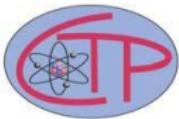


# GoSAM: A PROGRAM FOR AUTOMATED ONE-LOOP CALCULATIONS

Giovanni Ossola

New York City College of Technology  
City University of New York (CUNY)



In collaboration with:

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

**“LoopFest XI” Radiative Corrections for the LHC and beyond**  
**University of Pittsburgh – May 10-12, 2012**

# CREDITS

Recent work:

- “Automated One-Loop Calculations with GoSam,” G. Cullen, N. Greiner, G. Heinrich, G. Luisoni, P. Mastrolia, G.O., T. Reiter, F. Tramontano, [arXiv:1111.2034](#).
- “NLO QCD corrections to the production of  $W^+ W^-$  plus two jets at the LHC,” N. Greiner, G. Heinrich, P. Mastrolia, G.O., T. Reiter, F. Tramontano, [arXiv:1202.6004](#) .

Related work in **talk of Edoardo Mirabella** (on Thursday):

- “On the Integrand-Reduction Method for Two-Loop Scattering Amplitudes,” P. Mastrolia, G.O., [arXiv:1107.604](#).
- “Integrand reduction of one-loop scattering amplitudes through Laurent series expansion”, P. Mastrolia, E. Mirabella, T. Peraro, [arXiv:1203.0291](#).

# GoSAM @ MPI, MUNICH – JANUARY 2012



J. F. Soden-Fraunhofen (MPI), P. Mastrolia (MPI), T. Reiter (MPI), N. Greiner (MPI), G. Heinrich (MPI), F. Tramontano (CERN), G. O. (CUNY), E. Mirabella (MPI), G. Luisoni (IPPP), G. Cullen (DESY)  
not in the picture: J. Schlenk, T. Peraro, H. van Deurzen (MPI)

# WHEN WISHLISTS BECOME REALITY

## Progress on $2 \rightarrow 4(5)$

$pp \rightarrow W + 3 \text{ jets}$  Rocket (2009), Blackhat (2009)

$pp \rightarrow t\bar{t}bb$  Bredenstein *et al.* (2009), Helac-NLO (2009)

$pp \rightarrow Z(\gamma) + 3 \text{ jets}$  Blackhat (2010)

$pp \rightarrow t\bar{t}jj$  Helac-NLO (2010)

$pp \rightarrow W^+W^-bb$  Denner *et al.* (2010), Helac-NLO (2010)

$pp \rightarrow W^+W^+jj$  Rocket (2010)

$pp \rightarrow W(Z) + 4 \text{ jets}$  Blackhat (2011)

$pp \rightarrow b\bar{b}bb$  Golem/Samurai (2011)

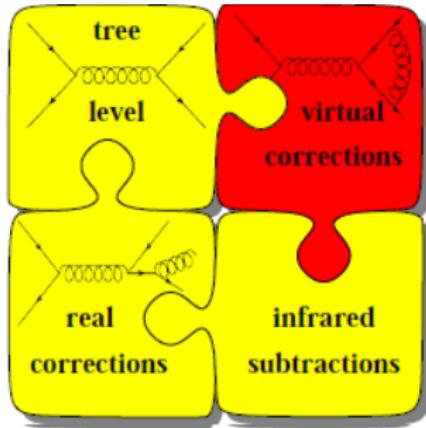
$pp \rightarrow 4 \text{ jets}$  Blackhat (2011)

$pp \rightarrow W\gamma\gamma \text{ jet}$  Campanario *et al.* (2011)

$pp \rightarrow W^+W^- + 2 \text{ jets}$  Rocket (2011), GoSam (2012)

**Virtual + Real + Subtraction Terms + Integration...**

# NLO CALCULATIONS



- LO tree-level  $2 \rightarrow N$
- NLO virtual  $2 \rightarrow N$
- NLO real  $2 \rightarrow N + 1$
- Subtraction terms

**GoSam** provides the **virtual corrections**

Tree-level, real emission, subtraction terms, integration:

- MadGraph+MadDipole+MadEvent ( $pp \rightarrow WWjj$  and  $pp \rightarrow 4b$ )
- Interface with Sherpa/Powheg via BLHA

# GoSAM: AUTOMATED VIRTUAL ONE-LOOP

Target: **Automated Tool** for the evaluation of One-Loop Virtual Amplitudes

- **be general/model independent** (QCD, EW, BSM, ...)
- **interface with other existing tools** (Sherpa, Powheg, ...)
- build upon **open source tools**

# GoSAM: AUTOMATED VIRTUAL ONE-LOOP

Target: **Automated Tool** for the evaluation of One-Loop Virtual Amplitudes

- **be general/model independent** (QCD, EW, BSM, ...)
- **interface with other existing tools** (Sherpa, Powheg, ...)
- build upon **open source tools**

**Algebraic approach** to Automation

- Generate unintegrated amplitudes with **Feynman diagrams**
- Manipulate and simplify them
- Perform the reduction (**integrand-level** or tensorial)

# GoSAM: AUTOMATED VIRTUAL ONE-LOOP

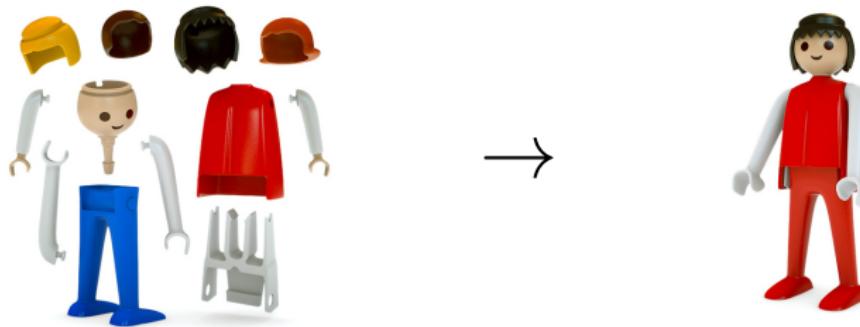
Target: **Automated Tool** for the evaluation of One-Loop Virtual Amplitudes

- **be general/model independent** (QCD, EW, BSM, ...)
- **interface with other existing tools** (Sherpa, Powheg, ...)
- build upon **open source tools**

**Algebraic approach** to Automation

- Generate unintegrated amplitudes with **Feynman diagrams**
- Manipulate and simplify them
- Perform the reduction (**integrand-level** or tensorial)

**Modular Structure** → Flexibility to **adapt** and **improve** the code



◇ **Diagram Generation** → Automated, based on Feynman diagrams

- FORM [Vermaseren](#)
- QGRAF [Nogueira](#)
- Haggies [Reiter](#)
- Spinney [Cullen, Koch-Janusz, Reiter](#)

◇ **Diagram Reduction** → Several Options available at runtime

- ▶ Default Option: **Samurai** [Mastrolia, G.O., Reiter, Tramontano](#)
- OR **Golem95** [Binoth, Cullen, Guillet, Heinrich, Kleinschmidt, et al.](#)
- OR **Tensorial Reconstruction** [Heinrich, G.O., Reiter, Tramontano](#)

◇ **Scalar (Master) Integrals**

- ▶ Default Option: **OneLOop** [A. van Hameren](#)
- OR **QCDloop** [Ellis, Zanderighi](#)
- OR **Golem95C** [Cullen, Guillet, Heinrich, Kleinschmidt, Pilon, et al.](#)
- OR New interface: **PJFry** [Yundin](#), **LoopTools** [T. Hahn](#)

# I. Diagram Generation

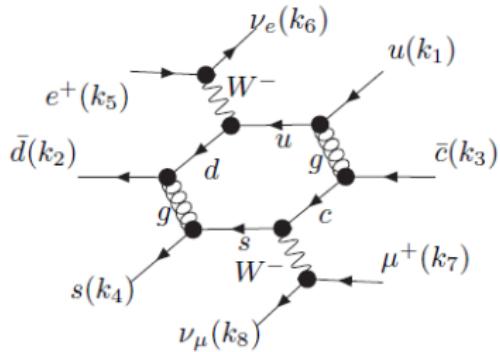


Diagram 1

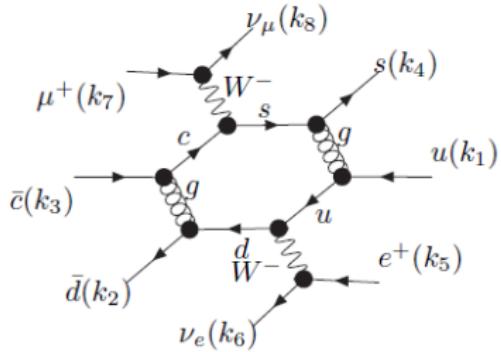
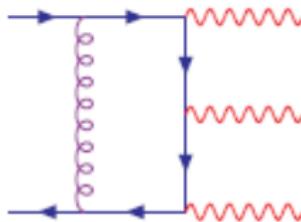


Diagram 2

# ONE-LOOP – DEFINITIONS



Any  $m$ -point one-loop amplitude can be written, before integration, as

$$A(\bar{q}) = \frac{N(\bar{q})}{\bar{D}_0 \bar{D}_1 \cdots \bar{D}_{m-1}}$$

where

$$\bar{D}_i = (\bar{q} + p_i)^2 - m_i^2 \quad , \quad \bar{q}^2 = q^2 - \mu^2 \quad , \quad \bar{D}_i = D_i - \mu^2$$

Our task is to calculate, for each phase space point:

$$\mathcal{M} = \int d^d \bar{q} \ A(\bar{q}) = \int d^d \bar{q} \frac{N(\bar{q})}{\bar{D}_0 \bar{D}_1 \cdots \bar{D}_{m-1}}$$

# DIAGRAM GENERATION: QGRAF AND PYTHON

- ▶ Tree-level and one-loop integrands are generated by **QGRAF** supplemented by **Python**
- ▶ Three sets of output (for each diagram)
  - algebraic expression for the diagram
  - code for drawing the diagram (**LATEX** file)
  - code for computing the properties of the diagram
- ▶ Model files: built-in (EW, QCD) or generated by the user via LanHEP, FeynRules
- ▶ The **Python** program automatically performs several operations
  - diagrams who turn out to be zero are dropped (color, vanishing loop integrals)
  - number of propagators, **tensor rank** and kinematic invariants are computed
  - flag diagrams with similar structures (i.e **sets of denominators**)

**Crucial information to optimize the integrand-level reduction  
and allow for a gain in efficiency**

## DIAGRAM GENERATION: haggies AND spinney

- ◊ The amplitudes, generated in terms of algebraic expressions, are **processed with a FORM program**, using spinney for the **spinor algebra**.
  - External vectors are 4-dimensional, internal momenta are d-dimensional

$$\bar{D}_i = (\bar{q} + p_i)^2 - m_i^2 \quad , \quad \bar{q}^2 = q^2 - \mu^2 \quad , \quad \bar{D}_i = D_i - \mu^2$$

- ◊ We separate terms containing  $\mu^2$  and dot products involving the integration momentum from all other factors.
- ◊ All subexpressions which do not depend on either  $q$  or  $\mu^2$  are substituted by abbreviations, which are evaluated only once per phase space point.
- ◊ Each of the two parts is then processed by haggies (java) which generates optimized Fortran95 code for the numerical evaluation

The unintegrated amplitudes (written as Fortran95 code)  
are now ready for the reduction

# AUTOMATION AT ONE-LOOP: “ALGEBRAIC WAY”

Main features of the “Algebraic Way”:

- Amplitudes generated with **Feynman diagrams**
- **Algebraic manipulations are allowed** before starting the numerical integration
- The **generation** of numerators is executed **separately from the numerical reduction**
- Optimization: **grouping of diagrams**, smart **caching**
- Control over sub-parts of the computation (move in/out subsets of diagrams)
- Algebra in dimension  $d$ , different schemes

**Great flexibility in the reduction**

**Choice between different reduction algorithms at runtime**

## II. Diagram Reduction

$$\mathcal{A} = c_4 \text{ (square diagram)} + c_3 \text{ (triangle diagram)} + c_2 \text{ (line segment)} + c_1 \text{ (circle)} + \mathcal{R}$$

# DIAGRAM REDUCTION: SAMURAI

Default Option: Diagram Generation + **Samurai** + OneLoop

**Algebraic** automated generation of **d-dimensional integrands** via **Feynman diagrams**

Reduction at the Integrand Level: **d-dimensional extension of integrand level reduction**



## SAMURAI Mastrolia, G.O., Reiter, Tramontano

- OPP Reduction Algorithm      G.O., Papadopoulos, Pittau
- d-dimensional extension      Ellis, Giele, Kunszt, Melnikov
- Coefficients of Polynomials via DFT      Mastrolia, G.O., Papadopoulos, Pittau
- Model-independent Computation of the full Rational Term

# INTEGRAND-LEVEL APPROACH

Integrand level decomposition:

$$\begin{aligned} N(\bar{q}) &= \sum_{i << m}^{n-1} \Delta_{ijklm}(\bar{q}) \prod_{h \neq i,j,k,\ell,m}^{n-1} \bar{D}_h + \sum_{i << \ell}^{n-1} \Delta_{ijkl}(\bar{q}) \prod_{h \neq i,j,k,\ell}^{n-1} \bar{D}_h + \\ &+ \sum_{i << k}^{n-1} \Delta_{ijk}(\bar{q}) \prod_{h \neq i,j,k}^{n-1} \bar{D}_h + \sum_{i < j}^{n-1} \Delta_{ij}(\bar{q}) \prod_{h \neq i,j}^{n-1} \bar{D}_h + \sum_i^{n-1} \Delta_i(\bar{q}) \prod_{h \neq i}^{n-1} \bar{D}_h \end{aligned}$$

- both sides are polynomial in  $q$  and  $\mu^2$
- the functional form is **process-independent**
- the process-dependent coefficients are contained in the  $\Delta$ 's
- **polynomial fitting replaces the integration**
- bonus: we can solve “on the multi-cuts”

G. O., Papadopoulos and Pittau (2007)  
Ellis, Giele, Kunszt, Melnikov (2008), Melnikov, Schulze (2010)  
Mastrolia, G.O., Reiter, Tramontano (2010)

# RATIONAL TERMS

$$\mathcal{A} = c_4 \text{ (square loop)} + c_3 \text{ (triangle loop)} + c_2 \text{ (double line)} + c_1 \text{ (single line)} + \mathcal{R}$$

**Rational Terms:**  $\mathcal{R} = \mathcal{R}_1 + \mathcal{R}_2$

$\mathcal{R}_1$  originates from the reconstruction of  $d$ -dimensional denominators  $\bar{D}_i$ .

$\mathcal{R}_2$  originates from the  $d$ -dimensional part of the numerator function  $\bar{N}(\bar{q})$

$$\bar{N}(\bar{q}) = N(q) + \tilde{N}(\tilde{q}^2, q, \epsilon)$$

In the original OPP approach,  $\mathcal{R}_2$  is computed using an ad-hoc set of  
**tree-level like Feynman Rules**

G.O., Papadopoulos, Pittau (2007)  
Draggiotis, Garzelli, Malamos, Pittau (2009-2011)

## GoSAM AND RATIONAL TERMS $\mathbf{R}_2$

$\mathbf{R}_1$  is “automatic” in the **d-dim** integrand reduction

GoSam offers different options for calculation of  $\mathbf{R}_2$

Thanks to the fact that we generate analytic expressions for the *d*-dimensional numerator function  $\bar{N}(\bar{q})$

- ▷ **implicit**:  $\mathbf{R}_2$  terms are kept in the numerator and reduced at runtime using the *d*-dimensional decomposition of the numerator
- ▷ **explicit**:  $\mathbf{R}_2$  terms are calculated analytically (without entering in the numerical decomposition)
- ▷ **only**: only the  $\mathbf{R}_2$  term is kept in the final result  
(this option does not require any additional libraries)
- ▷ **off**: all  $\mathbf{R}_2$  terms are set to zero

# INPUT CARD

**Preparation of the “card”:** we use as example  $u\bar{d} \rightarrow \bar{s}c e^- \bar{\nu}_e \mu^+ \nu_\mu$

**in**= u,d~

**out**= nmu, mu+, e-, ne~, s~, c

**model**=smdiag

**order**=gw,4,4; order=gs,2,4

**zero**=mB,mC,mS,mU,mD,me,mmu

**one**=gs,e

**helicities**=-+-+-+-

**extensions**=samurai, dred

# INPUT CARD → TEST ON PS-POINT

**Preparation of the “card”:** we use as example  $u\bar{d} \rightarrow \bar{s}c e^- \bar{\nu}_e \mu^+ \nu_\mu$

```
in= u,d~  
out= nmu, mu+, e-, ne~, s~, c  
model=smdiag  
order=gw,4,4; order=gs,2,4  
zero=mB,mC,mS,mU,mD,me,mmu  
one=gs,e  
helicities=-+-+-+-  
extensions=samurai, dred
```

## RESULT

```
NLO/LO, finite part -15.91575134226371  
NLO/LO, single pole 7.587050691447690  
NLO/LO, double pole -5.333333333333456
```

timing (ms) = 5.3999999999999995

# PRECISION TESTS / RESCUE SYSTEM

Precision Tests should be available **during runtime**

- ▶ Reconstruction Level → Use the decomposition of the numerator function  $N(\bar{q})$  after determining all coefficients
  - 1  $N = N$  test (global or local)
  - 2 Power-test
- ▶ Amplitude Level → **Check IR poles** after UV renormalization

Are those methods **reliable** in detecting **unstable phase space points**?

If so, what did we do with discarded points?



- Reprocess with different method → tensorial reduction
- Go to higher precision: from \*8 → \*10 → \*16
- Discard

# CALCULATIONS TESTED WITH GoSAM

Time required for **code generation** and calculation of **one phase-space point**

Intel(R) Core(TM) i7 CPU 950 @ 3.07GHz

Process	Generation [s]	Evaluation [ms]
$bg \rightarrow Hb$	236	2.49
$d\bar{d} \rightarrow t\bar{t}$	341	4.71
$dg \rightarrow dg$	398	3.08
$e^+e^- \rightarrow t\bar{t}$	221	1.27
$gg \rightarrow gg$	525	1.69
$gg \rightarrow gZ$	529	15.18
$gg \rightarrow t\bar{t}$	1132	24.65
$gb \rightarrow e^-\bar{\nu}_e t$	337	2.89
$\bar{u}d \rightarrow W^+W^+\bar{c}s$	1295	17.37
$u\bar{d} \rightarrow Wb\bar{b}$		7
$u\bar{u} \rightarrow H t\bar{t}$		44
$gg \rightarrow H t\bar{t}$		223
$u\bar{u} \rightarrow t\bar{t} b\bar{b}$		393
$gg \rightarrow t\bar{t} b\bar{b}$		$\sim 10$ sec
$u\bar{d} \rightarrow W^+ ggg$		$\sim 2.5$ sec

$$pp \rightarrow W^+ W^- jj$$

Melia, Melnikov, Rontsch, Zanderighi (2011)

Greiner, Heinrich, Mastrolia, G.O., Reiter, Tramontano (2012)

Eight basic partonic sub-processes  
(doubly resonant)

$$d \bar{u} \rightarrow c \bar{s} W^+ W^-$$

$$u \bar{u} \rightarrow c \bar{c} W^+ W^-$$

$$d \bar{d} \rightarrow s \bar{s} W^+ W^-$$

$$u \bar{u} \rightarrow d \bar{d} W^+ W^-$$

$$u \bar{u} \rightarrow u \bar{u} W^+ W^-$$

$$d \bar{d} \rightarrow d \bar{d} W^+ W^-$$

$$u \bar{u} \rightarrow g g W^+ W^-$$

$$d \bar{d} \rightarrow g g W^+ W^-$$

Tree-level + real emission: **MadGraph**

Subtraction terms: **MadDipole**

Phase space integration: **MadEvent**

Virtual Part: **GoSam**

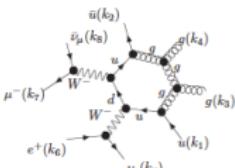
Samples of tree-level unweighted events  
+ **reweighting** to obtain the virtual  
corrections.

Done independently for each subprocess

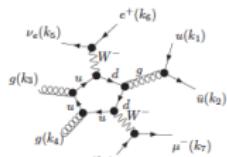
$$pp \rightarrow W^+ W^- jj$$

In the calculation of **virtual corrections**, the Feynman diagrams have been divided into three groups (called **A**, **B**, and **C**).

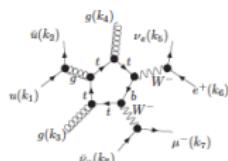
Part A:



Part B:



Part C:



**Part A:** Only **first two two generations** and **no gauge vector bosons** attached to the loop

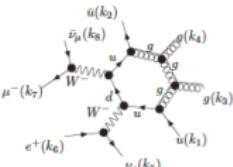
**Part B:** Only **first two generations** and only diagrams **with gauge vector bosons** attached to the loop

**Part C:** **Third generation only**, all diagrams

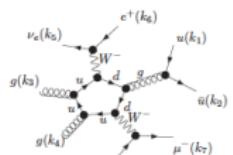
$$pp \rightarrow W^+ W^- jj$$

In the calculation of **virtual corrections**, the Feynman diagrams have been divided into three groups (called **A**, **B**, and **C**).

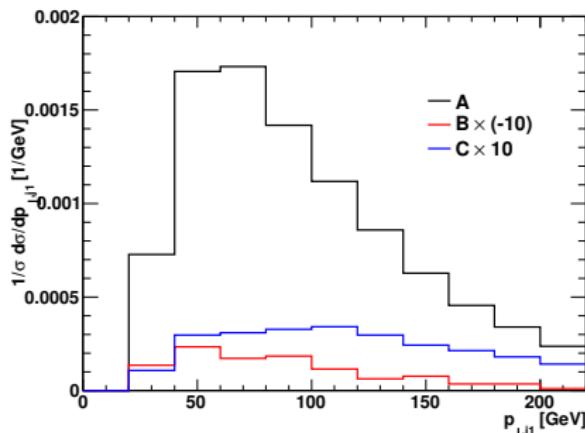
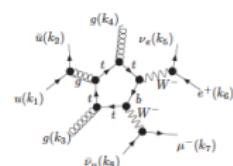
Part A:



Part B:



Part C:



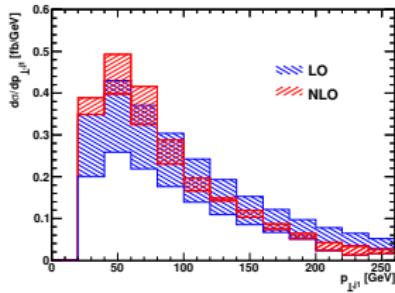
**Part A:** Leading contribution (same as Melia et al.)

**Part B:** Only 1%

**Part C:** Only 3%

$$pp \rightarrow W^+ W^- jj$$

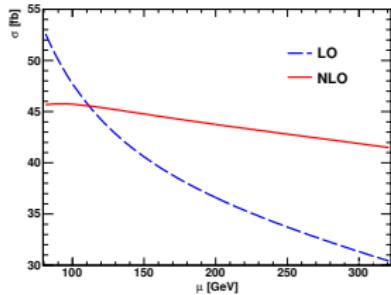
Parameters	
$M_W = 80.399$ GeV	$\Gamma_W = 2.085$ GeV
$M_Z = 91.188$ GeV	$\Gamma_Z = 2.4952$ GeV
$M_t = 171.2$ GeV	$\Gamma_t = 0.$ GeV
$M_b = 4.7$ GeV	$\Gamma_b = 0.$ GeV
$\alpha(M_Z) = 1/128.802$	$c_W^2 = M_W^2/M_Z^2$



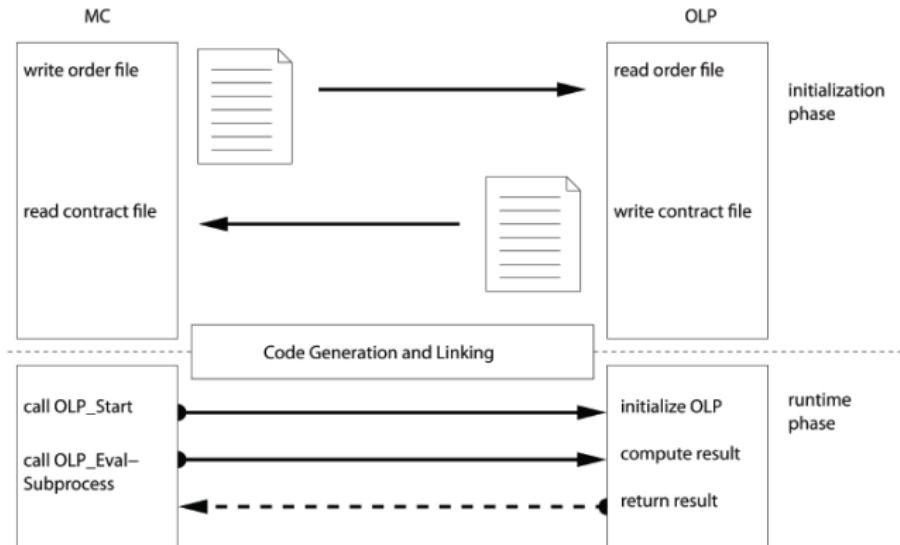
Adding the NLO strongly reduces the theoretical uncertainties

$$\sigma_{\text{lo}} = 39.57^{+34\%}_{-23\%} \text{ fb}$$

$$\sigma_{\text{nlo}} = 44.51^{+2.5\%}_{-7.4\%} \text{ fb}$$

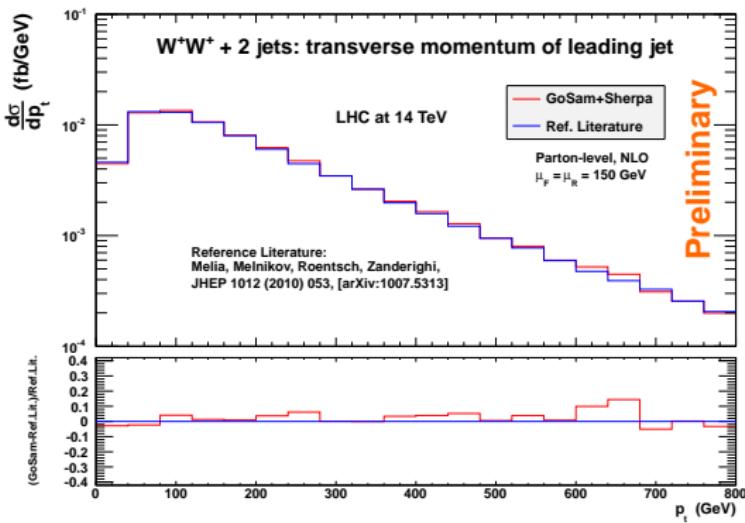


GoSam supports the BLHA interface  
for the full integration with MC codes



# EXAMPLE: GoSAM + SHERPA

$$pp \rightarrow W^+ W^+ jj$$



## NUMBER OF EVENTS:

Born : 1'000'000 × 10

Real : 50'000'000 × 10

Virtual: 1'000'000 × 10

## TIMINGS:

Running

Born : ~30 min

Real : ~ 13h 30 min

Virtual: ~7h

## Machine

Intel(R) Core(TM)2 Quad CPU Q6600 @

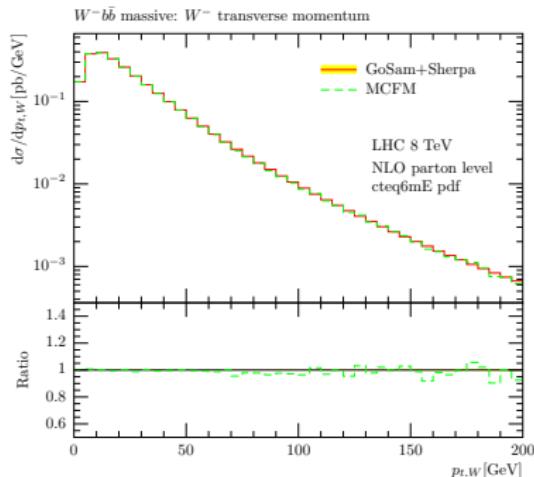
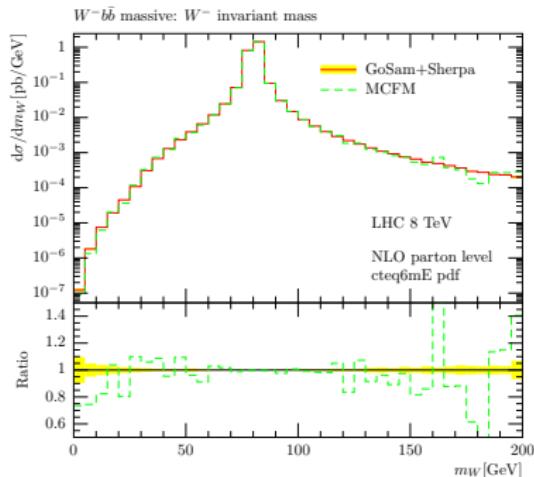
2.40GHz

RAM 6 GB

Thanks to G. Luisoni and F. Tramontano

# EXAMPLE: GoSAM + SHERPA

$pp \rightarrow W^- b\bar{b}$  (massive b)



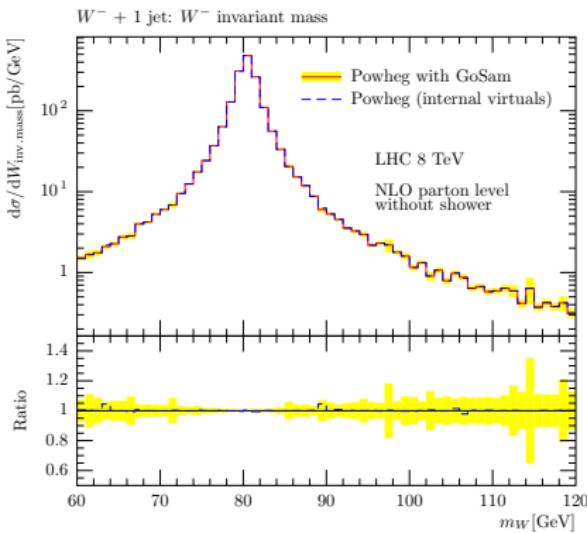
NUMBER OF EVENTS:  
Born : 5'000'000 × 5  
Real : 50'000'000 × 10  
Virtual: 5'000'000 × 10

TIMINGS:  
Born : 9 min  
Real : 5h 20 min  
Virtual: 11h

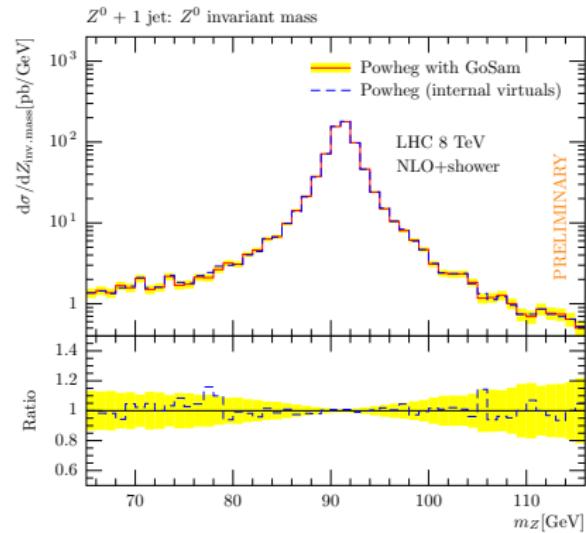
Thanks to G. Luisoni and F. Tramontano

# EXAMPLE: GoSAM + POWHEG

$W + 1 \text{ jet}$  (NLO without shower)



$Z + 1 \text{ jet}$  (NLO + shower)



Thanks to G. Luisoni and C. Oleari

# CONCLUSIONS

## GoSam is a flexible and broadly applicable tool for NLO calculations

- based on Feynman diagrams + d-dimensional integrand-level reduction
- generated code is fast and efficient, tests passed for several subprocesses
- GoSam has been used in the computation of  $pp \rightarrow 4b$  and  $pp \rightarrow W^+W^-jj$  cross sections at the LHC
- interface allows to compute all processes with Sherpa and with Powheg
- it is **publicly available**

Several projects are in progress:

- BSM: MSSM neutralino production
- Upgrade GoSam to higher rank
- More NLO processes and phenomenology