HF jets analysis

16.09.2020 ALICE@IFJ meeting

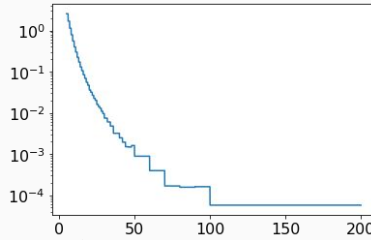
Sebastian Bysiak

Outline

1. Stability improvement:
    - reweighting with GB reweighter
    - training in bins of pT (*multimodel*)

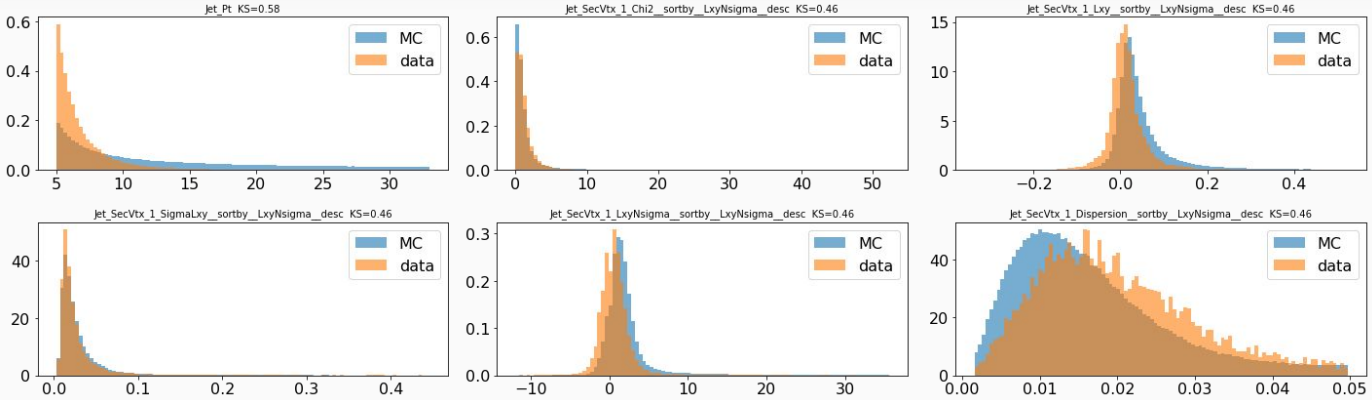
Reweighting with GB reweighter

- IDEA: instead of reweighting in bins (curse of dimensionality!)
  use BDT to determine proper splitting of the feature space
- proposed to validate with 2-sample Kolmogorow-Smirnov test
- https://arxiv.org/abs/1608.05806 , https://github.com/arogozhnikov/hep_ml



- in my case: 84 dims, several hundreds of samples used for reweighting
- KS greatly improved, when compared to “no reweighting”
  i.e. data with MC in hard-pt-bins
- just weighting MC using pT already improves the KS scores significantly (= ”standard”)
- one trial may not be enough as this is another BDT which can be tuned

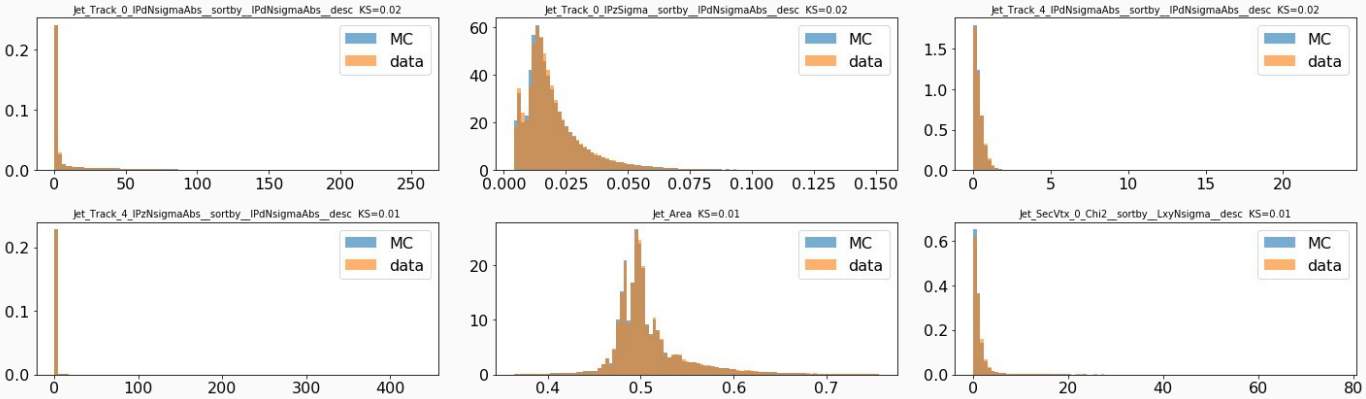
Reweighting with GB reweighter



6 most differing (out of 84) features shown

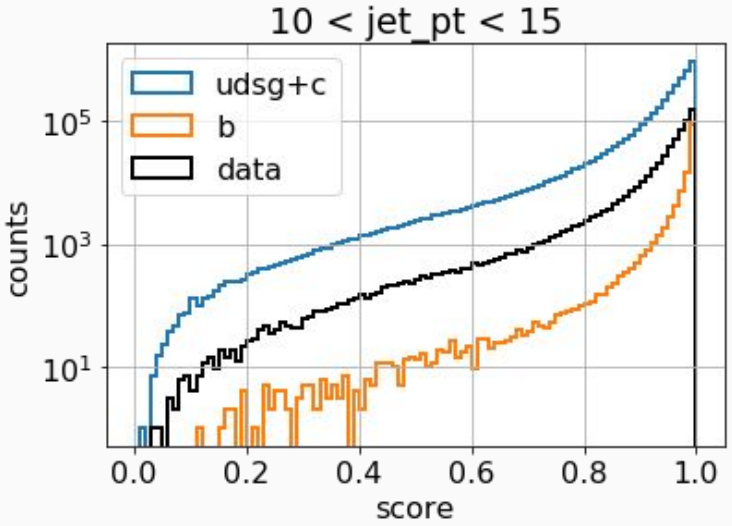
<- before reweighting

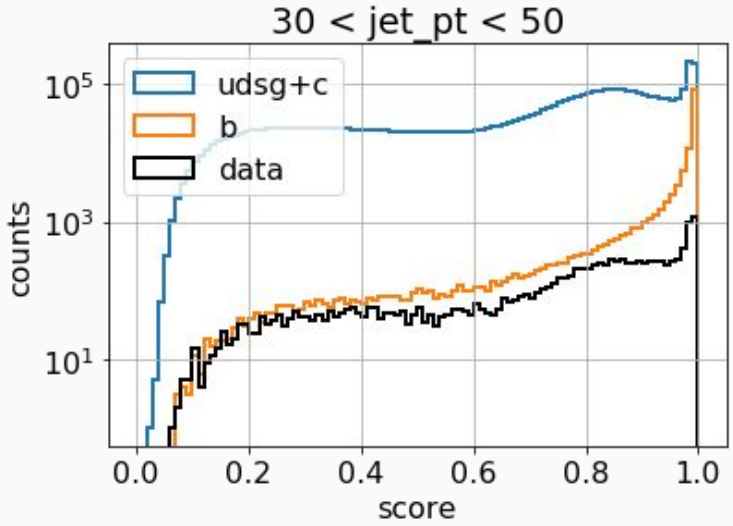
after ->



training in bins of pT (multimodel)

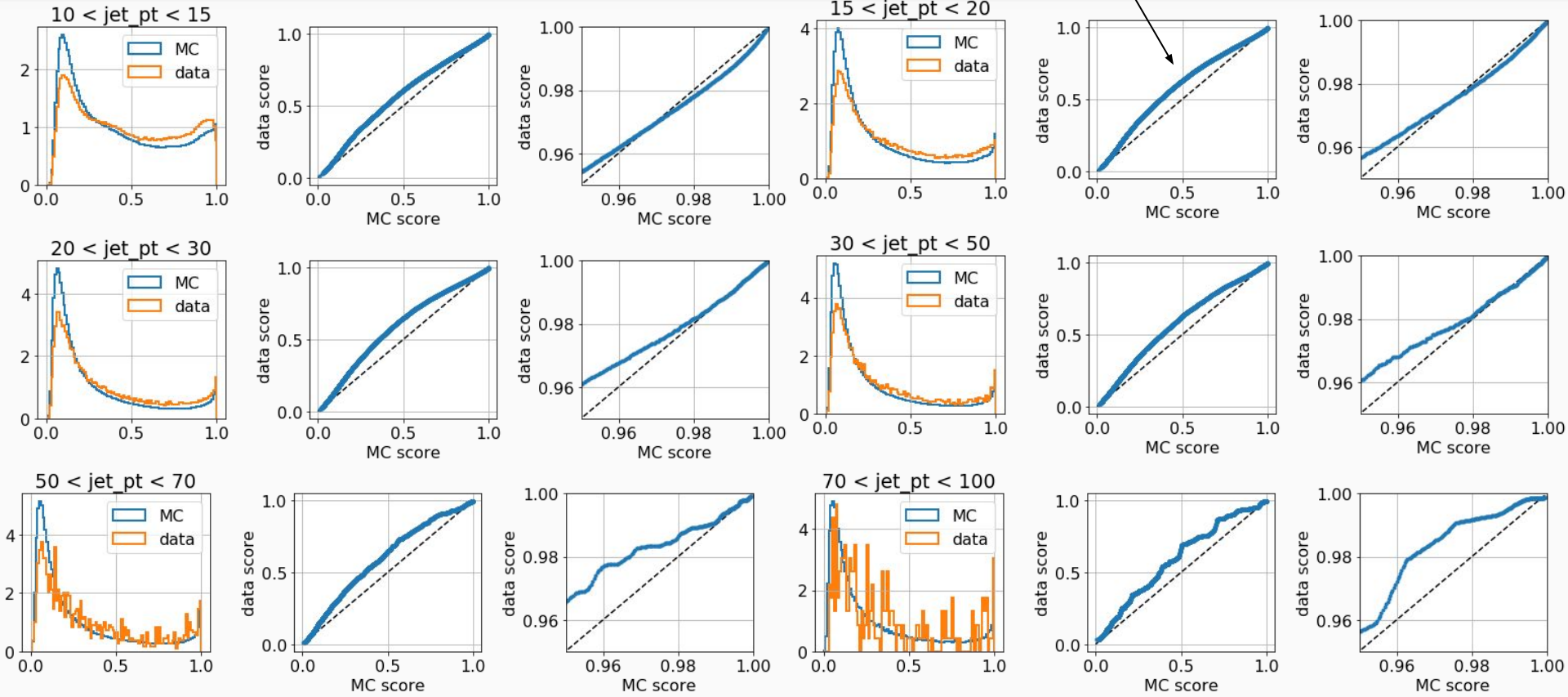
- training is 6 bins: pT = 5, 10, 20, 30, 50, 100, 200 GeV/*c* ;
  > 500k samples for each but last bin
- managed to run it over night - not well validated in terms of overfitting etc
- I got strange, skewed score distributions:





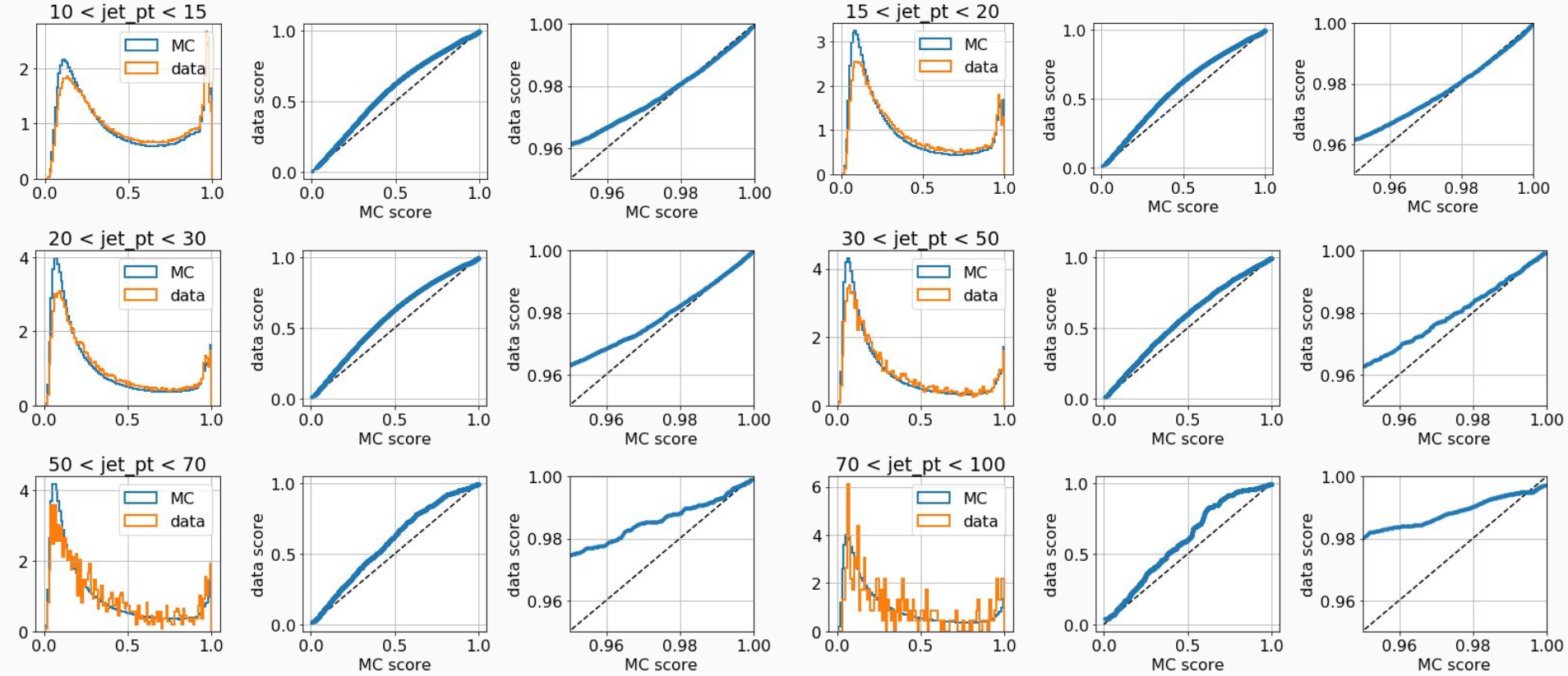
Score distr.

each dot = score values in data and MC corresponding to this quantile

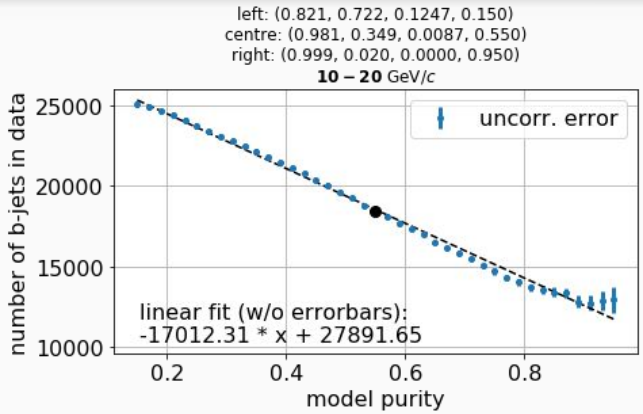


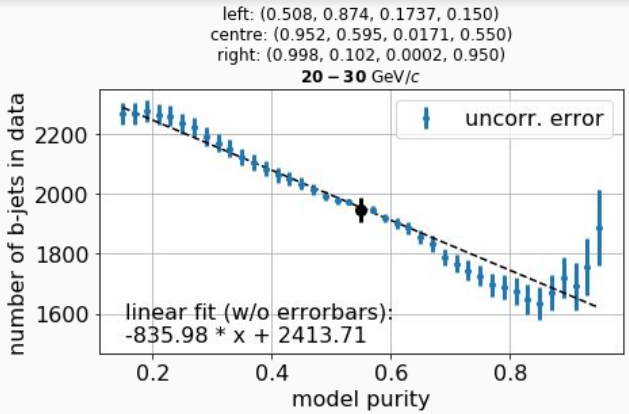
Score distr. (reweighted)

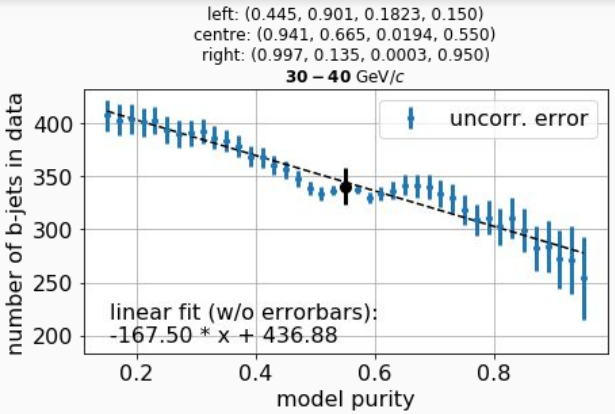
different score distr. shapes in general
slightly smaller mismatch data-MC?

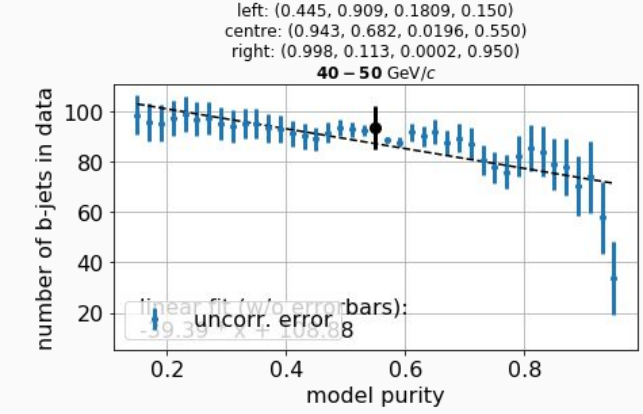


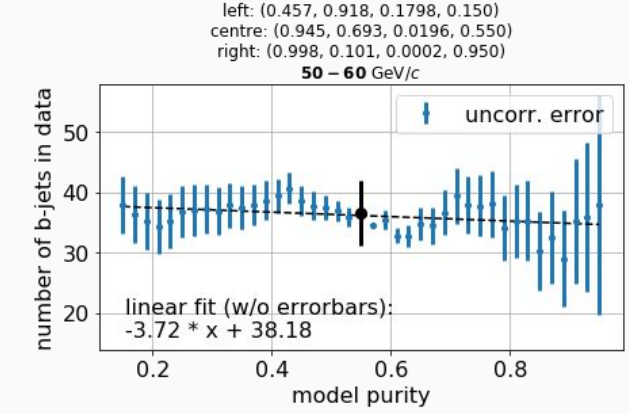
0.15 < P < 0.95, standard

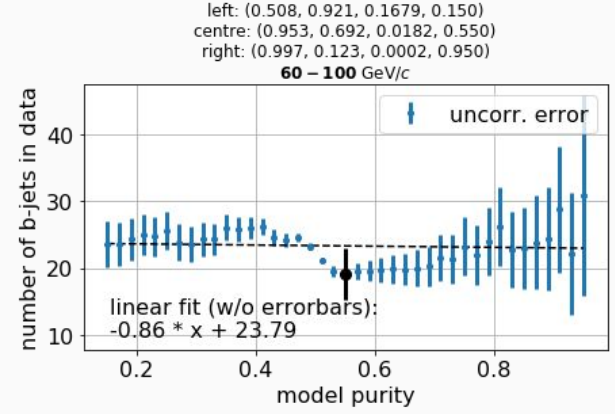




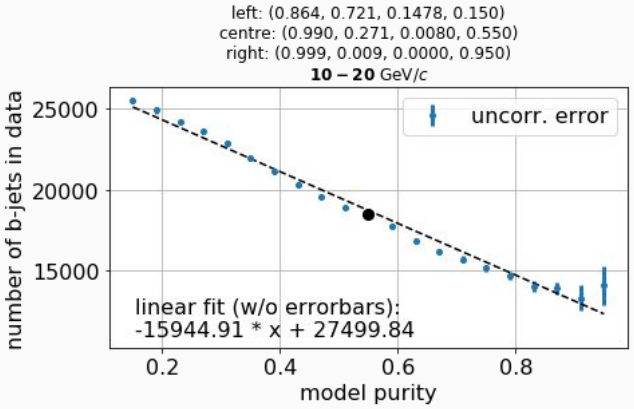


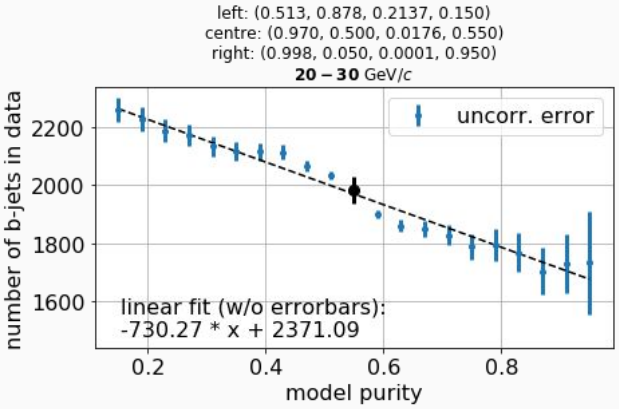


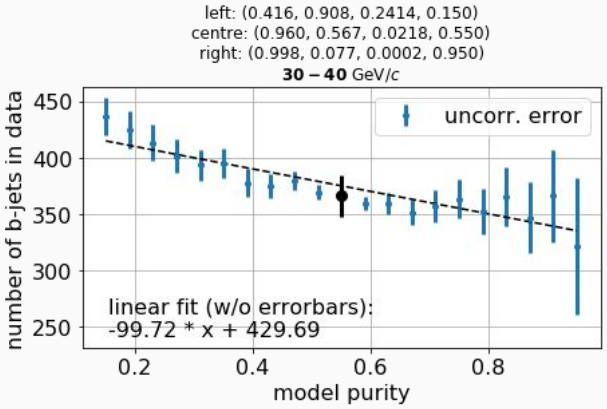


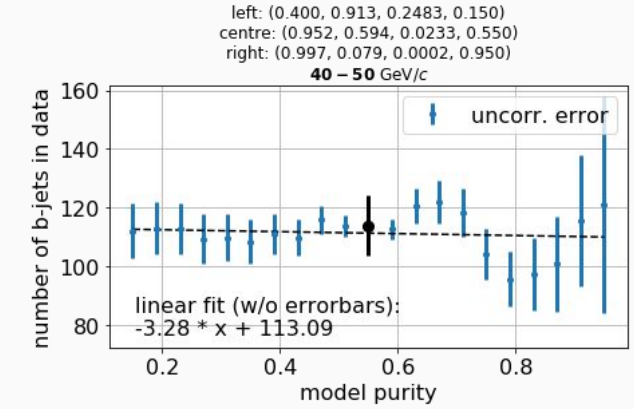


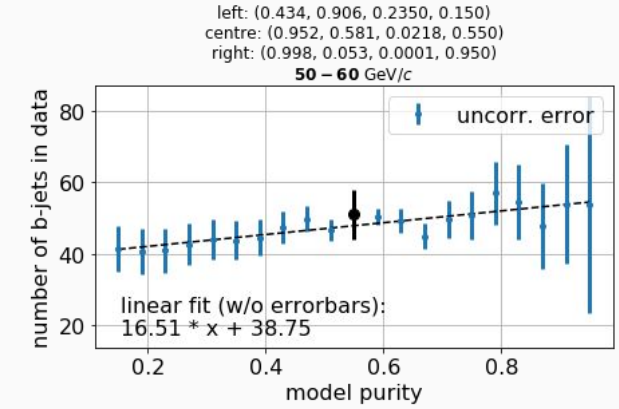
0.15 < P < 0.95, reweighted

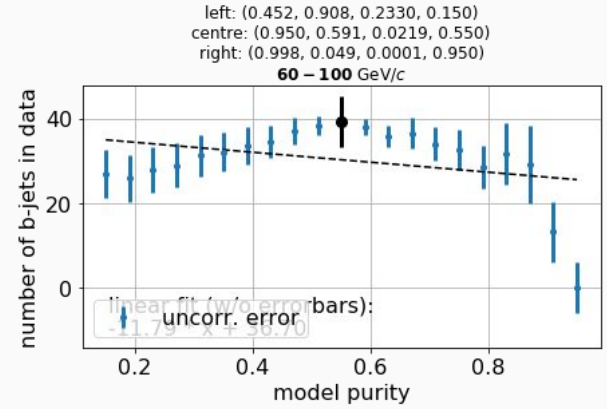




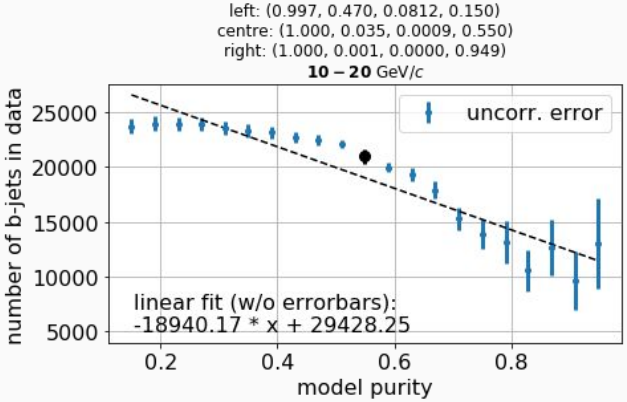


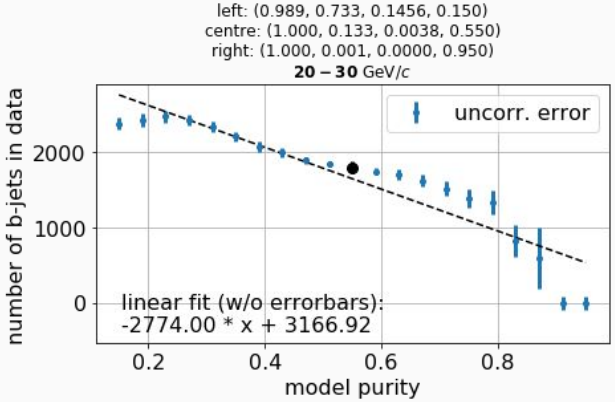


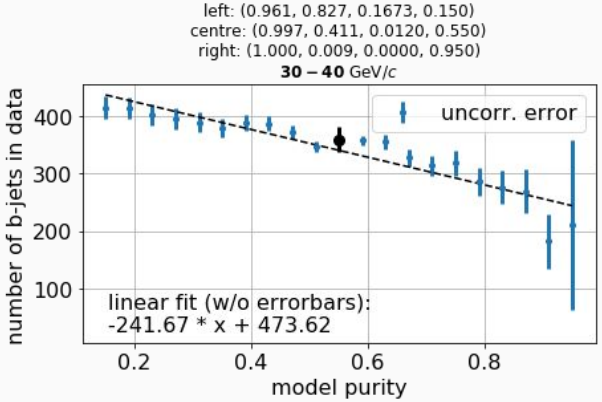


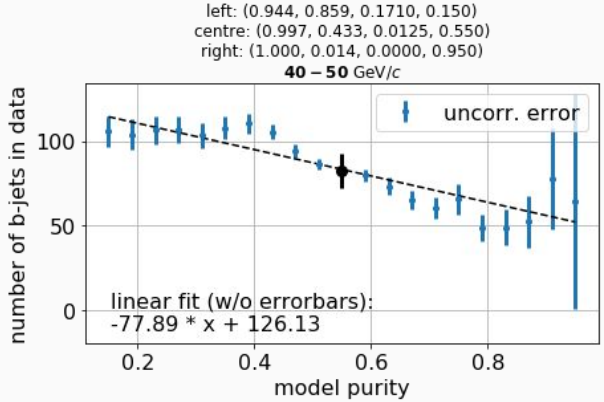


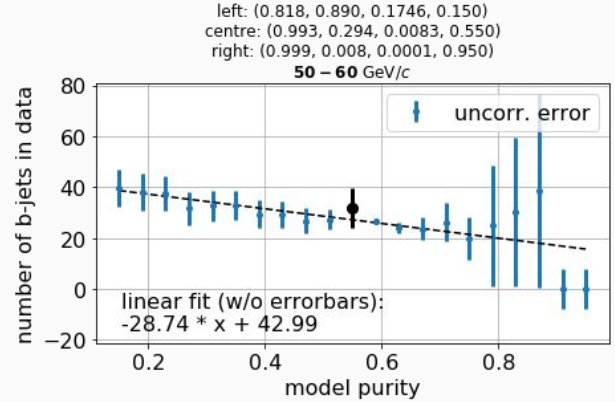
0.15 < P < 0.95, multimodel

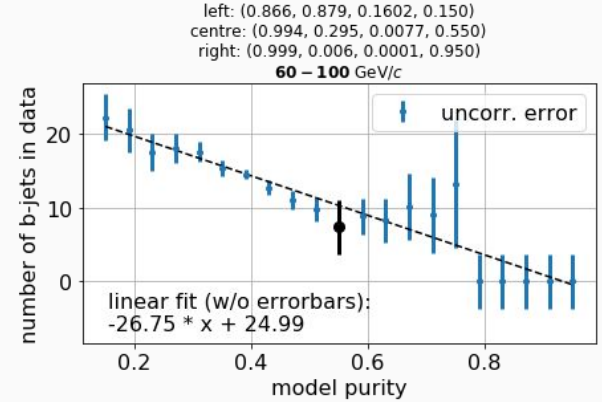






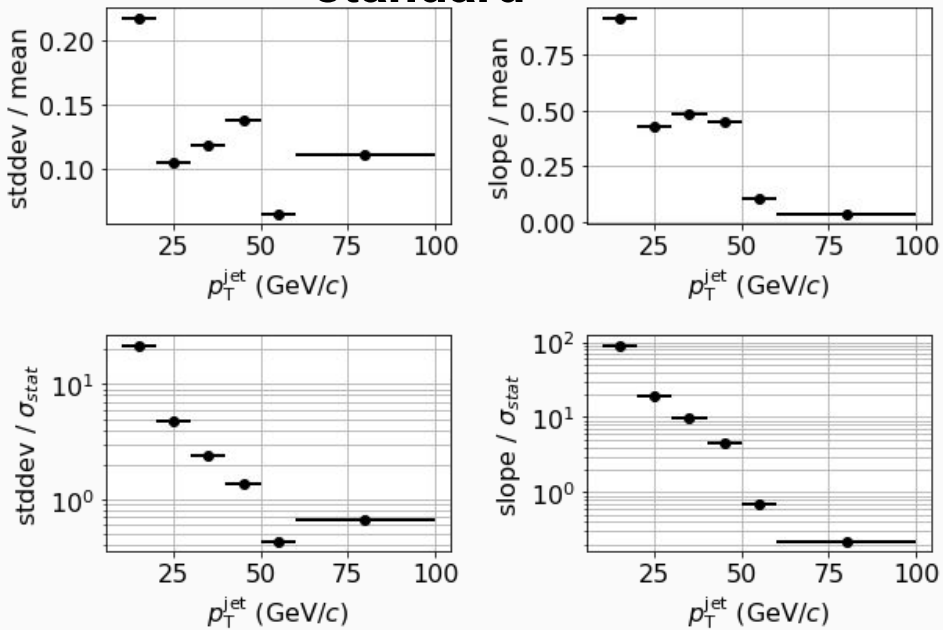




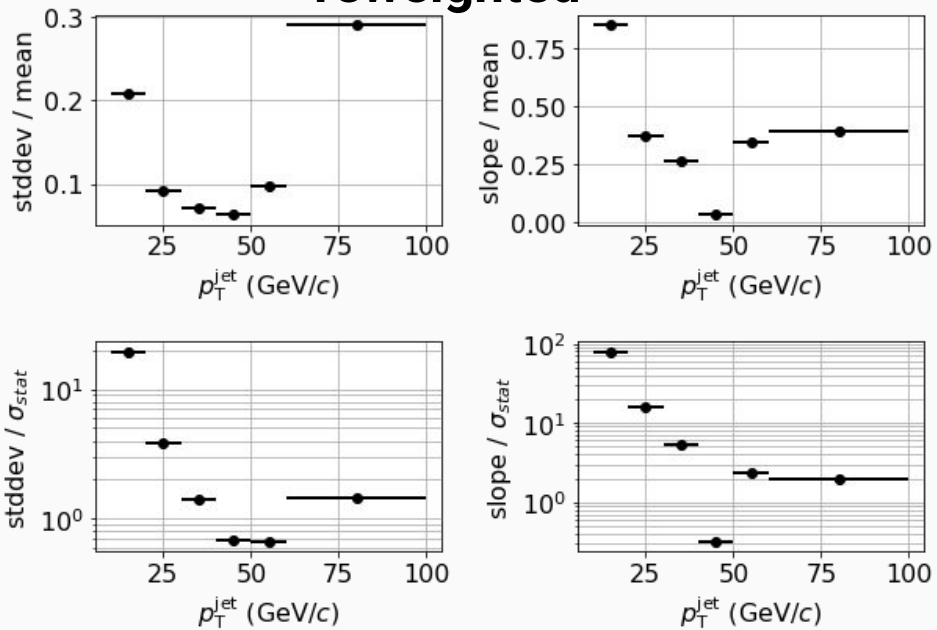


0.15 < P < 0.95, standard VS reweighted

standard



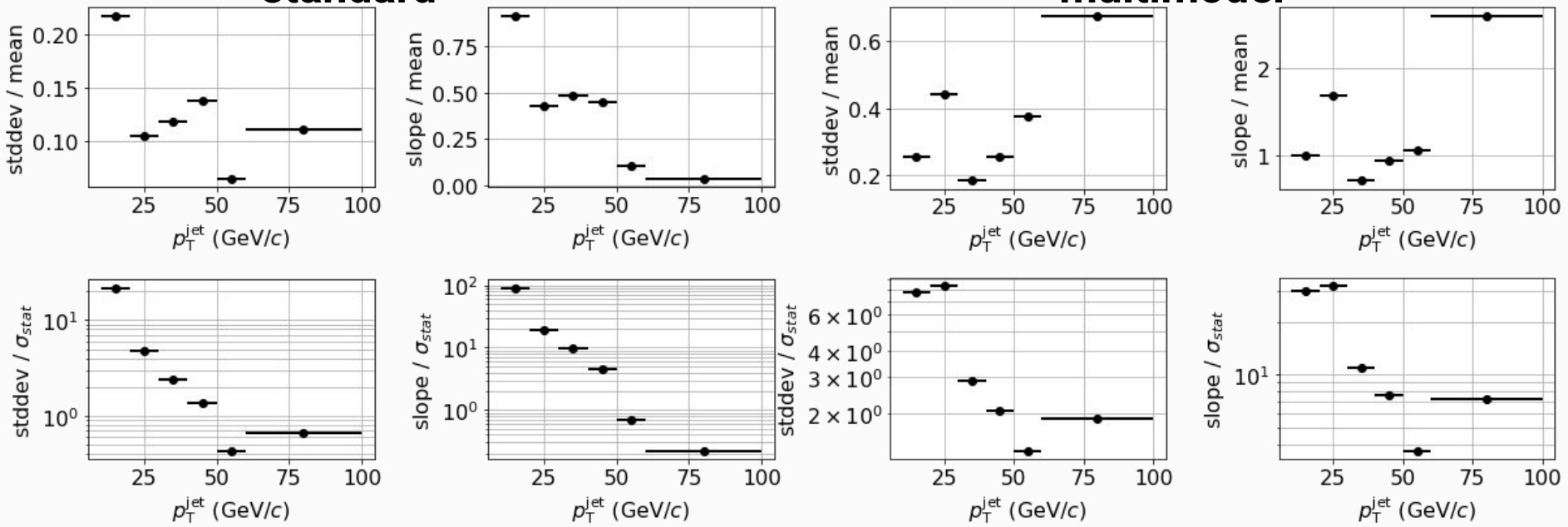
reweighted



rather similar performance
reweighted: better at mid-pT

0.15 < P < 0.95, standard VS multimodel

standard **multimodel**



multimodel: performance much more flat in pT, but usually simply worse than standard

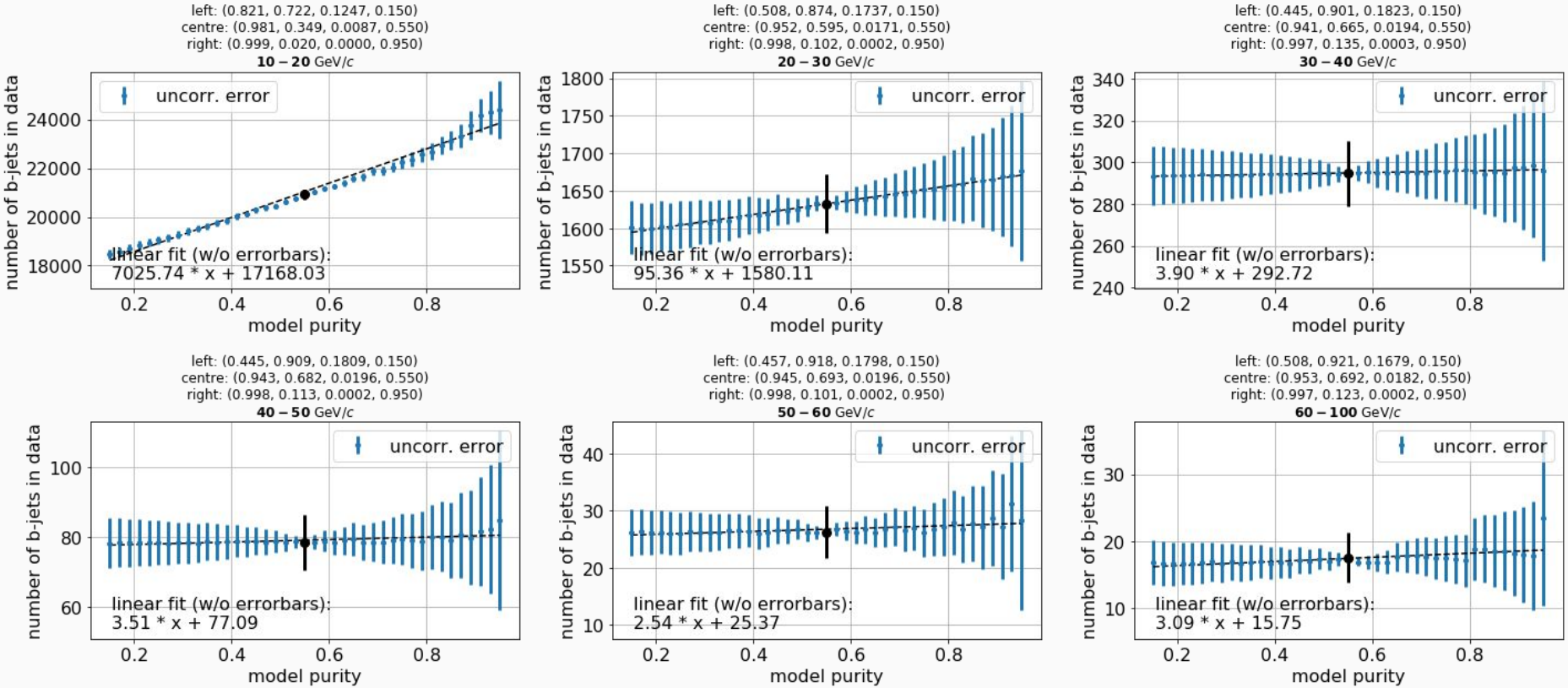
Summary

- no significant improvement with reweighting or multimodel
- both can be optimized as one-shot tried so far
- QQ method show last week yields best results so far, but the other options would be prefered in case of comparable performance

JacekO: other idea for reweighting? (yesterday slack)

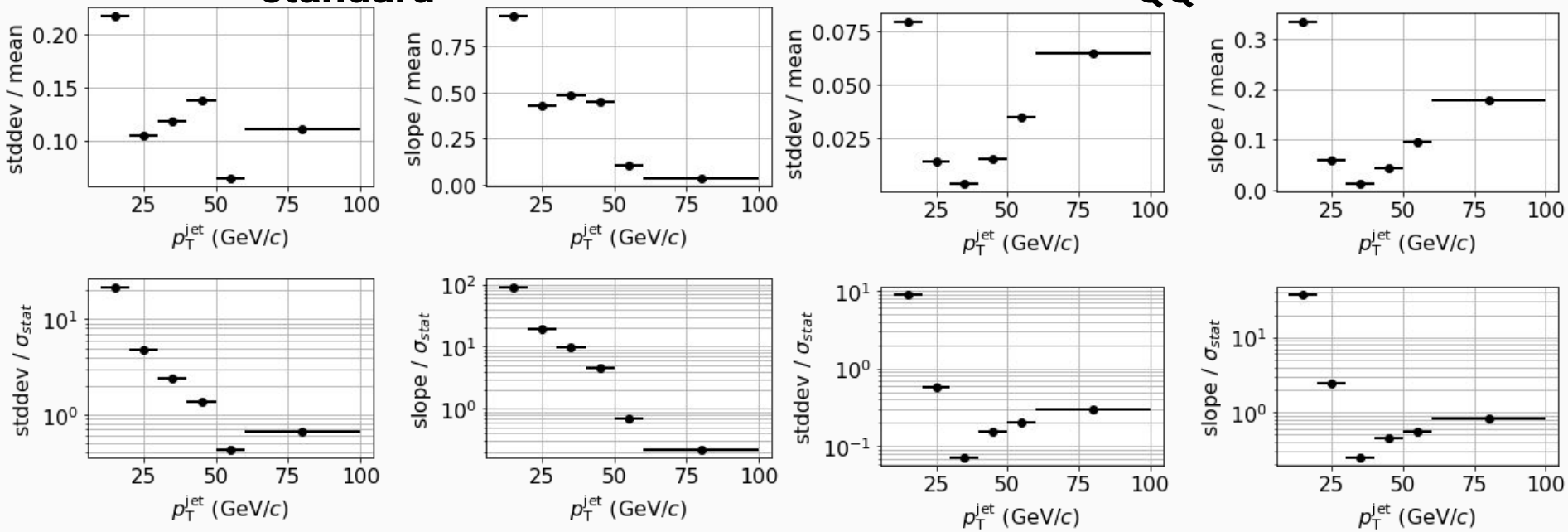
Backup

0.15 < P < 0.95, QQ

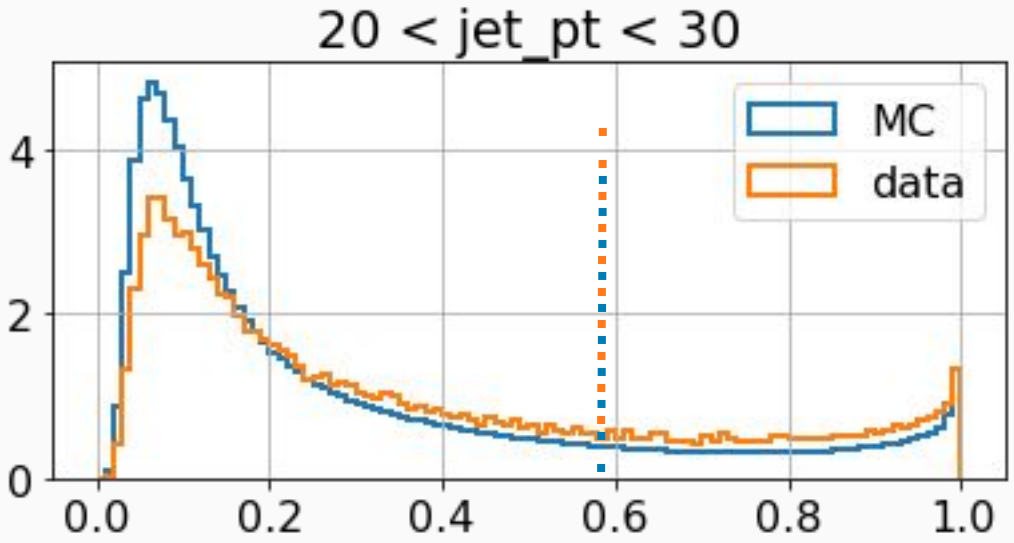

left: (0.821, 0.722, 0.1247, 0.150)
centre: (0.981, 0.349, 0.0087, 0.550)
right: (0.999, 0.020, 0.0000, 0.950)
10 – 20 GeV/c
uncorr. error
linear fit (w/o errorbars):
7025.74 * x + 17168.03
number of b-jets in data
model purity
left: (0.508, 0.874, 0.1737, 0.150)
centre: (0.952, 0.595, 0.0171, 0.550)
right: (0.998, 0.102, 0.0002, 0.950)
20 – 30 GeV/c
uncorr. error
linear fit (w/o errorbars):
95.36 * x + 1580.11
number of b-jets in data
model purity
left: (0.445, 0.901, 0.1823, 0.150)
centre: (0.941, 0.665, 0.0194, 0.550)
right: (0.997, 0.135, 0.0003, 0.950)
30 – 40 GeV/c
uncorr. error
linear fit (w/o errorbars):
3.90 * x + 292.72
number of b-jets in data
model purity
left: (0.445, 0.909, 0.1809, 0.150)
centre: (0.943, 0.682, 0.0196, 0.550)
right: (0.998, 0.113, 0.0002, 0.950)
40 – 50 GeV/c
uncorr. error
linear fit (w/o errorbars):
3.51 * x + 77.09
number of b-jets in data
model purity
left: (0.457, 0.918, 0.1798, 0.150)
centre: (0.945, 0.693, 0.0196, 0.550)
right: (0.998, 0.101, 0.0002, 0.950)
50 – 60 GeV/c
uncorr. error
linear fit (w/o errorbars):
2.54 * x + 25.37
number of b-jets in data
model purity
left: (0.508, 0.921, 0.1679, 0.150)
centre: (0.953, 0.692, 0.0182, 0.550)
right: (0.997, 0.123, 0.0002, 0.950)
60 – 100 GeV/c
uncorr. error
linear fit (w/o errorbars):
3.09 * x + 15.75
number of b-jets in data
model purity

0.15 < P < 0.95, standard VS QQ

standard **QQ**

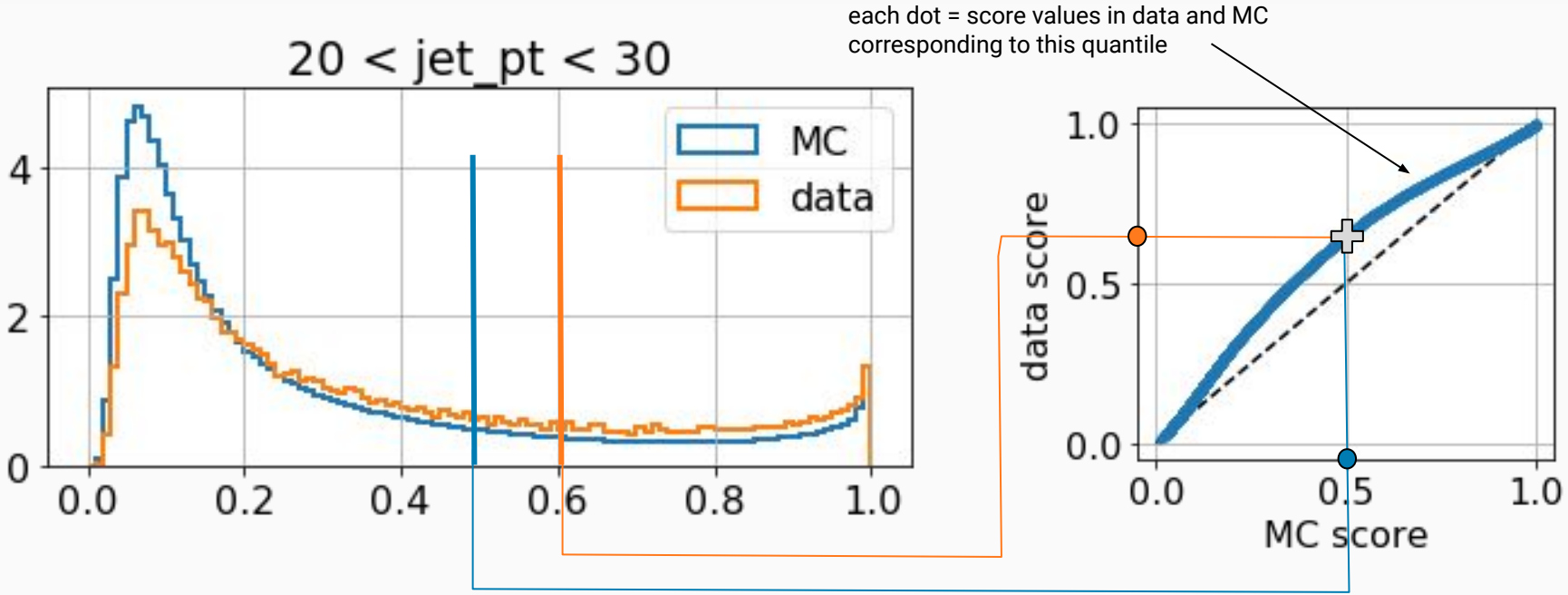


Old way of tagging



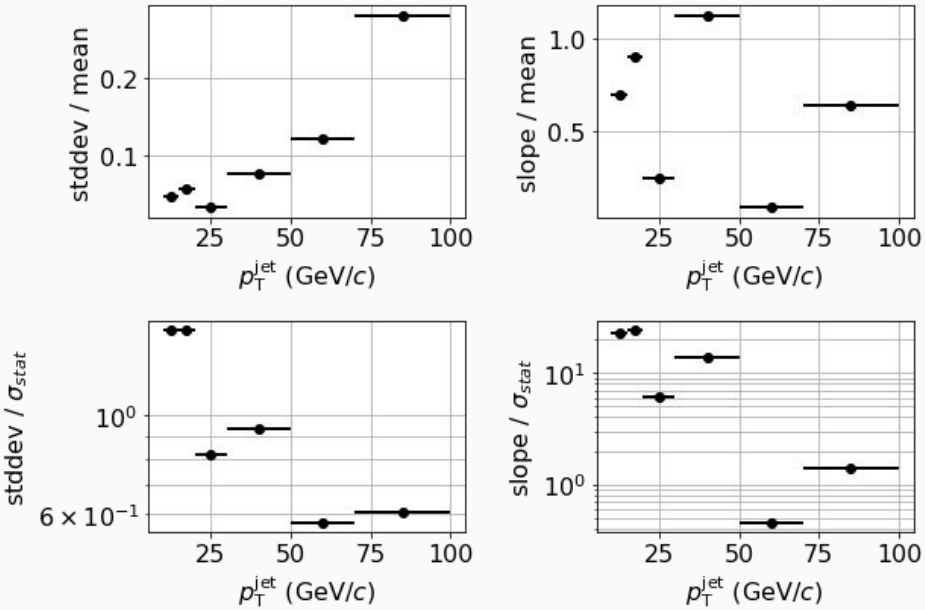
same cut value for data and MC

“QQ method”



different cut value for data and MC
OR
same cut value if our observable is score quantile instead of score

0.75 < P < 0.95

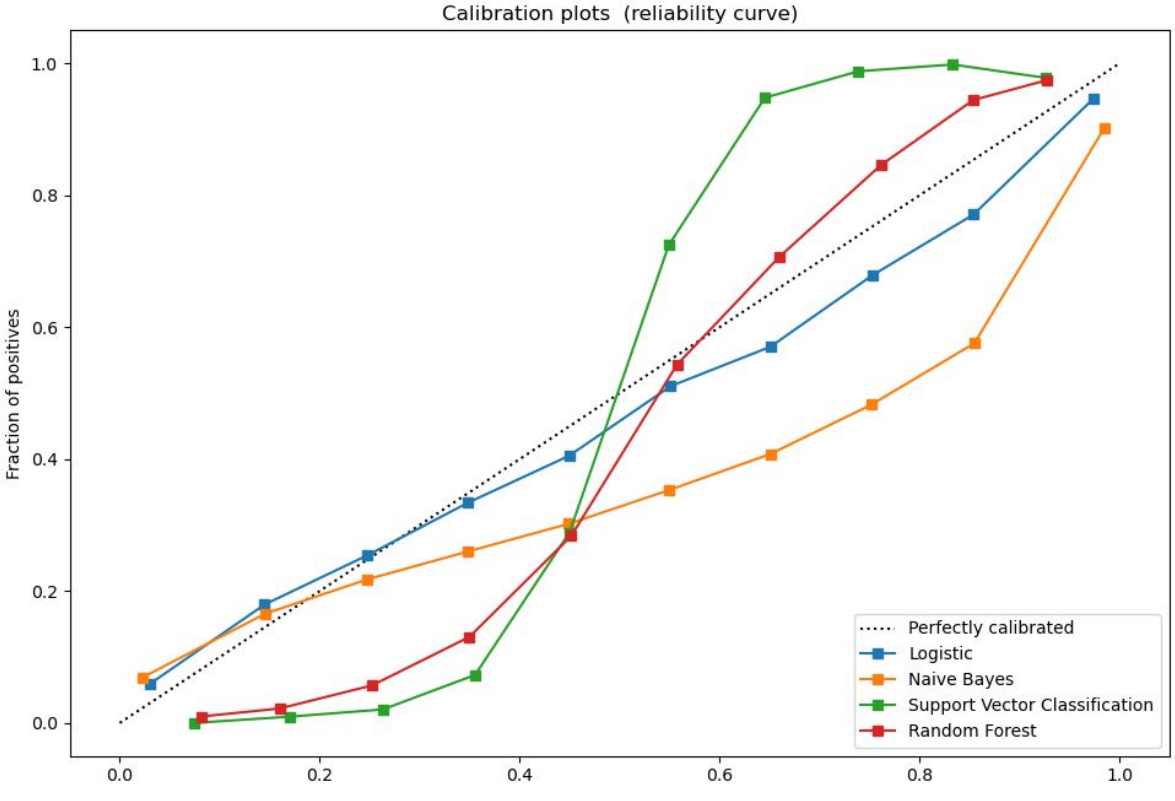


right: stddev ~ 1 σ_{sys}

left: slope ~ maximal deviation, in range 0 < P < 1

top row: compared to mean

bottom row: compared to σ_{stat}



maybe model does not assign proper score, but it at least sorts the jets correctly