distribution into another
  avoid learning some features
- example: off-shell top decays from on-shell top decays

$$x \sim p_{\text{on}}(x) \quad \longleftrightarrow \quad x \sim p_{\text{model}}(x) \sim p_{\text{off}}(x)$$

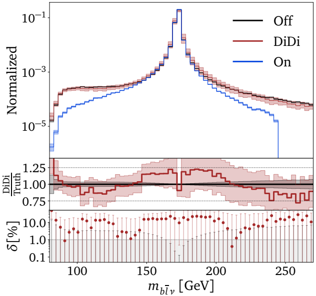
- standard CFM with boundary conditions

$$p(x,t) \to \begin{cases} p_{\text{off}}(x) & t \to 0 \\ p_{\text{on}}(x) & t \to 1 \end{cases}$$

Precision benefits

- data-driven optimal transport
- high-precision features
- minimal failure modes

$\to$ More applications?



(ML-)Unfolding

Number of analyses

- optimal forward inference:
  full signal and background simulations
  high-dimensional, unbinned SBI
- CPU-limitation for many signals

→ Unfold detectors once

Optimal analyses

- theory limiting many LHC analyses
  make best use of continuous progress
- allow for analyses to be updated

→ Unfold detectors/soft QCD and save data

Public LHC data

- common lore:
  LHC data too complicated for amateurs
  no way to even try to publish LHC data
- in truth:
  hard scattering and decay simulations easy
  BSM physics not in hadronization and detector

→ Unfold to hard scattering

Unfolding without and with ML

Basic idea

- four phase space distributions

$$\begin{tikzcd}
p_\text{sim}(x_\text{part}) \arrow[rr, leftrightarrow, "\text{unfolding inference}"] \arrow[d, "p(x_\text{reco}|x_\text{part})"'] & & p_\text{unfold}(x_\text{part}) \\
p_\text{sim}(x_\text{reco}) \arrow[rr, leftrightarrow, "\text{forward inference}"] & & p_\text{data}(x_\text{reco}) \arrow[u, "p(x_\text{part}|x_\text{reco})"']
\end{tikzcd}$$

- two conditional probabilities

$$p(x_\text{part}|x_\text{reco}) = p(x_\text{reco}|x_\text{part})\,\frac{p_\text{sim}(x_\text{part})}{p_\text{sim}(x_\text{reco})}$$

Unfolding without and with ML

Basic idea

- four phase space distributions

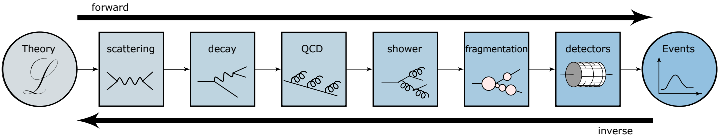
$$
\begin{tikzcd}
p_\text{sim}(x_\text{part}) \arrow[rr, leftrightarrow, "\text{unfolding inference}"] \arrow[d, "p(x_\text{reco}|x_\text{part})"'] & & p_\text{unfold}(x_\text{part}) \\
p_\text{sim}(x_\text{reco}) \arrow[rr, leftrightarrow, "\text{forward inference}"] & & p_\text{data}(x_\text{reco}) \arrow[u, "p(x_\text{part}|x_\text{reco})"']
\end{tikzcd}
$$

- two conditional probabilities

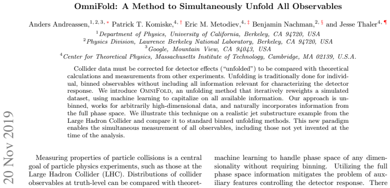
$$
p(x_\text{part}|x_\text{reco}) = p(x_\text{reco}|x_\text{part})\,\frac{p_\text{sim}(x_\text{part})}{p_\text{sim}(x_\text{reco})}
$$

LHC simulations

- paired
- stochastic, usually single-mode [nothing LHC is deterministic]
- following energy scale/resolution
- starting from fundamental parameters



Unfolding by reweighting


OmniFold: A Method to Simultaneously Unfold All Observables

Anders Andreassen,[1,2,3,*] Patrick T. Komiske,[4,†] Eric M. Metodiev,[4,‡] Benjamin Nachman,[2,§] and Jesse Thaler[4,¶]

[1] *Department of Physics, University of California, Berkeley, CA 94720, USA*
[2] *Physics Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA*
[3] *Google, Mountain View, CA 94043, USA*
[4] *Center for Theoretical Physics, Massachusetts Institute of Technology, Cambridge, MA 02139, U.S.A.*

Collider data must be corrected for detector effects ("unfolded") to be compared with theoretical calculations and measurements from other experiments. Unfolding is traditionally done for individual, binned observables without including all information relevant for characterizing the detector response. We introduce OmniFold, an unfolding method that iteratively reweights a simulated dataset, using machine learning to capitalize on all available information. Our approach is unbinned, works for arbitrarily high-dimensional data, and naturally incorporates information from the full phase space. We illustrate this technique on a realistic jet substructure example from the Large Hadron Collider and compare it to standard binned unfolding methods. This new paradigm enables the simultaneous measurement of all observables, including those not yet invented at the time of the analysis.

20 Nov 2019

Measuring properties of particle collisions is a central goal of particle physics experiments, such as those at the Large Hadron Collider (LHC). Distributions of collider observables at truth-level can be compared with theoret- machine learning to handle phase space of any dimensionality without requiring binning. Utilizing the full phase space information mitigates the problem of auxiliary features controlling the detector response. There

OmniFold

- use paired events $(x_\text{part}, x_\text{reco})$
  learn $p_\text{sim}(x_\text{reco}) \leftrightarrow p_\text{data}(x_\text{reco})$
  reweight $p_\text{sim}(x_\text{part}) \to p_\text{unfold}(x_\text{part})$

$$
\begin{tikzcd}
p_\text{sim}(x_\text{part}) \arrow[rr, "\text{classifier weights (3)}"] \arrow[d, leftrightarrow, "\text{pull (2)/push weights(4)}"'] & & p_\text{unfold}(x_\text{part}) \\
p_\text{sim}(x_\text{reco}) \arrow[rr, leftrightarrow, "\text{classifier weights (1)}"] & & p_\text{data}(x_\text{reco})
\end{tikzcd}
$$

- unbinned classifier weight [Neyman-Pearson lemma, CWoLa]

$$
w_D(x_i) = \frac{D(x_i)}{1 - D(x_i)} \to \frac{p_1(x_i)}{p_2(x_i)}
$$

- high-dimensional classification, like jet tagging

$\to$ Driven by (now) established ML-classification

Unfolding by generation

SciPost Physics Submission

How to GAN away Detector Effects

Marco Bellagente[1], Anja Butter[1], Gregor Kasieczka[2], Tilman Plehn[1], and Ramon Winterhalder[1]

1 Institut für Theoretische Physik, Universität Heidelberg, Germany
2 Institut für Experimentalphysik, Universität Hamburg, Germany
bellagente@thphys.uni-heidelberg.de

April 23, 2022

1 Dec 2019

Abstract

Sampling conditional probability

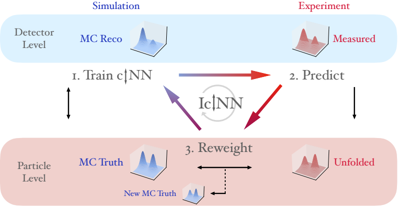
- just like forward ML-generation
- learn inverse conditional probability also from paired events $(x_{\text{part}}, x_{\text{reco}})$

$$
\begin{tikzcd}
p_{\text{sim}}(x_{\text{part}}) \arrow[d, leftrightarrow, "\text{paired data}"'] & & p_{\text{unfold}}(x_{\text{part}}) \\
p_{\text{sim}}(x_{\text{reco}}) \arrow[rr, leftrightarrow, "\text{correspondence}"] & & p_{\text{data}}(x_{\text{reco}}) \arrow[u, "p_{\text{model}}(x_{\text{part}}|x_{\text{reco}})"']
\end{tikzcd}
$$

Two improvements needed [taking some time]

1. likelihood loss to generate posterior $\rightarrow$ cINN, CFM
2. remove training prior $\rightarrow$ IcINN [Backes, Butter, Dunford, Malaescu]

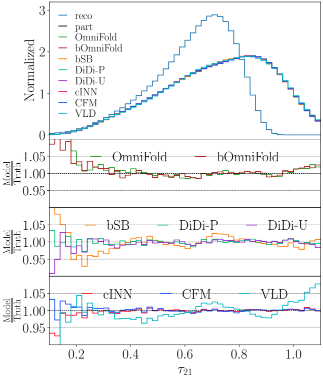
$\rightarrow$ Driven by generative networks



Heidelberg-Berkeley-Irvine

Detector unfolding: jets

- 1. event reweighting (b)Omnifold
  2. distribution mapping DiDi,(b)SB
  3. conditional generation cINN, CFM, VLD



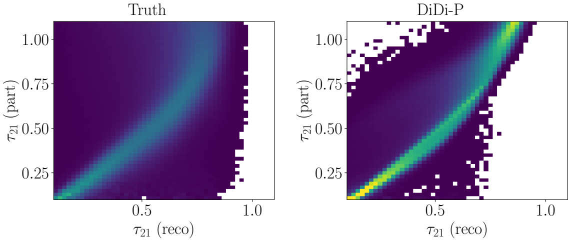
Heidelberg-Berkeley-Irvine

Detector unfolding: jets

- 1. event reweighting — (b)Omnifold
  2. distribution mapping — DiDi,(b)SB
  3. conditional generation — cINN, CFM, VLD

Event migration: jets

- trained on paired events
  event migration known
- DiDi paried: too sharp



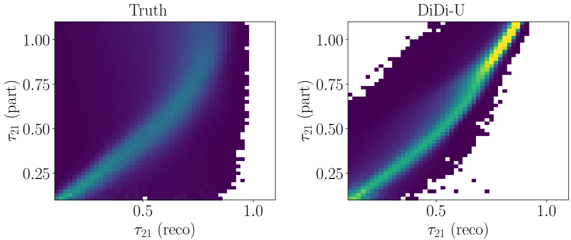
Heidelberg-Berkeley-Irvine

Detector unfolding: jets

- 1. event reweighting (b)Omnifold
  2. distribution mapping DiDi,(b)SB
  3. conditional generation cINN, CFM, VLD

Event migration: jets

- trained on paired events
  event migration known
- DiDi paried: too sharp
  DiDi unpaired: more noisy



Heidelberg-Berkeley-Irvine

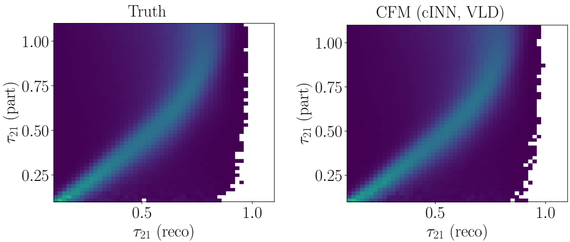
Detector unfolding: jets

- 1. event reweighting (b)Omnifold
  2. distribution mapping DiDi,(b)SB
  3. conditional generation cINN, CFM, VLD

Event migration: jets

- trained on paired events
  event migration known
- DiDi paried: too sharp
  DiDi unpaired: more noisy

$\to$ generative: correct conditional posterior



Heidelberg-Berkeley-Irvine

Detector unfolding: jets

- 1. event reweighting (b)Omnifold
  2. distribution mapping DiDi,(b)SB
  3. conditional generation cINN, CFM, VLD

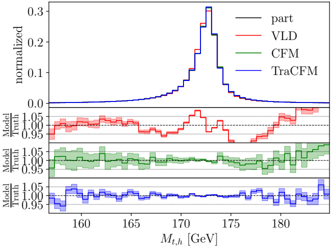
Event migration: jets

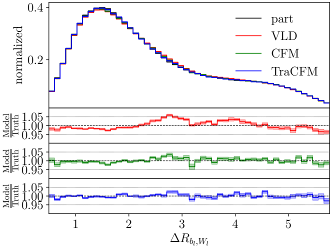
- trained on paired events
  event migration known
- DiDi paried: too sharp
  DiDi unpaired: more noisy

$\rightarrow$ generative: correct conditional posterior

Unfolding to partons: $t_h \bar{t}_\ell$

- phase space parametrization key
- transformer for combinatorics





Heidelberg-Berkeley-Irvine

Detector unfolding: jets

- 1. event reweighting (b)Omnifold
  2. distribution mapping DiDi,(b)SB
  3. conditional generation cINN, CFM, VLD

Event migration: jets

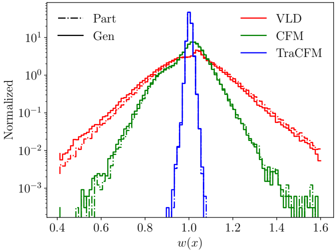
- trained on paired events
  event migration known
- DiDi paried: too sharp
  DiDi unpaired: more noisy

→ generative: correct conditional posterior

Unfolding to partons: $t_h \bar{t}_\ell$

- phase space parametrization key
- transformer for combinatorics
- trained classifier test [Das, Favaro, Heimel, Krause, TP, Shih]

→ Consistently high precision



Unfolding top decays

Enjoying a technical challenge [Favaro, Kogler, Paasch, Palacios Schweitzer, TP, Schwarz]

- first measure m_t in unfolded boosted decays
  then unfold kinematics of 3 subjets
- model dependence m_s vs m_d

$$\begin{tikzcd}
p_{\text{sim}}(x_{\text{part}}|m_s) \arrow[d, "p(x_{\text{reco}}|x_{\text{part}})"'] & p_{\text{unfold}}(x_{\text{part}}|m_s, m_d) \\
p_{\text{sim}}(x_{\text{reco}}|m_s) \arrow[r, leftrightarrow, "\text{correspondence}"] & p_{\text{data}}(x_{\text{reco}}|m_d) \arrow[u, "p_{\text{model}}(x_{\text{part}}|x_{\text{reco}}, m_s)"']
\end{tikzcd}$$

Unfolding top decays

Enjoying a technical challenge [Favaro, Kogler, Paasch, Palacios Schweitzer, TP, Schwarz]

- first measure m_t in unfolded boosted decays
  then unfold kinematics of 3 subjets
- complete training bias $m_d \to m_s$ [too bad to reweight]

$$\begin{tikzcd}
p_\text{sim}(x_\text{part}|m_s) \arrow[d, "p(x_\text{reco}|x_\text{part})"'] & & p_\text{unfold}(x_\text{part}|m_s, \cancel{m_d}) \\
p_\text{sim}(x_\text{reco}|m_s) \arrow[rr, leftrightarrow, "\text{correspondence}"] & & p_\text{data}(x_\text{reco}|m_d) \arrow[u, "p_\text{model}(x_\text{part}|x_\text{reco}, m_s)"']
\end{tikzcd}$$

1. weaken bias by training on range of m_s-values
2. strengthen data by including batch-wise $m_d \sim M_{jjj} \in x_\text{reco}$

Unfolding top decays

Enjoying a technical challenge [Favaro, Kogler, Paasch, Palacios Schweitzer, TP, Schwarz]

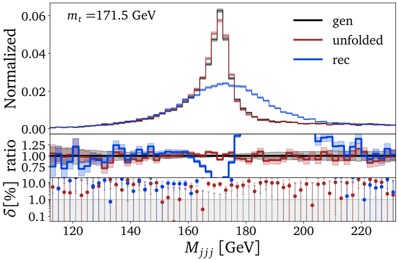
- first measure m_t in unfolded boosted decays
  then unfold kinematics of 3 subjets
- complete training bias $m_d \to m_s$ [too bad to reweight]

$$\begin{tikzcd}[column sep=huge, row sep=huge]
p_\text{sim}(x_\text{part}|m_s) \arrow[d, "p(x_\text{reco}|x_\text{part})"'] & p_\text{unfold}(x_\text{part}|m_s, \cancel{m_d}) \\
p_\text{sim}(x_\text{reco}|m_s) \arrow[r, leftrightarrow, "\text{correspondence}"] & p_\text{data}(x_\text{reco}|m_d) \arrow[u, "p_\text{model}(x_\text{part}|x_\text{reco}, m_s)"']
\end{tikzcd}$$

1. weaken bias by training on range of m_s-values
2. strengthen data by including batch-wise $m_d \sim M_{jjj} \in x_\text{reco}$

Reduced phase space [TraCFM]

- dedicated parametrization
- 4D for calibration and top mass



Unfolding top decays

Enjoying a technical challenge [Favaro, Kogler, Paasch, Palacios Schweitzer, TP, Schwarz]

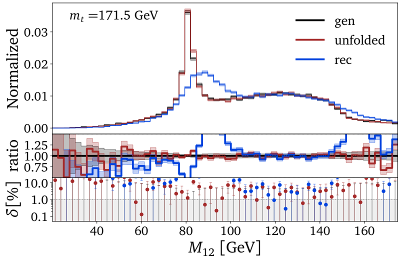
- first measure m_t in unfolded boosted decays
  then unfold kinematics of 3 subjets
- complete training bias $m_d \to m_s$ [too bad to reweight]

$$\begin{tikzcd}
p_\text{sim}(x_\text{part}|m_s) \arrow[d, "p(x_\text{reco}|x_\text{part})"'] & & p_\text{unfold}(x_\text{part}|m_s, \cancel{m_d}) \\
p_\text{sim}(x_\text{reco}|m_s) \arrow[rr, leftrightarrow, "\text{correspondence}"] & & p_\text{data}(x_\text{reco}|m_d) \arrow[u, "p_\text{model}(x_\text{part}|x_\text{reco}, m_s)"']
\end{tikzcd}$$

1. weaken bias by training on range of m_s-values
2. strengthen data by including batch-wise $m_d \sim M_{jjj} \in x_\text{reco}$

Reduced phase space [TraCFM]

- dedicated parametrization
- 4D for calibration and top mass



Unfolding top decays

Enjoying a technical challenge [Favaro, Kogler, Paasch, Palacios Schweitzer, TP, Schwarz]

- first measure m_t in unfolded boosted decays
  then unfold kinematics of 3 subjets
- complete training bias $m_d \to m_s$ [too bad to reweight]

$$\begin{tikzcd}
p_\text{sim}(x_\text{part}|m_s) \arrow[d, "p(x_\text{reco}|x_\text{part})"'] & & p_\text{unfold}(x_\text{part}|m_s, \cancel{m_d}) \\
p_\text{sim}(x_\text{reco}|m_s) \arrow[rr, leftrightarrow, "\text{correspondence}"] & & p_\text{data}(x_\text{reco}|m_d) \arrow[u, "p_\text{model}(x_\text{part}|x_\text{reco}, m_s)"']
\end{tikzcd}$$

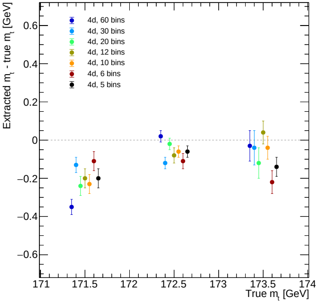
1 weaken bias by training on range of m_s-values

2 strengthen data by including batch-wise $m_d \sim M_{jjj} \in x_\text{reco}$

Reduced phase space [TraCFM]

- dedicated parametrization
- 4D for calibration and top mass

$\to$ unbiased top mass



Unfolding top decays

Enjoying a technical challenge [Favaro, Kogler, Paasch, Palacios Schweitzer, TP, Schwarz]

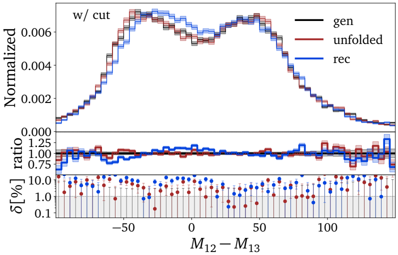
- first measure m_t in unfolded boosted decays
  then unfold kinematics of 3 subjets
- complete training bias $m_d \to m_s$ [too bad to reweight]

$$
\begin{tikzcd}
p_\text{sim}(x_\text{part} | m_s) \arrow[d, "p(x_\text{reco} | x_\text{part})"'] & & p_\text{unfold}(x_\text{part} | m_s, \cancel{m_d}) \\
p_\text{sim}(x_\text{reco} | m_s) \arrow[rr, leftrightarrow, "\text{correspondence}"] & & p_\text{data}(x_\text{reco} | m_d) \arrow[u, "p_\text{model}(x_\text{part} | x_\text{reco}, m_s)"']
\end{tikzcd}
$$

1. weaken bias by training on range of m_s-values
2. strengthen data by including batch-wise $m_d \sim M_{jjj} \in x_\text{reco}$

Full 12D unfolding

- using measured top mass
- azimuthal angle derived
- correlations not special



Unfolding top decays

Enjoying a technical challenge [Favaro, Kogler, Paasch, Palacios Schweitzer, TP, Schwarz]

- first measure m_t in unfolded boosted decays
  then unfold kinematics of 3 subjets
- complete training bias $m_d \to m_s$ [too bad to reweight]

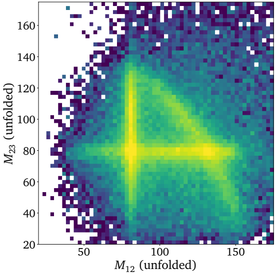
$$\begin{tikzcd}
p_\text{sim}(x_\text{part}|m_s) \arrow[d, "p(x_\text{reco}|x_\text{part})"'] & & p_\text{unfold}(x_\text{part}|m_s, \cancel{m_d}) \\
p_\text{sim}(x_\text{reco}|m_s) \arrow[rr, leftrightarrow, "\text{correspondence}"] & & p_\text{data}(x_\text{reco}|m_d) \arrow[u, "p_\text{model}(x_\text{part}|x_\text{reco}, m_s)"']
\end{tikzcd}$$

1. weaken bias by training on range of m_s-values
2. strengthen data by including batch-wise $m_d \sim M_{jjj} \in x_\text{reco}$

Full 12D unfolding

- using measured top mass
- azimuthal angle derived
- correlations not special

$\to$ CMS data next



ML for LHC Theory

ML-applications

- just another numerical tool for a numerical field
- driven by money from data science and medical research
- goals are...

  ...improve established tasks
  ...develop new tools for established tasks
  ...transform through new ideas
- xAI through...

  ...precision control
  ...uncertainties
  ...symmetries
  ...formulas

$\rightarrow$ Lots of fun with hard LHC problems

Modern Machine Learning for LHC Physicists

Tilman Plehn[a,*], Anja Butter[a,b], Barry Dillon[a], Claudius Krause[a,c], and Ramon Winterhalder[d]

[a] Institut für Theoretische Physik, Universität Heidelberg, Germany
[b] LPNHE, Sorbonne Université, Université Paris Cité, CNRS/IN2P3, Paris, France
[c] NHETC, Dept. of Physics and Astronomy, Rutgers University, Piscataway, USA
[d] CP3, Université Catholique de Louvain, Louvain-la-Neuve, Belgium

July 21, 2023

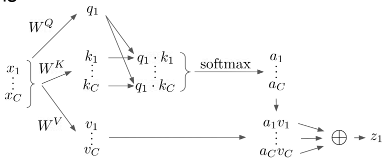
Abstract

Modern machine learning is transforming particle physics, faster than we can follow, and bullying its way into our numerical tool box. For young researchers it is crucial to stay on top of this development, which means applying cutting-edge methods and tools to the full range of LHC physics problems. These lecture notes are meant to lead students with basic knowledge of particle physics and significant enthusiasm for machine learning to relevant applications as fast as possible. They start with an LHC-specific motivation and a non-standard introduction to neural networks and then cover classification, unsupervised classification, generative networks, and inverse problems. Two themes defining much of the discussion are well-defined loss functions reflecting the problem at hand and uncertainty-aware networks. As part of the applications, the notes include some aspects of theoretical LHC physics. All examples are chosen from particle physics publications of the last few years. Given that these notes will be outdated already at the time of submission, the week of ML4Jets 2022, they will be updated frequently.

JetGPT

Correlations through self-attention [2305.10475]

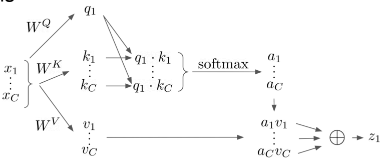
- think of data as bins in phase-space directions
  self-attention: encode relation between bins
  input x, learn relation $x_i \leftrightarrow x_j$
- latent query representation $q = W^Q x$
  latent key representation $k = W^K x$
  define correlation as $A_{ij} = q_i \cdot k_j$
- latent value representation $v = W^V x$
  output $z = A\, v$



JetGPT

Correlations through self-attention [2305.10475]

- think of data as bins in phase-space directions
  self-attention: encode relation between bins
  input x, learn relation $x_i \leftrightarrow x_j$
- latent query representation $q = W^Q x$
  latent key representation $k = W^K x$
  define correlation as $A_{ij} = q_i \cdot k_j$
- latent value representation $v = W^V x$
  output $z = A\, v$



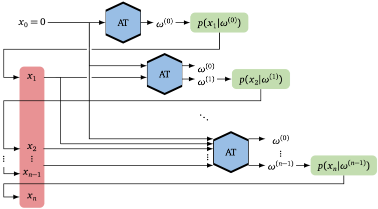
Autoregressive generator

- factorized density

$$p_\text{model}(x|\theta) = \prod_i p(x_i|x_1, \dots, x_{i-1})$$

- bins $\rightarrow$ Gaussian mixture model
- autoregressive $A_{ij} = 0$ for $j > i$

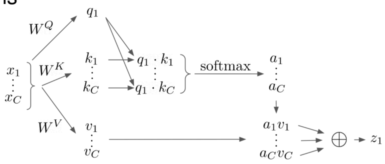
$\rightarrow$ Bayesian version for uncertainties



JetGPT

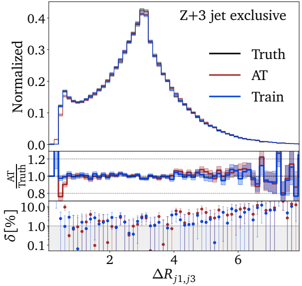
Correlations through self-attention [2305.10475]

- think of data as bins in phase-space directions
  self-attention: encode relation between bins
  input x, learn relation $x_i \leftrightarrow x_j$
- latent query representation $q = W^Q x$
  latent key representation $k = W^K x$
  define correlation as $A_{ij} = q_i \cdot k_j$
- latent value representation $v = W^V x$
  output $z = A\, v$



Bayesian JetGPT

- sometimes you win...



ML-Theory

Tilman Plehn

LHC physics

Generation

Uncertainties

MadNIS

Control

Expressivity

Inversion

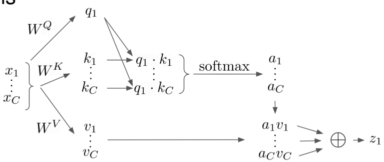
Ideas

News

JetGPT

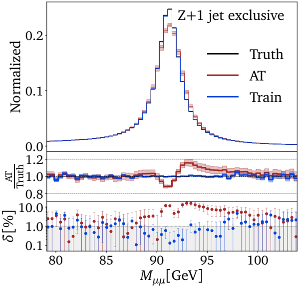
Correlations through self-attention [2305.10475]

- think of data as bins in phase-space directions
  self-attention: encode relation between bins
  input x, learn relation $x_i \leftrightarrow x_j$
- latent query representation $q = W^Q x$
  latent key representation $k = W^K x$
  define correlation as $A_{ij} = q_i \cdot k_j$
- latent value representation $v = W^V x$
  output $z = A\, v$



Bayesian JetGPT

- sometimes you win...
  ...and sometimes there is work to do...



ML-Theory

Tilman Plehn

LHC physics

Generation

Uncertainties

MadNIS

Control

Expressivity

Inversion

Ideas

News

High-dimensional and unbinned

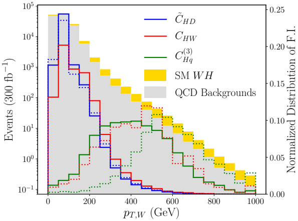
Simple process $pp \to W_\ell H_{bb}$ [Brehmer, Dawson, Homiller, Kling, TP, long time ago]

- example operators [wf vs vertex structure vs 4-point]

$$\widetilde{\mathcal{O}}_{HD} = (\phi^\dagger \phi)\Box(\phi^\dagger \phi) - \frac{1}{4}(\phi^\dagger D^\mu \phi)^* (\phi^\dagger D_\mu \phi)$$

$$\mathcal{O}_{HW} = \phi^\dagger \phi W^a_{\mu\nu} W^{\mu\nu a}$$

$$\mathcal{O}^{(3)}_{Hq} = (\phi^\dagger i\overleftrightarrow{D}^a_\mu \phi)(\overline{Q}_L \sigma^a \gamma^\mu Q_L)$$



High-dimensional and unbinned

Simple process $pp \to W_\ell H_{bb}$ [Brehmer, Dawson, Homiller, Kling, TP, long time ago]

- example operators [wf vs vertex structure vs 4-point]

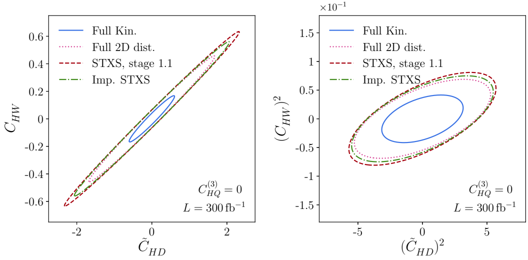
$$\widetilde{\mathcal{O}}_{HD} = (\phi^\dagger \phi)\Box(\phi^\dagger \phi) - \frac{1}{4}(\phi^\dagger D^\mu \phi)^*(\phi^\dagger D_\mu \phi)$$

$$\mathcal{O}_{HW} = \phi^\dagger \phi W^a_{\mu\nu} W^{\mu\nu a}$$

$$\mathcal{O}^{(3)}_{Hq} = (\phi^\dagger i\overleftrightarrow{D}^a_\mu \phi)(\overline{Q}_L \sigma^a \gamma^\mu Q_L)$$

Full kinematics vs $p_{T,W} - m_{T,\text{tot}}$

- bulk operators



High-dimensional and unbinned

Simple process $pp \to W_\ell H_{bb}$ [Brehmer, Dawson, Homiller, Kling, TP, long time ago]

- example operators [wf vs vertex structure vs 4-point]

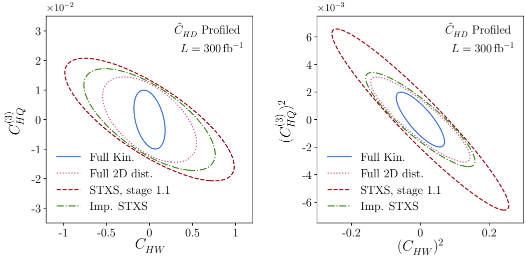
$$\widetilde{\mathcal{O}}_{HD} = (\phi^\dagger \phi)\Box(\phi^\dagger \phi) - \frac{1}{4}(\phi^\dagger D^\mu \phi)^*(\phi^\dagger D_\mu \phi)$$

$$\mathcal{O}_{HW} = \phi^\dagger \phi W^a_{\mu\nu} W^{\mu\nu a}$$

$$\mathcal{O}^{(3)}_{Hq} = (\phi^\dagger i\overleftrightarrow{D}^a_\mu \phi)(\overline{Q}_L \sigma^a \gamma^\mu Q_L)$$

Full kinematics vs $p_{T,W} - m_{T,\text{tot}}$

- bulk operators
- tail operator



$\to$ Full, unbinned kinematics the key [top groups doing better]

Inverting to QCD

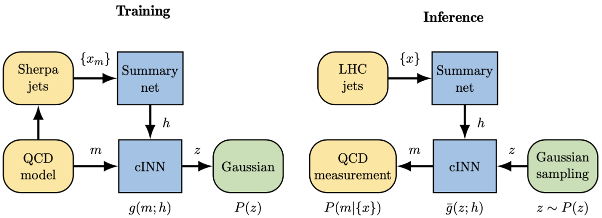
cINN for inference [Bieringer, Butter, Heimel, Höche, Köthe, TP, Radev]

- condition jets with QCD parameters
  train model parameters $\to$ Gaussian latent space
  test Gaussian sampling $\to$ parameter measurement
- beyond C_A vs C_F

$$P_{qq} = C_F \left[D_{qq} \frac{2z(1-y)}{1-z(1-y)} + F_{qq}(1-z) + C_{qq} yz(1-z) \right]$$

$$P_{gg} = 2C_A \left[D_{gg} \left(\frac{z(1-y)}{1-z(1-y)} + \frac{(1-z)(1-y)}{1-(1-z)(1-y)} \right) + F_{gg} z(1-z) + C_{gg} yz(1-z) \right]$$

$$P_{gq} = T_R \left[F_{qq} \left(z^2 + (1-z)^2 \right) + C_{gq} yz(1-z) \right]$$



Inverting to QCD

cINN for inference [Bieringer, Butter, Heimel, Höche, Köthe, TP, Radev]

- condition jets with QCD parameters
  train model parameters $\to$ Gaussian latent space
  test Gaussian sampling $\to$ parameter measurement
- beyond C_A vs C_F

$$P_{qq} = C_F \left[D_{qq} \frac{2z(1-y)}{1-z(1-y)} + F_{qq}(1-z) + C_{qq} yz(1-z) \right]$$

$$P_{gg} = 2C_A \left[D_{gg} \left(\frac{z(1-y)}{1-z(1-y)} + \frac{(1-z)(1-y)}{1-(1-z)(1-y)} \right) + F_{gg} z(1-z) + C_{gg} yz(1-z) \right]$$

$$P_{gq} = T_R \left[F_{qq} \left(z^2 + (1-z)^2 \right) + C_{gq} yz(1-z) \right]$$

- idealized shower [Sherpa]
- More ML-opportunities...