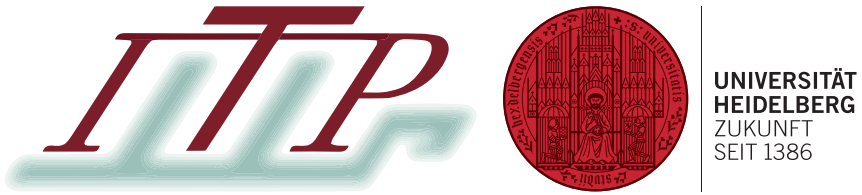
Introduction to ML for particle physics in the precision era

Henning Bahl



IRN Terascale, IPHC Strasbourg, 20.5.2025

The goal of particle physics

→ Answer the big fundamental questions!

Nature of EWSB

Neutrino masses

.....

Dark matter

Baryon asymmetry

Naturalness

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!**

The goal of particle physics

→ Answer the big fundamental questions!

Nature of EWSB

Neutrino masses

.....

Dark matter

Baryon asymmetry

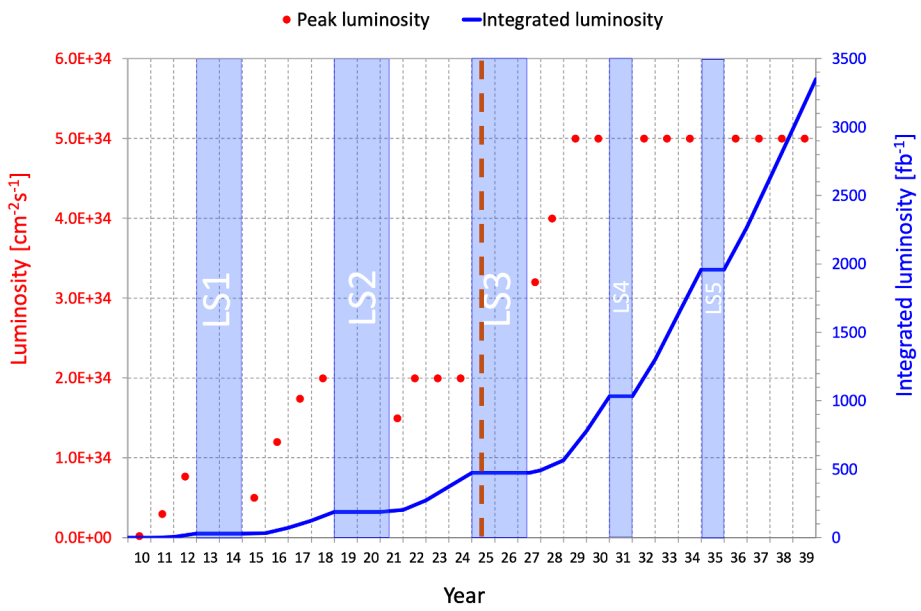
Naturalness

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

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

The challenge ahead

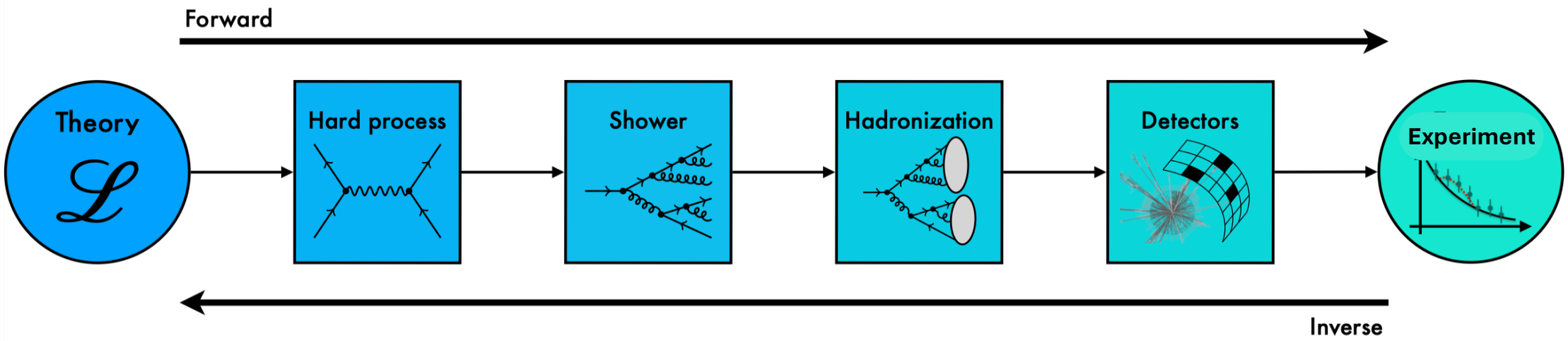
- general trend: larger-and-larger experiments collecting more-and-more data
- e.g. LHC: already enormous dataset will be further enlarged by a factor ~ 10
- costs for future experiments increasing



Fully exploit the available data!

- new analysis methods
- theory precision $\simeq$ experimental precision

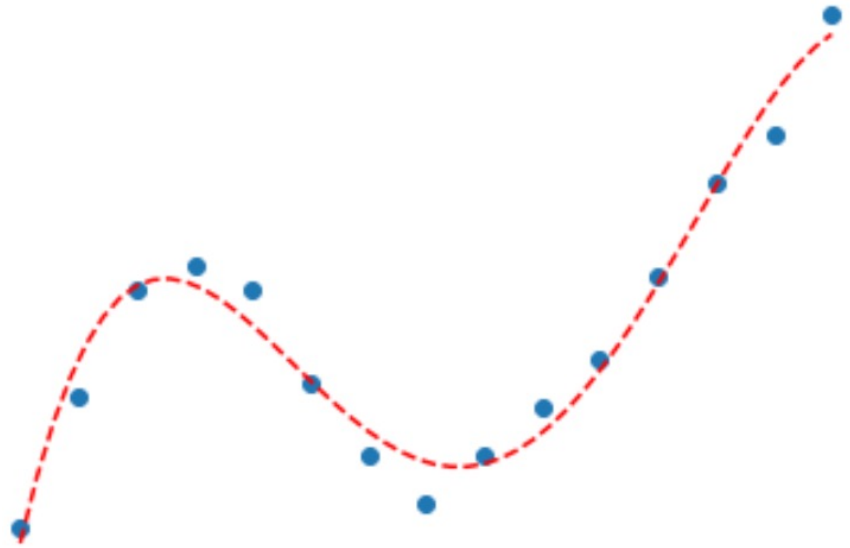
The particle physics workflow



ML can help with each of these steps by increasing

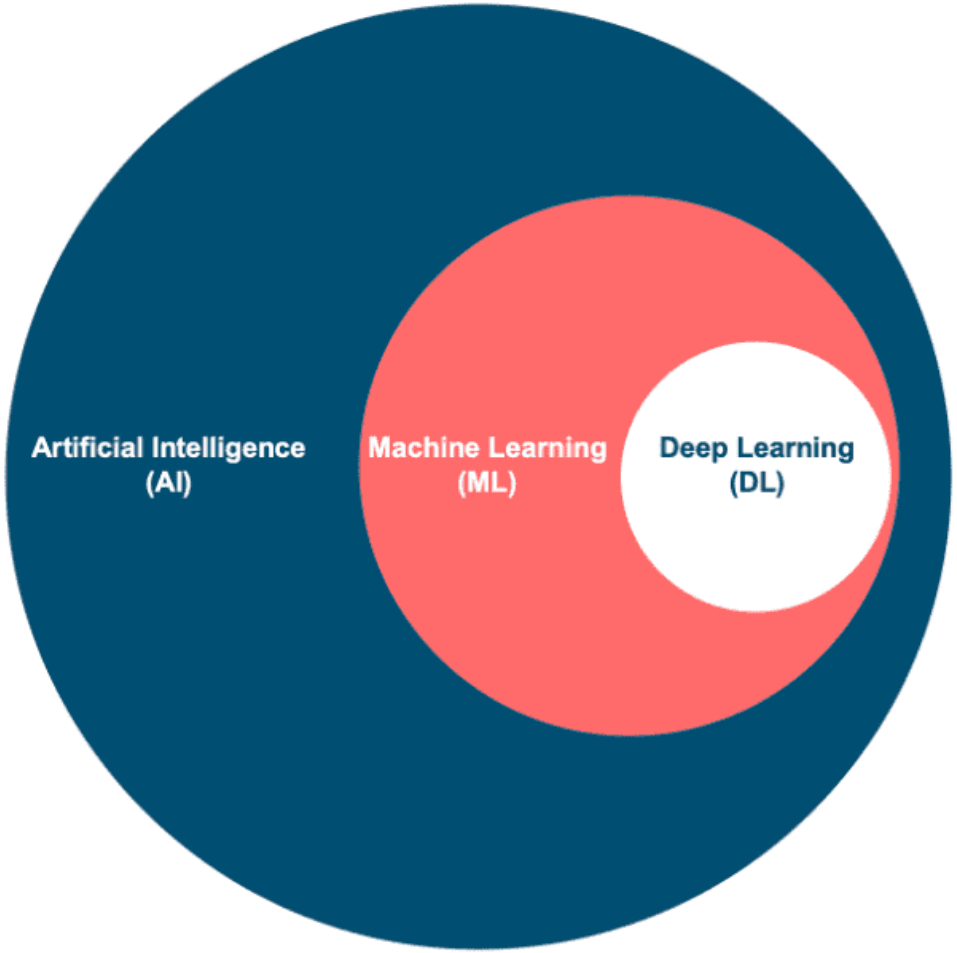
- accuracy/performance and/or
- increase speed

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



Types of ML (selection)

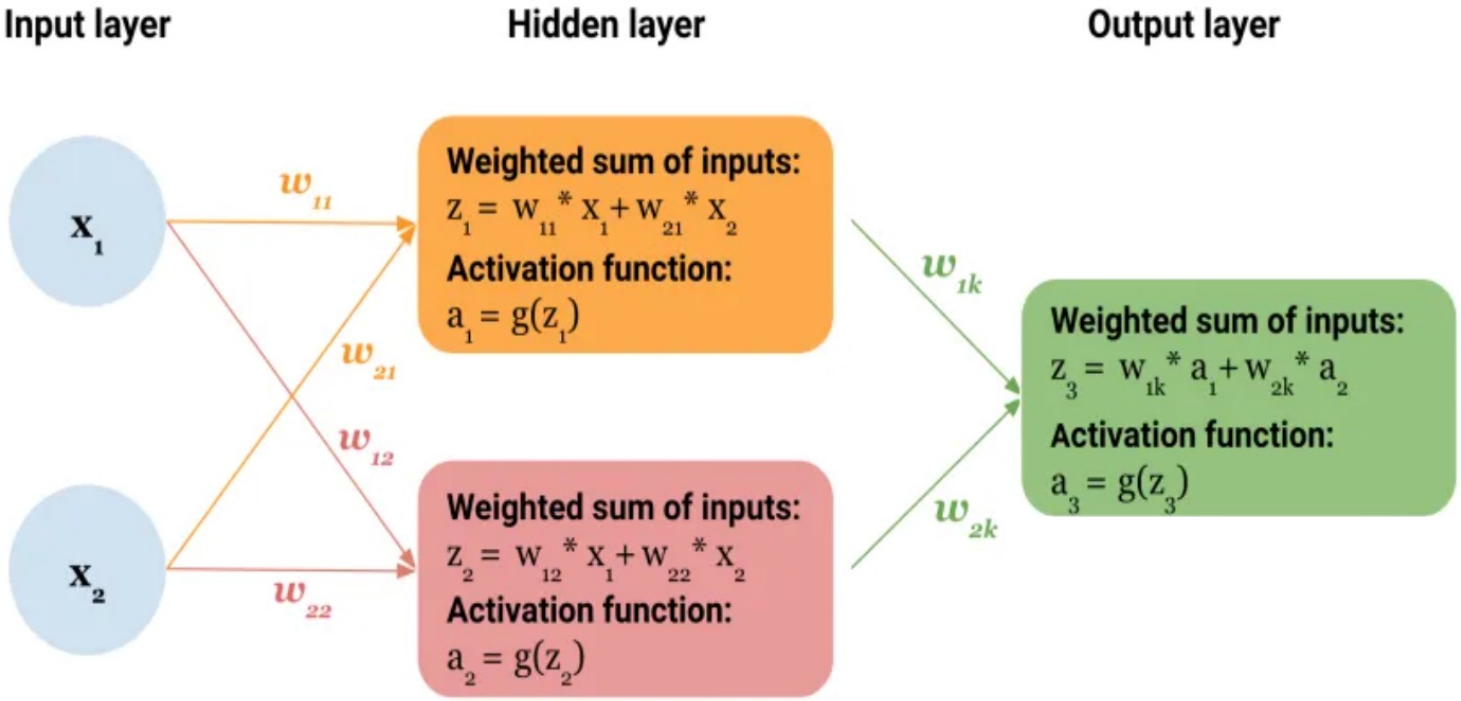
Tasks

- 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)

Neural networks



- activation introduces non-linearity (e.g. $g(x) = \max(0, x)$)
- adjust weights by minimizing loss
- large enough network can in principle approximate any function

ML workflow

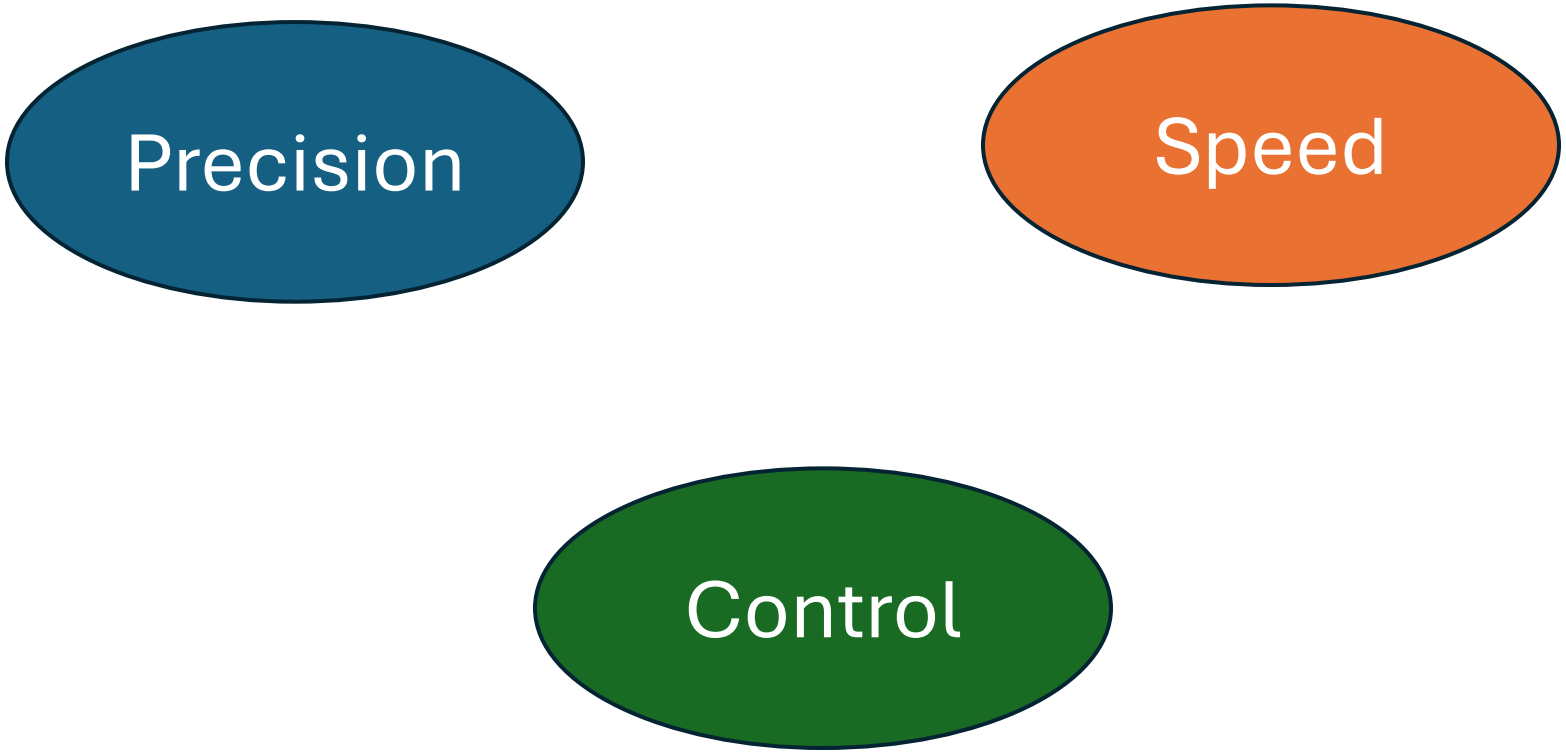
1. define the problem
2. collect and preprocess the dataset
3. define your ML model
4. training
5. evaluation

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, …*

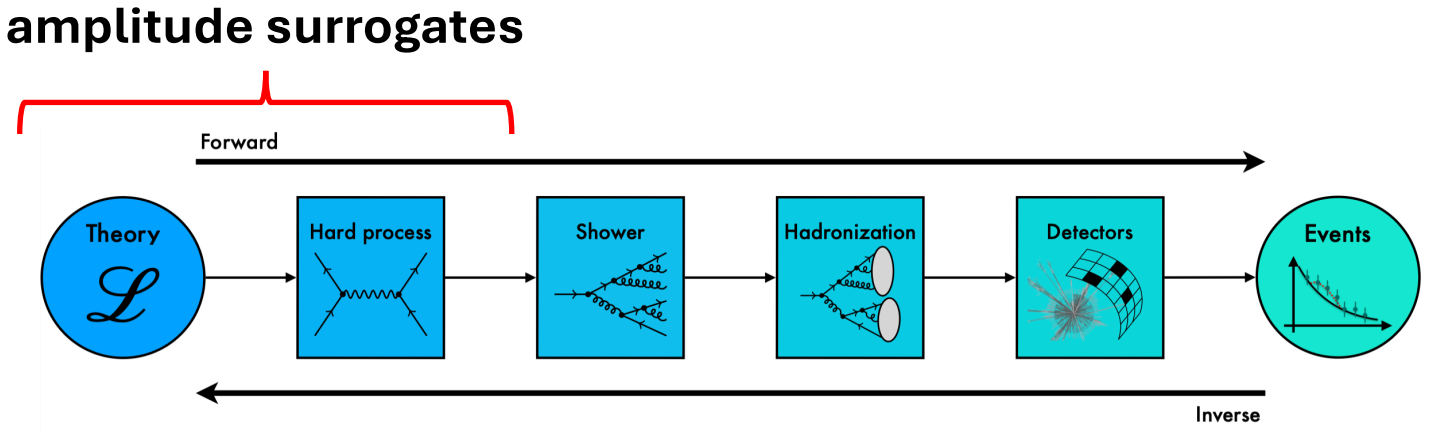
ML for particle physics — requirements



Precision ML with uncertainties

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

— George Box (adapted)



Amplitude surrogates

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

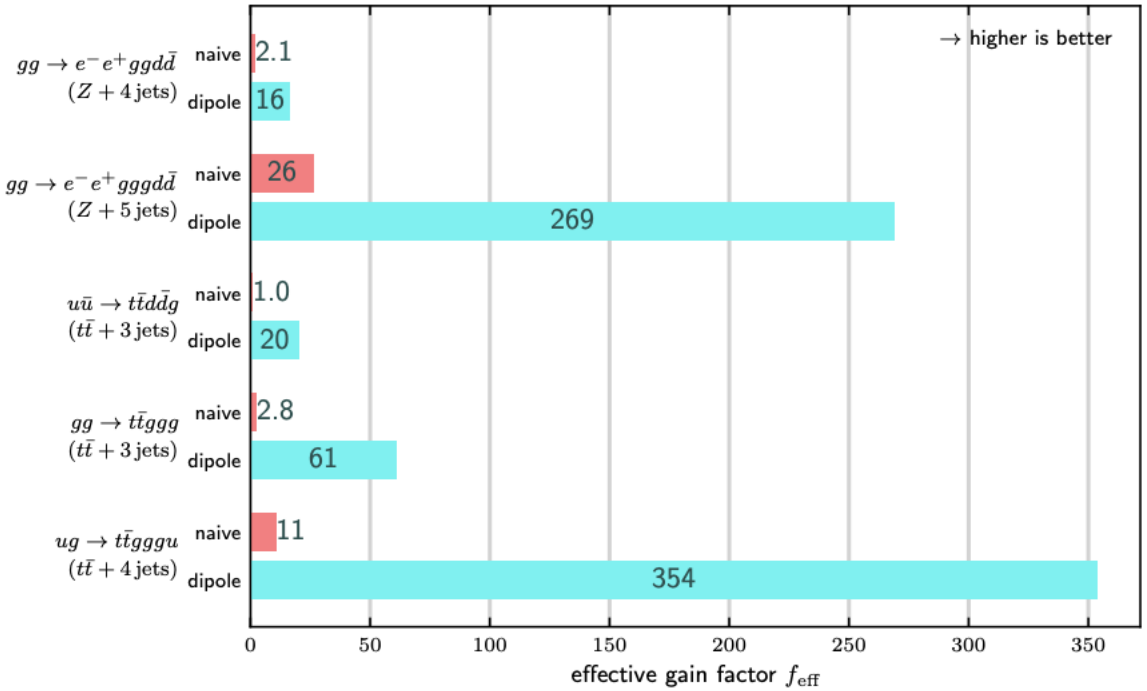
- idea:
  - generate small training sample using full analytic expression
  - train a NN to approximate $|\mathcal{M}|^2$
  - generate events using NN surrogate, which is much faster to evaluate

→ Fast high-precision event generation

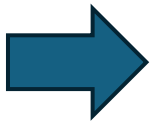
Does it work?

[Janßen et al.,2301.13562]

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



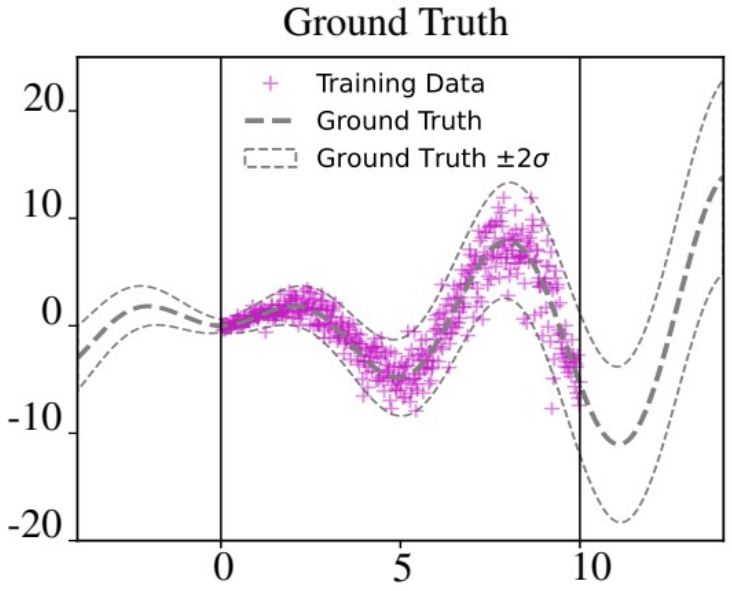
	SHERPA default			with dipole-model surrogate				
Process	t_{ME}[ms]	t_{PS}[ms]	ϵ_{full}	t_{surr}[ms]	$x_{\max}$	$\epsilon_{\text{1st,surr}}$	$\epsilon_{\text{2nd,surr}}$	f_{eff}
$gg \to e^- e^+ ggd\bar{d}$	54	0.40	1.411 %	0.14	2.6	1.418 %	39 %	16
$gg \to e^- e^+ gggd\bar{d}$	16 216	5.70	0.076 %	0.20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



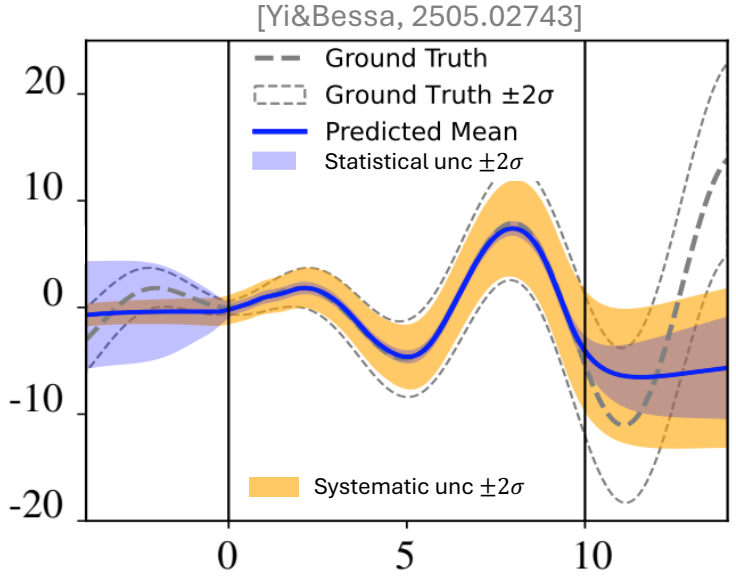
Large speed-ups possible!

Can we also control the uncertainties?

Regression with uncertainties



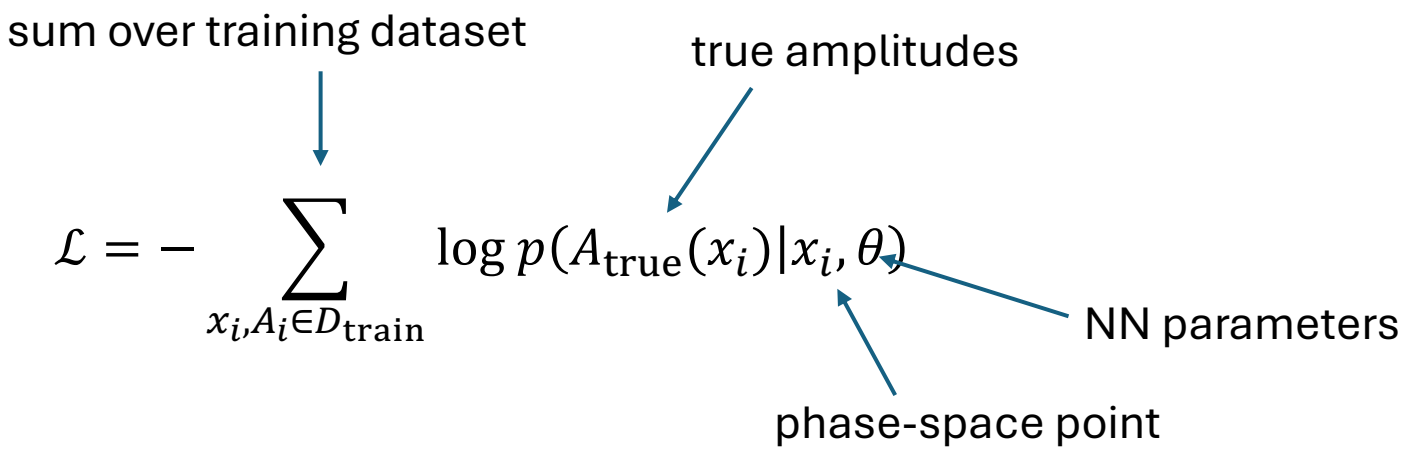
[Yi&Bessa, 2505.02743]



- statistical uncertainty ≙ lack of training data
- systematic uncertainty ≙ noise in the data, lack in model expressivity

Modelling the systematic uncertainty

- log-likelihood loss:

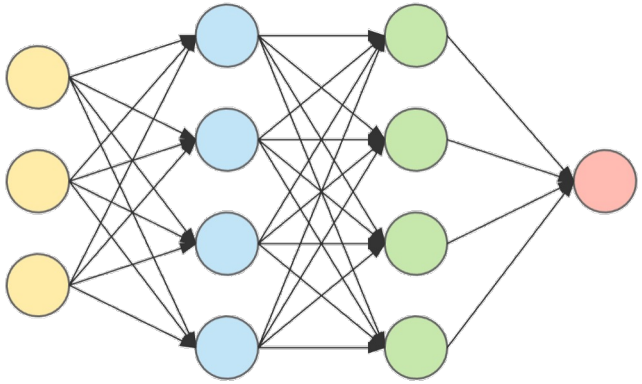


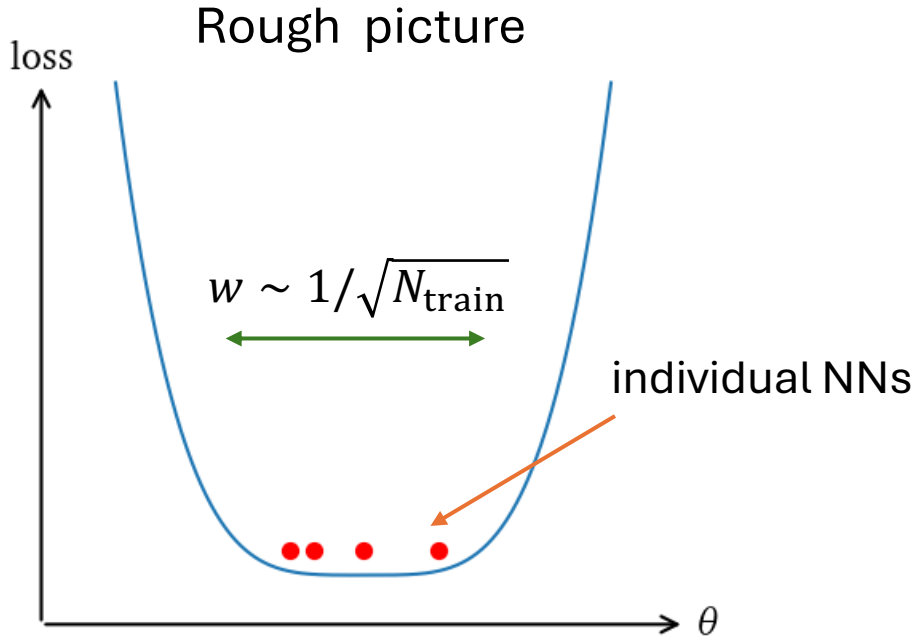
- assume Gaussian likelihood: $p(A|x,\theta) = \mathcal{N}(\overline{A}(x), \sigma^2_{\text{syst}}(x))$
- NN learns both: $\overline{A}(x)$ and $\sigma_{\text{syst}}(x)$

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

- Constant $\sigma_{\text{syst}} \rightarrow$ MSE loss

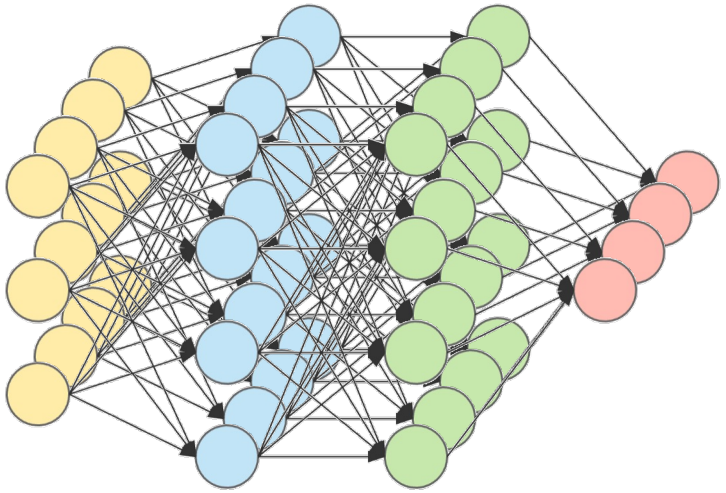
Modelling the statistical uncertainty

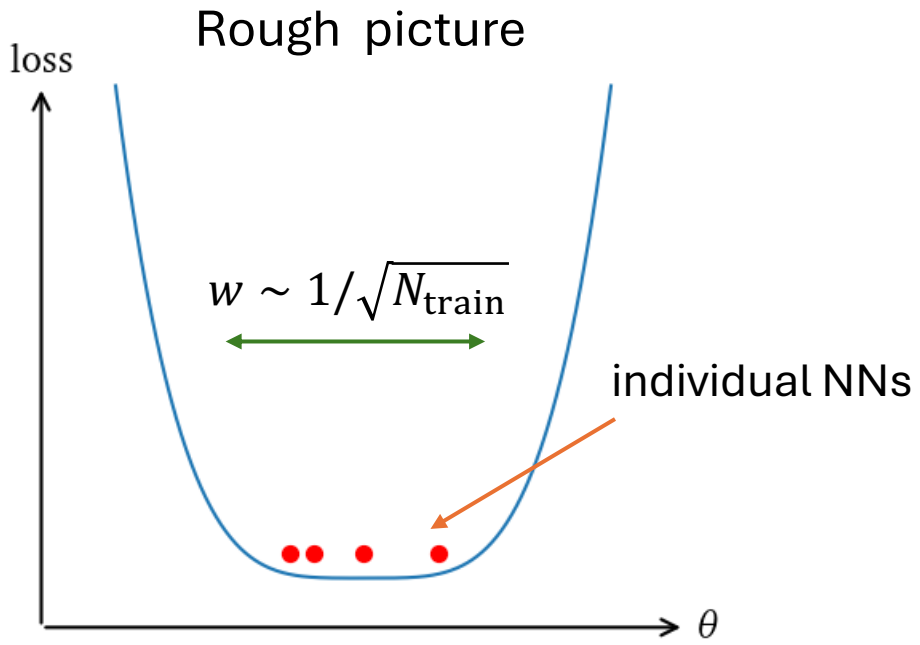




- train ensemble of networks
- each networks leads to slightly different result
- spread of network predictions ~ statistical uncertainty
- less data → higher spread

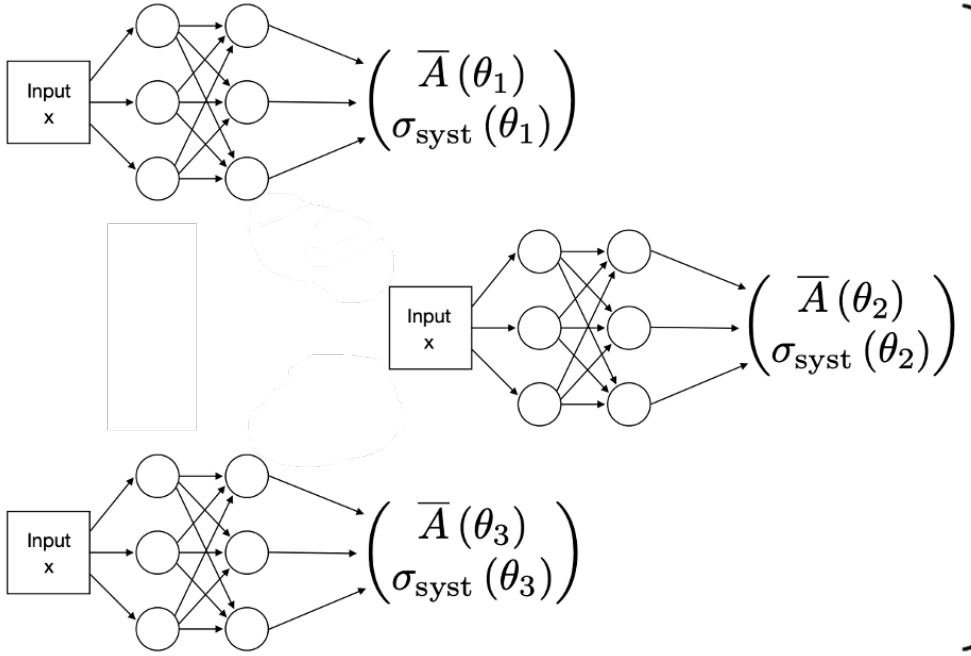
Modelling the statistical uncertainty





- 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



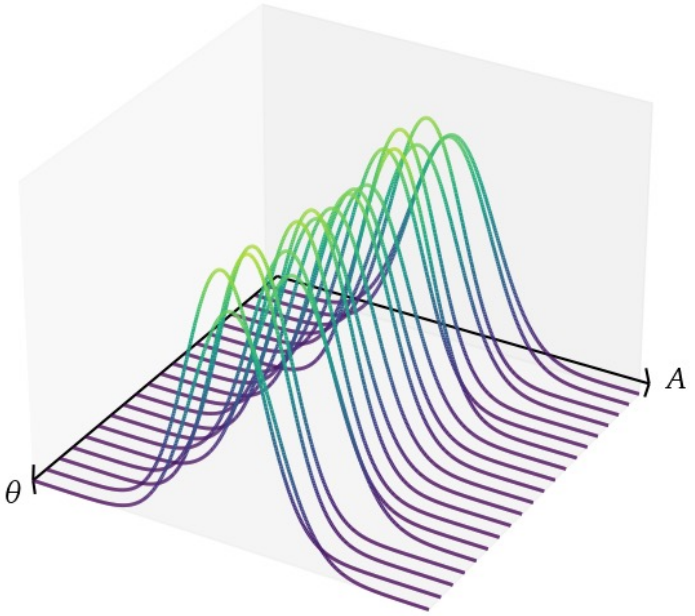
$$\begin{pmatrix} \overline{A}(\theta_1) \\ \sigma_{\text{syst}}(\theta_1) \end{pmatrix} \quad \begin{pmatrix} \overline{A}(\theta_2) \\ \sigma_{\text{syst}}(\theta_2) \end{pmatrix} \quad \begin{pmatrix} \overline{A}(\theta_3) \\ \sigma_{\text{syst}}(\theta_3) \end{pmatrix}$$

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

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

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

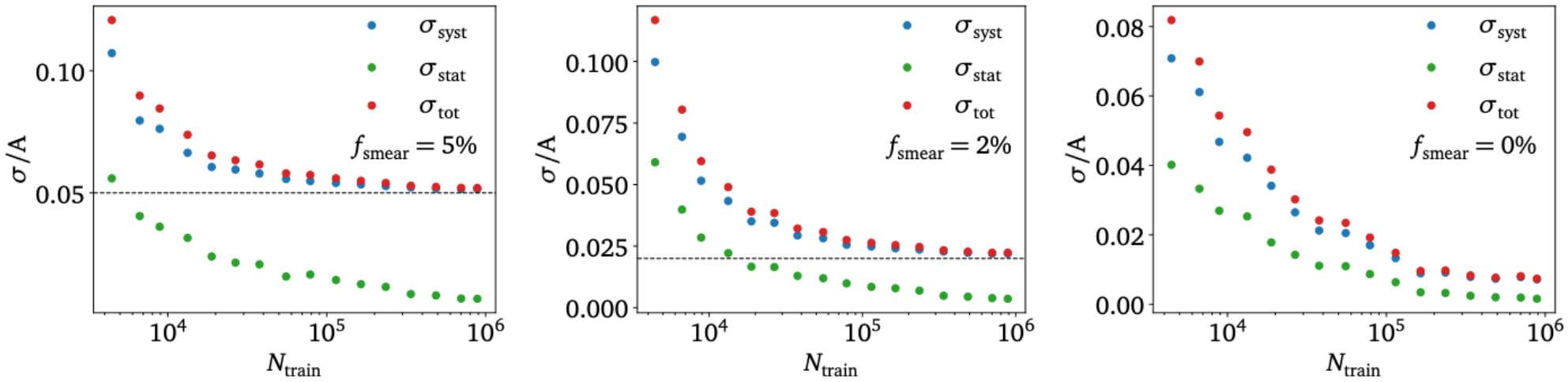
$$\sigma^2_{\text{tot}} = \sigma^2_{\text{syst}} + \sigma^2_{\text{stat}}$$



Combined learnable modelling of systematic and statistical uncertainties!

Behavior of uncertainties

[HB et al.,2412.12069]



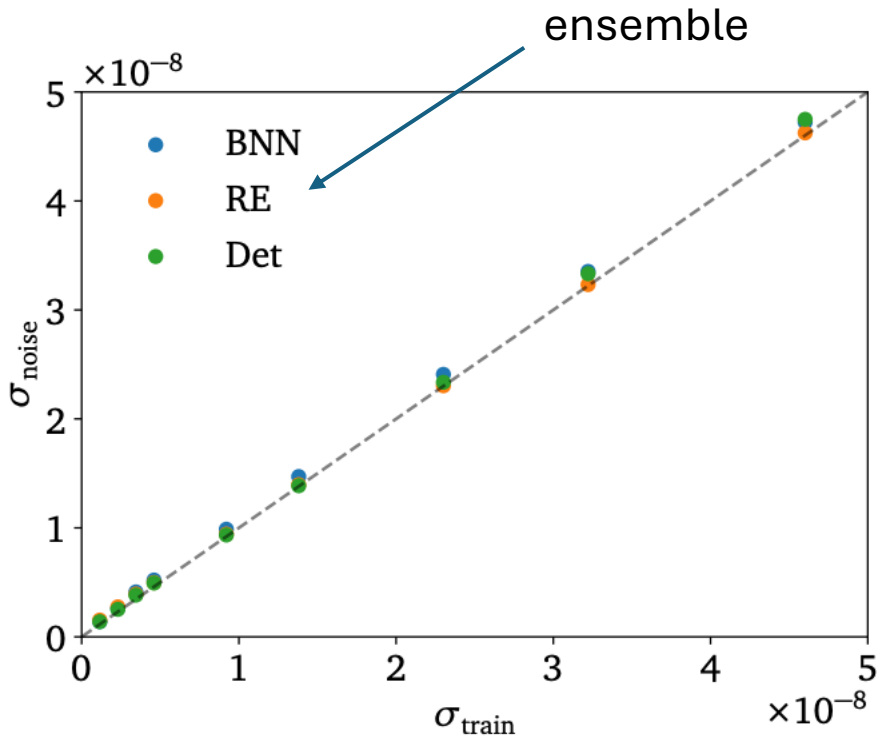
Test: apply different levels of Gaussian noise to amplitudes

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

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

Behavior of uncertainties

[HB et al.,2412.12069]



NNs can reliably extract noise level!

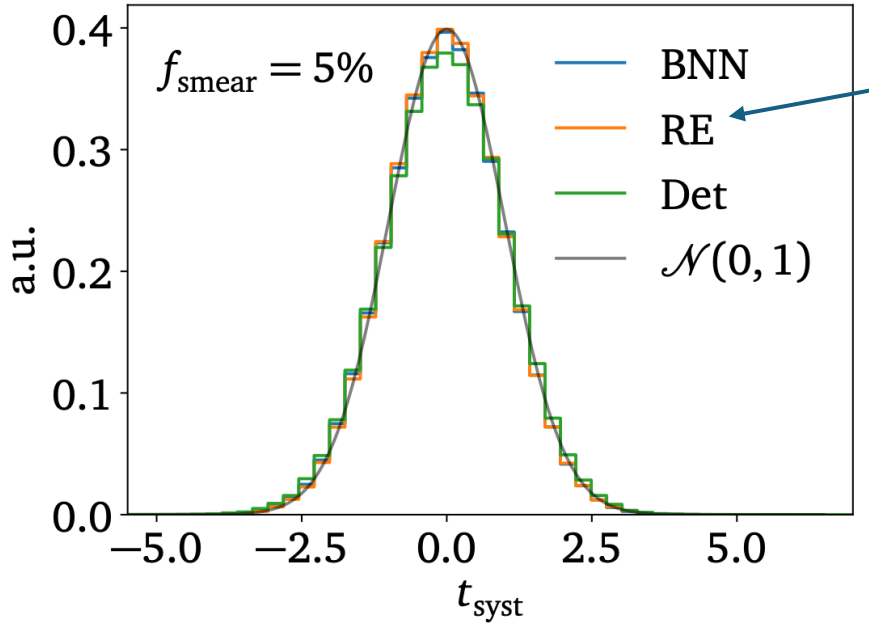
→ Are these uncertainties calibrated?

Calibration of uncertainties

- statistical uncertainties negligible for our application
- 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 $\mathcal{N}(0, 1)$

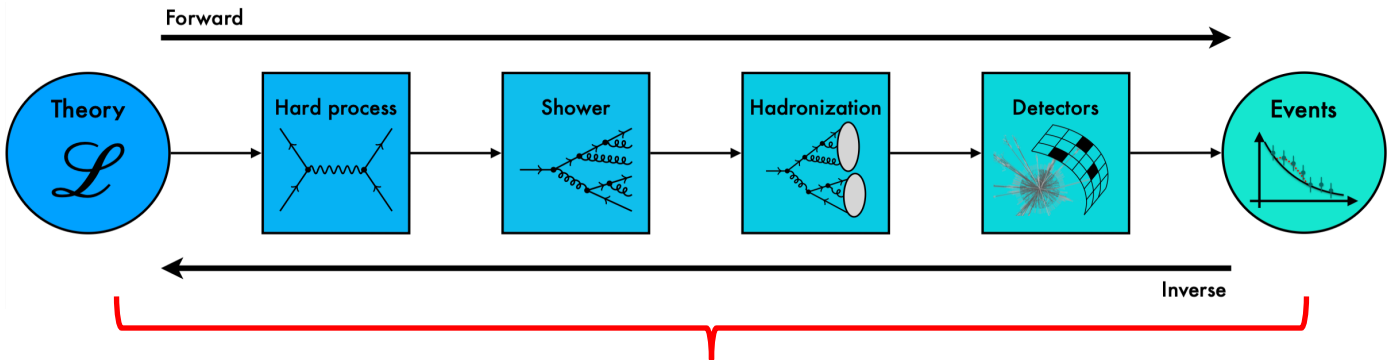


→ Almost perfectly calibration → reliable uncertainty estimate

Same techniques also applicable to all kind of other problems!

Fully exploiting the data

not only do we have a lot of data but it's also high-dimensional



Simulation-Based Inference (SBI)

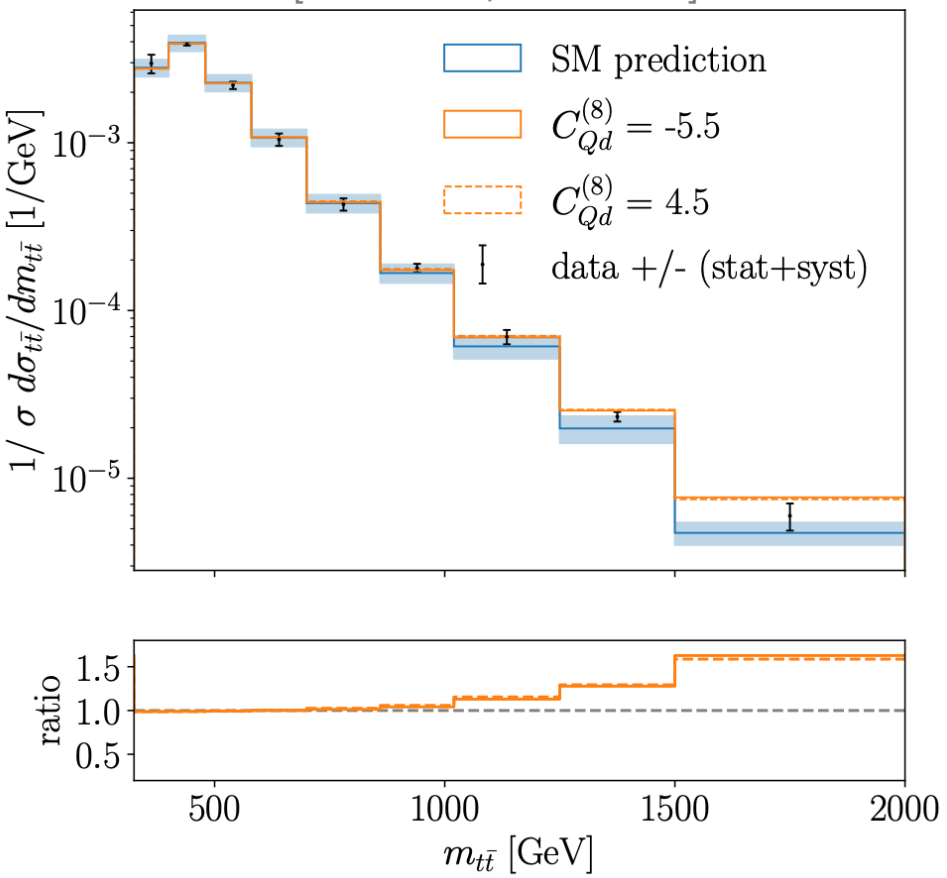
Classical parameter inference

- Reduce dimension of phase space
  → summary statistics
- Bin summary statistics
- Compare resulting histogram to SM/BSM predictions

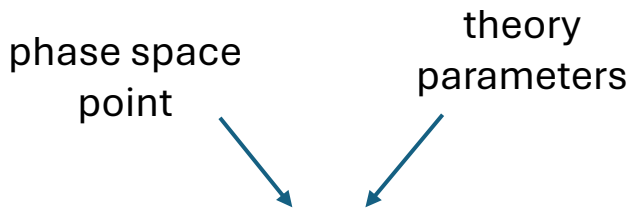
Advantage: humanly digestible plots

Disadvantage: loss of information

[Elmer et al.,2312.12502]



Full likelihood



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

- train classifier D to distinguish BSM sample ($\sim p(x|\theta)$) and SM sample ($\sim p(x|\theta_0)$) :

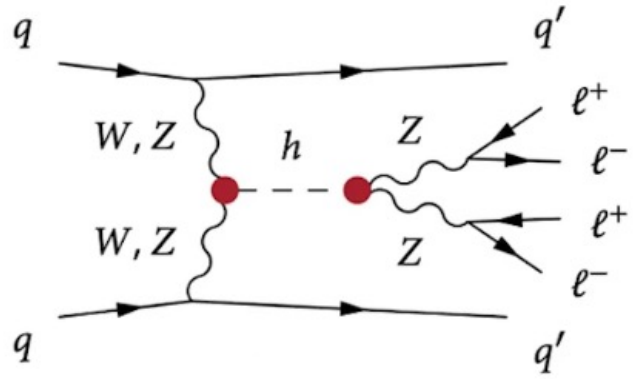
$$D_{\text{opt}}(x|\theta,\theta_0) = \frac{p(x|\theta_0)}{p(x|\theta)+p(x|\theta_0)} \rightarrow \text{likelihood ratio } \frac{p(x|\theta)}{p(x|\theta_0)} = \frac{1-D_{\text{opt}}}{D_{\text{opt}}}$$

- Neyman-Pearson lemma: likelihood ratio is most powerful statistical test

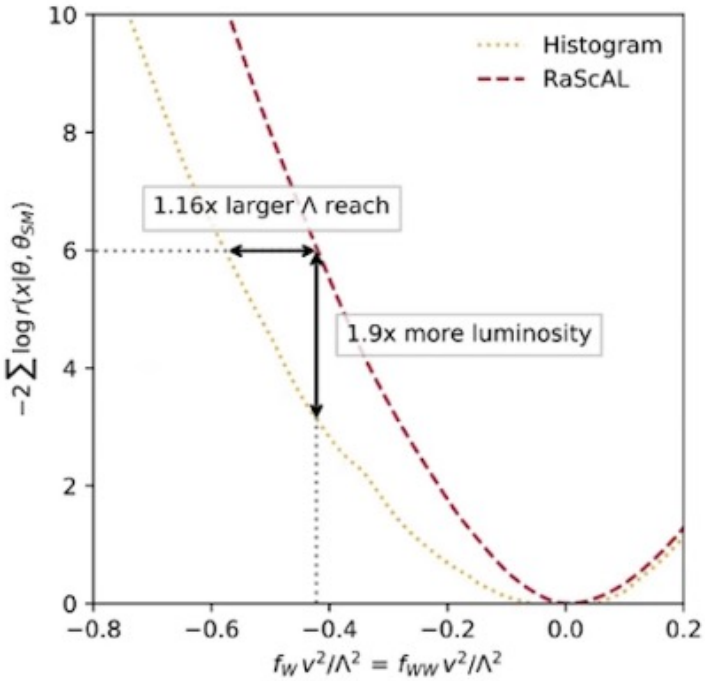
Unbinned multi-dimensional inference without information loss

Pheno examples: VBF with $H \to 4\ell$

[Brehmer et al., 1805.00013]



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



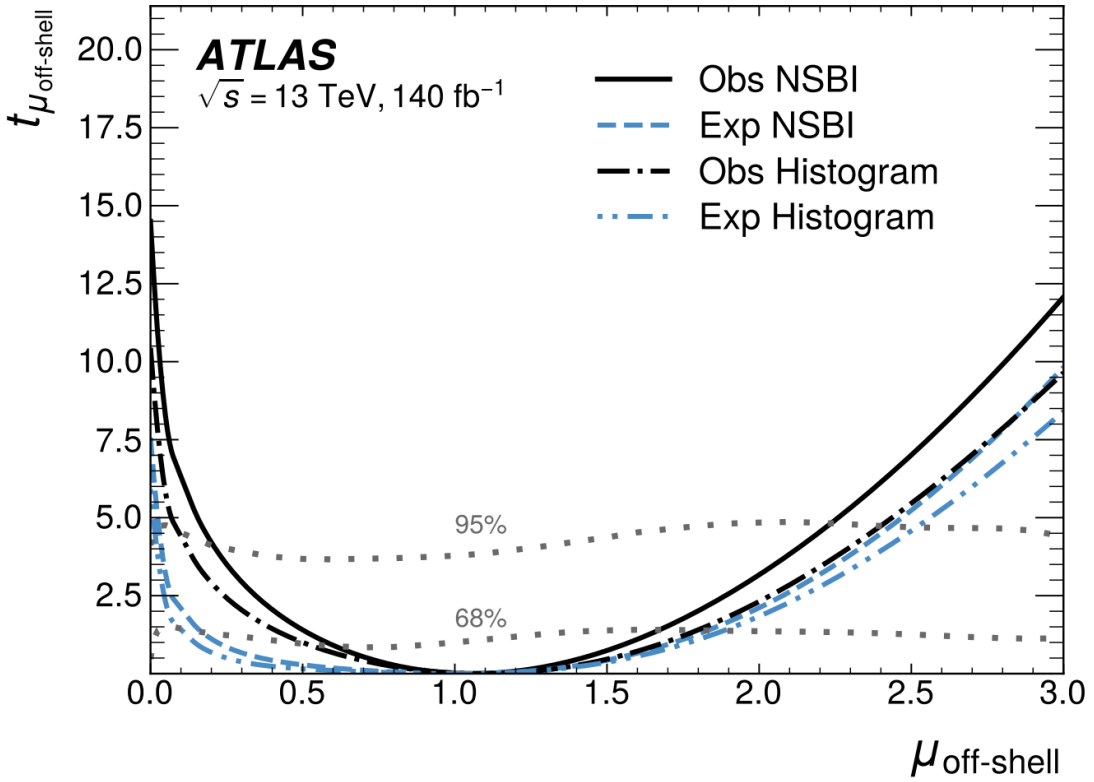
➡ Huge potential to improve sensitivity of a wide variety of measurements/searches

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

1st experimental SBI analysis

[ATLAS-CONF-2024-016]

- Goal: measure off-shell signal strength in $H \rightarrow ZZ$ channel
- Full treatment of statistical and systematic uncertainties
- Large sensitivity improvement for low $\mu_{\text{off-shell}}$



➡ Proves potential of SBI for full experimental analysis

Conclusions

Conclusions

- Particle physics is in the precision era
  → huge amounts of multidimensional data

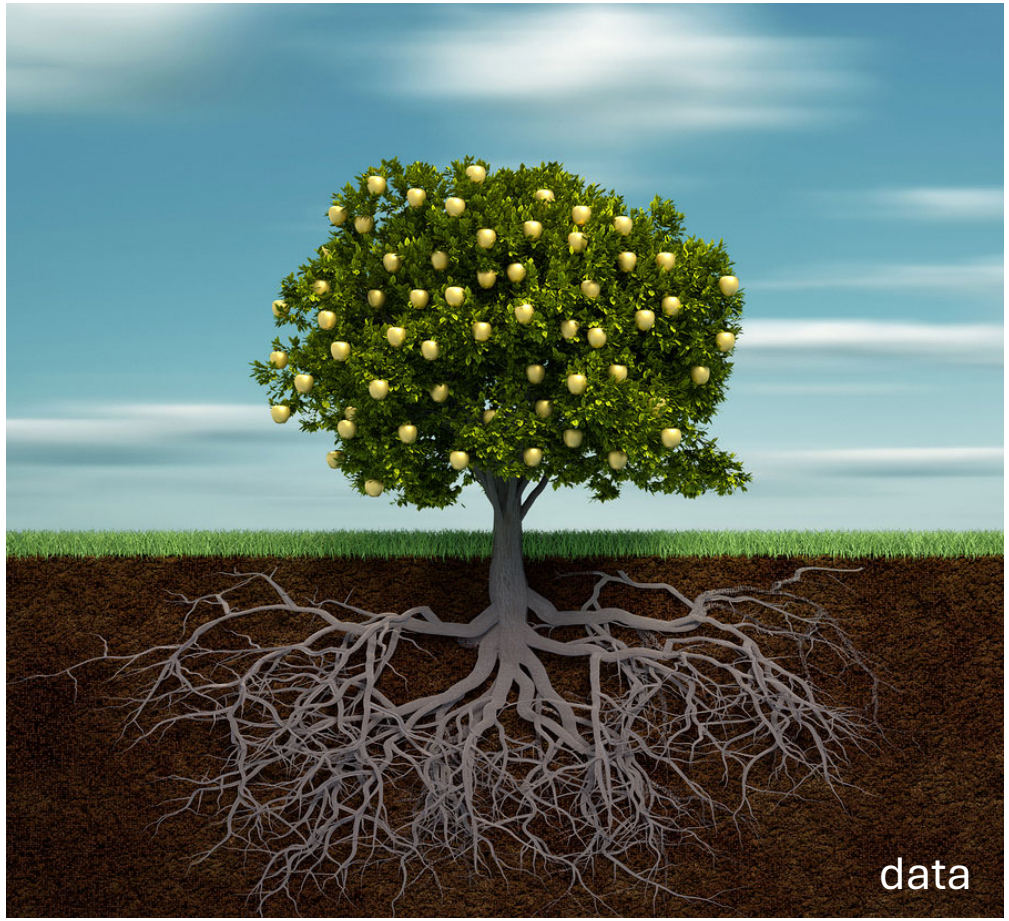
- ML methods excel in such an environment

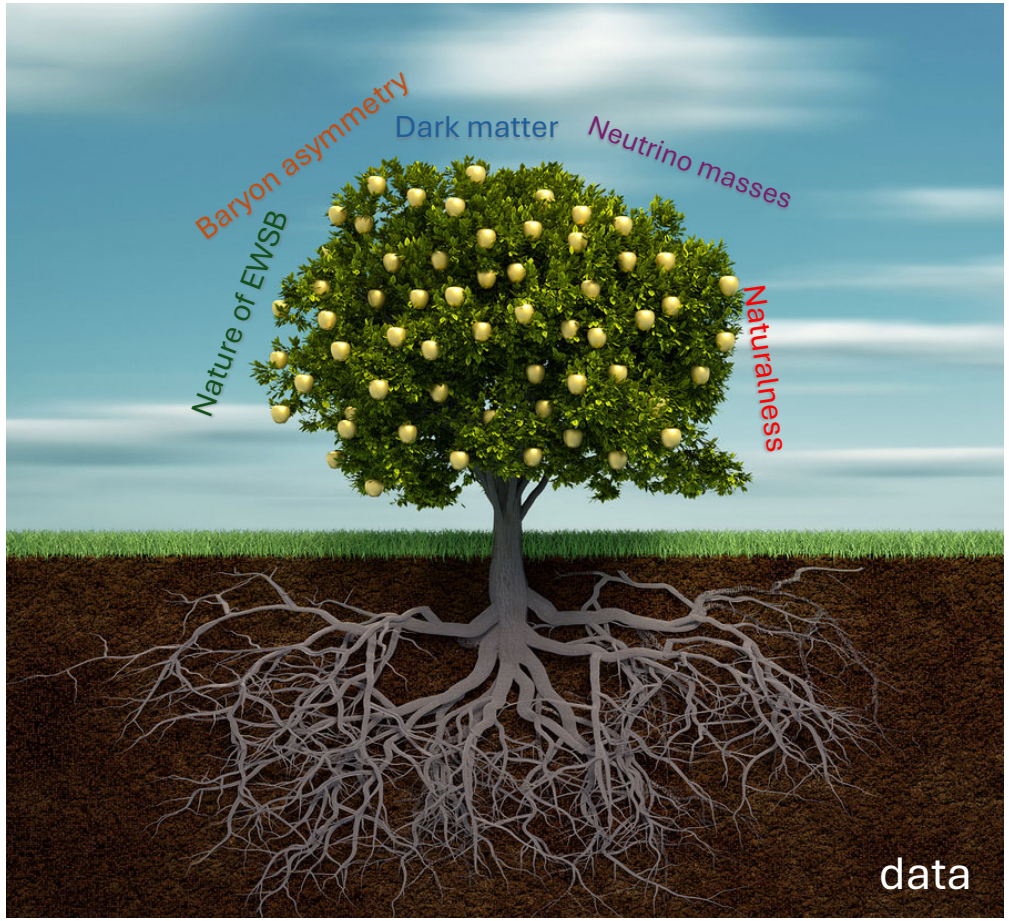
- Huge potential for increasing

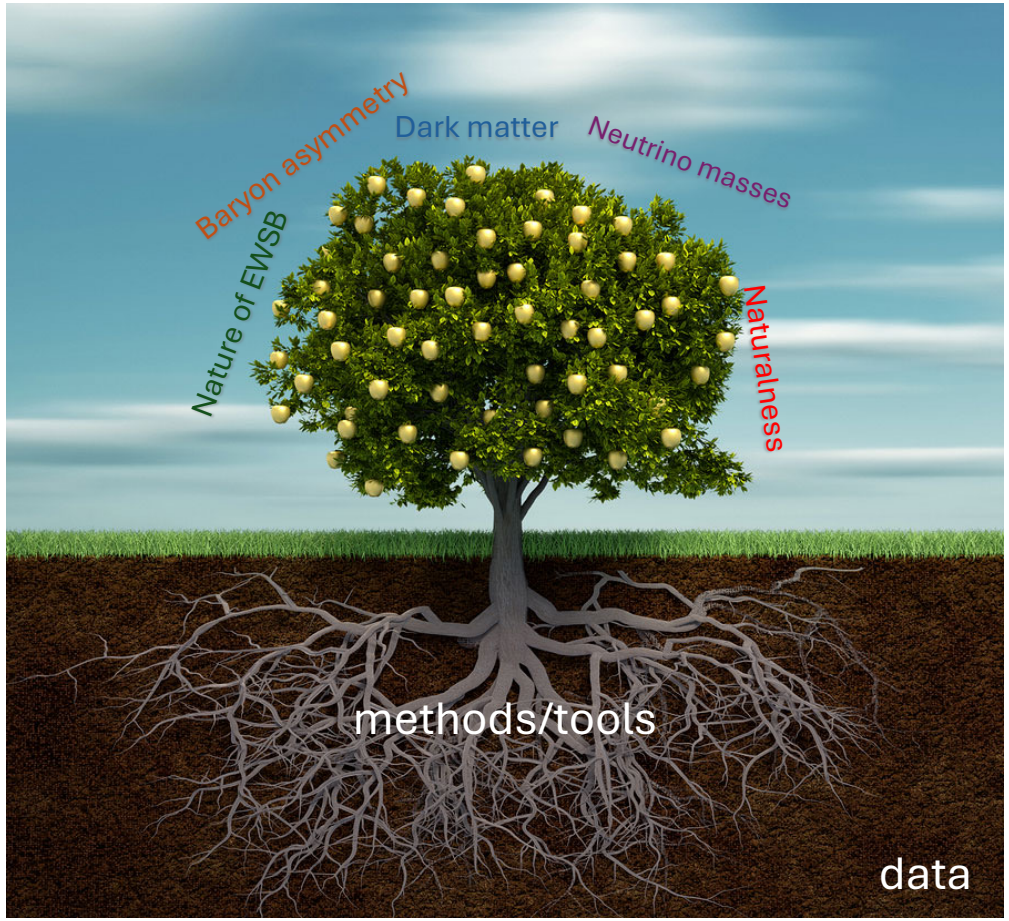
  - speed → e.g., amplitude surrogates
  - sensitivity → e.g., simultation-based inference

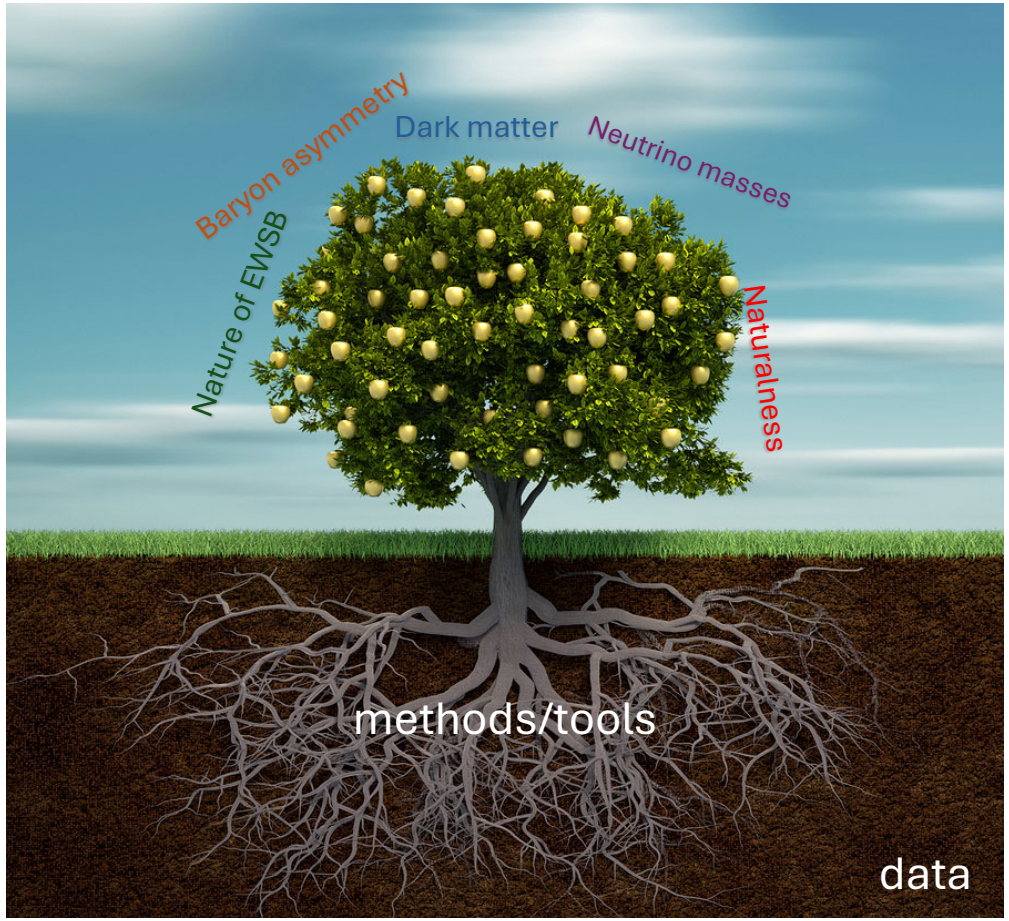
- Uncertainty-aware NNs allow for controlled modelling

→ ML methods will be indispensable for the future of particle physics


data


Baryon asymmetry
Dark matter
Neutrino masses
Nature of EWSB
Naturalness
data

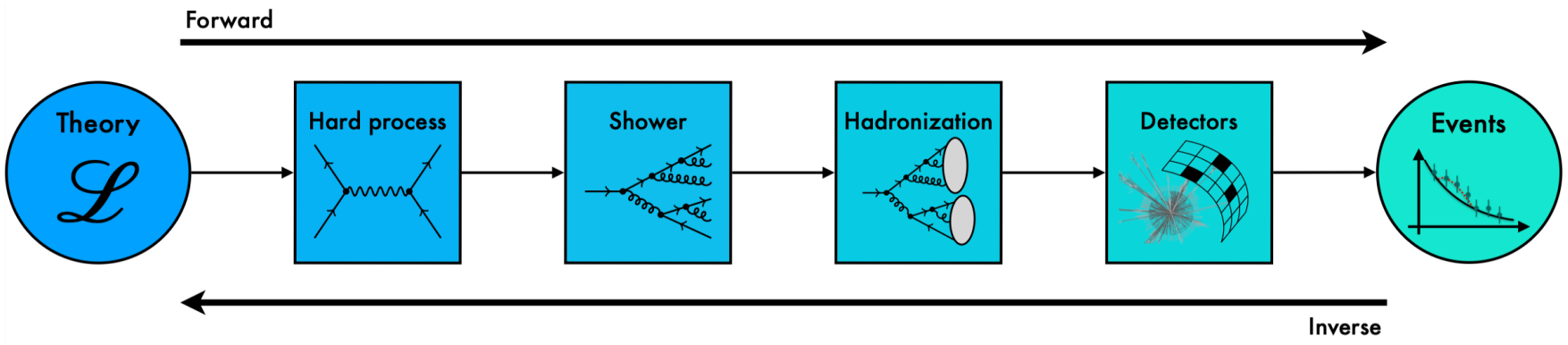

Baryon asymmetry
Dark matter
Neutrino masses
Nature of EWSB
Naturalness
methods/tools
data

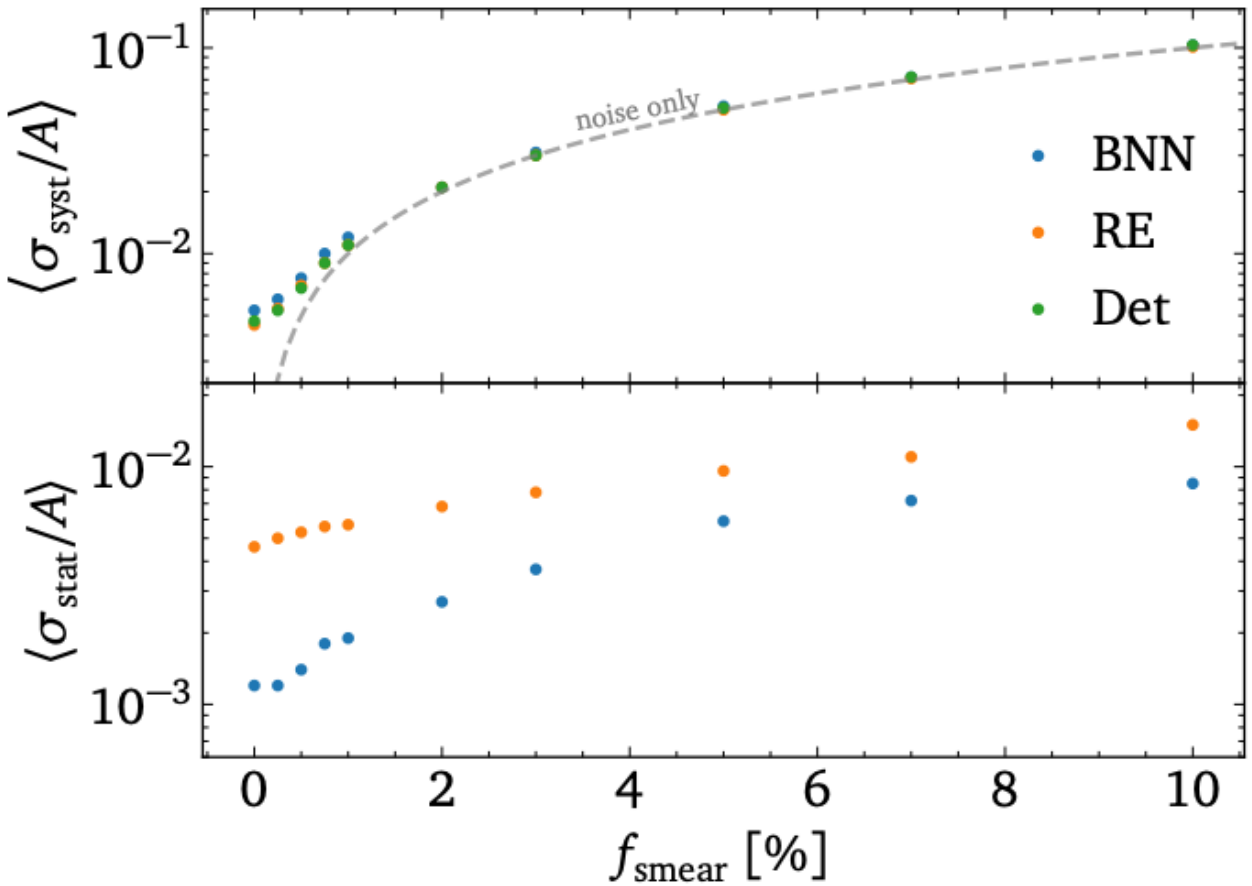


Thanks for your attention!

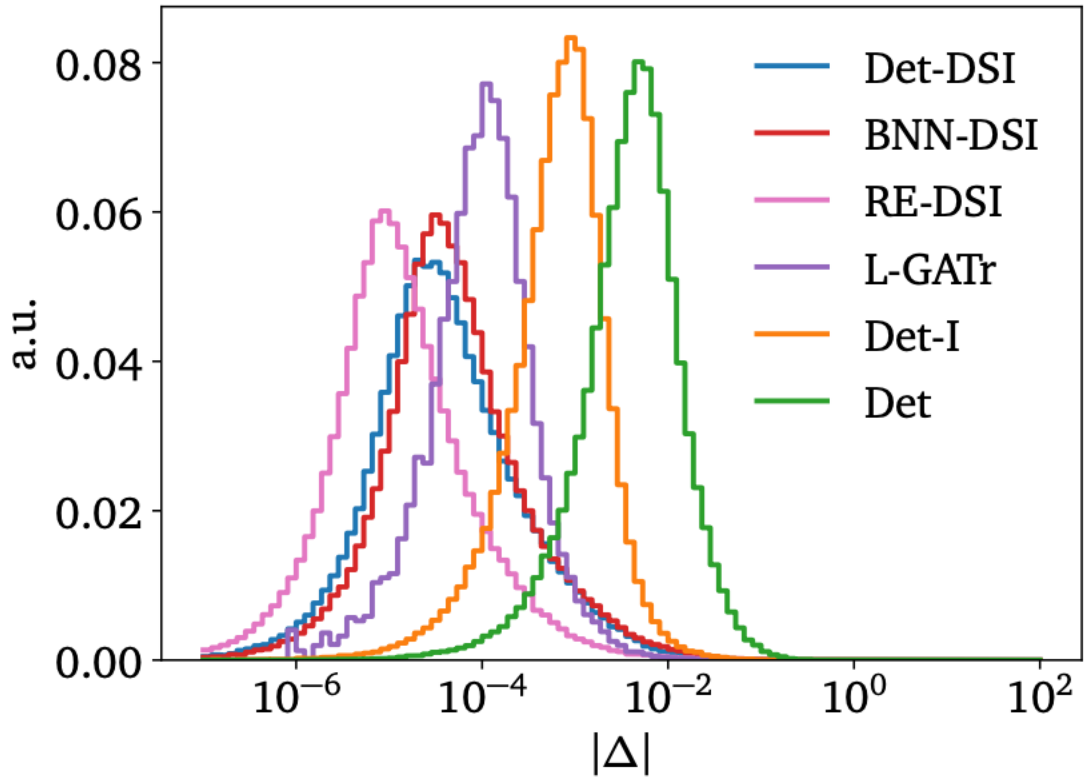
Appendix

Upcoming talks

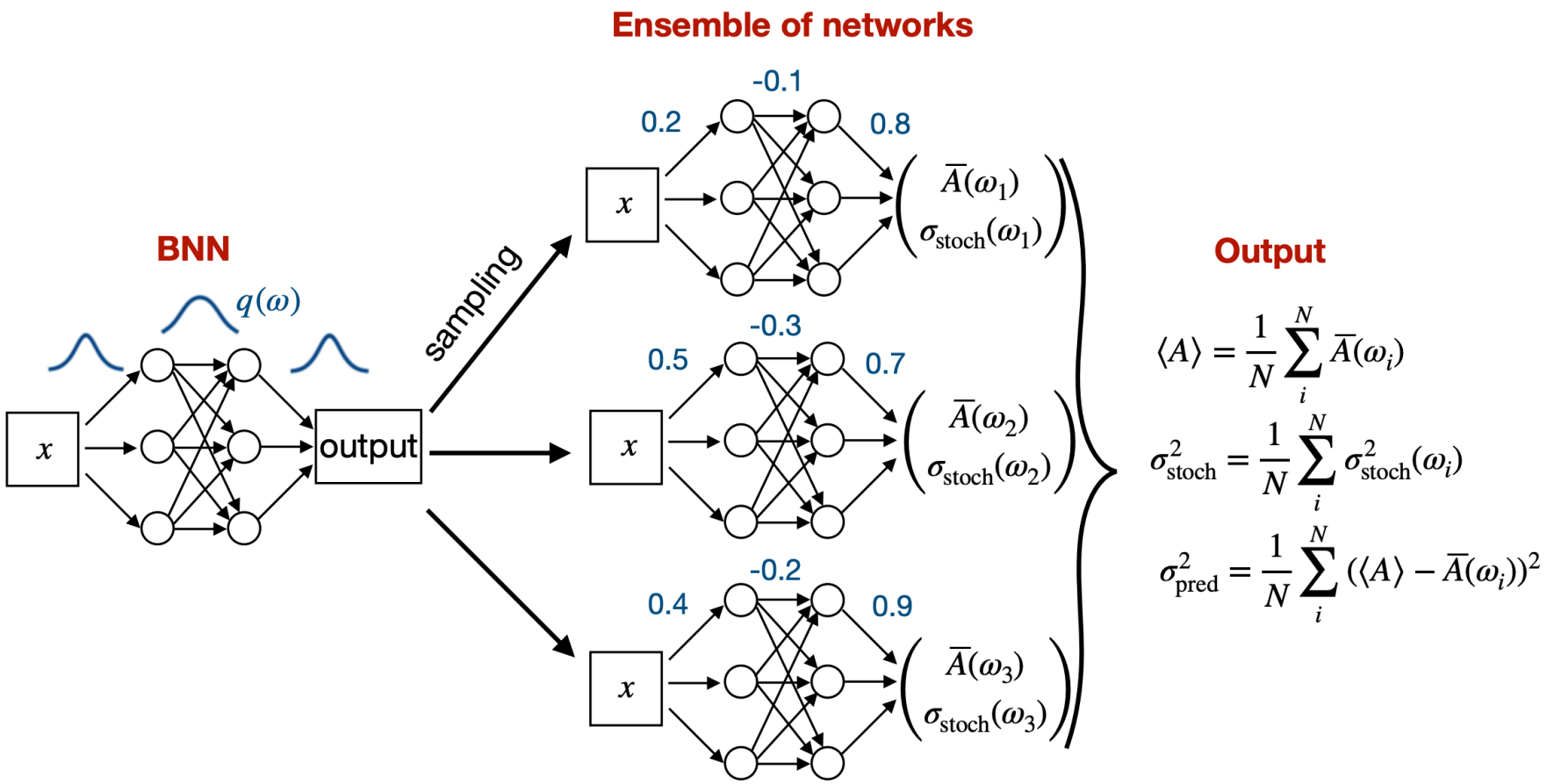



10^{-1}
10^{-2}
$\langle \sigma_{\mathrm{syst}}/A \rangle$
noise only
BNN
RE
Det
10^{-2}
10^{-3}
$\langle \sigma_{\mathrm{stat}}/A \rangle$
0
2
4
6
8
10
f_{smear} [%]

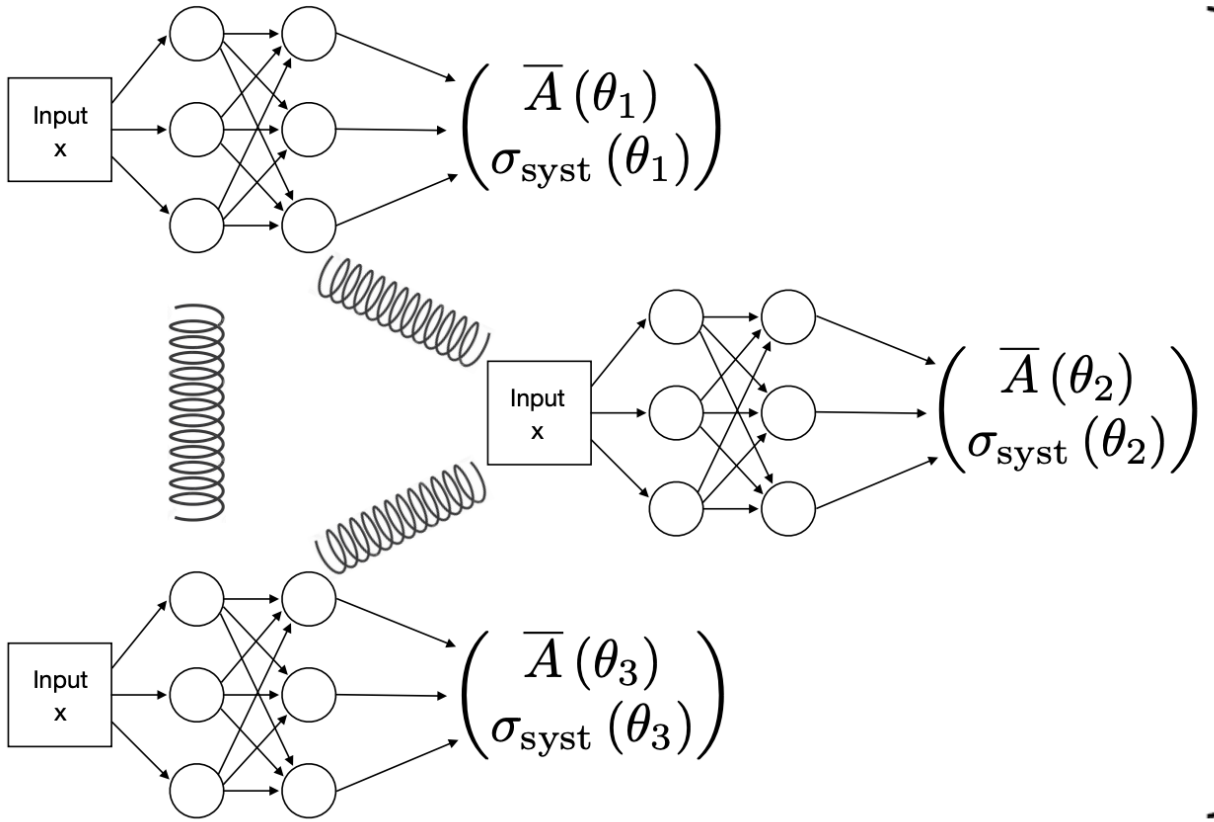
Encoding our physics knowledge



Bayesian neural networks



Repulsive ensembles



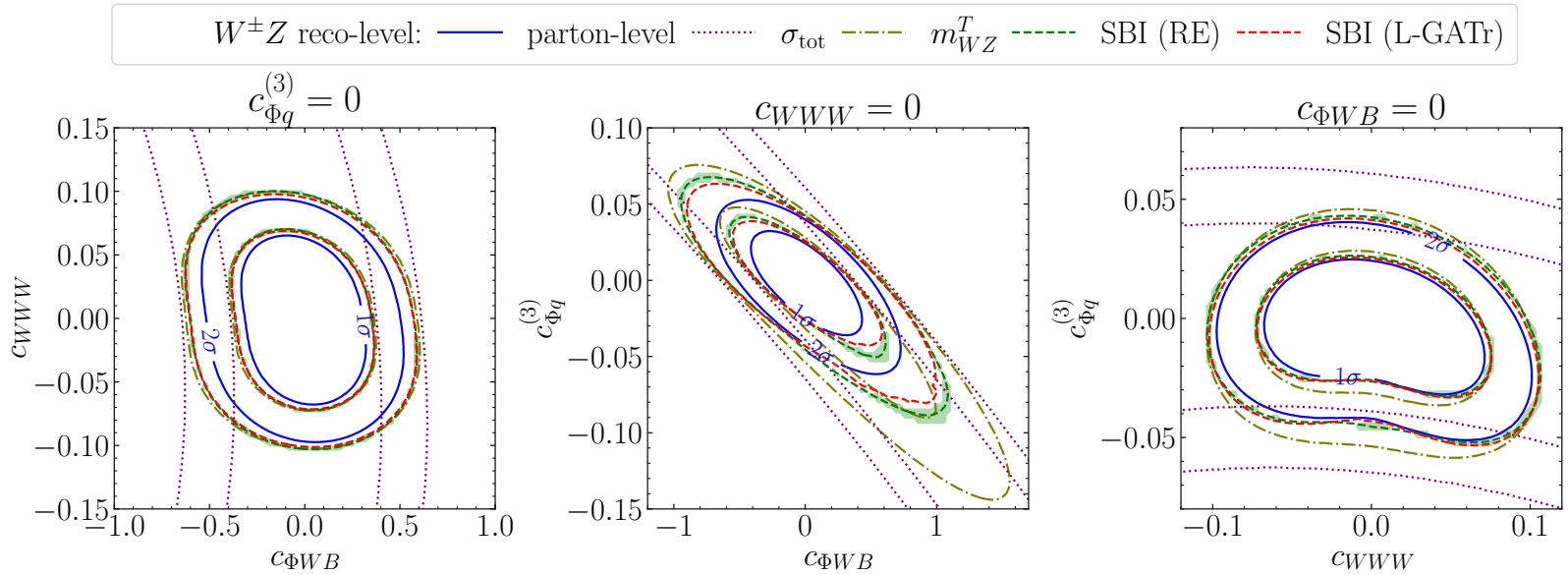
$$\langle A \rangle = \frac{1}{N} \sum_i^N \overline{A}(\theta_i)$$

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

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

Advanced SBI tools

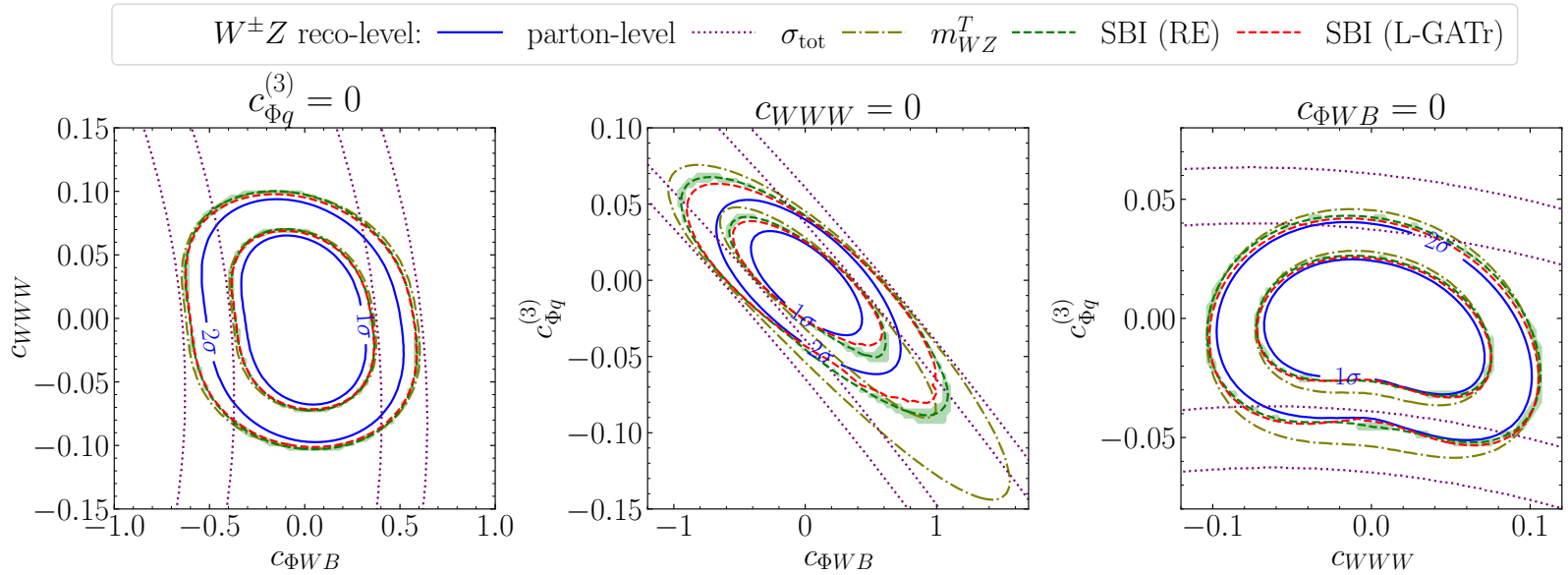
[HB et al.,2410.07315]



- target: SMEFT operators in $W^\pm Z$ production
- numerically stable results
- significantly better bounds than for histogram
- variety of cross-checks allows validating results

Advanced SBI tools

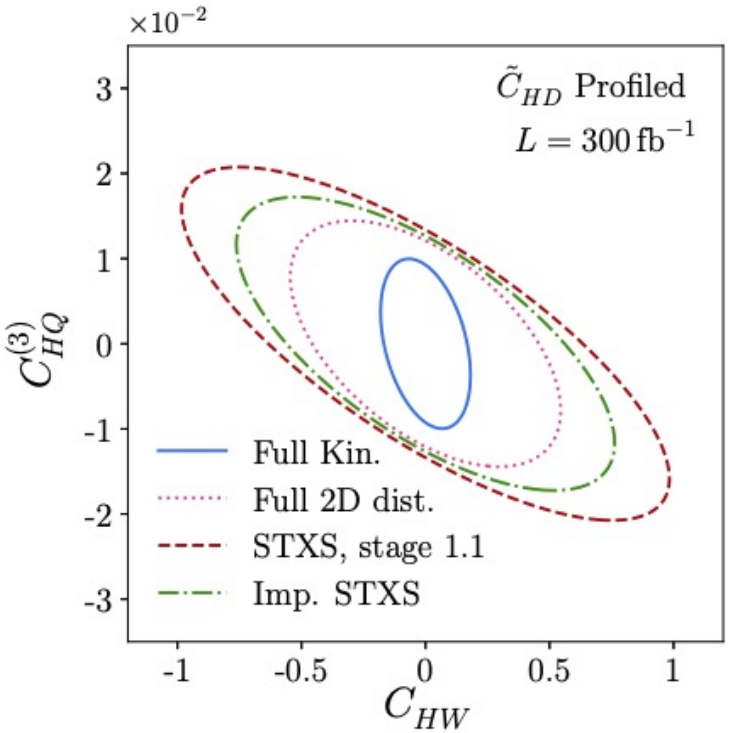
[HB et al.,2410.07315]



- target: SMEFT operators in $W^\pm Z$ production
- numerically stable results
- significantly better bounds than for histogram
- variety of cross-checks allows validating results

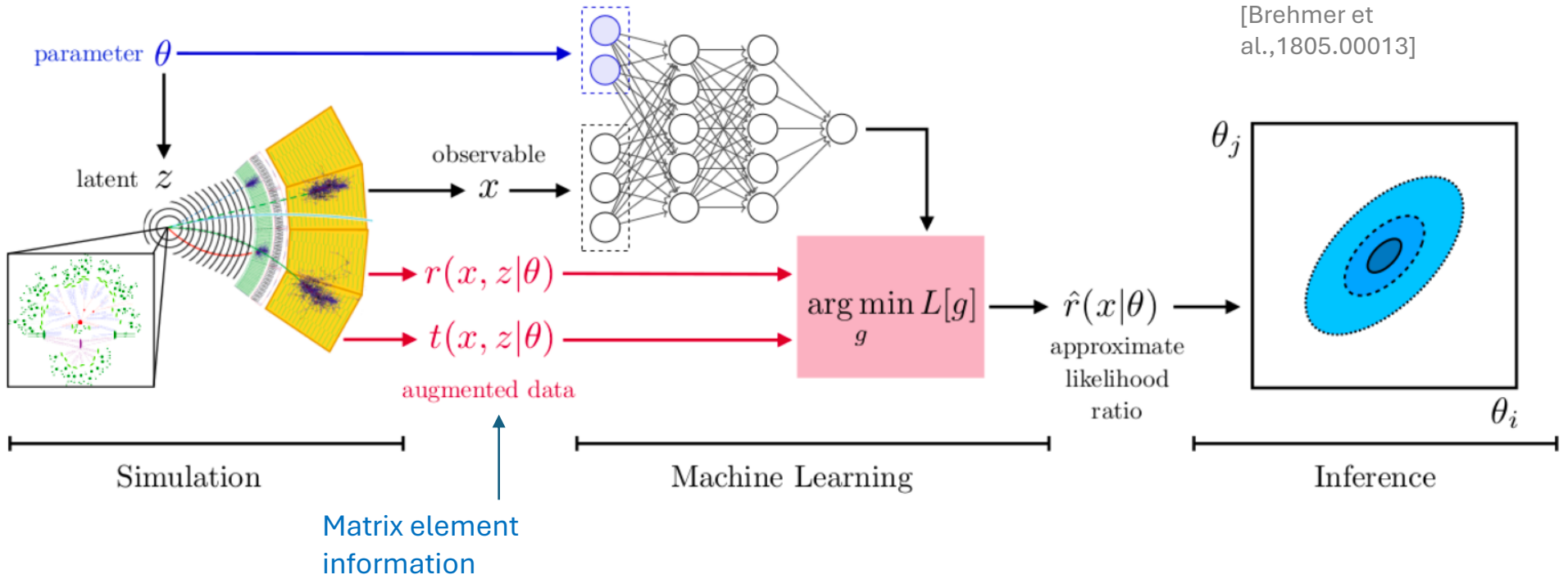
Future directions:

- application to masses, NLO corrections
- more pheno studies
- work towards real data application


$\times 10^{-2}$
$\tilde{C}_{HD}$ Profiled
$L = 300\,\mathrm{fb}^{-1}$
Full Kin.
Full 2D dist.
STXS, stage 1.1
Imp. STXS
$C_{HQ}^{(3)}$
C_{HW}
3
2
1
0
-1
-2
-3
-1
-0.5
0
0.5
1

Simulation-based inference

[Brehmer et al.,1906.01578,1805.12244,1805.00013,1805.00020,1808.00973]



[Brehmer et al.,1805.00013]

- Allows to extract the full available information (maximal sensitivity).
- No information loss due to binning (as for BDT analysis).
- No approximation of shower and detector effects (as for matrix-element approach).
- Use implementation in public code MadMiner designed to work with MadGraph + Pythia + Delphes. [Brehmer,Kling,Espejo,Cranmer,1907.10621]