Fermilab

Automatic Complete NLO Calculations with MadGraph

Johan Alwall
Fermilab

For the MadGraph@NLO (MadFKS, MadLoop, aMC@NLO) team

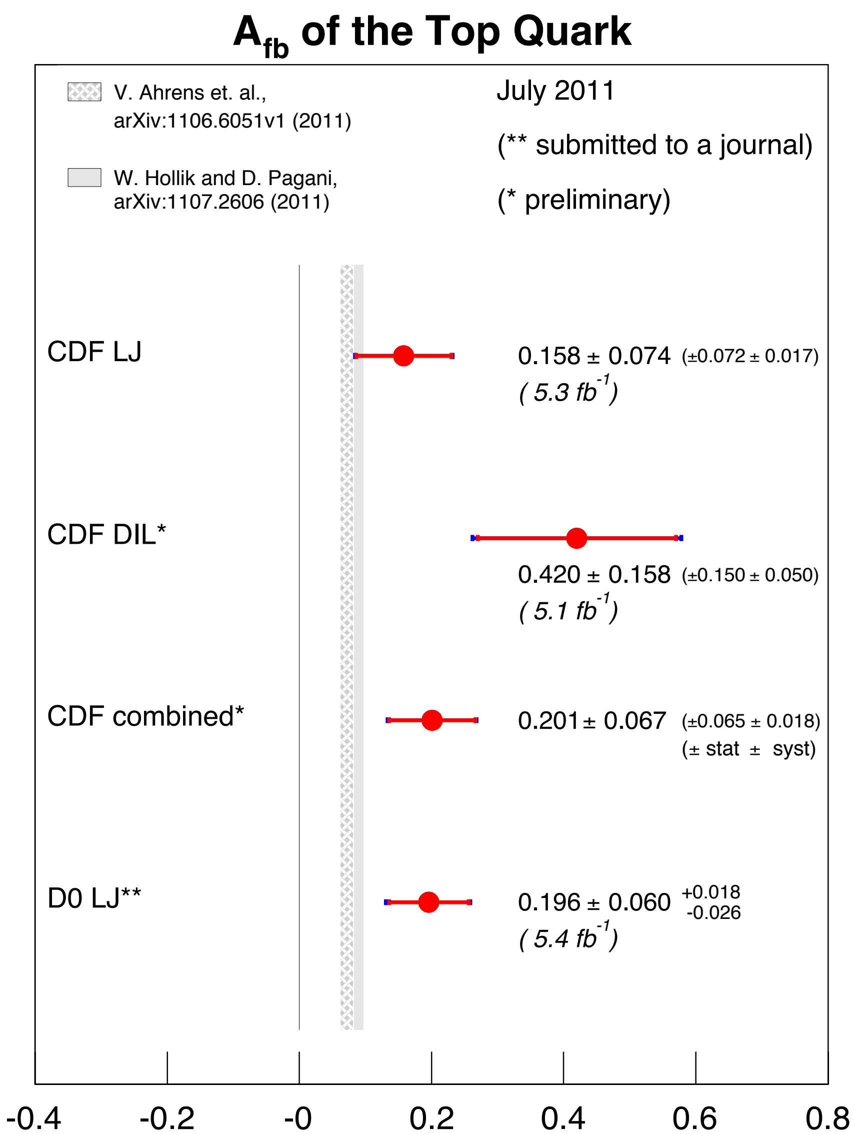
Chung Yuan HEP Seminar
1 March 2012

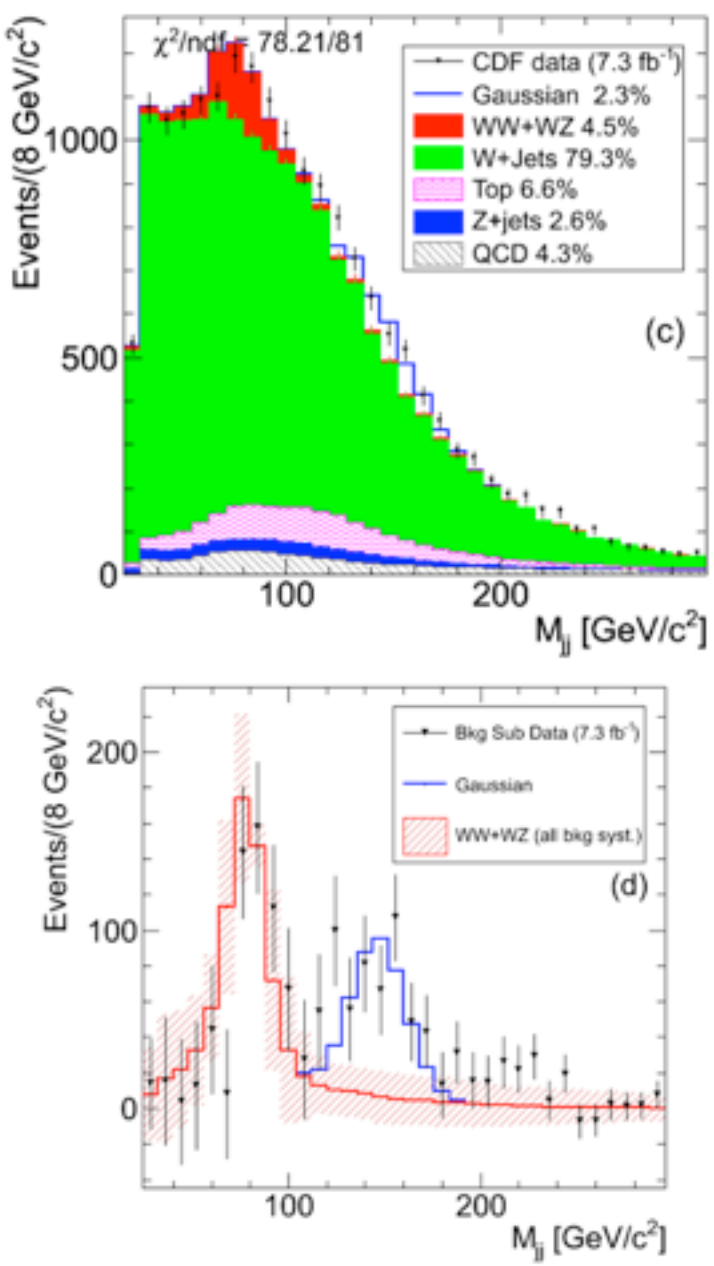
Thanks to Rikkert Frederix and Valentin Hirschi for many slides!

Exciting times!

Exciting times!

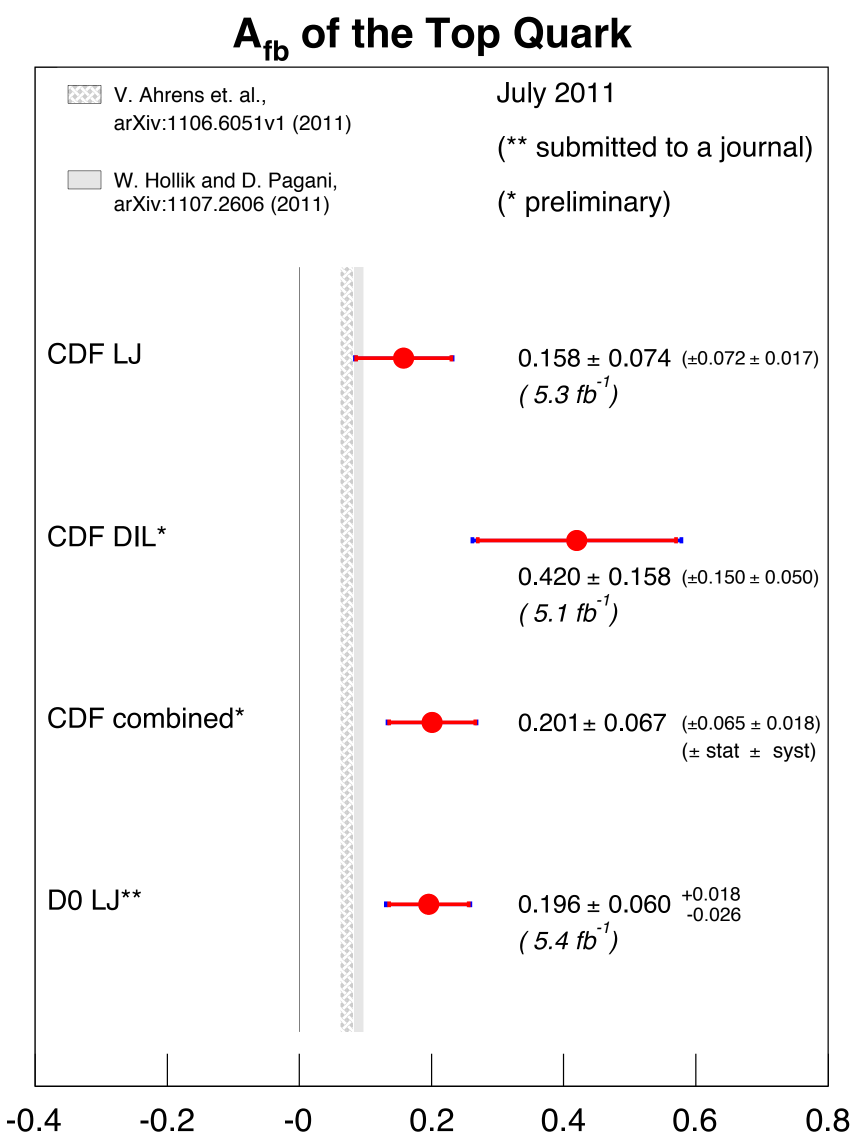
The Tevatron has left a legacy of tantalizing hints

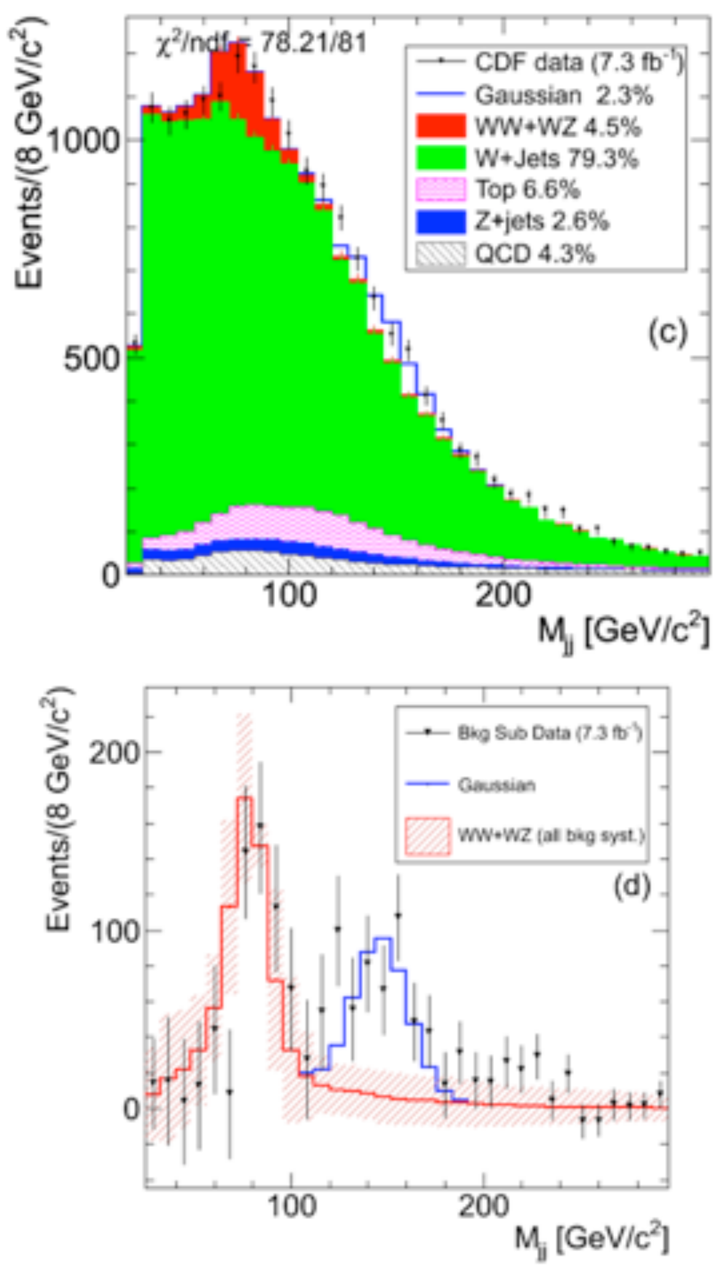




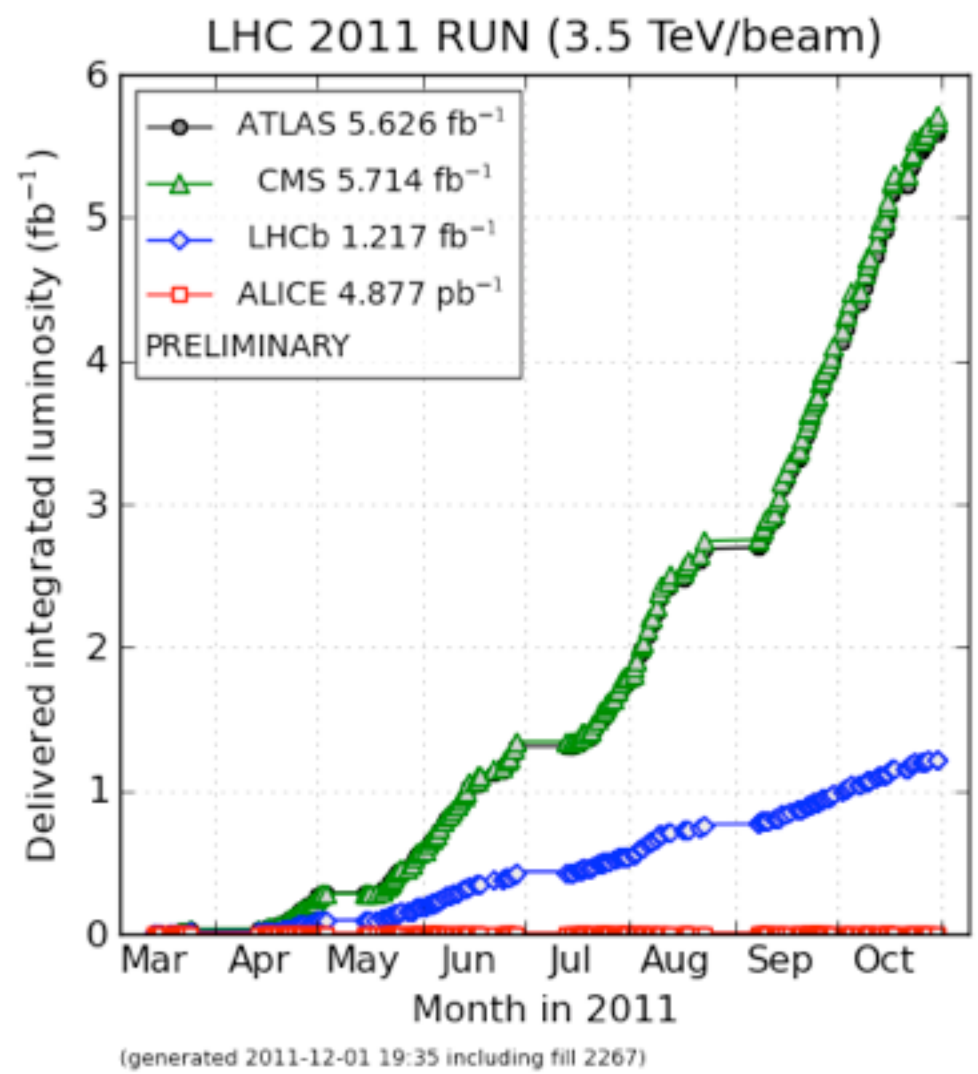
Exciting times!

The Tevatron has left a legacy of tantalizing hints



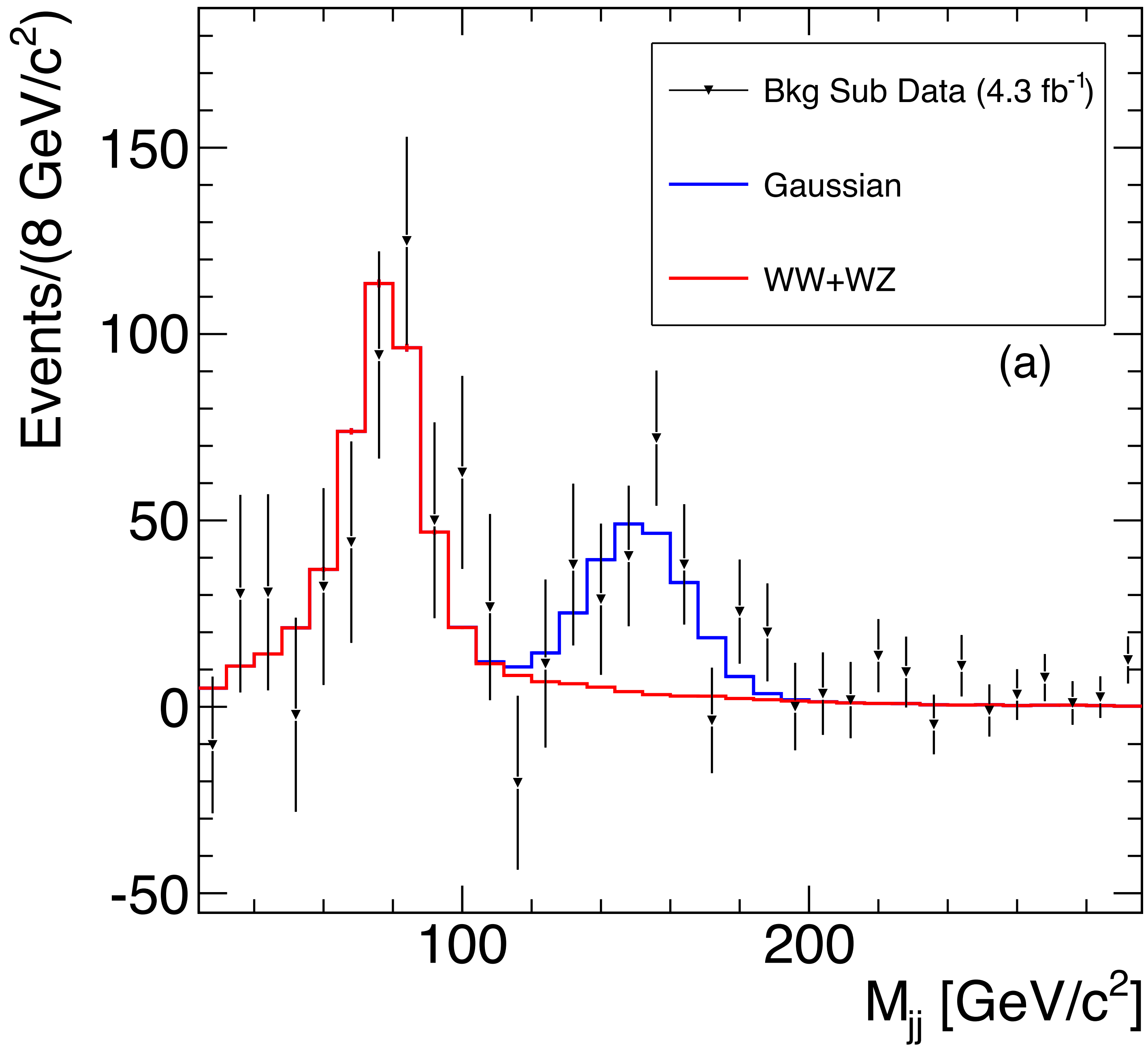


while the LHC is collecting luminocity at a spectacular rate!



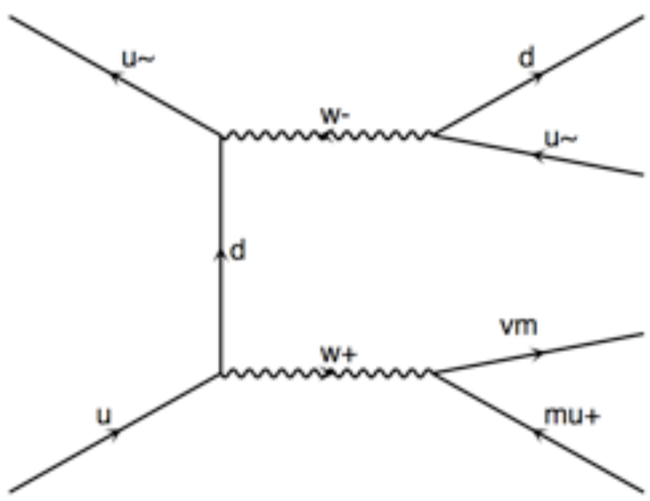
Example: CDF excess in W + 2 jets

CDF collaboration, arXiv:1104.0699



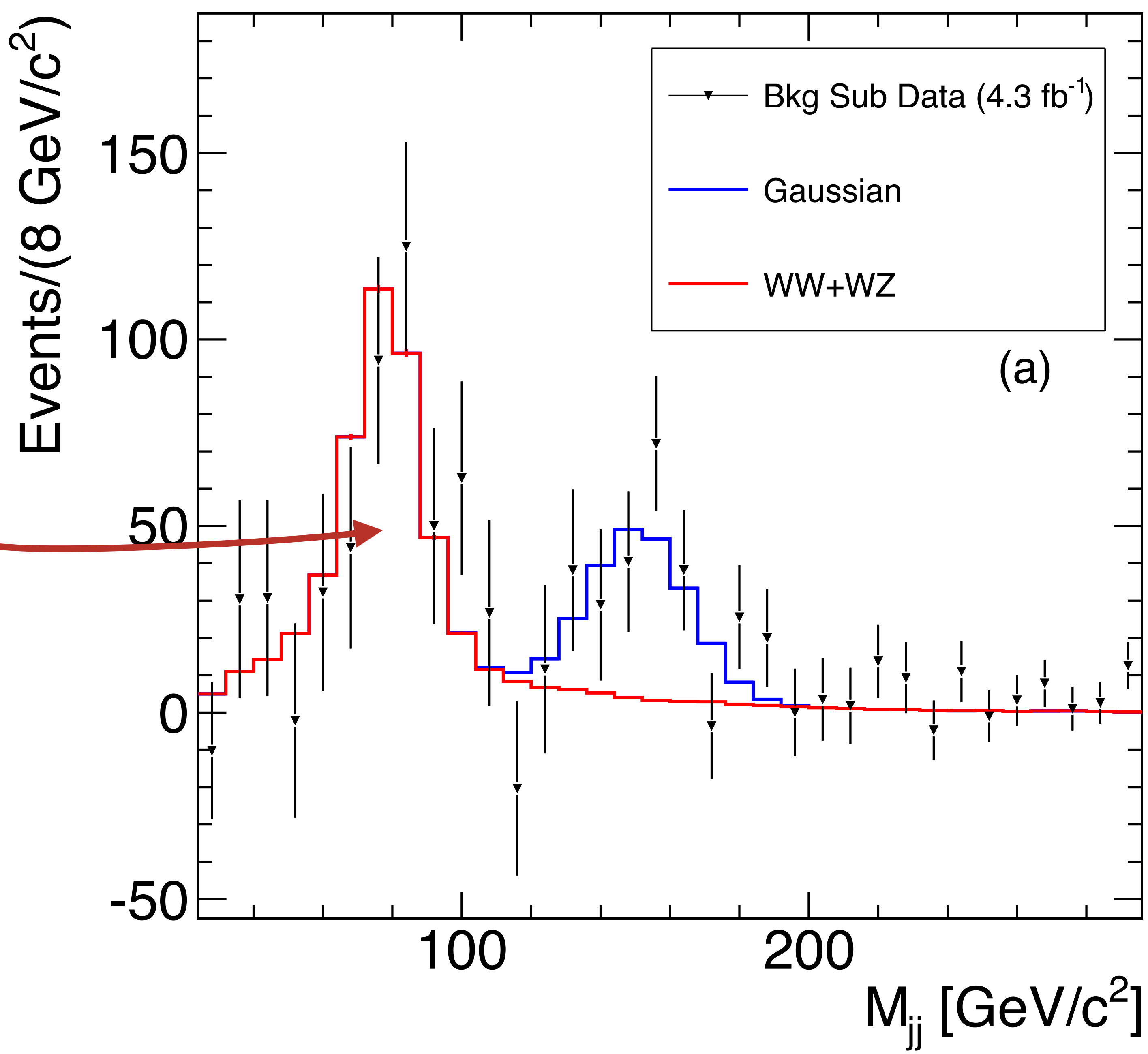
Background subtracted data (except WW/WZ)

Example: CDF excess in W + 2 jets



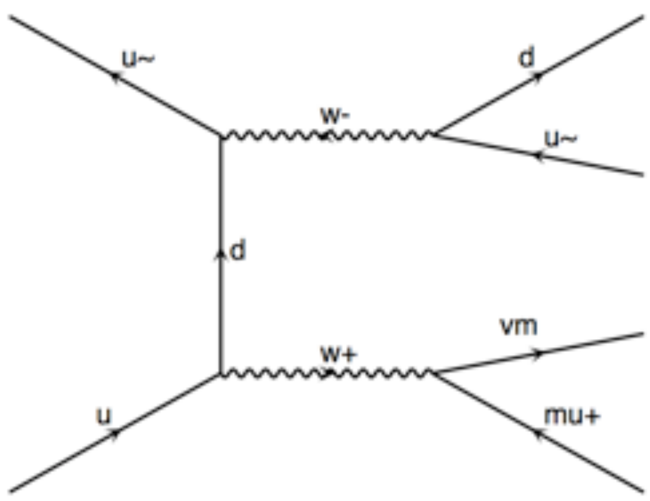
WW, WZ

CDF collaboration, arXiv:1104.0699

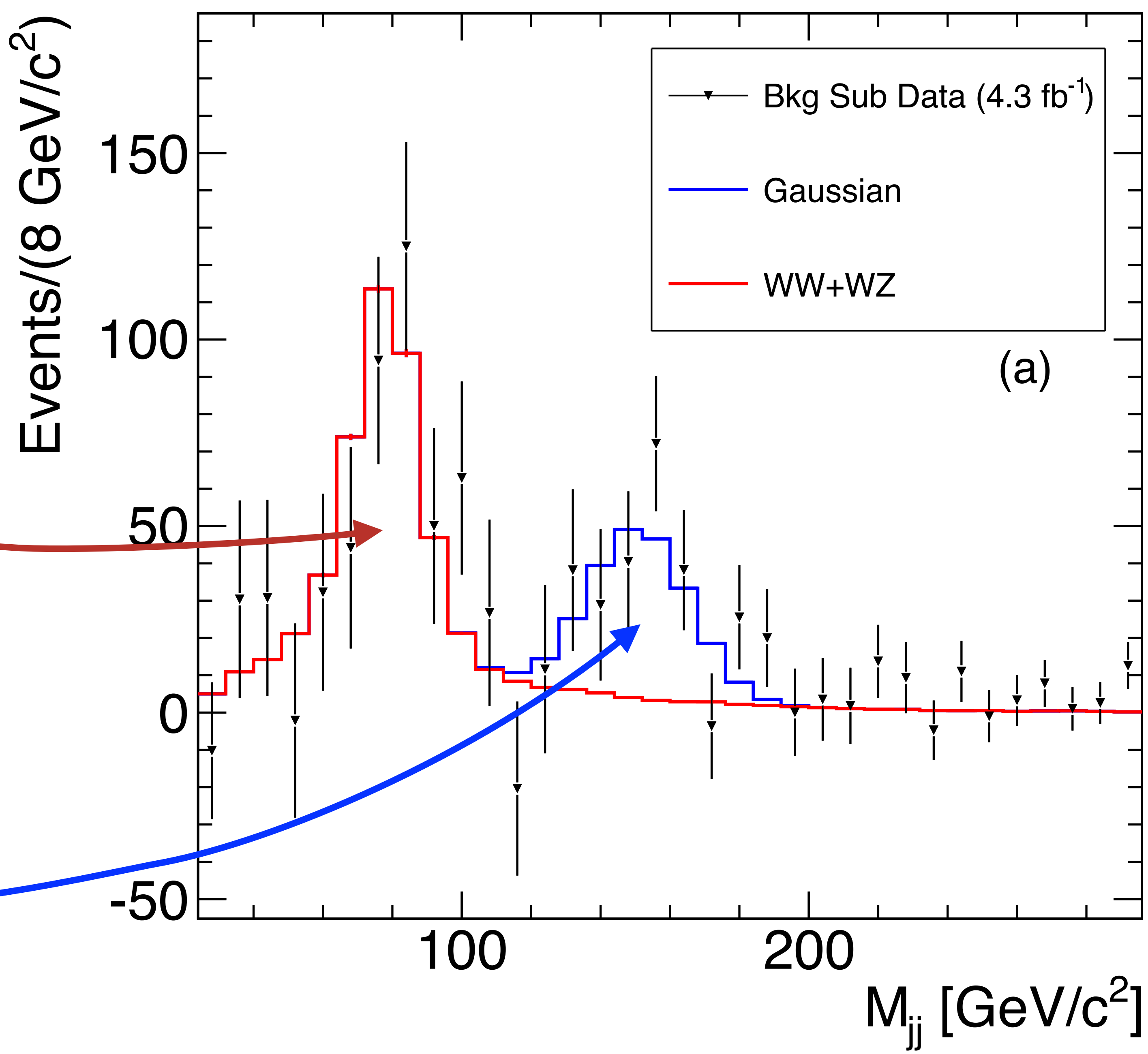


Background subtracted data (except WW/WZ)

Example: CDF excess in W + 2 jets



CDF collaboration, arXiv:1104.0699



WW, WZ

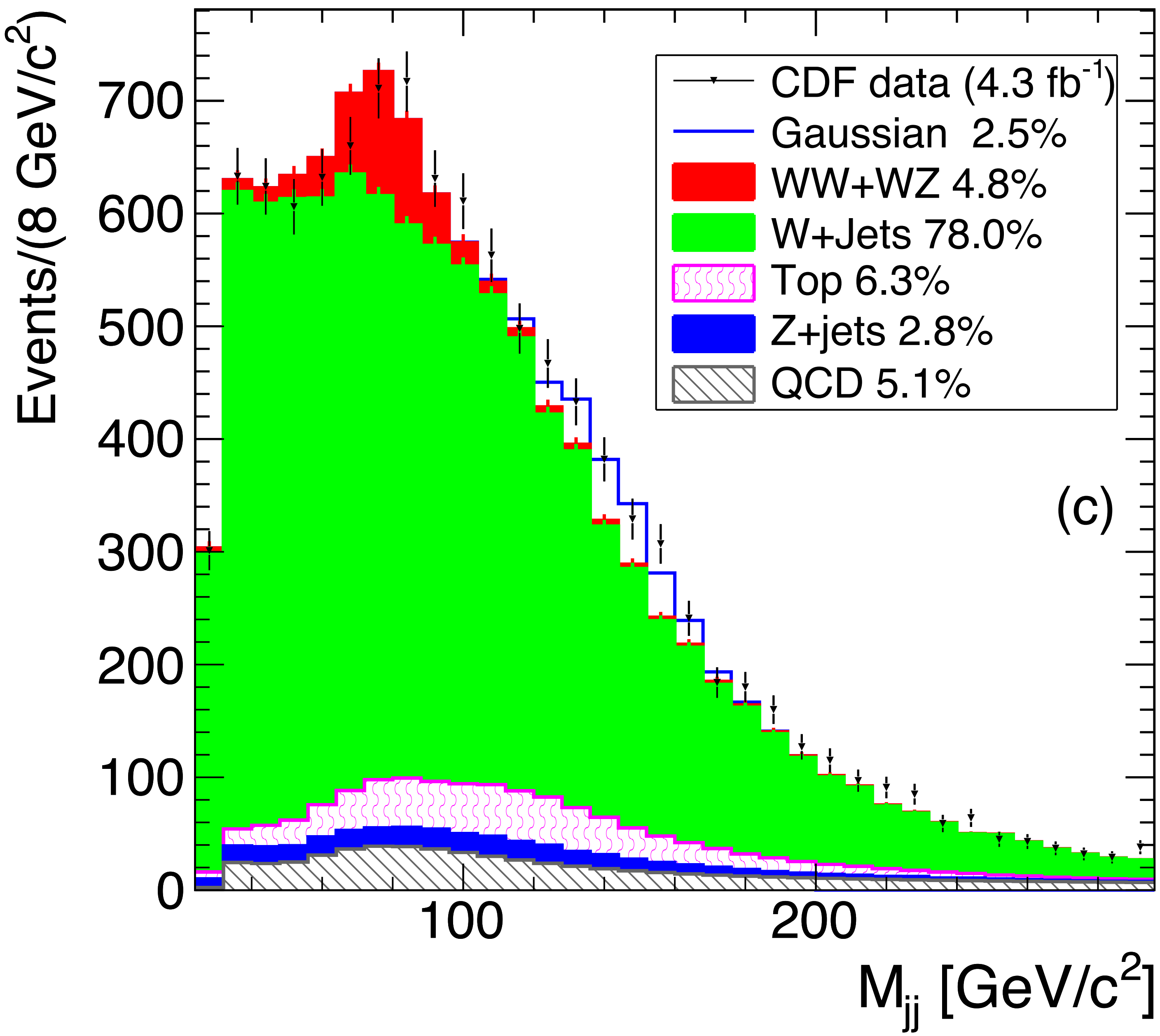
W + Nobody knows what?

Background subtracted data (except WW/WZ)

Example: CDF excess in W + 2 jets

A more complete picture

CDF collaboration, arXiv:1104.0699

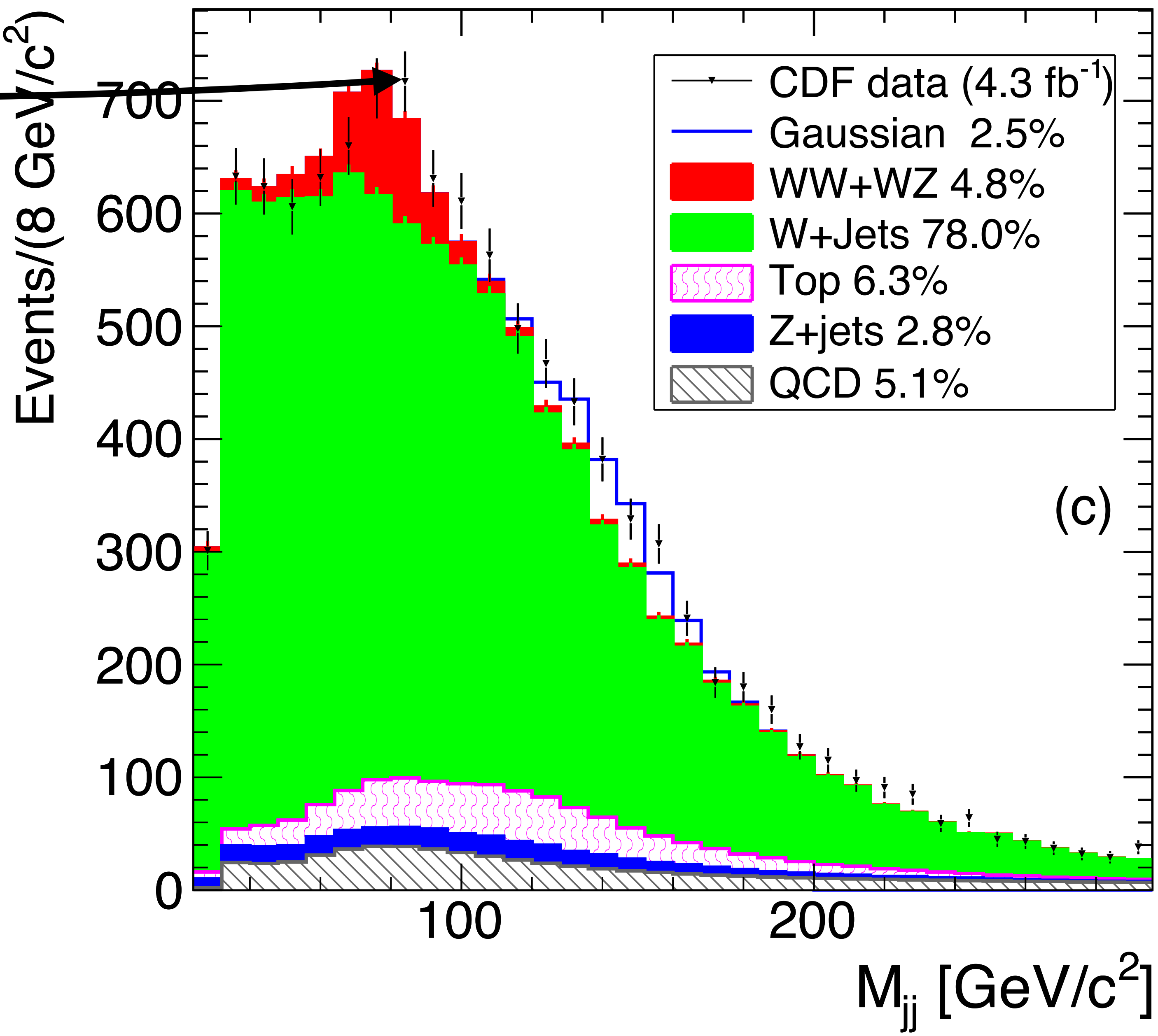


Example: CDF excess in W + 2 jets

A more complete picture

CDF collaboration, arXiv:1104.0699

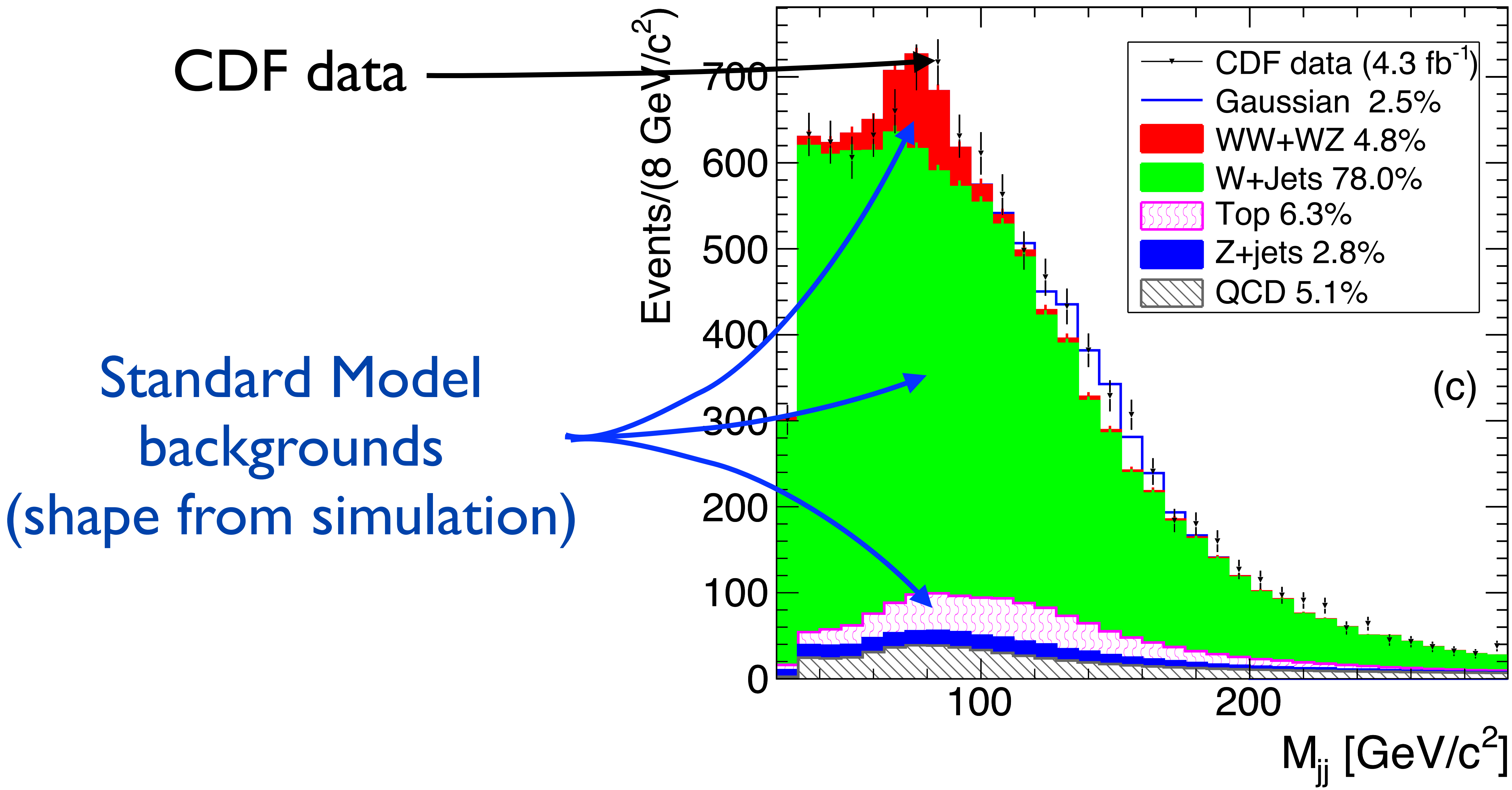
CDF data



Example: CDF excess in W + 2 jets

A more complete picture

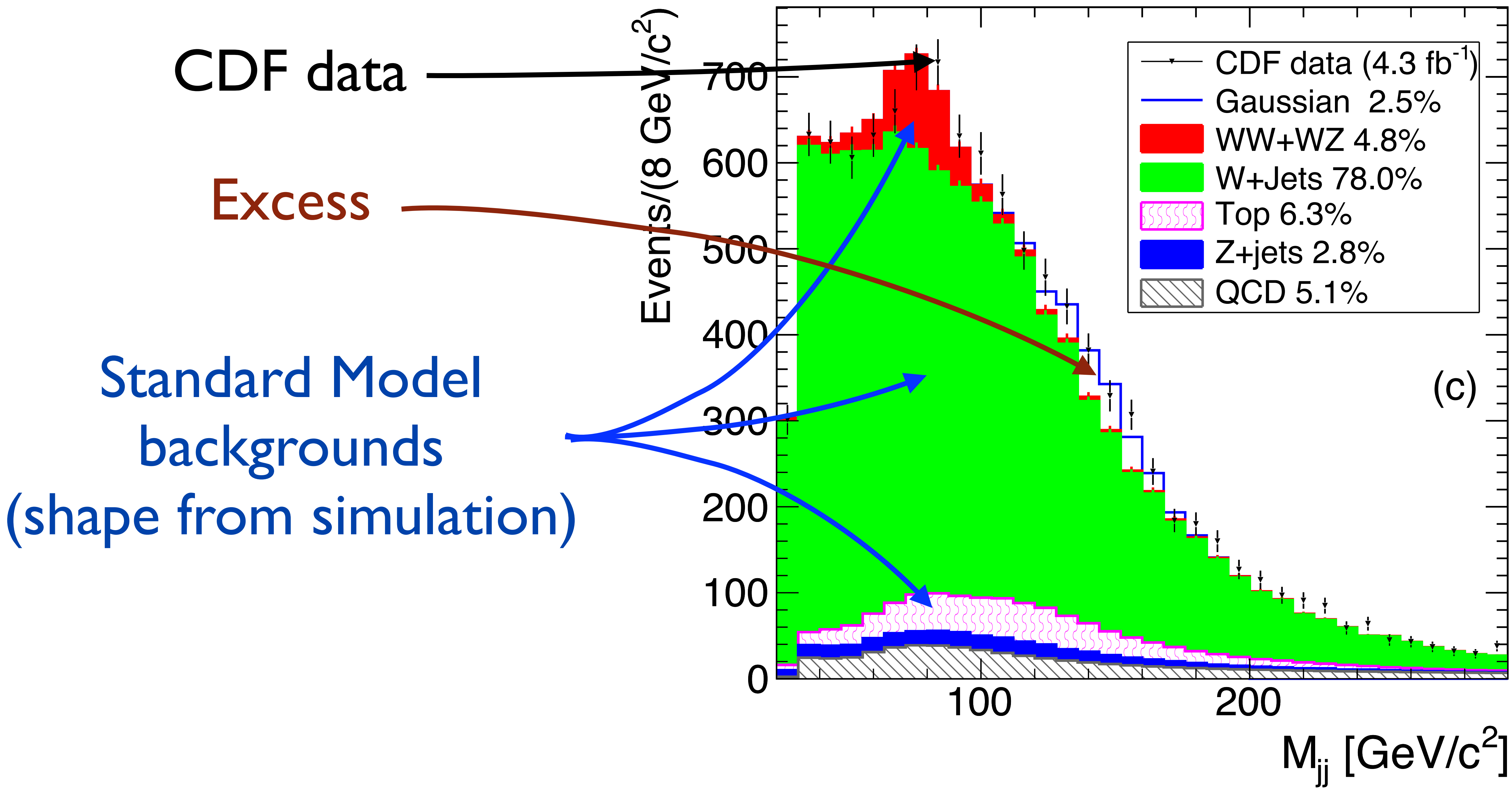
CDF collaboration, arXiv:1104.0699



Example: CDF excess in W + 2 jets

A more complete picture

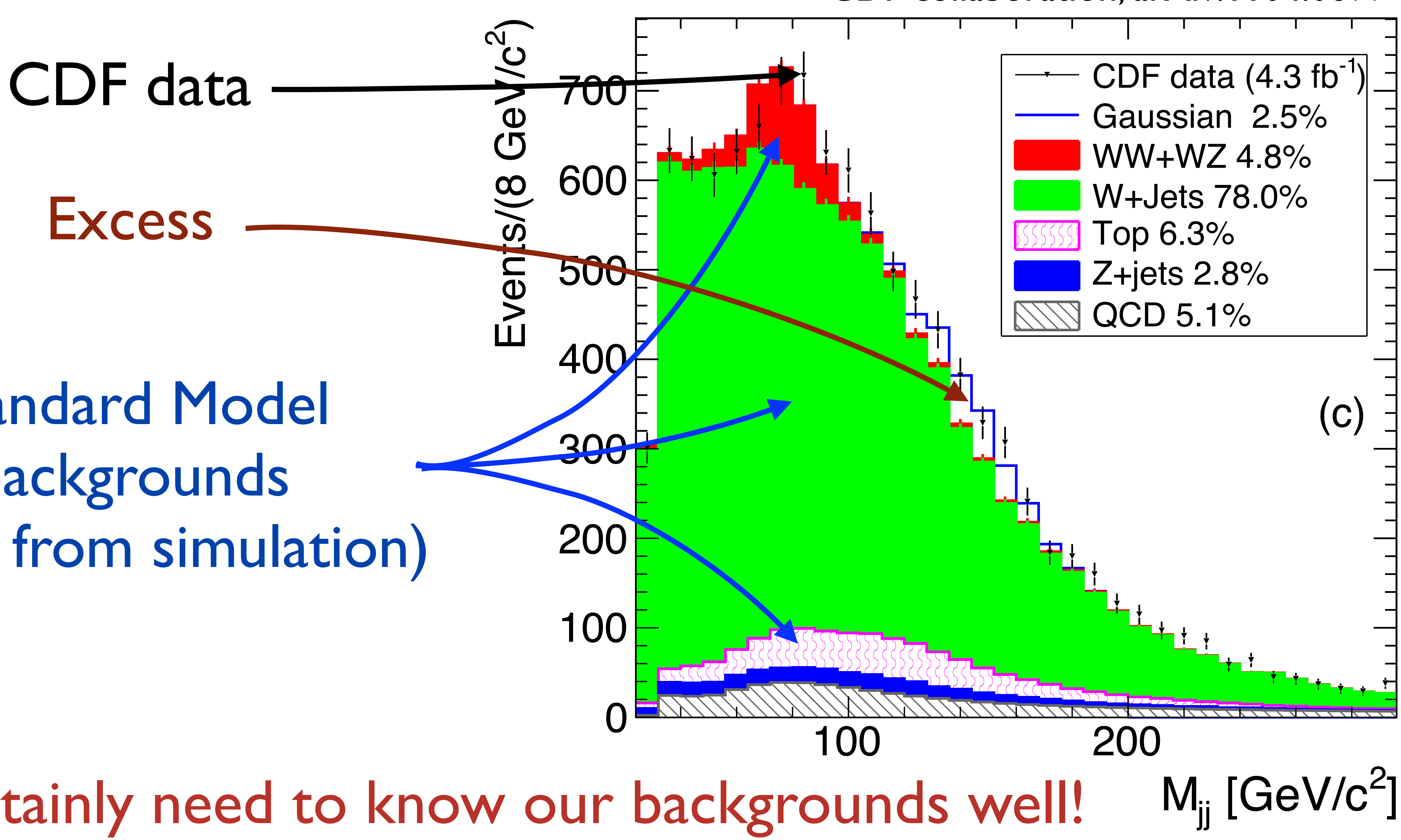
CDF collaboration, arXiv:1104.0699



Example: CDF excess in W + 2 jets

A more complete picture

CDF collaboration, arXiv:1104.0699



We certainly need to know our backgrounds well!

How ensure sufficient precision in SM background simulations?

How ensure sufficient precision in SM background simulations?

The answer is NLO!

$$\hat{\sigma} = \sigma^{\mathrm{Born}}\left(\boxed{1 + \frac{\alpha_s}{2\pi}\sigma^{(1)}} + \left(\frac{\alpha_s}{2\pi}\right)^2\sigma^{(2)} + \left(\frac{\alpha_s}{2\pi}\right)^3\sigma^{(3)} + \ldots\right)$$

How ensure sufficient precision in SM background simulations?

The answer is NLO!

$$\hat{\sigma} = \sigma^{\text{Born}} \left(\boxed{1 + \frac{\alpha_s}{2\pi}\sigma^{(1)}} + \left(\frac{\alpha_s}{2\pi}\right)^2 \sigma^{(2)} + \left(\frac{\alpha_s}{2\pi}\right)^3 \sigma^{(3)} + \dots \right)$$

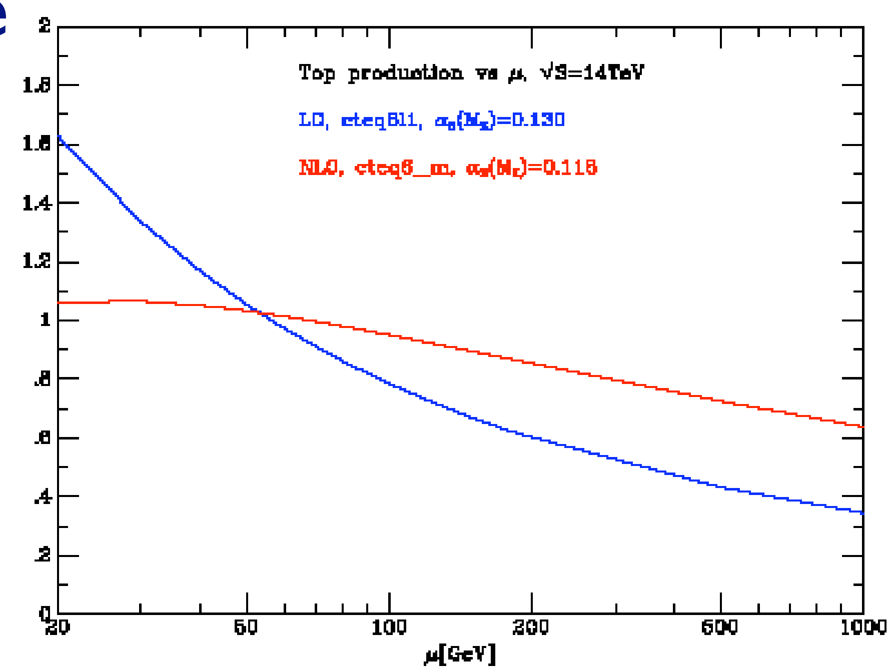
- Leading order QCD has large dependence on the choice of renormalization and factorization scales

How ensure sufficient precision in SM background simulations?

The answer is NLO!

$$\hat{\sigma} = \sigma^{\text{Born}} \left(\boxed{1 + \frac{\alpha_s}{2\pi}\sigma^{(1)}} + \left(\frac{\alpha_s}{2\pi}\right)^2 \sigma^{(2)} + \left(\frac{\alpha_s}{2\pi}\right)^3 \sigma^{(3)} + \dots \right)$$

- Leading order QCD has large dependence on the choice of renormalization and factorization scales
- At next-to-leading order, the dependence on scales for the running coupling and PDFs is compensated by loop corrections

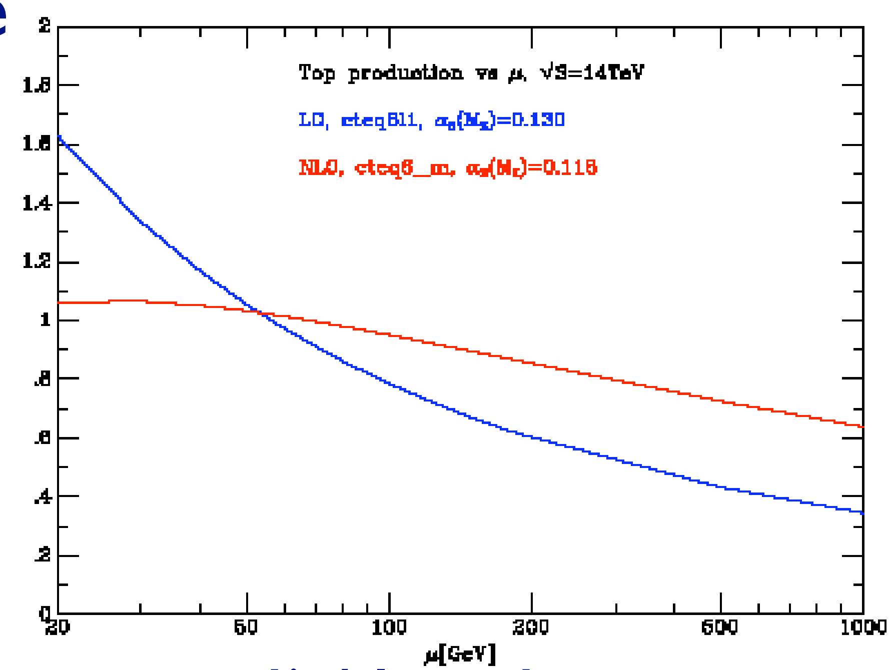


How ensure sufficient precision in SM background simulations?

The answer is NLO!

$$\hat{\sigma} = \sigma^{\text{Born}} \left(\boxed{1 + \frac{\alpha_s}{2\pi}\sigma^{(1)}} + \left(\frac{\alpha_s}{2\pi}\right)^2 \sigma^{(2)} + \left(\frac{\alpha_s}{2\pi}\right)^3 \sigma^{(3)} + \ldots \right)$$

- Leading order QCD has large dependence on the choice of renormalization and factorization scales



- At next-to-leading order, the dependence on scales for the running coupling and PDFs is compensated by loop corrections
- First order where cross section is well-determined! Also shape predictions from leading order have to be validated with NLO.

NLO wishlist for LHC (anno 2005)

process ($V \in \{Z, W, \gamma\}$)	background to
1. $pp \to V\,V$ jet	$t\bar{t}H$, new physics
2. $pp \to H + 2$ jets	H production by VBF
3. $pp \to t\bar{t}\,b\bar{b}$	$t\bar{t}H$
4. $pp \to t\bar{t} + 2$ jets	$t\bar{t}H$
5. $pp \to V\,V\,b\bar{b}$	VBF$\to H \to VV$, $t\bar{t}H$, new physics
6. $pp \to V\,V + 2$ jets	VBF$\to H \to VV$
7. $pp \to V + 3$ jets	various new physics signatures
8. $pp \to V\,V\,V$	SUSY trilepton

Slide from Gudrun Heinrich

"The NLO Revolution"

One indicator of NLO progress

pp → W + 0 jet	1978	Altarelli, Ellis, Martinelli
pp → W + 1 jet	1989	Arnold, Ellis, Reno
pp → W + 2 jets	2002	Campbell, Ellis
pp → W + 3 jets	2009	BH+Sherpa Ellis, Melnikov, Zanderighi
pp → W + 4 jets	2010	BH+Sherpa

Slide from Lance Dixon

"The NLO Revolution"

- The "loop revolution": new techniques for computing one-loop matrix elements are now established:
  - ➡ Generalized unitarity (e.g. BlackHat, Rocket, ...) [Bern, Dixon, Dunbar, Kosower, 1994...; Ellis Giele Kunst 2007 + Melnikov 2008;...]
  - ➡ Integrand reduction (e.g. CutTools, GoSam) [Ossola, Papadopoulos, Pittau 2006; del Aguila, Pittau 2004; Mastrolia, Ossola, Reiter, Tramontano 2010;...]
  - ➡ Tensor reduction (e.g. Golem) [Passarino, Veltman 1979; Denner, Dittmaier 2005; Binoth Guillet, Heinrich, Pilon, Reiter 2008]

“The NLO Revolution”

“The NLO Revolution”

- Since many years, one of the (most) important jobs of HEP theorists have been to provide experiments with NLO predictions for observables

"The NLO Revolution"

- Since many years, one of the (most) important jobs of HEP theorists have been to provide experiments with NLO predictions for observables
- Despite the exceptional development indicated on the last slides, there is still much manual work going into implementing a single new process

“The NLO Revolution”

- Since many years, one of the (most) important jobs of HEP theorists have been to provide experiments with NLO predictions for observables
- Despite the exceptional development indicated on the last slides, there is still much manual work going into implementing a single new process
- Every process must be implemented by several groups independently to ensure that the code is correct

"The NLO Revolution"

- Since many years, one of the (most) important jobs of HEP theorists have been to provide experiments with NLO predictions for observables
- Despite the exceptional development indicated on the last slides, there is still much manual work going into implementing a single new process
- Every process must be implemented by several groups independently to ensure that the code is correct
- Good for training PhD students, but bad when a background is needed urgently, or we need to know that the implementation is bug free!

"The NLO Revolution"

"The NLO Revolution"

- At leading order, we have access to powerful, completely automated process generation and event generation since many years!

"The NLO Revolution"

- At leading order, we have access to powerful, completely automated process generation and event generation since many years!
- When will this be reality for NLO?

MadGraph Home Page

madgraph.hep.uiuc.edu

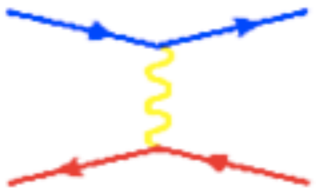
High Energy Physics Illinois

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.
Please note the correct reference for MadGraph 5, JHEP 1106(2011)128, arXiv:1106.0522 [hep-ph].
You can still use **MadGraph 4** here.

Code can be generated either by:

I. Fill the form:

Model: SM ○ LO Model descriptions

Input Process: [ ] ◉ NLO Examples/format

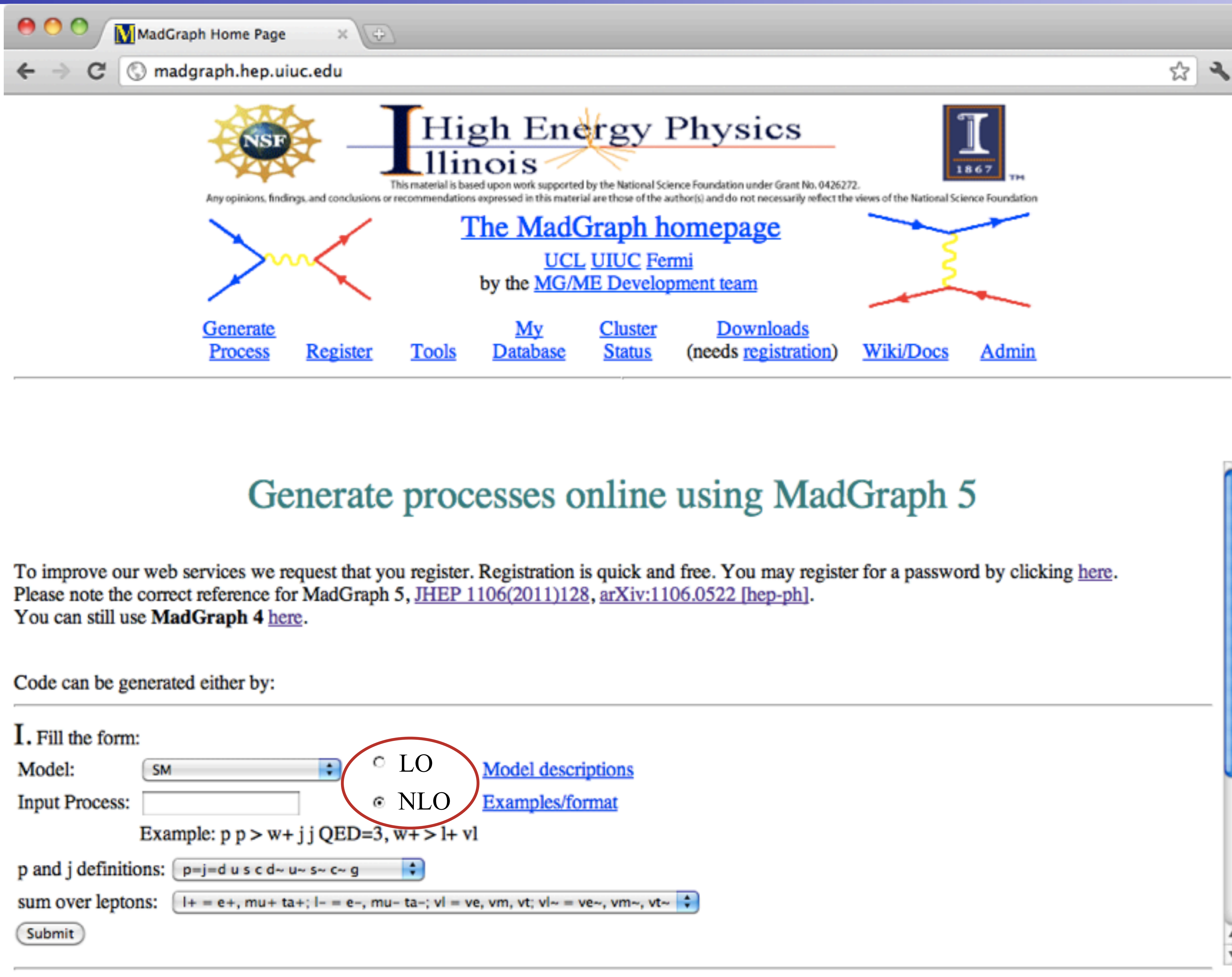
Example: p p > w+ j j QED=3, w+ > l+ vl

p and j definitions: p=j=d u s c d~ u~ s~ c~ g

sum over leptons: l+ = e+, mu+ ta+; l- = e-, mu- ta-; vl = ve, vm, vt; vl~ = ve~, vm~, vt~

Submit

Fermilab


MadGraph Home Page
madgraph.hep.uiuc.edu
NSF
High Energy Physics
Illinois
1867
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.
Please note the correct reference for MadGraph 5, JHEP 1106(2011)128, arXiv:1106.0522 [hep-ph].
You can still use MadGraph 4 here.
Code can be generated either by:
I. Fill the form:
Model:
SM
LO
NLO
Model descriptions
Input Process:
Examples/format
Example: p p > w+ j j QED=3, w+ > l+ vl
p and j definitions:
p=j=d u s c d~ u~ s~ c~ g
sum over leptons:
l+ = e+, mu+ ta+; l- = e-, mu- ta-; vl = ve, vm, vt; vl~ = ve~, vm~, vt~
Submit

MadGraph Home Page

madgraph.hep.uiuc.edu

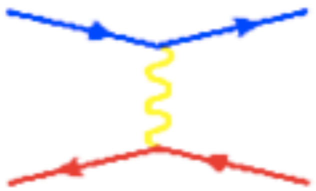
High Energy Physics Illinois

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.
Please note the correct reference for MadGraph 5, JHEP 1106(2011)128, arXiv:1106.0522 [hep-ph].
You can still use **MadGraph 4** here.

Code can be generated either by:

I. Fill the form:

Model: SM — ○ LO / ◉ NLO — Model descriptions

Input Process: — Examples/format

Example: p p > w+ j j QED=3, w+ > l+ vl

p and j definitions: p=j=d u s c d~ u~ s~ c~ g

sum over leptons: l+ = e+, mu+ ta+; l- = e-, mu- ta-; vl = ve, vm, vt; vl~ = ve~, vm~, vt~

Submit

We are there!

MadGraph Home Page

madgraph.hep.uiuc.edu

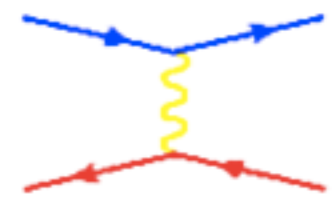
High Energy Physics Illinois

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.
Please note the correct reference for MadGraph 5, JHEP 1106(2011)128, arXiv:1106.0522 [hep-ph].
You can still use **MadGraph 4** here.

Code can be generated either by:

I. Fill the form:

Model: SM ○ LO ⊙ NLO Model descriptions

Input Process: Examples/format

Example: p p > w+ j j QED=3, w+ > l+ vl

p and j definitions: p=j=d u s c d~ u~ s~ c~ g

sum over leptons: l+ = e+, mu+ ta+; l- = e-, mu- ta-; vl = ve, vm, vt; vl~ = ve~, vm~, vt~

Submit

We are there!
(but it's not yet quite this public)

Why automation?

Why automation?

- Save time
  - ➡ Trade time spent on computing a process with time on studying the physics!

Why automation?

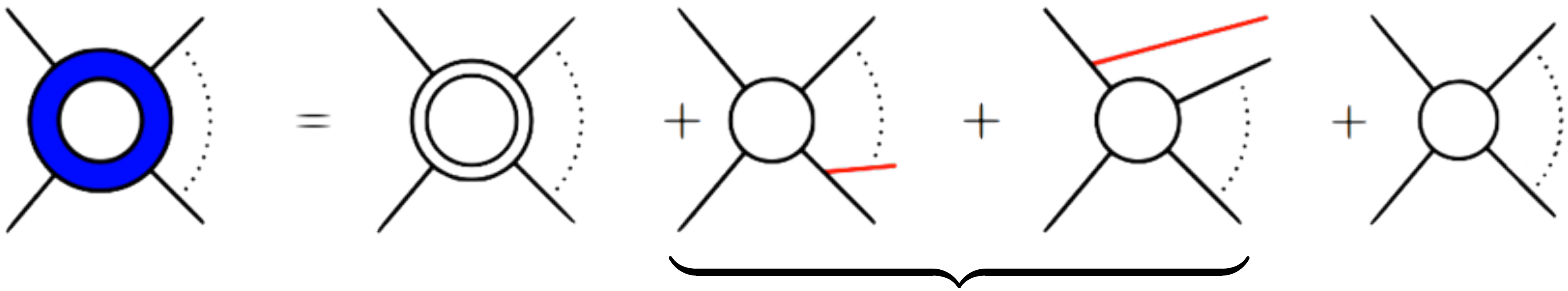
- Save time
  - ➡ Trade time spent on computing a process with time on studying the physics!
- Avoid bugs
  - ➡ Having a trusted program extensively checked once and for all, eliminates bugs when running different processes!

Why automation?

- Save time
  - ➡ Trade time spent on computing a process with time on studying the physics!
- Avoid bugs
  - ➡ Having a trusted program extensively checked once and for all, eliminates bugs when running different processes!
- Use of the same framework for all processes
  - ➡ It only requires to know how to efficiently use one single program to do all NLO phenomenology!

NLO Basics

NLO contributions have two parts



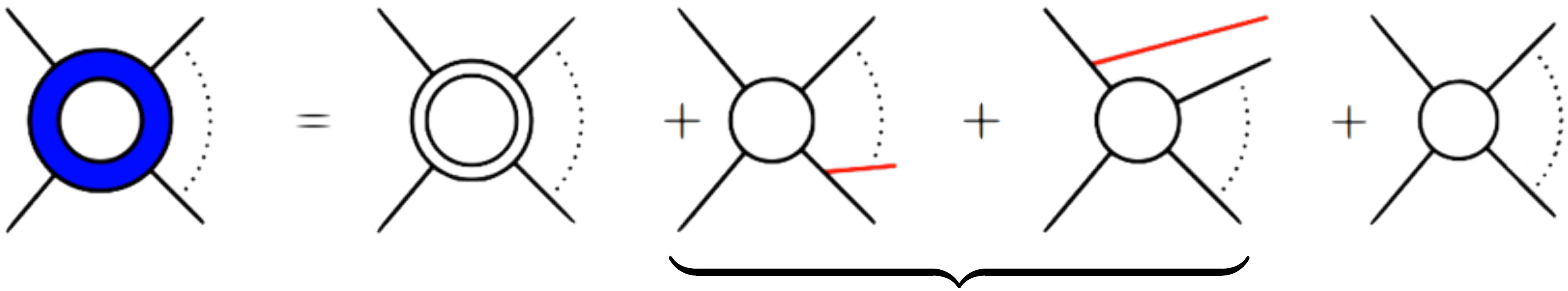
$$\sigma^{\text{NLO}} = \int_m d^{(d)}\sigma^V + \int_{m+1} d^{(d)}\sigma^R + \int_m d^{(4)}\sigma^B$$

Virtual (loop) part — Real emission part — Born

Individually divergent, but divergencies cancel

NLO Basics

NLO contributions have two parts



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

Virtual (loop) part — Real emission part — Born

- Used to be bottleneck of NLO computations
- Algorithms for automation known in principle but not previously efficiently implemented

- Automated for multiple methods
- Challenge is the systematic extraction and cancellation of singularities

Subtraction terms

$$\sigma^{\mathrm{NLO}} \sim \int d^4\Phi_m\, B(\Phi_m)$$
$$+ \int d^4\Phi_m \left[\int_{\mathrm{loop}} d^d l\, V(\Phi_m) + \int d^d\Phi_1 G(\overline{\Phi}_{m+1}) \right]_{\epsilon\to 0}$$
$$+ \int d^4\Phi_{m+1} \left[R(\Phi_{m+1}) - G(\overline{\Phi}_{m+1}) \right]$$

Subtraction terms

$$\sigma^{\rm NLO} \sim \int d^4\Phi_m \, B(\Phi_m)$$
$$+ \int d^4\Phi_m \left[\int_{\rm loop} d^d l \, V(\Phi_m) + \int d^d\Phi_1 G(\overline{\Phi}_{m+1}) \right]_{\epsilon \to 0}$$
$$+ \int d^4\Phi_{m+1} \left[R(\Phi_{m+1}) - G(\overline{\Phi}_{m+1}) \right]$$

- The soft and collinear divergencies of the real emissions must be subtracted out

Subtraction terms

$$\sigma^{\rm NLO} \sim \int d^4\Phi_m\, B(\Phi_m) + \int d^4\Phi_m \left[\int_{\rm loop} d^d l\, V(\Phi_m) + \int d^d\Phi_1 G(\overline{\Phi}_{m+1}) \right]_{\epsilon\to 0} + \int d^4\Phi_{m+1} \left[R(\Phi_{m+1}) - G(\overline{\Phi}_{m+1}) \right]$$

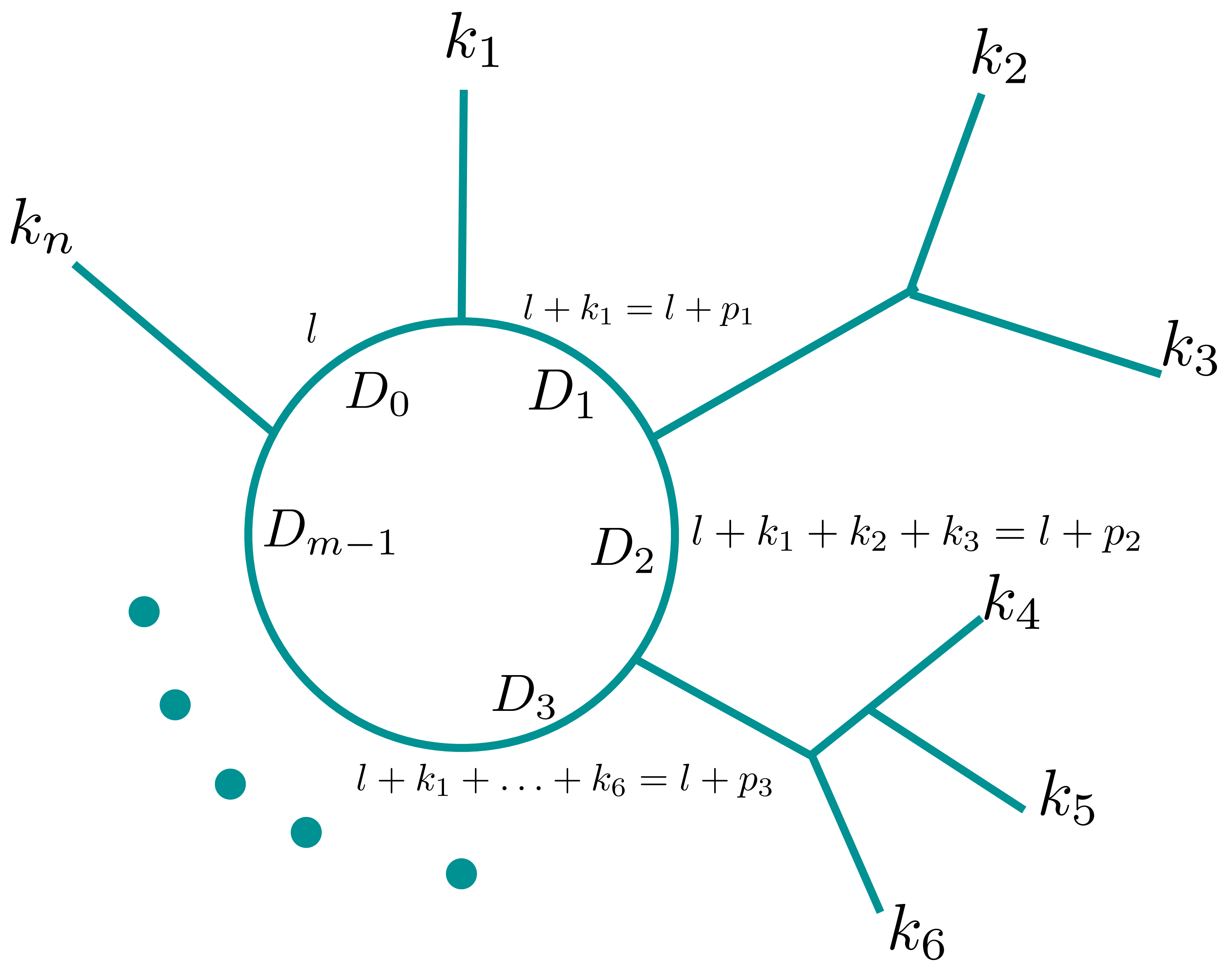
- The soft and collinear divergencies of the real emissions must be subtracted out
- These terms are then integrated over the one-parton phase space (analytically) to get the explicit poles 1/ε and added to the virtual corrections so that these poles cancel
    - These are process-independent terms proportional to the (color-linked) Borns

Subtraction terms

$$\sigma^{\text{NLO}} \sim \int d^4\Phi_m \, B(\Phi_m)$$
$$+ \int d^4\Phi_m \left[\int_{\text{loop}} d^d l \, V(\Phi_m) + \int d^d\Phi_1 G(\overline{\Phi}_{m+1}) \right]_{\epsilon \to 0}$$
$$+ \int d^4\Phi_{m+1} \left[R(\Phi_{m+1}) - G(\overline{\Phi}_{m+1}) \right]$$

- The soft and collinear divergencies of the real emissions must be subtracted out
- These terms are then integrated over the one-parton phase space (analytically) to get the explicit poles 1/ε and added to the virtual corrections so that these poles cancel
  - These are process-independent terms proportional to the (color-linked) Borns
- In MadGraph, both CS and FKS terms are implemented - FKS is used here

One-loop integral



- Consider this *m*-point loop diagram with *n* external momenta
- The integral to compute is

$$\int d^d l \, \frac{N(l)}{D_0 D_1 D_2 \cdots D_{m-1}}$$

$$D_i = (l + p_i)^2 - m_i^2$$

Integrand reduction

$$\mathcal{M}^{\text{1-loop}} = \sum_{i_0<i_1<i_2<i_3} d_{i_0 i_1 i_2 i_3} \text{Box}_{i_0 i_1 i_2 i_3} + \sum_{i_0<i_1<i_2} c_{i_0 i_1 i_2} \text{Triangle}_{i_0 i_1 i_2} + \sum_{i_0<i_1} b_{i_0 i_1} \text{Bubble}_{i_0 i_1} + \sum_{i_0} a_{i_0} \text{Tadpole}_{i_0} + R + \mathcal{O}(\epsilon)$$

Integrand reduction

- Express loop amplitude as sum of scalar integrals at integrand level

$$
\begin{aligned}
\mathcal{M}^{1\text{-loop}} = & \sum_{i_0<i_1<i_2<i_3} d_{i_0 i_1 i_2 i_3} \mathrm{Box}_{i_0 i_1 i_2 i_3} \\
& + \sum_{i_0<i_1<i_2} c_{i_0 i_1 i_2} \mathrm{Triangle}_{i_0 i_1 i_2} \\
& + \sum_{i_0<i_1} b_{i_0 i_1} \mathrm{Bubble}_{i_0 i_1} \\
& + \sum_{i_0} a_{i_0} \mathrm{Tadpole}_{i_0} \\
& + R + \mathcal{O}(\epsilon)
\end{aligned}
$$

Integrand reduction

- Express loop amplitude as sum of scalar integrals at integrand level
- Allows to calculate integral for a given phase space point by solving a system of equations

$$\mathcal{M}^{\text{1-loop}} = \sum_{i_0<i_1<i_2<i_3} d_{i_0 i_1 i_2 i_3} \text{Box}_{i_0 i_1 i_2 i_3} + \sum_{i_0<i_1<i_2} c_{i_0 i_1 i_2} \text{Triangle}_{i_0 i_1 i_2} + \sum_{i_0<i_1} b_{i_0 i_1} \text{Bubble}_{i_0 i_1} + \sum_{i_0} a_{i_0} \text{Tadpole}_{i_0} + R + \mathcal{O}(\epsilon)$$

Integrand reduction

- Express loop amplitude as sum of scalar integrals at integrand level
- Allows to calculate integral for a given phase space point by solving a system of equations
- Can decompose the amplitude at Feynman diagram level, at amplitude level or in between

$$\mathcal{M}^{\text{1-loop}} = \sum_{i_0<i_1<i_2<i_3} d_{i_0 i_1 i_2 i_3} \text{Box}_{i_0 i_1 i_2 i_3} + \sum_{i_0<i_1<i_2} c_{i_0 i_1 i_2} \text{Triangle}_{i_0 i_1 i_2} + \sum_{i_0<i_1} b_{i_0 i_1} \text{Bubble}_{i_0 i_1} + \sum_{i_0} a_{i_0} \text{Tadpole}_{i_0} + R + \mathcal{O}(\epsilon)$$

Integrand reduction

- Express loop amplitude as sum of scalar integrals at integrand level
- Allows to calculate integral for a given phase space point by solving a system of equations
- Can decompose the amplitude at Feynman diagram level, at amplitude level or in between

$$\mathcal{M}^{\text{1-loop}} = \sum_{i_0<i_1<i_2<i_3} d_{i_0 i_1 i_2 i_3} \text{Box}_{i_0 i_1 i_2 i_3} + \sum_{i_0<i_1<i_2} c_{i_0 i_1 i_2} \text{Triangle}_{i_0 i_1 i_2} + \sum_{i_0<i_1} b_{i_0 i_1} \text{Bubble}_{i_0 i_1} + \sum_{i_0} a_{i_0} \text{Tadpole}_{i_0} + R + \mathcal{O}(\epsilon)$$

- The procedure is automated using the OPP [Ossola, Papadopoulos, Pittau 2006] reduction method in the CutTools program [arXiv:0711.3596]

Rational terms

Rational terms

- In the OPP method, the rational terms are split into two contributions, generally called

$$R = R_1 + R_2$$

Rational terms

- In the OPP method, the rational terms are split into two contributions, generally called

$$R = R_1 + R_2$$

- Both have their origin in the UV part of the model, but only R_1 can be directly computed in the OPP reduction and is given by the CutTools program

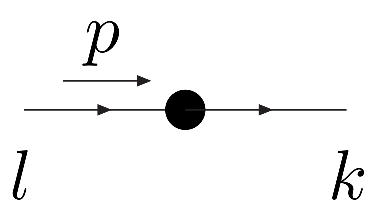
R_2 Feynman rules

R_2 Feynman rules

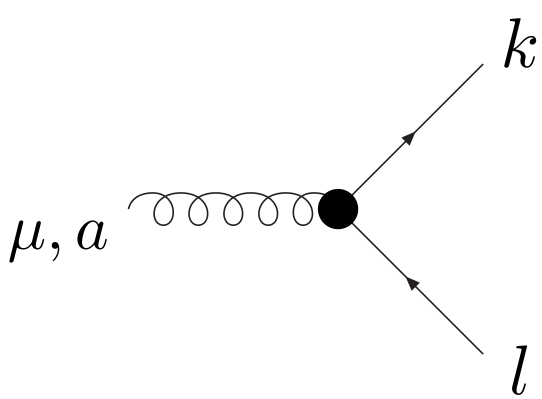
- Given that the R_2 contributions are of UV origin, only up to 4-point functions contribute to it (in a renormalizable theory)

R_2 Feynman rules

- Given that the R_2 contributions are of UV origin, only up to 4-point functions contribute to it (in a renormalizable theory)
- They can be computed using special Feynman rules, similarly to the UV counter term Feynman rules needed for the UV renormalization, e.g.



$$= \frac{ig^2}{16\pi^2}\frac{N_{col}^2-1}{2N_{col}}\delta_{kl}(-\not{p}+2m_q)\,\lambda_{HV}$$



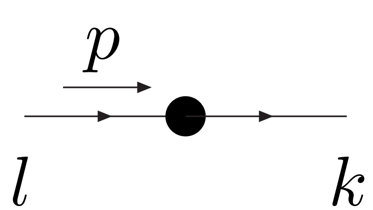
$$= \frac{ig^3}{16\pi^2}\frac{N_{col}^2-1}{2N_{col}}\,t^a_{kl}\gamma_\mu\,(1+\lambda_{HV})$$

[Draggiotis, Garzelli, Papadopoulos, Pittau]

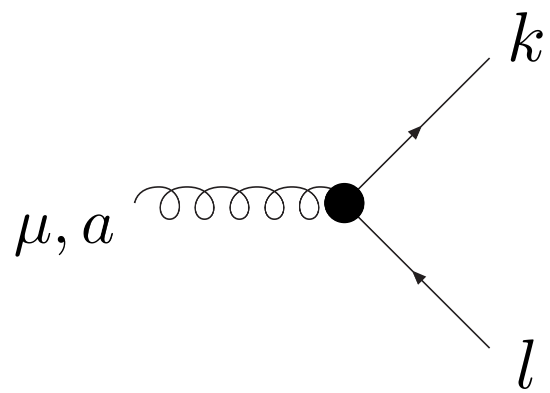
R_2 Feynman rules

- Given that the R_2 contributions are of UV origin, only up to 4-point functions contribute to it (in a renormalizable theory)

- They can be computed using special Feynman rules, similarly to the UV counter term Feynman rules needed for the UV renormalization, e.g.



$$= \frac{ig^2}{16\pi^2}\frac{N_{col}^2-1}{2N_{col}}\delta_{kl}(-\not{p}+2m_q)\,\lambda_{HV}$$



$$= \frac{ig^3}{16\pi^2}\frac{N_{col}^2-1}{2N_{col}}\,t^a_{kl}\gamma_\mu\,(1+\lambda_{HV})$$

[Draggiotis, Garzelli, Papadopoulos, Pittau]

- Unfortunately these Feynman rules are model dependent.
  ⇒ Work ongoing to use FeynRules+FeynArts to compute these terms, as well as the UV counter terms, for any model!

Background: MadGraph

[Stelzer, Long, 1994; Maltoni, Stelzer, 2002; Alwall et al, 2007; Alwall et al, 2011]

- ➡ MadGraph is an automated leading order matrix element generator and event generator
- ➡ Specify a collider or decay process in a simple syntax (download code locally or run online!)
  - ✓ `p p > l+ l- j j j j`
  - ✓
```
p p > go go, (go > sq q, sq > q n1) \
             (go > sq q, sq > l+ l- q n1)
```
- ➡ MadGraph automatically generates Feynman diagrams for all subprocesses, creates the source code to calculate cross sections and generate events, and performs optimized event generation (locally or online, in serial, multiprocessor or cluster parallelized mode)

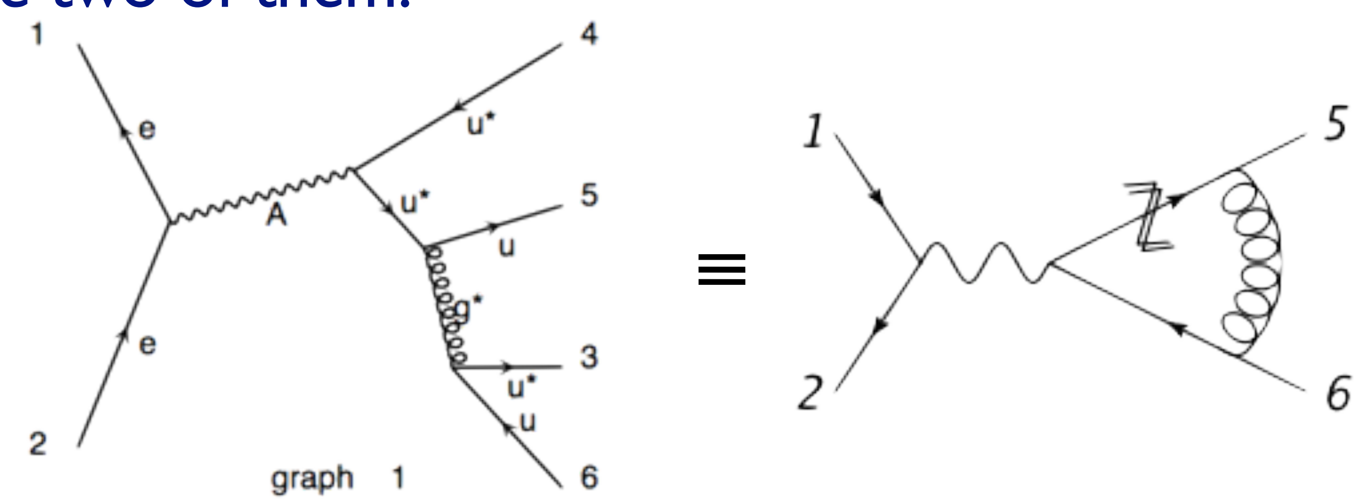
Loop amplitudes in MadGraph

Loop amplitudes in MadGraph

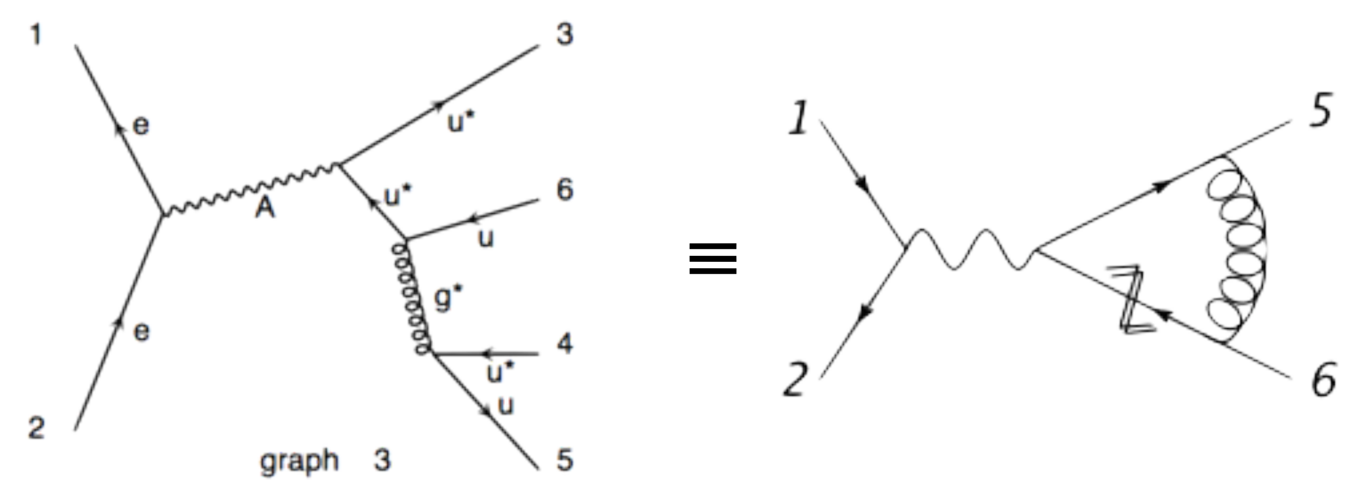
- Instead of writing a new code to generate loop diagrams, we use the existing, well-tested MadGraph code to generate tree-level diagrams with the loop cut open

Loop amplitudes in MadGraph

- Instead of writing a new code to generate loop diagrams, we use the existing, well-tested MadGraph code to generate tree-level diagrams with the loop cut open
- A loop diagrams with the loop cut open has to extra external particles. Consider $e^+e^- \to$ u* ubar* u ubar (loop particles are denoted with a star). MadGraph will generate 8 L-cut diagrams. Here are two of them:



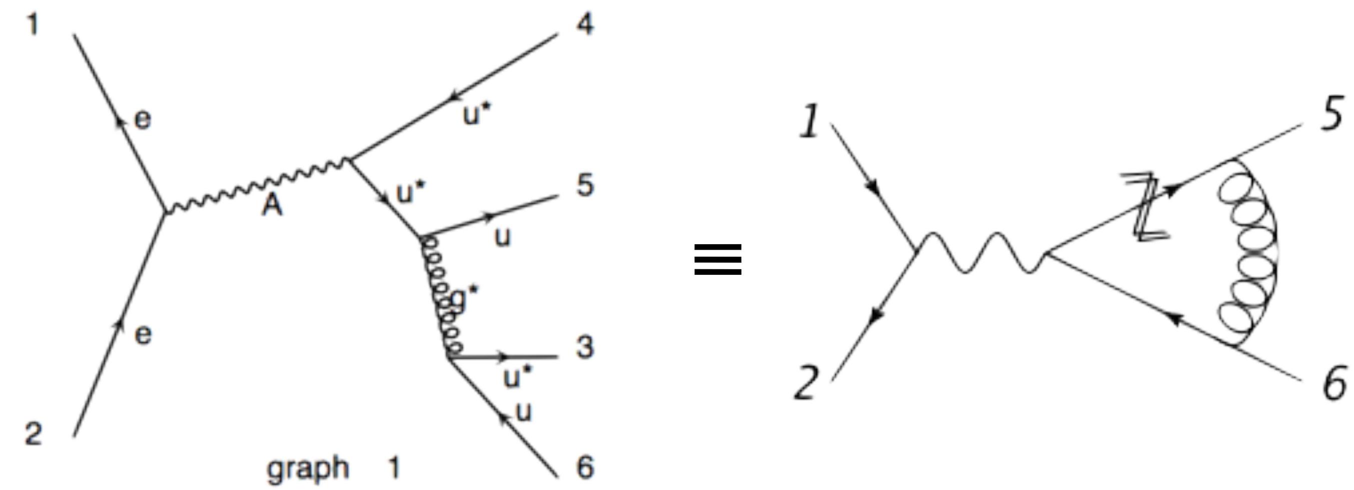
$$\text{Diag}_1 = [u^*(6)g^*(5)u^*(A)]$$



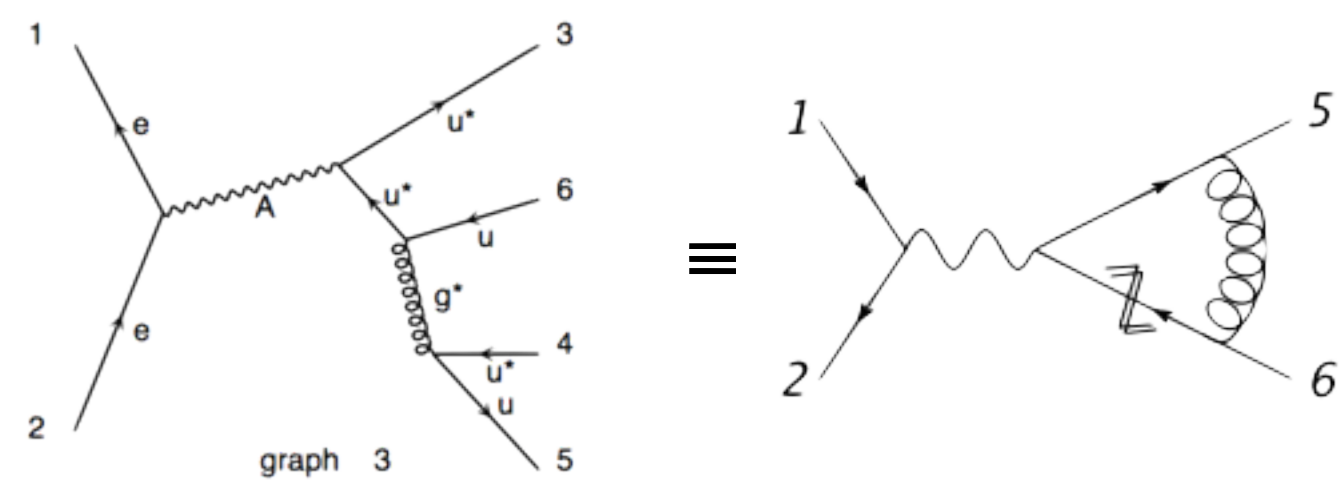
$$\text{Diag}_3 = [u^*(A)u^*(6)g^*(5)]$$

Loop amplitudes in MadGraph

- Instead of writing a new code to generate loop diagrams, we use the existing, well-tested MadGraph code to generate tree-level diagrams with the loop cut open
- A loop diagrams with the loop cut open has to extra external particles. Consider $e^+e^- \rightarrow$ u* ubar* u ubar (loop particles are denoted with a star). MadGraph will generate 8 L-cut diagrams. Here are two of them:
  - All diagrams with two extra particles are generated and the ones that are needed are filtered out
  - Each diagram gets an unique tag: any mirror and/or cyclic permutations of tags of diagrams already in the set are taken out
  - Additional filter to eliminate tadpoles and bubbles attached to external lines



$\text{Diag}_1 = [u^*(6)g^*(5)u^*(A)]$



$\text{Diag}_3 = [u^*(A)u^*(6)g^*(5)]$

Local checks

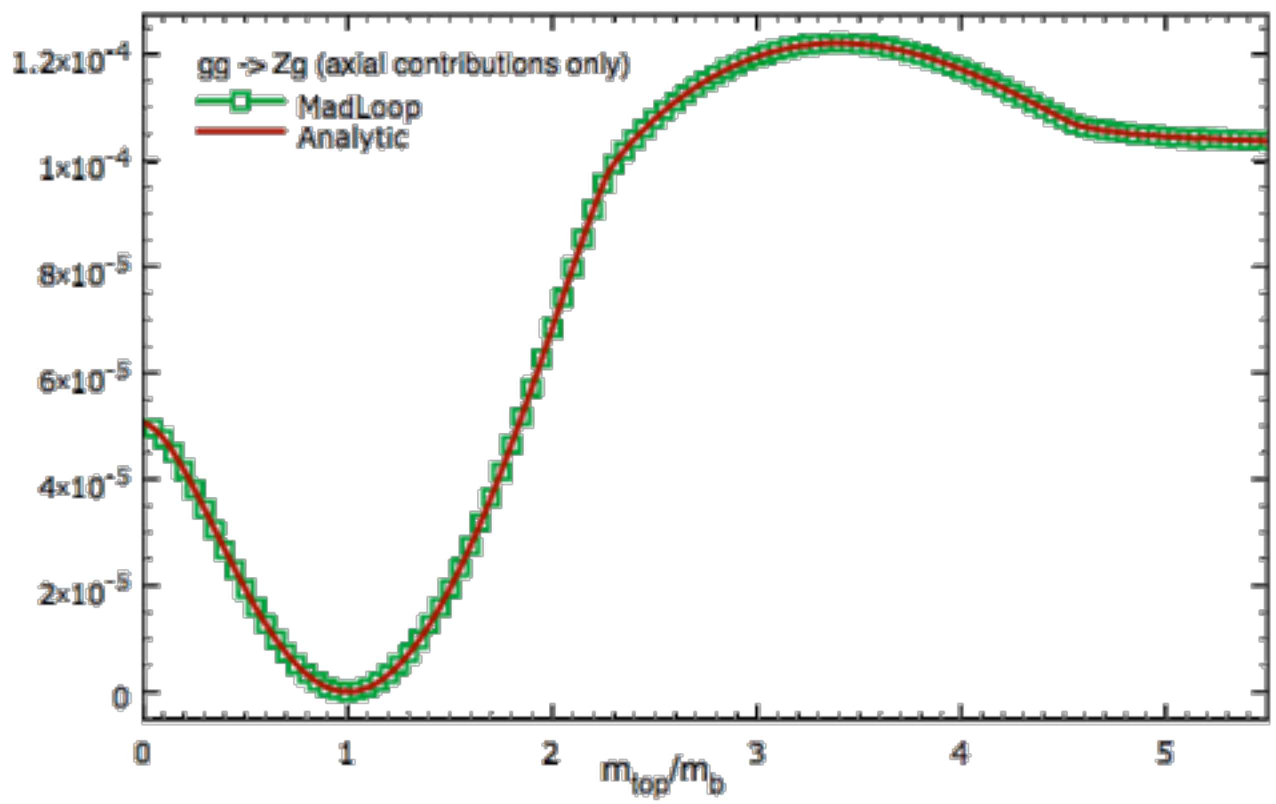
$u\bar{u} \to W^+W^-b\bar{b}$	MadLoop	Ref. [33]
a_0	2.338047209268890E-008	2.338047130649064E-008
c_{-2}	-2.493920703542680E-007	-2.493916939359002E-007
c_{-1}	-4.885901939046758E-007	-4.885901774740355E-007
c_0	-2.775800623041098E-007	-2.775787767591390E-007
$gg \to W^+W^-b\bar{b}$		
a_0	1.549795815702494E-008	1.549794572435312E-008
c_{-2}	-2.686312747217639E-007	-2.686310592221201E-007
c_{-1}	-6.078687041491385E-007	-6.078682316434646E-007
c_0	-5.519004042667462E-007	-5.519004727276688E-007

Ref. [33] : A. van Hameren *et al.* arXiv:0903.4665

- The code is very robust - e.g., MadLoop helped spot mistakes in published loop computations (Zjj , W^+W^+jj)

- The numerics are pin-point on analytical calculations, even with several mass scales.

- Analytic computations from an independent implementation of the helicity amplitudes by J.J van der Bij *et al.*



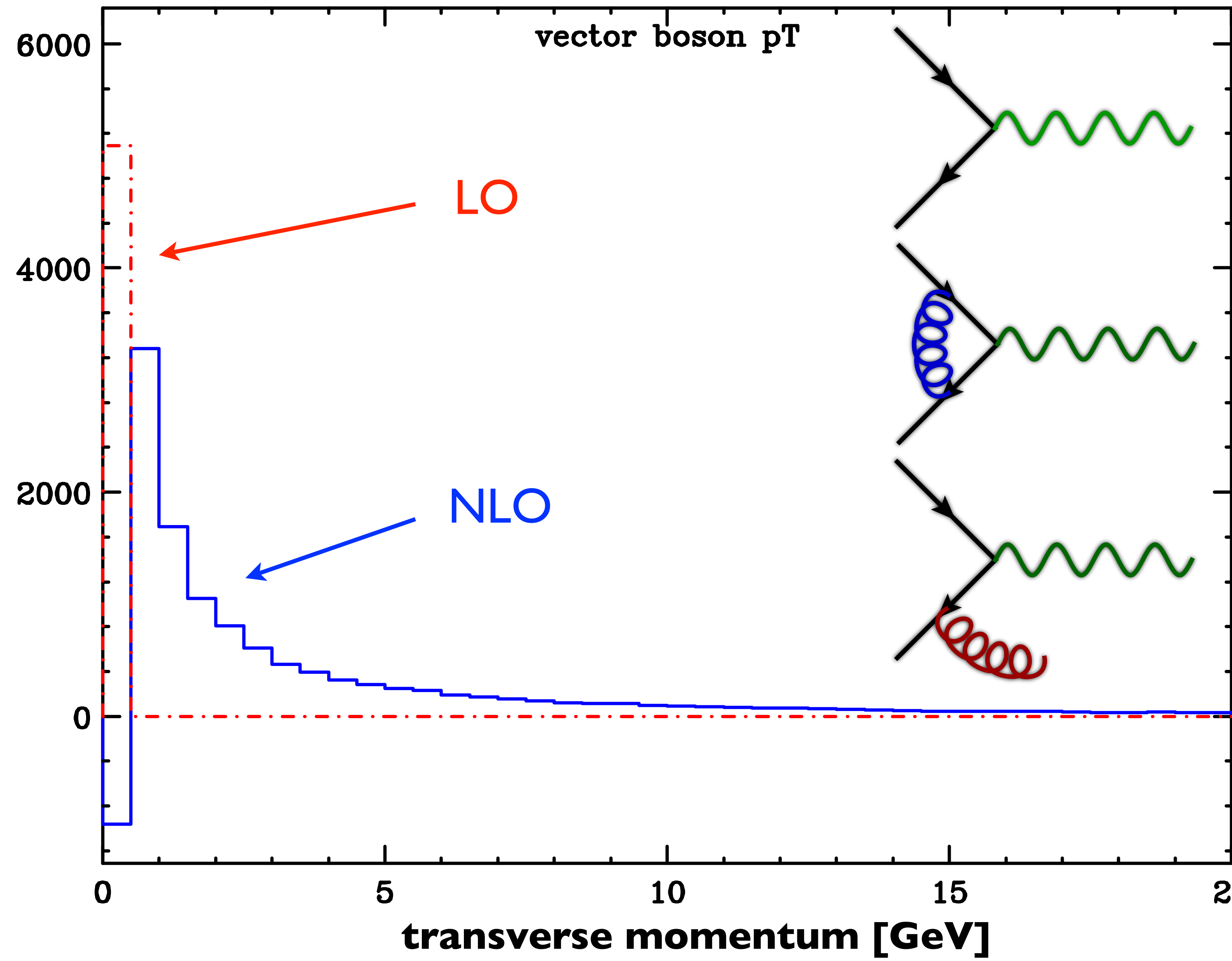
Integrated Results

- Running time: Two weeks on a 150+ node cluster
- Proof of efficient EPS handling with $Zt\bar{t}$
- Successful cross-check against known results
- Sometimes large K-factors
- No cuts on b, robust numerics with small P_T

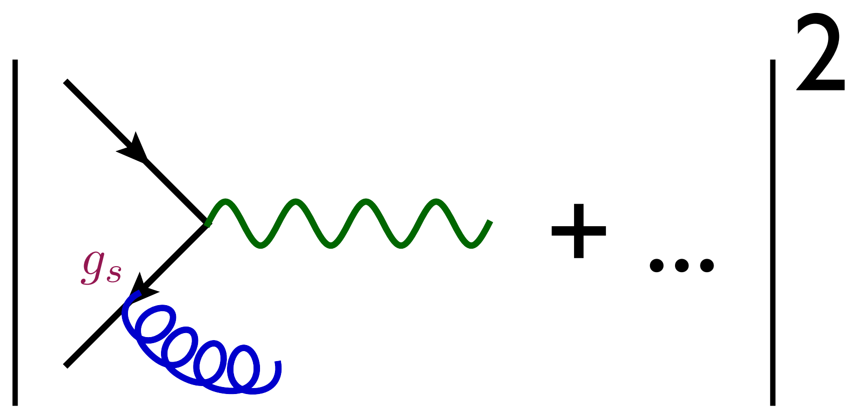
	Process	μ	n_{lf}	Cross section (pb) LO	Cross section (pb) NLO
a.1	$pp \to t\bar{t}$	m_{top}	5	123.76 ± 0.05	162.08 ± 0.12
a.2	$pp \to tj$	m_{top}	5	34.78 ± 0.03	41.03 ± 0.07
a.3	$pp \to tjj$	m_{top}	5	11.851 ± 0.006	13.71 ± 0.02
a.4	$pp \to t\bar{b}j$	$m_{top}/4$	4	25.62 ± 0.01	30.96 ± 0.06
a.5	$pp \to t\bar{b}jj$	$m_{top}/4$	4	8.195 ± 0.002	8.91 ± 0.01
b.1	$pp \to (W^+ \to) e^+ \nu_e$	m_W	5	5072.5 ± 2.9	6146.2 ± 9.8
b.2	$pp \to (W^+ \to) e^+ \nu_e j$	m_W	5	828.4 ± 0.8	1065.3 ± 1.8
b.3	$pp \to (W^+ \to) e^+ \nu_e jj$	m_W	5	298.8 ± 0.4	300.3 ± 0.6
b.4	$pp \to (\gamma^*/Z \to) e^+ e^-$	m_Z	5	1007.0 ± 0.1	1170.0 ± 2.4
b.5	$pp \to (\gamma^*/Z \to) e^+ e^- j$	m_Z	5	156.11 ± 0.03	203.0 ± 0.2
b.6	$pp \to (\gamma^*/Z \to) e^+ e^- jj$	m_Z	5	54.24 ± 0.02	56.69 ± 0.07
c.1	$pp \to (W^+ \to) e^+ \nu_e b\bar{b}$	$m_W + 2m_b$	4	11.557 ± 0.005	22.95 ± 0.07
c.2	$pp \to (W^+ \to) e^+ \nu_e t\bar{t}$	$m_W + 2m_{top}$	5	0.009415 ± 0.000003	0.01159 ± 0.00001
c.3	$pp \to (\gamma^*/Z \to) e^+ e^- b\bar{b}$	$m_Z + 2m_b$	4	9.459 ± 0.004	15.31 ± 0.03
c.4	$pp \to (\gamma^*/Z \to) e^+ e^- t\bar{t}$	$m_Z + 2m_{top}$	5	0.0035131 ± 0.0000004	0.004876 ± 0.000002
c.5	$pp \to \gamma t\bar{t}$	$2m_{top}$	5	0.2906 ± 0.0001	0.4169 ± 0.0003
d.1	$pp \to W^+W^-$	$2m_W$	4	29.976 ± 0.004	43.92 ± 0.03
d.2	$pp \to W^+W^- j$	$2m_W$	4	11.613 ± 0.002	15.174 ± 0.008
d.3	$pp \to W^+W^+ jj$	$2m_W$	4	0.07048 ± 0.00004	0.1377 ± 0.0005
e.1	$pp \to HW^+$	$m_W + m_H$	5	0.3428 ± 0.0003	0.4455 ± 0.0003
e.2	$pp \to HW^+ j$	$m_W + m_H$	5	0.1223 ± 0.0001	0.1501 ± 0.0002
e.3	$pp \to HZ$	$m_Z + m_H$	5	0.2781 ± 0.0001	0.3659 ± 0.0002
e.4	$pp \to HZj$	$m_Z + m_H$	5	0.0988 ± 0.0001	0.1237 ± 0.0001
e.5	$pp \to Ht\bar{t}$	$m_{top} + m_H$	5	0.08896 ± 0.00001	0.09869 ± 0.00003
e.6	$pp \to Hb\bar{b}$	$m_b + m_H$	4	0.16510 ± 0.00009	0.2099 ± 0.0006
e.7	$pp \to Hjj$	m_H	5	1.104 ± 0.002	1.036 ± 0.002

Need for parton shower

- Consider Drell-Yan production
- What happens if we plot the transverse momentum of the vector boson?
- Both the LO and the NLO distributions are non-physical
- We need resummation of any number of parton emissions via a parton shower



At NLO

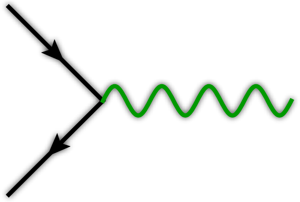


- We have to integrate the real emission over the **complete** phase-space of the one particle that can go soft or collinear to obtain the infra-red poles that will cancel against the virtual corrections
- We cannot use a LO matching procedure: requiring that all partons should produce separate jets is not infrared safe
- We have to invent a new procedure to match NLO matrix elements with parton showers

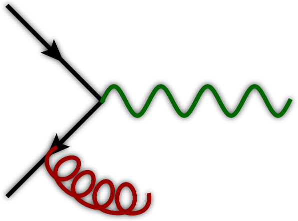
Sources of double counting

Parton shower

Born+Virtual:



Real emission:

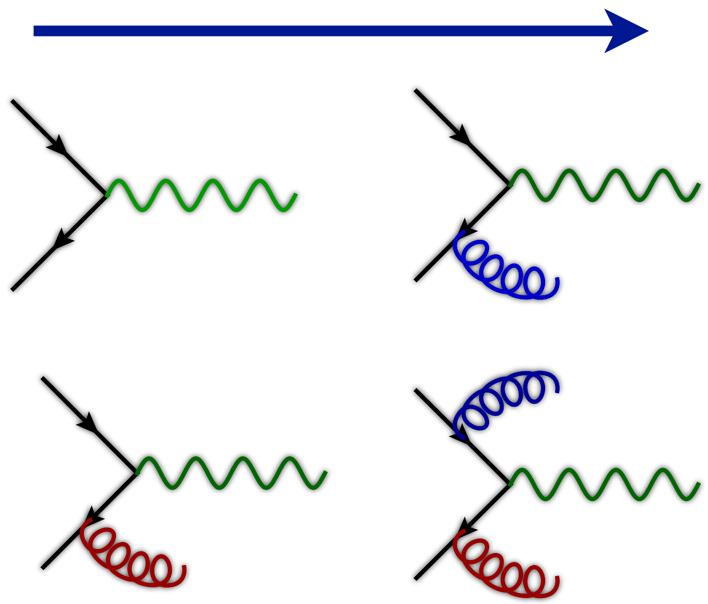


Sources of double counting

Parton shower

Born+Virtual:

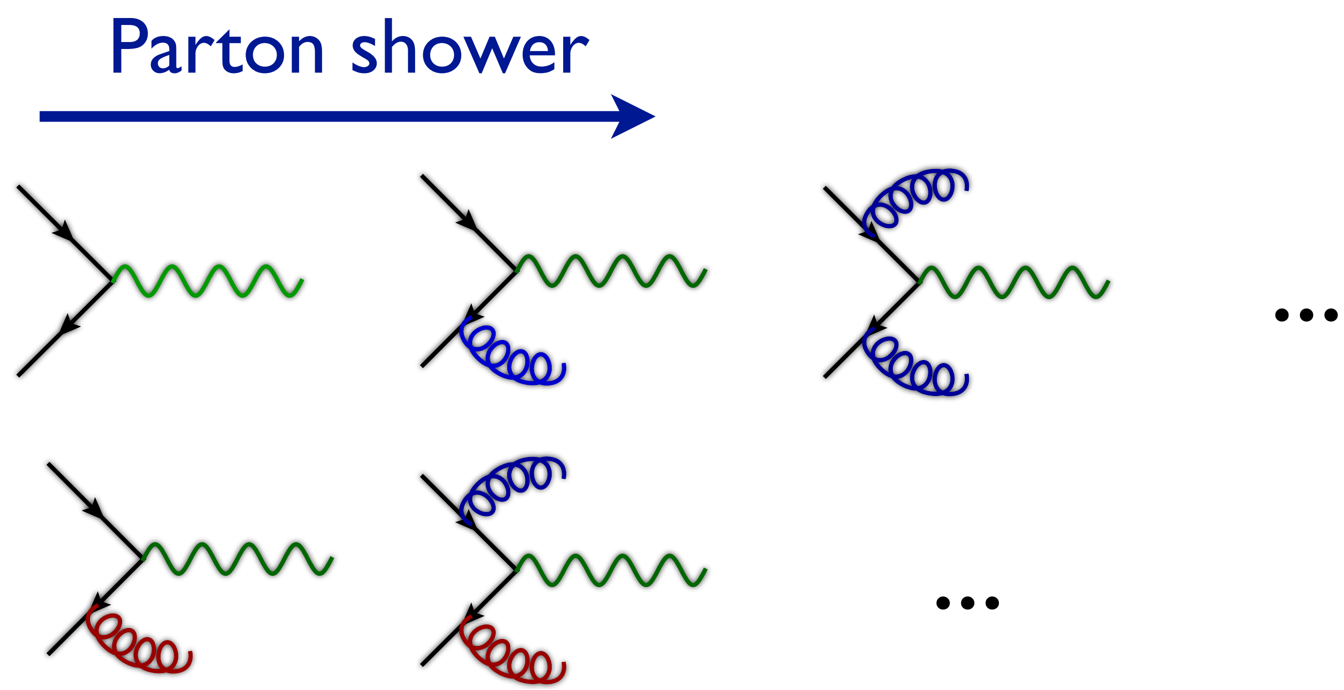
Real emission:



Sources of double counting

Born+Virtual:

Real emission:

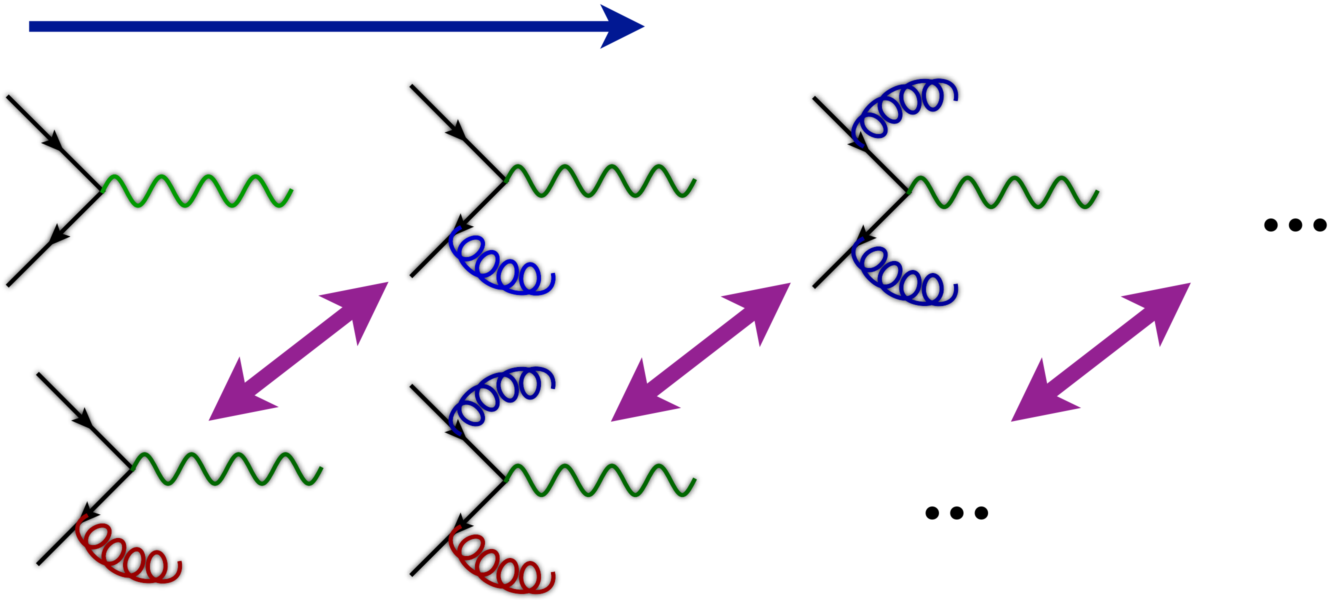


Sources of double counting

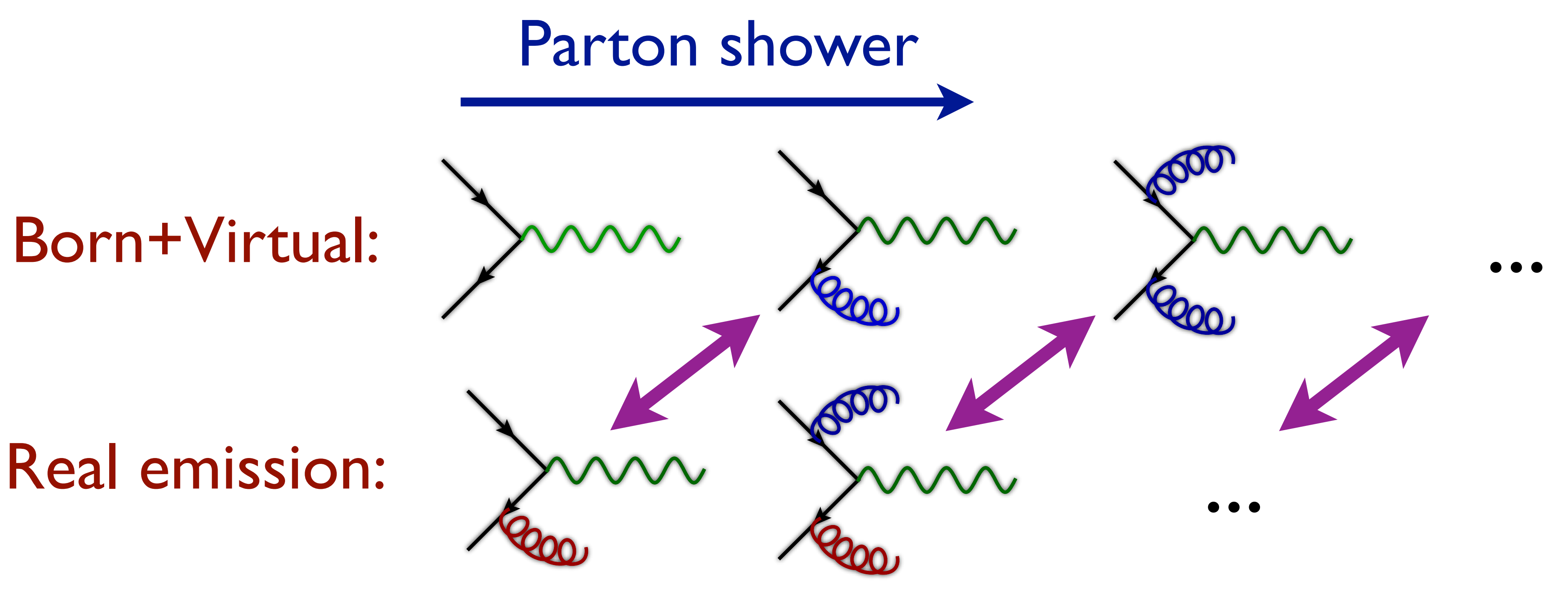
Parton shower

Born+Virtual:

Real emission:



Sources of double counting



- There is double counting between the real emission matrix elements and the parton shower: the extra radiation can come from the matrix elements or the parton shower
- There is also an overlap between the virtual corrections and the Sudakov suppression in the zero-emission probability

MC@NLO procedure

[Frixione, Webber]

- To remove the double counting, we can add and subtract the same term to the *m* and *m*+1 body configurations

$$\frac{d\sigma_{\mathrm{NLOwPS}}}{dO} = \left[d\Phi_m \left(B + \int_{\mathrm{loop}} V + \int d\Phi_1 MC\right)\right] I_{\mathrm{MC}}^{(m)}(O) + \left[d\Phi_{m+1} (R - MC) \right] I_{\mathrm{MC}}^{(m+1)}(O)$$

Where the *MC* are defined to be the contribution of the parton shower to get from the *m* body Born final state to the *m*+1 body real emission final state

MC@NLO properties

- Good features of including the subtraction counter terms
  1. **Double counting avoided**: The rate expanded at NLO coincides with the total NLO cross section
  2. **Smooth matching**: MC@NLO coincides (in shape) with the parton shower in the soft/collinear region, while it agrees with the NLO in the hard region
  3. **Stability**: weights associated to different multiplicities are separately finite. The *MC* term has the same infrared behavior as the real emission (there is a subtlety for the soft divergence)
- Not so nice feature:
  1. **Parton shower dependence**: the form of the *MC* terms depends on what the parton shower does exactly. Need special subtraction terms for each parton shower to which we want to match

Workflow for NLO in MadGraph

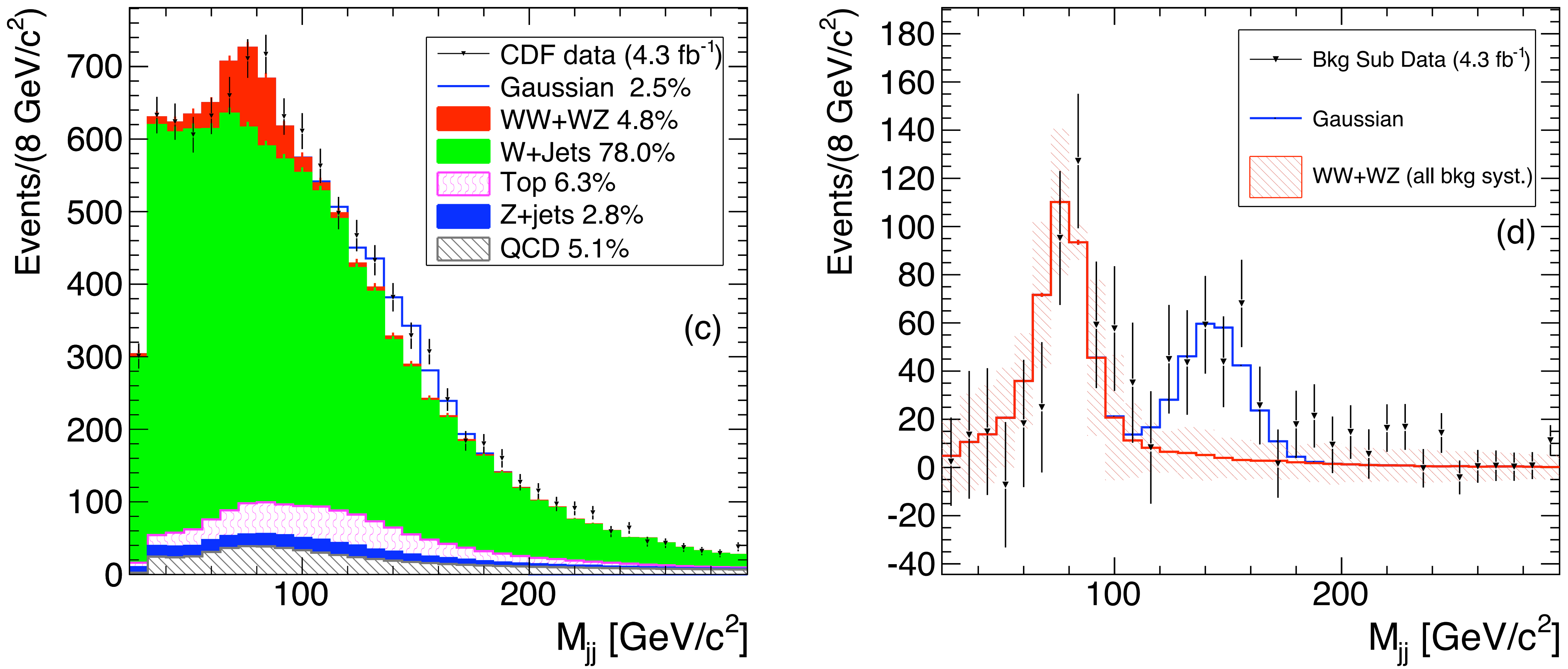
R. Frederix, S. Frixione, V. Hirschi, F. Maltoni, R. Pittau, P. Torrielli [arXiv:1104.5613]

- MadFKS computes all contributions to the NLO computation, except for the finite part of the virtual amplitude
- MadLoop computes the virtual corrections to any process in the SM using the OPP method as implemented in CutTools
- Combine MadFKS and MadLoop to get any distribution/cross section at (parton-level) NLO accuracy
- Add terms to remove double counting when matching to the parton shower: a(utomatic)MC@NLO
- Shower the generated events using Herwig or Pythia to get fully exclusive predictions at NLO accuracy (for IR-safe observables).

Results using aMC@NLO

- Recently published results (all within ~6 months) using aMC@NLO:
  - (Pseudo-)scalar Higgs production in association with a top-antitop pair
    [*Frederix, Frixione, Hirschi, Maltoni, Pittau & Torrielli, arXiv:1104.5613*]
  - Vector boson production in association with a bottom-antibottom pair
    [*Frederix, Frixione, Hirschi, Maltoni, Pittau & Torrielli, arXiv:1106.6019*]
  - Four charged lepton production at hadron colliders
    [*Frederix, Frixione, Hirschi, Maltoni, Pittau & Torrielli, arXiv:1110.4738*]
  - Wjj at the Tevatron
    [*Frederix, Frixione, Hirschi, Maltoni, Pittau & Torrielli, arXiv:1110.5502*]

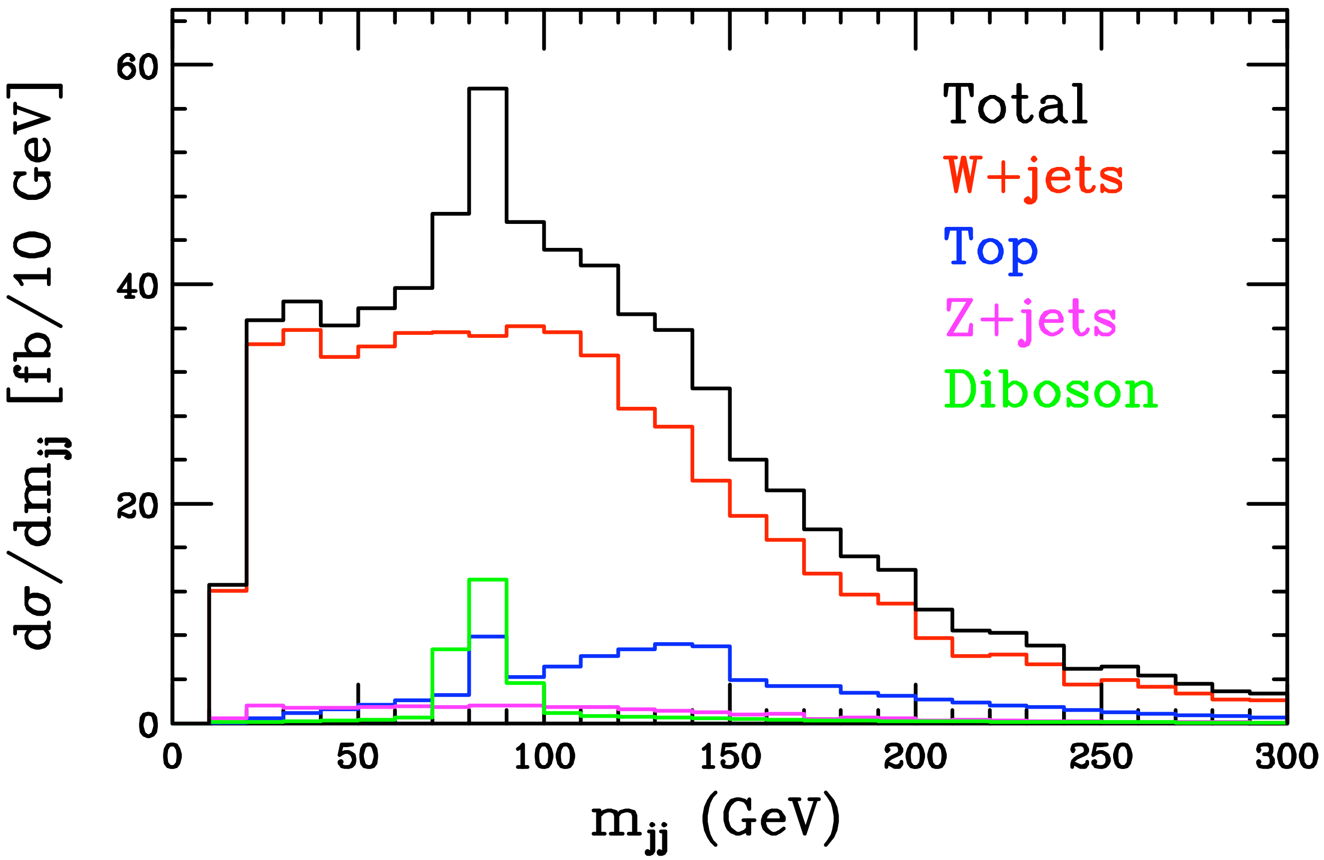
Wjj at CDF



- In April CDF reported an excess of events with 3.2 standard deviation significance in the dijet invariant mass distribution (with invariant mass 130-160 GeV) for Wjj events
- The update in June (using 7.3 fb^{-1} of data) increased significance of the excess to 4.1 standard deviations

NLO effects

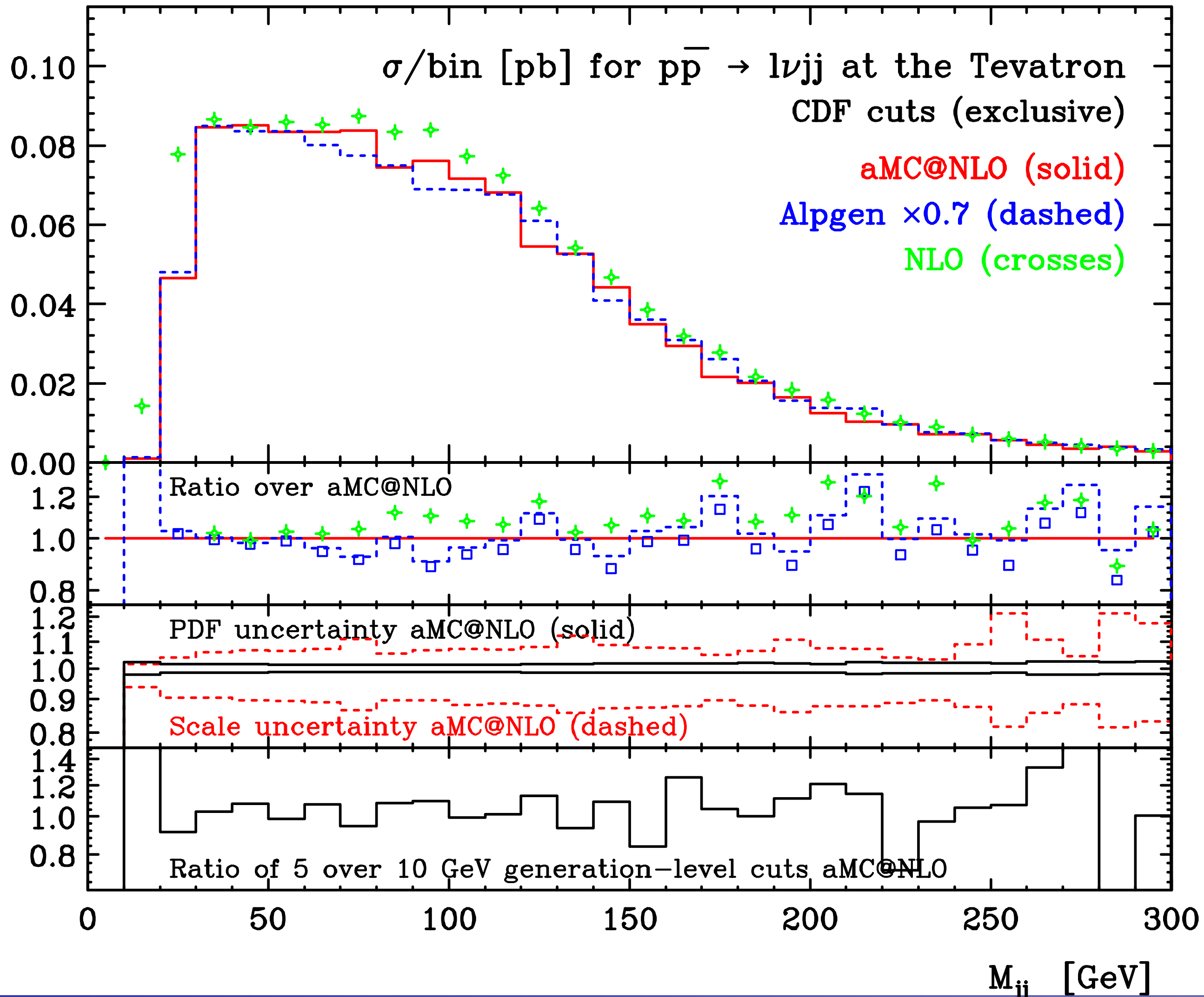
- Both CDF and DØ estimate their backgrounds using LO SMC programs (Alpgen+Pythia & Sherpa) normalized to (N)NLO or to the data
- J. Campbell, A. Martin & C. Williams have looked at the same distribution at parton level to study the impact of NLO corrections on differential distributions

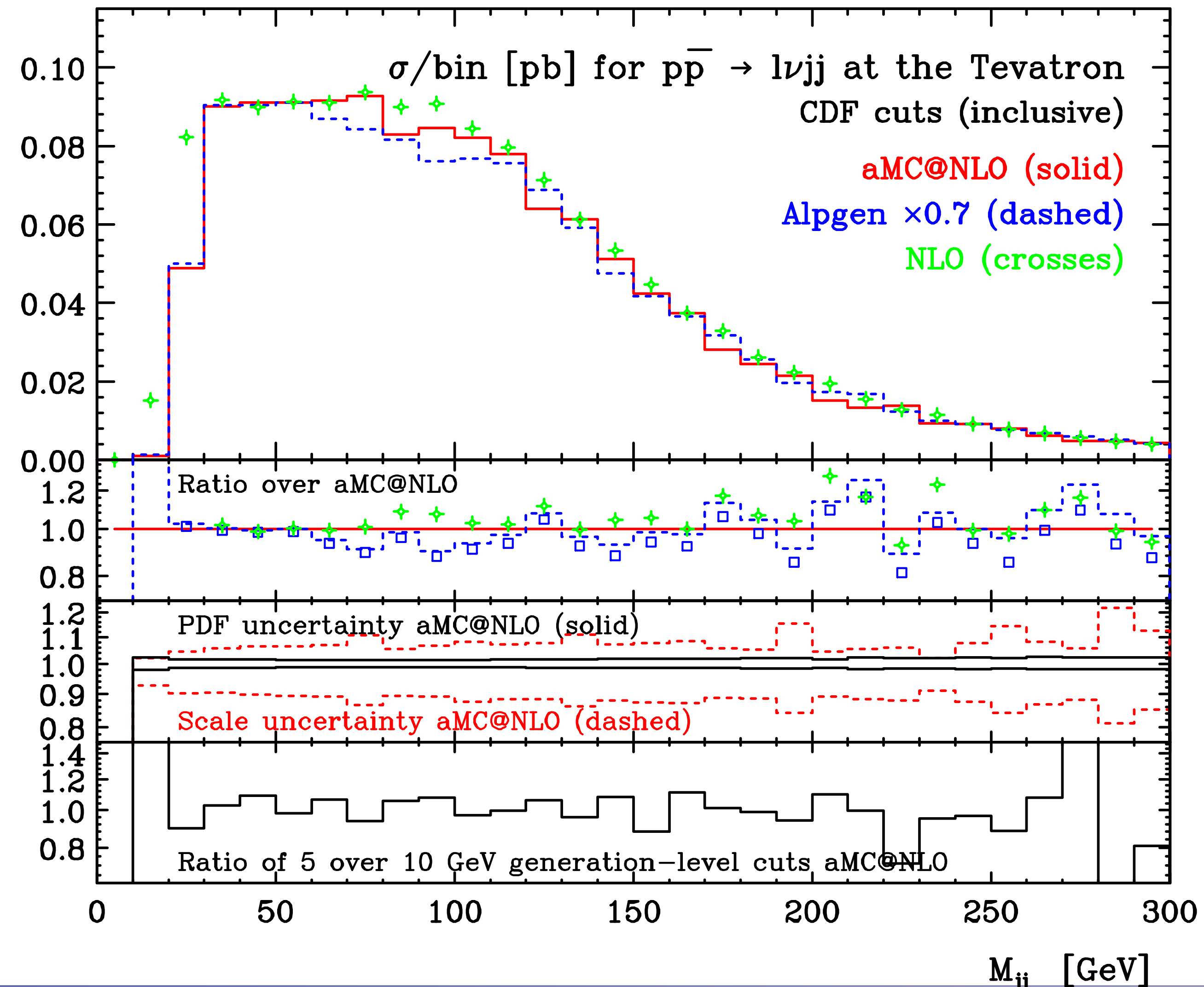


- Using aMC@NLO, we could address the main background, W+2j, at the NLOwPS level to see how well LOwPS or fixed order NLO describe this distribution

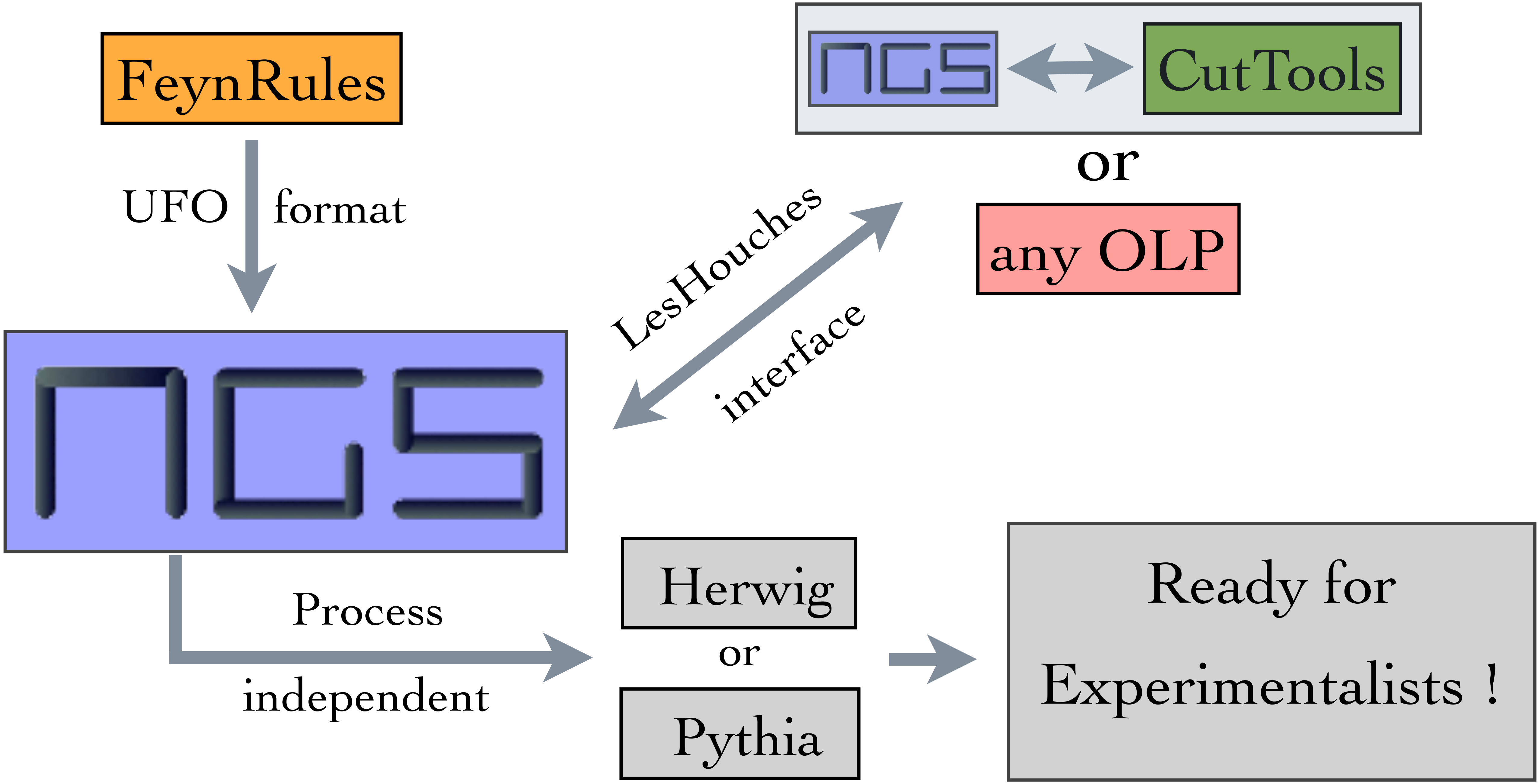
$p\bar{p} \rightarrow Wjj$: Dijet invariant mass

- Dijet invariant mass with/without jet veto
- This is the distribution in which CDF found an excess of events around 130-160 GeV
- No differences in shape between the 5 and 10 GeV generation level cuts
- No sign of enhancement over (N)LO or LOwPS in the mass range 130-160 GeV





Going ahead - Towards a fast, public version



MadGraph 5

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

- Complete rewrite of the old (leading order) MadGraph using the Python programming language
- Order of magnitude improvements of
  - Process generation speed
  - Event generation speed
  - Stability of results
  - Modularity and extensibility
- Any process from ANY Lagrangian-based model (by FeynRules +UFO/ALOHA)
- Fast and reliable simulation of completely new classes of processes including unlimited-length decay chains, multiparton processes, etc.

MadFKS: From v4 to v5

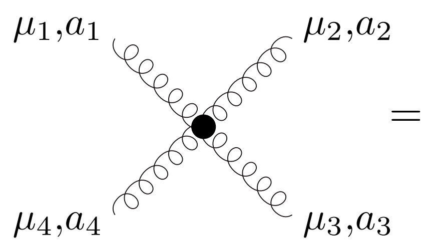
J.A., Frederix, Zaro

- By rewriting MadFKS in MadGraph 5, we will greatly improve speed, efficiency and flexibility:
  - Faster matrix elements thanks to more efficient diagram generation
  - Group processes with similar pole structure to reduce number of integration channels
  - Faster and more flexible color algebra
  - Allow any model (that can be written as a Lagrangian) from FeynRules
  - Take advantage of ongoing implementation of color-ordered recursion relations for fast matrix element calculation

MadLoop: From v4 to v5

J.A., Hirschi

- Limitations on the MadGraph 4 MadLoop code:
  - ➡ No four-gluon vertex at the Born level: the special vertex to compute the remainder is too complicated to implement in MadGraph v4



$$= -\frac{i g^4 N_{col}}{96\pi^2} \sum_{P(234)} \Big\{ \Big[\frac{\delta_{a_1a_2}\delta_{a_3a_4}+\delta_{a_1a_3}\delta_{a_4a_2}+\delta_{a_1a_4}\delta_{a_2a_3}}{N_{col}} + 4\,Tr(t^{a_1}t^{a_3}t^{a_2}t^{a_4}+t^{a_1}t^{a_4}t^{a_2}t^{a_3})\,(3+\lambda_{HV}) - Tr(\{t^{a_1}t^{a_2}\}\{t^{a_3}t^{a_4}\})\,(5+2\lambda_{HV})\Big]\, g_{\mu_1\mu_2}g_{\mu_3\mu_4} + 12\frac{N_f}{N_{col}} Tr(t^{a_1}t^{a_2}t^{a_3}t^{a_4}) \left(\frac{5}{3} g_{\mu_1\mu_3}g_{\mu_2\mu_4} - g_{\mu_1\mu_2}g_{\mu_3\mu_4} - g_{\mu_2\mu_3}g_{\mu_1\mu_4}\right)\Big\}$$

  - ➡ For EW bosons in the loops, the reduction by CutTools might not work because of gauge choice (rank of diagrams can become too large)
  - ➡ No finite-width effects for massive particles also appearing in the loops
  - ➡ All Born contributions must factorize the same power of all coupling orders

MadLoop: From v4 to v5

J.A., Hirschi

- The MadGraph 5 implementation:
  - ➡ removing all present limitations of the code
  - ➡ making it faster:
    - Recycling of tree-structures attached to the loops
    - Identify identical contributions (e.g. massless fermion loops of different flavors)
    - Call CutTools not per diagram, but per set of diagrams with the same loop kinematics
    - Use recursion relations for multi-parton amplitudes
  - ➡ Allowing for the automatic generation of UV renormalization and R_2 vertices using FeynRules [*Christensen, Duhr et al.*] for BSM physics

Final words

- aMC@NLO shows that an experimental analysis fully at NLO done without theory support is no longer science fiction!
- Fully automated parton-level NLO calculation with MadLoop+MadFKS has been tested against literature for over 30 (very) complex processes
- Several (fully automated) completely new physics results already published using MadLoop+MadFKS+aMC@NLO
- Expect fast, public version (in MG5) within a few months!
- Find us at:
  - http://amcatnlo.cern.ch/
  - http://madgraph.hep.uiuc.edu/

Additional Slides

FKS subtraction

Frixione, Kunszt & Signer 1996

Frederix, Frixione, Maltoni, Stelzer arXiv:0908.4272

- Real emission part : $d\sigma^R = |M^{n+1}|^2 d\phi_{n+1}$
- $|M^{n+1}|^2$ diverges as $\frac{1}{\chi_i^2}\frac{1}{1-y_{ij}}$ with $\begin{aligned}\chi_i &= \frac{E_i}{\sqrt{\hat{s}}}\\ y_{ij} &= \cos\theta_{ij}\end{aligned}$
- Divide phase-space so that each partition has at most one soft and one collinear singularity

$$d\sigma^R = \sum_{ij} S_{ij}|M^{n+1}|^2 d\phi_{n+1} \qquad \sum_{ij} S_{ij} = 1$$

- Use plus distribution to regulate the singularities $\int d\chi \left(\frac{1}{\chi}\right)_+ f(\chi) = \int d\chi \frac{f(\chi)-f(0)}{\chi}$

$$d\tilde{\sigma}^R = \sum_{ij}\left(\frac{1}{\chi_i}\right)_+ \left(\frac{1}{1-y_{ij}}\right)_+ \chi_i^2(1-y_{ij})S_{ij}|M^{n+1}|^2 d\phi_{n+1}$$

FKS vs CS dipoles

FKS vs CS dipoles

- CS uses soft singularities to organize the subtractions :
  - → Three-body kernels, so naive n^3 scaling
  - → Each subtraction term has a different kinematics
  - → All subtraction terms must be subtracted from $\mathcal{M}^{(r)}$

FKS vs CS dipoles

- CS uses soft singularities to organize the subtractions :
  - → Three-body kernels, so naive n^3 scaling
  - → Each subtraction term has a different kinematics
  - → All subtraction terms must be subtracted from $\mathcal{M}^{(r)}$
- FKS, based on the collinear structures :
  - → The majority of the subtractions can be grouped together.
    Ex: The 2 → N gluons process as 3 subtractions ∀N
  - → Soft and collinear counter-terms can be defined to have the same kinematics so that the subtraction term is unique.
  - → The collinear structure is better suited to existing formalisms for parton shower matching @NLO.
- Model- and process-independent implementation: MadFKS [0908.4247]

Existing public tools

- Public, flexible tools for NLO predictions do not exist:
  - MCFM *[Campbell & Ellis & ...]* has it available almost all relevant process for background studies at the Tevatron and LHC, but gives only fixed-order, parton-level results
  - MC@NLO *[Frixione & Webber & ...]* has matching to the parton shower to describe fully exclusive final states, but the list of available processes is relatively short
  - POWHEG BOX *[Nason et al.]* provides a framework to match any existing parton level NLO computation to a parton shower. However, the NLO computation is not automated and some work by the user is needed to implement a new process
- Idea: write an automatic tool that is flexible and allows for any process to be computed at NLO accuracy, including matching to the parton shower to deliver events ready for experimentalists → **a**MC@NLO

OPP decomposition

$$\int d^d l \, \frac{N(l)}{D_0 D_1 D_2 \cdots D_{m-1}}$$

$$D_i = (l + p_i)^2 - m_i^2$$

- For the numerator of any integrand of a one-loop computation we can write

$$\begin{aligned} N(l) = & \sum_{i_0<i_1<i_2<i_3}^{m-1} \left[d_{i_0 i_1 i_2 i_3} + \tilde{d}_{i_0 i_1 i_2 i_3}(l) \right] \prod_{i \neq i_0, i_1, i_2, i_3}^{m-1} D_i \\ & + \sum_{i_0<i_1<i_2}^{m-1} \left[c_{i_0 i_1 i_2} + \tilde{c}_{i_0 i_1 i_2}(l) \right] \prod_{i \neq i_0, i_1, i_2}^{m-1} D_i \\ & + \sum_{i_0<i_1}^{m-1} \left[b_{i_0 i_1} + \tilde{b}_{i_0 i_1}(l) \right] \prod_{i \neq i_0, i_1}^{m-1} D_i \\ & + \sum_{i_0}^{m-1} \left[a_{i_0} + \tilde{a}_{i_0}(l) \right] \prod_{i \neq i_0}^{m-1} D_i \\ & + \tilde{P}(l) \prod_{i}^{m-1} D_i \end{aligned}$$

with the "~" coefficients being "spurious" terms of known functional form, that integrate to 0 [del Aguila, Pittau 2004]

How it works...

- For each phase-space point we have to solve a system of equations. This is done automatically by the CutTools program [arXiv:0711.3596]
- The system greatly reduces when picking special values for the loop momentum (where some D terms are 0)
- We can decompose the system at the level of the amplitude, diagram or in between, as long as we provide the corresponding numerator function. In MadGraph 4 we decompose diagram by diagram
- For a given phase-space point, CutTools will call the numerator function several times (~50 or so for a 2 → 3 process)

Comparison with Passarino-Veltman

Comparison with Passarino-Veltman

- In PV reduction, we need analytic expressions for all the integrals. Possible to automate, but in practice too many terms which are difficult to simplify ☺☹

Comparison with Passarino-Veltman

- In PV reduction, we need analytic expressions for all the integrals. Possible to automate, but in practice too many terms which are difficult to simplify ☺☹
- In OPP reduction we reduce the system at the **integrand** level.

Comparison with Passarino-Veltman

- In PV reduction, we need analytic expressions for all the integrals. Possible to automate, but in practice too many terms which are difficult to simplify ☺☹
- In OPP reduction we reduce the system at the **integrand** level.
    - We can solve the system numerically: we only need a numerical function of the (numerator of) integrand. We can set-up a system of linear equations by choosing specific values for the loop momentum *l*, depending on the kinematics of the event ☺

Comparison with Passarino-Veltman

- In PV reduction, we need analytic expressions for all the integrals. Possible to automate, but in practice too many terms which are difficult to simplify ☺☹
- In OPP reduction we reduce the system at the **integrand** level.
    - We can solve the system numerically: we only need a numerical function of the (numerator of) integrand. We can set-up a system of linear equations by choosing specific values for the loop momentum *l*, depending on the kinematics of the event ☺
    - Choosing *l* such that internal propagators go on-shell enormously simplifies the resulting system ☺

Comparison with Passarino-Veltman

- In PV reduction, we need analytic expressions for all the integrals. Possible to automate, but in practice too many terms which are difficult to simplify ☺☹
- In OPP reduction we reduce the system at the **integrand** level.
  - We can solve the system numerically: we only need a numerical function of the (numerator of) integrand. We can set-up a system of linear equations by choosing specific values for the loop momentum *l*, depending on the kinematics of the event ☺
  - Choosing *l* such that internal propagators go on-shell enormously simplifies the resulting system ☺
  - OPP reduction is implemented in CutTools (publicly available). Given the integrand, CutTools provides all the coefficients in front of the scalar integrals and the R_1 term ☺

Comparison with Passarino-Veltman

- In PV reduction, we need analytic expressions for all the integrals. Possible to automate, but in practice too many terms which are difficult to simplify ☺☹
- In OPP reduction we reduce the system at the **integrand** level.
  - We can solve the system numerically: we only need a numerical function of the (numerator of) integrand. We can set-up a system of linear equations by choosing specific values for the loop momentum *l*, depending on the kinematics of the event ☺
  - Choosing *l* such that internal propagators go on-shell enormously simplifies the resulting system ☺
  - OPP reduction is implemented in CutTools (publicly available). Given the integrand, CutTools provides all the coefficients in front of the scalar integrals and the R_1 term ☺
  - Analytic information is needed for the R_2 term, but can be computed once and for all for a given model ☹☺

MadLoop

- Several new features needed to be implemented in MadGraph (v4)
  - Recognition of the loop topologies in order to filter L-cut diagrams
  - Structure to deal with two MadGraph processes simultaneously (L-cut and Born-like)
  - Treat the color to obtain the correct interference between the Born and the loop diagrams
  - Special form of the integrand for CutTools: no propagator denominators, complex momenta and reconstruction of the missing propagator for sewed particles (e.g., when L-cut particle is a gluon, $\sum \epsilon^{\mu}(p)\epsilon^{\nu}(p) \rightarrow g^{\mu\nu}$)
  - Implementation of QCD ghosts
  - Implementation of the special vertices for the rational part R_2 and the UV renormalization

Exceptional phase-space points

- CutTools can asses the numerical stability of the computation of a loop by

➥ By sending $m_i^2 \to m_i^2 + M^2$, CT has an independent reconstruction of the numerator and can check if both match.
➥ CT ask MadLoop to evaluate the integrand at a given loop momentum and check if the result is close enough to the one from the reconstructed integrand.

- When an EPS occurs, MadLoop tries to cure it:

➥ Check if Ward Identities hold at a satisfactory level

➥ Shift the PS point by rescaling momenta : $k_i^3 = (1 + \lambda_\pm) k_i^3$

➥ Provide an estimate of the virtual for the original PS point with uncertainty:

$$v_{\lambda_\pm}^{FIN} = \frac{V_{\lambda_\pm}^{FIN}}{|\mathcal{A}_{\lambda=0}^{born}|^2} \quad c = \frac{1}{2}\left(v_{\lambda_+}^{FIN} + v_{\lambda_-}^{FIN}\right) \quad \Delta = \left|v_{\lambda_+}^{FIN} - v_{\lambda_-}^{FIN}\right| \quad V_{\lambda=0}^{FIN} = \left|\mathcal{A}_{\lambda=0}^{born}\right|^2 (c \pm \Delta)$$

➥ If nothing works, then use the median of the results of the last 100 stable points

MadLoop V4 to V5 (present status)

✓ = non-optimal | ✓ = done optimally | ✕ = not done | ✕ = not done YET

Task	MadLoop V4	MadLoop V5
Generation of L-Cut diagrams, loop-basis selection	✓-	✓++
Drawing of Loop diagrams	✕	✓
Full SM implementation	✓	✕
Counter-term (UV/R2) diagrams generation	✓-	✓
Complex mass scheme and massive bosons in the loop	✕	✕
Color Factor computation	✓-	✓
File output	✓--	✓
4-gluon R2 computation	✕	✓ (checks still needed)
Virtual squared	✓-	✕
Decay Chains	✕	✕
EPS handling	✓ (no mp)	✕
Sanity checks (Ward, ε^{-2})	✓	✕
Mixed order perturbation (generation level)	✕	✓
Automatic loop-model creation	✕	✕
Symmetry factor automatic computation	✕	✕

Loop-Cut diagrams

- How much faster are they generated?

Process	Generation time[1]		Output size[2]		Compilation time[3]		Running time[4]	
d d~ > u u~	8.750 s	5.378 s	200 Kb	268 Kb	0.931 s	2.996 s	0.0088 s	0.0094 s
d d~ > d d~ g	17.04 s	104.8 s	124 Kb	1.7 Mb	4.799 s	19.181 s	0.64 s	0.74 s
d d~ > d d~ u u~	22.50 s	2094 s	232 Kb	3.3 Mb	37.75 s	45.02 s	1.93 s	2.34 s
g g > g g g g	2277 s	×	25 Mb	×	NOT COMPILING YET	×	NOT COMPILING YET	×

Preliminary

[1] : Process generated in a massless $n_f=2$ QCD model with reduced particle content.
[2] : Of the equivalent matrix.f file. [4] : Per PS points, computed over 1000 PS points.
[3] : In MG5, no smart line-breaks for the JAMP definition. MG5@NLO = ♦ (green), MadLoop = ♦ (purple)

Loop-Cut diagrams

- How much faster are they generated?

Process	Generation time[1]		Output size[2]		Compilation time[3]		Running time[4]	
d d~ > u u~	8.750 s	5.378 s	200 Kb	268 Kb	0.931 s	2.996 s	0.0088 s	0.0094 s
d d~ > d d~ g	17.04 s	104.8 s	124 Kb	1.7 Mb	4.799 s	19.181 s	0.64 s	0.74 s
d d~ > d d~ u u~	22.50 s	2094 s	232 Kb	3.3 Mb	37.75 s	45.02 s	1.93 s	2.34 s
g g > g g g g	2277 s	×	25 Mb	×	NOT COMPILING YET	×	NOT COMPILING YET	×

Preliminary

[1] : Process generated in a massless $n_f=2$ QCD model with reduced particle content.
[2] : Of the equivalent matrix.f file. [4] : Per PS points, computed over 1000 PS points.
[3] : In MG5, no smart line-breaks for the JAMP definition. MG5@NLO = ♦, MadLoop = ♦

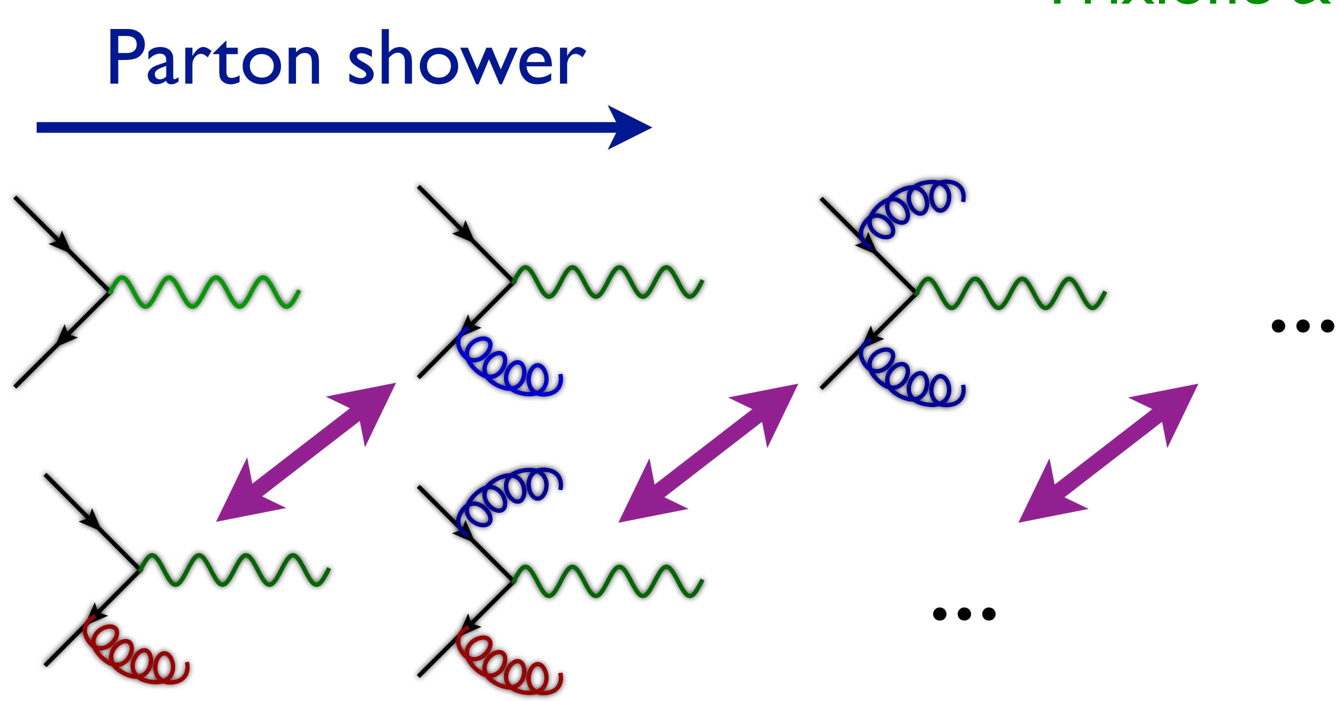
- Why?
  - The MG5 diagram generation is already much faster for tree-level diagrams.
  - It is modified so that external bubbles and tadpoles are not generated.
  - When generating diagrams for a given L-Cut particle, all previously considered L-Cut particles are vetoed from being loop-lines.

MC@NLO procedure

Frixione & Webber

Born+Virtual:

Real emission:



$$\frac{d\sigma_{\mathrm{NLOwPS}}}{dO} = \left[d\Phi_m \left(B + \int_{\mathrm{loop}} V + \int d\Phi_1 MC\right)\right] I_{\mathrm{MC}}^{(m)}(O)$$

$$+ \left[d\Phi_{m+1} (R - MC) \right] I_{\mathrm{MC}}^{(m+1)}(O)$$

- Double counting is explicitly removed by including the "shower subtraction terms"

NLO parton shower matching

[Torrielli, Frederix & Frixione]

$$d\sigma^{(\mathbb{H})}_{\mathrm{MC@NLO}} = d\phi_{n+1}\left(\mathcal{M}^{(r)}(\phi_{n+1}) - \mathcal{M}^{(\mathrm{MC})}(\phi_{n+1})\right)$$

$$d\sigma^{(\mathbb{S})}_{\mathrm{MC@NLO}} = \int_{+1} d\phi_{n+1}\left(\mathcal{M}^{(b+v+rem)}(\phi_n) - \mathcal{M}^{(c.t.)}(\phi_{n+1}) + \mathcal{M}^{(\mathrm{MC})}(\phi_{n+1})\right)$$

- In black: pure NLO (fully tested in MadFKS)
- In red: MC counter terms (implemented for Herwig6, Pythia and Herwig++, but only fully tested for Herwig)
- FKS subtraction is based on a collinear picture, and so are the MC counter terms: branching structure is for free
- Automatic determination of color partners
- Works also when MC-ing over helicities

Negative weights

$$\frac{d\sigma_{\mathrm{NLOwPS}}}{dO} = \left[d\Phi_m (B + \int_{\mathrm{loop}} V + \int d\Phi_1 MC) \right] I_{\mathrm{MC}}^{(m)}(O) + \left[d\Phi_{m+1} (R - MC) \right] I_{\mathrm{MC}}^{(m+1)}(O)$$

- We generate events for the two terms between the square brackets (“S”- and “H”-events) separately
- There is no guarantee that these contributions are separately positive (even though predictions for infra-red safe observables should always be positive!)
- Therefore, when we do event unweighting we can only unweight the events **up to a sign**. These signs should be taken into account when doing a physics analysis (i.e. making plots etc.)
- The events are only physical when they are showered

pp → Htt̄/Att̄

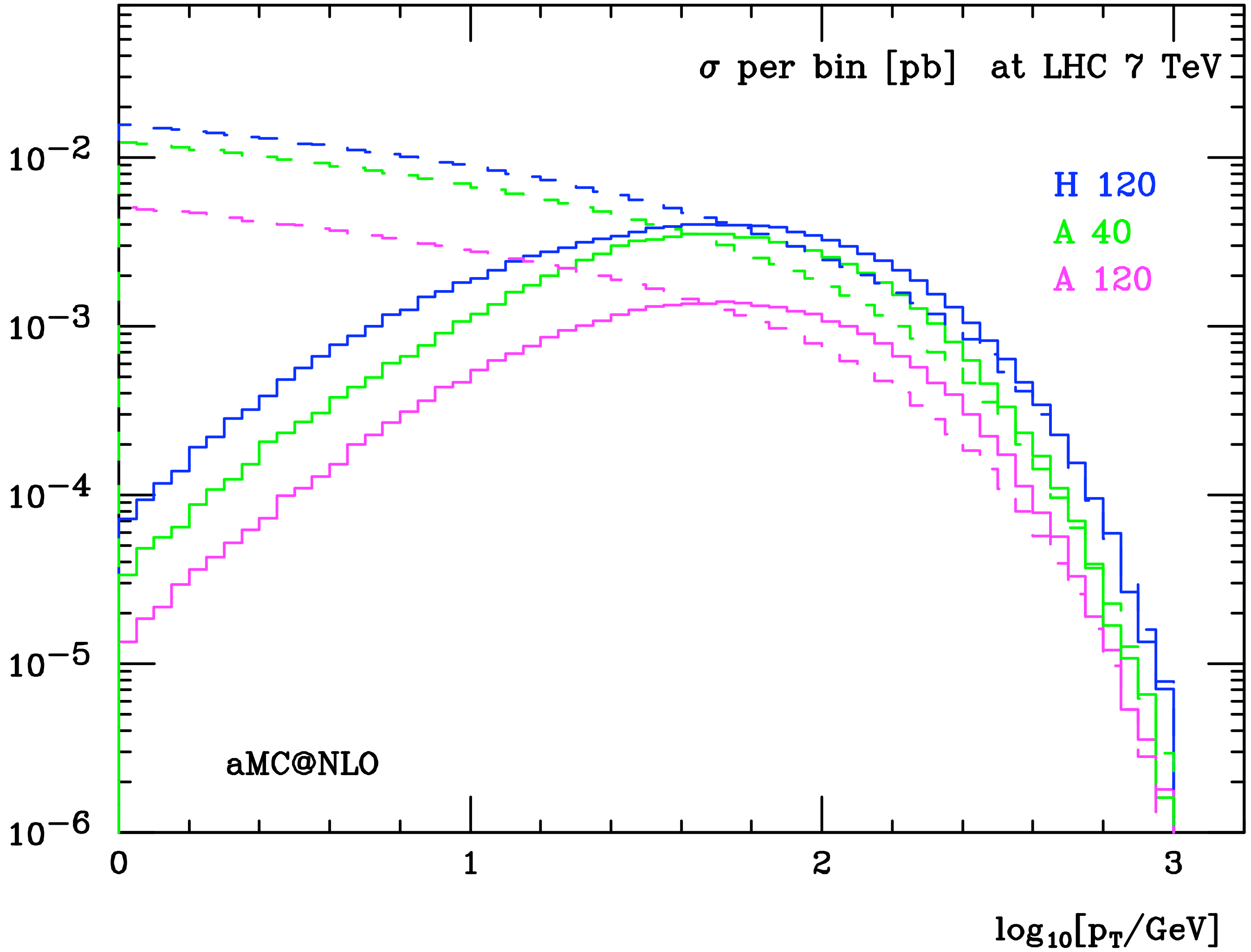
arXiv:1104.5613

- Top pair production in association with a (pseudo-)scalar Higgs boson
- Three scenarios
  I) scalar Higgs H, with m_H = 120 GeV
  II) pseudo-scalar Higgs A, with m_A = 120 GeV
  III) pseudo-scalar Higgs A, with m_A = 40 GeV
- SM-like Yukawa coupling, $y_t/\sqrt{2}=m_t/v$
- Renormalization and factorization scales $\mu_F = \mu_R = \left(m_T^t m_T^{\bar{t}} m_T^{H/A}\right)^{\frac{1}{3}}$ with $m_T = \sqrt{m^2 + p_T^2}$ and $m_t^{pole} = m_t^{\overline{MS}} = 172.5$ GeV
- Note: first time that pp → ttA has been computed beyond LO

Impact of the shower

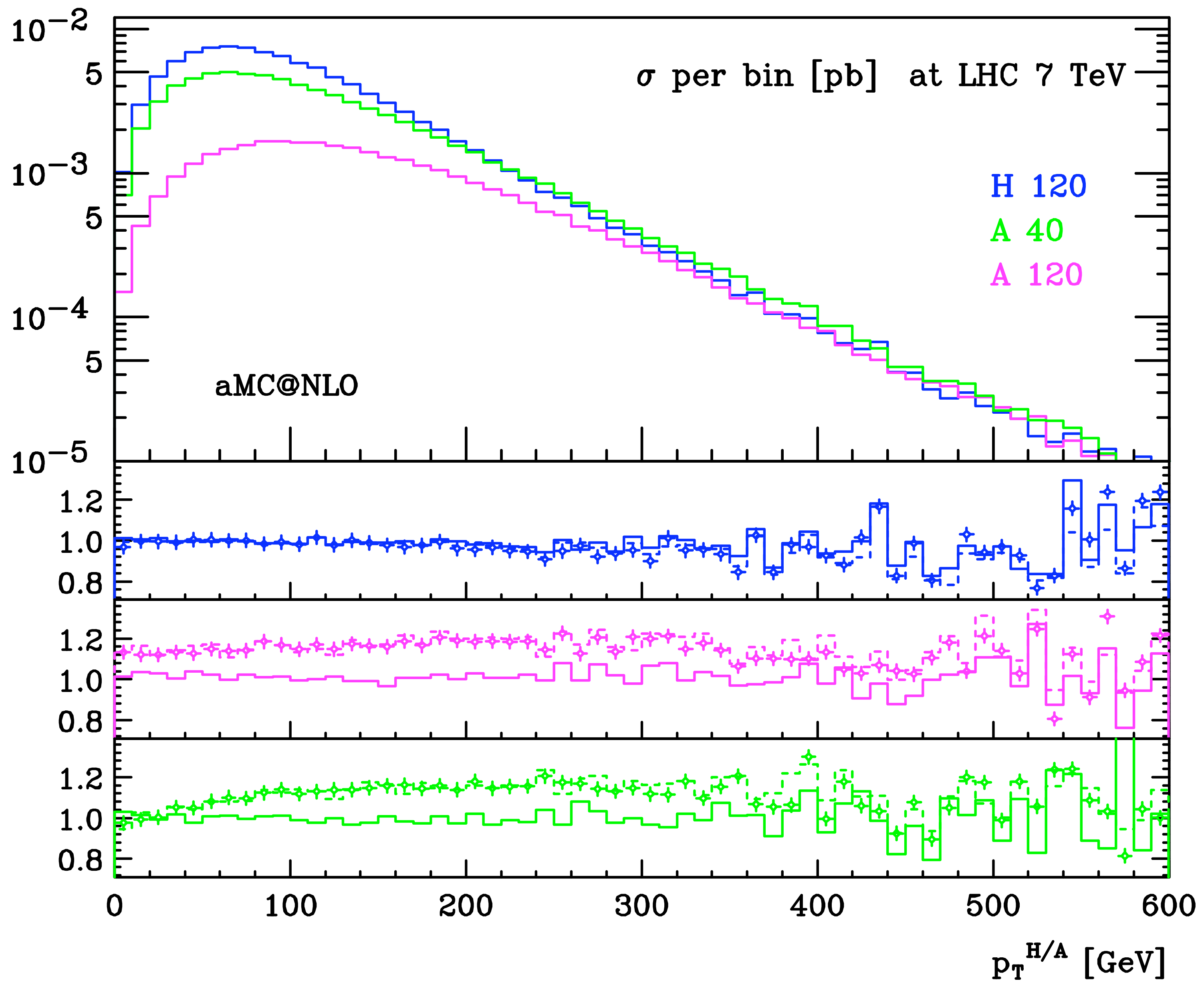
- Three particle transverse momentum, p_T(H/A t tbar), is sensitive to the impact of the parton shower
- Infrared sensitive observable at the pure-NLO level for $p_T \rightarrow 0$
- aMC@NLO displays Sudakov suppression for small p_T
- At large p_T the MC@NLO and parton-level NLO descriptions coincide in shape and rate

Dashed: NLO, Solid: aMC@NLO



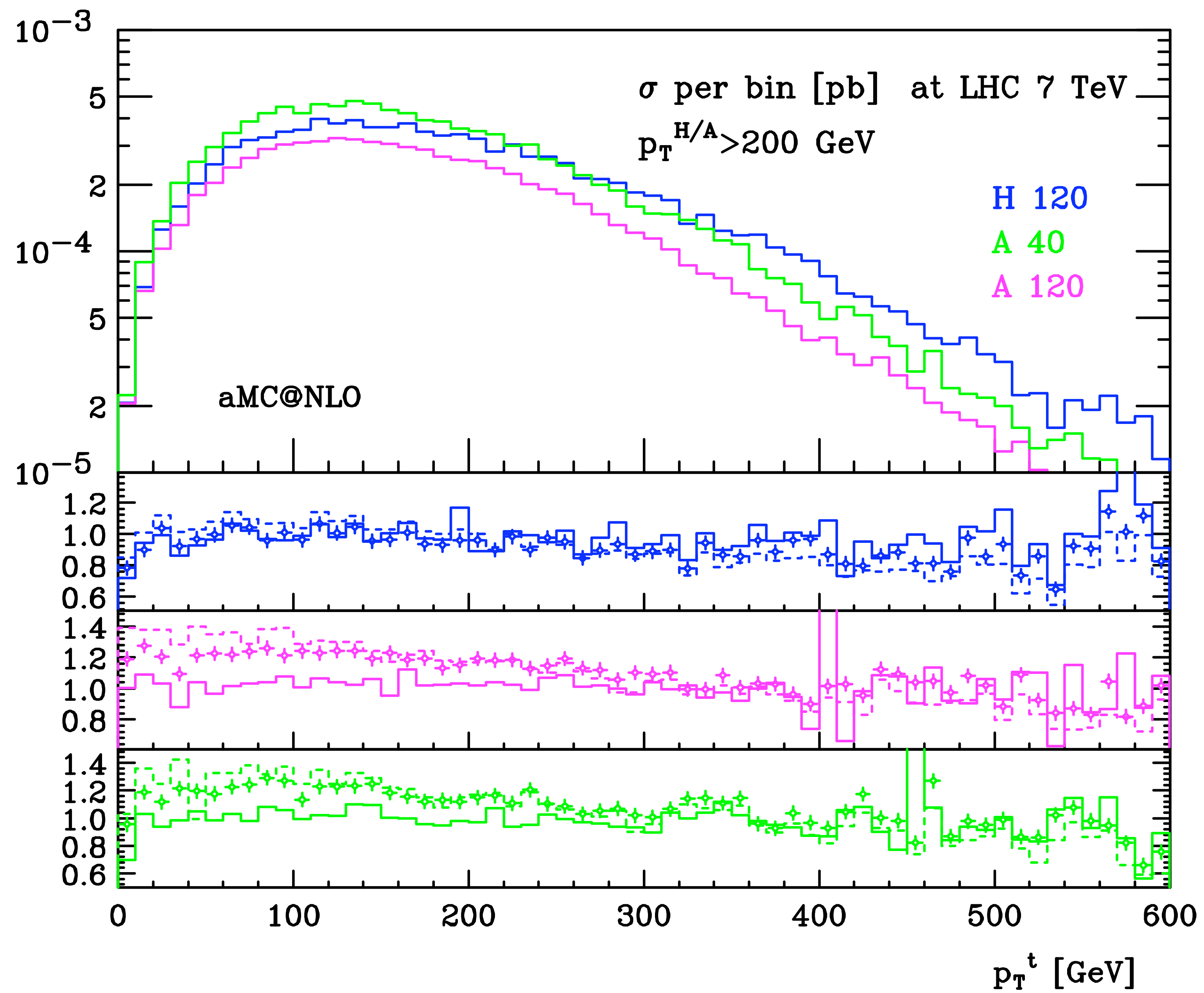
Higgs p_T

- Transverse momentum of the Higgs boson
- Lower panels show the ratio of aMC@NLO with LO (dotted), NLO (solid) and LO MC (crosses)
- Corrections are small and fairly constant
- At large p_T, scalar and pseudo-scalar production coincide: boosted Higgs scenario [*Butterworth et al., Plehn et al.*] should work equally well for pseudo-scalar Higgs



Boosted Higgs

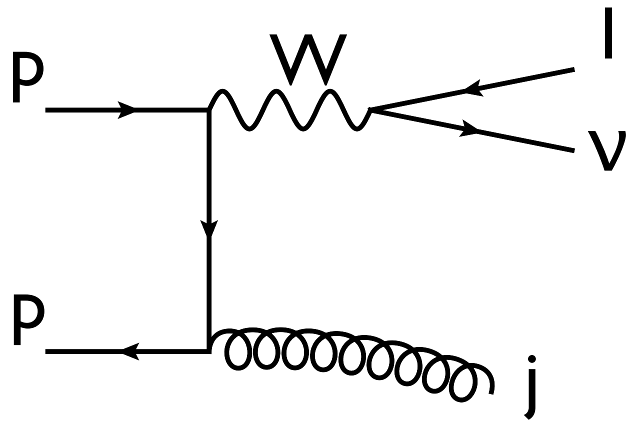
- Boosted Higgs: $p_T^{H/A} > 200$ GeV
- Transverse momentum of the top quark
- Corrections compared to LO are significant and cannot be approximated by a constant K-factor



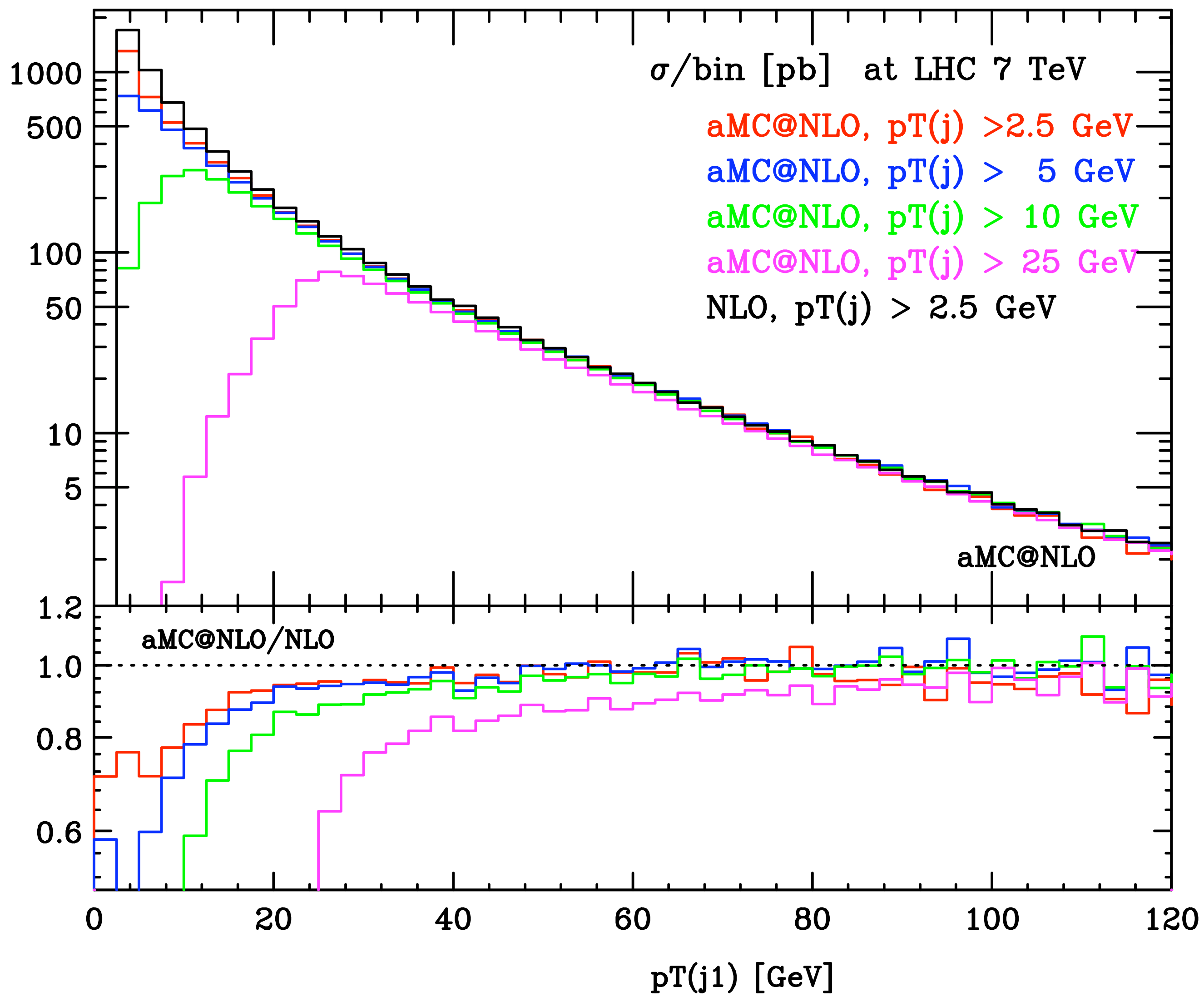
Computational challenge

- This is the first time that such a process with so many scales and possible (IR) divergences is matched to a parton shower at NLO accuracy
- Start with W+1j production to validate processes which need cuts at the matrix-element level
- To check the insensitivity to this cut:
  - generate a couple of event samples with different cuts and show that the distributions after analysis cuts are statistically equivalent

$pp \to Wj$



- For W+1j the easiest cut would be in on the p_T of the W boson
- However, for validation purposes it is more appropriate to apply this cut on the jet instead (because that is what we'll be doing in W+2j). Same at LO, but different at NLO
- Different cuts at generation level yield the same distributions at analysis level if the analysis level cut is 3-4 times larger



$pp \to Wjj$
Setup

- Two event samples with 5 GeV and 10 GeV p_T cuts on the jets at generation level, respectively, each with 10 million unweighted events
- Renormalization and factorization scales equal to $\mu_R = \mu_F = H_T/2$

  $$2\mu_R = 2\mu_F = H_T = \sqrt{(p_{T,l\nu}^2 + m_{l\nu}^2)} + \sum |p_{T,i}|$$

  where sum is over the 2 or 3 partons (and the matrix element level)
- Jets are defined with anti-k_T and R=0.4
- MSTW2008(N)LO PDF set for the (N)LO predictions (with $\alpha_s(m_Z)$ from PDF set using (2)1-loop running)
- $m_W = 80.419$ GeV,
  $G_F = 1.16639 \cdot 10^{-5}\ \mathrm{GeV}^{-2}$,
  $\alpha^{-1} = 132.507$,
  $\Gamma_W = 2.0476$ GeV

$p\bar{p} \to Wjj$ CDF/DØ analysis cuts

- minimal transverse energy for the lepton: $E_T(l) > 20$ GeV;
- maximal pseudo rapidity for the lepton: $|\eta(l)| < 1$;
- minimal missing transverse energy: $\not{E}_T > 25$ GeV;
- minimal transverse W-boson mass: $M_T(l\nu_l) > 30$ GeV;
- jet definition: JetClu algorithm with 0.75 overlap and $R = 0.4$;
- minimal transverse jet energy: $E_T(j) > 30$ GeV;
- maximal jet pseudo rapidity: $|\eta(j)| < 2.4$;
- minimal jet pair transverse momentum: $p_T(j_1 j_2) > 40$ GeV;
- minimal jet-lepton separation: $\Delta R(lj) > 0.52$;
- minimal jet-missing energy separation: $\Delta\phi(\not{E}_T j) > 0.4$;
- hardest jets close in pseudorapidity: $|\Delta\eta(j_1 j_2)| < 2.5$;
- jet veto: no third jet with $E_T(j) > 30$ GeV and $|\eta(j)| < 2.4$;
- lepton isolation: transverse hadronic energy smaller than 10% of the lepton transverse energy in a cone of $R = 0.4$ around the lepton.

- To slightly simplify the analysis, the MC truth is used to assign the lepton to the W-boson decay
- Only W^+ events (simply a factor 2)
- No underlying event