UFO@NLO:
Automating NLO For BSM

Hua-Sheng Shao

LPTHE
LABORATOIRE DE PHYSIQUE
THEORIQUE ET HAUTES ENERGIES

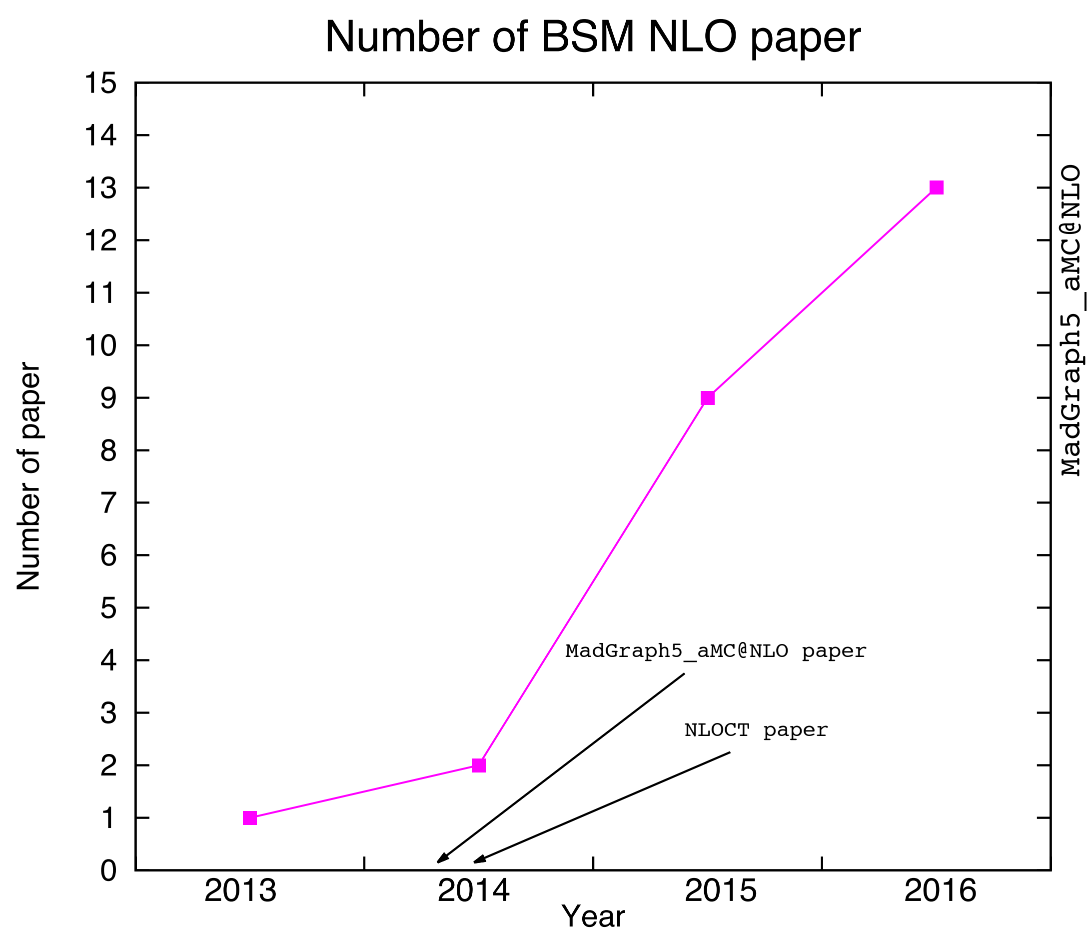
Mini-workshop on automating Higgs and BSM calculations
Jussieu, Paris
10 October 2017

BSM (NLO) PROCESS OVERVIEW IN MG5AMC

- Colored particle production
  - Colored scalar pair production — Degrande, Fuks, Hirschi, Proudom, HSS (PRD'15)
  - Supersymmetric QCD — Degrande, Fuks, Hirschi, Proudom, HSS (PLB'16)
  - Vector-like quark pair production — Les Houches 2015 (1605.02684) ; Fuks, HSS (1610.04622)
  - MSSM — Degrande, Fuks, Goncalves-Netto, Hirschi, Lopez-Val, Mawatari, Pagani, Proudom, HSS, Zaro (in preparation)
- BSM Higgs production
  - Higgs characterisation model — Artoisenet et al. (JHEP'13); Maltoni, Mawatari, Zaro (EPJC'14); Demartin, Maltoni, Mawatari, Page, Zaro (EPJC'14); Demartin, Maltoni, Mawatari, Zaro (EPJC'15)
  - Two-Higgs-Doublet Model — Degrande (CPC'15); Degrande, Ubiali, Wiesemann, Zaro (JHEP'15); Degrande, Frederix, Hirschi, Ubiali, Wiesemann, Zaro (1607.05291)
  - Georgi-Machacek model — Degrande, Hartling, Logan, Peterson, Zaro (PRD'16)
- Spin-2 particle production — Das, Degrande, Hirschi, Maltoni, HSS (1605.09359)
- Dark matter collider production
  - s-channel mediator
    - spin 0 or 1 mediator — Mattelaer Vryonidou (EPJC'15); Backovic, Kramer, Maltoni, Martini, Mawatari, Pellen (EPJC'15); Neubert, Wang, Zhang (JHEP'16); Arina et al. (JHEP'16)
    - spin 2 mediator — Das, Degrande, Hirschi, Maltoni, Mawatari, HSS et al. (in preparation)
  - t-channel mediator — Fuks, Hirschi, Mattelaer et al. (in preparation)

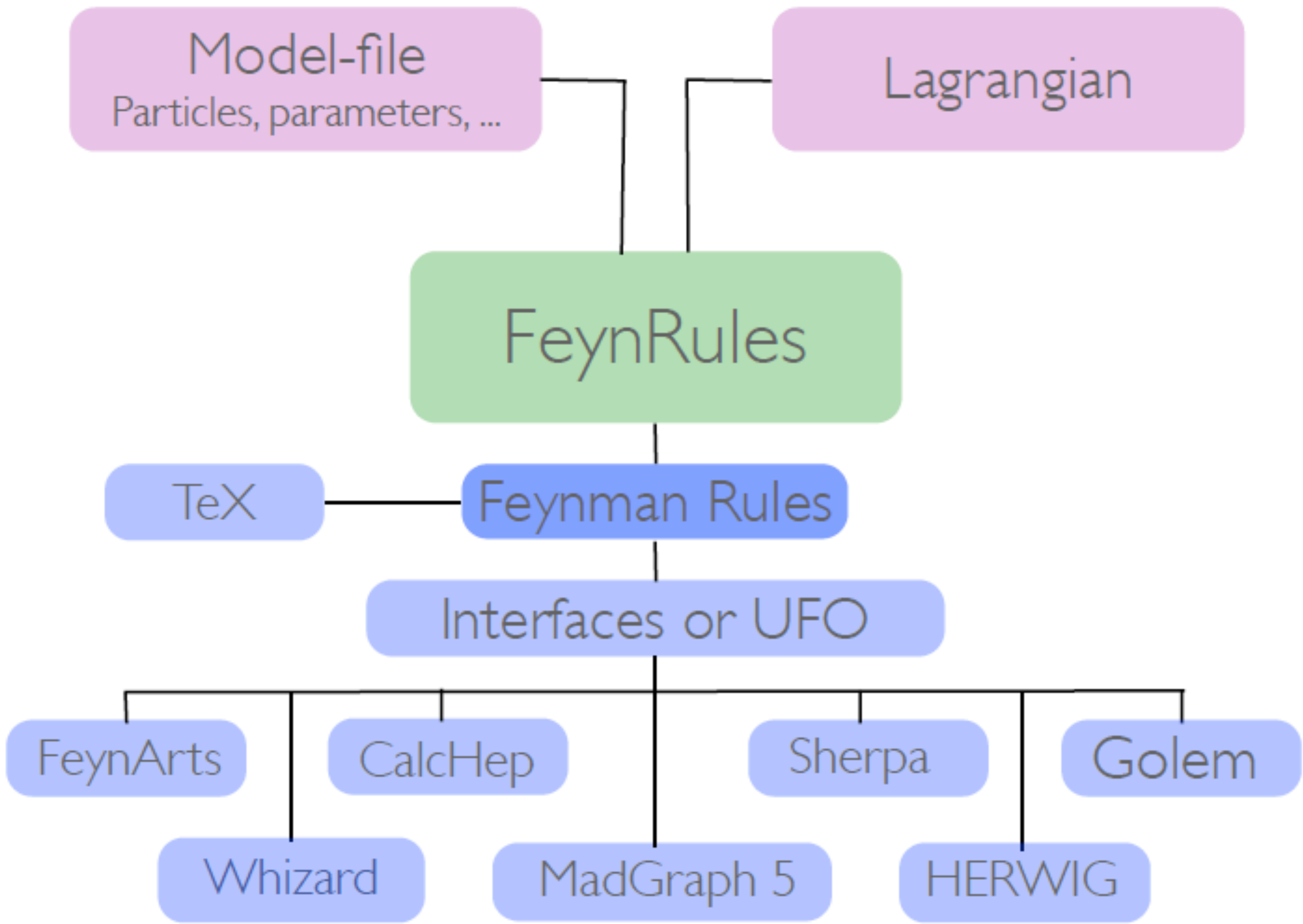
BSM (NLO) PROCESS OVERVIEW IN MG5AMC

- SM effective field theory
  - Top FCNC processes — Degrande, Maltoni, Wang, Zhang (PRD'15); Durieux, Maltoni, Zhang (PRD'15)
  - ttbarZ/gamma/Higgs production — Bylund, Maltoni, Tsinikos, Vryonidou, Zhang (JHEP'16); Maltoni, Vryonidou, Cen (JHEP'16)
  - Single-top production — Zhang (PRL'16)
  - Top pair production via chromomagnetic dipole momenta — Franzosi, Zhang (PRD'15)
  - Higgs production with dimension-6 operators — Degrande, Fuks, Mawatari, Mimasu, Sanz (1609.04833)
- Other colorless particle production
  - Heavy neutrino production — Degrande, Mattelaer, Ruiz, Tumer (PRD'16)



FEYNRULES: STRUCTURE

Christensen, Duhr (CPC'09); Alloul, Christensen, Duhr, Degrande, Fuks (CPC'14)



FEYNRULES: INPUT

Christensen, Duhr (CPC'09); Alloul, Christensen, Duhr, Degrande, Fuks (CPC'14)

FEYNRULES: INPUT

Christensen, Duhr (CPC'09); Alloul, Christensen, Duhr, Degrande, Fuks (CPC'14)

- Definitions of particles

```
(* ************************** *)
(* **** Particle classes **** *)
(* ************************** *)
M$ClassesDescription = {
              . . .
   V[4] == {
     ClassName        -> G,
     SelfConjugate    -> True,
     Indices          -> {Index[Gluon]},
     Mass             -> 0,
     Width            -> 0,
     ParticleName     -> "g",
     PDG              -> 21,
     PropagatorLabel  -> "G",
     PropagatorType   -> C,
     PropagatorArrow  -> None,
     FullName         -> "G"
   },
              . . .
 };
```

FEYNRULES: INPUT

Christensen, Duhr (CPC'09); Alloul, Christensen, Duhr, Degrande, Fuks (CPC'14)

- Definitions of particles

```
(* ************************** *)
(* **** Particle classes **** *)
(* ************************** *)
M$ClassesDescription = {
                ...
   V[4] == {
     ClassName        -> G,
     SelfConjugate    -> True,
     Indices          -> {Index[Gluon]},
     Mass             -> 0,
     Width            -> 0,
     ParticleName     -> "g",
     PDG              -> 21,
     PropagatorLabel  -> "G",
     PropagatorType   -> C,
     PropagatorArrow  -> None,
     FullName         -> "G"
   },
                ...
 };
```

- Definitions of parameters

```
(* ************************** *)
(* *****   Parameters   ***** *)
(* ************************** *)
M$Parameters = {
                ...
      aS    == {
        ParameterType    -> External,
        BlockName        -> SMINPUTS,
        OrderBlock       -> 3,
        Value            -> 0.1184,
        InteractionOrder -> {QCD,2},
        TeX              -> Subscript[\[Alpha],s],
        Description      -> "Strong coupling constant at the Z p
      },
                ...
};
```

FEYNRULES: INPUT

Christensen, Duhr (CPC'09); Alloul, Christensen, Duhr, Degrande, Fuks (CPC'14)

- Definitions of particles

```
(* ************************** *)
(* **** Particle classes **** *)
(* ************************** *)
M$ClassesDescription = {
                ...
   V[4] == {
     ClassName         -> G,
     SelfConjugate     -> True,
     Indices           -> {Index[Gluon]},
     Mass              -> 0,
     Width             -> 0,
     ParticleName      -> "g",
     PDG               -> 21,
     PropagatorLabel   -> "G",
     PropagatorType    -> C,
     PropagatorArrow   -> None,
     FullName          -> "G"
   },
                ...
};
```

- Definitions of parameters

```
(* ************************** *)
(* *****    Parameters   ***** *)
(* ************************** *)
M$Parameters = {
                ...
     aS     == {
       ParameterType    -> External,
       BlockName        -> SMINPUTS,
       OrderBlock       -> 3,
       Value            -> 0.1184,
       InteractionOrder -> {QCD,2},
       TeX              -> Subscript[\[Alpha],s],
       Description      -> "Strong coupling constant at the Z p
     },
                ...
};
```

- Definitions of gauge groups

```
(* ************************** *)
(* *****  Gauge groups  ***** *)
(* ************************** *)
M$GaugeGroups = {
          SU3C == {
            Abelian           -> False,
            CouplingConstant  -> gs,
            GaugeBoson        -> G,
            StructureConstant -> f,
            Representations   -> {T,Colour},
            SymmetricTensor   -> dSUN
          }
        };
```

FEYNRULES: INPUT

Christensen, Duhr (CPC'09); Alloul, Christensen, Duhr, Degrande, Fuks (CPC'14)

- Definitions of particles

```
(* ************************** *)
(* **** Particle classes **** *)
(* ************************** *)
M$ClassesDescription = {
            ...
    V[4] == {
      ClassName        -> G,
      SelfConjugate    -> True,
      Indices          -> {Index[Gluon]},
      Mass             -> 0,
      Width            -> 0,
      ParticleName     -> "g",
      PDG              -> 21,
      PropagatorLabel  -> "G",
      PropagatorType   -> C,
      PropagatorArrow  -> None,
      FullName         -> "G"
    },
            ...
};
```

- Definitions of gauge groups

```
(* ************************** *)
(* *****  Gauge groups  ***** *)
(* ************************** *)
M$GaugeGroups = {
        SU3C == {
          Abelian           -> False,
          CouplingConstant  -> gs,
          GaugeBoson        -> G,
          StructureConstant -> f,
          Representations   -> {T,Colour},
          SymmetricTensor   -> dSUN
        }
      };
```

- Definitions of parameters

```
(* ************************** *)
(* *****   Parameters   ***** *)
(* ************************** *)
M$Parameters = {
            ...
    aS     == {
      ParameterType    -> External,
      BlockName        -> SMINPUTS,
      OrderBlock       -> 3,
      Value            -> 0.1184,
      InteractionOrder -> {QCD,2},
      TeX              -> Subscript[\[Alpha],s],
      Description      -> "Strong coupling constant at the Z p
    },
            ...
};
```

- The Lagrangian

$$\mathcal{L}_{\mathrm{QCD}} = -\frac{1}{4} G^a_{\mu\nu} G^{a,\mu\nu} + i\bar{q}\gamma^\mu D_\mu q - M_q \bar{q} q$$

```
L =
-1/4 FS[G,mu,nu,a] FS[G,mu,nu,a]
+ I qbar.Ga[mu].DC[q,mu]
- MQ qbar.q
```

FEYNRULES: DERIVATION

Christensen, Duhr (CPC'09); Alloul, Christensen, Duhr, Degrande, Fuks (CPC'14)

- Above information can be incorporated in .fr files and load the file via

```
LoadModel[ < file.fr >, < file2.fr >, ... ]
```

- Extracting the Feynman rules

FeynmanRules[L]

- Output the model files for various generators

WriteCHOutput[L] ⟶ CalcHep

WriteFeynArtsOutput[L] ⟶ FeynArts

WriteSHOutput[L] ⟶ Sherpa

WriteWOOutput[L] ⟶ Whizard

WriteUFO[L] ⟵ { MadGraph5_aMC@NLO, Sherpa, GoSam }

FEYNRULES: DERIVATION

Christensen, Duhr (CPC'09); Alloul, Christensen, Duhr, Degrande, Fuks (CPC'14)

- Above information can be incorporated in .fr files and load the file via

```
LoadModel[ < file.fr >, < file2.fr >, ... ]
```

- Extracting the Feynman rules

```
FeynmanRules[ L ]
```

- Output the model files for various generators

WriteCHOutput[L] ⟶ CalcHep

WriteFeynArtsOutput[L] ⟶ FeynArts

WriteSHOutput[L] ⟶ Sherpa

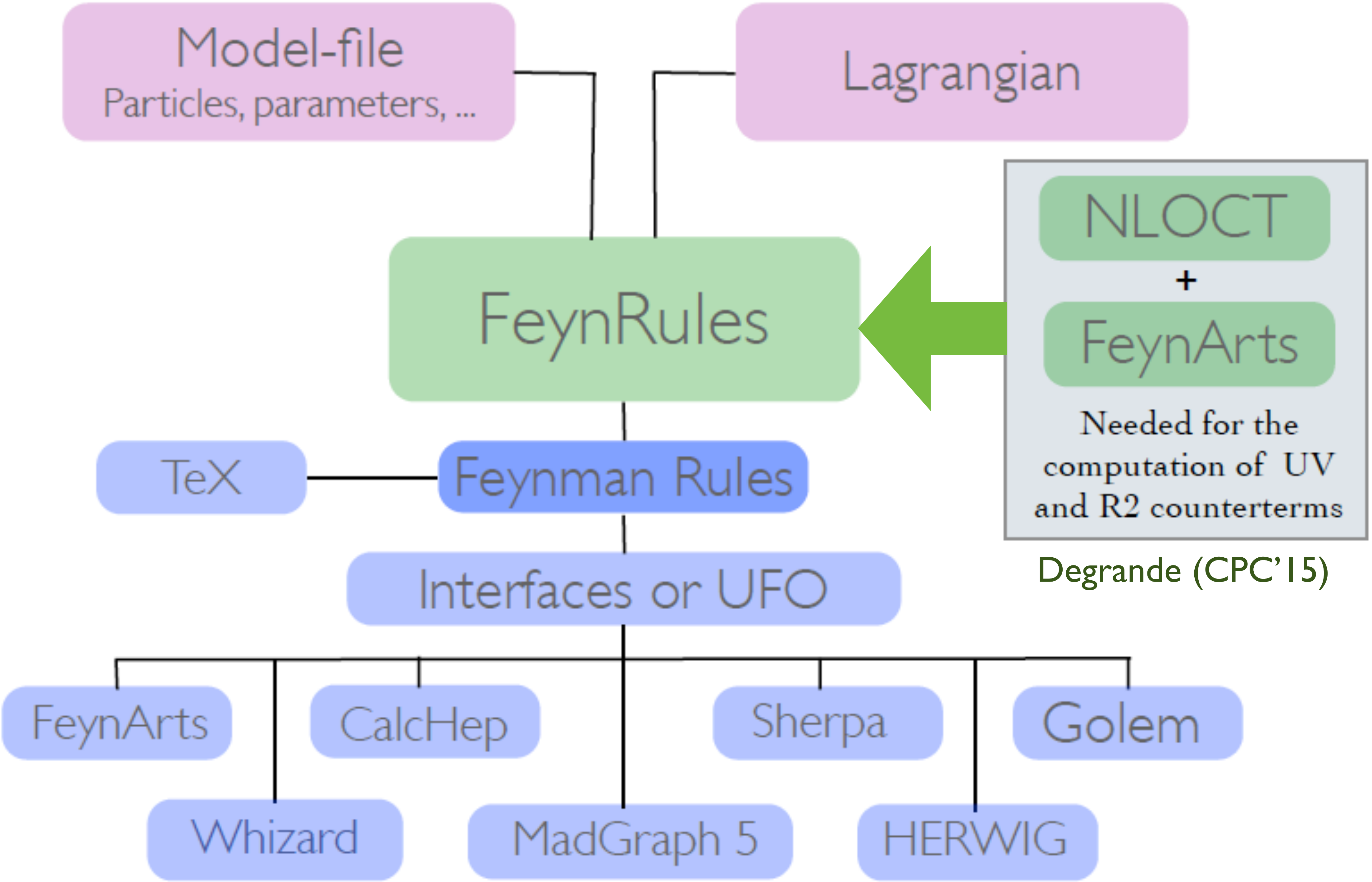
WriteWOOutput[L] ⟶ Whizard

WriteUFO[L] ⟶ { MadGraph5_aMC@NLO, Sherpa, GoSam }

NEW Standard !

FEYNRULES: STRUCTURE

Christensen, Duhr (CPC'09); Alloul, Christensen, Duhr, Degrande, Fuks (CPC'14); Degrande (CPC'15)



Degrande (CPC'15)

FEYNRULES: DERIVATION@NLO I

Degrande (CPC'15)

Load FeynRules and Model file

```
$FeynRulesPath = SetDirectory["~/feynrules/trunk/feynrules-development"]; << FeynRules`
LoadModel["~/feynrules/trunk/feynrules-development/Models/SM/SM.fr",
  "~/Dropbox/aMCatNLO-MSSM/FeynRulesInput/coloredscalars.fr"];
```

Load restrict card

```
LoadRestriction["~/feynrules/trunk/feynrules-development/Models/SM/Masslessbuttb.rst"]
LoadRestriction["~/feynrules/trunk/feynrules-development/Models/SM/DiagonalCKM.rst"]
```

Derive Feynman rule (optional)

```
FeynmanRules[LSM + LagNP]
```

Get renormalization Lagrangian

```
Lren = OnShellRenormalization[LSM + LagNP (*ExpandIndices[LSM,FlavorExpand→True]/.
     {u[__]→0,c[__]→0,d[__]→0,s[__]→0,ve[__]→0,vm[__]→0,e[__]→0,mu[__]→0}*), QCDOnly → True, FlavorMixing → False]; //
 Timing
```

Generate NLO FeynArts Model

```
SetDirectory["~/FeynArts-3.8/Models"];
WriteFeynArtsOutput[Lren, Output → "colscalarsnomixgs"]
```

FEYNRULES: DERIVATION@NLO II

Degrande (CPC'15)

Quit kernel

```
Quit[]
```

Load FeynArts and NLOCT

```
SetDirectory["~/FeynArts-3.8"];
<< FeynArts`
SetDirectory["~/feynrules/trunk/feynrules-development"];
<< NLOCT`
SetDirectory["~/feynrules/trunk/R2"];
```

Compute UV and R2 CT

```
WriteCT["colscalarsnomix/colscalarsnomix", "colscalarsnomix/colscalarsnomix", Output -> "colscalartest",
  LabelInternal → True, QCDOnly → True, CTparameters → False, KeptIndices → {}, ZeroMom → {{aS, {F[7], V[4], -F[7]}, C
  Assumptions -> {MT > 0},
  GenericVertexList → {{F, F}, {S, S}, {V, V}, {F, F, V}, {F, F, S}, {F, F, V, V}(*,{V,V,S},{V,S,S},{S,S,S},{V,V,V},
    {V,V,V,V},{V,V,S,S}*)}, MaxDim → 5, EvenOnly → False, IsFeynmanGauge → False] // Timing
```

FEYNRULES: DERIVATION@NLO III

Quit kernel

```
Quit[]
```

Load FeynRules and Model file again

```
$FeynRulesPath = SetDirectory["~/feynrules/trunk/feynrules-development"]; << FeynRules`
LoadModel["~/feynrules/trunk/feynrules-development/Models/SM/SM.fr",
  "~/Dropbox/aMCatNLO-MSSM/FeynRulesInput/coloredscalars.fr"];
```

Load restrict card again

```
LoadRestriction["~/feynrules/trunk/feynrules-development/Models/SM/DiagonalCKM.rst"]
LoadRestriction["~/feynrules/trunk/feynrules-development/Models/SM/Masslessbuttb.rst"]
```

Get the UV and R2 computed by NLOCT

```
SetDirectory["~/feynrules/trunk/R2"];
Get["colscalartest.nlo"];
```

Write out NLO UFO model

```
FR$Loop = False;
SetDirectory["~/2.2.0/models"];
WriteUFO[LSM + LagNP, UVCounterterms → (UV$vertlist /. FR$IR → 1), R2Vertices → R2$vertlist, Output → "ColScalarFull",
 Debug → False]
```

UFO: UNIVERSAL FEYNRULES OUTPUT

Degrande, Duhr, Fuks, Grellscheid, Mattelaer, Reiter (CPC'12)

- The UFO is a set of PYTHON files
  - Particle information (particles.py)
  - Interaction information (vertices.py, couplings.py, lorentz.py, couplings_orders.py)
  - Parameter information (parameters.py)
  - Propagator information (propagators.py)
  - Tools (function_library.py, object_library.py, write_param_card.py, decays.py)
  - NLO counterterms (CT_couplings.py, CT_parameters.py, CT_vertices.py)

For example: SUSY QCD

```
bogon:SUSYQCD_CTprm_UFO erdissshaw$ ls
CT_couplings.py        SUSYQCD_CTprm_UFO.log   couplings.py            object_library.py       propagators.py
CT_parameters.py       __init__.py             function_library.py     parameters.py           vertices.py
CT_vertices.py         coupling_orders.py      lorentz.py              particles.py            write_param_card.py
```

UFO: UNIVERSAL FEYNRULES OUTPUT

Degrande, Duhr, Fuks, Grellscheid, Mattelaer, Reiter (CPC'12)

- The UFO is a set of PYTHON files
  - Particle information (particles.py)
  - Interaction information (vertices.py, couplings.py, lorentz.py, couplings_orders.py)
  - Parameter information (parameters.py)
  - Propagator information (propagators.py)
  - Tools (function_library.py, object_library.py, write_param_card.py, decays.py)
  - NLO counterterms (CT_couplings.py, CT_parameters.py, CT_vertices.py)

For example: SUSY QCD

```
bogon:SUSYQCD_CTprm_UFO erdissshaw$ ls
CT_couplings.py         SUSYQCD_CTprm_UFO.log   couplings.py            object_library.py       propagators.py
CT_parameters.py        __init__.py             function_library.py     parameters.py           vertices.py
CT_vertices.py          coupling_orders.py      lorentz.py              particles.py            write_param_card.py
```

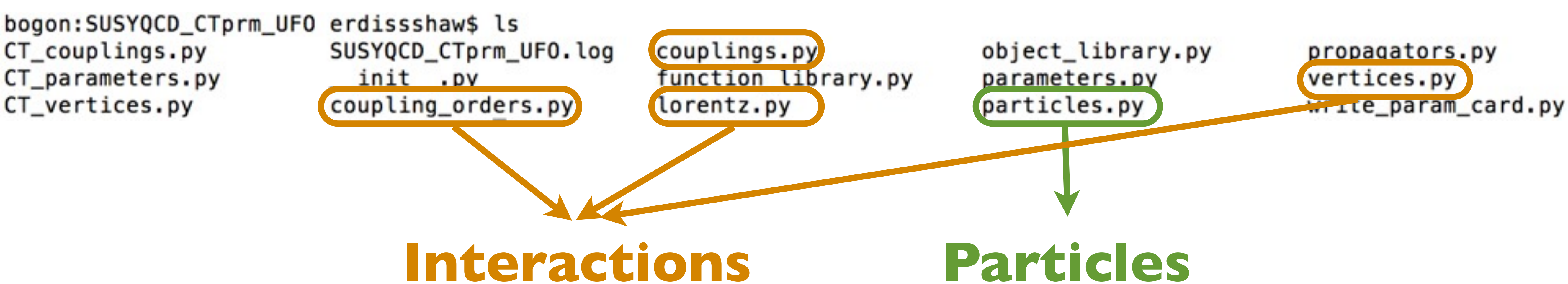
Particles

UFO: UNIVERSAL FEYNRULES OUTPUT

Degrande, Duhr, Fuks, Grellscheid, Mattelaer, Reiter (CPC'12)

- The UFO is a set of PYTHON files
  - Particle information (particles.py)
  - Interaction information (vertices.py, couplings.py, lorentz.py, couplings_orders.py)
  - Parameter information (parameters.py)
  - Propagator information (propagators.py)
  - Tools (function_library.py, object_library.py, write_param_card.py, decays.py)
  - NLO counterterms (CT_couplings.py, CT_parameters.py, CT_vertices.py)

For example: SUSY QCD

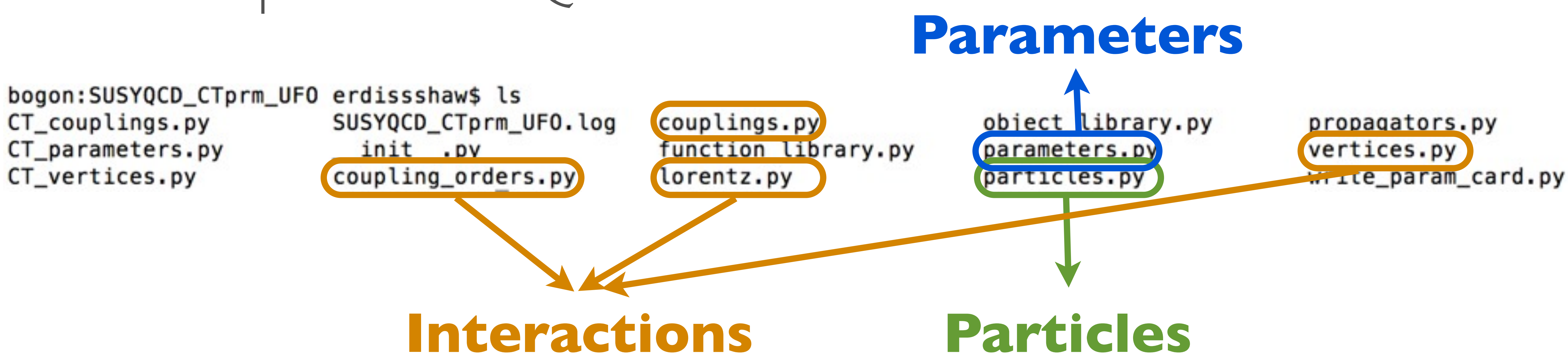


UFO: UNIVERSAL FEYNRULES OUTPUT

Degrande, Duhr, Fuks, Grellscheid, Mattelaer, Reiter (CPC'12)

- The UFO is a set of PYTHON files
  - Particle information (particles.py)
  - Interaction information (vertices.py, couplings.py, lorentz.py, couplings_orders.py)
  - Parameter information (parameters.py)
  - Propagator information (propagators.py)
  - Tools (function_library.py, object_library.py, write_param_card.py, decays.py)
  - NLO counterterms (CT_couplings.py, CT_parameters.py, CT_vertices.py)

For example: SUSY QCD

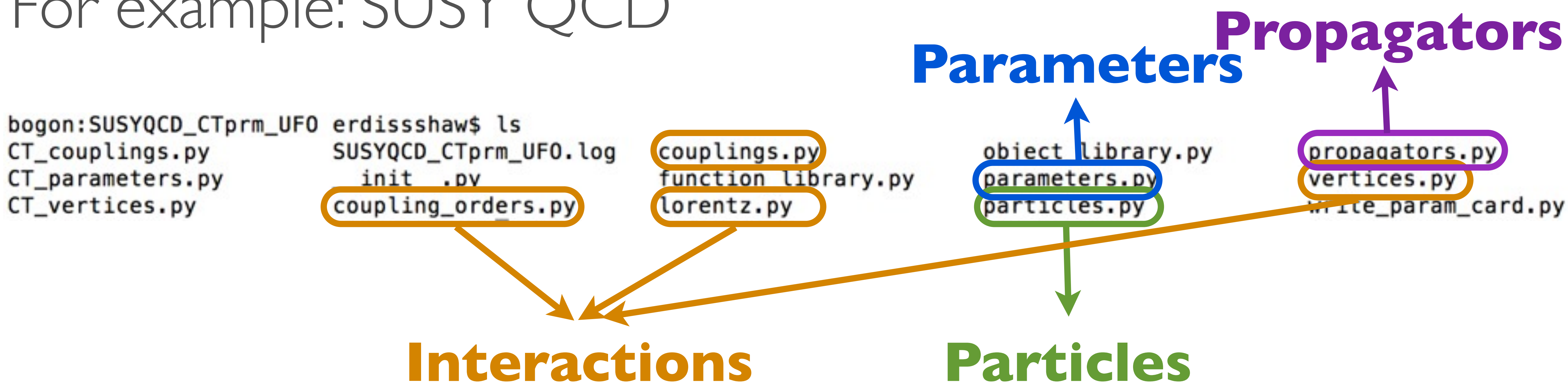


UFO: UNIVERSAL FEYNRULES OUTPUT

Degrande, Duhr, Fuks, Grellscheid, Mattelaer, Reiter (CPC'12)

- The UFO is a set of PYTHON files
  - Particle information (particles.py)
  - Interaction information (vertices.py, couplings.py, lorentz.py, couplings_orders.py)
  - Parameter information (parameters.py)
  - Propagator information (propagators.py)
  - Tools (function_library.py, object_library.py, write_param_card.py, decays.py)
  - NLO counterterms (CT_couplings.py, CT_parameters.py, CT_vertices.py)

For example: SUSY QCD

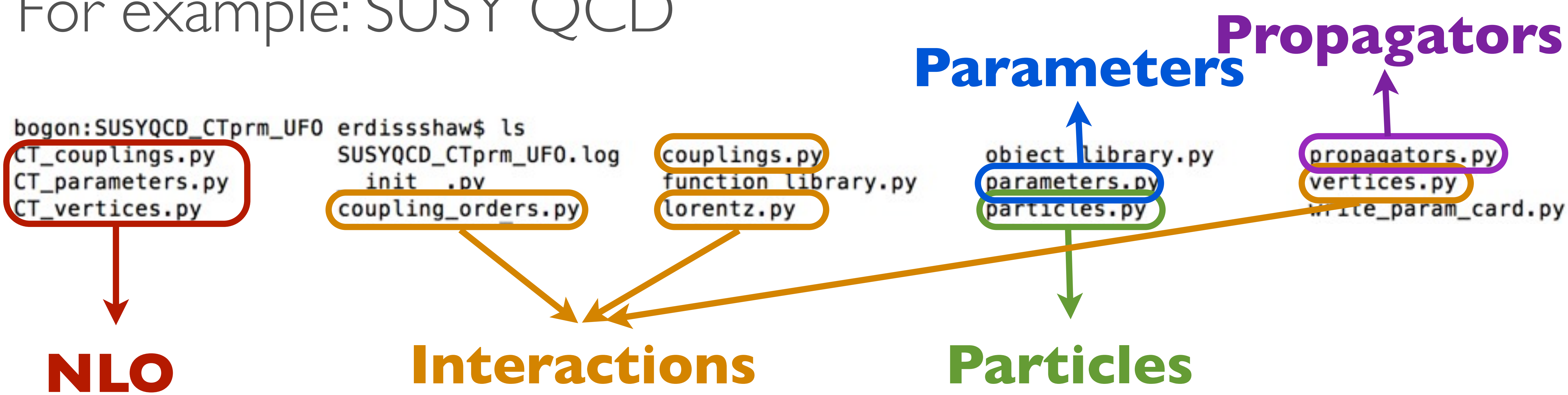


UFO: UNIVERSAL FEYNRULES OUTPUT

Degrande, Duhr, Fuks, Grellscheid, Mattelaer, Reiter (CPC'12)

- The UFO is a set of PYTHON files
  - Particle information (particles.py)
  - Interaction information (vertices.py, couplings.py, lorentz.py, couplings_orders.py)
  - Parameter information (parameters.py)
  - Propagator information (propagators.py)
  - Tools (function_library.py, object_library.py, write_param_card.py, decays.py)
  - NLO counterterms (CT_couplings.py, CT_parameters.py, CT_vertices.py)

For example: SUSY QCD



UFO: UNIVERSAL FEYNRULES OUTPUT

Degrande, Duhr, Fuks, Grellscheid, Mattelaer, Reiter (CPC'12)

UFO: UNIVERSAL FEYNRULES OUTPUT

Degrande, Duhr, Fuks, Grellscheid, Mattelaer, Reiter (CPC'12)

- **Particles are in** particles.py
  - Instances of the particle class
  - spin, color, mass, width, PDG etc

```
go = Particle(pdg_code = 1000021,
              name = 'go',
              antiname = 'go',
              spin = 2,
              color = 8,
              mass = Param.Mgo,
              width = Param.Wgo,
              texname = 'go',
              antitexname = 'go',
              charge = 0,
              GhostNumber = 0,
              LeptonNumber = 0,
              Y = 0)
```

UFO: UNIVERSAL FEYNRULES OUTPUT

Degrande, Duhr, Fuks, Grellscheid, Mattelaer, Reiter (CPC'12)

- **Particles are in** particles.py
  - Instances of the particle class
  - spin, color, mass, width, PDG etc

```
go = Particle(pdg_code = 1000021,
              name = 'go',
              antiname = 'go',
              spin = 2,
              color = 8,
              mass = Param.Mgo,
              width = Param.Wgo,
              texname = 'go',
              antitexname = 'go',
              charge = 0,
              GhostNumber = 0,
              LeptonNumber = 0,
              Y = 0)
```

- **Parameters are in** parameters.py
  - External parameters are in LHA-like
  - Python-compliant formula for int. para

```
aS = Parameter(name = 'aS',
               nature = 'external',
               type = 'real',
               value = 0.1184,
               texname = '\\alpha _s',
               lhablock = 'SMINPUTS',
               lhacode = [ 3 ])

G = Parameter(name = 'G',
              nature = 'internal',
              type = 'real',
              value = '2*cmath.sqrt(aS)*cmath.sqrt(cmath.pi)',
              texname = 'G')
```

UFO: UNIVERSAL FEYNRULES OUTPUT

Degrande, Duhr, Fuks, Grellscheid, Mattelaer, Reiter (CPC'12)

- **Interactions are in** vertices.py, couplings.py, lorentz.py, coupling_orders,py
  - Vertices are decomposed in a spin x color basis, coupling being coordinates
  - Example: the quartic gluon vertex can be written as

$$
\begin{aligned}
&ig_s^2\, f^{a_1a_2b} f^{ba_3a_4}\, (\eta^{\mu_1\mu_4}\eta^{\mu_2\mu_3} - \eta^{\mu_1\mu_3}\eta^{\mu_2\mu_4}) \\
&+ ig_s^2\, f^{a_1a_3b} f^{ba_2a_4}\, (\eta^{\mu_1\mu_4}\eta^{\mu_2\mu_3} - \eta^{\mu_1\mu_2}\eta^{\mu_3\mu_4}) \\
&+ ig_s^2\, f^{a_1a_4b} f^{ba_2a_3}\, (\eta^{\mu_1\mu_3}\eta^{\mu_2\mu_4} - \eta^{\mu_1\mu_2}\eta^{\mu_3\mu_4})
\end{aligned}
\Rightarrow
\left(f^{a_1a_2b} f^{ba_3a_4},\ f^{a_1a_3b} f^{ba_2a_4},\ f^{a_1a_4b} f^{ba_2a_3}\right)
\times \begin{pmatrix} ig_s^2 & 0 & 0 \\ 0 & ig_s^2 & 0 \\ 0 & 0 & ig_s^2 \end{pmatrix}
\begin{pmatrix} \eta^{\mu_1\mu_4}\eta^{\mu_2\mu_3} - \eta^{\mu_1\mu_3}\eta^{\mu_2\mu_4} \\ \eta^{\mu_1\mu_4}\eta^{\mu_2\mu_3} - \eta^{\mu_1\mu_2}\eta^{\mu_3\mu_4} \\ \eta^{\mu_1\mu_3}\eta^{\mu_2\mu_4} - \eta^{\mu_1\mu_2}\eta^{\mu_3\mu_4} \end{pmatrix}
$$

- vertices.py: define all Feynman rules for vertices in the model

```
V_37 = Vertex(name = 'V_37',
              particles = [ P.g, P.g, P.g, P.g ],
              color = [ 'f(-1,1,2)*f(3,4,-1)', 'f(-1,1,3)*f(2,4,-1)', 'f(-1,1,4)*f(2,3,-1)' ],
              lorentz = [ L.VVVV2, L.VVVV3, L.VVVV4 ],
              couplings = {(1,0):C.GC_20,(0,0):C.GC_20,(2,1):C.GC_20,(0,1):C.GC_19,(2,2):C.GC_19,(1,2):C.GC_19})
```

- lorentz.py: define the Lorentz structure in the model

```
VVVV2 = Lorentz(name = 'VVVV2',
                spins = [ 3, 3, 3, 3 ],
                structure = 'Metric(1,4)*Metric(2,3)')
```

- couplings.py: define the coupling constant in the model

```
GC_20 = Coupling(name = 'GC_20',
                 value = 'complex(0,1)*G**2',
                 order = {'QCD':2})
```

- coupling_orders.py: define the coupling orders in the model

```
QCD = CouplingOrder(name = 'QCD',
                    expansion_order = 99,
                    hierarchy = 1,
                    perturbative_expansion = 1)
```

UFO: UNIVERSAL FEYNRULES OUTPUT

Degrande, Duhr, Fuks, Grellscheid, Mattelaer, Reiter (CPC'12)

- **Interactions are in** vertices.py, couplings.py, lorentz.py, coupling_orders,py
  - Vertices are decomposed in a spin x color basis, coupling being coordinates
  - Example: the quartic gluon vertex can be written as

$$\begin{aligned} & i g_s^2\, f^{a_1 a_2 b} f^{b a_3 a_4} \left(\eta^{\mu_1\mu_4}\eta^{\mu_2\mu_3} - \eta^{\mu_1\mu_3}\eta^{\mu_2\mu_4}\right) \\ &+ i g_s^2\, f^{a_1 a_3 b} f^{b a_2 a_4} \left(\eta^{\mu_1\mu_4}\eta^{\mu_2\mu_3} - \eta^{\mu_1\mu_2}\eta^{\mu_3\mu_4}\right) \\ &+ i g_s^2\, f^{a_1 a_4 b} f^{b a_2 a_3} \left(\eta^{\mu_1\mu_3}\eta^{\mu_2\mu_4} - \eta^{\mu_1\mu_2}\eta^{\mu_3\mu_4}\right) \end{aligned} \Rightarrow \begin{aligned} & \left(f^{a_1 a_2 b} f^{b a_3 a_4},\; f^{a_1 a_3 b} f^{b a_2 a_4},\; f^{a_1 a_4 b} f^{b a_2 a_3}\right) \\ & \times \begin{pmatrix} i g_s^2 & 0 & 0 \\ 0 & i g_s^2 & 0 \\ 0 & 0 & i g_s^2 \end{pmatrix} \begin{pmatrix} \eta^{\mu_1\mu_4}\eta^{\mu_2\mu_3} - \eta^{\mu_1\mu_3}\eta^{\mu_2\mu_4} \\ \eta^{\mu_1\mu_4}\eta^{\mu_2\mu_3} - \eta^{\mu_1\mu_2}\eta^{\mu_3\mu_4} \\ \eta^{\mu_1\mu_3}\eta^{\mu_2\mu_4} - \eta^{\mu_1\mu_2}\eta^{\mu_3\mu_4} \end{pmatrix} \end{aligned}$$

- vertices.py: define all Feynman rules for vertices in the model

```
V_37 = Vertex(name = 'V_37',
              particles = [ P.g, P.g, P.g, P.g ],
              color = [ 'f(-1,1,2)*f(3,4,-1)', 'f(-1,1,3)*f(2,4,-1)', 'f(-1,1,4)*f(2,3,-1)' ],
              lorentz = [ L.VVVV2, L.VVVV3, L.VVVV4 ],
              couplings = {(1,0):C.GC_20,(0,0):C.GC_20,(2,1):C.GC_20,(0,1):C.GC_19,(2,2):C.GC_19,(1,2):C.GC_19})
```

- lorentz.py: define the Lorentz structure in the model

```
VVVV2 = Lorentz(name = 'VVVV2',
                spins = [ 3, 3, 3, 3 ],
                structure = 'Metric(1,4)*Metric(2,3)')
```

- couplings.py: define the coupling constant in the model

```
GC_20 = Coupling(name = 'GC_20',
                 value = 'complex(0,1)*G**2',
                 order = {'QCD':2})
```

- coupling_orders.py: define the coupling orders in the model

```
QCD = CouplingOrder(name = 'QCD',
                    expansion_order = 99,
                    hierarchy = 1,
                    perturbative_expansion = 1)
```

Make sure > 0 for NLO QCD

UFO@NLO

UFO@NLO

- Provide renormalization scale in parameters.py

```
MU_R = Parameter(name = 'MU_R',
                 nature = 'external',
                 type = 'real',
                 value = 91.188,
                 texname = '\\text{\\mu_r}',
                 lhablock = 'LOOP',
                 lhacode = [1])
```

UFO@NLO

- Provide renormalization scale in parameters.py

```
MU_R = Parameter(name = 'MU_R',
                 nature = 'external',
                 type = 'real',
                 value = 91.188,
                 texname = '\\text{\\mu_r}',
                 lhablock = 'LOOP',
                 lhacode = [1])
```

- CT_vertices.py:UV, R2 counter term vertices

```
V_2 = CTVertex(name = 'V_2',
               type = 'R2',
               particles = [ P.g, P.g, P.g, P.g ],
               color = [ 'd(-1,1,3)*d(-1,2,4)', 'd(-1,1,3)*f(-1,2,4)', 'd(-1,1,4)*d(-1,2,3)', 'd(-1,1,4)*f(-1,2,3)', 'd(-1,2,3)*f(-1,1,4)', 'd(-1,2,4)*f(-1,1,3)', 'f(-1,1\
,2)*f(-1,3,4)', 'f(-1,1,3)*f(-1,2,4)', 'f(-1,1,4)*f(-1,2,3)', 'Identity(1,2)*Identity(3,4)', 'Identity(1,3)*Identity(2,4)', 'Identity(1,4)*Identity(2,3)' ],
               lorentz = [ L.VVVV2, L.VVVV3, L.VVVV4 ],
               loop_particles = [ [ [P.b], [P.c], [P.d], [P.s], [P.t], [P.u] ], [ [P.g] ], [ [P.go] ] ],
               couplings = {(2,0,0):C.R2GC_101_4,(2,0,1):C.R2GC_100_3,(2,0,2):C.R2GC_100_2,(0,0,0):C.R2GC_101_4,(0,0,1):C.R2GC_100_3,(0,0,2):C.R2GC_100_2,(4,0,0):C.R2GC_9\
9_171,(4,0,1):C.R2GC_99_172,(4,0,2):C.R2GC_99_173,(3,0,0):C.R2GC_99_171,(3,0,1):C.R2GC_99_172,(3,0,2):C.R2GC_99_173,(8,0,0):C.R2GC_100_1,(8,0,1):C.R2GC_100_2,(8,0,2):C.R2\
GC_100_3,(6,0,0):C.R2GC_110_22,(6,0,1):C.R2GC_112_26,(6,0,2):C.R2GC_110_23,(7,0,0):C.R2GC_111_24,(7,0,1):C.R2GC_105_11,(7,0,2):C.R2GC_111_25,(5,0,0):C.R2GC_99_171,(5,0,1)\
:C.R2GC_99_172,(5,0,2):C.R2GC_99_173,(1,0,0):C.R2GC_99_171,(1,0,1):C.R2GC_99_172,(1,0,2):C.R2GC_99_173,(11,0,0):C.R2GC_103_7,(11,0,1):C.R2GC_103_8,(11,0,2):C.R2GC_103_9,(\
10,0,0):C.R2GC_103_7,(10,0,1):C.R2GC_103_8,(10,0,2):C.R2GC_103_9,(9,0,1):C.R2GC_102_5,(9,0,2):C.R2GC_102_6,(2,1,0):C.R2GC_101_4,(2,1,1):C.R2GC_100_3,(2,1,2):C.R2GC_100_2,\
(0,1,0):C.R2GC_101_4,(0,1,1):C.R2GC_100_3,(0,1,2):C.R2GC_100_2,(4,1,0):C.R2GC_99_171,(4,1,1):C.R2GC_99_172,(4,1,2):C.R2GC_99_173,(3,1,0):C.R2GC_99_171,(3,1,1):C.R2GC_99_1\
72,(3,1,2):C.R2GC_99_173,(8,1,0):C.R2GC_100_1,(8,1,1):C.R2GC_105_11,(8,1,2):C.R2GC_100_3,(6,1,0):C.R2GC_115_29,(6,1,1):C.R2GC_115_30,(6,1,2):C.R2GC_115_31,(7,1,0):C.R2GC_\
111_24,(7,1,1):C.R2GC_100_2,(7,1,2):C.R2GC_111_25,(5,1,0):C.R2GC_99_171,(5,1,1):C.R2GC_99_172,(5,1,2):C.R2GC_99_173,(1,1,0):C.R2GC_99_171,(1,1,1):C.R2GC_99_172,(1,1,2):C.\
R2GC_99_173,(11,1,0):C.R2GC_103_7,(11,1,1):C.R2GC_103_8,(11,1,2):C.R2GC_103_9,(10,1,0):C.R2GC_103_7,(10,1,1):C.R2GC_103_8,(10,1,2):C.R2GC_103_9,(9,1,1):C.R2GC_102_5,(9,1,\
2):C.R2GC_102_6,(0,2,0):C.R2GC_101_4,(0,2,1):C.R2GC_100_3,(0,2,2):C.R2GC_100_2,(2,2,0):C.R2GC_101_4,(2,2,1):C.R2GC_100_3,(2,2,2):C.R2GC_100_2,(5,2,0):C.R2GC_99_171,(5,2,1\
):C.R2GC_99_172,(5,2,2):C.R2GC_99_173,(1,2,0):C.R2GC_99_171,(1,2,1):C.R2GC_99_172,(1,2,2):C.R2GC_99_173,(7,2,0):C.R2GC_114_27,(7,2,1):C.R2GC_104_10,(7,2,2):C.R2GC_114_28,\
(4,2,0):C.R2GC_99_171,(4,2,1):C.R2GC_99_172,(4,2,2):C.R2GC_99_173,(3,2,0):C.R2GC_99_171,(3,2,1):C.R2GC_99_172,(3,2,2):C.R2GC_99_173,(8,2,0):C.R2GC_100_1,(8,2,1):C.R2GC_10\
4_10,(8,2,2):C.R2GC_100_3,(6,2,0):C.R2GC_110_22,(6,2,2):C.R2GC_110_23,(11,2,0):C.R2GC_103_7,(11,2,1):C.R2GC_103_8,(11,2,2):C.R2GC_103_9,(10,2,0):C.R2GC_103_7,(10,2,1):C.R\
2GC_103_8,(10,2,2):C.R2GC_103_9,(9,2,1):C.R2GC_102_5,(9,2,2):C.R2GC_102_6})
```

UFO@NLO

- Provide renormalization scale in parameters.py

```
MU_R = Parameter(name = 'MU_R',
                 nature = 'external',
                 type = 'real',
                 value = 91.188,
                 texname = '\\text{\\mu_r}',
                 lhablock = 'LOOP',
                 lhacode = [1])
```

- CT_vertices.py:UV, R2 counter term vertices

```
V_351 = CTVertex(name = 'V_351',
                 type = 'UV',
                 particles = [ P.g, P.g, P.g, P.g ],
                 color = [ 'd(-1,1,3)*d(-1,2,4)', 'd(-1,1,3)*f(-1,2,4)', 'd(-1,1,4)*d(-1,2,3)', 'd(-1,1,4)*f(-1,2,3)', 'd(-1,2,3)*f(-1,1,4)', 'd(-1,2,4)*f(-1,1,3)', 'f(-1,1,2)*f(-1,3,4)', 'f(-1,1,3)*f\
(-1,2,4)', 'f(-1,1,4)*f(-1,2,3)', 'Identity(1,2)*Identity(3,4)', 'Identity(1,3)*Identity(2,4)', 'Identity(1,4)*Identity(2,3)' ],
                 lorentz = [ L.VVVV2, L.VVVV3, L.VVVV4 ],
                 loop_particles = [ [ [P.b] ], [ [P.b], [P.c], [P.d], [P.s], [P.sbL], [P.sbR], [P.scL], [P.scR], [P.sdL], [P.sdR], [P.ssL], [P.ssR], [P.stL], [P.stR], [P.suL], [P.suR], [P.t], [P.u] ],\
 [ [P.b], [P.c], [P.d], [P.s], [P.t], [P.u] ], [ [P.c] ], [ [P.d] ], [ [P.g] ], [ [P.ghG] ], [ [P.go] ], [ [P.s] ], [ [P.sbL] ], [ [P.sbL], [P.sbR], [P.scL], [P.scR], [P.sdL], [P.sdR], [P.ssL], [P.ssR\
], [P.stL], [P.stR], [P.suL], [P.suR] ], [ [P.sbR] ], [ [P.scL] ], [ [P.scR] ], [ [P.sdL] ], [ [P.sdR] ], [ [P.ssL] ], [ [P.ssR] ], [ [P.stL] ], [ [P.stR] ], [ [P.suL] ], [ [P.suR] ], [ [P.t] ], [ [P.\
u] ] ],
                 couplings = {(2,0,5):C.UVGC_100_2,(2,0,6):C.UVGC_100_1,(0,0,5):C.UVGC_100_2,(0,0,6):C.UVGC_100_1,(4,0,5):C.UVGC_99_1085,(4,0,6):C.UVGC_99_1086,(3,0,5):C.UVGC_99_1085,(3,0,6):C.UVGC_99\
_1086,(8,0,5):C.UVGC_100_1,(8,0,6):C.UVGC_100_2,(6,0,0):C.UVGC_112_137,(6,0,3):C.UVGC_112_138,(6,0,4):C.UVGC_112_139,(6,0,5):C.UVGC_112_140,(6,0,6):C.UVGC_112_141,(6,0,7):C.UVGC_112_142,(6,0,8):C.UVGC\
_112_143,(6,0,9):C.UVGC_112_144,(6,0,11):C.UVGC_112_145,(6,0,12):C.UVGC_112_146,(6,0,13):C.UVGC_112_147,(6,0,14):C.UVGC_112_148,(6,0,15):C.UVGC_112_149,(6,0,16):C.UVGC_112_150,(6,0,17):C.UVGC_112_151,\
(6,0,18):C.UVGC_112_152,(6,0,19):C.UVGC_112_153,(6,0,20):C.UVGC_112_154,(6,0,21):C.UVGC_112_155,(6,0,22):C.UVGC_112_156,(6,0,23):C.UVGC_112_157,(7,0,0):C.UVGC_112_137,(7,0,3):C.UVGC_112_138,(7,0,4):C.\
UVGC_112_139,(7,0,5):C.UVGC_105_31,(7,0,6):C.UVGC_113_158,(7,0,7):C.UVGC_112_142,(7,0,8):C.UVGC_112_143,(7,0,9):C.UVGC_112_144,(7,0,11):C.UVGC_112_145,(7,0,12):C.UVGC_112_146,(7,0,13):C.UVGC_112_147,(\
7,0,14):C.UVGC_112_148,(7,0,15):C.UVGC_112_149,(7,0,16):C.UVGC_112_150,(7,0,17):C.UVGC_112_151,(7,0,18):C.UVGC_112_152,(7,0,19):C.UVGC_112_153,(7,0,20):C.UVGC_112_154,(7,0,21):C.UVGC_112_155,(7,0,22):\
C.UVGC_112_156,(7,0,23):C.UVGC_112_157,(5,0,5):C.UVGC_99_1085,(5,0,6):C.UVGC_99_1086,(1,0,5):C.UVGC_99_1085,(1,0,6):C.UVGC_99_1086,(11,0,5):C.UVGC_103_5,(11,0,6):C.UVGC_103_6,(10,0,5):C.UVGC_103_5,(10\
,0,6):C.UVGC_103_6,(9,0,5):C.UVGC_102_3,(9,0,6):C.UVGC_102_4,(2,1,5):C.UVGC_100_2,(2,1,6):C.UVGC_100_1,(0,1,5):C.UVGC_100_2,(0,1,6):C.UVGC_100_1,(4,1,5):C.UVGC_99_1085,(4,1,6):C.UVGC_99_1086,(3,1,5):C\
.UVGC_99_1085,(3,1,6):C.UVGC_99_1086,(8,1,0):C.UVGC_105_28,(8,1,3):C.UVGC_105_29,(8,1,4):C.UVGC_105_30,(8,1,5):C.UVGC_105_31,(8,1,6):C.UVGC_105_32,(8,1,7):C.UVGC_105_33,(8,1,8):C.UVGC_105_34,(8,1,9):C\
.UVGC_105_35,(8,1,11):C.UVGC_105_36,(8,1,12):C.UVGC_105_37,(8,1,13):C.UVGC_105_38,(8,1,14):C.UVGC_105_39,(8,1,15):C.UVGC_105_40,(8,1,16):C.UVGC_105_41,(8,1,17):C.UVGC_105_42,(8,1,18):C.UVGC_105_43,(8,\
1,19):C.UVGC_105_44,(8,1,20):C.UVGC_105_45,(8,1,21):C.UVGC_105_46,(8,1,22):C.UVGC_105_47,(8,1,23):C.UVGC_105_48,(6,1,0):C.UVGC_114_159,(6,1,3):C.UVGC_114_160,(6,1,4):C.UVGC_114_161,(6,1,5):C.UVGC_115_\
179,(6,1,6):C.UVGC_115_180,(6,1,7):C.UVGC_114_163,(6,1,8):C.UVGC_114_164,(6,1,9):C.UVGC_115_181,(6,1,11):C.UVGC_115_182,(6,1,12):C.UVGC_115_183,(6,1,13):C.UVGC_115_184,(6,1,14):C.UVGC_115_185,(6,1,15)\
:C.UVGC_115_186,(6,1,16):C.UVGC_115_187,(6,1,17):C.UVGC_115_188,(6,1,18):C.UVGC_115_189,(6,1,19):C.UVGC_115_190,(6,1,20):C.UVGC_115_191,(6,1,21):C.UVGC_115_192,(6,1,22):C.UVGC_114_177,(6,1,23):C.UVGC_\
114_178,(7,1,1):C.UVGC_110_133,(7,1,5):C.UVGC_100_1,(7,1,6):C.UVGC_111_136,(7,1,7):C.UVGC_110_134,(5,1,5):C.UVGC_99_1085,(5,1,6):C.UVGC_99_1086,(1,1,5):C.UVGC_99_1085,(1,1,6):C.UVGC_99_1086,(11,1,5):C\
.UVGC_103_5,(11,1,6):C.UVGC_103_6,(10,1,5):C.UVGC_103_5,(10,1,6):C.UVGC_103_6,(9,1,5):C.UVGC_102_3,(9,1,6):C.UVGC_102_4,(0,2,5):C.UVGC_100_2,(0,2,6):C.UVGC_100_1,(2,2,5):C.UVGC_100_2,(2,2,6):C.UVGC_10\
0_1,(5,2,5):C.UVGC_99_1085,(5,2,6):C.UVGC_99_1086,(1,2,5):C.UVGC_99_1085,(1,2,6):C.UVGC_99_1086,(7,2,0):C.UVGC_114_159,(7,2,3):C.UVGC_114_160,(7,2,4):C.UVGC_114_161,(7,2,5):C.UVGC_104_10,(7,2,6):C.UVG\
C_114_162,(7,2,7):C.UVGC_114_163,(7,2,8):C.UVGC_114_164,(7,2,9):C.UVGC_114_165,(7,2,11):C.UVGC_114_166,(7,2,12):C.UVGC_114_167,(7,2,13):C.UVGC_114_168,(7,2,14):C.UVGC_114_169,(7,2,15):C.UVGC_114_170,(\
7,2,16):C.UVGC_114_171,(7,2,17):C.UVGC_114_172,(7,2,18):C.UVGC_114_173,(7,2,19):C.UVGC_114_174,(7,2,20):C.UVGC_114_175,(7,2,21):C.UVGC_114_176,(7,2,22):C.UVGC_114_177,(7,2,23):C.UVGC_114_178,(4,2,5):C\
.UVGC_99_1085,(4,2,6):C.UVGC_99_1086,(3,2,5):C.UVGC_99_1085,(3,2,6):C.UVGC_99_1086,(8,2,0):C.UVGC_104_7,(8,2,3):C.UVGC_104_8,(8,2,4):C.UVGC_104_9,(8,2,5):C.UVGC_104_10,(8,2,6):C.UVGC_104_11,(8,2,7):C.\
UVGC_104_12,(8,2,8):C.UVGC_104_13,(8,2,9):C.UVGC_104_14,(8,2,11):C.UVGC_104_15,(8,2,12):C.UVGC_104_16,(8,2,13):C.UVGC_104_17,(8,2,14):C.UVGC_104_18,(8,2,15):C.UVGC_104_19,(8,2,16):C.UVGC_104_20,(8,2,1\
7):C.UVGC_104_21,(8,2,18):C.UVGC_104_22,(8,2,19):C.UVGC_104_23,(8,2,20):C.UVGC_104_24,(8,2,21):C.UVGC_104_25,(8,2,22):C.UVGC_104_26,(8,2,23):C.UVGC_104_27,(6,2,2):C.UVGC_110_133,(6,2,6):C.UVGC_102_3,(\
6,2,7):C.UVGC_110_134,(6,2,10):C.UVGC_110_135,(11,2,5):C.UVGC_103_5,(11,2,6):C.UVGC_103_6,(10,2,5):C.UVGC_103_5,(10,2,6):C.UVGC_103_6,(9,2,5):C.UVGC_102_3,(9,2,6):C.UVGC_102_4})
```

UFO@NLO

- Provide renormalization scale in parameters.py

```
MU_R = Parameter(name = 'MU_R',
                 nature = 'external',
                 type = 'real',
                 value = 91.188,
                 texname = '\\text{\\mu_r}',
                 lhablock = 'LOOP',
                 lhacode = [1])
```

- CT_vertices.py:UV, R2 counter term vertices

- CT_couplings.py: couplings for UV and R2 counter terms

```
UVGC_104_23 = Coupling(name = 'UVGC_104_23',
                       value = '-((FRCTdeltaxaSxstR*complex(0,1)*G**2)/aS) - 2*FRCTdeltaZxGGxstR*complex(0,1)*G**2 + (complex(0,1)*G**4*invFREps)/(32.*cmath.pi**2)',
                       order = {'QCD':4})
```

UFO@NLO

- Provide renormalization scale in parameters.py

```
MU_R = Parameter(name = 'MU_R',
                 nature = 'external',
                 type = 'real',
                 value = 91.188,
                 texname = '\\text{\\mu_r}',
                 lhablock = 'LOOP',
                 lhacode = [1])
```

- CT_vertices.py:UV, R2 counter term vertices
- CT_couplings.py: couplings for UV and R2 counter terms

```
UVGC_104_23 = Coupling(name = 'UVGC_104_23',
                       value = '-((FRCTdeltaxaSxstR*complex(0,1)*G**2)/aS) - 2*FRCTdeltaZxGGxstR*complex(0,1)*G**2 + (complex(0,1)*G**4*invFREps)/(32.*cmath.pi**2)',
                       order = {'QCD':4})
```

- CT_parameters.py: parameters for UV and R2

```
FRCTdeltaZxttLxtG = CTParameter(name = 'FRCTdeltaZxttLxtG',
                                type = 'complex',
                                value = {-1:'-G**2/(6.*cmath.pi**2)',0:'-G**2/(3.*cmath.pi**2) + (G**2*reglog(MT/MU_R))/(2.*cmath.pi**2)'},
                                texname = 'FRCTdeltaZxttLxtG')
```

UFO@NLO

- Provide renormalization scale in parameters.py

```
MU_R = Parameter(name = 'MU_R',
                 nature = 'external',
                 type = 'real',
                 value = 91.188,
                 texname = '\\text{\\mu_r}',
                 lhablock = 'LOOP',
                 lhacode = [1])
```

- CT_vertices.py:UV, R2 counter term vertices

- CT_couplings.py: couplings for UV and R2 counter terms

```
UVGC_104_23 = Coupling(name = 'UVGC_104_23',
                       value = '-((FRCTdeltaxaSxstR*complex(0,1)*G**2)/aS) - 2*FRCTdeltaZxGGxstR*complex(0,1)*G**2 + (complex(0,1)*G**4*invFREps)/(32.*cmath.pi**2)',
                       order = {'QCD':4})
```

- CT_parameters.py: parameters for UV and R2

```
FRCTdeltaZxttLxtG = CTParameter(name = 'FRCTdeltaZxttLxtG',
                                type = 'complex',
                                value = {-1:'-G**2/(6.*cmath.pi**2)',0:'-G**2/(3.*cmath.pi**2) + (G**2*reglog(MT/MU_R))/(2.*cmath.pi**2)'},
                                texname = 'FRCTdeltaZxttLxtG')
```

coefficient of $\frac{1}{\epsilon}$

- Provide renormalization scale in parameters.py

```
MU_R = Parameter(name = 'MU_R',
                 nature = 'external',
                 type = 'real',
                 value = 91.188,
                 texname = '\\text{\\mu_r}',
                 lhablock = 'LOOP',
                 lhacode = [1])
```

- CT_vertices.py: UV, R2 counter term vertices
- CT_couplings.py: couplings for UV and R2 counter terms

```
UVGC_104_23 = Coupling(name = 'UVGC_104_23',
                       value = '-((FRCTdeltaxaSxstR*complex(0,1)*G**2)/aS) - 2*FRCTdeltaZxGGxstR*complex(0,1)*G**2 + (complex(0,1)*G**4*invFREps)/(32.*cmath.pi**2)',
                       order = {'QCD':4})
```

- CT_parameters.py: parameters for UV and R2

```
FRCTdeltaZxttLxtG = CTParameter(name = 'FRCTdeltaZxttLxtG',
                                type = 'complex',
                                value = {-1:'-G**2/(6.*cmath.pi**2)',0:'-G**2/(3.*cmath.pi**2) + (G**2*reglog(MT/MU_R))/(2.*cmath.pi**2)'},
                                texname = 'FRCTdeltaZxttLxtG')
```

-1: coefficient of $\frac{1}{\epsilon}$

0: finite piece

UFO@NLO

- Provide renormalization scale in parameters.py

```
MU_R = Parameter(name = 'MU_R',
                 nature = 'external',
                 type = 'real',
                 value = 91.188,
                 texname = '\\text{\\mu_r}',
                 lhablock = 'LOOP',
                 lhacode = [1])
```

- CT_vertices.py: UV, R2 counter term vertices

- CT_couplings.py: couplings for UV and R2 counter terms

```
UVGC_104_23 = Coupling(name = 'UVGC_104_23',
                       value = '-((FRCTdeltaxaSxstR*complex(0,1)*G**2)/aS) - 2*FRCTdeltaZxGGxstR*complex(0,1)*G**2 + (complex(0,1)*G**4*invFREps)/(32.*cmath.pi**2)',
                       order = {'QCD':4})
```

- CT_parameters.py: parameters for UV and R2

```
FRCTdeltaZxttLxgostL = CTParameter(name = 'FRCTdeltaZxttLxgostL',
                                   type = 'complex',
                                   value = {0:'( 0 if 2*Mgo*MstL + MT**2>=Mgo**2 + MstL**2 and MT**2<=(Mgo + MstL)**2 else ( 0 if Mgo==MstL else ( 0 if Mgo==MT else ( 0 if MstL==MT else \
(G**2*cmath.sqrt(MstL**4/MU_R**4 + (-(Mgo**2/MU_R**2) + MT**2/MU_R**2)**2 - (2*MstL**2*(Mgo**2/MU_R**2 + MT**2/MU_R**2))/MU_R**2))/(12.*cmath.pi**2*cmath.sqrt((-4*Mgo**2*MstL**2)/MU_R**4\
 + (Mgo**2/MU_R**2 + MstL**2/MU_R**2 - MT**2/MU_R**2)**2)) + (G**2*Mgo**2*cmath.sqrt(MstL**4/MU_R**4 + (-(Mgo**2/MU_R**2) + MT**2/MU_R**2)**2 - (2*MstL**2*(Mgo**2/MU_R**2 + MT**2/MU_R**2\
))/MU_R**2))/(12.*cmath.pi**2*MT**2*cmath.sqrt((-4*Mgo**2*MstL**2)/MU_R**4 + (Mgo**2/MU_R**2 + MstL**2/MU_R**2 - MT**2/MU_R**2)**2)) - (G**2*MstL**2*cmath.sqrt(MstL**4/MU_R**4 + (-(Mgo**\
2/MU_R**2) + MT**2/MU_R**2)**2 - (2*MstL**2*(Mgo**2/MU_R**2 + MT**2/MU_R**2))/MU_R**2))/(12.*cmath.pi**2*MT**2*cmath.sqrt((-4*Mgo**2*MstL**2)/MU_R**4 + (Mgo**2/MU_R**2 + MstL**2/MU_R**2 \
- MT**2/MU_R**2)**2)) - (G**2*Mgo**4*cmath.sqrt(MstL**4/MU_R**4 + (-(Mgo**2/MU_R**2) + MT**2/MU_R**2)**2 - (2*MstL**2*(Mgo**2/MU_R**2 + MT**2/MU_R**2))/MU_R**2)*reglog(Mgo/MstL))/(12.*cm\
ath.pi**2*MT**4*cmath.sqrt((-4*Mgo**2*MstL**2)/MU_R**4 + (Mgo**2/MU_R**2 + MstL**2/MU_R**2 - MT**2/MU_R**2)**2)) + (G**2*Mgo**2*MstL**2*cmath.sqrt(MstL**4/MU_R**4 + (-(Mgo**2/MU_R**2) + \
MT**2/MU_R**2)**2 - (2*MstL**2*(Mgo**2/MU_R**2 + MT**2/MU_R**2))/MU_R**2)*reglog(Mgo/MstL))/(6.*cmath.pi**2*MT**4*cmath.sqrt((-4*Mgo**2*MstL**2)/MU_R**4 + (Mgo**2/MU_R**2 + MstL**2/MU_R*\
*2 - MT**2/MU_R**2)**2)) - (G**2*MstL**4*cmath.sqrt(MstL**4/MU_R**4 + (-(Mgo**2/MU_R**2) + MT**2/MU_R**2)**2 - (2*MstL**2*(Mgo**2/MU_R**2 + MT**2/MU_R**2))/MU_R**2)*reglog(Mgo/MstL))/(12\
.*cmath.pi**2*MT**4*cmath.sqrt((-4*Mgo**2*MstL**2)/MU_R**4 + (Mgo**2/MU_R**2 + MstL**2/MU_R**2 - MT**2/MU_R**2)**2)) - (G**2*Mgo**2*cmath.sqrt(MstL**4/MU_R**4 + (-(Mgo**2/MU_R**2) + MT**\
2/MU_R**2)**2 - (2*MstL**2*(Mgo**2/MU_R**2 + MT**2/MU_R**2))/MU_R**2)*reglog(Mgo/MstL))/(12.*cmath.pi**2*MT**2*cmath.sqrt((-4*Mgo**2*MstL**2)/MU_R**4 + (Mgo**2/MU_R**2 + MstL**2/MU_R**2 \
- MT**2/MU_R**2)**2)) + (G**2*MstL**2*cmath.sqrt(MstL**4/MU_R**4 + (-(Mgo**2/MU_R**2) + MT**2/MU_R**2)**2 - (2*MstL**2*(Mgo**2/MU_R**2 + MT**2/MU_R**2))/MU_R**2)*reglog(Mgo/MstL))/(12.*c\
math.pi**2*MT**2*cmath.sqrt((-4*Mgo**2*MstL**2)/MU_R**4 + (Mgo**2/MU_R**2 + MstL**2/MU_R**2 - MT**2/MU_R**2)**2)) ) ) ) ) + ( ( 0 if Mgo==MstL else ( 0 if Mgo==MT else ( 0 if MstL==MT el\
se (G**2*Mgo*MstL*re((-(MU_R**2*(Mgo**2/MU_R**2 + MstL**2/MU_R**2 - MT**2/MU_R**2 + cmath.sqrt((-4*Mgo**2*MstL**2)/MU_R**4 + (Mgo**2/MU_R**2 + MstL**2/MU_R**2 - MT**2/MU_R**2)**2)))/(2.*\
Mgo*MstL) - (MU_R**2*(-(Mgo**2/MU_R**2) - MstL**2/MU_R**2 + MT**2/MU_R**2 + cmath.sqrt((-4*Mgo**2*MstL**2)/MU_R**4 + (Mgo**2/MU_R**2 + MstL**2/MU_R**2 - MT**2/MU_R**2)**2)))/(2.*Mgo*MstL\
))*reglog((MU_R**2*(Mgo**2/MU_R**2 + MstL**2/MU_R**2 - MT**2/MU_R**2 + cmath.sqrt((-4*Mgo**2*MstL**2)/MU_R**4 + (Mgo**2/MU_R**2 + MstL**2/MU_R**2 - MT**2/MU_R**2)**2)))/(2.*Mgo*MstL))))/\
(12.*cmath.pi**2*MT**2) ) ) ) if 2*Mgo*MstL + MT**2>=Mgo**2 + MstL**2 and MT**2<(Mgo + MstL)**2 else 0 ) + ( ( 0 if Mgo==MstL else ( 0 if Mgo==MT else ( 0 if MstL==MT else (MU_R**2*G**2*\
Mgo**2*re(((MT**2*cmath.sqrt(MstL**4/MU_R**4 + (-(Mgo**2/MU_R**2) + MT**2/MU_R**2)**2 - (2*MstL**2*(Mgo**2/MU_R**2 + MT**2/MU_R**2))/MU_R**2))/MU_R**2 + (-(Mgo**2/MU_R**2) + MstL**2/MU_R\
**2)*cmath.sqrt(MstL**4/MU_R**4 + (-(Mgo**2/MU_R**2) + MT**2/MU_R**2)**2 - (2*MstL**2*(Mgo**2/MU_R**2 + MT**2/MU_R**2))/MU_R**2)*reglog(Mgo/MstL) + (MstL**4/MU_R**4 + (Mgo**2*(Mgo**2/MU_\
R**2 - MT**2/MU_R**2))/MU_R**2 - (MstL**2*((2*Mgo**2)/MU_R**2 + MT**2/MU_R**2))/MU_R**2)*reglog((MU_R**2*(Mgo**2/MU_R**2 + MstL**2/MU_R**2 - MT**2/MU_R**2 + cmath.sqrt((-4*Mgo**2*MstL**2\
)/MU_R**4 + (Mgo**2/MU_R**2 + MstL**2/MU_R**2 - MT**2/MU_R**2)**2)))/(2.*Mgo*MstL)))/cmath.sqrt((-4*Mgo**2*MstL**2)/MU_R**4 + (Mgo**2/MU_R**2 + MstL**2/MU_R**2 - MT**2/MU_R**2)**2)))/(12\
.*cmath.pi**2*MT**4) - (MU_R**2*G**2*MstL**2*re(((MT**2*cmath.sqrt(MstL**4/MU_R**4 + (-(Mgo**2/MU_R**2) + MT**2/MU_R**2)**2 - (2*MstL**2*(Mgo**2/MU_R**2 + MT**2/MU_R**2))/MU_R**2))/MU_R*\
*2 + (-(Mgo**2/MU_R**2) + MstL**2/MU_R**2)*cmath.sqrt(MstL**4/MU_R**4 + (-(Mgo**2/MU_R**2) + MT**2/MU_R**2)**2 - (2*MstL**2*(Mgo**2/MU_R**2 + MT**2/MU_R**2))/MU_R**2)*reglog(Mgo/MstL) + \
(MstL**4/MU_R**4 + (Mgo**2*(Mgo**2/MU_R**2 - MT**2/MU_R**2))/MU_R**2 - (MstL**2*((2*Mgo**2)/MU_R**2 + MT**2/MU_R**2))/MU_R**2)*reglog((MU_R**2*(Mgo**2/MU_R**2 + MstL**2/MU_R**2 - MT**2/M\
U_R**2 + cmath.sqrt((-4*Mgo**2*MstL**2)/MU_R**4 + (Mgo**2/MU_R**2 + MstL**2/MU_R**2 - MT**2/MU_R**2)**2)))/(2.*Mgo*MstL)))/cmath.sqrt((-4*Mgo**2*MstL**2)/MU_R**4 + (Mgo**2/MU_R**2 + MstL\
**2/MU_R**2 - MT**2/MU_R**2)**2)))/(12.*cmath.pi**2*MT**4) + (MU_R**2*G**2*re(((MT**2*cmath.sqrt(MstL**4/MU_R**4 + (-(Mgo**2/MU_R**2) + MT**2/MU_R**2)**2 - (2*MstL**2*(Mgo**2/MU_R**2 + M\
T**2/MU_R**2))/MU_R**2))/MU_R**2 + (-(Mgo**2/MU_R**2) + MstL**2/MU_R**2)*cmath.sqrt(MstL**4/MU_R**4 + (-(Mgo**2/MU_R**2) + MT**2/MU_R**2)**2 - (2*MstL**2*(Mgo**2/MU_R**2 + MT**2/MU_R**2)\
```

- Provide renormalization scale in parameters.py

```
MU_R = Parameter(name = 'MU_R',
                 nature = 'external',
                 type = 'real',
                 value = 91.188,
                 texname = '\\text{\\mu_r}',
                 lhablock = 'LOOP',
                 lhacode = [1])
```

- CT_vertices.py:UV, R2 counter term vertices
- CT_couplings.py: couplings for UV and R2 counter terms

```
UVGC_104_23 = Coupling(name = 'UVGC_104_23',
                       value = '-((FRCTdeltaxaSxstR*complex(0,1)*G**2)/aS) - 2*FRCTdeltaZxGGxstR*complex(0,1)*G**2 + (complex(0,1)*G**4*invFREps)/(32.*cmath.pi**2)',
                       order = {'QCD':4})
```

- CT_parameters.py: parameters for UV and R2

```
FRCTdeltaZxttLxgostL = CTParameter(name = 'FRCTdeltaZxttLxgostL',
                                   type = 'complex',
                                   value = {0:'( 0 if 2*Mgo*MstL + MT**2>=Mgo**2 + MstL**2 and MT**2<=(Mgo + MstL)**2 else ( 0 if Mgo==MstL else ( 0 if Mgo==MT else ( 0 if MstL==MT else \
(G**2*cmath.sqrt(MstL**4/MU_R**4 + (-(Mgo**2/MU_R**2) + MT**2/MU_R**2)**2 - (2*MstL**2*(Mgo**2/MU_R**2 + MT**2/MU_R**2))/MU_R**2))/(12.*cmath.pi**2*cmath.sqrt((-4*Mgo**2*MstL**2)/MU_R**4 \
 + (Mgo**2/MU_R**2 + MstL**2/MU_R**2 - MT**2/MU_R**2)**2)) + (G**2*Mgo**2*cmath.sqrt(MstL**4/MU_R**4 + (-(Mgo**2/MU_R**2) + MT**2/MU_R**2)**2 - (2*MstL**2*(Mgo**2/MU_R**2 + MT**2/MU_R**2\
))/MU_R**2))/(12.*cmath.pi**2*MT**2*cmath.sqrt((-4*Mgo**2*MstL**2)/MU_R**4 + (Mgo**2/MU_R**2 + MstL**2/MU_R**2 - MT**2/MU_R**2)**2)) - ...
...*2)'},
```

Complicated mass spectrum makes the computation heavy !!!

Tuesday, October 10, 17

ONE-LOOP INTEGRAL

$$\int d^{(4-2\epsilon)} \frac{N(l)}{D_0 D_1 D_2 \cdots D_{m-1}} = \sum_{0\leq i_0<i_1<i_2<i_3\leq m-1} d_{i_0 i_1 i_2 i_3} \mathcal{I}_0(i_0 i_1 i_2 i_3) + \sum_{0\leq i_0<i_1<i_2\leq m-1} c_{i_0 i_1 i_2} \mathcal{I}_0(i_0 i_1 i_2) + \sum_{0\leq i_0<i_1\leq m-1} b_{i_0 i_1} \mathcal{I}_0(i_0 i_1) + \sum_{0\leq i_0\leq m-1} a_{i_0} \mathcal{I}_0(i_0) + R,$$

- Integral can be reduced to a minimal basis that was known.
- Rational term R is in general process dependent.

$$\mathcal{I}_0(i_0 i_1 i_2 i_3) \equiv \int d^{(4-2\epsilon)} l \frac{1}{D_{i_0} D_{i_1} D_{i_2} D_{i_3}},$$
$$\mathcal{I}_0(i_0 i_1 i_2) \equiv \int d^{(4-2\epsilon)} l \frac{1}{D_{i_0} D_{i_1} D_{i_2}},$$
$$\mathcal{I}_0(i_0 i_1) \equiv \int d^{(4-2\epsilon)} l \frac{1}{D_{i_0} D_{i_1}},$$
$$\mathcal{I}_0(i_0) \equiv \int d^{(4-2\epsilon)} l \frac{1}{D_{i_0}}.$$

Tuesday, October 10, 17

ONE-LOOP INTEGRAL

$$\int d^{(4-2\epsilon)} \frac{N(l)}{D_0 D_1 D_2 \cdots D_{m-1}} = \sum_{0\le i_0<i_1<i_2<i_3\le m-1} d_{i_0 i_1 i_2 i_3} \mathcal{I}_0(i_0 i_1 i_2 i_3) + \sum_{0\le i_0<i_1<i_2\le m-1} c_{i_0 i_1 i_2} \mathcal{I}_0(i_0 i_1 i_2) + \sum_{0\le i_0<i_1\le m-1} b_{i_0 i_1} \mathcal{I}_0(i_0 i_1) + \sum_{0\le i_0\le m-1} a_{i_0} \mathcal{I}_0(i_0) + R,$$

Rational functions of external momentum and masses

- Integral can be reduced to a minimal basis that was known.
- Rational term R is in general process dependent.

$$\mathcal{I}_0(i_0 i_1 i_2 i_3) \equiv \int d^{(4-2\epsilon)} l \frac{1}{D_{i_0} D_{i_1} D_{i_2} D_{i_3}},$$

$$\mathcal{I}_0(i_0 i_1 i_2) \equiv \int d^{(4-2\epsilon)} l \frac{1}{D_{i_0} D_{i_1} D_{i_2}},$$

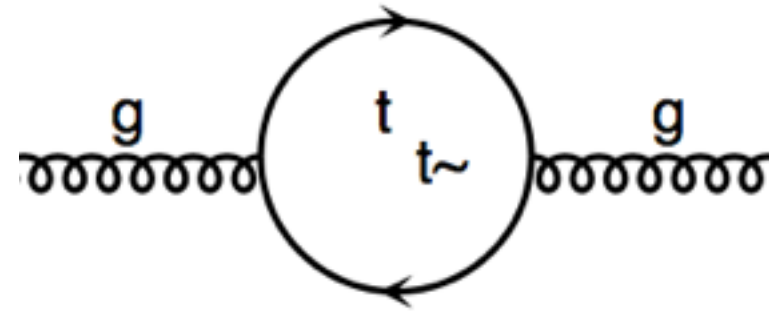
$$\mathcal{I}_0(i_0 i_1) \equiv \int d^{(4-2\epsilon)} l \frac{1}{D_{i_0} D_{i_1}},$$

$$\mathcal{I}_0(i_0) \equiv \int d^{(4-2\epsilon)} l \frac{1}{D_{i_0}}.$$

R2

- The numerator N(l) is a complicated function (Clifford algebra etc) in d- and 4-dimensional quantities.
- It is usually conv. to work N(l) in 4-dim --> super useful for numerical calculations.
- We need a special rational term R2 !

- For example, gluon SE:



$$N(l) = -\frac{\alpha_S}{(2\pi)^3}\delta_{ab}\mathrm{Tr}[\gamma^\mu(\not{l}+m_t)\gamma^\nu(\not{l}+\not{p}+m_t)]\varepsilon_\mu\varepsilon_\nu$$

- Dirac algebra gives the (d-4) numerator

$$\tilde{N}(\tilde{l}) = 4\frac{\alpha_S}{(2\pi)^3}\delta_{ab}g^{\mu\nu}\tilde{l}^2\varepsilon_\mu\varepsilon_\nu$$

- With the integration

$$\int d^{(4-2\epsilon)}l\frac{\tilde{l}^2}{(l^2-m_t^2)((l+p)^2-m_t^2)} = -\frac{i\pi^2}{2}\left(2m_t^2-\frac{p^2}{3}\right)+\mathcal{O}(\epsilon)$$

- The corresponding R2 term

$$R_2 = -\frac{i\alpha_S}{4\pi}(2m_t^2-\frac{p^2}{3})\delta_{ab}g^{\mu\nu}\varepsilon_\mu\varepsilon_\nu$$

R2

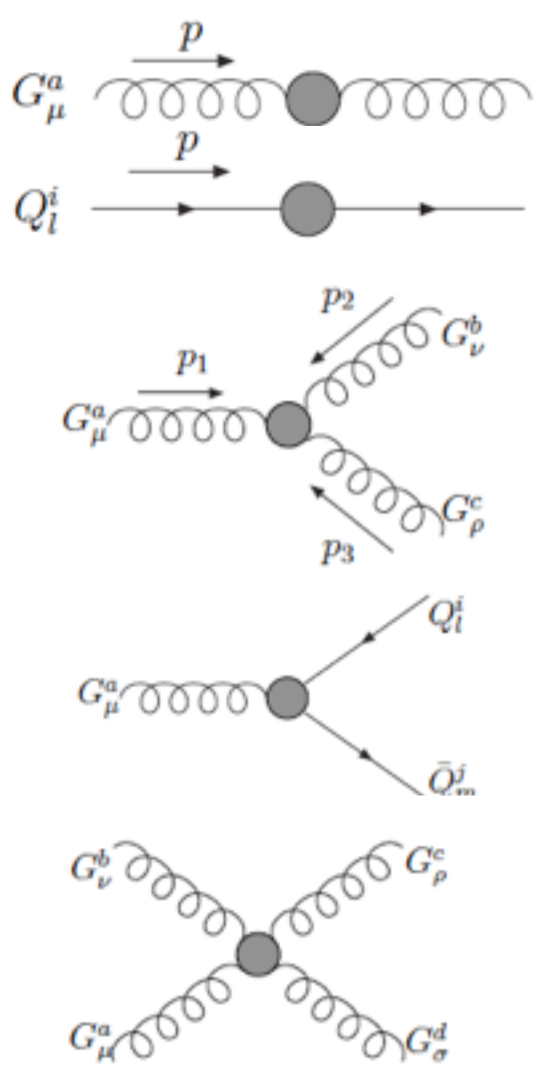
Draggiotis, Garzelli, Papadopoulos, Pittau (JHEP'09); HSS, Zhang, Chao (JHEP'11)

- It was proven that R2 is only UV related, hence universal (i.e. model dependent only), which can be derived by R2 counterterm Feynman rules and should be derived once for all in each model.

R2

Draggiotis, Garzelli, Papadopoulos, Pittau (JHEP'09); HSS, Zhang, Chao (JHEP'11)

- It was proven that R2 is only UV related, hence universal (i.e. model dependent only), which can be derived by R2 counterterm Feynman rules and should be derived once for all in each model.



$= \mathrm{Vert}(G^a_\mu, G^b_\nu)$

$= \mathrm{Vert}(Q^i_l, \bar{Q}^j_m)$

$= \mathrm{Vert}(G^a_\mu, G^b_\nu, G^c_\rho)$

$= \mathrm{Vert}(G^a_\mu, Q^i_l, \bar{Q}^j_m)$

$= \mathrm{Vert}\left(G^a_\mu, G^b_\nu, G^c_\rho, G^d_\sigma\right)$

$$\mathrm{Vert}(G^a_\mu, G^b_\nu) = \frac{i g_s^2 N_c}{48\pi^2}\,\delta^{ab}\left[\frac{p^2}{2} g_{\mu\nu} + \lambda_{HV}\left(g_{\mu\nu}p^2 - p_\mu p_\nu\right) + \sum_Q \frac{p^2 - 6m_Q^2}{N_c} g_{\mu\nu}\right]$$

$$\mathrm{Vert}(Q^i_l, \bar{Q}^j_m) = \frac{i g_s^2}{16\pi^2}\,\frac{N_c^2-1}{2N_c}\,\delta^{ij}\,\delta_{lm}\,\left(-\not{p} + 2m_{Q_l}\right)\lambda_{HV}.$$

$$\mathrm{Vert}(G^a_\mu, G^b_\nu, G^c_\rho) = -\frac{g_s^3 N_c}{48\pi^2}\left(\frac{7}{4} + \lambda_{HV} + \frac{2N_f}{N_c}\right) f^{abc}\,V_{\mu\nu\rho}(p_1,p_2,p_3)$$

$$V_{\mu\nu\rho}(p_1,p_2,p_3) = g_{\mu\nu}(p_2-p_1)_\rho + g_{\nu\rho}(p_3-p_2)_\mu + g_{\rho\mu}(p_1-p_3)_\nu.$$

$$\mathrm{Vert}(G^a_\mu, Q^i_l, \bar{Q}^j_m) = \delta_{lm}\,\frac{i g_s^3}{16\pi^2}\,T^a_{ji}\,\frac{N_c^2-1}{2N_c}\,\gamma_\mu\left(1+\lambda_{HV}\right)$$

$$\mathrm{Vert}\left(G^a_\mu, G^b_\nu, G^c_\rho, G^d_\sigma\right) = \frac{i g_s^4}{48\pi^2}\left(C_1\, g_{\mu\nu}g_{\rho\sigma} + C_2\, g_{\mu\rho}g_{\nu\sigma} + C_3\, g_{\mu\sigma}g_{\nu\rho}\right),$$

$$\begin{aligned} C_1 = {} & Tr(\{T^a,T^b\}\{T^c,T^d\})\,\left(5N_c + 2\lambda_{HV}N_c + 6N_f\right) \\ & - \left(Tr(T^aT^cT^bT^d) + Tr(T^aT^dT^bT^c)\right)\left(12N_c + 4\lambda_{HV}N_c + 10N_f\right) \\ & - \left(\delta^{ab}\delta^{cd} + \delta^{ac}\delta^{bd} + \delta^{ad}\delta^{bc}\right), \quad C_2 = C_1(b \leftrightarrow c) \qquad C_3 = C_1(b \leftrightarrow d) \end{aligned}$$