Introduction to MadGraph/MadEvent 5 part 2

Rikkert Frederix
University of Zurich

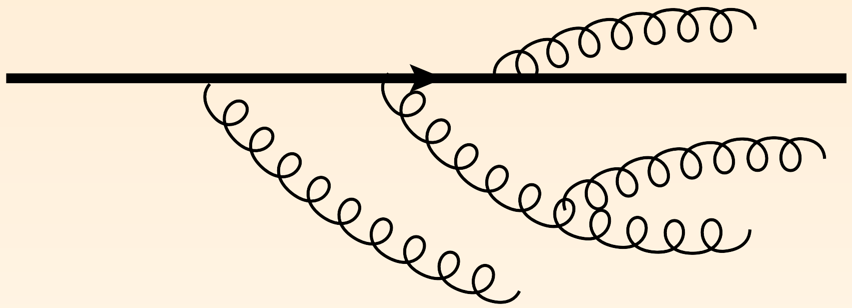
2011 IPMU-YITP School on Monte Carlo Tools for the LHC, Kyoto, Japan, September 5-9, 2011

Outline

- MadGraph/MadEvent beginner
- Advanced user
  - Interface to parton shower
  - Under the hood
  - New physics
  - Future

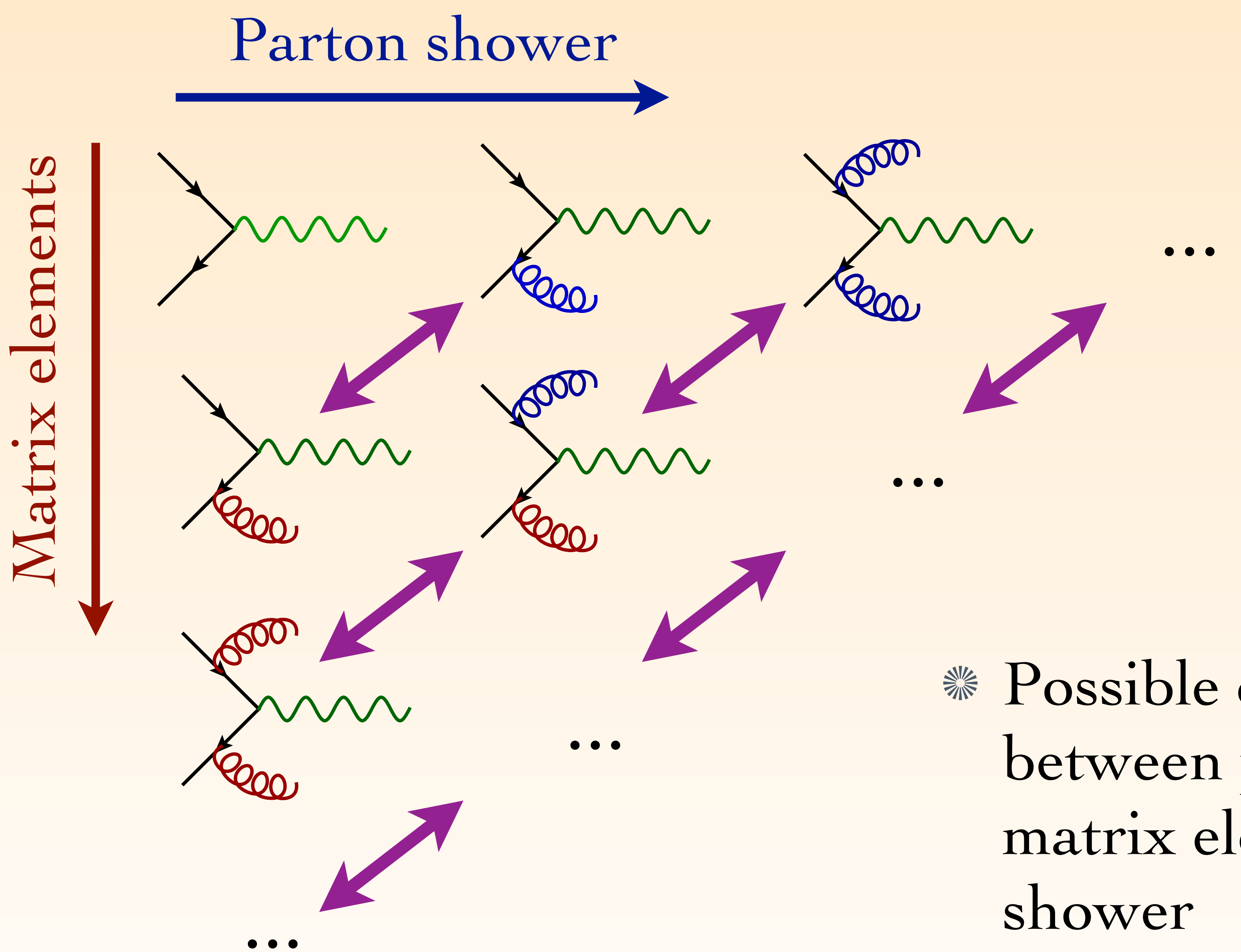
Parton showering

- For the description of any exclusive final state (that can be passed to a hadronization model and detector simulation) partons need to be showered



- MadGraph has no in-house parton shower
  - The LHE events that can be downloaded from the MadGraph website follow the Les Houches standard, and can therefore be showered by any parton shower available
- From the MadGraph website, interface to Pythia is available

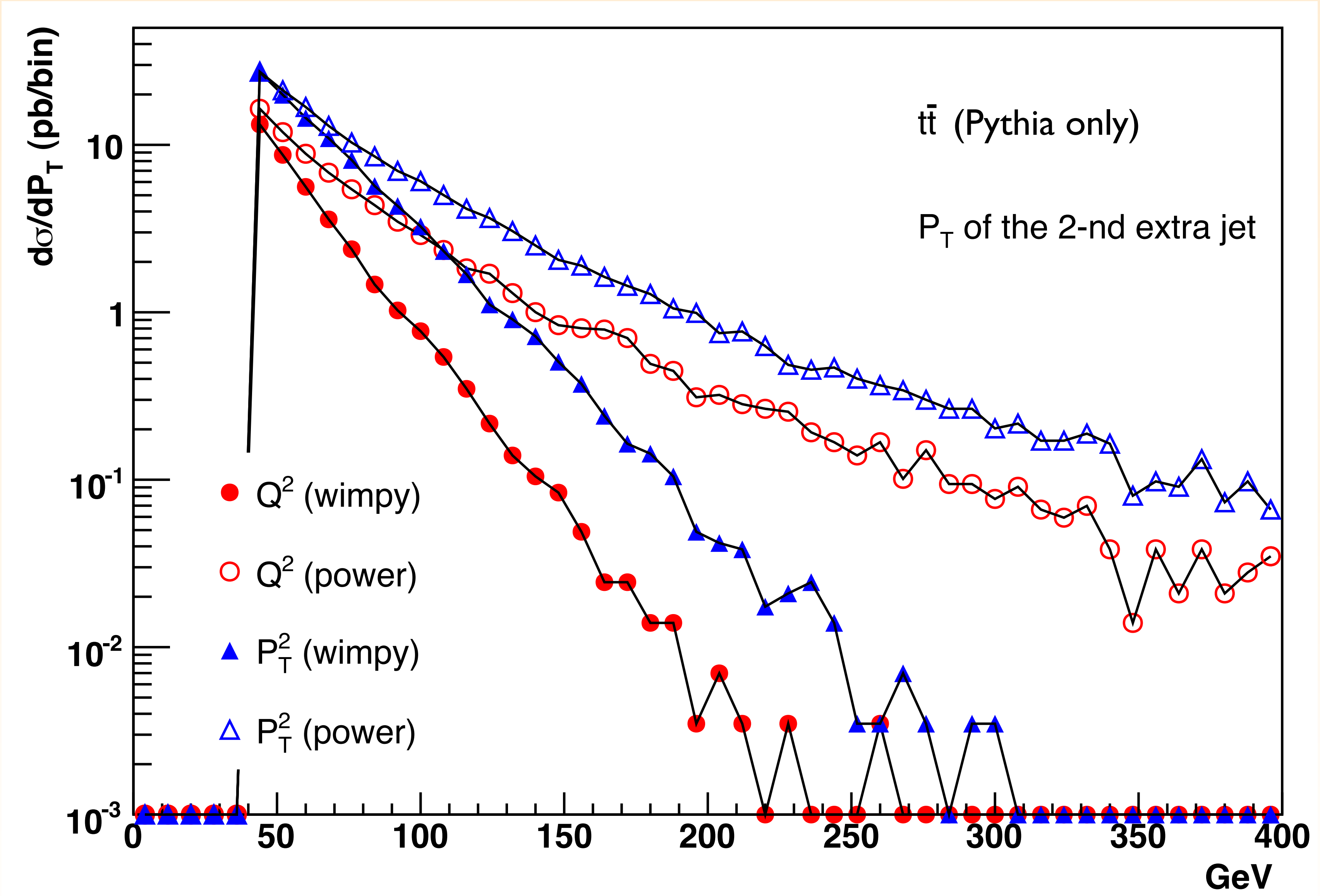
Possible double counting



- Possible double counting between partons from matrix elements and parton shower
- Use MLM prescription

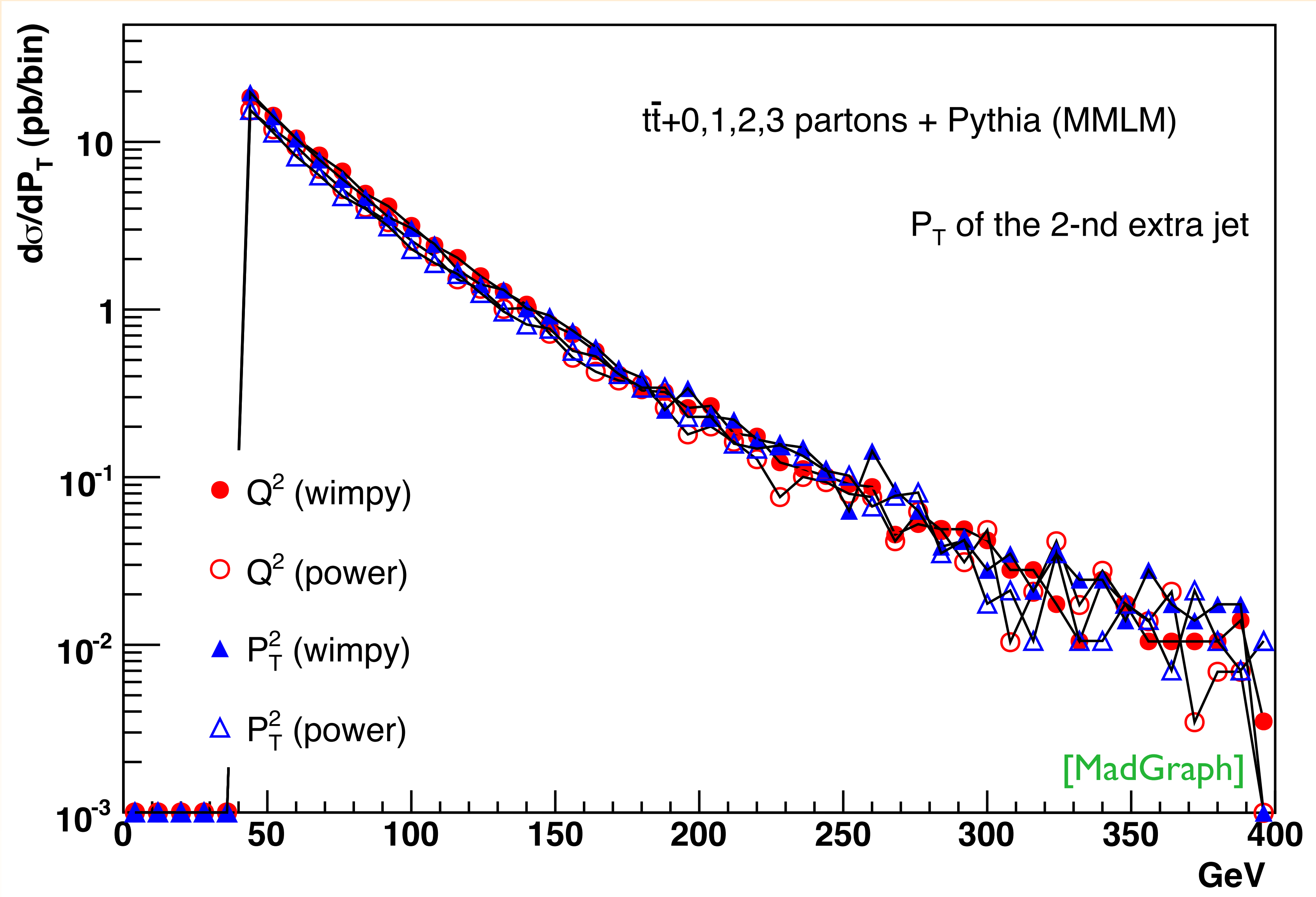
Need for matching

- Transverse momentum of the 2-nd extra jet in top pair production without matching: (too) much room for tuning



Need for matching

- This uncertainty is greatly reduced with the matrix-element parton-shower matching



Parton Shower & Detector simulation

- When using the MadGraph interface to Pythia, the multi-particle matrix-element parton-shower matching is available (MLM)
  - Directly available from the MadGraph websites
- Interface to (simplified) detector simulation also directly available from the MadGraph websites:
  - PGS ("pretty good simulations")
  - Delphes

Matching is automated

- Matching is automatically done when running through the MadEvent/Pythia interface
  - Example: simulation of e^+e^- with 0, 1, 2, 3 ME jets

```
proc_card.dat

generate p p > e+ e-      @0
add process p p > e+ e- j   @1
add process p p > e+ e- j j   @2
add process p p > e+ e- j j j   @3
```

```
run_card.dat
...
   1 = ickkw
...
  15 = xqcut
...
```

Matching on

Matching scale in GeV (cut between ME and PS)

Read more on the MadGraph wiki

Execises IV

- Generate events for the signal for Higgs boson production via gluon fusion at the LHC, p p > H, H -> e- ve~ mu+ vm
- Generate the backgrounds with the same final state ("non-reducible backgrounds")
- Run Pythia and the PGS detector simulation
- Compare the differences between the results at parton and detector level
- Think about which other (reducible) backgrounds might be important

Outline

- MadGraph/MadEvent beginner
- Advanced user
  - Interface to parton shower
  - Under the hood
  - New physics
  - Future

Under the hood

Algorithms

- Let us have a closer look at 2 crucial internal algorithms
  - Diagram generation
  - Writing of the amplitudes

Diagram generation

1. Generate hash maps (called libraries in Python) to map possible combinations of particles to their corresponding interactions
2. Start from external particles, and create all possible groupings of these particles
   - If all particles can be grouped → a diagram has been formed
   - If only two (the same) particles left → a diagram has been formed
   - The grouped particles form new "external" particles
   - Only keep combinations in which at least two groupings where performed in this step
3. Iterate step 2

Example: diagram generation

1st iteration	Groupings	After replacements	Result
$e^-, e^+, u, \bar{u}, g$	$(e^-, e^+), u, \bar{u}, g$	$(\gamma), u, \bar{u}, g$	Failed (only 1 FG=`True`)
		$(Z), u, \bar{u}, g$	Failed (only 1 FG=`True`)
	$e^-, e^+, (u, \bar{u}), g$	$e^-, e^+, (\gamma), g$	Failed (only 1 FG=`True`)
		$e^-, e^+, (Z), g$	Failed (only 1 FG=`True`)
		$e^-, e^+, (g), g$	Failed (only 1 FG=`True`)
	$e^-, e^+, (u, g), \bar{u}$	$e^-, e^+, (u), \bar{u}$	Failed (only 1 FG=`True`)
	$e^-, e^+, u, (\bar{u}, g)$	$e^-, e^+, u, (\bar{u})$	Failed (only 1 FG=`True`)
	$(e^-, e^+), (u, \bar{u}), g$	$(\gamma), (\gamma), g$	Failed (no vertex)
		$(\gamma), (Z), g$	Failed (no vertex)
		$(\gamma), (g), g$	Failed (no vertex)
		$(Z), (\gamma), g$	Failed (no vertex)
		$(Z), (Z), g$	Failed (no vertex)
		$(Z), (g), g$	Failed (no vertex)
	$(e^-, e^+), (u, g), \bar{u}$	$(\gamma), (u), \bar{u}$	Diagram 1
		$(Z), (u), \bar{u}$	Diagram 2
	$(e^-, e^+), u, (\bar{u}, g)$	$(\gamma), u, (\bar{u})$	Diagram 3
		$(Z), u, (\bar{u})$	Diagram 4

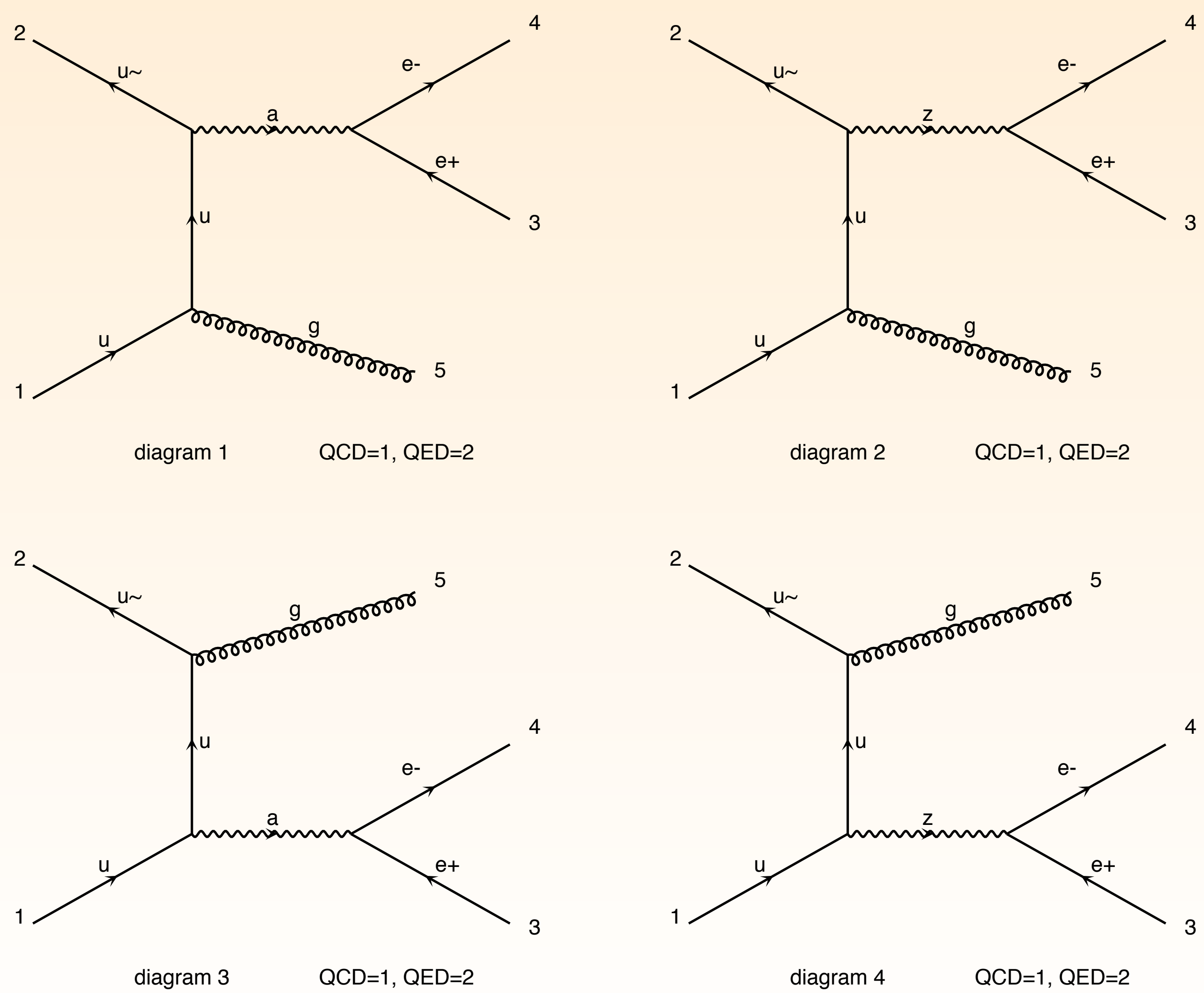
Performance

- The algorithm described above essentially uses only the "dictionary" syntax of Python
  - Highly optimized Python code
  - Trivially extended to include higher dimension (multiplicity) vertices

Process	MadGraph 4	MadGraph 5	Subprocesses	Diagrams
$pp \to jjj$	29.0 s	25.8 s	34	307
$pp \to jjl^+l^-$	341 s	103 s	108	1216
$pp \to jjje^+e^-$	1150 s	134 s	141	9012
$u\bar{u} \to e^+e^-e^+e^-e^+e^-$	772 s	242 s	1	3474
$gg \to ggggg$	2788 s	1050 s	1	7245
$pp \to jj(W^+ \to l^+\nu_l)$	146 s	25.7 s	82	304
$pp \to t\bar{t}$+full decays	5640 s	15.7 s	27	45
$pp \to \tilde{q}/\tilde{g}\,\tilde{q}/\tilde{g}$	222 s	107 s	313	475
7 particle decay chain	383 s	13.9 s	1	6
$gg \to (\tilde{g} \to u\bar{u}\tilde{\chi}_1^0)(\tilde{g} \to u\bar{u}\tilde{\chi}_1^0)$	70 s	13.9 s	1	48
$pp \to (\tilde{g} \to jj\tilde{\chi}_1^0)(\tilde{g} \to jj\tilde{\chi}_1^0)$	—	251 s	144	11008

Writing of the amplitudes

- MadGraph uses the helicity method for computing diagrams
- Completely numerical method
- Build on the HELAS library




```
C     ----------
C     BEGIN CODE
C     ----------
      CALL IXXXXX(P(0,1),ZERO,NHEL(1),+1*IC(1),W(1,1))
      CALL OXXXXX(P(0,2),ZERO,NHEL(2),-1*IC(2),W(1,2))
      CALL IXXXXX(P(0,3),ZERO,NHEL(3),-1*IC(3),W(1,3))
      CALL OXXXXX(P(0,4),ZERO,NHEL(4),+1*IC(4),W(1,4))
      CALL VXXXXX(P(0,5),ZERO,NHEL(5),+1*IC(5),W(1,5))
      CALL FFV1_2(W(1,1),W(1,5),GC_5,ZERO, ZERO, W(1,6))
      CALL FFV1_3(W(1,3),W(1,4),GC_3,ZERO, ZERO, W(1,7))
C     Amplitude(s) for diagram number 1
      CALL FFV1_0(W(1,6),W(1,2),W(1,7),GC_2,AMP(1))
      CALL FFV2_4_3(W(1,3),W(1,4),GC_21,GC_24,MZ, WZ, W(1,8))
C     Amplitude(s) for diagram number 2
      CALL FFV2_5_0(W(1,6),W(1,2),W(1,8),GC_22,GC_23,AMP(2))
      CALL FFV1_1(W(1,2),W(1,5),GC_5,ZERO, ZERO, W(1,9))
C     Amplitude(s) for diagram number 3
      CALL FFV1_0(W(1,1),W(1,9),W(1,7),GC_2,AMP(3))
C     Amplitude(s) for diagram number 4
      CALL FFV2_5_0(W(1,1),W(1,9),W(1,8),GC_22,GC_23,AMP(4))
      JAMP(1)=+AMP(1)+AMP(2)+AMP(3)+AMP(4)

      MATRIX1 = 0.D0
      DO I = 1, NCOLOR
        ZTEMP = (0.D0,0.D0)
        DO J = 1, NCOLOR
          ZTEMP = ZTEMP + CF(J,I)*JAMP(J)
        ENDDO
        MATRIX1=MATRIX1+ZTEMP*DCONJG(JAMP(I))/DENOM(I)
      ENDDO
      AMP2(1)=AMP2(1)+AMP(1)*DCONJG(AMP(1))
      AMP2(2)=AMP2(2)+AMP(2)*DCONJG(AMP(2))
      AMP2(3)=AMP2(3)+AMP(3)*DCONJG(AMP(3))
      AMP2(4)=AMP2(4)+AMP(4)*DCONJG(AMP(4))
      DO I = 1, NCOLOR
        JAMP2(I)=JAMP2(I)+JAMP(I)*DCONJG(JAMP(I))
      ENDDO
```

Performance

Generation time for 10000 unweighted events

Process	Subproc. dirs.		Channels		Directory size		Event gen. time	
	MG 4	MG 5	MG 4	MG 5	MG 4	MG 5	MG 4	MG 5
$pp \to W^+j$	6	2	12	4	79 MB	35 MB	3:15 min	1:55 min
$pp \to W^+jj$	41	4	138	24	438 MB	64 MB	9:15 min	4:19 min
$pp \to W^+jjj$	73	5	1164	120	842 MB	110 MB	21:41 min*	8:14 min*
$pp \to W^+jjjj$	296	7	15029	609	3.8 GB	352 MB	2:54 h*	46:50 min*
$pp \to W^+jjjjj$	-	8	-	2976	-	1.5 GB	-	11:39 h*
$pp \to l^+l^-j$	12	2	48	8	149 MB	44 MB	21:46 min	3:00 min
$pp \to l^+l^-jj$	54	4	586	48	612 MB	83 MB	2:40 h	11:52 min
$pp \to l^+l^-jjj$	86	5	5408	240	1.2 GB	151 MB	49:18 min*	16:38 min*
$pp \to l^+l^-jjjj$	235	7	65472	1218	5.3 GB	662 MB	7:16 h*	2:45 h*
$pp \to t\bar{t}$	3	2	5	3	49 MB	39 MB	2:39 min	1:55 min
$pp \to t\bar{t}j$	7	3	45	17	97 MB	56 MB	10:24 min	3:52 min
$pp \to t\bar{t}jj$	22	5	417	103	274 MB	98 MB	1:50 h	32:37 min
$pp \to t\bar{t}jjj$	34	6	3816	545	620 MB	209 MB	2:45 h*	23:15 min*

* run on a cluster

BSM with Aloha

- For BSM physics that includes interactions between particles for which the Lorentz structure is not SM-like, new HELAS subroutines need to be written
- In theory a simple task, but in practice it's very dull and it needs a lot of debugging to get it correct
- Aloha can generate these new HELAS subroutine automatically starting from the Model file
- Any Lorentz structure allowed for spin-0, 1/2, 1 and 2 particles (also higher dimensional)

Exercises V

- Download the MadGraph 5 code, untar it and execute

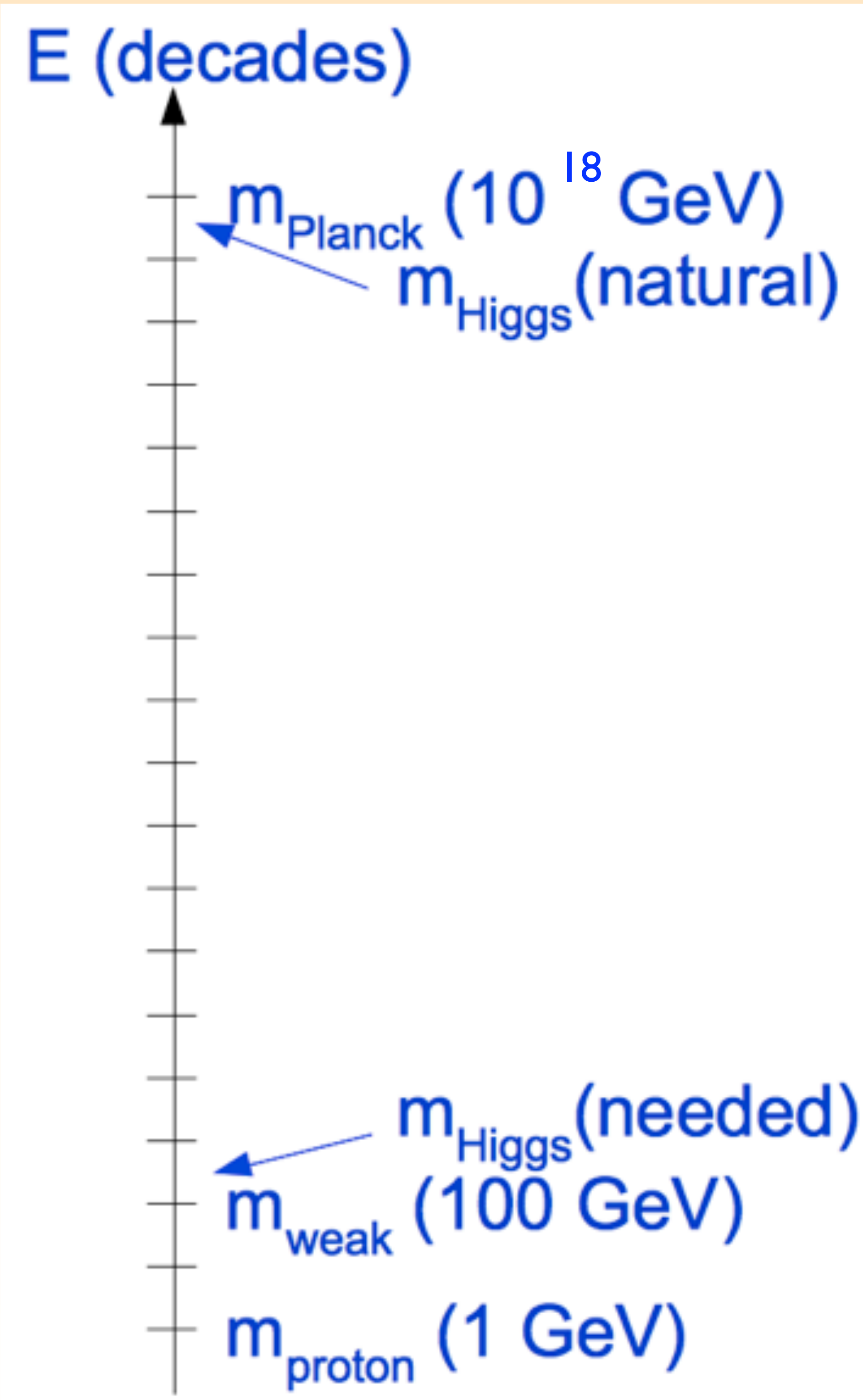
  ./bin/mg5

  This will enter the interactive mode of the MadGraph 5 code
- Start the tutorial and follow it...
  (note that there is tab-completion, like in a standard linux shell)

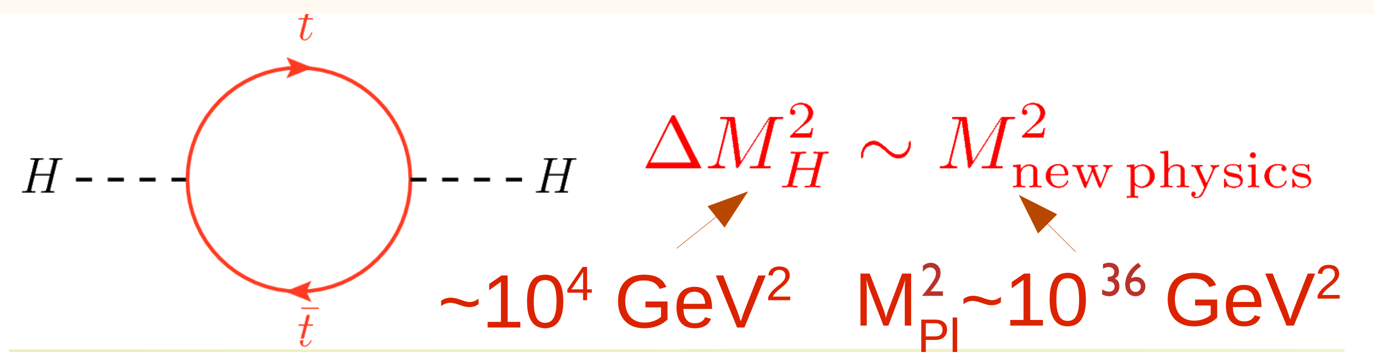
Outline

- MadGraph/MadEvent beginner
- Advanced user
  - Interface to parton shower
  - Under the hood
  - New physics
  - Future

Why New Phyics?

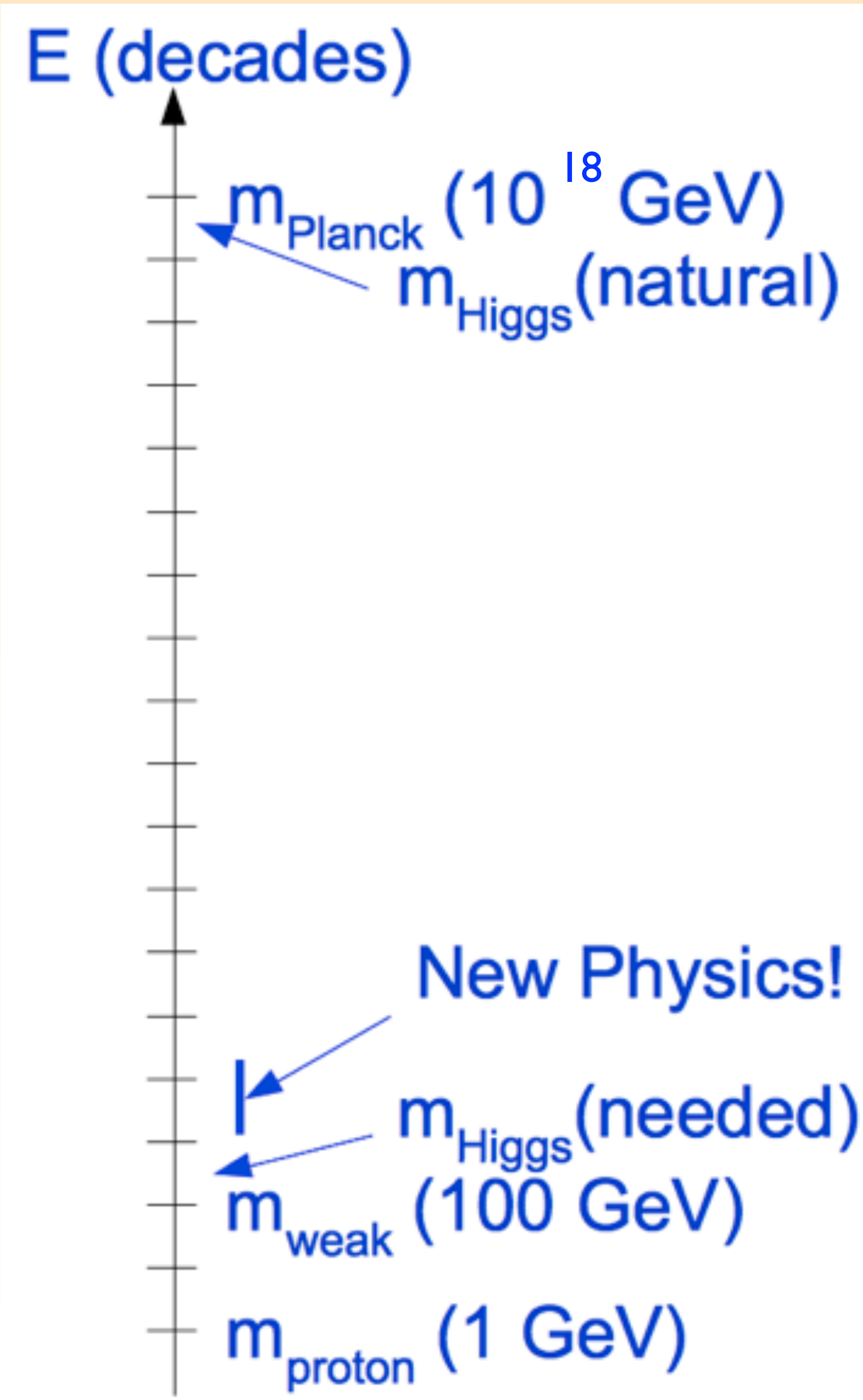


- Higgs bosom mass is naturally at the mass of new physics (only known new physics scale is the Planck scale at 10^{18} GeV)
- Standard model works if Higgs mass is below ~800 GeV
- New physics scale communicated through quantum loop contributions to Higgs mass

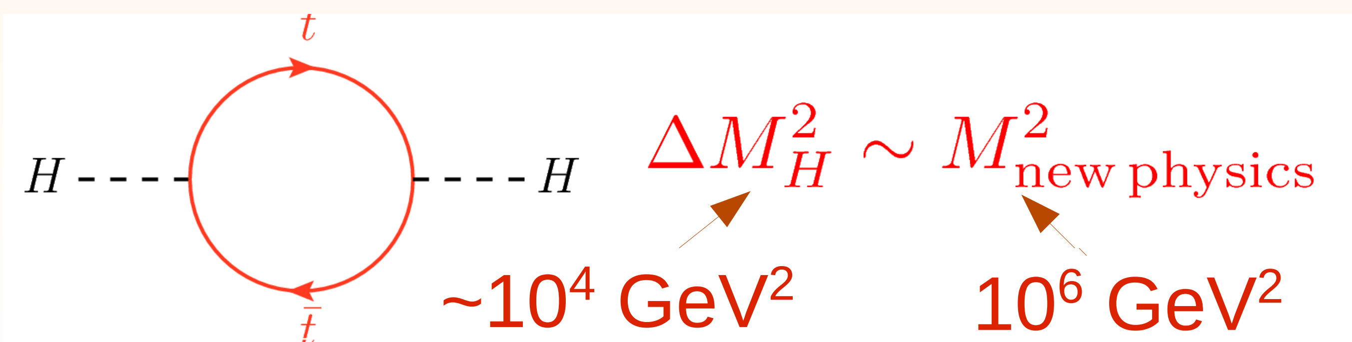


Rikkert Frederix, University of Zurich

Why New Phyics?



- ΔM_H^2 contribution must be cancelled by the bare mass term. For fine-tuning less than 1%, need New Physics which cuts off the quadratic loops at 1 TeV



Rikkert Frederix, University of Zurich

Why New Phyics?

- The hierarchy problem, together with Dark Matter (and, to some extend, Grand Unification) have been the driving force behind New Physics model building in the past 30 years
  - Supersymmetry
  - Large extra dimensions
  - Randall-Sundrum (warped extra dimensions)
  - Little Higgs theories
  - ... (mostly variants/combinations)

Specification of a physics model

- A (new) physics model is normally defined by:
  - Field content + Lagrangian
  - Particle content + Feynman rules + coupling definitions

→ Suitable for Matrix Element generators

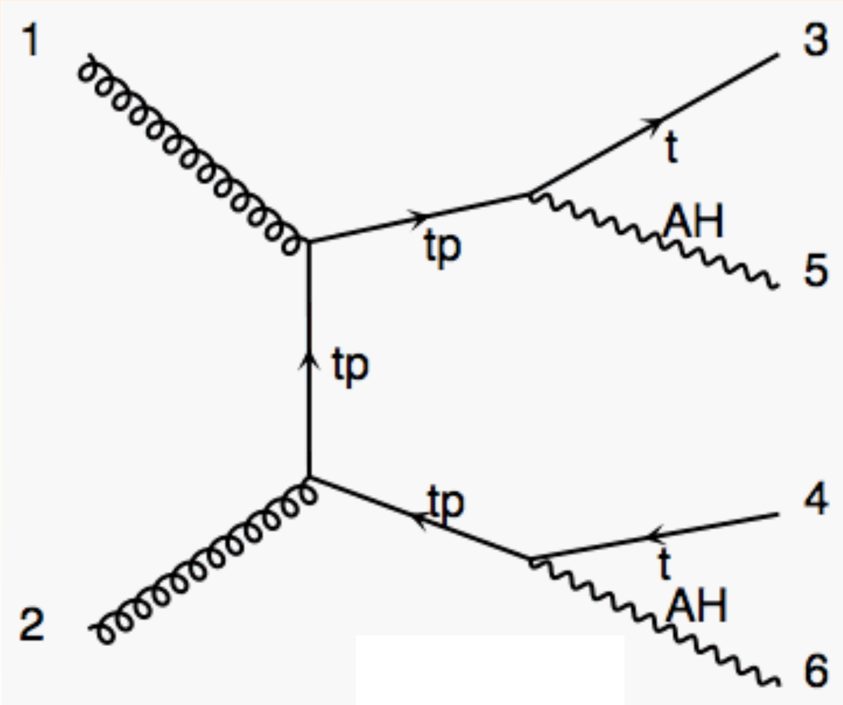
+ parameters, masses and decay widths

Implementing New Physics

- Three ways to implement new physics in MadGraph
  - Modify an existing model (e.g. changing only a coupling or a mass)
  - User model framework (new particles/interactions)
    1. Add new particles
    2. Add new interactions
    3. Enter expressions for the new couplings
    4. A script generates all Fortran files
  - FeynRules
    Mathematica package to translate Lagrangian into MadGraph (among others) friendly input

User Model implementation

- User model framework
  - Start from the Standard Model (`./models/usermod_v4`)
  - Easy and quick implementation when the complexity of the added sector is not too large
  - Only SM-like interactions
  - Example:
    A QCD t' pair production with $t' \to A_H t$ in Little Higgs model with T-parity
$A_H^\mu \bar{t}'\_t$	$\frac{2ig'}{5} c_\lambda \gamma_\mu \left(c_\lambda \frac{v}{f} P_L + P_R \right)$



User Model implementation

- User model framework
  - Specify particles and interactions

```
particles.dat
#Name anti_Name  Spin   Linetype Mass  Width Color Label Model
#xxx  xxxx       SFV     WSDC    str   str   STO    str  PDG code
#MODEL EXTENSION
tp       tp~      F       S      TPMASS TPWID T     TP     8
zp       zp       V       W      ZPMASS ZPWID S     ZP    32
# END
```

```
interactions.dat
#    USRVertex
tp tp g  GG    QCD
tp t  zp GTPZP QED
t  tp zp GTPZP QED
```

  - Run script; specify couplings

```
couplings.f
c***********************************
c UserMode couplings
c***********************************

      GTPZP(1)=dcmplx(ee*param1,Zero)
      GTPZP(2)=dcmplx(ee*param1,Zero)
```

User Model implementation

- User model framework
  - Specify particles and interactions

```
particles.dat
#Name anti_Name  Spin   Linetype Mass  Width  Color Label Model
#xxx   xxxx      SFV     WSDC    str    str   STO    str  PDG code
#MODEL EXTENSION
tp       tp~      F       S      TPMASS TPWID  T     TP     8
zp       zp       V       W      ZPMASS ZPWID  S     ZP    32
# END
```

```
interactions.dat
#    USRVertex
tp tp g   GG     QCD
tp t   zp GTPZP QED
t   tp zp GTPZP QED
```

  - Run script; specify couplings

```
couplings.f
c************************************
c UserMode couplings
c************************************

      GTPZP(1)=dcmplx(ee*param1,Zero)
      GTPZP(2)=dcmplx(ee*param1,Zero)
```

And you're ready to generate the process and study its properties

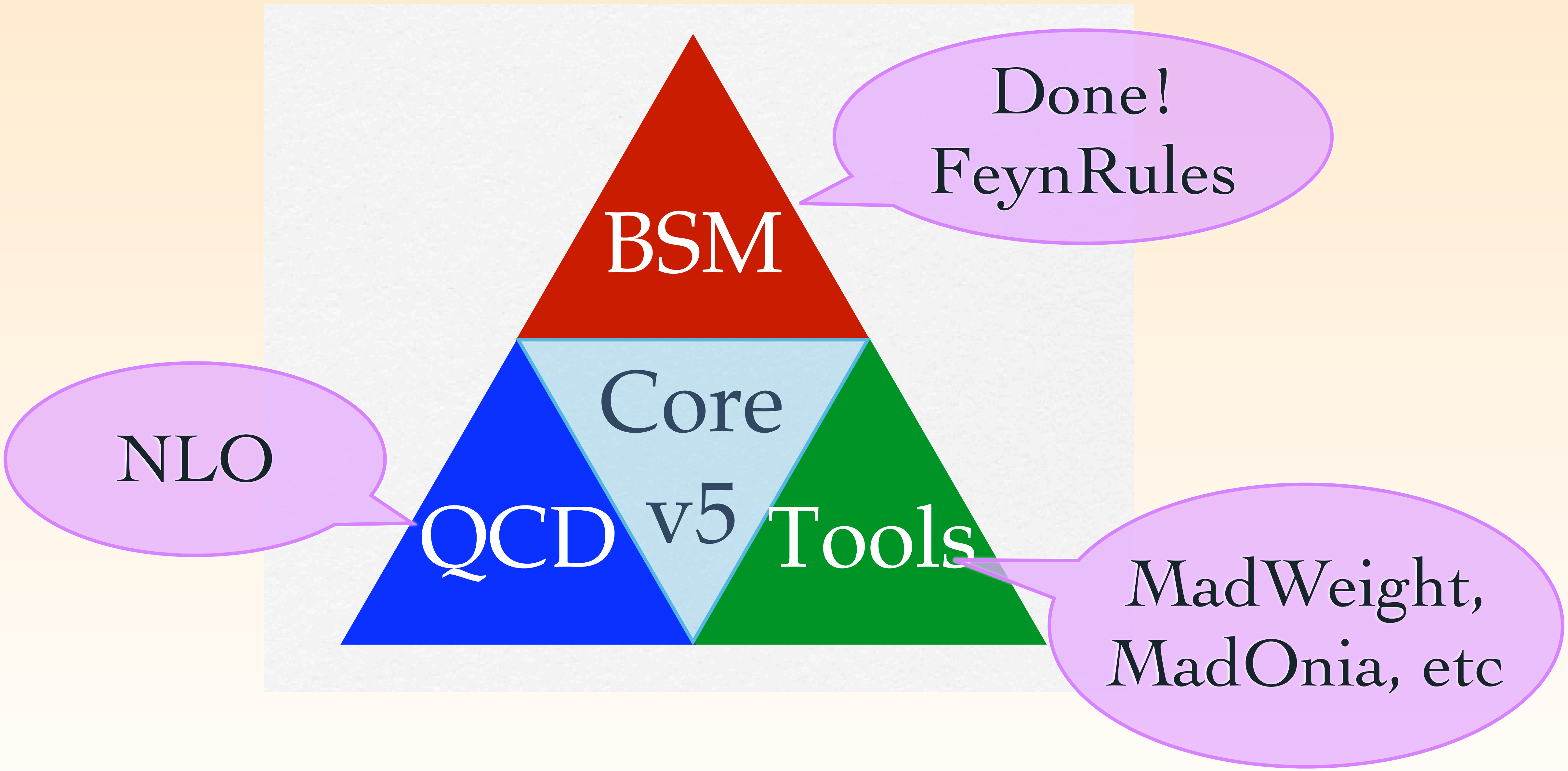
Implementing New Physics

- Three ways to implement new physics in MadGraph
  - Modify an existing model (e.g. changing only a coupling or a mass)
  - User model framework (new particles/interactions)
    1. Add new particles
    2. Add new interactions
    3. Enter expressions for the new couplings
    4. A script generates all Fortran files
  - FeynRules
    Mathematica package to translate Lagrangian into MadGraph (among others) friendly input
    ☞ Claude Duhr's lecture on Thursday

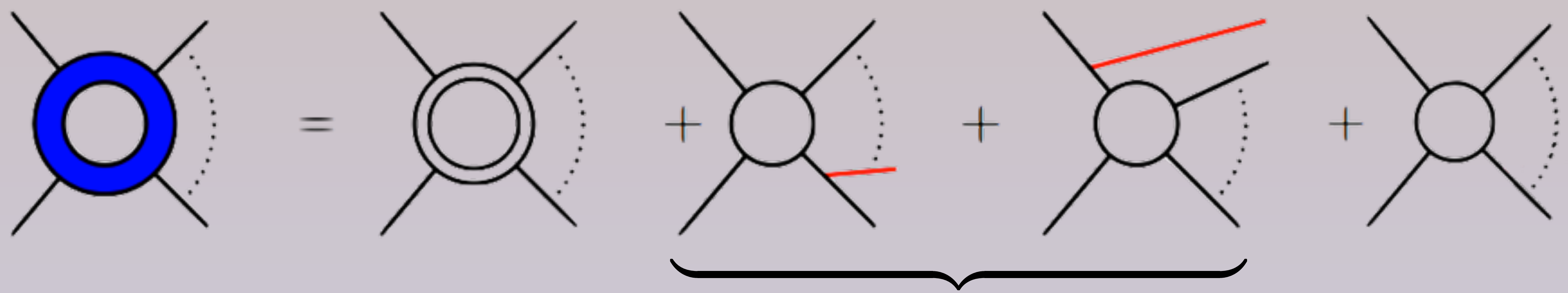
Outline

- MadGraph/MadEvent beginner
- Advanced user
  - Interface to parton shower
  - Under the hood
  - New physics
  - Future

Into the future



NLO computations



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

'Virtual' or 'one-loop' NLO corrections

'Real emission' NLO corrections

'Born' or 'LO' contribution

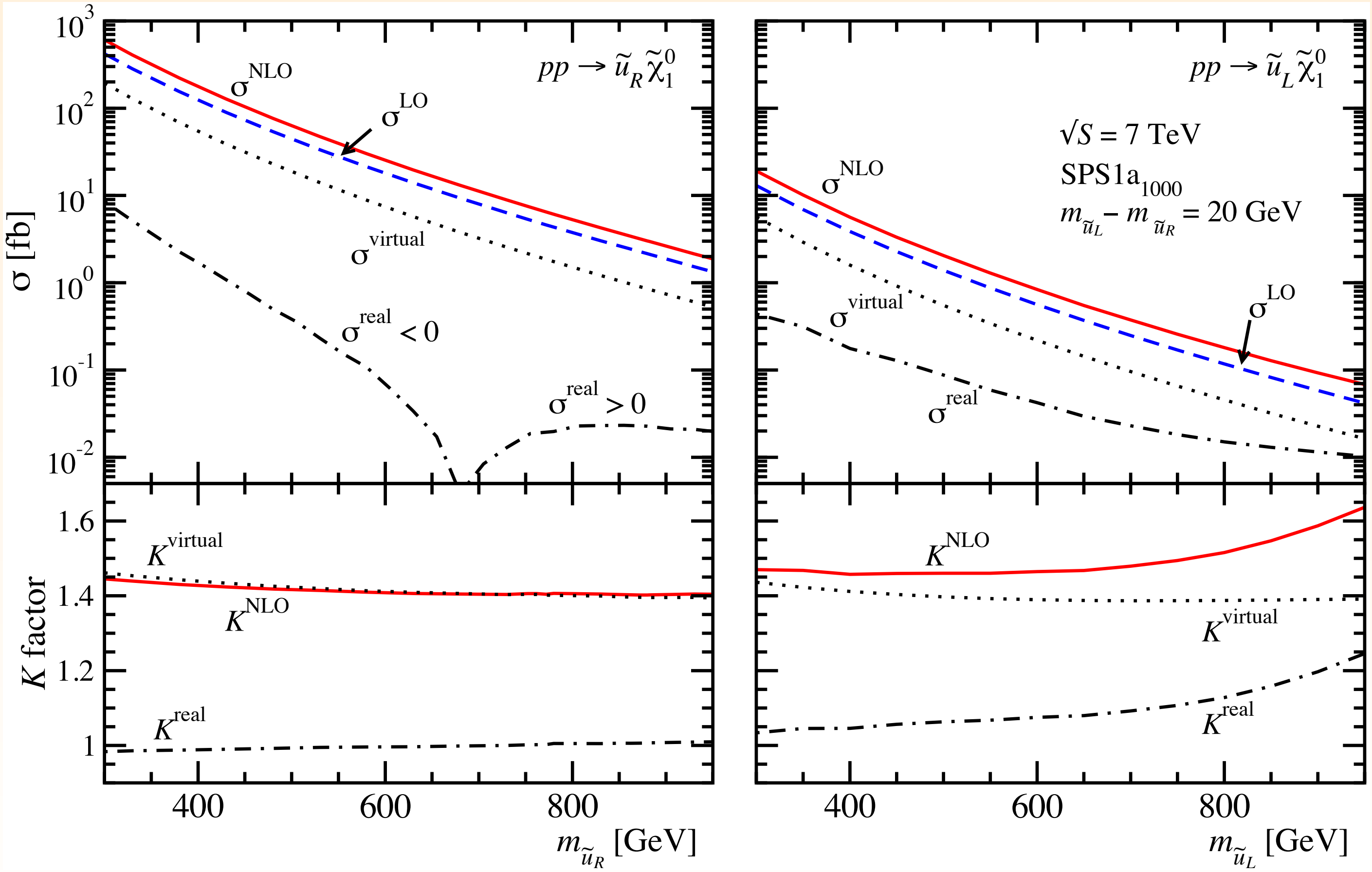
AMC@NLO

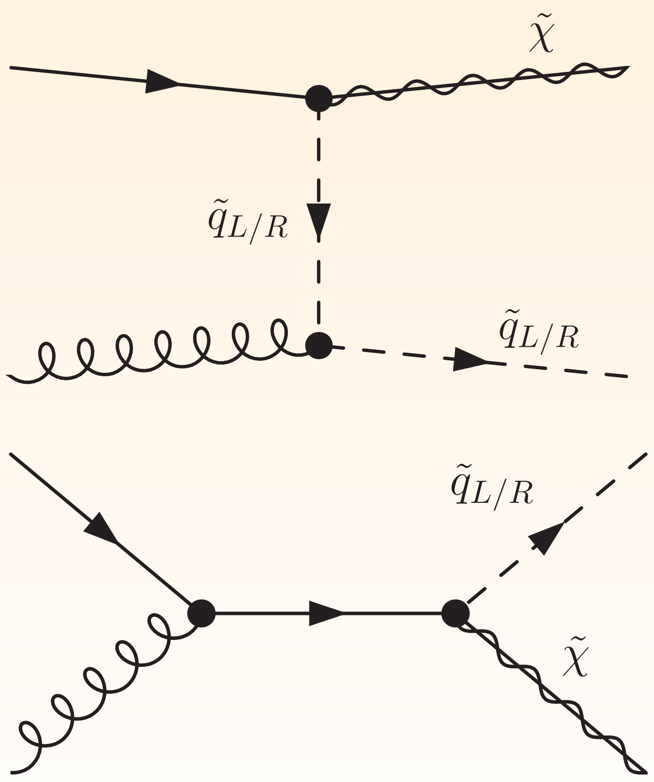
- Package to generate unweighted NLO events (using the MC@NLO method) within the Standard Model in a completely automatic way
- It uses MadLoop + CutTools to compute the virtual corrections
- MadFKS for the Real-emission corrections
- Working package in MadGraph v4
- Currently working on an improved implementation in MadGraph v5 --> we will only go public when this is done
- website: http://amcatnlo.cern.ch

MadGolem

☞ Kentarou Matawari's talk on Saturday

- Package to generate distributions for observables automatically for BSM physics
- MadDipole for the real-emission; Golem for the virtuals
- First results obtained for Squark-Neutralino production





Summary

- MadGraph is a parton-level event generator interfaced to parton showers and detector simulation
- Efficient code that can be run via the web on our clusters
- Running locally gives more freedom: implementing new Physics Models using usrmod or FeynRules made easy
- The new MadGraph version 5 is already a mature, well-tested code
  - All core features of MadGraph 4 are available in MG5
- Publicly available automatic NLO event generation available soon