Validation of single-particle test samples with SDHCal in ILCSoft

SDHCAL group meeting

06/05/21

Camilo,Guillaume for the SDHCAL group

GOBIERNO DE ESPAÑA
MINISTERIO DE CIENCIA E INNOVACIÓN

Ciemat
Centro de Investigaciones Energéticas, Medioambientales y Tecnológicas

cfp CIEMAT física de partículas

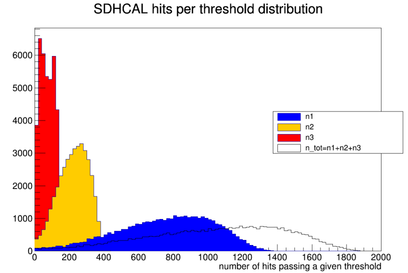
Introduction

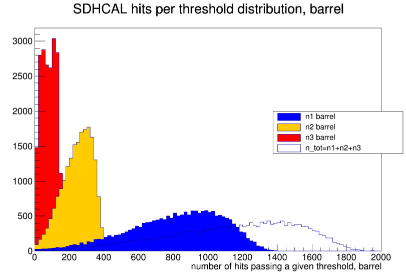
- A bug in ILCSoft concerning the managing of the overlay for the SDHCAL has been identified and solved by Hector on the first of March..
- Link to the PR $\rightarrow$ `https://github.com/iLCSoft/Overlay/pull/21`
- The pull request has been merged in ILCSoft and a new MC production for the o2 has been produced on the 30th of march.
  - single particle of muon and K_0^L `https://ild.ngt.ndu.ac.jp/elog/dbd-prod/467`
  - inclusive H $\rightarrow$ qqh, which was requested in January.`https://ild.ngt.ndu.ac.jp/elog/dbd-prod/468`
- In this presentation we are showing our SDHCAL hit and energy validation for this new MC production.

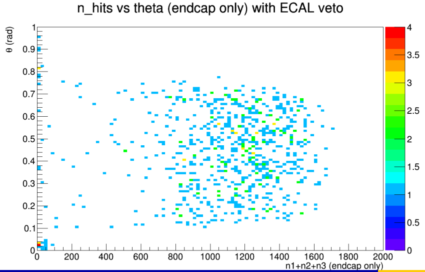
Wofkflow

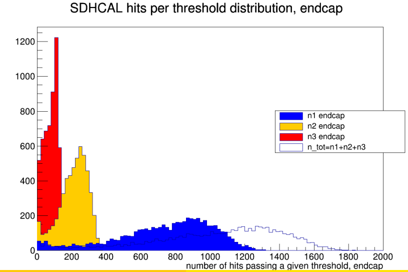
- We focus for the moment only in the K_0^L sample.
  o2 Energy range: (1,2,5,10,20,30,40,50,60,70,80,90,100,110) GeV.
- A copy of the *dstm* dataset to our SDHCAL-cluster in CIEMAT has been done and it is available (for those with access) here: `/pool/calice3/data/MonteCarlo/sdhcal_validation/k0_long_v02-02`
- A copy of the *rec* dataset to our SDHCAL-cluster in CIEMAT has been done and it is available (for those with access) here: `/pool/calice3/data/MonteCarlo/sdhcal_validation/k0_long_v02-02/rec`
- Out of these datasets, two LCTuples have been produced for every single energy point:
  - LCTuple by default. `/pool/calice2/data/Simulacion/sdhcal_validation/k0_long_v02-02/lctuples/`
  - Customized LCTuple including the ECAL hits (to correctly identify the parton showers startin in our subdetector, explained in next slides) `/pool/calice2/data/Simulacion/sdhcal_validation/k0_long_v02-02/rec/lctuples/with_ecal`
- We select events with SDHCAL hits only in the endcap and we a applay a ECAL veto to ensure that the shower actually started in the SDHCAL.

SDHCAL Hit Level Analysis 110 GeV, with overlay bug

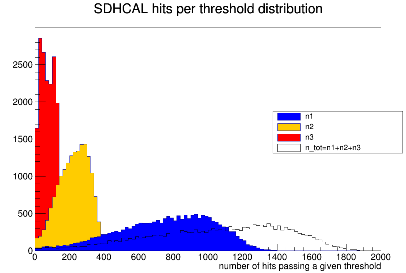


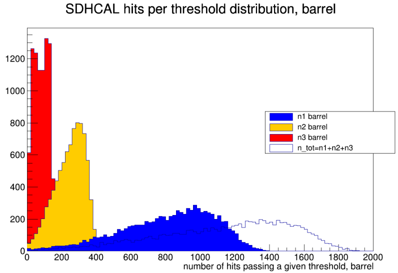


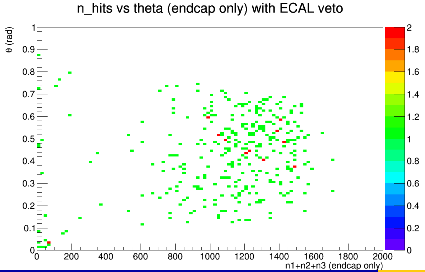


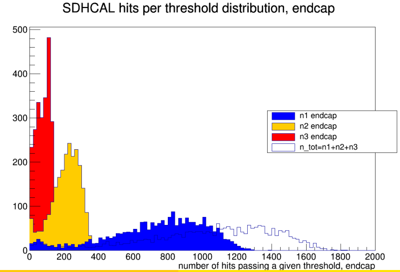


SDHCAL Hit Level Analysis 110 GeV, without overlay bug

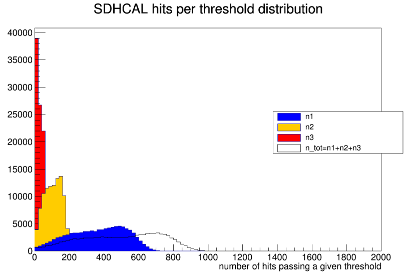


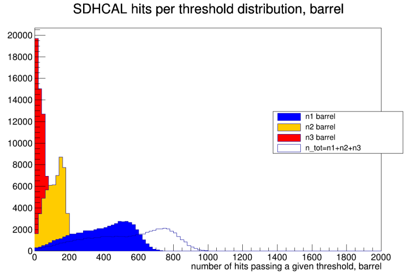


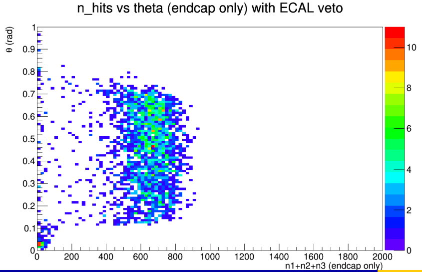


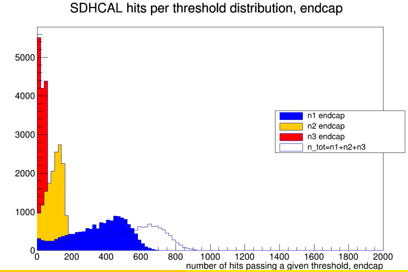


SDHCAL Hit Level Analysis 50 GeV, without overlay bug

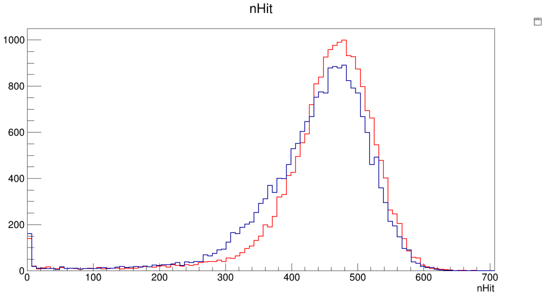








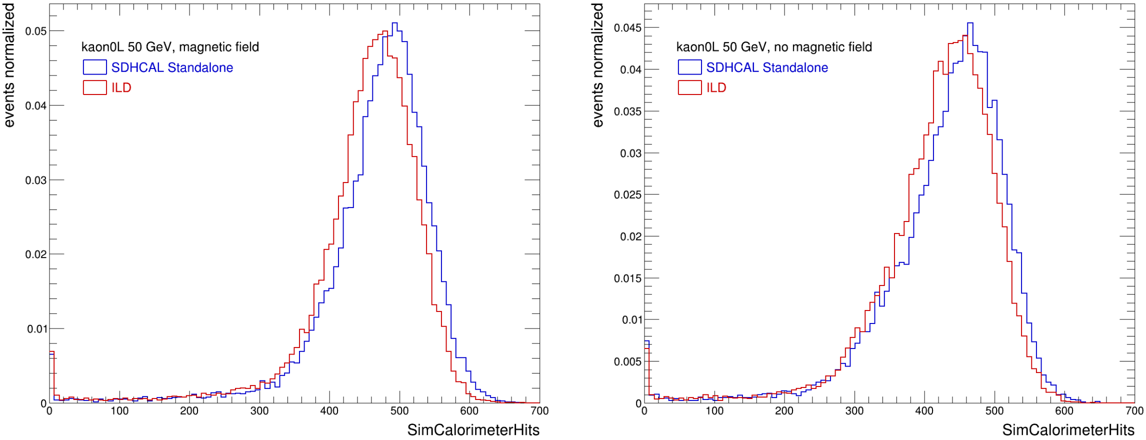
SDHCAL K_L^0 Geant 4 standalone simulation of prototype (50 GeV):



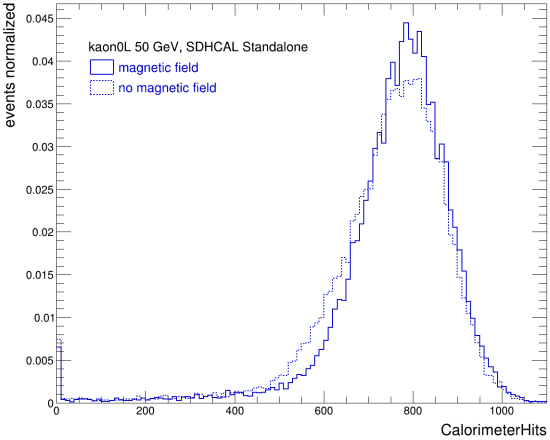
- SimCalorimeterHits number is compatible between ILD and standalone SDHCAL (red ILD, blue SDHCAL).
- Tails discrepancy

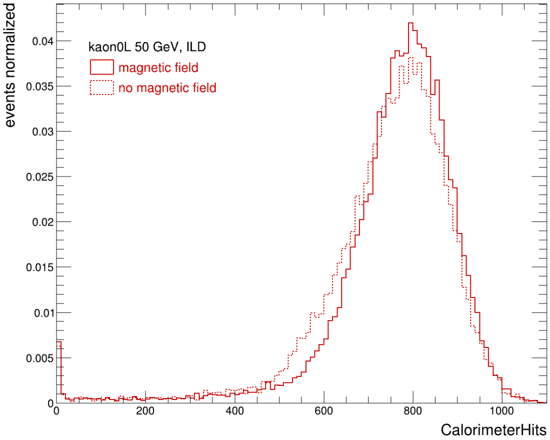
- K_0^L in ILD just before the SDHCAL (after the ECAL), and in the middle of a layer (to avoid cracks)
- The model is ILD-l5-v02 hybrid Tesla
- The gun position is x = 2050mm y = 550mm z = 1120mm (after the ECAL) to avoid leaks.
- SDHCAL standalone simulation adapted to match the ILD parameters but it is not perfect :
  - pad size from 10.408mm $\times$ 10.408mm to 10mm $\times$ 10mm
  - steel density from 8 to 7.85
  - hybrid model in the standalone simulation (G4-POLYSTYRENE instead of our

SDHCAL StandAlone and ILCSoft simulation SimHits with and without Magnetic field (50 GeV) per thershold:

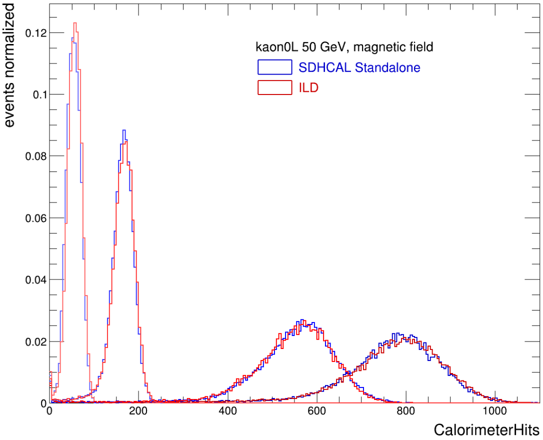


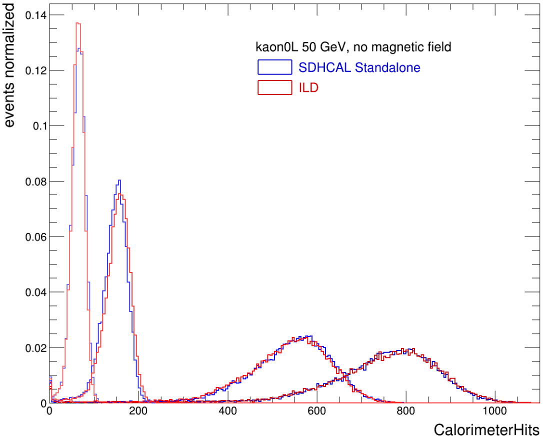
SDHCAL StandAlone and ILCSoft simulation recHits with and without Magnetic field (50 GeV):





SDHCAL StandAlone and ILCSoft simulation **recHits** with and without Magnetic field (50 GeV) per thershold:





Conclusion

- all plots `http://wwwae.ciemat.es/~carrillo/calice/sdhcal_hit_analysis_withECAL_after_bug/`
- Agreement ILDSoft vs our simulation seems to be understood

Backup

Identifying the start of the shower to filter events

We had 3 options we use the ECAL one:

- The parameter should be in ilcSoft (maybe stored in the LCTuple may be not (if not we can add it and produce again the ntuples).
- As another option we can try to identify the position where the shower starts using the test beam analysis tools (Hector)
  - What information would be needed?
  - x, y, z or i, j(path) plane of each hit would be enough?
- As you can see in the list of available variables the most promising variables: rcmo(x, y, z)[nrec]
- **As a 3rd option, selecting events without ECAL activity**

The Standard LCTuple xml Marling, no ECAL hits

```xml
<processor name="MergeSimCaloHits" type="MergeCollections">
  <!--MergeCollections creates a transient subset collection that merges all input collections -->
  <!--Names of all input collections-->
  <parameter name="InputCollections" type="StringVec">
    BeamCalCollection
    EcalBarrelCollection
    EcalEndcapRingCollection
    EcalEndcapsCollection
    HcalBarrelRegCollection
    HcalEndcapRingCollection
    HcalEndcapsCollection
    LumiCalCollection
    YokeBarrelCollection
    YokeEndcapsCollection
    LHCalCollection
  </parameter>
 <!--Optional IDs for input collections - if given id will be added to all objects in merged collections as ext<CollID>()-->
  <parameter name="InputCollectionIDs" type="IntVec">
    25 20 21 29 22 23 30 24 27 31
  </parameter>
  <parameter name="OutputCollection" type="string">SimCalorimeterHits </parameter>
  <!--verbosity level of this processor ("DEBUG0-4,MESSAGE0-4,WARNING0-4,ERROR0-4,SILENT")-->
  <parameter name="Verbosity" type="string">DEBUG </parameter>
</processor>


<processor name="MergeCaloHits" type="MergeCollections">
  <!--MergeCollections creates a transient subset collection that merges all input collections -->
  <!--Names of all input collections-->
  <parameter name="InputCollections" type="StringVec">
    HcalBarrelCollectionDigi
    HcalEndcapRingCollectionDigi
```

The Standard LCTuple xml Marling, with ECAL hits

```xml
<processor name="MergeSimCaloHits" type="MergeCollections">
  <!--MergeCollections creates a transient subset collection that merges all input collections -->
  <!--Names of all input collections-->
  <parameter name="InputCollections" type="StringVec">
    BeamCalCollection
    EcalBarrelCollection
    EcalEndcapRingCollection
    EcalEndcapsCollection
    HcalBarrelRegCollection
    HcalEndcapRingCollection
    HcalEndcapsCollection
    LumiCalCollection
    YokeBarrelCollection
    YokeEndcapsCollection
    LHCalCollection
  </parameter>
 <!--Optional IDs for input collections - if given id will be added to all objects in merged collections as ext<CollID>()-->
  <parameter name="InputCollectionIDs" type="IntVec">
    25 20 21 29 22 23 30 24 27 31
  </parameter>
  <parameter name="OutputCollection" type="string">SimCalorimeterHits </parameter>
  <!--verbosity level of this processor ("DEBUG0-4,MESSAGE0-4,WARNING0-4,ERROR0-4,SILENT")-->
  <parameter name="Verbosity" type="string">DEBUG </parameter>
</processor>


<processor name="MergeCaloHits" type="MergeCollections">
  <!--MergeCollections creates a transient subset collection that merges all input collections -->
  <!--Names of all input collections-->
  <parameter name="InputCollections" type="StringVec">
    HcalBarrelCollectionDigi
    HcalEndcapRingCollectionDigi
```

List of variables available in the standard LCTuple

```
// Fixed size dimensions of array or collections stored in the TTree if any.

   // Declaration of leaf types
   Int_t           evevt;
   Int_t           evrun;
   Float_t         evwgt;
   Long64_t        evtim;
   Float_t         evsig;
   Float_t         evene;
   Float_t         evpoe;
   Float_t         evpop;
   Int_t           evnch;
   Char_t          evpro[1];   //[evnch]
   Int_t           nmcp;
   Int_t           mcori[807];   //[nmcp]
   Int_t           mcpdg[807];   //[nmcp]
   Int_t           mcgst[807];   //[nmcp]
   Int_t           mcsst[807];   //[nmcp]
   Float_t         mcvtx[807];   //[nmcp]
   Float_t         mcvty[807];   //[nmcp]
   Float_t         mcvtz[807];   //[nmcp]
   Float_t         mcepx[807];   //[nmcp]
   Float_t         mcepy[807];   //[nmcp]
   Float_t         mcepz[807];   //[nmcp]
   Float_t         mcmox[807];   //[nmcp]
   Float_t         mcmoy[807];   //[nmcp]
   Float_t         mcmoz[807];   //[nmcp]
   Float_t         mcmas[807];   //[nmcp]
   Float_t         mcene[807];   //[nmcp]
   Float_t         mccha[807];   //[nmcp]
   Float_t         mctim[807];   //[nmcp]
   Float_t         mcspx[807];   //[nmcp]
   Float_t         mcspy[807];   //[nmcp]
   Float_t         mcspz[807];   //[nmcp]
   Int_t           mccf0[807];   //[nmcp]
   Int_t           mccf1[807];   //[nmcp]
   Int_t           mcpa0[807];   //[nmcp]
   Int_t           mcpa1[807];   //[nmcp]
   Int_t           mcda0[807];   //[nmcp]
   Int_t           mcda1[807];   //[nmcp]
   Int_t           mcda2[807];   //[nmcp]
   Int_t           mcda3[807];   //[nmcp]
   Int_t           mcda4[807];   //[nmcp]
   Int_t           nrec;
   Int_t           rcori[41];   //[nrec]
   Int_t           rccid[41];   //[nrec]
   Int_t           rctyp[41];   //[nrec]
   Float_t         rccov[41][10];   //[nrec]
   Float_t         rcrpx[41];   //[nrec]
   Float_t         rcrpy[41];   //[nrec]
   Float_t         rcrpz[41];   //[nrec]
   Float_t         rcgpi[41];   //[nrec]
   Int_t           rcpiu[41];   //[nrec]
   Int_t           rcnpi[41];   //[nrec]
   Int_t           rcfpi[41];   //[nrec]
   Float_t         rcmox[41];   //[nrec]
   Float_t         rcmoy[41];   //[nrec]
   Float_t         rcmoz[41];   //[nrec]
   Float_t         rcmas[41];   //[nrec]
   Float_t         rcene[41];   //[nrec]
   Float_t         rccha[41];   //[nrec]
   Int_t           rcntr[41];   //[nrec]
   Int_t           rcncl[41];   //[nrec]
   Int_t           rcnrp[41];   //[nrec]
   Int_t           rcftr[41];   //[nrec]
```

```
   Int_t           rcftr[41];   //[nrec]
   Int_t           rcvts[41];   //[nrec]
   Int_t           rcvte[41];   //[nrec]
   Int_t           rccom[41];   //[nrec]
   Int_t           npid;
   Int_t           pityp[212];   //[npid]
   Int_t           pipdg[212];   //[npid]
   Float_t         pillh[212];   //[npid]
   Int_t           pialg[212];   //[npid]
   Int_t           ntrk;
   Int_t           trori[36];   //[ntrk]
   Int_t           trtyp[36];   //[ntrk]
   Float_t         trch2[36];   //[ntrk]
   Int_t           trndf[36];   //[ntrk]
   Float_t         tredx[36];   //[ntrk]
   Float_t         trede[36];   //[ntrk]
   Float_t         trrih[36];   //[ntrk]
   Int_t           trshn[36][12];   //[ntrk]
   Int_t           trnts[36];   //[ntrk]
   Int_t           trfts[36];   //[ntrk]
   Int_t           trsip[36];   //[ntrk]
   Int_t           trsfh[36];   //[ntrk]
   Int_t           trslh[36];   //[ntrk]
   Int_t           trsca[36];   //[ntrk]
   Int_t           ntrst;
   Int_t           tsloc[144];   //[ntrst]
   Float_t         tsdze[144];   //[ntrst]
   Float_t         tsphi[144];   //[ntrst]
   Float_t         tsome[144];   //[ntrst]
   Float_t         tszze[144];   //[ntrst]
   Float_t         tstnl[144];   //[ntrst]
   Float_t         tscov[144][15];   //[ntrst]
   Float_t         tsrpx[144];   //[ntrst]
   Float_t         tsrpy[144];   //[ntrst]
   Float_t         tsrpz[144];   //[ntrst]
   Int_t           nsth;
   Int_t           stori[1];   //[nsth]
   Int_t           stci0[1];   //[nsth]
   Int_t           stci1[1];   //[nsth]
   Double_t        stpox[1];   //[nsth]
   Double_t        stpoy[1];   //[nsth]
   Double_t        stpoz[1];   //[nsth]
   Float_t         stedp[1];   //[nsth]
   Float_t         sttim[1];   //[nsth]
   Float_t         stmox[1];   //[nsth]
   Float_t         stmoy[1];   //[nsth]
   Float_t         stmoz[1];   //[nsth]
   Float_t         stptl[1];   //[nsth]
   Int_t           stmcp[1];   //[nsth]
   Int_t           nsch;
   Int_t           scori[1];   //[nsch]
   Int_t           scci0[1];   //[nsch]
   Int_t           scci1[1];   //[nsch]
   Float_t         scpox[1];   //[nsch]
   Float_t         scpoy[1];   //[nsch]
   Float_t         scpoz[1];   //[nsch]
   Float_t         scene[1];   //[nsch]
   Int_t           r2mnrel;
   Int_t           r2mf[169];   //[r2mnrel]
   Int_t           r2mt[169];   //[r2mnrel]
   Float_t         r2mw[169];   //[r2mnrel]
```

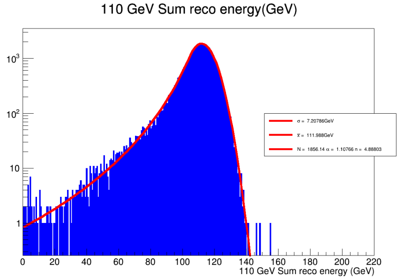
Previous results: links with all results, please explore yourself:

- first test production
  - `http://wwwae.ciemat.es/~carrillo/calice/indexk0o1.html`
  - `http://wwwae.ciemat.es/~carrillo/calice/indexk0o2.html`
- second test production
  - `http://wwwae.ciemat.es/~carrillo/calice/indexk0o1v2.html`
  - `http://wwwae.ciemat.es/~carrillo/calice/indexk0o2v2.html`

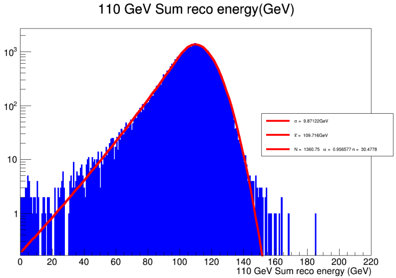
Comparison o1/o2 $\otimes$ $1^{st}/2^{nd}$ Test Production, K_L^0 110 GeV

2^{nd}TP

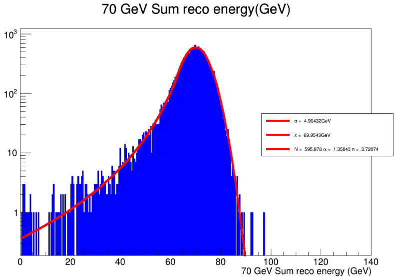
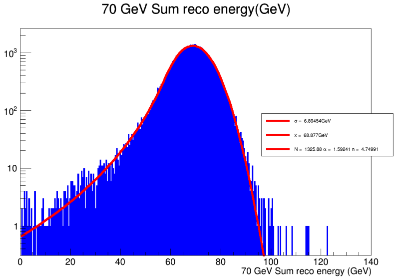
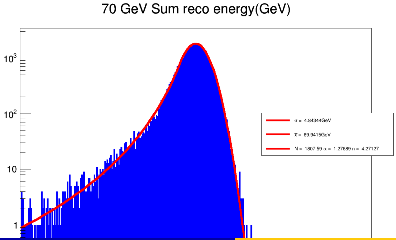
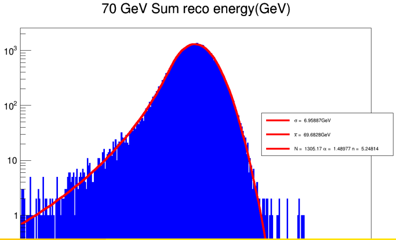
AHCAL(o1)



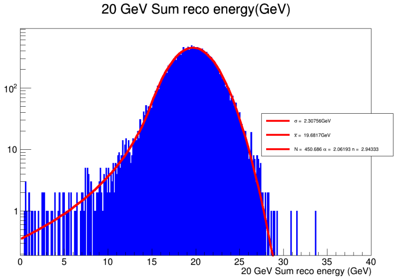
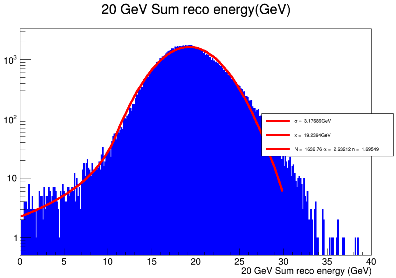
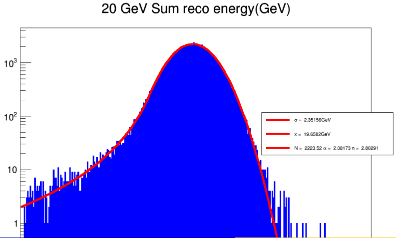
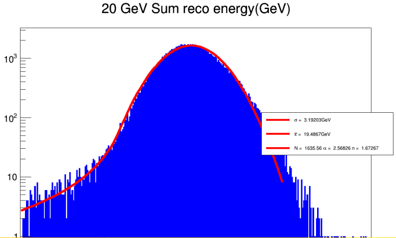
SDHCAL(o2)



Comparison o1/o2 $\otimes$ $1^{st}/2^{nd}$ Test Production, K_L^0 70 GeV

	AHCAL(o1)	SDHCAL(o2)
1^{st}TP		
2^{nd}TP		

Comparison o1/o2 $\otimes$ $1^{st}/2^{nd}$ Test Production, K_L^0 20 GeV

	AHCAL(o1)	SDHCAL(o2)
1^{st}TP		
2^{nd}TP		

Previous results: resolution and discrepancy for o1 and o2, fit results

1^{st}TP

sim p (GeV)	mean (GeV)	sigma (GeV)	**resolution (%)**	**discrepancy(%)**
		o1		
1	0.85	0.34	**39.6%**	**15.1%**
2	1.64	0.61	**37.2%**	**18.3%**
5	4.37	1.19	**27.3%**	**12.5%**
10	9.11	1.80	**19.7%**	**8.9%**
20	19.68	2.31	**11.7%**	**1.6%**
30	29.75	2.91	**9.8%**	**0.8%**
40	39.75	3.39	**8.5%**	**0.6%**
50	49.50	3.94	**7.9%**	**1.0%**
70	69.95	4.90	**7.0%**	**0.1%**
		o2		
1	0.79	0.31	**38.6%**	**20.8%**
2	1.48	0.56	**38.2%**	**26.2%**
5	3.86	1.14	**29.6%**	**22.9%**
10	8.28	1.88	**22.7%**	**17.2%**
20	19.24	3.18	**16.5%**	**3.8%**
30	29.51	4.11	**13.9%**	**1.6%**
40	39.27	4.85	**12.4%**	**1.8%**
60	58.95	6.27	**10.6%**	**1.8%**
70	68.88	6.90	**10.0%**	**1.6%**
80	78.77	7.62	**9.7%**	**1.5%**
90	88.45	8.40	**9.5%**	**1.7%**
100	98.50	8.91	**9.0%**	**1.5%**

$$\text{resolution} = \frac{sigma}{mean}, \text{ discrepancy} = \frac{sim\ p - mean}{sim\ p}$$

2^{nd}TP

sim p (GeV)	mean (GeV)	sigma (GeV)	**resolution (%)**	**discrepancy(%)**
		o1		
1	0.66	0.23	**34.9%**	**34.0%**
2	1.63	0.62	**37.9%**	**18.5%**
5	4.37	1.19	**27.4%**	**12.7%**
10	9.12	1.79	**19.6%**	**8.9%**
20	19.66	2.35	**12.0%**	**1.7%**
30	29.73	2.90	**9.7%**	**0.9%**
40	39.76	3.37	**8.5%**	**0.6%**
50	49.71	3.90	**7.8%**	**0.6%**
60	59.82	4.33	**8.7%**	**17.2%**
70	69.94	4.84	**8.1%**	**14.5%**
80	80.13	5.46	**7.8%**	**12.6%**
90	90.63	6.02	**7.5%**	**11.0%**
100	101.20	6.71	**7.4%**	**9.4%**
110	112.00	7.21	**7.1%**	**8.0%**
		o2		
1	0.81	0.31	**38.4%**	**19.1%**
2	1.51	0.56	**37.2%**	**24.5%**
5	3.92	1.16	**29.6%**	**21.7%**
10	8.40	1.90	**22.6%**	**16.0%**
20	19.49	3.19	**16.4%**	**2.6%**
30	29.86	4.09	**13.7%**	**0.5%**
40	39.74	4.80	**12.1%**	**0.6%**
50	49.64	5.56	**11.2%**	**0.7%**
60	59.63	6.31	**10.6%**	**0.6%**
70	69.68	6.96	**10.0%**	**0.5%**
80	79.63	7.70	**9.7%**	**0.5%**
90	89.66	8.40	**9.4%**	**0.4%**

Previous results: Resolution for the four scenarions: o1/o2 $\otimes$ $1^{st}/2^{nd}$ TP