MadGraph/MadEvent

Getting ready for the uncertain future...

Michel Herquet - NIKHEF TH

MC4LHC Readiness Workshop

Ready ? For what ?

Ready ? For what ?

NLO

Exp. software integration

Very exotic models

Multi-jet samples

Exotic models

Effective theories

Advanced analysis techniques

DECAY CHAINS

Real corrections

Cluster/Grid computing

Merging ME/PS

DECAY PACKAGES

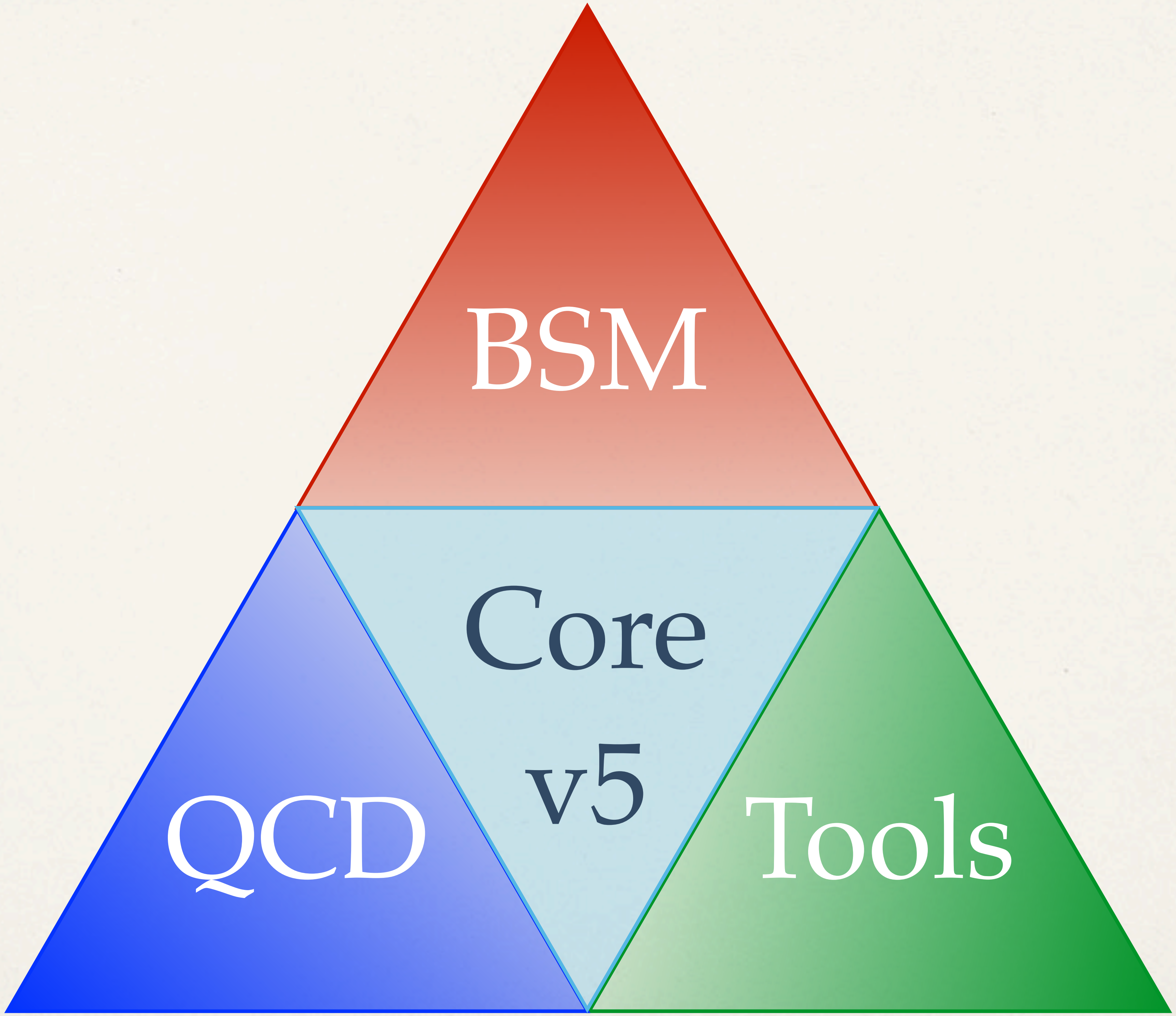
Testing / robustness

NNLO

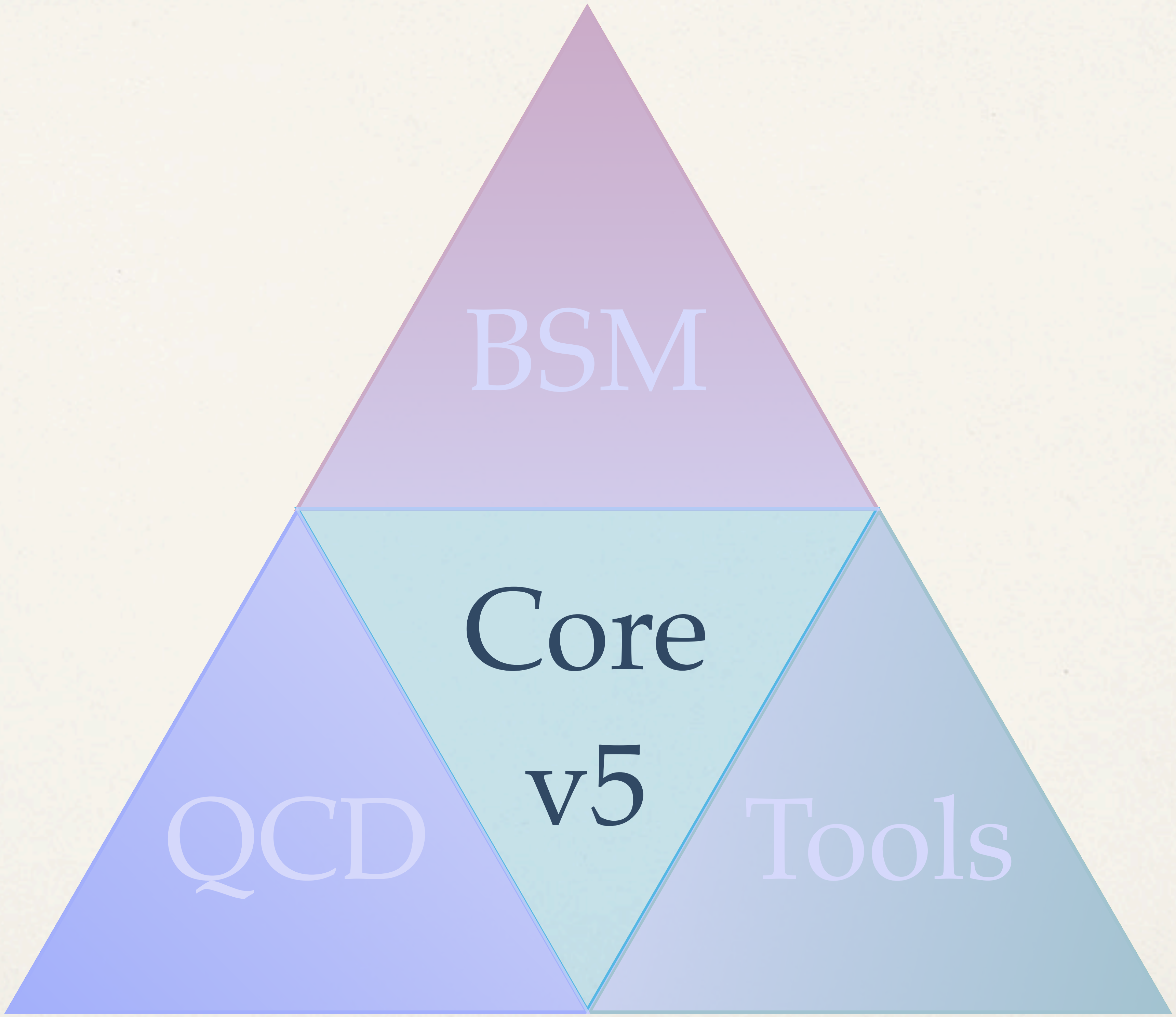
MATRIX ELEMENTS

User Interface

Short term plan



Short term plan



Core code: MadGraph v5

[J. Alwal, M. Herquet, F. Maltoni, O. Mattelaer, T. Stelzer]

- Development strategy
- Structure
- Innovations
- Benchmark v4 versus v5

Development strategy

- Use the “eXtreme Programming” software engineering scheme:
  - “Bazaar” design (features first, structure after)
  - Pair programming (two brains, one computer)
  - Systematic testing (unit, acceptance, parallel)
  - Planning game (dynamic feature list, short release plan)
- Intensive use of Distributed Versioning (Bazaar+Launchpad) and collaborative tools (wikis, ...)

Development strategy (ctd.)

- Programming language: Python
  - (Very) high level (Object Oriented, functional programming, ...)
  - Easy to learn/write/maintain and concise (x4 compared to F77)
  - Easily available on all platforms and no compilation required
  - Slow, but fast standard library (99% of calculations) and easily expandable

Structure (MadGraph)

- Abstract and dissociate layers:

  input → parser → object → calculation → object → parser → output

- Modern architecture:
  - madgraph/ The main library, divided into modules (core, iolibs, interfaces, ...), usable as any Python library
  - tests/ Various test suites
  - apidoc/ Automatically generated documentation

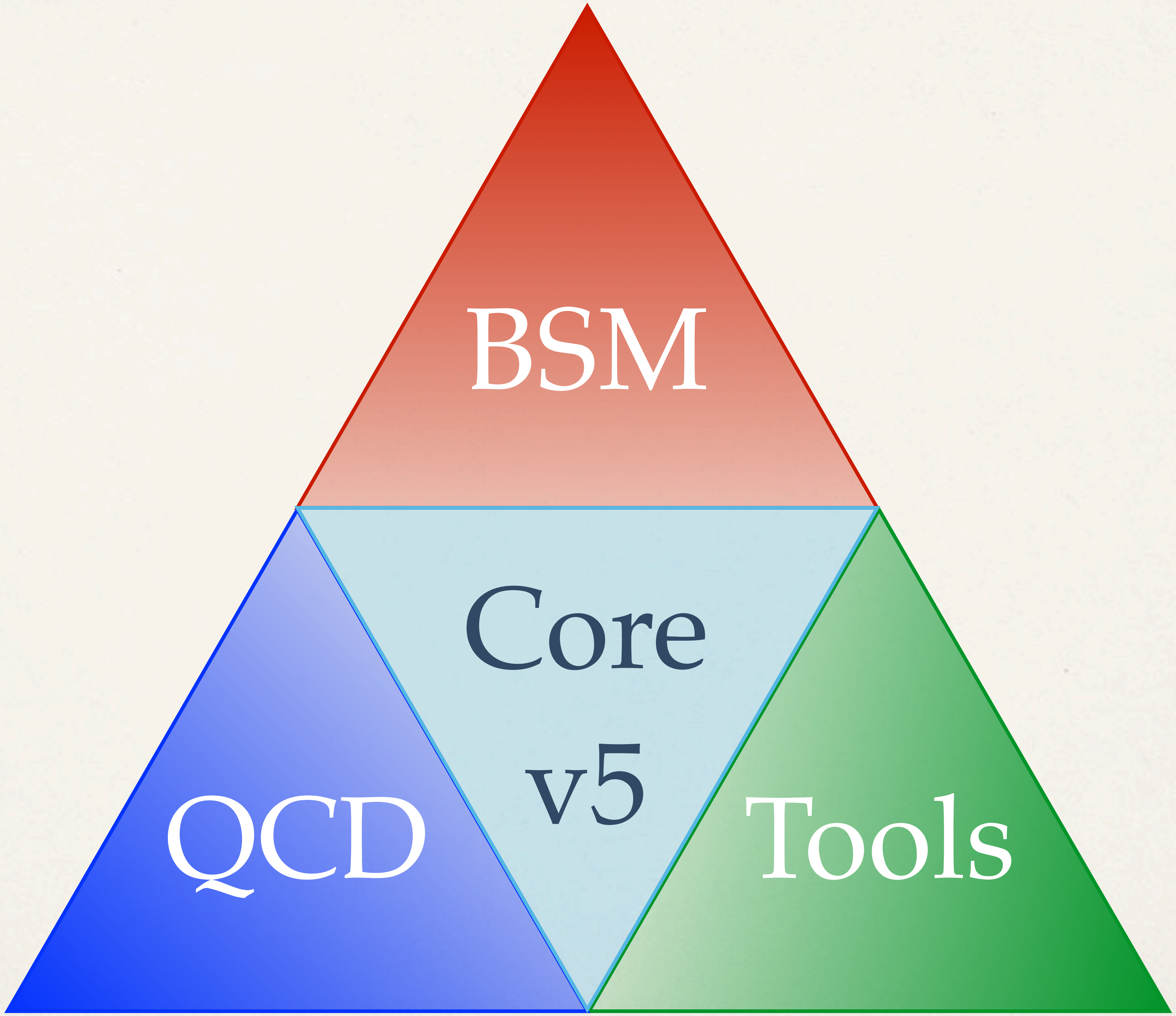
Innovations

- User friendly command line interface (a la ROOT)
- Completely new diagram generation algorithm
  - Makes optimal use of model information
  - Deal with multiprocesses very efficiently (keep track of discarded combinations, ...)
- Completely new HELAS call generation algorithm (90% less calls for critical cases!)
- Generic and "smart" new color calculation library
- New, faster and generic diagram drawing library
- Matrix elements outputs: Standalone, MadEvent v4, ... and more!
- ... and (much) more to come !!!

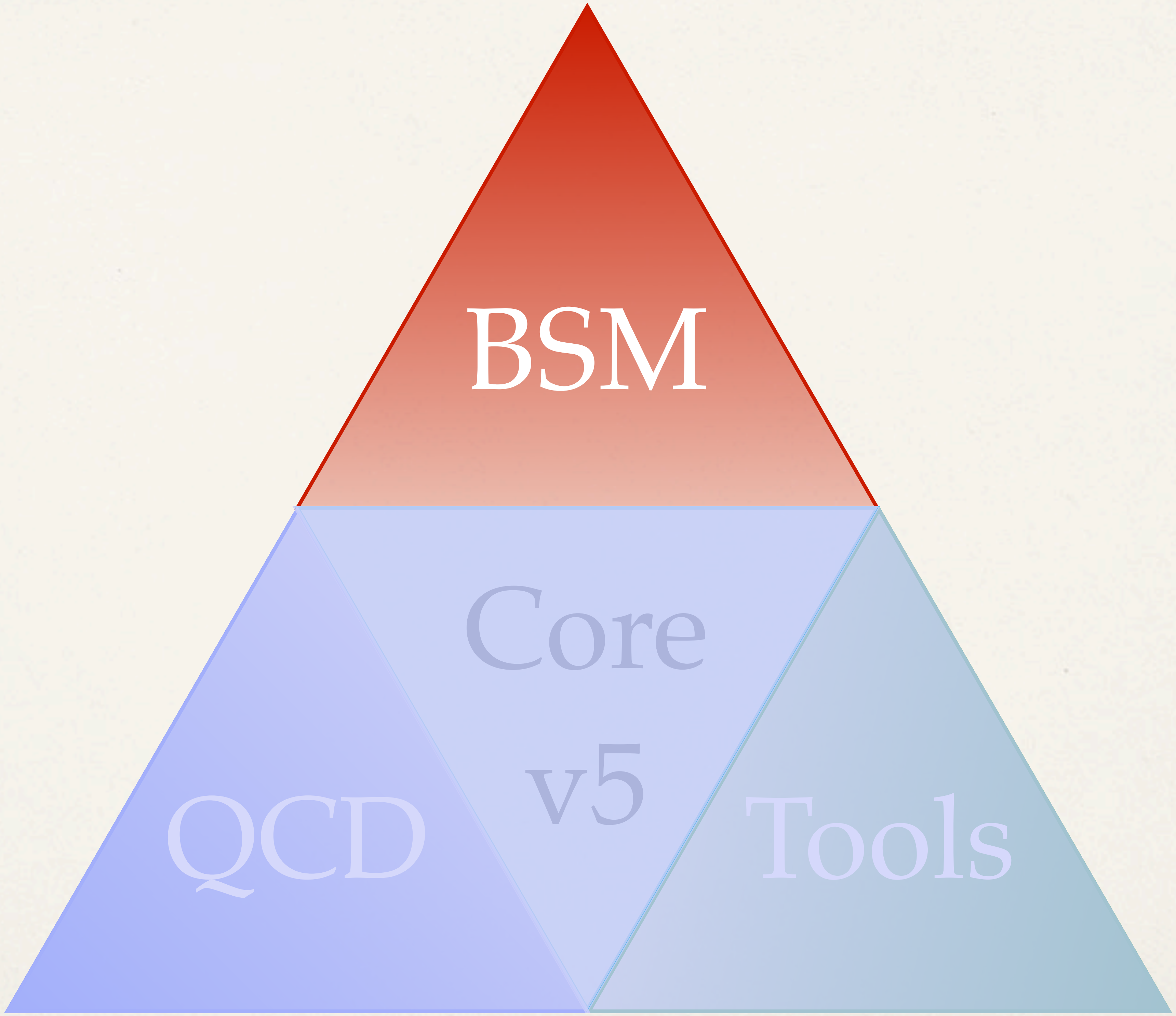
Benchmarks

	MG4 standalone	MG5 0.3 Alpha
SM 2→2	4 min	1 min
SM 2→3	70 min	26 min

Short term plan



Short term plan



New physics models

- The new FeynRules interface
- Generic color structures
- Generic Lorentz structures

The new FeynRules interface

[C. Duhr, D. Grellscheid, M. Herquet, W. Link, O. Mattelaer]

- Full use of Object Oriented notation (in Python)
- Lists of particles, interactions, coupling expressions, parameters (internal and external), but also color and Lorentz structures!
- Not restricted to MadGraph, easy to extend
- The most ambitious Lagrangian-to-MC interface up-to-date, first step towards unprecedented BSM possibilities

```
vertices[0] = {
    'particles':[u, u, g],
    'color':[ 'T(a3,i2,i1)', ...],
    'lorentz':[ L1, L2, ...],
    'couplings':[ (0,0):'g1', (0,1):'g2', ...],
    'orders':['QCD', ...]
    }
```

Generic color structures

- Color is now completely generic (tested SM 2→ 2, 2 → 3):
  - The color structure of a vertex is described inside the model using a textbook notation, e.g.:

```
'color':[[f(0, 1, -1),f(2, 3, -1)],
         [f(2, 0, -1),f(1, 3, -1)],
         [f(1, 2, -1),f(0, 3, -1)]]
```

  - The full color factor associated with a diagram is simplified using (easy to implement and modify) simple rules, e.g.,

```
f(a,b,c) = -2 I Tr(a,b,c) + 2 I Tr(c,b,a)

Tr(a,x,b)T(c,x,d,i,j) = 1/2(T(c,b,a,d,i,j)
                        -1/Nc Tr(a,b)T(c,d,i,j))
```

    to build the color basis and color matrices for squared amplitudes

Generic Lorentz structures

[P. de Aquino, W. Link, O. Mattelaer]

- Lorentz is now completely generic (tested SM 2→ 2, 99% of SM 2 → 3 yesterday!):

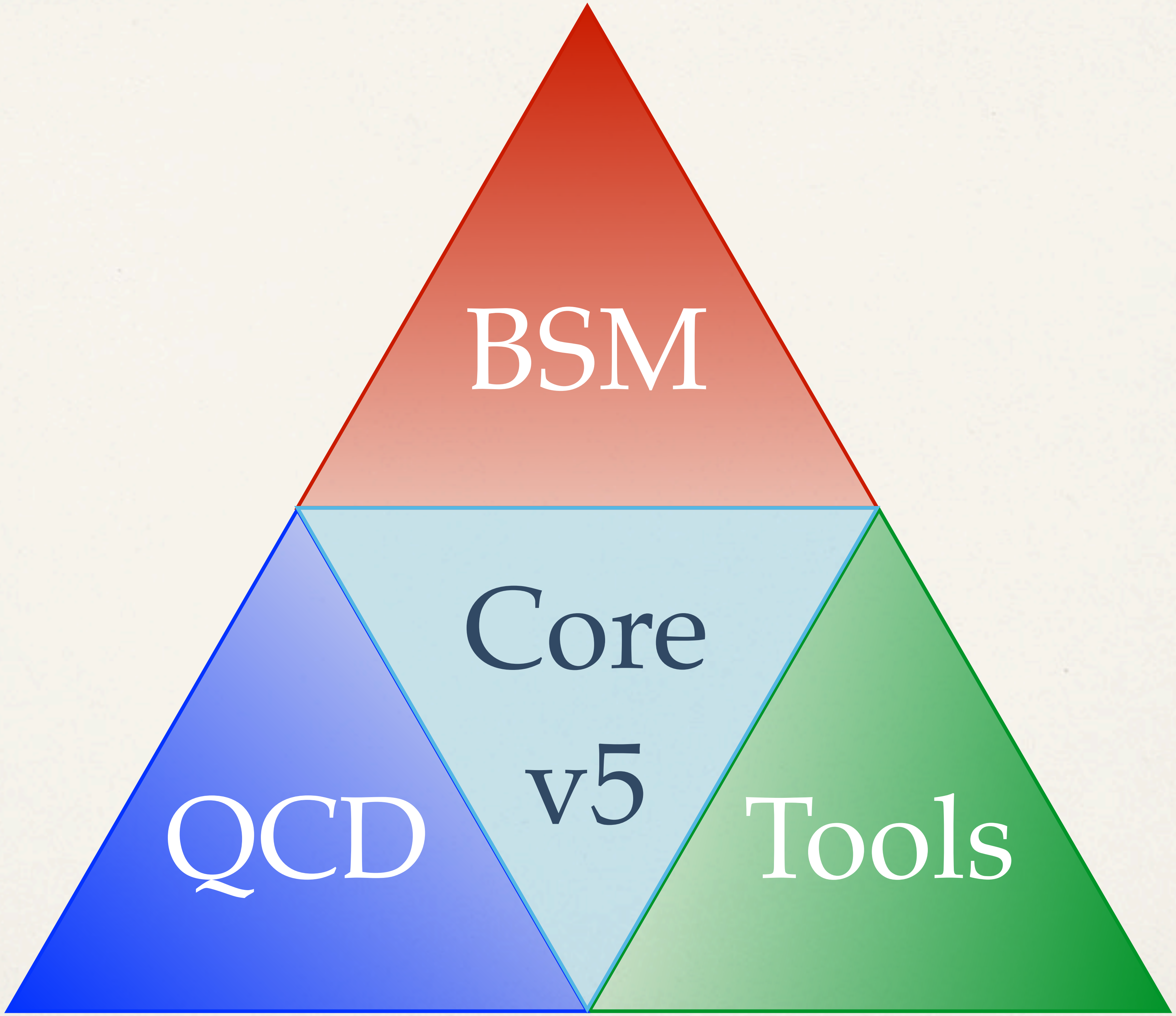
  - The color structure of a vertex is described inside the model using a textbook notation, e.g.:

```
'Structure':[complex(0,1)*Gamma(1,2,'a')*ProjM('a',3)]
```

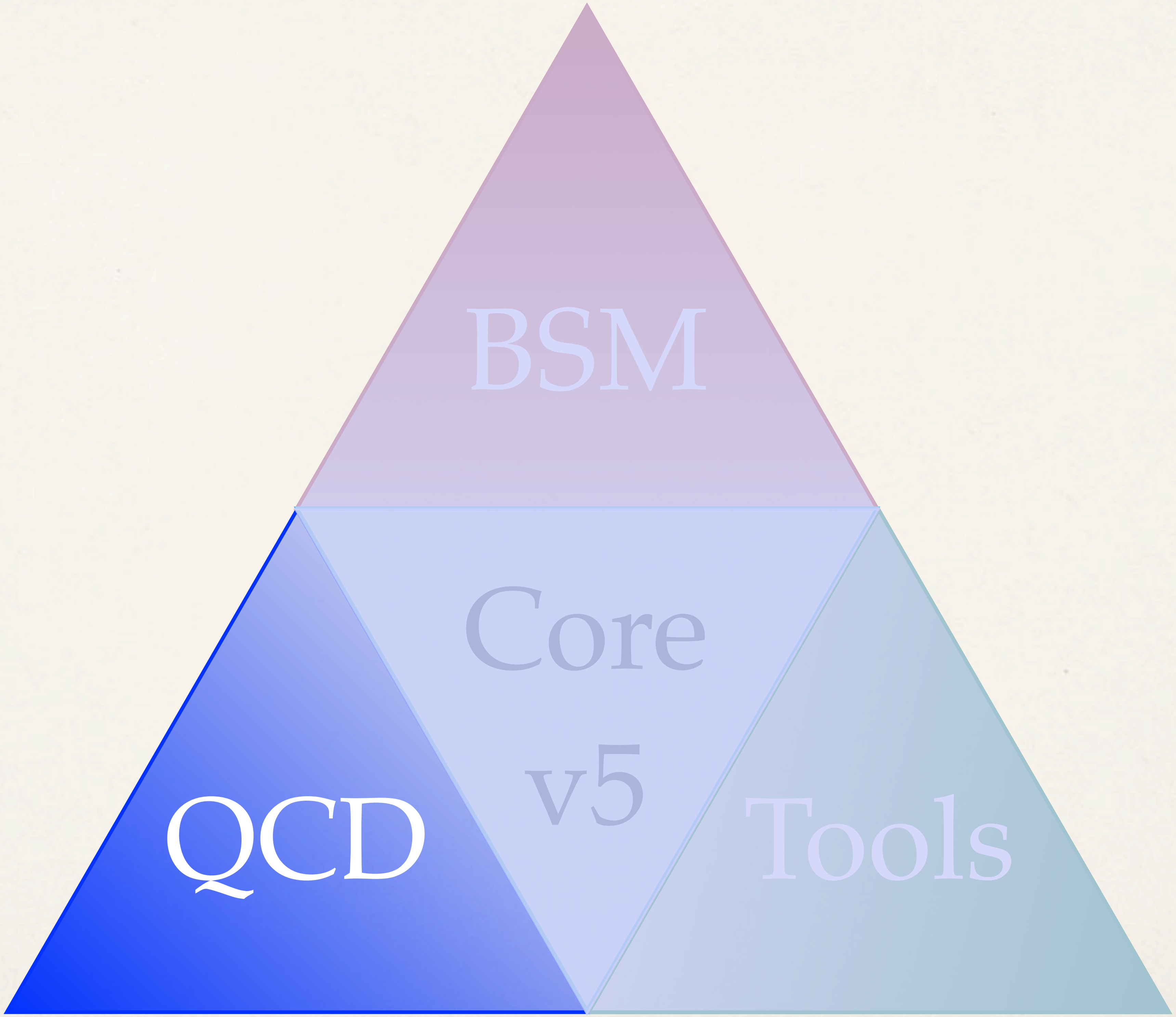
  - The corresponding optimized "HELAS" routines are produced automatically

```
      SUBROUTINE VERTEX1_111(C,V1,F2,F3,VERTEX)
      IMPLICIT NONE
      DOUBLE PRECISION C
      DOUBLE COMPLEX V1(6)
      DOUBLE COMPLEX F2(6)
      DOUBLE COMPLEX F3(6)
      DOUBLE COMPLEX VERTEX
      VERTEX = C*((F3(4)*V1(1)gra*F2(2))+(F3(4)*V1(4)*F2(2))+(F3(4)*V1(2)
     $ *F2(1))+1.*(0,1.)*(F3(4)*V1(3)*F2(1))+(F3(3)*V1(2)*F2(2))
     $ +-1.*(0,1.)*(F3(3)*V1(3)*F2(2))+(F3(3)*V1(1)*F2(1))+-(F3(3)
     $ *V1(4)*F2(1))+(F3(2)*V1(1)*F2(4))+-(F3(2)*V1(4)*F2(4))
     $ +-(F3(2)*V1(2)*F2(3))+-1.*(0,1.)*(F3(2)*V1(3)*F2(3))+-(F3(1)
     $ *V1(2)*F2(4))+1.*(0,1.)*(F3(1)*V1(3)*F2(4))+(F3(1)*V1(1)*F2(3))
     $ +(F3(1)*V1(4)*F2(3)))
      END
```

Short term plan



Short term plan

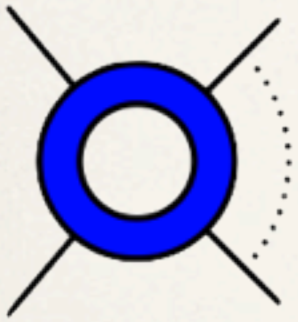


QCD

- NLO calculations
- Matching/merging ME/PS

NLO: the problem

NLO

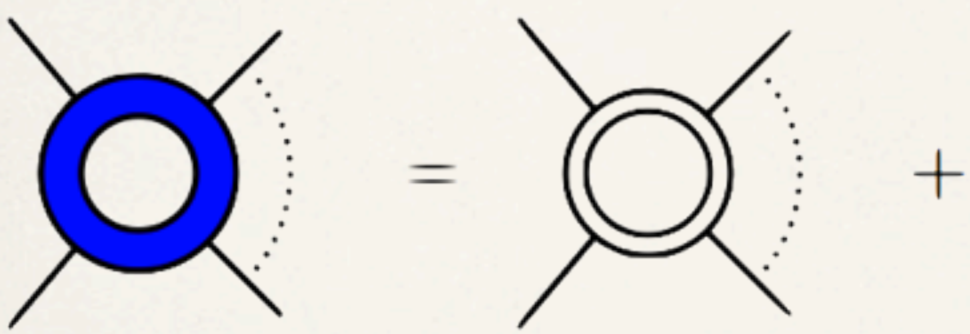


$=$

$$\sigma^{\text{NLO}} =$$

NLO: the problem

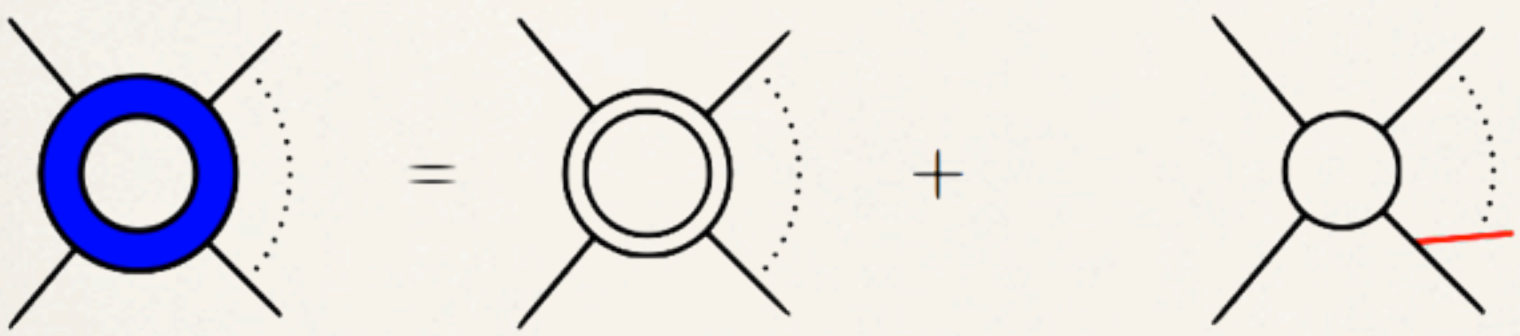
NLO Virtual



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

NLO: the problem

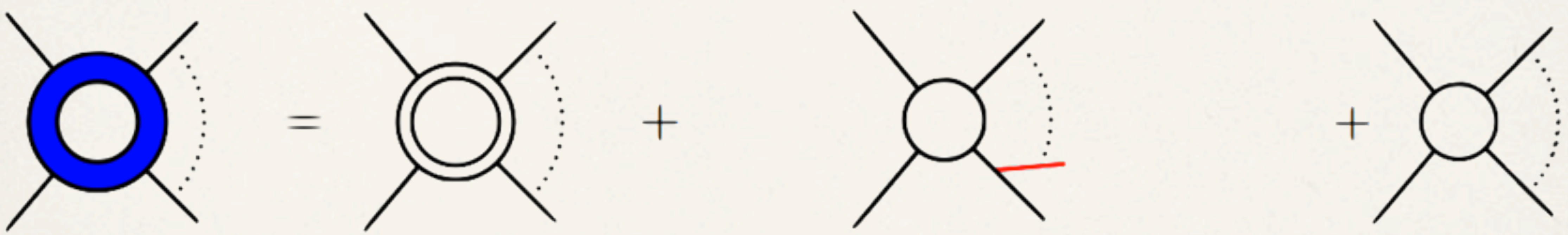
NLO Virtual Real



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

NLO: the problem

NLO　　Virtual　　Real　　Born

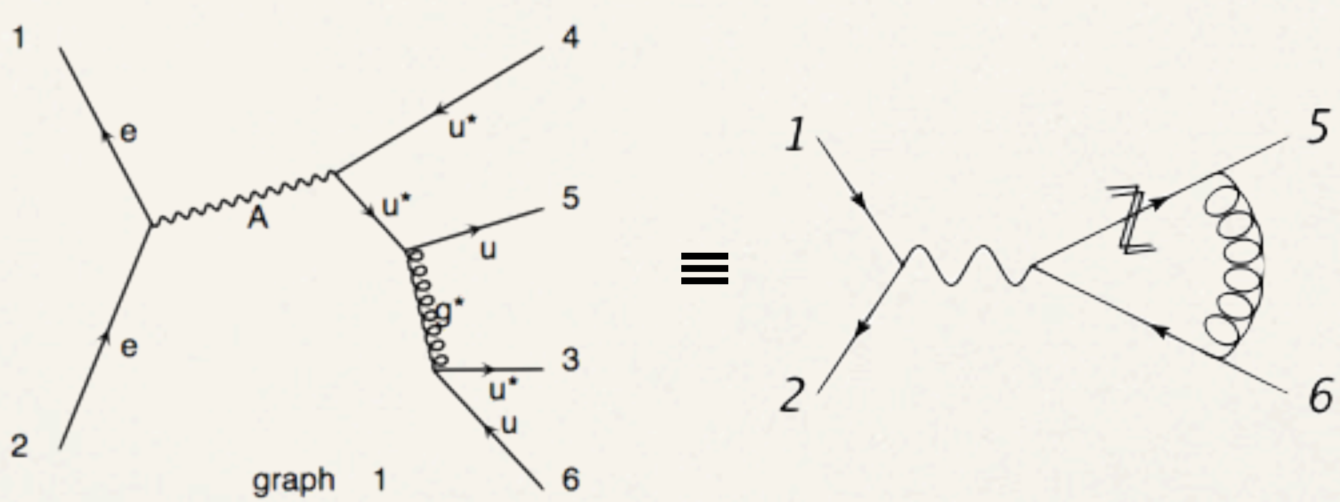


$$\sigma^{\text{NLO}} = \int_m d^{(d)}\sigma^V + \int_{m+1} d^{(d)}\sigma^R + \int_m d^{(4)}\sigma^B$$

NLO: virtual contributions

[V. Hirschi, R. Pittau, M. V. Garzielli; R. Frederix]

- Two (complementary) approaches:
  - Use MG to generate diagrams and calculate n+2 amplitudes to build the NLO result (CutTools technique), e+e- → 2 and 3 jets already checked. Advantages: valid for any BSM model



  - Rely on external tool(s) (BlackHat, Rocket, Golem, ...) using the Binoth-LHA accord (see Rikkert's talk). Various e+e- and hadronic processes checked. Advantage: strong optimization possibilities.

NLO: real contributions

[R. Frederix, S. Frixione, T. Gehrmann, N. Greiner, F. Maltoni, T. Stelzer]

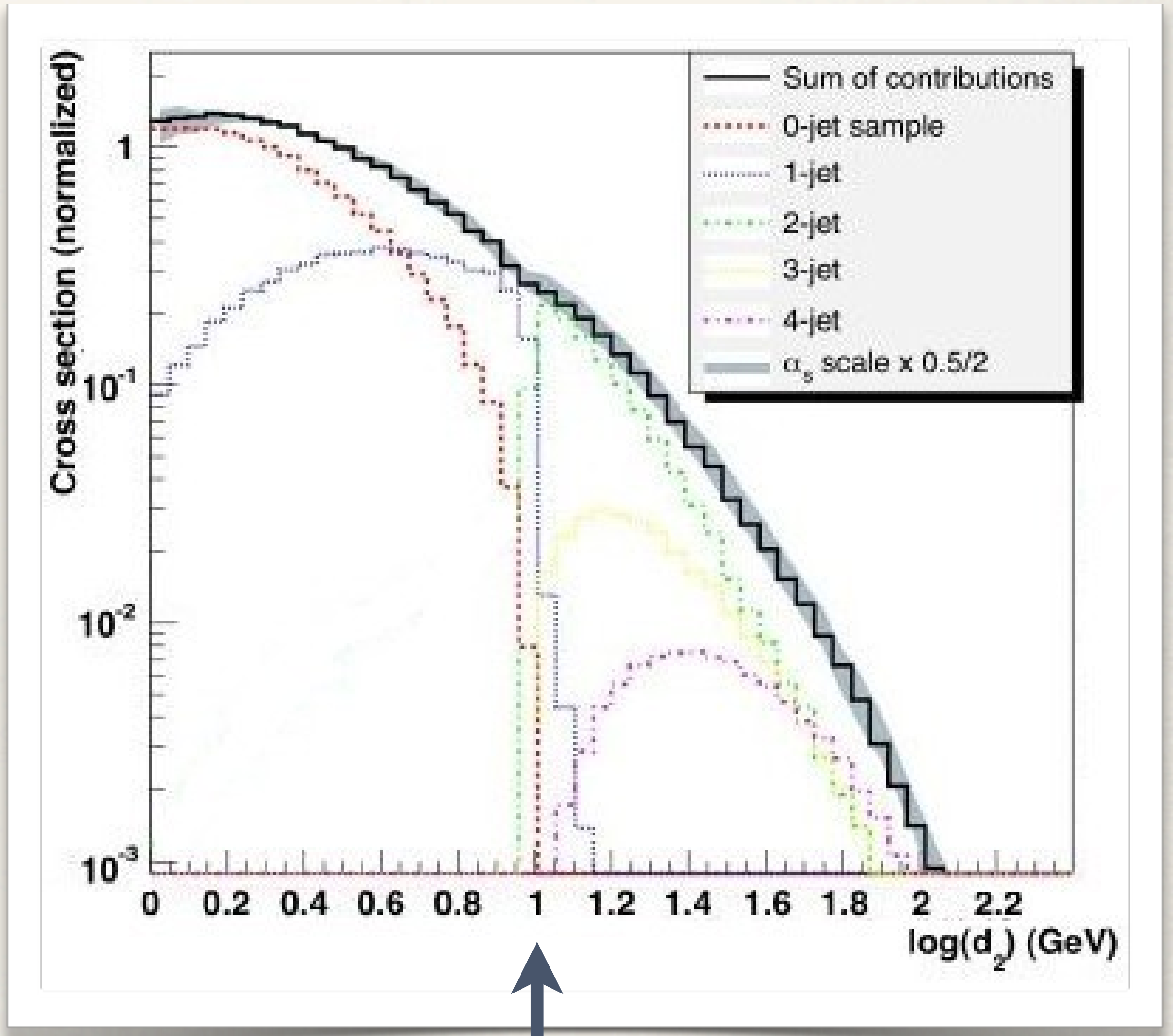
- Two approaches:
  - MadDipole: Catani-Seymour dipole substraction scheme, standalone implementation (TH), cancellation of singularities checked, and dipoles checked against MCFM
  - MadFKS: Frixione-Kunszt-Signer substraction scheme, integration is available (TH+PH), cancellation of singularities checked + see Stefano's talk
- Both: usable both for SM and BSM processes, and for massless and massive external particles

ME/PS Matching

[Alwall et al.]

- Matching schemes implemented with Pythia: kT and cone jet MLM schemes, new "shower kT" scheme
- Both Q^2- and pT-ordered Pythia parton showers
- Extensively validated, W+jets compared with other generators and Tevatron data
- Allows matching in most SM and BSM processes

Jet resolution for 1 to 2 jets

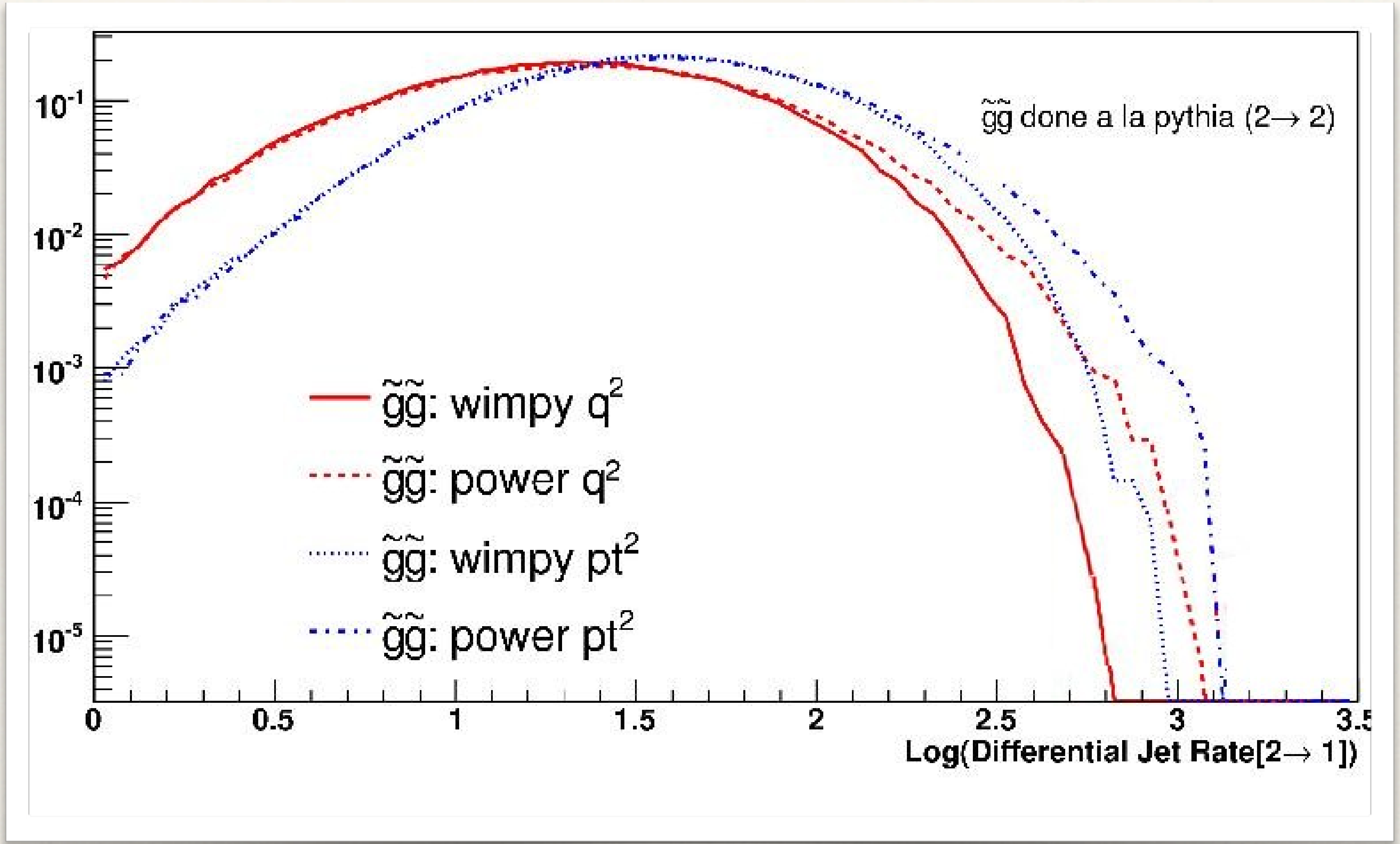


Cutoff (unphysical)

Matching for BSM processes

[J. Alwall, S. de Visscher, F. Maltoni]

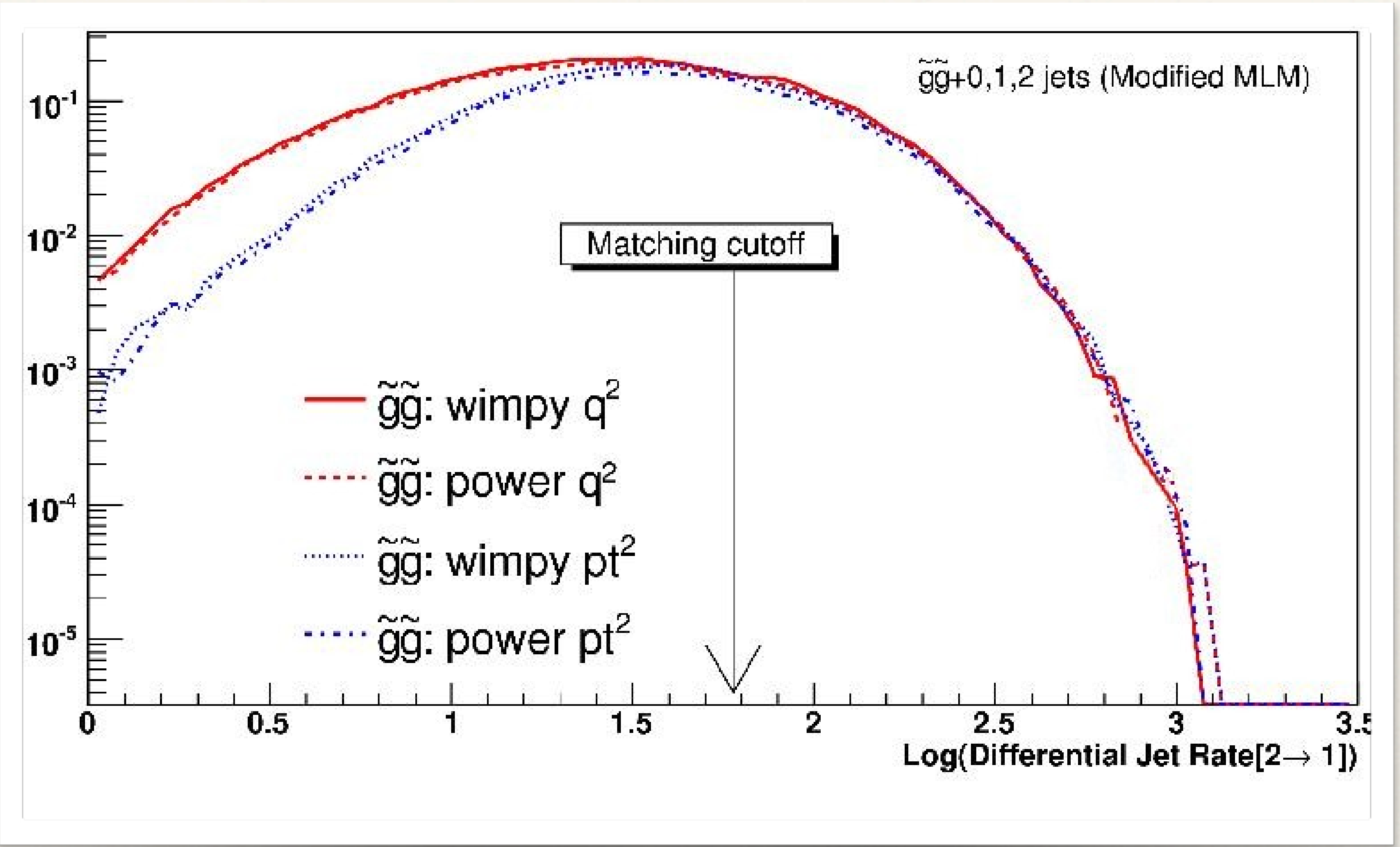
600 GeV gluino pair production at the LHC



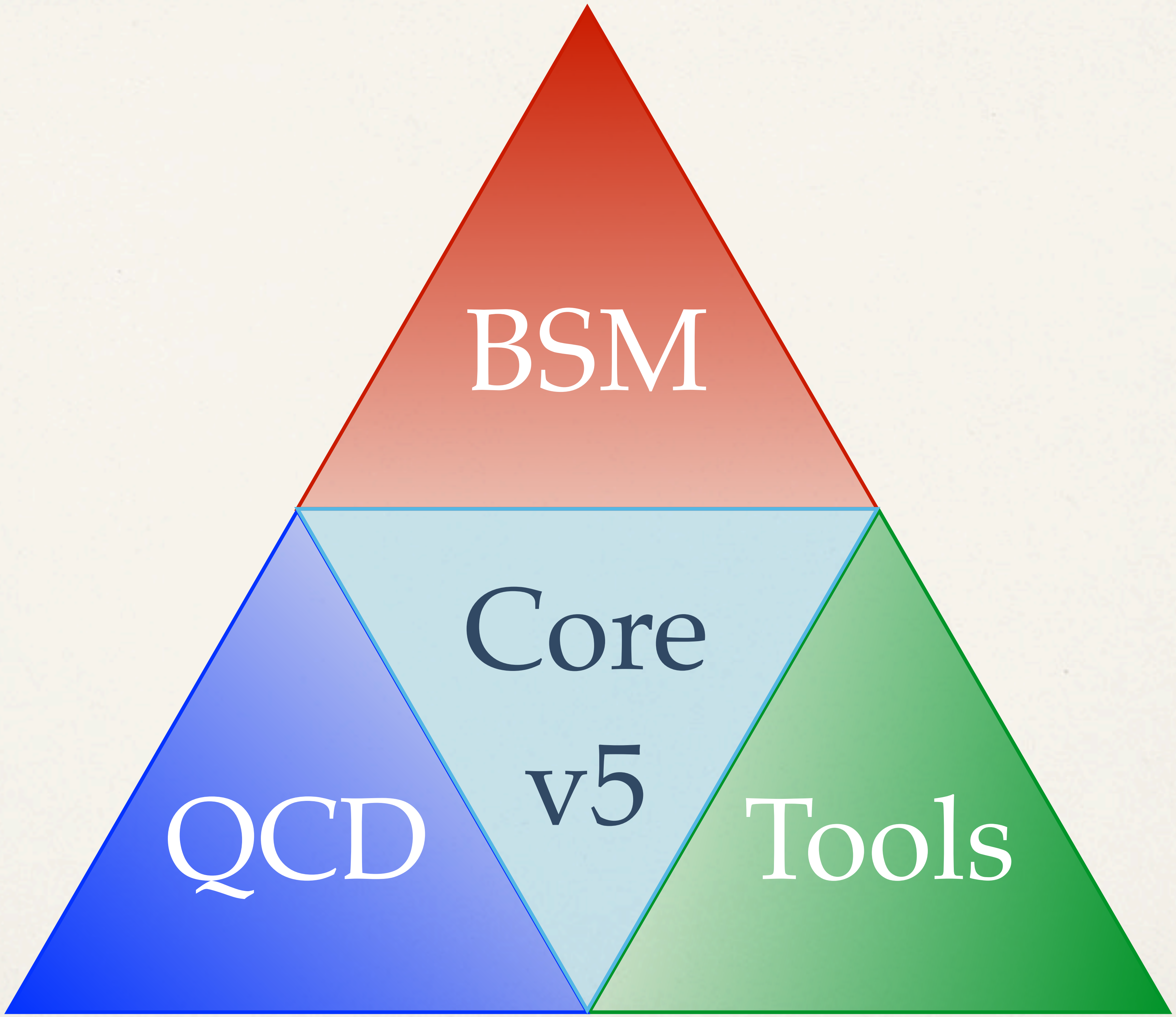
Matching for BSM processes

[J. Alwall, S. de Visscher, F. Maltoni]

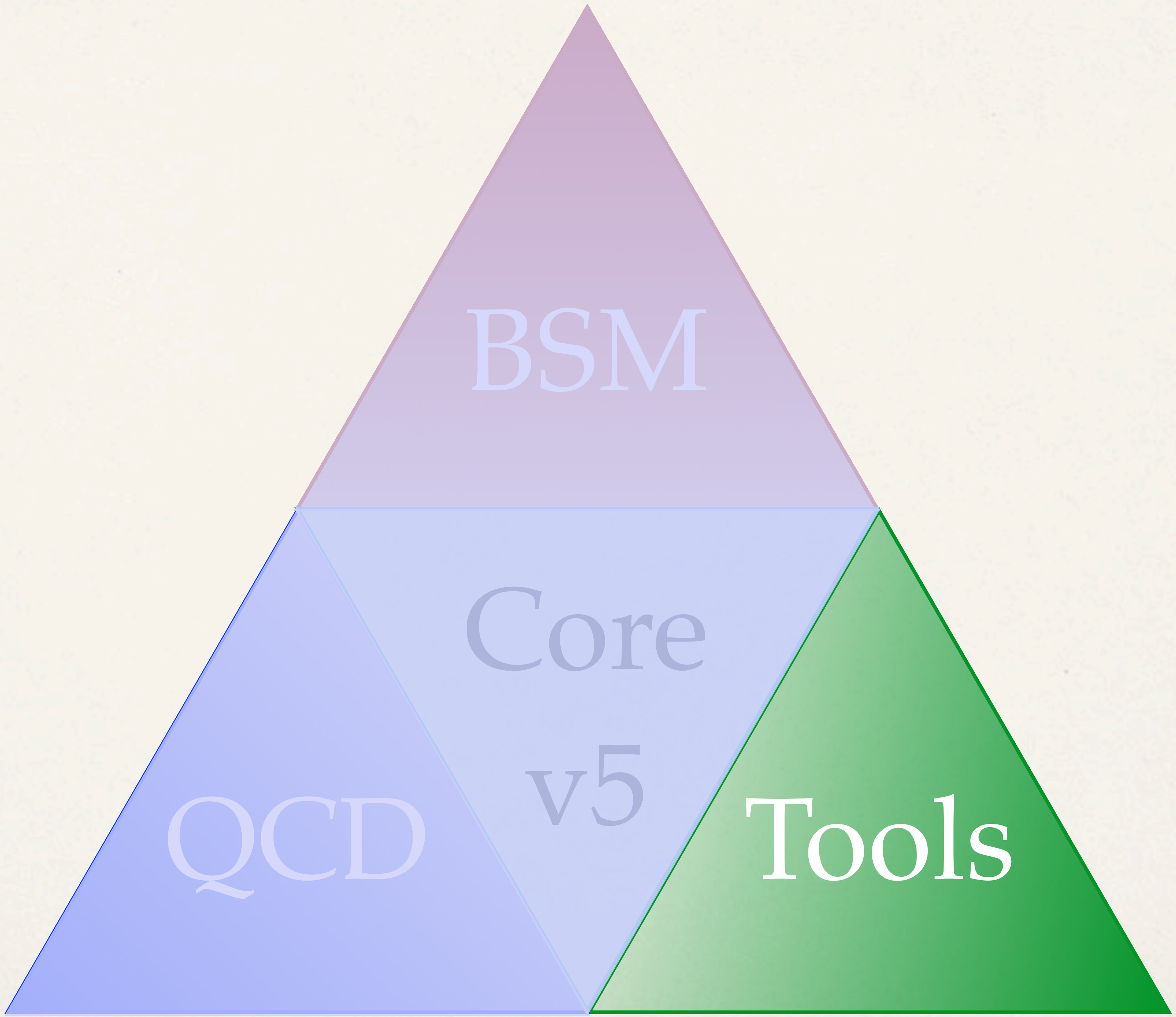
600 GeV gluino pair production at the LHC



Short term plan



Short term plan



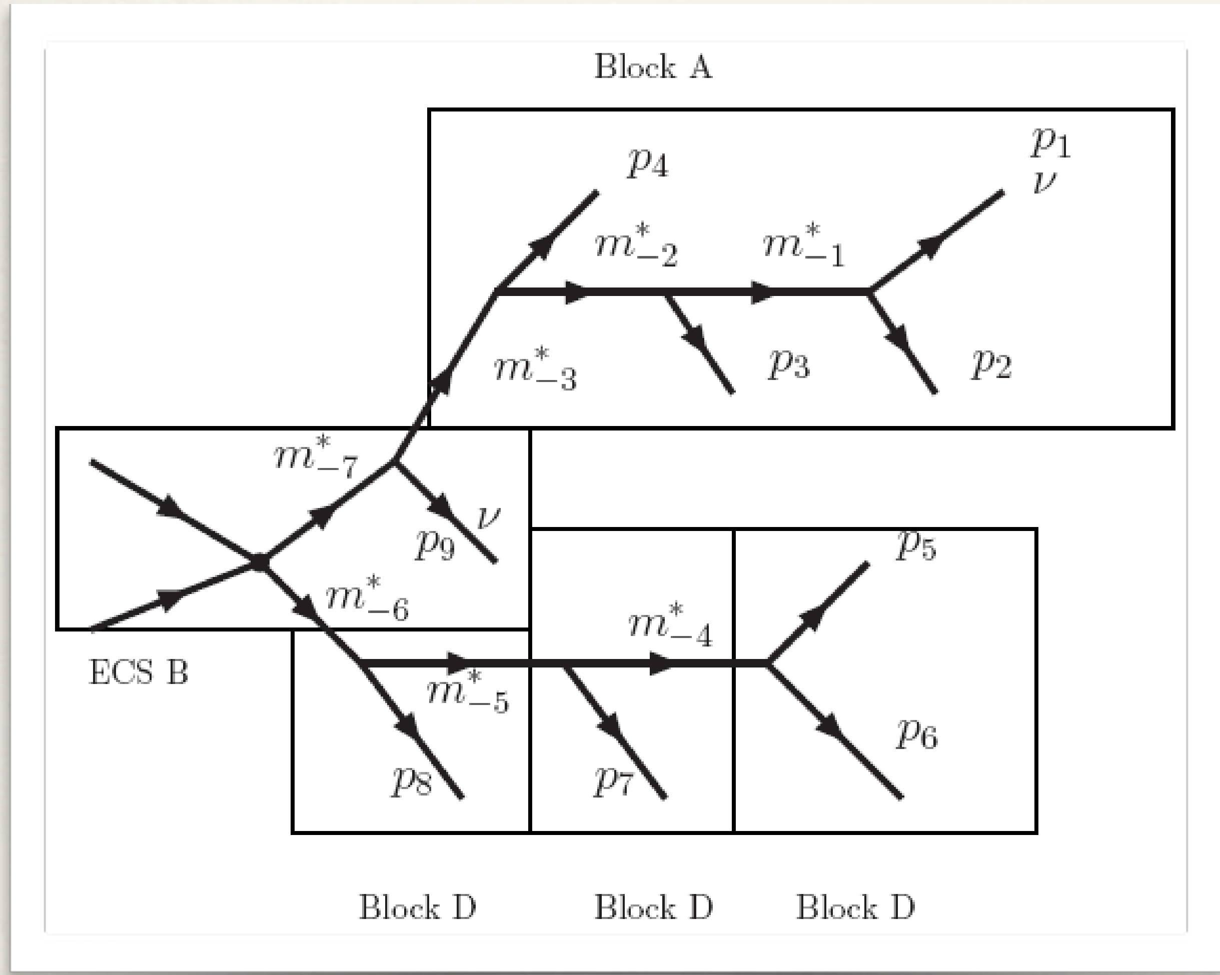
Tools

- MadWeight: Matrix Element methods
- MadOnia: Onium production
- MadGraph on a graphic card
- Mass production

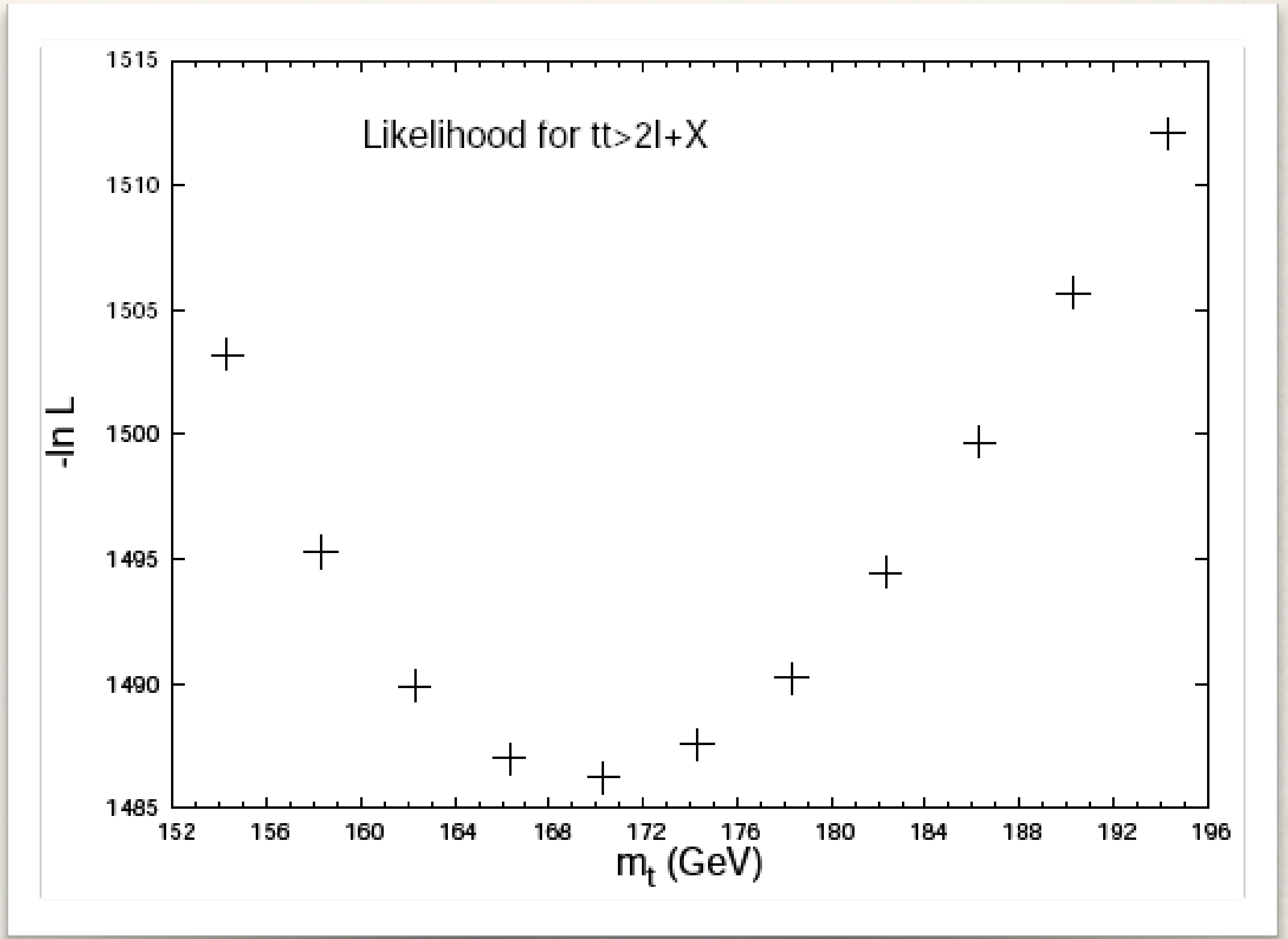
MadWeight

[P. Artoisenet, V. Lemaitre, F. Maltoni, O. Mattelaer]

- Tool to find matrix element weight of exp. events for (almost) any process in any model:



Phase space integration using automatic change of variables aligned with peaks

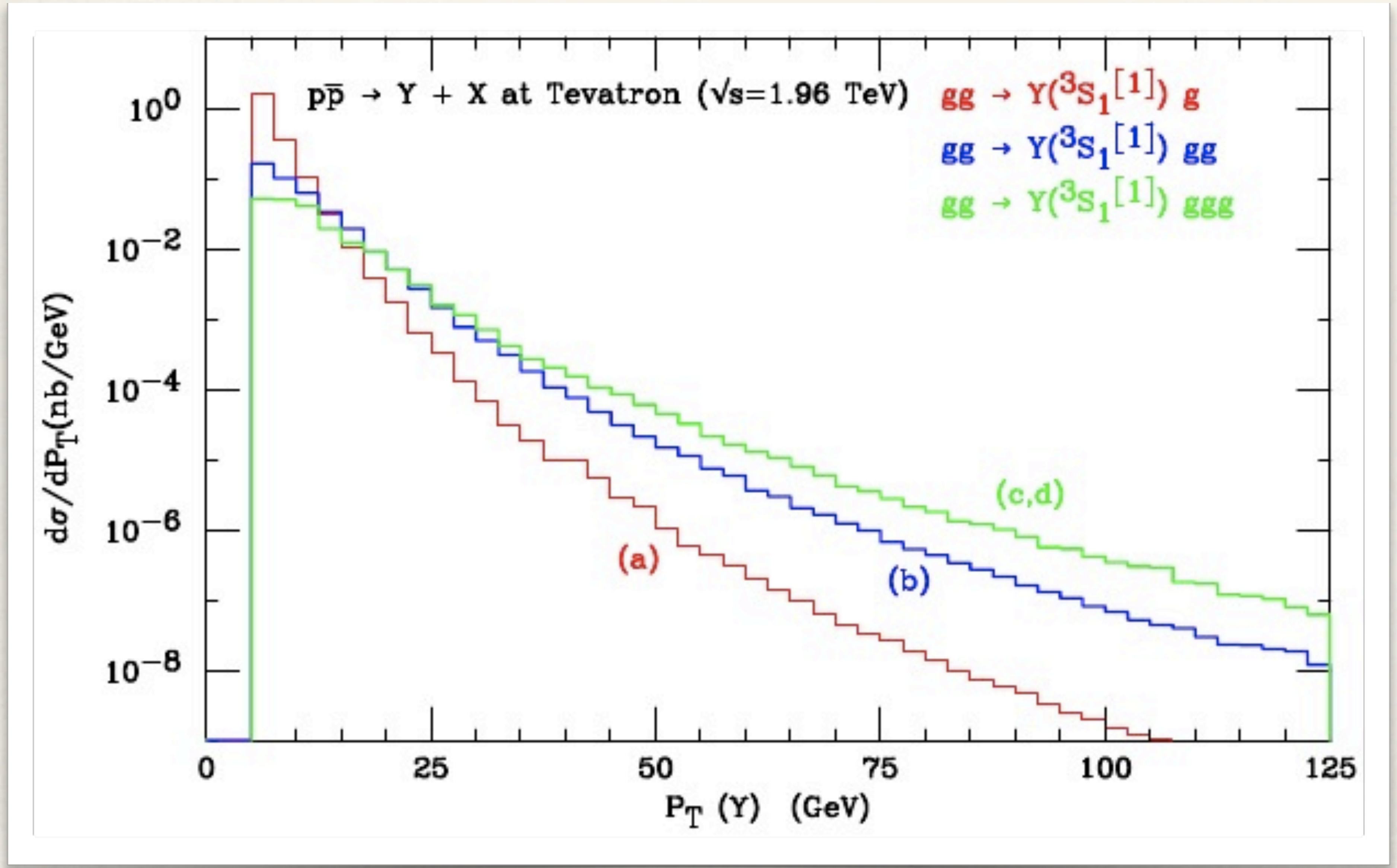


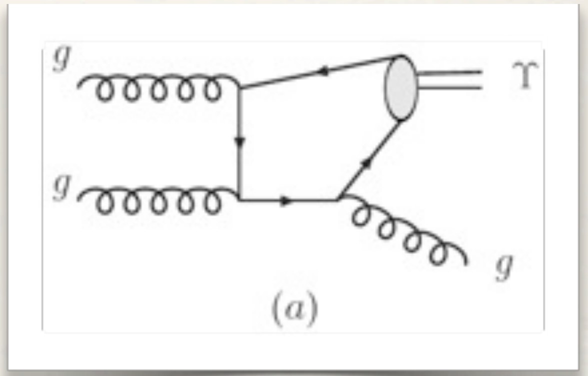
Find likelihood for model parameters (here top mass)

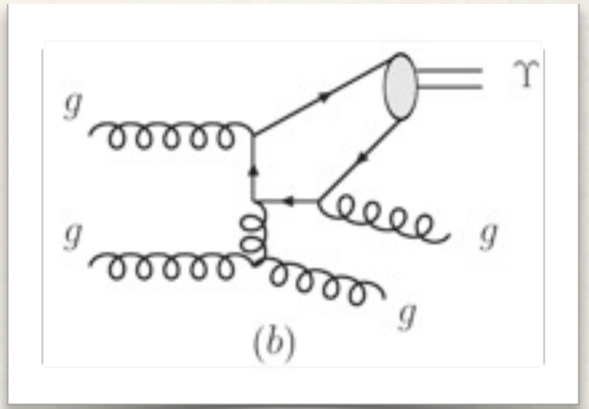
MadOnia

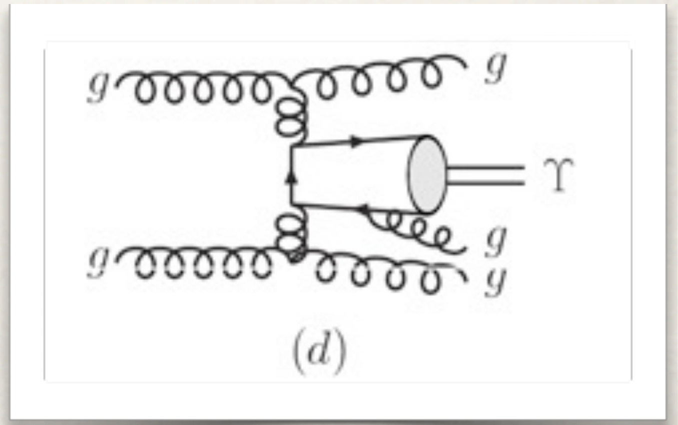
[P. Artoisenet, F. Maltoni, T. Stelzer]

- Production of quarkonium events at tree level within non relativistic QCD
- Example of application: Υ+ jets in hadron collisions





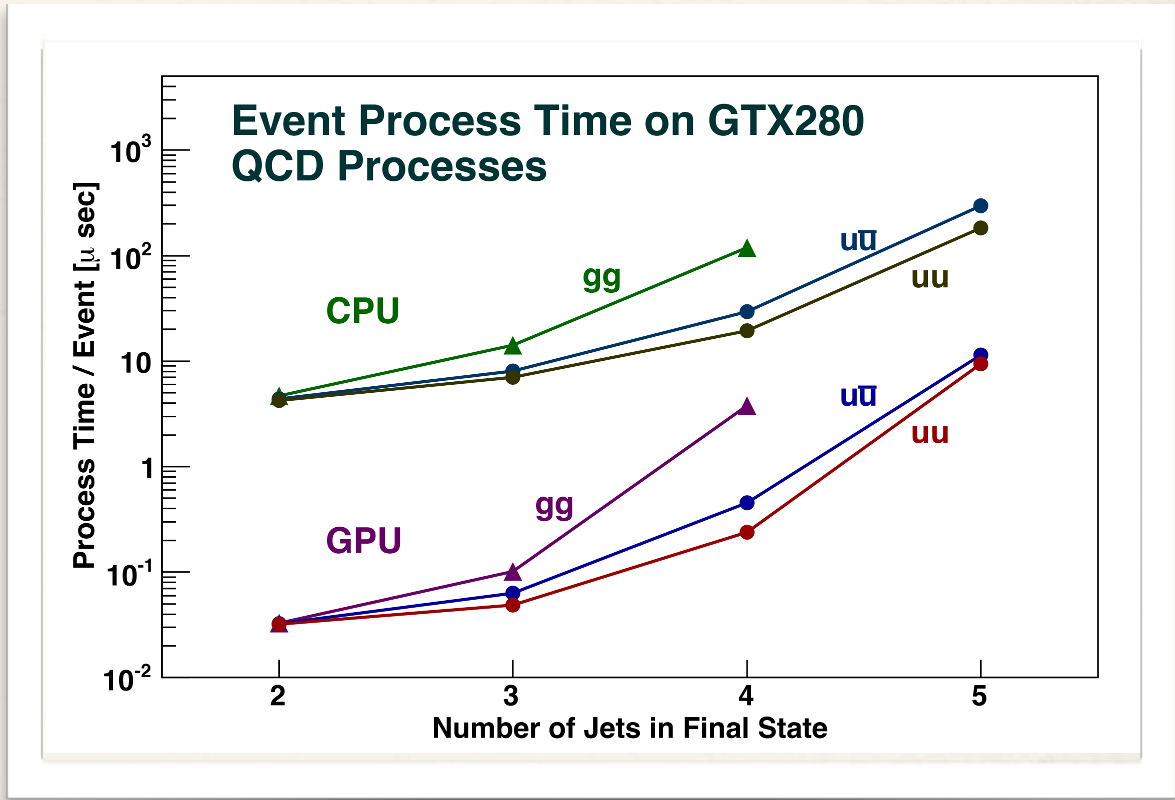


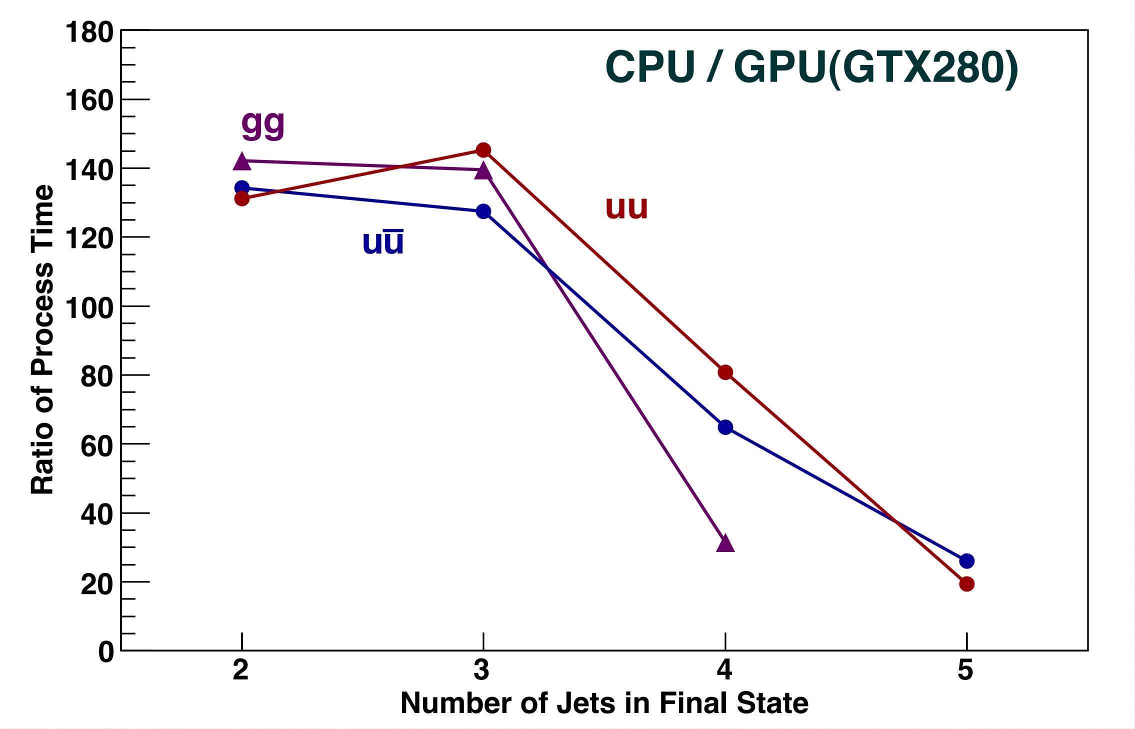


MadGraph on a graphic card

[K. Hagiwara, J. Kanzaki, N. Okamura, D. Rainwater, T. Stelzer]

- Use a graphics processing unit (GPU) for fast calculations of helicity amplitudes
- New HELAS in CUDA library, HEGET, and convertor for MG
- First studies for QED and QCD processes
- Impressive speed improvements (x 20-150)





Mass production

- "Gridpack" version of MG/ME:
  - Completely frozen, self contained package for a given process/set of cuts (only inputs: number of events and random seed)
  - Designed to be sent over the Grid
- Public library of several SM backgrounds (jets, W,Z+jets, tops+jets,...) available and validated (matching,...). Currently ~100 gridpacks for 10 and 14 TeV.
- Used for massive production of SM backgrounds by the CMS collaboration

Timeline

V4 V5

	Sept 09	Dec 09	Mar 10	June 10	Sept 10	Dec 10
MG						
ME						
BSM						
NLO V						
NLO R						
Tools						

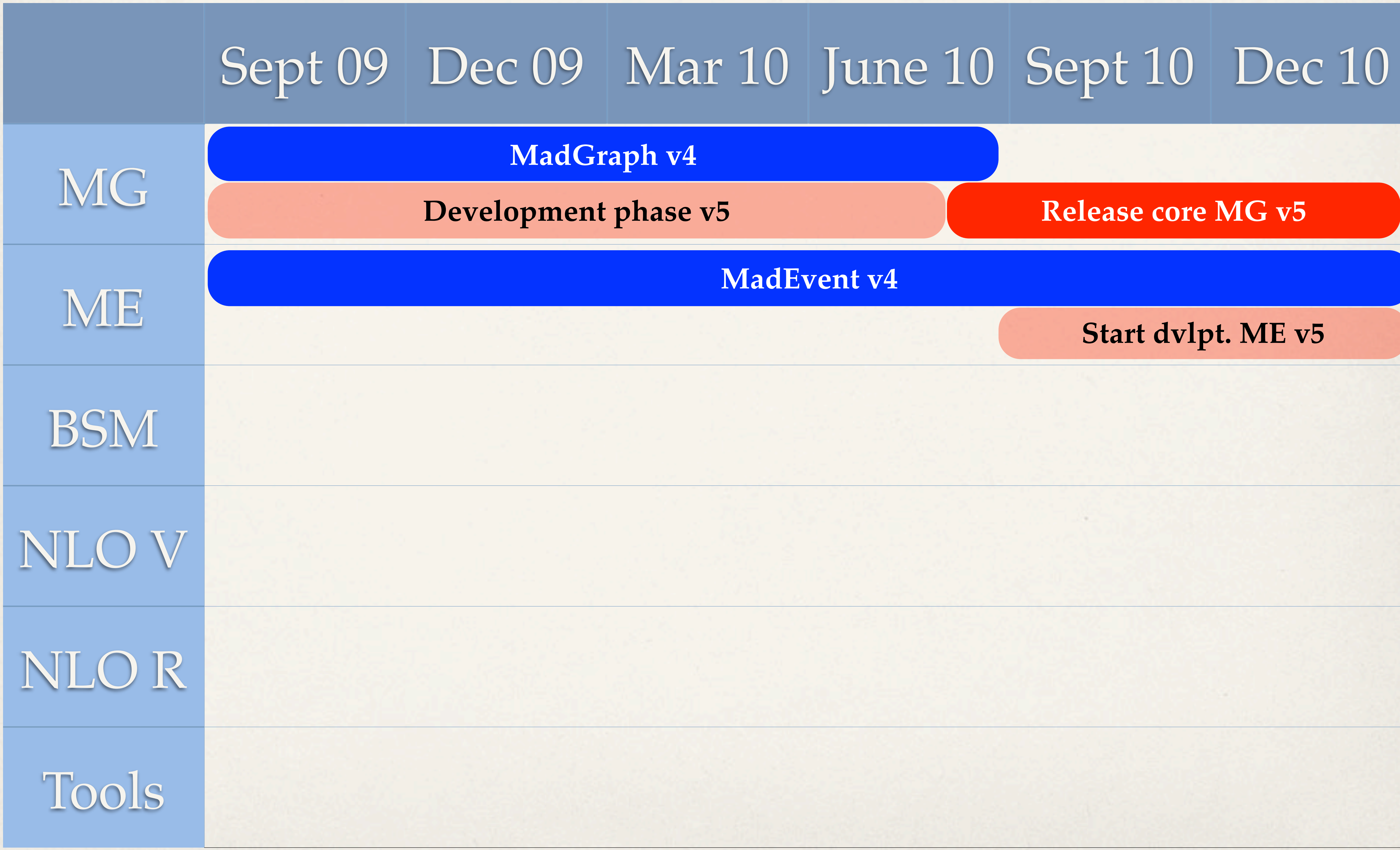
Timeline

V4 V5

	Sept 09	Dec 09	Mar 10	June 10	Sept 10	Dec 10
MG	MadGraph v4					
	Development phase v5				Release core MG v5	
ME						
BSM						
NLO V						
NLO R						
Tools						

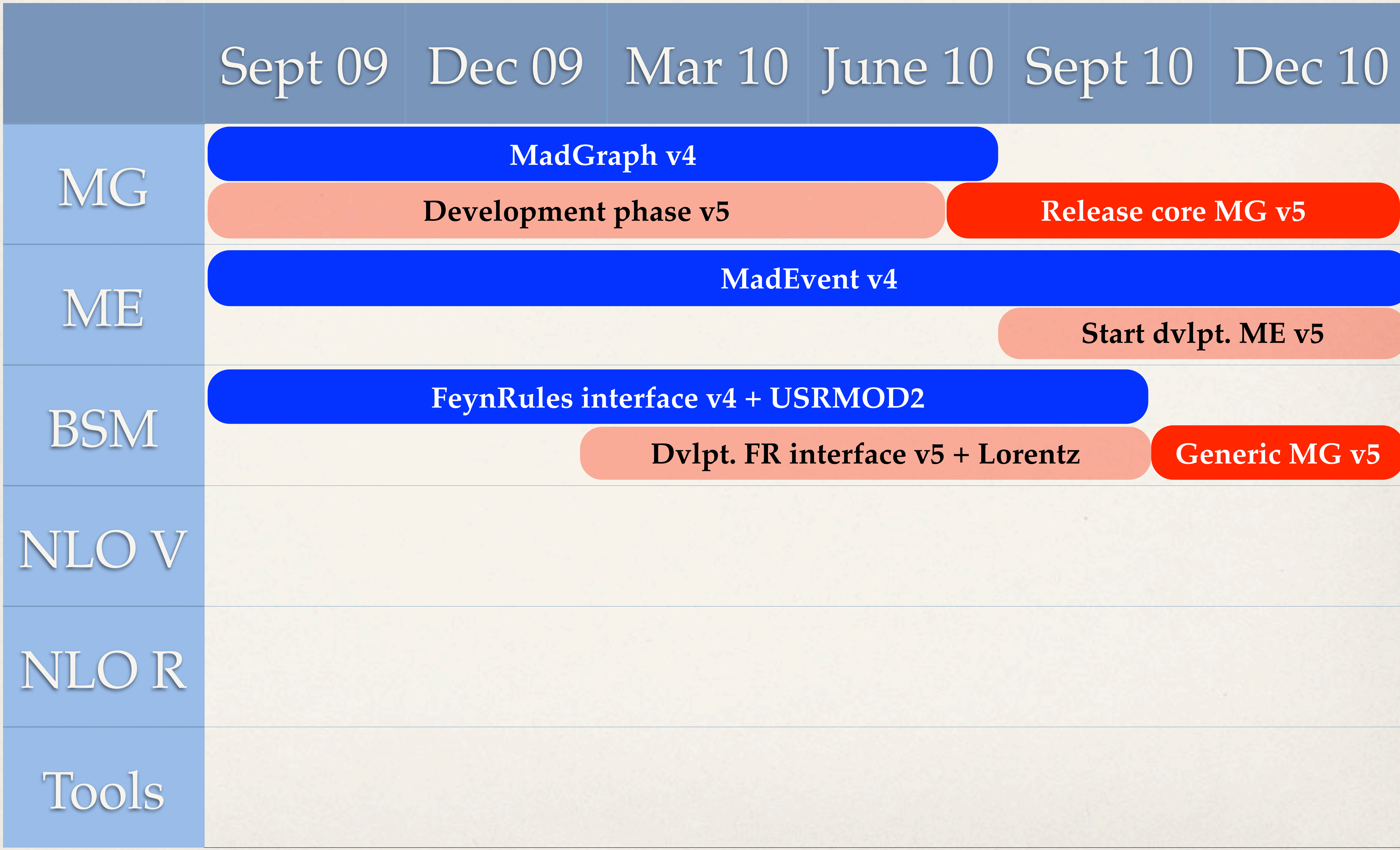
Timeline

V4 V5



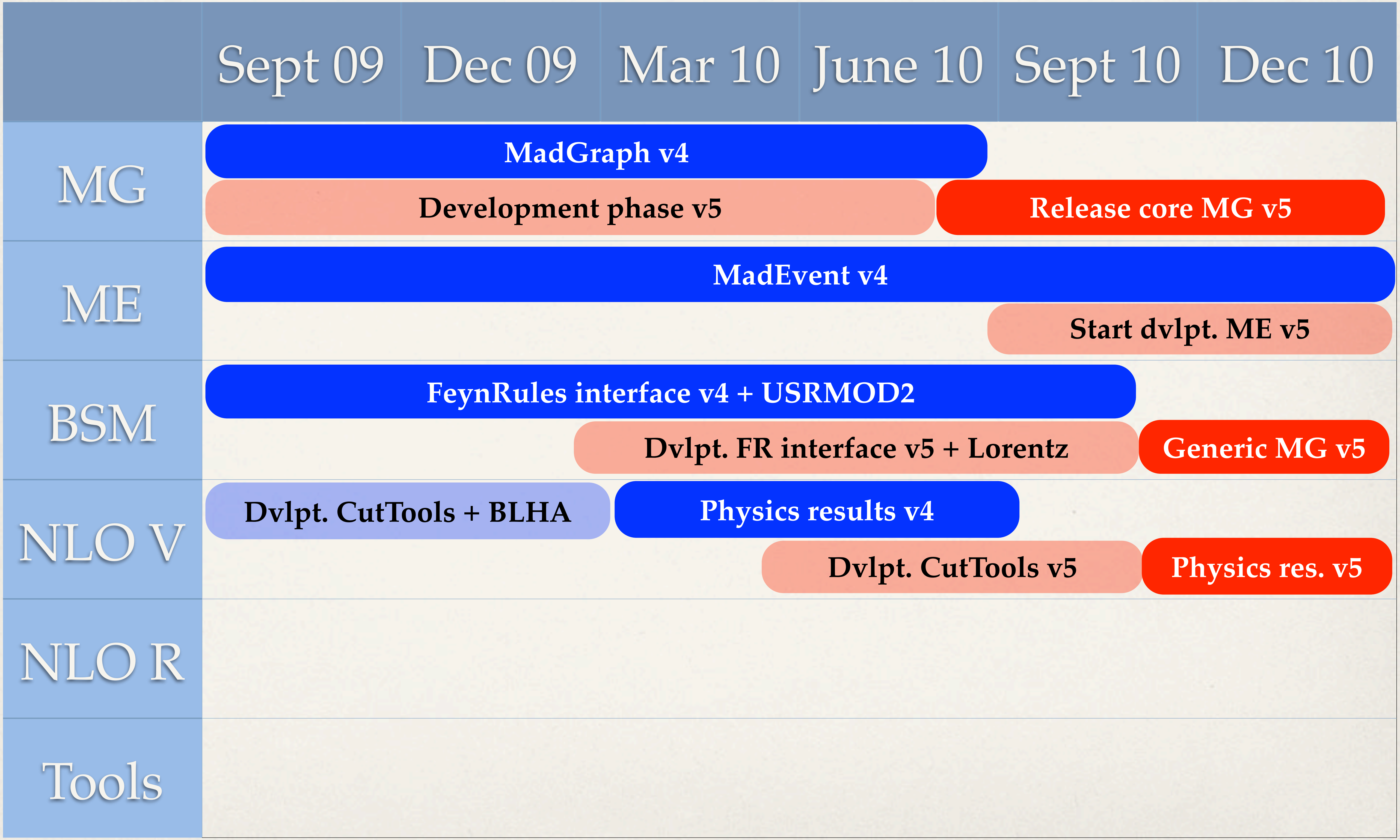
Timeline

V4 V5



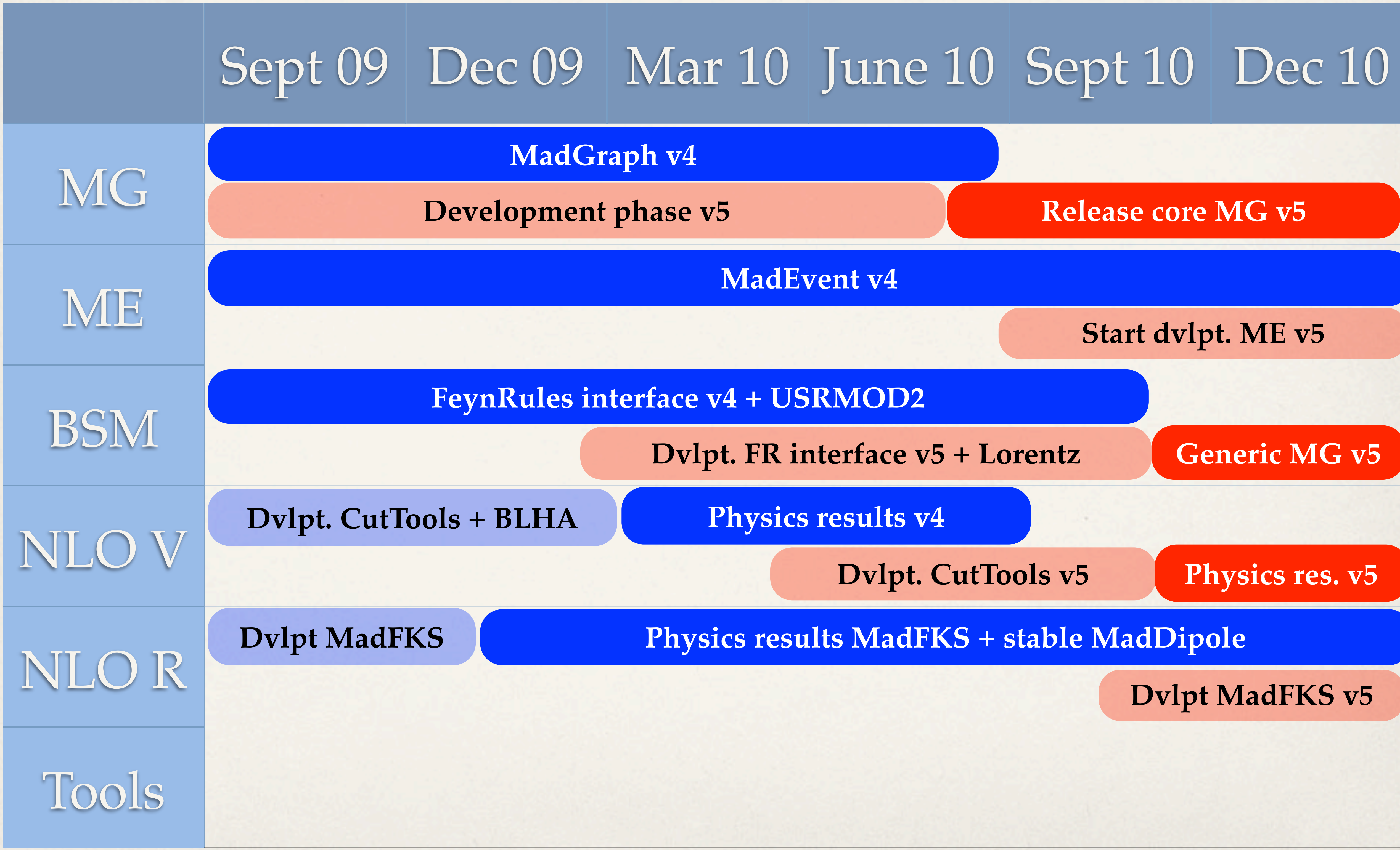
Timeline

V4 V5



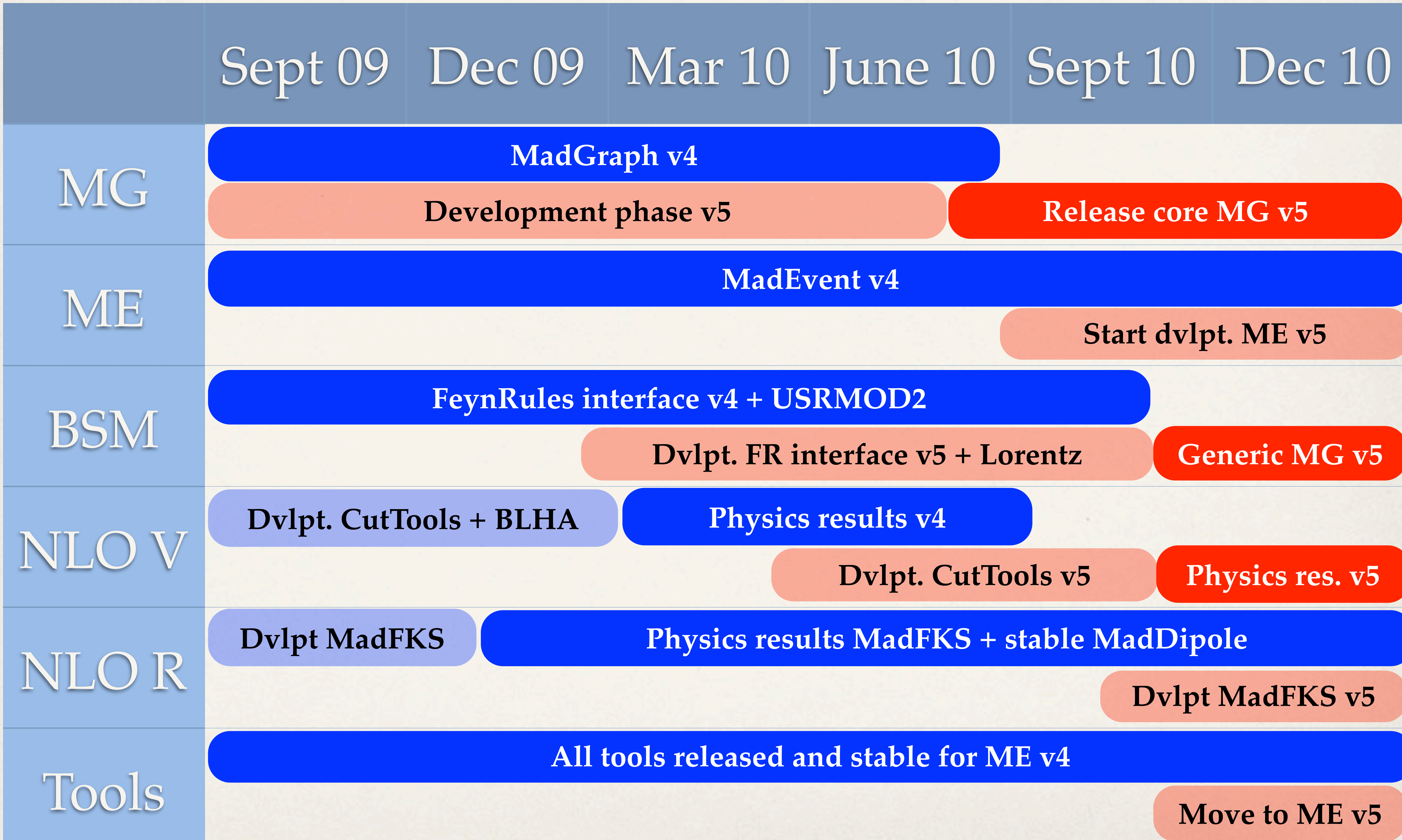
Timeline

V4 V5



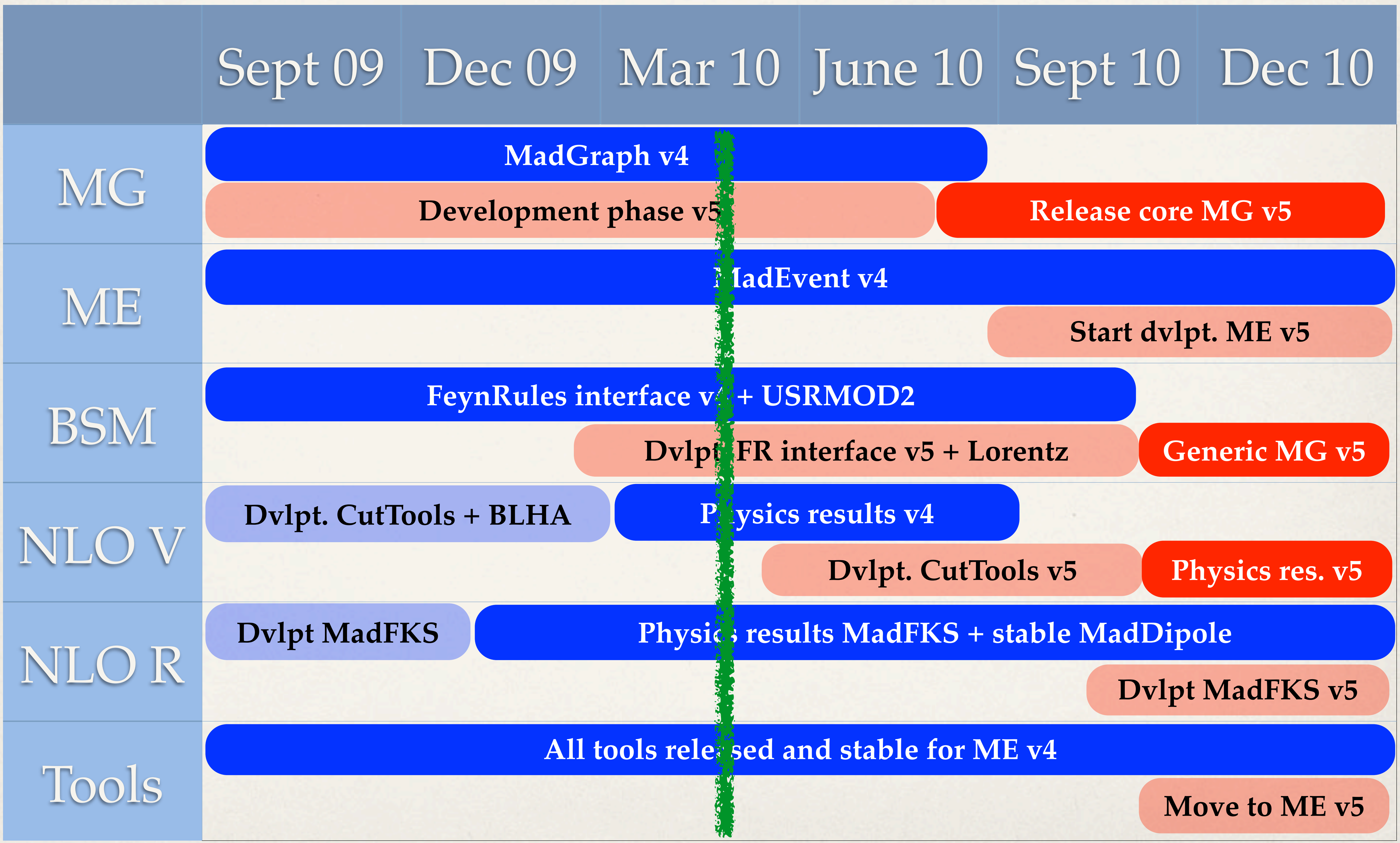
Timeline

V4 V5



Timeline

V4 V5



To bring back home…

To bring back home...

- MG/ME v4 is now a mature, well established and stable code coming with several features for BSM and QCD physics, and numerous peripheral tools

To bring back home...

- MG/ME v4 is now a mature, well established and stable code coming with several features for BSM and QCD physics, and numerous peripheral tools

- MG/ME v5 is behind the corner, with important and unprecedented improvements in all directions. Stable release of core MadGraph v5 by summer.

Thanks!