ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Loop Evaluations

The MadGraph way

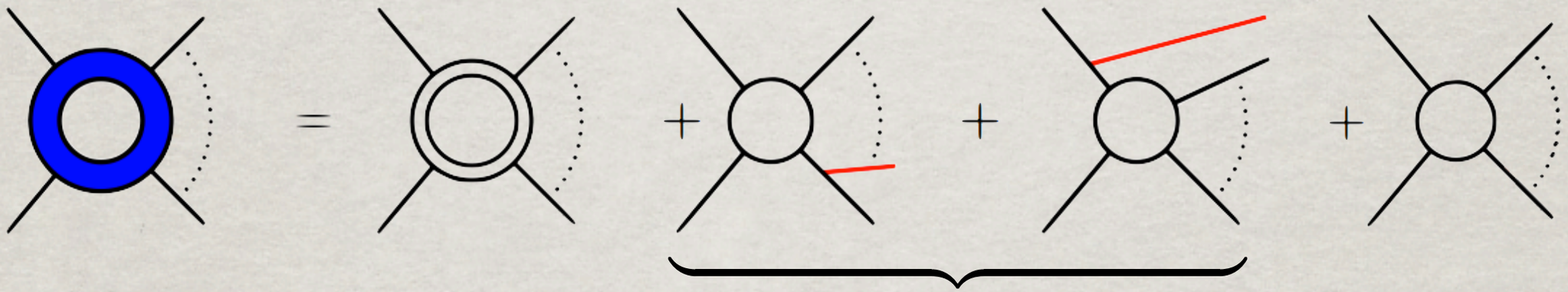
Valentin Hirschi
EPFL

1st October 2012

Lecture
MG/FR School @ Natal

NLO Basics

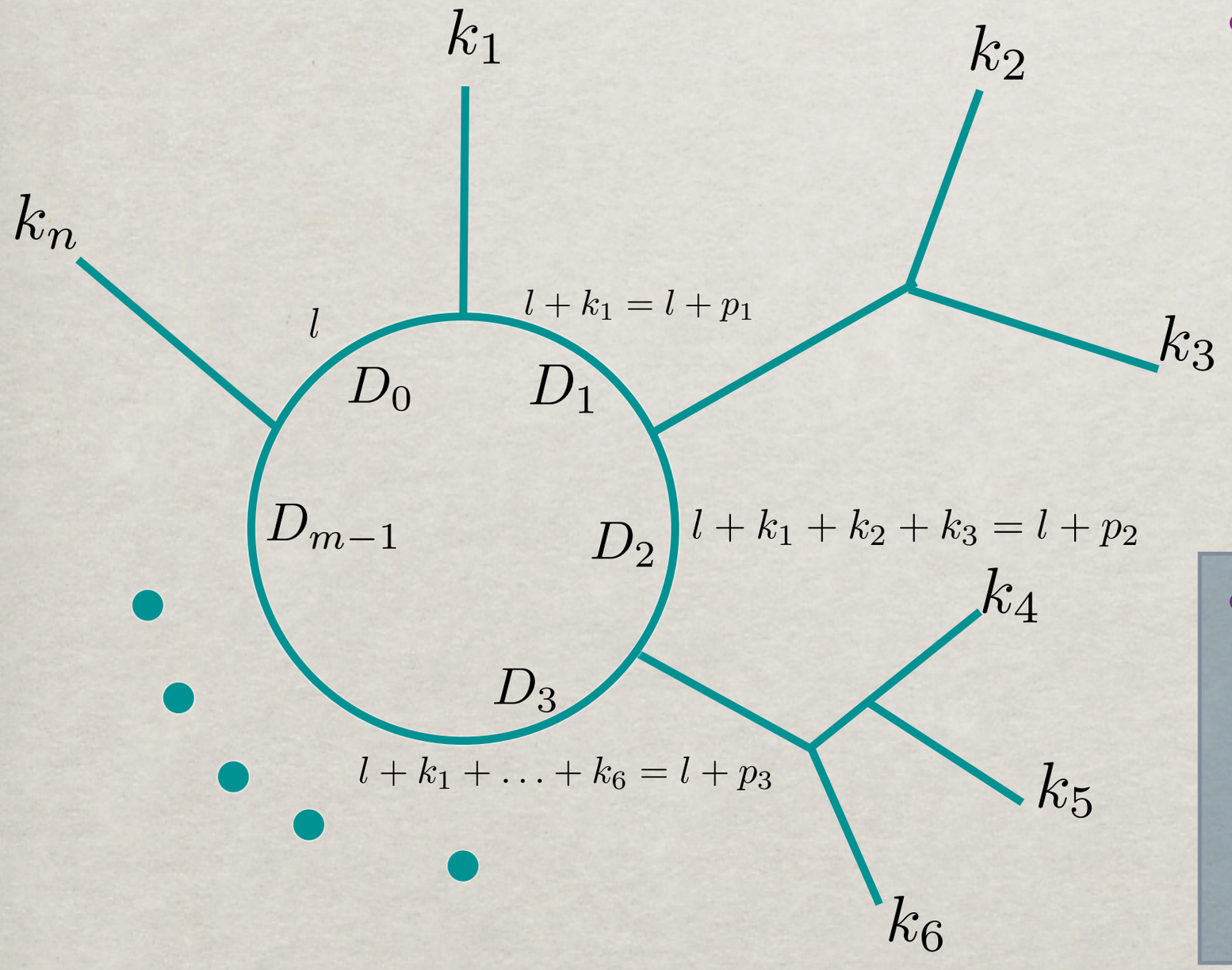
Fixed-order NLO contributions have two parts



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

Virtual part	Real emission part	Born
Today's topic	Rikkert's talk	Solved

ONE-LOOP INTEGRAL



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

- The integral to compute is

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

$$D_i = (l + p_i)^2 - m_i^2$$

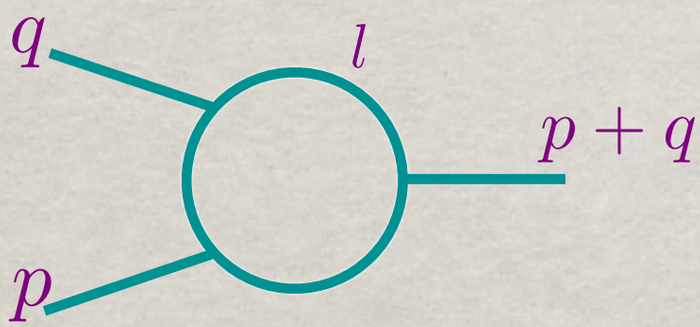
Standard Approach

- 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”

- Let’s do an example:
  Suppose we want to calculate this triangle integral



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

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

- 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

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

- 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

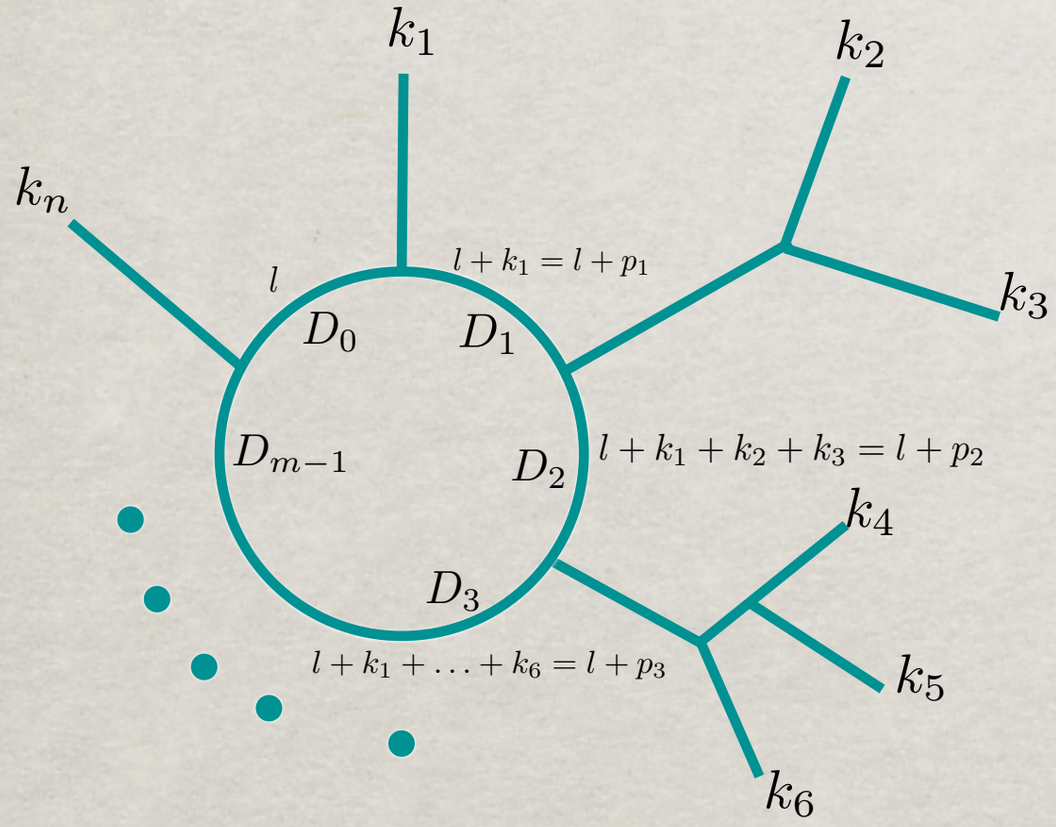
$$\begin{pmatrix} C_1 \\ C_2 \end{pmatrix} = G^{-1} \begin{pmatrix} R_1 \\ R_2 \end{pmatrix}$$

and we have expressed 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 integrals and we are done!

Higher point integrals



- For loop integrals with many legs, the reduction to scalar integrals can still be performed
- Only up to 4-point scalar integrals are needed (in 4 dimensions)!
- The proof is beyond the scope of these lectures (it is straight forward by using the Van Neerven-Vermaseren basis for the loop momentum); it is related to the fact that in 4 dimensions only four 4-vectors can be linearly independent

Basis of scalar integrals

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

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

$$D_i = (l+p_i)^2 - m_i^2$$

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

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

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

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

- All these scalar integrals are known and available in computer libraries (FF [v. Oldenborgh], QCDLoop [Ellis, Zanderighi], OneLOop [v. Hameren])

Divergences

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

$$D_i = (l+p_i)^2 - m_i^2$$

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

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

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

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

- The coefficients d, c, b and a are finite and do not contain poles in $1/\epsilon$
- The $1/\epsilon$ dependence is in the scalar integrals (and the UV renormalization)
- When we have solved this system (and included the UV renormalization) we have the full dependence on the soft/collinear divergences in terms of coefficients in front of the poles. These divergences should cancel against divergences in the real emission corrections (according to KLN theorem)

$$\text{Virtual} \sim v_0 + \frac{v_1}{\epsilon} + \frac{v_2}{\epsilon^2}$$

About the R term

- In our example the decomposition to scalar integrals was "exact", i.e. there were no left-over terms.
- This is true for most integrals. Only if the rank of the integral is

$$r \geq \max\{(N-1), 2\}$$

there are some extra contributions which are called "Rational terms" that are not proportional to a scalar integral

- They are of UV origin and come from the ϵ (dimensional regulator) dependence of the integral times a scalar integral that is UV divergent

$$\text{Rational terms} \sim \epsilon B_0(p, m_1, m_2)$$

(The Bubble scalar integrals are the only UV divergent scalar integrals)

- When taking the limit $\epsilon \to 0$, only the leading contribution remains, which are independent from the scalar integral itself

Automation

- Advantage:
  - The method above can be straight-forwardly generalized to any one-loop integral (appearing in a renormalizable theory)
- Disadvantage:
  - For relatively simple processes, the number of terms already explodes (several 100 MB of code is no exception for the matrix elements of a $2 \to 3$ process); simplifications require hard work and are difficult to do in a general way
  - Does only work when the integrals are known analytically

PV-Reduction chain

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

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

The "NLO revolution"

One indicator of NLO progress

pp → W + 0 jet	1978	Altarelli, Ellis, Martinelli
pp → W + 1 jet	1989	Arnold, Ellis, Reno
pp → W + 2 jets	2002	Campbell, Ellis
pp → W + 3 jets	2009	BH+Sherpa Ellis, Melnikov, Zanderighi
pp → W + 4 jets	2010	BH+Sherpa

Slide from Lance Dixon

New loop techniques

- The "loop revolution": new techniques for computing one-loop matrix elements are now established:
  - Generalized unitarity (e.g. BlackHat, Rocket, ...)
    [Bern, Dixon, Dunbar, Kosower, 1994...; Ellis Giele Kunst 2007 + Melnikov 2008;...]
  - Integrand reduction (e.g. CutTools, GoSam)
    [Ossola, Papadopoulos, Pittau 2006; del Aguila, Pittau 2004; Mastrolia, Ossola, Reiter, Tramontano 2010;...]
  - Tensor reduction (e.g. Golem)
    [Passarino, Veltman 1979; Denner, Dittmaier 2005; Binoth Guillet, Heinrich, Pilon, Reiter 2008]

Integrand reduction

- Any one-loop integral can be decomposed in scalar integrals
- The task is to find these coefficients efficiently (analytically or numerically)
- The integrand (or OPP [Ossola, Papadopoulos, Pittau 2006]) reduction method is a method that has been automated in the **CutTools** program to find these coefficients in an automated way
- The integrand reduction technique is what we have adopted to use in MadGraph to compute the loop diagrams

At the integrand level

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

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

- This is exactly what the OPP reduction does
  - The decomposition is the same, except that there might be contributions that integrate to zero

At the integrand level

- Consider, e.g., the Box coefficient:

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

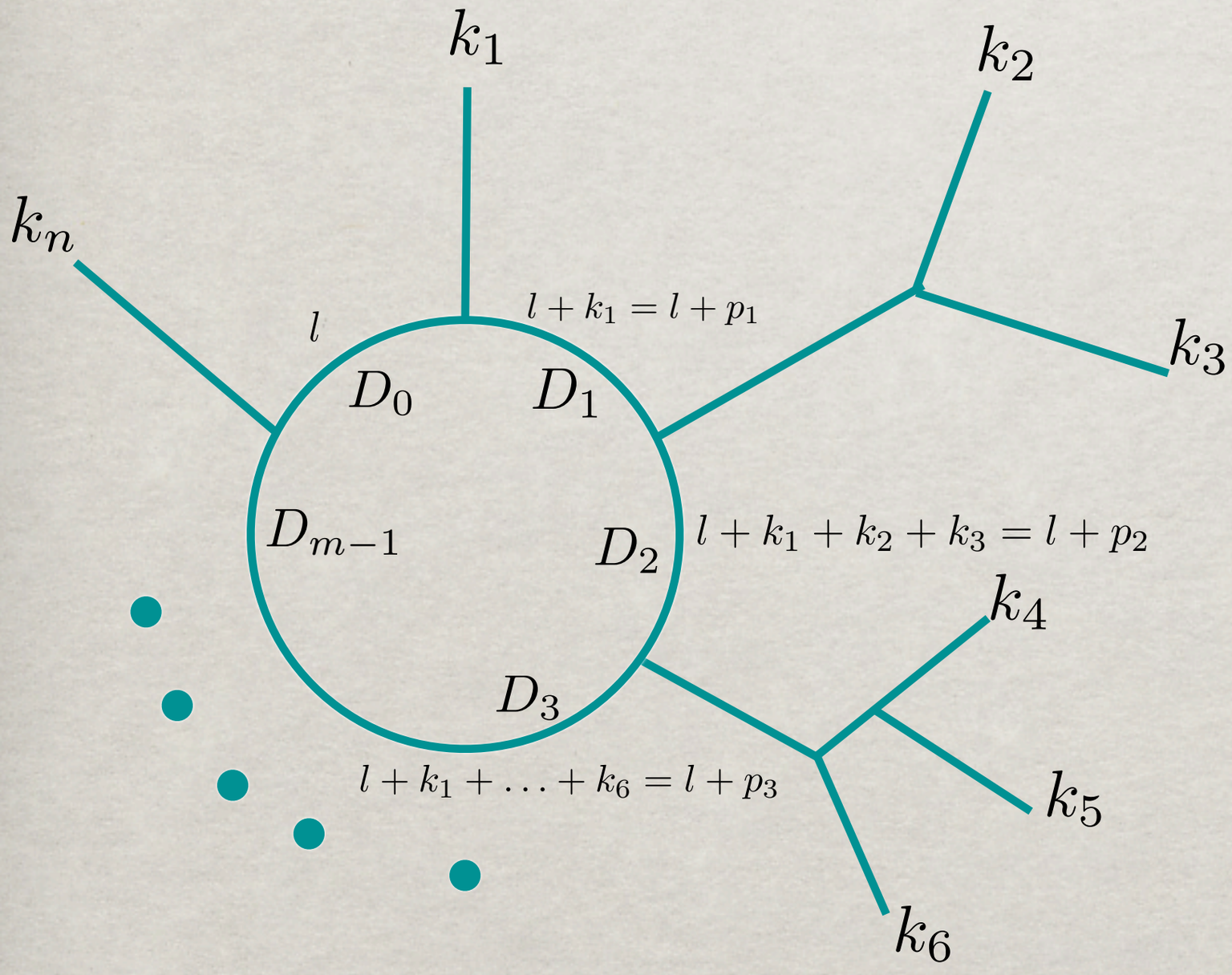
$$= \int d^d l \frac{d_{i_0 i_1 i_2 i_3}}{D_{i_0} D_{i_1} D_{i_2} D_{i_3}}$$

$$= \int d^d l \frac{d_{i_0 i_1 i_2 i_3} + \tilde{d}_{i_0 i_1 i_2 i_3}(l)}{D_{i_0} D_{i_1} D_{i_2} D_{i_3}}$$

where $$\int d^d l \frac{\tilde{d}_{i_0 i_1 i_2 i_3}(l)}{D_{i_0} D_{i_1} D_{i_2} D_{i_3}} = 0$$

- And similarly for the c, b, a and R terms
- The contributions that vanish when doing the integral are called "spurious terms"

ONE-LOOP INTEGRAL



- Consider this m-point loop diagram with n external momenta
- The integral to compute is

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

$$D_i = (l + p_i)^2 - m_i^2$$

OPP DECOMPOSITION

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

$$D_i = (l + p_i)^2 - m_i^2$$

- For the numerator of any integrand of a one-loop computation we can therefore write

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

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

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

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

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

Numerical evaluation

- By choosing specific values for the loop momentum l, we end up with a system of linear equations
  - In a renormalizable theory, the rank of the integrand is always smaller (or equal) to the number of particles in the loop (with a conveniently chosen gauge)
  - We can straight-forwardly set the it up by sampling the numerator numerically for various values of the loop momentum l
  - By choosing l smartly, the system greatly reduces
    - In particular when we chose l to be a complex 4-vector

Functional form of the spurious terms

- 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

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

How it works...

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

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

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

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

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

To solve the OPP reduction, choosing special values for the loop momenta 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

How it works...

$$N(l^{\pm}) = d_{0123} + \tilde{d}_{0123}(l^{\pm}) \prod_{i \neq 0,1,2,3}^{m-1} D_i(l^{\pm})$$

- Two values are enough given the functional form for the spurious term. We can immediately determine the Box coefficient

$$d_{0123} = \frac{1}{2}\left[\frac{N(l^{+})}{\prod_{i\neq 0,1,2,3}^{m-1} D_i(l^{+})} + \frac{N(l^{-})}{\prod_{i\neq 0,1,2,3}^{m-1} D_i(l^{-})}\right]$$

- By choosing other values for l, that set other combinations of 4 "denominators" to zero, we can get all the Box coefficients

How it works...

- Now that we have all the Box coefficients we can start choosing values for *l* that set 3 "denominators" to zero to get the Triangle coefficients. Of course, now both the first and the second lines contribute.

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

- We already have solved the coefficients of the first line in the previous iteration, so also here there is only a simple system of equations to solve
- Once we have all the Triangle coefficients, we can continue to determine the Bubble coefficients; and finally the Tadpole coefficients

How it works...

- For each phase-space point we have to solve the system of equations
- Due to the fact that the system reduces when picking special values for the loop momentum, the system greatly reduces
- We can decompose the system at the level of the squared matrix element, amplitude, diagram or anywhere in between. As long as we provide the corresponding numerator function. Since each reduction with CutTools is computationally heavy, we directly reduce the squared element with MadGraph.
- For a given phase-space point, we have to compute the numerator function several times (~50 or so for a box loop)

A classical example

- Suppose we want to compute this integral $\int d^d l \frac{1}{D_0 D_1 D_2 D_3 D_4 D_5 D_6}$
- So we that the numerator is $N(l) = 1$ $\qquad D_i = (l + p_i)^2 - m_i^2$
- We know that we need only Box, Triangle, Bubble (and Tadpole) contributions. Let's find the first Box integral coefficient.

- Take the two solutions of

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

- And use the relation we found before and we directly have

$$d_{0123} = \frac{1}{2}\left[\frac{1}{D_4(l^+)D_5(l^+)D_6(l^+)} + \frac{1}{D_4(l^-)D_5(l^-)D_6(l^-)}\right]$$

$\ldots$

Complications in d dimensions

- In the previous consideration I was very sloppy in considering if we are working in 4 or d dimensions
- In general, external momenta and polarization vectors are in 4 dimensions; only the loop momentum is in d dimensions

- To be more correct, we compute the integral

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

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

d dim 4 dim epsilon dim

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

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

Implications

$$\sum_{0\leq i_0<i_1<i_2<i_3}^{m-1} d(i_0i_1i_2i_3)\int d^d\bar{\ell}\,\frac{1}{\bar{D}_{i_0}\bar{D}_{i_1}\bar{D}_{i_2}\bar{D}_{i_3}}$$

$$+\sum_{0\leq i_0<i_1<i_2}^{m-1} c(i_0i_1i_2)\int d^d\bar{\ell}\,\frac{1}{\bar{D}_{i_0}\bar{D}_{i_1}\bar{D}_{i_2}}$$

$$+\sum_{0\leq i_0<i_1}^{m-1} b(i_0i_1)\int d^d\bar{\ell}\,\frac{1}{\bar{D}_{i_0}\bar{D}_{i_1}}$$

$$+\sum_{i_0=0}^{m-1} a(i_0)\int d^d\bar{\ell}\,\frac{1}{\bar{D}_{i_0}}$$

$$+R\,.$$

- The decomposition in terms of scalar integrals has to be done in d dimensions
- This is why the rational part R is needed

Rational terms

- The main difference is how we get the rational terms (we already saw them in the Passarino-Veltman reduction)
- In the OPP method, they are split into two contributions, generally called

$$R = R_1 + R_2$$

- Both have their origin in the UV part of the model, but only R_1 can be directly computed in the OPP reduction and is given by the CutTools program

R_1

- The origin of R_1 is coming is the denominators of the propagators in the loop

$$\frac{1}{D_i} \to \frac{1}{\bar{D}_i} = \frac{1}{D}\left(1 - \frac{\tilde{l}^2}{D_i}\right)$$

- Of course, the propagator structure is known, so these contributions can be included in the OPP reduction

- They give contributions proportional to

$$\int d^d\bar{l} \frac{\tilde{l}^2}{\bar{D}_i \bar{D}_j} = -\frac{i\pi^2}{2}\left[m_i^2 + m_j^2 - \frac{(p_i - p_j)^2}{3}\right] + \mathcal{O}(\epsilon)$$

$$\int d^d\bar{l} \frac{\tilde{l}^2}{\bar{D}_i \bar{D}_j \bar{D}_k} = -\frac{i\pi^2}{2} + \mathcal{O}(\epsilon)$$

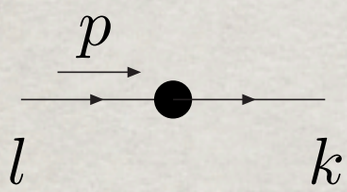
$$\int d^d\bar{l} \frac{\tilde{l}^4}{\bar{D}_i \bar{D}_j \bar{D}_k \bar{D}_l} = -\frac{i\pi^2}{6} + \mathcal{O}(\epsilon)$$

R_2

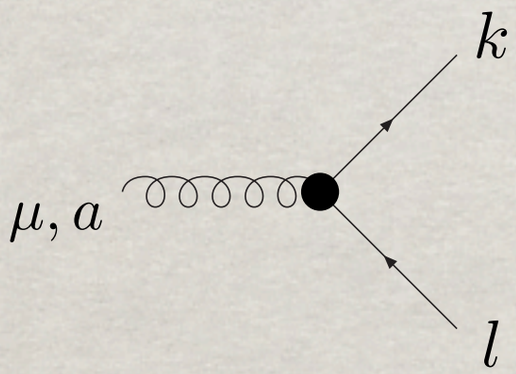
- The other origin of rational terms is the numerator itself. For integrals with rank > 2 we can have dependence in the numerator that is proportional to $\bar{l}^2$
- Unfortunately, this dependence can be quite hidden; maybe it is only explicitly there after doing the Clifford algebra
- Because we want to solve the system without doing this algebra analytically (we want to solve it numerically) we cannot get these contributions directly within the OPP reduction
- Within a given model, there is only a finite number of sources that can give these contributions; They have all been identified within the SM, and can be computed with the "R_2 counter terms"

R_2 Feynman rules

- Given that the R_2 contributions are of UV origin, only up to 4-point functions contribute to it (in a renormalizable theory)
- They can be computed using special Feynman rules, similarly to the UV counter term Feynman rules needed for the UV renormalization, e.g.



$$= \frac{ig^2}{16\pi^2}\frac{N_{col}^2-1}{2N_{col}}\delta_{kl}(-\not{p}+2m_q)\,\lambda_{HV}$$

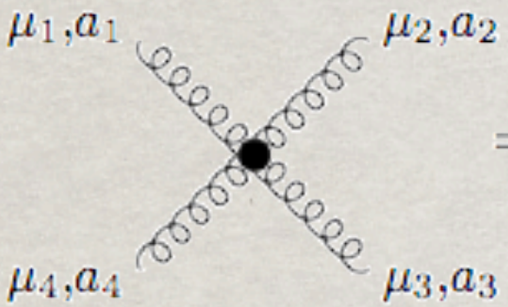


$$= \frac{ig^3}{16\pi^2}\frac{N_{col}^2-1}{2N_{col}}\,t^a_{kl}\gamma_\mu\,(1+\lambda_{HV})$$

[Draggiotis, Garzelli, Papadopoulos, Pittau]

- Unfortunately these Feynman rules are model dependent.
  ⇒ Maybe we can use FeynRules+FeynArts to compute them for any model?

R_2 Feynman rules



$$= -\frac{ig^4 N_{col}}{96\pi^2} \sum_{P(234)} \left\{ \left[\frac{\delta_{a_1 a_2}\delta_{a_3 a_4} + \delta_{a_1 a_3}\delta_{a_4 a_2} + \delta_{a_1 a_4}\delta_{a_2 a_3}}{N_{col}} \right.\right.$$

$$+ 4\,Tr(t^{a_1}t^{a_3}t^{a_2}t^{a_4} + t^{a_1}t^{a_4}t^{a_2}t^{a_3})\,(3 + \lambda_{HV})$$

$$\left. - Tr(\{t^{a_1}t^{a_2}\}\{t^{a_3}t^{a_4}\})\,(5 + 2\lambda_{HV}) \right] g_{\mu_1\mu_2} g_{\mu_3\mu_4}$$

$$+ 12 \frac{N_f}{N_{col}} Tr(t^{a_1}t^{a_2}t^{a_3}t^{a_4}) \left(\frac{5}{3} g_{\mu_1\mu_3} g_{\mu_2\mu_4} - g_{\mu_1\mu_2} g_{\mu_3\mu_4} - g_{\mu_2\mu_3} g_{\mu_1\mu_4} \right) \Bigg\}$$

- Not always simple, we implemented them for SM, but in general for BSM, automation is needed.

Summary: Beyond Passarino-Veltman

- In PV reduction, we need analytic expressions for all the integrals. Possible to automate, but in practice too many terms which are difficult to simplify ☺☹
- In OPP reduction we reduce the system at the **integrand** level.
  - We can solve the system numerically: we only need a numerical function of the (numerator of) integrand. We can set-up a system of linear equations by choosing specific values for the loop momentum l, depending on the kinematics of the event ☺
  - OPP reduction is implemented in CutTools (publicly available). Given the integrand, CutTools provides all the coefficients in front of the scalar integrals and the R_1 term ☺
  - The OPP reduction leads to numerical unstabilities whose origins are not well under control. The reach of OPP in double precision is unclear. ☹☺
  - Analytic information is needed for the R_2 term, but can be compute once and for all for a given model ☹☺

The MadGraph Solution

- MadGraph good at giving numerical expressions for matrix elements. Exactly what is needed by CutTools to get the coefficients of the scalar integrals.

  ➥ However, it is only tree-level...

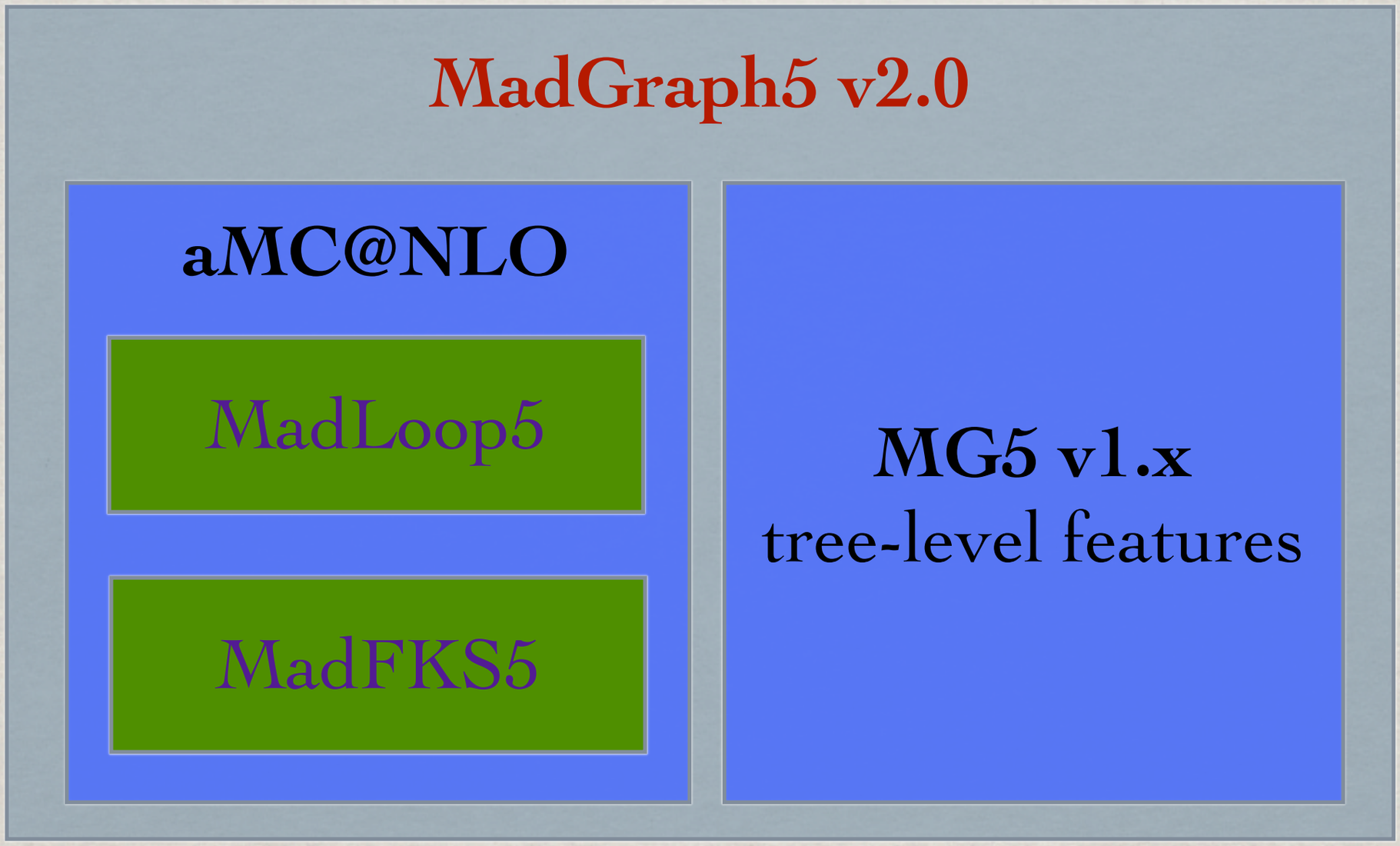
- Need to upgrade MadGraph so to generate loop diagrams and numerical code for the integrand N(q):

MadLoop

MadLoop5

Virtuals for aMC@NLO

Nomenclature

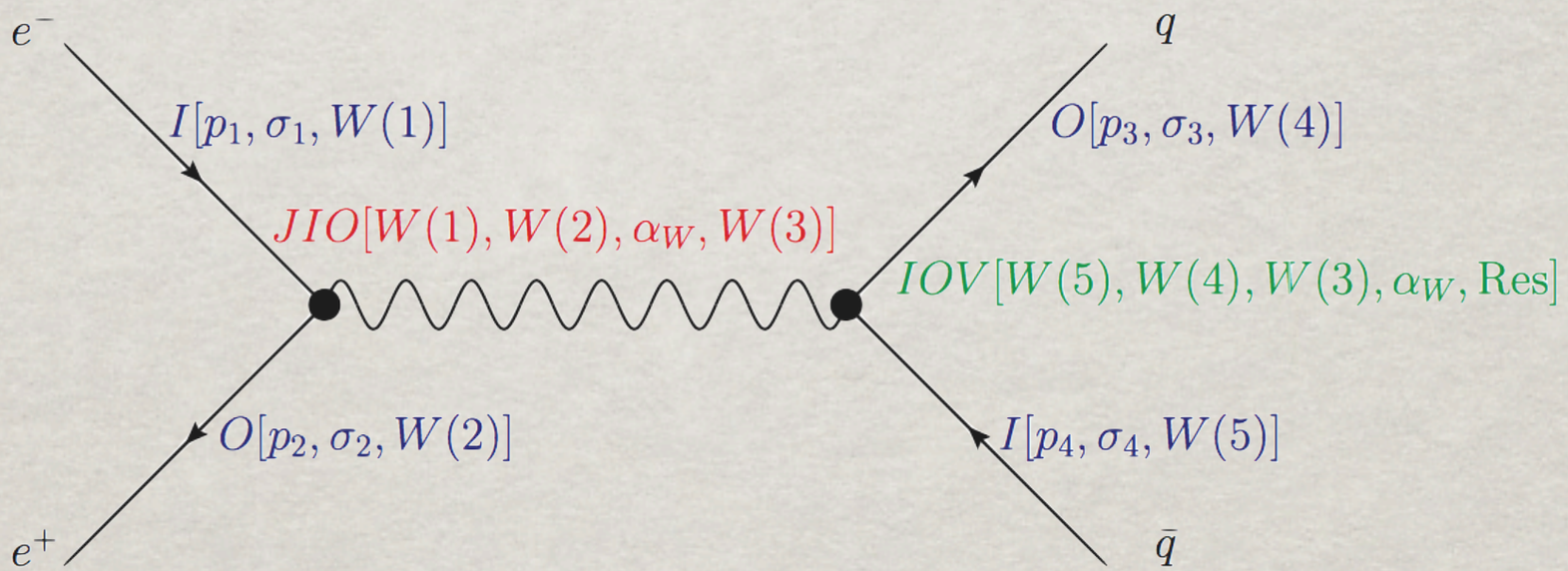


But this separation is now transparent to the users!

A BASIC REMINDER

The evolutive way of computing tree-diagrams

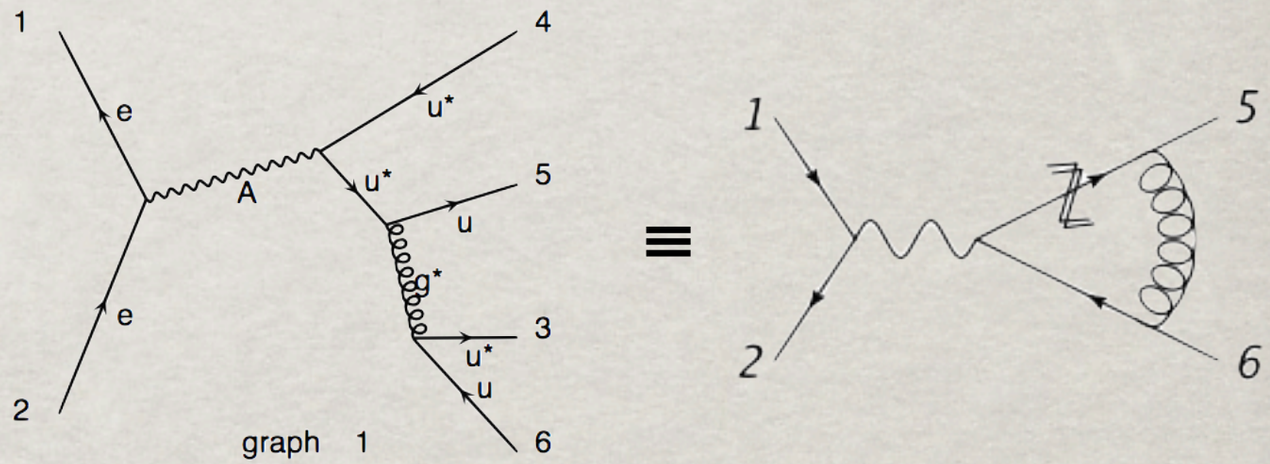
- First generates all tree-level Feynman Diagrams
- Compute the amplitude of each diagram using a chain of calls to HELAS subroutines



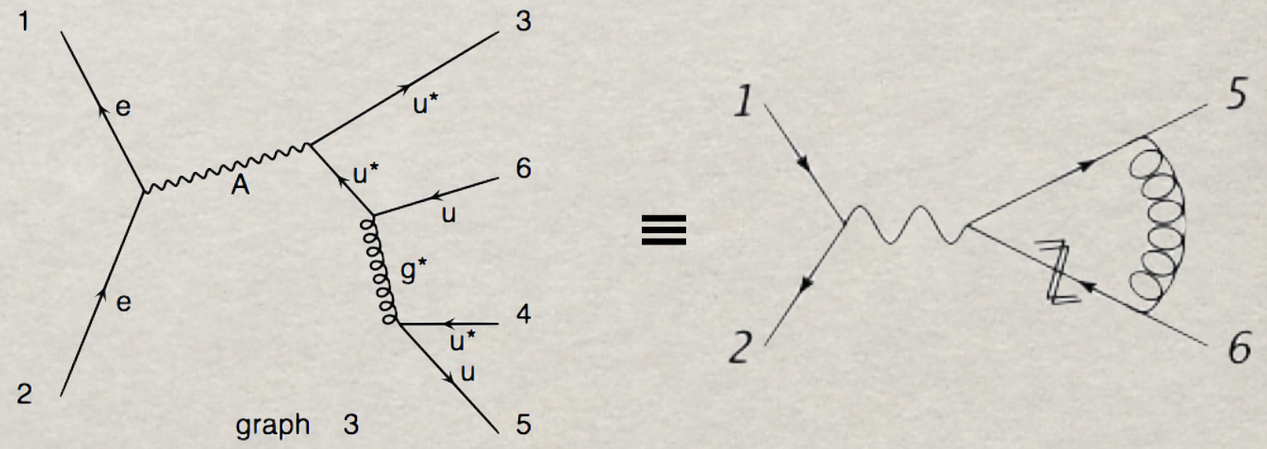
- Finally square all the related amplitude with their right color factors to construct the full LO amplitude

Generating loop diagrams

- Instead of using an external tool for loop diagram generation, we recycle MadGraph5 power for tree level diagram generation.
- A loop diagrams with the loop cut open has to extra external particles. Consider $e^+e^- \rightarrow u\ u\sim\ u\ u\sim$ (loop particles are in green). MadGraph will generate 8 L-cut diagrams. Here are two of them:
  - All diagrams with two extra particles are generated and the ones that are needed are filtered out
  - Each diagram gets an unique tag: any mirror and/or cyclic permutations of tags of diagrams already in the set are taken out
  - Additional filter to eliminate tadpoles and bubbles attached to external lines



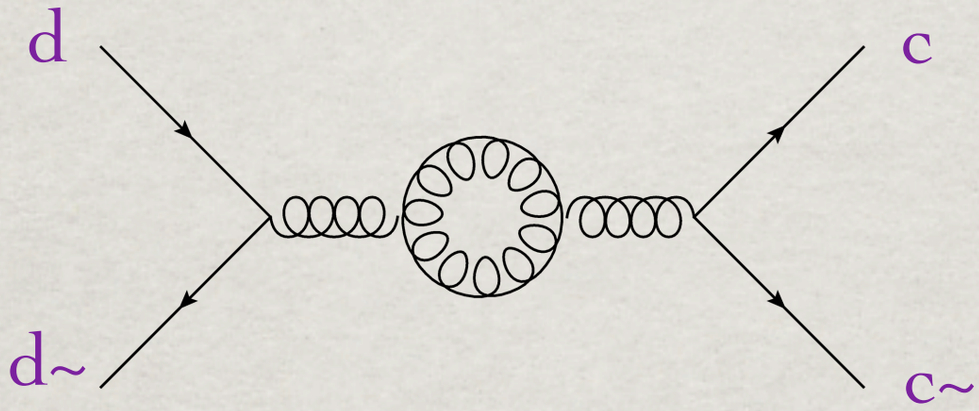
$$\text{Diag}_1 = [u^*(6)g^*(5)u^*(A)]$$



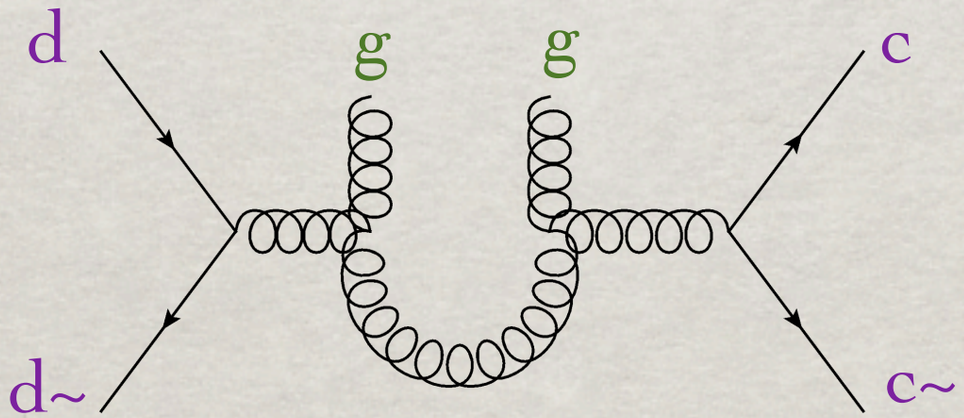
$$\text{Diag}_3 = [u^*(A)u^*(6)g^*(5)]$$

MadLoop

- It is clear though that d d~ → c c~ u u~ will not get you this loop :



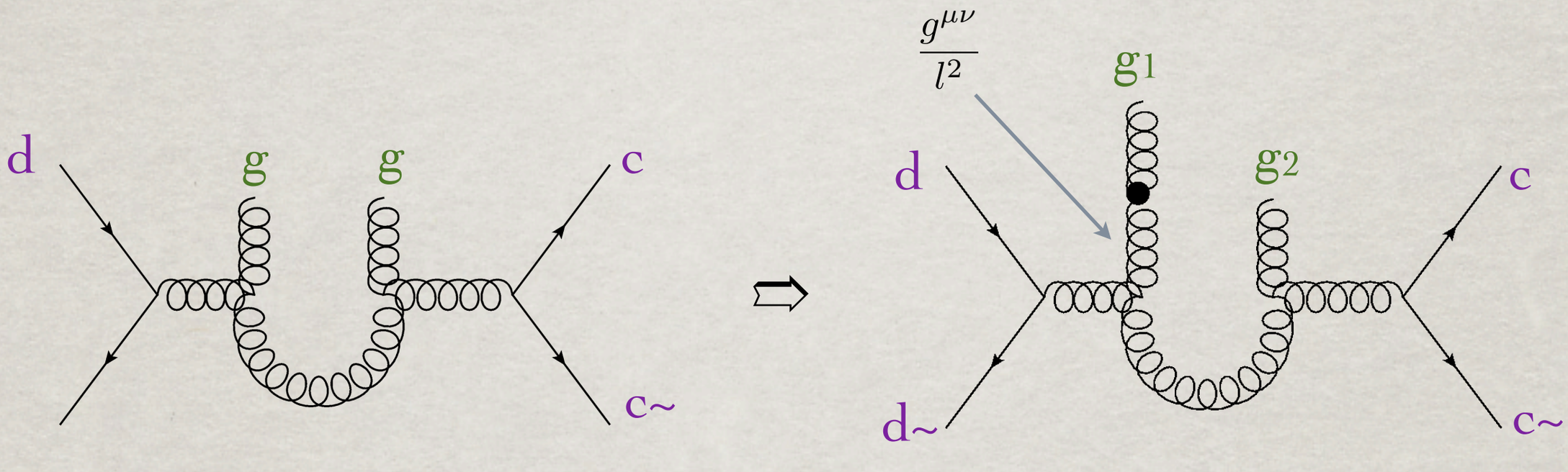
- For this one you necessarily need to generate the born process with the additional two L-cut particles being gluons!



- Loops including a u-quark were already generated with d d~ → c c~ u u~, so you can speed up the d d~ → c c~ g g generation forbidding u in the loop!

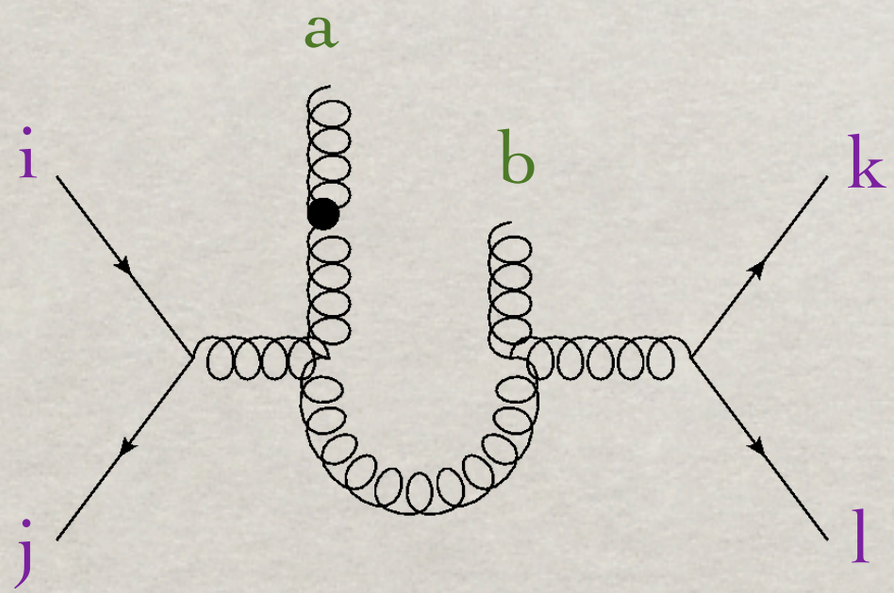
MadLoop

- It is not yet what we want, we are missing the l-cut propagator



d~

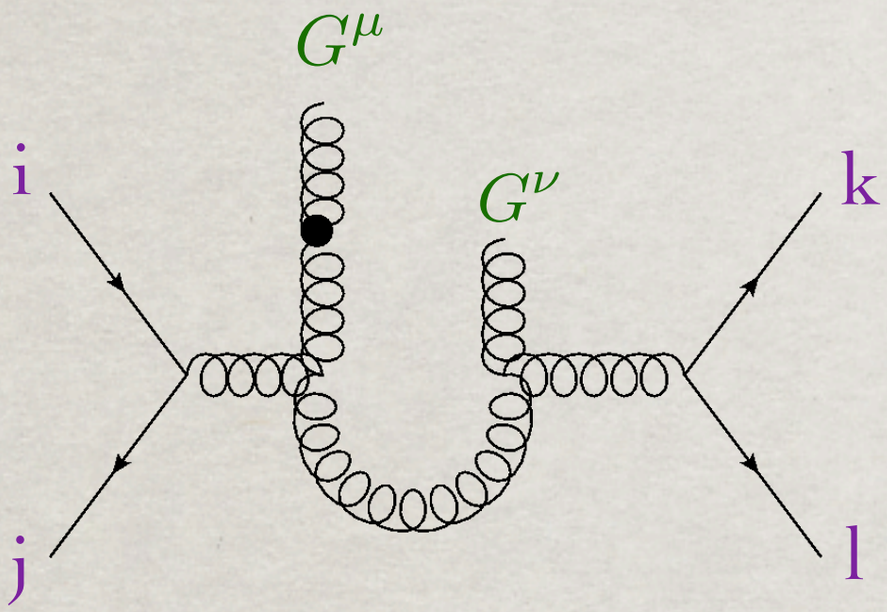
- Also close the color trace → insert a δ^{ab} or δ^{ij} to the color chain



$$\Rightarrow \quad T^r_{ij} f^{ars} f^{sbt} T^t_{kl} \delta^{ab}$$

MadLoop

- Closing the lorentz trace :



$$\Rightarrow \quad \delta^{\mu\nu} = \sum_{i=0}^{4} \underbrace{\delta^{\mu i}}_{G^\mu} \underbrace{\delta^{i\nu}}_{G^\nu}$$

- Two other modifications :
  - ➥ Allow for the loop momentum to be complex
  - ➥ Remove the denominator of the loop propagators

- Ok, now this gives you $\mathcal{N}(l^\mu)$, the integrand numerator to be fed to CT!

RESULTS

- Errors are the MC integration uncertainty only
- Cuts on jets, γ*/Z decay products and photons, but no cuts on b quarks (their mass regulates the IR singularities)
- Efficient handling of exceptional phase-space points: their uncertainty always at least two orders of magnitude smaller than the integration uncertainty
- Running time: two weeks on ~150 node cluster leading to rather small integration uncertainties
- MadFKS+MadLoop results are fully differential in the final states (but only parton-level)

	Process	μ	n_{lf}	Cross section (pb) LO	Cross section (pb) NLO
a.1	$pp \to t\bar{t}$	m_{top}	5	123.76 ± 0.05	162.08 ± 0.12
a.2	$pp \to tj$	m_{top}	5	34.78 ± 0.03	41.03 ± 0.07
a.3	$pp \to tjj$	m_{top}	5	11.851 ± 0.006	13.71 ± 0.02
a.4	$pp \to t\bar{b}j$	$m_{top}/4$	4	25.62 ± 0.01	30.96 ± 0.06
a.5	$pp \to t\bar{b}jj$	$m_{top}/4$	4	8.195 ± 0.002	8.91 ± 0.01
b.1	$pp \to (W^+ \to)e^+\nu_e$	m_W	5	5072.5 ± 2.9	6146.2 ± 9.8
b.2	$pp \to (W^+ \to)e^+\nu_e\, j$	m_W	5	828.4 ± 0.8	1065.3 ± 1.8
b.3	$pp \to (W^+ \to)e^+\nu_e\, jj$	m_W	5	298.8 ± 0.4	300.3 ± 0.6
b.4	$pp \to (\gamma^*/Z \to)e^+e^-$	m_Z	5	1007.0 ± 0.1	1170.0 ± 2.4
b.5	$pp \to (\gamma^*/Z \to)e^+e^-\, j$	m_Z	5	156.11 ± 0.03	203.0 ± 0.2
b.6	$pp \to (\gamma^*/Z \to)e^+e^-\, jj$	m_Z	5	54.24 ± 0.02	56.69 ± 0.07
c.1	$pp \to (W^+ \to)e^+\nu_e b\bar{b}$	$m_W + 2m_b$	4	11.557 ± 0.005	22.95 ± 0.07
c.2	$pp \to (W^+ \to)e^+\nu_e t\bar{t}$	$m_W + 2m_{top}$	5	0.009415 ± 0.000003	0.01159 ± 0.00001
c.3	$pp \to (\gamma^*/Z \to)e^+e^- b\bar{b}$	$m_Z + 2m_b$	4	9.459 ± 0.004	15.31 ± 0.03
c.4	$pp \to (\gamma^*/Z \to)e^+e^- t\bar{t}$	$m_Z + 2m_{top}$	5	0.0035131 ± 0.0000004	0.004876 ± 0.000002
c.5	$pp \to \gamma t\bar{t}$	$2m_{top}$	5	0.2906 ± 0.0001	0.4169 ± 0.0003
d.1	$pp \to W^+W^-$	$2m_W$	4	29.976 ± 0.004	43.92 ± 0.03
d.2	$pp \to W^+W^-\, j$	$2m_W$	4	11.613 ± 0.002	15.174 ± 0.008
d.3	$pp \to W^+W^+\, jj$	$2m_W$	4	0.07048 ± 0.00004	0.1377 ± 0.0005
e.1	$pp \to HW^+$	$m_W + m_H$	5	0.3428 ± 0.0003	0.4455 ± 0.0003
e.2	$pp \to HW^+\, j$	$m_W + m_H$	5	0.1223 ± 0.0001	0.1501 ± 0.0002
e.3	$pp \to HZ$	$m_Z + m_H$	5	0.2781 ± 0.0001	0.3659 ± 0.0002
e.4	$pp \to HZ\, j$	$m_Z + m_H$	5	0.0988 ± 0.0001	0.1237 ± 0.0001
e.5	$pp \to Ht\bar{t}$	$m_{top} + m_H$	5	0.08896 ± 0.00001	0.09869 ± 0.00003
e.6	$pp \to Hb\bar{b}$	$m_b + m_H$	4	0.16510 ± 0.00009	0.2099 ± 0.0006
e.7	$pp \to Hjj$	m_H	5	1.104 ± 0.002	1.036 ± 0.002

Open-loops

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

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

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

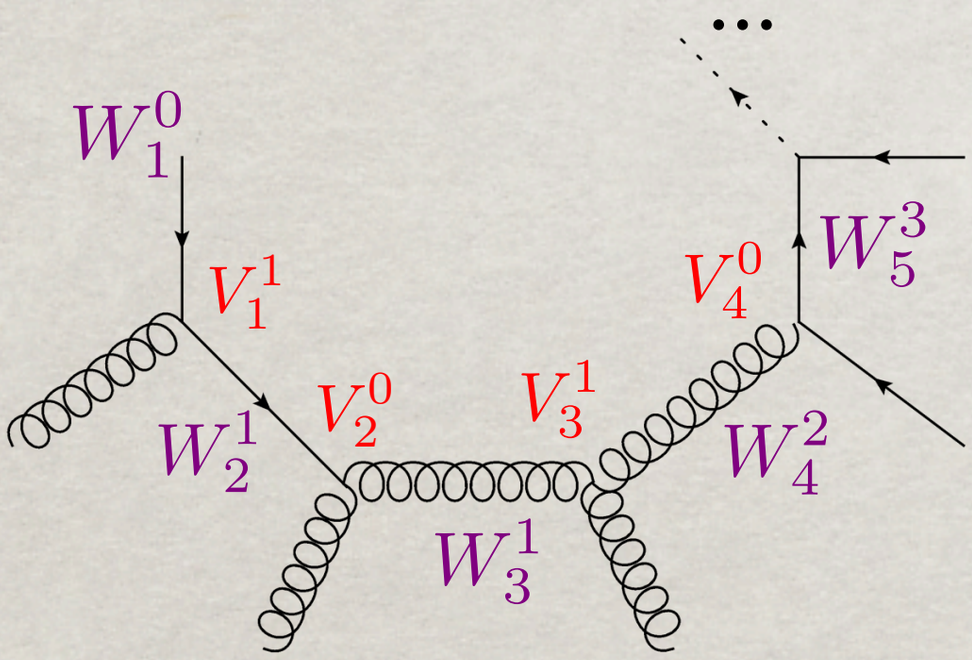
Open-loops

[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

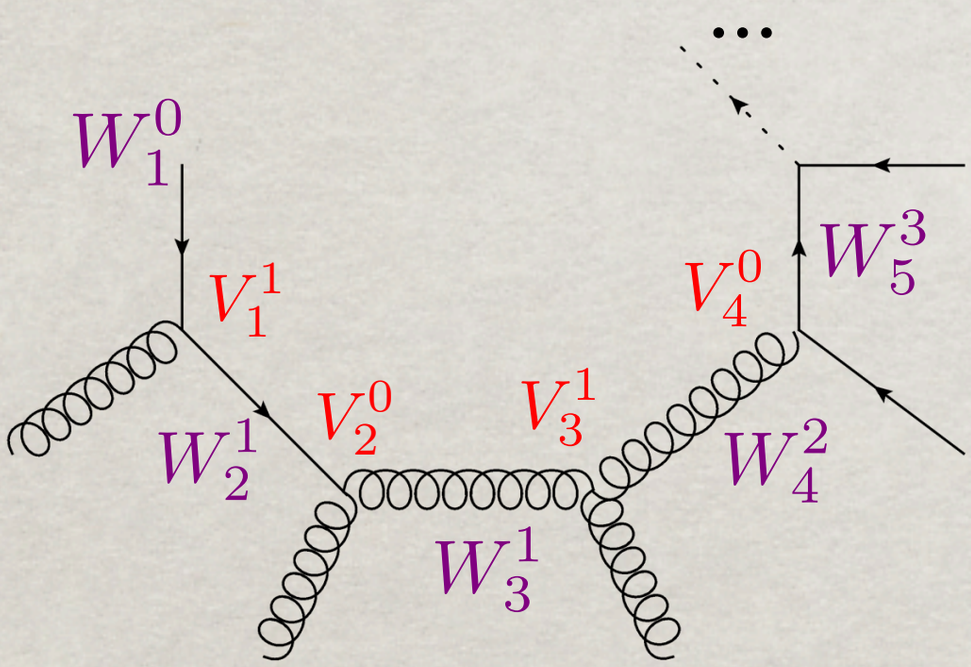
[S. Pozzorini & al. hep-ph/1111.5206]

- 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

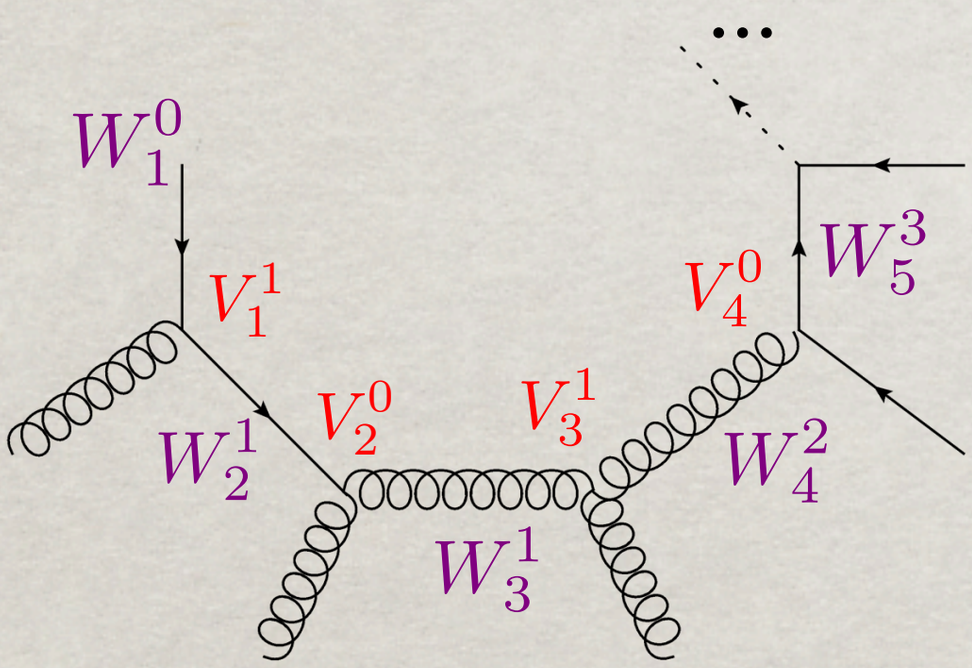
[S. Pozzorini & al. hep-ph/1111.5206]

- 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

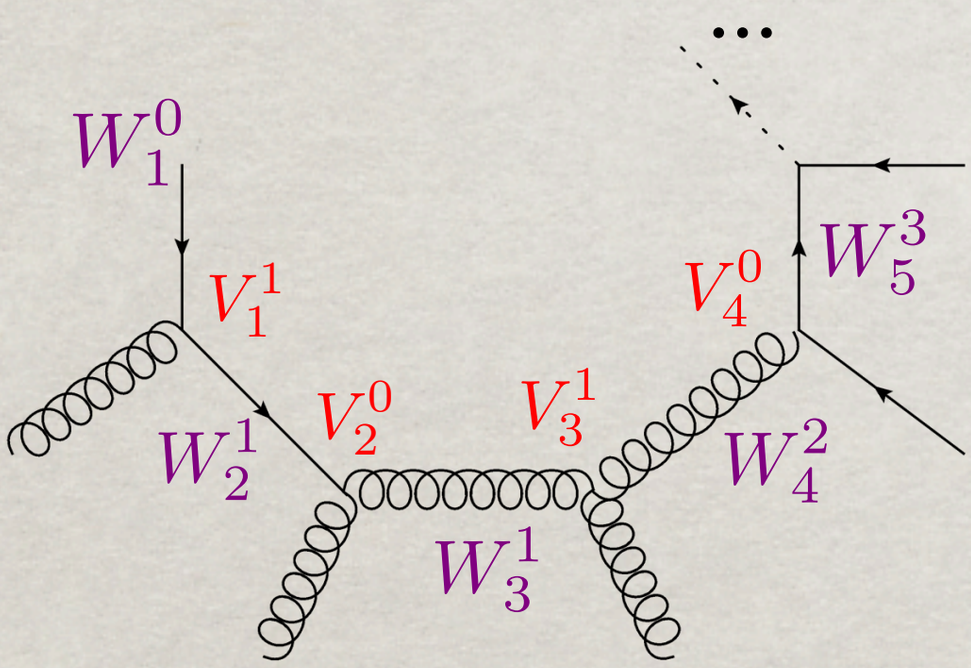
[S. Pozzorini & al. hep-ph/1111.5206]

- 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)

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



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

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

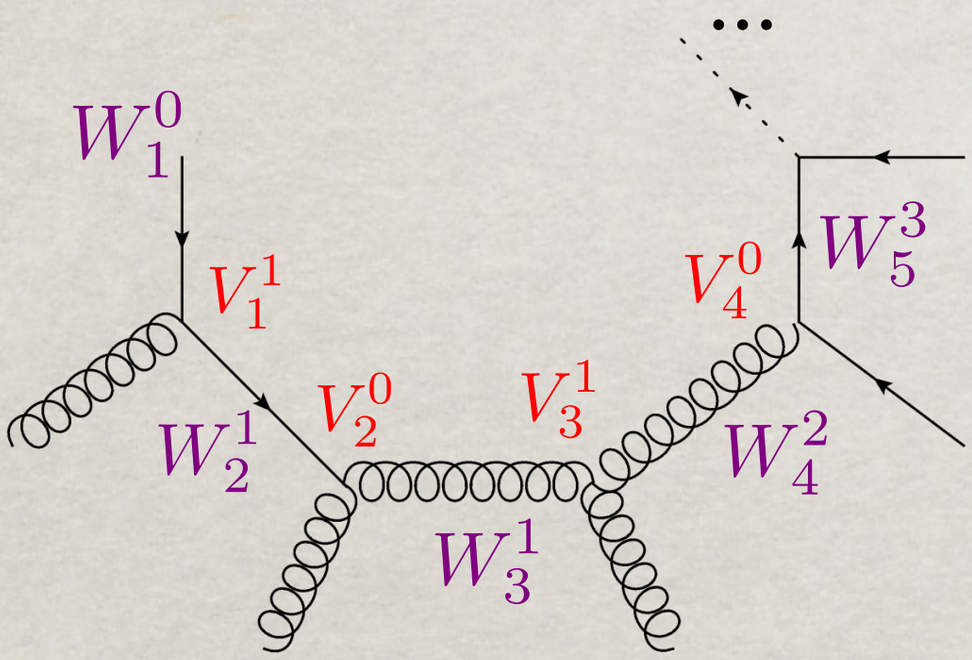
Open-loops

[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)



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

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

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

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

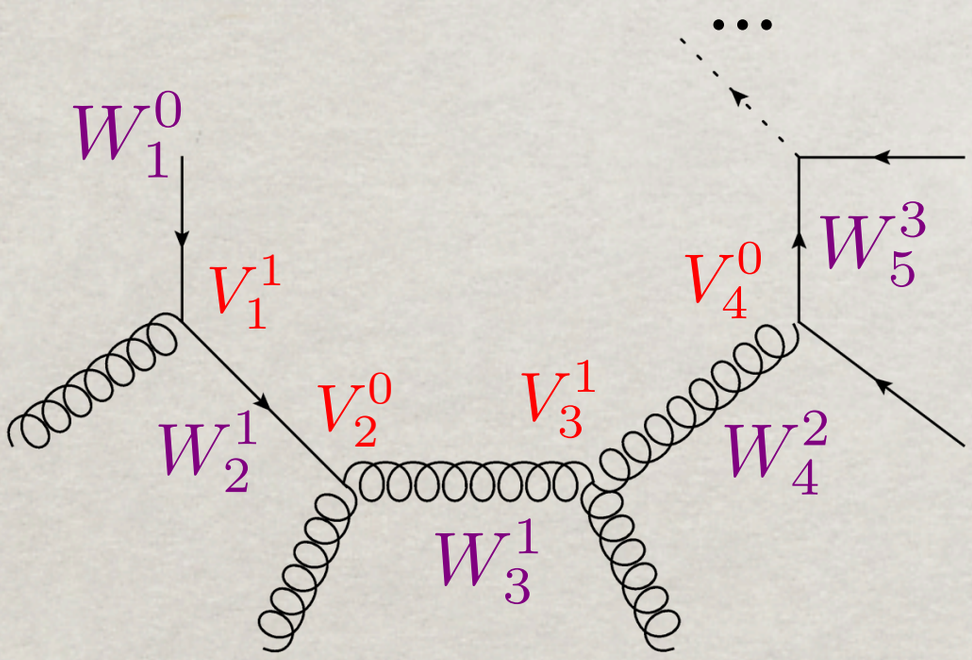
Open-loops

[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)



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

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

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

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

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

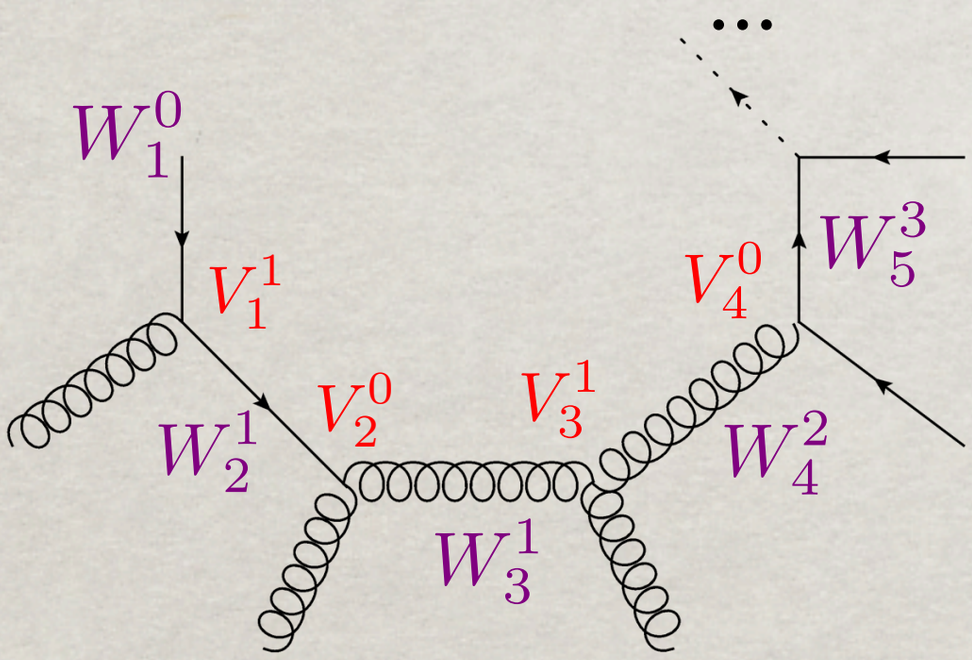
Open-loops

[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 $C^{(2)} = v_3^1 v_2^0 v_1^1 w_1^0, C^{(1)} = v_2^0 w_1^0 (v_3^1 v_0^1 + v_3^0 v_1^1), C^0 = \cdots$

Optimizations

- Summing over helicities first, then reducing the matrix element squared.

$$\mathcal{M} = \sum_{l=loop} 2\Re\Big(\sum_{h=hel} \underbrace{\mathrm{CT}\Big[\int \frac{d^D q \mathcal{N}_{l,h}}{D_0 D_1 \cdots D_{n-1}}\Big]}_{\mathcal{A}_l} \mathcal{A}_h^* \Big) \quad \Rightarrow \quad \mathcal{M} = \sum_{l=loop} 2\Re\Big(\mathrm{CT}\Big[\int d^D q \frac{\sum_{h=hel}\sum_{b=born} \mathcal{N}_{l,h}\mathcal{A}_{b,h}^*}{D_0 D_1 \cdots D_{n-1}}\Big]\Big)$$

  - ➥ Result: Number of OPP calls decreases from N_{loops} x N_{hels} to $N_{loop\_topology}$!

    Also grouping together diagrams with the same denominator structures.

- Exploit the open-loops[F.Cascioli,P.Maierhöfer,S.Pozzorini] technology.
  - ➥ Faster numerator evaluations.
  - ➥ Optimal recycling of the loop wavefunctions.
  - ➥ Remains flexible as ALOHA outputs the building blocks [Work by O.Mattelaer].

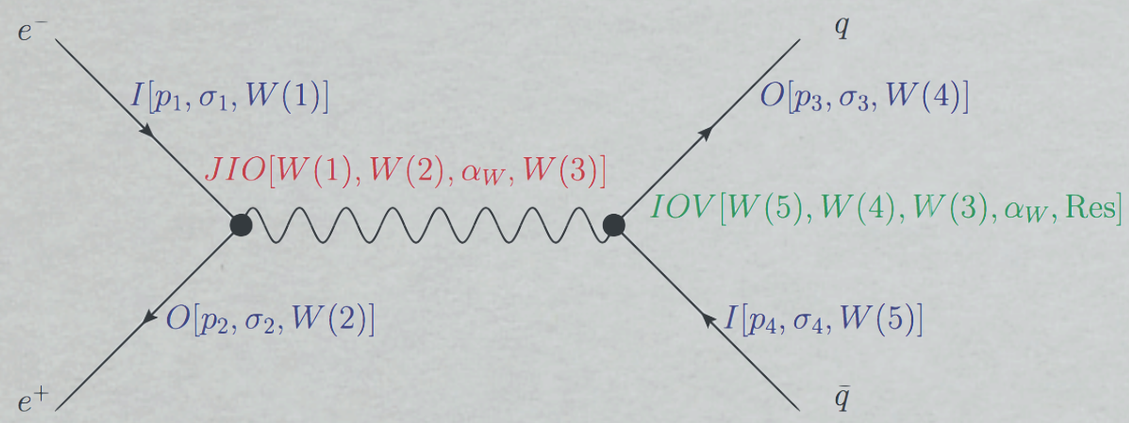
- Automatically numerically detect zero and CP-dependent helicity configurations.

- Efficient reconstruction of the missing L-cut propagator.

Overall speedup of a factor 10+ w.r.t *MLA*

Further Optimizations

- Recycling wavefunction accross helicity configurations

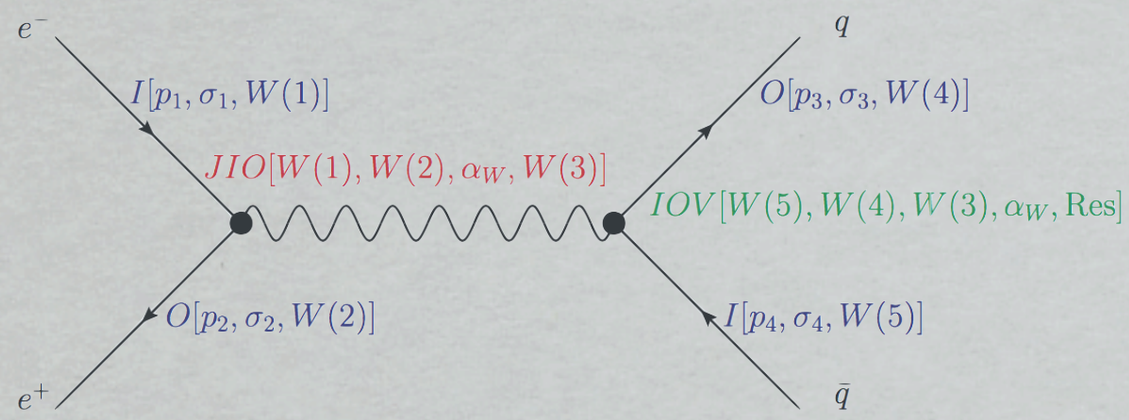


Ex. The same `JIO[e+,e-]` can be used for the two helicity configs of `q q~`

Thanks to open-loops, the loop wavefunctions can also be recycled.

Further Optimizations

- Recycling wavefunction accross helicity configurations



Ex. The same `JIO[e+,e-]` can be used for the two helicity configs of `q q~`

Thanks to open-loops, the loop wavefunctions can also be recycled.

- Grouping diagrams with similar denominator structures

$$\int d^D q \frac{\mathcal{N}_A(q)}{\bar{D}_1 \bar{D}_{12} \bar{D}_{123} \bar{D}_{1234}} + \int d^D q \frac{\mathcal{N}_B(q)}{\bar{D}_1 \bar{D}_{12} \bar{D}_{1234}}$$

$$= \int d^D q \frac{\mathcal{N}_A(q) + \mathcal{N}_B(q) D_{123}}{\bar{D}_1 \bar{D}_{12} \bar{D}_{123} \bar{D}_{1234}}$$

A given triangle and its corresponding box can be reduced at once!

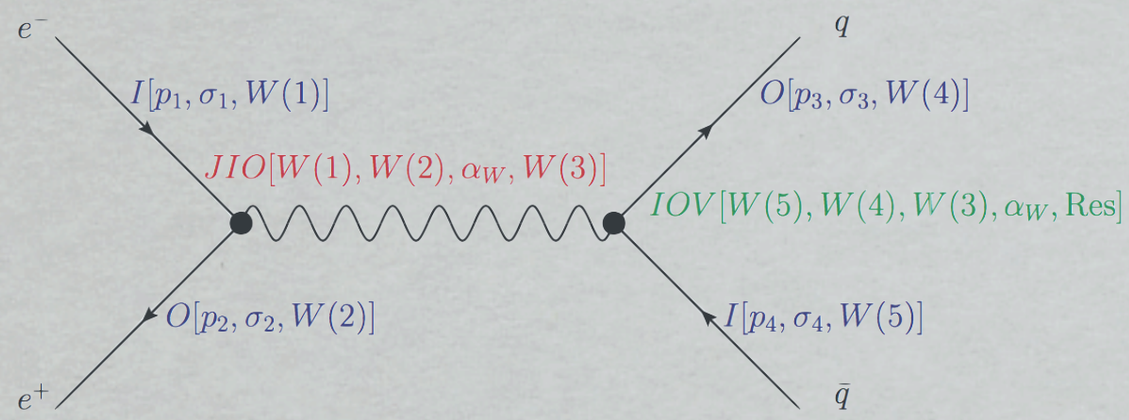
Ex: `g g > g g` would require only six calls to OPP, one per box topology!

But tedious book-keeping and also needs care with dimensionality.

Only useful if dominated by OPP!

Further Optimizations

- Recycling wavefunction accross helicity configurations



Ex. The same `JIO[e+,e-]` can be used for the two helicity configs of `q q~`

Thanks to open-loops, the loop wavefunctions can also be recycled.

- Grouping diagrams with similar denominator structures

$$\int d^D q \frac{\mathcal{N}_A(q)}{\bar{D}_1 \bar{D}_{12} \bar{D}_{123} \bar{D}_{1234}} + \int d^D q \frac{\mathcal{N}_B(q)}{\bar{D}_1 \bar{D}_{12} \bar{D}_{1234}}$$

$$= \int d^D q \frac{\mathcal{N}_A(q) + \mathcal{N}_B(q) D_{123}}{\bar{D}_1 \bar{D}_{12} \bar{D}_{123} \bar{D}_{1234}}$$

A given triangle and its corresponding box can be reduced at once!

Ex: g g > g g would require only six calls to OPP, one per box topology!

But tedious book-keeping and also needs care with dimensionality.

Only useful if dominated by OPP!

- Linking MadLoop5 vs Tensor Integral Reduction (TIR).

because TIR can reduce

$$\mathcal{N}(l) = l^{\mu_1} \dots l^{\mu_r}$$

MadLoop5 in MG5 v2.0

FRIEND OF USERS

- 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 blank MG5 interface.
  - Very simple one:

```
[ 2.5s   ] generate g g > t t~ [virt=QCD]
[ 6.1s   ] output
[ 4.2 ms*] launch
```

  - With options specified:

```
[ 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] QCD=6 WEIGHTED=14
[ 2min  ] output standalone MyProc
[ 1.4s*] launch -f
```

* time per phase-space point, summed over helicities and colors.

Handling BSM Models

UFO MODELS @ NLO

- Additional features in UFO@NLO:

CouplingOrder

- `expansion_order`
- `perturbative_expansion`
- `hierarchy`

CTVertices

```
V_GGZA = CTVertex(name = 'V_GGZA',
                  particles = [P.G, P.G, P.Z, P.A],
                  color = ['Tr(1,2)'],
                  lorentz = [L.R2_GGVV],
                  loop_particles = [[[P.u], [P.c], [P.t]], [[P.d], [P.s], [P.b]]],
                  couplings = {(0,0,0) : C.R2_GGZAup, (0,0,1) : C.R2_GGZAdown},
                  type = 'R2')
```

CTParameters

```
MyCTParam = CTParameter(name = 'MyCTParam',
                        type = 'real',
                        value = {-1 : 'A', 0 : 'B'}
                        texname = 'MadRules')
```

counterterm

attribute to Parameters and Particles

```
Param.GS.counterterm = {(1,0,0) : CTParam.G_UVq.value,
                        (1,0,1) : CTParam.G_UVb.value,
                        (1,0,2) : CTParam.G_UVt.value,
                        (1,0,3) : CTParam.G_UVg.value}
```

Speed of one-loop amplitudes

Color summed, with OPP

Process	t_{pol} [ms]	n_{hel}	t_{unpol} [ms]
u u~ → t t~	0,52	**3**/16	0,72
u u~ → W^+ W^-	0,43	**10**/36	1,00
u d~ → W^+ g	0,87	**6**/24	1,51
g g → t t~	2,51	**6**/16	5,42
u u~ → t t~ g	7,44	**16**/32	27,5
u u~ → W^+ W^- g	9,3	**36**/72	81,8
u d~ → W^+ g g	13,5	**12**/48	36,9
g g → t t~ g	40,8	**32**/32	381
u u~ → t t~ g g	142	**32**/64	1010
u u~ → W^+ W^- g g	166	**72**/144	2820
u d~ → W^+ g g g	260	**24**/96	1'310
g g → t t~ g g	826	**64**/64	16'900
u d~ → W^+ g g g g	9400	**48**/192	90'900

Polarized timing **competitive**

$t_{2\to2} : t_{2\to3} : t_{2\to4} \lesssim 1 : 40 : 800$ ms

Unpolarized timing
Good enough for $2 \to 3$

Might need further
improvement for $2 \to 4$

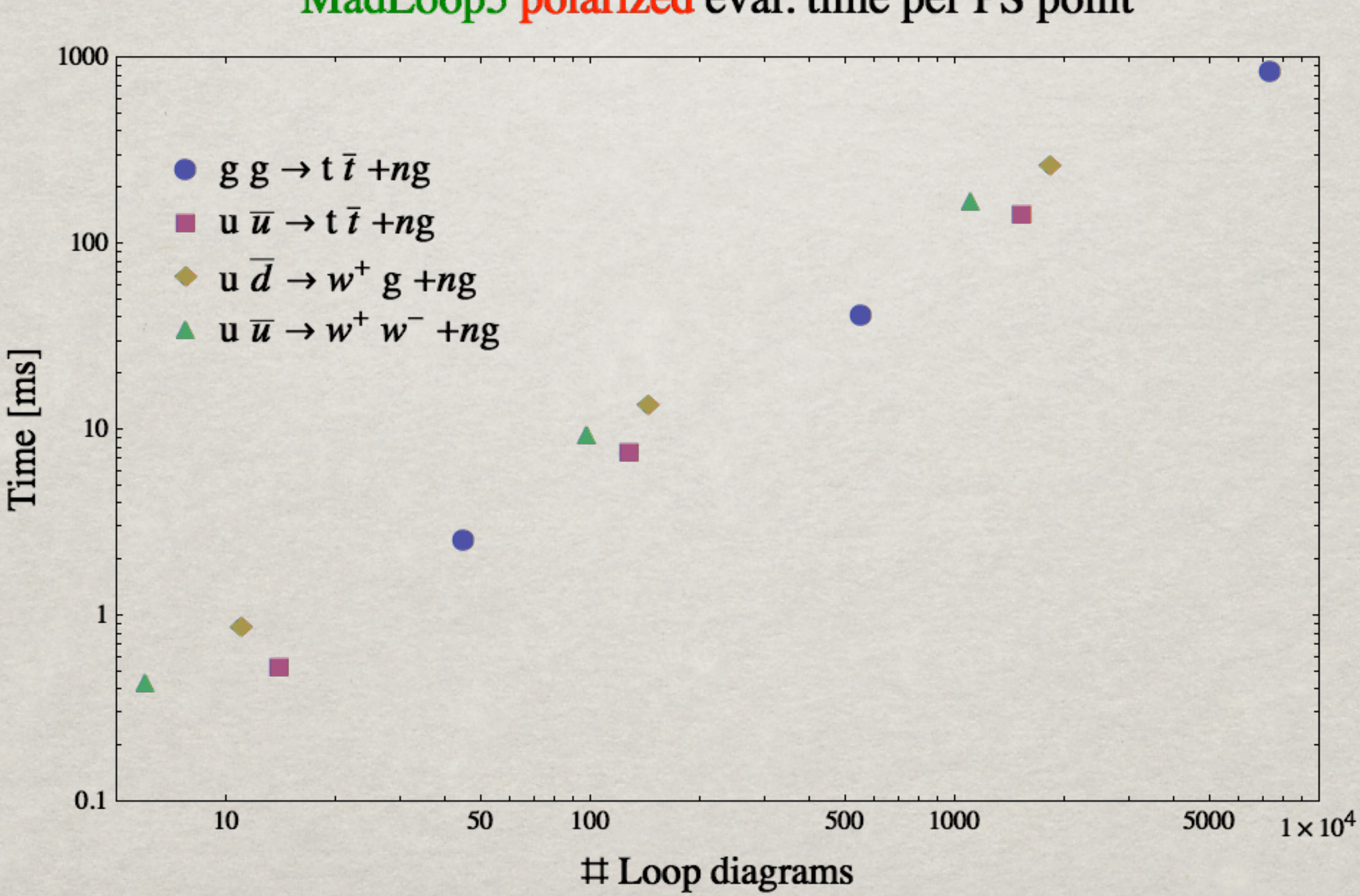
Higher multiplicity
$2 \to 5$ generation feasible

But evaluation is slow, so only useful to cross-check other codes

(Ex. gg→gggg successfully cross-checked vs NGluon[S. Badger])

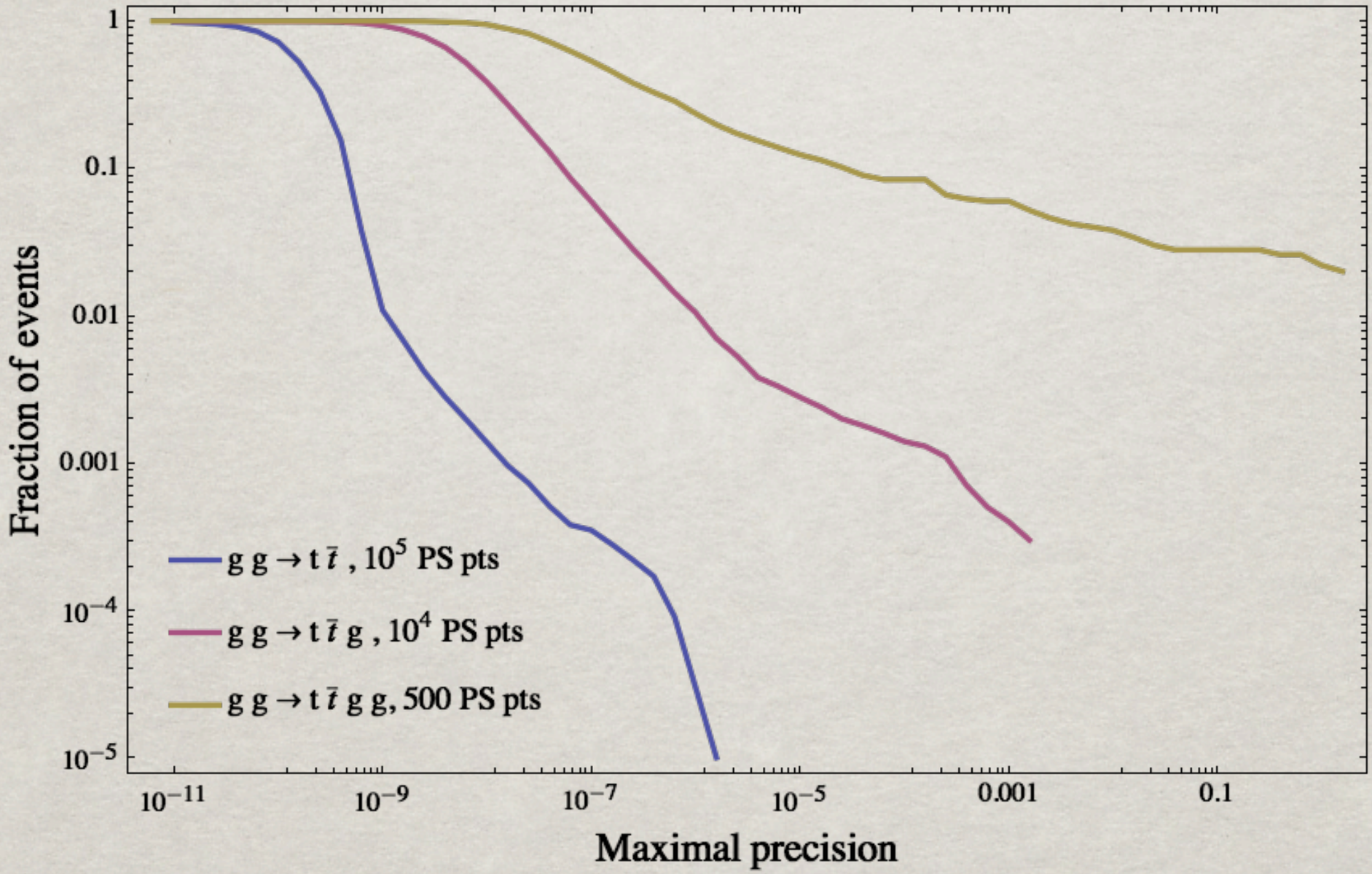
Linear scaling with # loop diags

Higher rank loops appearing at larger multiplicities are no obstacle!



Numerical stability with OPP

2 > 4, problems ahead...



Uniformly distributed points with $\sqrt{s}$ = 1TeV, p_t > 50 GeV and ΔR_{ij}>0.5

Numerical stability with OPP

Double precision is not always enough!

Stability probed by two methods:

- Loop reading direction : $D_0D_1...D_{n-1}D_n \rightarrow D_nD_{n-1}...D_1D_0$
  - ➥ Advantage: The coefficients of N(q) need not be recomputed.

- Two PS point rotations : (E,x,y,z) → (E,z,-x,-y) and (E,x,y,z) → (E,-z,y,x)

Fraction of points with less than 3 digits accuracy:

$2 \rightarrow 2$ $<< 10^{-3}$ %

$2 \rightarrow 3$ ~ 0.01 %

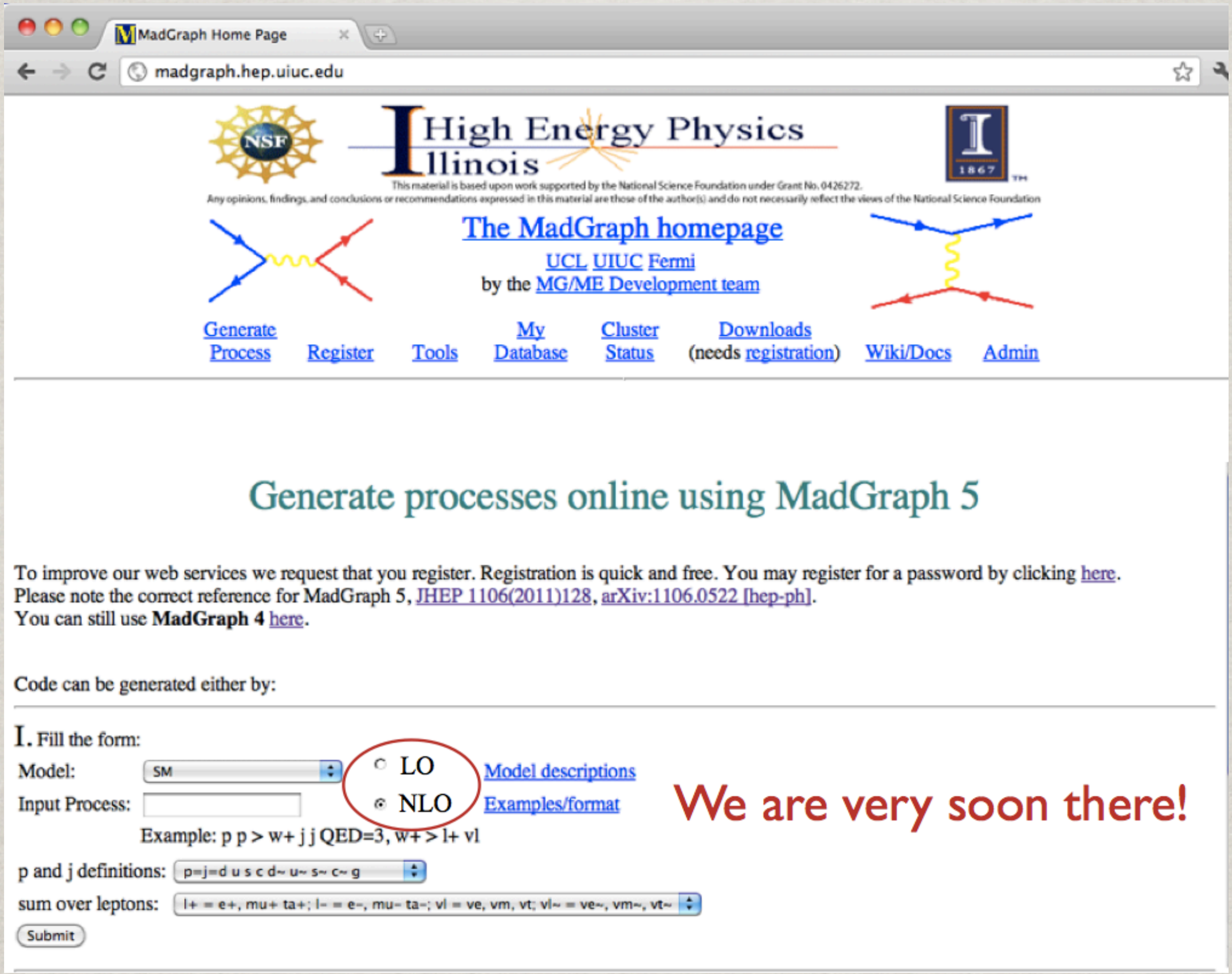
$2 \rightarrow 4$ ~ 7 % (!)

Further investigation necessary for $2 \rightarrow 4$. (TIR might solve...)

Summary

- One-loop integrals can be written as coefficients a, b, c and d times scalar functions and a rational part *R*
- The traditional approach for computing one-loop diagrams (Passarino-Veltman reduction) needs analytical knowledge of the loop and leads to very large expressions for loops of higher ranks. Not for automation...
- The OPP reduction works at the integrand level: choosing specific values of the loop momentum results in a linear system of equations, which can be solved numerically and efficiently
- MadLoop efficiently generates loop diagrams by cutting them open, which results in tree-level diagrams with two extra external particles
- MadGraph5 reach for NLO QCD computation is: $2 \to 3$, $2 \to 4$

Thanks



We are very soon there!

Small live Demo

Additional Slides

Thought-to-be Final Word

Be ready to try the MadGraph v2.0 by yourself

MadLoop5 in **MadGraph5 v2.0**, a new 1-loop generator

- Numerical, diagrammatic, some recursive features
- Open-loops method exploited, *i.e.* loop-momentum polynomials
- PUBLIC release very soon (keep an eye on launchpad.net/madgraph5)

User-friendly, Automated, Flexible, Unique framework

- BSM model covered thanks to UFO and ALOHA flexibility.
- User-friendly thanks to MG5 interfaces.
- Fully automated, from the hard process output to event generation.

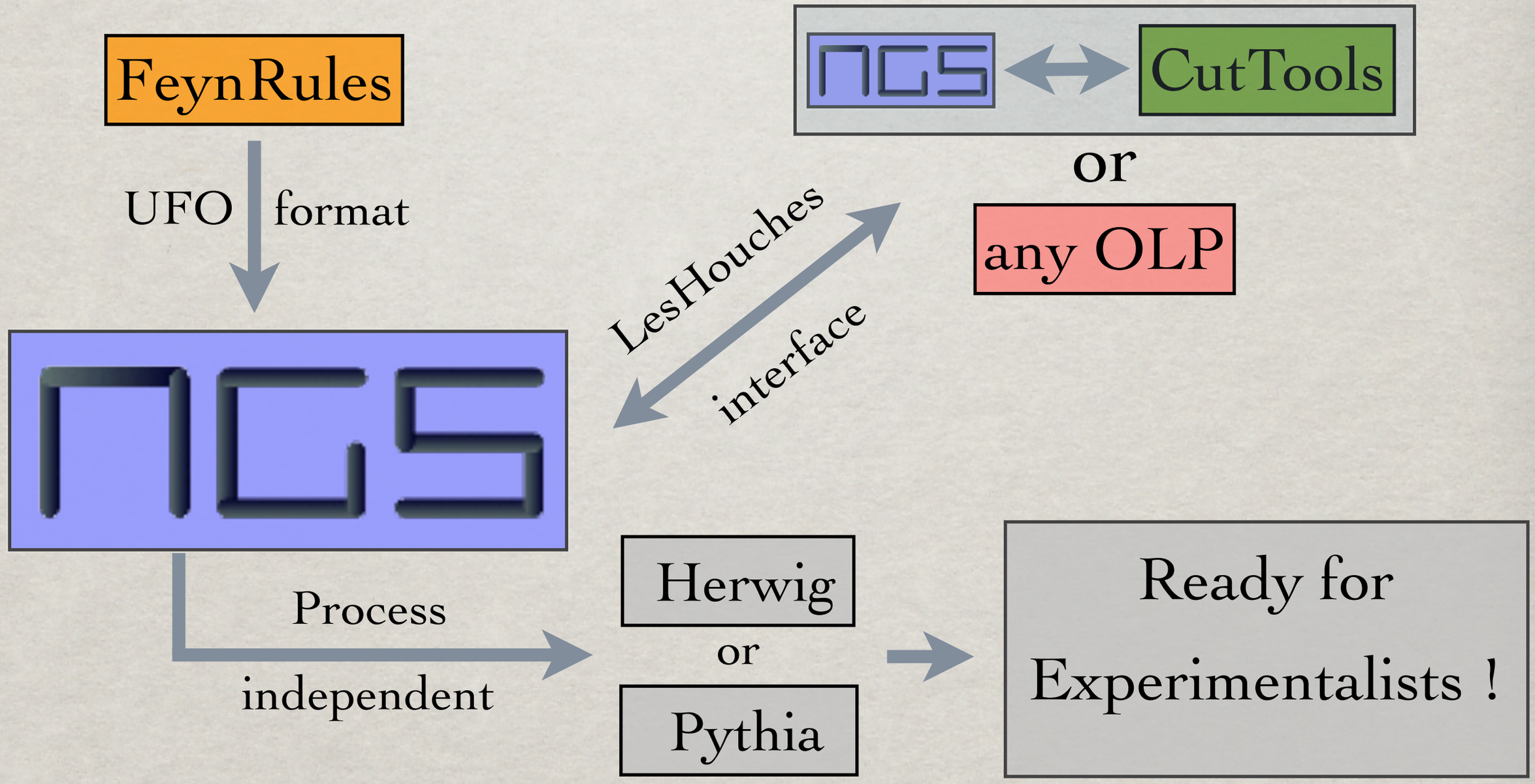
Fast, Stable

- Fast enough to cover today's processes of interest, $2 \to 4$ takes O(5-50)s
- Stable thanks to quadruple precision when needed.

AMC@NLO

FULL AUTOMATION...

... in MadGraph5 v2.0!



Numerical stability with OPP

Quadruple precision solves

→ In general, accuracy is **worse** than with Tensor Integral Reduction

→ Quadruple precision cures the Unstable PS (UPS) points but...

- ... is 100 times slower! (This is for complete qd, but double-double would be only 8 times slower)
  - ➥ So 1% of UPS is already enough to double the integration time.

- ... a very (very) small fraction of the points will remain unstable.
  - ➥ What to do with these Exceptional PS points (EPS)?

→ Need to assess that the stability tests used are accurate.

→ Also need to investigate possible correlation between small weight of the ME and the unstability of its evaluation.

MadLoop V4 to V5

Great improvements

✓ = non-optimal | ✓ = done optimally | ✗ = not done | ✗ = not done YET

Task	MadLoop V4	MadLoop V5
Generation of L-Cut diagrams, loop-basis selection	✓-	✓++
Color Factor computation	✓-	✓
Counter-term (UV/R2) diagrams generation	✓-	✓
Mixed order perturbation (generation level)	✗	✓
File output and run-time speed	✓--	✓++
Drawing of Loop diagrams	✗	✓
4-gluon R2 computation	✗	✓
Automated parallel tests	✗	✓
Automatic output sanity checks (Ward, ε^{-2})	✓	✓
EPS handling	✓-- (no qp)	✓- (qp)
Virtual squared	✓-	✓
Decay Chains	✗	✗
Automatic loop-model creation	✗	✗
Complex mass scheme and massive bosons in the loop	✗	✓/✗

Process Details

Process	unpol t_{coef} / t_{tot}	pol t_{coef} / t_{tot}	n_{loops} / n_{loop_groups}
u u~ → t t~	42%	20%	8 / 14
u u~ → W^+ W^-	69%	21%	5 / 6
u d~ → W^+ g	52%	16%	9 / 11
g g → t t~	66%	25%	26 / 45
u u~ → t t~ g	78%	18%	54 / 128
u u~ → W^+ W^- g	91%	24%	40 / 98
u d~ → W^+ g g	69%	17%	61 / 144
g g → t t~ g	92%	29%	164 / 556
u u~ → t t~ g g	88%	22%	374 / 1530
u u~ → W^+ W^- g g	95%	25%	260 / 1108
u d~ → W^+ g g g	84%	20%	405 / 1827
g g → t t~ g g	97%	35%	1168 / 7356
u d~ → W^+ g g g g	94%	21%	3255 / 25666

Default vs Open-loop timings

MadLoop5 opt vs default polarized eval. time per PS point