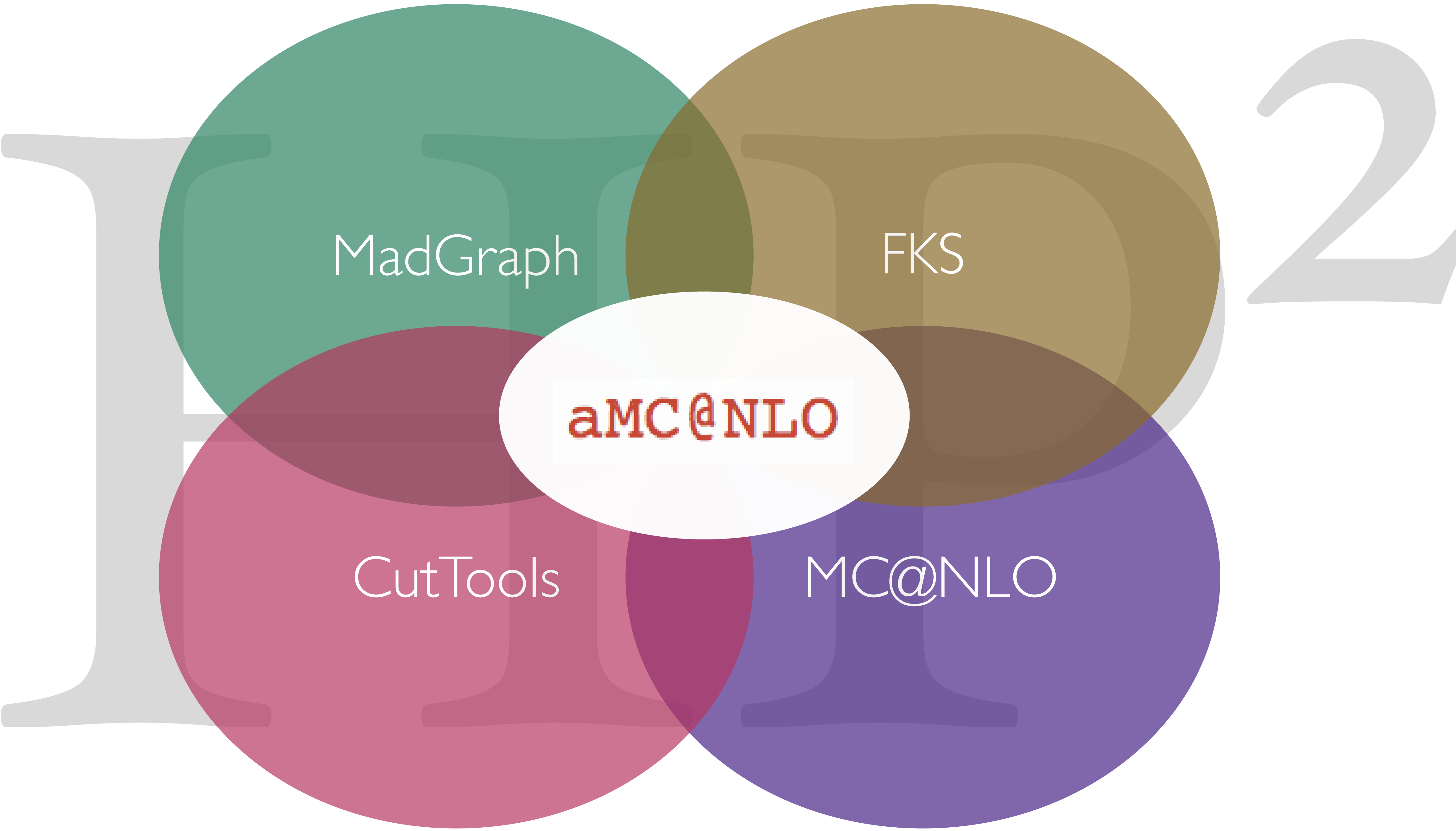
aMC@NLO with MadGraph5

Marco Zaro, CP3 - UCLouvain
in collaboration with
Rikkert Frederix, Stefano Frixione, Fabio Maltoni, Paolo Torrielli, Valentin Hirschi
& the MadGraph5 Team

HP2 2012,
Max Planck Institute for Physics, Munich

HP^2



MadGraph
FKS
aMC@NLO
CutTools
MC@NLO

aMC@NLO: motivations

aMC@NLO: motivations

- Why automation?
  - Time: trade time spent to code/debug with time to do pheno
  - Trust: results from automatic tools are "correct by definition"
  - Easy: automatic tools can be used as black-boxes: no need for highly-skilled users

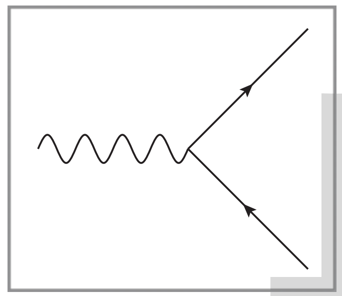
aMC@NLO: motivations

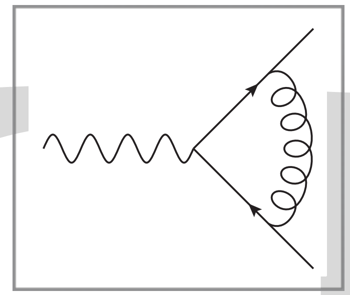
- Why automation?
  - Time: trade time spent to code/debug with time to do pheno
  - Trust: results from automatic tools are "correct by definition"
  - Easy: automatic tools can be used as black-boxes: no need for highly-skilled users
- Why NLO?
  - Reliable prediction of total rates
  - Reduction of theoretical uncertainties

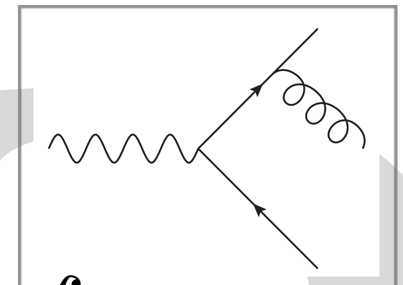
aMC@NLO: motivations

- Why automation?
  - Time: trade time spent to code/debug with time to do pheno
  - Trust: results from automatic tools are "correct by definition"
  - Easy: automatic tools can be used as black-boxes: no need for highly-skilled users
- Why NLO?
  - Reliable prediction of total rates
  - Reduction of theoretical uncertainties
- Why PS-matching?
  - We do not see partons at colliders
  - Matching to PS cures some fixed-order ill-behaved observables

NLO basics

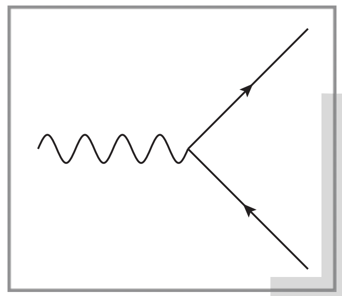
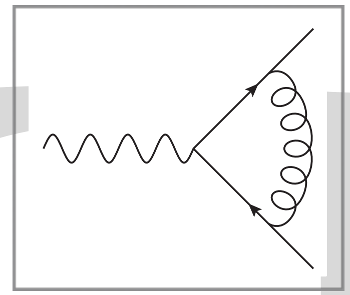
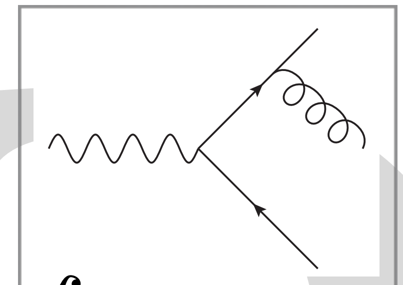






$$d\sigma^n_{NLO} = d\sigma^n_{LO} + d\sigma^n_V + \int d\Phi_1 \; d\sigma^{n+1}_R$$

NLO basics

$$d\sigma^n_{NLO} = d\sigma^n_{LO} + d\sigma^n_V + \int d\Phi_1 \; d\sigma^{n+1}_R$$

$$d\sigma^n_{NLO} = d\sigma^n_{LO} + d\sigma^n_V - \int d\Phi_1 \; C + \int d\Phi_1 \; \left(C + d\sigma^{n+1}_R \right)$$

- Generate virtual Matrix Elements
- Generate real emission MEs and counterterms, put them together and integrate them in an efficient way

The OPP Method

Ossola, Papadopoulos, Pittau, arXiv:hep-ph/0609007 & arXiv:0711.3596

- Exploit the OPP method: reduce the virtual amplitude at the integrand level to a linear combination of scalar integrals, plus rational terms

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

$$\begin{aligned} N(q) = & \sum_{i_0<i_1<i_2<i_3}^{m-1} \left[d(i_0i_1i_2i_3) + \tilde{d}(q;i_0i_1i_2i_3)\right] \prod_{i\neq i_0,i_1,i_2,i_3}^{m-1} D_i \\ & + \sum_{i_0<i_1<i_2}^{m-1} \left[c(i_0i_1i_2) + \tilde{c}(q;i_0i_1i_2)\right] \prod_{i\neq i_0,i_1,i_2}^{m-1} D_i \\ & + \sum_{i_0<i_1}^{m-1} \left[b(i_0i_1) + \tilde{b}(q;i_0i_1)\right] \prod_{i\neq i_0,i_1}^{m-1} D_i \\ & + \sum_{i_0}^{m-1} \left[a(i_0) + \tilde{a}(q;i_0)\right] \prod_{i\neq i_0}^{m-1} D_i \\ & + \tilde{P}(q) \prod_{i}^{m-1} D_i\,. \end{aligned}$$

- Use CutTools: feed with the loop numerator, and obtain the coefficients of the scalar integrals and CC rational terms (R1)
- Add R2-rational terms/UV counterterms (process-independent)

The FKS subtraction

- Find parton pairs *i, j* that can give collinear singularities
- Split the phase space into regions with one collinear sing
  - Soft singularities are split into the collinear ones

$$|M|^2 = \sum_{ij} S_{ij} |M|^2 = \sum_{ij} |M|^2_{ij} \qquad \sum S_{ij} = 1$$

$$S_{ij} \to 1 \text{ if } k_i \cdot k_j \to 0 \qquad S_{ij} \to 0 \text{ if } k_{m \neq i} \cdot k_{n \neq j} \to 0$$

- Integrate them independently
  - Parallelize integration
  - Choose ad-hoc phase space parameterization
- Advantages:
  - # of contributions ~ n^2 (dipole ~ n^3)
  - Exploit symmetries: 3 contributions for x y > ng

MC@NLO basics

- MC@NLO: match NLO computation with PS
  - Define MC counterterms in the cross-section computation

$$\frac{d\sigma_{MC@NLO}}{dO} = \left[\int d\Phi_n (B + V + \int d\Phi_1 MC)\right] I^n_{MC}(O) + \left[\int d\Phi_{n+1}(R - MC)\right] I^{n+1}_{MC}(O)$$

  - MC is the cross-section for the first emission from the shower

$$I^k_{MC} = \Delta + \Delta d\Phi_1 \frac{MC}{B} + \dots \qquad \Delta = \exp\left[-\int d\Phi_1 \frac{MC}{B}\right]$$

  - NLO normalization is kept
  - MC are MC-dependent, but process-independent
    - Currently available for Herwig6, Herwig++, Pithya6

$$MC = J \frac{1}{t_{MC}} \frac{\alpha_s}{2\pi} P(z^{MC}) B$$

Automation with MadGraph

HP^2

Automation with MadGraph

- MadGraph is an automatic tree-level matrix element generator

Automation with MadGraph

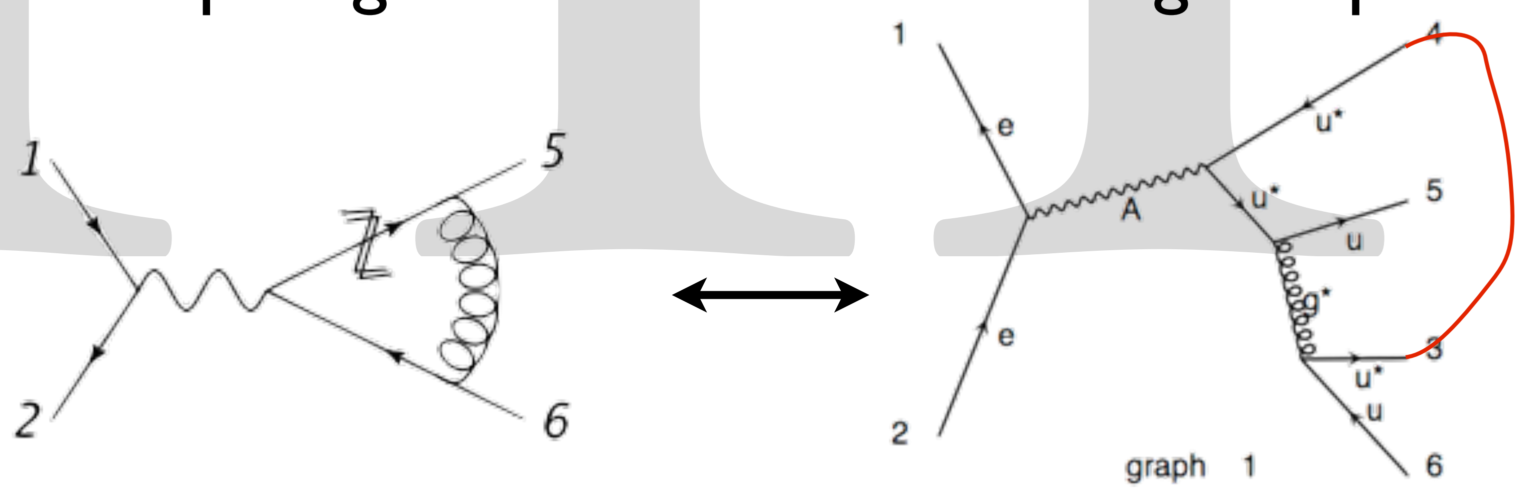
- MadGraph is an automatic tree-level matrix element generator
- Pro: Can generate any tree-level matrix element
  - Born MEs
  - Real MEs

Automation with MadGraph

- MadGraph is an automatic tree-level matrix element generator
- Pro: Can generate any tree-level matrix element
  - Born MEs
  - Real MEs
- Con: Can generate only tree-level matrix elements
  - What about the virtuals?

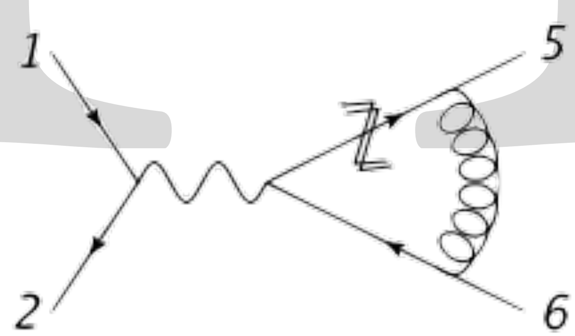
Automation with MadGraph

- MadGraph is an automatic tree-level matrix element generator
- Pro: Can generate any tree-level matrix element
  - Born MEs
  - Real MEs
- Con: Can generate only tree-level matrix elements
  - What about the virtuals?
    - Think of loop diagrams as trees with two glued particles

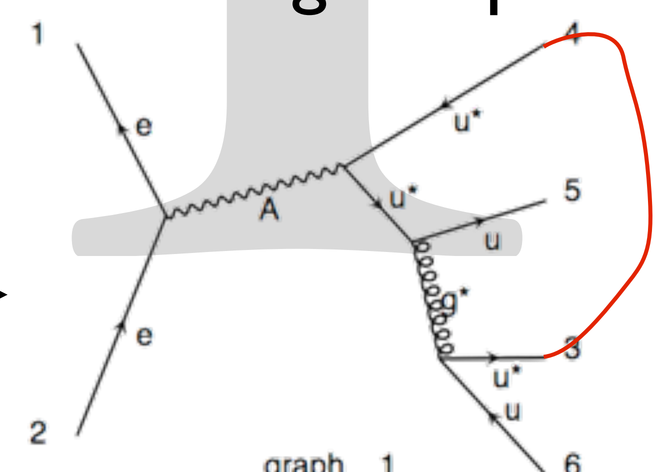


Automation with MadGraph

- MadGraph is an automatic tree-level matrix element generator
- Pro: Can generate any tree-level matrix element
  - Born MEs
  - Real MEs
- Con: Can generate only tree-level matrix elements
  - What about the virtuals?
    - Think of loop diagrams as trees with two glued particles



↔



    - UV/R2 terms are just new vertices

MadLoop & MadFKS

HP^2

MadLoop & MadFKS

- MadLoop (see Valentin's talk):
  - Computes the loop numerator for any given amplitude and feeds it to CutTools
  - Adds R2/UV counterterms (coded as new vertices)

MadLoop & MadFKS

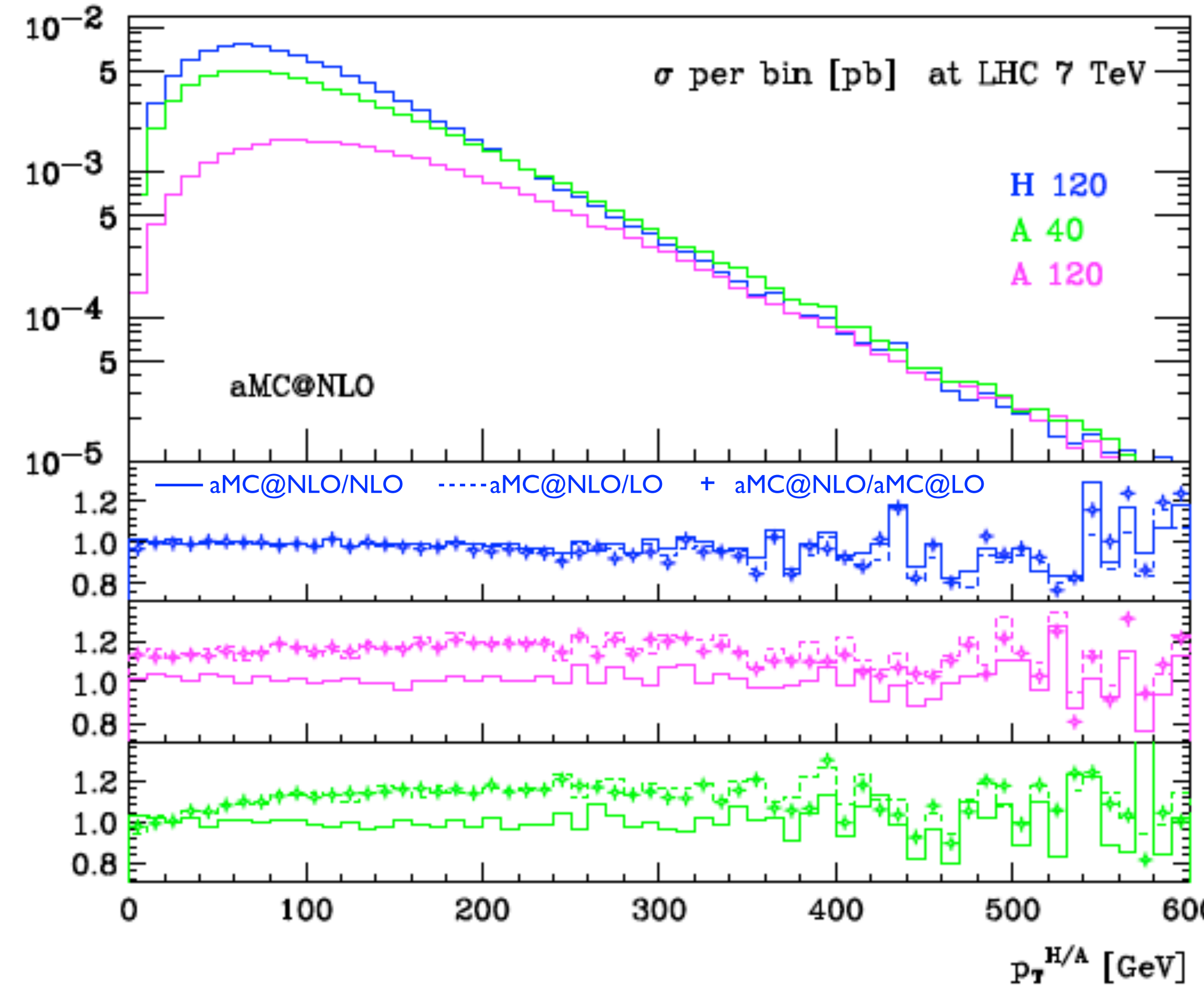
- MadLoop (see Valentin's talk):
  - Computes the loop numerator for any given amplitude and feeds it to CutTools
  - Adds R2/UV counterterms (coded as new vertices)
- MadFKS:
  - Generates the real and born MEs and counterterms (color- and spin-linked borns)
  - Links the virtuals from MadLoop or via BLHA
  - Organizes the integration of n and n+1 body cross-section
  - Generates events to be showered

Physics results with MG4

HP^2

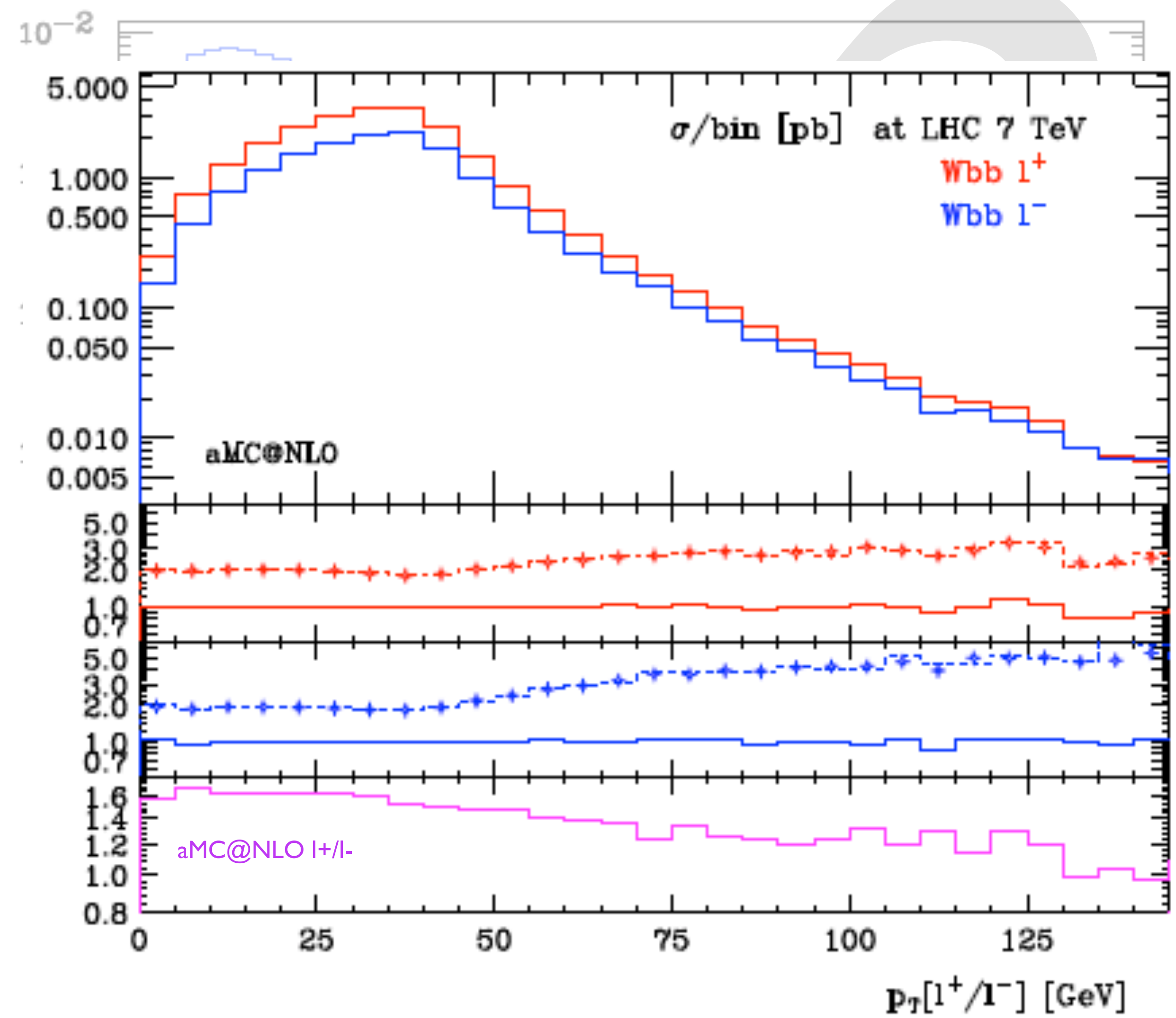
Physics results with MG4

- ttH/ttA, arXiv:1104.5613



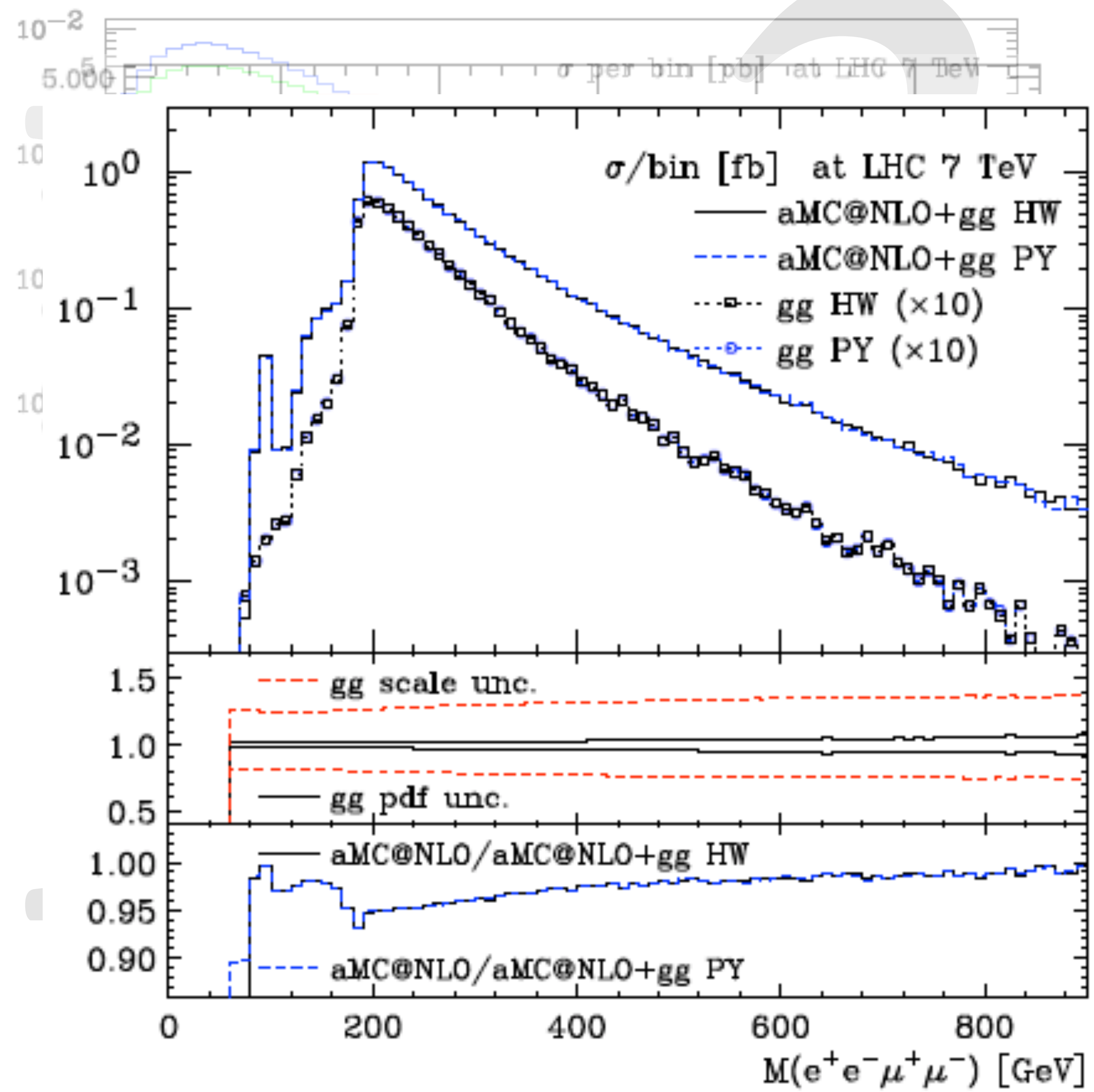
Physics results with MG4

- ttH/ttA, arXiv:1104.5613
- Vbb, arXiv:1106:6019



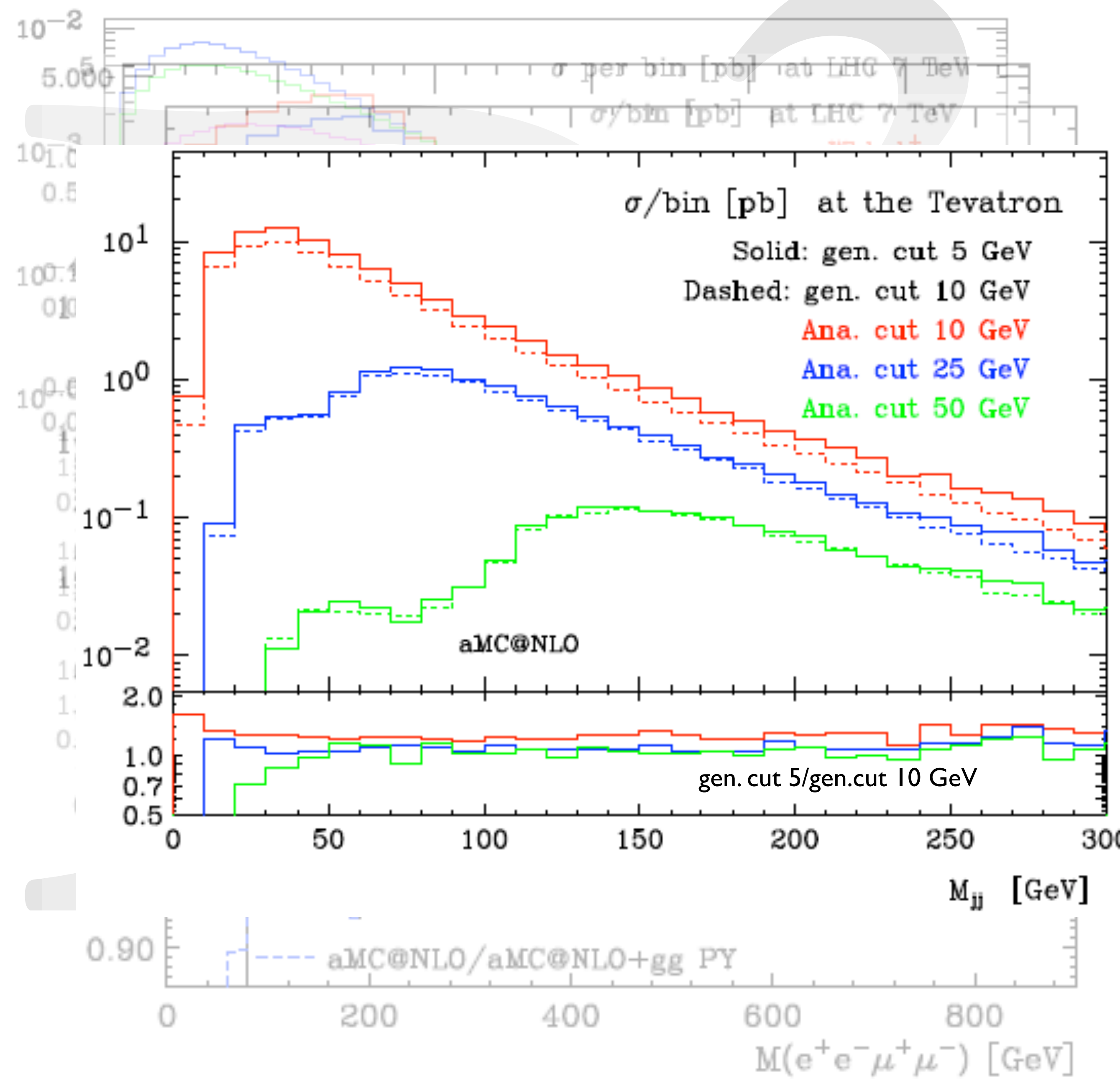
Physics results with MG4

- ttH/ttA, arXiv:1104.5613
- Vbb, arXiv:1106:6019
- 4 leptons, arXiv:1110.4738
  - Scales uncertainities included via reweighting
  - Added Pythia counterterms



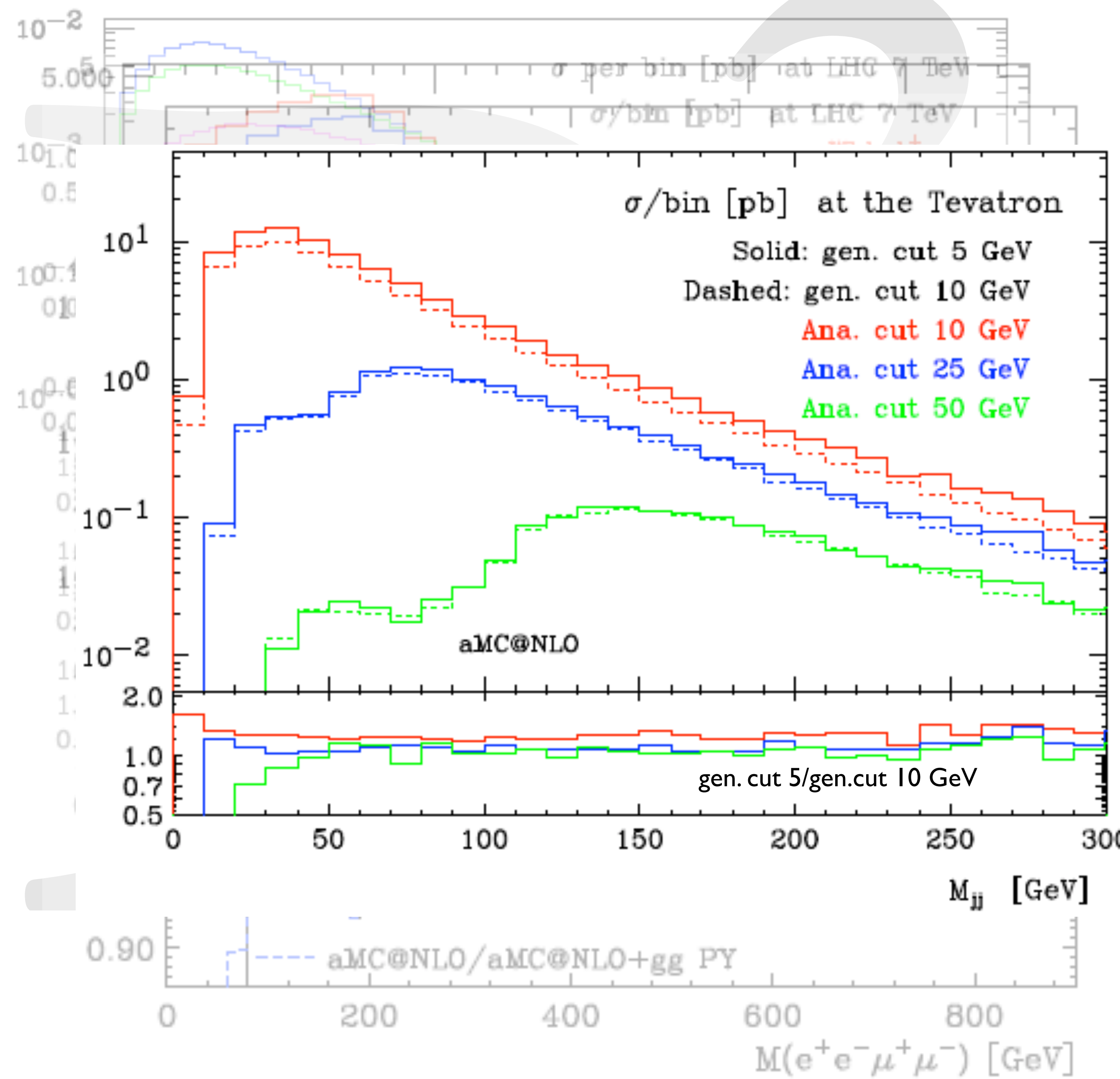
Physics results with MG4

- ttH/ttA, arXiv:1104.5613
- Vbb, arXiv:1106:6019
- 4 leptons, arXiv:1110.4738
  - Scales uncertainities included via reweighting
  - Added Pythia counterterms
- Wjj, arXiv:1110.5502



Physics results with MG4

- ttH/ttA, arXiv:1104.5613
- Vbb, arXiv:1106:6019
- 4 leptons, arXiv:1110.4738
  - Scales uncertainities included via reweighting
  - Added Pythia counterterms
- Wjj, arXiv:1110.5502



What happened in 2012?

Limitations with MGv4

- Identification of FKS i/j pair hardcoded and based on particles identities rather than on interactions
  - Difficult to generalize to new models or non-QCD perturbations
- No finite width effects in the loops
- 4glu R2 too much complicated to be implemented in Fortran
  - Virtuals for processes with 4glu vertices incomplete

From MG4 to MG5

- MadGraph completely rewritten in Python
  - Modern coding language
  - Modular/Object oriented structure
  - Unit/Acceptance/Parallel tests to test the code behaviour
  - New diagram generation algorithms
    - Code generation time drastically reduced
  - Possibility to reuse common part of diagrams
    - Running time improvement
  - Any model can be loaded via the FR/UFO interface
  - Possibility do define vertices with arbitrary # of legs and color structures

From MG4 to MG5

Process	MadGraph 4	MadGraph 5	Subprocesses	Diagrams
$pp \to jjj$	29.0 s	25.8 s	34	307
$pp \to jjl^+l^-$	341 s	103 s	108	1216
$pp \to jjje^+e^-$	1150 s	134 s	141	9012
$u\bar{u} \to e^+e^-e^+e^-e^+e^-$	772 s	242 s	1	3474
$gg \to ggggg$	2788 s	1050 s	1	7245
$pp \to jj(W^+ \to l^+\nu_l)$	146 s	25.7 s	82	304
$pp \to t\bar{t}$+full decays	5640 s	15.7 s	27	45
$pp \to \tilde{q}/\tilde{g}\,\tilde{q}/\tilde{g}$	222 s	107 s	313	475
7 particle decay chain	383 s	13.9 s	1	6
$gg \to (\tilde{g} \to u\bar{u}\tilde{\chi}_1^0)(\tilde{g} \to u\bar{u}\tilde{\chi}_1^0)$	70 s	13.9 s	1	48
$pp \to (\tilde{g} \to jj\tilde{\chi}_1^0)(\tilde{g} \to jj\tilde{\chi}_1^0)$	—	251 s	144	11008

- MadGraph c
  - Modern cc
  - Modular/C
  - Unit/Accep
  - New diagra
    - Code gene
  - Possibility to reuse common part of diagrams
    - Running time improvement
  - Any model can be loaded via the FR/UFO interface
  - Possibility do define vertices with arbitrary # of legs and color structures

UCL
Université catholique de Louvain

From MG4 to MG5

- MadGraph c
  - Modern cc
  - Modular/C
  - Unit/Accep

Process	MadGraph 4	MadGraph 5	Subprocesses	Diagrams
$pp \to jjj$	29.0 s	25.8 s	34	307
$pp \to jjl^+l^-$	341 s	103 s	108	1216
$pp \to jjje^+e^-$	1150 s	134 s	141	9012
$u\bar{u} \to e^+e^-e^+e^-e^+e^-$	772 s	242 s	1	3474
$gg \to ggggg$	2788 s	1050 s	1	7245
$pp \to jj(W^+ \to l^+\nu_l)$	146 s	25.7 s	82	304
$pp \to t\bar{t}$+full decays	5640 s	15.7 s	27	45
$pp \to \tilde{q}/\bar{\tilde{q}}\,\tilde{q}/\bar{\tilde{q}}$	222 s	107 s	313	475
		13.9 s	1	6
		13.9 s	1	48
		251 s	144	11008

grams

O interface

ary # of legs and

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

Process	Subproc. dirs.		Channels		Directory size		Event gen. time	
	MG 4	MG 5	MG 4	MG 5	MG 4	MG 5	MG 4	MG 5
$pp \to W^+j$	6	2	12	4	79 MB	35 MB	3:15 min	1:55 min
$pp \to W^+jj$	41	4	138	24	438 MB	64 MB	9:15 min	4:19 min
$pp \to W^+jjj$	73	5	1164	120	842 MB	110 MB	21:41 min*	8:14 min*
$pp \to W^+jjjj$	296	7	15029	609	3.8 GB	352 MB	2:54 h*	46:50 min*
$pp \to W^+jjjjj$	-	8	-	2976	-	1.5 GB	-	11:39 h*
$pp \to l^+l^-j$	12	2	48	8	149 MB	44 MB	21:46 min	3:00 min
$pp \to l^+l^-jj$	54	4	586	48	612 MB	83 MB	2:40 h	11:52 min
$pp \to l^+l^-jjj$	86	5	5408	240	1.2 GB	151 MB	49:18 min*	16:38 min*
$pp \to l^+l^-jjjj$	235	7	65472	1218	5.3 GB	662 MB	7:16 h*	2:45 h*
$pp \to t\bar{t}$	3	2	5	3	49 MB	39 MB	2:39 min	1:55 min
$pp \to t\bar{t}j$	7	3	45	17	97 MB	56 MB	10:24 min	3:52 min
$pp \to t\bar{t}jj$	22	5	417	103	274 MB	98 MB	1:50 h	32:37 min
$pp \to t\bar{t}jjj$	34	6	3816	545	620 MB	209 MB	2:45 h*	23:15 min*

aMC@NLO with MG5: tackling the old limitations

HP2

aMC@NLO with MG5: tackling the old limitations

Limitations in v4

- Identification of FKS i/j pair hardcoded and based on particles identities rather than on interactions
  - Difficult to generalize to new models or non-QCD perturbations
- 4glu R2 too much complicated to be implemented in Fortran
  - Virtuals for processes with 4glu vertices incomplete
- No finite width effects in the loops

Improvements in v5

- Splitting of legs based on the interactions in the model, and on their perturbation (QCD, QED...)
  - Fully model independent, possibility to extend to non-QCD perturbation
- 4glu R2 available, treated as sum of different color/Lorentz structures
  - Any virtual ME can be generated
- Loop-induced processes can be generated
- Complex-mass scheme soon available

aMC@NLO with MG5: further improvements

aMC@NLO with MG5: further improvements

MadFKS

- Drastic speed improvement on code-generation (for pp > 3j ~ 15 mins)
- More compact code structure, sum/MC over different real emissions
  - Reduction number of channels for multichannel integration (real > born), possibility to use real MEs not based on Feynman diagrams

MadLoop

- Speed improvements:
  - Recycle trees used in different diagrams
  - Call CutTools after helicity sum
  - Exploit open-loops techniques
- Multiprecision support

MadFKS4 vs MadFKS5: timings

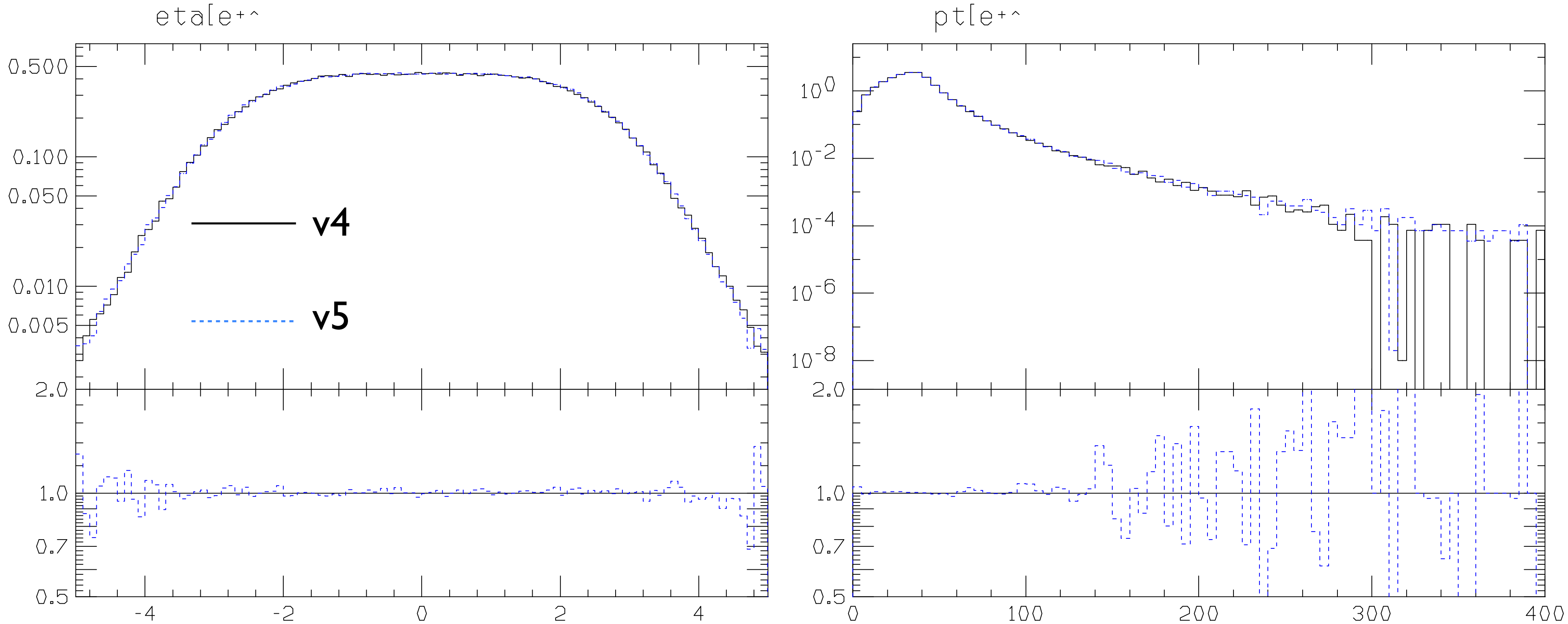
Generation, compilation and integration times, without virtuals
(on my MacBook Pro)

p p > t t~	MG4	MG5
code gen.	43s	5s
compilation	326s	50s
#channels	36	5
integration	270s (0.1%)	260s (0.1%)
p p > w+ b b~	MG4	MG5
code gen.	155s	5s
compilation	136s	36s
#channels	24	4
integration	316s (0.8%)	270s (1%)

MAdFKS4 vs MadFKS5: not so impressive... or?

- Running time for a given precision are ~ the same. However:
  - ALOHA subroutines are ~2 slower than HELAS ones
  - More subchannels make the error on the total rate smaller (sum in quadrature)
    - The relative error on the single channel is ~2-3 times smaller in v5
  - This do not include the virtuals from MadLoop

MAdFKS4 vs MadFKS5: Wbb

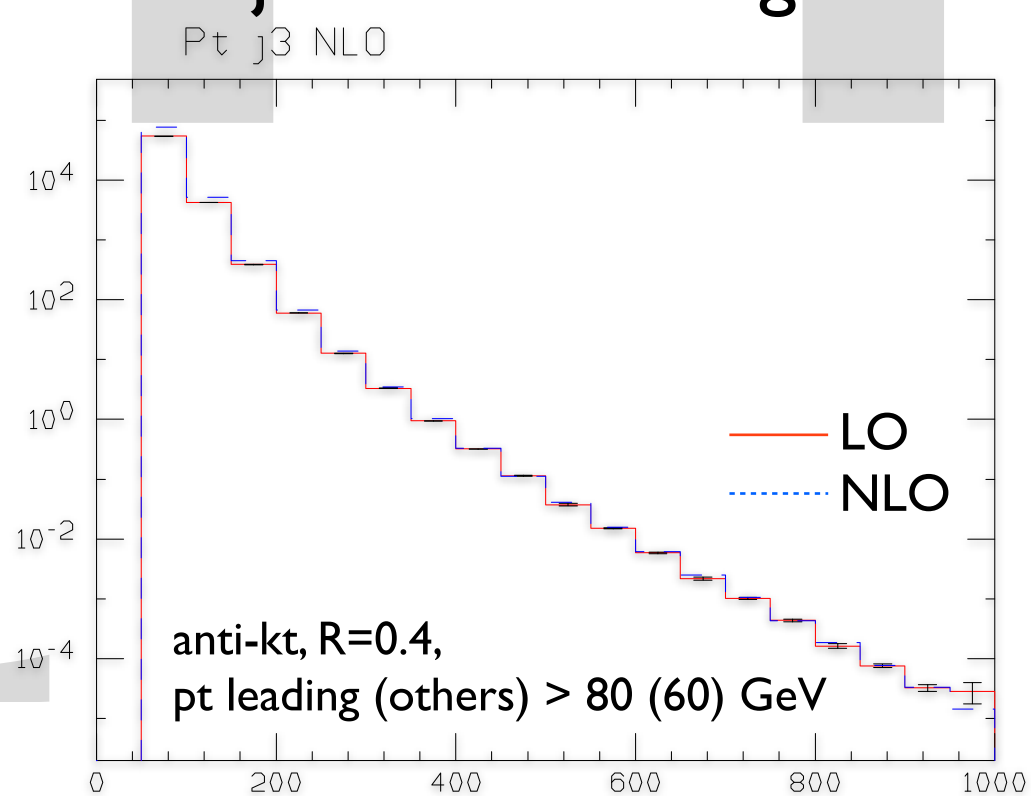


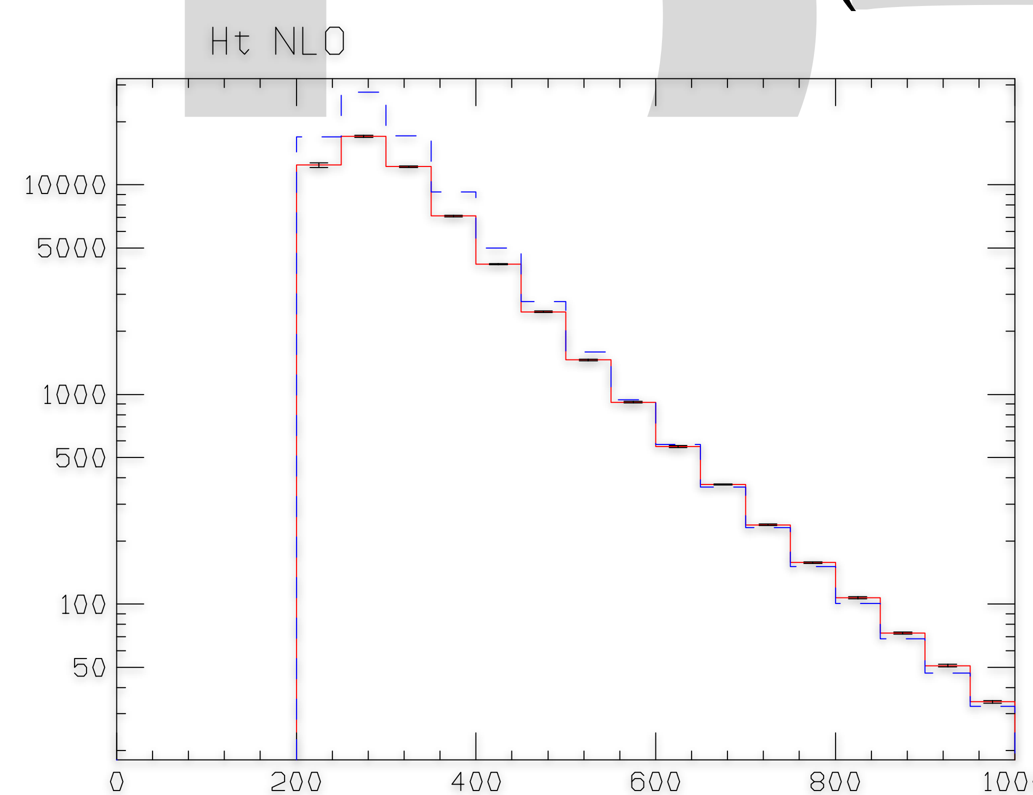
1M event sample, no cuts

Excellent agreement between the 2 codes

Physics: multi-jet production

- Virtuals from NGluon (Badger, Biedermann, Uwer, arXiv:1011.2900) via MadFKS BLHA interface
- 2 and 3 jets validated against numbers in arXiv:1112.3940 (BlackHat)





- Code for 4 jets ready
- Ongoing studies for dijet production matched to shower

Ongoing projects: production + decay

HP^2

Ongoing projects: production + decay

- How to deal with unstable final state particles (e.g. top) @NLO?

Ongoing projects: production + decay

- How to deal with unstable final state particles (e.g. top) @NLO?
  - Let the shower do the decay
    - Spin correlations lost

Ongoing projects: production + decay

- How to deal with unstable final state particles (e.g. top) @NLO?
  - Let the shower do the decay
    - Spin correlations lost
  - Generate full process with only stable particles (p p > l+l- v v~ b b~)
    - Includes spin correlations, off-shell effects, non resonant contributions, ...
    - Needs special treatment of intermediate resonances (e.g. cpx mass)
    - Computationally very expensive
    - Only needed when background is enhanced or when aiming at very high precision

Ongoing projects: production + decay

- How to deal with unstable final state particles (e.g. top) @NLO?
  - Let the shower do the decay
    - Spin correlations lost
  - Generate full process with only stable particles (p p > l+l- v v~ b b~)
    - Includes spin correlations, off-shell effects, non resonant contributions, ...
    - Needs special treatment of intermediate resonances (e.g. cpx mass)
    - Computationally very expensive
    - Only needed when background is enhanced or when aiming at very high precision

Anything in between?

The DECAY package

Artoisenet, Frederix, Rietkerk

The DECAY package

Artoisenet, Frederix, Rietkerk

- Wish-list:
  - For a given event sample (LO or MC@NLO), include the decay of any final state particle
  - Keep spin correlations
  - Generate decayed unweighted events

The DECAY package

Artoisenet, Frederix, Rietkerk

- Wish-list:
  - For a given event sample (LO or MC@NLO), include the decay of any final state particle
  - Keep spin correlations
  - Generate decayed unweighted events
- Solution:
  - Read event
  - Generate decay kinematics
  - Reweight the event with ratio $|M_{P+D}|^2 / |M_P|^2$
  - Or do secondary unweighting
    - Generate many decay configurations until
      $$|M_{P+D}|^2 / |M_P|^2 > \text{Rand()}\ \max\left(|\mathrm{M_{P+D}}|^2 / |\mathrm{M_P}|^2\right)$$

The DECAY package

Artoisenet, Frederix, Rietkerk

- Wish-list:
  - For a given event sample (LO or MC@NLO), include the decay of any final state particle
  - Keep spin correlations
  - Generate decayed unweighted events
- Solution:
  - Read event
  - Generate decay kinematics
  - Reweight the event with ratio $|M_{P+D}|^2 / |M_P|^2$
  - Or do secondary unweighting
    - Generate many decay configurations until

$$|M_{P+D}|^2 / |M_P|^2 > \mathrm{Rand}() \ \max\left(|\mathrm{M_{P+D}}|^2 / |\mathrm{M_P}|^2\right)$$

- This has already been done for t t~ and singletop

Frixione, Leanen, Motylinski, Webber, arXiv:hep-ph/0702198

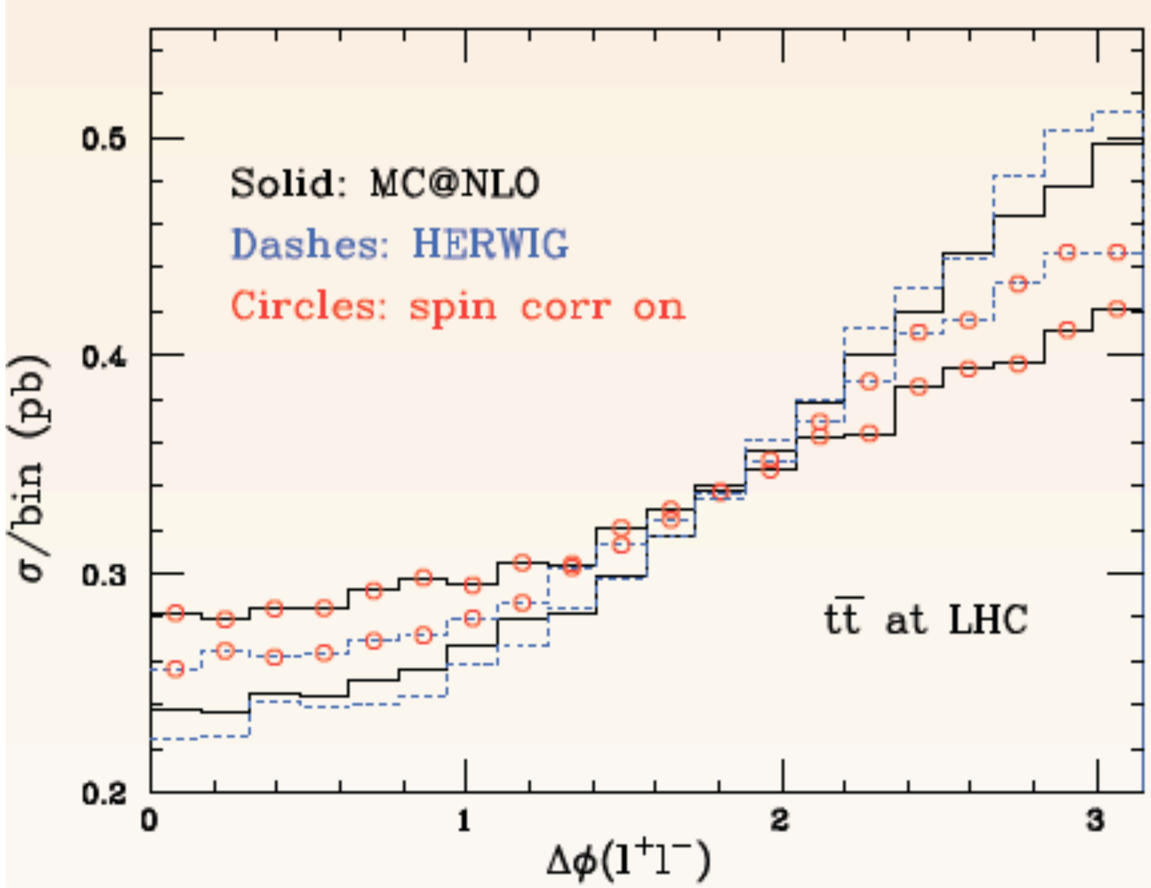
The DECAY package

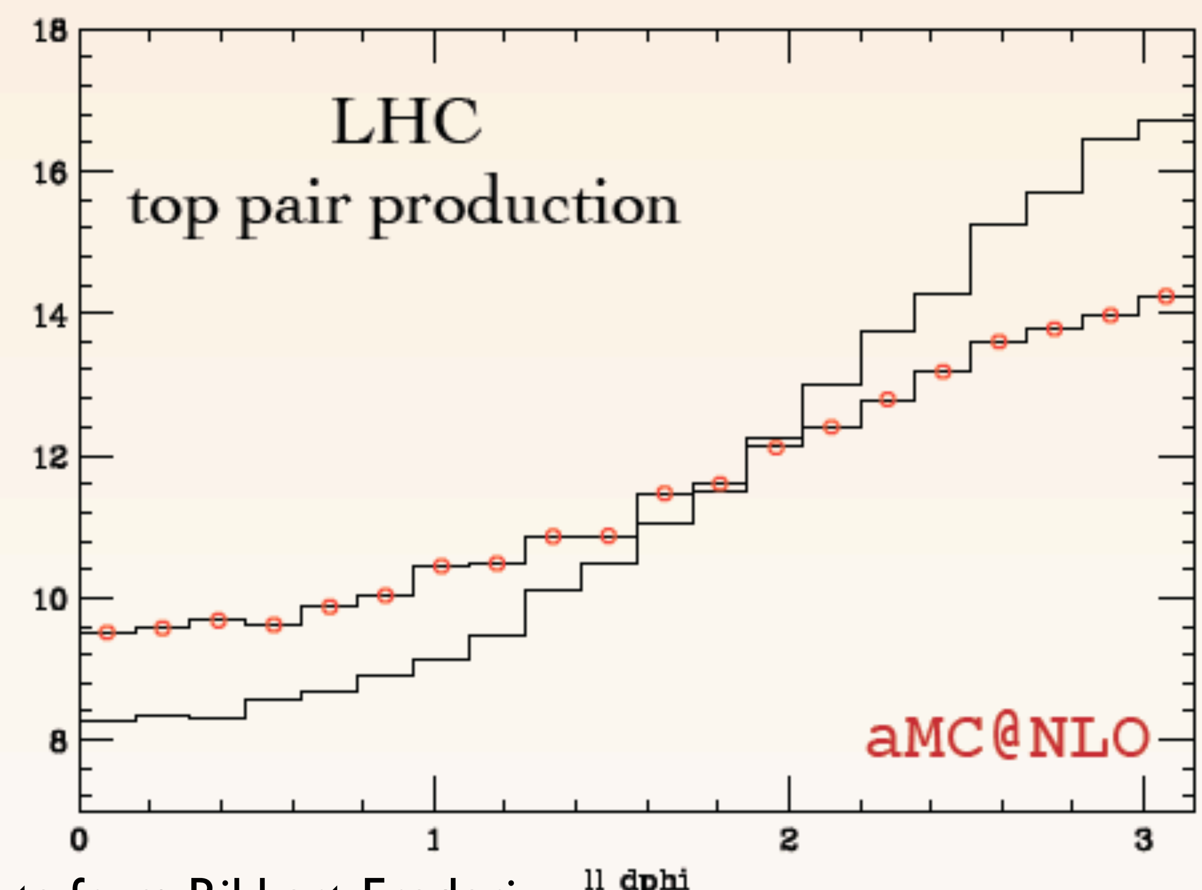
- How to deal with MC@NLO events?
- Spin correlations usually have tiny effects on observables
  - Include them at tree level
- For H (*n*+*1* body) events, use decayed real-emission matrix-element (generated on the fly with MG)
- For S (*n* body) events, use decayed born matrix-element
- This guarantees NLO accuracy for observables related to production (e.g. top pt)
- This includes all spin correlation for observables related to production + decay (apart non-factorizable ones)

The decay package: validation

Frixione, Laenen, Motylinski & Webber, hep-ph/0702198

aMC@NLO+DecayPackage



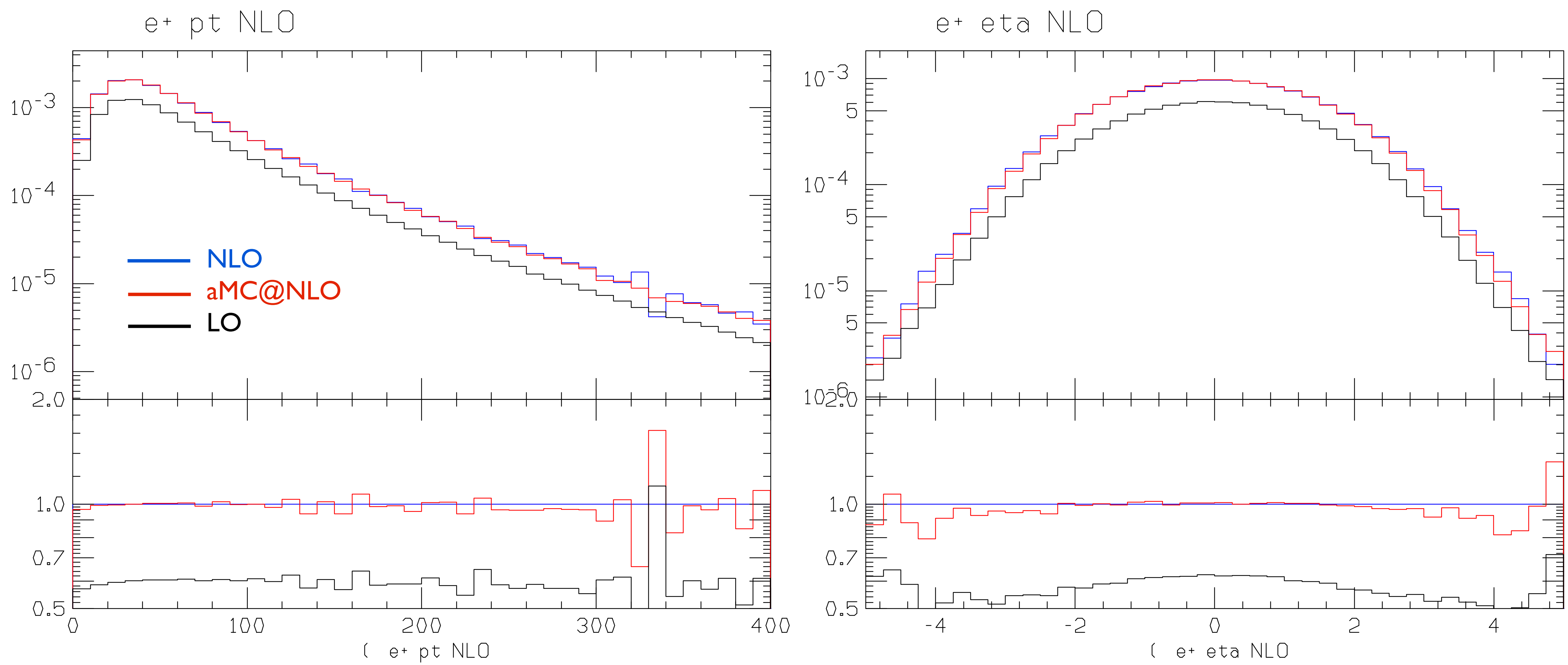


Plots from Rikkert Frederix

How to?

```
./bin/mg5
> generate p p > t t~ w+ [QCD]
> output myppttw
> quit
cd myppttw
./bin/compile_amcatnlo.sh
./bin/run_amcatnlo.sh
```

How to?



Integration/event generation and analyse made between yesterday and this morning!

Conclusions

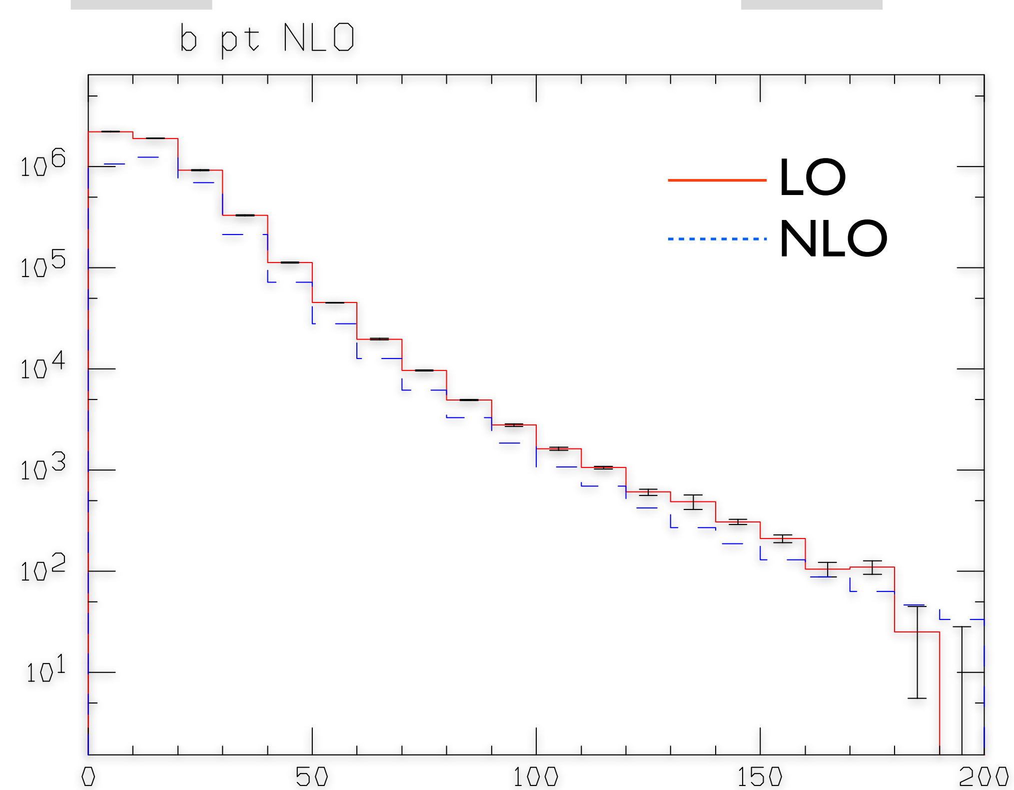
- aMC@NLO can generate events for any SM process at NLO accuracy, matched to parton showers
  - Only limited by CPU speed
- Automation: reliable results, more time do do real physics
- Rewritten into MG5 framework
  - All v4 limitations removed
- Use any model via the FR/UFO/ALOHA interface (work in progress...)
- Decay package to include spin correlations @NLO
- Code publicly available soon!
- Stay tuned on http://amcatnlo.cern.ch

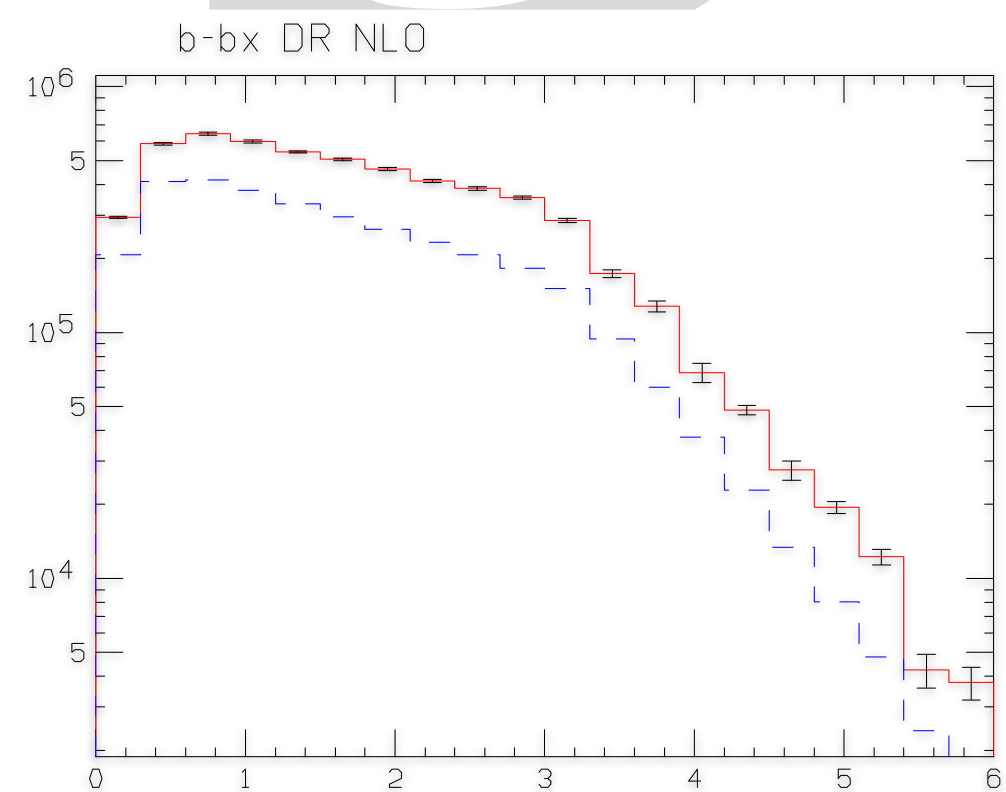
Backup slides

- Parton level results

bbj@NLO

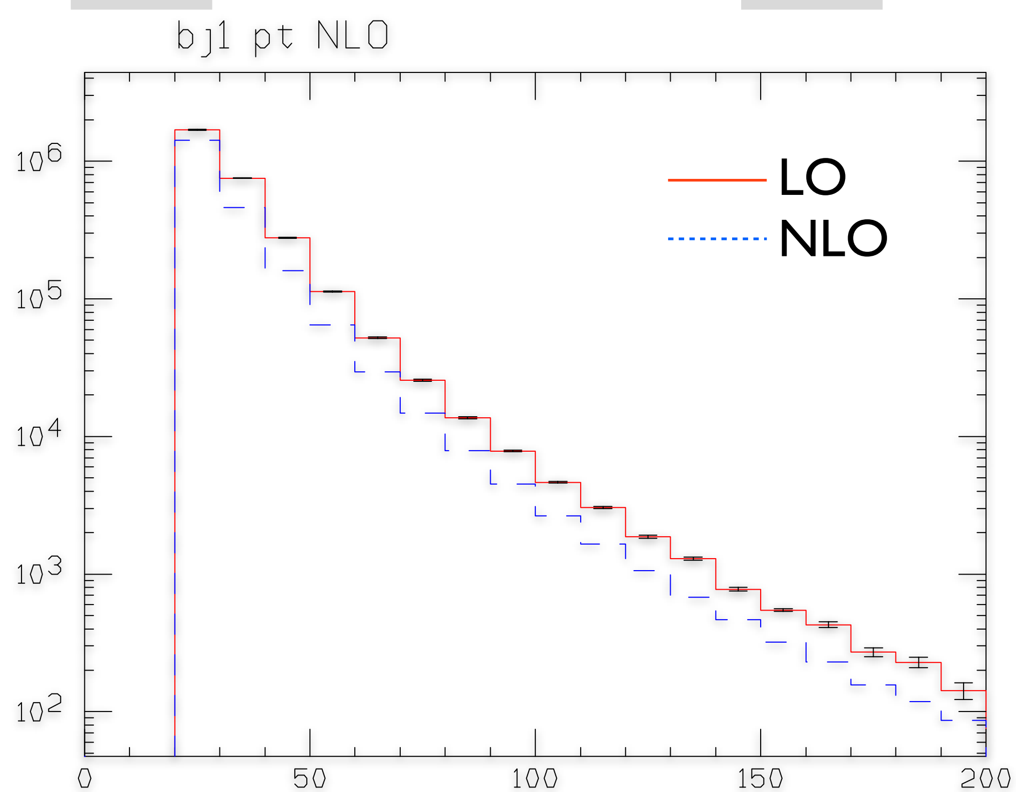
- Virtuals from Dittmaier, Weinzerl, Uwer (arXiv:0810.0452)
  - validated against MadLoop5
- 4 flavour scheme (and PDF)
- Kt algo for jets, R=0.5, pt > 20 GeV
- K factor ~ 2

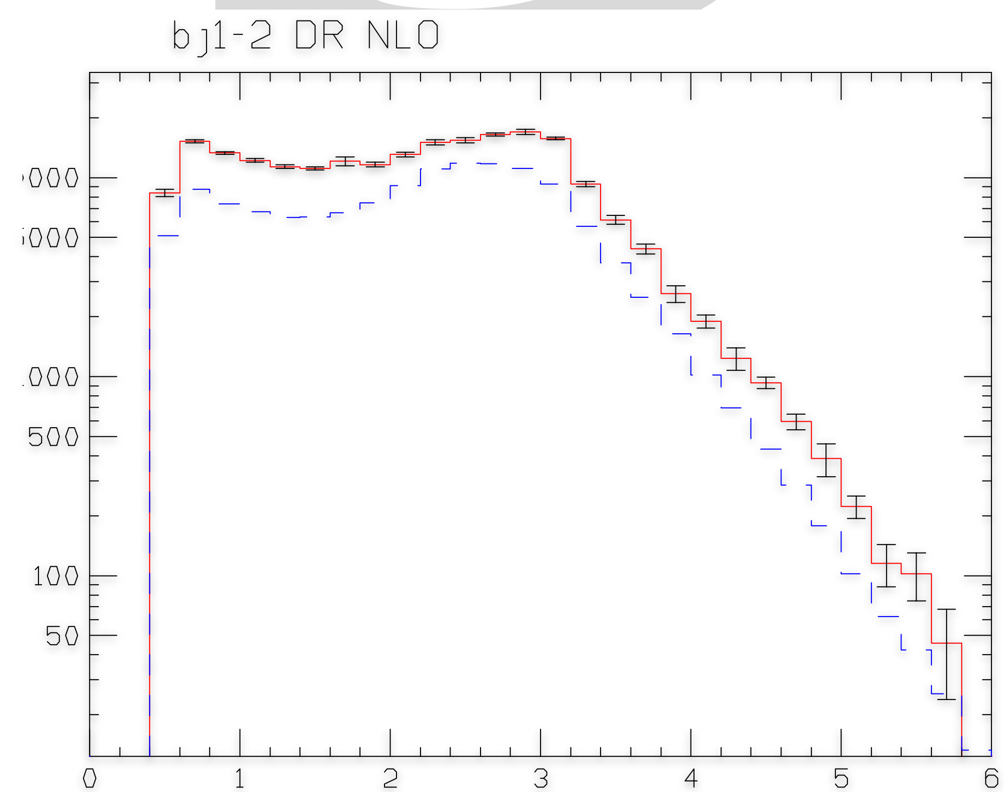




bbj@NLO

- Virtuals from Dittmaier, Weinzerl, Uwer (arXiv:0810.0452)
  - validated against MadLoop5
- 4 flavour scheme (and PDF)
- Kt algo for jets, R=0.5, pt > 20 GeV
- K factor ~ 2





Wbj@NLO

- Whole code from aMC@NLO
- 5 flavour scheme (j can be b)
  - Need to impose special cuts in order to have finite cross-section
    - Ask for >2 jets, at least one containing a b, but not bb