Updates on sectormap training

T. Lück

Weekly tracking meeting

March 2, 2018

outline

- Introduction
- Changes to MC track finding
- Changes to SectorMap training
- Performance new SectorMap
- Open issues
- Summary

Introduction

- current SectorMap contains connections in the wrong direction:
  - VXDTF2 assumes implicitly that all connections of SpacePoints go from outer to inner layer
  - investigation of SectorMap showed also connections from inner to outer layer
  - danger of creating loops e.g. L6 $\rightarrow$ L5 $\rightarrow$ L4 $\rightarrow$ L3 $\rightarrow$ L6 ....
  - likely cause: loopers in the trainings sample
- a fix was implemented: connections in the wrong direction are now rejected (not put into the SectorMap) when the SectorMap is loaded
- ultimate goal: proper training of SectorMap that this does not happen
- some ideas (will present today):
  - only use fitted MC tracks for training (so far unfitted): hope to reject zig-zagging tracks (scattered)
  - better looper rejection by modifying the MC track finder

Changes to the TrackFinderMCTruthRecoTracksModule

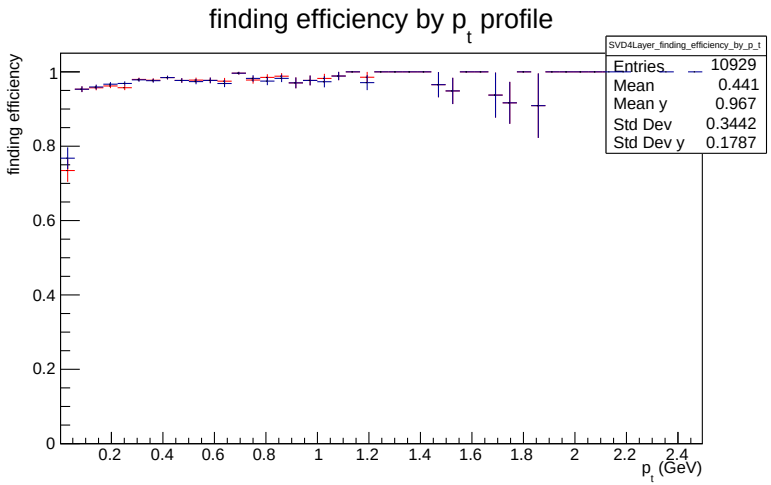
- so far already option to only use hits on the first n loops:
  - implemented for CDC hits only
  - all other hits will be marked as auxiliary
  - auxiliary hits will still be added to the track candidate but not used for efficiency estimation
  - time of flight is used to distinguish between the loops
- adopted the method implemented for CDC hits and extended it on SVD and PXD hits
  - will have an impact on the validation plots for hit efficiency (small for full validation, more relevant for SVD tracking)
- added option to discard auxiliary hits:
  - will not be used for creating the track candidate
  - by default turned off

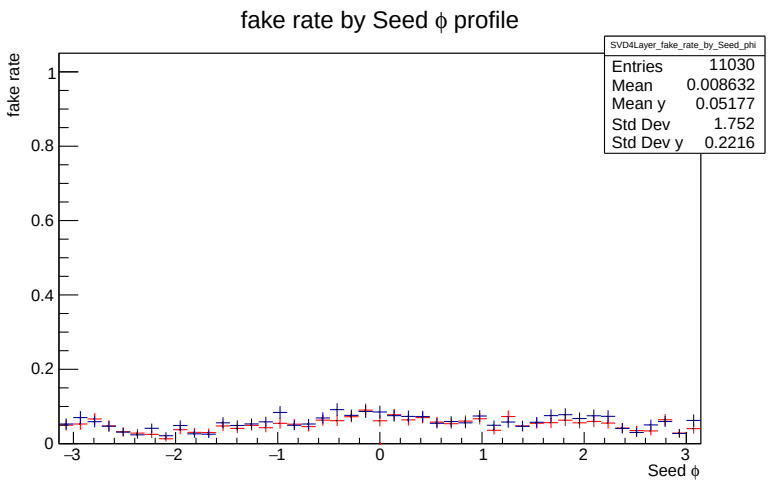
Changes to the SectorMap training

- only fitted tracks are now used for training:
  - RT2SPTCConverterModule is used to convert RecoTrack (from MC track finder) to SpacePointTrackCand (used in the training)
  - added option to reject RecoTracks without fit in the RT2SPTCConverterModule (RecoTrack::wasFitSuccessful())
- only SVD hits (PXD hits, if used) from the first 0.5 loop are used for the MC track candidate
  - for loopers only the outgoing arm of the track will be used
  - reduces the number of wrong direction connections but not completely (later more)

Performance SVD only tracking

- trained SectorMap on 1 million (Y(4S) + 1 muon) events
- standard tracking validation scripts (1k Y(4S) events with 1xbkg)
- red: new SectorMap; blue: default SectorMap (from DB)





Performance SVD only tracking

- number of wrong connections drastically reduced from 75 to 5 but not 0 (decays outside of layer 3?!?)

0xBkg

tag	ϵ	fake rate	clone rate	hit eff.
reference	0.9697	0.0189	0.0173	0.9625
nightly-2018-02-26	0.9798	0.0186	0.0177	0.8937
this Branch	0.9777	0.0174	0.0159	0.8942

1xBkg

tag	ϵ	fake rate	clone rate	hit eff.
default SecMap	0.9685	0.0582	0.0215	0.8865
this Branch	0.9670	0.0518	0.0191	0.8860

Open issues

SPTCReferee needs some rework

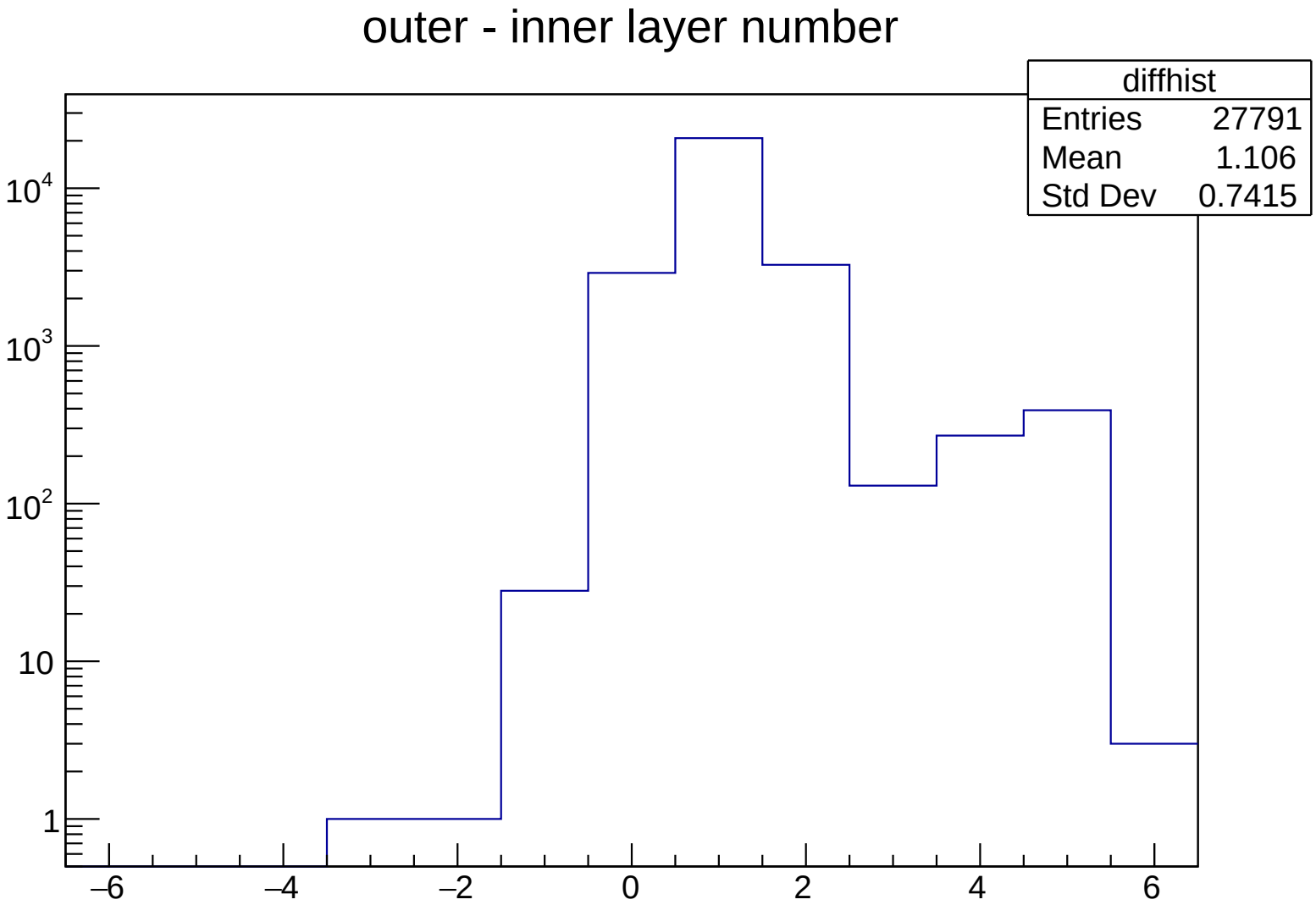
- used to perform some quality checks to the track candidates used for training
- sets "referee" status but referee status is ignored in the following
- some checks are ignored: i.e. cut on number of SP
- for other checks SP are removed from the track candidate instead of removing that track candidate completely (my personal opinion): i.e. two SP on same sensor; distance between two SP
- already implemented looper rejection needs to be revisited as it does not seem to perform as expected

Summary

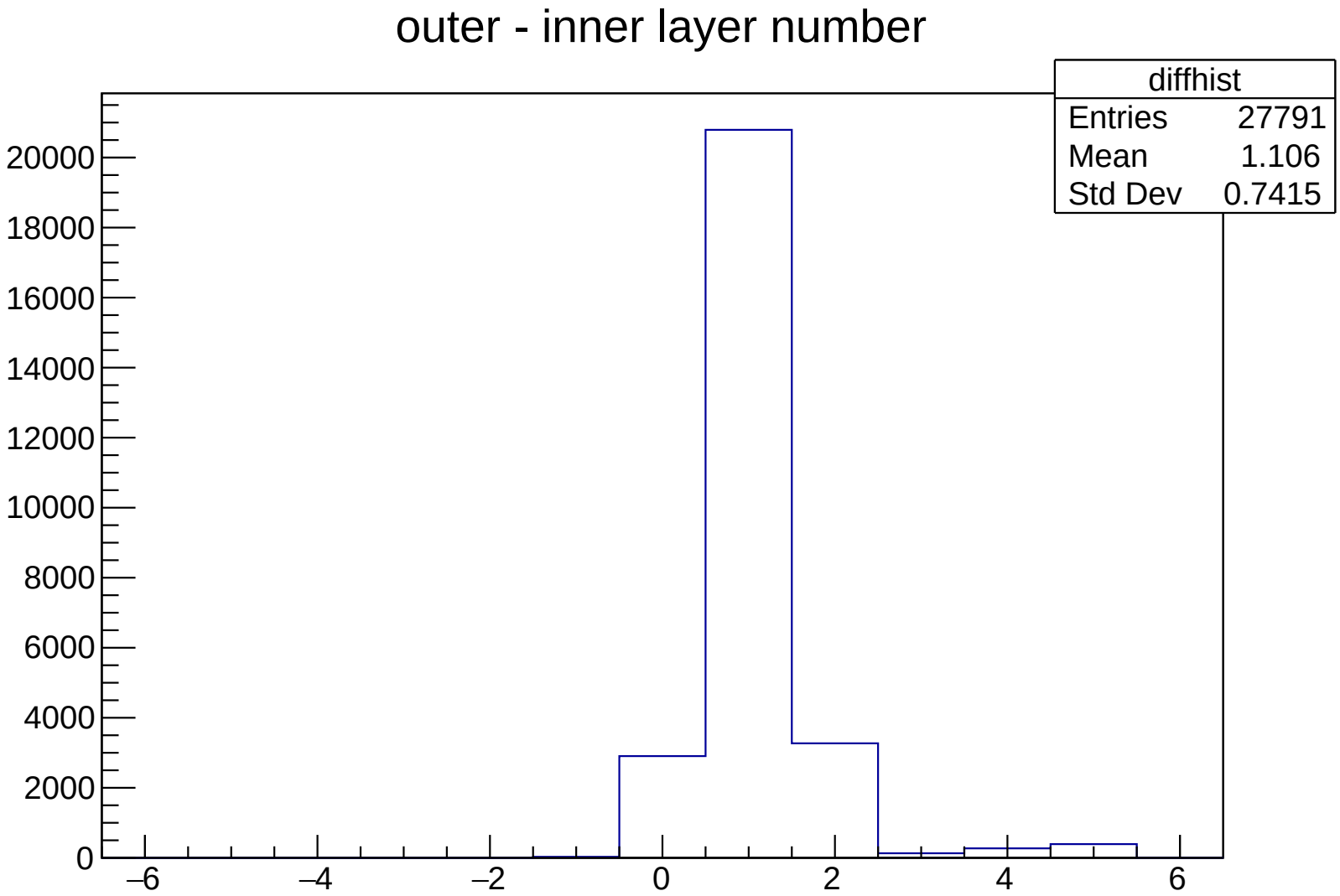
- implemented changes in the training of the SectorMap
- performance very similar to current SectorMap
- issue with wrong connections not completely resolved
- need to compare performance with fitable tracks only
- need to study other performance bench marks (CPU time, memory consumption)
- try with Phase 2 geometry (wrong connection more relevant here)

BACKUP

Phase3 SectorMap



Phase3 SectorMap



Phase2 SectorMap