MatchingDB: a format for matching dictionaries

Juan Carlos Criado

University of Granada

Matching dictionaries

1. Construct a generic UV completion. Include all possible new fields and couplings that contribute at a given order in the EFT.
2. Integrate out the new fields.

$\Longrightarrow$ Matching to the EFT done **once and for all**,
no new matching calculations are needed at the given $1/\Lambda$ and loop order.

- `J. de Blas, JCC, M. Perez-Victoria, J. Santiago [1711.10391]`
  Tree-level matching to dim-6 SMEFT.
- `G. Guedes, P. Olgoso, J. Santiago [2303.16965]`
  One-loop matching to dim-6 SMEFT, for scalars and fermions.
- Dim. 8, one-loop for vectors, higher-spin, other EFTs, ...?

Matching tools

Tree-level and one-loop matching are automatized now:

`MatchingTools`, `CoDEx`, `Matchete`, `Matchmakereft`, . . .

Why matching dicts?

- Classification of all UV completions contributing to a given order.
- Easiness of use.

Using a matching dictionary (tree SMEFT dim-6 example)

UV theory

$$
\begin{aligned}
-\mathcal{L}^{(4)}_{\text{leptons}} = {} & (\lambda_N)_{ri}\bar{N}_{Rr}\tilde{\phi}^\dagger l_{Li} + (\lambda_E)_{ri}\bar{E}_{Rr}\phi^\dagger l_{Li} \\
& + (\lambda_{\Delta_1})_{ri}\bar{\Delta}_{1Lr}\phi e_{Ri} + (\lambda_{\Delta_3})_{ri}\bar{\Delta}_{3Lr}\tilde{\phi} e_{Ri} \\
& + \frac{1}{2}(\lambda_\Sigma)_{ri}\bar{\Sigma}^a_{Rr}\tilde{\phi}^\dagger\sigma^a l_{Li} + \frac{1}{2}(\lambda_{\Sigma_1})_{ri}\bar{\Sigma}^a_{1Rr}\phi^\dagger\sigma^a l_{Li} \\
& + (\lambda_{N\Delta_1})_{rs}\bar{N}^c_{Rr}\phi^\dagger\Delta_{1Rs} + (\lambda_{E\Delta_1})_{rs}\bar{E}_{Lr}\phi^\dagger\Delta_{1Rs} \\
& + (\lambda_{E\Delta_3})_{rs}\bar{E}_{Lr}\tilde{\phi}^\dagger\Delta_{3Rs} + \frac{1}{2}(\lambda_{\Sigma\Delta_1})_{rs}\bar{\Sigma}^{ca}_{Rr}\tilde{\phi}^\dagger\sigma^a\Delta_{1Rs} \\
& + \frac{1}{2}(\lambda_{\Sigma_1\Delta_1})_{rs}\bar{\Sigma}^a_{1Lr}\phi^\dagger\sigma^a\Delta_{1Rs} + \frac{1}{2}(\lambda_{\Sigma_1\Delta_3})_{rs}\bar{\Sigma}^a_{1Lr}\tilde{\phi}^\dagger\sigma^a\Delta_{3Rs} + \text{h.c.},
\end{aligned}
$$

Top-down

UV field → EFT operator tables

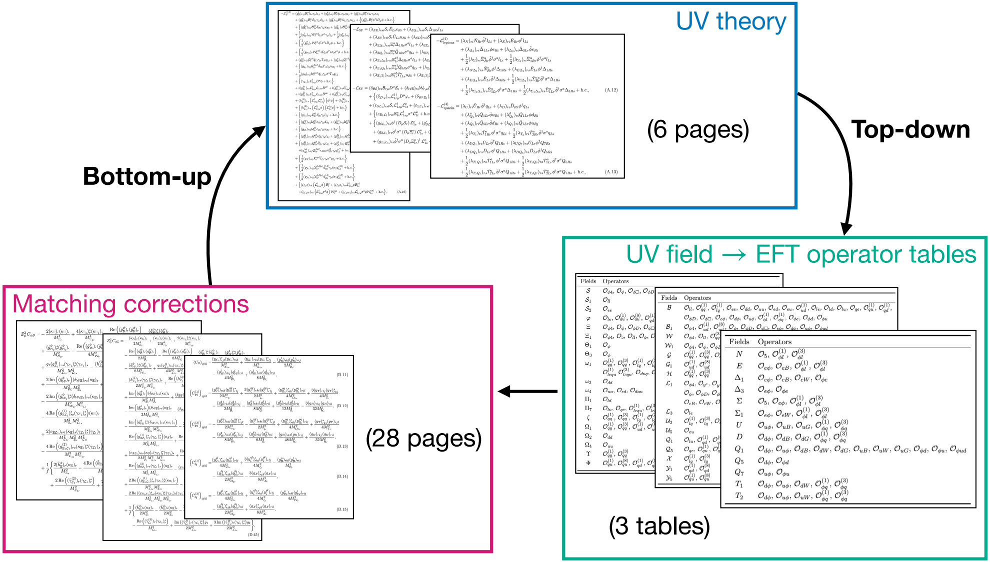
Fields	Operators
N	$\mathcal{O}_5$, $\mathcal{O}^{(1)}_{\phi l}$, $\mathcal{O}^{(3)}_{\phi l}$
E	$\mathcal{O}_{e\phi}$, $\mathcal{O}_{eB}$, $\mathcal{O}^{(1)}_{\phi l}$, $\mathcal{O}^{(3)}_{\phi l}$
Δ_1	$\mathcal{O}_{e\phi}$, $\mathcal{O}_{eB}$, $\mathcal{O}_{eW}$, $\mathcal{O}_{\phi e}$
Δ_3	$\mathcal{O}_{e\phi}$, $\mathcal{O}_{\phi e}$
Σ	$\mathcal{O}_5$, $\mathcal{O}_{e\phi}$, $\mathcal{O}^{(1)}_{\phi l}$, $\mathcal{O}^{(3)}_{\phi l}$
Σ_1	$\mathcal{O}_{e\phi}$, $\mathcal{O}_{eW}$, $\mathcal{O}^{(1)}_{\phi l}$, $\mathcal{O}^{(3)}_{\phi l}$
U	$\mathcal{O}_{u\phi}$, $\mathcal{O}_{uB}$, $\mathcal{O}_{uG}$, $\mathcal{O}^{(1)}_{\phi q}$, $\mathcal{O}^{(3)}_{\phi q}$
D	$\mathcal{O}_{d\phi}$, $\mathcal{O}_{dB}$, $\mathcal{O}_{dG}$, $\mathcal{O}^{(1)}_{\phi q}$, $\mathcal{O}^{(3)}_{\phi q}$
Q_1	$\mathcal{O}_{d\phi}$, $\mathcal{O}_{u\phi}$, $\mathcal{O}_{dB}$, $\mathcal{O}_{dW}$, $\mathcal{O}_{dG}$, $\mathcal{O}_{uB}$, $\mathcal{O}_{uW}$, $\mathcal{O}_{uG}$, $\mathcal{O}_{\phi d}$, $\mathcal{O}_{\phi u}$, $\mathcal{O}_{\phi ud}$
Q_5	$\mathcal{O}_{d\phi}$, $\mathcal{O}_{\phi d}$
Q_7	$\mathcal{O}_{u\phi}$, $\mathcal{O}_{\phi u}$
T_1	$\mathcal{O}_{d\phi}$, $\mathcal{O}_{u\phi}$, $\mathcal{O}_{dW}$, $\mathcal{O}^{(1)}_{\phi q}$, $\mathcal{O}^{(3)}_{\phi q}$
T_2	$\mathcal{O}_{d\phi}$, $\mathcal{O}_{u\phi}$, $\mathcal{O}_{uW}$, $\mathcal{O}^{(1)}_{\phi q}$, $\mathcal{O}^{(3)}_{\phi q}$

Matching corrections

$$
\begin{aligned}
Z_\phi\left(C^{(1)}_{\phi l}\right)_{ij} = {} & \frac{(\lambda_N)^*_{ri}(\lambda_N)_{rj}}{4M^2_{N_r}} - \frac{(\lambda_E)_{rj}(\lambda_E)^*_{ri}}{4M^2_{E_r}} + \frac{3(\lambda_\Sigma)^*_{ri}(\lambda_\Sigma)_{rj}}{16M^2_{\Sigma_r}} - \frac{3(\lambda_{\Sigma_1})_{rj}(\lambda_{\Sigma_1})^*_{ri}}{16M^2_{\Sigma_{1r}}} \\
& - \frac{\mathrm{Re}\left((\hat{g}^\phi_{\mathcal{B}})_r\right)(g^l_{\mathcal{B}})_{rij}}{M^2_{\mathcal{B}_r}} - \frac{g_1\delta_{ij}(g^B_{\mathcal{L}_1})_{rs}(\gamma_{\mathcal{L}_1})^*_r(\gamma_{\mathcal{L}_1})_s}{4M^2_{\mathcal{L}_{1r}}M^2_{\mathcal{L}_{1s}}} \\
& + \frac{i(\lambda_N)_{rj}(z_{N\mathcal{L}_1})^*_{rsi}(\gamma_{\mathcal{L}_1})^*_s}{4M_{N_r}M^2_{\mathcal{L}_{1s}}} - \frac{i(\lambda_N)^*_{ri}(z_{N\mathcal{L}_1})_{rsj}(\gamma_{\mathcal{L}_1})_s}{4M_{N_r}M^2_{\mathcal{L}_{1s}}}
\end{aligned}
$$

Bottom-up

Dictionary size (tree SMEFT dim-6 example)



MatchingDB

MatchingDB Python interface

Load the tree-level dictionary for the SMEFT at dim. 6:

```
from matchingdb import JsonDB
db = JsonDB.load("smeft_dim6_tree.json")
```

MatchingDB Python interface

Get information about the coefficient of the $\mathcal{O}_{ll}$ Warsaw-basis operator:

```
db.select_terms(coefficient="ll", output_format="pandas")
```

```
  coefficient  fields      couplings
0          ll    [S1]     [yS1, yS1]
1          ll   [Xi1]   [yXi1, yXi1]
2          ll     [B]     [g1B, g1B]
3          ll     [W]     [g1W, g1W]
4          ll     [W]     [g1W, g1W]
```

MatchingDB Python interface

Get information about the $\mathcal{S}_1$ field:

```
db.select_fields(name="S1", output_format="pandas")
```

```
    name    real  representation
0   S1     False         S(1,1,1)
```

List all terms generated by this field:

```
db.select_terms(fields=["S1"], output_format="pandas")
```

```
  coefficient  fields    couplings
0          ll    [S1]   [yS1, yS1]
```

MatchingDB Python interface

Get information about the Ξ_1 field:

```
db.select_fields(name="Xi1", output_format="pandas")
```

```
     name    real  representation
0    Xi1     False       S(1,3,1)
```

List all terms generated by this field:

```
db.select_terms(fields=["Xi1"], output_format="pandas")
```

```
   coefficient  fields                                  couplings
0     llphiphi   [Xi1]                          [kappaXi1, yXi1]
1         dphi   [Xi1]                [Yd, kappaXi1, kappaXi1]
...
9           ll   [Xi1]                              [yXi1, yXi1]
...
```

MatchingDB Python interface

Obtain a LaTeX representation for the Wilson coefficients:

```
from IPython.display import Math
Math(db.select_terms(fields=["S1"], output_format="latex")["ll"])
```

$$+\frac{\left(y_{\mathcal{S}_1}\right)^*_{ajl}\left(y_{\mathcal{S}_1}\right)_{aik}}{M^2_{\mathcal{S}_1,a}}$$

MatchingDB Python interface

Obtain a LaTeX representation for the Wilson coefficients:

```
from IPython.display import Math
Math(db.select_terms(fields=["Xi1"], output_format="latex")["ephi"])
```

$$+\frac{(-1)i\left(\hat{y}^e\right)^*_{ci}\left(z_{\Delta_1\mathcal{L}_1}\right)^*_{abc}\left(\gamma_{\mathcal{L}_1}\right)^*_b\left(\lambda_{\Delta_1}\right)_{aj}}{2M^2_{\mathcal{L}_1,b}M_{\Delta_1,a}}+\frac{(-1)i\left(\hat{y}^e\right)^*_{ci}\left(z_{\Delta_1\mathcal{L}_1}\right)_{abj}\left(\lambda_{\Delta_1}\right)^*_{ac}\left(\gamma_{\mathcal{L}_1}\right)_b}{2M_{\Delta_1,a}M^2_{\mathcal{L}_1,b}}$$

MatchingDB Python interface

Plugging in numerical values for UV parameters into the ll Wilson coefficient:

```
evaluator = db.select_terms(
    fields=["S1"], output_format="numeric", parameters={"yS1", "M_S1"},
)

from numpy.random import random
n = 2  # number of S1 flavors
parameters = {"yS1": random(size=(n, 3, 3)), "M_S1": random(size=(n,))}

evaluator(parameters, expand_flavor=True)
```

```
{'ll_0000': ..., 'll_0001': ..., 'll_0002': ...,  ...}
```

The output here is in WCxf format.

MatchingDB: definition and features

A unified format for matching dictionaries in EFTs

- Matching results up to one-loop order.
- Stores info about the UV theory, including the heavy fields that have been integrated out and their couplings.
- Easily access:
  - Bottom-up info: list UV fields and couplings that generate a given effective operator.
  - Top-down info: list effective operators generated by some set of fields.

MatchingDB: documentation and tools

MatchingDB data can be stored as **JSON** or **SQLite**.

`gitlab.com/jccriado/matchingdb/` **includes:**

- Machine-readable definition, to be used to check any given dictionary.
- Human-readable descriptions of the both the JSON and SQLite versions.
- The Python interface.
- The collection of dictionaries (currently just the tree-dim-6 one).

Example JSON data


```
{
  "fields": [
    {"name": "E", "real": false, "representation": "F(1,1,-1)", ...}
  ],
  "couplings": [
    {"name": "lambda_tilde", "fields": ["E"], "real": false, ...},
    {"name": "g1", "fields": [], "real": true, "latex": "g\_1", ...},
    {"name": "g2", "fields": [], "real": true, "latex": "g\_2", ...},
    {"name": "Ye", "fields": [], "real": false, "latex": "Y\_e", ...}
  ],
  "constants": [
    {"name": "pi", "value": 3.141592653589793, "latex": "\\pi"},
    {"name": "kdelta", "value": [[1, 0, 0], [0, 1, 0], [0, 0, 1]], ...}
  ],
  "terms": [...]
}
```

Example JSON data for a term in a Wilson coefficient

```json
{
  "coefficient": "phie",
  "fields": ["E"],
  "factors": [
    ["numerical", 2, 1],
    ["numerical", 240, -1],
    ["constant", "pi", -2, []],
    ["constant", "kdelta", 1, ["i", "j"]],
    ["coupling", "g1", 4, false, []],
    ["mass", "E", "a", -2]
  ],
  "free_indices": ["i", "j"]
}
```

MatchingDB
JSON schema

MatchingDB (object)

fields	array of *field*
couplings	array of *coupling*
constants	array of *constant*
terms	array of *term*

field (object)

name	string
real	boolean
representation	string
latex	string

coupling (object)

name	string
fields	array of string
real	boolean
latex	string
latex_interaction	string

constant (object)

name	string
value	nested array of number or {“Re”: …, “Im”: …}
latex	string

term (object)

coefficient	string
fields	array of string
real	boolean
factors	array of *_factor
latex	string

x^n → *numerical_factor* (tuple)

number	number
base	exponent

$k^n_{ij\dots}$ → *constant_factor* (tuple)

string	number	array of string
name	exponent	indices

$g^{n(*)}_{ij\dots}$ → *coupling_factor* (tuple)

string	number	boolean	array of string
name	exponent	conjugated	indices

$M^n_{F,i}$ → *mass_factor* (tuple)

string	string	number
field name	field index	exponent

$(M^2_{F,i} - M^2_{G,j})^n$ → *mass_difference_factor* (tuple)

string	string	string	string	number
field1 name	field1 index	field2 name	field2 index	exponent

$\log(M^n_{F,i}/\mu^n)$ → *log_mass_factor* (tuple)

string	string	number
field name	field index	exponent

$\log(M^n_{F,i}/M^n_{G,j})$ → *log_mass_ratio_factor* (tuple)

string	string	string	string	number
field1 name	field1 index	field2 name	field2 index	exponent

Conclusions

A tool-agnostic format for generic matching results up to one loop, making matching dictionaries easier to share and use:

- Sharing results from different tools in a unified way.
- Quick exploration of the data:
  - Effective operators $\rightarrow$ list of UV models
  - UV model $\rightarrow$ list of effective operators
- Output LaTeX expressions for Wilson coefficients and UV Lagrangian.
- Output numerical data in WCxf-like format, ready to use by other tools.