Representation Learning for LHC Physics, All Quantum

Tilman Plehn

Universität Heidelberg

Quantum100⊗AI Münster, November 2025

Neural networks in physics

Similar to fit

- approximate $f_\theta(x) \approx f(x)$
- x low-D phase space
- f_θ numerical function

$\rightarrow$ θ data representation

Probabilities over phase space

- regression $\quad x \to A_\theta(x)$
- classification $\quad x \to p_\theta(x)$ [likelihood ratio]
- generation $\quad r \sim \mathcal{N} \to x \sim p_\theta(x)$
- conditional generation $\quad r \sim \mathcal{N} \to x \sim p_\theta(x|y)$

LHC-specific ML

- accuracy
- precision
- structures

$\rightarrow$ Physics-specific representation learning

Training

Learned scalar field $f_\theta(x) \approx f(x)$

- maximize parameter probability given (f_j, σ_j)

$$\theta = \text{argmax}\; p(\theta|x) = \text{argmax}\; \frac{p(x|\theta)\; p(\theta)}{p(x)}$$

$\rightarrow$ Gaussian likelihood loss

$$p(x|\theta) \propto \prod_j \exp\left(-\frac{|f_j - f_\theta(x_j)|^2}{2\sigma_j^2}\right)$$

$$\Rightarrow \qquad \mathcal{L} \equiv -\log p(x|\theta) = \sum_j \frac{|f_j - f_\theta(x_j)|^2}{2\sigma_j^2}$$

Training

Learned scalar field $f_\theta(x) \approx f(x)$

- maximize parameter probability given (f_j, σ_j)

$$\theta = \text{argmax}\; p(\theta|x) = \text{argmax}\; \frac{p(x|\theta)\; p(\theta)}{p(x)}$$

$\rightarrow$ Gaussian likelihood loss

$$p(x|\theta) \propto \prod_j \exp\left(-\frac{|f_j - f_\theta(x_j)|^2}{2\sigma_j^2}\right)$$

$$\Rightarrow \qquad \mathcal{L} \equiv -\log p(x|\theta) = \sum_j \frac{|f_j - f_\theta(x_j)|^2}{2\sigma_j^2}$$

Unknown uncertainties

- loss including normalization

$$\mathcal{L} = \frac{|f(x) - f_\theta(x)|^2}{2\sigma_\theta(x)^2} + \log \sigma_\theta(x) + \cdots$$

- if needed replace with Gaussian mixture model
- similar for Bayesian networks and evidential regression

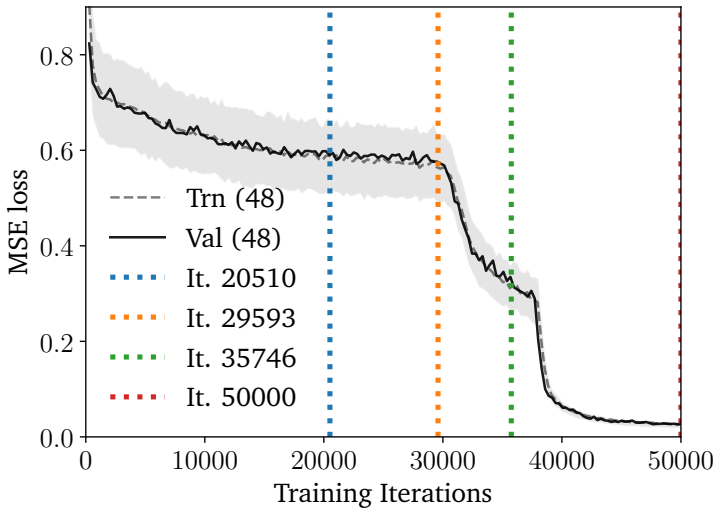
$\rightarrow$ Learning function and (systematic) uncertainty

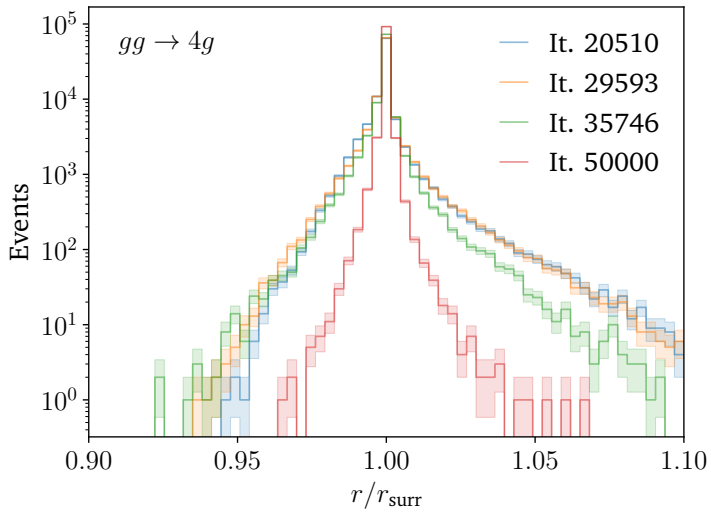
Accuracy from symmetry

Amplitude ratio regression [Villadamigo, Frederix, TP, Vitos, Winterhalder]

- full vs. leading color ratio r for $gg \to 4g$
- standard transformer training [not for MLP, GNN, L-GATr]

$\to$ Related to accuracy gain





Accuracy from symmetry

Amplitude ratio regression [Villadamigo, Frederix, TP, Vitos, Winterhalder]

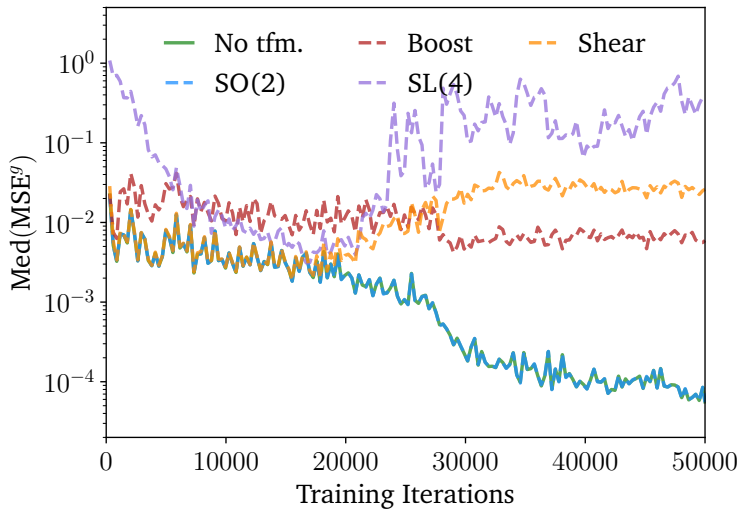
- full vs. leading color ratio r for $gg \to 4g$
- standard transformer training [not for MLP, GNN, L-GATr]

$\to$ Related to accuracy gain

Symmetries?

- evaluate MSE on transformed data

  wrong: 4D rotation SL(4), $y-z$-shear
  right: Lorentz boosts, SO(2) rotations
- initially learning general structures
  then penalizing wrong symmetries
  Lorentz symmetry stable
  SO(2) trivially good
- Symmetries help accuracy

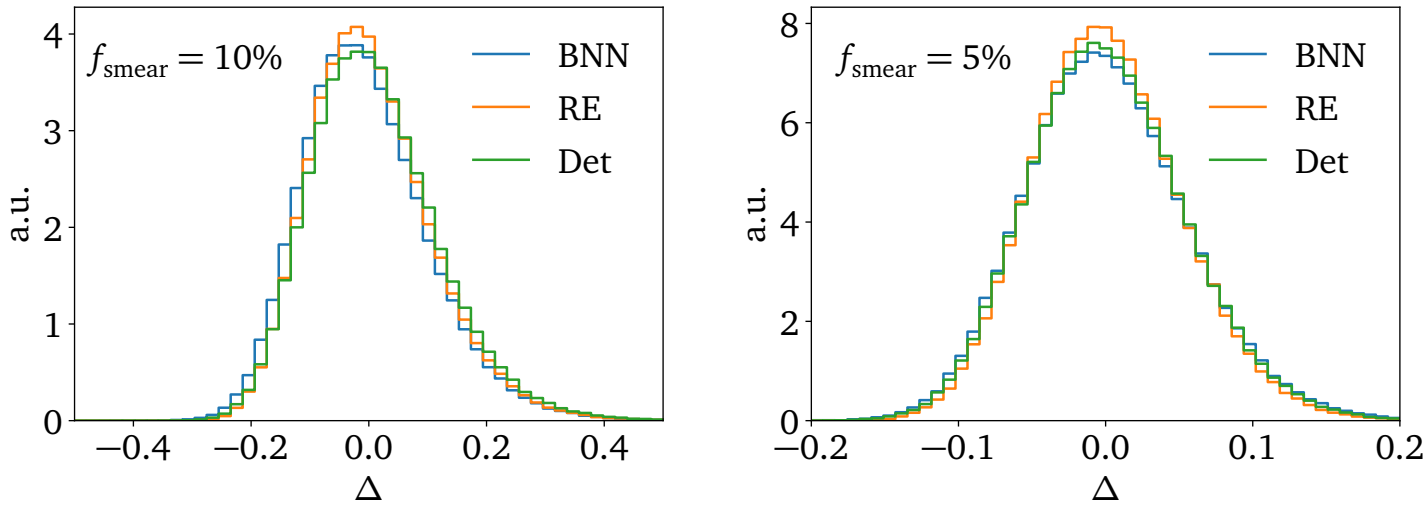


Calibrated uncertainties

Amplitude regression [Badger, Butter, Luchmann, Pitz, TP]

- loop amplitude $gg \to \gamma\gamma g(g)$ over
- systematics: artificial noise
- statistics plateau
- accuracy over phase space

$$\Delta(x) = \frac{A_{\text{NN}}(x) - A_{\text{true}}(x)}{A_{\text{true}}(x)}$$



Calibrated uncertainties

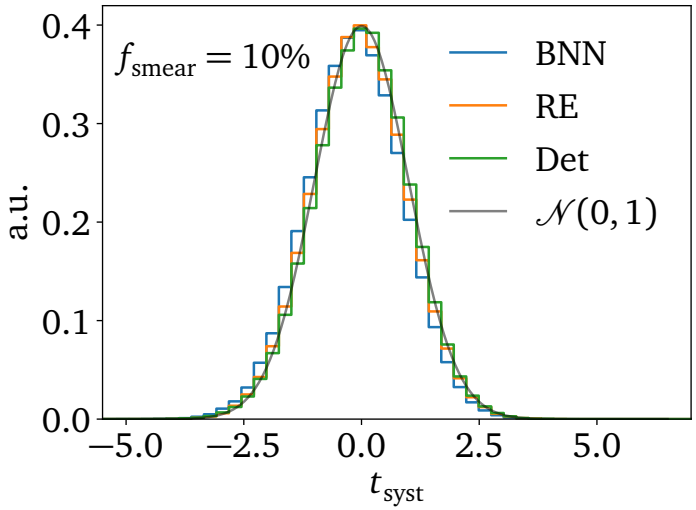
Amplitude regression [Badger, Butter, Luchmann, Pitz, TP]

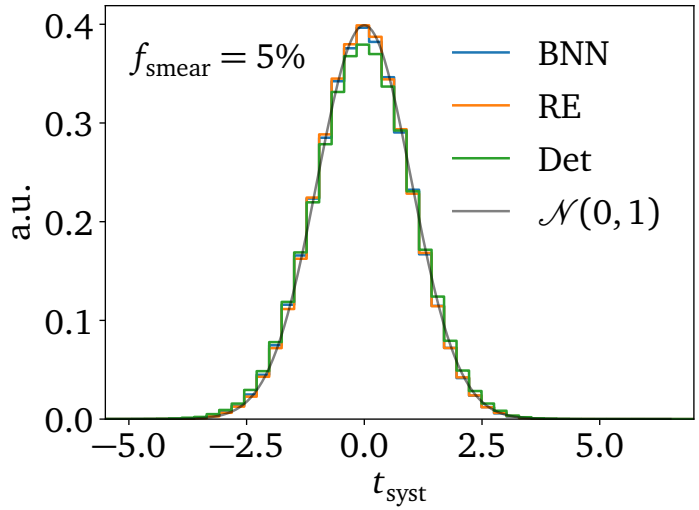
- loop amplitude $gg \to \gamma\gamma g(g)$ over
- systematics: artificial noise
- statistics plateau
- accuracy over phase space

$$\Delta(x) = \frac{A_{\text{NN}}(x) - A_{\text{true}}(x)}{A_{\text{true}}(x)}$$

- pull over phase space

$$t_{\text{syst}}(x) = \frac{A_{\text{NN}}(x) - A_{\text{true}}(x)}{\sigma_{\text{syst}}(x)}$$





Calibrated uncertainties

Amplitude regression [Badger, Butter, Luchmann, Pitz, TP]

- loop amplitude $gg \to \gamma\gamma g(g)$ over
- systematics: artificial noise
- statistics plateau
- accuracy over phase space

$$\Delta(x) = \frac{A_{\text{NN}}(x) - A_{\text{true}}(x)}{A_{\text{true}}(x)}$$

- pull over phase space

$$t_{\text{syst}}(x) = \frac{A_{\text{NN}}(x) - A_{\text{true}}(x)}{\sigma_{\text{syst}}(x)}$$

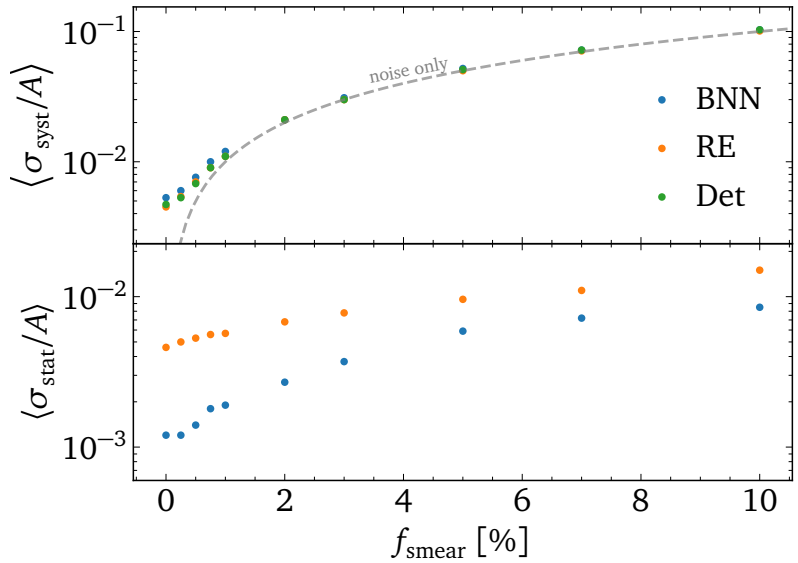
Towards zero noise

- scaling

$$\sigma^2_{\text{syst}} - \sigma^2_{\text{syst,0}} \approx \sigma^2_{\text{train}}$$

- plateau $\langle \sigma_{\text{syst}}/A \rangle \sim 0.4\%$

$\to$ Limiting factor??



Calibrated uncertainties

Amplitude regression [Badger, Butter, Luchmann, Pitz, TP]

- loop amplitude $gg \to \gamma\gamma g(g)$ over
- systematics: artificial noise
- statistics plateau
- accuracy over phase space

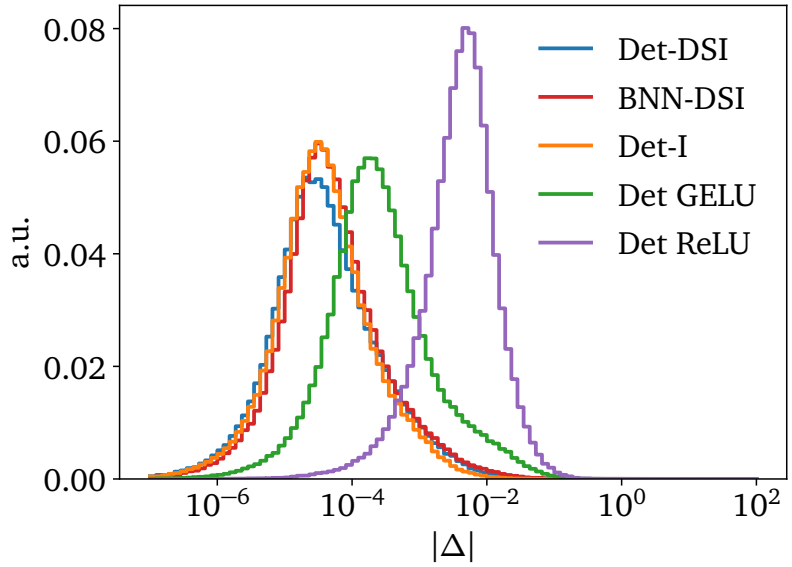
$$\Delta(x) = \frac{A_{\text{NN}}(x) - A_{\text{true}}(x)}{A_{\text{true}}(x)}$$

- pull over phase space

$$t_{\text{syst}}(x) = \frac{A_{\text{NN}}(x) - A_{\text{true}}(x)}{\sigma_{\text{syst}}(x)}$$

Data pre-processing

- amplitude from invariants
- learn Minkowski metric
- Deep-sets-invariant network
  L-GATr transformer the same



Calibrated uncertainties

Amplitude regression [Badger, Butter, Luchmann, Pitz, TP]

- loop amplitude $gg \to \gamma\gamma g(g)$ over
- systematics: artificial noise
- statistics plateau
- accuracy over phase space

$$\Delta(x) = \frac{A_{\text{NN}}(x) - A_{\text{true}}(x)}{A_{\text{true}}(x)}$$

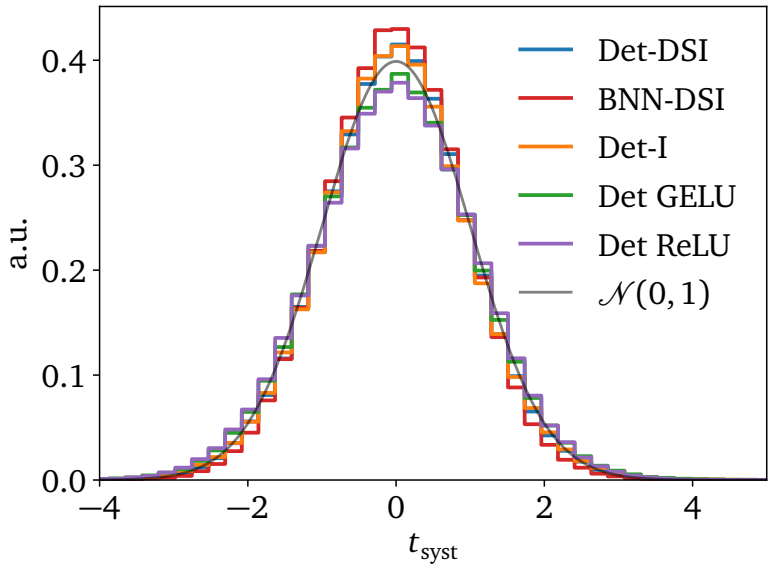
- pull over phase space

$$t_{\text{syst}}(x) = \frac{A_{\text{NN}}(x) - A_{\text{true}}(x)}{\sigma_{\text{syst}}(x)}$$

Data pre-processing

- amplitude from invariants
- learn Minkowski metric
- Deep-sets-invariant network
  L-GATr transformer the same

$\to$ Calibrated precision



LHC representation: L-GATr

Encode known symmetries [Brehmer, Breso, de Haan, TP, Qu, Spinner, Thaler]

1- permutation invariance $\to$ graph or transformer
transformer $\equiv$ learned representation with scalar product

2- Lorentz co-variance/equivariance $[S, f_\theta](x) = 0$

- input: 4-vectors vs Mandelstams $\to$ scalars, vectors, etc?
- geometric product extending vector space

$$xy = \frac{\{x,y\}}{2} + \frac{[x,y]}{2} \qquad \text{with} \qquad \{\gamma^\mu, \gamma^\nu\} = 2g^{\mu\nu}$$

- metric reducing, bivector increasing grade

$$\gamma^\mu \gamma^\nu = \frac{\{\gamma^\mu, \gamma^\nu\}}{2} + \frac{[\gamma^\mu, \gamma^\nu]}{2} \equiv g^{\mu\nu} + \sigma^{\mu\nu}$$

- multivector of geometric algebra [just like supersymmetry in 90s]

$$x = x^S\, 1 + x^V_\mu\, \gamma^\mu + x^B_{\mu\nu}\, \sigma^{\mu\nu} + x^A_\mu\, \gamma^\mu\gamma^5 + x^P\, \gamma^5 \qquad \text{with} \qquad \begin{pmatrix} x^S \\ x^V_\mu \\ x^B_{\mu\nu} \\ x^A_\mu \\ x^P \end{pmatrix} \in \mathbb{R}^{16}$$

- example input (PID, p_μ)

$$x^S = \text{PID} \qquad x^V_\mu = p_\mu \qquad x^T_{\mu\nu} = x^A_\mu = x^P = 0$$

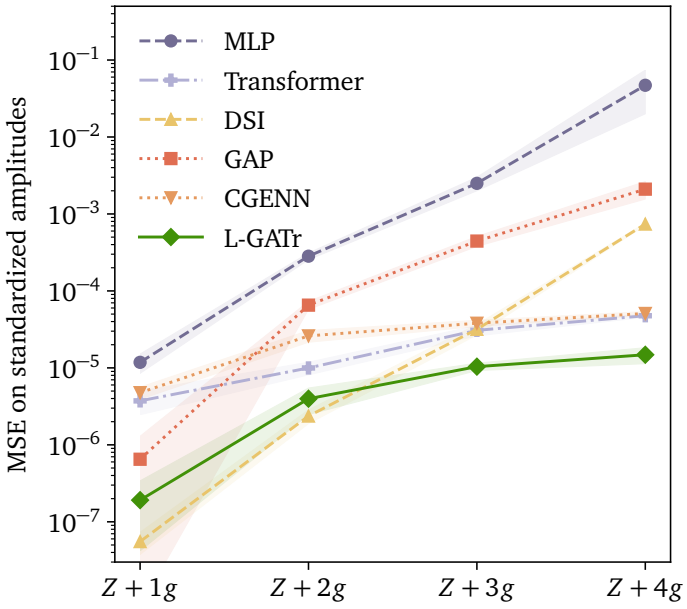
$\to$ Organize network layers by grade: L-GATr

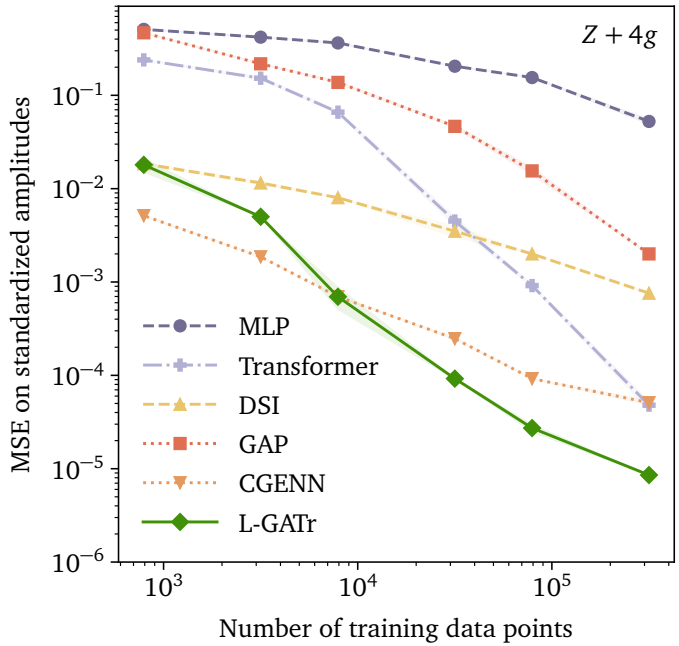
L-GATr amplitudes

Performance is all you need

- LO transition amplitudes $q\bar{q} \to Z + 1...4\ g$
- amplitude regression: cost scaling with dimensionality
  $\to$ low-dimensional representation better
- performance scaling with multiplicity and training data
  $\to$ transformers

$\to$ Equivariant transformers current winner



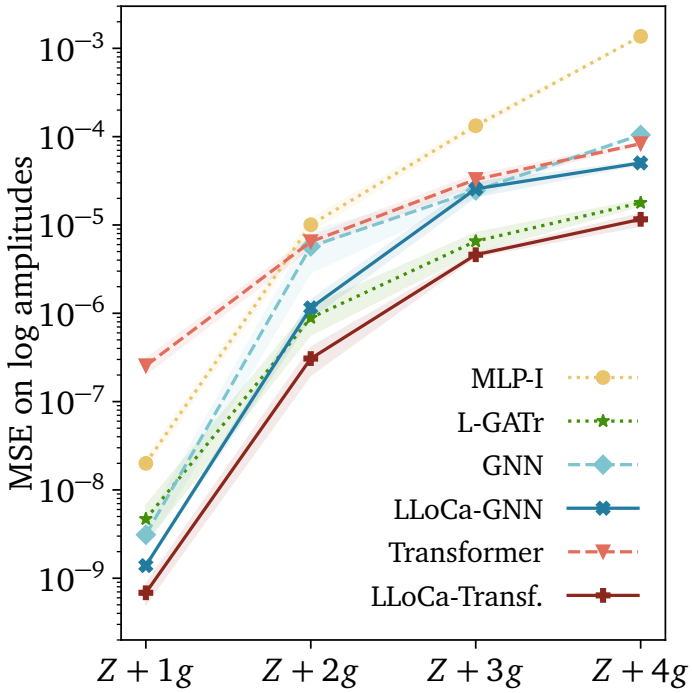


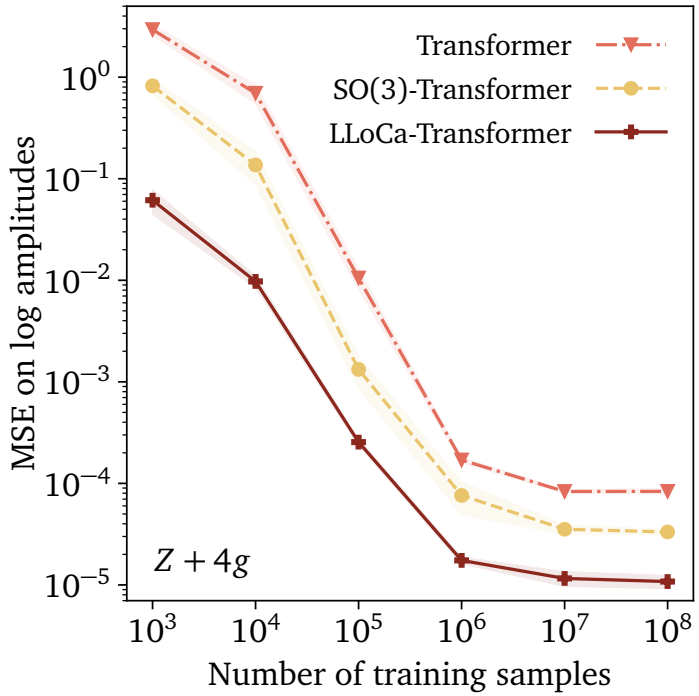
LLoCa vs L-GATr

Alternative implementation [Favaro, TP, Spinner + Hamprecht group]

- Better performance is all you need
- local Lorentz frame for each particle
  message passing between particles in local frames
- scaling with multiplicity and training data
  $\to$ L-GATs and LLoCa comparable
- intermediate symmetries means intermediate performance

$\to$ Advantage of equivariance for all implementations

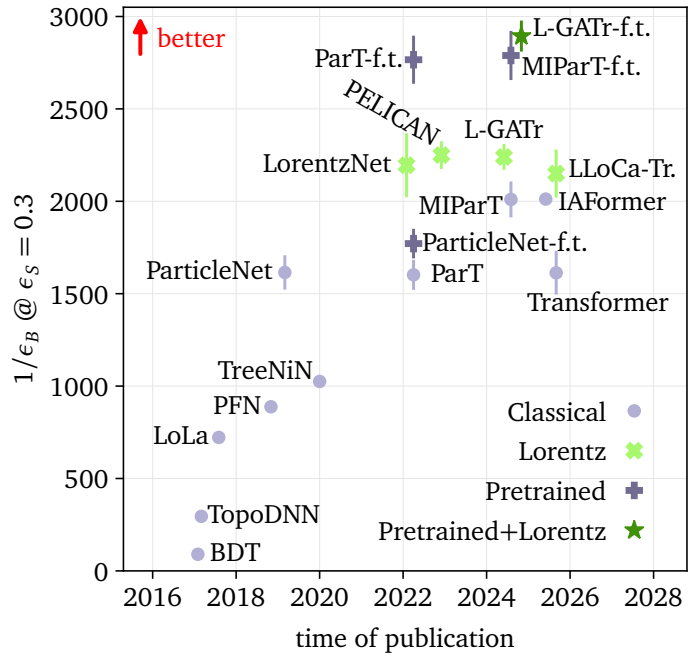




Equivariant jet tagging

Fat jet tagging benchmarks [same bottom line as flavor tagging]

- problem: jet axis breaking Lorentz-symmetry
  1- give jet axis explicitly as additional 4-vector
  2- reduce LLoCa symmetry
- top tagging performance
  $\to$ equivariance + pre-training leading
  $\to$ 30-fold background rejection from best BDT
  $\to$ only beaten by very large networks [Micuni, Nachman...]

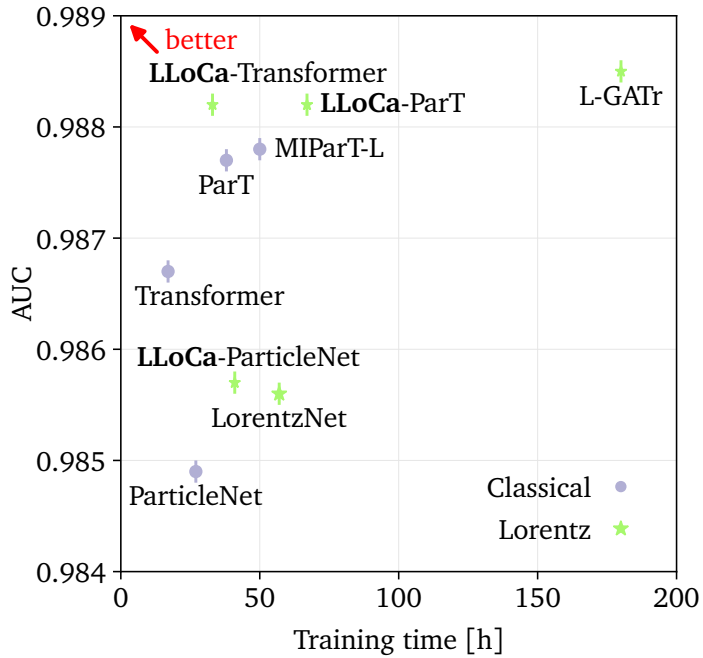


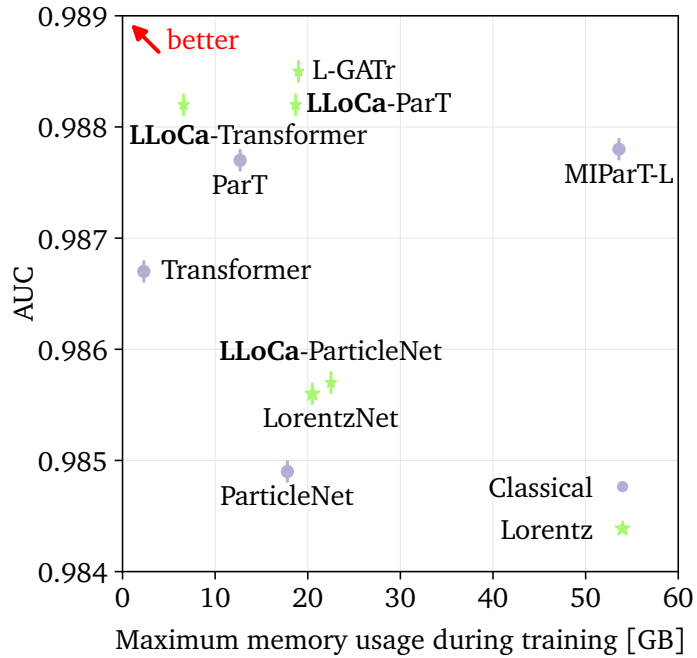
Equivariant jet tagging

Fat jet tagging benchmarks [same bottom line as flavor tagging]

- problem: jet axis breaking Lorentz-symmetry
  1- give jet axis explicitly as additional 4-vector
  2- reduce LLoCa symmetry
- top tagging performance
  → equivariance + pre-training leading
  → 30-fold background rejection from best BDT
  → only beaten by very large networks [Micuni, Nachman...]

→ Efficiency is what you also need...





Extrapolating ISR

Universal QCD jet radiation [$Z + 1 \ldots 8$ jets]

- from n to $n+1$ jets

$$R_{(n+1)/n} = \frac{\sigma_{n+1}}{\sigma_n} \quad \text{and} \quad P(n) = \frac{\sigma_n}{\sigma_{\text{tot}}} \quad \text{with} \quad \sigma_{\text{tot}} = \sum_{n=0}^{\infty} \sigma_n \; .$$

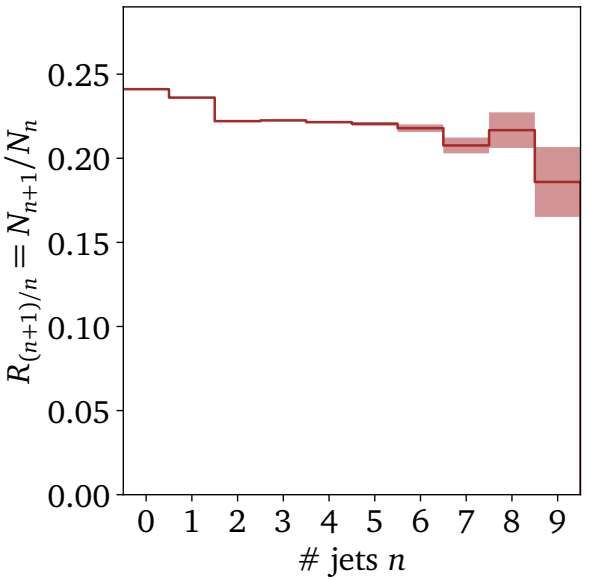
- large scale drop: Poisson scaling

$$R_{(n+1)/n} = \frac{\bar{n}}{n+1} \quad \Leftrightarrow \quad P(n) = \frac{\bar{n}^n e^{-\bar{n}}}{n!} \; .$$

- democratic scales: staircase scaling

$$R_{(n+1)/n} = 1 - \tilde{\Delta}_g(Q^2) \equiv P(n+1|n)$$

$\rightarrow$ Universal pattern learnable?



Extrapolating ISR

Universal QCD jet radiation [$Z + 1 \ldots 8$ jets]

- democratic scales: staircase scaling

$$R_{(n+1)/n} = 1 - \tilde{\Delta}_g(Q^2) \equiv P(n+1|n)$$

$\rightarrow$ Universal pattern learnable?

Autoregressive transformer [Butter, Charton, Villadamigo, Ore, TP, Spinner]

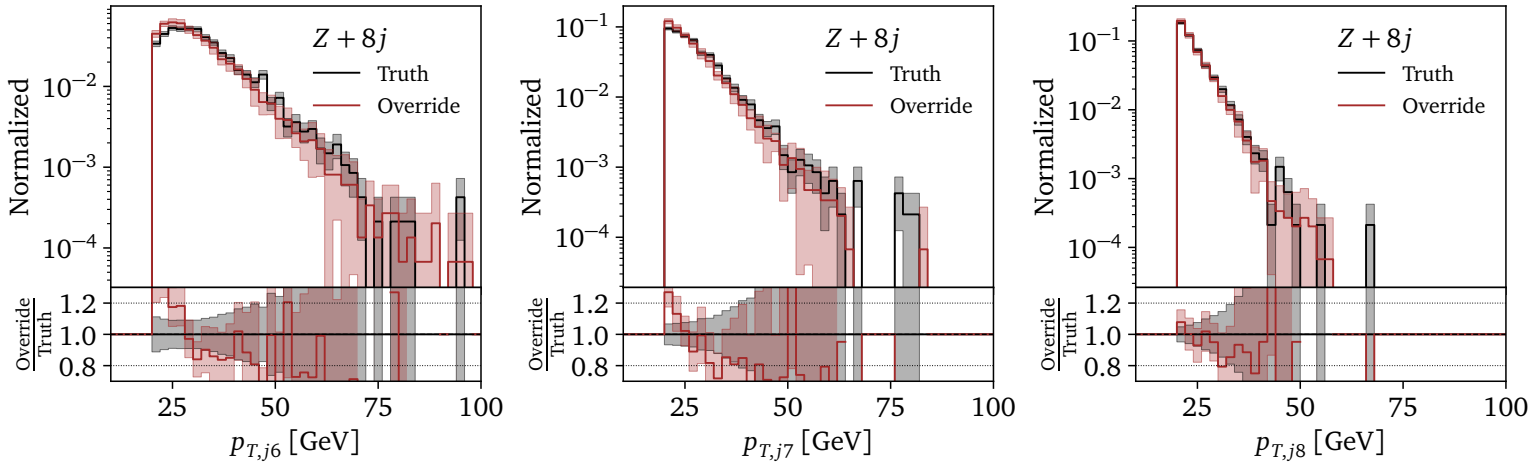
- factorized probability and loss function

$$p(x_i|x_{1:i-1}) = p_{\text{kin}}(x_i|x_{1:i-1})\; p_{\text{split}}(x_{1:i-1})$$

$$p(x_{1:n}) = \left[\prod_{i=1}^{n} p_{\text{kin}}(x_i|x_{1:i-1})\right] \left[\prod_{i=1}^{n} p_{\text{split}}(x_{1:i-1})\right] \left[1 - p_{\text{split}}(x_{1:n})\right] ,$$

- train up to 6 jets, generate 7 and 8

$\rightarrow$ full extrapolation with right latent representation



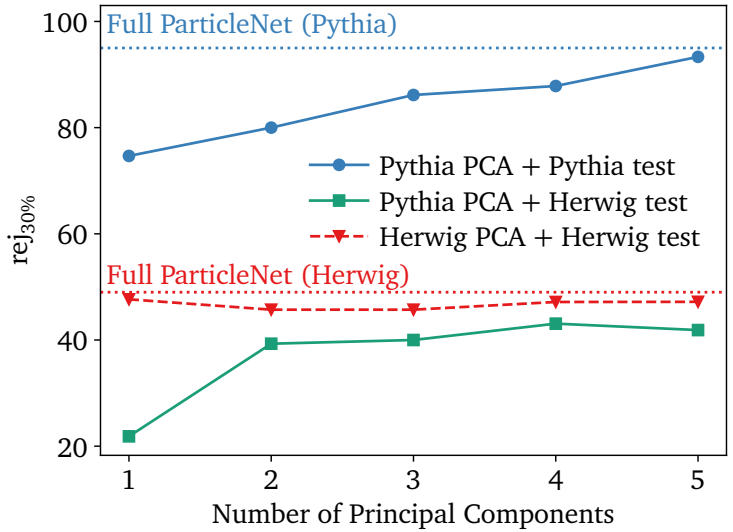
Latent physics

Quarks vs gluons from trained ParticleNet [Vent, Winterhalder, TP]

- sensitive substructure variables

$$n_{\text{pf}} = \sum_i 1 \quad w_{\text{pf}} = \frac{\sum_i p_{T,i} \Delta R_{i,\text{jet}}}{p_{T,\text{jet}}} \quad p_T D = \frac{\sqrt{\sum_i p_{T,i}^2}}{\sum_i p_{T,i}} \quad C_\beta = \frac{\sum_{i<j} p_{T,i} p_{T,j} (\Delta R_{ij})^\beta}{\left(\sum_i p_{T,i}\right)^2}$$

- related to max 5 principle components? [linear decorrelation]



Latent physics

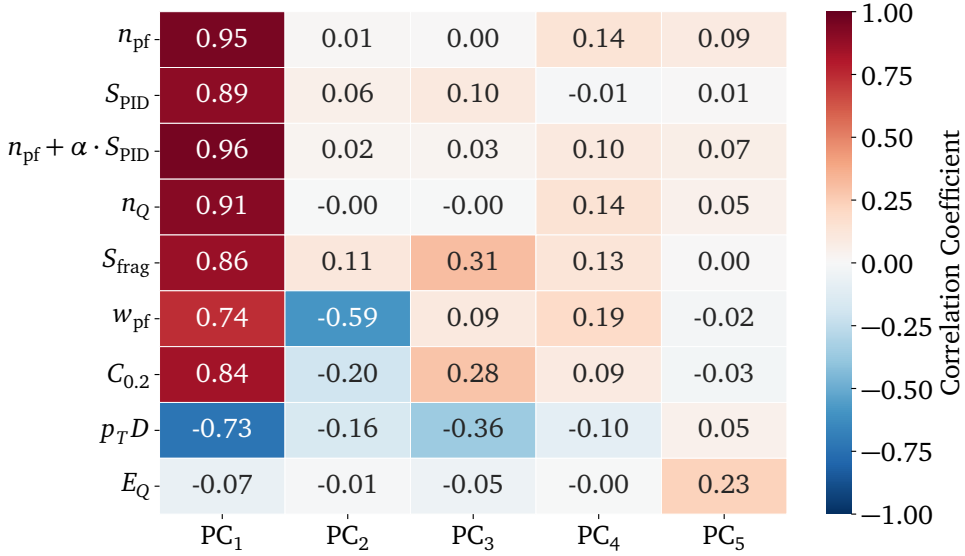
Quarks vs gluons from trained ParticleNet [Vent, Winterhalder, TP]

- sensitive substructure variables

$$n_{\text{pf}} = \sum_i 1 \quad w_{\text{pf}} = \frac{\sum_i p_{T,i} \Delta R_{i,\text{jet}}}{p_{T,\text{jet}}} \quad p_T D = \frac{\sqrt{\sum_i p_{T,i}^2}}{\sum_i p_{T,i}} \quad C_\beta = \frac{\sum_{i<j} p_{T,i} p_{T,j} (\Delta R_{ij})^\beta}{\left(\sum_i p_{T,i}\right)^2}$$

- related to max 5 principle components? [linear decorrelation]
- PC_1: constituent number and diversity

$$n_{\text{pf}} + \alpha \cdot S_{\text{PID}} \qquad \text{with} \qquad S_{\text{PID}} = -\sum_{\text{type}j} f_j \log f_j$$



Latent physics

Quarks vs gluons from trained ParticleNet [Vent, Winterhalder, TP]

- sensitive substructure variables

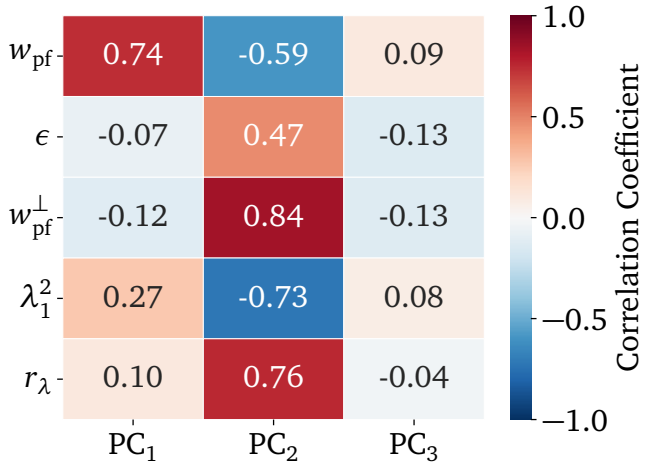
$$n_{\text{pf}} = \sum_i 1 \quad w_{\text{pf}} = \frac{\sum_i p_{T,i} \Delta R_{i,\text{jet}}}{p_{T,\text{jet}}} \quad p_T D = \frac{\sqrt{\sum_i p_{T,i}^2}}{\sum_i p_{T,i}} \quad C_\beta = \frac{\sum_{i<j} p_{T,i} p_{T,j} (\Delta R_{ij})^\beta}{\left(\sum_i p_{T,i}\right)^2}$$

- related to max 5 principle components? [linear decorrelation]
- PC_1: constituent number and diversity

$$n_{\text{pf}} + \alpha \cdot S_{\text{PID}} \quad \text{with} \quad S_{\text{PID}} = -\sum_{\text{type}j} f_j \log f_j$$

- PC_2: radial energy profile

$$w_{\text{pf}}^\perp = \alpha \cdot n_{\text{pf}} - w_{\text{pf}} \quad \text{and} \quad r_\lambda = \frac{\lambda_{0.5}^1}{\lambda_1^2} \quad \lambda_k^\beta = \sum_i z_i^\beta \Delta R^k$$



Latent physics

Quarks vs gluons from trained ParticleNet [Vent, Winterhalder, TP]

- sensitive substructure variables

$$n_{\text{pf}} = \sum_i 1 \quad w_{\text{pf}} = \frac{\sum_i p_{T,i} \Delta R_{i,\text{jet}}}{p_{T,\text{jet}}} \quad p_T D = \frac{\sqrt{\sum_i p_{T,i}^2}}{\sum_i p_{T,i}} \quad C_\beta = \frac{\sum_{i<j} p_{T,i} p_{T,j} (\Delta R_{ij})^\beta}{\left(\sum_i p_{T,i}\right)^2}$$

- related to max 5 principle components? [linear decorrelation]
- PC_1: constituent number and diversity

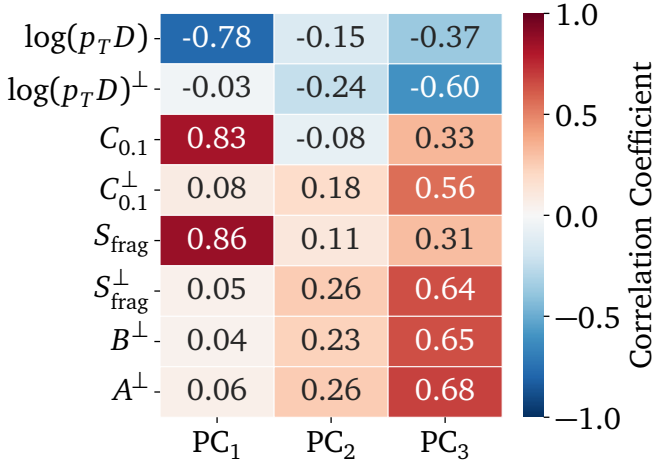
$$n_{\text{pf}} + \alpha \cdot S_{\text{PID}} \quad \text{with} \quad S_{\text{PID}} = -\sum_{\text{type}j} f_j \log f_j$$

- PC_2: radial energy profile

$$w_{\text{pf}}^\perp = \alpha \cdot n_{\text{pf}} - w_{\text{pf}} \quad \text{and} \quad r_\lambda = \frac{\lambda_{0.5}^1}{\lambda_1^2} \quad \lambda_k^\beta = \sum_i z_i^\beta \Delta R^k$$

- PC_3: fragmentation and energy dispersion

$$S_{\text{frag}} = -\sum_i z_i \log z_i$$



Latent physics

Quarks vs gluons from trained ParticleNet [Vent, Winterhalder, TP]

- sensitive substructure variables

$$n_{\text{pf}} = \sum_i 1 \quad w_{\text{pf}} = \frac{\sum_i p_{T,i} \Delta R_{i,\text{jet}}}{p_{T,\text{jet}}} \quad p_T D = \frac{\sqrt{\sum_i p_{T,i}^2}}{\sum_i p_{T,i}} \quad C_\beta = \frac{\sum_{i<j} p_{T,i} p_{T,j} (\Delta R_{ij})^\beta}{\left(\sum_i p_{T,i}\right)^2}$$

- related to max 5 principle components? [linear decorrelation]
- PC_1: constituent number and diversity

$$n_{\text{pf}} + \alpha \cdot S_{\text{PID}} \qquad \text{with} \qquad S_{\text{PID}} = -\sum_{\text{type}j} f_j \log f_j$$

- PC_2: radial energy profile

$$w_{\text{pf}}^\perp = \alpha \cdot n_{\text{pf}} - w_{\text{pf}} \qquad \text{and} \qquad r_\lambda = \frac{\lambda_{0.5}^1}{\lambda_1^2} \qquad \lambda_k^\beta = \sum_i z_i^\beta \Delta R^k$$

- PC_3: fragmentation and energy dispersion

$$S_{\text{frag}} = -\sum_i z_i \log z_i$$

- $\text{PC}_{4,5}$: charge information etc

$$E_Q = \frac{E_{\text{charged}}}{E_{\text{jet}}} \qquad \text{and} \qquad A^\perp = S_{\text{frag}} \frac{C_{0.1}}{C_{0.05}} - 0.03 \cdot n_{\text{pf}} + 1.95 w_{\text{pf}}^\perp$$

$\rightarrow$ Latent distributions learn physics

ParticleNet beyond PCA

Disentangled latent classifier

- learning compressed, decorrelated representation

$$\mathcal{L} = \underbrace{\sum_{i=1}^{N} |x_i - \hat{x}_i|^2}_{\mathcal{L}_\text{reco}} + \underbrace{\sum_{i=1}^{N} \Big[y_i \log \sigma(z_i) + (1-y_i) \log(1-\sigma(z_i)) \Big]}_{\mathcal{L}_\text{class}} + \underbrace{\sum_{j \neq k} \Big[\text{Cov}(z_j, z_k) \Big]^2}_{\mathcal{L}_\text{disentangle}}$$

$\rightarrow$ 5 latent dimensions plenty

Latent Dim	1	2	3	4
AUC	0.893(2)	0.9001(4)	0.9024(4)	0.9034(2)
$\text{rej}_{30\%}$	72(3)	77(3)	´ 95(5)	95(3)
ΔC	1.8(3)	0.93(5)	1.0(16)	0.9(15)

ParticleNet beyond PCA

Disentangled latent classifier

- learning compressed, decorrelated representation

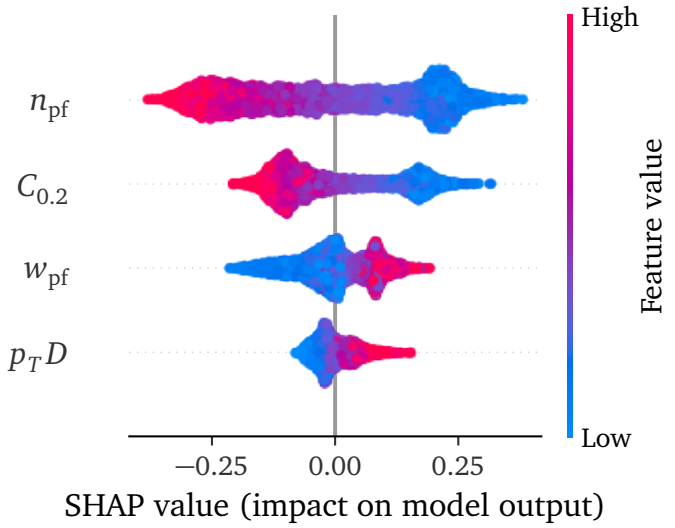
$$\mathcal{L} = \underbrace{\sum_{i=1}^{N} |x_i - \hat{x}_i|^2}_{\mathcal{L}_\text{reco}} + \underbrace{\sum_{i=1}^{N} \Big[y_i \log \sigma(z_i) + (1-y_i) \log(1-\sigma(z_i)) \Big]}_{\mathcal{L}_\text{class}} + \underbrace{\sum_{j \neq k} \Big[\text{Cov}(z_j, z_k) \Big]^2}_{\mathcal{L}_\text{disentangle}}$$

$\rightarrow$ 5 latent dimensions plenty

Shapley variables

- pretty pictures, with weird patterns [quark jets: low multiplicity and low girth]

$\rightarrow$ Little insight from SHAP implementation



ParticleNet beyond PCA

Disentangled latent classifier

- learning compressed, decorrelated representation

$$\mathcal{L} = \underbrace{\sum_{i=1}^{N} |x_i - \hat{x}_i|^2}_{\mathcal{L}_\text{reco}} + \underbrace{\sum_{i=1}^{N} \Big[y_i \log \sigma(z_i) + (1-y_i) \log(1-\sigma(z_i)) \Big]}_{\mathcal{L}_\text{class}} + \underbrace{\sum_{j \neq k} \Big[\text{Cov}(z_j, z_k) \Big]^2}_{\mathcal{L}_\text{disentangle}}$$

$\rightarrow$ 5 latent dimensions plenty

Shapley variables

- pretty pictures, with weird patterns [quark jets: low multiplicity and low girth]

$\rightarrow$ Little insight from SHAP implementation

Symbolic Regression

- learn formula with given complexity
- classifier output not power series
- AUC and calibration dual goal

observables	model	AUC	$\text{Rej}_{30\%}$
$(n_\text{pf}, p_T D, C_{0.2}, r_\lambda, S_\text{PID}, S_\text{frag}, E_Q)$	MLP	0.872	66.87
	PySR	0.871	66.58

$$p_\text{quark} = \tanh^3 \Big[0.55 \cdot C_{0.2} + 2 \left(-0.02 \cdot r_\lambda \cdot \left(C_{0.2} \cdot p_T D \cdot S_\text{PID} \cdot S_\text{frag} - 0.25 \right) + 1 \right)^3 \Big]$$

$\rightarrow$ Formulas as physics regularizers? [Bahl, Fuchs, Menem, TP]

ML for LHC Theory

ML is particle physics method development

1 just another numerical tool for a numerical field

2 completely transformative new language

- driven by (money from) industry and medical research
- particle physics should be leading scientific AI
- improve established tools
  develop new tools for established tasks
  transform through new ideas

→ Complexity our friend

→ Representations the key

Modern Machine Learning for LHC Physicists

Tilman Plehn[a,b*], Anja Butter[a,c], Barry Dillon[a,d],
Theo Heimel[a,e], Claudius Krause[a,f], and Ramon Winterhalder[a,g]

[a] Institute for Theoretical Physics, Heidelberg University, Germany
[b] Interdisciplinary Center for Scientific Computing (IWR), Heidelberg University, Germany
[c] LPNHE, Sorbonne Université, Université Paris Cité, CNRS/IN2P3, Paris, France
[d] Intelligent Systems Research Centre, Ulster University, Northern Ireland
[e] CP3, Université catholique de Louvain, Louvain-la-Neuve, Belgium
[f] Marietta Blau Institute for Particle Physics, Austrian Academy of Sciences, Vienna, Austria
[g] TIFLab, Universitá degli Studi di Milano & INFN Sezione di Milano, Italy

November 10, 2025

Abstract

Depending on the point of view, modern machine learning is either providing an unprecedented boost to the numerical methods of particle physics, or it is transforming the way we learn fundamental physics from vast amounts of complex data. These lecture notes lead students with basic knowledge of particle physics and significant enthusiasm for machine learning to relevant applications. The notes start with an LHC-specific motivation and a physics-driven introduction to neural networks. As applications they cover classification, unsupervised classification, generative networks, and inverse problems. The applications lead to an underlying theme, representation learning uniting the methodological aspects of statistically defined training, accuracy, and precision. All examples are chosen from particle physics publications of the last few years, and many of them come with corresponding tutorials. [1]

ATLAS calibration

Energy calibration with uncertainties [ATLAS + Heimel, TP, Vogel]

- interpretable calorimeter phase space x
- learned calibration function

$$\mathcal{R}_{\text{NN}}(x) \pm \Delta\mathcal{R}_{\text{NN}}(x) \approx \frac{E^{\text{obs}}(x)}{E^{\text{dep}}(x)}$$

- trained on simulations, statistics neglibigle
- systematics: noise in data
  network expressivity
  data representation ...

ATLAS calibration

Energy calibration with uncertainties [ATLAS + Heimel, TP, Vogel]

- interpretable calorimeter phase space x
- learned calibration function

$$\mathcal{R}_{\text{NN}}(x) \pm \Delta\mathcal{R}_{\text{NN}}(x) \approx \frac{E^{\text{obs}}(x)}{E^{\text{dep}}(x)}$$

- trained on simulations, statistics neglibigle
- systematics: noise in data
  network expressivity
  data representation ...

$\rightarrow$ Understand (simulated) detector