Machine Learning for Heavy-Flavor Reconstruction

Shyam Kumar*, Annalisa Mastroserio, Domenico Elia

INFN Bari, Italy

Supported by The FAIR Spoke 6 Project, funded by the NextGenerationEU program in Italy

Future Artificial Intelligence Research

Symbiotic AI

[hipe4ml package for ML](#)

Ref: https://doi.org/10.5281/zenodo.5070131

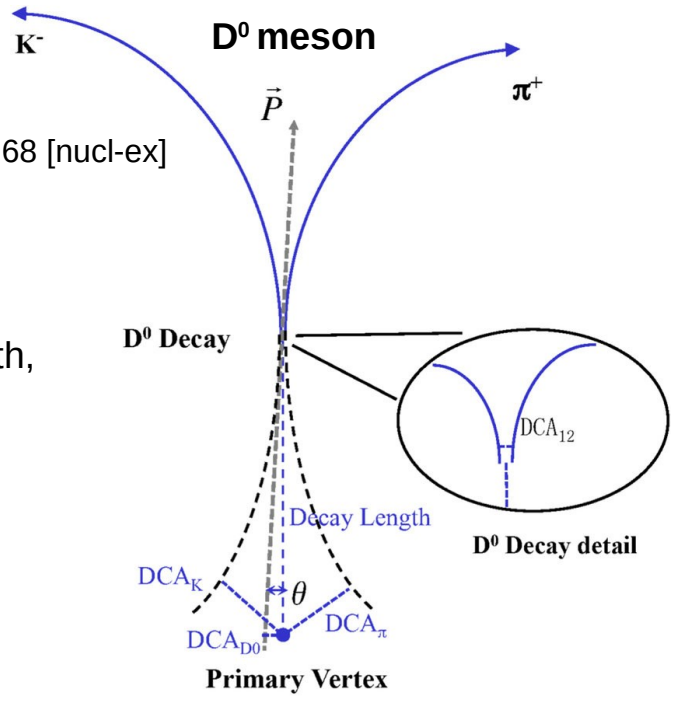
Primary Vertex Reconstruction:

- **CentralTrackVertices**

arXiv:1911.12168 [nucl-ex]

Primary vertex: allows access to the DCA of daughter tracks

Secondary vertex: allows access to more topological variables, e.g., decay length, pointing angle (cosθ), DCA_{D0} or $DCA_{\Lambda c}$ etc.



Topological Variables

$$\vec{dl} = \vec{SV} - \vec{PV}$$

$$\cos\theta = \frac{\vec{dl} \cdot \vec{p_{D0}}}{|\vec{dl}||\vec{p_{D0}}|}$$

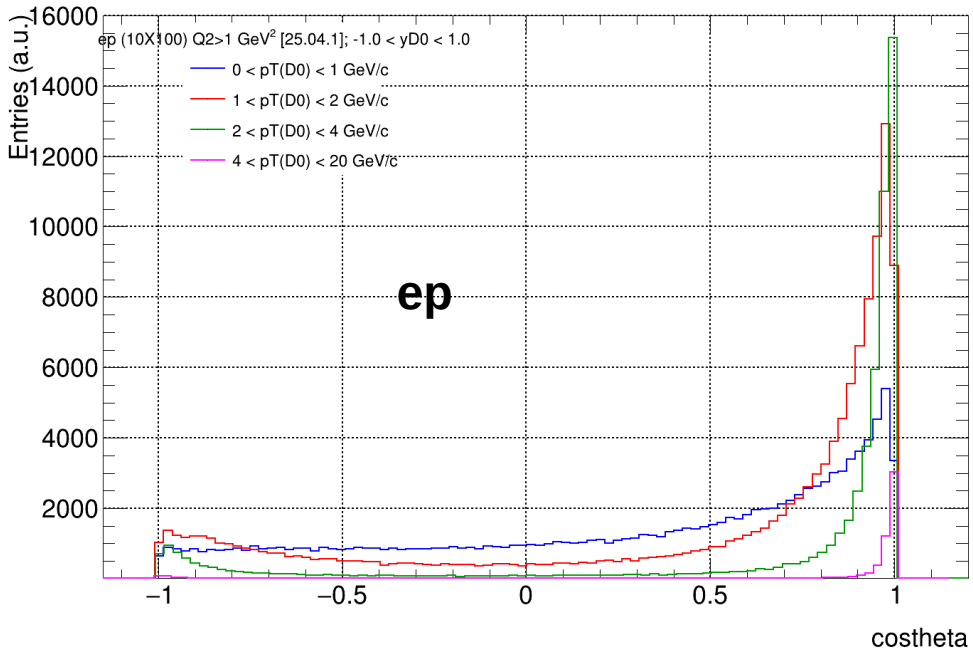
$$DCA_{D0} = |\vec{dl}| \sin\theta$$

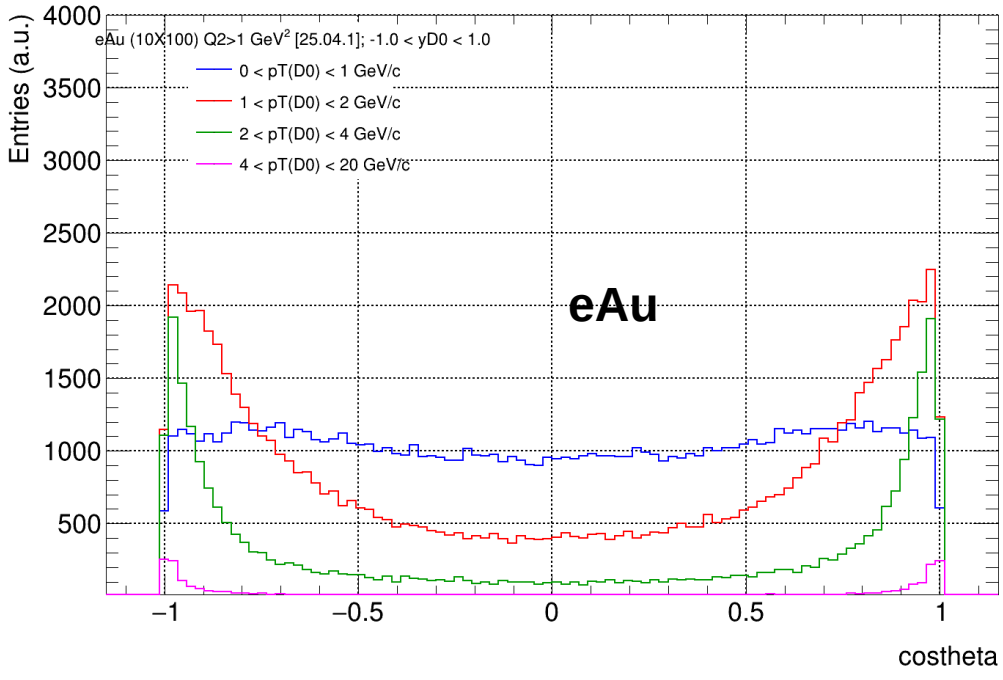
Invariant mass: $m_{D^0} = \sqrt{(E_{K^-} + E_{\pi^+})^2 - (\vec{p_{K^-}} + \vec{p_{\pi^+}})^2}$

Cosθ: Angle between **dl** vector and $\mathbf{p_{D0}}$

Update on Cosine Pointing Angle

Signal pair: From D^0 decay





For ep: In some cases, Cosθ < 0 because of **dl <0**

Secondary vertex is reconstructed before primary vertex due to poor resolution on vertices

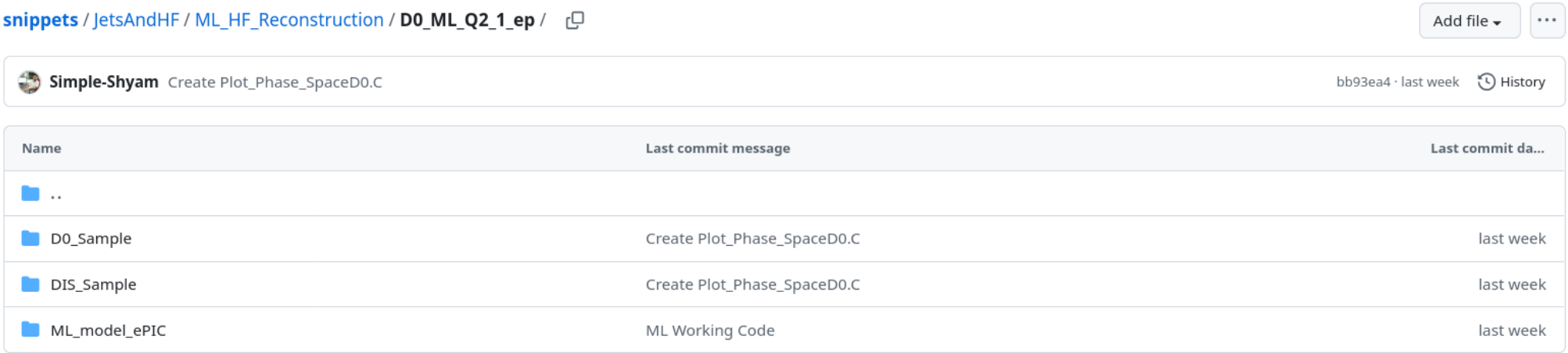
For eAu: ~50% cases, Cosθ < 0 because of **dl <0**

This is a bug because of D^0 decay vertex is at the primary vertex: Still fix going on

ML Code for HF Studies

ML algorithm: BDT Binary Classifier

ML_HF_Reconstruction



Put the root files from analysis code in D0_Sample and DIS_Sample directory

Script produces all the plots which I usually show in the presentation

Working Code with existing root files

If some package missing try to install pip install package name

Main Script

snippets / JetsAndHF / ML_HF_Reconstruction / D0_ML_Q2_1_ep / ML_model_ePIC / **Run_ML_withData.sh**

Simple-Shyam Update and rename Run_ML_Shyam_withData.sh to Run_ML_withData.sh

Do **source Run_ML_withData.sh**

Code | Blame | 40 lines (38 loc) · 2.08 KB

```
#!/bin/bash
#script to run the machine learning differential in pT and y
# Shyam Kumar; INFN Bari, Italy;
# shyam.kumar@ba.infn.it; shyam055119@gmail.com
# Supported by The FAIR Spoke 6 Project, funded by the NextGenerationEU program in Italy
rm *.png
prepare_sample=true
preselection_cuts='(mass_D0 > 1.6 && mass_D0 < 2.5) && (d0xy_pi>0.02 && d0xy_pi<10.) && (d0xy_k>0.02 && d0xy_k<10.) && decay_length <100.'
nEvents_D0Sample=1.747e+9 # Simulated ~1747M
nTotalEvents=4.72162e+9 #10fb^-1 (ep)
y_arr_ML=(-1.0 1.0 3.0)
pt_arr_ML=(0.0 1.0 2.0)

if $prepare_sample; then
find . -type f \( -name "*.root" -o -name "*.png" -o -name "*.txt" \) -delete # delete all root, txt, png files in directory
cd Data_Preparation/Filtered_D0Sample/
root -b -l -q Create_Signal_with_Cuts.C'("'"$preselection_cuts"'")'
root -b -l -q Create_bkg_with_Cuts.C'("'"$preselection_cuts"'")'
cd ../Filtered_DISSample
root -b -l -q Create_Signal_with_Cuts.C'("'"$preselection_cuts"'")'
root -b -l -q Create_bkg_with_Cuts.C'("'"$preselection_cuts"'")'
cd ../../
# Signal and Backgrounds for ML
hadd Data_Preparation/SignalD0.root  Data_Preparation/Filtered_D0Sample/SignalD0.root Data_Preparation/Filtered_DISSample/SignalD0.root
cp Data_Preparation/Filtered_DISSample/BkgD0.root Data_Preparation/
fi
# Prepare data using resampling and applying proper scaling factor
cd Data_Preparation/Merge_Data/
source Merged_Data.sh "$nEvents_D0Sample" "$nTotalEvents"
cd ../../
# RunML differential in y and pT
for ((i=0; i<${#y_arr_ML[@]}-1; i++)); do
for ((j=0; j<${#pt_arr_ML[@]}-1; j++)); do
rm -rf ML_Output_Optuna_${y_arr_ML[i]}_${y_arr_ML[i+1]}_${pt_arr_ML[j]}_${pt_arr_ML[j+1]} && mkdir ML_Output_Optuna_${y_arr_ML[i]}_${y_arr_ML[i+1]}_${pt_arr_ML[j]}_${pt_arr_ML[j+1]}
python3 machine_learning_Final.py --ymin ${y_arr_ML[i]} --ymax ${y_arr_ML[i+1]} --ptmin ${pt_arr_ML[j]} --ptmax ${pt_arr_ML[j+1]}
root -b -l -q draw_soverb_significance.C'('${y_arr_ML[i]}', '${y_arr_ML[i+1]}','${pt_arr_ML[j]}','${pt_arr_ML[j+1]}')'
root -b -l -q Superimpose_BDT_efficiencies.C'('${y_arr_ML[i]}', '${y_arr_ML[i+1]}','${pt_arr_ML[j]}','${pt_arr_ML[j+1]}')'
done
done
```

Preselection

Event Scaling

y and pt bins for ML

Main Script

machine_learning_Final.py

Change this signal and background ranges for different hadrons

```
54    selsig_cuts = sig.get_subset(f'(pt_D0 > {args.ptmin} and pt_D0 < {args.ptmax}) and (y_D0 > {args.ymin} and y_D0 < {args.ymax}) and (mass_D0 > 1.7 and mass_D0 < 2.1)') # ,size=promptDr
55    selbkg_cuts = bkg.get_subset(f'(pt_D0 > {args.ptmin} and pt_D0 < {args.ptmax}) and (y_D0 > {args.ymin} and y_D0 < {args.ymax}) and (1.6< mass_D0 < 1.7 or 2.1 < mass_D0 < 2.5)') # , si
```

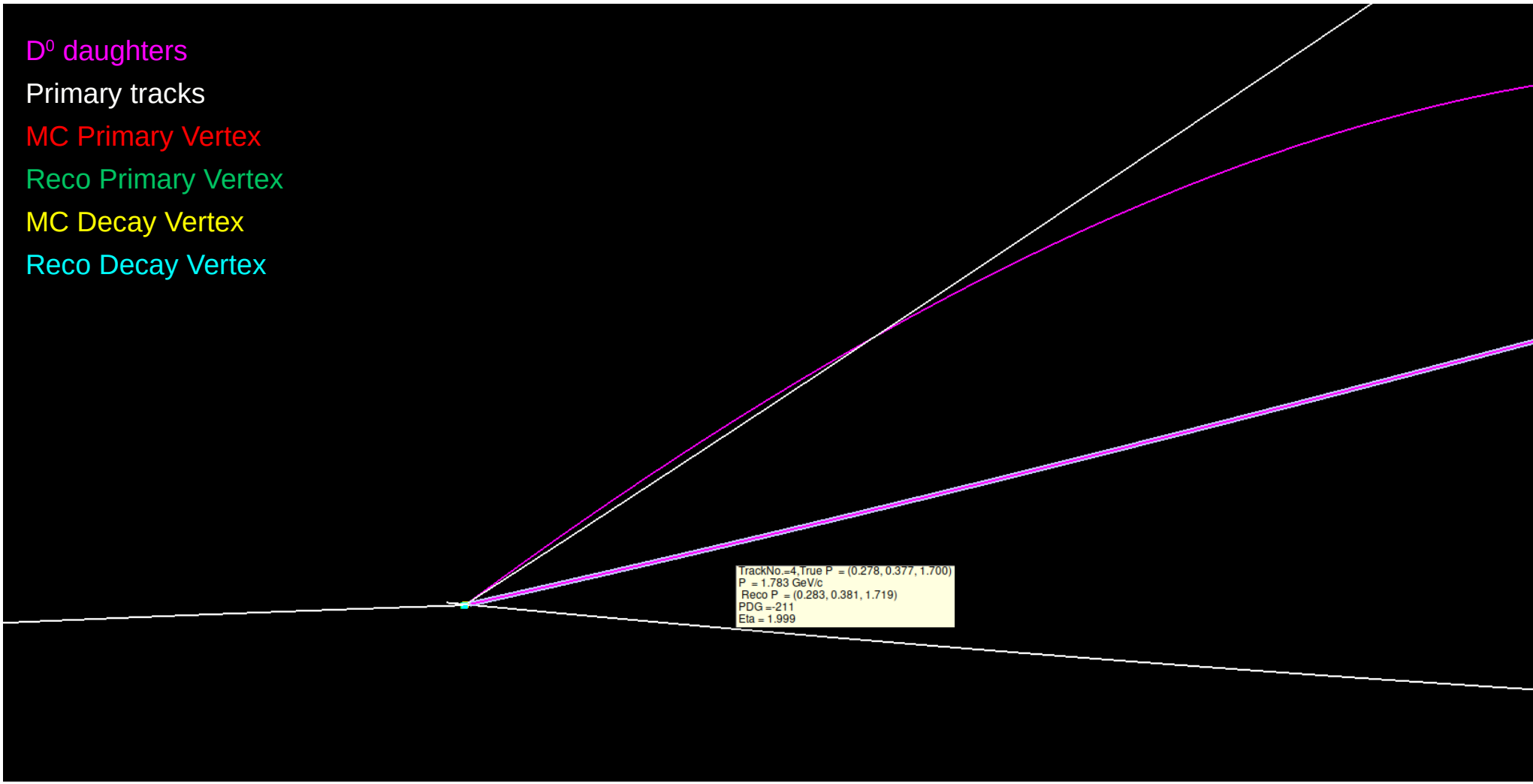
Play with these parameters if model is not trained properly

```
hyper_pars_ranges = {'n_estimators': (100, 500), 'max_depth': (1, 3), 'learning_rate': (0.01, 0.1)}
study = model_hdl.optimize_params_optuna(train_test_data, hyper_pars_ranges, cross_val_scoring='roc_auc', timeout=120, n_jobs=-1, n_trials=100, direction='maximize')
```

Check the intermediate plots carefully

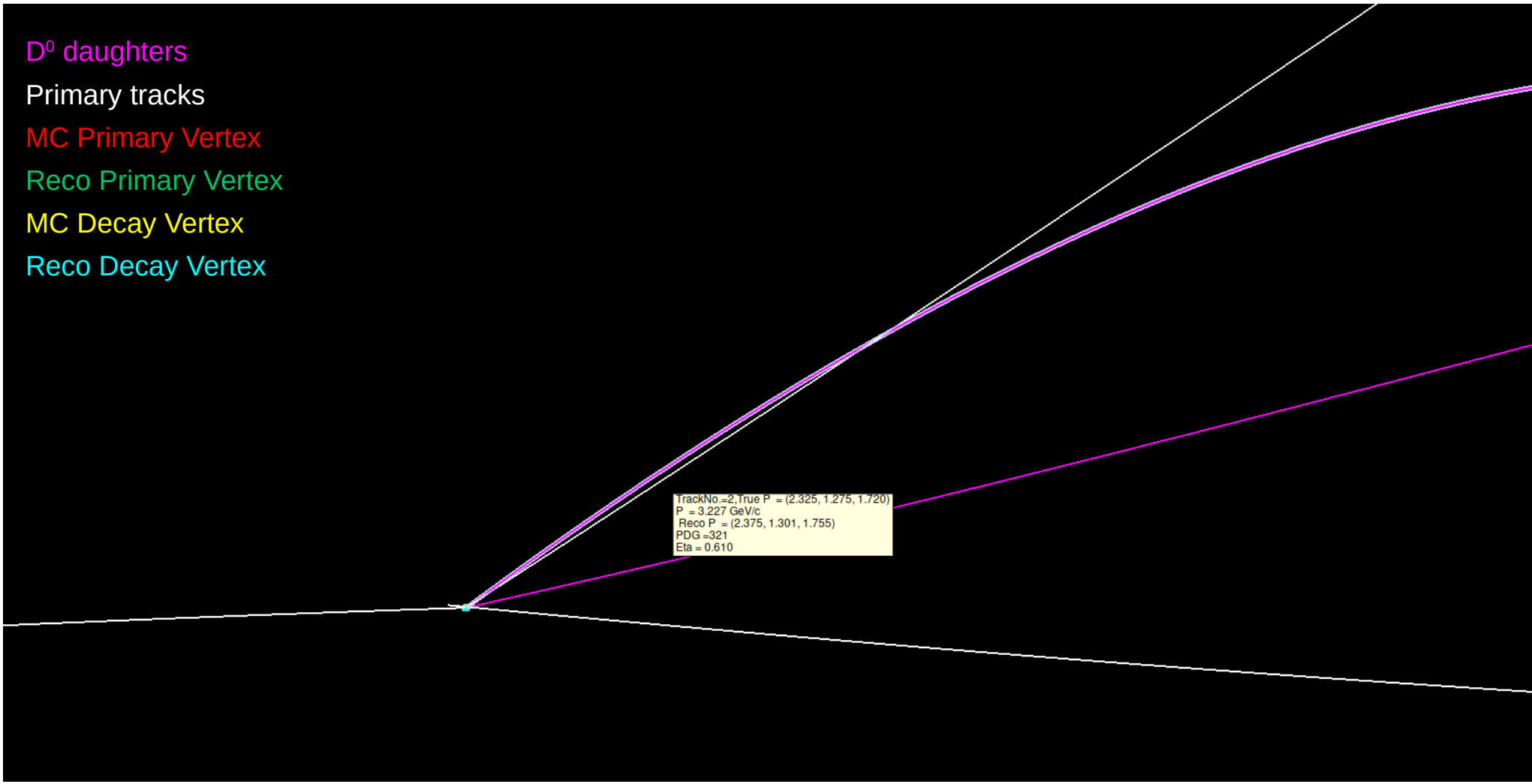
Event display Pictures

ep,10x100, $Q^2>1$ (D^0 sample)



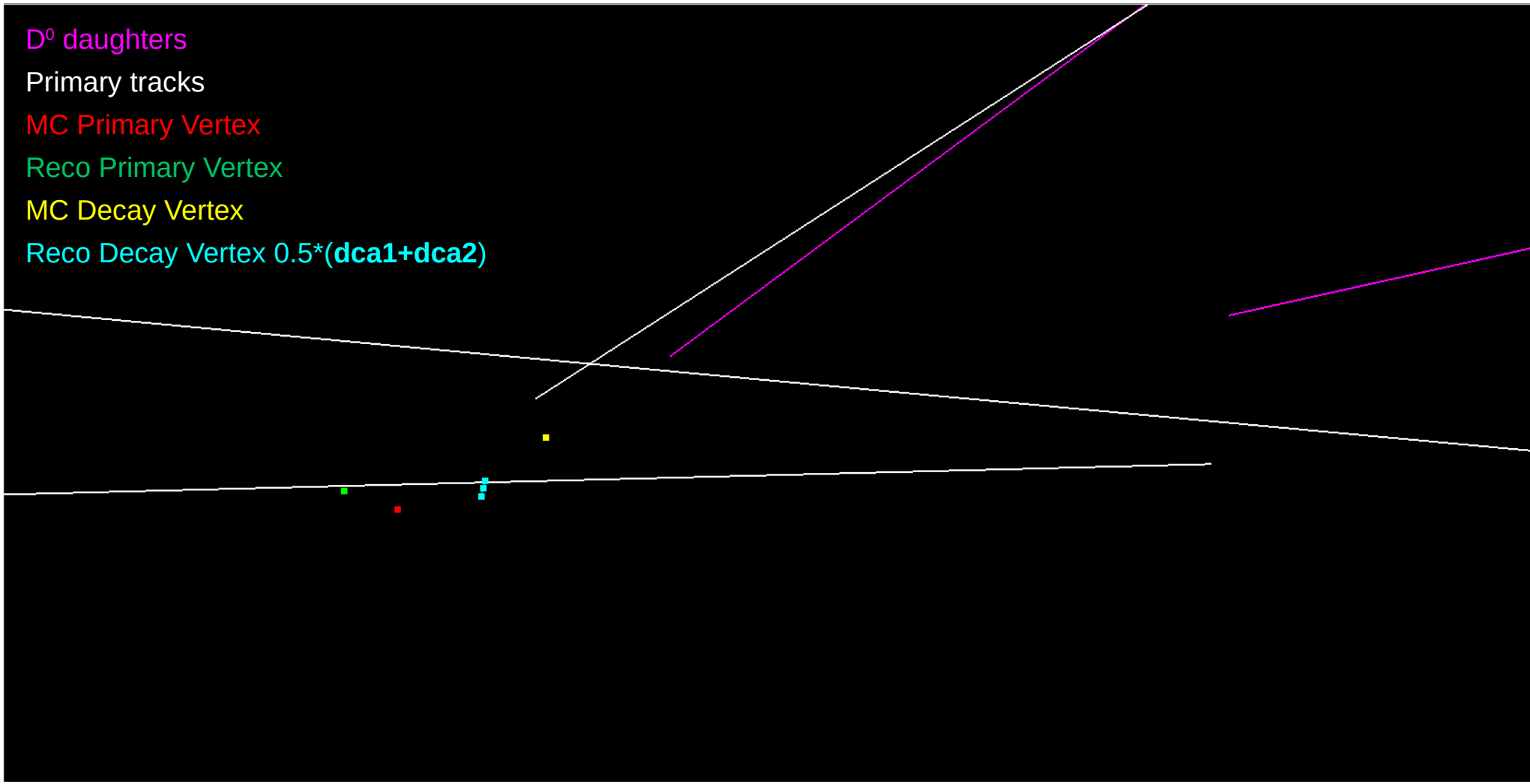
Event display Pictures

ep,10x100, $Q^2>1$ (D^0 sample)



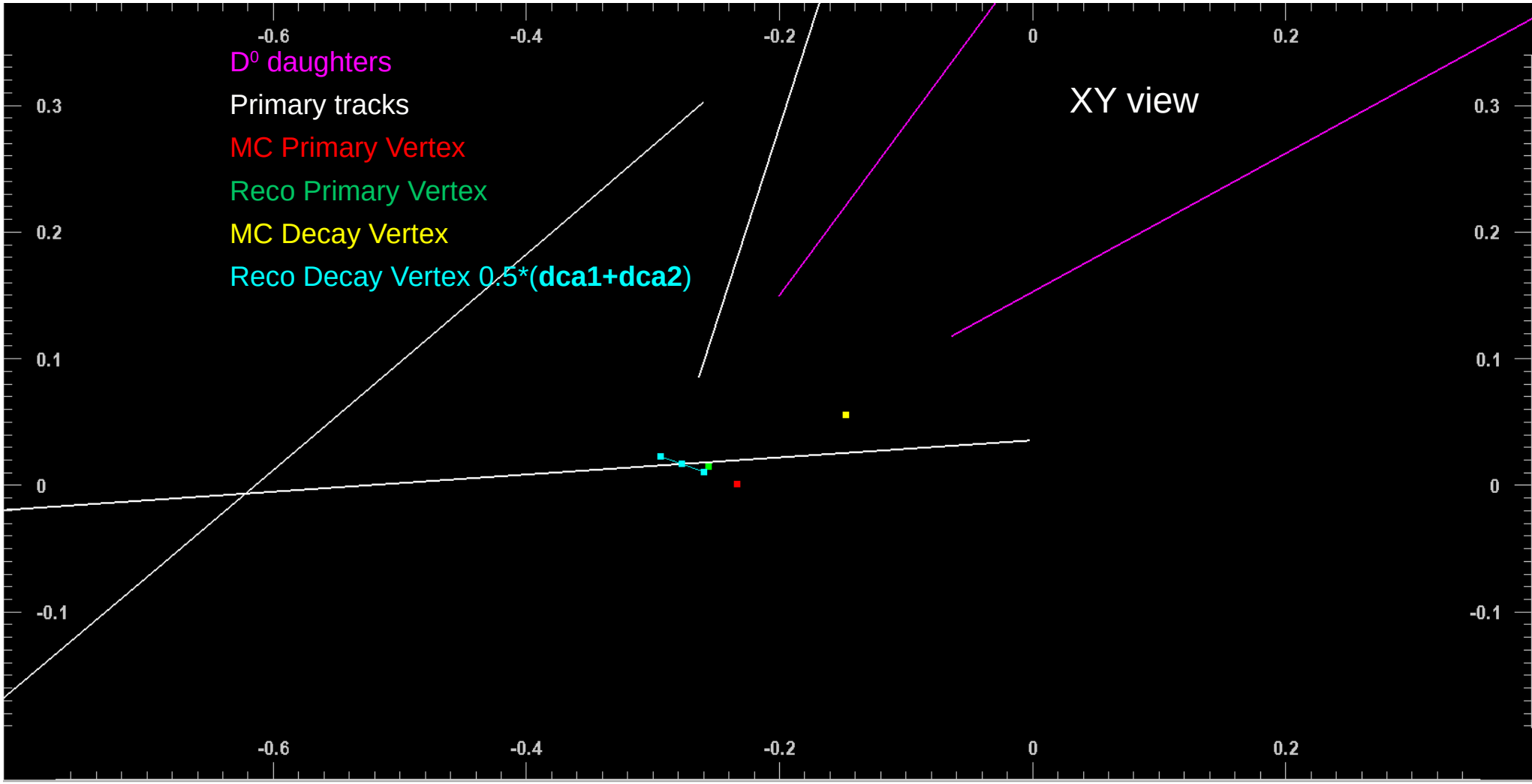
Event display Pictures

ep,10x100, $Q^2>1$ (D^0 sample)



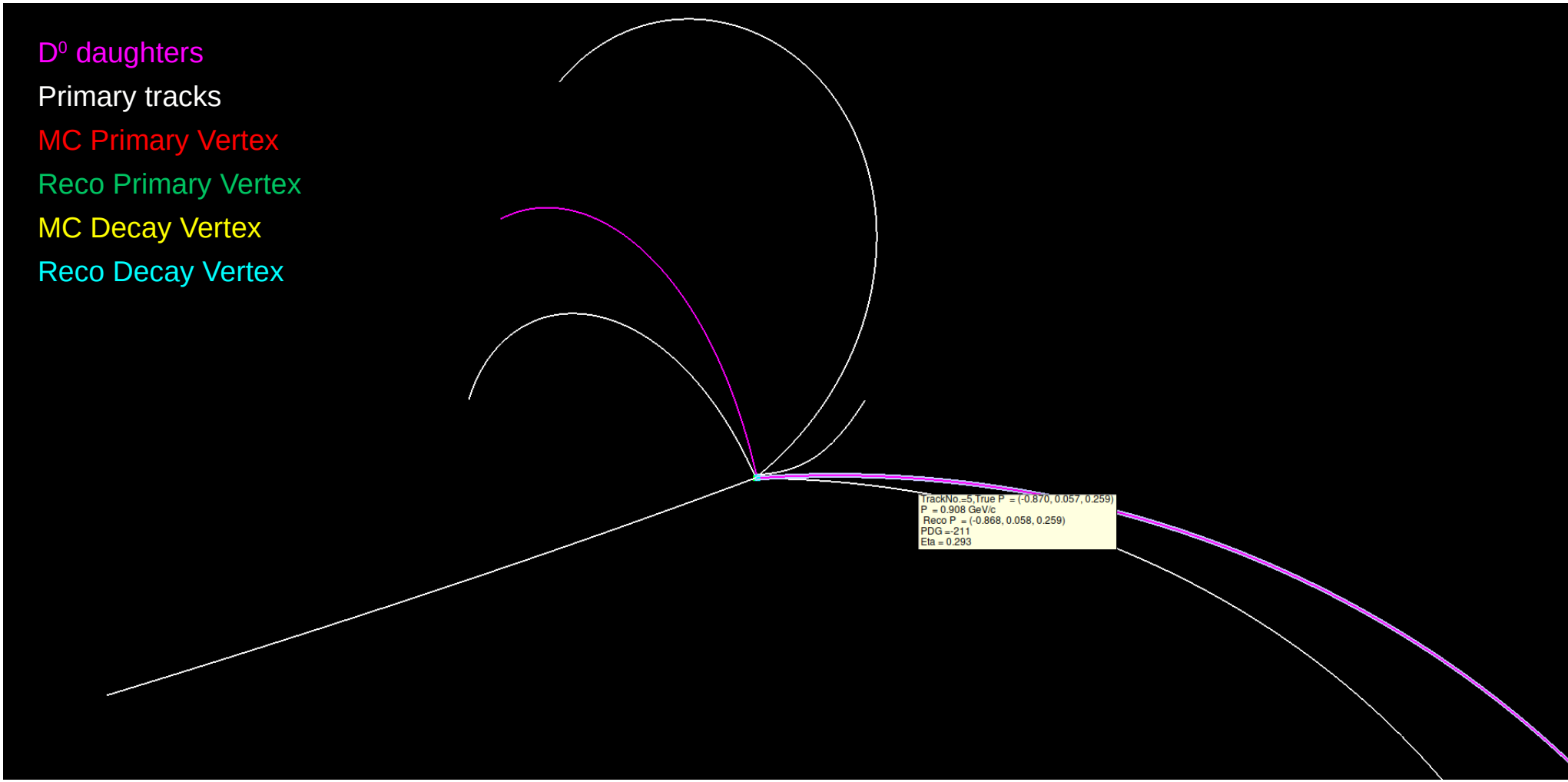
Event display Pictures

ep,10x100, $Q^2>1$ (D^0 sample)



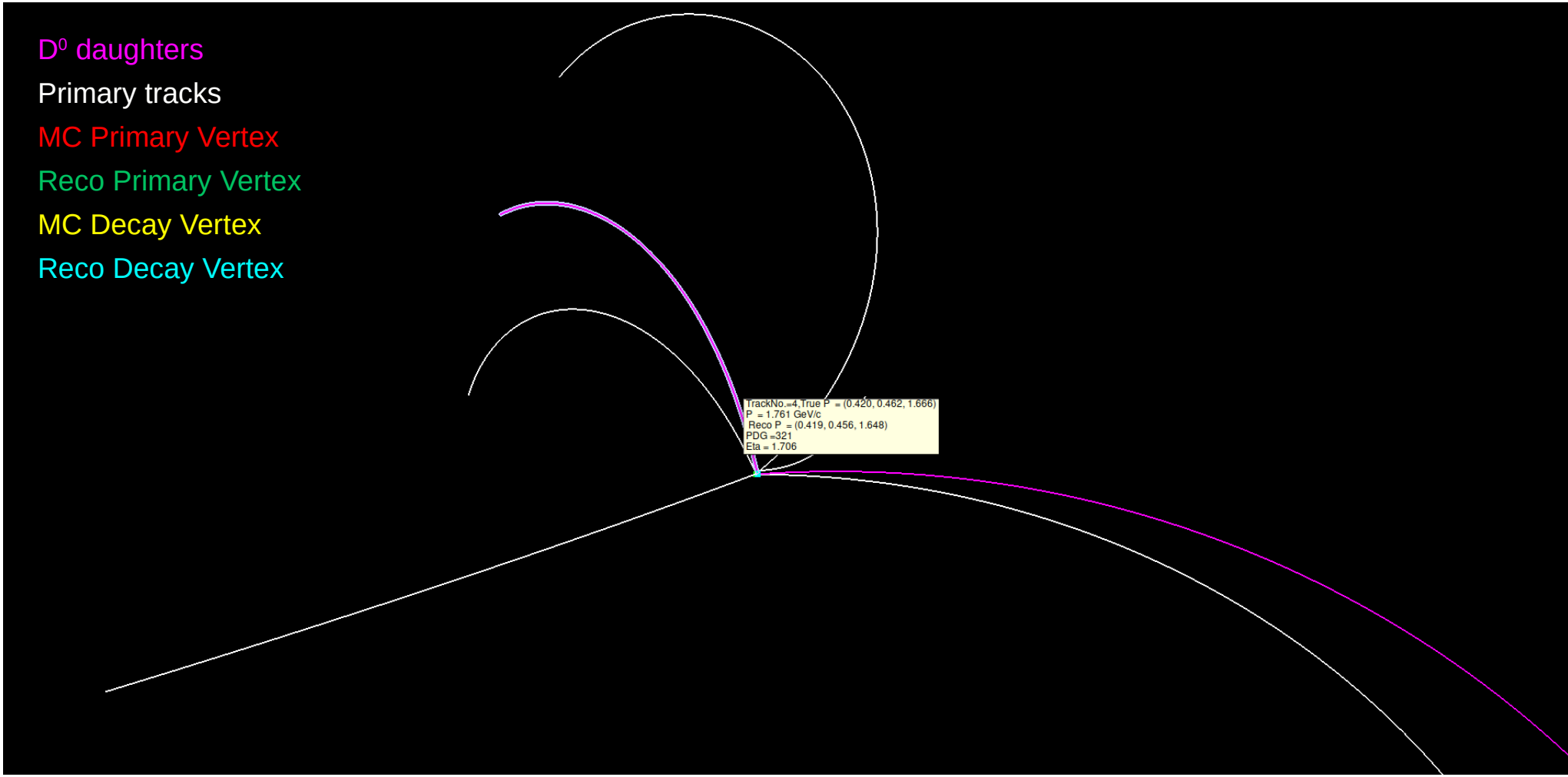
Event display Pictures

eAu,10x100, $Q^2>1$ (D^0 sample)



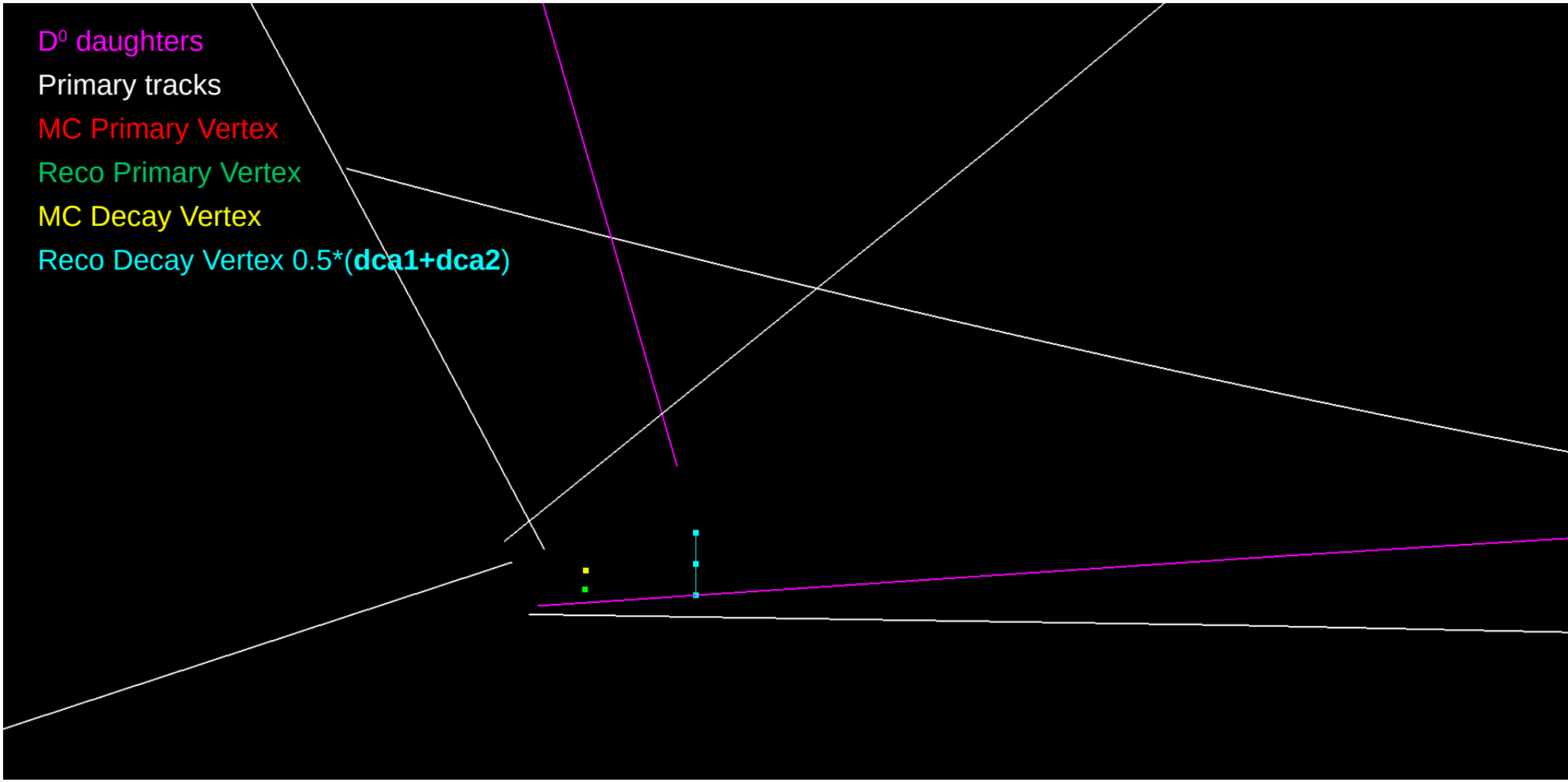
Event display Pictures

eAu,10x100, $Q^2>1$ (D^0 sample)



Event display Pictures

eAu,10x100, $Q^2>1$ (D^0 sample)



Event display Pictures

eAu,10x100, $Q^2>1$ (D^0 sample)