Dielectron analysis: Lost electrons

Sudhir Pandurang Rode

LHEP, JINR

April 16, 2024

MPD Cross-PWG Meeting

Quick recap

- With the help of low p_{T} electron tracks, in dielectron analysis, signficant amount of improvement in S/B is foreseen.
- As discussed in the last presentations, part of it is can seen already $\rightarrow \approx 2$ factor.
- However, still there is room for more rejection of combinatorial background(CB).

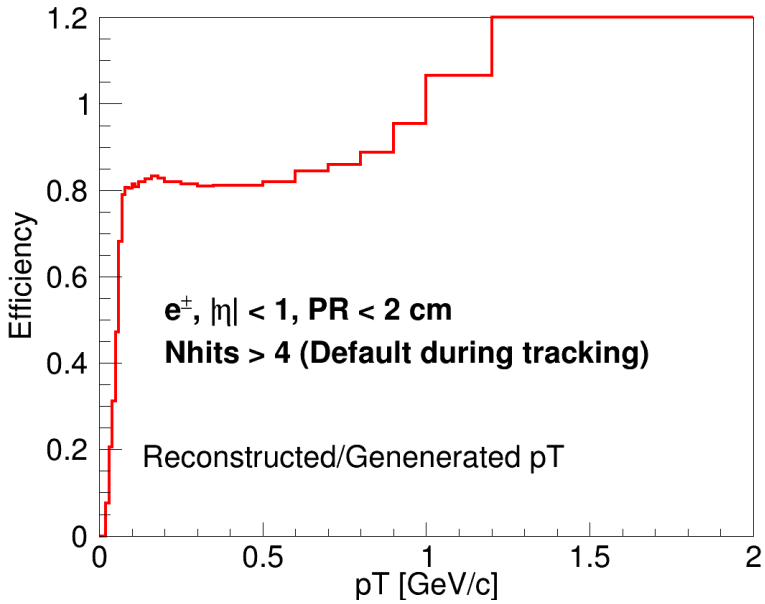
Source of remaining CB after close TPC cut[1]

Total reconstructed tracks after close TPC cut:	**1.69268e+06**
Below: Only Conversion and π^0 Dalitz sources are considered --	
a. Track has Partner with pT < 35 MeV ($\|\eta\|$ < 2.5):	**419595 (~25%)**
b. Track has Partner inside TPC i.e. 35 < pT < 100 MeV ($\|\eta\|$ < 2.5):	**580428 (~34%)**
c. Track has Partner with pT > 110 MeV ($\|\eta\|$ < 2.5):	**266075 (~16%)**
Track is hadron:	**102041 (~6%)**
Rest (Signal (η, etc), conversion, π^0 Dalitz whose partner outside TPC, ...)	**324536 (~19%)**

- Category b and c indicate substantial amount of partner electrons are not reconstructed.
- However, not all inefficiency is due to reconstruction algorithm.
- Partial effect is coming from simulations, where some amount of electrons do not leave MC response in the TPC.
- So no MC point in the TPC $\rightarrow$ no track reconstruction.

[1] These numbers/analysis are using MPDROOT version: request 25 version of MPDROOT (commit b95c9cb8 on `https://git.jinr.ru/nica/mpdroot/-/commits/massprod`

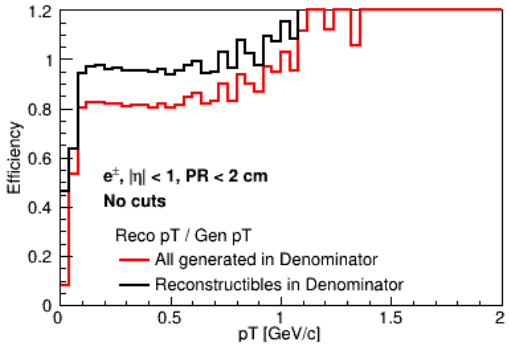
Track reconstruction efficiency[2]

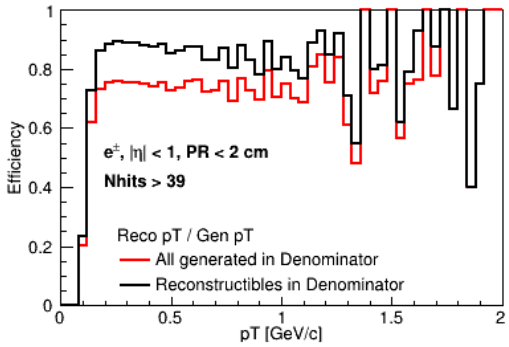


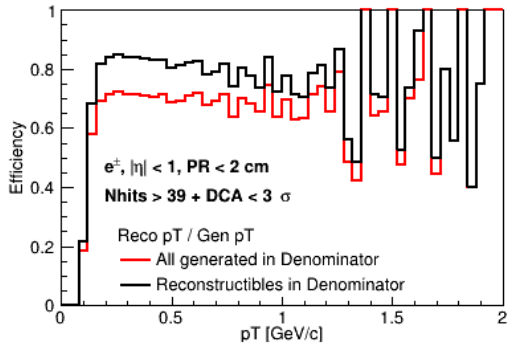
- This ratio of reconstructed electron tracks with Nhits in the TPC > 4 to the all reconstructible tracks (has MC points in the TPC) should be close to 1.
- It is not because some electrons are "lost" as they do not leave any MC points in the TPC $\rightarrow$ not "reconstructible".
- Effect propagates through different sleection cuts and gives significantly less efficiency than what we should achieve.

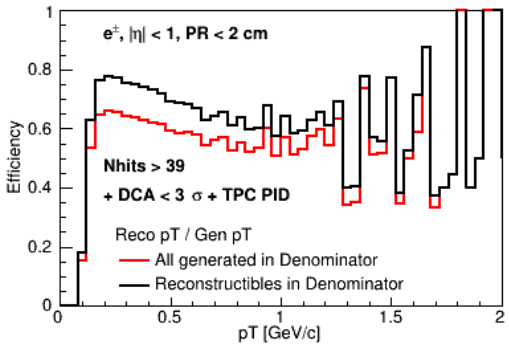
[2]These numbers/analysis are using MPDROOT version: request 25 version of MPDROOT (commit b95c9cb8 on `https://git.jinr.ru/nica/mpdroot/-/commits/massprod`

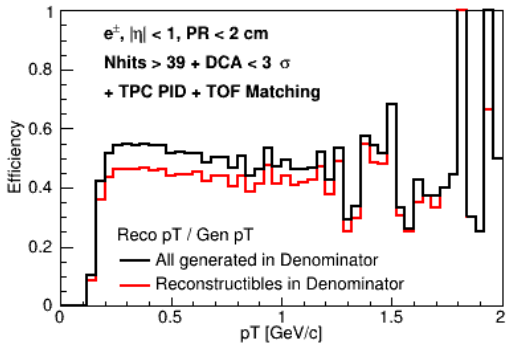
Step by step demonstration: reconstruction efficiency

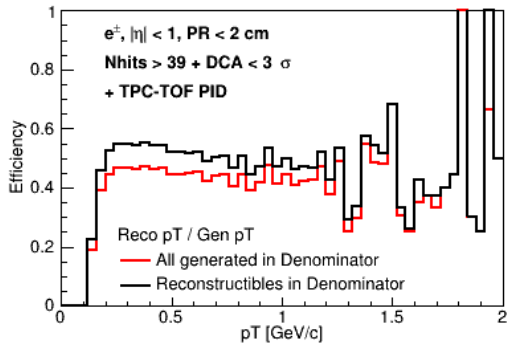


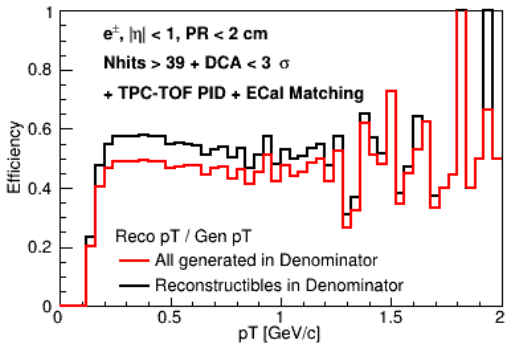


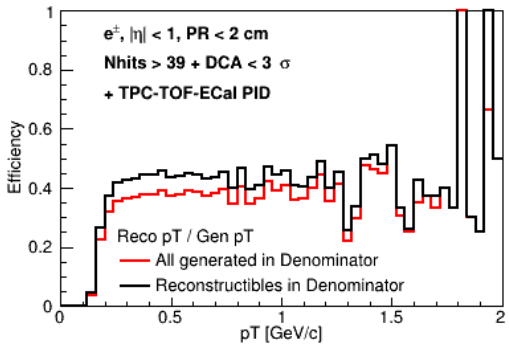






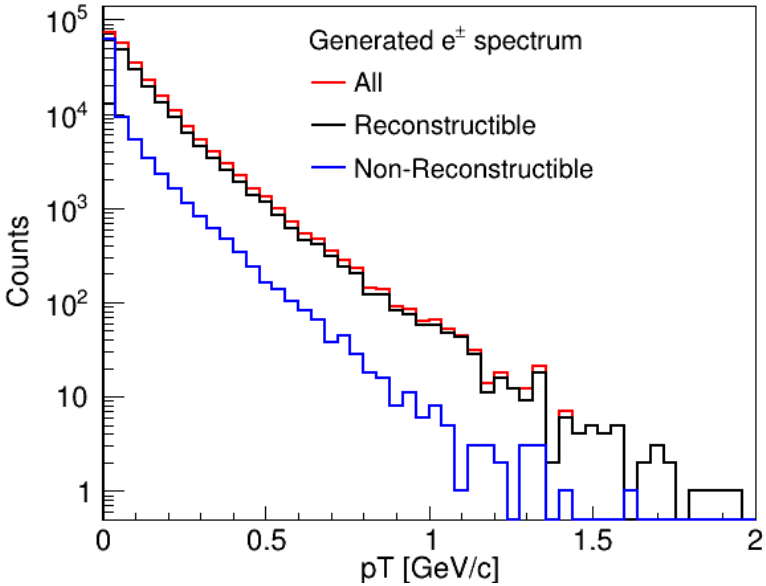


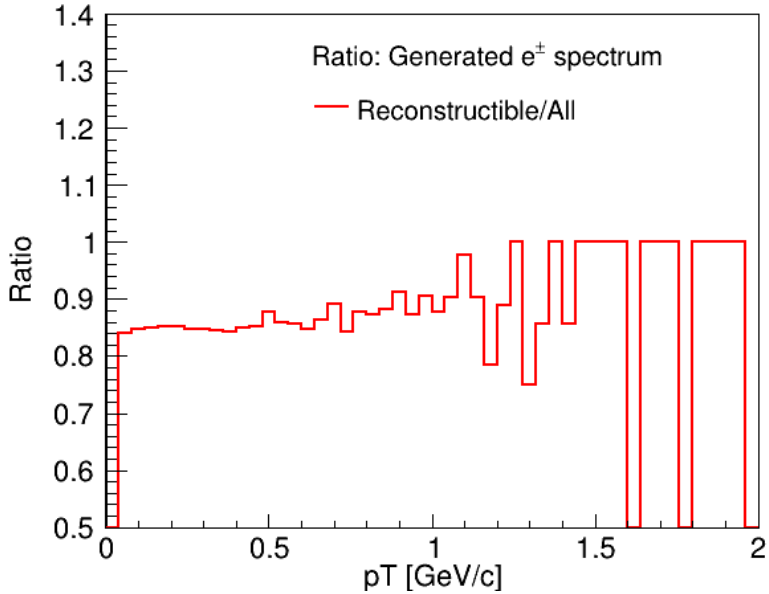




- Request 25.
- All generated electrons in the denominator.
- Only electrons having MC points in the TPC in the denominator.

Primary $e^{\pm}$ within $|\eta| < 1.0$: Lost electrons $\rightarrow$ Private production[3]

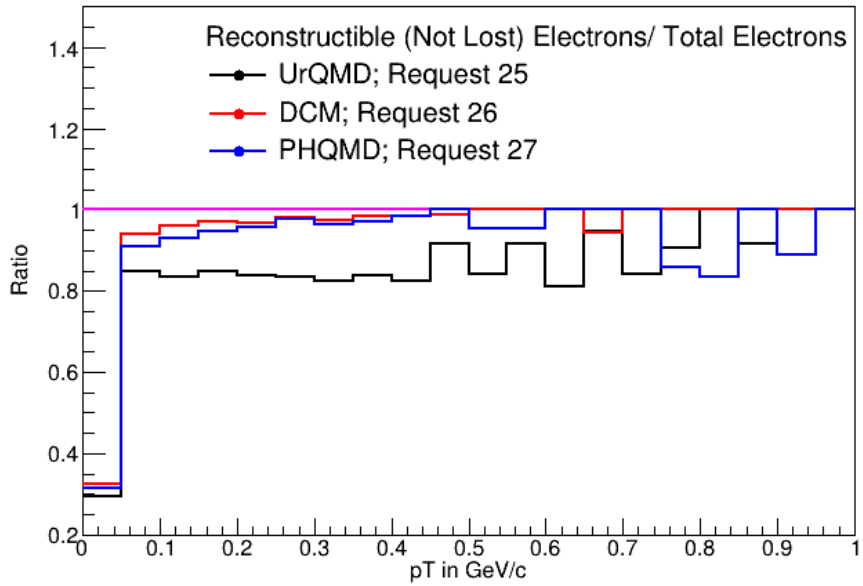




- As mentioned, the effect is from generated level.
- ≈15% of electrons do not leave MC points in the TPC hence not reconstruction of those electrons.

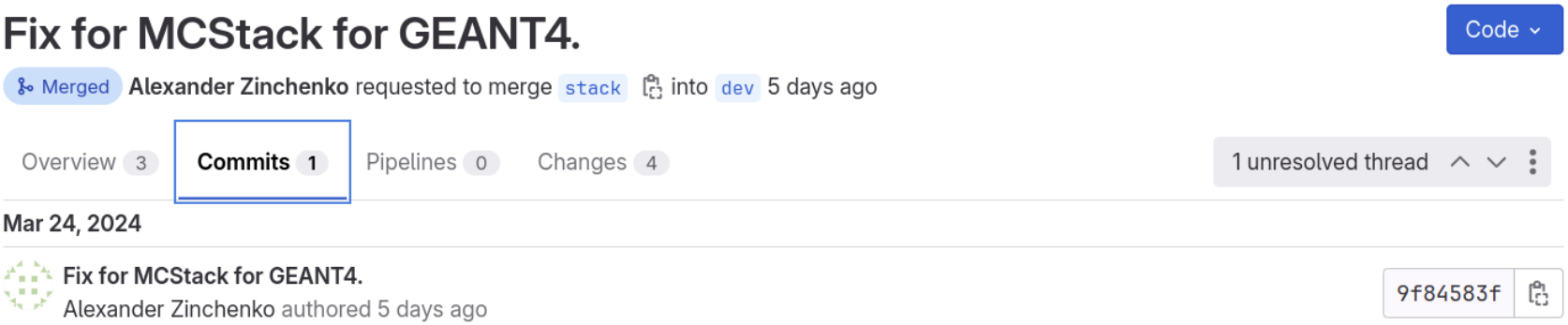
[3] These numbers/analysis are using MPDROOT version: request 25 version of MPDROOT (commit b95c9cb8 on `https://git.jinr.ru/nica/mpdroot/-/commits/massprod`

Primary $e^{\pm}$ within $|\eta| < 1.0$: Lost electrons $\rightarrow$ Official productions



- This problem is not observed in other productions, only in Request 25.
- In request 25, we use external pythia8 decayer.
- After reporting the problem, Alexander Zinchenko has fixed the issue in the MC track GEANT4 settings $\rightarrow$ next slides.

Problems with MCStack



- The main problem was that with GEANT4 previous input settings the MCStack was not used for handling decay products, while it was used to put secondary particles in Pythia decayer.
- This may have affected the cascade decays, i.e., for example, π^0-mesons from omegas. If the input setting "stackPopper" in g4Config.C is added, the stack starts to be used.
- In addition, due to usage of some internal variable to pass some information (which was overwritten by GEANT4), the particle with some index in the event (number 11) was lost.

Problems with MCStack

Fix for MCStack for GEANT4.

Code

Merged **Alexander Zinchenko** requested to merge `stack` into `dev` 5 days ago

Overview 3 | **Commits 1** | Pipelines 0 | Changes 4 | 1 unresolved thread

Mar 24, 2024

Fix for MCStack for GEANT4.
Alexander Zinchenko authored 5 days ago

9f84583f

gconfig/MpdDecayConfig.txt +1 −0 View file @ 9f84583f

```
@@ -6,6 +6,7 @@
 # Spaces are channel separators.
 # ":*" means inclusive decay modes, i.e. affected only the channels
 # explicitly mentioned
+# If mother particle differs from anti-particle, the particle should be used
 #113:* 113:11:-11:x1
 #223:* 223:11:-11:x1 223:111:11:-11:x1
 #333:* 333:11:-11:x1 333:221:11:-11:x1
```

gconfig/g4Config.C 100755 → 100644 +2 −1 View file @ 9f84583f

```
@@ -28,7 +28,8 @@ void Config()
    //TG4RunConfiguration* runConfiguration
            //= new TG4RunConfiguration("geomRoot", "FTFP_BERT", "stepLimiter+specialCuts");
    TG4RunConfiguration* runConfiguration
-           = new TG4RunConfiguration("geomRoot", "FTFP_BERT+optical");
+      //AZ-240324           = new TG4RunConfiguration("geomRoot", "FTFP_BERT+optical");
+      = new TG4RunConfiguration("geomRoot", "FTFP_BERT+optical", "stepLimiter+stackPopper"); //AZ-240324

 /// Create the G4 VMC
    TGeant4* geant4 = new TGeant4("TGeant4", "The Geant4 Monte Carlo", runConfiguration);
```

Problems with MCStack

simulation/generators/mpdGen/MpdDecayerPyt8.cxx +2 −1 View file @ 9f84583f

```
@@ -88,7 +88,8 @@ void MpdDecayerPyt8::Init()
88  88      fBranch                = channel.bRatio();
89  89
90  90      // Random number generator
91      -   fRandom = new TRandom(0); // time-dependent seed
    91  +   // AZ-220324 fRandom = new TRandom(0); // time-dependent seed
    92  +   fRandom = new TRandom(1); // AZ-220324 time-independent seed
92  93      // fRandom = gRandom;
93  94
94  95      // fPythia8->Pythia8()->particleData.isResonance(113,kTRUE);
```

simulation/mcStack/MpdStack.cxx +23 −17 View file @ 9f84583f

```
@@ -133,31 +133,32 @@ void MpdStack::PushTrack(Int_t toBeDone, Int_t parentId, Int_t pdgCode, Double_t
133 133
134 134       for (Int_t j = 0; j < npart; ++j) {
135 135          TParticle tpart = *((TParticle *)fDecays->UncheckedAt(j));
136     -        // tpart.Print();
    136 +        // tpart.Print(); //AZ-210324
137 137          vDecays.push_back(tpart);
138 138       }
139 139
140 140       for (Int_t j = 0; j < npart; ++j) { // skip mother particle
141 141          if (j == 0) {
142     -           particle->SetStatusCode(11); // decayed particle
143     -           continue;                    // skip mother particle
    142 +           // AZ-200324 particle->SetStatusCode(11); // decayed particle
    143 +           particle->SetDaughter(0, 11); // AZ-200324 decayed particle
    144 +           continue;                    // skip mother particle
144 145          }
145 146          // TParticle *part = (TParticle*) fDecays->UncheckedAt(j);
146 147          TParticle *part = &vDecays[j];
147 148          Int_t      toDo = 1;
148     -        if (TString(gMC->GetName()) == "TGeant4") toDo = 0;
149     -        // PushTrack(toBeDone, trackId, part->GetPdgCode(),
    149 +        // AZ-210324 if (TString(gMC->GetName()) == "TGeant4") toDo = 0;
    150 +        // PushTrack(toBeDone, trackId, part->GetPdgCode(),
150 151          PushTrack(toDo, trackId, part->GetPdgCode(), part->Px(), part->Py(), part->Pz(), part->Energy(), part->Vx(),
151 152                    part->Vy(), part->Vz(), time, polx, poly, polz, proc, ntr, weight, is);
152 153       }
153     -   if (parentId < 0) fNPrimaries += (npart - 1); // treat decay products as primaries (dirty trick)
    154 +   // AZ-220324 if (parentId < 0) fNPrimaries += (npart - 1); // treat decay products as primaries (dirty trick)
154 155    } else {
155 156       // --> Push particle on the stack if toBeDone is set
156 157       if (toBeDone == 1) fStack.push(particle);
157 158    }
158 159
159 160    // --> Push particle on the stack if toBeDone is set
160     -  // AZ if (toBeDone == 1) fStack.push(particle);
    161 +  // AZ if (toBeDone == 1) fStack.push(particle); //
161 162 }
162 163
```

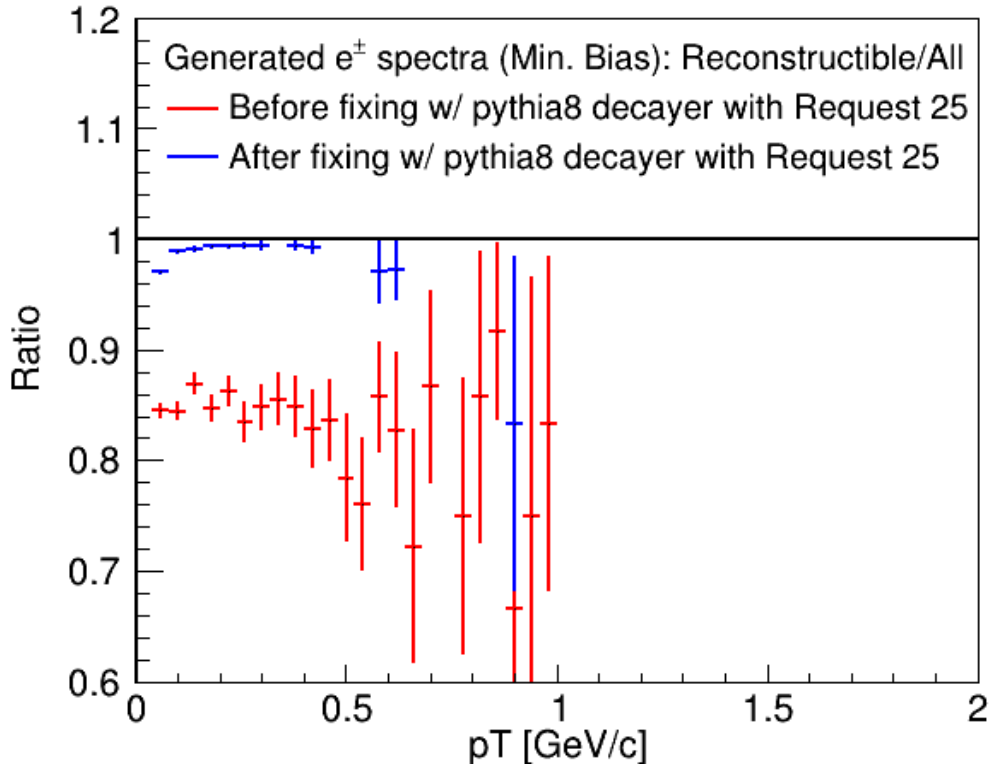
simulation/mcStack/MpdStack.cxx +23 −17 View file @ 9f84583f

```
@@ -226,18 +227,19 @@ void MpdStack::PushTrack(Int_t toBeDone, Int_t parentId, Int_t pdgCode, Double_t
226 227
227 228       for (Int_t j = 0; j < npart; ++j) {
228 229          if (j == 0) {
229     -           particle->SetStatusCode(11); // decayed particle
230     -           continue;                    // skip mother particle
    230 +           // AZ-200324 particle->SetStatusCode(11); // decayed particle
    231 +           particle->SetDaughter(0, 11); // AZ-200324 decayed particle
    232 +           continue;                    // skip mother particle
231 233          }
232 234          // TParticle *part = (TParticle*) fDecays->UncheckedAt(j);
233 235          TParticle *part = &vDecays[j];
234 236          Int_t      toDo = 1;
235     -        if (TString(gMC->GetName()) == "TGeant4") toDo = 0;
236     -        // PushTrack(toBeDone, trackId, part->GetPdgCode(),
    237 +        // AZ-210324 if (TString(gMC->GetName()) == "TGeant4") toDo = 0;
    238 +        // PushTrack(toBeDone, trackId, part->GetPdgCode(),
237 239          PushTrack(toDo, trackId, part->GetPdgCode(), part->Px(), part->Py(), part->Pz(), part->Energy(), part->Vx(),
238 240                    part->Vy(), part->Vz(), time, polx, poly, polz, proc, ntr, weight, is);
239 241       }
240     -   if (parentId < 0) fNPrimaries += (npart - 1); // treat decay products as primaries (dirty trick)
    242 +   // AZ-220324 if (parentId < 0) fNPrimaries += (npart - 1); // treat decay products as primaries (dirty trick)
241 243    } else {
242 244       // --> Push particle on the stack if toBeDone is set
243 245       if (toBeDone == 1) fStack.push(particle);
@@ -270,6 +272,7 @@ TParticle *MpdStack::PopNextTrack(Int_t &iTrack)
270 272    fCurrentTrack = thisParticle->GetStatusCode();
271 273    iTrack        = fCurrentTrack;
272 274
    275 +  if (thisParticle->GetDaughter(0) == 11) return (TParticle *)NULL; // AZ-220324
273 276    return thisParticle;
274 277 }
275 278 // -----------------------------------------------------------------------
@@ -298,7 +301,8 @@ TParticle *MpdStack::PopPrimaryForTracking(Int_t iPrim)
298 301       Fatal("MpdStack::PopPrimaryForTracking", "Not a primary track");
299 302    }
```

Details

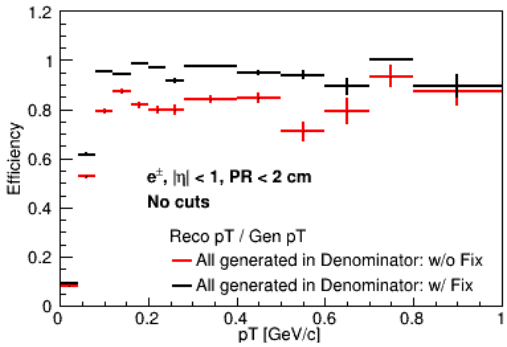
- For this study, the files in the commit shown in previous slides were added to the request 25 version of MPDROOT (commit b95c9cb8 on `https://git.jinr.ru/nica/mpdroot/-/commits/massprod`).
- And this version is used to get these results.
- I have also updated the beam pipe geometry (air $\rightarrow$ vacuum).
- Before fix: 6242 events.
- After fix: 7649 events.

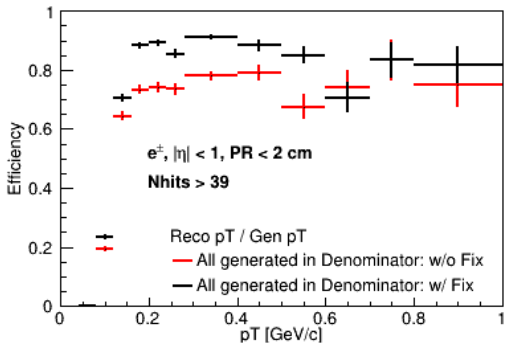
Primary $e^{\pm}$ within $|\eta| < 1.0$ - $\approx$ 6-8K Min. Bias UrQMD BiBi events

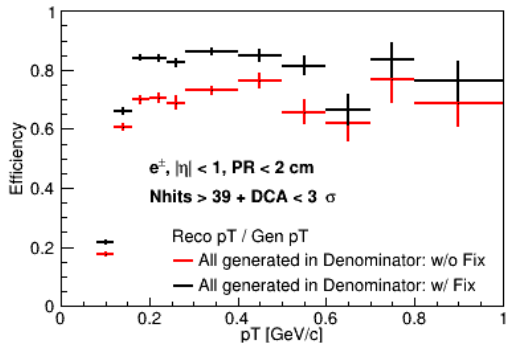


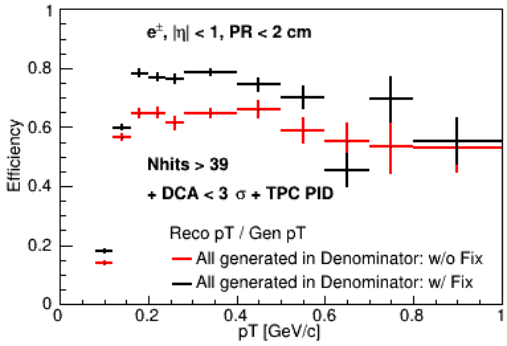
- With new updates in the MCStack and GEANT4 settings, new issue of lost electrons due to external pythia decayer seem to have vanished.
- Before and after fix scenario - MPDROOT version (one used for Request 25).
- Results with latest versions also show similar improvement (see the back up).

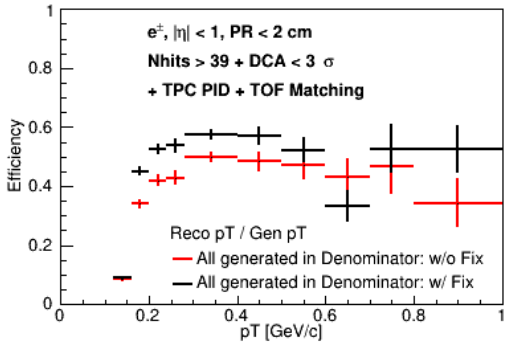
Primary $e^{\pm}$ within $|\eta| < 1.0$ - $\approx$ 6-8K Min. Bias UrQMD BiBi events

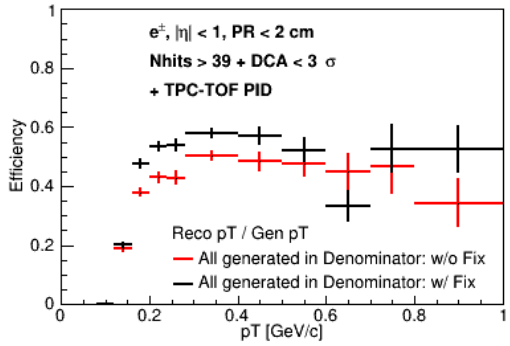


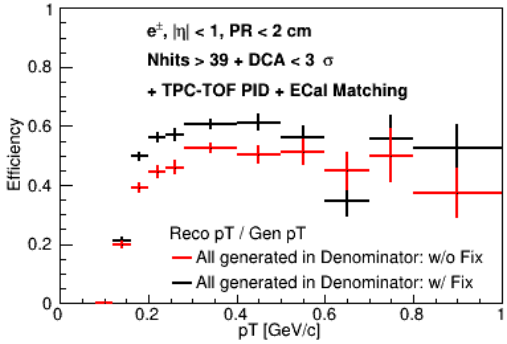


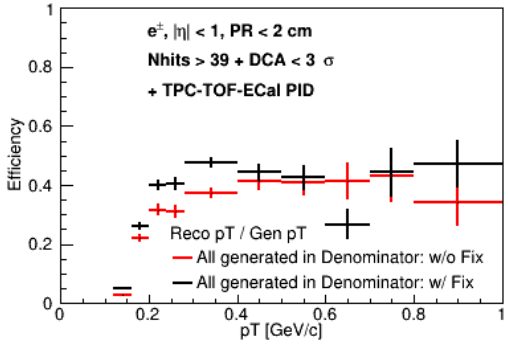












- This provides improvement in the single electron efficiency.
- This will give big boost to the di-electron analysis $\rightarrow$ CB rejection.
- Therefore, we would like to request for a new production for dielectrons.

Improvement in the reconstruction efficiency

- For $0.2 < p_{\mathrm{T}} < 2$ GeV/c.

Cuts	Efficiency±Error	Improvement±Error
Nhits > 4	0.820±0.008	
-	0.959±0.004	1.17±0.01
Nhits > 39	0.755±0.009	
-	0.882±0.006	1.17±0.01
+ DCA cut	0.712±0.010	
-	0.837±0.007	1.18±0.02
+ TPC PID	0.633±0.010	
-	0.753±0.009	1.19±0.02
+ TOF Matching	0.456±0.011	
-	0.541±0.010	1.19±0.03
+ TOF PID	0.462±0.011	
-	0.547±0.010	1.18±0.03
+ ECAL Matching	0.484±0.011	
-	0.576±0.010	1.19±0.03
+ ECAL PID	0.355±0.010	
-	0.427±0.010	1.20±0.04

Conclusions

- The issue of lost electrons in the TPC during Geant transport seem to have been fixed.
- The effect of this on the electron reconstruction and PID efficiency ($\approx$ 20%) is seen (Request 25 MPDROOT version).
- Similar effect is also seen with latest versions of MPDROOT[4]
- This is expected to have an effect on the dielectron analysis and would help in rejecting the CB as well as improve the signal.
- If the agreement is reached, a new production for dielectrons with this fix is requested.

[4] the results are in the back-up, however, there is some inconsistency related to the conversions which need to be cross-checked.

BACK-UP

Effect on multiplicities of electron sources

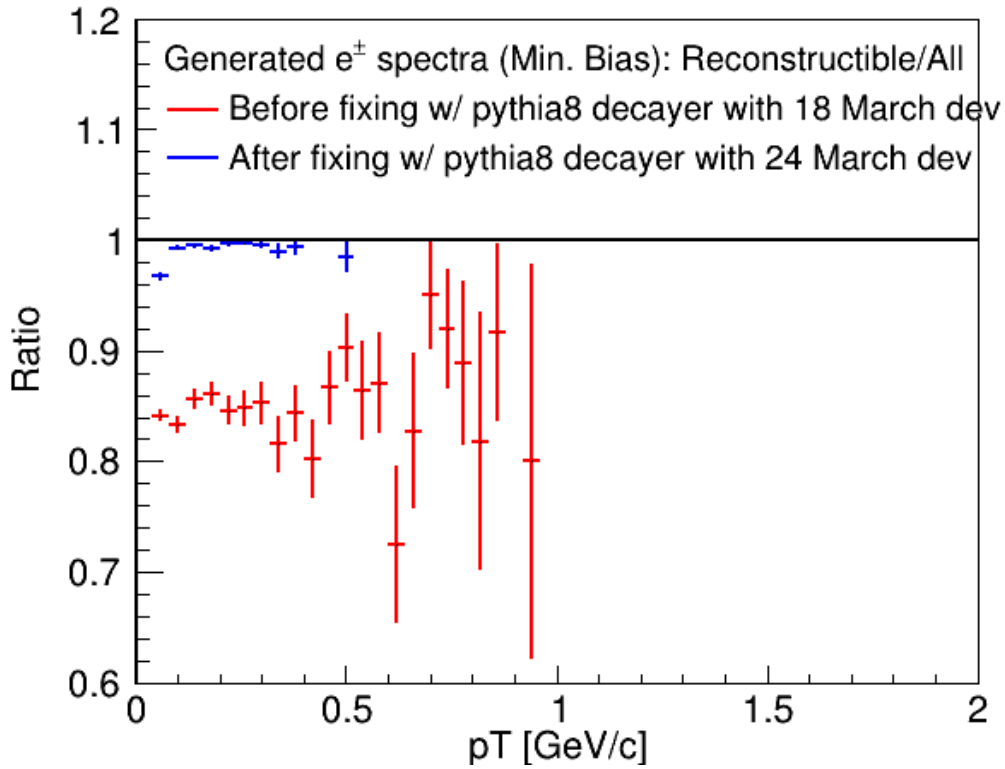
- Average multiplicities for $p_T > 200$ MeV/c per 100 events ($|\eta| < 1.0$ and produced within 2 cm)

	Cuts	Average Multiplicity±Error	Improvement±Error
Before Fix	π^0-Dalitz	10.97±0.42	
After Fix	π^0-Dalitz	12.89±0.50	1.17±0.06
Before Fix	η-Dalitz	0.95±0.12	
After Fix	η-Dalitz	1.15±0.15	1.22±0.22
Before Fix	ρ^0	0.11±0.04	
After Fix	ρ^0	0.09±0.04	0.82±0.47
Before Fix	ω	0.24±0.06	
After Fix	ω	0.10±0.05	0.44±0.25

Details

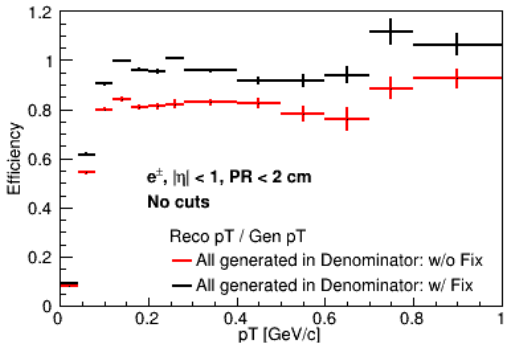
- For this study, the files in the commit shown in these slides were added to the March 24, 2024 version of MPDROOT (commit 9f84583f on `https://git.jinr.ru/nica/mpdroot/-/commit/9f84583fe2c2544d3bcad1739bf0fbf6104e5dc9`).
- And this version is used to get these results.
- I have also updated the beam pipe geometry (air $\rightarrow$ vacuum).
- For before fix scenario: March 18, 2024 version of MPDROOT (commit aa3dfb40 on `https://git.jinr.ru/nica/mpdroot/-/commit/aa3dfb40011f813366964321eb8be754cb06621a`).
- Before fix: 6264 events.
- After fix: 7715 events.

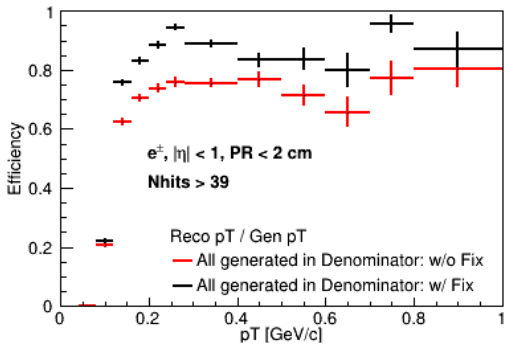
Primary $e^{\pm}$ within $|\eta| < 1.0$ - $\approx$ 6-8K Min. Bias UrQMD BiBi events

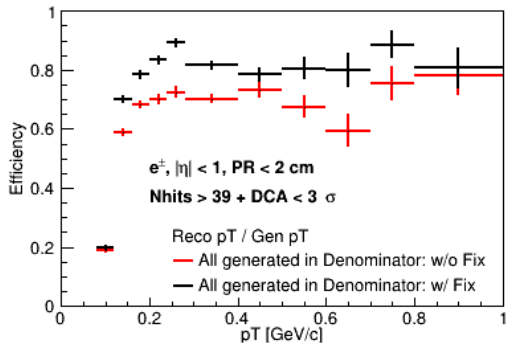


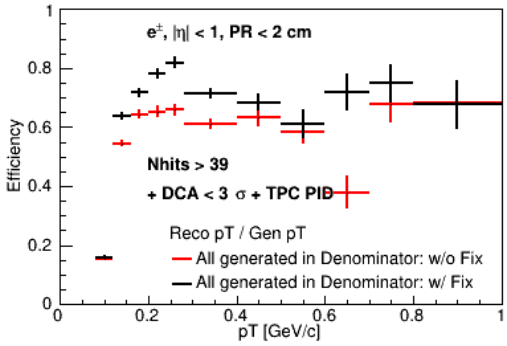
- With new updates in the MCStack and GEANT4 settings, new issue of lost electrons due to external pythia decayer seem to have vanished.

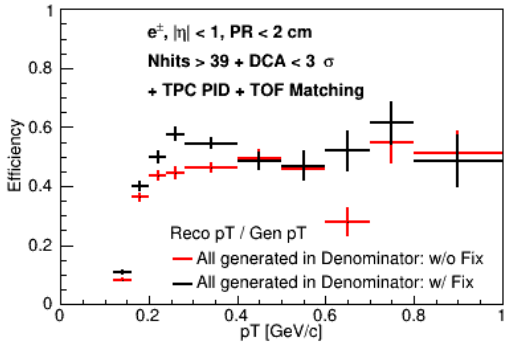
Primary $e^{\pm}$ within $|\eta| < 1.0$ - $\approx$ 6-8K Min. Bias UrQMD BiBi events

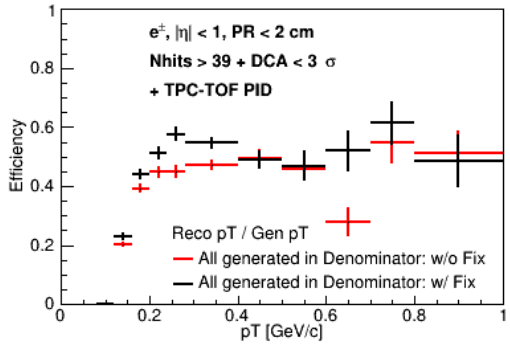


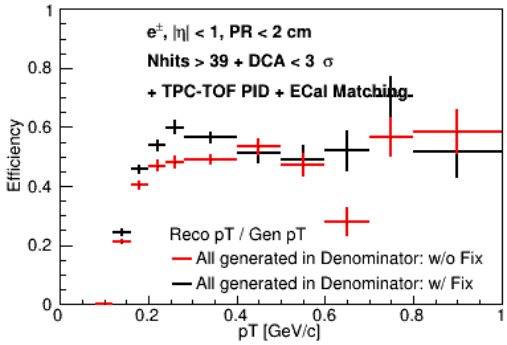


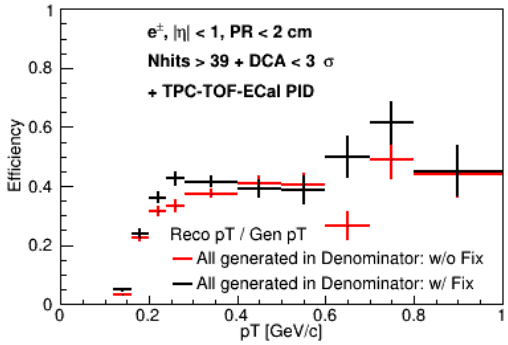












- This provides improvement in the single electron efficiency.
- This will give big boost to the di-electron analysis $\rightarrow$ CB rejection.
- Therefore, we would like to request for a new production for dielectrons.

Improvement in the reconstruction efficiency

- For $0.2 < p_{\mathrm{T}} < 2$ GeV/c.

Cuts	Efficiency±Error	Improvement±Error
Nhits $>$ 4	0.824± 0.008	
-	0.967± 0.004	1.17± 0.01
Nhits $>$ 39	0.749± 0.009	
-	0.889± 0.007	1.19 ±0.01
+ DCA cut	0.708± 0.009	
-	0.833± 0.008	1.18± 0.02
+ TPC PID	0.629± 0.009	
-	0.743± 0.010	1.18 ±0.02
+ TOF Matching	0.454± 0.010	
-	0.528± 0.011	1.16 ±0.03
+ TOF PID	0.460± 0.010	
-	0.533± 0.011	1.16± 0.03
+ ECAL Matching	0.483± 0.010	
-	0.555± 0.011	1.15 ±0.03
+ ECAL PID	0.358± 0.010	
-	0.406± 0.011	1.14± 0.04

Effect on multiplicities of electron sources

- Average multiplicities for $p_T > 200$ MeV/c per 100 events ($|\eta| < 1.0$ and produced within 2 cm)

	Cuts	Average Multiplicity±Error	Improvement±Error
Before Fix	π^0-Dalitz	11.12±0.38	
After Fix	π^0-Dalitz	13.60±0.47	1.22±0.06
Before Fix	η-Dalitz	1.28±0.13	
After Fix	η-Dalitz	1.15±0.14	0.90±0.14
Before Fix	ρ^0	0.03±0.02	
After Fix	ρ^0	0.1±0.04	3.33±2.59
Before Fix	ω	0.23±0.06	
After Fix	ω	0.14±0.05	0.61±0.27
Before Fix	γ	2.58±0.18	
After Fix	γ	2.17±0.19	0.84±0.1

Contribution from conversions before beam pipe despite using updated geometry $\rightarrow$ needs cross-check.