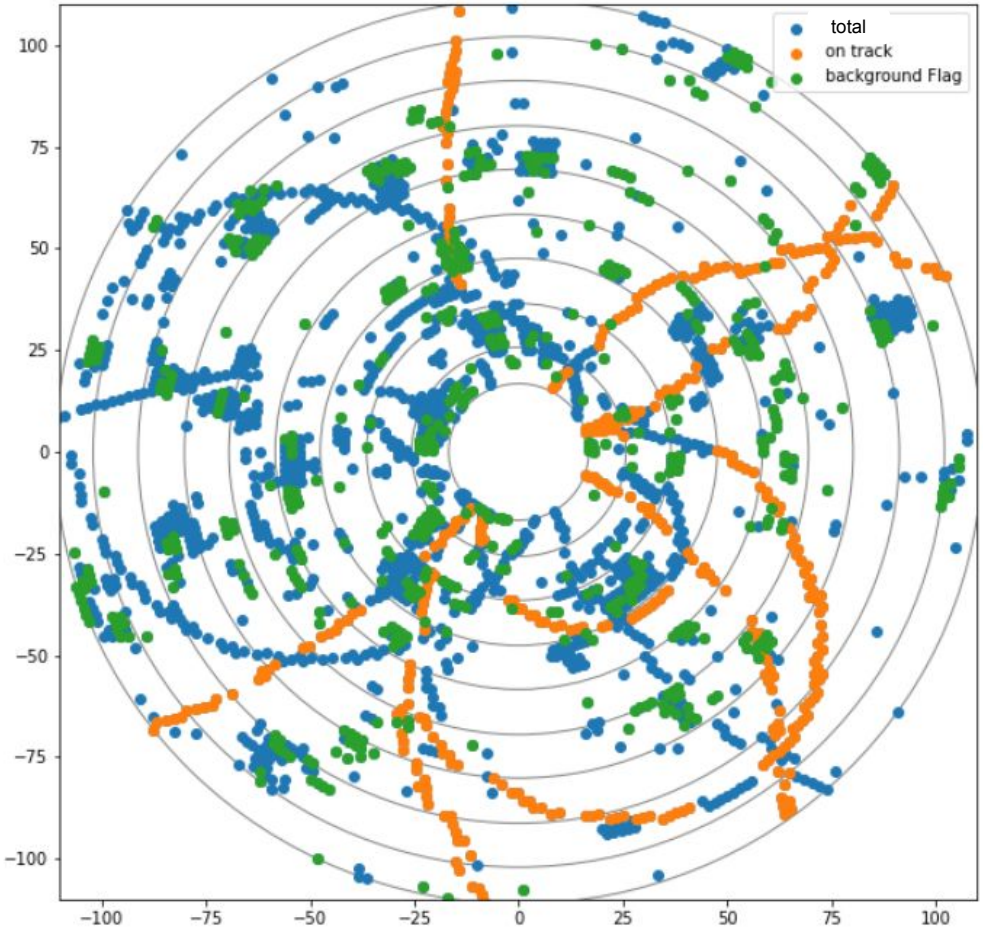
MVA HIT-BACKGROUND FILTERS IN CDC

Yulan Fan, Alexander Glazov, Christian Wessel

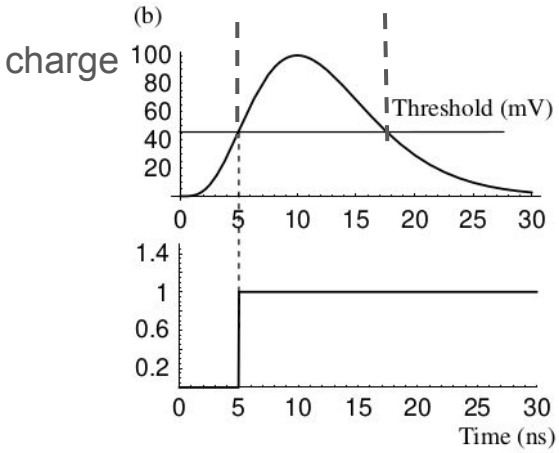
1st Oct, 2024 @ Belle II Germany Meeting

Motivation



An example of event display in CDC

Motivation



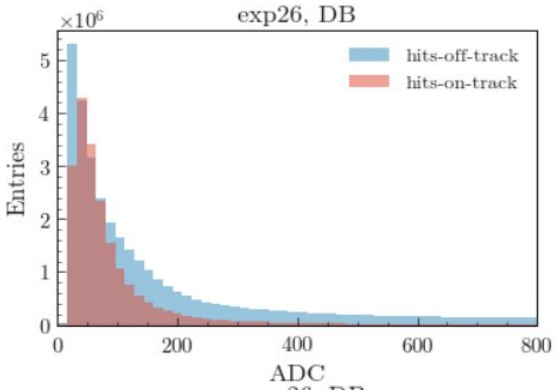
$$\text{ADC} = \int \frac{dc}{dt} dt$$

Time over threshold (TOT)

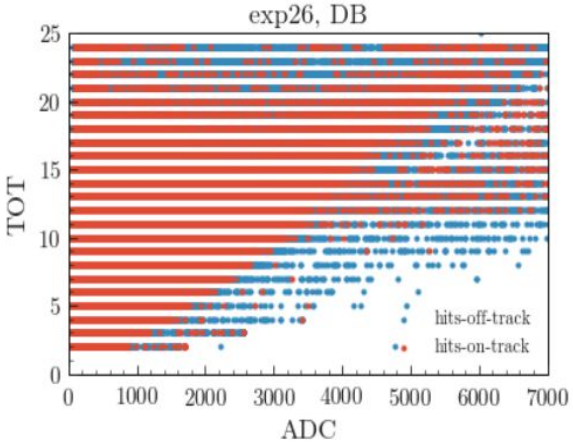
Database(DB) as reference:

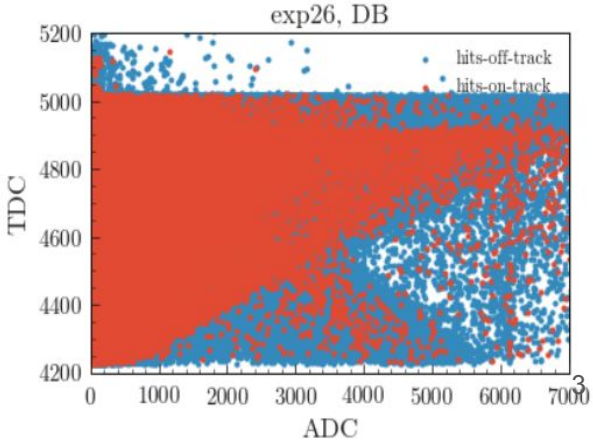
ADC>=15, TOT>=2, ADC/TOT>=3 (Inner super layers)

ADC>=18, TOT>=2, ADC/TOT>=3 (Outer super layers)



Sizable amount of hits-off-tracks & different correlation patterns for hits on tracks and hits off tracks for ADC, TOT and TDC, MVA can be used to distinguish them





Strategy

Introduce a MVA to suppress background hits, **especially in the low and upper limits of ADC**

> **Training a MVA**

>> Training on raw data, using hits on tracks as signal and hits off tracks as background

>> exclude the asic background

> **Track finding efficiency**

> **Time consumption of related-modules**

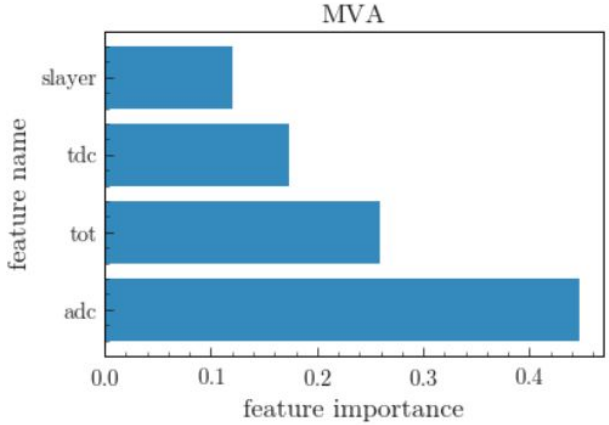
MVA: training preparation and performance

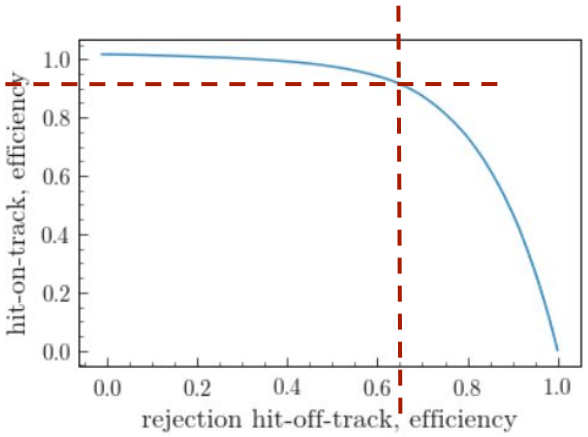
All:
ADC>=1, TOT>=1, ADC/TOT>=0
(payload generated in wireHitRequirement)

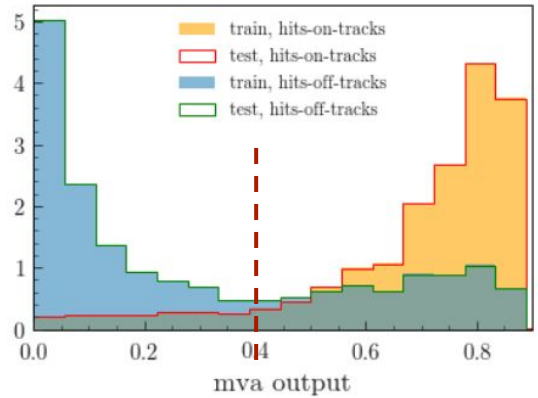
Training Variables:
ADC, TOT, TDC and inner/outer super layer

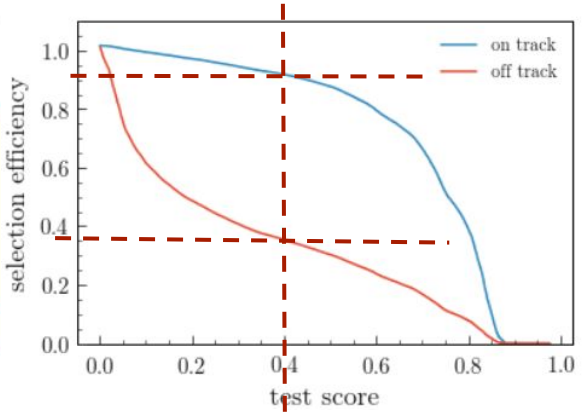
Hyper-parameter:
optimized with Optuna

Training sample:
1.2 million exp26 raw data









MVA: implement into basf2

C.W. initial version of implementation

CDCWireHitPreparer

Specify the parameter in WireHitPrepare module

```
else if (filterName == "mva") {
   return std::make_unique<MVACDCWireHitFilter>();}
namespace {
 using MVACDCWireHitFilter = MVAFilter<CDCWireHitVarSet>;}
```

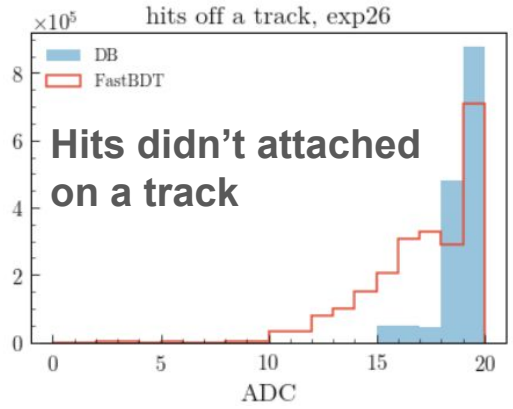
Loop each wire-hit to get training variables

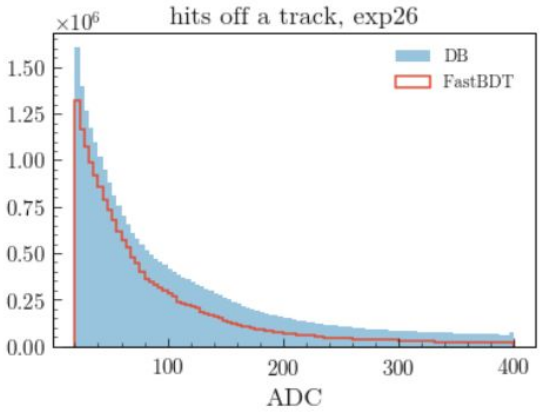
```
bool CDCWireHitVarSet::extract(const CDCWireHit* wireHit)
{
 const auto* cdcHit = wireHit->getHit();
 var<named("adc")>() = cdcHit->getADCCount();
 var<named("tot")>() = cdcHit->getTOT();
 var<named("tdc")>() = cdcHit->getTDCCount();
 var<named("slayer")>() = cdcHit->getISuperLayer() == 0 ? 0 : 1;
 return true;
}
```

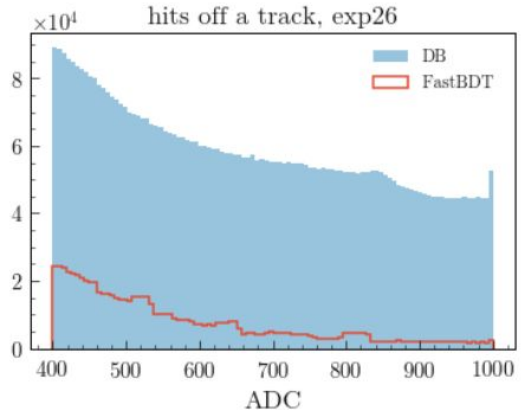
MVA payload need to be added during the reconstruction

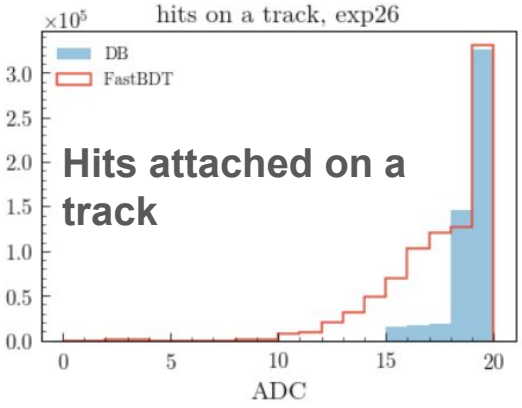
```
Tools: write_tracking_mva_filter_payloads_to_db(
   "trackfindingcdc_FastBDT_ADCFilter_in_CDC", iov, " FastBDT_ADCFilter_in_CDC", 0.4)
```

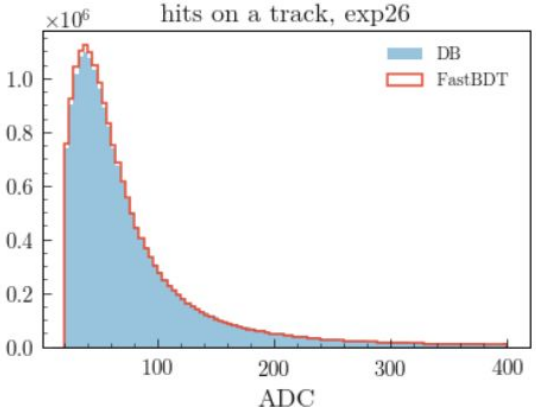
MVA: compare with DB for wirehit level—adc

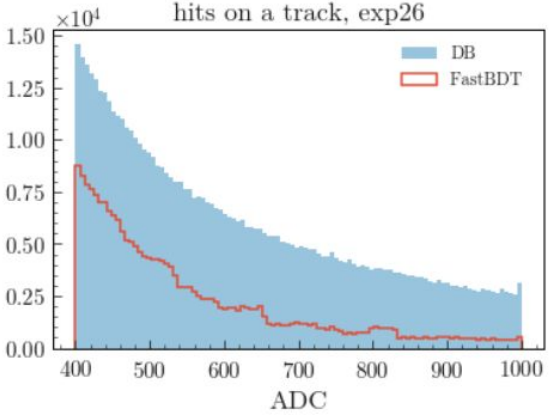






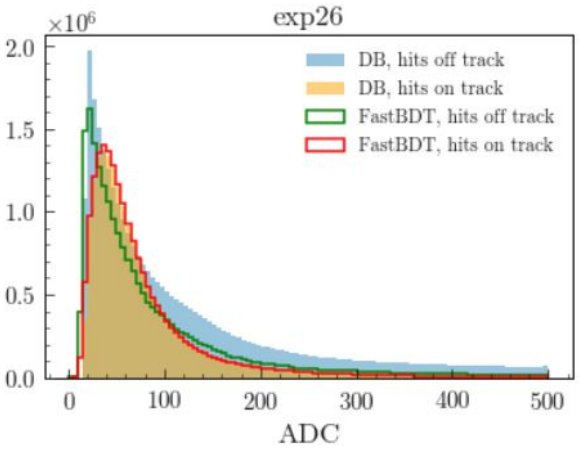


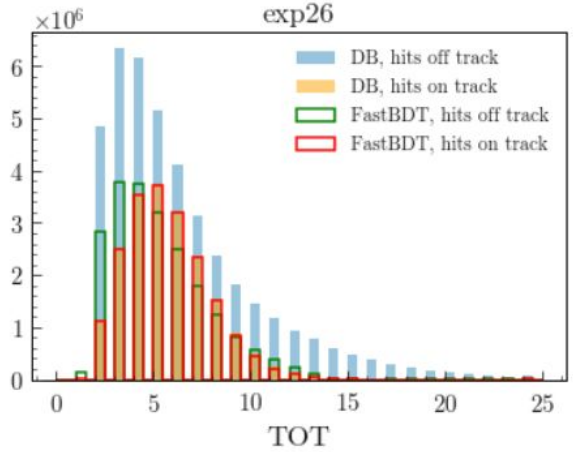


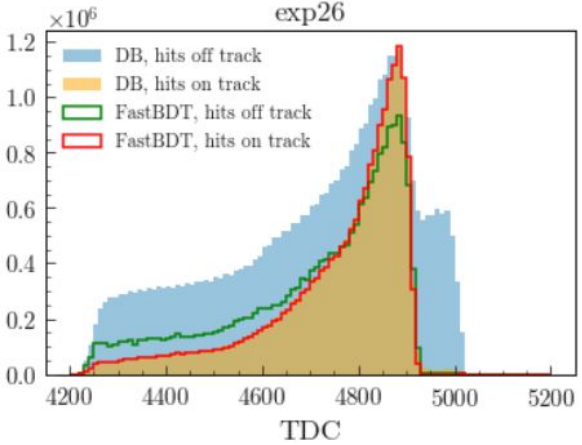


hits off tracks are restricted to those that did not get the background flag

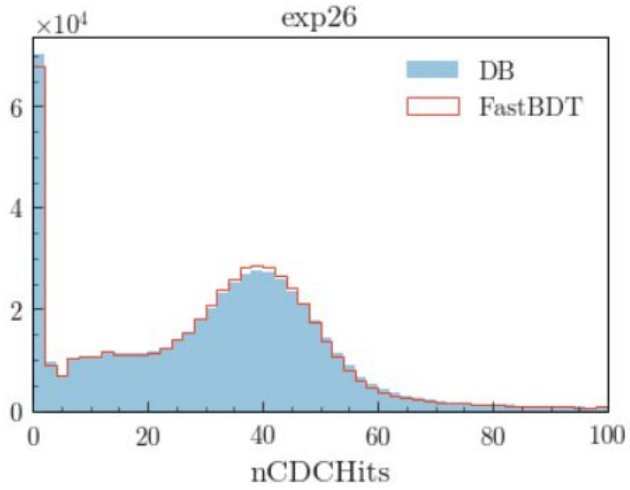
MVA: compare with DB in wirehit level—adc, tot and tdc

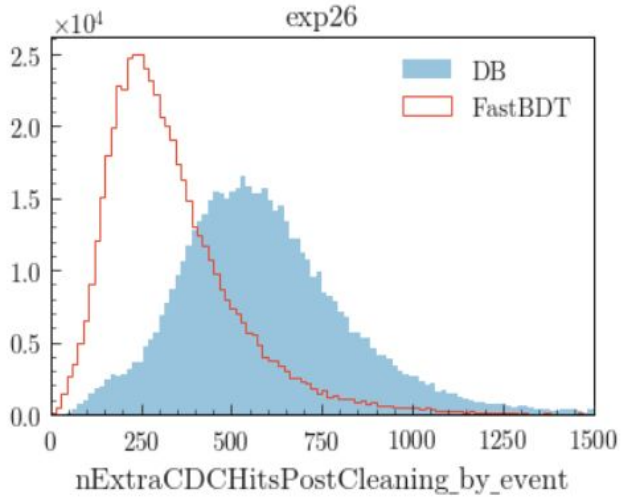






MVA: compare with DB in track and event level

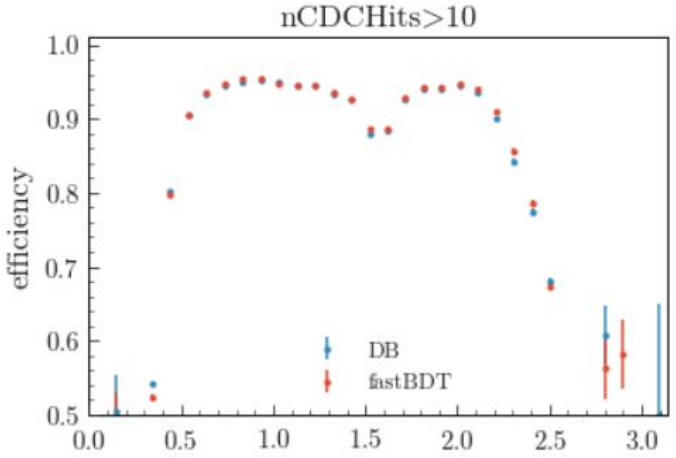


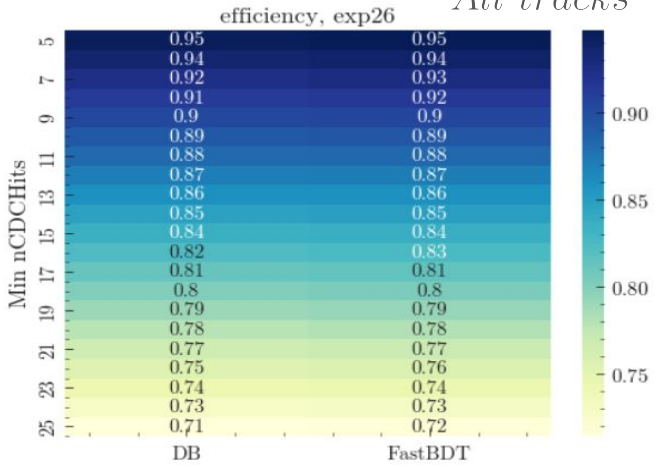


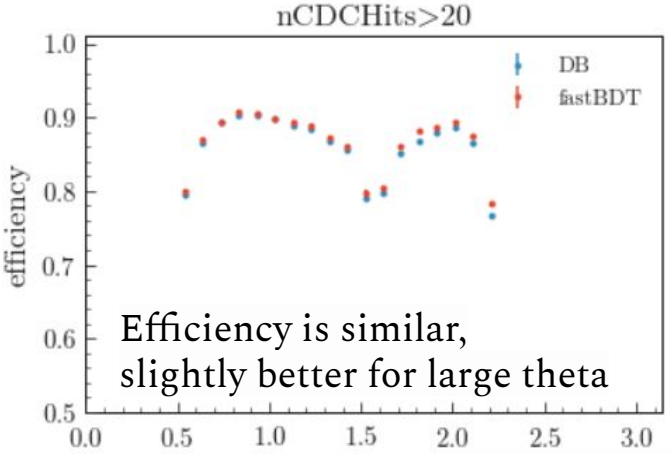
Amount of hits per track is similar,
significant reduction of hits considered for track reconstruction

Track-finding efficiency (vs. theta acceptance)

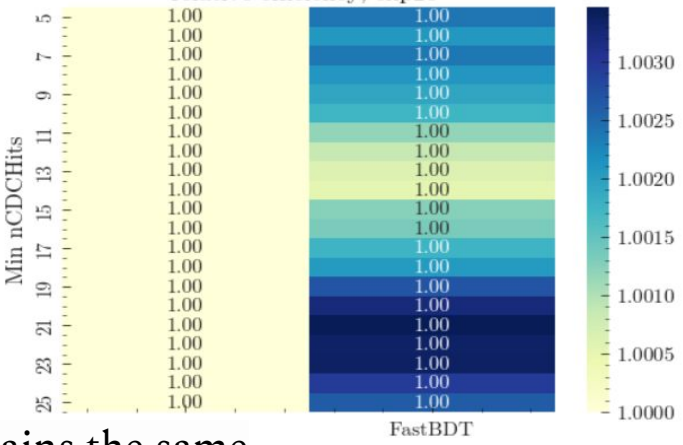
$$\epsilon = \frac{Tracks\ with nCDCHits > n}{All\ tracks}$$







Efficiency is similar,
slightly better for large theta



Efficiency remains the same

Time consumption

XGBoost and FastBDT provide very similar track-finding efficiency. XGBoost is easy to play and implement outside basf2 .

However, xgb consumed more than x1000 to the current DB

(ms)	DB	xgboost (main)
TFCDC_WireHitPrepare	**3.77 +/-1.00**	**4543.61 +/- 1288.58**

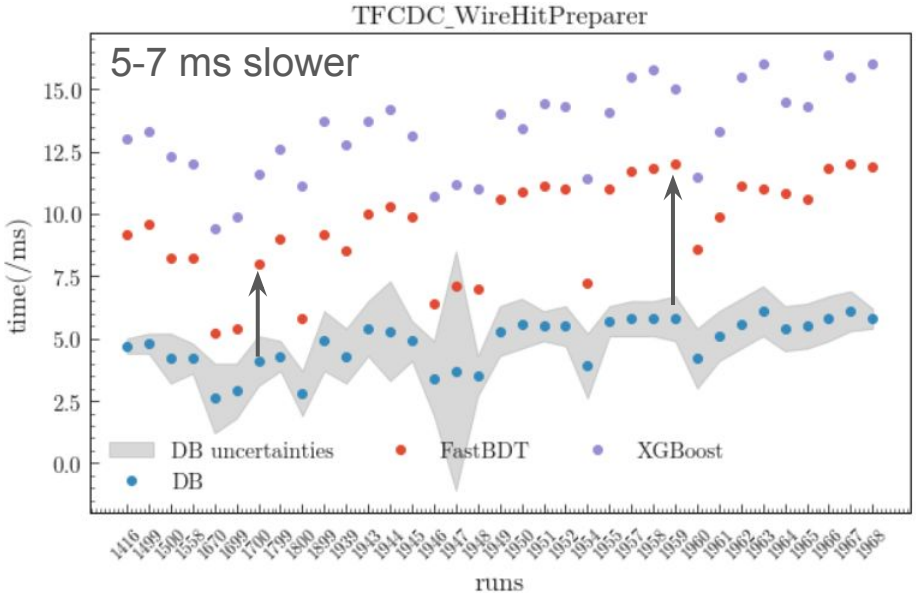
Main reason: python interface and evaluation of background flag tracking MVAExpert
Solution: https://gitlab.desy.de/belle2/software/basf2/-/merge_requests/2977 (A.G)

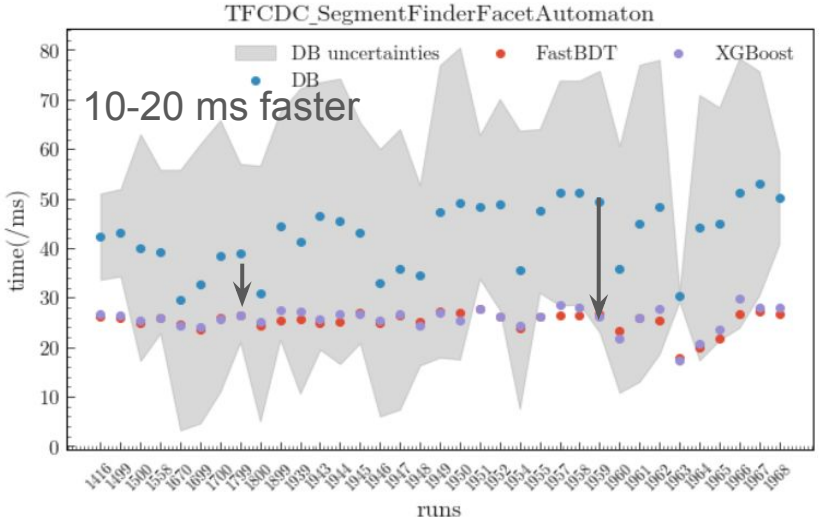
- Introduce a function which using pre-arranged vector of floats to calculate predictions
- Get the orders of training variables from payload

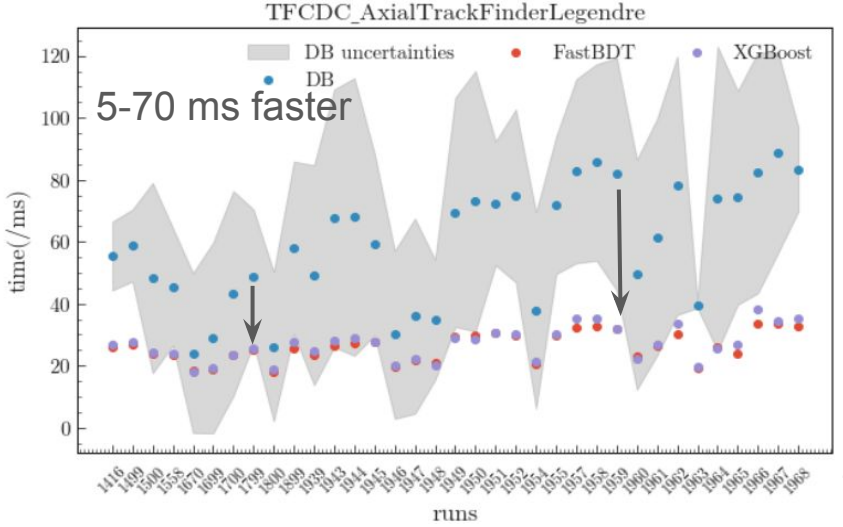
10~15 ms

Use fastBDT as default since it is faster with similar background suppression performance

Time consumption





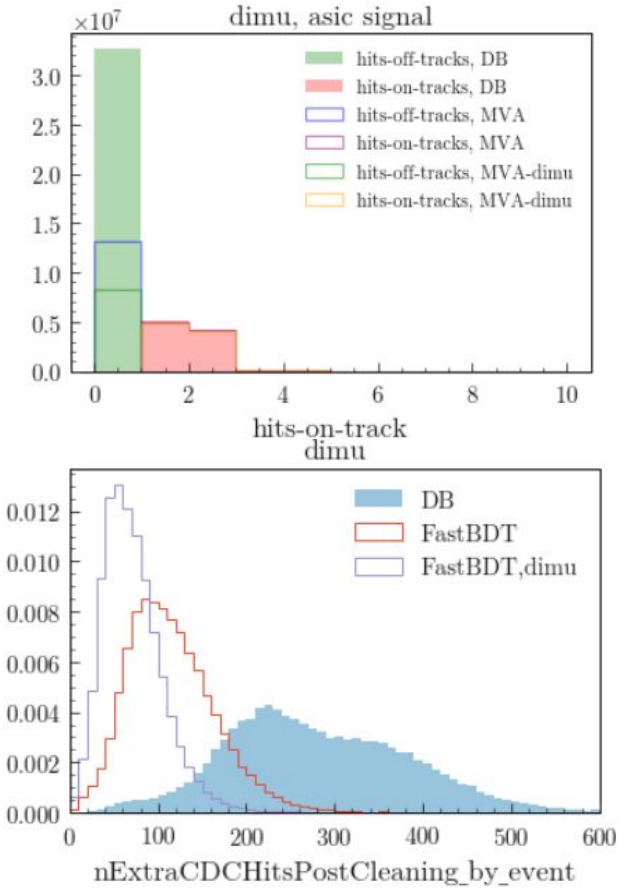


The main CDC tracking algorithms becomes almost twice faster !

Further validation in raw-data/mc

1. Di-muon data/mc
2. MC16RI exp0, 1003, 1004
3. MCRD
4. Exp30 data
5. Proton: lambda data and rimc
6. Resolution of dimu events

Cross-check with di-moun data



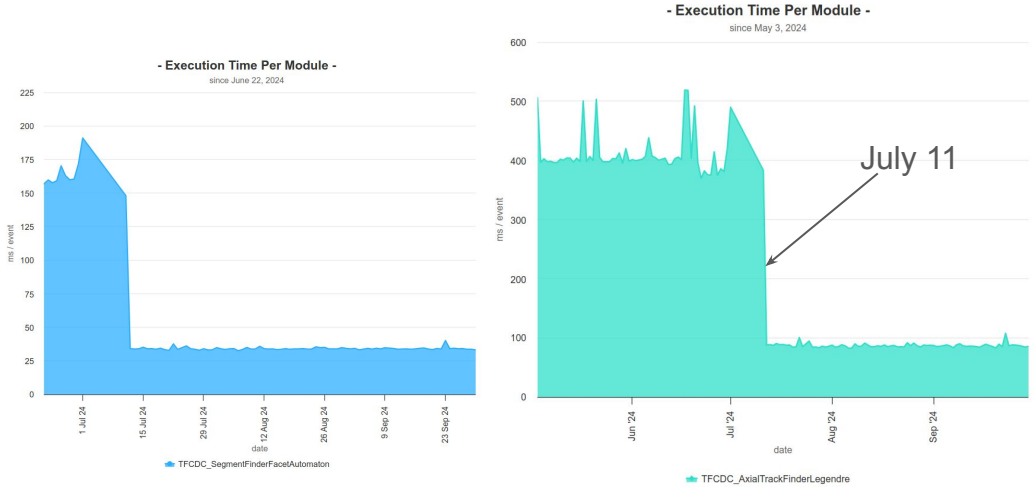
Standard validation: rel8 nominal background, cdc standalone

CDC standalone (%)	DB	hadron MVA	dimu MVA	MC0 MVA
finding_charge_efficiency	**69.1**	**78.5**	**79.6**	**80.8**
finding_efficiency	**71.3**	**80.9**	**81.7**	**83.2**
charge_efficiency	**96.9**	**97.1**	**97.5**	**97.1**
charge_asymmetry	**2.31**	**1.60**	**0.68**	**0.84**
fake_rate	**3.39**	**3.52**	**2.88**	**2.93**
clone_rate	**0.65**	**0.64**	**0.58**	**0.53**
hit_efficiency	**80.6**	**82.4**	**81.0**	**85.7**

Standard validation: rel8 nominal background, full tracking

Full tracking (%)	DB	hadron MVA	dimu MVA	MC0 MVA
finding_charge_efficiency	**90.3**	**91.5**	**91.9**	**92.0**
finding_efficiency	**91.4**	**92.5**	**92.9**	**93.1**
charge_efficiency	**98.8**	**98.9**	**98.9**	**98.8**
charge_asymmetry	**0.37**	**0.34**	**0.07**	**-0.15**
fake_rate	**7.05**	**6.44**	**6.09**	**6.38**
clone_rate	**3.86**	**3.60**	**3.41**	**3.31**
hit_efficiency	**75.4**	**78.5**	**77.8**	**81.1**

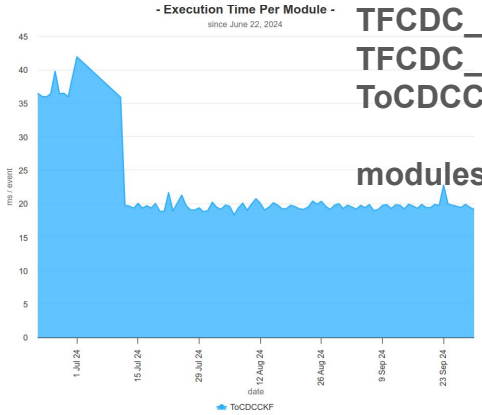
Official validation of B2bot

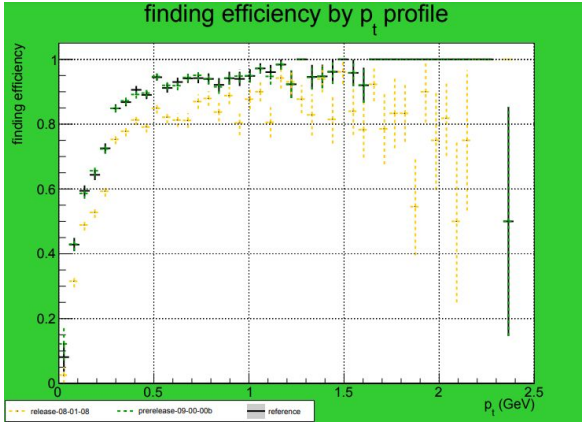


Reduction of execution time of

TFCDC_SegmentFinderFacetAutomaton
TFCDC_AxialTrackFinderLegendre
ToCDCCKF

modules in ms/event

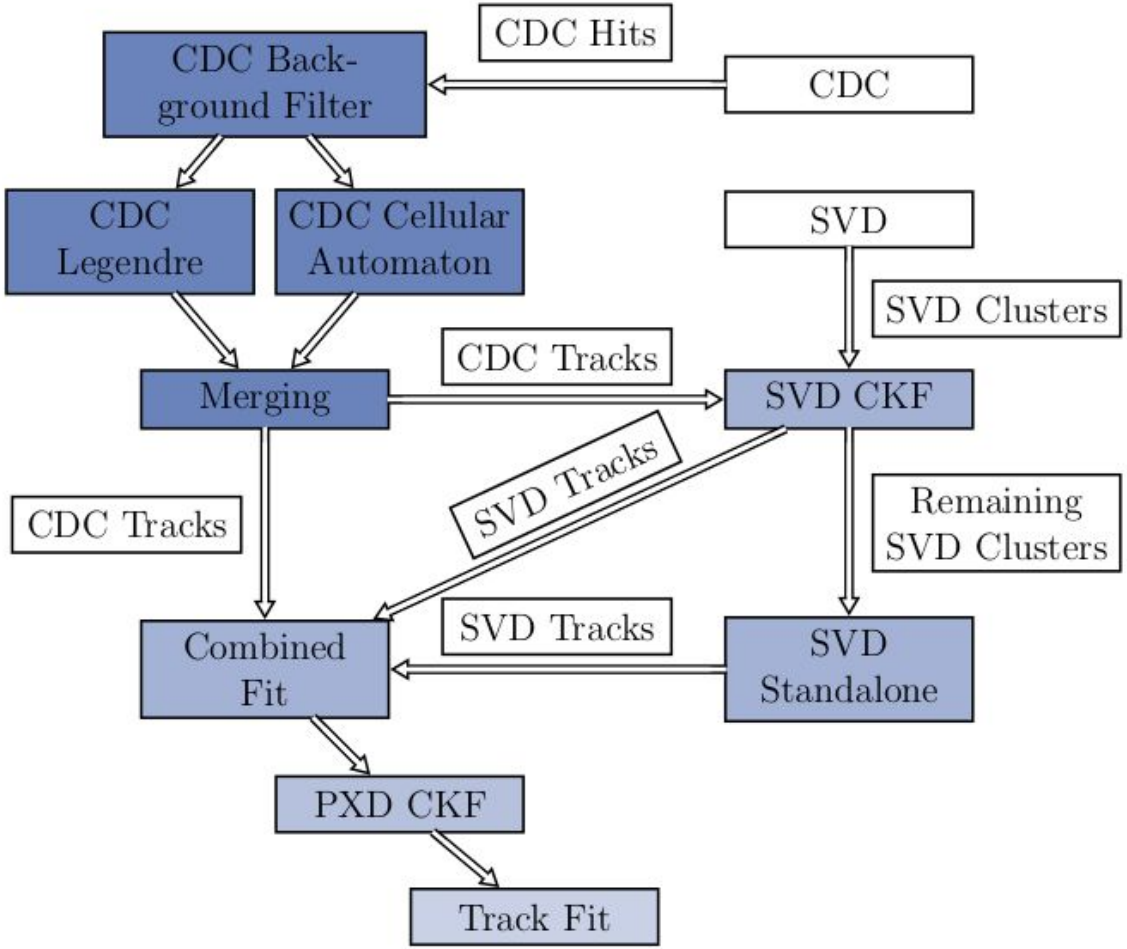




CDC Full-Tracking Validation Bkg

[MR:activate the cdc bkg filter]

BACKUP


CDC Hits
CDC Back-
ground Filter
CDC
CDC
Legendre
CDC Cellular
Automaton
SVD
SVD Clusters
CDC Tracks
Merging
SVD CKF
SVD Tracks
CDC Tracks
Remaining
SVD Clusters
SVD Tracks
Combined
Fit
SVD
Standalone
PXD CKF
Track Fit

Standard validation: items-explanation

- finding_efficiency = matched tracks/all primary tracks
- charge_efficiency = matched tracks with correct charge / matched primary tracks
- finding_charge_efficiency = matched tracks with correct charge / all primary tracks

- fake_rate =pattern recognition tracks that are not related to a particle / all pattern recognition tracks
- clone_rate = ratio of clones divided the number of tracks that are related to a particle (clones and matches)

- charge_asymmetry = (# matched pos - # matched neg) / all matched (pos+neg)
- hit_efficiency = hit efficiency of matched tracks

Hits with bgFlag vs. hits attached on a track