Status and prospects of aMC@NLO

Rikkert Frederix
University of Zurich

FeynRules 2012 workshop: towards NLO, Mont Sainte-Odile, France, March 25-30, 2012

Leading order

- For many of the theory predictions needed in the searches for new physics as well as measuring properties of the SM, tree-level matrix element generators (Alpgen, MadGraph/ MadEvent, Sherpa) plus multi-parton merging techniques (CKKW-L, MLM) are used
- The reasons for this are clear:
  - In many regions of phase-space they do a very good job, in particular for shapes of distributions
  - Parton showers and hadronizations models are tuned to data
  - Many flexible lowest order (LO) tools are readily available and easy to use

Why we need NLO...

- Why even bother to go beyond LOwPS matching?
  Some typical motivations (by theorists) for (parton level) NLO are:
  - Better description of jet structure
    (2 partons instead of 1)
  - New channels opening up
    (e.g. qg vs gg initial states)
  - "NLO" effects on distributions
    (e.g. kinematics dependent K-factor)
- However, these are actually motivations to do tree-level parton-shower matching. Genuine NLO effects are not an issue

Why we really need NLO

- Genuine reasons to go to (N)NLO:
  - Total rates are much better described
  - Reduced theoretical uncertainties due to meaningful scale dependence
  - Proper estimate of the PDF uncertainties
  - Description of pure higher order effects*
    (like ttbar Forward-Backward asymmetry)

*although, one might consider this to be LO

Why we need NLOwPS

- When can we better make use of NLOwPS predictions?
  - Large backgrounds that are difficult to normalize to data
  - When multivariate techniques (e.g. Boosted decision trees, Neural Network) are essential
  - Over-stretching fixed-order results can give biases
  - It doesn't hurt using them (when they are easy and quick to run)
- So, in general, when precision is an issue and results dependent on theoretical input.

NLOwPS allows one to use NLO information in all aspects of an experimental analysis

Why we need NLOwPS

- Some prime examples are:
  - b-tagging: same as in experiment, for processes in which NLO results are desirable (e.g. Vector boson plus heavy flavor)
  - Behavior of extra jets (described by perturbative or non-perturbative physics). Typical example is jet-veto systematics
  - Description of genuine NLO effects. Typical example is top-antitop forward-backward asymmetry

NLOwPS: IMPLEMENTATION

- The three difficulties of NLOwPS event generation
  - Virtual amplitudes: how to compute the loops automatically in a reasonable amount of time
  - How to deal with infra-red divergences and phase-space integration in an efficient way: virtual corrections and real-emission corrections are separately divergent and only their sum is finite (for IR-safe observables) according to the KLN theorem
  - How to match these processes to a parton shower without double counting
- All three implemented in the automatic `aMC@NLO` package!

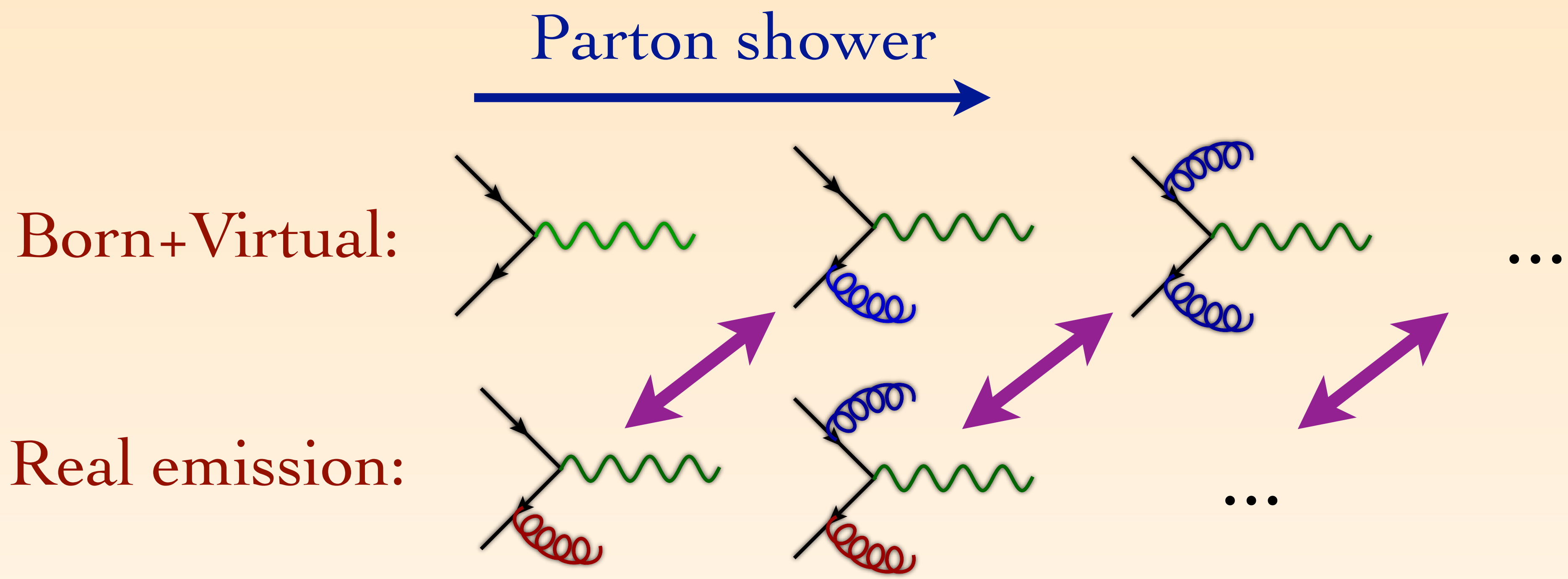
Virtual corrections

- MadLoop [Hirschi, RF, Frixione, Garzelli, Maltoni, Pittau (2011)] uses the OPP method [Ossola, Papadopoulos & Pittau (2006)] as implemented in CutTools [Ossola, Papadopoulos & Pittau (2007)] to compute virtual contributions from tree-level diagrams
- Based on setting up a system of linear equations to find the coefficients in front of the basis of scalar integrals by sampling the integrand
- Needs special treatment to get also the rational term
- Completely general (and numerical) method
- More in Valentin's presentation about current status and prospects for MadLoop in MadGraph 5

Factoring IR poles

- The MadFKS [RF., Frixione, Maltoni & Stelzer (2009)] code uses the FKS subtraction scheme [Frixione, Kunszt, Signer (1995)] to factor the soft and collinear poles out of the phase-space integrals and cancel them against the poles from the virtual corrections
- Based on splitting the phase-space integrals in regions in which there is (maximally) one collinear and one soft divergence
- Allows for optimized numerical phase-space integration
- Parallel in nature: can make use of many CPUs simultaneously to speed-up the calculation
- Process independent & Model independent

Matching to the Parton Shower



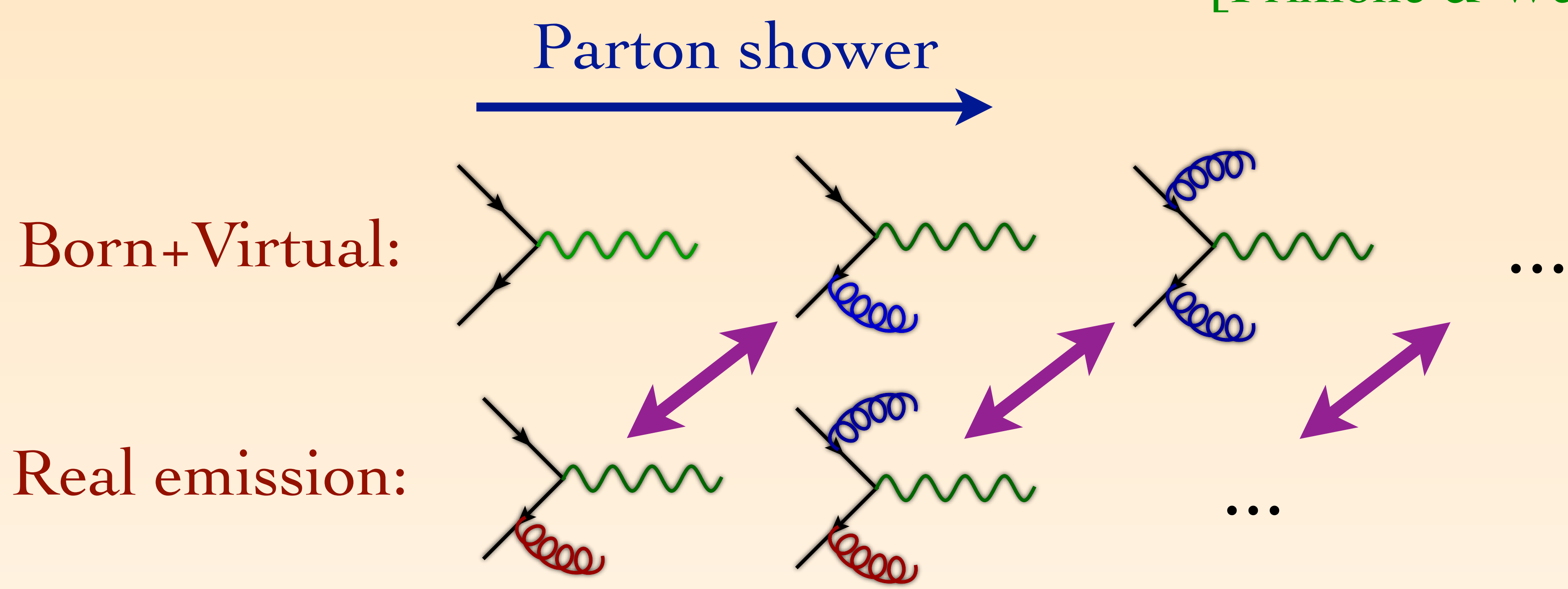
- There is double counting between the real emission matrix elements and the parton shower: the extra radiation can come from the matrix elements or the parton shower
- There is also an overlap between the virtual corrections and the Sudakov suppression in the zero-emission probability

Double counting in virtual/Sudakov

- The Sudakov factor Δ (which is responsible for the resummation of all the radiation in the shower) is the no-emission probability
- It's defined to be $\Delta = 1 - P$, where P is the probability for a branching to occur
- By using the conservation of probability in this way, Δ contains contributions from the virtual corrections implicitly
- Because at NLO the virtual corrections are already included via explicit matrix elements, Δ is double counting with the virtual corrections
- In fact, because the shower is unitary, what we are double counting in the real emission corrections is exactly equal to what we are double counting in the virtual corrections (but with opposite sign)!

MC@NLO procedure

[Frixione & Webber (2002)]



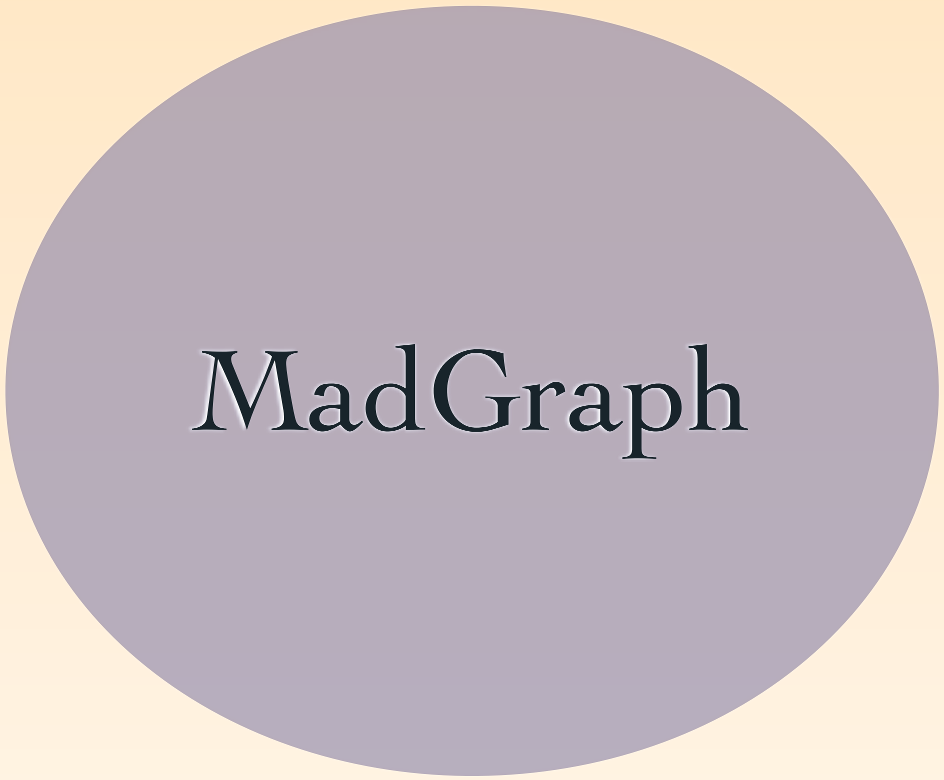
$$\frac{d\sigma_{\mathrm{MC@NLO}}}{dO} = \left[d\Phi_m \left(B + \int_{\mathrm{loop}} V + \int d\Phi_1 MC \right) \right] \mathcal{F}_{\mathrm{MC}}^{(m)} + \left[d\Phi_{m+1} (R - MC) \right] \mathcal{F}_{\mathrm{MC}}^{(m+1)}$$

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

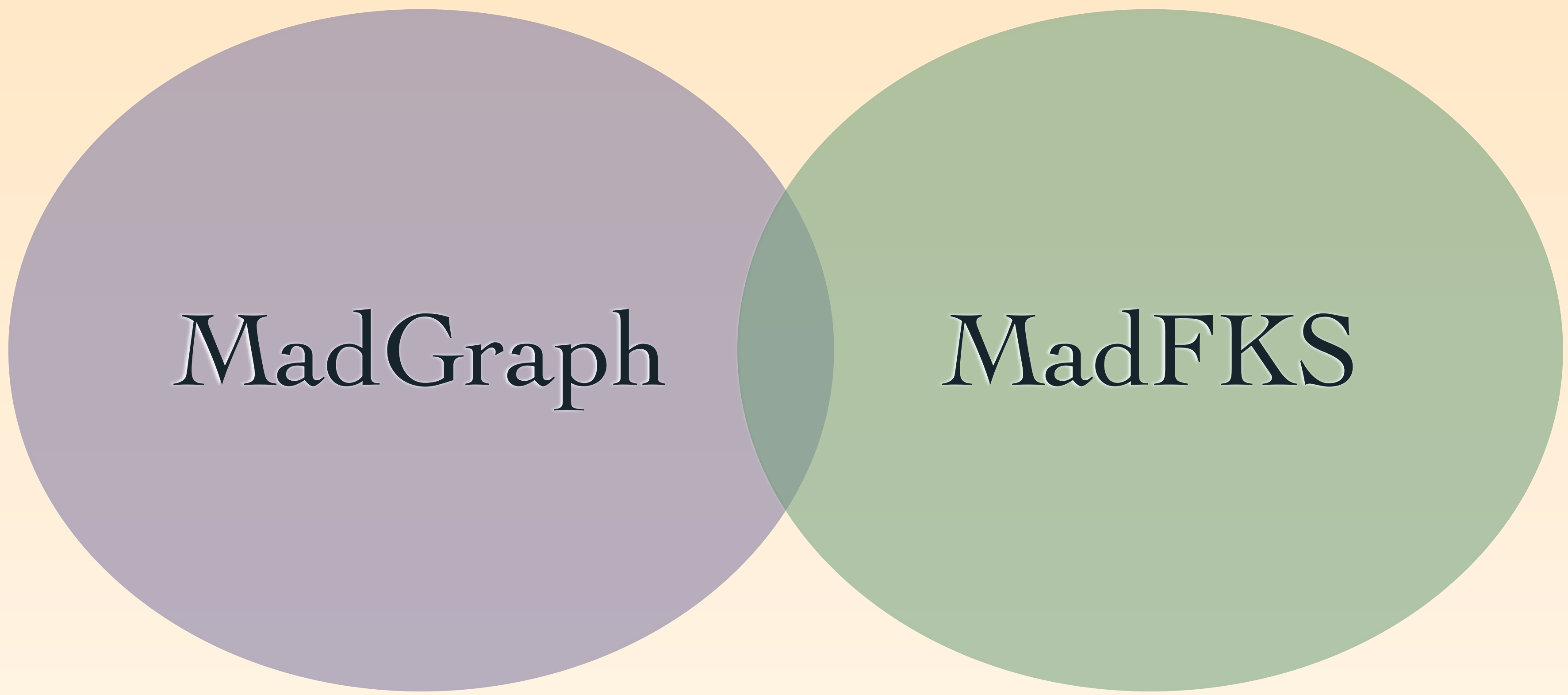
Current status of MC subtraction

- The MC subtraction terms are Shower Monte Carlo specific: each partons shower needs different subtraction terms
- Current status of aMC@NLO is
  - aMC@NLO/Herwig6: working and fully tested
  - aMC@NLO/Pythia6 (Q^2-ordered): working and well-tested
  - aMC@NLO/Pythia6 (p_T-ordered): initial state implemented, final state is work in progress. High priority
  - aMC@NLO/Pythia8: initial state implemented, final state is work in progress. High priority
  - aMC@NLO/Herwig++: all implemented but final state needs still validation. Lower priority

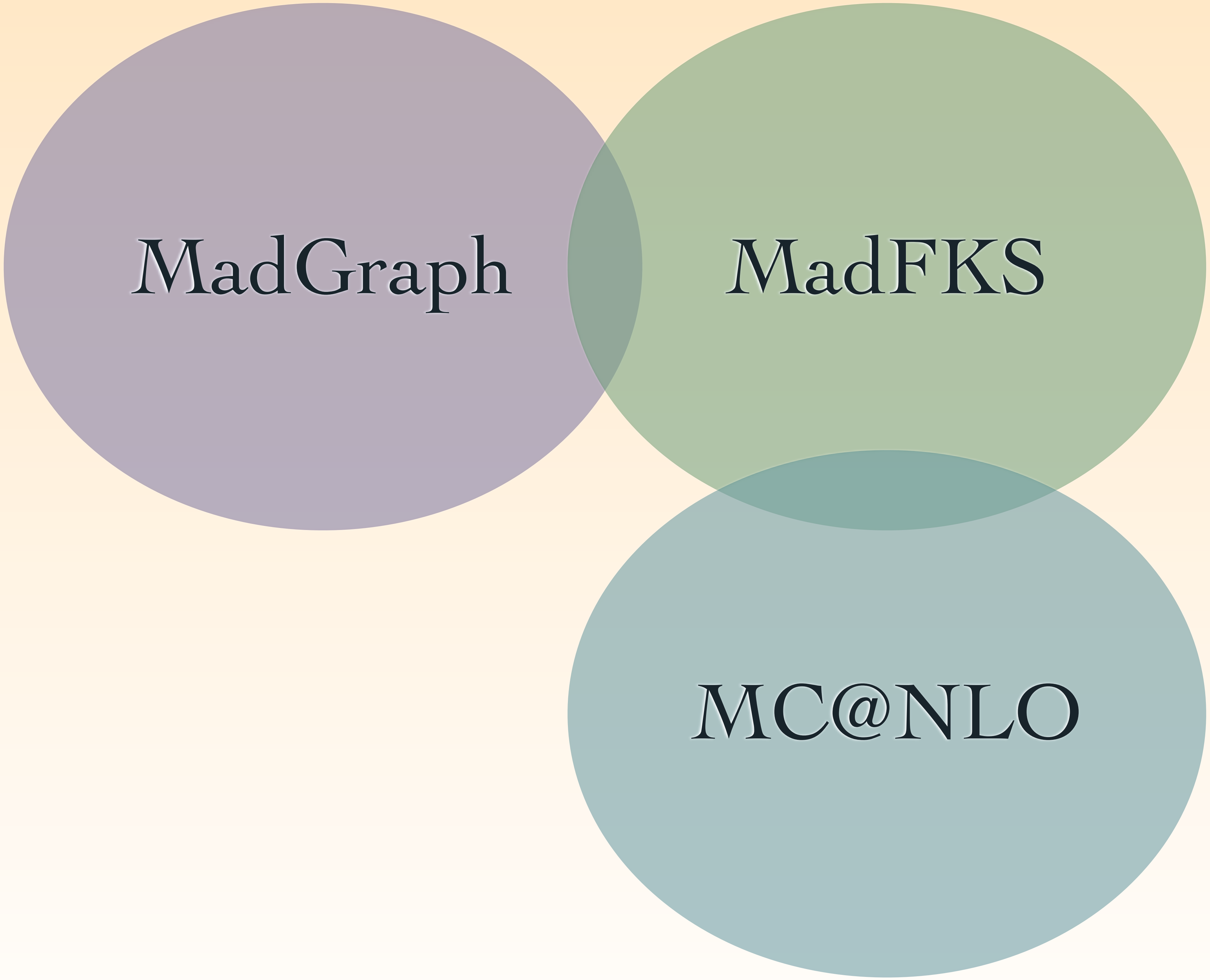
The aMC@NLO code



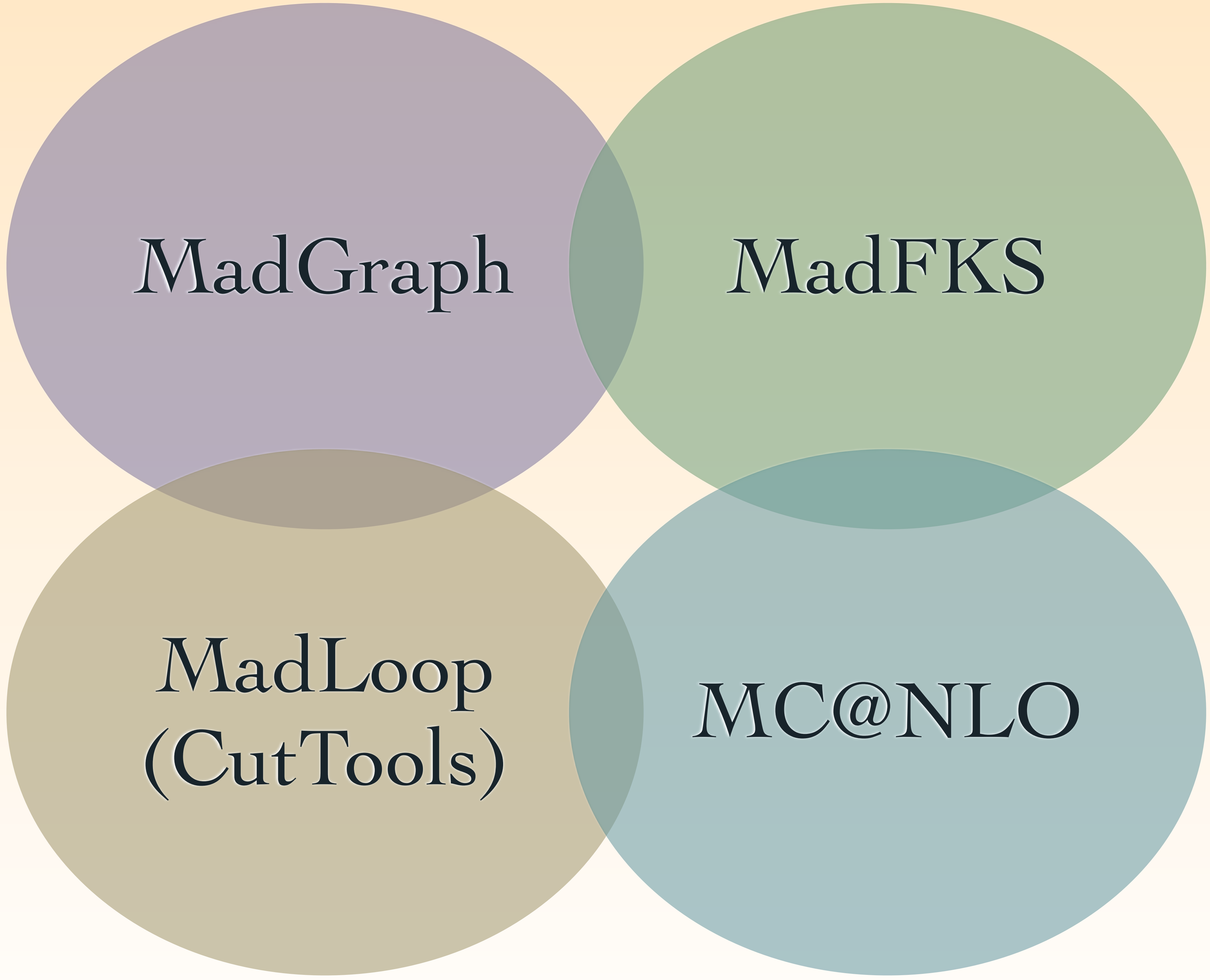
The aMC@NLO code



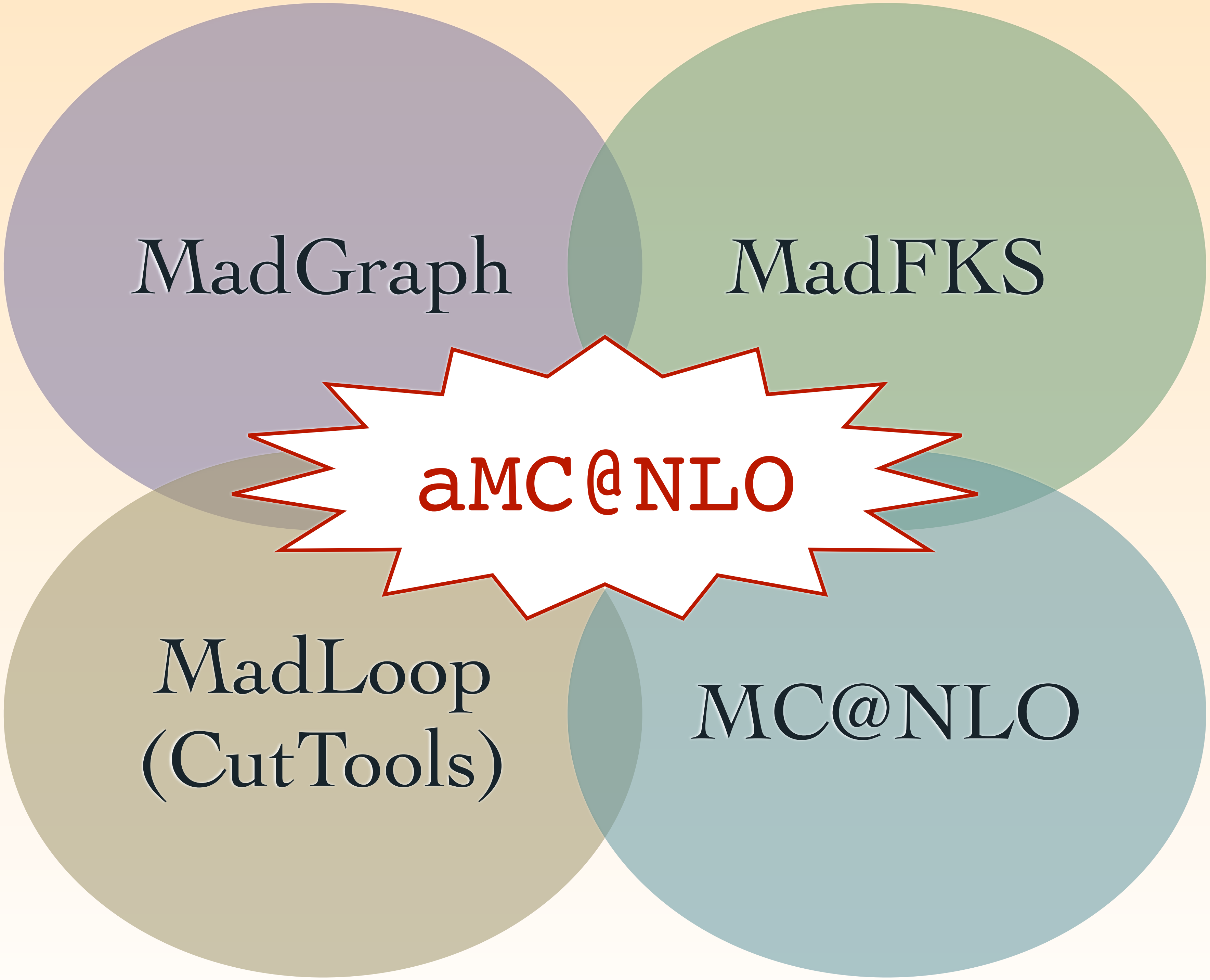
The aMC@NLO code



The aMC@NLO code



The aMC@NLO code



http://amcatnlo.cern.ch

Scale dependence & PDF uncertainties

Scale dependence and PDF uncertainties

- Any short-distance cross section can be written as a linear combination of scale and PDF dependent terms, with coefficients independent of both scales and PDFs.
- Therefore, saving these coefficients in the event file allows for a posterior evaluation of scale and PDF uncertainties, by evaluating their dependence event-by-event, without needing to rerun the generation of the events

Reweighting at LO

- Straight-forward at LO
  - Factorization scale only enters PDFs
  - Renormalization scale only enters in alpha_s

$$f_a(\mu_F) \otimes f_b(\mu_F) \otimes \alpha_s(\mu_R)^b |\overline{M}|^2$$

- So, we can simply reweight event-by-events with the factor

$$\mathcal{R}_i = \frac{f'_1(x_{1;i},\mu'_F) f'_2(x_{2;i},\mu'_F) g_S^{2b}(\mu'_R)}{f_1(x_{1;i},\mu_F) f_2(x_{2;i},\mu_F) g_S^{2b}(\mu_R)}$$

to get the scale & PDF dependence

Reweighting at NLO

- A bit more involved at NLO
  - Scales also enters the process explicitly

$$\mathcal{R}_i^{(\alpha)} = f_1'(x_{1;i}^{(\alpha)}, \mu_F'^{(\alpha)}) f_2'(x_{2;i}^{(\alpha)}, \mu_F'^{(\alpha)}) \Bigg[g_S^{2b+2}(\mu_R'^{(\alpha)}) \left(\widehat{W}_0^{(\alpha)}(\mathcal{K}_{n+1;i}^{(\alpha)}) + \widehat{W}_F^{(\alpha)}(\mathcal{K}_{n+1;i}^{(\alpha)}) \log \frac{{\mu_F'^{(\alpha)}}^2}{Q^2} + \widehat{W}_R^{(\alpha)}(\mathcal{K}_{n+1;i}^{(\alpha)}) \log \frac{{\mu_R'^{(\alpha)}}^2}{Q^2} \right) + g_S^{2b}(\mu_R'^{(\alpha)}) \widehat{W}_B(\mathcal{K}_{n+1;i}^{(\alpha)}) \delta_{\alpha S} \Bigg] \Bigg/ \frac{d\sigma^{(\mathrm{NLO},\alpha)}}{d\mu_{Bj} d\mu_{n+1}}(\mathcal{K}_{n+1;i}^{(\alpha)}, x_{1;i}^{(\alpha)}, x_{2;i}^{(\alpha)}) \,. \tag{1.19}$$

- In aMC@NLO it is even a bit more involved...

Reweighting at MC@NLO

$$\mathcal{R}_i^{(\mathbb{H})} = \Bigg\{ f_1'(x_{1;i}^{(E)}, \mu_F'^{(E)}) f_2'(x_{2;i}^{(E)}, \mu_F'^{(E)}) g_S^{2b+2}(\mu_R'^{(E)}) \times \left[\widehat{W}_0^{(E)}(\mathcal{E}_{\mathbb{H};i}) + \widehat{W}_F^{(E)}(\mathcal{E}_{\mathbb{H};i}) \log\left(\frac{\mu_F'^{(E)}}{Q}\right)^2 + \widehat{W}_R^{(E)}(\mathcal{E}_{\mathbb{H};i}) \log\left(\frac{\mu_R'^{(E)}}{Q}\right)^2 \right] - \sum_c f_1'(x_{1;i}^{(\mathrm{MC},c)}, \mu_F'^{(E)}) f_2'(x_{2;i}^{(\mathrm{MC},c)}, \mu_F'^{(E)}) g_S^{2b+2}(\mu_R'^{(E)}) w^{(\mathrm{MC},c)} \Bigg\} \Bigg/ \frac{d\sigma^{(\mathbb{H})}}{d\chi_{Bj} d\chi_{n+1}}(\mathcal{E}_{\mathbb{H};i}), \quad (2.31)$$

$$\mathcal{R}_i^{(\mathbb{S})} = \Bigg\{ \sum_c f_1'(x_{1;i}^{(\mathrm{MC},c)}, \mu_F'^{(E)}) f_2'(x_{2;i}^{(\mathrm{MC},c)}, \mu_F'^{(E)}) g_S^{2b+2}(\mu_R'^{(E)}) w^{(\mathrm{MC},c)} + \sum_{\alpha=S,C,SC} f_1'(x_{1;i}^{(\alpha)}, \mu_F'^{(S)}) f_2'(x_{2;i}^{(\alpha)}, \mu_F'^{(S)}) \Bigg[g_S^{2b+2}(\mu_R'^{(S)}) \left(\widehat{W}_0^{(\alpha)}(\mathcal{E}_{\mathbb{S};i}) + \widehat{W}_F^{(\alpha)}(\mathcal{E}_{\mathbb{S};i}) \log\left(\frac{\mu_F'^{(S)}}{Q}\right)^2 + \widehat{W}_R^{(\alpha)}(\mathcal{E}_{\mathbb{S};i}) \log\left(\frac{\mu_R'^{(S)}}{Q}\right)^2 \right) + g_S^{2b}(\mu_R'^{(S)}) \widehat{W}_B(\mathcal{E}_{\mathbb{S};i}) \delta_{\alpha S} \Bigg] \Bigg\} \Bigg/ \frac{d\sigma^{(\mathbb{S})}}{d\chi_{Bj} d\chi_{n+1}}(\mathcal{E}_{\mathbb{S};i}). \quad (2.32)$$

More details can be found in

[*RF, Frixione, Hirschi, Maltoni, Pittau, Torrielli, 1110.4783*]

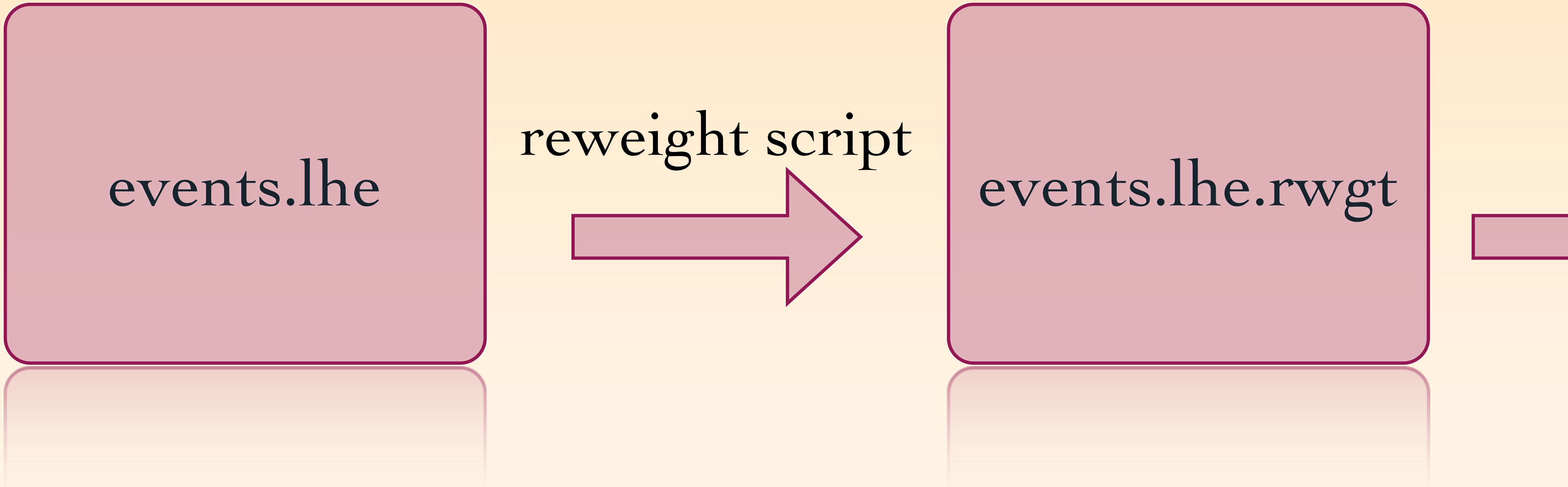
In practice: NLO event file

```
<event>
8      66 0.21341783D-07 0.17630329D+03 0.75467723D-02 0.12982704D+00
       5 -1     0     0   501     0 0.00000000D+00 0.00000000D+00 0.35995691D+02 0.36334450D+02 0.49500000D+01 0.0000D+00
      -5 -1     0     0     0   502 0.00000000D+00 0.00000000D+00 -.21452182D+03 0.21457892D+03 0.49500000D+01 0.0000D+00
      23  2     1     2     0     0 -.36758481D+01 -.29575520D+02 -.16296618D+03 0.21728455D+03 0.14059294D+03 0.0000D+00
     -11  1     3     3     0     0 0.38255724D+01 -.48895441D+01 -.47220436D+01 0.78000218D+01 0.00000000D+00 0.0000D+00
      11  1     3     3     0     0 0.28490378D+02 0.41808349D+02 -.81435178D+02 0.95871413D+02 0.00000000D+00 0.0000D+00
     -13  1     3     3     0     0 -.14251913D+02 -.58912946D+02 -.57843590D+02 0.83783847D+02 0.00000000D+00 0.0000D+00
      13  1     3     3     0     0 -.21739885D+02 -.75813794D+01 -.18965366D+02 0.29829265D+02 0.00000000D+00 0.0000D+00
      21  1     1     2   501   502 0.36758481D+01 0.29575520D+02 -.15559952D+02 0.33628825D+02 0.75000000D+00 0.0000D+00
# 2  7  2  2  1 0.29803074D+02 0.29803074D+02 3  7  0 0.00000000D+00 0.00000000D+00 0.00000000D+00 0.00000000D+00 0.000
<rwgt>
0.81082415D-04 0.00000000D+00
0.10340521D-01 0.61345451D-01 0.00000000D+00 0.00000000D+00 0.00000000D+00 0.00000000D+00 0.00000000D+00 0.00000000D+0
0.95757349D+02 0.00000000D+00 0.00000000D+00 0.95757349D+02
0.81150042D+02 0.00000000D+00 0.00000000D+00 -.81150042D+02
0.64382801D+01 0.38255724D+01 -.48895441D+01 0.17055220D+01
0.52604721D+02 0.28490378D+02 0.41808349D+02 -.14408920D+02
0.61081025D+02 -.14251913D+02 -.58912946D+02 0.75524433D+01
0.23593055D+02 -.21739885D+02 -.75813794D+01 0.51509551D+01
0.33190310D+02 0.36758481D+01 0.29575520D+02 0.14607308D+02
0.70296472D+02 0.00000000D+00 0.00000000D+00 0.70296472D+02
0.70296472D+02 0.00000000D+00 0.00000000D+00 -.70296472D+02
0.56527889D+01 0.39818980D+01 -.36317634D+01 0.17055220D+01
0.63313468D+02 0.29989085D+02 0.53866808D+02 -.14408920D+02
0.49672604D+02 -.12819978D+02 -.47391733D+02 0.75524433D+01
0.21954084D+02 -.21151006D+02 -.28433110D+01 0.51509551D+01
0.00000000D+00 0.00000000D+00 0.00000000D+00 0.00000000D+00
0.4567946201D-15 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00
0.0000000000D+00 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00
0.00000000D+00 0.00000000D+00 0.00000000D+00 0.00000000D+00
</rwgt>
</event>
```

Coefficients needed to do the reweighting.

From these coefficients, the event weight can be computed for any new functional form for the scales, or any PDF set

Reweight script



- The reweight script reads the event file and computes the new scale and PDF dependence
- No new matrix element evaluations and therefore very quick (reading/writing the event files takes the most amount of time...)

NLO EVENT FILE

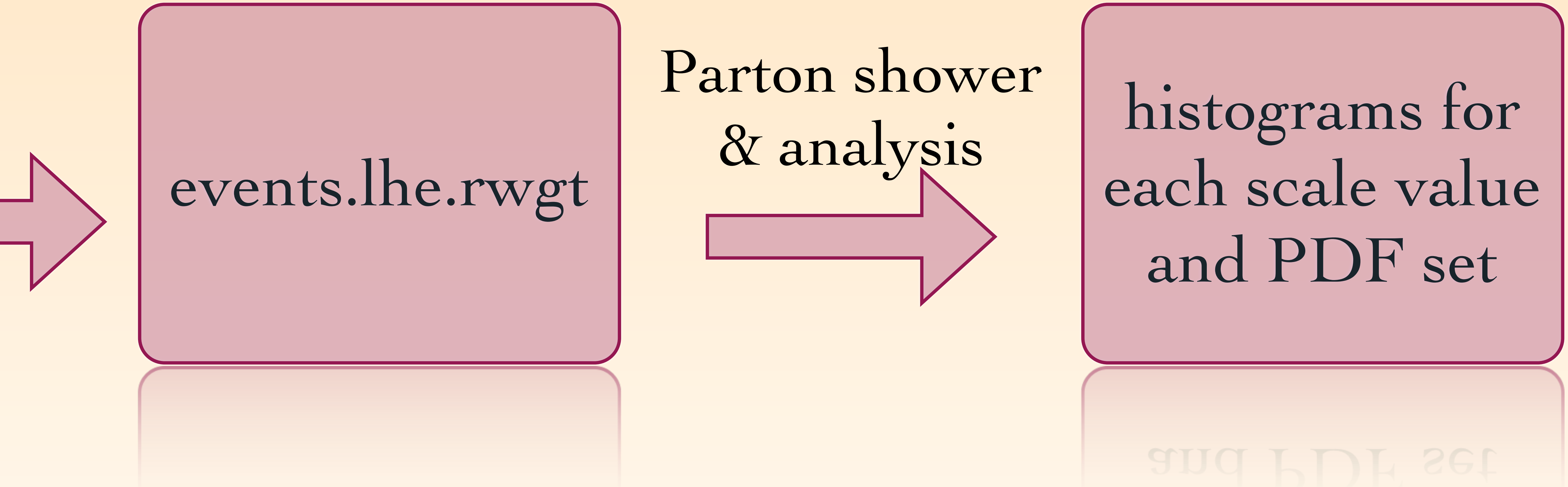
```
 <event>
 8      66 0.21943624E-07 0.67725434E+02 0.75467723E-02 0.10678127E+00
       -2 -1     0     0     0   501 0.00000000E+00 0.00000000E+00 0.21865056E+02 0.21867397E+02 0.32000000E+00 0.0000E
        2 -1     0     0   501     0 0.00000000E+00 0.00000000E+00 -.48855436E+03 0.48855447E+03 0.32000000E+00 0.0000E
       23  2     1     2     0     0 0.16090158E+02 0.31869394E+01 -.12083553E+03 0.15224618E+03 0.91151646E+02 0.0000E
       23  2     1     2     0     0 -.16090158E+02 -.31869394E+01 -.34585377E+03 0.35817569E+03 0.91683921E+02 0.0000E
      -11  1     4     4     0     0 0.16220397E+01 -.23289878E+02 -.32030340E+03 0.32115310E+03 0.00000000E+00 0.0000E
       11  1     4     4     0     0 -.17712197E+02 0.20102938E+02 -.25550375E+02 0.37022584E+02 0.00000000E+00 0.0000E
      -13  1     3     3     0     0 0.12060879E+02 -.42965600E+02 -.73678665E+02 0.86139731E+02 0.00000000E+00 0.0000E
       13  1     3     3     0     0 0.40292781E+01 0.46152539E+02 -.47156870E+02 0.66106447E+02 0.00000000E+00 0.0000E
# 1   7   1   1   2 0.23702464D+01 0.23702464D+01 8   0   0 0.99999991D+00 0.84548723D+00 0.11830766D+01 0.98156783D+00 0
 <rwgt>
0.10306710E-01 0.10306709E-01    3   20
0.10306709E-01 0.90862088E-02 0.11513117E-01
0.10871266E-01 0.95373645E-02 0.12193627E-01
0.98411827E-02 0.87141917E-02 0.10951977E-01
0.10313303E-01 0.10301480E-01
0.10321776E-01 0.10289802E-01
0.10277207E-01 0.10350505E-01
0.10293111E-01 0.10315859E-01
0.10316430E-01 0.10293186E-01
0.10357758E-01 0.10275027E-01
0.10321112E-01 0.10287327E-01
0.10368743E-01 0.10284177E-01
0.10335460E-01 0.10243760E-01
0.10319797E-01 0.10294243E-01
0.10172879E-01 0.10545410E-01
0.10382578E-01 0.10259974E-01
0.10286192E-01 0.10316187E-01
0.10321502E-01 0.10305549E-01
0.10339783E-01 0.10241545E-01
0.10327529E-01 0.10298882E-01
0.10288284E-01 0.10319506E-01
0.10288937E-01 0.10317910E-01
0.10252000E-01 0.10321048E-01
0.10336564E-01 0.10298902E-01
 </rwgt>
 </event>
```

Scale uncertainties
(3x3 dependence on fac and ren scales)

PDF uncertainties
(40 MSTW error sets)

Cannot simply save max and min: dependence is correlated within a given bin of a distribution.
First the plots need to be filled before the dependence can be computed

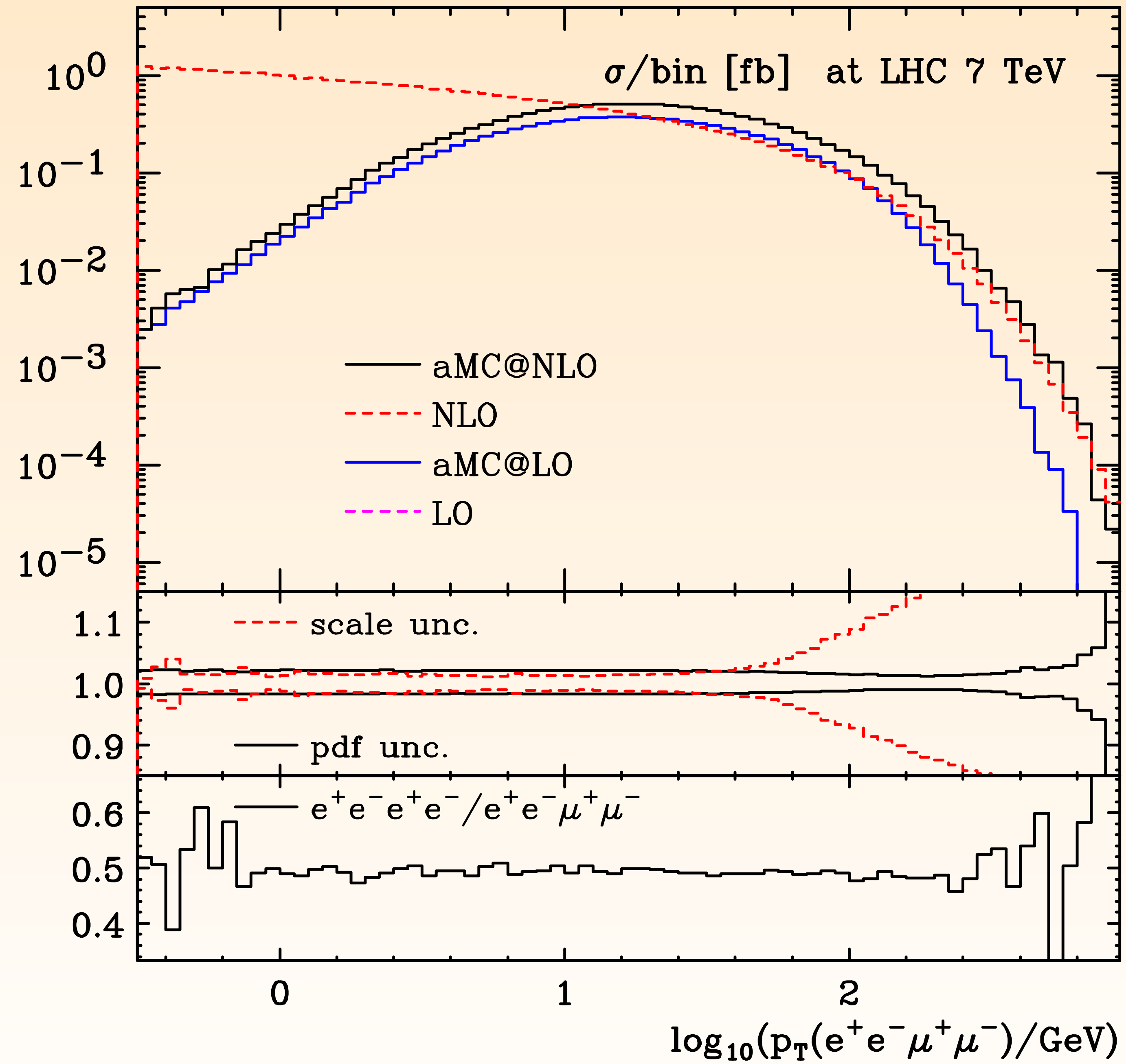
Filling histograms



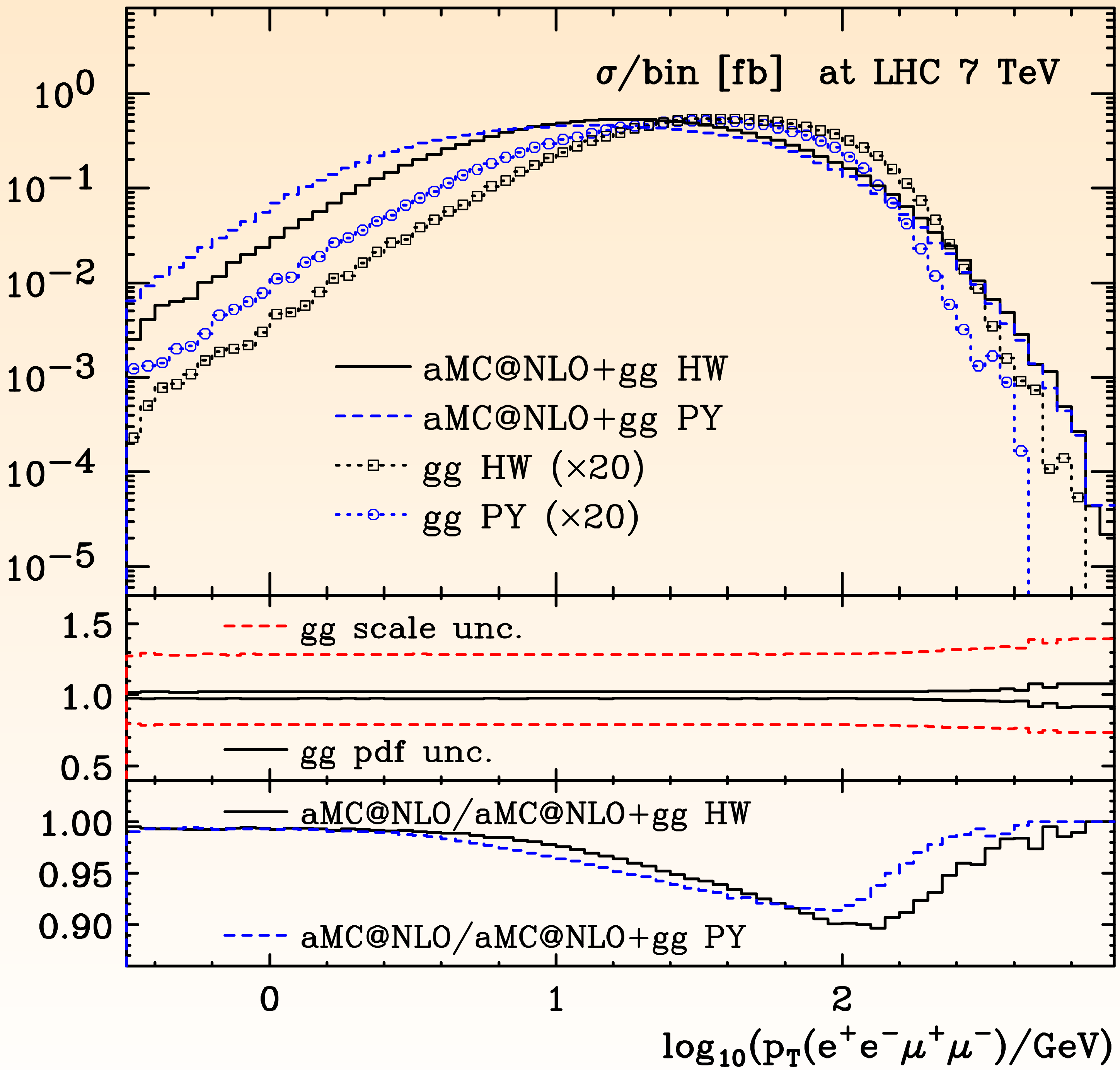
- Shower the LHE events as usual, but fill a separate histogram for each of the values of the scales and PDF set
- Compute --from the final set of histograms-- the uncertainties bin-by-bin

Results
4-lepton production

- Scale dependence and PDF uncertainties computed without extra CPU cost
- Statistical fluctuations are correlated: cleaner extraction of uncertainties
- Low-p_T region dominated by S-events (NLO scale dependence), high-p_T region by H-events ("LO+1j" scale dependence)



Results
4-lepton production



- Scales and PDFs not the only sources of uncertainties
- Dependence on the shower can be significant; in particular in the region of phase-space where the Sudakov dominates

aMC@NLO in MG5

aMC@NLO in MadGraph 5

- Marco Zaro has rewritten MadFKS in Madgraph 5:
  - "MadFKS from real" is identical in structure and function as current MadFKS for MadGraph 4. Working without problems, but not as well tested yet
  - "MadFKS from Born" allows for more efficient combination of integration channels, reducing one of the major limitations of current MadFKS. In particular, it allows for a Monte-Carlo sum over the real-emission processes (with FKS damping) contributing to a single Born process.
    - Still needs to be tested and validated. Unfortunately, first tests not as promising as I had hoped for...
- No complications for aMC@NLO (structure identical to MadFKS: if MadFKS is working, so is aMC@NLO)

Running aMC@NLO v5

- Generation of the process is similar to LO MadGraph5:

With MadLoop	Without MadLoop (real-emission corrections only)
set fks_mode born	set fks_mode born
import model loop_sm	import model sm (or any other model)
generate p p > e+ ve [QCD]	generate p p > e+ ve [real=QCD]
output PROCESS_DIR	output PROCESS_DIR
quit	quit

And run in the process directory itself. No ‘launch’ command yet.

Summary

- Current aMC@NLO up and running smoothly in MadGraph v4:
  - MadFKS for factoring IR singularities
  - MadLoop for the virtual corrections
  - Shower subtraction terms implemented for Herwig6 and Pythia6 (Q^2), and ongoing for Herwig++, Pythia6 (p_T) and Pythia8
- MadFKS (and therefore also aMC@NLO) has been rewritten in MadGraph 5. “MadFKS from Born” reduces the number of integration channels enormously, but speed-up not so significant. **More testing for more non-trivial processes needed**

aMC@NLO website

aMC@NLO

http://amcatnlo.cern.ch

- On the aMC@NLO website you can find
  - Latest news on aMC@NLO
  - NLO event samples ready for showering and analysis
  - Compare with MadLoop: a single phase-space point for the virtual for any user-defined process in the SM. Useful for comparison/checking private calculations