PPFX at MicroBooNE

Nitish Nayak

LArSoft Co-ordination Meeting
August 8th, 2023

Introduction

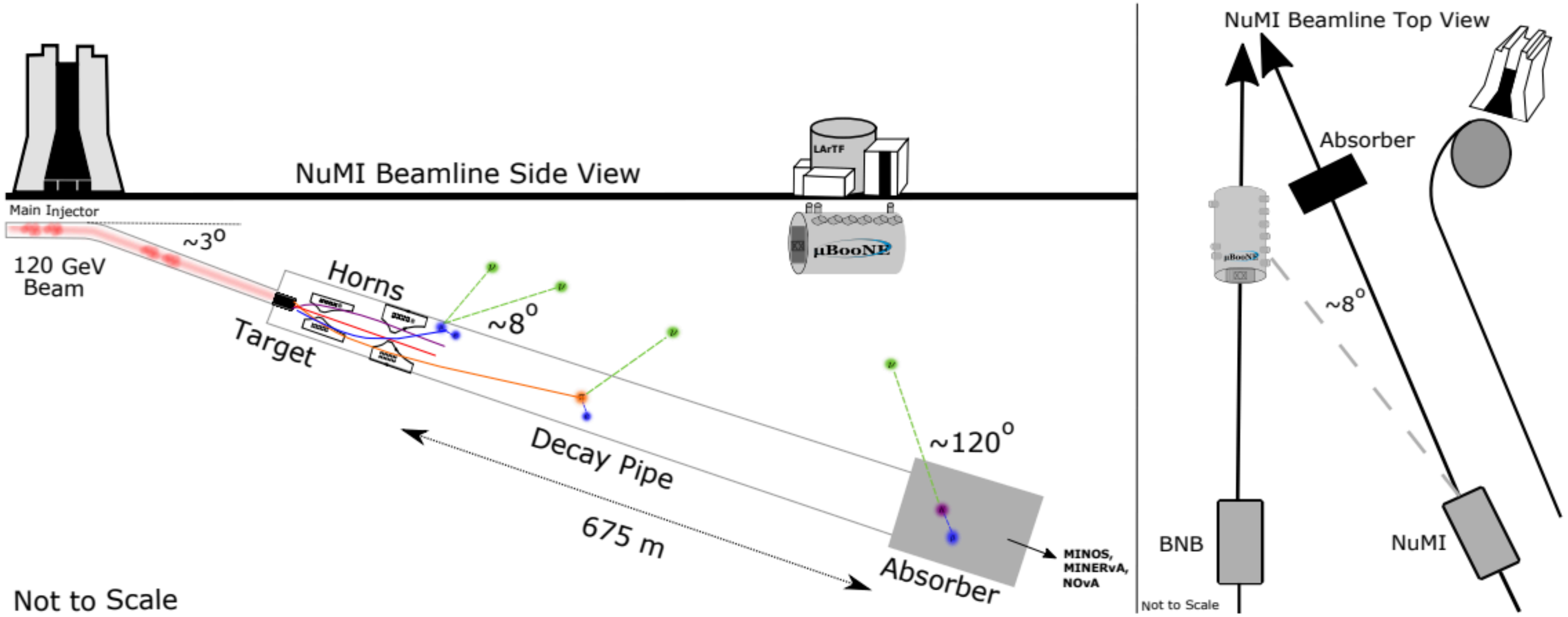
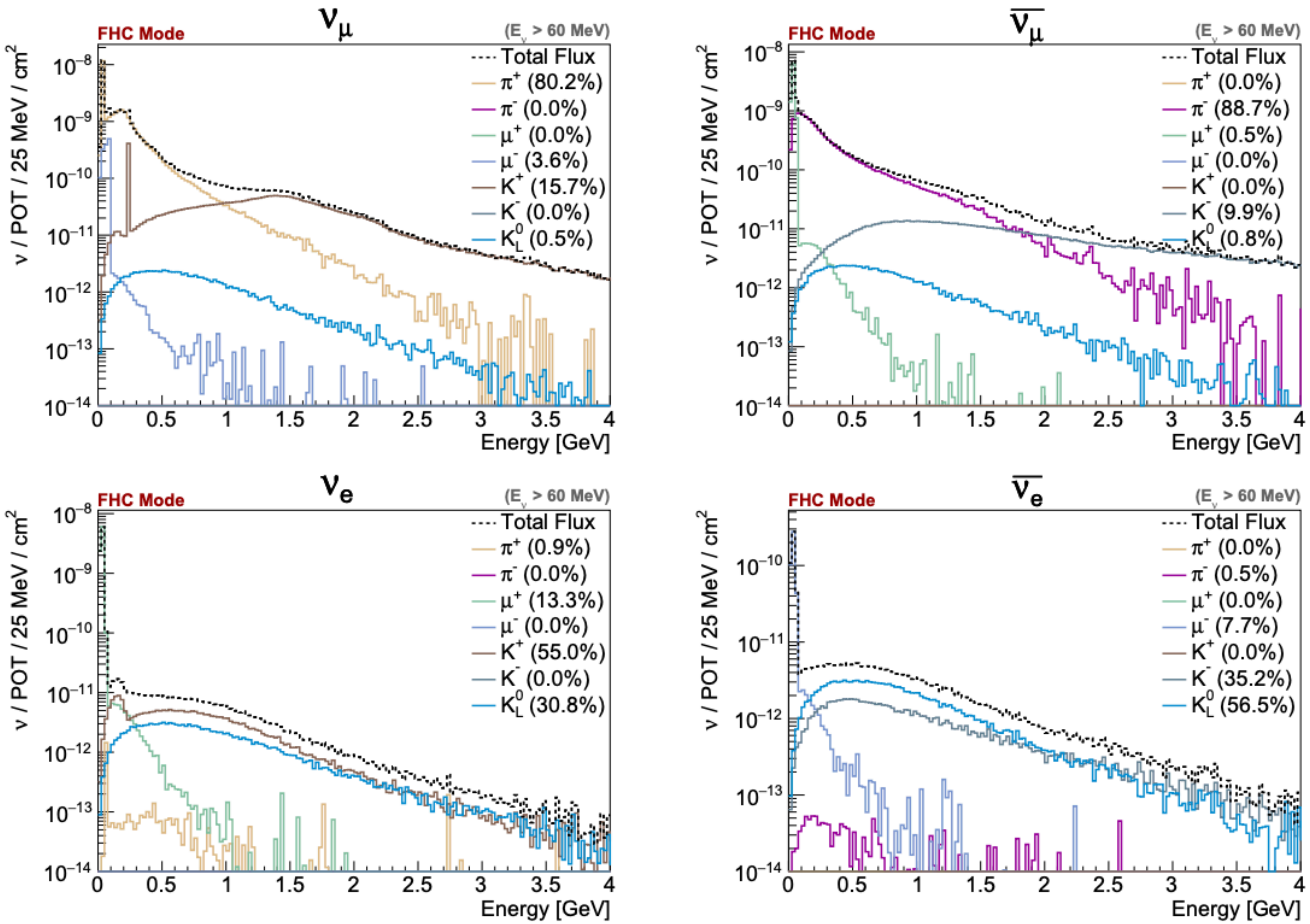


Figure 6.15: A schematic of the NuMI beamline for MicroBooNE from the elevation and top views.

- MicroBooNE sees neutrinos from two beams, BNB and NuMI
  - BNB is p-Be at 8 GeV : simulation adapted from MiniBooNE
  - NuMI is p-C at 120 GeV : use PPFX to reweight G4NuMI prediction. PPFX developed for MINERvA/NOvA but adapted for uBooNE angles as well
- We discuss current implementation, considerations for large angle beams and possible changes needed especially for us
  - Also useful for SBND, ICARUS (Some issues found in ICARUS also relevant to us, for eg. xF calculation bug)

NuMI Flux

- Flux peaks at low energies, but if we ignore peak ~0
  - 80% of ν_μs come from π-decay
  - At larger off-axis angles, a significant portion of ν_e flux from kaons including neutral (KLs)
- We study KLs predominantly in next slides but some of it relevant to all hadrons produced



Data Coverage

- Our ν_μ flux (coming mostly from π-decay) is pretty well covered by NA49
- Kaons are a slightly different story
  - Current dataset only covers xF > 0
  - Especially for KLs (as we'll see), significant fraction of events have xF < 0

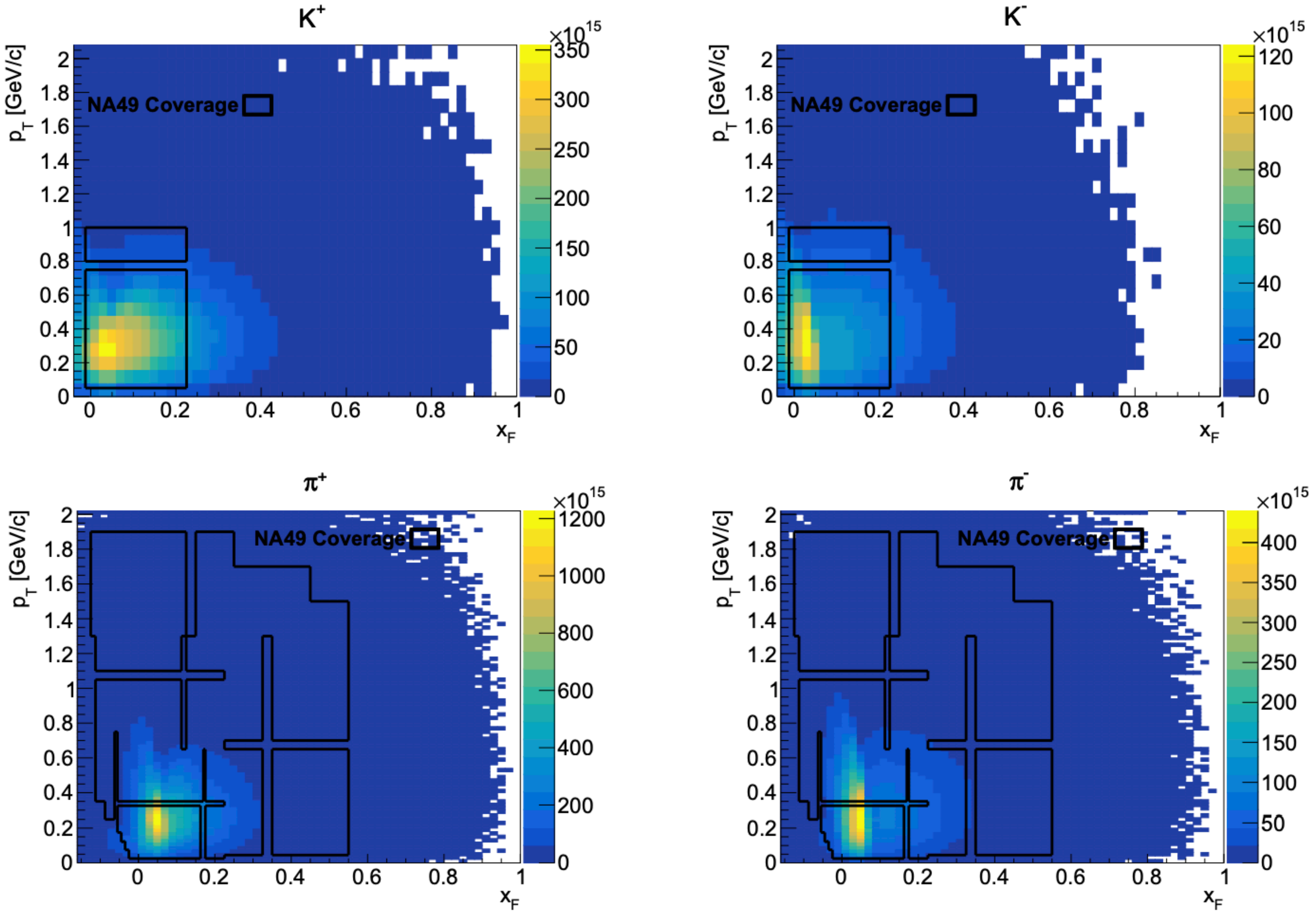
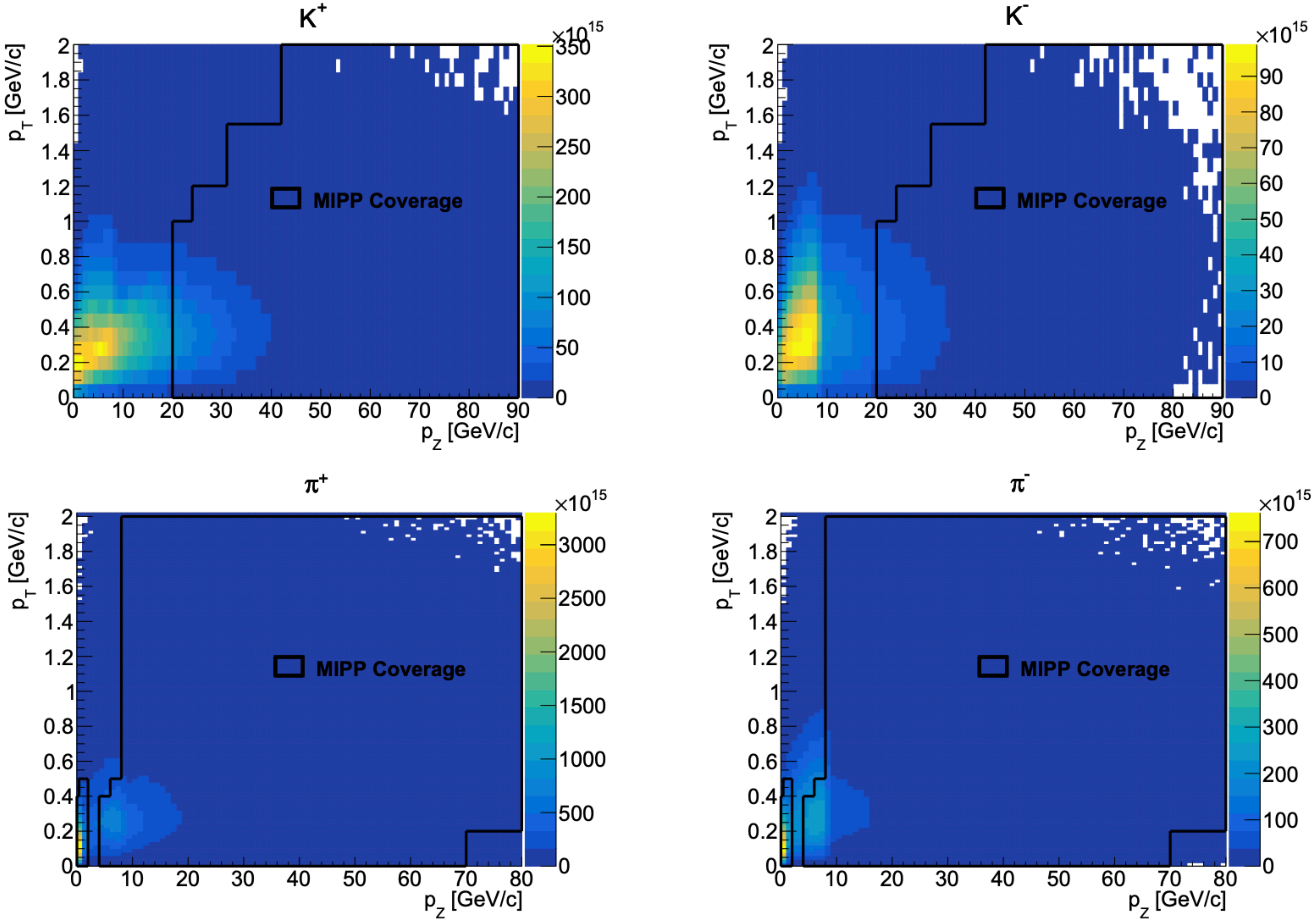


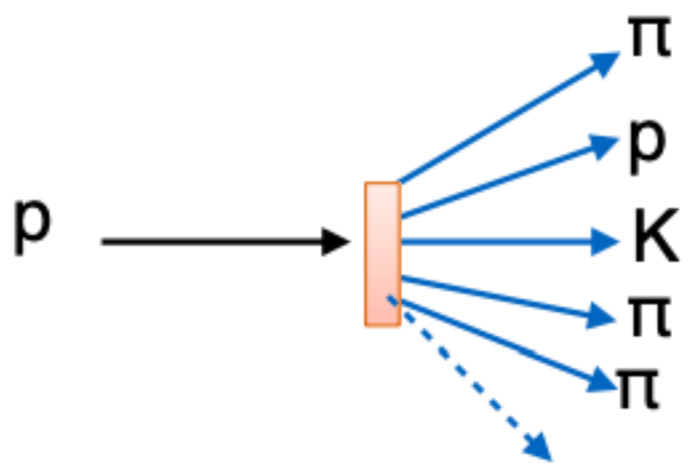
Figure 4.1: Diagrams of the kinematic phase space of kaons and pions for the MicroBooNE flux with the NA49 data coverage overlaid with the black box.

Data Coverage

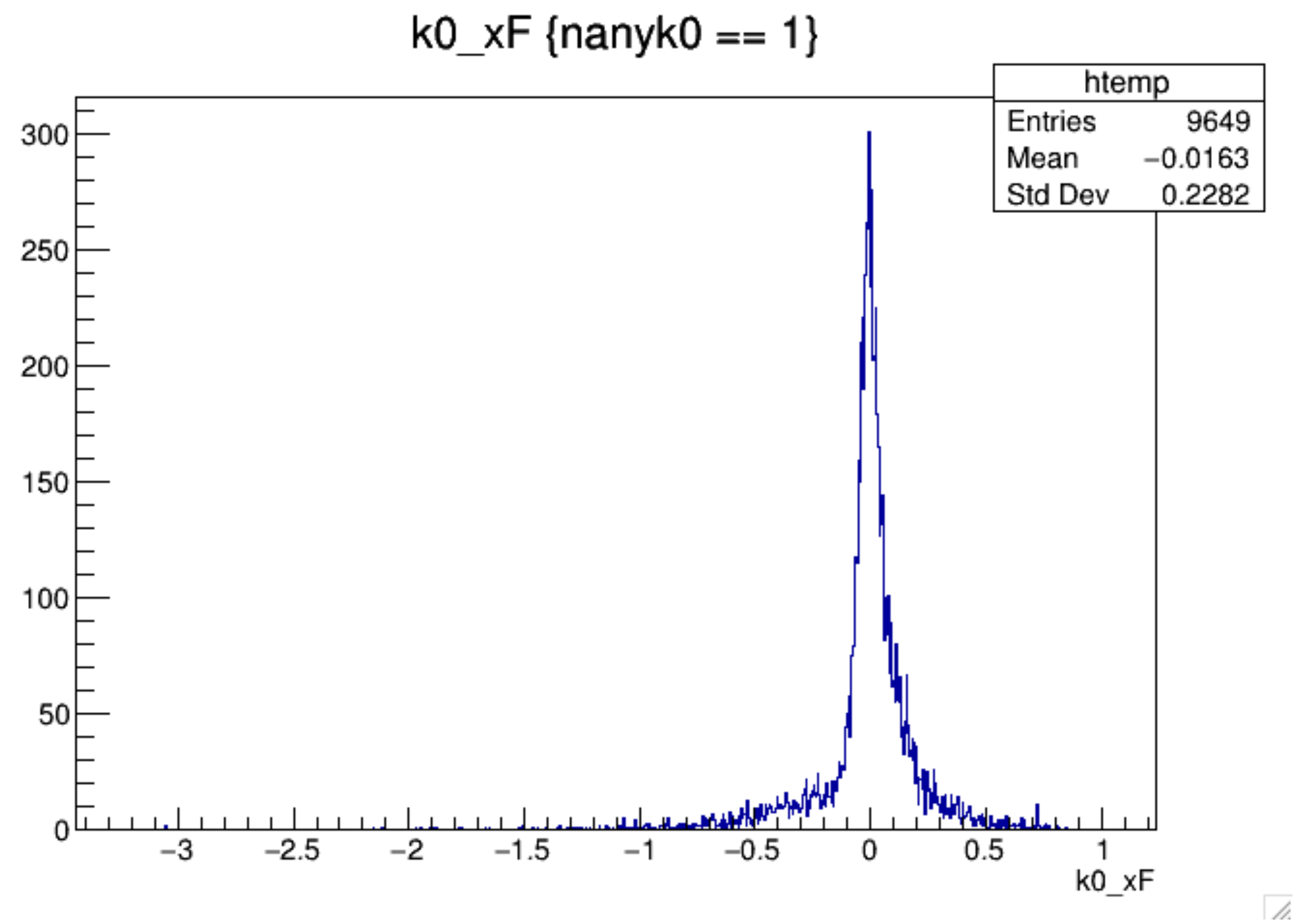


- MIPP coverage also not very good for kaons at our angles
  - We use PPFX with MIPPNuMIOff settings => ~only thin-target data used

KL xF distribution

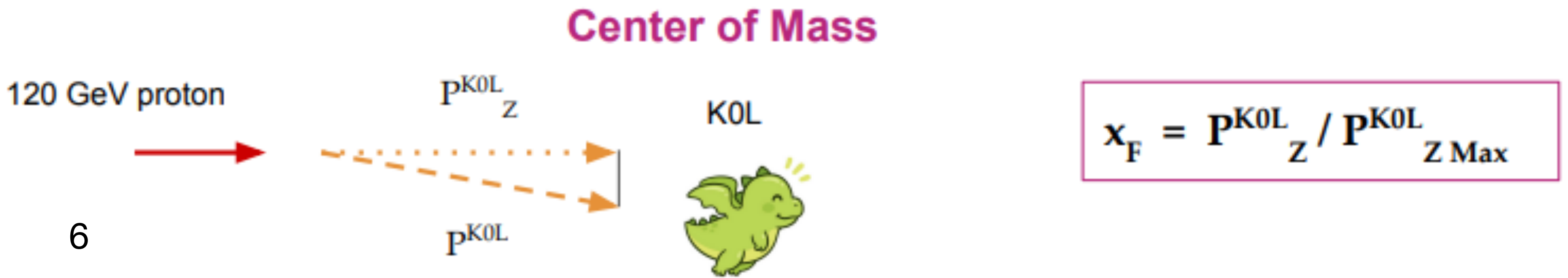


- PPFX assigns weight + uncertainties to each step in the cascade process that produced neutrinos
- If we take KLs as an example :
  - Either decays into neutrinos directly or interacts further producing other particles that decay into our neutrinos
  - Saving information about most downstream KLs (~95% of which are direct parents to ν_e sample)

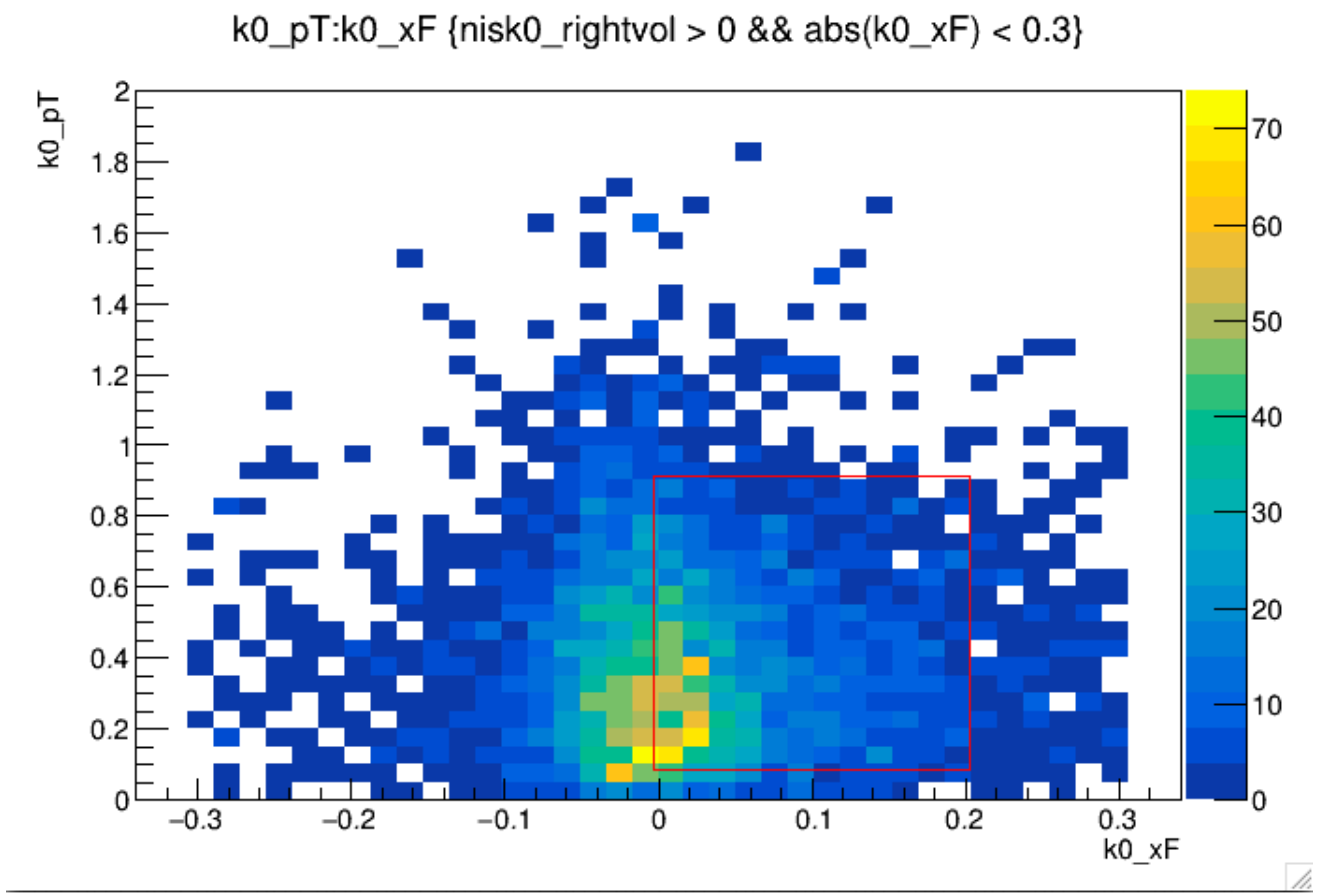


- ~50% of events with 1 KL in the decay chain have KL xF < 0
  - For large off-axis expts this makes sense, i.e we see neutrinos from this region in phase space. MINERvA, NOvA probably don't
- Some are even < -1 (bug in PPFX calculation -> will get into it later)
- xF directly as calculated and reported by PPFX for the particular hadron species (I don't calculate it myself) : $xF = 2P_L/\sqrt{s}$ (P_L in COM frame)

Graphic from Elena



NA49 phase space



- Only ~1/3rd of primary KLs are covered by external data for us! (=> small data-related uncertainties. Most events get mainly GEANT-level uncertainties, i.e from FTFP-BERT model)
- A large fraction of primary KLs have negative xF and fall outside NA49 box
- Uncertainties within NA49 box determined by statistical ensembles made from NA49 data
  - Outside box => different treatment based on interaction characteristics of that hadron species (parent, target nucleus, secondary KLs etc)

xF < 0 Issue

$PPFX_DIR/src/ThinTargetnucleonAReweighter.cpp

Nucleon-A

xF < 0

```
bool ThinTargetnucleonAReweighter::canReweight(const InteractionData& aa){

  //checking:
  if(aa.Inc_pdg != 2212 && aa.Inc_pdg != 2112)return false;
```

$PPFX_DIR/data/BINS/ThinTarget_MesonIncident.xml

4 equally spaced bins of xF b/w (0, 1)

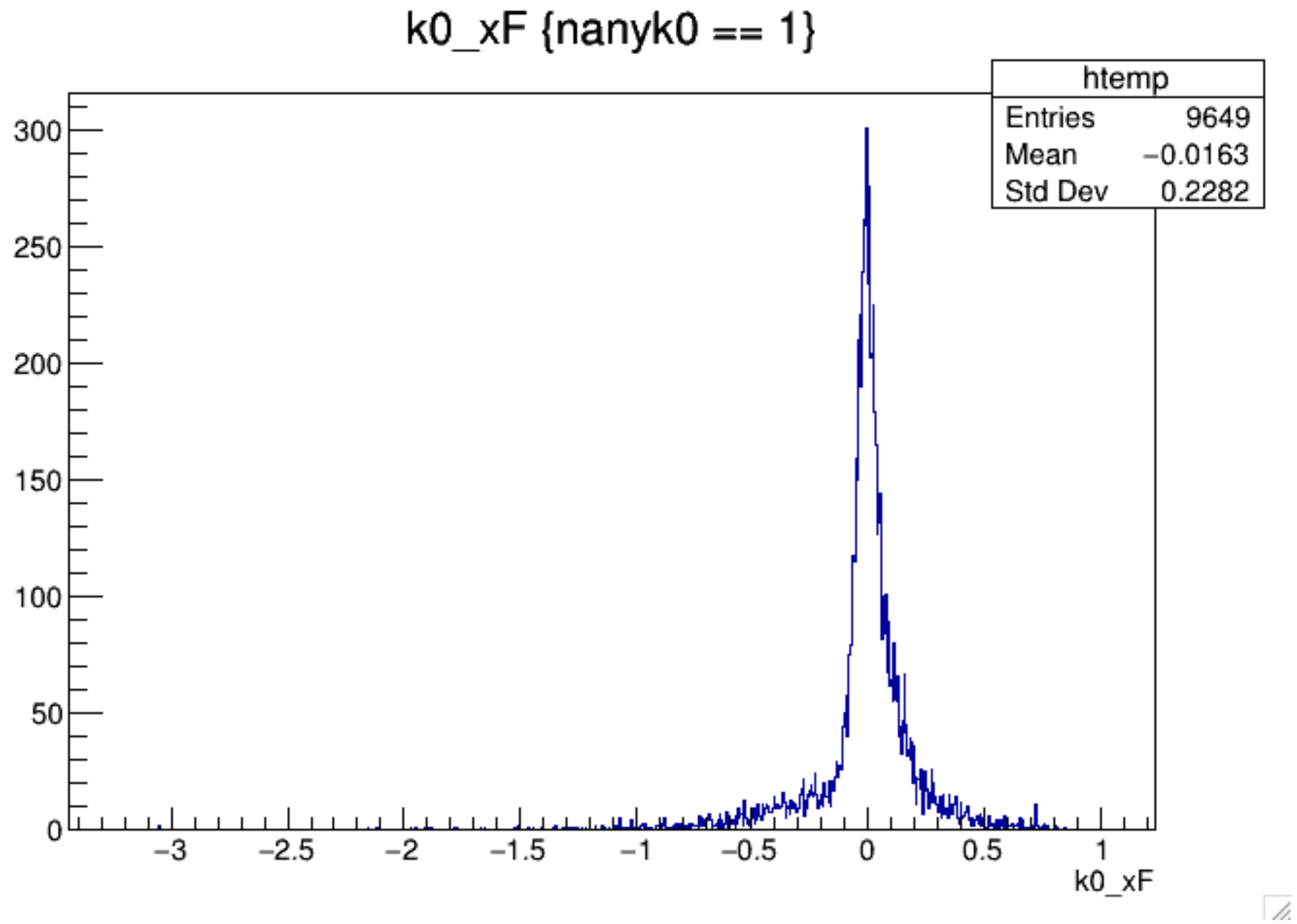
```
    //trick... using a function for meson incident... same binning.
    int binnu       = Thinbins->meson_inc_BinID(aa.xF,aa.Pt,211);
    if(binnu<0)return 1.0;
```

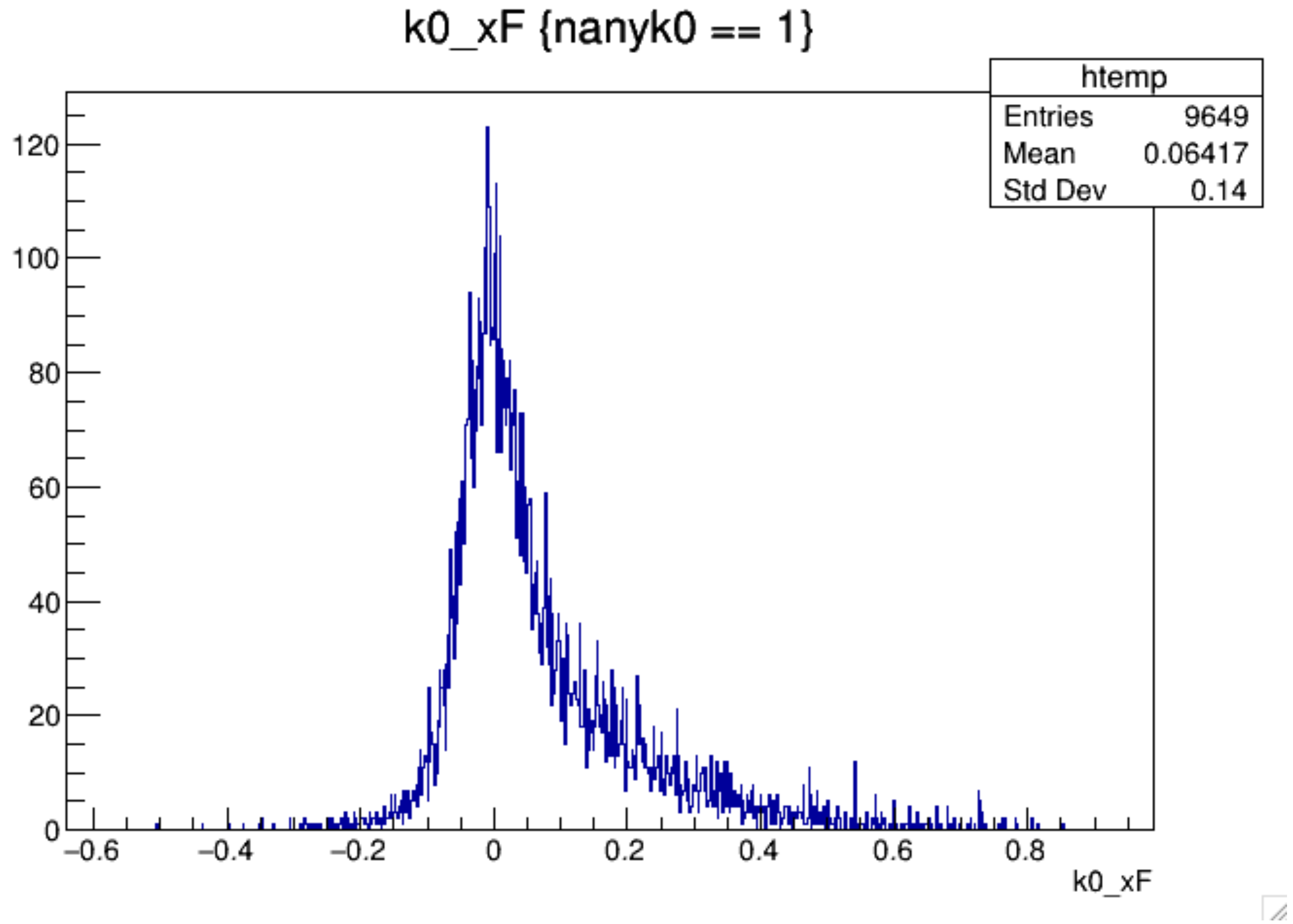
return 1

- Main issue is in nucleon-A (parent of hadron is nucleon) when it falls out of data coverage
  - Happens for xF < 0
  - PPFX doesn't expect this to happen significantly
  - Mistakenly applies no uncertainties (returns a weight of 1.)
  - Went unnoticed for small off-axis angle expts but we get a significant fraction of our events with xF < 0!
- Question about how to assign uncertainties to such events
  - Generally, PPFX assigns GEANT level uncertainties on a "best guess" based on how FTFP-BERT performs on external datasets
  - Level of disagreement generally ~40%
    - Seems like not much info on whether its uncorrelated/correlated across xF bins/hadrons produced, what regions of kinematic phase space its relevant etc
- We evaluated different options and converged to an uncorrelated + correlated choice after discussions with PPFX experts

xF Calculation Issue

Link to ICARUS PR : https://github.com/kordosky/ppfx/pull/1





- For xF calculation, PPFX assumes incident hadron particle has same mass as target nucleon. For meson incident processes especially, this is a problem
  - Pushed more towards negative values, even to unphysical
  - Fixing the calculation as in that presentation pushes it to more positive values (unphysical values gone as well)
- Haven't done extensive checks but ICARUS slides indicate significant impact (lower uncertainties in general but lower correlations between FHC/RHC and different flux flavors)

Found in ICARUS context

```
+    /**
+     * Center of mass of the system of projectile
+     * and nucleon (not the nucleus!)
+     */
+    static const double NUCLEON_MASS =
+      (particle->GetParticle("proton")->Mass()
+       + particle->GetParticle("neutron")->Mass())/2;
+    static const double NUCLEON_MASS2 = NUCLEON_MASS * NUCLEON_MASS;
+
     double inc_E_lab = std::sqrt(Inc_P*Inc_P + pow(Inc_Mass,2));
-    InteractionData::Ecm       = std::sqrt(2.*pow(Inc_Mass,2)+2.*inc_E_lab*Inc_Mass);
-    InteractionData::Betacm    = std::sqrt(pow(inc_E_lab,2)-pow(Inc_Mass,2.0))/(inc_E_lab + Inc_Mass);
+    InteractionData::Ecm       = std::sqrt(pow(Inc_Mass,2) + NUCLEON_MASS2 + 2.*inc_E_lab*NUCLEON_MASS);
+    InteractionData::Betacm    = std::sqrt(pow(inc_E_lab,2)-pow(Inc_Mass,2.0))/(inc_E_lab + NUCLEON_MASS);
     InteractionData::Gammacm   = 1./std::sqrt(1.-pow(Betacm,2.0));
```

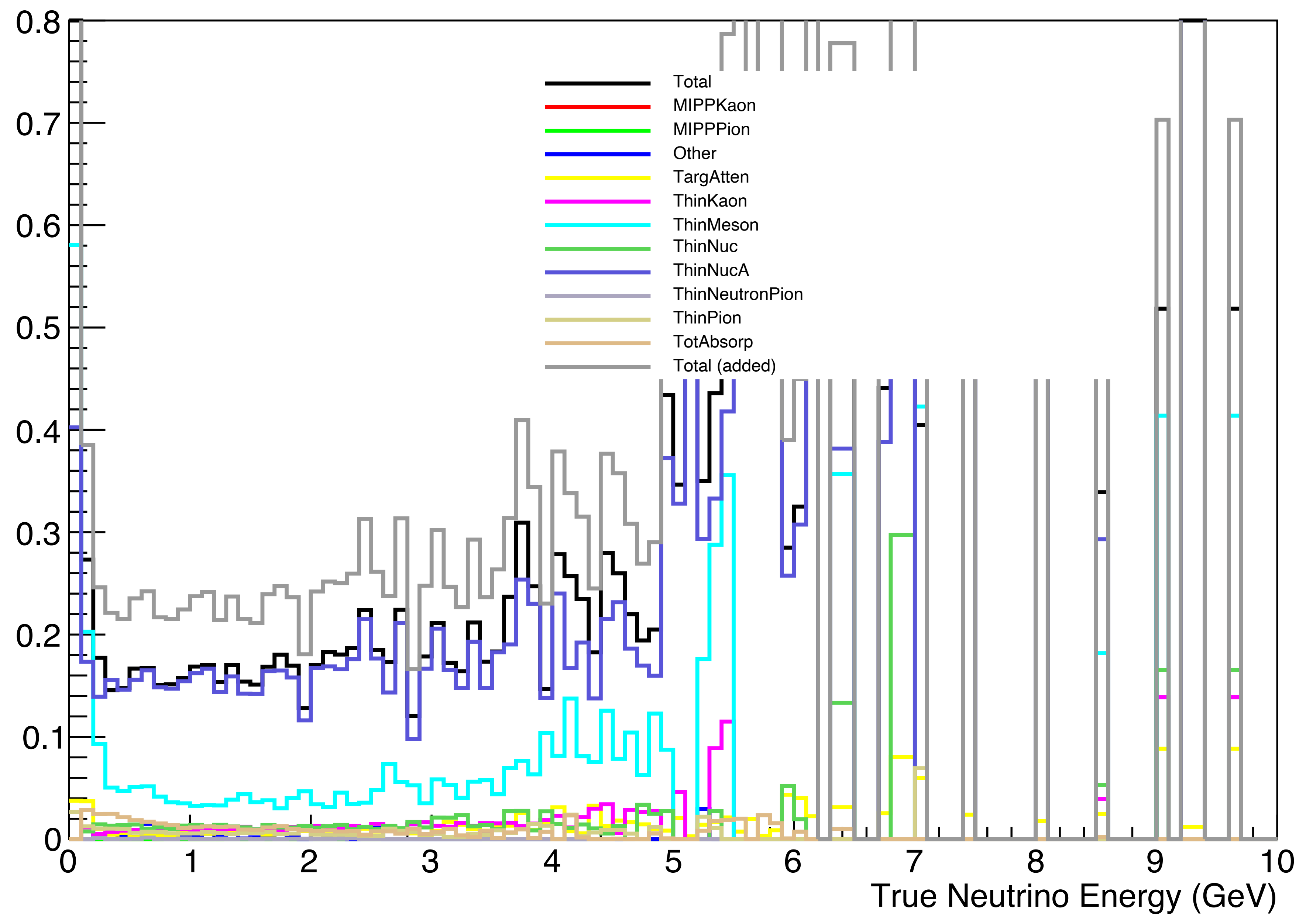
Why uncorr. + corr. ?

- GEANT model — total black box (based on discussions with Laura F. and Leo A.)
- PPFX treatment based on best guess without being too conservative for original purpose :
  - Uncorrelated across hadron species and different xF bins (within xF bin — fully correlated)
  - Mainly developed for ~on-axis NuMI (good HP coverage by external data, geant-level uncertainties not as dominant)
  - MicroBooNE might have to be a bit more conservative (Include correlations across hadron species for eg.)
- Planning on using 40% uncorr. (8 xF bins in total) + 40% corr. (for xF < 0)
  - Because of a tight analysis schedule, we want to do something conservative and then maybe revisit
- We expect :
  - Increased ν_e flux uncertaintes
  - Not too much impact on ν_μ flux (mainly from π covered by data)
  - Reduced correlations between ν_e and ν_μ (xF bug fix, different decay channels etc)

40% fully corr. + 40% uncorr. (8 xF bins)

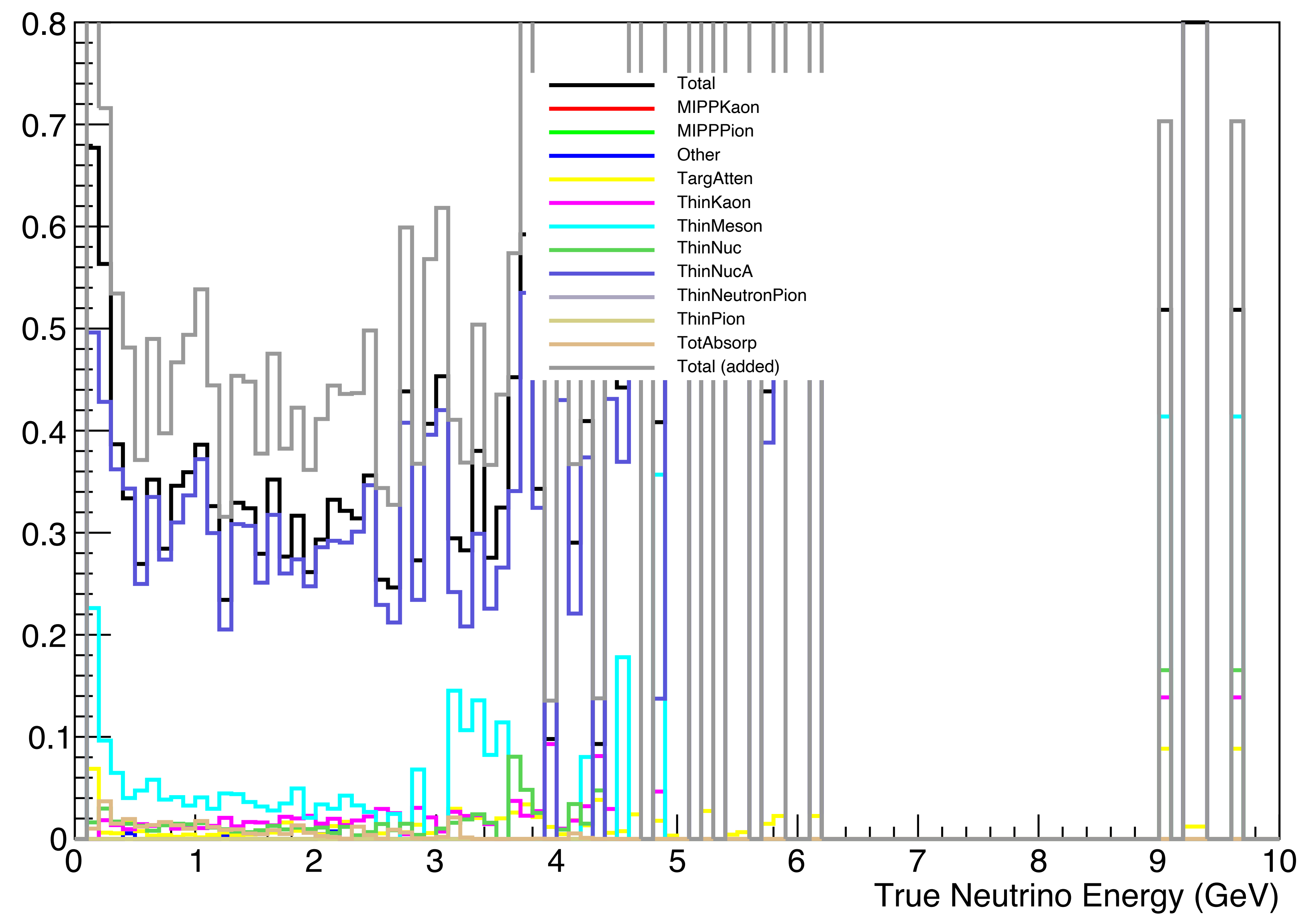
All ν_e - FHC

Total Unc. (%) : 17.16



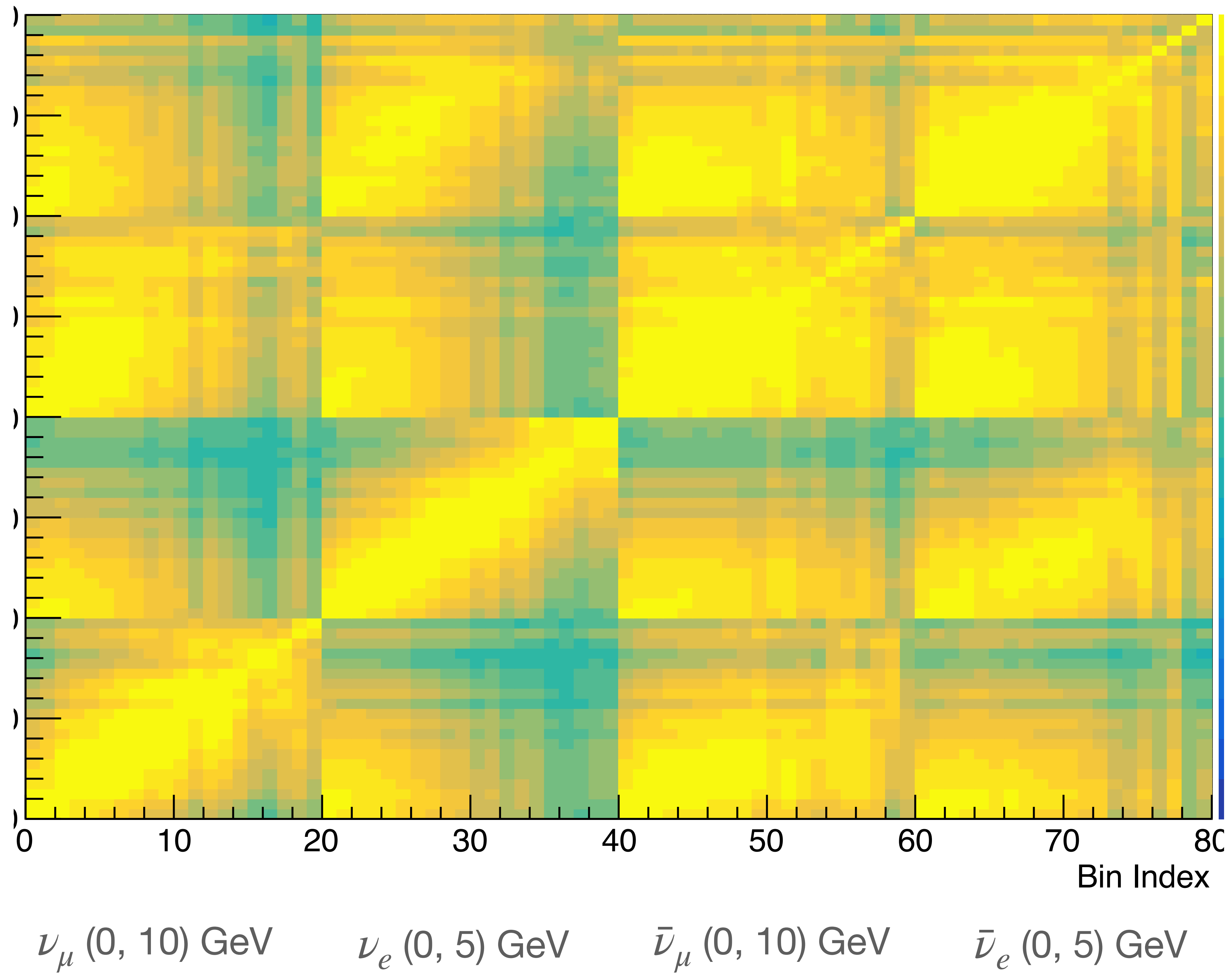
ν_e - FHC (Primary KLs)

Total Unc. (%) : 33.20

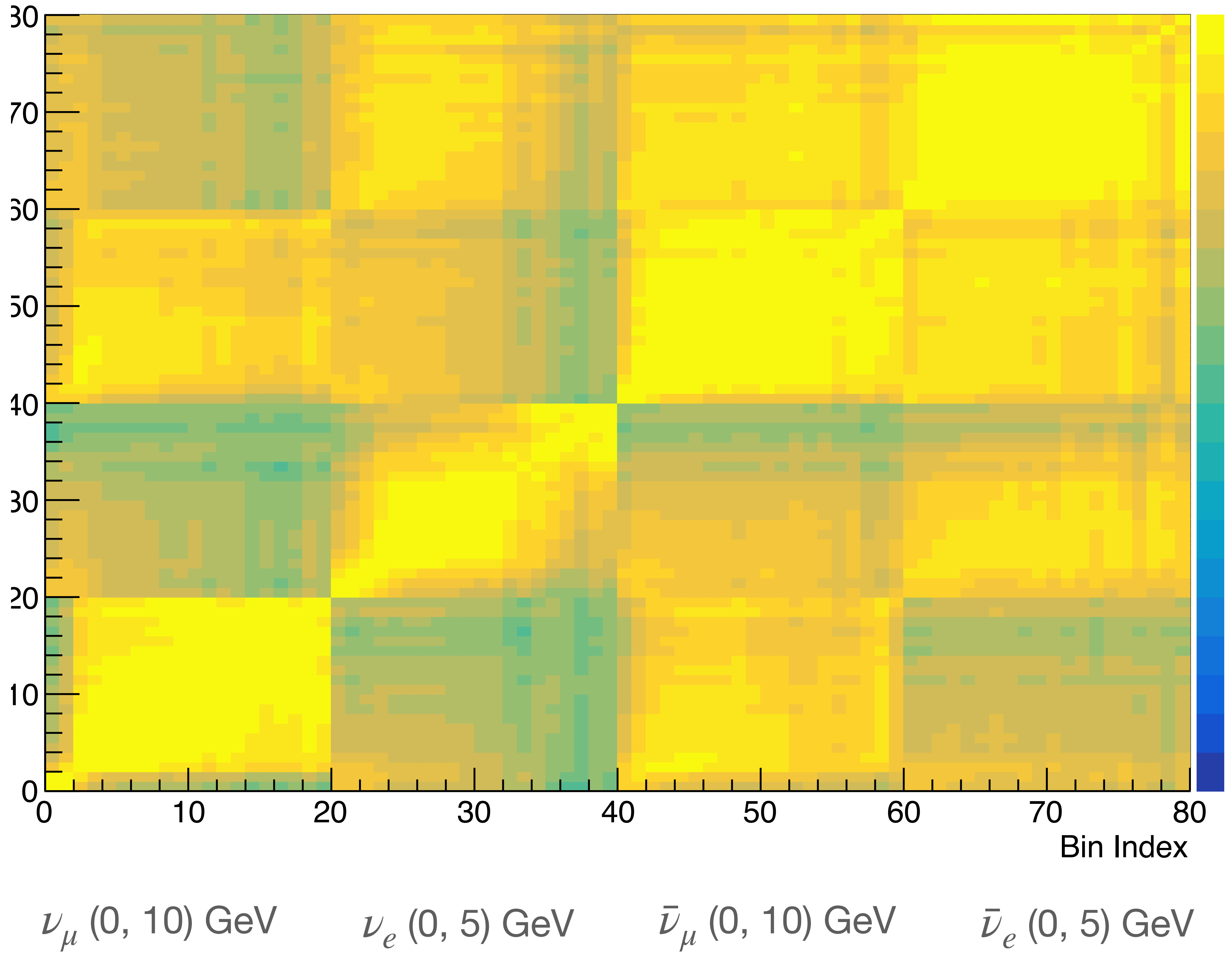


- xF Bug fixed
- Overall uncertainties ~17%
- Large uncertainty on primary KLs

Old - out of the box PPFX



40% + 40% + xF bug fix



- Flux-only correlation matrix across 4 flavors in FHC
- Do see reduced correlations across flavors.
- However within ν_e flavors we see a slight increase in correlations -> related to 40% correlated on nucleon-A HP for xF < 0

Propagating Changes

- PPFX development is a bit splintered
  - Redmine repo - read-only access : https://cdcvs.fnal.gov/redmine/projects/ppfx
  - Also moved to private repo : https://github.com/kordosky/ppfx
- MicroBooNE branches off of an old larsoft version (v8)
  - Uses ppfx in branch lar_v2_11_br
- Larsoft integration release is v9_XX already
  - Uses ppfx in branch lar_v2_18_br
- Possibly some development on private GitHub repo — not sure about updates here

Propagating Changes

- Created my fork here : https://github.com/nitish-nayak/ppfx with appropriate feature branches for merging everywhere
  - feature/bnayak_ppfxpatch -> v2_11_br
  - feature/bnayak_ppfxpatch_integration -> v2_18_br
    - These have both the fixes (neg xF uncertainties + xF bug fix)
  - feature/bnayak_negxfuboone -> kordosky:main (PR here)
    - This has only the neg xF uncertainty fix since there's already an existing PR with the xF bug fix

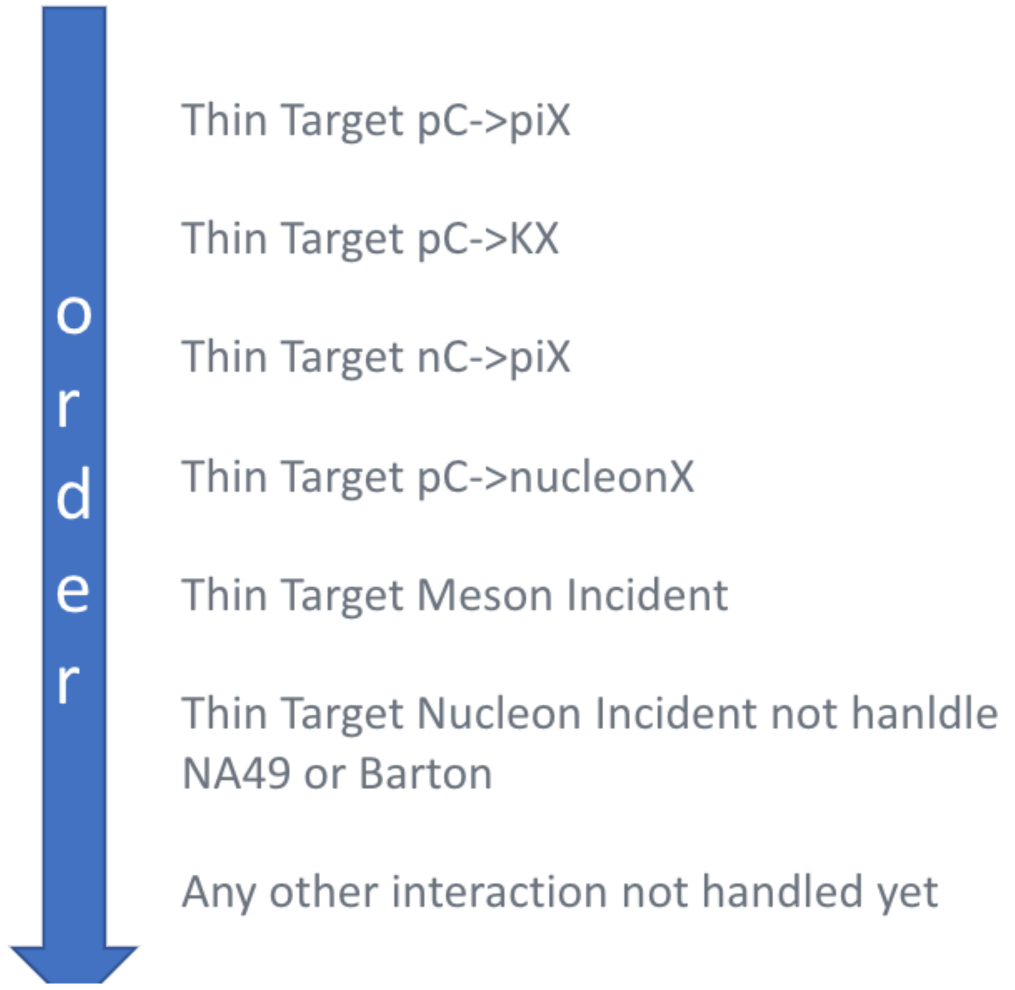
Summary

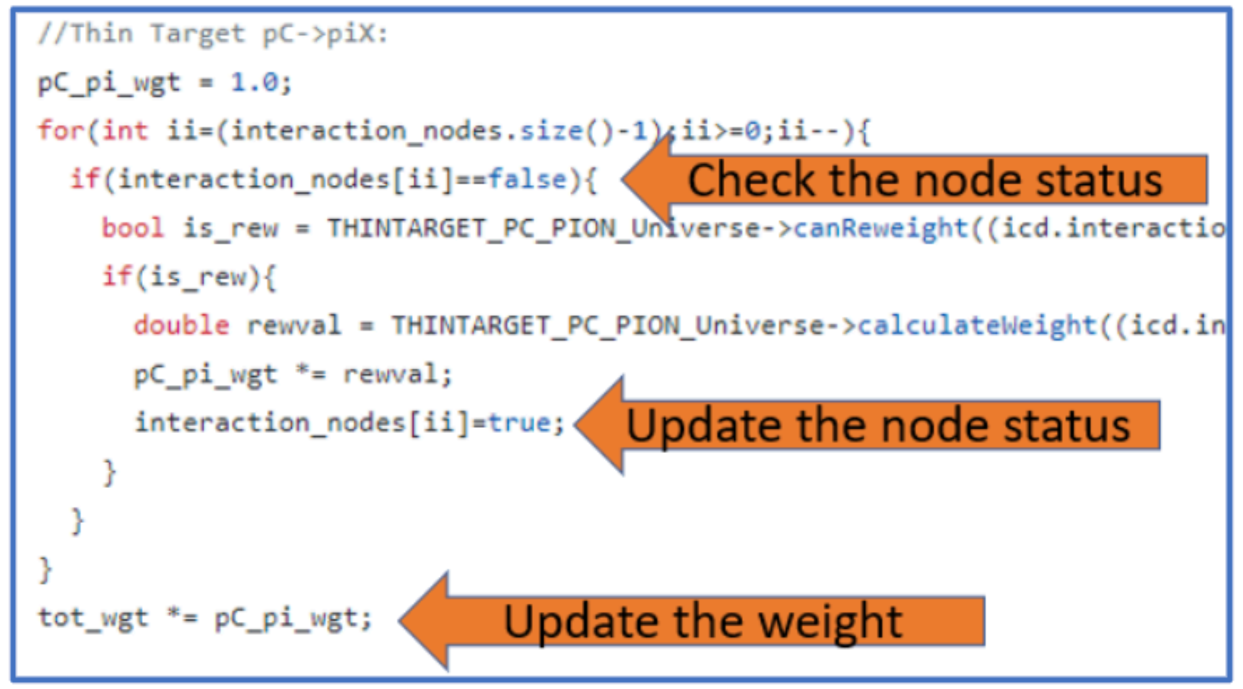
- Able to merge ppfx bug fixes/updates for MicroBooNE and all larsoft/numi experiments
- Also have PR to the private GitHub repo in case it becomes relevant
- Could have significant physics impact for NuMI off-axis (SBND, ICARUS for eg)
  - Physics evaluation and some judgement is needed but these are clearly bugs that need to be addressed somehow by experiments
  - Discussed with PPFX experts and they agree
  - Should have minimal/negligible impact for NuMI ~on-axis (MINERvA, NOvA, DUNE)
- Any other things to consider?

Backup

Other KL Uncertainties

Graphic from Xiangpan





```
//Thin Target pC->piX:
pC_pi_wgt = 1.0;
for(int ii=(interaction_nodes.size()-1);ii>=0;ii--){
  if(interaction_nodes[ii]==false){
    bool is_rew = THINTARGET_PC_PION_Universe->canReweight((icd.interactio
    if(is_rew){
      double rewval = THINTARGET_PC_PION_Universe->calculateWeight((icd.in
      pC_pi_wgt *= rewval;
      interaction_nodes[ii]=true;
    }
  }
}
tot_wgt *= pC_pi_wgt;
```


Fraction of events with 1 KL : ~40%

- Fraction of events with 1 KL in pC->KX category : 7.48% ← NA49 measurement uncertainties
- Fraction of events with 1 KL in MesonInc category : 11.11%
  - Fraction of events with 1 KL in MesonInc w/ xF < 0 : 6.96% ← Flat 40%, remaining uncorrelated 40% in 4 xF bins b/w (0,1)
- Fraction of events with 1 KL in NucleonInc category : 21.38%
  - Fraction of events with 1 KL in Nucleon Inc in NA49 : 2.82% ← NA49 measurement uncertainties
  - Fraction of events with 1 KL in NucleonInc w/ xF < 0 : 13.58% ← 0% uncertainties, remaining uncorrelated 40% in 4 xF bins b/w (0, 1)
- Fraction of evts with 1 KL in Other category : 0.39% ← Flat 20%

40% uncorr.

- 40% corr change :
  - It's not just changing KLs but each hadron produced (K+/-, pi+/-, KL, n, p) with negative xF
  - Also changing them in a fully correlated way,
    - i.e a single knob that assigns 40% uncertainty to each such event, no matter which hadron is produced or whether the parent was neutron or proton
- PPFX suggests that they could have some shape as well (needn't be correlated) which is why they divided it into 4 xF bins separated by parent, hadron species in the first place
  - For mesonInc => 4 x 5 x 7 uncorrelated knobs w/ each 40%
  - For nucleonInc => 4 x 2 x 7 uncorrelated knobs w/ each 40%

4 equally spaced bins of xF b/w (0, 1)

5 meson parents for mesonInc : (K+/-, pi+/-, K0)

2 nucleon parents for nucleonInc : (p, n)

7 produced species : (K+/-, pi+/-, K0, p, n)

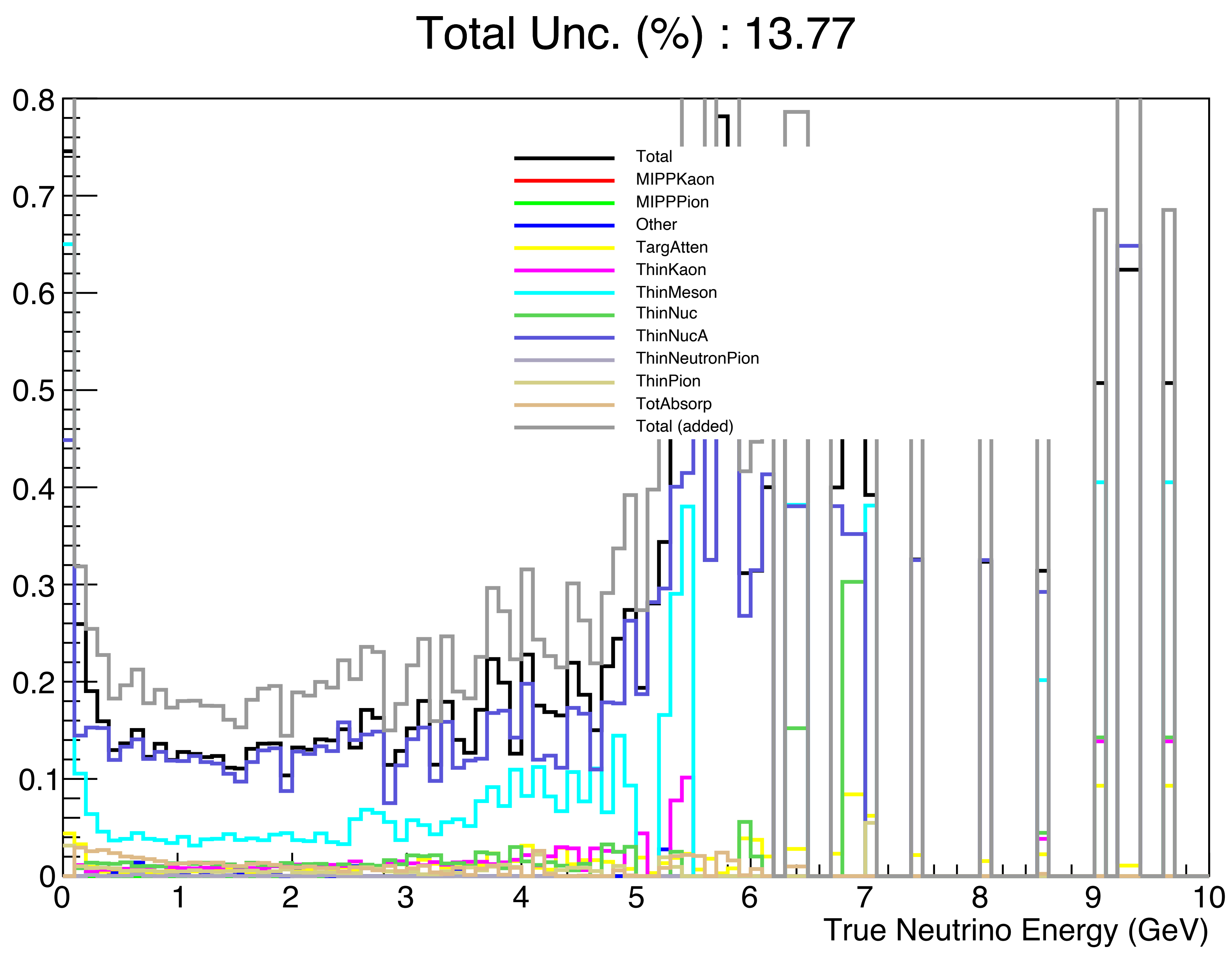
https://arxiv.org/pdf/1607.00704.pdf

Mesons traversing beamline elements often interact to produce particles that eventually lead to a neutrino. Unfortunately there is little applicable data for the 10-40 GeV mesons of interest here. We estimate the uncertainty by noting that Geant4-FTFP is a microphysical, first principles model of hadronic interactions. Our *ansatz* is that the level of agreement between FTFP and existing hadron production datasets is indicative of FTFP's ability to model interactions for which no data is currently available. Meson and nucleon production measurements exist for pC and, more generally, for pA interactions. That data agrees with the simulation at better than 40% across a broad range of relevant kinematics. We assume that this verifies the FTFP model at the 40% level. In addition, we note that the observed data-simulation discrepancies for production of $\pi^{\pm}, K^{\pm}, n$ and p do not appear to be correlated in any obvious way. Therefore, to handle meson incident interactions we categorize the interactions based on incident particle $(\pi^{\pm}, K^{\pm})$ and produced particle $(\pi^{\pm}, K^{\pm}, n, p)$. For each combination we break the range $0 < x_F < 1$ into 4 equally sized bins. In each bin we assign a 40% uncertainty and we treat each bin as being uncorrelated with the others.

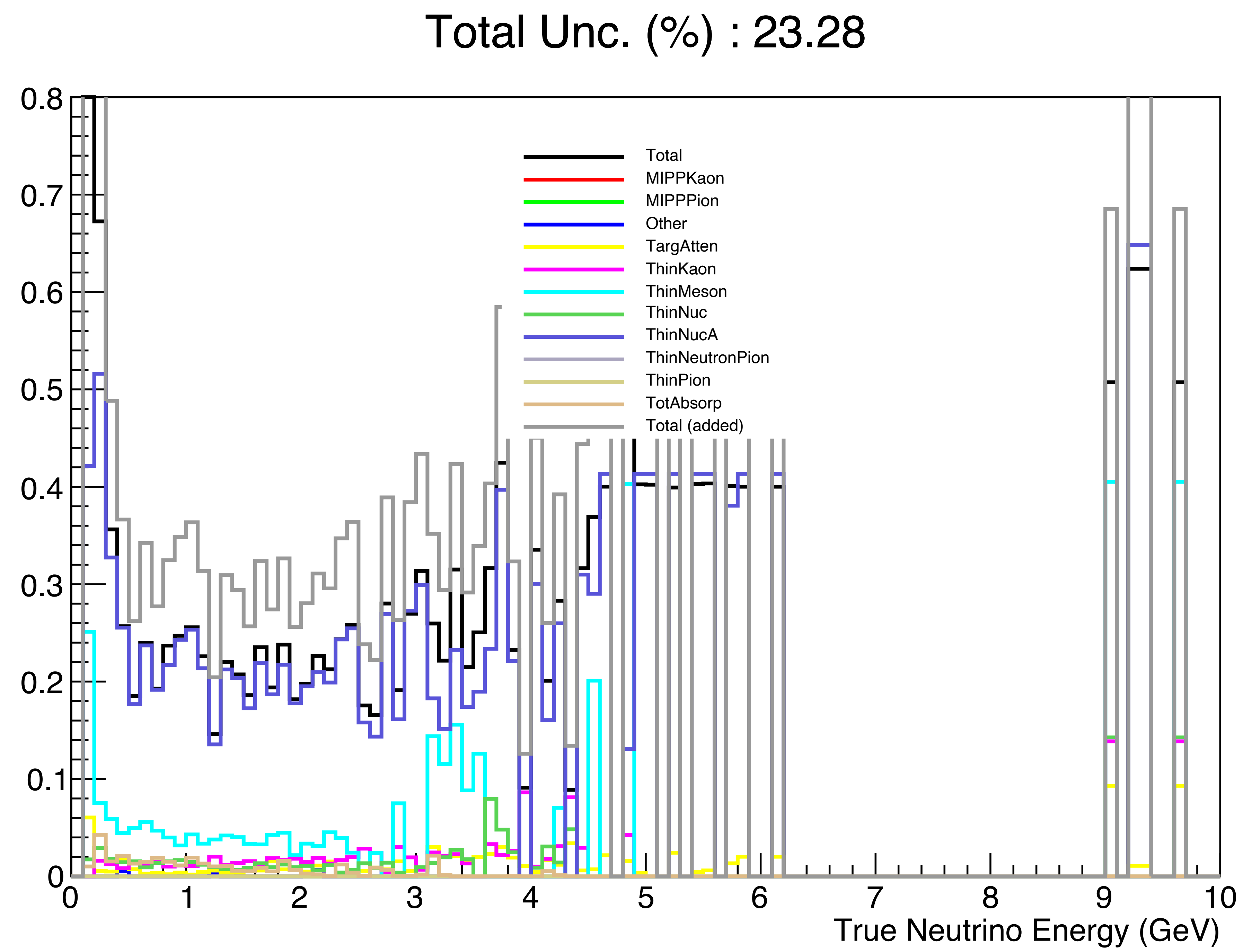
- Having so many knobs varied simultaneously and in an uncorrelated way => final impact is reduced but could influence the flux shape/correlations
- Maybe we should be doing this too w/ such xF < 0 events?
- Instead of 4 xF bins b/w (0, 1), change to => 5 xF bins. Extra bin is xF in (-1, 0) : Also do this for mesonIncident stage (no longer flat fully correlated 40% unc for xF < 0)

40% uncorr.

All ν_e - FHC

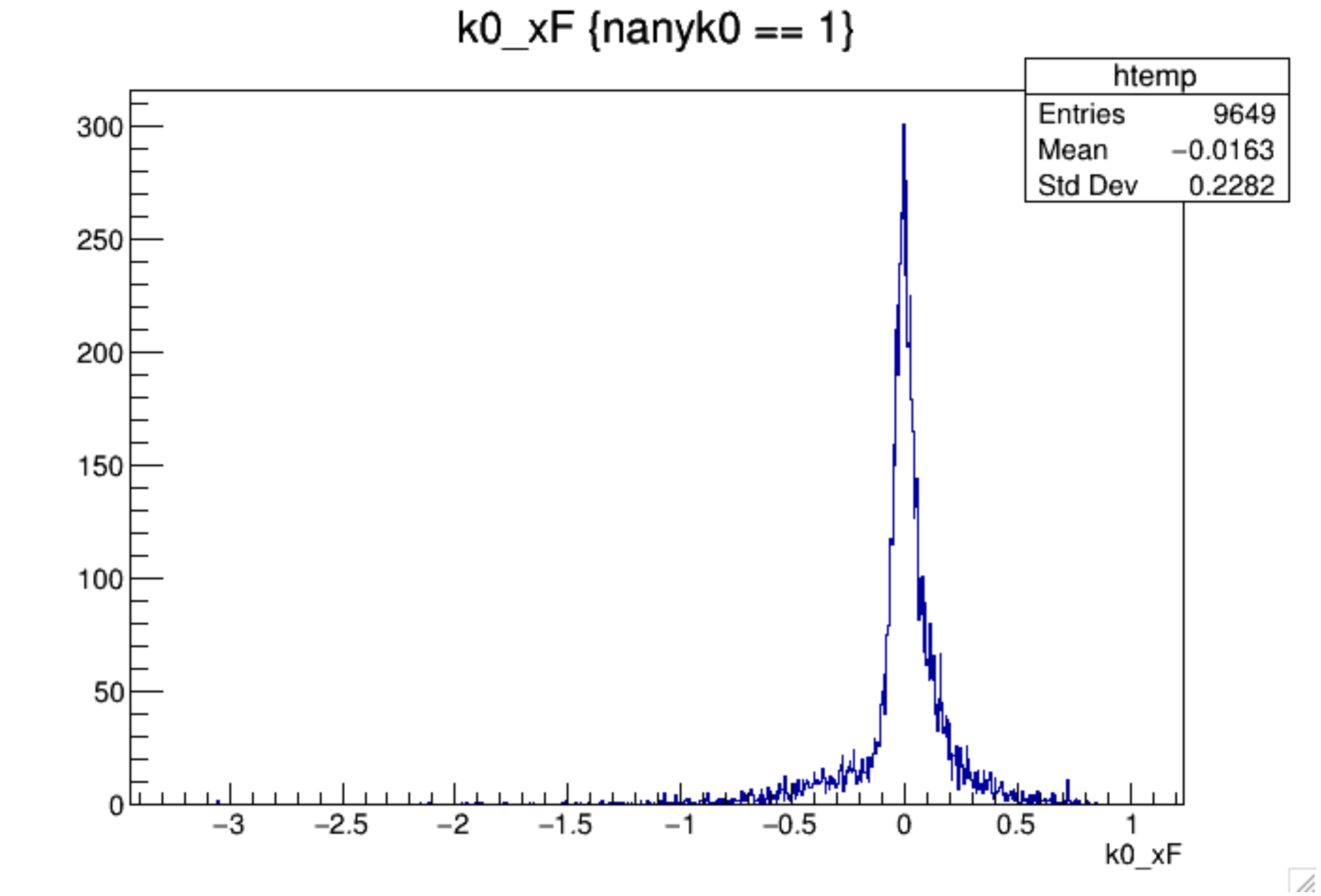


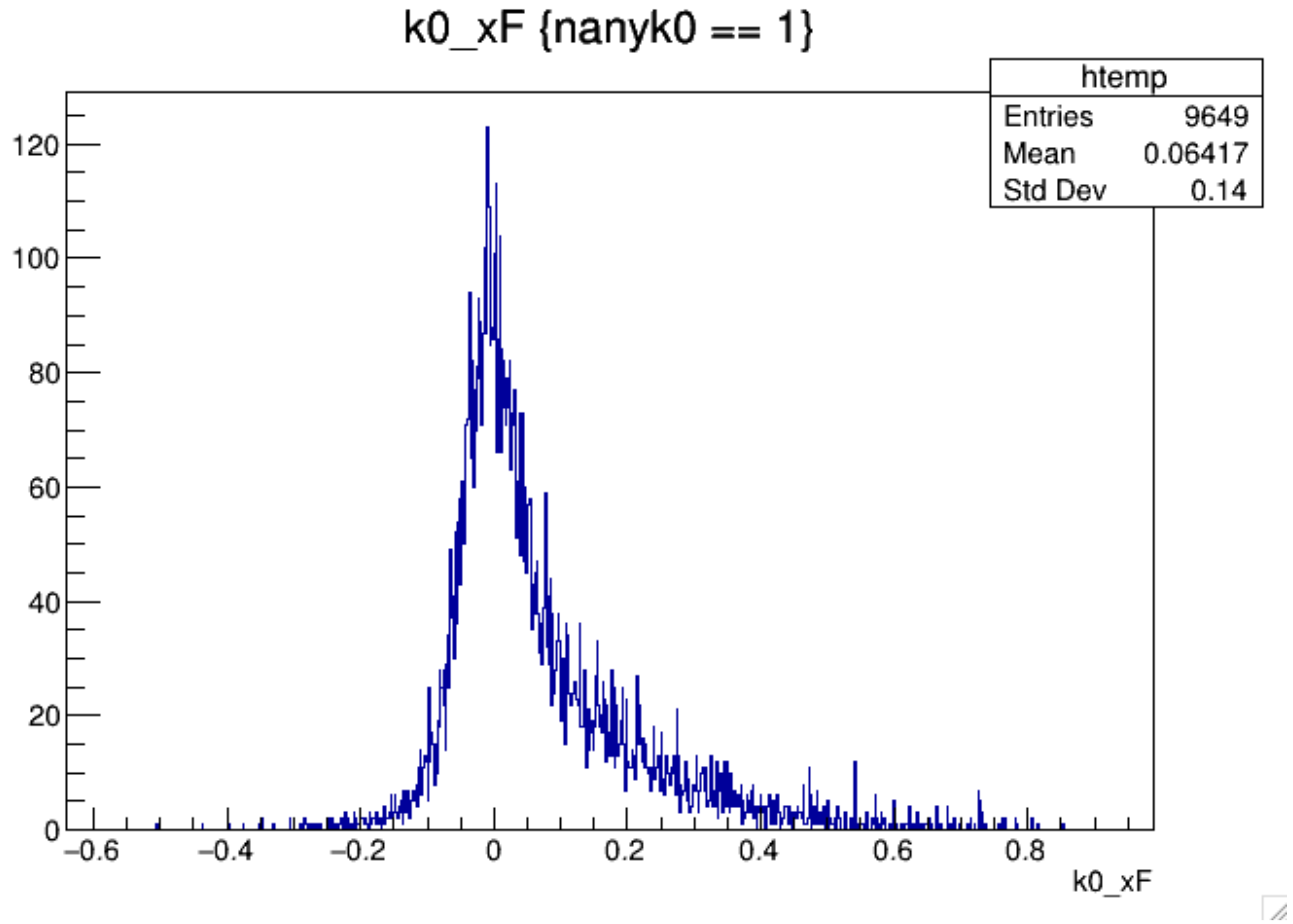
ν_e - FHC (Primary KLs)



- Uncertainty on primary KL as before since we know the hadron parent, xF bin
- Total uncertainty on all ν_e is not as high : ~1.5% increase from out-of-box PPFX

xF Calculation Issue





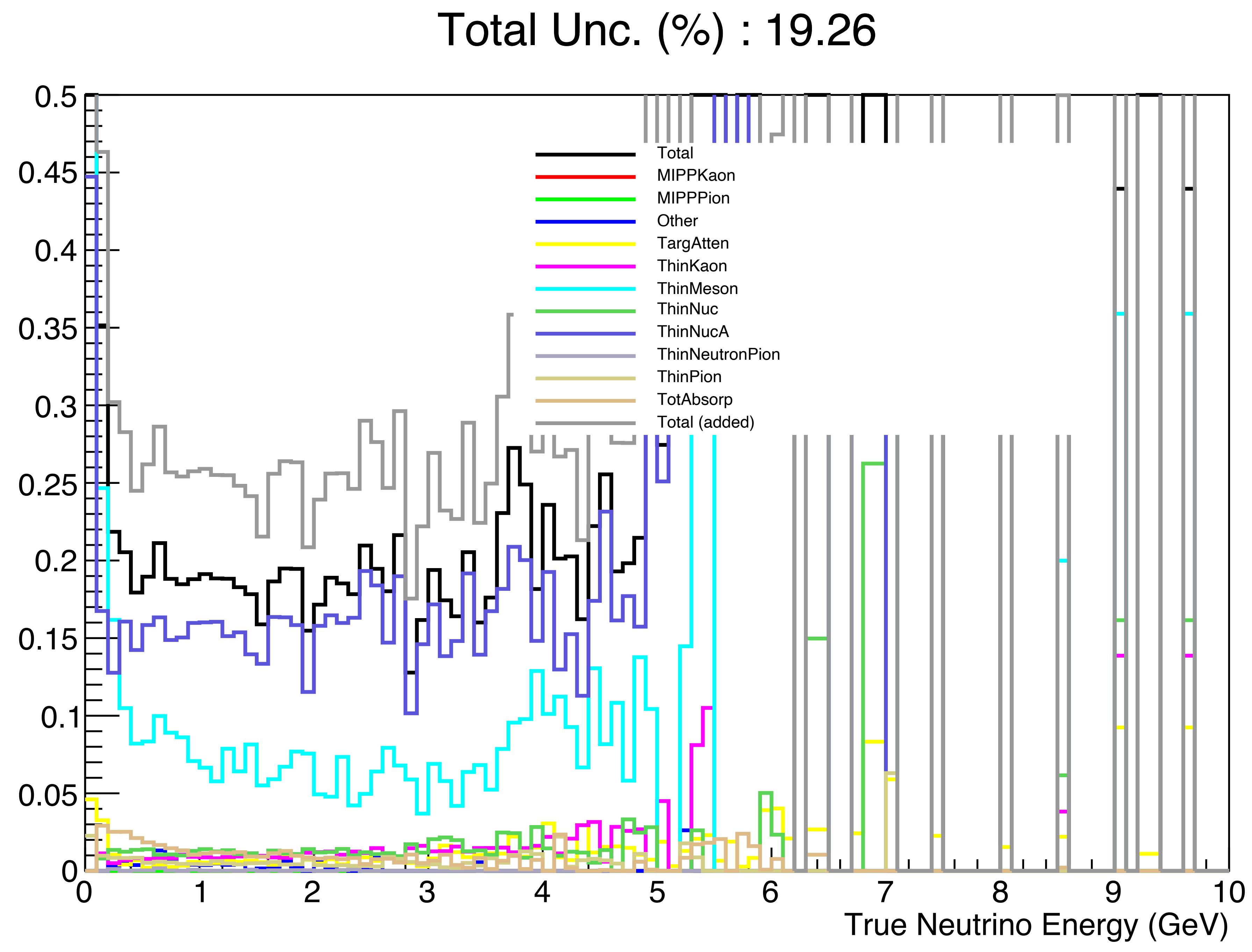
Found in ICARUS context

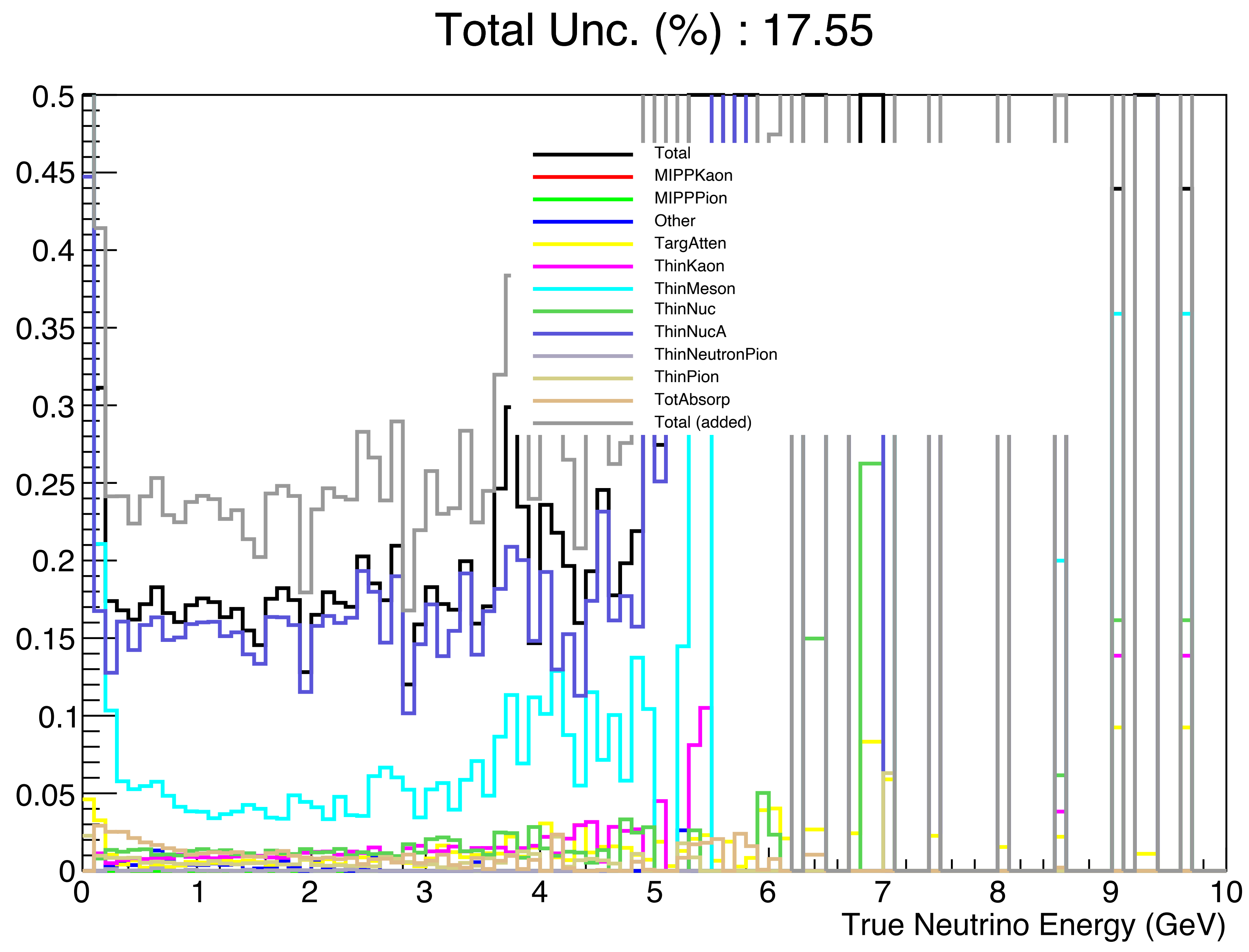
```
+    /**
+     * Center of mass of the system of projectile
+     * and nucleon (not the nucleus!)
+     */
+    static const double NUCLEON_MASS =
+      (particle->GetParticle("proton")->Mass()
+       + particle->GetParticle("neutron")->Mass())/2;
+    static const double NUCLEON_MASS2 = NUCLEON_MASS * NUCLEON_MASS;
+
     double inc_E_lab = std::sqrt(Inc_P*Inc_P + pow(Inc_Mass,2));
-    InteractionData::Ecm       = std::sqrt(2.*pow(Inc_Mass,2)+2.*inc_E_lab*Inc_Mass);
-    InteractionData::Betacm    = std::sqrt(pow(inc_E_lab,2)-pow(Inc_Mass,2.0))/(inc_E_lab + Inc_Mass);
+    InteractionData::Ecm       = std::sqrt(pow(Inc_Mass,2) + NUCLEON_MASS2 + 2.*inc_E_lab*NUCLEON_MASS);
+    InteractionData::Betacm    = std::sqrt(pow(inc_E_lab,2)-pow(Inc_Mass,2.0))/(inc_E_lab + NUCLEON_MASS);
     InteractionData::Gammacm   = 1./std::sqrt(1.-pow(Betacm,2.0));
```

- For xF calculation, PPFX assumes incident hadron particle has same mass as target nucleon. For meson incident processes especially, this is a problem
  - Pushed more towards negative values, even to unphysical
  - Fixing the calculation as in that presentation pushes it to more positive values (unphysical values gone as well)
- Haven't done extensive checks but ICARUS slides indicate significant impact (lower uncertainties in general but lower correlations between FHC/RHC and different flux flavors)

Impact on ν_e Uncertainties

40% fully corr.

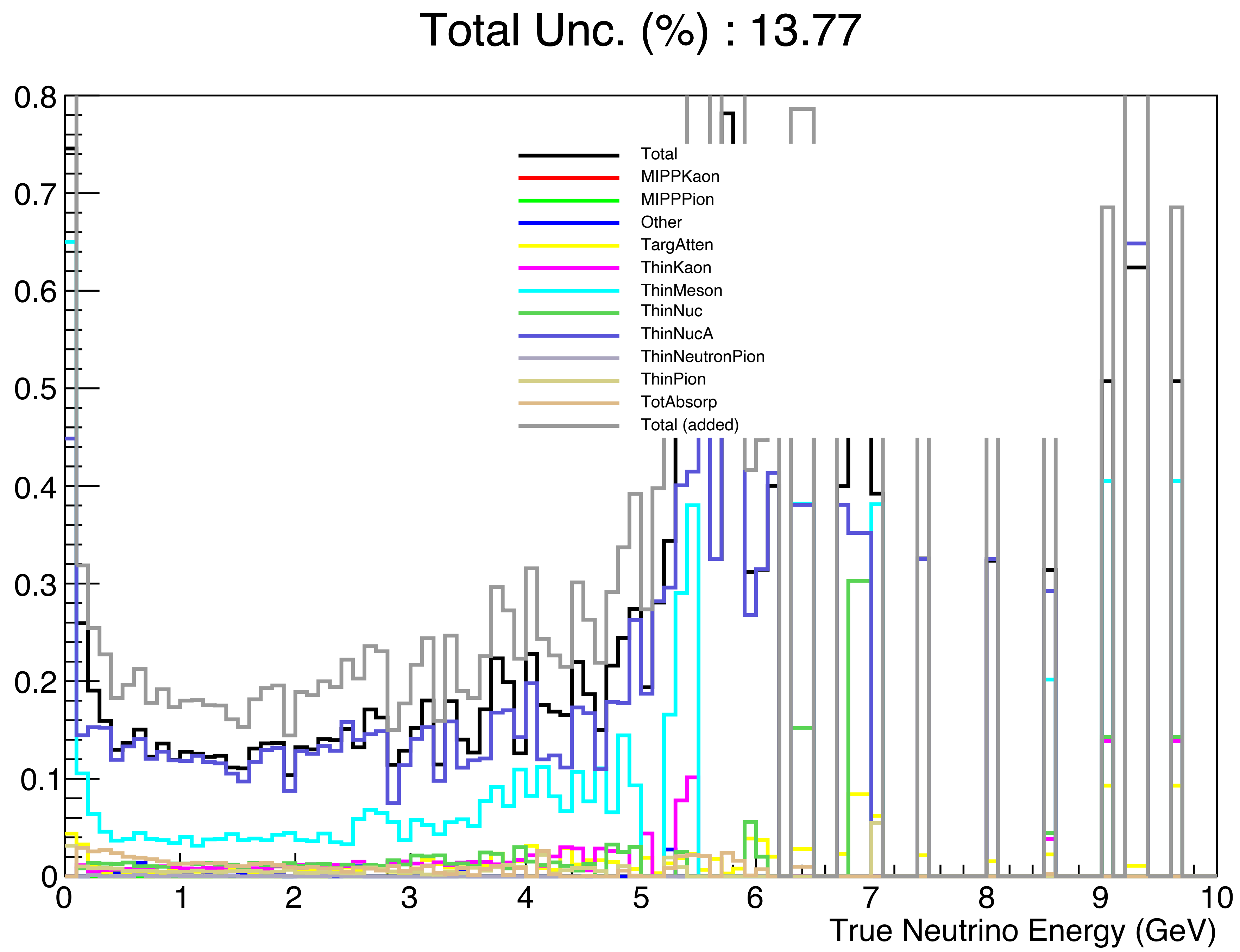




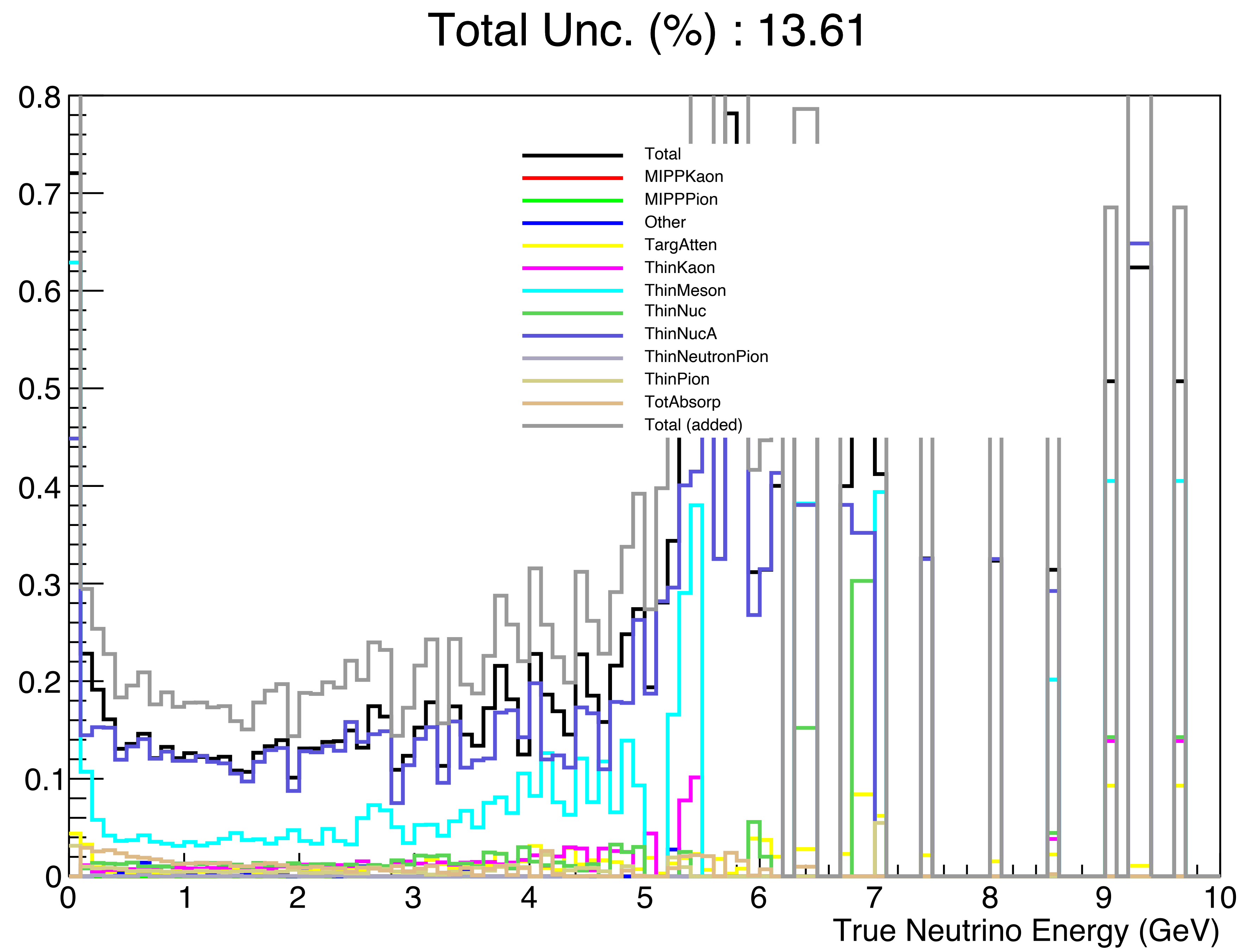
- If we assign 40% flat uncertainty to negative xF hadrons w/ nucleon parents (correlated across all species) :
  - Saw a 7% increase from out of box PPFX
  - After xF bug fix, back down to 17.5% ~ 5% increase from PPFX out of the box level
  - xF bug fix impacts MesonInc stage only pretty much (as expected since for nucleon parents of hadron species, xF calculation is correct)

Impact on ν_e Uncertainties

40% uncorr. (1 xF < 0 bin)



→



- If we assign 40% flat uncertainty to negative xF hadrons (uncorrelated across all hadron species/parents) :
  - Saw a 1.5% increase from out of box PPFX
  - After xF bug fix, doesn't change as much :
    - This makes sense since the only change comes from events that migrate across the xF bins but change is uncorrelated across parent/hadron species so it can go both ways

Summary Table

	Total nue	Total nue (xF Bug fix)	Primary KL parent	Primary KL parent (xF Bug fix)
Out of the Box	12.41%	-	10.37%	-
20% fully corr.	14.12%	12.54%	13.93%	13.97%
40% fully corr.	19.26%	17.55%	23.00%	22.85%
40% uncorr. w/ extra (-1, 0) xF bin	13.77%	13.61%	23.28%	23.28%
Mirrored	-	14.49%	-	26.60%
40% uncorr. w/ 4 extra xF bins between (-1, 0)	-	12.44%	-	22.65%
40% uncorr w/ 1 extra xF bin + 20% corr.	-	15.94%	-	26.23%
40% uncorr w/ 1 extra xF bin + 40% corr.	-	19.89%	-	32.20%
40% uncorr w/ 4 extra xF bins + 40% corr.	-	17.16%	-	33.20%

- Looked at 5 other scenarios (w/ xF bug fix) :
  - Assign same weight to xF < 0 as xF > 0 (maybe expected to be similar physics?)
  - Add 4 extra equally spaced xF bins b/w (-1, 0) for a total of 8 bins instead : 40% uncorr. in each, separate for each hadron species, parent
  - 40% uncorr. + 40% fully corr. across (xF, hadron species) (x 3 knobs for nucleons/meson parent) : 4 negative xF bins
  - 40% uncorr. + 20% fully corr. across (xF, hadron species) (x 3 knobs for nucleons/meson parent) : just 1 negative xF bin
  - 40% uncorr. + 40% fully corr. across (xF, hadron species) (x 3 knobs for nucleons/meson parent) : just 1 negative xF bin

Nucleon-A : $PPFX_DIR/src/ThinTargetnucleonAReweighter.cpp

```cpp
    int binnu     = Thinbins->meson_inc_BinID(aa.xF,aa.Pt,211);
    // if(binnu < 0) return 1.0;  // orig ppfx
    // if(binnu<0){
    //   binnu = Thinbins->meson_inc_BinID(-aa.xF, aa.Pt, 211); // mirrored ppfx
    // }
    if(binnu < 0){
      if(aa.Inc_pdg == 2212) return bin_prtleftover_inc;         // 40% corr or after adding negative xF bins
      else if(aa.Inc_pdg == 2112) return bin_neuleftover_inc;
    }

    if(aa.Inc_pdg ==2212){
      if(aa.Prod_pdg == 211) wgt = vbin_prt_inc_pip[binnu];
      else if(aa.Prod_pdg ==-211) wgt = vbin_prt_inc_pim[binnu];
      else if(aa.Prod_pdg == 321) wgt = vbin_prt_inc_kap[binnu];
      else if(aa.Prod_pdg ==-321) wgt = vbin_prt_inc_kam[binnu];
      else if(aa.Prod_pdg ==130 || aa.Prod_pdg ==310) wgt = vbin_prt_inc_k0[binnu];
      else if(aa.Prod_pdg ==2212) wgt = vbin_prt_inc_p[binnu];
      else if(aa.Prod_pdg ==2112) wgt = vbin_prt_inc_n[binnu];
      else wgt = bin_prtleftover_inc;
      // if(aa.xF < 0.) wgt *= ((bin_prtleftover_inc-1.)*0.5) + 1.;    // 20% corr + 40% uncorr after adding negative xF bins
      // if(aa.xF < 0.) wgt *= bin_prtleftover_inc;                   // 40% corr + 40% uncorr after adding negative xF bins
    }
    else if(aa.Inc_pdg ==2112){
      if(aa.Prod_pdg == 211) wgt = vbin_neu_inc_pip[binnu];
      else if(aa.Prod_pdg ==-211) wgt = vbin_neu_inc_pim[binnu];
      else if(aa.Prod_pdg == 321) wgt = vbin_neu_inc_kap[binnu];
      else if(aa.Prod_pdg ==-321) wgt = vbin_neu_inc_kam[binnu];
      else if(aa.Prod_pdg ==130 || aa.Prod_pdg ==310) wgt = vbin_neu_inc_k0[binnu];
      else if(aa.Prod_pdg ==2212) wgt = vbin_neu_inc_p[binnu];
      else if(aa.Prod_pdg ==2112) wgt = vbin_neu_inc_n[binnu];
      else wgt = bin_neuleftover_inc;
      // if(aa.xF < 0.) wgt *= ((bin_neuleftover_inc-1.)*0.5) + 1.;
      // if(aa.xF < 0.) wgt *= bin_neuleftover_inc;
    }
    if(wgt < 0.) return 1.0;
    if(wgt > 10.) return 1.0;

    return wgt;

  }


}
"src/ThinTargetnucleonAReweighter.cpp" 267L, 12179C
```

$PPFX_DIR/uncertainties/
Parameters_default.xml

```xml
<ThinTarget_neu_incident_pim>
<cvs>1.0 1.0 1.0 1.0</cvs>
<errs>0.4 0.4 0.4 0.4</errs>
</ThinTarget_neu_incident_pim>

<ThinTarget_neu_incident_kap>
<cvs>1.0 1.0 1.0 1.0</cvs>
<errs>0.4 0.4 0.4 0.4</errs>
</ThinTarget_neu_incident_kap>

<ThinTarget_neu_incident_kam>
<cvs>1.0 1.0 1.0 1.0</cvs>
<errs>0.4 0.4 0.4 0.4</errs>
</ThinTarget_neu_incident_kam>

<ThinTarget_neu_incident_k0>
<cvs>1.0 1.0 1.0 1.0</cvs>
<errs>0.4 0.4 0.4 0.4</errs>
</ThinTarget_neu_incident_k0>

<ThinTarget_neu_incident_p>
<cvs>1.0 1.0 1.0 1.0</cvs>
<errs>0.4 0.4 0.4 0.4</errs>
</ThinTarget_neu_incident_p>

<ThinTarget_neu_incident_n>
<cvs>1.0 1.0 1.0 1.0</cvs>
<errs>0.4 0.4 0.4 0.4</errs>
</ThinTarget_neu_incident_n>

<ThinTarget_neuleftover_incident>
<cvs>1.0</cvs>
<errs>0.4</errs>
</ThinTarget_neuleftover_incident>
```

$PPFX_DIR/data/BINS/
ThinTarget_MesonIncident.xml

```xml
<bins>
<!-- List of bins -->

    <ThinTarget_MesonIncident>

      <ThinTarget_MesonIncident_0>
        <xfrange>0.00 0.25</xfrange>
        <ptrange>0.00 2.00</ptrange>
      </ThinTarget_MesonIncident_0>

      <ThinTarget_MesonIncident_1>
        <xfrange>0.25 0.50</xfrange>
        <ptrange>0.00 2.00</ptrange>
      </ThinTarget_MesonIncident_1>

      <ThinTarget_MesonIncident_2>
        <xfrange>0.50 0.75</xfrange>
        <ptrange>0.00 2.00</ptrange>
      </ThinTarget_MesonIncident_2>

      <ThinTarget_MesonIncident_3>
        <xfrange>0.75 1.0</xfrange>
        <ptrange>0.00 2.00</ptrange>
      </ThinTarget_MesonIncident_3>

    </ThinTarget_MesonIncident>


</bins>
```

canReweight conditions

pC->KX : $PPFX_DIR/src/ThinTargetpCKaonReweighter.cpp

```
bool ThinTargetpCKaonReweighter::canReweight(const InteractionData& aa){
  //checking:
  std::string mode(getenv("MODE"));
  if(aa.Inc_pdg != 2212)return false;
  if(aa.Inc_P < 12.0)return false;
  //volume check:
  bool is_wrong_volume = aa.Vol != "TGT1" && aa.Vol != "BudalMonitor" && aa.Vol != "Budal_HFVS" && aa.Vol != "Budal_VFHS";
  if( (mode=="REF") || (mode=="OPT") ){
    is_wrong_volume = aa.Vol != "TargetFinHorizontal" && aa.Vol != "TargetNoSplitSegment";
  }
  if(is_wrong_volume)return false;
  //
  if(aa.Prod_pdg != 321 && aa.Prod_pdg != -321 && aa.Prod_pdg != 310 && aa.Prod_pdg != 130)return false;

  //Looking for low pz kaon:
  ThinTargetBins*  Thinbins =  ThinTargetBins::getInstance();
  int bin      = Thinbins->BinID_pC_k(aa.xF,aa.Pt,aa.Prod_pdg);
  if(bin>=0)return true; //found NA49 kaon

  //Looking for high pz kaon:
  int mipp_bin = Thinbins->mipp_BinID_pC_k(aa.Pz,aa.Pt,aa.Prod_pdg);
  if(mipp_bin<0)return false; //no mipp thin target kaon

```

MesonInc : $PPFX_DIR/src/ThinTargetMesonIncidentReweighter.cpp

```
bool ThinTargetMesonIncidentReweighter::canReweight(const InteractionData& aa){
  /*
  if(aa.Inc_pdg != 211 && aa.Inc_pdg != -211 && aa.Inc_pdg != 321 && aa.Inc_pdg != -321 && aa.Inc_pdg != 130 && aa.Inc_pdg !
  if(aa.Prod_pdg != 211 && aa.Prod_pdg != -211 && aa.Prod_pdg != 321 && aa.Prod_pdg != -321 && aa.Prod_pdg != 130 && aa.Proc
  */
  if(aa.Proc.find("Inelastic")>100)return false;

  //  ThinTargetBins*  Thinbins =  ThinTargetBins::getInstance();
  //int bin      = Thinbins->meson_inc_BinID(aa.xF,aa.Pt,aa.Prod_pdg);

  bool is_mesoninc = (aa.Inc_pdg >99 && aa.Inc_pdg < 1000) || (aa.Inc_pdg <-99 && aa.Inc_pdg > -1000);

  //    if(bin>=0 || is_mesoninc)return true;
  if(is_mesoninc)return true;
  else return false;

}
```

Nucleon-A : $PPFX_DIR/src/ThinTargetnucleonAReweighter.cpp

```
bool ThinTargetnucleonAReweighter::canReweight(const InteractionData& aa){

  //checking:
  if(aa.Inc_pdg != 2212 && aa.Inc_pdg != 2112)return false;
```

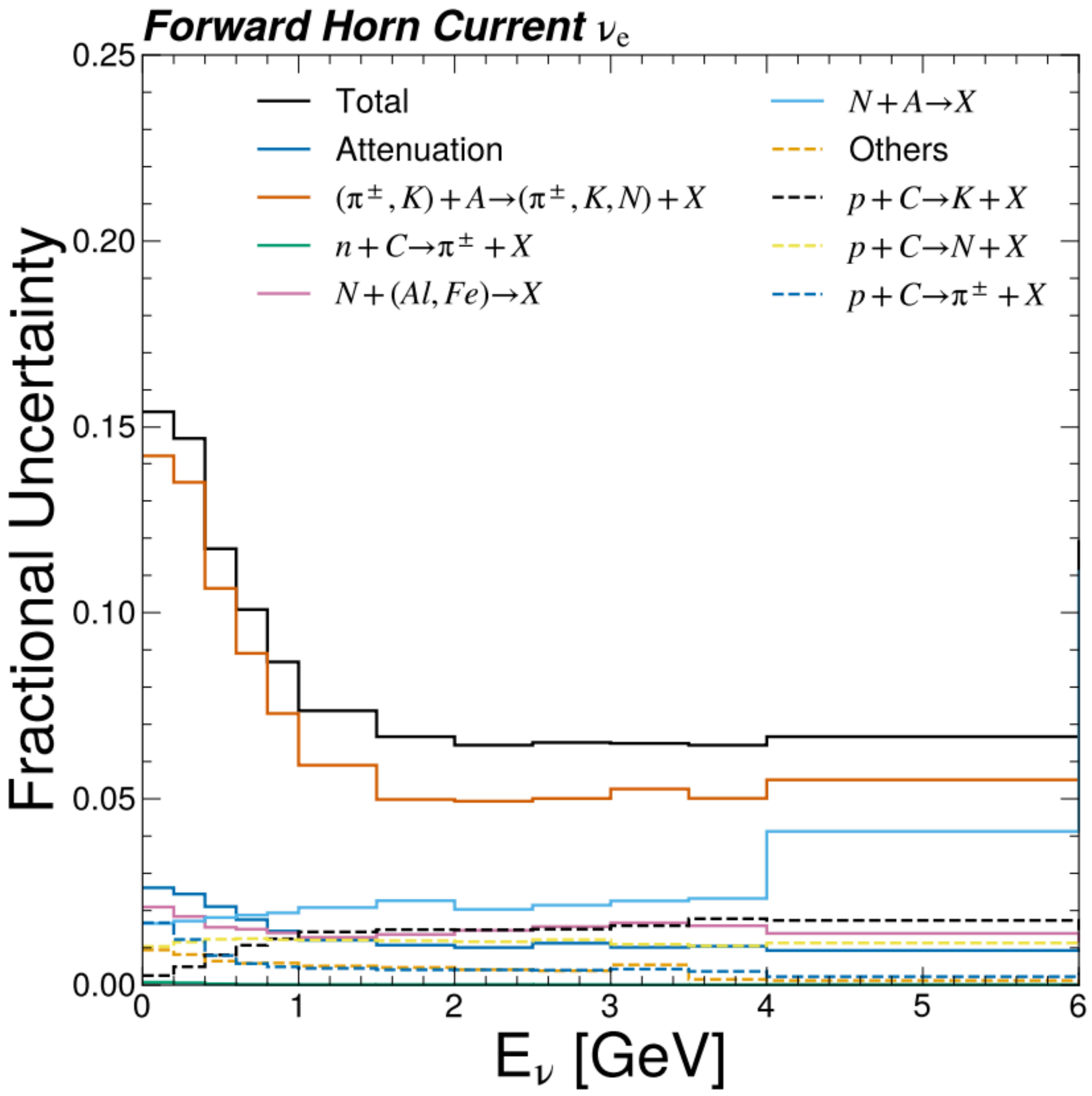
Other : $PPFX_DIR/src/OtherReweighter.cpp

```
bool OtherReweighter::canReweight(const InteractionData& aa){

  if(aa.Proc.find("Inelastic")<100){
    return true;
  }
  else return false;

}
```

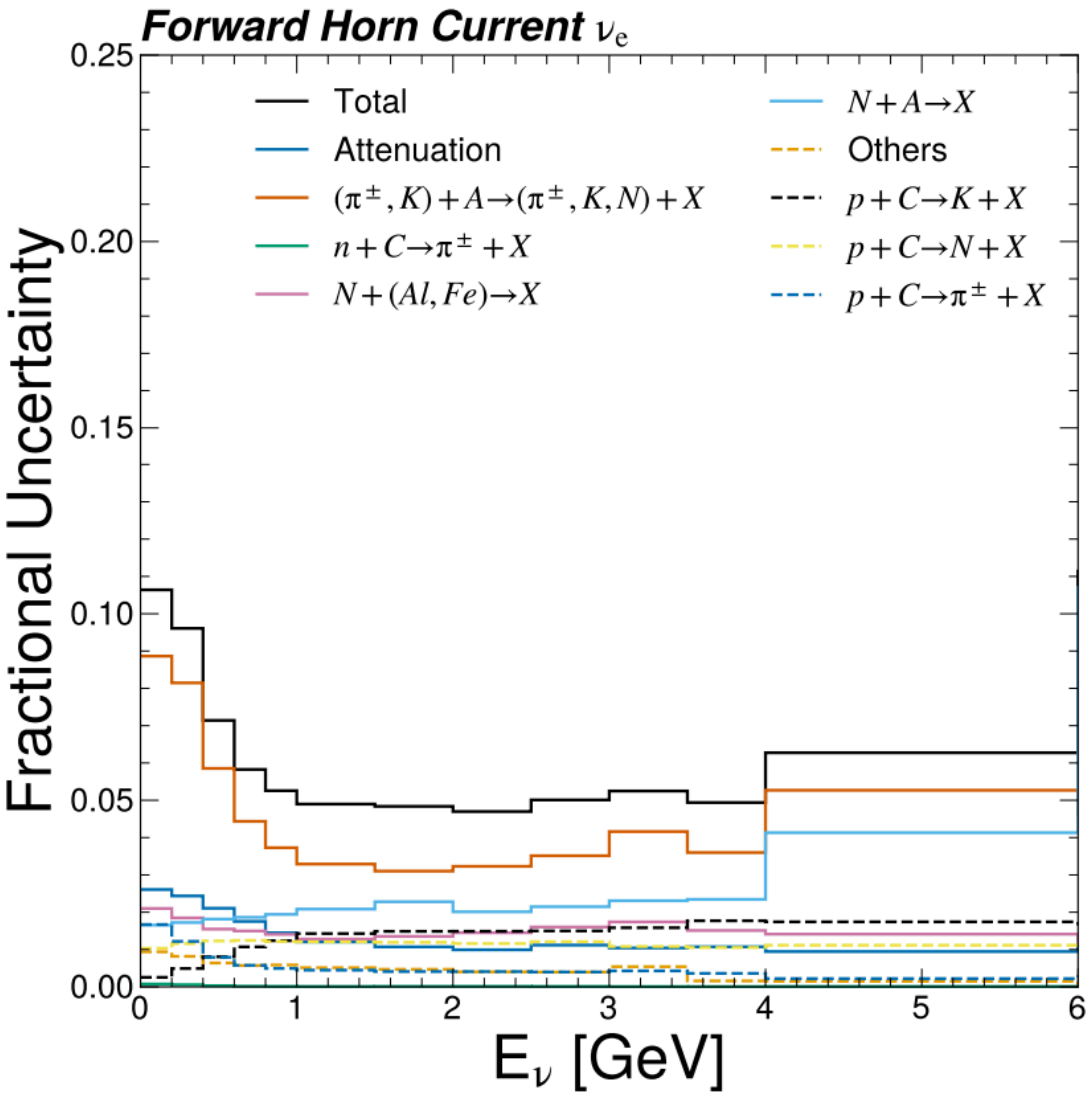
- NB : if hadron passes the `canReweight` condition of some stage, a weight always gets assigned and other stages are ignored for that hadron
- If it fails, then it moves to the next `canReweight` condition
- For nucleon-A the condition only checks if parent of that hadron species is a nucleon

ν_e total uncertainties

Before the bugfix

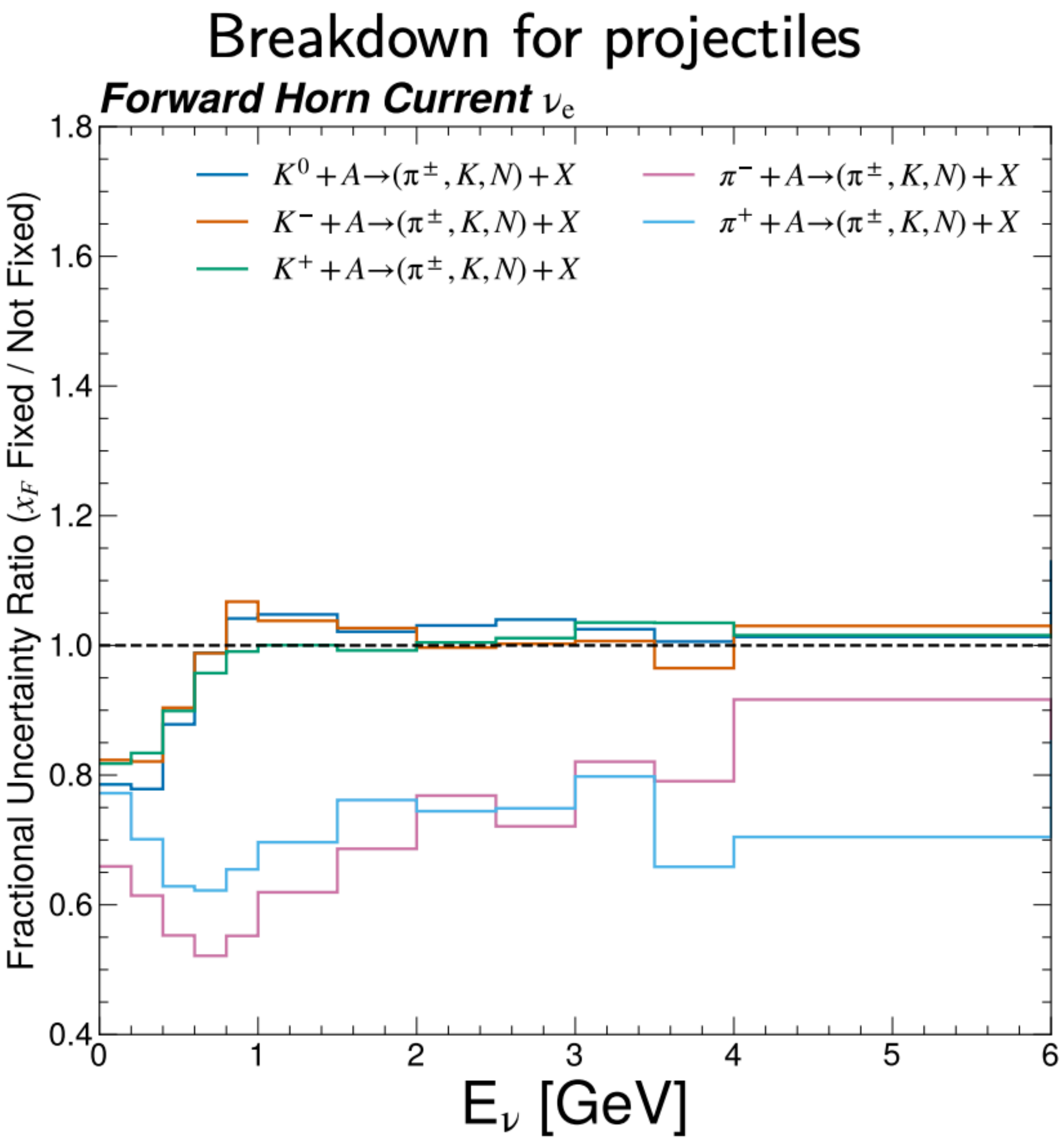


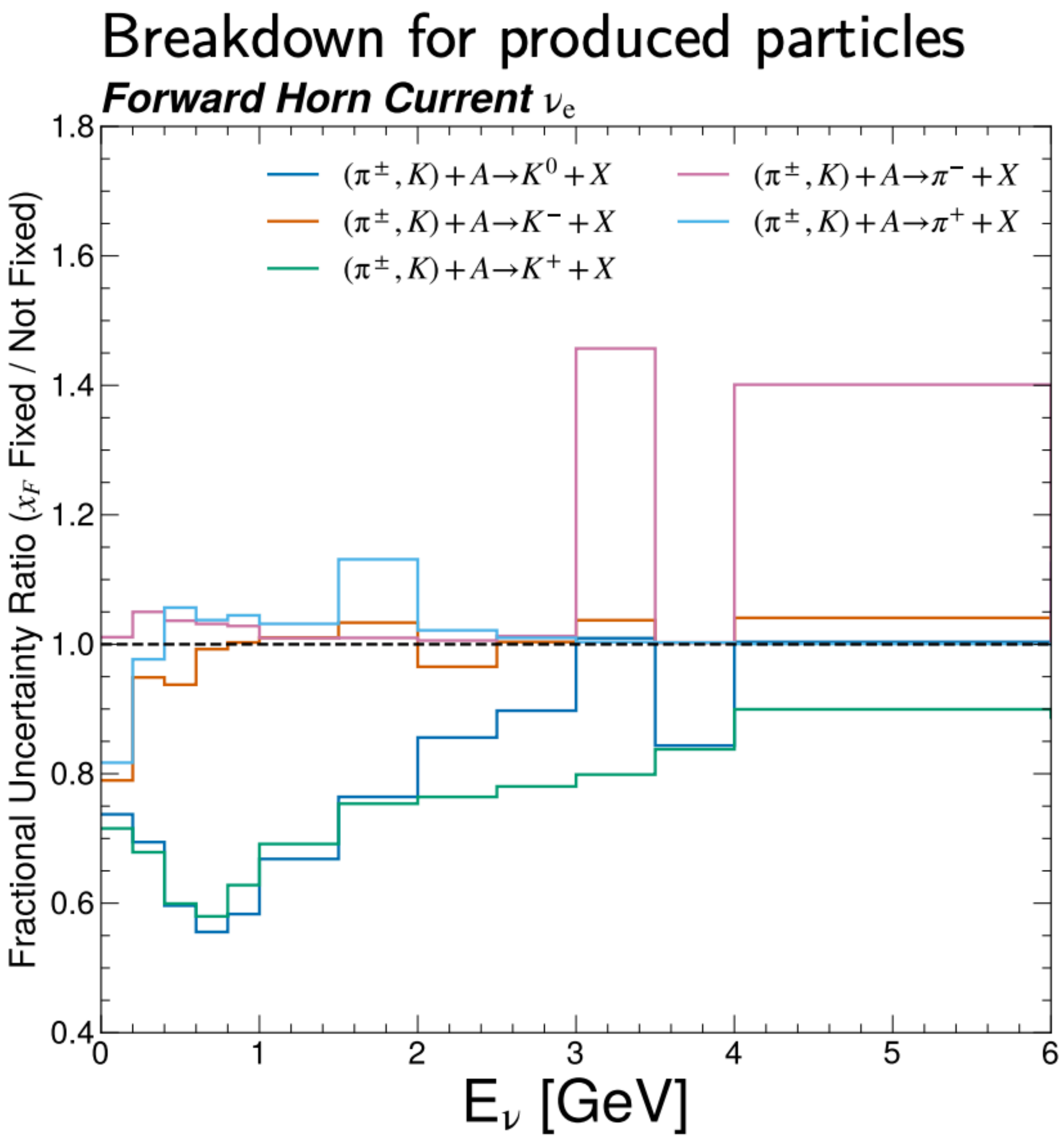
With the bugfix



- Total uncertainty decreases by 5% at low energies

ν_e breakdown of impact of the bugfix

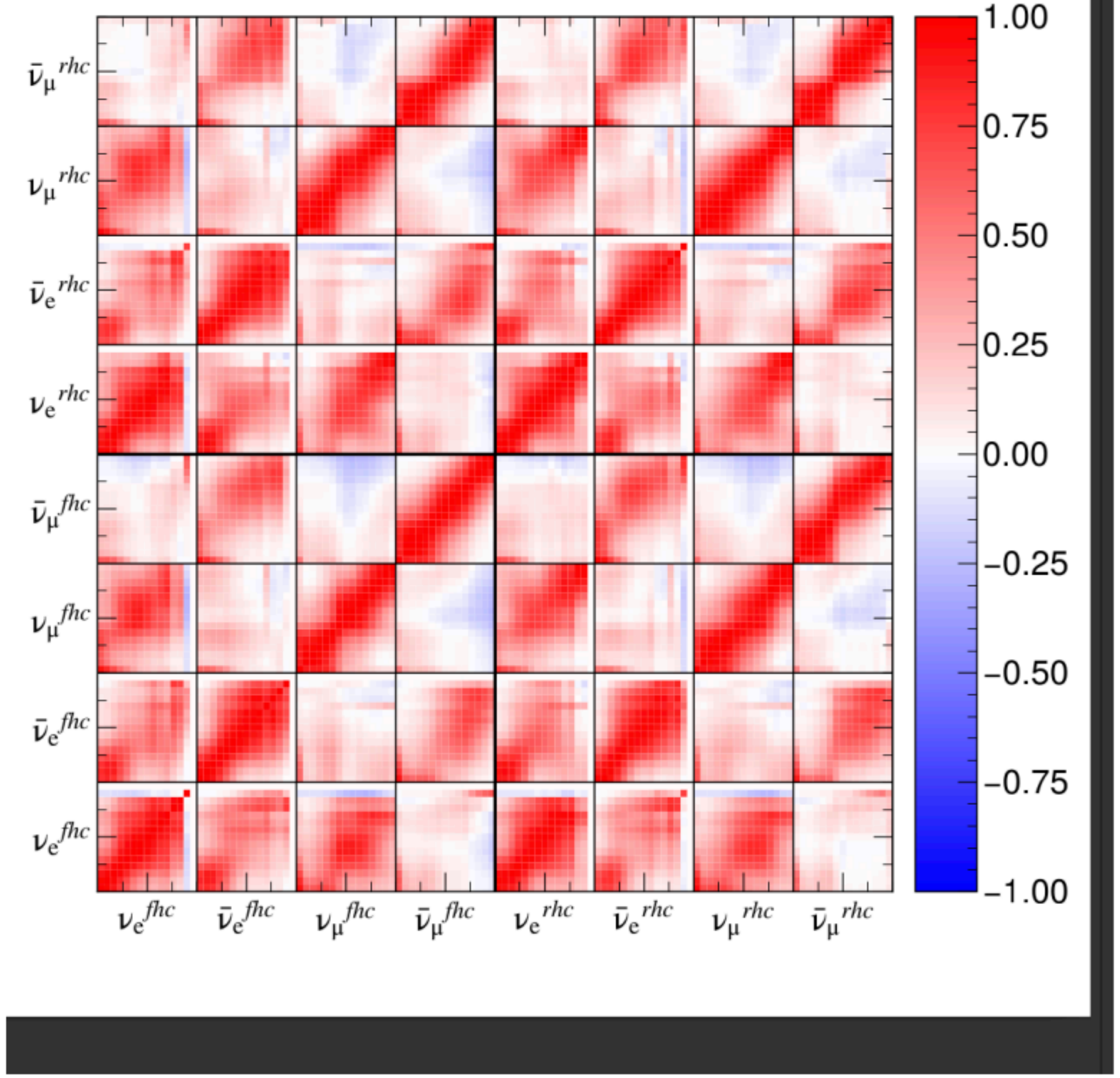




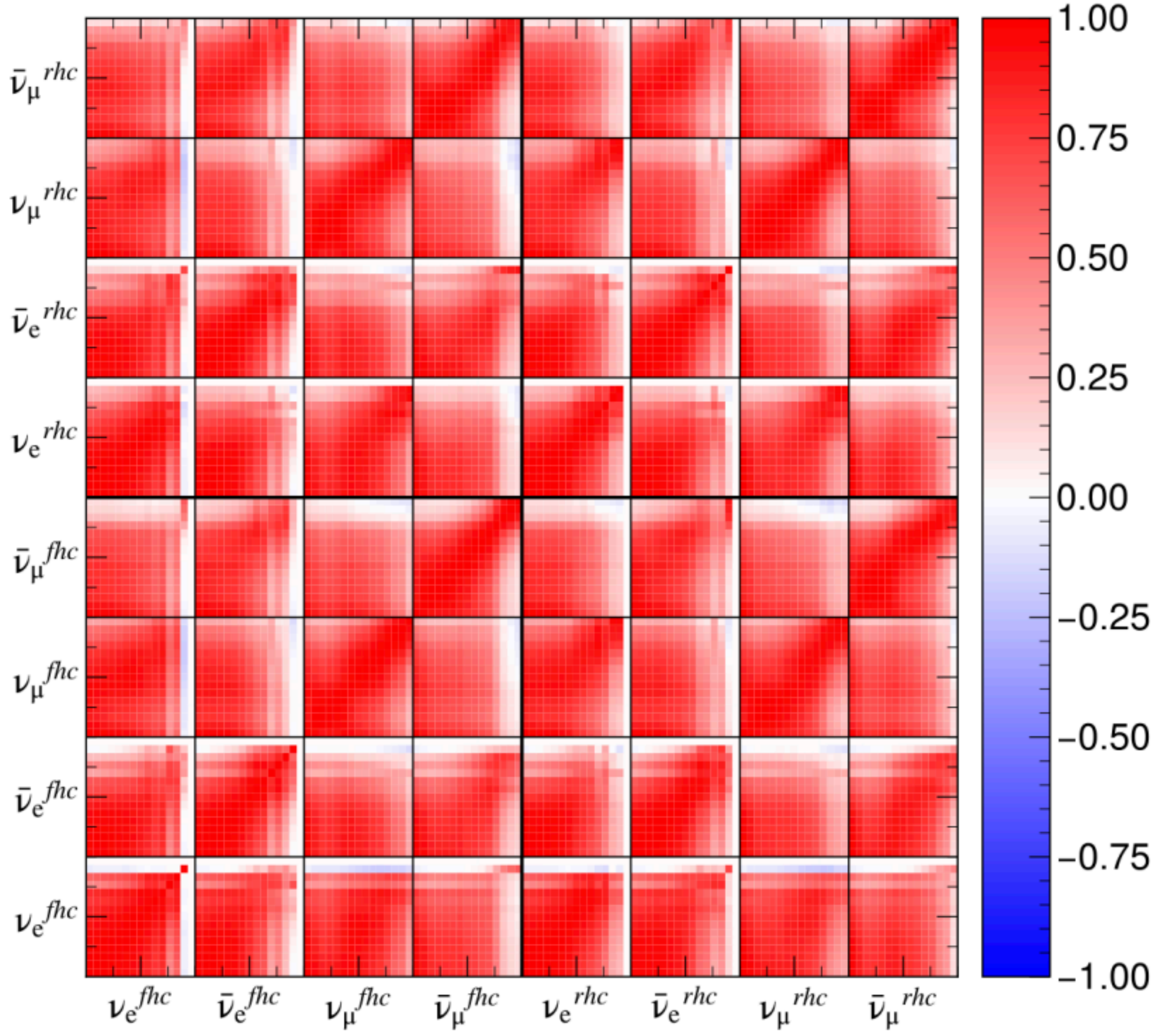
- The largest impact for π

https://indico.fnal.gov/event/59606/contributions/265597/attachments/166570/221916/xf_bugfix.pdf

With bugfix



Without bugfix



- Huge decrease of correlations