Recent Developments in MadGraph5_aMC@NLO

Valentin Hirschi

Université catholique de Louvain

ETH zürich

LHCTheory meeting

22th March 2017

Roadmap

- Interface to **Pythia8** and **MadAnalysis5**
- **COLLIER** interface to MadLoop
- **Loop-induced** processes at **NLO**
- Automated mixed **NLO QCD+EW**
- **SUSY @ NLO QCD**, OS subtraction
- Ad: **Plugins** in MG5_aMC
- A particular take on long-term plans...

PYTHIA8 INTERFACE

Pending publication (if ever :/): V.H., O. Mattelaer, S. Prestel

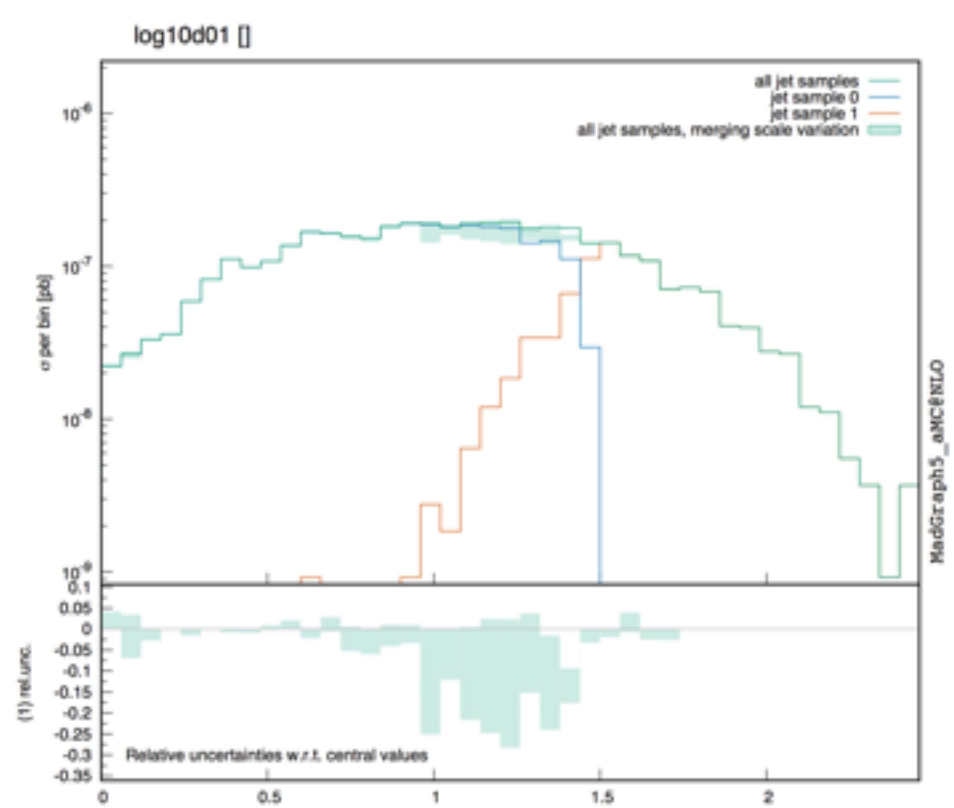
Pythia8 installation and use:

```
MG5_aMC > install pythia8
MG5_aMC > install mg5amc_py8_interface
ProcOuput > shower pythia8 run_01
[...]/ProcOuput/Cards/pythia8
```

- Supports CKKW-L for LO merging
- Merging systematics computed on-the-fly
- Parallelization of Pythia8 runs
- Merging systematics weights propagated through HEPMC event files
- Ability easily output HEPMC events to a FIFO file
- Do-it-all Pythia8 driver.

MLM p p > Z + {0,1}j

```
Cross-section :   1535 +- 4.319 pb
Nb of events :  10000
Pythia8 merged cross-sections are:
 > Merging scale = 10     : 653.9      +/- 1.7     [pb]
 > Merging scale = 20     : 698.42     +/- 1.7     [pb]
 > Merging scale = 30     : 712.55     +/- 1.7     [pb]
 > Merging scale = 40     : 709.02     +/- 1.7     [pb]
 > Merging scale = 50     : 706.56     +/- 1.7     [pb]
```



- **No excuse** anymore for sticking to Pythia6!

MadAnalysis5 interface

V.H., B. Fuks

MA5 installation and use:

`MG5_aMC > install madanalysis5` (no longer requires root)

```
/------------------------------------------------------------------------------\
| 1. Choose the shower/hadronization program:               shower = OFF        |
| 2. Choose the detector simulation program:             detector = Not installed |
| 3. Run an analysis package on the events generated:    analysis = MADANALYSIS_5 |
| 4. Decay particles with the MadSpin module:             madspin = OFF        |
| 5. Add weights to events for different model hypothesis: reweight = OFF       |
\------------------------------------------------------------------------------/
```

- Implemented both for LO and NLO matched.
- Independent control on parton-level, hadron-level and recasting analysis
- One can bypass HEPMC and do the analysis directly from FIFO files.
- Analysis cards automatically generated and tailored to the process of interest

```
/--------------------------------------------------------------\
| 1. param                : param_card.dat                     |
| 2. run                  : run_card.dat                       |
| 3. pythia8              : pythia8_card.dat                   |
| 4. madanalysis5_parton  : madanalysis5_parton_card.dat       |
| 5. madanalysis5_hadron  : madanalysis5_hadron_card.dat       |
\--------------------------------------------------------------/
```

COLLIER

COLLIER IN MADLOOP

COLLIER from A. Denner, S. Dittmaier, L. Hofer [arXiv:1604.06792]

Reminder: list of other loop reduction tool interfaced:

`CutTools, PJFry++, IREGI, Golem95, Samurai, NINJA`

Automatic COLLIER installation and use:

```
MG5_aMC > install collier

#MLReductionLib
6|7|1
```

COLLIER is a mature code, featuring the following improvements:

- Improved stability by expansions around zero-Grams.
- Fastest algo. and implementation of tensor integral reduction.
- Unlimited number of loop propagators and integrand rank.
- Ability to numerically handle logs from small masses.
- Ability to provide separately IR and UV pole residues.

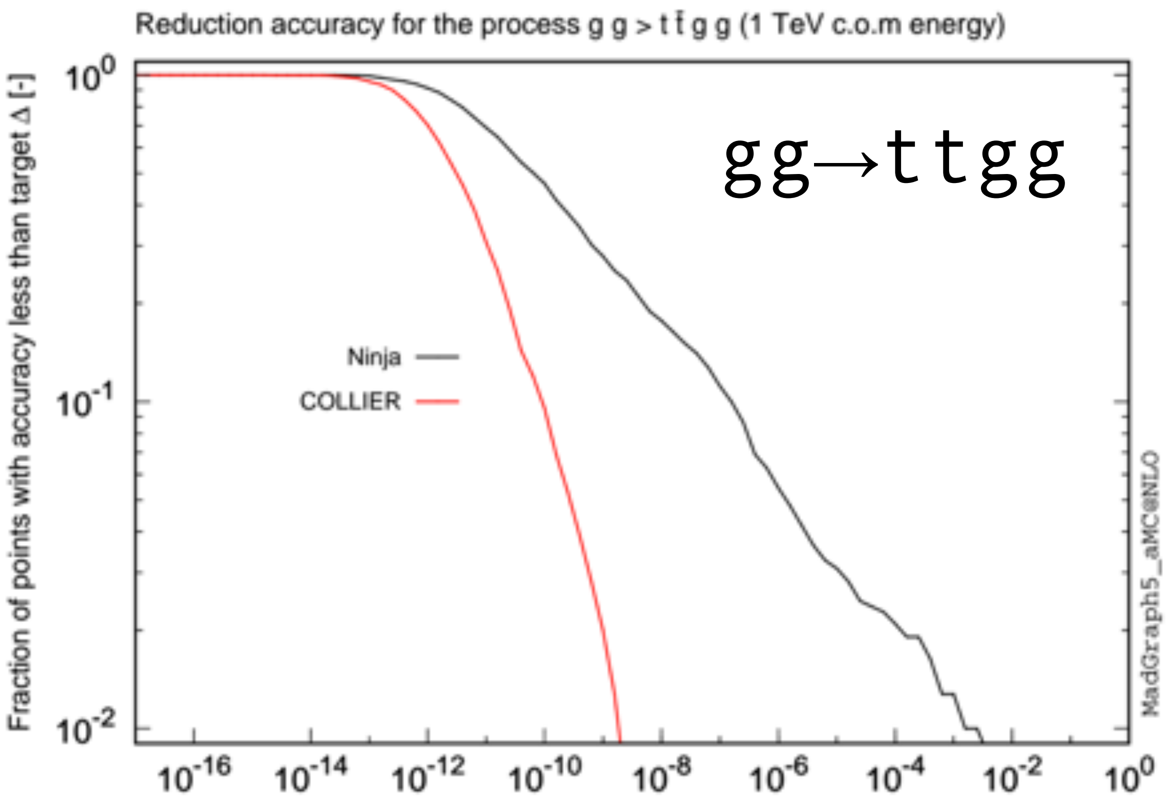
COLLIER was helpful for the EFT Spin-2 NLO computations presented in:

G. Das, C. Degrande, V.H., F. Maltoni, H-S Shao, [arXiv:1605.09359]

Interface validated, public and profiled too.

COLLIER Stability

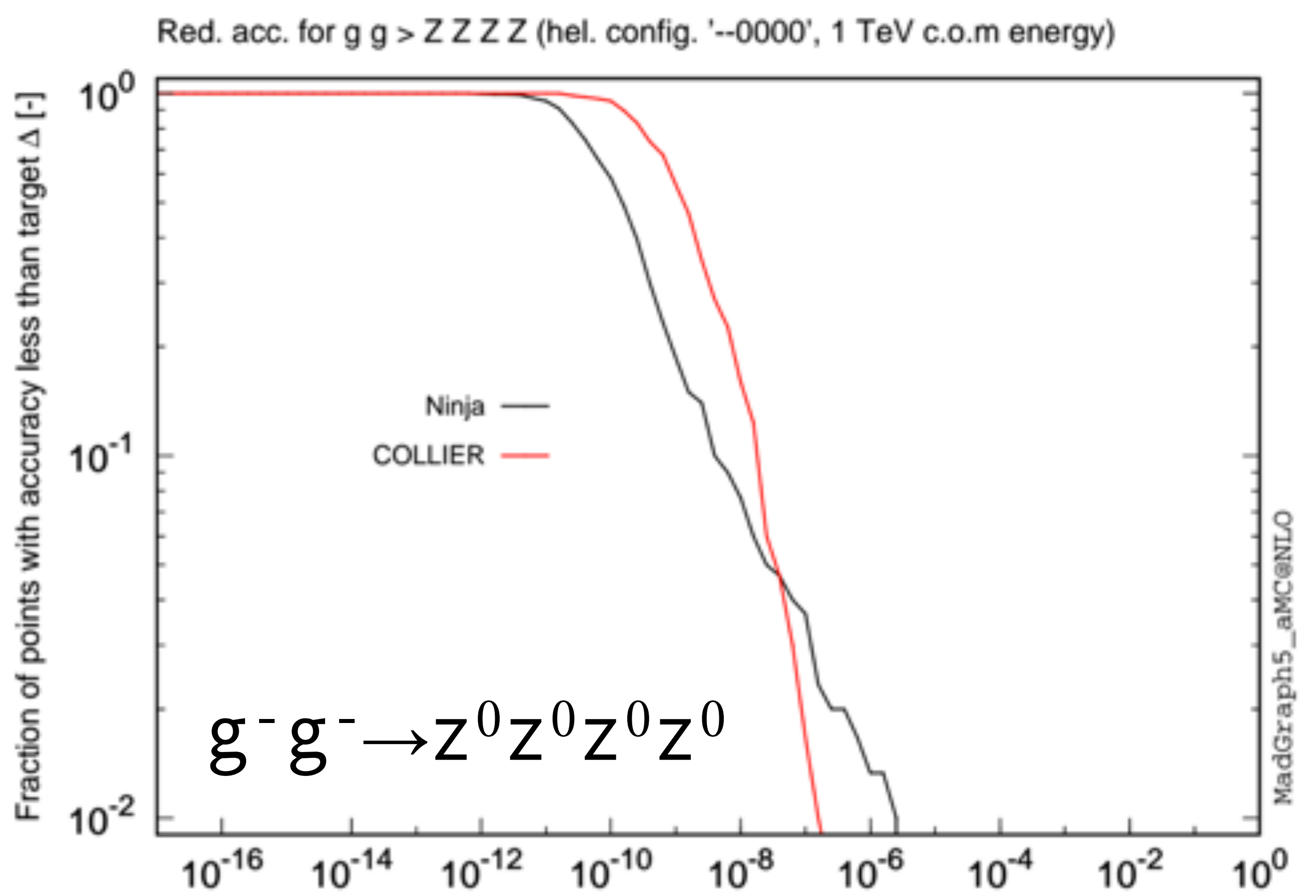
Related to V.H., T. Perraro [arXiv:1604.01363]



Indeed the most stable option of all

How much so mostly depends on:

- multiplicity
- **loop numerator rank.**



However still unclear how much more stable it is close to IR limits :

Probably little improvements, if any.

Quad. prec. still necessary

COLLIER Speed

Related to V.H., T. Perraro [arXiv: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)
COLI, no global cache	0.90	15.7	620	1656
CUTTOOLS reduction time	1.3	23.2	330	301
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)
COLI, no global cache	2.8	64	1820
CUTTOOLS reduction time	4.1	59	1080
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	57	1.8
COLI (no global cache)	1.9	65	2.5 (2.6 no local)
COLLIER/ NINJA	1.3	2.9	5.6

COLLIER provides its own stability test.

→ Needs:

`COLLIER/NINJA > 2`

For Ninja to really be faster in production.

Integrand-level (Ninja) reduction faster for large multiplicities

Difference in speed marginal for most processes.

→ `#MLReductionLib 6|7|1`

Loop-Induced at NLO

Loop-Induced at NLO

Ongoing collaborative effort: V.H., O. Mattelaer, F. Maltoni, E.Vryonidou, N. Kauer, A. Shivaji, M.K.Mandal, ...

Two avenues for simulating LI at NLO in MG5aMC

- Reweighting approach with O. Mattelaer's module.
- Direct integration in MadFKS

Reweighting **Pros** and **Cons**:

- Easy implementation, development and public distribution
- Requires building an ad-hoc underlying model
- Never truly has systematics under control
- Potentially slower
- Color information corrupted (for matching)

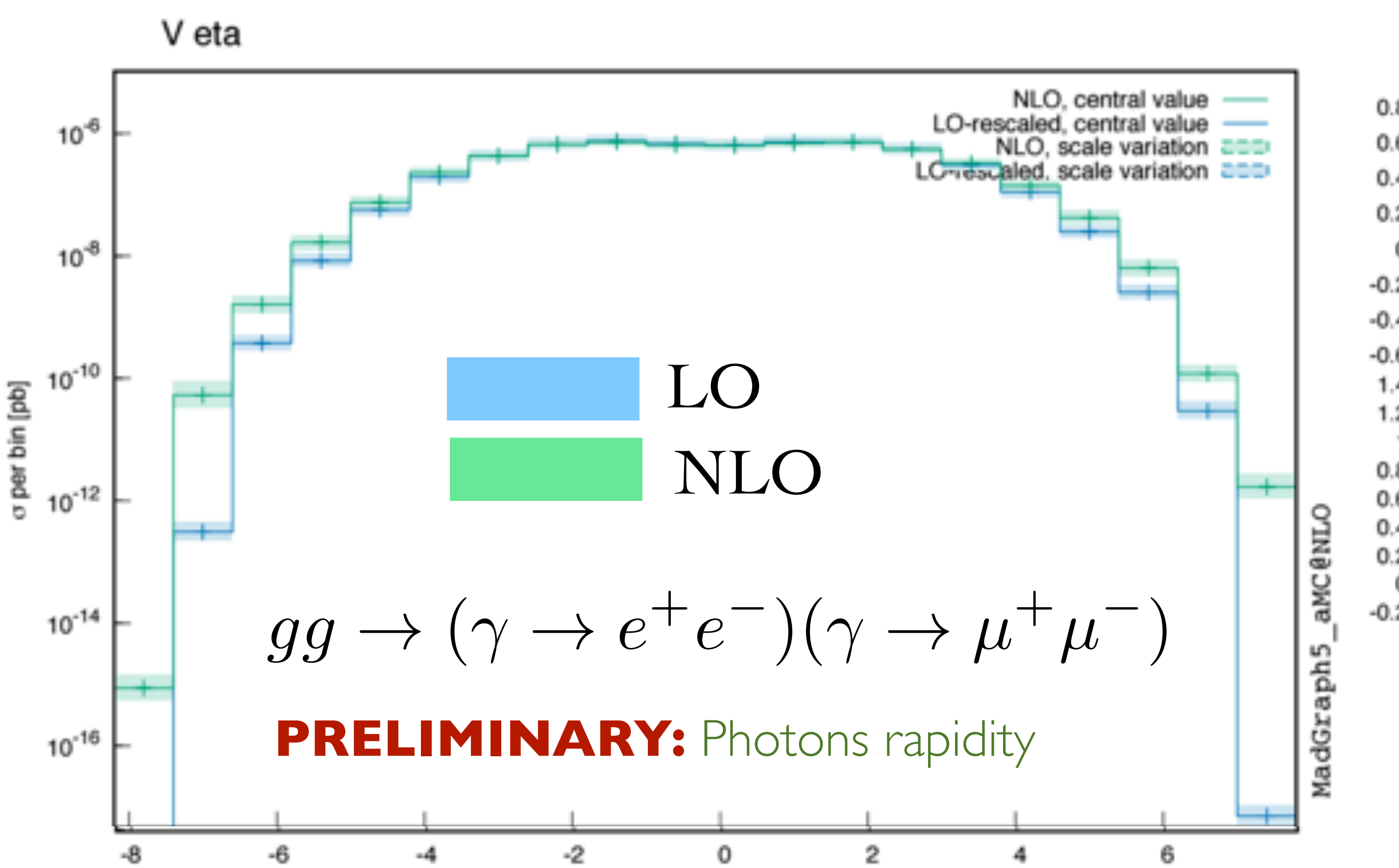
Direct integration **Pros** and **Cons**:

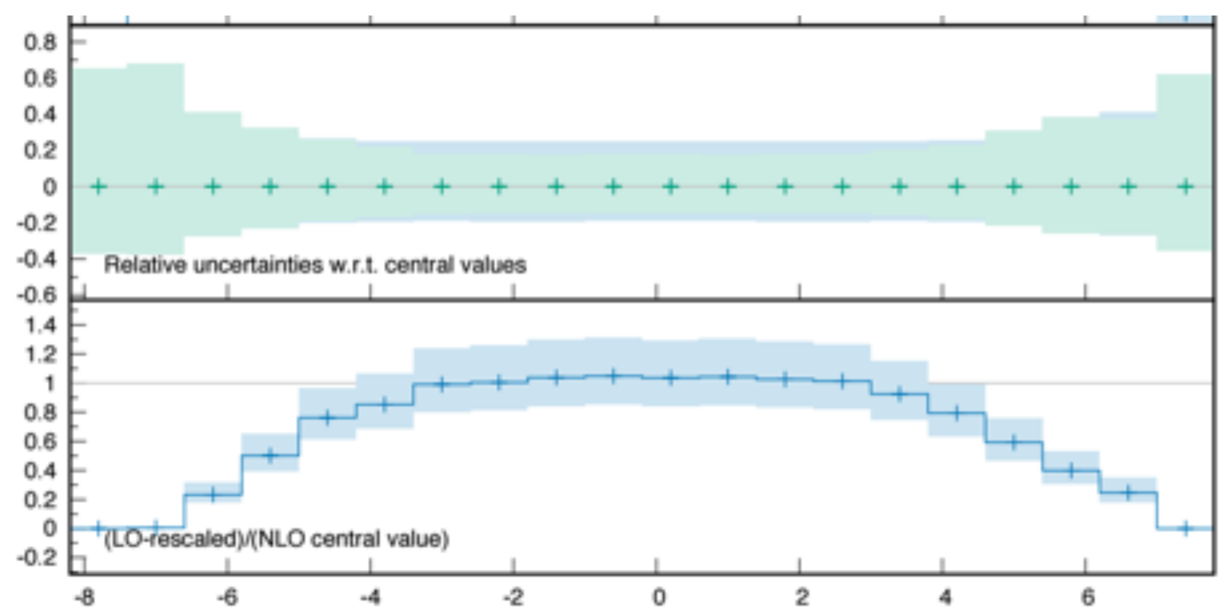
- None of the above drawbacks
- Directly benefits from the virt-tricks, so potentially fast enough.
- **Requires deep improvements in our existing integrator**
- Feasibility study established.

LOOP-INDUCED AT NLO

Feasibility study completed for diphoton decayed:

- 2-loop amplitudes from **VVamp** (A.Manteuffel, L.Tancredi [arXiv:1503.08835])
- Needed ad-hoc parallelization of MadFKS.
- Performed with ad-hoc linking/interface of 2-loop, Born and Reals MEs.
- Threshold for the distance to IR singularities where reals are replaced by local counterterms had to be increased by two 10-folds.





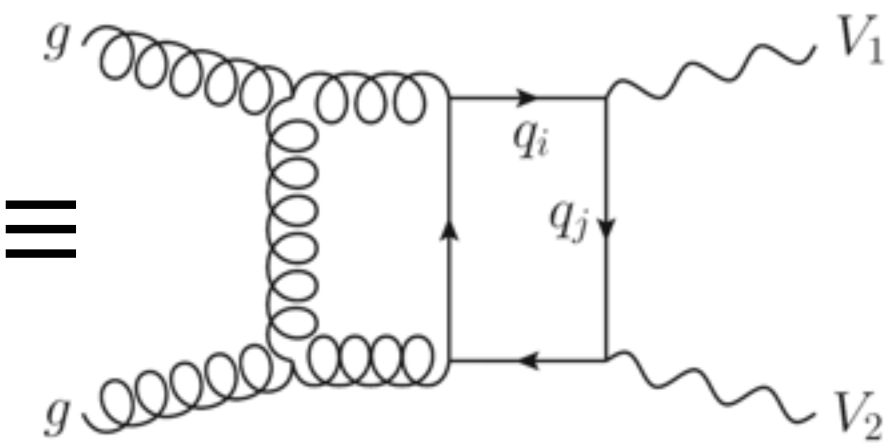
13 TeV. Rescaled curves. K-factor ~ 2

- Flexible implementation of the of 2-loop helicity amplitudes in their covariant form as a **UFO vertex**.

2-Loop Hel. Amplitude as a UFO vertex

$$S^{\mu\nu\rho\sigma}(p_1,p_2,p_3) = a_1 g^{\mu\nu} g^{\rho\sigma} + a_2 g^{\mu\rho} g^{\nu\sigma} + a_3 g^{\mu\sigma} g^{\nu\rho}$$
$$+ \sum_{j_1,j_2=1}^{3} \Big(b^{(1)}_{j_1 j_2}\, g^{\mu\nu}\, p^{\rho}_{j_1}\, p^{\sigma}_{j_2} + b^{(2)}_{j_1 j_2}\, g^{\mu\rho}\, p^{\nu}_{j_1}\, p^{\sigma}_{j_2} + b^{(3)}_{j_1 j_2}\, g^{\mu\sigma}\, p^{\nu}_{j_1}\, p^{\rho}_{j_2}$$
$$+ b^{(4)}_{j_1 j_2}\, g^{\nu\rho}\, p^{\mu}_{j_1}\, p^{\sigma}_{j_2} + b^{(5)}_{j_1 j_2}\, g^{\nu\sigma}\, p^{\mu}_{j_1}\, p^{\rho}_{j_2} + b^{(6)}_{j_1 j_2}\, g^{\rho\sigma}\, p^{\mu}_{j_1}\, p^{\nu}_{j_2} \Big)$$
$$+ \sum_{j_1,j_2,j_3,j_4=1}^{3} c_{j_1 j_2 j_3 j_4} p^{\mu}_{j_1}\, p^{\nu}_{j_2} p^{\rho}_{j_3}\, p^{\sigma}_{j_4},$$

$\equiv$



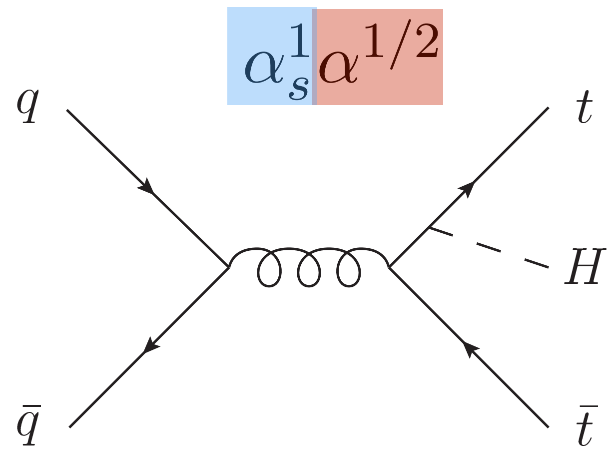

```
GGAA = Vertex(name = 'GGAA',
         particles = [ P.G, P.G, P.A, P.A ],
         color = [ 'Identity(1,2)' ],
         lorentz = [ L.A, L.B, L.C, L.D, L.E,
                     L.F, L.G, L.H, L.I, L.J,
                     L.K, L.L, L.M, L.N, L.O,
                     L.P, L.Q, L.R, L.S, L.T
                     ],
          couplings = {(0,0):C.GGAA_C1, (0,1):C.GGAA_C2, (0,2):C.GGAA_C3, (0,3):C.GGAA_C4, (0,4):C.GGAA_C5,
             (0,5):C.GGAA_C6, (0,6):C.GGAA_C7, (0,7):C.GGAA_C8, (0,8):C.GGAA_C9, (0,9):C.GGAA_C10,
             (0,10):C.GGAA_C11, (0,11):C.GGAA_C12, (0,12):C.GGAA_C13, (0,13):C.GGAA_C14, (0,14):C.GGAA_C15,
             (0,15):C.GGAA_C16, (0,16):C.GGAA_C17, (0,17):C.GGAA_C18, (0,18):C.GGAA_C19, (0,19):C.GGAA_C20
           })
```

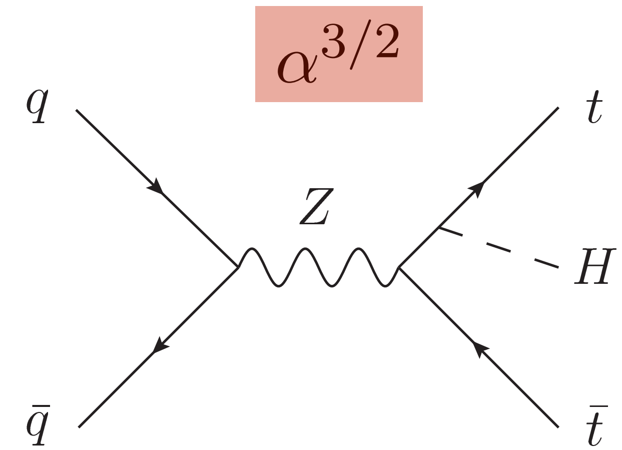
- Allows a tool like MG5_aMC to generate arbitrary 2-loop amplitudes containing this loop (with any decay or vector quantum numbers.)
- The above should be viewed as template for distributing two-loop computations analytical results. UFO extension?

Mixed EW+QCD NLO Computations

Structure of NLO EW-QCD corrections

The ttH case: S.Frixione, V.Hirschi, D. Pagani, H.-S. Shao, M. Zaro [arXiv:1504.03446]

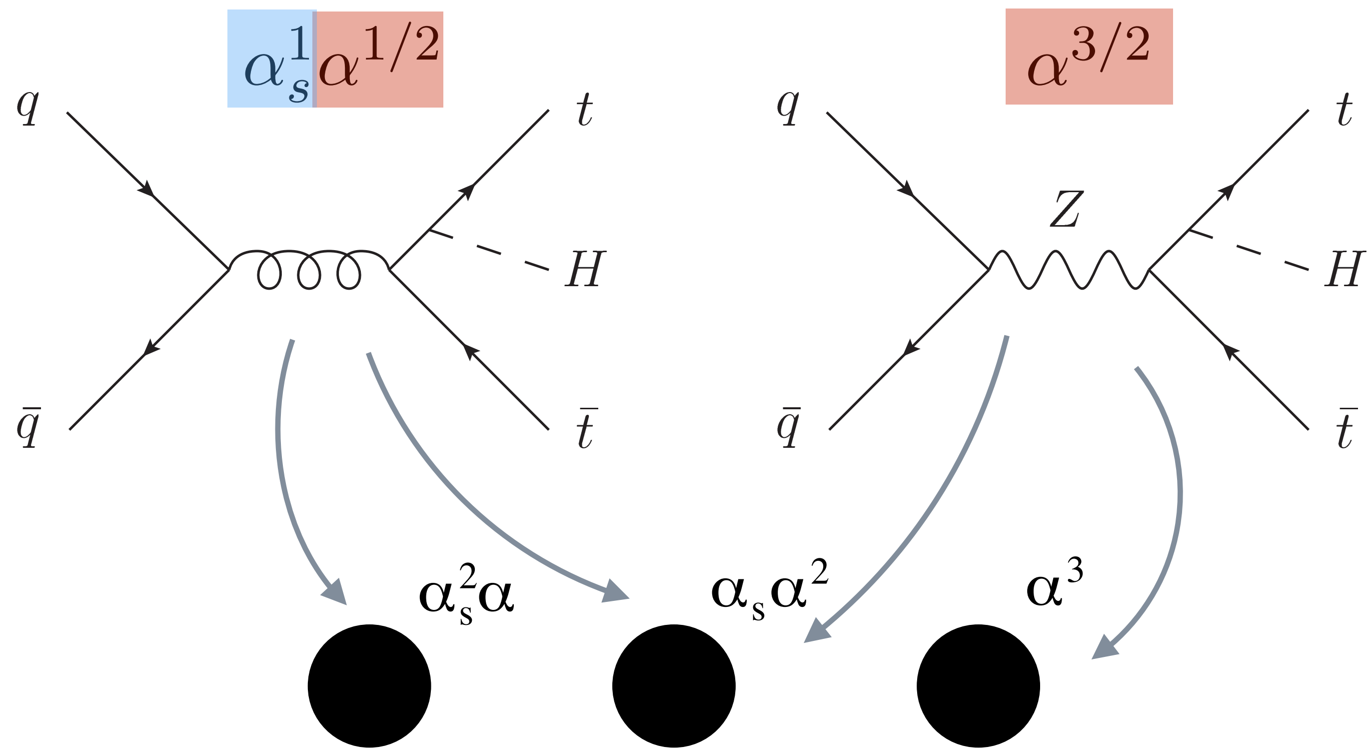




LO	$\alpha_s^2\alpha$	$\alpha_s\alpha^2$	α^3

Structure of NLO EW-QCD corrections

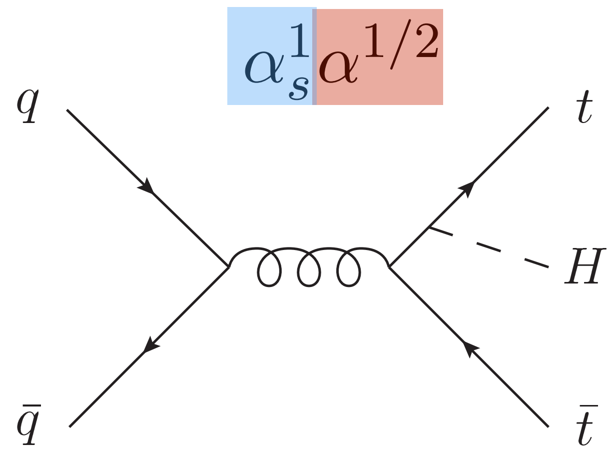
The ttH case: S.Frixione, V.Hirschi, D. Pagani, H.-S. Shao, M. Zaro [arXiv:1504.03446]

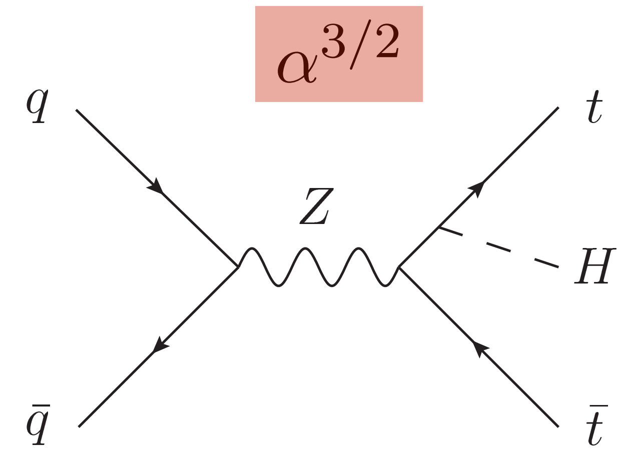


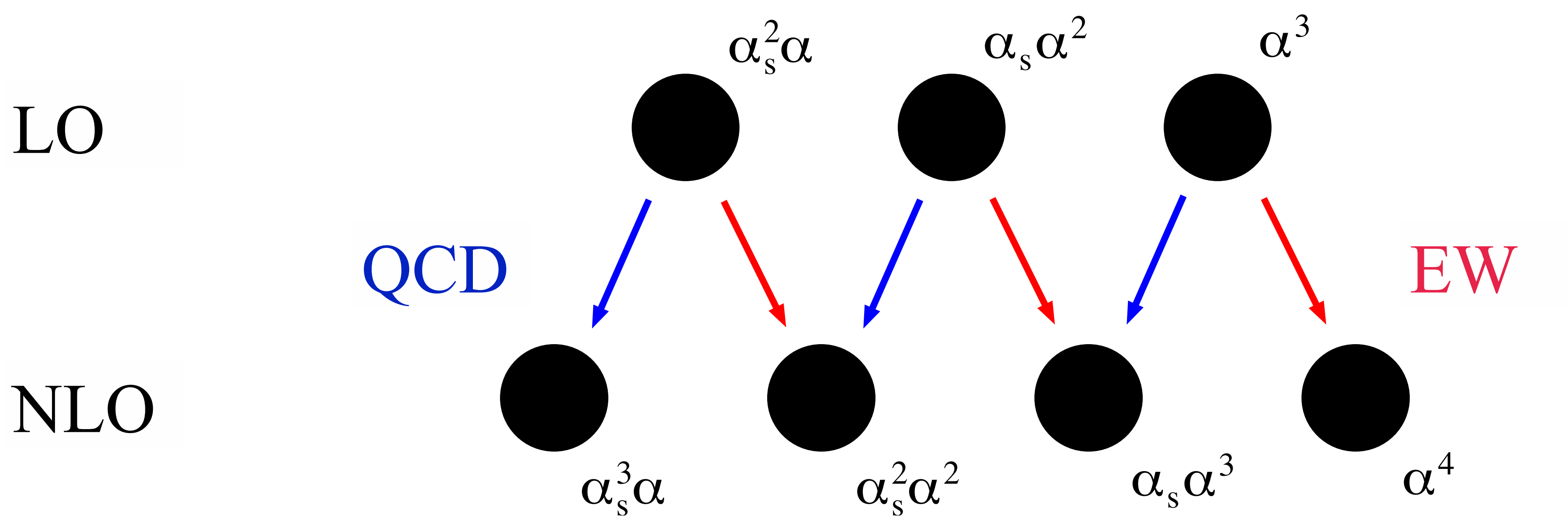
LO

Structure of NLO EW-QCD corrections

The ttH case: S.Frixione, V.Hirschi, D. Pagani, H.-S. Shao, M. Zaro [arXiv:1504.03446]

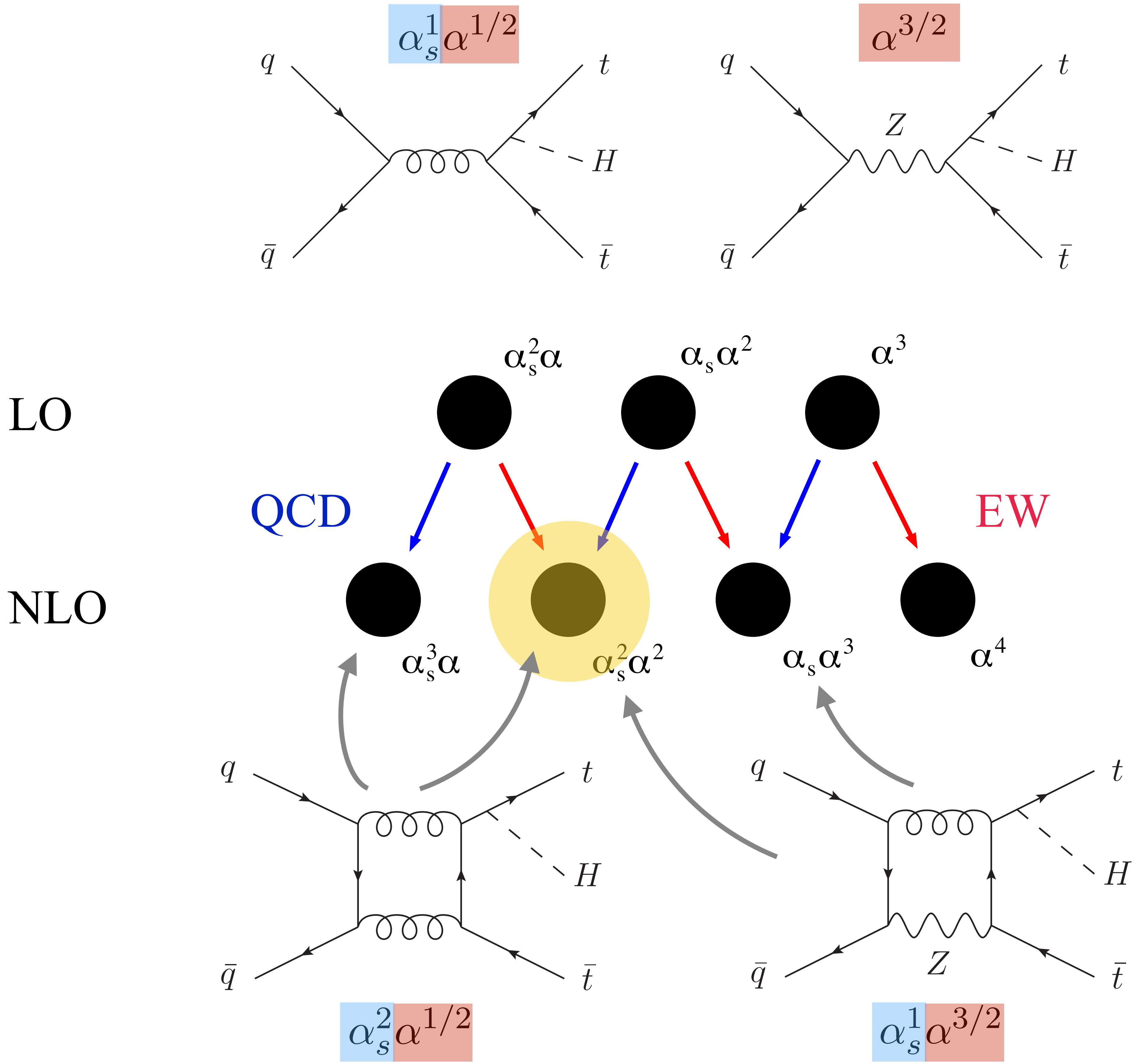






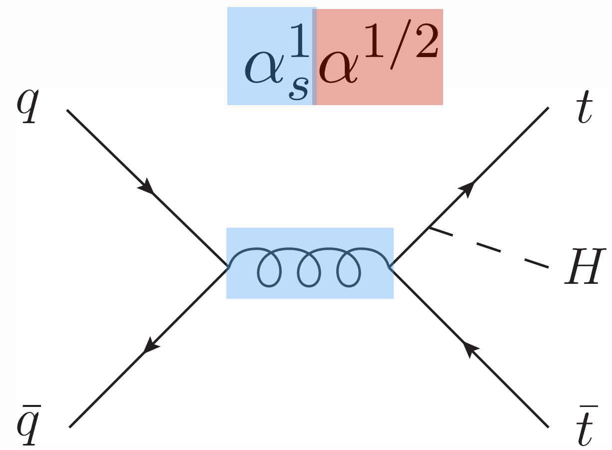
Structure of NLO EW-QCD corrections

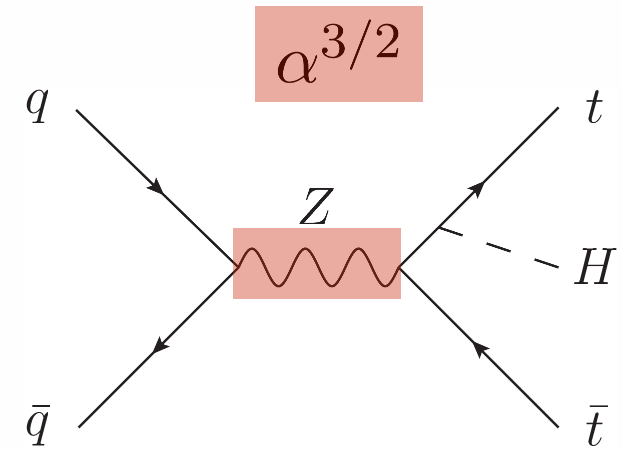
The ttH case: S.Frixione, V.Hirschi, D. Pagani, H.-S. Shao, M. Zaro [arXiv:1504.03446]

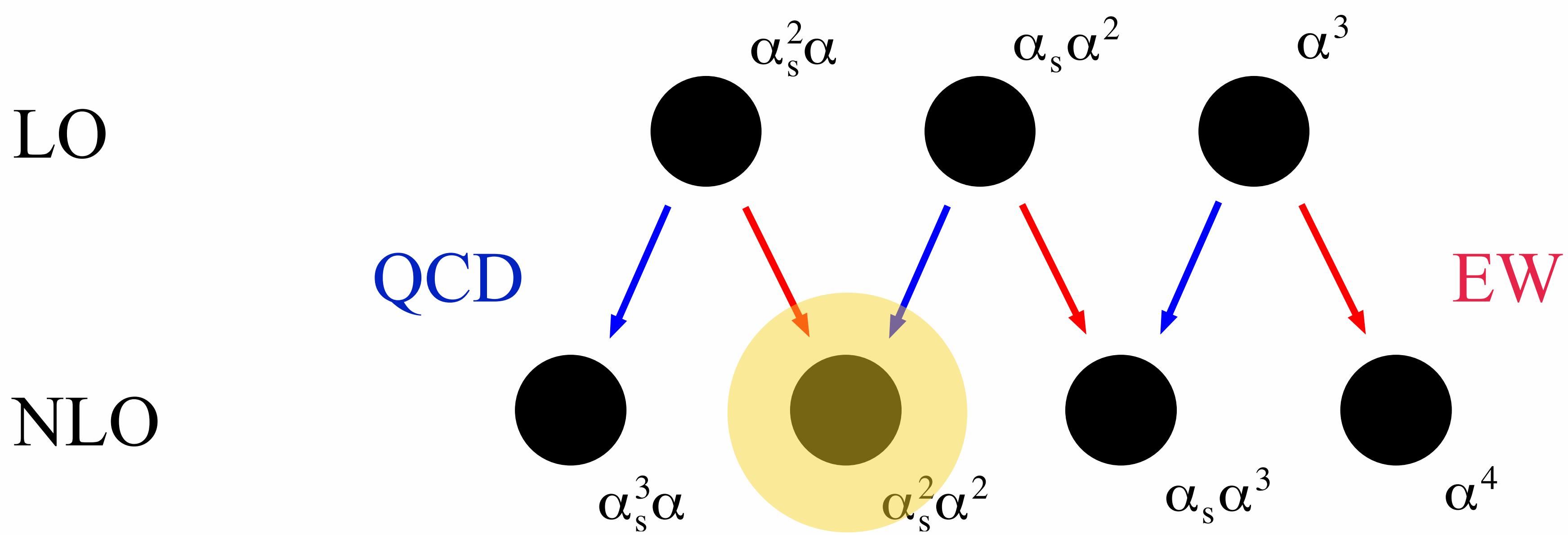


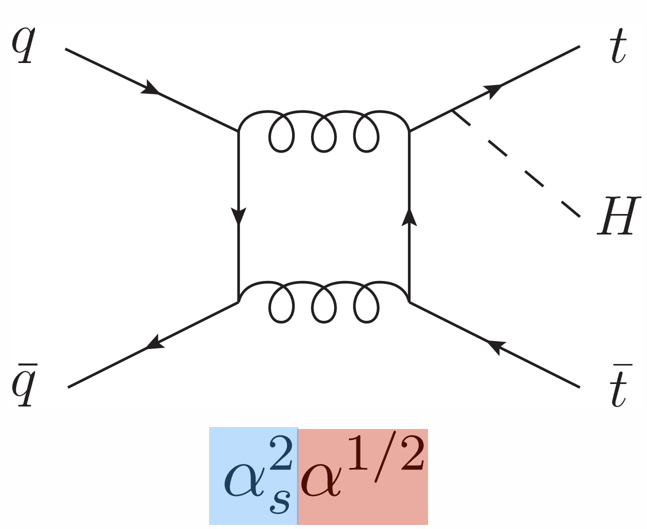
Structure of NLO EW-QCD corrections

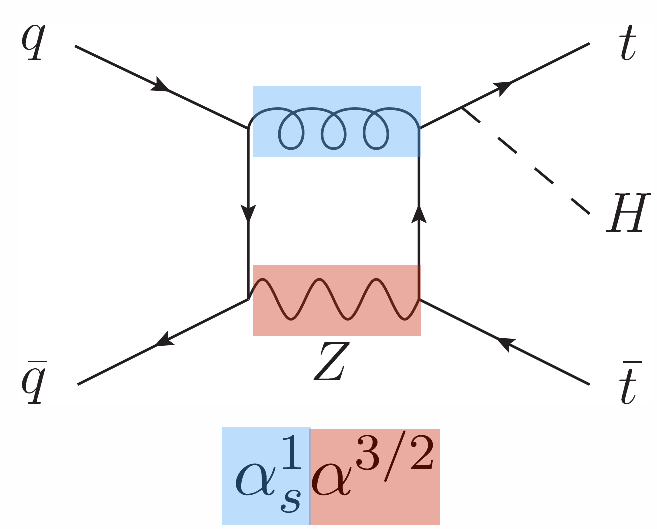
The ttH case: S.Frixione, V.Hirschi, D. Pagani, H.-S. Shao, M. Zaro [arXiv:1504.03446]



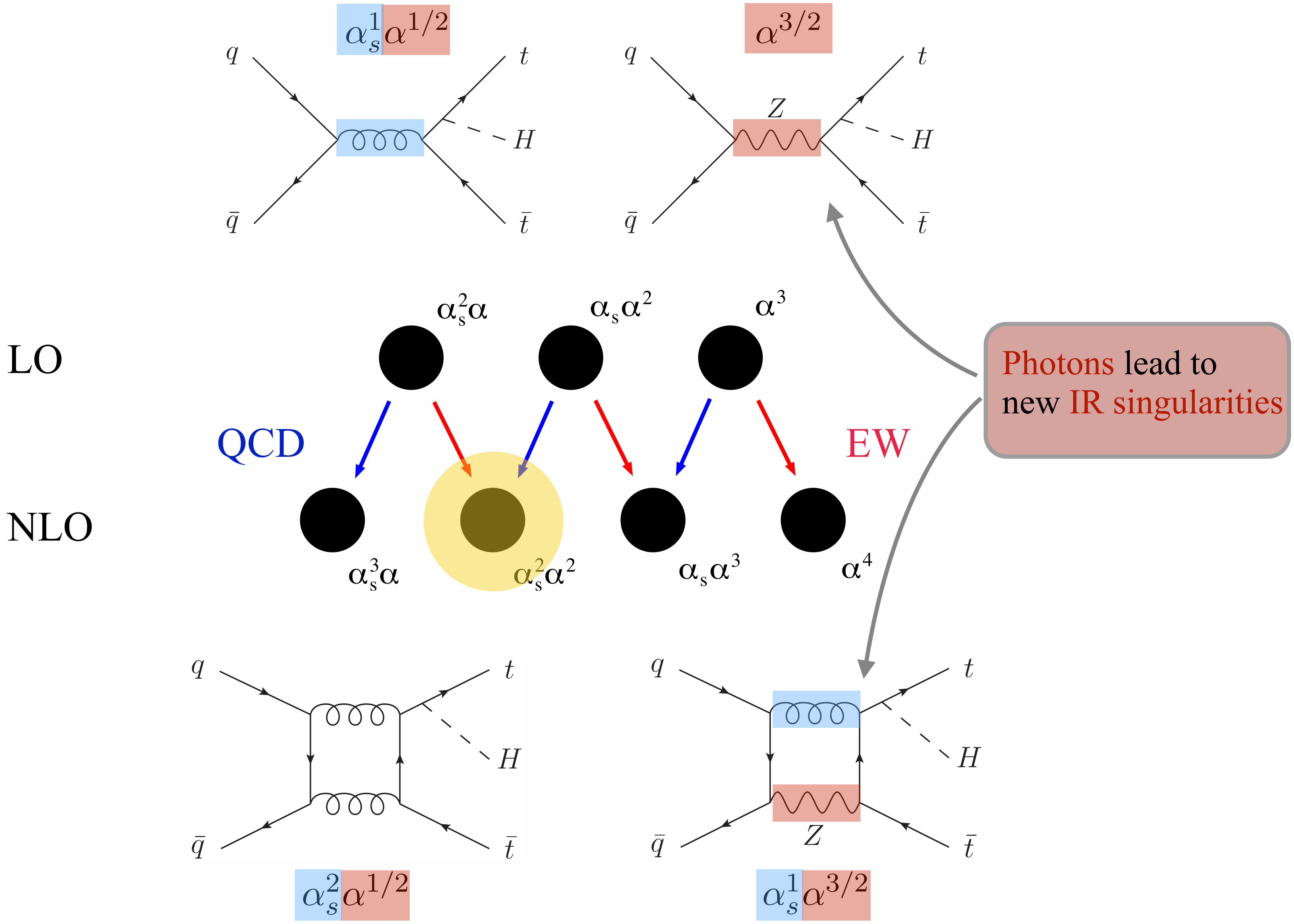






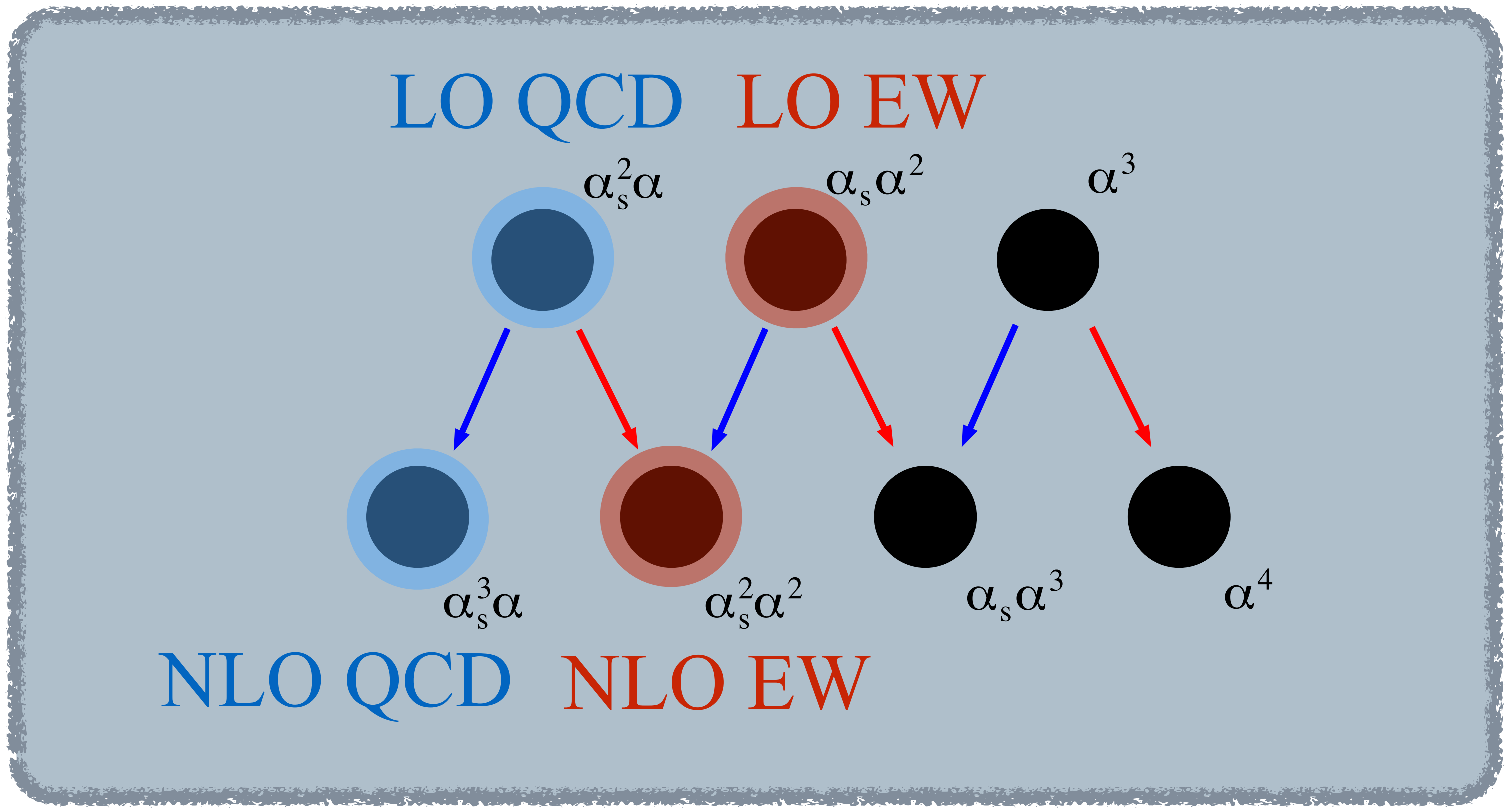


Structure of NLO EW-QCD corrections



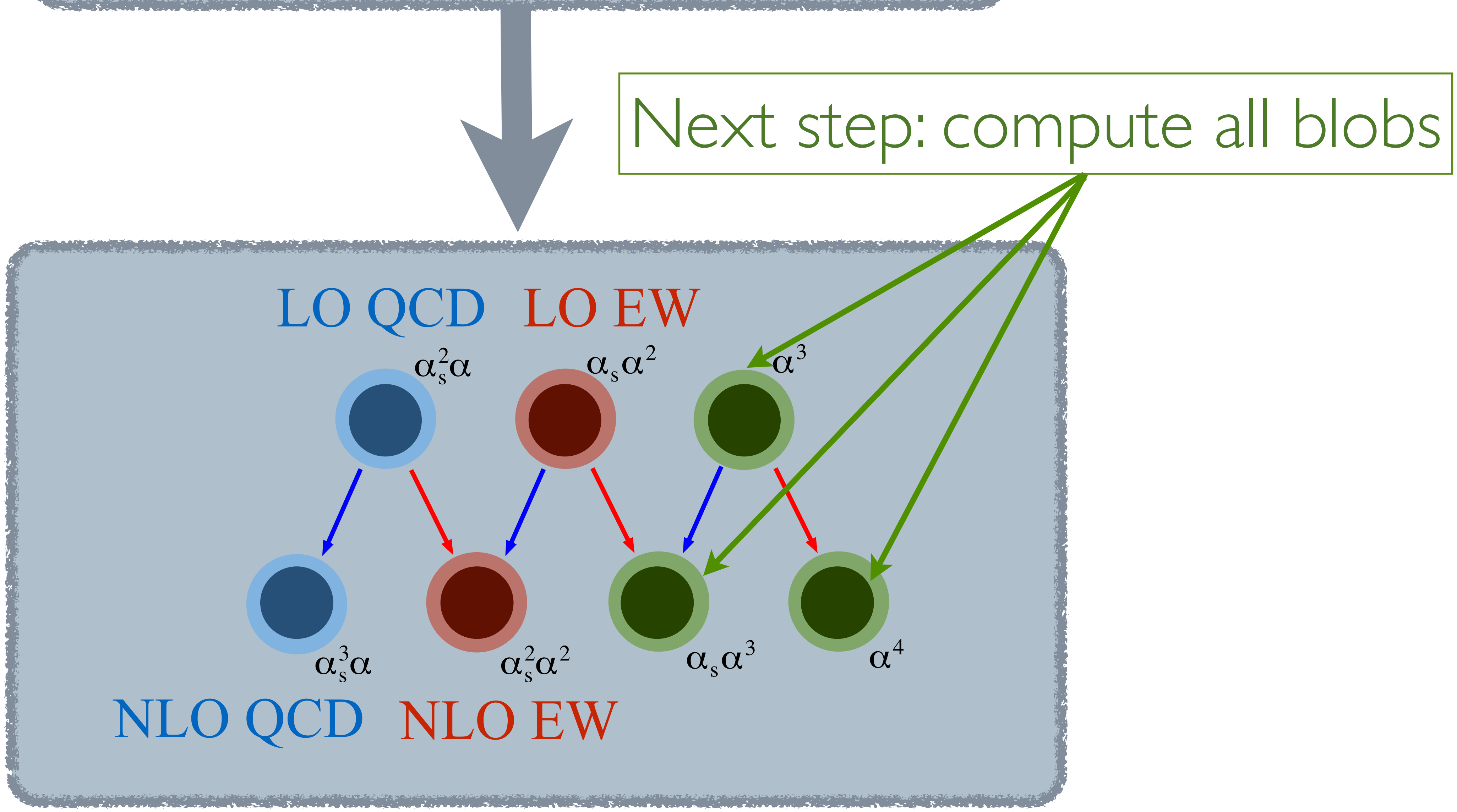
What You See Is What You Get

```
MG5_aMC> define p = p b b~ a
MG5_aMC> generate p p > t t~ h [QCD QED]
MG5_aMC> output ttbarh_QCD_QED
MG5_aMC> launch
```



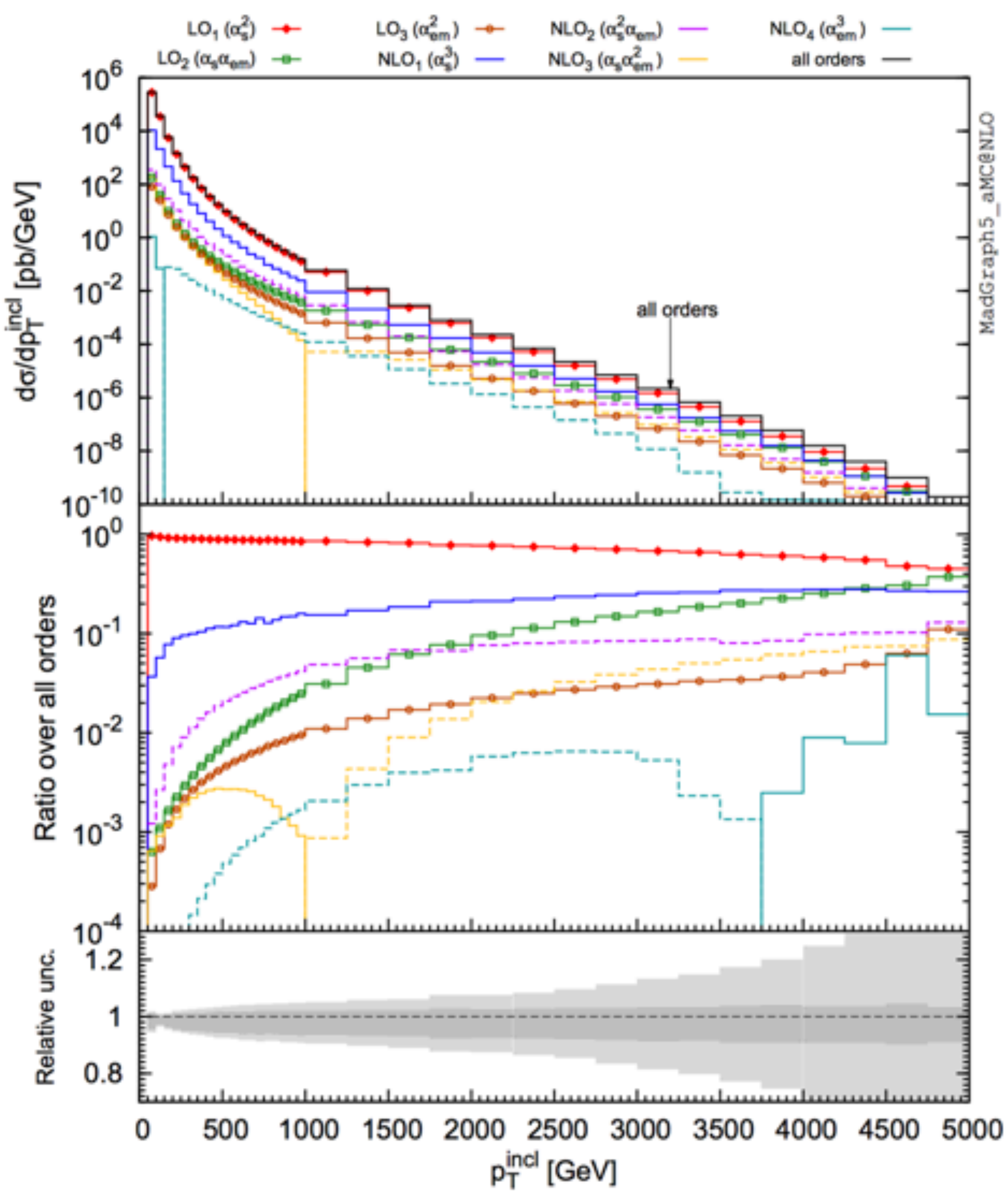
What You See Is What You Get

```
MG5_aMC> define p = p b b~ a
MG5_aMC> generate p p > t t~ h [QCD QED]
MG5_aMC> output ttbarh_QCD_QED
MG5_aMC> launch
```

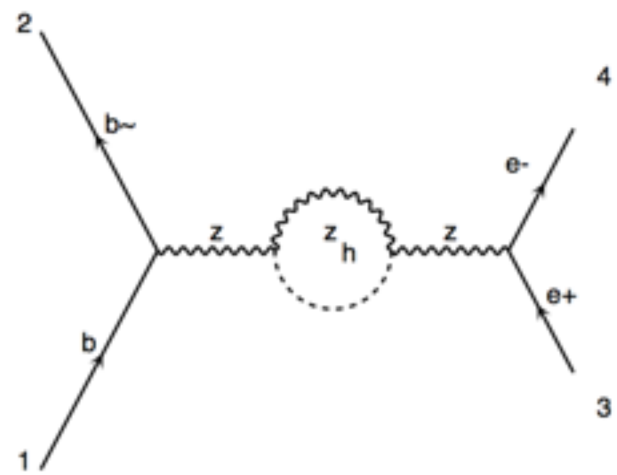


COMPLETE DIJET QCD+EW NLO CORRECTIONS

R. Frederix, S. Frixione, V. H., D. Pagani, H-S.Shao M.Zaro [arXiv:1612.06548]



- All $\mathcal{O}(\alpha_s^m, \alpha^n), m+n=2,3$ contributions to dijet. Use G_μ-scheme
- This process involves the whole particle spectrum of the SM. Yes, even the Higgs!



- Use democratic jets and proposed a novel definition of (anti-)tagged photons
- Necessitated massive computing resources O(weeks), 219 subprocesses
- Pheno conclusion: No significant Sudakov enhancement at LHC13, even at high Pt.

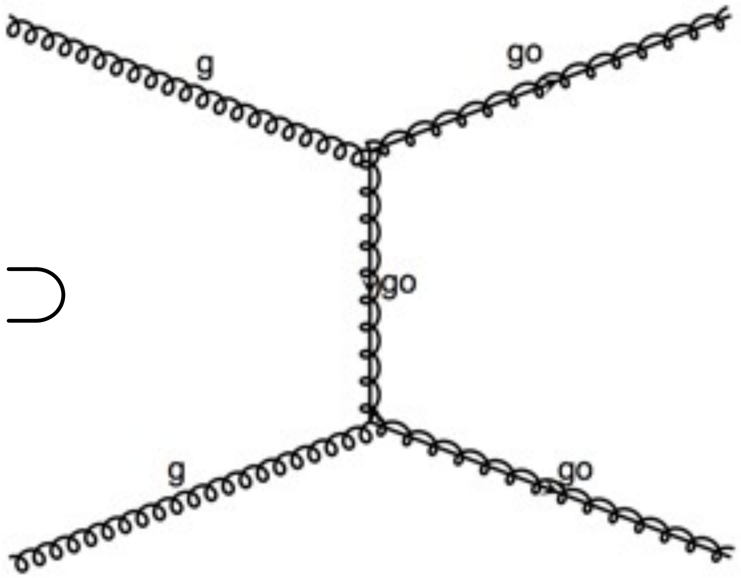
Complete SUSY model @ NLO QCD

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...

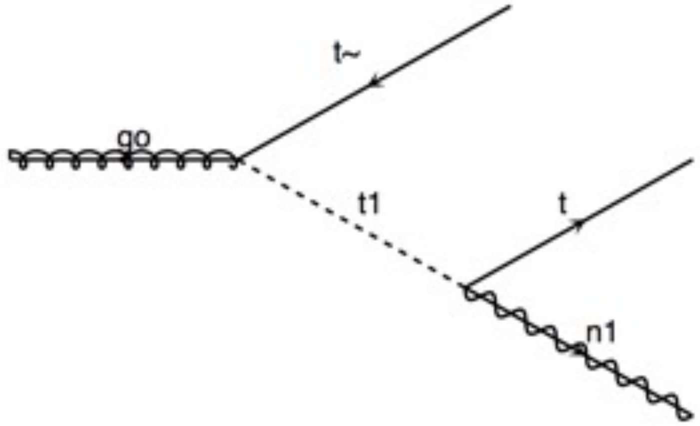
$$\mathcal{L}_{\rm SQCD} = D_\mu \tilde{q}_L^\dagger D^\mu \tilde{q}_L + D_\mu \tilde{q}_R^\dagger D^\mu \tilde{q}_R + \frac{i}{2}\bar{\tilde{g}} \not{D} \tilde{g} - m^2_{\tilde{q}_L} \tilde{q}_L^\dagger \tilde{q}_L - m^2_{\tilde{q}_R} \tilde{q}_R^\dagger \tilde{q}_R - \frac{1}{2} m_{\tilde{g}} \bar{\tilde{g}}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



$$\otimes$$

- ... including the squark decay.

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



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

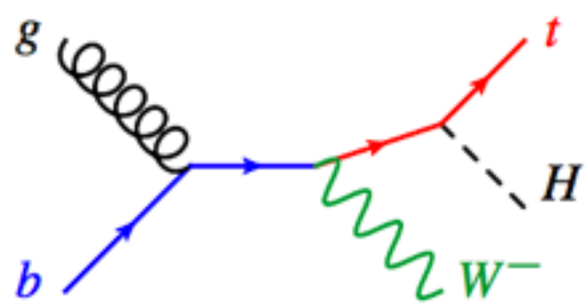
Complete SUSY Model for NLO QCD

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

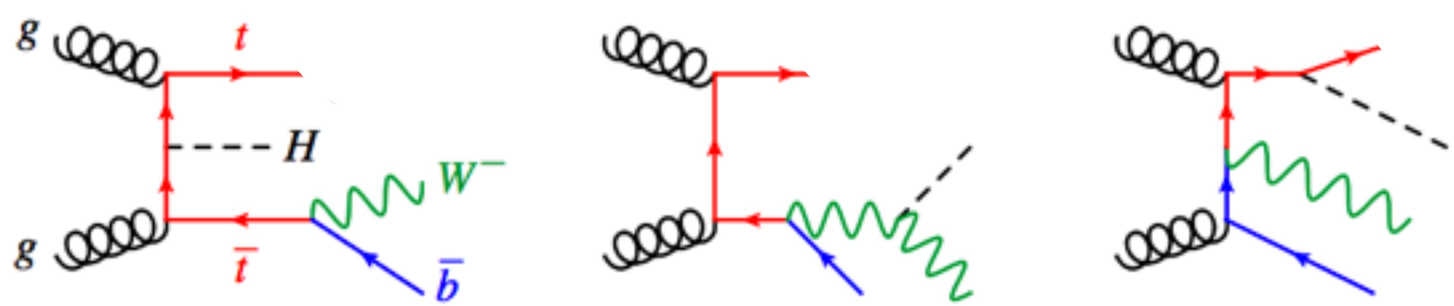
- A key component here is Onshell-Subtraction (OS) in aMC@NLO, which is now available, and was introduced in

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

LO ⊃



NLO (reals) ⊃

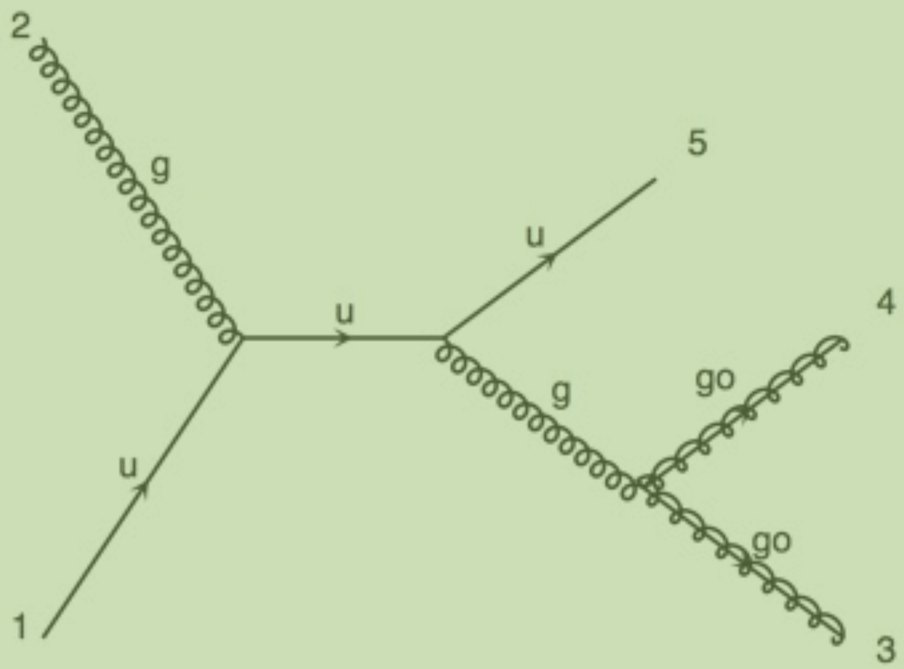


$(\mathcal{A}_{2t})$ $(\mathcal{A}_{1t})$ $(\mathcal{A}_{1t})$

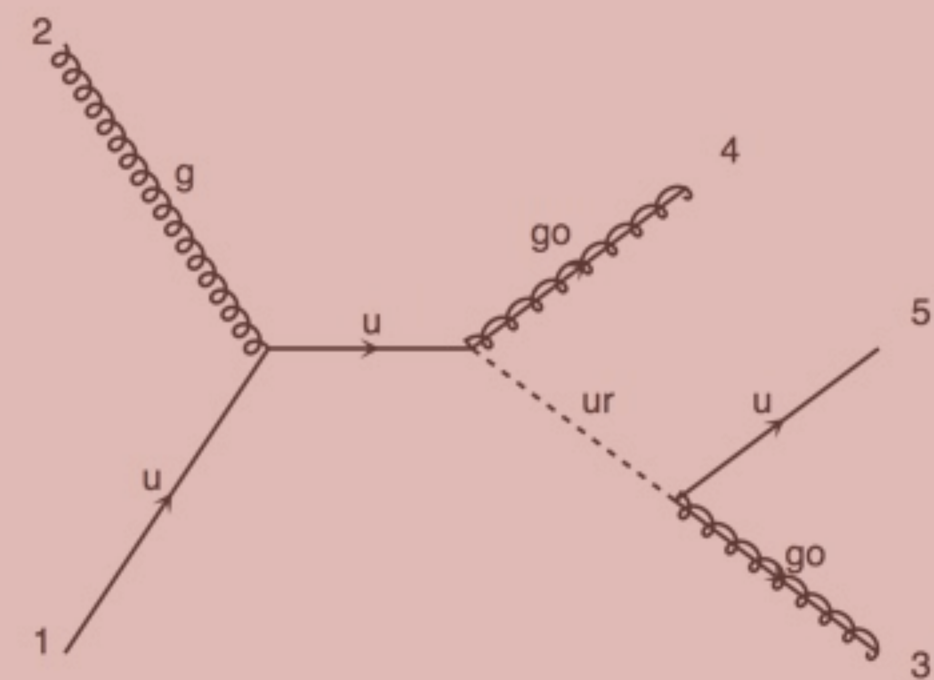
X ✓ ✓

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.

Future Plans

Plugins in MG5aMC

Structure developed by O.Mattelaer

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

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

Let people implement those themeselves!

- → Ideal projects for **students**
- → Dev. and maintenance **independent** from **MG5_aMC**
- → Also **authorship** of PLUGINS are more **properly credited**.
- → Flexible: can implement highly **complicated tasks:**
  Ex: MadDM or shower evolution kernels generation

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
```

Long term plans:

FR + NLOCT	Model
MG5 / MadLoop	Matrix Element
MadEvent / MadFKS	Partonic Events
Pythia / Herwig	Hadron level
PGS / DELPHES	Detector level

Long term plans: MadEvent7?

Oldest and "weakest" link of the chain:

FR + NLOCT	Model
MG5 / MadLoop	Matrix Element
MadEvent / MadFKS	**Partonic Events**
Pythia / Herwig	Hadron level
PGS / DELPHES	Detector level

Objectives for MadEvent7

- Insist on modularity. Independent building blocks:

Integrator | Phase-space sampler | Integrand(s)
Matrix elements | Observable operator | Mappings
Subtraction counterterms | ... (structurally similar to sherpa)

- Organized so as to offer arbitrarily scalable parallelization and MPI-support.
- Implement various grid update strategies. Maybe account for correlations between a couple of dimensions. Implement better integrators for low dims.
- More generic support of various topologies:
  t-channel enhancement, n-point interactions, etc...
- More systematic handling of zero contributions and numerical instabilities
- Offer a highly abstract integration framework to support the intricate book-keeping of higher-order computations
- Keep RAM, disk-space and generation time under control.

Objectives for MadEvent7

- Advanced profiling and real-time monitoring of the integration.
- Adaptative Multi-channeling weights.
- Grids pre-training on cuts.
- Easy implementation of on-the-fly reweighting / bias. Need a streamlined interface to other tools for these weights. Multi-loops libs, showers, etc...
- Would probably be full-fledged python, with the couple of time-consuming bits via C++/fortran imports and/or Numpy.

MadEvent and MadFKS current structures are a hinderance to many current projects and will be even more so in future ones.

We need to seriously **discuss** about **their successor**.

I now welcome your comments