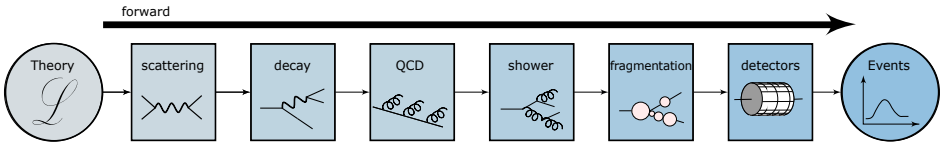
Controlled Precision Generators

Tilman Plehn

Universität Heidelberg

Bützmich Workshop, Hamburg, February 2024

LHC theory predictions

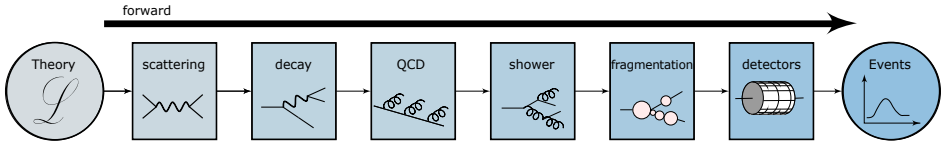


First-principle simulations

- 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

$\rightarrow$ Simulations, not modeling!

LHC theory predictions



First-principle simulations

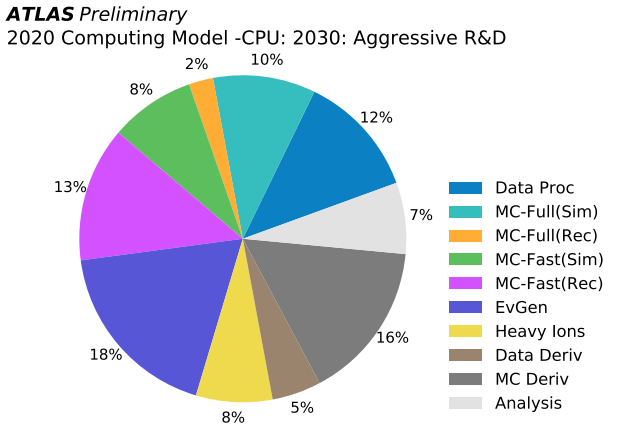
- start with Lagrangian
  generate Feynman diagrams
- compute hard scattering amplitudes
  for on-shell, include decays
  add QCD jet radiation [ISR/FSR]
- add parton shower [still QCD]
  push fragmentation towards QCD
- all theory, except for detectors

→ 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



Generative-network revolution

Generative networks

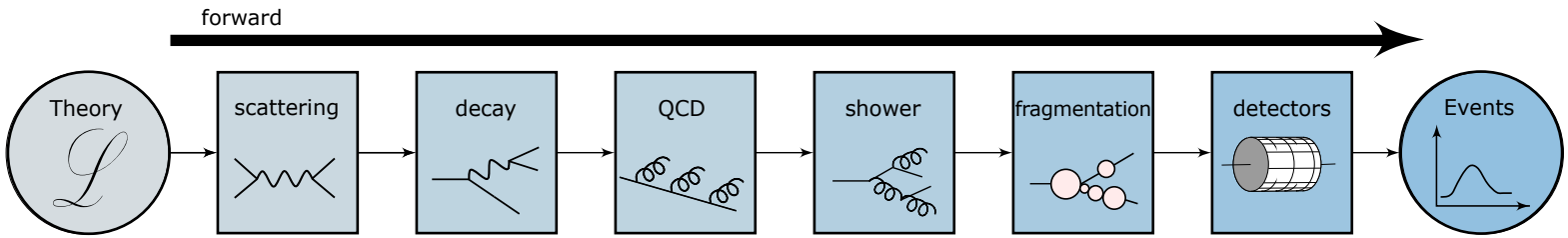
- encode density in target space
  sample from Gaussian into target space
- reproduce training data, statistically independently

Generative-network revolution

Generative networks

- encode density in target space
  sample from Gaussian into target space
- reproduce training data, statistically independently
- Variational Autoencoder
  $\to$ low-dimensional physics, high-dimensional representation
- Generative Adversarial Network
  $\to$ generator playing against discriminator
- Normalizing Flow/INN
  $\to$ stable (bijective) mapping
- Diffusion Model
  $\to$ discrete (okay) or continuous (great)
- Generative Transformer
  $\to$ learning correlations successively

$\to$ Bayesian NN uncertainty on estimated density



2012 B-INN as starting point

LHC event generation

- n-particle phase space $n \times 4$ 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]

2012 B-INN as starting point

LHC event generation

- n-particle phase space $n \times 4$ 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

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

- tractable Jacobian

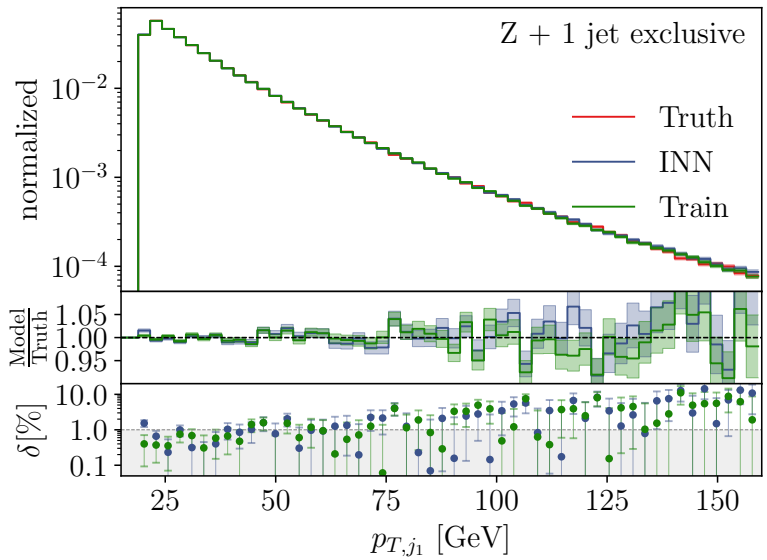
$$dx\; p_\text{model}(x) = dr\; p_\text{latent}(r)$$

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

- likelihood loss

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

$\Rightarrow$ Per-cent precision possible



Controlled precision generator

Best of GAN: discriminator

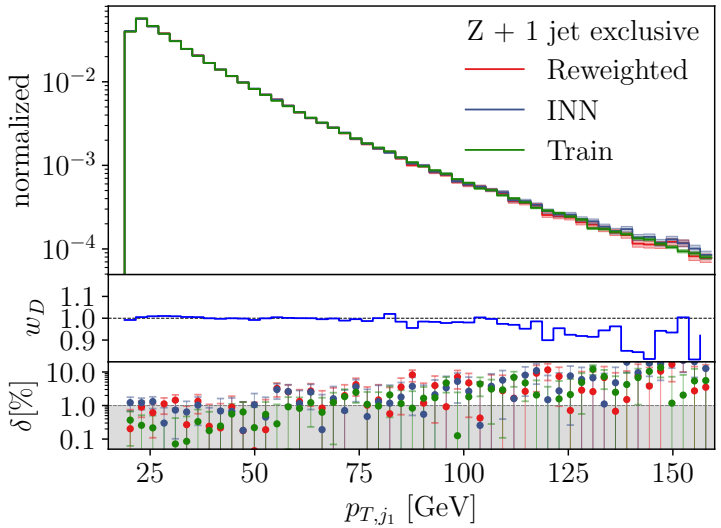
- $D = 0$ (generator) vs $D = 1$ (training)
- Neyman Pearson-optimal discriminator

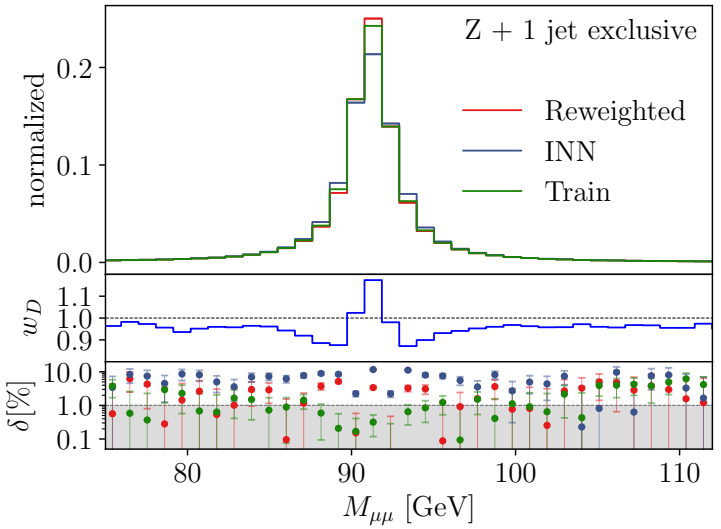
$$D(x) \to \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_{\text{model}}(x)} \to \frac{1}{2}$$

- learned event weight

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

$\Rightarrow$ Dual purpose: control and reweight





Controlled precision generator

Best of GAN: discriminator

- $D = 0$ (generator) vs $D = 1$ (training)
- Neyman Pearson-optimal discriminator

$$D(x) \to \frac{p_\text{data}(x)}{p_\text{data}(x) + p_\text{model}(x)} \to \frac{1}{2}$$

- learned event weight

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

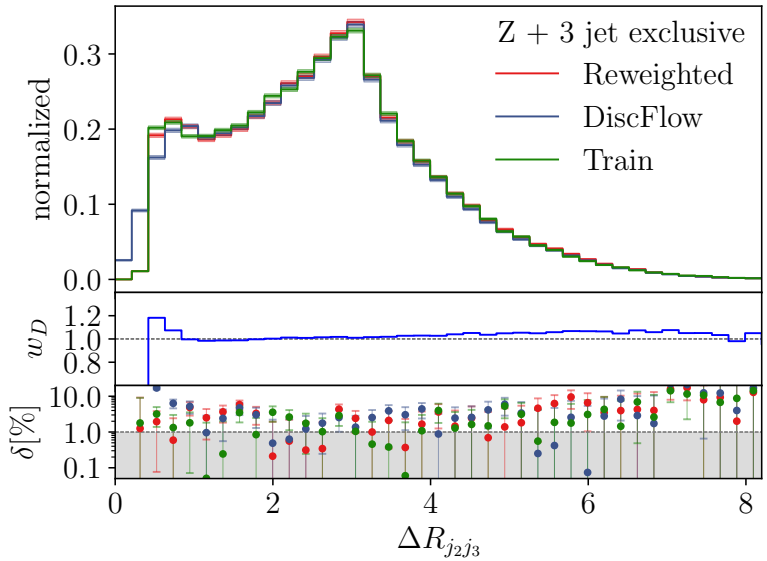
$\Rightarrow$ Dual purpose: control and reweight

Joint training [GAN inspiration]

- GAN-like training unstable [Nash equilibrium??]
- coupling through weights

$$\mathcal{L} = - \int dx \, \frac{p_\text{data}^{\alpha+1}(x)}{p_\text{model}^{\alpha}(x)} \, \log \frac{p_\text{model}(x)}{p_\text{data}(x)}$$

$\Rightarrow$ Unweighted, controlled events

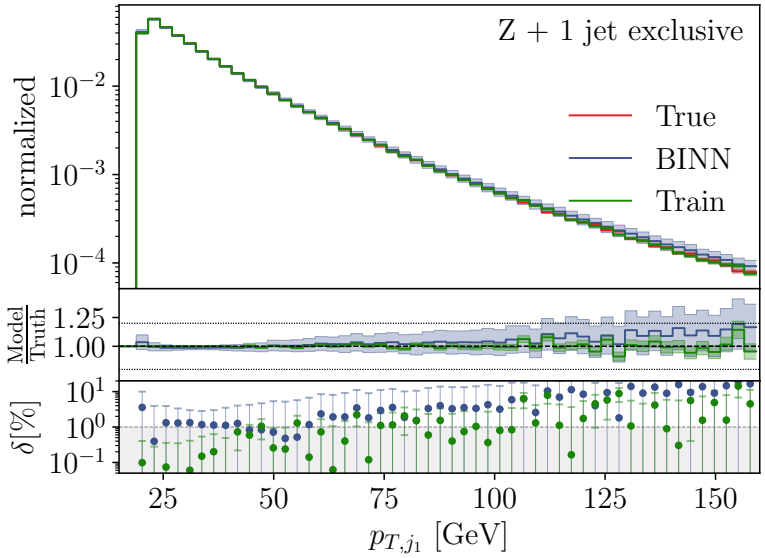


Precision generator with uncertainties

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



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

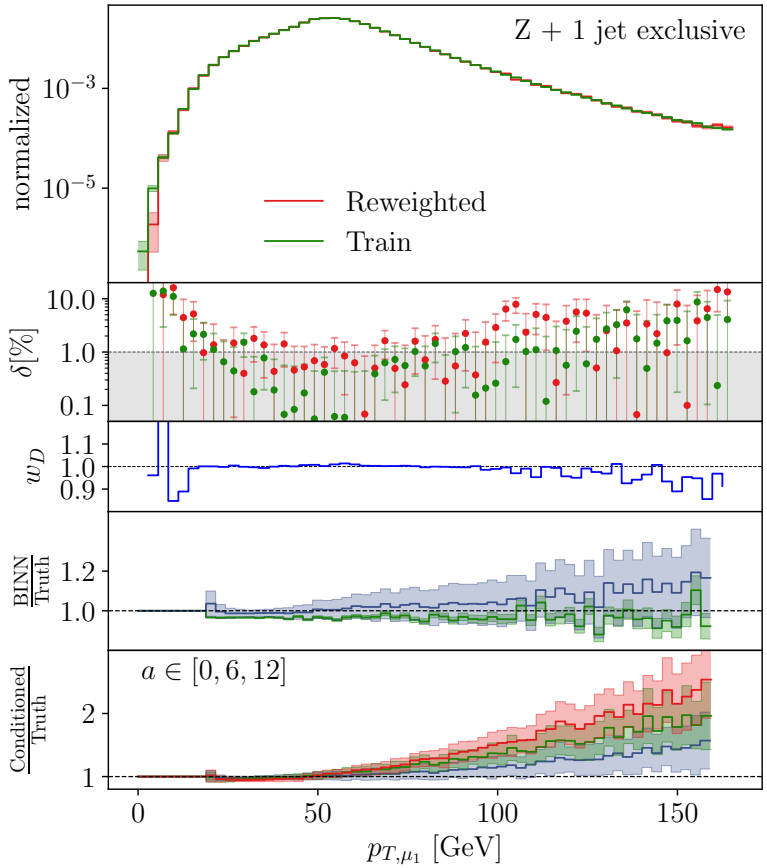
Theory uncertainties

- 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 [$a = 0 \ldots 30$]
- train conditionally on a
  error bar from sampling a

$\Rightarrow$ Controlled per-cent precision



Controlling generative networks

Compare generated with training data [2305.16774]

- 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 [2305.16774]

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

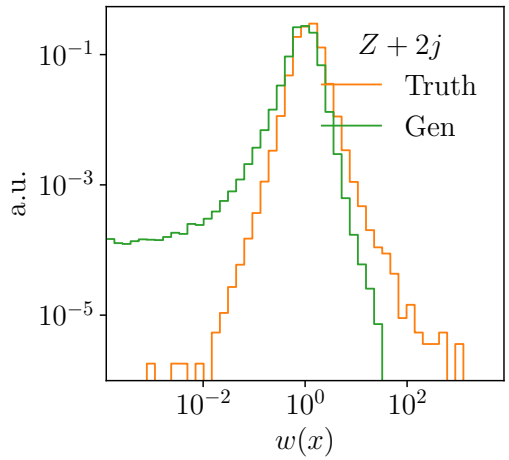
$$w(x_i) = \frac{D(x_i)}{1 - D(x_i)} = \frac{p_{\text{data}}(x_i)}{p_{\text{model}}(x_i)}$$

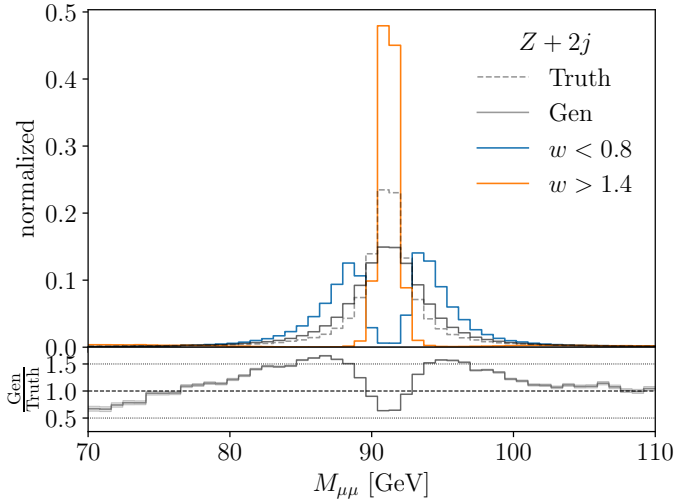
- classifier more precise and reliable

$\rightarrow$ 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 [2305.16774]

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

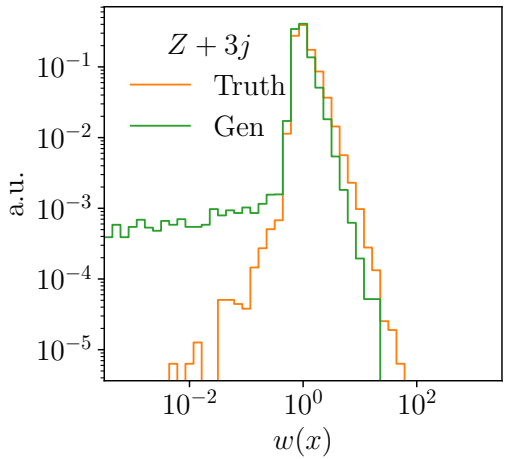
$$w(x_i) = \frac{D(x_i)}{1 - D(x_i)} = \frac{p_{\text{data}}(x_i)}{p_{\text{model}}(x_i)}$$

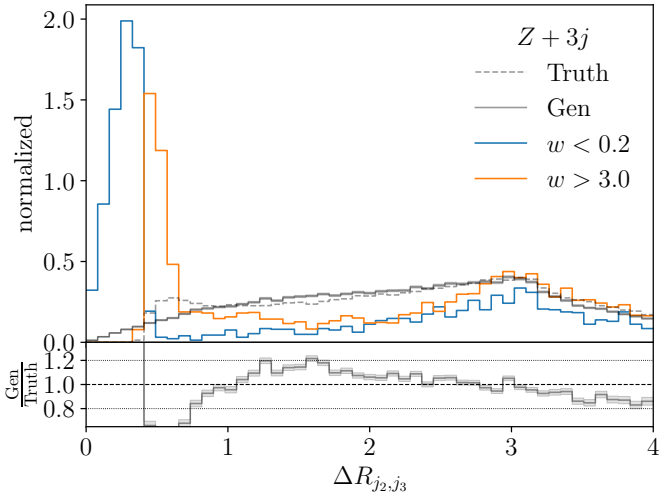
- classifier more precise and reliable

$\to$ Weight ratio over interpretable phase space

Event generators [same for jets, calorimeter showers]

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





$\to$ Generative xAI

Conditional flow matching

Diffusion, better than flows [2305.10475]

- 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 [2305.10475]

- 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]

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

- linear interpolation conditional on data distribution

$$x(t|x_0) = (1-t)x_0 + tr \to \begin{cases} x_0 & t \to 0 \\ r \sim \mathcal{N}(0,1) & t \to 1 \end{cases}$$

$$p(x,t|x_0) = \mathcal{N}(x;(1-t)x_0,t)$$

$$v(x(t|x_0),t|x_0) = \frac{dx(t|x_0)}{dt} = -x_0 + r$$

- conditional continuity equation

$$\frac{\partial p(x,t|x_0)}{\partial t} + \nabla_x \left[p(x,t|x_0) v(x,t|x_0) \right] = 0$$

$\to$ evolution for single path done

Conditional flow matching

Unconditional density and velocity

- probability distribution using prior

$$p(x,t) = \int dx_0 \; p(x,t|x_0) \; p_{\text{data}}(x_0)$$

- velocity from continuity equation

$$v(x,t) = \int dx_0 \; \frac{p(x,t|x_0) v(x,t|x_0) p_{\text{data}}(x_0)}{p(x,t)}$$

- regression loss for velocity [no likelihood, fudged BNN option]

$$\mathcal{L}_{\text{CFM}} = \left\langle \left[v_\theta(x(t|x_0),t) - v(x(t|x_0),t|x_0) \right]^2 \right\rangle_{t,x_0 \sim p_{\text{data}},r}$$

Conditional flow matching

Unconditional density and velocity

- probability distribution using prior

$$p(x,t) = \int dx_0 \; p(x,t|x_0) \; p_{\text{data}}(x_0)$$

- velocity from continuity equation

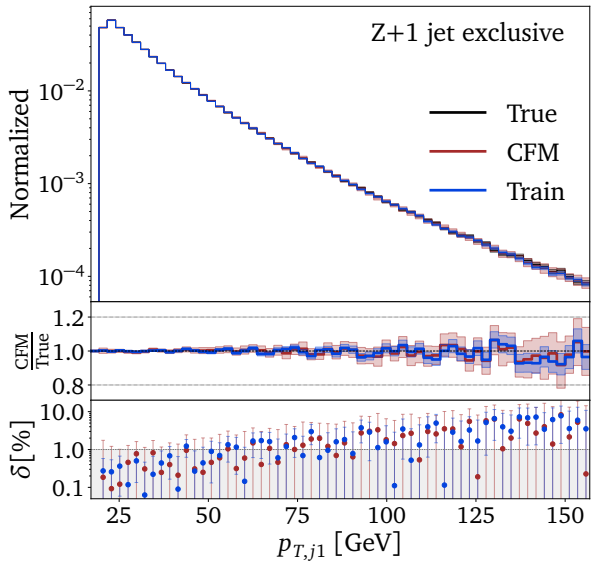
$$v(x,t) = \int dx_0 \; \frac{p(x,t|x_0) v(x,t|x_0) p_{\text{data}}(x_0)}{p(x,t)}$$

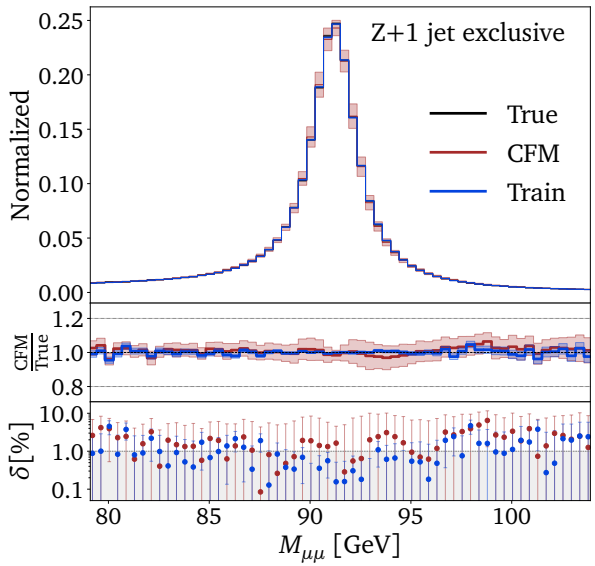
- regression loss for velocity [no likelihood, fudged BNN option]

$$\mathcal{L}_{\text{CFM}} = \left\langle \left[v_\theta(x(t|x_0),t) - v(x(t|x_0),t|x_0) \right]^2 \right\rangle_{t,x_0\sim p_{\text{data}},r}$$

B-CFM for LHC events

- toy models: CFM more expressive
- events:





Conditional flow matching

Unconditional density and velocity

- probability distribution using prior

$$p(x,t) = \int dx_0 \; p(x,t|x_0) \; p_{\text{data}}(x_0)$$

- velocity from continuity equation

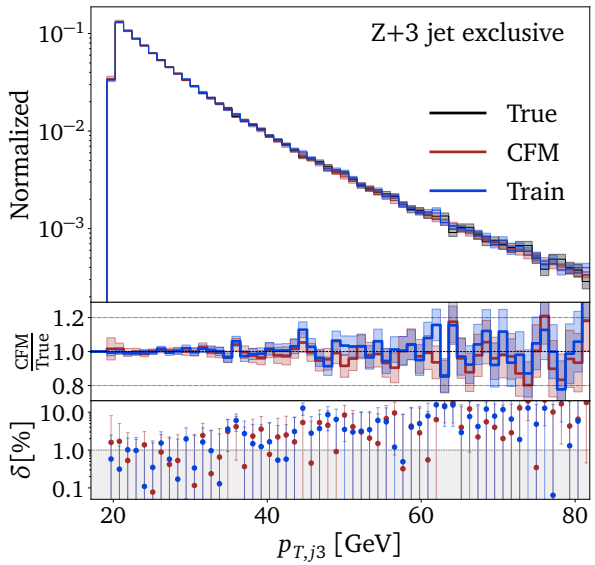
$$v(x,t) = \int dx_0 \; \frac{p(x,t|x_0) v(x,t|x_0) p_{\text{data}}(x_0)}{p(x,t)}$$

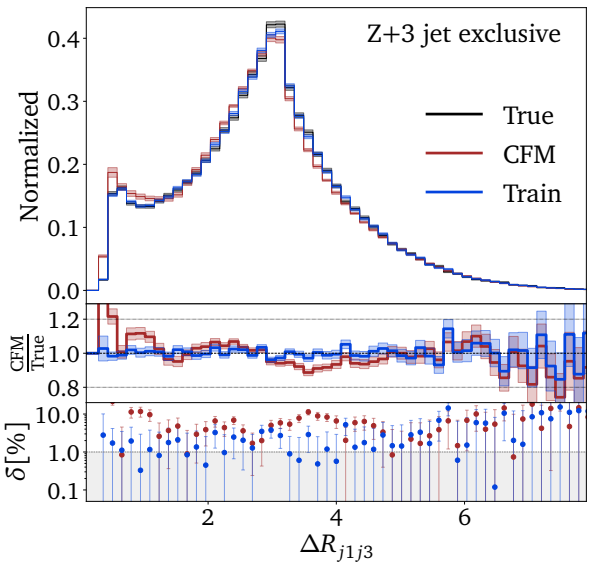
- regression loss for velocity [no likelihood, fudged BNN option]

$$\mathcal{L}_{\text{CFM}} = \left\langle \left[v_\theta(x(t|x_0),t) - v(x(t|x_0),t|x_0) \right]^2 \right\rangle_{t,x_0 \sim p_{\text{data}},r}$$

B-CFM for LHC events

- toy models: CFM more expressive
- events:





Conditional flow matching

Unconditional density and velocity

- probability distribution using prior

$$p(x,t) = \int dx_0 \; p(x,t|x_0) \; p_{\text{data}}(x_0)$$

- velocity from continuity equation

$$v(x,t) = \int dx_0 \; \frac{p(x,t|x_0) v(x,t|x_0) p_{\text{data}}(x_0)}{p(x,t)}$$

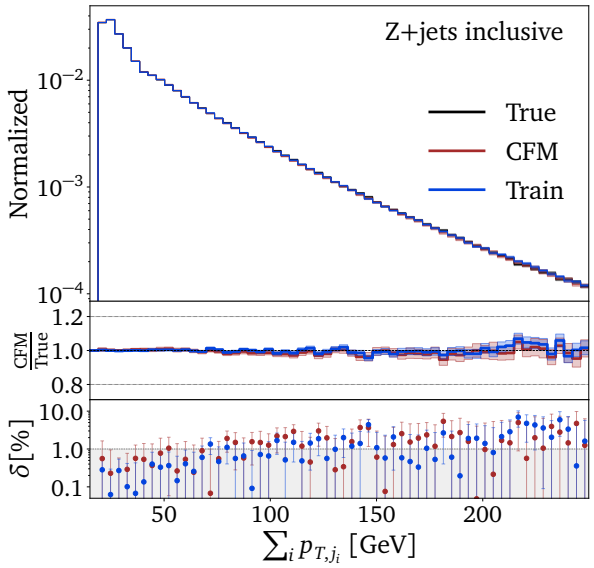
- regression loss for velocity [no likelihood, fudged BNN option]

$$\mathcal{L}_{\text{CFM}} = \left\langle \left[v_\theta(x(t|x_0),t) - v(x(t|x_0),t|x_0) \right]^2 \right\rangle_{t,x_0 \sim p_{\text{data}},r}$$

B-CFM for LHC events

- toy models: CFM more expressive
- events:

$\rightarrow$ Sub-percent precision



Direct diffusion

Structural advantage of CFM model [2311.17175]

- 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

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

$\to$ Similar to Flows4Flows, much easier

Direct diffusion

Structural advantage of CFM model [2311.17175]

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

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

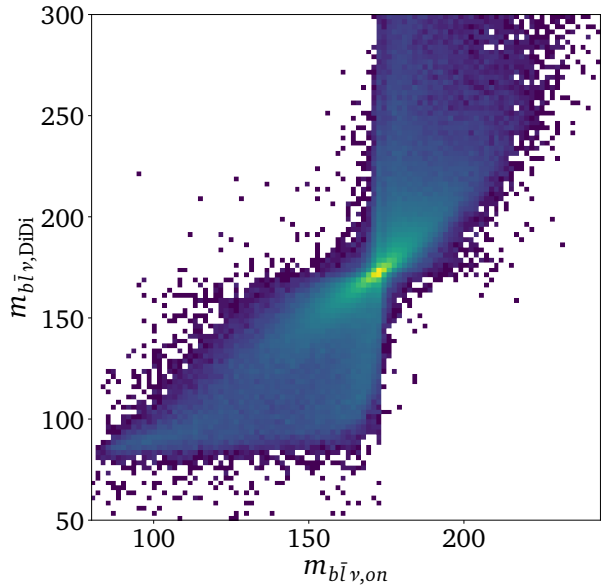
- standard CFM with boundary conditions

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

$\to$ Similar to Flows4Flows, much easier

Precision benefits

- data-driven optimal transport



Direct diffusion

Structural advantage of CFM model [2311.17175]

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

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

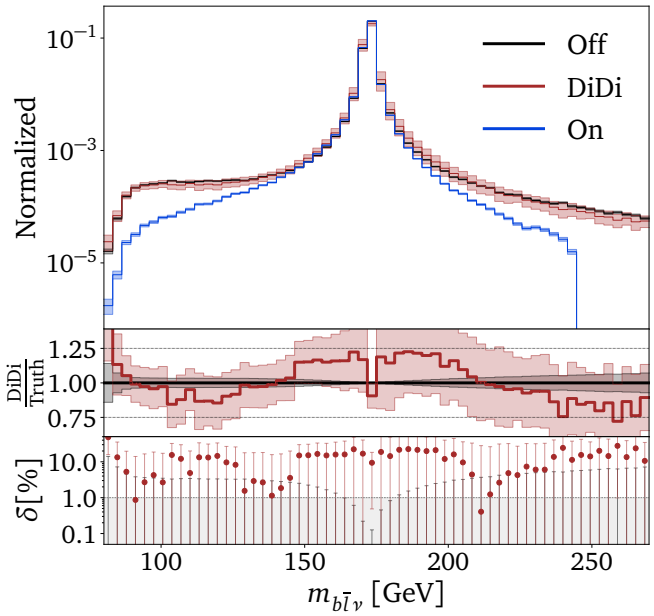
- standard CFM with boundary conditions

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

$\to$ Similar to Flows4Flows, much easier

Precision benefits

- data-driven optimal transport
- high-precision features



Direct diffusion

Structural advantage of CFM model [2311.17175]

- 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

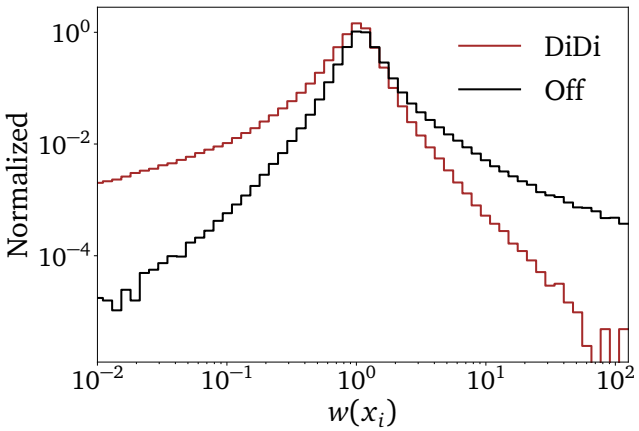
$$p(x,t) \to \begin{cases} p_{\text{off}}(x) & t \to 0 \\ p_{\text{on}}(x) & t \to 1 \end{cases}$$

$\to$ Similar to Flows4Flows, much easier

Precision benefits

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

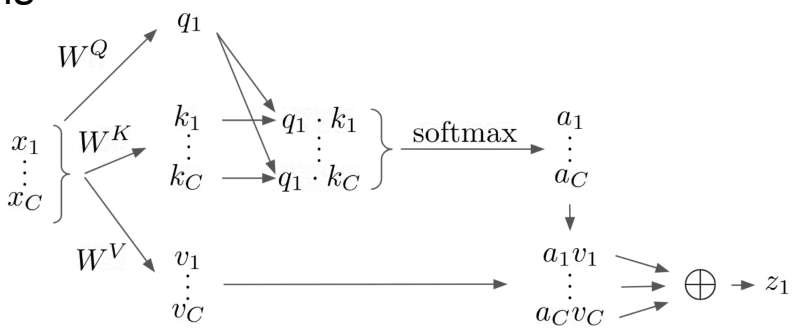
$\to$ More applications?



JetGPT

Correlations through self-attention [2305.10475]

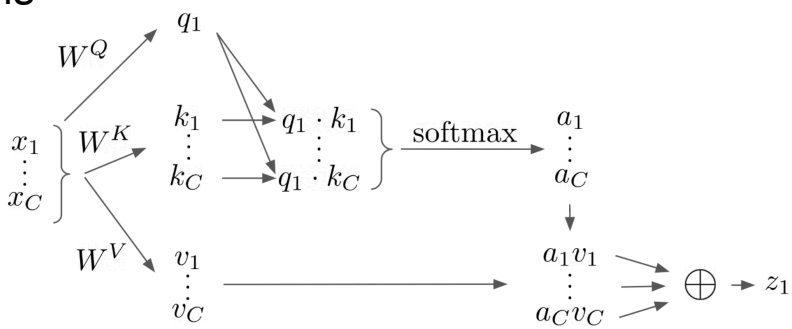
- think of data as bins in phase-space directions
  self-attention: encode relation between bins
  input x, learn relation $x_i \leftrightarrow x_j$
- latent query representation $q = W^Q x$
  latent key representation $k = W^K x$
  define correlation as $A_{ij} = q_i \cdot k_j$
- latent value representation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



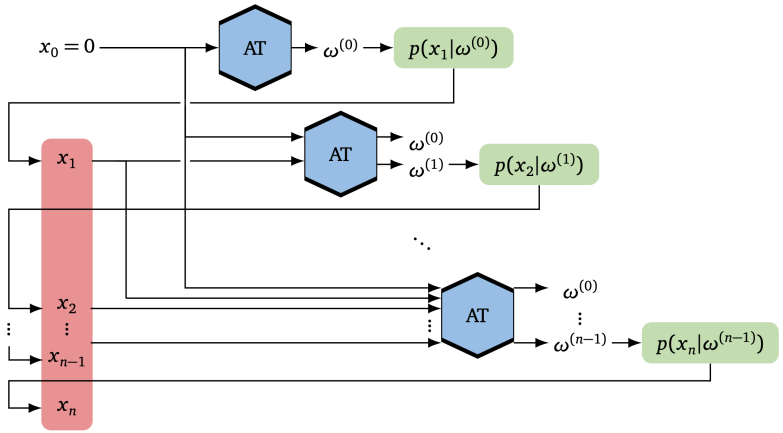
Autoregressive generator

- factorized density

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

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

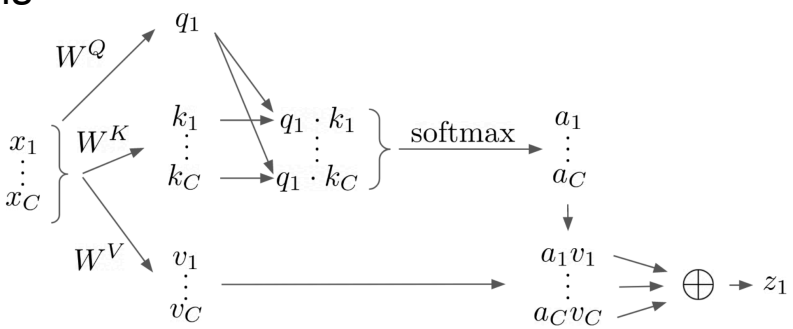
$\rightarrow$ Bayesian version for uncertainties



JetGPT

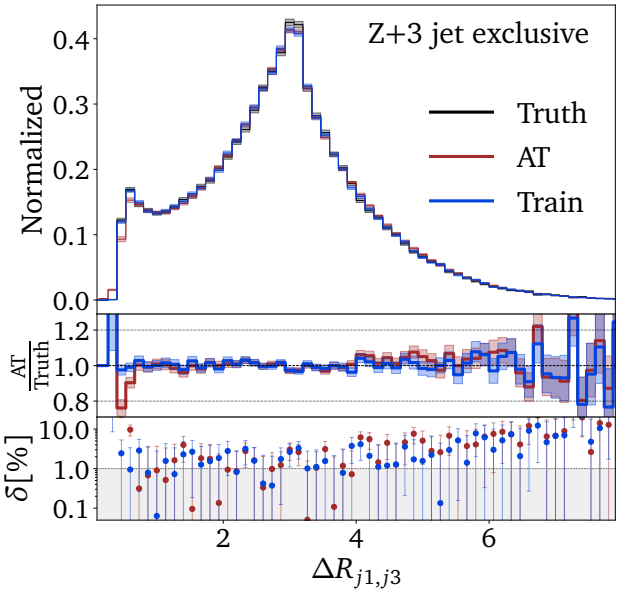
Correlations through self-attention [2305.10475]

- 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

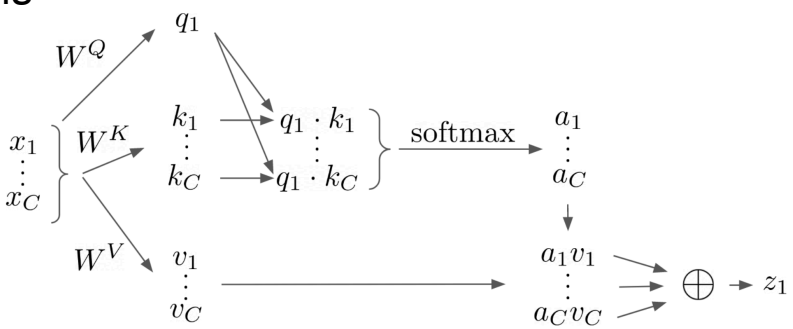
- sometimes you win...



JetGPT

Correlations through self-attention [2305.10475]

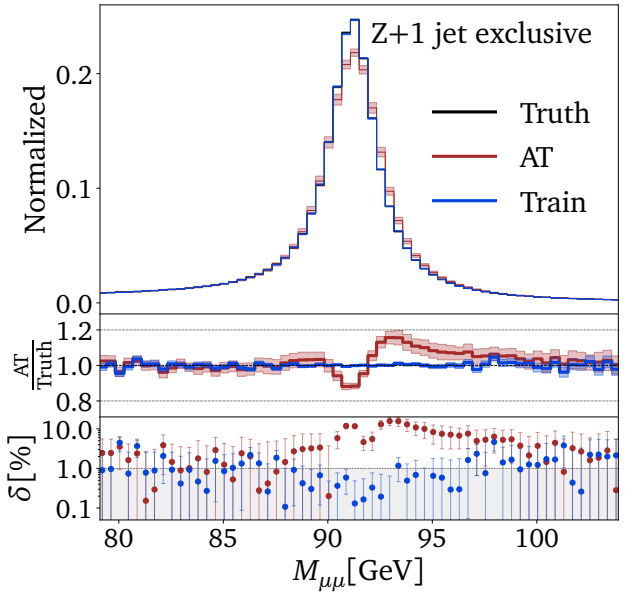
- think of data as bins in phase-space directions
  self-attention: encode relation between bins
  input x, learn relation $x_i \leftrightarrow x_j$
- latent query representation $q = W^Q x$
  latent key representation $k = W^K x$
  define correlation as $A_{ij} = q_i \cdot k_j$
- latent value representation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



Bayesian JetGPT

- sometimes you win...

  ...and sometimes there is work to do...



Towards ML-Madgraph/MadNIS

MadNIS [2212.06172,2311.01548]

- replacing Vegas in Madgraph
- INN with online-buffered training
- multi-channel network

→ Gain over Madgraph

Matrix element method [2210.00019,2310.07752]

- parton-level likelihood per event
- CFM/transformer for evolution
- INN for integration
- efficiency network

→ Method working

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.