STCF快速模拟介绍

秦小帅

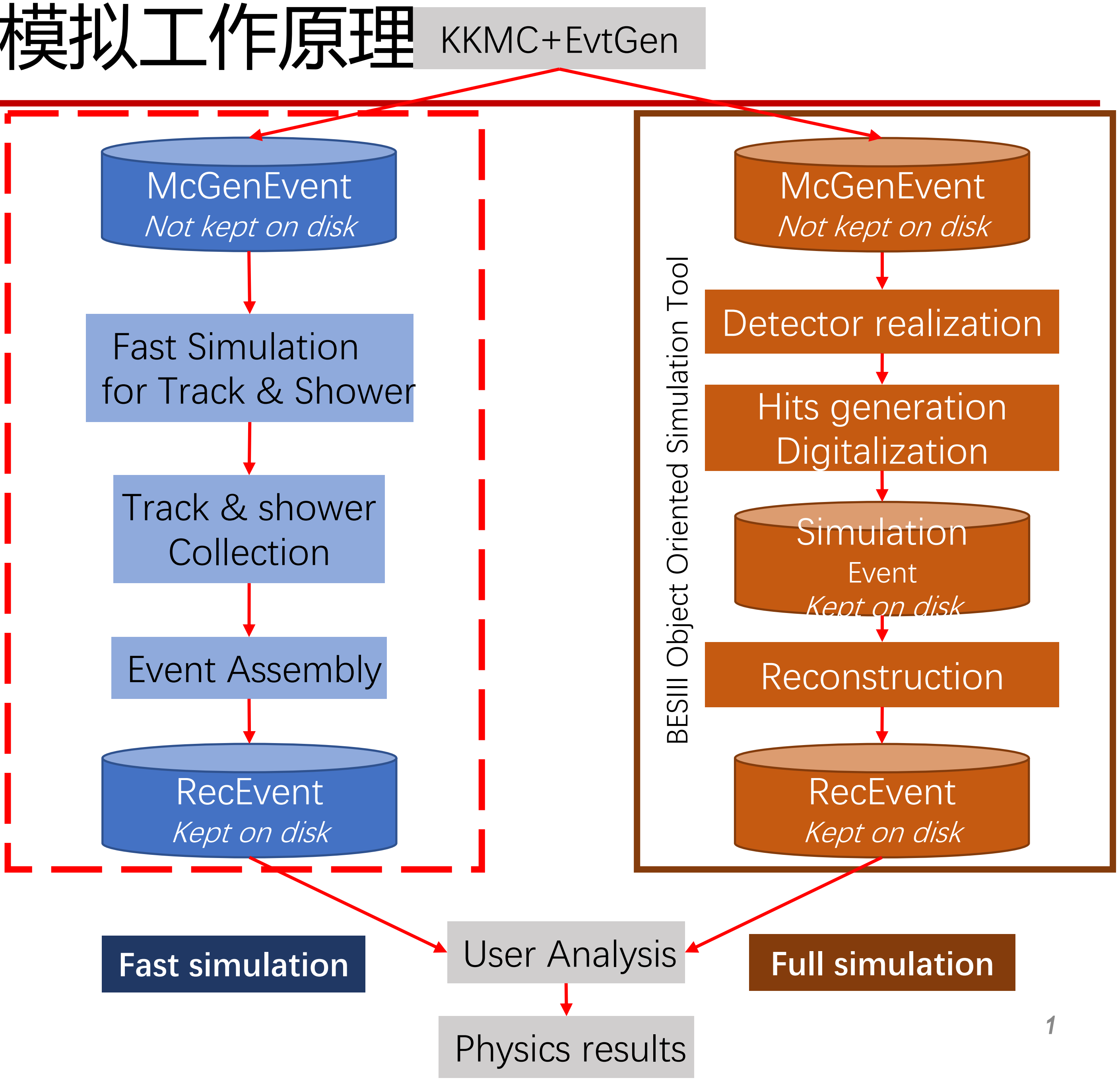
山东大学

快速模拟工作原理

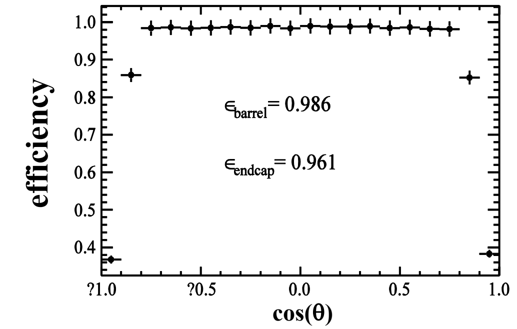
- 应对高亮度对撞实验中大统计量数据的产生、存储挑战

- 通过全重建样本得到探测器性能：
- 能动量分辨、位置分辨、重建效率、粒子鉴别效率等
- 细分到特定 P, cos(θ), ϕ 区间， 以便准确描述
- 获得探测器响应的CDF曲线、效率直方图

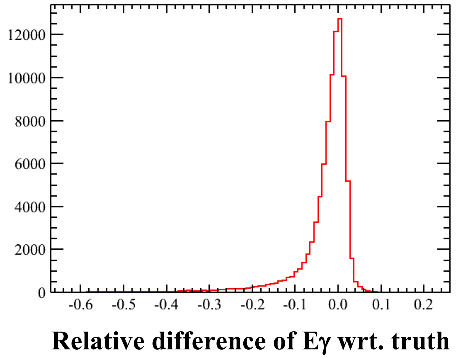
- 通过产生子拿到粒子四动量信息
- 利用探测器响应的CDF曲线、效率直方图进行抽样

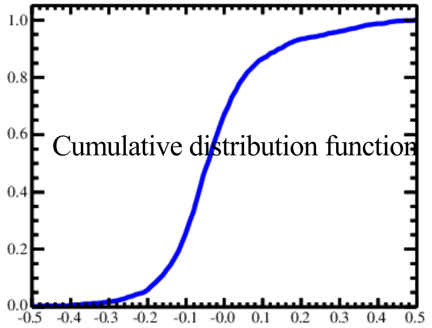


快速模拟的实现



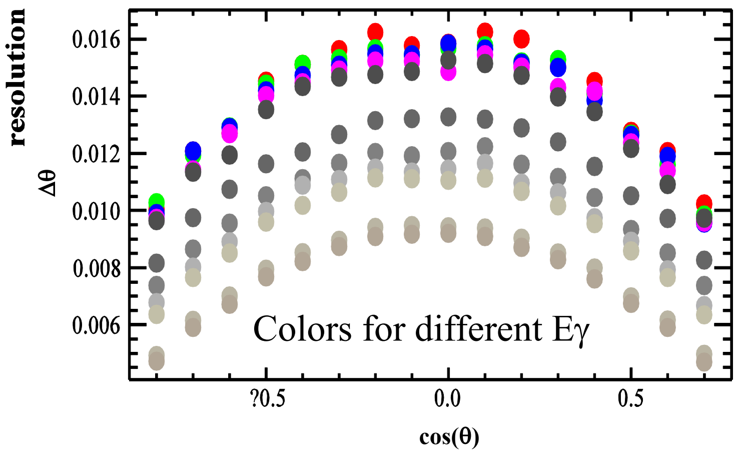
Efficiency sampling

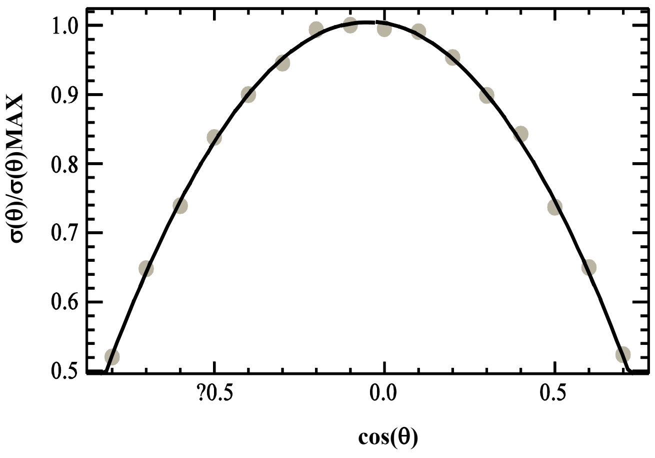




Smear

- ◆Efficiency sampling
- ◆Momentum smear
- ◆Direction parameterization/smear





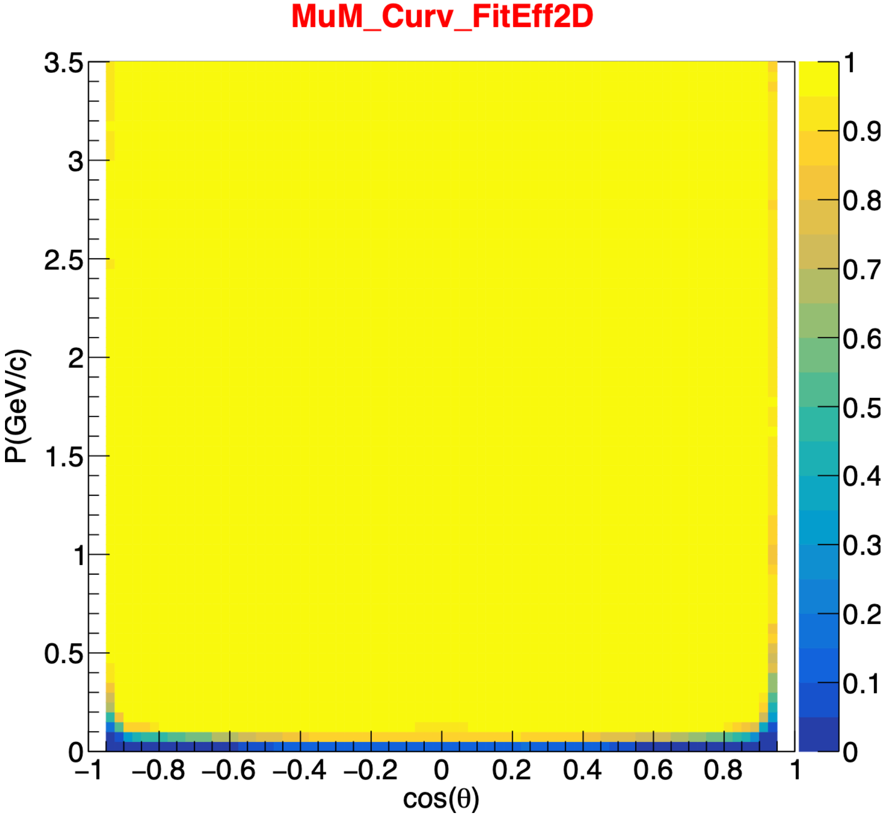
Parameterization
σ(pt, cos(θ))

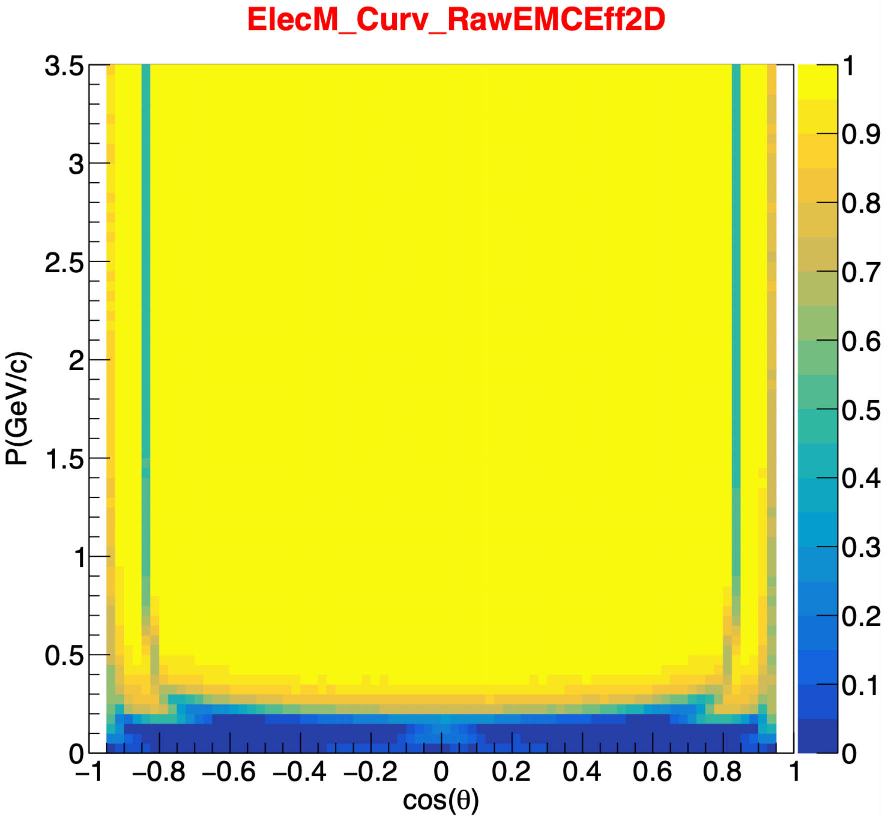
OSCAR下重建效率与分辨

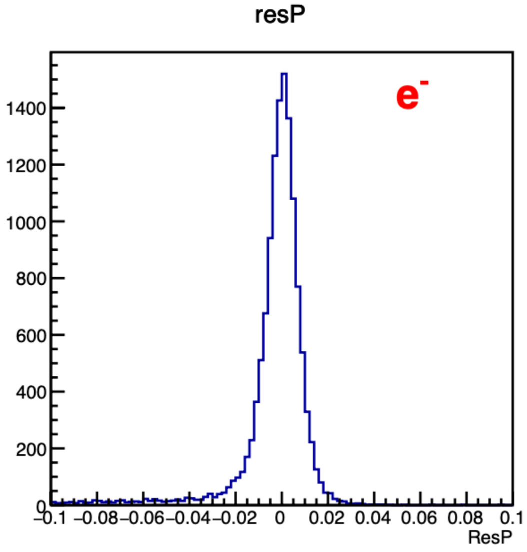
- ◆以原点产生的按角度、动量细分的单粒子重建样本为基础
- ◆**报告中效率、分辨等图未更新到最新版本，仅作为方法展示**

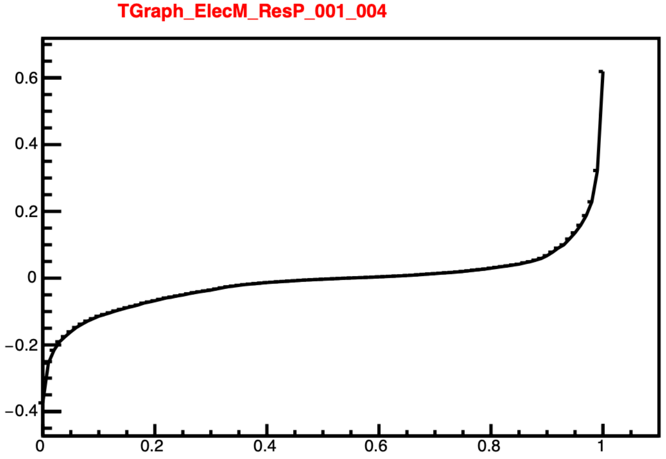
μ^-径迹重建效率

e^-在ECAL中的重建效率



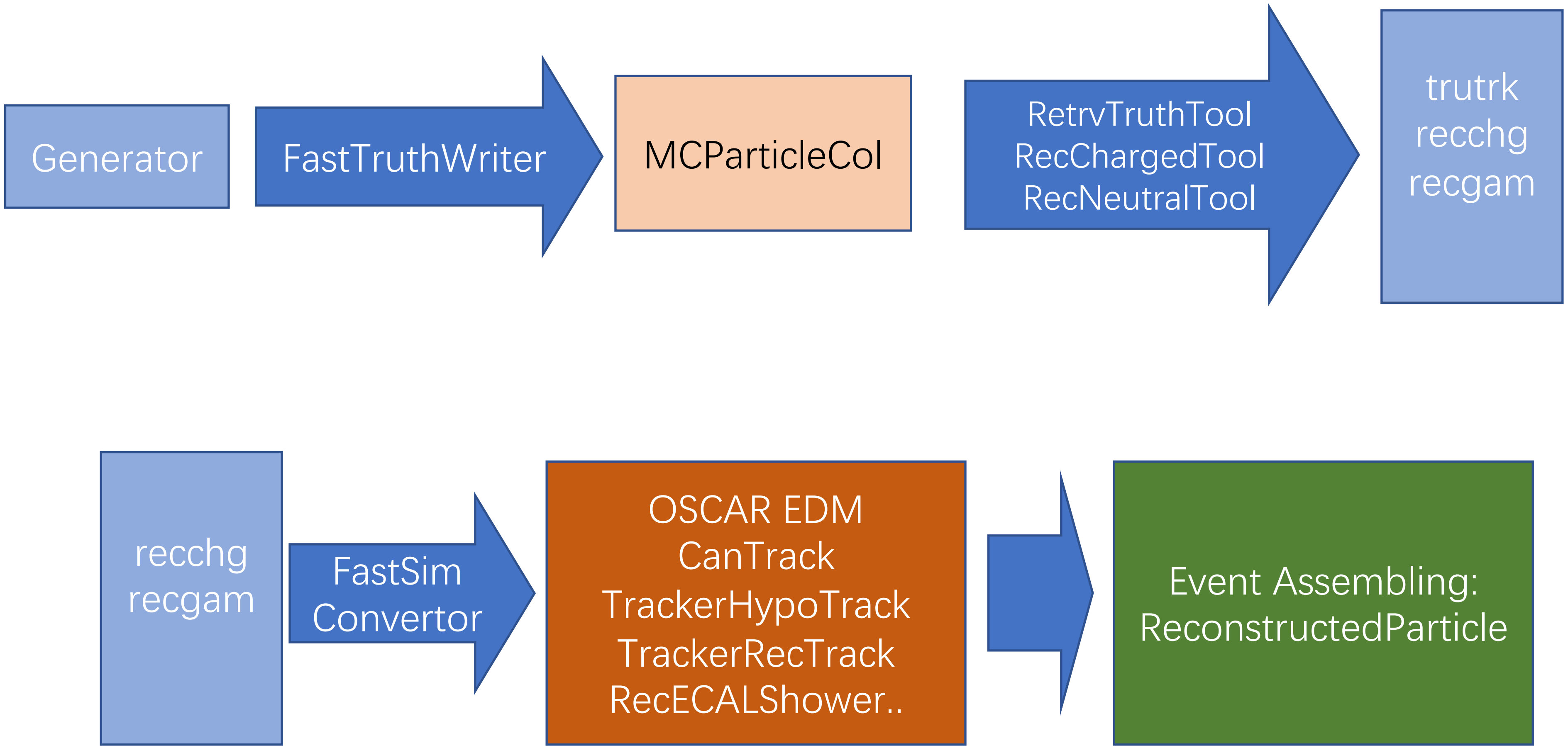






数据流

- Track and shower assembling
- Event assembling



快速模拟算法简介

- **$OFFLINETOP/Simulation/FastSim/**

- **算法包: $OFFLINETOP/Simulation/FastSim/**
  - **FastTruthWriter**:
    通过产生子给出的信息构建MCParticleCol
  - **FastSim**:
    快速模拟的主算法

- Analysis/FastValidAlg 简单的辅助算法
  快速查看快模拟实现了哪些类的具体数据

```
import FastValidAlg
alg = task.createAlg("FastValidAlg")
alg.property("check").set(True)
alg.setLogLevel(4)
```

算法功能分解

- FastTruthWriter
  - 通过产生子产生的HepMC::GenEvent构建MCParticle
  - 可添加Boost
- FastSim
  - FastSim 待用Tool，最后封装成TrackerRecTrack,TrackerHypoTrack,RecECALShower,ReconstructedParticle等数据
  - RetrvTruthTool
    - 从MCParticle抽取信息构建结构体trutrk, recchg, recgam
  - RecNeutralTool
    - 实现中性粒子的高级信息抽样
  - RecChargedTool
    - 实现带电粒子的高级信息抽样
- Analysis/FastValidAlg 简单的辅助算法
  - 快速查看快模拟实现了哪些类的具体数据

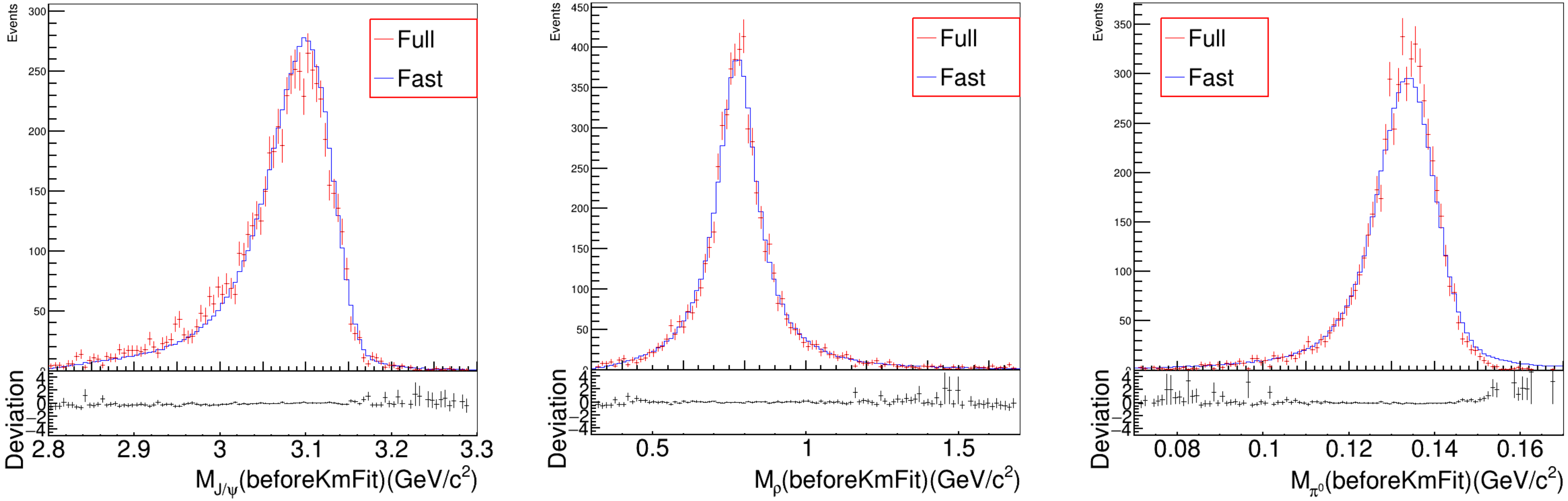
快速模拟样本产生

- $OFFLINETOP/Simulation/FastSim/share/fastsim.rhopi.py

```
#****************** FastSimulation
import FastSim
fasttw= task.createAlg("FastTruthWriter")
#fasttw.property("Boost").set(False)  // switch off boost if you need
fastsim = task.createAlg("FastSim")
```

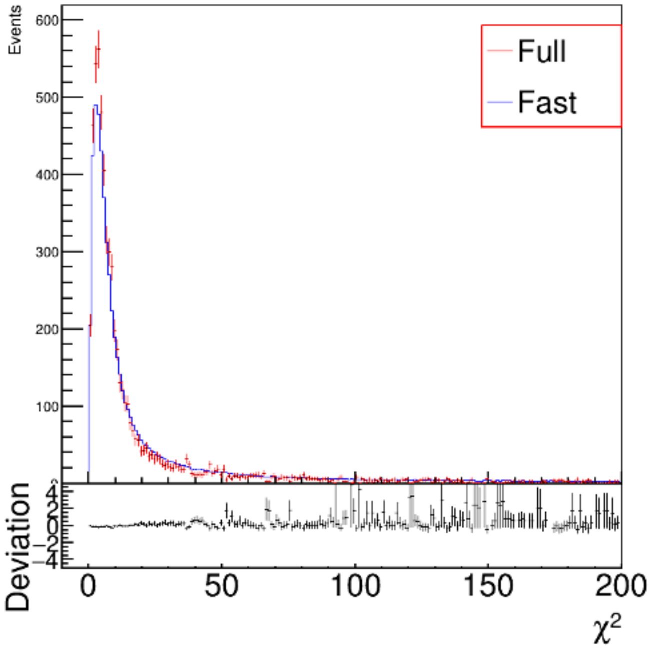
快速模拟性能验证

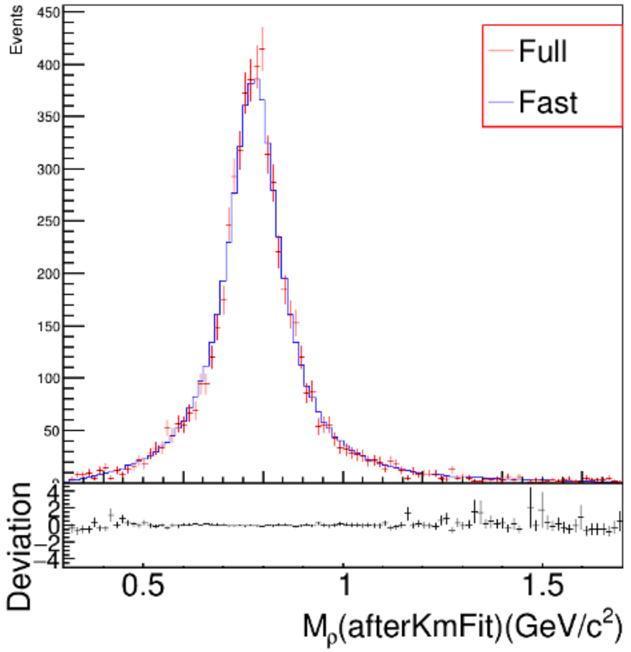
- 根据STCF探测器响应性能进行抽样和参数化，并可灵活配置以开展预研究
- 通过多个衰变道进行性能测试：$\rho\pi^0, \pi\pi J/\psi, \Lambda\bar{\Lambda}...$
- 速度提升和存储空间压缩：
  - ~2ms/事例, J/ψ四径迹末态事例 ~2kb 存储空间

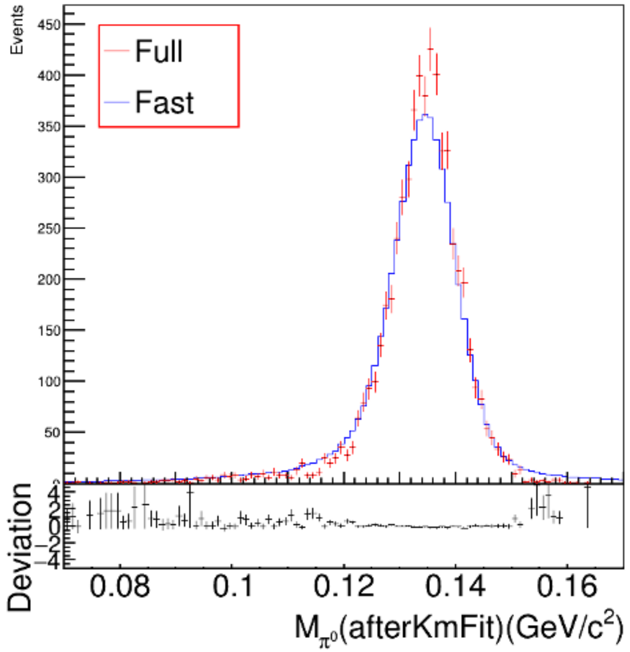


快速模拟性能验证

- 根据STCF探测器响应性能进行抽样和参数化，并可灵活配置以开展预研究
- 通过多个衰变道进行性能测试：$\rho\pi^0, \pi\pi J/\psi, \Lambda\bar{\Lambda}...$
- 速度提升和存储空间压缩：
  - ~2ms/事例, J/ψ四径迹末态事例 ~2kb 存储空间







与全模拟的差别

- 没有探测器几何，没有Geant4实现粒子输运
  - 没有击中信息
  - 没有衰变信息
  - 没有电子对产生（Gamma Conversion）
  - 没有ExtTrack，没有径迹的外推信息
    - 带电粒子对应的中性径迹已经关联好
- 没有子探测器的测量量，比如dE/dx, RICH likelihood, MUD likelihood (有ECAL沉积能量)
- 尚未添加噪声
- PID: 有PID的判选结果，对应的有虚构的概率(0或1)

注意事项和应用场景

- 其他注意事项
  - 没有HitCollection
  - ExtTrackCollection为空，没有径迹外推至各个子探测器的信息
  - 如果因为进一步优化的需求如进一步压缩大小，则可能导致数据模型与全重建不一致，会造成用户分析算法上的差别

- 应用场景：
  - 服务于样本统计量需求非常大的物理预研究
  - 对探测器响应性能的快速配置和设计

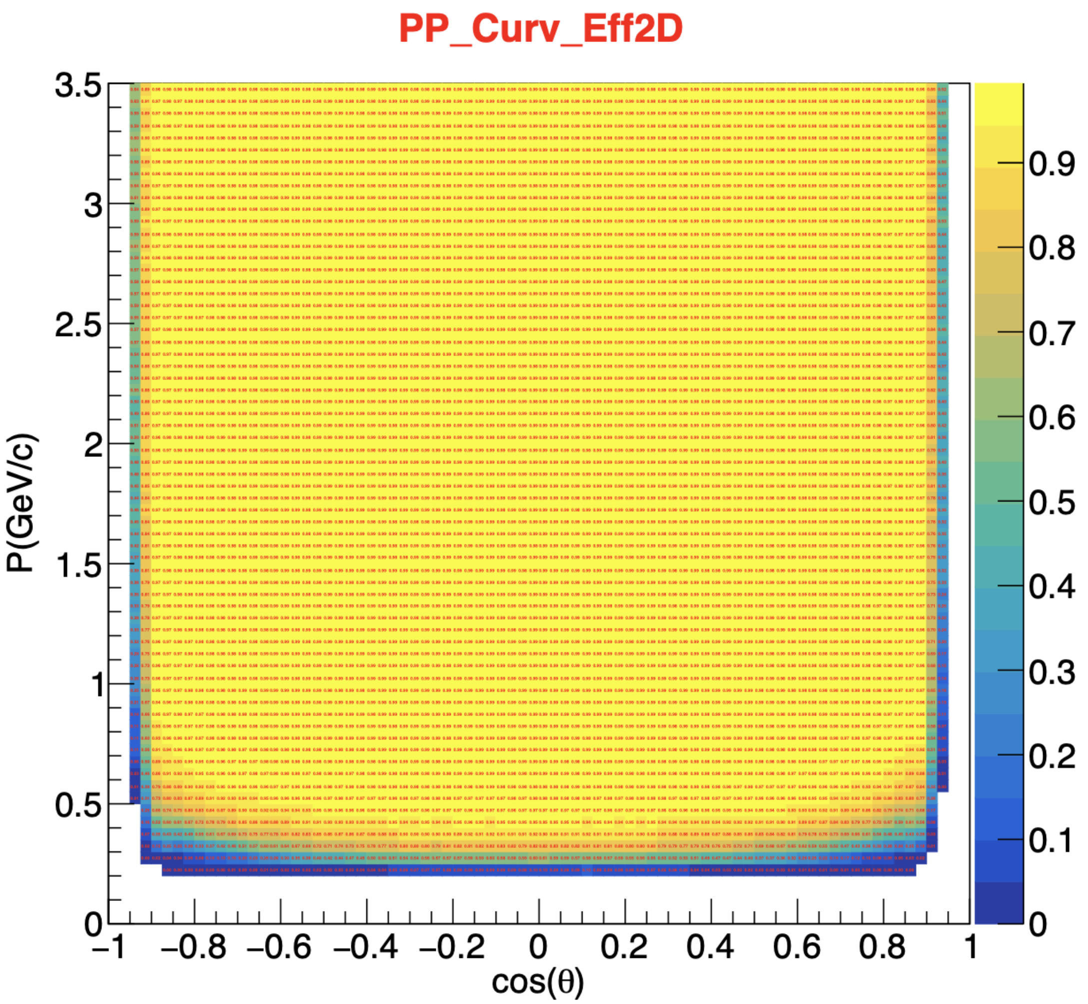
快模拟信息：带电粒子

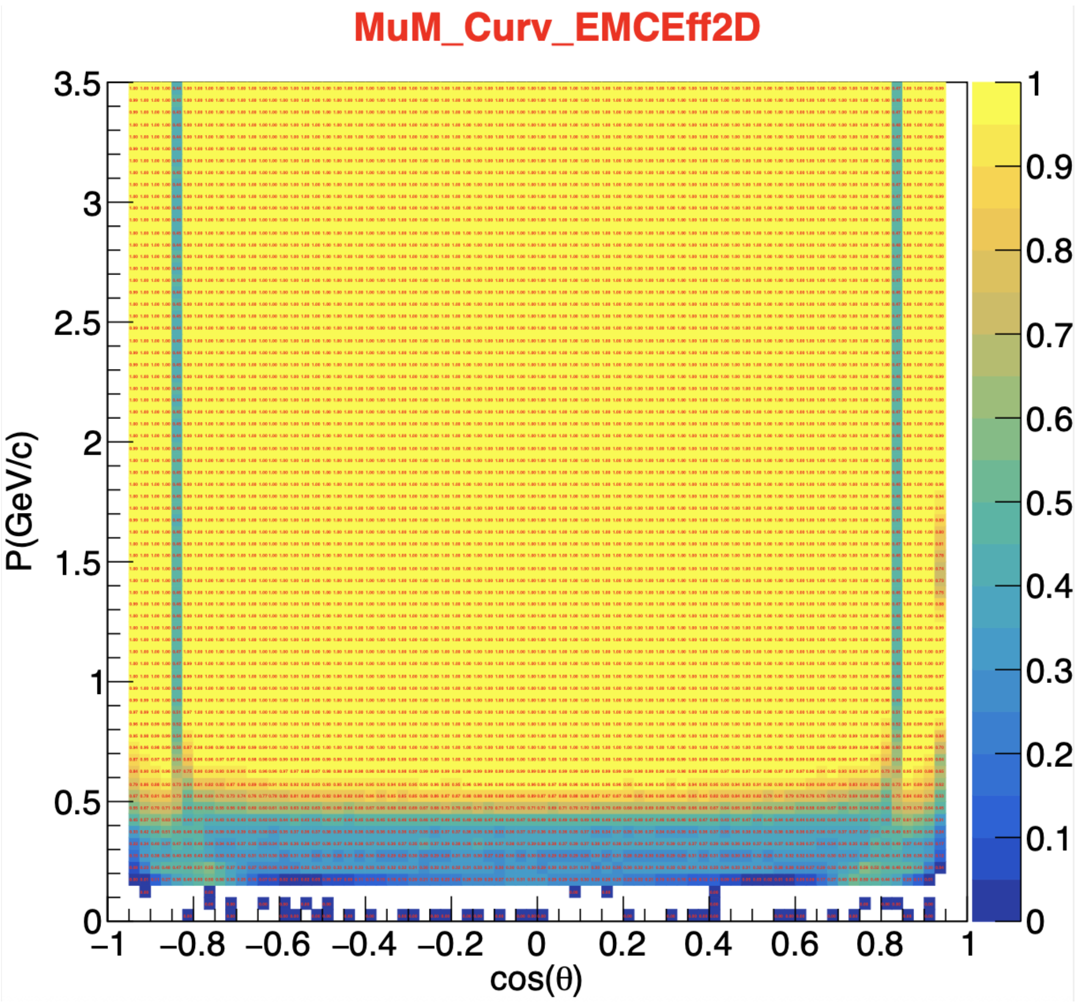
- 快模拟信息：
  - MDC: $p, V_r, V_z, \theta, \phi$;径迹参数及误差矩阵; 径迹重建效率
  - ECAL: E_{Ecal}; shower重建效率
  - PID

- 抽样变量：
  - MDC: $\frac{\Delta p}{p} = \frac{p_{rec}-p_{truth}}{p_{truth}}$
  - ECAL: $E = \frac{E_{rec}}{E_{truth}}$
  - $\Delta\theta = \theta_{rec} - \theta_{truth}$
  - $\Delta\phi = \phi_{rec} - \phi_{truth}$
  - 效率
  - PID prob

带电粒子效率

- ◆径迹重建效率
- ◆shower重建效率





helix参数与误差矩阵

- helix 6参数 (x, y, z, p_x, p_y, p_z)
  - pos:产生点位置在truth附近根据Vr, Vz 进行smear
  - momentum:动量大小、角度smear
  - 根据position, momentum构建helix，径迹六参数使用POCA位置的参数
- 误差矩阵
  - 三维分bin：$f(p, \cos(\theta), \phi)$
  - 非对角元按照与对角元的比值进行抽样
  - $V_{02}/\sqrt{V_{00} * V_{22}}$

次级带电粒子的径迹参数

- helix 6参数 (x, y, z, p_x, p_y, p_z)
  - pos:产生点位置在truth附近根据Vr, Vz 进行smear
  - momentum:动量大小、角度smear
  - 根据position, momentum构建helix，径迹六参数使用POCA位置的参数
- 误差矩阵
  - 误差矩阵处理与IP点产生粒子一致
  - 尝试过V_{00}, V_{11}, V_{22}简单的倍数化操作，效果不理想

粒子鉴别

- 按粒子鉴别为各种粒子的效率总和归一，按0-1随机抽样，判定为某种粒子类型，并将其prob设置为1，其他粒子假设的prob设置为0

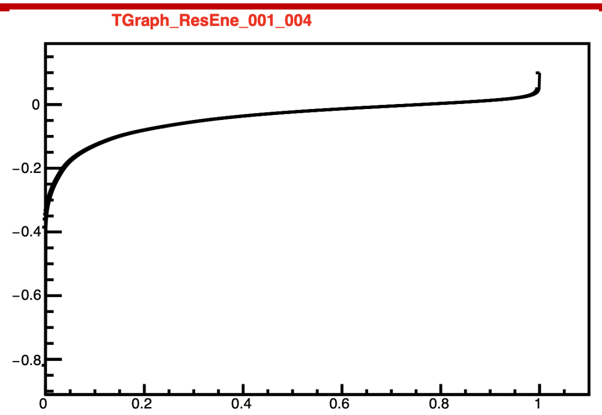
中性粒子-光子

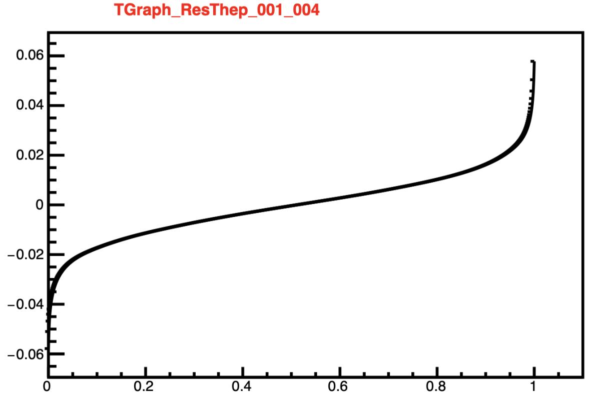
◆能量、角度、效率

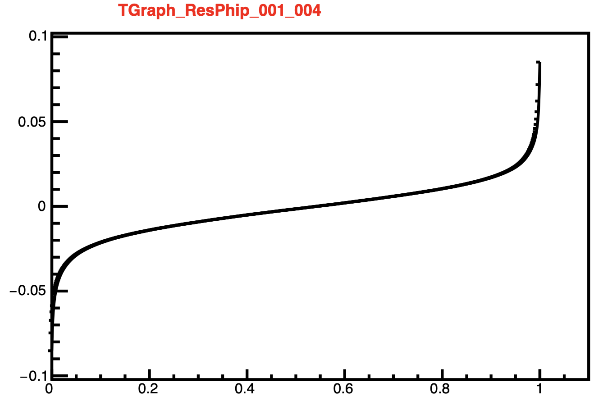
- $\frac{\Delta E}{E} = \frac{E_{rec} - E_{truth}}{E_{truth}}$
- $\Delta\theta = \theta_{rec} - \theta_{truth}$
- $\Delta\phi = \phi_{rec} - \phi_{truth}$
- 位置按简单的圆柱面近似

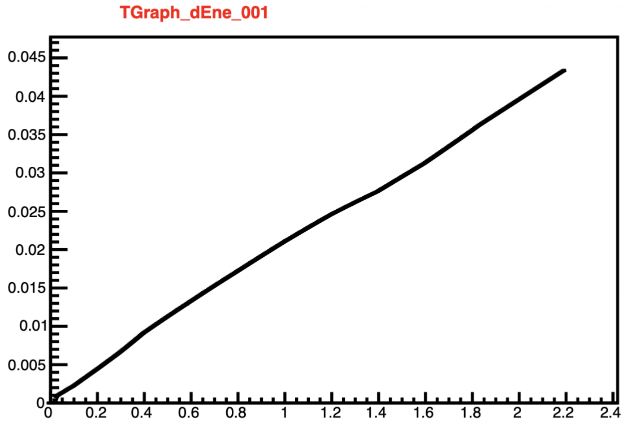
◆误差:

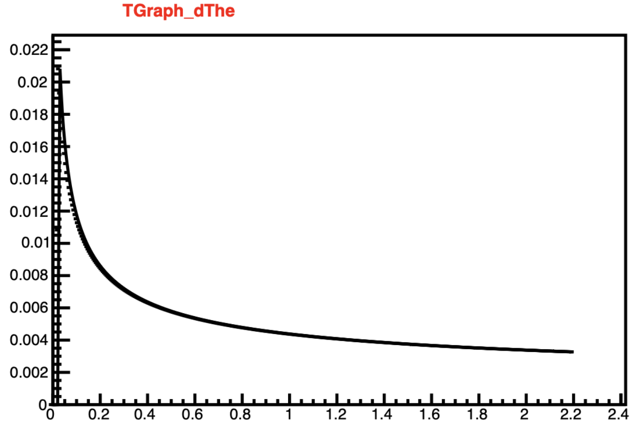
- dE 不同角度bin内抽取CDF
- $d\theta = \frac{p_1}{E} + \frac{p_2}{\sqrt{E}} + p_3$
- $d\phi = \frac{p_1'}{E} + \frac{p_2'}{\sqrt{E}} + p_3'$

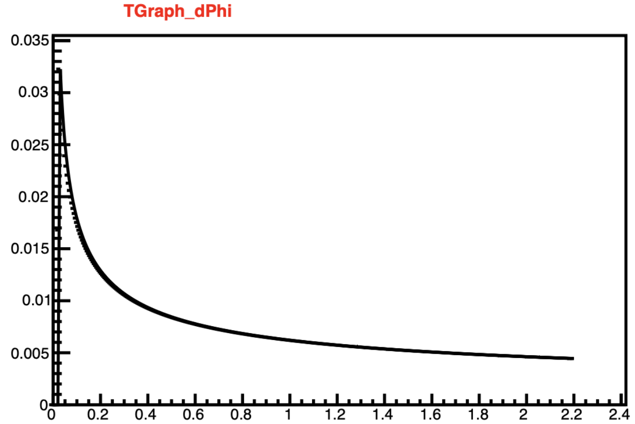








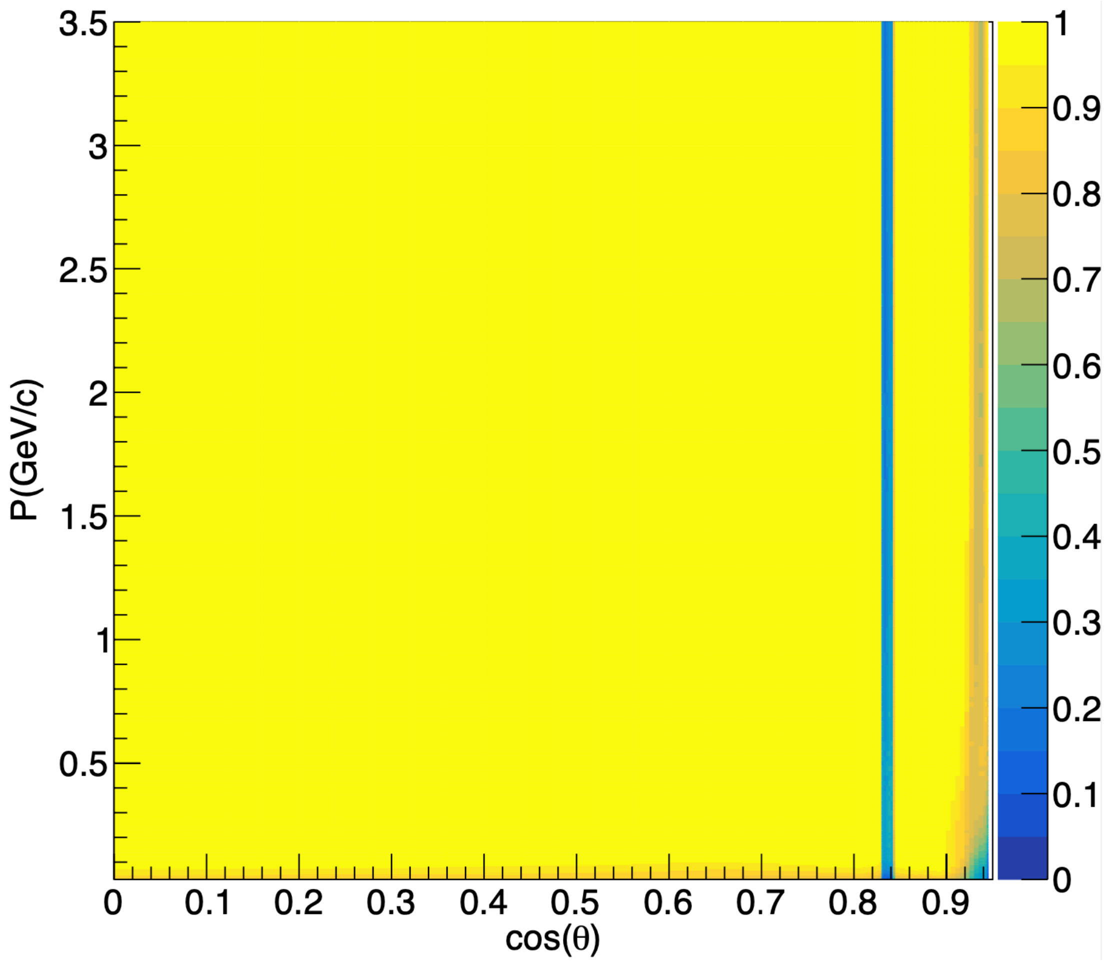




Shower其他信息

- recstat: /// 1, killed by efficiency, 3 by acception, 4 by energy,

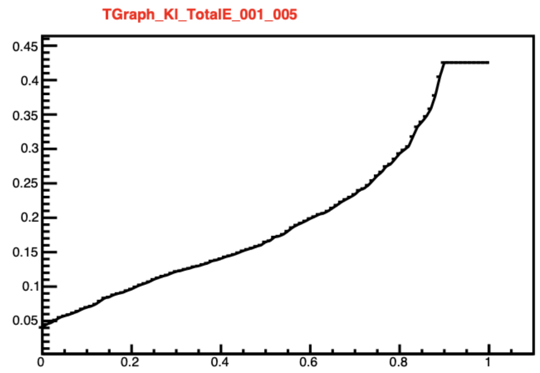
  <0 problem!, 0-ok
- rectime=7
- recnumhits 29 for γ, 31 for $\bar{N}$.
- emcpos: ECAL简化的圆柱面结合shower从IP引出的方向给出位置
- 效率不均匀分bin，在低能和端盖区分bin较多

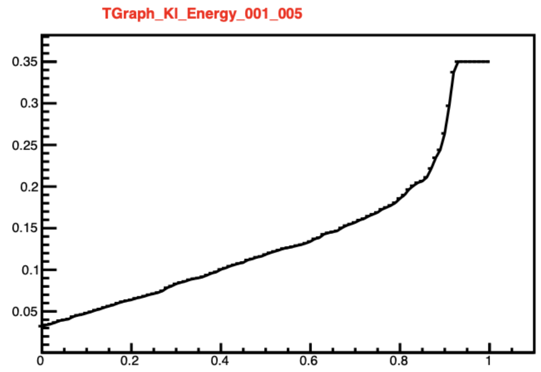


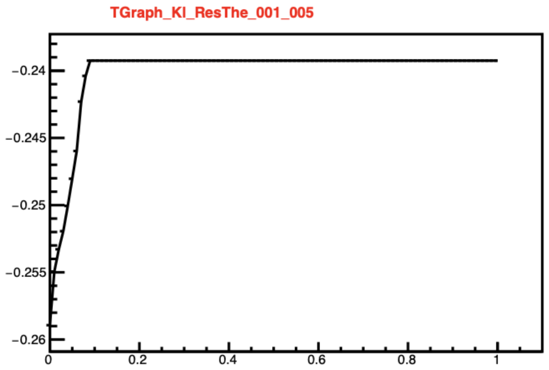
中性粒子：$K_L, \bar{N}$

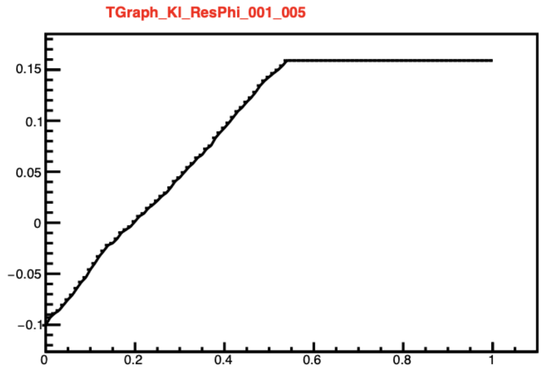
- K_L :能量、总沉积能量、角度、效率
  - E_{Ecal}, E_{total}
  - $\Delta\theta = \theta_{rec} - \theta_{truth}$
  - $\Delta\phi = \phi_{rec} - \phi_{truth}$

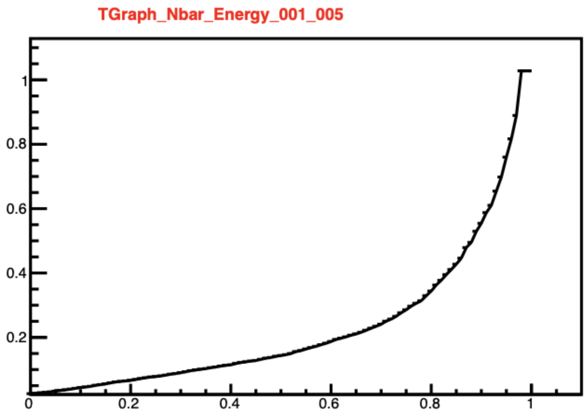
- $\bar{N}$:能量、角度、效率
  - E_{Ecal}
  - $\Delta\theta = \theta_{rec} - \theta_{truth}$
  - $\Delta\phi = \phi_{rec} - \phi_{truth}$

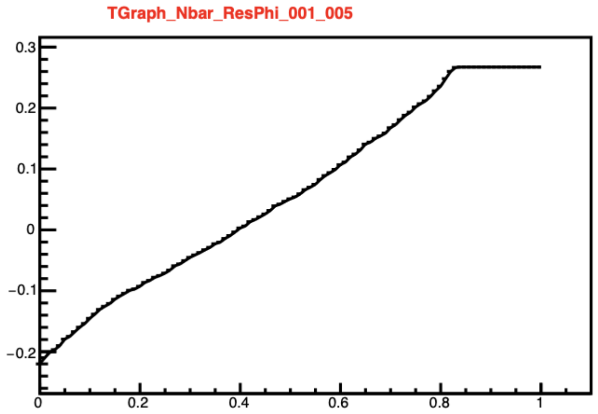


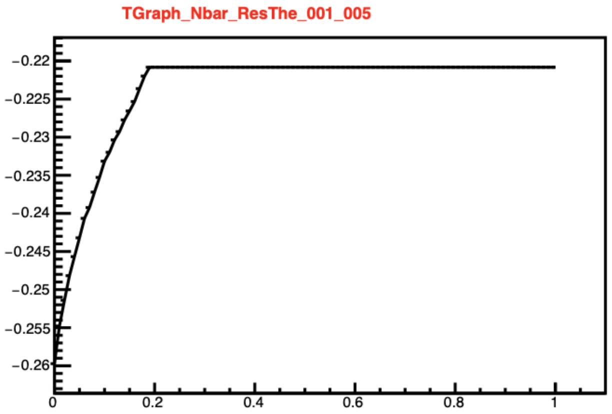










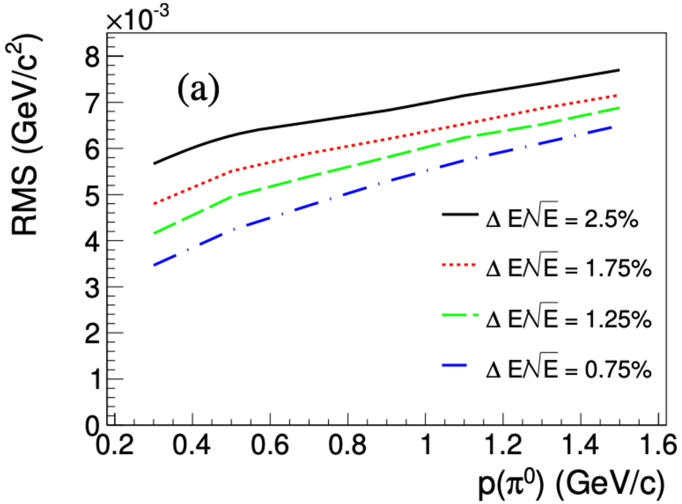


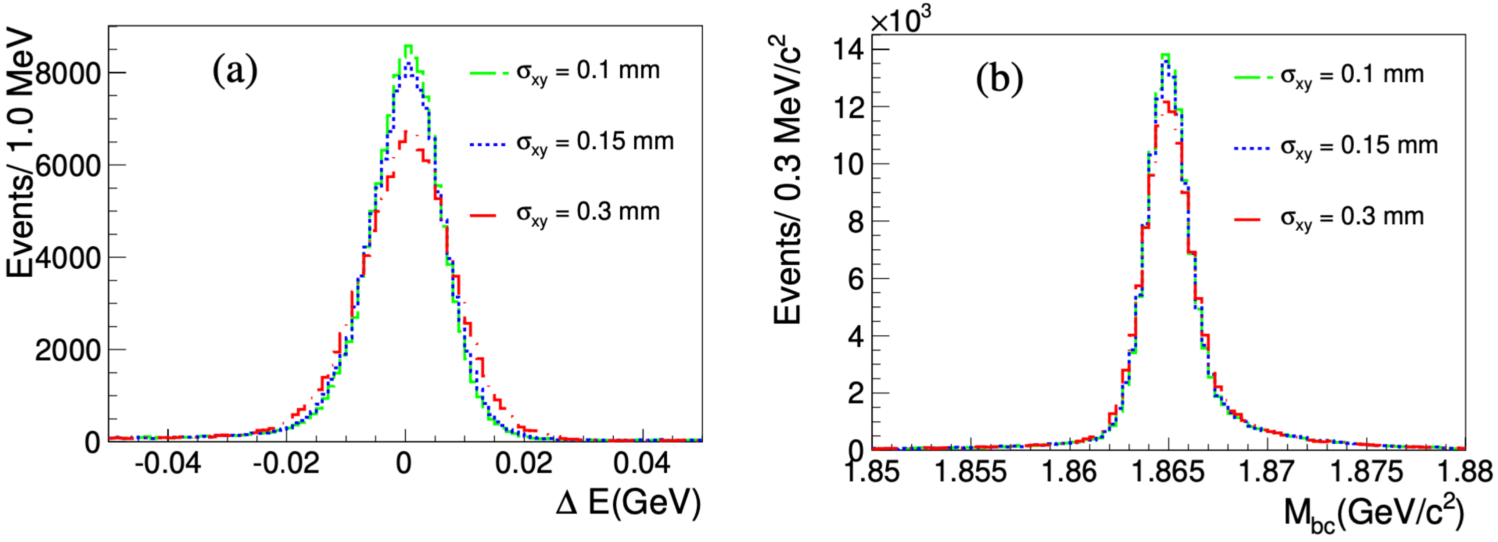
关于探测器性能的scaling

- 中性粒子 光子 能量、θ角度、ϕ角度、效率
- 带电粒子 V_r, V_z

```
declProp( "gamEneScale" , par_gamEneScale=1.0 );
declProp( "gamTheScale" , par_gamTheScale=1.0 );
declProp( "gamPhiScale" , par_gamPhiScale=1.0 );
declProp( "gamRecEffScale" , par_gamRecEffScale=1.0 );
declProp( "chgSigXY"          , par_chgSigXY=-1 );    /
declProp( "chgSigZ"           , par_chgSigZ=-1 );
declProp( "chgRecEffScale" , par_chgRecEffScale=1.0 );
declProp( "chgVrScale"   , par_chgVrScale=1.0 );
declProp( "chgVzScale"   , par_chgVzScale=1.0 );
declProp( "vtxRadScale" , par_vtxRadScale=1.0 );
declProp( "vtxZdrScale" , par_vtxZdrScale=1.0 );
```

- 旧版本示意图 *JINST* **16** P03029





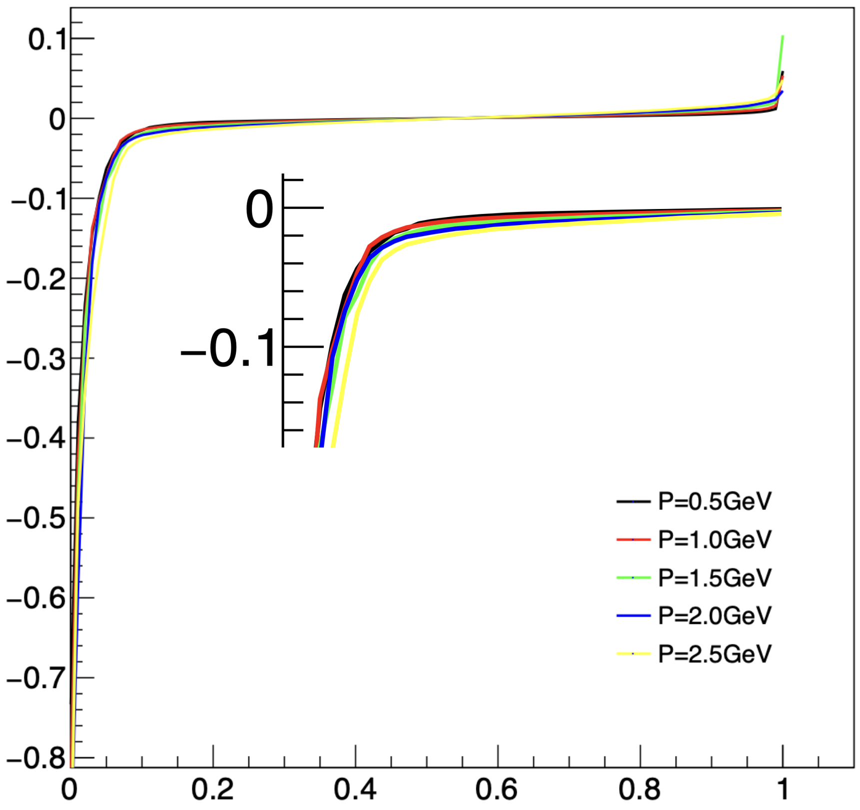
总结

- ◆基于抽样方法实现了快模拟功能
- ◆应用于TDR物理研究和探测器性能预研

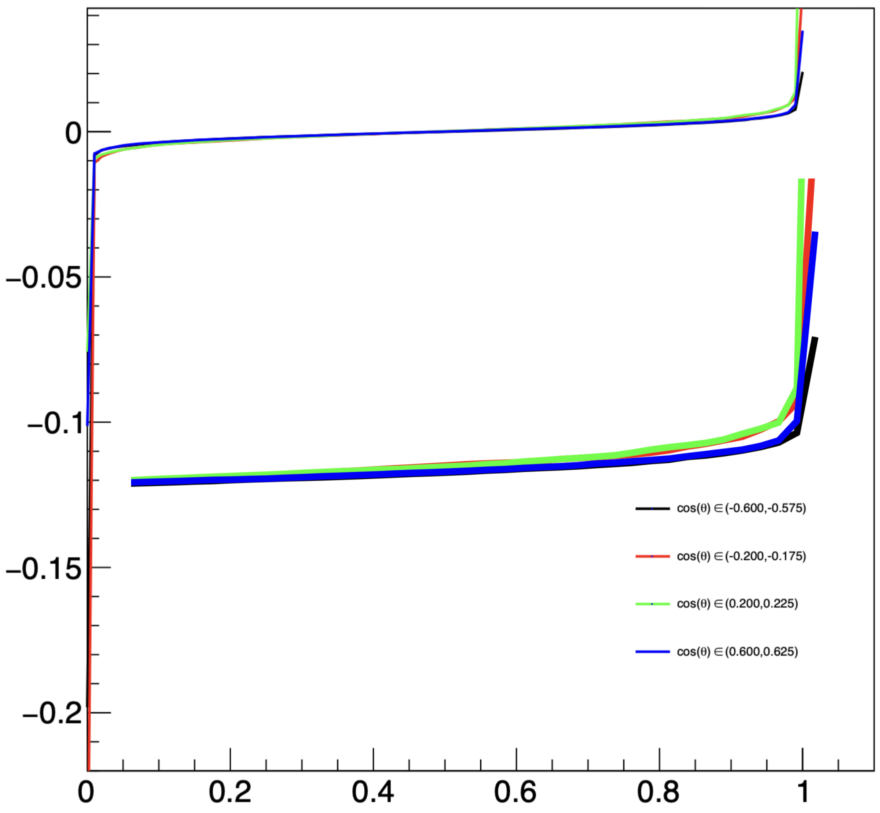
- ◆后续工作
  - 随Oscar版本和探测器版本的常态更新
  - scaling与探测器优化
  - 次级粒子径迹参数等
  - 事例大小和内存优化

谢谢！

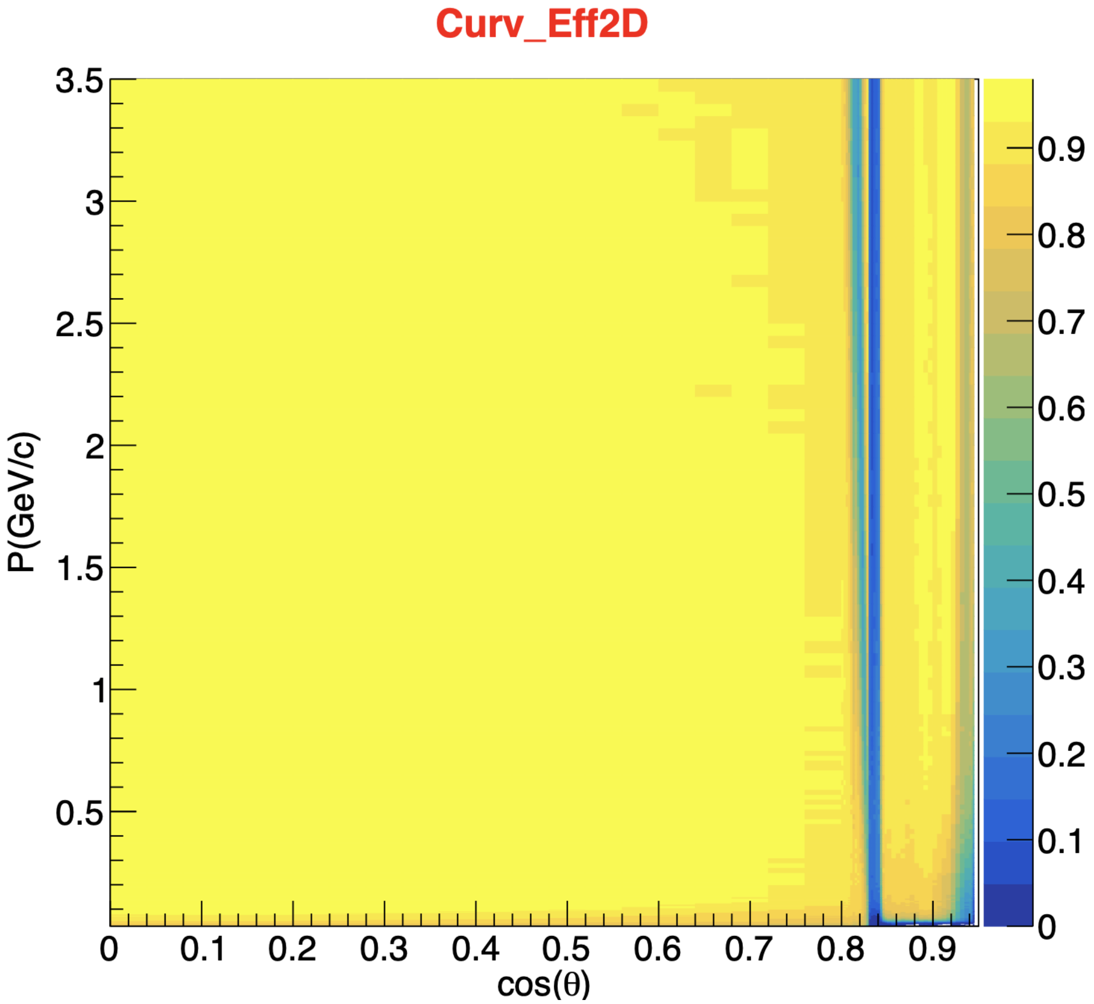
backup

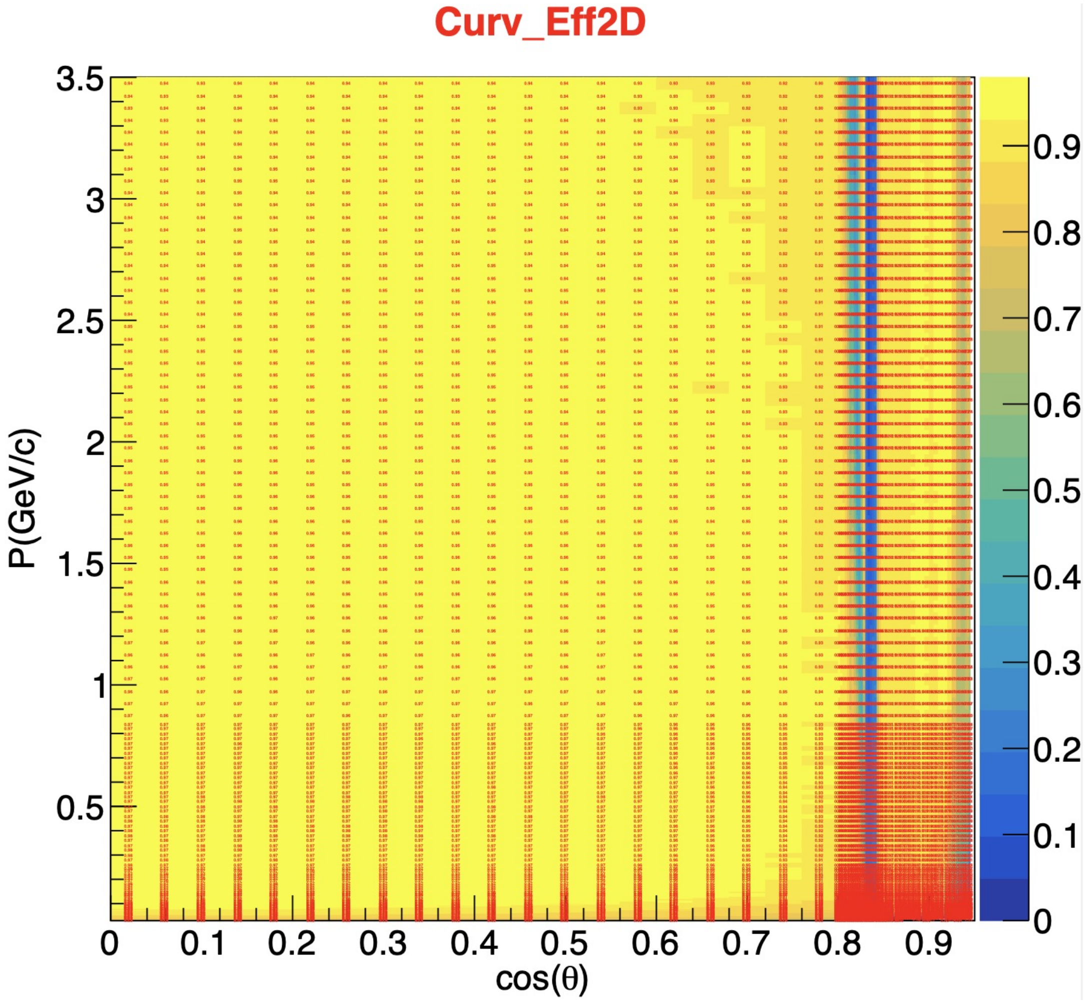


0.1
0
−0.1
−0.2
−0.3
−0.4
−0.5
−0.6
−0.7
−0.8
0
0.2
0.4
0.6
0.8
1
0
−0.1
P=0.5GeV
P=1.0GeV
P=1.5GeV
P=2.0GeV
P=2.5GeV



0
−0.05
−0.1
−0.15
−0.2
0
0.2
0.4
0.6
0.8
1
cos(θ) ∈ (-0.600,-0.575)
cos(θ) ∈ (-0.200,-0.175)
cos(θ) ∈ (0.200,0.225)
cos(θ) ∈ (0.600,0.625)


Curv_Eff2D
P(GeV/c)
cos(θ)
3.5
3
2.5
2
1.5
1
0.5
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9


Curv_Eff2D
P(GeV/c)
cos(θ)
3.5
3
2.5
2
1.5
1
0.5
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9

FastSim 快速模拟主算法

- ◆FastSim算法类
- ◆RetrvTruthTool: 读取MCParticleCol，并构建带电粒子、中性粒子的truth信息(结构体)
- ◆分别调用RecChargedTool, RecNeutralTool完成效率、分辨的抽样(快速模拟)
- ◆FastSim::evtAssemble() 组装成ReconstructedParticle

```
FASTSIM
  include
    constant.h
    FastSim.h
    Helix.h
    MyUtil.h
    RecChargedTool.h
    RecNeutralTool.h
    RetrvTruthTool.h
```

```cpp
class FastSim: public AlgBase{
    FastSim(const std::string& name);
    ~FastSim();
    bool initialize();
    bool execute();
    bool finalize();
    bool copyMCParticles();
  private:
    void debug( bool tmp=false ) { m_debugSim=tmp; }
    void clearVector();
    bool clearTDSCollection();
    bool regMdcTracks();
    bool regECALShowers();
    bool regMUDTracks();
    bool evtAssemble();
    void setCanTrack( MutableCanTrack&, recchg& );
    void setTrackerRecTrack( MutableTrackerRecTrack&, recchg& )
    void setRecECALShower( MutableRecECALShower&, recgam& );
    void setRecChgECALShower( MutableRecECALShower&, recchg& );
    void setRecChgMUDTrack( MutableMUDTrack&, recchg& );
```

FastSim 快速模拟主算法

- ◆RecChargedTool: 带电径迹重建 e, μ, π, K, p
- ◆RecNeutralTool: 中性粒子重建 γ , K_L, Nbar

```
class RecNeutralTool : public ToolBase{
    void setRunEvtNum( int run, int evt
    void setResScales( double ene=1.0,
    { m_scaleResEne=ene; m_scaleResThe=

    /// do the neutral reconstruction;
    bool recGamEmcShower( trutrk&, recga
    int         effGamEmcShower( double,
    //bool resGamEmcShower( std::string,
    bool resGamEmcShower( int, double,
    //bool resGamEmcShower( std::string,

    /// get the function;
    double      gamEneError( double, dou
```

```
class RecChargedTool: public ToolBase{
    /// do the charged reconstruction;
    bool recChargedTrk( trutrk, recchg&
    /// int     enebinGamShower( HepLore
    /// int     catgryGamShower( HepLore
    int   effChgMDCTrack( double, double
    int   PIDChgTrack( double, double, d
    int   STCFPIDChgTrack( double, doubl
    int   PIDMucChgTrack( double, double
    int   PIDSTCFMucChgTrack( double, do
    double EMCChgTrack( double, double,
    bool RecHelix(HepPoint3D, double, d
    bool resChgTrack( std::string, doub
    bool resChgTrack( std::string, doub
    bool HelixErrChgTrack( double, std:
```

FastSim 快速模拟主算法

- FastSim算法:
  - 实现模拟的能动量范围：0.03-3.5GeV
  - $e, \mu, \pi, K, p, \gamma\ , K_L, \bar{N}$

- 重建性能的快速设置，便于探测器设计的预研究

```
declProp( "gamEneScale" , par_gamEneScale=1.0 );
declProp( "gamTheScale" , par_gamTheScale=1.0 );
declProp( "gamPhiScale" , par_gamPhiScale=1.0 );
declProp( "gamRecEffScale" , par_gamRecEffScale=1.0 );
declProp( "chgSigXY"            , par_chgSigXY=-1 );   // sigma_xy from MDC, unit: mum
declProp( "chgSigZ"             , par_chgSigZ=-1 );     // sigma_z from MDC, unit: mum
declProp( "chgRecEffScale" , par_chgRecEffScale=1.0 );
declProp( "chgVrScale"   , par_chgVrScale=1.0 );
declProp( "chgVzScale"   , par_chgVzScale=1.0 );
declProp( "vtxRadScale" , par_vtxRadScale=1.0 );
declProp( "vtxZdrScale" , par_vtxZdrScale=1.0 );
```

FastValidAlg 简单的辅助算法

◆利用输出算符重载快速查看数据信息

- std::ostream& operator<<(std::ostream& o, const ReconstructedParticle& value)

```
for (size_t i=0; i<m_recParticle->size(); ++i) {
  auto recpar = m_recParticle->at(i);
  auto mcpar =recpar.getMCParticle();
  auto momentum = recpar.getMomentum();
  cout<<"preMc,"<<count[0]<<","<<momentum<<"," \
    <<i<<","<<m_recParticle->size()<<endl;
  if(!mcpar.isAvailable()) continue;
  cout<<"RecPar[----------------"<<endl;
  cout<<recpar<<endl;
  cout<<"RecPar----------------]"<<endl;
  validMC=1;
  auto track = recpar.getTrack();
  auto shower = recpar.getShower();
  if(track.isAvailable()) {
    cout<<"TrackerRecTrack[-------"<<endl;
    cout<<track<<endl;
    cout<<"TrackerRecTrack-------]"<<endl;
    auto hypo = track.getTrackHypos(2);
    if(!hypo.isAvailable()) continue;
    cout<<"TrackerHypoTrack[------"<<endl;
    cout<<hypo<<endl;
    cout<<"TrackerHypoTrack------]"<<endl;
  }
```

全重建样本

```
RecPar[----------------
 id: 300000000
 trackID : 0
 energy : 0
 momentum : 0.753494 1.10499 0.583928
 referencePoint : 0 0 0
 charge : 1
 mass : 0
 MCParticle : 20000012
 RICHPid : 230000000
 track : 110000000
 dEdX : 140000000
 shower : 180000004
 mudTrack : 270000000
 mudCluster : 4274967294
```

快模拟样本

```
RecPar[----------------
 id: 90000000
 trackID : 0
 energy : 0
 momentum : 0.0756972 -0.0368283 -0.186064
 referencePoint : 10000 10000 10000
 charge : 1
 mass : 0
 MCParticle : 20000008
 RICHPid : 4274967294
 track : 40000000
 dEdX : 4274967294
 shower : 70000000
 mudTrack : 60000000
 mudCluster : 4274967294
```