FWD electron calibration miscellanea

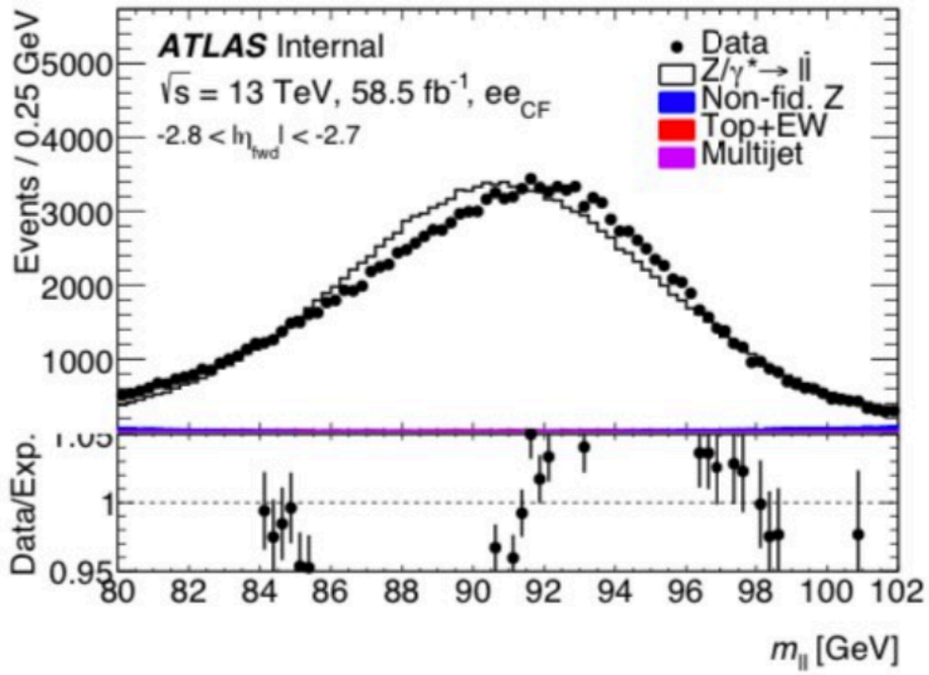
ETA REGION 2.7-2.8

BAd bin

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

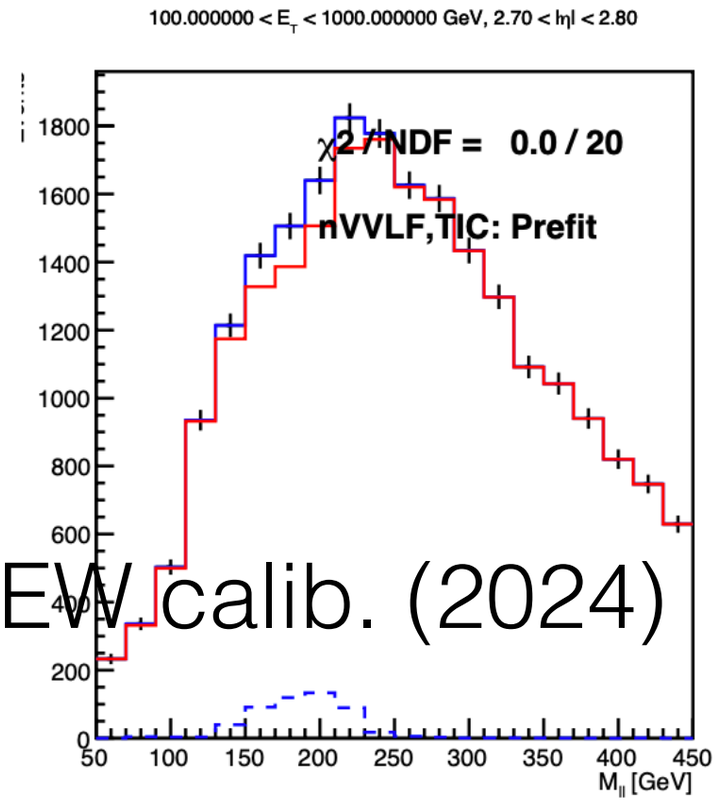
- Most likely due to calibration failing/not converging to an optimal point and I doubt a linearity correction would help much, the resolution looks quite good already

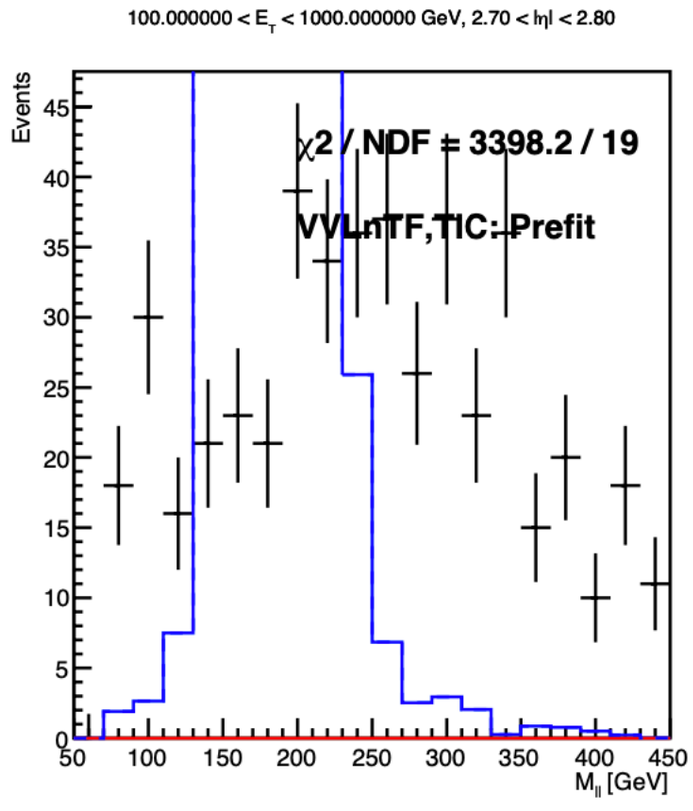


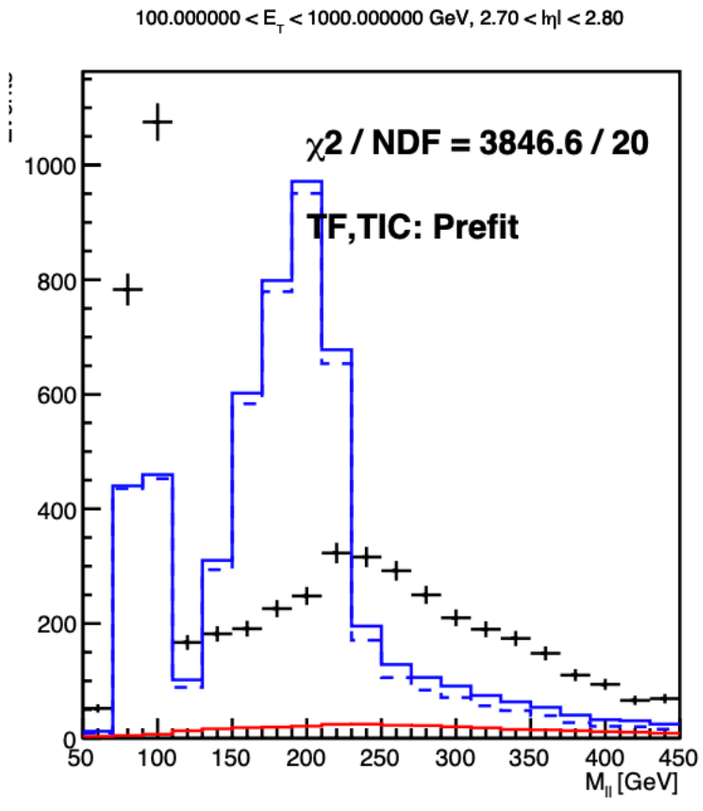
bad bin eta 2.7-2.8

but we never managed to be sure it is material (even if 2.7-2.8 could be affected by Pixel services). At some stage we wanted to run some MC with extra material in this region to check the sensitivity, but I am not sure we never did...

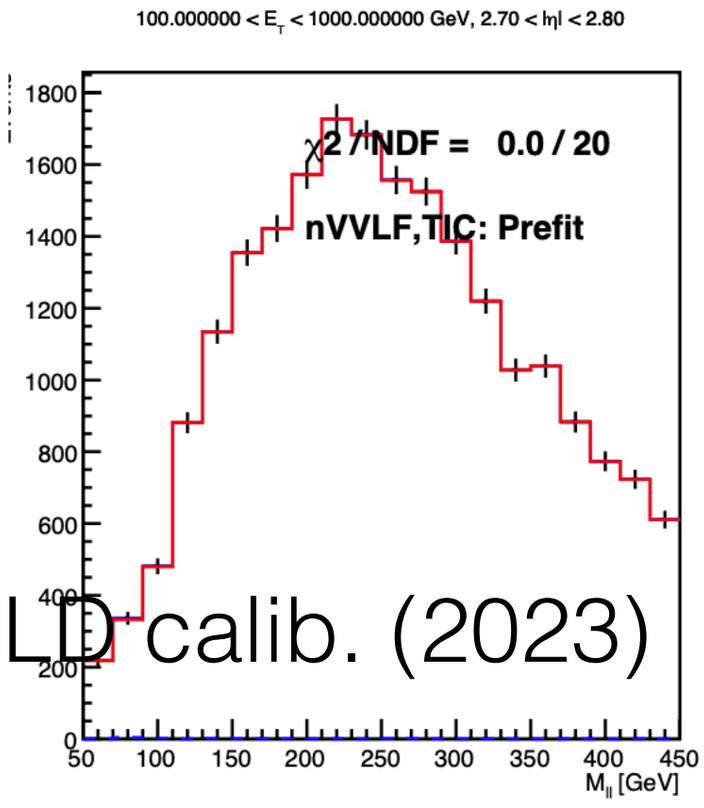
yes, the bkg is probably OK there, the issue is really non-gaussian tails in the mass resolution.

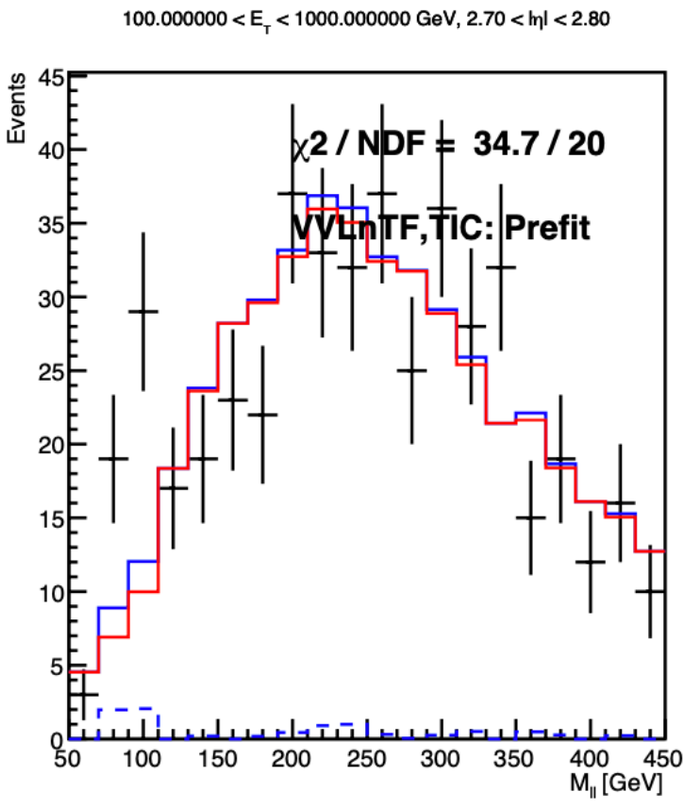


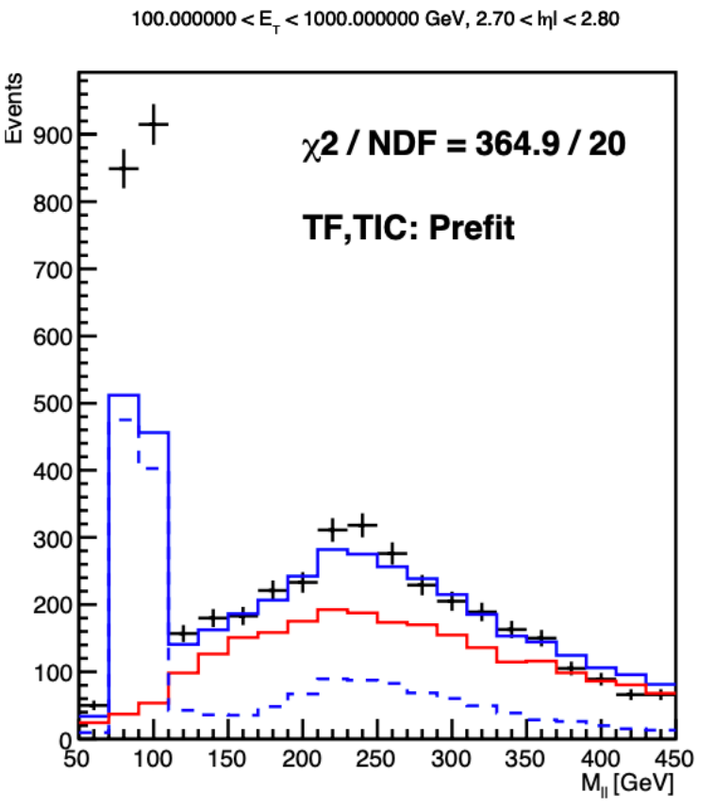




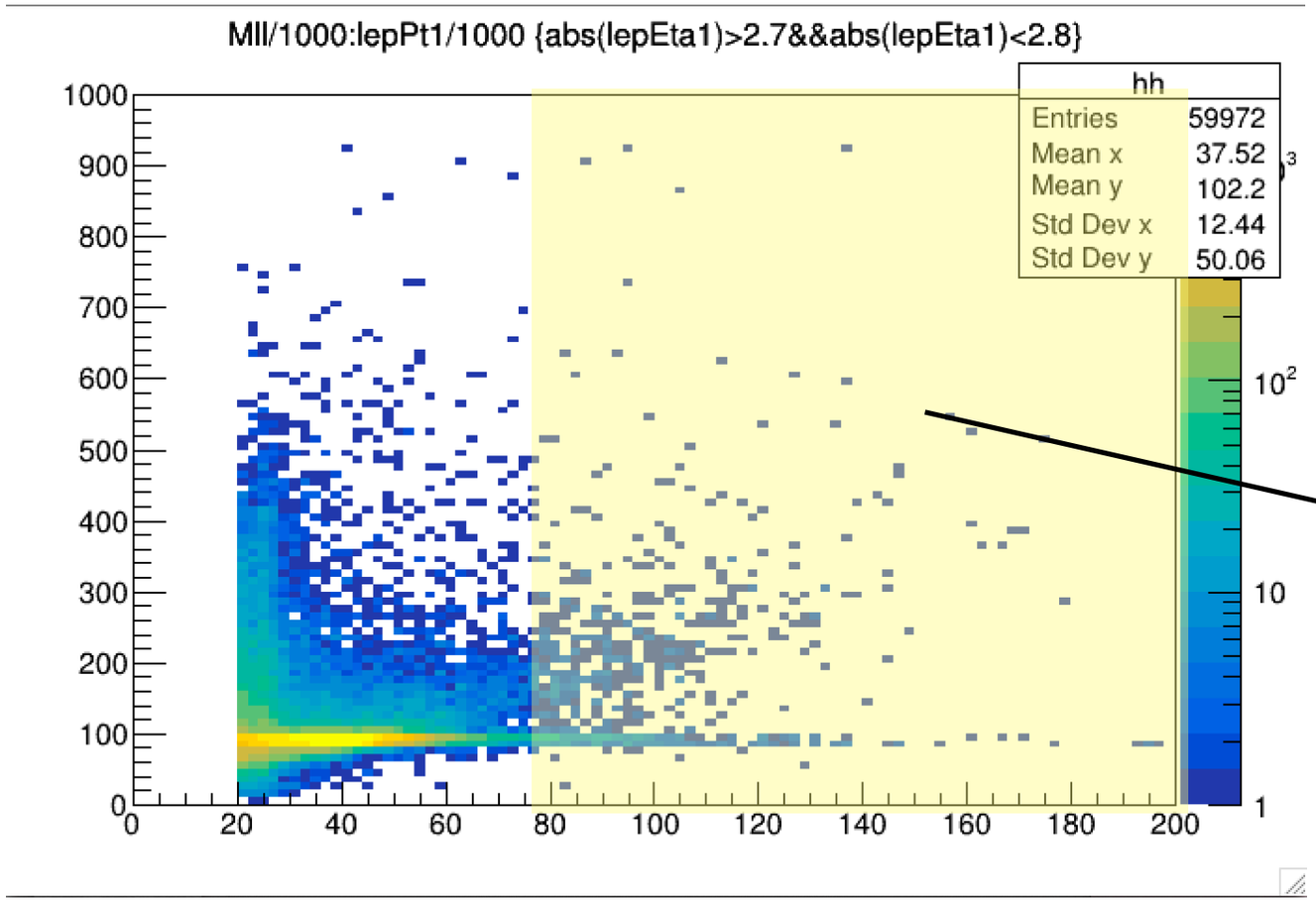
NEW calib. (2024)

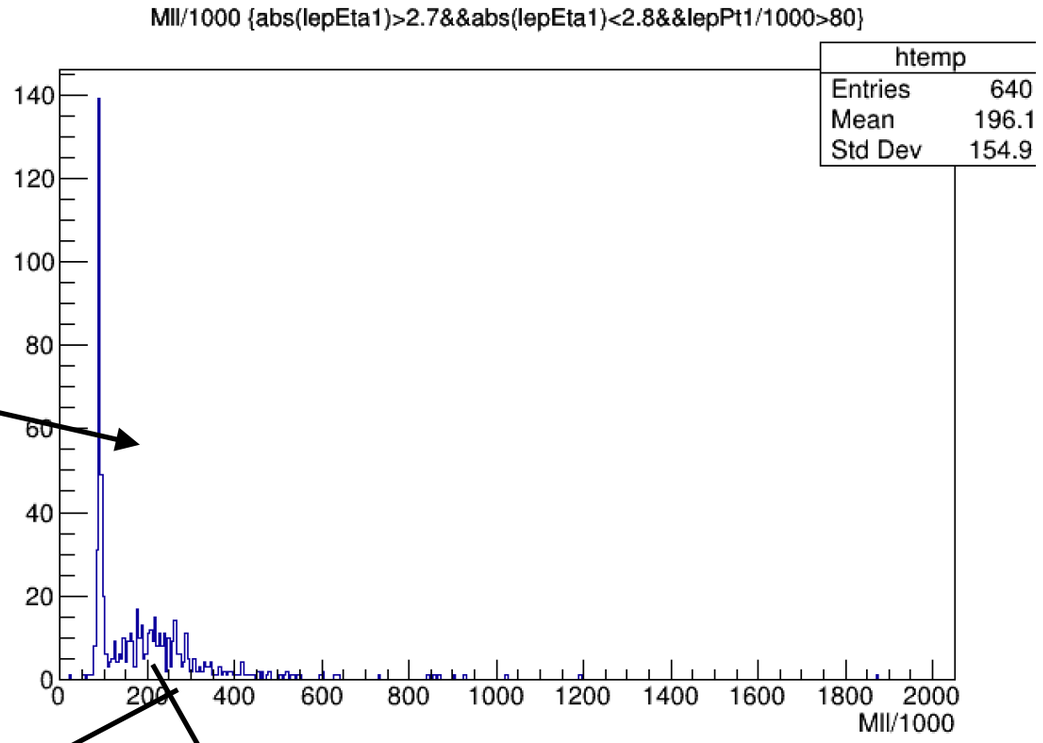


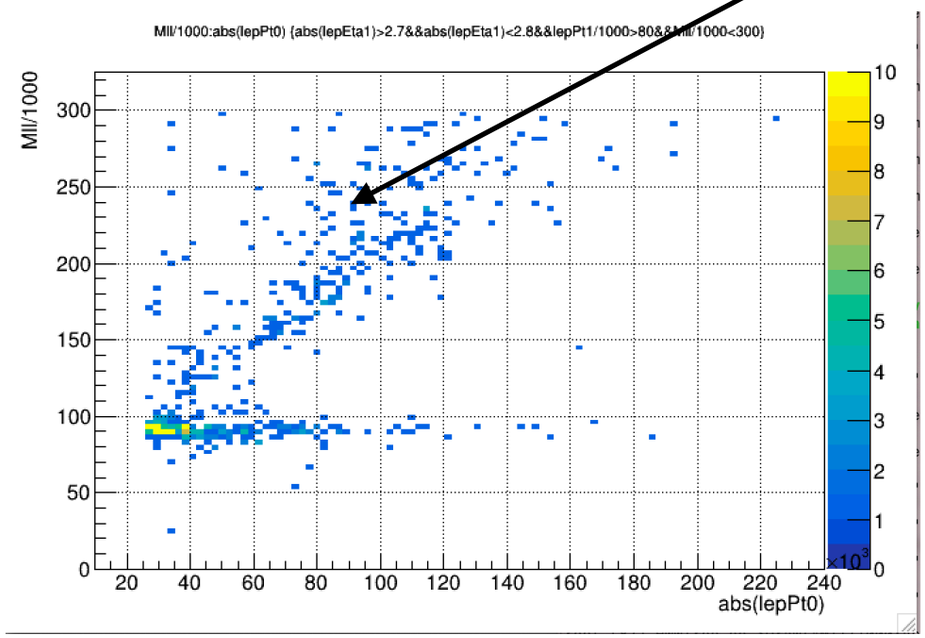


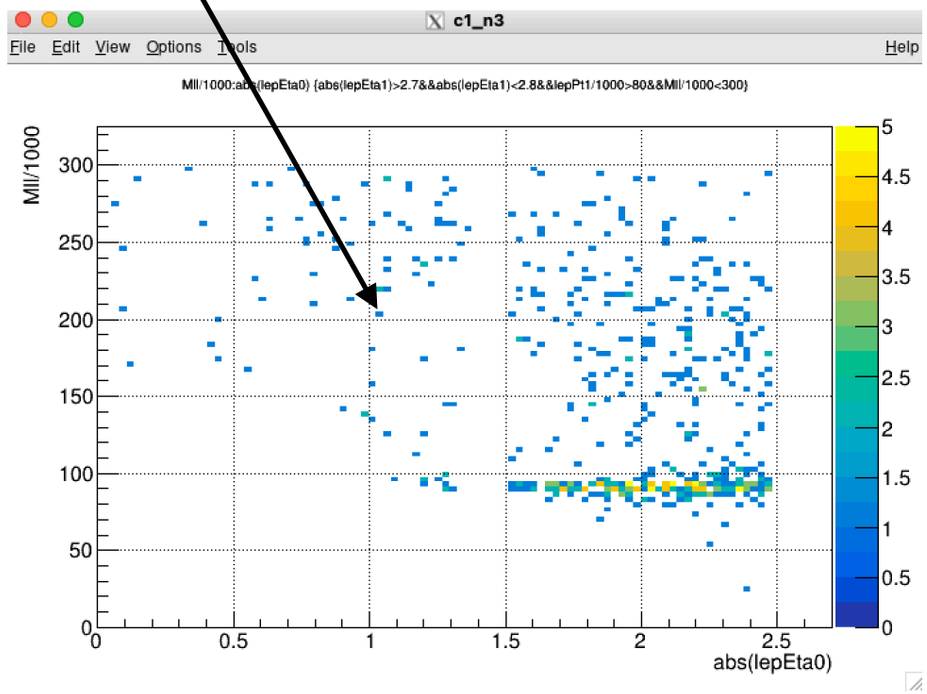


OLD calib. (2023)


Mll/1000:lepPt1/1000 {abs(lepEta1)>2.7&&abs(lepEta1)<2.8}
hh
Entries 59972
Mean x 37.52
Mean y 102.2
Std Dev x 12.44
Std Dev y 50.06


Mll/1000 {abs(lepEta1)>2.7&&abs(lepEta1)<2.8&&lepPt1/1000>80}
htemp
Entries 640
Mean 196.1
Std Dev 154.9
Mll/1000


Mll/1000:abs(lepPt0) {abs(lepEta1)>2.7&&abs(lepEta1)<2.8&&lepPt1/1000>80&&Mll/1000<300}
Mll/1000
abs(lepPt0)


c1_n3
File Edit View Options Tools Help
Mll/1000:abs(lepEta0) {abs(lepEta1)>2.7&&abs(lepEta1)<2.8&&lepPt1/1000>80&&Mll/1000<300}
Mll/1000
abs(lepEta0)

futher checks

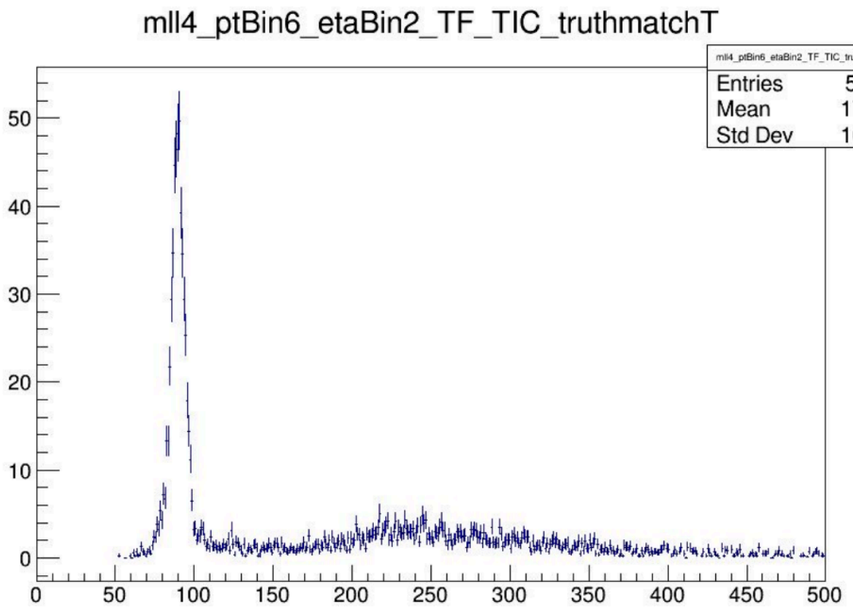
16/07/24

futher checks

- https://indico.desy.de/event/45074/contributions/169531/attachments/91526/123646/bad_bin_eta2_pt6.pdf

MC reco scan in bad bin (eta: 2.7-2.8, pt: >100GeV)

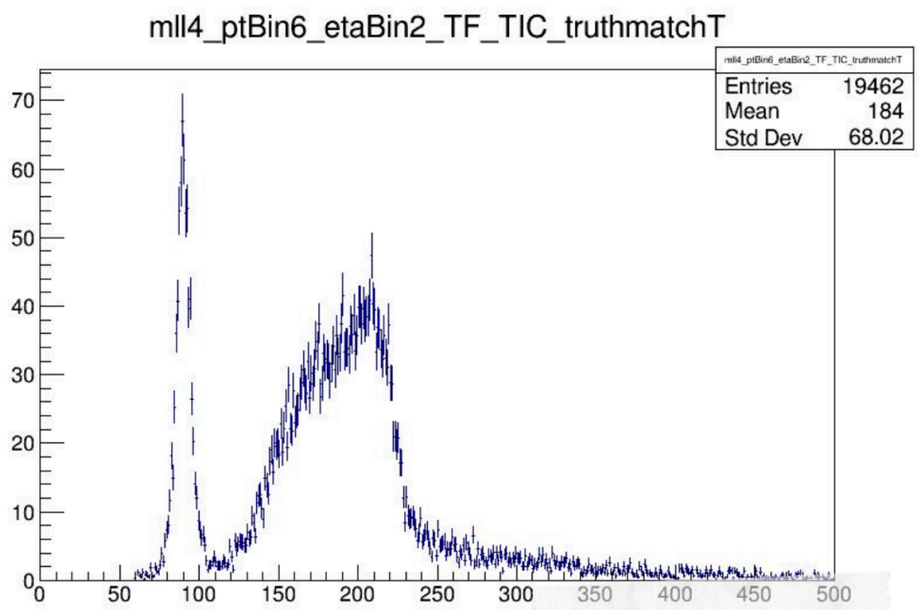
new production + old calibration + Alma9 aidy



2023 calib: CF_DSCB_preCalibAlignSF_17_1210_2023_bm2_p1_x0_t1

(bin mode 8)

new production + new calibration + Alma9 aidy

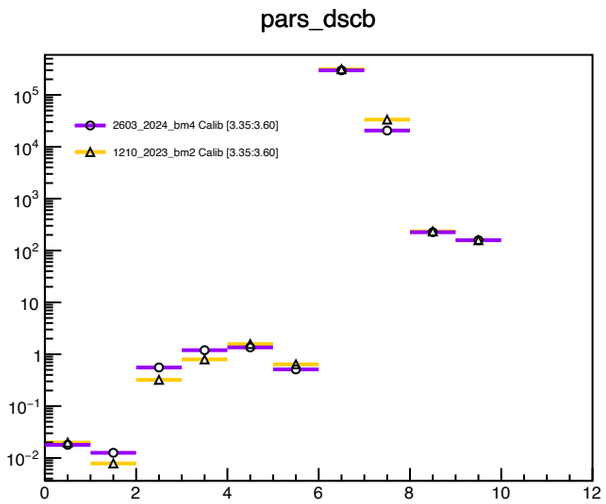


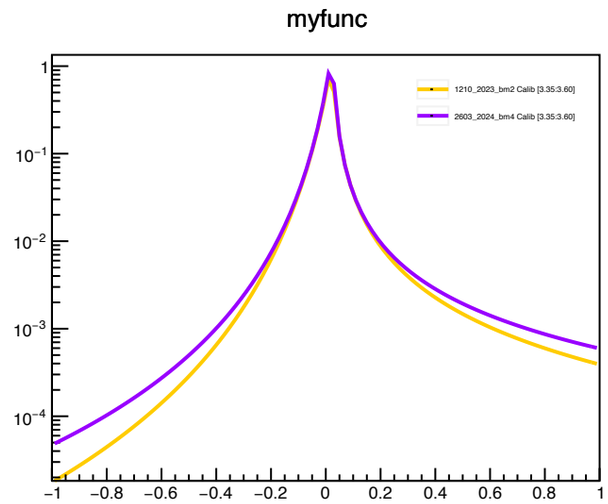
2024 calib:
CF_DSCB_preCalibAlignSF_FFEta6Pt0_M10_LukasConfig_17_2603_2024_bm4_p1_x0_t1

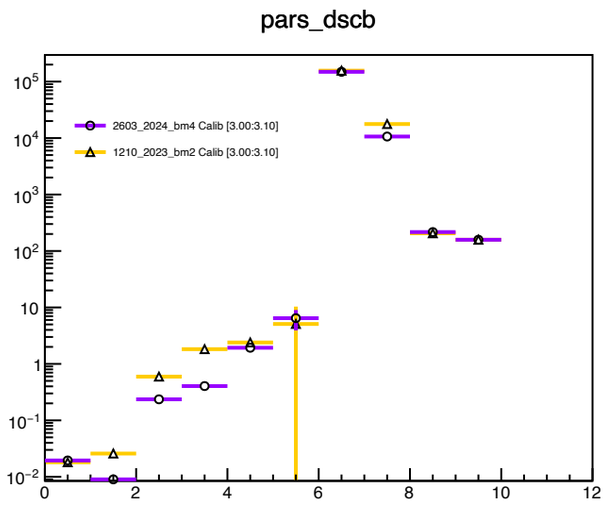
(bin model 9)

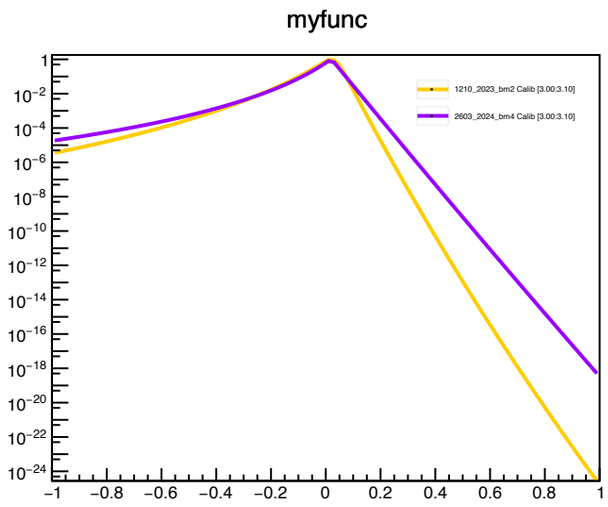
decide to check those 2 calibration comparing input files to the scans

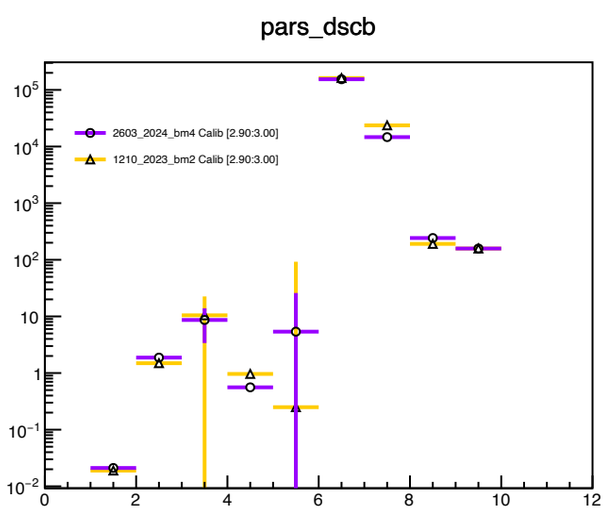
DSCB par

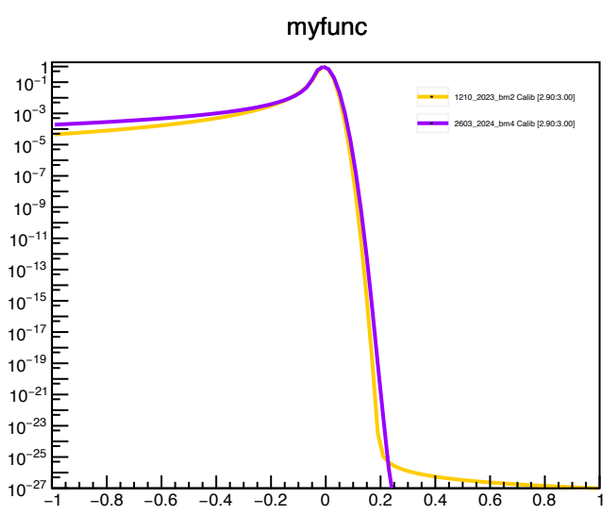








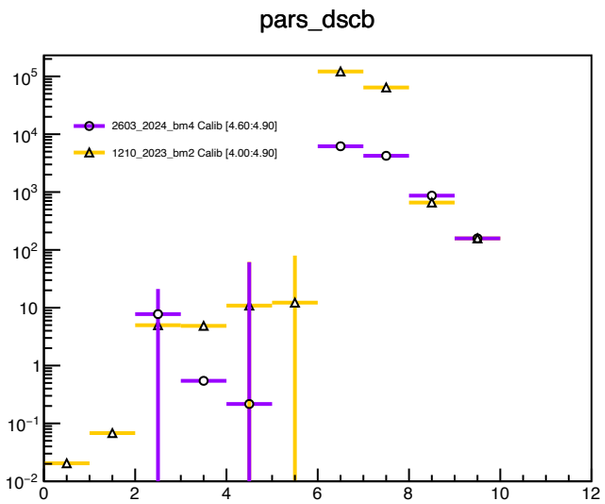


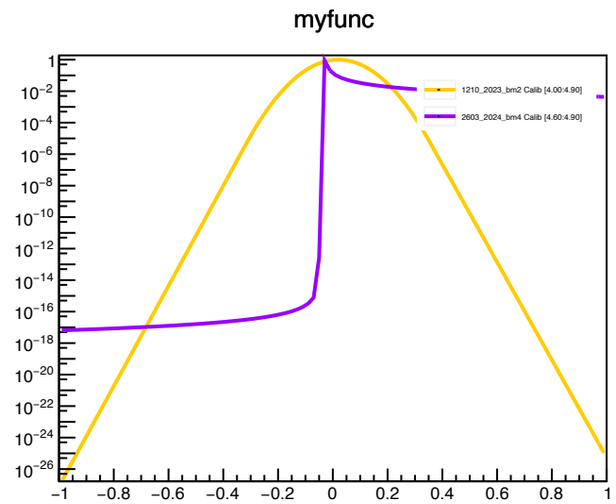


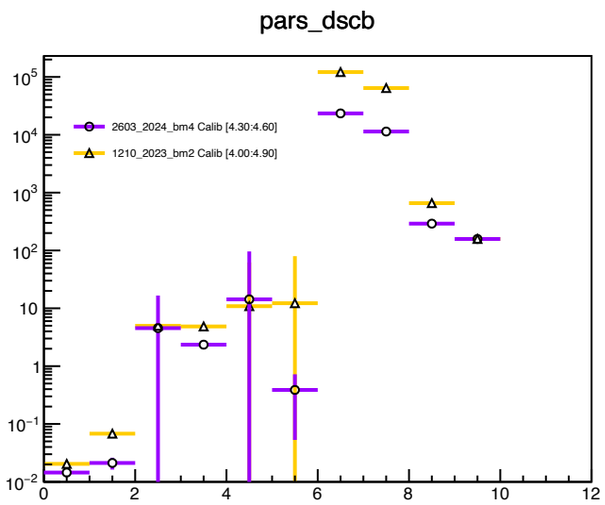

```
h_pars_dscb->SetBinContent(1, x0->getVal());
h_pars_dscb->SetBinError(1, x0->getError());

h_pars_dscb->SetBinContent(2, sigma->getVal());
h_pars_dscb->SetBinError(2, sigma->getError());

h_pars_dscb->SetBinContent(3, alpha_low->getVal());
h_pars_dscb->SetBinError(3, alpha_low->getError());

h_pars_dscb->SetBinContent(4, alpha_high->getVal());
h_pars_dscb->SetBinError(4, alpha_high->getError());

h_pars_dscb->SetBinContent(5, logn_low->getVal());
h_pars_dscb->SetBinError(5, logn_low->getError());

h_pars_dscb->SetBinContent(6, logn_high->getVal());
h_pars_dscb->SetBinError(6, logn_high->getError());

//************************
h_pars_dscb->SetBinContent(9, chi2.first);
h_pars_dscb->SetBinContent(10, chi2.second);

//RooRealVar* nsig;
if ((nsig = (RooRealVar*)params.find("nsig_conv")))
{
  h_pars_dscb->SetBinContent(7, nsig->getVal());
  h_pars_dscb->SetBinError(7, nsig->getError());
}

//RooRealVar* nbkg;
if ((nbkg = (RooRealVar*)params.find("nbkg_conv")))
{
  h_pars_dscb->SetBinContent(8, nbkg->getVal());
  h_pars_dscb->SetBinError(8, nbkg->getError());
}
```

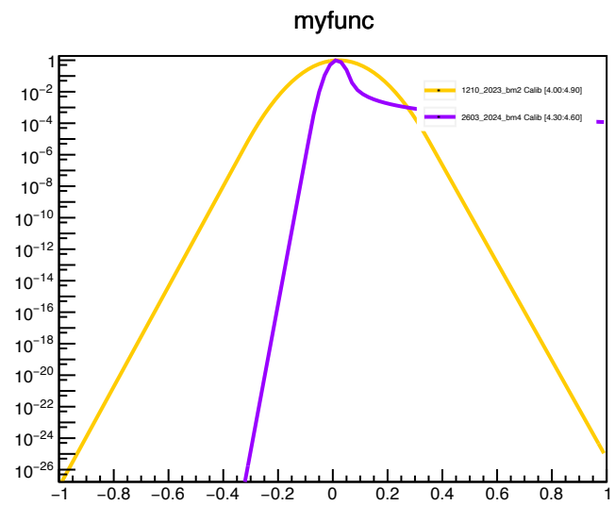
code

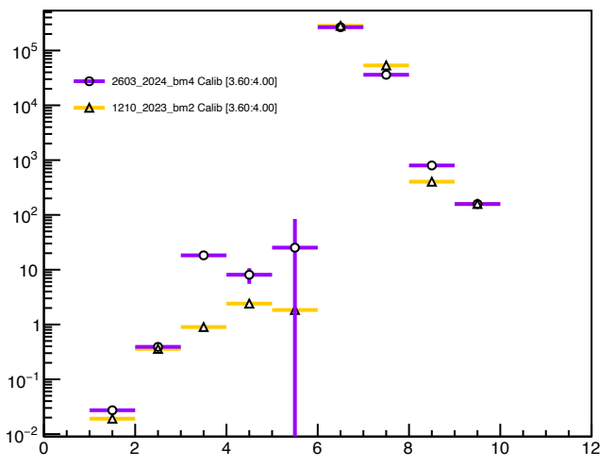
DSCB parameters

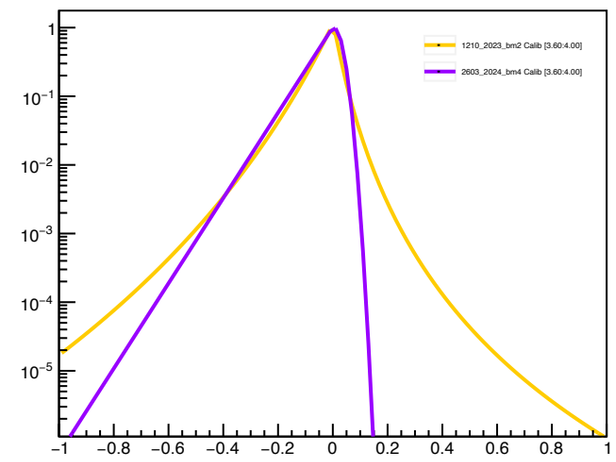




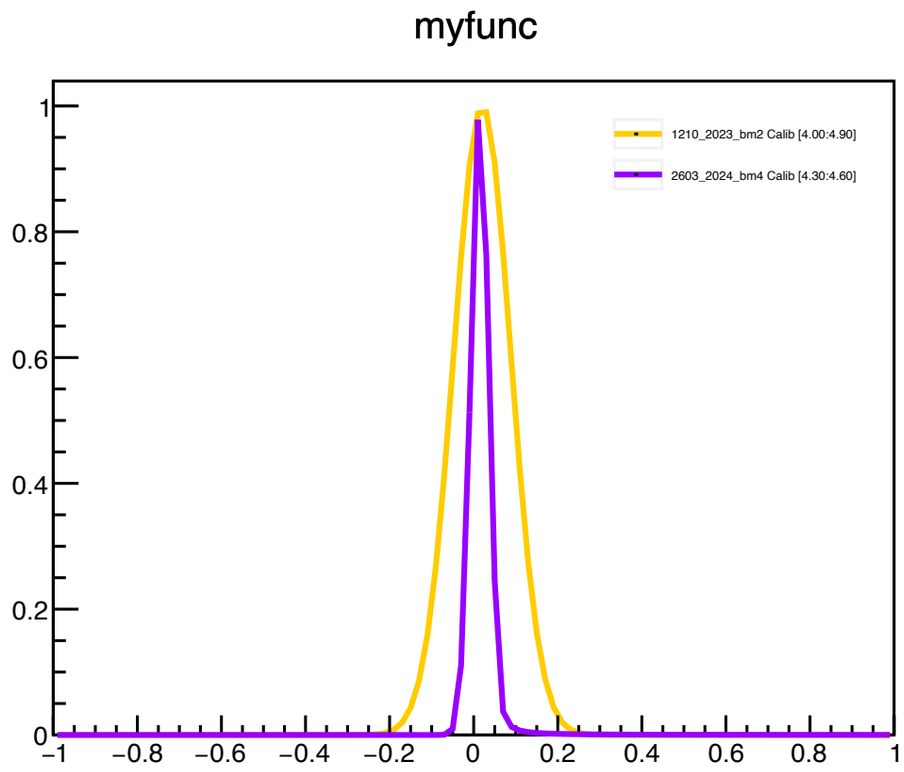




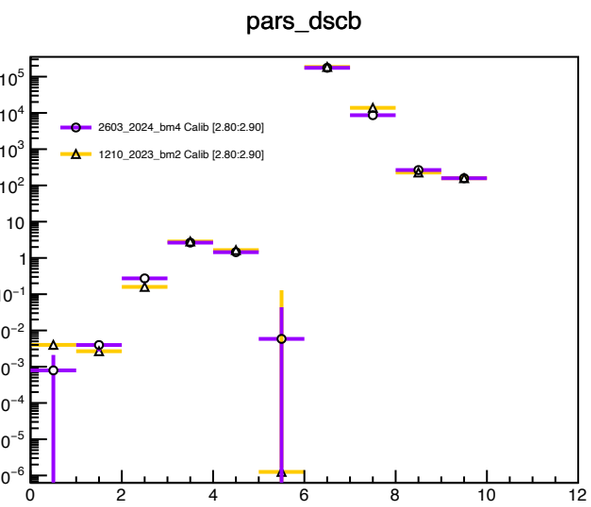


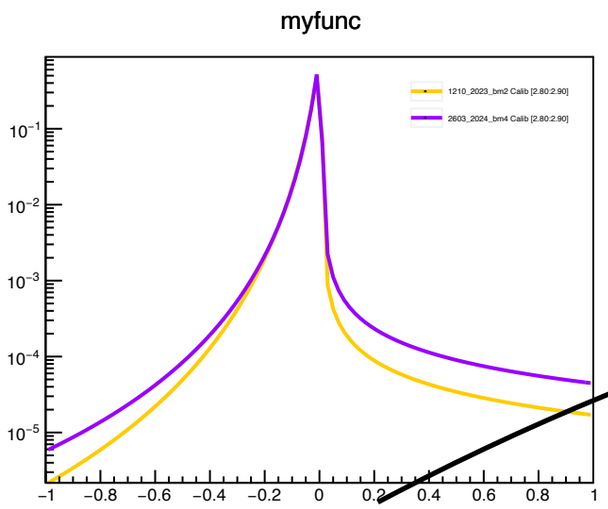


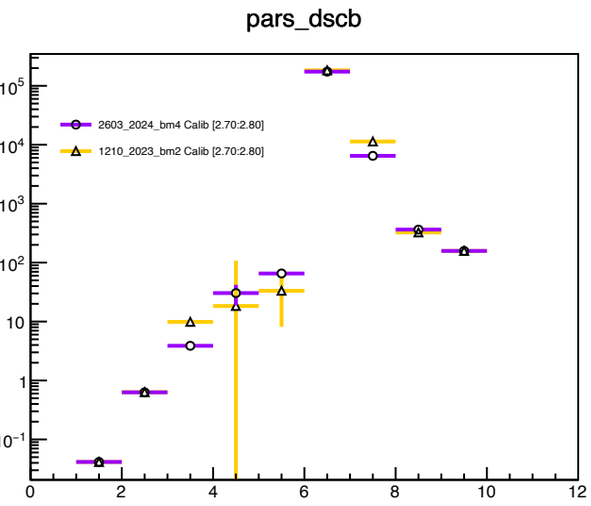
We cna not fit small bins in the FCAL !

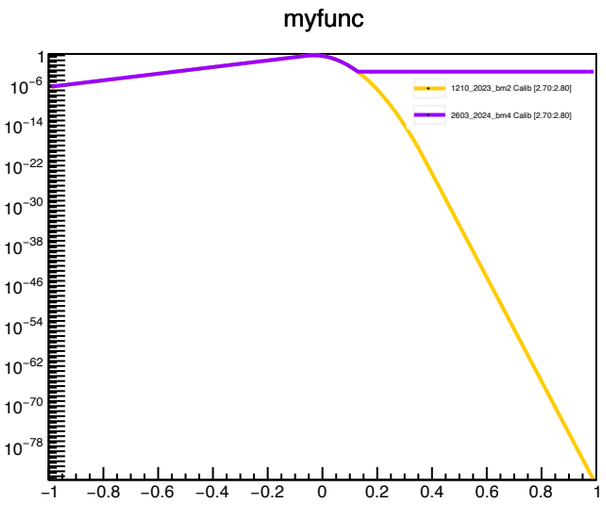


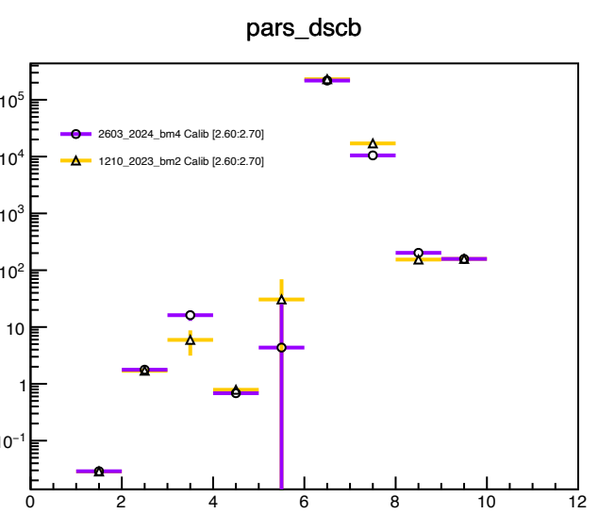
DSCB par

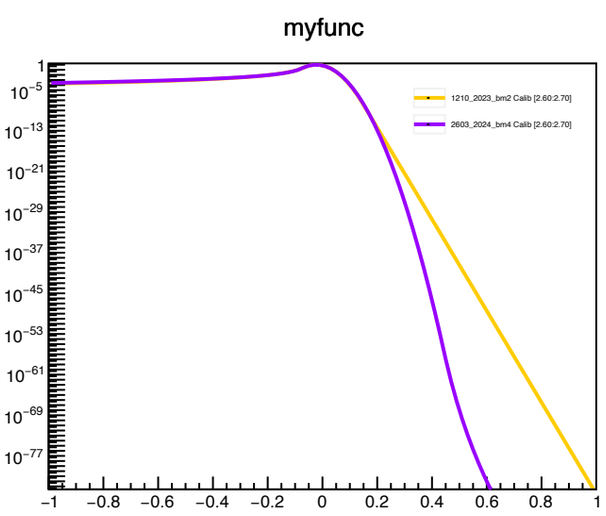




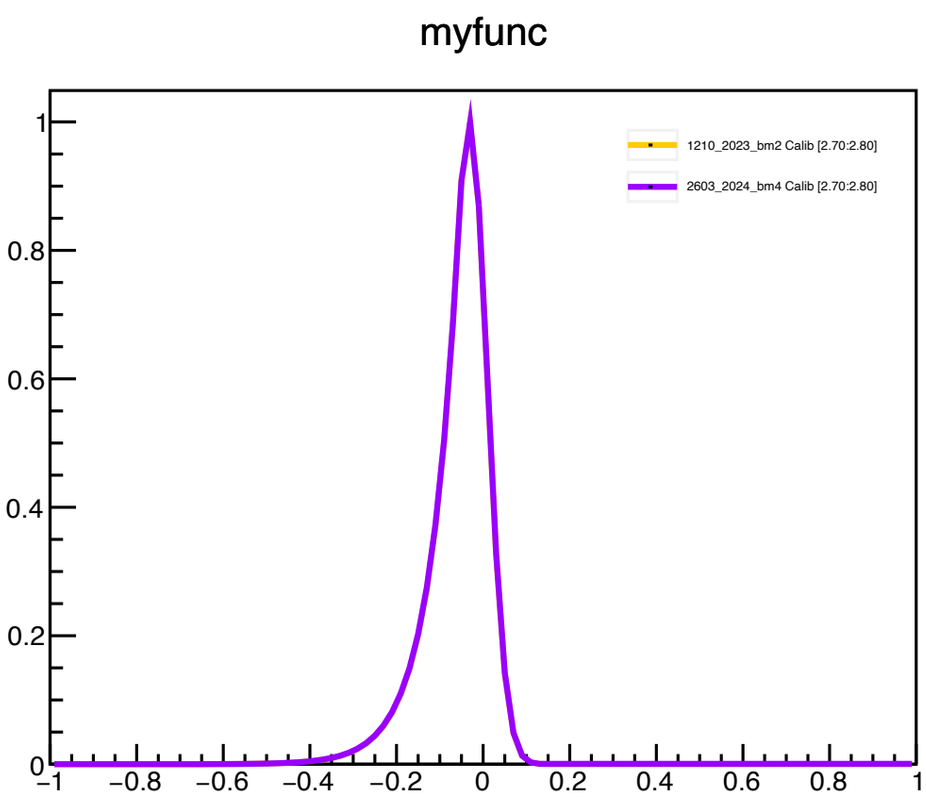




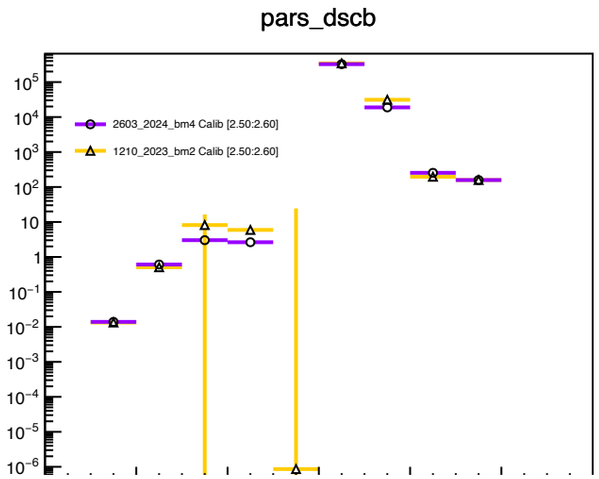


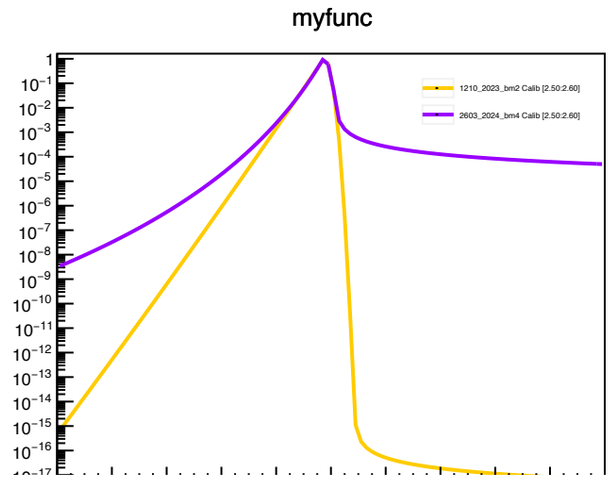


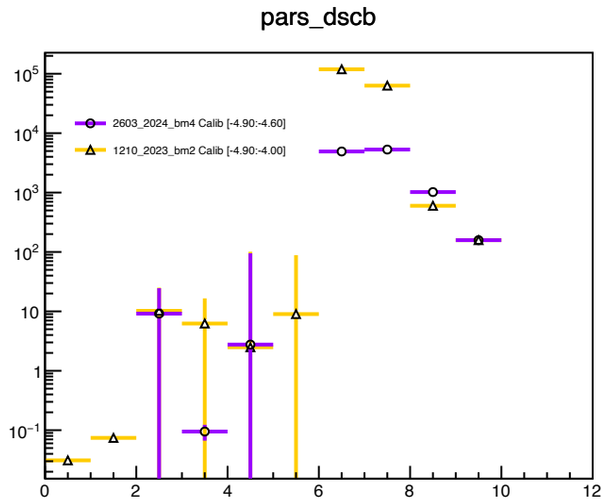
in particular the [2.7-2.8] seems very similar ?

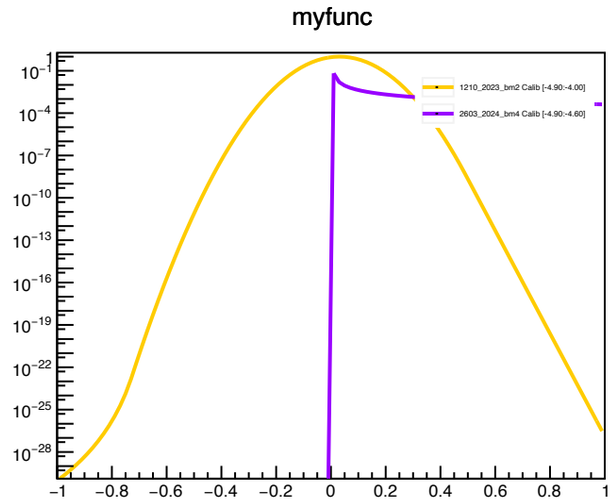


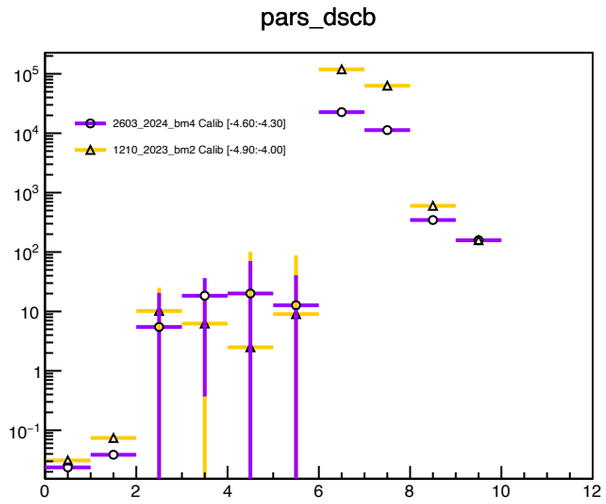
DSCB comparisons

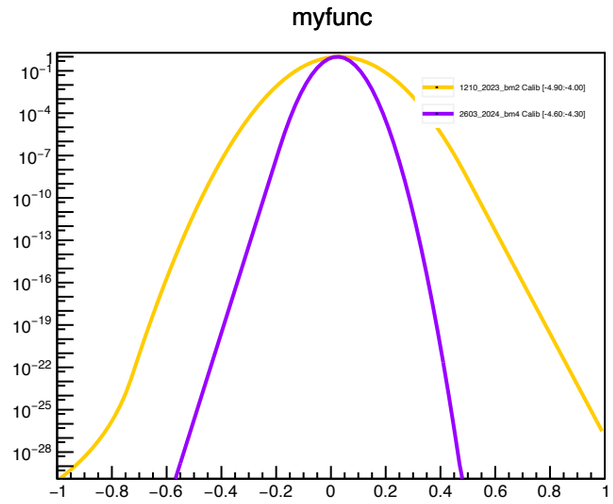






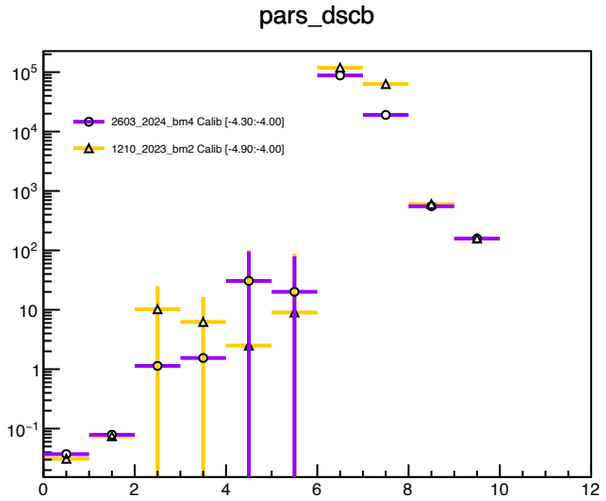


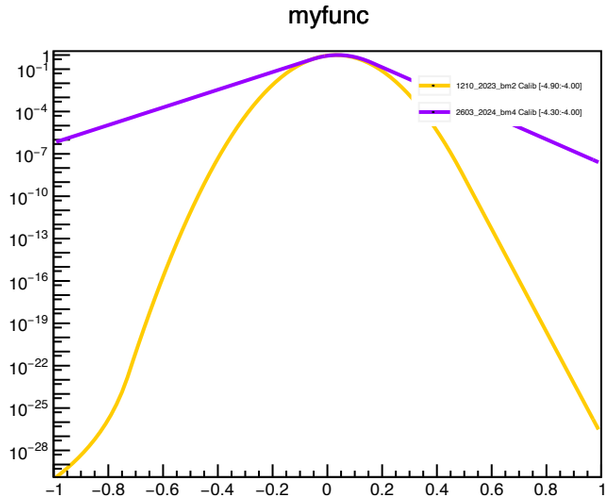


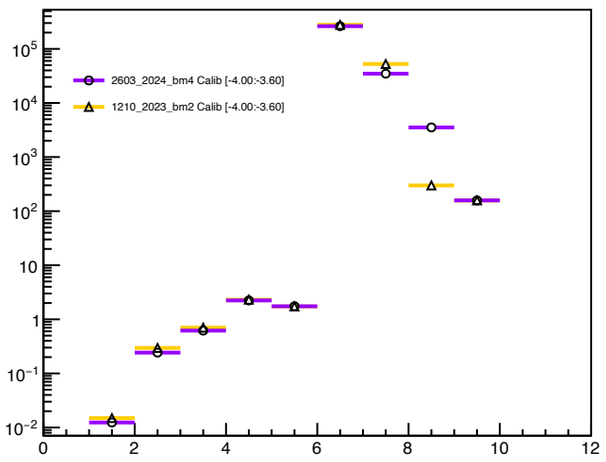


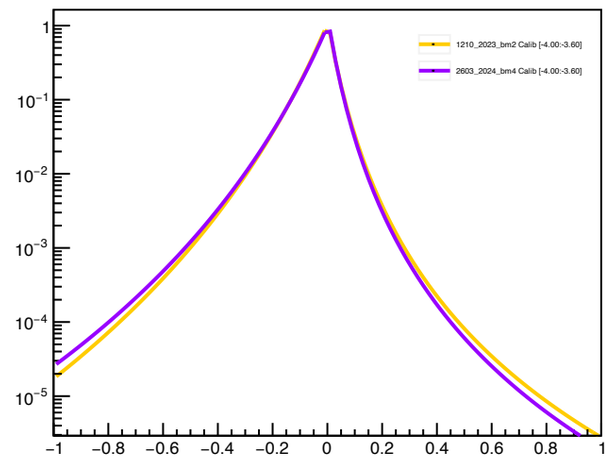
We cna not fit small bins in the FCAL !

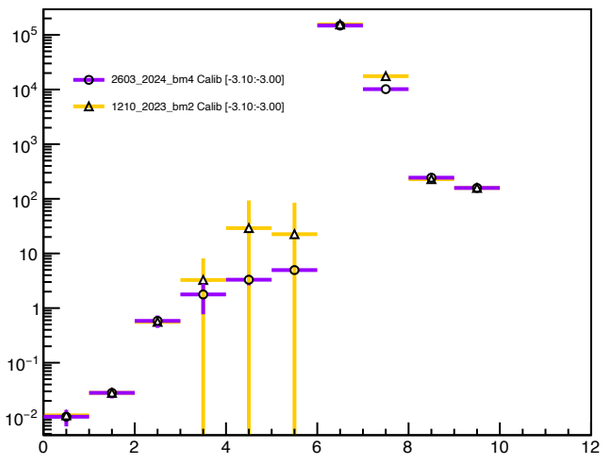
DSCB comparisons

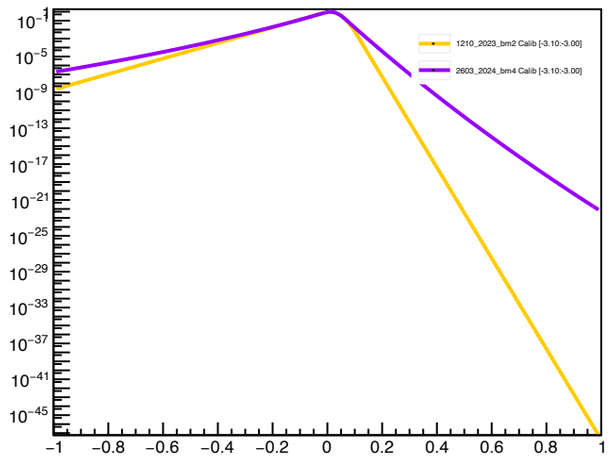




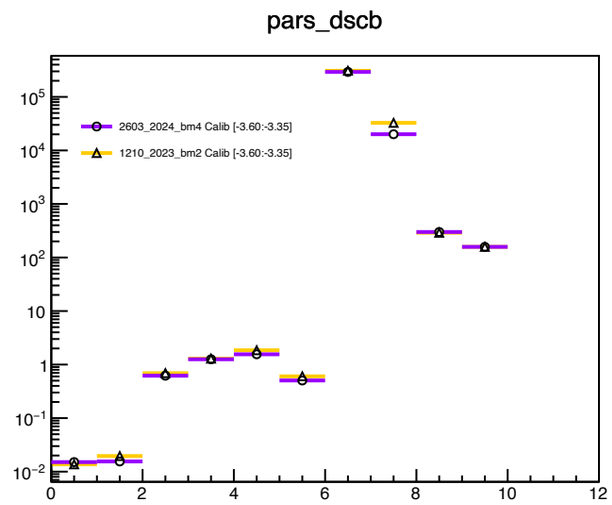


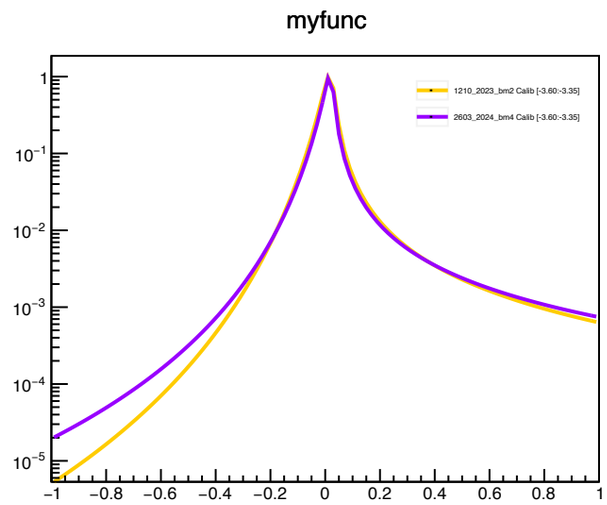


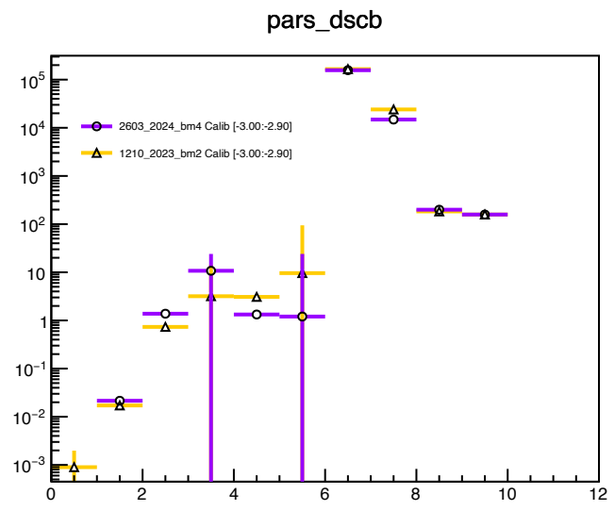


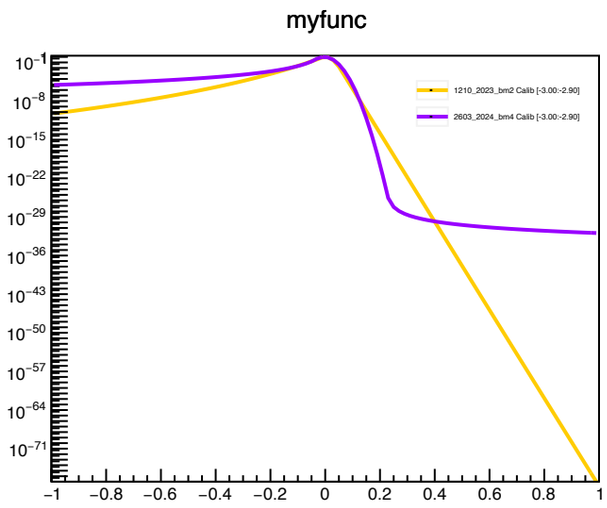


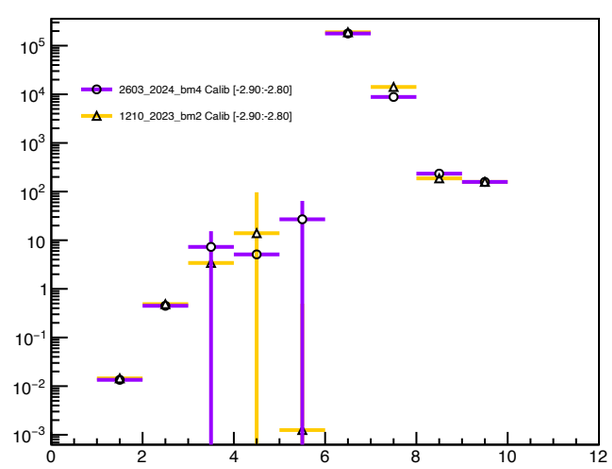
DSCB comparisons

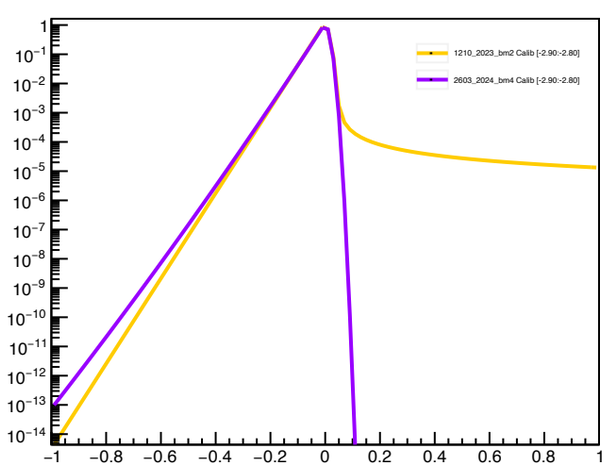




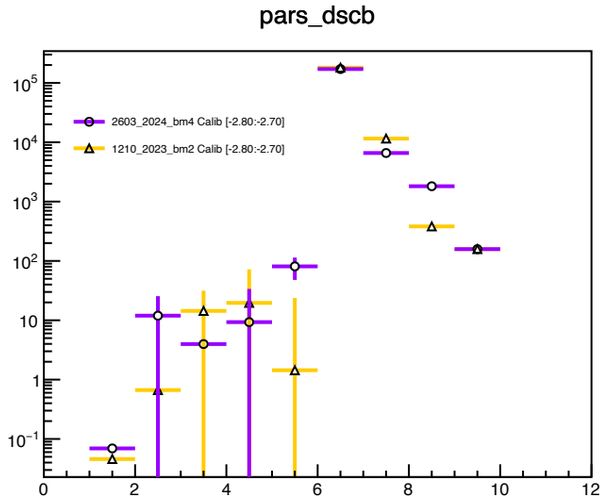


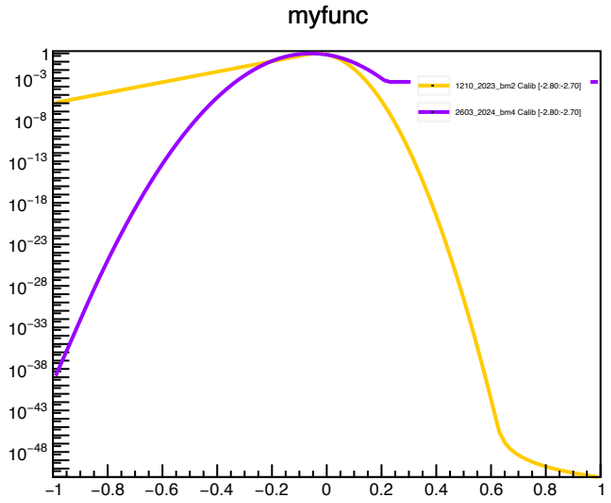


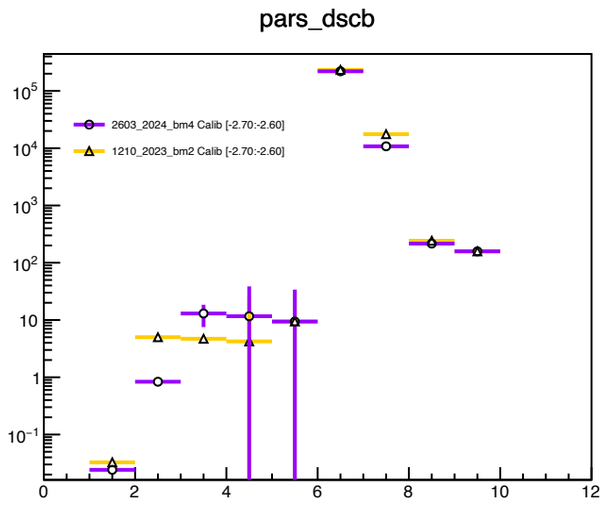


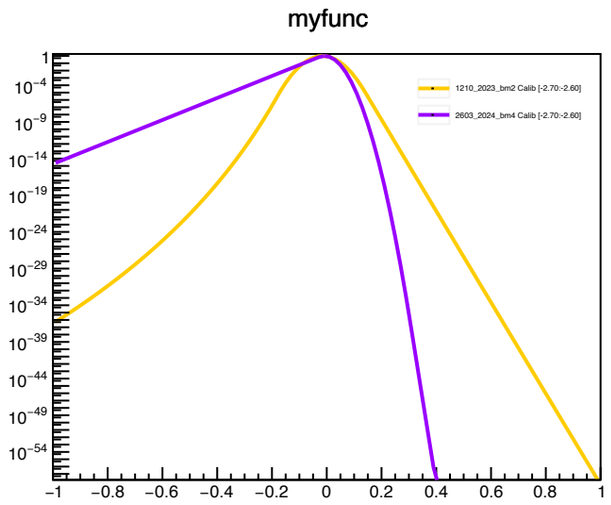


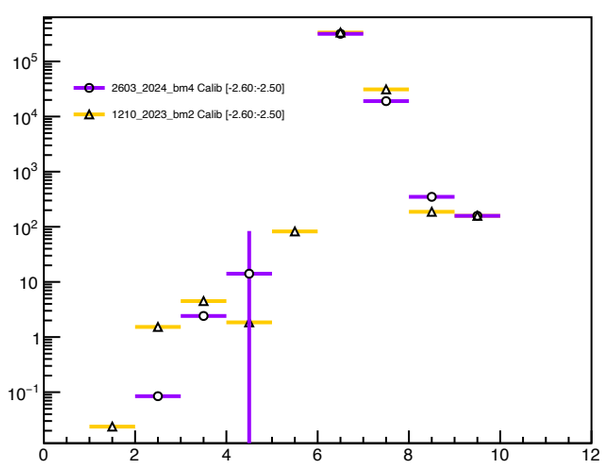
DSCB comparisons

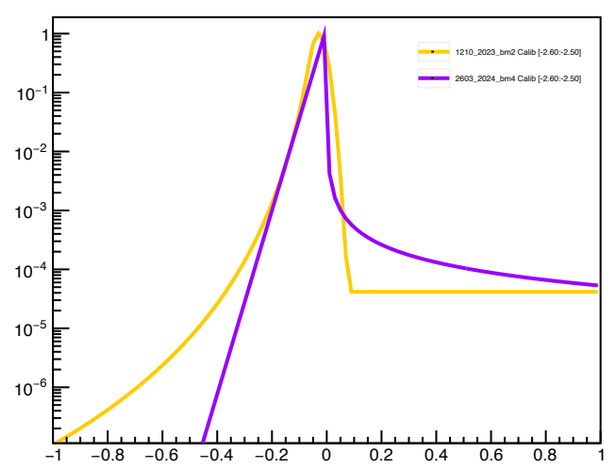




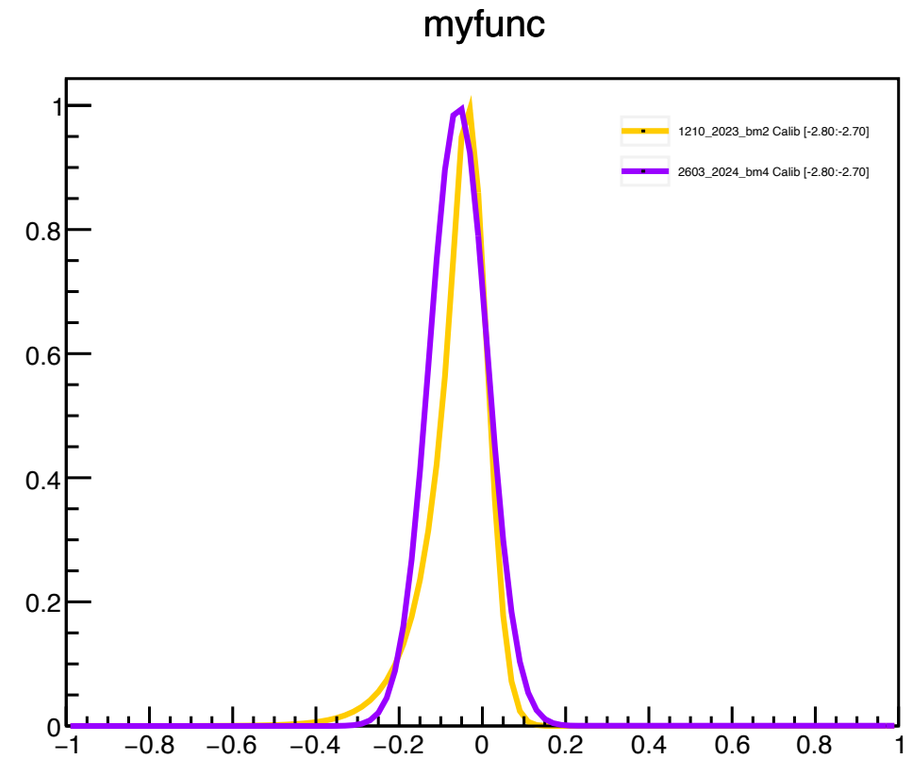






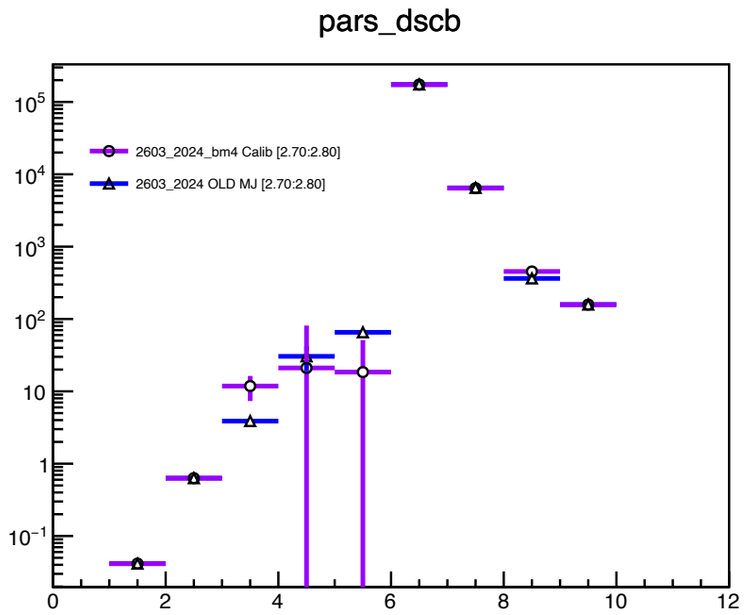


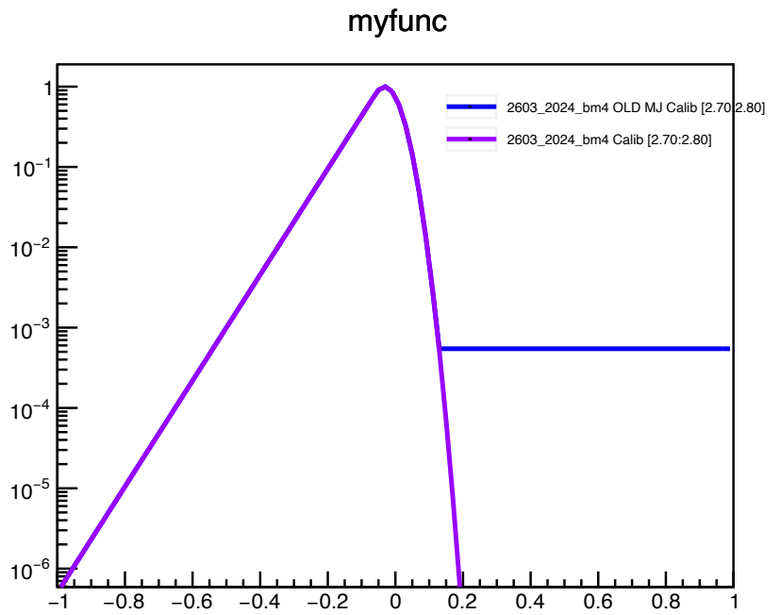
the [-2.8-2.7] seems very different !

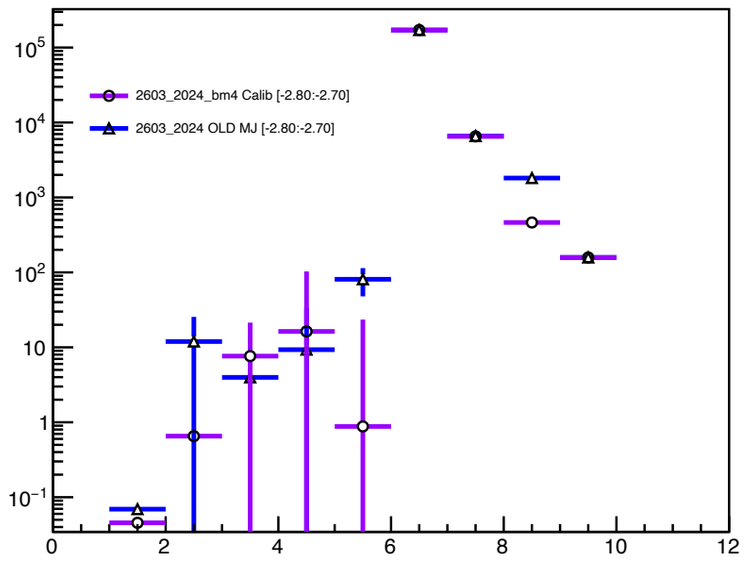


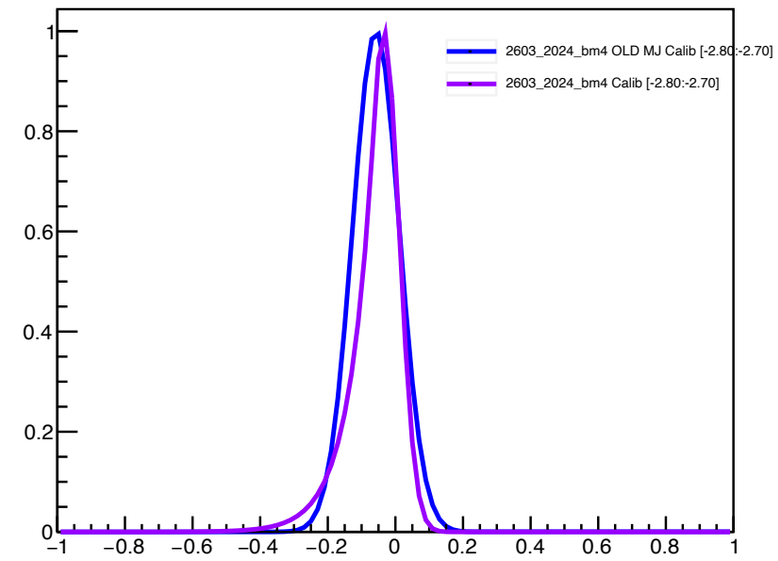
terate with better MJ (2024 calib

- using : /nfs/dust/atlas/user/wellscj/ZAi/eeCF/alma9/aidy/fwd-corrections/calib-files/CF_DSCB_preCalibAlignSF_FFEta6Pt0_M10_UMJ_LukasConfig_17_2603_2024_bm4_p1_x0_t1/Nominal/





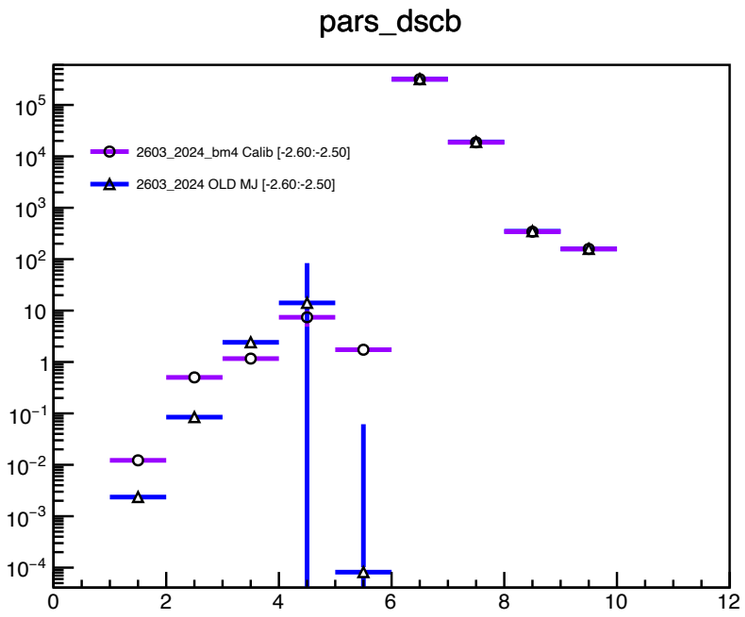


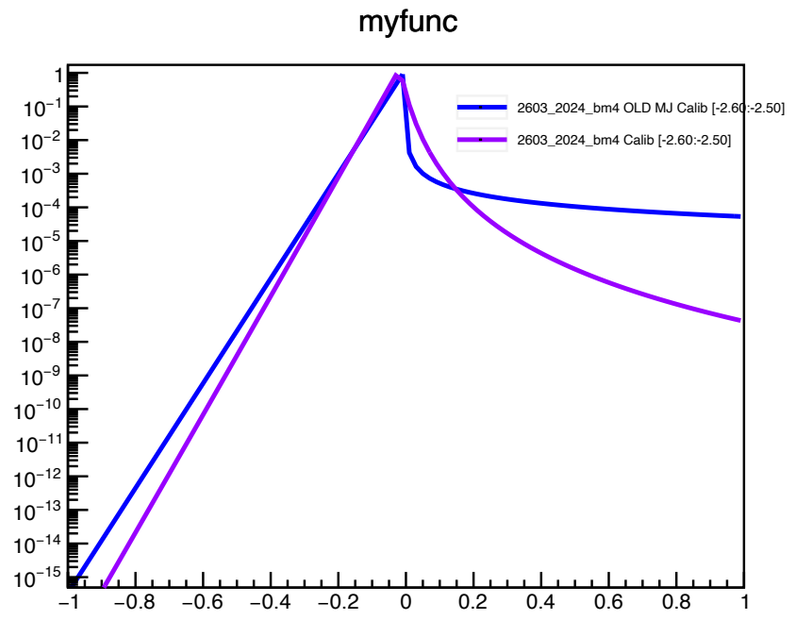


better MJ estimate reduce the high tails

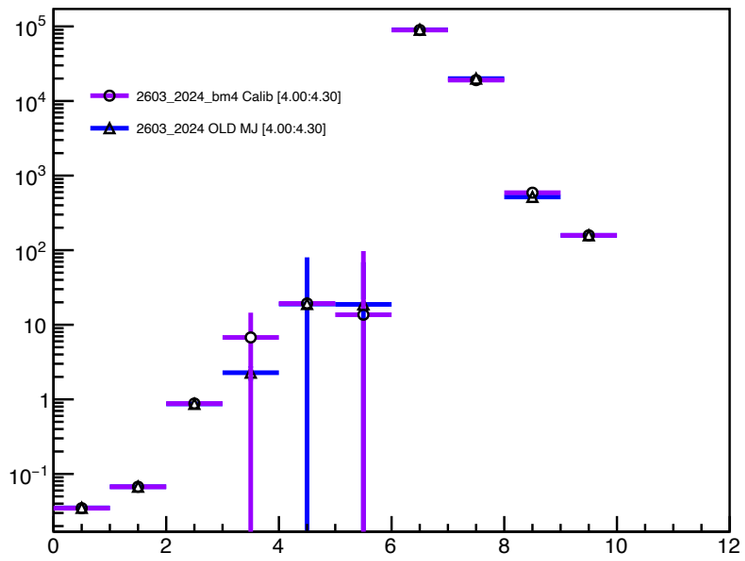
but also could shift the peak

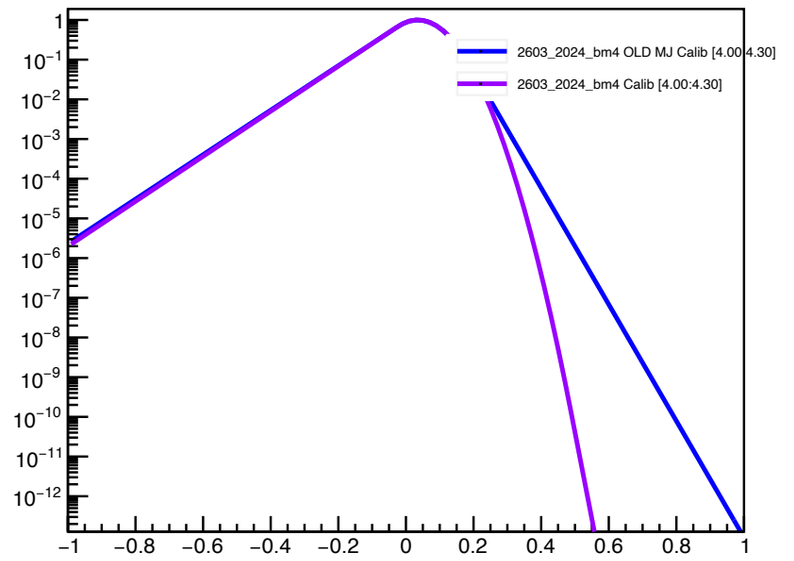
2024 calib



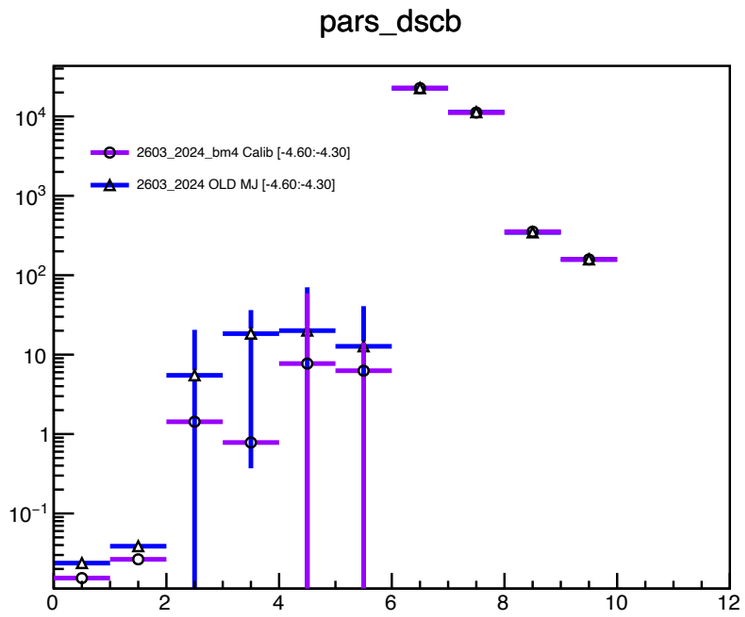


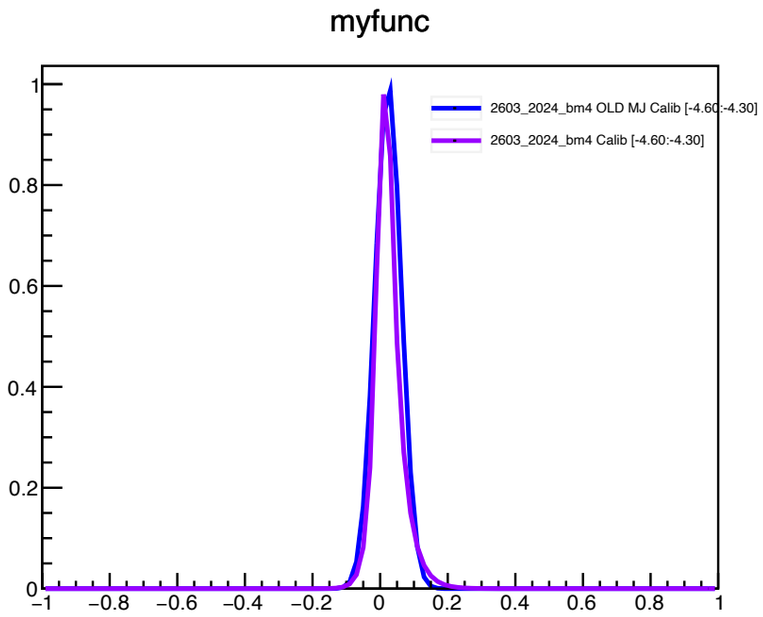
better MJ estimate
reduce the high tails

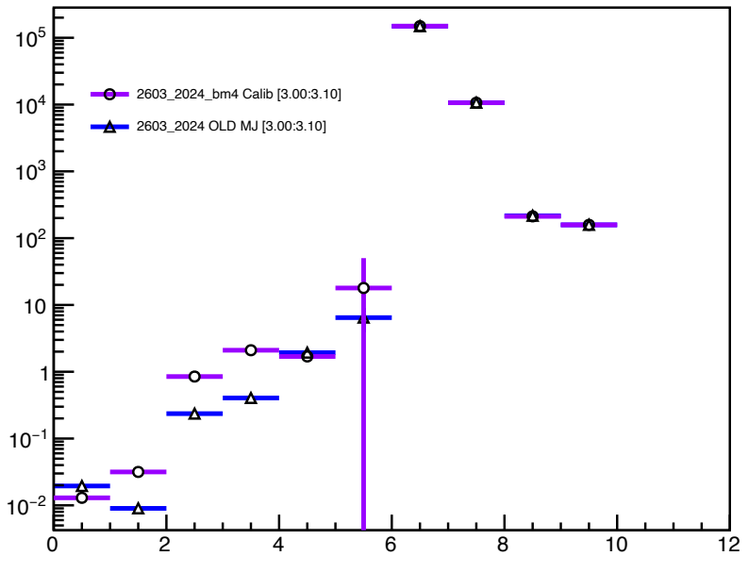


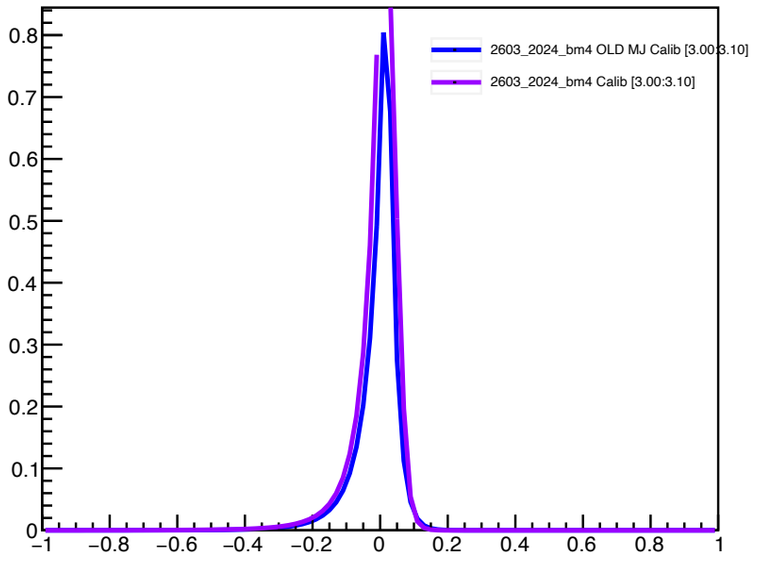


2024 calib









better MJ estimate reduce the high tails

but also could shift the peak

conclusion

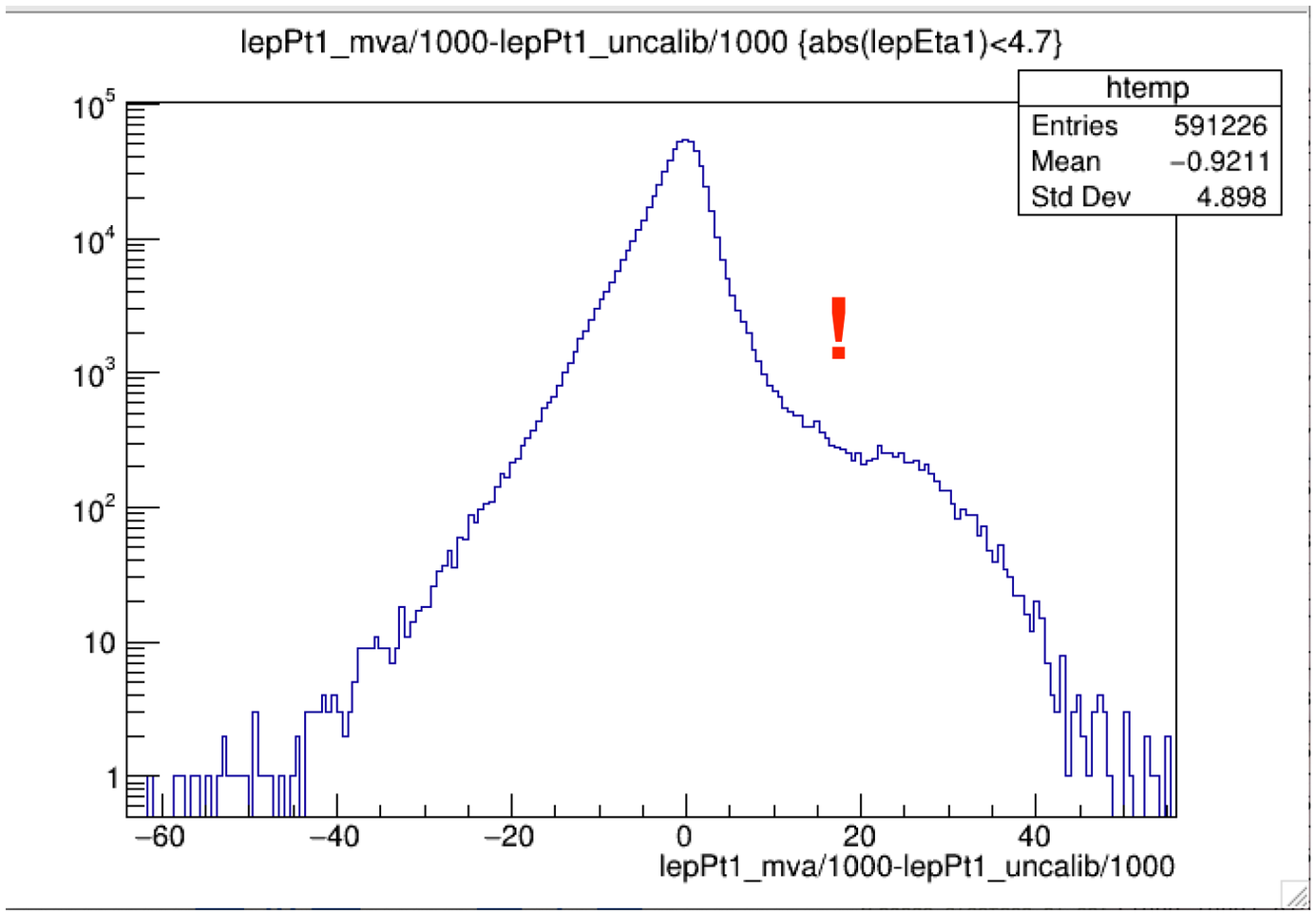
- even if there are quite some differences between the calibration campaign 2023 vs 2024 I dont see how this would be the cause of what lukas is seeing ?

```
if (doDSCB){
  //cout << "Entering pars retrieval for DSCB " << endl;

  for (int i=0;i<6;i++) m_pars[ieta][i] = h->GetBinContent(i+1);
  //cout << "Retrieved pars" << endl;

  // Minimal alpha values
  m_pars[ieta][2] = max(m_pars[ieta][2], 0.01);
  m_pars[ieta][3] = max(m_pars[ieta][3], 0.01);

  m_pars[ieta][4] = min(m_pars[ieta][4], 100.); // cap logn to avoid numerical instability original
  m_pars[ieta][5] = min(m_pars[ieta][5], 100.); // cap logn to avoid numerical instability original

  //cout << "Capped log n" << endl;
  m_pars[ieta][4] = exp(m_pars[ieta][4]); // convert logn to n
  //cout << "Converted n low" << endl;
  m_pars[ieta][5] = exp(m_pars[ieta][5]); // convert logn to n
  //cout << "Converted n high" << endl;

  m_pars[ieta][6] = -5; //xtrunc_L
  m_pars[ieta][7] = 5; //xtrunc_R

  std::string DSCB_name = "dscb_" + std::to_string(ieta);
  std::string DSCB_x_name = "x" + std::to_string(ieta);
  std::string DSCB_mu_name = "mu" + std::to_string(ieta);
  std::string DSCB_sigma_name = "sigma" + std::to_string(ieta);
  std::string DSCB_alphaLo_name = "alphaLo" + std::to_string(ieta);
  std::string DSCB_alphaHi_name = "alphaHi" + std::to_string(ieta);
  std::string DSCB_nLo_name = "nLo" + std::to_string(ieta);
  std::string DSCB_nHi_name = "nHi" + std::to_string(ieta);
```

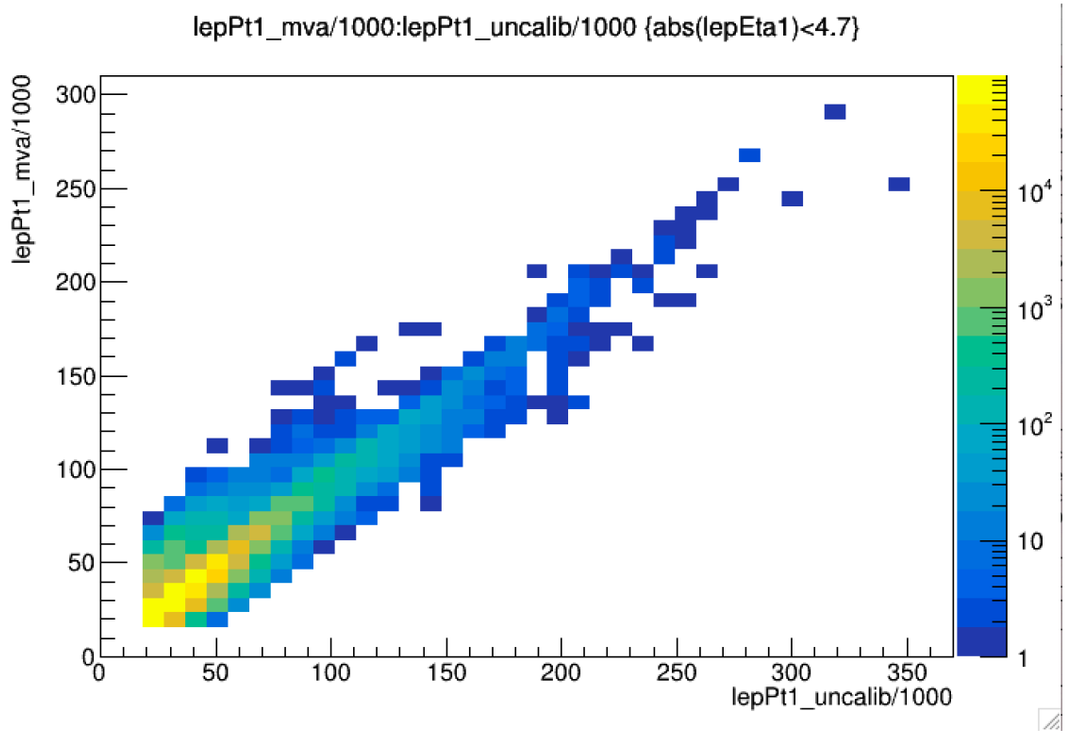
code

MVA

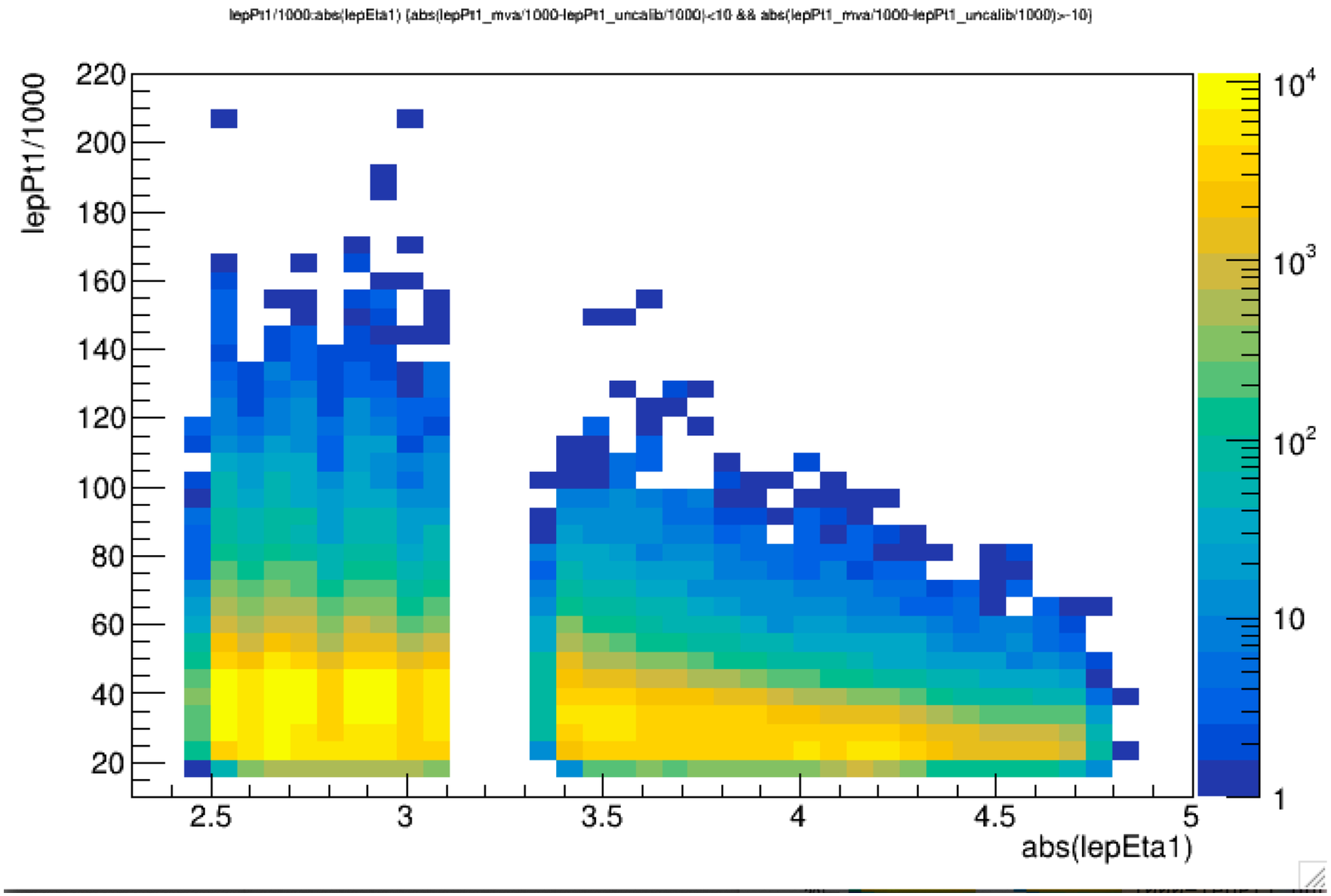
How much MVA move Et of the FWD electron?



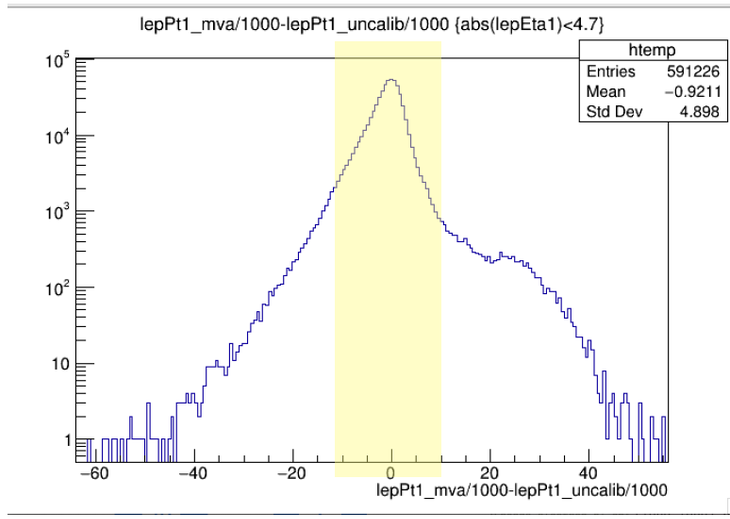
Fwd electron energy moved more then 10GeV



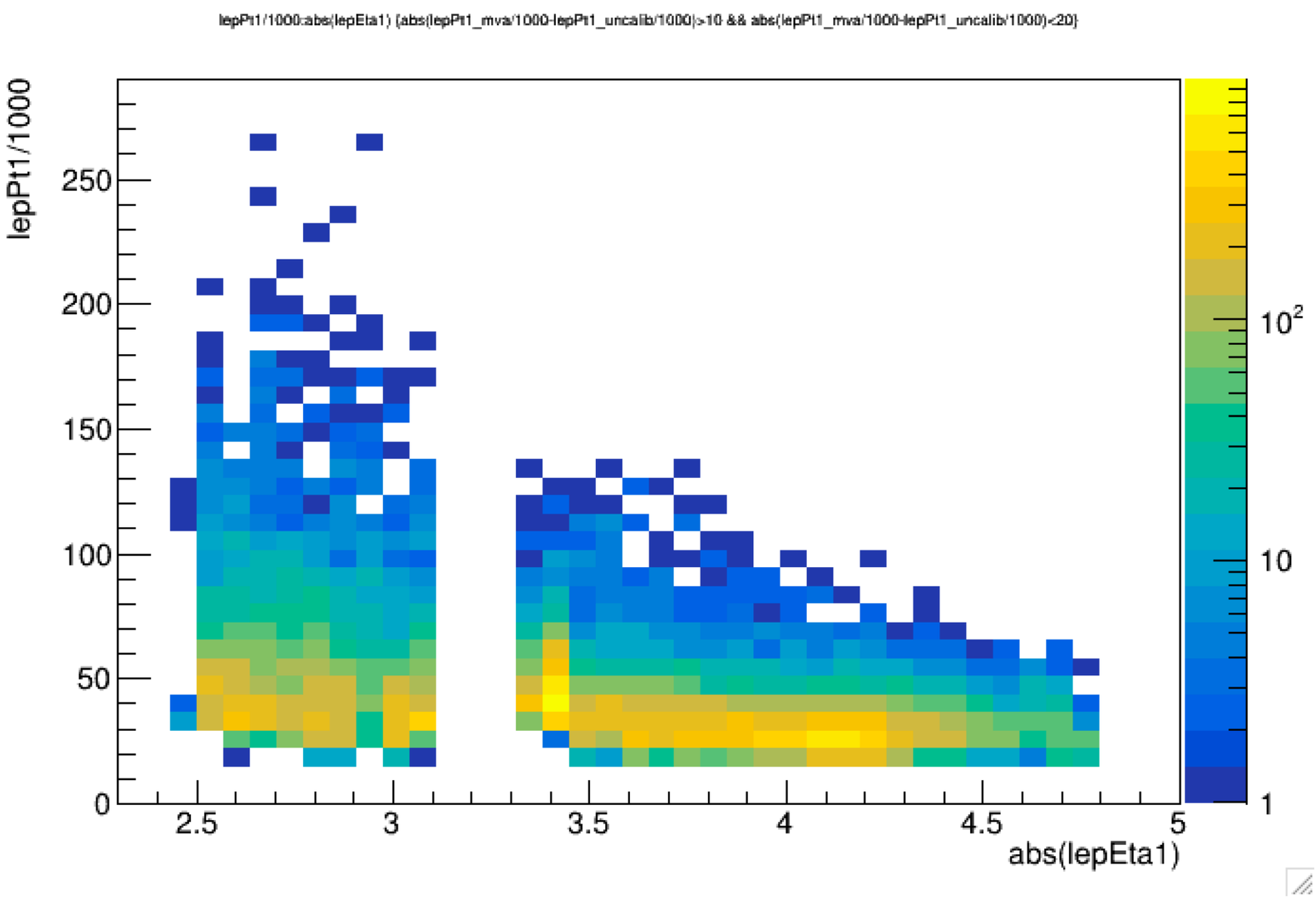
How much MVA move Et of the FWD electron?



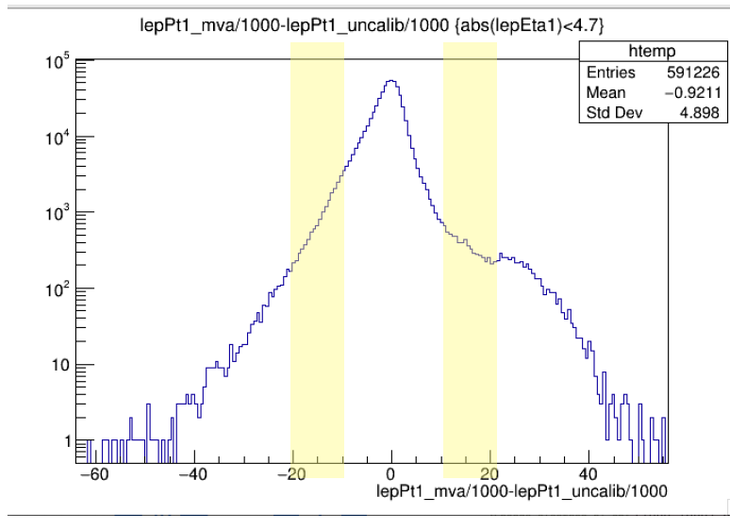
where the MVA corrects more ?



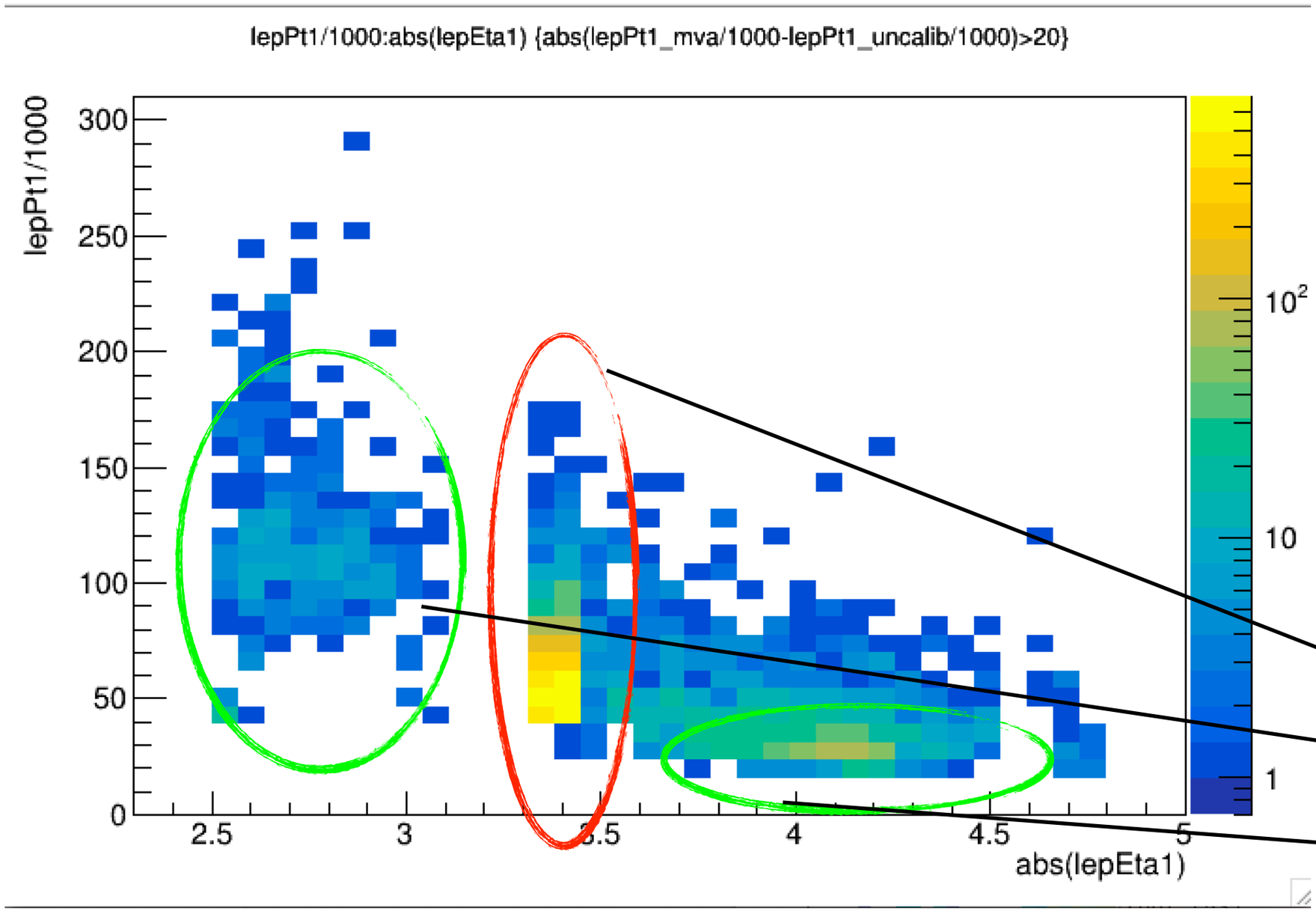
How much MVA move Et of the FWD electron



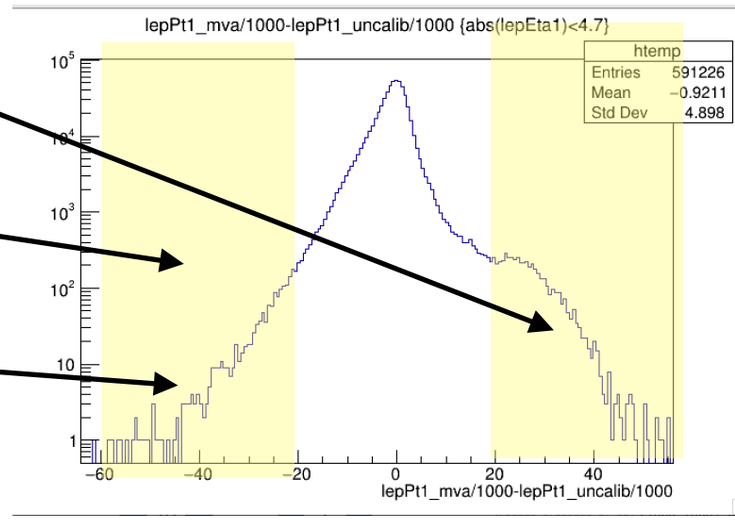
where the MVA corrects more ?



How much MVA move Et of the FWD electron

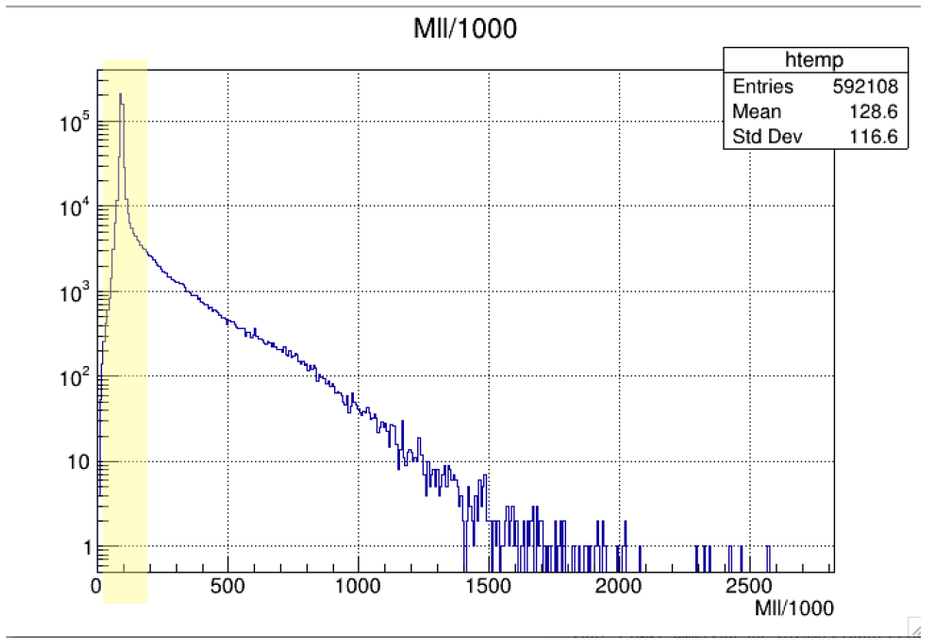


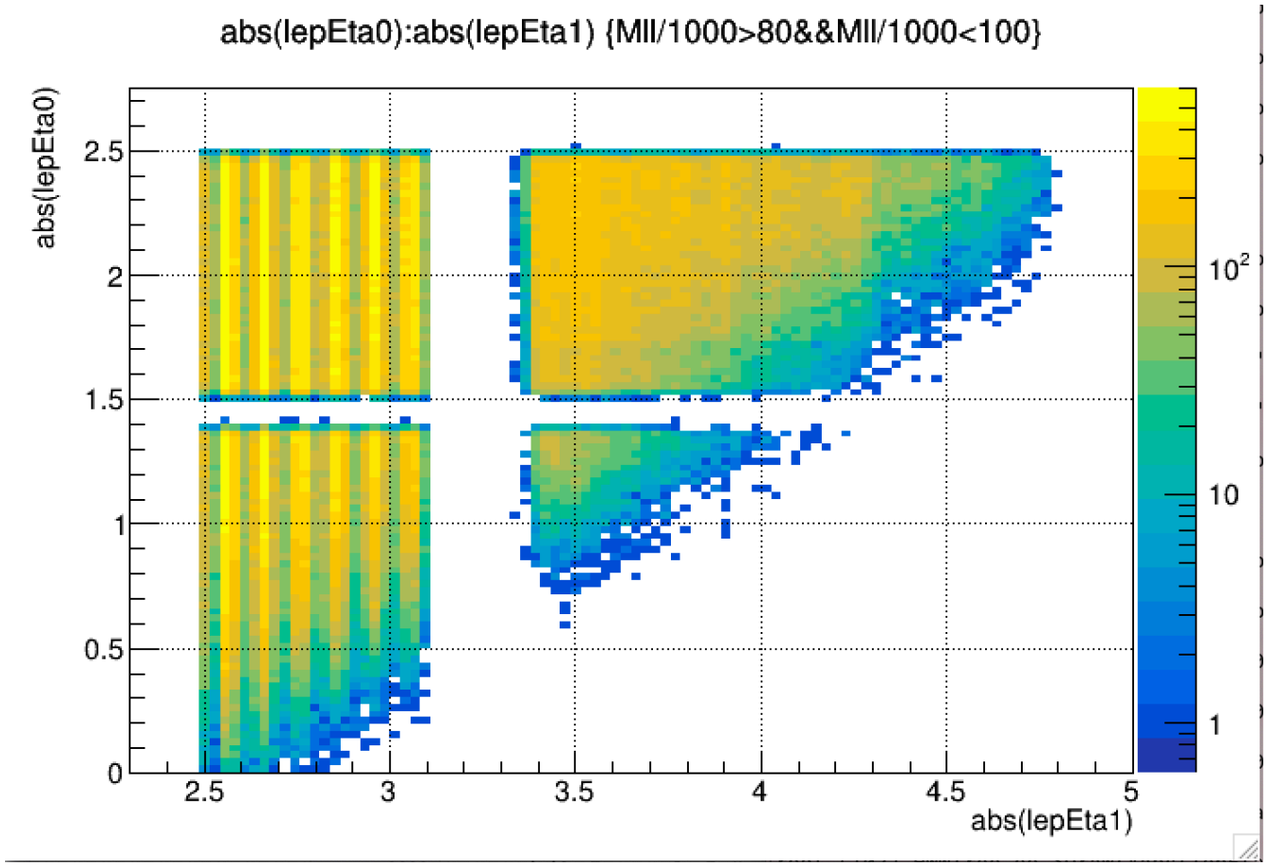
where the MVA corrects more ?

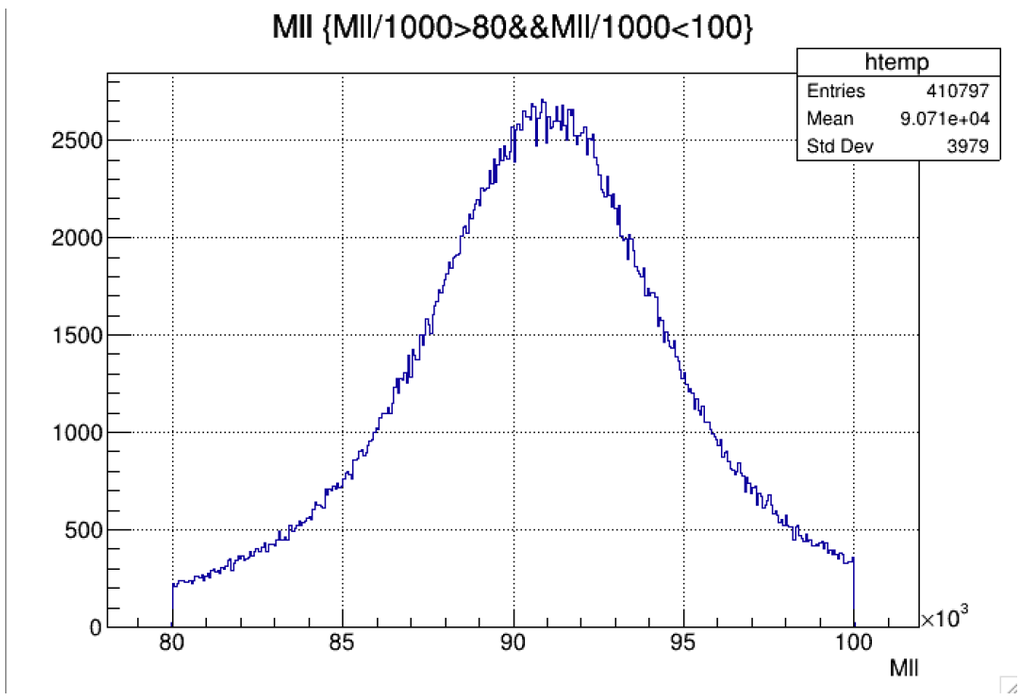


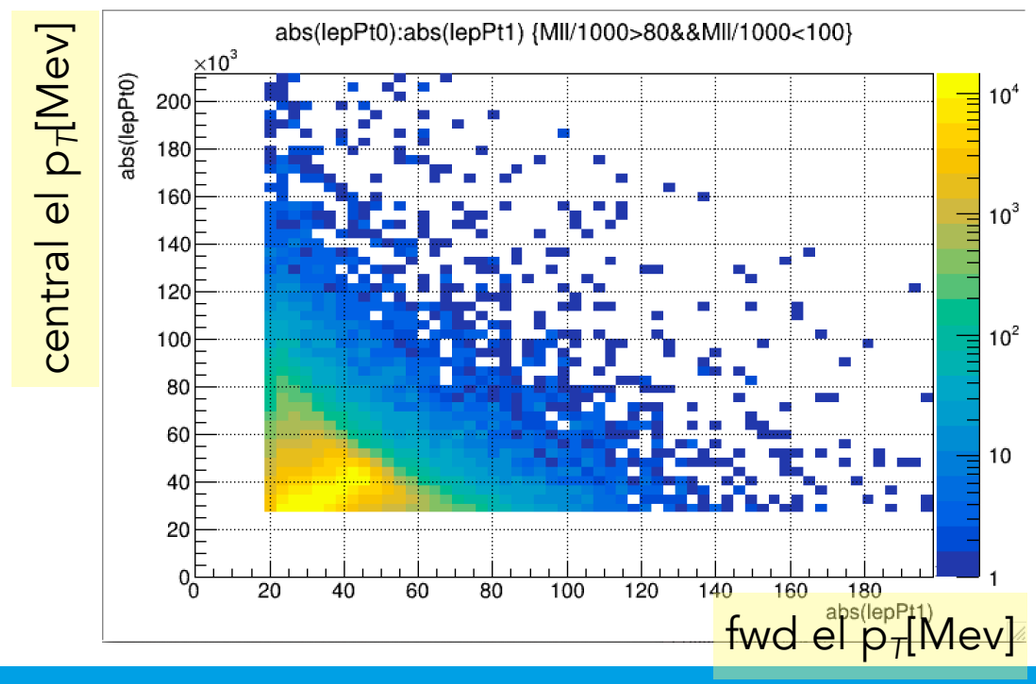
ADDING Z KINEMATICS

80<Mee(CF)<100

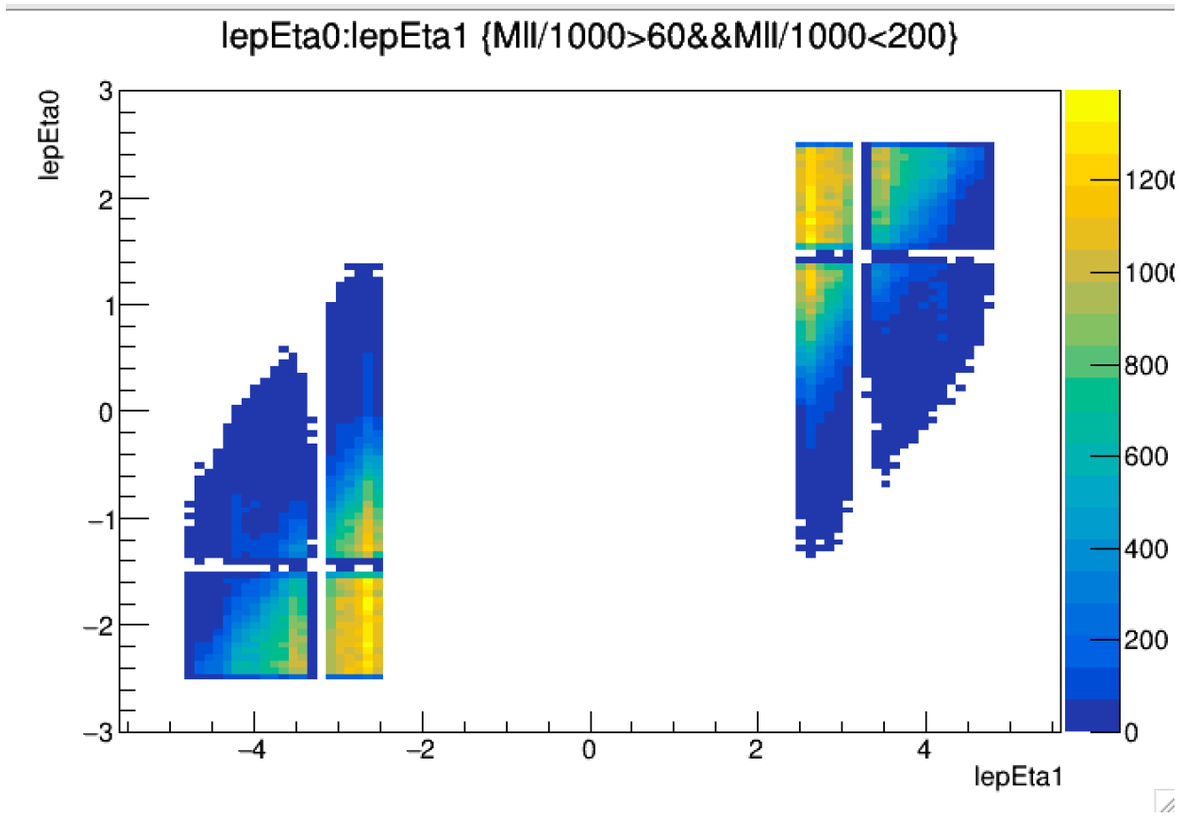

Mll/1000
htemp
Entries 592108
Mean 128.6
Std Dev 116.6
Mll/1000

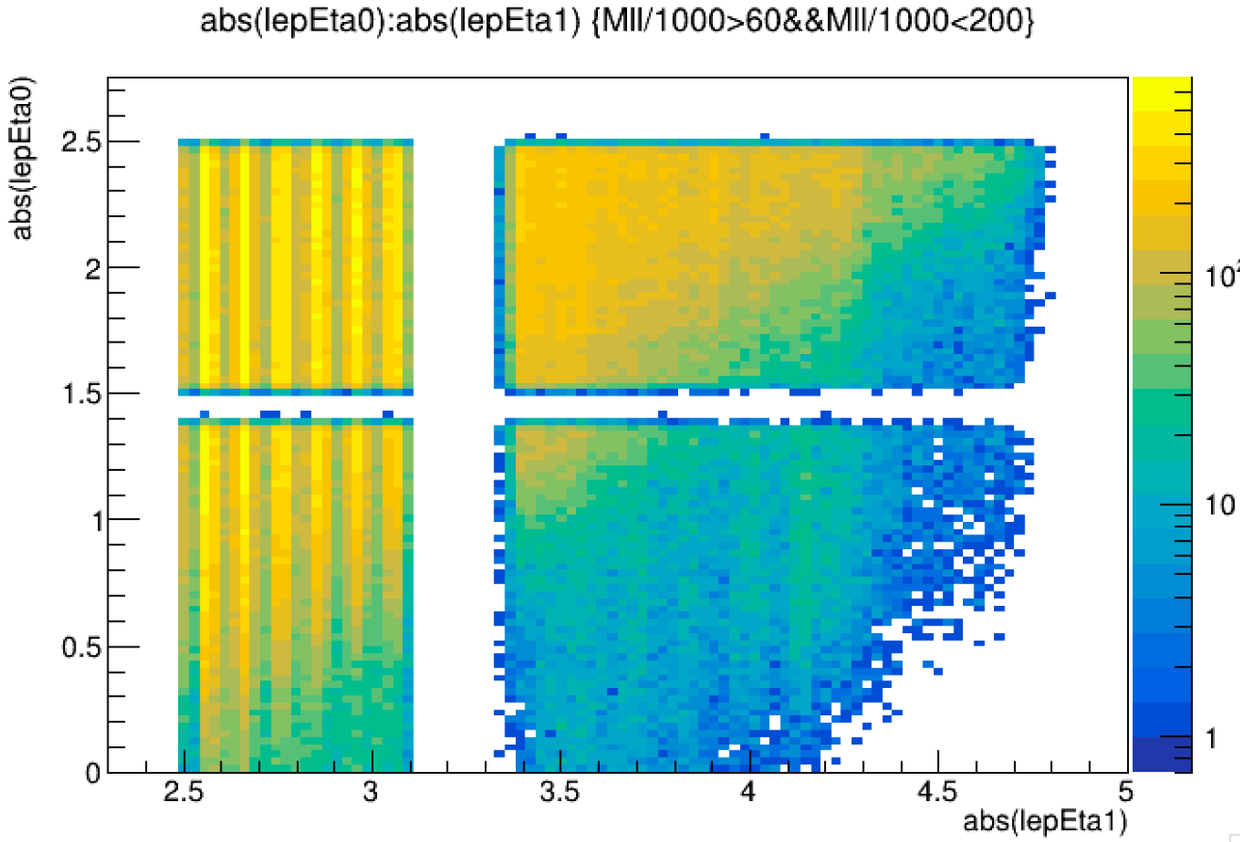

abs(lepEta0):abs(lepEta1) {Mll/1000>80&&Mll/1000<100}
abs(lepEta0)
abs(lepEta1)

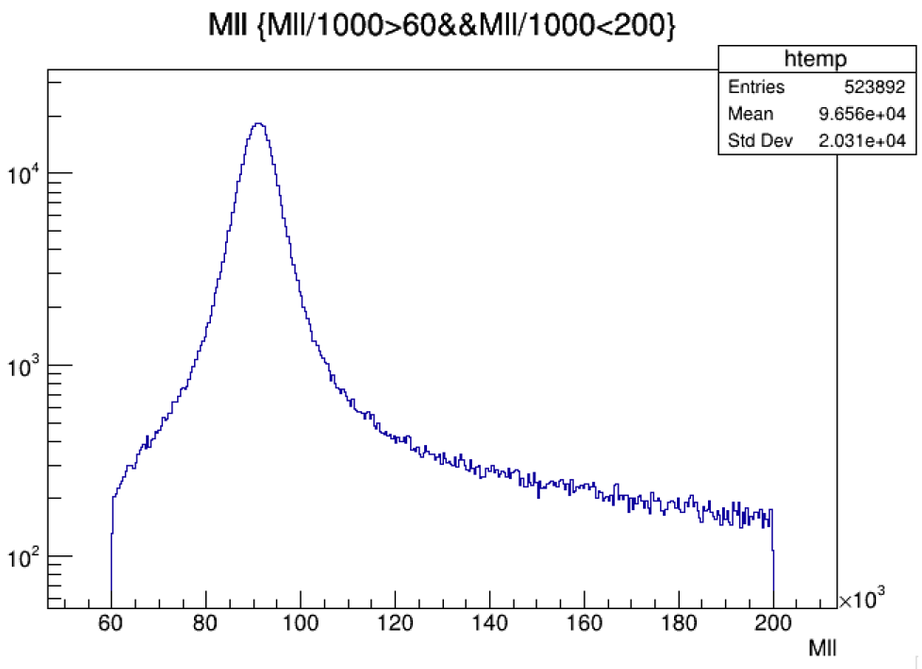

Mll {Mll/1000>80&&Mll/1000<100}
htemp
Entries 410797
Mean 9.071e+04
Std Dev 3979
Mll

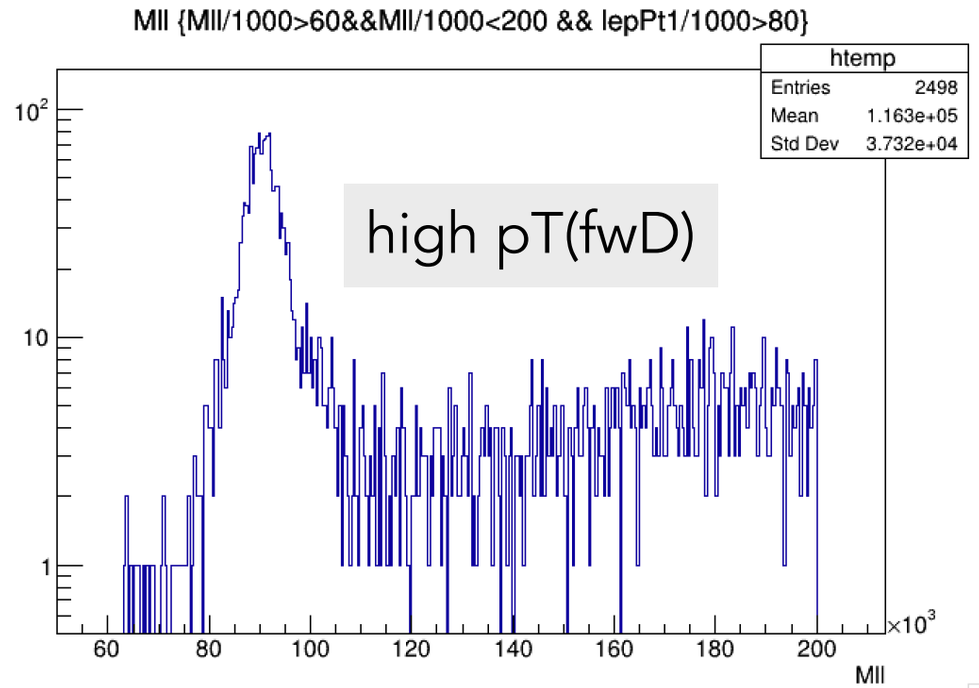

abs(lepPt0):abs(lepPt1) {Mll/1000>80&&Mll/1000<100}
central el p_T[Mev]
fwd el p_T[Mev]

60<Mee(CF)<200

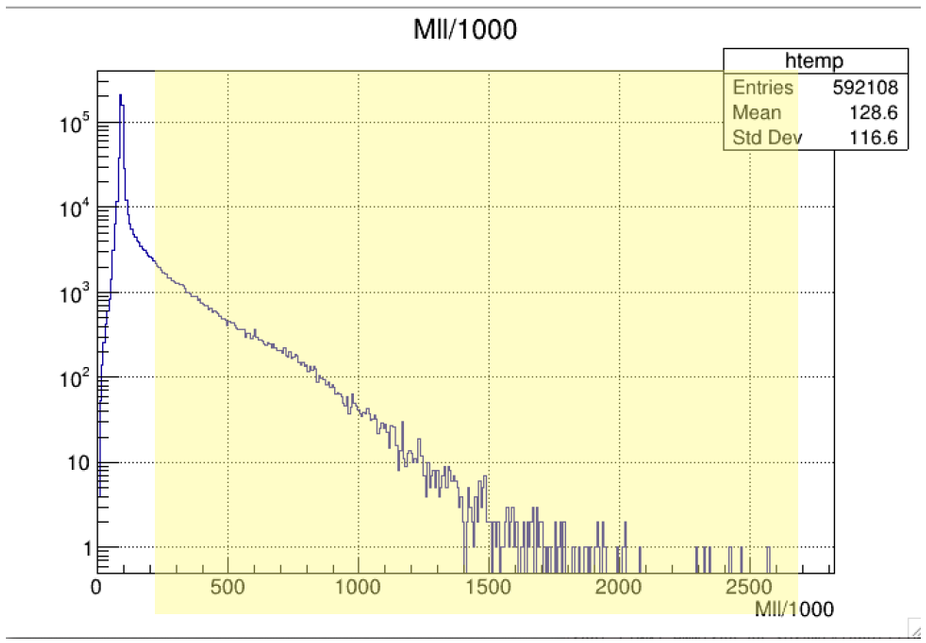

lepEta0:lepEta1 {Mll/1000>60&&Mll/1000<200}
lepEta0
lepEta1

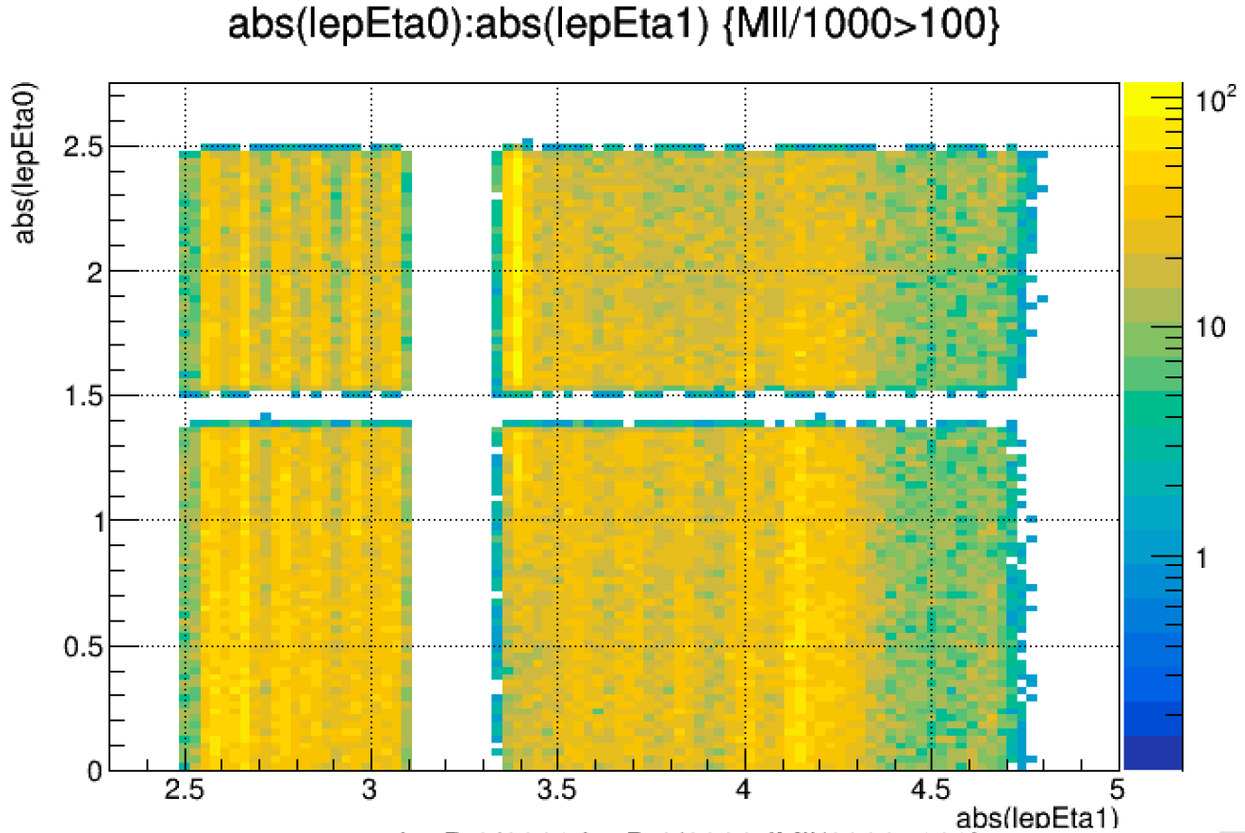

abs(lepEta0):abs(lepEta1) {Mll/1000>60&&Mll/1000<200}
abs(lepEta0)
abs(lepEta1)

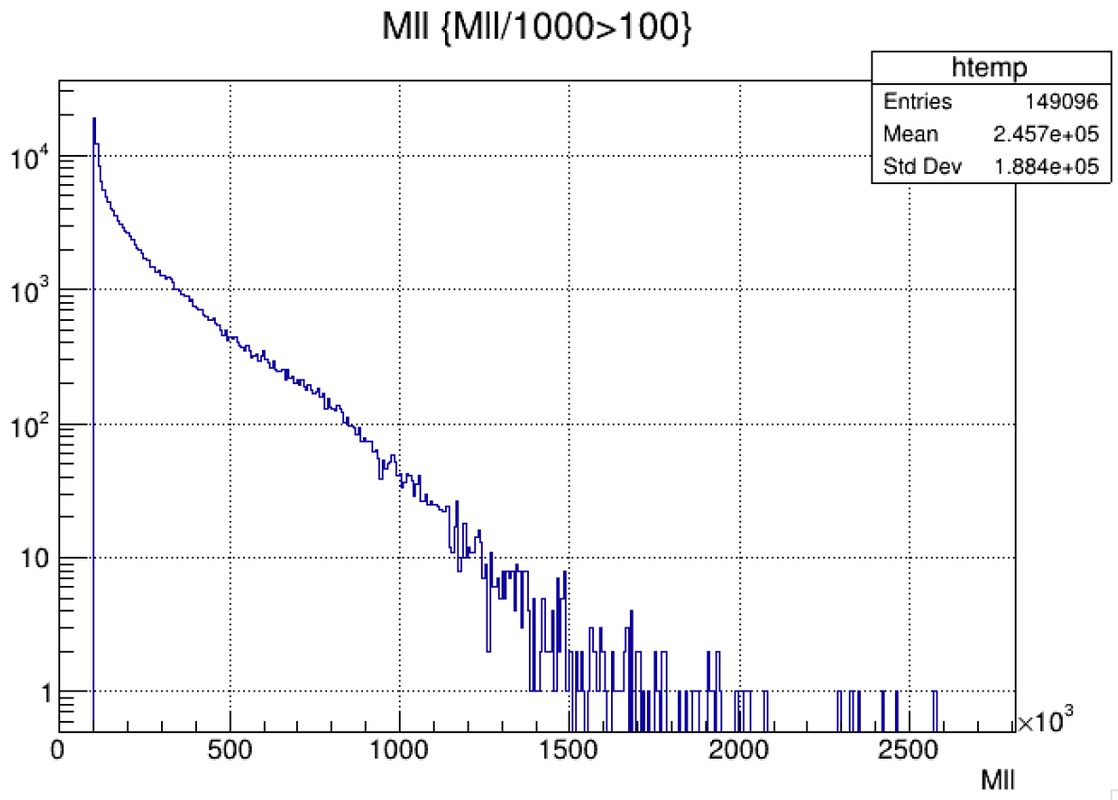

Mll {Mll/1000>60&&Mll/1000<200}
htemp
Entries 523892
Mean 9.656e+04
Std Dev 2.031e+04
Mll

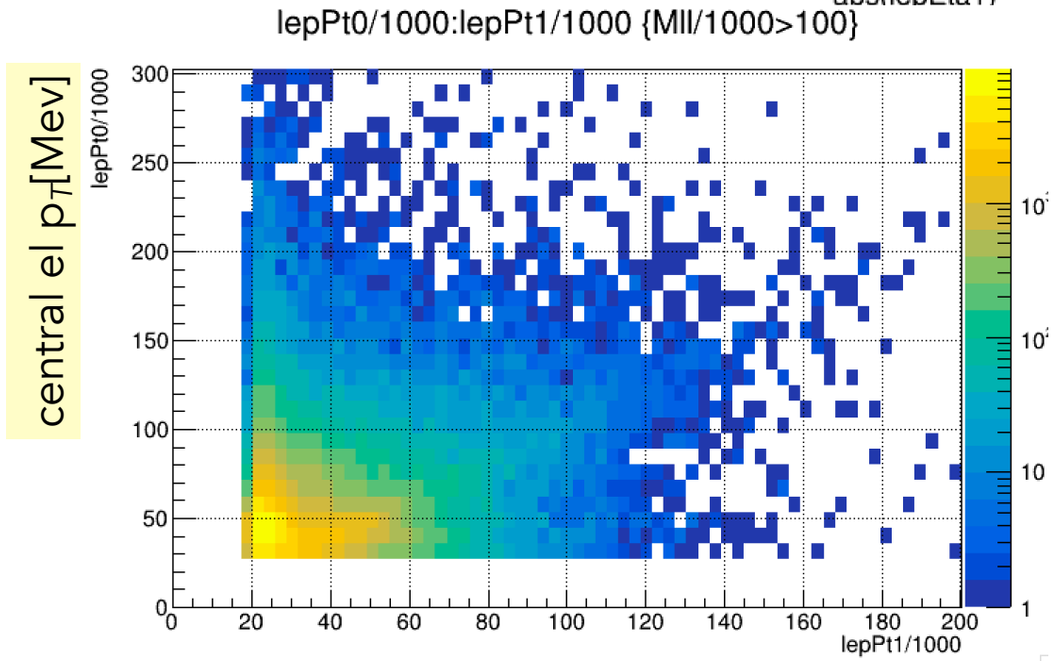

Mll {Mll/1000>60&&Mll/1000<200 && lepPt1/1000>80}
htemp
Entries 2498
Mean 1.163e+05
Std Dev 3.732e+04
high pT(fwD)
Mll

Mee(CF)>100

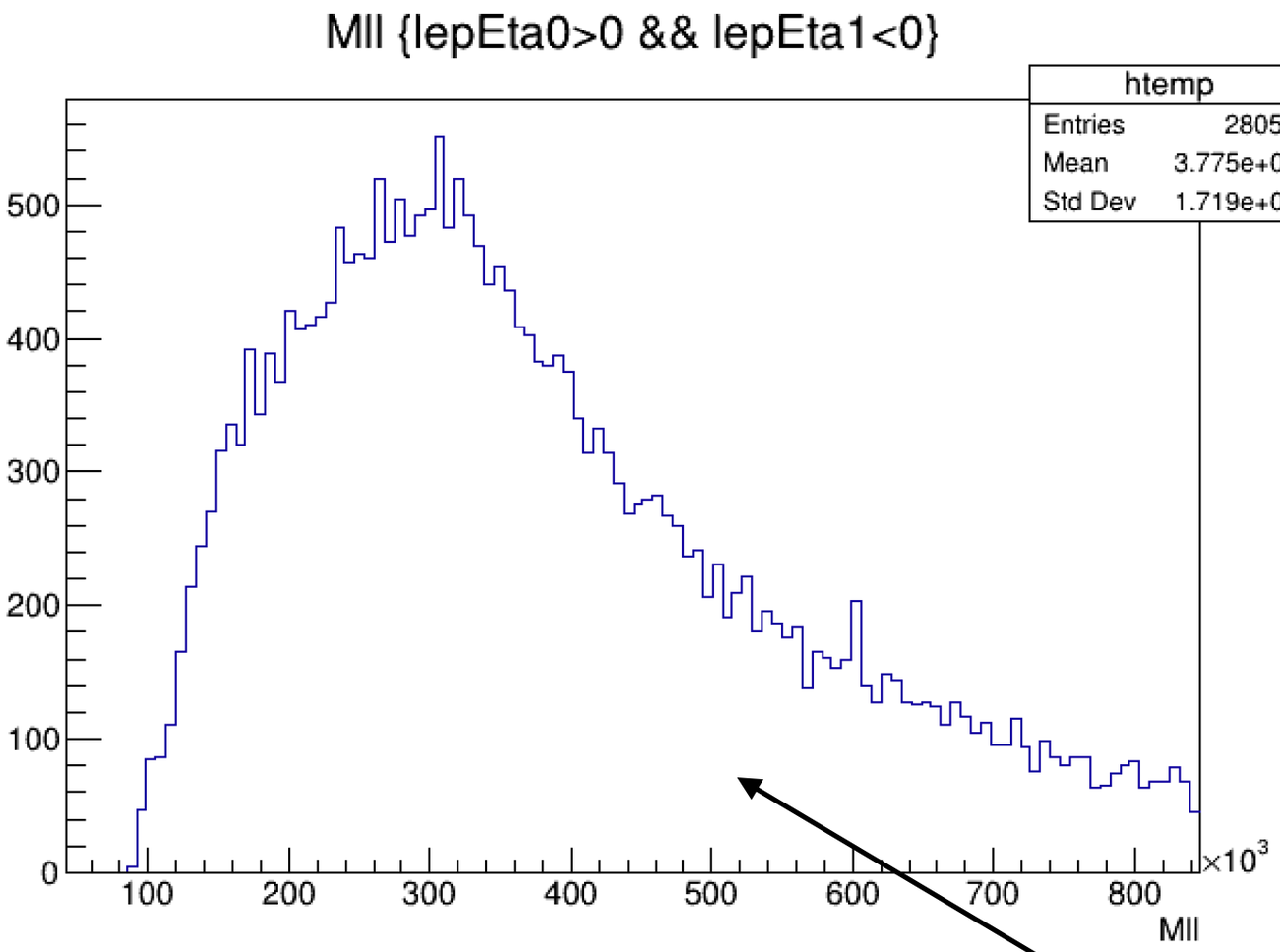

MII/1000
htemp
Entries 592108
Mean 128.6
Std Dev 116.6
MII/1000


abs(lepEta0):abs(lepEta1) {MII/1000>100}
abs(lepEta0)
abs(lepEta1)


MII {MII/1000>100}
htemp
Entries 149096
Mean 2.457e+05
Std Dev 1.884e+05
×10^3
MII


lepPt0/1000:lepPt1/1000 {MII/1000>100}
central el p_T[Mev]
lepPt0/1000
lepPt1/1000
fwd el p_T[Mev]

interesting feature



positive eta Central
negative eta FWD !