Vertex Reconstruction Progress Report

Ziyuan LI

SUN-YAT-SEN UNIVERSITY
zli@cern.ch

JUNO Software Workshop @ WHU

May 11, 2018

Overview

Thanks for useful guide and discussion from Tao.LIN, Qin.LIU, Yan.ZHANG, Wenjie.WU, etc.

Data Production

refer to scripts here :
/junofs/production/data-production/Releases/J16v1r4/Positron/uniform

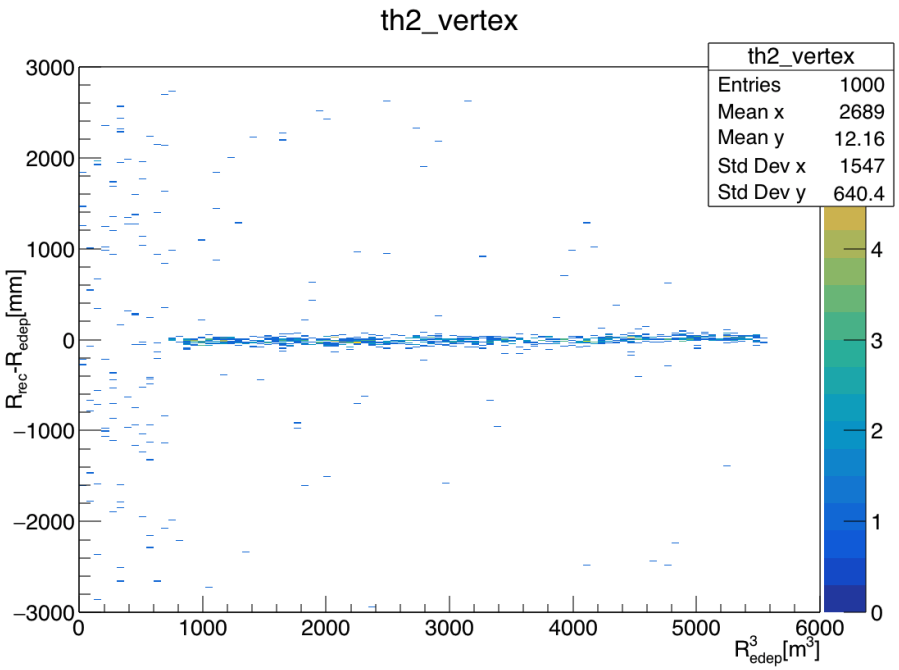
modify to latest version :
/junofs/users/lizy/juno/data/J17v1r1/job
sim_run.sh && cal_run.sh && rec_run.sh

submit job to IHEP Condor and get data samples : 1000 events in each root file and 100 root files for P $= \{0..6\}$ MeV.

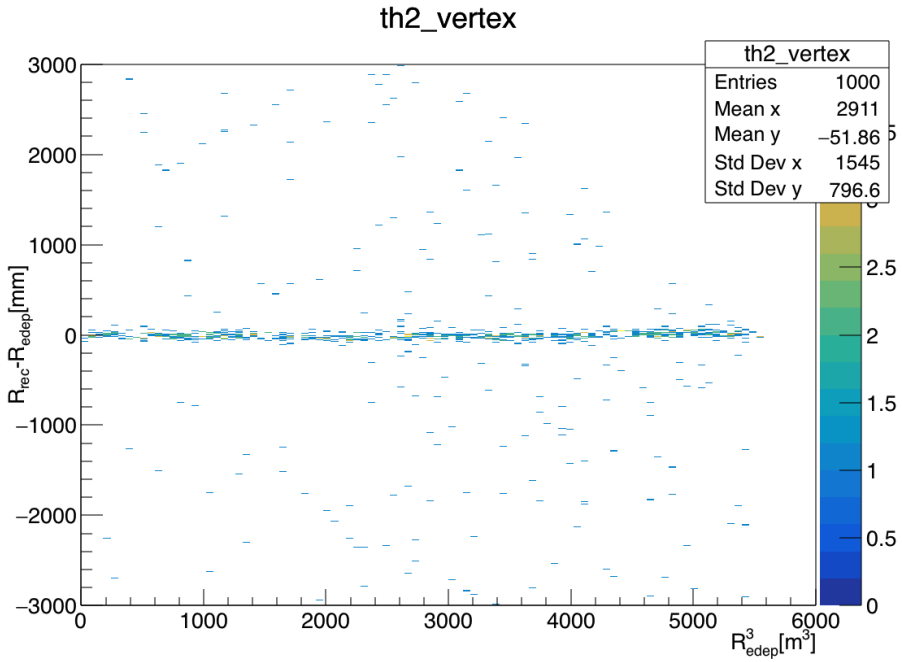
```
[lizy@lxslc601 e+_1.022MeV]$ ls sim-*.root
sim-1000.root  sim-1015.root  sim-1030.root  sim-1045.root  sim-1060.root  sim-1075.root  sim-1090.root
sim-1001.root  sim-1016.root  sim-1031.root  sim-1046.root  sim-1061.root  sim-1076.root  sim-1091.root
sim-1002.root  sim-1017.root  sim-1032.root  sim-1047.root  sim-1062.root  sim-1077.root  sim-1092.root
sim-1003.root  sim-1018.root  sim-1033.root  sim-1048.root  sim-1063.root  sim-1078.root  sim-1093.root
sim-1004.root  sim-1019.root  sim-1034.root  sim-1049.root  sim-1064.root  sim-1079.root  sim-1094.root
sim-1005.root  sim-1020.root  sim-1035.root  sim-1050.root  sim-1065.root  sim-1080.root  sim-1095.root
sim-1006.root  sim-1021.root  sim-1036.root  sim-1051.root  sim-1066.root  sim-1081.root  sim-1096.root
sim-1007.root  sim-1022.root  sim-1037.root  sim-1052.root  sim-1067.root  sim-1082.root  sim-1097.root
sim-1008.root  sim-1023.root  sim-1038.root  sim-1053.root  sim-1068.root  sim-1083.root  sim-1098.root
sim-1009.root  sim-1024.root  sim-1039.root  sim-1054.root  sim-1069.root  sim-1084.root  sim-1099.root
sim-1010.root  sim-1025.root  sim-1040.root  sim-1055.root  sim-1070.root  sim-1085.root
sim-1011.root  sim-1026.root  sim-1041.root  sim-1056.root  sim-1071.root  sim-1086.root
sim-1012.root  sim-1027.root  sim-1042.root  sim-1057.root  sim-1072.root  sim-1087.root
sim-1013.root  sim-1028.root  sim-1043.root  sim-1058.root  sim-1073.root  sim-1088.root
sim-1014.root  sim-1029.root  sim-1044.root  sim-1059.root  sim-1074.root  sim-1089.root
```

Potential Bug

Distribution from 0 - 800 is not expected.



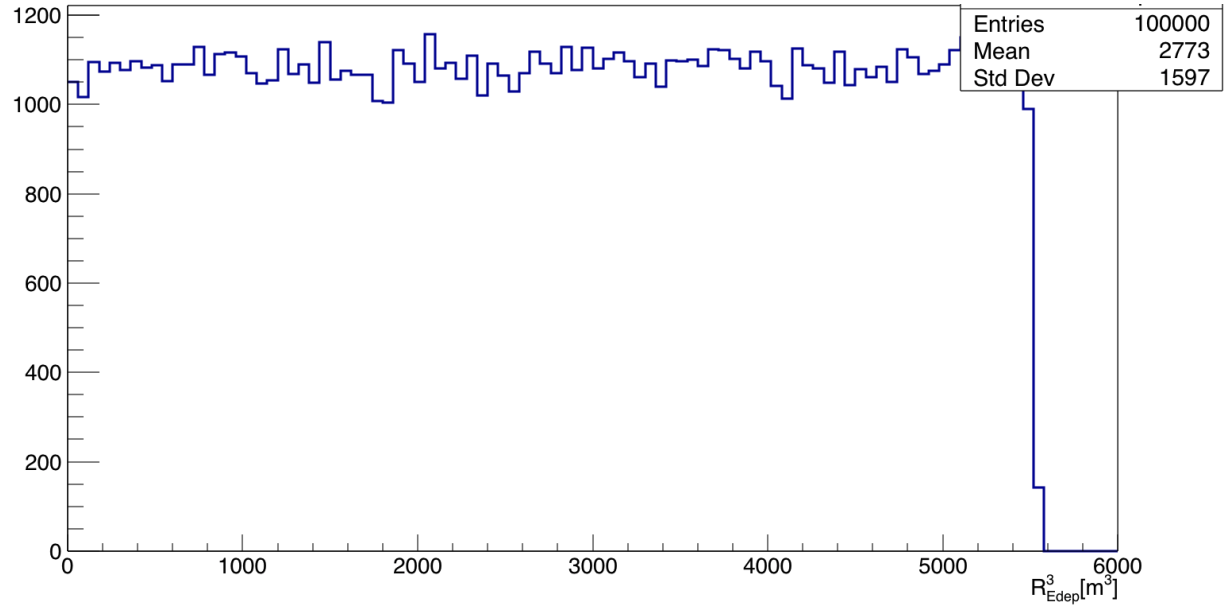
J17v1r1



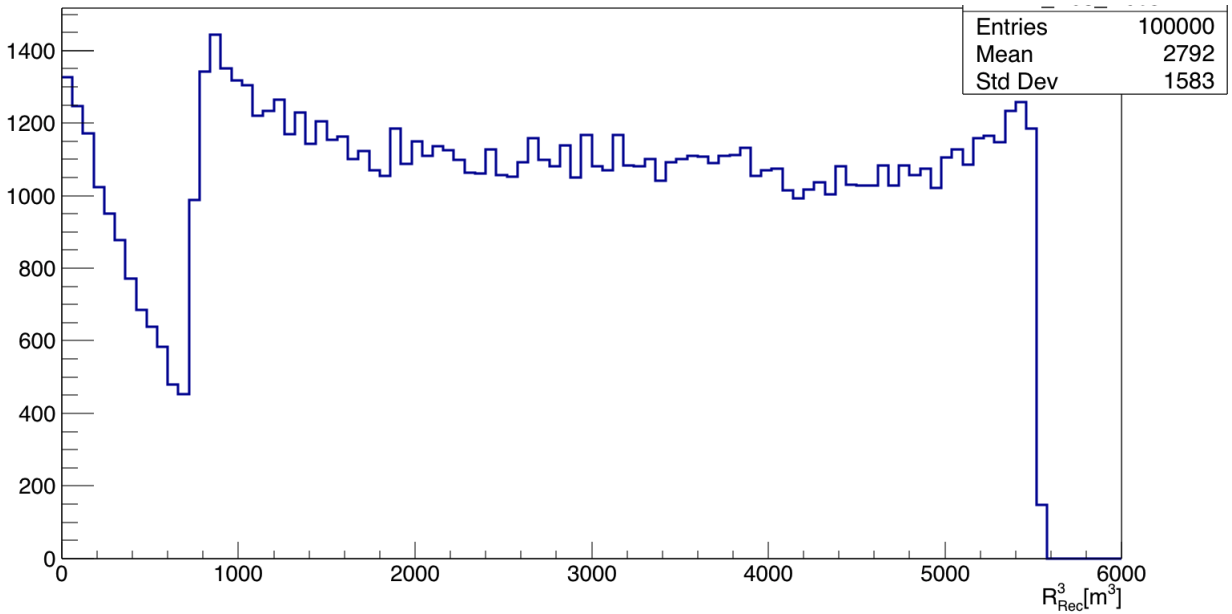
J17v1r1-Pre1

Potential Bug

R_{Edep}



R_{Rec}



Q&A

A. 这种在 `R^3 < 1000 m^3` 的范围内重事例重建失败，是因为对起始时间 `T0` 重建失败。在 `RecTimeLikeAlg::GridSearch()` 方法中，对 `T0` 的重建范围选择：`for(int bin_t = -4;bin_t<5;bin_t++)`，如果真实的 `T0` 超过该范围，则重建失败；增加该参数的范围可以避免失败。

Fix Bug

```
bool RecTimeLikeAlg::GridSearch()
{  m_like_time=1000.0;
   tmp_t_zero = 0;
   for(int init_val = 0;init_val < 1;init_val++){
   //Begin with Charge Center Reconstruction;
   tmp_ve_x =(1.2+init_val*0.1)*ChaCenRec_x;
   tmp_ve_y =(1.2+init_val*0.1)*ChaCenRec_y;
   tmp_ve_z =(1.2+init_val*0.1)*ChaCenRec_z;
//   std::cout<<tmp_ve_x<<std::endl;
   //tag for grid search
   int tag_x = 4;
   int tag_y = 4;
   int tag_z = 4;
   int tag_t = 10;

   double delta = 1e10;

   double step_length = 1000;

   for(int iteration = 0; iteration < 100; iteration++){//100
       if(step_length<0.1) break;
       for(int bin_t = -4;bin_t<5;bin_t++){//-4,5
       for(int bin_x = -1; bin_x<2; bin_x++){//-1,2
           for(int bin_y = -1; bin_y<2; bin_y++){
               for(int bin_z = -1; bin_z<2; bin_z++){
//  std::cout<<"tst binx:"<<bin_x<<"  biny:"<<bin_y<<"  binz:"<<bin_z<<std::endl;
                   double tmp_x = tmp_ve_x+bin_x*step_length;
                   double tmp_y = tmp_ve_y+bin_y*step_length;
                   double tmp_z = tmp_ve_z+bin_z*step_length;
   //       std::cout<<"vtx:"<<tmp_x<<","<<tmp_y<<","<<tmp_z<<","<<tmp_t_zero+bin_t*step_length/100.0<<
                   if((tmp_x*tmp_x+tmp_y*tmp_y+tmp_z*tmp_z)<17700.*17700.){
                   double d = HittimeGoodness(tmp_t_zero+bin_t*step_length/100.0,
                                              tmp_ve_x+bin_x*step_length,
                                              tmp_ve_y+bin_y*step_length,
                                              tmp_ve_z+bin_z*step_length);
//  std::cout<<"tst binx:"<<bin_x<<"  biny:"<<bin_y<<"  binz:"<<bin_z<<std::endl;
                   if(d<delta){
                      tag_x = bin_x;
                      tag_y = bin_y;
                      tag_z = bin_z;
                      tag_t = bin_t;
                      delta = d;
                   }
                  }
                  else continue;
                 }
                }
               }
<Release/J17v1r1-Pre1/offline/Reconstruction/RecTimeLikeAlg/src/RecTimeLikeAlg.cc [RO] 581,12    76%
```

```
bool RecTimeLikeAlg::GridSearch()
{  m_like_time=1000.0;
   tmp_t_zero = 0;
   for(int init_val = 0;init_val < 1;init_val++){
   //Begin with Charge Center Reconstruction;
   tmp_ve_x =(1.2+init_val*0.1)*ChaCenRec_x;
   tmp_ve_y =(1.2+init_val*0.1)*ChaCenRec_y;
   tmp_ve_z =(1.2+init_val*0.1)*ChaCenRec_z;
//   std::cout<<tmp_ve_x<<std::endl;
   //tag for grid search
   int tag_x = 4;
   int tag_y = 4;
   int tag_z = 4;
   int tag_t = 10;

   double delta = 1e10;

   double step_length = 1000;

   for(int iteration = 0; iteration < 100; iteration++){//100
       if(step_length<0.1) break;
       for(int bin_t = -4;bin_t<5;bin_t++){//-4,5
       for(int bin_x = -1; bin_x<2; bin_x++){//-1,2
           for(int bin_y = -1; bin_y<2; bin_y++){
               for(int bin_z = -1; bin_z<2; bin_z++){
//  std::cout<<"tst binx:"<<bin_x<<"  biny:"<<bin_y<<"  binz:"<<bin_z<<std
                   double tmp_x = tmp_ve_x+bin_x*step_length;
                   double tmp_y = tmp_ve_y+bin_y*step_length;
                   double tmp_z = tmp_ve_z+bin_z*step_length;
   //       std::cout<<"vtx:"<<tmp_x<<","<<tmp_y<<","<<tmp_z<<","<<tmp_t_z
                   if((tmp_x*tmp_x+tmp_y*tmp_y+tmp_z*tmp_z)<17700.*17700.)
                   double d = HittimeGoodness(tmp_t_zero+bin_t*step_length
                                              tmp_ve_x+bin_x*step_length,
                                              tmp_ve_y+bin_y*step_length,
                                              tmp_ve_z+bin_z*step_length);
//  std::cout<<"tst binx:"<<bin_x<<"  biny:"<<bin_y<<"  binz:"<<bin_z<<std
                   if(d<delta){
                      tag_x = bin_x;
                      tag_y = bin_y;
                      tag_z = bin_z;
                      tag_t = bin_t;
                      delta = d;
                   }
                  }
                  else continue;
                 }
                }
               }
<Release/J17v1r1-Pre2/offline/Reconstruction/RecTimeLikeAlg/src/RecTimeLikeA
```

Fix Bug

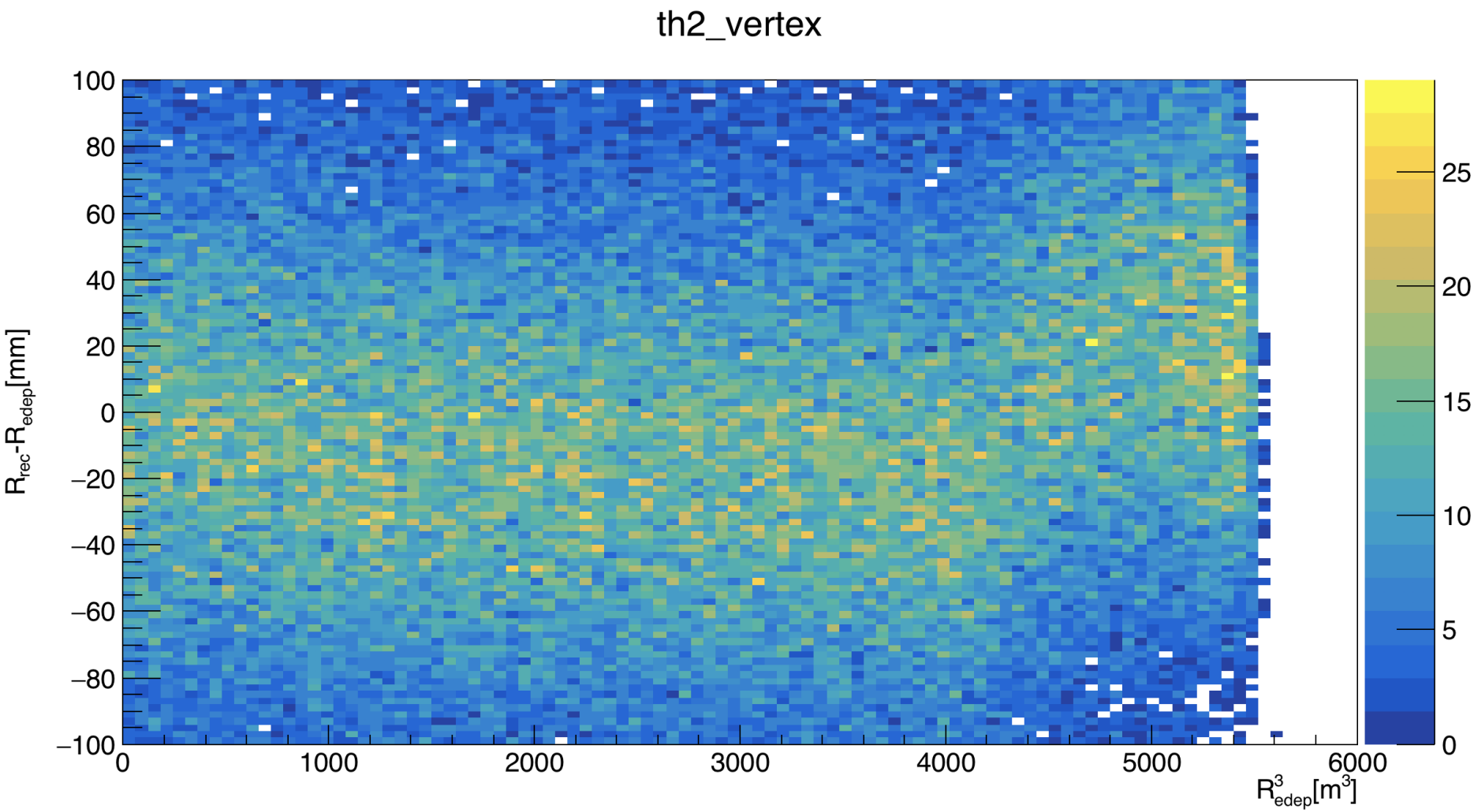
```
//     std::cout<<"HittimeGoodness"<<std::endl;
   for(int i = 0; i<num_PMT; i++){
    // if(HIT_CUT[i] == false){
     if(Readout_PE.at(i)!=0){
  //    std::cout<<Readout_hit_time.at(i)<<std::endl;
      double timeflight = RecTimeLikeAlg::CalculateTOF(Vert_x, Vert_y, Vert_z, i);
      relaTime=Readout_hit_time.at(i)-timeflight+T_zero;

      if(relaTime>-30&&relaTime<300){//by default it is 300 ns
        int BinTime = (int)((relaTime+200)*200/7);

        double pdfValue;

        if(m_Pdf_Value == 0){
             if(Readout_PE.at(i)<1.5)                              pdfValue = pdf_1hit->GetBinContent(B
             else if(Readout_PE.at(i)>=1.5&&Readout_PE.at(i)<2.5) pdfValue = pdf_2hit->GetBinContent(B
             else if(Readout_PE.at(i)>=2.5&&Readout_PE.at(i)<3.5) pdfValue = pdf_3hit->GetBinContent(B
             else if(Readout_PE.at(i)>=3.5&&Readout_PE.at(i)<4.5) pdfValue = pdf_4hit->GetBinContent(B
             else pdfValue = pdf_5hit->GetBinContent(BinTime);
        }
        else if(m_Pdf_Value == 1){
           if(PMT_TTS.at(i)>8.0){
              if(Readout_PE.at(i)<1.5)                              pdfValue = pdf_1hit->GetBinContent(B
              else if(Readout_PE.at(i)>=1.5&&Readout_PE.at(i)<2.5) pdfValue = pdf_2hit->GetBinContent(B
              else if(Readout_PE.at(i)>=2.5&&Readout_PE.at(i)<3.5) pdfValue = pdf_3hit->GetBinContent(B
              else if(Readout_PE.at(i)>=3.5&&Readout_PE.at(i)<4.5) pdfValue = pdf_4hit->GetBinContent(B
              else pdfValue = pdf_5hit->GetBinContent(BinTime);
             }
           else if(PMT_TTS.at(i)<8.0){
             if(Readout_PE.at(i)<1.5)                              pdfValue = pdf_1hit_1ns->GetBinConte
              else if(Readout_PE.at(i)>=1.5&&Readout_PE.at(i)<2.5) pdfValue = pdf_2hit_1ns->GetBinConte
              else if(Readout_PE.at(i)>=2.5&&Readout_PE.at(i)<3.5) pdfValue = pdf_3hit_1ns->GetBinConte
              else if(Readout_PE.at(i)>=3.5&&Readout_PE.at(i)<4.5) pdfValue = pdf_4hit_1ns->GetBinConte
              else pdfValue = pdf_5hit->GetBinContent(BinTime);
            }
         }
        if(pdfValue!=0){
           Goodness = Goodness-TMath::Log(pdfValue);
        }
        else{
           Goodness = Goodness+0;
        }

        FiredPMT++;
      }
    }
```

<Release/J17v1r1-Pre1/offline/Reconstruction/RecTimeLikeAlg/src/RecTimeLikeAlg.cc [RO] 518,6 67%

```
//     std::cout<<"HittimeGoodness"<<std::endl;
   for(int i = 0; i<num_PMT; i++){
    // if(HIT_CUT[i] == false){
     if(Readout_PE.at(i)!=0){
  //    std::cout<<Readout_hit_time.at(i)<<std::endl;
      double timeflight = RecTimeLikeAlg::CalculateTOF(Vert_x, Vert_y, Ver
      relaTime=Readout_hit_time.at(i)-timeflight+T_zero-min_hit_time;

      if(relaTime>-30&&relaTime<300){//by default it is 300 ns
        int BinTime = (int)((relaTime+200)*200/7);

        double pdfValue;

        if(m_Pdf_Value == 0){
             if(Readout_PE.at(i)<1.5)                              pdfValu
             else if(Readout_PE.at(i)>=1.5&&Readout_PE.at(i)<2.5) pdfValu
             else if(Readout_PE.at(i)>=2.5&&Readout_PE.at(i)<3.5) pdfValu
             else if(Readout_PE.at(i)>=3.5&&Readout_PE.at(i)<4.5) pdfValu
             else pdfValue = pdf_5hit->GetBinContent(BinTime);
        }
        else if(m_Pdf_Value == 1){
           if(PMT_TTS.at(i)>8.0){
              if(Readout_PE.at(i)<1.5)                              pdfValu
              else if(Readout_PE.at(i)>=1.5&&Readout_PE.at(i)<2.5) pdfValu
              else if(Readout_PE.at(i)>=2.5&&Readout_PE.at(i)<3.5) pdfValu
              else if(Readout_PE.at(i)>=3.5&&Readout_PE.at(i)<4.5) pdfValu
              else pdfValue = pdf_5hit->GetBinContent(BinTime);
             }
           else if(PMT_TTS.at(i)<8.0){
             if(Readout_PE.at(i)<1.5)                              pdfValu
              else if(Readout_PE.at(i)>=1.5&&Readout_PE.at(i)<2.5) pdfValu
              else if(Readout_PE.at(i)>=2.5&&Readout_PE.at(i)<3.5) pdfValu
              else if(Readout_PE.at(i)>=3.5&&Readout_PE.at(i)<4.5) pdfValu
              else pdfValue = pdf_5hit->GetBinContent(BinTime);
            }
         }
        if(pdfValue!=0){
           Goodness = Goodness-TMath::Log(pdfValue);
        }
        else{
           Goodness = Goodness+0;
        }

        FiredPMT++;
      }
    }
```

<Release/J17v1r1-Pre2/offline/Reconstruction/RecTimeLikeAlg/src/RecTimeLikeA

Fix Bug


th2_vertex
100
80
60
40
20
0
−20
−40
−60
−80
−100
R_{rec}-R_{edep}[mm]
0
1000
2000
3000
4000
5000
6000
R^3_{edep}[m^3]
25
20
15
10
5
0

X,Y,Z Direction

X,Y,Z direction resolution from Q.LIU, NIMA (arXiv:1803.09394v1)
$\sigma_{x,y,z} \sim 42.5mm$.

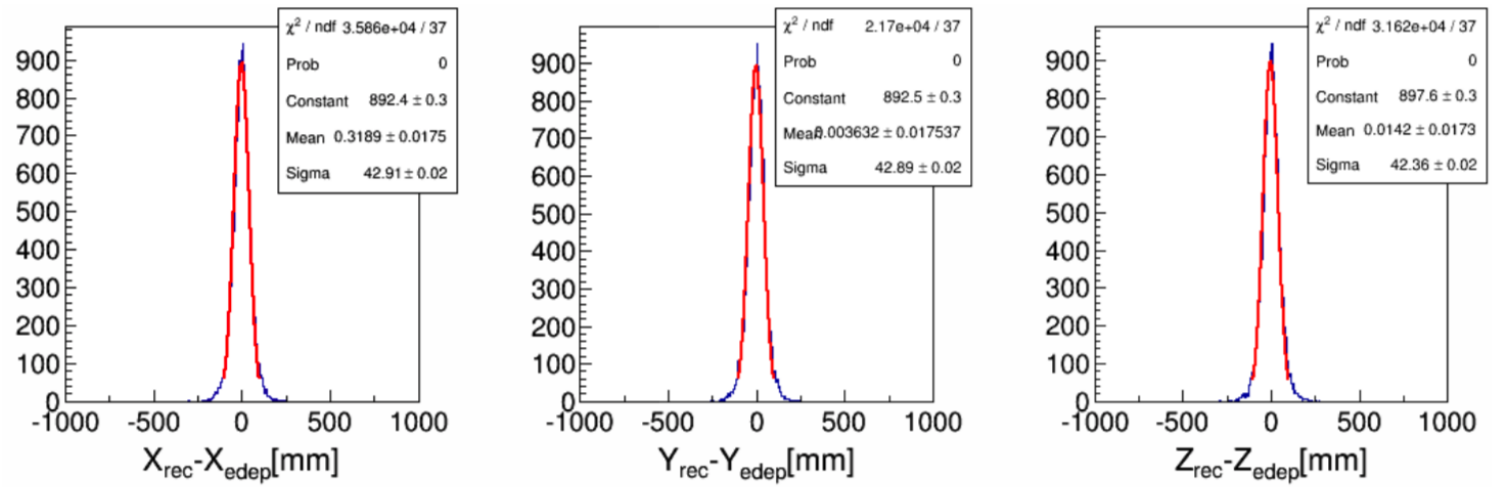
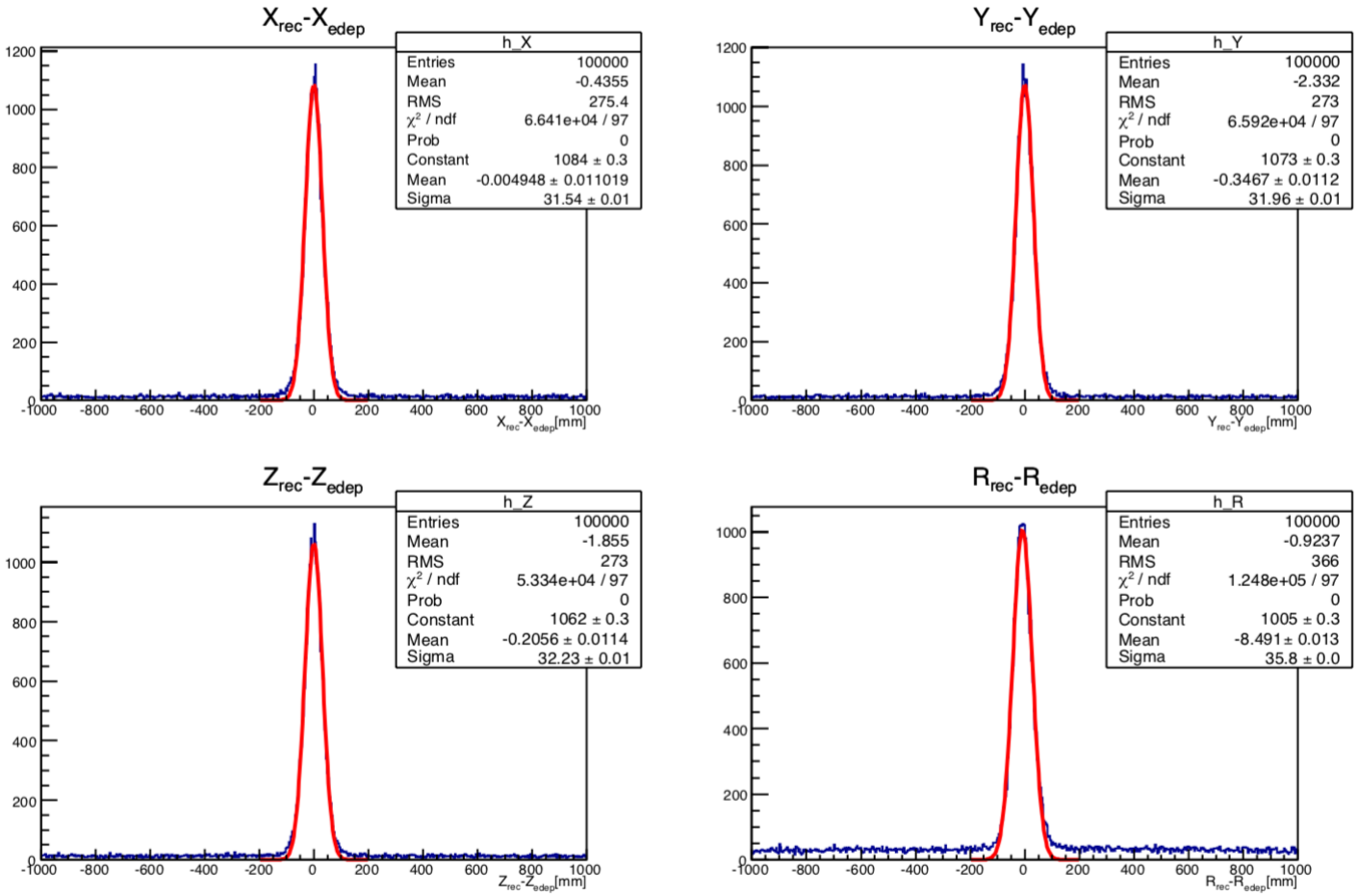


Fig. 11. Reconstructed vertex resolution of X, Y, Z direction. The mean values are consistent with MC truth and σ are similar for three directions.

X,Y,Z Direction (Reproduce Q.LIU's Plot)

X,Y,Z direction resolution for J17v1r1, e+, P=2MeV

$\sigma_{x,y,z} \sim 32mm, \sigma_r \sim 36mm.$



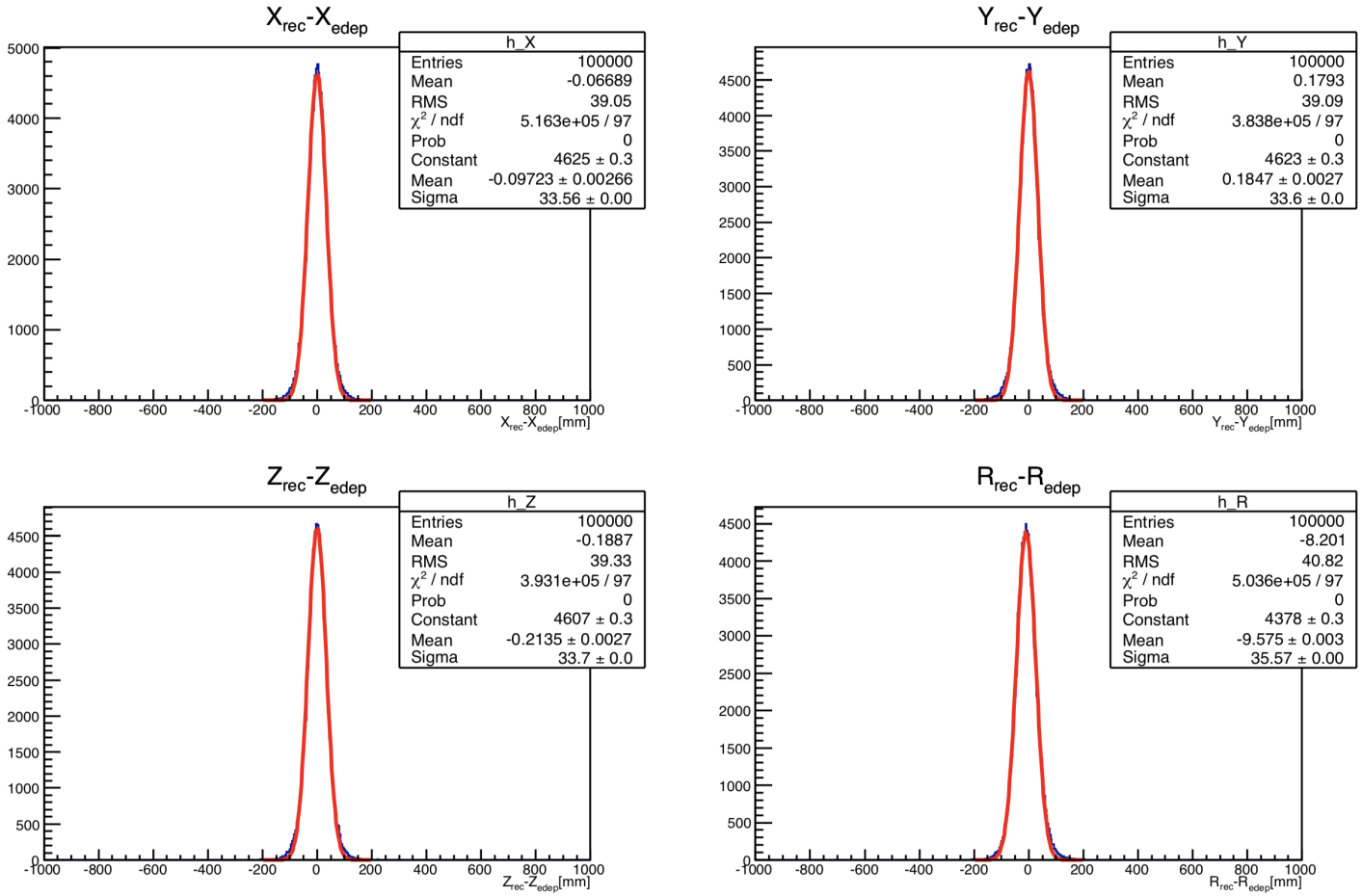
Q&A

A. 没有 `selection`，可能原因：同上，还有一个可能原因是在 `tut_det2calib.py` 中，`parser.add_argument("--split-gap-time")` 把这个参数默认值调大，这是一个类似时间窗口的设置（只在没有电子学模拟的时候用到），窗口过小会导致将一个模拟事例拆成两个，特别是对低能的事例，从而导致重建和模拟事例不一一对应。建议对高能的事例也产生一些结果观察。

X,Y,Z Direction (Reproduce Q.LIU's Plot)

X,Y,Z direction resolution for J17v1r1(change split-gap-tme), e+, P=2MeV

$\sigma_{x,y,z} \sim 32mm, \sigma_r \sim 36mm.$



Vertex Bias

Vertex Bias from Q.LIU, NIMA (arXiv:1803.09394v1)

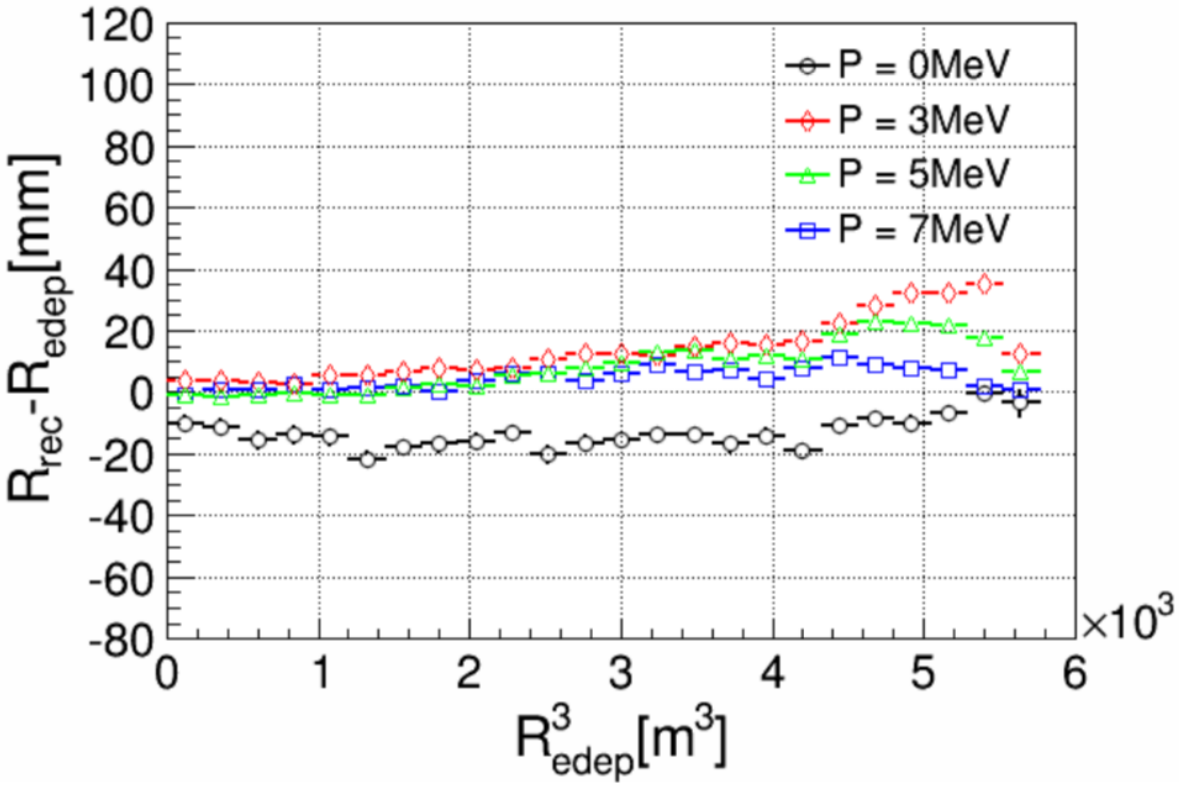
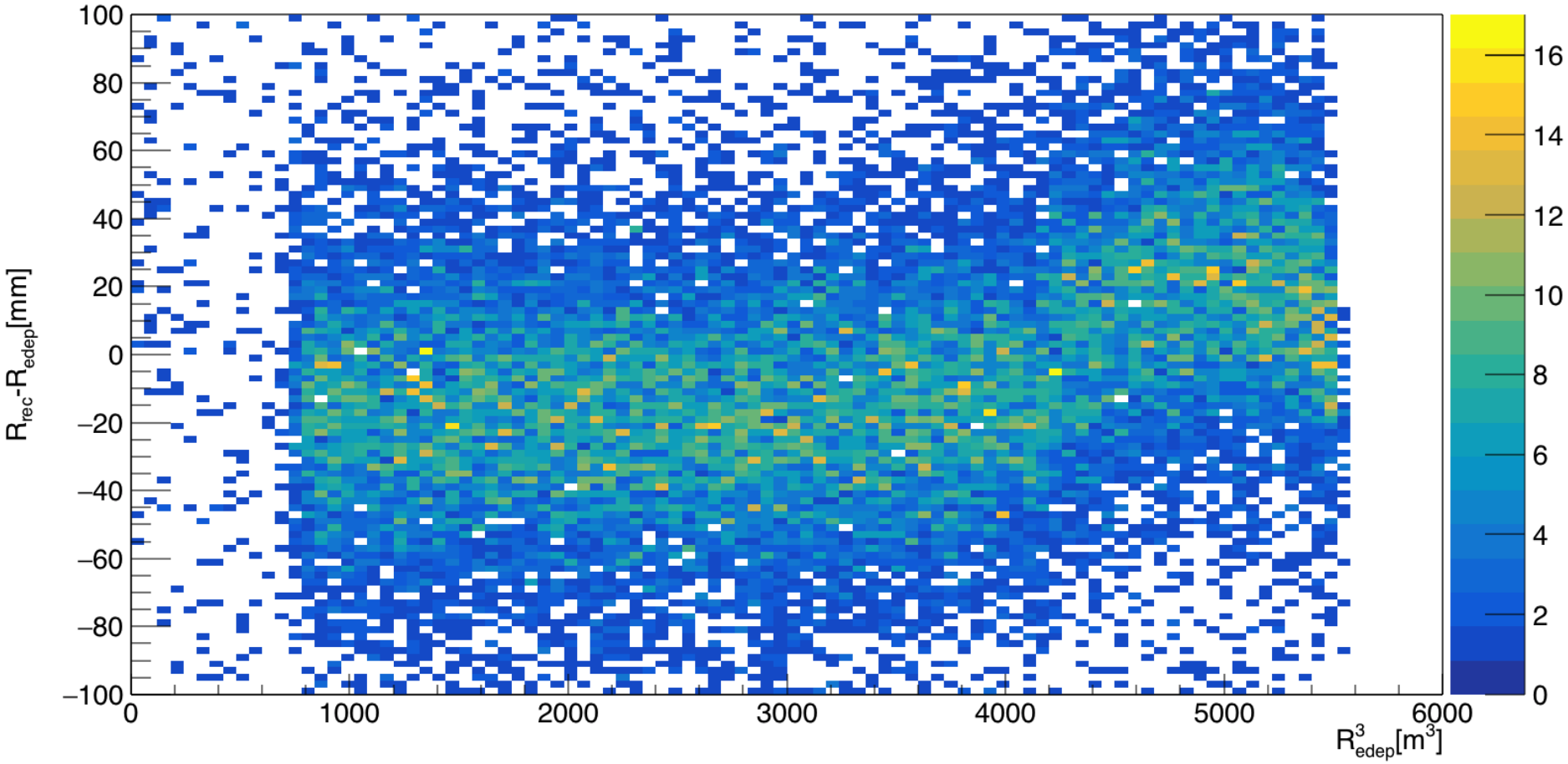


Fig. 10. Reconstructed vertex bias. By deploying the 5-pdf, considering refraction and set a time window from -5 ns to 30 ns, the bias of reconstructed vertex is less than 4 cm without any correction.

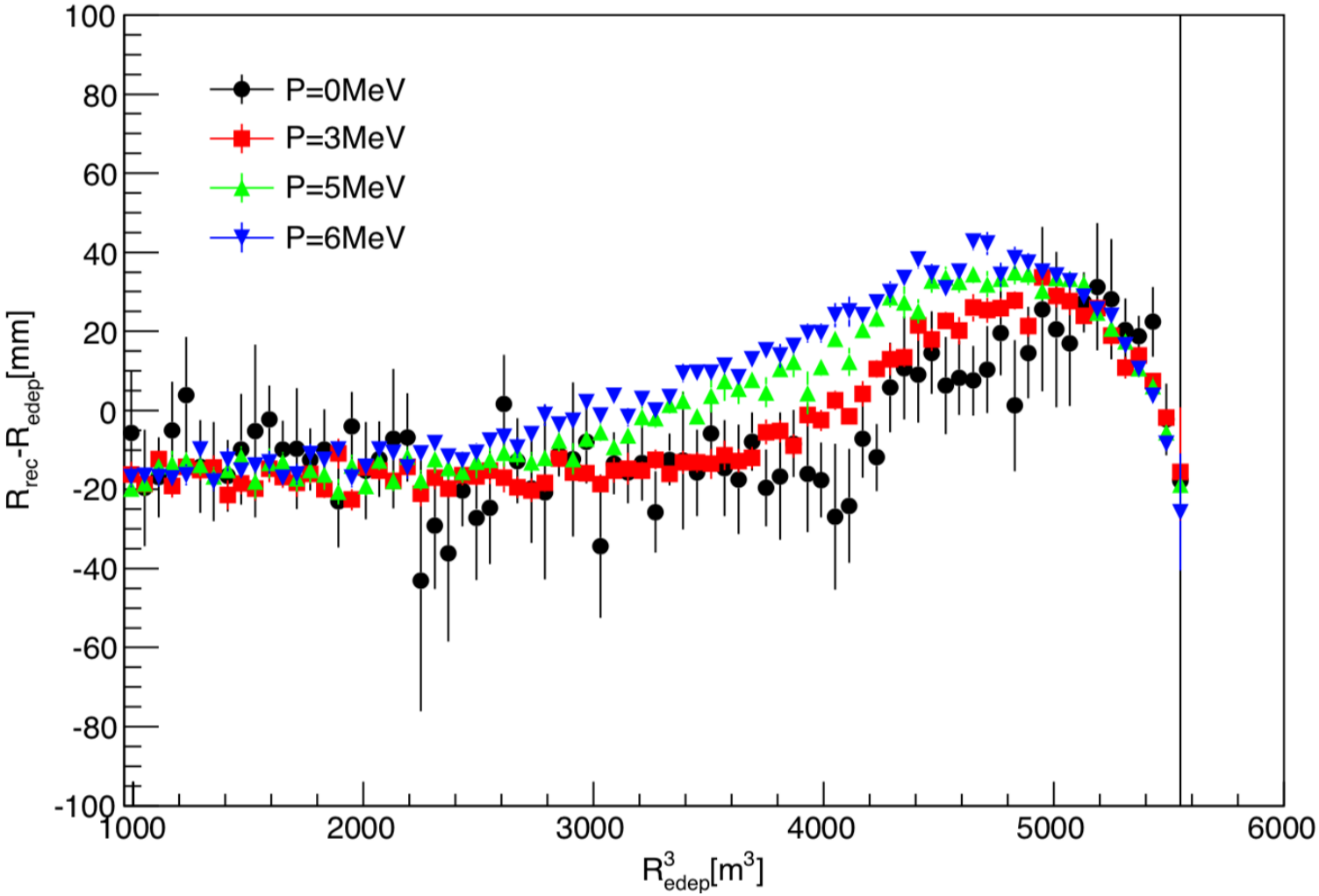
Vertex Bias (Reproduce Q.LIU's Plot)

Vertex Bias for J17v1r1, e+, P=2MeV
Use th2.FitSlicesY() to get mean and sigma.



Vertex Bias (Reproduce Q.LIU's Plot)

Vertex Bias for J17v1r1, e+, P=0,3,5,6MeV
Use th2.FitSlicesY() to get mean and sigma.

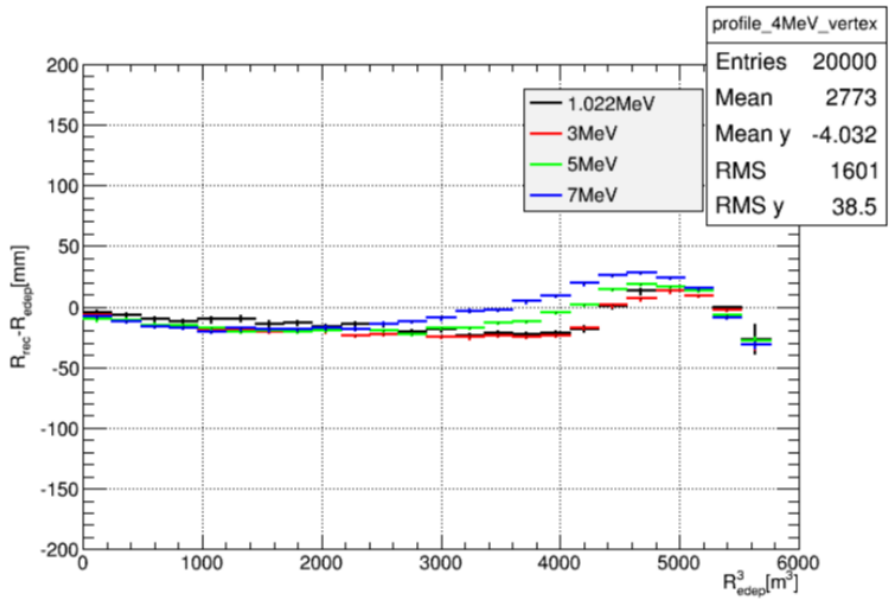


Vertex Bias

JUNO-doc-3144-v2, from Qin LIU, 2018

Residual bias

- Use straight line and n_{LS_eff}= 1.54 to calculate TOF;
- Time window range: [-30ns,300ns];
- Pdf:4.4MeV γ@center
- With electronic simulation, w/o dark noise.
- Residual bias is less than 3cm, but there seems to be some structure.



Vertex Resolution

Vertex resolution from Q.LIU, NIMA (arXiv:1803.09394v1)

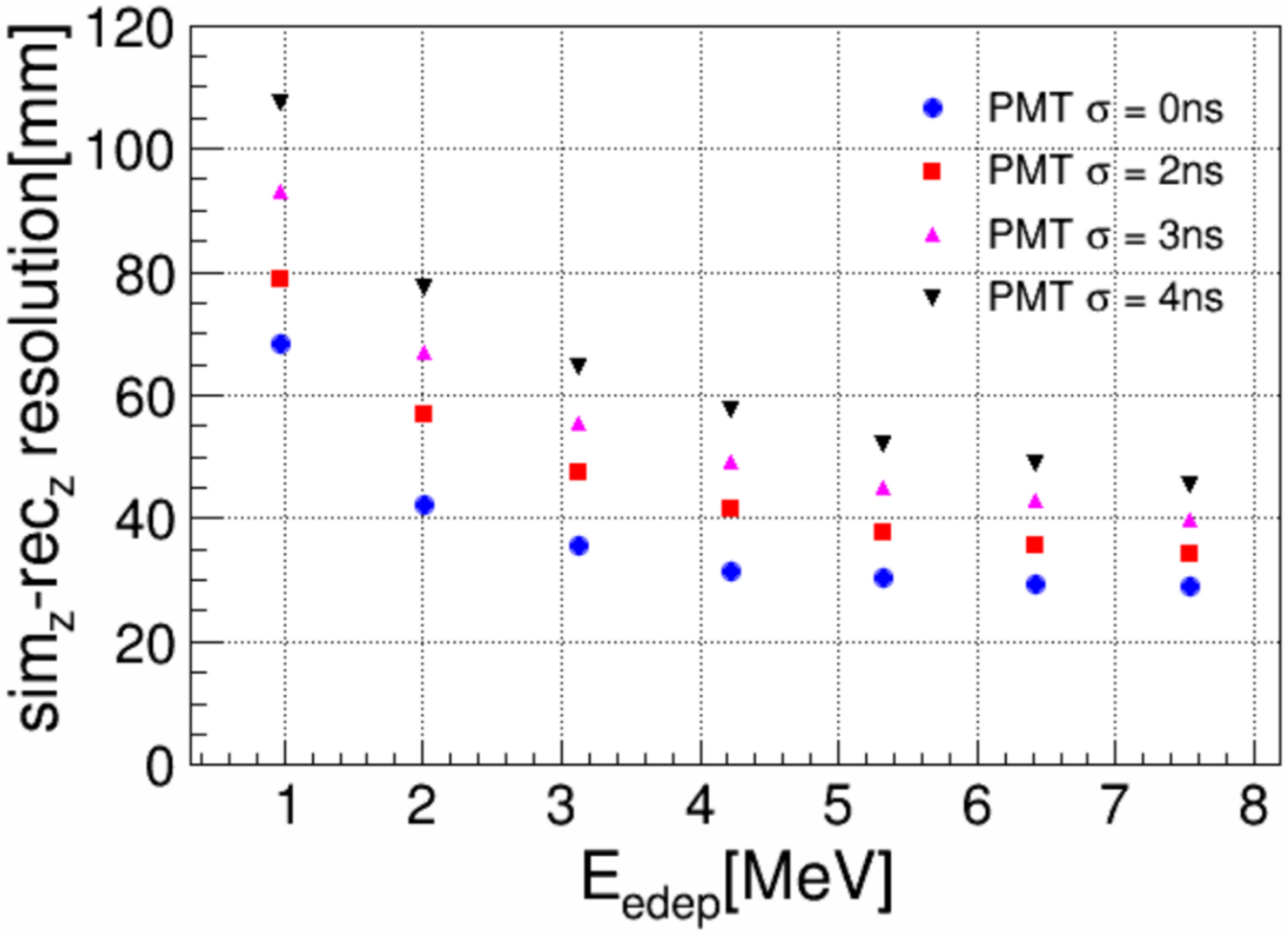
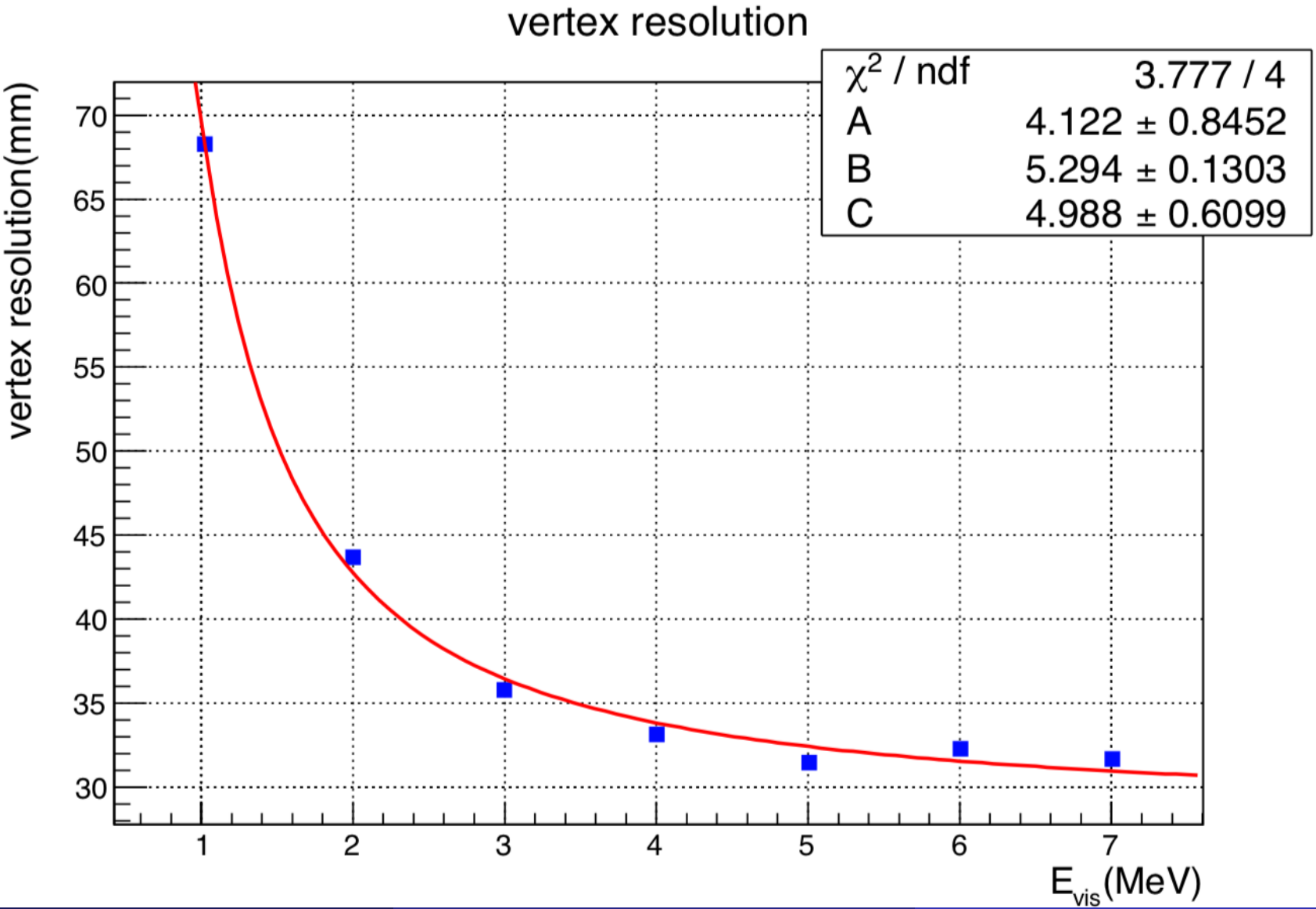


Fig. 12. PMT TTS impact on vertex resolution.

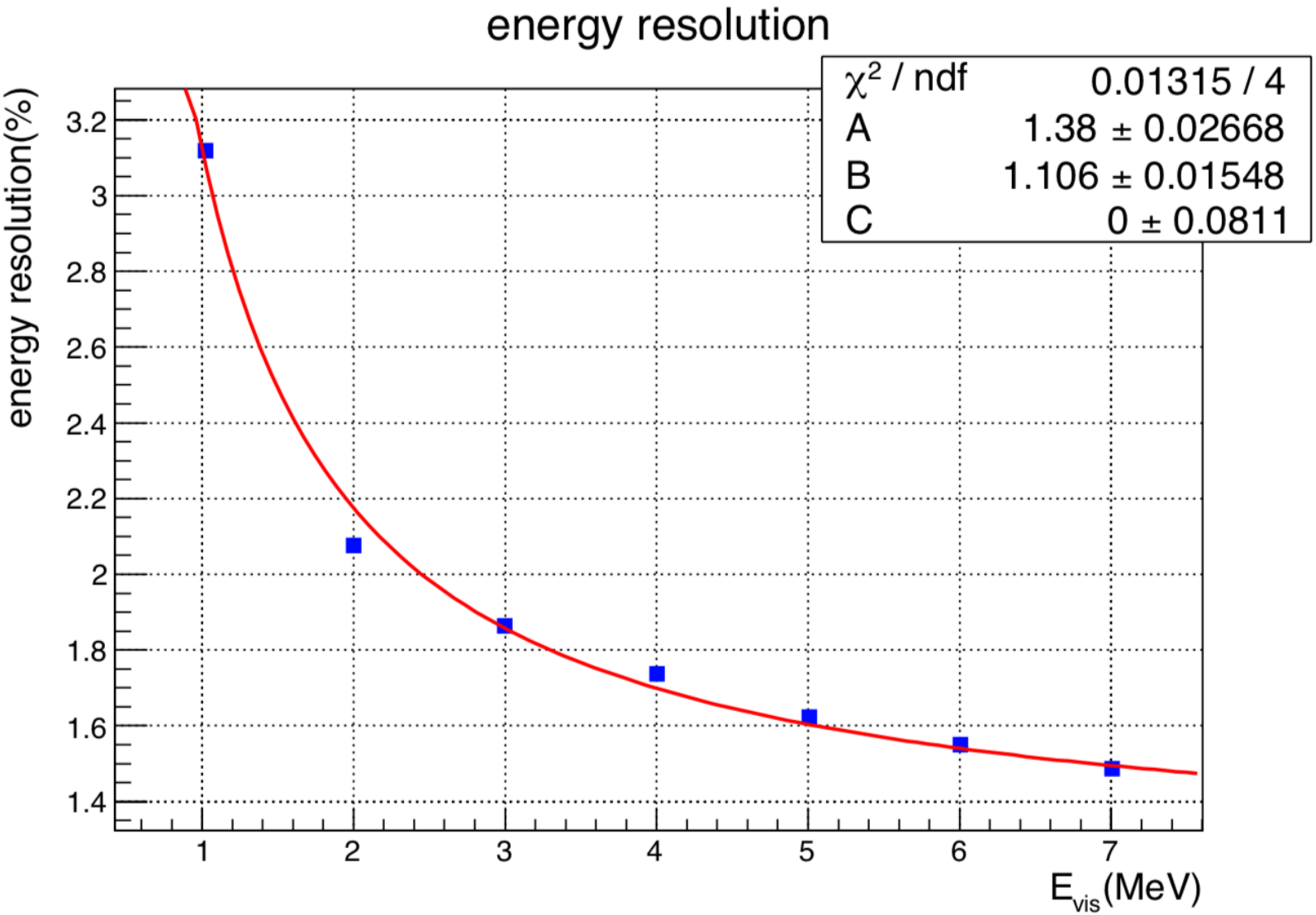
Vertex Resolution (Reproduce Q.LIU's Plot)

Vertex resolution for J17v1r1, e+



Energy Resolution (Reproduce Q.LIU's Plot)

Energy resolution for J17v1r1, e+



JUNO-doc-3144-v2, from Qin LIU, 2018

Algorithm principle

When particles like e+, e- or γ, deposit energy in liquid scintillator and light, define residual time:

$$t_{i,res} = t_i - tof - t_0,$$

where
t_i: *first hit time of ith PMT*;
tof: *time of flight for scintillation photon*;
t_0: *real time of an event*.

For point-like events(deposit energy <10MeV), the residual time obey the formula:

$$f(t_{res}) = \frac{1}{\sqrt{2\pi}\sigma} exp\left(-\frac{(t-t_0)^2}{2\sigma^2}\right) \otimes \left[\frac{\omega}{\tau_1} e^{-\frac{t}{\tau_1}} + \frac{1-\omega}{\tau_2} e^{-\frac{t}{\tau_2}}\right]$$

σ: *systematic uncertainty of first hit time*;
τ_1/τ_2: *time constant of fast/slow component of LS*

Theoretical Foundation of the Algorithm

PRL **101,** 077402 (2008) PHYSICAL REVIEW LETTERS week ending 15 AUGUST 2008

Luminescence Decay and the Absorption Cross Section of Individual Single-Walled Carbon Nanotubes

Stéphane Berciaud, Laurent Cognet, and Brahim Lounis*

Centre de Physique Moléculaire Optique et Hertzienne, Université de Bordeaux and CNRS, 351 cours de la Libération, Talence, F-33405, France

(Received 4 April 2008; published 14 August 2008)

The absorption cross section of highly luminescent individual single-walled carbon nanotubes is determined using time-resolved and cw luminescence spectroscopy. A mean value of $\sim 1 \times 10^{-17}$ cm^2 per carbon atom is obtained for (6, 5) tubes excited at their second optical transition, and corroborated by single tube photothermal absorption measurements. Biexponential luminescence decays are systematically observed, with short and long lifetimes around 45 and 250 ps. This behavior is attributed to the band edge exciton fine structure with a dark level lying a few meV below a bright one.

DOI: 10.1103/PhysRevLett.101.077402 PACS numbers: 78.67.Ch, 73.22.Dj, 78.47.Cd

Theoretical Foundation of the Algorithm

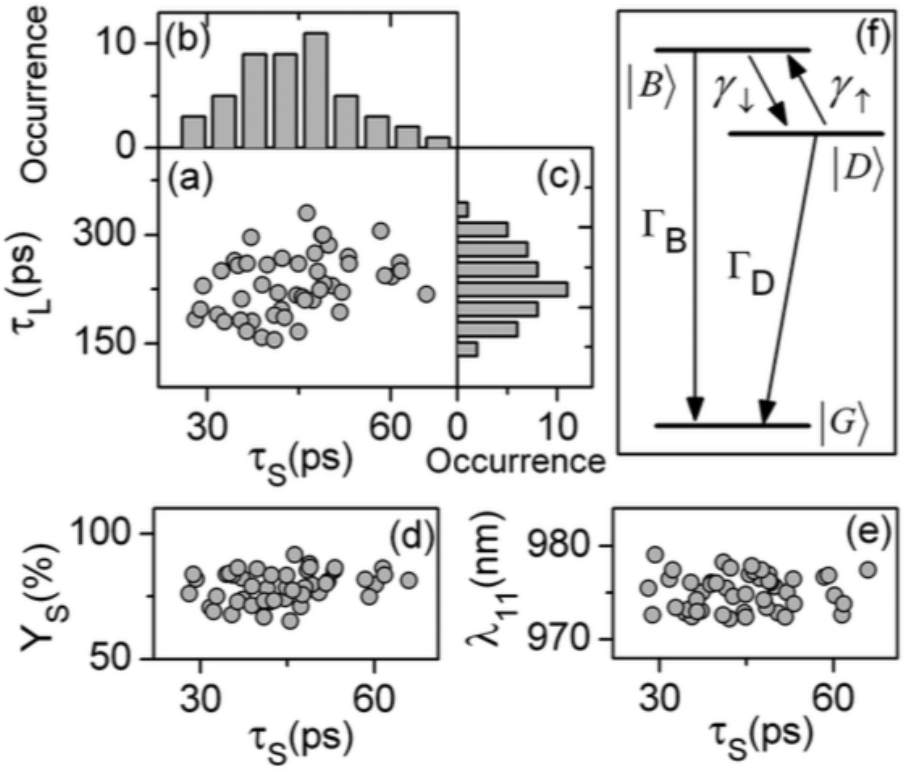


FIG. 2. (a) Long decay times as function of short ones, deduced from 48 luminescence decays of individual (6, 5) SWNTs. (b),(c) The corresponding histograms. Weights of the short decay components (d) and peak emission wavelengths (e) as a function of the short decay times. (f) Three level scheme used in the model.

Figure 1(c) shows the luminescence decay obtained from the nanotube of Fig. 1(a). In order to prevent multiple exciton formation, a very low photon fluence of 10^{12} photons/pulse/cm^2 was used with 10 min integration times. We further checked that the nanotubes displayed constant emission intensities during the acquisitions [Fig. 1(d)] and identical spectra at the end of the acquisitions. All the tube decays were accurately fitted by biexponential functions while monoexponential fits could not reproduce the experimental data [Figs. 1(c), 1(e), and 1(f)]. For the fits we used the convolution of the instrumental response function with a double exponential function $A_S \exp(-t/\tau_S) + A_L \exp(-t/\tau_L)$ with $\tau_L > \tau_S$. The statistical distributions of the short and long times built from 55 luminescence decays are shown in Figs. 2(a)–2(e). The fast decay times τ_S ranged from 28 to 66 ps [mean value of 44 ± 9 ps, Fig. 2(b)] while τ_L values are all found between 155 and 330 ps [mean 230 ± 40 ps, Fig. 2(c)]. No correlation is observed between the short and long decay times. In addition, the decay times are uncorrelated with the peak emission wavelengths and the weight of the fast decay, defined as $Y_S = A_S\tau_S/(A_S\tau_S + A_L\tau_L)$ (mean $80 \pm 6\%$).

Algorithm based on first hit time

- Initial vertex : charge center method
- For a certain scanned point $\overrightarrow{R_0}$, calculate t_{res}:

$$t_{i,res} = t_i - \sum_\alpha \frac{L_\alpha(\overrightarrow{R_0}, \overrightarrow{R_i})}{c_\alpha} - t_0,$$

where α represent different materials.

- Calculate joint likelihood value:

$$F(\overrightarrow{R_0}, t_0) = -\sum_i \log(f(t_{i,res}))$$

Minimize the likelihood function to get best estimate value.

5-PDF

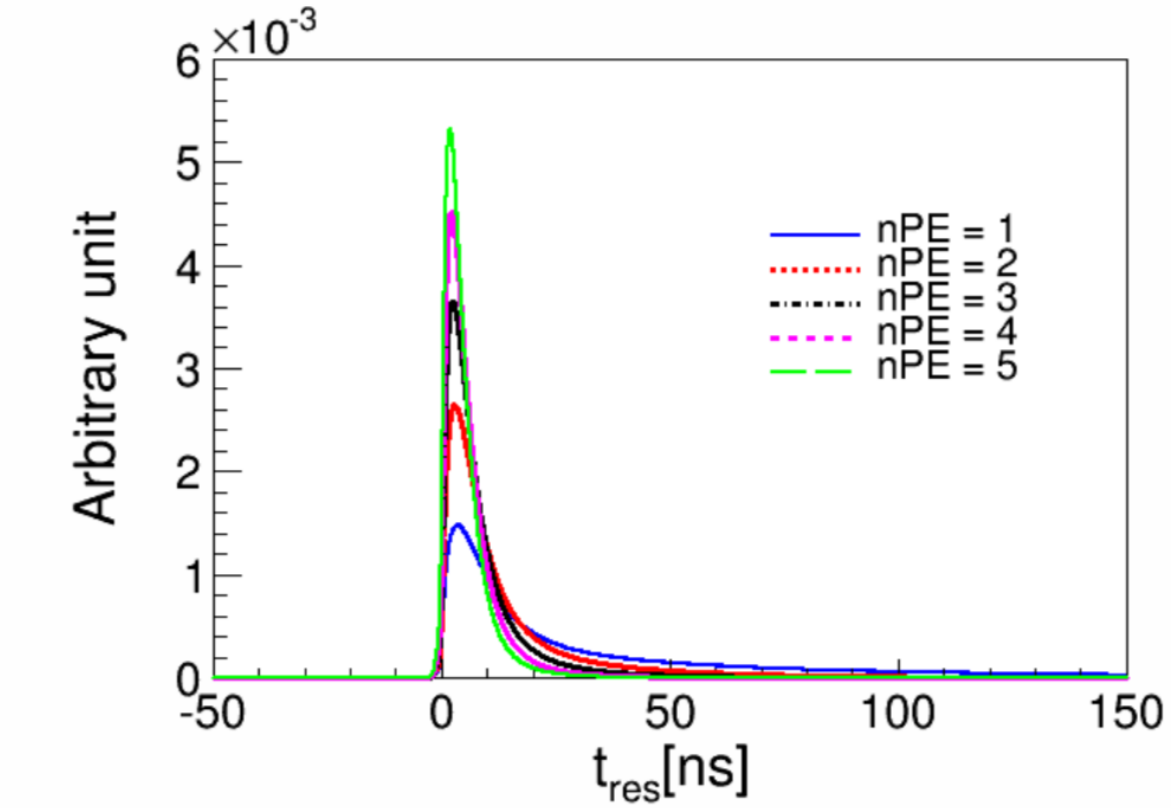


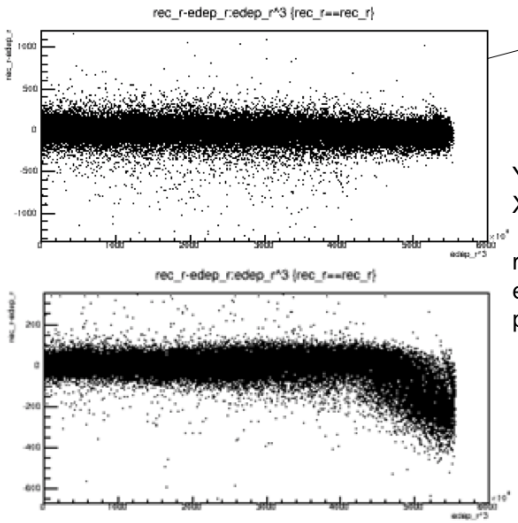
Fig. 7. PDF for different number of photons. The more PE, the sharper the PDF is. When number of PE is more than 5, the PDF are similar and obey gaussian distribution.

The reason to use 5 PDF

JUNO-doc-604-v1, from Xiaozhong HUANG, 2014

Problems(use RecTimeLikeAlg from USTC)

only use PMT with 1PE



P=0MeV
e+

Y:rec_r-edep_r(mm)
X:edep_r^3

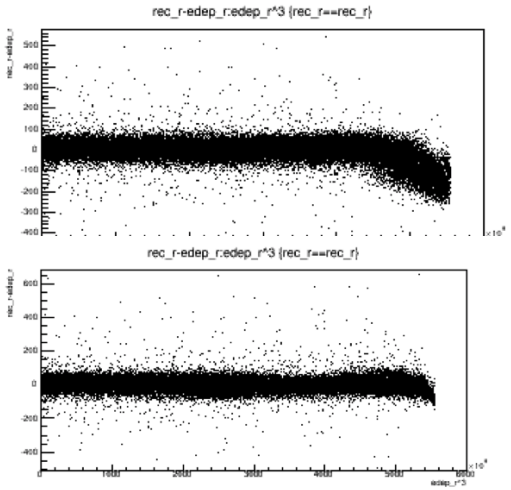
rec_r:radius of reconstructed vertex
edep_r:radius of energy deposit position

P=4.46MeV e+

9

Problems(use RecTimeLikeAlg from USTC)

P=4.46MeV e+



only use PMT with nPE(n=1,2,3,4)

:rec_r-edep_r(mm)
:edep_r^3

ec_r:radius of reconstructed vertex
edep_r:radius of energy deposit position

use PMT with nPE(n=1,2,3,4,5,.....)

10

The reason to use 5 PDF

JUNO-doc-604-v1, from Xiaozhong HUANG, 2014

Problems about Vertex Reconstruction

For e+ with high energy:

If the vertex is close to the acrylic ball, the nearby PMT will be hit by more than 5PE.

If we only use PMT with 1 PE, we will not use PMT near the vertex, thus making the reconstructed vertex approach center of ball.

We can get the "right" result by using all fired PMTs, but I think the performance is not stable.

Q&A

A. 算法中关于不同 pe 的 pdf，是直接从 MC 模拟中得到的，并没有用公式计算。将 4.4MeV 的 gamma 放置在球心，分别记录不同 pe 的 PMT 的第一击中时间。对于多 pe 的 pdf，由于统计量锐减，因此不是直接从 MC 得到，而是利用 Npe 的 pdf 与单 pe 的 pdf 之间的关系推导出来的（paper 中有提到）。公式中的参数，可以对 MC 得到的 pdf 进行拟合得到。

$$f(t,N) = N f(t) \left(\int_t^{+\infty} f(x)dx \right)^{N-1}$$

Discussion

- What's the main issue of this method?
- How to improve the performance?

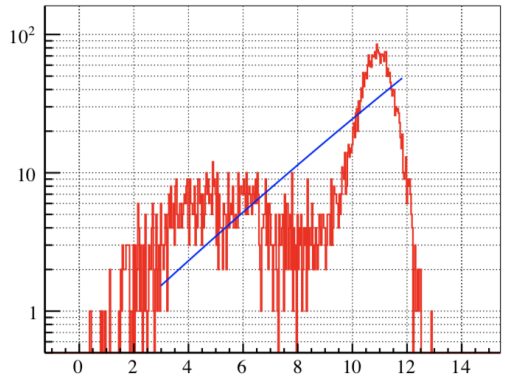
Discussion

JUNO-doc-3126-v1, from Xuefeng DING, 2018

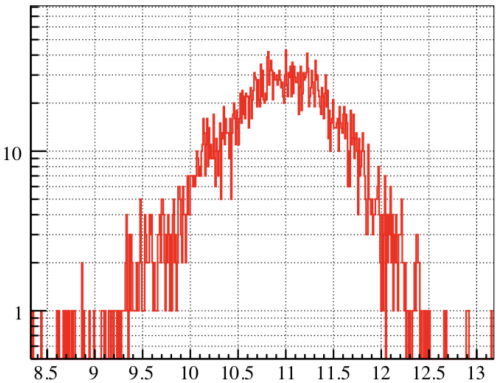
dark bump for low energies events

Almost same algorithms, except different initial guess

RecTimeLikeAlg



PosRecTimeLikeAlgV3



e-, E_k = 0.4 MeV
DarkRate 30k Hz
Trigger = 300 N_{pmt}
simulate at (0,0,11) m

Next Step

- Reproduce 5-PDF independently.
- Try new way to define PDF and compare the performance.
- Restart professor YOU's work in 2013 about Vertex Reconstruction and check the performance with better understanding of our detector.
- Try Machine Learning method in the future.