What's New in Matchete?

Matthias König
JGU Mainz
on behalf of the
Matchete collaboration

"SMEFT-Tools 2025"
MITP Mainz
Jan 28, 2025

European Research Council
Established by the European Commission

Cluster of Excellence
PRISMA+
Precision Physics,
Fundamental Interactions
and Structure of Matter

Effective Field Theories established tool of the trade for understanding the **low-energy phenomenology** of a given **NP model**:

- For most applications, the **target basis is known** (SMEFT/HEFT/LEFT)

Effective Field Theories established tool of the trade for understanding the **low-energy phenomenology** of a given **NP model**:

- For most applications, the **target basis is known** (SMEFT/HEFT/LEFT)
- Low-energy phenomenology has been **worked out** in terms of their parameters

Effective Field Theories established tool of the trade for understanding the **low-energy phenomenology** of a given **NP model**:

- For most applications, the **target basis is known** (SMEFT/HEFT/LEFT)
- Low-energy phenomenology has been **worked out** in terms of their parameters
- Large **logarithmic corrections** well under control by their **renormalization group evolution**

Effective Field Theories established tool of the trade for understanding the **low-energy phenomenology** of a given **NP model**:

- For most applications, the **target basis is known** (SMEFT/HEFT/LEFT)
- Low-energy phenomenology has been **worked out** in terms of their parameters
- Large **logarithmic corrections** well under control by their **renormalization group evolution**
- Community has been hard at work developing some really amazing **tools** to **streamline** the workflow.

Effective Field Theories established tool of the trade for understanding the **low-energy phenomenology** of a given **NP model**:

- For most applications, the **target basis is known** (SMEFT/HEFT/LEFT)
- Low-energy phenomenology has been **worked out** in terms of their parameters
- Large **logarithmic corrections** well under control by their **renormalization group evolution**
- Community has been hard at work developing some really amazing **tools** to **streamline** the workflow.

Matching of BSM $\to$ EFT is a **laborious** task:

Effective Field Theories established tool of the trade for understanding the **low-energy phenomenology** of a given **NP model**:

- For most applications, the **target basis is known** (SMEFT/HEFT/LEFT)
- Low-energy phenomenology has been **worked out** in terms of their parameters
- Large **logarithmic corrections** well under control by their **renormalization group evolution**
- Community has been hard at work developing some really amazing **tools** to **streamline** the workflow.

Matching of BSM $\to$ EFT is a **laborious** task:

- Target **basis** is **large**, many things to compute

Effective Field Theories established tool of the trade for understanding the **low-energy phenomenology** of a given **NP model**:

- For most applications, the **target basis is known** (SMEFT/HEFT/LEFT)
- Low-energy phenomenology has been **worked out** in terms of their parameters
- Large **logarithmic corrections** well under control by their **renormalization group evolution**
- Community has been hard at work developing some really amazing **tools** to **streamline** the workflow.

Matching of BSM $\to$ EFT is a **laborious** task:

- Target **basis** is **large**, many things to compute
- Interesting pheno often **beyond tree-level**, especially in flavor

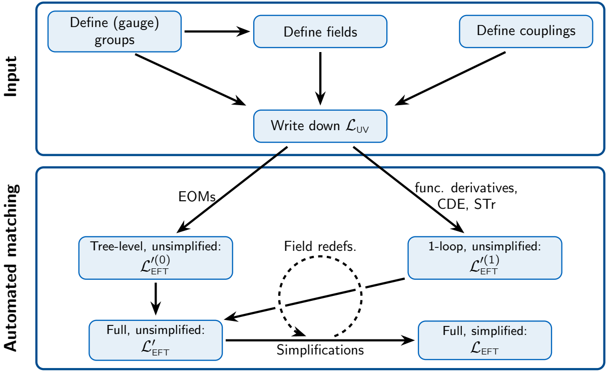
Effective Field Theories established tool of the trade for understanding the **low-energy phenomenology** of a given **NP model**:

- For most applications, the **target basis is known** (SMEFT/HEFT/LEFT)
- Low-energy phenomenology has been **worked out** in terms of their parameters
- Large **logarithmic corrections** well under control by their **renormalization group evolution**
- Community has been hard at work developing some really amazing **tools** to **streamline** the workflow.

Matching of BSM $\to$ EFT is a **laborious** task:

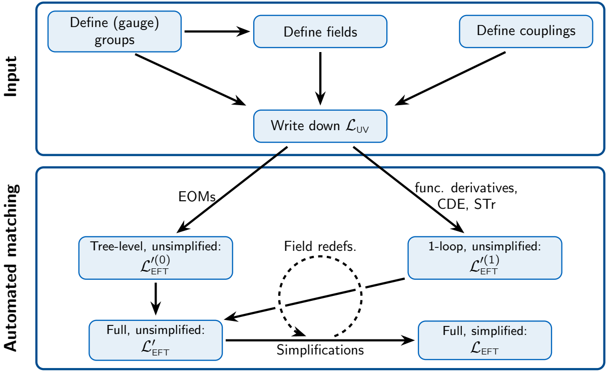
- Target **basis** is **large**, many things to compute
- Interesting pheno often **beyond tree-level**, especially in flavor
- Not a terribly difficult, but quite frankly **boring** to do by hand

Matchete is a **Mathematica package** to automate **EFT matching**.



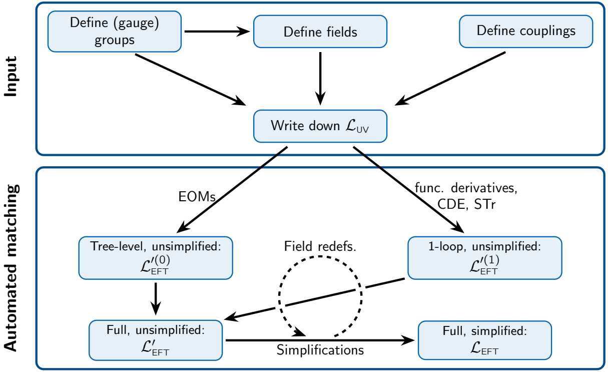
- Input in an intuitive, **user-friendly** form.

Matchete is a **Mathematica package** to automate **EFT matching**.



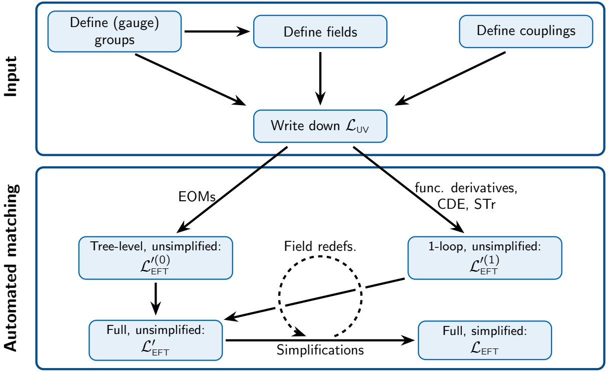
- Input in an intuitive, **user-friendly** form.
- effective Lagrangian up to **one-loop order** using **functional methods**.

Matchete is a **Mathematica package** to automate **EFT matching**.



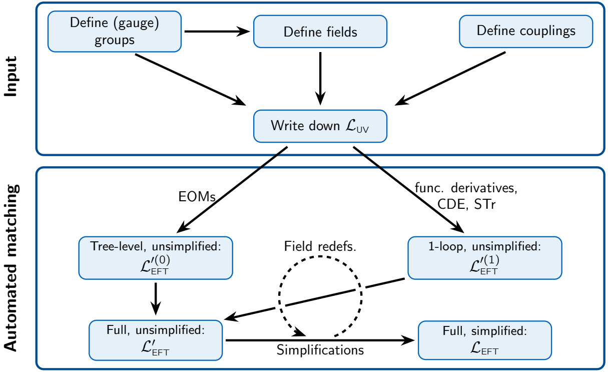
- Input in an intuitive, **user-friendly** form.
- effective Lagrangian up to **one-loop order** using **functional methods**.
- **no knowledge** of the target **basis** required!

Matchete is a **Mathematica package** to automate **EFT matching**.



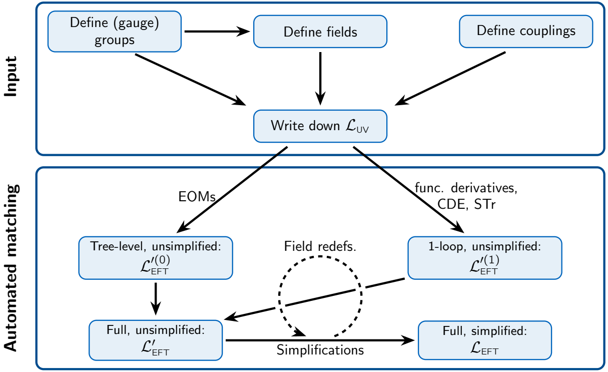
- Input in an intuitive, **user-friendly** form.
- effective Lagrangian up to **one-loop order** using **functional methods**.
- **no knowledge** of the target **basis** required!
- not limited to **dimension six**

Matchete is a **Mathematica package** to automate **EFT matching**.



- Input in an intuitive, **user-friendly** form.
- effective Lagrangian up to **one-loop order** using **functional methods**.
- **no knowledge** of the target **basis** required!
- not limited to **dimension six**
- automatic **simplifications and reductions**

Matchete is a **Mathematica package** to automate **EFT matching**.



- Input in an intuitive, **user-friendly** form.
- effective Lagrangian up to **one-loop order** using **functional methods**.
- **no knowledge** of the target **basis** required!
- not limited to **dimension six**
- automatic **simplifications and reductions**
- **human-readable** output

Toy Model Example

In[5]:=
```
DefineGaugeGroup[QED, U1, e, A]

DefineField[ψ, Fermion, Mass → 0, Charges → {QED[-1]}]
DefineField[ϕ, Scalar, Mass → Heavy, SelfConjugate → True]
DefineCoupling[y, SelfConjugate → True]

LUV = FreeLag[] + y[] × ϕ[] Bar[ψ[]] ** ψ[];
NiceForm@ LUV
```

Out[10]//NiceForm=

$$-\frac{1}{4} A^{\mu\nu 2} + \frac{1}{2} (D_\mu \phi)^2 - \frac{1}{2} M\phi^2 \phi^2 + i \left(\bar{\psi} \cdot \gamma_\mu \cdot D_\mu \psi\right) + y\, \phi\, \left(\bar{\psi} \cdot \psi\right)$$

In[11]:=
```
LEFT = EchoTiming[EOMSimplify@ Match[LUV]];
NiceForm@ LEFT
```

0.804748

Out[12]//NiceForm=

$$-\frac{1}{4} A^{\mu\nu 2} + i \left(\bar{\psi} \cdot \gamma_\mu \cdot D_\mu \psi\right) +$$
$$\left(\frac{1}{2} y^2 \frac{1}{M\phi^2} + \hbar \left(-\frac{3}{2}\frac{1}{\epsilon} y^4 \frac{1}{M\phi^2} + \frac{1}{M\phi^2}\left(-\frac{5}{4} y^4 - \frac{3}{2} y^4 \, \mathrm{Log}\left[\frac{\bar{\mu}^2}{M\phi^2}\right]\right)\right)\right) \left(\bar{\psi} \cdot \psi\right)^2 +$$
$$\hbar \left(-\frac{1}{3}\frac{1}{\epsilon} e^2 y^2 \frac{1}{M\phi^2} + \frac{1}{M\phi^2}\left(-\frac{4}{9} e^2 y^2 - \frac{1}{3} e^2 y^2 \, \mathrm{Log}\left[\frac{\bar{\mu}^2}{M\phi^2}\right]\right)\right) \left(\bar{\psi} \cdot \gamma_\mu \cdot \psi\right)^2 +$$
$$\hbar \left(\frac{1}{2}\frac{1}{\epsilon} e^2 y^2 \frac{1}{M\phi^2} + \frac{1}{M\phi^2}\left(\frac{3}{4} e^2 y^2 + \frac{1}{2} e^2 y^2 \, \mathrm{Log}\left[\frac{\bar{\mu}^2}{M\phi^2}\right]\right)\right) \left(\bar{\psi} \cdot \Gamma_{\mu\nu} \cdot \psi\right)^2$$

Toy Model Example

In[5]:=
```
DefineGaugeGroup[QED, U1, e, A]

DefineField[ψ, Fermion, Mass → 0, Charges → {QED[-1]}]
DefineField[ϕ, Scalar, Mass → Heavy, SelfConjugate → True]
DefineCoupling[y, SelfConjugate → True]

LUV = FreeLag[] + y[] × ϕ[] Bar[ψ[]] ** ψ[];
NiceForm@ LUV
```

define a $U(1)$ gauge symmetry

Out[10]//NiceForm=

$$-\frac{1}{4} A^{\mu\nu 2} + \frac{1}{2} (D_\mu \phi)^2 - \frac{1}{2} M\phi^2 \phi^2 + i \left(\bar{\psi} \cdot \gamma_\mu \cdot D_\mu \psi\right) + y\,\phi\,(\bar{\psi} \cdot \psi)$$

In[11]:=
```
LEFT = EchoTiming[EOMSimplify@ Match[LUV]];
NiceForm@ LEFT
```

0.804748

Out[12]//NiceForm=

$$-\frac{1}{4} A^{\mu\nu 2} + i \left(\bar{\psi} \cdot \gamma_\mu \cdot D_\mu \psi\right) +$$
$$\left(\frac{1}{2} y^2 \frac{1}{M\phi^2} + \hbar \left(-\frac{3}{2}\frac{1}{\epsilon} y^4 \frac{1}{M\phi^2} + \frac{1}{M\phi^2}\left(-\frac{5}{4} y^4 - \frac{3}{2} y^4 \,\mathrm{Log}\left[\frac{\bar{\mu}^2}{M\phi^2}\right]\right)\right)\right) (\bar{\psi} \cdot \psi)^2 +$$
$$\hbar \left(-\frac{1}{3}\frac{1}{\epsilon} e^2 y^2 \frac{1}{M\phi^2} + \frac{1}{M\phi^2}\left(-\frac{4}{9} e^2 y^2 - \frac{1}{3} e^2 y^2 \,\mathrm{Log}\left[\frac{\bar{\mu}^2}{M\phi^2}\right]\right)\right) (\bar{\psi} \cdot \gamma_\mu \cdot \psi)^2 +$$
$$\hbar \left(\frac{1}{2}\frac{1}{\epsilon} e^2 y^2 \frac{1}{M\phi^2} + \frac{1}{M\phi^2}\left(\frac{3}{4} e^2 y^2 + \frac{1}{2} e^2 y^2 \,\mathrm{Log}\left[\frac{\bar{\mu}^2}{M\phi^2}\right]\right)\right) (\bar{\psi} \cdot \Gamma_{\mu\nu} \cdot \psi)^2$$

Toy Model Example

In[5]:=
```
DefineGaugeGroup[QED, U1, e, A]

DefineField[ψ, Fermion, Mass → 0, Charges → {QED[-1]}]
DefineField[ϕ, Scalar, Mass → Heavy, SelfConjugate → True]
DefineCoupling[y, SelfConjugate → True]

LUV = FreeLag[] + y[] × ϕ[] Bar[ψ[]] ** ψ[];
NiceForm@ LUV
```

define a $U(1)$ gauge symmetry

fields and couplings

Out[10]//NiceForm=

$$-\frac{1}{4} A^{\mu\nu 2} + \frac{1}{2} (D_\mu \phi)^2 - \frac{1}{2} M\phi^2 \phi^2 + i \left(\bar{\psi} \cdot \gamma_\mu \cdot D_\mu \psi\right) + y \phi \left(\bar{\psi} \cdot \psi\right)$$

In[11]:=
```
LEFT = EchoTiming[EOMSimplify@ Match[LUV]];
NiceForm@ LEFT
```

0.804748

Out[12]//NiceForm=

$$-\frac{1}{4} A^{\mu\nu 2} + i \left(\bar{\psi} \cdot \gamma_\mu \cdot D_\mu \psi\right) +$$
$$\left(\frac{1}{2} y^2 \frac{1}{M\phi^2} + \hbar \left(-\frac{3}{2} \frac{1}{\epsilon} y^4 \frac{1}{M\phi^2} + \frac{1}{M\phi^2} \left(-\frac{5}{4} y^4 - \frac{3}{2} y^4 \mathrm{Log}\left[\frac{\bar{\mu}^2}{M\phi^2}\right]\right)\right)\right) \left(\bar{\psi} \cdot \psi\right)^2 +$$
$$\hbar \left(-\frac{1}{3} \frac{1}{\epsilon} e^2 y^2 \frac{1}{M\phi^2} + \frac{1}{M\phi^2} \left(-\frac{4}{9} e^2 y^2 - \frac{1}{3} e^2 y^2 \mathrm{Log}\left[\frac{\bar{\mu}^2}{M\phi^2}\right]\right)\right) \left(\bar{\psi} \cdot \gamma_\mu \cdot \psi\right)^2 +$$
$$\hbar \left(\frac{1}{2} \frac{1}{\epsilon} e^2 y^2 \frac{1}{M\phi^2} + \frac{1}{M\phi^2} \left(\frac{3}{4} e^2 y^2 + \frac{1}{2} e^2 y^2 \mathrm{Log}\left[\frac{\bar{\mu}^2}{M\phi^2}\right]\right)\right) \left(\bar{\psi} \cdot \Gamma_{\mu\nu} \cdot \psi\right)^2$$

Toy Model Example

In[5]:=
```
DefineGaugeGroup[QED, U1, e, A]

DefineField[ψ, Fermion, Mass → 0, Charges → {QED[-1]}]
DefineField[ϕ, Scalar, Mass → Heavy, SelfConjugate → True]
DefineCoupling[y, SelfConjugate → True]

LUV = FreeLag[] + y[] × ϕ[] Bar[ψ[]] ** ψ[];
NiceForm@ LUV
```

define a $U(1)$ gauge symmetry

fields and couplings

kinetic Lagrangian generated automatically

Out[10]//NiceForm=

$$-\frac{1}{4} A^{\mu\nu 2} + \frac{1}{2} (D_\mu \phi)^2 - \frac{1}{2} M\phi^2 \phi^2 + i \left(\bar{\psi} \cdot \gamma_\mu \cdot D_\mu \psi\right) + y\,\phi\,(\bar{\psi} \cdot \psi)$$

In[11]:=
```
LEFT = EchoTiming[EOMSimplify@ Match[LUV]];
NiceForm@ LEFT
```

0.804748

Out[12]//NiceForm=

$$-\frac{1}{4} A^{\mu\nu 2} + i \left(\bar{\psi} \cdot \gamma_\mu \cdot D_\mu \psi\right) +$$
$$\left(\frac{1}{2} y^2 \frac{1}{M\phi^2} + \hbar \left(-\frac{3}{2} \frac{1}{\epsilon} y^4 \frac{1}{M\phi^2} + \frac{1}{M\phi^2} \left(-\frac{5}{4} y^4 - \frac{3}{2} y^4 \operatorname{Log}\left[\frac{\bar{\mu}^2}{M\phi^2}\right]\right)\right)\right) (\bar{\psi} \cdot \psi)^2 +$$
$$\hbar \left(-\frac{1}{3} \frac{1}{\epsilon} e^2 y^2 \frac{1}{M\phi^2} + \frac{1}{M\phi^2} \left(-\frac{4}{9} e^2 y^2 - \frac{1}{3} e^2 y^2 \operatorname{Log}\left[\frac{\bar{\mu}^2}{M\phi^2}\right]\right)\right) (\bar{\psi} \cdot \gamma_\mu \cdot \psi)^2 +$$
$$\hbar \left(\frac{1}{2} \frac{1}{\epsilon} e^2 y^2 \frac{1}{M\phi^2} + \frac{1}{M\phi^2} \left(\frac{3}{4} e^2 y^2 + \frac{1}{2} e^2 y^2 \operatorname{Log}\left[\frac{\bar{\mu}^2}{M\phi^2}\right]\right)\right) (\bar{\psi} \cdot \Gamma_{\mu\nu} \cdot \psi)^2$$

Toy Model Example

In[5]:=
```
DefineGaugeGroup[QED, U1, e, A]

DefineField[ψ, Fermion, Mass → 0, Charges → {QED[-1]}]
DefineField[ϕ, Scalar, Mass → Heavy, SelfConjugate → True]
DefineCoupling[y, SelfConjugate → True]

LUV = FreeLag[] + y[] × ϕ[] Bar[ψ[]] ** ψ[];
NiceForm@ LUV
```

define a $U(1)$ gauge symmetry

fields and couplings

kinetic Lagrangian generated automatically

Out[10]//NiceForm=

$$-\frac{1}{4} A^{\mu\nu 2} + \frac{1}{2}(D_\mu\phi)^2 - \frac{1}{2} M\phi^2 \phi^2 + i\,(\bar\psi\cdot\gamma_\mu\cdot D_\mu\psi) + y\,\phi\,(\bar\psi\cdot\psi)$$

In[11]:=
```
LEFT = EchoTiming[EOMSimplify@ Match[LUV]];
NiceForm@ LEFT
```

one-loop matching dimension six, on-shell

0.804748

Out[12]//NiceForm=

$$-\frac{1}{4} A^{\mu\nu 2} + i\,(\bar\psi\cdot\gamma_\mu\cdot D_\mu\psi) +$$
$$\left(\frac{1}{2}y^2\frac{1}{M\phi^2} + \hbar\left(-\frac{3}{2}\frac{1}{\epsilon}y^4\frac{1}{M\phi^2} + \frac{1}{M\phi^2}\left(-\frac{5}{4}y^4 - \frac{3}{2}y^4\,\mathrm{Log}\left[\frac{\bar\mu^2}{M\phi^2}\right]\right)\right)\right)(\bar\psi\cdot\psi)^2 +$$
$$\hbar\left(-\frac{1}{3}\frac{1}{\epsilon}e^2y^2\frac{1}{M\phi^2} + \frac{1}{M\phi^2}\left(-\frac{4}{9}e^2y^2 - \frac{1}{3}e^2y^2\,\mathrm{Log}\left[\frac{\bar\mu^2}{M\phi^2}\right]\right)\right)(\bar\psi\cdot\gamma_\mu\cdot\psi)^2 +$$
$$\hbar\left(\frac{1}{2}\frac{1}{\epsilon}e^2y^2\frac{1}{M\phi^2} + \frac{1}{M\phi^2}\left(\frac{3}{4}e^2y^2 + \frac{1}{2}e^2y^2\,\mathrm{Log}\left[\frac{\bar\mu^2}{M\phi^2}\right]\right)\right)(\bar\psi\cdot\Gamma_{\mu\nu}\cdot\psi)^2$$

Matchete: An Ongoing Effort

Matchete had its **inititial release** `v0.1.0` in December 2022

- ▷ `https://gitlab.com/matchete/matchete`

[Matchete, EPJ-C 83, 662 (2023), `arXiv:2212.04510`]

Being in `v0.2.0` now, what has happened?

- Matching computation migrated from **Covariant Derivative Expansion** to manifest covariance using **Wilson Lines**

Matchete: An Ongoing Effort

Matchete had its **initital release** `v0.1.0` in December 2022

- ▷ `https://gitlab.com/matchete/matchete`

[Matchete, EPJ-C 83, 662 (2023), `arXiv:2212.04510`]

Being in `v0.2.0` now, what has happened?

- Matching computation migrated from **Covariant Derivative Expansion** to manifest covariance using **Wilson Lines**
- Improvements to **simplification routines** with regards to **flavor**, operator **symmetries**, etc.

Matchete: An Ongoing Effort

Matchete had its **initital release** `v0.1.0` in December 2022

▷ `https://gitlab.com/matchete/matchete`

[Matchete, EPJ-C 83, 662 (2023), `arXiv:2212.04510`]

Being in `v0.2.0` now, what has happened?

- Matching computation migrated from **Covariant Derivative Expansion** to manifest covariance using **Wilson Lines**
- Improvements to **simplification routines** with regards to **flavor**, operator **symmetries**, etc.
- **Optimizations** to many of the functions, all of which leads to substantial **performance increases**:

Version	`Match` [s]	`EOMSimplify` [s]
`v0.1.0`	74	281
`v0.2.0`	12	81
(internal)	12	12

Model: SM + S_1 + S_3 leptoquarks one-loop matching at dimension six running on Apple M3

Matchete: An Ongoing Effort

Matchete had its **initital release** `v0.1.0` in December 2022

- ▷ `https://gitlab.com/matchete/matchete`

[Matchete, EPJ-C 83, 662 (2023), `arXiv:2212.04510`]

Being in `v0.2.0` now, what has happened?

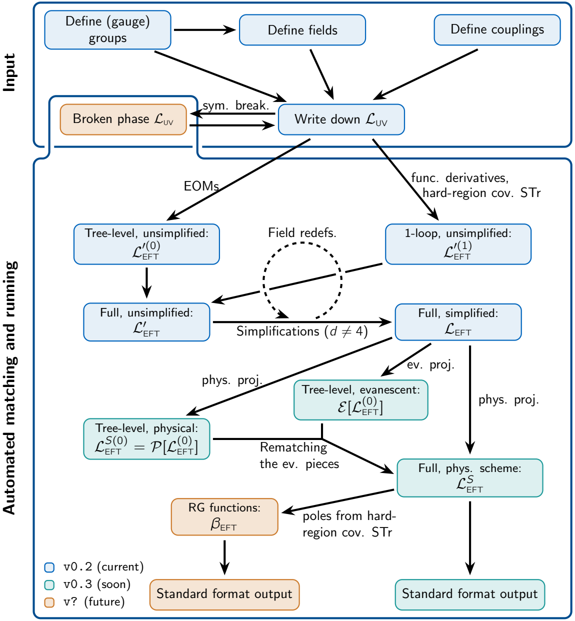
- Matching computation migrated from **Covariant Derivative Expansion** to manifest covariance using **Wilson Lines**
- Improvements to **simplification routines** with regards to **flavor**, operator **symmetries**, etc.
- **Optimizations** to many of the functions, all of which leads to substantial **performance increases**:

Version	`Match` [s]	`EOMSimplify` [s]
`v0.1.0`	74	281
`v0.2.0`	12	81
(internal)	12	12

Model: SM + S_1 + S_3 leptoquarks
one-loop matching at dimension six
running on Apple M3

- Various bugfixes, thanks for everyone's **feedback**, keep it coming!

What's next for Matchete?



Coming features:

- Automatic handling of **evanescent** contributions, simplifications now include Fierzing, ...
- Matching to a **given basis**
- **Interface** to other EFT codes
- Computation of **anomalous dimensions**
- Built-in documentation

- Spontaneous Symmetry Breaking & **massive vectors**

Evanescent Operators

Evanescent Contributions

Some simplification identities are **only valid** in $d = 4$ dimensions (**Fierz transformations**, Chisholm identities, ...)

Evanescent Contributions

Some simplification identities are **only valid** in $d = 4$ dimensions (**Fierz transformations**, Chisholm identities, ...)

When applying them, an $\mathcal{O}(\epsilon)$ **leftover** remains:

$$\mathcal{O}_d = \mathcal{P}\mathcal{O}_d + (\mathbf{1} - \mathcal{P})\mathcal{O}_d$$

The **evanescent remnant** can have non-vanishing **loop** matrix elements!

Some simplification identities are **only valid** in $d = 4$ dimensions (**Fierz transformations**, Chisholm identities, ...)

When applying them, an $\mathcal{O}(\epsilon)$ **leftover** remains:

$$\mathcal{O}_d = \mathcal{P}\mathcal{O}_d + (\mathbf{1} - \mathcal{P})\mathcal{O}_d$$

The **evanescent remnant** can have non-vanishing **loop** matrix elements!

These contributions can be **absorbed** into a **finite renormalization** of other **(physical) operators**.

Once done, the **evanescent operator** can be (has to be) dropped.

Evanescent Operators - An Example

Consider 2HDM, integrate out second Higgs doublet Φ:

$$\mathcal{L} \supset \mathcal{L}_{\text{SM}} + D_\mu \Phi^\dagger D^\mu \Phi - M_\Phi^2 \Phi^\dagger \Phi - \left(y_{\Phi e}^{pr} \, \overline{\ell}_p \Phi e_r + \text{h.c.} \right)$$

Evanescent Operators - An Example

Consider 2HDM, integrate out second Higgs doublet Φ:

$$\mathcal{L} \supset \mathcal{L}_{\text{SM}} + D_\mu \Phi^\dagger D^\mu \Phi - M_\Phi^2 \Phi^\dagger \Phi - \left(y_{\Phi e}^{pr} \, \bar{\ell}_p \Phi e_r + \text{h.c.} \right)$$

Generates, amongst many other operators:

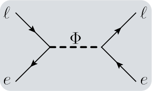
Evanescent Operators - An Example

Consider 2HDM, integrate out second Higgs doublet Φ:

$$\mathcal{L} \supset \mathcal{L}_{\text{SM}} + D_\mu \Phi^\dagger D^\mu \Phi - M_\Phi^2 \Phi^\dagger \Phi - \left(y_{\Phi e}^{pr} \, \bar{\ell}_p \Phi e_r + \text{h.c.} \right)$$

Generates, amongst many other operators:

$$[R_{\ell e}]^{prst} = (\bar{\ell}_p e_r)(\bar{e}_s \ell_t)$$



Consider 2HDM, integrate out second Higgs doublet Φ:

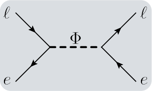
$$\mathcal{L} \supset \mathcal{L}_{\text{SM}} + D_\mu \Phi^\dagger D^\mu \Phi - M_\Phi^2 \Phi^\dagger \Phi - \left(y_{\Phi e}^{pr} \, \bar{\ell}_p \Phi e_r + \text{h.c.} \right)$$

Generates, amongst many other operators:

$$[R_{\ell e}]^{prst} = (\bar{\ell}_p e_r)(\bar{e}_s \ell_t)$$

$$\downarrow$$

$$-\frac{1}{2}[Q_{\ell e}]^{ptsr} = -\frac{1}{2}(\bar{\ell}_p \gamma_\mu \ell_t)(\bar{e}_s \gamma^\mu e_r)$$



Evanescent Operators - An Example

Consider 2HDM, integrate out second Higgs doublet Φ:

$$\mathcal{L} \supset \mathcal{L}_{\mathrm{SM}} + D_\mu \Phi^\dagger D^\mu \Phi - M_\Phi^2 \, \Phi^\dagger \Phi - \left(y_{\Phi e}^{pr} \, \bar{\ell}_p \Phi e_r + \mathrm{h.c.} \right)$$

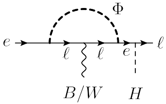
Generates, amongst many other operators:

$$[R_{\ell e}]^{prst} = (\bar{\ell}_p e_r)(\bar{e}_s \ell_t)$$

$$\downarrow$$

$$-\frac{1}{2}[Q_{\ell e}]^{ptsr} = -\frac{1}{2}(\bar{\ell}_p \gamma_\mu \ell_t)(\bar{e}_s \gamma^\mu e_r)$$

$$[Q_{eW}]^{pr} = \bar{\ell}_p \sigma_{\mu\nu} e_r \, \tau^I H W^{I\mu\nu}$$



Evanescent Operators - An Example

Consider 2HDM, integrate out second Higgs doublet Φ:

$$\mathcal{L} \supset \mathcal{L}_{\text{SM}} + D_\mu \Phi^\dagger D^\mu \Phi - M_\Phi^2 \, \Phi^\dagger \Phi - \left(y_{\Phi e}^{pr} \, \bar{\ell}_p \Phi e_r + \text{h.c.} \right)$$

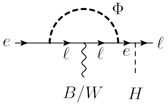
Generates, amongst many other operators:

$$[R_{\ell e}]^{prst} = (\bar{\ell}_p e_r)(\bar{e}_s \ell_t)$$

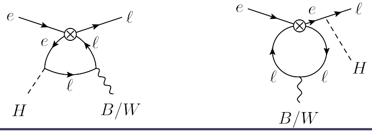
$$\downarrow$$

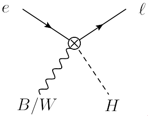
$$-\frac{1}{2}[Q_{\ell e}]^{ptsr} = -\frac{1}{2}(\bar{\ell}_p \gamma_\mu \ell_t)(\bar{e}_s \gamma^\mu e_r)$$

$$[Q_{eW}]^{pr} = \bar{\ell}_p \sigma_{\mu\nu} e_r \, \tau^I H W^{I\mu\nu}$$



The operators $R_{\ell e}$ and $Q_{\ell e}$ generate **different** amplitudes for the **dipole transitions**!





Evanescent Operators - An Example

Consider 2HDM, integrate out second Higgs doublet Φ:

$$\mathcal{L} \supset \mathcal{L}_{\text{SM}} + D_\mu \Phi^\dagger D^\mu \Phi - M_\Phi^2 \, \Phi^\dagger \Phi - \left(y_{\Phi e}^{pr} \, \bar{\ell}_p \Phi e_r + \text{h.c.} \right)$$

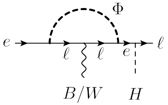
Generates, amongst many other operators:

$$[R_{\ell e}]^{prst} = (\bar{\ell}_p e_r)(\bar{e}_s \ell_t)$$

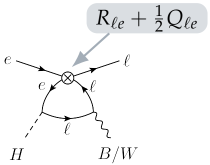
$$\downarrow$$

$$-\frac{1}{2}[Q_{\ell e}]^{ptsr} = -\frac{1}{2}(\bar{\ell}_p \gamma_\mu \ell_t)(\bar{e}_s \gamma^\mu e_r)$$

$$[Q_{eW}]^{pr} = \bar{\ell}_p \sigma_{\mu\nu} e_r \, \tau^I H W^{I\mu\nu}$$



The operators $R_{\ell e}$ and $Q_{\ell e}$ generate **different** amplitudes for the **dipole transitions**!



This graph contributes through $\epsilon \cdot (1/\epsilon_{\text{UV}})$ and can be absorbed into a **finite counterterm** to Q_{eW}!

Handling Evanescence in Matchete

Matchete is equipped to handle such operators **automatically**.

Loading the SM and defining a redundant operator:

```
In[4]:= LSM = LoadModel["SM"];

DefineCoupling[c, Indices → {Flavor, Flavor, Flavor, Flavor}];
redundant =
  PlusHc[c[p, r, s, t] Bar@l[i, p] ** e[r] Bar@e[s] ** l[i, t]] //
   RelabelIndices;
redundant // NiceForm
```

Out[7]//NiceForm=

$$c^{prst} \left(\bar{e}^{s} \cdot P_L \cdot l^{it}\right) \left(\bar{l}_i^{p} \cdot P_R \cdot e^{r}\right) + \bar{c}^{prst} \left(\bar{e}^{r} \cdot P_L \cdot l^{ip}\right) \left(\bar{l}_i^{t} \cdot P_R \cdot e^{s}\right)$$

The naive 4d identity:

```
In[8]:= NiceForm@GreensSimplify[redundant,
  ReductionIdentities → FourDimensional]
```

Out[8]//NiceForm=

$$\frac{1}{2} \left(-\bar{c}^{rspt} - c^{tpsr}\right) \left(\bar{e}^{s} \cdot \gamma_\mu P_R \cdot e^{p}\right) \left(\bar{l}_i^{t} \cdot \gamma_\mu P_L \cdot l^{ir}\right)$$

With the “Evanescent” option, a term is added:

```
In[9]:= NiceForm@GreensSimplify[redundant,
  ReductionIdentities → Evanescent]
```

Out[9]//NiceForm=

$$\bar{c}^{prst} E_1^{rpts} + c^{prst} E_1^{stpr} + \frac{1}{2} \left(-\bar{c}^{rspt} - c^{tpsr}\right) \left(\bar{e}^{s} \cdot \gamma_\mu P_R \cdot e^{p}\right) \left(\bar{l}_i^{t} \cdot \gamma_\mu P_L \cdot l^{ir}\right)$$

Simplification routines will include option “`ReductionIdentities`”:

- “`FourDimensional`” treats everything as an expression in $d = 4$.

Handling Evanescence in Matchete

Matchete is equipped to handle such operators **automatically**.

Loading the SM and defining a redundant operator:

```
In[4]:= LSM = LoadModel["SM"];

DefineCoupling[c, Indices → {Flavor, Flavor, Flavor, Flavor}];
redundant =
  PlusHc[c[p, r, s, t] Bar@l[i, p] ** e[r] Bar@e[s] ** l[i, t]] //
   RelabelIndices;
redundant // NiceForm
```

Out[7]//NiceForm=

$$c^{prst}\,(\bar{e}^s \cdot P_L \cdot l^{it})\,(\bar{l}_i^p \cdot P_R \cdot e^r) + \bar{c}^{prst}\,(\bar{e}^r \cdot P_L \cdot l^{ip})\,(\bar{l}_i^t \cdot P_R \cdot e^s)$$

The naive 4d identity:

```
In[8]:= NiceForm@GreensSimplify[redundant,
  ReductionIdentities → FourDimensional]
```

Out[8]//NiceForm=

$$\frac{1}{2}\left(-\bar{c}^{rspt} - c^{tpsr}\right)(\bar{e}^s \cdot \gamma_\mu P_R \cdot e^p)\,(\bar{l}_i^t \cdot \gamma_\mu P_L \cdot l^{ir})$$

With the “Evanescent” option, a term is added:

```
In[9]:= NiceForm@GreensSimplify[redundant,
  ReductionIdentities → Evanescent]
```

Out[9]//NiceForm=

$$\bar{c}^{prst} E_1^{rpts} + c^{prst} E_1^{stpr} + \frac{1}{2}\left(-\bar{c}^{rspt} - c^{tpsr}\right)(\bar{e}^s \cdot \gamma_\mu P_R \cdot e^p)\,(\bar{l}_i^t \cdot \gamma_\mu P_L \cdot l^{ir})$$

Simplification routines will include option “`ReductionIdentities`”:

- “`FourDimensional`” treats everything as an expression in $d = 4$.
- “`Evanescent`” will write out an evanescent operator.

Matchete is equipped to handle such operators **automatically**.

Loading the SM and defining a redundant operator:

In[4]:=
```
LSM = LoadModel["SM"];

DefineCoupling[c, Indices → {Flavor, Flavor, Flavor, Flavor}];
redundant =
  PlusHc[c[p, r, s, t] Bar@l[i, p] ** e[r] Bar@e[s] ** l[i, t]] //
   RelabelIndices;
redundant // NiceForm
```

Out[7]//NiceForm=

$$c^{prst}\,(\bar{e}^{s}\cdot P_L\cdot l^{it})\,(\bar{l}^{p}_{i}\cdot P_R\cdot e^{r}) + \bar{c}^{prst}\,(\bar{e}^{r}\cdot P_L\cdot l^{ip})\,(\bar{l}^{t}_{i}\cdot P_R\cdot e^{s})$$

The naive 4d identity:

In[8]:=
```
NiceForm@GreensSimplify[redundant,
  ReductionIdentities → FourDimensional]
```

Out[8]//NiceForm=

$$\frac{1}{2}\left(-\bar{c}^{rspt} - c^{tpsr}\right)\,(\bar{e}^{s}\cdot\gamma_\mu P_R\cdot e^{p})\,(\bar{l}^{t}_{i}\cdot\gamma_\mu P_L\cdot l^{ir})$$

With the "Evanescent" option, a term is added:

In[9]:=
```
NiceForm@GreensSimplify[redundant,
  ReductionIdentities → Evanescent]
```

Out[9]//NiceForm=

$$\bar{c}^{prst}\,E_1^{rpts} + c^{prst}\,E_1^{stpr} + \frac{1}{2}\left(-\bar{c}^{rspt} - c^{tpsr}\right)\,(\bar{e}^{s}\cdot\gamma_\mu P_R\cdot e^{p})\,(\bar{l}^{t}_{i}\cdot\gamma_\mu P_L\cdot l^{ir})$$

Simplification routines will include option "`ReductionIdentities`":

- "`FourDimensional`" treats everything as an expression in $d = 4$.
- "`Evanescent`" will write out an evanescent operator.
- "`dDimensional`" only applies exact identities (valid in arbitrary dimensions).

Handling Evanescence in Matchete

Matchete is equipped to handle such operators **automatically**.

Loading the SM and defining a redundant operator:

```
In[4]:= LSM = LoadModel["SM"];

        DefineCoupling[c, Indices → {Flavor, Flavor, Flavor, Flavor}];
        redundant =
          PlusHc[c[p, r, s, t] Bar@l[i, p] ** e[r] Bar@e[s] ** l[i, t]] //
           RelabelIndices;
        redundant // NiceForm
```

Out[7]//NiceForm=

$$c^{prst}\left(\bar{e}^{s}\cdot P_L\cdot l^{it}\right)\left(\bar{l}^{p}_{i}\cdot P_R\cdot e^{r}\right)+\bar{c}^{prst}\left(\bar{e}^{r}\cdot P_L\cdot l^{ip}\right)\left(\bar{l}^{t}_{i}\cdot P_R\cdot e^{s}\right)$$

The naive 4d identity:

```
In[8]:= NiceForm@GreensSimplify[redundant,
          ReductionIdentities → FourDimensional]
```

Out[8]//NiceForm=

$$\frac{1}{2}\left(-\bar{c}^{rspt}-c^{tpsr}\right)\left(\bar{e}^{s}\cdot\gamma_\mu P_R\cdot e^{p}\right)\left(\bar{l}^{t}_{i}\cdot\gamma_\mu P_L\cdot l^{ir}\right)$$

With the “Evanescent” option, a term is added:

```
In[9]:= NiceForm@GreensSimplify[redundant,
          ReductionIdentities → Evanescent]
```

Out[9]//NiceForm=

$$\bar{c}^{prst}E_1^{rpts}+c^{prst}E_1^{stpr}+\frac{1}{2}\left(-\bar{c}^{rspt}-c^{tpsr}\right)\left(\bar{e}^{s}\cdot\gamma_\mu P_R\cdot e^{p}\right)\left(\bar{l}^{t}_{i}\cdot\gamma_\mu P_L\cdot l^{ir}\right)$$

Simplification routines will include option “`ReductionIdentities`”:

- “`FourDimensional`” treats everything as an expression in $d=4$.
- “`Evanescent`” will write out an evanescent operator.
- “`dDimensional`” only applies exact identities (valid in arbitrary dimensions).
- “`EvanescenceFree`” immediately applies finite renormalization shifts.

Handling Evanescence in Matchete

Matchete is equipped to handle such operators **automatically**.

When a complete Lagrangian is given, the simplifications evaluate the Evanescent shifts automatically and add them to the Lagrangian:

```
In[10]:= Lreduced = GreensSimplify[LSM + redundant,
            ReductionIdentities → EvanescenceFree];
         Lreduced - LSM // GreensSimplify // NiceForm
```

Out[11]//NiceForm=

$$
\begin{aligned}
&\hbar \left(\bar{Y}_e^{st}\,\bar{Y}_e^{uv}\,Y_e^{ut}\,c^{svrp} + \bar{Y}_e^{tu}\,\bar{Y}_e^{vs}\,Y_e^{vu}\,\bar{c}^{prst}\right) \bar{H}_i\,\bar{H}_j\,H^j \left(\bar{e}^r \cdot P_L \cdot l^{ip}\right) + \\
&\hbar \left(\bar{Y}_e^{uv}\,Y_e^{sv}\,Y_e^{ut}\,\bar{c}^{stpr} + \bar{Y}_e^{st}\,Y_e^{su}\,Y_e^{vt}\,c^{rpuv}\right) \bar{H}_i\,H^i\,H^j \left(\bar{l}_j^r \cdot P_R \cdot e^p\right) + \\
&\frac{1}{2}\hbar \left(\bar{Y}_e^{ts}\,\bar{c}^{prst} + \bar{Y}_e^{st}\,c^{strp}\right) D^2\bar{H}_i \left(\bar{e}^r \cdot P_L \cdot l^{ip}\right) + \\
&\frac{1}{2}\hbar \left(Y_e^{st}\,\bar{c}^{stpr} + Y_e^{ts}\,c^{rpst}\right) D^2 H^i \left(\bar{l}_i^r \cdot P_R \cdot e^p\right) + \\
&\frac{3\,i}{8}\hbar\, g_Y \left(\bar{Y}_e^{ts}\,\bar{c}^{prst} + \bar{Y}_e^{st}\,c^{strp}\right) \bar{H}_i\,B^{\mu\nu} \left(\bar{e}^r \cdot \Gamma_{\mu\nu} P_L \cdot l^{ip}\right) - \\
&\frac{i}{4}\hbar\, g_L \left(\bar{Y}_e^{ts}\,\bar{c}^{prst} + \bar{Y}_e^{st}\,c^{strp}\right) \bar{H}_i\,W^{\mu\nu I} \left(\bar{e}^r \cdot \Gamma_{\mu\nu} P_L \cdot l^{jp}\right) T_j^{Ii} + \\
&\frac{3\,i}{8}\hbar\, g_Y \left(Y_e^{st}\,\bar{c}^{stpr} + Y_e^{ts}\,c^{rpst}\right) H^i\,B^{\mu\nu} \left(\bar{l}_i^r \cdot \Gamma_{\mu\nu} P_R \cdot e^p\right) - \\
&\frac{i}{4}\hbar\, g_L \left(Y_e^{st}\,\bar{c}^{stpr} + Y_e^{ts}\,c^{rpst}\right) H^j\,W^{\mu\nu I} \left(\bar{l}_i^r \cdot \Gamma_{\mu\nu} P_R \cdot e^p\right) T_j^{Ii} + \\
&\frac{1}{4}\hbar \left(-\bar{Y}_e^{vt}\,Y_e^{up}\,\bar{c}^{usrv} - \bar{Y}_e^{vt}\,Y_e^{ur}\,\bar{c}^{uspv} - \bar{Y}_e^{us}\left(Y_e^{vp}\,c^{urtv} + Y_e^{vr}\,c^{uptv}\right)\right) \\
&\quad \left(\bar{e}^s \cdot \gamma_\mu P_R \cdot e^r\right) \left(\bar{e}^t \cdot \gamma_\mu P_R \cdot e^p\right) + \left(\frac{1}{2}\left(-\bar{c}^{rspt} - c^{tpsr}\right) + \right. \\
&\quad \left. \frac{1}{4}\hbar \left(-\bar{Y}_e^{vs}\,Y_e^{tu}\,\bar{c}^{rupv} - \bar{Y}_e^{us}\,Y_e^{tv}\,c^{upvr} - \bar{Y}_e^{rv}\,Y_e^{up}\,\bar{c}^{usvt} - \bar{Y}_e^{ru}\,Y_e^{vp}\,c^{tusv}\right)\right) \\
&\quad \left(\bar{e}^s \cdot \gamma_\mu P_R \cdot e^p\right) \left(\bar{l}_i^t \cdot \gamma_\mu P_L \cdot l^{ir}\right) + \frac{1}{4}\hbar \left(-\bar{Y}_e^{rv}\,Y_e^{su}\,\bar{c}^{puvt} - \bar{Y}_e^{pu}\,Y_e^{tv}\,c^{suvr}\right) \\
&\quad \left(\bar{l}_i^s \cdot \gamma_\mu P_L \cdot l^{ir}\right) \left(\bar{l}_j^t \cdot \gamma_\mu P_L \cdot l^{jp}\right)
\end{aligned}
$$

Simplification routines will include option "`ReductionIdentities`":

- "`FourDimensional`" treats everything as an expression in $d = 4$.
- "`Evanescent`" will write out an evanescent operator.
- "`dDimensional`" only applies exact identities (valid in arbitrary dimensions).
- "`EvanescenceFree`" immediately applies finite renormalization shifts.

Matching to a Specific Basis

Matching to a Specific Basis

As it stands, Matchete does **not** require a **target Basis** as an input - it is computed from the UV theory **directly**.

Matching to a Specific Basis

As it stands, Matchete does **not** require a **target Basis** as an input - it is computed from the UV theory **directly**.

Nice, since you do not have to **think** about a target basis, do not run the risk of forgetting operators to match.

Matching to a Specific Basis

As it stands, Matchete does **not** require a **target Basis** as an input - it is computed from the UV theory **directly**.

Nice, since you do not have to **think** about a target basis, do not run the risk of forgetting operators to match.

However, it means that you are at Matchete's mercy to simplify expressions to the operators you want to study.

Matching to a Specific Basis

As it stands, Matchete does **not** require a **target Basis** as an input - it is computed from the UV theory **directly**.

Nice, since you do not have to **think** about a target basis, do not run the risk of forgetting operators to match.

However, it means that you are at Matchete's mercy to simplify expressions to the operators you want to study.

Solution is **simple** in principle:

- Provide the matching result **and** a Lagrangian in your target basis (with undetermined couplings)

Matching to a Specific Basis

As it stands, Matchete does **not** require a **target Basis** as an input - it is computed from the UV theory **directly**.

Nice, since you do not have to **think** about a target basis, do not run the risk of forgetting operators to match.

However, it means that you are at Matchete's mercy to simplify expressions to the operators you want to study.

Solution is **simple** in principle:

- Provide the matching result **and** a Lagrangian in your target basis (with undetermined couplings)
- Run both through the simplifications, they should simplify to the same basis!

Matching to a Specific Basis

As it stands, Matchete does **not** require a **target Basis** as an input - it is computed from the UV theory **directly**.

Nice, since you do not have to **think** about a target basis, do not run the risk of forgetting operators to match.

However, it means that you are at Matchete's mercy to simplify expressions to the operators you want to study.

Solution is **simple** in principle:

- Provide the matching result **and** a Lagrangian in your target basis (with undetermined couplings)
- Run both through the simplifications, they should simplify to the same basis!
- Tweak the target-basis couplings such that the two outputs are equal.

Matching to a Specific Basis

As it stands, Matchete does **not** require a **target Basis** as an input - it is computed from the UV theory **directly**.

Nice, since you do not have to **think** about a target basis, do not run the risk of forgetting operators to match.

However, it means that you are at Matchete's mercy to simplify expressions to the operators you want to study.

Solution is **simple** in principle:

- Provide the matching result **and** a Lagrangian in your target basis (with undetermined couplings)
- Run both through the simplifications, they should simplify to the same basis!
- Tweak the target-basis couplings such that the two outputs are equal.

The result is a **mapping** between the **effective couplings** of the two Lagrangians.

Matching to a Specific Basis

This functionality is finally coming to **Matchete**, with a **convenient** function to map **directly to the SMEFT**:

```
newLag = GreensSimplify[LEFT, ReductionIdentities → FourDimensional] /. ϵ^-1 → 0;
```

```
coefs = MapToWarsaw@newLag;
```

The Warsaw Lagrangian currently simplifies with 4D IDENTITIES!

```
coefs // TableForm // NiceForm
```

NiceForm=

$$g_Y \to g_Y - \frac{1}{18}\hbar\, g_Y{}^3 \operatorname{Log}\left[\frac{\bar\mu^2}{M1^2}\right] - \frac{1}{6}\hbar\, g_Y{}^3 \operatorname{Log}\left[\frac{\bar\mu^2}{M3^2}\right]$$

$$g_s \to g_s - \frac{1}{12}\hbar\, g_s{}^3 \operatorname{Log}\left[\frac{\bar\mu^2}{M1^2}\right] - \frac{1}{4}\hbar\, g_s{}^3 \operatorname{Log}\left[\frac{\bar\mu^2}{M3^2}\right]$$

$$g_L \to g_L - \hbar\, g_L{}^3 \operatorname{Log}\left[\frac{\bar\mu^2}{M3^2}\right]$$

$$Y_d{}^{i1\_i2\_} \to Y_d{}^{i1i2}$$

$$Y_e{}^{i1\_i2\_} \to Y_e{}^{i1i2}$$

$$Y_u{}^{i1\_i2\_} \to Y_u{}^{i1i2}$$

$$\lambda \to \lambda$$

$$\mu 2 \to 3\hbar\, \lambda H1\, M1^2 + 9\hbar\, \lambda H3\, M3^2 + \mu 2 + 3\hbar\, \lambda H1\, M1^2 \operatorname{Log}\left[\frac{\bar\mu^2}{M1^2}\right] + 9\hbar\, \lambda H3\, M3^2 \operatorname{Log}\left[\frac{\bar\mu^2}{M3^2}\right]$$

$$C_{llHH}{}^{i1\_i2\_} \to 0$$

$$C_{dB}{}^{i1\_i2\_} \to \frac{1}{18}\hbar\, g_Y\, \overline{y1L}^{i1p}\, y1L^{rp}\, Y_d{}^{ri2} \frac{1}{M1^2} + \frac{1}{6}\hbar\, g_Y\, \overline{y3L}^{i1p}\, y3L^{rp}\, Y_d{}^{ri2} \frac{1}{M3^2}$$

$$C_{dd}{}^{i1\_i2\_i3\_i4\_} \to -\frac{1}{240}\hbar\, g_s{}^4 \frac{1}{M1^2}\delta^{i1i4}\delta^{i2i3} - \frac{1}{80}\hbar\, g_s{}^4 \frac{1}{M3^2}\delta^{i1i4}\delta^{i2i3} + \frac{1}{720}\hbar\, g_s{}^4 \frac{1}{M1^2}\delta^{i1i2}\delta^{i3i4} - \frac{1}{1620}\hbar\, g_Y{}^4 \frac{1}{M1^2}\delta^{i1i2}\delta^{i3i4} + \frac{1}{240}\hbar\, g_s{}^4$$

$$C_{dG}{}^{i1\_i2\_} \to -\frac{1}{24}\hbar\, g_s\, \overline{y1L}^{i1p}\, y1L^{rp}\, Y_d{}^{ri2} \frac{1}{M1^2} - \frac{1}{8}\hbar\, g_s\, \overline{y3L}^{i1p}\, y3L^{rp}\, Y_d{}^{ri2} \frac{1}{M3^2}$$

$$C_{dH}{}^{i1\_i2\_} \to \frac{1}{12}\hbar\, \overline{y1L}^{pr}\, y1L^{tr}\, \overline{Y_d}^{ps}\, Y_d{}^{ti2}\, Y_d{}^{i1s} \frac{1}{M1^2} + \frac{1}{4}\hbar\, \overline{y3L}^{pr}\, y3L^{tr}\, \overline{Y_d}^{ps}\, Y_d{}^{ti2}\, Y_d{}^{i1s} \frac{1}{M3^2} - \frac{1}{20}\hbar\, Y_d{}^{i1i2}\, g_L{}^4 \frac{1}{M3^2} - \frac{1}{2}\hbar\, \overline{y3L}^{i1p}\, y3L^{st}\, Y_d{}^{si}$$

$$C_{duq}{}^{i1\_i2\_i3\_i4\_} \to 0$$

Interfacing with Existing Codes

The mapping to Warsaw coefficients allows us now to develop **translation functions** for **other codes**, to enable a **streamlined pheno workflow**.

A **WIP** example: interfacing to `smelli/jelli` (see Aleks' talk)

```
ExportSMEFT["full_stack", LEFT]
```

full_stack.json

Schema: <No Schema Selected>


```
[
  "wilson_coefficients",
  [
    [
      "C_llphiphi",
      "0"
    ],
    [
      "C_dB",
      "-0.003472222222222222*(einsum_1*gY)\/(M_1**2*Pi**2) - (einsum_2*gY)\/(96.*M_3**2*Pi**2)"
    ],
    [
      "C_dd",
      "-0.0002604166666666666*(einsum_4*gs**4*(3*M_1**2 + M_3**2))\/(M_1**2*M_3**2*Pi**2) + (e
    ],
    [
      "C_dG",
      "(einsum_1*gs)\/(384.*M_1**2*Pi**2) + (einsum_2*gs)\/(128.*M_3**2*Pi**2)"
    ],
    [
      "C_dphi",
      "einsum_5*(-0.003125*(gL**4 - 20*lambda_H_3p**2)\/(M_3**2*Pi**2) - (3*lambda_H_13*lambda_
    ],
    [
      "C_duql",
      "0"
    ],
    [
      "C_duue",
      "-einsum_11"
    ],
    [
```

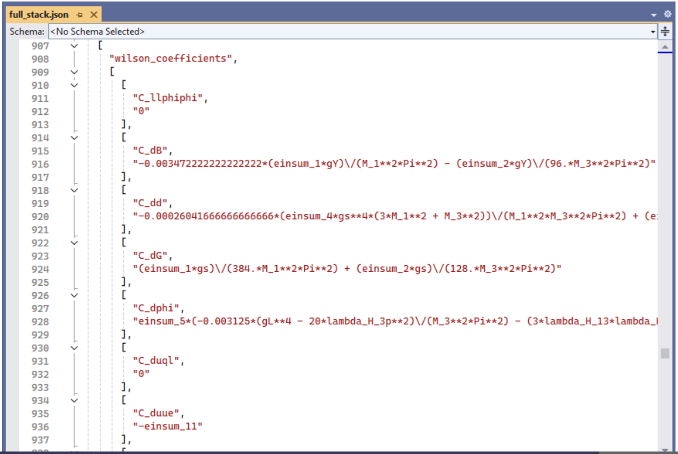

- output provides the details for target code to construct a function from UV parameters

Interfacing with Existing Codes

The mapping to Warsaw coefficients allows us now to develop **translation functions** for **other codes**, to enable a **streamlined pheno workflow**.

A **WIP** example: interfacing to `smelli/jelli` (see Aleks' talk)

```
ExportSMEFT["full_stack", LEFT]
```



- output provides the details for target code to construct a function from UV parameters
- Einstein summation has to be made explicit

Interfacing with Existing Codes

The mapping to Warsaw coefficients allows us now to develop **translation functions** for **other codes**, to enable a **streamlined pheno workflow**.

A **WIP** example: interfacing to `smelli/jelli` (see Aleks' talk)

```
ExportSMEFT["full_stack", LEFT]
```

full_stack.json

Schema: <No Schema Selected>


```
[
  "wilson_coefficients",
  [
    [
      "C_llphiphi",
      "0"
    ],
    [
      "C_dB",
      "-0.003472222222222222*(einsum_1*gY)\/(M_1**2*Pi**2) - (einsum_2*gY)\/(96.*M_3**2*Pi**2)"
    ],
    [
      "C_dd",
      "-0.0002604166666666666*(einsum_4*gs**4*(3*M_1**2 + M_3**2))\/(M_1**2*M_3**2*Pi**2) + (e
    ],
    [
      "C_dG",
      "(einsum_1*gs)\/(384.*M_1**2*Pi**2) + (einsum_2*gs)\/(128.*M_3**2*Pi**2)"
    ],
    [
      "C_dphi",
      "einsum_5*(-0.003125*(gL**4 - 20*lambda_H_3p**2)\/(M_3**2*Pi**2) - (3*lambda_H_13*lambda_
    ],
    [
      "C_duql",
      "0"
    ],
    [
      "C_duue",
      "-einsum_11"
    ],
    [
```


- output provides the details for target code to construct a function from UV parameters
- Einstein summation has to be made explicit
- target code should know loop functions to handle equal mass limit:

$$\int_\ell \frac{1}{\ell^2-m_1^2}\frac{1}{\ell^2-m_2^2} = \frac{A_0(m_1^2)-A_0(m_2^2)}{m_1^2-m_2^2}$$

Interfacing with Existing Codes

The mapping to Warsaw coefficients allows us now to develop **translation functions** for **other codes**, to enable a **streamlined pheno workflow**.

A **WIP** example: interfacing to `smelli/jelli` (see Aleks' talk)

```
ExportSMEFT["full_stack", LEFT]
```

full_stack.json

Schema: <No Schema Selected>


```
[
  "wilson_coefficients",
  [
    [
      "C_llphiphi",
      "0"
    ],
    [
      "C_dB",
      "-0.003472222222222222*(einsum_1*gY)\/(M_1**2*Pi**2) - (einsum_2*gY)\/(96.*M_3**2*Pi**2)"
    ],
    [
      "C_dd",
      "-0.0002604166666666666*(einsum_4*gs**4*(3*M_1**2 + M_3**2))\/(M_1**2*M_3**2*Pi**2) + (e
    ],
    [
      "C_dG",
      "(einsum_1*gs)\/(384.*M_1**2*Pi**2) + (einsum_2*gs)\/(128.*M_3**2*Pi**2)"
    ],
    [
      "C_dphi",
      "einsum_5*(-0.003125*(gL**4 - 20*lambda_H_3p**2)\/(M_3**2*Pi**2) - (3*lambda_H_13*lambda_
    ],
    [
      "C_duql",
      "0"
    ],
    [
      "C_duue",
      "-einsum_11"
    ],
    [
```


- output provides the details for target code to construct a function from UV parameters
- Einstein summation has to be made explicit
- target code should know loop functions to handle equal mass limit:
  $$\int_\ell \frac{1}{\ell^2 - m_1^2}\frac{1}{\ell^2 - m_2^2} = \frac{A_0(m_1^2) - A_0(m_2^2)}{m_1^2 - m_2^2}$$
- predefined contractions and loop functions make evaluation fast, allowing for large MC scans

Interfacing with Existing Codes

The `jelli` export is currently being worked out in collaboration with Peter Stangl, but we are **happy** to build interfaces to **other codes**!

The vision for the ideal BSM workflow:

- Easily define a BSM model

Interfacing with Existing Codes

The `jelli` export is currently being worked out in collaboration with Peter Stangl, but we are **happy** to build interfaces to **other codes**!

The vision for the ideal BSM workflow:

- Easily define a BSM model
- Compute one-loop matching to the SMEFT

Interfacing with Existing Codes

The `jelli` export is currently being worked out in collaboration with Peter Stangl, but we are **happy** to build interfaces to **other codes**!

The vision for the ideal BSM workflow:

- Easily define a BSM model
- Compute one-loop matching to the SMEFT
- RG-evolve coefficients (either in-house or interfacing to other tools, i.e. `DSixTools`)

Interfacing with Existing Codes

The `jelli` export is currently being worked out in collaboration with Peter Stangl, but we are **happy** to build interfaces to **other codes**!

The vision for the ideal BSM workflow:

- Easily define a BSM model
- Compute one-loop matching to the SMEFT
- RG-evolve coefficients (either in-house or interfacing to other tools, i.e. `DSixTools`)
- Interface to pheno codes for parameter scans, collider studies, ...

Interfacing with Existing Codes

The `jelli` export is currently being worked out in collaboration with Peter Stangl, but we are **happy** to build interfaces to **other codes**!

The vision for the ideal BSM workflow:

- Easily define a BSM model
- Compute one-loop matching to the SMEFT
- RG-evolve coefficients (either in-house or interfacing to other tools, i.e. `DSixTools`)
- Interface to pheno codes for parameter scans, collider studies, ...

The individual steps are **automatable** enough to turn **months-long** pheno studies into **quick** afternoon sessions of coding (+ drinking coffee while waiting for the machine to finish computing results)

Renormalization and RG Evolution

Matching proceeds through evaluating the **hard region** of the effective one-loop action.

Matching proceeds through evaluating the **hard region** of the effective one-loop action.

This targets **short-distance contributions** for matching purposes, but can also be used to isolate UV-divergences!

Matching proceeds through evaluating the **hard region** of the effective one-loop action.

This targets **short-distance contributions** for matching purposes, but can also be used to isolate UV-divergences!

Recipe is simple:

- Expand loop integrals around small parameters, $\ell \gg m_i, p_i$

$$\int \frac{d^d\ell}{(2\pi)^d} \frac{1}{l^2} \frac{1}{(l-p)^2 - m^2} = \int \frac{d^d\ell}{(2\pi)^d} \frac{1}{l^4} + \int \frac{d^d\ell}{(2\pi)^d} \frac{2l \cdot p}{l^6} + \dots$$

Matching proceeds through evaluating the **hard region** of the effective one-loop action.

This targets **short-distance contributions** for matching purposes, but can also be used to isolate UV-divergences!

Recipe is simple:

- Expand loop integrals around small parameters, $\ell \gg m_i, p_i$

$$\int \frac{d^d\ell}{(2\pi)^d} \frac{1}{l^2} \frac{1}{(l-p)^2 - m^2} = \int \frac{d^d\ell}{(2\pi)^d} \frac{1}{l^4} + \int \frac{d^d\ell}{(2\pi)^d} \frac{2l \cdot p}{l^6} + \dots$$

- Introduce IR regulators to isolate UV poles & evaluate

$$\rightarrow \int \frac{d^d\ell}{(2\pi)^d} \frac{1}{(l^2 - \Lambda^2)^2} + \int \frac{d^d\ell}{(2\pi)^d} \frac{2l \cdot p}{(l^2 - \Lambda^2)^3} = \frac{i}{16\pi^2\epsilon} + \text{(finite)}$$

Matching proceeds through evaluating the **hard region** of the effective one-loop action.

This targets **short-distance contributions** for matching purposes, but can also be used to isolate UV-divergences!

Recipe is simple:

- Expand loop integrals around small parameters, $\ell \gg m_i, p_i$

$$\int \frac{d^d\ell}{(2\pi)^d} \frac{1}{l^2} \frac{1}{(l-p)^2 - m^2} = \int \frac{d^d\ell}{(2\pi)^d} \frac{1}{l^4} + \int \frac{d^d\ell}{(2\pi)^d} \frac{2l\cdot p}{l^6} + \ldots$$

- Introduce IR regulators to isolate UV poles & evaluate

$$\to \int \frac{d^d\ell}{(2\pi)^d} \frac{1}{(l^2-\Lambda^2)^2} + \int \frac{d^d\ell}{(2\pi)^d} \frac{2l\cdot p}{(l^2-\Lambda^2)^3} = \frac{i}{16\pi^2\epsilon} + (\text{finite})$$

- Last step amounts to simply replacing $\int d^d\ell/\ell^4 \to i/\epsilon$ and calling it a day.

Renormalization & RG Evolution

This functionality is coming to **Matchete** soon™!

Renormalization & RG Evolution

This functionality is coming to **Matchete** soon™!

Simple **first step** is to compute the `UVDivergentAction`:

In[6]:=
```
LSM = LoadModel["SM"];
GreensSimplify@UVDivergentAction@LSM // NiceForm
```

Out[7]//NiceForm=

$$-\frac{41}{24}\hbar\frac{1}{\epsilon}g_Y^2 B^{\mu\nu 2}+\frac{3}{2}\hbar\frac{1}{\epsilon}g_s^2 G^{\mu\nu A2}+\frac{5}{8}\hbar\frac{1}{\epsilon}g_L^2 W^{\mu\nu I2}+$$
$$\frac{1}{4}\hbar\frac{1}{\epsilon}\mu 2\left(9g_L^2+3g_Y^2-4\left(3\bar{Y}_d^{pr}Y_d^{pr}+\bar{Y}_e^{pr}Y_e^{pr}+3\bar{Y}_u^{pr}Y_u^{pr}+3\lambda\right)\right)\bar{H}_i H^i+$$
$$\hbar\frac{1}{\epsilon}\left(\frac{9}{16}g_L^4+\frac{3}{16}g_Y^4-3\bar{Y}_d^{pr}\bar{Y}_d^{st}Y_d^{pt}Y_d^{sr}-\bar{Y}_e^{pr}\bar{Y}_e^{st}Y_e^{pt}Y_e^{sr}-3\bar{Y}_u^{pr}\bar{Y}_u^{st}Y_u^{pt}Y_u^{sr}+\frac{3}{8}g_L^2\left(g_Y^2-6\lambda\right)-\frac{3}{4}\lambda g_Y^2+3\bar{Y}_d^{pr}Y_d^{pr}\lambda+\bar{Y}_e^{pr}Y_e^{pr}\lambda+3\bar{Y}_u^{pr}Y_u^{pr}\lambda+3\lambda^2\right)\bar{H}_i\bar{H}_j H^i H^j+$$
$$\frac{1}{24}\hbar\frac{1}{\epsilon}\left(18\left(\bar{Y}_d^{ps}\bar{Y}_d^{tr}Y_d^{ts}-\bar{Y}_d^{sr}\bar{Y}_u^{pt}Y_u^{st}\right)+\bar{Y}_d^{pr}\left(-27g_L^2-96g_s^2-5g_Y^2+36\bar{Y}_d^{st}Y_d^{st}+12\bar{Y}_e^{st}Y_e^{st}+36\bar{Y}_u^{st}Y_u^{st}\right)\right)\bar{H}_i\left(\bar{d}_a^r\cdot P_L\cdot q^{aip}\right)+$$

From this, the **counterterms** - and with that the **anomalous dimensions** - are easily* extracted.

Computing the **effective action** does **not** generally yield a **complete basis** - it knows nothing about operators with vanishing matching coefficients.

Renormalization & RG Evolution

Computing the **effective action** does **not** generally yield a **complete basis** - it knows nothing about operators with vanishing matching coefficients.

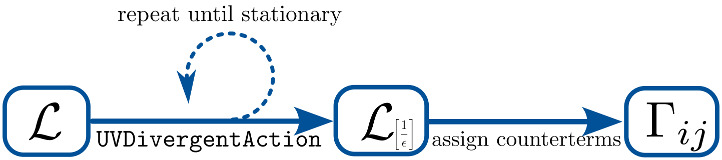
Thus, the `UVDivergentAction` can contain operators **not initially present** in the **input** Lagrangian.

Renormalization & RG Evolution

Computing the **effective action** does **not** generally yield a **complete basis** - it knows nothing about operators with vanishing matching coefficients.

Thus, the `UVDivergentAction` can contain operators **not initially present** in the **input** Lagrangian.

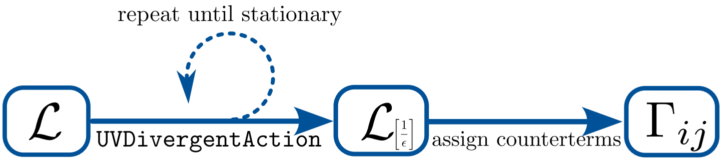
The calculation of anomalous dimensions then becomes an **iterative** procedure:



Computing the **effective action** does **not** generally yield a **complete basis** - it knows nothing about operators with vanishing matching coefficients.

Thus, the `UVDivergentAction` can contain operators **not initially present** in the **input** Lagrangian.

The calculation of anomalous dimensions then becomes an **iterative** procedure:



This procedure is **not** guaranteed to generate a complete basis, but it **will** produce all **operators** that **mix** with the ones in the starting $\mathcal{L}$ under renormalization.

Let's have a **peek** at what this will look like...

Defining a **toy model** with $\mathcal{L}_{\text{int}} = y\bar{\psi}_L\phi\chi_R + \text{h.c}$

and matching at **tree-level**:

```
DefineGaugeGroup[QED, U1, e, A]

DefineField[ψ, Fermion, Mass → 0, Charges → {QED[-1]}, Chiral → LeftHanded]
DefineField[χ, Fermion, Mass → 0, Charges → {QED[-1]}, Chiral → RightHanded]
DefineField[ϕ, Scalar, Mass → Heavy, SelfConjugate → True]
DefineCoupling[y, SelfConjugate → False]

LUV = FreeLag[] + PlusHc[y[] × ϕ[] Bar[ψ[]] ** χ[]];
LEFT = EOMSimplify@ Match[LUV, LoopOrder → 0];

NiceForm@ PlusHc[y[] × ϕ[] Bar[ψ[]] ** χ[]]
NiceForm@ SeriesEFT[LEFT, EFTOrder → {6}]
```

NiceForm=

$\bar{y}\,\phi\,(\bar{\chi}\cdot P_L\cdot\psi) + y\,\phi\,(\bar{\psi}\cdot P_R\cdot\chi)$

NiceForm=

$\frac{1}{2}\bar{y}^2\frac{1}{M\phi^2}(\bar{\chi}\cdot P_L\cdot\psi)^2 + \frac{1}{2}y^2\frac{1}{M\phi^2}(\bar{\psi}\cdot P_R\cdot\chi)^2 - \frac{1}{2}\bar{y}\,y\,\frac{1}{M\phi^2}(\bar{\chi}\cdot\gamma_\mu P_R\cdot\chi)(\bar{\psi}\cdot\gamma_\mu P_L\cdot\psi)$

Let's have a **peek** at what this will look like...

Defining a **toy model** with $\mathcal{L}_{\text{int}} = y\bar{\psi}_L \phi \chi_R + \text{h.c}$

and matching at **tree-level**:

```
DefineGaugeGroup[QED, U1, e, A]

DefineField[ψ, Fermion, Mass → 0, Charges → {QED[-1]}, Chiral → LeftHanded]
DefineField[χ, Fermion, Mass → 0, Charges → {QED[-1]}, Chiral → RightHanded]
DefineField[ϕ, Scalar, Mass → Heavy, SelfConjugate → True]
DefineCoupling[y, SelfConjugate → False]

LUV = FreeLag[] + PlusHc[y[] × ϕ[] Bar[ψ[]] ** χ[]];
LEFT = EOMSimplify@ Match[LUV, LoopOrder → 0];

NiceForm@ PlusHc[y[] × ϕ[] Bar[ψ[]] ** χ[]]
NiceForm@ SeriesEFT[LEFT, EFTOrder → {6}]
```

show only dimension-six terms

NiceForm=

$\bar{y}\,\phi\,(\bar{\chi}\cdot P_L\cdot\psi) + y\,\phi\,(\bar{\psi}\cdot P_R\cdot\chi)$

NiceForm=

$\frac{1}{2}\bar{y}^2\frac{1}{M\phi^2}(\bar{\chi}\cdot P_L\cdot\psi)^2 + \frac{1}{2}y^2\frac{1}{M\phi^2}(\bar{\psi}\cdot P_R\cdot\chi)^2 - \frac{1}{2}\bar{y}\,y\frac{1}{M\phi^2}(\bar{\chi}\cdot\gamma_\mu P_R\cdot\chi)(\bar{\psi}\cdot\gamma_\mu P_L\cdot\psi)$

Early Preview of RG Calculations

Let's have a **peek** at what this will look like...

Defining a **toy model** with $\mathcal{L}_{\text{int}} = y\bar{\psi}_L \phi \chi_R + \text{h.c}$

and matching at **tree-level**:

```
DefineGaugeGroup[QED, U1, e, A]

DefineField[ψ, Fermion, Mass → 0, Charges → {QED[-1]}, Chiral → LeftHanded]
DefineField[χ, Fermion, Mass → 0, Charges → {QED[-1]}, Chiral → RightHanded]
DefineField[ϕ, Scalar, Mass → Heavy, SelfConjugate → True]
DefineCoupling[y, SelfConjugate → False]

LUV = FreeLag[] + PlusHc[y[] × ϕ[] Bar[ψ[]] ** χ[]];
LEFT = EOMSimplify@ Match[LUV, LoopOrder → 0];

NiceForm@ PlusHc[y[] × ϕ[] Bar[ψ[]] ** χ[]]
NiceForm@ SeriesEFT[LEFT, EFTOrder → {6}]
```

← show only dimension-six terms

NiceForm=

$\bar{y}\,\phi\,(\bar{\chi}\cdot P_L\cdot\psi) + y\,\phi\,(\bar{\psi}\cdot P_R\cdot\chi)$

NiceForm=

$\frac{1}{2}\bar{y}^2\frac{1}{M\phi^2}(\bar{\chi}\cdot P_L\cdot\psi)^2 + \frac{1}{2}y^2\frac{1}{M\phi^2}(\bar{\psi}\cdot P_R\cdot\chi)^2 - \frac{1}{2}\bar{y}\,y\,\frac{1}{M\phi^2}(\bar{\chi}\cdot\gamma_\mu P_R\cdot\chi)\,(\bar{\psi}\cdot\gamma_\mu P_L\cdot\psi)$

← $\mathcal{L}_{\text{int}}$ contains $\psi^2\chi^2$ operators

Early Preview of RG Calculations

Let's have a **peek** at what this will look like...

Defining a **toy model** with $\mathcal{L}_{\text{int}} = y\bar{\psi}_L\phi\chi_R + \text{h.c}$

and matching at **tree-level**:

```
DefineGaugeGroup[QED, U1, e, A]

DefineField[ψ, Fermion, Mass → 0, Charges → {QED[-1]}, Chiral → LeftHanded]
DefineField[χ, Fermion, Mass → 0, Charges → {QED[-1]}, Chiral → RightHanded]
DefineField[ϕ, Scalar, Mass → Heavy, SelfConjugate → True]
DefineCoupling[y, SelfConjugate → False]

LUV = FreeLag[] + PlusHc[y[] × ϕ[] Bar[ψ[]] ** χ[]];
LEFT = EOMSimplify@ Match[LUV, LoopOrder → 0];

NiceForm@ PlusHc[y[] × ϕ[] Bar[ψ[]] ** χ[]]
NiceForm@ SeriesEFT[LEFT, EFTOrder → {6}]
```

← show only dimension-six terms

NiceForm=

$\bar{y}\,\phi\,(\bar{\chi}\cdot P_L\cdot\psi) + y\,\phi\,(\bar{\psi}\cdot P_R\cdot\chi)$

NiceForm=

$\frac{1}{2}\bar{y}^2\frac{1}{M\phi^2}(\bar{\chi}\cdot P_L\cdot\psi)^2 + \frac{1}{2}y^2\frac{1}{M\phi^2}(\bar{\psi}\cdot P_R\cdot\chi)^2 - \frac{1}{2}\bar{y}\,y\frac{1}{M\phi^2}(\bar{\chi}\cdot\gamma_\mu P_R\cdot\chi)(\bar{\psi}\cdot\gamma_\mu P_L\cdot\psi)$

← $\mathcal{L}_{\text{int}}$ contains $\psi^2\chi^2$ operators

Note how this is in no way a **complete basis**.

Early Preview of RG Calculations

The RG module does a **second pass** because it needs ψ^4 and χ^4 counterterms - operators that were **not present** in the **input** Lagrangian:

```
PrintBetafunctions@LEFT
```

» New operators found, rerunning... Plus[...]

Lagrangian:

$$-\frac{1}{4}A^{\mu\nu 2} + i\,(\bar{\chi}\cdot\gamma_\mu P_R\cdot D_\mu\chi) + i\,(\bar{\psi}\cdot\gamma_\mu P_L\cdot D_\mu\psi) + C_{\psi^2\chi^2}\,(\bar{\chi}\cdot P_L\cdot\psi)^2 + C_{\chi^4}\,(\bar{\chi}\cdot\gamma_\mu P_R\cdot\chi)^2 + C_{\psi^2\chi^2}\,(\bar{\psi}\cdot P_R\cdot\chi)^2 + C_{\chi^2\psi^2}\,(\bar{\chi}\cdot\gamma_\mu P_R\cdot\chi)\,(\bar{\psi}\cdot\gamma_\mu P_L\cdot\psi) + C_{\psi^4}\,(\bar{\psi}\cdot\gamma_\mu P_L\cdot\psi)^2$$

NiceForm=

$$\mu\frac{dC_{\chi^4}}{d\mu} \to \frac{1}{12}\frac{1}{\pi^2}C_{\chi^2\psi^2}e^2 + \frac{13}{12}\frac{1}{\pi^2}C_{\chi^4}e^2$$

$$\mu\frac{dC_{\psi^2\chi^2}}{d\mu} \to \frac{3}{4}\frac{1}{\pi^2}C_{\psi^2\chi^2}e^2$$

$$\mu\frac{dC_{\psi^4}}{d\mu} \to \frac{1}{12}\frac{1}{\pi^2}C_{\chi^2\psi^2}e^2 + \frac{13}{12}\frac{1}{\pi^2}C_{\psi^4}e^2$$

$$\mu\frac{dC_{\chi^2\psi^2}}{d\mu} \to -\frac{7}{12}\frac{1}{\pi^2}C_{\chi^2\psi^2}e^2 + \frac{1}{3}\frac{1}{\pi^2}C_{\chi^4}e^2 + \frac{1}{3}\frac{1}{\pi^2}C_{\psi^4}e^2$$

The RG module does a **second pass** because it needs ψ^4 and χ^4 counterterms - operators that were **not present** in the **input** Lagrangian:

```
PrintBetafunctions@LEFT
```

» New operators found, rerunning... Plus[...] ← new operators found, starting a second run

Lagrangian:

$$-\frac{1}{4} A^{\mu\nu 2} + i\left(\bar{\chi}\cdot\gamma_\mu P_R\cdot D_\mu\chi\right) + i\left(\bar{\psi}\cdot\gamma_\mu P_L\cdot D_\mu\psi\right) + C_{\psi^2\chi^2}\left(\bar{\chi}\cdot P_L\cdot\psi\right)^2 + C_{\chi^4}\left(\bar{\chi}\cdot\gamma_\mu P_R\cdot\chi\right)^2 + C_{\psi^2\chi^2}\left(\bar{\psi}\cdot P_R\cdot\chi\right)^2 + C_{\chi^2\psi^2}\left(\bar{\chi}\cdot\gamma_\mu P_R\cdot\chi\right)\left(\bar{\psi}\cdot\gamma_\mu P_L\cdot\psi\right) + C_{\psi^4}\left(\bar{\psi}\cdot\gamma_\mu P_L\cdot\psi\right)^2$$

NiceForm=

$$\mu\frac{dC_{\chi^4}}{d\mu} \to \frac{1}{12}\frac{1}{\pi^2}C_{\chi^2\psi^2}e^2 + \frac{13}{12}\frac{1}{\pi^2}C_{\chi^4}e^2$$

$$\mu\frac{dC_{\psi^2\chi^2}}{d\mu} \to \frac{3}{4}\frac{1}{\pi^2}C_{\psi^2\chi^2}e^2$$

$$\mu\frac{dC_{\psi^4}}{d\mu} \to \frac{1}{12}\frac{1}{\pi^2}C_{\chi^2\psi^2}e^2 + \frac{13}{12}\frac{1}{\pi^2}C_{\psi^4}e^2$$

$$\mu\frac{dC_{\chi^2\psi^2}}{d\mu} \to -\frac{7}{12}\frac{1}{\pi^2}C_{\chi^2\psi^2}e^2 + \frac{1}{3}\frac{1}{\pi^2}C_{\chi^4}e^2 + \frac{1}{3}\frac{1}{\pi^2}C_{\psi^4}e^2$$

Early Preview of RG Calculations

The RG module does a **second pass** because it needs ψ^4 and χ^4 counterterms - operators that were **not present** in the **input** Lagrangian:

```
PrintBetafunctions@LEFT
```

» New operators found, rerunning... Plus[...]

Lagrangian:

$$-\frac{1}{4} A^{\mu\nu 2} + i \left(\bar{\chi} \cdot \gamma_\mu P_R \cdot D_\mu \chi\right) + i \left(\bar{\psi} \cdot \gamma_\mu P_L \cdot D_\mu \psi\right) + C_{\psi^2\chi^2} (\bar{\chi} \cdot P_L \cdot \psi)^2 + C_{\chi^4} (\bar{\chi} \cdot \gamma_\mu P_R \cdot \chi)^2 + C_{\psi^2\chi^2} (\bar{\psi} \cdot P_R \cdot \chi)^2 + C_{\chi^2\psi^2} (\bar{\chi} \cdot \gamma_\mu P_R \cdot \chi) (\bar{\psi} \cdot \gamma_\mu P_L \cdot \psi) + C_{\psi^4} (\bar{\psi} \cdot \gamma_\mu P_L \cdot \psi)^2$$

couplings renamed
new terms introduced

NiceForm=

$$\mu \frac{dC_{\chi^4}}{d\mu} \to \frac{1}{12} \frac{1}{\pi^2} C_{\chi^2\psi^2} e^2 + \frac{13}{12} \frac{1}{\pi^2} C_{\chi^4} e^2$$

$$\mu \frac{dC_{\psi^2\chi^2}}{d\mu} \to \frac{3}{4} \frac{1}{\pi^2} C_{\psi^2\chi^2} e^2$$

$$\mu \frac{dC_{\psi^4}}{d\mu} \to \frac{1}{12} \frac{1}{\pi^2} C_{\chi^2\psi^2} e^2 + \frac{13}{12} \frac{1}{\pi^2} C_{\psi^4} e^2$$

$$\mu \frac{dC_{\chi^2\psi^2}}{d\mu} \to -\frac{7}{12} \frac{1}{\pi^2} C_{\chi^2\psi^2} e^2 + \frac{1}{3} \frac{1}{\pi^2} C_{\chi^4} e^2 + \frac{1}{3} \frac{1}{\pi^2} C_{\psi^4} e^2$$

The RG module does a **second pass** because it needs ψ^4 and χ^4 counterterms - operators that were **not present** in the **input** Lagrangian:

```
PrintBetafunctions@LEFT
```

» New operators found, rerunning... Plus[...]

Lagrangian:

$$-\frac{1}{4} A^{\mu\nu 2} + i\,(\bar{\chi}\cdot\gamma_\mu P_R\cdot D_\mu\chi) + i\,(\bar{\psi}\cdot\gamma_\mu P_L\cdot D_\mu\psi) + C_{\psi^2\chi^2}\,(\bar{\chi}\cdot P_L\cdot\psi)^2 + C_{\chi^4}\,(\bar{\chi}\cdot\gamma_\mu P_R\cdot\chi)^2 + C_{\psi^2\chi^2}\,(\bar{\psi}\cdot P_R\cdot\chi)^2 + C_{\chi^2\psi^2}\,(\bar{\chi}\cdot\gamma_\mu P_R\cdot\chi)\,(\bar{\psi}\cdot\gamma_\mu P_L\cdot\psi) + C_{\psi^4}\,(\bar{\psi}\cdot\gamma_\mu P_L\cdot\psi)^2$$

NiceForm=

$$\mu\frac{dC_{\chi^4}}{d\mu} \to \frac{1}{12}\frac{1}{\pi^2}C_{\chi^2\psi^2}e^2 + \frac{13}{12}\frac{1}{\pi^2}C_{\chi^4}e^2$$

$$\mu\frac{dC_{\psi^2\chi^2}}{d\mu} \to \frac{3}{4}\frac{1}{\pi^2}C_{\psi^2\chi^2}e^2$$

$$\mu\frac{dC_{\psi^4}}{d\mu} \to \frac{1}{12}\frac{1}{\pi^2}C_{\chi^2\psi^2}e^2 + \frac{13}{12}\frac{1}{\pi^2}C_{\psi^4}e^2$$

$$\mu\frac{dC_{\chi^2\psi^2}}{d\mu} \to -\frac{7}{12}\frac{1}{\pi^2}C_{\chi^2\psi^2}e^2 + \frac{1}{3}\frac{1}{\pi^2}C_{\chi^4}e^2 + \frac{1}{3}\frac{1}{\pi^2}C_{\psi^4}e^2$$

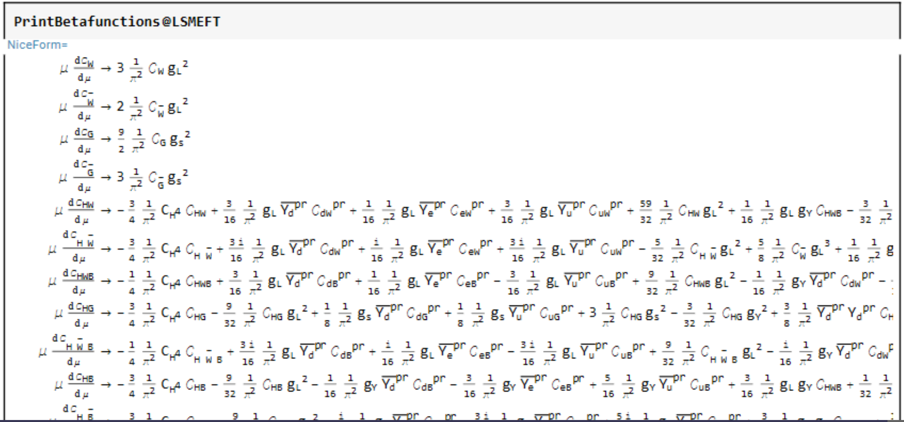
This is **very** work-in-progress, and **subject to lots of change**!

RG Calculations with Matchete

Once fully in place, these functions will allow the user to **quickly** perform the **matching & running** of BSM models from within Mathematica at the **push of a button**.

Once fully in place, these functions will allow the user to **quickly** perform the **matching & running** of BSM models from within Mathematica at the **push of a button**.

Especially **valuable** in cases where the target basis is **not just SMEFT**, i.e. SMEFT + light particles, **multi-scale** BSM models, ...



Documentation

Documentation

Matchete is **growing fast**.

Documentation

Matchete is **growing fast**.

By design, **core functionality** is achieved with very **few input commands**, and kept as **intuitive** as possible.

Documentation

Matchete is **growing fast**.

By design, **core functionality** is achieved with very **few input commands**, and kept as **intuitive** as possible.

However, want to keep everything **flexible** $\rightarrow$ functions can be tweaked with options.

Documentation

Matchete is **growing fast**.

By design, **core functionality** is achieved with very **few input commands**, and kept as **intuitive** as possible.

However, want to keep everything **flexible** $\rightarrow$ functions can be tweaked with options.

Some **leftover rigidity** for both physics & software reasons
(i.e. `EOMSimplify` requires input $\mathcal{L}$ to contain consistent kinetic terms, ...)

$\rightarrow$ Desirable to keep information and **documentation** easily **accessible**!

Documentation

Matchete is **growing fast**.

By design, **core functionality** is achieved with very **few input commands**, and kept as **intuitive** as possible.

However, want to keep everything **flexible** $\rightarrow$ functions can be tweaked with options.

Some **leftover rigidity** for both physics & software reasons
    (i.e. `EOMSimplify` requires input $\mathcal{L}$ to contain consistent kinetic terms, ...)

$\rightarrow$ Desirable to keep information and **documentation** easily **accessible**!

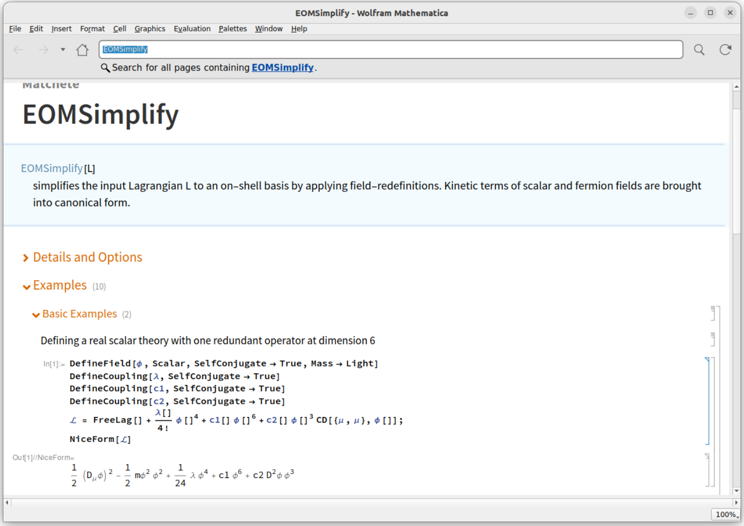
Achieved in `v0.2.0` through "usage strings" for all exported tokens:

```
?EOMSimplify
```

Symbol

EOMSimplify[L] takes a Lagrangian and removes redundant operators by the means of field redefinitions. The option 'EFTOrder' allows to define a maximal order in EFT counting. The standard value (Automatic) determines it from the input Lagrangian. The option 'DummyCoefficients' internally replaces matching coefficients with symbols, potentially improving performance when simplifying very large Lagrangians (standard is False).

For more details: use Mathematica’s **built-in documentation** system.



Just press “**F1**” on a command to bring up detailed **documentation**!

Conclusions

Conclusions

- Matchete keeps **growing** and **improving** to become **faster** & **more powerful**!

Conclusions

- Matchete keeps **growing** and **improving** to become **faster** & **more powerful**!
- Community **feedback** has been & continues to be **invaluable** help for bringing issues to our attention - **keep it coming**!

Conclusions

- Matchete keeps **growing** and **improving** to become **faster** & **more powerful**!
- Community **feedback** has been & continues to be **invaluable** help for bringing issues to our attention - **keep it coming**!
- Current version `v0.2.0` brings a number of **improvements** in both **stability** & **performance**

Conclusions

- Matchete keeps **growing** and **improving** to become **faster** & **more powerful**!
- Community **feedback** has been & continues to be **invaluable** help for bringing issues to our attention - **keep it coming**!
- Current version `v0.2.0` brings a number of **improvements** in both **stability** & **performance**
- Many **new features** are on their way: Automatic **Evanescence**, `smelli` **Interface**, **Renormalization**, ...

Conclusions

- Matchete keeps **growing** and **improving** to become **faster** & **more powerful**!
- Community **feedback** has been & continues to be **invaluable** help for bringing issues to our attention - **keep it coming**!
- Current version `v0.2.0` brings a number of **improvements** in both **stability** & **performance**
- Many **new features** are on their way: Automatic **Evanescence**, `smelli` **Interface**, **Renormalization**, ...
- Aim to make Matchete a **versatile component** in an interconnected chain of SMEFT Tools!

Conclusions

- Matchete keeps **growing** and **improving** to become **faster** & **more powerful**!
- Community **feedback** has been & continues to be **invaluable** help for bringing issues to our attention - **keep it coming**!
- Current version `v0.2.0` brings a number of **improvements** in both **stability** & **performance**
- Many **new features** are on their way: Automatic **Evanescence**, `smelli` **Interface**, **Renormalization**, ...
- Aim to make Matchete a **versatile component** in an interconnected chain of SMEFT Tools!

Thanks for listening!

https://gitlab.com/matchete/matchete