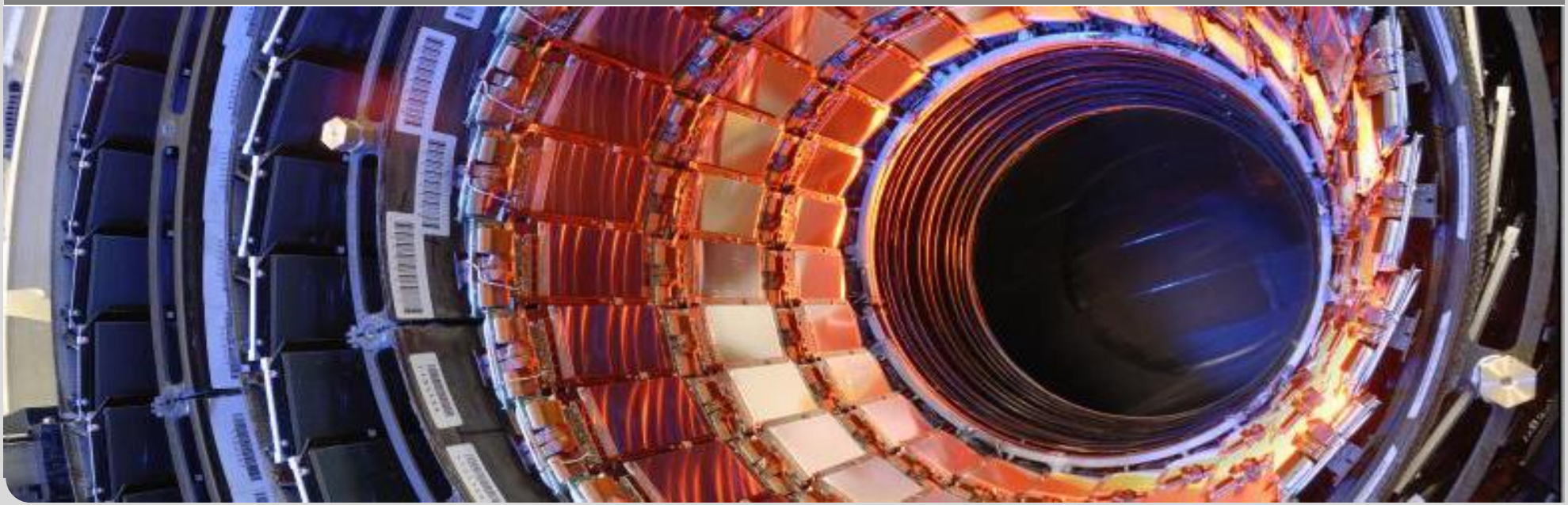
Karlsruhe Institute of Technology

NNLO interpolation grid production for Z+jet

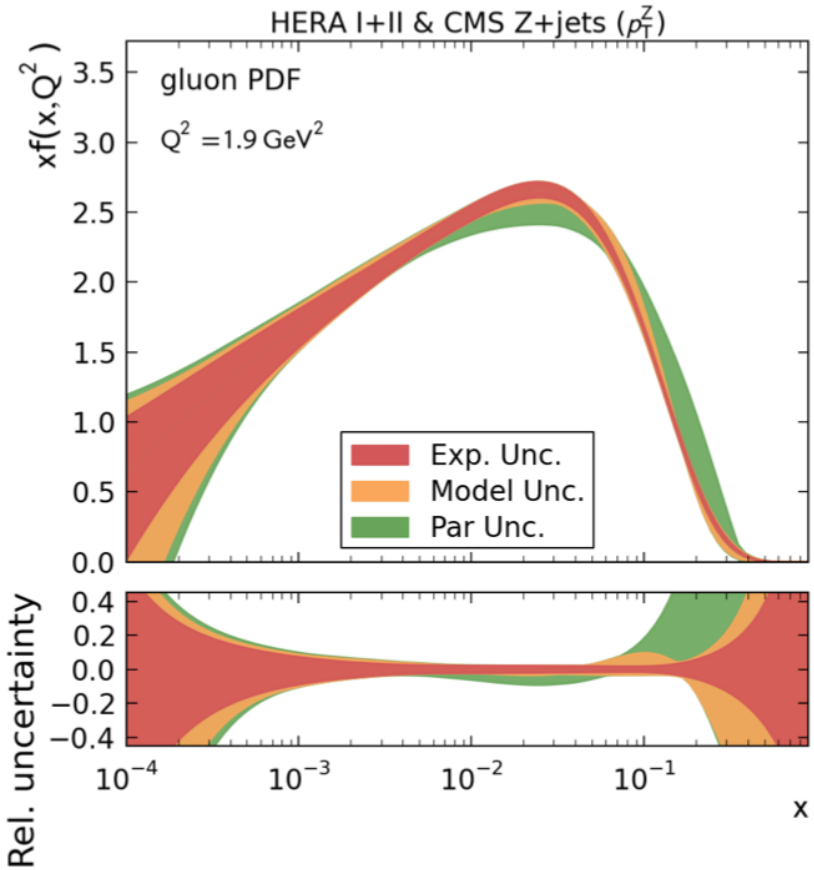
FSP CMS workshop Hamburg

Miguel Santos Correa, Klaus Rabbertz | 20. Sep 2018

INSTITUTE FOR EXPERIMENTAL PARTICLE PHYSICS (ETP) · KIT-FACULTY FOR PHYSICS



Parton Distribution Function (PDF)



Anna Friedel, EKP-2017-00030

- PDF indicates the chance to find a parton with momentum x in the proton
- Preparation of Z+Jet theory calculations for PDF fits
- **Goal:** Precise determination of PDFs, especially the gluon PDF

Interpolation of the cross section

Cross section for Proton-Proton-Collisions:

$$\sigma_{pp\to X} = \sum \int dx_1 dx_2 \underbrace{\alpha_s^n(\mu_r)}_{\text{coupling constant}} \underbrace{f_{a/p}(x_1,\mu_f) f_{b/p}(x_2,\mu_f)}_{\text{PDFs}} \underbrace{c_{a,b,n}(x_1,x_2,\mu_r,\mu_f)}_{\text{matrix element}}$$

Problem: Time intensive theory calculations must be repeated for variation in PDFs

Solution: Separation of PDFs from integrals through interpolation technique

$$\sigma_{pp\to X} = \sum \alpha_s^n(\mu_c) f_{a/p}(x_i,\mu_k) f_{b/p}(x_j,\mu_k) \underbrace{\widetilde{\sigma}_{n,i,j,k,a,b}}_{\text{interpolation grid}}$$

fastNLO .hepforge.org

Implemented in fastNLO and APPLgrid project applgrid.hepforge.org

Interpolation of the cross section

Approximate PDFs as a sum of eigenfunctions E_i at support nodes x_i

$$f_{a/p}(x) \approx \sum_i f_{a/p}(x_i) E_i(x)$$

Eigenfunctions are orthonormal and complete

$$E_i(x_k) = \delta_{ik}$$
$$\sum_i E_i(x) = 1$$

Interpolation coefficients are then determined by

$$d\widetilde{\sigma}_{n,i,j,k,a,b} = \int dx dx' E_i(x) E_j(x') E_k(\mu) d\sigma_{pp\to X}(x, x', \mu)$$

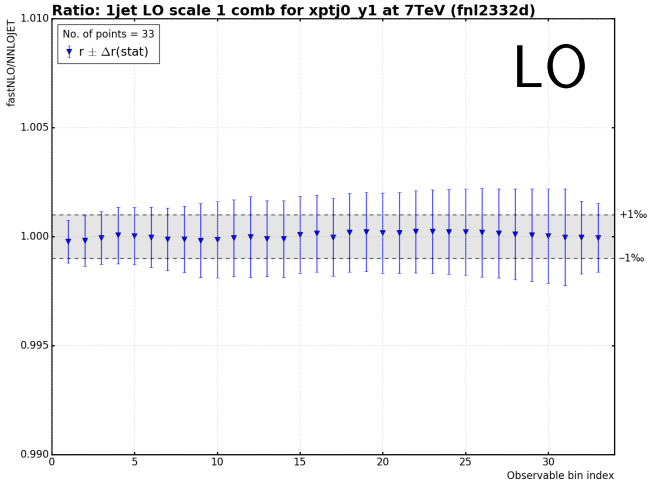
NNLOJET-Workflow

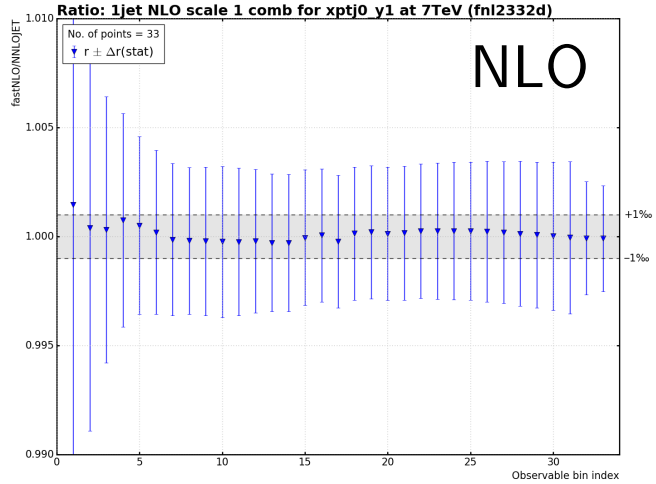
NNLOJET calculates cross sections in NNLO and can be used to fill APPLgrid and fastNLO grids with joint APPLfast interface
Z+Jets production in NNLOJET – *Gehrmann et al.,arXiv:1507.02850*

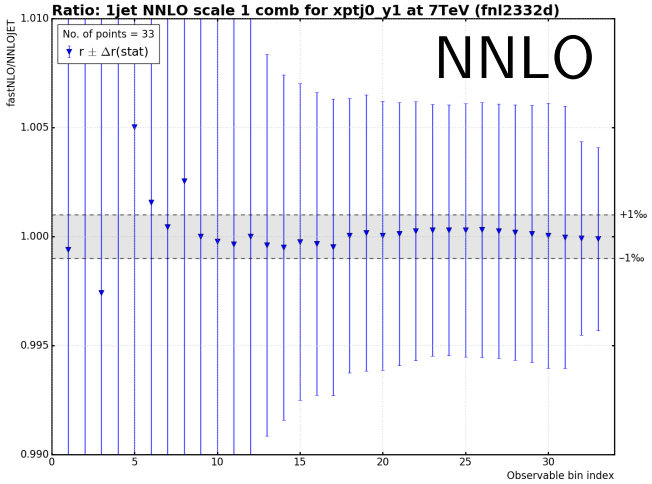
1. **NNLOJET Warm-up**
   Vegas-Integration
2. **Phasespace determination**
   Determine accessed the phasespace in x and Q^2
3. **Grid production**
   Thousands of parallel jobs
4. **Merging process**
   Statistical analysis of and combination of all interpolation grids

→ Analysis of results and comparison with experimental data

Closure: Interpolation/NNLOJET







Goal: Interpolation bias smaller than statistical error (blue bars)

Automation

- Workflow with NNLOJET and interpolation grids is tedious and time consuming
  $\rightarrow$ automation very desirable
- Reusable for different processes in the future
- In general: less human error and results are reproducible

Karlsruhe Institute of Technology

https://github.com/spotify/luigi

Luigi is a Python package that helps you build complex pipelines of batch jobs

Features:

- dependency resolution
- workflow management
- visualization
- handling failures
- command line integration
- and much more

law
luigi analysis workflow

RWTH AACHEN UNIVERSITY

KIT Karlsruhe Institute of Technology

https://github.com/riga/law

law is built on top of Luigi and adds abstractions for run locations, storage locations and software environments

Features:

- **Remote targets with automatic retries and local caching**
  WebDAV, HTTP, Dropbox, SFTP, all WLCG protocols (srm, xrootd, rfio, dcap, gsiftp, ...)
- **Automatic submission to batch systems from within tasks**
  HTCondor, LSF, gLite, ARC
- **Environment sandboxing, configurable on task level**
  Docker, Singularity, Sub-Shells

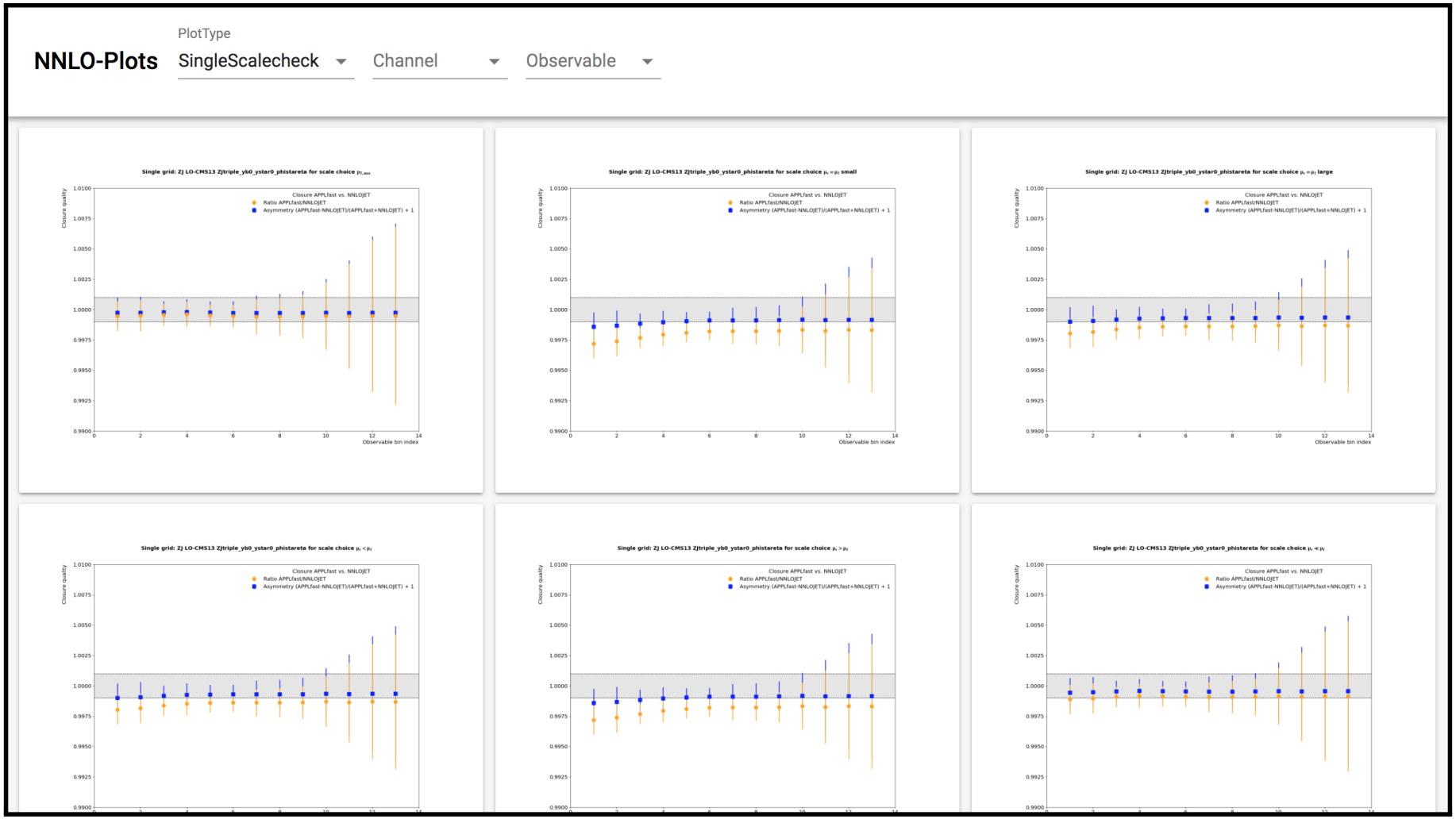
Features

- Pipeline works for any NNLOJET runcard and fastNLO steering files
- Automatic generation of plots for cross-checks
- Additional web UI for generated plots
- One config file to control the entire pipeline
- Output of any task can still be supplied manually
- Flexible, compute only what you need
- Expandable with custom tasks

luigi.cfg

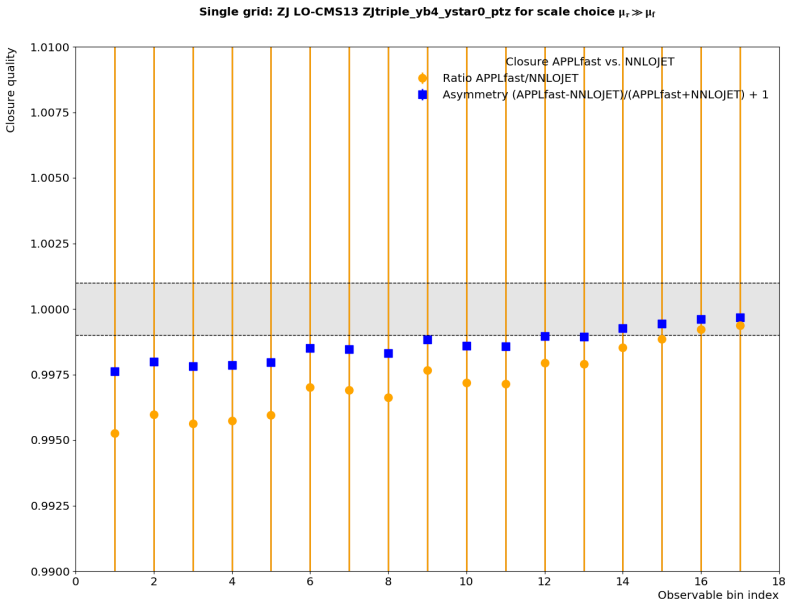
```
[DEFAULT]
name = ZJtriple
process = ZJ
channels = LO R V RRa RRb RV VV
wlcg_path = srm://cmssrm-kit.gridka.de:8443/srm/...
htcondor_accounting_group = cms.jet
htcondor_requirements = (TARGET.ProvidesCPU==true)
htcondor_request_cpus = 1
htcondor_request_memory = 4096
...

[Warmup]
warmup_events = 200000 10000 40000 5000 5000 10000 20000
warmup_iterations = 10 10 10 10 10 10 10
starting_seed = 0
htcondor_request_cpus = 20
htcondor_request_memory = 16384
```

Web UI for generated Plots

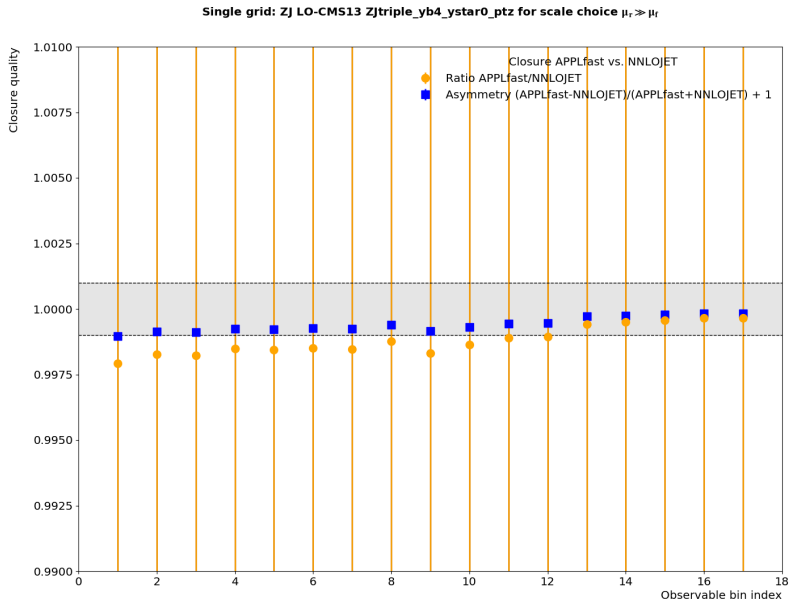


interpolation grid nodes

Adjust number of interpolation nodes to increase or decrease quality and file sizes



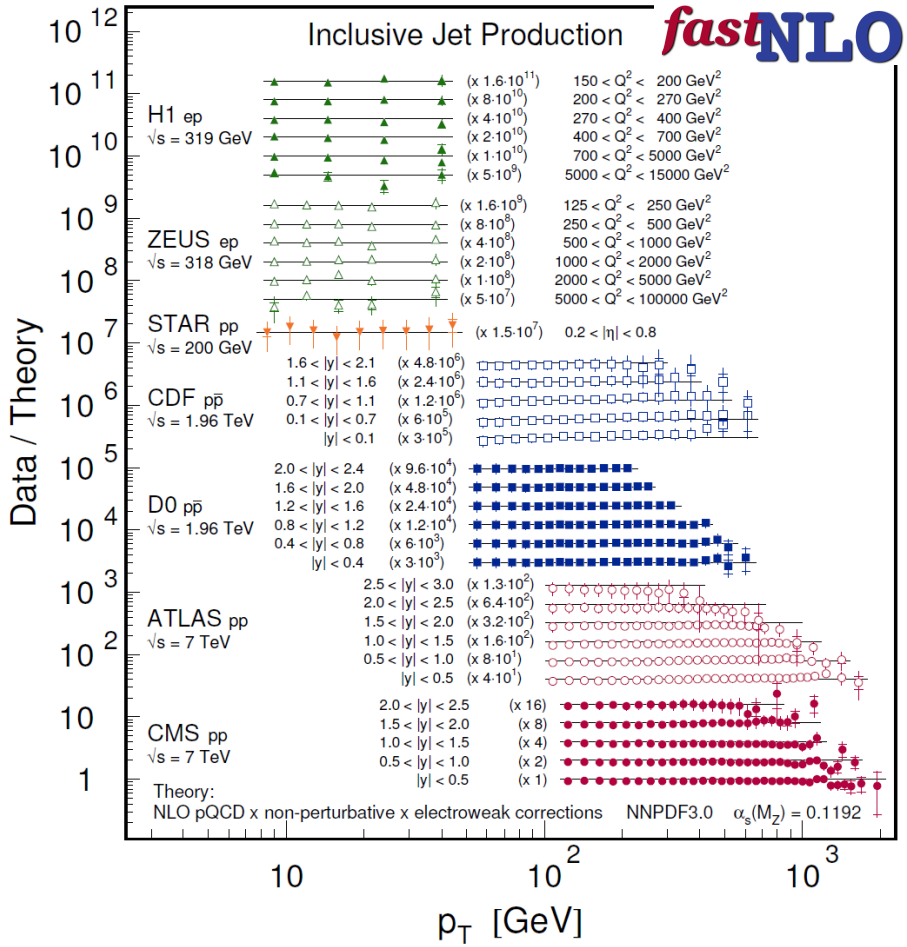
25 x-nodes



30 x-nodes

Conclusion

- Interpolation grids are well suited for PDF fits
- Pipeline for NNLOJET grid production fast and easy to set up
- Planned processes: Jet-Production, Z-Inclusive, W-Inclusive, W+Jet, ...



Grid Production

NNLOJET: Z+Jet-Production

Job-Type	# Jobs	Events/Job	Runtime/Job	# Events	Output	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