Automatic NLO calculations with **GoSam** via **BLHA**

Gionata Luisoni

luisonig@mpp.mpg.de

Max Planck Institute for Physics
Munich

In collaboration with:

G.Cullen, N. Greiner, G.Heinrich, P.Mastrolia, G.Ossola, T.Reiter, F. Tramontano

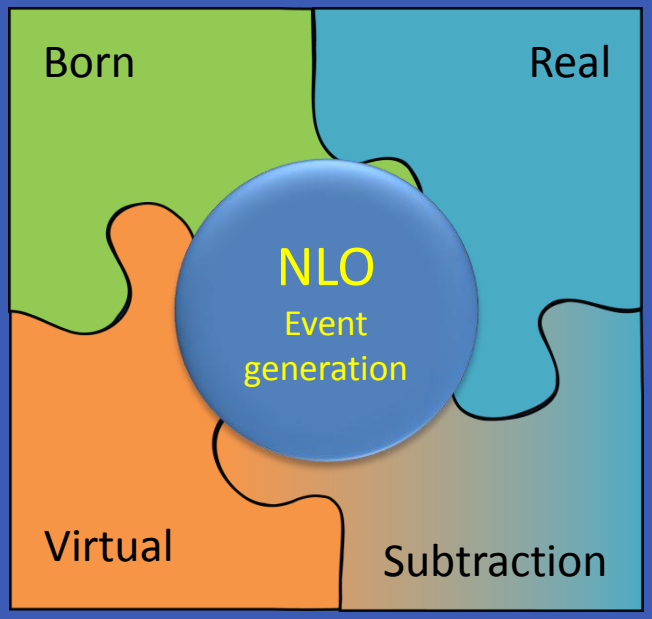
H. Van Deurzen, J.F.G. von Soden-Frauenhofer, E.Mirabella, T.Peraro, J. Reichel, M. Rodgers, J. Schlenk

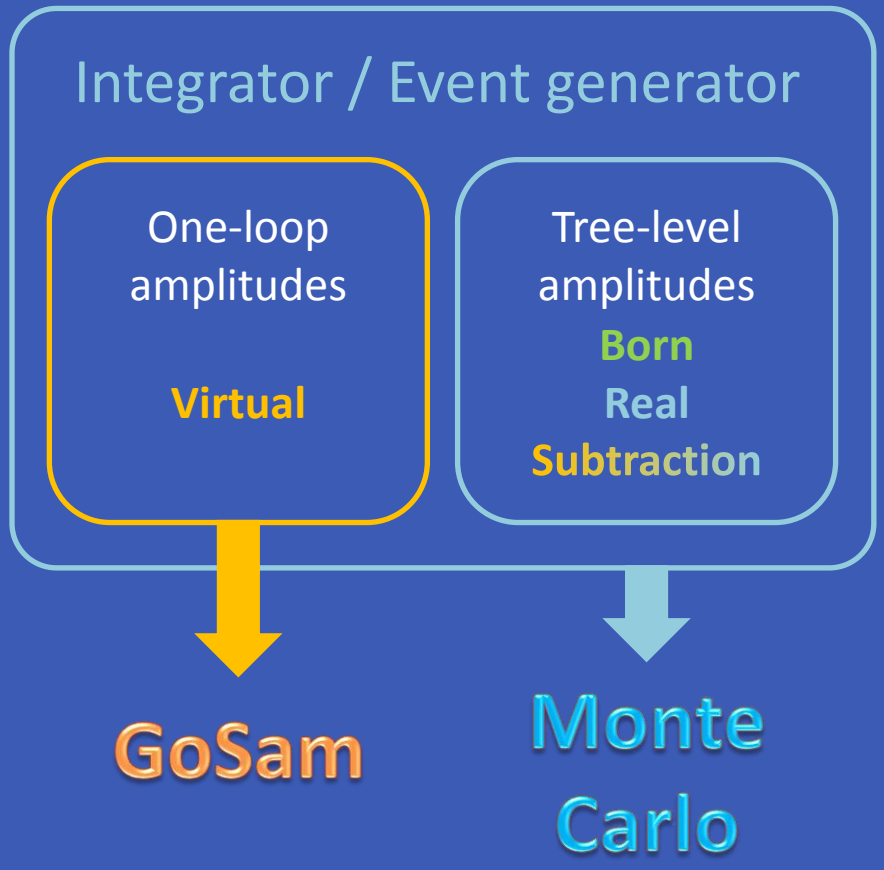
GoSam release: arXiv:1111.2034 [hep-ph] | http://gosam.hepforge.org/

TH/LPCC Institute, CERN Geneva 01.10.2012

NLO Automation (or not?)

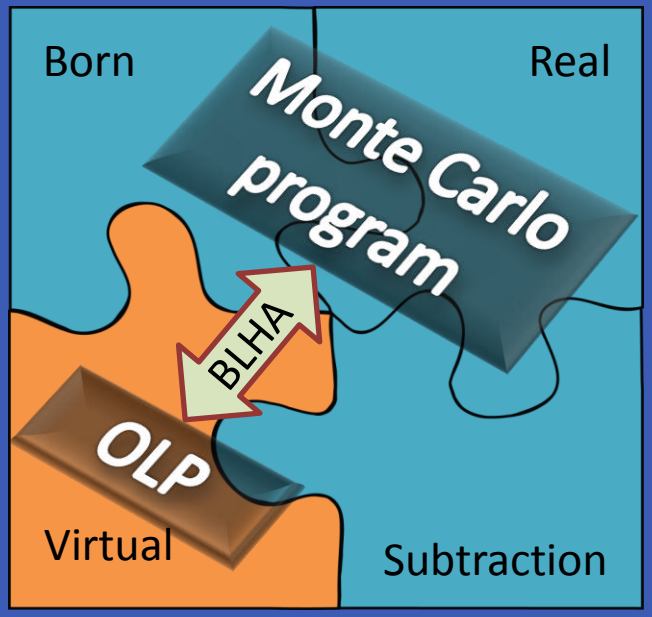
- To claim automation we need to be able to compute automatically the different ingredients of a NLO calculation:





NLO Communication

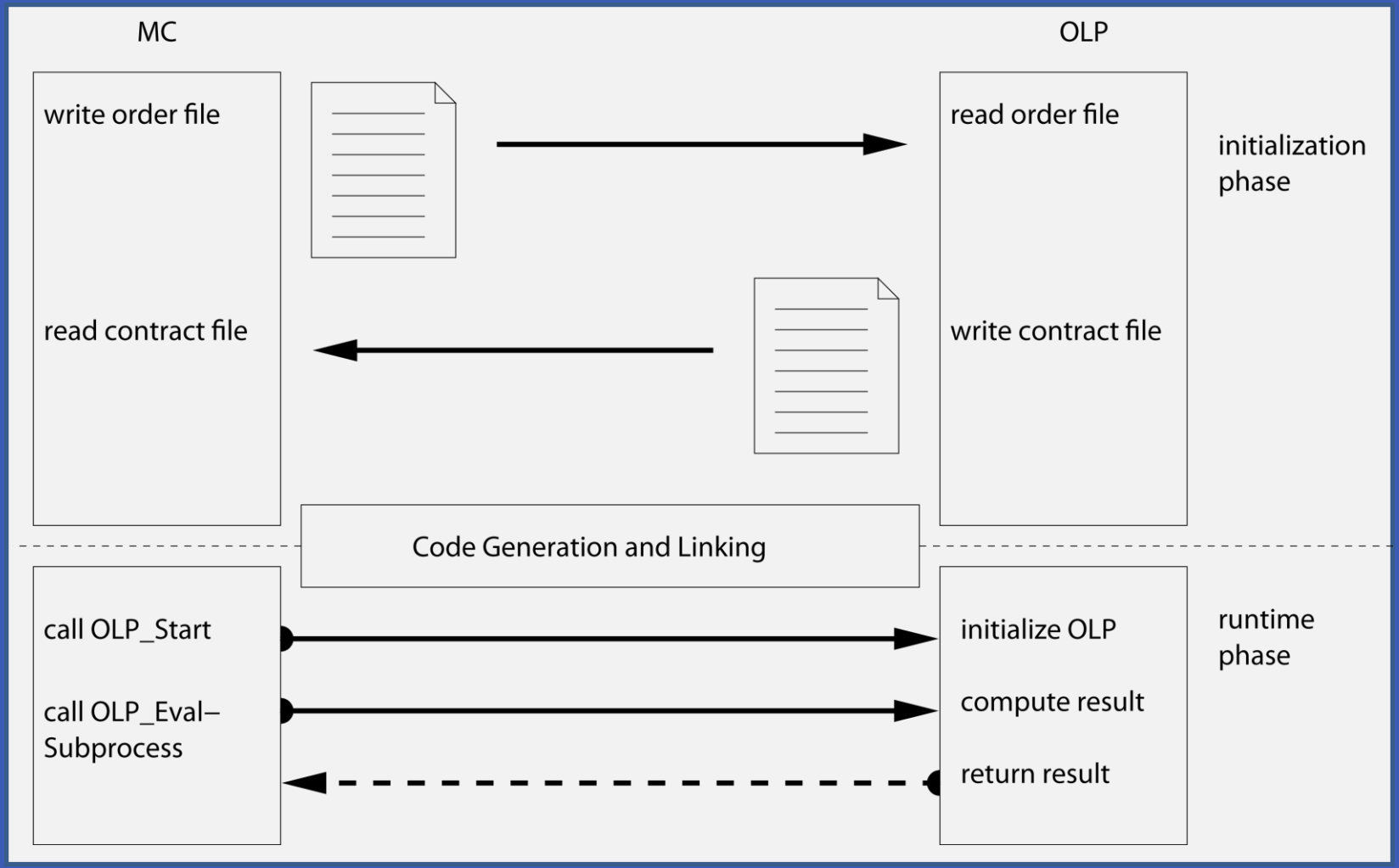
- Automation is important not only in generation and calculation, but also at the level of COMMUNICATION:



[BLHA, arXiv:1001.1307 [hep-ph]]

- Monte Carlo program:
  Born / real corr. / sub. terms
- One-loop Program (OLP):
  virtual corr.
- Pre-runtime comunication via “order“ and “contract“ files
- At runtime:
  - `OLP_Start()`
  - `OLP_EvalSubProcess()`

BLHA-interface: order & contract



GoSam with external MC

- GoSam + POWHEG [G.L., P.Nason, C.Oleari, F. Tramontano]
  - Development phase completed
  - Currently in testing phase
  - Go public soon
- GoSam + SHERPA [G.L., M.Schönherr, F. Tramontano]
  - Possible since Sherpa 1.4.0 (March 2012) : [--enable-lhole]
  - Little additional patch needed for parameter communication
  - Publicly available at: http://gosam.hepforge.org/proc/

→ Both interfaced via the BLHA

The gosamsherpa script

- Automatic generation of codes for QCD NLO corrections to SM processes by simply editing a Sherpa card and executing:

```
$./gosamsherpa MySherpaCard.dat
```

- Restrictions: not fully automated for
  - processes with massive gauge-bosons in the loops ,
  - processes with production of resonant massive top-quarks.

➡ Due to limited communication between MC and OLP

- Temporary solution is given by editing separate GoSam and Sherpa input cards.

Publicly available at:
http://gosam.hepforge.org/proc/

GoSam+Sherpa Process Packages

Available on: http://gosam.hepforge.org/proc/

- Single process packages with pre-generated virtual code available for selected LHC processes:

Process List:

- $pp/p\bar{p} \to W^-(\to e^- + \bar{\nu}_e) + jet$, wm1jet.tar.gz (437K)
- $pp/p\bar{p} \to W^+(\to e^+ + \nu_e) + jet$, wp1jet.tar.gz (431K)
- $pp/p\bar{p} \to Z/\gamma(\to e^- + e^+) + jet$, z1jet.tar.gz (383K)
- $pp/p\bar{p} \to W^-(\to e^- + \bar{\nu}_e) + b\bar{b}$, wmbbjet.tar.gz (772K)
- $pp/p\bar{p} \to W^+(\to e^+ + \nu_e) + b\bar{b}$, wpbbjet.tar.gz (771K)
- $pp/p\bar{p} \to W^-(\to e^- + \bar{\nu}_e) + 2\,jets$, wm2jets.tar.gz (3.49M)
- $pp/p\bar{p} \to W^+(\to e^+ + \nu_e) + 2\,jets$, wp2jets.tar.gz (3.46M)
- $pp/p\bar{p} \to W^+(\to \mu^+ + \nu_\mu) + W^-(\to e^- + \bar{\nu}_e)$, wpwm.tar.gz (716K)
- $pp/p\bar{p} \to W^+(\to \mu^+ + \nu_\mu) + W^+(\to e^+ + \nu_e) + 2\,jets$, wpwp2jets.tar.gz (3.76M)

NEW PROCESSES COMING SOON

- Only 3 steps for NLO:
  - download
  - un-tar package
  - run ‘makecode‘ script
- Script for plots is also attached
- Example of interface with Rivet
- Soon possibility to shower

Stability and rescue system

- Double check to detect unstable point:
  - at the single diagram/diagram-group level during reduction
  - using test on single pole for full amplitude:

$$\varepsilon = \frac{\text{Exact}_{\text{single pole}} - \text{Numerical}_{\text{single pole}}}{\text{Born}\,\frac{\alpha_s}{2\pi}} \leq 10^{-4}$$

    - Study correlation between numerical instabilities and relative importance of virtual corrections
    - Exact single pole from universal IR structure of QCD
    - Points which fail are written to file and can be reprocessed

GoSam+Sherpa vs MCFM: $W^+ + W^-$

Machine: Intel(R) Core(TM)2 Quad CPU Q6600 @ 2.40GHz

Timings:

Generation & Compilation

Virtual: <20 min

Running

Real : ~ 4h 20 min

Virtual: ~ 1h 35 min

MCFM:

ncalls 1: 600'000 pts

ncalls 2: 600'000 pts

itmx 1: 10

itmx 2: 10

time: ~ 3h

Rescue system:

Single Pole Threshold= 0.0001

Bad points: none

NUMBER OF EVENTS:

Born : 5'000'000 x 5

Real : 50'000'000 x 5

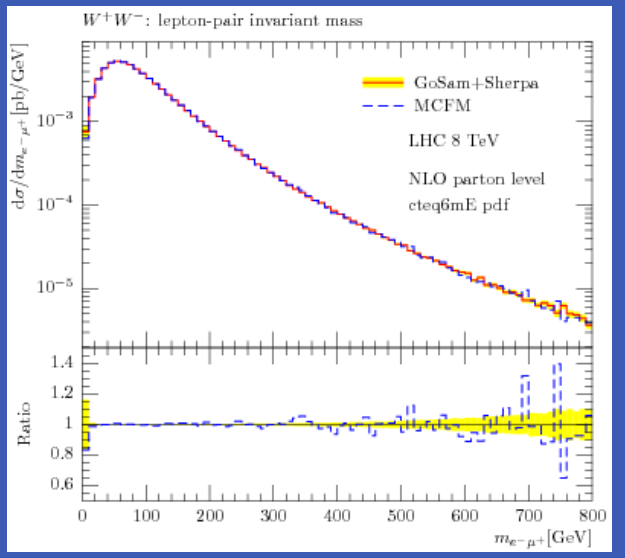
Virtual: 1'000'000 x 5

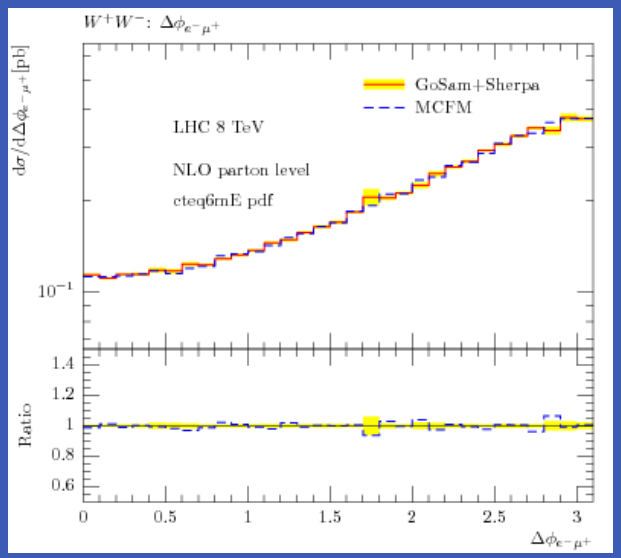
Plots for LHC at 8 TeV

Cuts no cuts in jets

Scale 80 GeV

PDFs cteq6mE.LHgrid





MAX-PLANCK-GESELLSCHAFT

G.Luisoni, 1st October 2012

GoSam+Sherpa vs MCFM: W^- + bb massive

Machine: Intel(R) Core(TM)2 Quad CPU Q6600 @ 2.40GHz

Timings:

Generation & Compilation

Virtual: ~ 22 min

Running

Real : ~ 5h 20 min

Virtual: ~ 11h

MCFM:

ncalls 1: 100'000 pts

ncalls 2: 100'000 pts

itmx 1: 10

itmx 2: 10

time: ~ 7h 10 min

Rescue system:

Single Pole Threshold= 0.0001

Bad points: none

NUMBER OF EVENTS:

Born : 5'000'000 x 5

Real : 50'000'000 x 10

Virtual: 1'000'000 x 10

Plots for LHC at 8 TeV

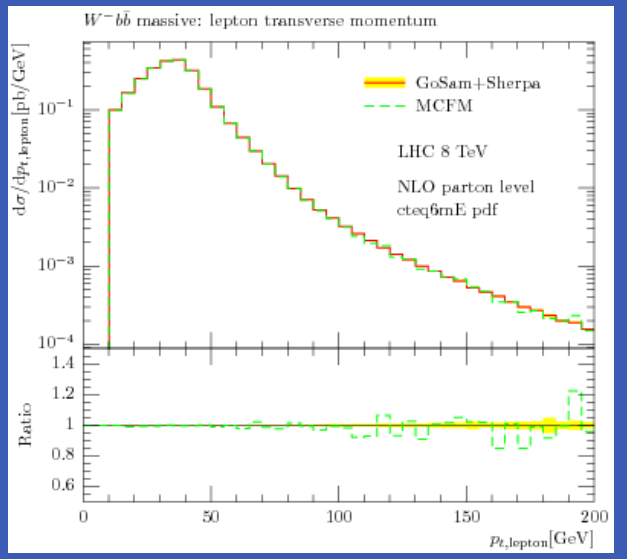
Cuts pt_miss > 20 GeV

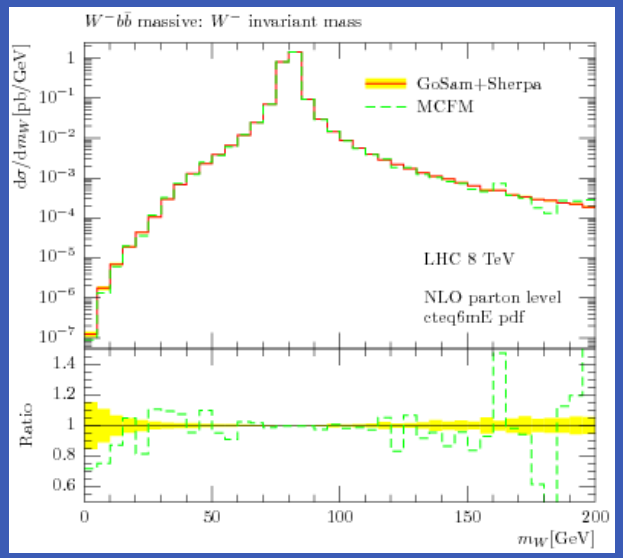
pt_lepton > 10 GeV

inclusive in jets

Scale H_T

PDFs cteq6mE.LHgrid





MAX-PLANCK-GESELLSCHAFT

G.Luisoni, 1st October 2012

GoSam+Sherpa vs MCFM: W^- + 2 jets

Machine: Intel(R) Core(TM)2 Quad CPU Q6600 @ 2.40GHz

Timings:

Generation & Compilation

Virtual: ~ 1d 5h 45min

Running

Real : ~ 17h

Virtual: ~ 15h 30 min

MCFM:

ncalls 1: 600'000 pts

ncalls 2: 600'000 pts

itmx 1: 10

itmx 2: 10

time: ~ 6d 21h

Rescue system:

Single Pole Threshold= 0.0001

Bad points: 6 pts.

NUMBER OF EVENTS:

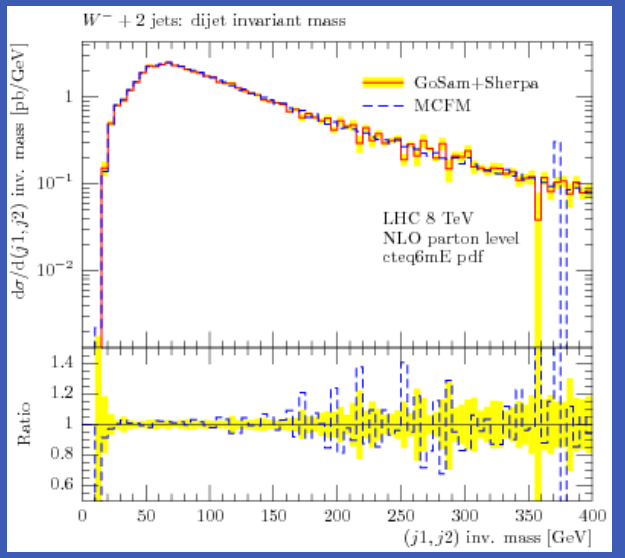
Born : 5'000'000 x 5

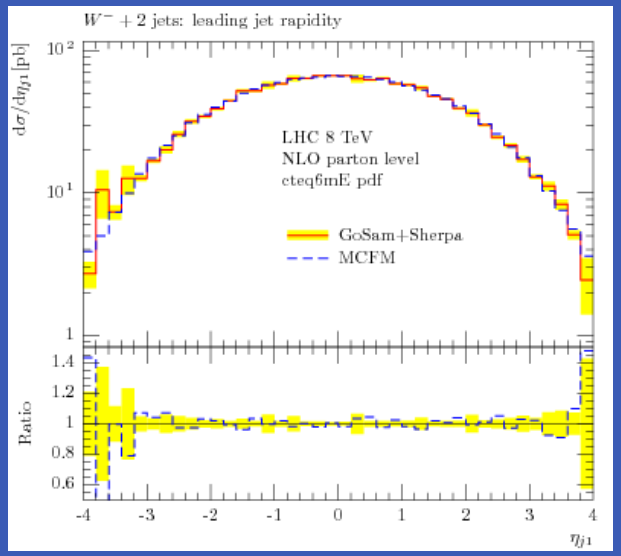
Real : 50'000'000 x 10

Virtual: 1'000'000 x 10

Plots for LHC at 8 TeV

Cuts	pt_jet > 20 GeV
	eta_jet < 4.0
	kt_alg, R=0.7
Scale	H_T
PDFs	cteq6mE.LHgrid





GoSam+Sherpa vs Melia et al.: W^+W^+ + 2 jets

Machine: Intel(R) Core(TM)2 Quad CPU Q6600 @ 2.40GHz

Timings:

Generation & Compilation

Virtual: ~ 5h 45 min

Running

Real : ~ 14h 15 min

Virtual: ~ 14h 40 min

NUMBER OF EVENTS:

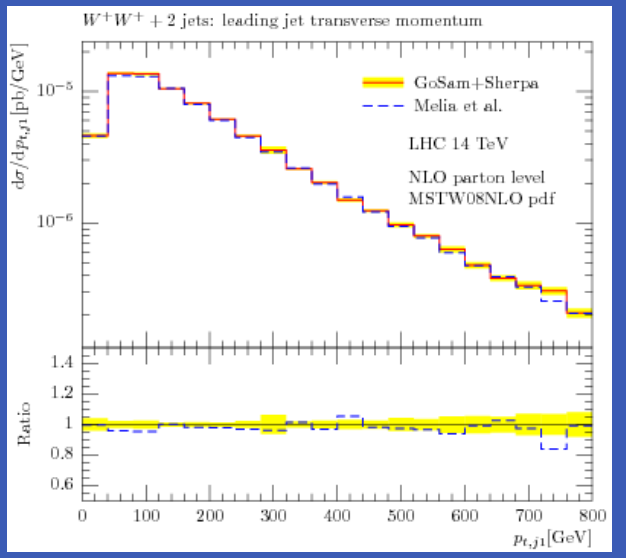
Born : 1'000'000 x 5

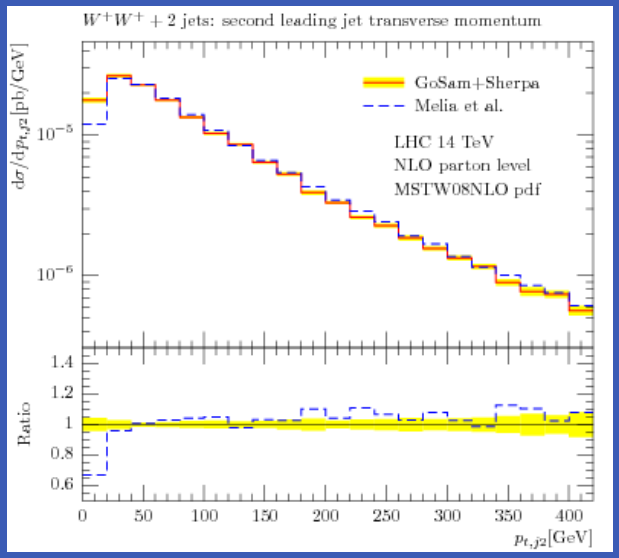
Real : 50'000'000 x 5

Virtual: 1'000'000 x 5

Comparison with:
Melia, Melnikov,
Roentsch, Zanderighi;
JHEP 1012 (2010) 053;
[arXiv:1007.5313]

Rescue system:
Single Pole Threshold= 0.0001
Bad points: 1062 pts in grid
Bad points: 1409 pts in run





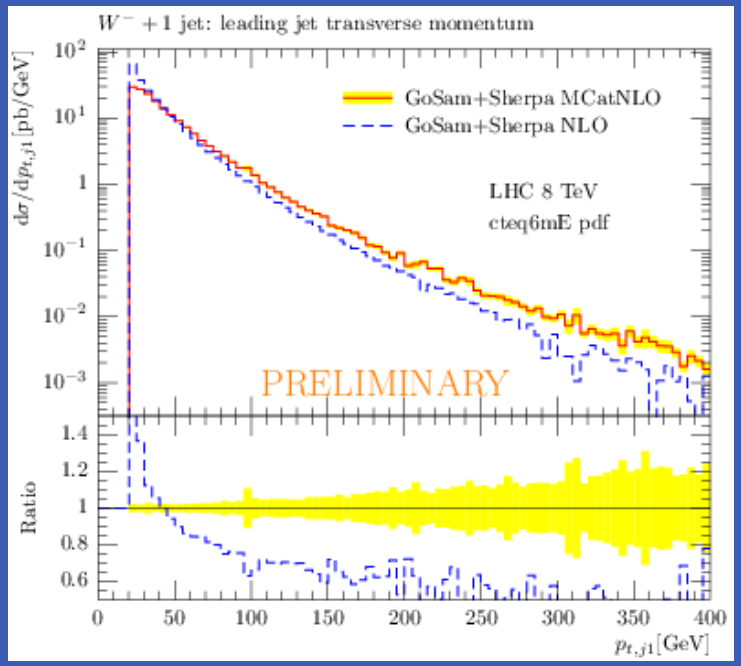
Plots for LHC at 14 TeV

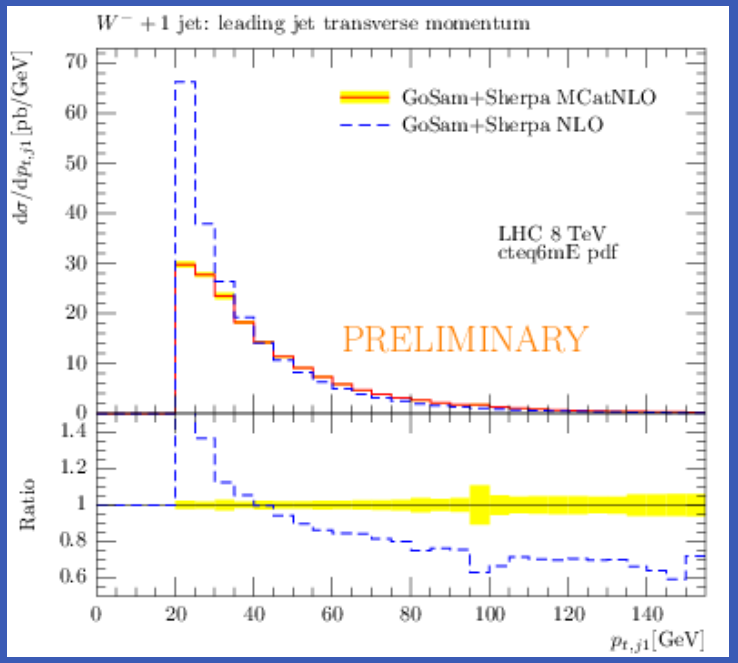
Cuts	pt_lep > 20 GeV
	|eta_lep| < 2.4
	pt_miss > 30 GeV
	antikt_alg, R=0.4
Scale	150 GeV
PDF	MSTW2008nlo.LHgrid

MAX-PLANCK-GESELLSCHAFT

GoSam+Sherpa with shower: W^- + 1 jets

- First preliminary results for GoSam+Sherpa with MCatNLO shower:





Plots for LHC at 8 TeV

Cuts	pt_jet > 20 GeV
	eta_jet < 4.0
	kt_alg, R=0.7
Scale	H_T
PDFs	cteq6mE.LHgrid

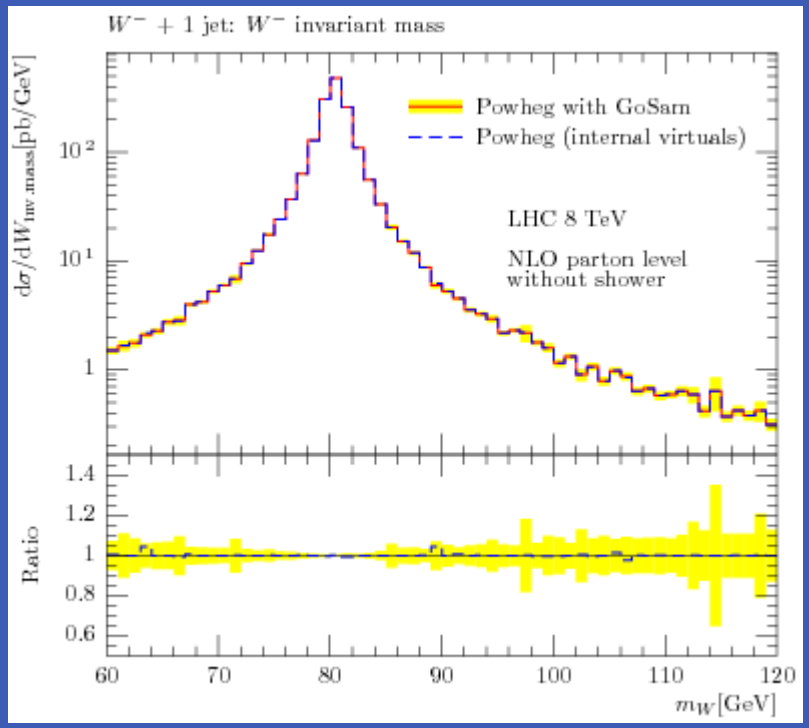
NUMBER OF EVENTS:

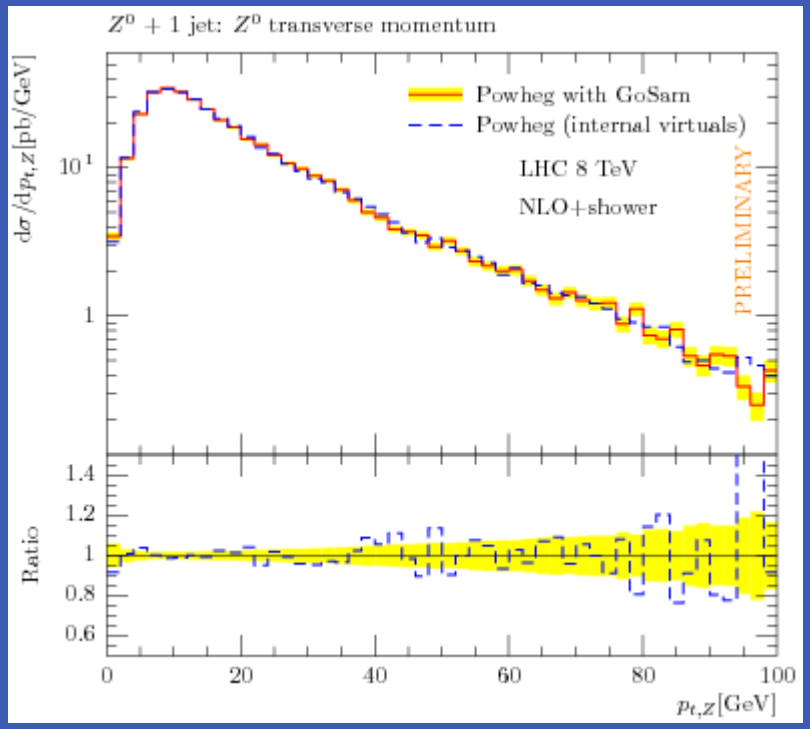
NLO : 40'000'000

NLO+SHOWER : 5'000'000

GoSam+Powheg Box

- Test examples against existing processes in the Powheg Box both at NLO and NLO+Shower:





Possible BLHA extensions

- Transfer parameters values (masses,width):
  - Statical parameters (fixed during calculation)
  - Dynamical parameters (PS-point dependent)
    - `OLP_Parameter(string, double)`
- Diagram selection in common with MC
- Loop-induced processes
  - e.g. gg -> ZZ -> 4 leptons / Higgs in gluon-fusion

Conclusions

http://gosam.hepforge.org/

- BLHA interface successfully used in GoSam to interface with:
  Powheg Sherpa
  - Interface allows for fully automatic generation of QCD NLO codes:
    gosamsherpa script
- Double rescue system at diagram- and amplitude-level successfully cures most of the unstabilities
- Extensions of BLHA to support more general parameter transfer and allow for further automation

Conclusions

http://gosam.hepforge.org/

NLO automation (or NOT?)

Many other effords and great achievements:

Blackhat / HELAC-NLO / Looptools / Madloop / Ngluon / Numerical integration / Openloops / Rocket / …

➡ We are definitely on the right track!

BACKUP SLIDES

The GoSam Project: the codes

GoSam Project

GoSam: Python package to write code (fortran95)

Code generation

- Diagram generation:
  - QGRAF [Nogueira 92]
- Algebra:
  - FORM [Vermaseren 91]
  - SPINNEY [Cullen, Koch-Janusz, Reiter 10]
- Code generator:
  - HAGGIES [Reiter 09]

Yellow codes distributed separately

Generated code execution

- Loop integral reduction:
  - SAMURAI [Mastrolia, Ossola, Reiter, Tramontano 10]
  - GOLEM95 [Binoth, Cullen, Guillet, Heinrich, Pilon, Reiter 08]
- Scalar integral evaluation:
  - AVHOLO [van Hameren]
  - QCDLOOP [Ellis, Zanderighi]
  - GOLEM95C [Cullen, Guillet, Heinrich, Kleinschmidt, Pilon, Reiter, Rodgers 11]

All codes in gosam-contrib package

Reduction methods

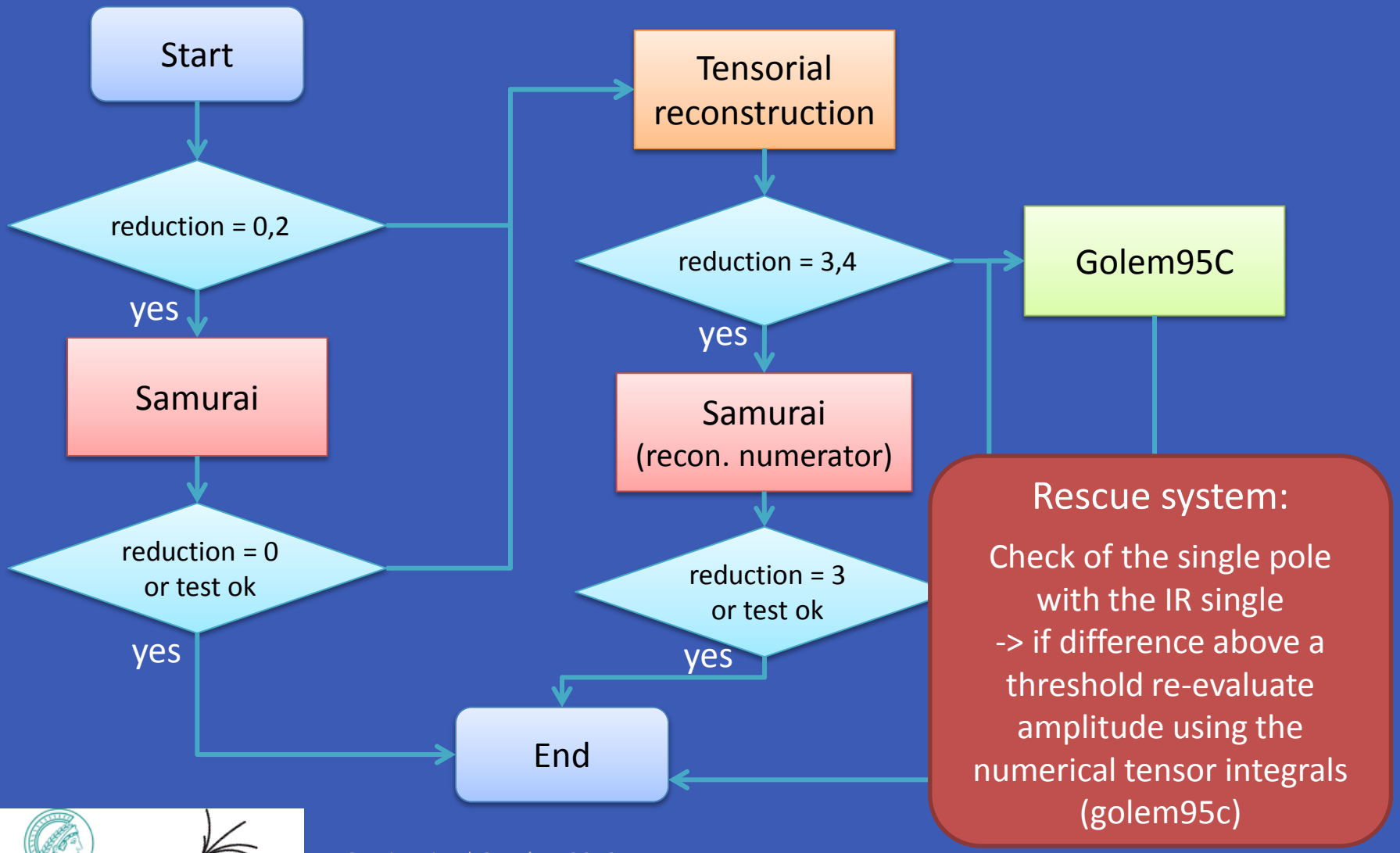
- SAMURAI
  [Mastrolia, Ossola, Reiter, Tramontano 10]

Reduction method can be choosen at runtime

- Tensorial integrand-level reconstruction
  [Heinrich, Ossola, Reiter, Tramontano 10]

  with
  - GOLEM95C [Binoth, Cullen, Guillet, Heinrich, Kleinschmidt, Pilon, Reiter, Rodgers 11]
  - SAMURAI [Mastrolia, Ossola, Reiter, Tramontano 10]
  - PJFry [Yundin]

Reduction: strategies



In practice: GoSam+ Sherpa

[In collaboration with M.Schönherr]

- Few steps needed to compute e.g. Z+1 jet @NLO:
  - Prepare Sherpa card according to your need and run it once
    - The "order" file and the necessary tree-level code is generated
  - Run GoSam feeding the "order" file and a configuration file with further needed inputs (paths / filtering options / …)
  - After the virtual code is set up, generate and compile it with **`configure / make / make install`**
  - The produced library **`libgolem_olp.so`** must be added to the **`SHERPA_LDADD`** option in the Sherpa card

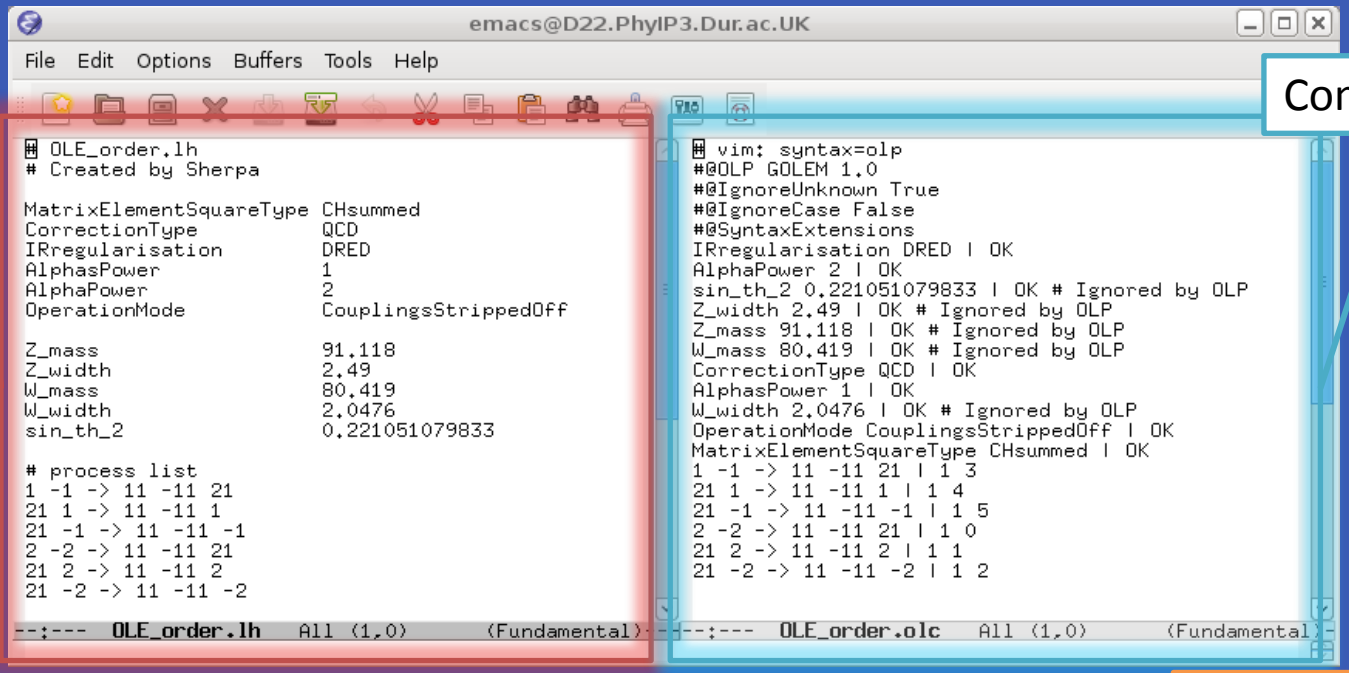
➧ HAVE FUN WITH PHENOMENOLOGY

High level of automation and optimization in the generated code

In practice: GoSam+ Sherpa

Order file

Contract file



GoSam produces only the code strictly needed avoiding redundancies and exploiting crossing-symmetry