Validation agreement for MC implementations of the SMEFT

Gauthier Durieux
(Technion)

[1906.12310]

Gauthier Durieux (ed.), Ilaria Brivio (ed.),
Fabio Maltoni (ed. ex officio), Michael Trott (ed. ex officio),
Simone Alioli, Andy Buckley, Mauro Chiesa, Jorge de Blas, Athanasios Dedes, Céline Degrande,
Ansgar Denner, Christoph Englert, James Ferrando, Benjamin Fuks, Peter Galler, Admir Greljo,
Valentin Hirschi, Gino Isidori, Wolfgang Kilian, Frank Krauss, Jean-Nicolas Lang, Jonas Lindert,
Michelangelo Mangano, David Marzocca, Olivier Mattelaer, Kentarou Mawatari, Emanuele
Mereghetti, David J. Miller, Ken Mimasu, Michael Paraskevas, Tilman Plehn, Laura Reina, Janusz
Rosiek, Jürgen Reuter, José Santiago, Kristaq Suxho, Lampros Trifyllis, Eleni Vryonidou,
Christopher White, Cen Zhang, Hantian Zhang

VBSCan WG1 meeting
23 October 2019

Motivation

Implementations of the SMEFT have multiplied.

Cross validations are slightly non trivial
and basically inexistent (or private).

Experimentalists may be somewhat lost.

Mandate

from the LHC HXS/EW/TOP WGs

to implementation authors

inspired by the TOP WG EFT standards [1802.07237]

where some top-centric comparisons were performed

Groups involved

dim6top	https://feynrules.irmp.ucl.ac.be/wiki/dim6top
SMEFTsim	https://feynrules.irmp.ucl.ac.be/wiki/SMEFT
SMEFT@NLO	https://feynrules.irmp.ucl.ac.be/wiki/SMEFTatNLO
SmeftFR	https://www.fuw.edu.pl/smeft
HEL	https://feynrules.irmp.ucl.ac.be/wiki/HEL
BSMC	https://feynrules.irmp.ucl.ac.be/wiki/BSMCharacterisation
Powheg SM-EFT	https://powhegbox.mib.infn.it
Recola2	https://recola.hepforge.org
HiggsPO	https://www.physik.uzh.ch/data/HiggsPO
Whizard	https://whizard.hepforge.org
Sherpa	https://sherpa.hepforge.org

comparisons performed so far:

- dim6top–SMEFTsim
- dim6top–SMEFT@NLO
- Recola2–SMEFTsim

Agreement principles

decentralized pair-wise comparisons

(by implementation authors, first between sets of similar ones, lots of diversity)

based on squared matrix elements (SM,linear,squares)

at random phase-space points (only num. unc., no MC)

for equivalent parameter points (sometimes tricky)

information stored and shared in LHE+SLHA formats

(as standard as possible)

Further details

couplings & parameters needs to be fully EFT-linearized

(equivalence theorem otherwise breaks)

input scheme with all masses

m_Z and m_W in particular (to avoid EFT-dependent propagators)

so far:

- widths=0 (to avoid EFT-dependent propagators)
- spin+colour averages
- CKM=1 by default
- LO but extendable to loop level following BLHA

Equivalent parameter points

sometimes tricky to establish

- different assumptions (flavour, CP, etc.) and sets of operators
- different linear combinations, signs and normalizations
- different covariant derivative conventions (i.e. field definitions)
- different parametrizations (e.g. Re+Im vs. abs+phase)
- ...

encouraged implementation of conversions in public codes

`Rosetta/wcxf-python`

MadGraph plugin for UFO comparisons

streamline the amplitude computation
given equivalent SLHA parameter cards

append to/create LHE comparison output
+ human-readable tables

commented example and outputs at

`https://bazaar.launchpad.net/~rwgtdim6/mg5amcnlo/plugin_eft_contrib/view/head:/example/compare_models_top.py`

The top-EFT example

SMEFT@NLO ↔ dim6top ↔ SMEFTsim

-1. Work out the correspondence between model parameters

```
...
cos(cuBPh3x3)*cuBAbs3x3 = (-cw)/(sw)*ctW+(1)/(sw)*ctZ
cos(cuGPh3x3)*cuGAbs3x3 = -ctG
cos(cuHPh3x3)*cuHAbs3x3 = ctp
cos(cuWPh3x3)*cuWAbs3x3 = -ctW
cos(cud1Ph1331)*cud1Abs1331 = 4/9*cbtud8+1/3*cbtud1
...
```

0. Create sets of equivalent SLHA parameter cards

...

- param_cbtud1.dat
- param_cbtud1I.dat
- param_cbtud8.dat
- param_cbtud8I.dat
- param_cbW.dat
- param_cbWI.dat
- param_cpQ3.dat
- param_cpQM.dat
- param_cpt.dat
- param_cptb.dat
- param_cptbI.dat
- param_cQb1.dat
- param_cQb8.dat

...

The top-EFT example

SMEFT@NLO ↔ dim6top ↔ SMEFTsim

1. Specify the SLHA benchmark locations

```
# Paths to the directories containing param_<tag>.dat cards for each model
benchmarks = [
    os.path.join(curr_dir,'dim6top_LO_UFO'),
    os.path.join(curr_dir,'SMEFTatNLO_U2_2_U3_3_cG_4F_LO_UFO'),
    os.path.join(curr_dir,'SMEFTsim_A_general_MwScheme_UFO_v2')]
# * param_sm.dat may be used to define a SM-like benchmark.
```

2. Specify the models (and restrictions)

```
# List of models to compare
model_list = [p+'-all' for p in benchmarks]
# * A restriction card can be loaded with each model by appending
#   '-<restriction>' to its name (like the +'-all' string above), if the model
#   directory contains a restrict_<restriction>.dat file ('restrict_all.dat' here).
# * If the LHE file has to be generated it is generated based on the first model.
#   If that model contains a 'param_card_generic.dat' it is used at that point.
#   Note that this model needs to be generic enough since process without
#   Feynman diagram for this model will not be include in the LHE file.
# * If the LHE file is already present, the reweighting backend will need to be
#   able to find the model with which it was initially created (as specified by
#   the "import model" command in the <MG5ProcCard> subsection of the <header>
#   section).
```

The top-EFT example

SMEFT@NLO ↔ dim6top ↔ SMEFTsim

3. Specify the LHE input/output file

```
# Path to the output or input LHE file
lhe_path = 'events.lhe'
# * If this file already exists, we will add weight each event in that file
#   and do not generate any additional event. The 'processes', 'set_orders'
#   inputs won't be taken into account.
```

4. Specify the amplitudes to compare

```
# List of processes to generate
processes = ['g g > t~ t', 'g a > t~ t', 'g t > z t', 'g t > h t', 'g b > w- t',
# * This can also be a list containing a path to a text file with one process
#   per line.
# * This is ignored if a LHE file is provided. The later must contain a <header>
#   section with a <MG5ProcCard> sub-section in which "generate ... add
#   process ..." commands specify the list of processes.
```

The top-EFT example

SMEFT@NLO ↔ dim6top ↔ SMEFTsim

5. Specify how to get SM,linear,squares in each model

```
# Functions that takes each process string above as input and returns a string
# that specifies coupling orders.
set_orders_model1 = [
    lambda process: process + ' FCNC=0 QED=99 QCD=99 DIM6=0',
    lambda process: process + ' FCNC=0 QED=99 QCD=99 DIM6=1 DIM6^2==1',
    lambda process: process + ' FCNC=0 QED=99 QCD=99 DIM6=1 DIM6^2==2']
set_orders_model2 = [
    lambda process: process + ' QED=99 QCD=99 NP=0',
    lambda process: process + ' QED=99 QCD=99 NP=2 NP^2==2',
    lambda process: process + ' QED=99 QCD=99 NP=2 NP^2==4']
set_orders_model3 = [
    lambda process: process + ' SMHLOOP=0 QED=99 QCD=99 NP=0',
    lambda process: process + ' SMHLOOP=0 QED=99 QCD=99 NP=1 NP^2==1',
    lambda process: process + ' SMHLOOP=0 QED=99 QCD=99 NP=1 NP^2==2']
set_orders = [set_orders_model1, set_orders_model2, set_orders_model3]
# * In the case of an EFT model, a triplet of function for each model specifies
#   the coupling orders restrictions to obtain the SM squared amplitude, its
#   interference with one insertion of EFT operator, and the square of
#   amplitudes with one EFT operator insertion: [sm,int,sq].
# * It can be used to forbid some interactions (e.g. FCNC=0, SMHLOOP=0).
# * Given that different models would by default include different QED and QCD
#   orders, specifying them explicitly is safer (beware of conflicts with the
#   orders specified already in the processes list).
# * The ORDER^2==n syntax is used to isolate the interference and quadratic
#   EFT contributions. (It does not work with the decay chain syntax, or beyond
#   leading order.)
# * Note that some model define one new-physics coupling insertion to have
#   NP_ORDER=2 instead of 1.
```

The top-EFT example

SMEFT@NLO ↔ dim6top ↔ SMEFTsim

6. Run!

```
run_comparison.run(lhe_path, model_list, processes, benchmarks, set_orders, **opts)
```

7. Get the LHE output

```
<event>
 4      1 +4.8535595e-01 9.11180000e+01 0.00000000e+00 1.18400000e-01
       21 -1    0    0    0    0 +0.0000000000e+00 +0.0000000000e+00 +5.0000000000e+02 5.0000000000e+02 0.0000000000e+00 0.
       21 -1    0    0    0    0 +0.0000000000e+00 +0.0000000000e+00 -5.0000000000e+02 5.0000000000e+02 0.0000000000e+00 0.
       -6  1    1    2    0    0 +1.9434628146e+01 -4.5815572793e+02 +1.0065596947e+02 5.0000000000e+02 1.7200000000e+02 0.
        6  1    1    2    0    0 -1.9434628146e+01 +4.5815572793e+02 -1.0065596947e+02 5.0000000000e+02 1.7200000000e+02 0.
<rwgt>
<wgt id='dim6top_LO_UFO-all-sm-sq'> +4.4179812e-01 </wgt>
<wgt id='dim6top_LO_UFO-all-ctGI-int'> +9.1515248e-19 </wgt>
<wgt id='dim6top_LO_UFO-all-ctG-int'> +1.3905392e-01 </wgt>
<wgt id='dim6top_LO_UFO-all-ctGI-sq'> +8.1727506e-02 </wgt>
<wgt id='dim6top_LO_UFO-all-ctG-sq'> +8.7713058e-02 </wgt>
<wgt id='SMEFTatNLO_U2_2_U3_3_cG_4F_LO_UFO-all-sm-sq'> +4.4179812e-01 </wgt>
<wgt id='SMEFTatNLO_U2_2_U3_3_cG_4F_LO_UFO-all-ctG-int'> +1.3905391e-01 </wgt>
<wgt id='SMEFTatNLO_U2_2_U3_3_cG_4F_LO_UFO-all-ctG-sq'> +8.7713050e-02 </wgt>
<wgt id='SMEFTsim_A_general_MwScheme_UFO_v2-all-ctGI-int'> -4.5442110e-08 </wgt>
<wgt id='SMEFTsim_A_general_MwScheme_UFO_v2-all-ctG-int'> +1.3905392e-01 </wgt>
<wgt id='SMEFTsim_A_general_MwScheme_UFO_v2-all-ctGI-sq'> +8.1727506e-02 </wgt>
<wgt id='SMEFTsim_A_general_MwScheme_UFO_v2-all-ctG-sq'> +8.7713058e-02 </wgt>
```

8. Examine PDF table

1.20 $gg \to \bar{t}t$

	interference			square		
	mod1	mod2	$1 - \text{mod2}/\text{mod1}$	mod1	mod2	$1 - \text{mod2}/\text{mod1}$
ctG	0.1391	0.1391	0	0.08771	0.08771	0
ctGI	$9.152 \cdot 10^{-19}$	$-4.544 \cdot 10^{-8}$	$4.966 \cdot 10^{10}$	0.08173	0.08173	0
sm	0	0	0	0.4418	0.4418	0

Summary

A validation protocol for SMEFT implementations
has been developed.

UFO model comparisons are facilitated by a MadGraph plugin.

Providing parameter conversions in public tools is encouraged.

Your model comparisons are welcome!