

# CoDEx: Wilson coefficient calculator connecting SMEFT to UV theory

Supratim Das Bakshi<sup>a</sup>, Joydeep Chakraborty<sup>b</sup>, Sunando Kumar Patra<sup>c</sup>

Department of Physics, Indian Institute of Technology, Kanpur 208016, India

Received: 30 September 2018 / Accepted: 10 November 2018 / Published online: 11 January 2019  
© The Author(s) 2019

**Abstract** CoDEx is a *Mathematica*<sup>®</sup> package that calculates the Wilson coefficients (WCs) corresponding to effective operators up to mass dimension 6. Once the part of the Lagrangian involving single and multiple degenerate heavy fields, belonging to some beyond standard model (BSM) theory, is given, the package can then integrate out propagators from the tree and 1-loop diagrams of that BSM theory. It then computes the associated WCs up to 1-loop level, for two different bases: "Warsaw" and "SILH". "CoDEx" requires only very basic information as regards the heavy field(s), e.g., color, isospin, hypercharge, mass, and spin. The package first calculates the WCs at the high scale (mass of the heavy field(s)). We then have an option to perform the renormalization group evolutions (RGEs) of these operators in "Warsaw" basis, a complete one (unlike "SILH"), using the anomalous dimension matrix. Thus, one can get all effective operators at the electro-weak scale, generated from any such BSM theory, containing heavy fields of spin 0, 1/2, and 1. We provide many example models (both here and in the package documentation) that more or less encompass different choices of heavy fields and interactions. Relying on the status of the present day precision data, we restrict ourselves up to dimension-6 effective operators. This will be generalized for any dimensional operators in a later version.

## 1 Introduction

It is a perplexing time for particle physics. On one side we are cherishing the discovery of the standard model (SM)-Higgs like particle, considered to be the pinnacle of success of the SM; on the other hand we have enough reason to believe the existence of theories beyond it. To address the shortcomings of the SM, many beyond-standard model (BSM) scenarios

are proposed at very different scales. It is believed that any such theory, which contains the SM as a part of it, will affect the electro-weak and the Higgs sector. Thus the precision observables are expected to carry the footprints of the *new* physics, unless it is in the decoupling limit.

The ongoing and proposed future experiments are expected to improve the sensitivity of these precision observables at per mille level. Thus we can indirectly estimate the allowed room left for some BSM physics, even in the case of non-observation of new resonances. This motivates us to look into the BSM scenario through the tinted glass of standard model effective field theory (SMEFT). The basic idea of SMEFT is quite straightforward: integrate out heavy non-SM degrees of freedom and capture their impact through the higher mass dimensional operators— $\sum_i (1/\Lambda^{d_i-4}) C_i \mathcal{O}_i$ . Here  $d_i$  is the mass dimensionality of the operator  $\mathcal{O}_i$  (starting from 5), and  $C_i$  is the corresponding Wilson coefficient, a function of the BSM parameters. It is important to note that the choice of operator basis, i.e., the explicit structure of the  $\mathcal{O}_i$ , is not unique. Among different choices we restrict ourselves to the "SILH" [1,2] and "Warsaw" [3–6] bases. These bases can be transformed from one to another.  $\Lambda$  is the cut-off scale at which all WCs are computed ( $C_i(\Lambda)$ ) and usually identified as the mass of the heavy field being integrated out. This EFT approach relies on the validity of the perturbative expansion of the S-matrix in powers of  $\Lambda^{-1}$  (UV-scale), and the resultant series is expected to pass the convergence test. As this scale is higher than the scale  $M_Z$ , where the precision test is performed, dimension-6 operators are more suppressed than the dimension-5 ones and so on. Now we ask where to truncate the  $1/\Lambda$  series. This decision is made case by case, based on the achieved (expected) precision level of the observables at present (future) experiments [7]. One can consult the lectures in [8–12] where effective field theory has been introduced and discussed in great detail. Several other packages and libraries are available in the literature, which address various issues regarding SMEFT operators and the corresponding Wilson

<sup>a</sup> e-mail: [sdbakshi13@gmail.com](mailto:sdbakshi13@gmail.com)

<sup>b</sup> e-mail: [joydeep.chakraborty@gmail.com](mailto:joydeep.chakraborty@gmail.com)

<sup>c</sup> e-mail: [sunando.patra@gmail.com](mailto:sunando.patra@gmail.com)

coefficients, from basis transformation to running of the coefficients [13–17]. A universal data exchange format for BSM Wilson coefficients has also been developed (WCxf) [18].

Now the nagging questions are: (a) Why use SMEFT instead of doing the full calculation, using the supposedly more accurate BSM Lagrangian? (b) How can one ensure that the difference between the results, computed in the SMEFT approach using a truncated S-matrix and those obtained using the full BSM theory, is imperceptible (in the precision tests)?

The computation with the full BSM is involved and tedious, and that too at loop level. The cut-off  $\Lambda$  is chosen in such a way that the  $M_Z/\Lambda$  series is converging, which ensures that the truncation of this series at some finite order is safe and sufficient. Even then, the question remains: How do we connect the physics of two different scales, namely UV and the  $M_Z$ ? The WCs, which we are computing using SMEFT, are at the scale  $\Lambda$ , but the observables are measured at the  $M_Z$  scale. Hence, we need to evolve the  $C_i(\Lambda)$  to obtain  $C_i(M_Z)$ , using the anomalous dimension matrix ( $\gamma$ ). While performing the renormalization group evolutions (RGEs) of the  $C'_i$ , we need to choose  $\gamma$  carefully, as it is basis dependent. Thus we need to choose only those bases in which the precision observables are defined, and it is important to ensure that the basis we are working with is a complete one. As the matrix  $\gamma$  contains non-zero off-diagonal elements, it is indeed possible to generate, through RGEs, some new effective operators, which were absent at the  $\Lambda$  scale. These effective field theory approach has been successfully used in the context of precision data and Higgs phenomenology; for details see [19–41].

With this backdrop, we introduce **CoDEx**, a *Mathematica*<sup>®</sup> package, which can integrate out the heavy field propagator(s) from tree and 1-loop processes and can generate SM effective operators up to dimension 6. It also provides the Wilson coefficients as a function of the BSM parameters.

We know that the 1-loop results depend on the choice of renormalization scheme; thus the choice of the scheme is very important and crucial. Our code provides the 1-loop results under the  $\overline{\text{MS}}$  renormalization scheme. We would like to make a remark on the results obtained at the 1-loop level: The present version of this code can integrate out only heavy propagators, thus the 1-loop diagrams that involve both heavy and light types of propagators are ignored which will have less impact on the presence of other contributions; see Ref. [20] (Sect. 1.2). In the case of operators of dimension  $< 5$ , the inclusion of a 1-loop correction is related to the renormalization of the full theory. There are standard tools to compute such loop corrections. In this paper our focus is on the evaluation of higher dimensional operators after integrating out the heavy BSM fields. That is why we have not implemented the loop correction to the operators of mass dimension less than 5. From this perspective, the 1-loop results are incomplete.

The article is organized in the following manner: in Sect. 2, we briefly discuss the underlying principle of **CoDEx** from a theoretical perspective. Details as regards downloading and installation are in Sect. 3.1. In the remainder of Sect. 3, we provide a guideline to define the heavy field(s) and build the working Lagrangian (Sect. 3.2), provide a list of all the functions that are necessary to run **CoDEx** in detail (Sect. 3.3), and explain the way **CoDEx** takes care of the RGEs of the WCs down to the electro-weak scale (Sect. 3.4). In the next section, we provide the user with one detailed work flow to use the package for a model with a single heavy electro-weak  $SU(2)_L$  real singlet scalar in it (Sect. 3.5). Sequentially following these steps should enable one to find the effective operators up to mass dimension 6 and the respective WCs at the high scale for that model. In the appendix we provide various example models to encapsulate different types of fields that are used frequently to build BSM scenarios. One may consult Refs. [4–6, 42] regarding the running of the SM effective operators.



## Program Summary

*Program Title:* CoDEx

*Version:* 1.0.0

*Licensing provisions:* CC By 4.0

*Programming language:* Wolfram Language<sup>®</sup>

*Mathematica<sup>®</sup> Version:* 10 +

*URL:* <https://effexteam.github.io/CoDEx>

*Send BUG reports and Questions:* [effex.package@gmail.com](mailto:effex.package@gmail.com)

## 2 The underlying principle

In this section, we will briefly discuss the adopted method, based on the idea of a *Covariant Derivative Expansion* (CDE), to integrate out the heavy fields to compute the Wilson coefficients (WCs). This was introduced in Ref. [43] and then extended in Ref. [44]. As we are performing this “*integrating out of the field(s)*” order by order in perturbation theory, we need to respect the gauge invariance at each and every step. The perturbative expansion thus is required to be done in terms of some gauge covariant quantities. Thus, the covariant derivative is the ‘chosen one’. CDE is not only restricted to quantify the integrating out of heavy fields, rather it has a wider impact; see [19–21] for details. The method of integrating out different types of heavy fields using functional methods and the basis dependency is discussed in many places in the literature; see [14, 45–54] for details.

Considering the status of present and prospect of future experiments, we can adjudge ourselves safe, when we restrict ourselves to only dimension-6 operators including the tree

and 1-loop parts of the effective action. The modus operandi of **CoDEx** is based on the method of CDE discussed in [19–21]. Briefly, this is what **CoDEx** can give you:

- It will integrate out the heavy BSM field(s) while respecting the gauge invariance.
- It will generate Wilson coefficients at tree and/or 1-loop level.
- It will perform RG evolutions of the effective operators generated at some high scale and provide the operators at the electro-weak scale.

Let us consider a BSM Lagrangian in the following form:

$$\mathcal{L}^{(\text{BSM})} \equiv \mathcal{L}^{(\text{BSM})}(\phi, \Phi) = \mathcal{L}^{(\Phi)} + \mathcal{L}^{(\phi)} + \mathcal{L}_{int}^{(\phi, \Phi)}, \quad (1)$$

where  $\phi$  and  $\Phi$  represent light (SM) and heavy (BSM) fields, respectively. Here  $\mathcal{L}^{(\phi)}$ ,  $\mathcal{L}^{(\Phi)}$ , and  $\mathcal{L}_{int}^{(\phi, \Phi)}$  are three different sectors of the BSM Lagrangian containing, respectively, only heavy fields, only light fields, and both heavy and light fields.

To proceed, we have to first solve the following Euler–Lagrange (EL) equation to compute the solution for the heavy field:

$$\frac{\partial}{\partial \Phi} \mathcal{L}^{(\text{BSM})}(\phi, \Phi) = \mathcal{D}_\mu \frac{\partial}{\partial (\mathcal{D}_\mu \Phi)} \mathcal{L}^{(\text{BSM})}(\phi, \Phi), \quad (2)$$

where  $\mathcal{D}_\mu$  is the covariant derivative corresponding to the heavy field  $\Phi$ . To find the EL-equations, we will concentrate on the  $\mathcal{L}^{(\text{BSM})}(\phi, \Phi)$  part only. If we express this part of the Lagrangian as a polynomial in  $\Phi$ , e.g.,  $\mathcal{L}_I^{(\phi)} \cdot \Phi + \mathcal{L}_{II}^{(\phi)} \cdot (\Phi)^2 + \dots$ , the coefficients can then be written as

$$\mathcal{L}_I^{(\phi)} = \mathcal{O}_D \cdot \hat{\Phi}, \quad (3)$$

where  $\mathcal{O}_D$  contains the information regarding the covariant derivative of the heavy field and the functional of the light (SM) fields. In general,  $\mathcal{O}_D$  is in the form of an elliptic operator, e.g.  $\mathcal{O}_D = \mathcal{D}^2 + M^2 + \mathcal{L}_{II}^\phi$ , where  $M$  is the mass of the heavy field to be integrated out. Thus the heavy field solution can be rewritten as

$$\hat{\Phi} = [\mathcal{O}_D]^{-1} \cdot \mathcal{L}_I^{(\phi)}. \quad (4)$$

The operator  $[\mathcal{O}_D]^{-1}$  can be Taylor-expanded, where terms are suppressed by  $M^{2n}$  ( $n$  takes integer values starting from 1). This series is convergent as  $M$  is much larger than the allowed maximum momentum transfer in the low energy theory. Thus we can truncate this series based on our requirement of the mass dimension of the effective operator. It is important to note that in the theories where the  $\mathcal{L}_I^{(\phi)}$  term is absent, any tree-level effective operator will not be generated after integrating out  $\Phi$ .

The next task is to compute these effective operators at loop level. We will restrict our computation to the 1-loop level, relying on the precision of present data. The effective Lagrangian at 1-loop level is given by [19–21]

$$\begin{aligned} \mathcal{L}_{1\text{-loop}}^{(\text{dim-6})}[\phi] &= \frac{c_s}{(4\pi)^2} \text{tr} \left\{ \frac{1}{m^2} \left[ -\frac{1}{60} (P_\mu G'_{\mu\nu})^2 - \frac{1}{90} G'_{\mu\nu} G'_{\nu\sigma} G'_{\sigma\mu} \right. \right. \\ &\quad \left. \left. - \frac{1}{12} (P_\mu U)^2 - \frac{1}{6} U^3 - \frac{1}{12} U G'_{\mu\nu} G'_{\mu\nu} \right] \right. \\ &\quad \left. + \frac{1}{m^4} \left[ \frac{1}{24} U^4 + \frac{1}{12} U (P_\mu U)^2 + \frac{1}{120} (P^2 U)^2 \right. \right. \\ &\quad \left. \left. + \frac{1}{24} (U^2 G'_{\mu\nu} G'_{\mu\nu}) - \frac{1}{120} [(P_\mu U), (P_\nu U)] G'_{\mu\nu} \right. \right. \\ &\quad \left. \left. - \frac{1}{120} [U[U, G'_{\mu\nu}]] G'_{\mu\nu} \right] + \frac{1}{m^6} \left[ -\frac{1}{60} U^5 - \frac{1}{20} U^2 (P_\mu U)^2 \right. \right. \\ &\quad \left. \left. - \frac{1}{30} (U P_\mu U)^2 \right] + \frac{1}{m^8} \left[ \frac{1}{120} U^6 \right] \right\}, \quad (5) \end{aligned}$$

where  $c_s = \frac{1}{2}, 1, -\frac{1}{2}$ , and  $\frac{1}{2}$  for real scalar, complex scalar, fermion, and gauge boson, respectively. Here  $P_\mu = i \mathcal{D}_\mu$  and  $G'_{\mu\nu} = [\mathcal{D}_\mu, \mathcal{D}_\nu]$ .  $U$  is a collection of coefficients of the terms which are bi-linear in the heavy field.  $U$  can be written in matrix form of  $\delta[\mathcal{L}^{(\text{BSM})}(\phi, \Phi)]/\delta\Phi_i \delta\Phi_j$ , evaluated at  $\hat{\Phi}_i$  [19–21]. It is important to note that ‘tr’ in the above equation is the trace performed over the internal symmetry indices.

Once we find the effective operators and their respective Wilson coefficients at high scale, i.e., the scale of new physics, we can run them down to the electro-weak scale. These operators are evolved with the energy scale as

$$\frac{d\mathcal{O}_i}{d \ln \mu} = \gamma_{ij} \mathcal{O}_j, \quad (6)$$

where  $\gamma_{ij}$  is the anomalous dimension matrix. This is also implemented in **CoDEx**, but only for the "Warsaw" basis, as it is the complete one.

### 3 The package, in detail

#### 3.1 Installing and loading CoDEx

Installing **CoDEx** is quite straightforward. This can be done in one of the following ways:

##### 3.1.1 Automatic installation

**CoDEx** can be installed in a Mathematica environment by downloading and importing the installer file ‘install.m’ available at this link. The installer can also be automati-

**Table 1** Running 'InstallCoDEx' can be customized using its options

Options	Default	Detail
<code>AutoDisableInsufficientVersionWarning</code>	<b>None</b>	True → warning messages for notebooks created with newer Mathematica version automatically disabled <sup>a</sup> None → user will be asked by a dialog False → warnings will not be disabled
<code>AutoOverwriteCoDExDirectory</code>	<b>None</b>	True → Previous installations automatically deleted None → user will be asked by a dialog False → the directory will be overwritten
<code>InstallCoDExDevelopmentVersion</code>	<b>False</b>	True → Install latest <i>development</i> version False → Install the latest <i>stable</i> version
<code>InstallCoDExTo</code>	<code>"path"</code>	Specifies custom full path to installation directory

<sup>a</sup>Needed to generate documentation in older versions

cally loaded inside the Mathematica environment by using the command<sup>1</sup>

```
In[1]:= Import["https://github.com/effExTeam/CoDEx-1.0.0/raw/master/install.m"];
```

This loads two functions in the working kernel: 'InstallCoDEx' and 'InstallCoDExQuiet'. A typical way of running them is

```
In[2]:= InstallCoDEx[]
```

One can use `InstallCoDExQuiet` instead of `InstallCoDEx`, which is equivalent to running

```
In[2]:= InstallCoDEx[AutoDisableInsufficientVersionWarning→True,
AutoOverwriteCoDExDirectory→True]
```

using `InstallCoDEx`. You just need to run it in a slightly different way.

You need to copy the path of the downloaded archive (e.g. if you are working in Windows and have downloaded the .zip file to the 'Downloads' folder, your path will be `"C:\...\Downloads\CoDEx.zip"`. Let us call it `"path"` from now).

Then run this command in your notebook after importing the 'install.m' file:

```
In[1]:= $PathToCoDExArchive = "path";
InstallCoDEx[]
```

This goes through exactly the same steps as in the previous section, but instead of downloading the archive from the

### 3.1.2 Download archive file

There is a provision to download and install the program locally (Table 1).

The `CoDEx` package is available in both .zip and .tar format in its 'Github' repository. While using the 'Install' option from the Notebook menu inside Mathematica or manually extracting the downloaded archive file in the 'Applications' folder inside the `$UserBaseDirectory` works perfectly, installing `CoDEx` is made a lot easier by downloading and importing the installer file 'install.m' available here.

If for some reason you choose to download the archive files and install `CoDEx` from them, this can still be done,

server, one uses the local file. After this, the package can always be loaded in *Mathematica* using

```
In[2]:= Needs["CoDEx`"]
```

The installer file is not our creation (Table 2). We have edited and simplified the installer for *FeynRules* [55].

## 3.2 How to build the Lagrangian

### 3.2.1 Building the Lagrangian: an example

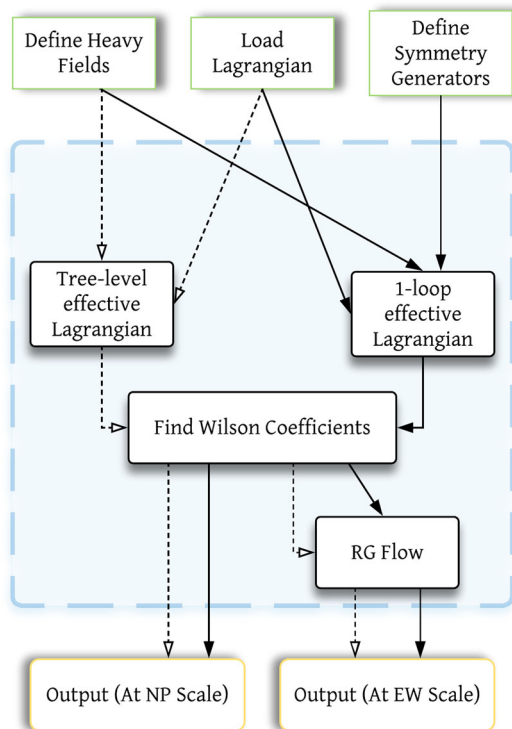
Let us demonstrate this with a toy example where the Lagrangian is given in its traditional form (Fig. 1):

<sup>1</sup> This requires a live internet connection and Mathematica should be able to connect to the internet.

**Table 2** Main functions provided by CoDEX

Function	Details
<code>CoDEXHelp</code>	Opens the <b>CoDEX</b> guide, with all help files listed
<code>treeOutput</code>	Calculates WCs generated from tree-level processes
<code>loopOutput</code>	Calculates WCs generated from 1-loop processes
<code>codexOutput</code>	Generic function for WCs calculation with choices for level, bases etc. given with <b>OptionValues</b>
<code>defineHeavyFields</code>	Creates representation of heavy fields
	Use the output to construct BSM Lagrangian
<code>texTable</code>	Given a <b>List</b> , returns the L <sup>A</sup> T <sub>E</sub> X output of a tabular environment, displayed and/or copied to clipboard <sup>a</sup>
<code>formPick</code>	Applied on a list of WCs from a specific operator basis, reformats the output in the specified style
<code>RGFlow</code>	RG Flow of WCs of dim. 6 operators in "Warsaw" basis, from matching scale to a lower (arbitrary) scale
<code>initializeLoop</code>	Prepares the Isospin and Color symmetry generators for a specific model with a specific heavy field content <b>loopOutput</b> can only be run after this step is done

<sup>a</sup>This is a simplified version of the package titled TeXTableForm [56]



**Fig. 1** Flow-chart for CoDEX

$$L(\Phi, H) \supset -\eta \Phi^a \Phi^a |H|^2 + 2\kappa H \tau^a \Phi^a H^\dagger - \frac{\lambda_\Phi}{4} (\Phi^a \Phi^a)^2. \tag{7}$$

Here  $\Phi$  is the heavy field which is going to be integrated out. From the user-end, the information for this heavy field is fed into the code as

```

fields =
{
    {fieldName, components,
    colorDim, isoDim,
    hyperCharge, spin, mass}
};
  
```

which contains the required information (within {...}) about the field to compute the WCs. The properties necessary to define the heavy field are listed in Table 3. Now, this code is equally applicable for multiple heavy BSM fields. In this case, the field definitions will be listed sequentially under the first set of curly braces (see Sect. 3.2.2). In the case of our example model, we have only one heavy field, a real triplet scalar (i.e. **components** → 3, **colorDim** → 1, **isoDim** → 3, **hypercharge** → 0, **spin** → 0). Let us denote **fieldName** → ‘ph’ and **mass** → ‘m’. This represents the field content of our model in the correct way:

```

In[1]:= fields =
{
    {ph, 3, 1, 3, 0, 0, m}
};
  
```

Now that our field definitions are ready, it is time to write the Lagrangian in a form that the code understands. For this purpose, first we have to create the representations of these fields in their component form. This is done by a specific function:

```

In[2]:= hfvecs = defineHeavyFields[fields]

Out[2]= {{{ph[1,1], ph[1,2], ph[1,3]}}}
  
```



**Table 3** Generic form of individual fields

Pattern	Details
<b>fieldName</b>	Field-name, used as array-head to construct the Lagrangian
<b>components</b>	Number of components in the heavy field
<b>colorDim</b>	Dimensionality of the heavy field under $SU(3)_C$
<b>isoDim</b>	Dimensionality of the heavy field under $SU(2)_L$
<b>hyperCharge</b>	Hypercharge of the heavy field under $U(1)_Y$
<b>spin</b>	Spin quantum number of the heavy field
<b>mass</b>	Variable representing the mass of the heavy field

To write the Lagrangian in a compact form one can define the heavy field as

```
In[3]:=  $\Phi = \text{hfvecs}[[1,1]]$ 
```

```
Out[3]= {ph[1,1], ph[1,2], ph[1,3]}
```

With this definition, the working Lagrangian, given in regular form in Eq. (7), can be written in compact form like

$$\text{In[4]:= } \text{LBSM} = 2 \kappa * \text{Table}[(\text{dag}[\text{H}].\text{tau}[\text{a}].\text{H}), \{\text{a}, 3\}] . \Phi - \eta \text{abs}[\text{H}]^2 * (\Phi . \Phi) - \frac{1}{4} \lambda (\Phi . \Phi)^2;$$

### 3.2.2 Other examples of defining fields

So far we have mentioned how to integrate out a single heavy field. In the case of multiple fields, the required format of the ‘fieldList’ would look like

```
fieldList =
{
  { fieldName1, components1,
    colorDim1, isoDim1,
    hyperCharge1, spin1,
    mass1 },
  { fieldName2, components2,
    colorDim2, isoDim2,
    hyperCharge2, spin2,
    mass2 },
  ...
  ...
};
```

Now we can create the heavy fields’ list:

```
In[.]:= hfVecs = defineHeavyFields[fieldList]
```

In general, if we have  $n$  heavy fields; then **hfVecs** is a list, whose  $i$ th element is the representation of the  $i$ th heavy field. Let us describe different cases in detail, where we have fields of different characteristics. A case with a single field is already shown in Sect. 3.2.1.

Following the same proposal, we can define multiple heavy fields. A possible example is

```
In[1]:= fieldList2 =
{
  {hf1, 3, 1, 3, 0, 0, m},
  {hf2, 6, 3, 2, 1/6, 0, m}
};
```

with which we define

```
In[2]:= hfarray2 = defineHeavyFields[fieldList2]
Out[2]= {{hf1[1,1], hf1[1,2], hf1[1,3]},
  {{hf2[1,1] + i ihf2[1,1], hf2[1,2]
  + i ihf2[1,2]}, {hf2[2,1]
  + i ihf2[2,1], hf2[2,2]
  + i ihf2[2,2]}, {hf2[3,1]
  + i ihf2[3,1], hf2[3,2] + i ihf2[3,2]}}
```

Here the representation for the first heavy field is

```
In[3]:= hfvec1 = hfarray2[[1,1]]
Out[3]= {hf1[1,1], hf1[1,2], hf1[1,3]}
```

and the second field is represented as

```
In[4]:= hfvec2 = hfarray2[[2]]
Out[4]= {{hf2[1,1] + i ihf2[1,1], hf2[1,2]+i ihf2[1,2]},
  {hf2[2,1] + i ihf2[2,1], hf2[2,2]+i ihf2[2,2]},
  {hf2[3,1] + i ihf2[3,1], hf2[3,2]+i ihf2[3,2]}}
```

Now, these two field representations (**hfvec1** and **hfvec2**) can be used to build the required Lagrangian.

For a spin-1 field, the field definition will be

```
In[1]:= fieldList3=
{
  {hf3, 3, 1, 3, 0, 1, m3}
};
```

We do not count the Lorentz components of a heavy field while writing the total number of field components (the second entry in **fieldList**).

```
In[2]:= hfarray3= defineHeavyFields[fieldList3]
Out[2]= {{hf3[1][1],hf3[1][2],hf3[1][3],hf3[1][4]},
          {hf3[2][1],hf3[2][2],hf3[2][3],hf3[2][4]},
          {hf3[3][1],hf3[3][2],hf3[3][3],hf3[3][4]}}
In[3]:= hfvec3 = hfarray3[[1]]
Out[3]= {{hf3[1][1],hf3[1][2],hf3[1][3],hf3[1][4]},
          {hf3[2][1],hf3[2][2],hf3[2][3],hf3[2][4]},
          {hf3[3][1],hf3[3][2],hf3[3][3],hf3[3][4]}}
```

In a similar manner, the spin-1/2 field is represented as

```
In[1]:= fieldList4 =
{
  {hf4, 3, 1, 3, 0,  $\frac{1}{2}$ , m4}
};
```

As before, we do not count the Lorentz components of a heavy field.

```
In[2]:= hfarray4= defineHeavyFields[fieldList4]
Out[2]= {{{{hf4[1,1][1],hf4[1,1][2],hf4[1,1][3],
            hf4[1,1][4]},
           {hf4b[1,1][1],hf4b[1,1][2],hf4b[1,1][3],
            hf4b[1,1][4]}},
          {{hf4[1,2][1],hf4[1,2][2],hf4[1,2][3],
            hf4[1,2][4]},
           {hf4b[1,2][1],hf4b[1,2][2],hf4b[1,2][3],
            hf4b[1,2][4]}},
          {{hf4[1,3][1],hf4[1,3][2],hf4[1,3][3],
            hf4[1,3][4]},
           {hf4b[1,3][1],hf4b[1,3][2],hf4b[1,3][3],
            hf4b[1,3][4]}}}}
In[3]:= hfvec4 = hfarray4[[1,1]][[All,1]]
Out[3]= {{hf4[1,1][1],hf4[1,1][2],hf4[1,1][3],
            hf4[1,1][4]},
          {hf4[1,2][1],hf4[1,2][2],hf4[1,2][3],
            hf4[1,2][4]},
          {hf4[1,3][1],hf4[1,3][2],hf4[1,3][3],
            hf4[1,3][4]}}
In[4]:= hfvec4b = hfarray4[[1,1]][[All,2]]
Out[4]= {{hf4b[1,1][1],hf4b[1,1][2],hf4b[1,1][3],
            hf4b[1,1][4]},
          {hf4b[1,2][1],hf4b[1,2][2],hf4b[1,2][3],
            hf4b[1,2][4]},
          {hf4b[1,3][1],hf4b[1,3][2],hf4b[1,3][3],
            hf4b[1,3][4]}}
```

Use **hfvec4** and **hfvec4b** as the field representation of the fermion (say,  $\psi$ ) and its Lorentz conjugate ( $\bar{\psi}$ ).

### 3.3 How to run the code

We have demonstrated how to build the Lagrangian in the previous section. Here we will discuss the necessary steps that need to be followed to compute the Wilson coefficients (WCs). Of the various options that **CoDEx** provides to the users, the first is the choice of operator bases (**operBasis**)

**Table 4** Available functions for calculating Wilson coefficients (WCs) for a given BSM Lagrangian

Function	Details
<b>treeOutput</b>	Calculates tree-level Wilson coefficients
<b>loopOutput</b>	Calculates 1 loop level Wilson coefficients
<b>codexOutput</b>	Generic function for WCs calculation up to 1 loop

between (i) "Warsaw", and (ii) "SILH". Here "Warsaw" is the default one. The next option is the level at which the user wants the WCs (Table 4).

One can compute WCs up to 1 loop using this code.

#### 3.3.1 Tree level

To obtain the tree-level Wilson coefficients, one needs to use the function **treeOutput**, used as

```
treeOutput [lagrangian, fieldList]
```

This will generate the WCs in the "Warsaw" basis. Now, to compute the same in "SILH" basis, one has to simply provide an explicit choice of the operator basis as

```
treeOutput [lagrangian, fieldList,
operBasis->"SILH"]
```

#### 3.3.2 1-loop level

To compute the WCs at 1-loop level only, we have to use another function, **loopOutput**. This can be used as

```
loopOutput ["model", lagrangian,
fieldList]
```

Unlike the tree-level case, here we need the transformation property of the heavy fields under the given gauge symmetry. More precisely, the structure of the generators determined by the dimensionality of the heavy field's representations must be provided explicitly. We have provided their structure up to fundamental and quadruplet cases for the  $SU(3)_C$  and  $SU(2)_L$  gauge groups, respectively. For more exotic BSM particles, we have kept provision for the user to define it. To do so, one has to run a function:

```
initializeLoop ["model", fieldList]
```

If the dimensionality of the heavy fields are within the mentioned ranges, then one does not need to provide the explicit structures of the generators. Otherwise, one has to provide all the generators explicitly for each and every heavy field as **isomodel**[p,i] and **colmodel**[p,a] where 'p'

**Table 5** The Options for `codexOutput`. Other than these, this function also takes all Options of `formPick`

Option	Default value	Details
<code>monitor</code>	<b>True</b>	Shows an animation while computing
<code>appearance</code>	"Percolate" (for Version $\geq$ 11)	Appearance of the animation
<code>operBasis</code>	"Warsaw"	Choice of basis of the Dim.-6 operators
<code>format</code>	<b>List</b>	The output format
<code>model</code>	" "	Takes the same input as <code>initializeLoop</code> If left blank, will give the tree-level
<code>outRange</code>	<b>All</b>	The level at which output is evaluated 'All' means the result calculates both tree and loop level results and combines them
<code>ibp</code>	<b>True</b>	Turns the 'Integration by Parts' option on Detail available on the <code>loopOutput</code> page

denotes the number of heavy fields, and 'i' and 'a' run from 1 to 3 and 1 to 8, respectively.

### 3.3.3 One function to find them all

Now, one can wish to get all the WCs, i.e., tree and 1-loop levels together. For this purpose, one can simply use the following function:

```
codexOutput [lagrangian, fieldList,
             model  $\rightarrow$  modelName_String]
```

Essentially, this function is all one needs to calculate the WCs and even format them in the correct way. With careful choice of `OptionValues`, one can obtain results at different levels, with different operator bases, and in different formats. We have enlisted the main options which can be found using `Options [codexOutput]`.

### 3.4 RGEs of WCs: anomalous dimension matrix and choice of basis

For a given BSM Lagrangian we can compute the effective dimension-6 operators and the associated Wilson coefficients by integrating out the heavy fields. `CODEx` can provide these results in two different bases: "Warsaw" and "SILH". In the "Warsaw" basis, operators form a complete basis unlike the "SILH" one. Thus we prefer to perform the running of the Wilson coefficients in "Warsaw" basis only. Once the WCs are computed at the high scale, one can run those effective operators using the Anomalous dimension matrix and can compute the operator structures at the electro-weak (EW) scale by using the `RGFlow` function. Results of this module will help to connect the EW observables and the BSM physics through the effective operators and the WCs (Table 5).

To perform the RG evolution of the WCs, we need to use the function

```
RGFlow [WCList_List, MatchingScale,
         $\mu$ ]
```

This function works in the following way:

- `RGFlow` takes a list as its first argument. List output from `treeOutput`, `loopOutput` and `codexOutput` functions should be used as the first argument.
- `RGFlow` takes Wilson coefficients of dimension-6 operators in the "Warsaw" basis only.
- The second argument is the scale at which the full BSM theory is matched with the EFT.
- $\mu$  can be any energy scale below the matching scale.

Here both the `MatchingScale` and the  $\mu$  can be symbolic inputs. The working principle of this function is as follows:

- Load the package through

```
In[1]:= Needs ["CODEx"]
```

- Say that the following is the output, in the form of a list of WCs, that you had found from an earlier session of `CODEx` and had saved. Let us give it a name:

```
In[2]:= trrt =
{
  {"qH",  $-\frac{\eta \kappa^2}{m^4}$ },
  {"qHbox",  $\frac{\kappa^2}{m^4}$ },
  {"qHD",  $-\frac{2 \kappa^2}{m^4}$ };
```



These WCs are evaluated at the high scale. Now, to compute the WCs at the electro-weak scale ( $\mu$ ) we need to perform their RGEs. After setting the matching scale (high scale) at the mass of the heavy particle ( $m$ ), we have to recall the function **RGFlow** as

- Remember that the RGE of WCs can only be performed in the "Warsaw" basis (as it a complete one) and not in the "SILH" basis.

```
In[3]:= floRes1 = RGFlow[trrt,m,\mu]
```

```
Out[3]= {
  {qH, -\frac{\eta \kappa^2}{m^4} + \frac{3 gW^2 \kappa^2 \text{Log}[\frac{\mu}{m}]}{3 gY^2 \kappa^2 \text{Log}[\frac{\mu}{m}]} + \frac{3 gW^4 \kappa^2 \text{Log}[\frac{\mu}{m}]}{32 m^4 \pi^2} - \frac{3 gY^2 \kappa^2 \text{Log}[\frac{\mu}{m}]}{27 gW^2 \eta \kappa^2 \text{Log}[\frac{\mu}{m}]} + \frac{9 gY^2 \eta \kappa^2 \text{Log}[\frac{\mu}{m}]}{32 m^4 \pi^2} + \frac{5 gW^2 \kappa^2 \lambda \text{Log}[\frac{\mu}{m}]}{32 m^4 \pi^2}},
  {qHbox, \frac{\kappa^2}{m^4} - \frac{32 m^4 \pi^2 \text{Log}[\frac{\mu}{m}]}{4 m^4 \pi^2} - \frac{7 gY^2 \kappa^2 \text{Log}[\frac{\mu}{m}]}{24 m^4 \pi^2}},
  {qHD, -\frac{2 \kappa^2}{m^4} - \frac{9 gW^2 \kappa^2 \text{Log}[\frac{\mu}{m}]}{16 m^4 \pi^2} + \frac{25 gY^2 \kappa^2 \text{Log}[\frac{\mu}{m}]}{48 m^4 \pi^2}},
  {qeH[1, 1], \frac{19 gW^2 \kappa^2 \text{Log}[\frac{\mu}{m}]}{48 m^4 \pi^2} \text{Yu}^\dagger[e] - \frac{3 gY^2 \kappa^2 \text{Log}[\frac{\mu}{m}]}{16 m^4 \pi^2} \text{Yu}^\dagger[e]},
  {quH[1, 1], \frac{19 gW^2 \kappa^2 \text{Log}[\frac{\mu}{m}]}{48 m^4 \pi^2} \text{Yu}^\dagger[u] - \frac{3 gY^2 \kappa^2 \text{Log}[\frac{\mu}{m}]}{16 m^4 \pi^2} \text{Yu}^\dagger[u]},
  {qdH[1, 1], \frac{19 gW^2 \kappa^2 \text{Log}[\frac{\mu}{m}]}{48 m^4 \pi^2} \text{Yu}^\dagger[d] - \frac{3 gY^2 \kappa^2 \text{Log}[\frac{\mu}{m}]}{16 m^4 \pi^2} \text{Yu}^\dagger[d]},
  {q1Hl[1, 1], \frac{gY^2 \kappa^2 \text{Log}[\frac{\mu}{m}]}{96 m^4 \pi^2}}, {q3Hl[1, 1], \frac{gW^2 \kappa^2 \text{Log}[\frac{\mu}{m}]}{96 m^4 \pi^2}},
  {qHe[1, 1], \frac{gY^2 \kappa^2 \text{Log}[\frac{\mu}{m}]}{48 m^4 \pi^2}}, {q1Hq[1, 1], -\frac{gY^2 \kappa^2 \text{Log}[\frac{\mu}{m}]}{288 m^4 \pi^2}},
  {q3Hq[1, 1], \frac{gW^2 \kappa^2 \text{Log}[\frac{\mu}{m}]}{96 m^4 \pi^2}}, {qHu[1, 1], -\frac{gY^2 \kappa^2 \text{Log}[\frac{\mu}{m}]}{72 m^4 \pi^2}},
  {qHd[1, 1], \frac{gY^2 \kappa^2 \text{Log}[\frac{\mu}{m}]}{144 m^4 \pi^2}}
}
```

- One can reformat, save, and/or export all these WCs corresponding to the effective operators at the electro-weak scale ( $\mu$ ) to L<sup>A</sup>T<sub>E</sub>X, using **formPick**. Below is an illustrative example:

```
In[4]:= formPick["Warsaw", "Detailed2", floRes1, Frame->All, FontSize->Medium,
  FontFamily->"Times New Roman"]
```

```
Out[4]=
```

$Q_H$	$(H^\dagger H)^3$	$\frac{\log(\frac{\mu}{m}) \left( -\frac{\eta \kappa^2 \left( -\frac{27gW^2}{2} - \frac{9gY^2}{2} \right)}{m^4} - \dots + \frac{40gW^2 \kappa^2 \lambda}{3m^4} \right)}{16\pi^2} - \frac{\eta \kappa^2}{m^4}$
$\vdots$	$\vdots$	$\vdots$
$Q_{Hd}$	$(H^\dagger i \overleftrightarrow{D}_{\mu} H)(\bar{d} \gamma^{\mu} d)$	$\frac{gY^2 \kappa^2 \log(\frac{\mu}{m})}{144\pi^2 m^4}$

Here we have shown a truncated version of the resulting long table.<sup>2</sup>

```
In[5]:= RGFlow[{"oH", \frac{\kappa^2}{m^4}}, m, \mu]
```

Out[5]= RG flow only works when the Wilson coefficient basis is complete. For now, it only works for the 'Warsaw' basis.

<sup>2</sup> Fun-fact: This table in L<sup>A</sup>T<sub>E</sub>X format, and other similar results used in this article are all created using **formPick** as well.

### 3.5 Detailed example: electro-weak $SU(2)_L$ real singlet scalar

Here we demonstrate the work flow of **CoDEx** with the help of a complete analysis of a representative model. This and many others are listed in the package documentation. We also list the results of the other models in Appendix A of this article. Say the Lagrangian is

$$\mathcal{L}_{\text{BSM}} = \mathcal{L}_{\text{SM}} + \frac{1}{2} (\partial_\mu \phi)^2 - \frac{1}{2} m_\phi^2 \phi^2 - c_a |H|^2 \phi - \frac{1}{2} \kappa |H|^2 \phi^2 - \frac{1}{3!} \mu \phi^3 - \frac{1}{4!} \lambda_\phi \phi^4. \quad (8)$$

---

```
In[5]:= ["ewrssl", fieldewrssl]
```

```
» Isospin Symmetry Generators for the field 'hf' are isoewrssl[1,a] = 0
» Color Symmetry Generators for the field 'hf' are colewrssl[1,a] = 0
```

---

Here  $\phi$  is the real singlet scalar. Once this field is integrated out, few effective operators will emerge. To obtain those effective dimension-6 operators and their respective Wilson

3. Then we need to build the relevant part of the Lagrangian (involving the heavy field only). As a side-note, we should mention that for **CoDEx** to function, it does not need the heavy field kinetic term (the covariant derivative and the mass terms). Thus, the only part of the Lagrangian we need here is

$$\text{In[4]:= Lpotenewrssl=Expand[-c_a*abs[H]^2*\phi - \frac{\kappa}{2}*abs[H]^2*\phi^2 - \frac{1}{3!} \mu*\phi^3 - \frac{1}{4!} \lambda*\phi^4];$$

4. Next, we need to construct the symmetry generators:

(See the documentation of [initializeLoop](#) for details.)

5. The last step is the computation of effective operators and associated WCs as

---

```
In[7]:= res1=codexOutput[Lpotenewrssl, fieldewrssl, model->"ewrssl", outRange->"Tree"];
formPick["Warsaw", "Detailed2", res1, FontSize->Medium,
FontFamily->"Times New Roman", Frame->All]
```

---

coefficients using **CoDEx**, we need to perform the following steps:

1. First, load the package:

```
In[1]:= Needs["CoDEx`"]
```

2. We have to define the field  $\phi$  as

```
In[1]:= fieldewrssl=
{
{hf, 1, 1, 1, 0, 0, m}
};
```

```
In[2]:= hfvecsewrssl=defineHeavyFields
[fieldewrssl];
```

```
In[3]:= phi = hfvecsewrssl[[1,1,1]]
```

```
Out[3]= hf [ 1, 1 ]
```

6. The output is obtained in "Warsaw" basis and is formatted as a detailed table in **TraditionalForm**. There is provision to export the result in  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  format. Table 6c is actually obtained from the output of the code above. We can compute the same in "SILH" basis as well and for that we have to use

**Table 6** Effective operators and Wilson coefficients for real singlet scalar. These results are calculated in the  $\overline{\text{MS}}$  renormalization scheme. The 1-loop result may be altered based on the choice of renormalization scheme, e.g. in this particular case we have noted mismatches with results given in Ref. [19] where a different renormalization scheme has

been considered. Here we have highlighted the extra terms obtained in our calculation in the  $\overline{\text{MS}}$  scheme in ‘Red’ color. We have further cross-checked these results in the other scheme, adopted in Ref. [19], explicitly. For results in ‘Warsaw’ basis, no such comparison has been done

(a) ‘‘SILH’’ basis (tree level)	
$O_H$	$\frac{c_a^2}{m^4}$
$O_6$	$\frac{\mu c_a^3}{6m^6} - \frac{\kappa c_a^2}{2m^4}$
(b) ‘‘SILH’’ basis (1-loop level)	
$O_H$	$\frac{\mu^2 c_a^2}{192\pi^2 m^6} - \frac{\kappa \mu c_a}{96\pi^2 m^4} + \frac{\lambda c_a^2}{16\pi^2 m^4} + \frac{\kappa^2}{192\pi^2 m^2}$
$O_6$	$-\frac{\kappa \lambda c_a^2}{32\pi^2 m^4} - \frac{\kappa^3}{192\pi^2 m^2}$
(c) ‘‘Warsaw’’ basis (tree level)	
$Q_H$	$\frac{\mu c_a^3}{6m^6} - \frac{\kappa c_a^2}{2m^4}$
$Q_{HD}$	$-\frac{2c_a^2}{m^4}$
(d) ‘‘Warsaw’’ basis (1-loop level)	
$Q_H$	$-\frac{\kappa \lambda c_a^2}{32\pi^2 m^4} - \frac{\kappa^3}{192\pi^2 m^2}$
$Q_{HD}$	$\frac{\mu^2 c_a^2}{96\pi^2 m^6} - \frac{\kappa \mu c_a}{48\pi^2 m^4} - \frac{\lambda c_a^2}{8\pi^2 m^4} + \frac{\kappa^2}{96\pi^2 m^2}$

```
In[8]:= res2=codexOutput[Lpotenewrss,fieldewrss,model->"ewrss",operBasis->"SILH",
outRange->"Tree"];
formPick["SILH","Detailed2",res2,FontSize->Medium,
FontFamily->"Times New Roman",Frame->All]
```

Output of this can be found in Table 6a. Similarly, results for a 1-loop calculation can be obtained by changing the option value of ‘outRange’ to ‘Loop’. The default value of outRange is ‘All’, which combines both tree and 1-loop results.

- As is demonstrated in Sect. 3.4, these resulting WCs can then be run down to the electro-weak scale, using **RGFlow**.

### 3.6 Miscellaneous

**CoDEx** is written in *Wolfram Language*<sup>®</sup> [57]. Careful steps have been taken to speed up the code using parallelization over multi-cores, when available, while keeping the customizability for the user. All the example models listed in this article and in the documentation have been run on different processors, with different operating systems and versions of Mathematica.

When run on a 1.6 GHz Intel<sup>®</sup> Core i5 processor, the models take ~ 20–2000 s to run. Tree-level runs never take more than a minute. The only exception is 2HDM, which we have run in a 16 core Xeon processor, with 32 GB RAM.

How much time it takes to get the WCs for the user’s model depends on its structure and complexity. We hope a

user can have a clear idea about run time if she runs all the examples given in documentation as trials.

We hope to enable the output of future versions of **CoDEx** in ‘WCxf’ format in addition to the present ones.

## 4 Summary

**CoDEx** allows one to integrate out single and/or multiple degenerate heavy field(s) in a gauge covariant way. The user needs to provide the part of the Lagrangian that involves heavy BSM fields only. One needs to identify that BSM field by providing its number of component fields, spin, mass and quantum numbers under standard model gauge symmetry in a certain way. **CoDEx** then integrates out the heavy field propagators from all tree-level and/or 1-loop processes, and generates the Wilson coefficients for an exhaustive set of effective operators in both ‘SILH’ and ‘Warsaw’ bases. It allows the user to run the operators in ‘Warsaw’ basis down to the electro-weak scale. As the precision observables can be recast in terms of the effective operators, it will be really helpful to test the BSM physics in the light of electro-weak precision data. A list of example models are provided along with the package. These include a variety of field representations usually used by the BSM model builders.

In the present version of **CoDEx**, the user can compute operators up to mass dimension 6 by integrating out particles up to spin 1. This package can integrate out only heavy field propagators at the tree and 1-loop levels. In a future version, we will include a few other aspects, such that it can deal with (i) loops containing light (SM) and heavy (BSM) mixed field propagators, and (ii) non-degenerate multiple heavy BSM fields [58].

**Acknowledgements** The authors acknowledge the useful discussions with Soumitra Nandi. This work is supported by the Department of Science and Technology, Government of India, under the Grant IFA12/PH/34 (INSPIRE Faculty Award); the Science and Engineering Research Board, Government of India, under the agreement SER-B/PHY/2016348 (Early Career Research Award), and the Initiation Research Grant, agreement no. IITK/PHY/2015077, by IIT Kanpur.

**Open Access** This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made. Funded by SCOAP<sup>3</sup>.

### Appendix A: Field representations

In the rest of the article, we build and use Lagrangians using the fields listed in Table 7. We have checked that for these given models, the **CoDEx** generated results in "**SILH**" basis are well in agreement with those given in Refs. [19–21]. Examples with spin-1/2 and spin-1 heavy fields are demonstrated in the package documentation. In Appendix C (example 2), we have considered a non-trivial example where we have two kinds of heavy fields:  $SU(2)_L$  doublet and singlet scalars and both are triplets under  $SU(3)_C$ . In this particular example, the heavy fields are not linearly coupled to the SM ones, thus the 1-loop results provided here are complete.

Here we would like to mention that whenever there is a linear coupling among the heavy BSM and SM fields, there exists a contribution to the tree-level Wilson coefficients. In these cases, the contributions to the WCs will be from the 1-loop level diagrams that involve not only heavy propaga-

tors but also both the light and the heavy propagators. In the examples given in Appendix B and the first example model in Appendix C, there will be an added contribution to the WCs (apart from those encoded in the tables there) due to light–heavy field mixing. In the present version of the code, this contribution is not computed. In addition to that, we are also not computing the contributions to the operators of mass dimension  $< 5$ . Thus, from this perspective, the quoted results are incomplete. To be precise, the present version of the code provides the Wilson coefficients corresponding to operators of dimensions 5 and 6 at tree and 1-loop level, involving only heavy propagators ignoring the light–heavy mixing contributions.

### Appendix B: Examples: single heavy BSM field

1. Electro-weak  $SU(2)_L$  Singlet Scalar with hypercharge  $Y = 0$ : Discussed in detail in Sect. 3.5.
2. Electro-weak  $SU(2)_L$  Triplet Scalar with hypercharge  $Y = 0$

$$\begin{aligned} \mathcal{L}_{\text{BSM}} = \mathcal{L}_{\text{SM}} + \frac{1}{2} (\mathcal{D}_\mu \Phi)^2 \\ - \frac{1}{2} m_\Phi^2 \Phi^a \Phi^a + 2 \kappa H^\dagger \tau^a H \Phi^a \\ - \eta |H|^2 \Phi^a \Phi^a - \frac{1}{4} \lambda_\Phi (\Phi^a \Phi^a)^2. \end{aligned} \quad (\text{B1})$$

Here the heavy field is  $\Phi$ . The internal quantum numbers and its other required properties are given in Table 7. Once the  $\Phi$  is integrated out using **CoDEx**, the effective operators up to dimension 6 for both "**SILH**" and "**Warsaw**" bases are generated, which are listed in Table 8.

3. Two Higgs doublet model (2HDM)

$$\begin{aligned} \mathcal{L}_{\text{BSM}} = \mathcal{L}_{\text{SM}} + |\mathcal{D}_\mu \varphi|^2 - m_\varphi^2 |\varphi|^2 \\ - \frac{\lambda_\varphi}{4} |\varphi|^4 + (\eta_H |\tilde{H}|^2 + \eta_\varphi |\varphi|^2)(\tilde{H}^\dagger \varphi + \varphi^\dagger \tilde{H}) \end{aligned}$$

**Table 7** SM gauge quantum numbers of BSM fields

BSM field	No. of field components	$SU(3)_C$ quantum number	$SU(2)_L$ quantum number	$U(1)_Y$ charge <sup>a</sup>	Spin	Mass
$\phi$	1	1	1	0	0	$m_\phi$
$\Phi$	3	1	3	0	0	$m_\Phi$
$\Delta$	3	1	3	1	0	$m_\Delta$
$\varphi$	2	1	2	-1/2	0	$m_\varphi$
$\tilde{Q}_{3L}$	6	3	2	1/6	0	$m_{\tilde{Q}_3}$
$\tilde{\tau}_R$	3	3	1	2/3	0	$m_{\tilde{\tau}_R}$

<sup>a</sup>Electromagnetic charge  $Q = T_3 + Y$ , where  $T_3$  is isospin quantum number

**Table 8** Effective operators and Wilson coefficients in for the real triplet scalar (Y=0) model. As the light-heavy mixing (which can be ignored safely for phenomenological analyses) is not included, this result is incomplete

Dimension-6 operators	Wilson coefficient
(a) ‘‘SILH’’ basis (tree level)	
$O_6$	$-\frac{\eta\kappa^2}{m_\Phi^4}$
$O_T$	$\frac{\kappa^2}{m_\Phi^4}$
$O_R$	$\frac{2\kappa^2}{m_\Phi^4}$
(b) ‘‘SILH’’ basis (1-loop level)	
$O_{2W}$	$\frac{g_W^2}{480\pi^2 m_\Phi^2}$
$O_{3W}$	$\frac{g_W^2}{480\pi^2 m_\Phi^2}$
$O_6$	$-\frac{5\eta\kappa^2\lambda_\Phi}{8\pi^2 m_\Phi^4} - \frac{\eta^3}{8\pi^2 m_\Phi^2}$
$O_H$	$\frac{\eta^2}{16\pi^2 m_\Phi^2}$
$O_R$	$\frac{5\kappa^2\lambda_\Phi}{4\pi^2 m_\Phi^4}$
$O_T$	$\frac{5\kappa^2\lambda_\Phi}{8\pi^2 m_\Phi^4}$
$O_{WW}$	$\frac{\eta}{96\pi^2 m_\Phi^2}$
(c) ‘‘Warsaw’’ basis (tree level)	
$Q_H$	$-\frac{\eta\kappa^2}{m_\Phi^4}$
$Q_{H\Box}$	$\frac{\kappa^2}{m_\Phi^4}$
$Q_{HD}$	$-\frac{2\kappa^2}{m_\Phi^4}$
(d) ‘‘Warsaw’’ basis (1-loop level)	
$Q_H$	$-\frac{5\eta\kappa^2\lambda_\Phi}{8\pi^2 m_\Phi^4} - \frac{\eta^3}{8\pi^2 m_\Phi^2}$
$Q_{H\Box}$	$\frac{5\kappa^2\lambda_\Phi}{8\pi^2 m_\Phi^4}$
$Q_{HD}$	$-\frac{5\kappa^2\lambda_\Phi}{4\pi^2 m_\Phi^4} + \frac{\eta^2}{8\pi^2 m_\Phi^2}$
$Q_{HW}$	$\frac{\eta g_W^2}{96\pi^2 m_\Phi^2}$
$Q_W$	$\frac{g_W^3}{2880\pi^2 m_\Phi^2}$

$$\begin{aligned}
 & -\lambda_1|\tilde{H}|^2|\varphi|^2 - \lambda_2|\tilde{H}^\dagger\varphi|^2 \\
 & -\lambda_3\left[(\tilde{H}^\dagger\varphi)^2 + (\varphi^\dagger\tilde{H})^2\right] \tag{B2}
 \end{aligned}$$

Here the heavy field is  $\varphi$ . The internal quantum numbers and its other required properties are given in Table 7. Once the  $\varphi$  is integrated out using **CoDeX**, the effective operators up to dimension 6 for both ‘‘SILH’’ and ‘‘Warsaw’’ bases are generated, which are listed in Table 9.

**Table 9** Effective operators and Wilson coefficients for two Higgs doublet model (2HDM). The SILH basis result is available in Ref. [19]. The terms marked in ‘Red’ differ from the results derived in the mentioned reference. As light-heavy mixing (which can be ignored safely for phenomenological analyses) is not included, this result is incomplete

(a) ‘‘SILH’’ basis (tree level)	
$O_6$	$\frac{\eta_H^2}{m_\varphi^2}$
(b) ‘‘SILH’’ basis (1-loop level)	
$O_H$	$-\frac{3\eta_H\eta_\varphi}{8\pi^2 m_\varphi^2} + \frac{\lambda_1\lambda_2}{48\pi^2 m_\varphi^2} + \frac{\lambda_1^2}{48\pi^2 m_\varphi^2}$ $+ \frac{\lambda_2^2}{192\pi^2 m_\varphi^2} + \frac{\lambda_3^2}{48\pi^2 m_\varphi^2}$
$O_T$	$\frac{\lambda_2^2}{192\pi^2 m_\varphi^2} - \frac{\lambda_3^2}{48\pi^2 m_\varphi^2}$
$O_R$	$-\frac{3\eta_H\eta_\varphi}{8\pi^2 m_\varphi^2} + \frac{\lambda_2^2}{96\pi^2 m_\varphi^2} + \frac{\lambda_3^2}{24\pi^2 m_\varphi^2}$
$O_6$	$\frac{3\eta_H\eta_\varphi\lambda_1}{8\pi^2 m_\varphi^2} + \frac{3\eta_H\eta_\varphi\lambda_2}{8\pi^2 m_\varphi^2} - \frac{\lambda_1^3}{48\pi^2 m_\varphi^2} - \frac{\lambda_1^2\lambda_2}{32\pi^2 m_\varphi^2}$ $- \frac{\lambda_1\lambda_2^2}{32\pi^2 m_\varphi^2} - \frac{\lambda_2^3}{96\pi^2 m_\varphi^2} - \frac{\lambda_1\lambda_3^2}{8\pi^2 m_\varphi^2} - \frac{\lambda_2\lambda_3^2}{8\pi^2 m_\varphi^2}$ $+ \frac{3\eta_H^2\lambda_\varphi}{32\pi^2 m_\varphi^2}$
$O_{WW}$	$\frac{2\lambda_1+\lambda_2}{768\pi^2 m_\varphi^2}$
$O_{2W}$	$\frac{g_W^2}{960\pi^2 m_\varphi^2}$
$O_{3W}$	$\frac{g_W^2}{960\pi^2 m_\varphi^2}$
$O_{WB}$	$\frac{\lambda_2}{384\pi^2 m_\varphi^2}$
$O_{BB}$	$\frac{2\lambda_1+\lambda_2}{768\pi^2 m_\varphi^2}$
$O_{2B}$	$\frac{g_Y^2}{960\pi^2 m_\varphi^2}$
(c) ‘‘Warsaw’’ basis (tree level)	
$Q_H$	$\frac{\eta_H^2}{m_\varphi^2}$
(d) ‘‘Warsaw’’ basis (1-loop level)	
$Q_H$	$\frac{3\eta_H^2\lambda_\varphi}{32\pi^2 m_\varphi^2} + \frac{\eta_H^2}{m_\varphi^2} + \frac{3\eta_H\eta_\varphi\lambda_1}{8\pi^2 m_\varphi^2} + \frac{3\eta_H\eta_\varphi\lambda_2}{8\pi^2 m_\varphi^2} - \frac{\lambda_1^3}{48\pi^2 m_\varphi^2}$ $- \frac{\lambda_1^2\lambda_2}{32\pi^2 m_\varphi^2} - \frac{\lambda_1\lambda_2^2}{32\pi^2 m_\varphi^2} - \frac{\lambda_1\lambda_3^2}{8\pi^2 m_\varphi^2} - \frac{\lambda_2^3}{96\pi^2 m_\varphi^2}$ $- \frac{\lambda_2\lambda_3^2}{8\pi^2 m_\varphi^2}$
$Q_{H\Box}$	$-\frac{3\eta_H\eta_\varphi}{16\pi^2 m_\varphi^2} + \frac{\lambda_2^2}{192\pi^2 m_\varphi^2} + \frac{\lambda_3^2}{48\pi^2 m_\varphi^2}$
$Q_{HD}$	$\frac{9\eta_H\eta_\varphi}{8\pi^2 m_\varphi^2} + \frac{\lambda_1^2}{24\pi^2 m_\varphi^2} + \frac{\lambda_1\lambda_2}{24\pi^2 m_\varphi^2} + \frac{\lambda_2^2}{48\pi^2 m_\varphi^2}$ $+ \frac{\lambda_3^2}{12\pi^2 m_\varphi^2}$
$Q_{HW}$	$\frac{g_W^2\lambda_1}{384\pi^2 m_\varphi^2} + \frac{g_W^2\lambda_2}{768\pi^2 m_\varphi^2}$
$Q_{HWB}$	$\frac{g_W g_Y \lambda_2}{192\pi^2 m_\varphi^2}$
$Q_W$	$\frac{g_W^3}{5760\pi^2 m_\varphi^2}$

**Appendix C: Examples: Multiple heavy BSM fields**

**1. Electro-weak complex triplet ( $\Delta$ ) and complex doublet ( $\varphi$ ) with  $Y = 1$  models**

Complex doublet and complex triplet Lagrangian:

$$\mathcal{L}_{\text{BSM}} = \mathcal{L}_{\text{SM}} + Tr[(\mathcal{D}_\mu \Delta)^\dagger (\mathcal{D}^\mu \Delta)] - m_\Delta^2 Tr[\Delta^\dagger \Delta] + \mathcal{L}_Y - V(H, \Delta) + \mathcal{L}_2 + \mathcal{L}_{\text{int}}$$

where

$$V(H, \Delta) = \zeta_1 (H^\dagger H) Tr[\Delta^\dagger \Delta] + \zeta_2 (H^\dagger \tau^i H) Tr[\Delta^\dagger \tau^i \Delta]$$

**Table 10** Effective operators and Wilson coefficients for complex doublet and complex triplet models. These two fields are degenerate in mass, i.e.,  $m_\Delta = m_\varphi$ . As the light-heavy mixing (which can be ignored safely for phenomenological analyses) is not included, this result is incomplete

(a) ‘‘SILH’’ basis (tree level)

$O_6$	$-\frac{\xi_1 \mu^2}{m_\Delta^4} - \frac{\zeta_2 \mu^2}{4m_\Delta^4} + \frac{\eta_H^2}{m_\varphi^2}$
$O_H$	$\frac{\mu^2}{2m_\Delta^4}$
$O_R$	$\frac{\mu^2}{m_\Delta^4}$
$O_T$	$-\frac{\mu^2}{2m_\Delta^4}$

(b) ‘‘Warsaw’’ basis (tree level)

$Q_H$	$-\frac{\xi_1 \mu^2}{m_\Delta^4} - \frac{\zeta_2 \mu^2}{4m_\Delta^4} + \frac{\eta_H^2}{m_\varphi^2}$
$Q_{H\Box}$	$\frac{\mu^2}{2m_\Delta^4}$
$Q_{HD}$	$-\frac{2\mu^2}{m_\Delta^4}$
$Q_{ll}$	$\frac{y_\Delta^2}{4m_\Delta^2}$

(c) ‘‘SILH’’ basis (1-loop level)

$O_{2B}$	$\frac{7g_Y^2}{960\pi^2 m_\Delta^2}$
$O_{2W}$	$\frac{g_W^2}{192\pi^2 m_\Delta^2}$
$O_{3W}$	$\frac{g_W^2}{192\pi^2 m_\Delta^2}$
$O_6$	$-\frac{3\xi_1 \mu^2 \lambda_\varphi m_\Delta^2}{16\pi^2 m_\varphi^6} + \frac{3\xi_2 \mu^2 \lambda_\varphi m_\Delta^2}{64\pi^2 m_\varphi^6} + \frac{\xi_1 \lambda_1 \mu_1^2}{16\pi^2 m_\Delta^4} + \frac{\xi_1 \lambda_2 \mu_1^2}{24\pi^2 m_\Delta^4}$ $-\frac{5\xi_1 \mu_1^4}{96\pi^2 m_\Delta^6} + \frac{\xi_1^2 \mu_1^2}{8\pi^2 m_\Delta^4}$ $+\frac{\xi_2^2 \mu_1^2}{192\pi^2 m_\Delta^4} - \frac{\xi_1^3}{4\pi^2 m_\Delta^2} - \frac{\xi_1 \xi_2^2}{32\pi^2 m_\Delta^2} - \frac{9\mu^2 \mu_1^2 \lambda_\varphi}{256\pi^2 m_\Delta^2 m_\varphi^4}$ $-\frac{5\lambda_1 \mu_1^4}{192\pi^2 m_\Delta^6} - \frac{\lambda_2 \mu_1^4}{48\pi^2 m_\Delta^6} + \frac{\lambda_1^2 \mu_1^2}{32\pi^2 m_\Delta^4} + \frac{\lambda_2^2 \mu_1^2}{48\pi^2 m_\Delta^4}$ $+\frac{\lambda_3^2 \mu_1^2}{12\pi^2 m_\Delta^4} + \frac{\lambda_1 \lambda_2 \mu_1^2}{24\pi^2 m_\Delta^4} - \frac{\lambda_1^3}{48\pi^2 m_\Delta^2} - \frac{\lambda_2^3}{96\pi^2 m_\Delta^2}$ $-\frac{\lambda_1 \lambda_2^2}{32\pi^2 m_\Delta^2} - \frac{\lambda_1 \lambda_3^2}{8\pi^2 m_\Delta^2} - \frac{\lambda_2 \lambda_3^2}{8\pi^2 m_\Delta^2}$ $-\frac{\lambda_1^2 \lambda_2}{32\pi^2 m_\Delta^2} + \frac{3\mu_1^6}{320\pi^2 m_\Delta^8}$
$O_B$	$\frac{3\mu_1^2}{640\pi^2 m_\Delta^4}$
$O_{BB}$	$\frac{\xi_1}{32\pi^2 m_\Delta^2} + \frac{\lambda_1}{384\pi^2 m_\Delta^2} + \frac{\lambda_2}{768\pi^2 m_\Delta^2} - \frac{7\mu_1^2}{2560\pi^2 m_\Delta^4}$

$$+ [\mu(H^T i\sigma^2 \Delta^\dagger H) + h.c.], \tag{C1}$$

$$\mathcal{L}_Y = y_\Delta L^T C i\tau^2 \Delta L + h.c., \tag{C2}$$

$$\mathcal{L}_2 = |\mathcal{D}_\mu \varphi|^2 - m_\varphi^2 |\varphi|^2 - \frac{\lambda_\varphi}{4} |\varphi|^4 + (\eta_H |\tilde{H}|^2 + \eta_\varphi |\varphi|^2)(\tilde{H}^\dagger \varphi + \varphi^\dagger \tilde{H}) - \lambda_1 |\tilde{H}|^2 |\varphi|^2 - \lambda_2 |\tilde{H}^\dagger \varphi|^2 - \lambda_3 [(\tilde{H}^\dagger \varphi)^2 + (\varphi^\dagger \tilde{H})^2], \tag{C3}$$

$$\text{and, } \mathcal{L}_{\text{int}} = \mu_1 H^\dagger \Delta \varphi + h.c. \tag{C4}$$

Here the heavy fields  $\Delta$  and  $\varphi$  are the same as mentioned in an earlier section. The interaction term among



Table 10 continued

$O_D$	$\frac{\mu_1^2}{320\pi^2 m_\Delta^4}$
$O_H$	$-\frac{\zeta_1 \mu_1^2}{16\pi^2 m_\Delta^4} + \frac{\zeta_2^2}{8\pi^2 m_\Delta^2} - \frac{\lambda_1 \mu_1^2}{32\pi^2 m_\Delta^4} - \frac{7\lambda_2 \mu_1^2}{384\pi^2 m_\Delta^4} + \frac{\lambda_1^2}{48\pi^2 m_\Delta^2}$ $+\frac{\lambda_2^2}{192\pi^2 m_\Delta^2} + \frac{\lambda_3^2}{48\pi^2 m_\Delta^2} + \frac{\lambda_1 \lambda_2}{48\pi^2 m_\Delta^2} + \frac{7\mu_1^4}{384\pi^2 m_\Delta^6}$
$O_R$	$-\frac{\zeta_1 \mu_1^2}{32\pi^2 m_\Delta^4} + \frac{\zeta_2^2}{96\pi^2 m_\Delta^2} - \frac{\lambda_1 \mu_1^2}{64\pi^2 m_\Delta^4} - \frac{\lambda_2 \mu_1^2}{64\pi^2 m_\Delta^4} + \frac{\lambda_3^2}{96\pi^2 m_\Delta^2}$ $+\frac{\lambda_2^2}{24\pi^2 m_\Delta^2} + \frac{\mu_1^4}{64\pi^2 m_\Delta^6}$
$O_T$	$\frac{\zeta_2^2}{192\pi^2 m_\Delta^2} + \frac{\lambda_2 \mu_1^2}{384\pi^2 m_\Delta^4} + \frac{\lambda_2^2}{192\pi^2 m_\Delta^2}$ $-\frac{\lambda_3^2}{48\pi^2 m_\Delta^2} + \frac{\mu_1^4}{384\pi^2 m_\Delta^6}$
$O_W$	$\frac{\mu_1^2}{1920\pi^2 m_\Delta^4}$
$O_{WB}$	$-\frac{\zeta_2}{96\pi^2 m_\Delta^2} + \frac{\lambda_2}{384\pi^2 m_\Delta^2} + \frac{\mu_1^2}{3840\pi^2 m_\Delta^4}$
$O_{WW}$	$\frac{\zeta_1}{48\pi^2 m_\Delta^2} + \frac{\lambda_1}{384\pi^2 m_\Delta^2} + \frac{\lambda_2}{768\pi^2 m_\Delta^2} - \frac{11\mu_1^2}{2560\pi^2 m_\Delta^4}$
(d) ‘‘Warsaw’’ basis (1-loop level)	
$Q_H$	$-\frac{3\zeta_1 \mu^2 \lambda_\varphi m_\Delta^2}{16\pi^2 m_\varphi^6} + \frac{3\zeta_2 \mu^2 \lambda_\varphi m_\Delta^2}{64\pi^2 m_\varphi^6} + \frac{\zeta_1 \lambda_1 \mu_1^2}{16\pi^2 m_\Delta^4}$ $+\frac{\zeta_1 \lambda_2 \mu_1^2}{24\pi^2 m_\Delta^4} - \frac{5\zeta_1 \mu_1^4}{96\pi^2 m_\Delta^6}$ $+\frac{\zeta_1^2 \mu_1^2}{8\pi^2 m_\Delta^4} + \frac{\zeta_2^2 \mu_1^2}{192\pi^2 m_\Delta^4} - \frac{\zeta_1^3}{4\pi^2 m_\Delta^2} - \frac{\zeta_1 \zeta_2^2}{32\pi^2 m_\Delta^2}$ $-\frac{9\mu^2 \mu_1^2 \lambda_\varphi}{256\pi^2 m_\Delta^4 m_\varphi^4} - \frac{5\lambda_1 \mu_1^4}{192\pi^2 m_\Delta^6} - \frac{\lambda_2 \mu_1^4}{48\pi^2 m_\Delta^6} + \frac{\lambda_1^2 \mu_1^2}{32\pi^2 m_\Delta^4}$ $+\frac{\lambda_2^2 \mu_1^2}{48\pi^2 m_\Delta^4} + \frac{\lambda_3^2 \mu_1^2}{12\pi^2 m_\Delta^4} + \frac{\lambda_1 \lambda_2 \mu_1^2}{24\pi^2 m_\Delta^4} - \frac{\lambda_1^3}{48\pi^2 m_\Delta^2}$ $-\frac{\lambda_2^3}{96\pi^2 m_\Delta^2} - \frac{\lambda_1 \lambda_2^2}{32\pi^2 m_\Delta^2} - \frac{\lambda_1 \lambda_3^2}{8\pi^2 m_\Delta^2} - \frac{\lambda_2 \lambda_3^2}{8\pi^2 m_\Delta^2}$ $-\frac{\lambda_1^2 \lambda_2}{32\pi^2 m_\Delta^2} + \frac{3\mu_1^6}{320\pi^2 m_\Delta^8}$
$Q_{HB}$	$\frac{\zeta_1 g_Y^2}{32\pi^2 m_\Delta^2} + \frac{g_Y^2 \lambda_1}{384\pi^2 m_\Delta^2} + \frac{g_Y^2 \lambda_2}{768\pi^2 m_\Delta^2} - \frac{7g_Y^2 \mu_1^2}{2560\pi^2 m_\Delta^4}$
$Q_{H\Box}$	$-\frac{\zeta_1 \mu_1^2}{64\pi^2 m_\Delta^4} + \frac{\zeta_2^2}{192\pi^2 m_\Delta^2} - \frac{\lambda_1 \mu_1^2}{128\pi^2 m_\Delta^4} - \frac{\lambda_2 \mu_1^2}{128\pi^2 m_\Delta^4}$ $+\frac{\lambda_2^2}{192\pi^2 m_\Delta^2} + \frac{\lambda_3^2}{48\pi^2 m_\Delta^2} + \frac{\mu_1^4}{128\pi^2 m_\Delta^6}$
$Q_{HD}$	$-\frac{5\zeta_1 \mu_1^2}{32\pi^2 m_\Delta^4} + \frac{\zeta_1^2}{4\pi^2 m_\Delta^2} + \frac{\zeta_2^2}{96\pi^2 m_\Delta^2}$ $-\frac{5\lambda_1 \mu_1^2}{64\pi^2 m_\Delta^4} - \frac{5\lambda_2 \mu_1^2}{96\pi^2 m_\Delta^4} + \frac{\lambda_1^2}{24\pi^2 m_\Delta^2} + \frac{\lambda_2^2}{48\pi^2 m_\Delta^2}$ $+\frac{\lambda_3^2}{12\pi^2 m_\Delta^2} + \frac{\lambda_1 \lambda_2}{24\pi^2 m_\Delta^2} + \frac{5\mu_1^4}{96\pi^2 m_\Delta^6}$
$Q_{HW}$	$\frac{\zeta_1 g_W^2}{48\pi^2 m_\Delta^2} + \frac{g_W^2 \lambda_1}{384\pi^2 m_\Delta^2} + \frac{g_W^2 \lambda_2}{768\pi^2 m_\Delta^2} - \frac{37g_W^2 \mu_1^2}{7680\pi^2 m_\Delta^4}$
$Q_{HWB}$	$-\frac{\zeta_2 g_W g_Y}{48\pi^2 m_\Delta^2} + \frac{g_W g_Y \lambda_2}{192\pi^2 m_\Delta^2} + \frac{g_W g_Y \mu_1^2}{1920\pi^2 m_\Delta^4}$
$Q_{ll}$	$\frac{3y_\Delta^2 \lambda_\varphi m_\Delta^2}{128\pi^2 m_\varphi^4}$
$Q_W$	$\frac{g_W^3}{1152\pi^2 m_\Delta^2}$

**Table 11** Effective operators and Wilson coefficients for colored complex  $SU(2)_L$  doublet and singlets models. These two fields are degenerate in mass, i.e.,  $m_{\tilde{Q}_{3L}} = m_{\tilde{\tau}_R} = m$

(a) ‘‘SILH’’ basis (1-loop level)

$O_{2B}$	$\frac{g_s^2}{320\pi^2 m^2}$
$O_{2G}$	$\frac{g_s^2}{320\pi^2 m^2}$
$O_{2W}$	$\frac{g_W^2}{320\pi^2 m^2}$
$O_{3G}$	$\frac{g_s^3}{320\pi^2 m^2}$
$O_{3W}$	$\frac{g_W^3}{320\pi^2 m^2}$
$O_6$	$\frac{\bar{k}\lambda_L X_L^2}{16\pi^2 m^4} + \frac{3\bar{k}\lambda_L^2}{32\pi^2 m^2} + \frac{3\bar{k}^2\lambda_L}{32\pi^2 m^2} + \frac{\bar{k}X_L^4}{64\pi^2 m^6}$ $+ \frac{\bar{k}\lambda_R X_L^2}{32\pi^2 m^4} + \frac{\bar{k}^2 X_L^2}{32\pi^2 m^4} + \frac{\bar{k}^3}{32\pi^2 m^2} + \frac{3k^3}{32\pi^2 m^2} + \frac{9k^2\lambda_L}{32\pi^2 m^2}$ $+ \frac{9k\lambda_L^2}{32\pi^2 m^2} + \frac{\lambda_L X_L^4}{64\pi^2 m^6} + \frac{\lambda_L\lambda_R X_L^2}{32\pi^2 m^4} + \frac{\lambda_L^2 X_L^2}{32\pi^2 m^4} + \frac{\lambda_L^3}{8\pi^2 m^2}$ $+ \frac{X_L^6}{320\pi^2 m^8} + \frac{\lambda_R X_L^4}{64\pi^2 m^6} + \frac{\lambda_R^2 X_L^2}{32\pi^2 m^4} + \frac{\lambda_R^3}{16\pi^2 m^2}$
$O_{BB}$	$-\frac{\bar{k}}{2304\pi^2 m^2} - \frac{k}{2304\pi^2 m^2} - \frac{\lambda_L}{1152\pi^2 m^2}$ $-\frac{67X_L^2}{69120\pi^2 m^4} - \frac{\lambda_R}{144\pi^2 m^2}$
$O_D$	$\frac{X_L^2}{320\pi^2 m^4}$
$O_{GG}$	$-\frac{\bar{k}}{192\pi^2 m^2} - \frac{\lambda_L}{192\pi^2 m^2} - \frac{X_L^2}{384\pi^2 m^4} - \frac{\lambda_R}{384\pi^2 m^2}$
$O_H$	$\frac{\bar{k}\lambda_L}{8\pi^2 m^2} + \frac{3\bar{k}X_L^2}{128\pi^2 m^4} + \frac{\bar{k}^2}{64\pi^2 m^2} + \frac{3k\bar{k}}{32\pi^2 m^2}$ $+ \frac{k^2}{64\pi^2 m^2} + \frac{k\lambda_L}{8\pi^2 m^2} + \frac{5kX_L^2}{128\pi^2 m^4} + \frac{\lambda_L X_L^2}{16\pi^2 m^4}$ $+ \frac{\lambda_L^2}{8\pi^2 m^2} + \frac{7X_L^4}{640\pi^2 m^6} + \frac{\lambda_R X_L^2}{16\pi^2 m^4} + \frac{\lambda_R^2}{16\pi^2 m^2}$
$O_R$	$\frac{\bar{k}X_L^2}{32\pi^2 m^4} + \frac{\bar{k}^2}{32\pi^2 m^2} - \frac{k\bar{k}}{16\pi^2 m^2} + \frac{k^2}{32\pi^2 m^2}$ $+ \frac{kX_L^2}{64\pi^2 m^4} + \frac{3\lambda_L X_L^2}{64\pi^2 m^4} + \frac{3X_L^4}{320\pi^2 m^6} + \frac{\lambda_R X_L^2}{32\pi^2 m^4}$
$O_T$	$-\frac{\bar{k}X_L^2}{128\pi^2 m^4} + \frac{\bar{k}^2}{64\pi^2 m^2} + \frac{k\bar{k}}{32\pi^2 m^2}$ $+ \frac{k^2}{64\pi^2 m^2} + \frac{kX_L^2}{128\pi^2 m^4} + \frac{X_L^4}{640\pi^2 m^6}$
$O_{WB}$	$\frac{\bar{k}}{384\pi^2 m^2} - \frac{k}{384\pi^2 m^2} + \frac{11X_L^2}{11520\pi^2 m^4}$
$O_{WW}$	$-\frac{\bar{k}}{256\pi^2 m^2} - \frac{k}{256\pi^2 m^2} - \frac{\lambda_L}{128\pi^2 m^2} - \frac{X_L^2}{2560\pi^2 m^4}$
$O_W$	$\frac{X_L^2}{640\pi^2 m^4}$
$O_B$	$\frac{X_L^2}{640\pi^2 m^4}$

(b) ‘‘Warsaw’’ basis (1-loop level)

$Q_G$	$\frac{g_s^3}{1920\pi^2 m^2}$
$Q_H$	$\frac{\bar{k}\lambda_L X_L^2}{16\pi^2 m^4} + \frac{3\bar{k}\lambda_L^2}{32\pi^2 m^2} + \frac{3\bar{k}^2\lambda_L}{32\pi^2 m^2} + \frac{\bar{k}X_L^4}{64\pi^2 m^6}$ $+ \frac{\bar{k}\lambda_R X_L^2}{32\pi^2 m^4} + \frac{\bar{k}^2 X_L^2}{32\pi^2 m^4} + \frac{\bar{k}^3}{32\pi^2 m^2}$ $+ \frac{3k^3}{32\pi^2 m^2} + \frac{9k^2\lambda_L}{32\pi^2 m^2} + \frac{9k\lambda_L^2}{32\pi^2 m^2}$ $+ \frac{\lambda_L X_L^4}{64\pi^2 m^6} + \frac{\lambda_L\lambda_R X_L^2}{32\pi^2 m^4} + \frac{\lambda_L^2 X_L^2}{32\pi^2 m^4} + \frac{\lambda_L^3}{8\pi^2 m^2}$ $+ \frac{X_L^6}{320\pi^2 m^8} + \frac{\lambda_R X_L^4}{64\pi^2 m^6} + \frac{\lambda_R^2 X_L^2}{32\pi^2 m^4} + \frac{\lambda_R^3}{16\pi^2 m^2}$

**Table 11** continued

$Q_{HB}$	$-\frac{\tilde{k}g_Y^2}{2304\pi^2m^2} - \frac{kg_Y^2}{2304\pi^2m^2} - \frac{g_Y^2\lambda_L}{1152\pi^2m^2}$
$Q_{H\Box}$	$-\frac{67g_Y^2X_t^2}{69120\pi^2m^4} - \frac{g_Y^2\lambda_R}{144\pi^2m^2}$
$Q_{HD}$	$\frac{\tilde{k}X_t^2}{64\pi^2m^4} + \frac{\tilde{k}^2}{64\pi^2m^2} - \frac{k\tilde{k}}{32\pi^2m^2} + \frac{k^2}{64\pi^2m^2}$ $+ \frac{kX_t^2}{128\pi^2m^4} + \frac{3\lambda_L X_t^2}{128\pi^2m^4} + \frac{3X_t^4}{640\pi^2m^6} + \frac{\lambda_R X_t^2}{64\pi^2m^4}$
$Q_{HG}$	$\frac{\tilde{k}\lambda_L}{8\pi^2m^2} + \frac{5\tilde{k}X_t^2}{64\pi^2m^4} + \frac{\tilde{k}^2}{16\pi^2m^2} + \frac{3k^2}{16\pi^2m^2} + \frac{3k\lambda_L}{8\pi^2m^2}$ $+ \frac{5\lambda_L X_t^2}{64\pi^2m^4} + \frac{\lambda_L^2}{4\pi^2m^2} + \frac{X_t^4}{32\pi^2m^6} + \frac{5\lambda_R X_t^2}{64\pi^2m^4}$ $+ \frac{\lambda_R^2}{8\pi^2m^2}$
$Q_{HW}$	$-\frac{\tilde{k}g_S^2}{192\pi^2m^2} - \frac{g_S^2\lambda_L}{192\pi^2m^2} - \frac{g_S^2X_t^2}{384\pi^2m^4} - \frac{g_S^2\lambda_R}{384\pi^2m^2}$
$Q_{HWB}$	$-\frac{\tilde{k}g_W^2}{256\pi^2m^2} - \frac{kg_W^2}{256\pi^2m^2} - \frac{g_W^2\lambda_L}{128\pi^2m^2} - \frac{g_W^2X_t^2}{2560\pi^2m^4}$
$Q_W$	$\frac{\tilde{k}g_W g_Y}{192\pi^2m^2} - \frac{kg_W g_Y}{192\pi^2m^2} + \frac{11g_W g_Y X_t^2}{5760\pi^2m^4}$

these heavy fields is given in  $\mathcal{L}_{int}$ . Once the  $\Delta$  and  $\varphi$  are integrated out using **CoDeX**, the effective operators up to dimension 6 for both **"SILH"** and **"Warsaw"** bases are generated which are listed in Table 10.

2.  $SU(3)_C$  Colored and complex  $SU(2)_L$  doublet ( $\tilde{Q}_{3L}$ ) and singlet ( $\tilde{t}_R$ ) model

$$\mathcal{L}_{BSM} = \mathcal{L}_{SM} + \tilde{Q}_{3L}^\dagger (\tilde{k}\tilde{H}\tilde{H}^\dagger + kHH^\dagger + \lambda_L|H|^2)\tilde{Q}_{3L} + X_t\tilde{Q}_{3L}^\dagger\tilde{H}\tilde{t}_R + X_t\tilde{t}_R^\dagger\tilde{H}^\dagger\tilde{Q}_{3L} + \lambda_R\tilde{t}_R^\dagger|H|^2\tilde{t}_R. \tag{C5}$$

The details as regards the heavy fields ( $\tilde{Q}_{3L}$  and  $\tilde{t}_R$ ) are given in Table 7. The effective operators up to dimension 6 for both **"SILH"** and **"Warsaw"** bases are generated which are listed in Table 11. These are 1-loop results,

since the heavy fields ( $\tilde{Q}_{3L}$  and  $\tilde{t}_R$ ) are not linearly coupled to the SM fields in the given Lagrangian (see Eq. (C5)) and hence there is no tree-level Wilson coefficient.

4.1 Dimension-6 operators in Warsaw basis

Here we list all 59 operators in the **"Warsaw"** basis in Table 12.

4.2 Dimension-6 operators in "SILH" bases

Here we have provided the operators in **"SILH"** basis in Table 13.

**Table 12** Dimension-6 operators in "Warsaw" basis

	Scalar	$Q_{lq}^{(1)}$	$(\bar{l}\gamma_\mu l)(\bar{q}\gamma^\mu q)$		Scalar–Fermion
$Q_H$	$(H^\dagger H)^3$	$Q_{lq}^{(3)}$	$(\bar{l}\tau^a\gamma_\mu l)(\bar{q}\tau^a\gamma^\mu q)$	$Q_{eH}$	$(H^\dagger H)(\bar{l}eH) + \text{h.c.}$
$Q_{H\Box}$	$(H^\dagger H)\Box(H^\dagger H)$	$Q_{ee}$	$(\bar{e}\gamma^\mu e)(\bar{e}\gamma_\mu e)$	$Q_{uH}$	$(H^\dagger H)(\bar{q}u\tilde{H}) + \text{h.c.}$
$Q_{HD}$	$(H^\dagger\mathcal{D}_\mu H)^*(H^\dagger\mathcal{D}^\mu H)$	$Q_{uu}$	$(\bar{u}\gamma^\mu u)(\bar{u}\gamma_\mu u)$	$Q_{dH}$	$(H^\dagger H)(\bar{q}dH) + \text{h.c.}$
	Gauge Boson	$Q_{dd}$	$(\bar{d}\gamma^\mu d)(\bar{d}\gamma_\mu d)$	$Q_{Hl}^{(1)}$	$(H^\dagger i\overleftrightarrow{\mathcal{D}}_\mu H)(\bar{l}\gamma^\mu l)$
$Q_G$	$f^{abc}G_\rho^{a,\mu}G_\mu^{b,\nu}G_\nu^{c,\rho}$	$Q_{eu}$	$(\bar{e}\gamma^\mu e)(\bar{u}\gamma_\mu u)$	$Q_{Hl}^{(3)}$	$(H^\dagger i\tau^a\overleftrightarrow{\mathcal{D}}_\mu H)(\bar{l}\tau^a\gamma^\mu l)$
$Q_{\tilde{G}}$	$f^{abc}\tilde{G}_\rho^{a,\mu}\tilde{G}_\mu^{b,\nu}\tilde{G}_\nu^{c,\rho}$	$Q_{ed}$	$(\bar{e}\gamma^\mu e)(\bar{d}\gamma_\mu d)$	$Q_{He}$	$(H^\dagger i\overleftrightarrow{\mathcal{D}}_\mu H)(\text{bare}\gamma^\mu e)$
$Q_W$	$\epsilon^{abc}W_\rho^{a,\mu}W_\mu^{b,\nu}W_\nu^{c,\rho}$	$Q_{ud}^{(1)}$	$(\bar{u}\gamma^\mu u)(\bar{d}\gamma_\mu d)$	$Q_{Hq}^{(1)}$	$(H^\dagger i\overleftrightarrow{\mathcal{D}}_\mu H)(\bar{q}\gamma^\mu q)$
$Q_{\tilde{W}}$	$\epsilon^{abc}\tilde{W}_\rho^{a,\mu}\tilde{W}_\mu^{b,\nu}\tilde{W}_\nu^{c,\rho}$	$Q_{ud}^{(8)}$	$(\bar{u}\lambda^a\gamma^\mu u)(\bar{d}\lambda^a\gamma_\mu d)$	$Q_{Hq}^{(3)}$	$(H^\dagger i\tau^a\overleftrightarrow{\mathcal{D}}_\mu H)(\bar{q}\tau^a\gamma^\mu q)$
	Scalar–gauge boson	$Q_{le}$	$(\bar{l}\gamma^\mu l)(\bar{e}\gamma_\mu e)$	$Q_{Hu}$	$(H^\dagger i\overleftrightarrow{\mathcal{D}}_\mu H)(\bar{u}\gamma^\mu u)$
$Q_{HG}$	$(H^\dagger H)G_{\mu\nu}^aG^{a,\mu\nu}$	$Q_{lu}$	$(\bar{l}\gamma^\mu l)(\bar{u}\gamma_\mu u)$	$Q_{Hd}$	$(H^\dagger i\overleftrightarrow{\mathcal{D}}_\mu H)(\bar{d}\gamma^\mu d)$

**Table 12** continued

$Q_{H\tilde{G}}$	$(H^\dagger H) \tilde{G}_{\mu\nu}{}^a G^{a,\mu\nu}$	$Q_{\text{Id}}$	$(\bar{l}\gamma^\mu l)(\bar{d}\gamma_\mu d)$	$Q_{\text{Hud}}$	$(\tilde{H}^\dagger i \overleftrightarrow{\mathcal{D}}_\mu H)(\bar{u}\gamma^\mu d) + \text{h.c.}$
$Q_{\text{HW}}$	$(H^\dagger H) W_{\mu\nu}{}^a W^{a,\mu\nu}$	$Q_{\text{qe}}$	$(\bar{q}\gamma^\mu q)(\bar{e}\gamma_\mu e)$		Fermion–scalar–gauge boson
$Q_{H\tilde{W}}$	$(H^\dagger H) \tilde{W}_{\mu\nu}{}^a W^{a,\mu\nu}$	$Q_{\text{qu}}^{(1)}$	$(\bar{q}\gamma_\mu q)(\bar{u}\gamma^\mu u)$	$Q_{\text{cW}}$	$(\bar{l}\sigma^{\mu\nu} e)\tau^a H W_{\mu\nu}{}^a + \text{h.c.}$
$Q_{\text{HB}}$	$(H^\dagger H) B_{\mu\nu} B^{\mu\nu}$	$Q_{\text{qu}}^{(8)}$	$(\bar{q}\gamma_\mu \lambda^a q)(\bar{u}\gamma^\mu \lambda^a u)$	$Q_{\text{cB}}$	$(\bar{l}\sigma^{\mu\nu} e) H B_{\mu\nu} + \text{h.c.}$
$Q_{H\tilde{B}}$	$(H^\dagger H) \tilde{B}_{\mu\nu} B^{\mu\nu}$	$Q_{\text{qd}}^{(1)}$	$(\bar{q}\gamma_\mu q)(\bar{d}\gamma^\mu d)$	$Q_{\text{uG}}$	$(\bar{q}\sigma^{\mu\nu} \lambda^a u) \tilde{H} G_{\mu\nu}{}^a + \text{h.c.}$
$Q_{\text{HWB}}$	$(H^\dagger \tau^a H) W_{\mu\nu}{}^a B^{\mu\nu}$	$Q_{\text{qd}}^{(8)}$	$(\bar{q}\lambda^a \gamma^\mu q)(\bar{d}\lambda^a \gamma_\mu d)$	$Q_{\text{uW}}$	$(\bar{q}\sigma^{\mu\nu} u)\tau^a \tilde{H} W_{\mu\nu}{}^a + \text{h.c.}$
$Q_{H\tilde{W}B}$	$(H^\dagger \tau^a H) \tilde{W}_{\mu\nu}{}^a B^{\mu\nu}$	$Q_{\text{ledq}}$	$(\bar{l}^j e)(\bar{d}^j q) + \text{h.c.}$	$Q_{\text{uB}}$	$(\bar{q}\sigma^{\mu\nu} u) \tilde{H} B_{\mu\nu} + \text{h.c.}$
	Fermions	$Q_{\text{quqd}}^{(1)}$	$(\bar{q}^j u) \epsilon_{jk} (\bar{q}^k d) + \text{h.c.}$	$Q_{\text{dG}}$	$(\bar{q}\sigma^{\mu\nu} \lambda^a d) H G_{\mu\nu}{}^a + \text{h.c.}$
$Q_{\text{ll}}$	$(\bar{l}\gamma_\mu l)(\bar{l}\gamma^\mu l)$	$Q_{\text{quqd}}^{(8)}$	$(\bar{q}^j \lambda^a u) \epsilon_{jk} (\bar{q}^k \lambda^a d) + \text{h.c.}$	$Q_{\text{dW}}$	$(\bar{q}\sigma^{\mu\nu} d)\tau^a H W_{\mu\nu}{}^a + \text{h.c.}$
$Q_{\text{qq}}^{(1)}$	$(\bar{q}\gamma_\mu q)(\bar{q}\gamma^\mu q)$	$Q_{\text{lequ}}^{(1)}$	$(\bar{l}^j e) \epsilon_{jk} (\bar{q}^k u) + \text{h.c.}$	$Q_{\text{dB}}$	$(\bar{q}\sigma^{\mu\nu} d) H B_{\mu\nu} + \text{h.c.}$
$Q_{\text{qq}}^{(3)}$	$(\bar{q}\lambda^a \gamma_\mu q)(\bar{q}\lambda^a \gamma^\mu q)$	$Q_{\text{lequ}}^{(3)}$	$(\bar{l}^j \sigma_{\mu\nu} e) \epsilon_{jk} (\bar{q}^k \sigma_{\mu\nu} d) + \text{h.c.}$		

**Table 13** Dimension-6 operators in ‘‘SILH’’ basis

$O_{GG}$	$g_s^2 (H^\dagger H) G_{\mu\nu}{}^a G^{a,\mu\nu}$	$O_H$	$\frac{1}{2} (\partial_\mu (H^\dagger H))^2$
$O_{\text{WW}}$	$g_W^2 (H^\dagger H) W_{\mu\nu}{}^a W^{a,\mu\nu}$	$O_T$	$\frac{1}{2} (H^\dagger \overleftrightarrow{\mathcal{D}}_\mu H)^2$
$O_{\text{BB}}$	$g_Y^2 (H^\dagger H) B_{\mu\nu} B^{\mu\nu}$	$O_R$	$(H^\dagger H) (\mathcal{D}_\mu H)^\dagger (\mathcal{D}^\mu H)$
$O_{\text{WB}}$	$2g_W g_Y (H^\dagger \tau^a H) (W_{\mu\nu}{}^a B^{\mu\nu})$	$O_D$	$(\mathcal{D}^2 H)^\dagger (\mathcal{D}^2 H)$
$O_W$	$i g_W (H^\dagger \tau^a \overleftrightarrow{\mathcal{D}}^\mu H) (\mathcal{D}^\nu W_{\mu\nu}{}^a)$	$O_6$	$(H^\dagger H)^3$
$O_B$	$\frac{i}{2} g_Y (H^\dagger \overleftrightarrow{\mathcal{D}}^\mu H) (\partial^\nu B_{\mu\nu})$	$O_{2G}$	$-\frac{1}{2} (\mathcal{D}^\mu G_{\mu\nu}{}^a)^2$
$O_{3G}$	$\frac{g_s}{3!} f^{abc} G_\rho{}^{a,\mu} G_\mu{}^{b,\nu} G_\nu{}^{c,\rho}$	$O_{2W}$	$-\frac{1}{2} (\mathcal{D}^\mu W_{\mu\nu}{}^a)^2$
$O_{3W}$	$\frac{g_W}{3!} \epsilon^{abc} W_\rho{}^{a,\mu} W_\mu{}^{b,\nu} W_\nu{}^{c,\rho}$	$O_{2B}$	$-\frac{1}{2} (\partial^\mu B_{\mu\nu})^2$

**References**

- G.F. Giudice, C. Grojean, A. Pomarol, R. Rattazzi, JHEP **06**, 045 (2007). [arXiv:hep-ph/0703164](#) [hep-ph]
- R. Contino, M. Ghezzi, C. Grojean, M. Muhlleitner, M. Spira, JHEP **07**, 035 (2013). [arXiv:1303.3876](#) [hep-ph]
- B. Grzadkowski, M. Iskrzynski, M. Misiak, J. Rosiek, JHEP **10**, 085 (2010). [arXiv:1008.4884](#) [hep-ph]
- E.E. Jenkins, A.V. Manohar, M. Trott, JHEP **10**, 087 (2013). [arXiv:1308.2627](#) [hep-ph]
- E.E. Jenkins, A.V. Manohar, M. Trott, JHEP **01**, 035 (2014). [arXiv:1310.4838](#) [hep-ph]
- R. Alonso, E.E. Jenkins, A.V. Manohar, M. Trott, JHEP **04**, 159 (2014). [arXiv:1312.2014](#) [hep-ph]
- R.J. Furnstahl, N. Klco, D.R. Phillips, S. Wesolowski, Phys. Rev. C **92**, 024005 (2015). [arXiv:1506.01343](#) [nucl-th]
- H. Georgi, Ann. Rev. Nucl. Part. Sci. **43**, 209 (1993)
- D.B. Kaplan, in *Beyond the standard model 5. Proceedings, 5th Conference, Balholm, Norway, April 29-May 4, 1997* 1995. [arXiv:nucl-th/9506035](#) [nucl-th]
- A.V. Manohar, *Perturbative and nonperturbative aspects of quantum field theory. Proceedings, 35. Internationale Universitätswochen für Kern- und Teilchenphysik: Schladming, Austria, March 2-9, 1996*, Lect. Notes Phys. **479**, 311 (1997). [arXiv:hep-ph/9606222](#) [hep-ph]
- C.P. Burgess, Ann. Rev. Nucl. Part. Sci. **57**, 329 (2007). [arXiv:hep-th/0701053](#) [hep-th]
- I.Z. Rothstein, [arXiv:hep-ph/0308266](#) [hep-ph] (2003)
- B. Gripaios, D. Sutherland, [arXiv:1807.07546](#) [hep-ph] (2018)
- A. Falkowski, B. Fuks, K. Mawatari, K. Mimasu, F. Riva, V. Sanz, Eur. Phys. J. C **75**, 583 (2015). [arXiv:1508.05895](#) [hep-ph]
- A. Celis, J. Fuentes-Martin, A. Vicente, J. Virto, Eur. Phys. J. C **77**, 405 (2017). [arXiv:1704.04504](#) [hep-ph]
- J.C. Criado, Comput. Phys. Commun. **227**, 42 (2018). [arXiv:1710.06445](#) [hep-ph]
- J. Aebischer, J. Kumar, D.M. Straub, [arXiv:1804.05033](#) [hep-ph] (2018)
- J. Aebischer et al., Comput. Phys. Commun. **232**, 71 (2018b). [arXiv:1712.05298](#) [hep-ph]
- B. Henning, X. Lu, H. Murayama, JHEP **01**, 023 (2016). [arXiv:1412.1837](#) [hep-ph]
- B. Henning, X. Lu, H. Murayama, JHEP **01**, 123 (2018). [arXiv:1604.01019](#) [hep-ph]
- B. Henning, X. Lu, T. Melia, H. Murayama, JHEP **08**, 016 (2017). [arXiv:1512.03433](#) [hep-ph]
- J. Elias-Miro, J.R. Espinosa, E. Masso, A. Pomarol, JHEP **11**, 066 (2013). [arXiv:1308.1879](#) [hep-ph]
- J. Elias-Miró, C. Grojean, R.S. Gupta, D. Marzocca, JHEP **05**, 019 (2014). [arXiv:1312.2928](#) [hep-ph]
- E.E. Jenkins, A.V. Manohar, M. Trott, JHEP **09**, 063 (2013b). [arXiv:1305.0017](#) [hep-ph]
- A.V. Manohar, Phys. Lett. B **726**, 347 (2013). [arXiv:1305.3927](#) [hep-ph]
- Z. Han, W. Skiba, Phys. Rev. D **71**, 075009 (2005). [arXiv:hep-ph/0412166](#) [hep-ph]
- G. Cacciapaglia, C. Csaki, G. Marandella, A. Strumia, Phys. Rev. D **74**, 033011 (2006). [arXiv:hep-ph/0604111](#) [hep-ph]
- F. Bonnet, M.B. Gavela, T. Ota, W. Winter, Phys. Rev. D **85**, 035016 (2012a). [arXiv:1105.5140](#) [hep-ph]

29. F. Bonnet, T. Ota, M. Rauch, W. Winter, Phys. Rev. D **86**, 093014 (2012b). [arXiv:1207.4599](#) [hep-ph]
30. F. del Aguila, J. de Blas, *Elementary particle physics and gravity. Proceedings, 10th Hellenic Schools and Workshops, Corfu 2010, Corfu, August 29-September 12, 2010*, Fortsch. Phys. **59**, 1036 (2011). [arXiv:1105.6103](#) [hep-ph]
31. C. Grojean, E.E. Jenkins, A.V. Manohar, M. Trott, JHEP **04**, 016 (2013). [arXiv:1301.2588](#) [hep-ph]
32. R. Contino, A. Falkowski, F. Goertz, C. Grojean, F. Riva, JHEP **07**, 144 (2016). [arXiv:1604.06444](#) [hep-ph]
33. J. Brehmer, A. Freitas, D. Lopez-Val, T. Plehn, Phys. Rev. D **93**, 075014 (2016). [arXiv:1510.03443](#) [hep-ph]
34. L. Berthier, M. Trott, JHEP **02**, 069 (2016). [arXiv:1508.05060](#) [hep-ph]
35. M. Gorbahn, J.M. No, V. Sanz, JHEP **10**, 036 (2015). [arXiv:1502.07352](#) [hep-ph]
36. L. Berthier, M. Trott, JHEP **05**, 024 (2015). [arXiv:1502.02570](#) [hep-ph]
37. W. Skiba, in *Physics of the large and the small, TASI 09, Proceedings of the Theoretical Advanced Study Institute in Elementary Particle Physics, Boulder, Colorado, USA, 1-26 June 2009* pp. 5–70, (2011). [arXiv:1006.2142](#) [hep-ph]
38. Z.U. Khandker, D. Li, W. Skiba, Phys. Rev. D **86**, 015006 (2012). [arXiv:1201.4383](#) [hep-ph]
39. C. Englert, E. Re, M. Spannowsky, Phys. Rev. D **88**, 035024 (2013). [arXiv:1306.6228](#) [hep-ph]
40. S. Banerjee, S. Mukhopadhyay, B. Mukhopadhyaya, JHEP **10**, 062 (2012). [arXiv:1207.3588](#) [hep-ph]
41. S. Banerjee, S. Mukhopadhyay, B. Mukhopadhyaya, Phys. Rev. D **89**, 053010 (2014). [arXiv:1308.4860](#) [hep-ph]
42. J.D. Wells, Z. Zhang, JHEP **06**, 122 (2016a). [arXiv:1512.03056](#) [hep-ph]
43. M.K. Gaillard, Nucl. Phys. B **268**, 669 (1986)
44. O. Cheyette, Phys. Rev. Lett. **55**, 2394 (1985)
45. L. Lehman, A. Martin, Phys. Rev. D **91**, 105014 (2015). [arXiv:1503.07537](#) [hep-ph]
46. C.-W. Chiang, R. Huo, JHEP **09**, 152 (2015). [arXiv:1505.06334](#) [hep-ph]
47. R. Huo, JHEP **09**, 037 (2015). [arXiv:1506.00840](#) [hep-ph]
48. R. Huo, Phys. Rev. D **97**, 075013 (2018). [arXiv:1509.05942](#) [hep-ph]
49. L. Lehman, A. Martin, JHEP **02**, 081 (2016). [arXiv:1510.00372](#) [hep-ph]
50. J.D. Wells, Z. Zhang, JHEP **01**, 123 (2016b). [arXiv:1510.08462](#) [hep-ph]
51. A. Drozd, J. Ellis, J. Quevillon, T. You, JHEP **03**, 180 (2016). [arXiv:1512.03003](#) [hep-ph]
52. F. del Aguila, Z. Kunszt, J. Santiago, Eur. Phys. J. C **76**, 244 (2016). [arXiv:1602.00126](#) [hep-ph]
53. J. Fuentes-Martin, J. Portoles, P. Ruiz-Femenia, JHEP **09**, 156 (2016). [arXiv:1607.02142](#) [hep-ph]
54. S.A.R. Ellis, J. Quevillon, T. You, Z. Zhang, Phys. Lett. B **762**, 166 (2016). [arXiv:1604.02445](#) [hep-ph]
55. A. Alloul, N.D. Christensen, C. Degrande, C. Duhr, B. Fuks, Comput. Phys. Commun. **185**, 2250 (2014). [arXiv:1310.1921](#) [hep-ph]
56. M. Schunter, “Texttableform, Version 1.0,” Teichstr, 1 (2018)
57. W.R. Inc., “Mathematica, Version 8.0,” Champaign, IL (2010)
58. S. Das Bakshi, J. Chakraborty, S. Patra, In preparation