APPLfast–NNLO

update on interpolation grids for NNLO

DIS17 workshop, Birmingham, UK

April 2017

A. Huss (ETH Zurich), T. Morgan (ex IPPP, Durham)

C. Gwenlan (Oxford), M. Sutton (Sussex)

D. Britzger (DESY), K. Rabbertz (KIT)

with support from IPPP associateship

introduction – the problem

- theoretical uncertainties (EG. proton PDFs, scales μ_R, μ_F) are dominant systematic on many important processes at the LHC (EG. searches for new, massive particles; Higgs production; as well as limiting precision on several fundamental quantities (α_s, M_W))
- thus need ever more sophisticated theoretical predictions (at least NLO; and state-of-the-art NNLO becoming increasingly available, and also necessary)
- many use cases require running the calculations many, many times
  EG. evaluation of theory uncertainties with different (eigenvector) PDFs, scales, α_s; and notably **QCD fits to determine PDFs, α_s**!
- BUT practical usage restricted by very long computational times (days, to weeks)

solution

- **method (in brief):**
- store perturbative coefficients of **(N)NLO QCD** calculations of final state observables in look-up tables (**grids**)
- calculation needs to be run in full just once, to store coefficients
- allows **a posteriori convolution** with any PDF of choice (a posteriori variation of α_s, scales μ_R, μ_F etc. also possible)
- finer details can of course be complex!

- **extensive applications,** ensuring most wide-spread practical use of state-of-the art calculations, EG.
- evaluation of theoretical uncertainties
- improved proton PDF determination; determined by comparing data to theory in iterative fits with the PDF changed in each iteration
- require computing cross sections **up to 100s of times** (prohibitive without fast, grid techniques)
- **convolution takes order milliseconds!**

more details of numerical technique in backups,
and in relevant publications and numerous talks by APPLgrid and fastNLO

two main packages on market:
APPLgrid and fastNLO

APPLfast, Eur Phys J C66 (2010) 503; fastNLO, arXiv:0609285 [hep-ph]

solution

- **method (in brief):**

APPLgrid, Carli et al EPJC66, 2010
FASTNLO, Kluge et al 2010
aMCfast, Berton et al , 2014
MCgrid, Del Debbio et al, 2014
APFELgrid, Bertone et al , 2016
APPLfast, 2017 (?)

- Great progress also in tools to interface NLO (NNLO) codes to PDF fitting code

(M. Ubiali, this conference)

complex!

- **extensive applications,** ensuring

- **convolution takes order milliseconds!**

more details of numerical technique in backups,
and in relevant publications and numerous talks by APPLgrid and fastNLO

two main packages on market:
APPLgrid and fastNLO

APPLfast, Eur Phys J C66 (2010) 503; fastNLO, arXiv:0609285 [hep-ph]

solution

- **method (in brief):**

APPLgrid, Carli et al EPJC66, 2010
FASTNLO, Kluge et al 2010
aMCfast, Berton et al , 2014
MCgrid, Del Debbio et al, 2014
APFELgrid, Bertone et al , 2016
APPLfast, 2017 (?)

(M. Ubiali, this conference)

- Great progress also in tools to interface NLO (NNLO) codes to PDF fitting code

this talk

- **extensive applications,** ensuring

complex!

- **convolution takes order milliseconds!**

more details of numerical technique in backups,
and in relevant publications and numerous talks by APPLgrid and fastNLO

two main packages on market:
APPLgrid and fastNLO

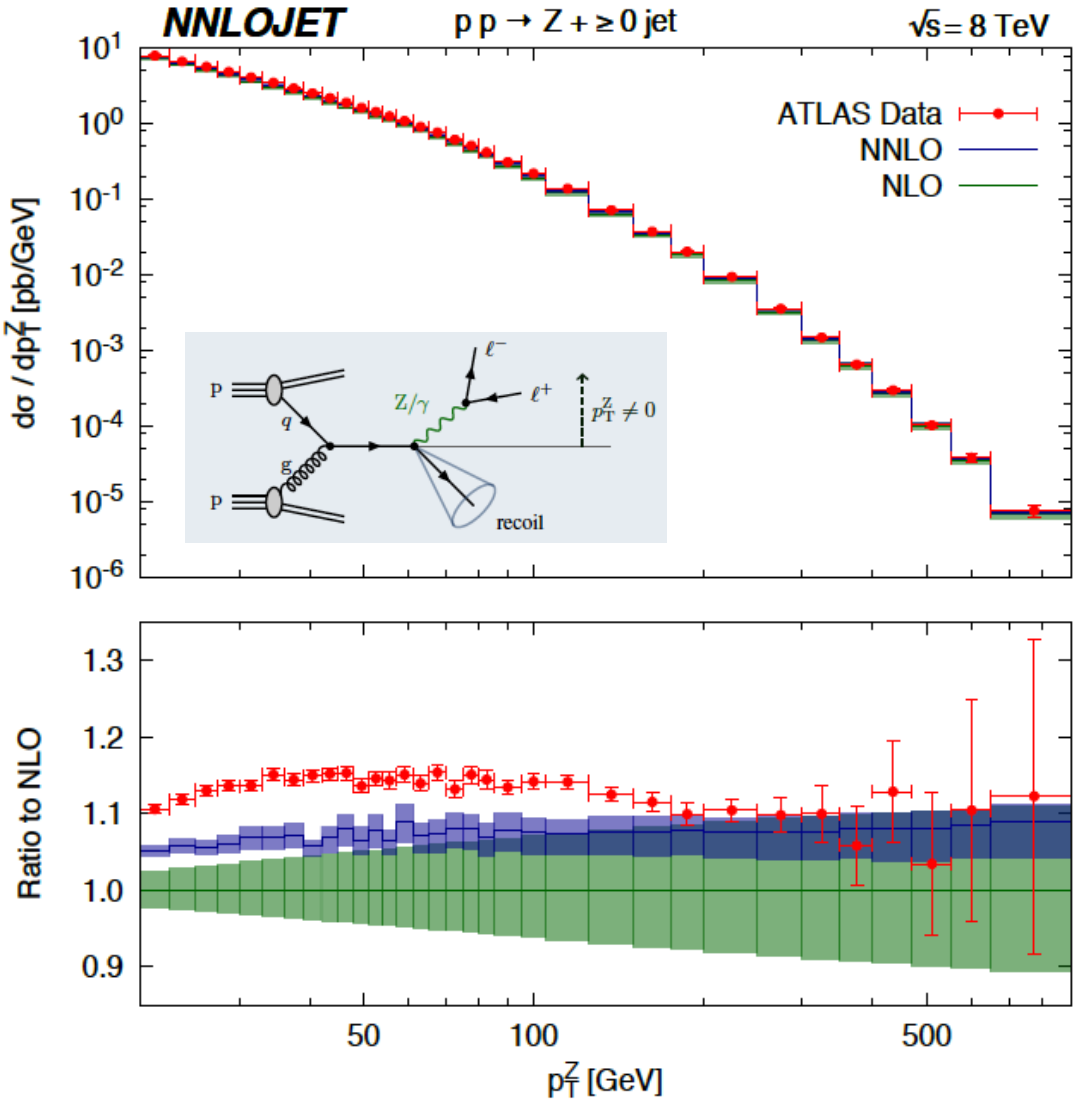
APPLfast, Eur Phys J C66 (2010) 503; fastNLO, arXiv:0609285 [hep-ph]

NNLOJET and APPLfast

- **what is NNLOJET?**
- semi-automated calculation of cross sections at NNLO QCD from IPPP, Zurich, ETH and others

- **what is APPLfast-NNLO?**
- single, combined interface for **NNLOJET** with both **APPLgrid** and **fastNLO** (collaboration between developers from all three packages)

- many processes available in **NNLOJET**; APPLfast is generic interface for **all available** processes



E.G. Z+Jet at NNLO (used for initial APPLfast proof-of-concept)

A. Gehrmann-De Ridder et al., JHEP 1607 (2016) 133

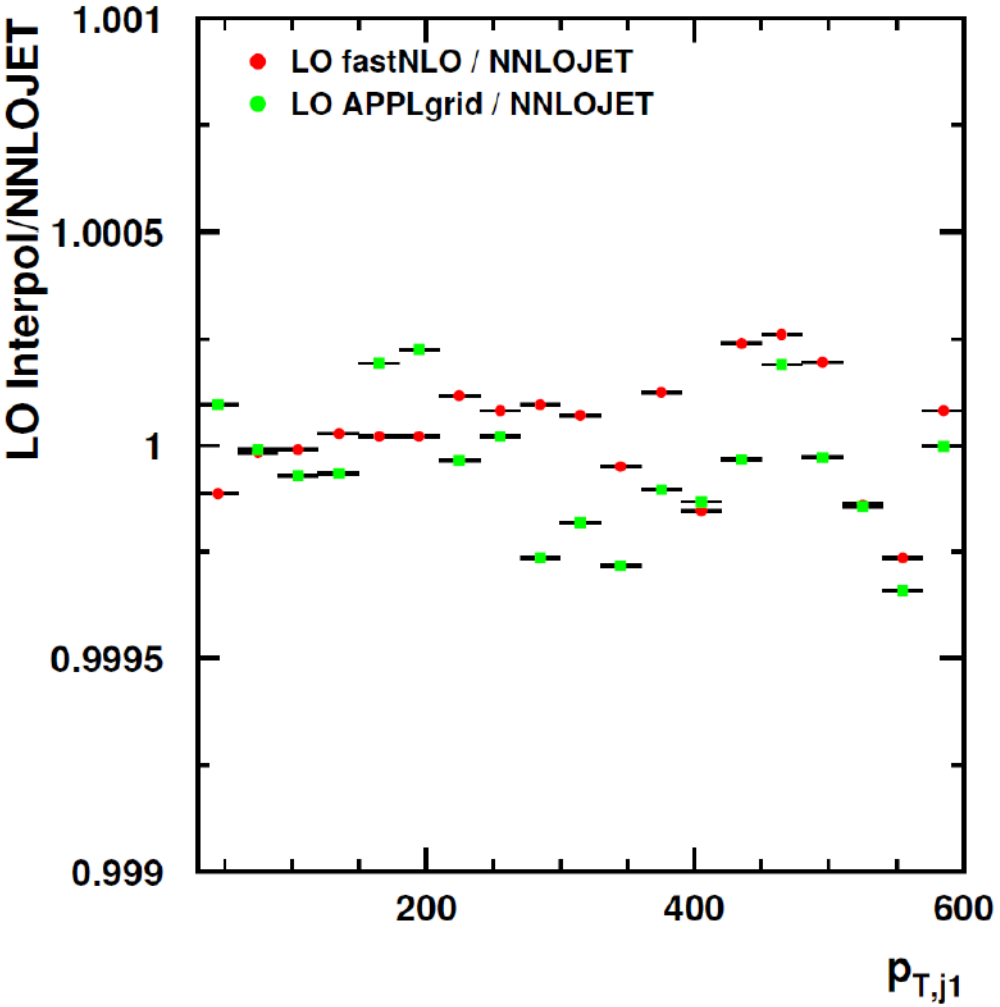
APPLfast grid generation (workflow)

1. **pre-processing:** establish details of parameterisation (EG. number of grid nodes, interpolation order etc.) using limited statistics jobs
2. **NNLOJET warmup:** optimise the NNLOJET VEGAS phase space in dedicated NNLOJET job [1 long (multicore) job per process]
3. **NNLOJET – APPLfast warmup:** run with grid filling enabled to establish optimised phase space (exact strategy differs between APPLgrid and fastNLO, but this is hidden in interface); only phase space provided by NNLOJET (significant speed up in doing this – implemented by NNLOJET authors)
4. **production run:** many jobs in parallel; many thousands of CPU hours required, most notably for the double-real contribution due to the highly differential phase space
5. **post-processing:** combine output sub-grids from production run

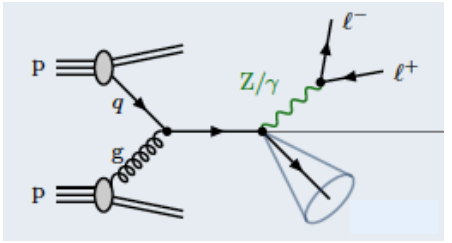
all above stages are implemented, but still being fully validated

step 1: pre-processing

Z+Jet LO test: proof-of-concept validation



(K. Rabbertz, PDF4LHC, March 2017)



Z+Jet test setup:

E_{CM} = 8TeV

Pt_{jet} > 30 GeV

$|y_{jet}| < 3$

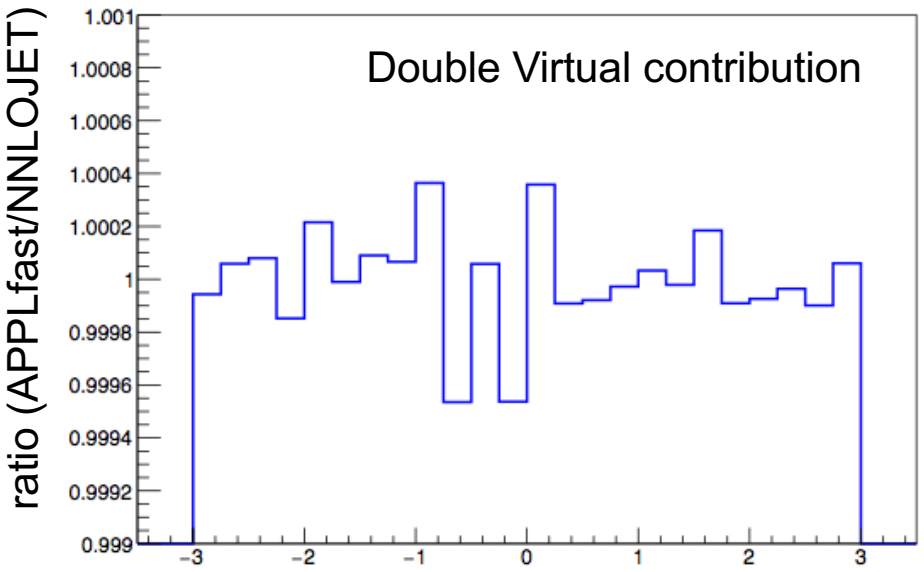
$|y_{ll}| < 5$

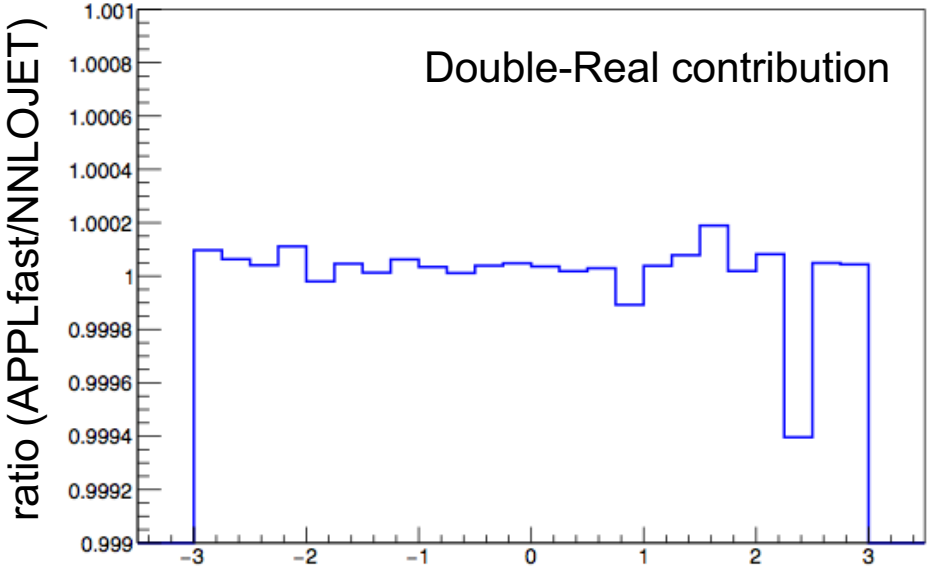
$80 < M_{ll} < 100$ GeV

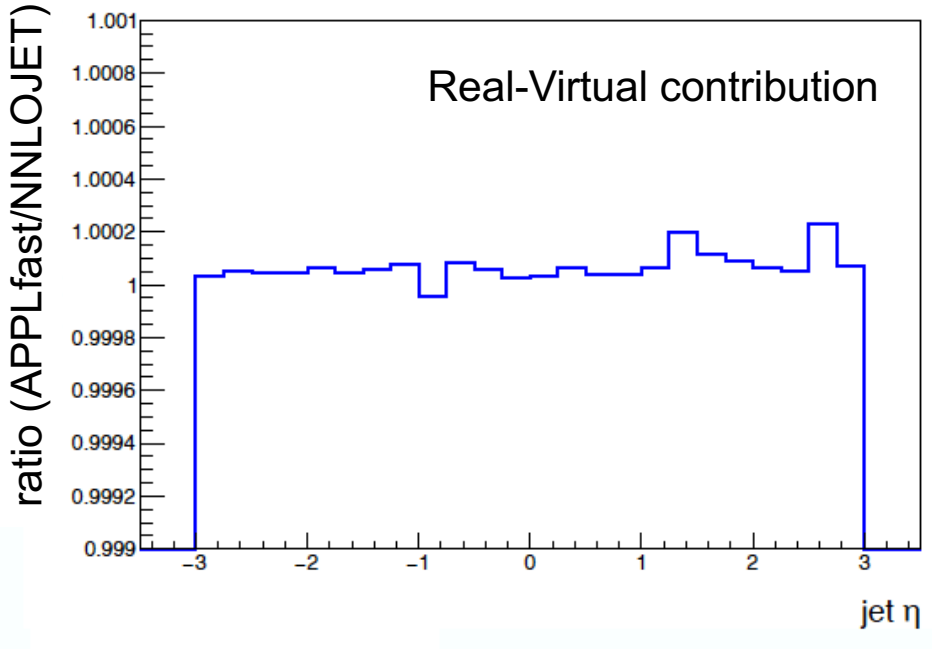
$\mu_R = \mu_F = M_Z$ = fixed

note y-axis range; sub-permille agreement reached at LO, NLO and NNLO in validation jobs

step 1: pre-processing







NNLO Z+Jet

note y-axis range; large excursions in some bins due to limited statistics used for determination of optimised phase space

(early proof-of-concept presented first at QCD@LHC 2016, M. Sutton; (APPLgrid example))

step 2: initial NNLOJET VEGAS warmup

- **NNLOJET warmup – no grid generation**
- need **one job** per cross section contribution type, EG. LO, R, V, RR(a,b), RV, VV
- internal NNLOJET 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

(K. Rabbertz, using fastNLO grid generation on the BwUniCluster at KIT, thanks to Baden-Württemberg High Performance Computing (HPC) support)

step 3: grid phase space optimisation

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

- **NNLOJET run with only phase space filling calculated**; CPU-expensive weight calculation not executed
- at least one job per process needed, EG. LO, NLO-R, NLO-V etc.; jobs can be run in parallel
- reasonably fast; non-trivial numbers, but nowhere near that needed for full production runs

step 4: mass production

job type	# jobs	events/job	runtime/job	# events	total output	total runtime
LO	10	140 M	20.6 h	1.4G	24 MB	206 h
NLO-R	200	6 M	19.0 h	1.2G	1.3 GB	3800 h
NLO-V	200	5 M	21.2 h	1.0G	1.2 GB	4240 h
NNLO-RRa	5000	60 M	22.5 h	0.3G	26 GB	112500 h
NNLO-RRb	5000	40 M	20.3 h	0.2G	27 GB	101500 h
NNLO-RV	1000	200 M	19.8 h	0.2G	6.4 GB	19800 h
NNLO-VV	300	4 M	20.5 h	1.2G	2.0 GB	6150 h
total	11710	–	–	5.5G	64 GB	248196 h

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

- **NNLOJET + APPLfast**
- massive parallelised computing on virtual machine farm with 24h lifetime

(running on BwForCluster NEMO in Freiburg,
thanks to Baden-Württemberg High Performance Computing (HPC) support)

step 4: mass production

job type	# jobs	events/job	runtime/job	# events		
LO	10	140 M	20.6 h			
NLO-R	200	6 M				3800 h
NLO-V	200	5 M			.2 GB	4240 h
NNLO-RRa	5000				26 GB	112500 h
NNLO-RRb				0.2G	27 GB	101500 h
NN			19.8 h	0.2G	6.4 GB	19800 h
			20.5 h	1.2G	2.0 GB	6150 h
	10	–	–	5.5G	64 GB	248196 h

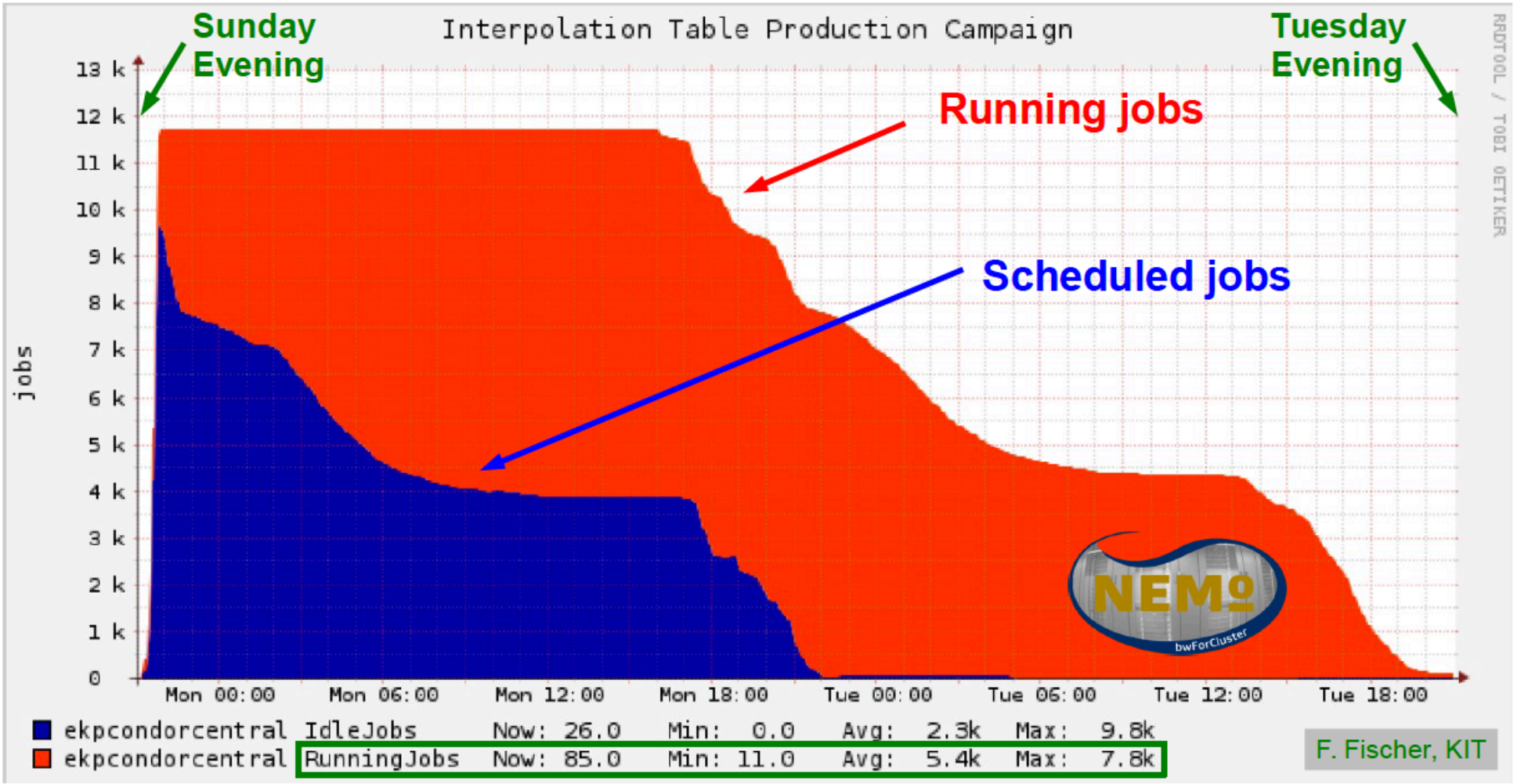
roughly, **factor of two** penalty for grid production versus NNLOJET alone (depends in detail on physics process and grid setup) – **still gain 100k's of CPU hours for each avoided full run of calculation**

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

- **NNLOJET + APPLfast**
- massive parallelised computing on virtual machine farm with 24h lifetime

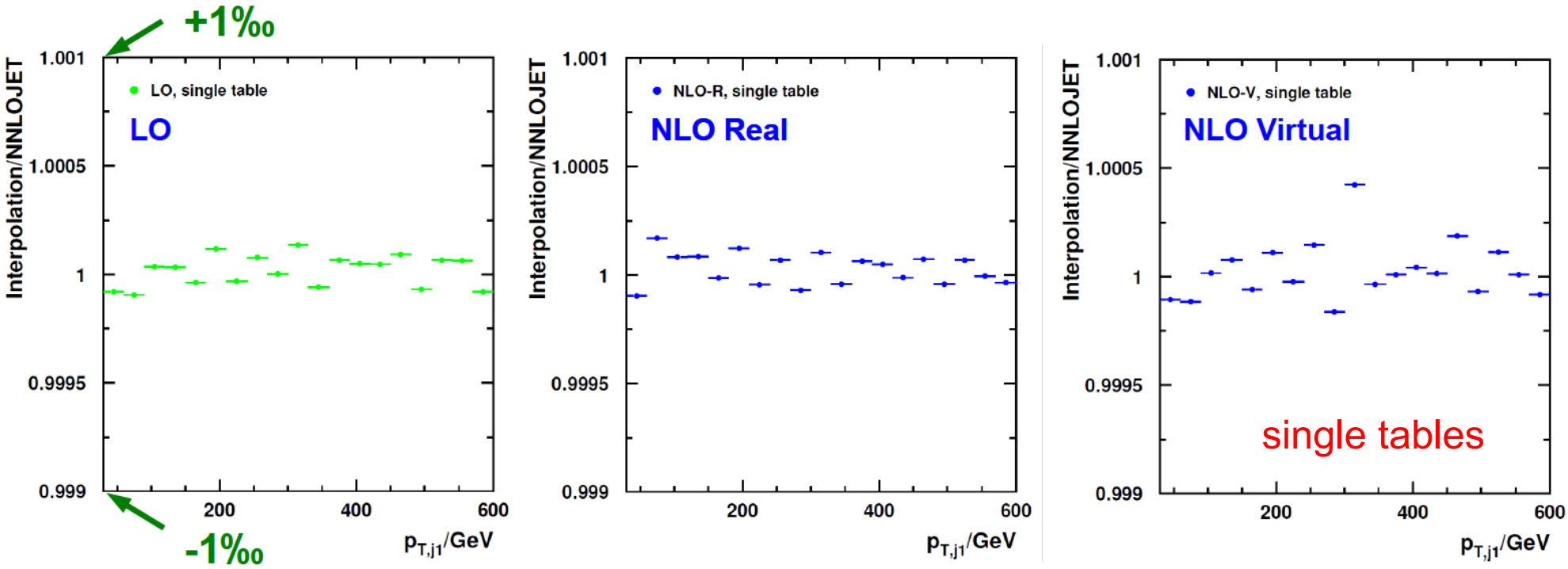
(running on BwForCluster NEMO in Freiburg,
thanks to Baden-Württemberg High Performance Computing (HPC) support)

production campaign



(K. Rabbertz, PDF4LHC, March 2017)

LO and NLO closure tests

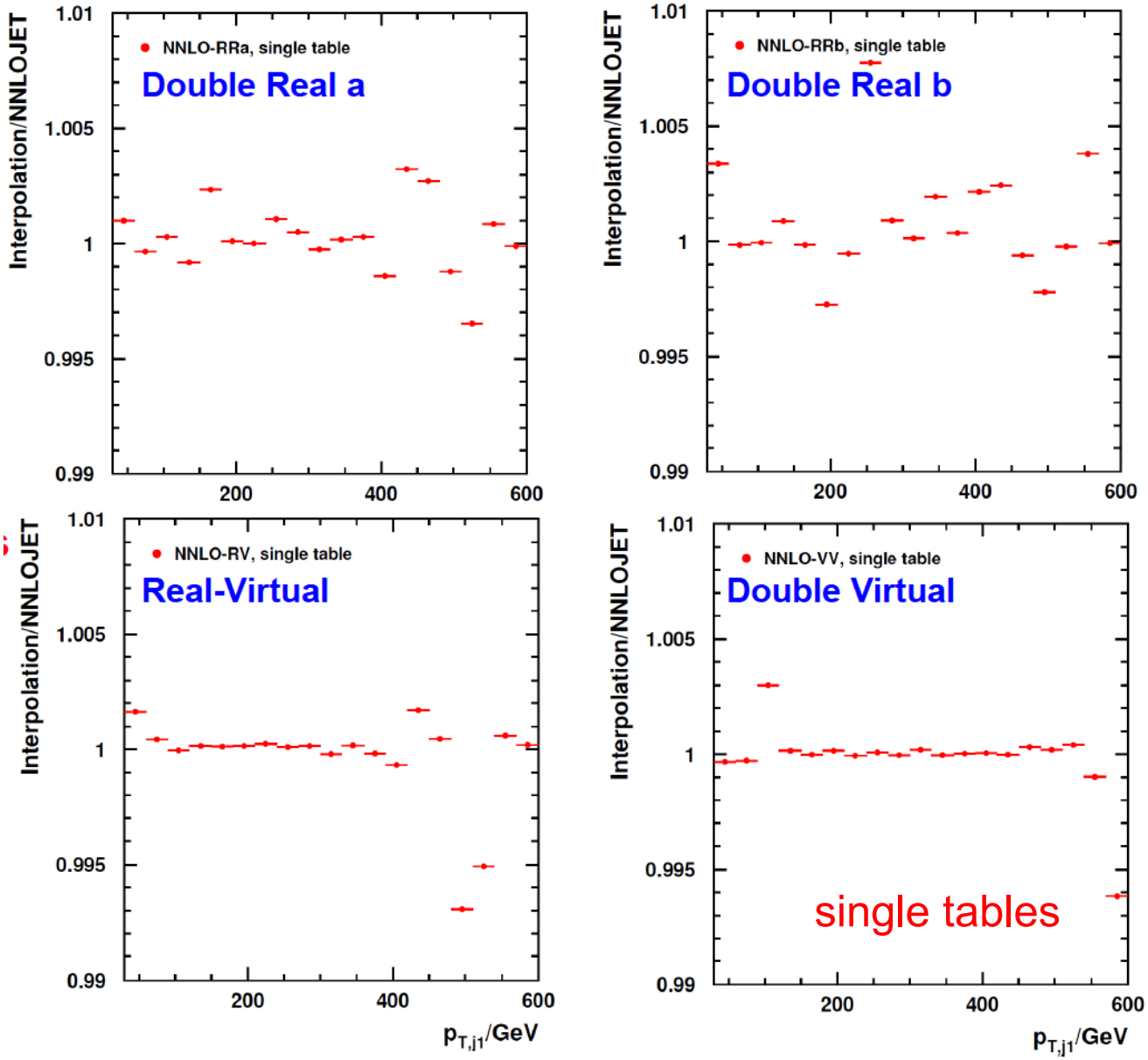


- **check closure of individual grids/tables from each job separately**
- general agreement to much better than per mille level
- at this level, interpolation of LHAPDF may be limiting factor

NNLO closure

- **agreement to sub percent level**
- some impact of fluctuations visible; to be dealt with in global procedure to combine grids for final cross sections

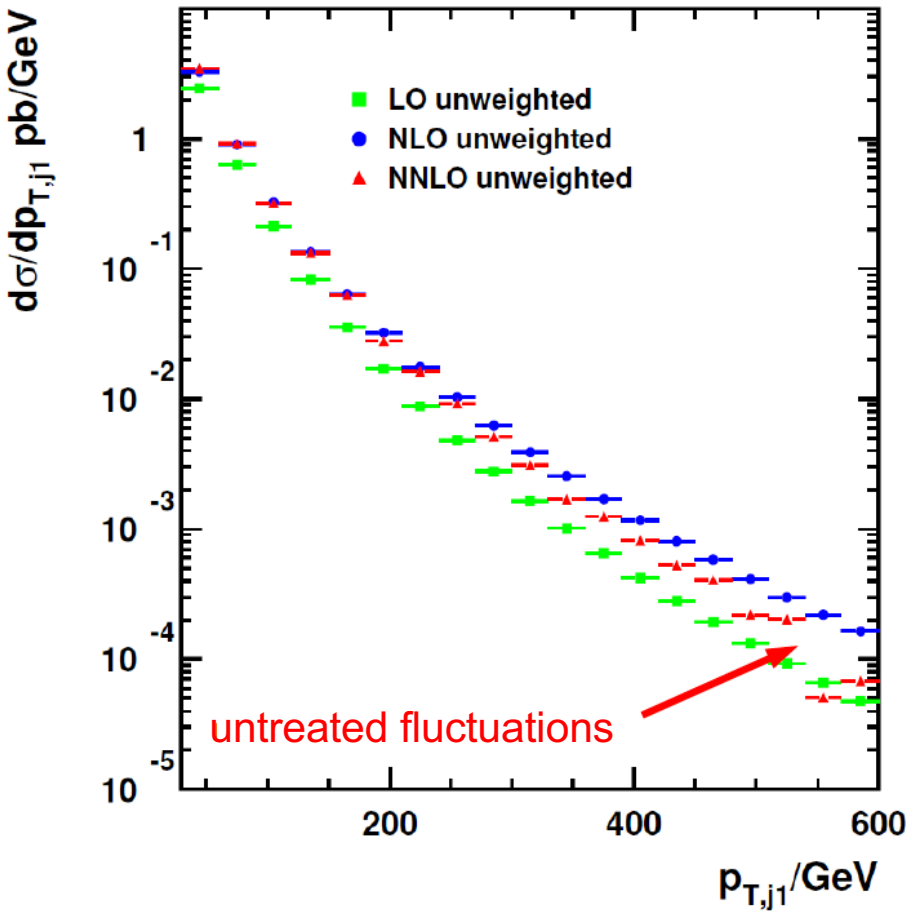
(a/b indicates a technical phase space separation for RR)

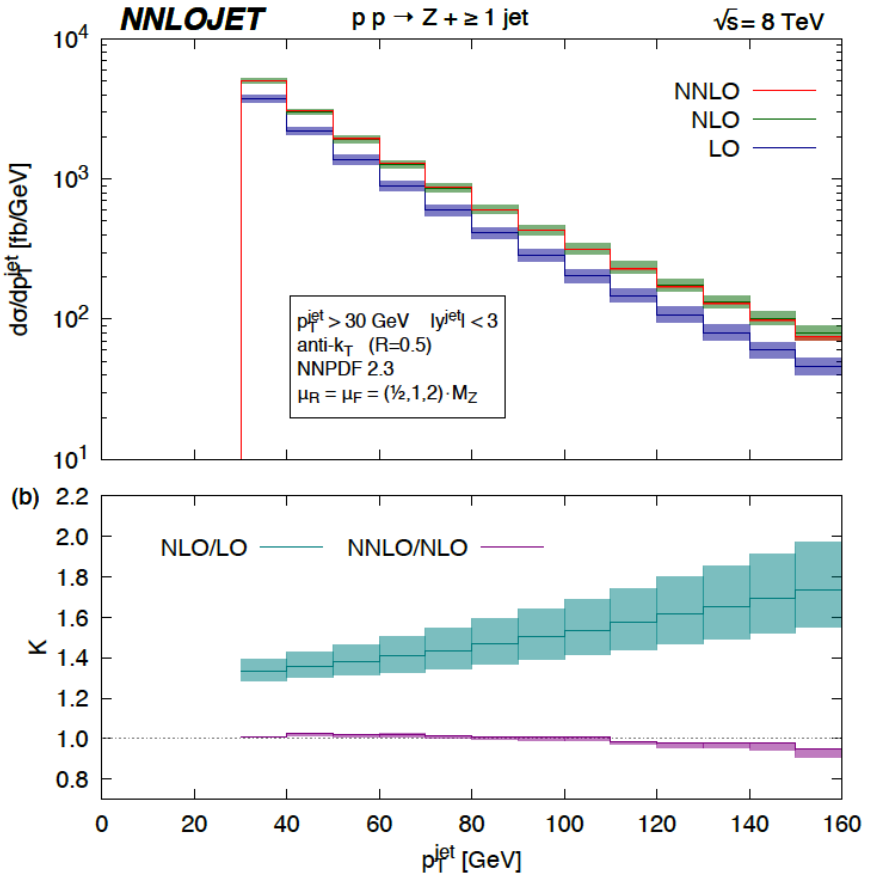


unweighted combination

- cross sections from naïve, unweighted combination (not final)
  - order of magnitude OK
  - fluctuations still to be dealt with in detailed weighted combination method (code exists, and procedure being validated for APPLfast grids)

(PRL, 117 (2016) 022001)

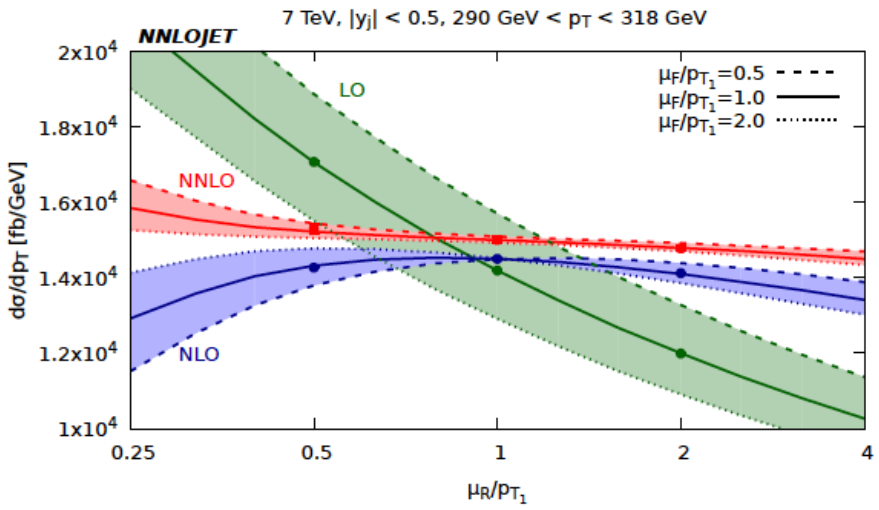


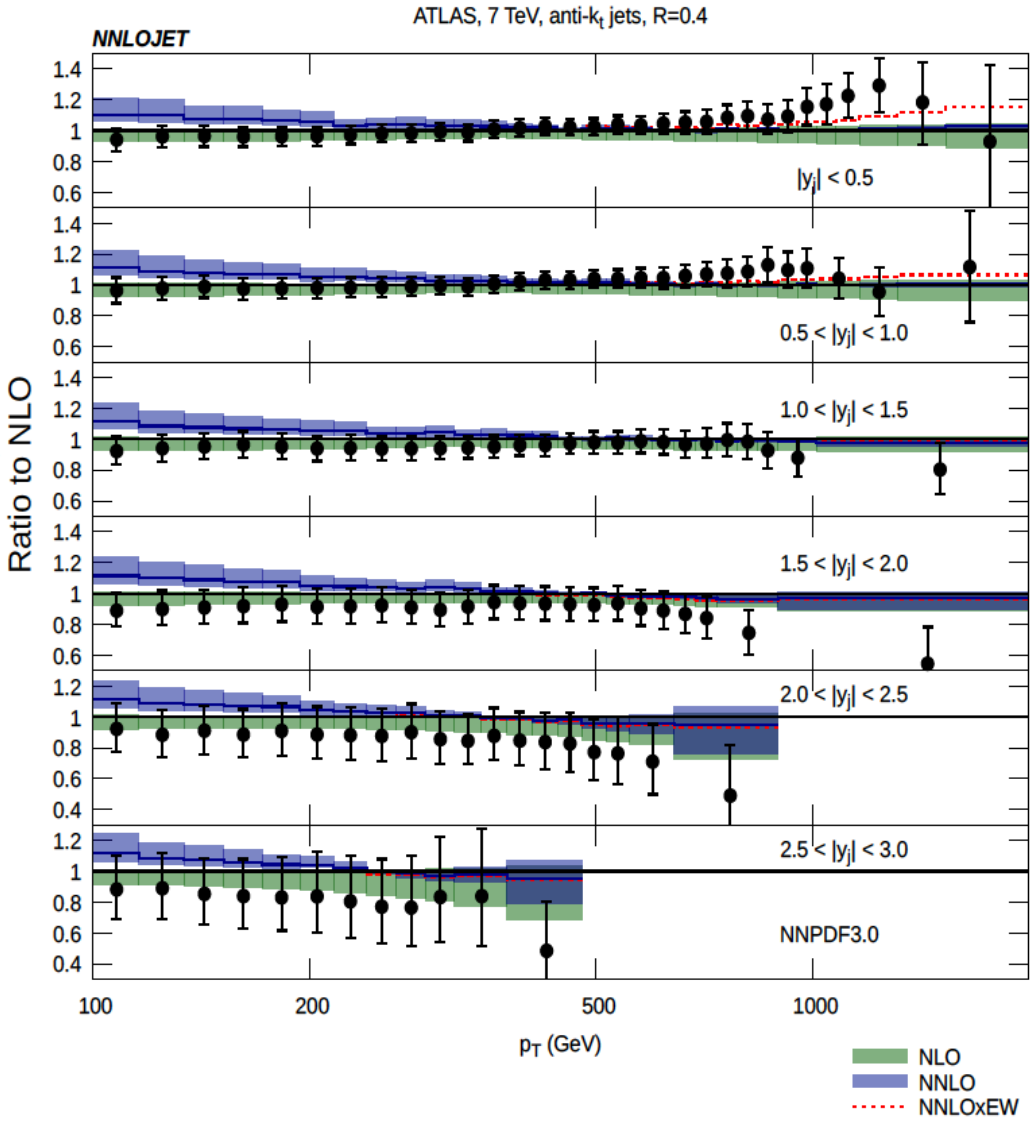


other processes

- **APPLfast interface is generic – in principle, works for any process available in NNLOJET!**
- currently: inclusive W, Z; Z+Jet; H, H+Jet; QCD jets in pp and ep, multijets in e+e-
- **of course, still need to check each process individually**
- main validation effort and performance testing in Z+Jet so far, but other processes being studied / developed
- EG. so far, effort devoted to:
- **QCD inclusive jets (pp)**
- **DIS jets (ep)** – see talk by D. Britzger, this conference
- inclusive Z

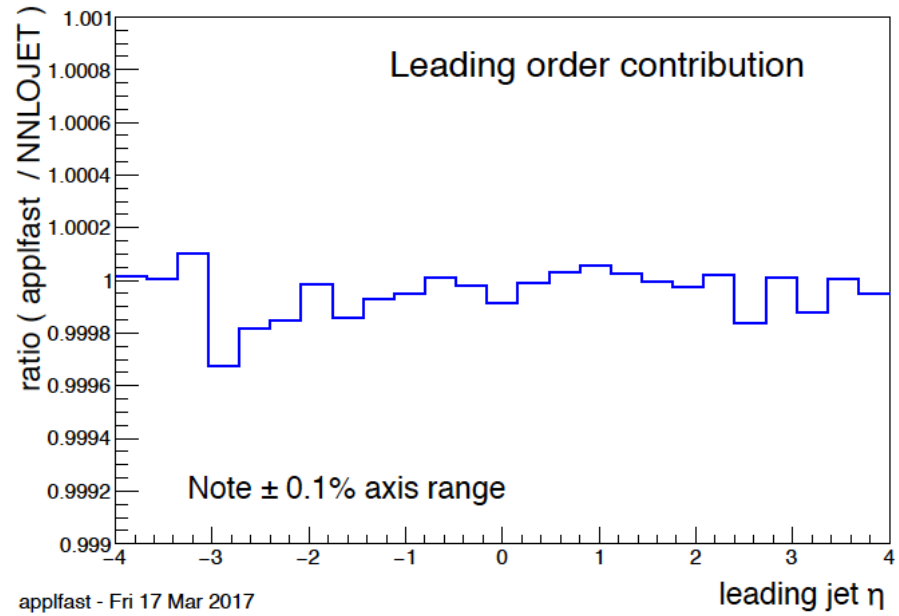
(pp) inclusive jet production

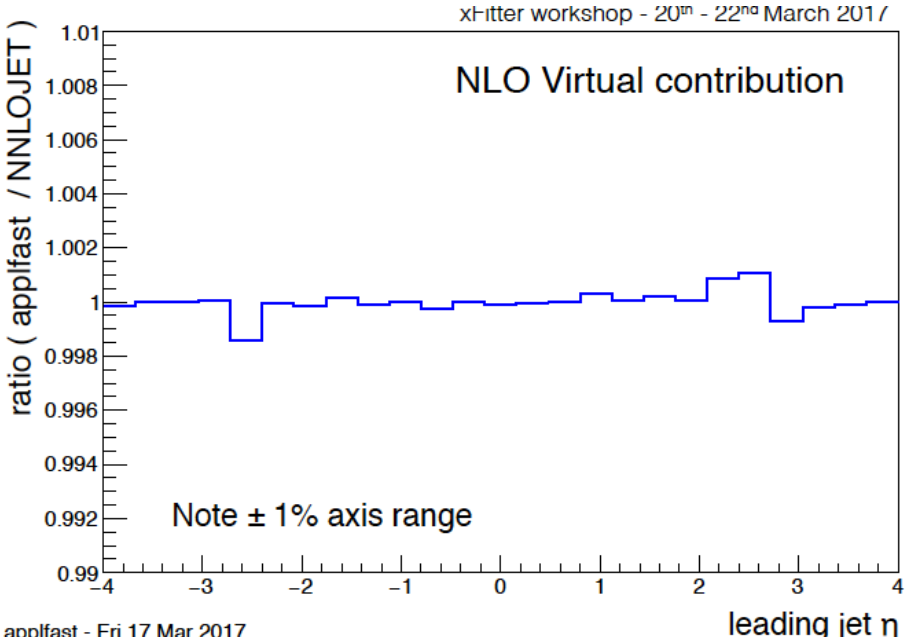




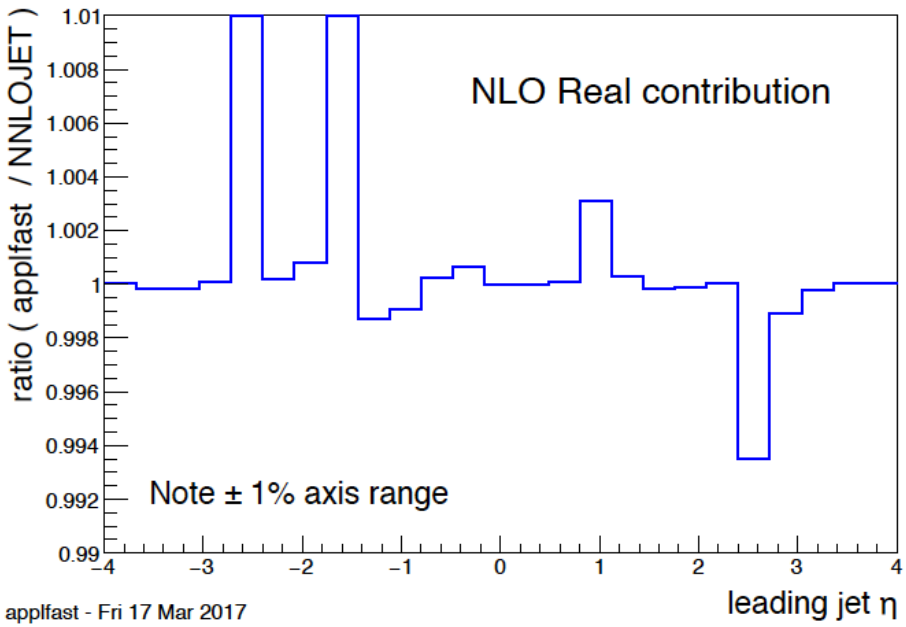
- validation tests have also been run for recently available NNLO QCD inclusive jets calculation (J. Currie, E.W.N. Glover, J. Pires, arXiv:1611.01460)
- for initial studies, scale choice: $\mu_R = \mu_F = pt_{lead}$

(pp) inclusive jets up to NLO



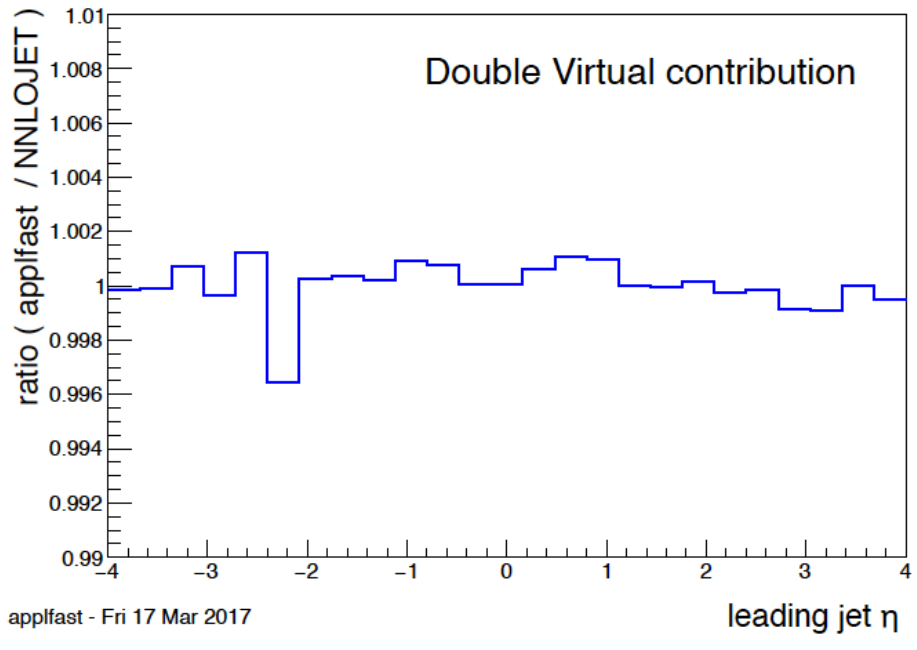


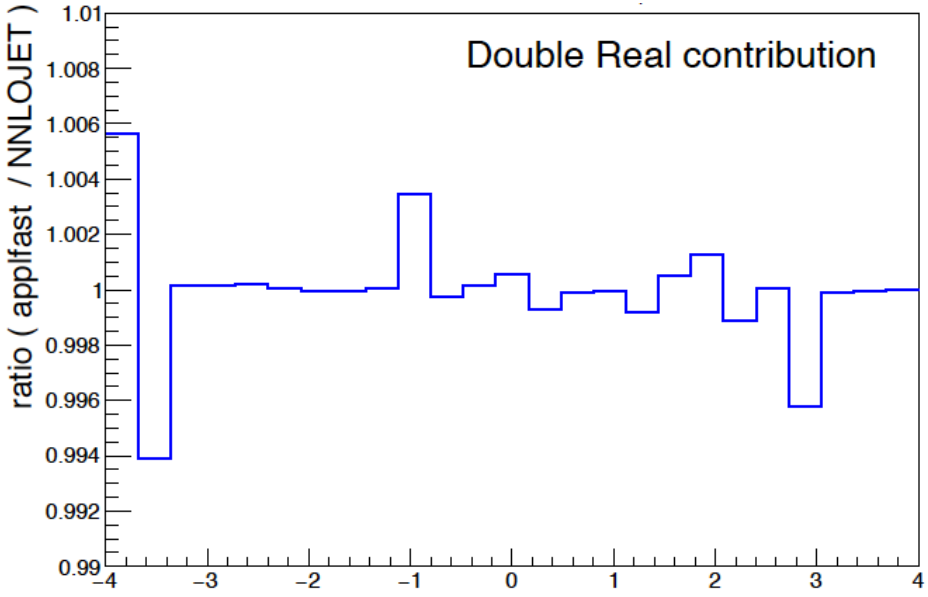
- tests of grid closure of NLO QCD components
- event with short warmup, agreement better than 0.1%

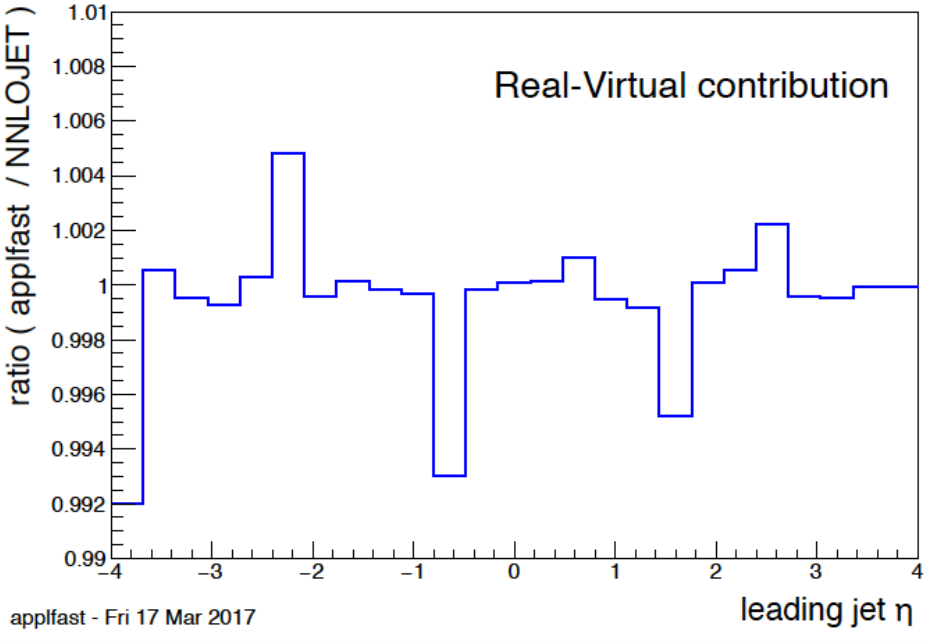


(M. Sutton, xFitter, March 2017)

(pp) inclusive jets up to NNLO







- NNLO contributions also generally agree to better than 0.1%; should improve with longer warmup

(intend to start larger scale production in near future)

(M. Sutton, xFitter, March 2017)

summary and outlook

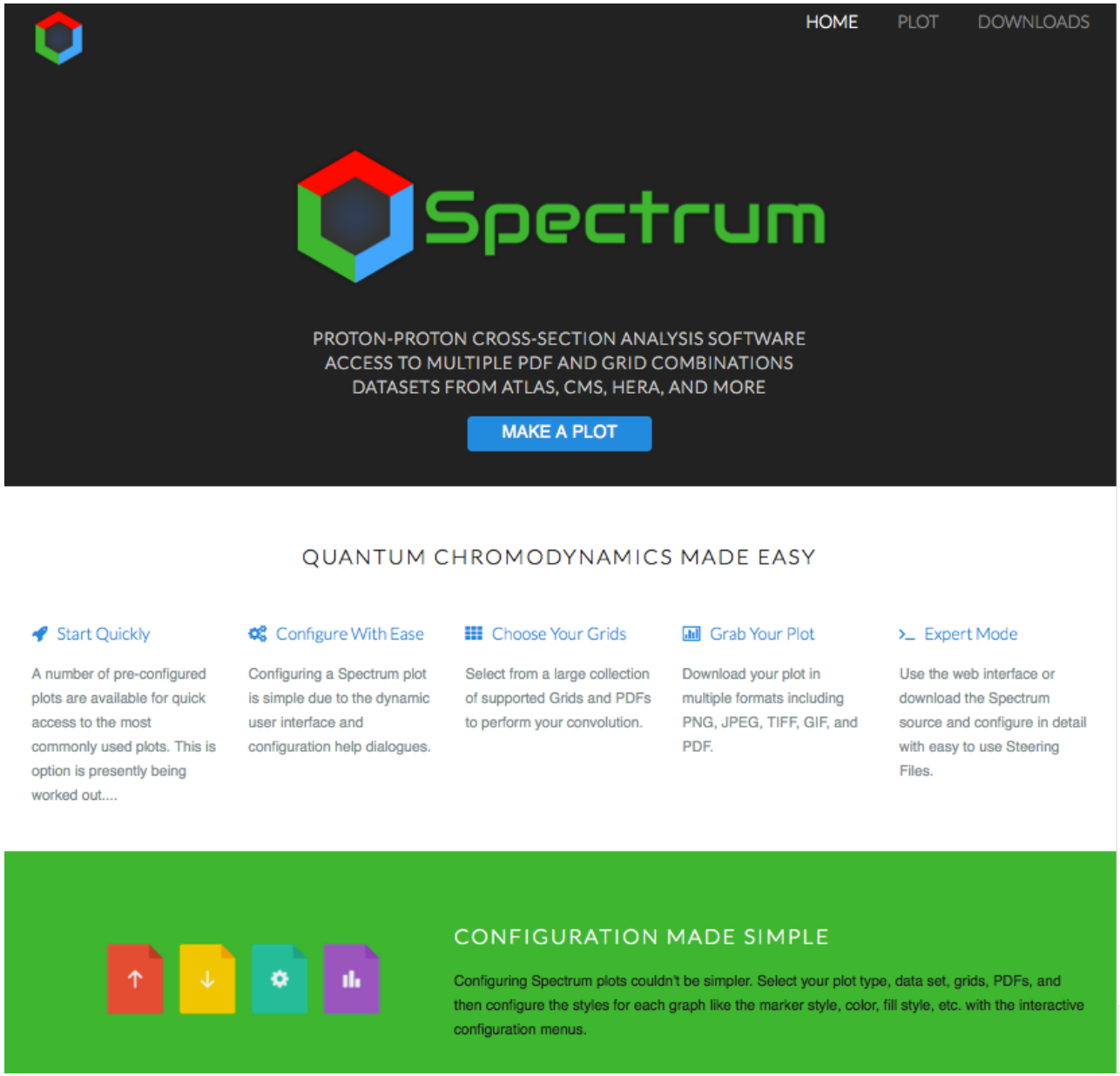
- **NNLOJET project reaching fruition – exceptional progress made on many key processes**
- common, semi-automated interface for multiple physics processes; **the framework where all future development and new calculations from these authors will go**

- **APPLfast-NNLO proof-of-concept maturing**
- **working interface in place**; proof-of-concept validated with common interface for both APPLgrid and fastNLO at LO, NLO and NNLO; technical checks ongoing

- **large scale production launched for Z+Jets in pp, and also DIS jets**
- working on final validation and completion of correct combination of large scale production results
- **more processes currently under development**

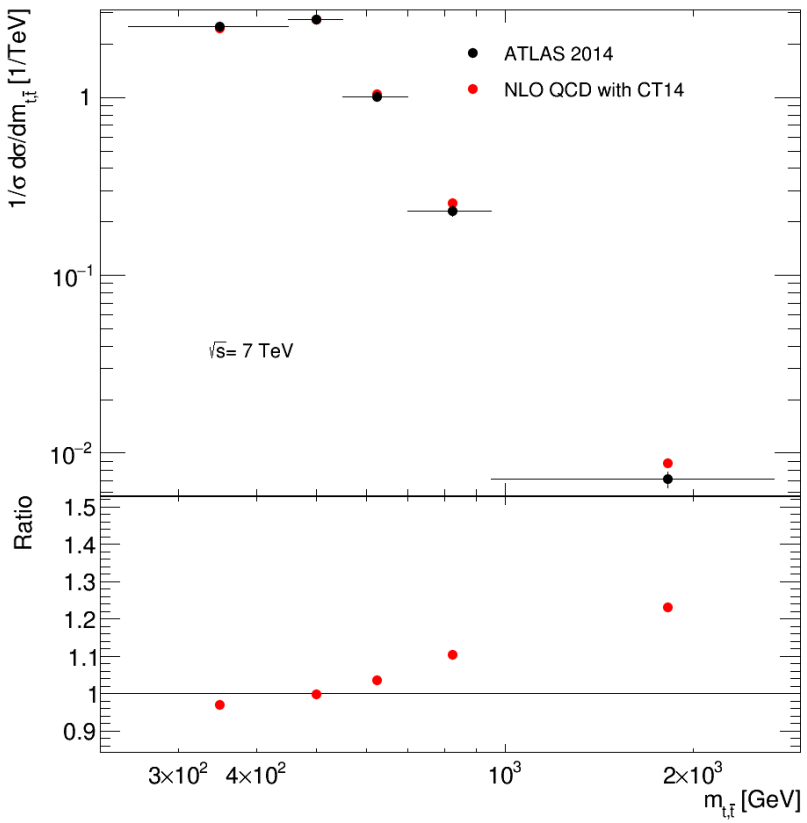
- **look forward to completion of validation for many new NNLO grids**
- when full validation complete – provide community with grids for available processes, **enabling fast NNLO QCD calculations for multiple use cases**

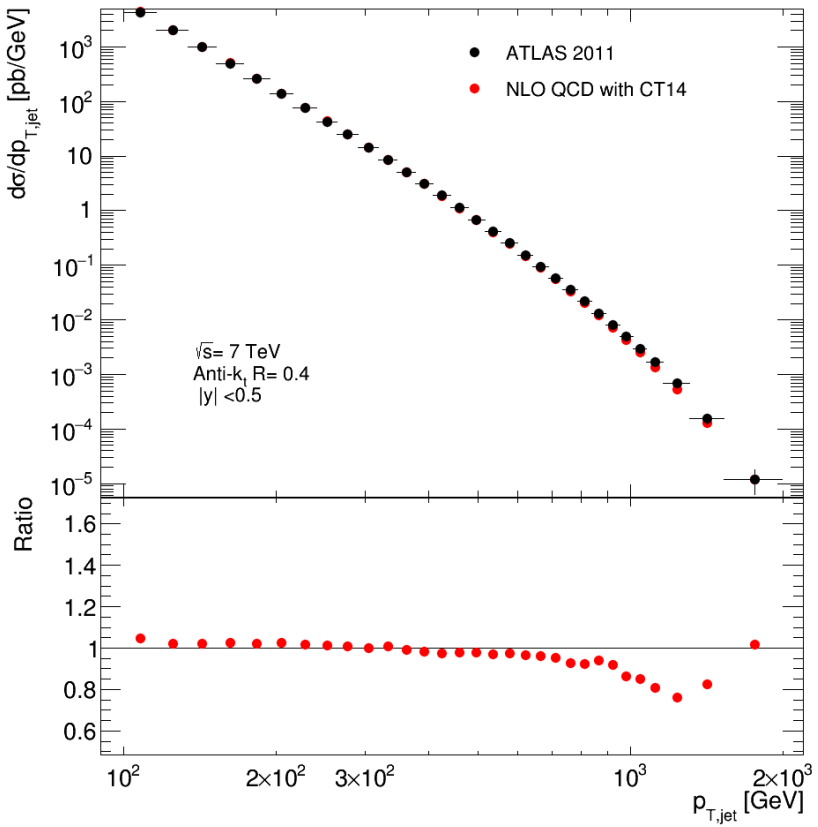
http://spectrum.web.cern.ch/spectrum/index.html



- Spectrum web-server for data to NLO QCD predictions developed – **allows easy plotting of data-theory comparisons**
- grids will be available (via tarballs that can be downloaded); also data files
- **currently a prototype** a place where grids can be collected to form repository

http://spectrum.web.cern.ch/spectrum/index.html





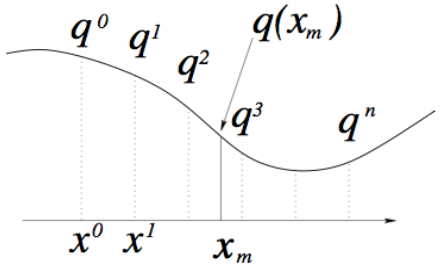
- **several sets of cross section data and grids already available**
- compare data and calculation with any choice of PDF; with or without hadronisation, EW corrections etc.
- aim to collect all available processes and grids (will be downloadable, with appropriate authorship attribution – so please donate your grids!)
- **other FEEDBACK welcome, to:** T. Carli, CERN

extras

Recap of the Numerical Technique

- For a calculation of a cross section from $m = 1 \ldots N$ weights, w_m , from a Monte Carlo integration with momentum fraction x_m, form the product

$$\sum_m w(x_m) q(x_m)$$



- Can interpolate the function $q(x_m)$

$$q(x_m) \approx \sum_i q^{(i)} I^{(i)}(x_m - x^{(i)})$$

- such that

$$\begin{aligned} \sum_m w(x_m) q(x_m) &\approx \sum_i q^{(i)} \sum_m w(x_m) I^{(i)}(x_m - x^{(i)}) \\ &\approx \sum_i q^{(i)} W^{(i)} \end{aligned}$$

- For a calculation of a cross section with $m = 1 \ldots N$ weights, from a Monte Carlo integration with momentum transfer Q^2

$$\begin{aligned} d\sigma &= \sum_p \sum_{m=1}^{N} w_m^{(p)} \left(\frac{\alpha_s(Q_m^2)}{2\pi} \right)^p q(x_m, Q_m^2) \\ &= \sum_p \sum_{ij} q(x_{(i)}, Q_{(j)}^2) \left(\frac{\alpha_s(Q_{(j)}^2)}{2\pi} \right)^p \sum_m^N w_m^{(p)} I_i^x(x_m) I_j^{Q^2}(Q_m^2) \\ &= \sum_p \sum_{ij} q(x_{(i)}, Q_{(j)}^2) \left(\frac{\alpha_s(Q_{(j)}^2)}{2\pi} \right)^p W_{ij}^{(p)} \end{aligned}$$

Proton-Proton Collisions

- For pp collisions need an extra dimension for the PDF of the second colliding hadron

$$d\sigma = \sum_{p} \sum_{m=1}^{N} w_m^{(p)} \left(\frac{\alpha_s(Q_m^2)}{2\pi} \right)^p q_1(x_{1m}, Q_m^2)\ q_2(x_{2m}, Q_m^2)$$

- But there is an implicit summation over parton flavours. Make use of symmetries in the matrix elements to use a vector of $k = 1 \ldots M$ independent weights such that

$$\sum_{ij=q,\bar{q},g} w_{ij} q_{1i}(x_1) q_{2j}(x_2) = \sum_{k=1}^{M} w^{(k)} F^{(k)}(x_1, x_2)$$

- so that

$$d\sigma = \sum_{p} \sum_{k=1}^{M} \sum_{m=1}^{N} w_m^{(p)(k)} \left(\frac{\alpha_s(Q_m^2)}{2\pi} \right)^p F_m^{(k)}(x_{1m}, x_{2m}, Q_m^2)$$

- Which can be placed on a grid in the same way as for DIS

- So from the summation, everything is down to the quality of the interpolation of the pdf at the grid nodes
  - It is s **pure quadrature technique** and is not, in principle subject to statistical fluctuation, or put another way …
  - Each **individual** weight gets added to the grid, and should be well approximated **individually**

Z+Jet NLO subprocesses

- **NNLOJET** calculates many (150) distinct internal processes, many with same input partons
- automatically reduce to 33 parton luminosities for NLO (keep internal mapping of internal process ID to parton luminosity)

0	13 27 41 42 65 73 103 133	$(d,\bar{d})+(s,\bar{s})+(b,\bar{b})$
1	14 28 43 44 66 74 104 134	$(u,\bar{u})+(c,\bar{c})$
2	15 29 45 46 67 75 105 135	$(\bar{d},d)+(\bar{s},s)+(\bar{b},b)$
3	16 30 47 48 68 76 106 136	$(\bar{u},u)+(\bar{c},c)$
4	17 31 77 107 137	$(d,g)+(s,g)+(b,g)$
5	18 32 78 108 138	$(u,g)+(c,g)$
6	19 33 79 109 139	$(g,d)+(g,s)+(g,b)$
7	20 34 80 110 140	$(g,u)+(g,c)$
8	21 22 35 36 85 86 115 116 145 146	(g,g)
9	23 37 81 111 141	$(g,\bar{d})+(g,\bar{s})+(g,\bar{b})$
10	24 38 82 112 142	$(g,\bar{u})+(g,\bar{c})$
11	25 39 83 113 143	$(\bar{d},g)+(\bar{s},g)+(\bar{b},g)$
12	26 40 84 114 144	$(\bar{u},g)+(\bar{c},g)$
13	49 91 121 151	$(d,\bar{d})+(d,\bar{s})+(d,\bar{b})+(s,\bar{d})+(s,\bar{s})+(s,\bar{b})+(b,\bar{d})+(b,\bar{s})+(b,\bar{b})$
14	50 92 122 152	$(d,\bar{u})+(d,\bar{c})+(s,\bar{u})+(s,\bar{c})+(b,\bar{u})+(b,\bar{c})$
15	51 93 123 153	$(u,\bar{d})+(u,\bar{s})+(u,\bar{b})+(c,\bar{d})+(c,\bar{s})+(c,\bar{b})$
16	52 94 124 154	$(u,\bar{u})+(u,\bar{c})+(c,\bar{u})+(c,\bar{c})$
17	53 87 117 147	$(d,d)+(d,s)+(d,b)+(s,d)+(s,s)+(s,b)+(b,d)+(b,s)+(b,b)$
18	54 88 118 148	$(d,u)+(d,c)+(s,u)+(s,c)+(b,u)+(b,c)$
19	55 89 119 149	$(u,d)+(u,s)+(u,b)+(c,d)+(c,s)+(c,b)$
20	56 90 120 150	$(u,u)+(u,c)+(c,u)+(c,c)$
21	57 99 129 159	$(\bar{d},\bar{d})+(\bar{d},\bar{s})+(\bar{d},\bar{b})+(\bar{s},\bar{d})+(\bar{s},\bar{s})+(\bar{s},\bar{b})+(\bar{b},\bar{d})+(\bar{b},\bar{s})+(\bar{b},\bar{b})$
22	58 100 130 160	$(\bar{d},\bar{u})+(\bar{d},\bar{c})+(\bar{s},\bar{u})+(\bar{s},\bar{c})+(\bar{b},\bar{u})+(\bar{b},\bar{c})$
23	59 101 131 161	$(\bar{u},\bar{d})+(\bar{u},\bar{s})+(\bar{u},\bar{b})+(\bar{c},\bar{d})+(\bar{c},\bar{s})+(\bar{c},\bar{b})$
24	60 102 132 162	$(\bar{u},\bar{u})+(\bar{u},\bar{c})+(\bar{c},\bar{u})+(\bar{c},\bar{c})$
25	61 95 125 155	$(\bar{d},d)+(\bar{d},s)+(\bar{d},b)+(\bar{s},d)+(\bar{s},s)+(\bar{s},b)+(\bar{b},d)+(\bar{b},s)+(\bar{b},b)$
26	62 96 126 156	$(\bar{d},u)+(\bar{d},c)+(\bar{s},u)+(\bar{s},c)+(\bar{b},u)+(\bar{b},c)$
27	63 97 127 157	$(\bar{u},d)+(\bar{u},s)+(\bar{u},b)+(\bar{c},d)+(\bar{c},s)+(\bar{c},b)$
28	64 98 128 158	$(\bar{u},u)+(\bar{u},c)+(\bar{c},u)+(\bar{c},c)$
29	69	$(d,d)+(s,s)+(b,b)$
30	70	$(u,u)+(c,c)$
31	71	$(\bar{d},\bar{d})+(\bar{s},\bar{s})+(\bar{b},\bar{b})$
32	72	$(\bar{u},\bar{u})+(\bar{c},\bar{c})$

Z+Jet NNLO subprocesses

- many more individual internal processes (794); can be reduced down to same 33 parton luminosities

0	163 177 191 205 206 245 246 285 301 317 347 377 391 405 435 449 450 489 490 529 530 569 593 617 647 677 707 737 767 899 905 906 935 941 942	$(d, \bar{d}) + (s, \bar{s}) + (b, \bar{b})$
1	164 178 192 207 208 247 248 286 302 318 348 378 392 406 436 451 452 491 492 531 532 570 594 618 648 678 708 738 768 900 907 908 936 943 944	$(u, \bar{u}) + (c, \bar{c})$
2	165 179 193 209 210 249 250 287 303 319 349 379 393 407 437 453 454 493 494 533 534 571 595 619 649 679 709 739 769 909 917 918 945 953 954	$(\bar{d}, d) + (\bar{s}, s) + (\bar{b}, b)$
3	166 180 194 211 212 251 252 288 304 320 350 380 394 408 438 455 456 495 496 535 536 572 596 620 650 680 710 740 770 910 919 920 946 955 956	$(\bar{u}, u) + (\bar{c}, c)$
4	167 181 195 229 230 269 270 289 305 321 351 381 395 409 439 481 482 521 522 561 562 585 586 609 610 621 651 681 711 741 771 817 818 861 862 901 902 937 938	$(d, g) + (s, g) + (b, g)$
5	168 182 196 231 232 271 272 290 306 322 352 382 396 410 440 483 484 523 524 563 564 587 588 611 612 622 652 682 712 742 772 819 820 863 864 903 904 939 940	$(u, g) + (c, g)$
6	169 183 197 233 234 273 274 293 309 323 353 383 397 411 441 473 474 513 514 553 554 577 578 601 602 623 653 683 713 743 773 801 802 845 846 889 890 925 926	$(g, d) + (g, s) + (g, b)$
7	170 184 198 235 236 275 276 294 310 324 354 384 398 412 442 475 476 515 516 555 556 579 580 603 604 624 654 684 714 744 774 803 804 847 848 891 892 927 928	$(g, u) + (g, c)$
8	171 172 185 186 199 200 325 326 355 356 385 386 399 400 413 414 443 444 629 630 659 660 689 690 719 720 749 750 779 780 797 798 799 800 841 842 843 844 885 886 887 888 921 922 923 924	(g, g)
9	173 187 201 237 238 277 278 295 311 327 357 387 401 415 445 477 478 517 518 557 558 581 582 605 606 625 655 685 715 745 775 805 806 849 850 893 894 929 930	$(g, \bar{d}) + (g, \bar{s}) + (g, \bar{b})$
10	174 188 202 239 240 279 280 296 312 328 358 388 402 416 446 479 480 519 520 559 560 583 584 607 608 626 656 686 716 746 776 807 808 851 852 895 896 931 932	$(g, \bar{u}) + (g, \bar{c})$
11	175 189 203 241 242 281 282 297 313 329 359 389 403 417 447 485 486 525 526 565 566 589 590 613 614 627 657 687 717 747 777 833 834 877 878 913 914 949 950	$(\bar{d}, g) + (\bar{s}, g) + (\bar{b}, g)$
12	176 190 204 243 244 283 284 298 314 330 360 390 404 418 448 487 488 527 528 567 568 591 592 615 616 628 658 688 718 748 778 835 836 879 880 915 916 951 952	$(\bar{u}, g) + (\bar{c}, g)$
13	213 253 335 365 423 457 497 537 635 665 695 725 755 785 813 821 822 857 865 866	$(d, \bar{d}) + (d, \bar{s}) + (d, \bar{b}) + (s, \bar{d}) + (s, \bar{s}) + (s, \bar{b}) + (b, \bar{d}) + (b, \bar{s}) + (b, \bar{b})$
14	214 254 336 366 424 458 498 538 636 666 696 726 756 786 814 858	$(d, \bar{u}) + (d, \bar{c}) + (s, \bar{u}) + (s, \bar{c}) + (b, \bar{u}) + (b, \bar{c})$
15	215 255 337 367 425 459 499 539 637 667 697 727 757 787 815 859	$(u, \bar{d}) + (u, \bar{s}) + (u, \bar{b}) + (c, \bar{d}) + (c, \bar{s}) + (c, \bar{b})$
16	216 256 338 368 426 460 500 540 638 668 698 728 758 788 816 823 824 860 867 868	$(u, \bar{u}) + (u, \bar{c}) + (c, \bar{u}) + (c, \bar{c})$
17	217 257 331 361 419 461 501 541 631 661 691 721 751 781 809 853	$(d, d) + (d, s) + (d, b) + (s, d) + (s, s) + (s, b) + (b, d) + (b, s) + (b, b)$
18	218 258 332 362 420 462 502 542 632 662 692 722 752 782 810 854	$(d, u) + (d, c) + (s, u) + (s, c) + (b, u) + (b, c)$
19	219 259 333 363 421 463 503 543 633 663 693 723 753 783 811 855	$(u, d) + (u, s) + (u, b) + (c, d) + (c, s) + (c, b)$
20	220 260 334 364 422 464 504 544 634 664 694 724 754 784 812 856	$(u, u) + (u, c) + (c, u) + (c, c)$
21	221 261 343 373 431 465 505 545 643 673 703 733 763 793 829 873	$(\bar{d}, \bar{d}) + (\bar{d}, \bar{s}) + (\bar{d}, \bar{b}) + (\bar{s}, \bar{d}) + (\bar{s}, \bar{s}) + (\bar{s}, \bar{b}) + (\bar{b}, \bar{d}) + (\bar{b}, \bar{s}) + (\bar{b}, \bar{b})$
22	222 262 344 374 432 466 506 546 644 674 704 734 764 794 830 874	$(\bar{d}, \bar{u}) + (\bar{d}, \bar{c}) + (\bar{s}, \bar{u}) + (\bar{s}, \bar{c}) + (\bar{b}, \bar{u}) + (\bar{b}, \bar{c})$
23	223 263 345 375 433 467 507 547 645 675 705 735 765 795 831 875	$(\bar{u}, \bar{d}) + (\bar{u}, \bar{s}) + (\bar{u}, \bar{b}) + (\bar{c}, \bar{d}) + (\bar{c}, \bar{s}) + (\bar{c}, \bar{b})$
24	224 264 346 376 434 468 508 548 646 676 706 736 766 796 832 876	$(\bar{u}, \bar{u}) + (\bar{u}, \bar{c}) + (\bar{c}, \bar{u}) + (\bar{c}, \bar{c})$
25	225 265 339 369 427 469 509 549 639 669 699 729 759 789 825 837 838 869 881 882	$(\bar{d}, d) + (\bar{d}, s) + (\bar{d}, b) + (\bar{s}, d) + (\bar{s}, s) + (\bar{s}, b) + (\bar{b}, d) + (\bar{b}, s) + (\bar{b}, b)$
26	226 266 340 370 428 470 510 550 640 670 700 730 760 790 826 870	$(\bar{d}, u) + (\bar{d}, c) + (\bar{s}, u) + (\bar{s}, c) + (\bar{b}, u) + (\bar{b}, c)$
27	227 267 341 371 429 471 511 551 641 671 701 731 761 791 827 871	$(\bar{u}, d) + (\bar{u}, s) + (\bar{u}, b) + (\bar{c}, d) + (\bar{c}, s) + (\bar{c}, b)$
28	228 268 342 372 430 472 512 552 642 672 702 732 762 792 828 839 840 872 883 884	$(\bar{u}, u) + (\bar{u}, c) + (\bar{c}, u) + (\bar{c}, c)$
29	291 307 573 597 897 933	$(d, d) + (s, s) + (b, b)$
30	292 308 574 598 898 934	$(u, u) + (c, c)$
31	299 315 575 599 911 947	$(\bar{d}, \bar{d}) + (\bar{s}, \bar{s}) + (\bar{b}, \bar{b})$
32	300 316 576 600 912 948	$(\bar{u}, \bar{u}) + (\bar{c}, \bar{c})$

APPLfast grid generation – some technicalities

- when generating a grid, need to know the optimal phase space in x_1, x_2 and Q^2
  - use a **warmup run** to determine phase space – fill with dummy weights, interpolation between grid points not performed
  - **optimise phase space to that determined by the warmup run**
- if inadequate stats generated during warmup, phase space will be incomplete
  - weights during **production run** can fall outside grid, causing large deviations from expected cross section
- to avoid this, can:
  1. run with more weights during warmup run
  2. extend phase space by additional buffer zone into which grid can overflow if required
  3. allow grid to dynamically add additional buffers
- options 1 and 2 are implemented:
  - since neither the interpolation nor weights are needed during warmup, option to run **NNLOJet** with **unit phase space** has been implemented
  - significantly greater than 100 times faster at **NNLO**
  - option 3 being developed

grid filling developments

- for the grid filling, call the fill methods for each process generated
  - at NNLO nearly 800 separate processes, each called **many times** for each **phase space point**
  - **calling fill methods for each weight is maximally time consuming**

- thus, implemented a filling cache:
  - each weight added to a **weight vector** corresponding to a **unique phase space point** and **observable** value
  - when phase space point changes, flush cache to grid
  - significantly reduces number of calls to grid filling, **by factors of 20–200**, depending on process

typical sample processing times

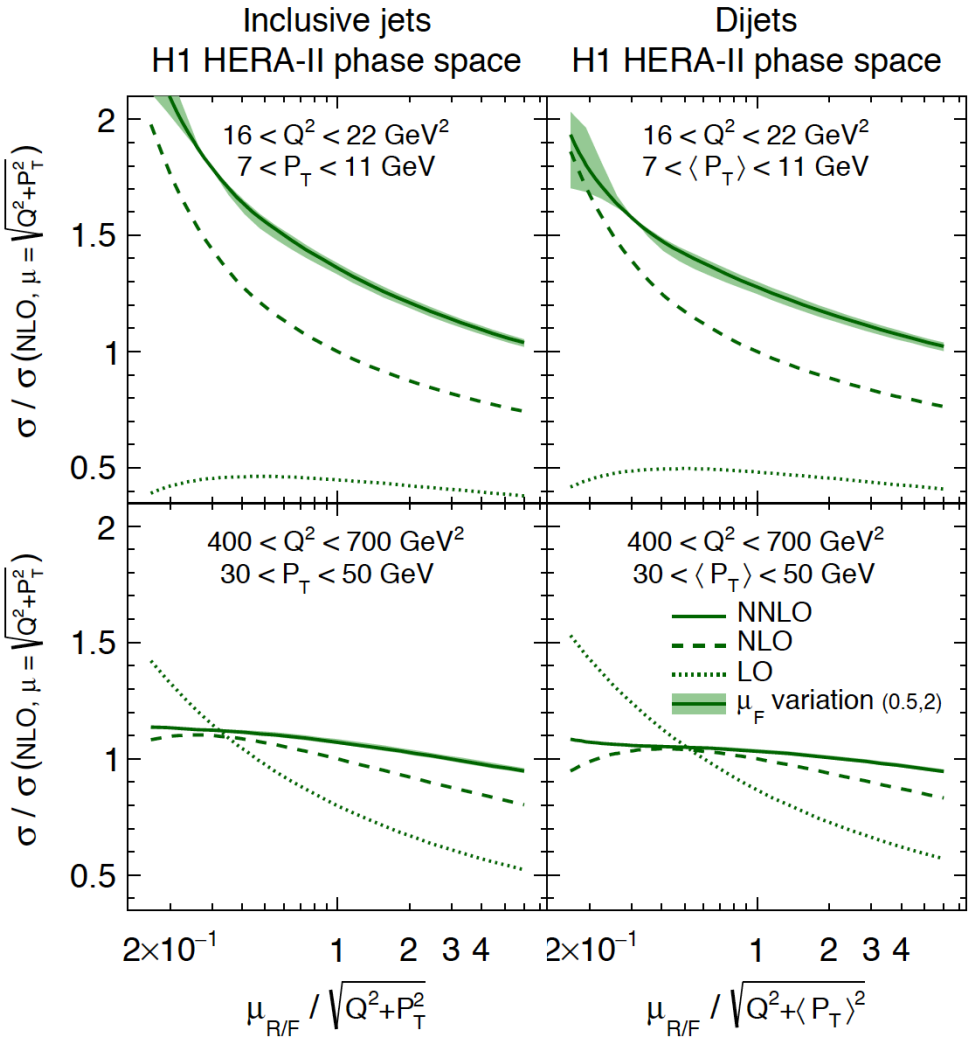
- **NNLOJET Z+Jet** without APPLfast grids
- LO : 100 jobs × (5 mins)
- NLO-V : 100 jobs × (10 mins)
- NLO-R : 100 jobs × (15 mins)
- NNLO-VV : 100 jobs × (1 h)
- NNLO-RV : 1000 jobs × (10–20 h)
- NNLO-RR : 5000 jobs × (20 h)

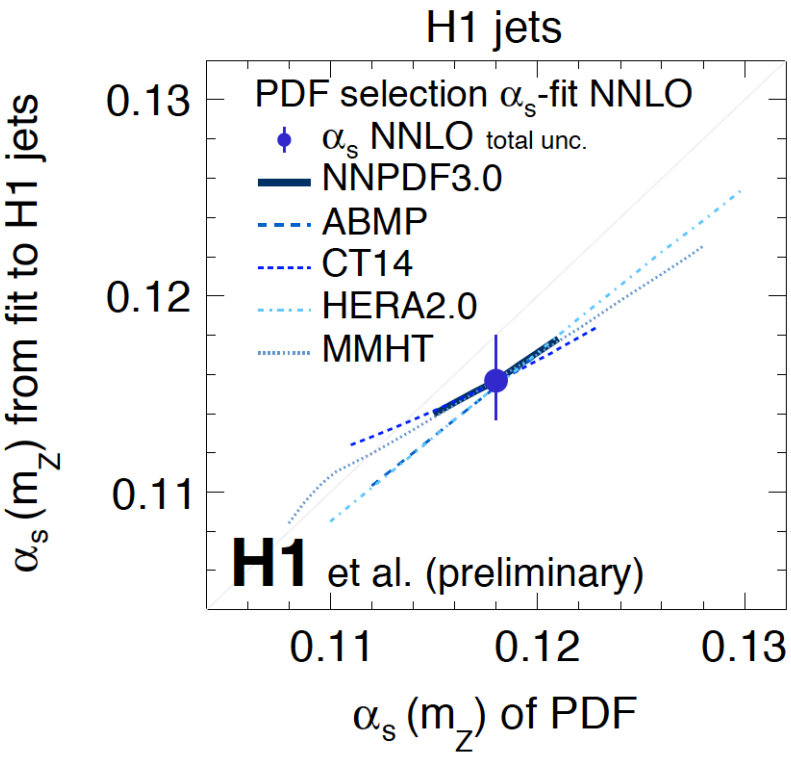
- **NNLOJET DIS jets** with grids
  each job 8–16 h CPU time
- LO : 50 jobs (5G events)
- NLO-V : 40 jobs (2G events)
- NLO-R : 80 jobs (2G events)
- NNLO-VV : 100 jobs (1.5G events)
- NNLO-RV : 5000 jobs (5G events)
- NNLO-RRa : 10000 jobs (5G events)
- NNLO-RRb : 2000 jobs (5G events)

- initial performance penalty of grid production versus NNLOJET alone, reduced to roughly a factor of only about 2 (depends in detail on physics process and grid setup)

- still gain 100k's of CPU hours for each avoided full run of calculation

NNLO DIS jets





- DIS jets also available in NNLOJET (arXiv:1703.05977, arXiv:1606.03991)
  APPLfast interface functional,
  as part of same project

used to extract value of α_s at NNLO using H1 DIS jets
(H1prelim-17-031; see talk by D. Britzger, this conf.)