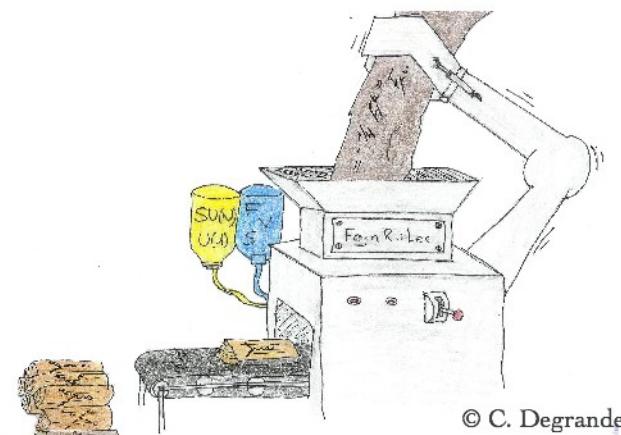


# The VISION of MadGraph and FeynRules



Johan Alwall  
Fermilab

MadGraph Spring 2011, Fermilab, May 3, 2011

# The LHC is on track!



but where are we heading...?

# What will be needed for the LHC?

Automatic  
**NLO**

Exp-TH  
communication

*Very exotic  
models*

*Exotic models*

**Effective theories**

**DECAY CHAINS**

**MATRIX  
ELEMENTS**

Advanced  
analysis  
techniques

Multi-jet samples

*Efficient Modelbuilding*

**Cluster/Grid  
computing**

Testing / robustness

Automatic scans over model space

**DECAY PACKAGES**

User friendliness

# The Beginnings



KEK Report 91-11  
January 1992  
H

## HELAS: HELicity Amplitude Subroutines for Feynman Diagram Evaluations

H. MURAYAMA, I. WATANABE and K. HAGIWARA



# The Beginnings

University of Wisconsin - Madison

MAD/PH/813

January 1994

## Automatic Generation of Tree Level Helicity Amplitudes

T. Stelzer

Physics Department, University of Durham  
Durham DH1 3LE, England

and

W. F. Long

Physics Department, University of Wisconsin-Madison  
Madison, WI 53706, USA

### Abstract

The program MadGraph is presented which automatically generates postscript Feynman diagrams and Fortran code to calculate arbitrary tree level helicity amplitudes by calling HELAS[1] subroutines. The program is written in Fortran and is available in Unix and VMS versions. MadGraph currently includes standard model interactions of QCD and QFD, but is easily modified to include additional models such as supersymmetry.

# The Beginnings



PUBLISHED BY INSTITUTE OF PHYSICS PUBLISHING FOR SISSA/ISAS

RECEIVED: September 4, 2002

REVISED: February 10, 2003

ACCEPTED: February 17, 2003

---

## MadEvent: automatic event generation with MadGraph\*

---

Fabio Maltoni and Tim Stelzer

*Department of Physics, University of Illinois at Urbana-Champaign*

*1110 West Green Street, Urbana, IL 61801, USA*

*E-mail: [tstelzer@uiuc.edu](mailto:tstelzer@uiuc.edu), [maltoni@uiuc.edu](mailto:maltoni@uiuc.edu)*

**ABSTRACT:** We present a new multi-channel integration method and its implementation in the multi-purpose event generator **MadEvent**, which is based on **MadGraph**. Given a process, **MadGraph** automatically identifies all the relevant subprocesses, generates both the amplitudes and the mappings needed for an efficient integration over the phase space, and passes them to **MadEvent**. As a result, a process-specific, stand-alone code is produced that allows the user to calculate cross sections and produce unweighted events in a standard output format. Several examples are given for processes that are relevant for physics studies at present and forthcoming colliders.

# First “official” BSM

APS » Journals » Phys. Rev. D » Volume 73 » Issue 5

< Previous Article | Next Article >

Phys. Rev. D 73, 054002 (2006) [16 pages]

## Weak boson fusion production of supersymmetric particles at the CERN LHC

Abstract

References

Citing Articles (18)

Download: PDF (259 kB) Buy this article Export: BibTeX or EndNote (RIS)

G. C. Cho<sup>1,\*</sup>, K. Hagiwara<sup>2,†</sup>, J. Kanzaki<sup>3,‡</sup>, T. Plehn<sup>4,§</sup>, D. Rainwater<sup>5,\*\*</sup>, and T. Stelzer<sup>6,||</sup>

<sup>1</sup>Ochanomizu University, Tokyo, Japan

<sup>2</sup>Theory Division, KEK, Tsukuba, Japan

<sup>3</sup>Institute of Particle and Nuclear Studies, KEK, Tsukuba, Japan

<sup>4</sup>Max Planck Institute for Physics, Munich, Germany and School of Physics, University of Edinburgh, Scotland

<sup>5</sup>Department of Physics and Astronomy, University of Rochester, Rochester, New York, USA

<sup>6</sup>Department of Physics, University of Illinois, Urbana, Illinois, USA

Received 9 January 2006; published 6 March 2006

We present a complete calculation of weak boson fusion production of colorless supersymmetric particles at the LHC, using the new matrix element generator SUSY-MADGRAPH . The cross sections are small, generally at the attobarn level, with a few notable exceptions which might provide additional supersymmetric parameter measurements. We discuss in detail how to consistently define supersymmetric weak couplings to preserve unitarity of weak gauge boson scattering amplitudes to fermions, and derive sum rules for weak supersymmetric couplings.

# The Web Generation



PUBLISHED BY INSTITUTE OF PHYSICS PUBLISHING FOR SISSA

RECEIVED: July 2, 2007

ACCEPTED: August 22, 2007

PUBLISHED: September 6, 2007

## MadGraph/MadEvent v4: the new web generation

Johan Alwall,<sup>a</sup> Pavel Demin,<sup>b</sup> Simon de Visscher,<sup>b</sup> Rikkert Frederix,<sup>b</sup> Michel Herquet,<sup>b</sup> Fabio Maltoni,<sup>b</sup> Tilman Plehn,<sup>c</sup> David L. Rainwater<sup>d</sup> and Tim Stelzer<sup>e</sup>

Center for Particle Physics and Phenomenology - CP3

MadGraph Version 4  
UCL UIUC Fermi  
by the MG/ME Development team

Generate Process Register Tools My Database Cluster Status Downloads (needs registration) Wiki/Docs Admin

Generate Code On-Line

High Energy Physics Illinois

This material is based upon work supported by the National Science Foundation under Grant No. 0426272. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

The MadGraph homepage  
UCL UIUC Fermi  
by the MG/ME Development team

Generate Process Register Tools My Database Cluster Status Downloads (needs registration) Wiki/Docs Admin

MUSEO STORICO DELLA FISICA E CENTRO STUDI E RICERCHE

MadGraph Version 4  
UCL UIUC Fermi  
by the MG/ME Development team

Generate Process Register Tools My Database Cluster Status Downloads (needs registration) Wiki/Docs Admin

To improve our web services we now request that you register. Registration is quick and free. You may register for a password by clicking

# Tools and BSM functionality

## New Developments in MadGraph/MadEvent

J. Alwall<sup>\*</sup>, P. Artoisenet<sup>†</sup>, S. de Visscher<sup>†</sup>, C. Duhr<sup>†</sup>, R. Frederix<sup>†</sup>, M. Herquet<sup>†</sup> and O. Mattelaer<sup>†</sup>

<sup>\*</sup>SLAC, Stanford University, Menlo Park, CA 94025, E-mail: alwall@slac.stanford.edu  
<sup>†</sup>Université Catholique de Louvain, Chemin du Cyclotron 2, B-1348 Louvain-la-Neuve, Belgium

**Abstract.** We here present some recent developments of MadGraph/MadEvent since the latest published version, 4.0. These developments include: Jet matching with Pythia parton showers for both Standard Model and Beyond the Standard Model processes, decay chain functionality, decay width calculation and decay simulation, process generation for the Grid, a package for calculation of quarkonium amplitudes, calculation of Matrix Element weights for experimental events, automatic dipole subtraction for next-to-leading order calculations, and an interface to FeynRules, a package for automatic calculation of Feynman rules and model files from the Lagrangian of any New Physics model.

**Keywords:** Monte Carlo Simulations, Beyond the Standard Model, New Physics, Matrix Element  
PACS: 24.10.Lx, 13.85.Hd, 12.60.-i

arXiv:0809.2410

## Automatic generation of quarkonium amplitudes in NRQCD

Pierre Artoisenet, Fabio Maltoni

Centre for Particle Physics and Phenomenology (CP3)  
Université Catholique de Louvain  
Chemin du Cyclotron 2, B-1348 Louvain-la-Neuve, Belgium  
E-mails: [pierre.artoisenet@uclouvain.be](mailto:pierre.artoisenet@uclouvain.be), [fabio.maltoni@uclouvain.be](mailto:fabio.maltoni@uclouvain.be)

Tim Stelzer

Department of Physics, University of Illinois at Urbana-Champaign,  
1110 West Green Street, Urbana, IL 61801  
E-mail: [tstelzer@uiuc.edu](mailto:tstelzer@uiuc.edu)

arXiv:0712.2770

## Automation of the matrix element reweighting method

Pierre Artoisenet <sup>a</sup>, Vincent Lemaître <sup>b</sup>, Fabio Maltoni <sup>b</sup>, Olivier Mattelaer <sup>b,c</sup>

<sup>a</sup> Physics Department, The Ohio State University,  
Columbus, Ohio 43210, USA

<sup>b</sup> Centre for Cosmology, Particle Physics and Phenomenology (CP3)  
Université Catholique de Louvain

arXiv:1007.3300

## HELAS and MadGraph/MadEvent with spin-2 particles

K. Hagiwara<sup>1</sup>, J. Kanzaki<sup>2,a</sup>, Q. Li<sup>3,b</sup>, and K. Mawatari<sup>4,c</sup>

<sup>1</sup> KEK, Theory Division and Sokendai, Tsukuba 305-0801, Japan

<sup>2</sup> KEK, Tsukuba 305-0801, Japan

<sup>3</sup> Institut für Theoretische Physik, Universität Karlsruhe, Postfach 6980, D-76128 Karlsruhe, Germany

<sup>4</sup> School of Physics, Korea Institute for Advanced Study, Seoul 130-722, Korea

arXiv:0805.2554

## HELAS and MadGraph with spin-3/2 particles

K. Hagiwara<sup>1</sup>, K. Mawatari<sup>2,3,a</sup>, and Y. Takaesu<sup>1,b</sup>

<sup>1</sup> KEK Theory Center, and Sokendai, Tsukuba 305-0801, Japan

<sup>2</sup> Theoretische Natuurkunde en IIHE/ELEM, Vrije Universiteit Brussel,  
and International Solvay Institutes, Pleinlaan 2, B-1050 Brussels, Belgium

<sup>3</sup> Institut für Theoretische Physik, Universität Heidelberg, Philosophenweg 16, D-69120 Heidelberg, Germany

arXiv:1010.4255

See talk by Kentarou

# Speed

## Fast calculation of HELAS amplitudes using graphics processing unit (GPU)

K. Hagiwara<sup>1</sup>, J. Kanzaki<sup>2,a</sup>, N. Okamura<sup>2,b</sup>, D. Rainwater<sup>3</sup>, and T. Stelzer<sup>4,c</sup>

<sup>1</sup> KEK Theory Center and Sokendai, Tsukuba 305-0801, Japan

<sup>2</sup> KEK, Tsukuba 305-0801, Japan

<sup>3</sup> Space and Geophysics Laboratory, Applied Research Laboratories, University of Texas, Austin, TX 78758, USA

<sup>4</sup> Dept. of Physics, University of Illinois, Urbana, IL, USA

arXiv:0908.4403

Generating QCD amplitudes in the color-flow basis  
with MadGraph

Kaoru Hagiwara<sup>1</sup> and Yoshitaro Takaesu<sup>2</sup>

KEK Theory Center and Sokendai  
Tsukuba, 305-0801, Japan

## Calculation of HELAS amplitudes for QCD processes using graphics processing unit (GPU)

arXiv:1010.0748

K. Hagiwara<sup>1</sup>, J. Kanzaki<sup>2,a</sup>, N. Okamura<sup>2,b</sup>, D. Rainwater<sup>3</sup>, and T. Stelzer<sup>4,c</sup>

<sup>1</sup> KEK Theory Center and Sokendai, Tsukuba 305-0801, Japan

<sup>2</sup> KEK, Tsukuba 305-0801, Japan

<sup>3</sup> Space and Geophysics Laboratory, Applied Research Laboratories, University of Texas, Austin, TX 78758, USA

<sup>4</sup> Dept. of Physics, University of Illinois, Urbana, IL, USA

Received: date / Revised version: September 29, 2009

arXiv:0909.5257

See talks by Junichi and Yoshitaro

# Next-to-leading Order

## MadDipole: Automation of the Dipole Subtraction Method in MadGraph/MadEvent

R. Frederix

*Center for Particle Physics and Phenomenology (CP3),  
Université Catholique de Louvain,  
Chemin du Cyclotron 2, 1348 Louvain-la-Neuve, Belgium*

T. Gehrmann, N. Greiner

*Institut für Theoretische Physik,  
Universität Zürich,  
Winterthurerstrasse 190, 8057 Zürich, Switzerland*  
arXiv:1004.2905

## Automation of next-to-leading order computations in QCD: the FKS subtraction

Rikkert Frederix,<sup>a,1</sup> Stefano Frixione,<sup>a,b,2</sup> Fabio Maltoni<sup>c</sup> and Tim Stelzer<sup>d</sup>

arXiv:0908.4272

## Integrated dipoles with MadDipole in the MadGraph framework

R. Frederix<sup>a</sup> T. Gehrmann<sup>a</sup>, N. Greiner<sup>a,b</sup>

<sup>a</sup> *Institut für Theoretische Physik, Universität Zürich,  
Winterthurerstrasse 190, 8057 Zürich, Switzerland*

<sup>b</sup> *Department of Physics, University of Illinois at Urbana-Champaign,  
1110 West Green Street, Urbana, IL 61801, USA*

arXiv:1004.2905

## Automation of one-loop QCD corrections

Valentin Hirschi

*ITPP, EPFL, CH-1015 Lausanne, Switzerland*

Rikkert Frederix

*Institut für Theoretische Physik, Universität Zürich, Winterthurerstrasse 190,  
CH-8057 Zürich, Switzerland*

Stefano Frixione\*

*PH Department, TH Unit, CERN, CH-1211 Geneva 23, Switzerland  
ITPP, EPFL, CH-1015 Lausanne, Switzerland*

Maria Vittoria Garzelli

*INFN, Sezione di Milano, I-20133 Milano, Italy  
Departamento de Física Teórica y del Cosmos y CAFPE  
Universidad de Granada, E-18071 Granada, Spain*

Fabio Maltoni

*Centre for Cosmology, Particle Physics and Phenomenology (CP3)  
Université catholique de Louvain  
Chemin du Cyclotron 2, B-1348 Louvain-la-Neuve, Belgium*

Roberto Pittau†

*PH Department, TH Unit, CERN, CH-1211 Geneva 23, Switzerland*

**ABSTRACT:** We present the complete automation of the computation of one-loop QCD corrections, including UV renormalization, to an arbitrary scattering process in the Standard Model. This is achieved by embedding the OPP integrand reduction technique, as implemented in CutTools, into the MadGraph framework. By interfacing the tool so constructed, which we dub MadLoop, with MadFKS, the fully automatic computation of any infrared-safe observable at the next-to-leading order in QCD is attained. We demonstrate the flexibility and the reach of our method by calculating the production rates for a variety of processes at the 7 TeV LHC.

arXiv:1103.0621

See talk by Valentin

- + Automatic MC@NLO (to be presented by Paolo)
- + Automatic SUSY@NLO (see talk by MadGOLEM team)

# MadGraph

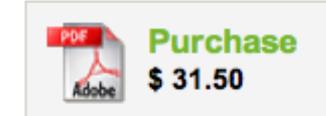
- MadGraph has evolved from pure matrix element generator to a multi-purpose platform for a large number of automatized tools
- Very strong trademark – very powerful tool
- However, inherent limitations and rigid output structure still place a limit on user friendliness and development possibilities
- Many developments have been veritable “tour-de-force” efforts with massive post-processing

# FeynRules

## FeynRules – Feynman rules made easy<sup>☆</sup>

Neil D. Christensen<sup>a</sup>,  and Claude Duhr<sup>b</sup>,  

<sup>a</sup>Department of Physics and Astronomy, Michigan State University, East Lansing, MI 48824, USA



<sup>b</sup>Université catholique de Louvain, Center for Particle Physics and Phenomenology (CP3), Chemin du cyclotron, 2, B-1348 Louvain-La-Neuve, Belgium

Received 4 August 2008; revised 14 January 2009; accepted 14 February 2009. Available online 26 February 2009.

### Abstract

In this paper we present FeynRules, a new *Mathematica* package that facilitates the implementation of new particle physics models. After the user implements the basic model information (e.g., particle content, parameters and Lagrangian), FeynRules derives the Feynman rules and stores them in a generic form suitable for translation to any Feynman diagram calculation program. The model can then be translated to the format specific to a particular Feynman diagram calculator via FeynRules translation interfaces. Such interfaces have been written for CalcHEP/CompHEP, FeynArts/FormCalc, MadGraph/MadEvent and Sherpa, making it possible to write a new model once and have it work in all of these programs. In this paper, we describe how to implement a new model, generate the Feynman rules, use a generic translation interface, and write a new translation interface. We also discuss the details of the FeynRules code.

–

arXiv:0806.4194

# FeynRules

## FeynRules – Feynman rules made easy<sup>☆</sup>

Neil D. Christensen<sup>a</sup>,  and Claude Duhr<sup>b</sup>, , 

<sup>a</sup>Department of Physics and Astronomy, Michigan State University, East Lansing, MI 48824, USA

<sup>b</sup>Université catholique de Louvain, Center for Particle Physics and Phenomenology (CP3), Chemin du cyclotron, 2, B-1348 Louvain-La-Neuve, Belgium

Received 4 August 2008; revised 14 January 2009; accepted 14 February 2009. Available online 26 February 2009.

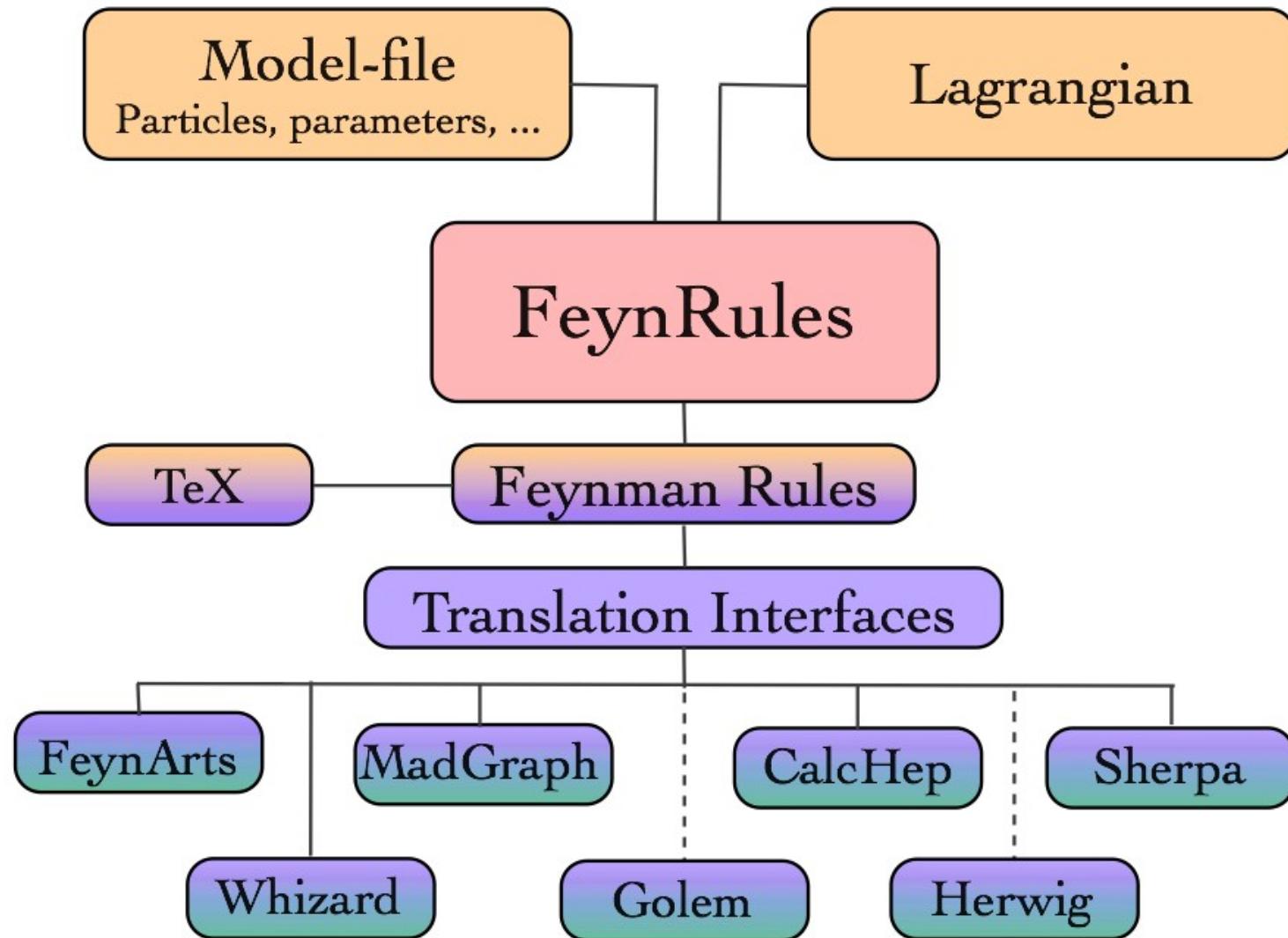
### Abstract

In this paper we present FeynRules, a new *Mathematica* package that facilitates the implementation of new particle physics models. After the user implements the basic model information (e.g., particle content, parameters and Lagrangian), FeynRules derives the Feynman rules and stores them in a generic form suitable for translation to any Feynman diagram calculation program. The model can then be translated to the format specific to a particular Feynman diagram calculator via FeynRules translation interfaces. Such interfaces have been written for CalcHEP/CompHEP, FeynArts/FormCalc, MadGraph/MadEvent and Sherpa, making it possible to write a new model once and have it work in all of these programs. In this paper, we describe how to implement a new model, generate the Feynman rules, use a generic translation interface, and write a new translation interface. We also discuss the details of the FeynRules code.



arXiv:0806.4194

# FeynRules



# FeynRules

- Easy-to-use but powerful Mathematica package
  - Generation of Feynman rules from any Lagrangean
  - Output of generator specific files using multiple generator translation interfaces
  - Continuous new improvements and developments

See talk by FeynRules team

- However – output limited by the capabilities of the target generator in terms of Lorentz/color structures, multiparticle vertices, effective vertices etc.
  - e.g., implemented HELAS routines

# Vision (anno 2008/9)

- Using FeynRules as a cornerstone, allows for unprecedented validation and testing of models, and efficient communication between theorists and experimentalists at any point in the simulation chain

**A comprehensive approach to new physics simulations**

---

Neil Christensen<sup>(1)</sup>, Priscila de Aquino<sup>(2,3)</sup>, Celine Degrande<sup>(2)</sup>, Claude Duhr<sup>(2)</sup>,  
Benjamin Fuks<sup>(4)</sup>, Michel Herquet<sup>(5)</sup>, Fabio Maltoni<sup>(2)</sup>, Steffen Schumann<sup>(6)</sup>

<sup>(1)</sup> Department of Physics and Astronomy, Michigan State University, East Lansing, MI  
48824, USA

<sup>(2)</sup> Center for Particle Physics and Phenomenology, Université Catholique de Louvain,  
B-1348 Louvain-la-Neuve, Belgium

<sup>(3)</sup> Instituut voor Theoretische Fysica, Katholieke Universiteit Leuven, Celestijnenlaan  
200D, B-3001 Leuven, Belgium

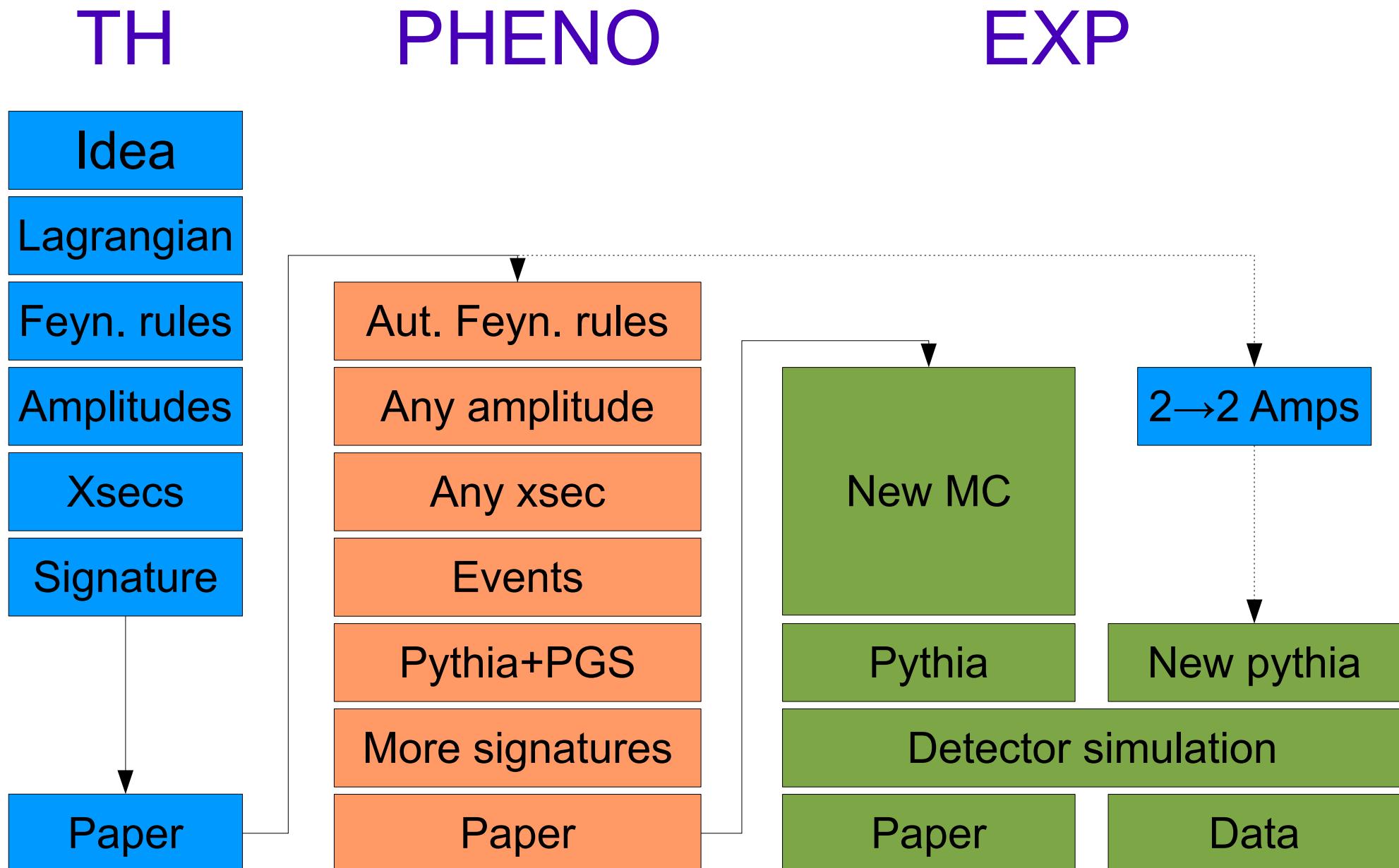
<sup>(4)</sup> Institut Pluridisciplinaire Hubert Curien, Université de Strasbourg, IN2P3-CNRS,  
BP28, F-67037 Strasbourg Cedex 2, France

<sup>(5)</sup> Nikhef Theory Group, Kruislaan 409, 1098 SJ Amsterdam, The Netherlands

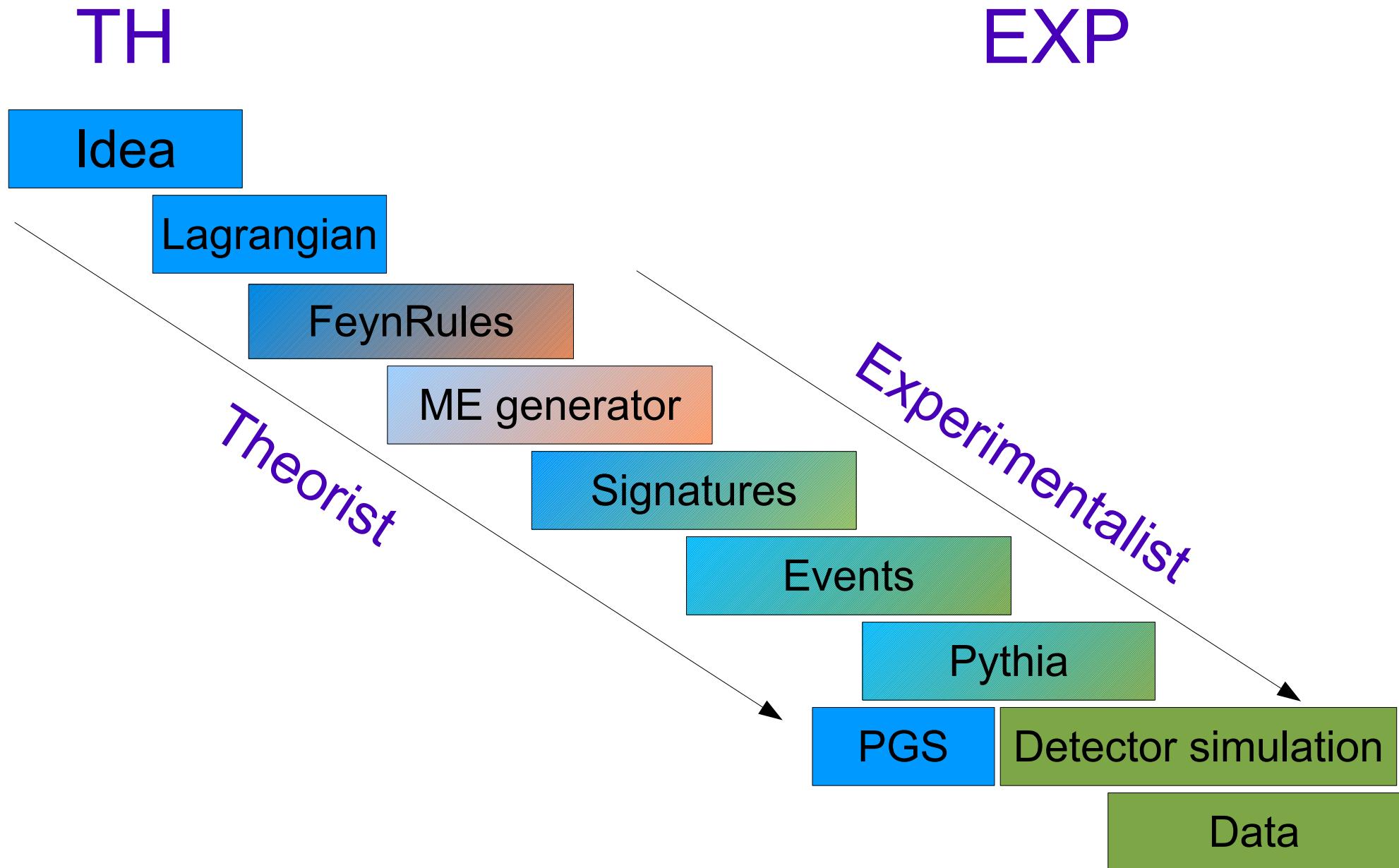
<sup>(6)</sup> Institut für Theoretische Physik, Universität Heidelberg, Philosophenweg 16,  
D-69120, Heidelberg, Germany

arXiv:0906.2474

# Replace this...



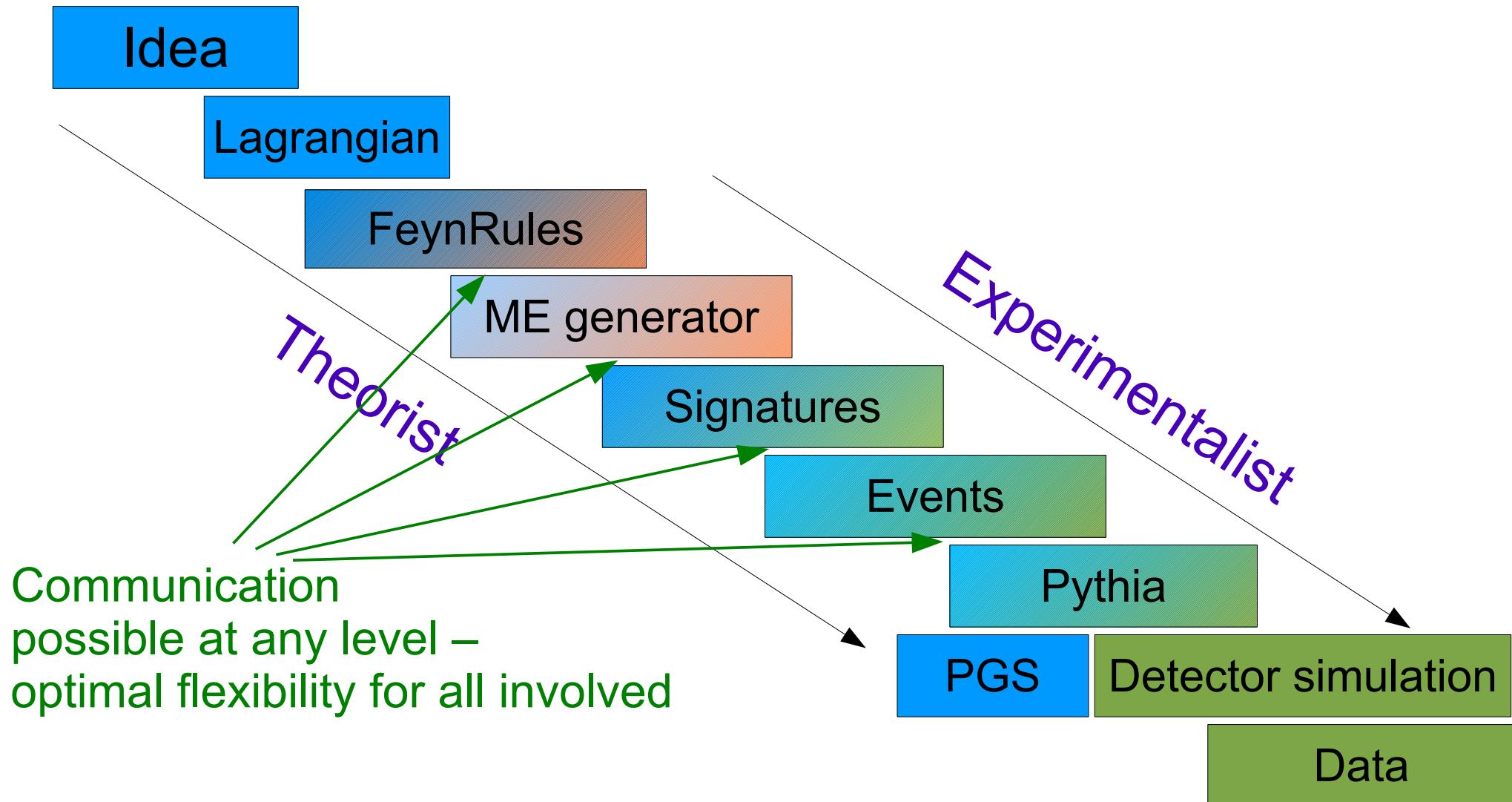
# ... by this!



# ... by this!

TH

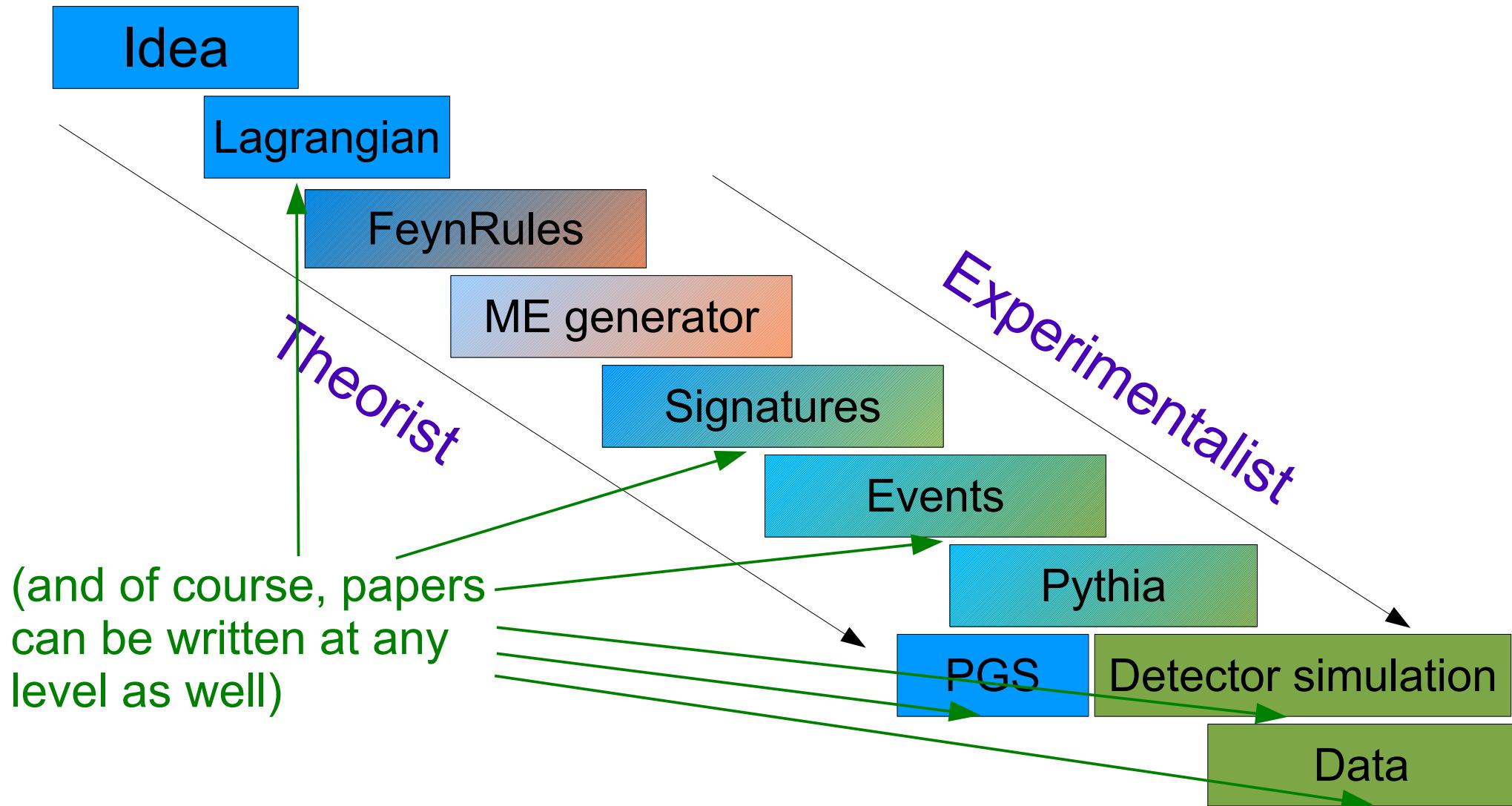
EXP



# ... by this!

TH

EXP



# Since then, we've raised the bar!

- Development of MadGraph 5 started in November 2009
- Explicit goal – remove all limitations on old MadGraph
  - Speed
  - Number of particles
  - Types of interactions
  - Output languages and formats
  - Flexibility and modularity
- Did we succeed?

MadGraph Home Page

High Energy Physics Illinois

This material is based upon work supported by the National Science Foundation under Grant No. 0426272. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

The MadGraph homepage  
UCL UIUC Fermi  
by the MG/ME Development team

Generate Process   Register   Tools   My Database   Cluster Status   Downloads (needs registration)   Wiki/Docs   Admin

## Generate processes online using MadGraph 5

To improve our web services we request that you register. Registration is quick and free. You may register for a password by clicking [here](#). You can still use MadGraph 4 [here](#).

Code can be generated either by:

### I. Fill the form:

Model:	<input type="button" value="SM"/>	<a href="#">Model descriptions</a>
Input Process:	<input type="text"/>	
Example: p p > w+ j j QED=3, w+ > l+ vl		
p and j definitions:	<input type="button" value="p=j=d u s c d~ u~ s~ c~ g"/>	
sum over leptons:	<input type="button" value="l+ = e+, mu+ ta+; l- = e-, mu- ta-; vl = ve, vm, vt; vl~ = ve~, vm~, vt~"/>	
<input type="button" value="Submit"/>		

### II. Upload the proc\_card.dat

[Process card examples](#)

proc\_card format

No file chosen

and  it to the server.

# Speed

Matrix Element  
generation:

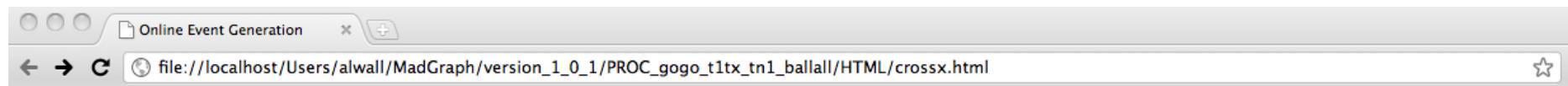
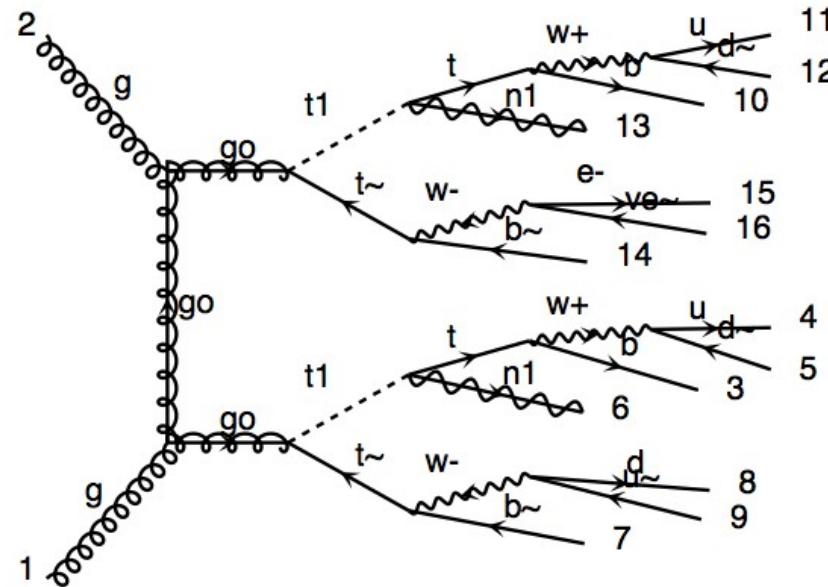
Process	MADGRAPH 4	MADGRAPH 5	Subprocesses	Diagrams
$pp \rightarrow jjj$	29.0 s	54.4 s	34	307
$pp \rightarrow jjl^+l^-$	341 s	258 s	108	1216
$pp \rightarrow jjje^+e^-$	1151 s	654 s	141	9012
$u\bar{u} \rightarrow e^+e^-e^+e^-e^+e^-$	772 s	175 s	1	3474
$gg \rightarrow ggggg$	2788 s	1049 s	1	7245
$pp \rightarrow jj(W^+ \rightarrow l^+\nu_l)$	146 s	70 s	82	304
$pp \rightarrow t\bar{t} + \text{full decays}$	5640 s	22 s	27	45
$pp \rightarrow \tilde{q}/\tilde{g} \tilde{q}/\tilde{g}$	222 s	286 s	313	475
7 particle decay chain	383 s	5.2 s	1	6
$gg \rightarrow (\tilde{g} \rightarrow u\bar{u}\tilde{\chi}_1^0)(\tilde{g} \rightarrow u\bar{u}\tilde{\chi}_1^0)$	70 s	5.5 s	1	48
$pp \rightarrow (\tilde{g} \rightarrow jj\tilde{\chi}_1^0)(\tilde{g} \rightarrow jj\tilde{\chi}_1^0)$	—	551 s	144	11008

Matrix Element  
evaluation  
(Fortran):

Process	Function calls		Run time	
	MG 4	MG 5	MG 4	MG 5
$u\bar{u} \rightarrow e^+e^-$	8	8	$< 6\mu\text{s}$	$< 6\mu\text{s}$
$u\bar{u} \rightarrow e^+e^-e^+e^-$	110	80	0.22 ms	0.14 ms
$u\bar{u} \rightarrow e^+e^-e^+e^-e^+e^-$	6668	3775	46.5 ms	19.0 ms
$u\bar{u} \rightarrow d\bar{d}$	6	6	$< 4\mu\text{s}$	$< 4\mu\text{s}$
$u\bar{u} \rightarrow d\bar{d}g$	16	16	$27 \mu\text{s}$	$27 \mu\text{s}$
$u\bar{u} \rightarrow d\bar{d}gg$	85	67	0.42 ms	0.31 ms
$u\bar{u} \rightarrow d\bar{d}ggg$	748	515	10.8 ms	6.75 ms
$u\bar{u} \rightarrow u\bar{u}gg$	160	116	1.24 ms	0.80 ms
$u\bar{u} \rightarrow u\bar{u}ggg$	1468	960	35.7 ms	17.2 ms
$u\bar{u} \rightarrow d\bar{d}dd\bar{d}$	42	33	$84 \mu\text{s}$	$83 \mu\text{s}$
$u\bar{u} \rightarrow d\bar{d}dd\bar{d}g$	310	197	1.88 ms	1.15 ms
$u\bar{u} \rightarrow d\bar{d}dd\bar{d}gg$	3372	1876	141 ms	34.4 ms
$u\bar{u} \rightarrow d\bar{d}dd\bar{d}dd\bar{d}$	1370	753	42.5 ms	6.6 ms

+ Ongoing work  
with recursion  
for multiparton  
calculations in  
MG5, see talk by  
Yoshitaro

# Number of particles



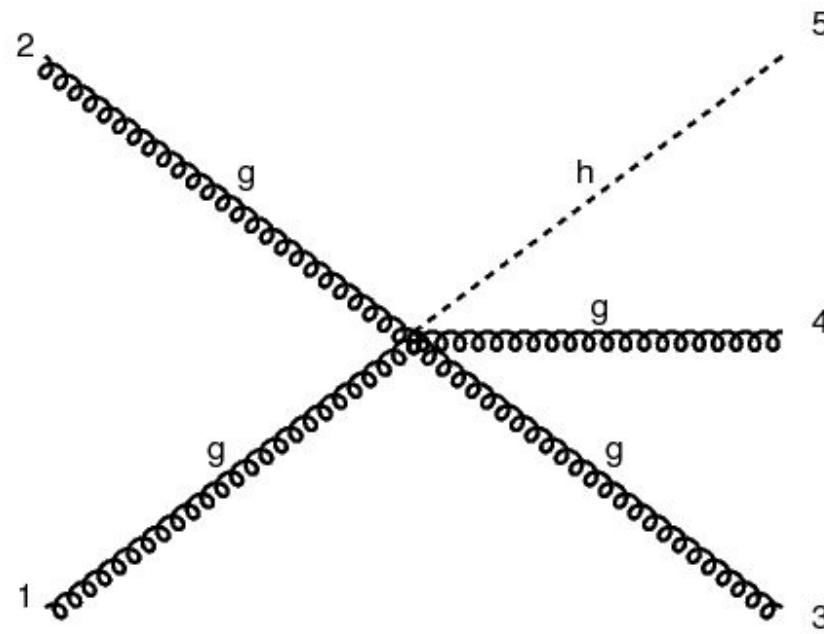
**Results for  $g g \rightarrow go go , (go \rightarrow t1 t\sim , t\sim \rightarrow b\sim \text{ all all / } h+) , (t1 \rightarrow t n1 , t \rightarrow b \text{ all all / } h+)$  in the mssm**

## Available Results

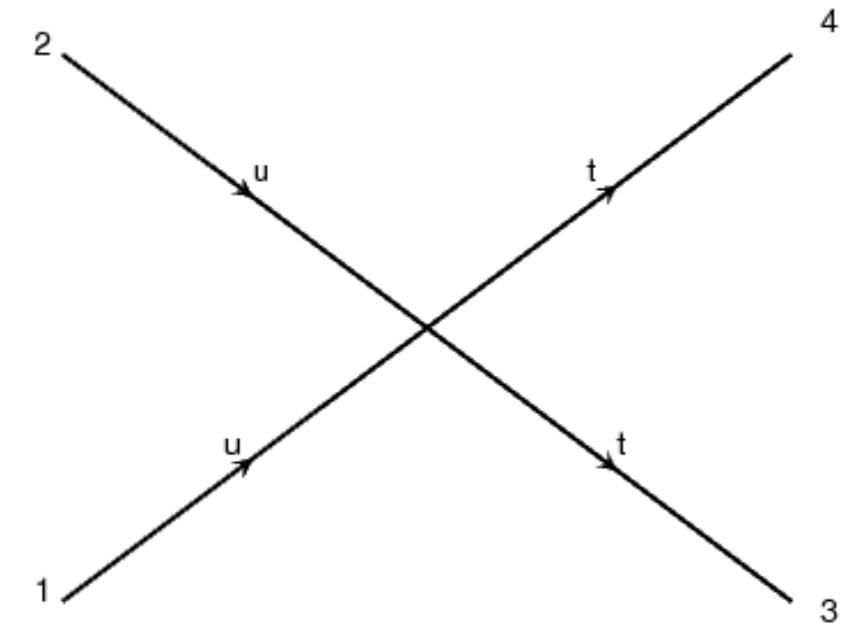
Links	Events	Tag	Run	Collider	Cross section (pb)	Events
<a href="#">results banner</a>	Parton-level <a href="#">LHE</a>	fermi	<a href="#">test</a>	$p p$ 7000 x 7000 GeV	.33857E-03	10000

[Main Page](#)

# Types of interactions



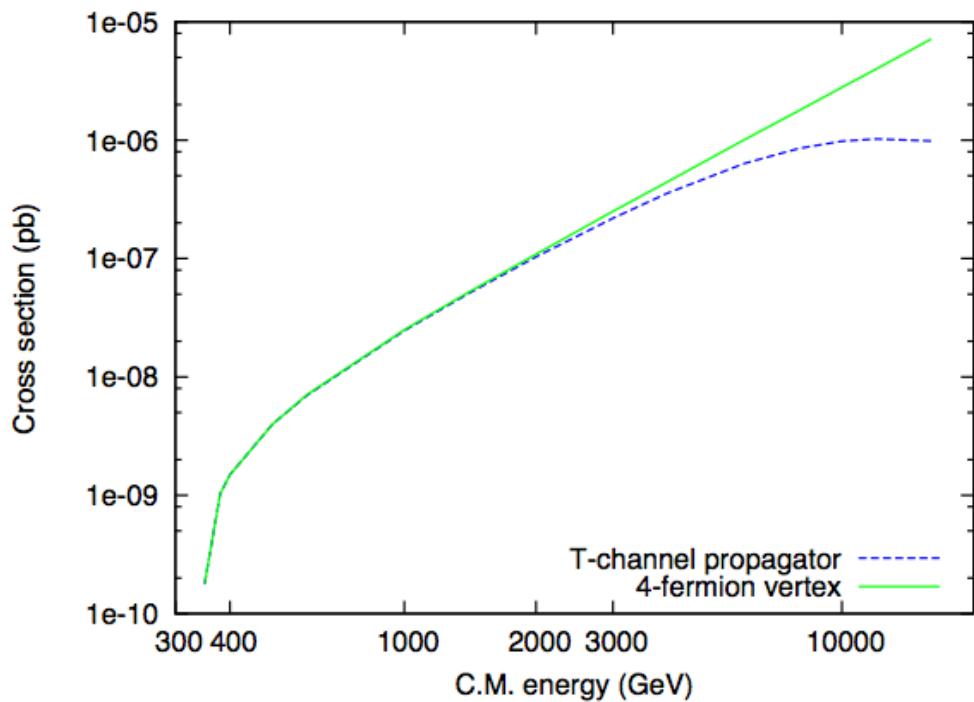
Higgs Effective Theory



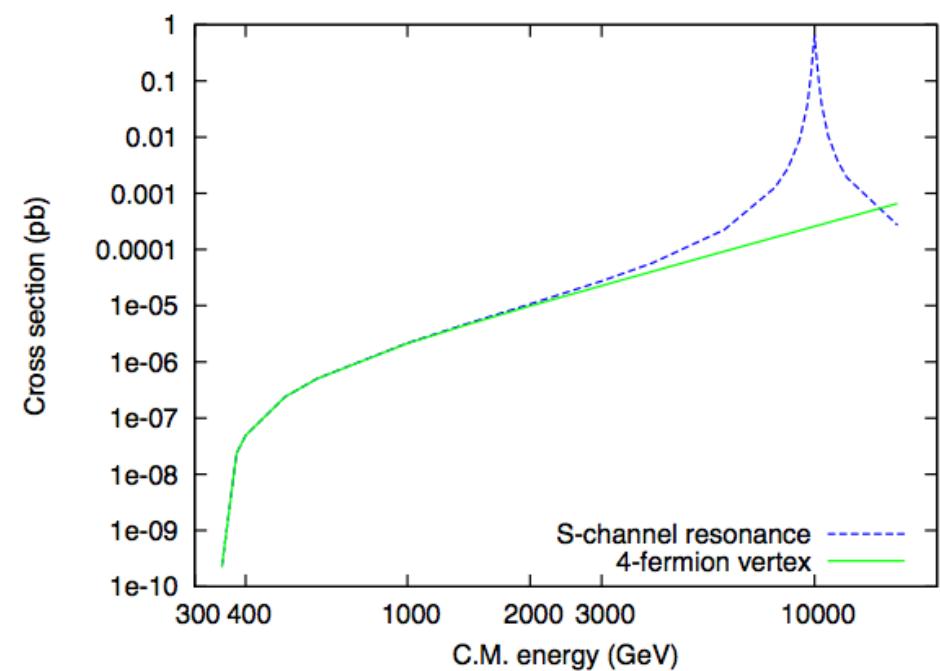
Multi-fermion vertices

# Types of interactions

Comparisons between explicit propagators  
and 4-fermion vertex



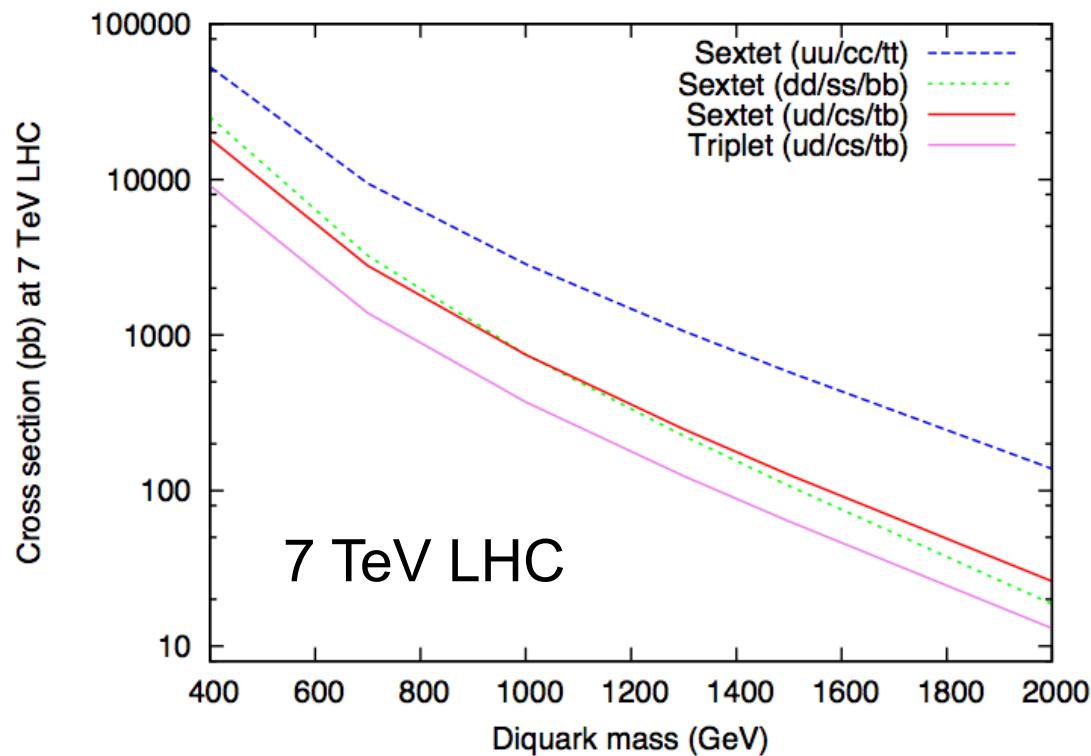
t-channel  $u\bar{u} \rightarrow t\bar{t}$



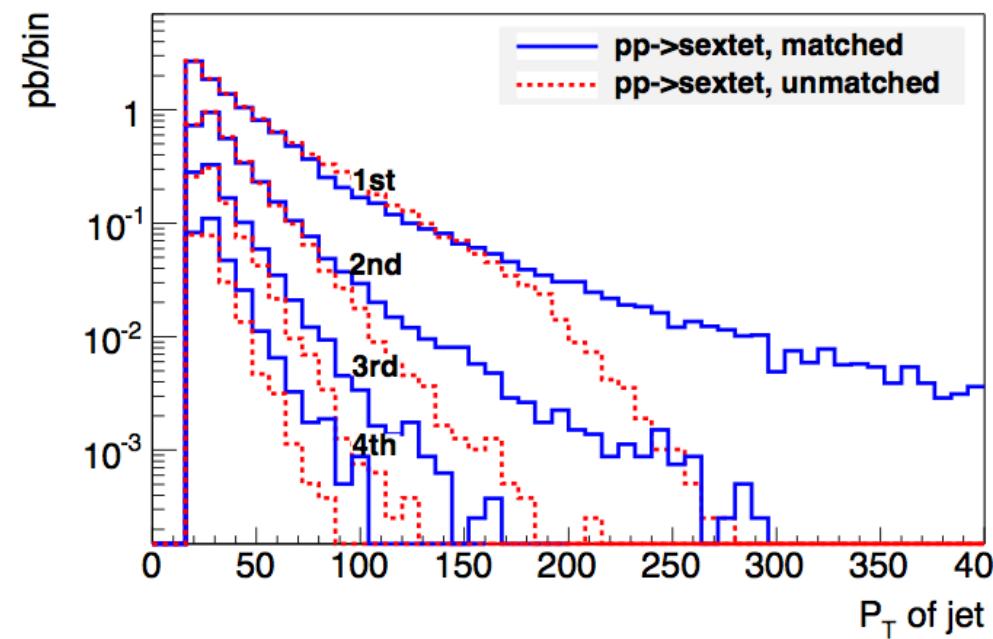
s-channel  $u\bar{u} \rightarrow t\bar{t}$

# Types of interactions

## Color sextet and $\epsilon^{ijk}$ implementations



Diquark cross sections with coupling 0.01



Jet  $p_T$ :s, fully matched  
 $pp \rightarrow D + 0, 1, 2$  jets

# Output languages and formats

Process	Subprocess directories		Channels for survey		Directory size	
	ME 4	ME 5	ME 4	ME 5	ME 4	ME 5
$pp \rightarrow W^+ j$	6	2	12	4	79 MB	35 MB
$pp \rightarrow W^+ jj$	41	4	138	29	438 MB	64 MB
$pp \rightarrow W^+ jjj$	73	5	1164	184	842 MB	110 MB
$pp \rightarrow W^+ jjjj$	296	7	15029	1327	3.8 GB	352 MB
$pp \rightarrow l^+ l^- j$	12	2	48	8	149 MB	44 MB
$pp \rightarrow l^+ l^- jj$	54	4	586	58	612 MB	83 MB
$pp \rightarrow l^+ l^- jjj$	86	5	5408	368	1.2 GB	151 MB
$pp \rightarrow l^+ l^- jjjj$	235	7	63114	2500	5.3 GB	662 MB
$pp \rightarrow tt$	3	2	5	4	49 MB	39 MB
$pp \rightarrow t\bar{t}j$	7	3	45	25	97 MB	56 MB
$pp \rightarrow t\bar{t}jj$	22	5	417	188	274 MB	98 MB
$pp \rightarrow t\bar{t}jjj$	34	6	3816	1300	620 MB	209 MB

Compactified and optimized multiprocess output for MadEvent

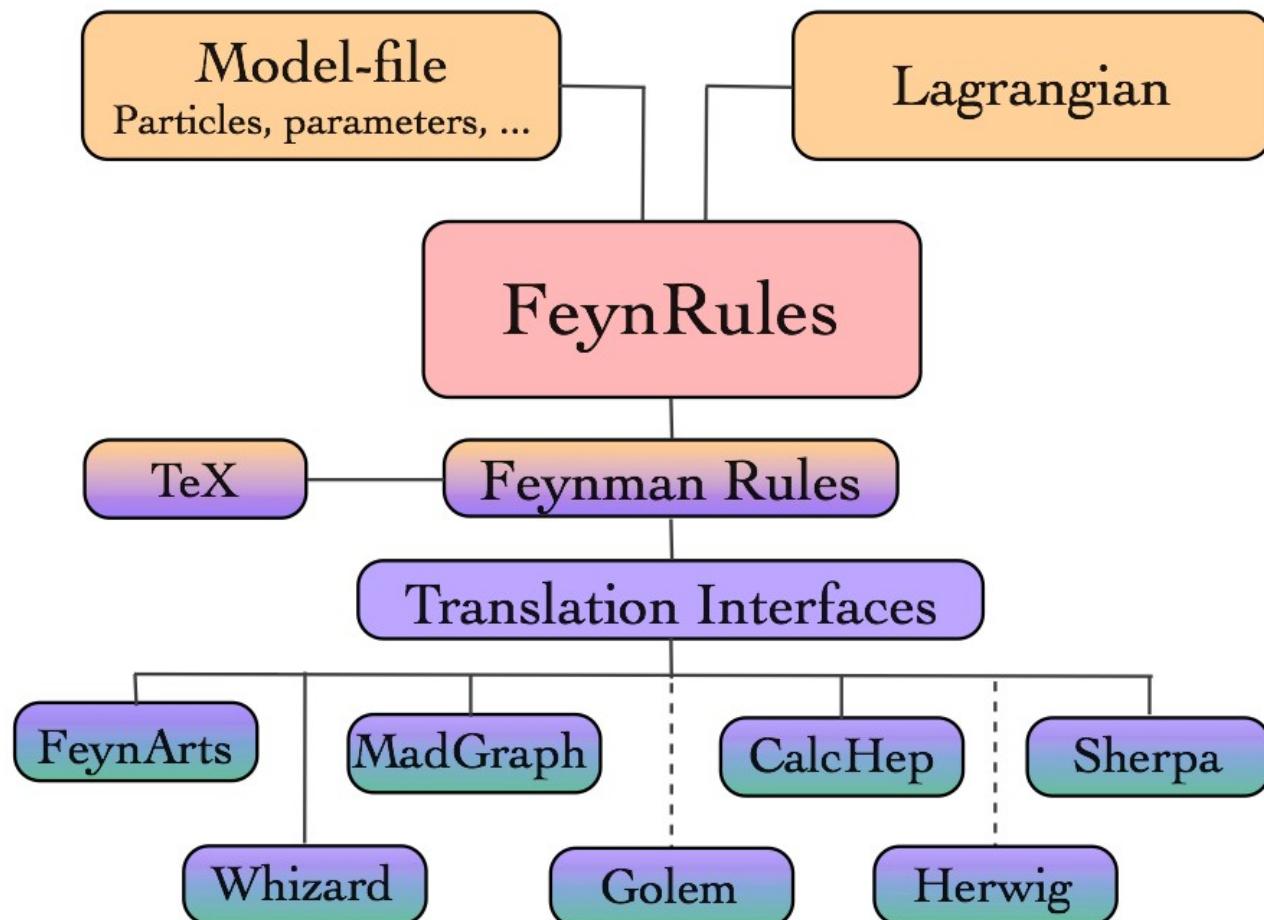
+ Standalone Matrix Element output in Fortran and C++

```
#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~ QED=0 @1
// Process: c c~ > t t~ QED=0 @1
// Process: d d~ > t t~ QED=0 @1
// Process: s s~ > t t~ QED=0 @1
//-----
```

```
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();
    ...
}
```

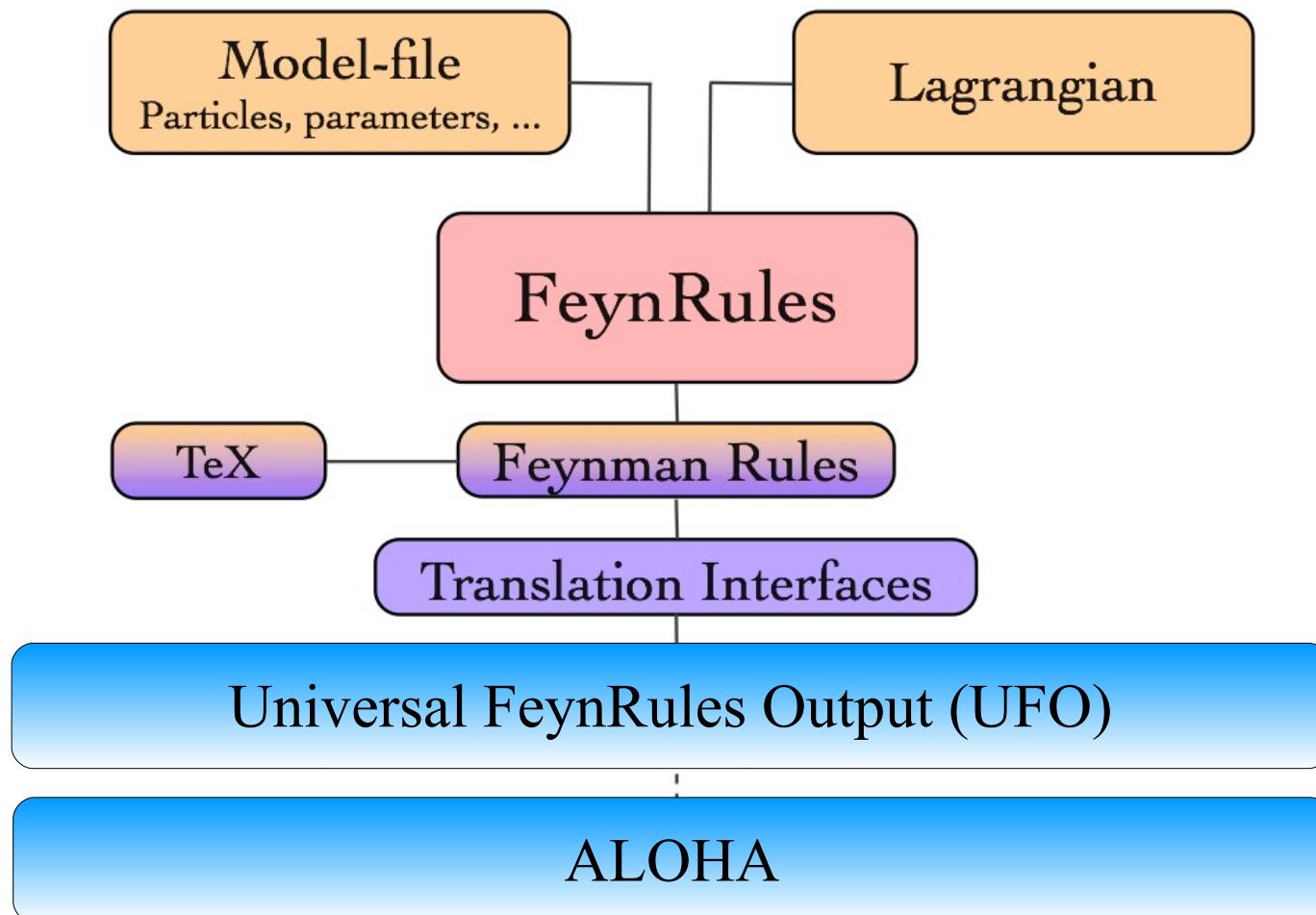
# But the biggest advances..

... are in a new interface from FeynRules!



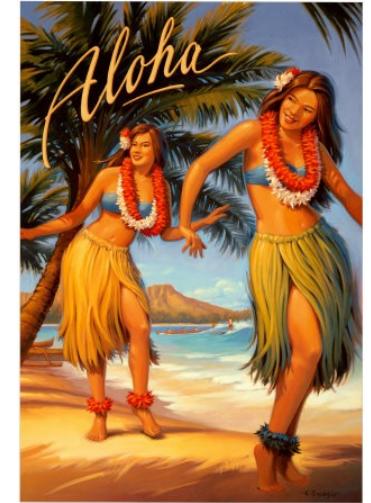
# But the biggest advances..

... are in a new interface from FeynRules!



# The biggest advances

- Universal FeynRules Output (UFO)
  - Includes color and Lorentz structure
  - Allows for complete specification of effective/non-renormalizable vertices
  - Allows for automatic output of model parameter calculations for any model and language
- Automatic Language-independent Output of Helicity Amplitudes (ALOHA)
  - Automatic generation the necessary helicity amplitude code for any new model (including effective theories, multi-fermion vertices,...) in Fortran/C++/Python/...



See talk by Olivier

# Universal FeynRules Output (UFO)

**particles.py:**

```
G = Particle(pdg_code = 21,  
             name = 'G',  
             antiname = 'G',  
             spin = 3,  
             color = 8,  
             mass = 'ZERO',  
             width = 'ZERO',  
             texname = 'G',  
             antitexname = 'G',  
             line = 'curly',  
             charge = 0,  
             LeptonNumber = 0,  
             GhostNumber = 0)
```

**lorentz.py:**

```
VVV1 = Lorentz(name = 'VVV1',  
                spins = [ 3, 3, 3 ],  
                Structure =  
                    'P(3,1)*Metric(1,2) -  
                    P(3,2)*Metric(1,2) -  
                    P(2,1)*Metric(1,3) +  
                    P(2,3)*Metric(1,3) +  
                    P(1,2)*Metric(2,3) -  
                    P(1,3)*Metric(2,3)')
```

**couplings.py:**

```
GC_4 = Coupling(name = 'GC_4',  
                  value = '-G',  
                  order = {'QCD':1})
```

**vertices.py:**

```
V_2 = Vertex(name = 'V_2',  
             particles = [ P.G, P.G, P.G ],  
             color = [ 'f(1,2,3)' ],  
             lorentz = [ L.VVV1 ],  
             couplings = {(0,0):C.GC_4})
```

# ALOHA output

```
SUBROUTINE VVV1_0(V1,V2,V3,C,VERTEX)
IMPLICIT NONE
DOUBLE COMPLEX V1(6)
DOUBLE COMPLEX V2(6)
DOUBLE COMPLEX V3(6)
DOUBLE COMPLEX C
DOUBLE COMPLEX VERTEX
DOUBLE PRECISION P2(0:3),P3(0:3),P1(0:3)
```

```
P2(0) = DBLE(V2(5))
P2(1) = DBLE(V2(6))
P2(2) = DIMAG(V2(6))
P2(3) = DIMAG(V2(5))
P3(0) = DBLE(V3(5))
P3(1) = DBLE(V3(6))
P3(2) = DIMAG(V3(6))
P3(3) = DIMAG(V3(5))
P1(0) = DBLE(V1(5))
P1(1) = DBLE(V1(6))
P1(2) = DIMAG(V1(6))
P1(3) = DIMAG(V1(5))
```

```
VERTEX = C*( (V3(1)*( (V1(1)*( (V2(2)*( (0, -1)*P1(1)+(0, 1)
$ *P3(1))))+( (V2(3)*( (0, -1)*P1(2)+(0, 1)*P3(2))))+(V2(4)*( (0,
$ -1)*P1(3)+(0, 1)*P3(3)))))+( (V2(1)*( (V1(2)*( (0, 1)*P2(1)
$ +(0, -1)*P3(1))))+( (V1(3)*( (0, 1)*P2(2)+(0, -1)*P3(2)))
$ +(V1(4)*( (0, 1)*P2(3)+(0, -1)*P3(3)))))+( (P1(0)*( (0, 1)
$ *(V2(2)*V1(2))+(0, 1)*(V2(3)*V1(3))+(0, 1)*(V2(4)*V1(4)))))
```

```
...
```

Fortran

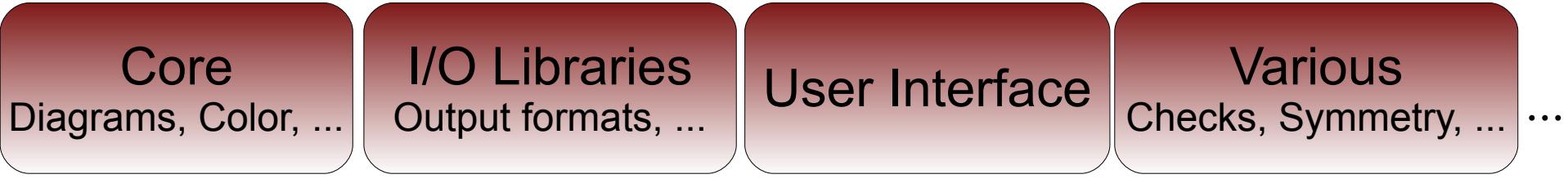
# ALOHA output

```
void VVV1_0(complex<double> V1[], complex<double> V2[], complex<double> V3[],  
complex<double> C, complex<double> & vertex)  
{  
    double P2[4], P3[4], P1[4];  
    P2[0] = V2[4].real();  
    P2[1] = V2[5].real();  
    P2[2] = V2[5].imag();  
    P2[3] = V2[4].imag();  
    P3[0] = V3[4].real();  
    P3[1] = V3[5].real();  
    P3[2] = V3[5].imag();  
    P3[3] = V3[4].imag();  
    P1[0] = V1[4].real();  
    P1[1] = V1[5].real();  
    P1[2] = V1[5].imag();  
    P1[3] = V1[4].imag();  
    vertex = C * ((V3[0] * ((V1[0] * ((V2[1] * (complex<double> (0., -1.) * P1[1]  
        + complex<double> (0., 1.) * P3[1]))) + ((V2[2] * (complex<double> (0.,  
        -1.) * P1[2] + complex<double> (0., 1.) * P3[2]))) + (V2[3] *  
        (complex<double> (0., -1.) * P1[3] + complex<double> (0., 1.) *  
        P3[3])))) + ((V2[0] * ((V1[1] * (complex<double> (0., 1.) * P2[1] +  
            complex<double> (0., -1.) * P3[1]))) + ((V1[2] * (complex<double> (0., 1.)  
            * P2[2] + complex<double> (0., -1.) * P3[2]))) + (V1[3] * (complex<double>  
            (0., 1.) * P2[3] + complex<double> (0., -1.) * P3[3])))) + ((P1[0] *  
            (complex<double> (0., 1.) * (V2[1] * V1[1]) + complex<double> (0., 1.) *  
            (V2[2] * V1[2]) + complex<double> (0., 1.) * (V2[3] * V1[3]))) + (P2[0] *  
            (complex<double> (0., -1.) * (V2[1] * V1[1]) + complex<double> (0., -1.)  
            * (V2[2] * V1[2]) + complex<double> (0., -1.) * (V2[3] * V1[3])))) +  
    ...  
}
```

C++

# Flexibility and modularity

## MadGraph 5



## Models



## ALOHA

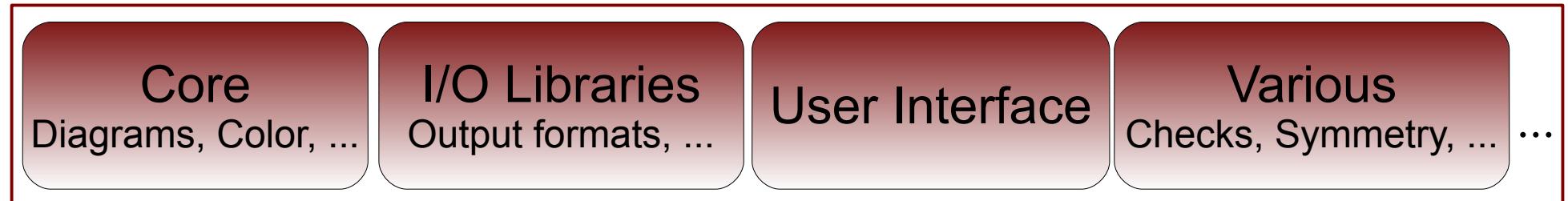


## Tools



# Flexibility and modularity

## MadGraph 5



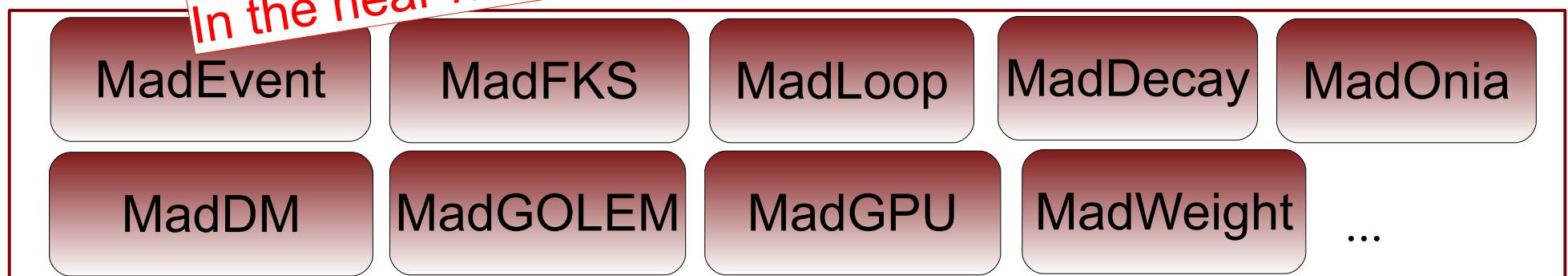
## Models



## ALOHA



## Tools

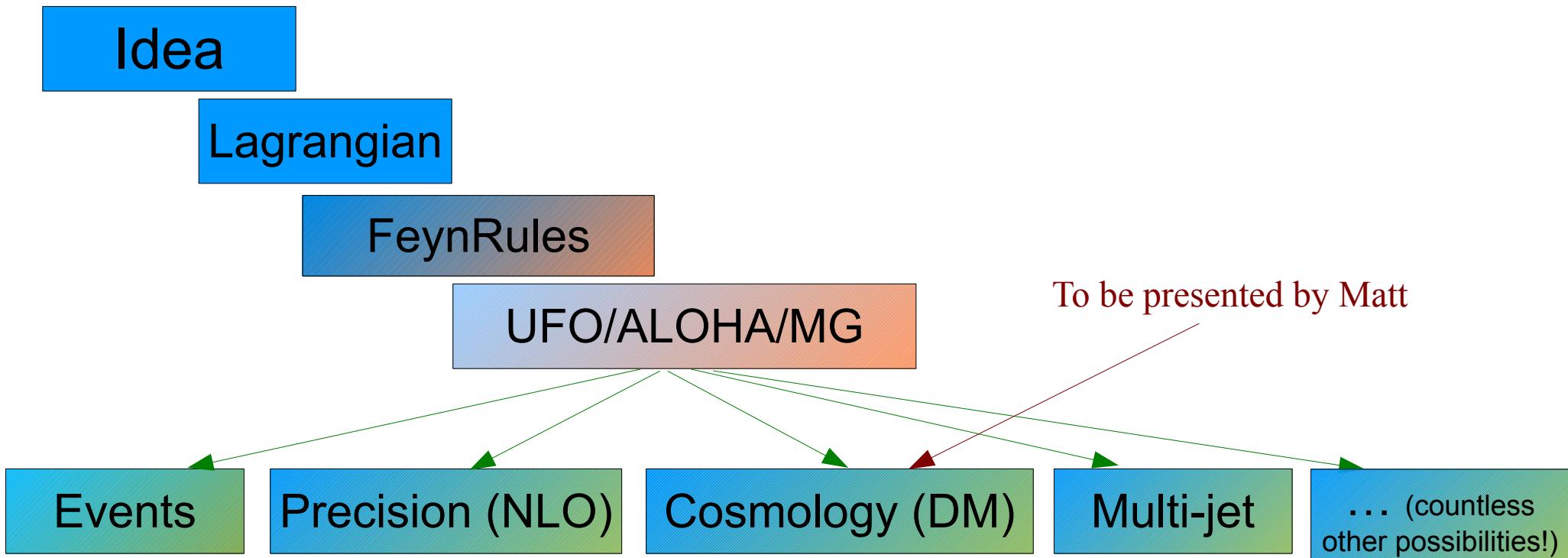


# So the foundation has been laid

... for a new, amazing set of tools

... for astonishing physics applications

... for making the LHC era a success!



# Let's get to work!

THANK YOU FOR COMING!

