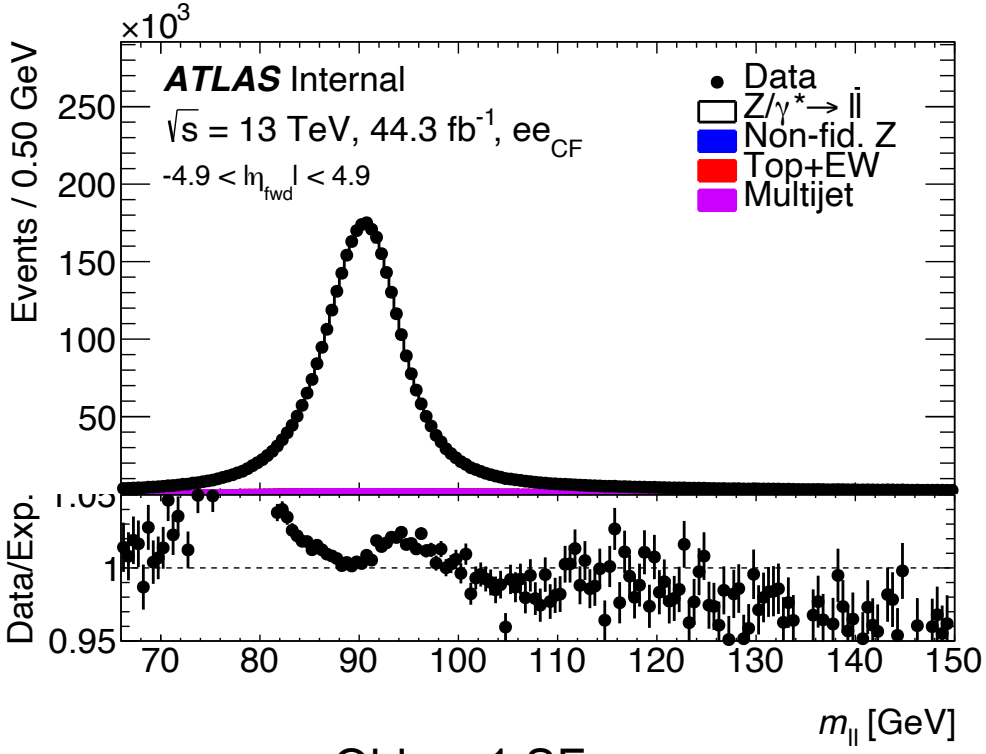
CF – 2017 New Calibration

- All plots made with newest CF version of aidy
- BUT this version has some bug, or I'm doing something wrong, when running the calibration macro.
  - Something causes a seg fault in root when the macro runs
- Use an older version of aidy to run the calibration
- FF version: Forward_2017_Lukas_17240311_EtaBinMode6 (6/0 EtaMode/ETMode)
- MVAFileFwdID: VVL_VL_WPs_2017_2024.txt
- SF version: 17240313 (0/3 EtaMode/ETMode)
- Calibration binning (and similarly for negative eta):
  - 2.5, 2.6, 2.7, 2.8, 2.9, 3.0, 3.1, 3.16, 3.35, 3.6, 4.0, 4.3, 4.6, 4.9

Are these right Lukas?

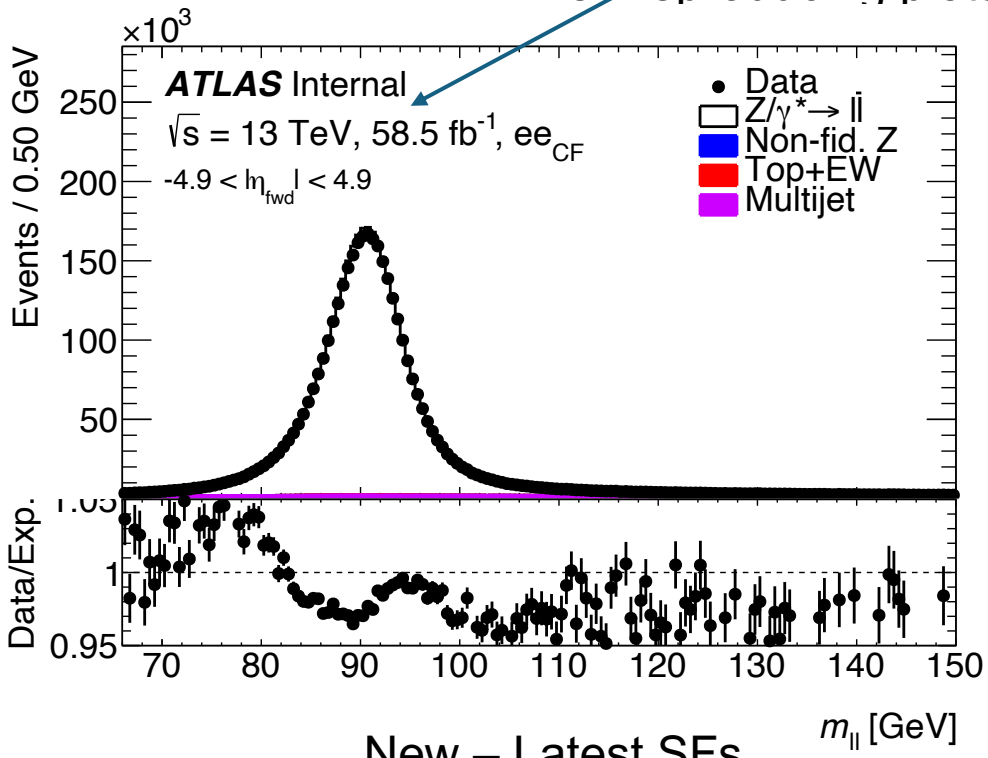
Previously 4.0 – 4.9, changed as we may remove some eta bins

2017 Inclusive Mll – No updated MJ



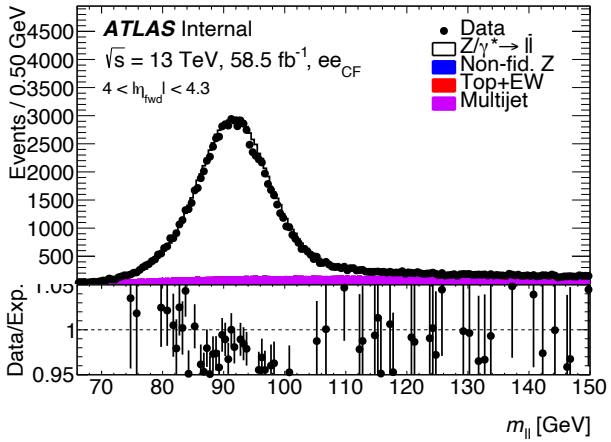
Old – v1 SFs

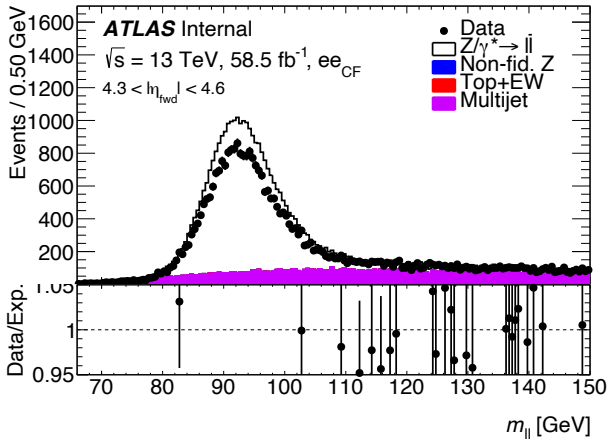
Only just noticed the wrong lumi tag, will fix when reproducing plots.

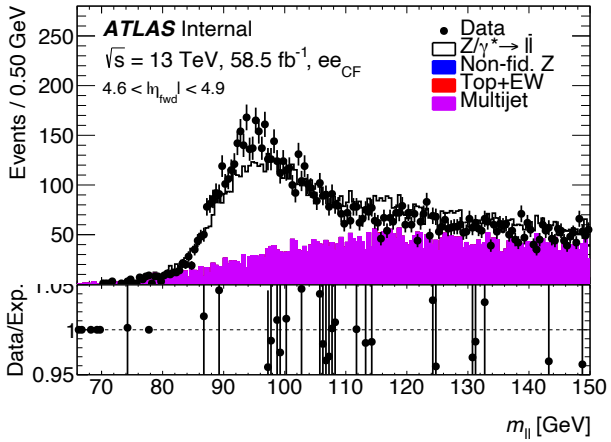


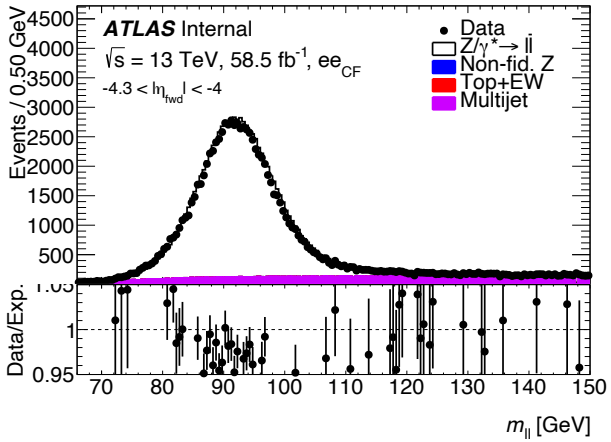
New – Latest SFs

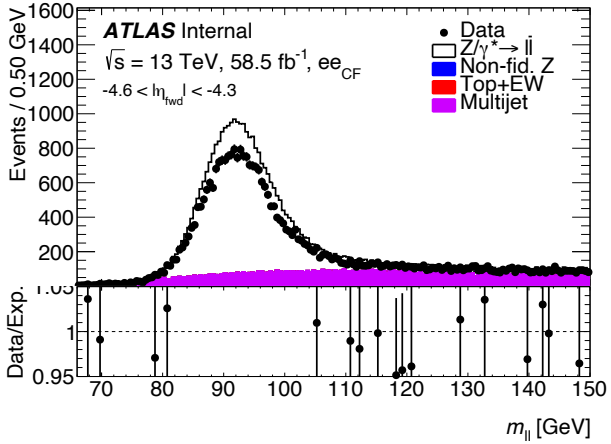
New Calibration – High Eta Bins

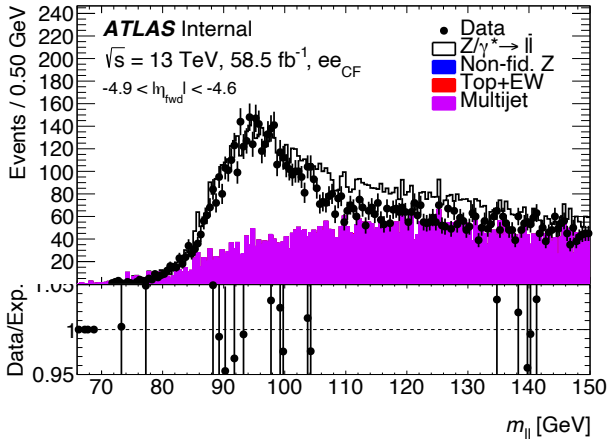








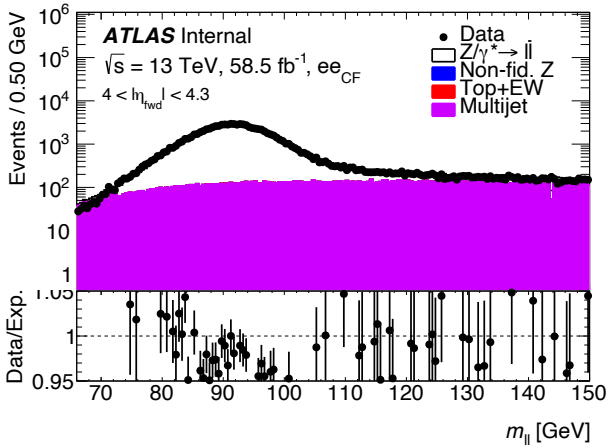


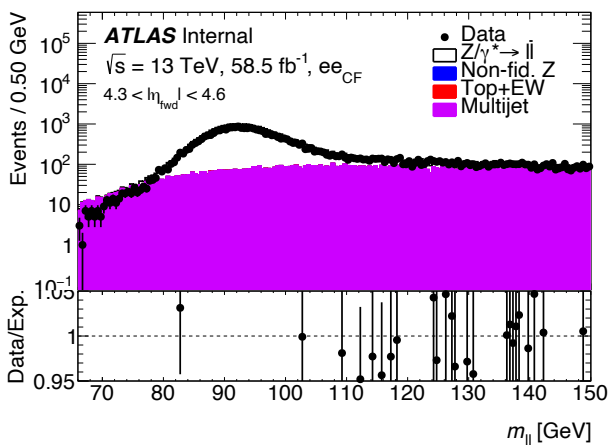


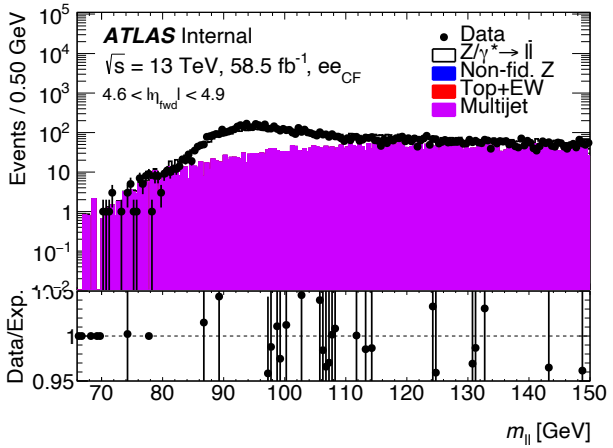
These bins aren't |eta|! Label should be changed

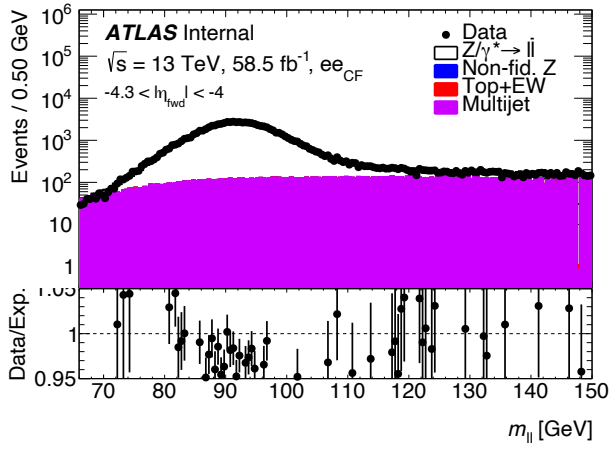
- SFs don't seem to work in 4.3 – 4.6 region. Do you see this Lukas or am I applying them wrong?
- Pure multijet regions are mainly at mll > 102 GeV, do we really want to cut these eta bins?
- More MJ than data still at low eta, only a very small amount though

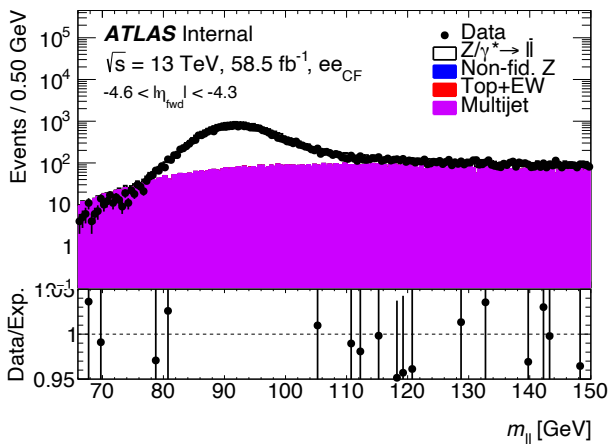
New Calibration – High Eta Bins

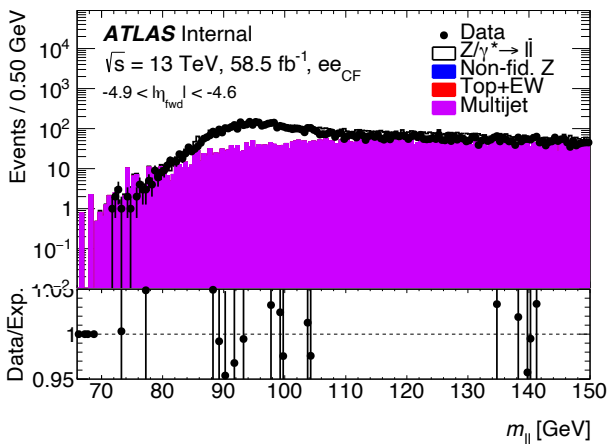








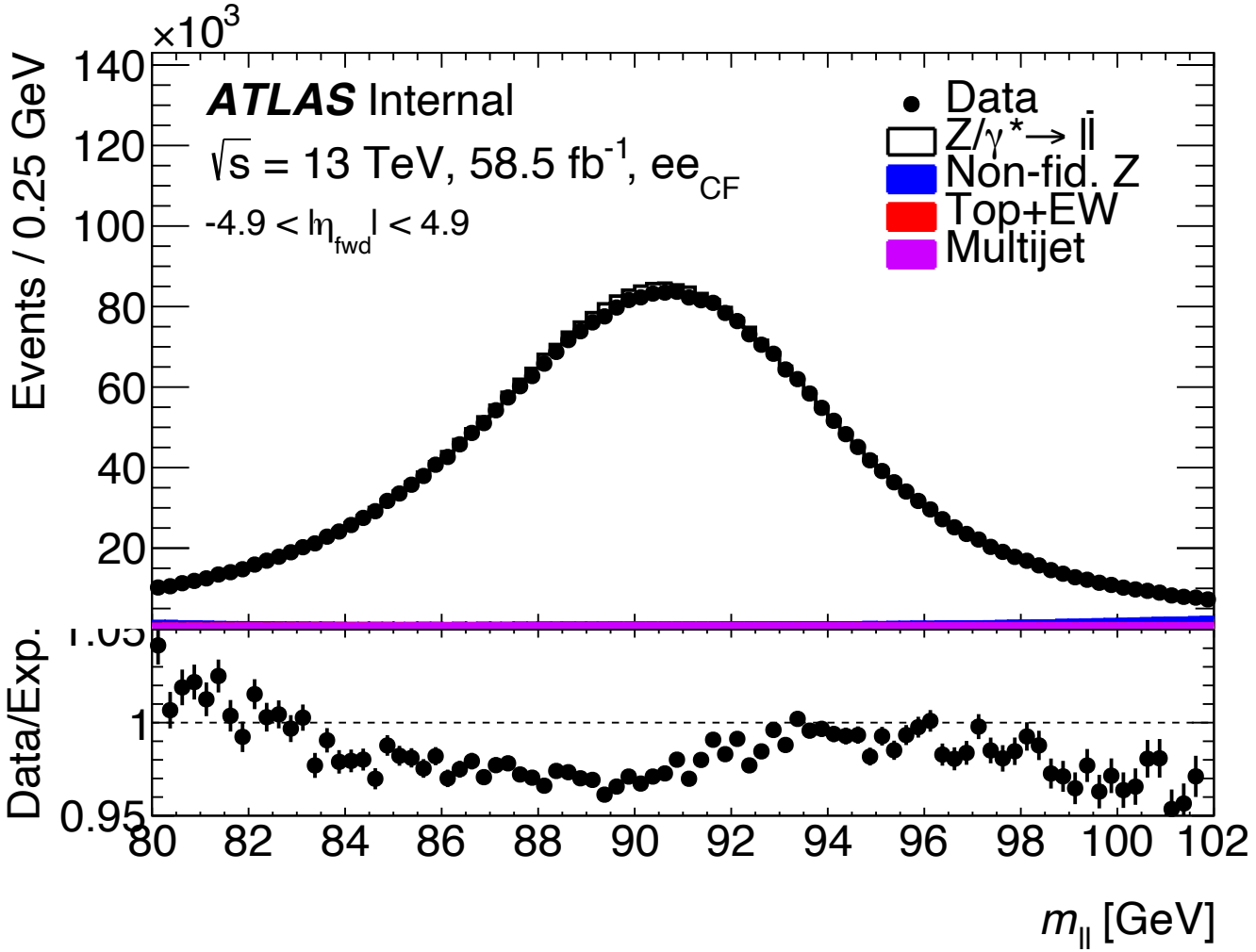




These bins aren't |eta|! Label should be changed

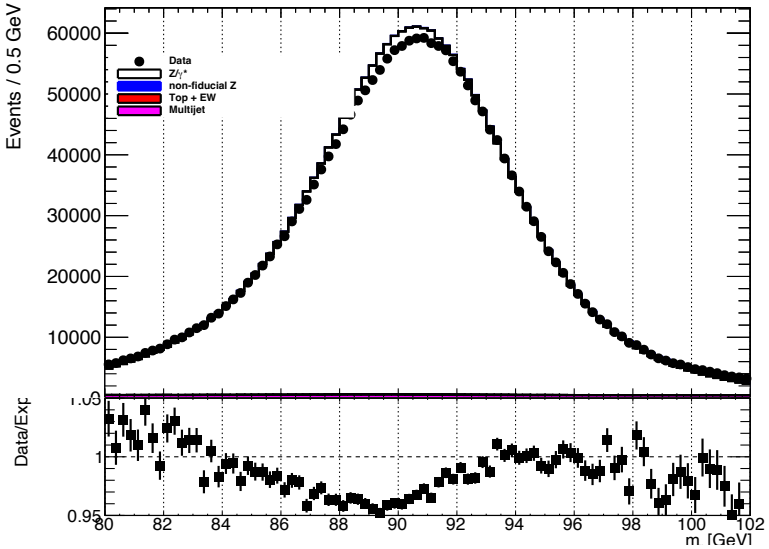
- SFs don't seem to work in 4.3 – 4.6 region. Do you see this Lukas or am I applying them wrong?
- Pure multijet regions are mainly at mll > 102 GeV, do we really want to cut these eta bins?
- More MJ than data still at low eta, only a very small amount though

Update MJ + apply new calibration

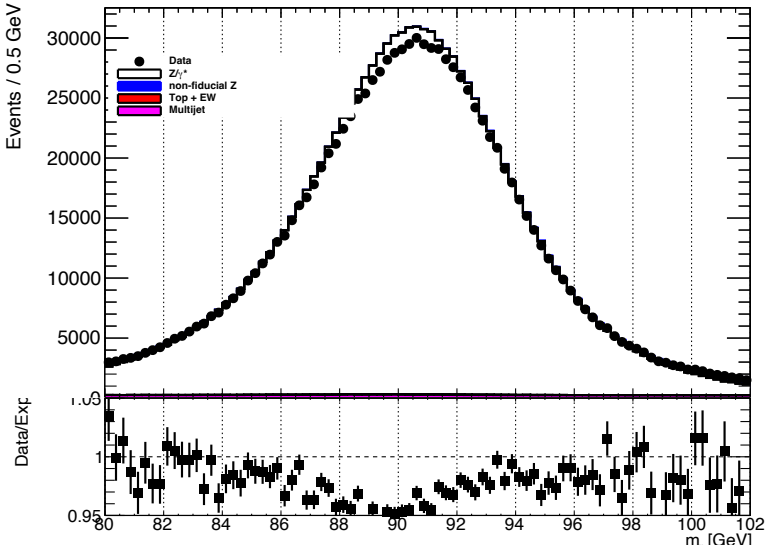


- Things look very nice in the signal region!
- Would need to rerun the scans with calibration mll range to get to see the effect of updating the multijet or if it's even necessary

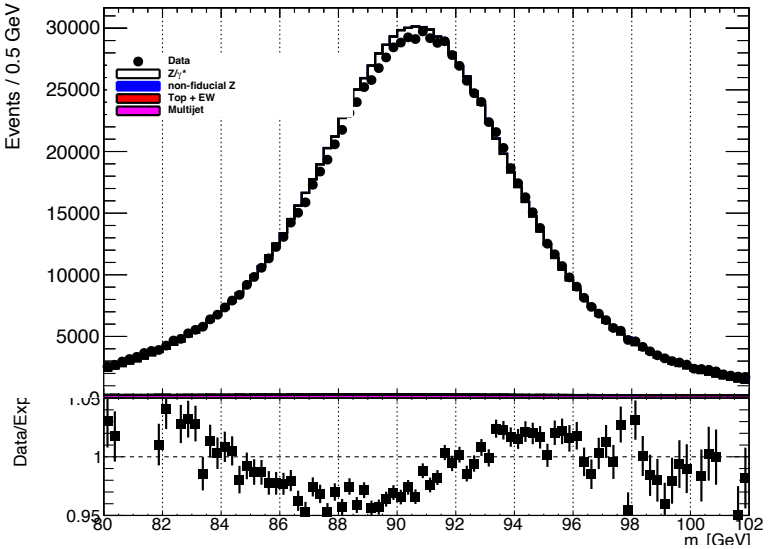
Update MJ + apply new calibration - EMEC



Inclusive



Positive eta

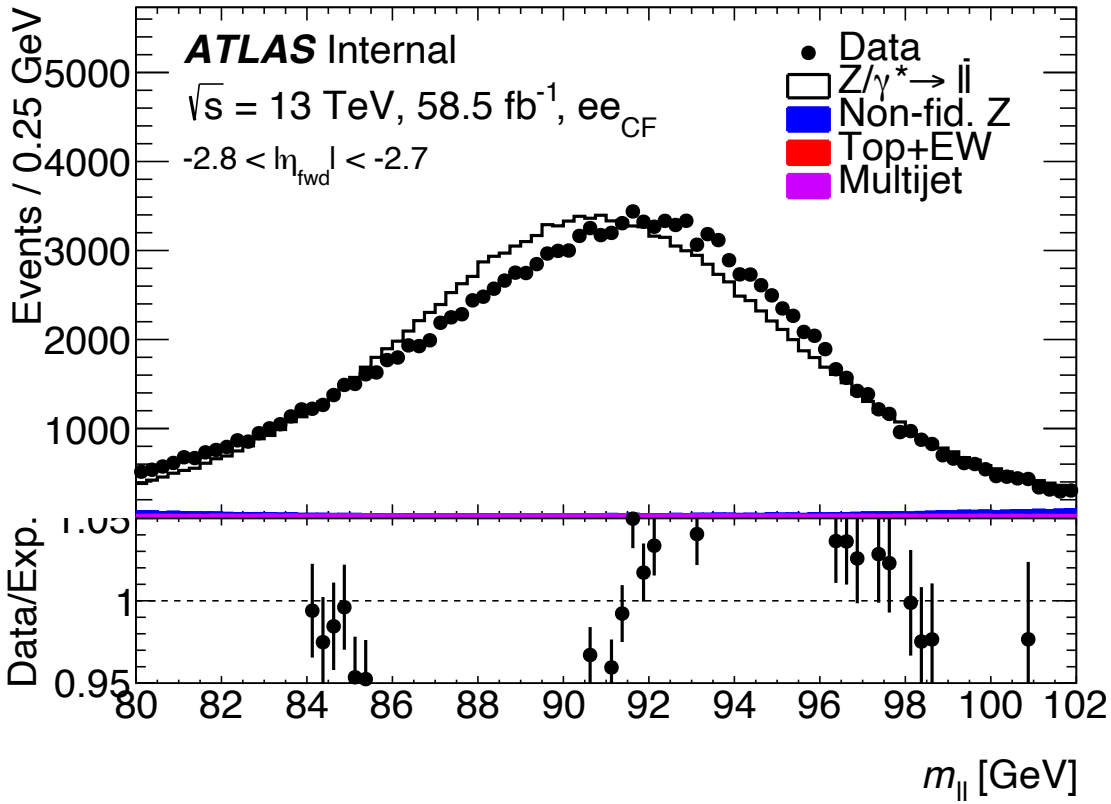


Negative eta

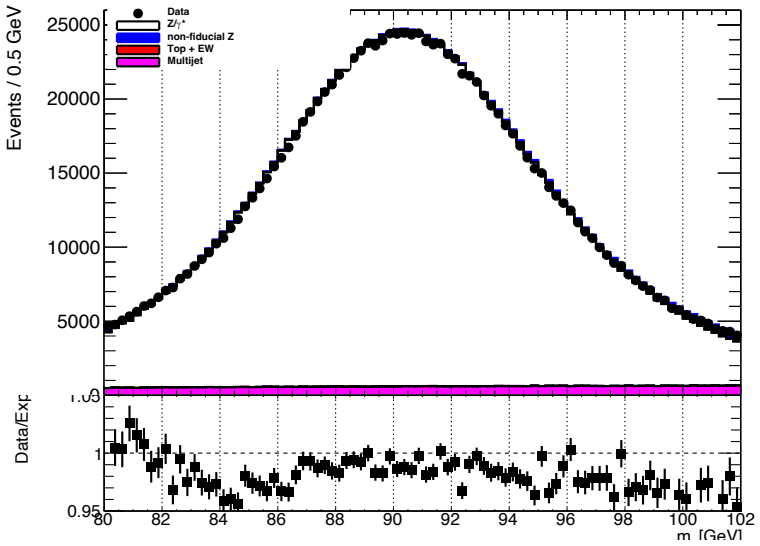
- EMEC looks like it may benefit from the linearity correction, indicated by the “v” structure in the ratio plot
- Good inclusive 93 – 98 GeV agreement is caused by cancellation between positive and negative eta
- Improving SF performance would also improve data/MC agreement
  - 86 – 92 GeV for eta > 0
  - 85 - 91 GeV for eta < 0

Update MJ + apply new calibration - EMEC

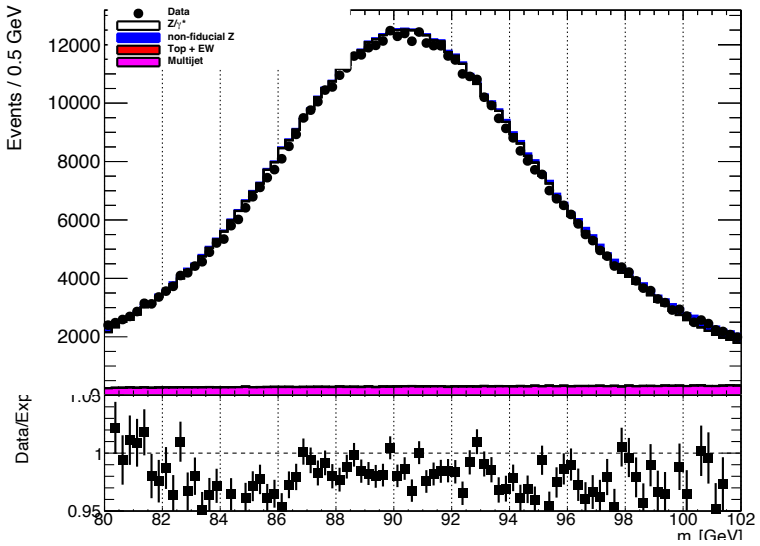
- Data/MC agreement certainly isn't helped by this bin!
  - Historically, this one has always looked a bit dodgy
- Most likely due to calibration failing/not converging to an optimal point and I doubt a linearity correction would help much, the resolution looks quite good already



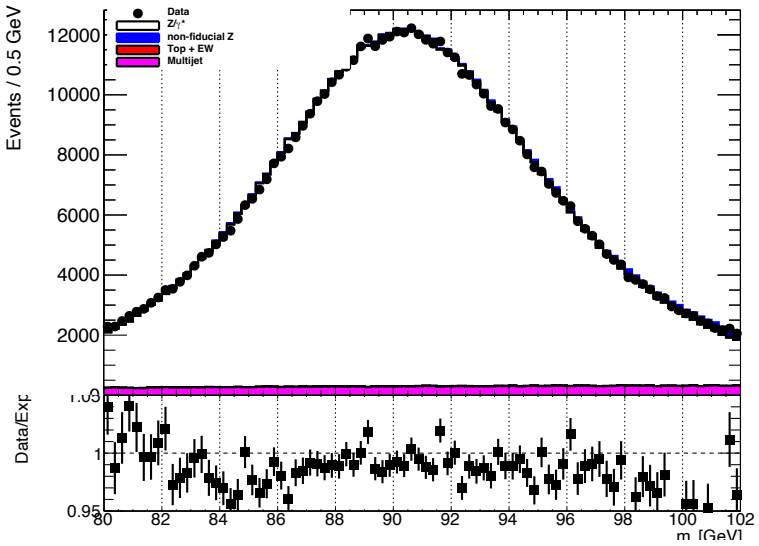
Update MJ + apply new calibration – FCal



Inclusive



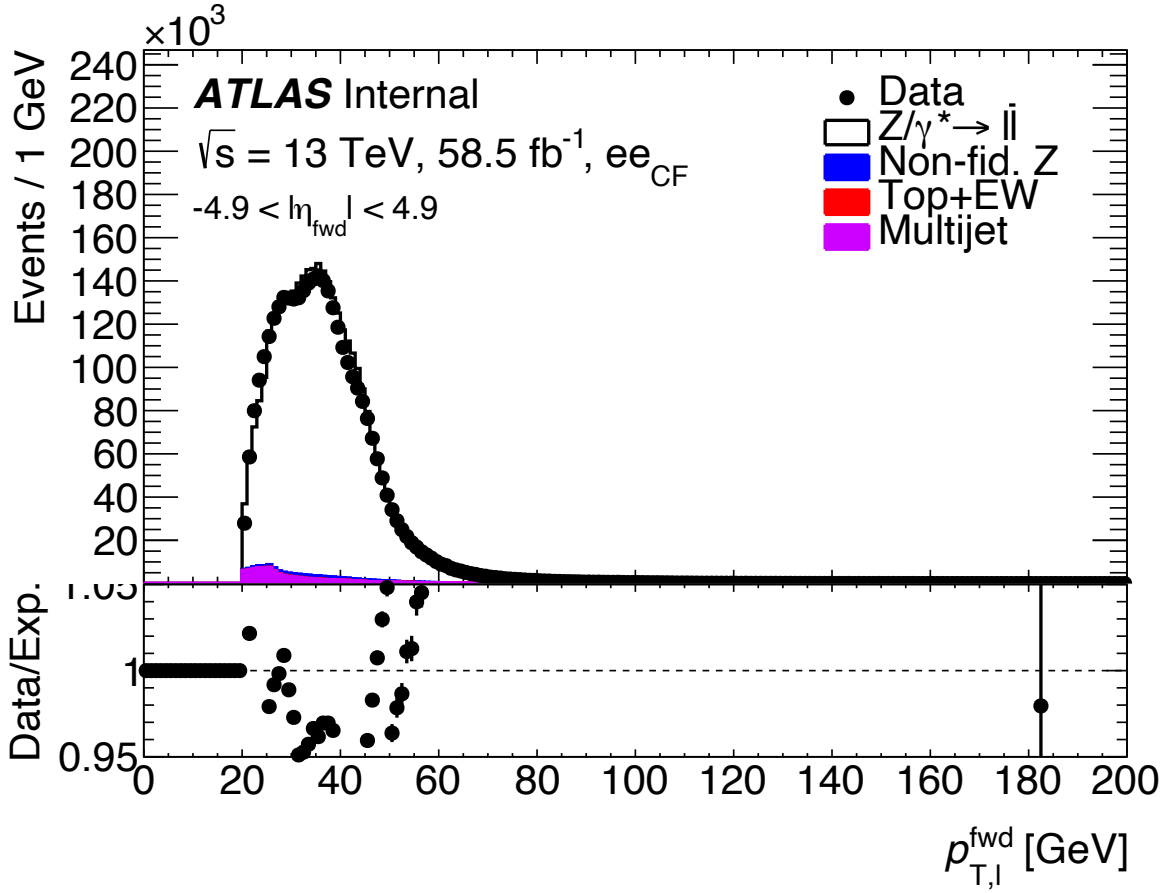
Positive eta



Negative eta

- No further calibration seems necessary in the FCal, a good data/MC agreement is reached.
- If the SFs in 4.3 – 4.6 are fixed, the agreement will probably improve further.

Aside –pT forward distribution

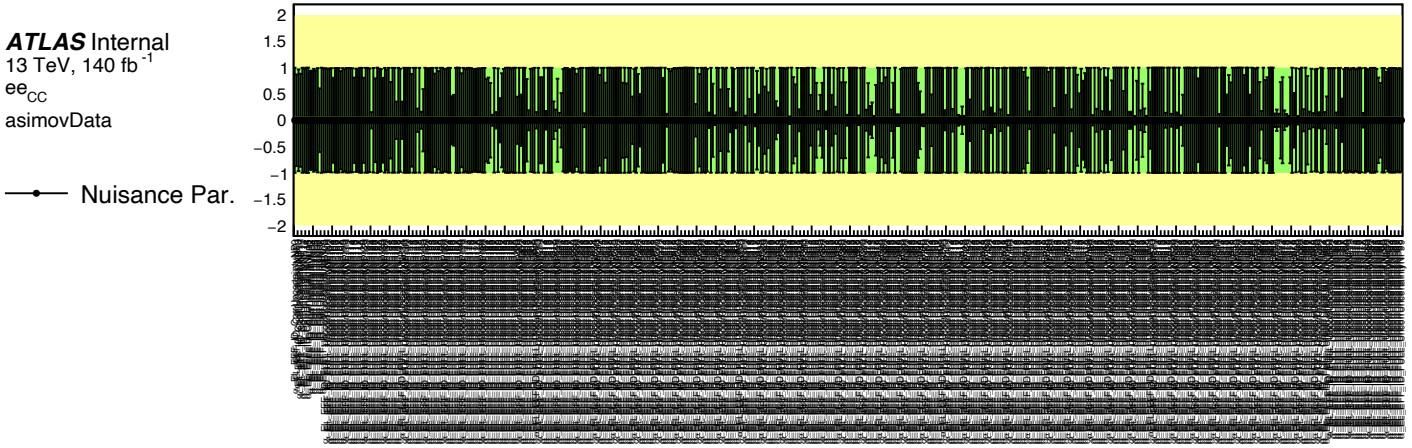


- I don’t see the bump that Lukas has shown but I should be using the exact same code to make the plots/apply SFs/calibration etc.
- Config options causing the difference?

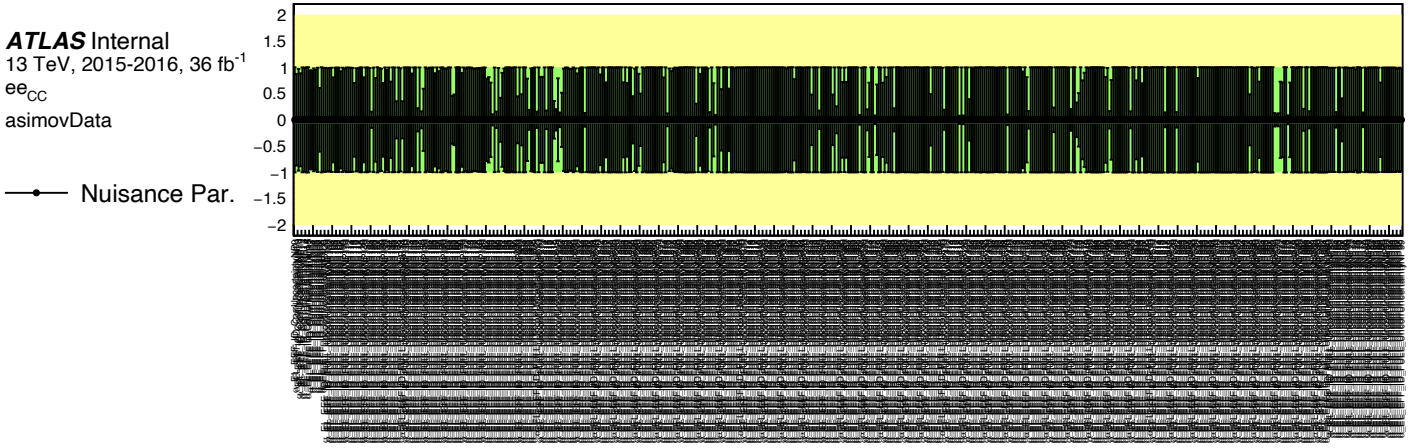
CF to dos/ Open points

- IMO the high eta FCal bins shouldn't be dropped since MJ is still an order of magnitude lower than data in the SR, if one is to be dropped, probably 4.6 – 4.9
  - If we drop the highest eta bin, I'd merge the 4.0 – 4.3 and 4.3 – 4.6 bins to improve performance
- Is it worth looking at a quick (1 day maximum) linearity correction study in the EMEC?
- Fine binning SFs for the central electron?
  - As far as I can see, fine binning is selected only for the CC channel, no idea what it's doing for the CF ntuplemaker
- Perform the calibration for 15/16 + 18:
  - I need the FFs and SFs for these years, Lukas. Can you try and prioritise running this if you haven't already?
- I'm planning on taking 18th April – 1st May off, Asimov s2w studies to be done in May. CF channel calibration has to be finished in early April i.e before e/gamma workshop!

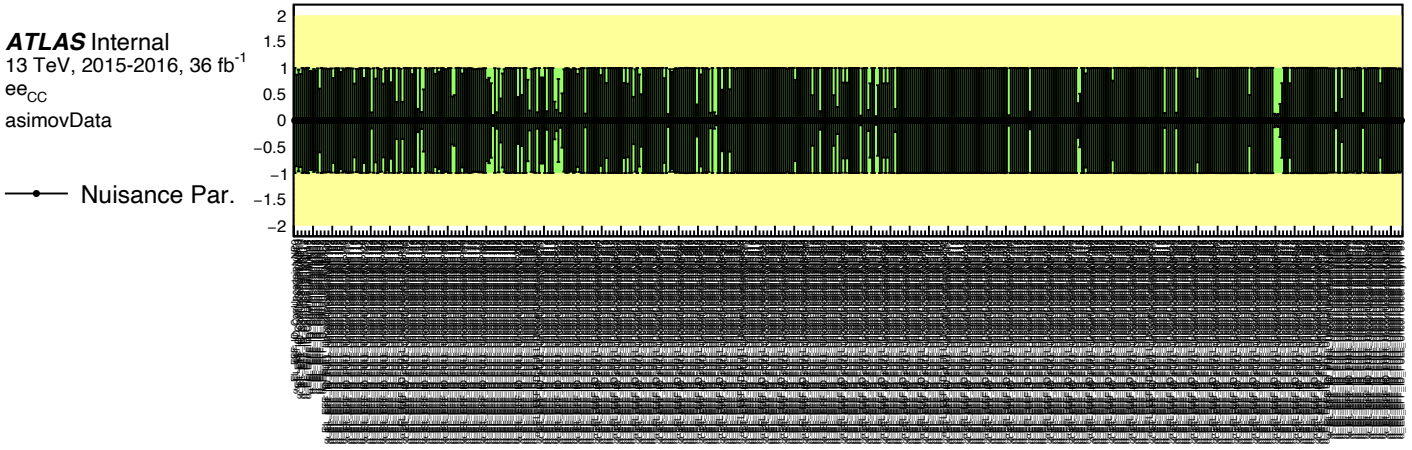
CC 1516 pull plots



Original
212 with error < 0.995

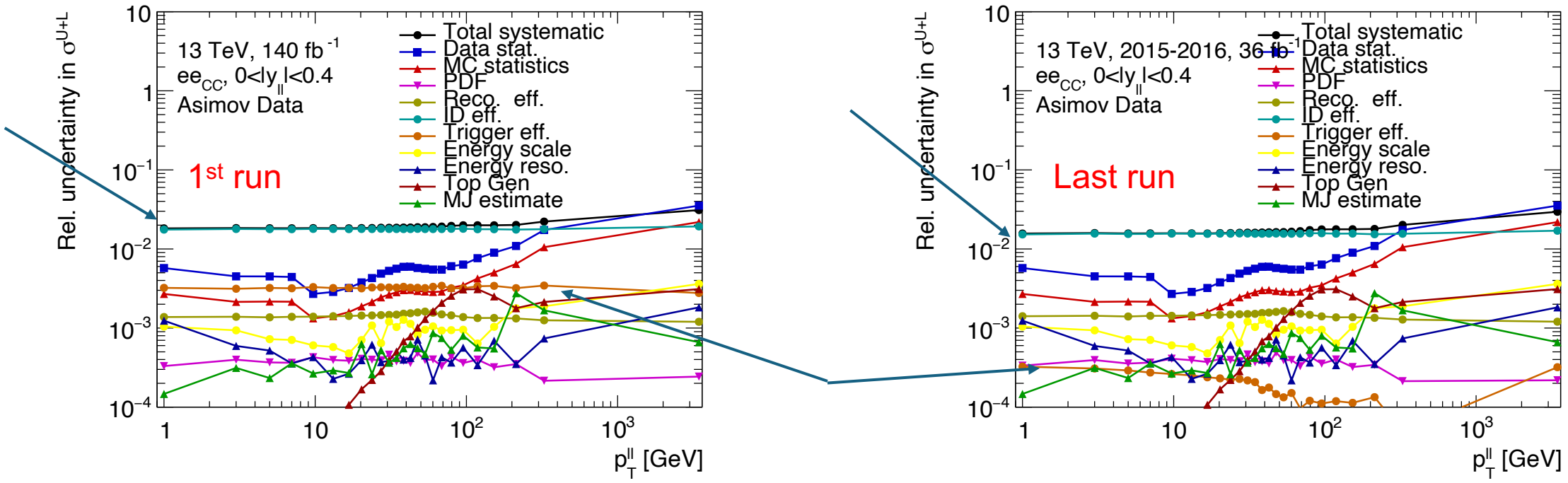


First job resubmission
178 with error < 0.995



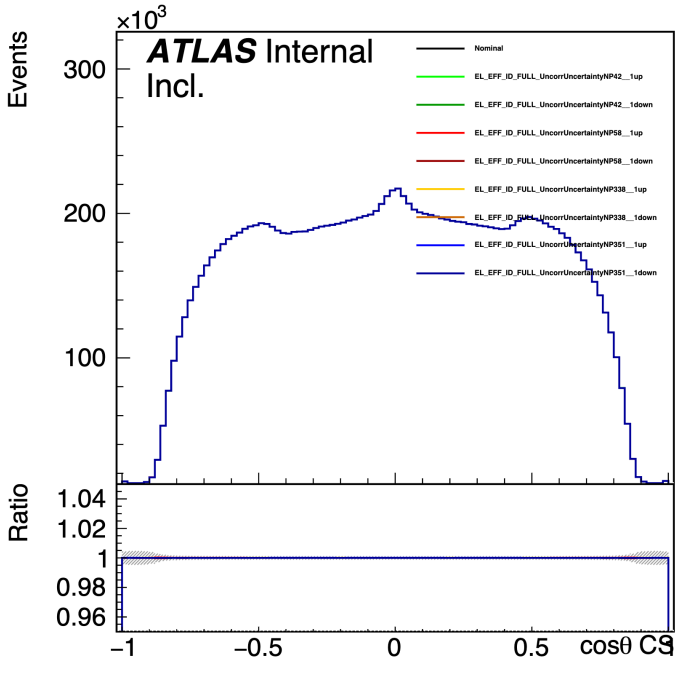
Second job resubmission
174 with error < 0.995

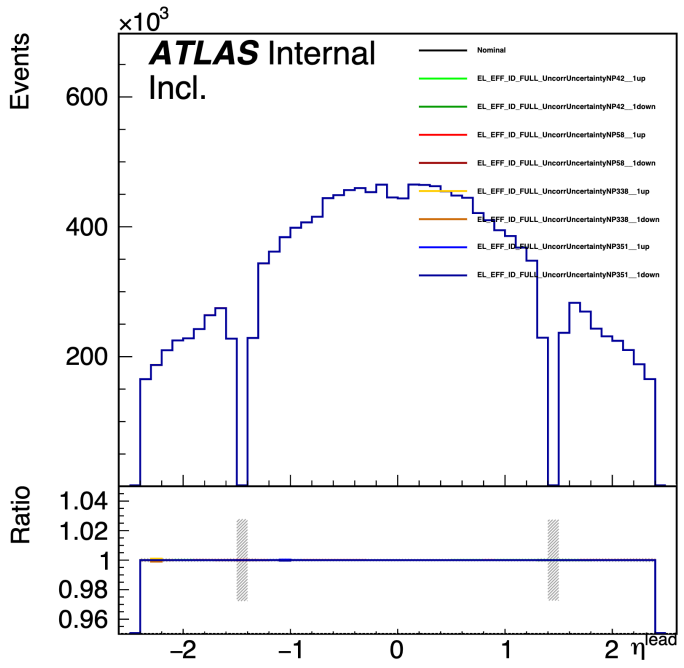
XSec Relative Uncertainty

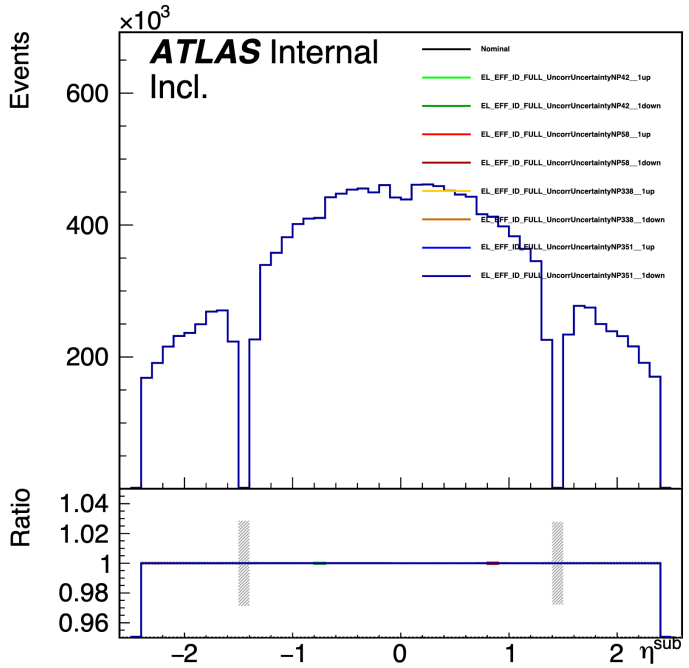


- ID efficiency is decreasing in the relative uncertainty on the cross section (good!) but it still far too high
- Trigger has now been fixed by rerunning the problematic NPs

Comparison to Nominal







- Some problematic NPs have very small deviations from 1 in the ratio wrt nominal in eta lead/sub distributions and cos theta CS.
- Naively I would expect this when varying the ID efficiency? Changes seem in any case too small to constrain NPs so much.

CC Conclusions

- The ID efficiency NPs are improving each time they are run but this doesn't feel like the best strategy going forward!
- Filip is testing his version of the job submission, should be easier/quicker to notice errors in jobs. Maybe they'll look fine for him?
- Unfortunately don't have the individual years on the old ntuples anymore, worth rerunning to get a comparison?
- Worth combining the latest 15/16 with 17 and 18 to see if the relative uncertainty falls to acceptable levels?

Appendix – Calibration Seg fault

```
==========================================================
There was a crash.
This is the entire stack trace of all threads:
==========================================================
#0  0x00002b58c65b660c in waitpid () from /lib64/libc.so.6
#1  0x00002b58c6533f62 in do_system () from /lib64/libc.so.6
#2  0x00002b58c56e9c27 in TUnixSystem::StackTrace() () from /cvmfs/sft.cern.ch/lcg/releases/ROOT/6.18.00-2459b/x86_64-centos7-gcc62-opt/lib/libCore.so
#3  0x00002b58c56ec394 in TUnixSystem::DispatchSignals(ESignals) () from /cvmfs/sft.cern.ch/lcg/releases/ROOT/6.18.00-2459b/x86_64-centos7-gcc62-opt/lib/libCore.so
#4  <signal handler called>
#5  __cxxabiv1::__dynamic_cast (src_ptr=0x2b58c68b87b8 <main_arena+88>, src_type=0x2b58d776b950 <typeinfo for RooAbsArg>, dst_type=0x2b58d776f510 <typeinfo for RooAbsLValue>, src2dst=-2) a
#6  0x00002b58d732b536 in RooDataHist::sum(RooArgSet const&, RooArgSet const&, bool, bool, std::map<RooAbsArg const*, std::pair<double, double>, std::less<RooAbsArg const*>, std::allocator
#7  0x00002b58d7388340 in RooHistPdf::analyticalIntegral(int, char const*) const () from /cvmfs/sft.cern.ch/lcg/releases/ROOT/6.18.00-2459b/x86_64-centos7-gcc62-opt/lib/libRooFitCore.so
#8  0x00002b58d7299d46 in RooAbsPdf::analyticalIntegralWN(int, RooArgSet const*, char const*) const () from /cvmfs/sft.cern.ch/lcg/releases/ROOT/6.18.00-2459b/x86_64-centos7-gcc62-opt/lib/
#9  0x00002b58d740dd5b in RooRealIntegral::evaluate() const () from /cvmfs/sft.cern.ch/lcg/releases/ROOT/6.18.00-2459b/x86_64-centos7-gcc62-opt/lib/libRooFitCore.so
#10 0x00002b58d72abb82 in RooAbsReal::traceEval(RooArgSet const*) const () from /cvmfs/sft.cern.ch/lcg/releases/ROOT/6.18.00-2459b/x86_64-centos7-gcc62-opt/lib/libRooFitCore.so
#11 0x00002b58d740887c in RooRealIntegral::getValV(RooArgSet const*) const () from /cvmfs/sft.cern.ch/lcg/releases/ROOT/6.18.00-2459b/x86_64-centos7-gcc62-opt/lib/libRooFitCore.so
#12 0x00002b58d729a517 in RooAbsPdf::syncNormalization(RooArgSet const*, bool) const () from /cvmfs/sft.cern.ch/lcg/releases/ROOT/6.18.00-2459b/x86_64-centos7-gcc62-opt/lib/libRooFitCore.s
#13 0x00002b58d7298890 in RooAbsPdf::getValV(RooArgSet const*) const () from /cvmfs/sft.cern.ch/lcg/releases/ROOT/6.18.00-2459b/x86_64-centos7-gcc62-opt/lib/libRooFitCore.so
#14 0x00002b58dcb58cad in MyRooFFTConvPdf::scanPdf(RooRealVar&, RooAbsPdf&, RooDataHist const&, RooArgSet const&, int&, int&, int&, double) const () from /nfs/dust/atlas/user/wellscj/ZAi/e
#15 0x00002b58dcb59327 in MyRooFFTConvPdf::fillCacheSlice(MyRooFFTConvPdf::FFTCacheElem&, RooArgSet const&) const () from /nfs/dust/atlas/user/wellscj/ZAi/eeCF/aidy/macros/fitCalib_C.so
#16 0x00002b58dcb672c9 in MyRooFFTConvPdf::fillCacheObject(MyRooAbsCachedPdf::MyPdfCacheElem&) const () from /nfs/dust/atlas/user/wellscj/ZAi/eeCF/aidy/macros/fitCalib_C.so
#17 0x00002b58dcb70143 in MyRooAbsCachedPdf::getCache(RooArgSet const*, bool) const () from /nfs/dust/atlas/user/wellscj/ZAi/eeCF/aidy/macros/fitCalib_C.so
#18 0x00002b58dcb70ce1 in MyRooAbsCachedPdf::getValV(RooArgSet const*) const [clone .localalias.520] () from /nfs/dust/atlas/user/wellscj/ZAi/eeCF/aidy/macros/fitCalib_C.so
#19 0x00002b58d72e7bfd in RooAddPdf::evaluate() const () from /cvmfs/sft.cern.ch/lcg/releases/ROOT/6.18.00-2459b/x86_64-centos7-gcc62-opt/lib/libRooFitCore.so
#20 0x00002b58d72989fb in RooAbsPdf::getValV(RooArgSet const*) const () from /cvmfs/sft.cern.ch/lcg/releases/ROOT/6.18.00-2459b/x86_64-centos7-gcc62-opt/lib/libRooFitCore.so
#21 0x00002b58d7436566 in RooSimultaneous::evaluate() const () from /cvmfs/sft.cern.ch/lcg/releases/ROOT/6.18.00-2459b/x86_64-centos7-gcc62-opt/lib/libRooFitCore.so
#22 0x00002b58d72989fb in RooAbsPdf::getValV(RooArgSet const*) const () from /cvmfs/sft.cern.ch/lcg/releases/ROOT/6.18.00-2459b/x86_64-centos7-gcc62-opt/lib/libRooFitCore.so
#23 0x00002b58dcb73c95 in fitCalib(std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> >, int, int, int, int, int, bool, bool, std::__cxx11::basic_string<char, st
#24 0x00002b58c4f89177 in ?? ()
#25 0x00007ffd00000000 in ?? ()
#26 0x00007ffd00000001 in ?? ()
#27 0x00007ffdea859bb0 in ?? ()
#28 0x0000000000000004 in ?? ()
#29 0x0000000000000001 in ?? ()
#30 0x00007ffd00000000 in ?? ()
#31 0x00007ffdea859b88 in ?? ()
#32 0x00000000ffffffff in ?? ()
#33 0x0000000000000015 in ?? ()
#34 0x00007ffdea859b70 in ?? ()
#35 0x0000000000000000 in ?? ()
==========================================================


The lines below might hint at the cause of the crash.
You may get help by asking at the ROOT forum http://root.cern.ch/forum
Only if you are really convinced it is a bug in ROOT then please submit a
```