ML and Uncertainties

Tilman Plehn

Universität Heidelberg

Freiburg RTG, October 2023

Transformers

Extracting correlations

- Start with (compact) query representation

$$x_i \longrightarrow q = \frac{x_i}{|x|}$$

- Orthonormal values basis [related to q through scalar product]

$$q = \sum_j (q \cdot v_j)\ v_j$$

- Simpler orthogonal keys basis

$$q = \sum_j (q \cdot k_j)\ v_j \quad \text{with} \quad k_j = \frac{v_j}{v^2}$$

$\rightarrow$ Self-attention representation

$$x_i \longrightarrow z_i = \sum_j (q \cdot k_j)\ v_j$$

Transformers

Extracting correlations

- Start with (compact) query representation

$$x_i \longrightarrow q = \frac{x_i}{|x|}$$

- Orthonormal values basis [related to q through scalar product]

$$q = \sum_j (q \cdot v_j)\, v_j$$

- Simpler orthogonal keys basis

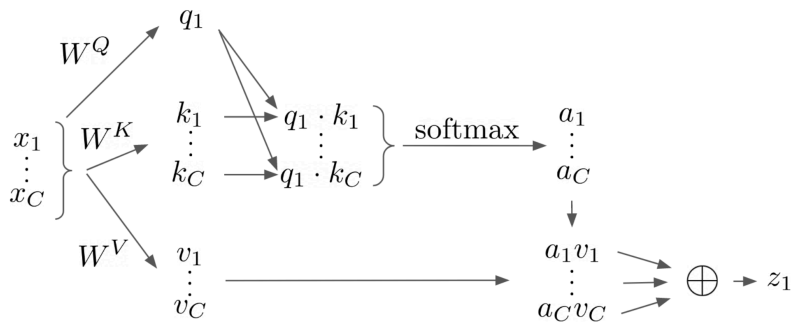
$$q = \sum_j (q \cdot k_j)\, v_j \quad \text{with} \quad k_j = \frac{v_j}{v^2}$$

$\rightarrow$ Self-attention representation

$$x_i \longrightarrow z_i = \sum_j (q \cdot k_j)\, v_j$$

LHC phase space

- learn bin-bin relation $x_i \leftrightarrow x_j$
- latent query representation $q = W^Q x$
  latent key representation $k = W^K x$
  correlation $A_{ij} = q_i \cdot k_j$
- latent value representation $v = W^V x$
  constructed representation $z = A\, v$



JetGPT

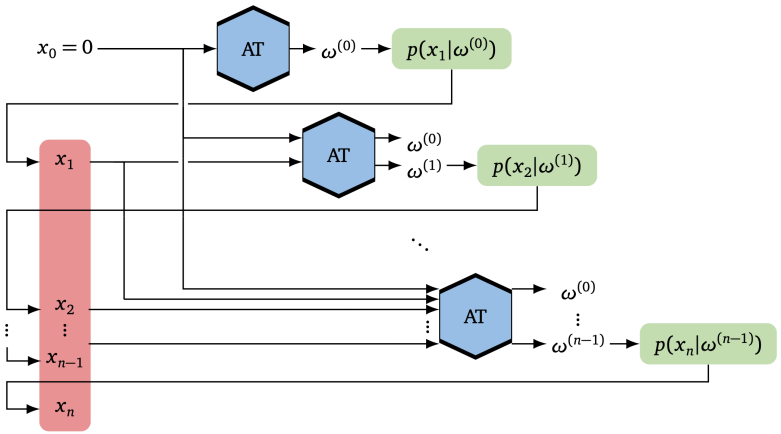
Autoregressive transformer

- factorized density

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

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

$\to$ Bayesian version for uncertainties



JetGPT

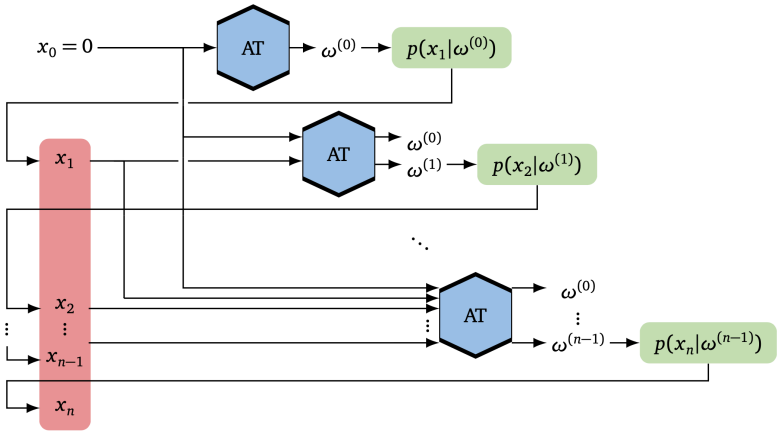
Autoregressive transformer

- factorized density

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

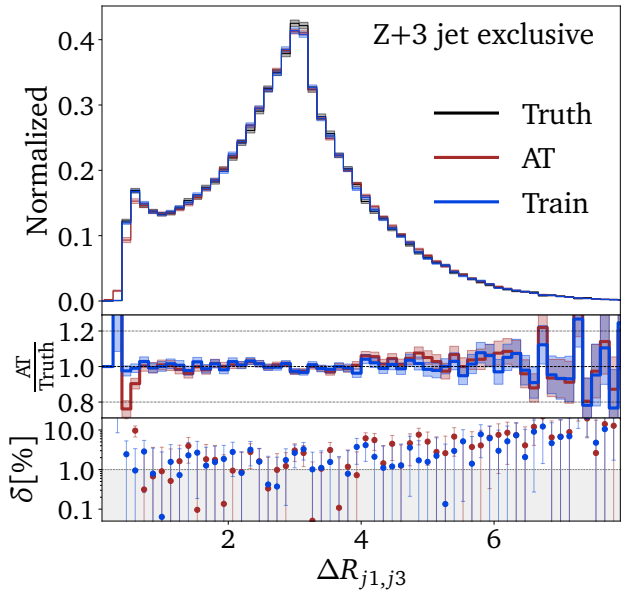
- bins $\to$ Gaussian mixture model
- autoregressive $A_{ij} = 0$ for $j > i$

$\to$ Bayesian version for uncertainties



Bayesian JetGPT

- sometimes you win...



JetGPT

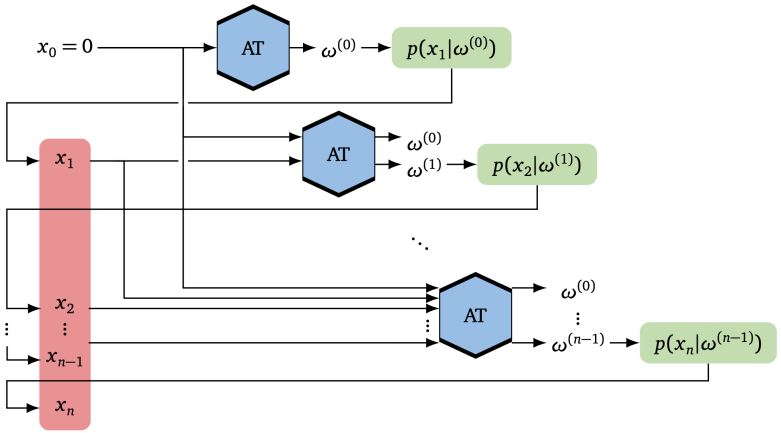
Autoregressive transformer

- 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 $j > i$

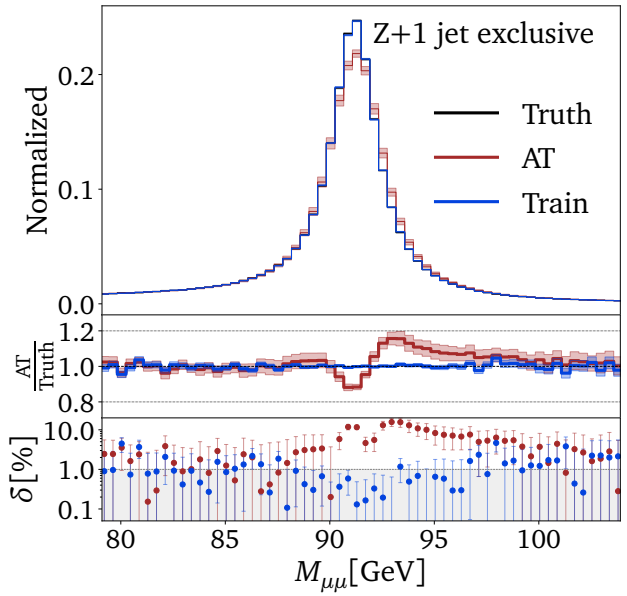
$\rightarrow$ Bayesian version for uncertainties



Bayesian JetGPT

- sometimes you win...

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



Likelihood loss & uncertainties

Loss to train θ-distributions

- energy measurement for jet j

$$\langle E \rangle = \int dE \; E \; p(E)$$

- weighted by reproduced training data $p(\theta|T)$

$$p(E) = \int d\theta \; p(E|\theta) \; p(\theta|T)$$

$\rightarrow$ θ-distributions means Bayesian NN

Likelihood loss & uncertainties

Loss to train θ-distributions

- energy measurement for jet j

$$\langle E \rangle = \int dE \; E \; p(E)$$

- weighted by reproduced training data $p(\theta|T)$

$$p(E) = \int d\theta \; p(E|\theta) \; p(\theta|T)$$

$\rightarrow$ θ-distributions means Bayesian NN

Variational approximation

- definition of training [think $q(\theta)$ as Gaussian with mean and width]

$$p(E) = \int d\theta \; p(E|\theta) \; p(\theta|T) \approx \int d\theta \; p(E|\theta) \; q(\theta)$$

- similarity through minimal KL-divergence [Bayes' theorem to remove unknown posterior]

$$\begin{aligned} D_{\text{KL}}[q(\theta), p(\theta|T)] &= \int d\theta \; q(\theta) \; \log \frac{q(\theta)}{p(\theta|T)} \\ &= \int d\theta \; q(\theta) \; \log \frac{q(\theta)p(T)}{p(T|\theta)p(\theta)} \\ &= D_{\text{KL}}[q(\theta), p(\theta)] - \int d\theta \; q(\theta) \; \log p(T|\theta) + \log p(T) \int d\theta \; q(\theta) \end{aligned}$$

Likelihood loss & uncertainties

Loss to train θ-distributions

- energy measurement for jet j

$$\langle E \rangle = \int dE \; E \; p(E)$$

- weighted by reproduced training data $p(\theta|T)$

$$p(E) = \int d\theta \; p(E|\theta) \; p(\theta|T)$$

$\rightarrow$ θ-distributions means Bayesian NN

Variational approximation

- definition of training [think $q(\theta)$ as Gaussian with mean and width]

$$p(E) = \int d\theta \; p(E|\theta) \; p(\theta|T) \approx \int d\theta \; p(E|\theta) \; q(\theta)$$

- similarity through minimal KL-divergence [Bayes' theorem to remove unknown posterior]

$$\begin{aligned} D_{\text{KL}}[q(\theta), p(\theta|T)] &= \int d\theta \; q(\theta) \; \log \frac{q(\theta)}{p(\theta|T)} \\ &= \int d\theta \; q(\theta) \; \log \frac{q(\theta)p(T)}{p(T|\theta)p(\theta)} \\ &= D_{\text{KL}}[q(\theta), p(\theta)] - \int d\theta \; q(\theta) \; \log p(T|\theta) + \log p(T) \end{aligned}$$

Likelihood loss & uncertainties

Loss to train θ-distributions

- energy measurement for jet j

$$\langle E \rangle = \int dE \; E \; p(E)$$

- weighted by reproduced training data $p(\theta|T)$

$$p(E) = \int d\theta \; p(E|\theta) \; p(\theta|T)$$

$\rightarrow$ θ-distributions means Bayesian NN

Variational approximation

- definition of training [think $q(\theta)$ as Gaussian with mean and width]

$$p(E) = \int d\theta \; p(E|\theta) \; p(\theta|T) \approx \int d\theta \; p(E|\theta) \; q(\theta)$$

- similarity through minimal KL-divergence [Bayes' theorem to remove unknown posterior]

$$\begin{aligned} D_{\text{KL}}[q(\theta), p(\theta|T)] &= \int d\theta \; q(\theta) \; \log \frac{q(\theta)}{p(\theta|T)} \\ &= \int d\theta \; q(\theta) \; \log \frac{q(\theta)p(T)}{p(T|\theta)p(\theta)} \\ &\approx D_{\text{KL}}[q(\theta), p(\theta)] - \int d\theta \; q(\theta) \; \log p(T|\theta) \equiv \mathcal{L} \end{aligned}$$

$\rightarrow$ Two-term loss: likelihood + prior

Relation to deterministic networks

Regularization

- BNN loss

$$\mathcal{L} = -\int d\theta \; q(\theta) \; \log p(T|\theta) + D_{\text{KL}}[q(\theta), p(\theta)]$$

Relation to deterministic networks

Regularization

- Gaussian prior

$$\mathcal{L} = -\int d\theta \; q(\theta) \; \log p(T|\theta) + \frac{\sigma_q^2 - \sigma_p^2 + (\mu_q - \mu_p)^2}{2\sigma_p^2} + \dots$$

- deterministic network

$$q(\theta) = \delta(\theta - \theta_0) \quad \Rightarrow \quad \mathcal{L} \approx -\log p(T|\theta_0) + \frac{(\theta_0 - \mu_p)^2}{2\sigma_p^2}$$

$\rightarrow$ Likelihood with L2-regularization

Relation to deterministic networks

Regularization

- Gaussian prior

$$\mathcal{L} = -\int d\theta\ q(\theta)\ \log p(T|\theta) + \frac{\sigma_q^2 - \sigma_p^2 + (\mu_q - \mu_p)^2}{2\sigma_p^2} + \dots$$

- deterministic network

$$q(\theta) = \delta(\theta - \theta_0) \quad \Rightarrow \quad \mathcal{L} \approx -\log p(T|\theta_0) + \frac{(\theta_0 - \mu_p)^2}{2\sigma_p^2}$$

$\to$ Likelihood with L2-regularization

Dropout

- Bernoulli weights

$$q(\theta) \to q(x) = \rho^x (1-\rho)^{1-x} \Big|_{x=0,1} \quad \text{with} \quad \theta = x\theta_0$$

- likelihood loss

$$\mathcal{L} = -\sum_{x=0,1} \left[\rho^x (1-\rho)^{1-x}\right]\ \log p(T|x\theta_0) = -\rho\ \log p(T|\theta_0)$$

- likelihood Gaussian or whatever else...

$\to$ Regularized likelihood with dropout

Statistics vs systematics

Network evaluation

- expectation value using trained network $q(\theta)$

$$\langle E \rangle = \int dE d\theta \; E \; p(E|\theta) \; q(\theta)$$

$$\equiv \int d\theta \; q(\theta) \overline{E}(\theta) \qquad \text{with} \qquad \overline{E}(\theta) = \int dE \; E \; p(E|\theta)$$

- NN-variance

$$\sigma^2_{\text{tot}} = \int dE d\theta \; \left(E - \langle E \rangle\right)^2 \; p(E|\theta) \; q(\theta)$$

$$= \int d\theta \; q(\theta) \left[\overline{E^2}(\theta) - 2\langle E \rangle \overline{E}(\theta) + \langle E \rangle^2\right]$$

$$= \int d\theta \; q(\theta) \left[\overline{E^2}(\theta) - \overline{E}(\theta)^2 + \left(\overline{E}(\theta) - \langle E \rangle\right)^2\right] \equiv \sigma^2_{\text{stoch}} + \sigma^2_{\text{pred}}$$

Two uncertainties

- statistical — vanishing for $q(\theta) \to \delta(\theta - \theta_0)$

$$\sigma^2_{\text{pred}} = \int d\theta \; q(\theta) \left[\overline{E}(\theta) - \langle E \rangle\right]^2$$

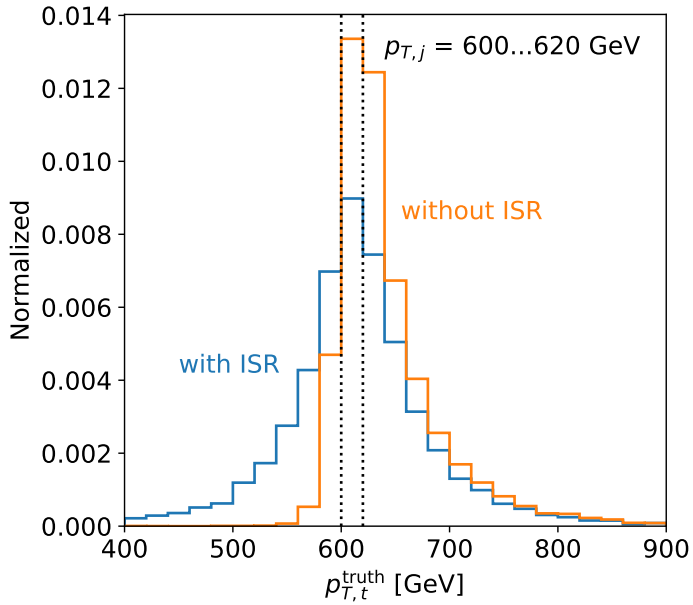
- systematic — vanishing for $p(E|\theta) \to \delta(E - E_0)$

$$\sigma^2_{\text{stoch}} \equiv \sigma^2_{\text{model}} = \int d\theta \; q(\theta) \left[\overline{E^2}(\theta) - \overline{E}(\theta)^2\right]$$

Jets and non-Gaussian errors

Measure $p_{T,t}$ of hadronically decaying top

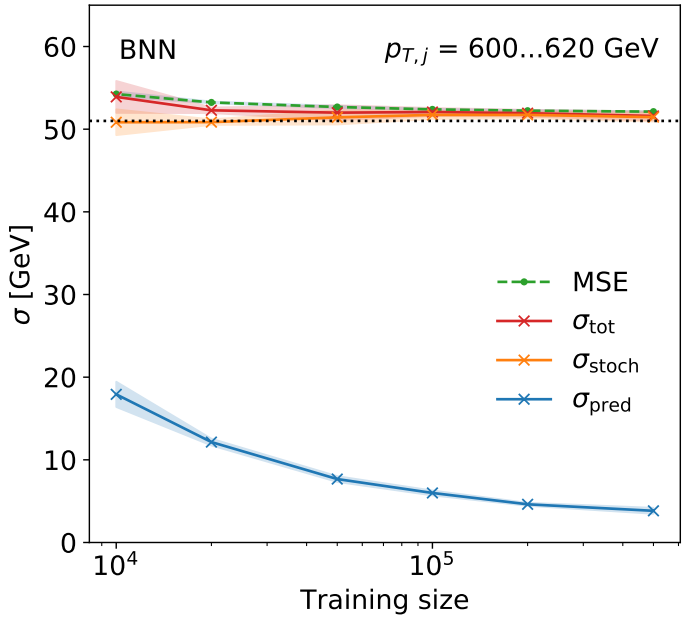
- BNN regression $p_{T,t}$
  p_T of (fat) jet decent estimate for $p_{T,t}^{\text{truth}}$
- non-Gaussian truth label
  symmetric in ISR-jet 'QCD heat bath'
  without ISR jets need for correction



Jets and non-Gaussian errors

Measure $p_{T,t}$ of hadronically decaying top

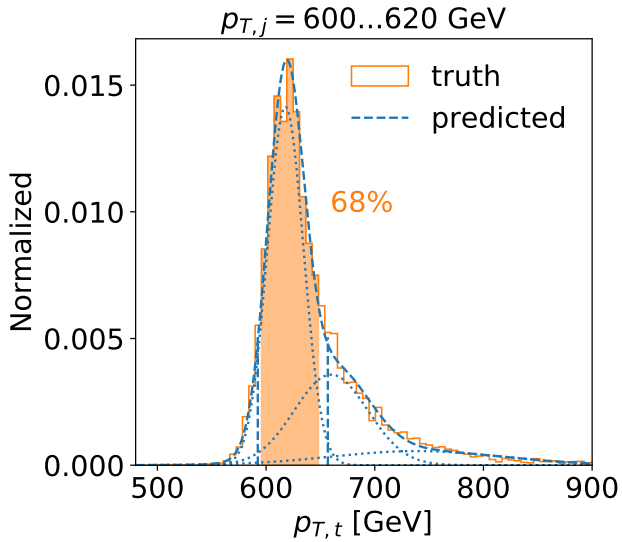
- BNN regression $p_{T,t}$
  p_T of (fat) jet decent estimate for $p_{T,t}^{\text{truth}}$
- non-Gaussian truth label
  symmetric in ISR-jet 'QCD heat bath'
  without ISR jets need for correction
- training sample size
  separate $\sigma_{\text{stoch}} \gg \sigma_{\text{pred}}$
  statistics not the problem [LHC theme]
  noisy label inherent limitation
  checked with ensembles



Jets and non-Gaussian errors

Measure $p_{T,t}$ of hadronically decaying top

- BNN regression $p_{T,t}$
  p_T of (fat) jet decent estimate for $p_{T,t}^{\text{truth}}$
- non-Gaussian truth label
  symmetric in ISR-jet 'QCD heat bath'
  without ISR jets need for correction
- training sample size
  separate $\sigma_{\text{stoch}} \gg \sigma_{\text{pred}}$
  statistics not the problem [LHC theme]
  noisy label inherent limitation
  checked with ensembles
- non-Gaussian network output
  remember $p_{T,t}^{\text{truth}}$ non-Gaussian
  model $p(T|\theta)$ as Gaussian mixture
  weight distribution $q(\theta)$ still Gaussian



Generative networks

Unsupervised Bayesian networks

- data: event sample [points in 2D space]

  learn phase space density
  normalizing flow mapping to latent space [INN]
  standard distribution in latent space [Gaussian]
  mapping bijective
  sample from latent space

- Bayesian version

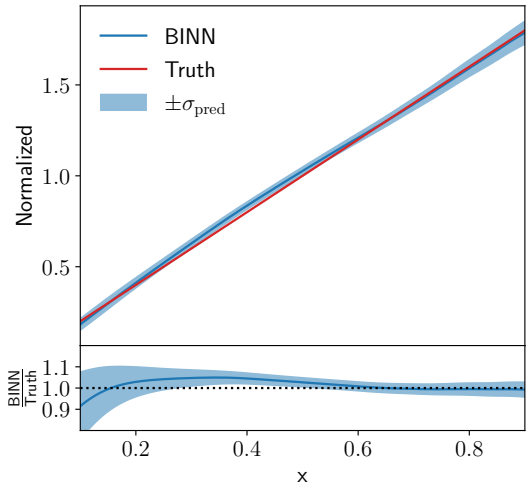
  allow weight distributions
  learn uncertainty map

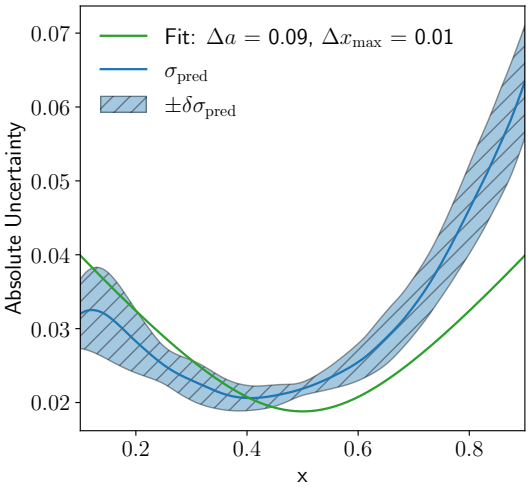
- 2D wedge ramp

$$p(x) = ax + b = ax + \frac{1 - \frac{a}{2}(x_{\text{max}}^2 - x_{\text{min}}^2)}{x_{\text{max}} - x_{\text{min}}}$$

$$(\Delta p)^2 = \left(x - \frac{1}{2}\right)^2 (\Delta a)^2 + \left(1 + \frac{a}{2}\right)^2 (\Delta x_{\text{max}})^2 + \left(1 - \frac{a}{2}\right)^2 (\Delta x_{\text{min}})^2$$

  explaining minimum in $\sigma_{\text{pred}}(x)$



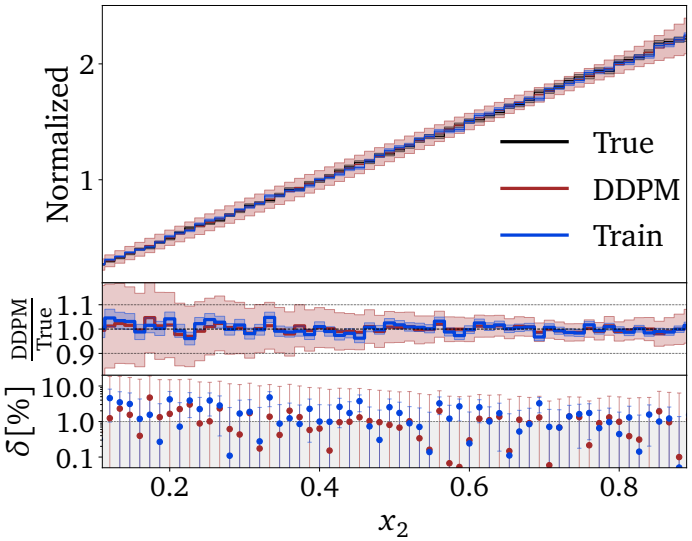


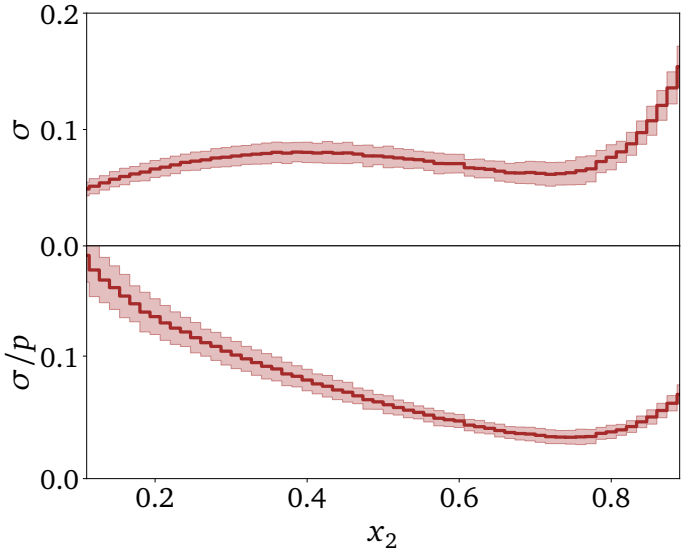
→ INNs just (non-parametric) fits

More generative networks

Alternative architectures

- always a fit?
- expressivity vs architecture?
- discrete diffusion model

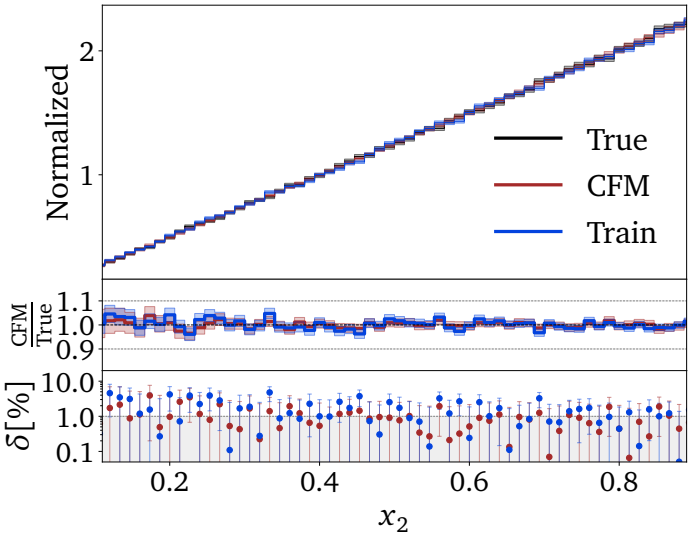


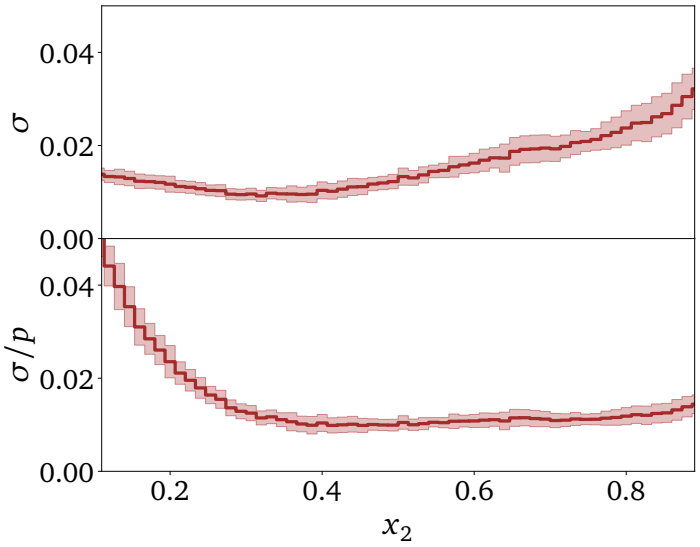


More generative networks

Alternative architectures

- always a fit?
- expressivity vs architecture?
- discrete diffusion model
- continuous diffusion model

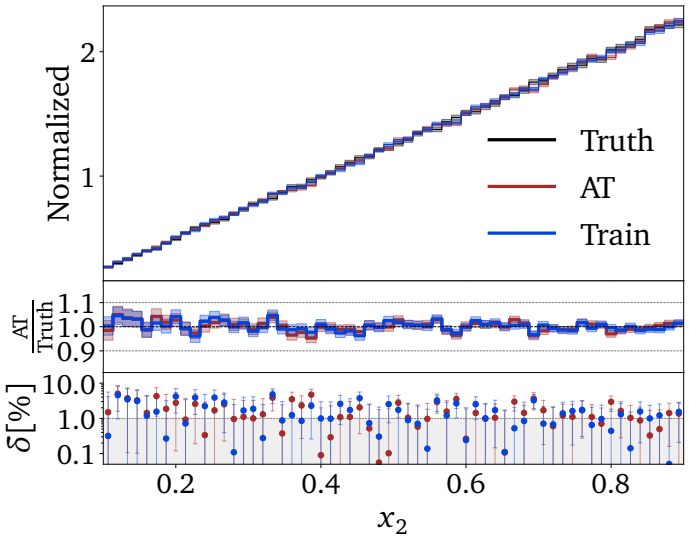


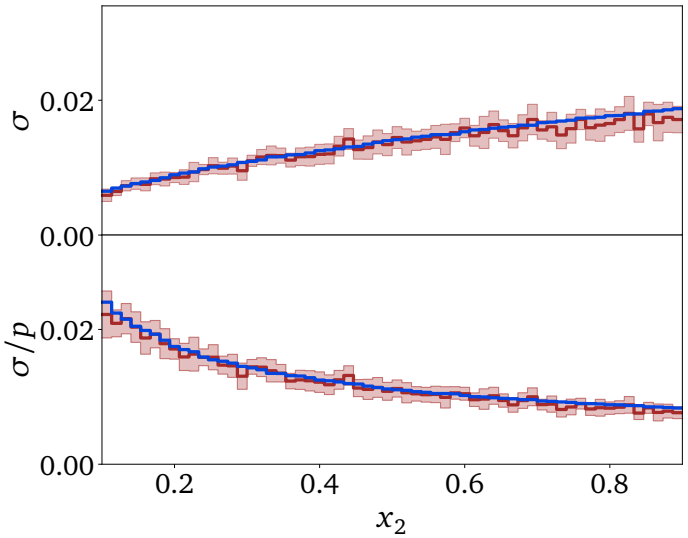


More generative networks

Alternative architectures

- always a fit?
- expressivity vs architecture?
- discrete diffusion model
- continuous diffusion model
- autoregressive transformer (bins)
-





Precision generator

Phase-space generators [typical LHC task]

- training from event samples
  no energy-momentum conservation
- every correlation counts
- $Z_{\mu\mu} + \{1, 2, 3\}$ jets [Z-peak, variable jet number, jet-jet topology]

Precision generator

Phase-space generators [typical LHC task]

- training from event samples
  no energy-momentum conservation
- every correlation counts
- $Z_{\mu\mu} + \{1,2,3\}$ jets [Z-peak, variable jet number, jet-jet topology]

INN-generator

- 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

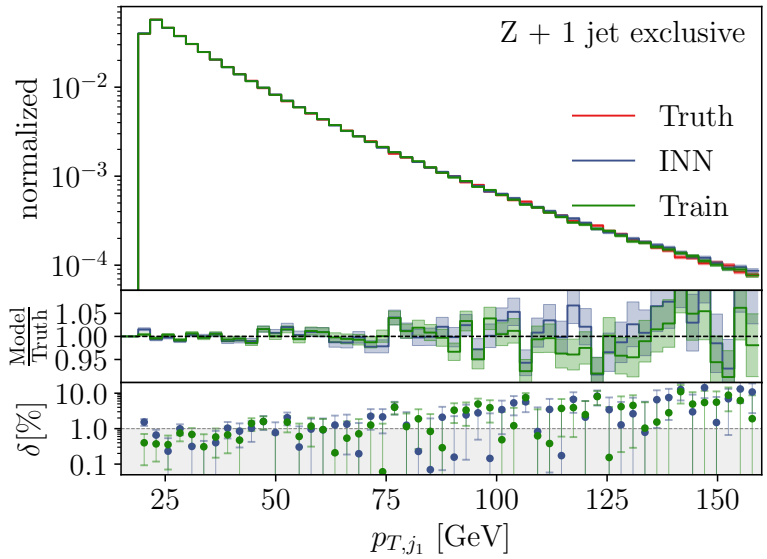
$$dx\; p_\text{model}(x) = dr\; p_\text{latent}(r)$$

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

- likelihood loss

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

$\Rightarrow$ Per-cent precision possible



Control and reweighting

Best of GANs: discriminator

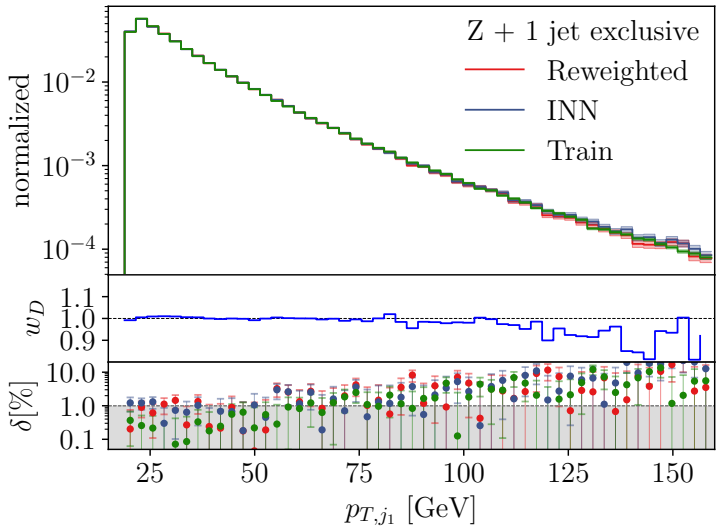
- input $\{p_T, \eta, \phi, M, M_{\mu\mu}, \Delta R\}$
- output $D = 0$ (generator) vs $D = 1$ (training)
- NP-optimal discriminator

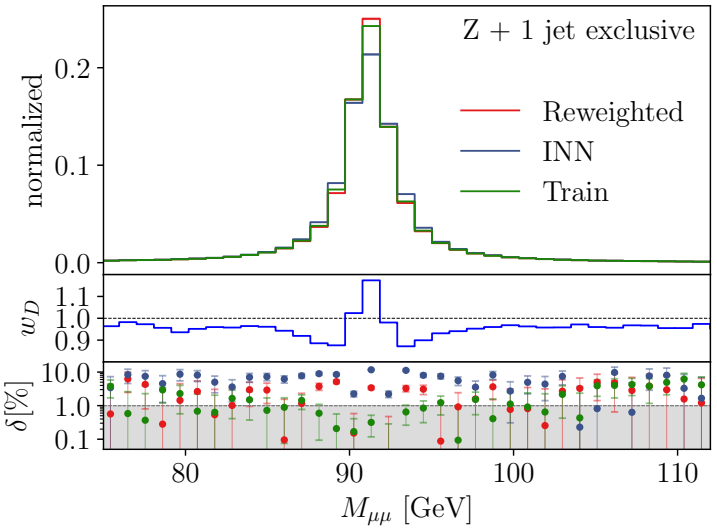
$$D(x) \to \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_{\text{model}}(x)} \to \frac{1}{2}$$

- learned event weight

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

$\Rightarrow$ Dual purpose: control and reweight



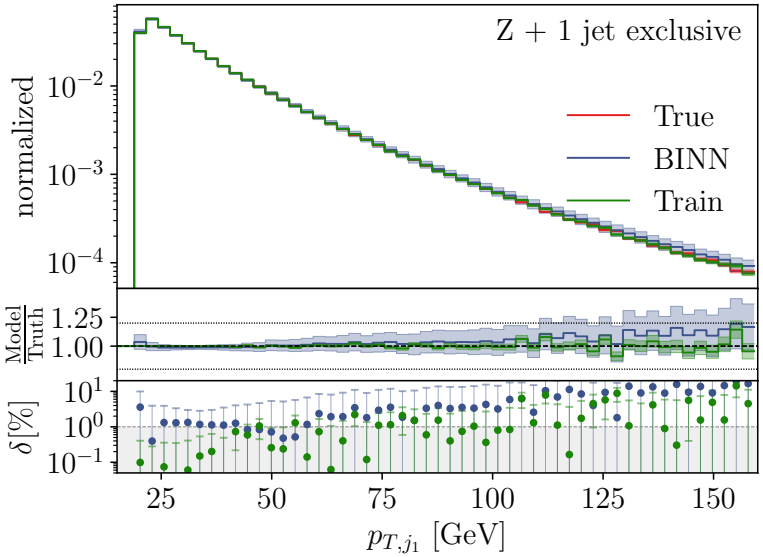


Events with uncertainties

Bayesian network generator

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

$\Rightarrow$ Training-related error bars



Events with uncertainties

Bayesian network generator

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

$\Rightarrow$ Training-related error bars

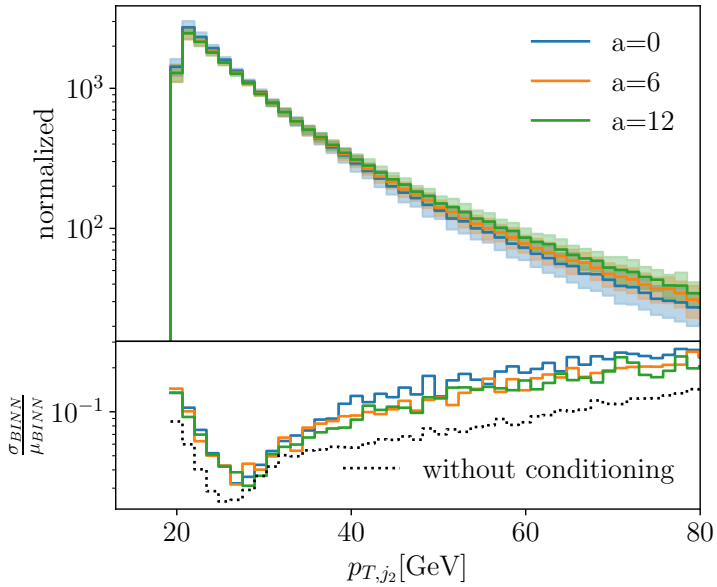
Theory uncertainties

- 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

$\Rightarrow$ Systematic/theory error bars



Events with uncertainties

Bayesian network generator

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

$\Rightarrow$ Training-related error bars

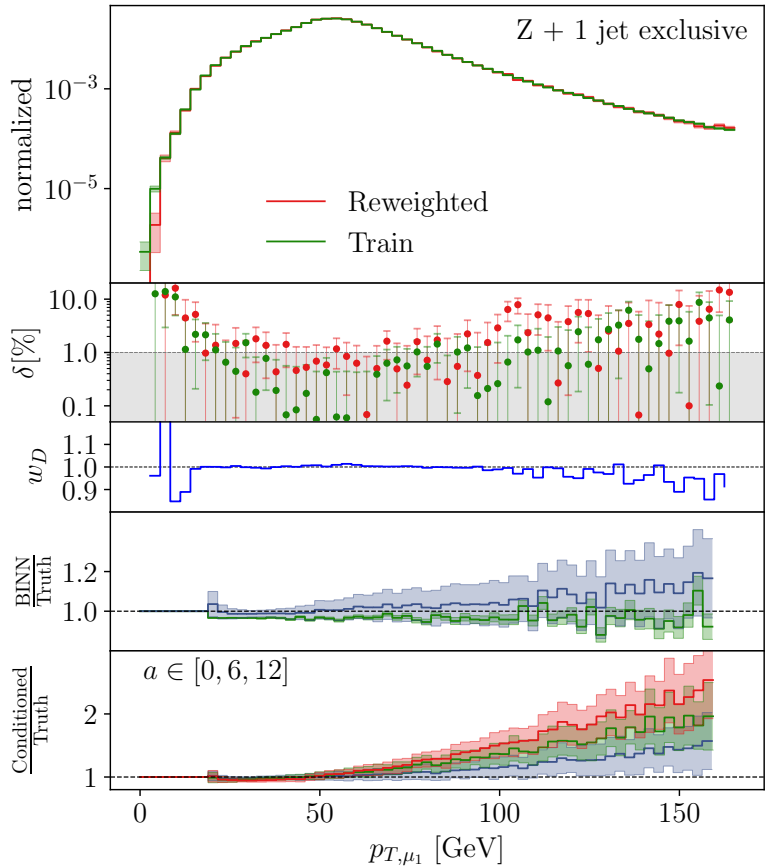
Theory uncertainties

- 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

$\Rightarrow$ Systematic/theory error bars



Testing generative networks

Compare network to training/test data

- supervised: histogram deviation [or pull]
- unsupervised density $\to$ histogram discriminator

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

$\to$ Using interpretable phase space

Testing generative networks

Compare network to training/test data

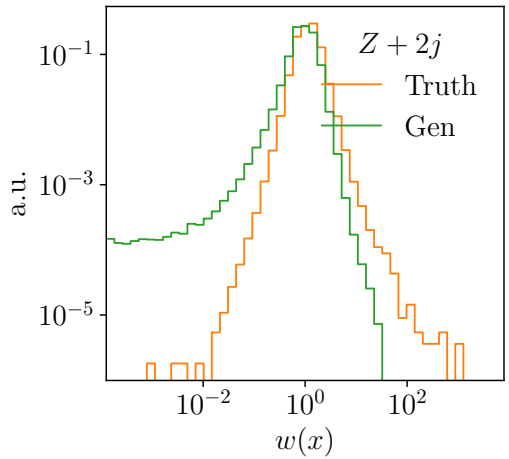
- supervised: histogram deviation [or pull]
- unsupervised density $\to$ histogram discriminator

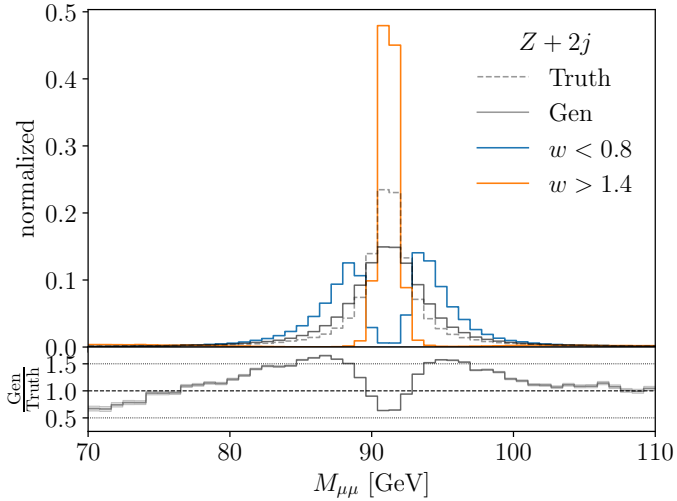
$$w(x_i) = \frac{D(x_i)}{1 - D(x_i)} = \frac{p_{\text{data}}(x_i)}{p_{\text{model}}(x_i)}$$

$\to$ Using interpretable phase space

Applied to event generators [also jets, calorimeter showers]

- shape and width of w-histogram
- pattern in (interpretable) phase space?





Testing generative networks

Compare network to training/test data

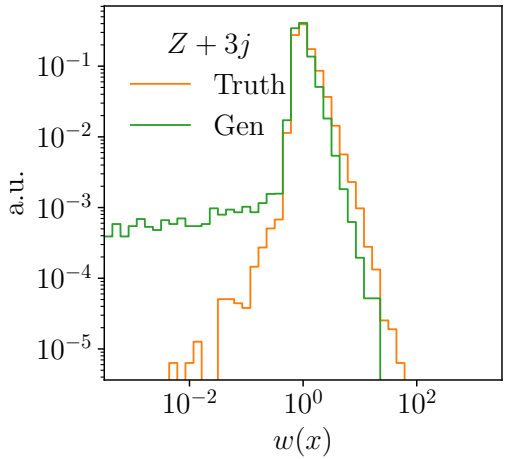
- supervised: histogram deviation [or pull]
- unsupervised density $\to$ histogram discriminator

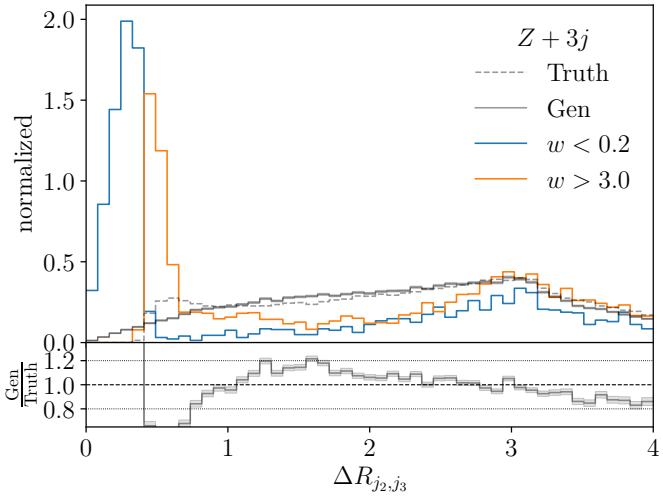
$$w(x_i) = \frac{D(x_i)}{1 - D(x_i)} = \frac{p_{\text{data}}(x_i)}{p_{\text{model}}(x_i)}$$

$\to$ Using interpretable phase space

Applied to event generators [also jets, calorimeter showers]

- shape and width of w-histogram
- pattern in (interpretable) phase space?





$\to$ Generative xAI for LHC physicists

Symbolic regression

Symbolic regression of score [PySR (M Cranmer) + final fit]

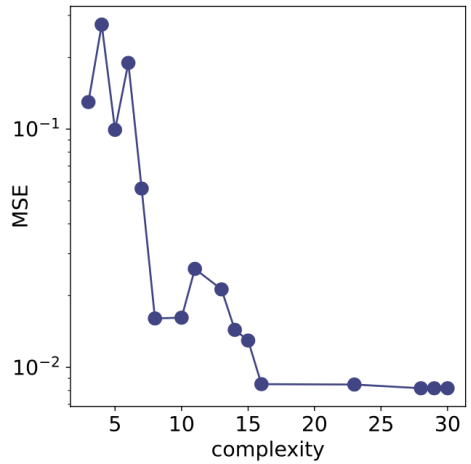
- function to approximate $t(x|\theta)$
- phase space parameters $x_p = p_T/m_H, \Delta\eta, \Delta\phi$ [node]
- operators $\sin x, x^2, x^3, x+y, x-y, x*y, x/y$ [node]
- represent formula as tree [complexity = number of nodes]

$\Rightarrow$ Figures of merit

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}\Big[g_i(x) - t(x,z|\theta)\Big]^2 \to \text{MSE} + \text{parsimony}\cdot\text{complexity}$$

Score around Standard Model

compl	dof	function	MSE
3	1	$a\,\Delta\phi$	$1.30\cdot10^{-1}$
4	1	$\sin(a\Delta\phi)$	$2.75\cdot10^{-1}$
5	1	$a\Delta\phi x_{p,1}$	$9.93\cdot10^{-2}$
6	1	$-x_{p,1}\sin(\Delta\phi+a)$	$1.90\cdot10^{-1}$
7	1	$(-x_{p,1}-a)\sin(\sin(\Delta\phi))$	$5.63\cdot10^{-2}$
8	1	$(a-x_{p,1})x_{p,2}\sin(\Delta\phi)$	$1.61\cdot10^{-2}$
14	2	$x_{p,1}(a\Delta\phi-\sin(\sin(\Delta\phi)))(x_{p,2}+b)$	$1.44\cdot10^{-2}$
15	3	$-(x_{p,2}(a\Delta\eta^2+x_{p,1})+b)\sin(\Delta\phi+c)$	$1.30\cdot10^{-2}$
16	4	$-x_{p,1}(a-b\Delta\eta)(x_{p,2}+c)\sin(\Delta\phi+d)$	$8.50\cdot10^{-3}$
28	7	$(x_{p,2}+a)(bx_{p,1}(c-\Delta\phi)$ $-x_{p,1}(d\Delta\eta+ex_{p,2}+f)\sin(\Delta\phi+g))$	$8.18\cdot10^{-3}$



Symbolic regression

Symbolic regression of score [PySR (M Cranmer) + final fit]

- function to approximate $t(x|\theta)$
- phase space parameters $x_p = p_T/m_H, \Delta\eta, \Delta\phi$ [node]
- operators $\sin x, x^2, x^3, x+y, x-y, x*y, x/y$ [node]
- represent formula as tree [complexity = number of nodes]

$\Rightarrow$ Figures of merit

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}\Big[g_i(x) - t(x,z|\theta)\Big]^2 \to \text{MSE} + \text{parsimony} \cdot \text{complexity}$$

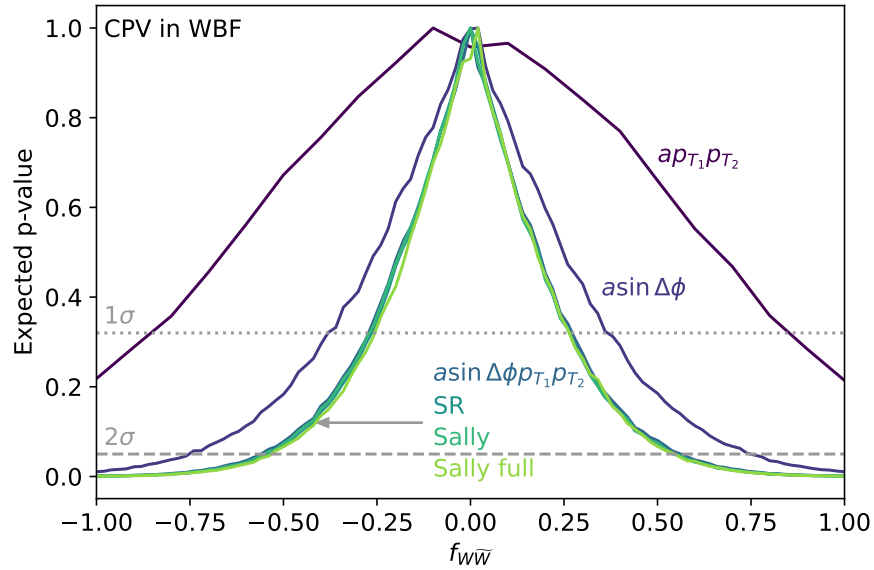
Score around Standard Model

- expected limits:

  very wrong formula
  wrong formula
- same within statistical limitation:

  right formula
  MadMiner

$\Rightarrow$ Formulas to numerics and back



ML Uncertainties

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

→ 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.