What is MadGraph 5?

(and what is all the fuss about?)

MadGraph 5

J.A., Herquet, Maltoni, Mattelaer, Stelzer, arXiv:1106.0522

- MadGraph 5 is the successor of the well-known MadGraph matrix element generator
- Original MadGraph by Tim Stelzer was written in Fortran, first version from 1994 hep-ph/9401258
- Event generation by MadEvent using the single diagram enhanced multichannel integration technique in 2002 (Stelzer, Maltoni) hep-ph/0208156
- Full support for BSM (and many other improvements) in MG/ME 4 (2006-2008)

arXiv:0706.2334, arXiv:0809.2410

MadGraph 5

J.A., Herquet, Maltoni, Mattelaer, Stelzer, arXiv:1106.0522

- MadGraph 5 development started in 2009
- Language chosen: Python
  - ➡ Exceptional flexibility and speed where needed
  - ➡ Easy to create modular structure (OO)
  - ➡ Automatic documentation
  - ➡ Easy to incorporate test suite
- New algorithms with exceptional speedups and generalizations for every aspect of the code

Diagram generation speed benchmarks

Full MadEvent subprocess directory output
Computer: Sony Vaio TZ laptop

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
$gg \to (\tilde g \to u\bar u\tilde\chi^0_1)(\tilde g \to u\bar u\tilde\chi^0_1)$	70 s	13.9 s	1	48
$pp \to (\tilde g \to jj\tilde\chi^0_1)(\tilde g \to jj\tilde\chi^0_1)$	>> 10^7 years	251 s	144	11008

$$gg \to (\tilde g \to u(\tilde{\bar u}_l \to \bar u(\tilde\chi^0_2 \to Z\tilde\chi^0_1)))(\tilde g \to u\tilde d\tilde\chi^-_1)$$

Diagram generation speed benchmarks

Full MadEvent subprocess directory output
Computer: Sony Vaio TZ laptop

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
$pp \to (\tilde{g} \to jj\tilde{\chi}_1^0)(\tilde{g} \to jj\tilde{\chi}_1^0)$	>> 10^7 years	251 s	144	11008

$$gg \to (\tilde{g} \to u(\bar{\tilde{u}}_l \to \bar{u}(\tilde{\chi}_2^0 \to Z\tilde{\chi}_1^0)))(\tilde{g} \to u\tilde{d}\tilde{\chi}_1^-)$$

Diagram generation speed benchmarks

Full MadEvent subprocess directory output
Computer: Sony Vaio TZ laptop

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
$pp \to (\tilde{g} \to jj\tilde{\chi}_1^0)(\tilde{g} \to jj\tilde{\chi}_1^0)$	>> 10^7 years	251 s	144	11008

$$gg \to (\tilde{g} \to u(\tilde{\bar{u}}_l \to \bar{u}(\tilde{\chi}_2^0 \to Z\tilde{\chi}_1^0)))(\tilde{g} \to u\tilde{\bar{d}}\tilde{\chi}_1^-)$$

Diagram generation speed benchmarks

Full MadEvent subprocess directory output
Computer: Sony Vaio TZ laptop

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
$pp \to (\tilde{g} \to jj\tilde{\chi}_1^0)(\tilde{g} \to jj\tilde{\chi}_1^0)$	>> 10^7 years	251 s	144	11008

$gg \to (\tilde{g} \to u(\tilde{\bar{u}}_l \to \bar{u}(\tilde{\chi}_2^0 \to Z\tilde{\chi}_1^0)))(\tilde{g} \to u\tilde{\bar{d}}\tilde{\chi}_1^-)$

Diagram generation speed benchmarks

Full MadEvent subprocess directory output
Computer: Sony Vaio TZ laptop

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
$pp \to (\tilde{g} \to jj\tilde{\chi}_1^0)(\tilde{g} \to jj\tilde{\chi}_1^0)$	>> 10^7 years	251 s	144	11008

$gg \to (\tilde{g} \to u(\tilde{\bar{u}}_l \to \bar{u}(\tilde{\chi}_2^0 \to Z\tilde{\chi}_1^0)))(\tilde{g} \to u\tilde{\bar{d}}\tilde{\chi}_1^-)$

Diagram generation speed benchmarks

Full MadEvent subprocess directory output
Computer: Sony Vaio TZ laptop

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
$pp \to (\tilde{g} \to jj\tilde{\chi}_1^0)(\tilde{g} \to jj\tilde{\chi}_1^0)$	>> 10⁷ years	251 s	144	11008

$gg \to (\tilde{g} \to u(\tilde{\bar{u}}_l \to \bar{u}(\tilde{\chi}_2^0 \to Z\tilde{\chi}_1^0)))(\tilde{g} \to u\tilde{d}\tilde{\chi}_1^-)$

Super-fast decay chains allow completely new types of processes!

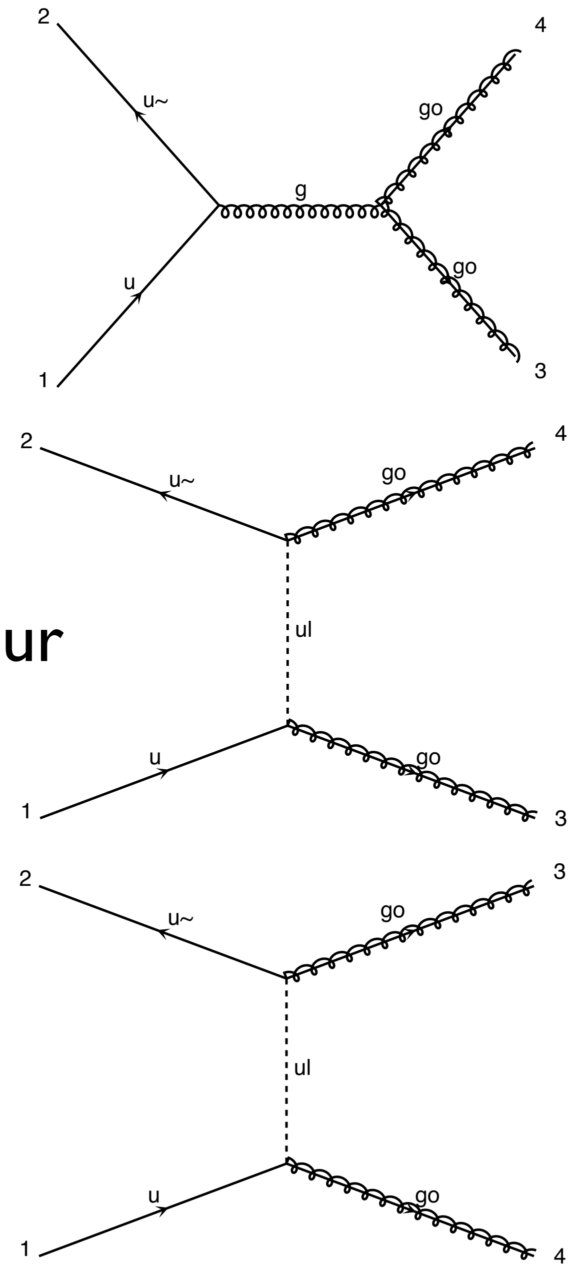
Decay chains

-
  ```
  p p > t t~ w+, (t > w+ b, w+ > l+ vl), \
                 (t~ > w- b~, w- > j j), \
                 w+ > l+ vl
  ```
- Separately generate core process and each decay
- Decays generated with the decaying particle as resulting wavefunction
- Iteratively combine decays and core processes
- Difficulty: Multiple diagrams in decays

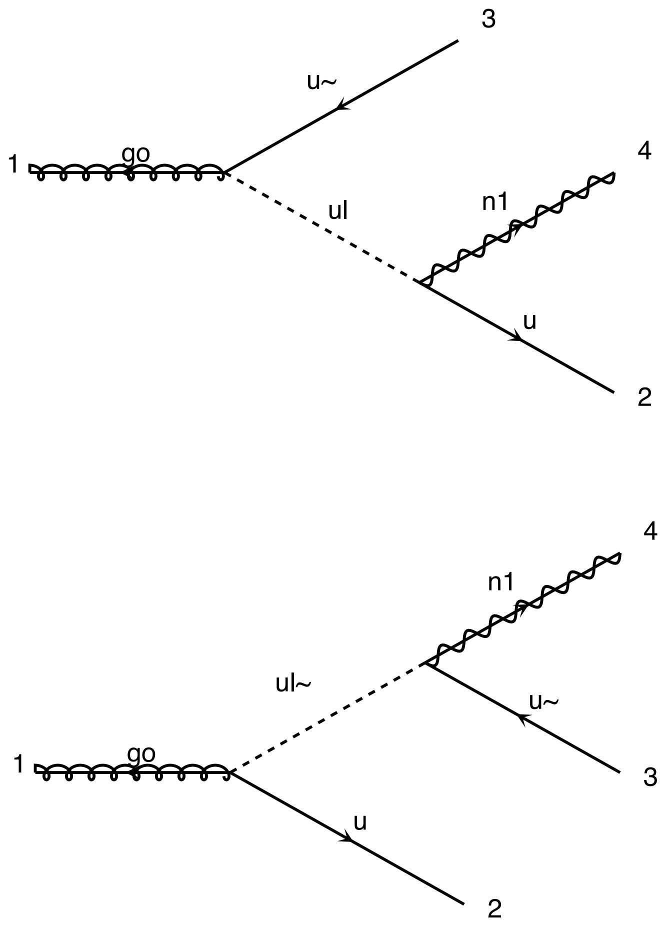
Decay chains

- If multiple diagrams in decays, need to multiply together core process and decay diagrams:

u u~ > go go / ur



x



2

go > u u~ n1 / ur

(to the second power since both gluinos decay)

Decay chains

- If multiple diagrams in decays, need to multiply together core process and decay diagrams:

u u~ > go go / ur, go > u u~ n1 / ur

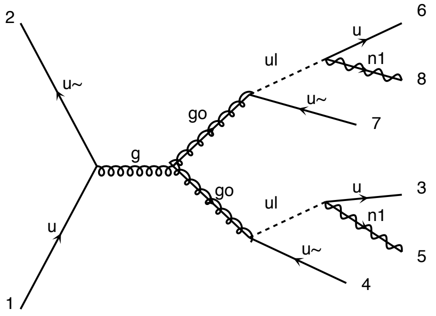


diagram 1 QCD=4, QED=2

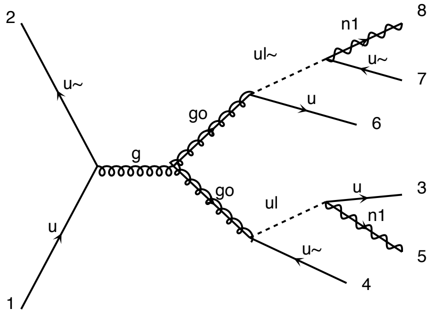


diagram 2 QCD=4, QED=2

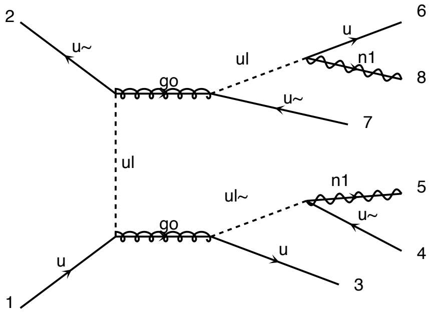


diagram 7 QCD=4, QED=2

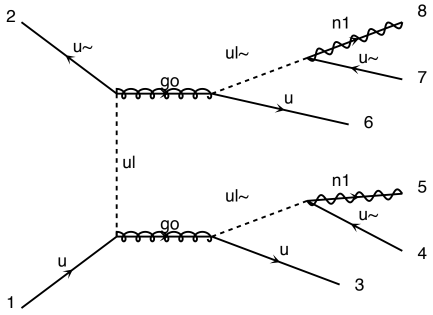


diagram 8 QCD=4, QED=2

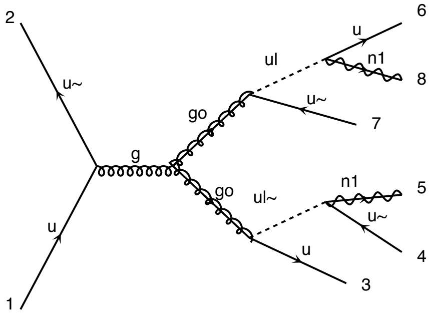


diagram 3 QCD=4, QED=2

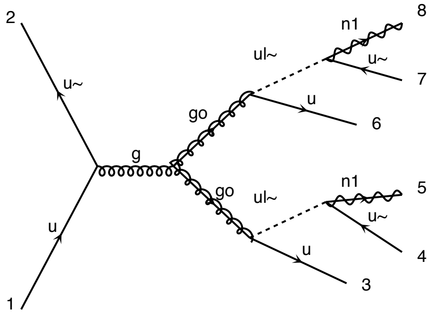


diagram 4 QCD=4, QED=2

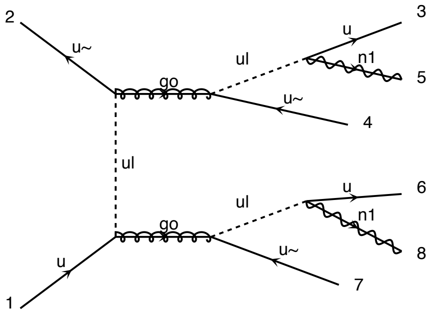


diagram 9 QCD=4, QED=2

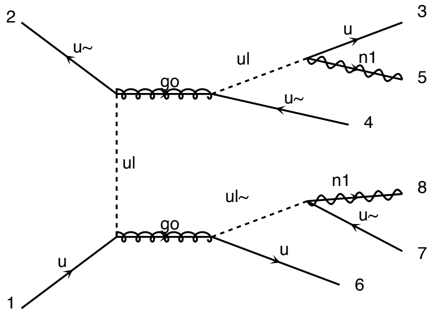


diagram 10 QCD=4, QED=2

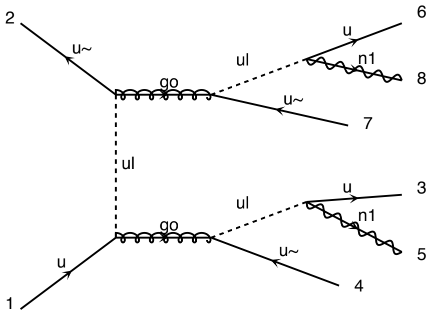


diagram 5 QCD=4, QED=2

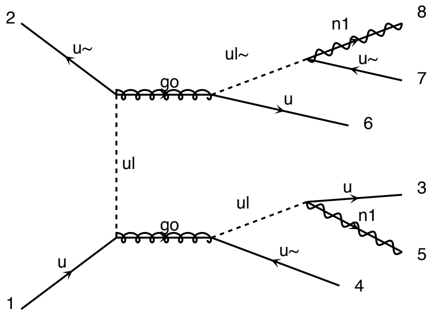


diagram 6 QCD=4, QED=2

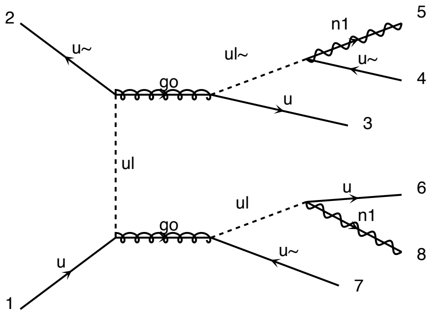


diagram 11 QCD=4, QED=2

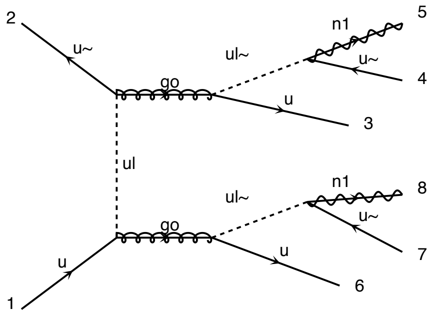
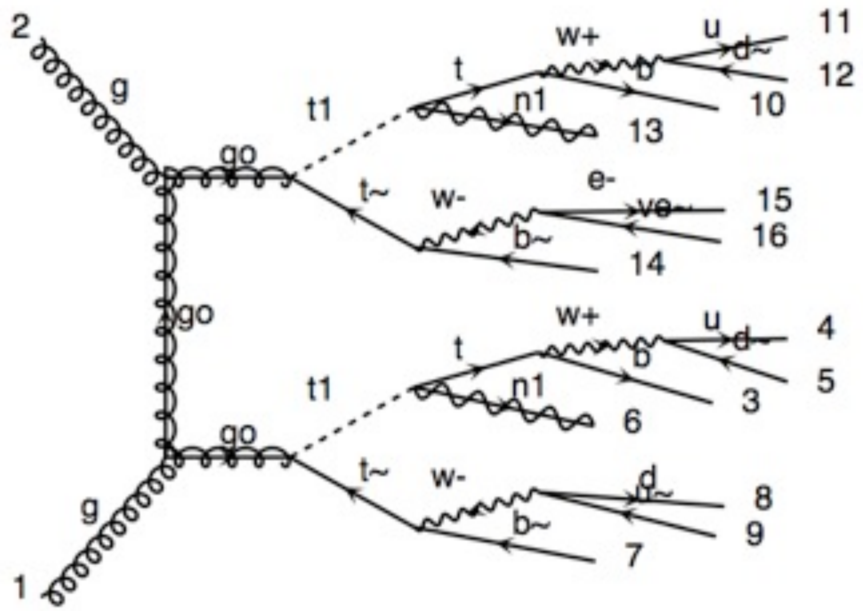


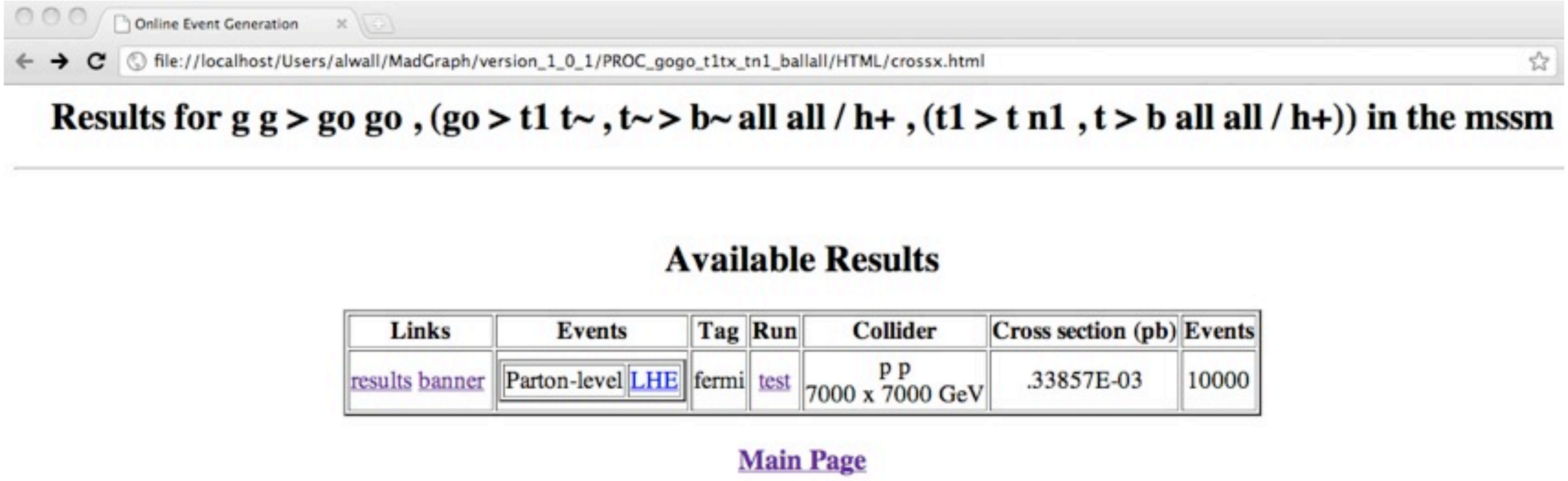
diagram 12 QCD=4, QED=2

Decay chains

- Decay chains retain **full matrix element** for the diagrams compatible with the decay
- Full spin correlations (within and between decays)
- Full width effects
- However, no interference with non-resonant diagrams
  - ➡ Description only valid close to pole mass
  - ➡ Cutoff at $|m \pm n\Gamma|$ where n is set in run_card.

Decay chains




Online Event Generation
file://localhost/Users/alwall/MadGraph/version_1_0_1/PROC_gogo_t1tx_tn1_ballall/HTML/crossx.html

Results for g g > go go , (go > t1 t~ , t~ > b~ all all / h+ , (t1 > t n1 , t > b all all / h+)) in the mssm

Available Results

Links	Events	Tag	Run	Collider	Cross section (pb)	Events
results banner	Parton-level LHE	fermi	test	p p 7000 x 7000 GeV	.33857E-03	10000

Main Page

Thanks to developments in MadEvent, also (very) long decay chains fast to simulate directly in MadGraph!

Event generation speed benchmarks

Generation of 10,000 unweighted events
Computer: Sony Vaio TZ laptop / *128-core cluster

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*

Event generation speed benchmarks

Generation of 10,000 unweighted events

Computer: Sony Vaio TZ laptop / *128-core cluster

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*

Event generation speed benchmarks

Generation of 10,000 unweighted events

Computer: Sony Vaio TZ laptop / *128-core cluster

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*

No problem running processes like $t\bar{t}$ + 0,1,2j on a laptop!

New MadEvent output

- New scheme for organizing MadEvent processes
  - ➡ Combine all processes with same type of initial/final states (color, spin, mass, width)
  - ➡ Combine all integration channels (diagrams) with same pole structures (and permutations)
  - ➡ Example: p p > l+ l- j j has subprocess directories:

	gg>llqq	-	gq>llgq	-	qq>llgg	-	qq>llqq	
with	8		16		10		14	int. channels

New MadEvent output

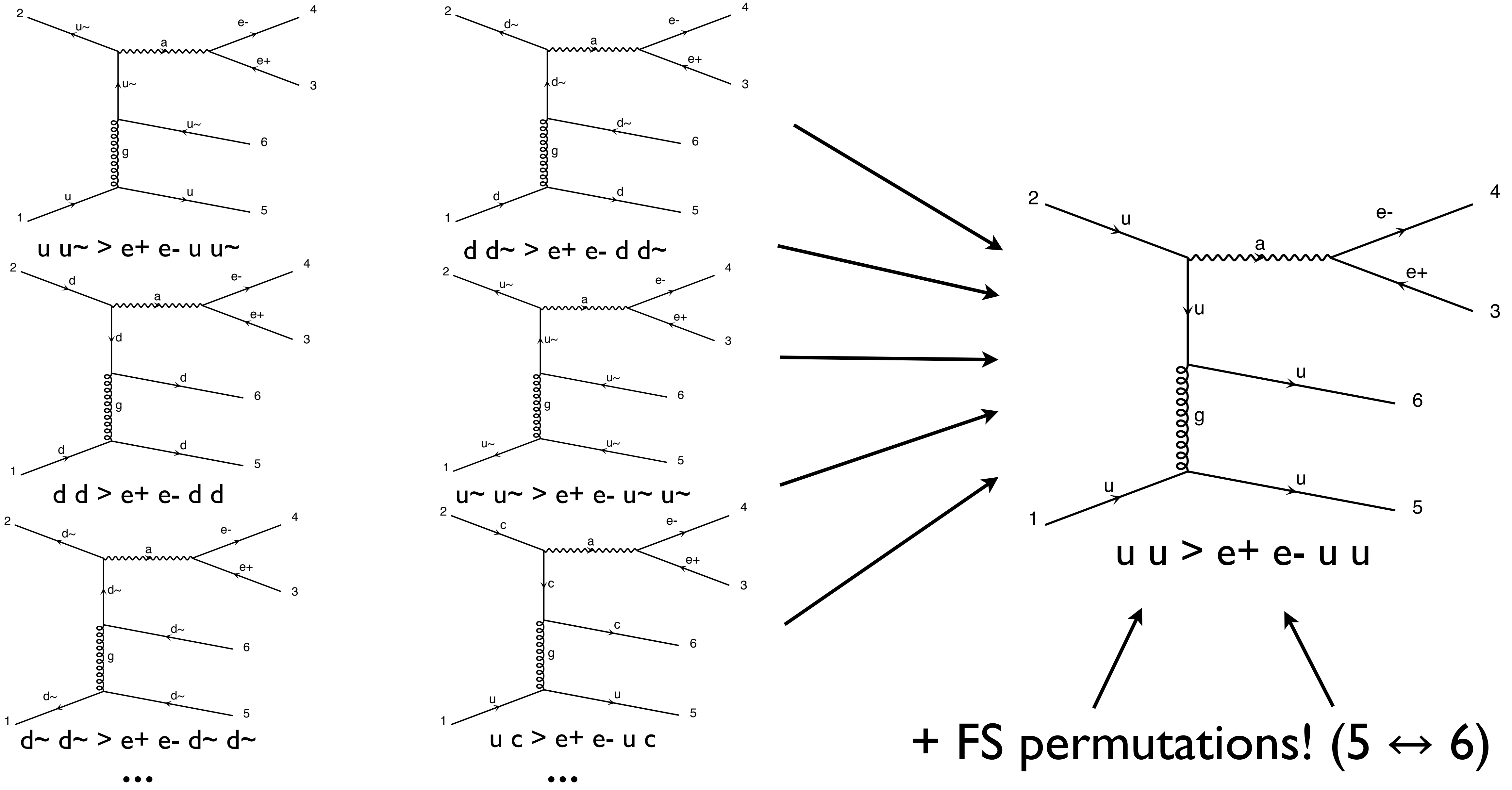
- New scheme for organizing MadEvent processes
  - ➡ Combine all processes with same type of initial/final states (color, spin, mass, width)
  - ➡ Combine all integration channels (diagrams) with same pole structures (and permutations)
  - ➡ Example: p p > l+ l- j j has subprocess directories:
    gg>llqq - gq>llgq - qq>llgg - qq>llqq

Corresponding to	2	4	2	20	matrix elements
with	8	16	10	14	int. channels

Compare with a total of 486 integration channels in MG4!

New MadEvent output

- Mapping/combination of similar integration channels:



u u~ > e+ e- u u~

d d~ > e+ e- d d~

d d > e+ e- d d

u~ u~ > e+ e- u~ u~

d~ d~ > e+ e- d~ d~

u c > e+ e- u c

...

...

u u > e+ e- u u

+ FS permutations! (5 ↔ 6)

News in MadGraph 5

- Super-fast decay chains
- Allow model vertices with any Lorentz structures, color structures, and any number of particles
- New color structures (color sextets and ε^{ijk})
- New, compact organization of subprocess directories and integration channels in MadEvent
- Output of matrix elements in Fortran (for MadEvent) or C++ (for Pythia 8)
- Automatic writing of models and helicity amplitudes for any Lorentz structures in Fortran or C by UFO-ALOHA
- Really user-friendly command line interface!

Code organization

madgraph

models

aloha

Template

bin

input

apidoc

vendor

tests

Code organization

madgraph

Python source code for MadGraph 5

models

UFO interface UFO and v4 models

aloha

Source code for output of ALOHA helicity amplitudes from UFO models

Template

Fortran source code and control files for event generation using MadEvent

bin

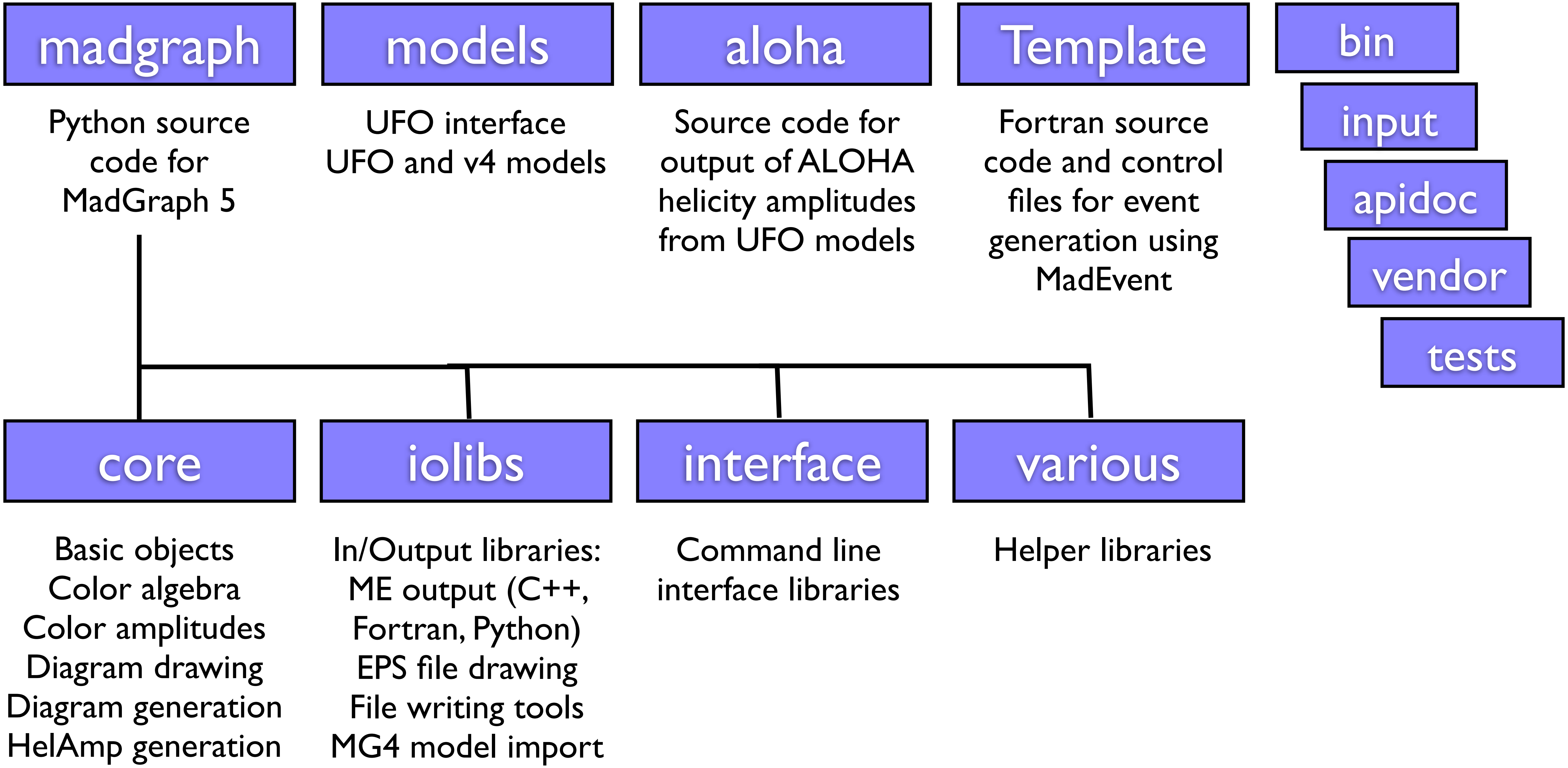
input

apidoc

vendor

tests

Code organization



madgraph

Python source code for MadGraph 5

models

UFO interface
UFO and v4 models

aloha

Source code for output of ALOHA helicity amplitudes from UFO models

Template

Fortran source code and control files for event generation using MadEvent

bin

input

apidoc

vendor

tests

core

Basic objects
Color algebra
Color amplitudes
Diagram drawing
Diagram generation
HelAmp generation

iolibs

In/Output libraries:
ME output (C++, Fortran, Python)
EPS file drawing
File writing tools
MG4 model import

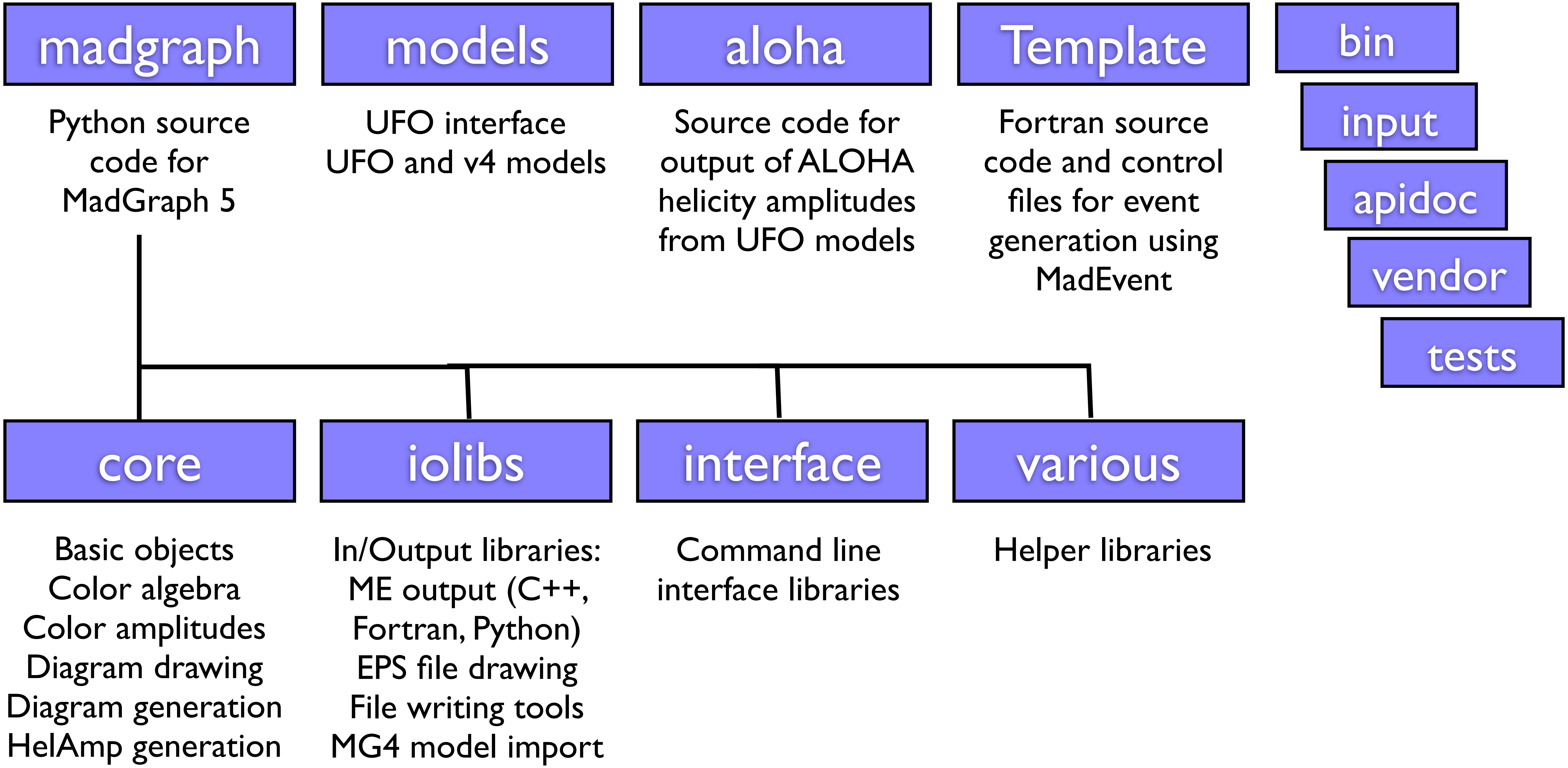
interface

Command line interface libraries

various

Helper libraries

Code organization



madgraph

Python source code for MadGraph 5

models

UFO interface UFO and v4 models

aloha

Source code for output of ALOHA helicity amplitudes from UFO models

Template

Fortran source code and control files for event generation using MadEvent

bin

input

apidoc

vendor

tests

core

Basic objects
Color algebra
Color amplitudes
Diagram drawing
Diagram generation
HelAmp generation

iolibs

In/Output libraries:
ME output (C++, Fortran, Python)
EPS file drawing
File writing tools
MG4 model import

interface

Command line interface libraries

various

Helper libraries

Pick and choose functionality/plug in new modules easy!

Code organization

Template

aloha

models

madgraph

bin: Run scripts

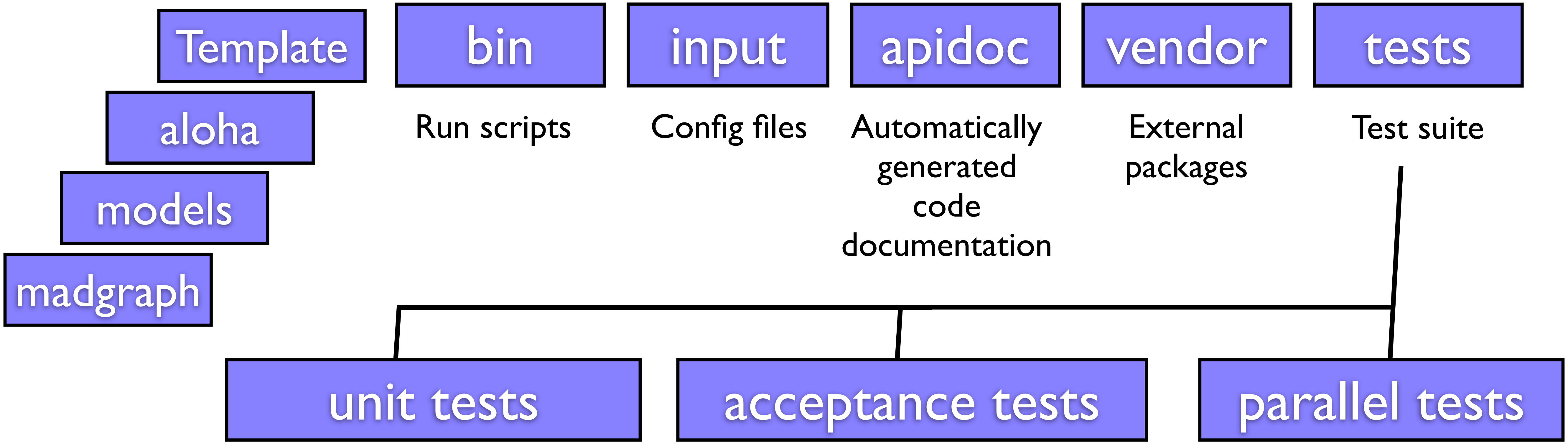
input: Config files

apidoc: Automatically generated code documentation

vendor: External packages

tests: Test suite

Code organization



Template

aloha

models

madgraph

bin

Run scripts

input

Config files

apidoc

Automatically generated code documentation

vendor

External packages

tests

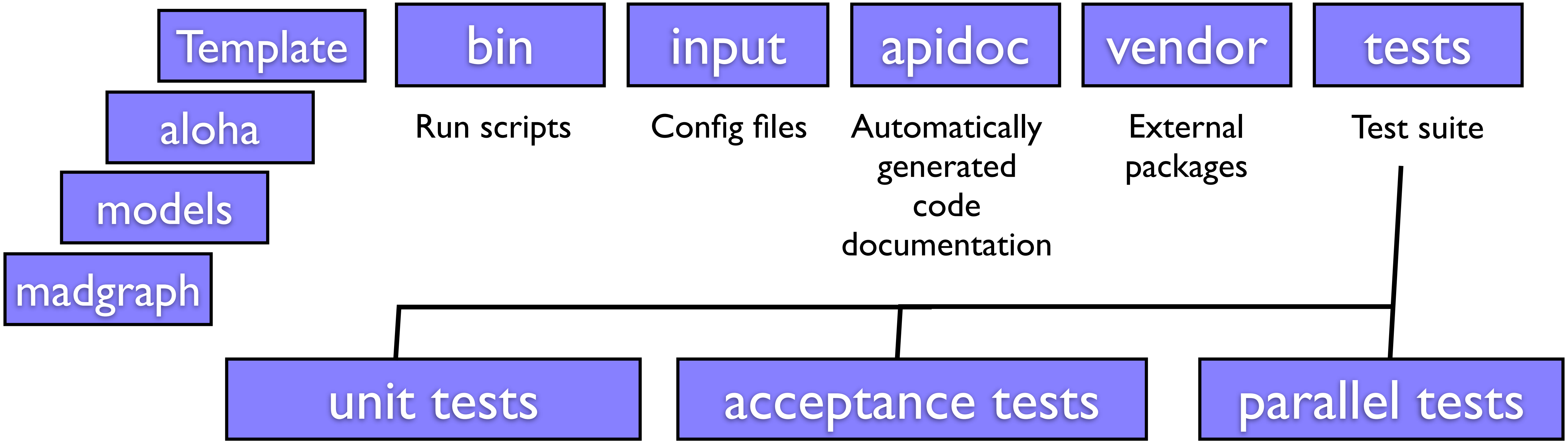
Test suite

unit tests

acceptance tests

parallel tests

Code organization



Template

aloha

models

madgraph

bin

Run scripts

input

Config files

apidoc

Automatically generated code documentation

vendor

External packages

tests

Test suite

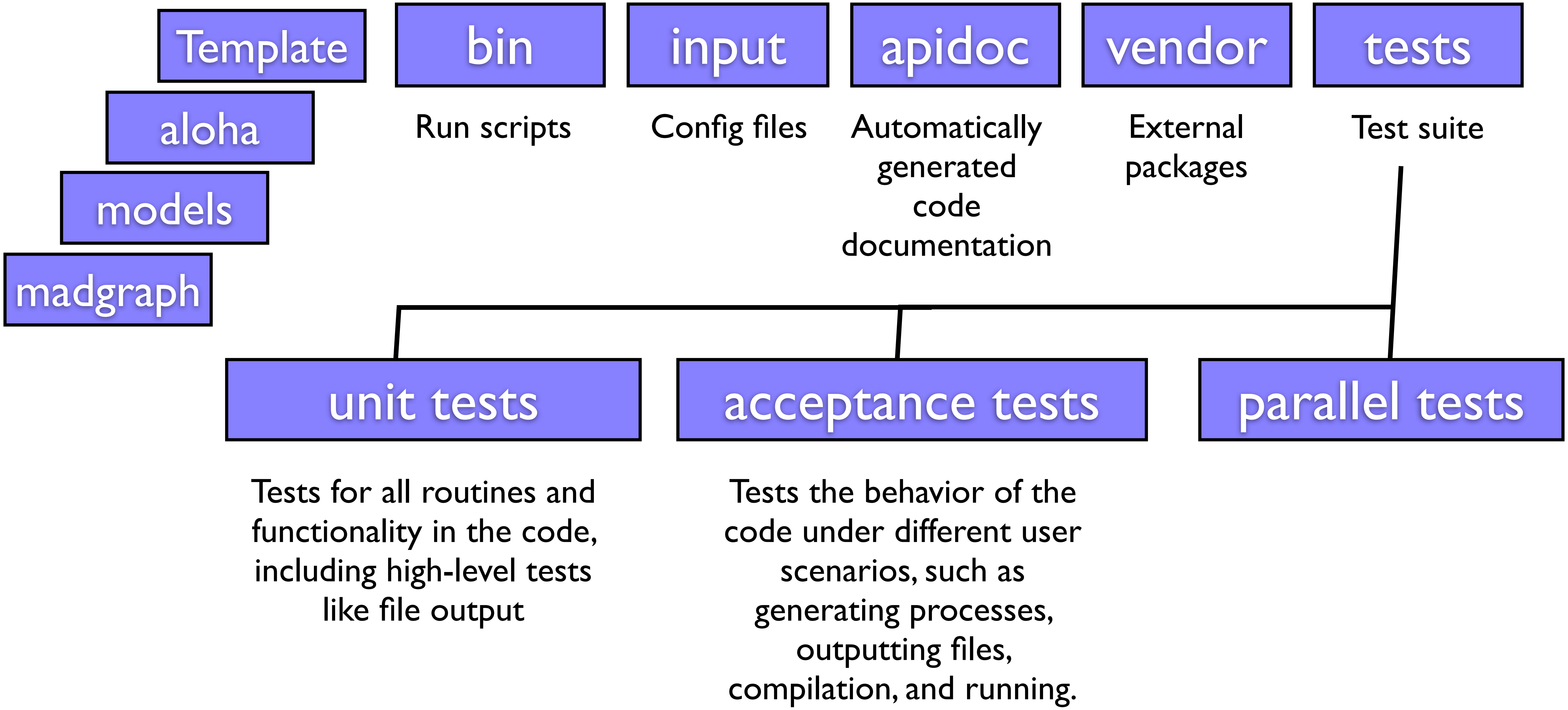
unit tests

acceptance tests

parallel tests

Tests for all routines and functionality in the code, including high-level tests like file output

Code organization



Template

aloha

models

madgraph

bin

Run scripts

input

Config files

apidoc

Automatically generated code documentation

vendor

External packages

tests

Test suite

unit tests

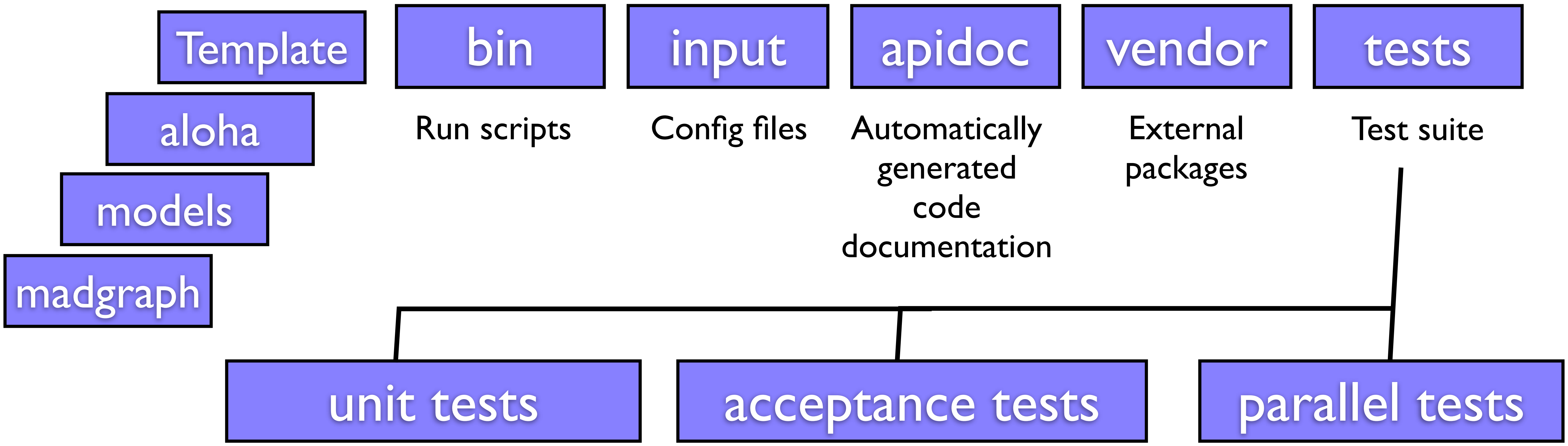
Tests for all routines and functionality in the code, including high-level tests like file output

acceptance tests

Tests the behavior of the code under different user scenarios, such as generating processes, outputting files, compilation, and running.

parallel tests

Code organization



Template

aloha

models

madgraph

bin

Run scripts

input

Config files

apidoc

Automatically generated code documentation

vendor

External packages

tests

Test suite

unit tests

Tests for all routines and functionality in the code, including high-level tests like file output

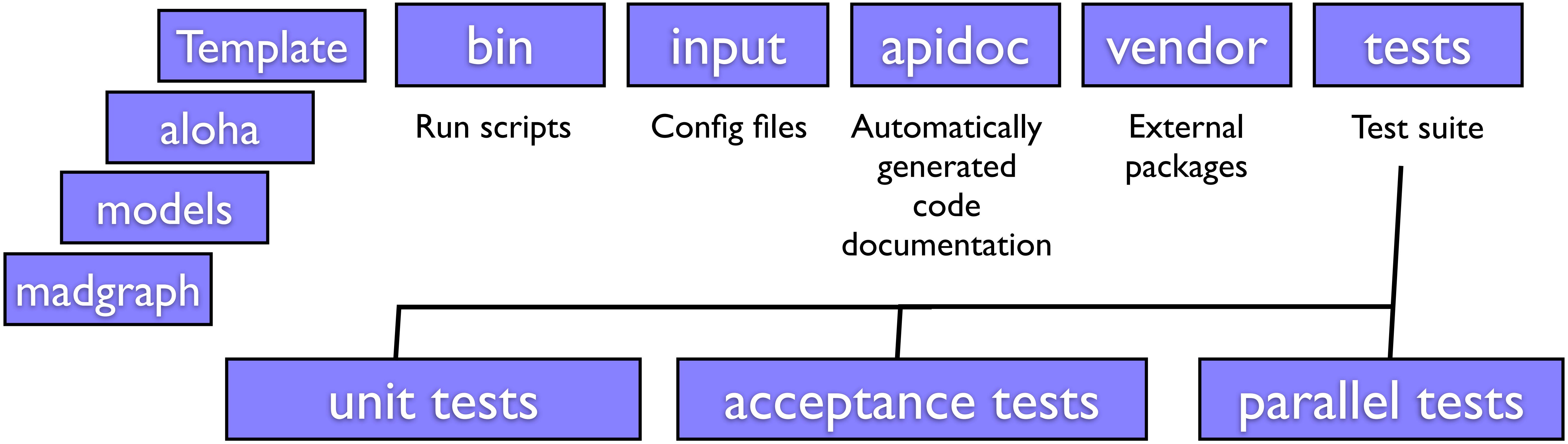
acceptance tests

Tests the behavior of the code under different user scenarios, such as generating processes, outputting files, compilation, and running.

parallel tests

Tests against another matrix element generator such as MadGraph 4, to ensure that the results are identical.
Over 6000 processes compared in SM, MSSM, HEFT and RS models.

Code organization



Template

aloha

models

madgraph

bin

Run scripts

input

Config files

apidoc

Automatically generated code documentation

vendor

External packages

tests

Test suite

unit tests

Tests for all routines and functionality in the code, including high-level tests like file output

acceptance tests

Tests the behavior of the code under different user scenarios, such as generating processes, outputting files, compilation, and running.

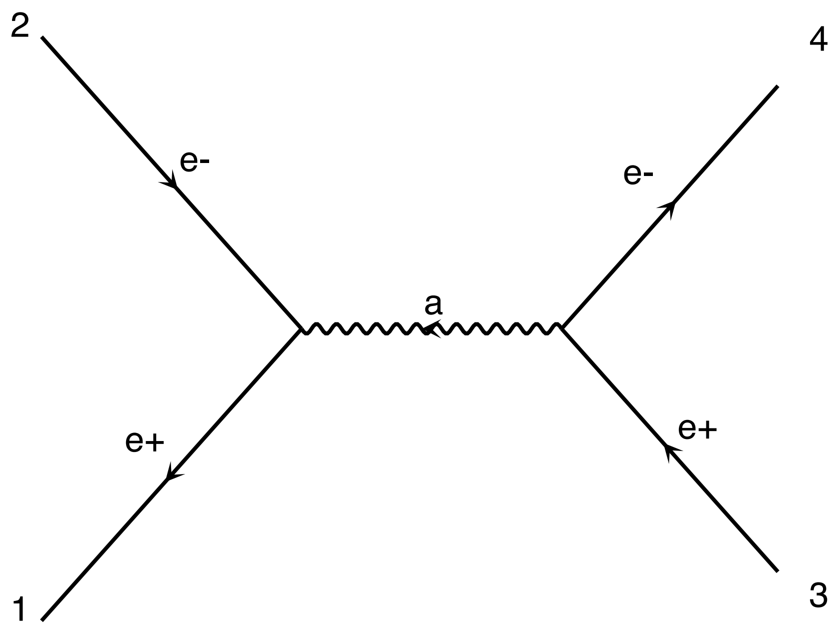
parallel tests

Tests against another matrix element generator such as MadGraph 4, to ensure that the results are identical.
Over 6000 processes compared in SM, MSSM, HEFT and RS models.

Test suite fundamental for stable development!

Basics: Helicity amplitudes

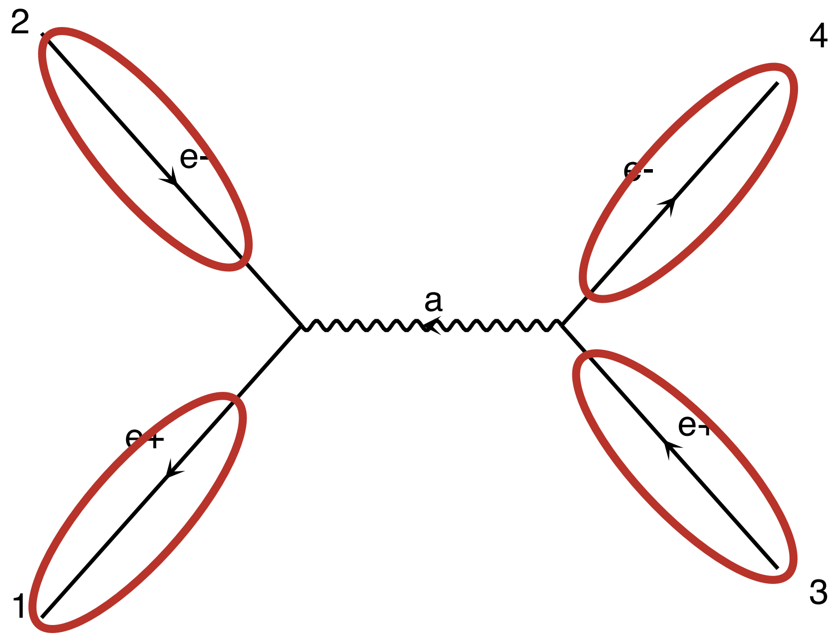
Idea: Evaluate $\mathscr{M}$ for fixed helicity of external particles



$$\mathcal{M} = \bar{u}\gamma^{\mu} v P_{\mu\nu} \bar{u}\gamma^{\nu} v$$

Basics: Helicity amplitudes

Idea: Evaluate $\mathscr{M}$ for fixed helicity of external particles

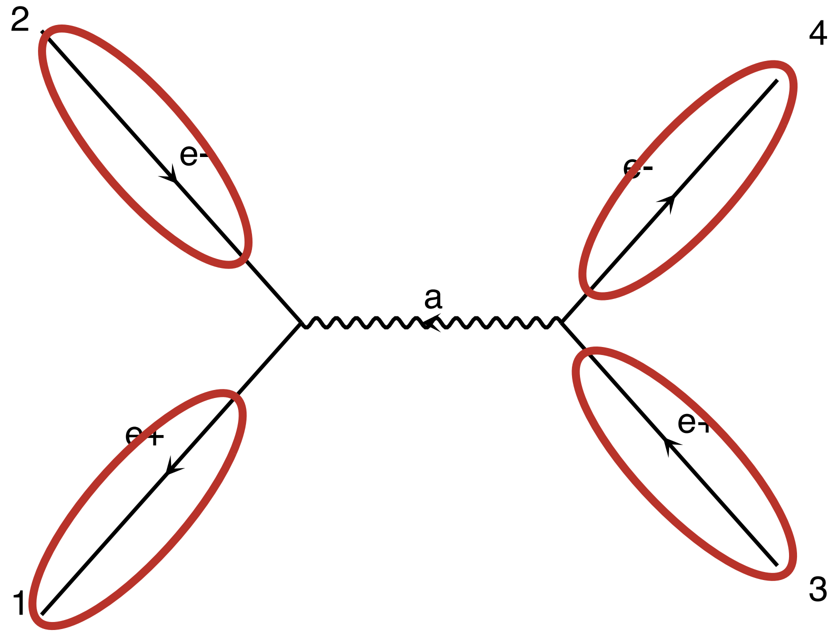


$$\mathcal{M} = \bar{u}\gamma^{\mu} v P_{\mu\nu} \bar{u}\gamma^{\nu} v$$

Numbers for given helicity and momenta

Basics: Helicity amplitudes

Idea: Evaluate $\mathscr{M}$ for fixed helicity of external particles



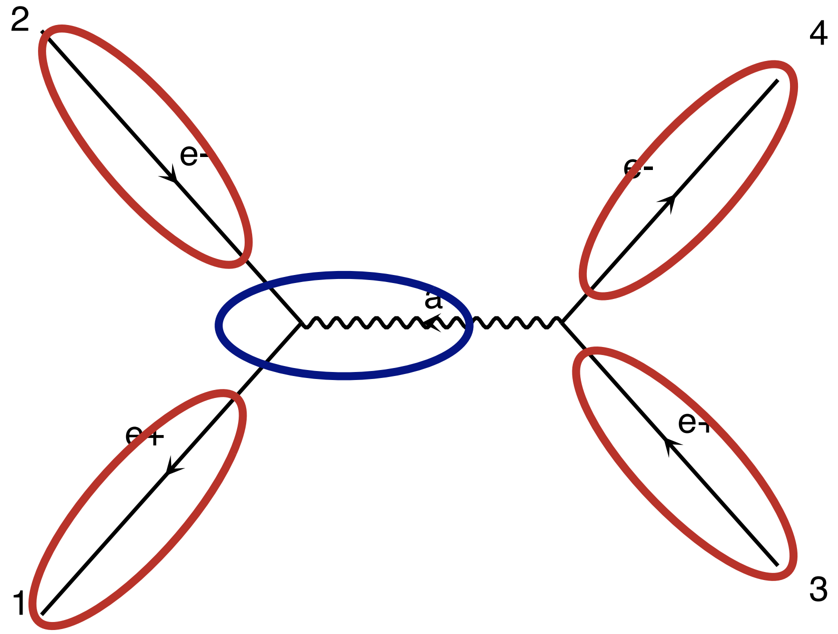
$$\mathcal{M} = \bar{u}\gamma^{\mu}v P_{\mu\nu}\bar{u}\gamma^{\nu}v$$

Numbers for given helicity and momenta

```
CALL OXXXXX(P(0,1),ZERO,NHEL(1),-1*IC(1),W(1,1))
CALL IXXXXX(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))
```

Basics: Helicity amplitudes

Idea: Evaluate $\mathscr{M}$ for fixed helicity of external particles



$$\mathcal{M} = \bar{u}\gamma^{\mu} v P_{\mu\nu} \bar{u}\gamma^{\nu} v$$

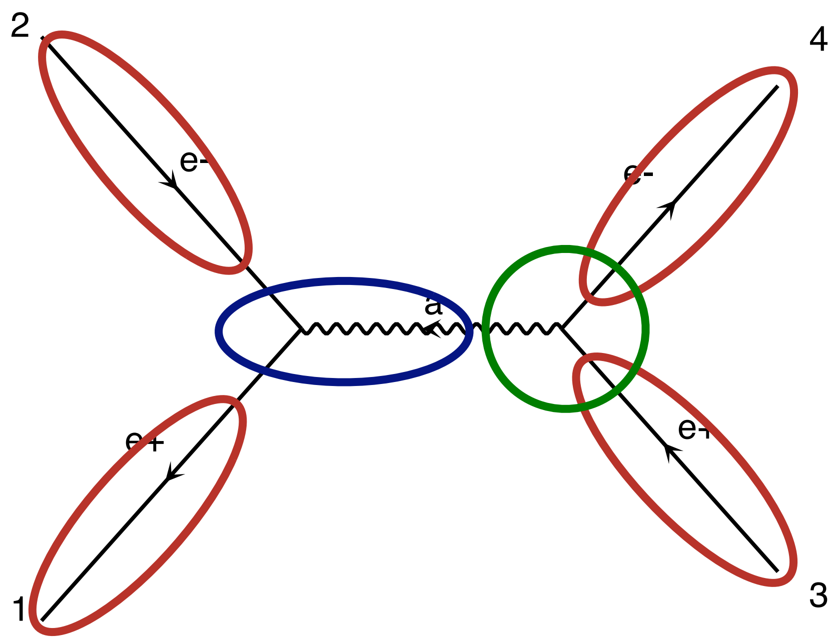
Numbers for given helicity and momenta

Calculate propagator wavefunctions

```
CALL OXXXXX(P(0,1),ZERO,NHEL(1),-1*IC(1),W(1,1))
CALL IXXXXX(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 JIOXXX(W(1,2),W(1,1),GAL,ZERO,ZERO,W(1,5))
```

Basics: Helicity amplitudes

Idea: Evaluate $\mathscr{M}$ for fixed helicity of external particles



$$\mathcal{M} = \bar{u}\gamma^{\mu} v P_{\mu\nu} \bar{u}\gamma^{\nu} v$$

Numbers for given helicity and momenta

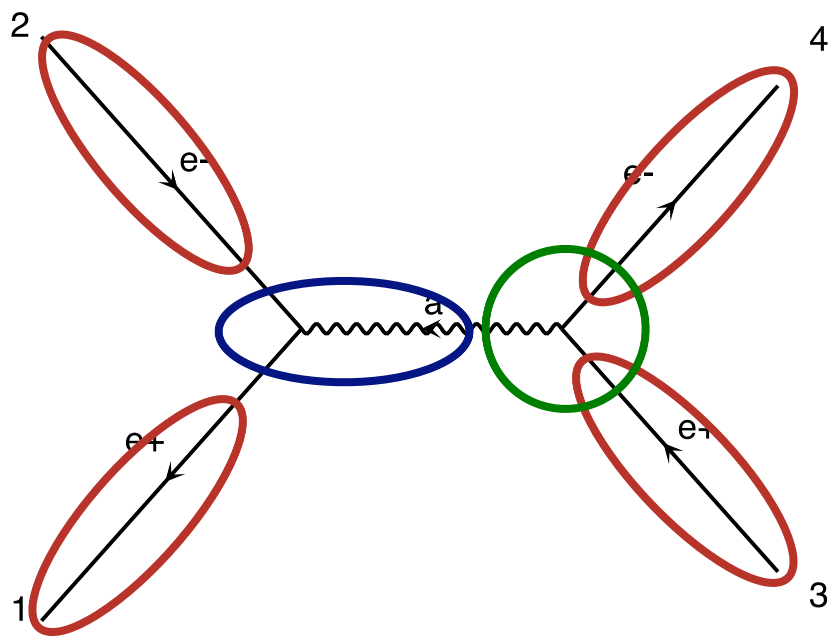
Calculate propagator wavefunctions

Finally evaluate amplitude (c-number)

```
CALL OXXXXX(P(0,1),ZERO,NHEL(1),-1*IC(1),W(1,1))
CALL IXXXXX(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 JIOXXX(W(1,2),W(1,1),GAL,ZERO,ZERO,W(1,5))
CALL IOVXXX(W(1,3),W(1,4),W(1,5),GAL,AMP(1))
```

Basics: Helicity amplitudes

Idea: Evaluate $\mathscr{M}$ for fixed helicity of external particles



$$\mathcal{M} = \bar{u}\gamma^{\mu} v P_{\mu\nu} \bar{u}\gamma^{\nu} v$$

Numbers for given helicity and momenta

Calculate propagator wavefunctions

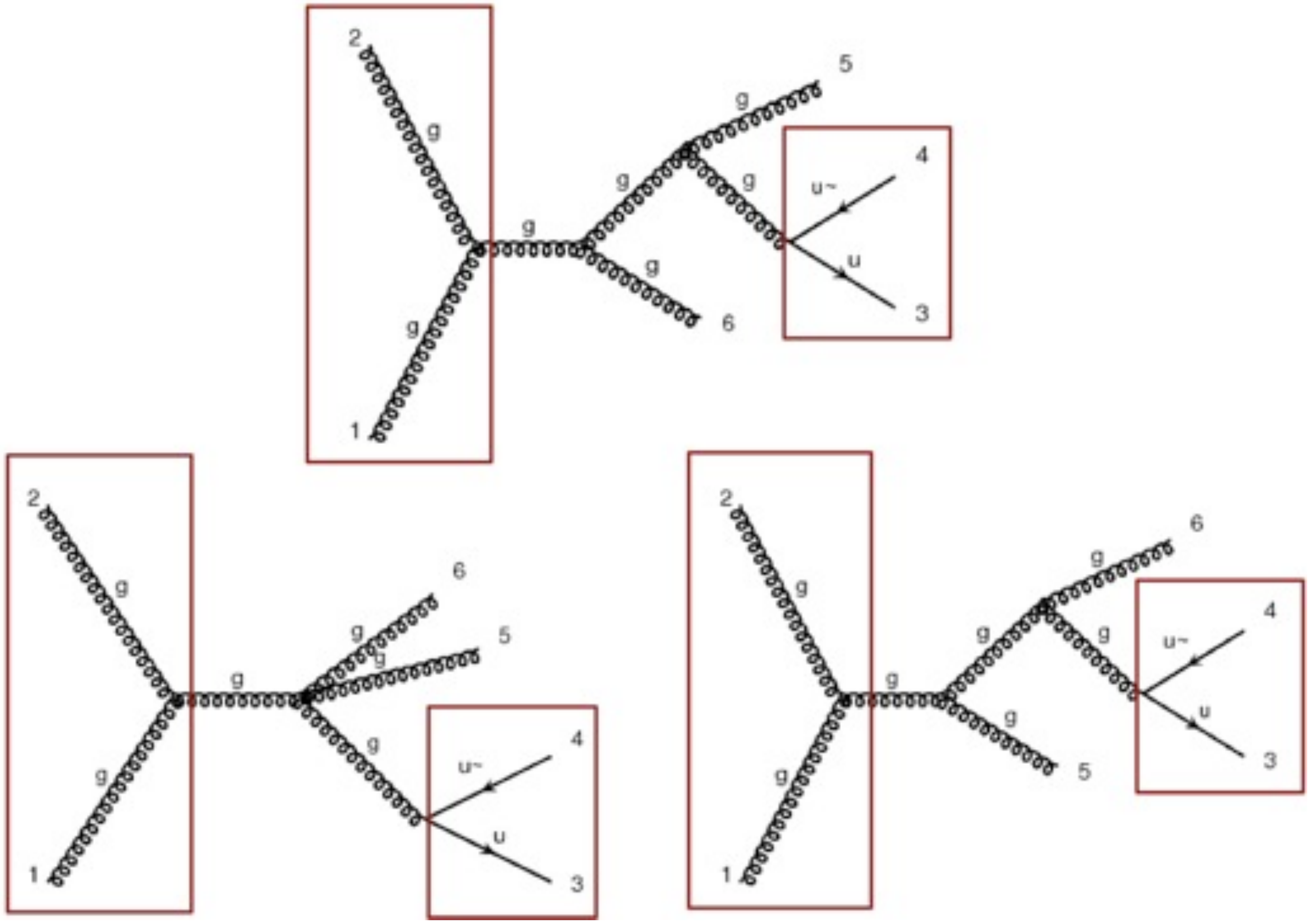
Finally evaluate amplitude (c-number)

Helicity amplitude calls written by MadGraph

```
CALL OXXXXX(P(0,1),ZERO,NHEL(1),-1*IC(1),W(1,1))
CALL IXXXXX(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 JIOXXX(W(1,2),W(1,1),GAL,ZERO,ZERO,W(1,5))
CALL IOVXXX(W(1,3),W(1,4),W(1,5),GAL,AMP(1))
```

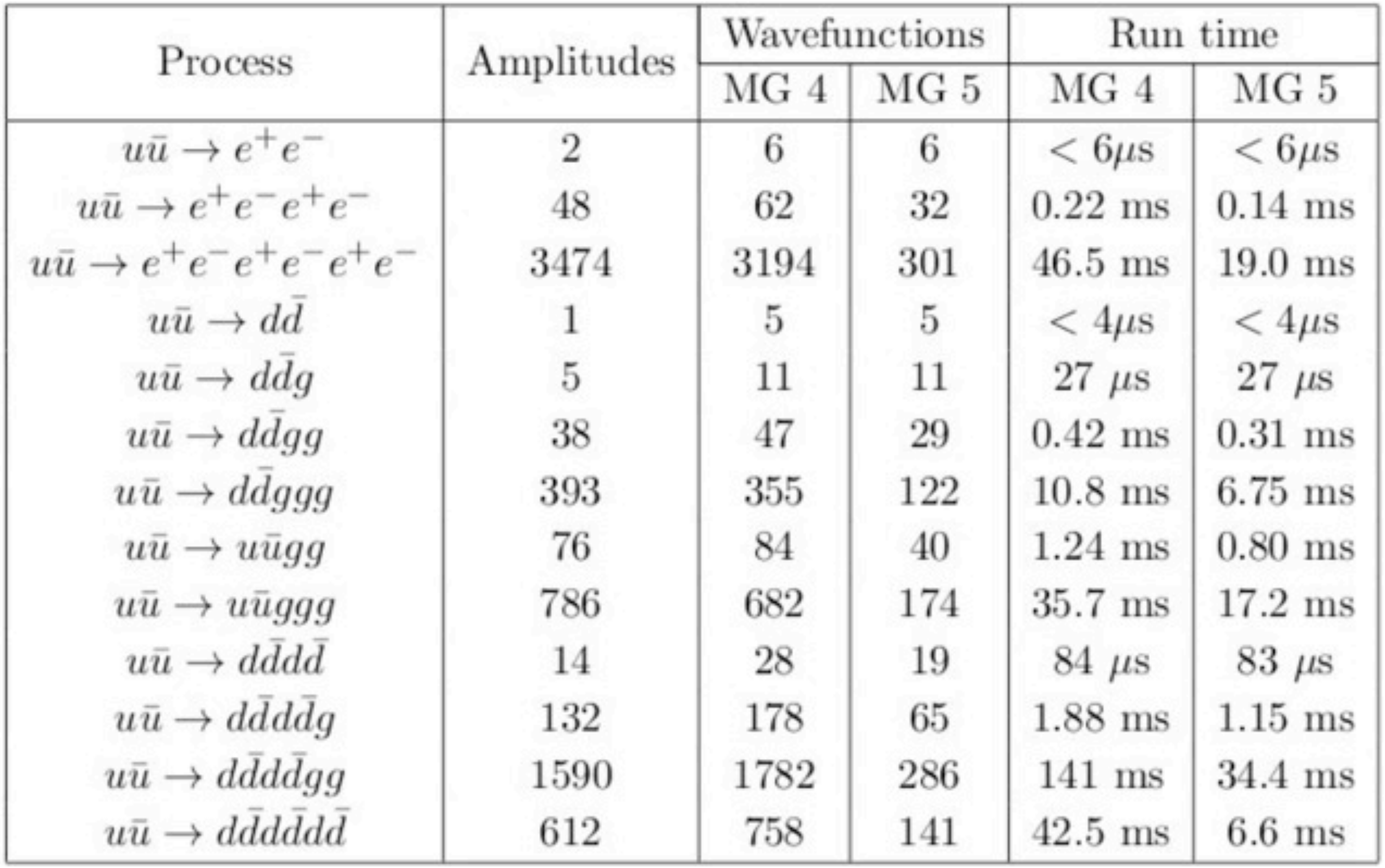
Basics: Helicity amplitudes

- Allows for fast calculation of any tree-level matrix element through efficient reuse of previously calculated wavefunctions across diagrams



Basics: Helicity amplitudes

- Thanks to new diagram generation algorithm, wf recycling much more efficient in MG5 than MG4

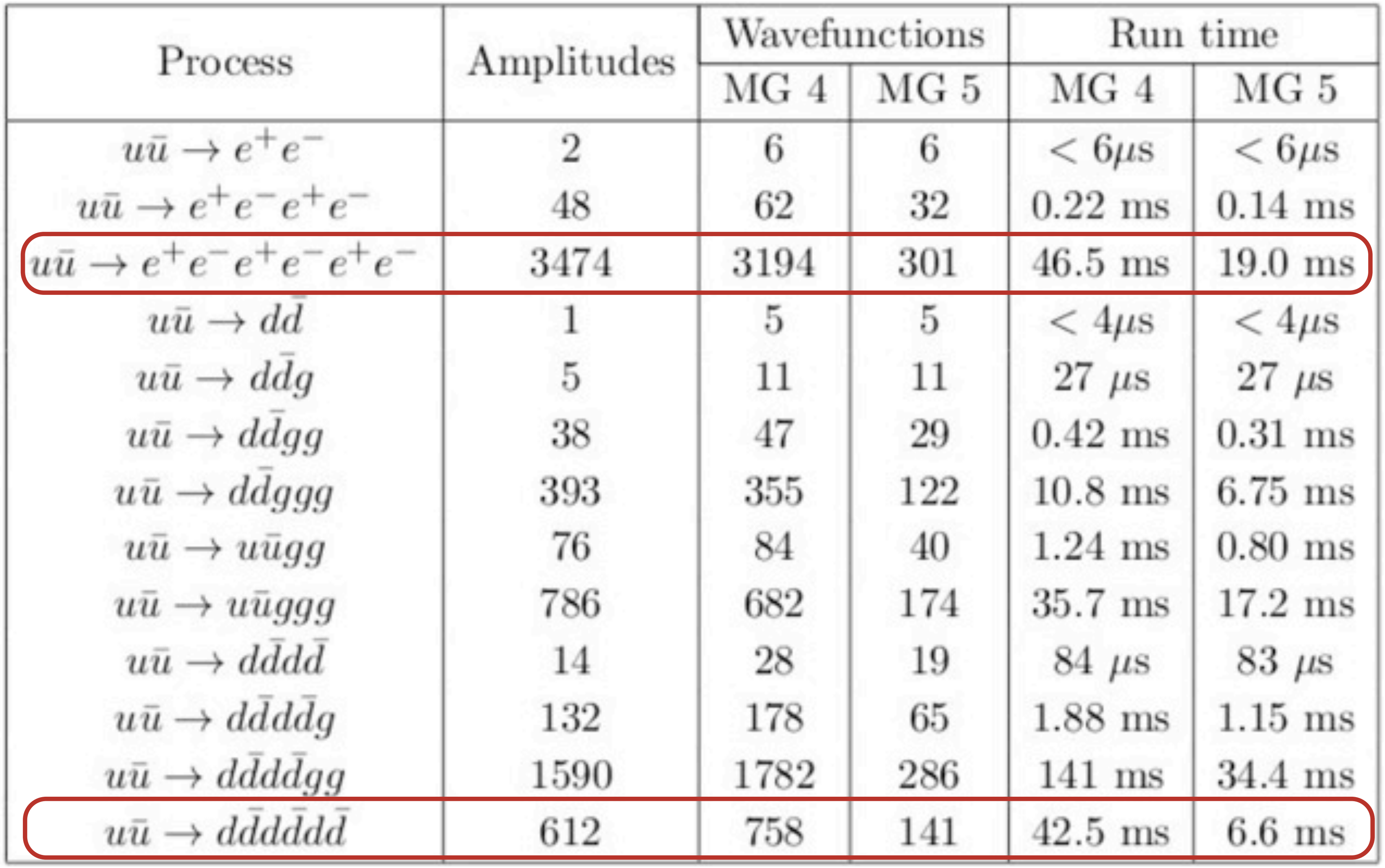


Process	Amplitudes	Wavefunctions		Run time	
		MG 4	MG 5	MG 4	MG 5
$u\bar{u} \to e^+e^-$	2	6	6	$< 6\mu$s	$< 6\mu$s
$u\bar{u} \to e^+e^-e^+e^-$	48	62	32	0.22 ms	0.14 ms
$u\bar{u} \to e^+e^-e^+e^-e^+e^-$	3474	3194	301	46.5 ms	19.0 ms
$u\bar{u} \to d\bar{d}$	1	5	5	$< 4\mu$s	$< 4\mu$s
$u\bar{u} \to d\bar{d}g$	5	11	11	27 μs	27 μs
$u\bar{u} \to d\bar{d}gg$	38	47	29	0.42 ms	0.31 ms
$u\bar{u} \to d\bar{d}ggg$	393	355	122	10.8 ms	6.75 ms
$u\bar{u} \to u\bar{u}gg$	76	84	40	1.24 ms	0.80 ms
$u\bar{u} \to u\bar{u}ggg$	786	682	174	35.7 ms	17.2 ms
$u\bar{u} \to d\bar{d}d\bar{d}$	14	28	19	84 μs	83 μs
$u\bar{u} \to d\bar{d}d\bar{d}g$	132	178	65	1.88 ms	1.15 ms
$u\bar{u} \to d\bar{d}d\bar{d}gg$	1590	1782	286	141 ms	34.4 ms
$u\bar{u} \to d\bar{d}d\bar{d}d\bar{d}$	612	758	141	42.5 ms	6.6 ms

Time for matrix element evaluation on a Sony Vaio TZ laptop

Basics: Helicity amplitudes

- Thanks to new diagram generation algorithm, wf recycling much more efficient in MG5 than MG4



Process	Amplitudes	Wavefunctions		Run time	
		MG 4	MG 5	MG 4	MG 5
$u\bar{u} \to e^+e^-$	2	6	6	$< 6\mu s$	$< 6\mu s$
$u\bar{u} \to e^+e^-e^+e^-$	48	62	32	0.22 ms	0.14 ms
$u\bar{u} \to e^+e^-e^+e^-e^+e^-$	3474	3194	301	46.5 ms	19.0 ms
$u\bar{u} \to d\bar{d}$	1	5	5	$< 4\mu s$	$< 4\mu s$
$u\bar{u} \to d\bar{d}g$	5	11	11	27 μs	27 μs
$u\bar{u} \to d\bar{d}gg$	38	47	29	0.42 ms	0.31 ms
$u\bar{u} \to d\bar{d}ggg$	393	355	122	10.8 ms	6.75 ms
$u\bar{u} \to u\bar{u}gg$	76	84	40	1.24 ms	0.80 ms
$u\bar{u} \to u\bar{u}ggg$	786	682	174	35.7 ms	17.2 ms
$u\bar{u} \to d\bar{d}d\bar{d}$	14	28	19	84 μs	83 μs
$u\bar{u} \to d\bar{d}d\bar{d}g$	132	178	65	1.88 ms	1.15 ms
$u\bar{u} \to d\bar{d}d\bar{d}gg$	1590	1782	286	141 ms	34.4 ms
$u\bar{u} \to d\bar{d}d\bar{d}d\bar{d}$	612	758	141	42.5 ms	6.6 ms

Time for matrix element evaluation on a Sony Vaio TZ laptop

HELAS

HELAS

- Original HELicity Amplitude Subroutine library

  Murayama, Watanabe, Hagiwara (1991)

HELAS

- Original HELicity Amplitude Subroutine library

  Murayama, Watanabe, Hagiwara (1991)

- All helicity amplitude routines needed for the Standard Model, MSSM and certain other applications in hand-written library

HELAS

- Original HELicity Amplitude Subroutine library

  Murayama, Watanabe, Hagiwara (1991)

- All helicity amplitude routines needed for the Standard Model, MSSM and certain other applications in hand-written library

- Any new Lorentz structures or other refinements need addition by hand

HELAS

- Original HELicity Amplitude Subroutine library

  Murayama, Watanabe, Hagiwara (1991)

- All helicity amplitude routines needed for the Standard Model, MSSM and certain other applications in hand-written library

- Any new Lorentz structures or other refinements need addition by hand

- Introduced a severe restriction on types of models that could be implemented in MadGraph

UFO-ALOHA

See lectures by Benjamin

UFO-ALOHA

See lectures by Benjamin

- Solution: Completely automated writing of Helicity Amplitude subroutines from model information

UFO-ALOHA

See lectures by Benjamin

- Solution: Completely automated writing of Helicity Amplitude subroutines from model information

- Necessary ingredients:
  - ➡ Complete specification of all Lorentz structures in model (by FeynRules in the UFO format)
  - ➡ Fast and flexible calculation of all helicity amplitudes from Lorentz structures
  - ➡ Separate output routines to allow output in multiple languages

UFO-ALOHA

See lectures by Benjamin

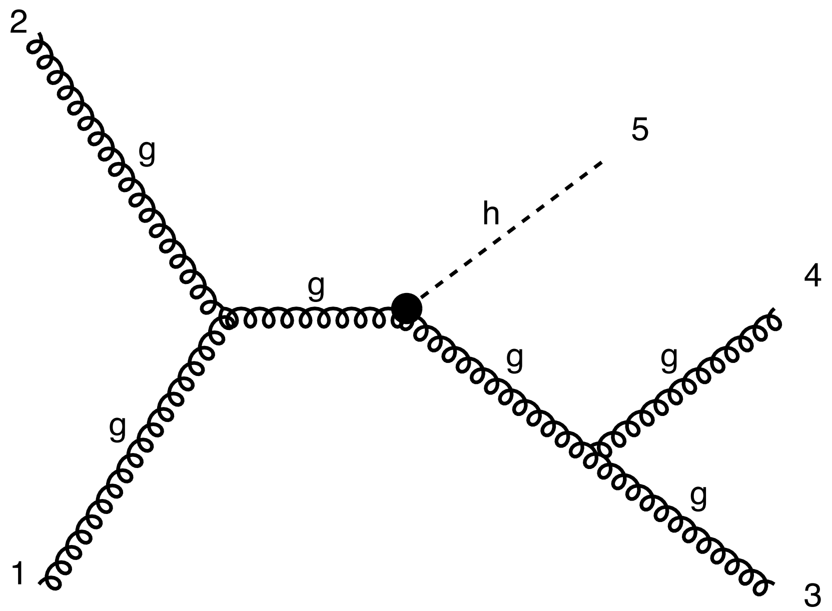
- Solution: Completely automated writing of Helicity Amplitude subroutines from model information
- Necessary ingredients:
  - ➡ Complete specification of all Lorentz structures in model (by FeynRules in the UFO format)
  - ➡ Fast and flexible calculation of all helicity amplitudes from Lorentz structures
  - ➡ Separate output routines to allow output in multiple languages
- Allows calculation of matrix elements for ANY Lorentz structure with ANY particle multiplicity

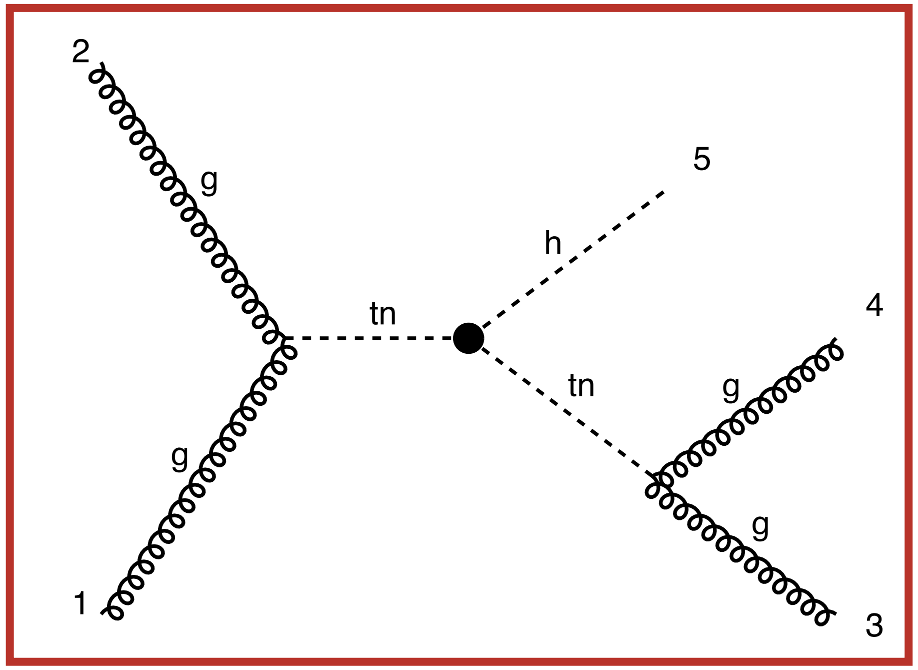
Examples 1: Multi-particle vertices

- Higgs effective couplings to gluons (through top loop)
  - ➡ From non-commutativity of QCD, get H + 2-, 3- and 4-gluon vertices
  - ➡ In MG4 model: 5-particle vertex handled by using non-propagating tensor particles

Examples 1: Multi-particle vertices

- Higgs effective couplings to gluons (through top loop)
  - ➡ From non-commutativity of QCD, get H + 2-, 3- and 4-gluon vertices
  - ➡ In MG4 model: 5-particle vertex handled by using non-propagating tensor particles





Examples 1: Multi-particle vertices

- Higgs effective couplings to gluons (through top loop)
  - ➡ From non-commutativity of QCD, get H + 2-, 3- and 4-gluon vertices
  - ➡ In MG4 model: 5-particle vertex handled by using non-propagating tensor particles

```
      CALL VXXXXX(P(0,1),ZERO,NHEL(1),-1*IC(1),W(1,1))
      CALL VXXXXX(P(0,2),ZERO,NHEL(2),-1*IC(2),W(1,2))
      CALL VXXXXX(P(0,3),ZERO,NHEL(3),+1*IC(3),W(1,3))
      CALL VXXXXX(P(0,4),ZERO,NHEL(4),+1*IC(4),W(1,4))
      CALL SXXXXX(P(0,5),+1*IC(5),W(1,5))
      CALL JVVXXX(W(1,1),W(1,2),G,ZERO,ZERO,W(1,6))
      CALL JVVXXX(W(1,3),W(1,4),G,ZERO,ZERO,W(1,7))
C     Amplitude(s) for diagram number 1
      CALL VVSHXX(W(1,6),W(1,7),W(1,5),GH,AMP(1))
      CALL UVVAXX(W(1,1),W(1,2),G,ZERO,ZERO,ZERO,W(1,8))
      CALL UVVAXX(W(1,3),W(1,4),G,ZERO,ZERO,ZERO,W(1,9))
C     Amplitude(s) for diagram number 2
      CALL TTSAXX(W(1,8),W(1,9),W(1,5),GH,AMP(2))
```

Examples 1: Multi-particle vertices

- Higgs effective couplings to gluons (through top loop)
  - ➡ From non-commutativity of QCD, get H + 2-, 3- and 4-gluon vertices
  - ➡ In MG4 model: 5-particle vertex handled by using non-propagating tensor particles

```
      CALL VXXXXX(P(0,1),ZERO,NHEL(1),-1*IC(1),W(1,1))
      CALL VXXXXX(P(0,2),ZERO,NHEL(2),-1*IC(2),W(1,2))
      CALL VXXXXX(P(0,3),ZERO,NHEL(3),+1*IC(3),W(1,3))
      CALL VXXXXX(P(0,4),ZERO,NHEL(4),+1*IC(4),W(1,4))
      CALL SXXXXX(P(0,5),+1*IC(5),W(1,5))
      CALL JVVXXX(W(1,1),W(1,2),G,ZERO,ZERO,W(1,6))
      CALL JVVXXX(W(1,3),W(1,4),G,ZERO,ZERO,W(1,7))
C     Amplitude(s) for diagram number 1
      CALL VVSHXX(W(1,6),W(1,7),W(1,5),GH,AMP(1))
      CALL UVVAXX(W(1,1),W(1,2),G,ZERO,ZERO,ZERO,W(1,8))
      CALL UVVAXX(W(1,3),W(1,4),G,ZERO,ZERO,ZERO,W(1,9))
C     Amplitude(s) for diagram number 2
      CALL TTSAXX(W(1,8),W(1,9),W(1,5),GH,AMP(2))
```

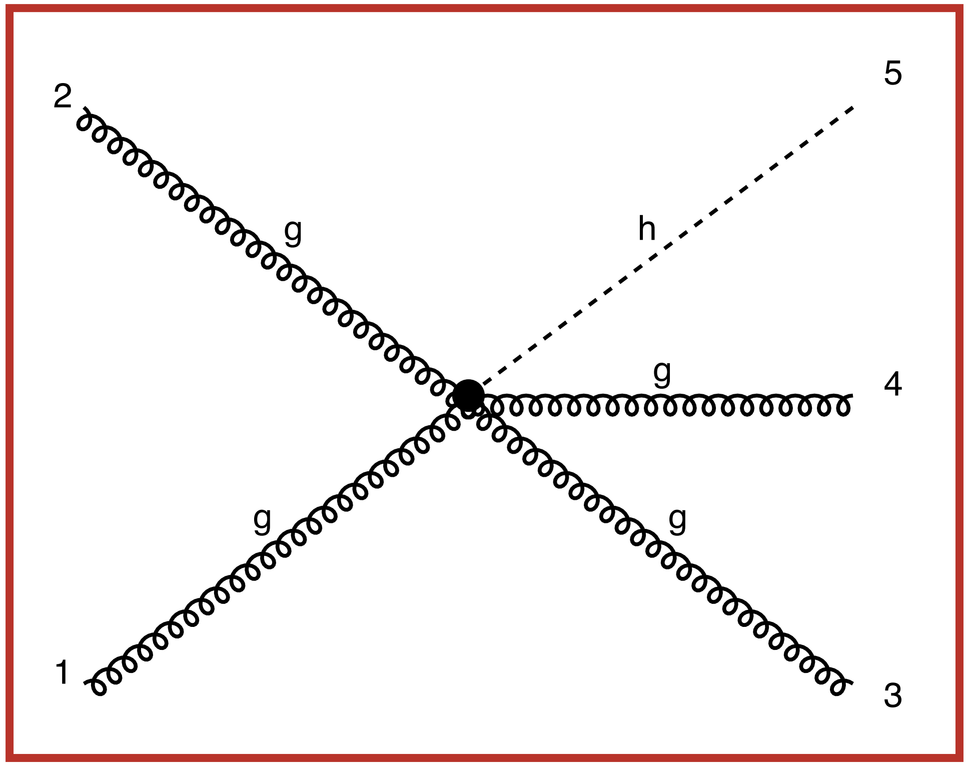
Tensor calls

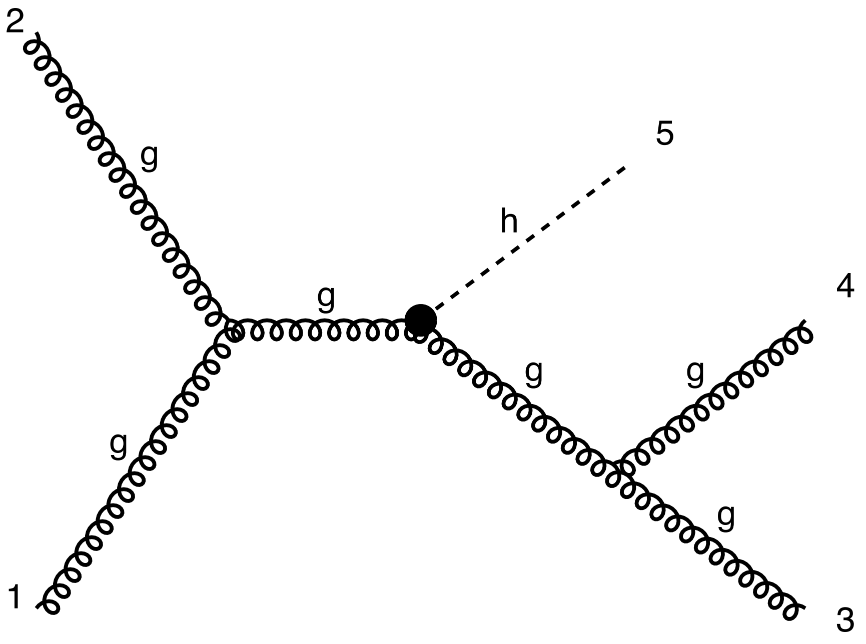
Examples 1: Multi-particle vertices

- In MG5 + UFO/ALOHA:
  Use 5-particle vertex directly!

Examples 1: Multi-particle vertices

- In MG5 + UFO/ALOHA:
  Use 5-particle vertex directly!





Examples 1: Multi-particle vertices

- In MG5 + UFO/ALOHA:
  Use 5-particle vertex directly!

```
      CALL VXXXXX(P(0,1),ZERO,NHEL(1),-1*IC(1),W(1,1))
      CALL VXXXXX(P(0,2),ZERO,NHEL(2),-1*IC(2),W(1,2))
      CALL VXXXXX(P(0,3),ZERO,NHEL(3),+1*IC(3),W(1,3))
      CALL VXXXXX(P(0,4),ZERO,NHEL(4),+1*IC(4),W(1,4))
      CALL SXXXXX(P(0,5),+1*IC(5),W(1,5))
C     Amplitude(s) for diagram number 1
      CALL VVVVS1_0(W(1,1),W(1,2),W(1,3),W(1,4),W(1,5),GC_10,AMP(1))
      CALL VVVVS2_0(W(1,1),W(1,2),W(1,3),W(1,4),W(1,5),GC_10,AMP(2))
      CALL VVVVS3_0(W(1,1),W(1,2),W(1,3),W(1,4),W(1,5),GC_10,AMP(3))
      CALL VVV1_1(W(1,1),W(1,2),GC_5,ZERO, ZERO, W(1,6))
      CALL VVV1_1(W(1,3),W(1,4),GC_5,ZERO, ZERO, W(1,7))
C     Amplitude(s) for diagram number 2
      CALL VVS3_0(W(1,6),W(1,7),W(1,5),GC_8,AMP(4))
```

Examples 1: Multi-particle vertices

- In MG5 + UFO/ALOHA:
  Use 5-particle vertex directly!

```
      CALL VXXXXX(P(0,1),ZERO,NHEL(1),-1*IC(1),W(1,1))
      CALL VXXXXX(P(0,2),ZERO,NHEL(2),-1*IC(2),W(1,2))
      CALL VXXXXX(P(0,3),ZERO,NHEL(3),+1*IC(3),W(1,3))
      CALL VXXXXX(P(0,4),ZERO,NHEL(4),+1*IC(4),W(1,4))
      CALL SXXXXX(P(0,5),+1*IC(5),W(1,5))
C     Amplitude(s) for diagram number 1
      CALL VVVVS1_0(W(1,1),W(1,2),W(1,3),W(1,4),W(1,5),GC_10,AMP(1))
      CALL VVVVS2_0(W(1,1),W(1,2),W(1,3),W(1,4),W(1,5),GC_10,AMP(2))
      CALL VVVVS3_0(W(1,1),W(1,2),W(1,3),W(1,4),W(1,5),GC_10,AMP(3))
      CALL VVV1_1(W(1,1),W(1,2),GC_5,ZERO, ZERO, W(1,6))
      CALL VVV1_1(W(1,3),W(1,4),GC_5,ZERO, ZERO, W(1,7))
C     Amplitude(s) for diagram number 2
      CALL VVS3_0(W(1,6),W(1,7),W(1,5),GC_8,AMP(4))
```

5-particle calls

Example 2: Multi-fermion vertices

- For multifermion vertices there is the additional difficulty of multiple fermion flows

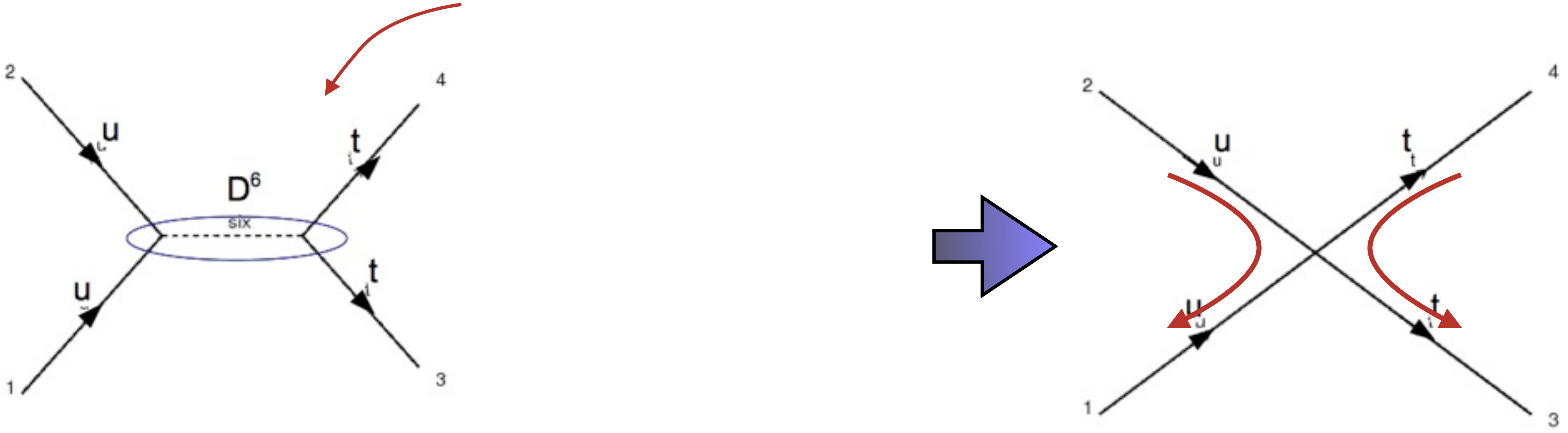
  ➡ Example: $\bar{u}\ \bar{u}\ t\ t \neq \bar{u}\ t\ \bar{u}\ t$ (from MG point of view*)

* Can interchange using Fierz identities by modified Lorentz structures

Example 2: Multi-fermion vertices

- For multifermion vertices there is the additional difficulty of multiple fermion flows

➡ Example: $\bar{u}\ \bar{u}\ t\ t \quad \neq \quad \bar{u}\ t\ \bar{u}\ t$ (from MG point of view*)

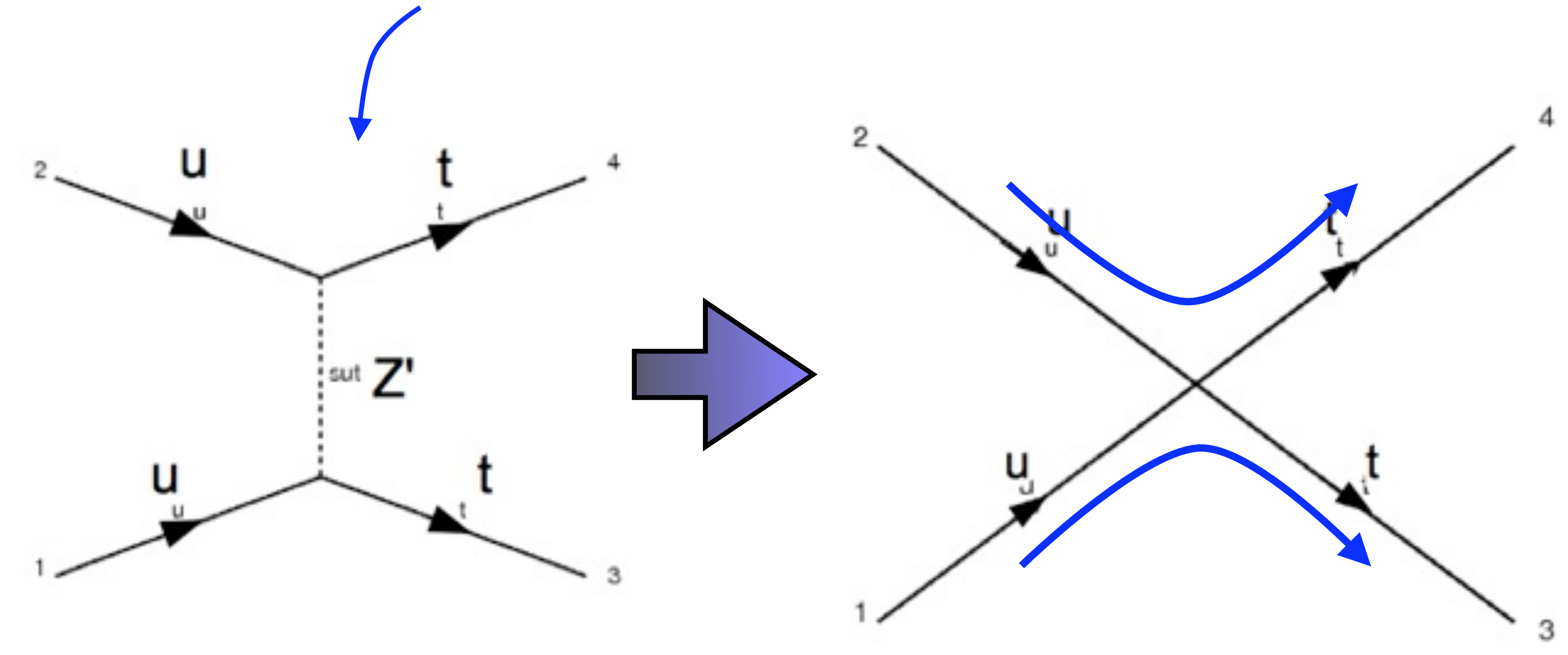


* Can interchange using Fierz identities by modified Lorentz structures

Example 2: Multi-fermion vertices

- For multifermion vertices there is the additional difficulty of multiple fermion flows

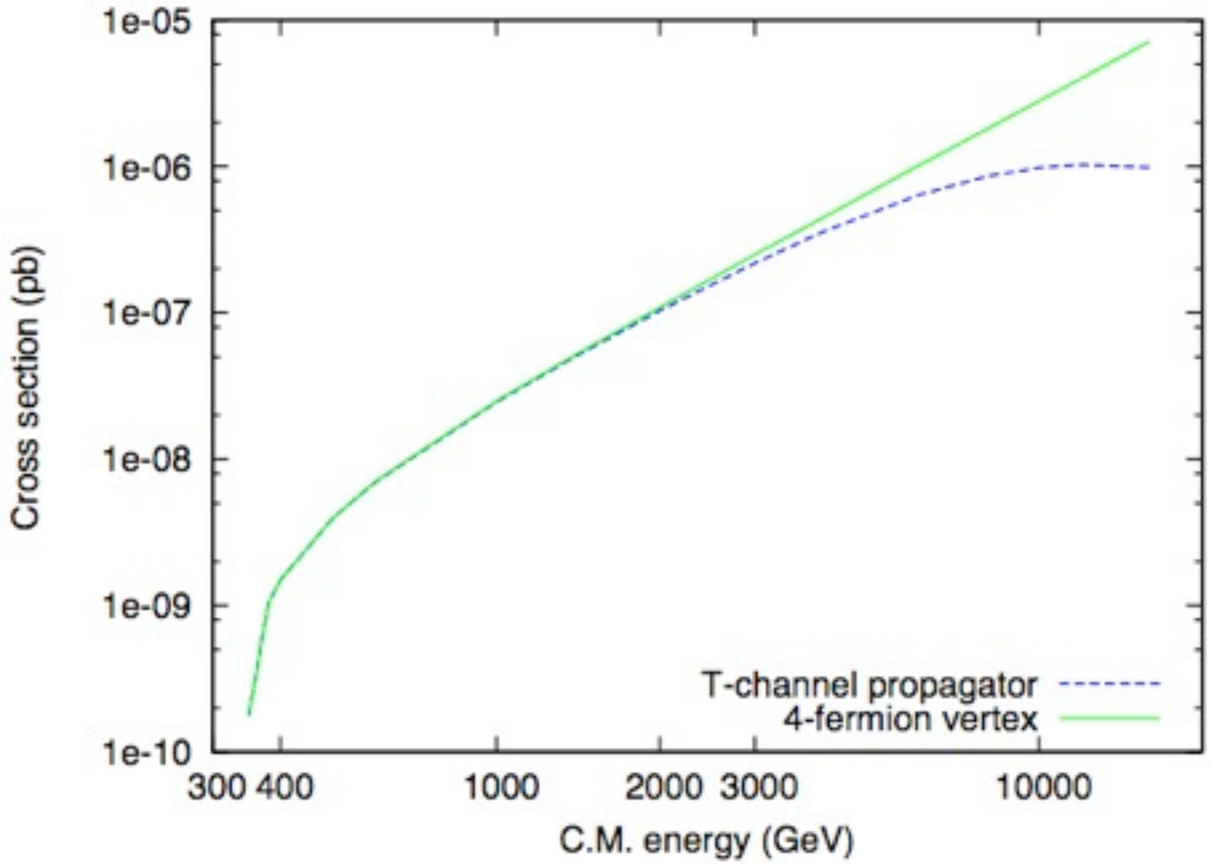
➡ Example: $\bar{u}\;\bar{u}\;t\;t \;\neq\; \bar{u}\;t\;\bar{u}\;t$ (from MG point of view*)



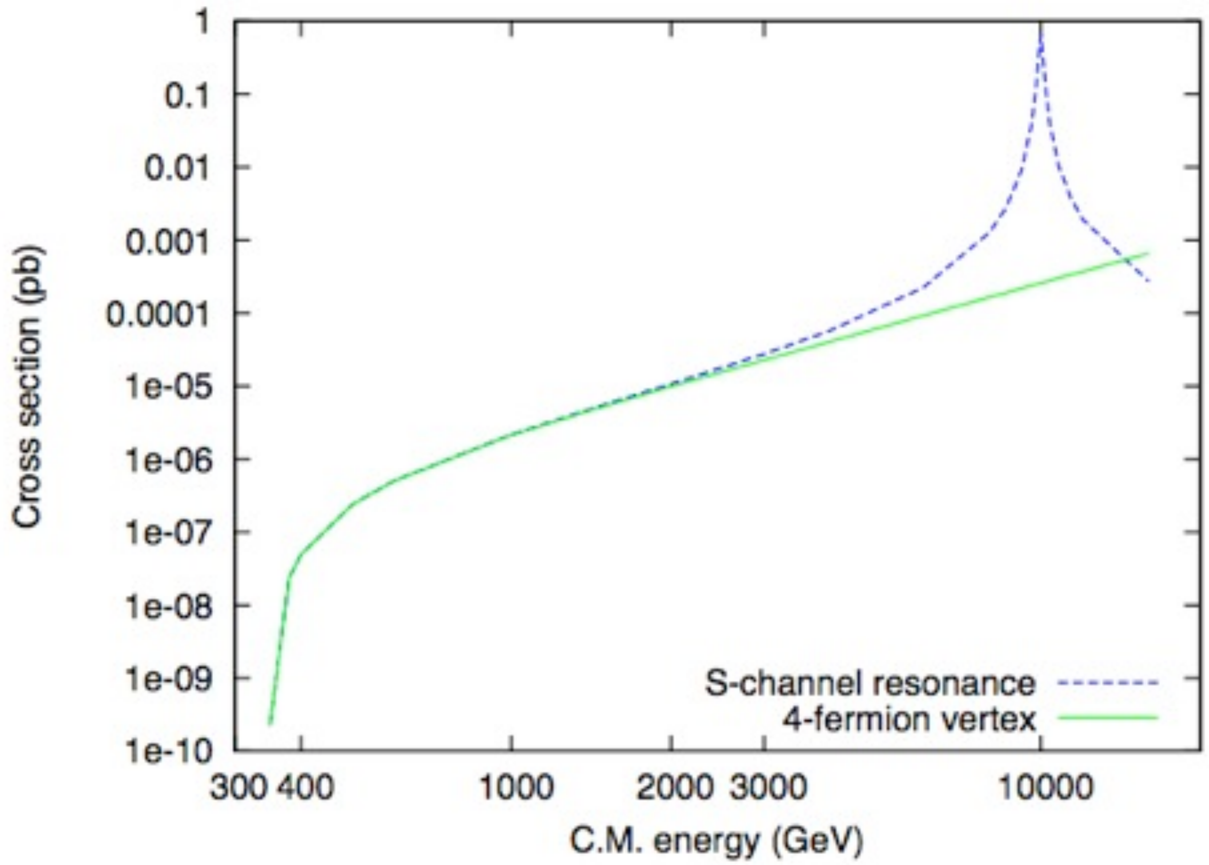
* Can interchange using Fierz identities by modified Lorentz structures

Example 2: Multi-fermion vertices

Comparisons between explicit propagators and 4-fermion vertex



t-channel u u > t t



s-channel u u > t t

Output formats in MadGraph 5

Output formats in MadGraph 5

- Thanks to UFO/ALOHA, we now have automatic helicity amplitude routines in any language

➡ So it makes sense to have also matrix element output in multiple languages!

Output formats in MadGraph 5

- Thanks to UFO/ALOHA, we now have automatic helicity amplitude routines in any language
  - ➡ So it makes sense to have also matrix element output in multiple languages!
- Presently implemented: Fortran, C++, Python
  - ➡ Fortran - for MadEvent and Standalone
  - ➡ C++ - for Pythia 8 and Standalone
  - ➡ Python - for internal use in MG5 (checks of gauge, perturbation and Lorentz invariance)

Pythia 8 Matrix Element output

- ➡ Library of process .h and .cc files, sorted by model
  + all needed model and
    helicity amplitude files
  + example main file
    (for user convenience!)
- ➡ Run as standard internal Pythia processes
- ➡ Allows using Pythia for ANY (2→1,2,3) process in ANY model at the push of a key!

Pythia 8 Matrix Element output

- ➡ Library of process .h and .cc files, sorted by model
  + all needed model and
    helicity amplitude files
  + example main file
    (for user convenience!)

- ➡ Run as standard internal Pythia processes

- ➡ Allows using Pythia for ANY (2→1,2,3) process in ANY model at the push of a key!

Sigma_sm_qq_ttx.h

```
#include "SigmaProcess.h"
#include "Parameters_sm.h"

using namespace std;

namespace Pythia8
{
//==================================================
// A class for calculating the matrix elements for
// Process: u u~ > t t~
// Process: c c~ > t t~
// Process: d d~ > t t~
// Process: s s~ > t t~
//--------------------------------------------------
class Sigma_sm_qq_ttx : public Sigma2Process
{
  public:

    // Constructor.
    Sigma_sm_qq_ttx() {}

    // Initialize process.
    virtual void initProc();

    // Calculate flavour-independent parts of cross section.
    virtual void sigmaKin();

    // Evaluate sigmaHat(sHat).
    virtual double sigmaHat();

    // Select flavour, colour and anticolour.
    virtual void setIdColAcol();
 ...
```

Implementation of New Physics models in MadGraph

Specification of a physics model

A new physics model can be defined by:

➡ Field content + Lagrangian

➡ Particle content + Feynman rules + coupling definitions

+ Parameters, masses, widths

Specification of a physics model

A new physics model can be defined by:

- ➡ Field content + Lagrangian
- ➡ Particle content + Feynman rules + coupling definitions

Suitable for Matrix Element generators

+ Parameters, masses, widths

Specification of a physics model

A new physics model can be defined by:

➡ Field content + Lagrangian

➡ Particle content + Feynman rules + coupling definitions

Translation done by FeynRules!

Suitable for Matrix Element generators

+ Parameters, masses, widths

Implementing New Physics Models

Ways to implement your own model in MadGraph:

- Modify existing model (e.g. only changing couplings or simplifying it)
- User model framework (new particles/interactions)
  1. Add new particles
  2. Add new interactions
  3. Enter expressions for the new couplings
  4. Script generates all needed Fortran files!
- FeynRules
  ➡ Directly from Lagrangian to implementation

Thanks for listening!

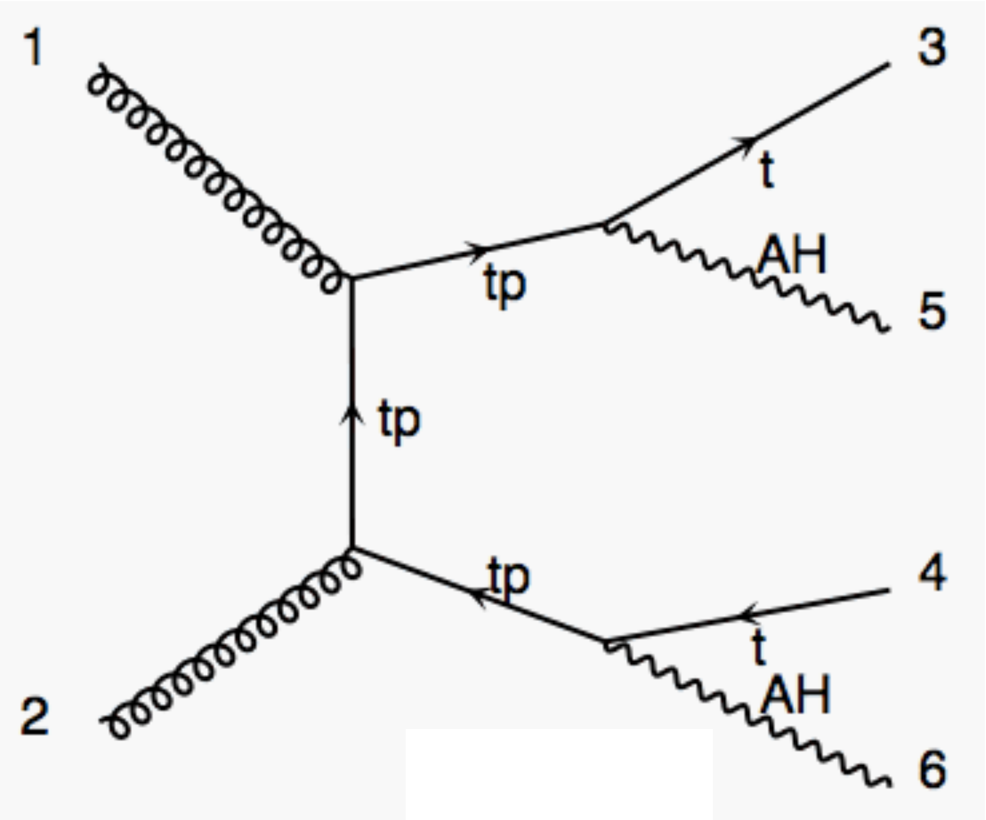
- This afternoon it will be your turn to practice!
- Generate processes, generate events, perform analyses
- Lots of fun!

User Model implementation

- Easy and quick implementation, especially when complexity of added sector is not too large
- Example:
  QCD T' pair production with
  T' → A$_h$t in Little Higgs
  model with T parity

$A_H^\mu \bar{t}'_- t$	$\frac{2ig'}{5} c_\lambda \gamma_\mu \left(c_\lambda \frac{v}{f} P_L + P_R \right)$

Hubisz, Meade [hep-ph/0411264]



User Model implementation

Specify new particles and interactions

User Model implementation

Specify new 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 TPWIDTH  T    TP     8
zp       zp         V        W      ZPMASS ZPWIDTH  S    ZP    32
# END
```

interactions.dat

```
#   USRVertex
tp tp g GG QCD
tp  t  zp GTPZP QED
t  tp  zp GTPZP QED
```

User Model implementation

Specify new 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 TPWIDTH  T    TP    8
zp       zp         V        W      ZPMASS ZPWIDTH  S    ZP    32
# END
```

interactions.dat

```
#   USRVertex
tp tp g GG QCD
tp  t  zp GTPZP QED
t  tp  zp GTPZP QED
```

Define external parameters

VariableName.dat

```
param1 #first variable name
```

User Model implementation

Specify new 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 TPWIDTH  T     TP     8
zp        zp        V         W       ZPMASS ZPWIDTH  S     ZP     32
# END
```

interactions.dat

```
#   USRVertex
tp tp g GG QCD
tp  t  zp GTPZP QED
t  tp  zp GTPZP QED
```

Define external parameters

VariableName.dat

```
param1 #first variable name
```

Run script. Modify couplings

couplings.f

```
c*************************************
c UserMode couplings
c*************************************
      GTPZP(1)=dcmplx(param1,Zero)
      GTPZP(2)=dcmplx(param1,Zero)
```

User Model implementation

Specify new 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 TPWIDTH  T     TP     8
zp       zp         V        W       ZPMASS ZPWIDTH  S     ZP     32
# END
```

interactions.dat

```
#   USRVertex
tp tp g GG QCD
tp  t  zp GTPZP QED
t  tp  zp GTPZP QED
```

Define external parameters

VariableName.dat

```
param1 #first variable name
```

Run script. Modify couplings

couplings.f

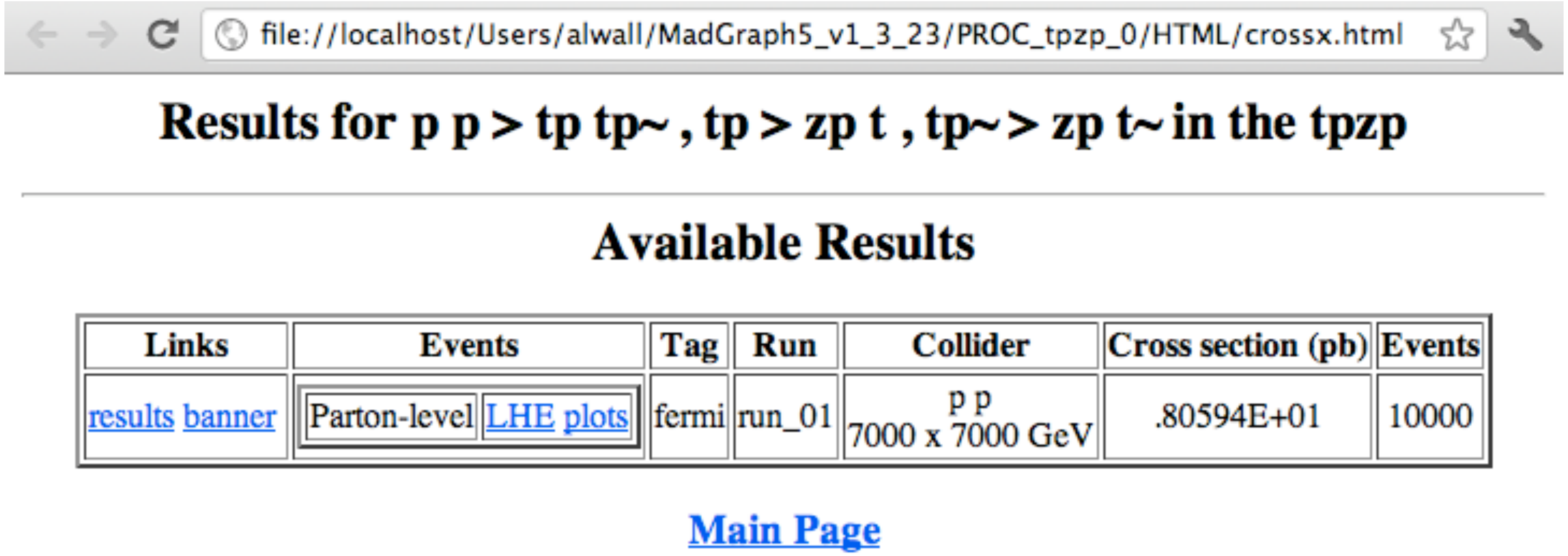
```
c*************************************
c UserMode couplings
c*************************************
      GTPZP(1)=dcmplx(param1,Zero)
      GTPZP(2)=dcmplx(param1,Zero)
```

You are ready to start generating processes!

User Model implementation

- Remember: v4 models are imported with import model_v4 modelname

```
mg5>import model_v4 tpzp
mg5>generate p p > tp tp~, tp > zp t, tp~ > zp t~
mg5>output
mg5>launch
```



file://localhost/Users/alwall/MadGraph5_v1_3_23/PROC_tpzp_0/HTML/crossx.html

Results for p p > tp tp~ , tp > zp t , tp~ > zp t~ in the tpzp

Available Results

Links	Events	Tag	Run	Collider	Cross section (pb)	Events
results banner	Parton-level LHE plots	fermi	run_01	p p 7000 x 7000 GeV	.80594E+01	10000

Main Page