Representation Learning for the LHC

Tilman Plehn

Universität Heidelberg

CMS-D, Hamburg, September 2025

‘Theory Talk’

Personal reasoning

- BSM physics no longer exciting
- Higgs physics not attractive
- precision-QCD too hard

- precision-LHC great success
- LHC theory all numerics
- conceptual/technical progress?

- AI research taking a very wrong turn
- scientific AI without fundamental physics??

→ Scientific AI for LHC

ML as representation learning

Similar to fit

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

→ θ data representation

Phase space probabilities

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

Requirements on θ

- accuracy
- precision
- structure [physics?]

→ Physics knowledge or knowledge-free upscaling?

Lorentz-equivariance

Encode known symmetries

- permutation co-variance → graph or transformer
- Lorentz co-variance $\to \Lambda(f_\theta(x)) = f_\theta(\Lambda(x))$ [4-vectors vs Mandelstams]

1- L-GATr geometric algebra representation

2- LLoCa local reference frame per particle

Lorentz-equivariance

Encode known symmetries

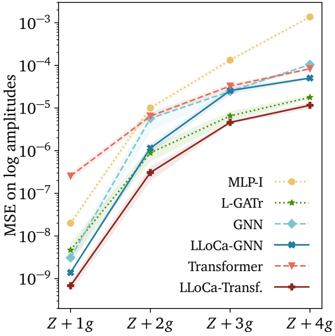
- permutation co-variance → graph or transformer
- Lorentz co-variance $\to \Lambda(f_\theta(x)) = f_\theta(\Lambda(x))$ [4-vectors vs Mandelstams]

1- L-GATr geometric algebra representation

2- LLoCa local reference frame per particle

Performance is all you need

- LO transition amplitudes $q\bar{q} \to Z + 1...4\ g$
- improved scaling with multiplicity



Lorentz-equivariance

Encode known symmetries

- permutation co-variance → graph or transformer
- Lorentz co-variance $\to \Lambda(f_\theta(x)) = f_\theta(\Lambda(x))$ [4-vectors vs Mandelstams]

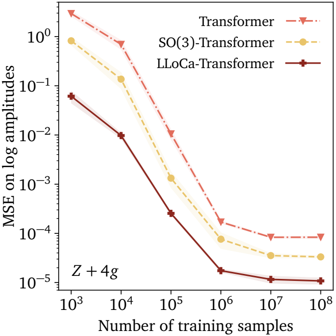
1- L-GATr geometric algebra representation

2- LLoCa local reference frame per particle

Performance is all you need

- LO transition amplitudes $q\bar{q} \to Z + 1...4\ g$
- improved scaling with multiplicity
- subset of symmetries

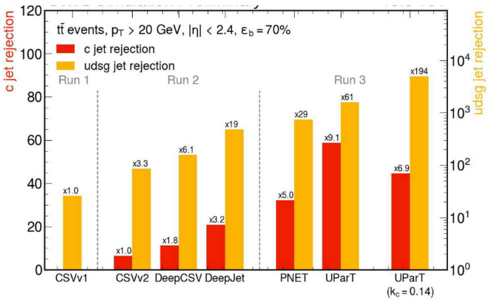
→ Advantage across implementations



Equivariant jet tagging

Tagging benchmarks

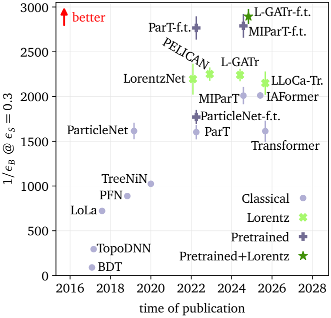
- flavor tagging and top tagging
  → more than 10x improvement over Gregor's BDT

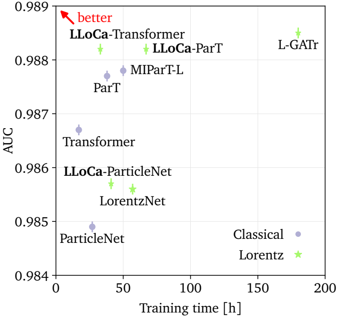


Equivariant jet tagging

Tagging benchmarks

- flavor tagging and top tagging
  → more than 10x improvement over Gregor's BDT
- problem: jet axis breaking Lorentz-symmetry
  1- give jet axis explicitly as additional 4-vector
  2- reduce encoded symmetry [LLoCa]





Equivariant jet tagging

Tagging benchmarks

- flavor tagging and top tagging
  → more than 10x improvement over Gregor’s BDT
- problem: jet axis breaking Lorentz-symmetry
  1- give jet axis explicitly as additional 4-vector
  2- reduce encoded symmetry [LLoCa]
- multi-class tagging the same

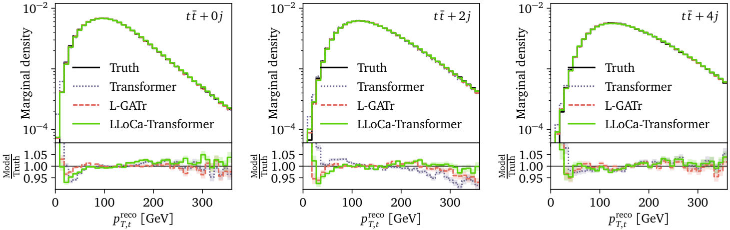
→ Resiliance, uncertainties, experiment next

	$H \to b\bar{b}$	$H \to c\bar{c}$	$H \to gg$	$H \to 4q$	$H \to l\nu q\bar{q}'$	$t \to bq\bar{q}'$	$t \to bl\nu$	$W \to q\bar{q}'$	$Z \to q\bar{q}$
	Rej$_{50\%}$	Rej$_{50\%}$	Rej$_{50\%}$	Rej$_{50\%}$	Rej$_{99\%}$	Rej$_{50\%}$	Rej$_{99.5\%}$	Rej$_{50\%}$	Rej$_{50\%}$
PFN [72]	2924	841	75	198	265	797	721	189	159
P-CNN [73]	4890	1276	88	474	947	2907	2304	241	204
MIParT-L [68]	10753	4202	123	1927	5450	31250	16807	542	402
LorentzNet [22]	8475	2729	111	1152	3515	13889	10257	400	303
L-GATr [26]	12987	4819	128	2311	6116	47619	20408	588	432
ParticleNet [18]	7634	2475	104	954	3339	10526	11173	347	283
LLoCa-ParticleNet*	7463	2833	105	1072	3155	10753	9302	403	306
ParT [19]	10638	4149	123	1864	5479	32787	15873	543	402
LLoCa-ParT*	11561	4640	125	2037	5900	41667	19231	552	419
Transformer	10753	3333	116	1369	4630	24390	17857	415	334
LLoCa-Transformer*	11628	4651	125	2037	5618	39216	17241	548	410

Equivariant event generation

MadGraph7: ML-event generation

- conditional flow matching with transformer-velocity
- end-to-end $pp \to t_\text{had} \bar{t}_\text{had} + 0...4j$ [22M training events]
- per-cent kinematics

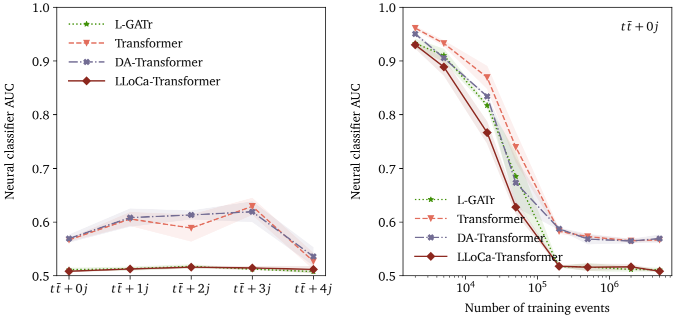


Equivariant event generation

MadGraph7: ML-event generation

- conditional flow matching with transformer-velocity
- end-to-end $pp \to t_\text{had} \bar{t}_\text{had} + 0...4j$ [22M training events]
- per-cent kinematics
- LHC: training-generation classifier

→ Accurate phase space a solved problem...



Representing mean and uncertainty

Remember a fit

- learn scalar field $f_\theta(x) \approx f(x)$
- statistics: maximize parameter probability given (f_j, σ_j)

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

$\rightarrow$ maximize likelihood instead

$$p(x|\theta) = \prod_j \frac{1}{\sqrt{2\pi}\sigma_j} \exp\left(- \frac{|f_j - f_\theta(x_j)|^2}{2\sigma_j^2} \right)$$

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

Representing mean and uncertainty

Remember a fit

- learn scalar field $f_\theta(x) \approx f(x)$
- statistics: maximize parameter probability given (f_j, σ_j)

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

→ maximize likelihood instead

$$p(x|\theta) = \prod_j \frac{1}{\sqrt{2\pi}\sigma_j} \exp\left(- \frac{|f_j - f_\theta(x_j)|^2}{2\sigma_j^2} \right)$$

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

Learned local uncertainty

- Gaussian log-likelihood with normalization

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

- if needed replace $\sigma_\theta(x)$ by mixture model

→ learn $f_\theta(x)$ and $\sigma_\theta(x)$ together

1- Bayesian networks

Learned function statistically

- amplitude over phase phase

$$\langle A \rangle = \int dA \; A \; p(A)$$

- internal representation θ of training data T [think Gaussian with mean and width]

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

→ θ-distribution defining Bayesian NN

Variational approximation

- definition of training

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

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

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

→ Two-term loss: likelihood + prior

1- Bayesian networks

Learned function statistically

- amplitude over phase phase

$$\langle A \rangle = \int dA \; A \; p(A)$$

- internal representation θ of training data T [think Gaussian with mean and width]

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

→ θ-distribution defining Bayesian NN

Two uncertainties

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

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

- systematic — vanishing for perfect data: $p(A|\theta) \to \delta(A - A_0)$

$$\sigma^2_\text{syst} = \int d\theta \; q(\theta) \; \left[\overline{A^2}(\theta) - \overline{A}(\theta)^2 \right]$$

→ Systematics dominant for LHC

2- Repulsive ensembles

Posterior from network ensemble

- OED vs continuity equation

$$\frac{d\theta}{dt} = v(\theta, t) \qquad \Leftrightarrow \qquad \frac{\partial \rho(\theta, t)}{\partial t} = -\nabla_\theta \left[v(\theta, t) \rho(\theta, t) \right]$$

- Fokker-Planck equation with stationary $\rho(\theta, t) = \pi(\theta)$

$$\frac{d\theta}{dt} = -\nabla_\theta \log \frac{\rho(\theta, t)}{\pi(\theta)}$$

- ODE describing training progress

$$\begin{aligned} \theta^{t+1} - \theta^t &\propto -\nabla_{\theta^t} \left[\log \rho(\theta^t) - \log \pi(\theta^t) \right] \\ &= -\nabla_{\theta^t} \left[\log \sum_j k(\theta^t, \theta^t_j) - \log p(\theta | x^t_\text{train}) \right] \equiv -\nabla_{\theta^t} \mathcal{L}_\text{RE} \end{aligned}$$

→ Joint ensemble training

Repulsive ensembles

- train network ensemble
- apply repulsive force
  kernel in function space

→ Alternative for statistical uncertainty

3- Evidential regression

Uncertainties from latent distribution, without sampling [Bahl, Elmer, TP, Winterhalder]

- evidential distribution

$$p(A) = \int d\lambda \; p(A|\lambda) \; p(\lambda|T) \approx \int d\lambda \; p(A|\lambda) \; p(\lambda|\theta_0)$$

assuming $p(A|\lambda) = \mathcal{N}\left(A| \, \overline{A}, \sigma^2\right)$ with $\lambda \equiv (\overline{A}, \sigma^2)$

- choose $p(\lambda|\theta_0)$ as conjugate prior

$$p(\lambda|\theta_0) = \frac{\beta^\alpha \sqrt{v}}{\Gamma(\alpha)\sqrt{2\pi\sigma^2}} \; \left(\frac{1}{\sigma^2}\right)^{\alpha+1} \exp\left(-\frac{2\beta + v(\gamma - \overline{A})^2}{2\sigma^2}\right)$$

with $\{\gamma, v, \alpha, \beta\} \; (x, \theta_0)$

- analytic likelihood: Student-t

$$p(A) = \text{St}\left(A\middle|\gamma, \frac{\beta(1+v)}{v\alpha}, 2\alpha\right)$$

$$A_\text{NN} = \int d\lambda \; \overline{A} \; p(\lambda|\theta_0) = \gamma$$

$$\sigma^2_\text{syst} = \int d\lambda \; \sigma^2 \; p(\lambda|\theta_0) = \frac{\beta}{\alpha - 1}$$

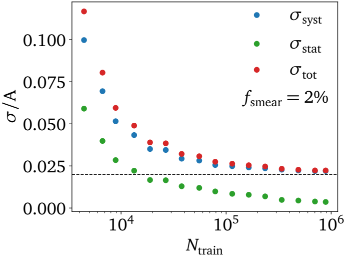
$$\sigma^2_\text{stat} = \int d\lambda \; \left[\overline{A} - A_\text{NN}\right]^2 \; p(\lambda|\theta_0) = \frac{\beta}{v(\alpha - 1)}$$

→ Another alternative for learned uncertainty

Amplitudes with calibrated uncertainties

Loop amplitude $gg \to \gamma\gamma g(g)$ [Bahl, Elmer, Favaro, Haußmann, TP, Winterhalder]

- systematics: artificial noise
- statistics plateau

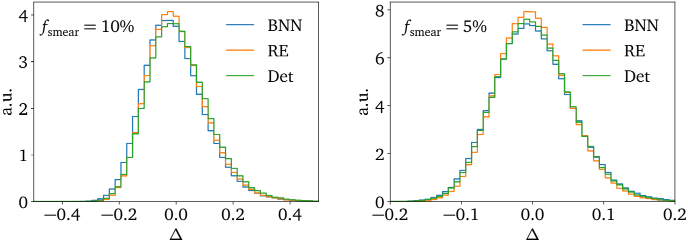


Amplitudes with calibrated uncertainties

Loop amplitude $gg \to \gamma\gamma g(g)$ [Bahl, Elmer, Favaro, Haußmann, TP, Winterhalder]

- systematics: artificial noise
- statistics plateau
- accuracy over phase space

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



Amplitudes with calibrated uncertainties

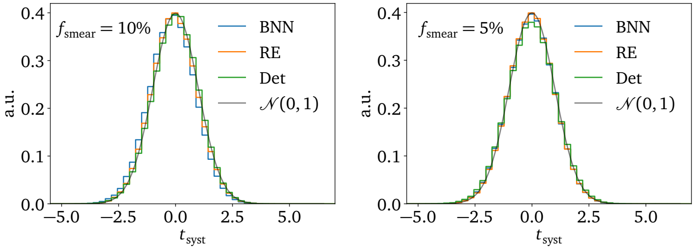
Loop amplitude $gg \to \gamma\gamma g(g)$ [Bahl, Elmer, Favaro, Haußmann, TP, Winterhalder]

- 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

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



Amplitudes with calibrated uncertainties

Loop amplitude $gg \to \gamma\gamma g(g)$

[Bahl, Elmer, Favaro, Haußmann, TP, Winterhalder]

- 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

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

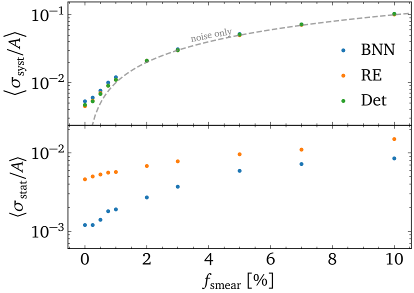
Towards zero noise

- scaling

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

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

→ Limiting factor??



Amplitudes with calibrated uncertainties

Loop amplitude $gg \to \gamma\gamma g(g)$ [Bahl, Elmer, Favaro, Haußmann, TP, Winterhalder]

- systematics: artificial noise
- statistics plateau
- accuracy over phase space

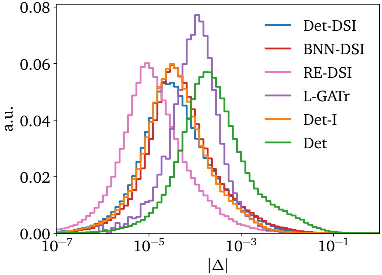
$$\Delta(x) = \frac{A_\text{NN}(x) - A_\text{true}(x)}{A_\text{true}(x)}$$

- pull over phase space

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

Data representation

- amplitude from invariants
- learn Minkowski metric
- Deep-sets-invariant network



Amplitudes with calibrated uncertainties

Loop amplitude $gg \to \gamma\gamma g(g)$ [Bahl, Elmer, Favaro, Haußmann, TP, Winterhalder]

- systematics: artificial noise
- statistics plateau
- accuracy over phase space

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

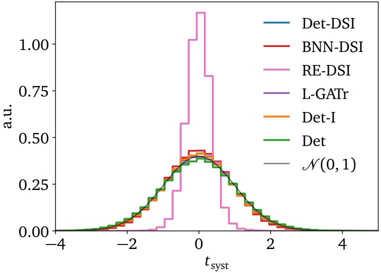
- pull over phase space

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

Data representation

- amplitude from invariants
- learn Minkowski metric
- Deep-sets-invariant network

→ Calibrated systematics



Phase space gaps

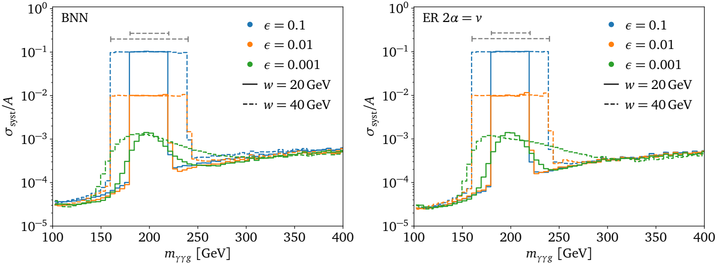
Noisy amplitude gap [Bahl, Elmer, TP, Winterhalder]

- step-function relative noise in $m_{\gamma\gamma g}$ gap

$$A_{\text{train}}(x) = \begin{cases} \mathcal{N}\left(A_{\text{true}}(x), \epsilon A_{\text{true}}(x)\right) & |m_{\gamma\gamma g}(x) - m_{\text{thresh}}| < w \\ A_{\text{true}} & |m_{\gamma\gamma g}(x) - m_{\text{thresh}}| \geq w \end{cases}$$

- compare Bayesian NN, ensembles, evidential regression

→ Only little noise corrected, any noise learned



GANplification

Amplification from generative networks [Bahl, Diefenbacher, Elmer, TP, Spinner]

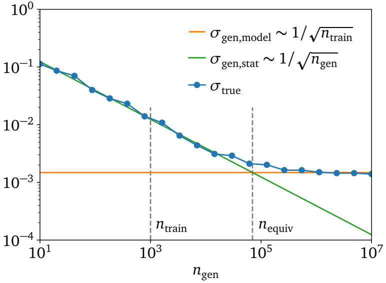
- phase space densities

$$p_\text{gen}(x) \approx p_\text{train}(x) \sim p_\text{true}(x) \qquad \Rightarrow \qquad p_\text{gen}(x) \overset{?}{\sim} p_\text{true}(x)$$

- 'How many events can I sample from a network trained on n events?'
  fewer, unless trained perfectly, $p_\text{gen}(x) \neq p_\text{train}(x)$
  more, generative network smoothen $p_\text{train}(x)$
- scaling of two uncertainties

$$\sigma^2_\text{gen,stat} + \sigma^2_\text{gen,model} = \begin{cases} \dfrac{a}{n_\text{gen}} & n_\text{gen} \ll n_\text{train} \\ \dfrac{b}{n_\text{train}} & n_\text{gen} \gg n_\text{train} \end{cases}$$

transition point $G \equiv \dfrac{n_\text{gen}}{n_\text{train}} = \dfrac{a}{b}$



GANplification

Amplification from generative networks [Bahl, Diefenbacher, Elmer, TP, Spinner]

- phase space densities

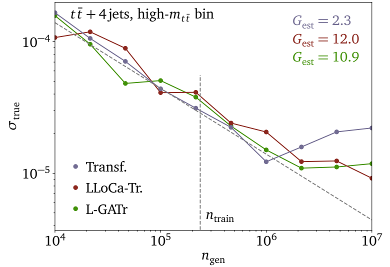
$$p_{\text{gen}}(x) \approx p_{\text{train}}(x) \sim p_{\text{true}}(x) \qquad \Rightarrow \qquad p_{\text{gen}}(x) \overset{?}{\sim} p_{\text{true}}(x)$$

- 'How many events can I sample from a network trained on n events?'
fewer, unless trained perfectly, $p_{\text{gen}}(x) \neq p_{\text{train}}(x)$
more, generative network smoothen $p_{\text{train}}(x)$
- scaling of two uncertainties

$$\sigma^2_{\text{gen,stat}} + \sigma^2_{\text{gen,model}} = \begin{cases} \dfrac{a}{n_{\text{gen}}} & n_{\text{gen}} \ll n_{\text{train}} \\ \dfrac{b}{n_{\text{train}}} & n_{\text{gen}} \gg n_{\text{train}} \end{cases} \qquad \text{transition point} \quad G \equiv \frac{n_{\text{gen}}}{n_{\text{train}}} = \frac{a}{b}$$

- averaging integral test
local KS-test
- gen-equivalent training size

→ L-GATr does amplify...



Physics from latent representation

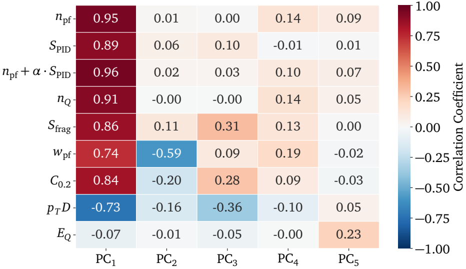
Quarks vs gluons from trained ParticleNet [Vent, Winterhalder, TP]

- sensitive substructure variables

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

- 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_1	PC_2	PC_3	PC_4	PC_5
n_pf	0.95	0.01	0.00	0.14	0.09
S_PID	0.89	0.06	0.10	-0.01	0.01
$n_\text{pf} + \alpha \cdot S_\text{PID}$	0.96	0.02	0.03	0.10	0.07
n_Q	0.91	-0.00	-0.00	0.14	0.05
S_frag	0.86	0.11	0.31	0.13	0.00
w_pf	0.74	-0.59	0.09	0.19	-0.02
$C_{0.2}$	0.84	-0.20	0.28	0.09	-0.03
$p_T D$	-0.73	-0.16	-0.36	-0.10	0.05
E_Q	-0.07	-0.01	-0.05	-0.00	0.23

Physics from latent representation

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

- sensitive substructure variables

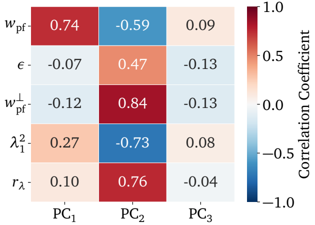
$$n_\text{pf} = \sum_i 1 \qquad w_\text{pf} = \frac{\sum_i p_{T,i} \Delta R_{i,\text{jet}}}{p_{T,\text{jet}}} \qquad p_T D = \frac{\sqrt{\sum_i p_{T,i}^2}}{\sum_i p_{T,i}} \qquad C_\beta = \frac{\sum_{i<j} p_{T,i} p_{T,j} (\Delta R_{ij})^\beta}{\left(\sum_i p_{T,i}\right)^2}$$

- 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$_1$	PC$_2$	PC$_3$
w_pf	0.74	-0.59	0.09
ϵ	-0.07	0.47	-0.13
$w_\text{pf}^\perp$	-0.12	0.84	-0.13
λ_1^2	0.27	-0.73	0.08
r_λ	0.10	0.76	-0.04

Physics from latent representation

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

- sensitive substructure variables

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

- PC_1: constituent number and diversity

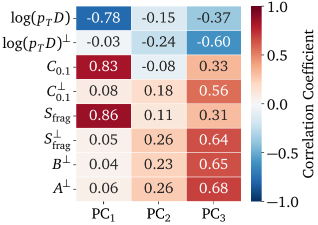
$$n_\text{pf} + \alpha \cdot S_\text{PID} \qquad \text{with} \qquad S_\text{PID} = - \sum_{\text{type} j} f_j \log f_j$$

- 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

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



	PC_1	PC_2	PC_3
$\log(p_T D)$	-0.78	-0.15	-0.37
$\log(p_T D)^\perp$	-0.03	-0.24	-0.60
$C_{0.1}$	0.83	-0.08	0.33
$C_{0.1}^\perp$	0.08	0.18	0.56
S_frag	0.86	0.11	0.31
$S_\text{frag}^\perp$	0.05	0.26	0.64
$B^\perp$	0.04	0.23	0.65
$A^\perp$	0.06	0.26	0.68

Physics from latent representation

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

- sensitive substructure variables

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

- 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

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

→ Latent distributions learn physics

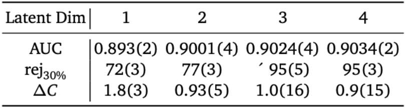
ParticleNet beyond PCA

Disentangled latent classifier

- learning compressed, decorrelated representation

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

→ 5 latent dimensions plenty



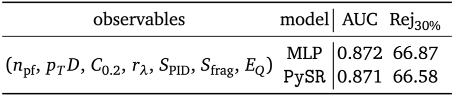
Latent Dim	1	2	3	4
AUC	0.893(2)	0.9001(4)	0.9024(4)	0.9034(2)
$\text{rej}_{30\%}$	72(3)	77(3)	´ 95(5)	95(3)
ΔC	1.8(3)	0.93(5)	1.0(16)	0.9(15)

Symbolic Regression

- learn formula with given complexity
- classifier output not power series

→ Formulas as physics regularizers? [Bahl, Fuchs, Menem, TP]

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



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 PySR	0.872 0.871	66.87 66.58

AI for fundamental physics

Develop AI for the best science

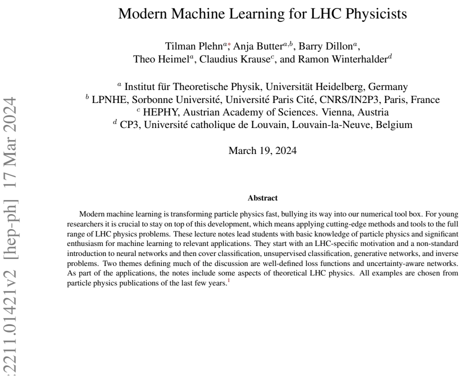
1 just another tool for a numerical field

2 transformative new language

- many applications in LHC theory

MadGraph7
MLhad
Higher orders
Simulation-based inference
SFitter global analyses
Unfolding
...

→ Make complexity our friend



Modern Machine Learning for LHC Physicists

Tilman Plehn[a,*], Anja Butter[a,b], Barry Dillon[a],
Theo Heimel[a], Claudius Krause[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] HEPHY, Austrian Academy of Sciences. Vienna, Austria
[d] CP3, Université catholique de Louvain, Louvain-la-Neuve, Belgium

March 19, 2024

Abstract

Modern machine learning is transforming particle physics fast, 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 lead students with basic knowledge of particle physics and significant enthusiasm for machine learning to relevant applications. 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 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.[1]

AI for fundamental physics

Develop AI for the best science

1 just another tool for a numerical field

2 transformative new language

· many applications in LHC theory

MadGraph7
MLhad
Higher orders
Simulation-based inference
SFitter global analyses
Unfolding
...

→ How long will we talk to actual people?

SciPost Physics **Submission**

Agents of Discovery

Sascha Diefenbacher[1], Anna Hallin[2],
Gregor Kasieczka[2], Michael Krämer[3], Anne Lauscher[4], Tim Lukas[2],

1 Physics Division, Lawrence Berkeley National Laboratory, Berkeley, USA
2 Institut für Experimentalphysik, Universität Hamburg, Germany
3 Institute for Theoretical Particle Physics and Cosmology, RWTH Aachen University, Germany
4 Data Science Group, Universität Hamburg, Germany

September 11, 2025

Abstract

The substantial data volumes encountered in modern particle physics and other domains of fundamental physics research allow (and require) the use of increasingly complex data analysis tools and workflows. While the use of machine learning (ML) tools for data analysis has recently proliferated, these tools are typically special-purpose algorithms that rely, for example, on encoded physics knowledge to reach optimal performance. In this work, we investigate a new and orthogonal direction: Using recent progress in large language models (LLMs) to create a team of *agents* — instances of LLMs with specific subtasks — that jointly solve data analysis-based research problems in a way similar to how a human researcher might: by creating code to operate standard tools and libraries (including ML systems) and by building on results of previous iterations. If successful, such agent-based systems could be deployed to automate routine analysis components to counteract the increasing complexity of modern tool chains. To investigate the capabilities of current-generation commercial LLMs, we consider the task of anomaly detection via the publicly available and highly-studied LHC Olympics dataset. Several current models by OpenAI (GPT-4o, o4-mini, GPT-4.1, and GPT-5) are investigated and their stability tested. Overall, we observe the capacity of the agent-based system to solve this data analysis problem. The best agent-created solutions mirror the performance of human state-of-the-art results.