

Scale Dependence and Scale Uncertainties due to Missing Higher Orders in NLO Jet Calculations

Summer Student Programme 2009, DESY

Henrike Fleischhack

Georg-August Universität Göttingen

Supervisors: Dr. G. Grindhammer, R. Kogler

H1 Group, MPI Munich

1st October 2009

Abstract

This report gives a summary of the work I did during the summer student programme 2009 and a subsequent three-week stay at DESY. I worked with the MPI Munich group of the H1 collaboration, my supervisors were Dr. G. Grindhammer and R. Kogler.

I examined jet cross sections in proton-electron collisions in next-to-leading order, with emphasis on scale dependence and the resulting uncertainties. I developed a method to of writing out information about events generated using the program `nlojet++`, and analyzed events that were obtained this way.

Contents

1	Introduction	1
1.1	Experimental Setup	1
1.2	DIS Process and Kinematics	1
1.3	Multijet Production	1
1.4	Cross Section Definitions	2
2	Cross Section Calculations	3
2.1	NLO Calculations	3
2.2	Monte Carlo Integration	4
3	The Program <code>nlojet++</code>	4
3.1	Some Remarks on Nomenclature	5
3.2	Usage	5
3.3	The User Routine	6
3.4	Writing out Events	6
3.5	Changes Made to the <code>nlojet++</code> Source Code	7
3.6	Scale Dependence of the Weights	8
4	Setup	9
4.1	Event Selection and Phase Space	9
4.2	Jet Algorithms	9
4.3	Parton Distribution Functions and Strong Coupling Constant	10
4.4	Factorization and Renormalization Scale, Theory Uncertainty	10
4.5	Statistical Uncertainties	10
4.6	Event Generation and Analysis	10
5	Results	11
5.1	Choice of μ_r	11
5.2	Comparison of Jet Algorithms	12
5.3	Pseudorapidity Range	12
5.4	Parton Momentum Fraction	12
5.5	Performance of the Analysis	13
6	Summay and Outlook	13
A	Plots	15

1 Introduction

1.1 Experimental Setup

The HERA collider at DESY was operational from 1991 to 2007. It accelerated electrons¹ and protons, which were then used for four experiments. Two of them, H1 and Zeus, recorded collisions between both beams. The scattered electrons as well as the scattered proton (for elastic processes) or the hadronic final state (for inelastic processes) were observed, their energies and momenta were measured. These measurements can be used for a variety of physics goals, e.g. to probe the structure of the proton and for precision measurements of the strong coupling α_s .

1.2 DIS Process and Kinematics

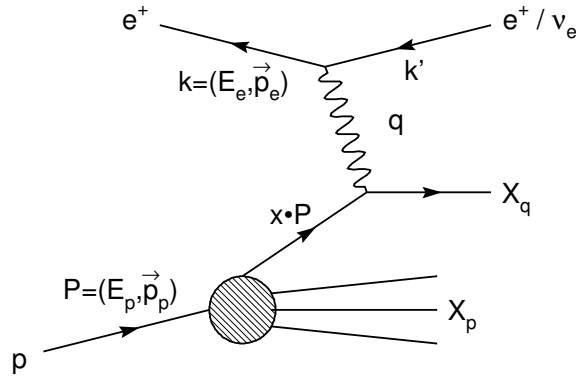


Figure 1: Leading Order Feynman Graph of Deep Inelastic Scattering

In electron-proton-scattering, the electron interacts with the proton by emitting a vector boson. This can be a photon or Z boson (neutral current interactions), or it can be a W boson (charged current interactions), in which case the electron turns into a neutrino. Because of the mass of the weak bosons, W and Z exchange are suppressed at low momentum transfers. In my studies, only neutral current processes were considered.

In this report, P , p_e , and p'_e will stand for the 4-momentum of the beam proton, the beam electron, and the scattered electron, respectively. p_i will denote the 4-momentum of the i -th jet in the final state, ordered by p_T . Other important kinematic quantities can be found in table 1. Observed events can be categorized according to the hadronic final state. In elastic processes, $W = M_P$, i.e. the proton does not break up. In inelastic processes, the proton breaks up. If the scale is hard enough (e.g. $Q^2 > 4 \text{ GeV}^2$), one talks about deep inelastic scattering (DIS). It can be described as the photon interacting with one parton inside the proton. The dominant Feynman diagram for this process can be seen in 1. The proton remnant disappears into the beam pipe, the scattered quark makes a jet which can be observed in the detector.

1.3 Multijet Production

The process shown in fig. 1 does not contain an explicit strong vertex (all strong processes have been “absorbed” into the proton). To measure α_s , one therefore has to use events with more

¹In the context of this report, “electron” will refer to both electrons and positrons.

	formula	meaning
s	$(P + p_e)^2$	available energy in the center-of-mass system
q	$p'_e - p_e$	momentum transfer between electron and proton
Q^2	$-q^2$	virtuality
x	$\frac{Q^2}{2 \cdot p \cdot q}$	Bjorken scaling variable
y	$\frac{Q^2}{s \cdot x}$	inelasticity
W	$\sqrt{\left(\sum_i p_i\right)^2}$	mass of the hadronic final state
M_{12}	$\sqrt{(p_1 + p_2)^2}$	mass of the first two jets (in dijet events)
ξ	$x \cdot \left(1 + \frac{M_{12}^2}{Q^2}\right)$	incoming parton's momentum fraction (LO)

Table 1: Kinematic variables in the DIS process

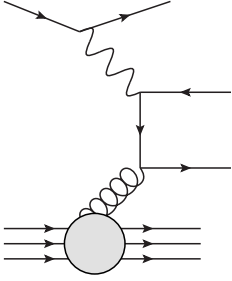


Figure 2: Boson-gluon fusion process

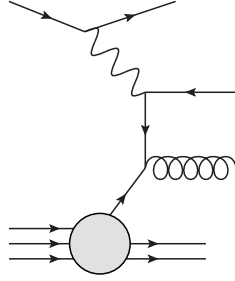


Figure 3: QCD Compton process

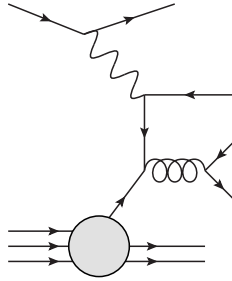


Figure 4: Real NLO contribution

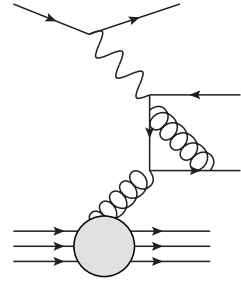


Figure 5: Virtual NLO contribution

than one jet. The dominant Feynman diagrams for dijet production in DIS can be seen in fig. 3 and 2. They are referred to as QCD compton process and boson-gluon fusion.

1.4 Cross Section Definitions

There are two different ways of defining jet cross sections. For inclusive jet cross sections, one counts jets that have certain criteria, independent of the properties of other jets in the event. For the so-called dijet (trijet) cross sections, one counts events with at least two (three) jets with certain criteria.

To get rid of QPM processes (cf. fig. 1), but not dijet events where one jet was not fully reconstructed, jet finding and analysis is often performed in the Breit frame. It is defined as the frame of reference where $q + 2 \cdot x \cdot P = 0$ and the scattered electron has $\varphi = 0$. In this frame, the outgoing parton in fig. 1 has $p_T = 0$. Partons can only have $p_T > 0$ if there is more than one outgoing parton. Cutting on jet p_T in the Breit frame can therefore be used to identify multijet events.

Differential cross sections can be defined approximately by counting events in certain bins. Making the bin size infinitesimally small would give the differential cross section. Because of finite statistics and detector resolution, finite bin sizes are usually used.

2 Cross Section Calculations

The cross section of a given process can be calculated by evaluating the relevant Feynman diagrams. Processes with quarks are generally difficult to calculate as quarks only propagate freely at very high energies/short distances. They can only be observed in (colorless) bound states, the hadrons. The processes inside hadrons can not be calculated perturbatively, Feynman calculus can not be used. Typically, one divides the process into a “hard” (perturbatively calculable) and a “soft” (non-perturbative) part. The nonperturbative part can often be parameterized by a function that can be determined from other measurements.

In the case of DIS, the “hard” process (e.g. $eg \rightarrow eq\bar{q}$) can be calculated perturbatively. The result has to be folded with the PDF, which describes the momentum distribution inside the proton. One can then choose to simulate parton showers, where the outgoing partons radiate soft/collinear gluons perturbatively, and hadronisation, which relies on non-perturbative models. In the studies described here, neither parton showers nor hadronisation were taken into account. One can now write a DIS cross section in the form

$$\sigma = \sum_{m=1}^{\infty} \sum_a \int_0^1 dx f_a(x, \mu_f^2) \cdot \int d\Gamma^{(m)}(\{p\}_m, x) |\mathcal{M}(\{p, a\}_m)|^2 \cdot F_J(\{p\}_m), \quad (1)$$

summing over the number m of partons in the final state as well as a generalized “flavor” a , which describes the kind of parton (gluon, up- or down-type quark). $\{p\}_m$ is the set of the momenta of the m particles in the final state, $d\Gamma^{(m)}$ the phase space for m (massless) particles. f_a is the parton distribution function, \mathcal{M} the matrix element of the hard subprocess (which in itself is a perturbation series in α_s), and F_J the jet function.

The factorization or separation into soft and hard subprocesses introduces an arbitrary scale μ_f that divides the perturbative and non-perturbative regions. Because μ_f is not a physical parameter but a byproduct of the assumptions we make in the non-perturbative regions, the physical cross section can not depend on μ_f . In finite-order calculations, the dependence on μ_f can therefore be taken as a measure of the theory uncertainty of a given observable.

2.1 NLO Calculations

Let us now consider cross sections with n jets in the final state. m now runs from n to ∞ . The perturbative parts of eq. 1, i.e. $\mathcal{M}(\{p, a\}_m)$, can be approximated by a power series in α_s . Each term can be visualized in the form of a Feynman diagram. Strong vertices in the Feynman diagrams correspond to factors α_s in the cross section: For example, the cross sections for the processes described by the diagrams in fig. 3 and 2 are both proportional to α_s . They are the leading order (also called born level) diagrams for dijet production. Diagrams of higher order in α_s contribute too to the observed cross section. Because of the relatively large value of α_s , it is very important not to neglect higher-order contributions in QCD processes. However, due to the limits of current algorithms, only next-to-leading order calculations can be performed, i.e. all terms of order α_s^{n+1} and larger (for n -jet production) are neglected because they can not be calculated at present time.

Considering only processes with n jets and taking only terms proportional to α_s^{n-1} and α_s^n into account, the sum in eq. 1 now runs from $m = n$ to $n + 1$. The matrix elements can be approximated as

$$|\mathcal{M}(\{p, a\}_n)|^2 = \left| \mathcal{M}^0(\{p, a\}_n) + \frac{\alpha_s}{4\pi} \cdot \mathcal{M}^1(\{p, a\}_n) + \dots \right|^2 \approx \underbrace{|\mathcal{M}_n^0|^2}_{\propto \sigma^B} + \underbrace{\frac{\alpha_s}{2\pi} \cdot \Re(\mathcal{M}_n^0 \cdot \mathcal{M}_n^1)}_{\propto \sigma^V} \quad (2)$$

and

$$|\mathcal{M}(\{p, a\}_{n+1})|^2 \approx |\mathcal{M}_{n+1}^0|^2 \propto \sigma^R. \quad (3)$$

The first summand of the first term is the leading order matrix element and corresponds to the cross section on *born* level, σ^B . The second summand of the first term corresponds to *virtual* corrections due to interference of loop- or box-diagrams and born level diagrams (cf. fig. 5). The second term corresponds to so-called *real* corrections with an additional parton in the final state (cf. fig 4). Additional Feynman diagrams (e.g. the virtual corrections to \mathcal{M}_{n+1}^0) are neglected because they are of order α_s^{n+1} or higher. Thus we can write the next-to leading order cross section as

$$\sigma^{NLO} = \sigma^{B(orn)} + \sigma^{V(irtual)} + \sigma^{R(eal)}, \quad (4)$$

where the three terms can be calculated from the three terms mentioned above by performing the phase space integral and summing over flavors etc. as needed. Trying to calculate each term separately, one finds that σ^V and σ^R are divergent. However, in the sum, the observed divergences cancel.

It should be mentioned that both the leading order and the α_s^3 loop diagrams also introduce divergences. These can be removed by requiring a physical (measurable) region in phase space (extremely collinear or soft jets cannot be resolved) or a redefinition of physical quantities (masses and coupling constants). This leads to the running of the coupling constants and introduces a second scale, the renormalization scale μ_r .

The two separate divergences that only cancel when the different contributions are summed up pose difficulties for computer-based calculations. Numerically, one can not calculate infinite integrals with a finite difference. One therefore has to cancel the divergences before evaluating the integrals. The phase-space slicing method solves the divergences by separating out the divergent regions of the phase space. The non-divergent regions can be integrated over without problems. In the regions close to the poles, one can approximate the integrands so that the two contributions cancel without performing the integrations. Problems with this method are that it is process dependent and introduces another arbitrary scale, the cutoff parameter in the phase space slicing.

The subtraction method is more general. It relies on adding a counter term to the virtual integral and subtracting the same term from the real contribution. This has to be done before performing the integral. The counter terms contain the divergences, so that the integration can now be performed safely. The difficulty lies in calculating the counter terms $d\sigma^A$.

2.2 Monte Carlo Integration

Rather than integrating (analytically) over the whole phase space, Monte Carlo event generators use a (pseudo-) random number generator to generate possible phase space configurations/events. These are then weighted according to the phase space/matrix element of the pertinent Feynman graphs. Adding up all weights and dividing by the number of generated events gives the total cross section. Most programs use importance sampling, i.e. generate more events in regions where the integrals diverge.

3 The Program nlojet++

`nlojet++`[1] by Zoltan Nagy is a program which calculates jet cross sections in next-to-leading order using the Monte Carlo technique. It contains NLO matrix elements for several processes. For the studies presented here, deep inelastic scattering events (with two or three partons in final

state in leading order) were studied. A user routine was set up that is able to save information about the generated events in the form of a ROOT tree. This tree can then be read back in and studied further, e.g. with different jet algorithms, cuts, ...

3.1 Some Remarks on Nomenclature

In NLO calculations like the ones performed by `nlojet++`, one needs to be careful with the definition of the word event. For each **event** with a fixed phase space (fixed x , Q^2), `nlojet++` generates several **configurations**, corresponding to the different Feynman graphs that contribute (e.g. leading order terms, interference terms etc.). Each configuration consists of several particles: Incoming and outgoing lepton, incoming parton and two to four outgoing partons, each described by its momentum. Each configuration has three **contributions**, one from gluons, one from up-type and one from down-type quarks.

Each configuration has a type. The possible types are **born**, **real**, **sub**, and **fini**. The latter can be further split into **finix** and **fini1**. **born** refers to leading order terms (i.e. QCD-Compton or boson-gluon fusion processes for dijet production), **real** to NLO terms with one additional parton in the final state. **sub** refers to dipole (subtraction) terms, **fini** to finite remainders of the interference terms after the divergences have been subtracted. For each phase space configuration, exactly one configuration of type **born**, **real**, **fini1** and **finix** is generated as well as several subtraction terms.

3.2 Usage

To generate events or calculate cross sections, the user has to provide a *user routine*. This needs to be compiled using `libtool` so it can be dynamically linked from the main program. A sample makefile can be found in the folder `/afs/desy.de/user/f/fleish/h1/nlojet/eptojets`. The program can then be started with

```
nlojet++ --calculate -c full -n name -u user-routine.la [--max-event
number] [--save-after number] [-s seed] [-d outputdir].
```

The option `--calculate` tells the program to start the event generation routines. With `-c option` one can choose between only LO contributions (*born*), only NLO contributions (*nlo*) or both (*full*). *name* is a name associated with the output file. This enables one to run `nlojet++` simultaneously with different options/user routines. The last four options are optional. `--max-event number` sets the number of events to be generated. If no such number is given, the program runs until it is terminated from the outside. `--save-after number` sets the number of events that are kept in memory before they are written to file. The default value is 10000. `-s seed` sets a seed for the random number generator. If no argument is given, the seed is calculated from the system time. *outputdir* is the name of the directory where the output files will be saved (default: `./output`).

The output is saved as a binary file². It contains information about the number of generated events as well as any histograms specified in the user routine. The filename consists of the name that was given earlier as well as some information about the generated process. A sample filename would be `./output/name-dis-full-2jet`.

If one wants to access the information in the histograms, one needs to call

```
nlojet++ --add -u user-routine.la [-d inputdir] [-r outputdir] name1
[name2 ...].
```

²One can switch to output in text format by using the option `--txt`. Version 4.0.1 of `nlojet++` has a bug, it needs a dummy argument after `--txt`, e.g. `--txt x`.

This adds the results from the earlier runs and normalizes them. Statistical errors are also calculated.

inputdir and *outputdir* refer to the directory where the binary files can be found resp. where the results should be placed (Defaults: *./output* resp. *./result*). *name1*, *name2* etc. refer to the names specified during generation, not to the full filename! The corresponding files should have been generated using the same user routine. The result file is placed in the *outputdir* directory. The filename is always *result-0*, the path depends on the simulated process, e.g. *./outputdir/dis/2-jet/full/result-0*. This file contains information about the filled histograms (titles, bin limits, values and their errors) in text format. It can be read in and parsed to produce a graphic output.

3.3 The User Routine

The user class should inherit from one of the existing base classes. A valid user routine needs to contain at least two functions, `initfunc()` and `userfunc()`. The first one is called once at the beginning of the calculations. It can be used to initialize global variables and to book histograms. The user function is called once for each event. It can be used to fill histograms or write information to file.

The functions `inputfunc()` and `psinput()` can be used to set the number of partons in the final state or the available phase space (beam energies or cuts in x , Q^2 , ...).

3.4 Writing out Events

In its original form, the output of `nlojet++` consists of the histograms that were filled in the user function. If one fills the histograms with weights calculated by `nlojet++`, they correspond to (differential) cross sections. However, the physical cross sections depend on the jet algorithm that was used, and of course on the cuts that were made. To be able to use different jet algorithms and cuts, it is therefore desirable to save information about each event that can be used to reconstruct the cross sections afterwards. As all the particles and jets are considered massless, we chose to write out the following information about each event:

1. The momentum of the outgoing electron
2. The momentum of the incoming parton
3. The momenta of the outgoing partons
4. The values of x and Q^2
5. The values of the scales μ_f and μ_r that were used to calculate the event weight
6. The event weight calculated by `nlojet++`
7. The value of α_s used to calculate the event weight
8. The type of event (born, real, ...)
9. The order of α_s
10. The values of the PDF for the three contributions (gluon, up- and down-type quarks) used to calculate the weight

11. For events of type `finix` and `fini1`: Two/three weights needed for the scale independent decomposition of the matrix element
12. For `finix` events: The value of the `_M_fini.x`-variable (needed to re-calculate the PDF for this type of event).

To be able to manipulate the weights with their three contributions comfortably and independently of the `nlojet++` libraries, the class `NLOweight` was created. Events are saved as objects of the `NLOevent` class. Some general information (e.g. beam momenta) is saved only once per file as it does not change on an event-to-event basis.

To make sure that no information is lost/that the write-out procedure worked, some simple histograms were filled during the write-out run and compared to histograms filled using events read in from the file.

The user routines set up to write out events (`write.cc` and `write_flat.cc`) can be found in the folder `/afs/desy.de/user/f/fleish/h1/nlojet/epetojets`.

3.5 Changes Made to the `nlojet++` Source Code

To be able to use ROOT classes as well as my own classes for NLO weights, the compilation/linking routine had to be changed. A sample Makefile can be found in the appendix. The necessary routine `my-create-nlojet-user` can be found in the folder `/afs/desy.de/user/f/fleish/h1/nlojet/epetojets` as well as the appendix. One also needs to set the library path, e.g. via executing `set_lib_path.sh` in the same folder.

The source files for the classes `NLOevent` and `NLOweight` can be found in the folder `/afs/desy.de/user/f/fleish/h1/nlotree`. The main routines of `nlojet++` are `main.cc`, `main_calc.cc`, and `main_add.cc` in the folder `nlojet++-4.0.1/src`. `main.cc` parses the first command line argument and then calls either the adding or the calculating routine. `main_calc.cc` or `main_add.cc` then parse the rest of the arguments, load the requested user routines and start the calculation/addition of histograms. In the `main.cc` routine, no changes were made. In the `main_calc.cc` and `main_add.cc` routines, a small change was necessary: For some reason, loading the custom-made user routine (including the root libraries) breaks the parsing of the command line arguments. This causes the main routine to start parsing again from the beginning and calling the calc/add routine again. It can be fixed by placing `optind = argc`; somewhere after `handle = lt_dlopen(user);`. Two smaller changes were also made to `main_calc.cc`: The line `{"txt", required_argument, 0, 0 }` was changed to `{"txt", no_argument, 0, 0 }` to reflect the fact that the option `--txt` does, in fact, not require an argument. The line `<<"Random Seed Value : "<<seed<<"\n"` was added after `<<"Time rate In+1:In : "<<in.time<<":1\\n"` to make reproducing results easier.

The actual cross section calculations are controlled by the routines `nlo-integral_i1f0.h` and `nlo-process_i1f0.h` in the folder `nlojet++-4.0.1/nlo-core/bits/` (copies of these files are placed in the include directory when compiling them). The function `calculate(unsigned long int mxne, unsigned int time_rate_rf)` (in `nlo-integral_i1f0.h`), sets up the phase space, calls the `initfunc()`, and contains the event generation loop. In this loop, events are generated, the amplitude calculations for the different contributions are set up and the user function is called once for every contribution. At the end of the loop, the function `end_of_event()` is called which increments the histograms as well as the event counter that determines when the histograms are being written to file. However, in the original version, the `end_of_event()` function was also called after a “`numeric_error`” or a “`fp_exception`” was caught (without the user function being called before). This led the counter controlling the histograms to differ from the one controlling the event generation loop, which caused problems when determining when to

close the file with the ROOT tree. Because of that, I took out the calls to the `end_of_event()` function after catching the errors.

Inside the `nlo-process_i1f0.h` routine, the weights of the different configurations and contributions are calculated. Because we need access to some previously private data members (e.g. the scale independent decompositions of the matrix element), I removed the “`private:`” in line 190.

The jet finding and recombining of partons to jets was performed inside the user function using the routine described in `h1jet.cc`. It implements a simple k_T algorithm with an p_T recombination scheme. There was a bug that caused the recombination routine to give back jets with a wrong azimuthal angle ϕ in the Breit frame, leading to a wrong jet pseudorapidity in the laboratory frame. The fixed recombination routine can be found in the folder `/afs/desy.de/user/f/fleish/h1/nlojet/eptojets`.

3.6 Scale Dependence of the Weights

For the cross section calculation, one needs to specify the values of the renormalization and factorization scale. The nominal values used for the analysis presented here are $\mu_f = \sqrt{Q^2}$, $\mu_r = \sqrt{\frac{Q^2 + \langle p_T \rangle^2}{2}}$ where $\langle p_T \rangle$ is the mean transverse momentum in the Breit frame of all jets with $p_T > 5$ GeV.

The weight calculated using the `amp(mu2f, mu2r)` function can be parameterized as follows:

$$w = W_{PS} \cdot \text{pdf}\left(\frac{x}{x_f}, \mu_f\right) \cdot \alpha_s^n(\mu_r^2) \cdot \mathcal{M}(\mu_f, \mu_r) \cdot \alpha_{em}^2(Q^2) \cdot (\hbar c)^2 \quad (5)$$

Here,

w is the event weight with its three components (the total weight is given as the sum of the three components),

W_{PS} is a (scalar) phase space weight,

$\text{pdf}\left(\frac{x}{x_f}, \mu_f\right)$ refers to the value of the parton distribution function for the incoming parton (three components),

x_f is an additional parameter needed to describe `fini1` configurations ($x_f = 1$ otherwise),

α_s is the strong coupling,

n is the number of strong vertices,

$\mathcal{M}(\mu_f, \mu_r)$ is the weight from the matrix element (Feynman diagram) without taking into account the coupling constants.

$\alpha_{em}^2(Q^2)$ is the electromagnetic coupling of the photon to the electron and quark,

$(\hbar c)^2 = 1$ is the factor needed to convert the cross section from GeV^{-2} to pb.

Of these factors, W_{PS} , α_{em}^2 , and $(\hbar c)^2 = 1$ are scale independent. pdf depends on μ_f . It is calculated using an external PDF library linked to `nlojet++`. These calculations can be redone using another scale independently of `nlojet++` if one wants to vary the factorization scale.

α_s depends on μ_r . It is also taken from the external PDF library. \mathcal{M} is a special case. For

born, **real**, and **sub** events, it is scale independent. For **fini** events, the scale dependence can be parameterized as

$$\mathcal{M}(\mu_f, \mu_r) = \mathcal{M}_0 + \mathcal{M}_f \cdot \log\left(\frac{\mu_f^2}{Q^2}\right) + \mathcal{M}_r \cdot \log\left(\frac{\mu_r^2}{Q^2}\right) \quad (6)$$

The scales represent an arbitrary “border” of perturbative QCD. If one could calculate physical quantities to all orders, they should be independent of any scales. As we can currently only calculate up to next-to-leading order in deep inelastic scattering, there is a residual scale dependence. This can be used to estimate the theory uncertainty due to missing higher order corrections. The convention is to vary scales up by a factor of 2 and down by a factor of 0.5 to get the theory uncertainty. For most of the results presented here, only μ_r was varied as the dependence of the cross sections on μ_f is much weaker.

With the parameterization introduced above, re-calculating the weights for a different value of the renormalization scale is almost trivial: For most configurations, only α_s has to be recalculated. For **fini1** configurations, the matrix element has to be recalculated according to 6.

4 Setup

4.1 Event Selection and Phase Space

All generated events have $150 \text{ GeV}^2 < Q^2 < 15000 \text{ GeV}^2$ and $0.2 < y < 0.7$. For inclusive jet and dijet studies, processes with two outgoing partons on born level were used. For trijet studies, three outgoing partons on born level were required. The kinematical cuts for the three type of studies were different.

inclusive jets: For these studies, calculating observables and filling histograms was done for each jet separately. All jets were required to have a transverse momentum between 7 GeV and 50 GeV in the Breit frame. The pseudorapidity had to satisfy $-0.8 < \eta < 2.5$ in the laboratory frame. Studies using jets with $-0.8 < \eta < 2.0$ were also done.

dijets: Here, observables were calculated once per event. Each dijet event has to have at least two jets with $5 \text{ GeV} < p_T < 50 \text{ GeV}$ and $-0.8 < \eta < 2.5(2.0)$. To avoid a region where perturbation theory is unreliable, we also require the invariant mass of the first two jets to be above 16 GeV.

Some of the first studies were done in a slightly different phase space: The transverse momentum cuts were between $5 \text{ GeV} < p_T < 200 \text{ GeV}$ for all jets, minimum η was -1.0 .

4.2 Jet Algorithms

Three different jet algorithms were studied and compared: Kt[2, 3], anti-Kt[4], and Cambridge-Aachen[5, 6]. All of these algorithms are longitudinally invariant, infrared and ultraviolet safe. They each have a parameter R that is analogous to the cone radius of a cone algorithm.

For combining the partons to jets, the p_T recombination scheme was used. It produces massless jets whose transverse momentum is the scalar sum of the transverse momenta of the constituents. Pseudorapidity η and azimuthal angle ϕ of the jet are given as the p_T -weighted sums of η and ϕ of the constituents.

4.3 Parton Distribution Functions and Strong Coupling Constant

The cteq6-pdf set was used for finding the parton distribution functions. The running strong coupling constant was extracted from the PDF.

4.4 Factorization and Renormalization Scale, Theory Uncertainty

For most of the studies presented here, the factorization scale was chosen as $\mu_f^2 = Q^2$. As the cross section does not depend strongly on μ_f (c.f. section 5), it was not varied to find the theory error.

The choice of the renormalization scale is not as trivial. There are several hard scales (Q^2 and jet p_T) in each event. Several combinations of these scales were tested. The recommended choice is $\mu_r^2 = 0.5 \cdot (p_T^2 + Q^2)$ for inclusive jets and $\mu_r^2 = 0.5 \cdot (\langle p_T \rangle^2 + Q^2)$ for dijets and trijets, where the mean is taken from the first two (three) jets that pass all required kinematic cuts.

The theory uncertainty was calculated by raising the value of μ_r by a factor of 2 and lowering it by a factor of 0.5.

4.5 Statistical Uncertainties

As only a finite amount of Monte Carlo events can be produced, there will always be statistical fluctuations in the weights and resulting uncertainties on the cross sections. However, the goal is to produce enough events so that the statistical uncertainties are much smaller than the theory uncertainty that comes from the scale dependence, and can be neglected.

An estimator for the differential cross section in each bin is given by $d\sigma = \langle w \rangle = \frac{1}{N} \cdot \sum_{i=1}^N w_i$, where N is the number of events that were generated and w_i is the sum of the weights of the different configurations of event i that contribute to that bin, summed over the three contributions (gluons, up, down type quarks). (If the bin is not filled during the i th event, $w_i = 0$.) Its uncertainty is given by $\sqrt{\frac{V(w)}{N}}$, where the variance is defined as $V(w) = \langle w^2 \rangle - \langle w \rangle^2$, assuming no correlations between the events. It would be wrong to use the weight from each configuration separately as the configurations of a given event are correlated with each other. This is the approach used by `nlojet++`.

Most of the results shown here were obtained using 938 samples with two million events each, i.e. 1876000000 events total, not all of which pass the jet cuts. One can see in the plots discussed further down that the statistical error is small compared to the theory uncertainty due to scale dependence. However, there are some regions in phase space where the cross section is low and larger statistics might be necessary.

4.6 Event Generation and Analysis

A given number of trees with NLO events can be generated by calling the script `submit.jobs.sh` in the folder `/afs/desy.de/user/f/fleish/h1/batchscript`. The number of events to be generated can be set in the script `submit.one.nlo.tree.sh`. It should be the same number as the one specified in the user routine! The folder where the results should be copied needs to be made by hand.

A sample analysis can be found at `/afs/desy.de/user/f/fleish/h1/nlotree/NLO_analyse.C`. It analyses a single tree. Histograms are scaled by the inverse of the number of events in the

tree and by the inverse bin width³. The histograms are then saved to files. The results from the different trees then have to be added, e.g. using the script `add.all.histos.sh` which uses the program `hadd` (part of the ROOT libraries, adds histograms and trees inside ROOT files). The final averaging and calculation of the statistical uncertainties is performed using the ROOT macro `NLOplot.C` which will also plot some of the histograms.

5 Results

5.1 Choice of μ_r

For the studies presented here, the factorization scale was chosen as $\mu_f^2 = Q^2$. As can be seen in fig. 7 (middle left), which shows the Q^2 distribution for inclusive jets, the dependence of the cross section on μ_f is small. Thus, when varying the scales to find the theory uncertainty, μ_f was left constant and only μ_r was varied.

There are several hard scales in the DIS process: The photon virtuality Q^2 and the transverse energy/momentum of the jets. It makes sense that the renormalization scale should depend on both of these scales. Several different variants were considered:

- $\mu_r^2 = p_T^2$, where in a multi-jet event each jet would have its own scale and its own weight,
- $\mu_r^2 = \langle p_T \rangle^2$, the mean transverse momentum of all jets with $p_T > 5 \text{ GeV}$,
- $\mu_r^2 = Q^2$,
- $\mu_r^2 = 0.5 \cdot (p_T^2 + Q^2)$, where in a multi-jet event each jet would have its own scale and its own weight,
- $\mu_r^2 = 0.5 \cdot (p_{T,max}^2 + Q^2)$, where $p_{T,max}$ is the largest transverse momentum of the event, and
- $\mu_r^2 = 0.5 \cdot (\langle p_T \rangle^2 + Q^2)$, where $\langle p_T \rangle$ is the mean transverse momentum of all jets with $p_T > 5 \text{ GeV}$.

The latter three choices lead to similar cross sections as well as similar scale dependences (cf. 7, middle right and bottom). For my studies, the fourth variant was used for inclusive jets, the last one for dijets and (using only the first two jets that pass all kinematic cuts when calculating the mean). It can be seen here that the uncertainty that comes from choosing a different scale μ_r is much smaller than the one that comes from varying the numerical value of the scale.

The first, third, fourth and last method are compared in fig. 6, which shows the differential cross section $\frac{d\sigma}{dQ^2}$ for inclusive jets. It can be seen that the fourth and last variants lead to almost the same cross section and scale dependence. Using $\mu_r^2 = Q^2$ gives a similar shape but a slightly lower total cross section. $\mu_r^2 = p_T^2$ gives a slightly higher total cross section and a non-linear scale dependence.

³The newer ROOT versions have a more comfortable way to do this, by calling `Scale(1.0/NEvents, "width")`. However, that functionality is bugged in the currently used version 5.22.

5.2 Comparison of Jet Algorithms

For the inclusive jets and dijet analyses, the same Monte Carlo samples were used (with two outgoing partons on born level). For these samples, all three jet algorithms find exactly the same jets. Thus, the cross sections do not depend on the jet algorithm used. They do depend on the value of the R parameter, though.

Three values of the R parameter were compared: $R = 0.7$, $R = 1.0$, and $R = 1.3$. In 8, the differential dijet cross section is shown for the three values of R . One can clearly see that the scale dependence increases for larger R in most bins, but it increases in the first bin ($5 \text{ GeV} < p_T < 7 \text{ GeV}$). A similar thing can be observed in figs. 9 and 10, which show the double differential cross section $\frac{d\sigma_{dijet}}{dQ^2 dp_T}$ as a function of $f_{\mu_r} := \frac{\mu_r}{\mu_{r,0}}$ for different values of R . For $R = 1.0$, the dependence on f_{μ_r} is stronger in almost all bins. However, in the first p_T bin ($5 \text{ GeV} < p_T < 7 \text{ GeV}$), the μ_r dependence is actually much stronger for $R = 0.7$. In this region, perturbation theory becomes unreliable and some of the calculated cross sections are negative. Because of this, one should not use dijet events with $p_T < 7 \text{ GeV}$, especially not in combination with a jet algorithm with $R = 0.7$. For the inclusive cross sections, one can observe a similar behavior (cf. figs. 11, 12). Here, $R = 0.7$ is favored in all bins.

5.3 Pseudorapidity Range

In fig. 13, one can see the pseudorapidity distribution in the laboratory and Breit frames for inclusive jets. The distribution is similar for the different values of R . The μ_r dependence increases strongly with η . To get a smaller overall scale uncertainty, it is advisable to use a tighter cut on the pseudorapidity, e.g. $-0.8 < \eta_{lab} < 2.0$. In fig. 8, one can see that such a cut leads to a smaller overall scale dependence. In figs. 14 and 15, the p_T and Q^2 distribution for inclusive jets is shown in bins of η_{lab} . One can see that the larger scale dependence in the forward region is not due to the p_T or Q^2 distribution in this region. Even though jets in the forward region tend to have larger p_T and Q^2 , the scale dependence increases in almost all bins in p_T and Q^2 when going to larger η_{lab} (for $30 \text{ GeV} < p_T < 50 \text{ GeV}$ and $-0.8 < \eta_{lab} < -0.5$, there is not enough statistics.). The f_{μ_R} “scans” (also in figs. 14 and 15) show that the scale dependence does in fact increase with η_{lab} and R .

5.4 Parton Momentum Fraction

The parton momentum fraction x_{PDF} is defined as the fraction of the proton momentum that is carried by the interacting parton *in the born configuration*⁴. The x_{PDF} distribution is important for PDF fits. It is shown in fig. 16 for different values of R and η_{max} . The scale dependence is quite large, especially for larger x_{PDF} . For small x_{PDF} , there is a “cross-over”, i.e. the cross section goes up for larger scales instead of down. That might indicate that perturbation theory starts breaking down in this region. This behavior seems to be worst for $R = 0.7$.

Fig. 17 shows the triple differential inclusive cross section $\frac{d^3\sigma}{dp_T dQ^2 dx_{PDF}}$ in bins of p_T and Q^2 for $R = 0.7$ and $\eta_{max} = 2.0$, with the gluon contribution shown separately. Gluon contribution dominates for small jet p_T and small Q^2 . In the largest Q^2 bin ($5000 \text{ GeV}^2 < Q^2 < 15000 \text{ GeV}^2$), quark-induced events dominate.

⁴The parton of the *finix* configuration has a slightly larger momentum than the parton of the associated *born* configuration. For the plots shown here, only the born-level x_{PDF} was used.

5.5 Performance of the Analysis

C	S	Events/Tree	Size	Total Time	Time/Event	CPU Time	Time/Event
0	0	700000	1.8 GB	481.28s	0.688ms	455.82s	0.651ms
0	1	600000	1.6 GB	504.4s	0.841ms	481.16s	0.802ms
0	99	700000	1.7 GB	632.83s	0.904ms	564.85s	0.807ms
1	0	2000000	1.6 GB	1458.7s	0.729ms	1386.06s	0.693ms
1	1	2000000	1.6 GB	1797.62s	0.899ms	1663.61s	0.832ms
1	99	2000000	1.6 GB	1575.74s	0.788ms	1460.1s	0.730ms
0	1 (flat)	700000	1.7 GB	534.75s	0.764ms	508.64s	0.727ms
1	1 (flat)	2000000	1.5 GB	1589.43s	0.795ms	1527.24s	0.764ms
0	0 (skip)	700000	1.8 GB	390.01s	0.557ms	363.25s	0.519ms
1	0 (skip)	2000000	1.6 GB	1281.68s	0.641ms	1142.31s	0.571ms
1	99 (skip)	2000000	1.6 GB	1329.53s	0.665ms	1195.33s	0.598ms

Table 2: Performance of the analysis on h1mpim11. C stands for compression level, S for split level. Flat refers to trees filled with single variables, not objects. Skip refers to analyses where the jet finding was skipped for *fini1* and *finix* contributions.

One motivation for my work was to eventually be able to use the events written out by `nlojet++` for fitting, e.g. to extract the strong coupling constant. To do that, a short analysis that reads in the trees, recalculated the weights and fills some histograms should be as fast as possible. The read speed for Root trees depends on the structure of the tree (split level, basket size) and the compression used for the file containing the Root tree. Several combinations of compression level and split level were tried. Without any compression, a tree (which should not be larger than about 2GB) can fit about 600000–700000 events (for dijet production), depending on the split level. The analyses were made for trees/branches generated and filled with complex objects of type `NLOevent` (containing `TClonesArrays`, `TVector3s`, doubles and integers as data members) as well as for a tree that had just integer, doubles and `TVector3` branches. The results can be seen in table 2. The performance seems to be best using unsplit branches with a no compression. Compressing the files slightly increases the time by about 6%, but decreases the disk space by about two thirds. Changing the basket size of the branch or the cache size of the file in memory might change these results.

Most of the time is actually spent on jet finding (including resetting/setting up the jet finder). It is possible to make the analysis faster by using the fact that some contributions will have the same final state partons as the born level one. The events configurations are produced by `nlojet++` in the order *born*, *fini1*, *finix*, *real*, *subtraction terms*. The *fini1* contribution and one of the subtraction terms have exactly the same partons in the final state as the *born* contribution. The *finix* contribution has one additional parton in the *z* direction which will not influence the output of the jet finder. The analysis time can be reduced by about 20% by just skipping the jet finding for *fini1* and *finix* contributions. One might be able to gain some more time by not recalculating the boost vector for every contribution.

6 Summay and Outlook

I have successfully used the program `nlojet++` to calculate NLO jet cross sections in DIS, and I have developed a method of writing out events from this program, making it possible to change the renormalization scale later on. I have examined the renormalization scale dependence as a

measure of the theory uncertainty, and shown that the choice of $R = 0.7$ for the jet algorithm, together with the cut $0.8 < \eta_{ab} < 2.0$, leads to a smaller overall scale dependence than other choices. However, for $R = 0.7$, some regions in phase space can not be described as well, e.g. the dijet cross section for $5 \text{ GeV} < \langle p_T \rangle < 7 \text{ GeV}$.

The speed of the analysis can and should be optimized further, e.g. by reducing the time spent on finding jets.

References

- [1] Zoltan Nagy. Next-to-leading order calculation of three jet observables in hadron hadron collision. *Phys. Rev.*, D68:094002, 2003.
- [2] S. Catani, Yuri L. Dokshitzer, M. H. Seymour, and B. R. Webber. Longitudinally invariant K_t clustering algorithms for hadron hadron collisions. *Nucl. Phys.*, B406:187–224, 1993.
- [3] Stephen D. Ellis and Davison E. Soper. Successive combination jet algorithm for hadron collisions. *Phys. Rev.*, D48:3160–3166, 1993.
- [4] Matteo Cacciari, Gavin P. Salam, and Gregory Soyez. The anti- k_t jet clustering algorithm. *JHEP*, 04:063, 2008.
- [5] Yuri L. Dokshitzer, G. D. Leder, S. Moretti, and B. R. Webber. Better Jet Clustering Algorithms. *JHEP*, 08:001, 1997.
- [6] M. Wobisch and T. Wengler. Hadronization corrections to jet cross sections in deep- inelastic scattering. 1998.

Thanks

First I want to thank my supervisors Dr. Günter Grindhammer and Roman Kogler, who taught me a lot and answered a lot of questions. Thank you both!

Second, I want to thank Prof. Dr. Joachim Meyer and Andrea Schrader, for organizing the DESY summer student programme and making my stay at DESY a very happy and successful one.

Last but not least, I want to thank Dr. Zoltan Nagy for his very helpful lessons on NLO calculations and `nlojet++`!

A Plots

All of the plots can be found in the folder `/afs/desy.de/user/f/fleish/h1/nlotree/plots/`. They are sorted into folders `histos_{\eta_{max}}_{\{R\}}_{\{jet-algo\}}`, where `jet-algo=0` means kt-jets. As the three jet algorithms that were tested give the same results, only this algorithm was used for the plots. The folder `histos_NLO_analyse_Q2/` contains plots produced using $\mu_r^2 = Q^2$, $\eta_{max} = 2.5$, and $R = 1.0$. The folder `histos_NLO_analyse_pt2/` contains plots produced using $\mu_r^2 = p_T^2$ for inclusive cross sections and $\mu_r^2 = \langle p_T \rangle^2$ for dijet cross sections, $\eta_{max} = 2.5$, and $R = 1.0$. The folder `histos_NLO_analyse_ptmean/` contains plots produced using $\mu_r^2 = 0.5 \cdot (Q^2 + \langle p_T \rangle^2)$ for all cross sections, $\eta_{max} = 2.5$, and $R = 1.0$.

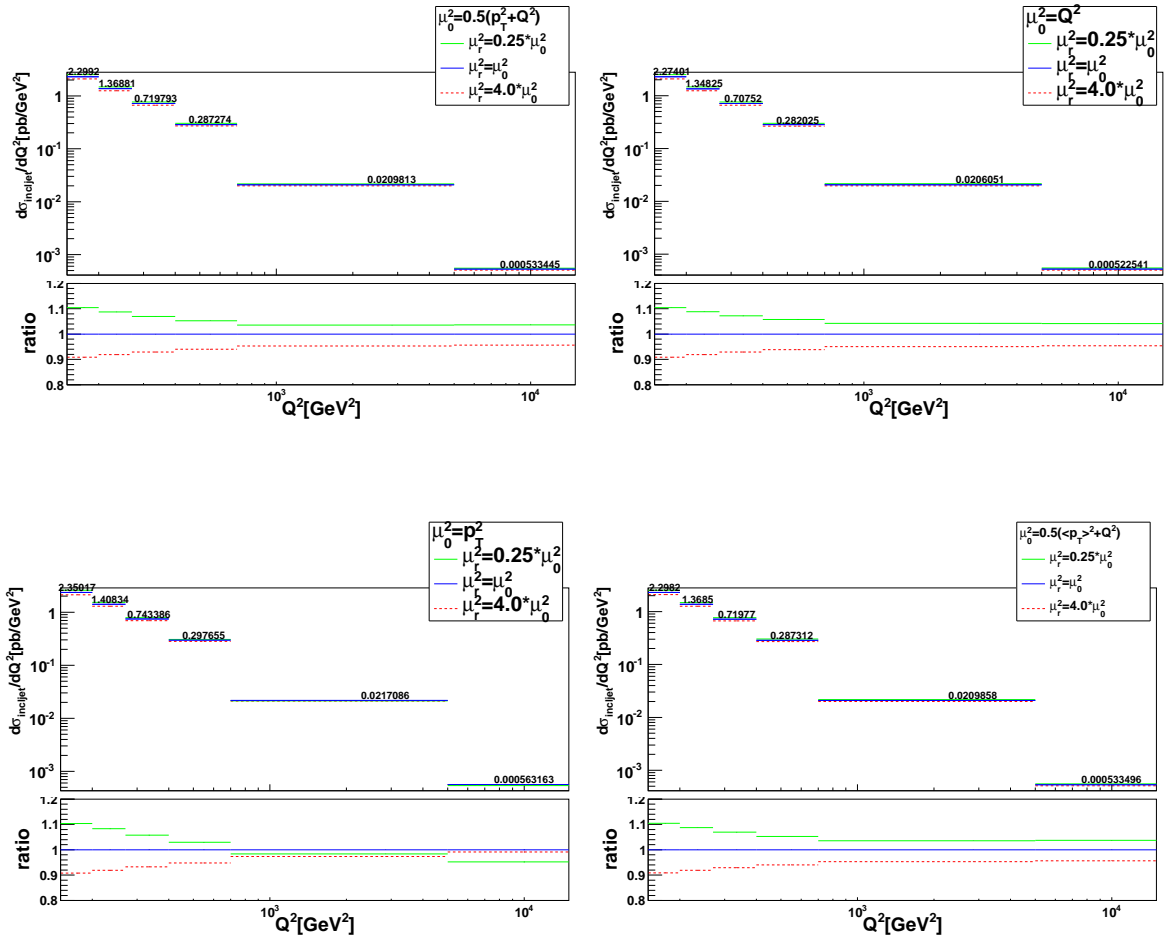


Figure 6: Differential inclusive jet cross section $\frac{d\sigma}{dQ^2}$ for several values of the renormalization scale μ_r .

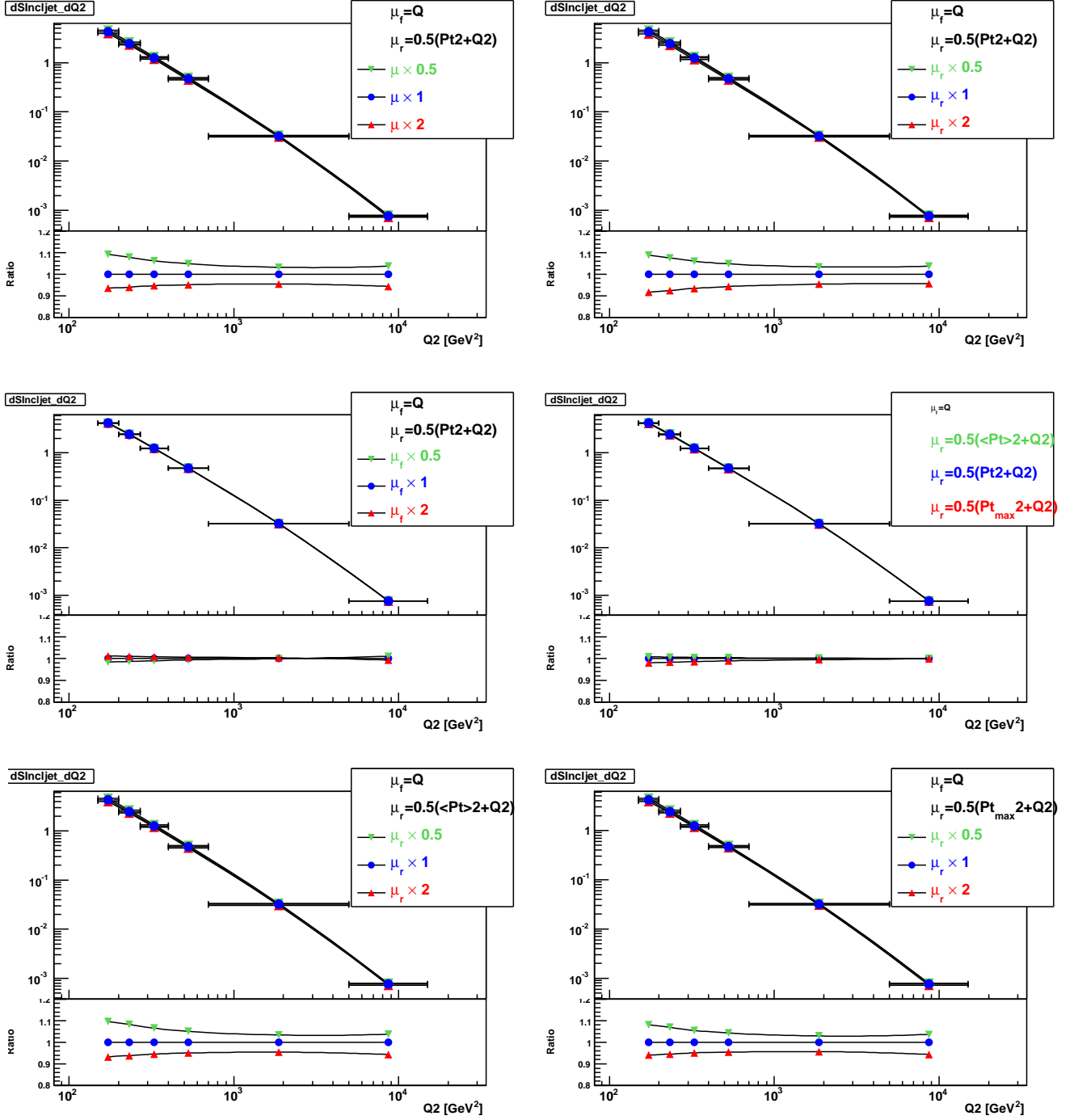


Figure 7: Differential inclusive jet cross section $\frac{d\sigma}{dQ^2}$ for several values of the renormalization and factorization scales.

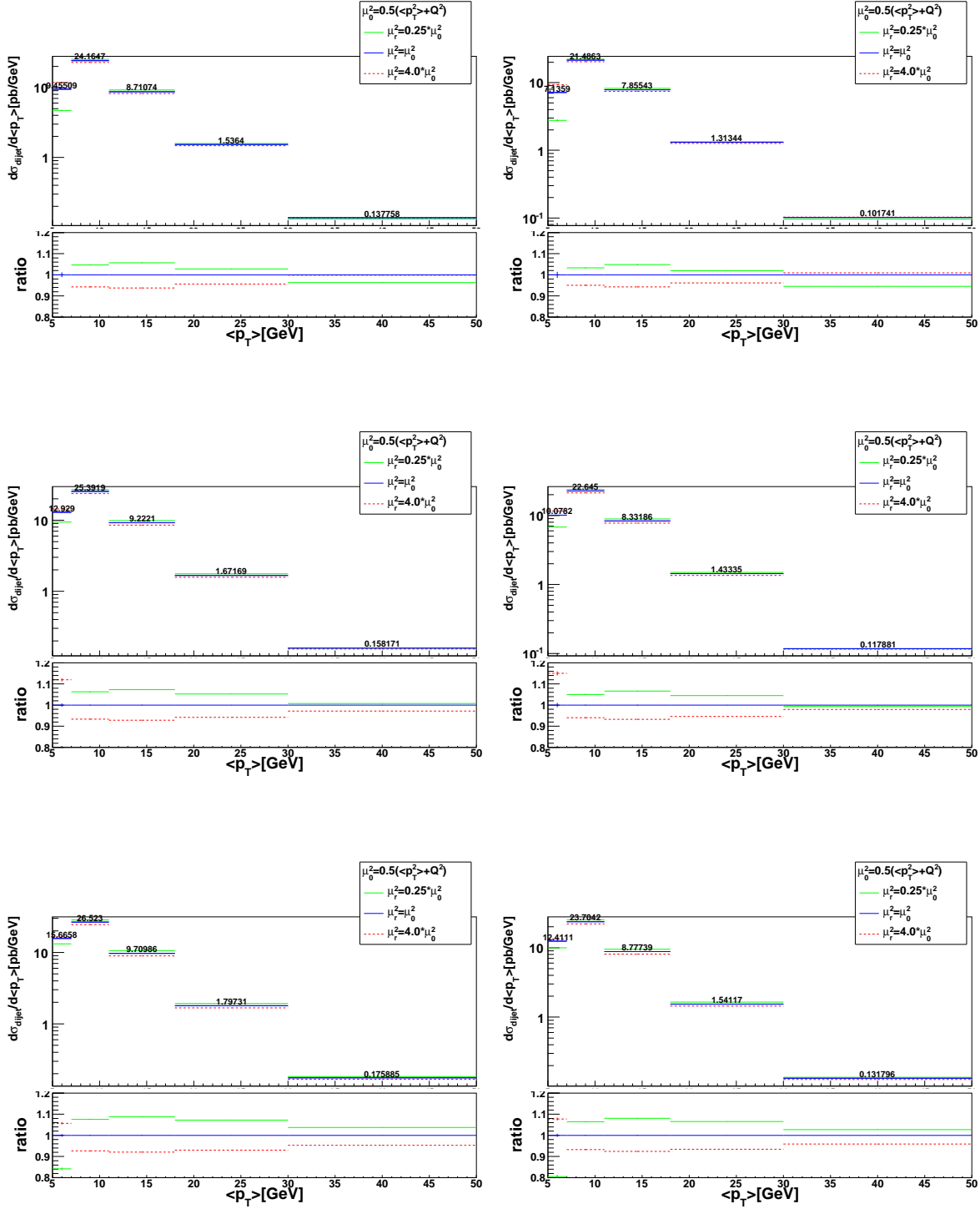


Figure 8: Differential dijet cross section $\frac{d\sigma}{d\langle p_T \rangle}$ for several values of the R parameter (top: $R = 0.7$, middle: $R = 1.0$, bottom: $R = 1.3$) and η_{max} (left: $\eta_{max} = 2.5$, right: $\eta_{max} = 2.0$).

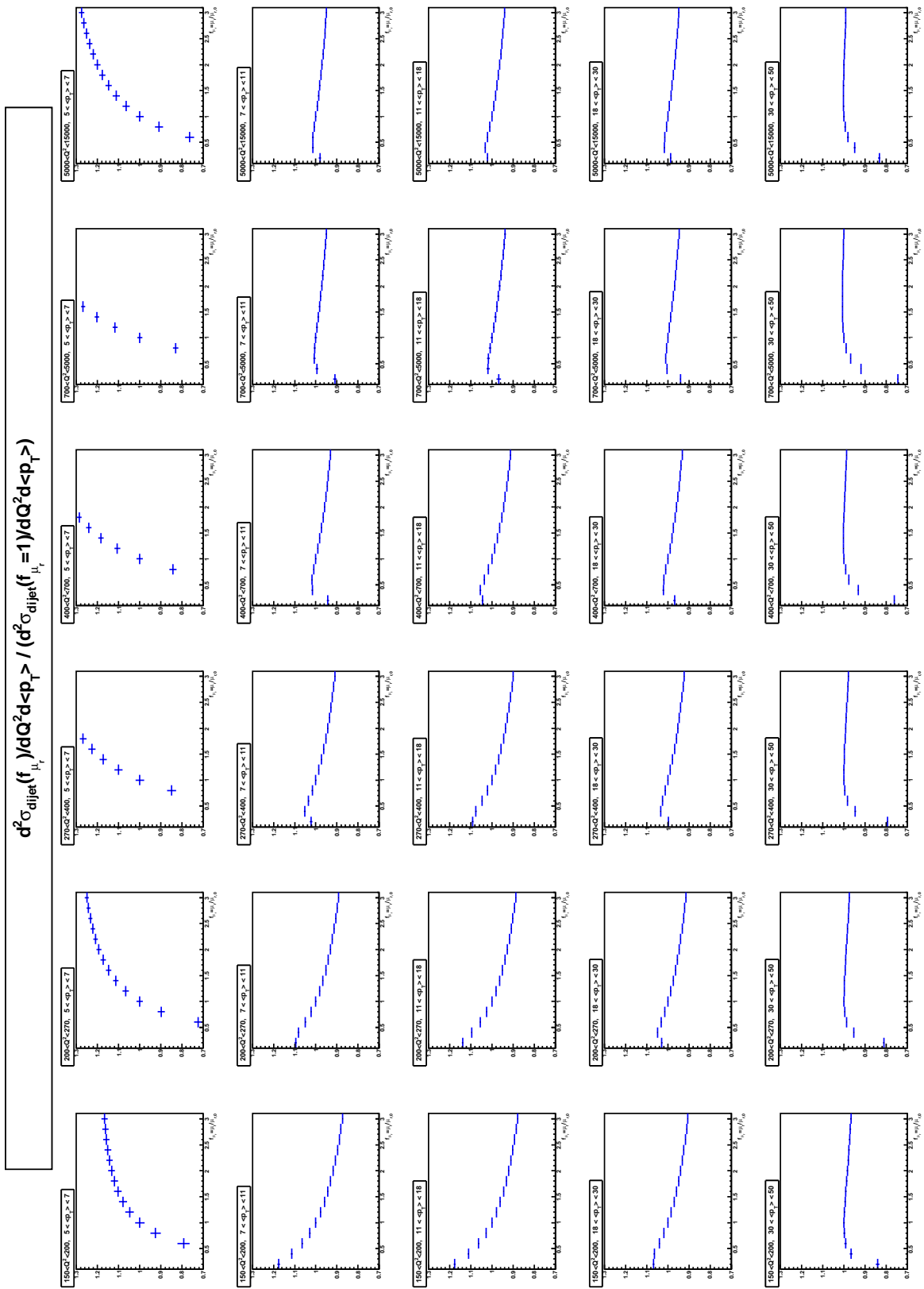


Figure 9: Double differential dijet cross section $\frac{d^2\sigma}{dQ^2 d\langle p_T \rangle}$ for $R = 0.7$ and $\eta_{max} = 2.5$ as a function of $f_{\mu_r} = \frac{\mu_r}{\mu_{r,0}}$, normalized to the value of the cross section at $f_{\mu_r} = 1$.

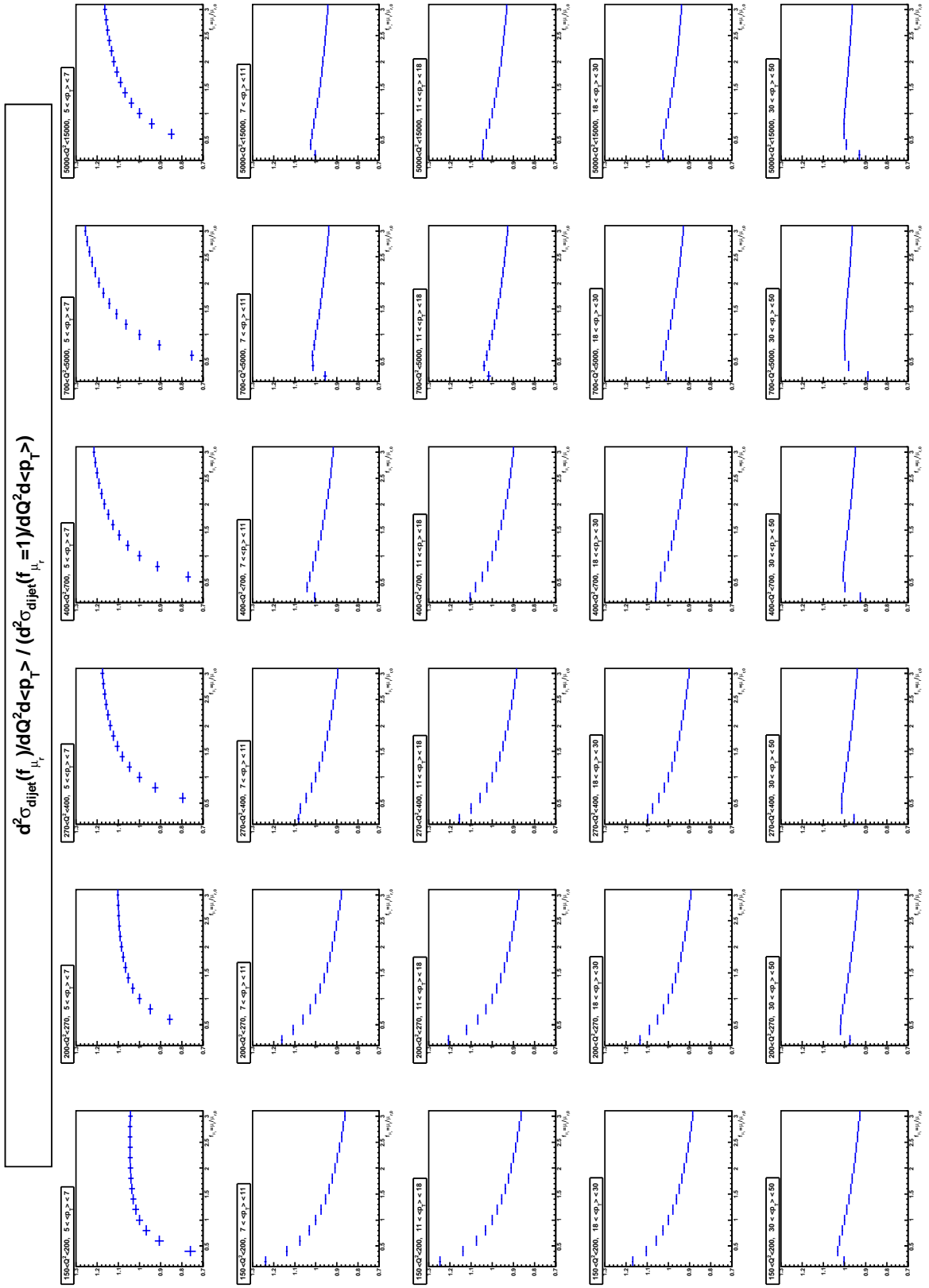


Figure 10: Double differential dijet cross section $\frac{d^2\sigma}{dQ^2 d\langle p_T \rangle}$ for $R = 1.0$ and $\eta_{max} = 2.5$ as a function of $f_{\mu_r} = \frac{\mu_r}{\mu_{r,0}}$, normalized to the value of the cross section at $f_{\mu_r} = 1$.

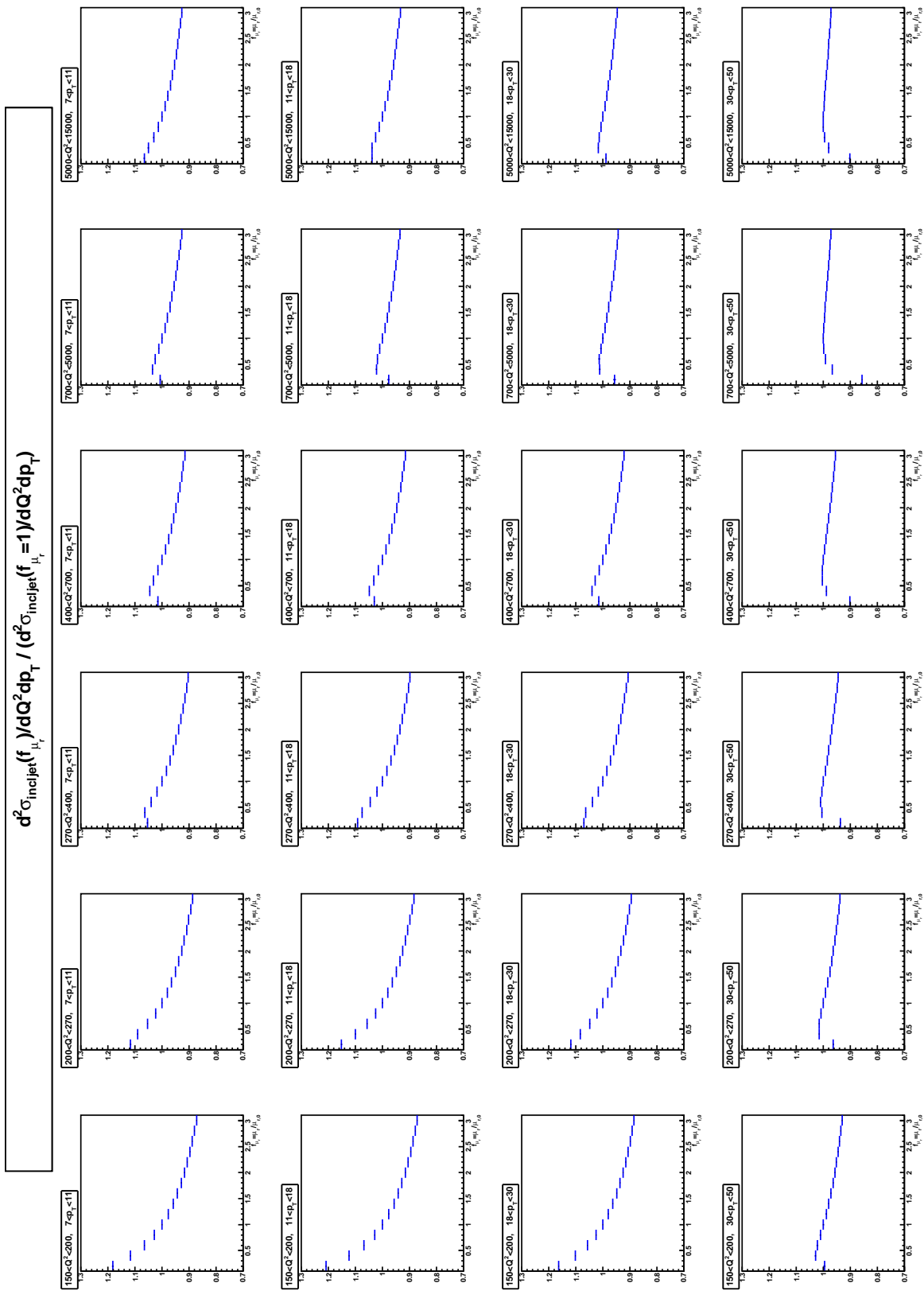


Figure 11: Double differential inclusive cross section $\frac{d^2\sigma}{dQ^2 dp_T}$ for $R = 0.7$ and $\eta_{max} = 2.5$ as a function of $f_{\mu_r} = \frac{\mu_r}{\mu_{r,0}}$, normalized to the value of the cross section at $f_{\mu_r} = 1$.

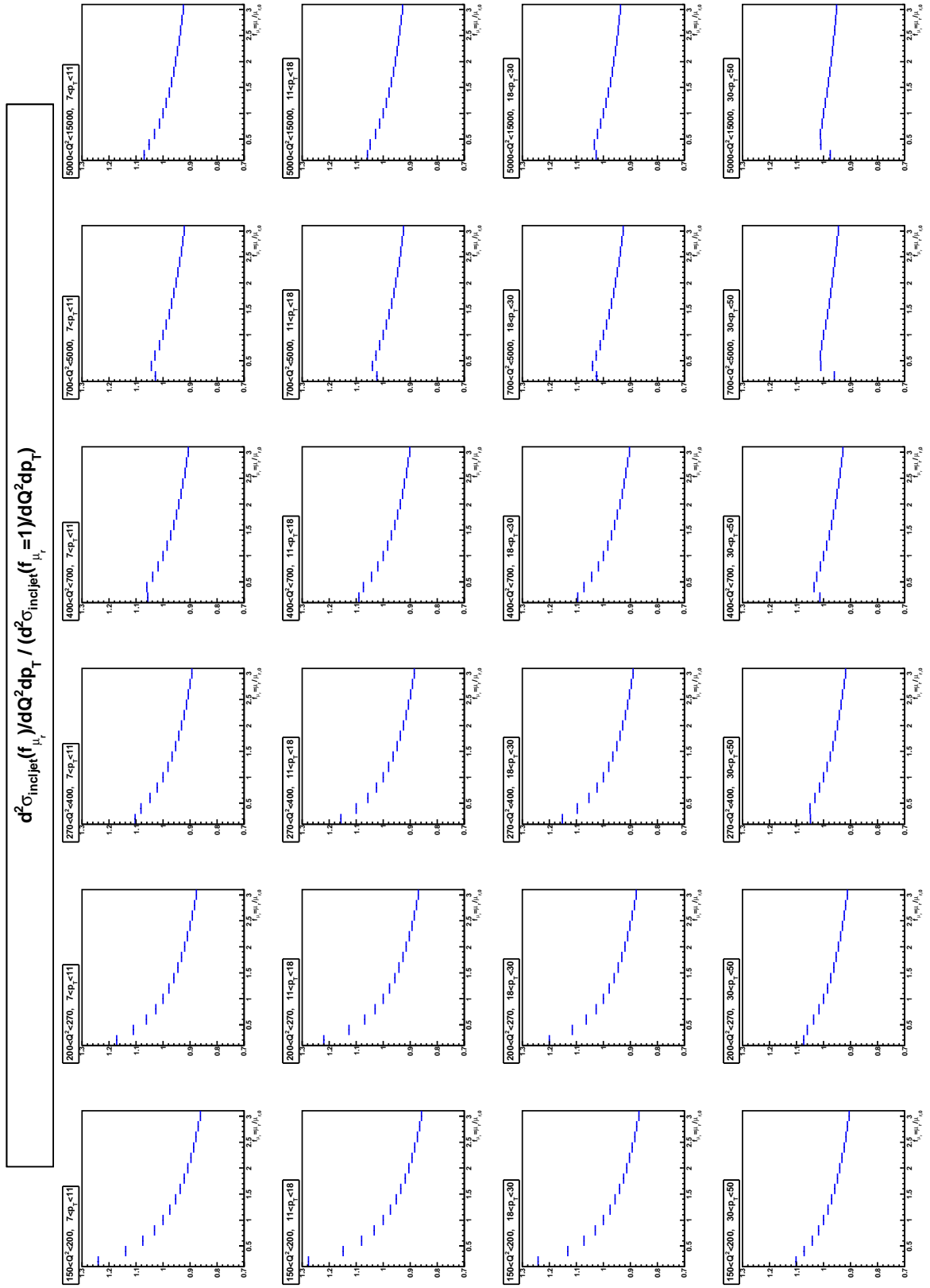


Figure 12: Double differential inclusive cross section $\frac{d^2\sigma}{dQ^2 dp_T}$ for $R = 1.0$ and $\eta_{max} = 2.5$ as a function of $f_{\mu_r} = \frac{\mu_r}{\mu_{r,0}}$, normalized to the value of the cross section at $f_{\mu_r} = 1$.

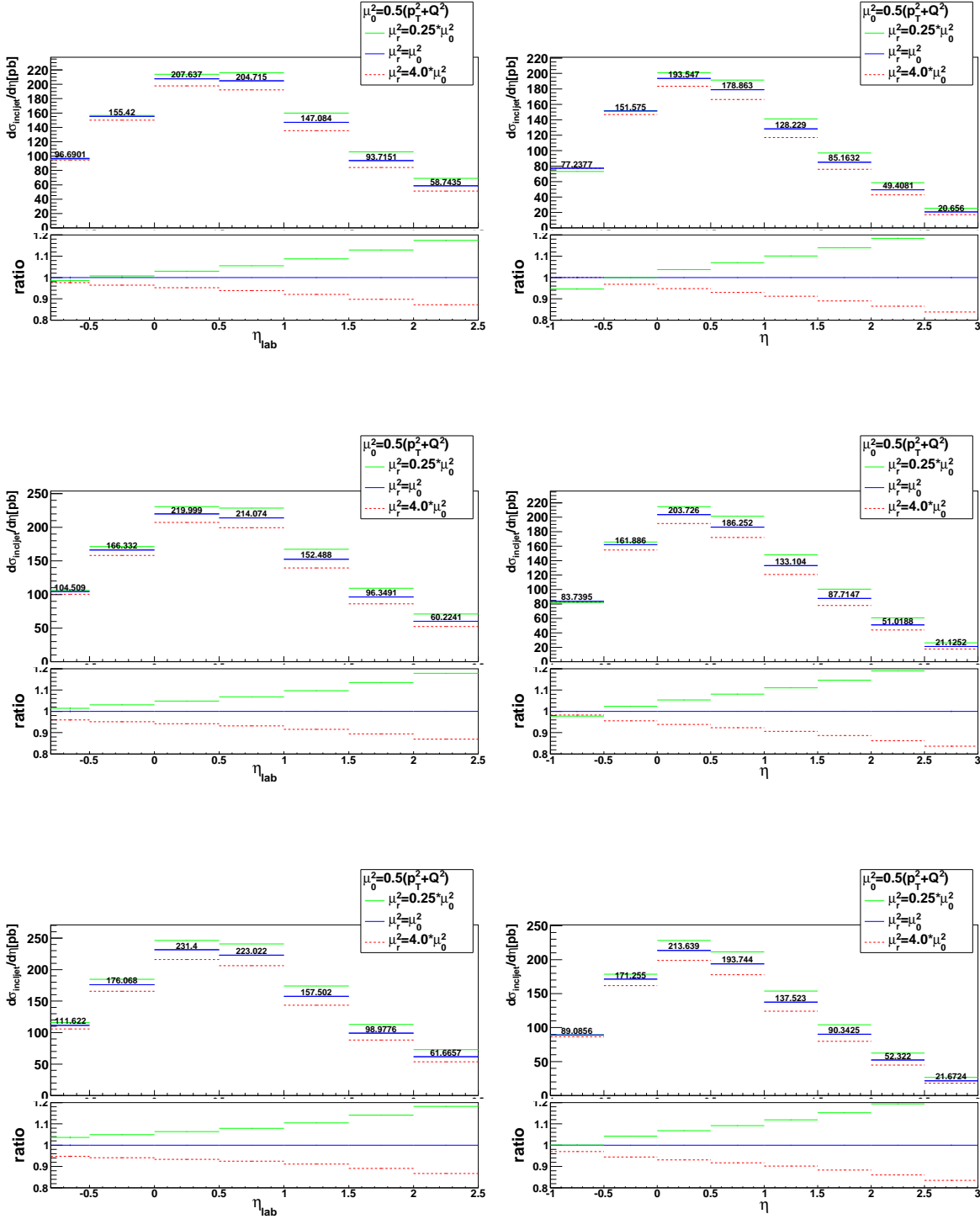


Figure 13: Differential dijet cross section $\frac{d\sigma}{d\eta_{\text{lab}}}$ (left) and $\frac{d\sigma}{d\eta}$ (right) for several values of the R parameter (top: $R = 0.7$, middle: $R = 1.0$, bottom: $R = 1.3$).

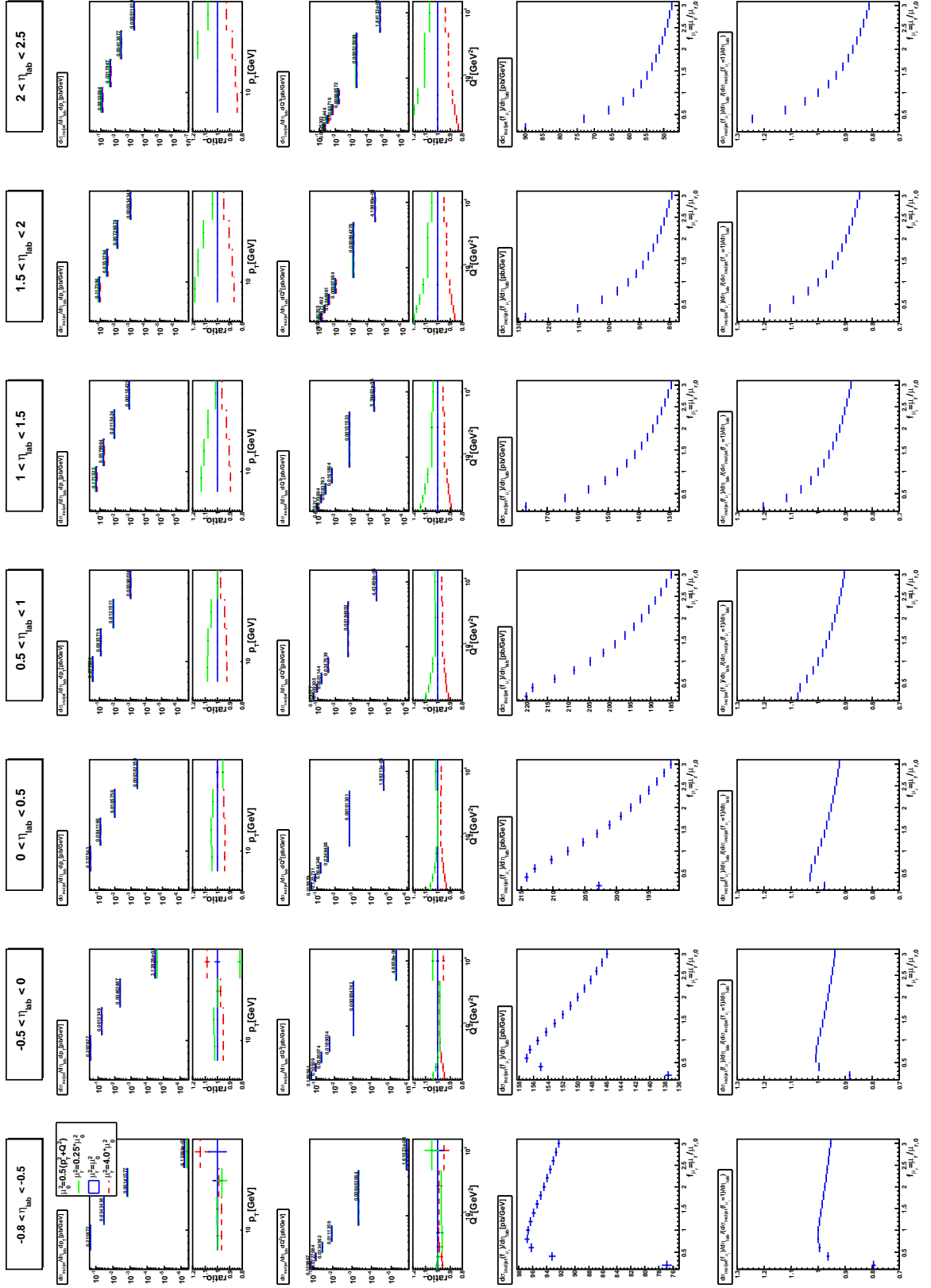


Figure 14: Double differential inclusive cross sections $\frac{d^2\sigma}{dp_T d\eta_{lab}}$ and $\frac{d^2\sigma}{dQ^2 d\eta_{lab}}$ for $R = 0.7$ and $\eta_{max} = 2.5$ as well as the single differential cross section $\frac{d\sigma}{d\eta_{lab}}$ as a function of $f_{\mu_r} = \frac{\mu_r}{\mu_{r,0}}$, total and normalized to the value of the cross section at $f_{\mu_r} = 1$.

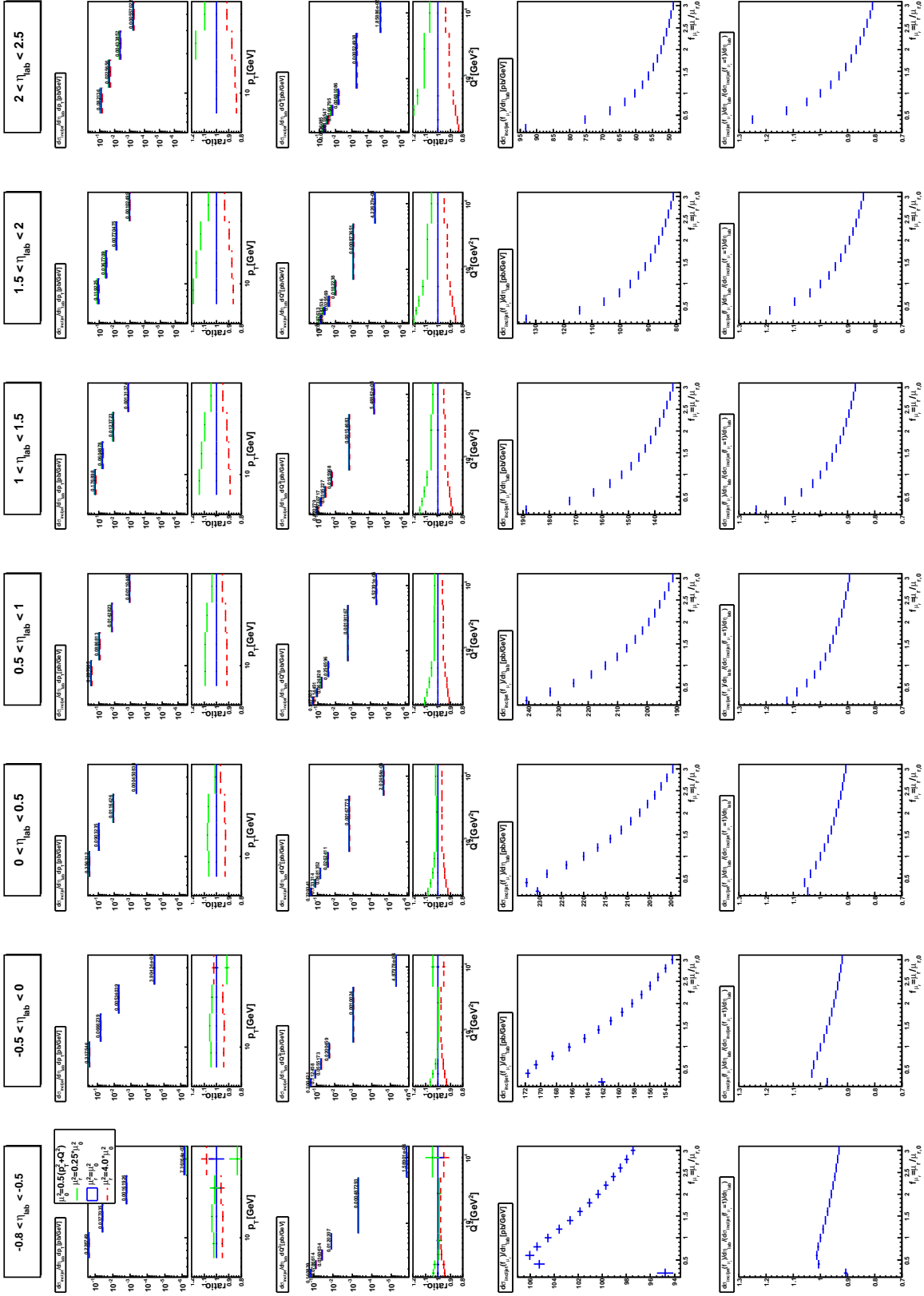


Figure 15: Double differential inclusive cross sections $\frac{d^2\sigma}{dp_T d\eta_{lab}}$ and $\frac{d^2\sigma}{dQ^2 d\eta_{lab}}$ for $R = 1.0$ and $\eta_{max} = 2.5$ as well as the single differential cross section $\frac{d\sigma}{d\eta_{lab}}$ as a function of $f_{\mu_r} = \frac{\mu_r}{\mu_{r,0}}$, total and normalized to the value of the cross section at $f_{\mu_r} = 1$.

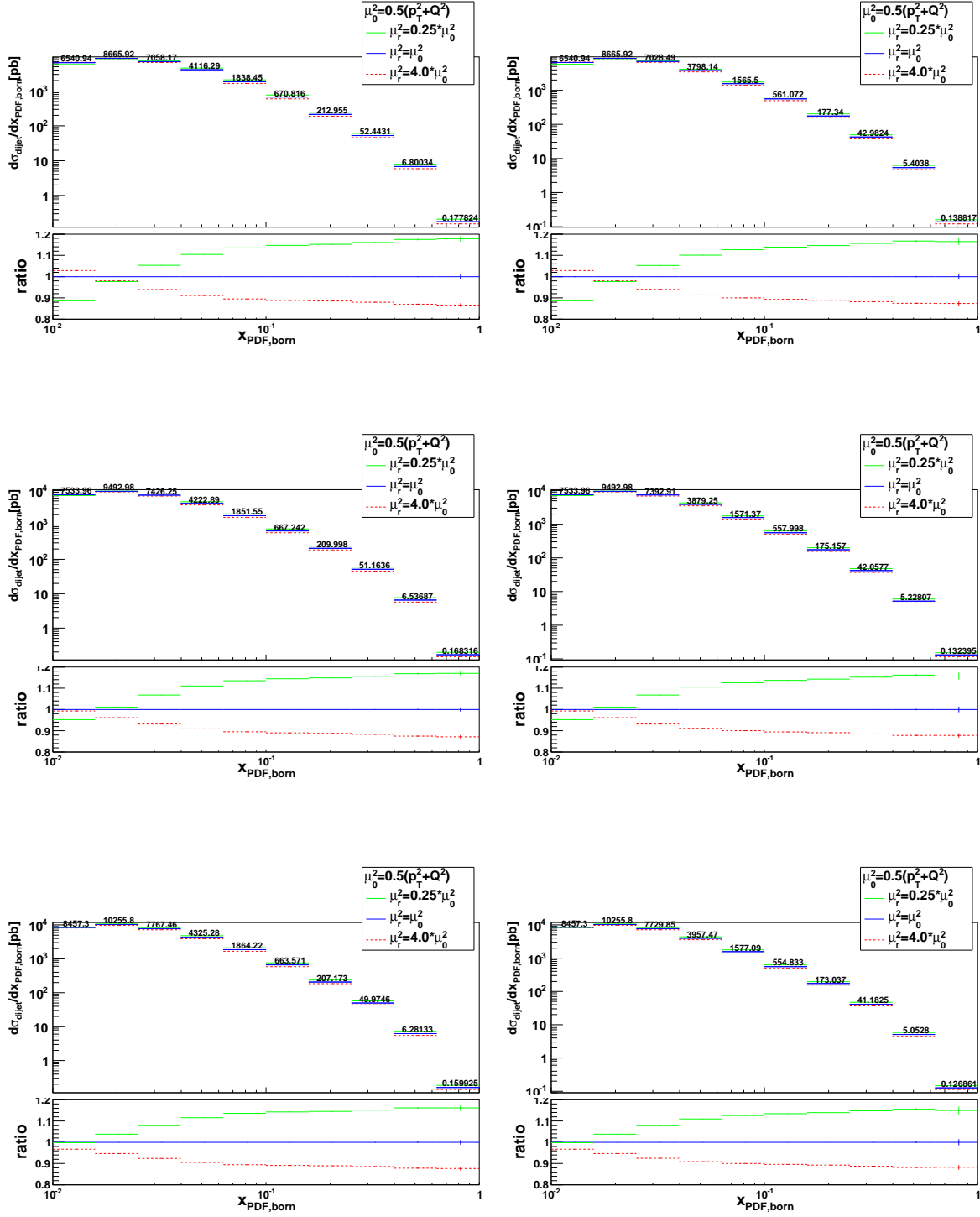


Figure 16: Differential dijet cross section $\frac{d\sigma}{dx_{PDF,born}}$ for several values of the R parameter (top: $R = 0.7$, middle: $R = 1.0$, bottom: $R = 1.3$) and η_{max} (left: $\eta_{max} = 2.5$, right: $\eta_{max} = 2.0$).

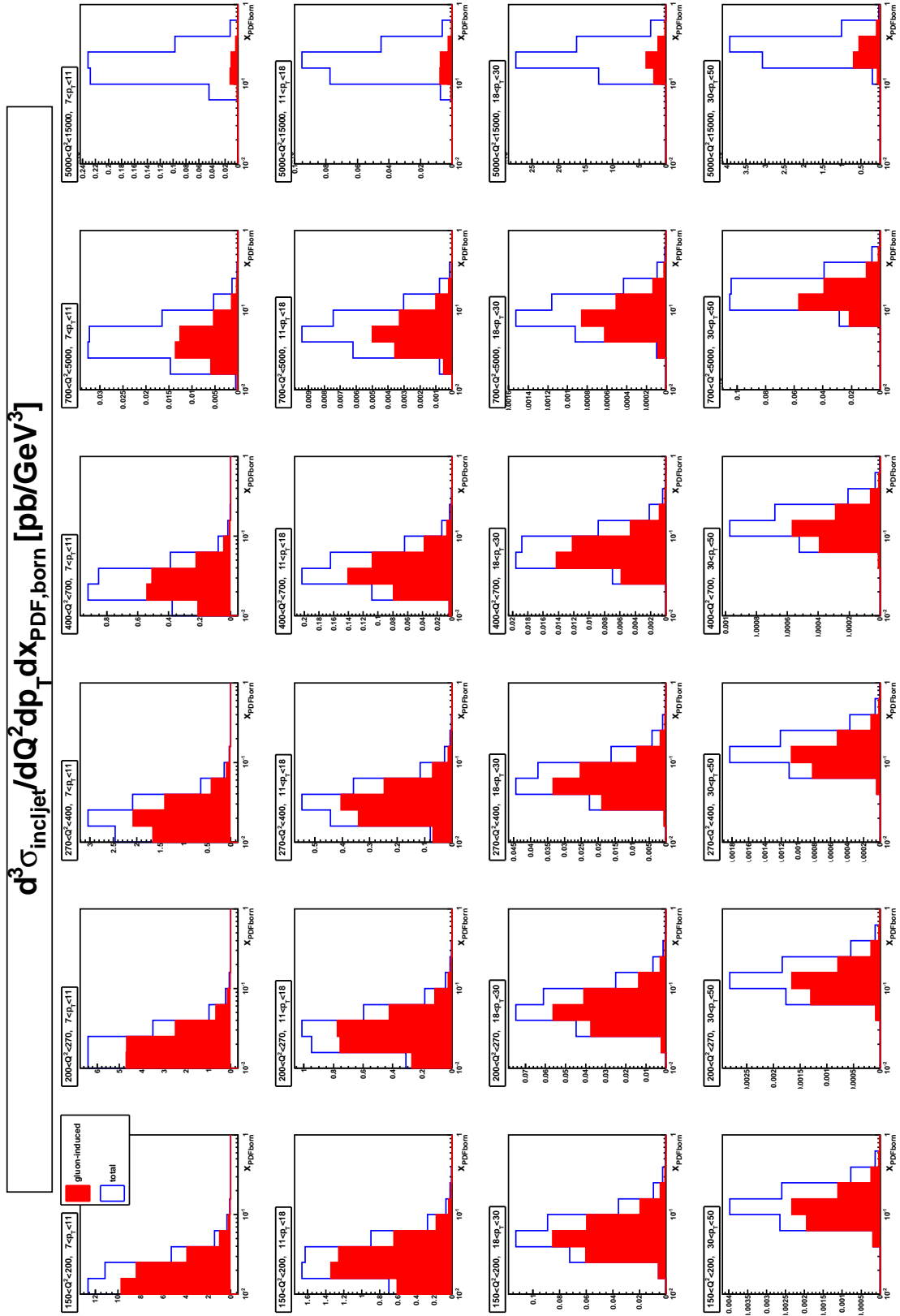


Figure 17: Triple differential inclusive cross section $\frac{d^3\sigma}{dp_T dQ^2 d\mathbf{x}_{PDF}}$ in bins of p_T and Q^2 for $R = 0.7$ and $\eta_{max} = 2.0$.