SFT Group Meeting

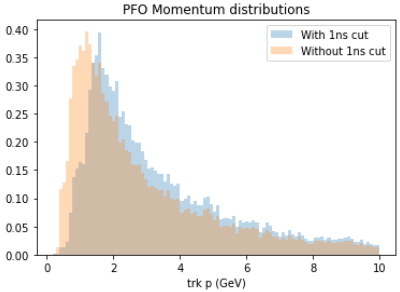
Status Update #12

Konrad Helms

14th September 2023

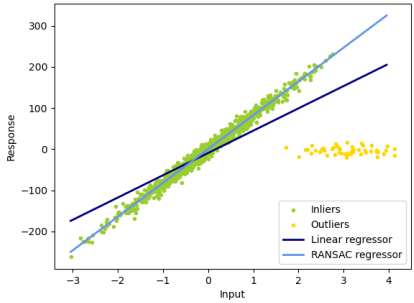
Reminder: Last Week ...

- outlier rejection using the cut:
  Δt(hit time 50 ps, d(IP, hit)/c)$\leq$1 ns
- that indeed introduces a momentum cut!



Now:

- outlier rejection using the RANSAC (RANdom SAmple Consensus) algorithm, which fits a model (here: linear, technically affine, function) from random subsets of inliers from the complete data set.
- visualisation from scikit-learn:

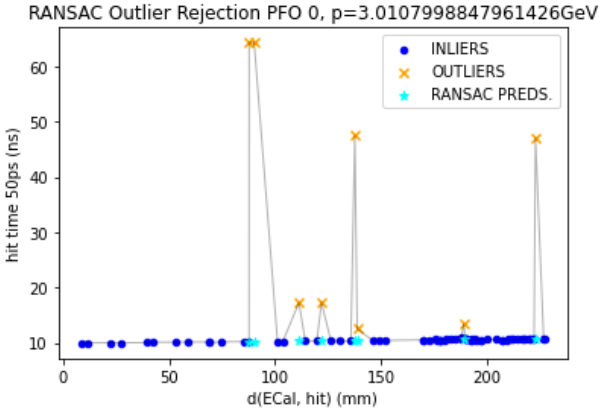


Code:

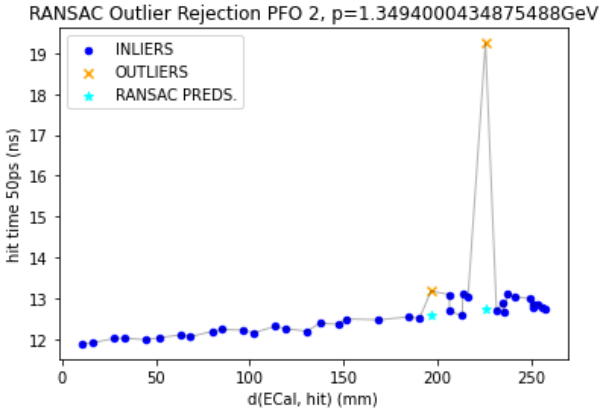
```
def ransac_outlier_rejection(pfo, dt=None):
    '''
    parameters:
        distances
        times
        keep_first_n_hits - the number of first n hits that are never rejected
    Outlier rejection using the RANSAC (RANdom SAmple Consensus) algorithm.
    This function fits a linear (affine) function to the data and rejects
    outliers deviating form the function.
    Since a positive correlation between distance and measured hit times is
    expected, a positive slope is enforced. The intercept of the linear fit
    is the predicted TOF, thus should also be positive. One function is fitted
    per shower, as we are not interested in learning a function on the whole
    dataset, but rather rejecting outliers.

    M. A. Fischler and R. C. Bolles (June 1981). "Random
    Sample Consensus: A Paradigm for Model Fitting with
    Applications to Image Analysis and Automated
    Cartography". Comm. of the ACM 24: 381--395.
    '''
    distances = pfo['d(ECal, hit) (mm)'].to_numpy()
    times = pfo['hit time 50ps (ns)'].to_numpy()
    distances = distances[:, np.newaxis]
    # linear regressor used in the RANSAC algorithm:
    linreg = LinearRegression(positive=True)
    # RANSAC regressor
    ransac = RANSACRegressor(estimator=linreg)
    # fit RANSAC regressor:
    ransac.fit(distances, times)
    # predict hit times at the hit distances
    ransac_preds = ransac.predict(distances)
    # define inliers
    # dt = 0.5 keeps approx 90% of the hits.
    inlier_mask = (times - ransac_preds) <= dt
    # RANSAC predicted TOF
    ransac_tof = ransac.estimator_.intercept_
    return inlier_mask, ransac_tof, ransac_preds
```

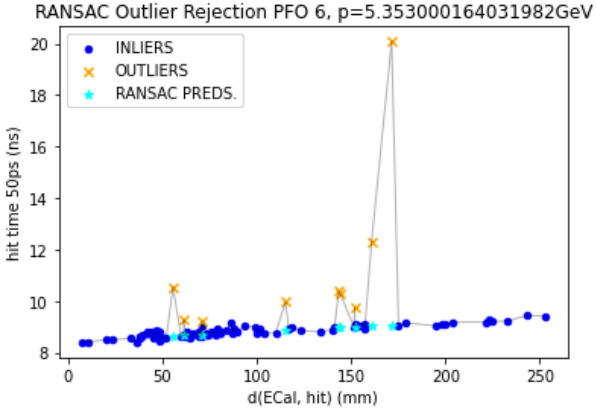
Example #1


RANSAC Outlier Rejection PFO 0, p=3.0107998847961426GeV
INLIERS
OUTLIERS
RANSAC PREDS.
hit time 50ps (ns)
d(ECal, hit) (mm)
60
50
40
30
20
10
0
50
100
150
200

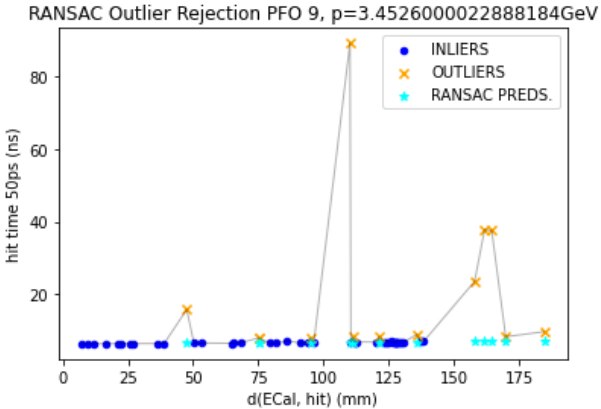
Example #2


RANSAC Outlier Rejection PFO 2, p=1.3494000434875488GeV
INLIERS
OUTLIERS
RANSAC PREDS.
hit time 50ps (ns)
d(ECal, hit) (mm)
12
13
14
15
16
17
18
19
0
50
100
150
200
250

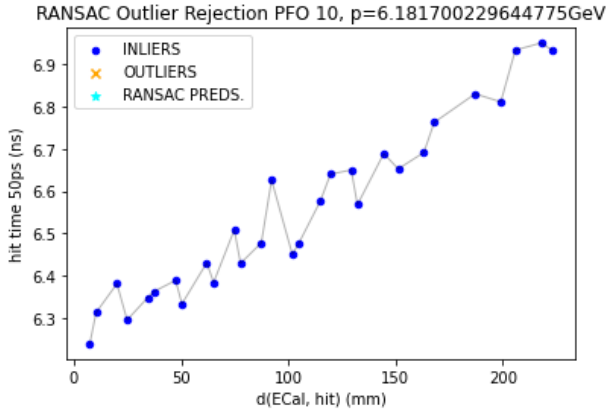
Example #3


RANSAC Outlier Rejection PFO 6, p=5.353000164031982GeV
INLIERS
OUTLIERS
RANSAC PREDS.
hit time 50ps (ns)
d(ECal, hit) (mm)
20
18
16
14
12
10
8
0
50
100
150
200
250

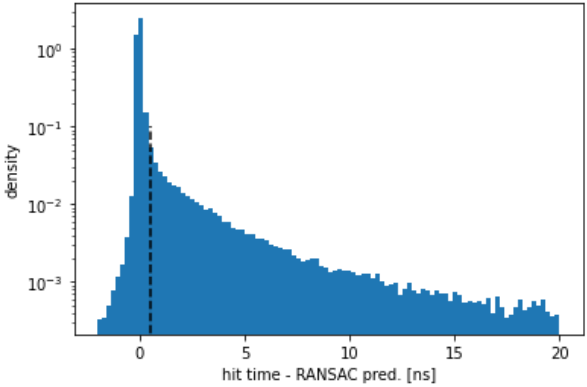
Example #4


RANSAC Outlier Rejection PFO 9, p=3.4526000022888184GeV
INLIERS
OUTLIERS
RANSAC PREDS.
hit time 50ps (ns)
d(ECal, hit) (mm)
0
25
50
75
100
125
150
175
20
40
60
80

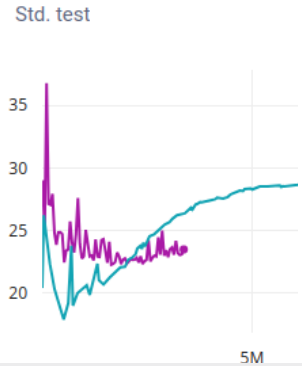
Example #5

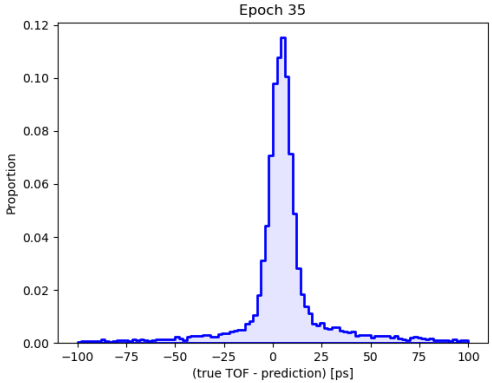

RANSAC Outlier Rejection PFO 10, p=6.181700229644775GeV
INLIERS
OUTLIERS
RANSAC PREDS.
hit time 50ps (ns)
6.9
6.8
6.7
6.6
6.5
6.4
6.3
0
50
100
150
200
d(ECal, hit) (mm)

Rejecting approx. 10% of hits

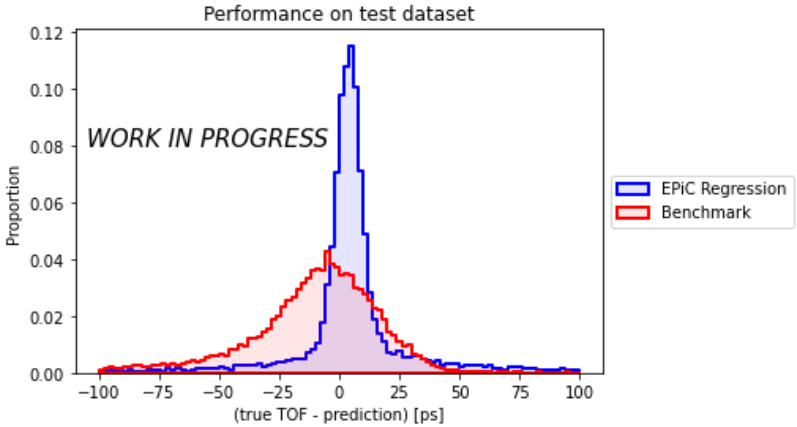


Training looks worse with RANSAC Rejection than with 1ns cut (as expected)





Performance Comparison to Benchmark



Outlook

- do RANSAC in 4D -> x,y,z vs. hit time (could potentially improve hit rejection)