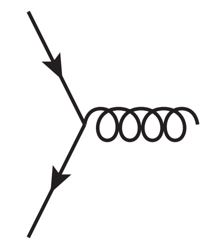
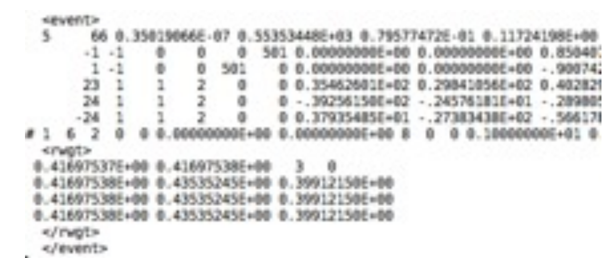
ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

19TH APRIL 2013 VALENTIN HIRSCHI

aMC@NLO and MadLoop

aMC@NLO core team: R.Frederix, S.frixione, V.H., F.Maltoni, O.Mattelaer, P.Torrielli, R.Pittau

Prediction chain

$SU(3) \times SU(2) \times U(1)$	**Symmetries**	$G^{\mu\nu}G_{\mu\nu} + \imath\bar{q}_{(i)}D_\mu\gamma^\mu q_{(i)} + \cdots$
$G^{\mu\nu}G_{\mu\nu} + \imath\bar{q}_{(i)}D_\mu\gamma^\mu q_{(i)} + [\cdots]$	**Model**	 $= \imath\gamma^\mu t^a_{ij}$, ...
$pp \to jj \quad \mathrm{QCD} = 2$	**Matrix Element**	$\mathcal{M}^2_{gg\to d\bar{d}}$, ...
`matrix.f`	**Partonic Events**	
`events.lhe`	**Hadron level**	$\{\pi^0, K^+, e^+, p, \cdots\}$
`events.hep`	**Detector level**	

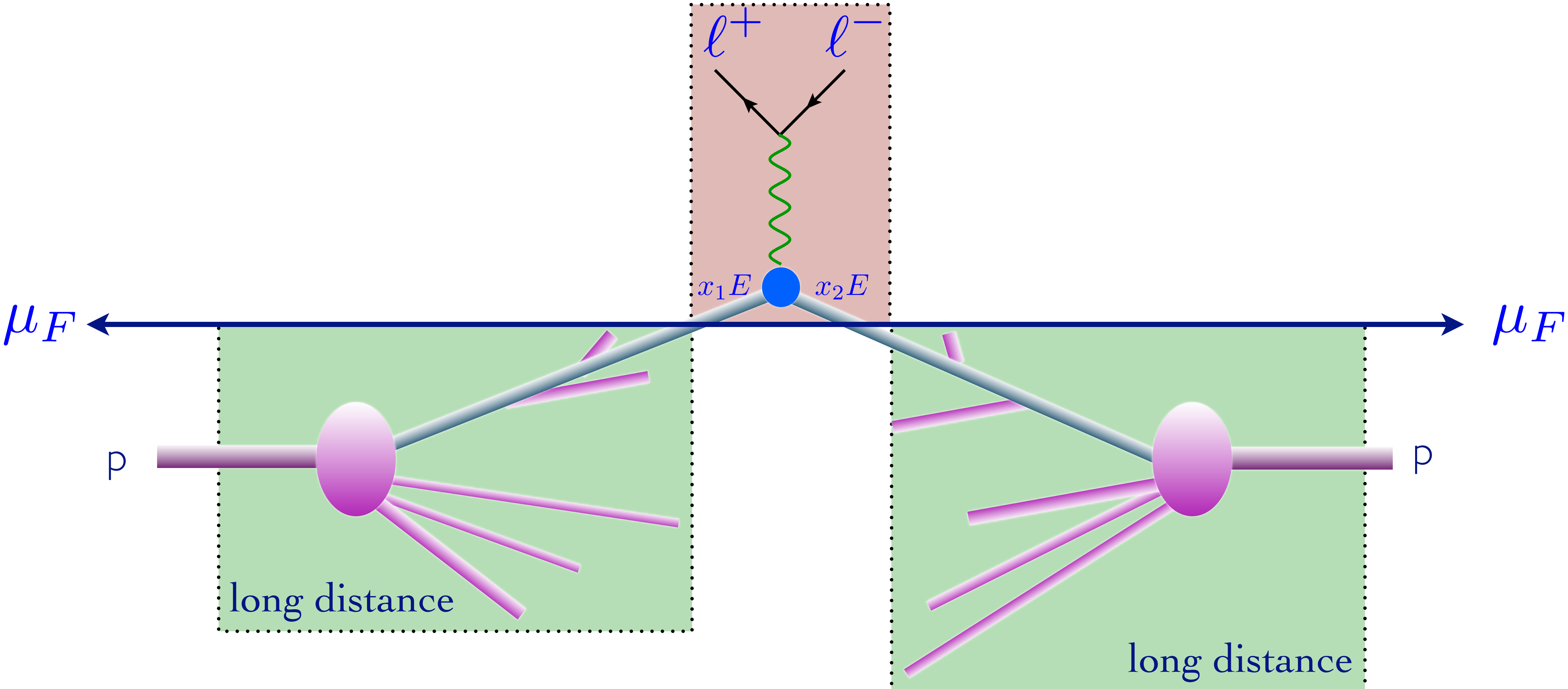

```
<event>
5     66 0.35019066E-07 0.55353448E+03 0.79577472E-01 0.11724198E+00
      -1 -1    0    0    0  501 0.00000000E+00 0.00000000E+00 0.85040
       1 -1    0    0  501    0 0.00000000E+00 0.00000000E+00 -.90074
      23  1    1    2    0    0 0.35462601E+02 0.29841056E+02 0.40282
      24  1    1    2    0    0 -.39256150E+02 -.24576181E+01 -.28980
     -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>
```

The Framework

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

MadAnalysis 5

Master formula for hadron collisions



$$\sum_{a,b} \int dx_1 dx_2 d\Phi_{\mathrm{FS}}\, f_a(x_1, \mu_F) f_b(x_2, \mu_F)\, \hat{\sigma}_{ab \to X}(\hat{s}, \mu_F, \mu_R)$$

Phase-space integral

Parton density functions

Parton-level cross section

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

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

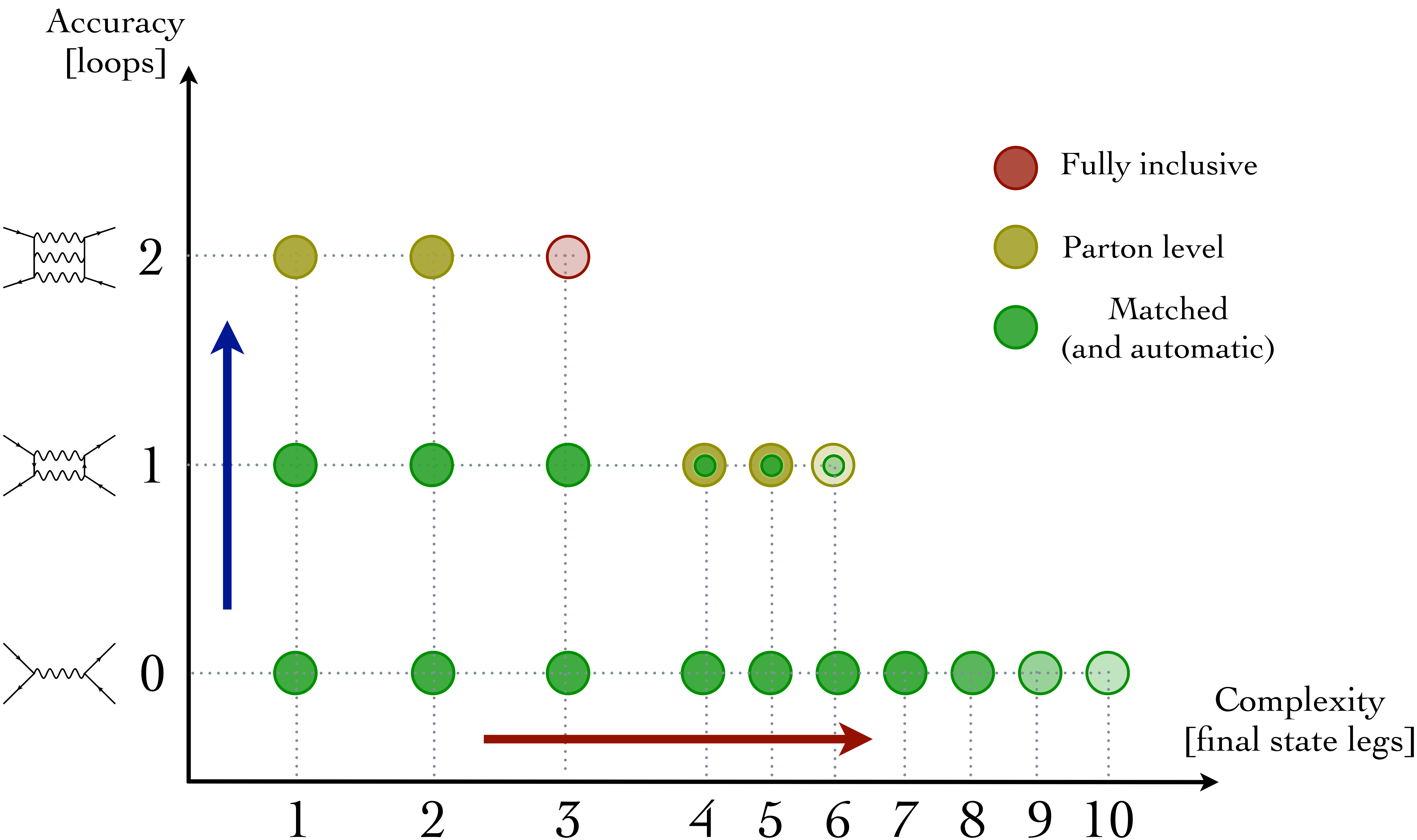
Why adding N's ?

- Some analysis strongly rely on the capability of accurately simulating the signal (single-top at Tevatron)

- Lack of predictivity means that people will overstretch predictions without noticing (*i.e.* power shower)

- Why fully exclusive predictions? To have them go through detector simulation

How to add N's ?



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

Merging multiplicities ?

- Better jet description

- Probing of new initial state channels opening up

- Most of 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)

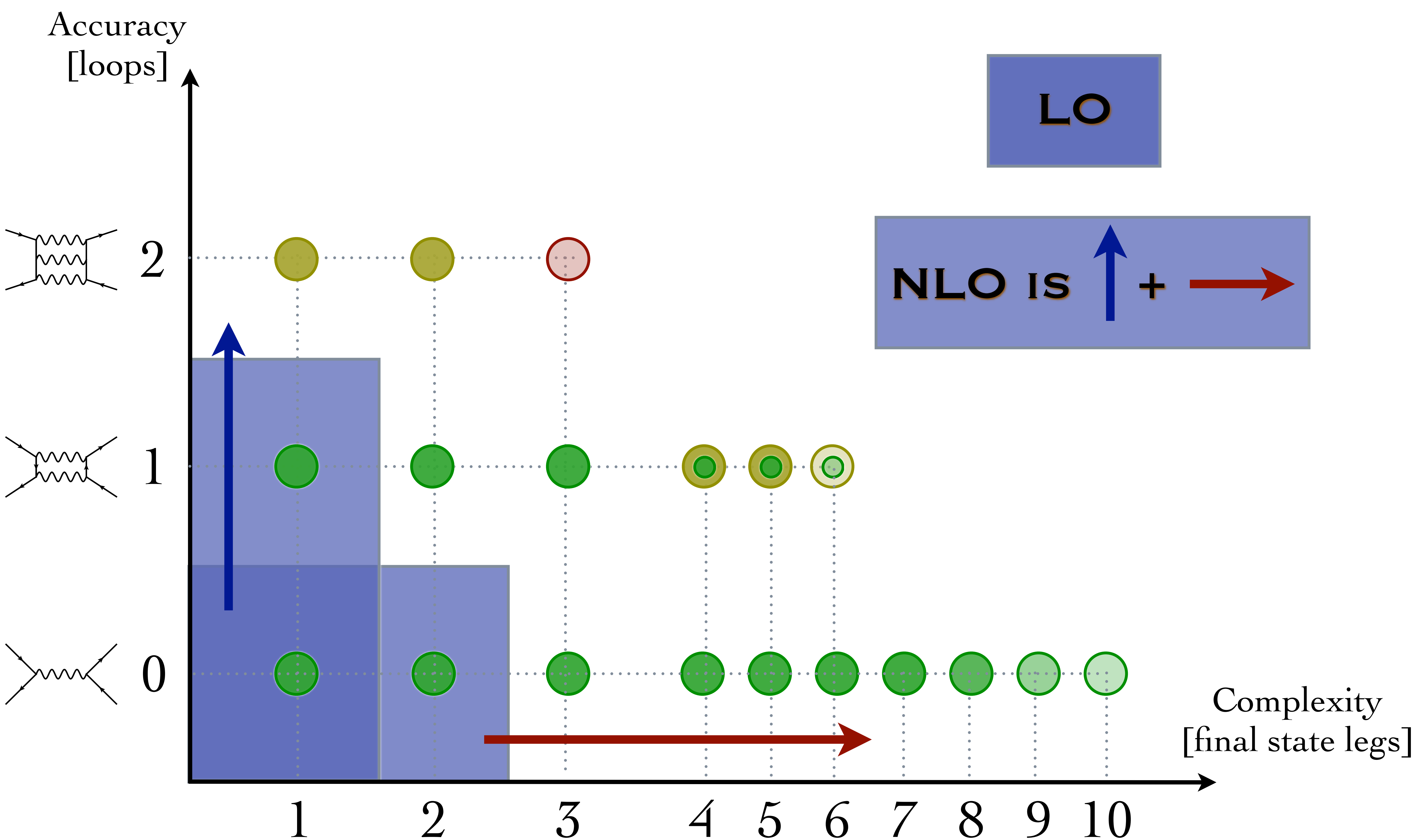
Need for Automation

- Merging trees is easy comparatively to computing loops , and they bring most of the interesting higher order features, needs incentive for not sticking to them!

- So, the advantage of having more precise results when also considering loops will only outweigh their considerable technical difficulties if their implementation is FULLY AUTOMATED (*i.e.* at zero human cost)

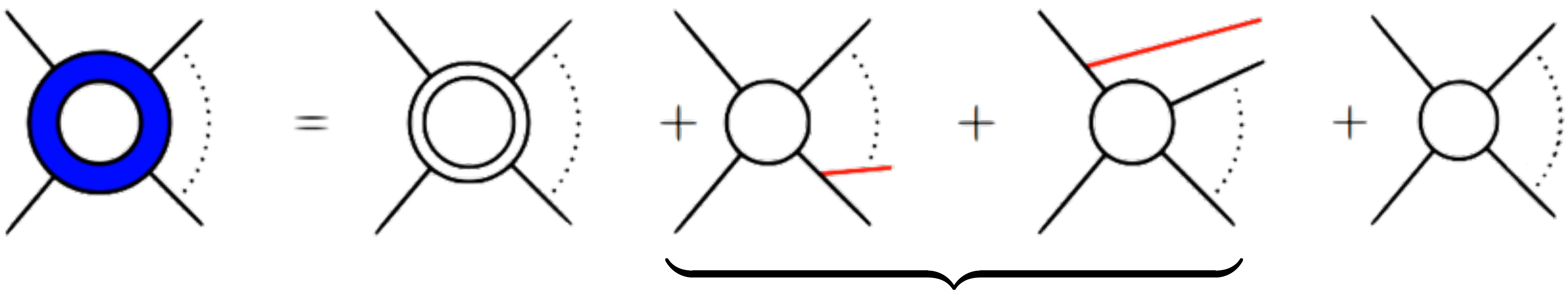
- Automation byproducts : User-friendly, all-in-one software, reliability (new results exploit the same elementary building blocks and are therefore correct almost by construction)

Computing NLO



NLO Anatomy

Fixed-order NLO contributions have two parts



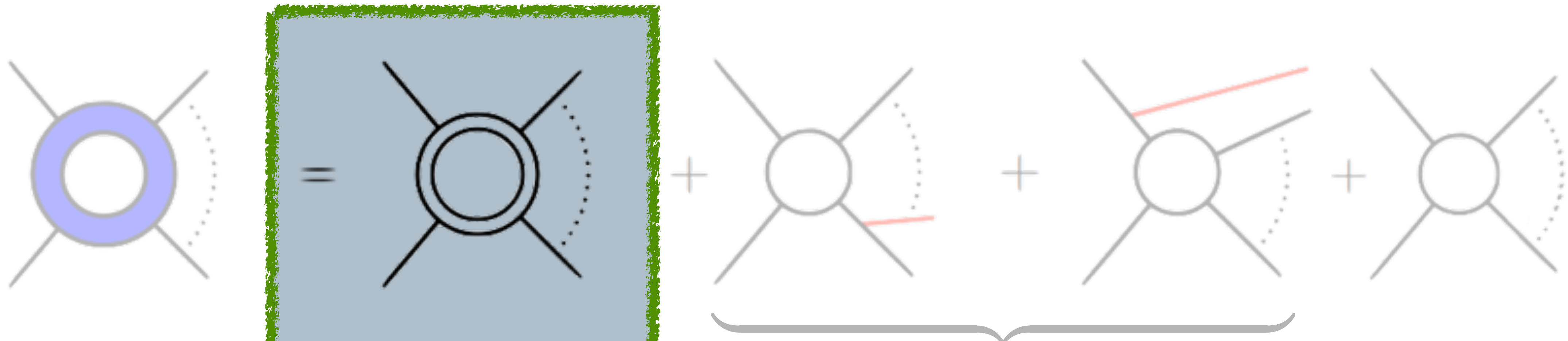
$$\sigma^{\rm NLO} = \int_m d^{(d)}\sigma^V + \int_{m+1} d^{(d)}\sigma^R + \int_m d^{(4)}\sigma^B$$

Virtual part — Real emission part — Born

MadLoop — MadFKS — Solved

aMC@NLO

Loop computations

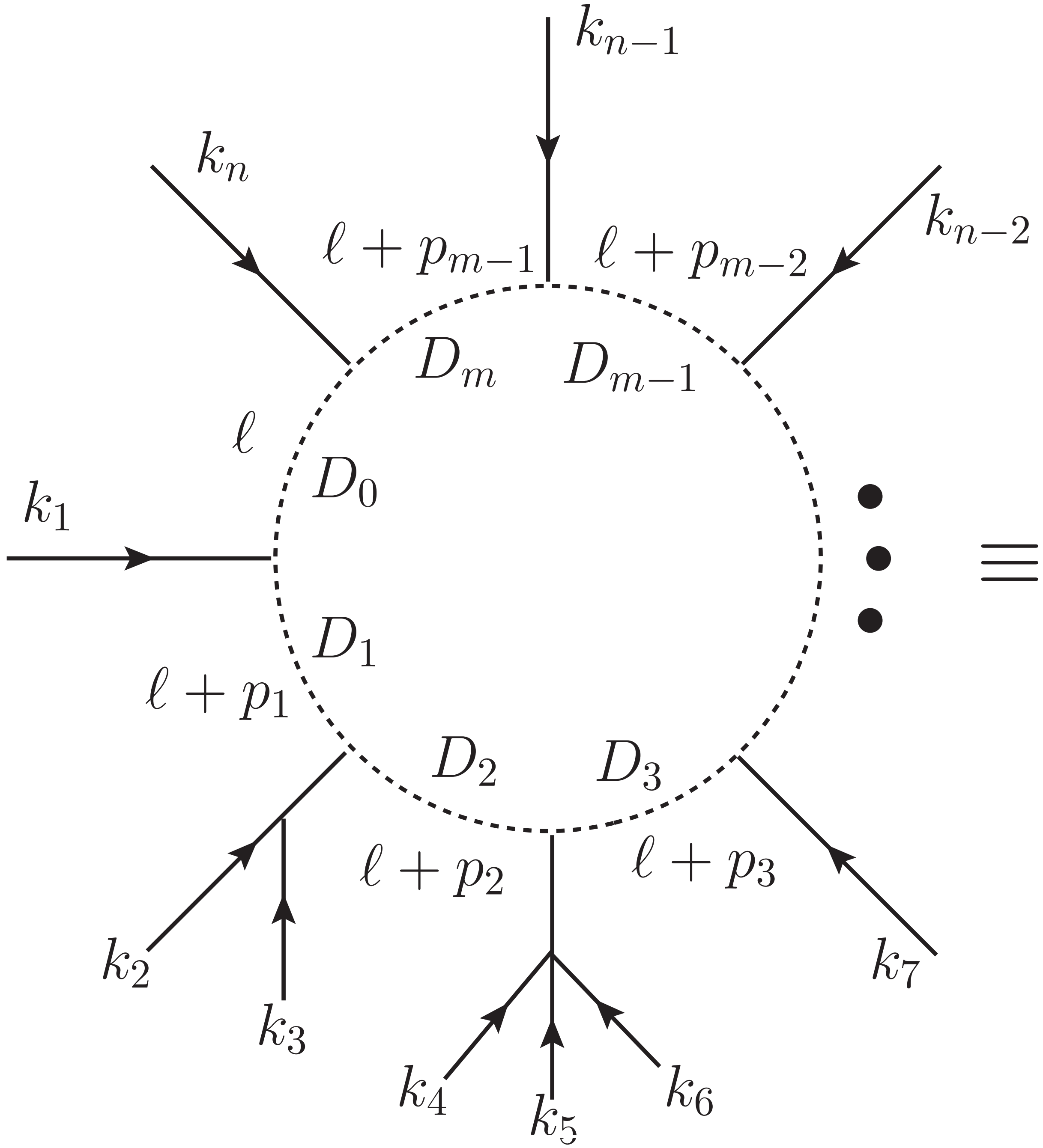


$$\sigma^{\rm NLO} = \int_m d^{(d)}\sigma^V + \int_{m+1} d^{(d)}\sigma^R + \int_m d^{(4)}\sigma^B$$

Virtual part — Real emission part — Born

MadLoop — MadFKS — Solved

ONE-LOOP INTEGRAL



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

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

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

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

Scalar integral Basis

Smart people + Lorentz invariance + 4-dimensional space time =

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

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

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

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

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

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

NLO AWAKENING

- The "loop revolution": new techniques for computing one-loop diagrams 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, SAMURAI)
    [Ossola, Papadopoulos, Pittau 2006; del Aguila, Pittau 2004; Mastrolia, Ossola, Reiter, Tramontano 2010;...]
  - Tensor reduction (e.g. Golem, COLLIER, PJFry++)
    [Passarino, Veltman 1979; Denner, Dittmaier 2005; Binoth Guillet, Heinrich, Pilon, Reiter 2008, Yundin 2011]

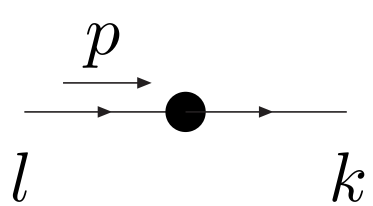
Integrand reduction

- The integrand (or OPP [Ossola, Papadopoulos, Pittau 2006]) reduction method is a purely numerical algorithm that has been automated in the CutTools and SAMURAI computer codes to find the scalar loop coefficients
- The OPP technique is what was adopted in our MadGraph-based framework to compute loop diagrams.
- This method takes the numerator of the loop integrand, $N(l)$, as input and is only limited by the rank in l of $N(l)$. Very well suited for an automated numerical approach.
- R_2 is a part of the finite contribution of the virtual which is unobtainable by OPP and must be recovered using additional effective "R_2 Feynman rules".

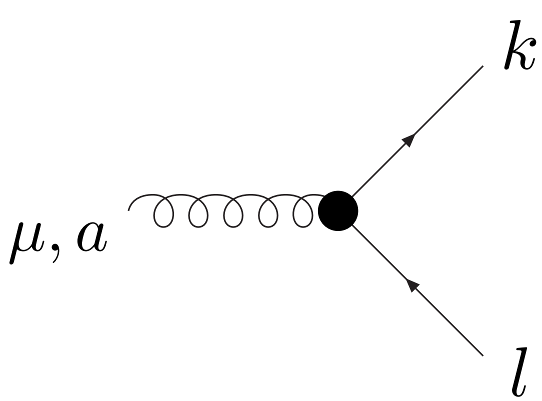
R_2 Feynman rules

- 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 those of the UV counterterms. Here at two examples of such R_2 rules:



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



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

[Draggiotis, Garzelli, Papadopoulos, Pittau]

- Unfortunately these Feynman rules are model dependent.
➥ We work on using FeynRules combined with FeynArts to derive them automatically for any model along with UV renormalization.

From a single loop to $M^{(1\text{-loop})}$

- Several public computer codes have been built to implement these loop reduction techniques to form the complete 1-loop ME

BlackHat

[C. F. Bergera, Z. Bernb, L. J. Dixonc, F. Febres Corderob, D. Fordec, H. Itab, D. A. Kosowerd, D. Maître]

GoSam

[G. Cullen, N, Greiner, G. Heinrich, G. Luisoni, P. Mastroliad, G. Ossola,h, T. Reiter, F. Tramontano]

MadGolem

[D. Goncalves-Netto, D. Lopez-Val, K. Mawatari, T. Plehn and I. Wigmore]

Helac-NLO

[G. Bevilacqua, M. Czakon, M. V. Garzelli, A. van Hameren, A. Kardos, C. G. Papadopoulos, R. Pittau, M. Worek]

MCFM

[J.M. Campbell, R. K. Ellis, and others]

NGluon

[S.Badger, B,Biedermann and P.Uwer]

The MadGraph Solution

- MadGraph good at numerically computing matrix elements. Exactly what is needed by CutTools to get the coefficients of the scalar integrals.

  ➥ However, MadGraph only handles tree-level diagrams

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

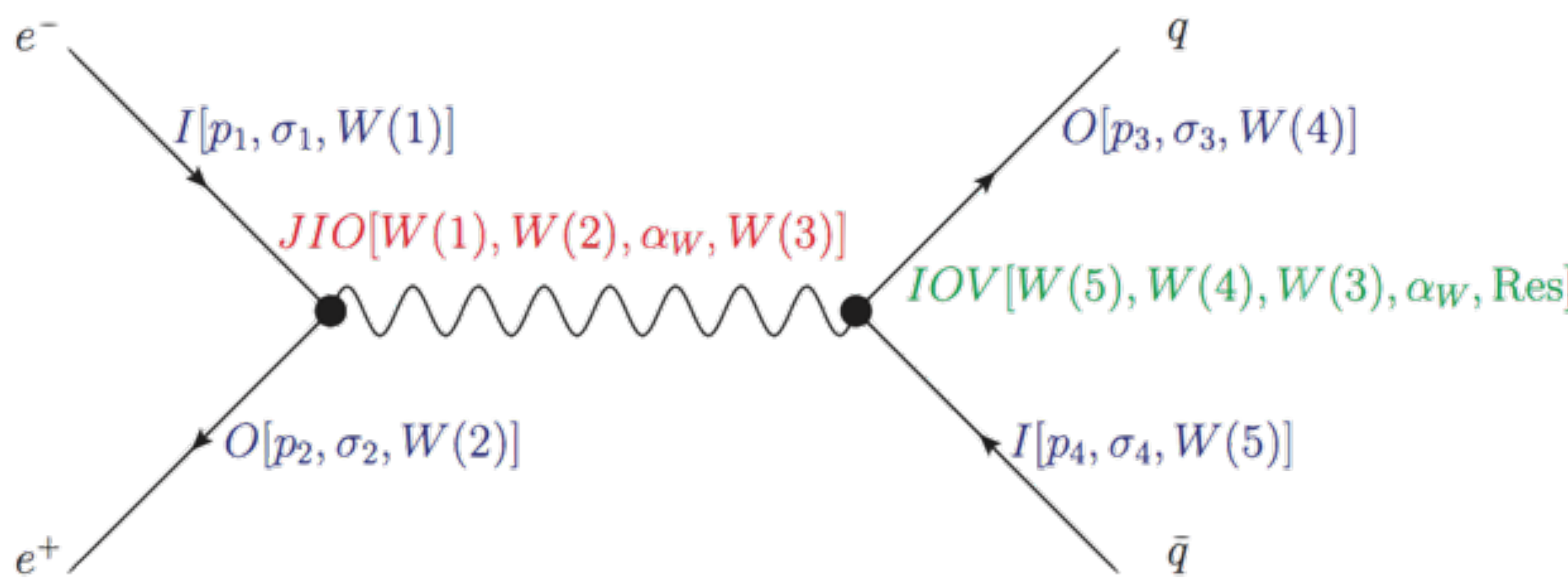
MadLoop and aMC@NLO

together in MadGraph5 v2.0

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

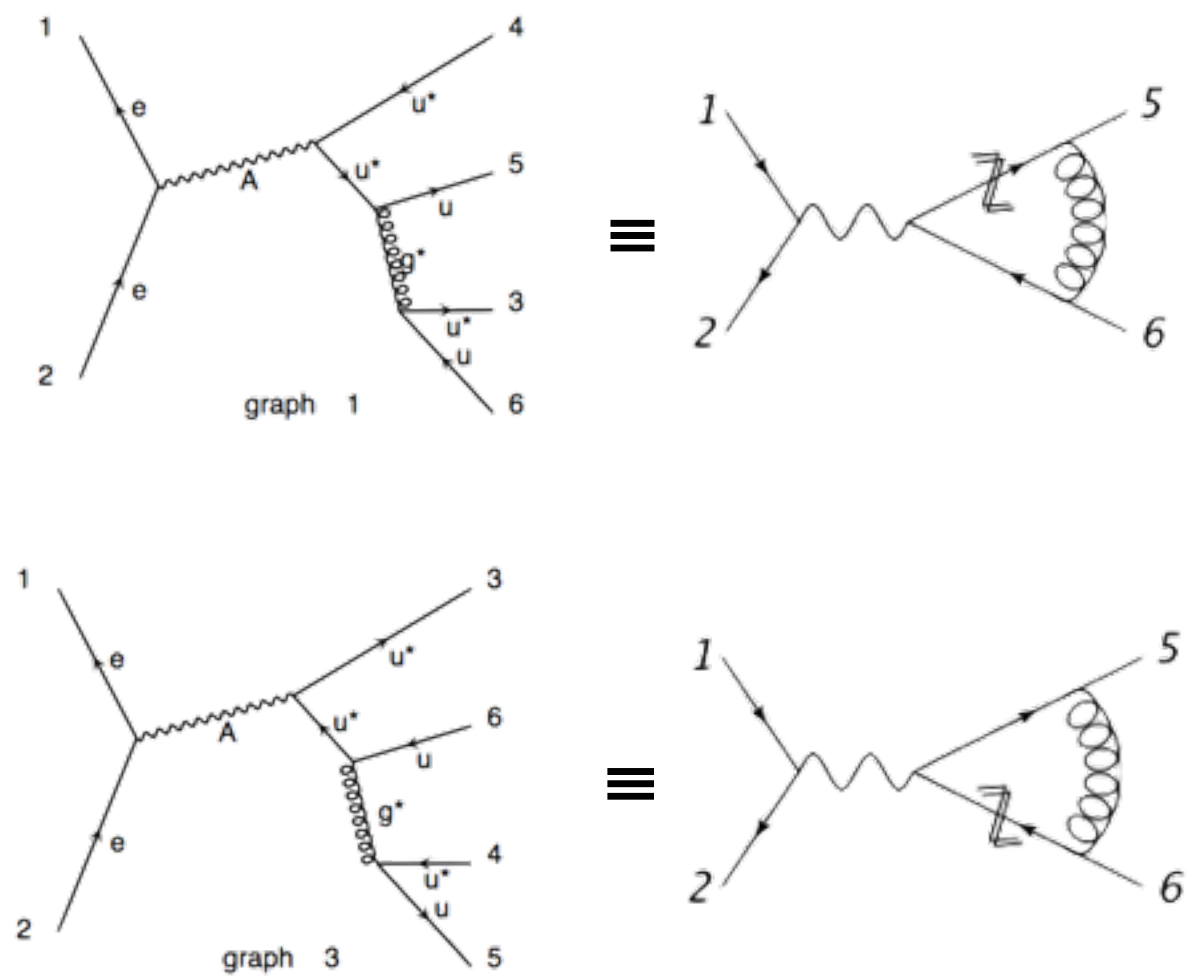
Generating loop diagrams

- Loop diagrams are nothing but tree diagrams with two FS merged.

Take advantage of MG5 efficient tree-diagram generation

Filter out tadpoles and wf renorm. loops on the fly.

Disregard loop-particles already considered as L-Cut particles.



OPTIMIZATIONS

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

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

➥ 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.

Overall speedup of a factor 10+ w.r.t ML4

Performances - Code generation

Process	Exe. size [MB]	t_{code} [s]
u u~ → t t~	3.4	9.1
u u~ → w+ w-	3.5	12.4
u d~ → w+ g	3.5	13.9
g g → t t~	3.6	12.8
u u~ → t t~ g	3.7	18
u u~ → w+ w- g	3.9	35
u d~ → w+ g g	3.8	24
g g → t t~ g	4.2	62
u u~ → t t~ g g	4.8	180
u u~ → w+ w- g g	4.8	204
u d~ → w+ g g g	5.2	254
g g → t t~ g g	9.9*	1230
u d~ → w+ g g g g	24**	9370

*,**: Color + helicity data = 25MB , 191 MB

Executable size: **a few MB**

Mild scaling with multiplicity.

Generation time **< 1 hour**

Not a limiting factor.

Could generate
u d~ → w+ g g g g
or even
g g → g g g g

Running 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 → 3**

Might need further improvement for **2 → 4**

Higher multiplicity

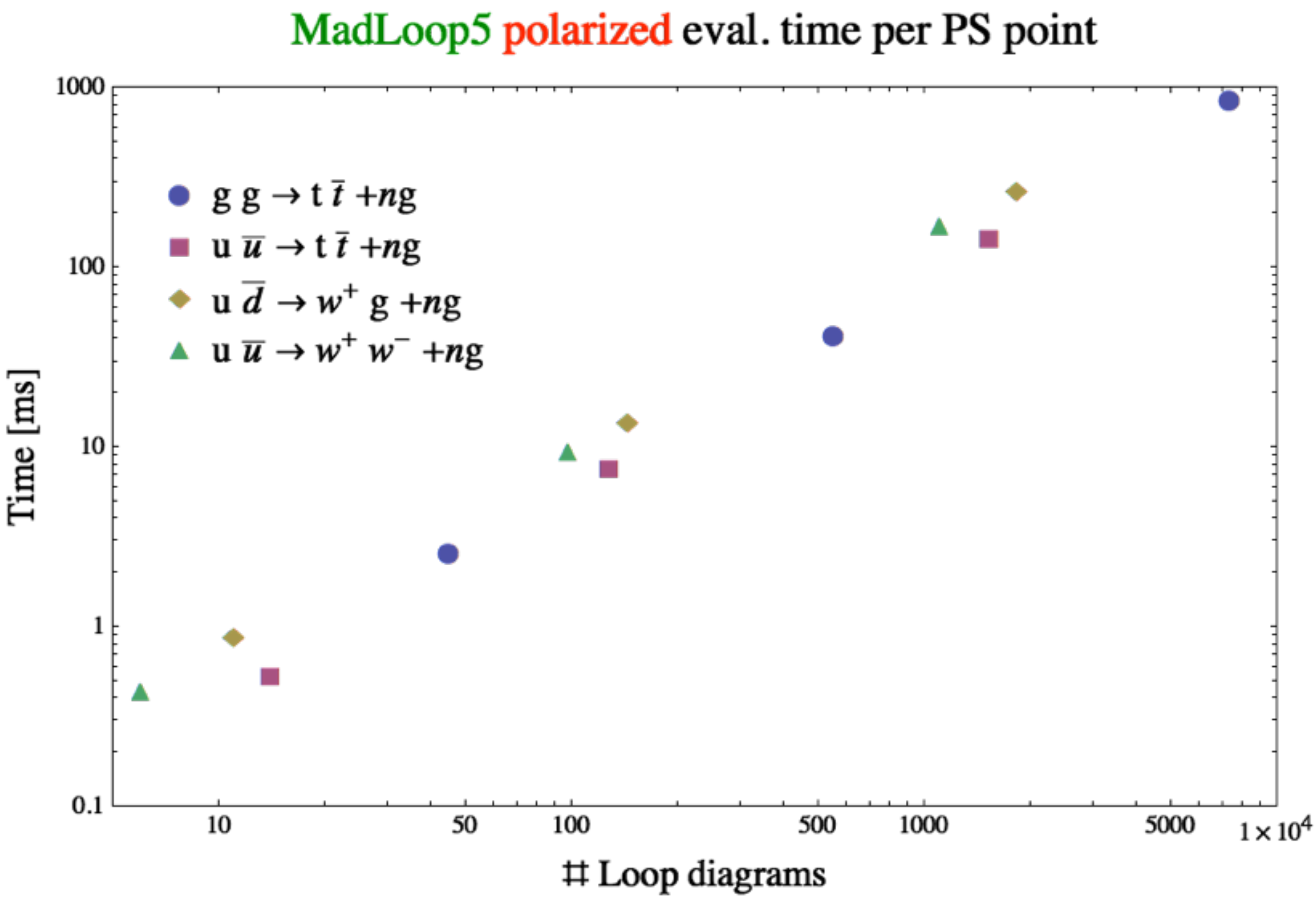
2 → 5 generation feasible

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

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

Timing scaling with # loop diags

Higher rank loops appearing at larger multiplicities are no obstacle!



MadLoop5 SA 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 MyProc
[ 1.4s* ] launch -f
```

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

Numerical stability with OPP

Double precision is not always enough!

Stability probed via two methods:

- Loop reading direction : $D_0 D_1 \ldots D_{n-1} D_n \rightarrow D_n D_{n-1} \ldots D_1 D_0$
  ➥ 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)

- Accuracy estimation : These independent computations $E_i^{(DP)}$ of the same quantity provide an estimation $\xi^{(DP)}$ of the numerical accuracy of the result provided.

When failing:

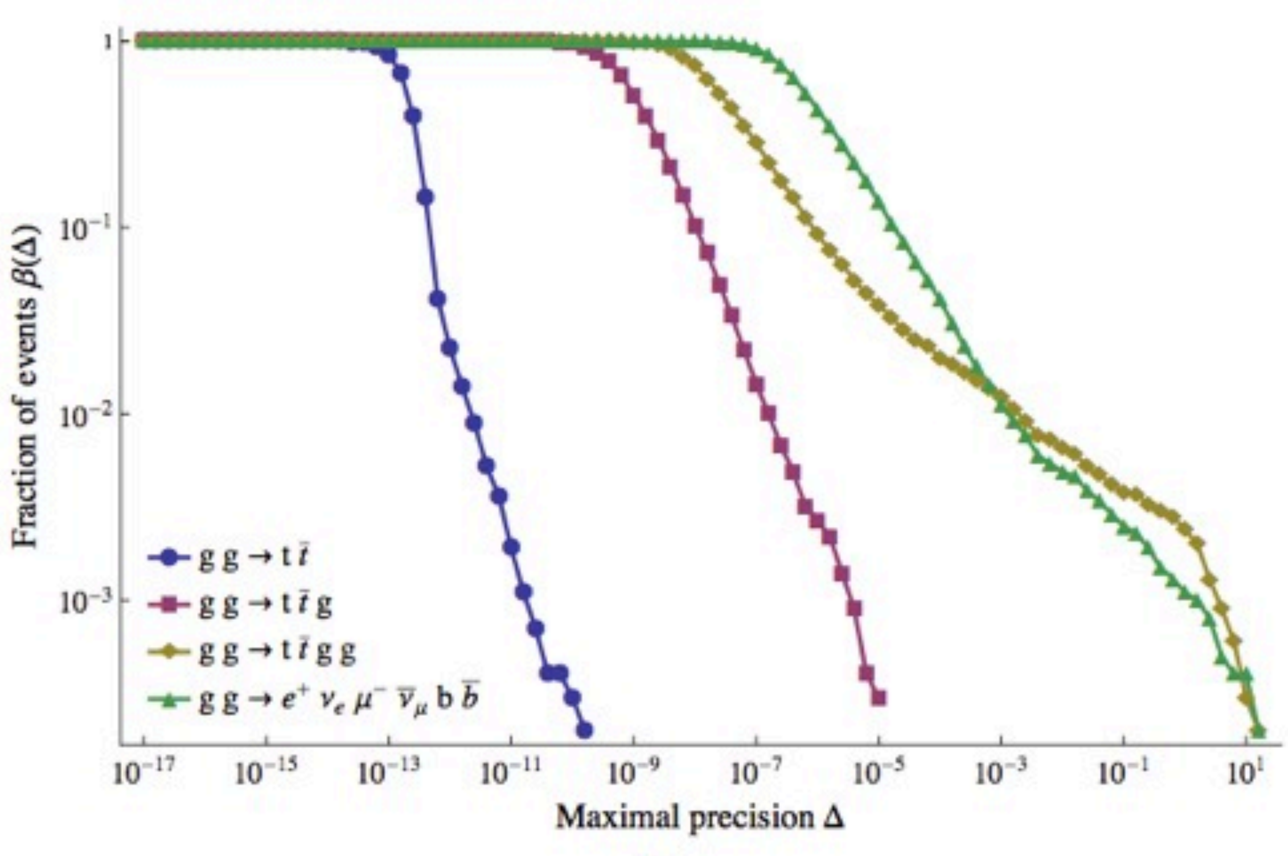
- Automatic switch to QP : MadLoop recomputes, on the flight, the loop ME for the unstable point, and its stability evaluation yields $\xi^{(QP)}$. If $\xi^{(QP)}$ is acceptable then point is UPS if not it is EPS

- EPS are inexistent from a practical point of view, but UPS can become a threat as their computation is 100 times slower.

Stability study of individual diagrams, or even helicity configurations, is given up (no great gain and did forbid a great optimization).

Numerical stability with OPP

2 > 4, problems ahead...



(a)

Process	$\beta(10^{-3})$	$-\log_{10}(\text{med}\{\chi^{DP}\})$	P	C
$gg \to t\bar{t}$	0%	12.7	-0.2	-0.6
$gg \to t\bar{t}\,g$	0%	9.0	-0.1	-0.9
$gg \to t\bar{t}\,g\,g$	1.25%	7.5	0.0	-1.1
$gg \to e^+\nu_e\mu^-\bar{\nu}_\mu b\bar{b}$	1.11%	6.1	0.2	-1.7

- Loop direction Power P:

  $\text{med}(\log_{10}[\,|E_0^{(DP)} - E_{dir}^{(DP)}| / \xi^{(DP)}\,])$

- Stability test consistency C:

  $\text{med}(\log_{10}[\,\xi^{(DP)} / |E_0^{(DP)} - E_0^{(QP)}|\,])$

- Possibility of running this analysis from a .lhe event file for a more realistic distribution.

10^4 uniformly distributed points with $\sqrt{s}$ = 1TeV, p_t > 50 GeV and $\Delta R_{ij}>0.5$
Plot automatically obtained by using the check' command

Numerical stability with OPP

In short:

Fraction of points with less than 3 digits accuracy:

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

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

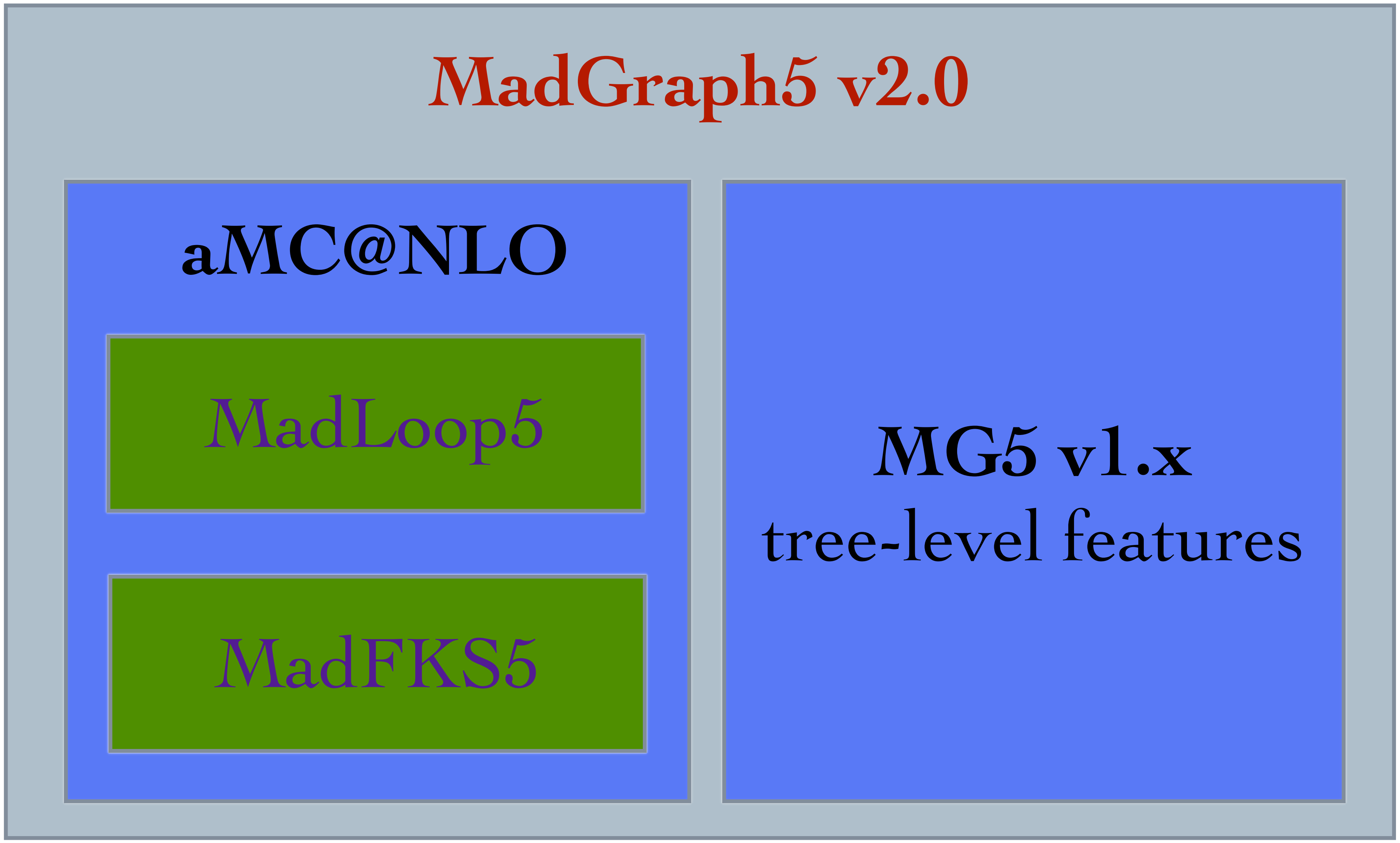
$2 \rightarrow 4$ ~ 1 % (!)

Further investigation necessary for $2 \rightarrow 4$.

OPP and TIR instability regions are not expected to overlap, so one could add another rescue method using TIR before relying on the slower quadruple precision.

(Remember that, ultimately, QP could be as fast as DP with proper hardware.)

Nomenclature



This separation is transparent to the users.

Publicly available since 8th Nov. 2012

aMC@NLO Framework

Model Building tools	Symmetries
FR inc. UV+R2	Model
MG5 inc. MadLoop	Matrix Element
MadFKS inc. MC@NLO	Partonic Events
Pythia / Herwig	Hadron level
PGS/DELPHES	Detector level

aMC@NLO5

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

	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	31.37 ± 0.03	32.86 ± 0.04
a.5	$pp \to t\bar{b}jj$	$m_{top}/4$	4	11.91 ± 0.006	7.299 ± 0.05
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	289.7 ± 0.3
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	54.1 ± 0.6
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	0.08241 ± 0.0004
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	1.333 ± 0.002

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}
```

Automatic Language-independent Output of Helicity Amplitude

O. Mattelaer *et al.* , arXiv:1108.2041 [hep-ph]

From UFO to MG5

ALOHA translate a UFO Lorentz structure

```
VVVV6 = Lorentz(name = 'VVVV6',
          spins = [ 3, 3, 3, 3 ],
          structure = 'Metric(1,4)*Metric(2,3) -Metric(1,3)*Metric(2,4)')
```

into pseudo-HELAS subroutine in a chosen language

```
 VERTEX = COUP*( (V4(1)*( (V2(1)*( (0, -1)*(V3(2)*V1(2))
$ +(0, -1)*(V3(3)*V1(3))+(0, -1)*(V3(4)*V1(4))))+(V1(1)*( (0, 1)
$ *(V3(2)*V2(2))+(0, 1)*(V3(3)*V2(3))+(0, 1)*(V3(4)*V2(4))))))
$ +( (V4(2)*( (V2(2)*( (0, -1)*(V3(1)*V1(1))+(0, 1)*(V3(3)*V1(3))
$ +(0, 1)*(V3(4)*V1(4))))+(V1(2)*( (0, 1)*(V3(1)*V2(1))+(0,
$ -1)*(V3(3)*V2(3))+(0, -1)*(V3(4)*V2(4))))))+( (V4(3)*( (V2(3)
$ *( (0, -1)*(V3(1)*V1(1))+(0, 1)*(V3(2)*V1(2))+(0, 1)*(V3(4)
$ *V1(4))))+(V1(3)*( (0, 1)*(V3(1)*V2(1))+(0, -1)*(V3(2)*V2(2))
$ +(0, -1)*(V3(4)*V2(4))))))+(V4(4)*( (V2(4)*( (0, -1)*(V3(1)
$ *V1(1))+(0, 1)*(V3(2)*V1(2))+(0, 1)*(V3(3)*V1(3))))+(V1(4)
$ *( (0, 1)*(V3(1)*V2(1))+(0, -1)*(V3(2)*V2(2))+(0, -1)*(V3(3)
$ *V2(3))))))))
 END
```

Available in Python, **C++** and F77

ALOHA available as a standalone release

New on Aloha

- ALOHA is optimizing the way it does analytical computation

Model name	Loading time, new ALOHA	Loading time, old ALOHA
SM	1.2 s	3 s
MSSM	1.4 s	5 s
Randall-Sundrum	90 s	15 min

- Abbreviation usage improves (marginally) compilation and running time
- Possibility to create ALOHA subroutine from the MG5 shell

```
mg5> output aloha FFV1_3
```

- New Outputs/Options (For the v2.0 public release)

Quadruple precision, Feynman Gauge, Spin 3/2,
Complex Mass Scheme, Open Loops techniques, generic propagators

Supported Models

Color Code	In MG5 v2.0	Ongoing progress

Effective Theories	N-Legs vertices, ∀N
color structures	Sextets, ε^{IJK}, virtually all
Lorentz Structures	All, thanks to Aloha
Spins supported	0, 1, 1/2, 3/2, 2
Gauges	Unitary, Feynman
Complex Mass Scheme	Automatic Model Conversion Available for NLO too!
Model with loop info	Import UFO Loop-models
Decays	MadSpin [P.Artoisenet,R.Frederix,O.Mattelaer,R.Rittkert]
Generic loop BSM	Need automatic UV+R2 from FR
Loop Merging	Implementing FxFx

Summarizing ...

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

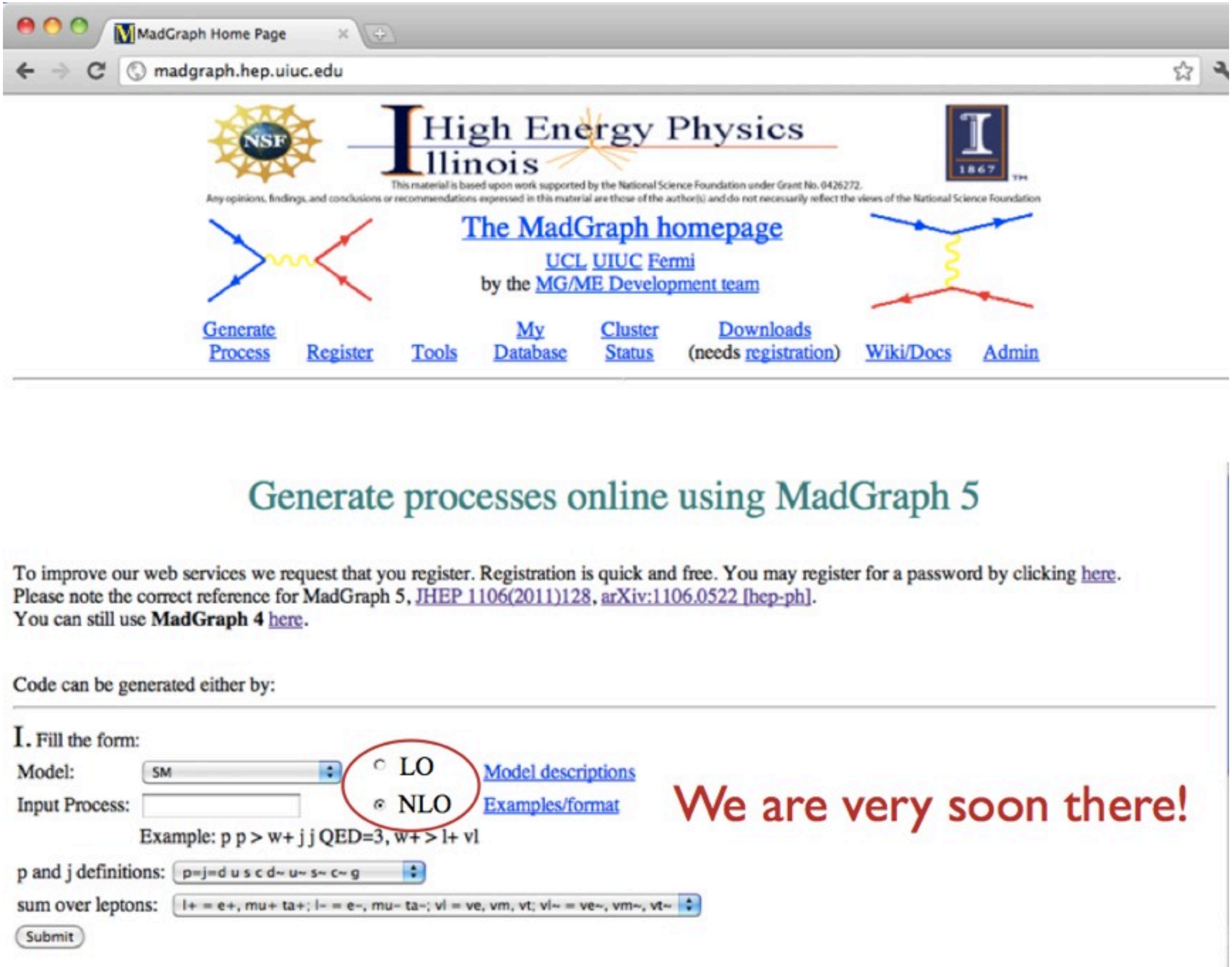
- Numerical, diagrammatic, some recursive features
- Open-loops method exploited, *i.e.* loop-momentum polynomials
- Publicly released (see on launchpad.net/madgraph5)

User-friendly, Fully Automated, Flexible : aMC@NLO

- 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, $2 \rightarrow 4$ takes O(1-5)s
- Stable thanks to quadruple precision when needed.

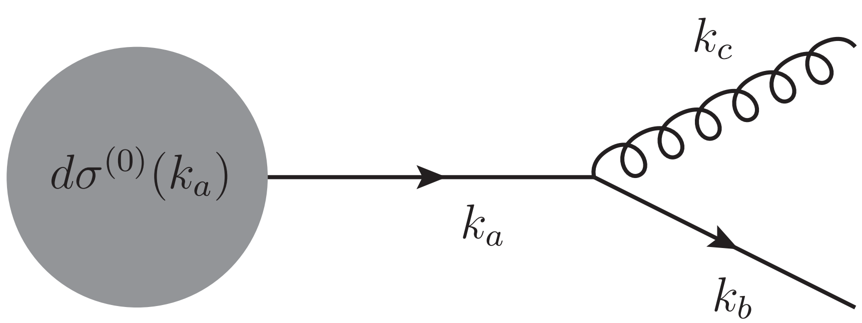


We are very soon there!

Additional Slides

The IR divergences

- The real emission trees are divergent when the emitted parton becomes soft and/or collinear.



$$k_b = z k_a + k_T + \beta_b \hat{n}$$

$$k_c = (1-z) k_a - k_T + \beta_c \hat{n}$$

$$d\sigma^{(1,R)} = \frac{\alpha_s}{2\pi} \int dk_T^2 \int_0^1 dz C_F \frac{1+z^2}{1-z} \frac{1}{k_T^2} d\sigma^{(0)}(k_a) + \mathcal{R}$$

- KLN theorem guarantees that they must cancel against those of the virtual contribution.

Subtraction

$$\sigma^{\rm NLO} \sim \int d^4\Phi_m\, B(\Phi_m) + \int d^4\Phi_m \int_{\rm loop} d^d l\, V(\Phi_m) + \int d^d\Phi_{m+1}\, R(\Phi_{m+1})$$

- To realize this poles cancellation in a semi-numerical way, one can use the subtraction method below

$$\sigma^{\rm NLO} \sim \int d^4\Phi_m\, B(\Phi_m)$$
$$+ \int d^4\Phi_m \left[\int_{\rm loop} d^d l\, V(\Phi_m) + \int d^d\Phi_1 G(\overline{\Phi}_{m+1}) \right]_{\epsilon\to 0}$$
$$+ \int d^4\Phi_{m+1} \left[R(\Phi_{m+1}) - G(\overline{\Phi}_{m+1}) \right]$$

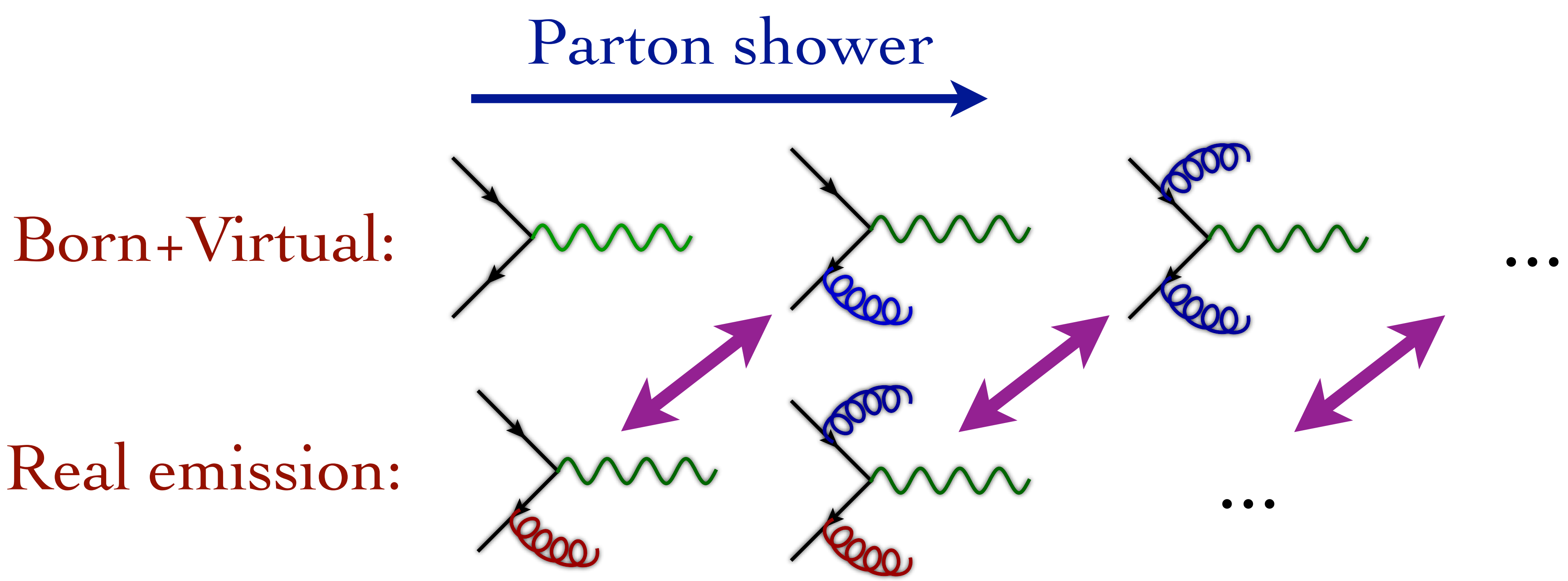
- Terms in brackets are finite. They can be integrated separately and in a 4 space-time dimensions.

MadFKS

- The counterterm $G(\overline{\Phi}_{m+1})$ must satisfy these properties:
  - ➡ Having the same poles and residues as $R(\Phi_{m+1})$
  - ➡ Numerically well-behaved (*i.e.* smooth)
  - ➡ Analytically integrable over the emitted particle d-dimensional phase-space, $\int d^d\Phi_1 G(\overline{\Phi}_{m+1})$, yielding the same poles in ε as the virtual contribution, but with residues of opposite in sign.
- Must identify and isolate each divergent splitting and devise such counterterm for each. The FKS formalism, unlike the CS-Dipoles, uses phase-space partitioning based on the collinear configurations.
- MadFKS implements this method in a fully automatic and process-independent way, while exploiting the symmetries of the process.

Matching to PSMC

- When matching NLO predictions to Parton Showers Monte-Carlo, one faces double-counting issues:



- And also part of the virtual contribution is double counted through the definition of the Sudakov factor Δ

- Two ways out have been proposed: POWHEG and MC@NLO

(A)MC@NLO

- One can compensate for this double-counting considering MC counterterms which are defined the $\mathcal{O}(\alpha_s)$ contribution of the PS to go from the Born (*m*)-body to the (*m+1*)-body configuration.

$$\frac{d\sigma^{\mathrm{NLOwPS}}}{d\mathcal{O}} \sim \int \left[d\Phi_m \left(B + (\int_{\mathrm{loop}} V + \int d\Phi_1 G) + \int d\Phi_1 (MC - G) \right] I_{PS}^{(m)}(\mathcal{O}) + \left[\int d\Phi_{m+1} \left(R - MC \right) \right] I_{PS}^{(m+1)}(\mathcal{O})$$

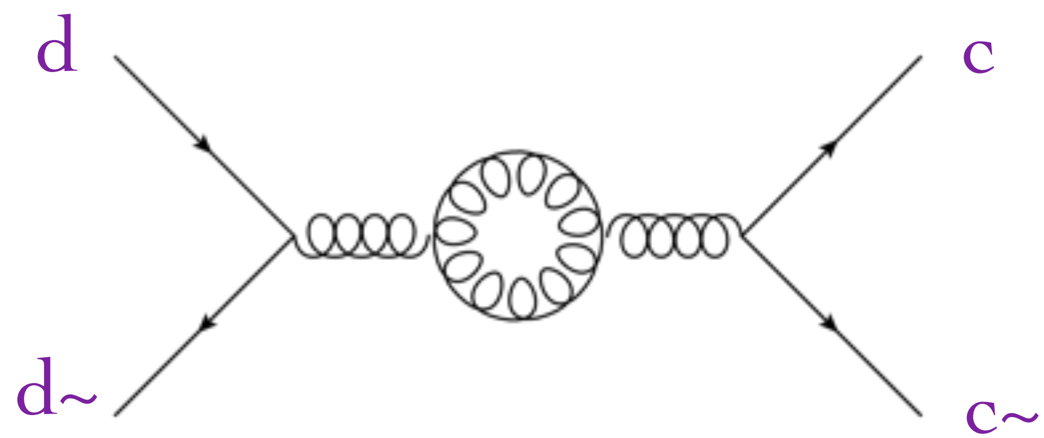
with $I_{PS}^{(n)}(\mathcal{O})$ the Parton Shower operator for the observable $\mathcal{O}$ on a parton level n-body configuration.

- The MC counterterms can be written in a process-independent way, so that the matching procedure is automated in aMC@NLO!

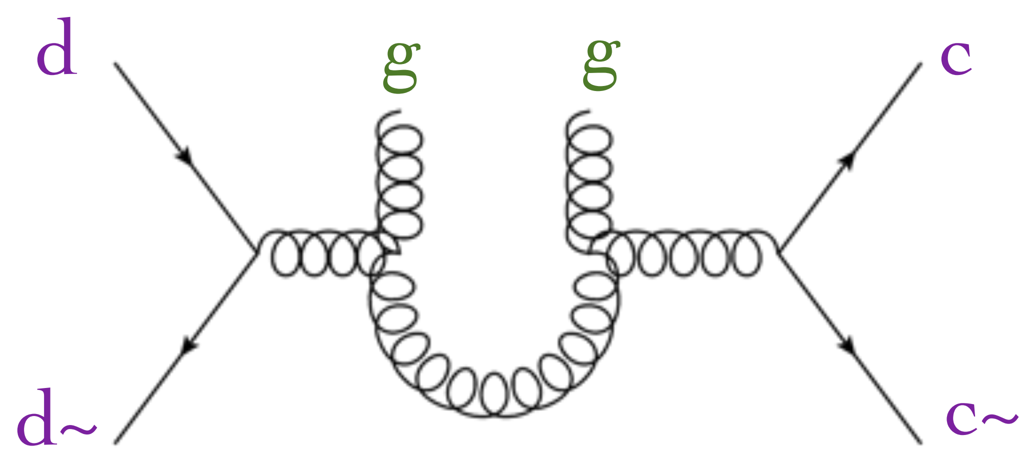
- The prediction obtained has the ME behavior in the hard emission region $I_{PS}^{(m+1)}(\mathcal{O}) \sim 1$, $I_{PS}^{(m)}(\mathcal{O}) \sim 0$, $MC \sim 0$ and the PSMC one in the soft region where $MC \sim R$, perfect!

Generating loop diagrams

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



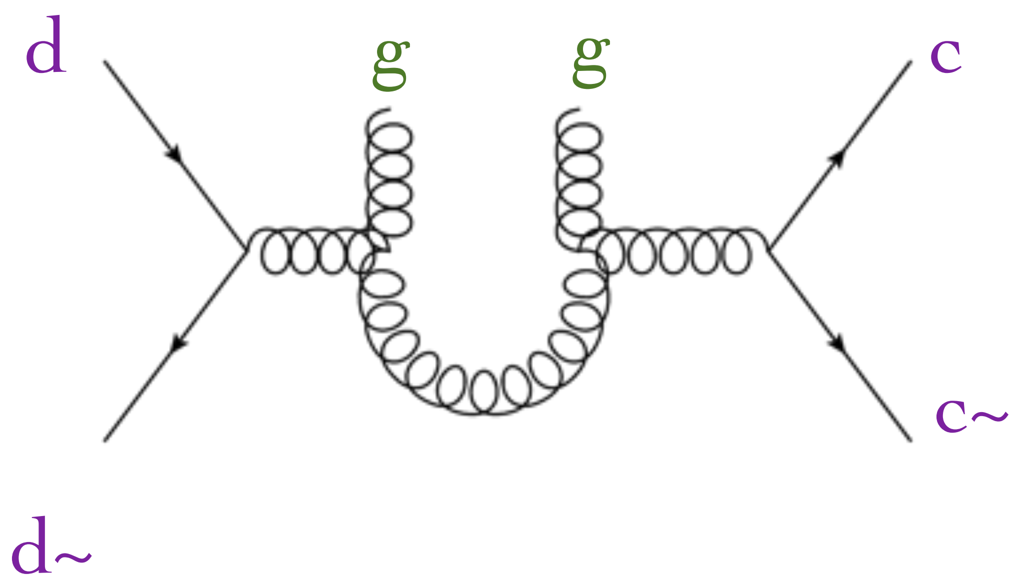
- For this one you necessarily need to generate the born process with the additional two L-cut particles being gluons!



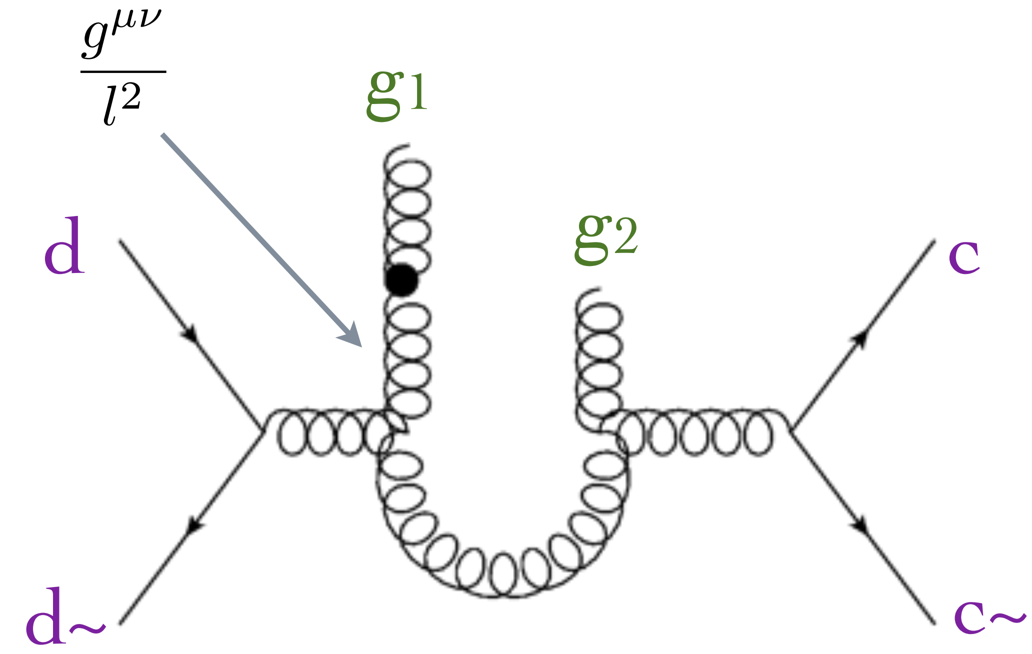
- Loops including a u-quark were already generated with d d~ ➞ c c~ u u~, so you can speed up the d d~ ➞ c c~ g g generation forbidding u in the loop!

Generating loop diagrams

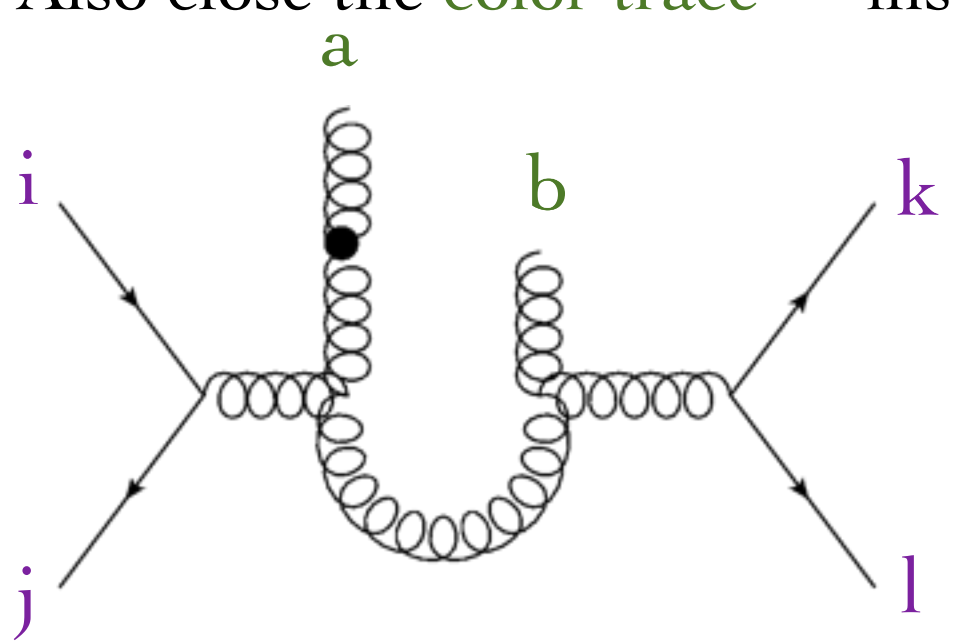
- It is not yet what we want, we are missing the l-cut propagator



⇨



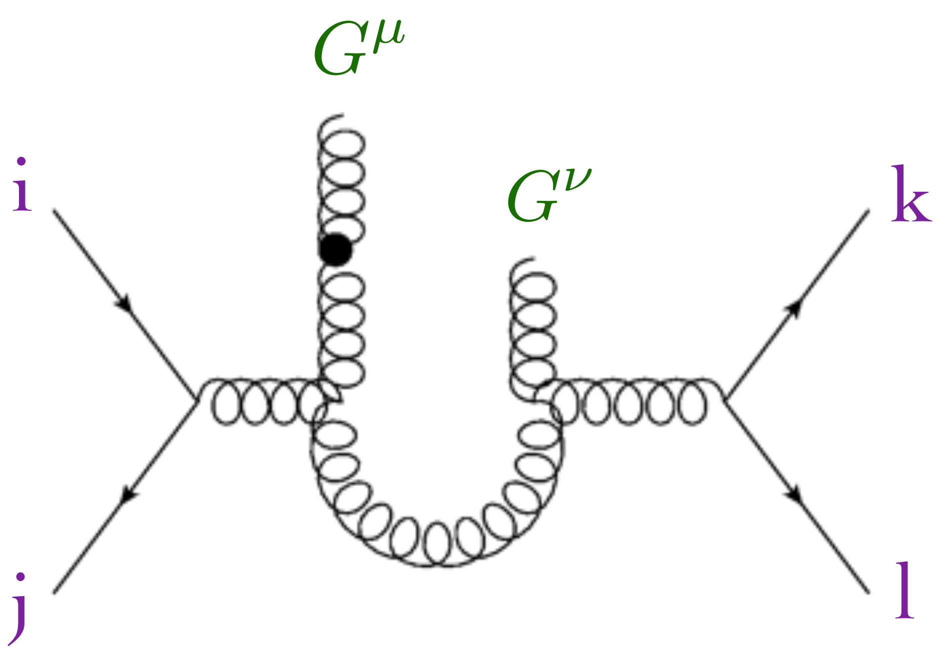
- Also close the color trace → insert a δ^{ab} or δ^{ij} to the color chain



⇨ $$T^{r}_{ij} f^{ars} f^{sbt} T^{t}_{kl} \delta^{ab}$$

Generating loop diagrams

- 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 OPP!

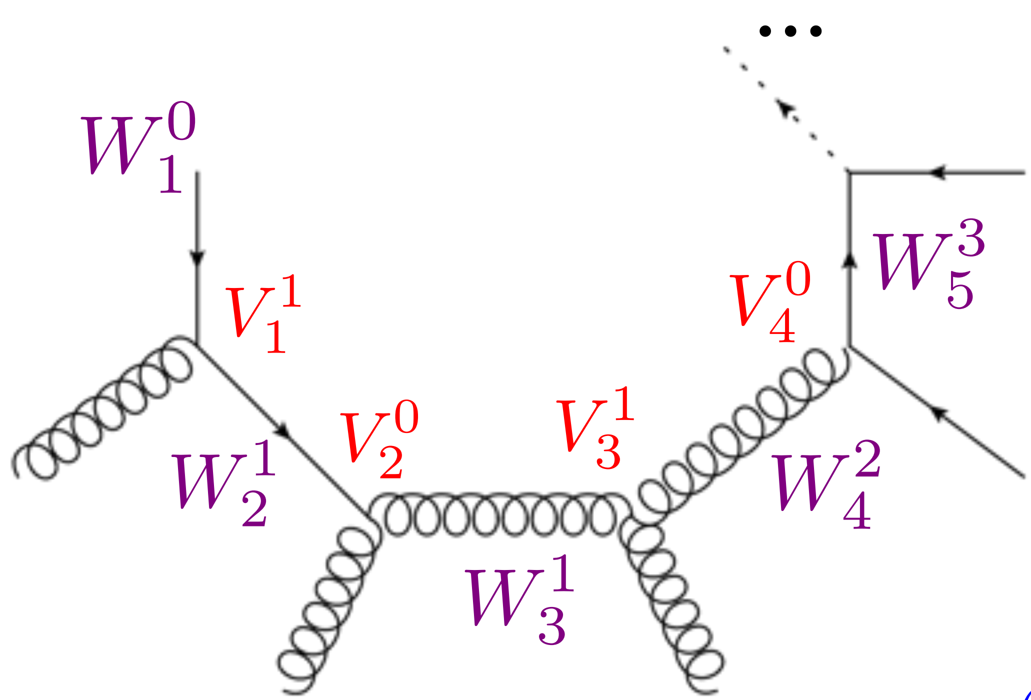
Open-Loops

[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)



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

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

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

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

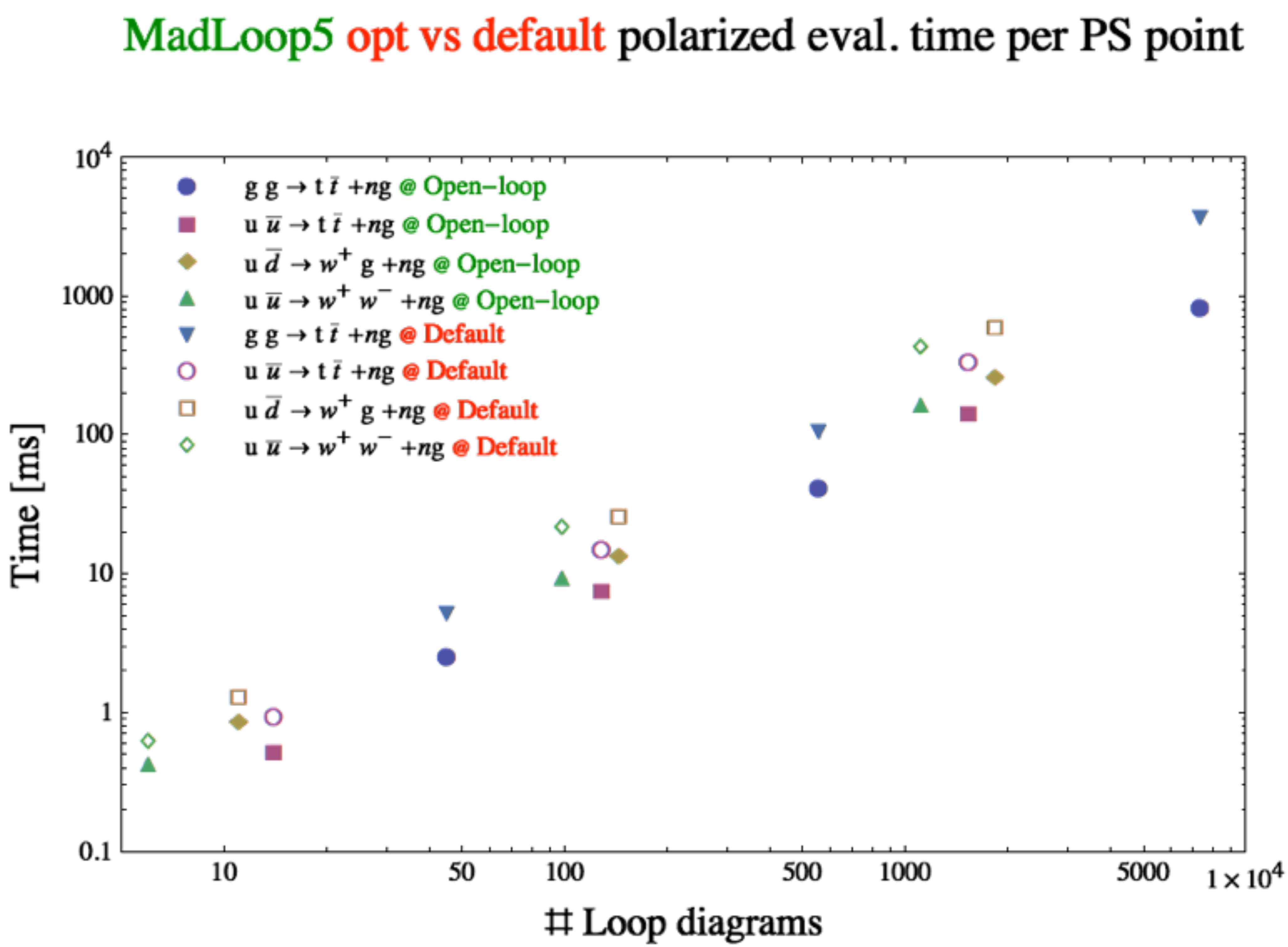
$$W_4^{(1)} = V_3^{(1)} W_2^{(1)} = (v_3^1 l + v_3^0) v_2^0 (v_1^1 l + v_1^0) w_1^0$$

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

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



FKS vs CS dipoles

N^2 vs N^3

- CS uses soft singularities to organize the subtractions :
  - ➔ Three-body kernels, so naive n^3 scaling
  - ➔ Each subtraction term has a different kinematics
  - ➔ All subtraction terms must be subtracted to

- MadFKS, based on the collinear structures :
  - ➔ The majority of the subtractions can be grouped together. *Ex:* The 2 → N gluons process as 3 subtractions ∀N
  - ➔ Soft and collinear counter-terms can be defined as to have the same kinematics so that the subtraction term is unique.
  - ➔ The collinear structure is better suited to existing formalisms for NLO parton shower matching.