LUND UNIVERSITY

ETH zürich

Simulation of BSM physics

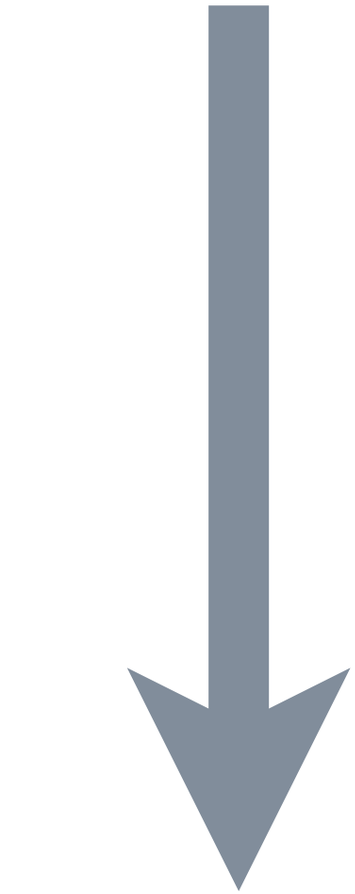
Valentin Hirschi

Lund MCnet 2017

3rd July 2017

Prediction chain

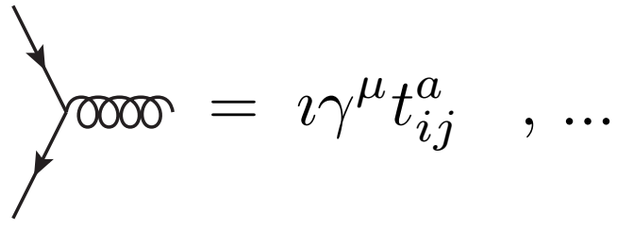
Theory



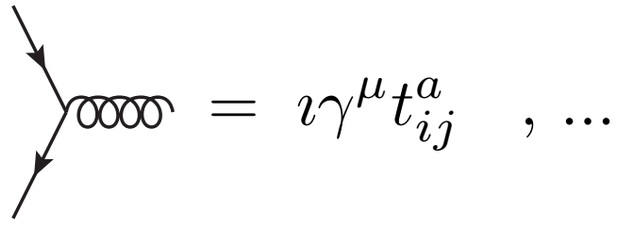
Exp. data

Prediction chain

Prediction chain

$G^{\mu\nu}G_{\mu\nu} + \imath\bar{q}_{(i)}D_\mu\gamma^\mu q_{(i)} + [\cdots]$ **Model**  $= \imath\gamma^\mu t^a_{ij}$, ...

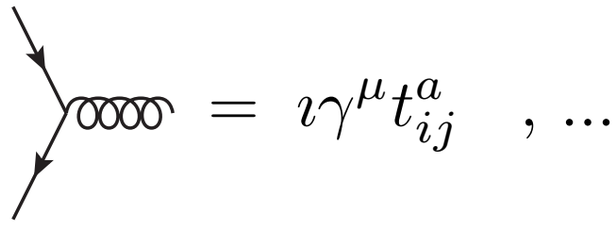
Prediction chain

$G^{\mu\nu}G_{\mu\nu} + \imath\bar{q}_{(i)}D_\mu\gamma^\mu q_{(i)} + [\cdots]$ **Model**  $= \imath\gamma^\mu t^a_{ij}$, ...

$pp \to jj \quad \mathrm{QCD} = 2$ **Matrix Element** $\mathcal{M}^2_{gg\to d\bar{d}}$, ...

Prediction chain

$G^{\mu\nu}G_{\mu\nu} + \imath\bar{q}_{(i)}D_\mu\gamma^\mu q_{(i)} + [\cdots]$ **Model**

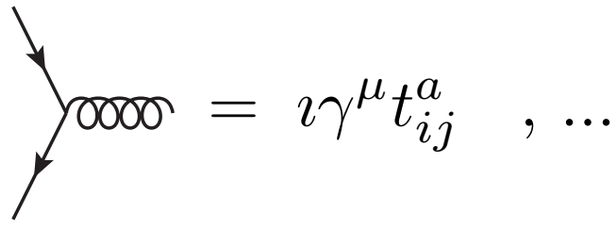


$pp \to jj \quad \mathrm{QCD} = 2$ **Matrix Element** $\mathcal{M}^2_{gg\to d\bar{d}}$, ...

matrix.f **Partonic Events**

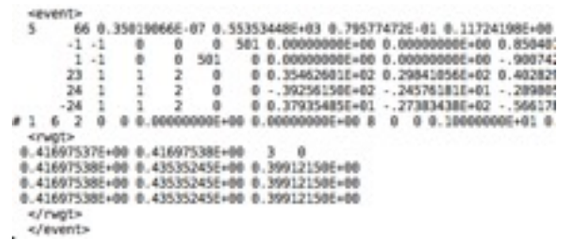
```
<event>
5      66 0.35819066E-07 0.55353448E+03 0.79577472E-01 0.11724198E+00
     -1 -1    0    0    0  501 0.00000000E+00 0.00000000E+00 0.85848
      1 -1    0    0  501    0 0.00000000E+00 0.00000000E+00 -.90874
     23  1    1    2    0    0 0.35462601E+02 0.29841056E+02 0.40282
     24  1    1    2    0    0 -.39256150E+02 -.24576181E+01 -.20980
    -24  1    1    2    0    0 0.37935485E+01 -.27383438E+02 -.56617
# 1  6  2  0  0 0.00000000E+00 0.00000000E+00 8  0  0 0.10000000E+01 0
  <rwgt>
 0.41697537E+00 0.41697538E+00   3   0
 0.41697538E+00 0.43535245E+00 0.39912150E+00
 0.41697538E+00 0.43535245E+00 0.39912150E+00
 0.41697538E+00 0.43535245E+00 0.39912150E+00
  </rwgt>
  </event>
```

Prediction chain

$G^{\mu\nu}G_{\mu\nu} + \imath\bar{q}_{(i)}D_\mu\gamma^\mu q_{(i)} + [\cdots]$ **Model**  $= \imath\gamma^\mu t^a_{ij}$, ...

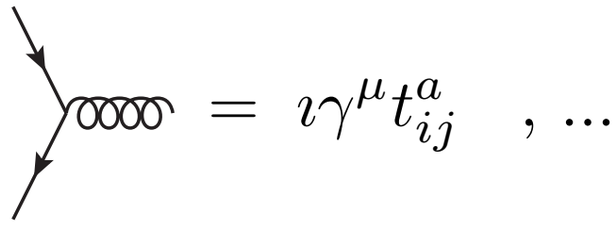
$pp \to jj$ QCD $= 2$ **Matrix Element** $\mathcal{M}^2_{gg\to d\bar{d}}$, ...

matrix.f **Partonic Events**



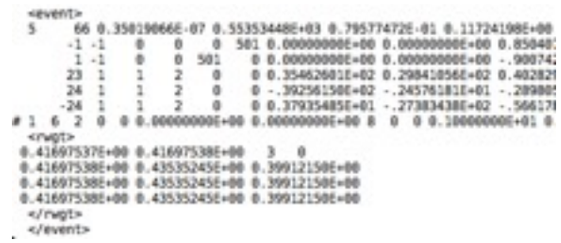
events.lhe **Hadron level** $\{\pi^0, K^+, e^+, p, \cdots\}$

Prediction chain

$G^{\mu\nu}G_{\mu\nu} + \imath\bar{q}_{(i)}D_\mu\gamma^\mu q_{(i)} + [\cdots]$ **Model**  $= \imath\gamma^\mu t^a_{ij}$, ...

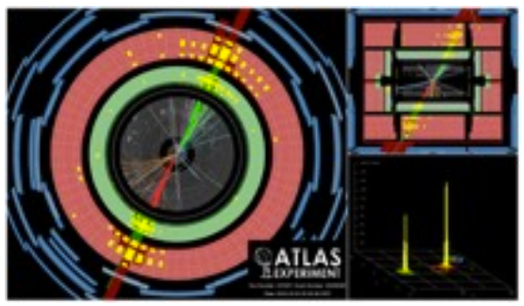
$pp \to jj$ QCD $= 2$ **Matrix Element** $\mathcal{M}^2_{gg\to d\bar{d}}$, ...

matrix.f **Partonic Events**



events.lhe **Hadron level** $\{\pi^0, K^+, e^+, p, \cdots\}$

events.hep **Detector level**

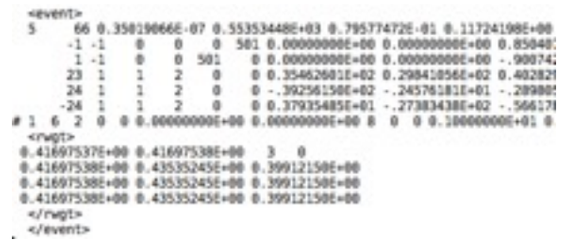


Prediction chain

FeynRules → **Model**

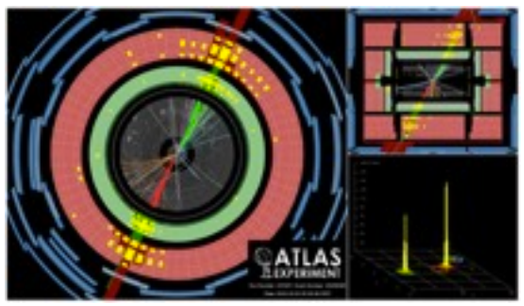
$pp \to jj \quad \mathrm{QCD} = 2$ → **Matrix Element** → $\mathcal{M}^2_{gg\to d\bar{d}}$, ...

matrix.f → **Partonic Events**



events.lhe → **Hadron level** → $\{\pi^0, K^+, e^+, p, \cdots\}$

events.hep → **Detector level**

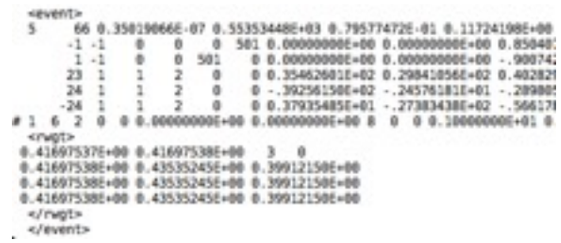


Prediction chain

FeynRules — Model

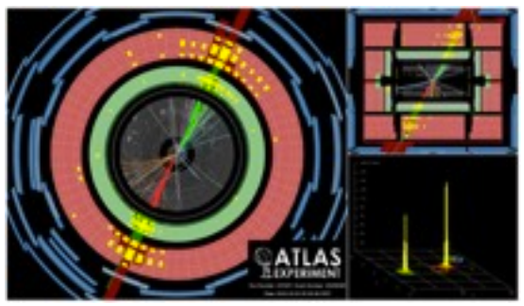
MadGraph 5 — Matrix Element

matrix.f — Partonic Events



events.lhe — Hadron level — $\{\pi^0, K^+, e^+, p, \cdots\}$

events.hep — Detector level



Prediction chain

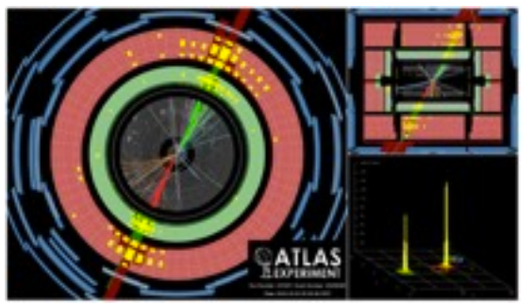
FeynRules → Model

MadGraph 5 → Matrix Element

MadEvent 5 → Partonic Events

events.lhe — Hadron level — $\{\pi^0, K^+, e^+, p, \cdots\}$

events.hep — Detector level



Prediction chain

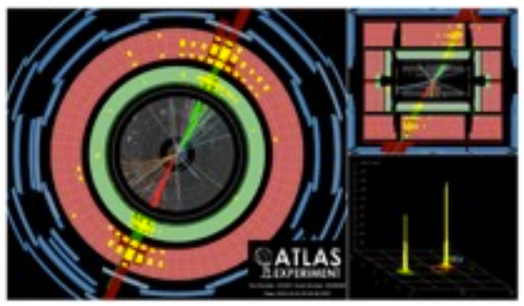
FeynRules → Model

MadGraph 5 → Matrix Element

MadEvent 5 → Partonic Events

Pythia / Herwig → Hadron level

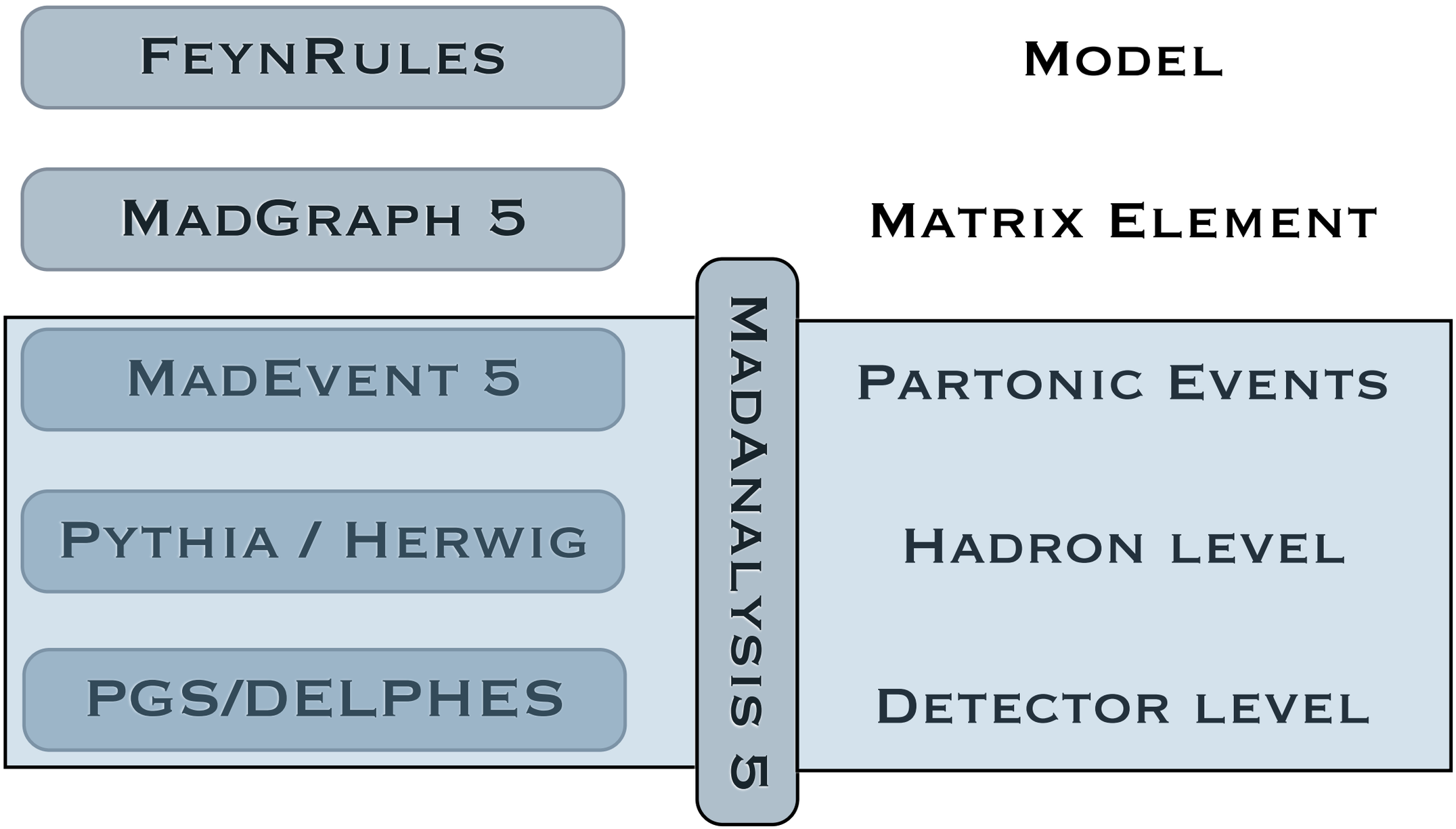
events.hep → Detector level



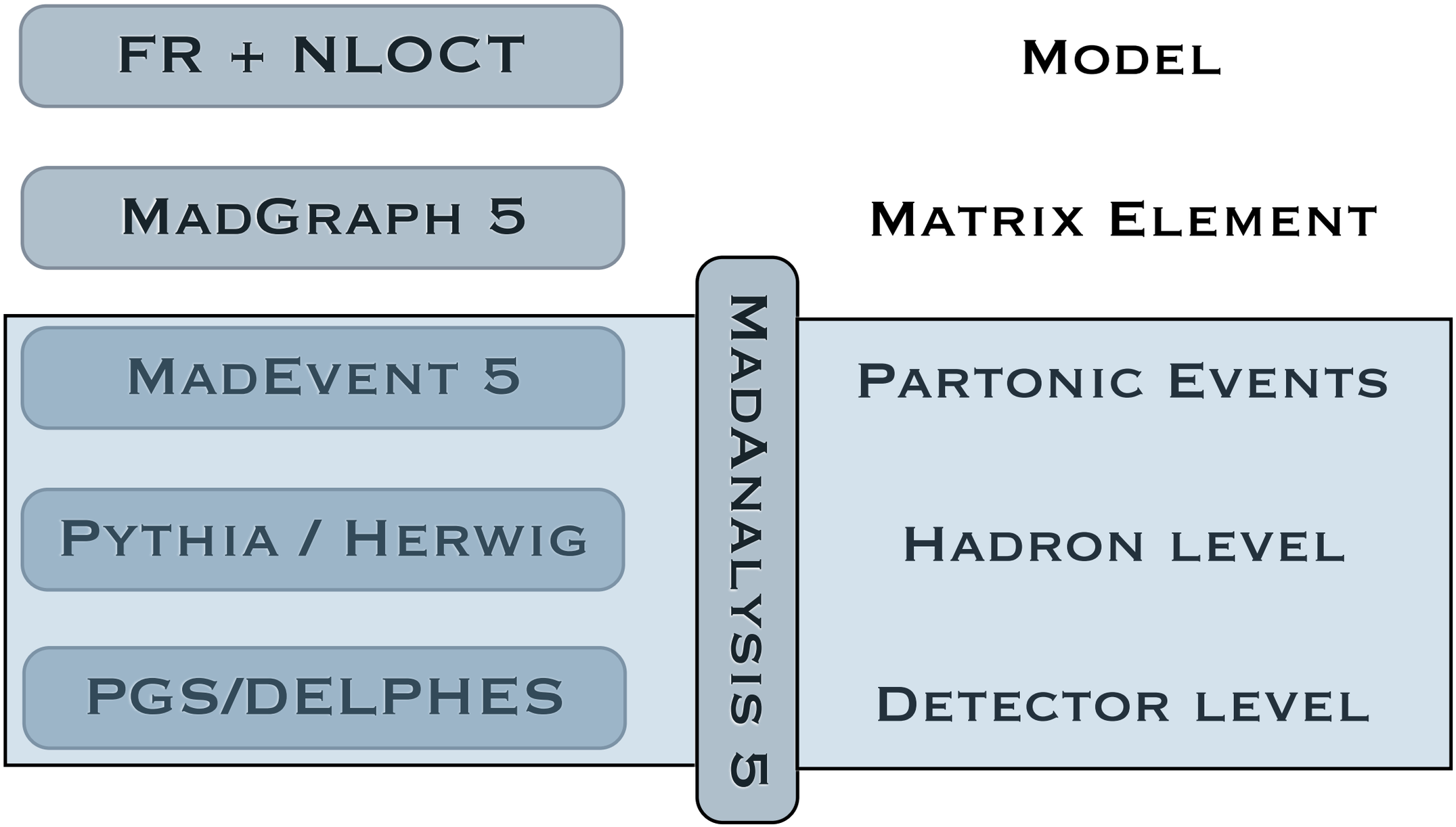
Prediction chain

FeynRules	Model
MadGraph 5	Matrix Element
MadEvent 5	Partonic Events
Pythia / Herwig	Hadron level
PGS/DELPHES	Detector level

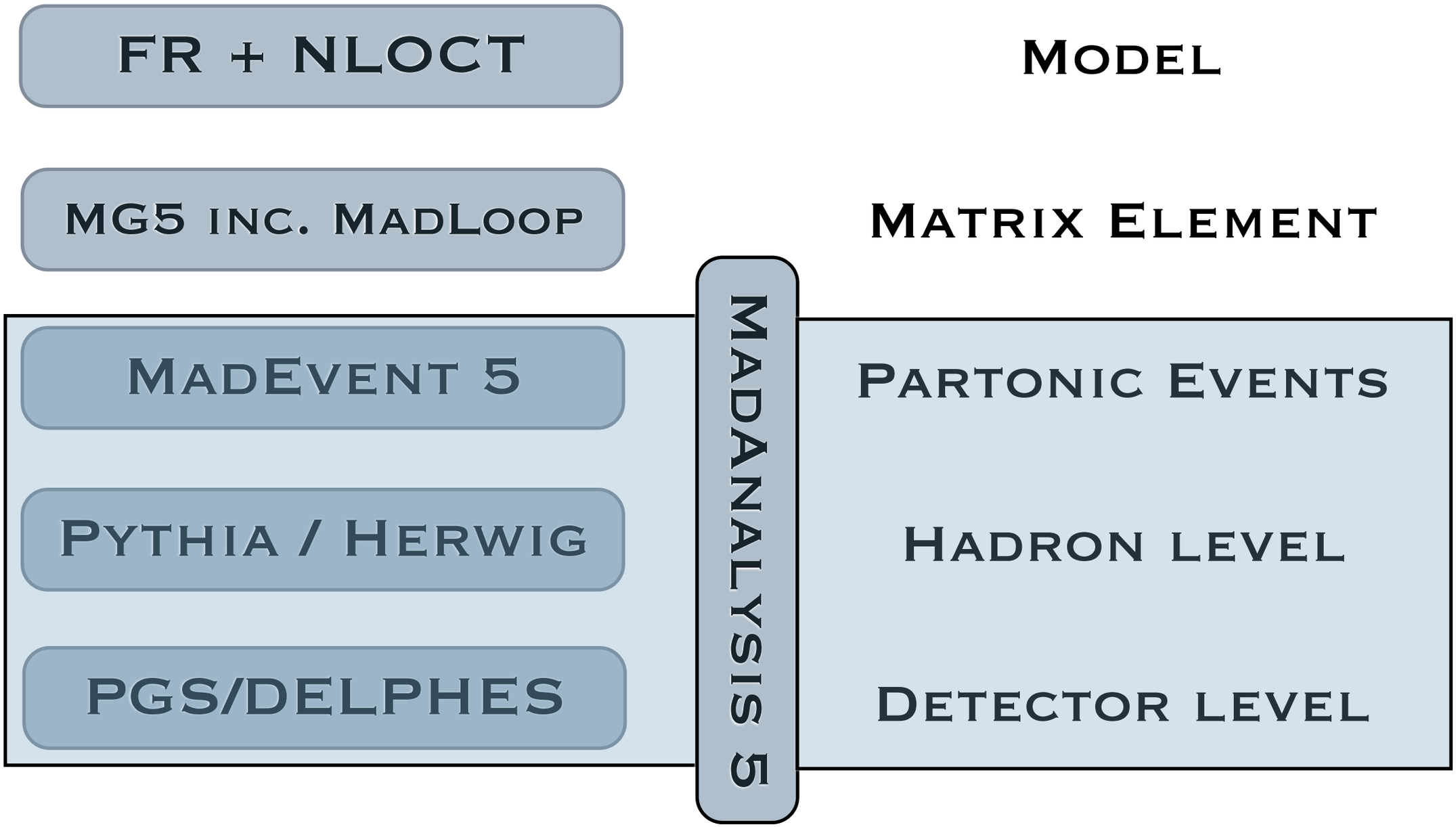
Prediction chain



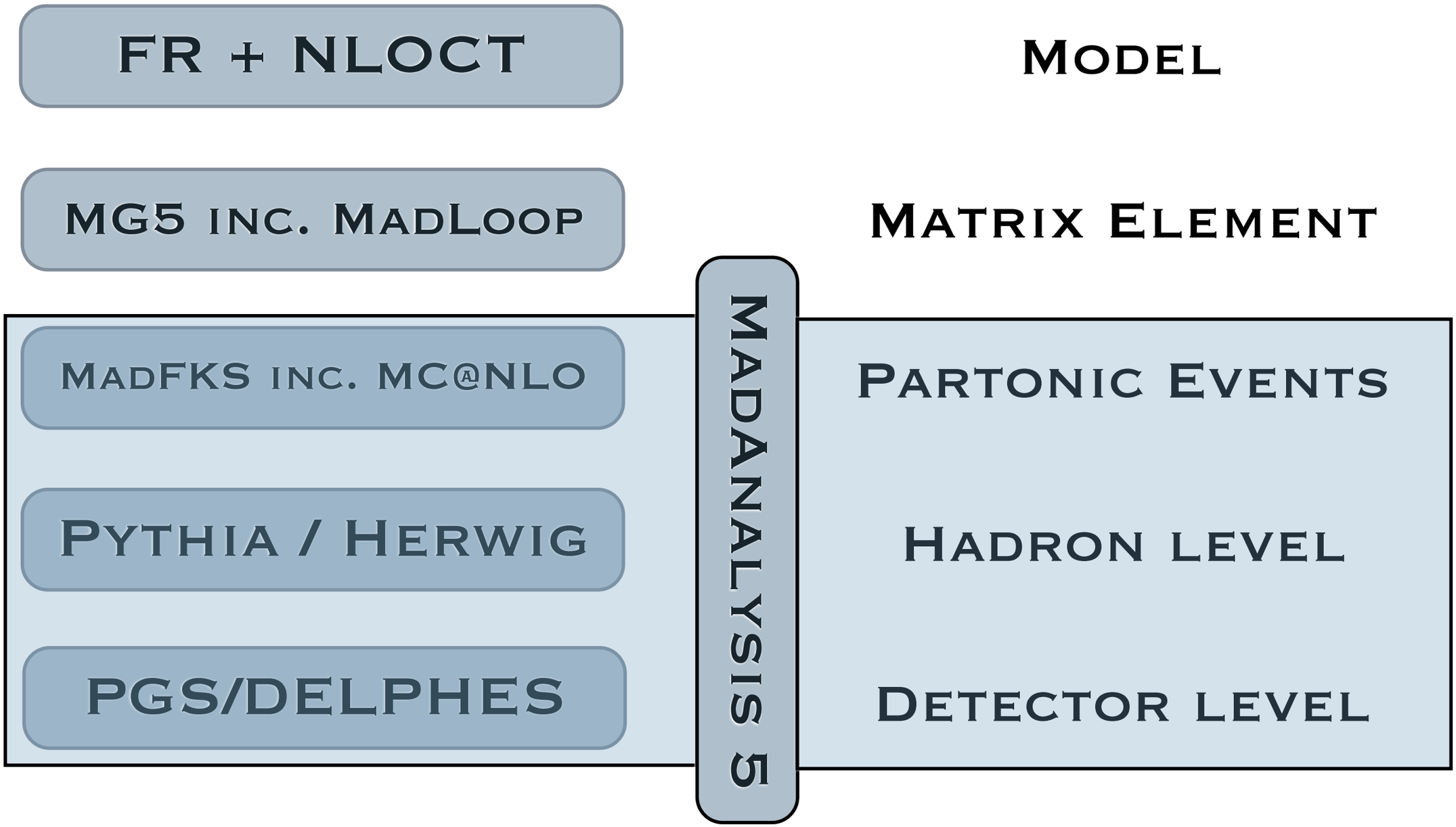
Prediction chain



Prediction chain



Prediction chain



Plan

- Build your model : FeynRules + UFO
- Computing amplitudes with Feynman diagrams
- Special techniques relevant to BSM
- Simulations at NLO accuracy
- MadLoop : evaluating loop Matrix Elements
- Recent loopy applications within MG5_aMC

FeynRules

[arXiv:1310.1921]

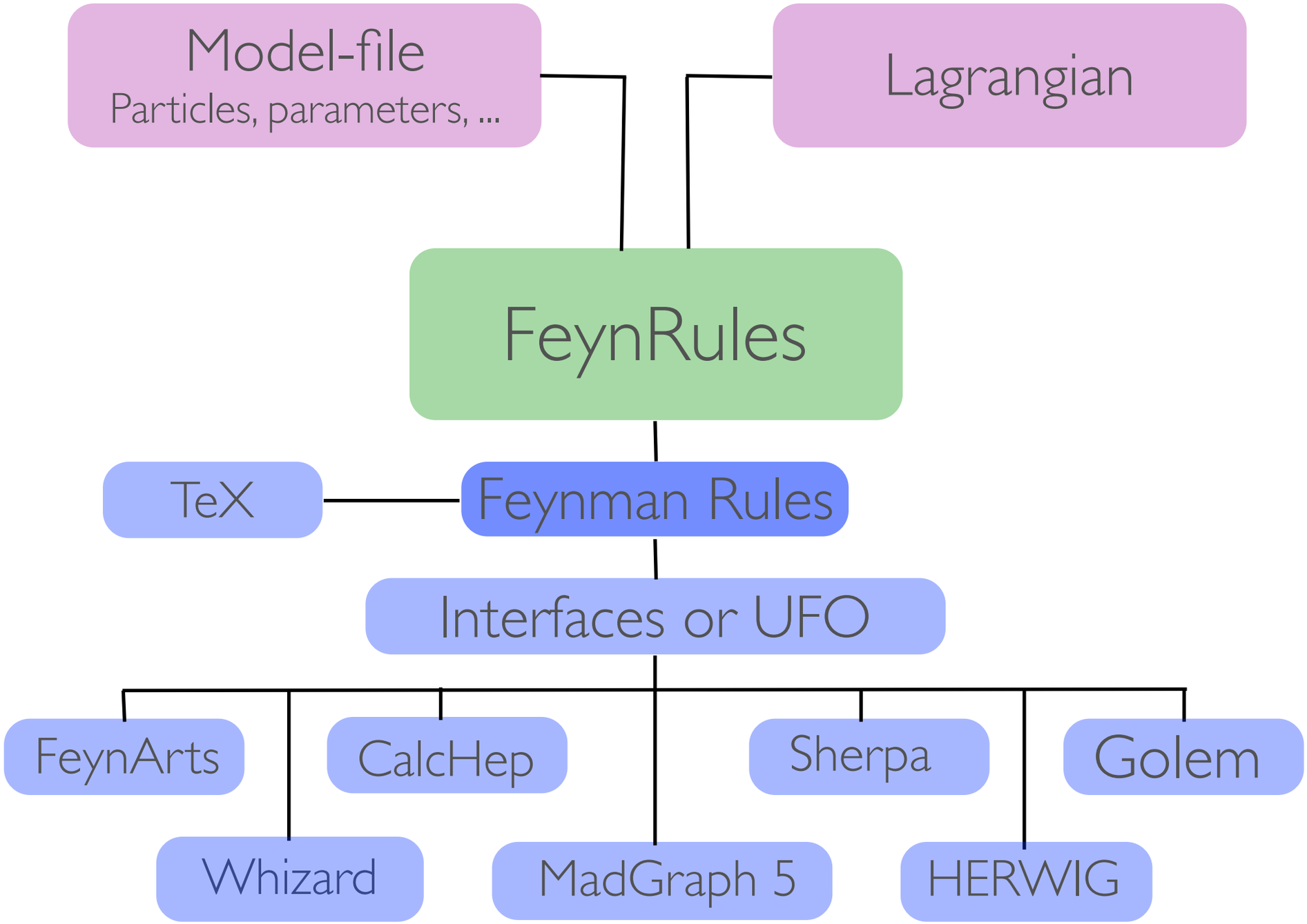
- Mathematica package to derive
  Feynman rules from a Lagrangian

- V 2.3 available at feynrules.irmp.ucl.ac.be

- Requirements:
  - All indices must be contracted (Lorentz + gauge invariance)
  - Locality (no interactions non-polynomial in the fields)
  - All spins supported up to 2, as well as anticommutating ghosts
  - Chiral and vector superfields are also supported
  - Fields can be massive, massless, self-conjugated or not.

FeynRules Structure

[Alloul, Christensen, Degrande, Duhr, Fuks]



How to write your model

In the Standard Model 'SM.fr' file, you will find....

Gauge symmetries Particles Parameters Lagrangian

```
M$GaugeGroups = {
  U1Y  == {
    Abelian          -> True,
    CouplingConstant -> g1,
    GaugeBoson       -> B,
    Charge           -> Y
  } [...]
```

How to write your model

In the Standard Model 'SM.fr' file, you will find....

Gauge symmetries Particles Parameters Lagrangian

```
SU2L == {
  Abelian             -> False,
  CouplingConstant    -> gw,
  GaugeBoson          -> Wi,
  StructureConstant -> Eps,
  Representations     -> {Ta,SU2D},
  Definitions         -> {Ta[a_,b_,c_]->PauliSigma[a,b,c]/2,
                          FSU2L[i_,j_,k_]:> I Eps[i,j,k]}
} [...]
```

How to write your model

In the Standard Model 'SM.fr' file, you will find....

Gauge symmetries Particles Parameters Lagrangian

```
M$GaugeGroups = {
  U1Y  == {
    Abelian          -> True,
    C
    G
    C
  } [
```

```
  SU2L == {
    Abelian          -> False,
    CouplingConstant -> gw,
    GaugeBoson       -> Wi,
    StructureConstant -> Eps,
    Representations  -> {Ta,SU2D},
    Definitions      -> {Ta[a_,
                         FSU2L
} [...]
```

```
SU3C == {
  Abelian           -> False,
  CouplingConstant  -> gs,
  GaugeBoson        -> G,
  StructureConstant -> f,
  Representations   -> {T,Colour},
  SymmetricTensor   -> dSUN
} [...]
```

How to write your model

In the Standard Model 'SM.fr' file, you will find....

Gauge symmetries **Particles** Parameters Lagrangian

```
M$ClassesDescription = {
  V[1] == {
    ClassName       -> A,
    SelfConjugate   -> True,
    Mass            -> 0,
    Width           -> 0,
    ParticleName    -> "a",
    PDG             -> 22,
    PropagatorLabel -> "a",
    PropagatorType  -> W,
    PropagatorArrow -> None,
    FullName        -> "Photon"
  },[...]
```

How to write your model

In the Standard Model 'SM.fr' file, you will find....

Gauge symmetries **Particles** Parameters Lagrangian

```
M$ClassesDescription = {
  V[1] == {
    ClassName        -> A,
    SelfConju
    Mass
    Width
    ParticleN
    PDG
    Propagato
    Propagato
    Propagato
    FullName
  },[...]
```

```
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"
},[...]
```

How to write your model

In the Standard Model 'SM.fr' file, you will find....

Gauge symmetries **Particles** Parameters Lagrangian

```
F[1] == {
  ClassName        -> vl,
  ClassMembers     -> {ve,vm,vt},
  Indices          -> {Index[Generation]},
  FlavorIndex      -> Generation,
  SelfConjugate    -> False,
  Mass             -> 0,
  Width            -> 0,
  QuantumNumbers   -> {LeptonNumber -> 1},
  PropagatorLabel  -> {"v", "ve", "vm", "vt"} ,
  PropagatorType   -> S,
  PropagatorArrow  -> Forward,
  PDG              -> {12,14,16},
  ParticleName     -> {"ve","vm","vt"},
  AntiParticleName -> {"ve~","vm~","vt~"},
  FullName         -> {"Electron-neutrino", "Mu-neutrino",
                       "Tau-neutrino"}
},[...]
```

How to write your model

In the Standard Model 'SM.fr' file, you will find....

Gauge symmetries Particles **Parameters** Lagrangian

```
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 pole"
  },
  gs == {
    ParameterType    -> Internal,
    Value            -> Sqrt[4 Pi aS],
    InteractionOrder -> {QCD,1},
    TeX              -> Subscript[g,s],
    ParameterName    -> G,
    Description      -> "Strong coupling constant at the Z pole"
  },[...]
```

How to write your model

In the Standard Model 'SM.fr' file, you will find....

Gauge symmetries Particles **Parameters** Lagrangian

```
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 pole"
  },
  gs == {
    ParameterType    -> Internal,
    Value            -> Sqrt[4 Pi aS],
    InteractionOrder -> {QCD,1},
    TeX              -> Subscript[g,s],
    ParameterName    -> G,
    Description      -> "Strong coupling constant at the Z pole"
  },[...]
```

Will appear in the `param_card.dat`

How to write your model

In the Standard Model 'SM.fr' file, you will find....

Gauge symmetries Particles **Parameters** Lagrangian

```
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 pole"
  },
  gs == {
    ParameterType    -> Internal,
    Value            -> Sqrt[4 Pi aS],
    InteractionOrder -> {QCD,1},
    TeX              -> Subscript[g,s],
    ParameterName    -> G,
    Description      -> "Strong coupling constant at the Z pole"
  },[...]
```

Will appear in the `param_card.dat`

Expressed in terms of external parameters

How to write your model

In the Standard Model 'SM.fr' file, you will find....

Gauge symmetries Particles **Parameters** Lagrangian

```
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 pole"
  },
  gs == {
    ParameterType    -> Internal,
    Value            -> Sqrt[4 Pi aS],
    InteractionOrder -> {QCD,1},
    TeX              -> Subscript[g,s],
    ParameterName    -> G,
    Description      -> "Strong coupling constant at the Z pole"
  },[...]
```

Will appear in the `param_card.dat`

Expressed in terms of external parameters

```
CKM == {
  ParameterType -> Internal,
  Indices       -> {Index[Generation], Index[Generation]},
  Unitary       -> True,
  Value         -> {CKM[1,1] -> Cos[cabi],  CKM[1,2] -> Sin[cabi], CKM[1,3] -> 0,
                    CKM[2,1] -> -Sin[cabi], CKM[2,2] -> Cos[cabi], CKM[2,3] -> 0,
                    CKM[3,1] -> 0,          CKM[3,2] -> 0,         CKM[3,3] -> 1},
  TeX          -> Superscript[V,CKM],
  Description -> "CKM-Matrix"}
```

How to write your model

In the Standard Model 'SM.fr' file, you will find....

Gauge symmetries Particles Parameters **Lagrangian**

```
LGauge := Block[{mu,nu,ii,aa},
  ExpandIndices[-1/4 FS[B,mu,nu] FS[B,mu,nu]
                 -1/4 FS[Wi,mu,nu,ii] FS[Wi,mu,nu,ii]
                 -1/4 FS[G,mu,nu,aa] FS[G,mu,nu,aa],
                                    FlavorExpand->SU2W]];
```

$$\mathcal{L} \supset -\frac{1}{4} F^{\mu\nu} F_{\mu\nu} - \frac{1}{4} W_i^{\mu\nu} W_{\mu\nu,i} - \frac{1}{4} C_a^{\mu\nu} G_{\mu\nu,a}$$

How to write your model

In the Standard Model 'SM.fr' file, you will find....

Gauge symmetries Particles Parameters **Lagrangian**

```
LFermions := Block[{mu},
  ExpandIndices[I*(
    QLbar.Ga[mu].DC[QL, mu] + LLbar.Ga[mu].DC[LL, mu] +
    uRbar.Ga[mu].DC[uR, mu] + dRbar.Ga[mu].DC[dR, mu] +
    lRbar.Ga[mu].DC[lR, mu]), FlavorExpand->{SU2W,SU2D}]
    /.{CKM[a_,b_] Conjugate[CKM[a_,c_]]->IndexDelta[b,c],
       CKM[b_,a_] Conjugate[CKM[c_,a_]]->IndexDelta[b,c]}];
```

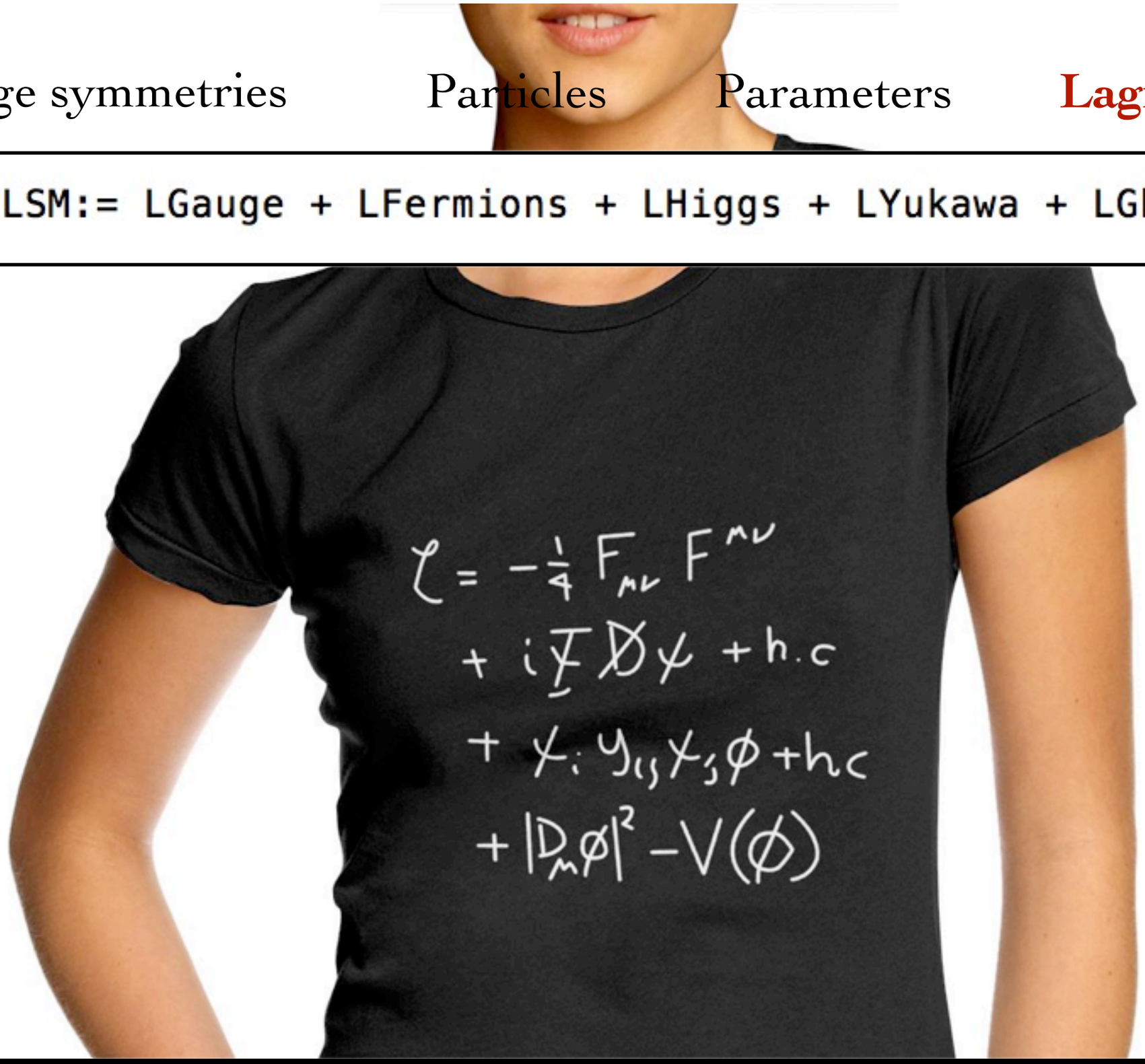
$$\mathcal{L} \supset \sum_f \left[\overline{Q}_{i,f} \not{D} Q_{i,f} + \overline{L}_{i,f} \not{D} L_{i,f} + \overline{q}_{R,f} \not{D} q_{R,f} + \overline{l}_{R,f} \not{D} l_{R,f} \right]$$

How to write your model

In the Standard Model 'SM.fr' file, you will find....

Gauge symmetries Particles Parameters **Lagrangian**

```
LSM:= LGauge + LFermions + LHiggs + LYukawa + LGhost;
```



How to write your model

In the Standard Model 'SM.fr' file, you will find....

Gauge symmetries Particles Parameters **Lagrangian**

```
LSM:= LGauge + LFermions + LHiggs + LYukawa + LGhost;
```

How to write your model

Now what can you do with it?

```
CheckHermicity[LSM]

CheckDiagonalMassTerm[LSM]

CheckKineticTermNormalisation[LSM]

...

FeynmanRules[LSM]
```

How to write your model

Now what can you do with it?

```
In[18]:= FeynmanGauge = True;
```

- **Gauge sector**

```
In[21]:= vertsGauge = FeynmanRules[LGauge, FlavorExpand → SU2W, ScreenOutput → True];
```

Starting Feynman rule calculation.

Expanding the Lagrangian...

Collecting the different structures that enter the vertex.

8 possible non−zero vertices have been found −> starting the computation: 8 / 8.

8 vertices obtained.

(* *)

Vertex 1

Particle 1 : Vector , G

Particle 2 : Vector , G

Particle 3 : Vector , G

Vertex:

$$g_s\, p_1^{\mu_3}\, f_{a_1,a_2,a_3}\, \eta_{\mu_1,\mu_2} - g_s\, p_2^{\mu_3}\, f_{a_1,a_2,a_3}\, \eta_{\mu_1,\mu_2} - g_s\, p_1^{\mu_2}\, f_{a_1,a_2,a_3}\, \eta_{\mu_1,\mu_3} + g_s\, p_3^{\mu_2}\, f_{a_1,a_2,a_3}\, \eta_{\mu_1,\mu_3} + g_s\, p_2^{\mu_1}\, f_{a_1,a_2,a_3}\, \eta_{\mu_2,\mu_3} - g_s\, p_3^{\mu_1}\, f_{a_1,a_2,a_3}\, \eta_{\mu_2,\mu_3}$$

(* *)

Vertex 2

[...]

How to write your model

Export it to event generators !

`WriteFeynArtsOutput[LSM]` →

FeynArts 3.9 User's Guide

`WriteSHOutput[LSM]` →

Sherpa

`WriteWOOutput[LSM]` →

Whizard

...

How to write your model

Export it to event generators !

WriteFeynArtsOutput[LSM] →

FeynArts 3.9 User's Guide

WriteSHOutput[LSM] →

Sherpa

WriteWOOutput[LSM] →

Whizard

...

WriteUFO[LSM] →

UFO becoming the standard

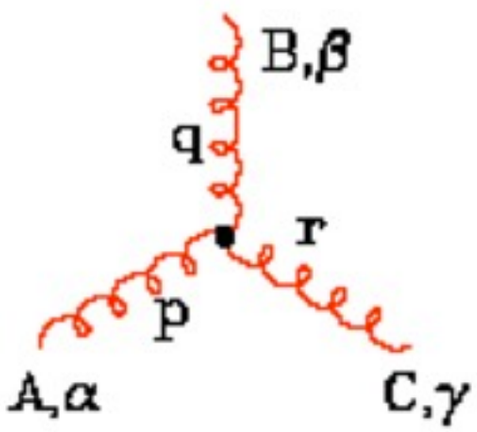
Universal FeynRules Output

[arXiv:1108.2040]

- Python module to specify a model for MC generators

`vertices.py` `couplings.py` `parameters.py` `lorentz.py`

```
V_36 = Vertex(name = 'V_36',
              particles = [ P.g, P.g, P.g ],
              color = [ 'f(1,2,3)' ],
              lorentz = [ L.VVV1 ],
              couplings = {(0,0):C.GC_10})
```



$$-g\ f^{ABC}[(p-q)^{\gamma}g^{\alpha\beta}+(q-r)^{\alpha}g^{\beta\gamma}+(r-p)^{\beta}g^{\gamma\alpha}]$$

(all momenta incoming)

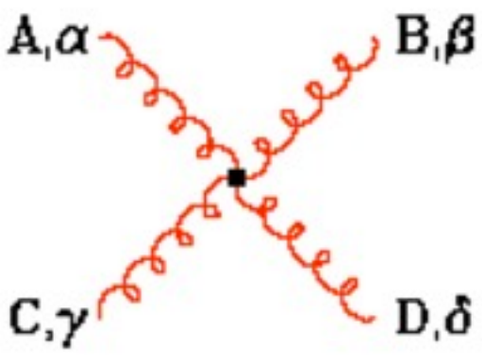
Universal FeynRules Output

[arXiv:1108.2040]

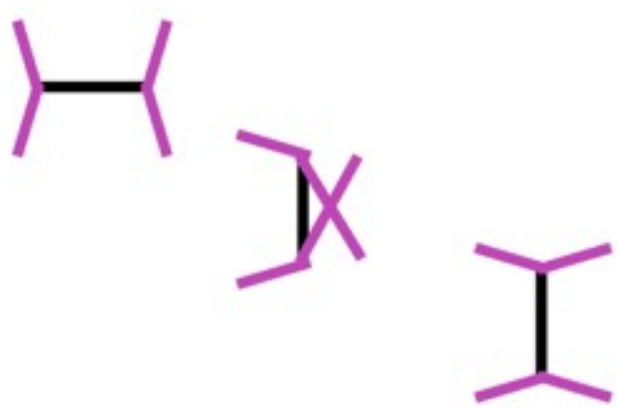
- Python module to specify a model for MC generators

`vertices.py` `couplings.py` `parameters.py` `lorentz.py`

```
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.VVVV1, L.VVVV3, L.VVVV4 ],
couplings = {(1,1):C.GC_12,
             (0,0):C.GC_12,
             (2,2):C.GC_12})
```



$$-ig^2 f^{XAC} f^{XBD} [g^{\alpha\beta} g^{\gamma\delta} - g^{\alpha\delta} g^{\beta\gamma}]$$
$$-ig^2 f^{XAD} f^{XBC} [g^{\alpha\beta} g^{\gamma\delta} - g^{\alpha\gamma} g^{\beta\delta}]$$
$$-ig^2 f^{XAB} f^{XCD} [g^{\alpha\gamma} g^{\beta\delta} - g^{\alpha\delta} g^{\beta\gamma}]$$



Universal FeynRules Output

[arXiv:1108.2040]

vertices.py **couplings.py** parameters.py lorentz.py

```
GC_12 = Coupling(name = 'GC_12',
                 value = 'complex(0,1)*G**2',
                 order = {'QCD':2})
```

coupling_orders.py

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

QED = CouplingOrder(name = 'QED',
                    expansion_order = 99,
                    hierarchy = 2)
```

Universal FeynRules Output

[arXiv:1108.2040]

vertices.py couplings.py **parameters.py** lorentz.py

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

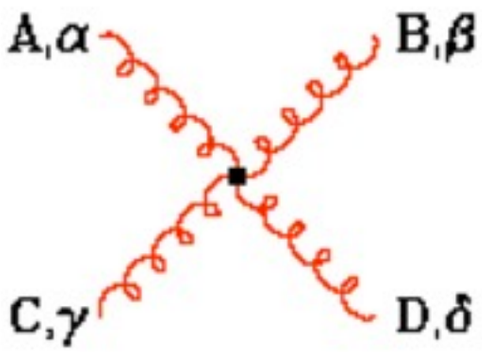
```
aS = Parameter(name = 'aS',
               nature = 'external',
               type = 'real',
               value = 0.118,
               texname = '\\alpha _s',
               lhablock = 'SMINPUTS',
               lhacode = [ 3 ])
```

Universal FeynRules Output

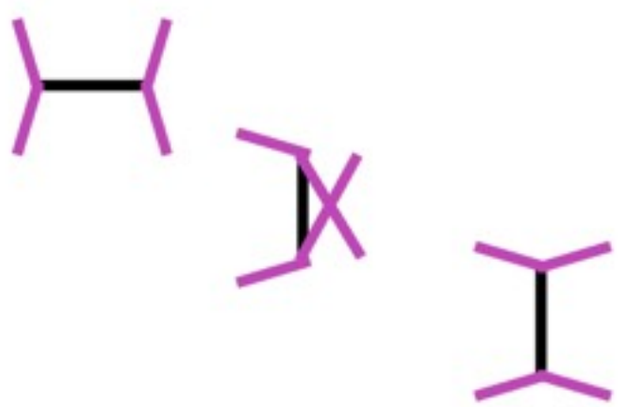
[arXiv:1108.2040]

vertices.py couplings.py parameters.py **lorentz.py**

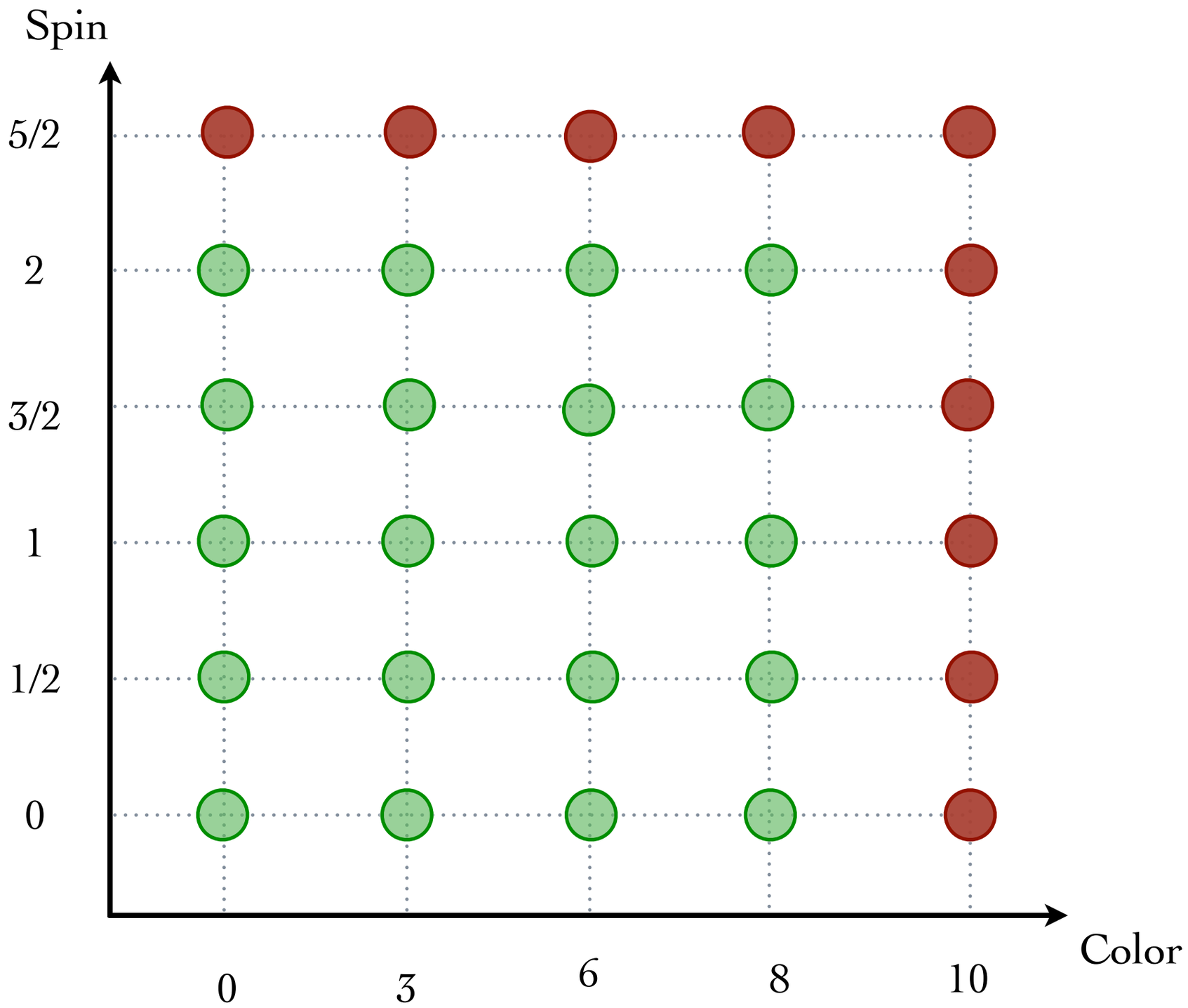
```
VVVV1 = Lorentz(name = 'VVVV1',
                spins = [ 3, 3, 3, 3 ],
                structure = 'Metric(1,4)*Metric(2,3) - Metric(1,3)*Metric(2,4)')

VVVV3 = Lorentz(name = 'VVVV3',
                spins = [ 3, 3, 3, 3 ],
                structure = 'Metric(1,4)*Metric(2,3) - Metric(1,2)*Metric(3,4)')

VVVV4 = Lorentz(name = 'VVVV4',
                spins = [ 3, 3, 3, 3 ],
                structure = 'Metric(1,3)*Metric(2,4) - Metric(1,2)*Metric(3,4)')
```



$$-ig^2 f^{XAC} f^{XBD} [g^{\alpha\beta} g^{\gamma\delta} - g^{\alpha\delta} g^{\beta\gamma}]$$
$$-ig^2 f^{XAD} f^{XBC} [g^{\alpha\beta} g^{\gamma\delta} - g^{\alpha\gamma} g^{\beta\delta}]$$
$$-ig^2 f^{XAB} f^{XCD} [g^{\alpha\gamma} g^{\beta\delta} - g^{\alpha\delta} g^{\beta\gamma}]$$



BSM ZOO



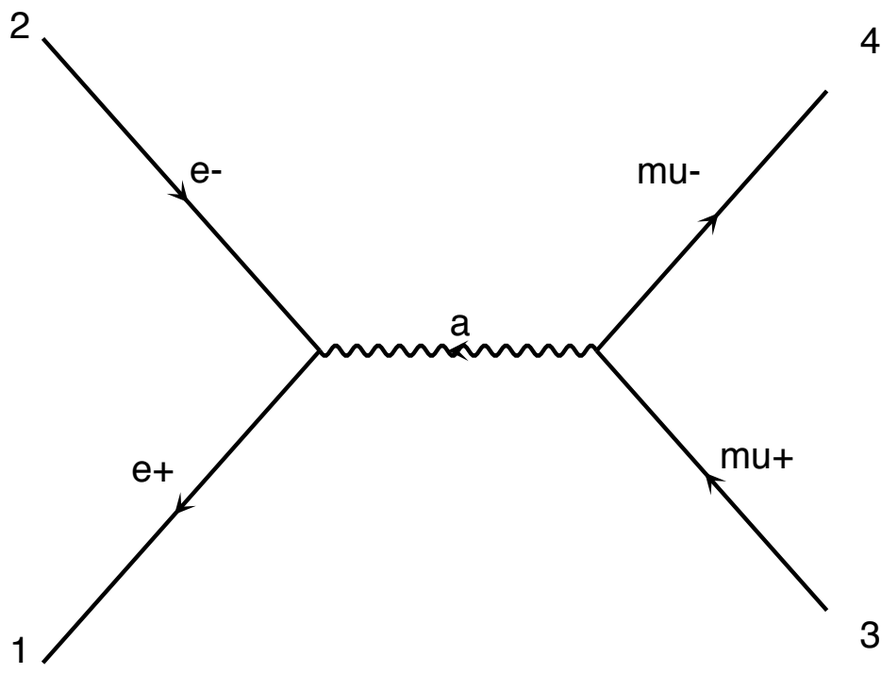
Plan

- Build your model : FeynRules + UFO
- Computing amplitudes with Feynman diagrams
- Special techniques relevant to BSM
- Simulations at NLO accuracy
- MadLoop : evaluating loop Matrix Elements
- Recent loopy applications within MG5_aMC

Plan

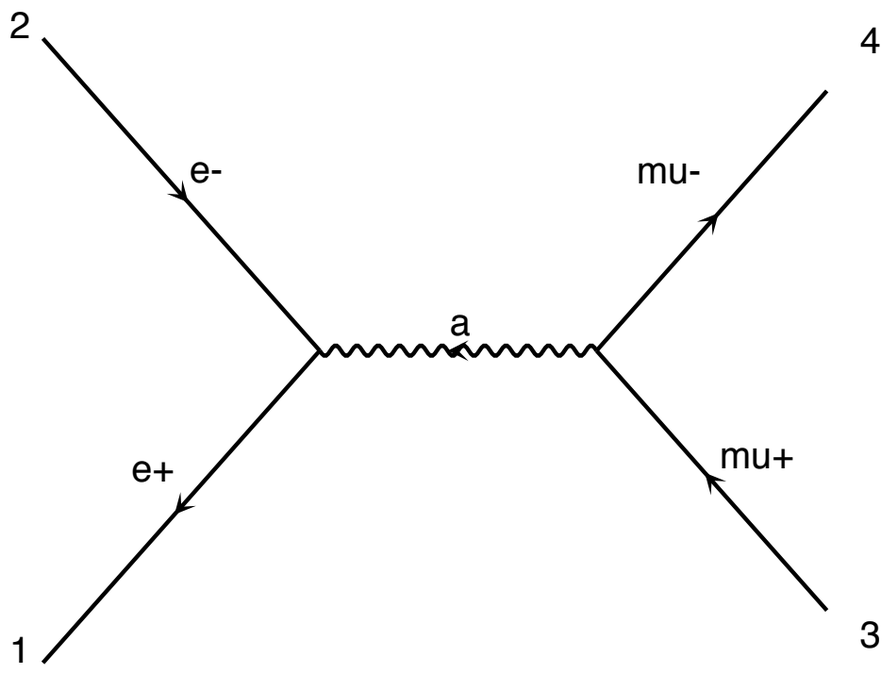
- Build your model : FeynRules + UFO
- Computing amplitudes with Feynman diagrams
- Special techniques relevant to BSM
- Simulations at NLO accuracy
- MadLoop : evaluating loop Matrix Elements
- Recent loopy applications within MG5_aMC

Matrix element computations



$$\mathcal{M} = e^2(\bar{u}\gamma^\mu v)\frac{g_{\mu\nu}}{q^2}(\bar{v}\gamma^\nu u)$$

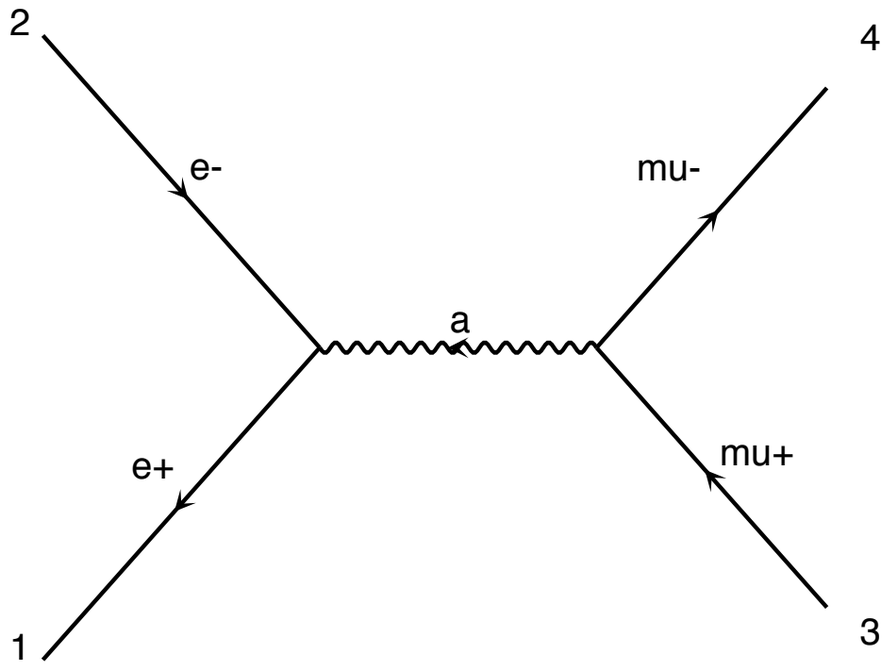
Matrix element computations



$$\mathcal{M} = e^2(\bar{u}\gamma^\mu v)\frac{g_{\mu\nu}}{q^2}(\bar{v}\gamma^\nu u)$$

$$\frac{1}{4}\sum_{pol}|\mathcal{M}|^2 = \frac{1}{4}\sum_{pol}\mathcal{M}^*\mathcal{M}$$

Matrix element computations

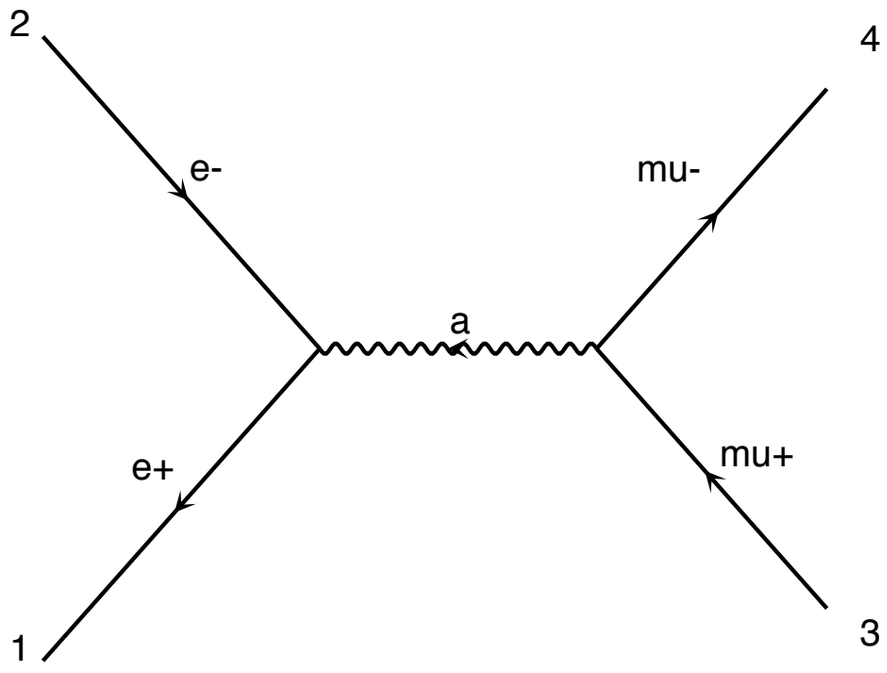


$$\mathcal{M} = e^2(\bar{u}\gamma^\mu v)\frac{g_{\mu\nu}}{q^2}(\bar{v}\gamma^\nu u)$$

$$\frac{1}{4}\sum_{pol}|\mathcal{M}|^2 = \frac{1}{4}\sum_{pol}\mathcal{M}^*\mathcal{M}$$

$$\sum_{pol}\bar{u}u = \not{p} + m$$

Matrix element computations



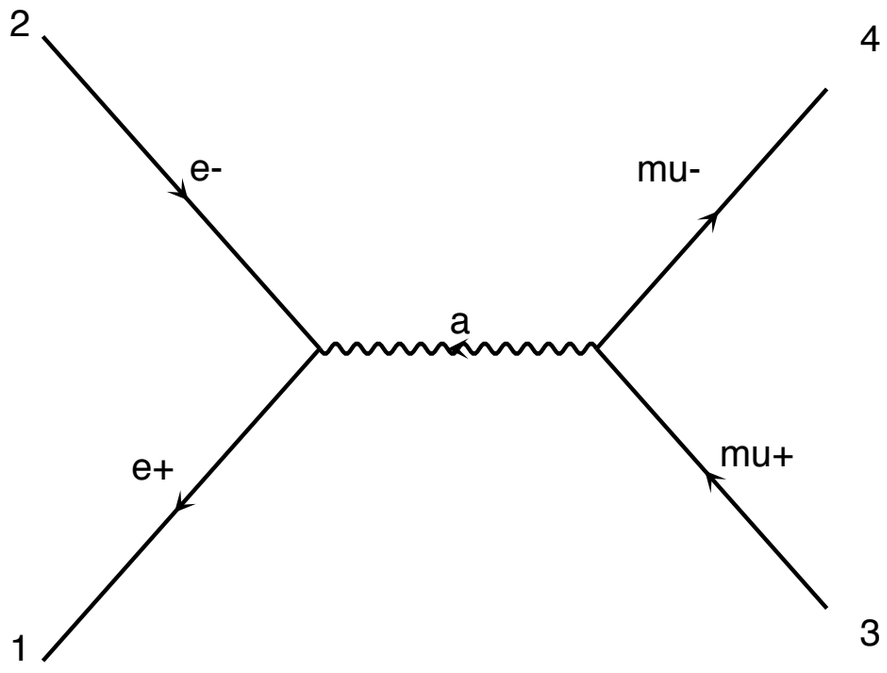
$$\mathcal{M} = e^2(\bar{u}\gamma^\mu v)\frac{g_{\mu\nu}}{q^2}(\bar{v}\gamma^\nu u)$$

$$\frac{1}{4}\sum_{pol}|\mathcal{M}|^2 = \frac{1}{4}\sum_{pol}\mathcal{M}^*\mathcal{M}$$

$$\sum_{pol}\bar{u}u = \not{p} + m$$

$$\rightarrow \quad \frac{e^4}{4q^4}Tr[\not{p}_1\gamma^\mu\not{p}_2\gamma^\nu]Tr[\not{p}_3\gamma_\mu\not{p}_4\gamma_\nu]$$

Matrix element computations



$$\mathcal{M} = e^2(\bar{u}\gamma^\mu v)\frac{g_{\mu\nu}}{q^2}(\bar{v}\gamma^\nu u)$$

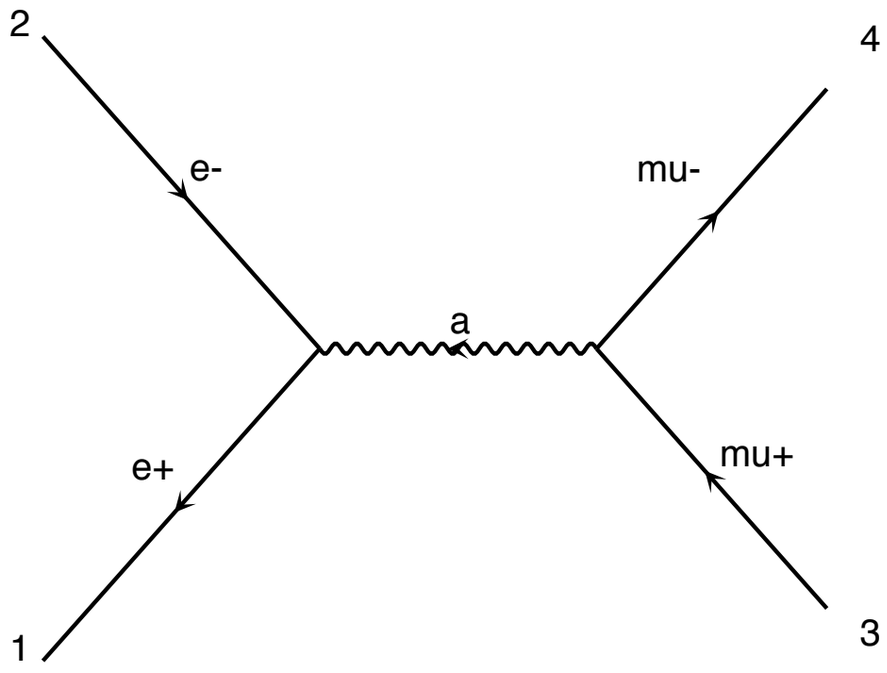
$$\frac{1}{4}\sum_{pol}|\mathcal{M}|^2 = \frac{1}{4}\sum_{pol}\mathcal{M}^*\mathcal{M}$$

$$\sum_{pol}\bar{u}u = \not{p} + m$$

$$\rightarrow \quad \frac{e^4}{4q^4}Tr[\not{p}_1\gamma^\mu\not{p}_2\gamma^\nu]Tr[\not{p}_3\gamma_\mu\not{p}_4\gamma_\nu]$$

$$\rightarrow \quad \frac{8e^4}{q^4}\left[(p_1.p_3)(p_2.p_4) + (p_1.p_4)(p_2.p_3)\right]$$

Matrix element computations



$$\mathcal{M} = e^2(\bar{u}\gamma^\mu v)\frac{g_{\mu\nu}}{q^2}(\bar{v}\gamma^\nu u)$$

$$\frac{1}{4}\sum_{pol}|\mathcal{M}|^2 = \frac{1}{4}\sum_{pol}\mathcal{M}^*\mathcal{M}$$

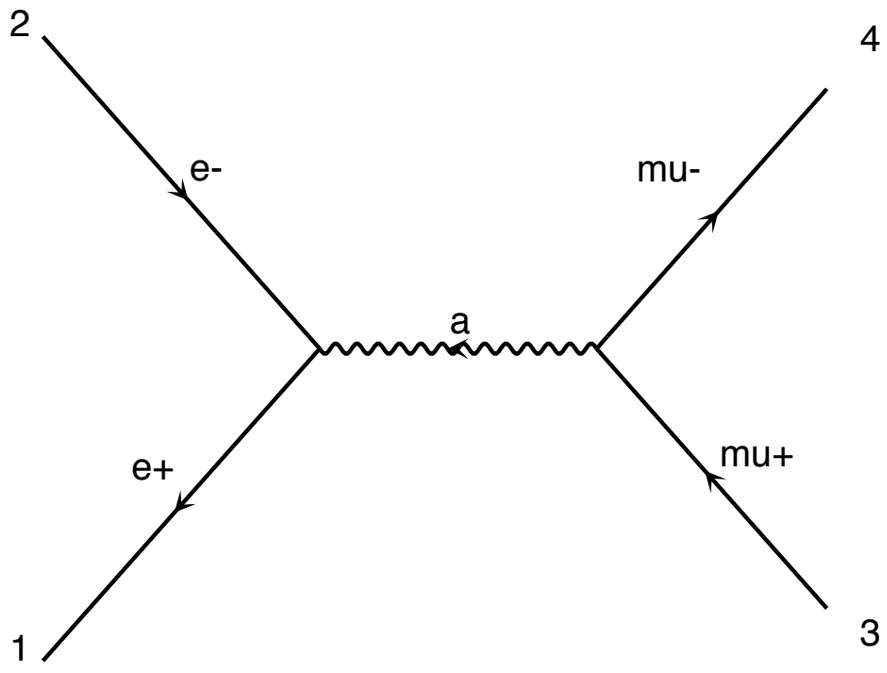
$$\sum_{pol}\bar{u}u = \not{p} + m$$

$$\rightarrow \quad \frac{e^4}{4q^4}Tr[\not{p}_1\gamma^\mu\not{p}_2\gamma^\nu]Tr[\not{p}_3\gamma_\mu\not{p}_4\gamma_\nu]$$

$$\rightarrow \quad \frac{8e^4}{q^4}\left[(p_1.p_3)(p_2.p_4) + (p_1.p_4)(p_2.p_3)\right]$$

Very Efficient !!!

Matrix element computations



$$\mathcal{M} = e^2(\bar{u}\gamma^\mu v)\frac{g_{\mu\nu}}{q^2}(\bar{v}\gamma^\nu u)$$

$$\frac{1}{4}\sum_{pol}|\mathcal{M}|^2 = \frac{1}{4}\sum_{pol}\mathcal{M}^*\mathcal{M}$$

$$\sum_{pol}\bar{u}u = \not{p} + m$$

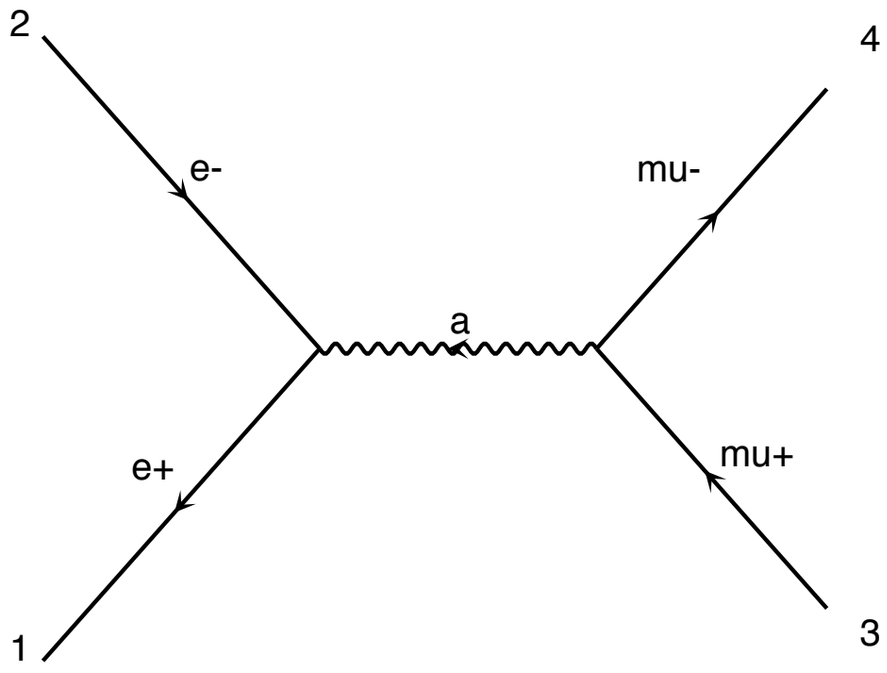
$$\rightarrow \quad \frac{e^4}{4q^4}Tr[\not{p}_1\gamma^\mu\not{p}_2\gamma^\nu]Tr[\not{p}_3\gamma_\mu\not{p}_4\gamma_\nu]$$

$$\rightarrow \quad \frac{8e^4}{q^4}\left[(p_1.p_3)(p_2.p_4) + (p_1.p_4)(p_2.p_3)\right]$$

Very Efficient !!!

Only for 2→2 and 2→3

Matrix element computations



$$\mathcal{M} = e^2(\bar{u}\gamma^\mu v)\frac{g_{\mu\nu}}{q^2}(\bar{v}\gamma^\nu u)$$

$$\frac{1}{4}\sum_{pol}|\mathcal{M}|^2 = \frac{1}{4}\sum_{pol}\mathcal{M}^*\mathcal{M}$$

$$\sum_{pol}\bar{u}u = \not{p} + m$$

$$\rightarrow \quad \frac{e^4}{4q^4}Tr[\not{p}_1\gamma^\mu\not{p}_2\gamma^\nu]Tr[\not{p}_3\gamma_\mu\not{p}_4\gamma_\nu]$$

$$\rightarrow \quad \frac{8e^4}{q^4}\left[(p_1.p_3)(p_2.p_4) + (p_1.p_4)(p_2.p_3)\right]$$

Very Efficient !!!

Only for 2→2 and 2→3

Because the number of terms rises as N^2

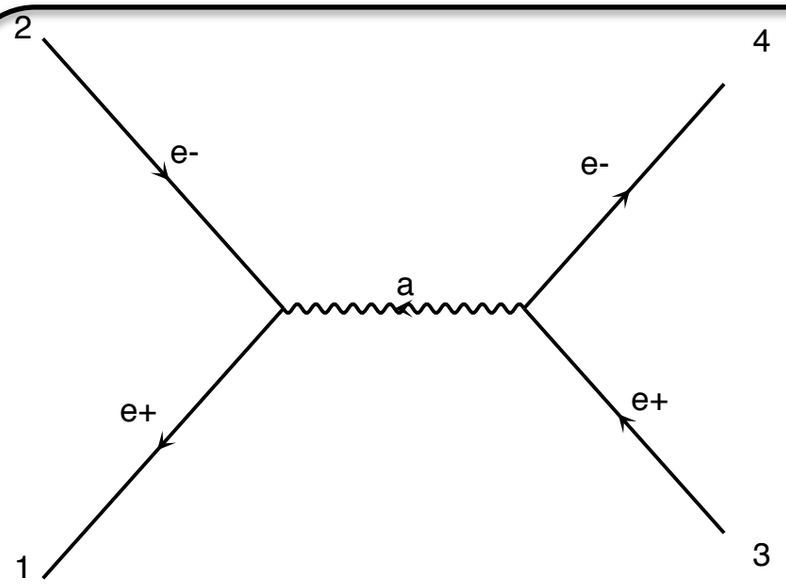
Single diagram computation

Idea

- Evaluate $\mathcal{M}$ for fixed helicity of external particles

➡ Multiply $\mathcal{M}$ with $\mathcal{M}^*$ -> $|\mathcal{M}|^2$

➡ Sum over helicity configurations and average the results



$$\mathcal{M} = (\bar{u}\, e\, \gamma^\mu v) \frac{g_{\mu\nu}}{q^2} (\bar{v}\, e\, \gamma^\nu u)$$

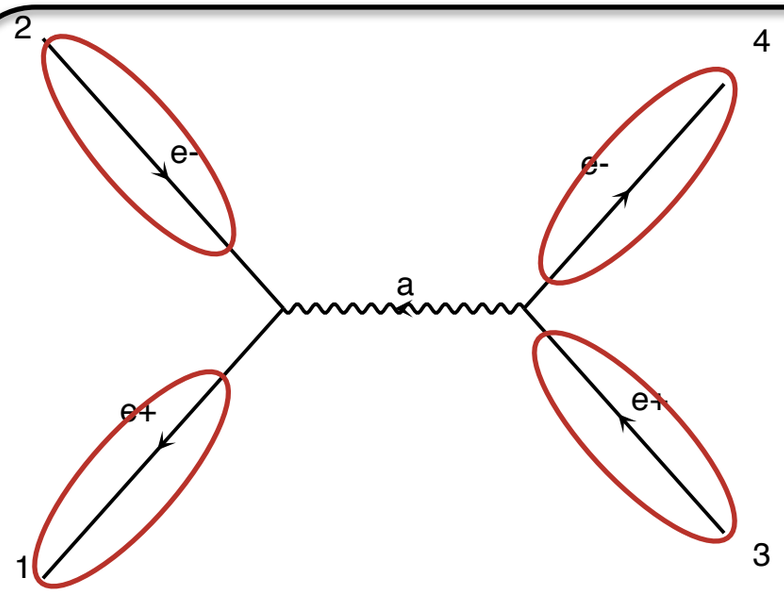
Single diagram computation

Idea

- Evaluate $\mathcal{M}$ for fixed helicity of external particles

➡ Multiply $\mathcal{M}$ with $\mathcal{M}^*$ -> $|\mathcal{M}|^2$

➡ Sum over helicity configurations and average the results



$$\mathcal{M} = (\bar{u}\, e\, \gamma^\mu v)\, \frac{g_{\mu\nu}}{q^2}\, (\bar{v}\, e\, \gamma^\nu u)$$

Numbers for given helicity and momenta

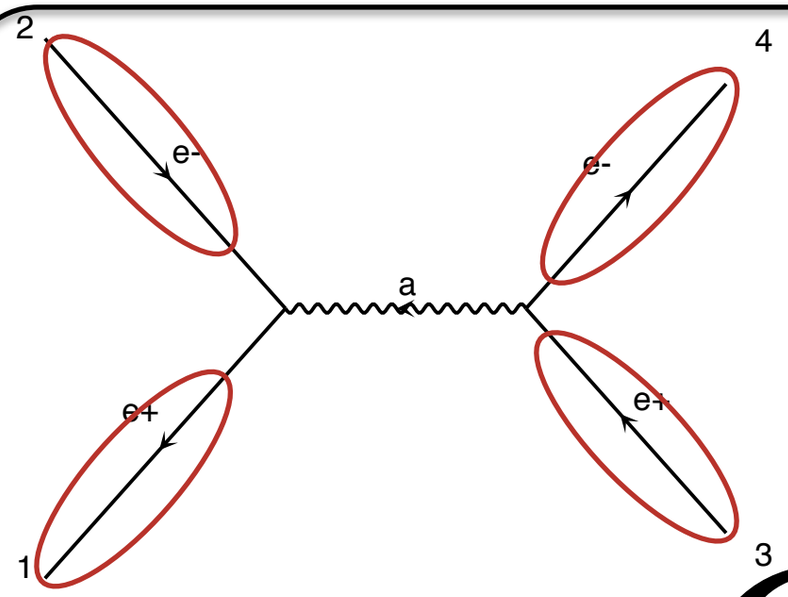
Single diagram computation

Idea

- Evaluate $\mathcal{M}$ for fixed helicity of external particles

➡ Multiply $\mathcal{M}$ with $\mathcal{M}^*$ -> $|\mathcal{M}|^2$

➡ Sum over helicity configurations and average the results



$$\mathcal{M} = (\bar{u}\, e\, \gamma^\mu v)\, \frac{g_{\mu\nu}}{q^2}\, (\bar{v}\, e\, \gamma^\nu u)$$

Numbers for given helicity and momenta

Lines present in the code.

$$\left.\begin{aligned} \bar{v}_1 &= fct(\vec{p}_1, m_1) \\ u_2 &= fct(\vec{p}_2, m_2) \\ v_3 &= fct(\vec{p}_3, m_3) \\ \bar{u}_4 &= fct(\vec{p}_4, m_4) \end{aligned}\right\}$$

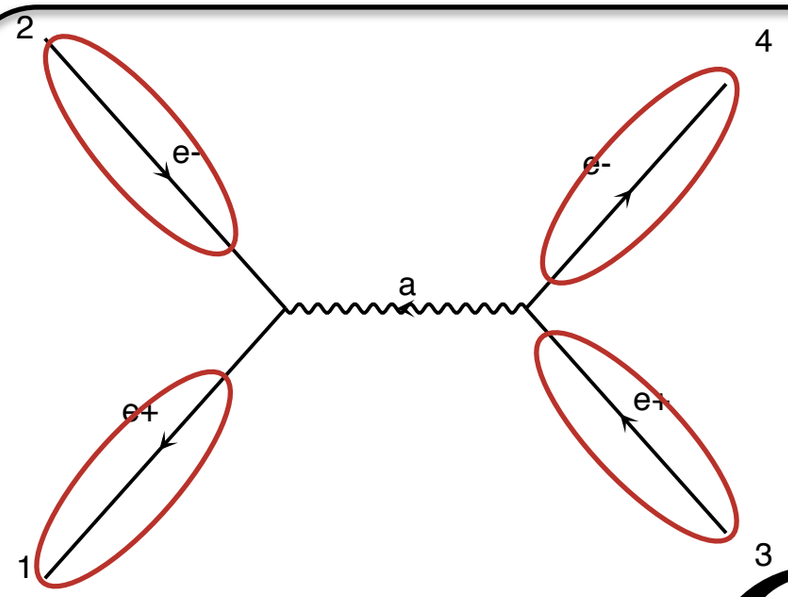
Single diagram computation

Idea

- Evaluate $\mathcal{M}$ for fixed helicity of external particles

➡ Multiply $\mathcal{M}$ with $\mathcal{M}^*$ -> $|\mathcal{M}|^2$

➡ Sum over helicity configurations and average the results



Lines present in the code.

$$\bar{v}_1 = fct(\vec{p}_1, m_1)$$
$$u_2 = fct(\vec{p}_2, m_2)$$
$$v_3 = fct(\vec{p}_3, m_3)$$
$$\bar{u}_4 = fct(\vec{p}_4, m_4)$$

$$\mathcal{M} = (\bar{u}\, e\, \gamma^\mu v)\, \frac{g_{\mu\nu}}{q^2}\, (\bar{v}\, e\, \gamma^\nu u)$$

Numbers for given helicity and momenta

$$u(p) = \begin{pmatrix} \omega_{-\lambda}(p)\chi_\lambda(\vec{p}) \\ \omega_\lambda(p)\chi_\lambda(\vec{p}) \end{pmatrix}$$

$$\omega_\pm(p) \equiv \sqrt{E \pm |\vec{p}|}.$$

$$\chi_+(\vec{p}) = \frac{1}{\sqrt{2|\vec{p}|(|\vec{p}| + p_z)}} \begin{pmatrix} |\vec{p}| + p_z \\ p_x + ip_y \end{pmatrix},$$

$$\chi_-(\vec{p}) = \frac{1}{\sqrt{2|\vec{p}|(|\vec{p}| + p_z)}} \begin{pmatrix} -p_x + ip_y \\ |\vec{p}| + p_z \end{pmatrix}.$$

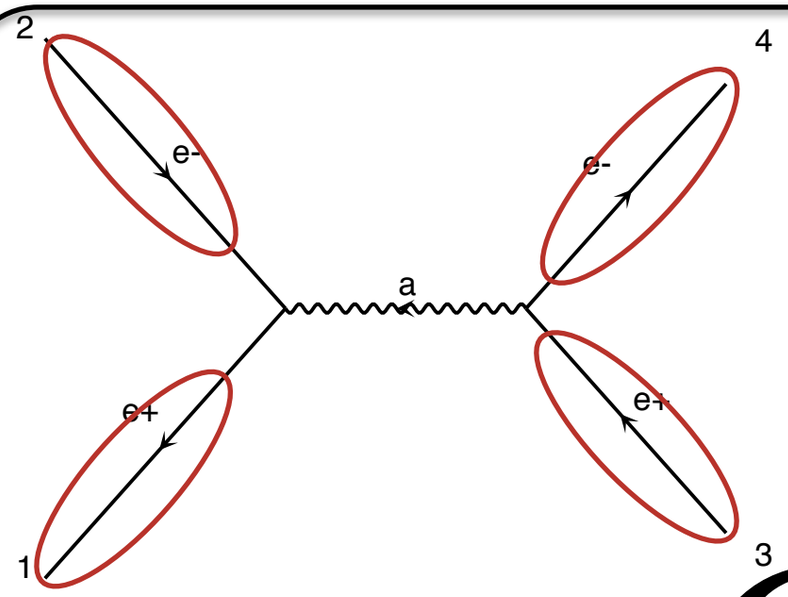
Single diagram computation

Idea

- Evaluate $\mathcal{M}$ for fixed helicity of external particles

➡ Multiply $\mathcal{M}$ with $\mathcal{M}^*$ -> $|\mathcal{M}|^2$

➡ Sum over helicity configurations and average the results



$$\mathcal{M} = (\bar{u}\, e\, \gamma^\mu v)\, \frac{g_{\mu\nu}}{q^2}\, (\bar{v}\, e\, \gamma^\nu u)$$

Numbers for given helicity and momenta

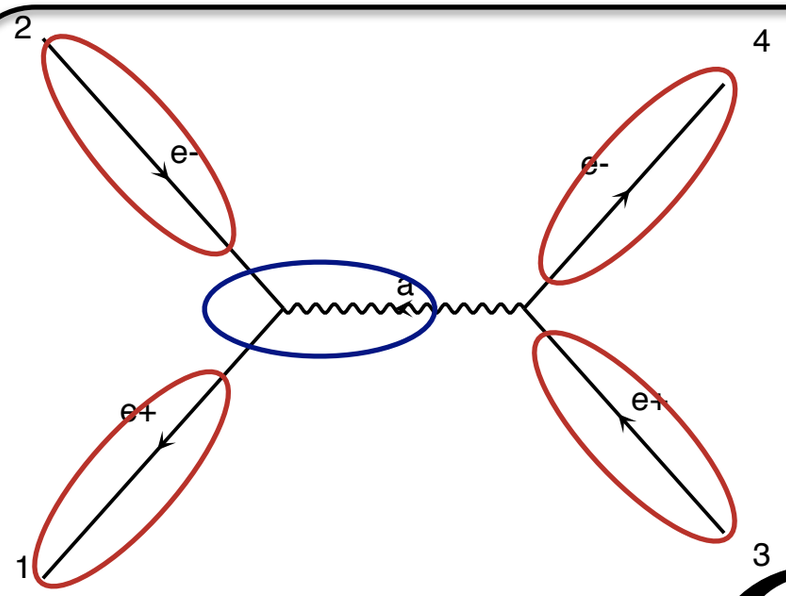
Lines present in the code.

$$\left.\begin{aligned} \bar{v}_1 &= fct(\vec{p}_1, m_1) \\ u_2 &= fct(\vec{p}_2, m_2) \\ v_3 &= fct(\vec{p}_3, m_3) \\ \bar{u}_4 &= fct(\vec{p}_4, m_4) \end{aligned}\right\}$$

Single diagram computation

Idea

- Evaluate $\mathcal{M}$ for fixed helicity of external particles
  - ➡ Multiply $\mathcal{M}$ with $\mathcal{M}^*$ -> $|\mathcal{M}|^2$
  - ➡ Sum over helicity configurations and average the results



$$\mathcal{M} = \left(\bar{u}\, e\, \gamma^\mu v\right) \frac{g_{\mu\nu}}{q^2} \left(\bar{v}\, e\, \gamma^\nu u\right)$$

Numbers for given helicity and momenta

Calculate propagator wavefunctions

Lines present in the code.

$$\left.\begin{aligned} \bar{v}_1 &= fct(\vec{p}_1, m_1) \\ u_2 &= fct(\vec{p}_2, m_2) \\ v_3 &= fct(\vec{p}_3, m_3) \\ \bar{u}_4 &= fct(\vec{p}_4, m_4) \end{aligned}\right\}$$

$$W_a = fct(\bar{v}_1, u_2, m_a, \Gamma_a)$$

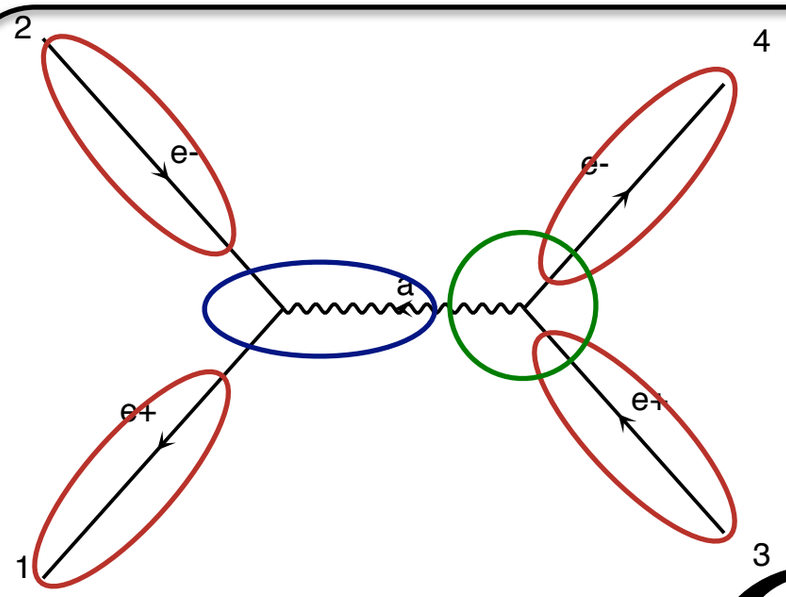
Single diagram computation

Idea

- Evaluate $\mathcal{M}$ for fixed helicity of external particles

➡ Multiply $\mathcal{M}$ with $\mathcal{M}^*$ -> $|\mathcal{M}|^2$

➡ Sum over helicity configurations and average the results



$$\mathcal{M} = \left(\bar{u}\, e\, \gamma^\mu v\right) \frac{g_{\mu\nu}}{q^2} \left(\bar{v}\, e\, \gamma^\nu u\right)$$

Numbers for given helicity and momenta

Calculate propagator wavefunctions

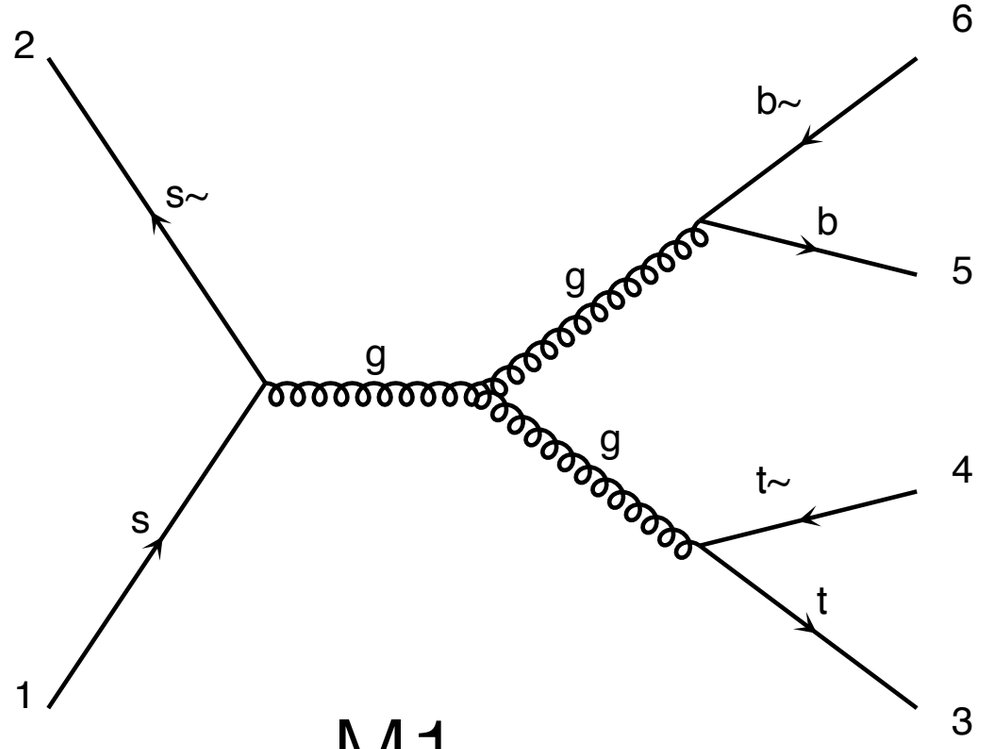
Finally evaluate amplitude (c-number)

Lines present in the code.

$$\left.\begin{aligned}
\bar{v}_1 &= fct(\vec{p}_1, m_1) \\
u_2 &= fct(\vec{p}_2, m_2) \\
v_3 &= fct(\vec{p}_3, m_3) \\
\bar{u}_4 &= fct(\vec{p}_4, m_4) \\
W_a &= fct(\bar{v}_1, u_2, m_a, \Gamma_a) \\
\mathcal{M} &= fct(v_3, \bar{u}_4, W_a)
\end{aligned}\right\}$$

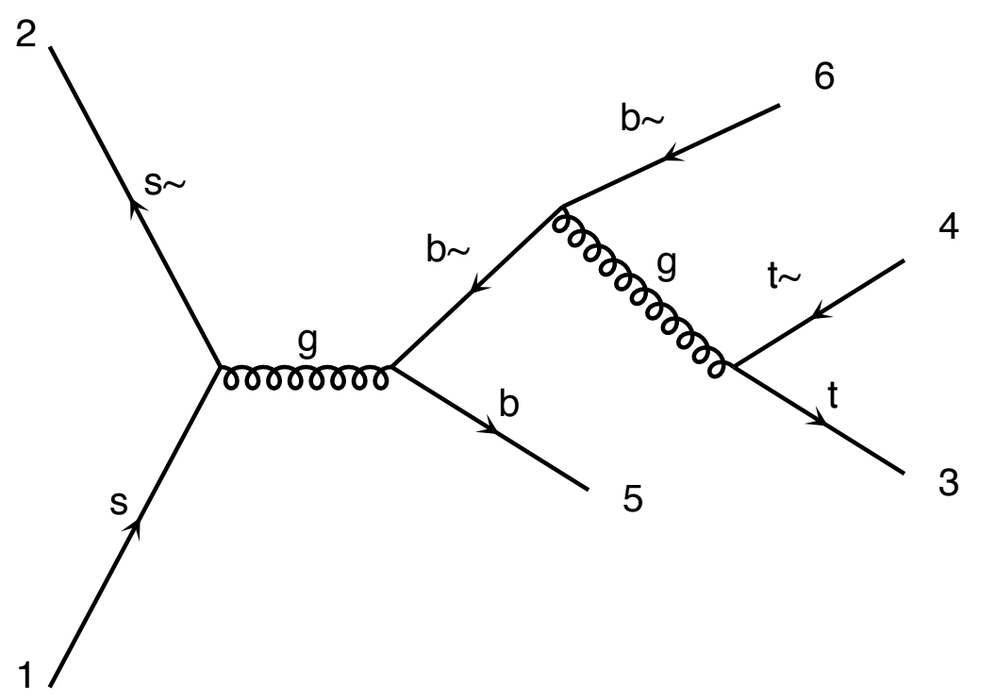
Multiple diagram computation

Known



M1

Number of routines: 0



M2

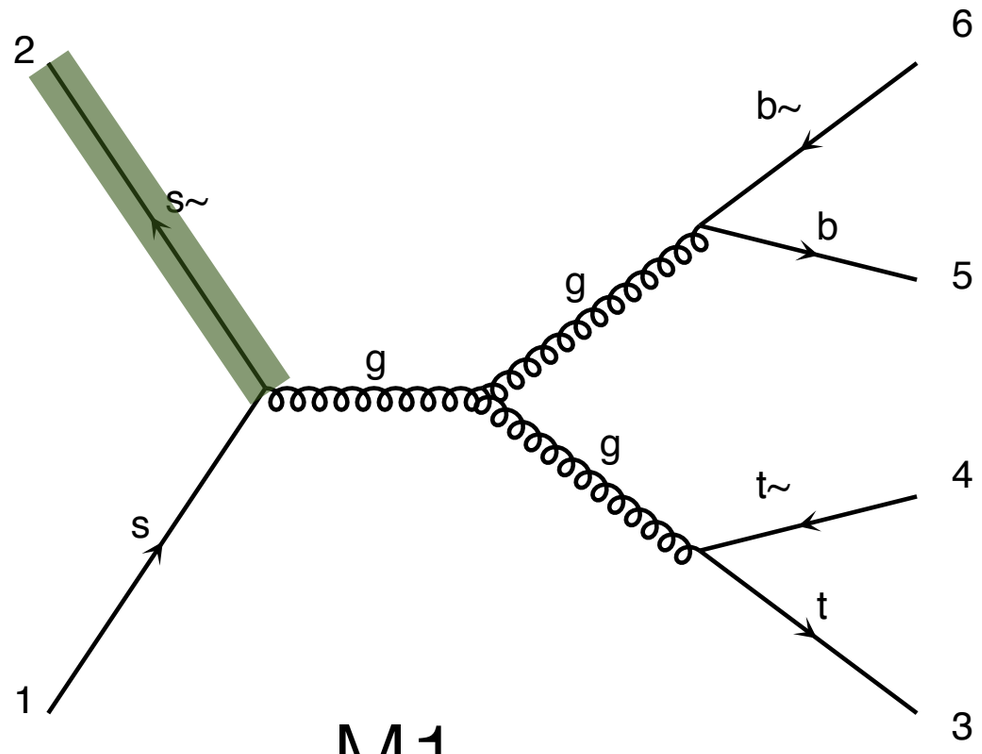
Number of routines: 0

Number of routines for both: 0

$$|M|^2 = |M_1 + M_2|^2$$

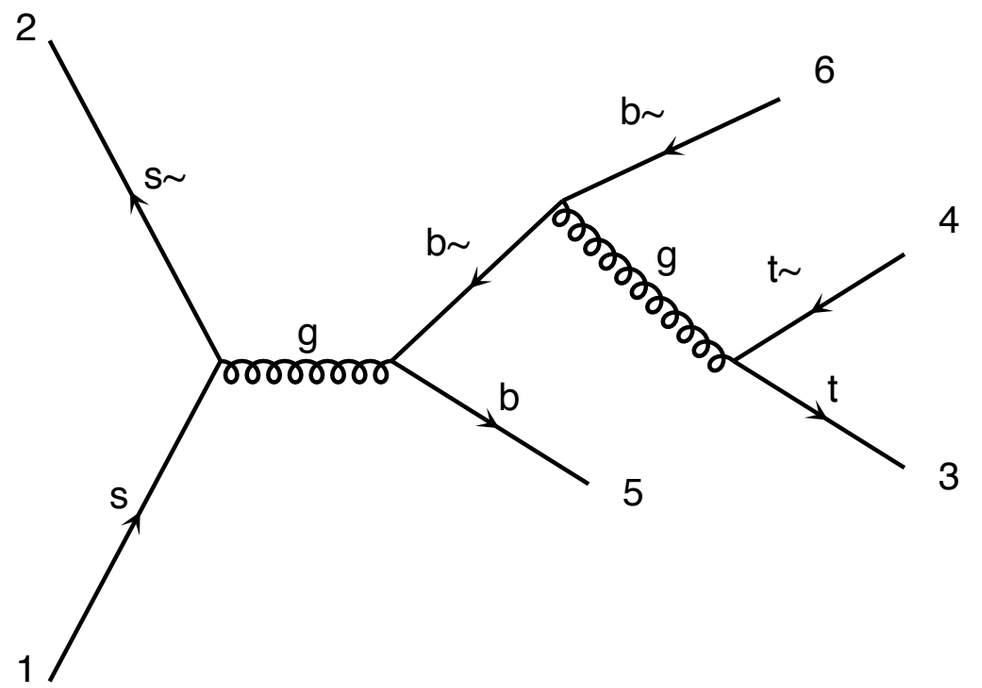
Multiple diagram computation

Known



M1

Number of routines: 1



M2

Number of routines: 0

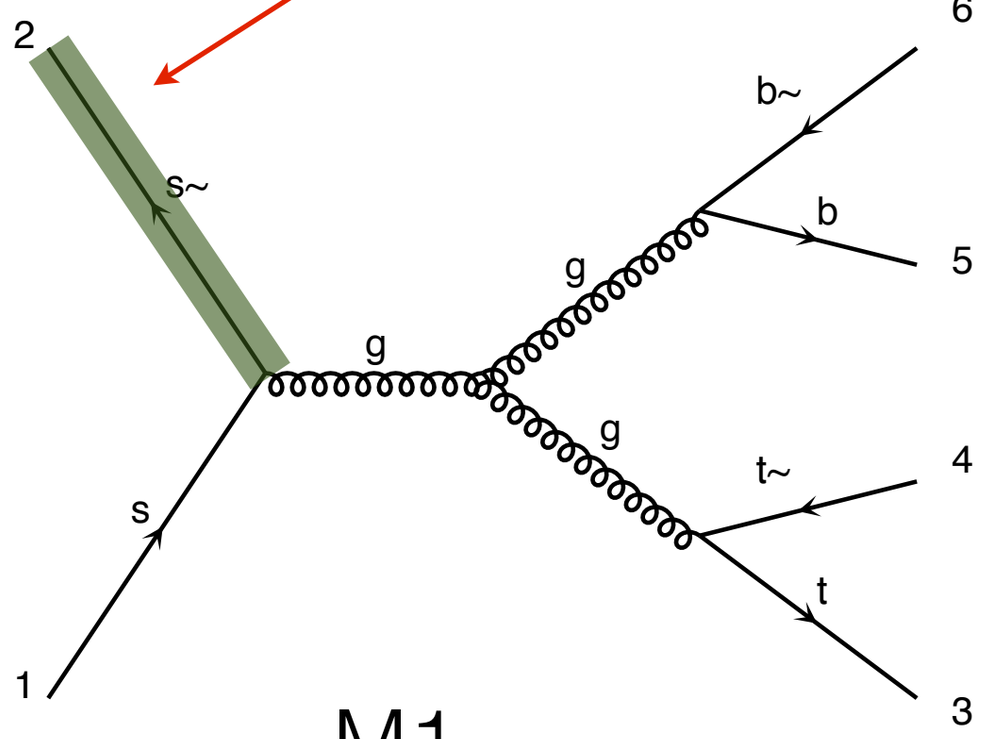
Number of routines for both: 1

$$|M|^2 = |M_1 + M_2|^2$$

Multiple diagram computation

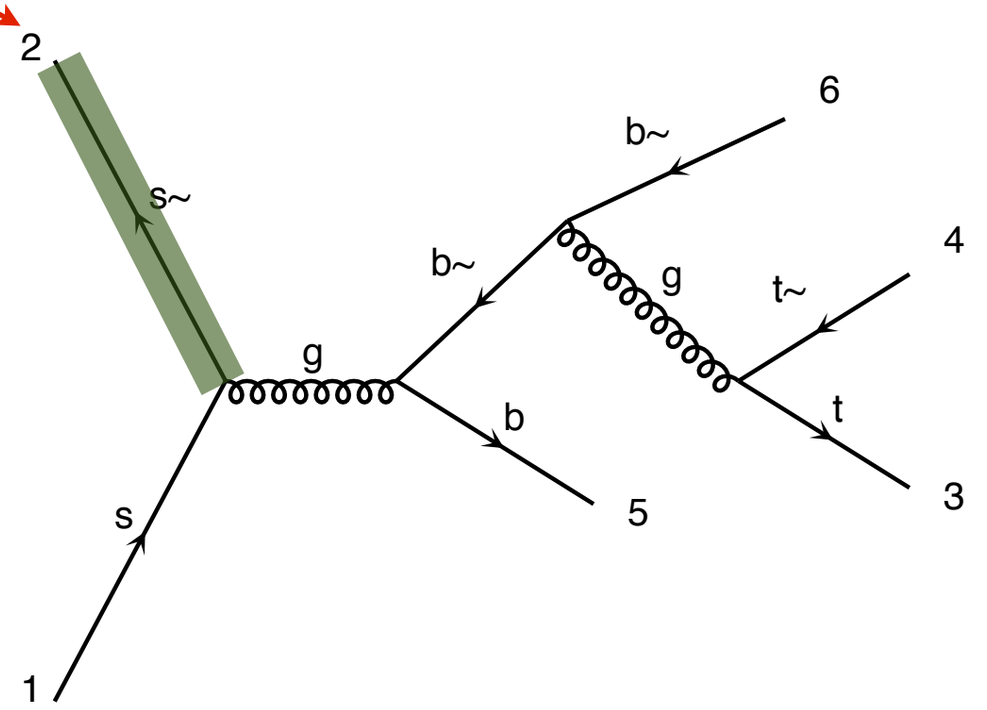
Identical

Known



M1

Number of routines: 1



M2

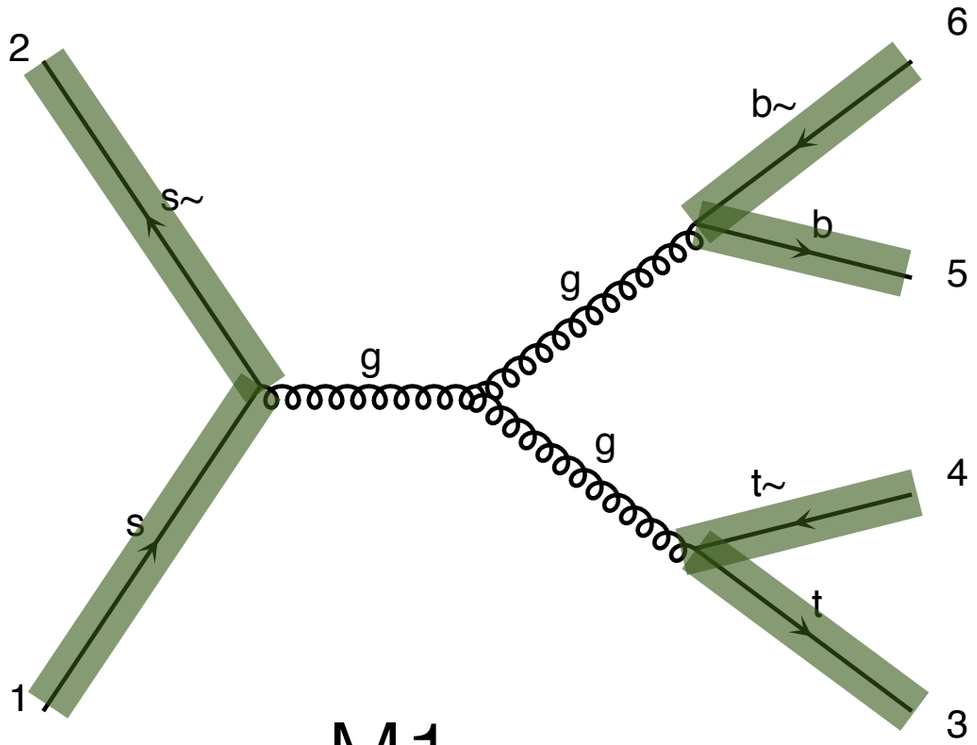
Number of routines: 1

Number of routines for both: 1

$$|M|^2 = |M_1 + M_2|^2$$

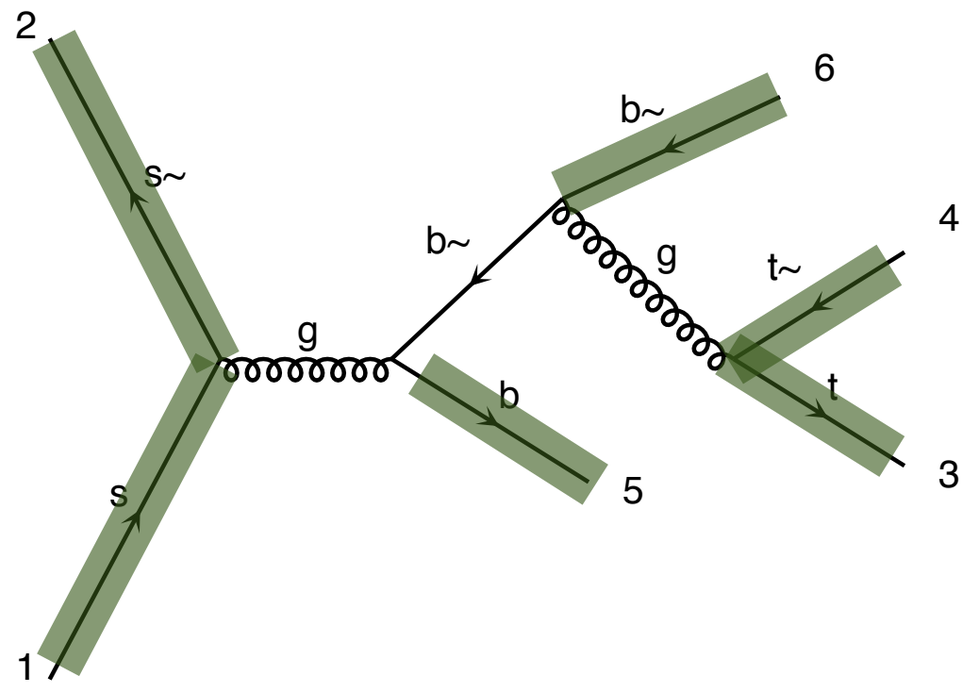
Multiple diagram computation

Known



M1

Number of routines: 6



M2

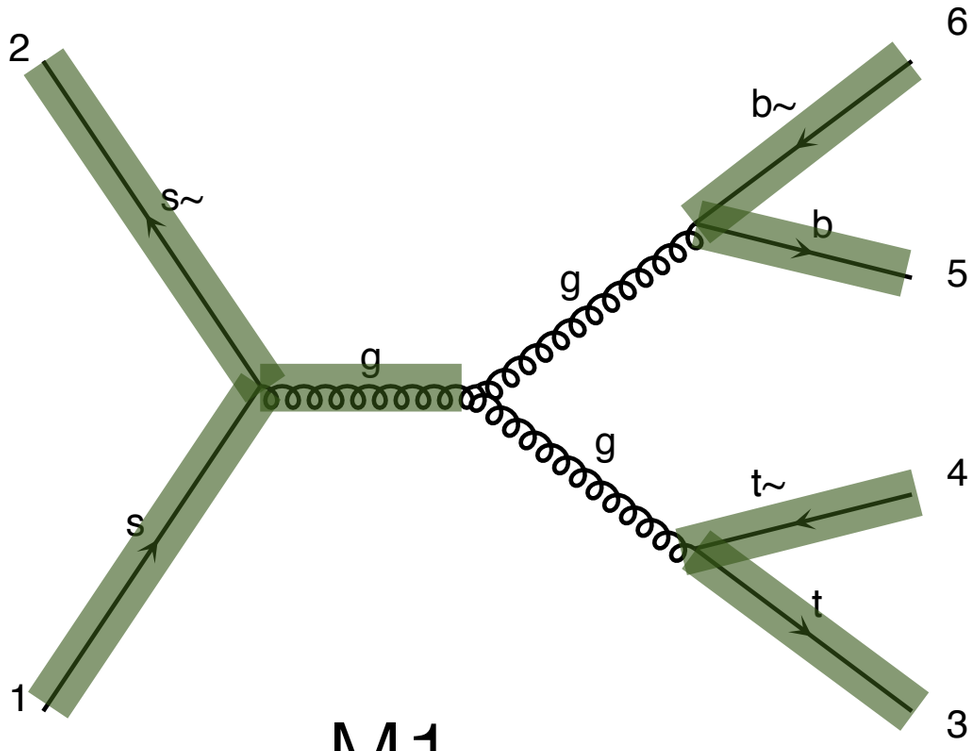
Number of routines: 6

Number of routines for both: 6

$$|M|^2 = |M_1 + M_2|^2$$

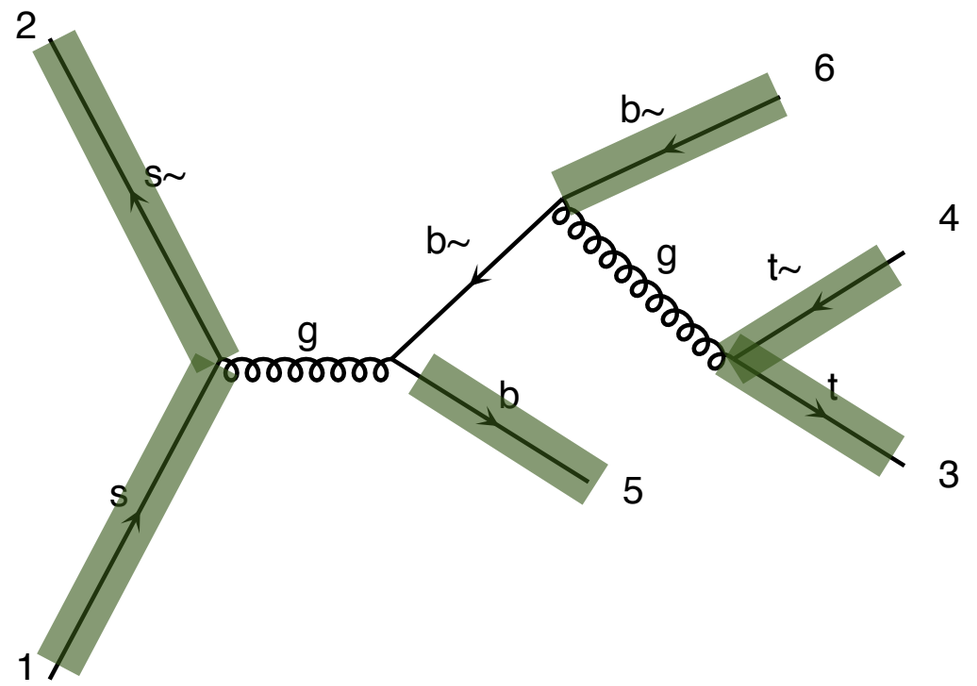
Multiple diagram computation

Known



M1

Number of routines: 7



M2

Number of routines: 6

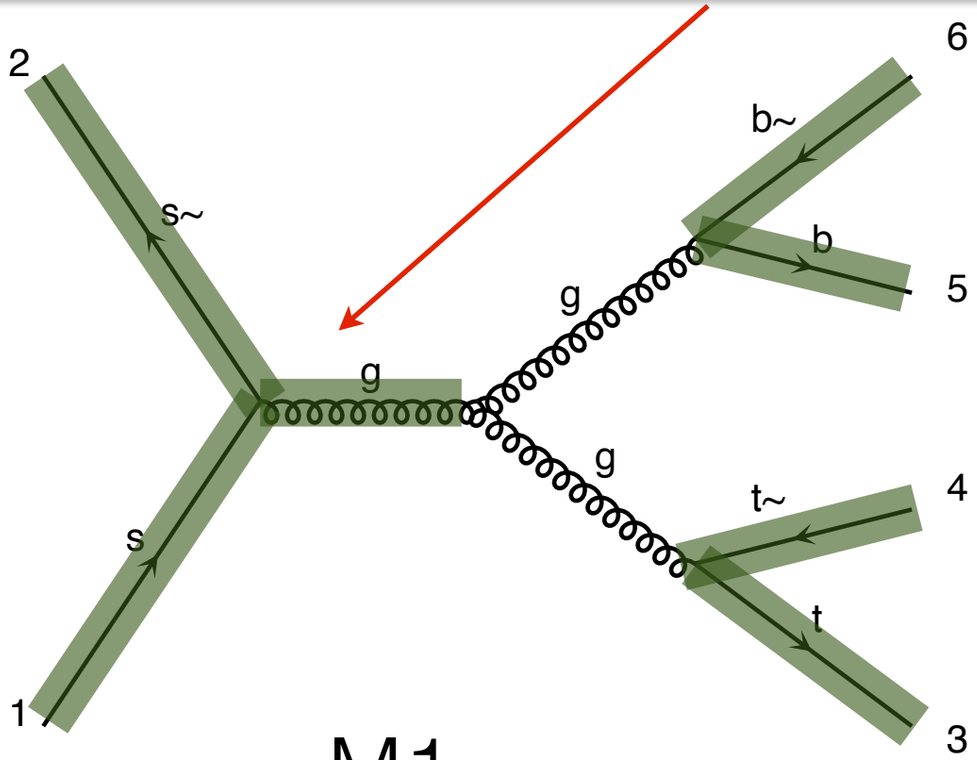
Number of routines for both: 7

$$|M|^2 = |M_1 + M_2|^2$$

Multiple diagram computation

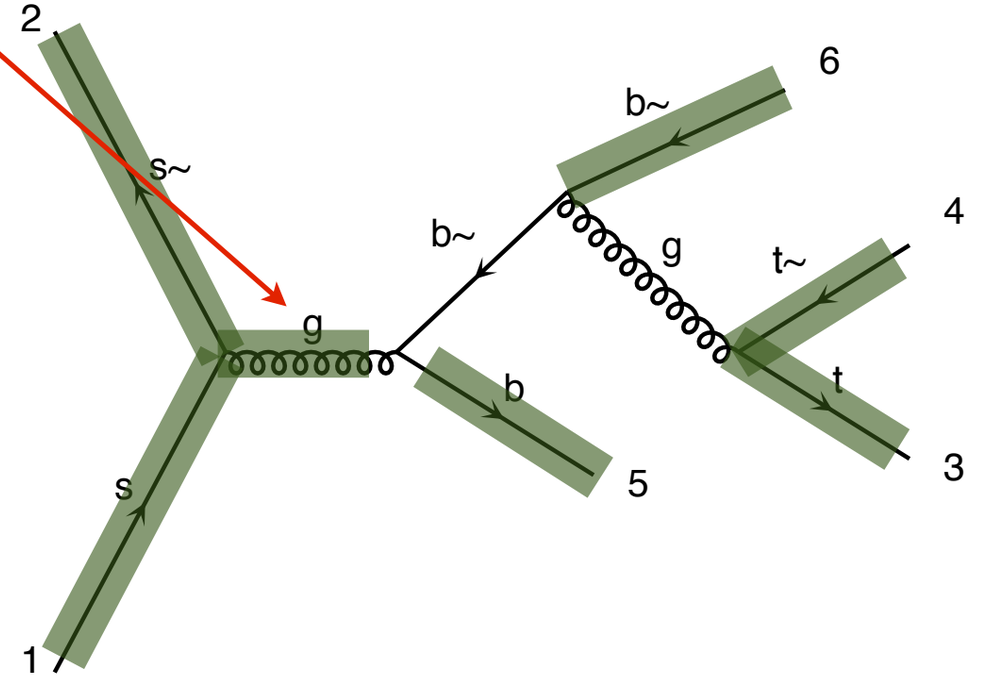
Known

Identical



M1

Number of routines: 7



M2

Number of routines: 7

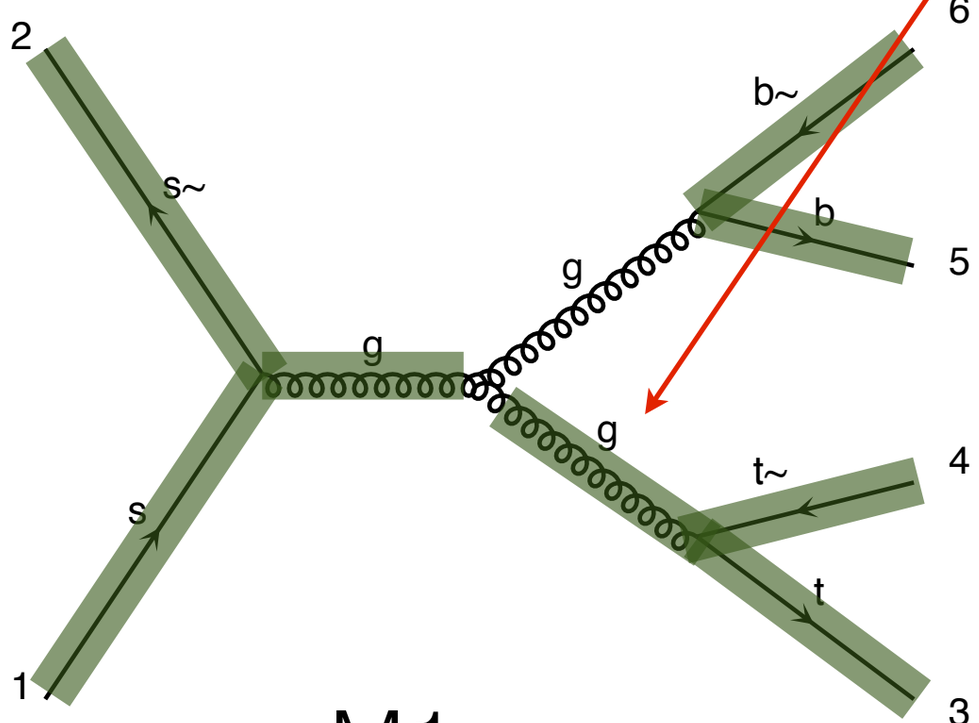
Number of routines for both: 7

$$|M|^2 = |M_1 + M_2|^2$$

Multiple diagram computation

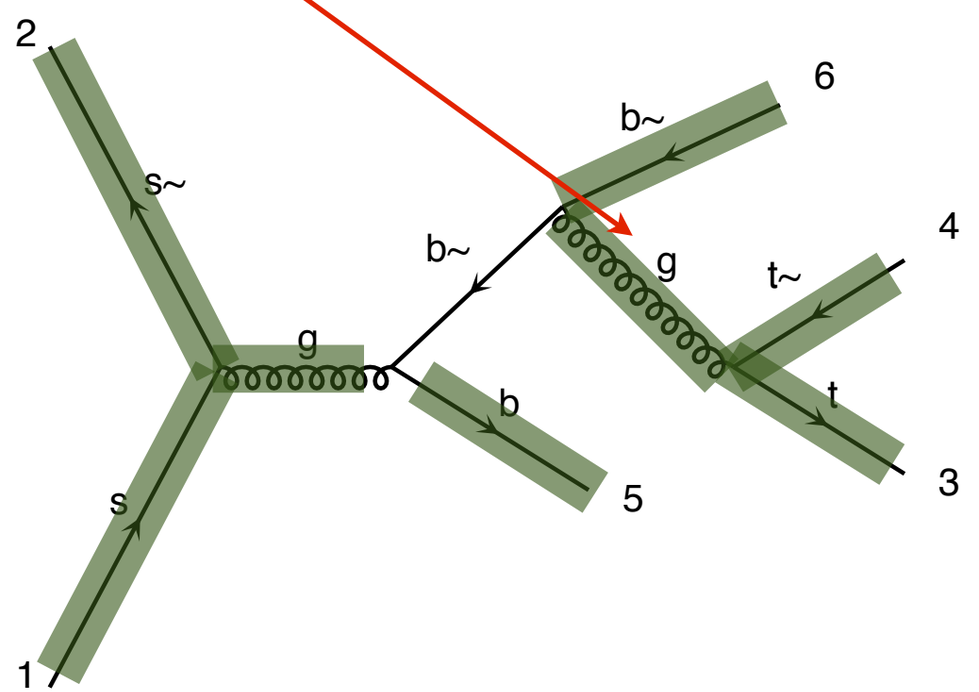
Known

Identical



M1

Number of routines: 8



M2

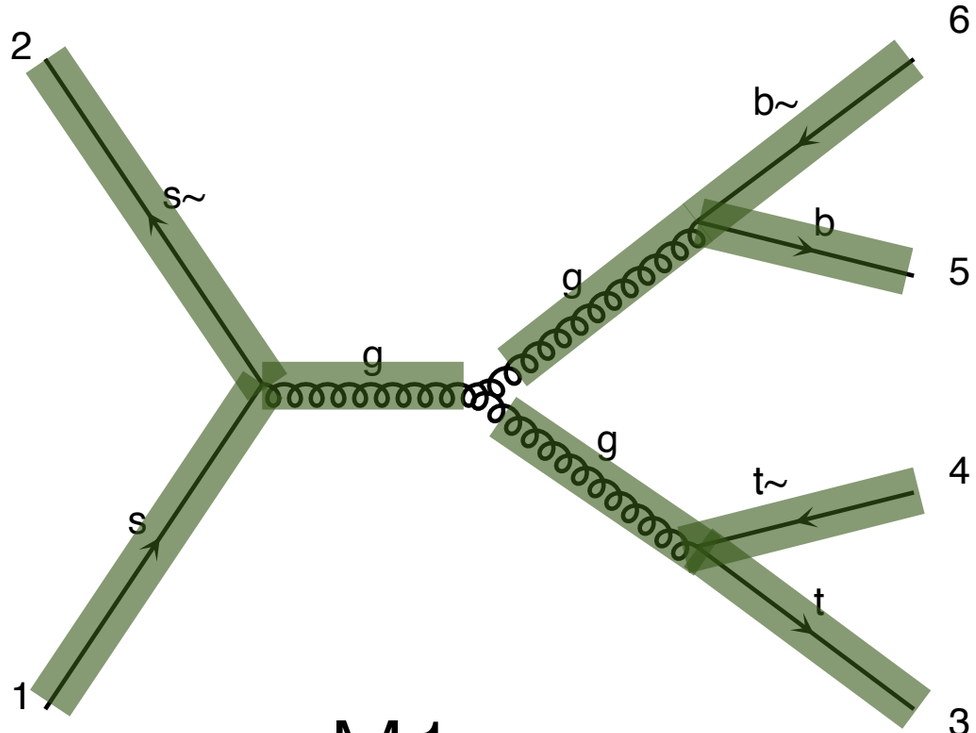
Number of routines: 8

Number of routines for both: 8

$$|M|^2 = |M_1 + M_2|^2$$

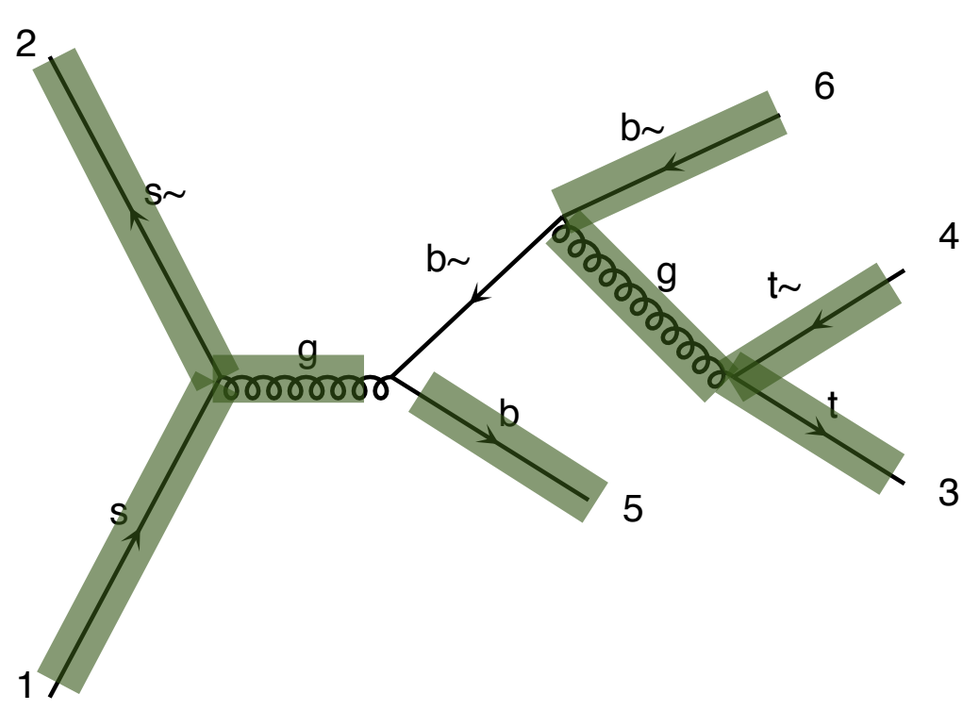
Multiple diagram computation

Known



M1

Number of routines: 9



M2

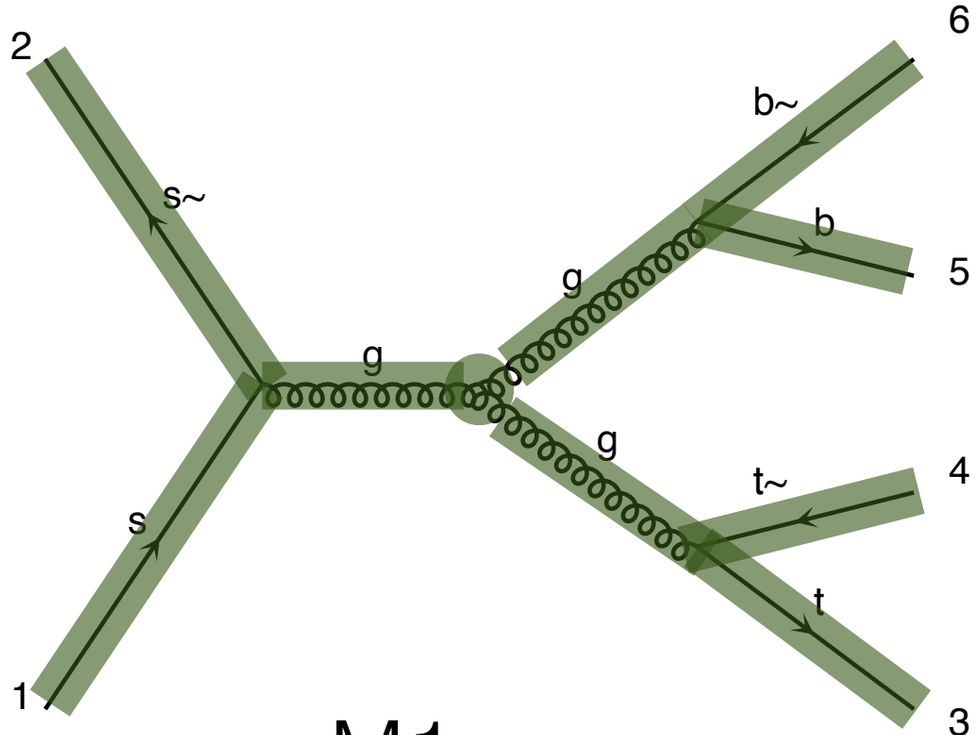
Number of routines: 8

Number of routines for both: 9

$$|M|^2 = |M_1 + M_2|^2$$

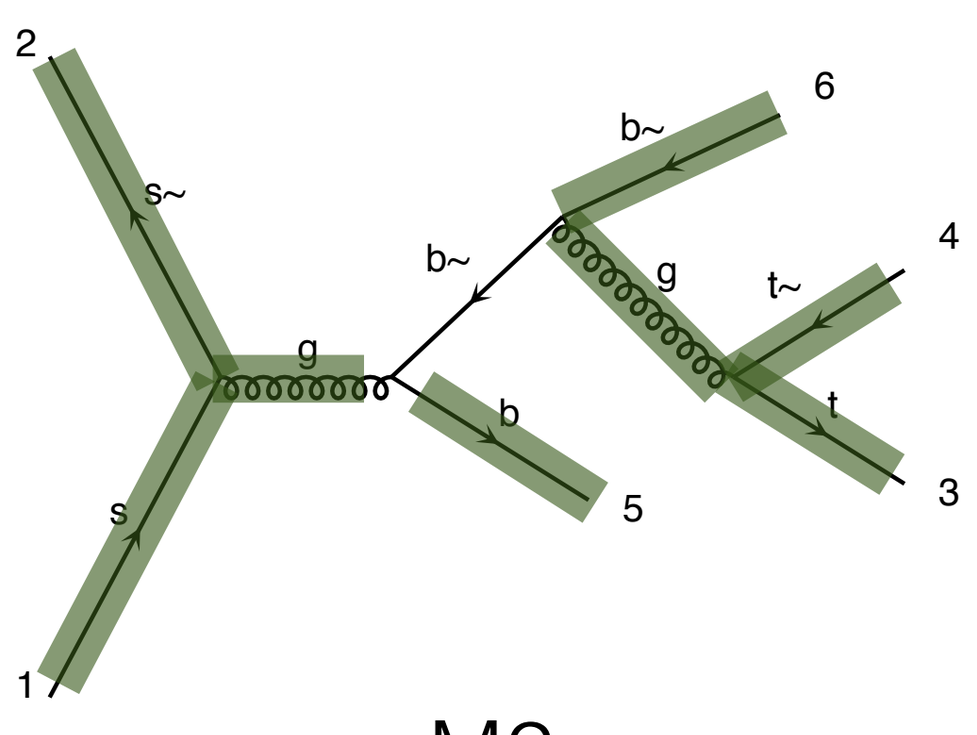
Multiple diagram computation

Known



M1

Number of routines: 10



M2

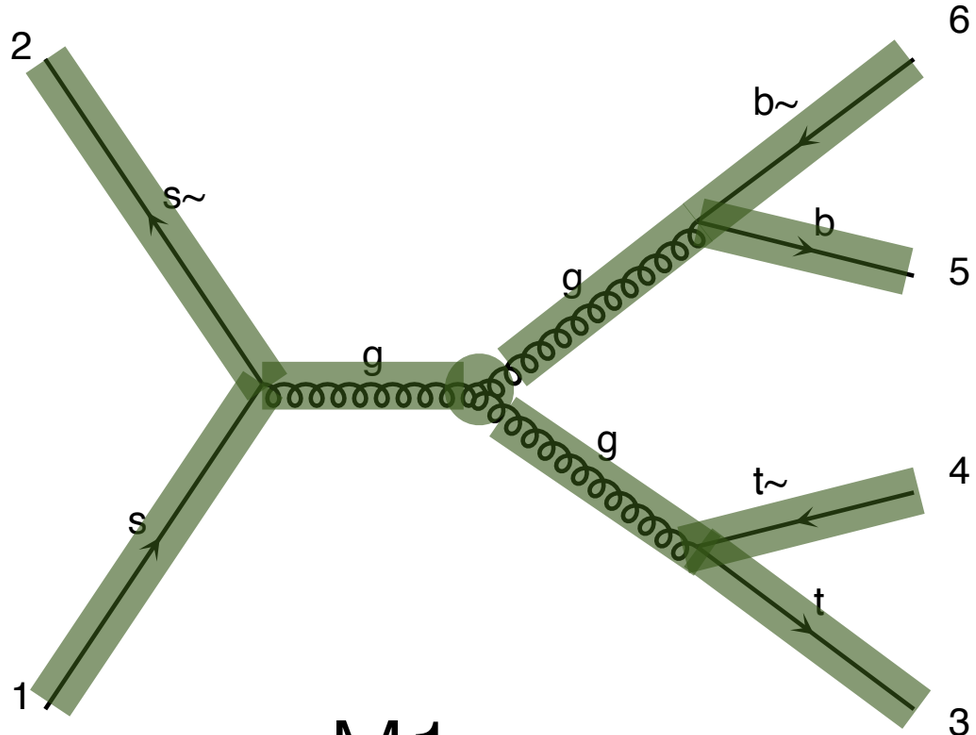
Number of routines: 8

Number of routines for both: 10

$$|M|^2 = |M_1 + M_2|^2$$

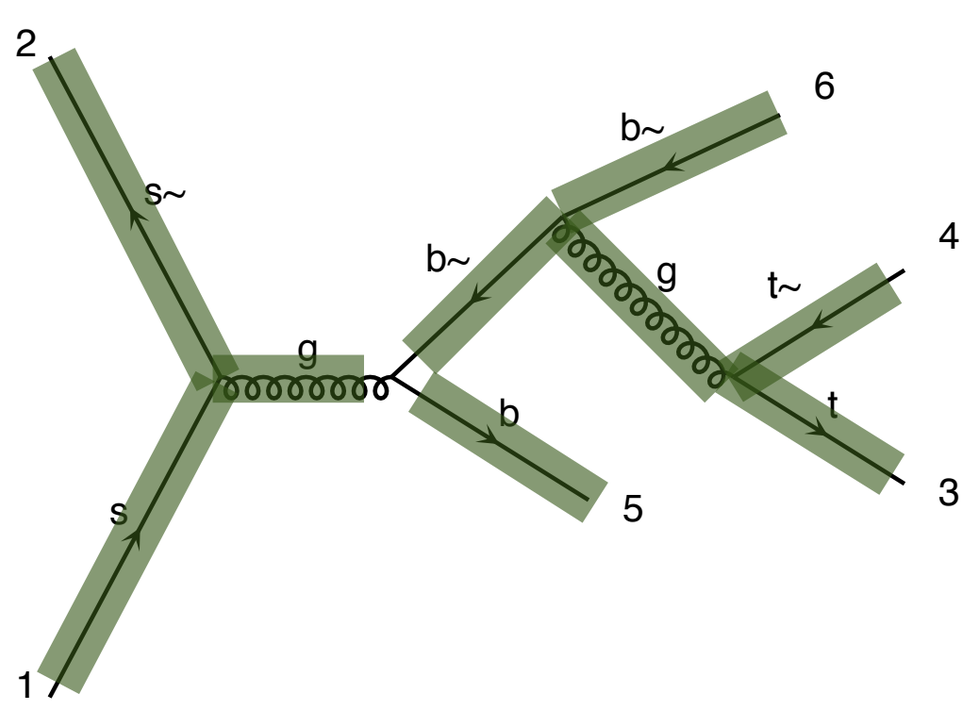
Multiple diagram computation

Known



M1

Number of routines: 10



M2

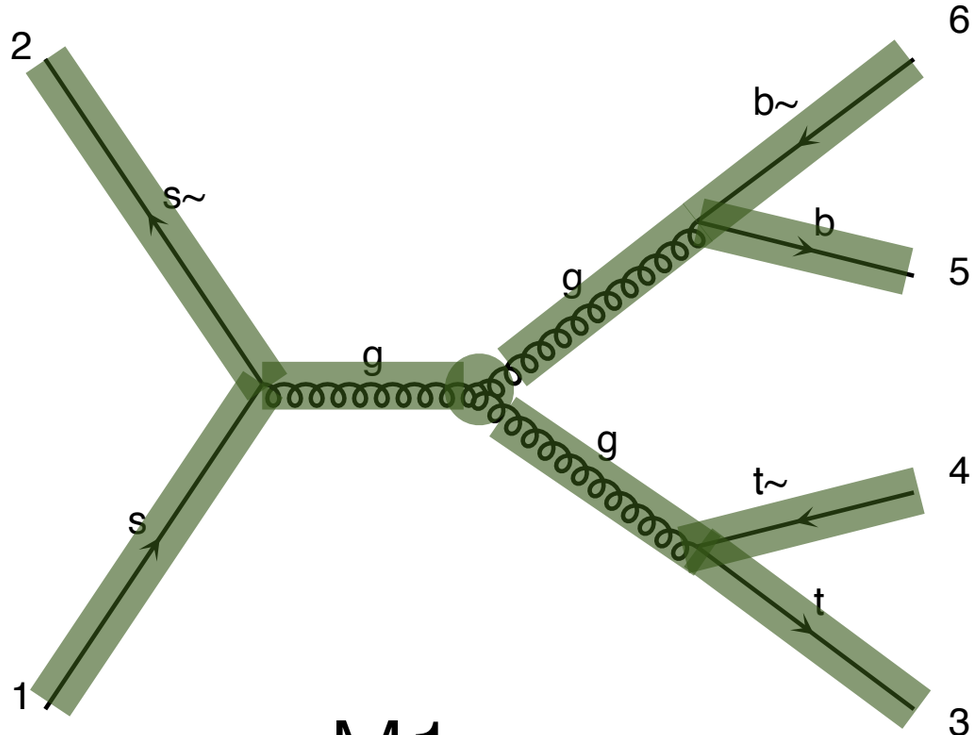
Number of routines: 9

Number of routines for both: 11

$$|M|^2 = |M_1 + M_2|^2$$

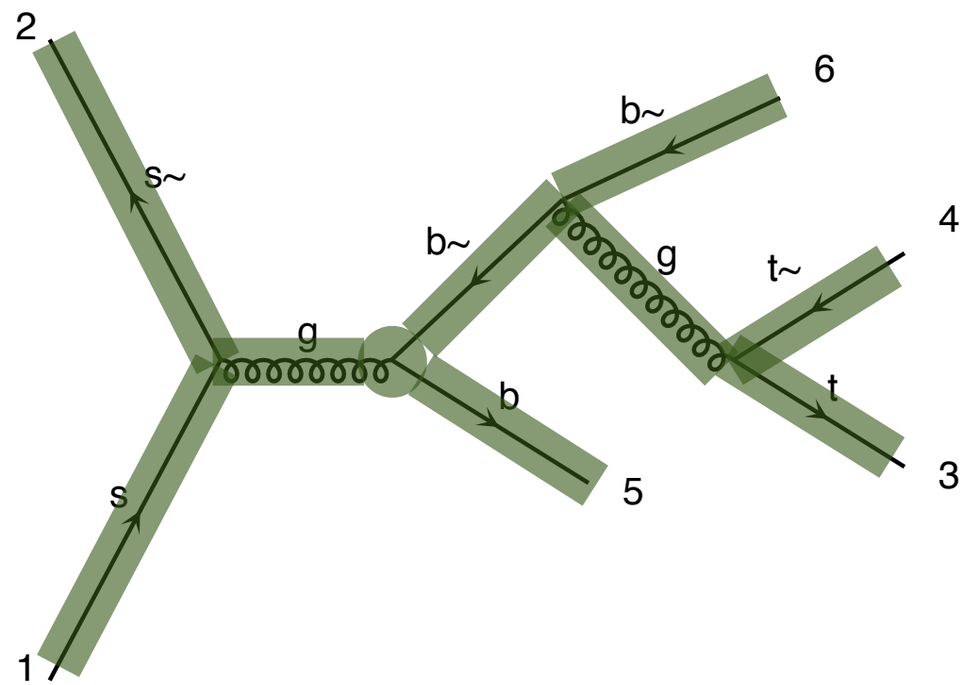
Multiple diagram computation

Known



M1

Number of routines: 10



M2

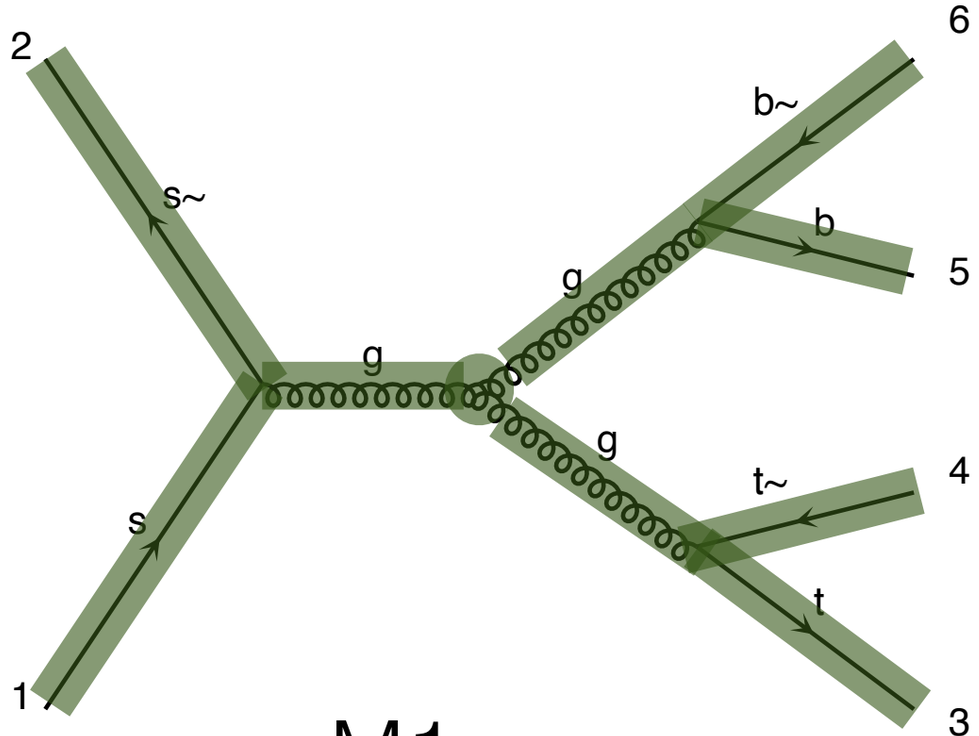
Number of routines: 10

Number of routines for both: 12

$$|M|^2 = |M_1 + M_2|^2$$

Multiple diagram computation

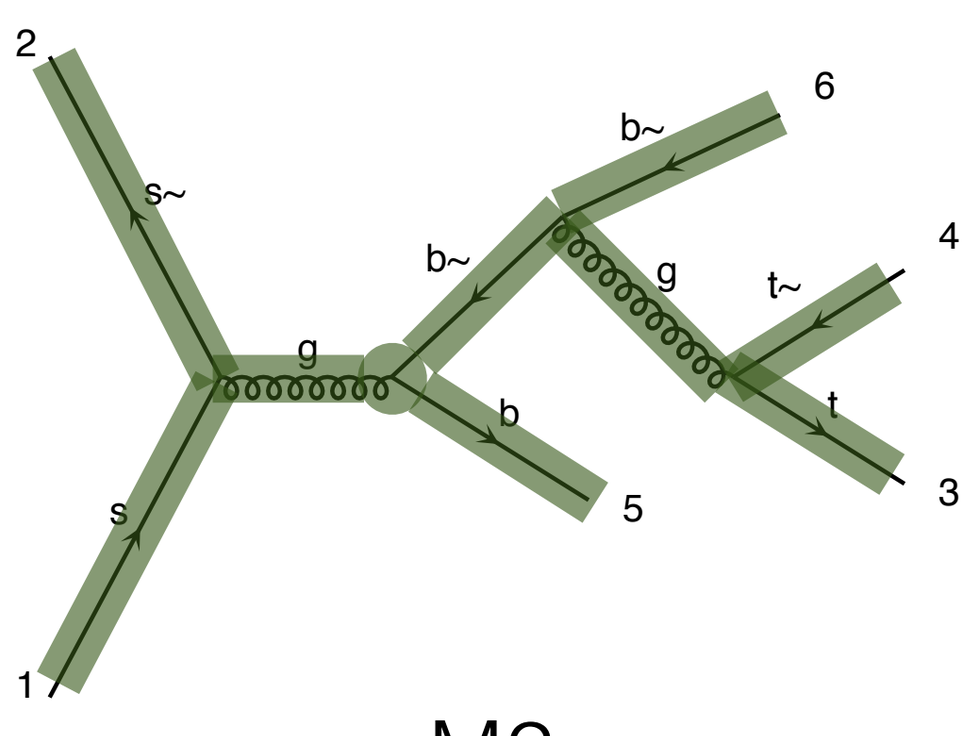
Known



M1

Number of routines: 10

2(N+1)



M2

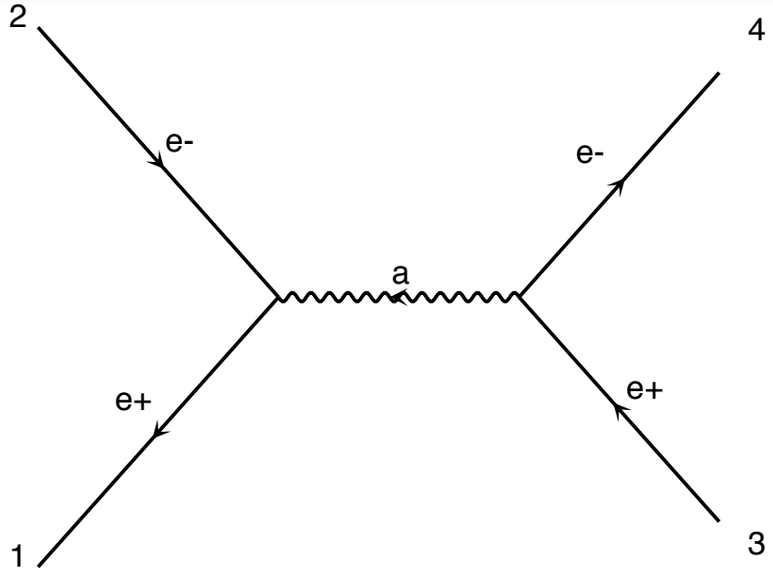
Number of routines: 10

2(N+1)

Number of routines for both: 12

$$|M|^2 = |M_1 + M_2|^2$$

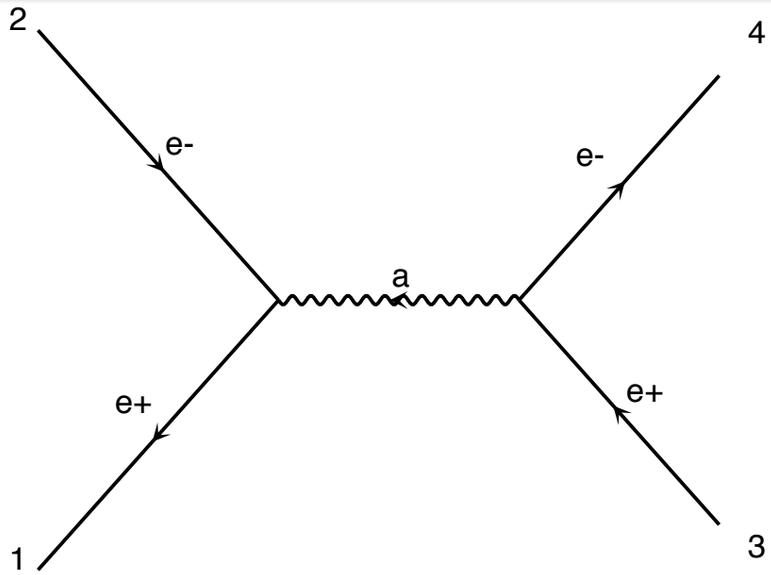
Building blocks : HELAS functions



$$\mathcal{M} = \bar{u}\gamma^{\mu}v P_{\mu\nu} \bar{u}\gamma^{\nu}v$$

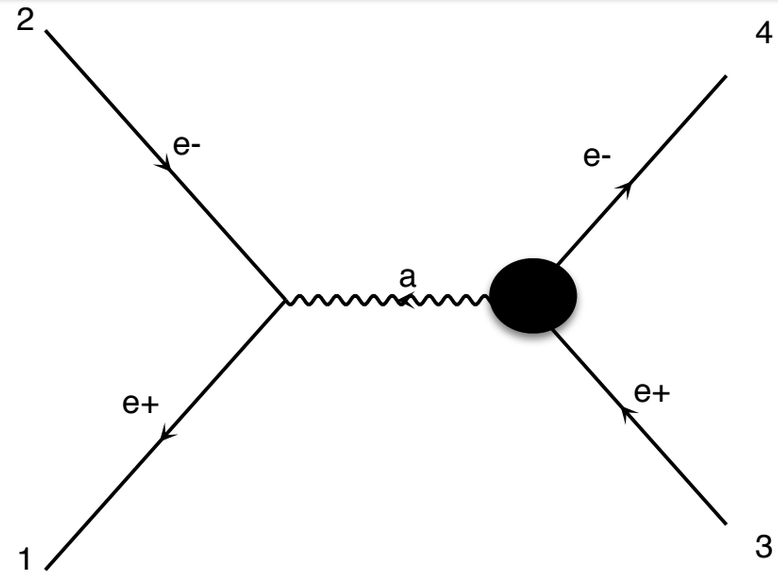
$$\bar{u}_1 = fct(\vec{p}_1, m)$$
$$v_2 = fct(\vec{p}_2, m)$$
$$\bar{u}_3 = fct(\vec{p}_3, m)$$
$$v_4 = fct(\vec{p}_4, m)$$
$$W_a = fct(\bar{u}_1, v_2, M_a, \Gamma_a)$$
$$\mathcal{M} = fct(\bar{u}_3, v_4, W_a)$$

Building blocks : HELAS functions



$$\mathcal{M} = \bar{u}\gamma^{\mu}vP_{\mu\nu}\bar{u}\gamma^{\nu}v$$

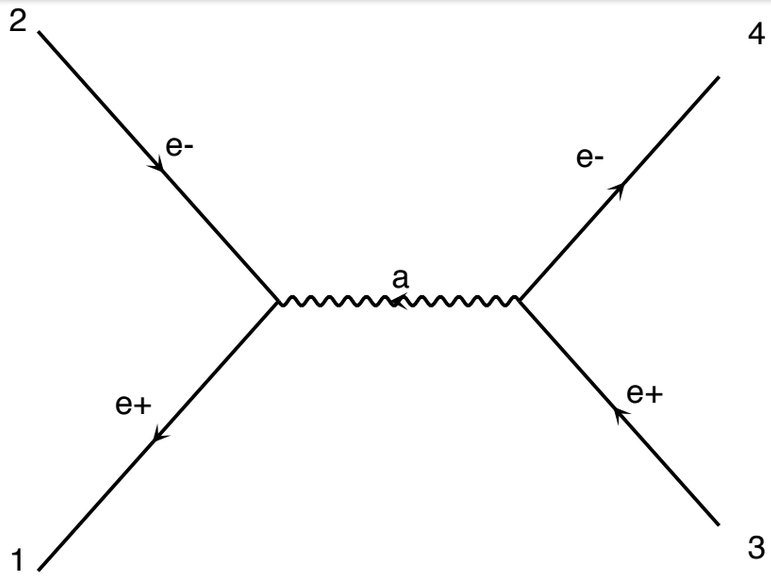
$$\bar{u}_1 = fct(\vec{p}_1, m)$$
$$v_2 = fct(\vec{p}_2, m)$$
$$\bar{u}_3 = fct(\vec{p}_3, m)$$
$$v_4 = fct(\vec{p}_4, m)$$
$$W_a = fct(\bar{u}_1, v_2, M_a, \Gamma_a)$$
$$\mathcal{M} = fct(\bar{u}_3, v_4, W_a)$$



$$\mathcal{M} = \bar{u}\gamma^{\mu}vP_{\mu\nu}\bar{u}\Gamma^{\mu}v$$

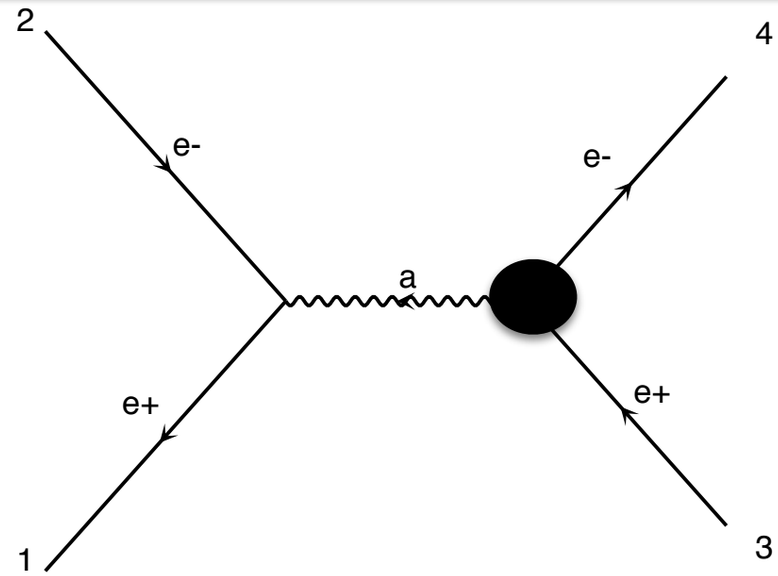
$$\bar{u}_1 = fct(\vec{p}_1, m)$$
$$v_2 = fct(\vec{p}_2, m)$$
$$\bar{u}_3 = fct(\vec{p}_3, m)$$
$$v_4 = fct(\vec{p}_4, m)$$
$$W_a = fct(\bar{u}_1, v_2, M_a, \Gamma_a)$$
$$\mathcal{M} = fct(\bar{u}_3, v_4, W_a)$$

Building blocks : HELAS functions



$$\mathcal{M} = \bar{u}\gamma^{\mu}vP_{\mu\nu}\bar{u}\gamma^{\nu}v$$

$$\bar{u}_1 = fct(\vec{p}_1, m)$$
$$v_2 = fct(\vec{p}_2, m)$$
$$\bar{u}_3 = fct(\vec{p}_3, m)$$
$$v_4 = fct(\vec{p}_4, m)$$
$$W_a = fct(\bar{u}_1, v_2, M_a, \Gamma_a)$$
$$\mathcal{M} = fct(\bar{u}_3, v_4, W_a)$$



$$\mathcal{M} = \bar{u}\gamma^{\mu}vP_{\mu\nu}\bar{u}\Gamma^{\mu}v$$

$$\bar{u}_1 = fct(\vec{p}_1, m)$$
$$v_2 = fct(\vec{p}_2, m)$$
$$\bar{u}_3 = fct(\vec{p}_3, m)$$
$$v_4 = fct(\vec{p}_4, m)$$
$$W_a = fct(\bar{u}_1, v_2, M_a, \Gamma_a)$$
$$\mathcal{M} = fct(\bar{u}_3, v_4, W_a)$$

From HELAS to ALOHA

- **Original HELicity Amplitude Subroutine library**
  [Murayama, Watanabe, Hagiwara]

From HELAS to ALOHA

- **Original HELicity Amplitude Subroutine library**
  [Murayama, Watanabe, Hagiwara]
- **One routine per Lorentz structure**

From HELAS to ALOHA

- **Original HELicity Amplitude Subroutine library**
  [Murayama, Watanabe, Hagiwara]

- **One routine per Lorentz structure**

  ➡**MSSM** [cho, al] hep-ph/0601063 (2006)

From HELAS to ALOHA

- **Original HELicity Amplitude Subroutine library**
  [Murayama, Watanabe, Hagiwara]

- **One routine per Lorentz structure**

  ➡**MSSM** [cho, al] hep-ph/0601063 (2006)

  ➡**HEFT** [Frederix] (2007)

From HELAS to ALOHA

- **Original HELicity Amplitude Subroutine library**
  [Murayama, Watanabe, Hagiwara]

- **One routine per Lorentz structure**
  - ➡**MSSM** [cho, al] hep-ph/0601063 (2006)
  - ➡**HEFT** [Frederix] (2007)
  - ➡**Spin 2** [Hagiwara, al] 0805.2554 (2008)

From HELAS to ALOHA

- **Original HELicity Amplitude Subroutine library**
  [Murayama, Watanabe, Hagiwara]

- **One routine per Lorentz structure**
  - ➡**MSSM** [cho, al] hep-ph/0601063 (2006)
  - ➡**HEFT** [Frederix] (2007)
  - ➡**Spin 2** [Hagiwara, al] 0805.2554 (2008)
  - ➡**Spin 3/2** [Mawatari, al] 1101.1289 (2011)

From HELAS to ALOHA

- **Original HELicity Amplitude Subroutine library**
  [Murayama, Watanabe, Hagiwara]

- **One routine per Lorentz structure**
  - ➡**MSSM** [cho, al] hep-ph/0601063 (2006)
  - ➡**HEFT** [Frederix] (2007)
  - ➡**Spin 2** [Hagiwara, al] 0805.2554 (2008)
  - ➡**Spin 3/2** [Mawatari, al] 1101.1289 (2011)

Chiral Perturbation

SLIH

BNV Model

Effective Field Theory

NMSSM

Chromo-magnetic operator

Full HEFT

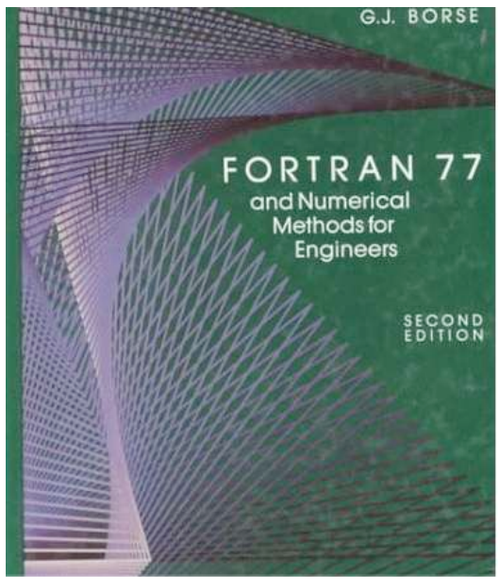
Black Holes

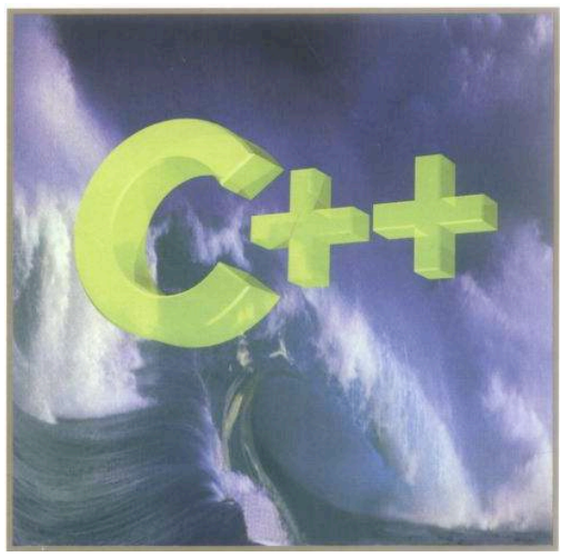
ALOHA

ALOHA
Google translate

From: UFO To: Helicity Translate

Type text or a website address or translate a document.





ALOHA

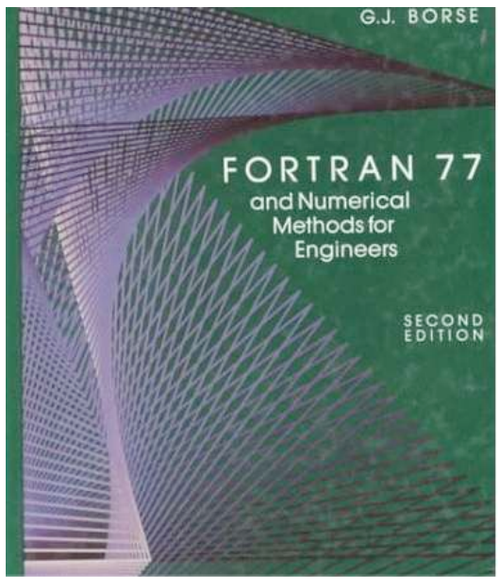
ALOHA

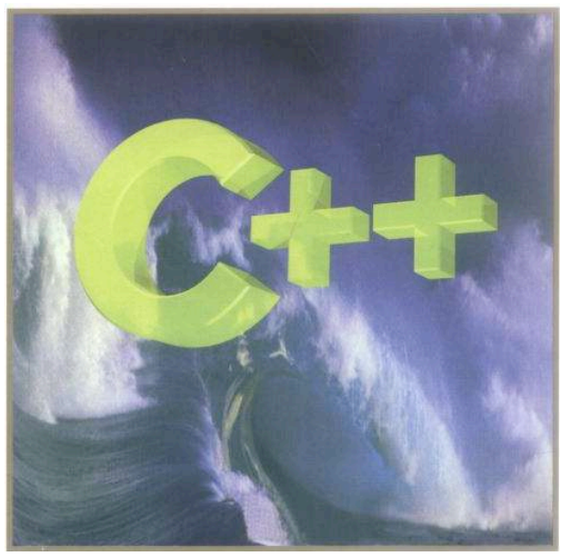
~~Google~~ translate

From: UFO To: Helicity Translate

Basically, any new operator can now be handled by MG5_aMC/Pythia8 out of the box

Type text or a website address or translate a document.





ALOHA, concretely

Input

```
FFV1 = Lorentz(name = 'FFV1',
               spins = [ 2, 2, 3 ],
               structure = 'Gamma(3,2,1)')
```

Output

```
C     This File is Automatically generated by ALOHA
C     The process calculated in this file is:
C     Gamma(3,2,1)
C
      SUBROUTINE FFV1_0(F1,F2,V3,C,VERTEX)
      IMPLICIT NONE
      DOUBLE COMPLEX F1(6)
      DOUBLE COMPLEX F2(6)
      DOUBLE COMPLEX V3(6)
      DOUBLE COMPLEX C
      DOUBLE COMPLEX VERTEX


 VERTEX = C*( (F2(1)*( (F1(3)*( (0, -1)*V3(1)+(0, 1)*V3(4)))
$ +(F1(4)*( (0, 1)*V3(2)+V3(3)))))+( (F2(2)*( (F1(3)*( (0, 1)
$ *V3(2)-V3(3)))+(F1(4)*( (0, -1)*V3(1)+(0, -1)*V3(4)))))
$ +( (F2(3)*( (F1(1)*( (0, -1)*V3(1)+(0, -1)*V3(4)))+(F1(2)
$ *( (0, -1)*V3(2)-V3(3)))))+(F2(4)*( (F1(1)*( (0, -1)*V3(2)
$ +V3(3)))+(F1(2)*( (0, -1)*V3(1)+(0, 1)*V3(4)))))))

      END
```

How does it look put together

matrix.f

```
C     ----------
C     BEGIN CODE
C     ----------
      CALL IXXXXX(P(0,1),ZERO,NHEL(1),+1*IC(1),W(1,1))
      CALL OXXXXX(P(0,2),ZERO,NHEL(2),-1*IC(2),W(1,2))
      CALL IXXXXX(P(0,3),ZERO,NHEL(3),-1*IC(3),W(1,3))
      CALL OXXXXX(P(0,4),ZERO,NHEL(4),+1*IC(4),W(1,4))
      CALL VXXXXX(P(0,5),ZERO,NHEL(5),+1*IC(5),W(1,5))
      CALL FFV1_2(W(1,1),W(1,5),GC_11,ZERO,ZERO,W(1,6))
      CALL FFV1P0_3(W(1,3),W(1,4),GC_3,ZERO,ZERO,W(1,7))
C     Amplitude(s) for diagram number 1
      CALL FFV1_0(W(1,6),W(1,2),W(1,7),GC_2,AMP(1))
      CALL FFV2_4_3(W(1,3),W(1,4),GC_50,GC_59,MDL_MZ,MDL_WZ,W(1,8))
C     Amplitude(s) for diagram number 2
      CALL FFV2_5_0(W(1,6),W(1,2),W(1,8),GC_51,GC_58,AMP(2))
      CALL FFV1_1(W(1,2),W(1,5),GC_11,ZERO,ZERO,W(1,6))
C     Amplitude(s) for diagram number 3
      CALL FFV1_0(W(1,1),W(1,6),W(1,7),GC_2,AMP(3))
C     Amplitude(s) for diagram number 4
      CALL FFV2_5_0(W(1,1),W(1,6),W(1,8),GC_51,GC_58,AMP(4))
      JAMP(1)=+AMP(1)+AMP(2)+AMP(3)+AMP(4)

      MATRIX = 0.D0
      DO I = 1, NCOLOR
        ZTEMP = (0.D0,0.D0)
        DO J = 1, NCOLOR
          ZTEMP = ZTEMP + CF(J,I)*JAMP(J)
        ENDDO
        MATRIX = MATRIX+ZTEMP*DCONJG(JAMP(I))/DENOM(I)
      ENDDO
```

Plan

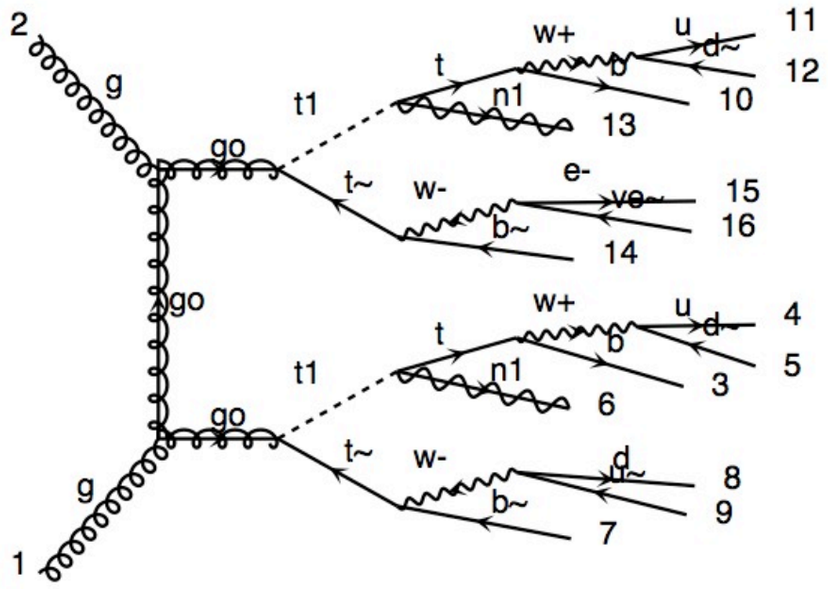
- Build your model : FeynRules + UFO
- Computing amplitudes with Feynman diagrams
- Special techniques relevant to BSM
- Simulations at NLO accuracy
- MadLoop : evaluating loop Matrix Elements
- Recent loopy applications within MG5_aMC

Plan

- Build your model : FeynRules + UFO
- Computing amplitudes with Feynman diagrams
- Special techniques relevant to BSM
- Simulations at NLO accuracy
- MadLoop : evaluating loop Matrix Elements
- Recent loopy applications within MG5_aMC

Decays

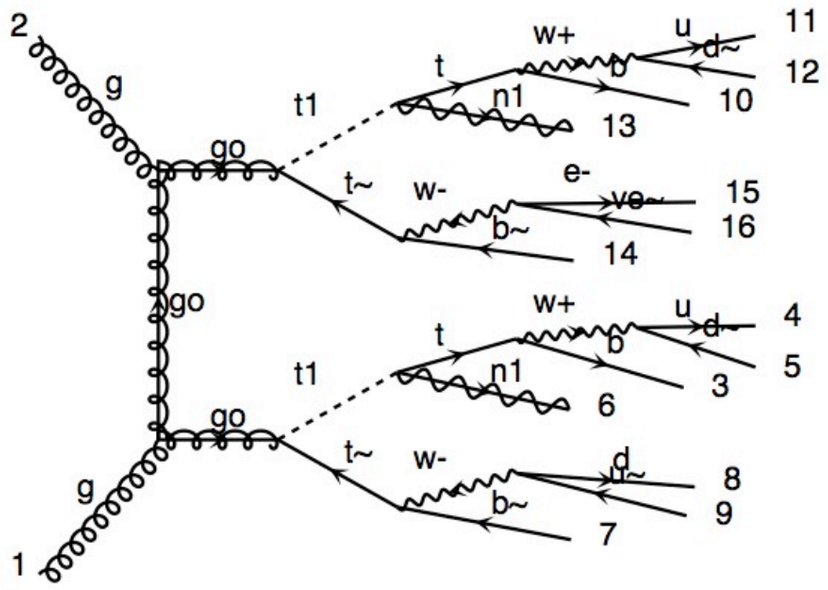
SUSY Case



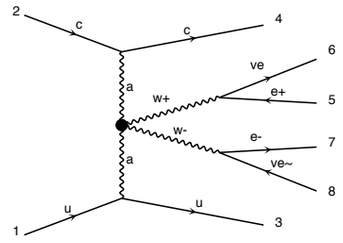
EFT Case

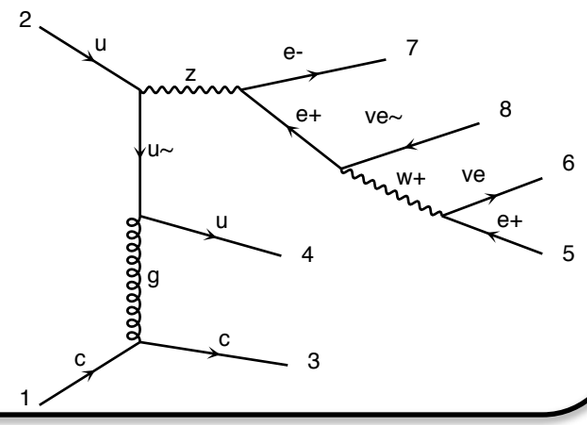
Decays

SUSY Case



EFT Case



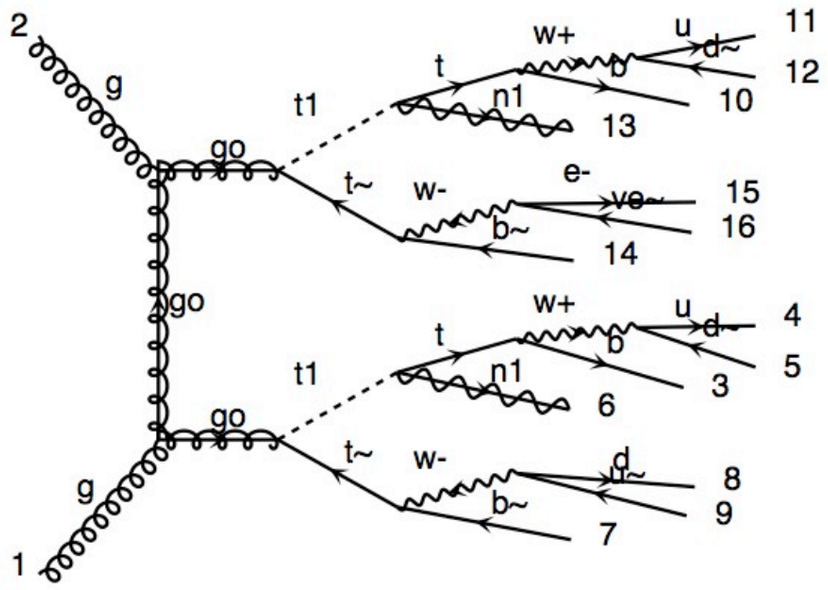


Problem

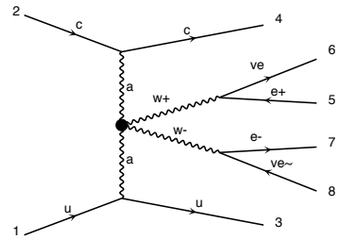
- Processes often too complicated to be able to retain all the off-shell contributions!

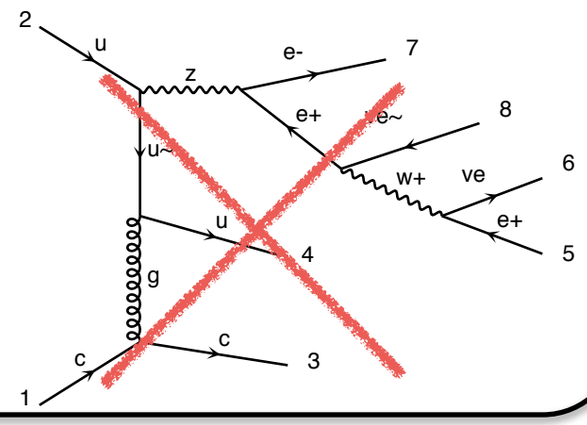
Decays

SUSY Case



EFT Case





Problem

- Processes often too complicated to be able to retain all the off-shell contributions!

Solution

- Well, only keep on-shell contributions!

Narrow-Width Approx.

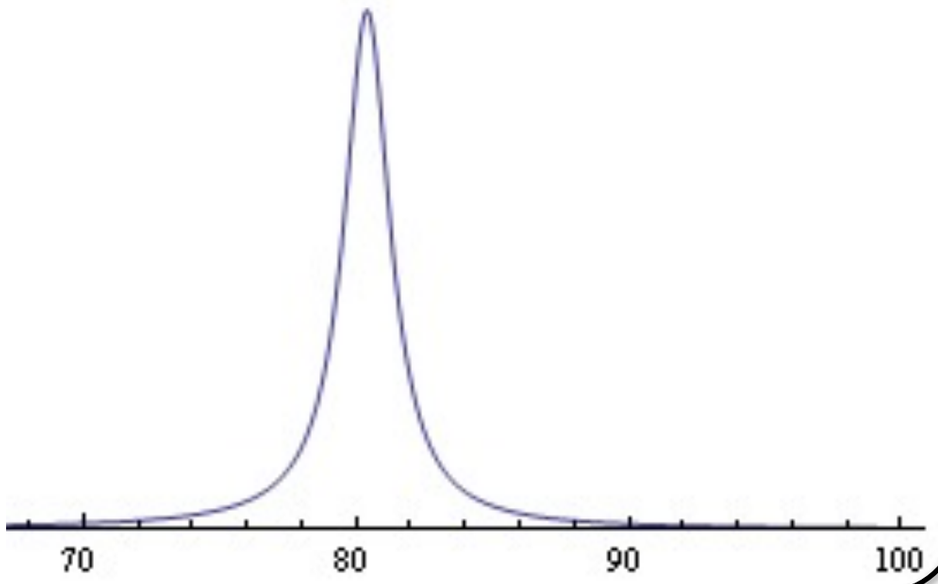
Theory

$$\int dq^2 \left| \frac{1}{q^2 - M^2 - iM\Gamma} \right|^2 \approx \frac{\pi}{M\Gamma}$$

$$\sigma_{full} = \sigma_{prod} * (BR + \mathcal{O}(\frac{\Gamma}{M}))$$

Comment

$$f(E) \sim \frac{1}{\left(E^2 - M^2\right)^2 + M^2\Gamma^2}.$$



Narrow-Width Approx.

Theory

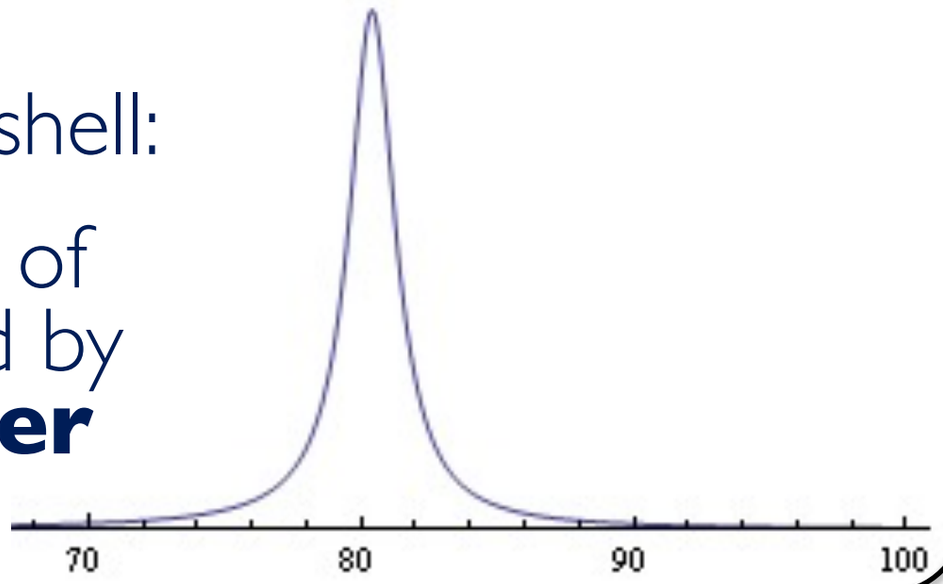
$$\int dq^2 \left| \frac{1}{q^2 - M^2 - iM\Gamma} \right|^2 \approx \frac{\pi}{M\Gamma}$$

$$\sigma_{full} = \sigma_{prod} * (BR + \mathcal{O}(\frac{\Gamma}{M}))$$

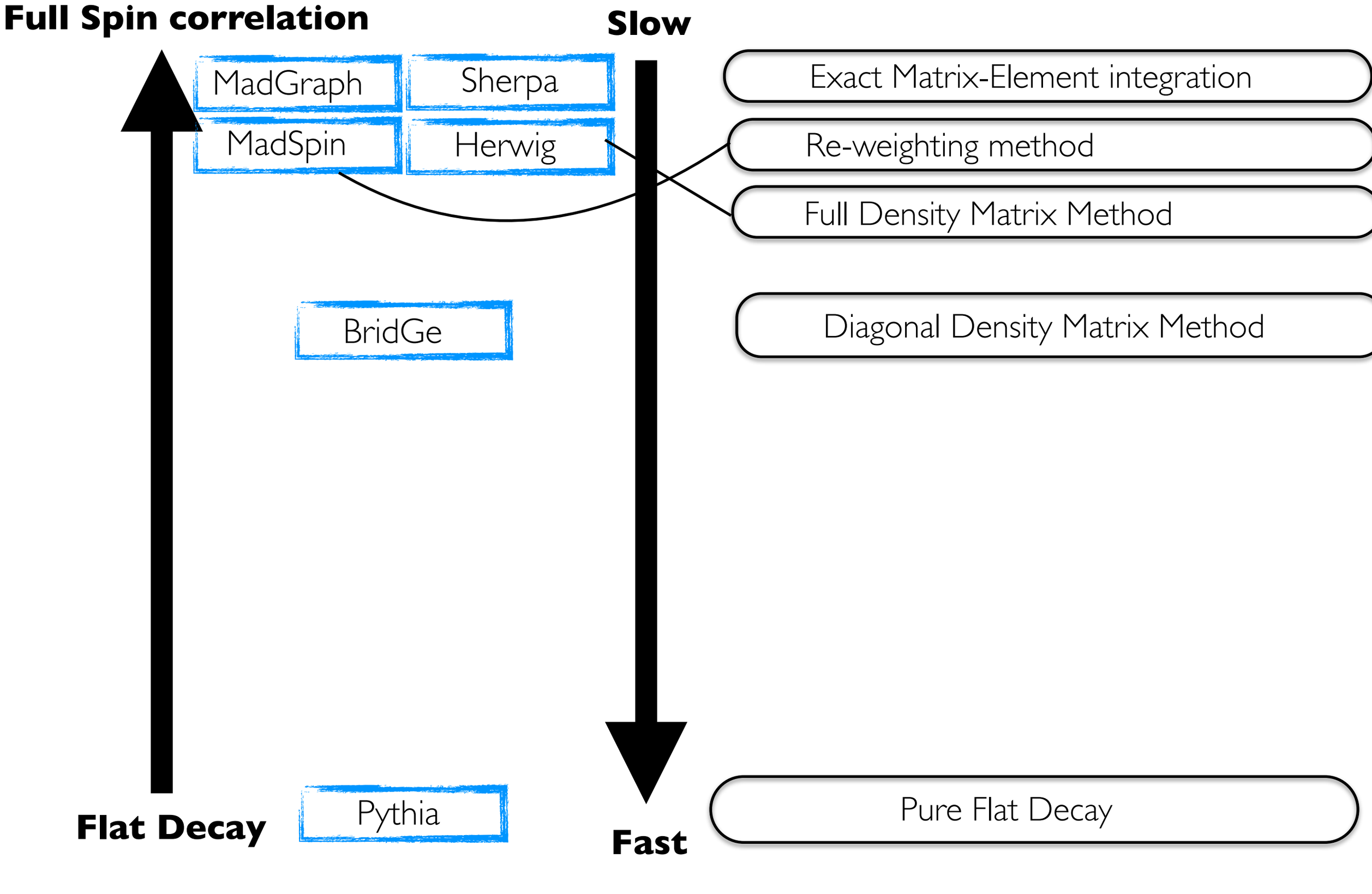
Comment

- This is an Approximation!
- This also forces the particle to be on-shell:
  - ➡ Some of the lost kinematic shape of the propagator can be recovered by re-introducing the **Breit-Wigner**

$$f(E) \sim \frac{1}{(E^2 - M^2)^2 + M^2\Gamma^2}.$$

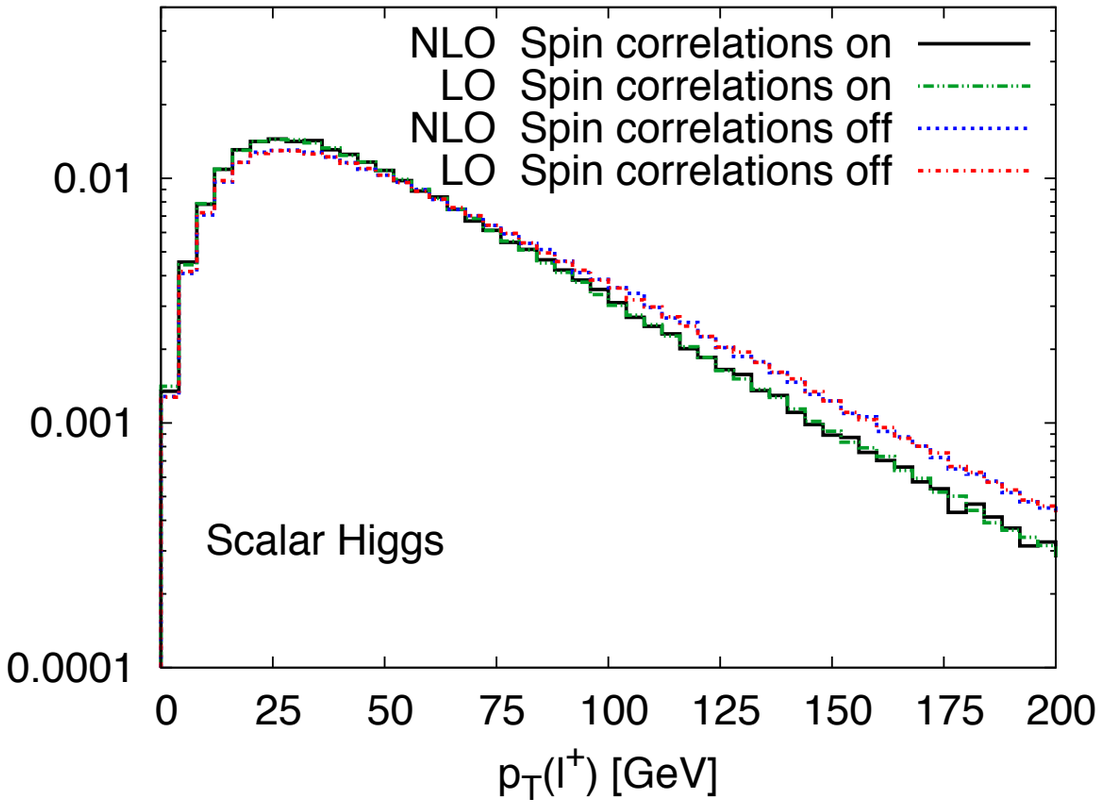


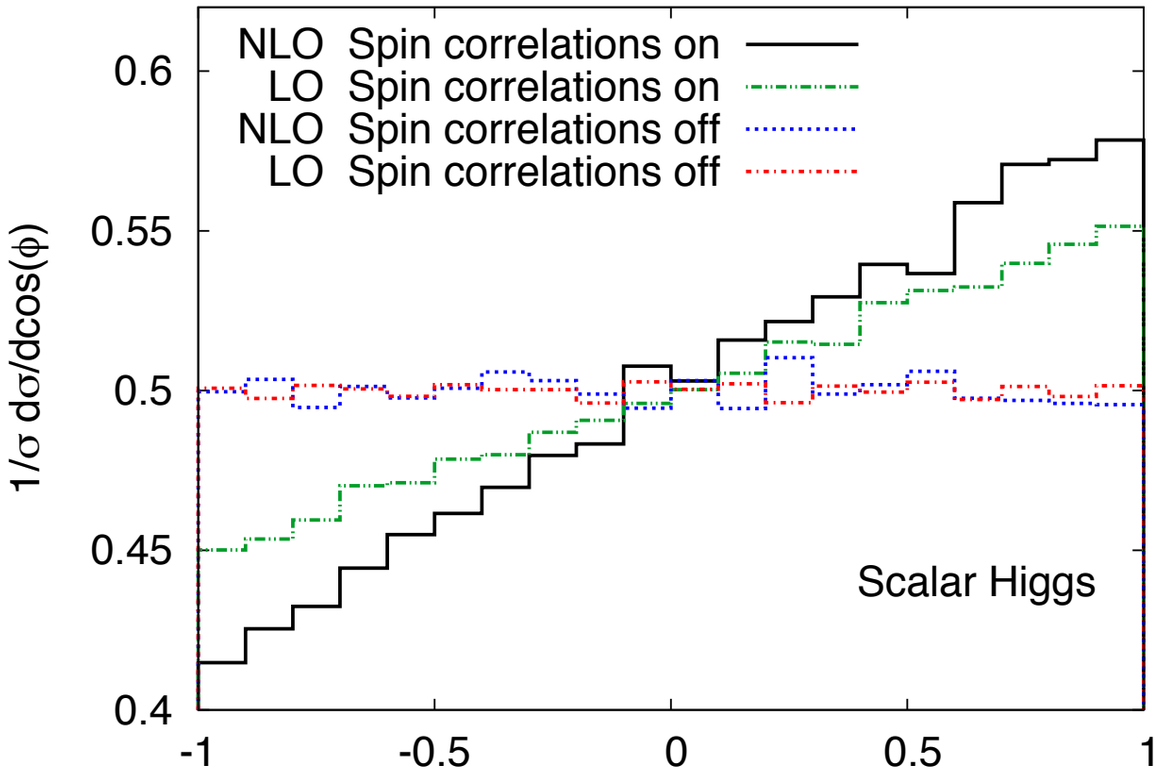
Restoring spin-correlations

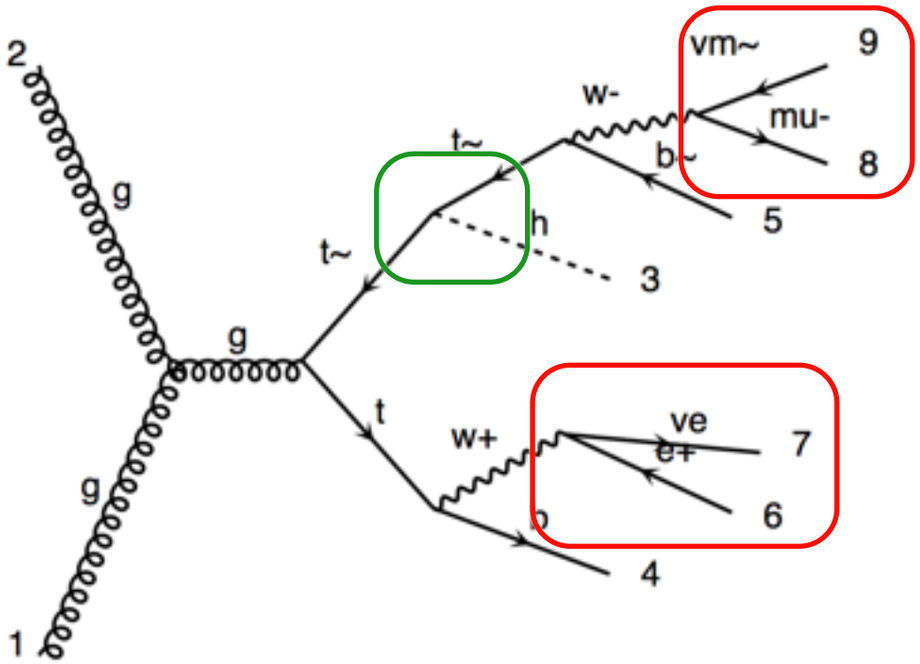


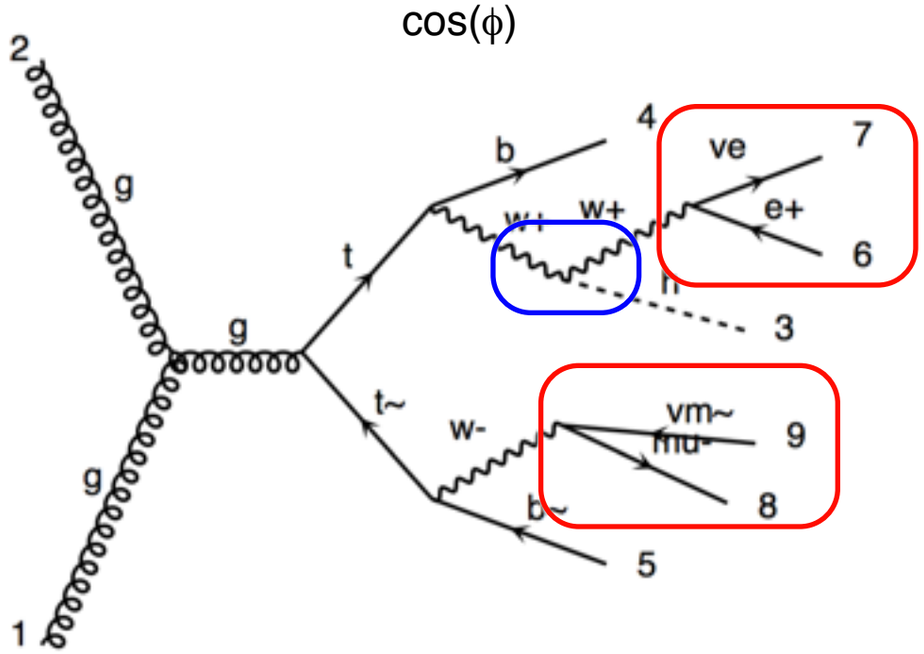
Relevance of spin-correlations

$$p\,p \;\to\; t\bar{t}H \;\to\; b\bar{b}e^{+}\nu_e\mu^{-}\bar{\nu}_\mu$$









MadSpin

[Frixione, Leanen, Motylinski,Webber (2007)]

[Artoisenet, OM et al. 1212.3460]

One Event

offshell	spin	unweighted
No	No	YES

MadSpin

[Frixione, Leanen, Motylinski,Webber (2007)]

[Artoisenet, OM et al. 1212.3460]

offshell	spin	unweighted
No	No	YES

One Event

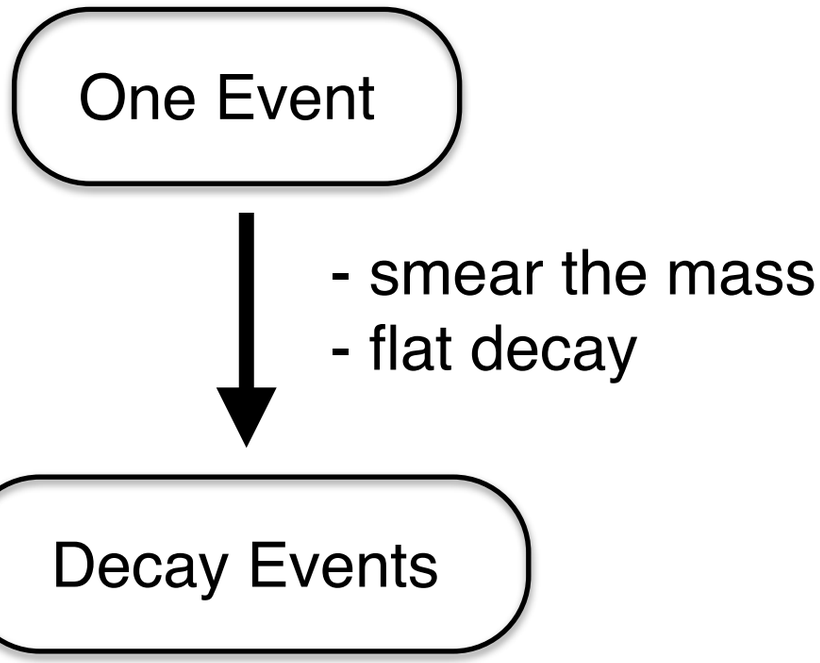
- smear the mass
- flat decay

Decay Events

MadSpin

[Frixione, Leanen, Motylinski,Webber (2007)]

[Artoisenet, OM et al. 1212.3460]

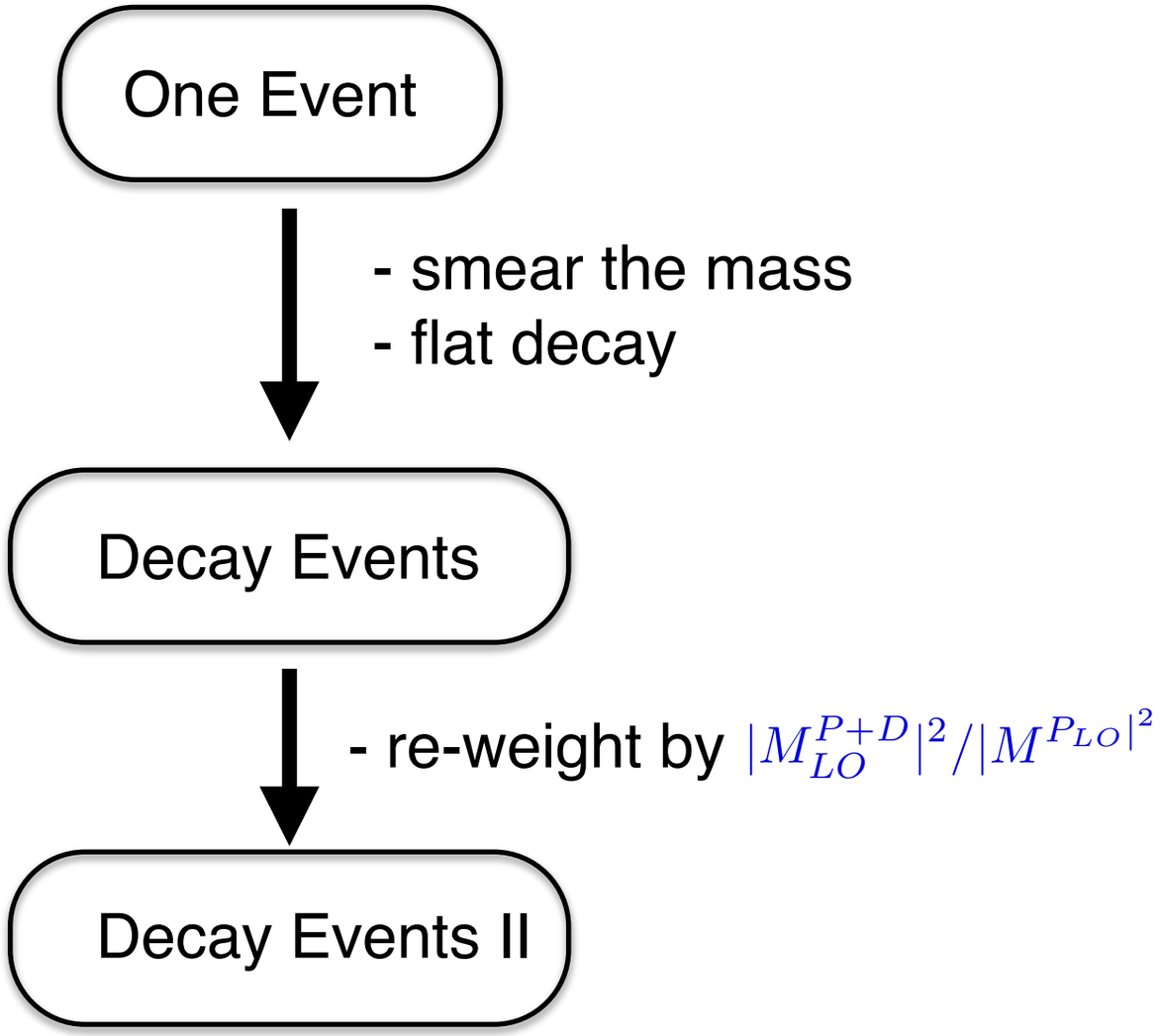


offshell	spin	unweighted
No	No	YES
YES	No	No

MadSpin

[Frixione, Leanen, Motylinski,Webber (2007)]

[Artoisenet, OM et al. 1212.3460]

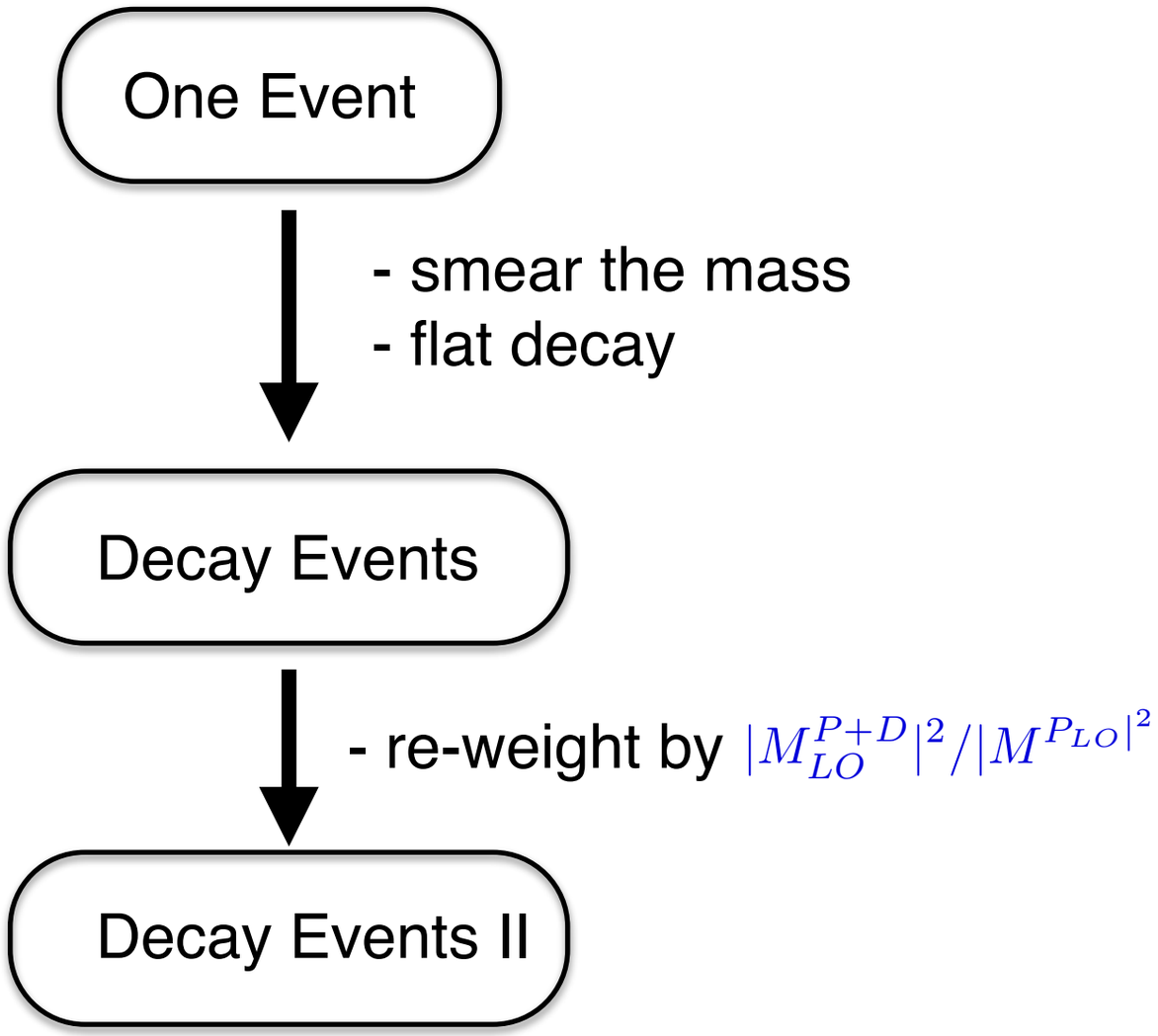


offshell	spin	unweighted
No	No	YES
YES	No	No

MadSpin

[Frixione, Leanen, Motylinski,Webber (2007)]

[Artoisenet, OM et al. 1212.3460]

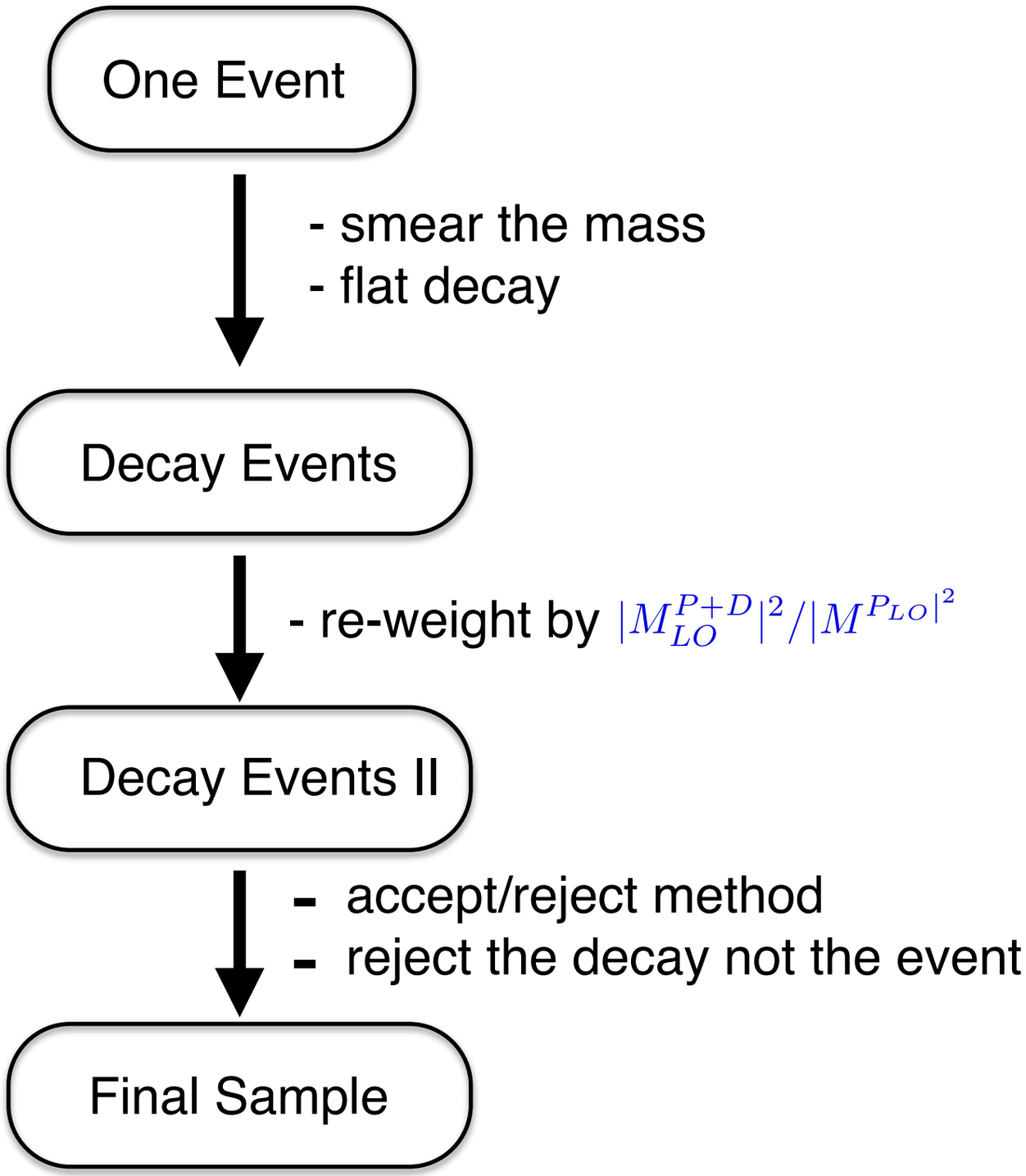


offshell	spin	unweighted
No	No	YES
YES	No	No
YES	YES	No

MadSpin

[Frixione, Leanen, Motylinski,Webber (2007)]

[Artoisenet, OM et al. 1212.3460]

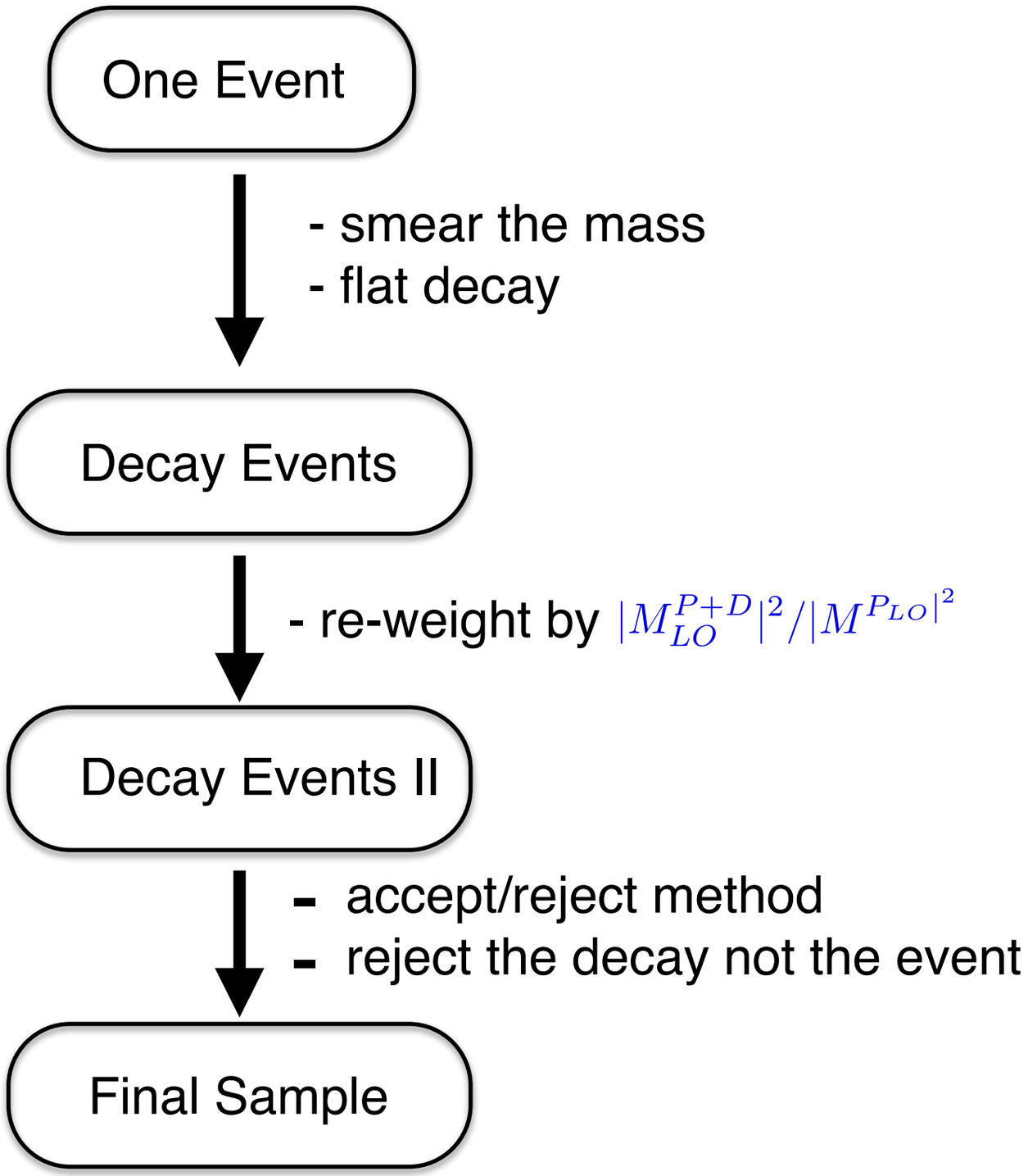


offshell	spin	unweighted
No	No	YES
YES	No	No
YES	YES	No

MadSpin

[Frixione, Leanen, Motylinski,Webber (2007)]

[Artoisenet, OM et al. 1212.3460]



offshell	spin	unweighted
No	No	YES
YES	No	No
YES	YES	No
YES	YES	YES

Model Scans

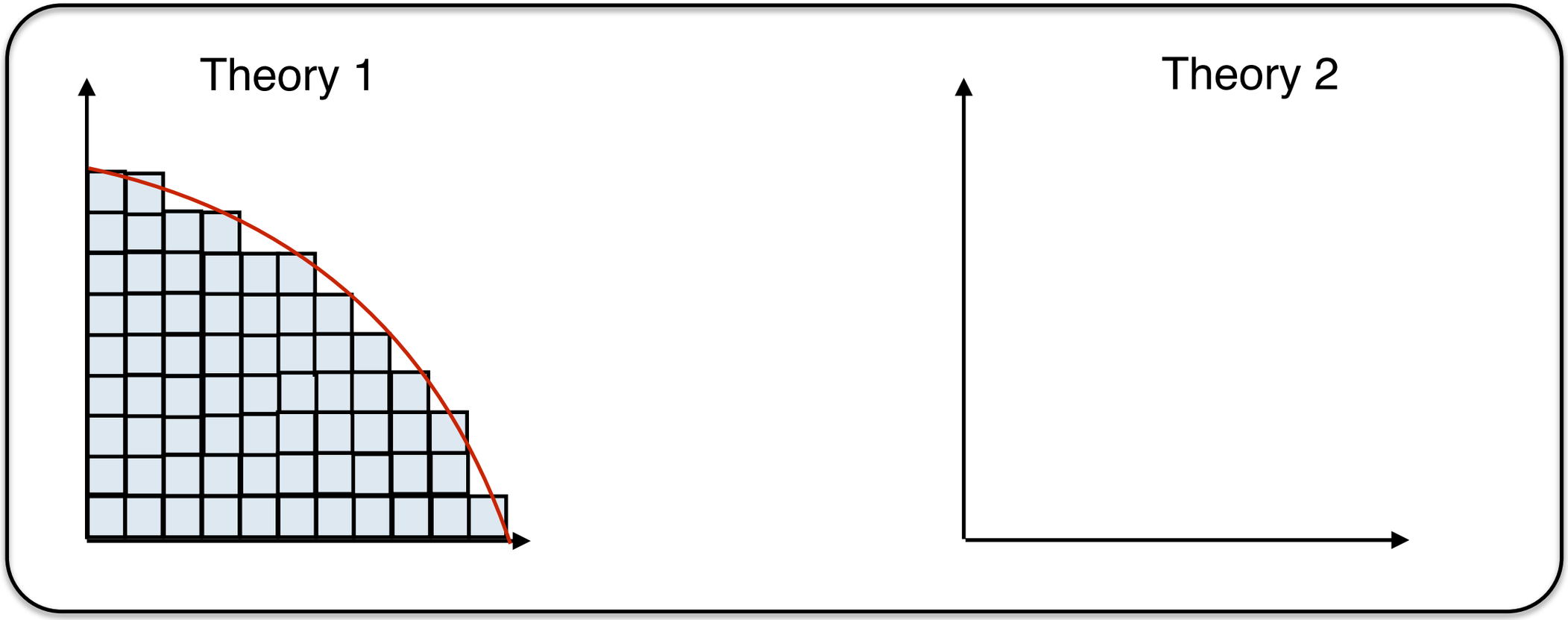
Idea

- Reuse the sample (Only one Full Sim)
- Change the weight of the events

$$W_{new} = \frac{|M_{new}|^2}{|M_{old}|^2} * W_{old}$$

1405.0301

1404.7129



Model Scans

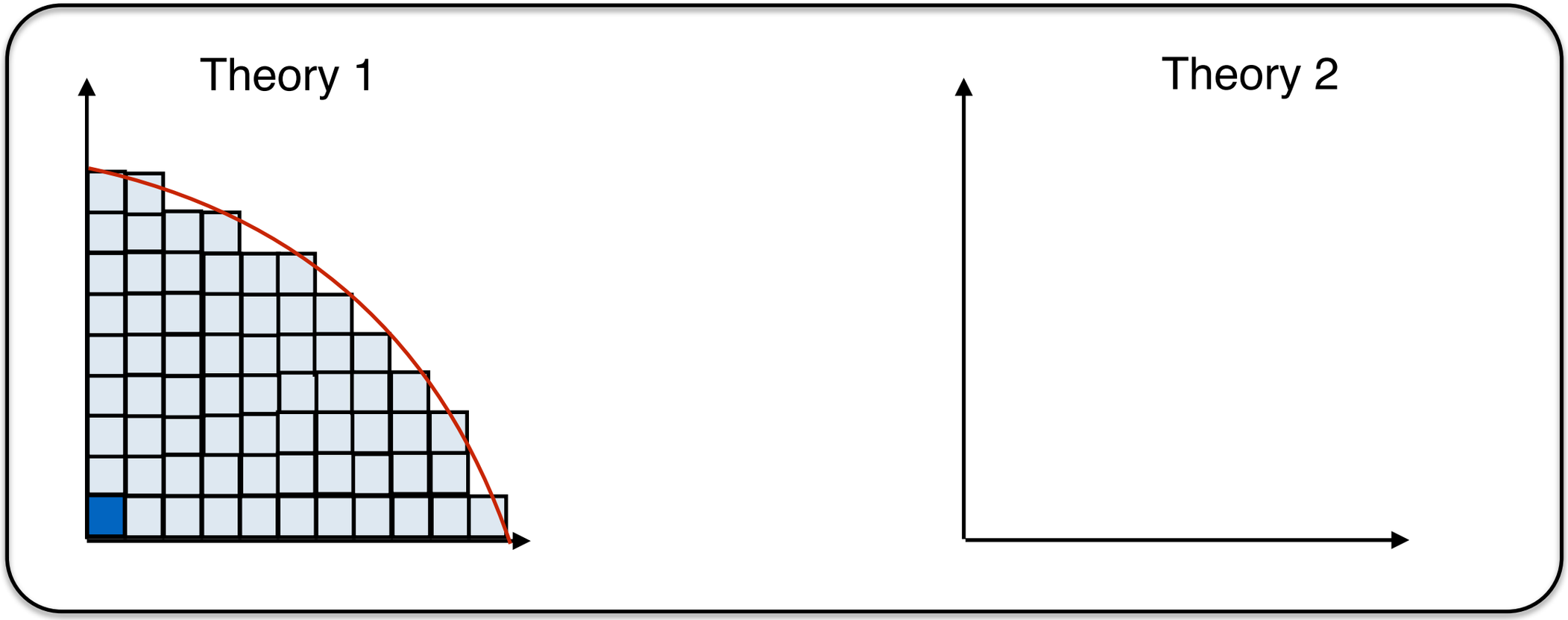
Idea

- Reuse the sample (Only one Full Sim)
- Change the weight of the events

$$W_{new} = \frac{|M_{new}|^2}{|M_{old}|^2} * W_{old}$$

1405.0301

1404.7129



Model Scans

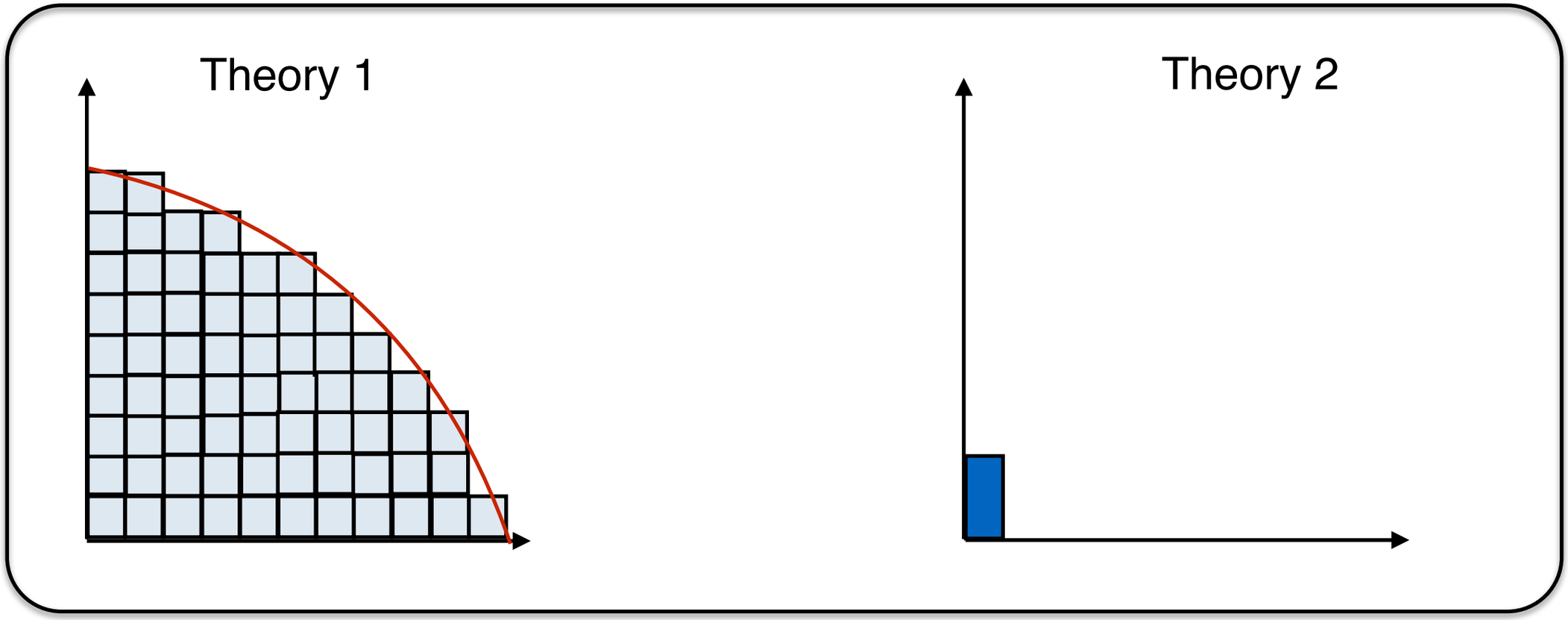
Idea

- Reuse the sample (Only one Full Sim)
- Change the weight of the events

$$W_{new} = \frac{|M_{new}|^2}{|M_{old}|^2} * W_{old}$$

1405.0301

1404.7129



Model Scans

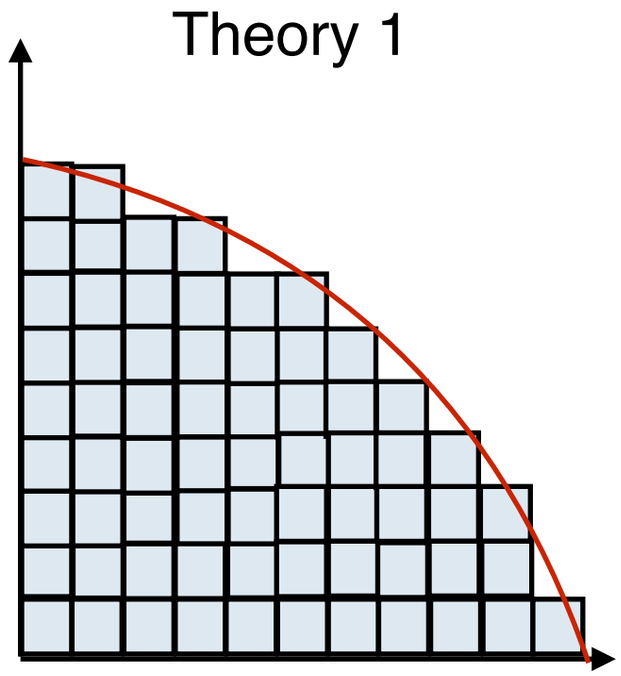
Idea

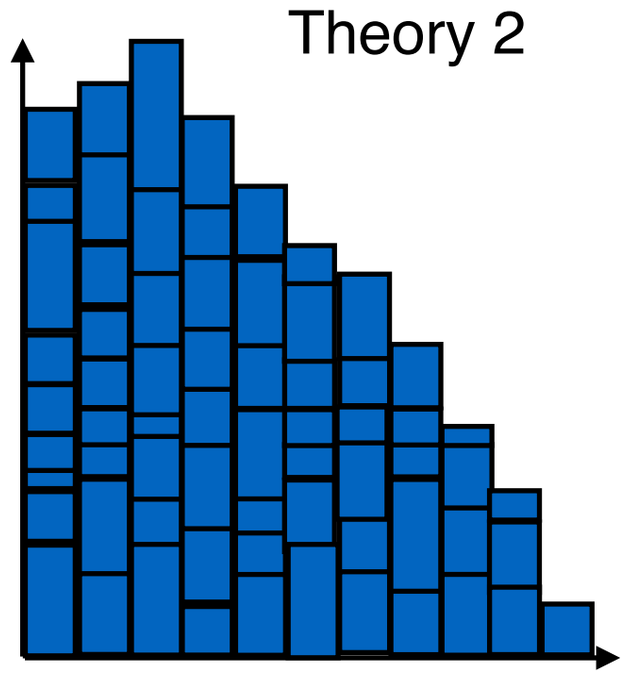
- Reuse the sample (Only one Full Sim)
- Change the weight of the events

$$W_{new} = \frac{|M_{new}|^2}{|M_{old}|^2} * W_{old}$$

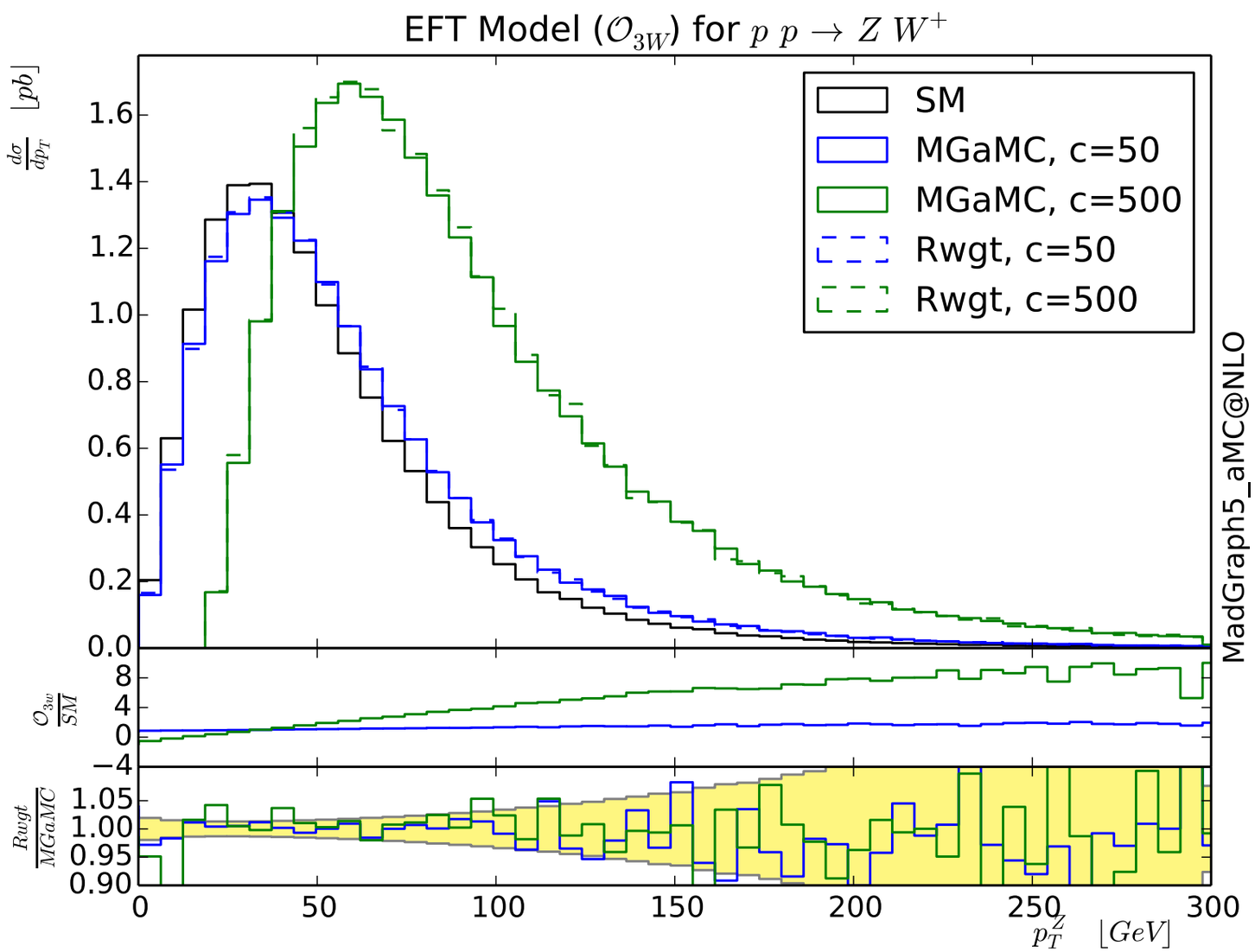
1405.0301

1404.7129



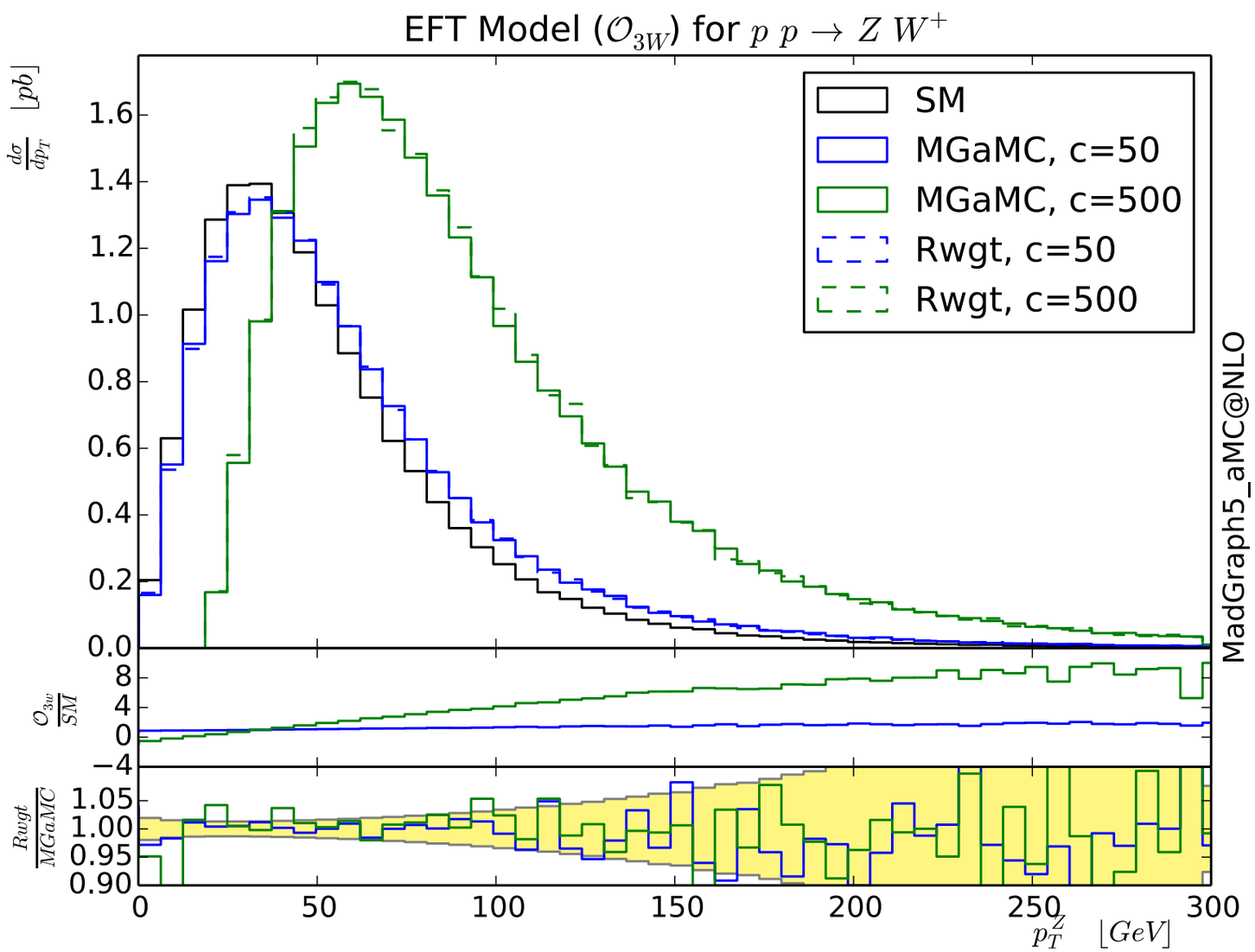


Reweighting validation



EFT Case $\mathcal{O}_{3W} = Tr\left[W_{\mu\nu}W^{\nu\rho}W_{\rho}{}^{\mu}\right]$

Reweighting validation



EFT Case $\mathcal{O}_{3W} = Tr\left[W_{\mu\nu}W^{\nu\rho}W_{\rho}{}^{\mu}\right]$

- **Word of caution:** Care for additional information written in the LHE file: helicity is OK, but color and intermediate particle isn't.

Th. Uncertainty

- New module in pure python
  - Rely to lhapdf interface trough python
- faster, more flexible

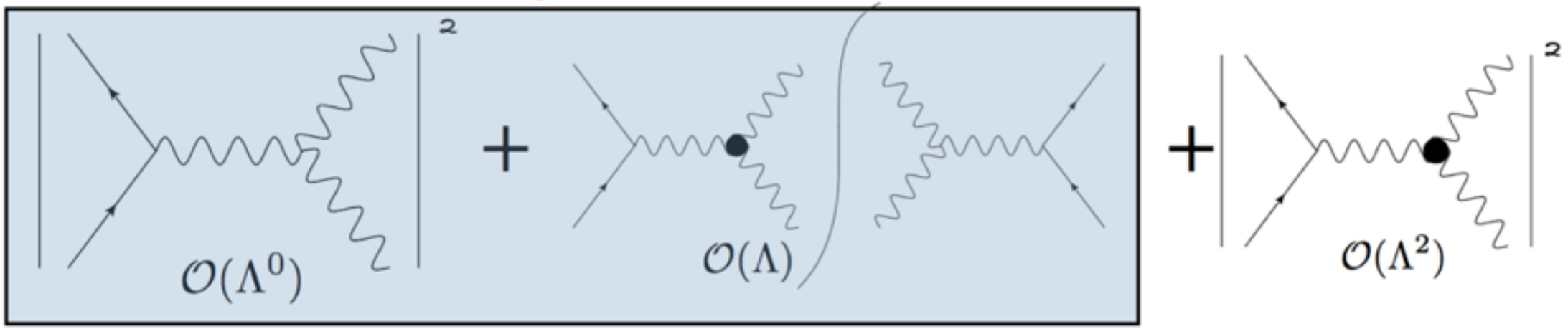
- generate p p > t t~
- output
- launch

- Body Level One
  - Body Level Two
    - Body Level Three
      - Body Level Four
        - Body Level Five

```
INFO: # events generated with PDF: NNPDF23_lo_as_0130_qed (247000)
INFO: #Will Compute 144 weights per event.
INFO: #*********************************************************************
#
# original cross-section: 505.430386046
#     scale variation: +30.2% -28.6%
#     central scheme variation: +3.36e-09% -25.9%
# PDF variation: +3.77% -3.77%
#
# dynamical scheme # 1 : 479.713 +29.9% -21.6% # \sum ET
# dynamical scheme # 2 : 395.728 +27.7% -20.5% # \sum\sqrt{m^2+pt^2}
# dynamical scheme # 3 : 505.43 +30.2% -21.7% # 0.5 \sum\sqrt{m^2+pt^2}
# dynamical scheme # 4 : 374.366 +27.1% -20.1% # \sqrt{\hat s}
#*********************************************************************
```

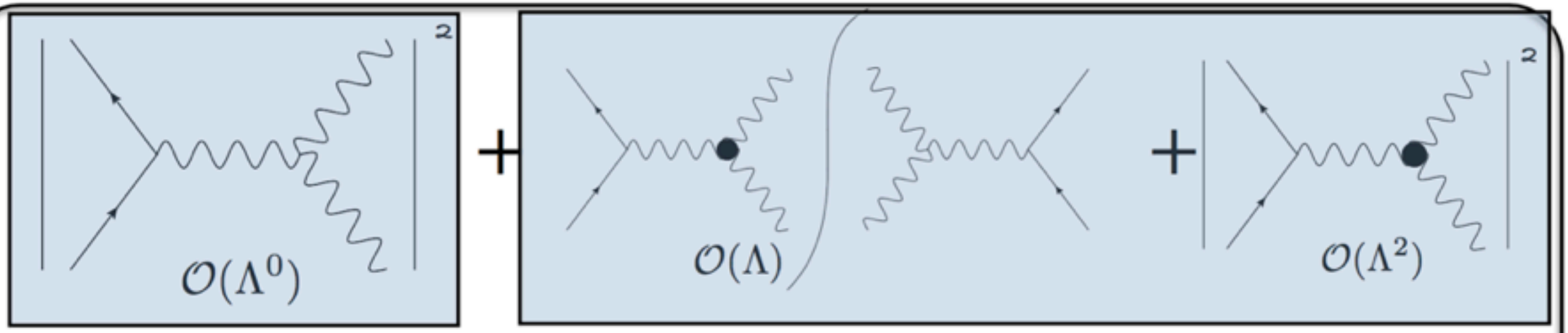
Interference contributions

Effective Field Theory:



Signal

Equivalent to Dimension 8 contribution



SM
Model independent
Dominant

BSM
Model dependent
Sub-Dominant

Get Involved!

https://cp3.irmp.ucl.ac.be/projects/madgraph/wiki/Plugin

MG5_aMC is a **framework** to **develop new ideas** for HEP,

Develop your own **PLUGIN** to **implement** them!

→ **Python** is an **easy-to-use** and **flexible** language

→ We provide **support** via our **launchpad** platform:

https://answers.launchpad.net/mg5amcnlo

→ Dev. and maintenance **independent** from **MG5_aMC**

So **authorship** of the PLUGIN is **properly credited**.

Simplest plugin implementation:

```
import madgraph.interface.master_interface as \
                                    master_interface

class NewInterface(master_interface.MasterCmd):

    def do_helloworld(self, line):
        """print hello world"""
        print "hello world " + line
```

→

```
./bin/mg5_aMC --mode=helloworld

MG5_aMC > helloworld ciaoTutti

hello world ciatoTutti
```

Plan

- Build your model : FeynRules + UFO
- Computing amplitudes with Feynman diagrams
- Special techniques relevant to BSM
- Simulations at NLO accuracy
- MadLoop : evaluating loop Matrix Elements
- Recent loopy applications within MG5_aMC

Plan

- Build your model : FeynRules + UFO
- Computing amplitudes with Feynman diagrams
- Special techniques relevant to BSM
- Simulations at NLO accuracy
- MadLoop : evaluating loop Matrix Elements
- Recent loopy applications within MG5_aMC

Perturbative Expansion

$\hat{\sigma}_{ab\to X}(\hat{s}, \mu_F, \mu_R)$ Parton-level cross section

- The parton-level cross section can be computed as a series in perturbation theory, using the coupling constant as an expansion parameter

$$\hat{\sigma} = \sigma^{\text{Born}}\left(1 + \frac{\alpha_s}{2\pi}\sigma^{(1)} + \left(\frac{\alpha_s}{2\pi}\right)^2\sigma^{(2)} + \left(\frac{\alpha_s}{2\pi}\right)^3\sigma^{(3)} + \dots\right)$$

Perturbative Expansion

$\hat{\sigma}_{ab\to X}(\hat{s}, \mu_F, \mu_R)$ Parton-level cross section

- The parton-level cross section can be computed as a series in perturbation theory, using the coupling constant as an expansion parameter

$$\hat{\sigma} = \sigma^{\text{Born}}\left(1 + \frac{\alpha_s}{2\pi}\sigma^{(1)} + \left(\frac{\alpha_s}{2\pi}\right)^2\sigma^{(2)} + \left(\frac{\alpha_s}{2\pi}\right)^3\sigma^{(3)} + \dots\right)$$

LO predictions

Perturbative Expansion

$\hat{\sigma}_{ab\to X}(\hat{s}, \mu_F, \mu_R)$ Parton-level cross section

- The parton-level cross section can be computed as a series in perturbation theory, using the coupling constant as an expansion parameter

$$\hat{\sigma} = \sigma^{\text{Born}}\left(1 + \frac{\alpha_s}{2\pi}\sigma^{(1)} + \left(\frac{\alpha_s}{2\pi}\right)^2\sigma^{(2)} + \left(\frac{\alpha_s}{2\pi}\right)^3\sigma^{(3)} + \dots\right)$$

LO predictions

NLO corrections

Perturbative Expansion

$\hat{\sigma}_{ab\to X}(\hat{s}, \mu_F, \mu_R)$ Parton-level cross section

- The parton-level cross section can be computed as a series in perturbation theory, using the coupling constant as an expansion parameter

$$\hat{\sigma} = \sigma^{\text{Born}}\left(1 + \frac{\alpha_s}{2\pi}\sigma^{(1)} + \left(\frac{\alpha_s}{2\pi}\right)^2\sigma^{(2)} + \left(\frac{\alpha_s}{2\pi}\right)^3\sigma^{(3)} + \dots\right)$$

LO predictions

NLO corrections

NNLO corrections

Perturbative Expansion

$\hat{\sigma}_{ab\to X}(\hat{s}, \mu_F, \mu_R)$ Parton-level cross section

- The parton-level cross section can be computed as a series in perturbation theory, using the coupling constant as an expansion parameter

$$\hat{\sigma} = \sigma^{\text{Born}}\left(1 + \frac{\alpha_s}{2\pi}\sigma^{(1)} + \left(\frac{\alpha_s}{2\pi}\right)^2\sigma^{(2)} + \left(\frac{\alpha_s}{2\pi}\right)^3\sigma^{(3)} + \dots\right)$$

LO predictions

NLO corrections

NNLO corrections

NNNLO corrections

Perturbative Expansion

$\hat{\sigma}_{ab\to X}(\hat{s}, \mu_F, \mu_R)$ Parton-level cross section

- The parton-level cross section can be computed as a series in perturbation theory, using the coupling constant as an expansion parameter

$$\hat{\sigma} = \sigma^{\text{Born}}\left(1 + \frac{\alpha_s}{2\pi}\sigma^{(1)} + \left(\frac{\alpha_s}{2\pi}\right)^2\sigma^{(2)} + \left(\frac{\alpha_s}{2\pi}\right)^3\sigma^{(3)} + \dots\right)$$

LO predictions

NLO corrections

NNLO corrections

NNNLO corrections

- Okay, but why bothering adding more 'N'?

Why adding N's ?

- Some analysis strongly rely on the capability of accurately simulating the signal (e.g. single-top)
- Lack of predictivity means that people will overstretch predictions without noticing.
- Going beyond K-factor rescaling, with fully exclusive predictions allows for example to have them go through detector simulation.
- Rule of thumbs:

  LO First order of magnitude estimate

Why adding N's ?

- Some analysis strongly rely on the capability of accurately simulating the signal (e.g. single-top)
- Lack of predictivity means that people will overstretch predictions without noticing.
- Going beyond K-factor rescaling, with fully exclusive predictions allows for example to have them go through detector simulation.
- Rule of thumbs:

LO	First order of magnitude estimate
NLO	First credible inclusive rate

Why adding N's ?

- Some analysis strongly rely on the capability of accurately simulating the signal (e.g. single-top)
- Lack of predictivity means that people will overstretch predictions without noticing.
- Going beyond K-factor rescaling, with fully exclusive predictions allows for example to have them go through detector simulation.
- Rule of thumbs:

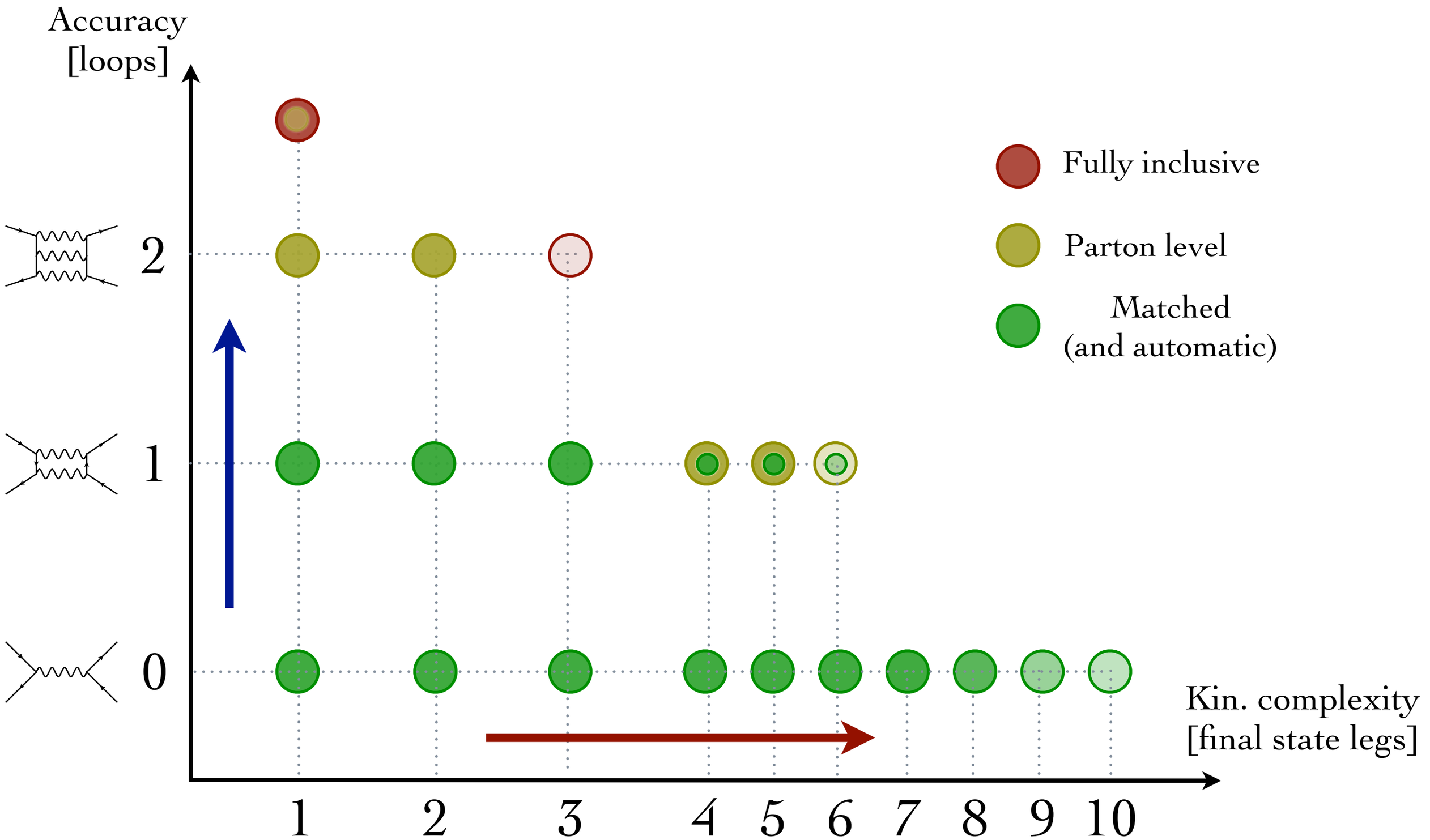
LO	First order of magnitude estimate
NLO	First credible inclusive rate
NNLO	First credible scale uncertainty

Why adding N's ?

- Some analysis strongly rely on the capability of accurately simulating the signal (e.g. single-top)
- Lack of predictivity means that people will overstretch predictions without noticing.
- Going beyond K-factor rescaling, with fully exclusive predictions allows for example to have them go through detector simulation.
- Rule of thumbs:

LO	First order of magnitude estimate
NLO	First credible inclusive rate
NNLO	First credible scale uncertainty
NNNLO	First that time it usually doesn't matter....

How to add N's ?


Accuracy
[loops]
2
1
0
Fully inclusive
Parton level
Matched
(and automatic)
Kin. complexity
[final state legs]
1
2
3
4
5
6
7
8
9
10

Adding multiplicities ?

- Better jet description

- Probing of new initial state channels opening up

- Most the NLO impact on the shape of distributions comes from the real-emissions diagrams, *i.e.* trees

Adding Loops ?

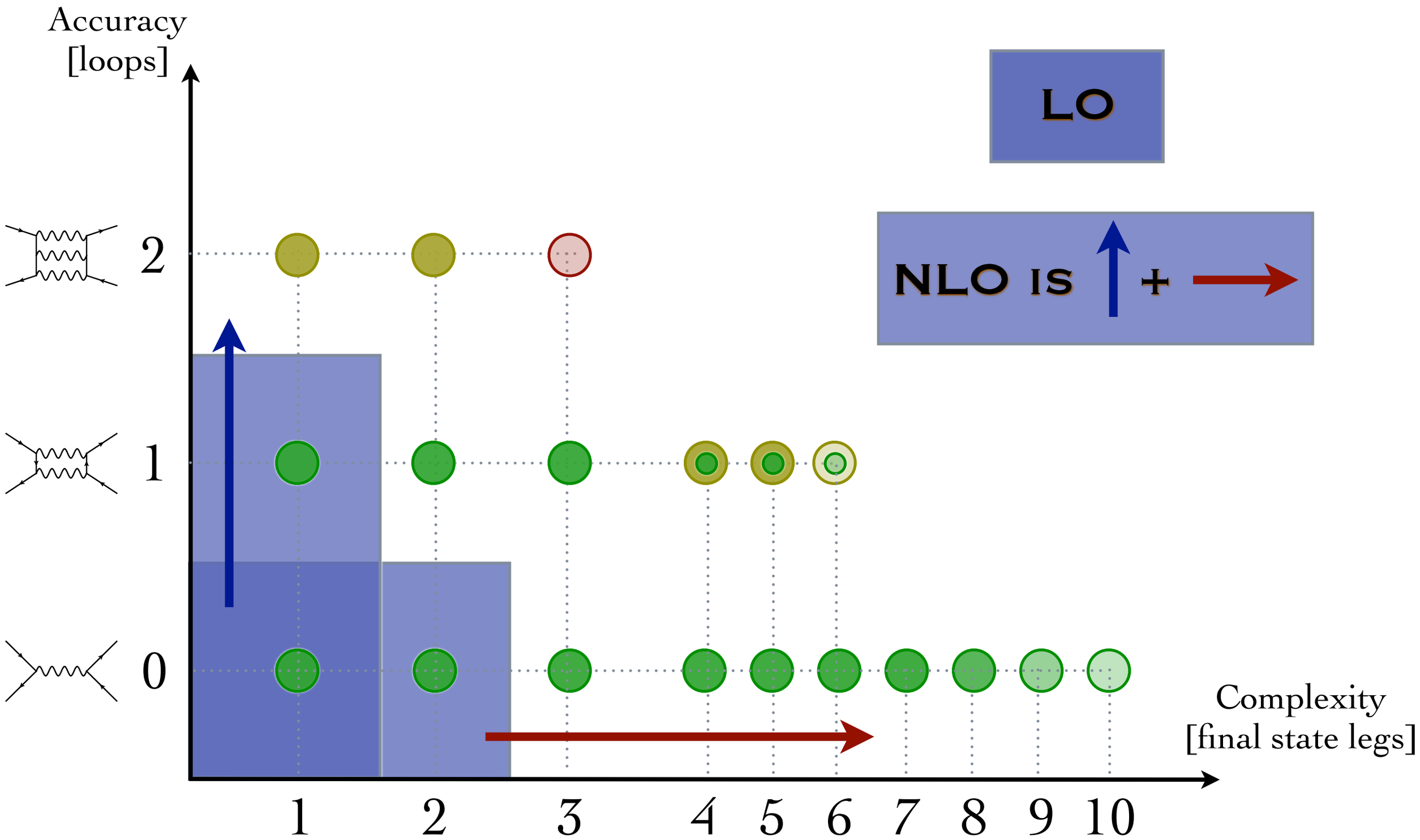
- Meaningful assessment of theoretical uncertainties with scales variation (μ_R , μ_F)

- Credible total rates predictions

- (Treat loop-induced processes without effective theories)

- Necessary for parameters extraction from measurements (*i.e* precision physics)

Computing NLO



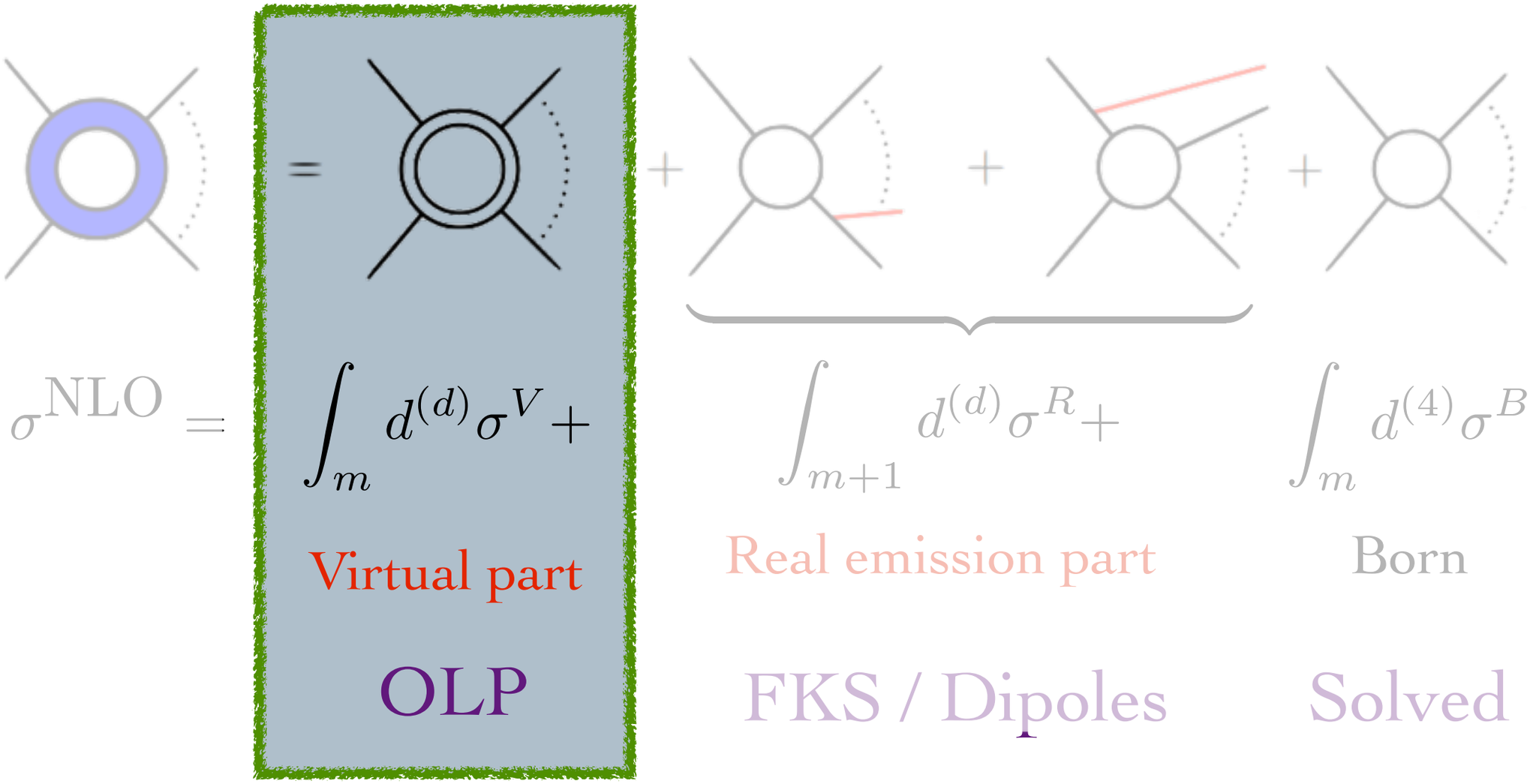
Plan

- Build your model : FeynRules + UFO
- Computing amplitudes with Feynman diagrams
- Special techniques relevant to BSM
- Simulations at NLO accuracy
- MadLoop : evaluating loop Matrix Elements
- Recent loopy applications within MG5_aMC

Plan

- Build your model : FeynRules + UFO
- Computing amplitudes with Feynman diagrams
- Special techniques relevant to BSM
- Simulations at NLO accuracy
- MadLoop : evaluating loop Matrix Elements
- Recent loopy applications within MG5_aMC

Loop computations



$$\sigma^{\mathrm{NLO}} = \int_m d^{(d)}\sigma^V + \int_{m+1} d^{(d)}\sigma^R + \int_m d^{(4)}\sigma^B$$

Virtual part — Real emission part — Born

OLP — FKS / Dipoles — Solved

Generating loop diagrams

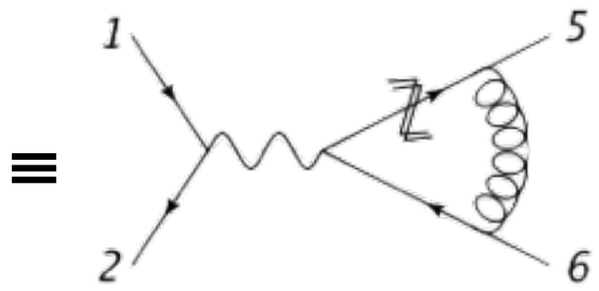
- No external tool for loop diagram generation:
  Reuse MG5_aMC efficient tree level diagram generation!
- Cut loops have two extra external particles

Trees ($e^+e^- \to u\,u\sim\,u\,u\sim$) ≡ Loops ($e^+e^- \to u\,u\sim$)

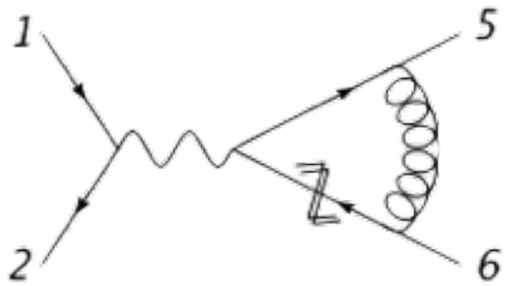
Generating loop diagrams

- No external tool for loop diagram generation:
  Reuse MG5_aMC efficient tree level diagram generation!
- Cut loops have two extra external particles

Trees ($e^+e^- \to u\,u\sim\,u\,u\sim$) ≡ Loops ($e^+e^- \to u\,u\sim$)



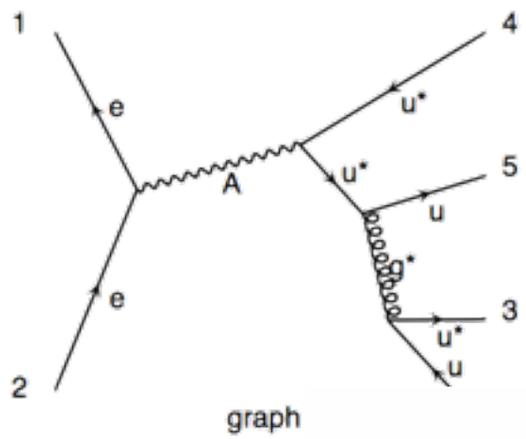
≡



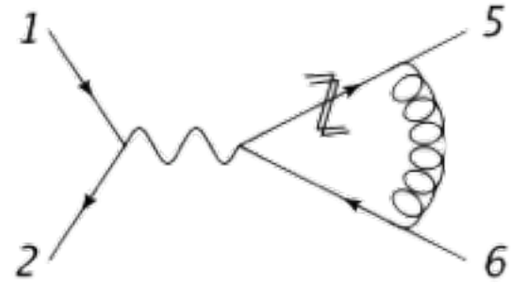
Generating loop diagrams

- No external tool for loop diagram generation:
  Reuse MG5_aMC efficient tree level diagram generation!
- Cut loops have two extra external particles

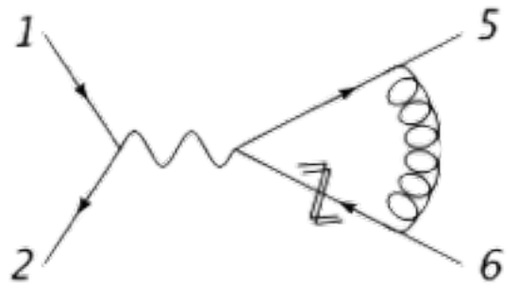
Trees ($e^+e^- \to u\,u\sim\,u\,u\sim$) ≡ Loops ($e^+e^- \to u\,u\sim$)



≡



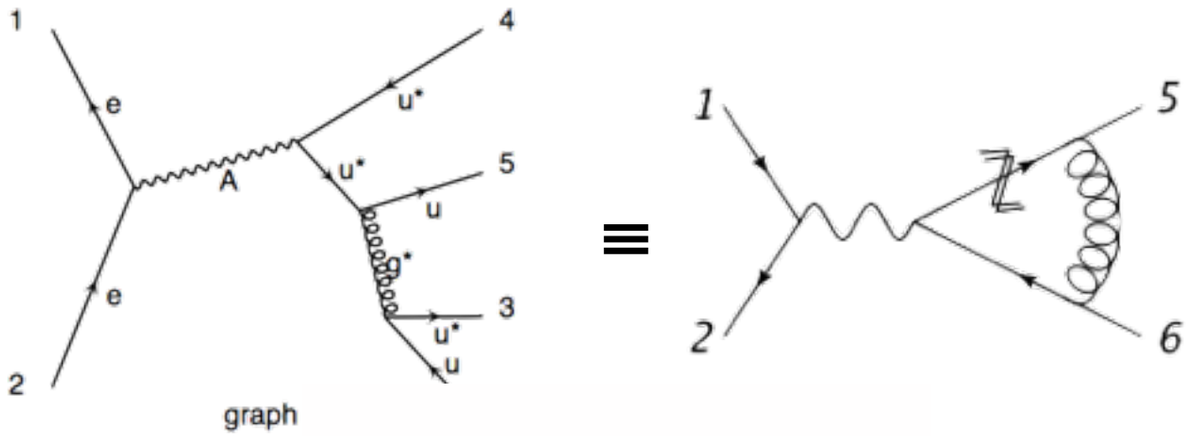
≡



Generating loop diagrams

- No external tool for loop diagram generation:
  Reuse MG5_aMC efficient tree level diagram generation!
- Cut loops have two extra external particles

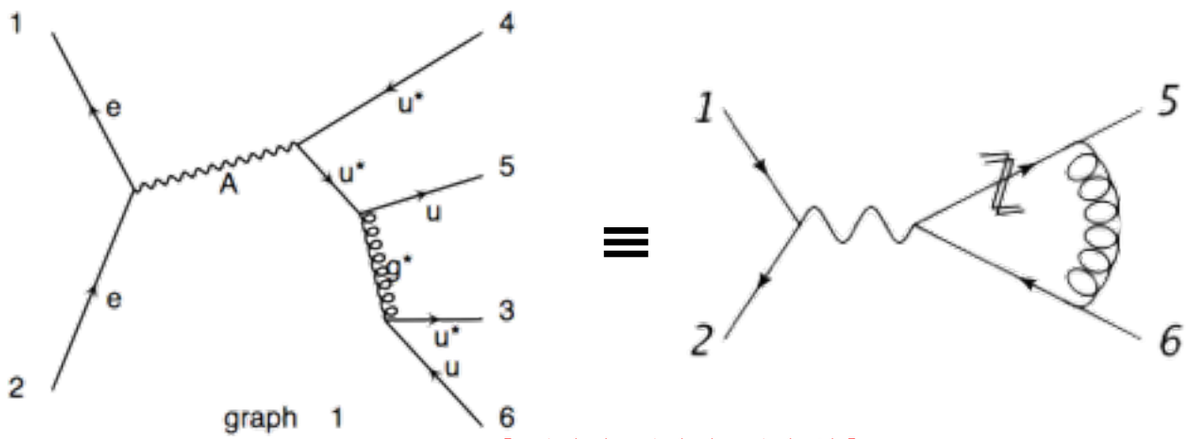
Trees ($e^+e^- \to u\,u\sim\,u\,u\sim$) ≡ Loops ($e^+e^- \to u\,u\sim$)



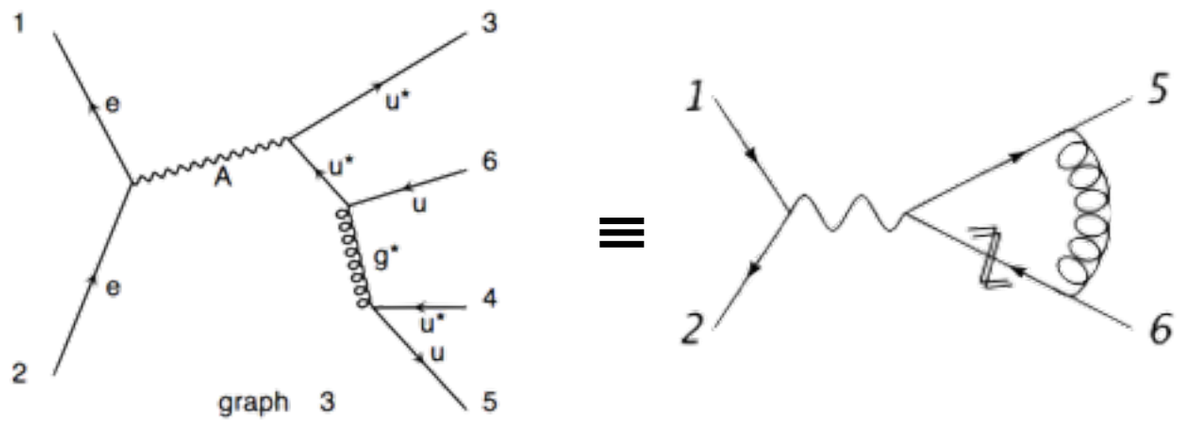
Generating loop diagrams

- No external tool for loop diagram generation:
  Reuse MG5_aMC efficient tree level diagram generation!
- Cut loops have two extra external particles

Trees ($e^+e^- \to u\,u\sim\,u\,u\sim$) ≡ Loops ($e^+e^- \to u\,u\sim$)



$$\text{Diag}_1 = [u^*(6)g^*(5)u^*(A)]$$



$$\text{Diag}_3 = [u^*(A)u^*(6)g^*(5)]$$

Open-loops

[S. Pozzorini & al. hep-ph/1111.5206]

- Lite-Motive: Be Numerical where you can and analytical where you should.

$$\mathcal{N}(l^\mu) = \sum_{r=0}^{r_{max}} C^{(r)}_{\mu_0\mu_1\cdots\mu_r} l^{\mu_0} l^{\mu_1} \cdots l^{\mu_r}$$

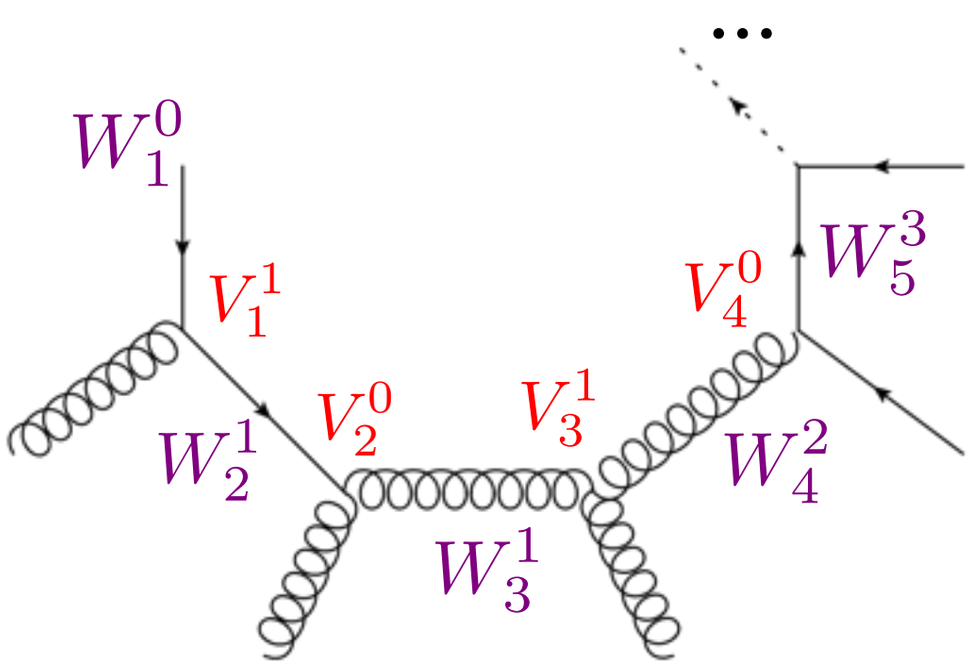
Open-loops

[S. Pozzorini & al. hep-ph/1111.5206]

- Lite-Motive: Be Numerical where you can and analytical where you should.

$$\mathcal{N}(l^\mu) = \sum_{r=0}^{r_{max}} C^{(r)}_{\mu_0\mu_1\cdots\mu_r} l^{\mu_0} l^{\mu_1} \cdots l^{\mu_r}$$

- How to get these coefficients? (Wavefunction and 4-momenta indices now omitted)



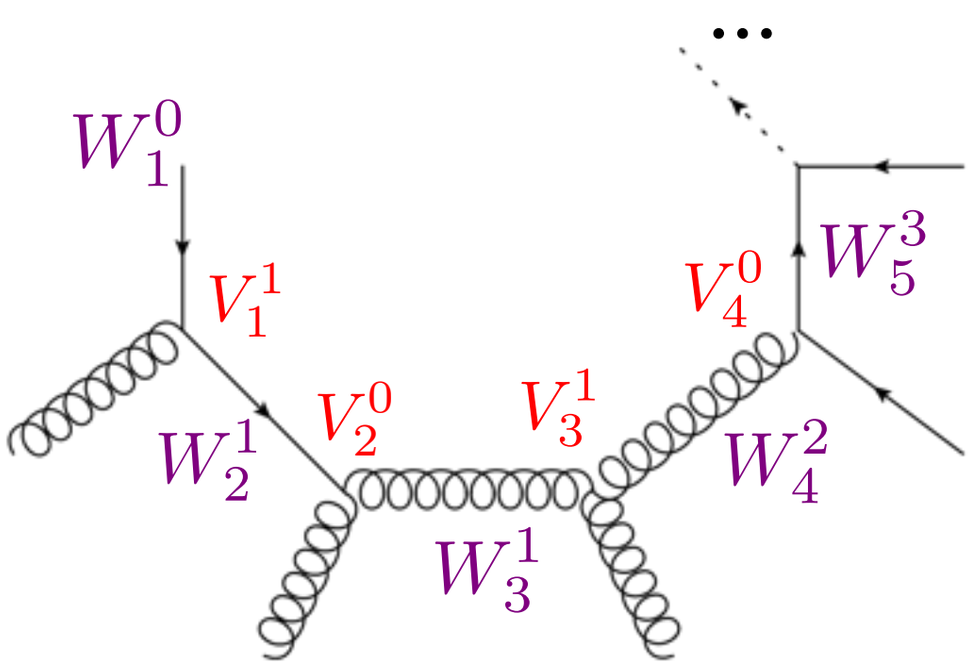
Open-loops

[S. Pozzorini & al. hep-ph/1111.5206]

- Lite-Motive: Be Numerical where you can and analytical where you should.

$$\mathcal{N}(l^\mu) = \sum_{r=0}^{r_{max}} C^{(r)}_{\mu_0\mu_1\cdots\mu_r} l^{\mu_0} l^{\mu_1} \cdots l^{\mu_r}$$

- How to get these coefficients? (Wavefunction and 4-momenta indices now omitted)



$$W_j^{(r)} = \sum_{i=0}^{r} w_j^i l^i \qquad V_j^{(r=0,1)} = \sum_{i=0}^{r} v_j^i l^i$$

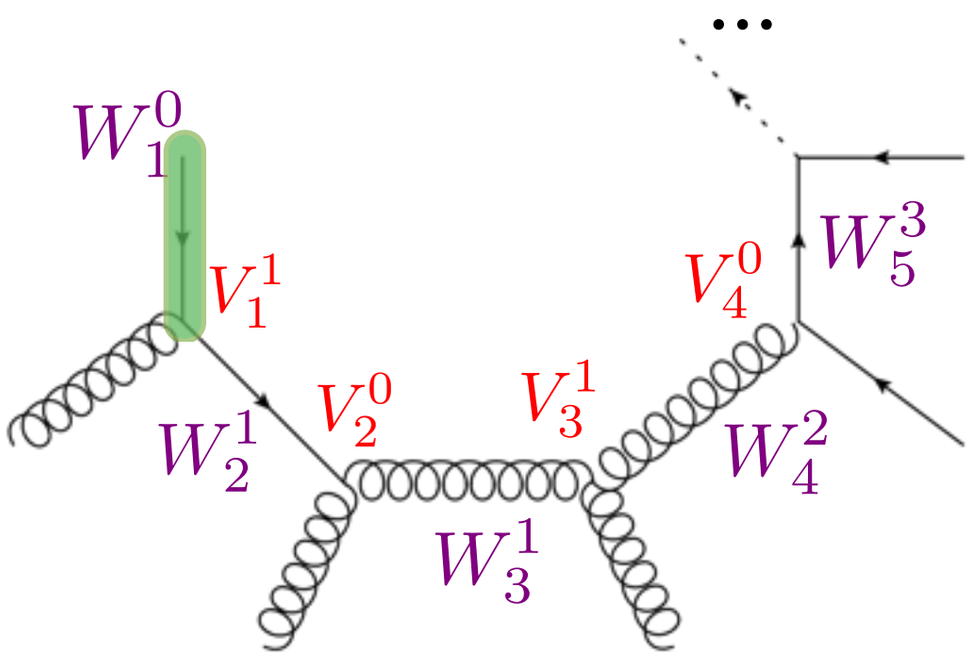
Open-loops

[S. Pozzorini & al. hep-ph/1111.5206]

- Lite-Motive: Be Numerical where you can and analytical where you should.

$$\mathcal{N}(l^\mu) = \sum_{r=0}^{r_{max}} C^{(r)}_{\mu_0\mu_1\cdots\mu_r} l^{\mu_0} l^{\mu_1} \cdots l^{\mu_r}$$

- How to get these coefficients? (Wavefunction and 4-momenta indices now omitted)



$$W_j^{(r)} = \sum_{i=0}^{r} w_j^i l^i \qquad V_j^{(r=0,1)} = \sum_{i=0}^{r} v_j^i l^i$$

$$W_1^{(0)} = w_1^0 = 1$$

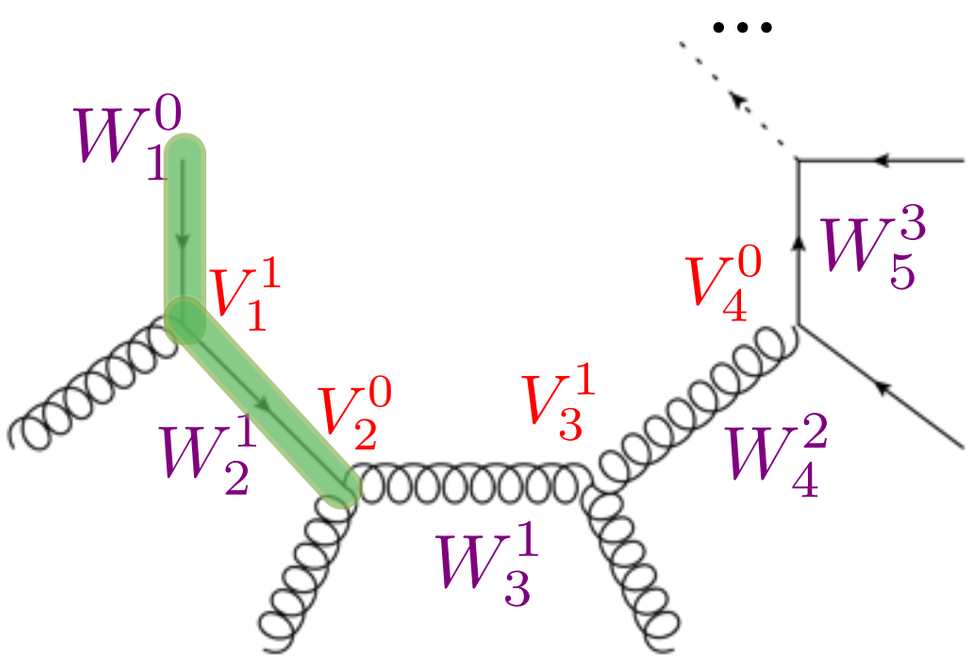
Open-loops

[S. Pozzorini & al. hep-ph/1111.5206]

- Lite-Motive: Be Numerical where you can and analytical where you should.

$$\mathcal{N}(l^\mu) = \sum_{r=0}^{r_{max}} C^{(r)}_{\mu_0\mu_1\cdots\mu_r} l^{\mu_0} l^{\mu_1} \cdots l^{\mu_r}$$

- How to get these coefficients? (Wavefunction and 4-momenta indices now omitted)



$$W_j^{(r)} = \sum_{i=0}^{r} w_j^i l^i \qquad V_j^{(r=0,1)} = \sum_{i=0}^{r} v_j^i l^i$$

$$W_1^{(0)} = w_1^0 = 1$$

$$W_2^{(1)} = (v_1^1 l + v_1^0) w_1^0$$

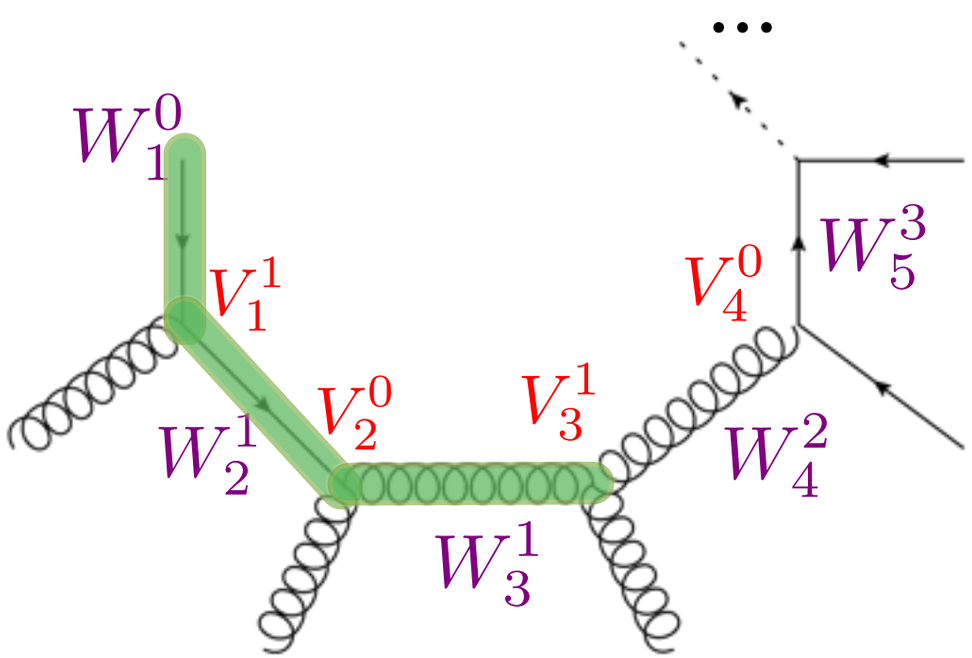
Open-loops

[S. Pozzorini & al. hep-ph/1111.5206]

- Lite-Motive: Be Numerical where you can and analytical where you should.

$$\mathcal{N}(l^\mu) = \sum_{r=0}^{r_{max}} C^{(r)}_{\mu_0\mu_1\cdots\mu_r} l^{\mu_0} l^{\mu_1} \cdots l^{\mu_r}$$

- How to get these coefficients? (Wavefunction and 4-momenta indices now omitted)



$$W_j^{(r)} = \sum_{i=0}^{r} w_j^i l^i \qquad V_j^{(r=0,1)} = \sum_{i=0}^{r} v_j^i l^i$$

$$W_1^{(0)} = w_1^0 = 1$$

$$W_2^{(1)} = (v_1^1 l + v_1^0) w_1^0$$

$$W_3^{(1)} = v_2^0 W_2^{(1)} = v_2^0 (v_1^1 l + v_1^0) w_1^0$$

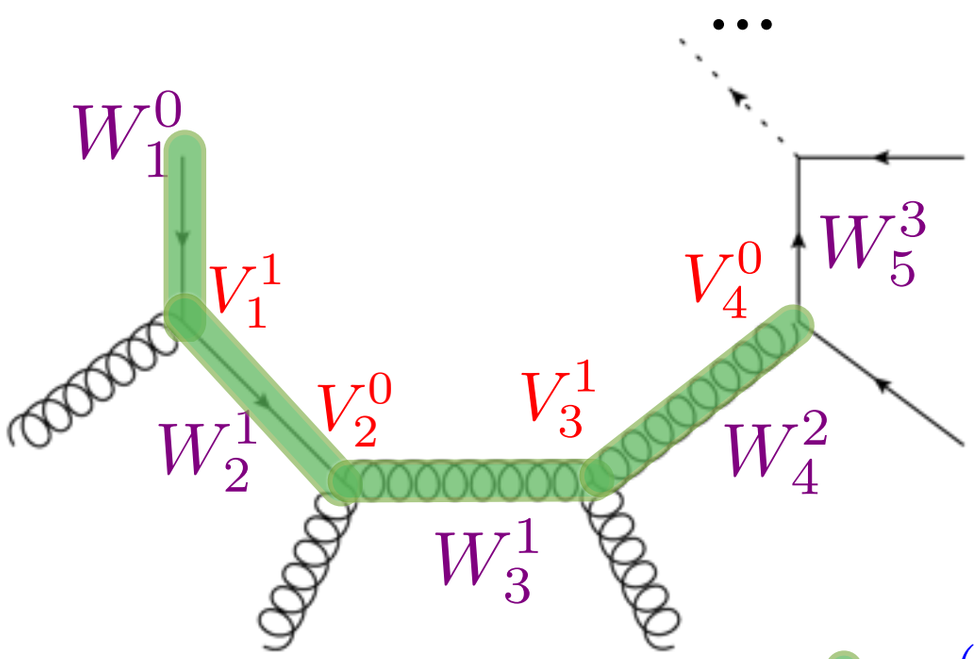
Open-loops

[S. Pozzorini & al. hep-ph/1111.5206]

- Lite-Motive: Be Numerical where you can and analytical where you should.

$$\mathcal{N}(l^\mu) = \sum_{r=0}^{r_{max}} C^{(r)}_{\mu_0\mu_1\cdots\mu_r} l^{\mu_0} l^{\mu_1} \cdots l^{\mu_r}$$

- How to get these coefficients? (Wavefunction and 4-momenta indices now omitted)



$$W_j^{(r)} = \sum_{i=0}^{r} w_j^i l^i \qquad V_j^{(r=0,1)} = \sum_{i=0}^{r} v_j^i l^i$$

$$W_1^{(0)} = w_1^1 = 1$$

$$W_2^{(1)} = (v_1^1 l + v_1^0) w_1^0$$

$$W_3^{(1)} = v_2^0 W_2^{(1)} = v_2^0 (v_1^1 l + v_1^0) w_1^0$$

$$W_4^{(1)} = V_3^{(1)} W_2^{(1)} = (v_3^1 l + v_3^0) v_2^0 (v_1^1 l + v_1^0) w_1^0$$

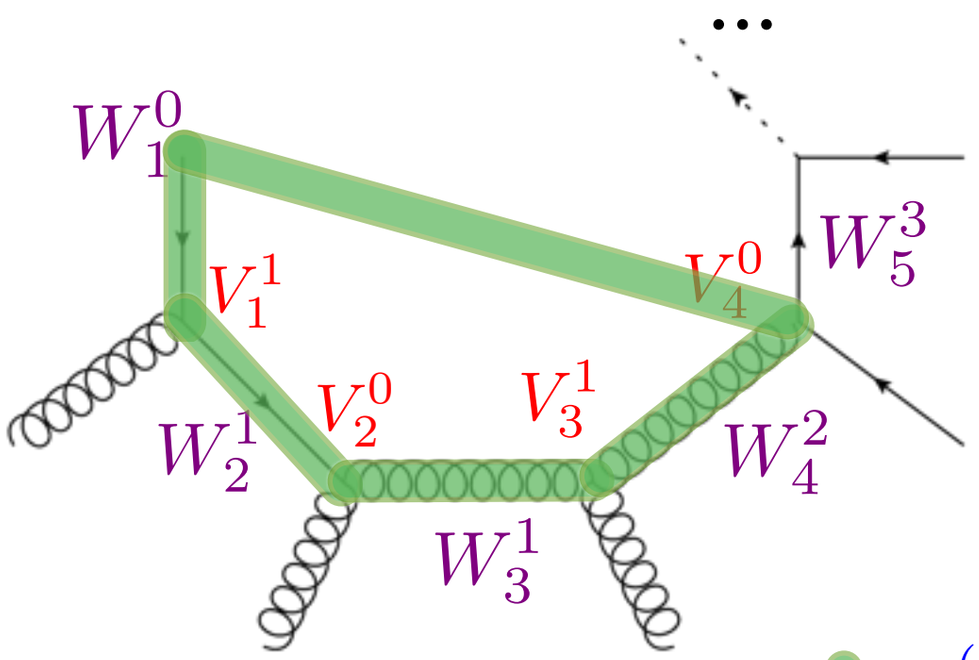
Open-loops

[S. Pozzorini & al. hep-ph/1111.5206]

- Lite-Motive: Be Numerical where you can and analytical where you should.

$$\mathcal{N}(l^\mu) = \sum_{r=0}^{r_{max}} C^{(r)}_{\mu_0\mu_1\cdots\mu_r} l^{\mu_0} l^{\mu_1} \cdots l^{\mu_r}$$

- How to get these coefficients? (Wavefunction and 4-momenta indices now omitted)



$$W_j^{(r)} = \sum_{i=0}^{r} w_j^i l^i \qquad V_j^{(r=0,1)} = \sum_{i=0}^{r} v_j^i l^i$$

$$W_1^{(0)} = w_1^0 = 1$$

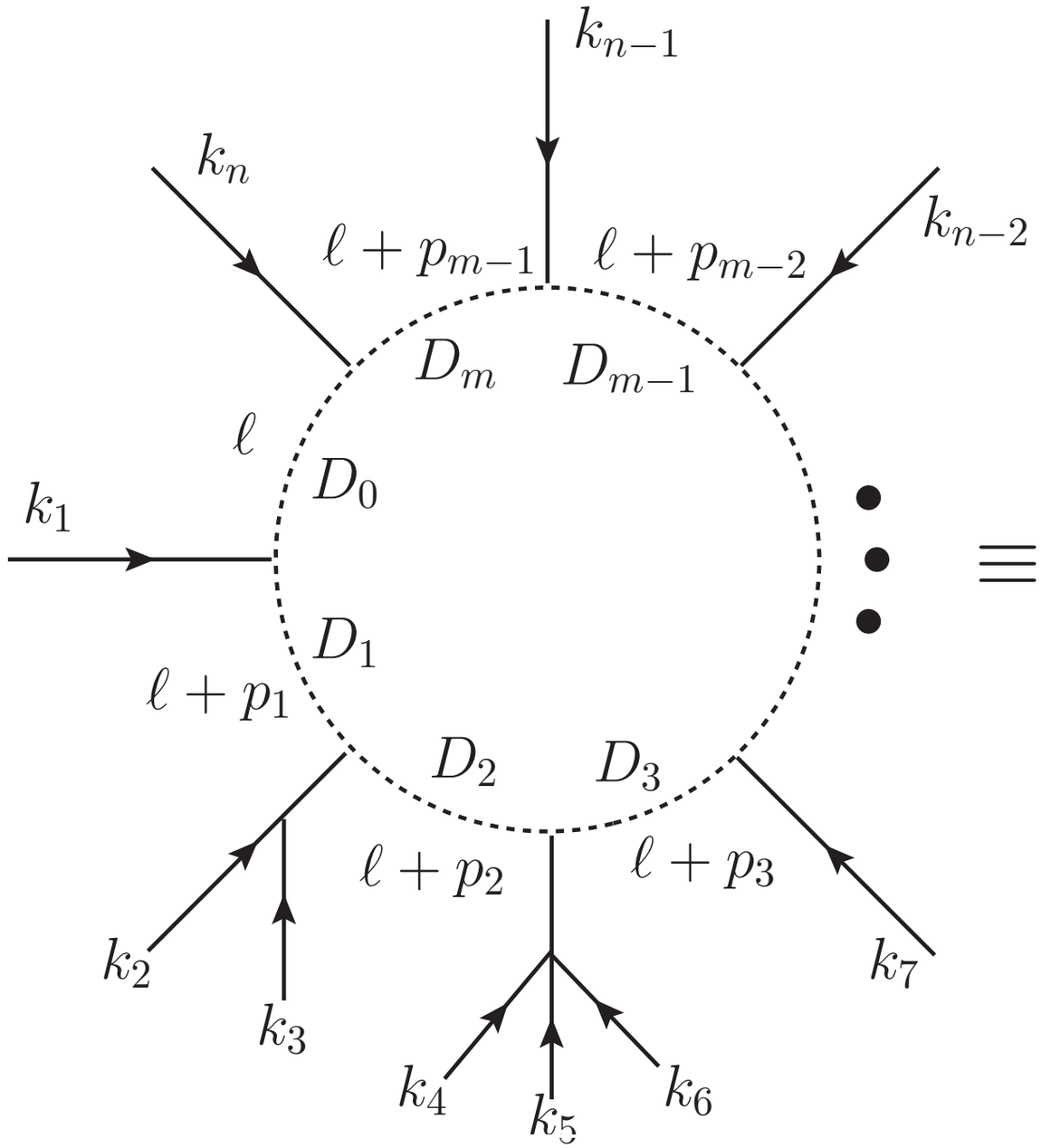
$$W_2^{(1)} = (v_1^1 l + v_1^0) w_1^0$$

$$W_3^{(1)} = v_2^0 W_2^{(1)} = v_2^0 (v_1^1 l + v_1^0) w_1^0$$

$$W_4^{(1)} = V_3^{(1)} W_2^{(1)} = (v_3^1 l + v_3^0) v_2^0 (v_1^1 l + v_1^0) w_1^0$$

... or end of loop and $C^{(2)} = v_3^1 v_2^0 v_1^1 w_1^0, C^{(1)} = v_2^0 w_1^0 (v_3^1 v_0^1 + v_3^0 v_1^1), C^0 = \cdots$

ONE-LOOP INTEGRAL



$$\equiv \int \frac{d^d\ell}{(2\pi)^d} \frac{\mathcal{N}(\ell)}{D_0 D_1 D_2 D_3 \cdots D_{m-2} D_{m-1}}$$

- Consider this *m*-point loop diagram with *n* external momenta

with $D_i = (\ell + p_i)^2 - m_i^2$

We will denote by $\mathcal{C}$ this integral.

Scalar integral Basis

$$\mathcal{C}^{\text{1-loop}} = \sum_{i_0<i_1<i_2<i_3} d_{i_0 i_1 i_2 i_3}\,\text{Box}_{i_0 i_1 i_2 i_3} + \sum_{i_0<i_1<i_2} c_{i_0 i_1 i_2}\,\text{Triangle}_{i_0 i_1 i_2} + \sum_{i_0<i_1} b_{i_0 i_1}\,\text{Bubble}_{i_0 i_1} + \sum_{i_0} a_{i_0}\,\text{Tadpole}_{i_0} + R + \mathcal{O}(\epsilon)$$

$$\text{Box}_{i_0 i_1 i_2 i_3} = \int d^d l \frac{1}{D_{i_0} D_{i_1} D_{i_2} D_{i_3}}$$

$$\text{Triangle}_{i_0 i_1 i_2} = \int d^d l \frac{1}{D_{i_0} D_{i_1} D_{i_2}}$$

$$\text{Bubble}_{i_0 i_1} = \int d^d l \frac{1}{D_{i_0} D_{i_1}}$$

$$\text{Tadpole}_{i_0} = \int d^d l \frac{1}{D_{i_0}}$$

The a, b, c, d and R coefficients depend only on external parameters and momenta.

Reduction of the loop to these scalar coefficients can be achieved using either Tensor Integral Reduction or Reduction at the integrand level

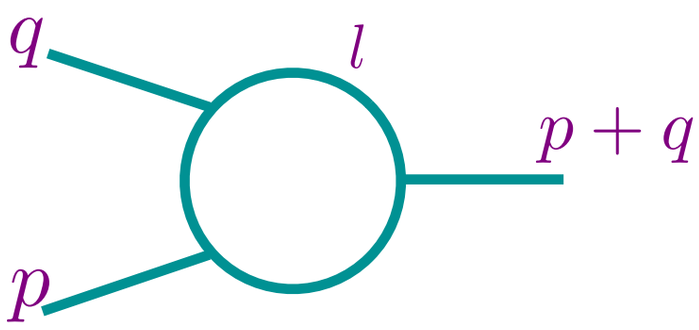
TIR: Passarino-Veltman

- Passarino-Veltman reduction:

$$\int d^d l \, \frac{N(l)}{D_0 D_1 D_2 \cdots D_{m-1}} \to \sum_i \text{coeff}_i \int d^d l \, \frac{1}{D_0 D_1 \cdots}$$

  - Reduce a general integral to "scalar integrals" by "completing the square"

- Example:
  Application of PV to this triangle rank-1 integral



$$\int \frac{d^n l}{(2\pi)^n} \frac{l^\mu}{(l^2 - m_1^2)((l+p)^2 - m_2^2)((l+q)^2 - m_3^2)}$$

- Implemented in codes such as:

COLLIER [A. Denner, S .Dittmaier, L. Hofer, 1604.06792]
GOLEM95 [T. Binoth, J.Guillet, G. Heinrich, E.Pilon, T.Reither, 0810.0992]

$$\int \frac{d^n l}{(2\pi)^n} \frac{l^\mu}{(l^2 - m_1^2)((l+p)^2 - m_2^2)((l+q)^2 - m_3^2)}$$

$$\int \frac{d^n l}{(2\pi)^n} \frac{l^\mu}{(l^2 - m_1^2)((l+p)^2 - m_2^2)((l+q)^2 - m_3^2)}$$

- The only independent four vectors are p^μ and q^μ . Therefore, the integral must be proportional to those. We can set-up a system of linear equations and try to solve for C_1 and C_2

$$\int \frac{d^n l}{(2\pi)^n} \frac{l^\mu}{(l^2 - m_1^2)((l+p)^2 - m_2^2)((l+q)^2 - m_3^2)} = \begin{pmatrix} p^\mu & q^\mu \end{pmatrix} \begin{pmatrix} C_1 \\ C_2 \end{pmatrix}$$

$$\int \frac{d^n l}{(2\pi)^n} \frac{l^\mu}{(l^2 - m_1^2)((l+p)^2 - m_2^2)((l+q)^2 - m_3^2)}$$

- The only independent four vectors are p^μ and q^μ . Therefore, the integral must be proportional to those. We can set-up a system of linear equations and try to solve for C_1 and C_2

$$\int \frac{d^n l}{(2\pi)^n} \frac{l^\mu}{(l^2 - m_1^2)((l+p)^2 - m_2^2)((l+q)^2 - m_3^2)} = \begin{pmatrix} p^\mu & q^\mu \end{pmatrix} \begin{pmatrix} C_1 \\ C_2 \end{pmatrix}$$

- We can solve for C_1 and C_2 by contracting with p and q

$$\begin{pmatrix} R_1 \\ R_2 \end{pmatrix} = \begin{pmatrix} [2l \cdot p] \\ [2l \cdot q] \end{pmatrix} = G \begin{pmatrix} C_1 \\ C_2 \end{pmatrix} \equiv \begin{pmatrix} 2p \cdot p & 2p \cdot q \\ 2p \cdot q & 2q \cdot q \end{pmatrix} \begin{pmatrix} C_1 \\ C_2 \end{pmatrix}$$

where $[2l \cdot p] = \int \frac{d^n l}{(2\pi)^n} \frac{2l \cdot p}{l^2 (l+p)^2 (l+q)^2}$ (For simplicity, the masses are neglected here)

$$\int \frac{d^n l}{(2\pi)^n} \frac{l^\mu}{(l^2 - m_1^2)((l+p)^2 - m_2^2)((l+q)^2 - m_3^2)}$$

- The only independent four vectors are p^μ and q^μ . Therefore, the integral must be proportional to those. We can set-up a system of linear equations and try to solve for C_1 and C_2

$$\int \frac{d^n l}{(2\pi)^n} \frac{l^\mu}{(l^2 - m_1^2)((l+p)^2 - m_2^2)((l+q)^2 - m_3^2)} = \begin{pmatrix} p^\mu & q^\mu \end{pmatrix} \begin{pmatrix} C_1 \\ C_2 \end{pmatrix}$$

- We can solve for C_1 and C_2 by contracting with p and q

$$\begin{pmatrix} R_1 \\ R_2 \end{pmatrix} = \begin{pmatrix} [2l \cdot p] \\ [2l \cdot q] \end{pmatrix} = G \begin{pmatrix} C_1 \\ C_2 \end{pmatrix} \equiv \begin{pmatrix} 2p \cdot p & 2p \cdot q \\ 2p \cdot q & 2q \cdot q \end{pmatrix} \begin{pmatrix} C_1 \\ C_2 \end{pmatrix}$$

where $[2l \cdot p] = \int \frac{d^n l}{(2\pi)^n} \frac{2l \cdot p}{l^2 (l+p)^2 (l+q)^2}$ (For simplicity, the masses are neglected here)

- By expressing $2l.p$ and $2l.q$ as a sum of denominators we can express R_1 and R_2 as a sum of simpler integrals, *e.g.*

$$R_1 = \int \frac{d^n l}{(2\pi)^n} \frac{2l \cdot p}{l^2 (l+p)^2 (l+q)^2} = \int \frac{d^n l}{(2\pi)^n} \frac{(l+p)^2 - l^2 - p^2}{l^2 (l+p)^2 (l+q)^2}$$

$$= \int \frac{d^n l}{(2\pi)^n} \frac{1}{l^2 (l+q)^2} - \int \frac{d^n l}{(2\pi)^n} \frac{1}{(l+p)^2 (l+q)^2} - p^2 \int \frac{d^n l}{(2\pi)^n} \frac{1}{l^2 (l+p)^2 (l+q)^2}$$

- And similarly for R_2

$$R_2 = \int \frac{d^n l}{(2\pi)^n} \frac{2l \cdot q}{l^2 (l+p)^2 (l+q)^2} = \int \frac{d^n l}{(2\pi)^n} \frac{(l+q)^2 - l^2 - q^2}{l^2 (l+p)^2 (l+q)^2}$$

$$= \int \frac{d^n l}{(2\pi)^n} \frac{1}{l^2 (l+p)^2} - \int \frac{d^n l}{(2\pi)^n} \frac{1}{(l+p)^2 (l+q)^2} - q^2 \int \frac{d^n l}{(2\pi)^n} \frac{1}{l^2 (l+p)^2 (l+q)^2}$$

- And similarly for R_2

$$R_2 = \int \frac{d^n l}{(2\pi)^n} \frac{2l \cdot q}{l^2 (l+p)^2 (l+q)^2} = \int \frac{d^n l}{(2\pi)^n} \frac{(l+q)^2 - l^2 - q^2}{l^2 (l+p)^2 (l+q)^2}$$
$$= \int \frac{d^n l}{(2\pi)^n} \frac{1}{l^2 (l+p)^2} - \int \frac{d^n l}{(2\pi)^n} \frac{1}{(l+p)^2 (l+q)^2} - q^2 \int \frac{d^n l}{(2\pi)^n} \frac{1}{l^2 (l+p)^2 (l+q)^2}$$

- Now we can solve the equation

$$\begin{pmatrix} R_1 \\ R_2 \end{pmatrix} = \begin{pmatrix} [2l \cdot p] \\ [2l \cdot q] \end{pmatrix} = G \begin{pmatrix} C_1 \\ C_2 \end{pmatrix} \equiv \begin{pmatrix} 2p \cdot p & 2p \cdot q \\ 2p \cdot q & 2q \cdot q \end{pmatrix} \begin{pmatrix} C_1 \\ C_2 \end{pmatrix}$$

by inverting the "Gram" matrix G

$$\begin{pmatrix} C_1 \\ C_2 \end{pmatrix} = G^{-1} \begin{pmatrix} R_1 \\ R_2 \end{pmatrix}$$

- And similarly for R_2

$$R_2 = \int \frac{d^n l}{(2\pi)^n} \frac{2l \cdot q}{l^2 (l+p)^2 (l+q)^2} = \int \frac{d^n l}{(2\pi)^n} \frac{(l+q)^2 - l^2 - q^2}{l^2 (l+p)^2 (l+q)^2}$$
$$= \int \frac{d^n l}{(2\pi)^n} \frac{1}{l^2 (l+p)^2} - \int \frac{d^n l}{(2\pi)^n} \frac{1}{(l+p)^2 (l+q)^2} - q^2 \int \frac{d^n l}{(2\pi)^n} \frac{1}{l^2 (l+p)^2 (l+q)^2}$$

- Now we can solve the equation

$$\begin{pmatrix} R_1 \\ R_2 \end{pmatrix} = \begin{pmatrix} [2l \cdot p] \\ [2l \cdot q] \end{pmatrix} = G \begin{pmatrix} C_1 \\ C_2 \end{pmatrix} \equiv \begin{pmatrix} 2p \cdot p & 2p \cdot q \\ 2p \cdot q & 2q \cdot q \end{pmatrix} \begin{pmatrix} C_1 \\ C_2 \end{pmatrix}$$

by inverting the "Gram" matrix G

$$\begin{pmatrix} C_1 \\ C_2 \end{pmatrix} = G^{-1} \begin{pmatrix} R_1 \\ R_2 \end{pmatrix}$$

- We have re-expressed, reduced, our original integral

$$\int \frac{d^n l}{(2\pi)^n} \frac{l^\mu}{(l^2 - m_1^2)((l+p)^2 - m_2^2)((l+q)^2 - m_3^2)} = \begin{pmatrix} p^\mu & q^\mu \end{pmatrix} \begin{pmatrix} C_1 \\ C_2 \end{pmatrix}$$

in terms of known, simpler *scalar* integrals

PV-Reduction chain

D_{ijkl}	$\rightarrow$	$D_{00ij}, D_{ijk}, C_{ijk}, C_{ij}, C_i, C_0$
D_{00ii}	$\rightarrow$	$D_{ijk}, D_{ij}, C_{ij}, C_i$
D_{0000}	$\rightarrow$	D_{00i}, D_{00}, C_{00}
D_{ijk}	$\rightarrow$	$D_{00i}, D_{ij}, C_{ij}, C_i$
D_{00i}	$\rightarrow$	D_{ij}, D_i, C_i, C_0
D_{ij}	$\rightarrow$	D_{00}, D_i, C_i, C_0
D_{00}	$\rightarrow$	D_i, D_0, C_0
D_i	$\rightarrow$	D_0, C_0
C_{ijk}	$\rightarrow$	$C_{00i}, C_{ij}, B_{ij}, B_i$
C_{00i}	$\rightarrow$	C_{ii}, C_i, B_i, B_0
C_{ij}	$\rightarrow$	C_{00}, C_i, B_i, B_0
C_{00}	$\rightarrow$	C_i, C_0, B_0
C_i	$\rightarrow$	C_0, B_0
B_{ii}	$\rightarrow$	B_{00}, B_i, A_0
B_{00}	$\rightarrow$	B_i, B_0, A_0
B_i	$\rightarrow$	B_0, A_0

Table from K.Ellis & al. hep-ph/1105.4319

Integrand reduction

- The integrand (or OPP [Ossola, Papadopoulos, Pittau 2006]) reduction method is a purely numerical algorithm that has been automated in computer codes such as

  CutTools [G.Ossola, C.Papadopoulos, R.Pittau, 0711.3596]
  NINJA [T. Peraro, 1403.1229] (interface to MadLoop in [VH, T. Peraro, 1604.01363]
  SAMURAI [P. Mastrolia, G. Ossola, T. Reiter, F. Tramontano 1006.0710]

  to find the scalar loop coefficients

- Both OPP and Tensor Integral Reduction techniques are interfaced in our MadLoop to compute loop diagrams.

How does OPP work?

INTEGRAND LEVEL

- The decomposition to scalar integrals presented before works at the level of the integrals

$$\mathcal{M}^{\text{1-loop}} = \sum_{i_0<i_1<i_2<i_3} d_{i_0 i_1 i_2 i_3} \text{Box}_{i_0 i_1 i_2 i_3}$$
$$+ \sum_{i_0<i_1<i_2} c_{i_0 i_1 i_2} \text{Triangle}_{i_0 i_1 i_2}$$
$$+ \sum_{i_0<i_1} b_{i_0 i_1} \text{Bubble}_{i_0 i_1}$$
$$+ \sum_{i_0} a_{i_0} \text{Tadpole}_{i_0}$$
$$+ R + \mathcal{O}(\epsilon)$$

If we would know a similar relation at the **integrand** level, we would be able to manipulate the integrands and extract the coefficients without doing the integrals

$$N(l) = \sum_{i_0<i_1<i_2<i_3}^{m-1} \left[d_{i_0 i_1 i_2 i_3} + \tilde{d}_{i_0 i_1 i_2 i_3}(l) \right] \prod_{i \neq i_0, i_1, i_2, i_3}^{m-1} D_i$$
$$+ \sum_{i_0<i_1<i_2}^{m-1} \left[c_{i_0 i_1 i_2} + \tilde{c}_{i_0 i_1 i_2}(l) \right] \prod_{i \neq i_0, i_1, i_2}^{m-1} D_i$$
$$+ \sum_{i_0<i_1}^{m-1} \left[b_{i_0 i_1} + \tilde{b}_{i_0 i_1}(l) \right] \prod_{i \neq i_0, i_1}^{m-1} D_i$$
$$+ \sum_{i_0}^{m-1} \left[a_{i_0} + \tilde{a}_{i_0}(l) \right] \prod_{i \neq i_0}^{m-1} D_i$$
$$+ \tilde{P}(l) \prod_{i}^{m-1} D_i$$

Spurious term

INTEGRAND LEVEL

- The functional form of the spurious terms is known (it depends on the rank of the integral and the number of propagators in the loop) [del Aguila, Pittau 2004]
  - for example, a box coefficient from a rank 1 numerator is

$$\tilde{d}_{i_0 i_1 i_2 i_3}(l) = \tilde{d}_{i_0 i_1 i_2 i_3}\, \epsilon^{\mu\nu\rho\sigma}\, l^\mu p_1^\nu p_2^\rho p_3^\sigma$$

(remember that p_i is the sum of the momentum that has entered the loop so far, so we always have $p_0 = 0$)

- The integral is zero

$$\int d^d l \frac{\tilde{d}_{i_0 i_1 i_2 i_3}(l)}{D_0 D_1 D_2 D_3} = \tilde{d}_{i_0 i_1 i_2 i_3} \int d^d l \frac{\epsilon^{\mu\nu\rho\sigma}\, l^\mu p_1^\nu p_2^\rho p_3^\sigma}{D_0 D_1 D_2 D_3} = 0$$

OPP Reduction

$$
\begin{aligned}
N(l) = & \sum_{i_0<i_1<i_2<i_3}^{m-1} \left[d_{i_0 i_1 i_2 i_3} + \tilde{d}_{i_0 i_1 i_2 i_3}(l) \right] \prod_{i \neq i_0,i_1,i_2,i_3}^{m-1} D_i \\
& + \sum_{i_0<i_1<i_2}^{m-1} \left[c_{i_0 i_1 i_2} + \tilde{c}_{i_0 i_1 i_2}(l) \right] \prod_{i \neq i_0,i_1,i_2}^{m-1} D_i \\
& + \sum_{i_0<i_1}^{m-1} \left[b_{i_0 i_1} + \tilde{b}_{i_0 i_1}(l) \right] \prod_{i \neq i_0,i_1}^{m-1} D_i \\
& + \sum_{i_0}^{m-1} \left[a_{i_0} + \tilde{a}_{i_0}(l) \right] \prod_{i \neq i_0}^{m-1} D_i \\
& + \tilde{P}(l) \prod_{i}^{m-1} D_i
\end{aligned}
$$

OPP Reduction

$$
\begin{aligned}
N(l) = & \sum_{i_0<i_1<i_2<i_3}^{m-1} \left[d_{i_0 i_1 i_2 i_3} + \tilde{d}_{i_0 i_1 i_2 i_3}(l) \right] \prod_{i \neq i_0,i_1,i_2,i_3}^{m-1} D_i \\
& + \sum_{i_0<i_1<i_2}^{m-1} \left[c_{i_0 i_1 i_2} + \tilde{c}_{i_0 i_1 i_2}(l) \right] \prod_{i \neq i_0,i_1,i_2}^{m-1} D_i \\
& + \sum_{i_0<i_1}^{m-1} \left[b_{i_0 i_1} + \tilde{b}_{i_0 i_1}(l) \right] \prod_{i \neq i_0,i_1}^{m-1} D_i \\
& + \sum_{i_0}^{m-1} \left[a_{i_0} + \tilde{a}_{i_0}(l) \right] \prod_{i \neq i_0}^{m-1} D_i \\
& + \tilde{P}(l) \prod_{i}^{m-1} D_i
\end{aligned}
$$

To solve the OPP reduction, choosing special values for the loop momentum helps a lot

OPP Reduction

$$N(l) = \sum_{i_0<i_1<i_2<i_3}^{m-1} \left[d_{i_0 i_1 i_2 i_3} + \tilde{d}_{i_0 i_1 i_2 i_3}(l)\right] \prod_{i\neq i_0,i_1,i_2,i_3}^{m-1} D_i$$
$$+ \sum_{i_0<i_1<i_2}^{m-1} \left[c_{i_0 i_1 i_2} + \tilde{c}_{i_0 i_1 i_2}(l)\right] \prod_{i\neq i_0,i_1,i_2}^{m-1} D_i$$
$$+ \sum_{i_0<i_1}^{m-1} \left[b_{i_0 i_1} + \tilde{b}_{i_0 i_1}(l)\right] \prod_{i\neq i_0,i_1}^{m-1} D_i$$
$$+ \sum_{i_0}^{m-1} \left[a_{i_0} + \tilde{a}_{i_0}(l)\right] \prod_{i\neq i_0}^{m-1} D_i$$
$$+ \tilde{P}(l) \prod_{i}^{m-1} D_i$$

To solve the OPP reduction, choosing special values for the loop momentum helps a lot

For example, choosing *l* such that

$$D_0(l^{\pm}) = D_1(l^{\pm}) = = D_2(l^{\pm}) = D_3(l^{\pm}) = 0$$

sets all the terms in this equation to zero except the first line

OPP Reduction

$$N(l) = \sum_{i_0<i_1<i_2<i_3}^{m-1} \left[d_{i_0 i_1 i_2 i_3} + \tilde{d}_{i_0 i_1 i_2 i_3}(l)\right] \prod_{i\neq i_0,i_1,i_2,i_3}^{m-1} D_i$$

$$+ \sum_{i_0<i_1<i_2}^{m-1} \left[c_{i_0 i_1 i_2} + \tilde{c}_{i_0 i_1 i_2}(l)\right] \prod_{i\neq i_0,i_1,i_2}^{m-1} D_i$$

$$+ \sum_{i_0<i_1}^{m-1} \left[b_{i_0 i_1} + \tilde{b}_{i_0 i_1}(l)\right] \prod_{i\neq i_0,i_1}^{m-1} D_i$$

$$+ \sum_{i_0}^{m-1} \left[a_{i_0} + \tilde{a}_{i_0}(l)\right] \prod_{i\neq i_0}^{m-1} D_i$$

$$+ \tilde{P}(l) \prod_{i}^{m-1} D_i$$

$$= 0$$

To solve the OPP reduction, choosing special values for the loop momentum helps a lot

For example, choosing *l* such that

$$D_0(l^{\pm}) = D_1(l^{\pm}) = D_2(l^{\pm}) = D_3(l^{\pm}) = 0$$

sets all the terms in this equation to zero except the first line

OPP Reduction

$$N(l) = \sum_{i_0<i_1<i_2<i_3}^{m-1} \left[d_{i_0 i_1 i_2 i_3} + \tilde{d}_{i_0 i_1 i_2 i_3}(l)\right] \prod_{i\neq i_0,i_1,i_2,i_3}^{m-1} D_i$$

$$+ \sum_{i_0<i_1<i_2}^{m-1} \left[c_{i_0 i_1 i_2} + \tilde{c}_{i_0 i_1 i_2}(l)\right] \prod_{i\neq i_0,i_1,i_2}^{m-1} D_i$$

$$+ \sum_{i_0<i_1}^{m-1} \left[b_{i_0 i_1} + \tilde{b}_{i_0 i_1}(l)\right] \prod_{i\neq i_0,i_1}^{m-1} D_i$$

$$+ \sum_{i_0}^{m-1} \left[a_{i_0} + \tilde{a}_{i_0}(l)\right] \prod_{i\neq i_0}^{m-1} D_i$$

$$+ \tilde{P}(l) \prod_{i}^{m-1} D_i$$

$$= 0$$

To solve the OPP reduction, choosing special values for the loop momentum helps a lot

For example, choosing *l* such that

$$D_0(l^{\pm}) = D_1(l^{\pm}) = D_2(l^{\pm}) = D_3(l^{\pm}) = 0$$

sets all the terms in this equation to zero except the first line

There are two (complex) solutions to this equation due to the quadratic nature of the propagators

OPP Reduction

$$
\begin{aligned}
N(l) = & \sum_{i_0<i_1<i_2<i_3}^{m-1} \left[d_{i_0 i_1 i_2 i_3} + \tilde{d}_{i_0 i_1 i_2 i_3}(l) \right] \prod_{i \neq i_0,i_1,i_2,i_3}^{m-1} D_i \\
& + \sum_{i_0<i_1<i_2}^{m-1} \left[c_{i_0 i_1 i_2} + \tilde{c}_{i_0 i_1 i_2}(l) \right] \prod_{i \neq i_0,i_1,i_2}^{m-1} D_i \\
& + \sum_{i_0<i_1}^{m-1} \left[b_{i_0 i_1} + \tilde{b}_{i_0 i_1}(l) \right] \prod_{i \neq i_0,i_1}^{m-1} D_i \\
& + \sum_{i_0}^{m-1} \left[a_{i_0} + \tilde{a}_{i_0}(l) \right] \prod_{i \neq i_0}^{m-1} D_i \\
& + \tilde{P}(l) \prod_{i}^{m-1} D_i
\end{aligned}
$$

$$D_0(l^i) = D_1(l^i) = D_2(l^i) = 0$$

Coefficient computed in a previous step

OPP Reduction

$$
\begin{aligned}
N(l) = & \sum_{i_0<i_1<i_2<i_3}^{m-1} \left[d_{i_0 i_1 i_2 i_3} + \tilde{d}_{i_0 i_1 i_2 i_3}(l) \right] \prod_{i \neq i_0,i_1,i_2,i_3}^{m-1} D_i \\
& + \sum_{i_0<i_1<i_2}^{m-1} \left[c_{i_0 i_1 i_2} + \tilde{c}_{i_0 i_1 i_2}(l) \right] \prod_{i \neq i_0,i_1,i_2}^{m-1} D_i \\
& + \sum_{i_0<i_1}^{m-1} \left[b_{i_0 i_1} + \tilde{b}_{i_0 i_1}(l) \right] \prod_{i \neq i_0,i_1}^{m-1} D_i \\
& + \sum_{i_0}^{m-1} \left[a_{i_0} + \tilde{a}_{i_0}(l) \right] \prod_{i \neq i_0}^{m-1} D_i \\
& + \tilde{P}(l) \prod_{i}^{m-1} D_i
\end{aligned}
$$

Now we choose l such that

$$D_0(l^i) = D_1(l^i) = D_2(l^i) = 0$$

sets all the terms in this equation to zero except the first and second line

Coefficient computed in a previous step

OPP Reduction

$$N(l) = \sum_{i_0<i_1<i_2<i_3}^{m-1} \left[d_{i_0 i_1 i_2 i_3} + \tilde{d}_{i_0 i_1 i_2 i_3}(l)\right] \prod_{i\neq i_0,i_1,i_2,i_3}^{m-1} D_i$$

$$+ \sum_{i_0<i_1<i_2}^{m-1} \left[c_{i_0 i_1 i_2} + \tilde{c}_{i_0 i_1 i_2}(l)\right] \prod_{i\neq i_0,i_1,i_2}^{m-1} D_i$$

$$+ \sum_{i_0<i_1}^{m-1} \left[b_{i_0 i_1} + \tilde{b}_{i_0 i_1}(l)\right] \prod_{i\neq i_0,i_1}^{m-1} D_i$$

$$+ \sum_{i_0}^{m-1} \left[a_{i_0} + \tilde{a}_{i_0}(l)\right] \prod_{i\neq i_0}^{m-1} D_i$$

$$+ \tilde{P}(l) \prod_{i}^{m-1} D_i$$

$$= 0$$

Now we choose l such that

$$D_0(l^i) = D_1(l^i) = D_2(l^i) = 0$$

sets all the terms in this equation to zero except the first and second line

Coefficient computed in a previous step

OPP Reduction

$$
\begin{aligned}
N(l) = & \sum_{i_0<i_1<i_2<i_3}^{m-1} \left[d_{i_0 i_1 i_2 i_3} + \tilde{d}_{i_0 i_1 i_2 i_3}(l) \right] \prod_{i \neq i_0,i_1,i_2,i_3}^{m-1} D_i \\
& + \sum_{i_0<i_1<i_2}^{m-1} \left[c_{i_0 i_1 i_2} + \tilde{c}_{i_0 i_1 i_2}(l) \right] \prod_{i \neq i_0,i_1,i_2}^{m-1} D_i \\
& + \sum_{i_0<i_1}^{m-1} \left[b_{i_0 i_1} + \tilde{b}_{i_0 i_1}(l) \right] \prod_{i \neq i_0,i_1}^{m-1} D_i \\
& + \sum_{i_0}^{m-1} \left[a_{i_0} + \tilde{a}_{i_0}(l) \right] \prod_{i \neq i_0}^{m-1} D_i \\
& + \tilde{P}(l) \prod_{i}^{m-1} D_i
\end{aligned}
$$

$$D_0(l^i) = D_1(l^i) = 0$$

Coefficient computed in a previous step

OPP Reduction

$$
\begin{aligned}
N(l) = & \sum_{i_0<i_1<i_2<i_3}^{m-1} \left[d_{i_0 i_1 i_2 i_3} + \tilde{d}_{i_0 i_1 i_2 i_3}(l) \right] \prod_{i \neq i_0,i_1,i_2,i_3}^{m-1} D_i \\
& + \sum_{i_0<i_1<i_2}^{m-1} \left[c_{i_0 i_1 i_2} + \tilde{c}_{i_0 i_1 i_2}(l) \right] \prod_{i \neq i_0,i_1,i_2}^{m-1} D_i \\
& + \sum_{i_0<i_1}^{m-1} \left[b_{i_0 i_1} + \tilde{b}_{i_0 i_1}(l) \right] \prod_{i \neq i_0,i_1}^{m-1} D_i \\
& + \sum_{i_0}^{m-1} \left[a_{i_0} + \tilde{a}_{i_0}(l) \right] \prod_{i \neq i_0}^{m-1} D_i \\
& + \tilde{P}(l) \prod_{i}^{m-1} D_i
\end{aligned}
$$

Now, choosing *l* such that

$$D_0(l^i) = D_1(l^i) = 0$$

sets all the terms in this equation to zero except the first, second and third line

Coefficient computed in a previous step

OPP Reduction

$$
\begin{aligned}
N(l) = & \sum_{i_0<i_1<i_2<i_3}^{m-1} \left[d_{i_0 i_1 i_2 i_3} + \tilde{d}_{i_0 i_1 i_2 i_3}(l) \right] \prod_{i \neq i_0,i_1,i_2,i_3}^{m-1} D_i \\
& + \sum_{i_0<i_1<i_2}^{m-1} \left[c_{i_0 i_1 i_2} + \tilde{c}_{i_0 i_1 i_2}(l) \right] \prod_{i \neq i_0,i_1,i_2}^{m-1} D_i \\
& + \sum_{i_0<i_1}^{m-1} \left[b_{i_0 i_1} + \tilde{b}_{i_0 i_1}(l) \right] \prod_{i \neq i_0,i_1}^{m-1} D_i \\
& + \sum_{i_0}^{m-1} \left[a_{i_0} + \tilde{a}_{i_0}(l) \right] \prod_{i \neq i_0}^{m-1} D_i \\
& + \tilde{P}(l) \prod_{i}^{m-1} D_i \\
= & \; 0
\end{aligned}
$$

Now, choosing *l* such that

$$D_0(l^i) = D_1(l^i) = 0$$

sets all the terms in this equation to zero except the first, second and third line

Coefficient computed in a previous step

OPP Reduction

$$N(l) = \sum_{i_0<i_1<i_2<i_3}^{m-1} \left[d_{i_0 i_1 i_2 i_3} + \tilde{d}_{i_0 i_1 i_2 i_3}(l)\right] \prod_{i\neq i_0,i_1,i_2,i_3}^{m-1} D_i$$

$$+ \sum_{i_0<i_1<i_2}^{m-1} \left[c_{i_0 i_1 i_2} + \tilde{c}_{i_0 i_1 i_2}(l)\right] \prod_{i\neq i_0,i_1,i_2}^{m-1} D_i$$

$$+ \sum_{i_0<i_1}^{m-1} \left[b_{i_0 i_1} + \tilde{b}_{i_0 i_1}(l)\right] \prod_{i\neq i_0,i_1}^{m-1} D_i$$

$$+ \sum_{i_0}^{m-1} \left[a_{i_0} + \tilde{a}_{i_0}(l)\right] \prod_{i\neq i_0}^{m-1} D_i$$

$$+ \tilde{P}(l) \prod_{i}^{m-1} D_i$$

Now, choosing *l* such that

$$D_1(l^i) = 0$$

sets the last line to zero

OPP Reduction

$$
\begin{aligned}
N(l) = & \sum_{i_0<i_1<i_2<i_3}^{m-1} \left[d_{i_0 i_1 i_2 i_3} + \tilde{d}_{i_0 i_1 i_2 i_3}(l) \right] \prod_{i \neq i_0,i_1,i_2,i_3}^{m-1} D_i \\
& + \sum_{i_0<i_1<i_2}^{m-1} \left[c_{i_0 i_1 i_2} + \tilde{c}_{i_0 i_1 i_2}(l) \right] \prod_{i \neq i_0,i_1,i_2}^{m-1} D_i \\
& + \sum_{i_0<i_1}^{m-1} \left[b_{i_0 i_1} + \tilde{b}_{i_0 i_1}(l) \right] \prod_{i \neq i_0,i_1}^{m-1} D_i \\
& + \sum_{i_0}^{m-1} \left[a_{i_0} + \tilde{a}_{i_0}(l) \right] \prod_{i \neq i_0}^{m-1} D_i \\
& + \tilde{P}(l) \prod_{i}^{m-1} D_i \\
= & \; 0
\end{aligned}
$$

Now, choosing *l* such that

$$D_1(l^i) = 0$$

sets the last line to zero

Complications in d-dimensions

- The previous expression should in fact be written in d-dimensions
- In the t'HV scheme, external momenta and polarization vectors are in 4 dimensions; only the loop momentum is in d dimensions

- The integral to be computed therefore reads

$$\int d^d l \, \frac{N(l,\tilde{l})}{\bar{D}_0 \bar{D}_1 \bar{D}_2 \cdots \bar{D}_{m-1}}$$

$$\bar{l} = l + \tilde{l}$$

d dim 4 dim epsilon dim

$$\bar{D}_i = (\bar{l} + p_i)^2 - m_i^2 = (l + p_i)^2 - m_i^2 + \tilde{l}^2 = D_i + \tilde{l}^2$$

$$l \cdot \tilde{l} = 0 \qquad \bar{l} \cdot p_i = l \cdot p_i \qquad \bar{l} \cdot \bar{l} = l \cdot l + \tilde{l} \cdot \tilde{l}$$

Complications in d-dimensions

$$R = R_1 + R_2$$

Complications in d-dimensions

- The d-dimensional contribution gives rise to the rational term which splits into two contributions

$$R = R_1 + R_2$$

- R_1 can be directly computed by the reduction algorithm, while R_2 can be computed from a finite set of process-independent additional Feynman rules.

Complications in d-dimensions

- The d-dimensional contribution gives rise to the rational term which splits into two contributions

$$R = R_1 + R_2$$

- R_1 can be directly computed by the reduction algorithm, while R_2 can be computed from a finite set of process-independent additional Feynman rules.

- R1: originates from the propagator (calculated in the reduction)

Complications in d-dimensions

- The d-dimensional contribution gives rise to the rational term which splits into two contributions

$$R = R_1 + R_2$$

- R_1 can be directly computed by the reduction algorithm, while R_2 can be computed from a finite set of process-independent additional Feynman rules.
  - R1: originates from the propagator (calculated in the reduction)
  - R2: originates from the numerator (additional Feynman rules)

R_2

Loop amplitude:

$$\frac{1}{(2\pi)^4}\int d^d\bar{q}\frac{\bar{N}(\bar{q})}{\bar{D}_0\bar{D}_1\cdots\bar{D}_{m-1}} \quad , \quad \bar{D}_i = (\bar{q}+p_i)^2 - m_i^2$$

R_2

Loop amplitude: $$\frac{1}{(2\pi)^4}\int d^d\bar{q}\frac{\bar{N}(\bar{q})}{\bar{D}_0\bar{D}_1\cdots\bar{D}_{m-1}} \quad , \quad \bar{D}_i=(\bar{q}+p_i)^2-m_i^2$$

Problem : numerical technique can only evaluate the numerator in 4 dimensions

R2

Loop amplitude: $$\frac{1}{(2\pi)^4}\int d^d\bar{q}\frac{\bar{N}(\bar{q})}{\bar{D}_0\bar{D}_1\cdots\bar{D}_{m-1}} \quad , \quad \bar{D}_i=(\bar{q}+p_i)^2-m_i^2$$

Problem : numerical technique can only evaluate the numerator in 4 dimensions

Solution : isolate the ε-dim part of the numerator: $$\underbrace{\bar{N}(\bar{q})}_{\text{d-dim}}=\underbrace{N(q)}_{\text{4-dim}}+\underbrace{\tilde{N}(\tilde{q},q,\epsilon)}_{\epsilon-dim}$$

R2

Loop amplitude: $$\frac{1}{(2\pi)^4}\int d^d\bar{q}\frac{\bar{N}(\bar{q})}{\bar{D}_0\bar{D}_1\cdots\bar{D}_{m-1}} \quad , \quad \bar{D}_i=(\bar{q}+p_i)^2-m_i^2$$

Problem : numerical technique can only evaluate the numerator in 4 dimensions

Solution : isolate the ε-dim part of the numerator: $$\underbrace{\bar{N}(\bar{q})}_{\text{d-dim}}=\underbrace{N(q)}_{\text{4-dim}}+\underbrace{\tilde{N}(\tilde{q},q,\epsilon)}_{\epsilon-dim}$$

Then : compute analytically the finite set of loops for which its contribution does not vanish, and re-express it in terms of an R2 Feynman rules.

$$R2\equiv\lim_{\epsilon\to 0}\frac{1}{(2\pi)^4}\int d^d\bar{q}\frac{\tilde{N}(\tilde{q},q,\epsilon)}{\bar{D}_0\bar{D}_1\cdots\bar{D}_{m-1}}$$

R2

Loop amplitude: $$\frac{1}{(2\pi)^4}\int d^d\bar{q}\frac{\bar{N}(\bar{q})}{\bar{D}_0\bar{D}_1\cdots\bar{D}_{m-1}} \quad , \quad \bar{D}_i=(\bar{q}+p_i)^2-m_i^2$$

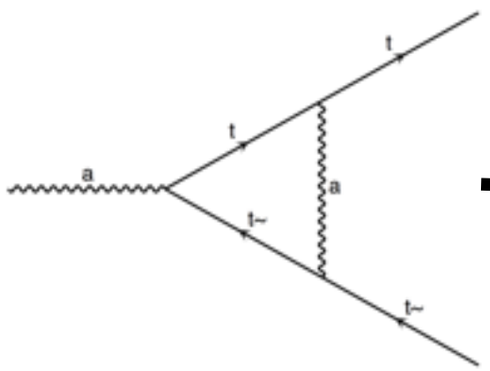
Problem : numerical technique can only evaluate the numerator in 4 dimensions

Solution : isolate the ε-dim part of the numerator: $$\underbrace{\bar{N}(\bar{q})}_{\text{d-dim}}=\underbrace{N(q)}_{\text{4-dim}}+\underbrace{\tilde{N}(\tilde{q},q,\epsilon)}_{\epsilon-dim}$$

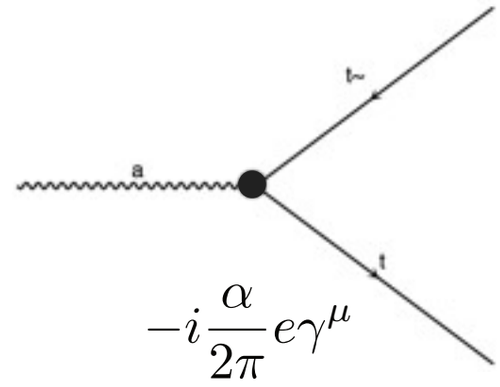
Then : compute analytically the finite set of loops for which its contribution does not vanish, and re-express it in terms of an R2 Feynman rules.

$$R2\equiv\lim_{\epsilon\to 0}\frac{1}{(2\pi)^4}\int d^d\bar{q}\frac{\tilde{N}(\tilde{q},q,\epsilon)}{\bar{D}_0\bar{D}_1\cdots\bar{D}_{m-1}}$$

Ex. :

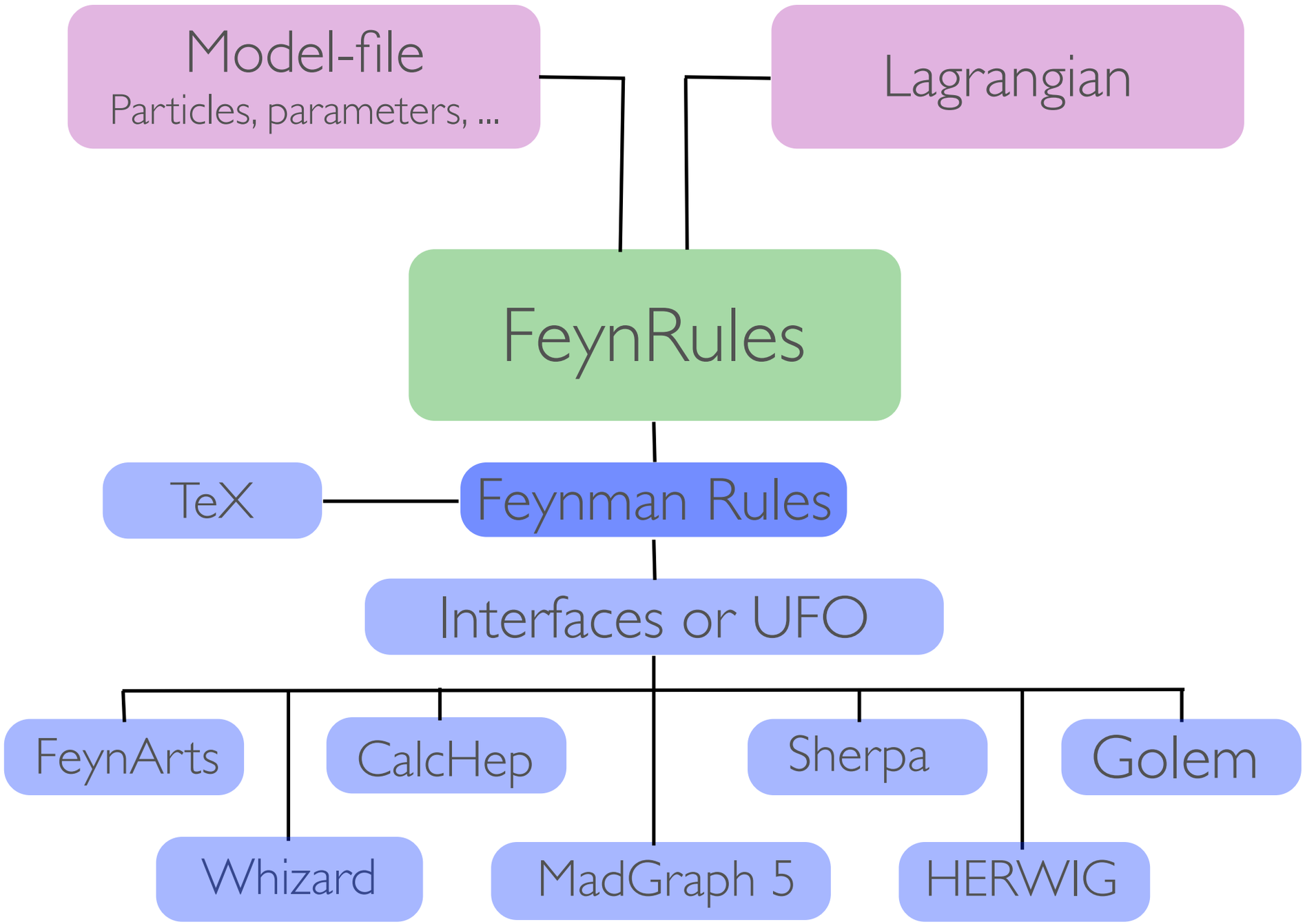


$$\supset\int d^d\bar{q}\frac{\tilde{q}^2}{\bar{D}_0\bar{D}_1\bar{D}_2}=-\frac{i\pi^2}{2}+\mathcal{O}(\epsilon)$$



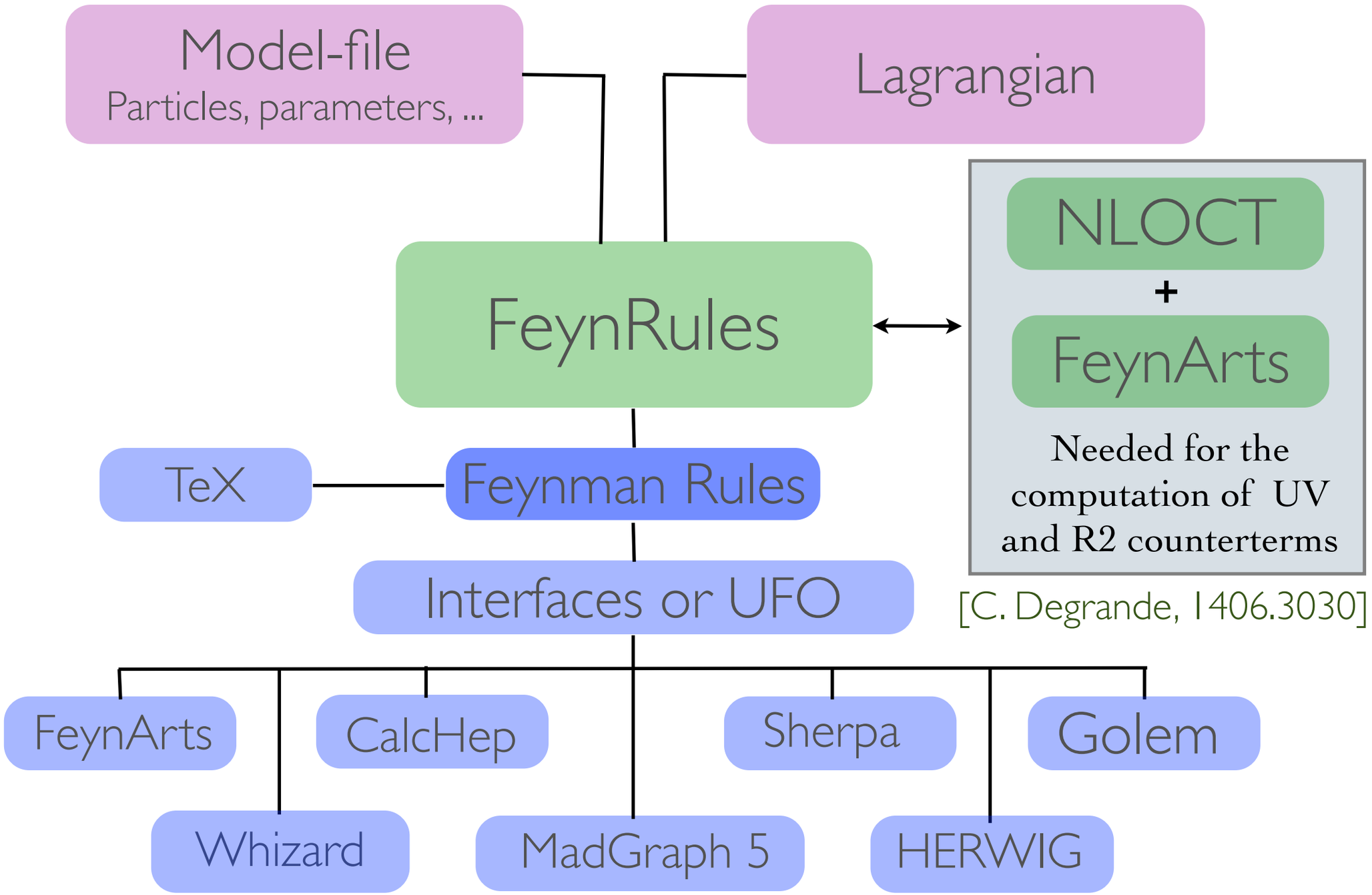
FeynRules @ NLO (version 2.1)

[Alloul, N. Christensen, C. Degrande, C. Duhr, B.Fuks, in 1310.1921]



FeynRules @ NLO (version 2.1)

[Alloul, N. Christensen, C. Degrande, C. Duhr, B.Fuks, in 1310.1921]



NLOCT

[C. Degrande, arXiv:1412.6955]

- UV counterterms:

NLOCT

[C. Degrande, arXiv:1412.6955]

- UV counterterms:

  A) Renormalize the Lagrangian

$$\left.\begin{array}{ll} \text{Fields} & \phi_0 \to (1+\frac{1}{2}\delta Z_{\phi\phi}) + \sum_\chi \frac{1}{2}\delta Z_{\phi\chi}\chi \\ \text{ext. params} & x_0 \to x + \delta x \\ \text{int. params} & g(x) \to g(x+\delta x) \end{array}\right\} \quad \mathcal{L}_0 \to \mathcal{L} + \delta\mathcal{L}$$

NLOCT

[C. Degrande, arXiv:1412.6955]

- UV counterterms:

  A) Renormalize the Lagrangian

$$\left.\begin{array}{ll} \text{Fields} & \phi_0 \to (1+\frac{1}{2}\delta Z_{\phi\phi}) + \sum_\chi \frac{1}{2}\delta Z_{\phi\chi}\chi \\ \text{ext. params} & x_0 \to x + \delta x \\ \text{int. params} & g(x) \to g(x+\delta x) \end{array}\right\} \quad \mathcal{L}_0 \to \mathcal{L} + \delta\mathcal{L}$$

  B) Compute the defining loops

  → Done in FeynArts. Notice that for $\overline{MS}$, only poles are needed.

NLOCT

[C. Degrande, arXiv:1412.6955]

- UV counterterms:

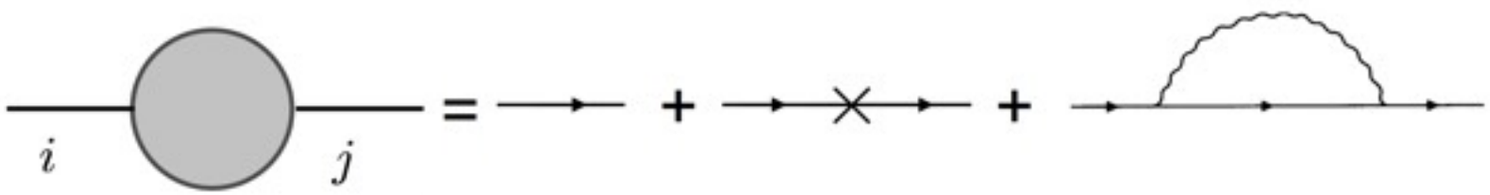
  A) Renormalize the Lagrangian

$$\left.\begin{array}{ll} \text{Fields} & \phi_0 \to (1+\frac{1}{2}\delta Z_{\phi\phi}) + \sum_\chi \frac{1}{2}\delta Z_{\phi\chi}\chi \\ \text{ext. params} & x_0 \to x + \delta x \\ \text{int. params} & g(x) \to g(x+\delta x) \end{array}\right\} \quad \mathcal{L}_0 \to \mathcal{L} + \delta\mathcal{L}$$

  B) Compute the defining loops

  → Done in FeynArts. Notice that for $\overline{MS}$, only poles are needed.

  C) Solve for the counterterms by applying renormalization conditions



NLOCT

[C. Degrande, arXiv:1412.6955]

- UV counterterms:

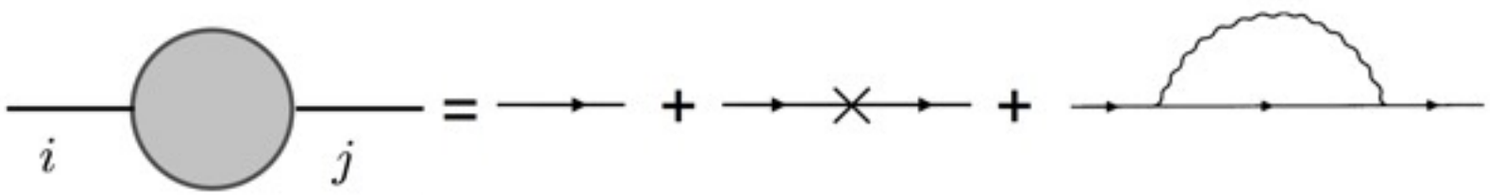
  A) Renormalize the Lagrangian

$$\left.\begin{array}{ll} \text{Fields} & \phi_0 \to (1+\frac{1}{2}\delta Z_{\phi\phi}) + \sum_\chi \frac{1}{2}\delta Z_{\phi\chi}\chi \\ \text{ext. params} & x_0 \to x + \delta x \\ \text{int. params} & g(x) \to g(x+\delta x) \end{array}\right\} \quad \mathcal{L}_0 \to \mathcal{L} + \delta\mathcal{L}$$

  B) Compute the defining loops

  → Done in FeynArts. Notice that for $\overline{MS}$, only poles are needed.

  C) Solve for the counterterms by applying renormalization conditions



  D) Derive and output the corresponding UV counterterms.

NLOCT

[C. Degrande, arXiv:1412.6955]

- UV counterterms:

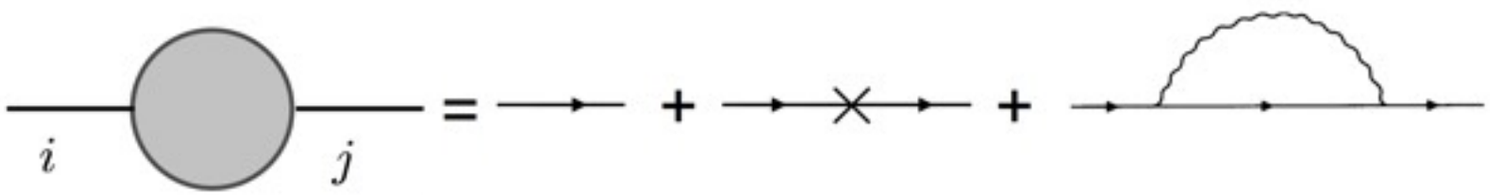
  A) Renormalize the Lagrangian

$$\left.\begin{array}{ll} \text{Fields} & \phi_0 \to (1+\frac{1}{2}\delta Z_{\phi\phi}) + \sum_\chi \frac{1}{2}\delta Z_{\phi\chi}\chi \\ \text{ext. params} & x_0 \to x + \delta x \\ \text{int. params} & g(x) \to g(x+\delta x) \end{array}\right\} \quad \mathcal{L}_0 \to \mathcal{L} + \delta\mathcal{L}$$

  B) Compute the defining loops

  → Done in FeynArts. Notice that for $\overline{MS}$, only poles are needed.

  C) Solve for the counterterms by applying renormalization conditions



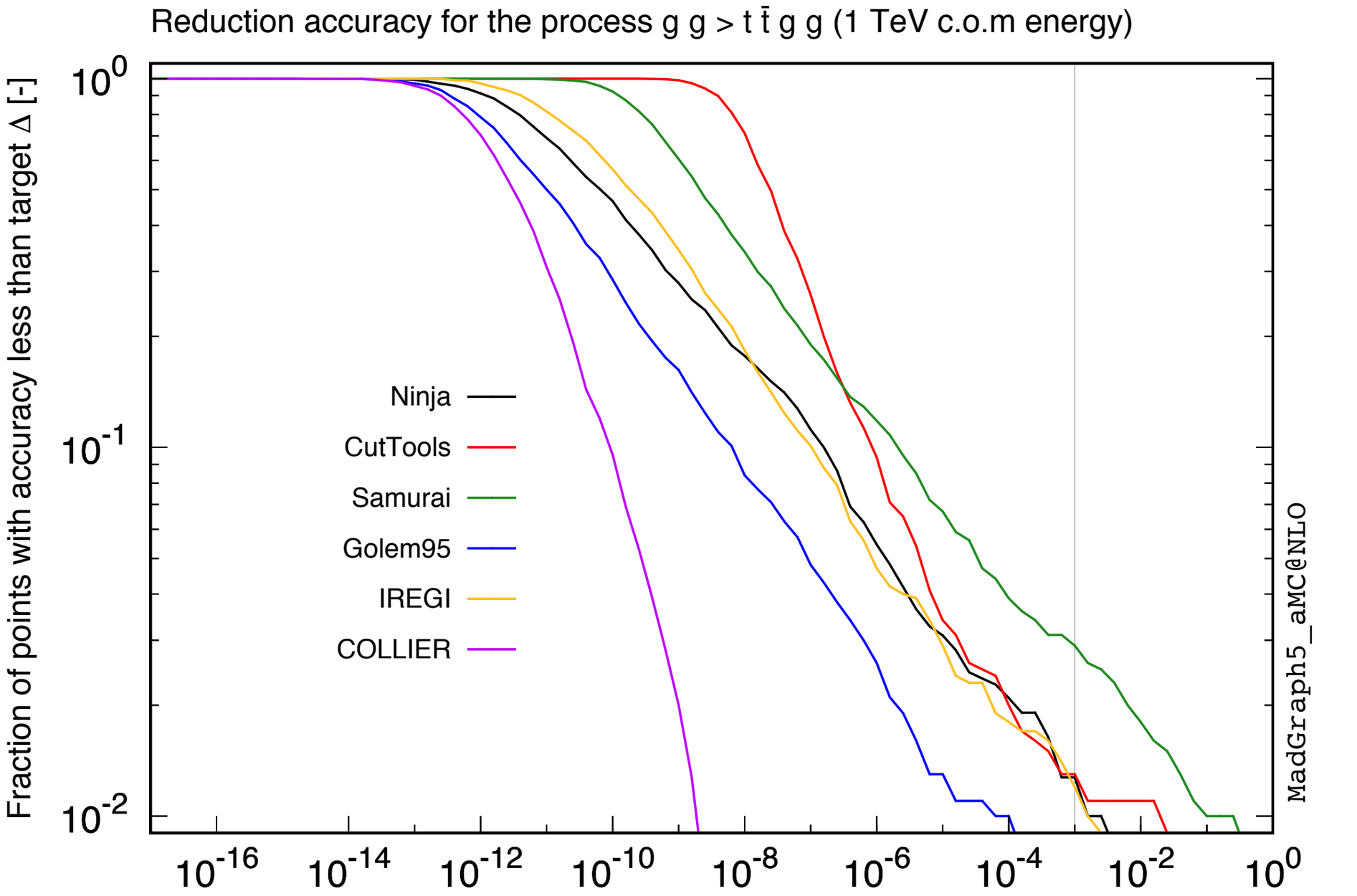
  D) Derive and output the corresponding UV counterterms.

- R2 counterterms, computed using FeynArts amplitudes as well.

TIR vs OPP: different performances

NINJA and COLLIER interfaced to MadLoop

[VH, T. Peraro, 1604.01363]



Unmatched numerical stability with COLLIER

Plan

- Build your model : FeynRules + UFO
- Computing amplitudes with Feynman diagrams
- Special techniques relevant to BSM
- Simulations at NLO accuracy
- MadLoop : evaluating loop Matrix Elements
- Recent loopy applications within MG5_aMC

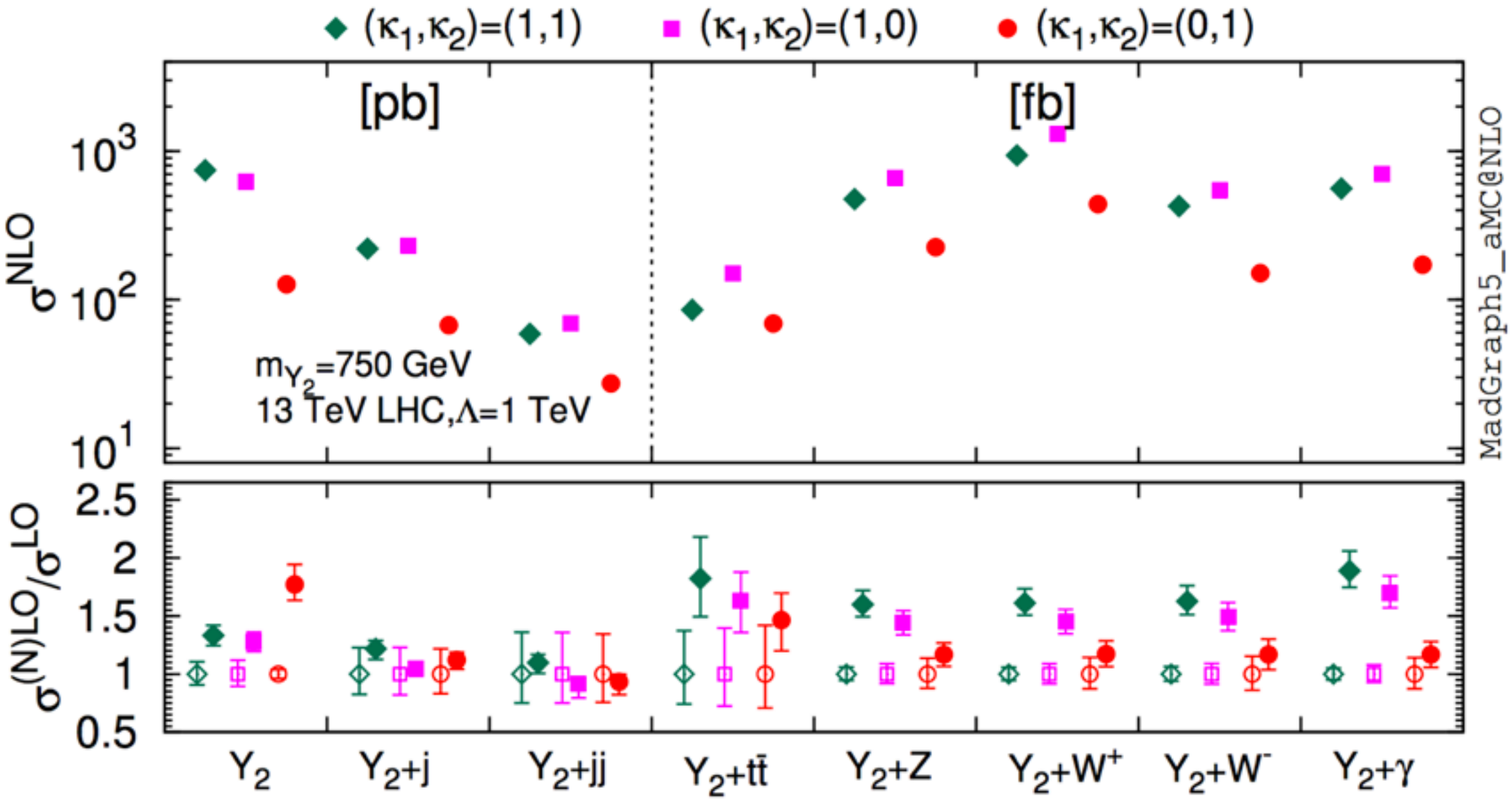
Plan

- Build your model : FeynRules + UFO
- Computing amplitudes with Feynman diagrams
- Special techniques relevant to BSM
- Simulations at NLO accuracy
- MadLoop : evaluating loop Matrix Elements
- Recent loopy applications within MG5_aMC

Spin-2 production @ NLO

[G. Das, C. Degrande, VH, F. Maltoni, H-S. Shao, 1605.09359]

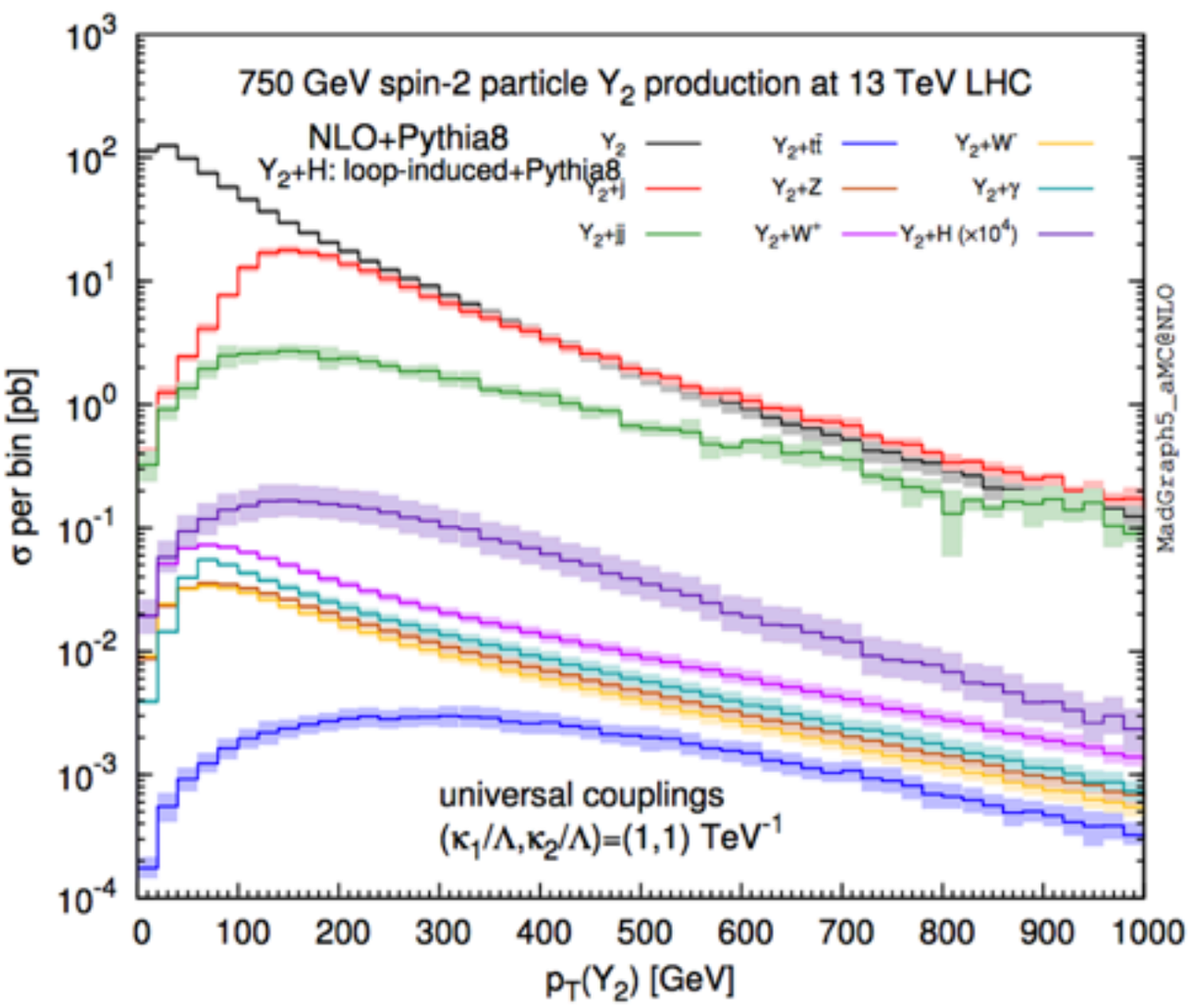
Simplified model: $\mathcal{L}^{Y_2}_{V,f} = -\frac{\kappa_{V,f}}{\Lambda} T^{V,f}_{\mu\nu} Y_2^{\mu\nu}$



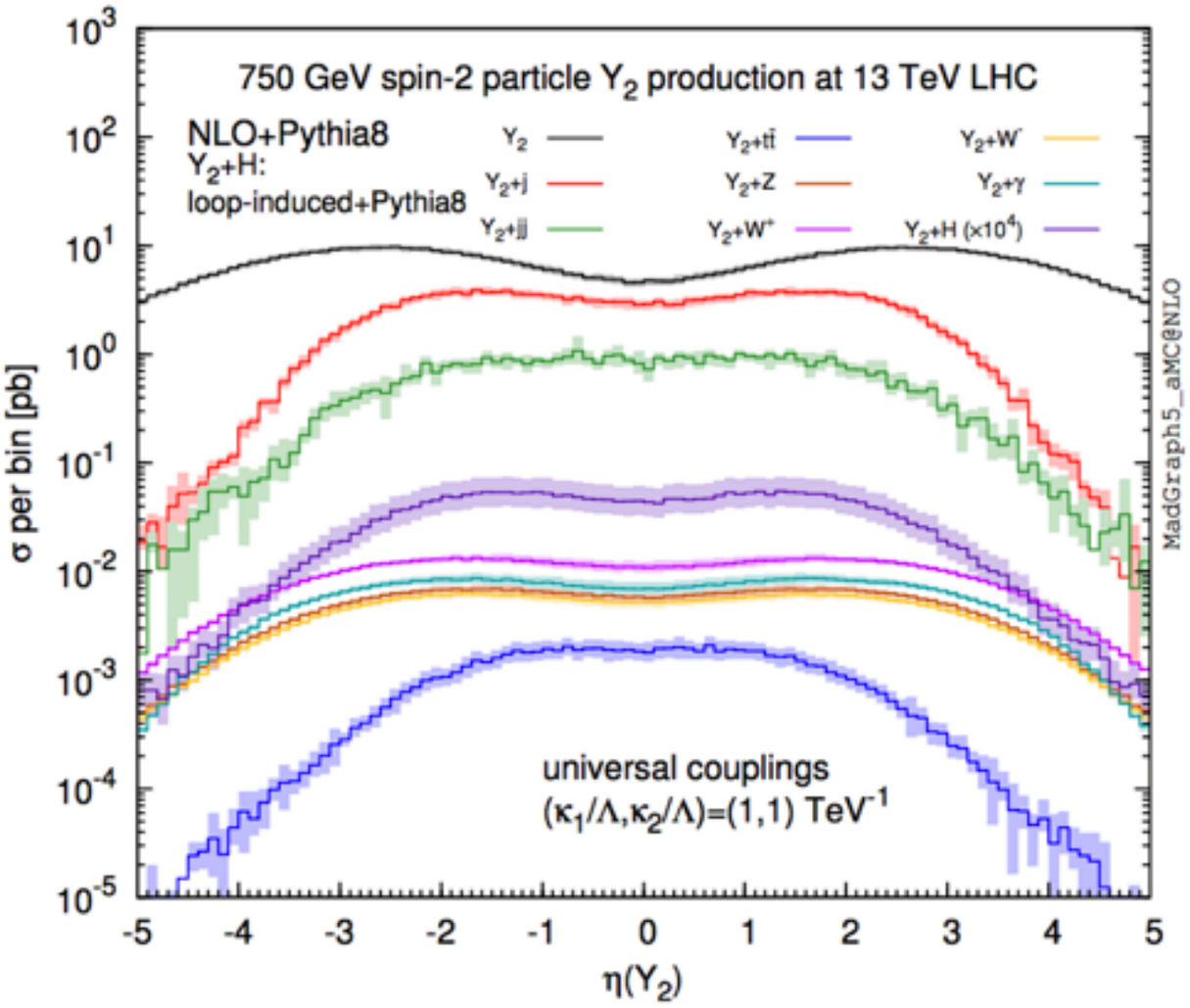
Large K-factors, unreliable uncertainties at LO.

Spin-2 : NLO QCD matched

[G. Das, C. Degrande, VH, F. Maltoni, H-S. Shao, 1605.09359]



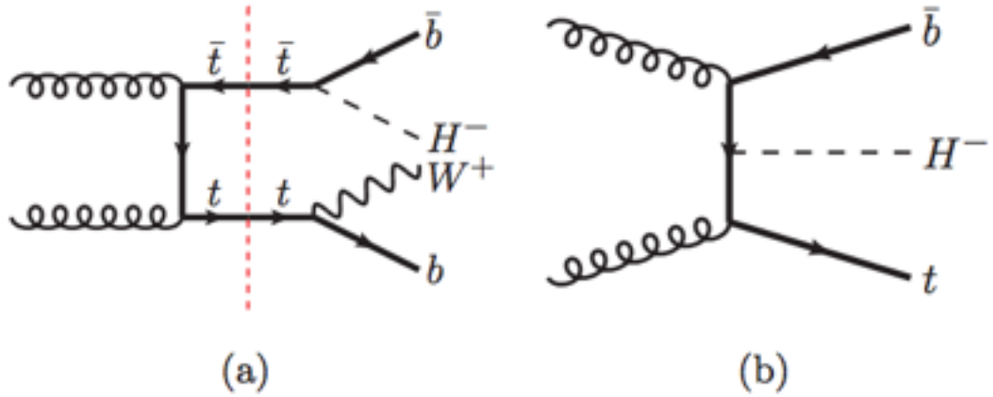
(a) Transverse momentum distribution



(b) Pseudorapidity distribution

H^+ prod. @ NLO, $M_H \sim M_T$

[C. Degrande, R. Frederix, VH, M. Ubiali, M. Wiesemann, M. Zaro 1607.05291]



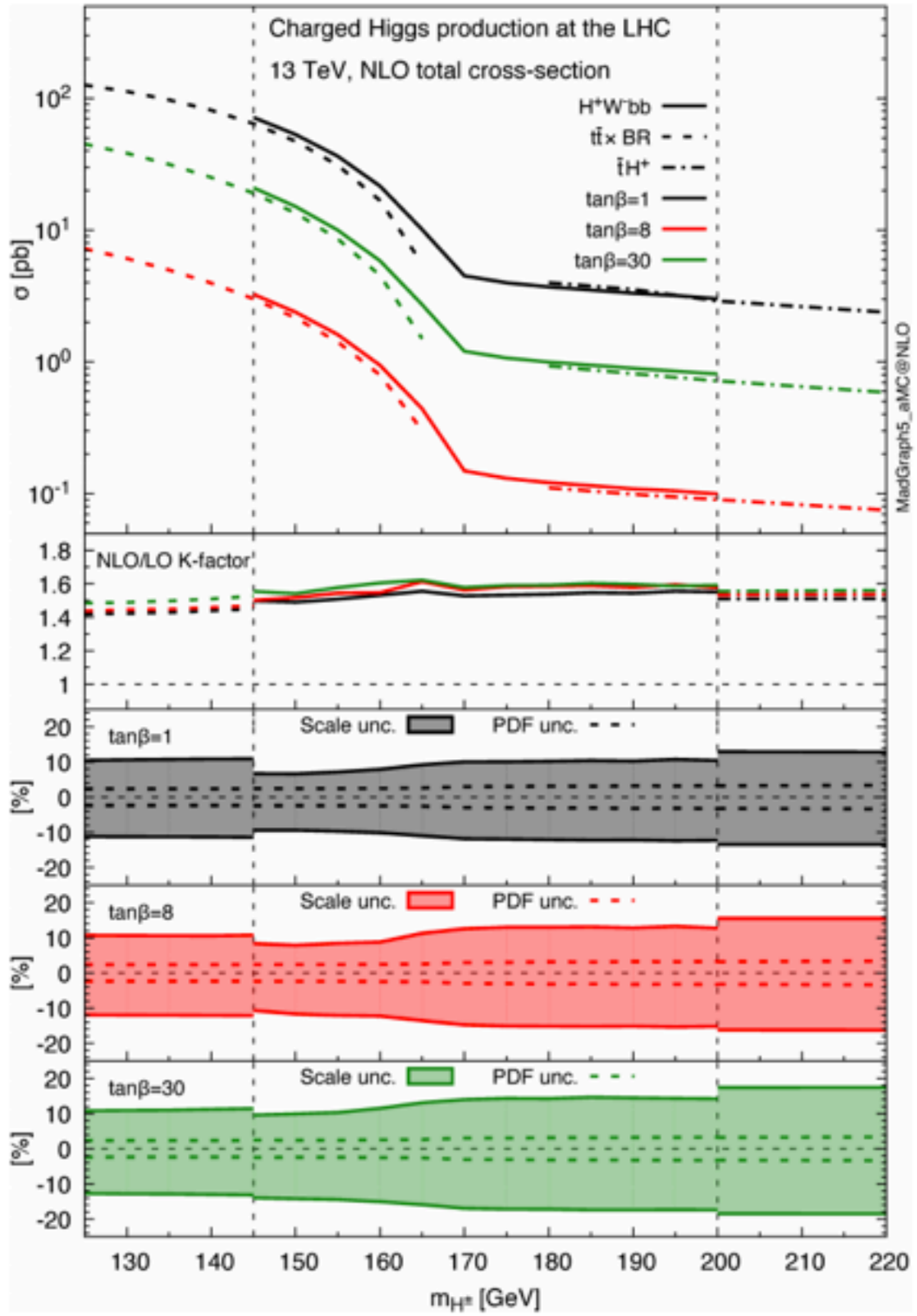
a) dominates for $m_H < 145$ GeV

b) dominates for $m_H > 200$ GeV

a)+b) For 145 GeV$< m_H <$ 200 GeV

-> Requires to honestly compute:

p p > H^+ W^- b b



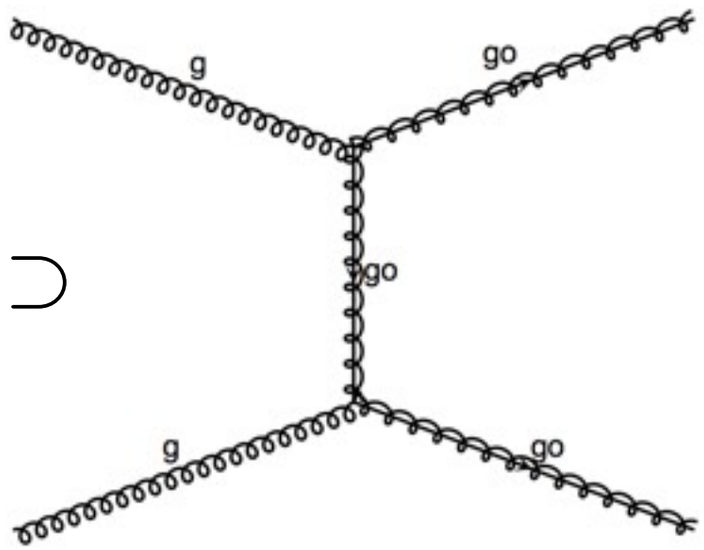
Towards full MSSM@NLO

SUSY QCD for the QCD sector only is already available in
C. Degrande, B. Fuks, V. H., J. Proudom, H-S.Shao [arXiv:1510.00391]

- Gluinos pair production...

$$\begin{aligned}\mathcal{L}_{\text{SQCD}} =& D_\mu \tilde{q}_L^\dagger D^\mu \tilde{q}_L + D_\mu \tilde{q}_R^\dagger D^\mu \tilde{q}_R + \frac{i}{2}\bar{\tilde{g}} \not{D} \tilde{g} \\ & - m^2_{\tilde{q}_L} \tilde{q}_L^\dagger \tilde{q}_L - m^2_{\tilde{q}_R} \tilde{q}_R^\dagger \tilde{q}_R - \frac{1}{2} m_{\tilde{g}} \bar{\tilde{g}} \tilde{g} \\ & + \sqrt{2} g_s \Big[- \tilde{q}_L^\dagger T (\bar{\tilde{g}} P_L q) + (\bar{q} P_L \tilde{g}) T \tilde{q}_R + \text{h.c.} \Big] \\ & - \frac{g_s^2}{2} \Big[\tilde{q}_R^\dagger T \tilde{q}_R - \tilde{q}_L^\dagger T \tilde{q}_L \Big] \Big[\tilde{q}_R^\dagger T \tilde{q}_R - \tilde{q}_L^\dagger T \tilde{q}_L \Big]\end{aligned}$$

$\supset$

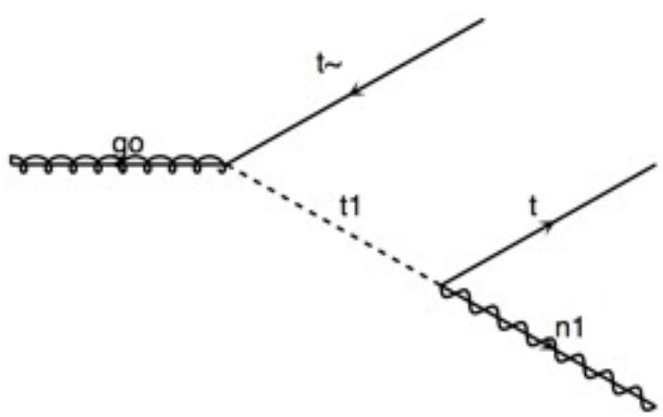


- ... including the squark decay.

$$\begin{aligned}\mathcal{L}_{\text{decay}} =& \frac{i}{2} \bar{\chi} \not{\partial} \chi - \frac{1}{2} m_\chi \bar{\chi} \chi \\ & + \sqrt{2} g' \Big[- \tilde{q}_L^\dagger Y_q (\bar{\chi} P_L q) + (\bar{q} P_L \chi) Y_q \tilde{q}_R + \text{h.c.} \Big]\end{aligned}$$

$\otimes$

$\supset$



Majorana flow, top quark mixing matrix renorm, SUSY restoring CT: Solved.

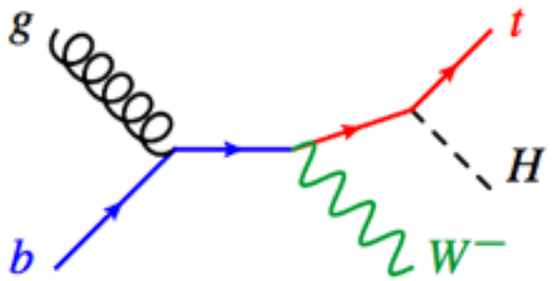
Complete SUSY Model for NLO QCD

- Requires improvements in NLOCT and further validation of the complex mass scheme.

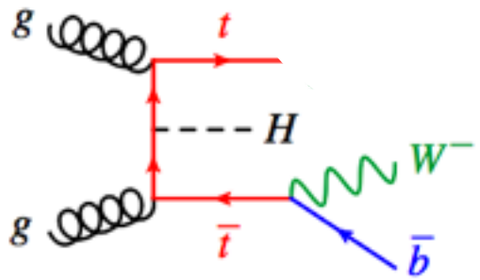
- A key component here is Onshell-Subtraction (OS) in aMC@NLO, which is now available, and was introduced in

F. Demartin, B. Maier, F. Maltoni, K. Mawatari, M. Zaro [arXiv:1607.05862]

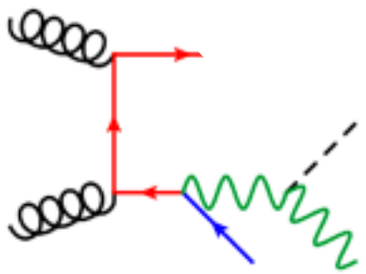
LO ⊃



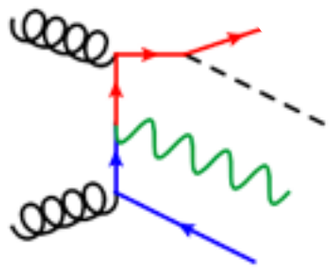
NLO (reals) ⊃



$(\mathcal{A}_{2t})$ X



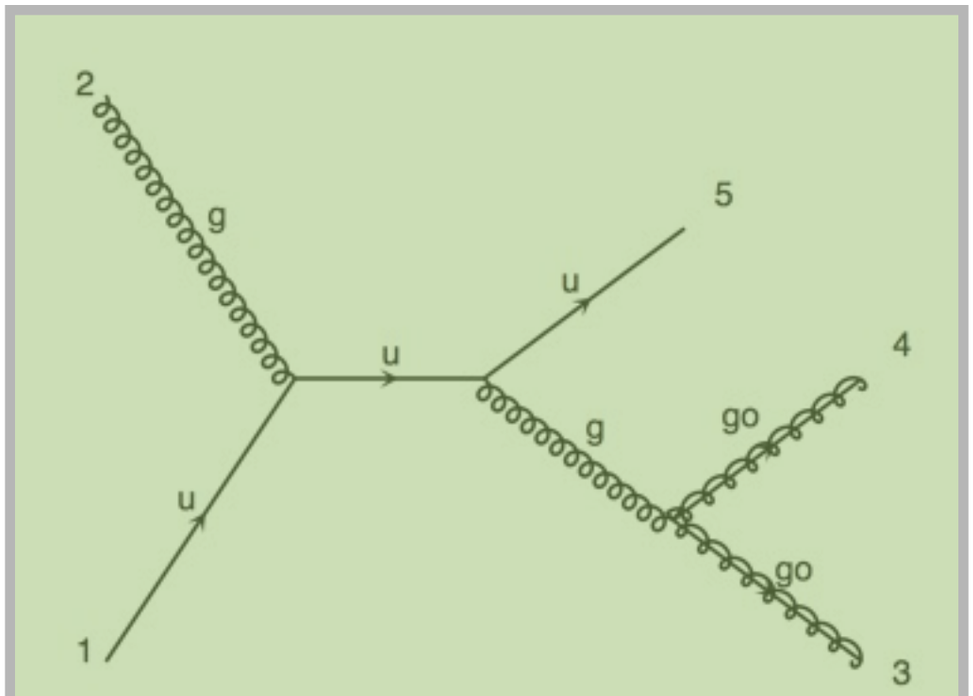
$(\mathcal{A}_{1t})$ ✓



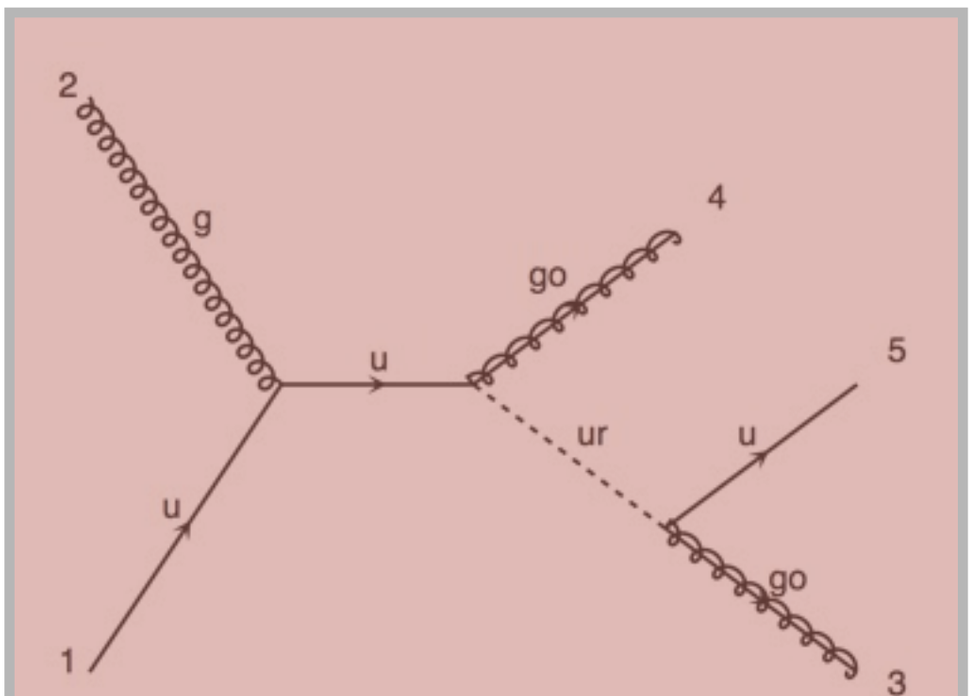
$(\mathcal{A}_{1t})$ ✓

Onshell subtraction for SUSY

Similar problem occurring in, e.g. $pp \to \tilde{g}\tilde{g}$



O.K, this is a legitimate QCD correction to the original Born process definition.

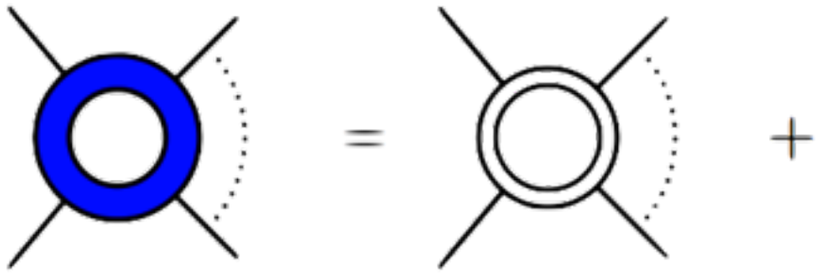


NO, this is simply the process $pp \to \tilde{g}\tilde{u}_R$ in disguise! Should be removed.

Final Ride: Loop-induced

What is different with Loop-induced (LI) ?

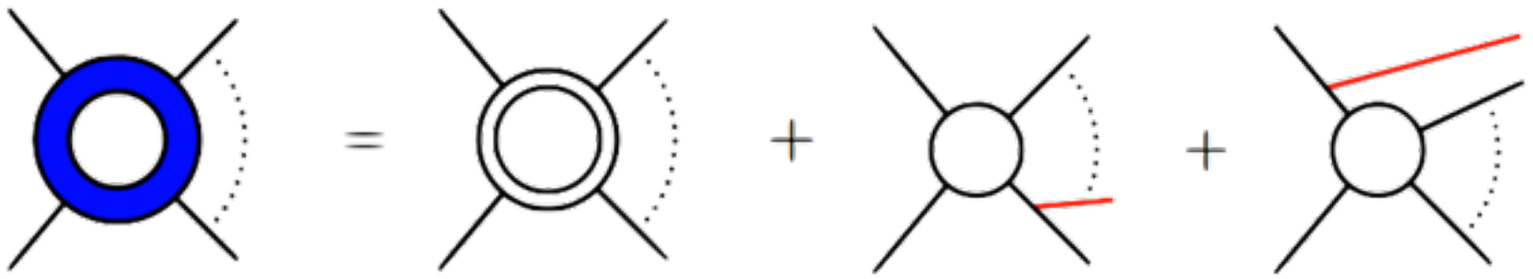
[VH, O. Mattelaer, 1507.00020]



$$\sigma^{\text{NLO}} = \int_m d^{(d)}\sigma^V +$$

What is different with Loop-induced (LI) ?

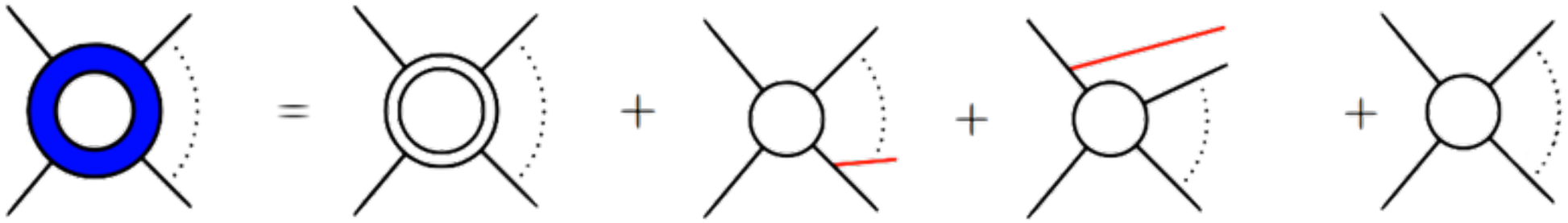
[VH, O. Mattelaer, 1507.00020]



$$\sigma^{\mathrm{NLO}} = \int_m d^{(d)}\sigma^V + \int_{m+1} d^{(d)}\sigma^R +$$

What is different with Loop-induced (LI) ?

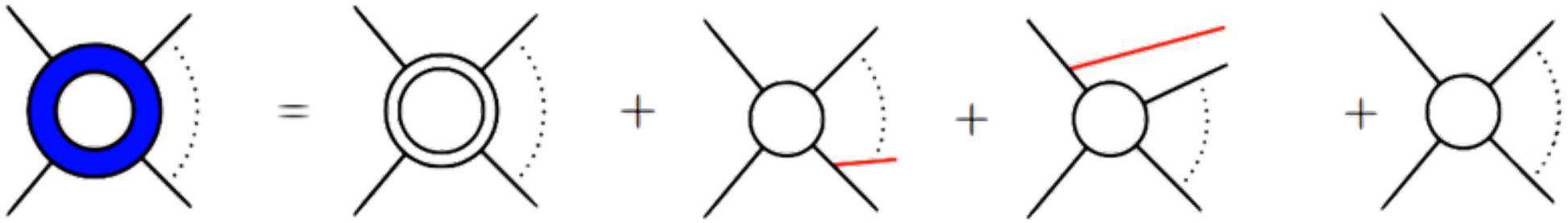
[VH, O. Mattelaer, 1507.00020]



$$\sigma^{\text{NLO}} = \int_m d^{(d)}\sigma^V + \int_{m+1} d^{(d)}\sigma^R + \int_m d^{(4)}\sigma^B$$

What is different with Loop-induced (LI) ?

[VH, O. Mattelaer, 1507.00020]



$$\sigma^{\mathrm{NLO}} = \int_m d^{(d)}\sigma^V + \int_{m+1} d^{(d)}\sigma^R + \int_m d^{(4)}\sigma^B$$

$$\sigma^{\mathrm{LI}} = \left| \quad \right|^2 = \int_m d^{(d)} \left|\mathcal{A}^{(1)}\right|^2$$

Loop-induced

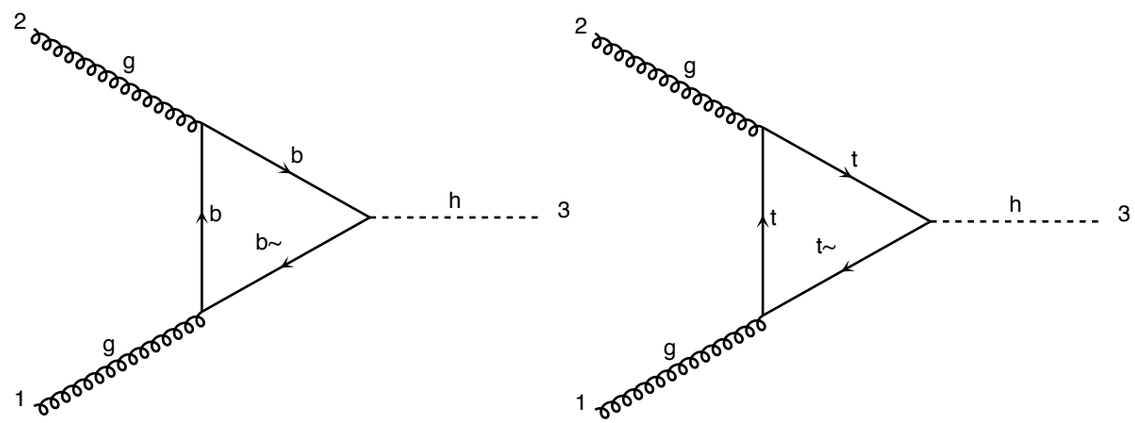
[VH, O. Mattelaer, 1507.00020]

User Input

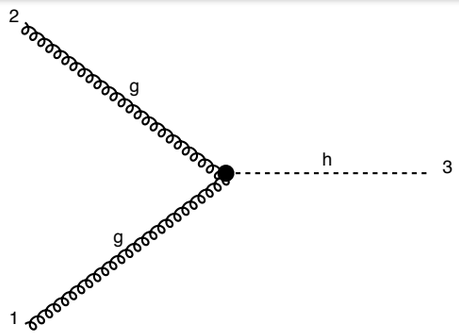
- generate g g > h [QCD]
- output
- launch

Loop Induced

$$\sigma_{loop} = 15.74(2)pb$$



HEFT



$$\sigma_{heft} = 17.63(2)pb$$

Loop-induced

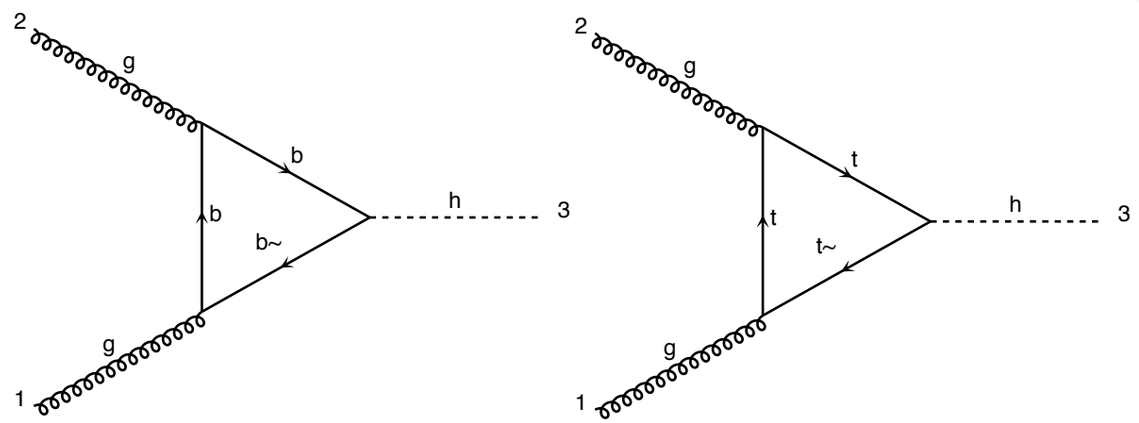
[VH, O. Mattelaer, 1507.00020]

User Input

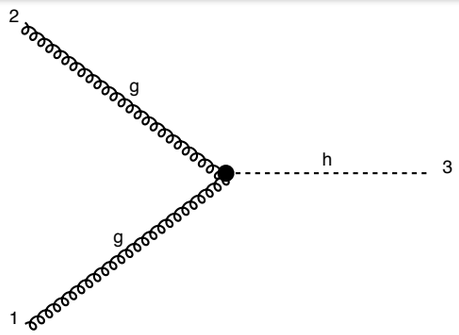
- generate g g > h [QCD]
- output
- launch

Loop Induced

$$\sigma_{loop} = 15.74(2)pb$$

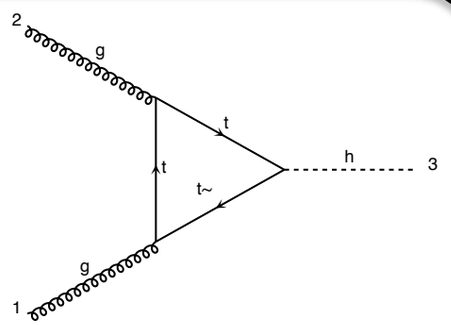


HEFT



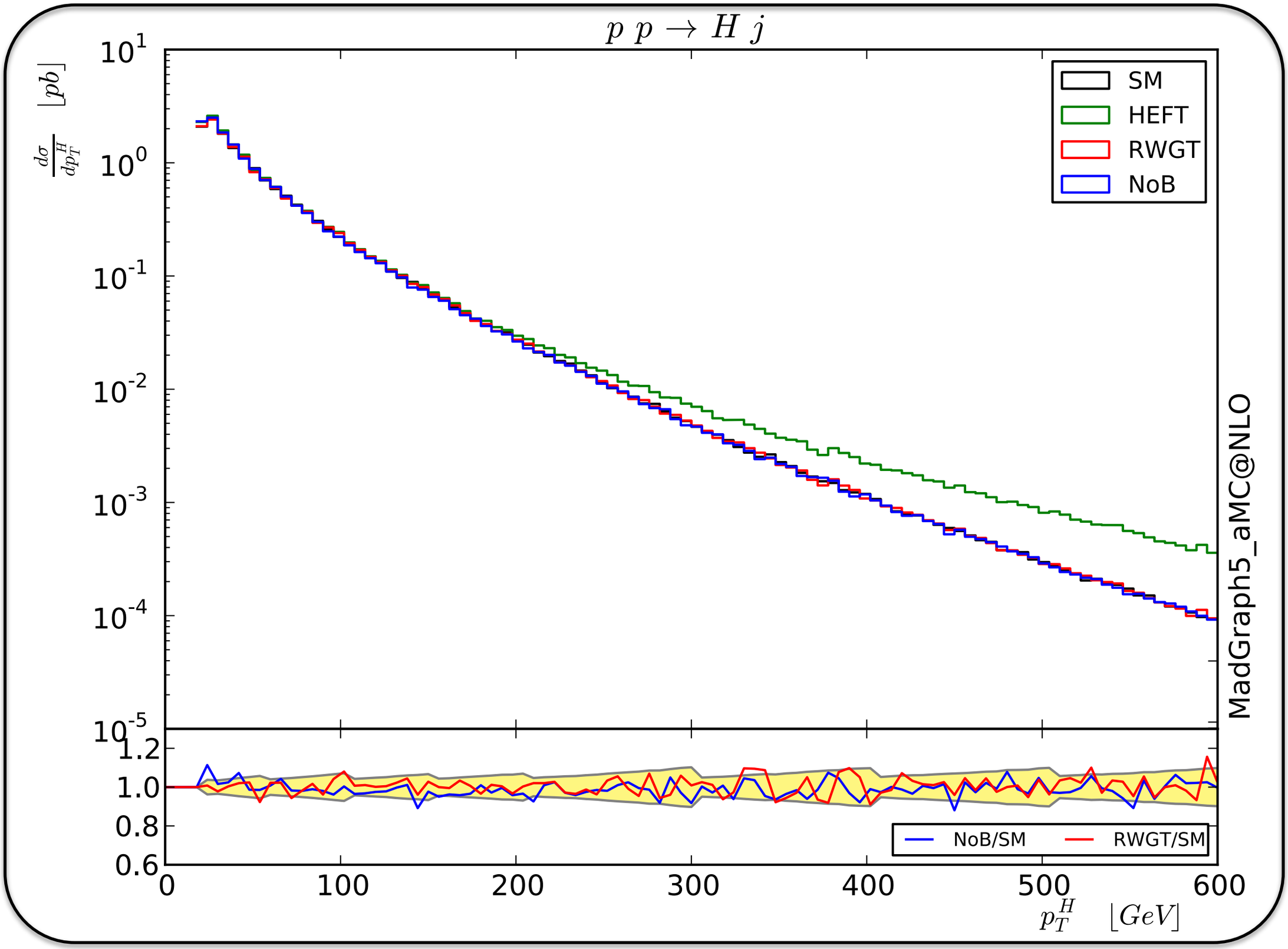
$$\sigma_{heft} = 17.63(2)pb$$

No bottom loop



$$\sigma_{toploop} = 17.65(2)pb$$

Validation p p > H J



Cross-sections tables

[VH, O. Mattelaer, 1507.00020]

[...]

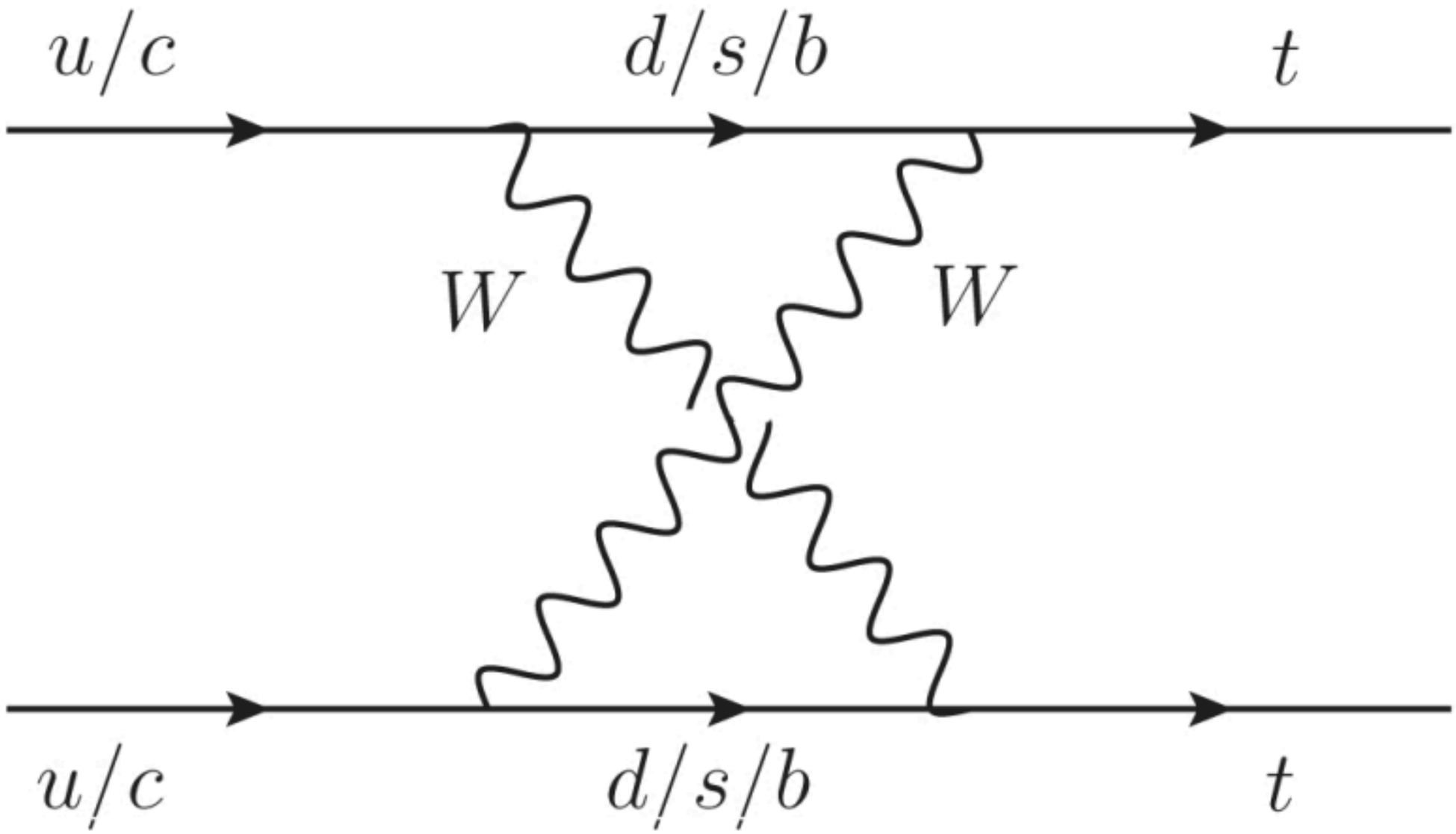
Process		Syntax	Cross section (pb)	$\Delta_{\hat{\mu}}$	Δ_{PDF}
Selected $2 \to 4$			$\sqrt{s} = 13$ TeV		
†d.1	$pp \to Hjjj$	`p p > h j j j QED=1 [QCD]`	2.519 ± 0.005	$^{+75.1\%}_{-39.8\%}$	$^{+0.6\%}_{-0.6\%}$
*d.2	$pp \to HHjj$	`p p > h h j j QED=1 [QCD]`	$1.085 \pm 0.002 \cdot 10^{-2}$	$^{+62.1\%}_{-35.8\%}$	$^{+1.2\%}_{-1.3\%}$
†d.3	$pp \to HHHj$	`p p > h h h j [QCD]`	$4.981 \pm 0.008 \cdot 10^{-5}$	$^{+46.3\%}_{-29.6\%}$	$^{+1.4\%}_{-1.4\%}$
†d.3	$pp \to HHHH$	`p p > h h h h [QCD]`	$1.080 \pm 0.003 \cdot 10^{-7}$	$^{+33.3\%}_{-23.4\%}$	$^{+1.7\%}_{-1.7\%}$
d.4	$gg \to e^+e^-\mu^+\mu^-$	`g g > e+ e- mu+ mu- [QCD]`	$2.022 \pm 0.003 \cdot 10^{-3}$	$^{+26.4\%}_{-19.4\%}$	$^{+0.7\%}_{-1.1\%}$
†d.5	$pp \to HZ\gamma j$	`g g > h z a g [QCD]`	$4.950 \pm 0.008 \cdot 10^{-6}$	$^{+45.8\%}_{-29.3\%}$	$^{+1.2\%}_{-1.3\%}$
Non-hadronic processes			$\sqrt{s} = 500$ GeV, no PDF		
†e.1	$e^+e^- \to ggg$	`e+ e- > g g g [QED]`	$2.526 \pm 0.004 \cdot 10^{-6}$	$^{+31.2\%}_{-22.0\%}$	
†e.2	$e^+e^- \to HH$	`e+ e- > h h [QED]`	$1.567 \pm 0.003 \cdot 10^{-5}$	$^{+0.0\%}_{-0.0\%}$	
†e.3	$e^+e^- \to HHgg$	`e+ e- > h h g g [QED]`	$6.629 \pm 0.010 \cdot 10^{-11}$	$^{+19.2\%}_{-14.8\%}$	
*e.4	$\gamma\gamma \to HH$	`a a > h h [QED]`	$3.198 \pm 0.005 \cdot 10^{-4}$	$^{+0.0\%}_{-0.0\%}$	
Miscellaneous			$\sqrt{s} = 13$ TeV		
†f.1	$pp \to tt$	`p p > t t [QED]`	$4.045 \pm 0.007 \cdot 10^{-15}$	$^{+0.2\%}_{-0.8\%}$	$^{+0.9\%}_{-1.0\%}$

★ : **Not publicly** available.

† : Computed here for the **first time**.

P P > T T, THE ANSWER

P P > T T, THE ANSWER



TIR vs OPP: different performances

[VH, T. Peraro, 1604.01363]

Add. scales and larg mult.	$gg \to t\bar{t}$	$gg \to t\bar{t}g$	$gg \to t\bar{t}gg$	$uu \to t\bar{t}b\bar{b}d\bar{d}$
Max. loop num. rank	3	4	5	4
Integrand computation time	0.26 ms	4.8 ms	170 ms	99 ms
NINJA reduction time	0.40 ms	5.3 ms	78 ms	104 ms
COLI and (DD)	0.83 (0.72)	13.6 (16.4)	220 (322)	1120 (N/A)
CUTTOOLS reduction time	1.3 ms	23.2 ms	330 ms	301 ms
COLLIER/ NINJA	2.1	2.6	2.8	10.8

Saturated rank (LI)	$gg \to 2 \cdot Z$	$gg \to 3 \cdot Z$	$gg \to 4 \cdot Z$
Max. loop num. rank	4	5	6
Integrand computation time	0.60 ms	7.2 ms	81 ms
NINJA reduction time	1.6 ms	21 ms	310 ms
COLI and (DD)	1.6 (1.6)	25 (46)	590 (661)
CUTTOOLS reduction time	4.1 ms	59 ms	1080 ms
COLLIER/ NINJA	1.0	1.2	1.9

Eff. theory, Y≡spin-2	$gg \to Yg$	$gg \to Ygg$	$gg \to Yggg$
Max. loop num. rank	5	6	7
Integrand computation time	2.2 ms	33 ms	1.4 s
NINJA reduction time	1.5 ms	20 ms	0.32 s
COLI reduction time	1.9 ms	57 ms	1.8 s
COLLIER/ NINJA	1.3	2.9	5.6

Table 1: All timings refer to the computation of the colour-summed loop amplitude for a single helicity and kinematic configuration . The test machine is using a single core of an Intel Core i7 CPU (2.7 GHz) and the MADLOOP is compiled with GNU `gfortran -O2` (v4.8.2).

NINJA slightly faster (ratio > 2) for large multiplicity processes.

MadLoop in MG5aMC

- Process generation
  - `import model <model_name>-<restrictions>`
  - `generate <process> <amp_orders_and_option> [<mode>=<pert_orders>] <squared_orders>`
  - `output <format> <folder_name>`
  - `launch <options>`
- Examples, starting from a default MG5aMC interface
  - Very simple one (in this case, generates the full code for NLO computations) :

```
[ 2.5s    ] generate p p > t t~ [QCD]
[ 6.1s    ] output
[ ~ mins*] launch
    * timing for 10k unweighted events on a laptop
```

  - With options specified (in this case, generates the one-loop matrix element code only):

```
[ 0.01s ] import model loop_sm-no_hwidth
[ 0.01s ] set complex_mass_scheme
[ 5min  ] generate g g > e+ ve mu- vm~ b b~ / h QED=2 [virt=QCD]
[ 2min  ] output MyProc
[ ~1 s* ] launch -f
    * time per phase-space point, summed over helicity configurations and colors.
```

Details on how to generate and use a MadLoop standalone library available @

cp3.irmp.ucl.ac.be/projects/madgraph/wiki/MadLoopStandaloneLibrary

MadLoop in MG5aMC

- Process generation
  - `import model <model_name>-<restrictions>`
  - `generate <process> <amp_orders_and_option> [<mode>=<pert_orders>] <squared_orders>`
  - `output <format> <folder_name>`
  - `launch <options>`
- Examples, starting from a default MG5aMC interface
  - Very simple one (in this case, generates the full code for NLO computations) :

```
[ 2.5s    ] generate p p > t t~ [QCD]
[ 6.1s    ] output
[ ~ mins*] launch
    * timing for 10k unweighted events on a laptop
```

  - With options specified (in this case, generates the one-loop matrix element code only):

```
[ 0.01s ] import model loop_sm-no_hwidth
[ 0.01s ] set complex_mass_scheme
[ 5min  ] generate g g > e+ ve mu- vm~ b b~ / h QED=2 [virt=QCD]
[ 2min  ] output MyProc
[ ~1 s* ] launch -f
    * time per phase-space point, summed over helicity configurations and colors.
```

Details on how to generate and use a MadLoop standalone library available @

cp3.irmp.ucl.ac.be/projects/madgraph/wiki/MadLoopStandaloneLibrary

MadLoop in MG5aMC

- Process generation
  - `import model <model_name>-<restrictions>`
  - `generate <process> <amp_orders_and_option> [<mode>=<pert_orders>] <squared_orders>`
  - `output <format> <folder_name>`
  - `launch <options>`
- Examples, starting from a default MG5aMC interface
  - Very simple one (in this case, generates the full code for NLO computations) :

```
[ 2.5s    ] generate p p > t t~ [QCD]
[ 6.1s    ] output
[ ~ mins*] launch
    * timing for 10k unweighted events on a laptop
```

  - With options specified (in this case, generates the one-loop matrix element code only):

```
[ 0.01s ] import model loop_sm-no_hwidth
[ 0.01s ] set complex_mass_scheme
[ 5min  ] generate g g > e+ ve mu- vm~ b b~ / h QED=2 [virt=QCD]
[ 2min  ] output MyProc
[ ~1 s* ] launch -f
    * time per phase-space point, summed over helicity configurations and colors.
```

Details on how to generate and use a MadLoop standalone library available @

cp3.irmp.ucl.ac.be/projects/madgraph/wiki/MadLoopStandaloneLibrary