Recent developments with APPLfast-NNLO for the LHC, and grid distribution

Mark Sutton
The University of Sussex

Daniel Britzger[†], Claire Gwenlan[‡], Alex Huss[§]
Tom Morgan[*], Klaus Rabbertz[a], Mark Sutton[‡]
Max Planck[†], Oxford[‡], CERN[§], IPPP Durham[*], Karlsruhe[a]

9th April 2019

Work supported by the IPPP Associateship program

Preface

- The excellent performance of the LHC is leading to ever higher precision measurements
- Recent significant developments in the precision of QCD calculations also significantly improve the description of the LHC cross section
- However, such calculations have significantly increased CPU requirements, so that more than ever, fast grid techniques such as fastNLO and APPLgrid are required to be able to use these data and calculations in QCD fits
  - In the past NNLO k-factors have been the standard, but with the APPLfast interface, using the full NNLO calculations becomes possible

- Increasingly, all PDF fitters are reliant on these fast interpolation grids from APPLgrid and fastNLO, but the extensive CPU required means that we, as a communittee would benefit greatly from more extensive sharing of the products of our grid calculations
- Into the fray, enters the Ploughshare project - a grid sharing which will be the primary source for the distribution of all the NNLO grids developed as part of the APPLfast project…

The APPLfast project

- Began as a joint project between the fastNLO, APPLgrid and NNLOJET developers at QCD@LHC
- Interface between the NNLOJET Code and both the APPLgird and fastNLO filling routines
- Original aim was to interface with the NNLOJET code in a minimally intrusive way
  - Since then the interface has developed in a number of ways to improve in the efficiency - has included developments to fastNLO, APPLgrid and NNLOJET

- Typically many hundreds of separate processes at NNLO
  - Automated generation of subprocess contributions and book keeping
  - Sub process combinations which share the same input partons are automatically combined
  - Many processes generated for the same phase space point, so weights are cached before filling the grids to reduce duplication of calculation of interpolating coefficients

- Grid generation typically in three stages …
  - VEGAS warmup - just run NNLOJET (no grid filling), optimise and fix generation phase space
  - Grid warmup - run NNLOJET with reduced phase space but filling grids to allow optimisation of grid node phase space
  - Grip production run - run NNLOJET with full statistics and full grid filling

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	$(\bar{u},\bar{u})+(\bar{c},\bar{c})$

Master formula for hadron-hadron convolution at NLO

- The master cross section for the convolution at the central scale just involved the convolution of the weight grids for each order with the sub process combination vector …

$$W(\xi_R,\xi_F) = \sum_{l=0}^{n_{\rm sub}-1} \sum_{i_{y_1}} \sum_{i_{y_2}} \sum_{i_\tau} \Bigg\{ \left(\frac{\alpha_s(\xi_R^2 Q^{2(i_\tau)})}{2\pi}\right)^{p_{\rm LO}} \times W^{(p_{\rm LO})(l)}_{i_{y_1},i_{y_2},i_\tau} F^{(l)}\big(x_1^{(i_{y_1})}, x_2^{(i_{y_1})}, \xi_F^2 Q^{2(i_\tau)}\big)$$
$$+ \left(\frac{\alpha_s(\xi_R^2 Q^{2(i_\tau)})}{2\pi}\right)^{p_{\rm NLO}} \times \Big[\big(W^{(p_{\rm NLO})(l)}_{i_{y_1},i_{y_2},i_\tau} \times F^{(l)}\big(x_1^{(i_{y_1})}, x_2^{(i_{y_1})}, \xi_F^2 Q^{2(i_\tau)}\big)\ \Big]\Bigg\}$$

- In practice the number of sub processes, nodes etc are different for each order
- At NLO the scale variation just requires the additional log terms times the LO weight grid including the convolution of the splitting function with the PDF for the factorisation scale dependent terms …

$$\sum_{l=0}^{n_{\rm sub}-1} \sum_{i_{y_1}} \sum_{i_{y_2}} \sum_{i_\tau} \Bigg\{ \left(\frac{\alpha_s(\xi_R^2 Q^{2(i_\tau)})}{2\pi}\right)^{p_{\rm NLO}} \times \Big[\ 2\pi\beta_0 p_{\rm LO} \ln\xi_R^2 W^{(p_{\rm LO})(l)}_{i_{y_1},i_{y_2},i_\tau}\big) \times F^{(l)}\big(x_1^{(i_{y_1})}, x_2^{(i_{y_1})}, \xi_F^2 Q^{2(i_\tau)}\big)$$
$$- \ln\xi_F^2 W^{(p_{\rm LO})(l)}_{i_{y_1},i_{y_2},i_\tau}\big(F^{(l)}_{q_1\to P_0\otimes q_1}\big(x_1^{(i_{y_1})}, x_2^{(i_{y_1})}, \xi_F^2 Q^{2(i_\tau)}\big) + F^{(l)}_{q_2\to P_0\otimes q_2}\big(x_1^{(i_{y_1})}, x_2^{(i_{y_1})}, \xi_F^2 Q^{2(i_\tau)}\big)\big)\Big]\Bigg\}$$

- At NNLO the central scale is hardly more complicated - just the extra term for the NNLO weight grid, but the scale variations are more involved …

NNLO scale variation

- NNLO cross section calculation at central scale …

$$\sigma(\mu_{R_0}, \mu_{F_0}, \alpha_s(\mu_{R_0})) = \left(\frac{\alpha_s(\mu_{R_0})}{2\pi}\right)^2 \hat{\sigma}^{(0)}_{ij} \otimes f_i(\mu_{F_0}) \otimes f_j(\mu_{F_0}) + \left(\frac{\alpha_s(\mu_{R_0})}{2\pi}\right)^3 \hat{\sigma}^{(1)}_{ij} \otimes f_i(\mu_{F_0}) \otimes f_j(\mu_{F_0}) + \left(\frac{\alpha_s(\mu_{R_0})}{2\pi}\right)^4 \hat{\sigma}^{(2)}_{ij} \otimes f_i(\mu_{F_0}) \otimes f_j(\mu_{F_0}) + \mathcal{O}(\alpha_s^5)\,.$$

NNLO contribution

- Full NNLO scale variation includes multiple log terms and convolutions with the splitting function
- **APPLgrid …**
  - Stores only the three grids for LO, NLO and NNLO
  - Calculates the log terms and convolutions with splitting functions dynamically (using hoppet) as required
    - Computationally more complex, but smaller grids - fewer terms in the a posteriori summation
- **fastNLO …**

$$\omega(\mu_R,\mu_F) = \omega_0 + \log(\mu_R^2)\omega_R + \log(\mu_F^2)\omega_F + \log^2(\mu_R^2)\omega_{RR} + \log^2(\mu_F^2)\omega_{FF} + \log(\mu_R^2)\log(\mu_F^2)\omega_{RF}$$

log's for NLO — additional log's in NNLO

  - Stores additional grids for each of the coefficients for the logs
    - 6 grids for the LO coefficients (NNLO contribution is NNLO + NLO × log + LO × log log etc)
    - 3 grids for the NLO coefficients
    - 1 grid for the NNLO coefficients
  - Less complex, but ~ 3 times larger grids for a single scale, more terms in the a posteriori convolution, but more straightforward when storing multiple scales

additional NLO terms

$$+L_R \left(\frac{\alpha_s(\mu_R)}{2\pi}\right)^3 2\beta_0\,\hat{\sigma}^{(0)}_{ij} \otimes f_i(\mu_F) \otimes f_j(\mu_F)$$

$$+L_F \left(\frac{\alpha_s(\mu_R)}{2\pi}\right)^3 \Big[-\hat{\sigma}^{(0)}_{ij} \otimes f_i(\mu_F) \otimes \left(P^{(0)}_{jk} \otimes f_k(\mu_F)\right) - \hat{\sigma}^{(0)}_{ij} \otimes \left(P^{(0)}_{ik} \otimes f_k(\mu_F)\right) \otimes f_j(\mu_F)\Big]$$

additional NNLO terms

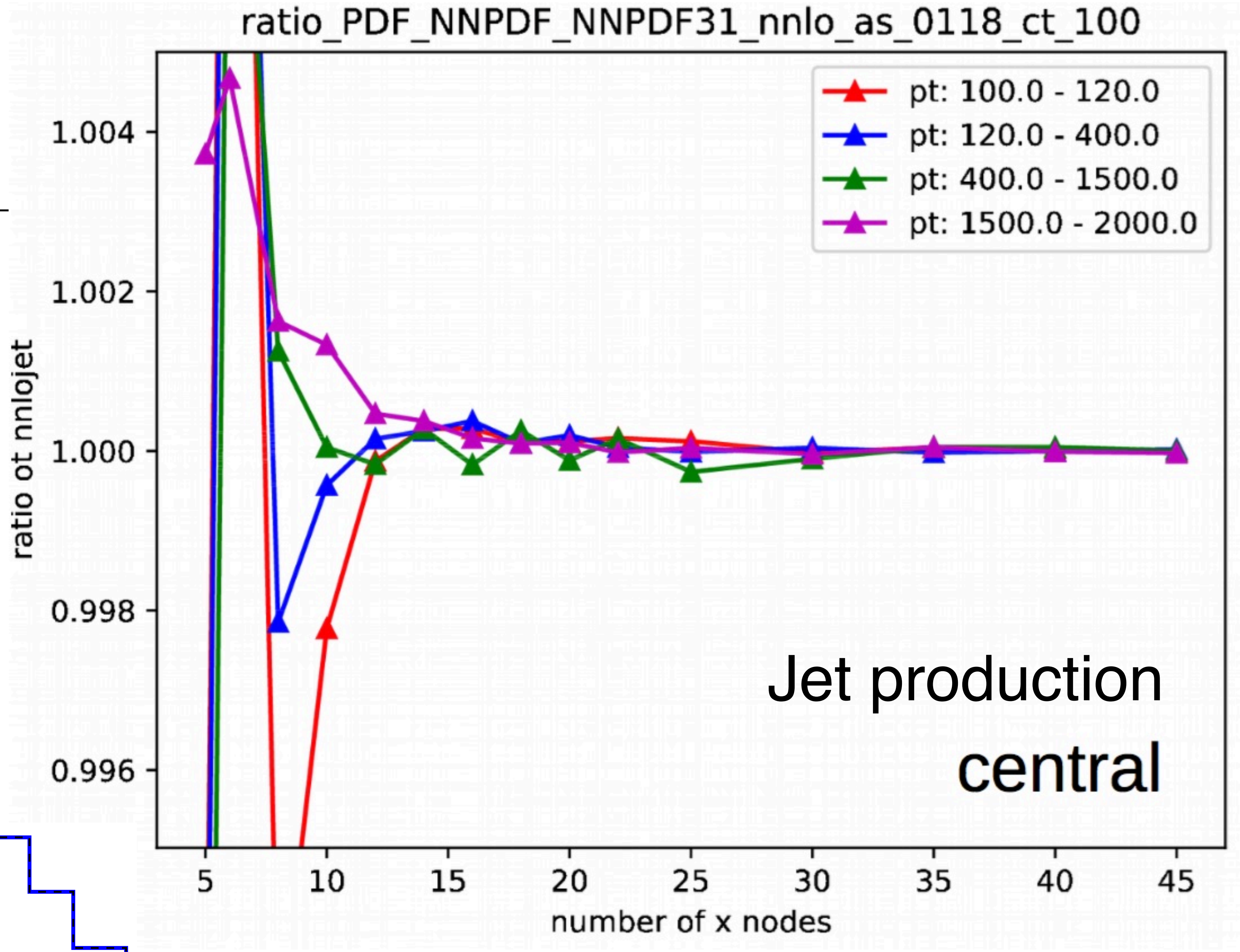
$$+L_R \left(\frac{\alpha_s(\mu_R)}{2\pi}\right)^4 \left(3\beta_0\,\hat{\sigma}^{(1)}_{ij} + 2\beta_1\,\hat{\sigma}^{(0)}_{ij}\right) \otimes f_i(\mu_F) \otimes f_j(\mu_F)$$

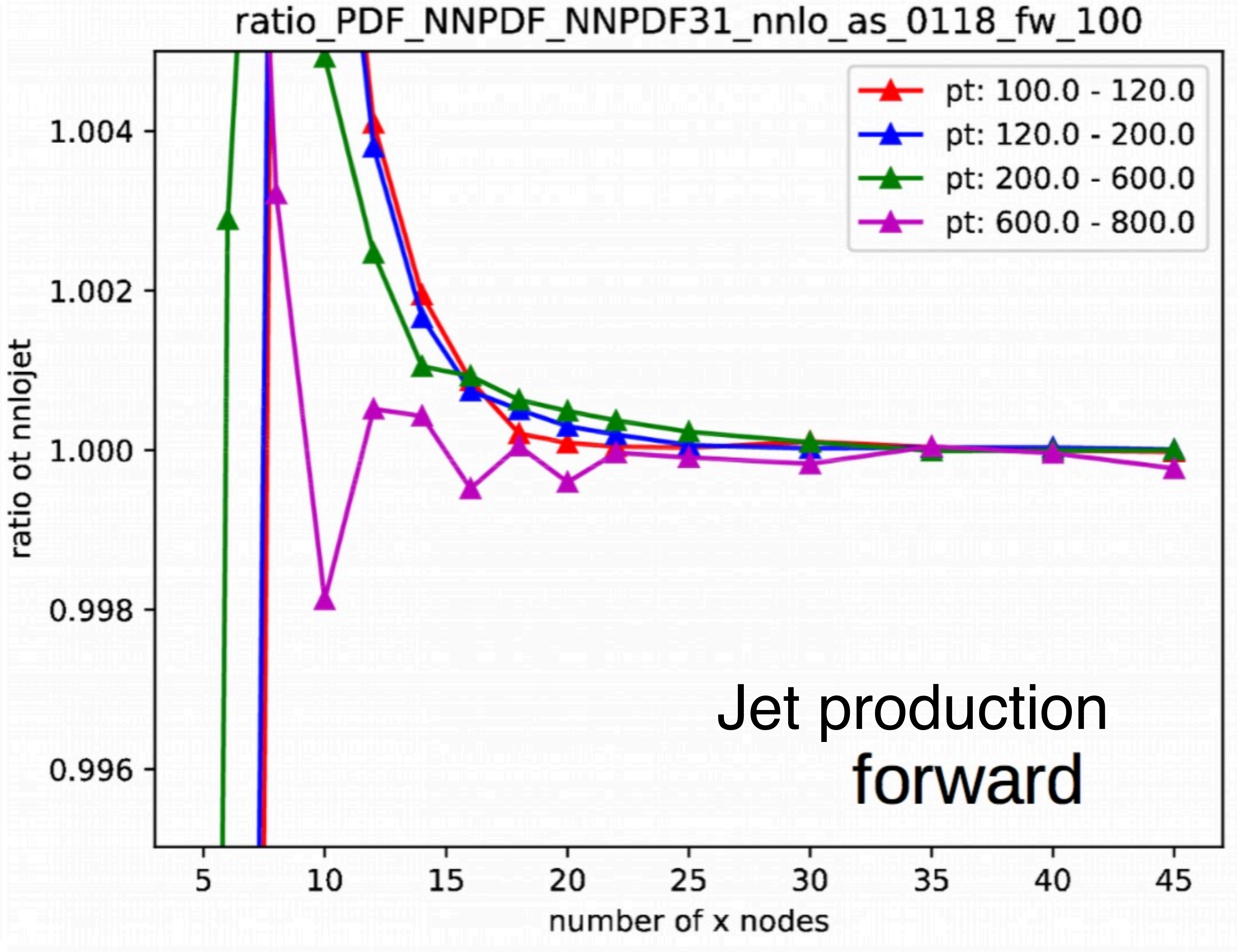
$$+L_R^2 \left(\frac{\alpha_s(\mu_R)}{2\pi}\right)^4 3\beta_0^2\,\hat{\sigma}^{(0)}_{ij} \otimes f_i(\mu_F) \otimes f_j(\mu_F)$$

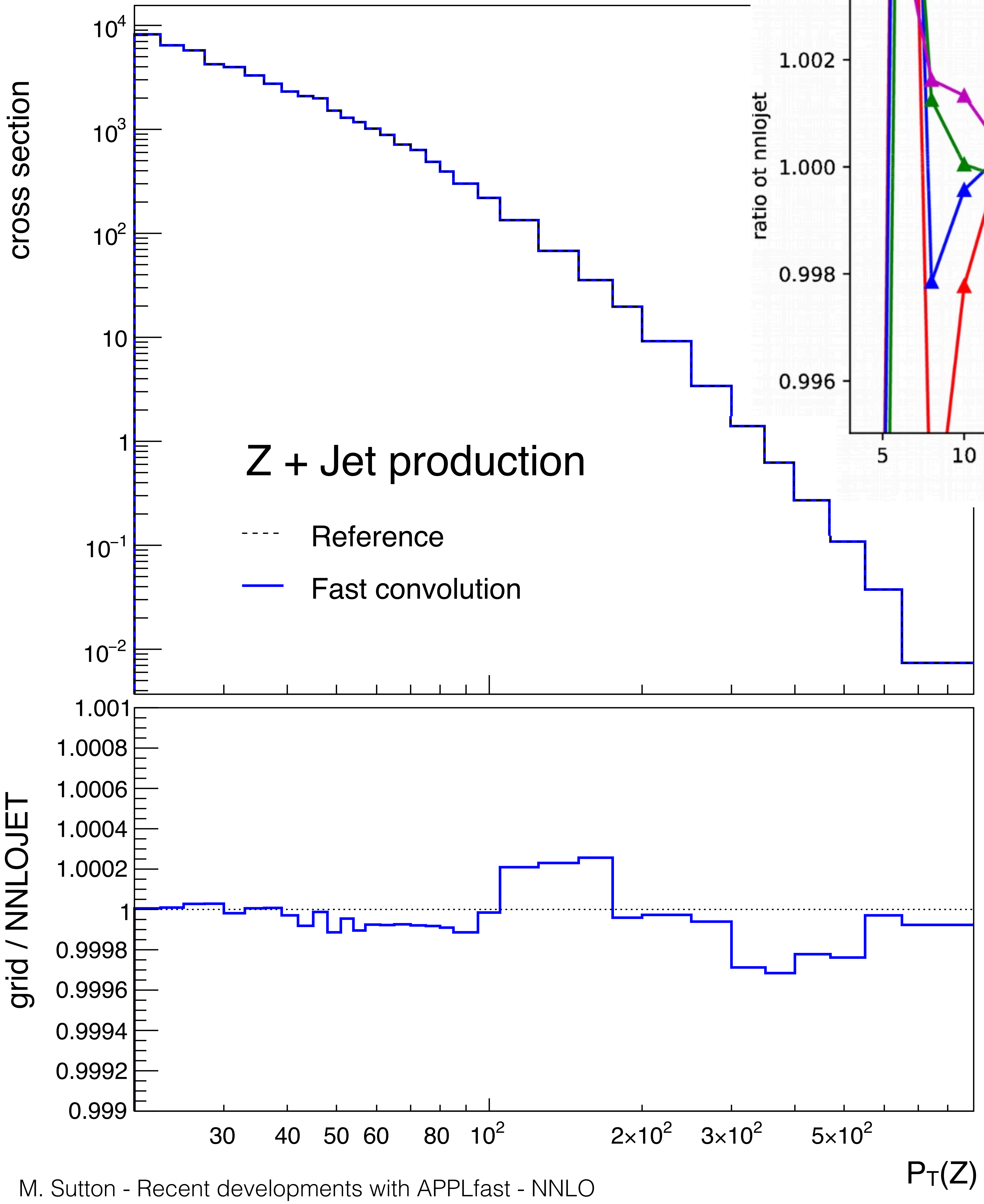
$$+L_F \left(\frac{\alpha_s(\mu_R)}{2\pi}\right)^4 \Big[-\hat{\sigma}^{(1)}_{ij} \otimes f_i(\mu_F) \otimes \left(P^{(0)}_{jk} \otimes f_k(\mu_F)\right) - \hat{\sigma}^{(1)}_{ij} \otimes \left(P^{(0)}_{ik} \otimes f_k(\mu_F)\right) \otimes f_j(\mu_F) - \hat{\sigma}^{(0)}_{ij} \otimes f_i(\mu_F) \otimes \left(P^{(1)}_{jk} \otimes f_k(\mu_F)\right) - \hat{\sigma}^{(0)}_{ij} \otimes \left(P^{(1)}_{ik} \otimes f_k(\mu_F)\right) \otimes f_j(\mu_F)\Big]$$

$$+L_F^2 \left(\frac{\alpha_s(\mu_R)}{2\pi}\right)^4 \Big[\hat{\sigma}^{(0)}_{ij} \otimes \left(P^{(0)}_{ik} \otimes f_k(\mu_F)\right) \otimes \left(P^{(0)}_{jl} \otimes f_l(\mu_F)\right) + \frac{1}{2}\hat{\sigma}^{(0)}_{ij} \otimes f_i(\mu_F) \otimes \left(P^{(0)}_{jk} \otimes P^{(0)}_{kl} \otimes f_l(\mu_F)\right) + \frac{1}{2}\hat{\sigma}^{(0)}_{ij} \otimes \left(P^{(0)}_{ik} \otimes P^{(0)}_{kl} \otimes f_l(\mu_F)\right) \otimes f_j(\mu_F) + \frac{1}{2}\beta_0\,\hat{\sigma}^{(0)}_{ij} \otimes f_i(\mu_F) \otimes \left(P^{(0)}_{jk} \otimes f_k(\mu_F)\right) + \frac{1}{2}\beta_0\,\hat{\sigma}^{(0)}_{ij} \otimes \left(P^{(0)}_{ik} \otimes f_k(\mu_F)\right) \otimes f_j(\mu_F)\Big]$$

$$+L_F L_R \left(\frac{\alpha_s(\mu_R)}{2\pi}\right)^4 \Big[-3\beta_0\,\hat{\sigma}^{(0)}_{ij} \otimes f_i(\mu_F) \otimes \left(P^{(0)}_{jk} \otimes f_k(\mu_F)\right) - 3\beta_0\,\hat{\sigma}^{(0)}_{ij} \otimes \left(P^{(0)}_{ik} \otimes f_k(\mu_F)\right) \otimes f_j(\mu_F)\Big]$$







Grid closure

- How well the grids reproduce the full calculation depends entirely on the quality of the interpolation
    - Does not improve with higher statistics - precision determined by the precision of the interpolation for each individual weight
- More grid nodes improves the interpolation - particularly in x, Grid size scales with N_x^2
- A little over half of the time for the standard convolution from to the evaluation of the PDF on the grid nodes
    - Scales with $2 \times N_x$
- NB: don't actually store grids in x and Q^2, use transformed variables, eg y = log(x) + k(1-x)

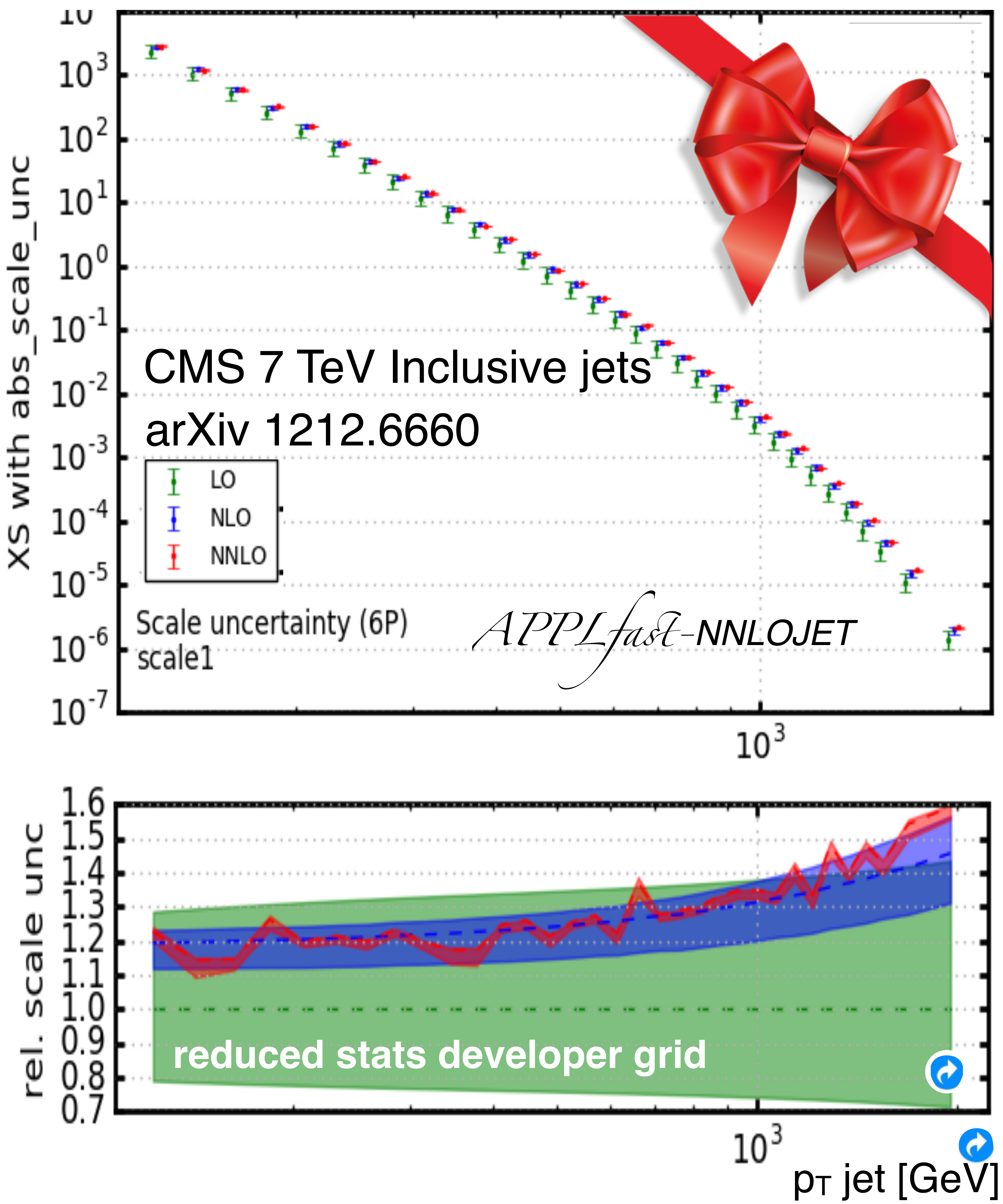
APPLfast test grids

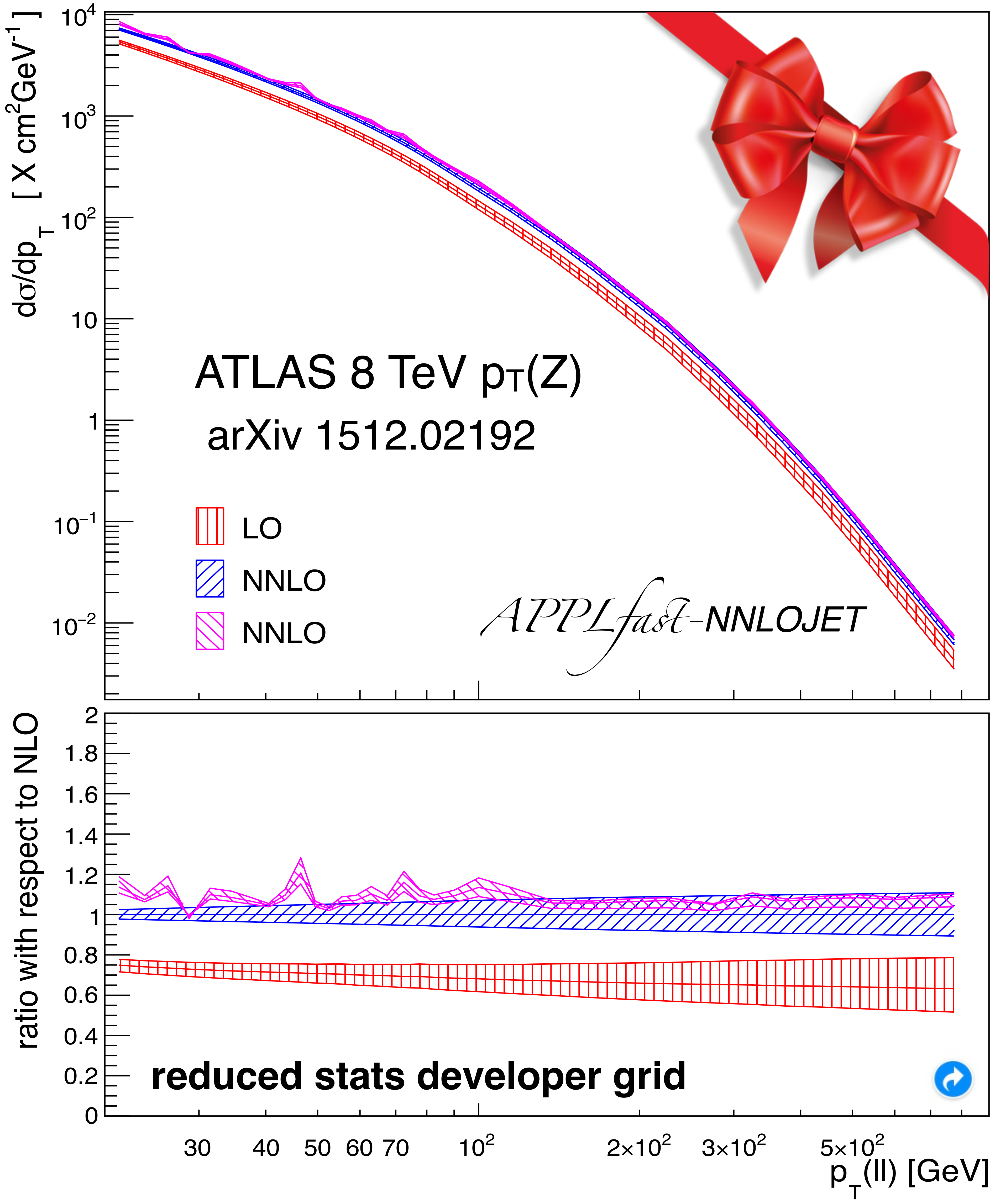
Experiment	Collision type	Energy	Process	Calculation	arxiv	link to paper details and grids
ATLAS	pp	7 TeV	1jet-R06-dev-ap	NNLOJET	1410.8857	applfast-atlas-1jet-r06-dev-ap-arxiv-1410.8857
						applfast-atlas-1jet-r06-dev-ap-arxiv-1410.8857-xsec00
						applfast-atlas-1jet-r06-dev-ap-arxiv-1410.8857-xsec00
ATLAS	pp	7 TeV	1jet-R06-dev-fn	NNLOJET	1410.8857	applfast-atlas-1jet-r06-dev-fn-arxiv-1410.8857
						applfast-atlas-1jet-r06-dev-fn-arxiv-1410.8857-xsec00
ATLAS	pp	8 TeV	ZJ-dev-ap	NNLOJET	1512.02192	applfast-atlas-zj-dev-ap-arxiv-1512.02192
						applfast-atlas-zj-dev-ap-arxiv-1512.02192-xsec000.root
						applfast-atlas-zj-dev-ap-arxiv-1512.02192-xsec001.root
						applfast-atlas-zj-dev-ap-arxiv-1512.02192-xsec002.root
ATLAS	pp	8 TeV	ZJ-dev-fn	NNLOJET	1512.02192	applfast-atlas-zj-dev-fn-arxiv-1512.02192
						applfast-atlas-zj-dev-fn-arxiv-1512.02192-xsec000.tab.
						applfast-atlas-zj-dev-fn-arxiv-1512.02192-xsec001.tab.
						applfast-atlas-zj-dev-fn-arxiv-1512.02192-xsec002.tab.
CMS	pp	7 TeV	1jet-ptj-dev-ap	NNLOJET	1212.6660	applfast-cms-1jet-ptj-dev-ap-arxiv-1212.6660
						applfast-cms-1jet-ptj-dev-ap-arxiv-1212.6660-xsec000.
						applfast-cms-1jet-ptj-dev-ap-arxiv-1212.6660-xsec001.
CMS	pp	7 TeV	1jet-ptj-dev	NNLOJET	1212.6660	applfast-cms-1jet-ptj-dev-arxiv-1212.6660
						applfast-cms-1jet-ptj-dev-arxiv-1212.6660-xsec000.tab
H1	ep	0.319 TeV	ptj-dev-ap	NNLOJET	1406.4709	applfast-h1-ptj-dev-ap-arxiv-1406.4709
						applfast-h1-ptj-dev-ap-arxiv-1406.4709-xsec000.root
						applfast-h1-ptj-dev-ap-arxiv-1406.4709-xsec001.root
H1	ep	0.319 TeV	ptj-dev	NNLOJET	1406.4709	applfast-h1-ptj-dev-arxiv-1406.4709
						applfast-h1-ptj-dev-arxiv-1406.4709-xsec000.tab.gz
						applfast-h1-ptj-dev-arxiv-1406.4709-xsec001.tab.gz

- Applfast has made available several test grids, on the ploughshare platform …
- These consist of a small number of processes - Z^0 transverse momenta, inclusive jets, and DIS jets
- These grids are all fully functional, but significantly have intentionally reduced statistics, or missing contributions for the NNLO calculation parts
  - Mostly, datasets with …
    - “ap” are applgrid (.root files),
    - “fn” are are fastnlo (.tag.gz files)

- They are provided for developers to develop and test functionality, but **should not** be used for physics

Some example processes

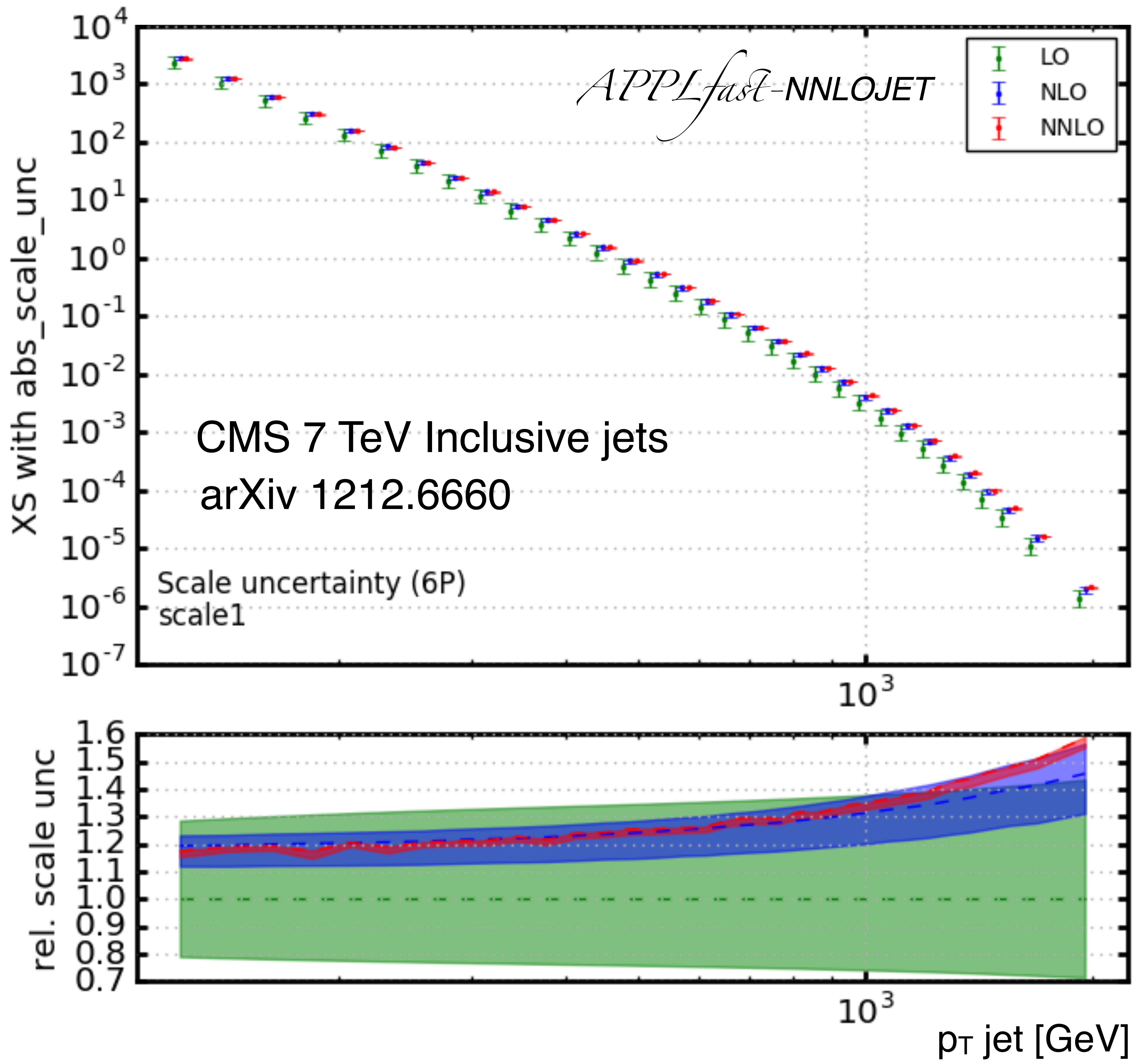


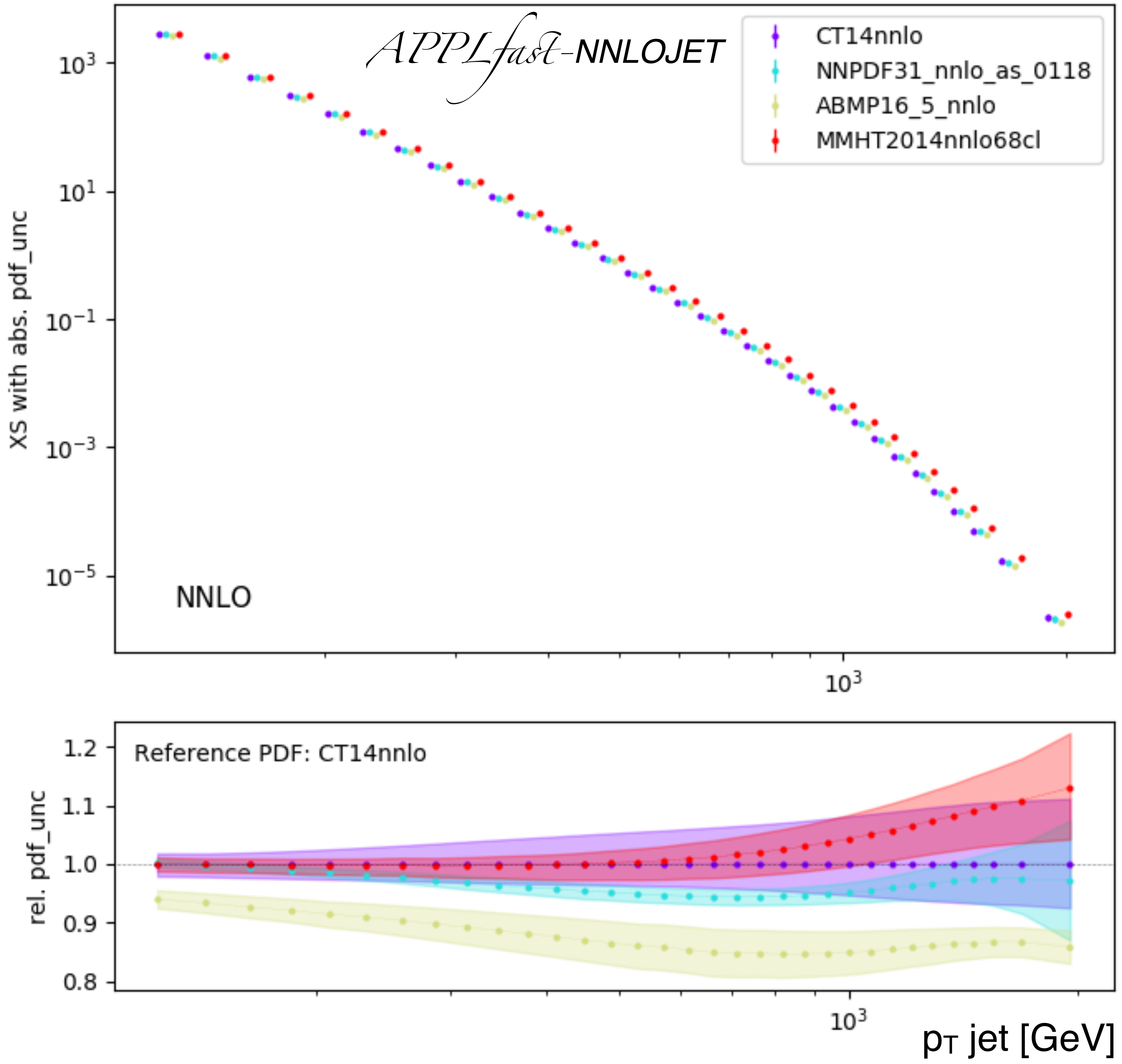


- Available as both APPLgrid and fastNLO tables
  - CMS and ATLAS inclusive jets at 7 TeV
  - ATLAS Z(p_T) at 8 TeV

- **NB: Note the reduced statistics for the NNLO contributions**

To look forward to …



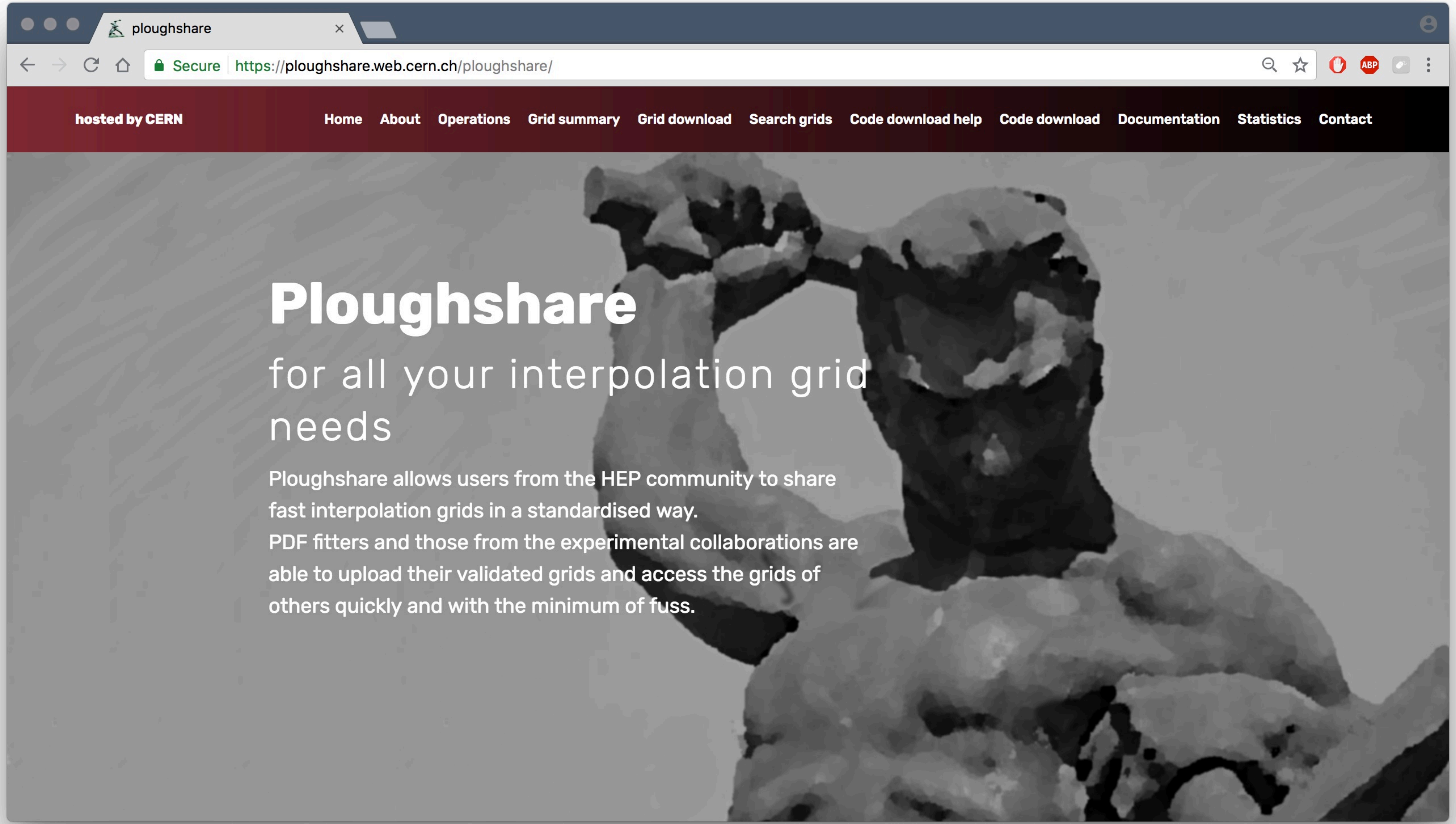


- Of course with more reasonable statistics the cross sections look very good - the grid closure is typically **around 0.1 per mille** file for reasonable grid sizes
- Scale variations and evaluation with the full error sets for multiple PDF sets is very fast

Scale variation - detailed variations for single bins

- Horizontal axis - renormalisation scale
- Shaded bands - factorisation scale variations from original NNLOJET paper
- Unshaded, hatched bands - a-posterioi scale valuation using the grids
- Scale variations cross checked between both APPLgrid and fastNLO tables - in very good agreement with the full NNLOJET calculations

ploughshare.web.cern.ch



Configuration continued

- In the “extra” lookup, the lookup information must be pairs of strings, containing
  - The name of the grid file, additional information about that file, if Table in paper is specified, the link to hepdata for that table will be included

- After upload, all the grid files are decoded and automatically renamed into the standard ploughshare naming convention, namely …

atlas-atlas-dijets-arxiv-1112.6297

preprint id

short info

experiment

submitting group

- Database also distributes a standard combined tgz file containing these grids with a .tgz extension
- Grids within the tgz file …

atlas-atlas-dijets-arxiv-1112.6297-xsec000.root

Index of the cross section within the tgz file

Full analysis records

Measurement of inclusive jet and dijet production in pp collisions at $\sqrt{s} = 7$ TeV using the ATLAS detector

Inclusive jet and dijet cross sections have been measured in proton-proton collisions at a centre-of-mass energy of 7 TeV using the ATLAS detector at the Large Hadron Collider. The cross sections were measured using jets clustered with the anti-kT algorithm with parameters R=0.4 and R=0.6. These measurements are based on the 2010 data sample, consisting of a total integrated luminosity of 37 inverse picobarns. Inclusive jet double-differential cross sections are presented as a function of jet transverse momentum, in bins of jet rapidity. Dijet double-differential cross sections are studied as a function of the dijet invariant mass, in bins of half the rapidity separation of the two leading jets. The measurements are performed in the jet rapidity range |y|<4.4, covering jet transverse momenta from 20 GeV to 1.5 TeV and dijet invariant masses from 70 GeV to 5 TeV. The data are compared to expectations based on next-to-leading order QCD calculations corrected for non-perturbative effects, as well as to next-to-leading order Monte Carlo predictions. In addition to a test of the theory in a new kinematic regime, the data also provide sensitivity to parton distribution functions in a region where they are currently not well-constrained.

journal: Phys.Rev. D86 (2012) 014022 (doi: 10.1103/PhysRevD.86.014022)
arxiv: 1112.6297
inspire: https://inspirehep.net/record/1082936
HepData: https://hepdata.net/record/ins1082936

link to full tarball

links to specific hep data table

Experiment	Physics process	Beam energy	Calculation	direct link
ATLAS	pp	7 TeV	NLOjet++	atlas-atlas-dijets-arxiv-1112.6297 tarball
				atlas-atlas-dijets-arxiv-1112.6297-xsec000.root :: Table 19 : d2sigma/dm_{12}dy* [pb/TeV], Anti-kT R=0.4, 0.0<y*<0.5
				atlas-atlas-dijets-arxiv-1112.6297-xsec001.root :: Table 20 : d2sigma/dm_{12}dy* [pb/TeV], Anti-kT R=0.4, 0.5<y*<1.0
				atlas-atlas-dijets-arxiv-1112.6297-xsec002.root :: Table 21 : d2sigma/dm_{12}dy* [pb/TeV], Anti-kT R=0.4, 1.0<y*<1.5
				atlas-atlas-dijets-arxiv-1112.6297-xsec003.root :: Table 22 : d2sigma/dm_{12}dy* [pb/TeV], Anti-kT R=0.4, 1.5<y*<2.0
				atlas-atlas-dijets-arxiv-1112.6297-xsec004.root :: Table 23 : d2sigma/dm_{12}dy* [pb/TeV], Anti-kT R=0.4, 2.0<y*<2.5
				atlas-atlas-dijets-arxiv-1112.6297-xsec005.root :: Table 24 : d2sigma/dm_{12}dy* [pb/TeV], Anti-kT R=0.4, 2.5<y*<3.0
				atlas-atlas-dijets-arxiv-1112.6297-xsec006.root :: Table 26 : d2sigma/dm_{12}dy* [pb/TeV], Anti-kT R=0.4, 3.5<y*<4.0

links to individual grids

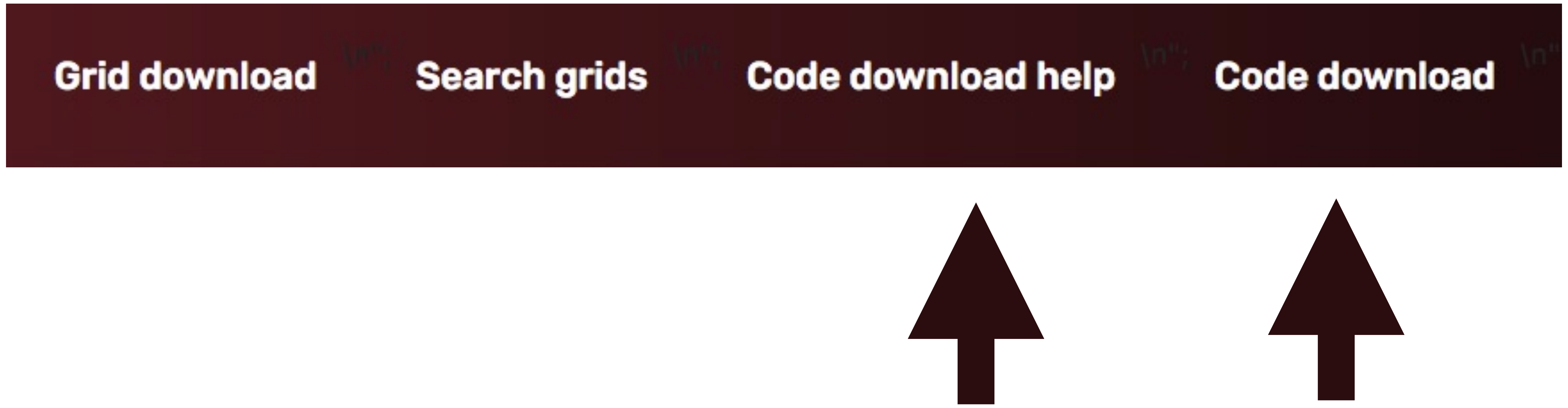
optional grid desciption

- Full title and abstract
- Links to …
  - Journal paper (DOI)
  - Preprint
  - Inspire
  - HepData
  - Table with all available grids

- This information is determined automatically from the required preprint ID when you upload grids
- For grids with no corresponding paper, a dummy arrive number arxiv: 0000.00000 can be used
- Users will be able to provide an additional HTML fragment in the tgz file if they require
- In the case of no available preprint the HTML fragment will be used as the analysis record …

Grid download library - basic example

- A few lines in the code will create the ploughshare instance
- Any requested grids are automatically downloaded if they are not already present locally …

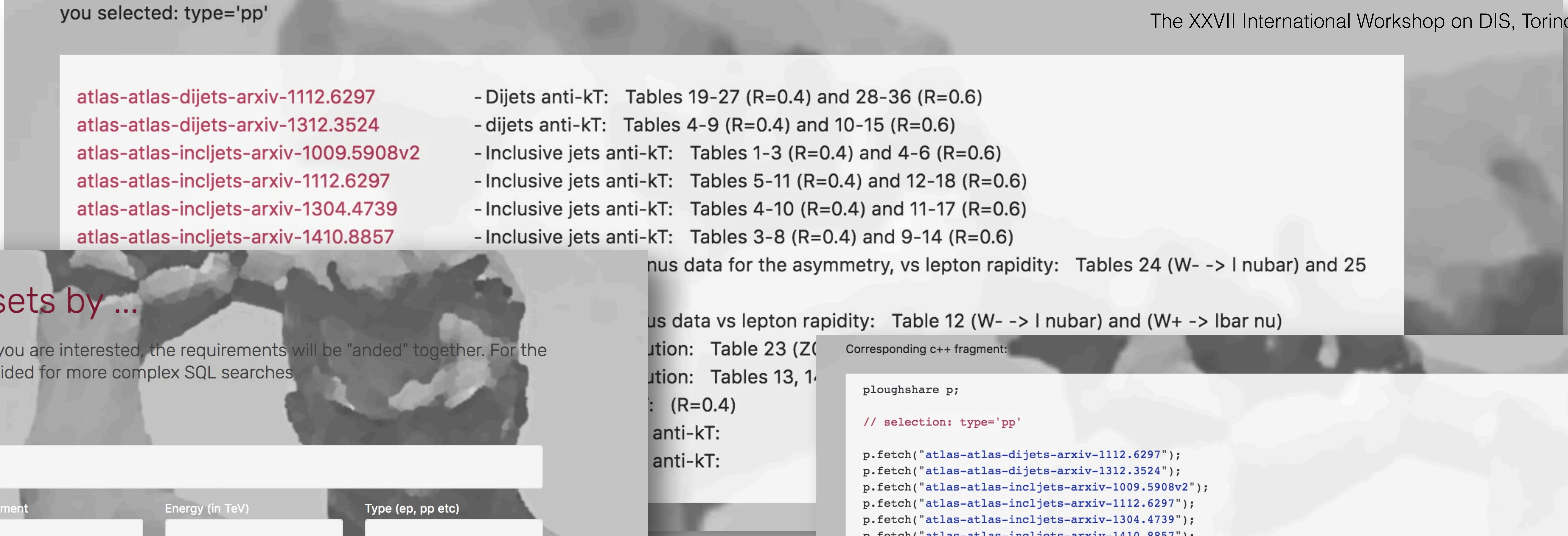


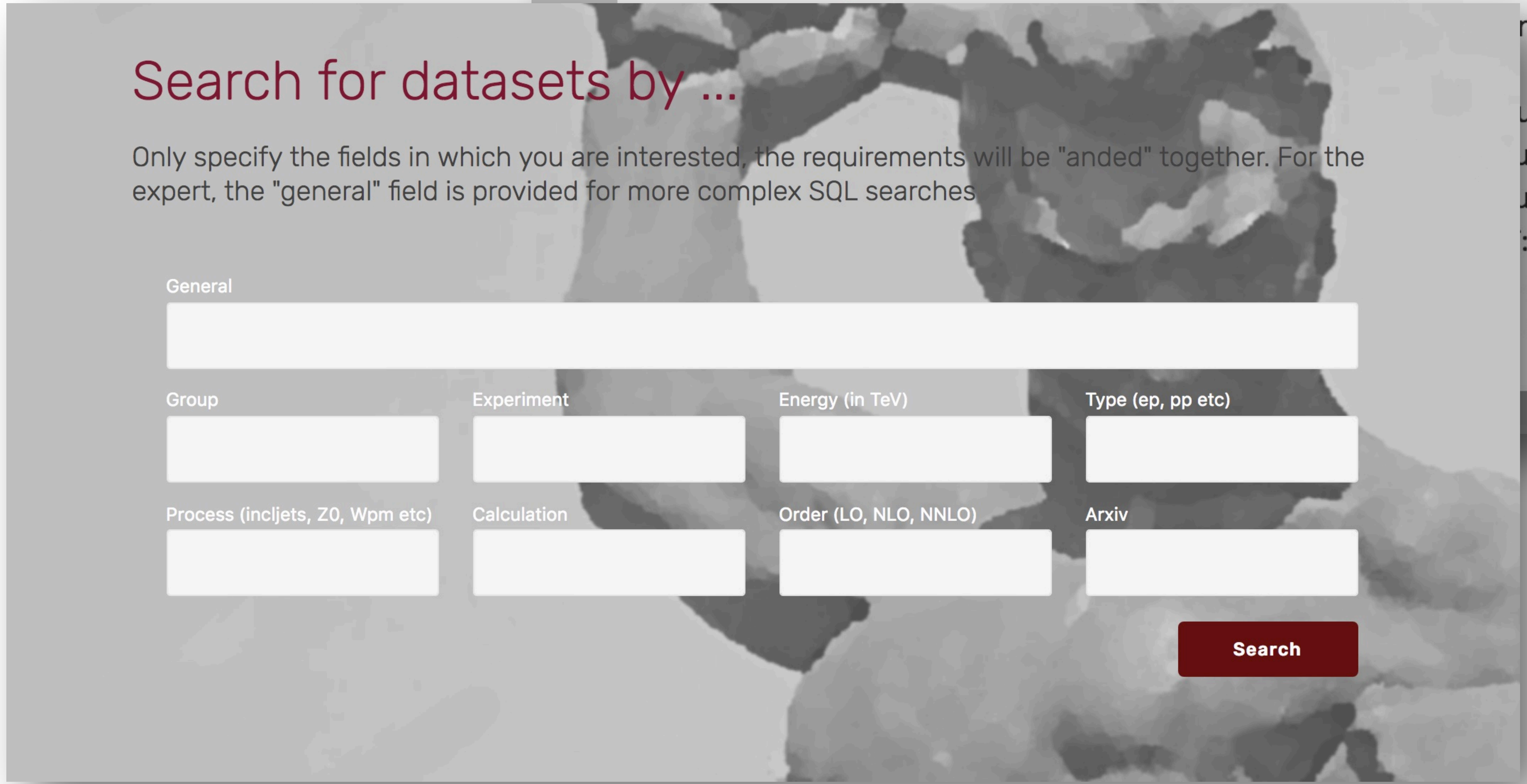

```
#include "ploughshare/ploughshare.h"
#include "appl_grid/appl_grid.h"


int main() {

  ploughshare p;

  p.verbose(true);

  std::cout << "path: " << p.path() << std::endl;

  p.fetch( "atlas-atlas-z0-arxiv-1612.03016" );

  std::cout << "dataset path:  " <<  p.path("atlas-atlas-z0-arxiv-1612.03016")  << std::endl;
  std::cout << "dataset grids: " << p.grids("atlas-atlas-z0-arxiv-1612.03016") << std::endl;

  std::vector<std::string> grids = p.grids("atlas-atlas-z0-arxiv-1612.03016" );

  for ( size_t i=0 ; i<grids.size() ; i++ ) {

     appl::grid g(grids[i]);

  }

  return 0;
}
```

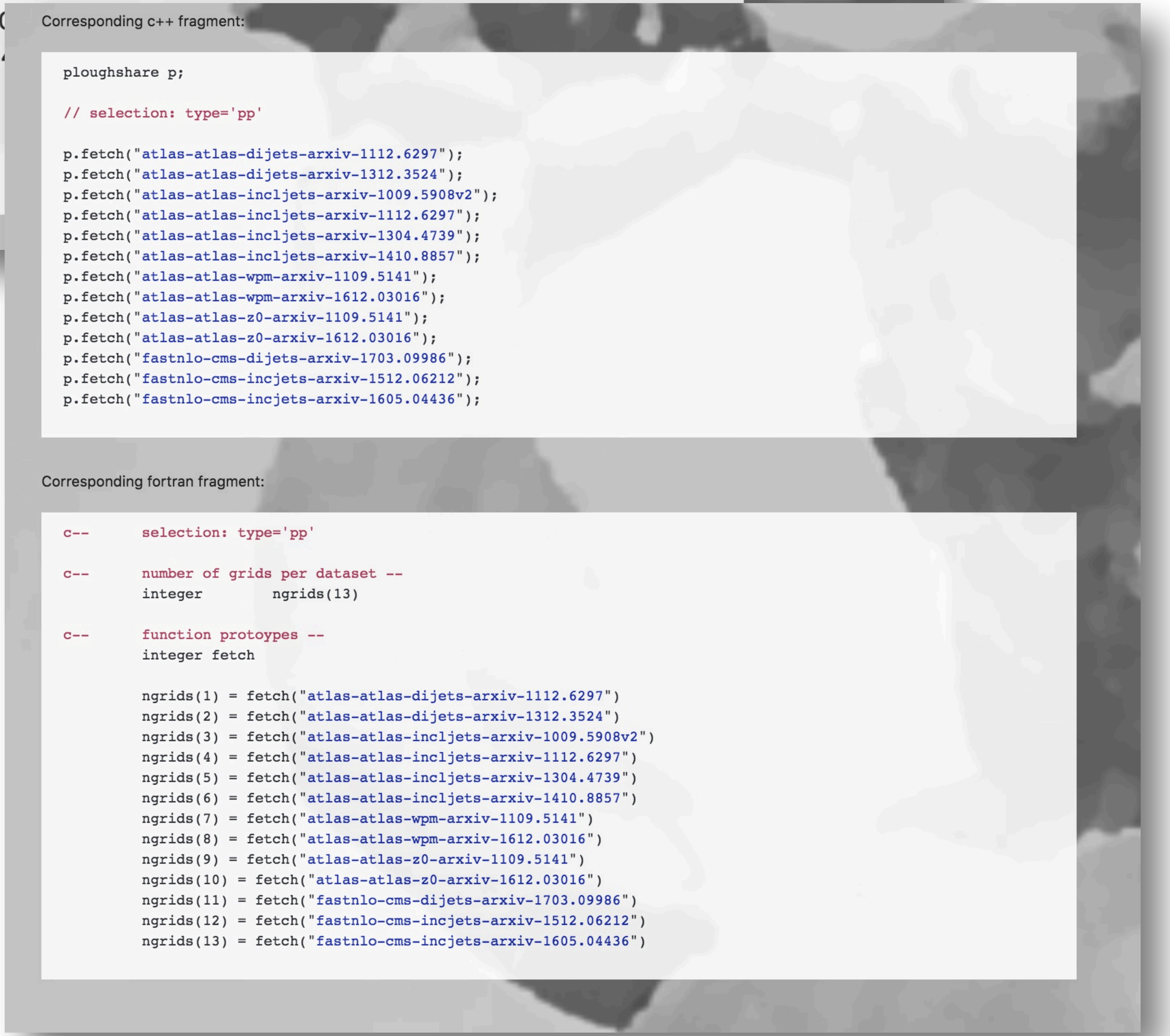
Grid download library - fortran interface

- Fortran interface acts the same way as the c++ interface but with a caveat
  - Since fortran can not store separate classes, the list of grids is not returned in a vector, but the path is returned by requesting each grid path individually

```
      subroutine pl

         implicit none

c--      ploughshare return stuff --
         character*1024 gridpath
         integer        ngrids

c--      function protoypes --
         integer fetch
         integer getnbins

c--      grid output paths - allow up to 20 grids in this example --
         character paths(20)*1024

c--      odds n ends --
         integer i, j

c--      this is the counter - it is automatically incremented
c--      whenever a new grid is opened ...
         integer tgrid

         tgrid = 0

c--      fetch the ploughshare dataset --
         ngrids = fetch( "atlas-atlas-dijets-arxiv-1312.3524" )

c--      read in all the grids --

         do i = 1, ngrids

c--         get the full path to specific grid in the data set --
            call getgridpath( "atlas-atlas-dijets-arxiv-1312.3524", i-1,
     &                        paths(i) )

            write (6,*) i, paths(i)

c--         read in the grids as usual --
c--         tgrid is automatically incremented when reading the grid
            call readgrid( tgrid, paths(i) )

         end do

      end
```







Search facility

- Detailed search facility is also available
- Provides basic list output with information
- Also provides output in forms **directly usable** as code fragments for either FORTRAN or C++
- May extend the functionality to include command line or C++ interfaces

Great expectations …

- Initial proof of concept code is fully functional
  - Fast convolution, including the full renormalisation and factorisation scale variations are complete and have been tested
- Some full statistics grids for LHC processes have been produced, others are underway
- Limited statistics fully NNLO grids for a number of LHC cross sections have been made available in both APPLgrid and fastNLO formats
- The next few years will be an extremely interesting time for grid technologies, already fits using full NNLO calculations are possible - see presentations from Daniel Britzger and Mandy Cooper-Sarkar
- Fits including HERA inclusive data, and jet data, and LHC cross sections at full NNLO precision should be here very soon, and then What larks!

Afterword: ploughshare notification twitter stream …

- ploughshare related notifications will be posted to the ploughshare twitter stream