Automated **matching dictionaries**

SMEFT-Tools 2022

Work in progress with Jose Santiago:
gitlab.com/jccriado/matchingdb

Juan Carlos Criado, IPPP Durham

J. de Blas, J. C. Criado, M. Perez-Victoria, J. Santiago, 1711.10391

Dim-6 SMEFT ⟷ Weakly coupled UV completions

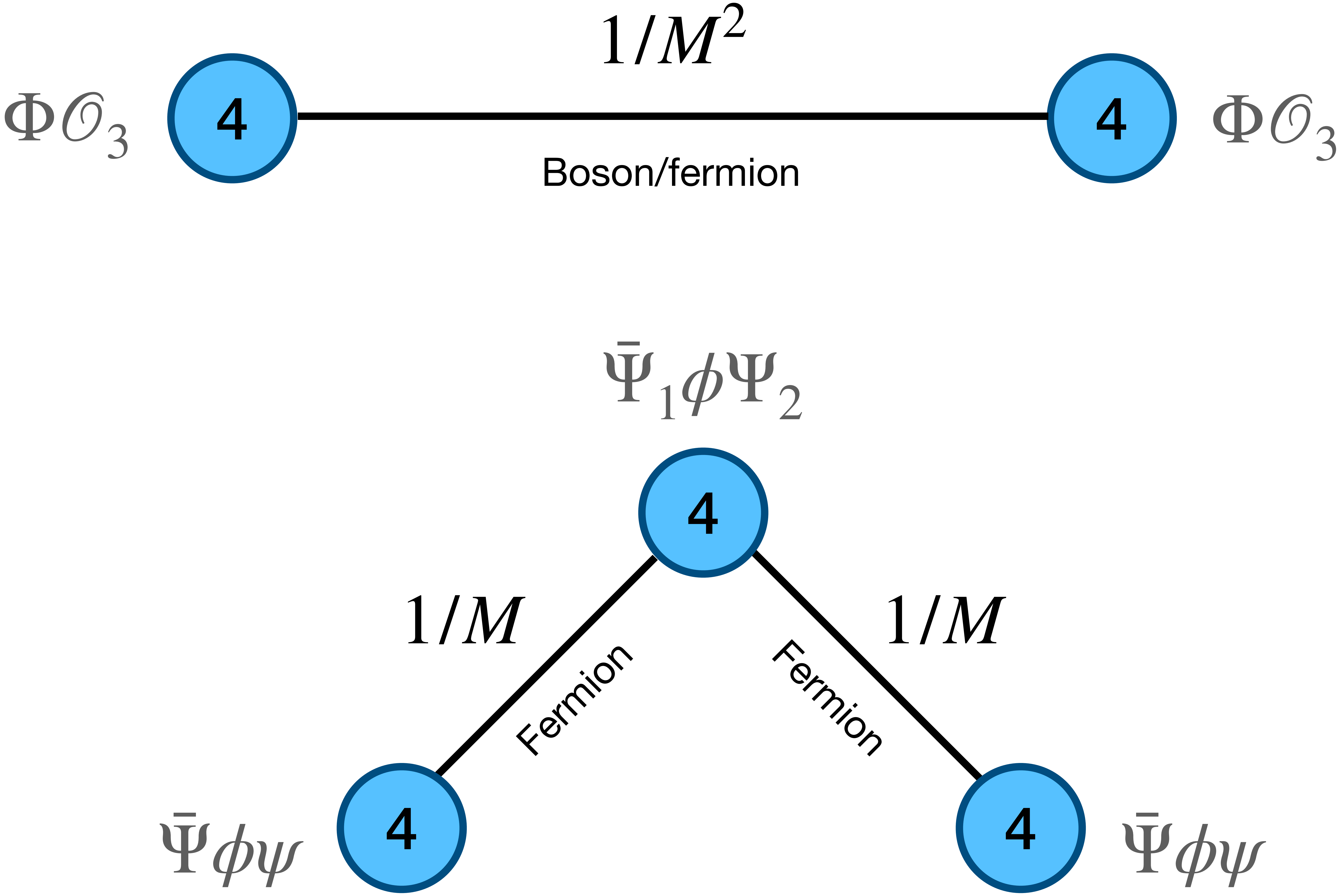
Tree-level

Extensions: dim-8, 1-loop, …

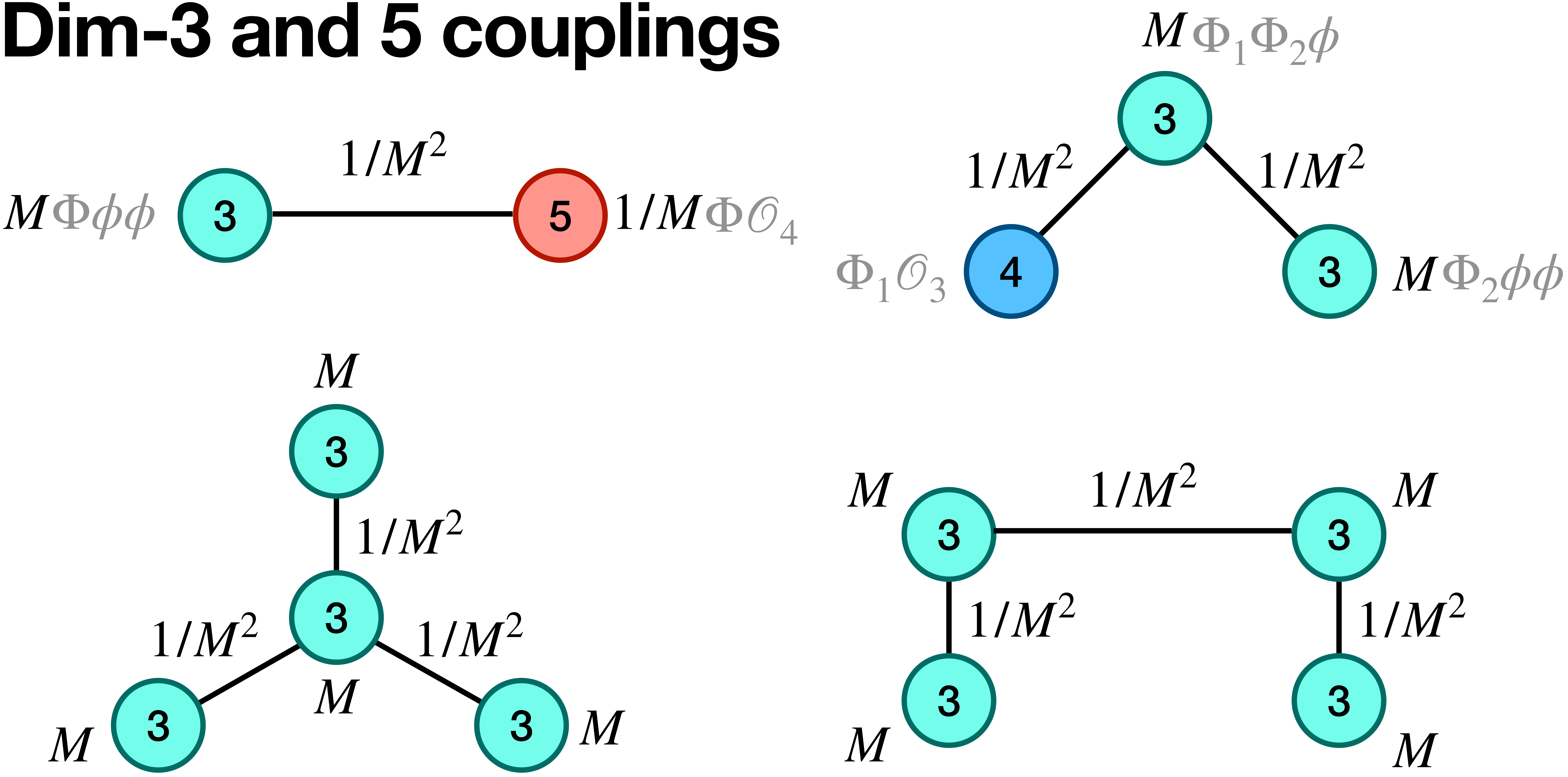
tree-level: MatchingTools

1-loop:
CoDEx
Matchete/SuperTracer
Matchingmakereft
STrEAM

Dim-4 couplings to the SM



Dim-3 and 5 couplings



1. Identify all heavy fields that generate dim-6 operators

= representations that couple **linearly** to the SM through **dim-3 or 4** operators

Scalars

Name	$\mathcal{S}$	$\mathcal{S}_1$	$\mathcal{S}_2$	φ	Ξ	Ξ_1	Θ_1	Θ_3
Irrep	$(1,1)_0$	$(1,1)_1$	$(1,1)_2$	$(1,2)_{\frac{1}{2}}$	$(1,3)_0$	$(1,3)_1$	$(1,4)_{\frac{1}{2}}$	$(1,4)_{\frac{3}{2}}$

Name	ω_1	ω_2	ω_4	Π_1	Π_7	ζ
Irrep	$(3,1)_{-\frac{1}{3}}$	$(3,1)_{\frac{2}{3}}$	$(3,1)_{-\frac{4}{3}}$	$(3,2)_{\frac{1}{6}}$	$(3,2)_{\frac{7}{6}}$	$(3,3)_{-\frac{1}{3}}$

Name	Ω_1	Ω_2	Ω_4	Υ	Φ
Irrep	$(6,1)_{\frac{1}{3}}$	$(6,1)_{-\frac{2}{3}}$	$(6,1)_{\frac{4}{3}}$	$(6,3)_{\frac{1}{3}}$	$(8,2)_{\frac{1}{2}}$

Fermions

Name	N	E	Δ_1	Δ_3	Σ	Σ_1
Irrep	$(1,1)_0$	$(1,1)_{-1}$	$(1,2)_{-\frac{1}{2}}$	$(1,2)_{-\frac{3}{2}}$	$(1,3)_0$	$(1,3)_{-1}$

Name	U	D	Q_1	Q_5	Q_7	T_1	T_2
Irrep	$(3,1)_{\frac{2}{3}}$	$(3,1)_{-\frac{1}{3}}$	$(3,2)_{\frac{1}{6}}$	$(3,2)_{-\frac{5}{6}}$	$(3,2)_{\frac{7}{6}}$	$(3,3)_{-\frac{1}{3}}$	$(3,3)_{\frac{2}{3}}$

Vectors

Name	$\mathcal{B}$	$\mathcal{B}_1$	$\mathcal{W}$	$\mathcal{W}_1$	$\mathcal{G}$	$\mathcal{G}_1$	$\mathcal{H}$	$\mathcal{L}_1$
Irrep	$(1,1)_0$	$(1,1)_1$	$(1,3)_0$	$(1,3)_1$	$(8,1)_0$	$(8,1)_1$	$(8,3)_0$	$(1,2)_{\frac{1}{2}}$

Name	$\mathcal{L}_3$	$\mathcal{U}_2$	$\mathcal{U}_5$	$\mathcal{Q}_1$	$\mathcal{Q}_5$	$\mathcal{X}$	$\mathcal{Y}_1$	$\mathcal{Y}_5$
Irrep	$(1,2)_{-\frac{3}{2}}$	$(3,1)_{\frac{2}{3}}$	$(3,1)_{\frac{5}{3}}$	$(3,2)_{\frac{1}{6}}$	$(3,2)_{-\frac{5}{6}}$	$(3,3)_{\frac{2}{3}}$	$(\bar{6},2)_{\frac{1}{6}}$	$(\bar{6},2)_{-\frac{5}{6}}$

2. Construct all their **interactions** that generate dim-6 ops

= those allowed by diagrammatic arguments

$$
\begin{aligned}
-\mathcal{L}_{\mathrm{S}}^{(\leq 4)} = {} & (\kappa_{\mathcal{S}})_r \mathcal{S}_r \phi^\dagger \phi + (\lambda_{\mathcal{S}})_{rs} \mathcal{S}_r \mathcal{S}_s \phi^\dagger \phi + (\kappa_{\mathcal{S}^3})_{rst} \mathcal{S}_r \mathcal{S}_s \mathcal{S}_t \\
& + \left\{ (y_{\mathcal{S}_1})_{rij} \mathcal{S}_{1r}^\dagger \bar{l}_{Li} i\sigma_2 l_{Lj}^c + \mathrm{h.c.} \right\} \\
& + \left\{ (y_{\mathcal{S}_2})_{rij} \mathcal{S}_{2k}^\dagger \bar{e}_{Ri} e_{Rj}^c + \mathrm{h.c.} \right\} \\
& + \left\{ (y_\varphi^e)_{rij} \varphi_r^\dagger \bar{e}_{Ri} l_{Lj} + (y_\varphi^d)_{rij} \varphi_r^\dagger \bar{d}_{Ri} q_{Lj} + (y_\varphi^u)_{rij} \varphi_r^\dagger i\sigma_2 \bar{q}_{Li}^T u_{Rj} \right. \\
& \quad \left. + (\lambda_\varphi)_r \left(\varphi_r^\dagger \phi \right) \left(\phi^\dagger \phi \right) + \mathrm{h.c.} \right\} \\
& + (\kappa_\Xi)_r \phi^\dagger \Xi_r^a \sigma^a \phi + (\lambda_\Xi)_{rs} \left(\Xi_r^a \Xi_s^a \right) \left(\phi^\dagger \phi \right) \\
& + \frac{1}{2} (\lambda_{\Xi_1})_{rs} \left(\Xi_{1r}^{a\dagger} \Xi_{1s}^a \right) \left(\phi^\dagger \phi \right) + \frac{1}{2} (\lambda'_{\Xi_1})_{rs} f_{abc} \left(\Xi_{1r}^{a\dagger} \Xi_{1s}^b \right) \left(\phi^\dagger \sigma^c \phi \right) \\
& + \left\{ (y_{\Xi_1})_{rij} \Xi_{1r}^{a\dagger} \bar{l}_{Li} \sigma^a i\sigma_2 l_{Lj}^c + (\kappa_{\Xi_1})_r \Xi_{1r}^{a\dagger} \left(\tilde{\phi}^\dagger \sigma^a \phi \right) + \mathrm{h.c.} \right\}
\end{aligned}
$$

$+\ldots$

3. Tree-level **matching** to dim-6 operators

Using MatchingTools + cross-checks by hand

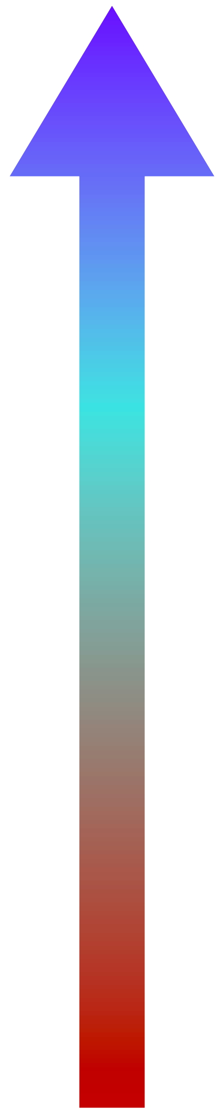
$$
\begin{aligned}
Z_\phi^2\, C_{\phi D} = &-\frac{2(\kappa_\Xi)_r(\kappa_\Xi)_r}{M_{\Xi_r}^4} + \frac{4(\kappa_{\Xi_1})_r^*(\kappa_{\Xi_1})_r}{M_{\Xi_{1r}}^4} - \frac{\mathrm{Re}\left((\hat{g}_{\mathcal{B}}^\phi)_r(\hat{g}_{\mathcal{B}}^\phi)_r\right)}{M_{\mathcal{B}_r}^2} - \frac{(\hat{g}_{\mathcal{B}}^\phi)_r^*(\hat{g}_{\mathcal{B}}^\phi)_r}{M_{\mathcal{B}_r}^2} \\
&+ \frac{(\hat{g}_{\mathcal{B}_1}^\phi)_r^*(\hat{g}_{\mathcal{B}_1}^\phi)_r}{M_{\mathcal{B}_{1r}}^2} - \frac{\mathrm{Re}\left((\hat{g}_{\mathcal{W}}^\phi)_r(\hat{g}_{\mathcal{W}}^\phi)_r\right)}{4M_{\mathcal{W}_r}^2} + \frac{(\hat{g}_{\mathcal{W}}^\phi)_r^*(\hat{g}_{\mathcal{W}}^\phi)_r}{4M_{\mathcal{W}_r}^2} - \frac{(\hat{g}_{\mathcal{W}_1}^\phi)_r^*(\hat{g}_{\mathcal{W}_1}^\phi)_r}{4M_{\mathcal{W}_{1r}}^2} \\
&+ \frac{g_1(g_{\mathcal{L}_1}^B)_{rs}(\gamma_{\mathcal{L}_1})_r^*(\gamma_{\mathcal{L}_1})_s}{M_{\mathcal{L}_{1r}}^2 M_{\mathcal{L}_{1s}}^2} - \frac{(h_{\mathcal{L}_1}^{(2)})_{rs}(\gamma_{\mathcal{L}_1})_r^*(\gamma_{\mathcal{L}_1})_s}{M_{\mathcal{L}_{1r}}^2 M_{\mathcal{L}_{1s}}^2} + \frac{2\,\mathrm{Re}\left((h_{\mathcal{L}_1}^{(3)})_{rs}(\gamma_{\mathcal{L}_1})_r^*(\gamma_{\mathcal{L}_1})_s^*\right)}{M_{\mathcal{L}_{1r}}^2 M_{\mathcal{L}_{1s}}^2} \\
&+ \frac{2\,\mathrm{Im}\left((\hat{g}_{\mathcal{W}}^\phi)_r\right)(\delta_{\mathcal{W}\Xi})_{rs}(\kappa_\Xi)_s}{M_{\mathcal{W}_r}^2 M_{\Xi_s}^2} + \frac{2(\delta_{\mathcal{W}\Xi})_{ts}(\delta_{\mathcal{W}\Xi})_{tr}(\kappa_\Xi)_r(\kappa_\Xi)_s}{M_{\Xi_r}^2 M_{\Xi_s}^2 M_{\mathcal{W}_t}^2} \\
&- \frac{2\,\mathrm{Im}\left((\hat{g}_{\mathcal{W}_1}^\phi)_r^*(\delta_{\mathcal{W}_1\Xi_1})_{rs}(\kappa_{\Xi_1})_s\right)}{M_{\mathcal{W}_{1r}}^2 M_{\Xi_{1s}}^2} - \frac{4(\delta_{\mathcal{W}_1\Xi_1})_{tr}^*(\delta_{\mathcal{W}_1\Xi_1})_{ts}(\kappa_{\Xi_1})_s(\kappa_{\Xi_1})_r^*}{M_{\Xi_{1r}}^2 M_{\Xi_{1s}}^2 M_{\mathcal{W}_{1t}}^2}
\end{aligned}
\quad + \ldots
$$

⟶ **Dim-6 tree-level matching done!**

Top-down

Fields	Operators
$\mathcal{S}$	$\mathcal{O}_{\phi 4}$, $\mathcal{O}_{\phi}$, $\mathcal{O}_{\phi\Box}$, $\mathcal{O}_{\phi B}$, $\mathcal{O}_{\phi \tilde{B}}$, $\mathcal{O}_{\phi W}$, $\mathcal{O}_{\phi \tilde{W}}$, $\mathcal{O}_{\phi G}$, $\mathcal{O}_{\phi \tilde{G}}$, $\mathcal{O}_{e\phi}$, $\mathcal{O}_{d\phi}$, $\mathcal{O}_{u\phi}$
$\mathcal{S}_1$	$\mathcal{O}_{ll}$
$\mathcal{S}_2$	$\mathcal{O}_{ee}$
φ	$\mathcal{O}_{le}$, $\mathcal{O}_{qu}^{(1)}$, $\mathcal{O}_{qu}^{(8)}$, $\mathcal{O}_{qd}^{(1)}$, $\mathcal{O}_{qd}^{(8)}$, $\mathcal{O}_{ledq}$, $\mathcal{O}_{quqd}^{(1)}$, $\mathcal{O}_{lequ}^{(1)}$, $\mathcal{O}_{\phi}$, $\mathcal{O}_{e\phi}$, $\mathcal{O}_{d\phi}$, $\mathcal{O}_{u\phi}$
Ξ	$\mathcal{O}_{\phi 4}$, $\mathcal{O}_{\phi}$, $\mathcal{O}_{\phi D}$, $\mathcal{O}_{\phi\Box}$, $\mathcal{O}_{\phi WB}$, $\mathcal{O}_{\phi W\tilde{B}}$, $\mathcal{O}_{e\phi}$, $\mathcal{O}_{d\phi}$, $\mathcal{O}_{u\phi}$
Ξ_1	$\mathcal{O}_{\phi 4}$, $\mathcal{O}_{5}$, $\mathcal{O}_{ll}$, $\mathcal{O}_{\phi}$, $\mathcal{O}_{\phi D}$, $\mathcal{O}_{\phi\Box}$, $\mathcal{O}_{e\phi}$, $\mathcal{O}_{d\phi}$, $\mathcal{O}_{u\phi}$
Θ_1	$\mathcal{O}_{\phi}$
Θ_3	$\mathcal{O}_{\phi}$
ω_1	$\mathcal{O}_{qq}^{(1)}$, $\mathcal{O}_{qq}^{(3)}$, $\mathcal{O}_{lq}^{(1)}$, $\mathcal{O}_{lq}^{(3)}$, $\mathcal{O}_{eu}$, $\mathcal{O}_{ud}^{(1)}$, $\mathcal{O}_{ud}^{(8)}$, $\mathcal{O}_{quqd}^{(1)}$, $\mathcal{O}_{quqd}^{(8)}$, $\mathcal{O}_{lequ}^{(1)}$, $\mathcal{O}_{lequ}^{(3)}$, $\mathcal{O}_{duq}$, $\mathcal{O}_{qqu}$, $\mathcal{O}_{qqq}$, $\mathcal{O}_{duu}$
ω_2	$\mathcal{O}_{dd}$
ω_4	$\mathcal{O}_{uu}$, $\mathcal{O}_{ed}$, $\mathcal{O}_{duu}$
Π_1	$\mathcal{O}_{ld}$
Π_7	$\mathcal{O}_{lu}$, $\mathcal{O}_{qe}$, $\mathcal{O}_{lequ}^{(1)}$, $\mathcal{O}_{lequ}^{(3)}$
ζ	$\mathcal{O}_{qq}^{(1)}$, $\mathcal{O}_{qq}^{(3)}$, $\mathcal{O}_{lq}^{(1)}$, $\mathcal{O}_{lq}^{(3)}$, $\mathcal{O}_{qqq}$
Ω_1	$\mathcal{O}_{qq}^{(1)}$, $\mathcal{O}_{qq}^{(3)}$, $\mathcal{O}_{ud}^{(1)}$, $\mathcal{O}_{ud}^{(8)}$, $\mathcal{O}_{quqd}^{(1)}$, $\mathcal{O}_{quqd}^{(8)}$
Ω_2	$\mathcal{O}_{dd}$
Ω_4	$\mathcal{O}_{uu}$
Υ	$\mathcal{O}_{qq}^{(1)}$, $\mathcal{O}_{qq}^{(3)}$
Φ	$\mathcal{O}_{qu}^{(1)}$, $\mathcal{O}_{qu}^{(8)}$, $\mathcal{O}_{qd}^{(1)}$, $\mathcal{O}_{qd}^{(8)}$, $\mathcal{O}_{quqd}^{(8)}$

Bottom-up



$$
\begin{aligned}
Z_\phi^2 C_{\phi D} = & -\frac{2(\kappa_{\Xi})_r(\kappa_{\Xi})_r}{M^4_{\Xi_r}} + \frac{4(\kappa_{\Xi_1})^*_r(\kappa_{\Xi_1})_r}{M^4_{\Xi_{1r}}} - \frac{\mathrm{Re}\left((\hat{g}^{\phi}_{\mathcal{B}})_r(\hat{g}^{\phi}_{\mathcal{B}})_r\right)}{M^2_{\mathcal{B}_r}} - \frac{(\hat{g}^{\phi}_{\mathcal{B}})^*_r(\hat{g}^{\phi}_{\mathcal{B}})_r}{M^2_{\mathcal{B}_r}} \\
& + \frac{(\hat{g}^{\phi}_{\mathcal{B}_1})^*_r(\hat{g}^{\phi}_{\mathcal{B}_1})_r}{M^2_{\mathcal{B}_{1r}}} - \frac{\mathrm{Re}\left((\hat{g}^{\phi}_{\mathcal{W}})_r(\hat{g}^{\phi}_{\mathcal{W}})_r\right)}{4M^2_{\mathcal{W}_r}} + \frac{(\hat{g}^{\phi}_{\mathcal{W}})^*_r(\hat{g}^{\phi}_{\mathcal{W}})_r}{4M^2_{\mathcal{W}_r}} - \frac{(\hat{g}^{\phi}_{\mathcal{W}_1})^*_r(\hat{g}^{\phi}_{\mathcal{W}_1})_r}{4M^2_{\mathcal{W}_{1r}}} \\
& + \frac{g_1(g^B_{\mathcal{L}_1})_{rs}(\gamma_{\mathcal{L}_1})^*_r(\gamma_{\mathcal{L}_1})_s}{M^2_{\mathcal{L}_{1r}}M^2_{\mathcal{L}_{1s}}} - \frac{(h^{(2)}_{\mathcal{L}_1})_{rs}(\gamma_{\mathcal{L}_1})^*_r(\gamma_{\mathcal{L}_1})_s}{M^2_{\mathcal{L}_{1r}}M^2_{\mathcal{L}_{1s}}} + \frac{2\,\mathrm{Re}\left((h^{(3)}_{\mathcal{L}_1})_{rs}(\gamma_{\mathcal{L}_1})^*_r(\gamma_{\mathcal{L}_1})^*_s\right)}{M^2_{\mathcal{L}_{1r}}M^2_{\mathcal{L}_{1s}}} \\
& + \frac{2\,\mathrm{Im}\left((\hat{g}^{\phi}_{\mathcal{W}})_r\right)(\delta_{\mathcal{W}\Xi})_{rs}(\kappa_{\Xi})_s}{M^2_{\mathcal{W}_r}M^2_{\Xi_s}} + \frac{2(\delta_{\mathcal{W}\Xi})_{ts}(\delta_{\mathcal{W}\Xi})_{tr}(\kappa_{\Xi})_r(\kappa_{\Xi})_s}{M^2_{\Xi_r}M^2_{\Xi_s}M^2_{\mathcal{W}_t}} \\
& - \frac{2\,\mathrm{Im}\left((\hat{g}^{\phi}_{\mathcal{W}_1})^*_r(\delta_{\mathcal{W}_1\Xi_1})_{rs}(\kappa_{\Xi_1})_s\right)}{M^2_{\mathcal{W}_{1r}}M^2_{\Xi_{1s}}} - \frac{4(\delta_{\mathcal{W}_1\Xi_1})^*_{tr}(\delta_{\mathcal{W}_1\Xi_1})_{ts}(\kappa_{\Xi_1})_s(\kappa_{\Xi_1})^*_r}{M^2_{\Xi_{1r}}M^2_{\Xi_{1s}}M^2_{\mathcal{W}_{1t}}}
\end{aligned}
$$

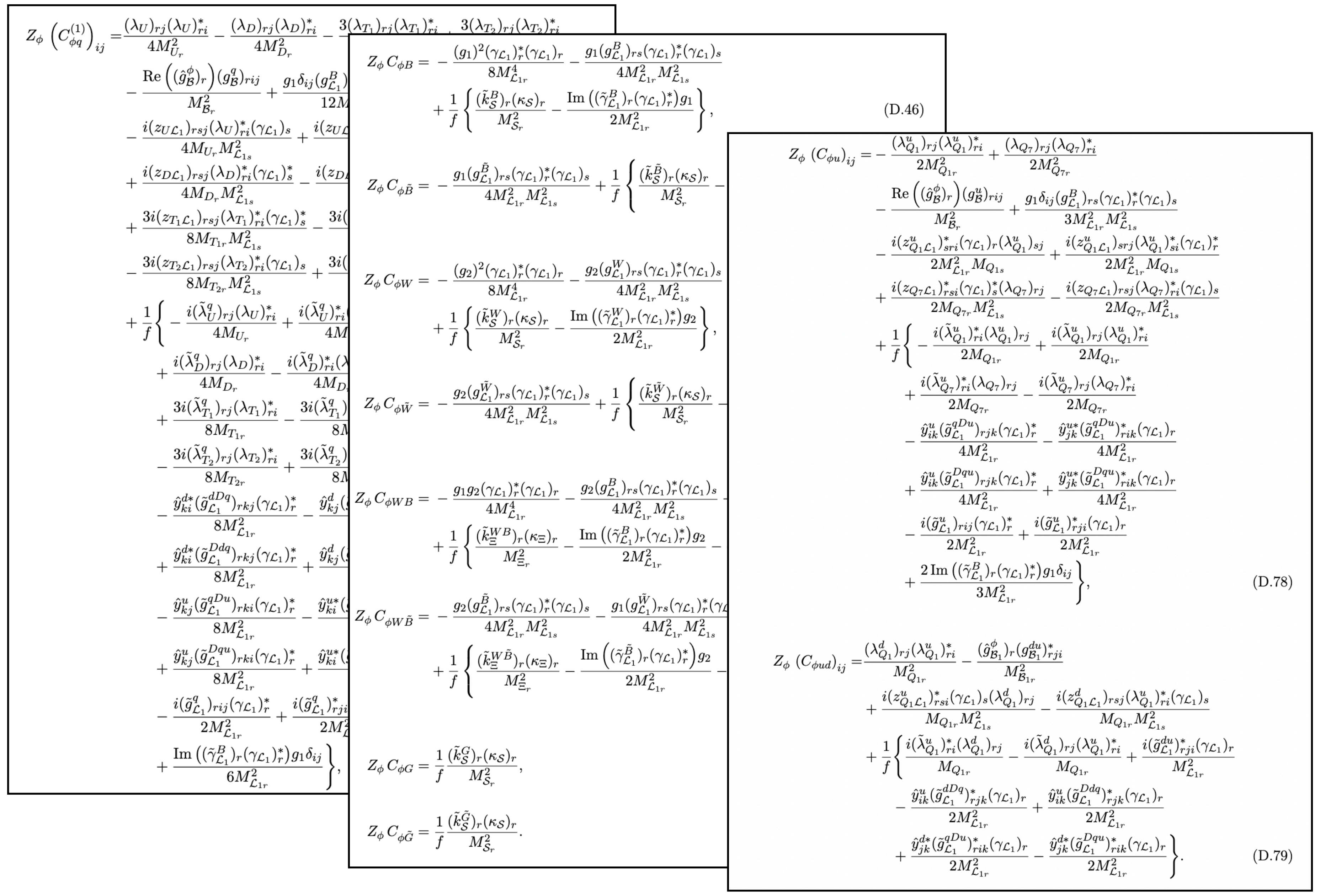
Example workflow

1. Parametrize experimental anomaly with Wilson coefficients
2. **Bottom-up dictionary**: identify (combinations of) heavy fields that generate the Wilson coefficients (in the correct combination).
3. **Top-down dictionary**: find other Wilson coefficients generated by the selected heavy fields.
4. Look for measurable effects of the other Wilson coefficients to

The need for automation

1711.10391:

- UV Lagrangian: 6 pages
- Wilson coefficients: 28 pages



$$Z_\phi\,(C_{\phi u})_{ij} = -\frac{(\lambda^u_{Q_1})_{rj}(\lambda^u_{Q_1})^*_{ri}}{2M^2_{Q_{1r}}} + \frac{(\lambda_{Q_7})_{rj}(\lambda_{Q_7})^*_{ri}}{2M^2_{Q_{7r}}} - \frac{\mathrm{Re}\left((\hat{g}^\phi_{\mathcal{B}})_r\right)(g^u_{\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}{3M^2_{\mathcal{L}_{1r}}M^2_{\mathcal{L}_{1s}}} - \frac{i(z^u_{Q_1\mathcal{L}_1})^*_{sri}(\gamma_{\mathcal{L}_1})_r(\lambda^u_{Q_1})_{sj}}{2M^2_{\mathcal{L}_{1r}}M_{Q_{1s}}} + \frac{i(z^u_{Q_1\mathcal{L}_1})_{srj}(\lambda^u_{Q_1})^*_{si}(\gamma_{\mathcal{L}_1})^*_r}{2M^2_{\mathcal{L}_{1r}}M_{Q_{1s}}} + \frac{i(z_{Q_7\mathcal{L}_1})^*_{rsi}(\gamma_{\mathcal{L}_1})^*_s(\lambda_{Q_7})_{rj}}{2M_{Q_{7r}}M^2_{\mathcal{L}_{1s}}} - \frac{i(z_{Q_7\mathcal{L}_1})_{rsj}(\lambda_{Q_7})^*_{ri}(\gamma_{\mathcal{L}_1})_s}{2M_{Q_{7r}}M^2_{\mathcal{L}_{1s}}} + \frac{1}{f}\Bigg\{ -\frac{i(\tilde{\lambda}^u_{Q_1})^*_{ri}(\lambda^u_{Q_1})_{rj}}{2M_{Q_{1r}}} + \frac{i(\tilde{\lambda}^u_{Q_1})_{rj}(\lambda^u_{Q_1})^*_{ri}}{2M_{Q_{1r}}} + \frac{i(\tilde{\lambda}^u_{Q_7})^*_{ri}(\lambda_{Q_7})_{rj}}{2M_{Q_{7r}}} - \frac{i(\tilde{\lambda}^u_{Q_7})_{rj}(\lambda_{Q_7})^*_{ri}}{2M_{Q_{7r}}} - \frac{\hat{y}^u_{ik}(\tilde{g}^{qDu}_{\mathcal{L}_1})_{rjk}(\gamma_{\mathcal{L}_1})^*_r}{4M^2_{\mathcal{L}_{1r}}} - \frac{\hat{y}^{u*}_{jk}(\tilde{g}^{qDu}_{\mathcal{L}_1})^*_{rik}(\gamma_{\mathcal{L}_1})_r}{4M^2_{\mathcal{L}_{1r}}} + \frac{\hat{y}^u_{ik}(\tilde{g}^{Dqu}_{\mathcal{L}_1})_{rjk}(\gamma_{\mathcal{L}_1})^*_r}{4M^2_{\mathcal{L}_{1r}}} + \frac{\hat{y}^{u*}_{jk}(\tilde{g}^{Dqu}_{\mathcal{L}_1})^*_{rik}(\gamma_{\mathcal{L}_1})_r}{4M^2_{\mathcal{L}_{1r}}} - \frac{i(\tilde{g}^u_{\mathcal{L}_1})_{rij}(\gamma_{\mathcal{L}_1})^*_r}{2M^2_{\mathcal{L}_{1r}}} + \frac{i(\tilde{g}^u_{\mathcal{L}_1})^*_{rji}(\gamma_{\mathcal{L}_1})_r}{2M^2_{\mathcal{L}_{1r}}} + \frac{2\,\mathrm{Im}\left((\tilde{\gamma}^B_{\mathcal{L}_1})_r(\gamma_{\mathcal{L}_1})^*_r\right)g_1\delta_{ij}}{3M^2_{\mathcal{L}_{1r}}}\Bigg\}, \quad \text{(D.78)}$$

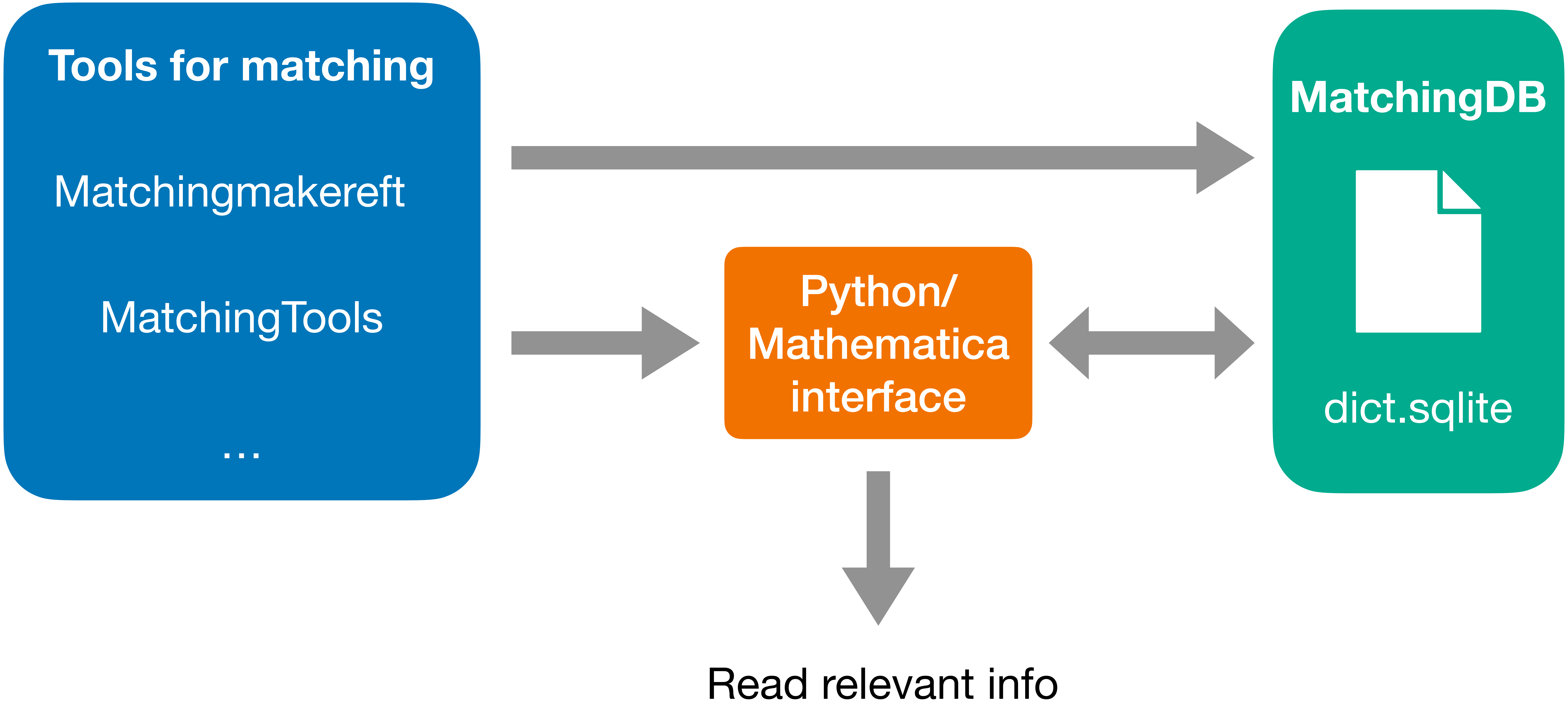
$$Z_\phi\,(C_{\phi ud})_{ij} = \frac{(\lambda^d_{Q_1})_{rj}(\lambda^u_{Q_1})^*_{ri}}{M^2_{Q_{1r}}} - \frac{(\hat{g}^\phi_{\mathcal{B}_1})_r(g^{du}_{\mathcal{B}_1})^*_{rji}}{M^2_{\mathcal{B}_{1r}}} + \frac{i(z^u_{Q_1\mathcal{L}_1})^*_{rsi}(\gamma_{\mathcal{L}_1})_s(\lambda^d_{Q_1})_{rj}}{M_{Q_{1r}}M^2_{\mathcal{L}_{1s}}} - \frac{i(z^d_{Q_1\mathcal{L}_1})_{rsj}(\lambda^u_{Q_1})^*_{ri}(\gamma_{\mathcal{L}_1})_s}{M_{Q_{1r}}M^2_{\mathcal{L}_{1s}}} + \frac{1}{f}\Bigg\{\frac{i(\tilde{\lambda}^u_{Q_1})^*_{ri}(\lambda^d_{Q_1})_{rj}}{M_{Q_{1r}}} - \frac{i(\tilde{\lambda}^d_{Q_1})_{rj}(\lambda^u_{Q_1})^*_{ri}}{M_{Q_{1r}}} + \frac{i(\tilde{g}^{du}_{\mathcal{L}_1})^*_{rji}(\gamma_{\mathcal{L}_1})_r}{M^2_{\mathcal{L}_{1r}}} - \frac{\hat{y}^u_{ik}(\tilde{g}^{dDq}_{\mathcal{L}_1})^*_{rjk}(\gamma_{\mathcal{L}_1})_r}{2M^2_{\mathcal{L}_{1r}}} + \frac{\hat{y}^u_{ik}(\tilde{g}^{Ddq}_{\mathcal{L}_1})^*_{rjk}(\gamma_{\mathcal{L}_1})_r}{2M^2_{\mathcal{L}_{1r}}} + \frac{\hat{y}^{d*}_{jk}(\tilde{g}^{qDu}_{\mathcal{L}_1})^*_{rik}(\gamma_{\mathcal{L}_1})_r}{2M^2_{\mathcal{L}_{1r}}} - \frac{\hat{y}^{d*}_{jk}(\tilde{g}^{Dqu}_{\mathcal{L}_1})^*_{rik}(\gamma_{\mathcal{L}_1})_r}{2M^2_{\mathcal{L}_{1r}}}\Bigg\}. \quad \text{(D.79)}$$

Dim-8, 1-loop → x10? More?

A format for matching dictionaries

Requirements

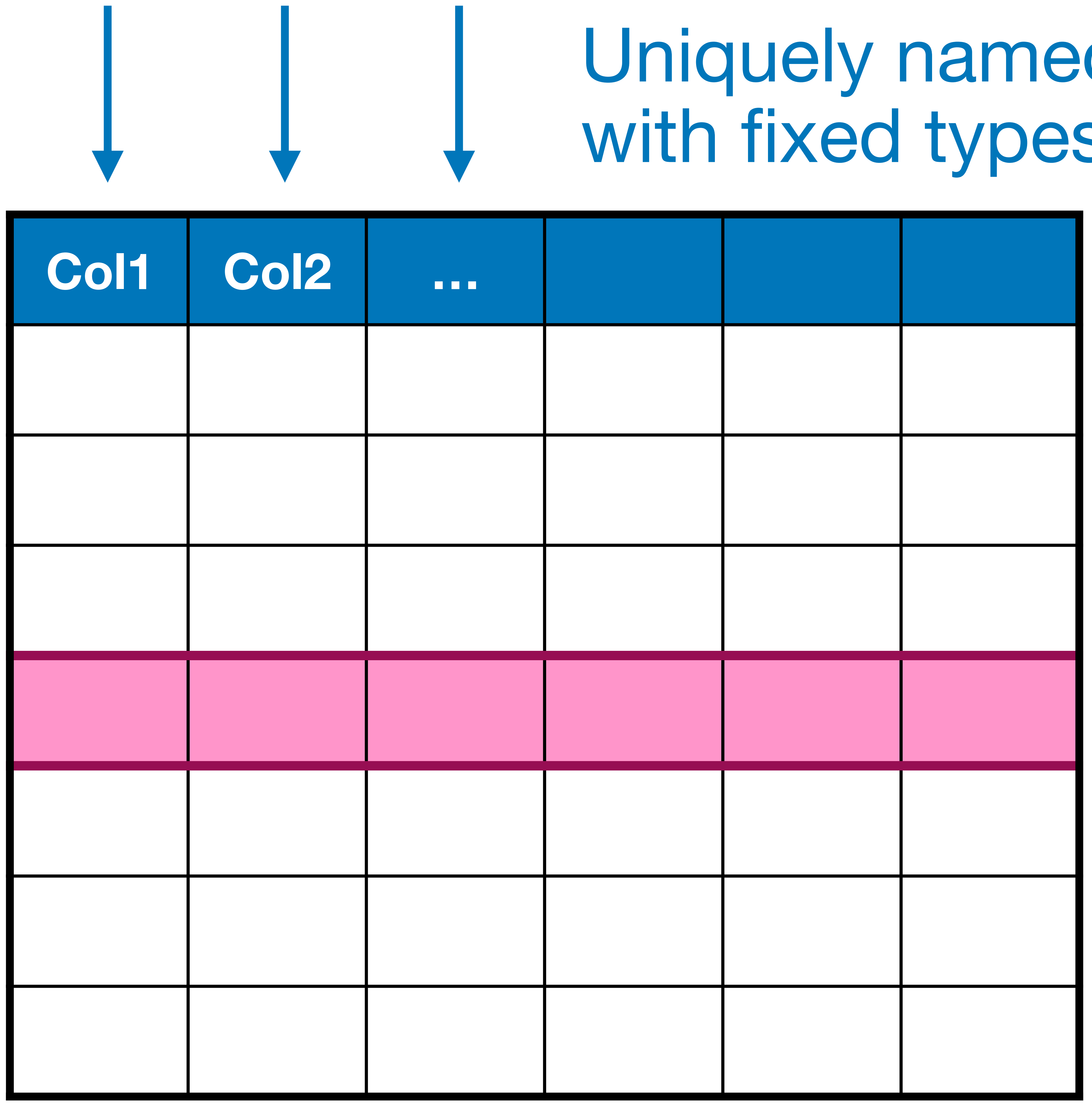
- Fully general for tree-level and **1-loop** matching
- Contain information about fields and couplings in the UV theory
- Easy access to:
  - **Bottom-up** info: UV fields and couplings that generate a WC
  - **Top-down** info: WCs generated by a given set of UV fields and couplings
- Input/output values of the WCs in several **formats**: numerical values (WCxf), LaTeX, Mathematica, SymPy, …


Tools for matching
Matchingmakereft
MatchingTools
...
MatchingDB
dict.sqlite
Python/
Mathematica
interface
Read relevant info

Relational databases

Multiple tables:

Uniquely named columns with fixed types

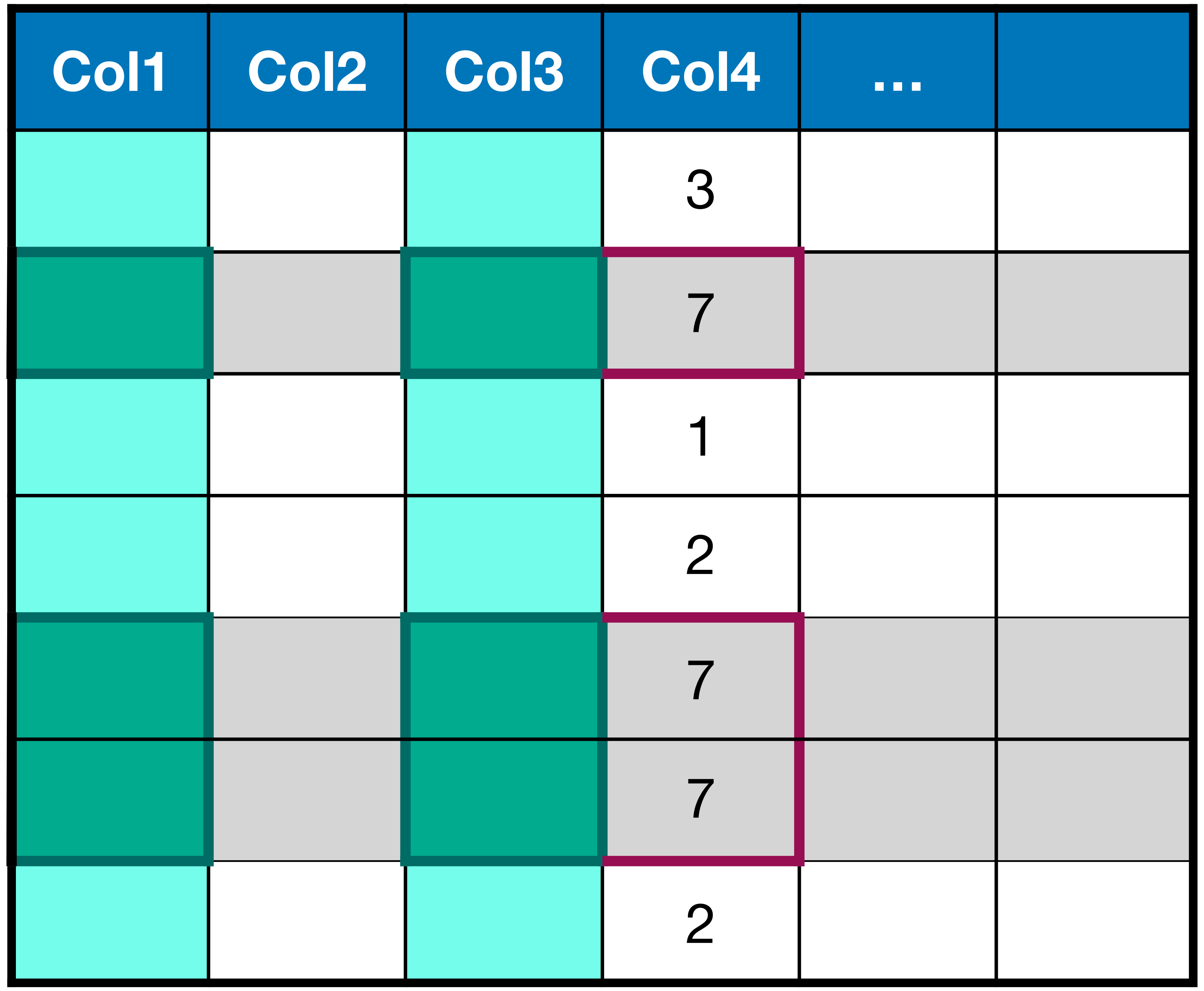


Rows are tuples with types:

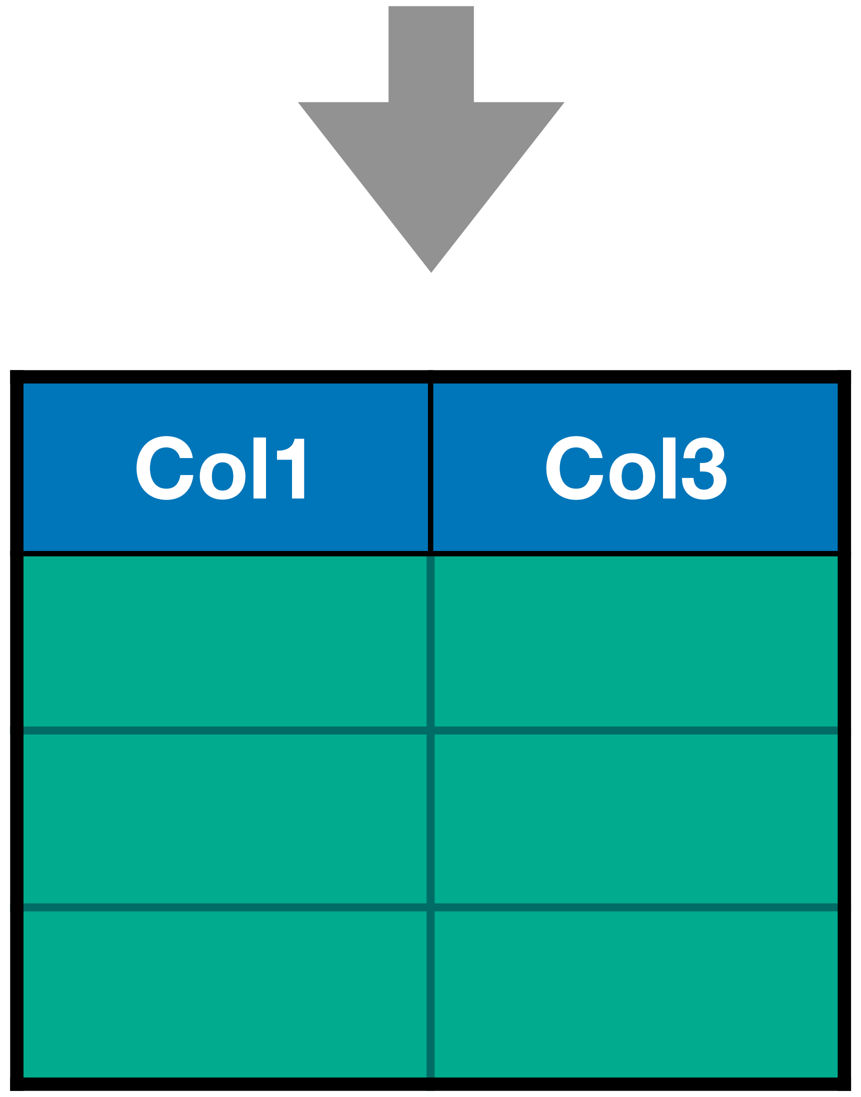
(type(Col1), type(Col2), …)

SQL

Table1:




```
SELECT Col1, Col3
FROM Table1
WHERE Col4=7;
```



SQLite

- Whole database in a **single file**: easy to share.
- No client-server system, just: `$ sqlite3 my_db.sqlite` and run queries.
- Bindings in the Python and Mathematica standard libraries:

```python
import sqlite3

connection = sqlite3.connect("my_db.sqlite")
cursor = connection.cursor()

cursor.execute("CREATE TABLE ...")
cursor.execute("SELECT ... FROM ...")
```

```
Needs["DatabaseLink`"]

db = OpenSQLConnection[JDBC["SQLite", "my_db.sqlite"]]

SQLExecute[db, "CREATE TABLE ..."]
SQLExecute[db, "SELECT ... FROM ..."]
```

How to store Wilson coefficients?

$$Z_\phi^2 C_{\phi D} = -\frac{2(\kappa_\Xi)_r(\kappa_\Xi)_r}{M_{\Xi_r}^4} + \frac{4(\kappa_{\Xi_1})_r^*(\kappa_{\Xi_1})_r}{M_{\Xi_{1r}}^4} - \frac{\mathrm{Re}\left((\hat{g}_\mathcal{B}^\phi)_r(\hat{g}_\mathcal{B}^\phi)_r\right)}{M_{\mathcal{B}_r}^2} - \frac{(\hat{g}_\mathcal{B}^\phi)_r^*(\hat{g}_\mathcal{B}^\phi)_r}{M_{\mathcal{B}_r}^2}$$
$$+ \frac{(\hat{g}_{\mathcal{B}_1}^\phi)_r^*(\hat{g}_{\mathcal{B}_1}^\phi)_r}{M_{\mathcal{B}_{1r}}^2} - \frac{\mathrm{Re}\left((\hat{g}_\mathcal{W}^\phi)_r(\hat{g}_\mathcal{W}^\phi)_r\right)}{4M_{\mathcal{W}_r}^2} + \frac{(\hat{g}_\mathcal{W}^\phi)_r^*(\hat{g}_\mathcal{W}^\phi)_r}{4M_{\mathcal{W}_r}^2} - \frac{(\hat{g}_{\mathcal{W}_1}^\phi)_r^*(\hat{g}_{\mathcal{W}_1}^\phi)_r}{4M_{\mathcal{W}_{1r}}^2}$$
$$+ \frac{g_1(g_{\mathcal{L}_1}^B)_{rs}(\gamma_{\mathcal{L}_1})_r^*(\gamma_{\mathcal{L}_1})_s}{M_{\mathcal{L}_{1r}}^2 M_{\mathcal{L}_{1s}}^2} - \frac{(h_{\mathcal{L}_1}^{(2)})_{rs}(\gamma_{\mathcal{L}_1})_r^*(\gamma_{\mathcal{L}_1})_s}{M_{\mathcal{L}_{1r}}^2 M_{\mathcal{L}_{1s}}^2} + \frac{2\,\mathrm{Re}\left((h_{\mathcal{L}_1}^{(3)})_{rs}(\gamma_{\mathcal{L}_1})_r^*(\gamma_{\mathcal{L}_1})_s^*\right)}{M_{\mathcal{L}_{1r}}^2 M_{\mathcal{L}_{1s}}^2}$$
$$+ \frac{2\,\mathrm{Im}\left((\hat{g}_\mathcal{W}^\phi)_r\right)(\delta_{\mathcal{W}\Xi})_{rs}(\kappa_\Xi)_s}{M_{\mathcal{W}_r}^2 M_{\Xi_s}^2} + \frac{2(\delta_{\mathcal{W}\Xi})_{ts}(\delta_{\mathcal{W}\Xi})_{tr}(\kappa_\Xi)_r(\kappa_\Xi)_s}{M_{\Xi_r}^2 M_{\Xi_s}^2 M_{\mathcal{W}_t}^2}$$
$$- \frac{2\,\mathrm{Im}\left((\hat{g}_{\mathcal{W}_1}^\phi)_r^*(\delta_{\mathcal{W}_1\Xi_1})_{rs}(\kappa_{\Xi_1})_s\right)}{M_{\mathcal{W}_{1r}}^2 M_{\Xi_{1s}}^2} - \frac{4(\delta_{\mathcal{W}_1\Xi_1})_{tr}^*(\delta_{\mathcal{W}_1\Xi_1})_{ts}(\kappa_{\Xi_1})_s(\kappa_{\Xi_1})_r^*}{M_{\Xi_{1r}}^2 M_{\Xi_{1s}}^2 M_{\mathcal{W}_{1t}}^2}$$

From 1711.10391

$$(\alpha_{He})_{ij} = \frac{1}{16\pi^2}\frac{1}{M_{E\,\alpha}^2}\left[g_1^2\left(\frac{2g_1^2}{15} - \frac{13\tilde{\lambda}_{k\alpha}\tilde{\lambda}_{k\alpha}^*}{36}\right)\delta_{ij} + \frac{1}{24}(Y_e^\dagger)_{ik}\tilde{\lambda}_{k\alpha}\tilde{\lambda}_{l\alpha}^*(Y_e)_{lj}\right.$$
$$\left. + \left(-\frac{g_1^2}{6}\tilde{\lambda}_{k\alpha}\tilde{\lambda}_{k\alpha}^*\delta_{ij} + \frac{1}{4}(Y_e^\dagger)_{ik}\tilde{\lambda}_{k\alpha}\tilde{\lambda}_{l\alpha}^*(Y_e)_{lj}\right)\log(\mu^2/M_{E\,\alpha}^2)\right]$$

From A. Carmona, A. Lazopoulos, P. Olgoso, S. Santiago, 2112.10787, (Matchmakereft)

Terms

General form

$$\texttt{term} := \frac{N}{D}\pi^{n_\pi} \left(\prod_i \texttt{coupling\_factor}_i\right) \left(\prod_j \texttt{mass\_factor}_j\right) \texttt{mass\_log}$$

$$\texttt{coupling\_factor} := \left(g_{i_1 i_2 \ldots}\right)^n \text{ or } \left(g^*_{i_1 i_2 \ldots}\right)^n$$

$$\texttt{mass\_factor} := M^n_{F,i} \text{ or } \left(M^2_{F_1,i_1} - M^2_{F_2,i_2}\right)^n$$

$$\texttt{mass\_log} := 1 \text{ or } \log\frac{M^n_{F,i}}{\mu^n} \text{ or } \log\frac{M^n_{F_1,i_1}}{M^n_{F_2,i_2}}$$

Terms

Encoding

$\texttt{coupling\_factor} := \left(g_{i_1 i_2 \ldots}\right)^n$ or $\left(g^*_{i_1 i_2 \ldots}\right)^n$

name, exponent, conjugate, indices

[str, int, bool, str]

$\texttt{mass\_factor} := M^n_{F,i}$ or $\left(M^2_{F_1,i_1} - M^2_{F_2,i_2}\right)^n$

field, index, exponent

[[str, str], int]

field, index, field, index, exponent

[[str, str], [str, str], int]

$\texttt{mass\_log} := 1$ or $\log \dfrac{M^n_{F,i}}{\mu^n}$ or $\log \dfrac{M^n_{F_1,i_1}}{M^n_{F_2,i_2}}$

[]

field, index, exponent

[[str, str], int]

field, index, field, index, exponent

[[str, str], [str, str], int]

The MatchingDB format

Specification (see MatchingDB.md)

Table **Terms**

coefficient	fields	numerical_fraction	pi_exponent	coupling_factors	mass_factors	mass_log
“phie”	“["E"]”	“[1, 2]”	0	“[[“g",1,True,"ik"],…]”	“[…]”	“[]”
“phie”	“["S","E"]”	“[-3, 16]”	-2	…	…	…
“uG”	…	…	…	…	…	…
…	…	…	…	…	…	…

Strings containing JSON arrays (of arrays)

```
In [1]: import json

In [2]: json.loads('["g", 1, true, "ik"]')
Out[2]: ['g', 1, True, 'ik']

In [3]: json.dumps(["g", 1, True, "ik"])
Out[3]: '["g", 1, true, "ik"]'
```

```
In[1]:= ImportString["[\"g\",1,true,\"ik\"]", "JSON"]

Out[1]= {g, 1, True, ik}

In[2]:= StringDelete[WhitespaceCharacter][ExportString[{"g", 1, True, "ik"}, "JSON"]]

Out[2]= ["g",1,true,"ik"]
```

The MatchingDB format

Specification (see MatchingDB.md)

Table **Fields**

name	real	lorentz_rep	su3_rep	su2_rep	u1_rep	latex
"S"	1	"scalar"	"1"	"1"	"0"	"\mathcal{S}"
"Xi1"	0	"scalar"	"1"	"3"	"1"	"\Xi_1"
"Delta1"	0	"vector"	"1"	"2"	"-1/2"	"\Delta_1"
…	…	…	…	…	…	…

The MatchingDB format

Specification (see MatchingDB.md)

Table **Couplings**

name	fields	real	latex	latex_interaction	latex_indices
“lambda_tilde”	“["E"]”	0	“\tilde{\lambda}”	“\bar{E}_a \phi^\dagger l_i”	“ai”
“kappa_S”	“["S"]”	1	“\kappa_{\mathcal{S}}”	“\mathcal{S}_a \phi^\dagger \phi”	“a”
…		…	…	…	…

Extracting info from MatchingDB

Tables	Information
Terms	Get all combinations of heavy **fields that contribute to a given Wilson coefficient**
Terms	Get all combinations of couplings that contribute to a given Wilson coefficient
Terms + Fields + Couplings	Write the expression for a Wilson coefficient in **LaTeX**, **Mathematica**, …
Terms (+ user-provided numerical values for couplings and masses)	Write the numerical values of Wilson coefficients (**WCxf**)
Fields	Get information (spin, representation, …) about a given field
Couplings	Get information (heavy fields involved, real/complex, …) about a given coupling
Fields + Couplings	Write the **relevant sector of a UV Lagrangian** for a given set of fields

Python interface

Writing to the database

$$C_{\phi e} = \frac{1}{64\pi^2 M_E^2} (Y_e)^*_{ik}\ \tilde{\lambda}_{ka}\ \tilde{\lambda}^*_{la}\ (Y_e)_{lj}\ \log \frac{\mu^2}{M^2_{E,a}} + \dots$$

```
mdb.Term(
    coefficient="phie",
    fields=["E"],
    numerical_fraction=(1, 64),
    pi_exponent=-2,
    coupling_factors=[
        ("Y_e", 1, True, "ki"),
        ("lambda_tilde", 1, False, "ka"),
        ("lambda_tilde", 1, True, "la"),
        ("Y_e", 1, False, "lj"),
    ],
    mass_factors=[(("E", "a"), -2)],
    mass_log=(("E", "a"), -2),
)
```

Python interface

Writing to the database

$$\mathscr{L}_{\mathrm{UV}} = \tilde{\lambda}_{ia}\bar{\ell}_{Li}\phi E_{Ra} + \ldots$$

```python
mdb.Coupling(
    name="lambda_tilde",
    fields=["E"],
    real=False,
    latex=r"\tilde{\lambda}",
    latex_interaction=r"\bar{\ell}_{Li} \phi E_{Ra}",
    latex_indices="ia",
)
```

Python interface

Writing to the database

$$E \sim \text{fermion} \times (1,1)_{-1}$$

```
mdb.Field(
    name="E",
    real=False,
    lorentz_rep="dirac",
    su3_rep="1",
    su2_rep="2",
    u1_rep="1",
    latex="E",
)
```

Python interface

Writing to the database

```python
import matching_db as mdb


example_dict = mdb.MatchingDB.from_file("my_dict.sqlite")
example_dict.create_tables()

example_dict.insert_all([Term(...), Term(...), ..., Field(...), ..., Coupling(...)])
```

Python interface

Reading from the database

[1]:
```python
from IPython.display import display, Math, Latex
import matching_db as mdb
import numpy as np
from pprint import pprint

db = mdb.MatchingDB.from_file("example.sqlite")
```

[2]:
```python
pprint(db.select_eft(coefficients=["phie"]))
```

```
{'phie': [(['E'], ['g1', 'kdelta']),
          (['E'], ['g1', 'lambda_tilde', 'lambda_tilde', 'kdelta']),
          (['E'], ['Y_e', 'lambda_tilde', 'lambda_tilde', 'Y_e']),
          (['E'], ['g1', 'lambda_tilde', 'lambda_tilde', 'kdelta']),
          (['E'], ['Y_e', 'lambda_tilde', 'lambda_tilde', 'Y_e'])]}
```

Python interface

Reading from the database

```
[3]: pprint(db.select_eft(fields=["E"]))

{'phie': [(['E'], ['g1', 'kdelta']),
          (['E'], ['g1', 'lambda_tilde', 'lambda_tilde', 'kdelta']),
          (['E'], ['Y_e', 'lambda_tilde', 'lambda_tilde', 'Y_e', 'kdelta']),
          (['E'], ['g1', 'lambda_tilde', 'lambda_tilde', 'kdelta']),
          (['E'], ['Y_e', 'lambda_tilde', 'lambda_tilde', 'Y_e'])],
 'phil3': [(['E'], ['lambda_tilde', 'lambda_tilde']),
           (['E'], ['g2', 'lambda_tilde', 'lambda_tilde', 'kdelta']),
           (['E'], ['g2', 'lambda_tilde', 'lambda_tilde', 'kdelta'])]}
```

Python interface

Reading from the database

```
[2]: display(Math(db.select_eft(coefficients=["phie"], output="latex")["phie"]))
```

$$+\frac{2(g_1)^4(\delta)_{ij}}{240\pi^2 M_{E,a}^2} - \frac{13(g_1)^2\left(\tilde{\lambda}\right)_{ka}\left(\tilde{\lambda}\right)^*_{ka}(\delta)_{ij}}{576\pi^2 M_{E,a}^2} + \frac{(Y_e)^*_{ki}\left(\tilde{\lambda}\right)_{ka}\left(\tilde{\lambda}\right)^*_{la}(Y_e)_{lj}}{384\pi^2 M_{E,a}^2}$$

$$-\frac{(g_1)^2\left(\tilde{\lambda}\right)_{ka}\left(\tilde{\lambda}\right)^*_{ka}(\delta)_{ij}}{96\pi^2 M_{E,a}^2}\log\left(\frac{M_{E,a}}{\mu}\right)^{-2} + \frac{(Y_e)^*_{ki}\left(\tilde{\lambda}\right)_{ka}\left(\tilde{\lambda}\right)^*_{la}(Y_e)_{lj}}{64\pi^2 M_{E,a}^2}\log\left(\frac{M_{E,a}}{\mu}\right)^{-2}$$

Python interface

Reading from the database

```
[3]: parameter_values = {
         "g1": 0.5,
         "g2": 0.7,
         "lambda_tilde": np.random.uniform(-1, 1, size=(3, 2)),
         "kdelta": np.eye(3),
         "mu": 0.5,
         "M_E": np.array([0.8, 1]),
         "Y_e": np.random.uniform(-1, 1, size=(3, 3)),
     }

     db.select_eft(fields=["E"], output="wcxf", parameter_values=parameter_values)
```

```
[3]: {'phie_00': -0.0012965327691838164,
      'phie_11': -0.0011858530754861464,
      'phie_22': -0.0006700521416365115,
      'phie_01': -0.0009215432024849207,
      'phie_02': 0.0005869762087864108,
      'phie_10': -0.0009215432024849205,
      'phie_12': 0.0005527777258284804,
      'phie_20': 0.0005869762087864111,
      'phie_21': 0.0005527777258284806,
      'phil3_00': -0.11443354938856719,
      'phil3_01': 0.17274347325716877,
      'phil3_02': 0.01976458787400561,
      'phil3_10': 0.17274347325716877,
      'phil3_11': -0.24678924349937148,
      'phil3_12': -0.028630346378964958,
      'phil3_20': 0.01976458787400561,
      'phil3_21': -0.028630346378964958,
      'phil3_22': 0.0009157914183244992}
```

Python interface

Reading from the database

```
[4]: db.select_uv(fields=["E"])

[4]: [('lambda_tilde', ['E'])]

[6]: db.get_field("E")

[6]: Field(name='E', real=False, lorentz_rep='dirac', su3_rep='1', su2_rep='2', u1
     _rep='1', latex='E')

[7]: db.get_coupling("lambda_tilde")

[7]: Coupling(name='lambda_tilde', fields=['E'], real=False, latex='\\tilde{\\lamb
     da}', latex_interaction='\\bar{\\ell}_i \\phi E_{Ra}', latex_indices='ia')
```

Ongoing/future work

- Polish/generalize MatchingDB specification. Comments are welcome!
- Add missing features to Python interface (UV Lagrangian, Mathematica output, SymPy?, …)
- Build **Mathematica interface**.
- Provide the output of **MatchingTools** and **Matchmakereft** in the MatchingDB format, using the Python/Mathematica interface.
- Make **all dictionaries available** at the GitHub repo.

Conclusions

- Matching dictionaries are an efficient and practically useful tool to deal with matching calculations in EFT.
- Huge size $\Longrightarrow$ automatic tools are necessary.
- Common format can be useful to present matching results from any tool.
- MatchingDB format: fully general for tree-level and 1-loop matching, contain info about the UV theory too.