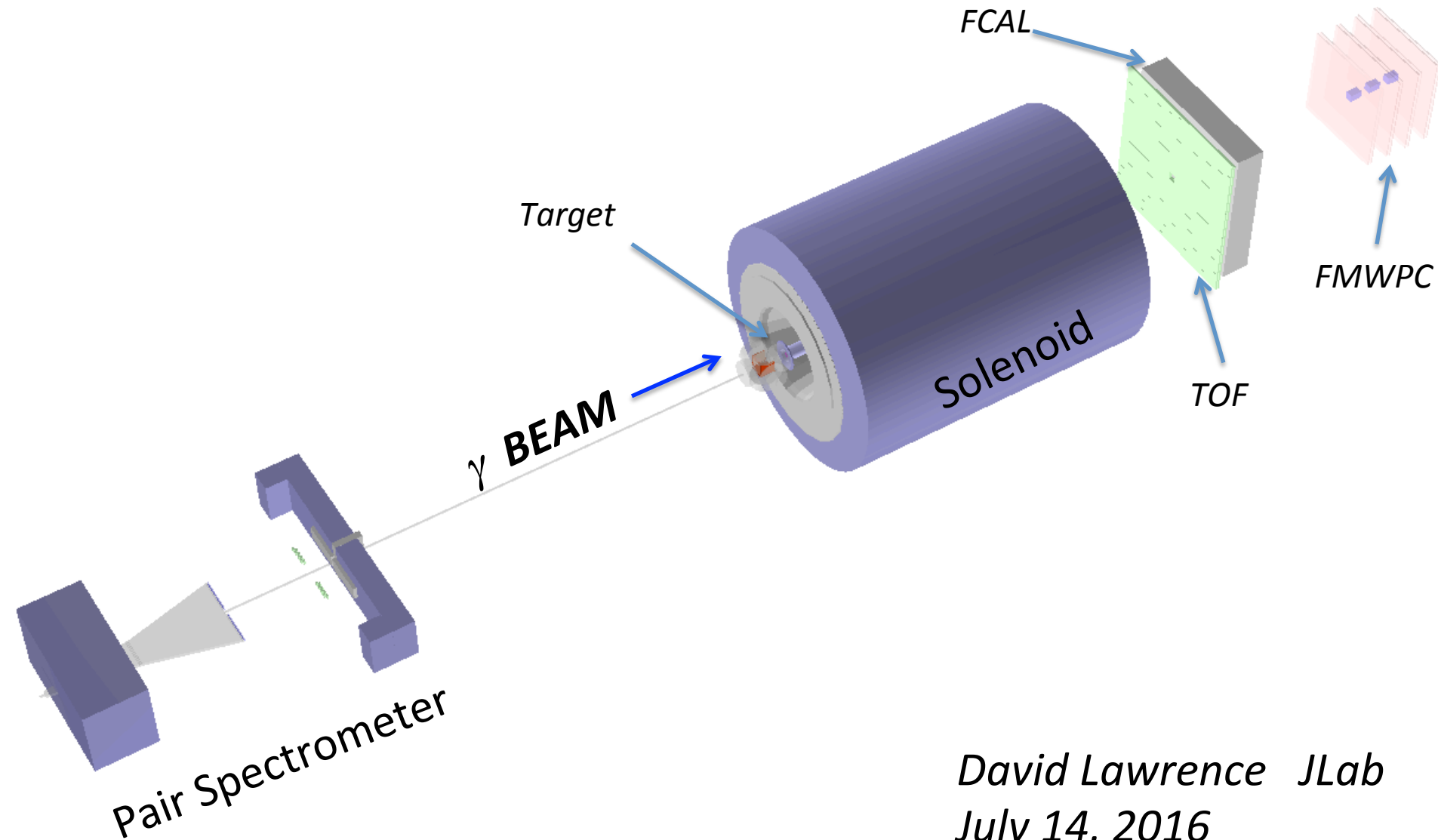



# CPPsim = GEANT 4 simulation



David Lawrence JLab  
July 14, 2016

## Software [\[edit\]](#)

---

- CPP software is mostly integrated into the standard [GlueX Offline Software](#). Get a working version of that first.
- Simulation : [CPPsim](#)
- Event Generation:
  - [CPP:gen\\_2mu](#)
- Analysis:
  - [CPP: Multi-variate Analysis](#)
- Misc:
  - A special directory has been set up in the subversion repository for files specific to the  $\pi$  polarizability measurement here:  
<https://halldsvn.jlab.org/repos/trunk/Experiments/PionPolarizability> 



# CPPsim

## Introduction [\[edit\]](#)

CPPsim is a GEANT4 simulation used for the **Charged Pion Polarizability** experiment in Hall-D. It is maintained as a separate project from the standard Hall-D simulation packages *hdgeant* and *HDGeant4*. It does, however, depend on the same [HDDS <https://github.com/JeffersonLab/hdds>] package for specifying the geometry in Hall-D as well as the *sim-recon* software suite for things such as the magnetic field map, Lorentz deflections tables, and HDDM I/O.

## Quick Start [\[edit\]](#)

1. Get a working version of *sim-recon* and make sure your environment is setup to use it
2. Make sure your ROOT version has *gdml* support enabled.
  - This is technically not needed if you have a *cproot.gdml* available (check if `$HDDS_HOME/$BMS_OSNAME/src/cproot.gdml` exists since if it is, then you're good to go)
  - If *cproot.gdml* is not in the `$HDDS_HOME/$BMS_OSNAME/src` then you'll have to find a copy or recompile ROOT to generate one from the HDDS xml source. Configure ROOT using the `"-enable-gdml"` flag before recompiling. Then you'll have to go back to step 1.
3. Get a working version of GEANT4 10.02p1 or greater and set your environment up to use it.
  - Here are the lines used on my MBP:
  - `> setenv G4 /usr/local/geant4/geant4.10.02p1.Darwin_macosx10.11-x86_64-llvm7.3.0`
  - `> source $G4/bin/geant4.csh $G4/bin`
  - `> source $G4/share/Geant4-10.*geant4make/geant4make.csh $G4/share/Geant4-10.*geant4make/`
4. Checkout CPPsim from subversion and build it via:
 

```
> svn co https://hallsvn.jlab.org/repos/trunk/Experiments/PionPolarizability/src/ CPPsim
> mkdir CPPsim-build
> cd CPPsim-build
> ../CPPsim/run_cmake
> make -j8
```

*n.b.* GEANT4 requires a recent version of *cmake*. (e.g. for Mac OS X I downloaded and installed 3.4.1 and used the command line executable in `/Applications/CMake 3.4.1.app/Contents/bin/cmake`)
5. Copy the *control.in* file from the CPPsim source directory to your working directory and modify to suit your job. This is a well commented file that should guide you through the options.
6. Run *CPPsim* that was left in the CPPsim-build directory with the `"-h"` option to see what options are available. No options will just run it using options from the *control.in* file.
  - *n.b.* CPPsim will automatically try running any file named `"vis,mac"` in the current working directory so if you don't want this, make sure that file is not there or use the `-nv` option when running CPPsim

### navigation

- [Main page](#)
- [Recent changes](#)
- [Random page](#)
- [Help](#)

### search

Search

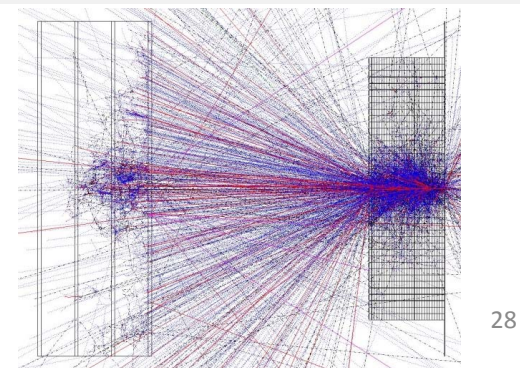
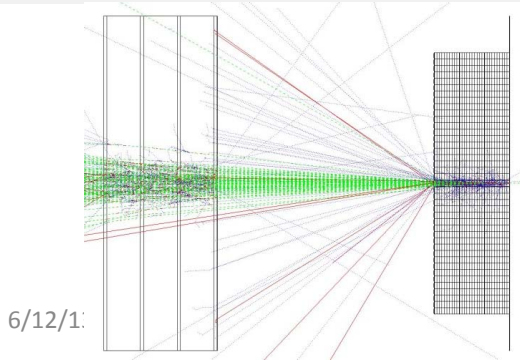
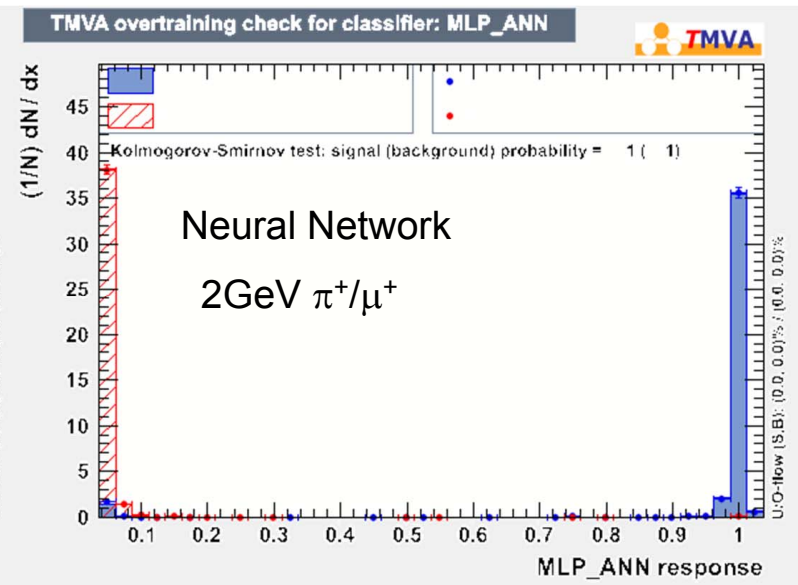
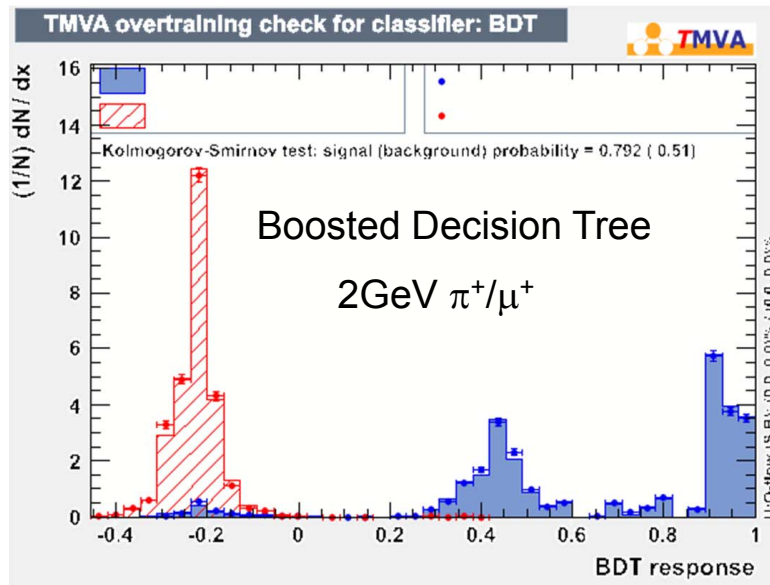
### tools

- [What links here](#)
- [Related changes](#)
- [Upload file](#)
- [Special pages](#)
- [Printable version](#)
- [Permanent link](#)
- [Page information](#)
- [Cite this page](#)

# From PAC 40 Presentation

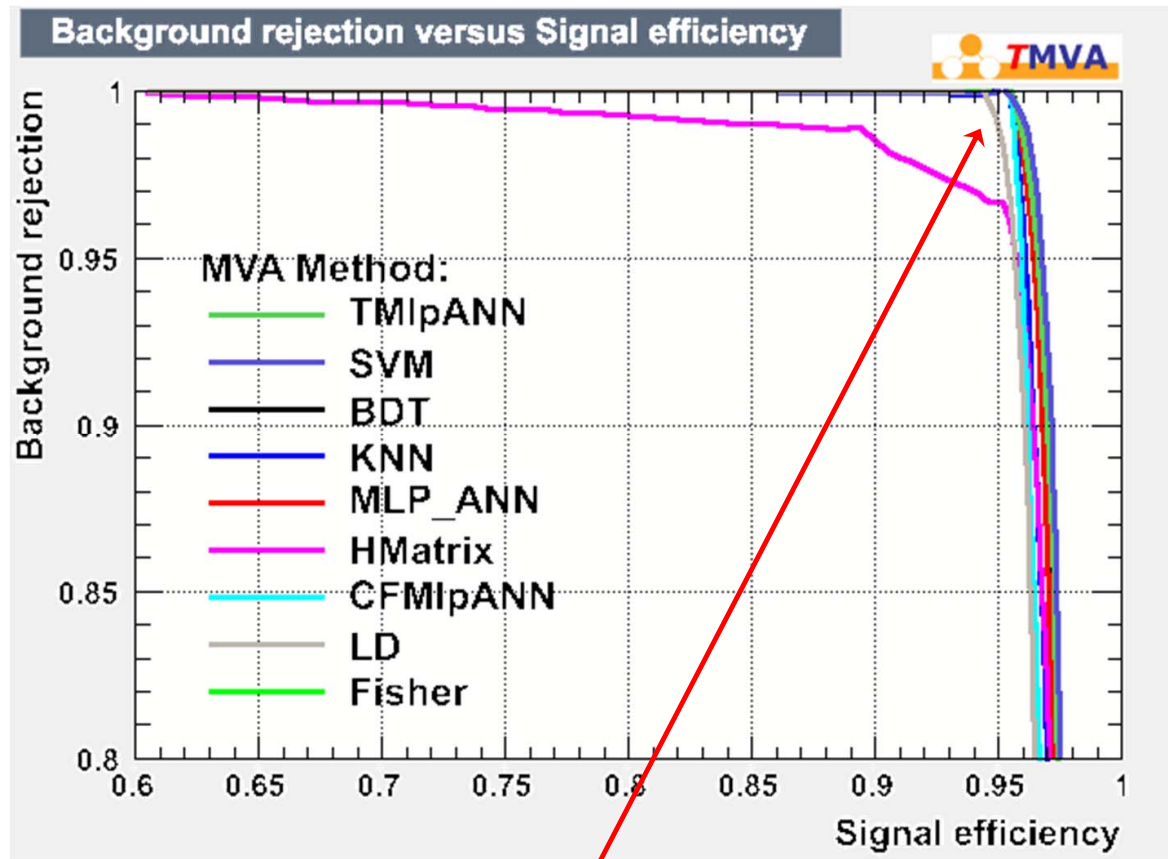
## MVA Classification Examples

Blue are  $\pi^+$  events, red are  $\mu^+$  events



From PAC 40 Presentation

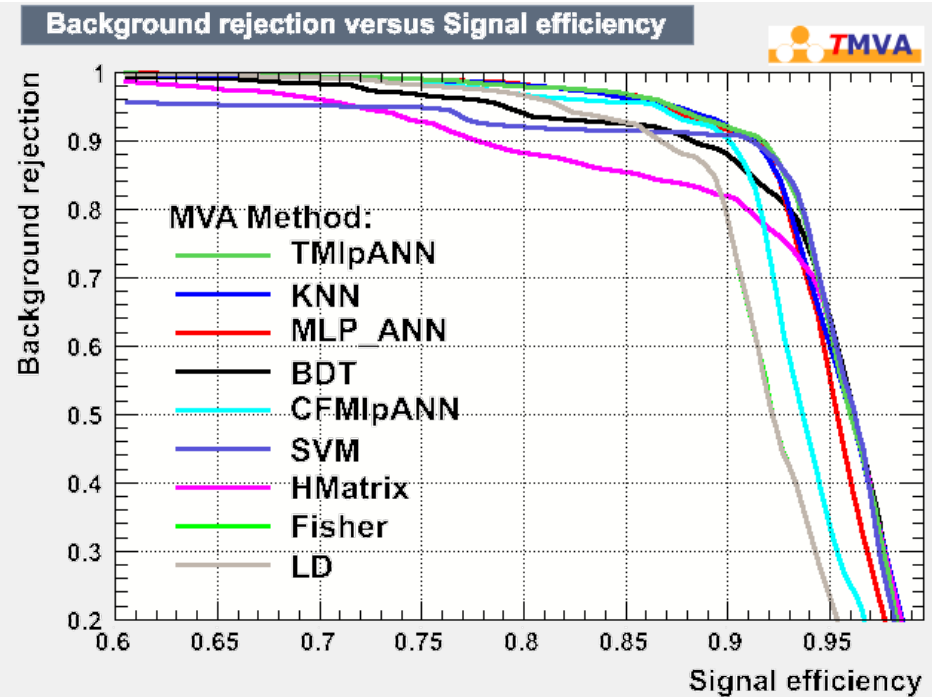
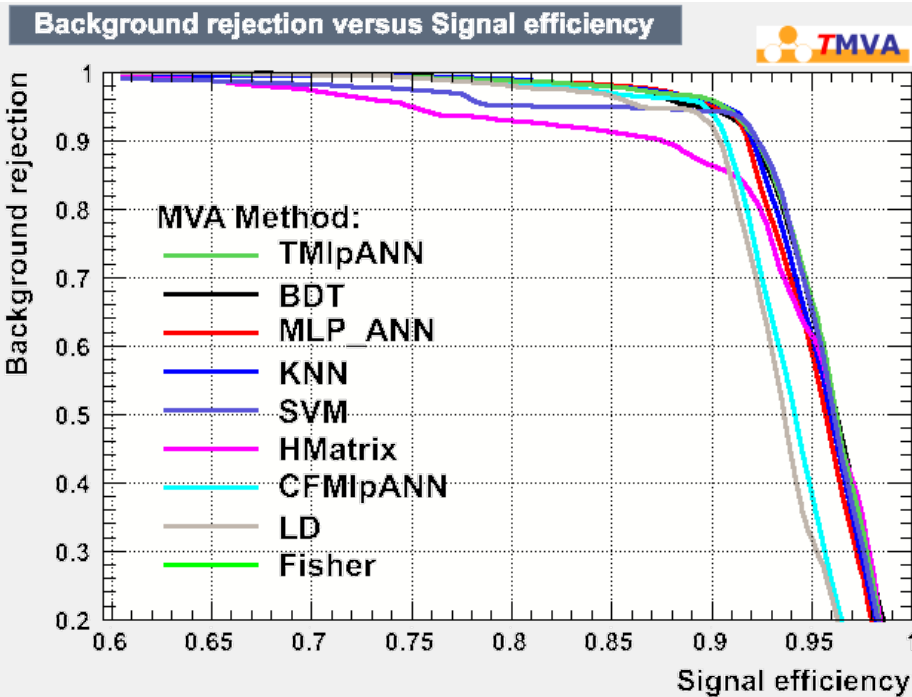
# Multi-Variate Analysis for 2 GeV $\pi^+$ and $\mu^+$



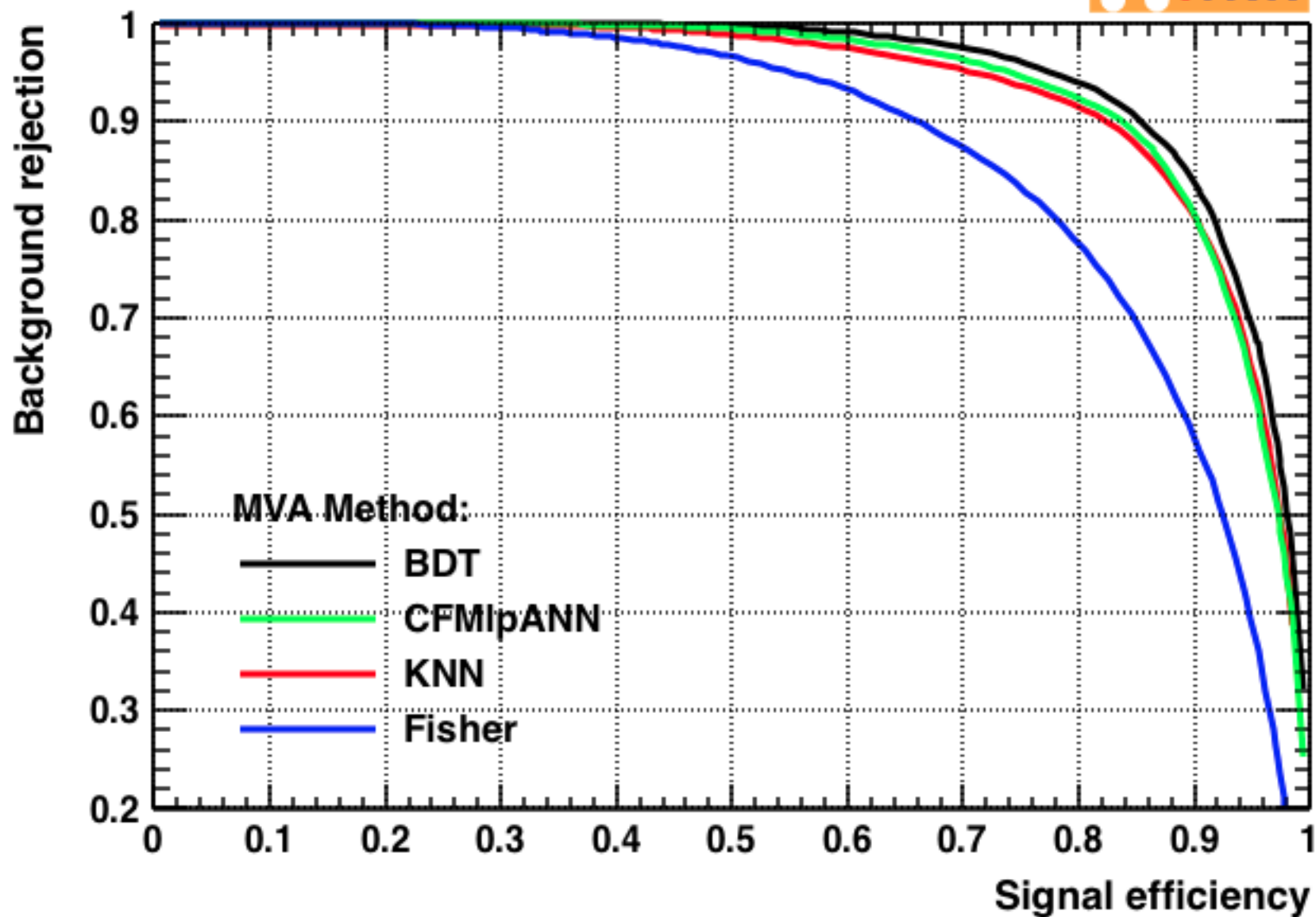
$\mu$  rejection at 0.998,  $\pi$  efficiency at 95%

# Realistic momentum Distribution

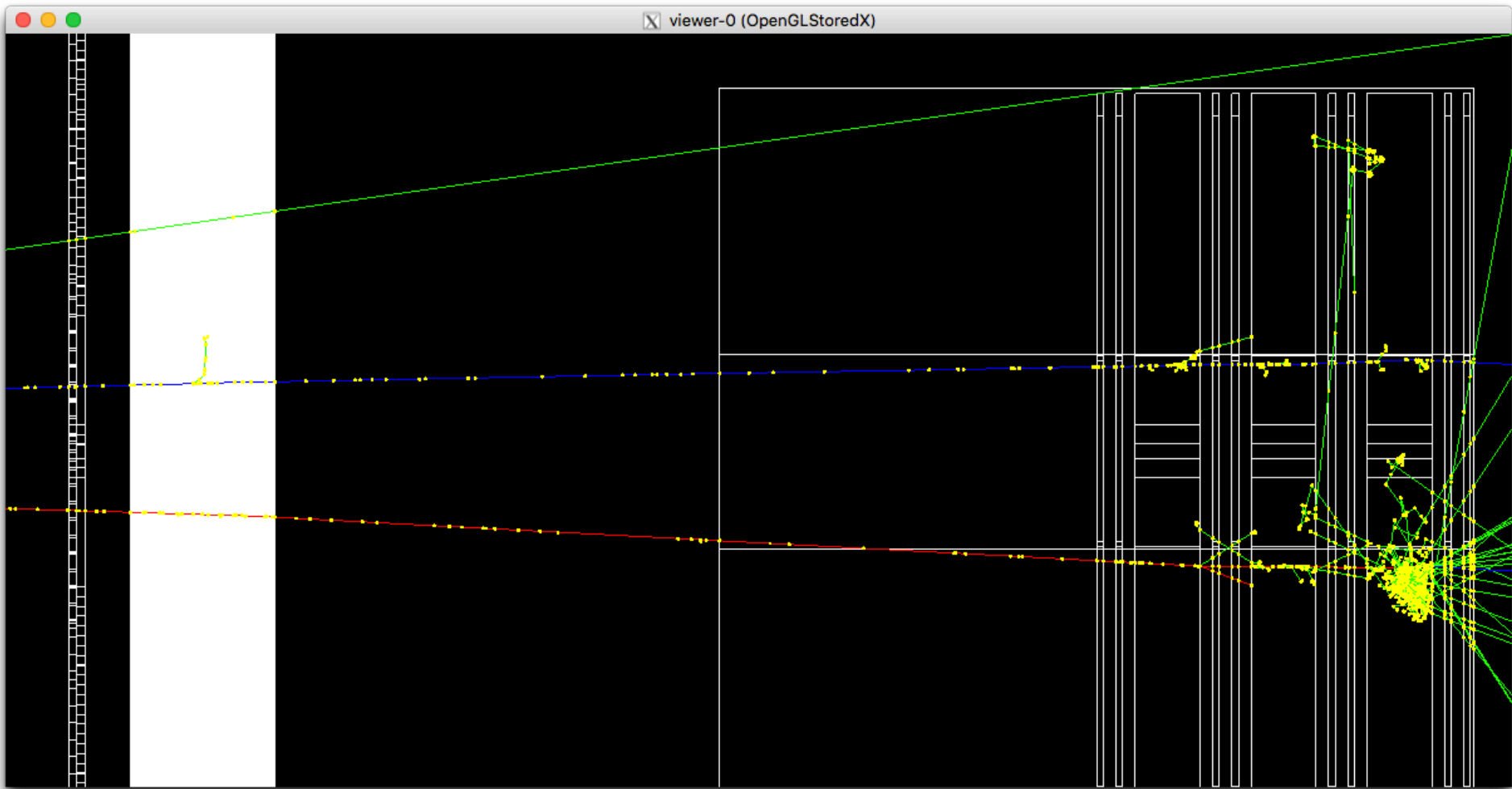
- *pb\_pol70\_10days: Primakoff + coherent  $\rho$  with 70% polarization*
- *10k events only (5k used for training, 5k used for testing)*



# Background rejection versus Signal efficiency

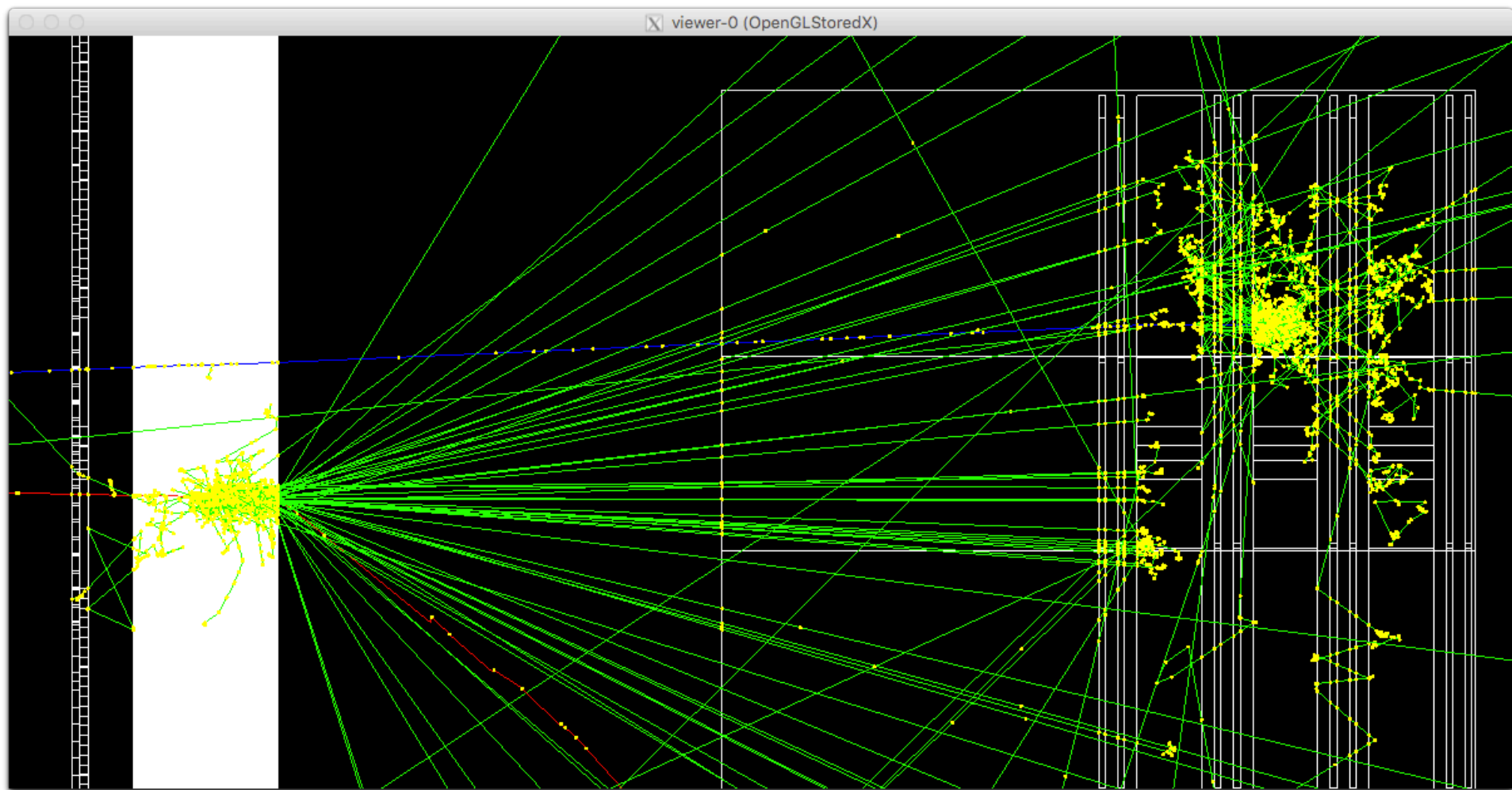


$$\pi^+\pi^-$$

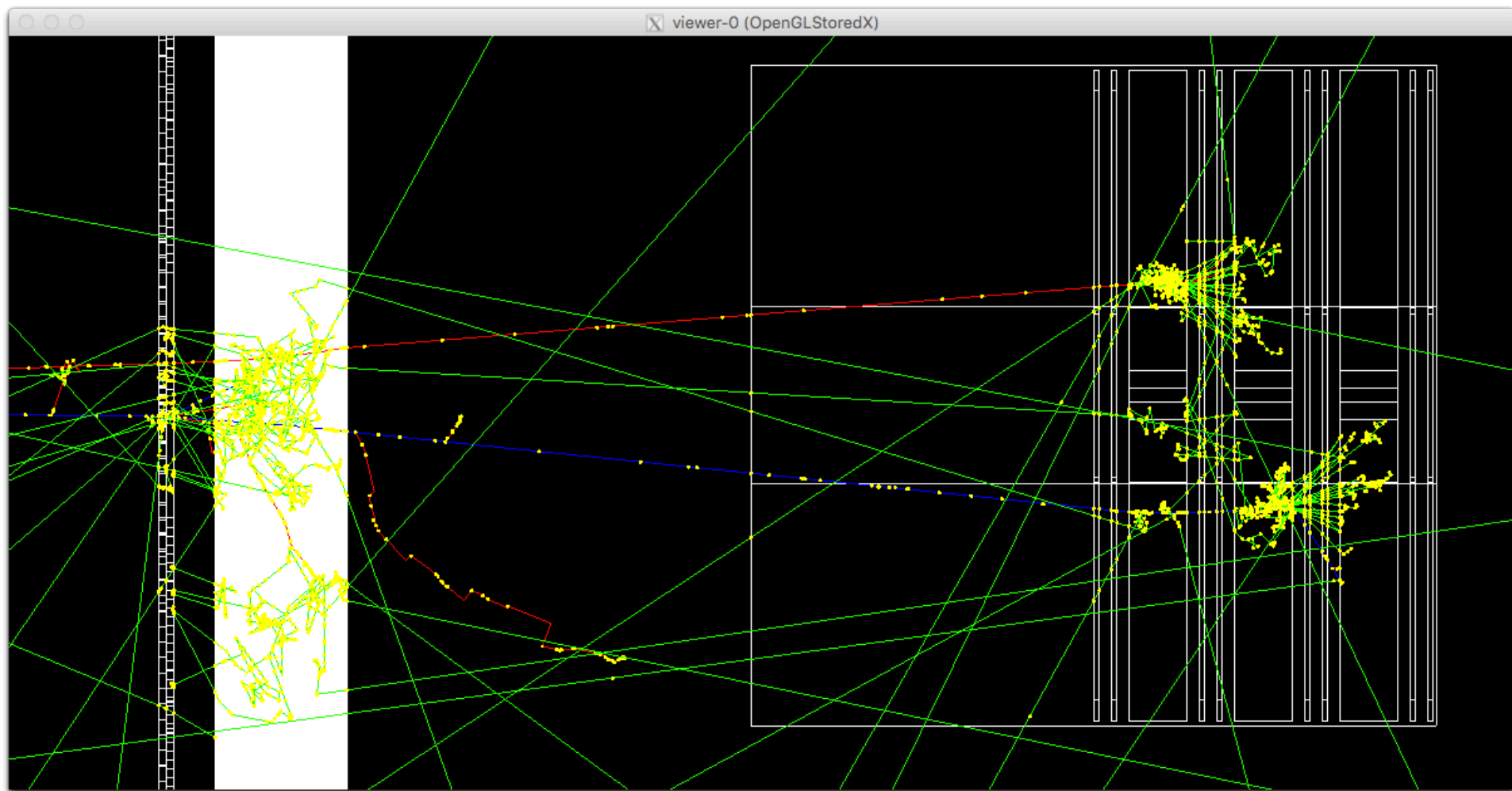




$$\pi^+ \pi^-$$

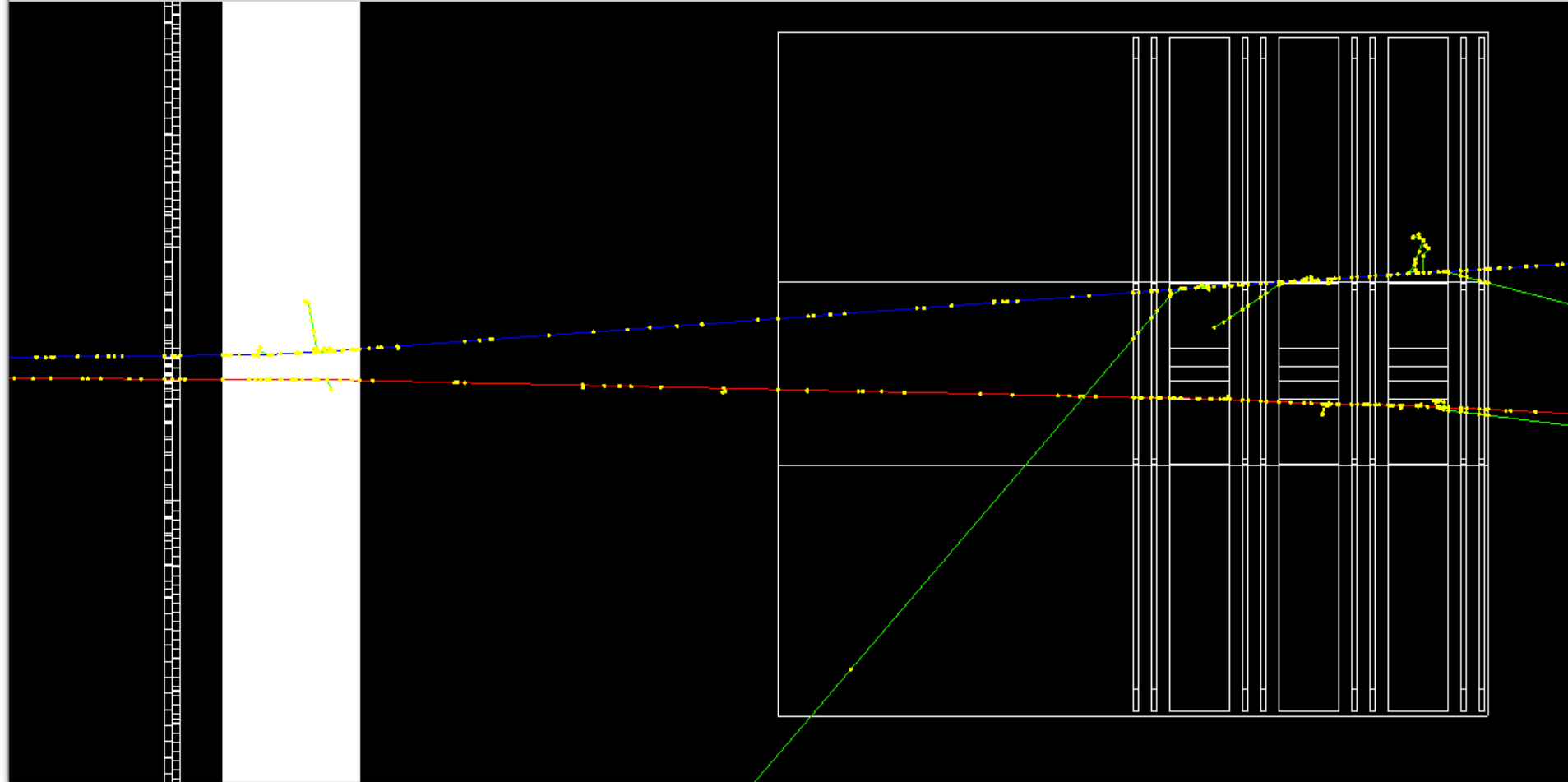


$$\pi^+ \pi^-$$

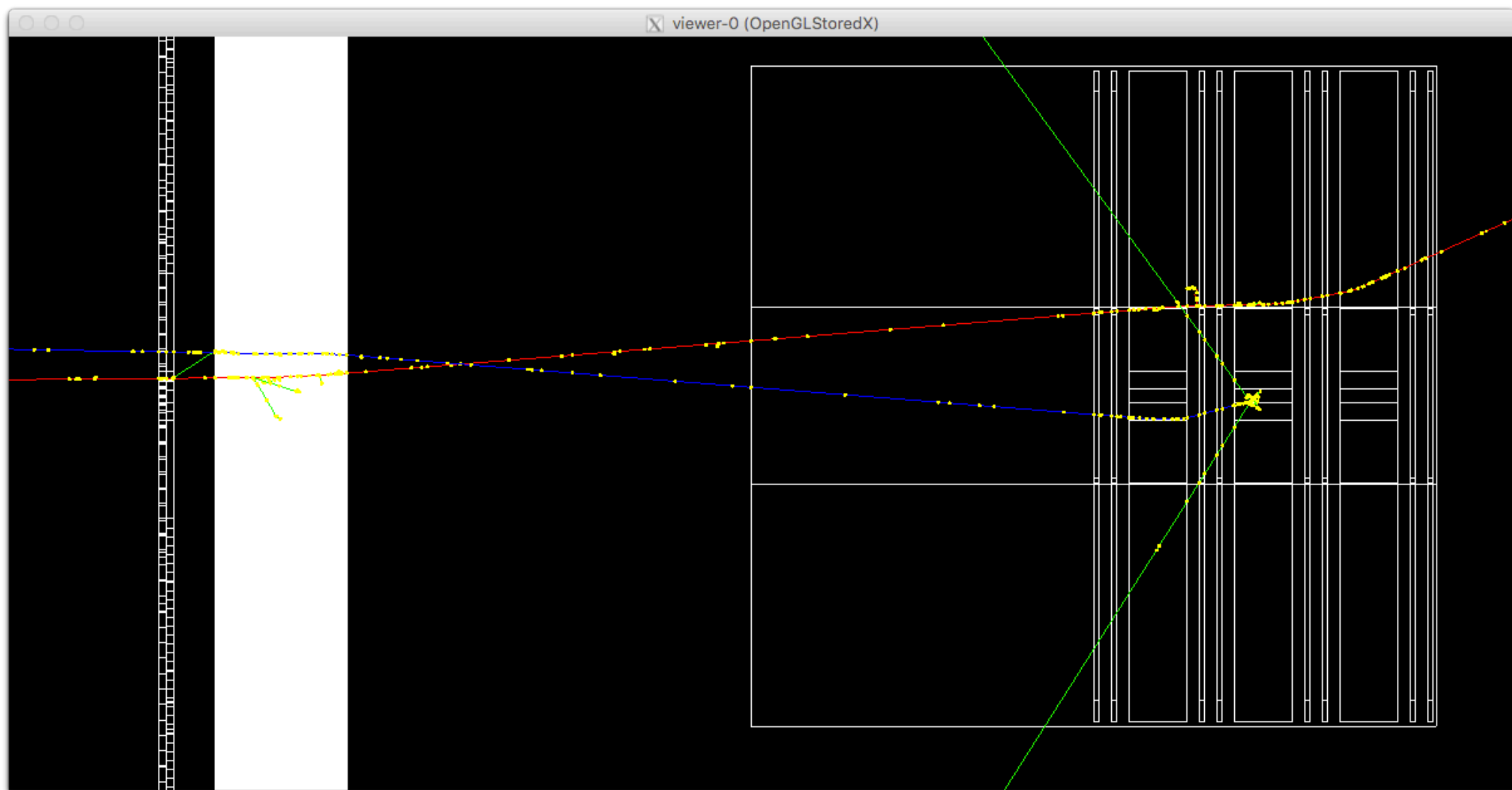


$$\mu^+ \mu^-$$

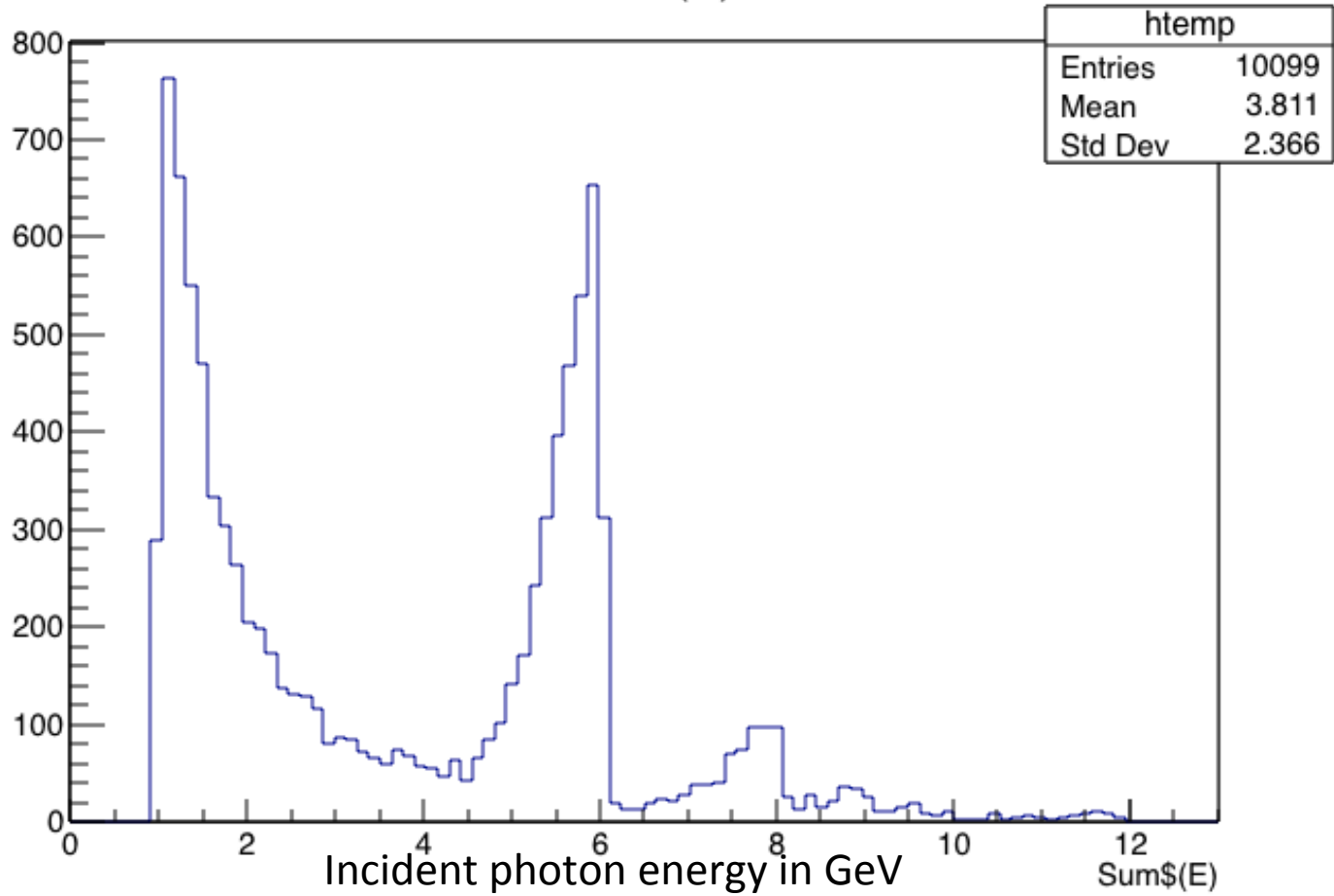
viewer-0 (OpenGLStoredX)



$$\mu^+ \mu^-$$

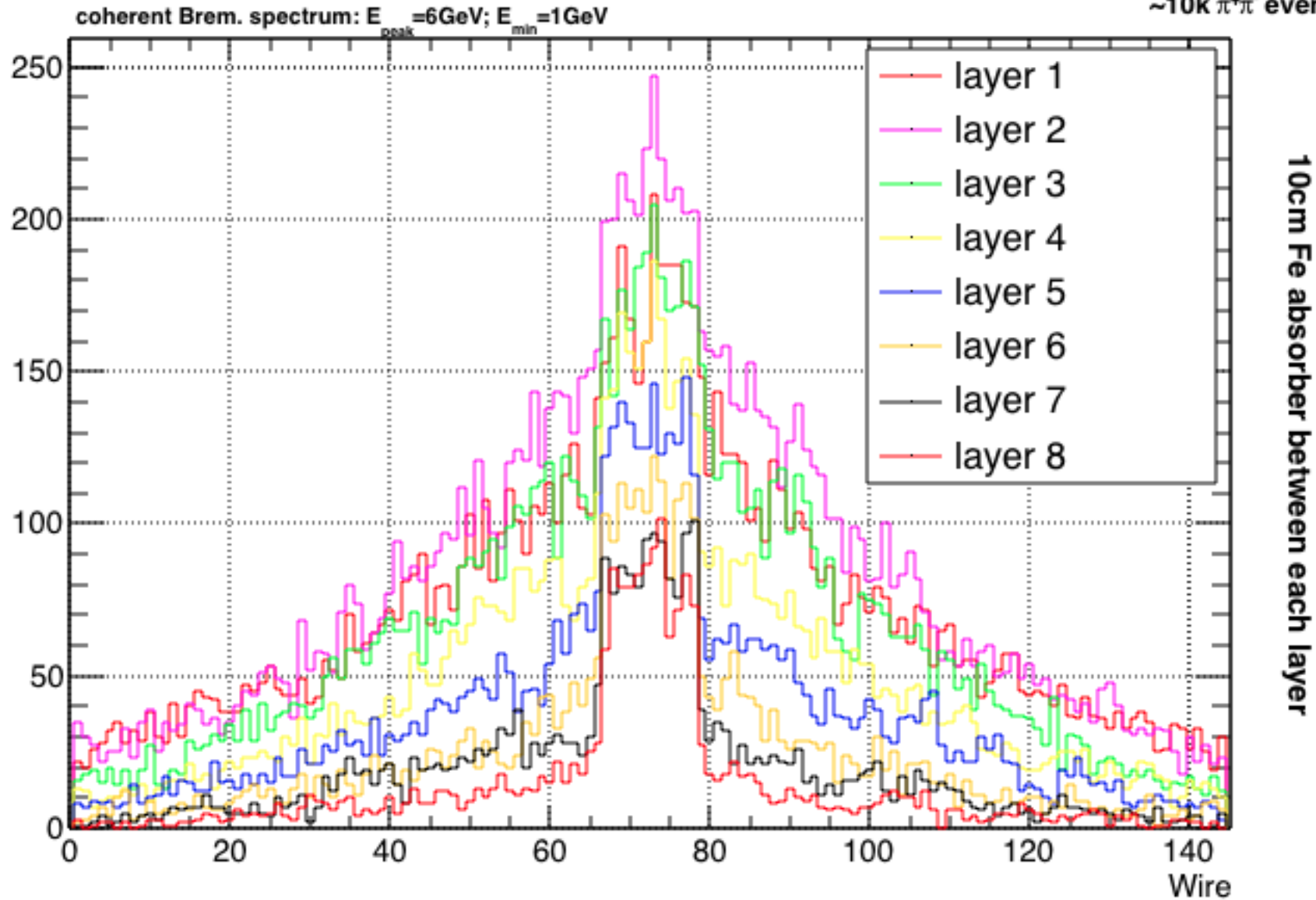


Sum\$(E)



# FMWPC Hits

July 14, 2016 DL  
git revision #abc1004  
~10k  $\pi^+\pi^-$  events



# Future

- Development of simulation tools continues
  - Able to modify geometry and test with MVA
  - Need better analysis and filtering of events prior to input into MVA (i.e. cut out low photon energy events)
  - Some dependence on calibration DB (variation=mc ??)