

APPL \int fast-NNLO

(amongst other things...)

Mark Sutton

The University of Sussex

Daniel Britzger[†], Claire Gwenlan[‡], Alex Huss[§]

Tom Morgan*, Klaus Rabbertz^a

DESY[†], Oxford[‡], ETH Zurich[§], IPPP Durham*, Karlsruhe^a

22nd August 2016

Work supported by the IPPP Associateship program

Timeline of grid technology - from QCD@LHC 2015

2000	First implementation of grid technique for DIS at H1- Markus Wobisch DESY-THESIS-2000-049
2001	First basic fit in DIS using jets H1 -EPJ C19, 289 (2001)
2004	First full fit in DIS using jets from ZEUS - PRD 67 0120071 (2003)
2005	APPLgrid for jets in hadron-hadron collisions - Carli, Salam, Siegert <ul style="list-style-type: none"> - C++, fully open source - user code for grid generation available - arbitrary scale variation
< 2006	fastNLO implementation for DIS and jets in hadron-hadron with NLOjet++ and threshold corrections fro Kidonakis et al, Kluge Rabbertz, Wobisch <ul style="list-style-type: none"> - Separate Fortran routines for only precomputed grids - limited precomputed scale choices - No user grid generation
Nov 2009	APPLgrid for jets in hadron-hadron - first full release <ul style="list-style-type: none"> - custom sparse memory structure for more efficient storage - arbitrary beam energy scaling - fastNLO interface - First implementation of non-jet cross sections - MCFM interface for inclusive W and Z production at NLO
2010	APPLgrid for other processes in hadron-hadron <ul style="list-style-type: none"> - Extension to heavy flavours in MCFM QQ - ttbar, bbbar, ccbar
Aug 2012	fastNLO 2 + toolkit produced <ul style="list-style-type: none"> - New C++ interface and user grid generation code made available
Jun 2013	APPLgrid for other processes in hadron-hadron <ul style="list-style-type: none"> - Extension to essentially all remaining processes in MCFM, including Z, W + jets, W+c etc
July 2013	APPLgrid Native interface to Sherpa - All NLO QCD in Sherpa
Dec 2013	APPLgrid modifications for MCFM integrated into official MCFM 6.7 MCgrid APPLgrid interface to Sherpa - All NLO QCD in Sherpa from within Rivet
June 2014	AMCfast - AMC@NLO interterface to APPLgrid - All NLO QCD in aMC@NLO
Sept 2015	fastNLO Interface to DiffTop
Early 2015	fastNLO integration with Sherpa using the MCgrid APPLgrid - Sherpa interface Watch this space ...

Timeline of grid technology (and associated items)

Oct 2015	Initial discussions for the APPLfast - NNLO project (of which more later)
March 2016	Implementation of the photon density within the photon Implementation with APPLgrid and aMCFast - Stefano Carrazza
May 2016	APFELgrid - modified grids using APFEL evolution Valerio Bertone, Stefano Carrazza, Nathan P. Hartland
July 2016	NNLOJET z+jets cross section A. Gehrmann-De Ridder, T. Gehrmann, E.W.N. Glover, A. Huss, T. A. Morgan
late 2016	top pair production at NNLO interfaced to fastNLO Michael Czakon, David Heymes and Alexander Mitov
August 2016	First public APPLfast status report (today !)

Recap of the Numerical Technique

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

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

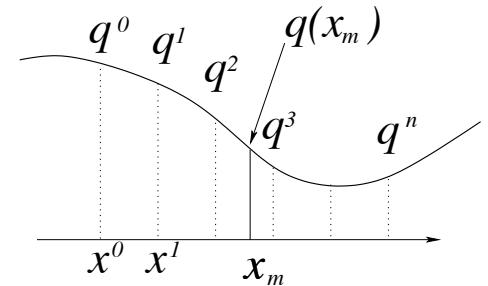
- Can interpolate the function $q(x_m)$

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

- such that $\sum_m w(x_m) q(x_m) \approx \sum_i q^{(i)} \sum_m w(x_m) I^{(i)}(x_m - x^{(i)}) \approx \sum_i q^{(i)} W^{(i)}$

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

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



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 $k = 1 \dots M$ 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 a **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**

appl(e)tise

/'ap(ə)ltʌɪz/

noun trademark

1. a sparkling fruit juice based drink created by blending apple juice with carbonated water (now appeltiser)

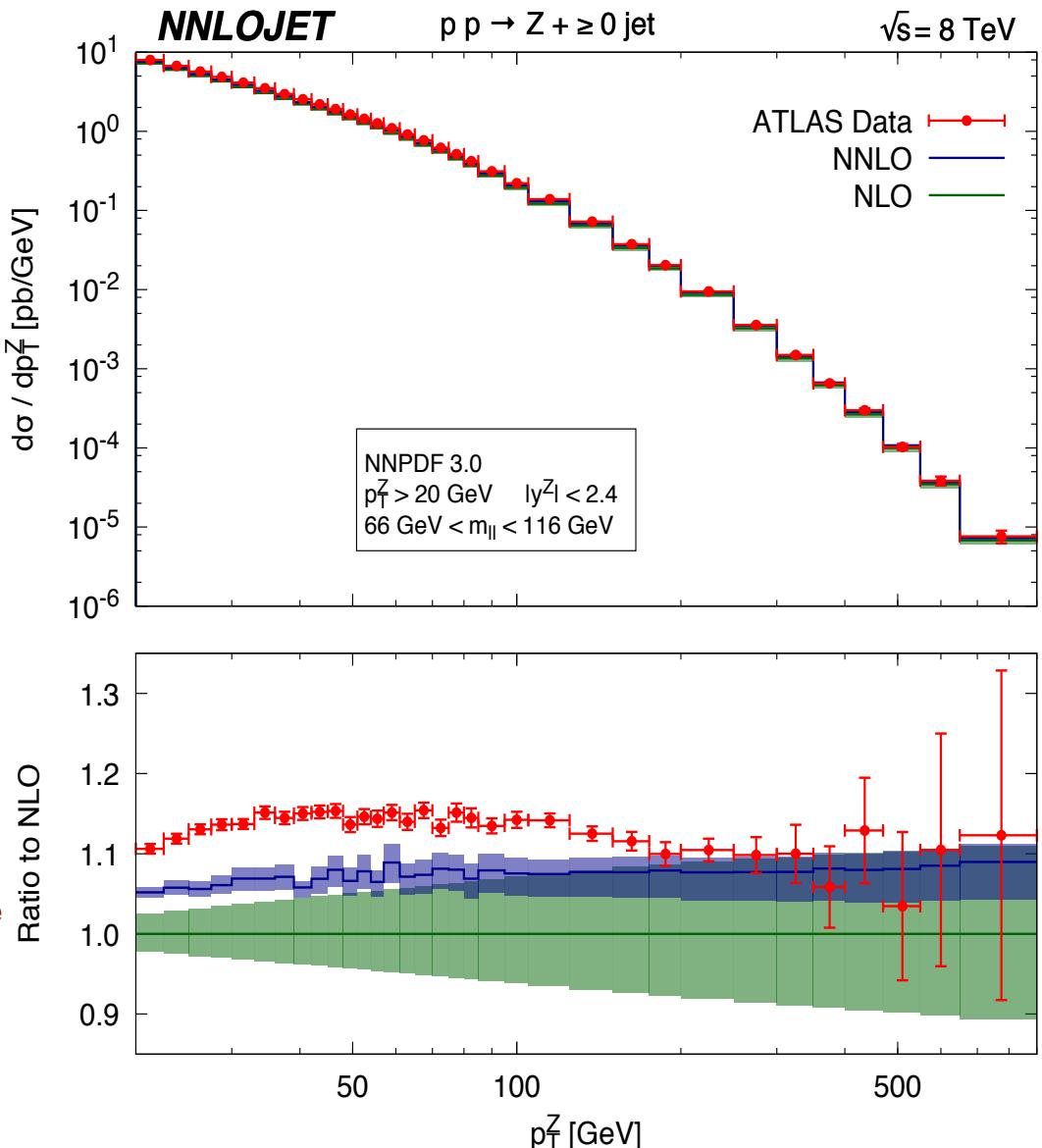
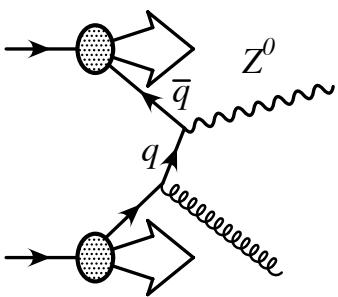
verb

1. to modify or add code (to an existing calculation) for the purposes of storing information (from the calculation) such that the approximate result (of the calculation) with different input parameters can be quickly determined



NNLOJET (and APPLfast-NNLO)

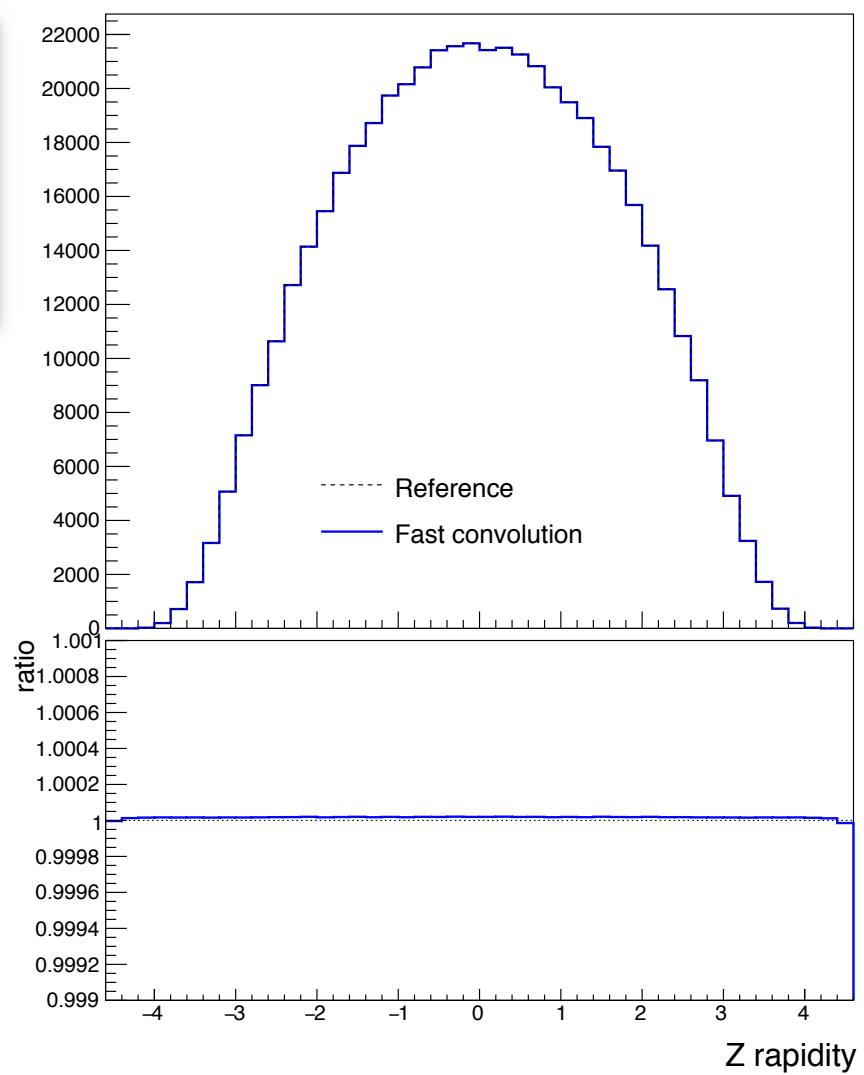
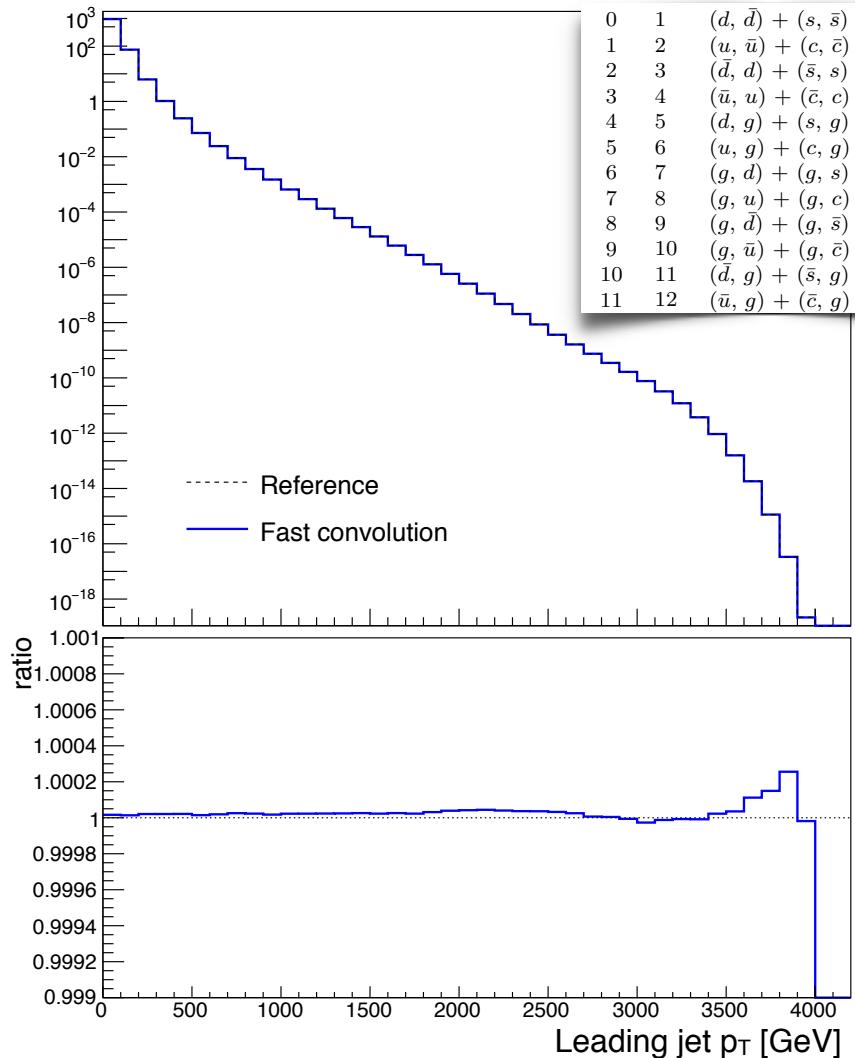
- Semi-automated calculation of cross sections at NNLO from the IPPP, Zurich, ETH and others
 - Gehrmann-De Ridder *et al* [arXiv: 1607.01749](https://arxiv.org/abs/1607.01749)
 - See talk from **Alex Huss** tomorrow
- APPLfast-NNLO
 - Developers from **NNLOJET**, **APPLgrid** and **fastNLO**
 - A single, combined interface for NNLOJET with both APPLgrid and fastNLO
- Many processes implemented in NNLOJET
 - Developing a generic interface for **all available** processes
 - Concentrating on Z + jets at NNLO for the initial development and proof-of-concept



Grid generation

- 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 the phase space - fill with dummy weights, interpolation isn't performed,
 - Optimise the phase space to that determined by the warmup run
- If inadequate statistics are generated during the warmup run, the phase space will be incomplete
 - Weights during the production run can fall outside the grid and will cause large deviations from the expected cross section
- To avoid this can
 1. Run with more weights during the warmup run
 2. Extend the phase space by some additional buffer zone into which the grid can overflow if required
 3. Allow the grid to dynamically add additional buffers
- The first two have been implemented :
 - Since neither the interpolation nor the weights are needed during the warmup, the option to run NNLOJET generating events with unit phase space has been implemented on the NNLOJET side
 - Much faster at LO, and significantly greater than 100 times faster at NNLO
 - Option 3 is being developed - but no concrete time scale at the moment

Z + jets at Leading order



- Note the $\pm 0.1\%$ maximum range in ratio plots
- As always, Leading Order component well reproduced - usual issues near the edges of the phase space

Z + jets at NLO

- NNLOJET calculates using many (150) distinct internal processes, many with the same input partons
 - Automatically reduce down to 33 parton luminosities for the NLO process - combine for Real and Virtual contributions
 - Keep the internal mapping of the internal process ID to the 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	$(d, g) + (\bar{s}, g) + (\bar{b}, g)$
12	26 40 84 114 144	$(\bar{u}, g) + (\bar{c}, g)$
13	49 91 121 151	$(d, \bar{d}) + (d, \bar{s}) + (d, \bar{b}) + (s, \bar{d}) + (s, \bar{b}) + (b, \bar{d}) + (b, \bar{s}) + (b, \bar{b})$
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	$(\bar{d}, \bar{d}) + (\bar{d}, \bar{s}) + (\bar{d}, \bar{b}) + (\bar{s}, \bar{d}) + (\bar{s}, \bar{b}) + (\bar{b}, \bar{d}) + (\bar{b}, \bar{s}) + (\bar{b}, \bar{b})$
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 + jets at NNLO

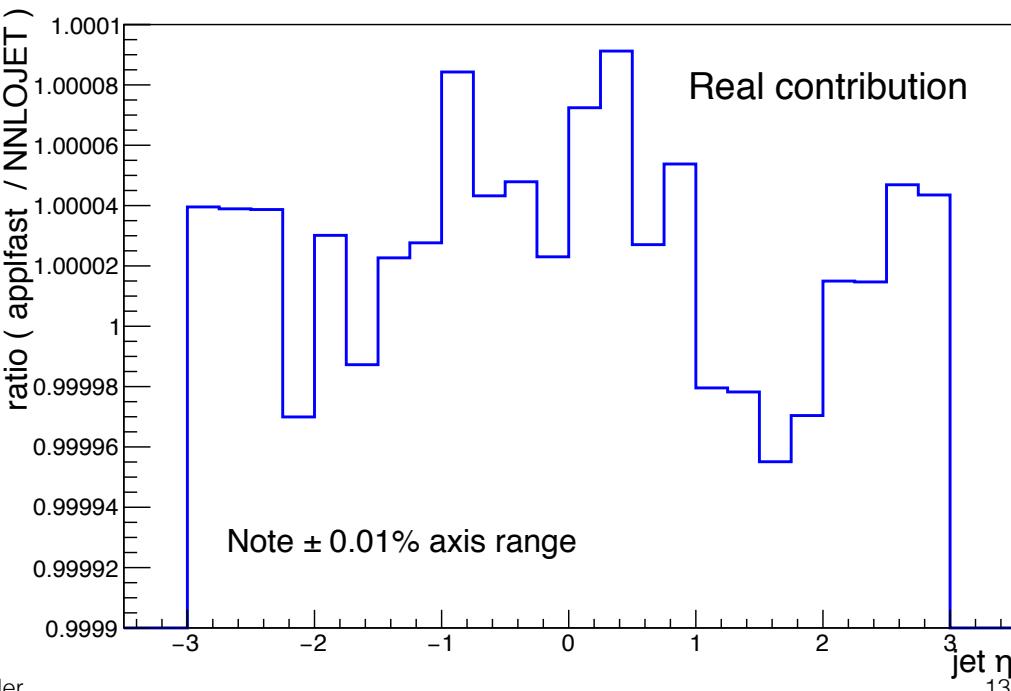
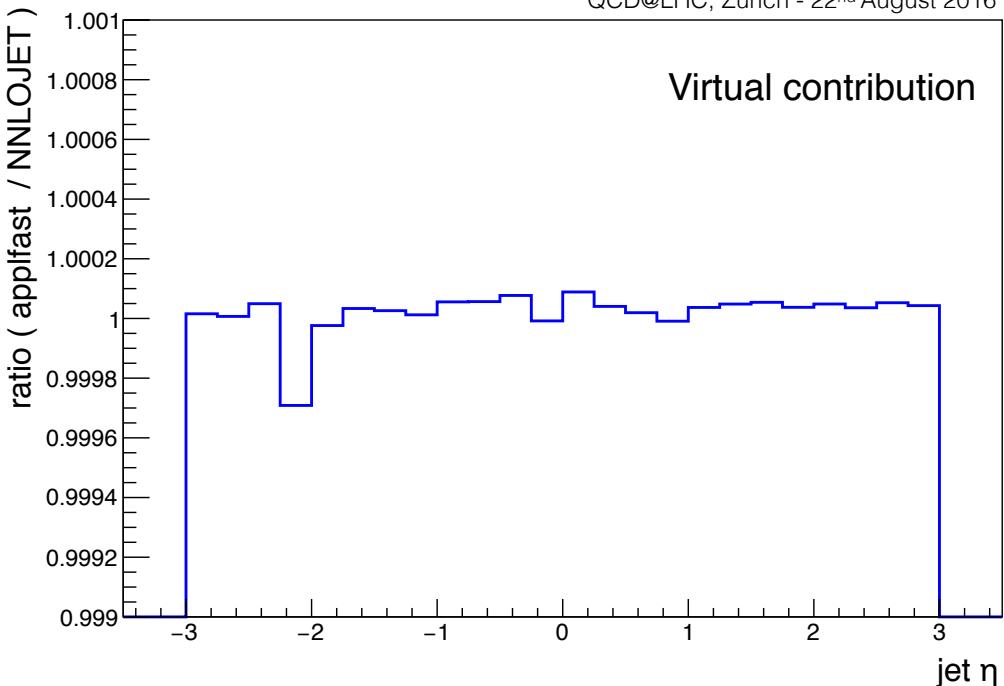
- Again the same 33 input parton luminosities as in the NNLO case, however, many more (794) individual internal processes

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	$(d, d) + (d, \bar{s}) + (d, \bar{b}) + (\bar{s}, d) + (\bar{s}, \bar{s}) + (\bar{s}, \bar{b}) + (\bar{b}, \bar{d}) + (\bar{b}, \bar{s}) + (\bar{b}, \bar{b})$
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	$(\bar{u}, \bar{u}) + (\bar{c}, \bar{c})$

Developments for grid filling

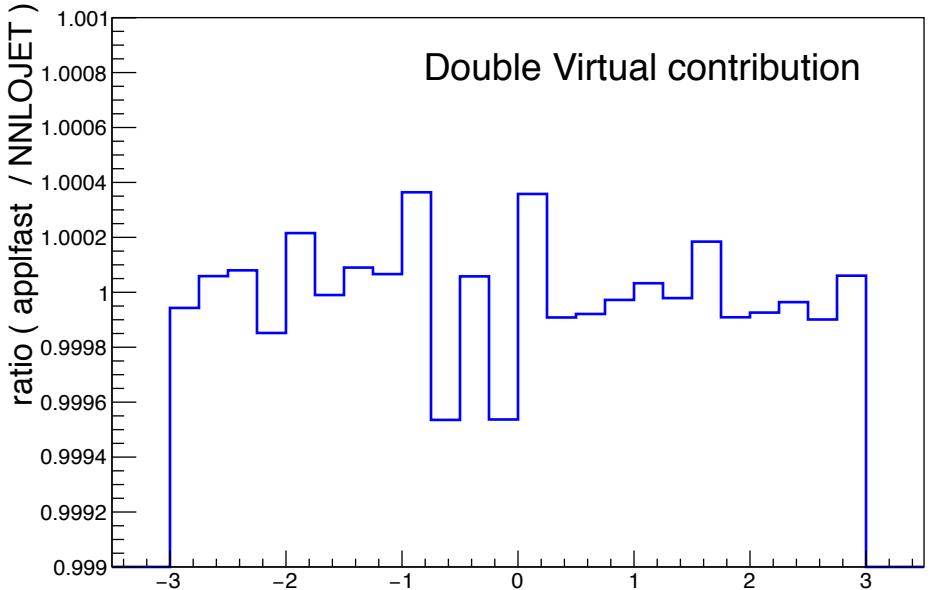
- For the filling call the fill methods for each process generated
 - At NNLO have over 700 separate processes, each called many times for each phase space point - the Real emission components may have several different but related phase space points
 - Calling the fill methods for each weight is maximally time consuming
- Implemented a filling cache
 - Each weight is added to a weight vector corresponding to a unique phase space point and observable value
 - When the phase space point changes, flush the cache to the grid
 - Significantly reduces the number of calls to the grid filling, by factors of between 20 and 200 depending on the process

Z + jets at NLO

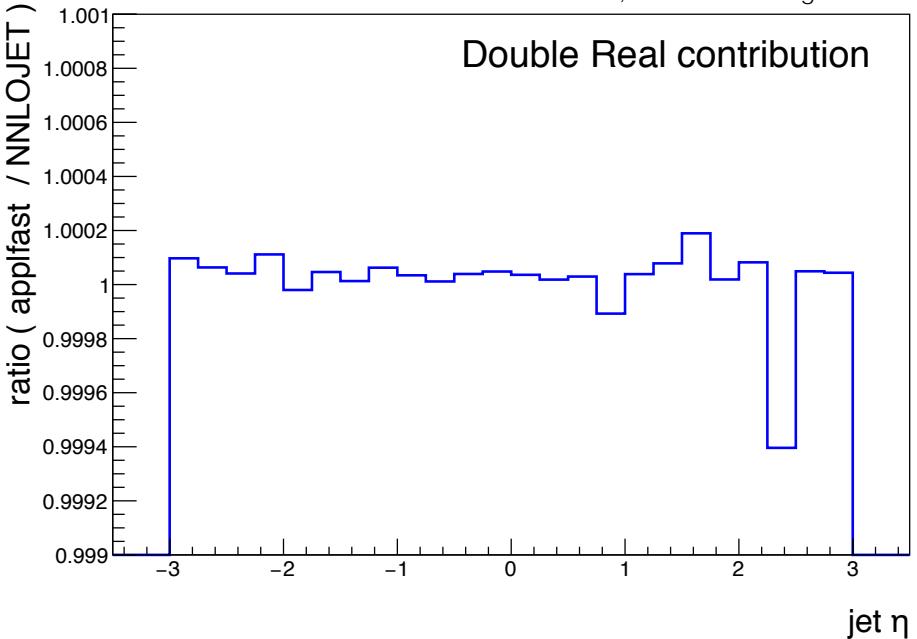


- Early tests of grid closure of the NLO components are promising
 - Only low statistics runs so far
 - Even with short warmup, agreement to well within 0.1%

Double Virtual contribution



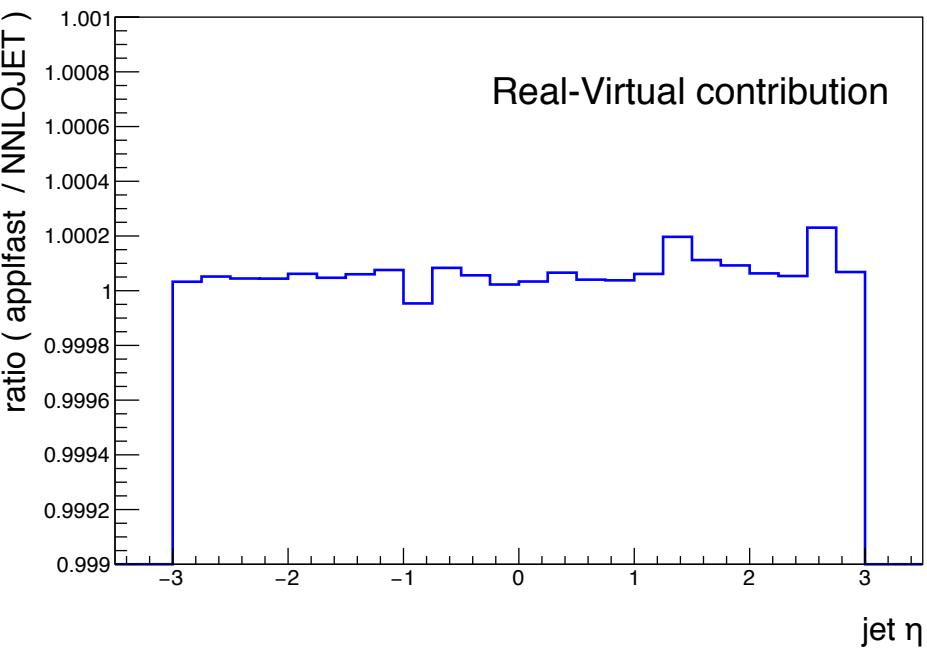
Double Real contribution



$Z + \text{jets}$ at NNLO

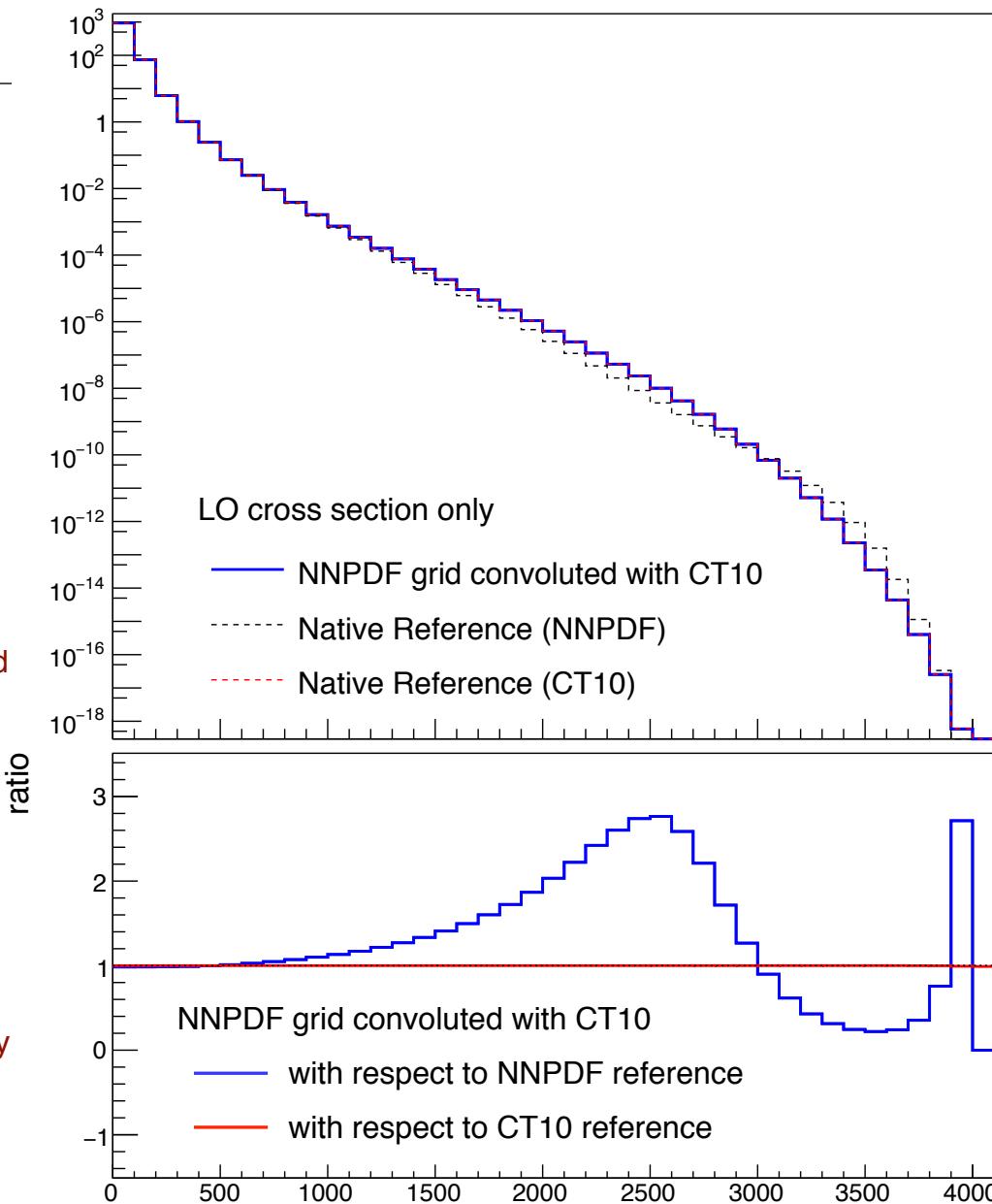
- Early proof-of-concept convolutions ratios
 - Note large excursions for some bins due to inadequate statistics used for the determination of the optimised phase space
- Working on improving the optimisation strategy

Real-Virtual contribution



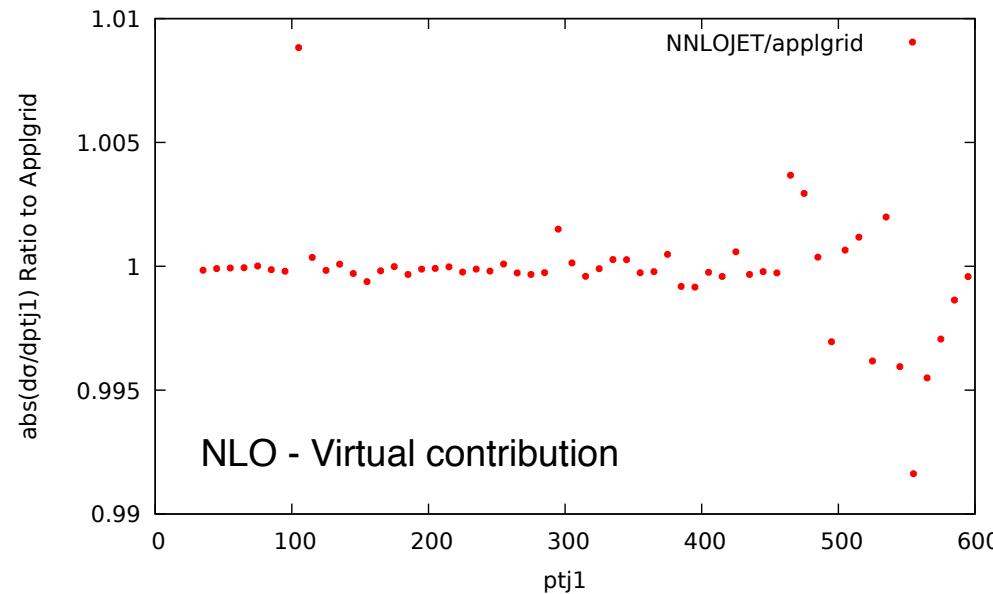
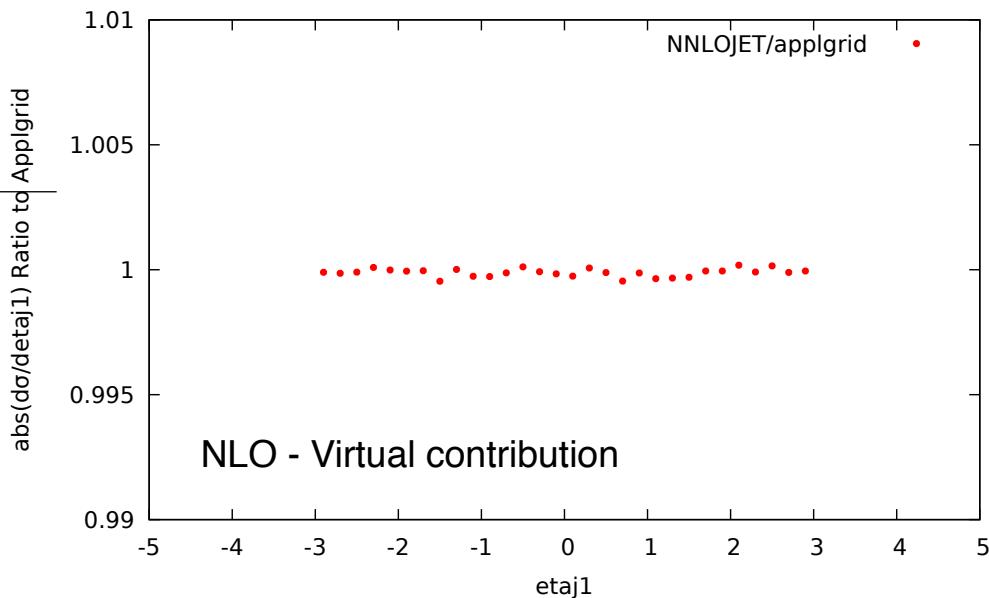
Grid closure - LO

- Need to check proper closure of the grids
 - Besides verification that grids generated with **PDF "X"** can reproduce the cross section ...
 - ... need to verify that the convolution with **PDF "Y"** with a grid generated with **PDF "X"** actually reproduces the expected cross section generated with **PDF "Y"**
- Essentially trivial proof of concept for the LO component only
 - Just a taster of things to come ...
 - Calculations with different PDFs are statistically independent so need many events
 - For NLO, NNLO comparison need the uncertainty on the calculation to be small
 - Lots of events needed - will take a while to check properly ...



Grid closure - NLO virtual component

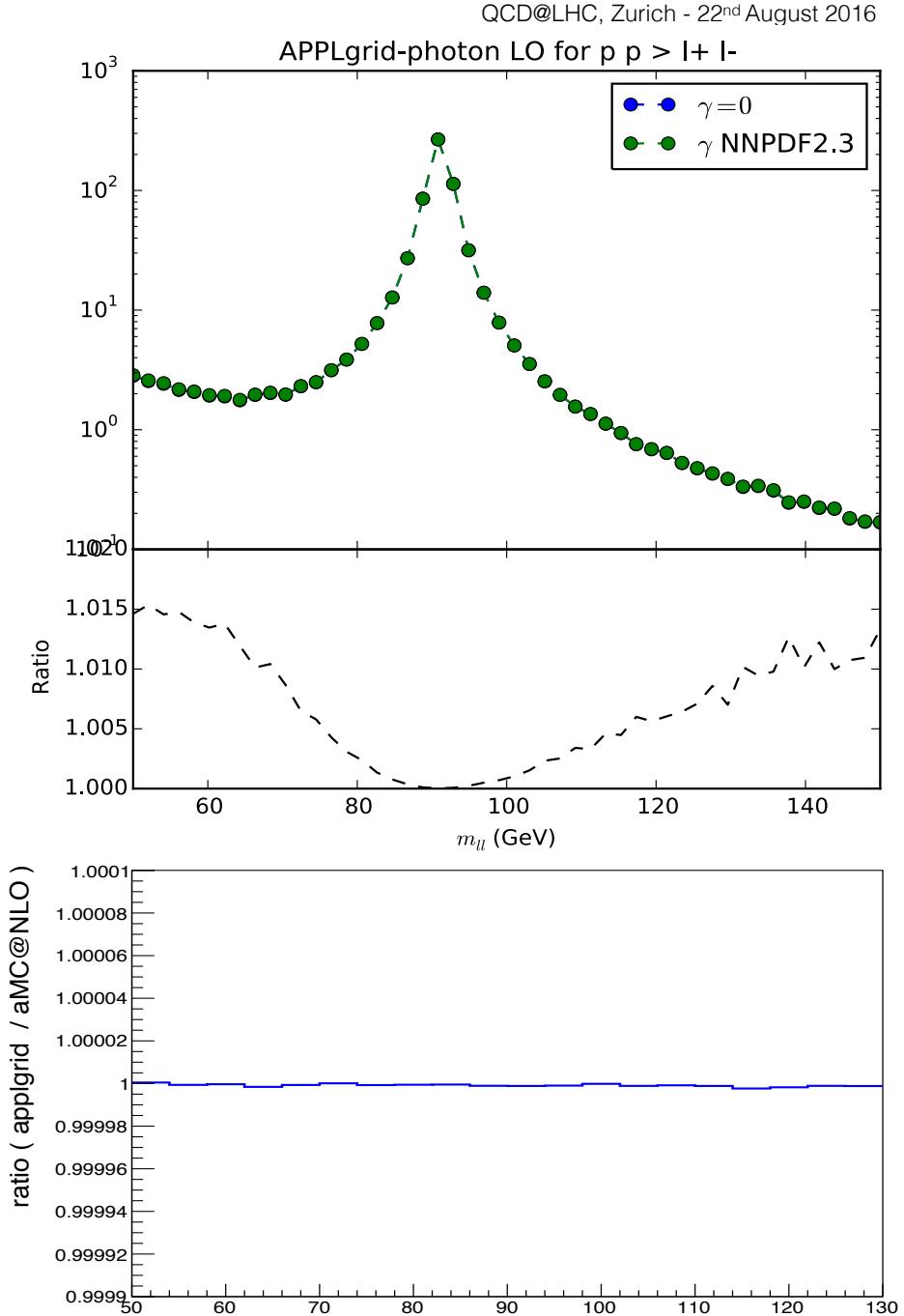
- Checking the closure of the NLO components:
 - Grid generated with NNPDF 2.3 nnlo
 - Convolved with CT14 compared to NNLOJET calculation with CT14
 - Agreement to well within 0.1 % - extremely encouraging
- Work ongoing testing the NNLO components



The other things ...

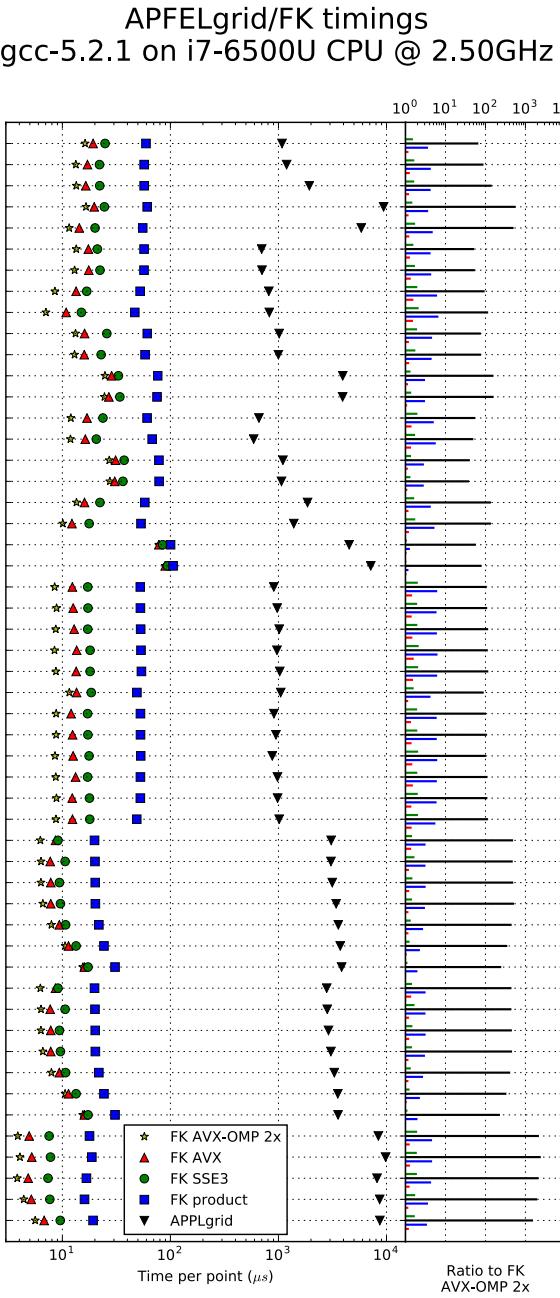
photon density in the proton

- Photon contribution to the proton
- First implementation of the addition of the photon PDF weights in the applgrid framework - **Stefano Carrazza**
- Code not yet available in the stable release
 - Initial developments to integrate this feature in aMC@NLO – LO DY applgrid convolution with and without an initial-state photon PDF (from NNPDF2.3QED)
- Ongoing developments to extend the store EW corrections in applgrid



The other things (contd) ...

- APFELgrid, Bertone, Carrazza, and Hartland
 - arxiv.org/abs/1605.02070
 - Manipulation of the full grids to reduce size for faster convolution
- Top pair production - Czakon, Heymes, and Mitov
 - arxiv.org/abs/1606.03350v1
 - fastNLO interface being developed by calculation authors
 - NLO component agrees well, still developing the interface with the NNLO component
- DiffTop - Guzzi, Lipka, and Moch
 - difftop.hepforge.org
 - Approximate NNLO calculation using threshold resummation
- All of these are possible because of the open source nature of the grid code (APPLgrid and fastNLO)
- It is worth noting that *appletising* a calculation is not difficult if you know the code well
- If you are the author of a calculation it may be worth thinking about the structure of your code in light of such an interface



Great expectations

- Over the last year there have been many significant developments for fast grid techniques
 - Photon density, NNLO calculations for top, APPLfast-NNLO interface to NNLOJET
- The community of folk implementing grid interfaces is growing - no longer just the APPLgrid or fastNLO developers
 - Interfacing new calculations with grid technology - either APPLgrid or fastNLO - should be reasonably straightforward if you know the calculation code - perhaps we need a forum to exchange experience
 - The more generic interfaces being implemented now will no doubt help these efforts
- The APPLfast-NNLO proof of concept development is starting to mature
 - Several stages still need to be checked - full combination of grids from multiple runs for large statistics, combination into single cross section grid
 - Still significant code development required, but starting to look as if we will be able to reproduce well the cross section at NNLO
- As the APPLfast-NNLO project comes to fruition, will hopefully be able to immediately take full advantage of the future developments of the NNLOJET code.
- The next few years will be an extremely interesting time for grid technologies and may start to allow genuine NNLO fits using the LHC data, and then, **What larks!**