Some studies with W + n (=1-4) jets with Blackhat+Sherpa (and ATLAS data)

J. Huston

Michigan State University

Editorial Comment

- Once we have the NLO calculations needed for the LHC, how do we (experimentalists) use them?
- If a theoretical calculation is done, but it can not be used by any experimentalists, does it make a sound?
- We need public programs and/or public ntuples
- Oftentimes, the program is too complex to be run by non-authors
- In that case, ROOT ntuples may be the best solution

Example

NLO with BlackHat+Sherpa

NLO cross section

Born — loop: lc and fmlc — real

$$\sigma_n^{NLO} = \int_n \sigma_n^{tree} + \int_n \left(\sigma_n^{virt} + \Sigma_n^{sub}\right) + \int_{n+1} \left(\sigma_{n+1}^{real} - \sigma_{n+1}^{sub}\right)$$

vsub

BlackHat

so this is not Sherpa the parton shower, but Sherpa used as a (very efficient) fixed order matrix element generator

Sherpa

How it's put together

NLO with BlackHat+Sherpa

NLO cross section

Born — loop: lc and fmlc — real

$$\sigma_n^{NLO} = \int_n \sigma_n^{tree} + \int_n \left(\sigma_n^{virt} + \Sigma_n^{sub}\right) + \int_{n+1} \left(\sigma_{n+1}^{real} - \sigma_{n+1}^{sub}\right)$$

vsub

for W+3 jets,
W+3 parton tree-level
matrix elements

the dipole subtraction terms evaluated in n-body phase space; to make matters more complex, vsub can be either + or -, compensated by other terms in the total cross section; note the sum over all quarks and antiquarks; makes matters more complex when coming to scale uncertainties

all of the real emission terms, (W+4 partons for W + 3 jets), modified by the dipole subtraction terms; divergences are gone

all of the virtual terms, both leading color and full-minus-leading color; the latter is typically a few % effect, but much of the complexity of the calculation

How it's put together

NLO with BlackHat+Sherpa

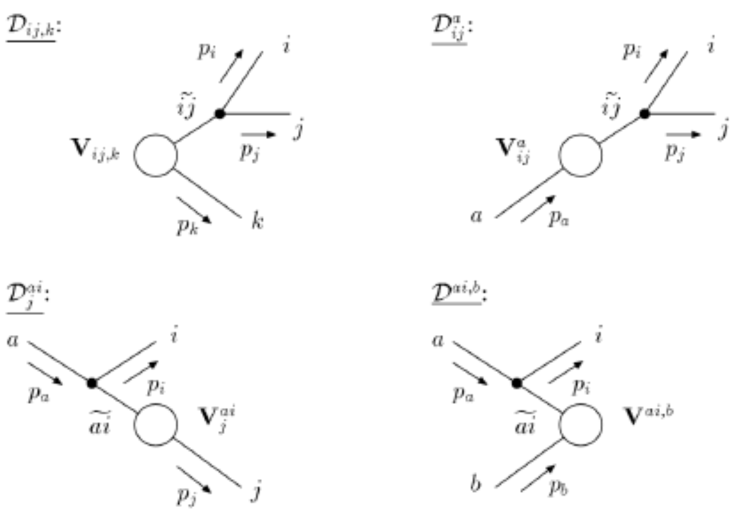
NLO cross section

Born — loop: lc and fmlc — real

$$\sigma_n^{NLO} = \int_n \sigma_n^{tree} + \int_n \left(\sigma_n^{virt} + \Sigma_n^{sub}\right) + \int_{n+1} \left(\sigma_{n+1}^{real} - \sigma_{n+1}^{sub}\right)$$

vsub

possible Catani-Seymour dipoles, for FF, FI,IF and II situations



the dipole subtraction terms evaluated in n-body phase space; to make matters more complex, vsub can be either + or -, compensated by other terms in the total cross section; note the sum over all quarks and antiquarks

all of the real emission terms, (W+4 partons for W + 3 jets), modified by the dipole subtraction terms; divergences are gone

many counterevents due to C-S dipoles that are correlated; have to use special weights/procedures to get correct statistical error

note the need for a 3rd parton, the 'spectator'; in the soft limit, it's the color partner

ROOT ntuples

- More complex to use than MCFM
  - no manual for example
  - and you don't produce the events yourself
- ntuples produced separately by Blackhat + Sherpa for →
- No jet clustering has been performed; that's up to the user
  - a difference from MCFM, where the program has to be re-run for each jet size/algorithm
- What algorithms/jet sizes that can be run depends on how the files were generated
  - i.e. whether the right counter-events are present
- For the files on the right at 7 TeV (for W^+ + 3 jets), one can use kT, antikT, siscone (f=0.75) for jet sizes of 0.4, 0.5, 0.6 and 0.7

- bornLO (stands alone for pure LO comparisons; not to be added with other contributions below)
  - 20 files, 5M events/file, 780 MB/file
- Born
  - 18 files, 5M events/file, 750 MB/file
- loop-lc (leading color loop corrections)
  - 398 files, 100K events/file, 19 MB/file
- loop-fmlc (needed for full color loop corrections)
  - 399 files, 15K events/file, 3 MB/file
- real (real emission terms)
  - 169 files, 2.5 M event/file, 5 GB/file
- vsub (subtraction terms)
  - 18 files, 10M events/file, 2.8 GB/file

Jet Clustering

- For jet clustering, we use SpartyJet, and store the jet results in SJ ntuples
  - and they tend to be big since we store the results for multiple jet algorithms/ sizes
- Then we friend the Blackhat +Sherpa ntuples with the SpartyJet ntuples producing analysis ntuples (histograms with cuts) for each of the event categories
- Add all event category histograms together to get the plots of relevant physical observables

http://projects.hepforge.org/spartyjet/

Branches in ntuple

branch name	type	Notes
id	I	id of the event. Real events and their associated counterterms share the same id. This allows for the correct treatment of statistical errors.
nparticle	I	number of particles in the final state
px	F[nparticle]	array of the x components of the final state particles
py	F[nparticle]	array of the y components of the final state particles
pz	F[nparticle]	array of the z components of the final state particles
E	F[nparticle]	array of the energy components of the final state particles
alphas	D	$alpha_s$ value used for this event
kf	I	PDG codes of the final state particles
weight	D	weight of the event
weight2	D	weight of the event to be used to treat the statistical errors correctly in the real part
me_wgt	D	matrix element weight, the same as weight but without pdf factors
me_wgt2	D	matrix element weight, the same as weight2 but without pdf factors
x1	D	fraction of the hadron momentum carried by the first incoming parton
x2	D	fraction of the hadron momentum carried by the second incoming parton
x1p	D	second momentum fraction used in the integrated real part
x2p	D	second momentum fraction used in the integrated real part
id1	I	PDG code of the firt incoming parton
id2	I	PDG code of the second incoming parton
fac_scale	D	factorization scale used
ren_scale	D	renormalization scale used
nuwgt	I	number of additional weights
usr_wgts	D[nuwgt]	additional weights needed to change the scale

Reweighting

can reweight each event to new

- -PDF
- -factorization scale
- -renormalization scale
- -α_s (tied to the relevant PDFs)

based on weights stored in ntuple (and linking with LHAPDF)

so, for example, the events were generated with CTEQ6, and were re-weighted to CTEQ6.6

2.1 Born and real contributions

The new weight is given by

$$w = \mathbf{me\_wgt2} \cdot f(\mathbf{id1}, \mathbf{x1}, \mu_F) F(\mathbf{id2}, \mathbf{x2}, \mu_F) \frac{\alpha_s(\mu_R)^n}{(\mathbf{alphas})^n} \quad (1)$$

with μ_F the new factorization scale, μ_R the new factorization scale, f the new PDF, α_s the correponding running coupling and n the number of strong coupling (the number of jets n_j for the born contribution and $n_j + 1$ for the real contribution). If the factorization scale is not changed, one can simplify the computation (and save the pdf function call):

$$w = \mathbf{weight2} \frac{\alpha_s(\mu_R)^n}{(\mathbf{alphas})^n} \quad (2)$$

2.2 Virtual contribution

The virtal contribution is treated like the real and born contribution, but the matrix element has a dependence on the renormalization scale parametrized using the additional weights **usr_wgts**.

$$w = m \cdot f(\mathbf{id1}, \mathbf{x1}, \mu_F) F(\mathbf{id2}, \mathbf{x2}, \mu_F) \frac{\alpha_s(\mu_R)^n}{(\mathbf{alphas})^n} \quad (3)$$

$$m = \mathbf{me\_wgt2} + l\,\mathbf{usr\_wgts}[0] + \frac{l^2}{2}\mathbf{usr\_wgts}[1] \quad (4)$$

$$l = \log\left(\frac{\mu_R^2}{\mathbf{ren\_scale}^2}\right) \quad (5)$$

Reweighting, cont.

2.3 Integrated subtraction

The computation of the new weight for the integrated subtraction is the most complicated. The ROOT file has 16 additional weights to make this possible.

$$
\begin{aligned}
w &= m\frac{\alpha_s(\mu_R)^n}{(\mathbf{alphas})^n} && (6)\\
m &= \mathbf{me\_wgt2}\cdot f(\mathbf{id1},\mathbf{x1},\mu_F)f(\mathbf{id2},\mathbf{x2},\mu_F) && (7)\\
&\quad+\left(f_a^1\omega_1+f_a^2\omega_2+f_a^3\omega_3+f_a^4\omega_4\right)F_b(x_b) && (8)\\
&\quad+\left(F_b^1\omega_5+F_b^2\omega_6+F_b^3\omega_7+F_b^4\omega_8\right)f_a(x_a) && (9)\\
\omega_i &= \mathbf{usr\_wgts}[i-1]+\mathbf{usr\_wgts}[i+7]\log\left(\frac{\mu_R^2}{\mathbf{ren\_scale}^2}\right) && (10)
\end{aligned}
$$

complex:
carry both
single and double
logs

9 → documentation was wrong; ntuples allow for built-in cross-checks

where

$$
f_a^1 = \begin{cases} a=\text{quark} & : \quad f_a(x_a,\mu_F)\\ a=\text{gluon} & : \quad \sum_{\text{quarks}} f_q(x_a,\mu_F)\end{cases} \qquad (11)
$$

$$
f_a^2 = \begin{cases} a=\text{quark} & : \quad \frac{f_a(x_a/x'_a,\mu_F)}{x'_a}\\ a=\text{gluon} & : \quad \sum_{\text{quarks}} \frac{f_q(x_a/x'_a,\mu_F)}{x'_a}\end{cases} \qquad (12)
$$

$$
f_a^3 = f_g(x_a,\mu_F) \qquad (13)
$$

$$
f_a^4 = \frac{f_g(x_a/x'_a,\mu_F)}{x'_a} \qquad (14)
$$

we run into the
sum over quarks
and antiquarks
again

and $n = n_j + 1$.

PDF Errors

Better than what is done in MCFM (as far as disk space is concerned); PDF errors are generated on-the-fly through calls to LHAPDF. But then don't store information for individual eigenvectors.

```
void BlackhatAnalysis::GetPdfErrors(const std::vector<Double_t> x,
                                    const Double_t f_c,
                                    const std::vector<int> flav,
                                    Double_t Q,
                                    bool shiftUp,
                                    Double_t &delta)
{
    Double_t f_p,f_m;
    // Loop over all eigenvectors
    for(int e=1;e<=m_nEigen;e++)
    {
        LHAPDF::initPDF(2, 2*e-1); // init positive shift pdf
        LHAPDF::initPDF(3, 2*e); // init negative shift pdf
        //std::cout << "Eigenvector " << e << std::endl;
        f_p = LHAPDF::xfx(2,x[0],Q,flav[0])/x[0]*LHAPDF::xfx(2,x[1],Q,flav[1])/x[1];
        f_m = LHAPDF::xfx(3,x[0],Q,flav[0])/x[0]*LHAPDF::xfx(3,x[1],Q,flav[1])/x[1];
        if(shiftUp) // if positive pdf shift
            delta += pow(std::max(std::max(f_p-f_c,f_m-f_c),0.0),2);
        else // if negative pdf shift
            delta += pow(std::max(std::max(f_c-f_p,f_c-f_m),0.0),2);
    }
    delta = sqrt(delta);
    if(!shiftUp) delta *= -1.0;
    //std::cout << "Total delta: " << delta << std::endl;
}
```

$$\Delta X^+_{\max} = \sqrt{\sum_{i=1}^{N}[\max(X_i^+ - X_0, X_i^- - X_0, 0)]^2},$$

$$\Delta X^-_{\max} = \sqrt{\sum_{i=1}^{N}[\max(X_0 - X_i^+, X_0 - X_i^-, 0)]^2}.$$

Logistics

- So total file disk space is quite large, multi-TB (and there are many events to be processed)
  - I bought a 20TB disk specifically for this purpose
- But they're divided into few GB files (Blackhat +SJ)
- So we can make our analysis parallel using 350 nodes at MSU
- Possible to run through W + 4 jet NLO analysis in ~few hours (much faster without the scale variations)

…so for example

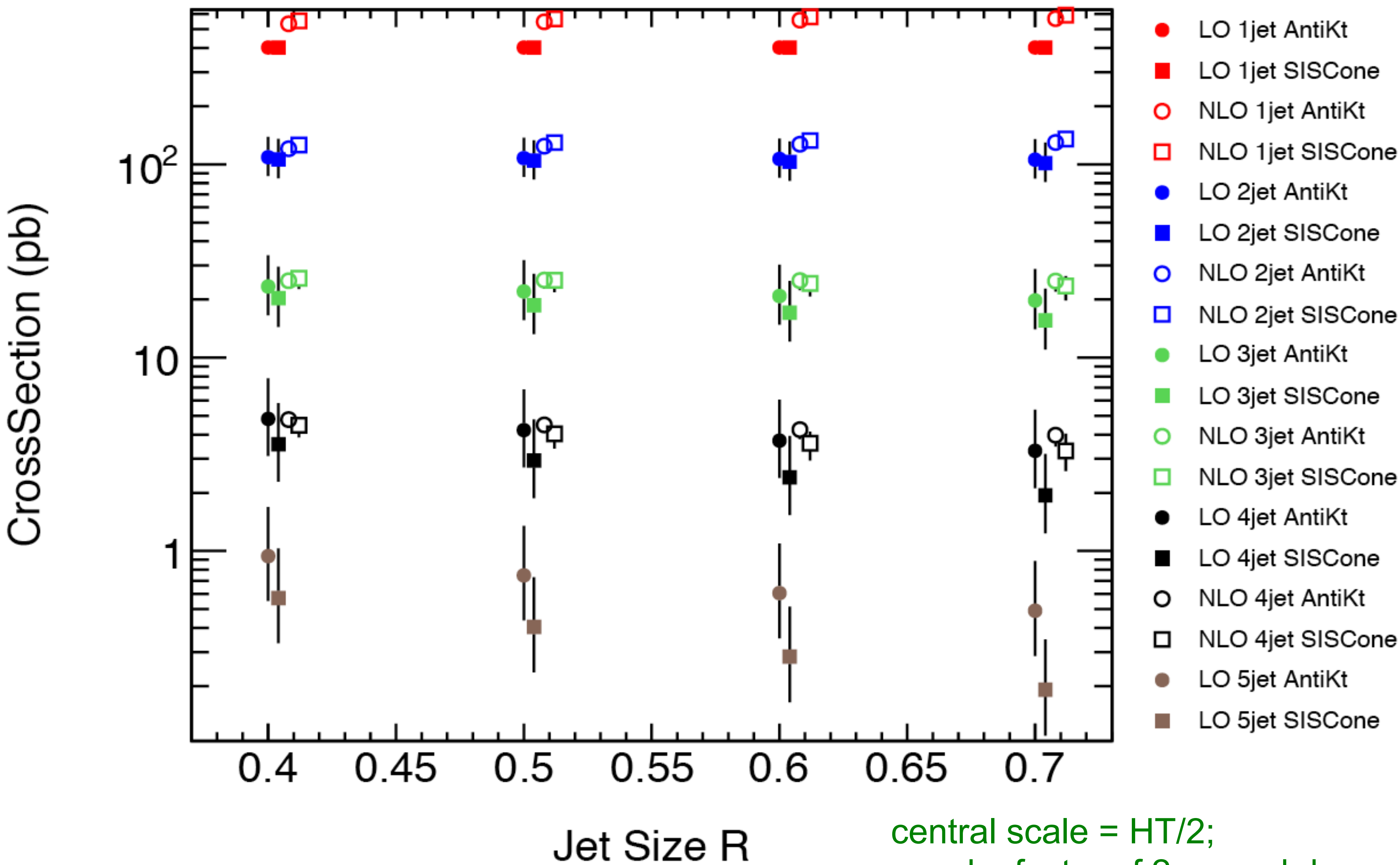
- W^+ + 3 jets at 7 TeV for standard cuts (plus for electron cuts)
  - $|y_\mu|<2.4$
  - $p_T^\mu>20$ GeV/c
  - $p_T^\nu>$ 25 GeV/c
  - $P_T^{jet}>20$ GeV/c
  - $|y_{jet}|<2.8$
  - $m_T(\mu,\nu)>40$ GeV
- New cuts or histograms means re-running through the ntuples

- For antikT4
  - born: 22.69 pb
  - loop-lc: -0.69 pb
  - loop-fmlc: 0.39 pb
  - vsub: 27.16 pb
  - real: -17.34 pb
  - <u>Total: 32.21 pb</u>

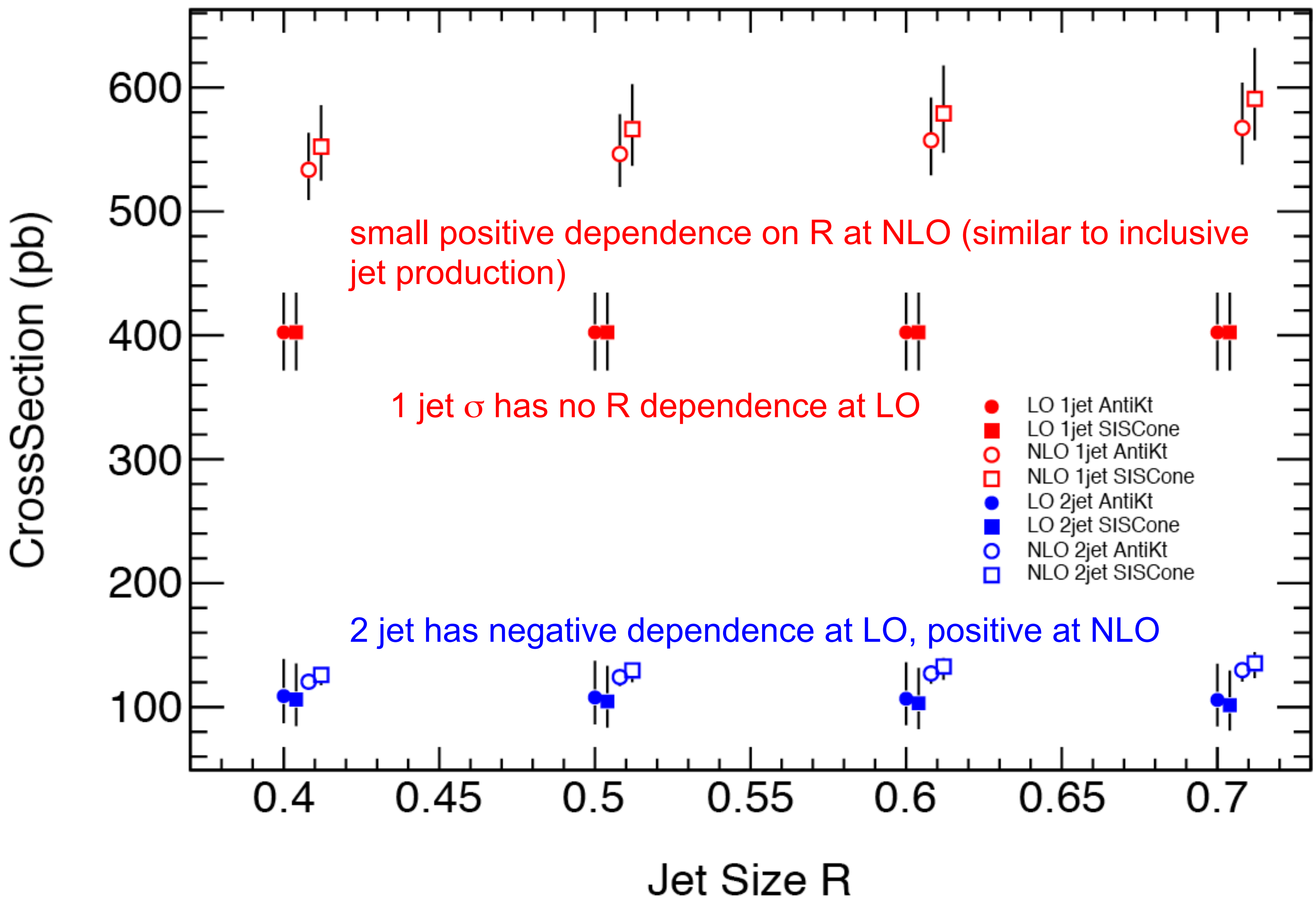
Running

- From Blackhat+Sherpa, we have NLO ntuples (in same format) for W + 1,2, 3, 4 and 5 (LO only) jets
- Makes it easy to make plots for different jet multiplicities and/or combined jet multiplicities, by just running Python scripts over processed ntuples

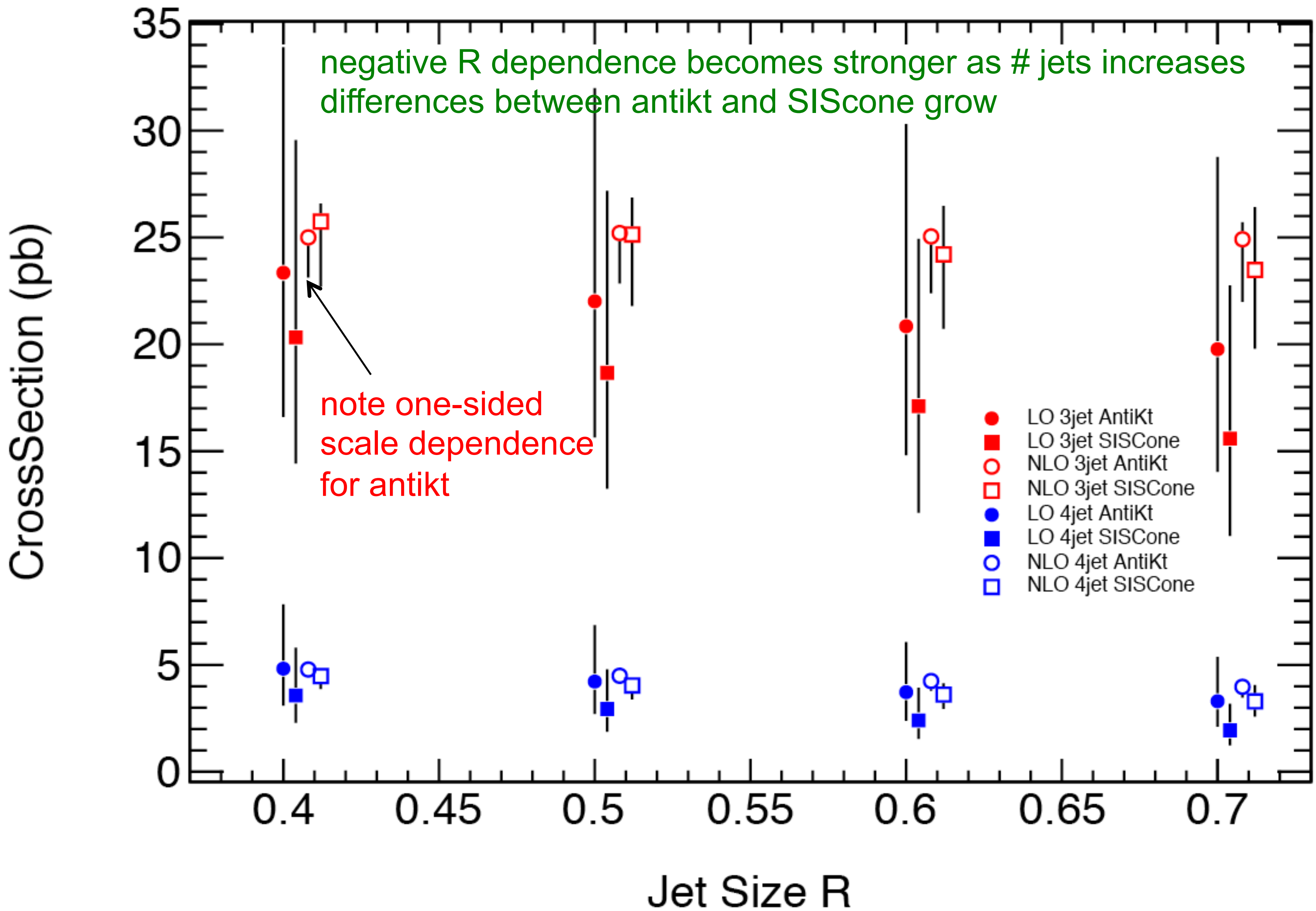
Look at jet size, algorithm dependences; scale uncertainty



central scale = HT/2;
vary by factor of 2 up and down



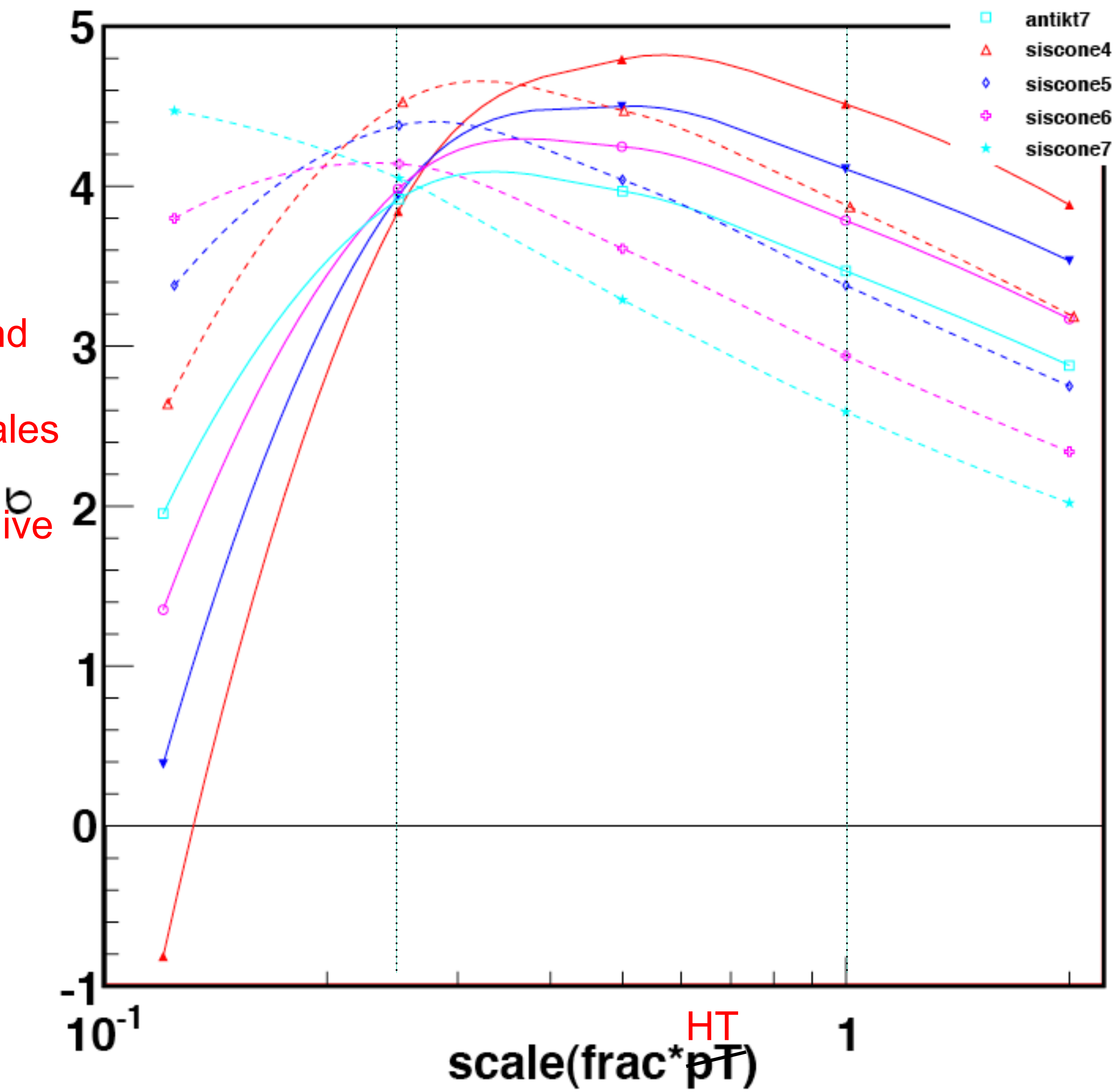
CrossSection (pb)
600
500
400
300
200
100
0.4
0.45
0.5
0.55
0.6
0.65
0.7
Jet Size R
small positive dependence on R at NLO (similar to inclusive jet production)
1 jet σ has no R dependence at LO
2 jet has negative dependence at LO, positive at NLO
LO 1jet AntiKt
LO 1jet SISCone
NLO 1jet AntiKt
NLO 1jet SISCone
LO 2jet AntiKt
LO 2jet SISCone
NLO 2jet AntiKt
NLO 2jet SISCone

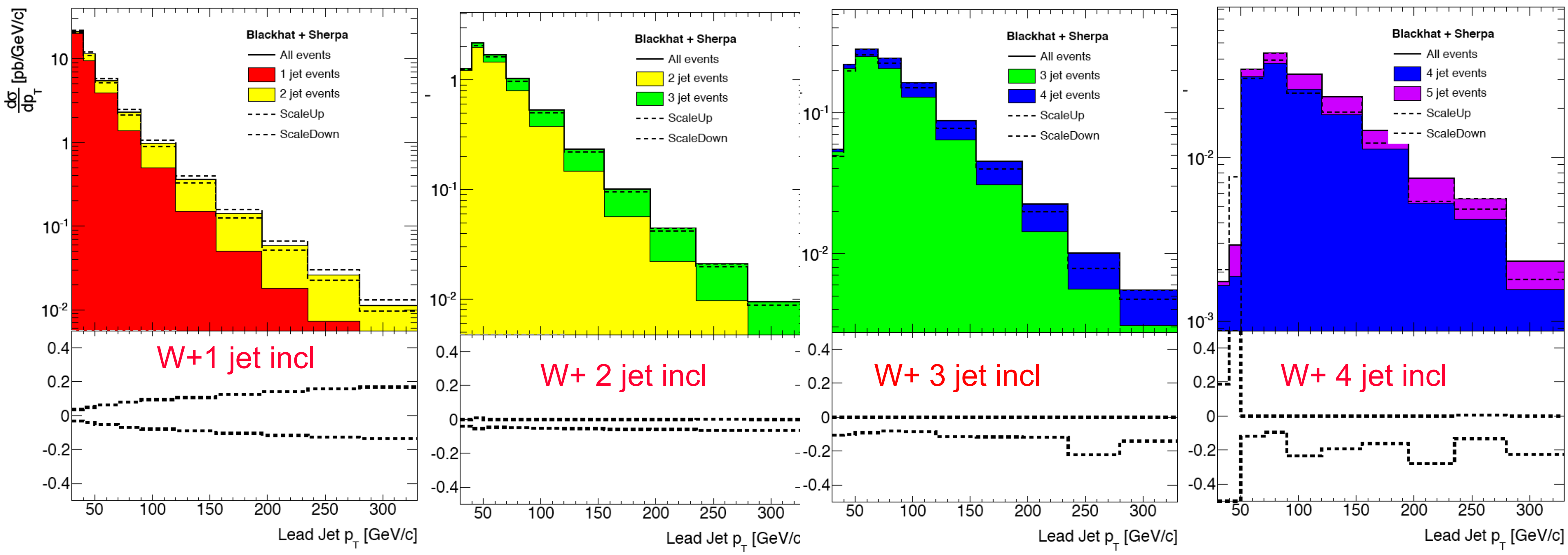


negative R dependence becomes stronger as # jets increases
differences between antikt and SIScone grow
note one-sided
scale dependence
for antikt
LO 3jet AntiKt
LO 3jet SISCone
NLO 3jet AntiKt
NLO 3jet SISCone
LO 4jet AntiKt
LO 4jet SISCone
NLO 4jet AntiKt
NLO 4jet SISCone
CrossSection (pb)
Jet Size R
0
5
10
15
20
25
30
35
0.4
0.45
0.5
0.55
0.6
0.65
0.7

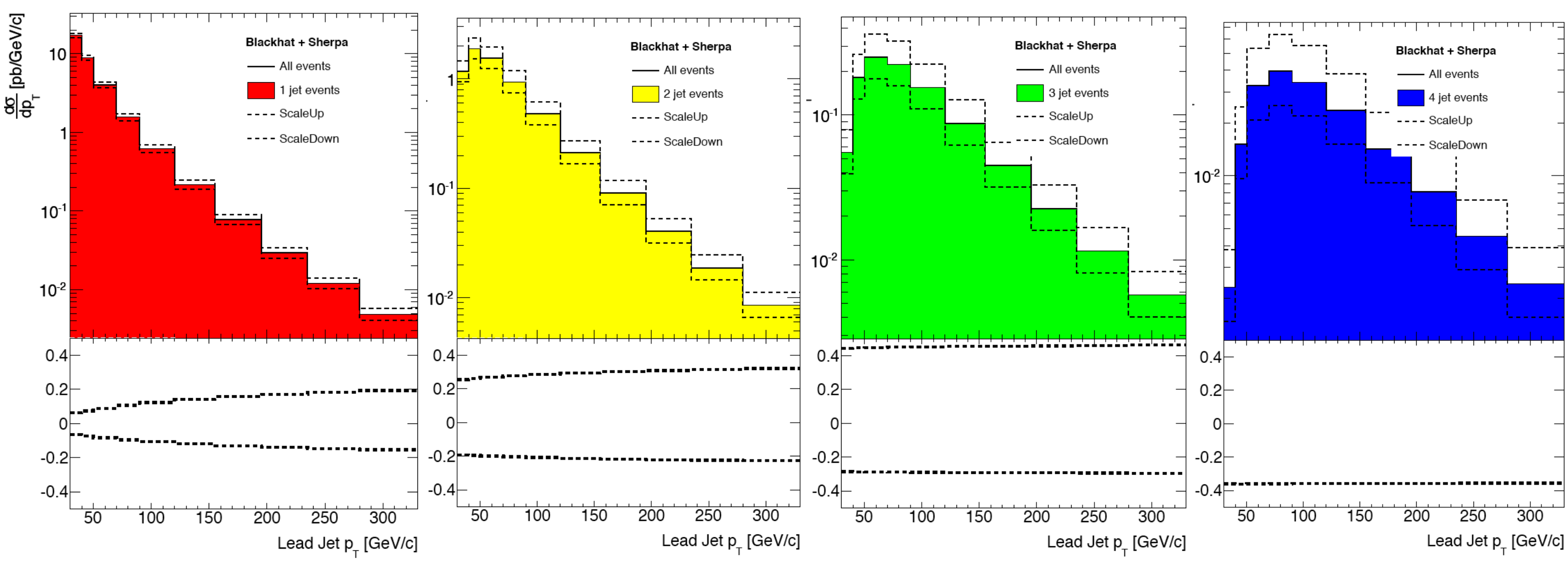
W+4 jets scale dependence

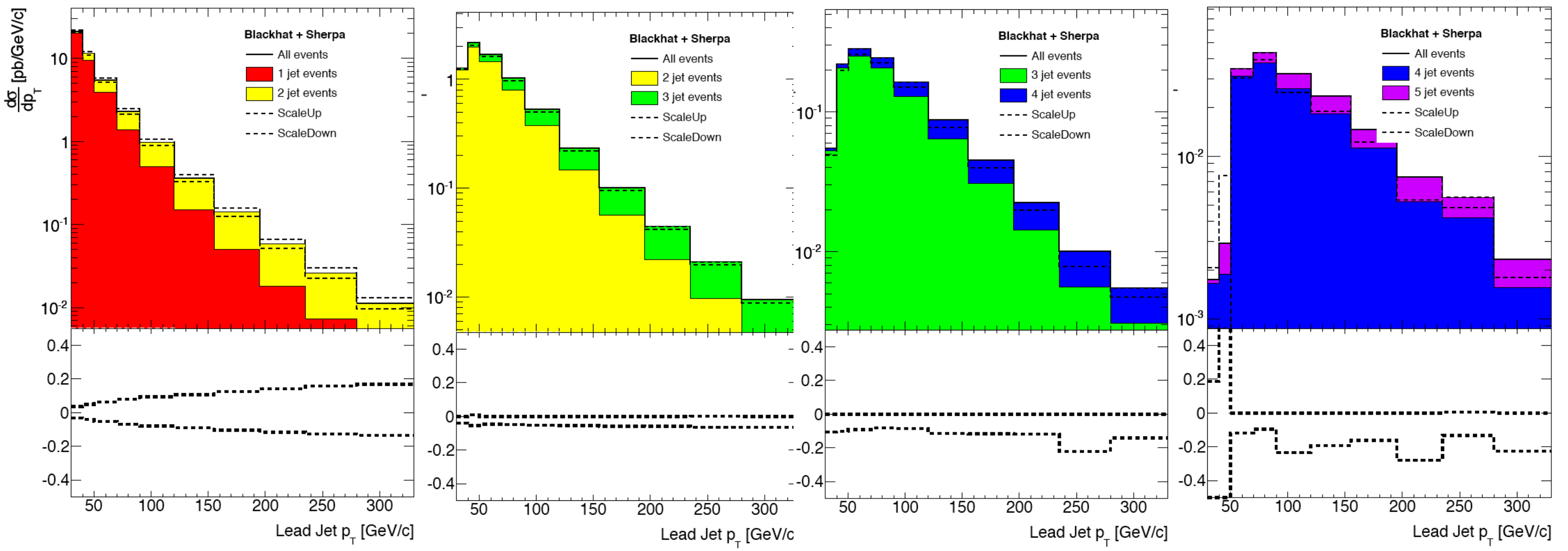
- A scale of HT/2 is ~ the peak for antikt4; so all deviations are negative
- Siscone peaks around HT/3
- Moves to smaller scales for larger R
- @HT/4, all antikt R give same result; that scale seems to be around HT/5 for siscone



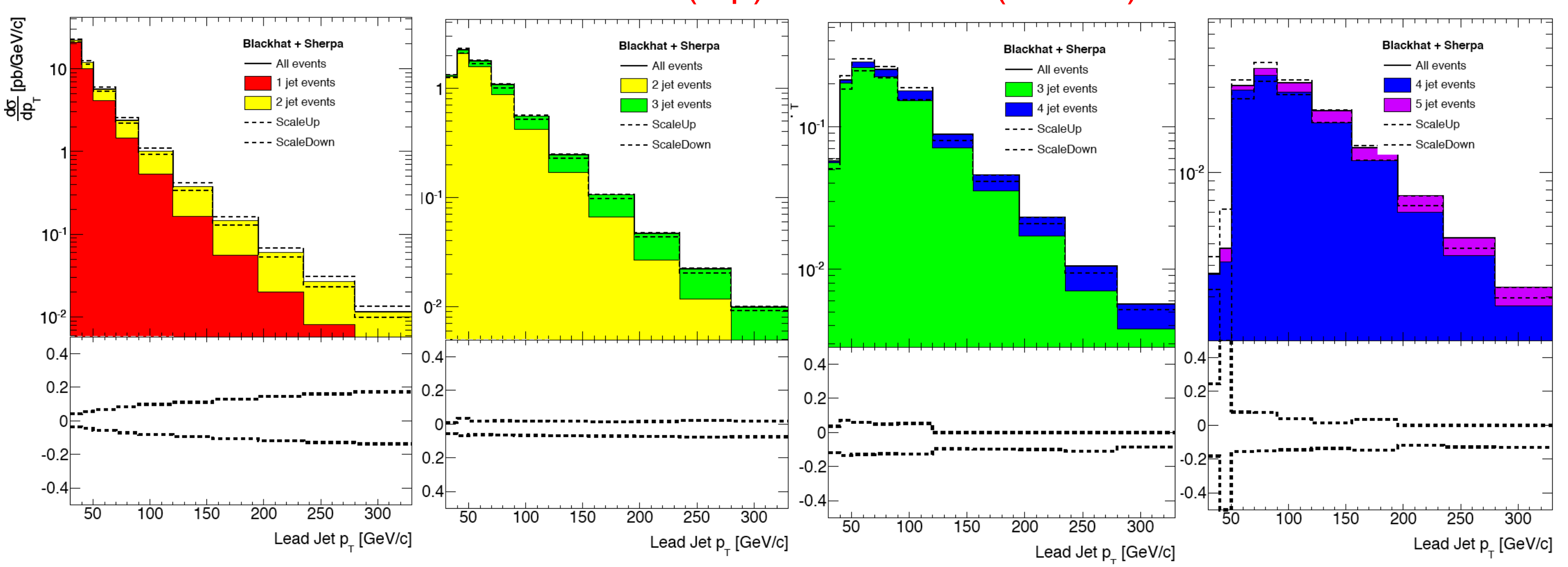


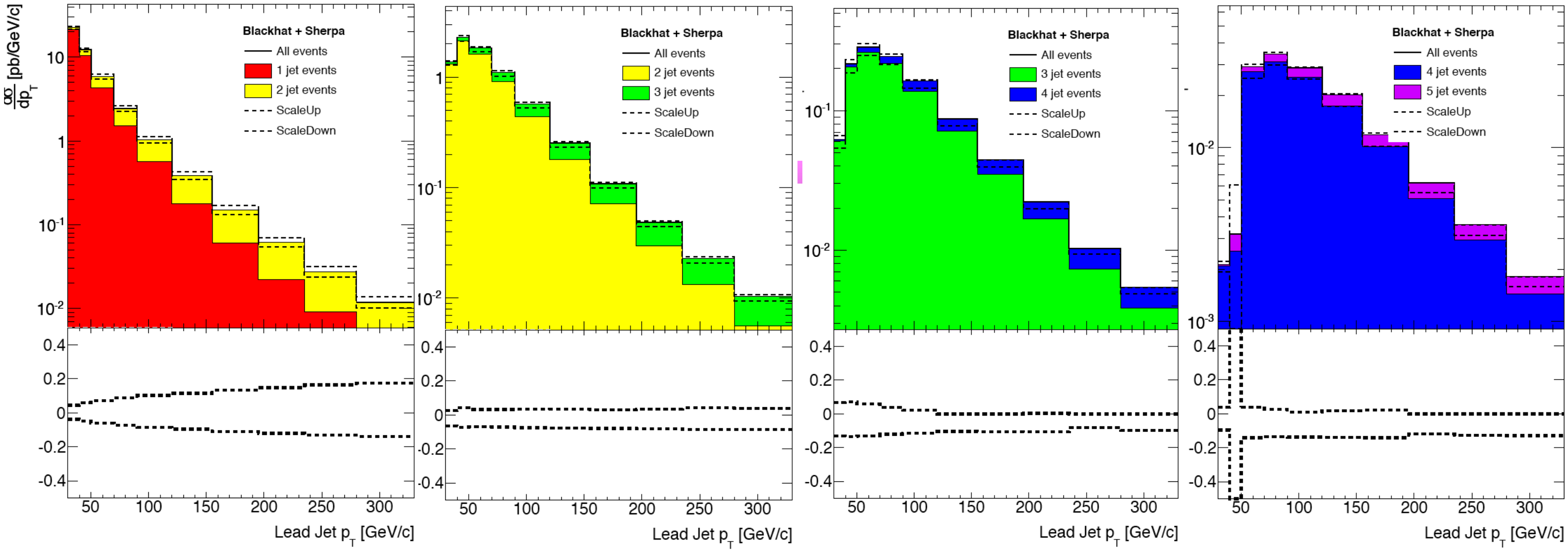
antikT4: NLO (top) and LO (bottom)



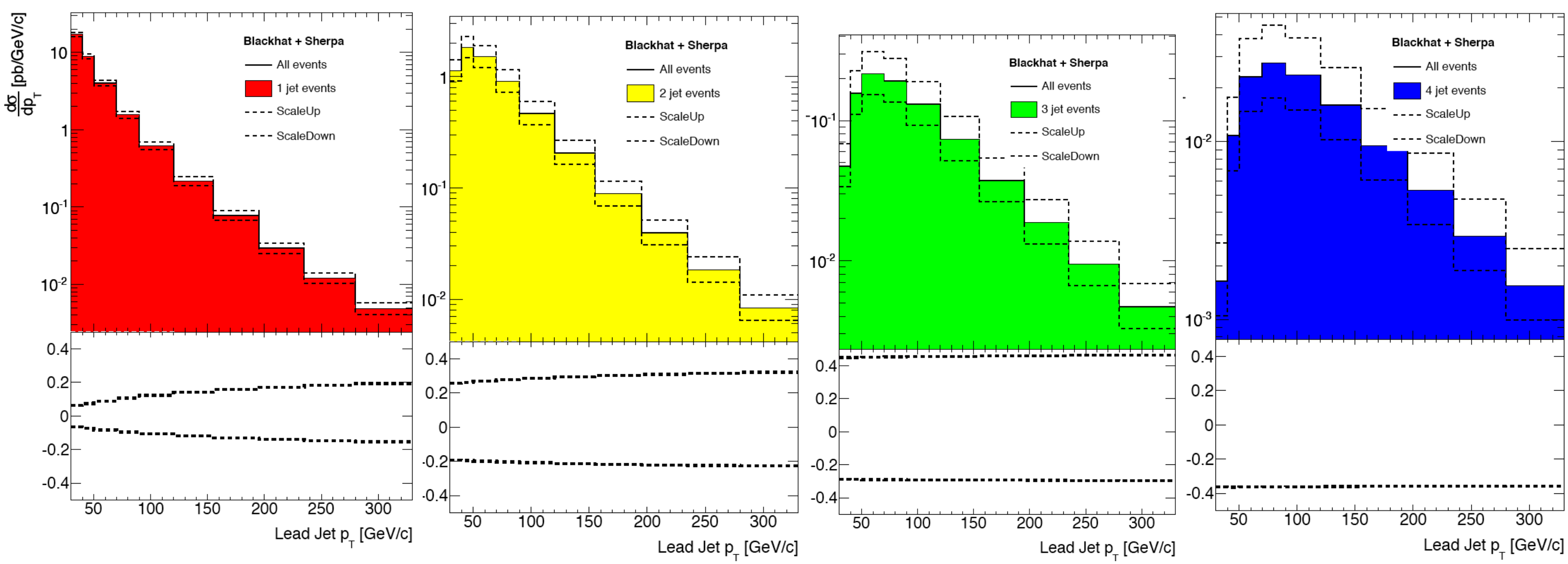


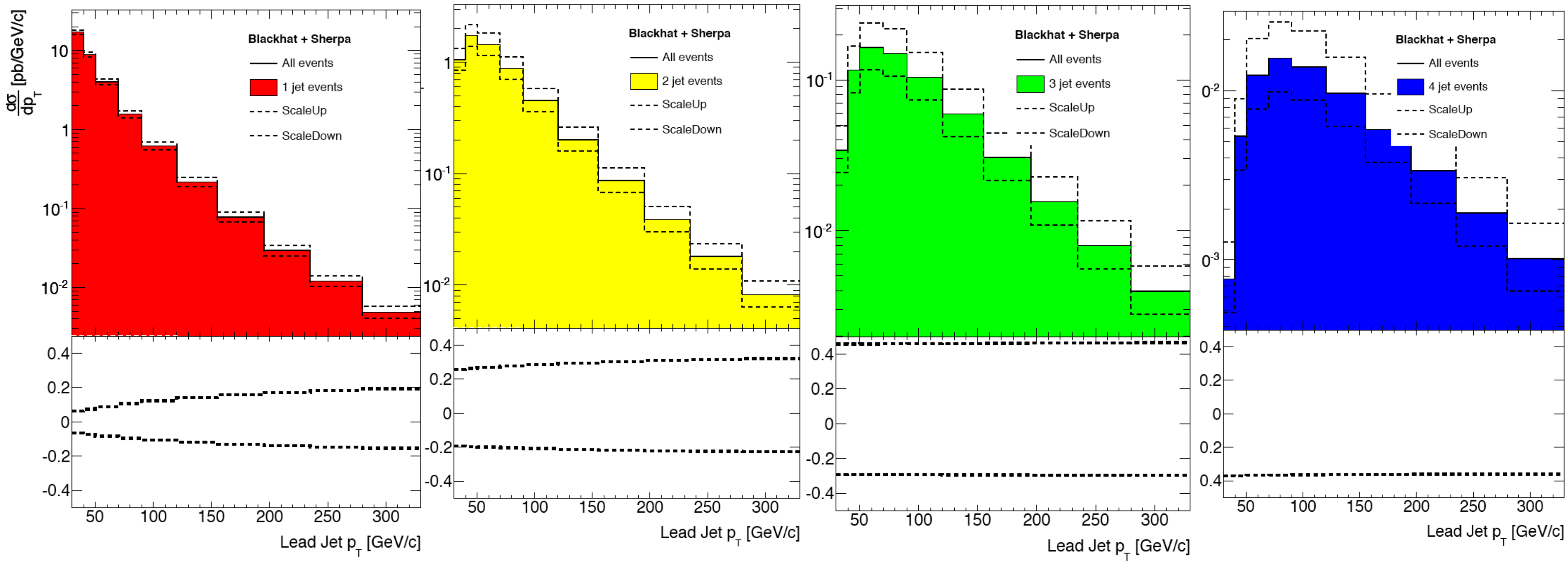
NLO: antikt4 (top) and siscone (bottom)



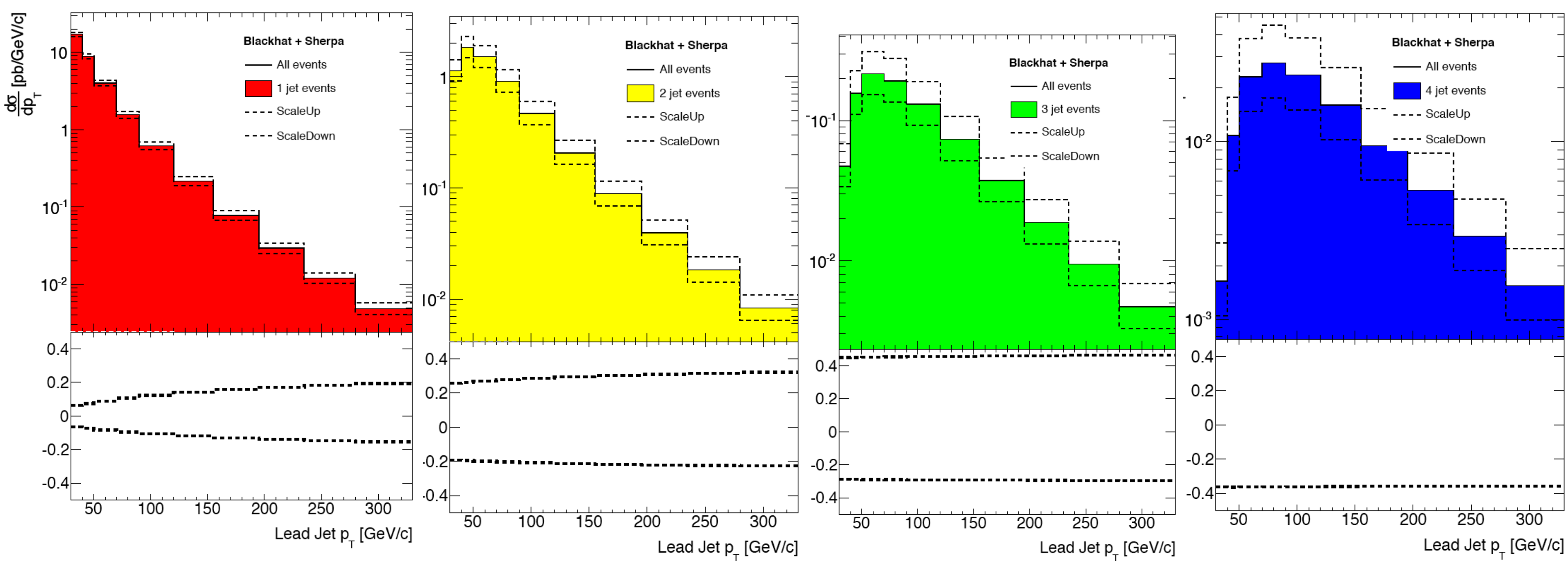


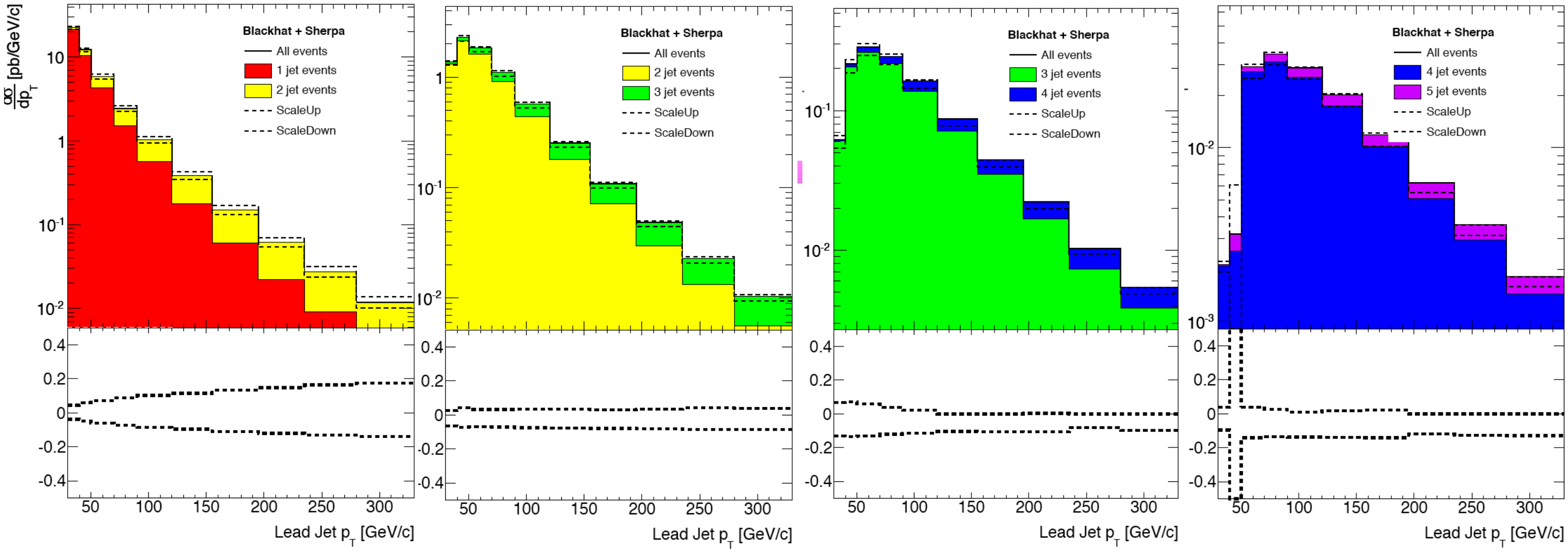
antikt7: NLO (top) and LO (bottom)



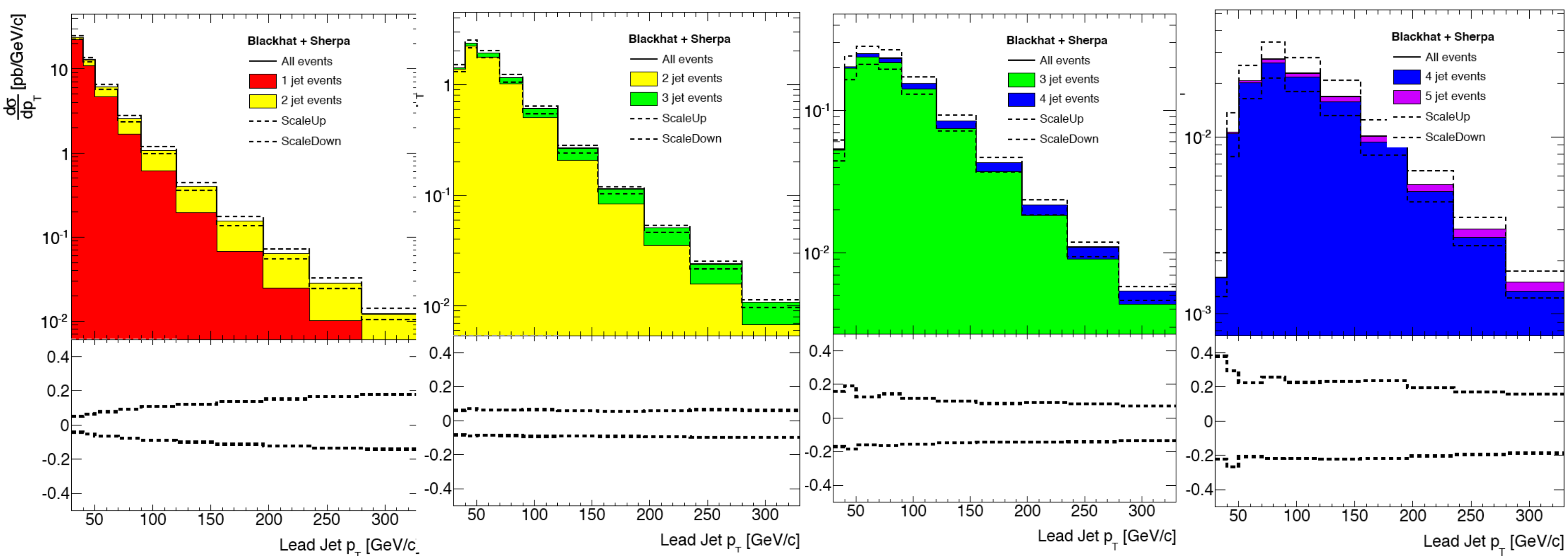


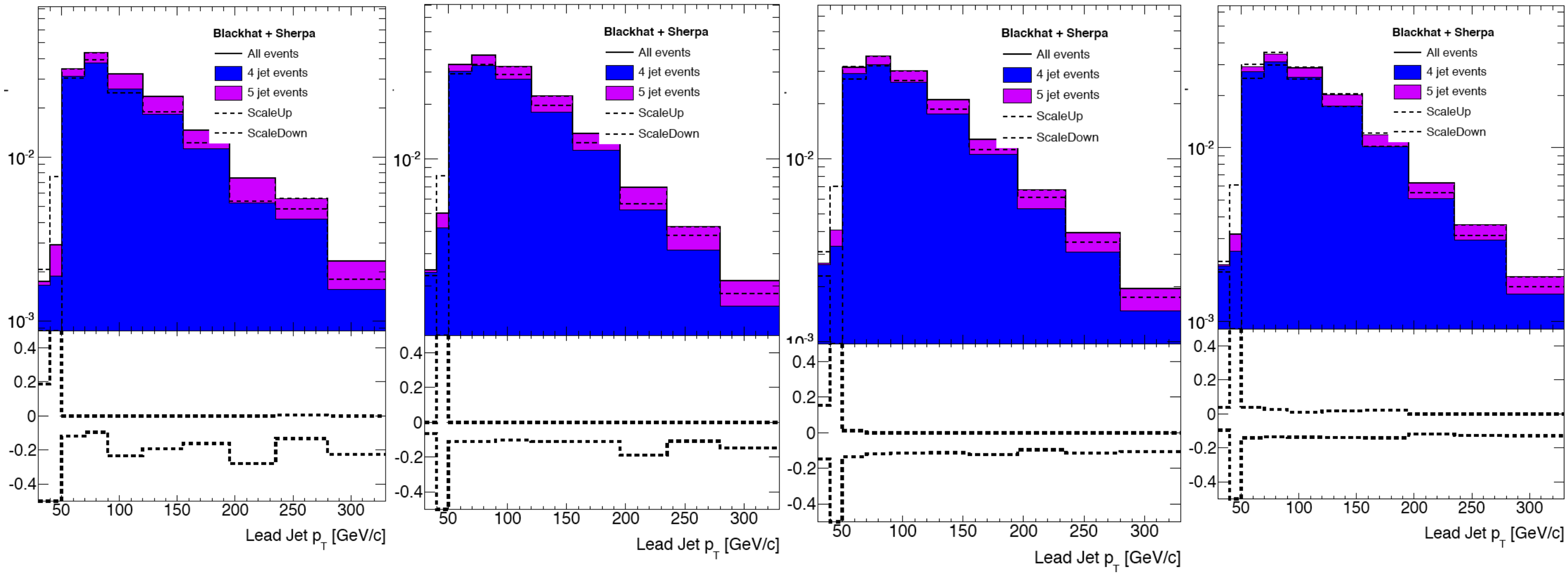
LO: siscone7 (top) and antikt7 (bottom)



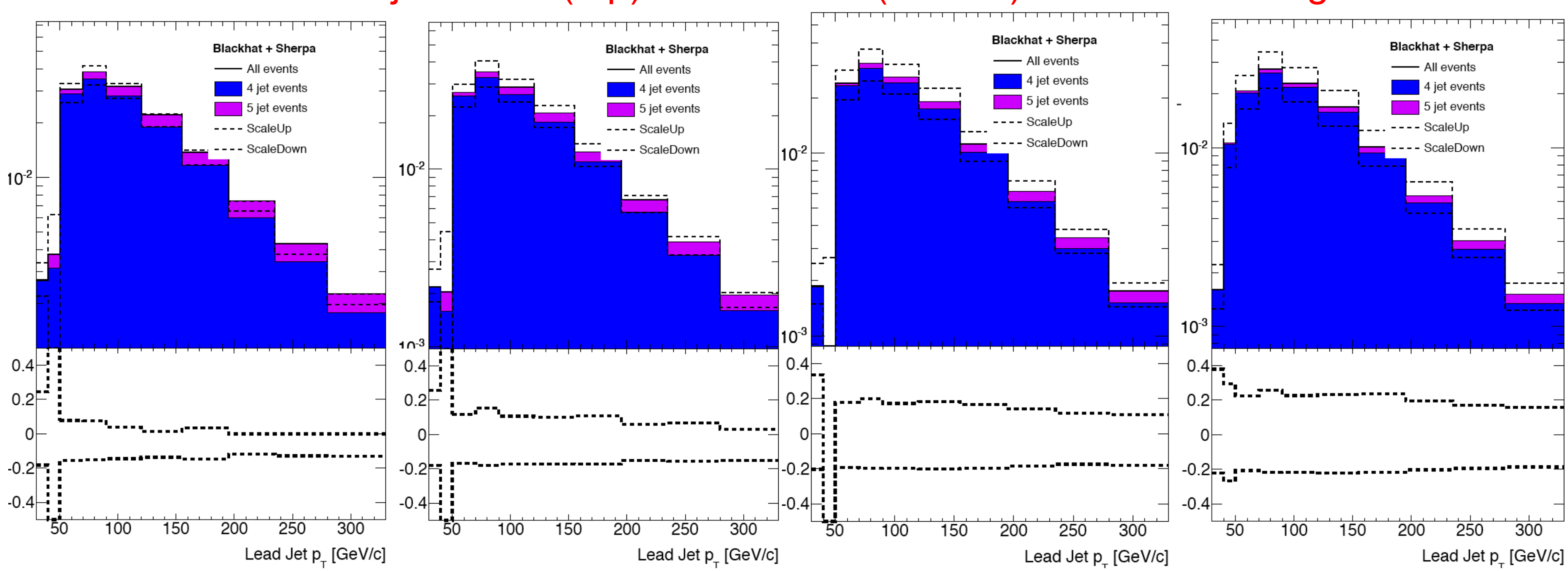


NLO: antitkt7 (top) and siscone7 (bottom)



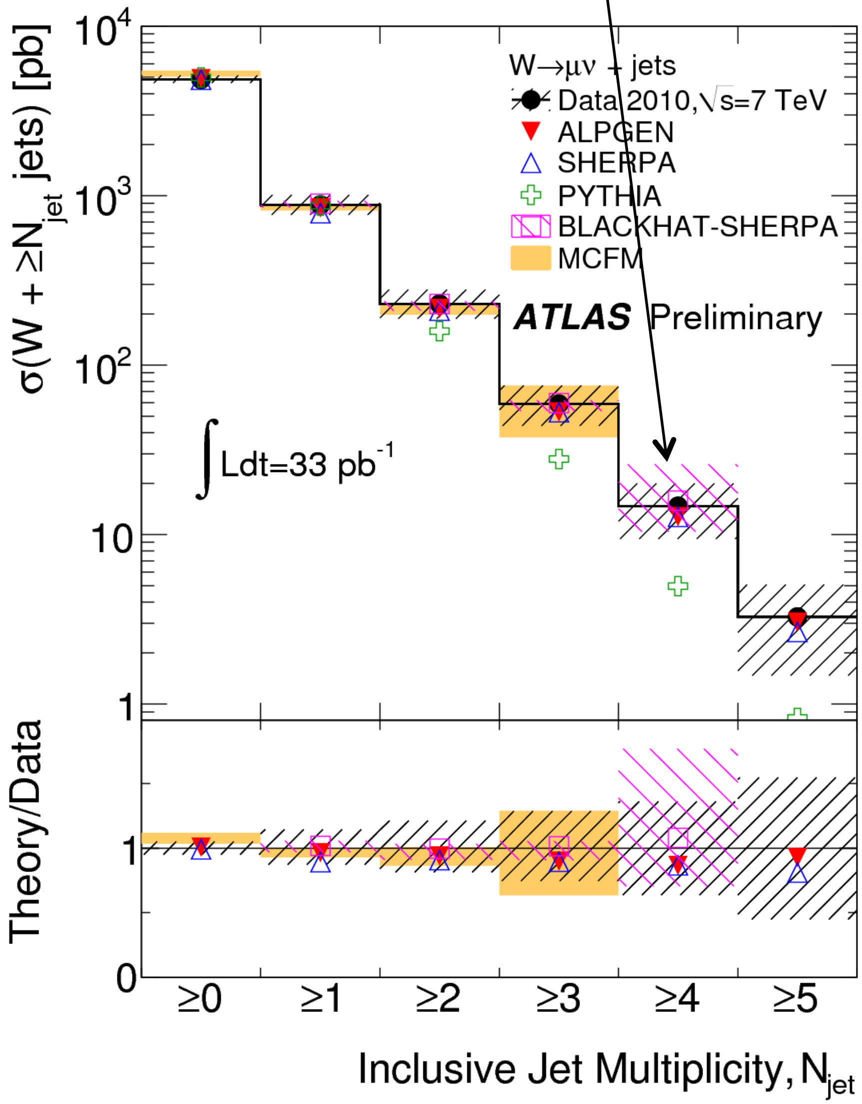


NLO 4 jet: antikt (top) and siscone (bottom): 0.4-0.7 left to right



LHC: W/Z+jets

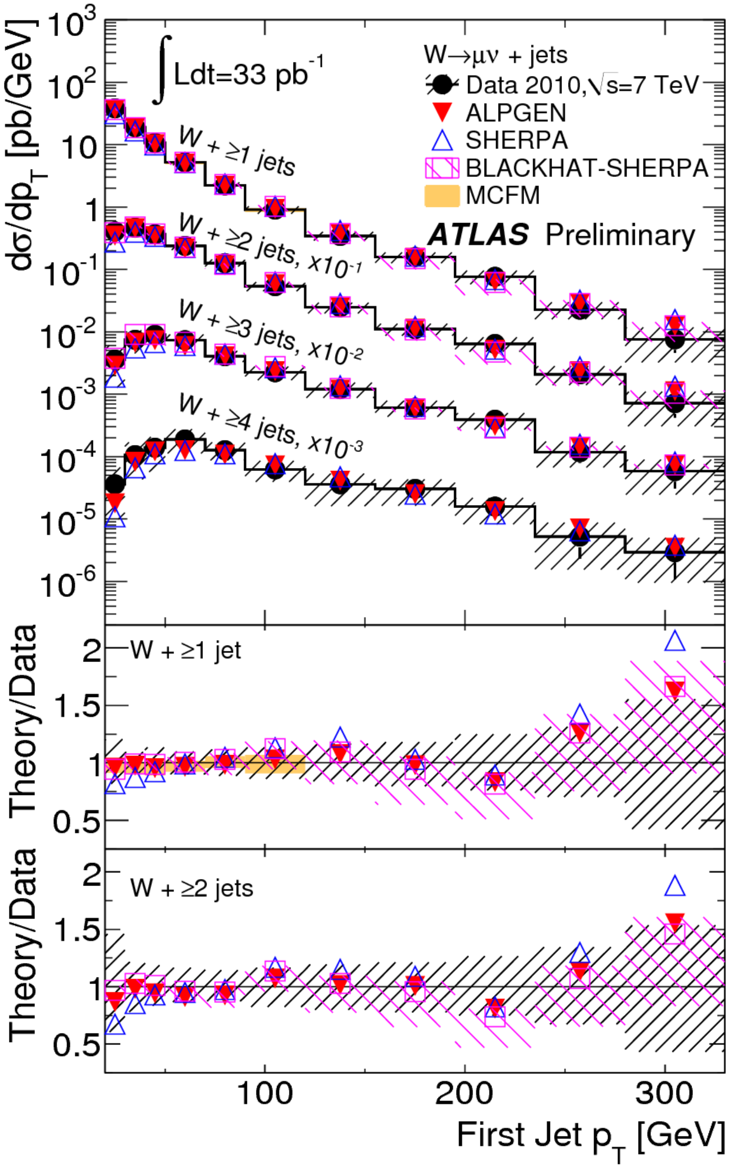
BH+S is LO only for 4th jet; NLO comparisons not yet public



Note the use of Blackhat+Sherpa for predictions for 3 and 4 jet final states. First use of B+S by experimentalists.

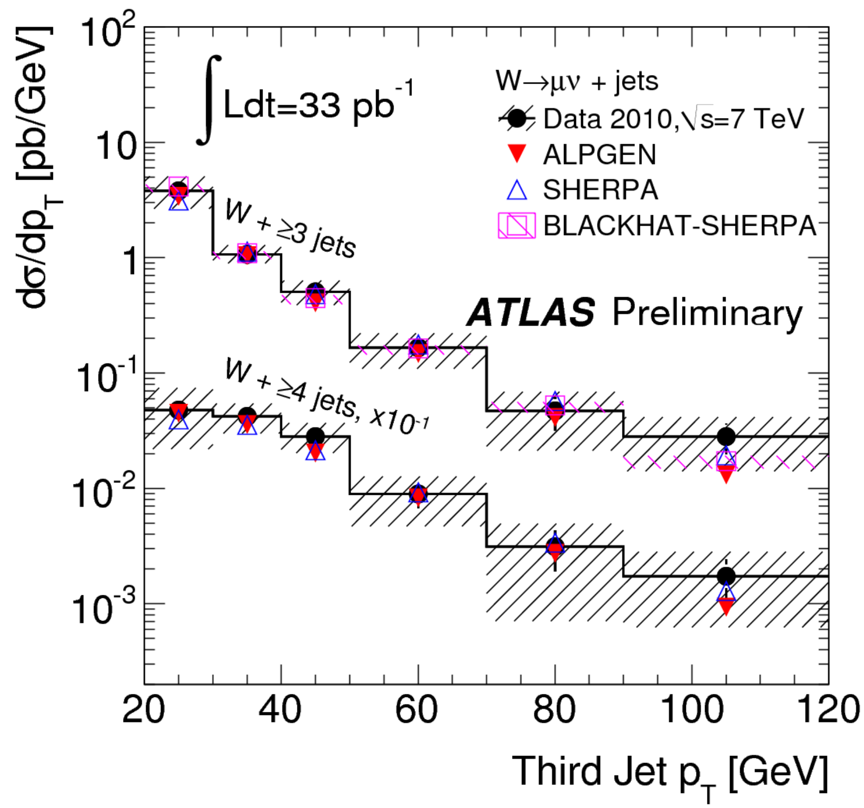
More (better) predictions available in W+jets paper

ATLAS: W+jets



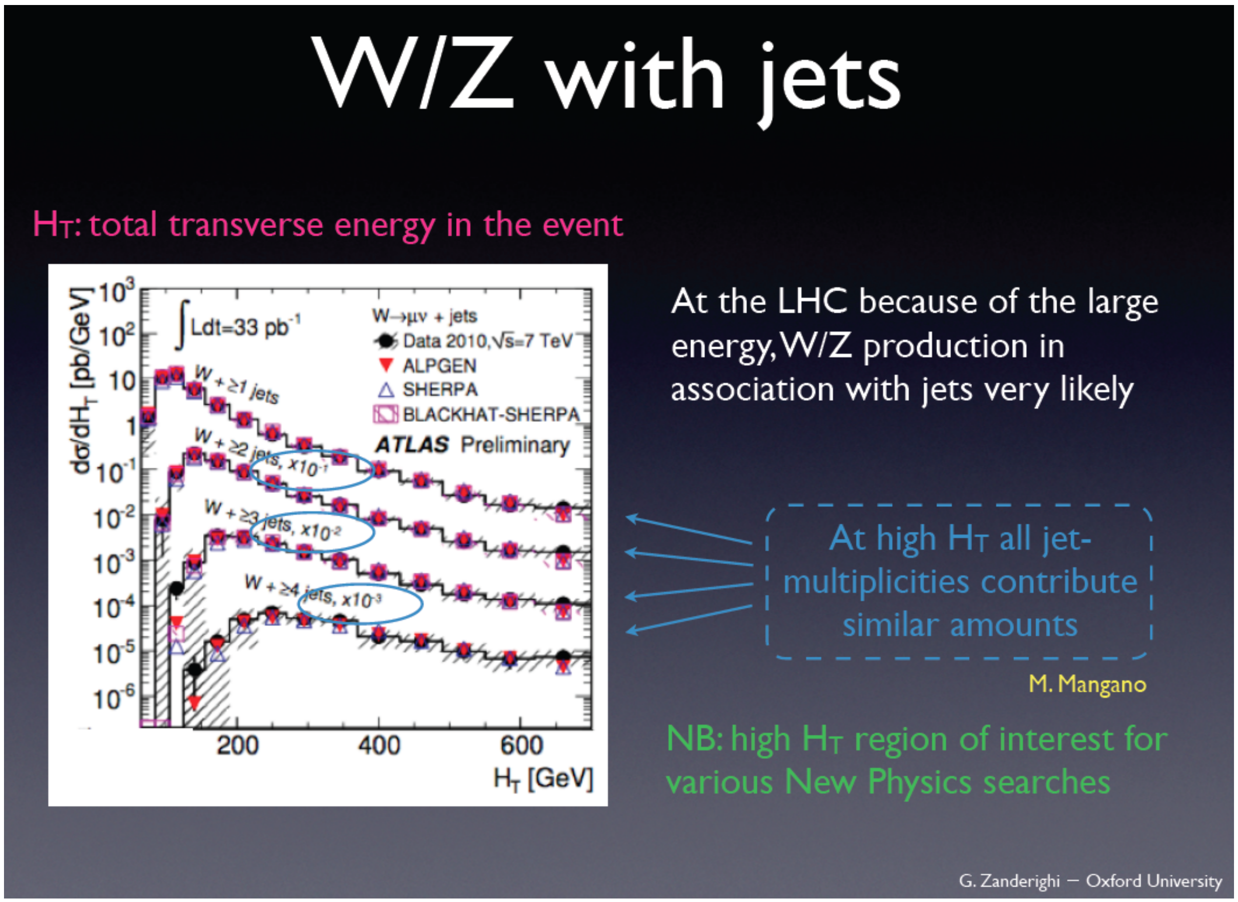
Good agreement observed between the predictions and the data over full kinematic range. Again, NLO BH+S comparisons to W+4 jets soon to be public.

Note that UE/hadronization corrections not trivial, i.e. they don't fall as $1/p_T$, due to to steepness of n jet cross section and multiplicative factor of n jets.

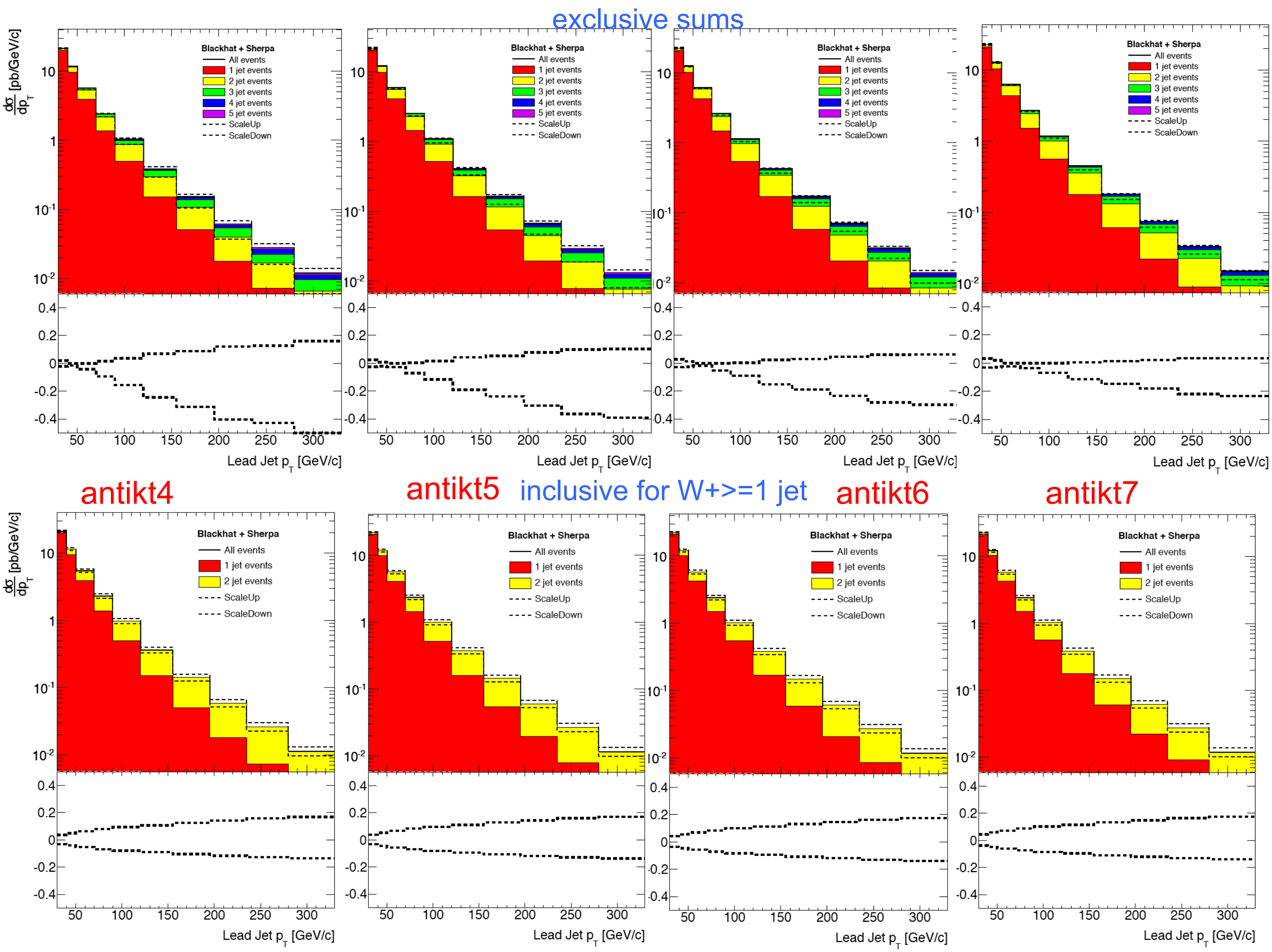


Exclusive sums

- …from Giulia's talk at LP11
- What about using NLO information from each jet multiplicity using an exclusive sum (cut on pTjet>30 GeV/c)
- No double-counting; but Sudakov FF's only from NLO ME



…not just for high H_T
for high p_T of lead jet (in data), there will be a large jet multiplicity for the event



exclusive sums
Blackhat + Sherpa
All events
1 jet events
2 jet events
3 jet events
4 jet events
5 jet events
ScaleUp
ScaleDown
$\frac{d\sigma}{dp_T}$ [pb/GeV/c]
Lead Jet p_T [GeV/c]
antikt4
antikt5
inclusive for W+>=1 jet
antikt6
antikt7

There's a lot of physics that can be investigated using the abundant NLO calculations now available.