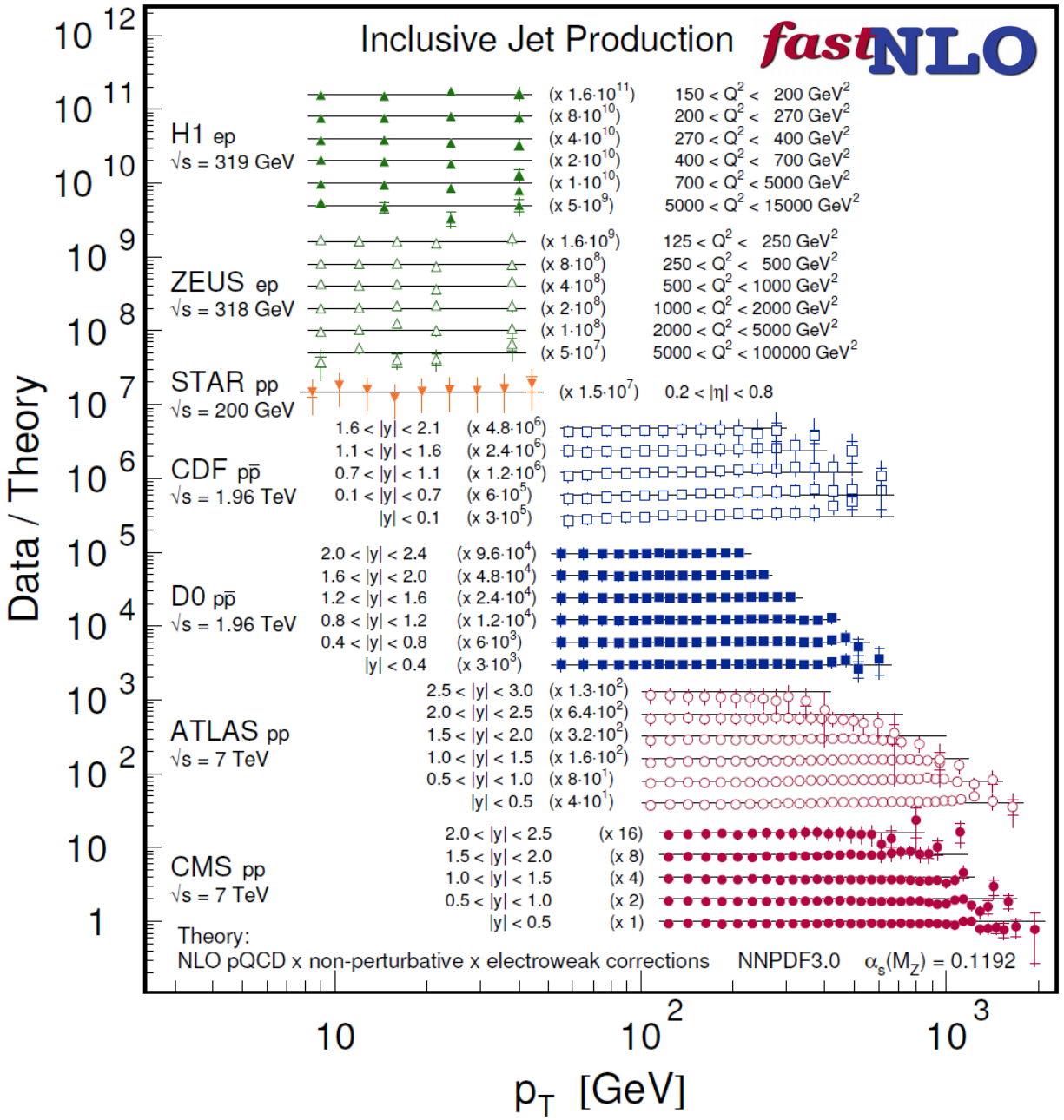
Possibilities and status of interpolation grids

K. Rabbertz (KIT)

Motivation

- Interpretation of experimental data requires reasonably fast theory
- Often: Repeated computation of same cross section:
  - Different PDF sets; PDF uncertainties
  - Variations of renormalisation & factorisation scales μ_R, μ_F
  - Variation of $\alpha_s(M_Z)$
  - SM parameter fits (→ xFitter)
- Jets at NLO were slow; nowadays NNLO in general very demanding!
  - Need procedure for fast repeated computations of higher order cross sections
  - Use interpolation grids like from fastNLO or APPLgrid (both are interfaced to xFitter)

Motivation

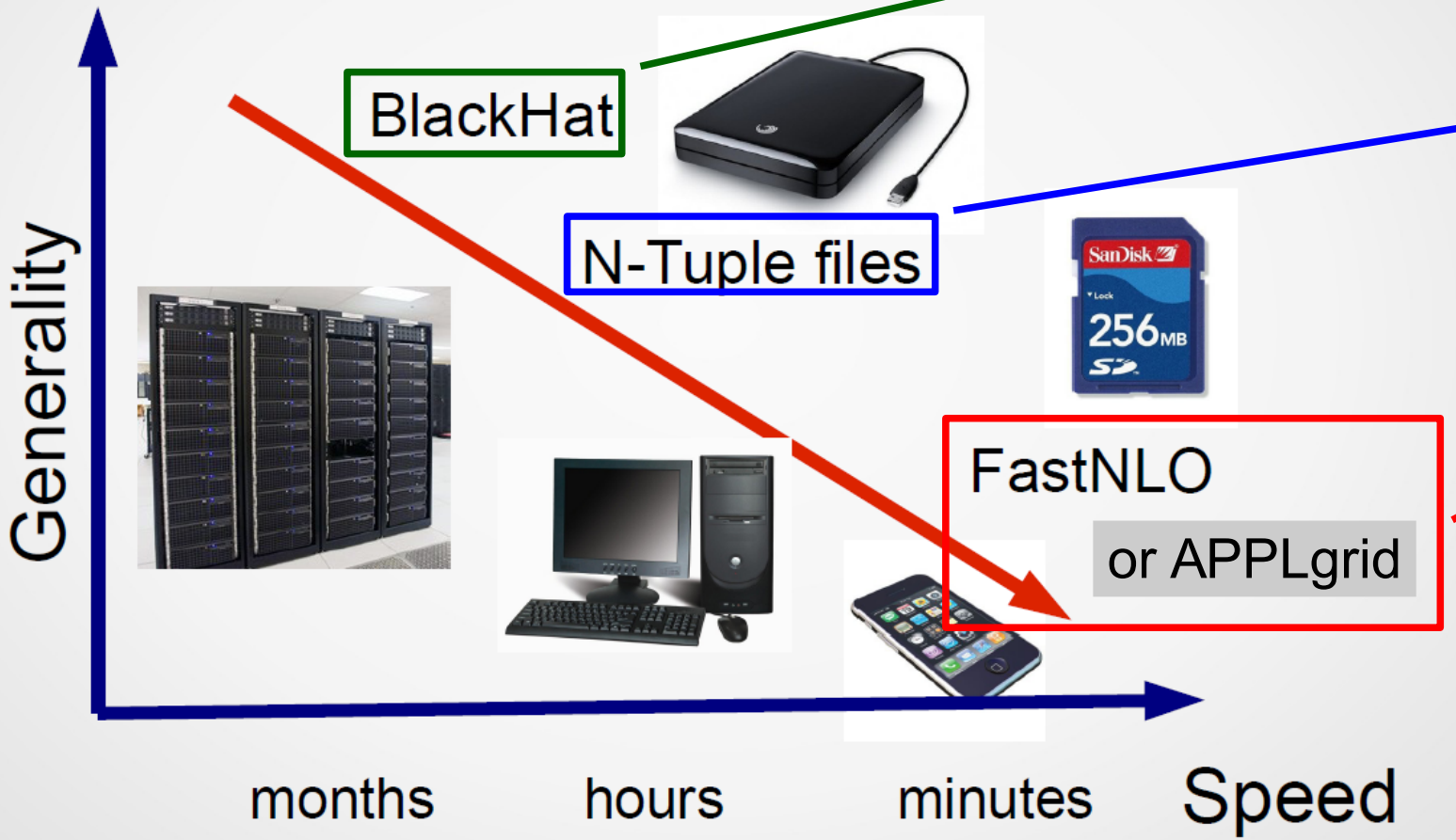


Not only do nice comparisons of theory to data …

but also redo them quickly for various choices of PDFs, scales μ_R, μ_F, α_S

Interpolation benefit

Speed vs Generality



Change everything

Still change observable, phase space, binning

Still change PDFs, μ_R, μ_F, α_s

Slide from Daniel Maitre

Loops and Legs 2014, Weimar, 1th May

Interpolation concept

Implemented in APPLgrid & fastNLO

APPLgrid, Carli et al., Eur. Phys. J. C, 2010, 66, 503.
fastNLO, Britzger et al., arXiv:0609285, 1208.3641.

Use interpolation kernel

- Introduce set of n discrete x-nodes, x_i's being equidistant in a function f(x)
- Take set of Eigenfunctions $E_i(x)$ around nodes x_i

→ Interpolation kernels

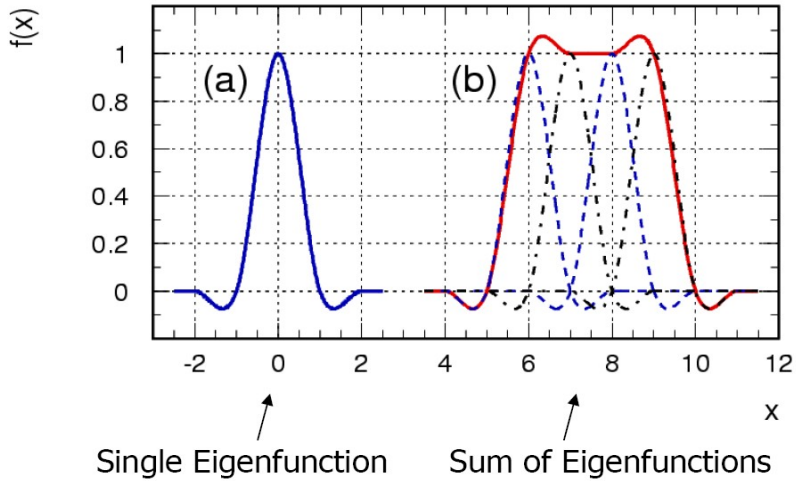
- Actually a rather old idea, see e.g.
  C. Pascaud, F. Zomer (Orsay, LAL), LAL-94-42

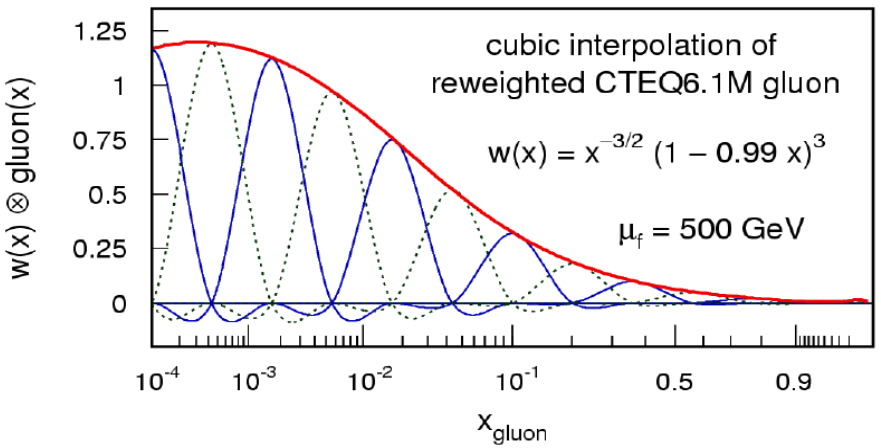
→ Single PDF is replaced by a linear combination of interpolation kernels

$$f_a(x) \cong \sum_i f_a(x_i) \cdot E^{(i)}(x)$$

→ Then the integrals are done only once

→ Afterwards only summation required to change PDF





Tabulate the convolution of the perturbative coefficients with the interpolation kernel

Software (fastNLO)

fastNLO page at HepForge: http://fastnlo.hepforge.org/

fastNLO

fast pQCD calculations for hadron-induced processes

Home | Documentation | Scenarios | Code | Interactive (maintenance) | Links

Choose fastNLO version

fastNLO Toolkit v2.3
Version 2.3
Instructions

fastNLO Reader v2.1
Version 2.1
Instructions

fastnlo_toolkit_2.3.1 (pre)releases	
pre-2411	Intermediate release, fixing some small issues, checked to work with LHAPDF-6.2.0, YODA-1.6.7, Rivet-2.5.4
pre-2402 ReleaseNotes ChangeLog	Public release, major new developments including extended table format, C++11 support in compiler is mandatory
pre-2212 ReleaseNotes ChangeLog	Public release, has producer for ROOT histograms and for statistical table evaluations
pre-2163 ReleaseNotes ChangeLog	Public release, can also provide PDF uncertainties via LHAPDF6
pre-2143 ReleaseNotes ChangeLog	Public release, works with Sherpa via to be released MCgrid package
pre-2087 ReleaseNotes ChangeLog	Public release, works with Sherpa via to be released MCgrid package
pre-1871 ReleaseNotes ChangeLog	Public release, fixes some little hiccups in optional parts and compiler warnings
pre-1854 ReleaseNotes ChangeLog	First public release, presented at DIS Warschau and CMS PDF Forum

Install two parts with: `configure, make, make install`

- **fastnlo_toolkit for interpolation grid creation and evaluation**
- **Interface package specific for desired theory code, e.g. fastnlo_interface_nlojet**
- **Latest release requires C++11 standard**
- **Alleviates using newest releases of LHAPDF, RIVET & YODA**

For APPLgrid see: http://applgrid.hepforge.org/

Theory interfaces

- Interface with NLOJet++ in use in H1, D0, CMS since a looong time!
- More recently fixed-order processes available through Sherpa & loop providers like BlackHat, OpenLoops, or NJET are interfaced through the MCgrid package (similar for APPLgrid)
- Since two years working with APPLgrid (M. Sutton, C. Gwenlan) & NNLOJET colleagues (T. Morgan, A. Huss, et al.) on common interface to the grid production
  - Approaching final results → further slides
- Within CMS contact KR for more details, software packages, and setup
- Further theory packages use either fastNLO or APPLgrid, e.g. BlackHat, DiffTop, Stripper (ttbar NNLO), MCFM, ...

NNLOJET+APPLfast Workflow

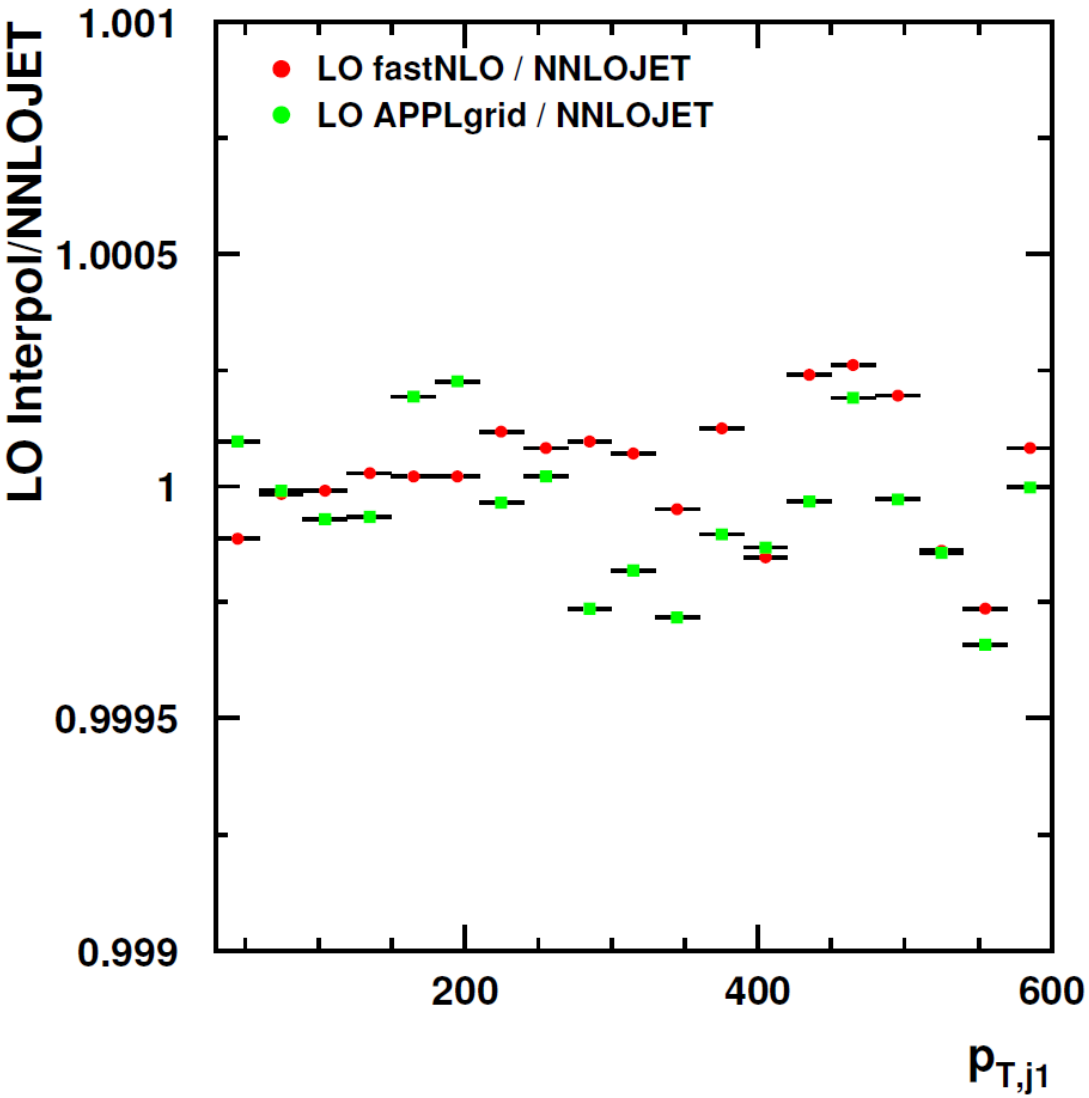
- **1. Preprocessing:** Check of interpolation quality
  - Short test jobs to check interpolation settings (& optimise if necessary) O(10 h)
- **2. NNLOJET Warm-up:** Vegas integration optimisation
  - 1 long (multi-core) job per process O(100 h)
- **3. APPLgrid/fastNLO Warm-up:** Adapt x- and scale-grids to accessed phase space (exact strategy differs between APPLgrid & fastNLO)
  - Only phase space provided from NNLOJET → significant speed-up O(100 h)
- **4. Interpolation grid production:**
  - Thousands of parallel jobs O(250 kh)
- **5. Postprocessing:** Statistical evaluation and combination of all produced grids …
  - Job to combine all grids and estimate statistical uncertainty O(100 h)
- **6. Present final results :-)**

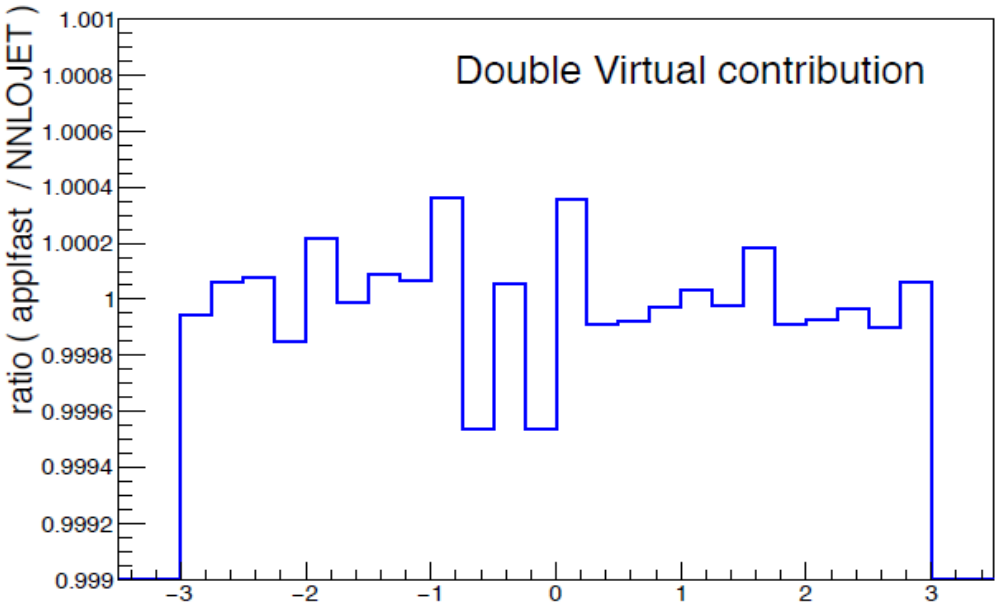
30 min :-)

Essentially the same steps needed with Sherpa+MCgrid

Step 1: Preprocessing

Z+jet LO Approximation Test
Similar performance at subpermille level





M. Sutton at QCD@LHC

Z+jet Test Setup: $p_{T,j1}$, η_{j1}, y_Z

- E_{cms} = 8 TeV
- $p_{T,jet}$ > 30 GeV
- $|y_{jet}| < 3$
- $|y_{l+,l-}| < 5$
- $80 < M_{l+l-} < 100$ GeV
- $\mu_r = \mu_f = M_Z$ = fixed

(→ no scale interpolation in this test)

Step 2: Vegas Integrations

- **NNLOJET Warm-up:**
  - **Must be one job per process type**
  - **Multi-threading possible**

Job Type	# Jobs	Threads / Job	Events / Job	Runtime / Job	Total Runtime
LO	1	16	32 M	0.35 h	0.35 h
NLO-R	1	16	16 M	1.0 h	1.0 h
NLO-V	1	16	16 M	1.0 h	1.0 h
NNLO-RRa	1	32	5 M	17.5 h	17.5 h
NNLO-RRb	1	32	5 M	20.7 h	20.7 h
NNLO-RV	1	16	8 M	22.4 h	22.4 h
NNLO-VV	1	16	8 M	24.6 h	24.6 h
Total	7	-	-	-	87.6 h

Step 3: Phase Space Exploration

- **APPLfast Warm-up:**
  - NNLOJET is run without CPU-time expensive weight calculation
  - At least 1 job per process needed to determine phase space limits individually
  - Grids created and optimised during warm-up (APPLgrid)
  - Grids created in production step from optimised x and Q-scale limits (fastNLO)
  - Warm-up can be parallelised, if necessary (fastNLO)
  - Presented table used for extensive testing; overkill for normal use

In this setup most x_{min} limits from LO runs, 3 from higher-order runs.

Job Type	# Jobs	Events / Job	Runtime / Job	# Events	Total Runtime
LO	5	500 M	12 h	2.5 G	60 h
NLO-R	5	300 M	18 h	1.5 G	90 h
NLO-V	5	500 M	13 h	2.5 G	65 h
NNLO-RRa	10	50 M	13 h	0.5 G	130 h
NNLO-RRb	10	50 M	15 h	0.5 G	150 h
NNLO-RV	5	300 M	19 h	1.5 G	90 h
NNLO-VV	5	500 M	12 h	2.5 G	60 h
Total	45	---	---	11.5 G	645 h

Step 4: Mass Production

- **NNLOJET + APPLfast**
  - **Massive parallelised computing on Virtual Machines with 24h lifetime**
  - **Example with fastNLO, APPLgrid example in progress**

Job Type	# Jobs	Events / Job	Runtime / Job	# Events	Total Output	Total Runtime
LO	10	140 M	20.6 h	1.4 G	24 MB	206 h
NLO-R	200	6 M	19.0 h	1.2 G	1.3 GB	3800 h
NLO-V	200	5 M	21.2 h	1.0 G	1.2 GB	4240 h
NNLO-RRa	5000	60 k	22.5 h	0.3 G	26 GB	112500 h
NNLO-RRb	5000	40 k	20.3 h	0.2 G	27 GB	101500 h
NNLO-RV	1000	200 k	19.8 h	0.2 G	6.4 GB	19800 h
NNLO-VV	300	4 M	20.5 h	1.2 G	2.0 GB	6150 h
Total	11710	---	---	5.5 G	64 GB	248196 h

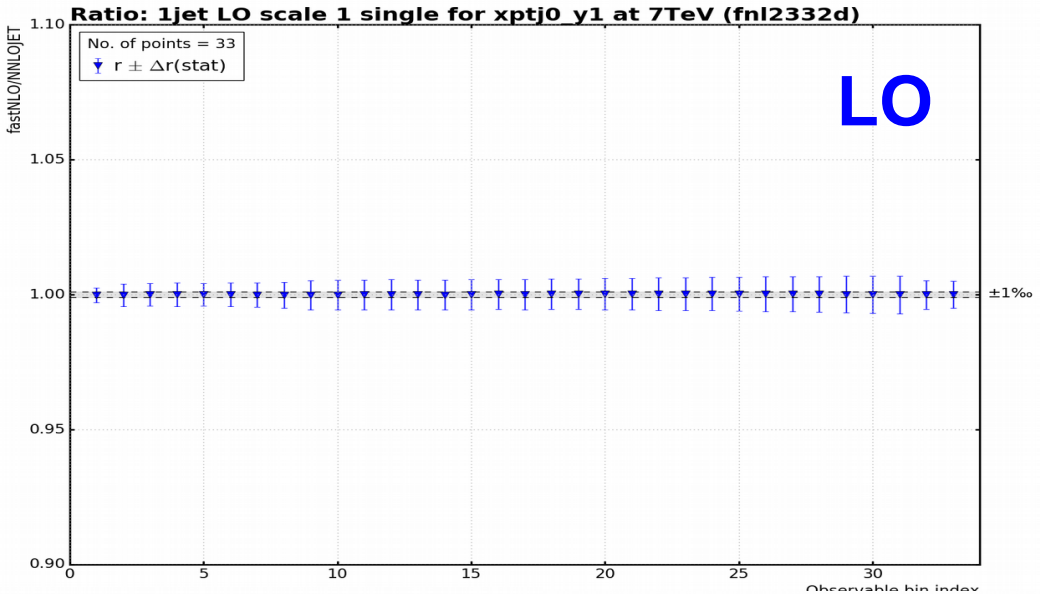
3 times 11710 grids/tables + all NNLOJET output!
Final 3 files for analysis are O(10 MB) each.

Step 5: Postprocessing

- **Checking, purging, combining:**
  - Check interpolation quality for individual grids
  - Run NNLOJET combination script → weight tables
  - Weighted merging of grids
  - Check and treat potential remaining unsuppressed fluctuations
  - Do some nice physics

Inclusive jet pT – single grid

Ratio FastNLO / NNLOJET

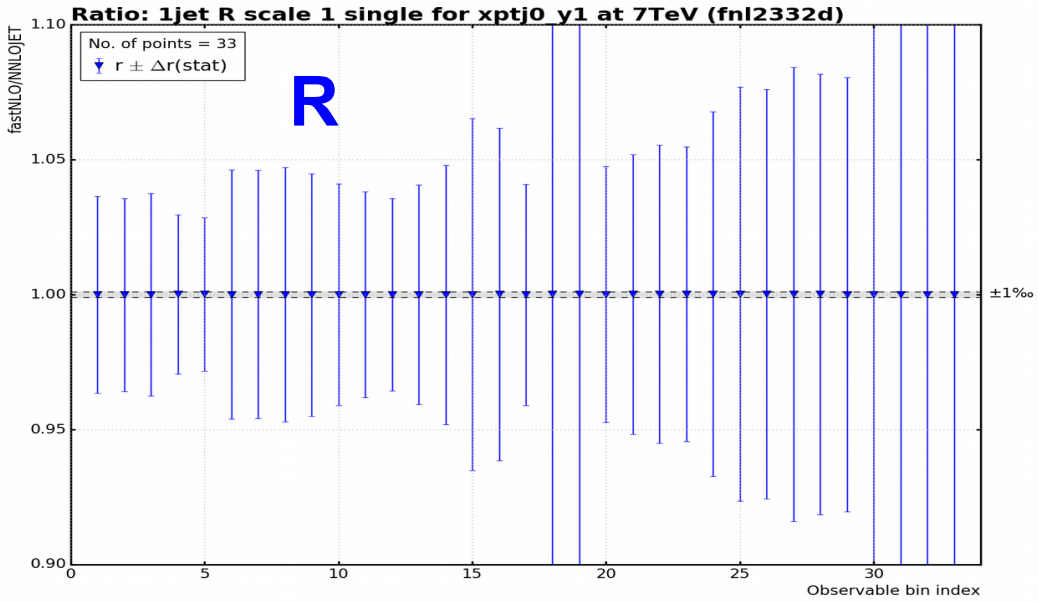


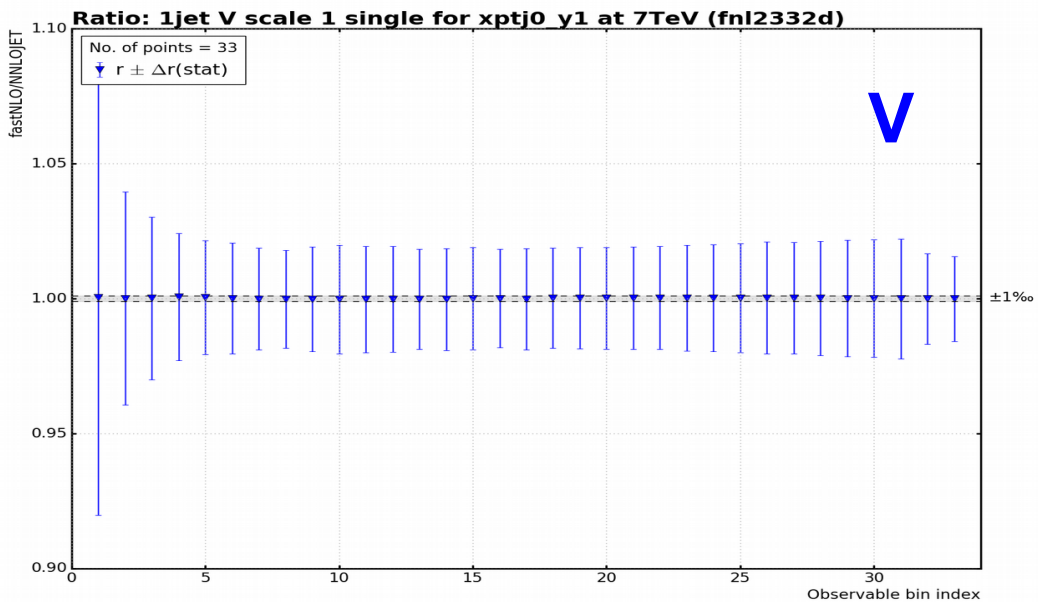
+10%

±1‰

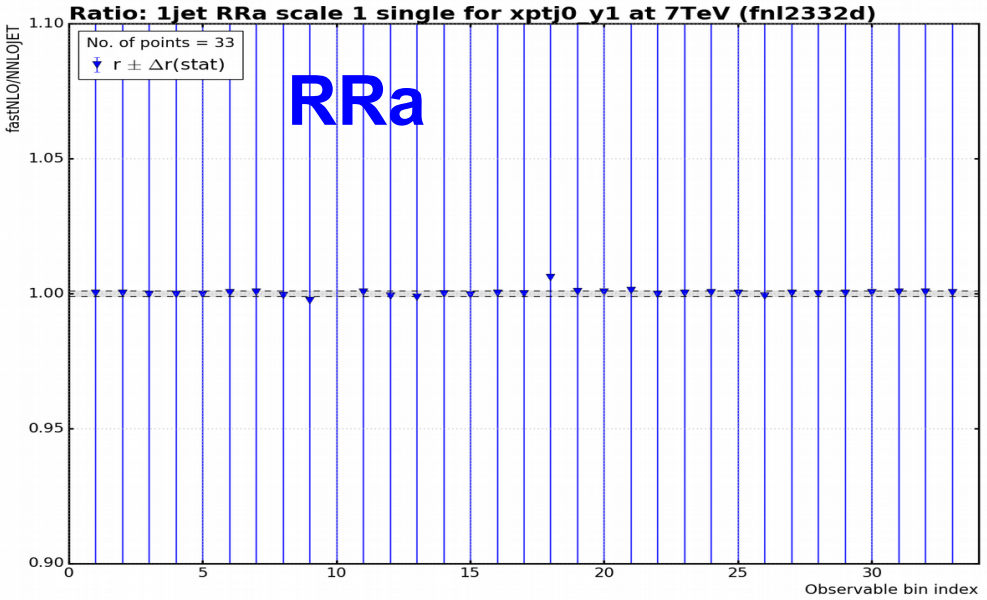
-10%

Error bars:
stat. uncertainty estimate from NNLOJET





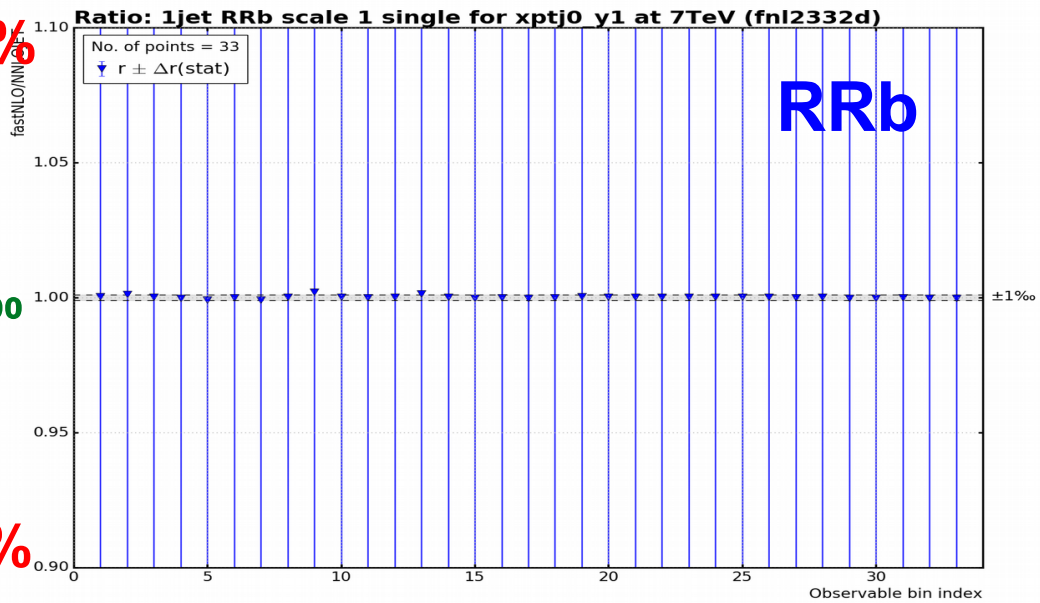
Inclusive jet pT – single grid

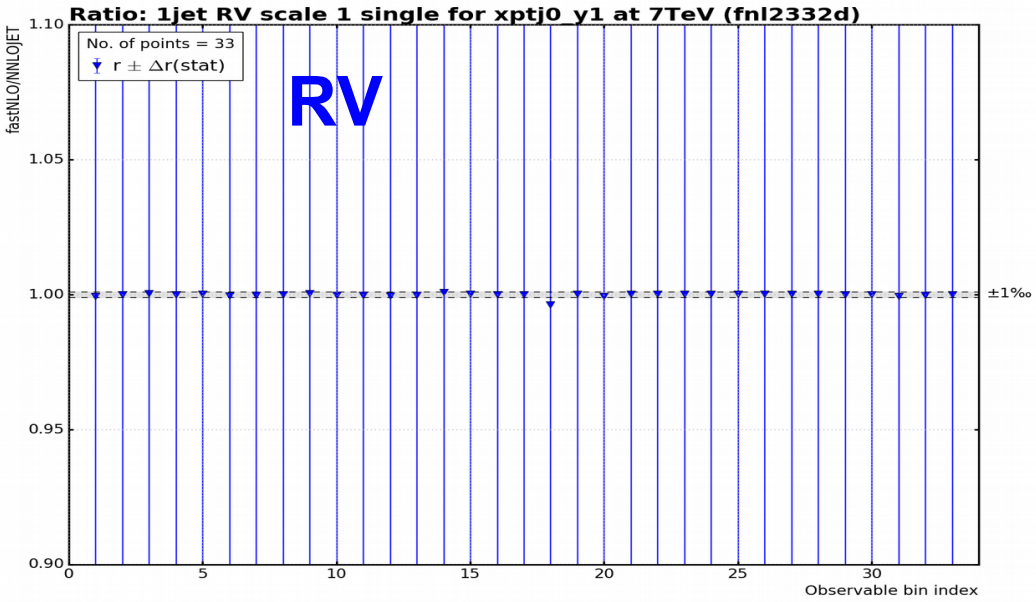


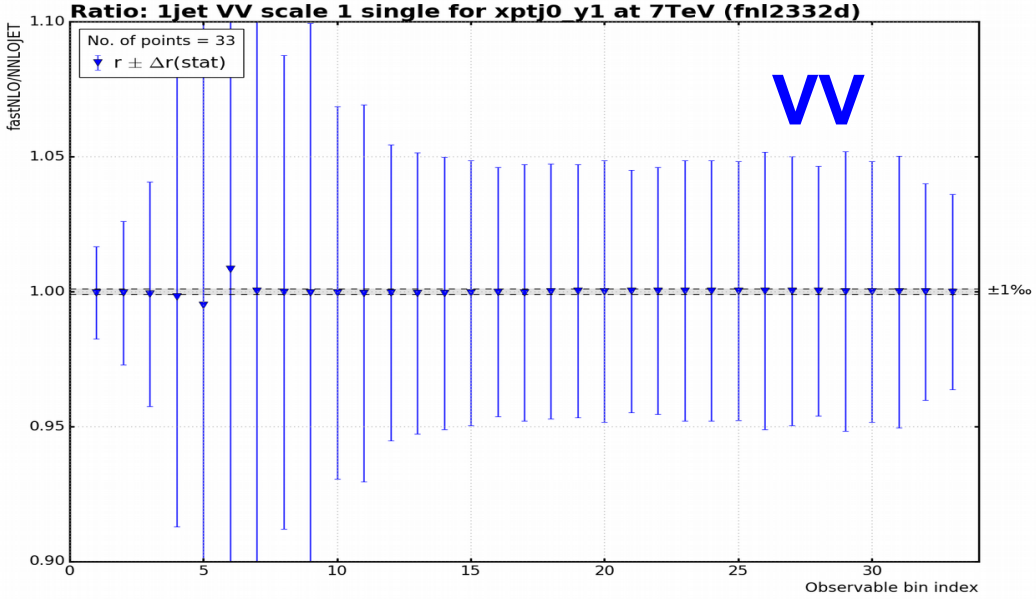
+10%

±1‰

-10%

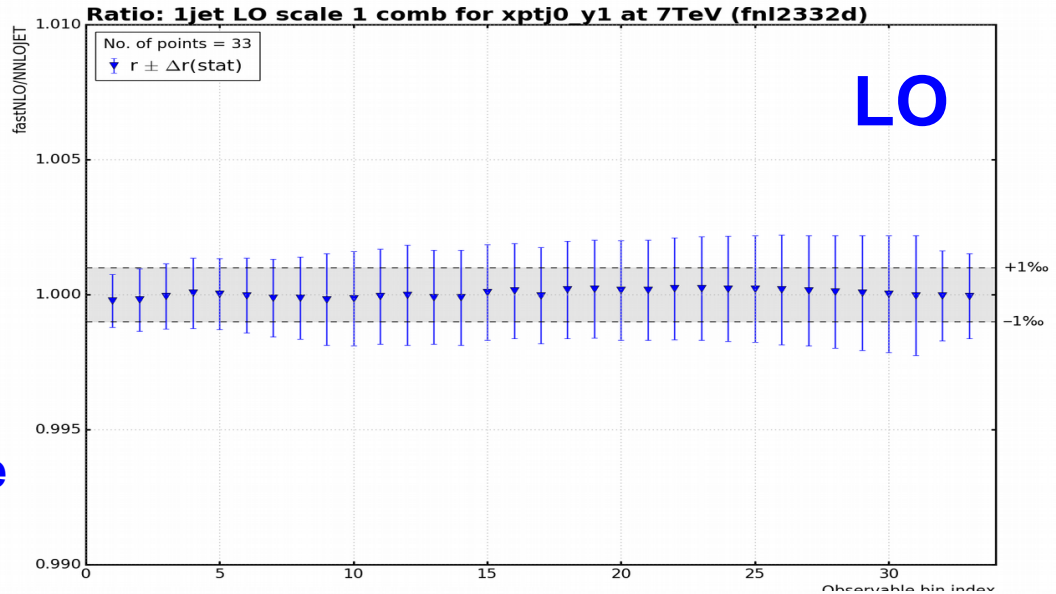






Inclusive jet pT – combined grid

Ratio FastNLO / NNLOJET

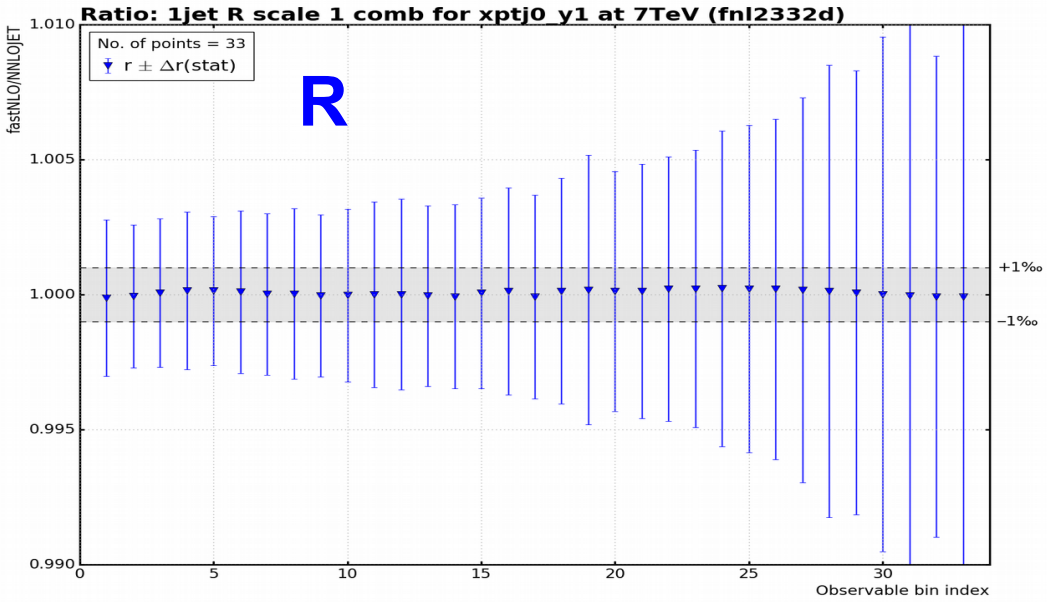


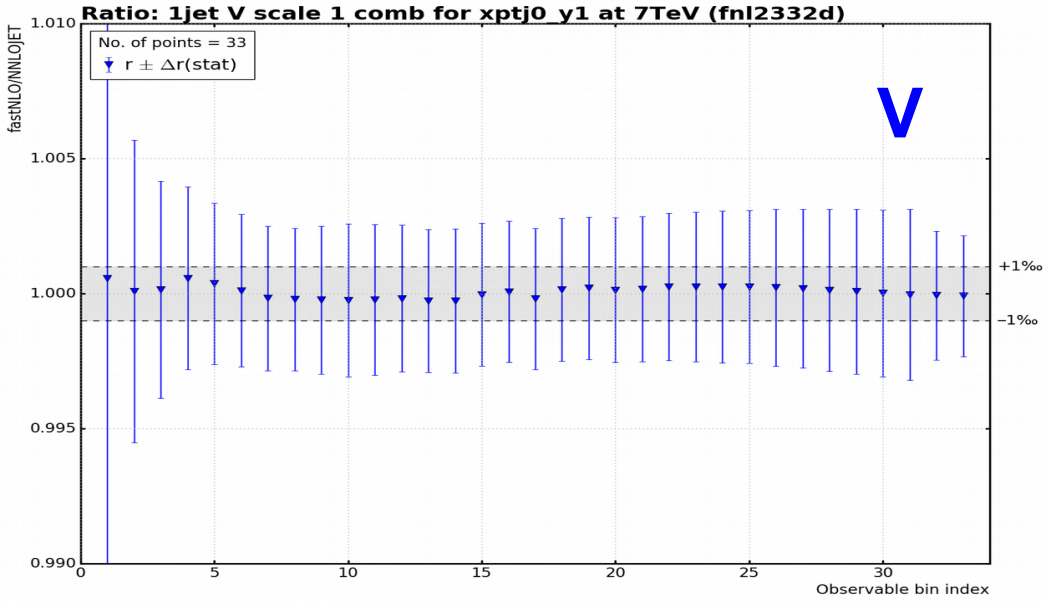
+1%

±1‰

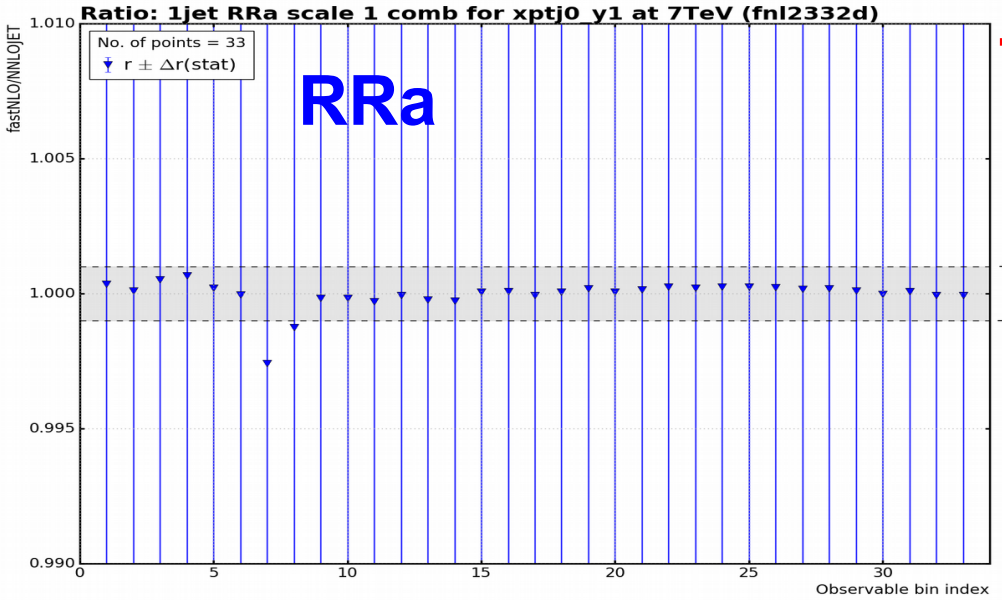
-1%

Error bars:
stat. uncertainty estimate from NNLOJET





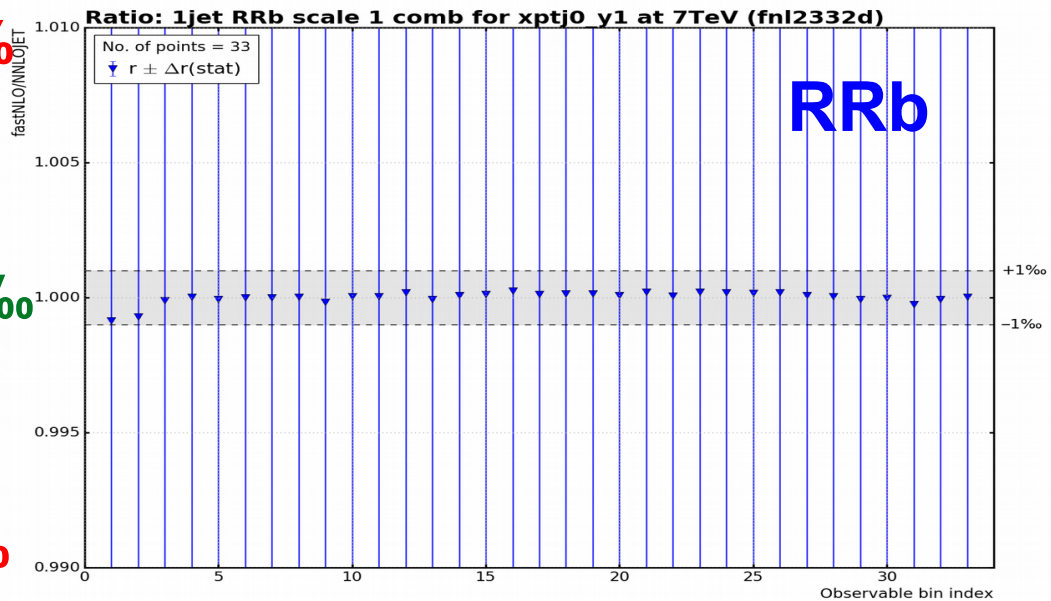
Inclusive jet pT – combined grid

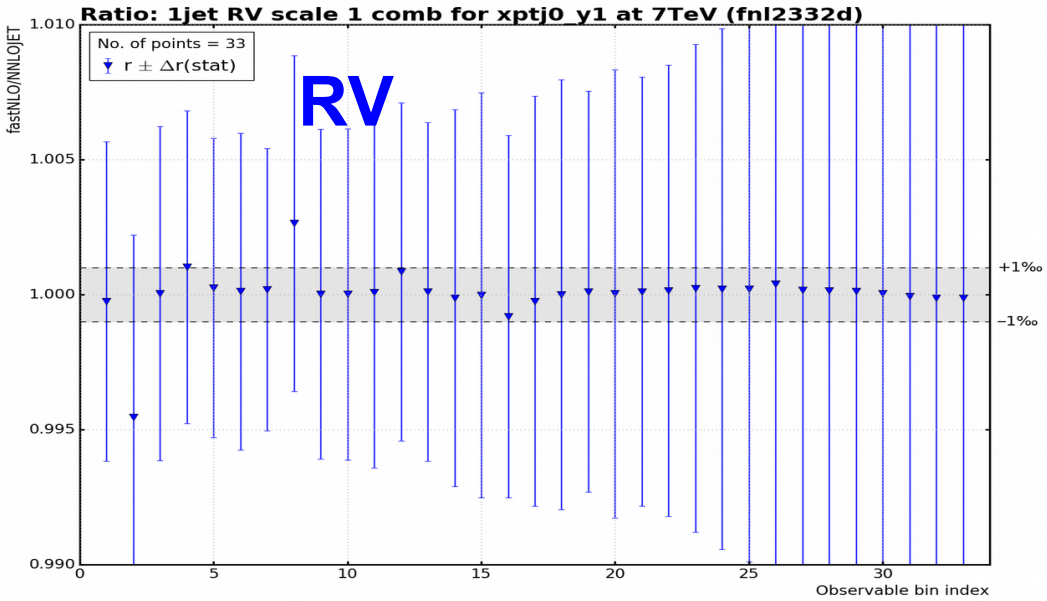


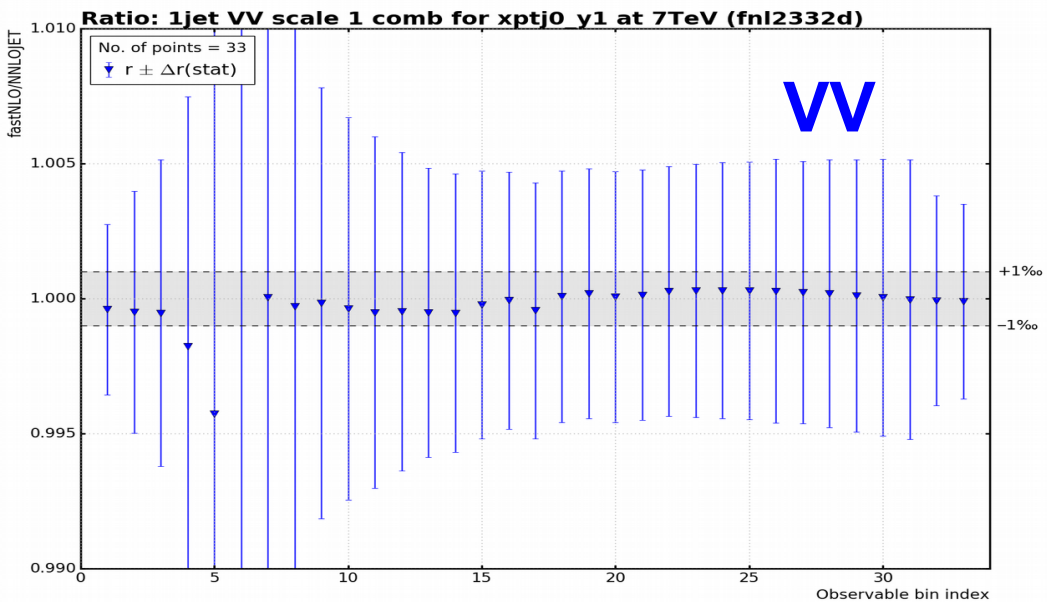
+1%

±1‰

-1%

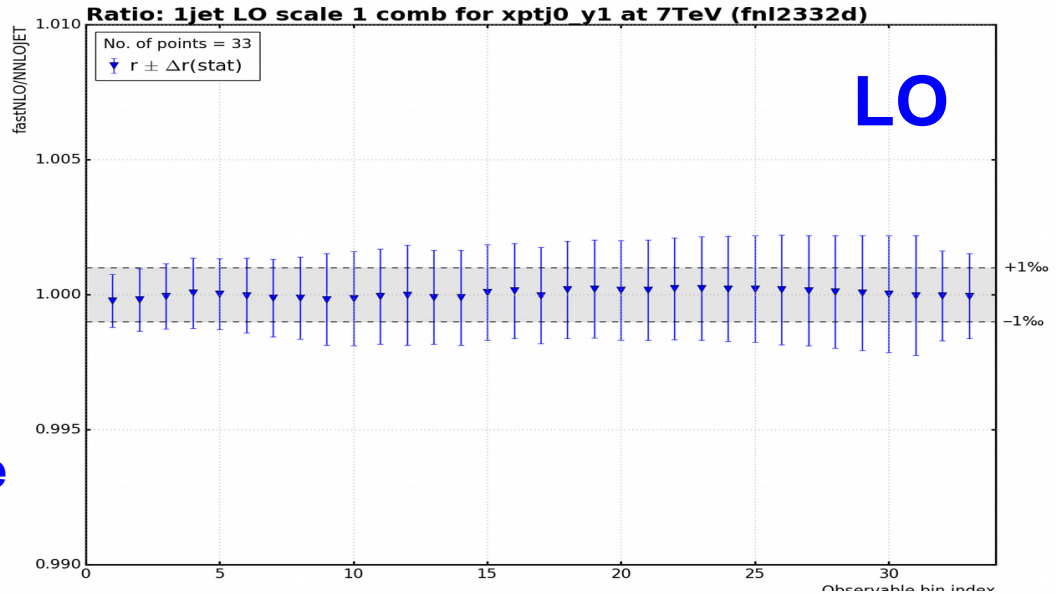






Inclusive jet pT – combined grid

Ratio FastNLO / NNLOJET

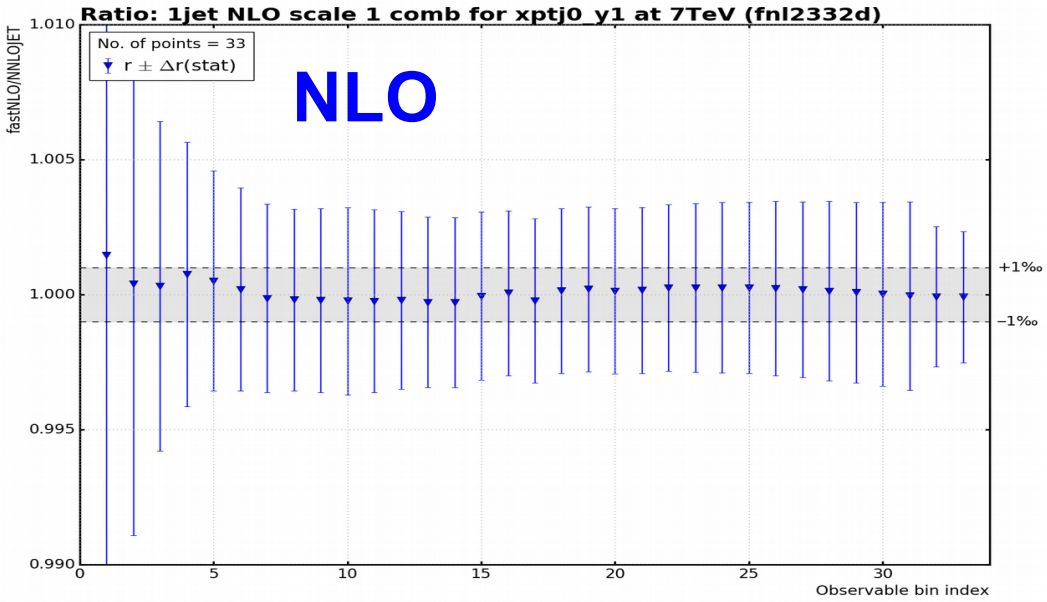


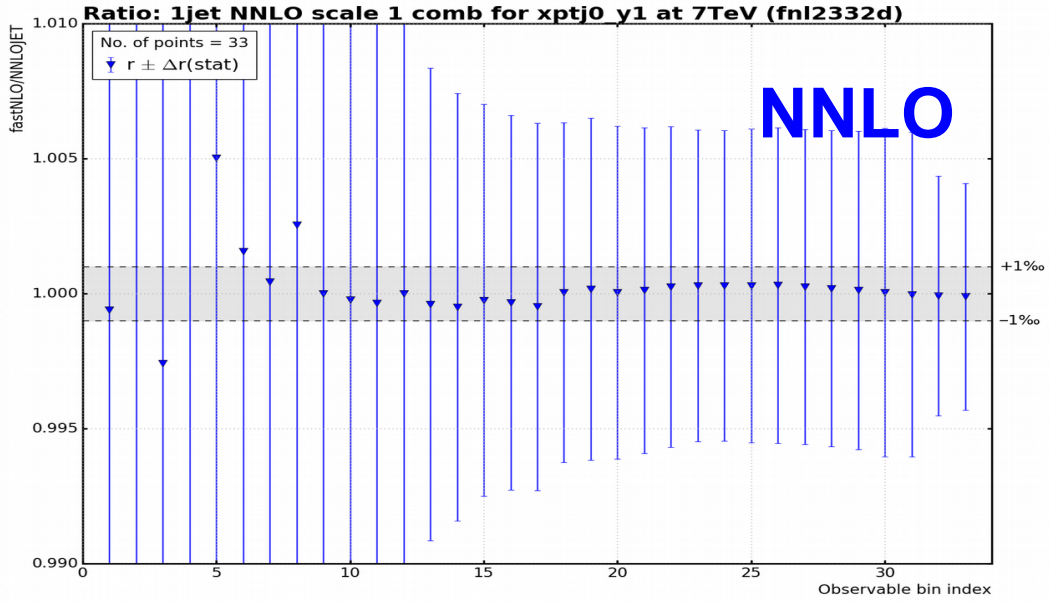
+1%

±1‰

-1%

Error bars:
stat. uncertainty estimate from NNLOJET





NNLO Outlook from March 2017

- NNLOJET provides NNLO in common interface for:
  - Z incl., Z+jet, W incl., pp jet+dijets, H incl., H+jet, DIS jet+dijets, e+e- 3jets
  - W+jet almost ready; more to come
- APPLgrid+fastNLO interface (NNLO-Bridge) is working
- Numerous adaptations implemented by all sides
- Large-scale productions tested for Z+jet and DIS jet
- Work in progress: Implementation of final combination procedure for interpolation grids
- Looking forward to many new NNLO interpolation grids in 2017

New NNLO Outlook

- NNLOJET provides NNLO in common interface for:
  - Z incl., Z+jet, W incl., pp jet+dijets, H incl., H+jet, DIS jet+dijets, e+e- 3jets
  - W+jet published; code to be released, see also previous talk by A. Huss!
- APPLgrid+fastNLO interface (NNLO-Bridge) is working
- Numerous adaptations implemented by all sides
- Large-scale productions tested for Z+jet and DIS jet and pp jet
- DIS NNLO results & α_s published with H1: Eur. Phys. J. C, 2017, 77, 791
- Received final combination prescription for NNLOJET results last Nov.
  - Removes fluctuations from bad cancellations
  - fastNLO table merging implemented; looks ok → check for outliers
  - Interpolated scale variations working except asymmetric $\mu_r \neq \mu_f$ TBD
- Looking forward to many new NNLO interpolation grids in 2018

Summary

- APPLfast interface (NNLO-Bridge) and interpolation is working
- Large-scale productions tested for Z+jet, DIS jet, and pp jets
- Combination of grids with weights a la NNLOJET implemented
- Address last issues and check on possible remaining outliers in grids even with weighted combination
- Start to produce a series of APPLgrid and/or fastNLO tables for various processes with publically available data
- Final grids will be made available via a common repository on HepForge, open for contributions from the community

Thank you for your attention!

Backup