Constraining effective field theories with machine learning

Johann Brehmer¹, Kyle Cranmer¹, Irina Espejo¹, Alexander Held¹,
Felix Kling², Gilles Louppe³, Juan Pavez⁴

¹ New York University
² SLAC National Accelerator Laboratory
³ University of Liège
³ Federico Santa María Technical University

CHEP 2019, Adelaide
November 7, 2019
Figure 5.7. Graphical representation of the results of Tables 5.2 and 5.4, where we compare the 95% CL bounds on the 34 degrees of freedom included in the present analysis, both in the marginalised (global) and in the individual fit cases, with the bounds reported in the LHC Top WG EFT note [10]. The individual bounds are in general rather tighter than the marginalised ones, except for some of the four-heavy-quark operators (and for $O_{tZ}$) where they are instead comparable.

Another useful way to present our results is by representing the bounds on $\frac{|c_i|}{\Lambda}$ that are derived from the fit. This is interesting because, assuming UV completions where the values of the fitted degrees of freedom $c_i$ are $O(1)$, plotting the results this way indicates the approximate reach in energy that is being achieved by the SMEFT global analysis. This comparison is shown in Fig. 5.8, which is the analogous plot as Fig. 5.7 now representing the same bounds as bounds on the ratio $\frac{|c_i|}{\Lambda}$ (now only for the marginalised bounds from the global fit). We find that for the degrees of freedom that are better constrained, we achieve sensitivity up to scales as high as $\lambda_1^{1.5} \text{TeV}$, in particular thanks to the chromomagnetic operator $O_{tG}$ which is well determined from the differential measurements of top quark pair production. Future measurements based on larger statistics should allow us to prove even higher scales, in particular by means of the high-luminosity LHC datasets.

5.3 The impact of the NLO QCD and $\mathcal{O}(\neq 4)$ corrections
The baseline fit results presented above are based on theory calculations that account both for the NLO QCD corrections to the SMEFT contributions and for the quadratic $\mathcal{O}!_\neq 4$ terms in Eq. (2.2), see also the discussion in Sect. 2. Here we aim to assess the robustness and stability of our results by comparing the baseline fit results with those of fits based on two alternative theory settings. Firstly we compare with a fit where only LO QCD effects are included for the SMEFT contributions, and then with a fit that includes only the linear $\mathcal{O}!_\neq 2$ terms in the effective theory expansion (but still based on NLO QCD for the SMEFT contributions).

The inference challenge

High-dimensional measurements $x$ are linked to parameters $\theta$ with likelihood $L(\theta) = p(x|\theta)$.
The full likelihood can be factorized:

\[ p \left( x \mid \theta \right) = \int d z_P d z_S d z_D \ p \left( z_P \mid \theta \right) p \left( z_S \mid z_P \right) p \left( z_D \mid z_S \right) p \left( x \mid z_D \right) \]

Parton level \( z_P \)

The dependence on parameters \( \theta \) is here.
There are four major Higgs boson production modes accessible in proton–proton collisions at the LHC, namely gluon–gluon fusion, vector boson fusion, top quark–antiquark annihilation, and single top quark production. The cross-section as a function of the center-of-mass (COM) energy are presented in figure 2.10 for a Higgs boson mass of 125 GeV. Figure 2.9 shows exemplary LO Feynman diagrams for these four modes, while the respective Higgs boson production cross-section calculation [34].

The loop-induced gluon–gluon fusion is the dominant production mode. Due to its large Yukawa coupling, V_H, the processes gg → H (N3LO QCD + NLO EW), tH (NLO QCD, t-ch + s-ch), and qqH (NNLO QCD + NLO EW) are observed in detectors. In the description of each of these phases different approximations are employed. In general the central parton-level or matrix-element (ME) computations based on matrix elements, which can be observed in detectors, are accurate at leading logarithmic order and, eventually, supplemented with exact first-order results. A transferring constant owing to the correspondence to exact perturbation theory, 

\[
p (x | \theta) = \int dz_P dz_S dz_D p (z_P | \theta) p (z_S | z_P) p (z_D | z_S) p (x | z_D)
\]

The dependence on parameters \( \theta \) is here.
The likelihood (3)

- The full likelihood can be factorized:

\[
p(x | \theta) = \int d_{z_P} d_{z_S} d_{z_D} \ p(z_P | \theta) \ p(z_S | z_P) \ p(z_D | z_S) \ p(x | z_D)
\]

**parton level** \(z_P\)

**parton shower** \(z_S\)

**detector interaction** \(z_D\)

The dependence on parameters \(\theta\) is here.
The likelihood (4)

- The full likelihood can be factorized:

\[
p(x | \theta) = \int dz_P dz_S dz_D \ p(z_P | \theta) \ p(z_S | z_P) \ p(z_D | z_S) \ p(x | z_D)
\]

- The dependence on parameters \( \theta \) is here.

The inner working of event generators is therefore one of the principal tasks of particle physicists: the simulation of the process by an event generator is a crucial ingredient in the derivation of predictions. Parton showers model multiple QCD bremsstrahlung in an appropriate way and are accurate at leading logarithmic order. At the hadronisation scale, which is of the order of a few TeV, the primary hadrons finally are decayed into particles that can be observed in detectors. In most cases phenomenological fragmentation models having typically a few parameters to be fitted to data are accurate to leading logarithmic order. At the hadroproduction scale, the underlying event, hadronisation effects of QCD bremsstrahlung in the initial and final states, where remnants of the incoming hadrons may experience secondary hard or semi-hard interactions. This mixture of QCD and electromagnetic processes is simulated by techniques that are beyond QCD factorization theorems and therefore no complete analytic solutions are available. A different approximations are employed. In general the central part of the event simulation is provided by the hard process piece of the event simulation is provided by the hard process generators is usually relies on factorized factorization (the dark red blob in the figure), which is simulated by techniques that are either hard-coded or provided by special programs called parton-level or matrix-element (ME) generators. The likelihood (4) can be observed in detectors. In most cases logical models are employed again, with more or less success, depending on the corresponding theoretical results. In this context, the purple blob in Fig. 1 is here.

\[ M(H) = 125 \text{ GeV} \]

\[ \sigma(pp \rightarrow H + X) \ [\text{pb}] \]

4. The Standard Model

\[
\frac{W}{H} (N3LO QCD + NLO EW)
\]

\[
\frac{t}{tH} (NLO QCD, t-ch + s-ch)
\]

\[
\frac{q}{qqH} (NNLO QCD + NLO EW)
\]

\[
\frac{g}{gq\bar{q}}\text{-}\text{fusion} (\text{top left}), \text{vector boson fusion} (\text{bottom left}), \text{and} \text{loop-induced} \text{gluon-gluon fusion} \text{is the dominant production mode. Due to its large Yukawa}
\]

\[
\begin{align*}
\theta & = 6 \\
\bar{t} & \rightarrow H + X \\
H & \rightarrow W^+ W^- Z \\
t & \rightarrow W^+ W^- Z
\end{align*}
\]
Methods in practice

• Generally **not possible to use the full likelihood** function $L(\theta) = p(x | \theta)$

• **Alternative methods** are used, but have **drawbacks**:  
  ‣ Summary statistics are commonly used, at the cost of information loss  
  ‣ Matrix element method approximates part of the integral
• **In particle physics, more can be done**
  ‣ Joint likelihood ratio can be calculated: 
    \[ r(x, z | \theta, \theta_0) = \frac{p(z | \theta)}{p(z | \theta_0)} \]
    - Depends on reconstructed objects \( x \) and parton-level kinematics \( z \)
  ‣ Also possible to take derivatives wrt. parameters: 
    \[ t(x, z | \theta) = \nabla_{\theta} \log p(x, z | \theta) \]

• A **neural network** can **learn** an estimator for the **likelihood ratio** \( \hat{r}(x | \theta) \)

• “**Mining Gold**”:
  ‣ Extract more information from simulator: \( x, r(x, z | \theta), t(x, z | \theta) \)
1) Simulate events \((x, z, \theta)\)

"Mining gold"
1) Simulate events \((x, z, \theta)\)

2) Learn the likelihood ratio

Simulator

Parameters \(\theta\) → Observables \(x\)

"Mining gold"
General approach (3)

1) Simulate events \((x, z, \theta)\)

2) Learn the likelihood ratio

3) Do inference

1) Simulate events \((x, z, \theta)\)

2) Learn the likelihood ratio

3) Do inference

“Mining gold”
The MadMiner package

- **MadMiner** implements this workflow in a ready-to-use python package
  - Standalone solution for phenomenological analyses
  - Modular structure, can replace elements for LHC-style analyses etc.
  - Code is on github: diana-hep/madminer

- **Get started!**
  - `pip install madminer`
  - Tutorials: notebooks with example implementations

- **Future plans:**
  - Integrate within frameworks for LHC experiment use
1) Event generation

Physics info about processes

MadMiner

MadGraph5+

Pythia

Parton-level events (.lhe)

Hadron-level events (.hepmc)

Parameterizes dependence on $\theta$ with event weights

* formally MadGraph5_aMC@NLO
1) Event generation

Physics info about processes

- MadMiner
  - MadGraph5$^*$
    - Pythia
      - Parton-level events (.lhe)
        - Hadron-level events (.hepmc))

2) Observables

Observables & cuts

- DelphesReader
  - LHEReader
    - Delphes
      - Detector-level Events (.root)

Modular structure, could in principle replace by GEANT4
1) Event generation
- MadMiner
- MadGraph5*
- Pythia
- Parton-level events (.lhe)
- Hadron-level events (.hepmc)

2) Observables
- Observables & cuts
- DelphesReader
- LHEReader
- Delphes
- Detector-level Events (.root)

3) Sampling
- Configuration
- SampleAugmenter
- Training data

Files
- Unweighted events for various settings of $\theta$

* formally MadGraph5_aMC@NLO
MadMiner structure (4)

1) Event generation
- Physics info about processes
  - MadMiner
  - MadGraph5
    - MadGraph5_aMC@NLO
  - Pythia
- External tools
  - MadMiner class
    - DelphesReader
    - LHEReader
- Parton-level events (.lhe)
- Hadron-level events (.hepmc)

2) Observables
- Observables & cuts
  - DelphesReader
  - LHEReader
- External tools
  - MadMiner class
    - Delphes
- Detector-level Events (.root)

3) Sampling
- Configuration
  - SampleAugmenter
  - LikelihoodEstimator
  - RatioEstimator
  - ScoreEstimator
  - Ensemble
- Training data

4) Machine learning
- Configuration
  - LikelihoodEstimator
  - RatioEstimator
  - ScoreEstimator
  - Ensemble
- Trained model 
  \( \hat{r}(x|\theta) \)

Can also use for histogram-based analysis

* formally MadGraph5_aMC@NLO
1) Event generation

Physics info about processes

MadMiner

- MadGraph5*+
- Pythia

Parton-level events (.lhe)

Hadron-level events (.hepmc))

2) Observables

Observables & cuts

DelphesReader
LHEReader

Delphes

Detector-level Events (.root)

3) Sampling

Configuration

SampleAugmenter

Configuration

LikelihoodEstimator
RatioEstimator
ScoreEstimator
Ensemble

4) Machine learning

Training data

Trained model $\hat{r}(x|\theta)$

5) Inference

Observed events

AsymptoticLimits

more planned…

* formally MadGraph5_aMC@NLO

Files

External tools

MadMiner class
Example application: VBF (1)

- **Example: VBF production**
  - Sensitive to two coefficients $f_W, f_{WW}$

![Feynman diagram for Higgs production in weak boson fusion](image)

- **Varying coefficients:**

![Graph showing distribution of two quantities](image)

Example application: VBF (2)

- Example: VBF production
  - Sensitive to two coefficients $f_W, f_{WW}$
  - Varying coefficients:

![Feynman diagram for Higgs production in weak boson fusion in the $4\ell$ mode.](image)

Learning likelihood ratio as function of 42 features:

- Truth	-Dotted line: 2D histogram	-Dashed line: RASCAL

Example application: VBF (3)

- Example: VBF production
  - Sensitive to two coefficients $f_W, f_{WW}$
  - Varying coefficients:

\[
W, Z, h
\]

Learning likelihood ratio as function of 42 features:

• **Example: Wilson coefficients with ttH**
  
  ‣ Using $t\bar{t}H(\gamma\gamma)$ process to measure three parameters: $c_{uG}, c_{u}, c_{G}$
  
  ‣ Systematic uncertainties used: PDF and scale variations ($\mu_R, \mu_F$)

• Neural network learns dependence of $\hat{f}(x \mid \theta)$ on systematic variations

Summary

• Inference with traditional methods does not scale well to measuring many parameters at once

• A new family of methods was developed to address this challenge
  ‣ Combining machine learning techniques with matrix element information
  ‣ Provides likelihood ratio \( \hat{r}(x|\theta) \) for inference
  ‣ Can define locally optimal observables
  ‣ Promising performance, especially for multi-parameter measurements (EFT etc.)
  ‣ Methods applicable beyond HEP (strong lensing example: arXiv:1909.02005)

• MadMiner is a python package automating the workflow with these new techniques
  ‣ Implements full chain for phenomenological study
  ‣ Future plans: integrate within frameworks for LHC experiment use
  ‣ pip install madminer
Backup
References

• Techniques:
  ‣ Johann Brehmer, Kyle Cranmer, Gilles Louppe, Juan Pavez: *Constraining Effective Field Theories with Machine Learning*
  ‣ Johann Brehmer, Kyle Cranmer, Gilles Louppe, Juan Pavez: *A Guide to Constraining Effective Field Theories with Machine Learning*
  ‣ Johann Brehmer, Gilles Louppe, Juan Pavez, Kyle Cranmer: *Mining gold from implicit models to improve likelihood-free inference*
    - arXiv:1805.12244
  ‣ Markus Stoye, Johann Brehmer, Gilles Louppe, Juan Pavez, Kyle Cranmer: *Likelihood-free inference with an improved cross-entropy estimator*
    - arXiv:1808.00973
    - arXiv:1907.10621

• Applications (beyond what is already mentioned above):
  ‣ Johann Brehmer, Sally Dawson, Samuel Homiller, Felix Kling, Tilman Plehn: *Benchmarking simplified template cross sections in WH production*
    - arXiv:1908.06980
  ‣ Johann Brehmer, Siddharth Mishra-Sharma, Joeri Hermans, Gilles Louppe, Kyle Cranmer: *Mining for Dark Matter Substructure: Inferring subhalo population properties from strong lenses with machine learning*
    - arXiv:1909.02005

• MadMiner: [https://github.com/diana-hep/madminer](https://github.com/diana-hep/madminer), DOI: 10.5281/zenodo.2574893
Scaling with training sample size

Figure 13: Performance of the techniques as a function of the training sample size. As a metric, we show the mean squared error (left) and trimmed mean squared error on \( \log r(\mathbf{x}_0, \mathbf{x}_1) \) weighted with a Gaussian prior, as discussed in the text. Note that we do not vary the size of the calibration data samples. The number of epochs are increased such that the number of epochs times the training sample size is constant, all other hyperparameters are kept constant. The Sally method works well even with very little data, but plateaus eventually due to the limitations of the local model approximation. The other algorithms learn faster the more information from the simulator is used.

In Fig. 14 we show the evolution of the likelihood estimation error and the cross-entropy of the classification problem during the training of the parameterized estimators. For comparison, we also show the optimal metrics based on the true likelihood ratio, and the results of the two-dimensional histogram approach. Once again we see that either Cascal or Rascal leads to the best results. This result also holds true for the cross entropy, hinting that the techniques we use to measure continuous parameters might also improve the power of estimators in discrete classification problems. Note that the Carl approach is more prone to overfitting than the others, visible as a significant difference between the metrics evaluated on the training and validation samples.

Equally important to the training efficiency is the computation time taken up by evaluating the likelihood ratio estimators \( \hat{r}(\mathbf{x}^e|\mathbf{x}_0, \mathbf{x}_1) \). We compare example evaluation times in Tbl. V. The traditional histogram approach takes the shortest time. But all tested algorithms are very fast: the likelihood ratio for fixed hypotheses \( \mathbf{x}_0, \mathbf{x}_1 \) for \( 50,000 \) events \( \{\mathbf{x}^e\} \) can always be estimated in around one second or less. The local score regression method is particularly efficient, since the estimator \( \hat{t}(\mathbf{x}|\text{score}_0, \mathbf{x}_1) \) has to be evaluated only once to estimate the likelihood ratio for any value of \( \mathbf{x}_0 \). Only the comparably fast step of density estimation has to be repeated for each tested value of \( \mathbf{x}_0 \). So after investing some training time upfront, all the measurement strategies developed here can be evaluated on any events with very little computational cost and amortize quickly. While this is not the focus of our paper, note that this distinguishes our approaches from the Matrix Element Method and Optimal Observable techniques. These well-established methods require the computationally expensive evaluation of complicated numerical integrals for every evaluation of the baseline: 2d histogram.

• 1.5M signal events, 1M background, 10M unweighted events for training
  ‣ Including background contributions from $t\bar{t} + \gamma\gamma$
• Three-layer fully-connected NN with 100 nodes per layer