ML for particle physics in the precision era

Henning Bahl

UNIVERSITÄT
HEIDELBERG
ZUKUNFT
SEIT 1386

HEP seminar, Universität Tübingen, 5.10.2025

The goal of particle physics

→ Answer the big fundamental questions!

Nature of EWSB

Neutrino masses

.....

Dark matter

Baryon asymmetry

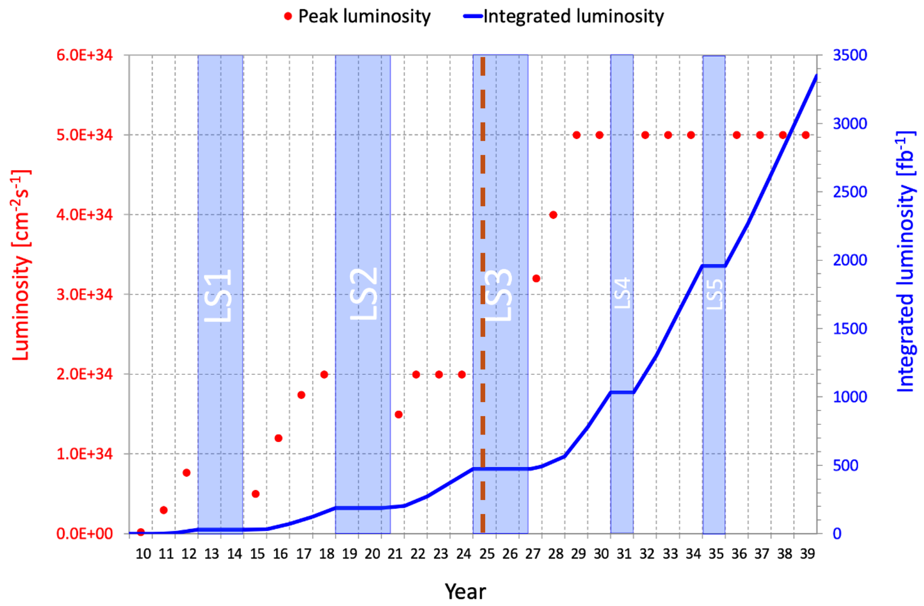
Naturalness

Can ML find answer these questions for us? **No!**

Can it help us with it? **Yes!**

The challenge ahead

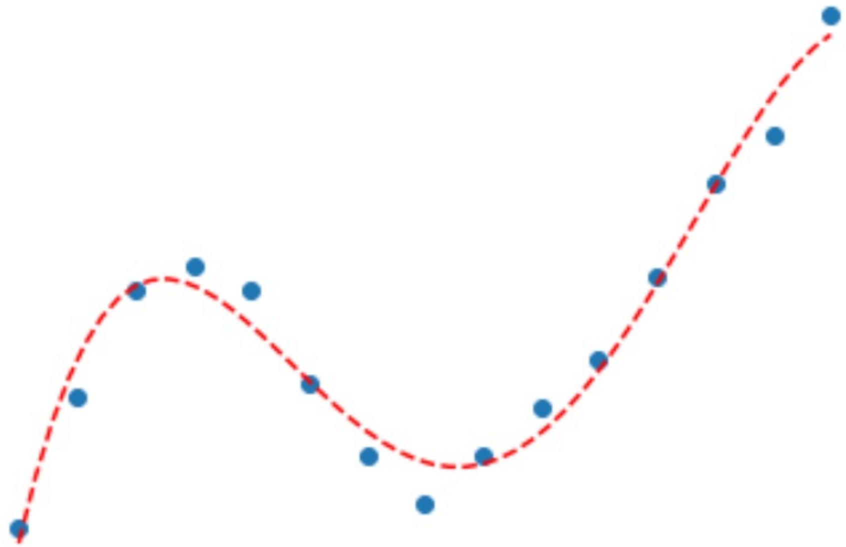
- general trend: larger-and-larger experiments collecting more-and-more data
- e.g. LHC: already enormous dataset will be further enlarged by a factor ~ 10
- costs for future experiments increasing



Fully exploit the available data!

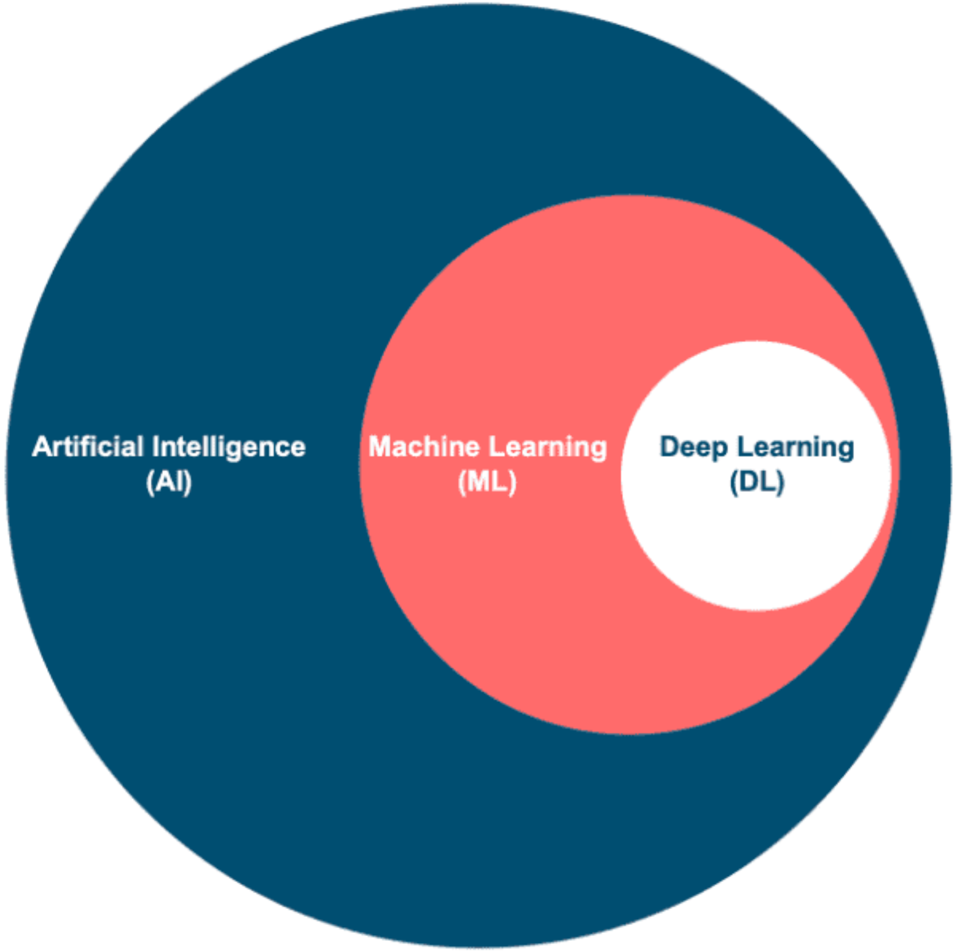
- new analysis methods
- theory precision $\simeq$ experimental precision
- in particular: high-precision MC simulation

ML in a nutshell

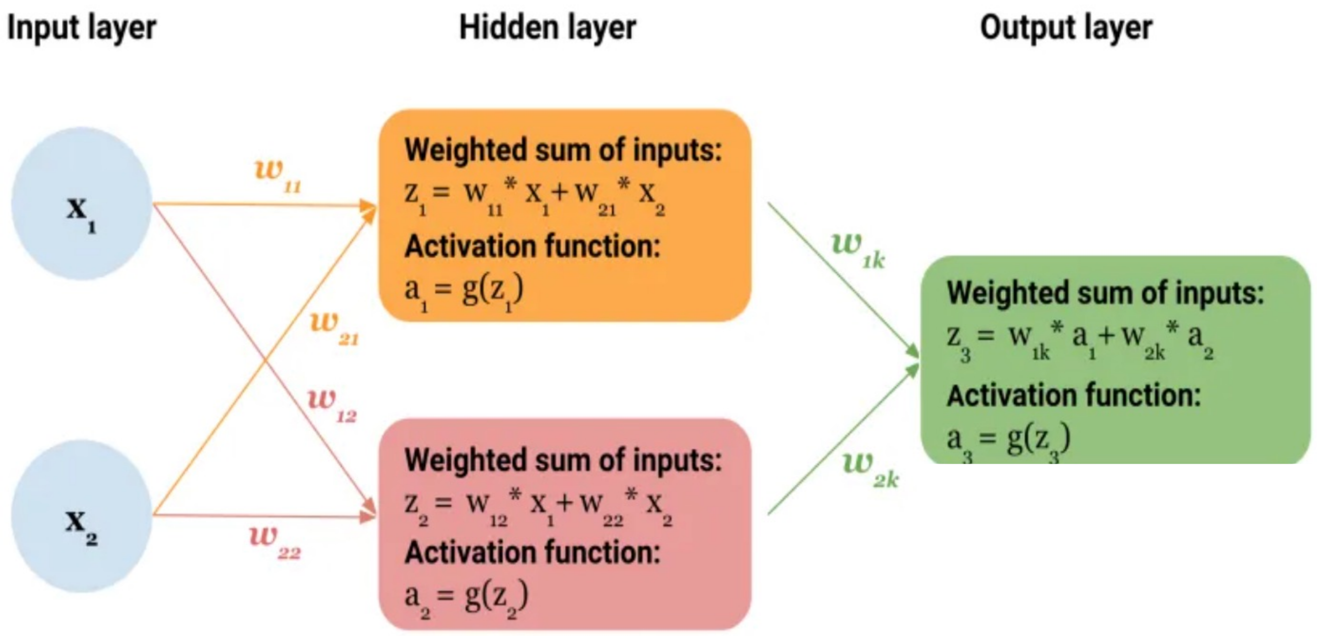


Terminology

- **Artificial Intelligence (AI)**
  - machines performing complex tasks
  - e.g. Feynman diagram generators, …
- **Machine Learning (ML)**
  - subfield of AI where machines learn from data
  - e.g. linear regression, BDTs, …
- **Deep Learning (DL)**
  - subfield of ML using deep neural networks



Neural networks

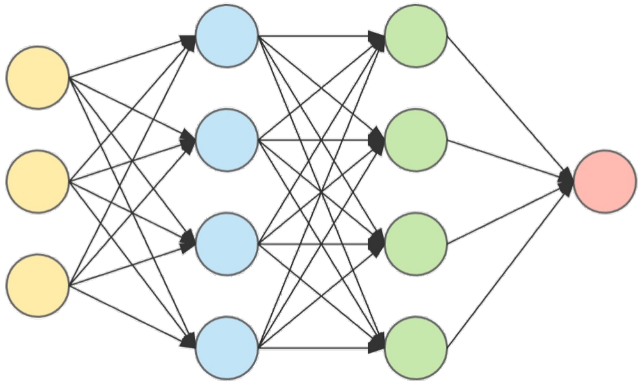


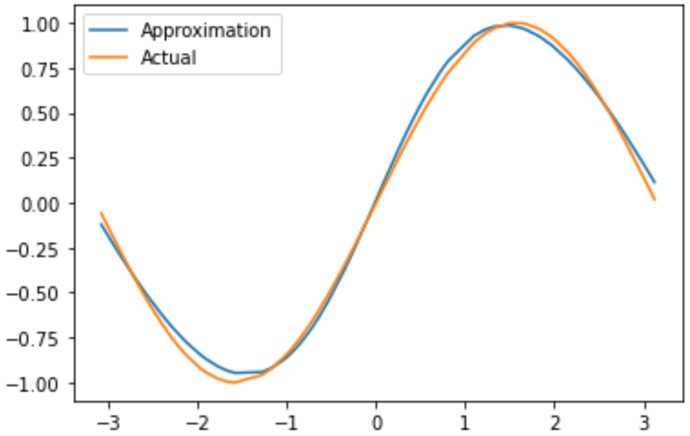
- activation introduces non-linearity (e.g. $g(x) = \max(0, x)$)
- adjust weights by minimizing loss

The Universal Approximation Theorem

Theorem (informal):

"A feedforward neural network with enough neurons can approximate any continuous function with arbitrary accuracy."





Limitations in practice:

1) amount of **training data**, 2) **size** of NN, and 3) **compute** spent for training

Types of ML (selection)

Tasks

- regression (e.g. calorimeter calibration)
- classification (e.g. jet tagging)
- generation (e.g. event generation)

Learning types

- supervised (e.g. amplitude regression)
- unsupervised (e.g. data clustering)
- semi-supervised (e.g. anomaly detection)

ML workflow

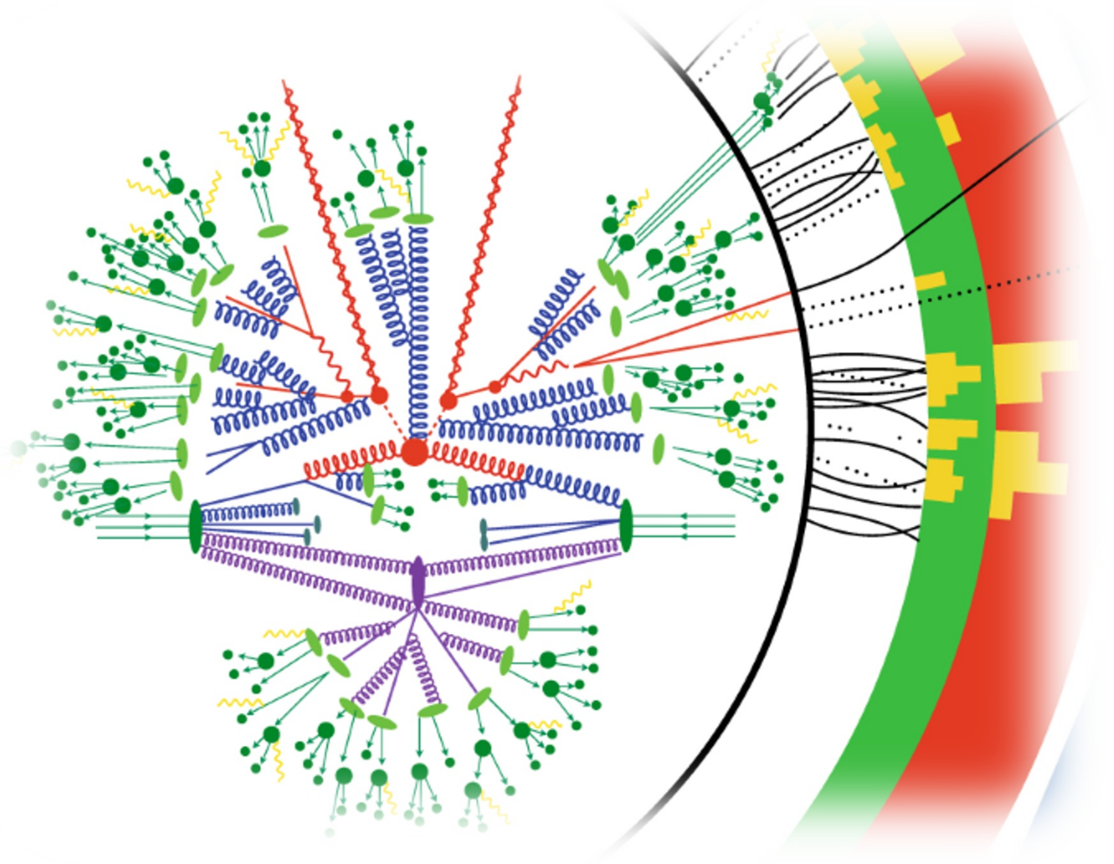
1. define the problem
2. collect and preprocess the dataset
3. define your ML model
4. training
5. evaluation

ML workflow

1. define the problem
2. collect and preprocess the dataset
3. define your ML model
4. training
5. evaluation

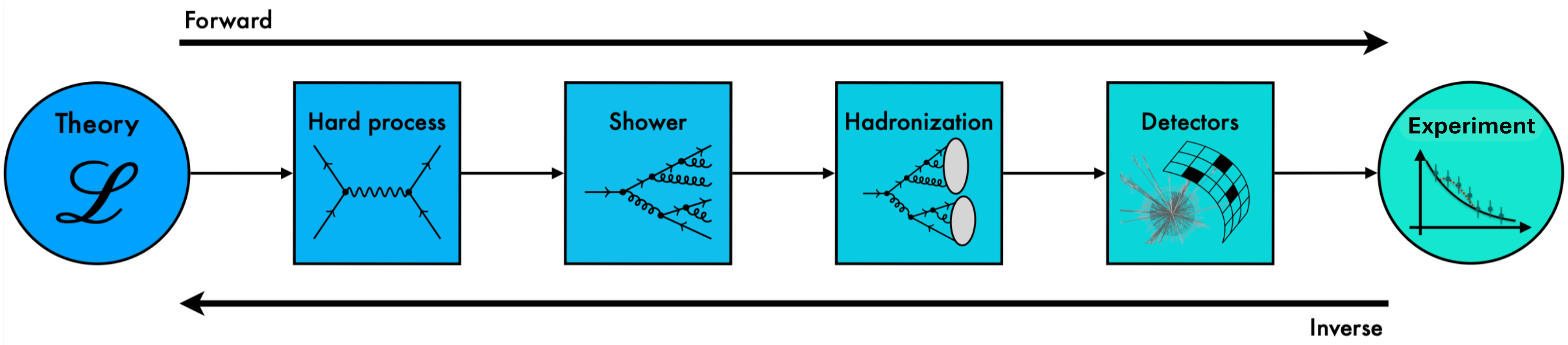
- ML strategy — *multiple ways to approach problem*
- loss — *what objective do I want to optimize?*
- architecture — *what is the best structure for my NN?*
- encode physics knowledge — *symmetries, …*

ML for particle physics



[Iulia Georgescu,2021]

The particle physics workflow



ML can help with each of these steps by increasing

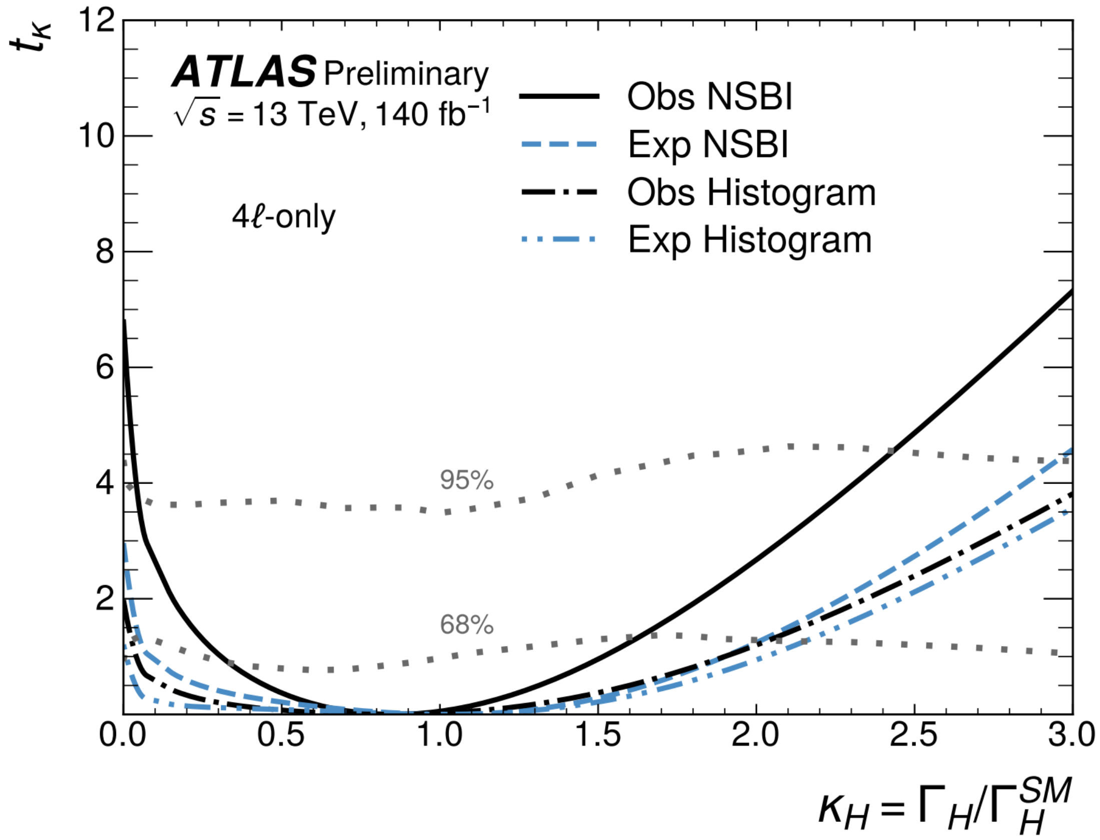
- accuracy/performance
- speed

Motivating example: simulation-based inference

- simulation-based inference (SBI) allows to exploit full kinematic information
- significant improvement in comparison to histogram approach

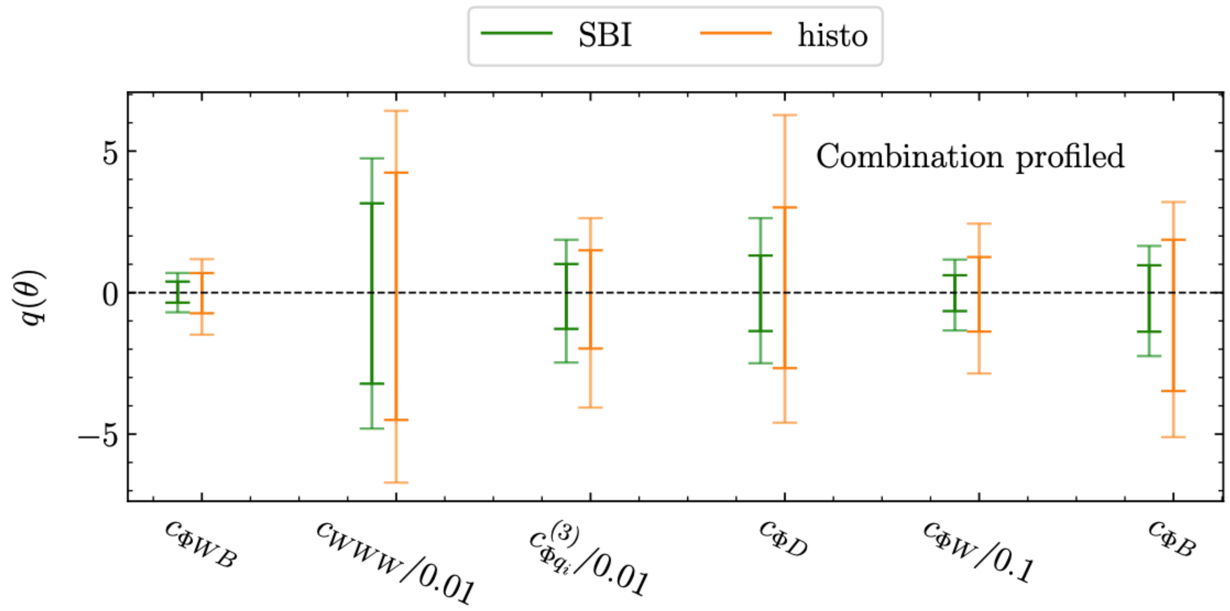
Motivating example: simulation-based inference

- simulation-based inference (SBI) allows to exploit full kinematic information
- significant improvement in comparison to histogram approach
- 1st experimental analysis: measure off-shell signal strength in $H \to ZZ$ channel
  [ATLAS-CONF-2024-016]



Motivating example: simulation-based inference

- simulation-based inference (SBI) allows to exploit full kinematic information
- significant improvement in comparison to histogram approach
- 1st experimental analysis: measure off-shell signal strength in $H \to ZZ$ channel
  [ATLAS-CONF-2024-016]
- also persists in global analyses
  [HB,Plehn,Schmal,2509.05409]



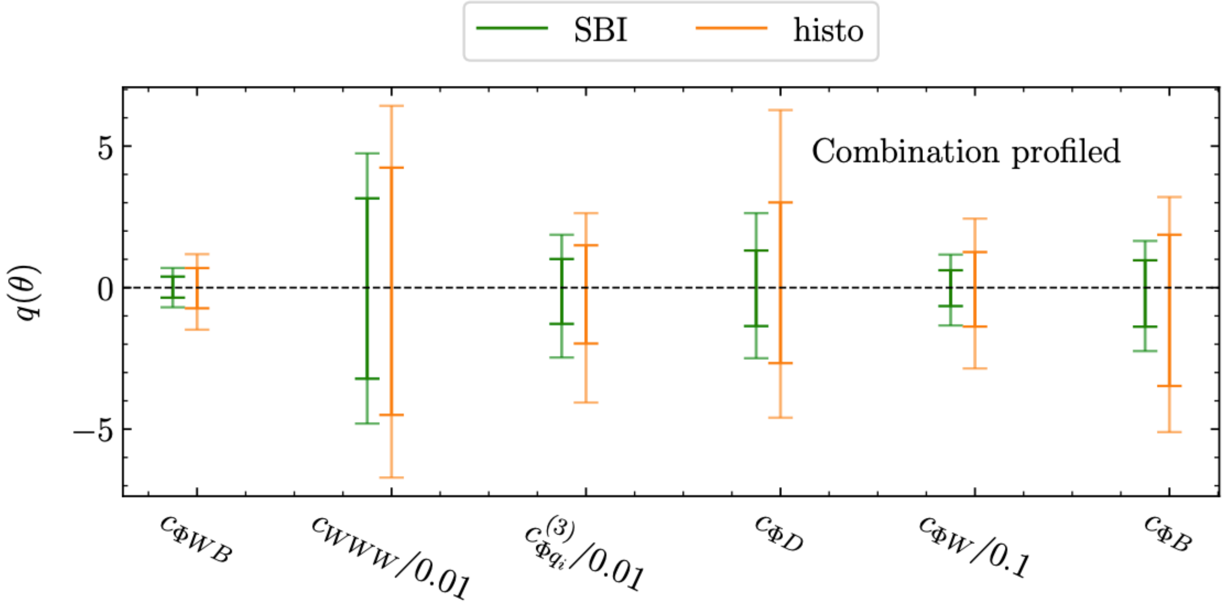
Motivating example: simulation-based inference

- simulation-based inference (SBI) allows to exploit full kinematic information

- significant improvement in comparison to histogram approach

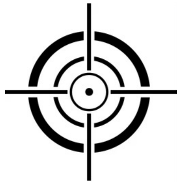
- 1st experimental analysis: measure off-shell signal strength in $H \to ZZ$ channel
  [ATLAS-CONF-2024-016]

- also persists in global analyses
  [HB,Plehn,Schmal,2509.05409]



➡ What is needed to apply ML successfully?

ML for particle physics

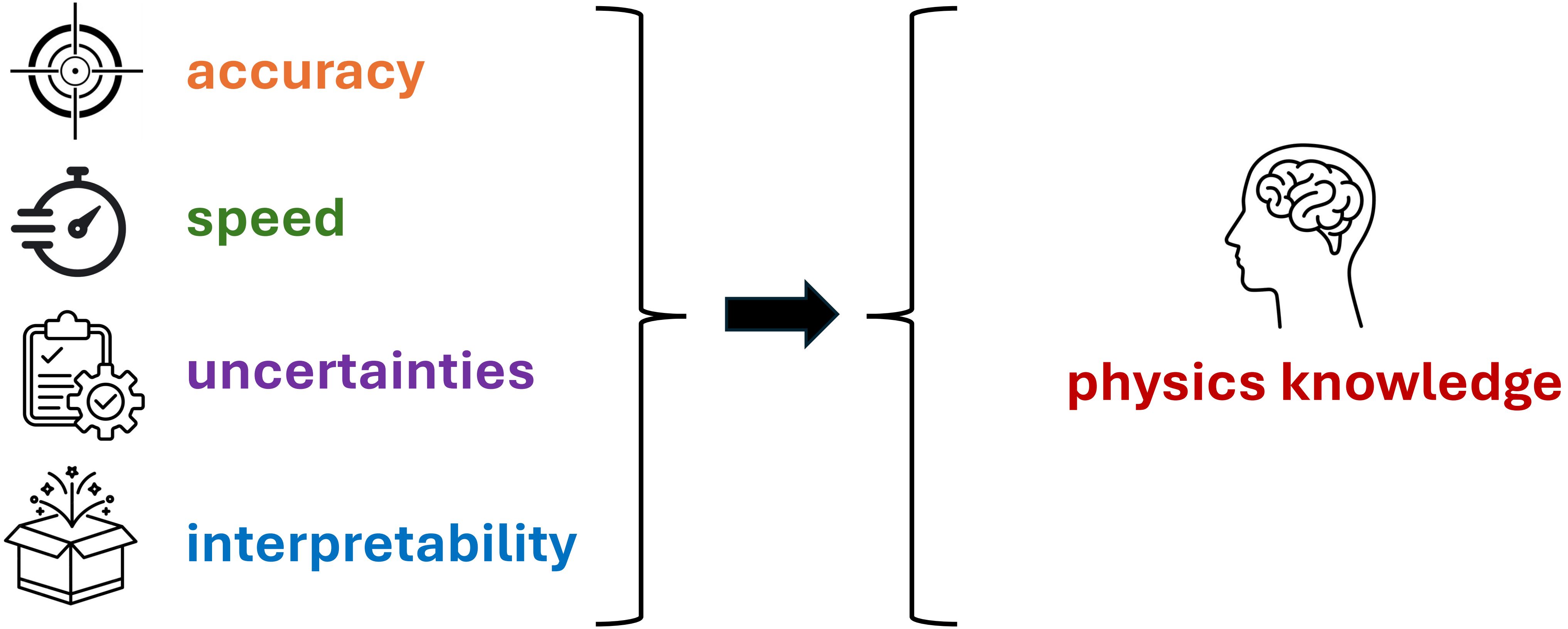
 accuracy

 speed

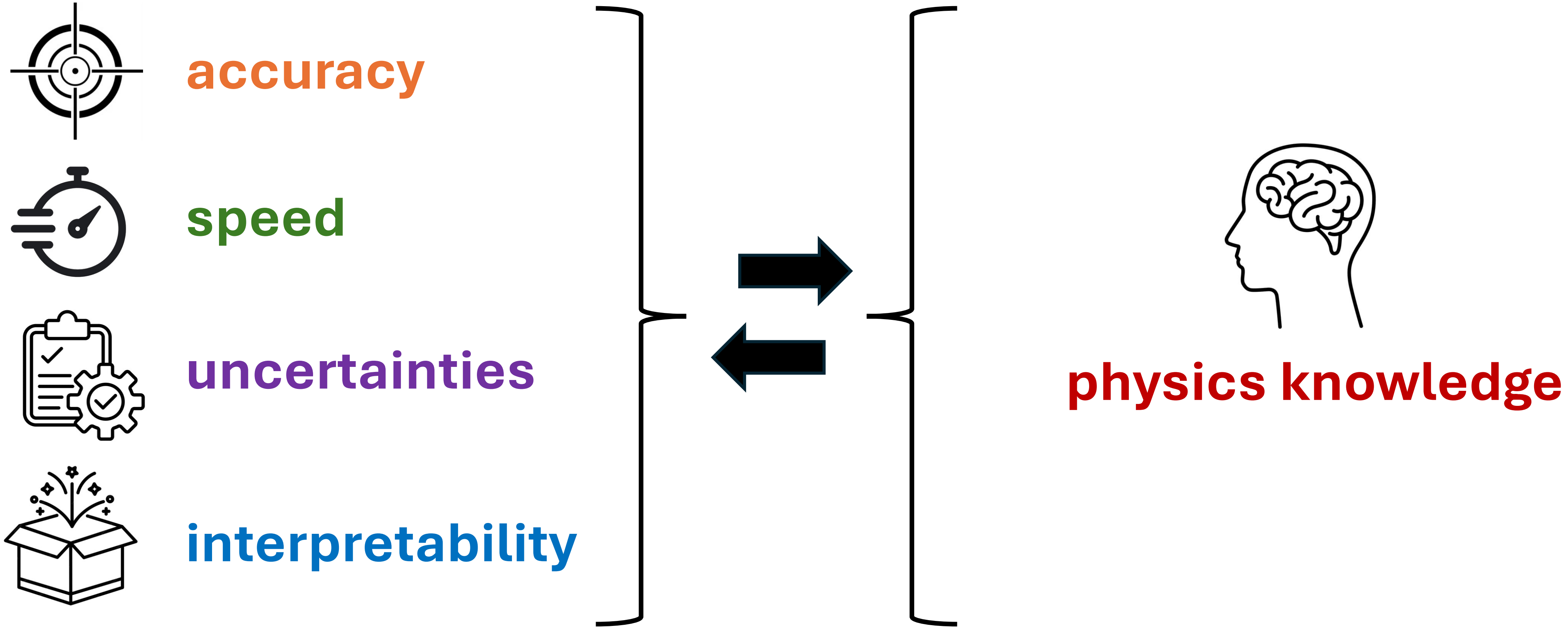
 uncertainties

 interpretability

ML for particle physics



ML for particle physics



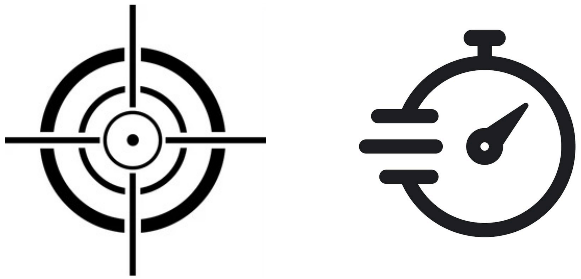
ML for particle physics

- accuracy
- speed
- uncertainties
- interpretability

⇄

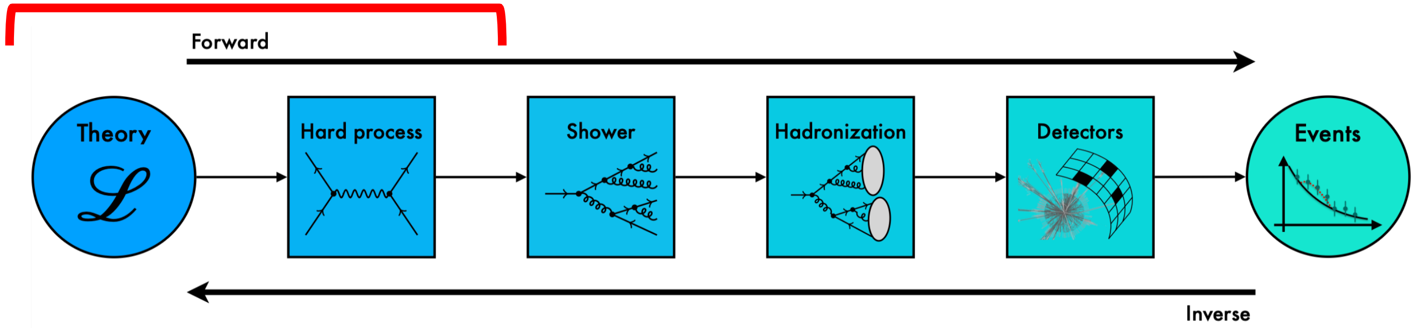
physics knowledge

➡ Key to all these aspects: finding good representations of the data



Accuracy & speed

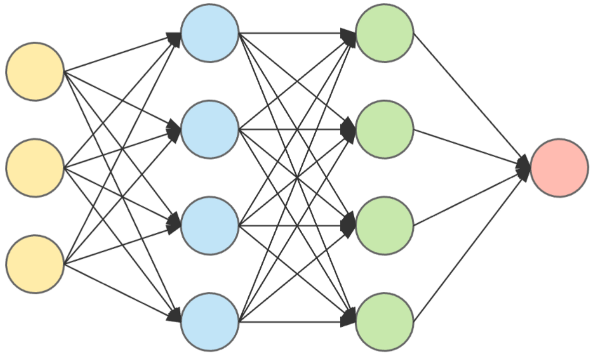
fast higher-order amplitude surrogates



Amplitude surrogates

- evaluating analytic expressions for amplitudes $|\mathcal{M}|^2$ can be very expensive due to
  - higher-order corrections
  - large final-state multiplicities

$$|\mathcal{M}|^2 \approx$$



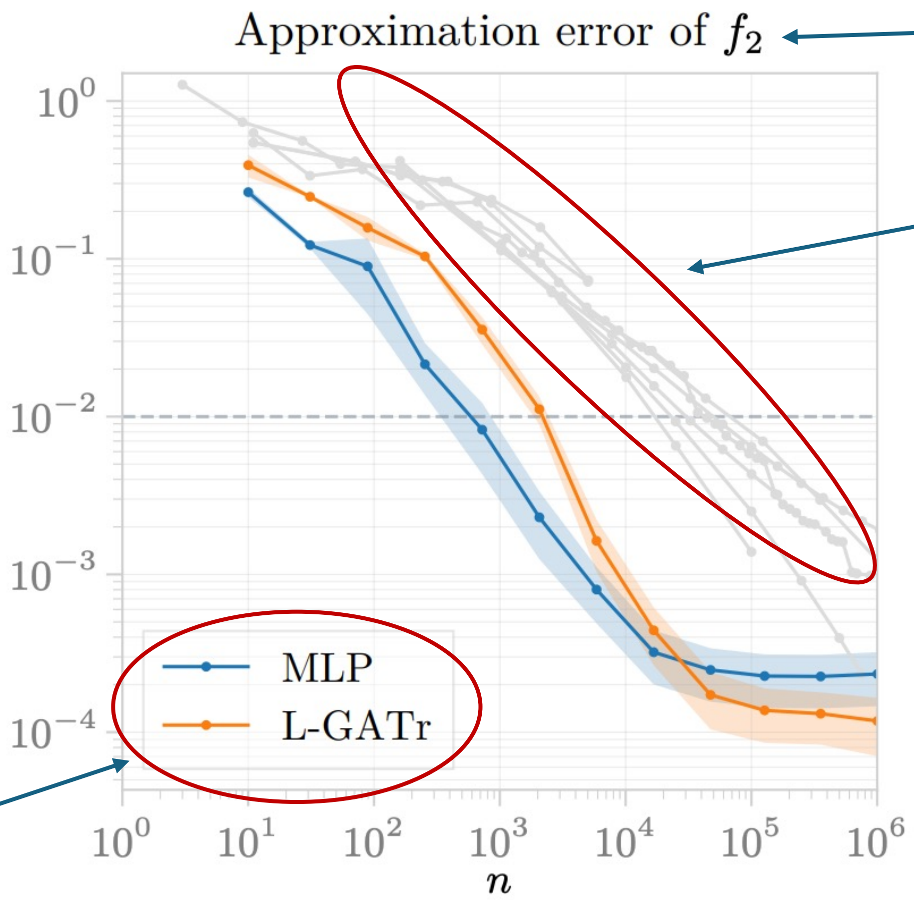
- idea:
  - generate small training sample using full analytic expression
  - train a NN to approximate $|\mathcal{M}|^2$
  - generate events using NN surrogate → fast to evaluate

→ fast high-precision event generation

Comparison to classical interpolation

[Bresó, Heinrich, et al., 2412.09534]

NLO $q\bar{q} \to t\bar{t}H$ amplitude



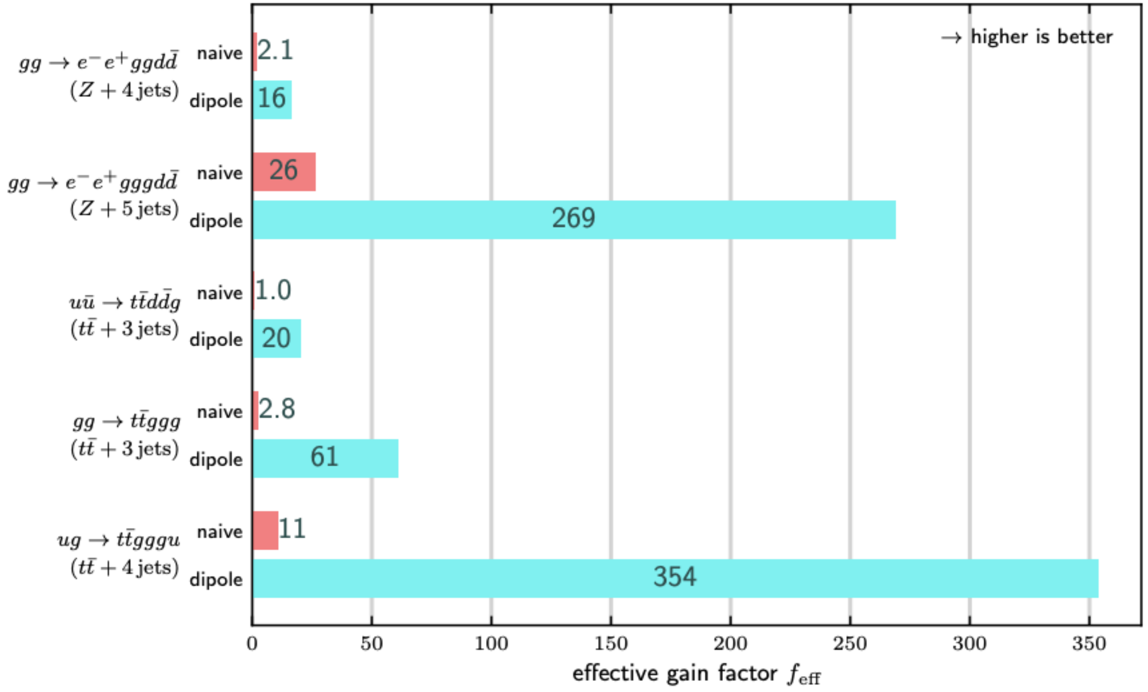
classical interpolation techniques

ML surrogates

➡ ML surrogates outperform classical interpolation techniques

Speed comparison

[Janßen et al.,2301.13562]



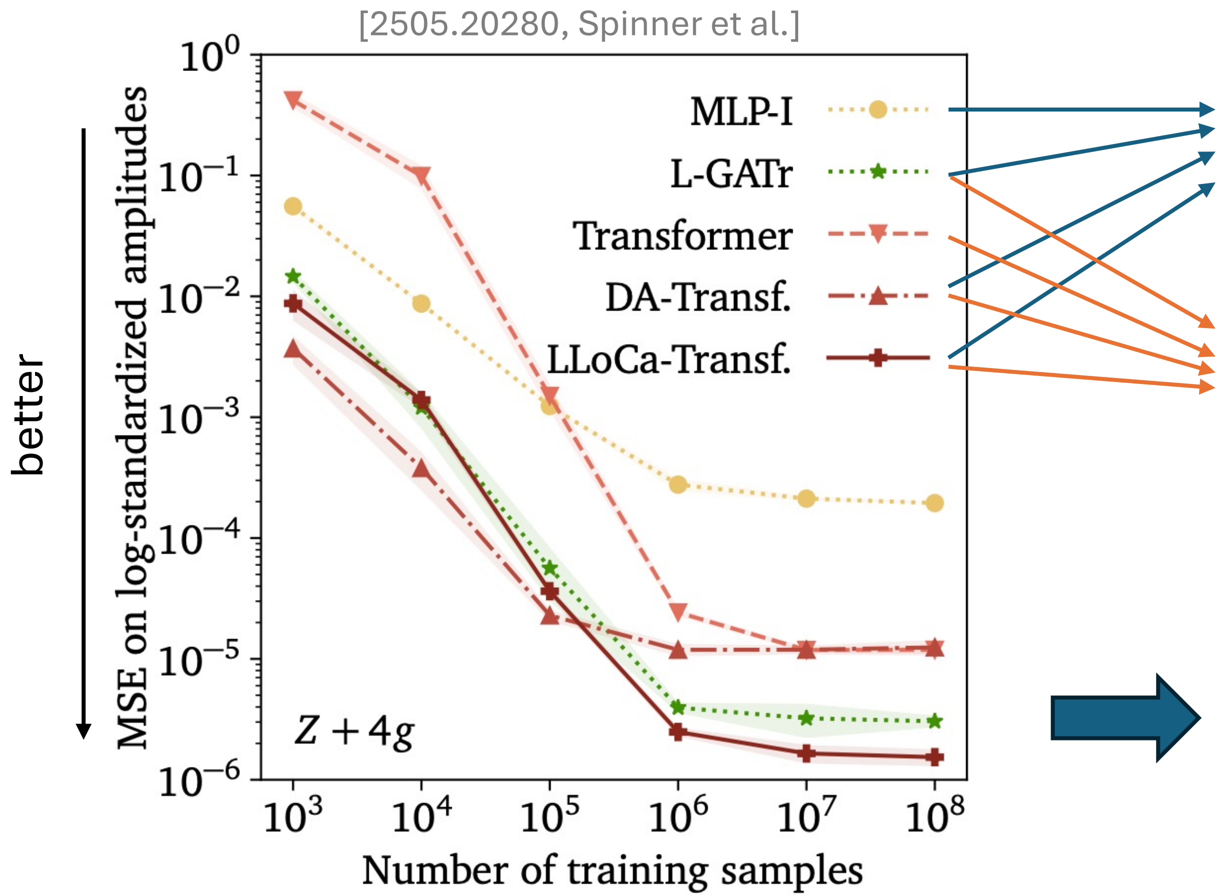
	SHERPA default			with dipole-model surrogate				
Process	t_{ME}[ms]	t_{PS}[ms]	ϵ_{full}	t_{surr}[ms]	x_{max}	$\epsilon_{\mathrm{1st,surr}}$	$\epsilon_{\mathrm{2nd,surr}}$	f_{eff}
$gg \to e^- e^+ gg d\bar{d}$	54	0.40	1.411 %	0.14	2.6	1.418 %	39 %	16
$gg \to e^- e^+ ggg d\bar{d}$	16 216	5.70	0.076 %	0.20	3.6	0.085 %	29 %	269

$$f_{\mathrm{eff}} = \frac{T_{\mathrm{standard}}}{T_{\mathrm{surrogate}}}$$

dipole vs naïve:
encode singularity structure of amplitudes

Large speed-ups possible!

Exploiting known symmetries



Lorentz invariance

permutation invariance ($g_i \leftrightarrow g_j$)

Enforcing symmetries drastically improves performance!

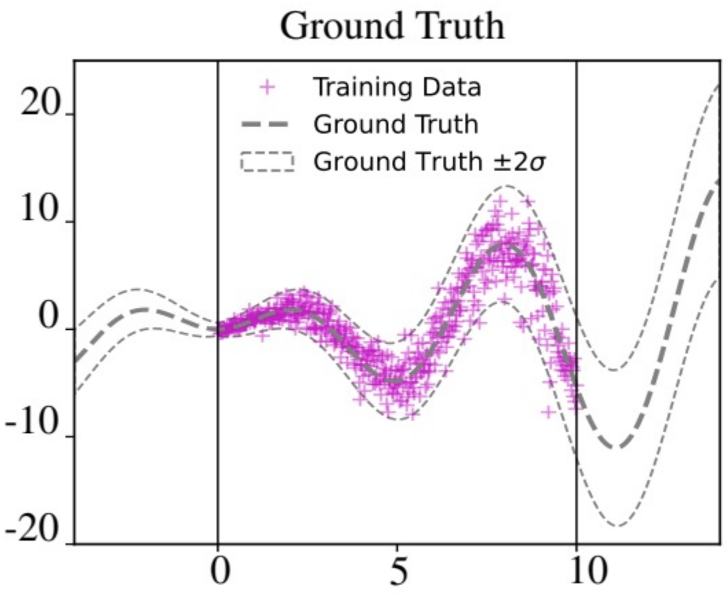
→ physic-informed data representations

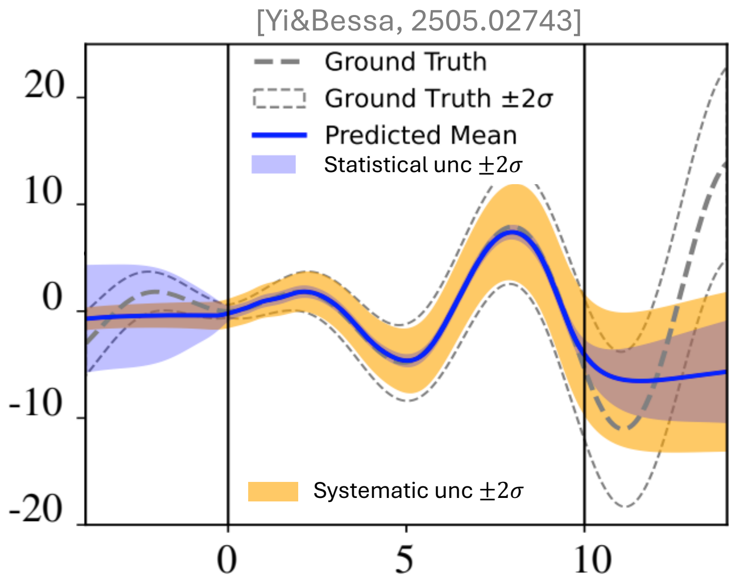
Uncertainties

"All models are wrong, but some — those that know when they can be trusted — are useful!"

George Box (adapted)

Regression with uncertainties





- statistical uncertainty $\widehat{=}$ lack of training data
- systematic uncertainty $\widehat{=}$ noise in the data, lack in model expressivity

➡ Can we the NNs encode a representation of their own uncertainties?

Probabilistic learning

Learn amplitude statistically

NN parameters

$$p(A|x) = \int \mathrm{d}\theta \; p(\theta|D_{\text{train}})\, p(A|x,\theta) \approx \int \mathrm{d}\theta \; q(\theta)\, p(A|x,\theta)$$

Then, we can calculate the mean prediction and uncertainties as

$$A_{\text{NN}}(x) = \int dA\, A\, p(A|x) = \int d\theta\, q(\theta) \bar{A}(x,\theta) \qquad \text{with} \quad \bar{A}(x,\theta) = \int dA\, A\, p(A|x,\theta)$$

$$\sigma^2_{\text{syst}}(x) = \int d\theta\, q(\theta) \left[\overline{A^2}(x,\theta) - \bar{A}(x,\theta)^2 \right]$$ → vanishes for perfect data: $p(A|\theta) \to \delta(A - A_0)$

$$\sigma^2_{\text{stat}}(x) = \int d\theta\, q(\theta) \left[\bar{A}(x,\theta) - A_{\text{NN}}(x) \right]^2$$ → vanishes for perfect training: $q(\theta) \to \delta(\theta - \theta_0)$

Modelling the systematic uncertainty

- log-likelihood loss:

$$\mathcal{L} = - \sum_{x_i, A_i \in D_{\text{train}}} \log p(A_{\text{true}}(x_i)|x_i, \theta)$$

true amplitudes

phase-space point

- assume Gaussian likelihood: $p(A|x) = \mathcal{N}(\overline{A}(x), \sigma_{\text{syst}}^2(x))$

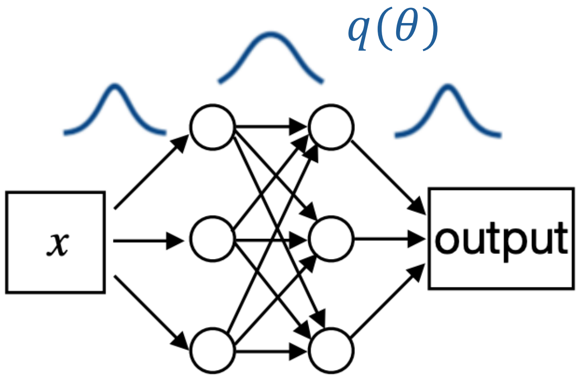
- NN learns both: $\overline{A}(x)$ and $\sigma_{\text{syst}}(x)$

$$\Rightarrow \text{heteroskedastic loss: } \mathcal{L} = \sum_i \left[\frac{\left(\overline{A}(x_i) - A_{\text{true}}(x_i)\right)^2}{2\sigma_{\text{syst}}^2(x_i)} + \log\left(\sigma_{\text{syst}}(x_i)\right) \right]$$

- if needed: replace by Gaussian mixture model

Modelling the statistical uncertainty

- variational approximation: $p(\theta|D_{\text{train}}) \simeq q(\theta)$
- promote each NN parameter to Gaussian distribution
- train by minimizing KL divergence:



$$\mathrm{KL}[q(\theta), p(\theta|D_{\text{train}})] = \int d\theta\, q(\theta) \log \frac{q(\theta)}{p(\theta|D_{\text{train}})}$$

Bayes' theorem →

$$= \int d\theta\, q(\theta) \log \frac{q(\theta) p(D_{\text{train}})}{p(\theta) p(D_{\text{train}}|\theta)}$$

$$= \underbrace{\mathrm{KL}[q(\theta), p(\theta)]}_{\text{prior}} - \underbrace{\int d\theta\, q(\theta) \log p(D_{\text{train}}|\theta)}_{\text{log likelihood}} + \ \dots$$

Alternative: repulsive ensembles

- can describe NN training via ODE or continuity equation:

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

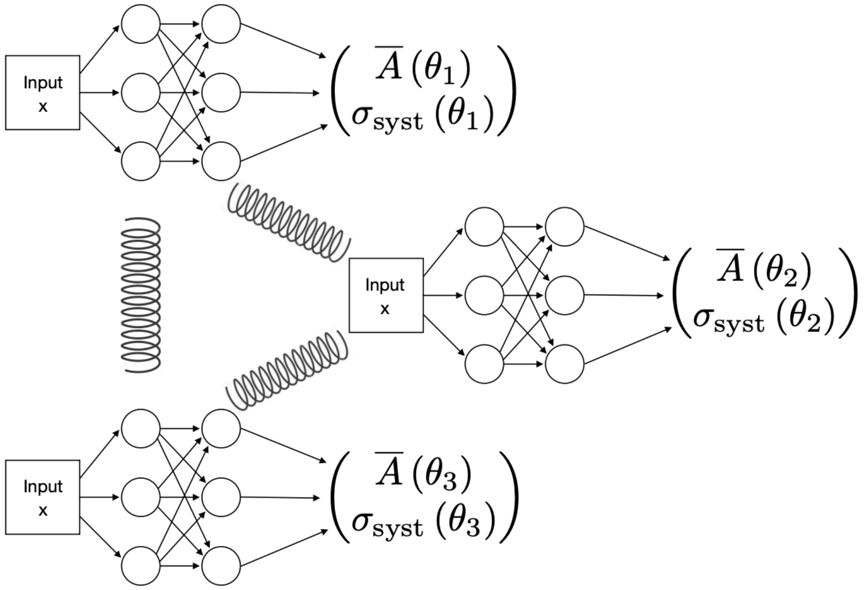
- choose $v(\theta, t) = -\nabla_\theta \log \frac{\rho(\theta, t)}{\pi(\theta)} \rightarrow$ solution: $\rho(\theta) = \pi(\theta) \equiv p(\theta | D_{\text{train}})$

- estimate density via NN ensemble: $\rho(\theta^t) \approx \frac{1}{n} \sum_{i=1}^{n} k(\theta^t, \theta_i^t)$

kernel

- NN parameter update rule

$$\frac{d\theta}{dt} = -\nabla_\theta \left[\log \left(\frac{1}{n} \sum_i k(\theta, \theta_i) \right) - \log p(\theta | x) \right]$$

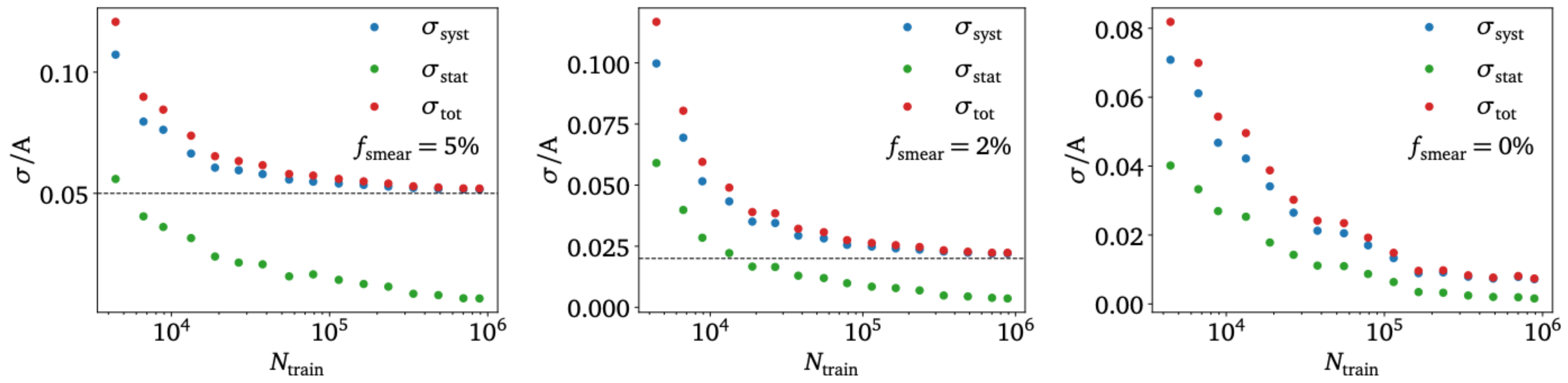


➡ NN ensemble with repulsive force ensuring $\theta \sim p(\theta | D_{\text{train}})$

Behavior of uncertainties

[HB,Elmer,Favaro,...,2412.12069]

$$A_{\text{train}} \sim \mathcal{N}(A_{\text{true}}, \sigma_{\text{train}}^2)$$
$$\sigma_{\text{train}} = f_{\text{smear}} A_{\text{true}}$$



test: apply Gaussian noise to $gg \to \gamma\gamma g$ amplitudes

- statistical unc. decreases with more training data
- systematic unc. converges to level of applied noise

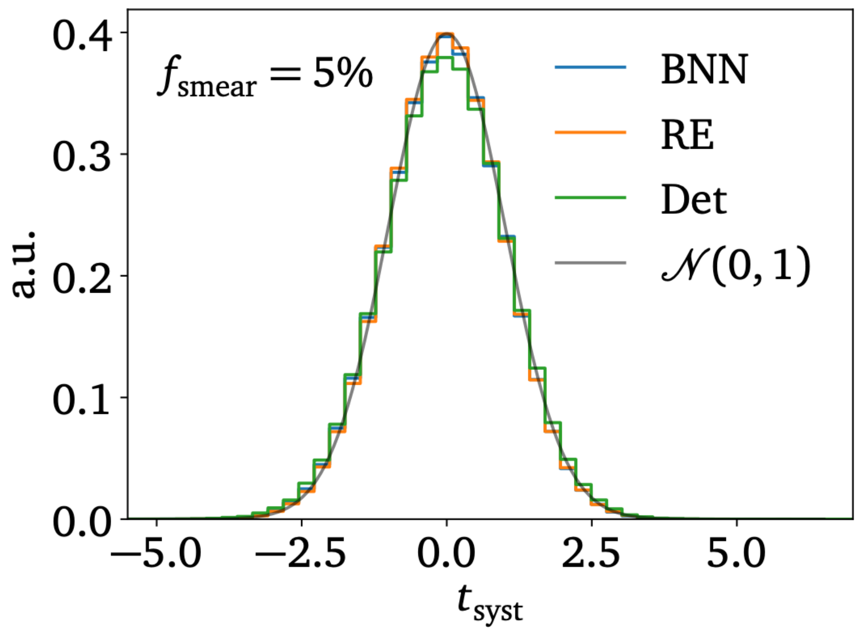
→ reliable uncertainty estimate

Are these uncertainties calibrated?

- statistical uncertainties play minor role for amplitude regression
- define systematic pull:

$$t_{\text{syst}} = \frac{\langle A \rangle(x) - A_{\text{train}}(x)}{\sigma_{\text{syst}}(x)}$$

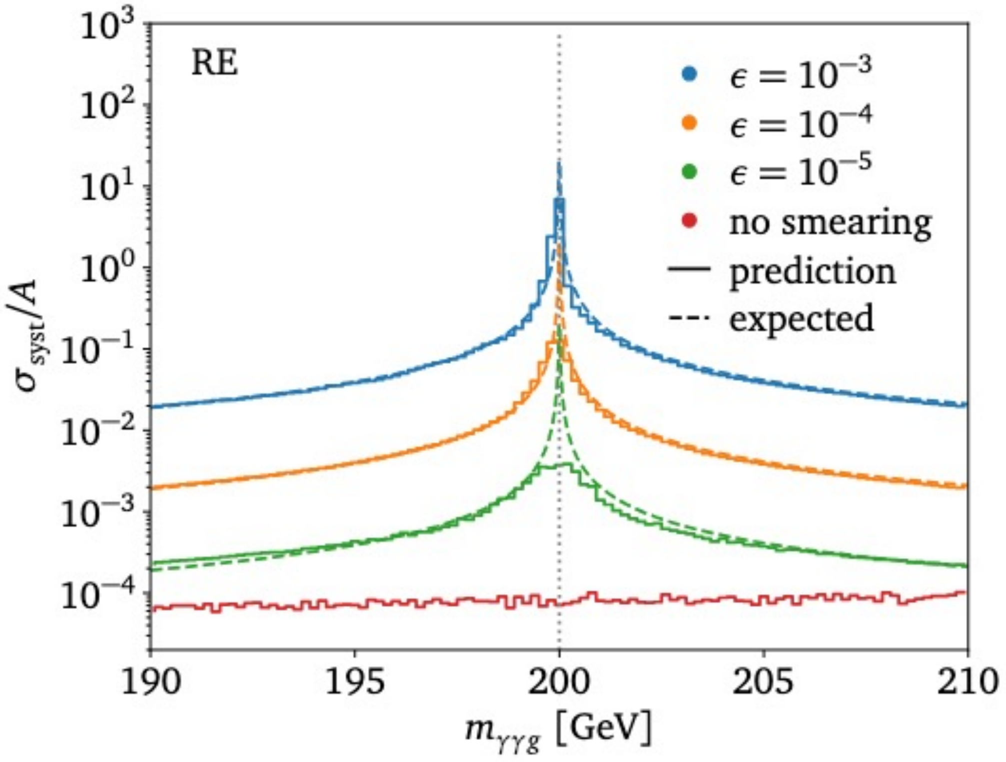
- if calibrated, t_{syst} distribution should follow $\mathcal{N}(0,1)$



Almost perfectly calibration → reliable uncertainty estimate

Localized noise

[HB,Elmer,Plehn,Winterhalder, 2509.00155]



$$A_{\text{train}}(x) = \mathcal{N}\left(A_{\text{true}}(x), \frac{\epsilon\, m_{\text{thresh}}}{|m_{\gamma\gamma g}(x) - m_{\text{thresh}}|} A_{\text{true}}(x)\right)$$

- emulates numerical noise close to threshold
- well captured by systematic uncertainties
- NN effectively finds the mean prediction
- uncertainties still well calibrated

Same techniques also applicable to all kind of other regression problems!

Controlling generative ML

[HB,Diefenbacher,Elmer,..., 2509.08048]

training dataset → $D_{\text{true}}^{n_{\text{train}}}$ ~ $p_{\text{true}}(x)$ ← true distribution

generated dataset → $D_{\text{gen}}^{n_{\text{gen}}}$ ~ $p_{\text{gen}}(x)$ ← learned distribution

$$D_{\text{true}}^{n_{\text{train}}} \sim p_{\text{true}}(x) \quad \Longleftrightarrow \quad D_{\text{gen}}^{n_{\text{gen}}} \sim p_{\text{gen}}(x) \overset{?}{\sim} p_{\text{true}}(x)$$

How can we quantify the performance of the generative NN?

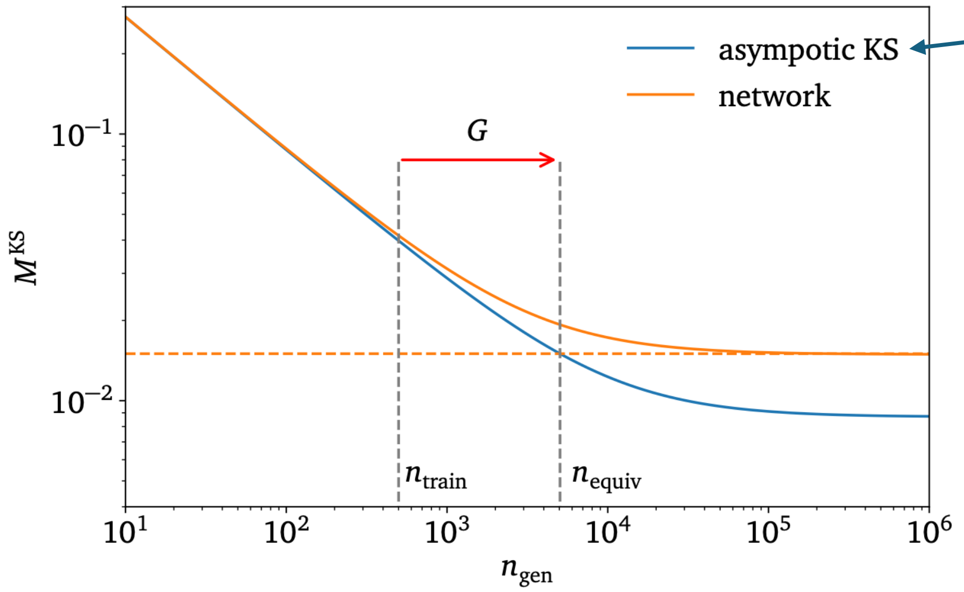
→ determine n_{equiv} such that $M\left(D_{\text{true}}^{n_{\text{equiv}}}, p_{\text{true}}(x)\right) \equiv M\left(D_{\text{gen}}^{n_{\text{gen}}}, p_{\text{true}}(x)\right)$ with $D_{\text{true}}^{n_{\text{equiv}}} \sim p_{\text{true}}(x)$

comparison metric (M)

⇒ amplification factor $G = \dfrac{n_{\text{equiv}}}{n_{\text{train}}}$

Controlling generative ML

one option for M: Kolmogorov-Smirnov test comparing D_{train} and D_{gen}



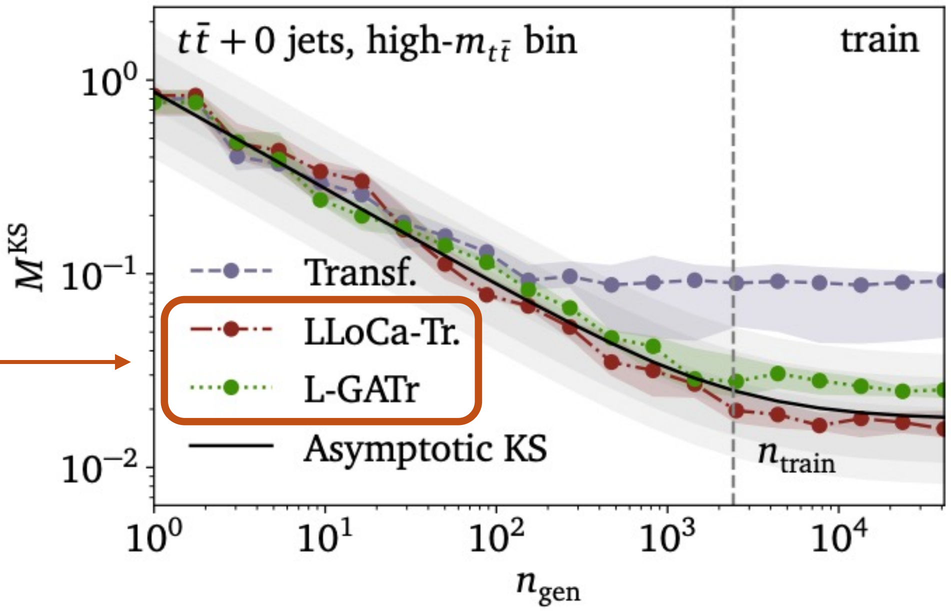
known asymptotic if $p_{\text{gen}} = p_{\text{true}}$

➡ systematic approach to assess quality of generative NNs

Controlling generative ML

one option for M: Kolmogorov-Smirnov test comparing D_{train} and D_{gen}

Lorentz-equivariant
&
permutation-invariant



systematic approach to assess quality of generative NNs

Interpretable ML

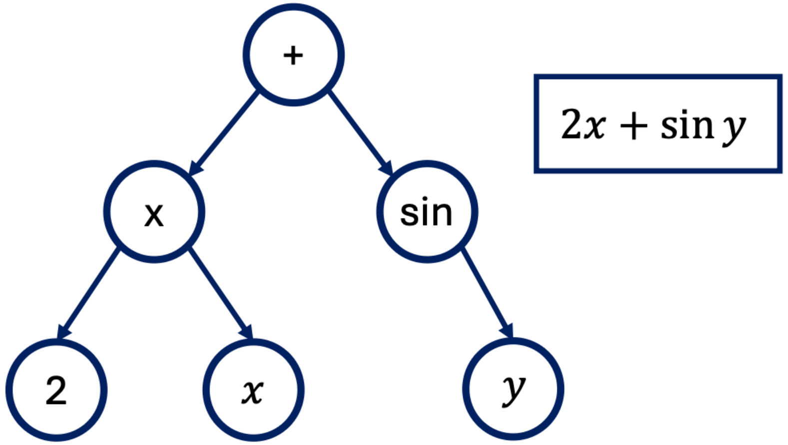
looking under the hood

Back to the formula – symbolic regression

- many ways to make ML interpretable
- goal: find most relevant representation/observables describing the data

→ maximal interpretability: analytic equation!

- construct them dynamically using symbolic regression
  [Schmidt&Lipson`09, Udrescu&Tegmark`19, Cranmer et al.`19,`20,`23]
- build upon genetic algorithm successively forming equation
- interplay between goodness-of-fit and complexity of equation



Example: Higgs CP test for VBF

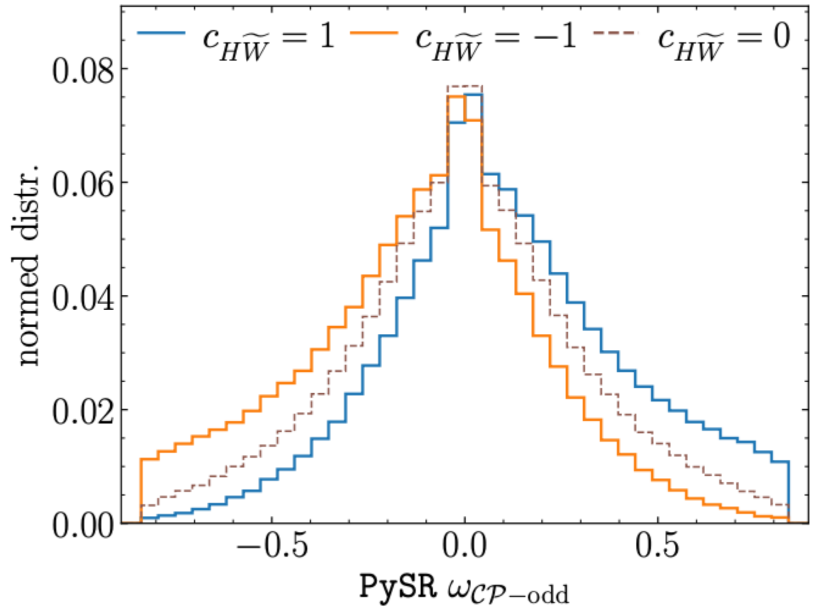
[HB,Menen,Fuchs,Plehn,2507.05858]

- consider dim-6 operator $\frac{c_{H\widetilde{W}}}{\Lambda^2}\Phi^\dagger\Phi\widetilde{W}^a_{\mu\nu}W^{a\mu\nu}$

- unambiguous CP test → CP-odd observables

- construct optimal reco-level CP-odd obs. by training a classifier on $c_{H\widetilde{W}} = \pm 1$ samples

- analytic equation → ensure learned observable is indeed CP-odd



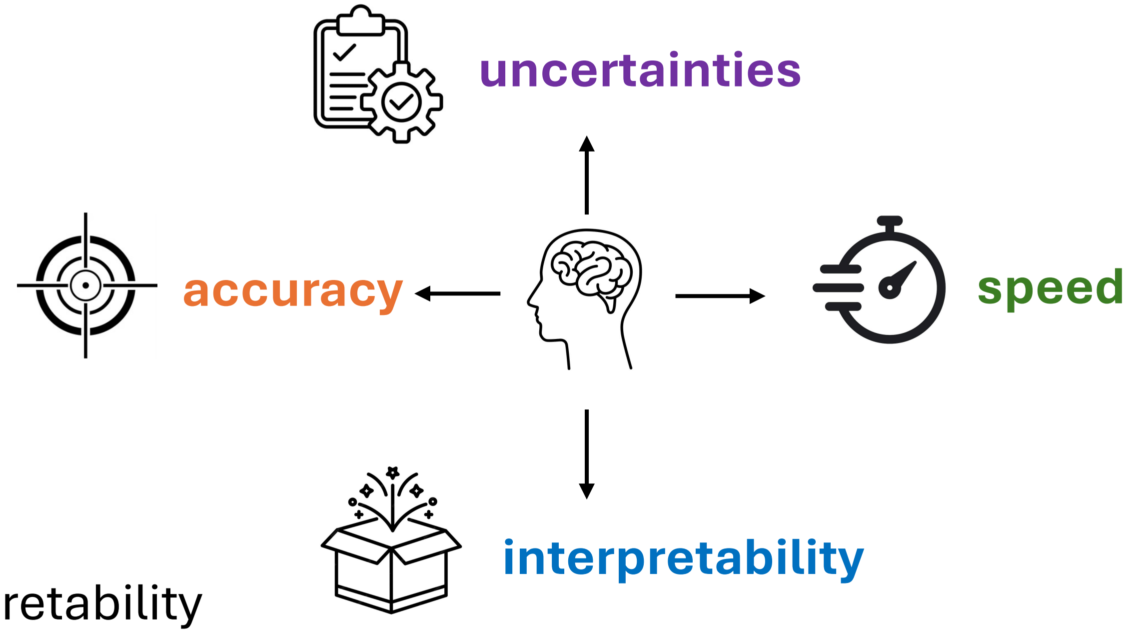
	$\sigma(c_{H\widetilde{W}} = 1$ vs. SM)
$p_{T,j_1}p_{T,j_2}\sin\Delta\phi_{jj}$	6.76
trained on $c_{H\widetilde{W}} = \pm 1$	
PySR	6.98
SymbolNet	7.07
BDT	6.71

$$d^{\text{PySR}} = \frac{1.8566\sin\Delta\phi_{jj}}{\left|\frac{0.3080x_{j_1}\log\Delta\eta_{jj} + \log\Delta\eta_{jj}\sinh(x_{j_2} - 2.5977) + 0.3080\sinh x_h}{x_{j_1}\log\Delta\eta_{jj} + \sinh x_h}\right| + 0.6047}$$

with $x = p_T/m_h$

Conclusions

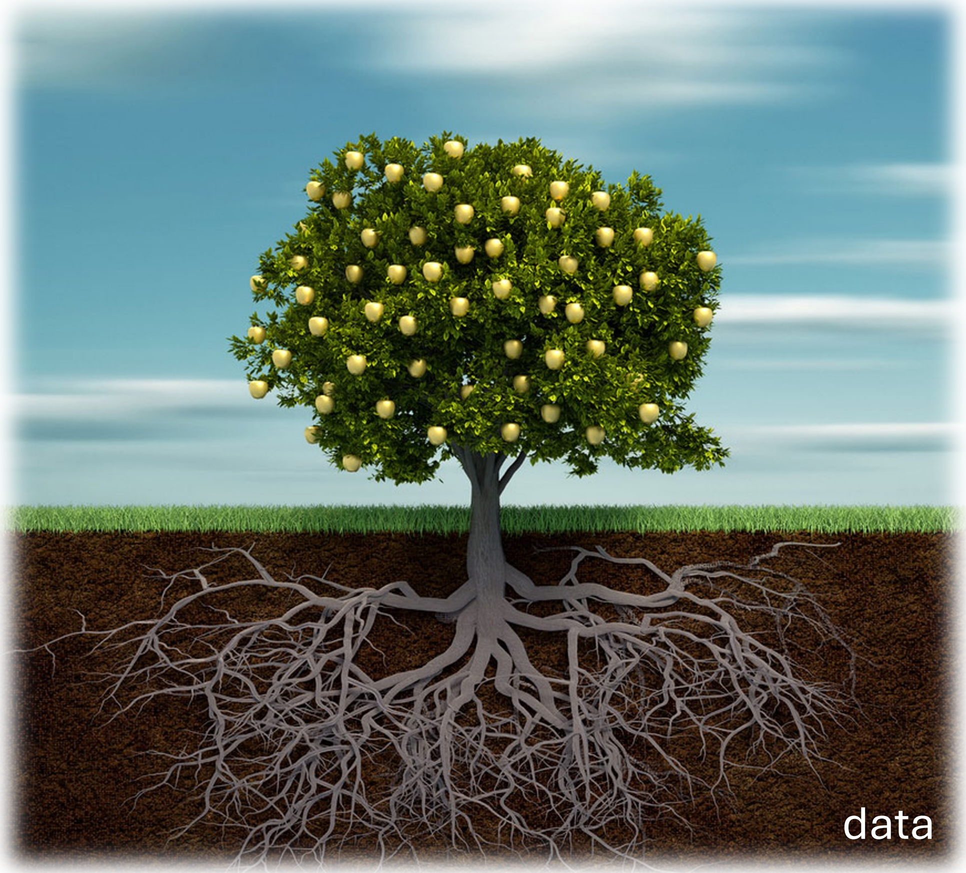
Conclusions

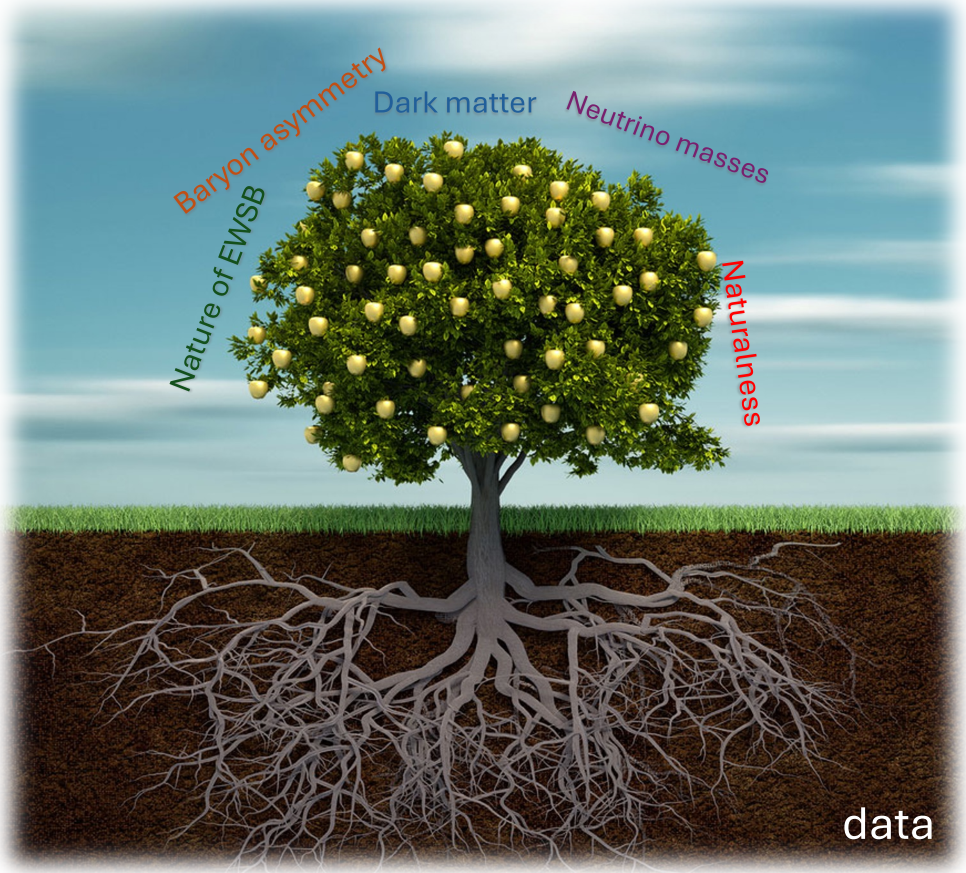


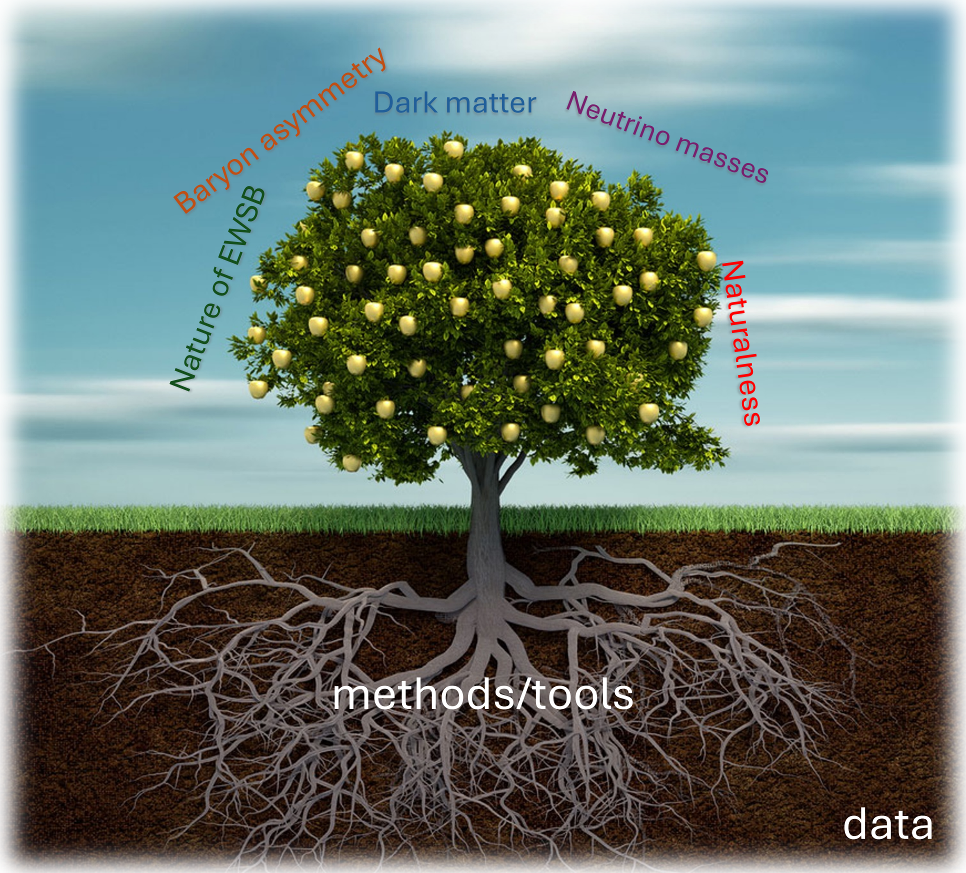
- particle physics is in the precision era
  → large amounts of multidimensional data
- ML methods excel in such an environment
- important requirements: uncertainties and interpretability
- key ingredient: representation learning based on particle theory
- methods widely applicable

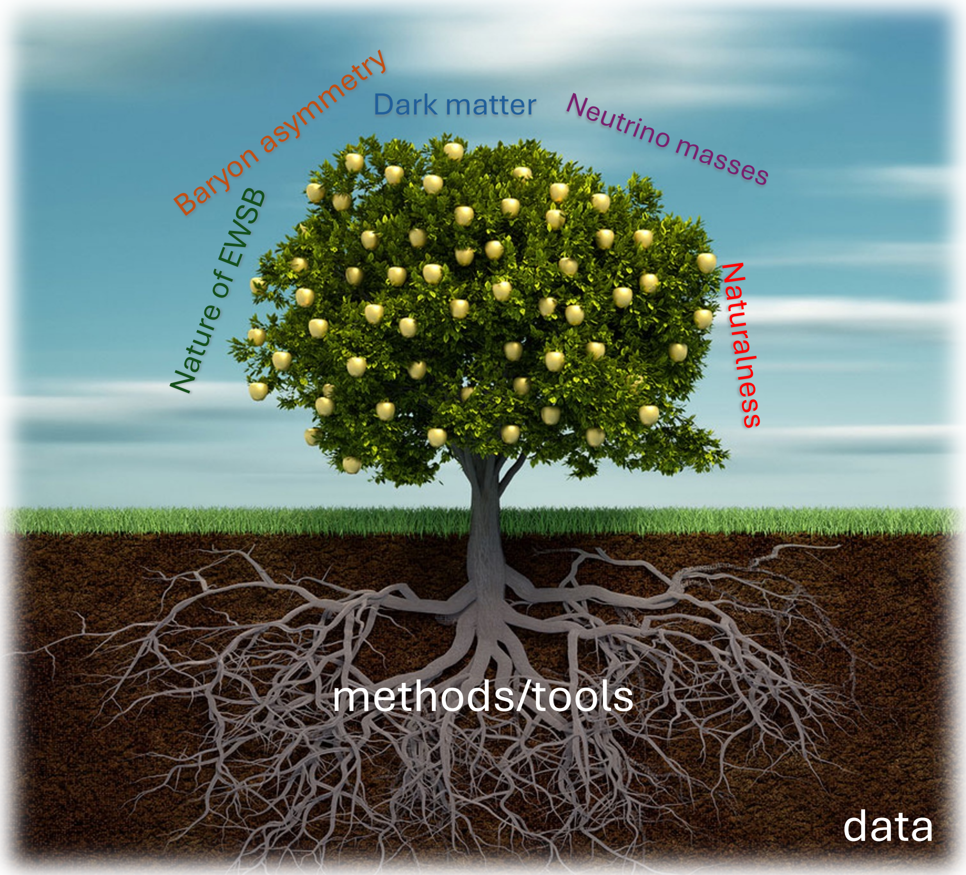
→ ML is an essential tool for the future of particle physics

Appendix


data

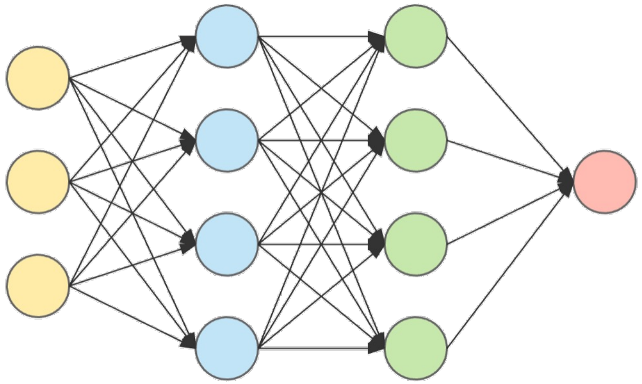

Baryon asymmetry
Dark matter
Neutrino masses
Nature of EWSB
Naturalness
data


Baryon asymmetry
Dark matter
Neutrino masses
Nature of EWSB
Naturalness
methods/tools
data



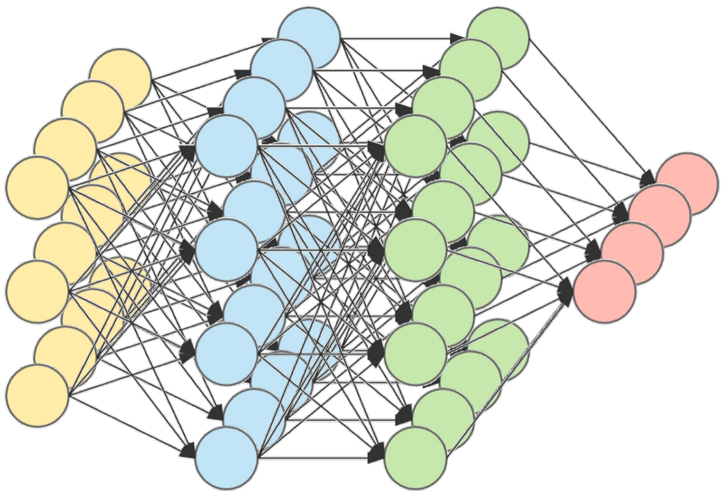
Thanks for your attention!

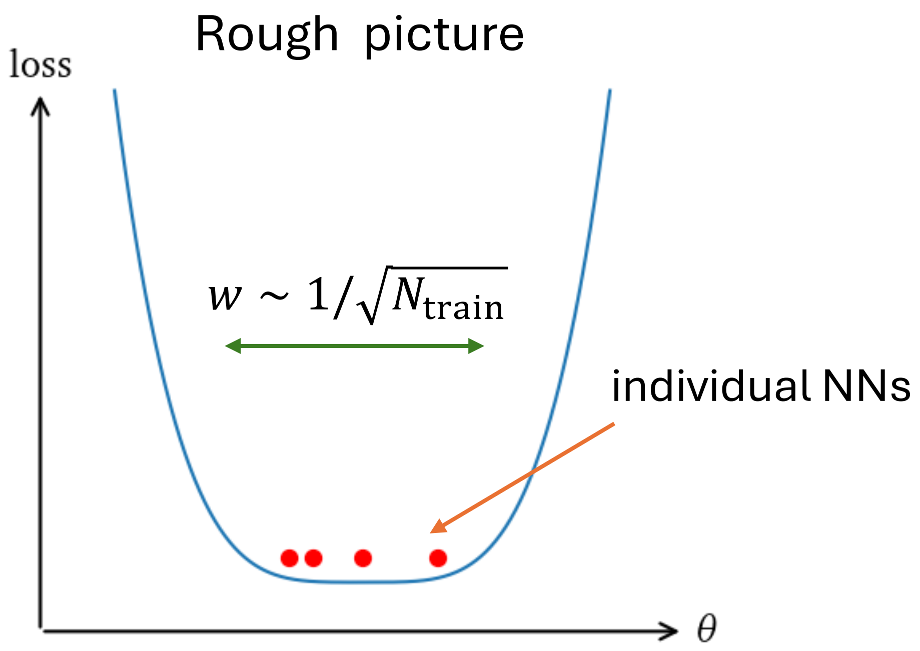
Modelling the statistical uncertainty



- train ensemble of networks
- each networks leads to slightly different result
- spread of network predictions $\sim$ statistical uncertainty
- less data $\rightarrow$ higher spread

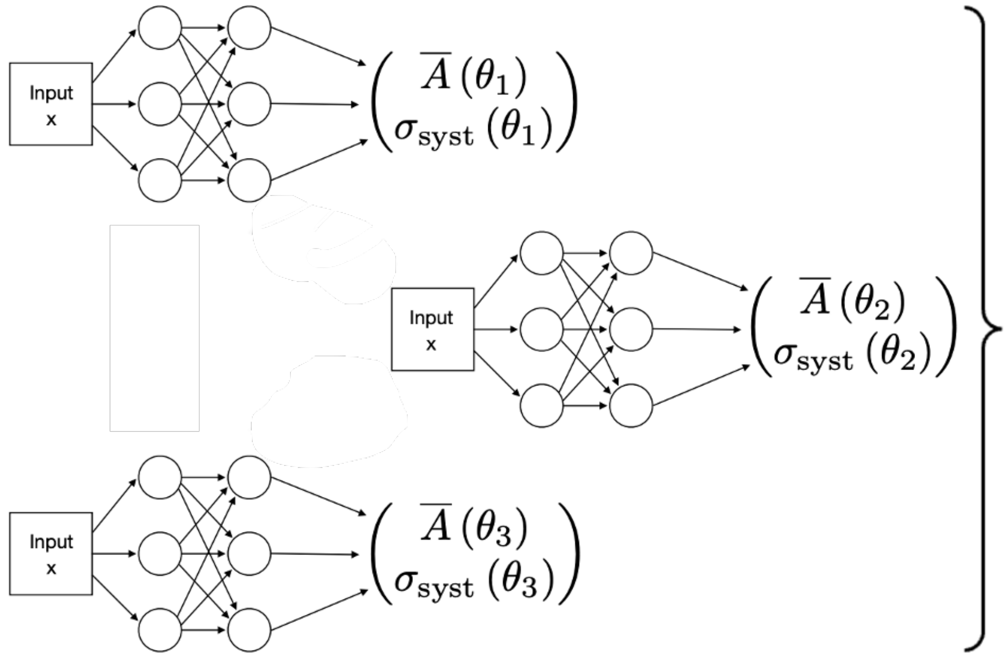
Modelling the statistical uncertainty





- train ensemble of networks
- each networks leads to slightly different result
- spread of network predictions ~ statistical uncertainty
- less data → higher spread

Bringing it all together

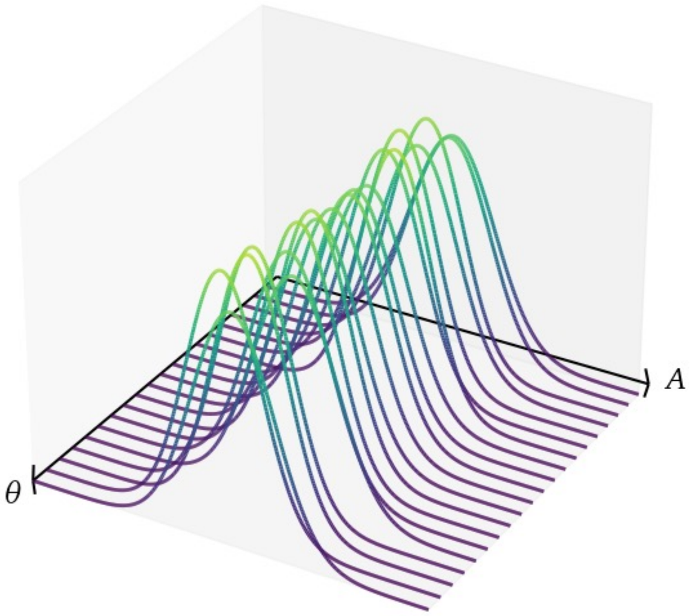


$$\langle A \rangle = \frac{1}{N} \sum_i^N \overline{A}(\theta_i)$$

$$\sigma_{\text{syst}}^2 = \frac{1}{N} \sum_i^N \sigma_{\text{syst}}^2(\theta_i)$$

$$\sigma_{\text{stat}}^2 = \frac{1}{N} \sum_i^N \left(\langle A \rangle - \overline{A}(\theta_i)\right)^2$$

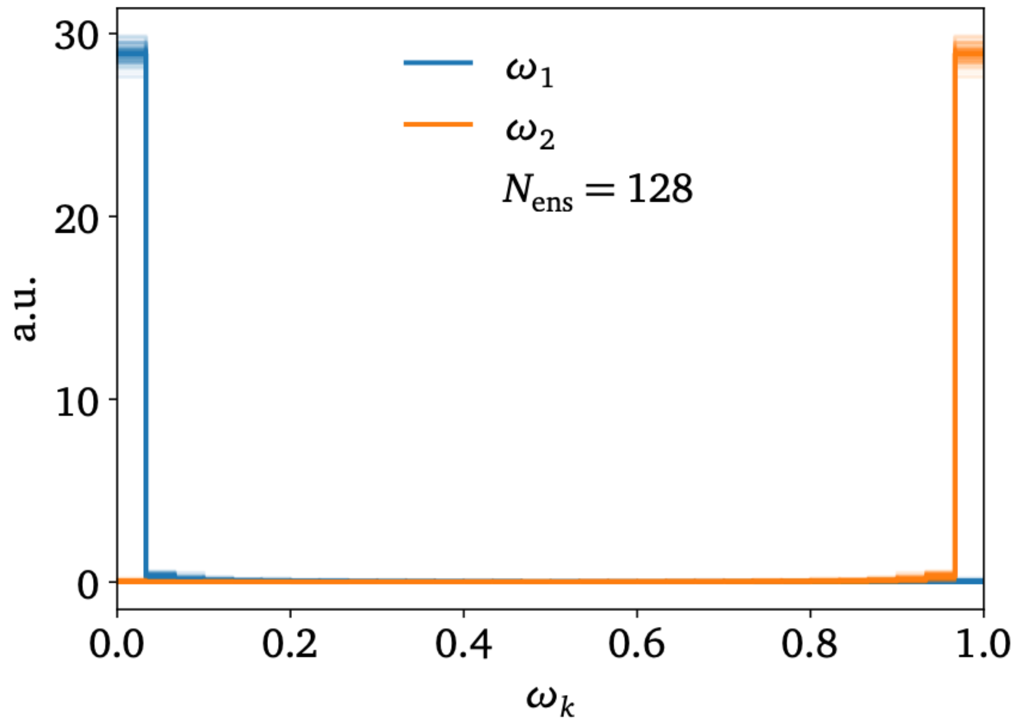
$$\sigma_{\text{tot}}^2 = \sigma_{\text{syst}}^2 + \sigma_{\text{stat}}^2$$



→ Combined learnable modelling of systematic and statistical uncertainties!

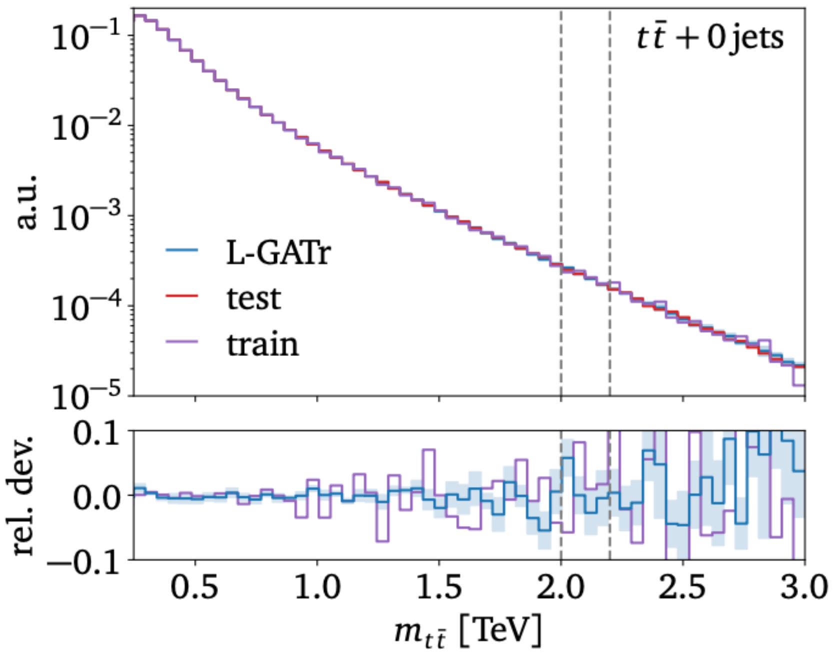
Alternative approaches: Bayesian neural networks, evidential regression

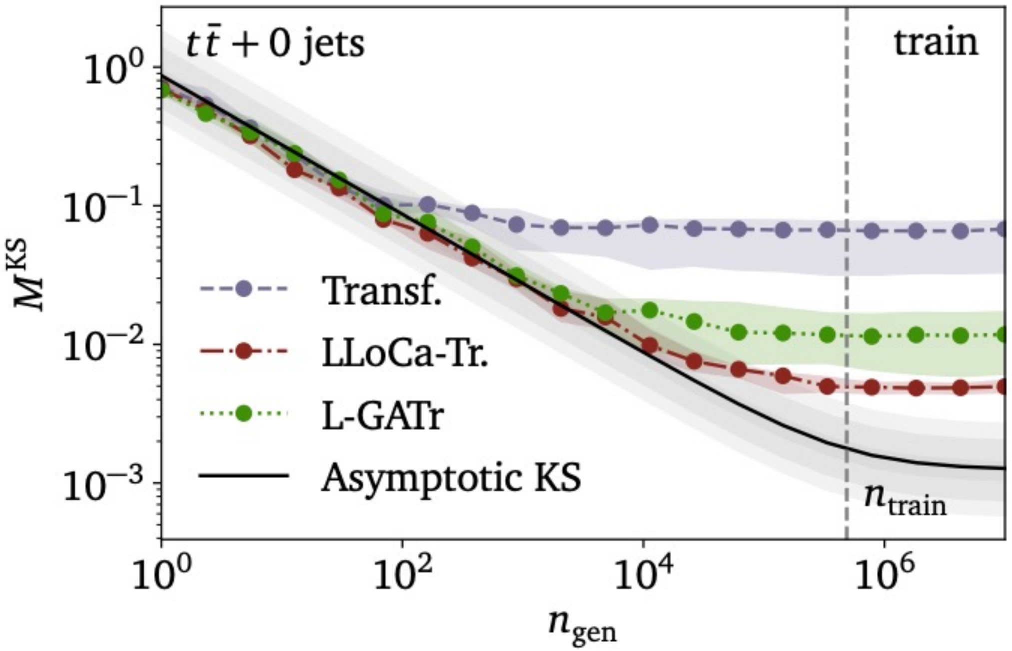
Gaussian mixture model



$$p_{\text{GMM}}(A|x,\theta) = \sum_{k=1}^{K} \omega_k(x,\theta)\, \mathcal{N}\Big(A\,\Big|\,\bar{A}_k(x,\theta),\, \sigma_k^2(x,\theta)\Big), \quad \text{with} \quad \sum_{k=1}^{K} \omega_k(x,\theta) = 1$$

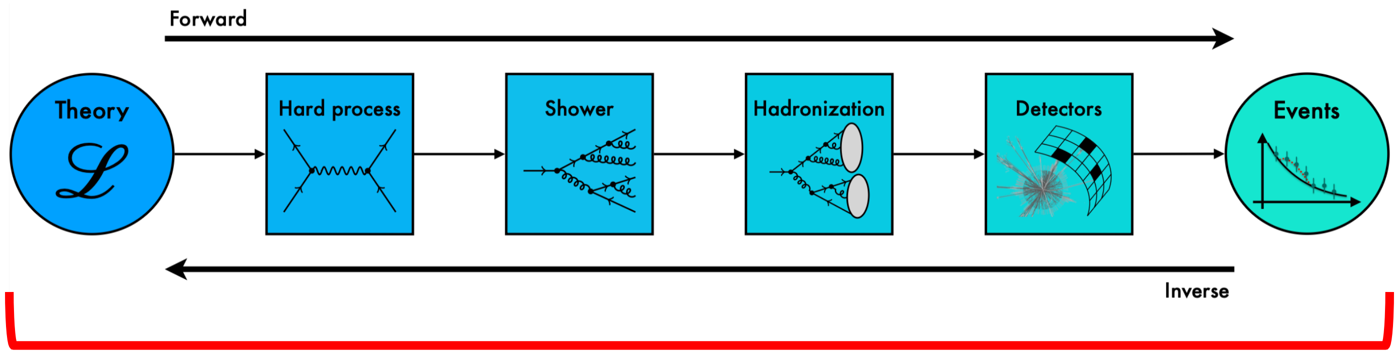
Controlling generative ML





Simulation-Based Inference

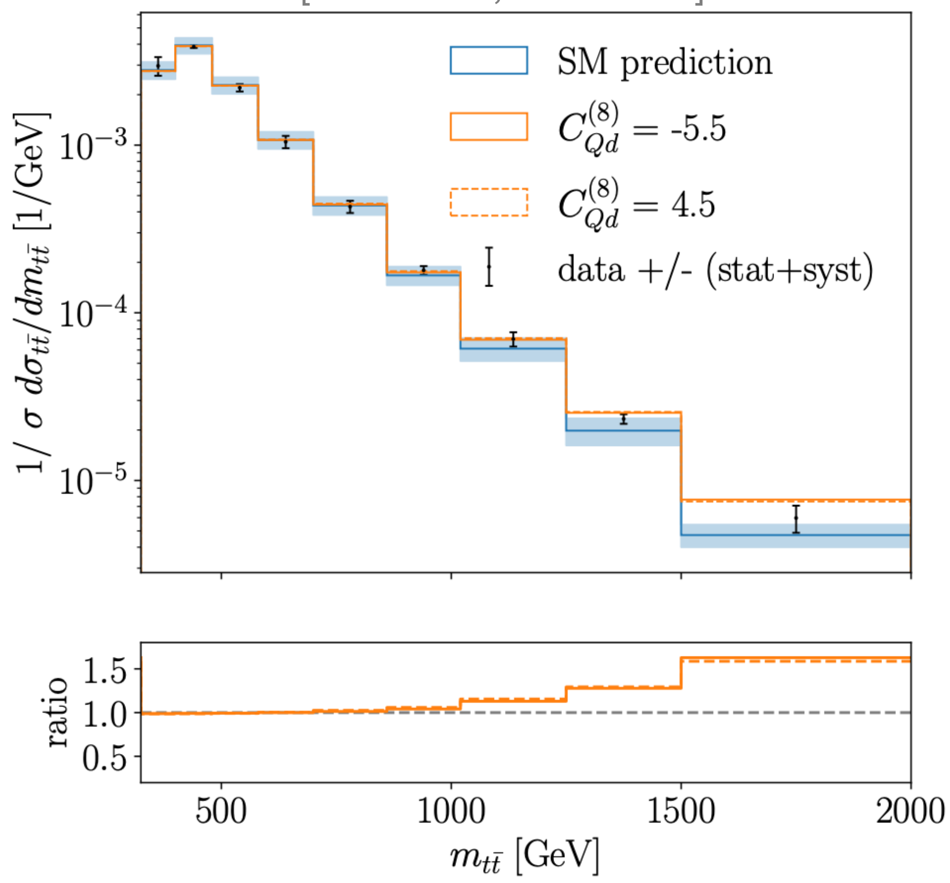
fully exploiting high-dimensional data



Classical parameter inference

- reduce dimension of phase space summary statistics →
- bin summary statistics
- compare resulting histogram to SM/BSM predictions

[Elmer et al.,2312.12502]



Advantage: humanly digestible plots

Disadvantage: loss of information

Full likelihood

phase space point → x; theory parameters → θ

- Monte-Carlo simulation chain allows us to sample full likelihood $p(x|\theta)$. But cannot directly compute it.

- Neyman-Pearson lemma: likelihood ratio $r(x|\theta,\theta_0) \equiv \dfrac{p(x|\theta)}{p(x|\theta_0)}$ is most powerful statistical test

- but we can regress to reco-level $r(x|\theta,\theta_0)$ using known parton-level $r(z_p|\theta,\theta_0)$:

$$\mathcal{L} = \left\langle \left[r(z_p|\theta,\theta_0) - \underbrace{r_\varphi(x|\theta,\theta_0)}_{\text{NN}} \right]^2 \right\rangle_{\underbrace{x,z_p \sim p(x|z_p)p(z_p|\theta);\theta\sim q(\theta)}_{\text{average over event sample}}}$$

→ unbinned multi-dimensional inference without information loss

Encoding amplitude structure

[Schöfbeck et al., 2107.10859, 2205.12976]

Theory structure for e.g. SMEFT:

$$\mathscr{L}_{\mathrm{SMEFT}} = \mathscr{L}_{\mathrm{SM}} + \sum_i \frac{c_i}{\Lambda^2} O_i \equiv \mathscr{L}_{\mathrm{SM}} + \sum_i \theta_i O_i$$

$$|\mathcal{M}(z_p|\theta)|^2 = |\mathcal{M}_{\mathrm{SM}}(z_p)|^2 + \theta_i |\mathcal{M}_i(z_p)|^2 + \theta_i \theta_j |\mathcal{M}_{ij}(z_p)|^2$$

encode into likelihood

$$R(x|\theta,\theta_0) \equiv \frac{d\sigma(x|\theta)/dx}{d\sigma(x|\theta_0)/dx} = \frac{\sigma(\theta)p(x|\theta)}{\sigma(\theta_0)p(x|\theta_0)}$$

$$R(x|\theta,\theta_0) = 1 + (\theta-\theta_0)_i R_i(x) + (\theta-\theta_0)_i(\theta-\theta_0)_j R_{ij}(x)$$

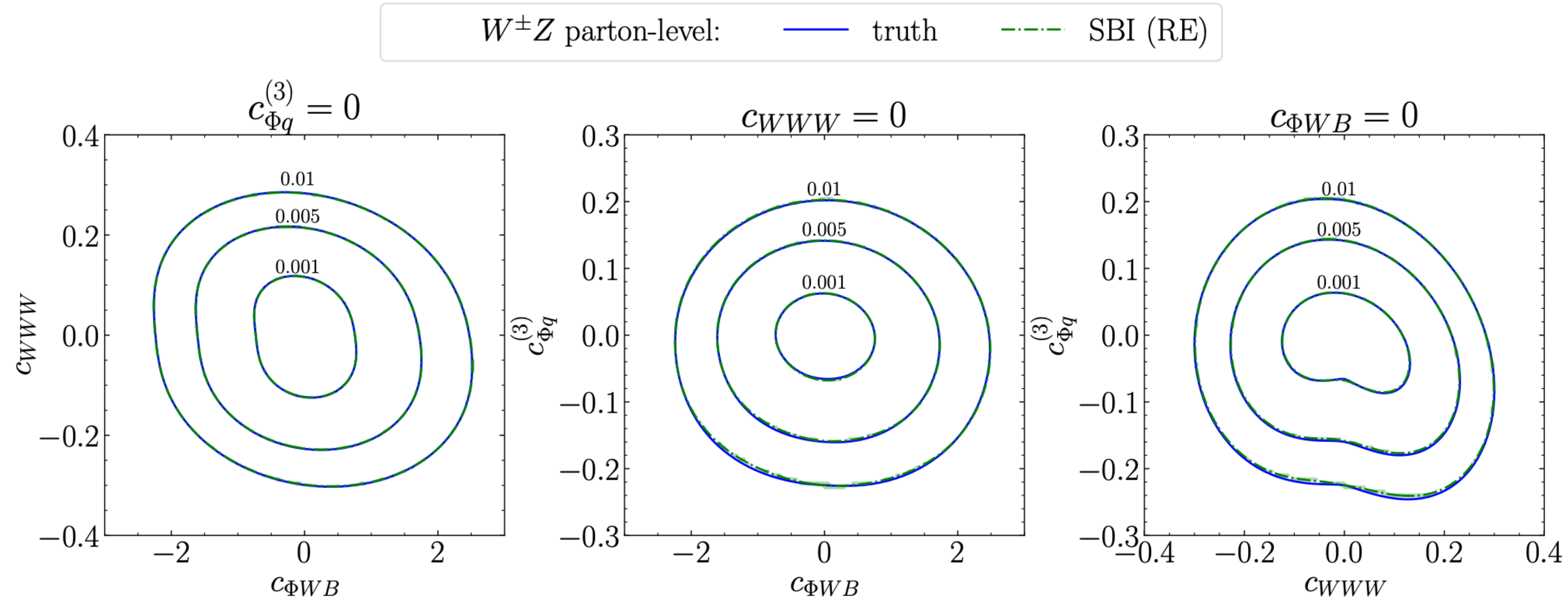
$$R_i(z_p) \equiv \frac{\partial}{\partial\theta_i} \frac{d\sigma(z_p|\theta)/dz_p}{d\sigma(z_p|\theta_0)/dz_p}\bigg|_{\theta=\theta_0} = \frac{\partial_{\theta_i}|\mathcal{M}(z_p|\theta)|^2}{|\mathcal{M}(z_p|\theta_0)|^2}\bigg|_{\theta_0}$$

$$R_{ij}(z_p) \equiv \frac{\partial^2}{\partial\theta_i\partial\theta_j} \frac{d\sigma(z_p|\theta)/dz_p}{d\sigma(z_p|\theta_0)/dz_p}\bigg|_{\theta=\theta_0} = \frac{\partial_{\theta_i}\partial_{\theta_j}|\mathcal{M}(z_p|\theta)|^2}{|\mathcal{M}(z_p|\theta)|^2}\bigg|_{\theta_0}$$

learn coefficients $R_{i,ij}$ separately → theory parameter dependence fully factored out

Parton-level cross-check: $W^{\pm}Z$ production

- consider effects of three SMEFT operators

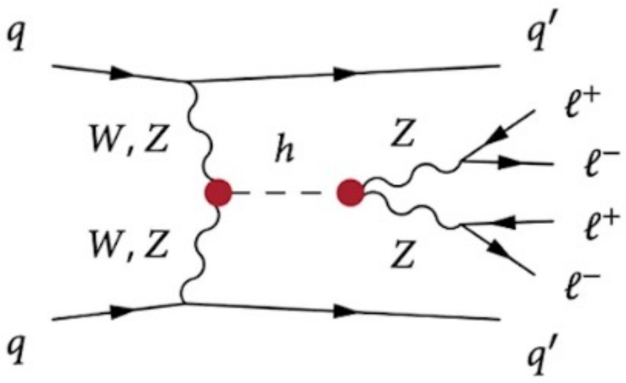


[HB et al.,2410.07315]

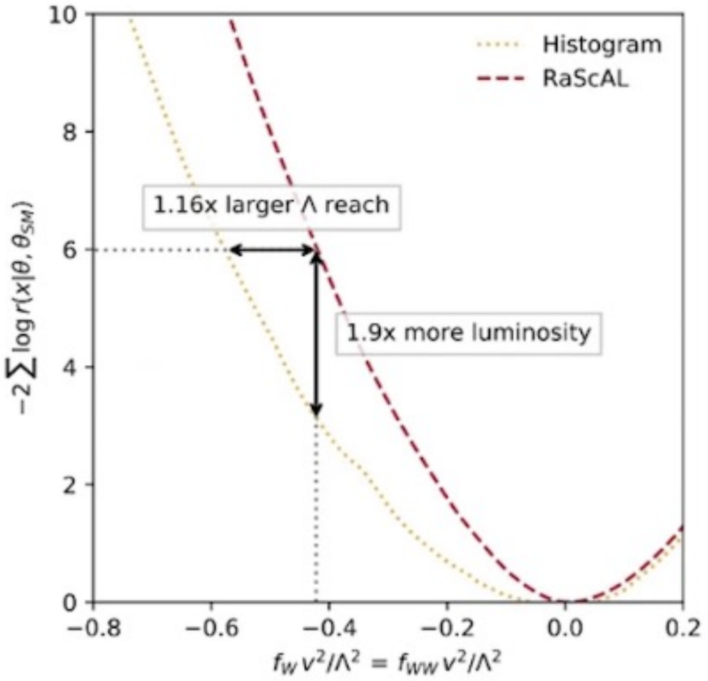
almost perfectly learns high-dimensional likelihood

Reco-level: VBF with $H \to 4\ell$

[Brehmer et al., 1805.00013]

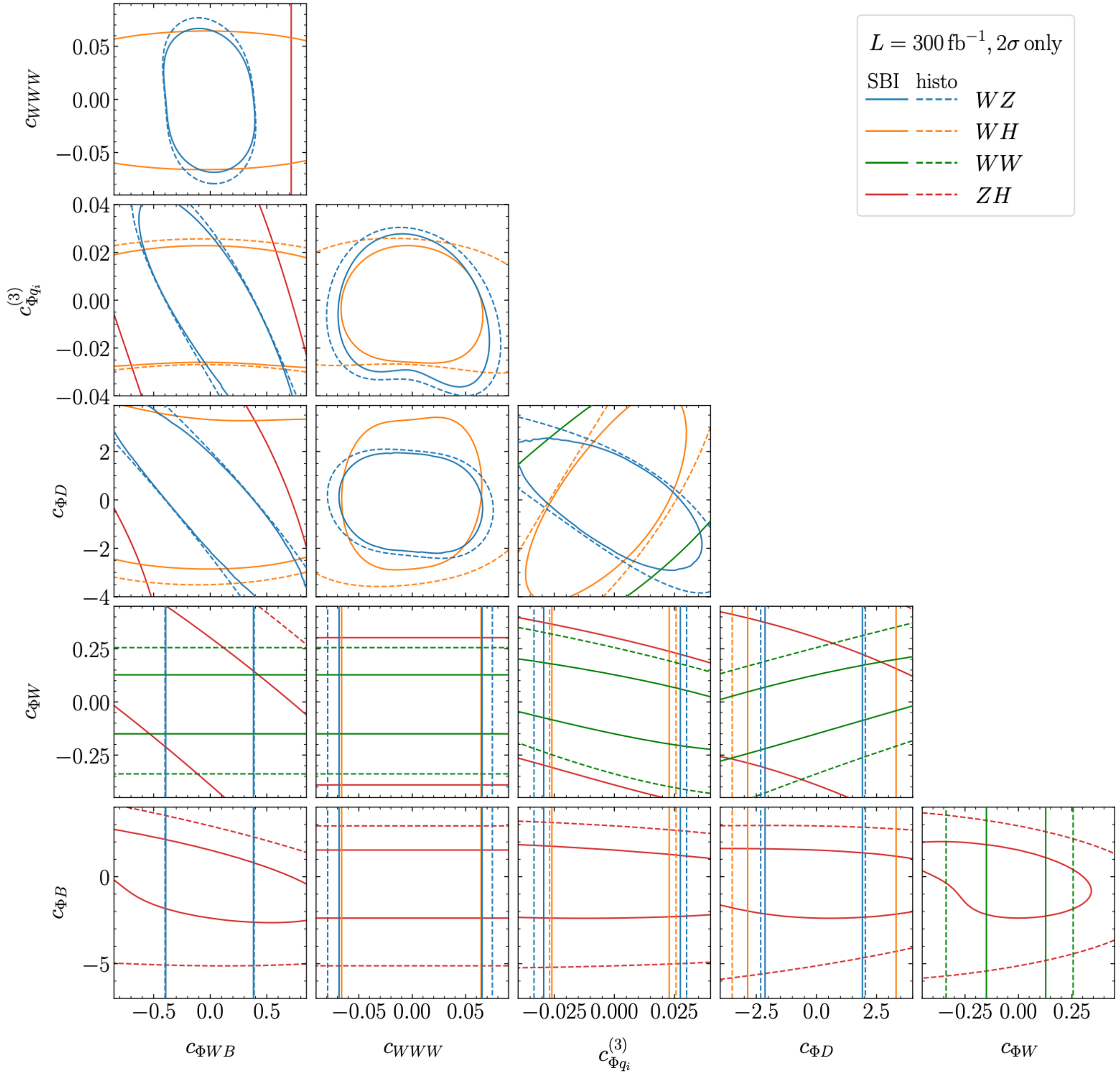


$$\mathcal{L} = \mathcal{L}_{\rm SM} + \boxed{\frac{f_W}{\Lambda^2}} \underbrace{\frac{ig}{2}\,(D^\mu\phi)^\dagger\,\sigma^a\,D^\nu\phi\,W^a_{\mu\nu}}_{\mathcal{O}_W} - \boxed{\frac{f_{WW}}{\Lambda^2}} \underbrace{\frac{g^2}{4}\,(\phi^\dagger\phi)\,W^a_{\mu\nu}\,W^{\mu\nu\,a}}_{\mathcal{O}_{WW}}$$

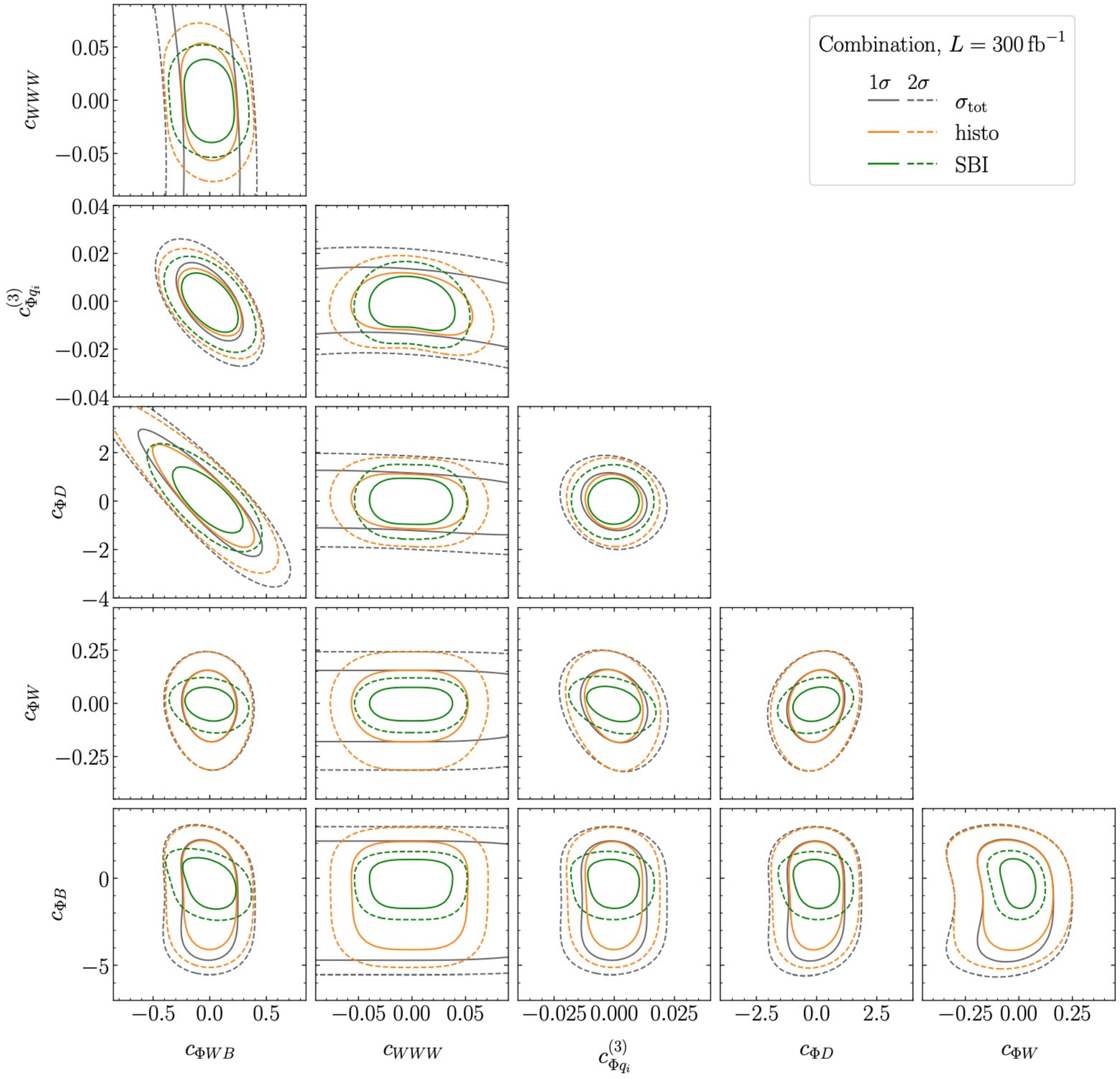


➡ Huge potential to improve sensitivity of a wide variety of measurements/searches

But is SBI also viable in a realistic analysis including uncertainties etc.?



$L = 300\,\mathrm{fb}^{-1}, 2\sigma$ only
SBI histo
WZ
WH
WW
ZH
c_{WWW}
$c^{(3)}_{\Phi q_i}$
$c_{\Phi D}$
$c_{\Phi W}$
$c_{\Phi B}$
$c_{\Phi WB}$



Combination, $L = 300\,\text{fb}^{-1}$
1σ 2σ
σ_{tot}
histo
SBI
c_{WWW}
$c^{(3)}_{\Phi q_i}$
$c_{\Phi D}$
$c_{\Phi W}$
$c_{\Phi B}$
$c_{\Phi WB}$
c_{WWW}
$c^{(3)}_{\Phi q_i}$
$c_{\Phi D}$
$c_{\Phi W}$