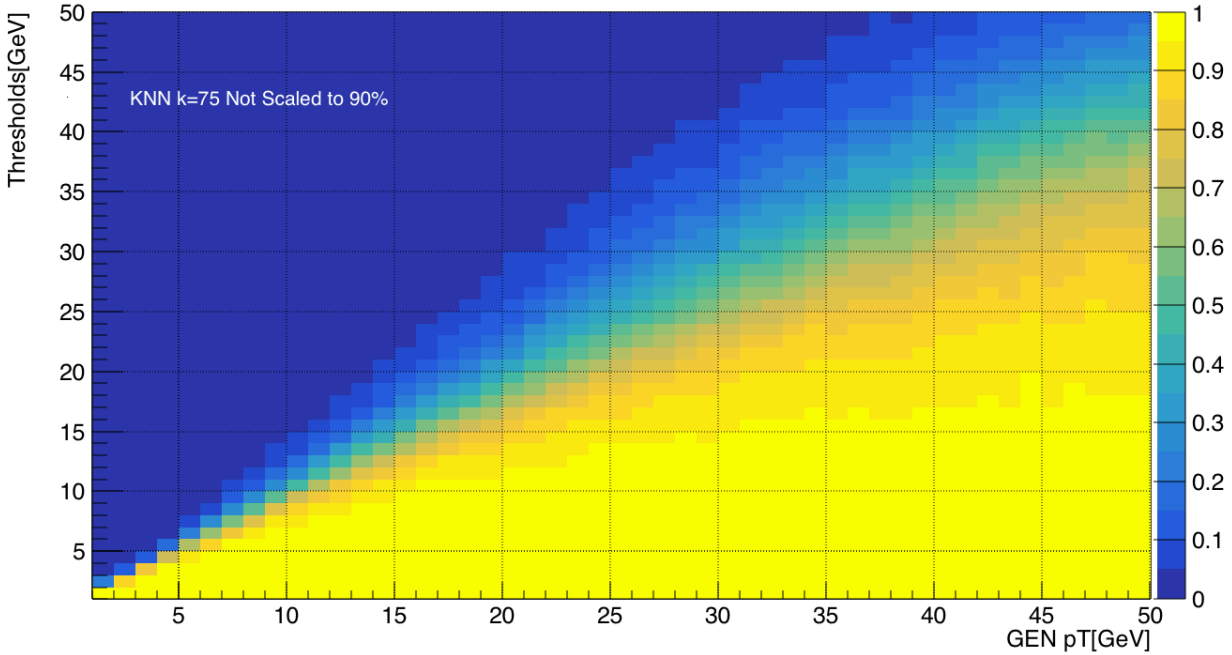
Mode 14 pT Training Result using k-NN

Wei Shi, Jamal Rorie

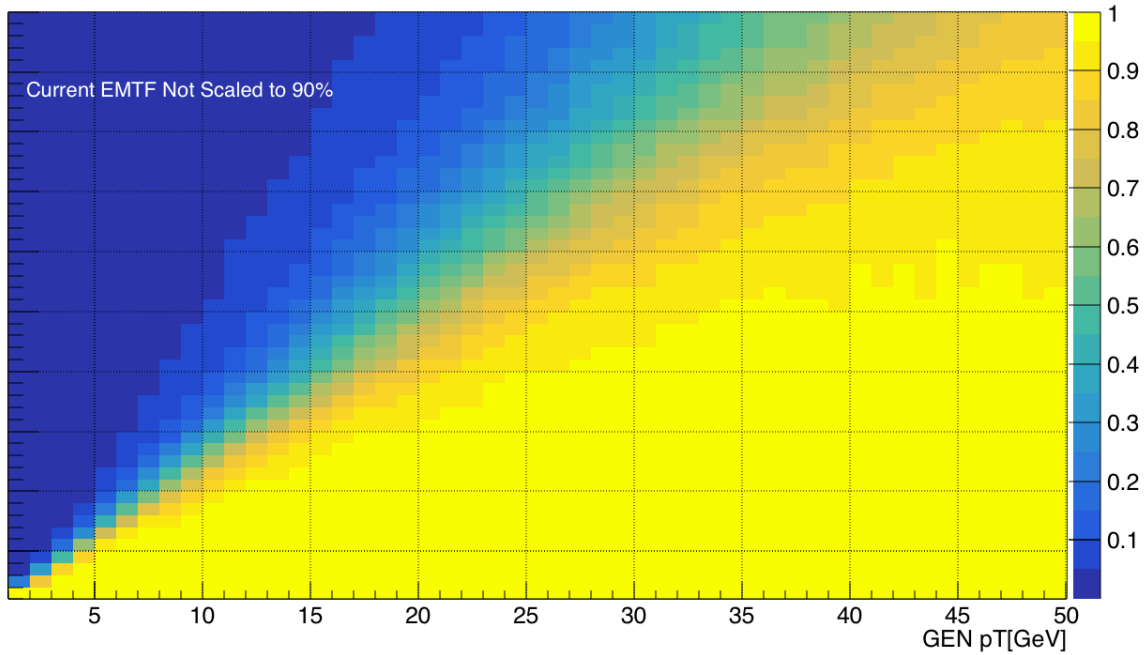
weishi@rice.edu

EMTF Working Meeting

Trigger efficiency(k=75)

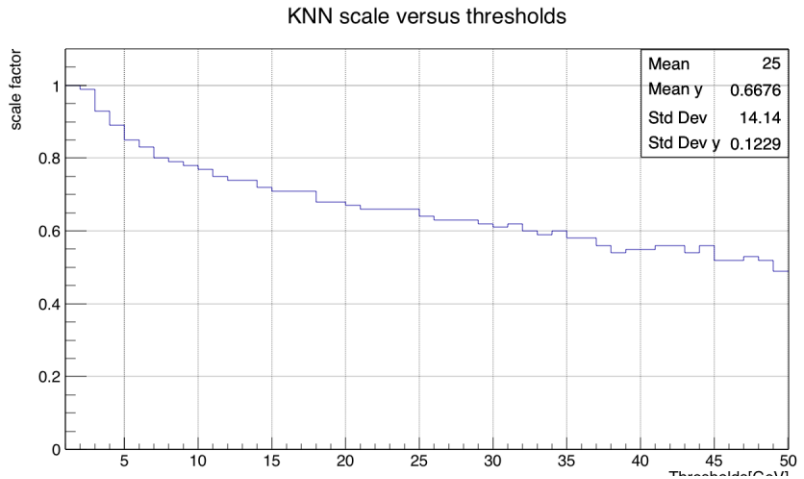


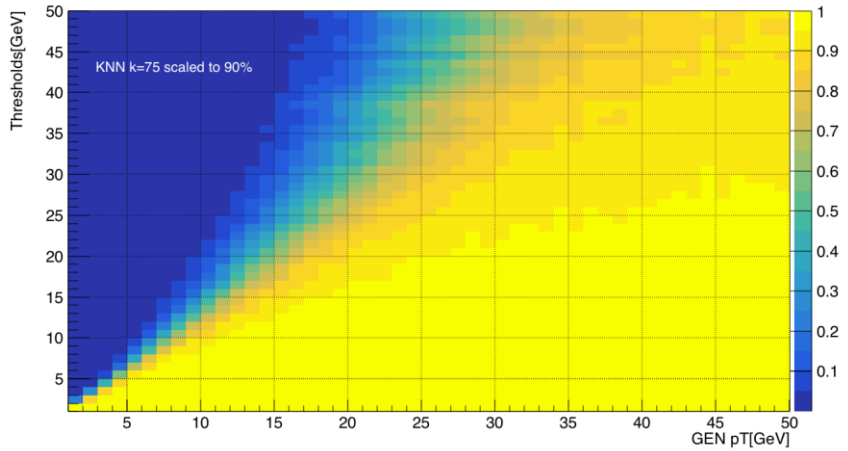
KNN (Not scaled)



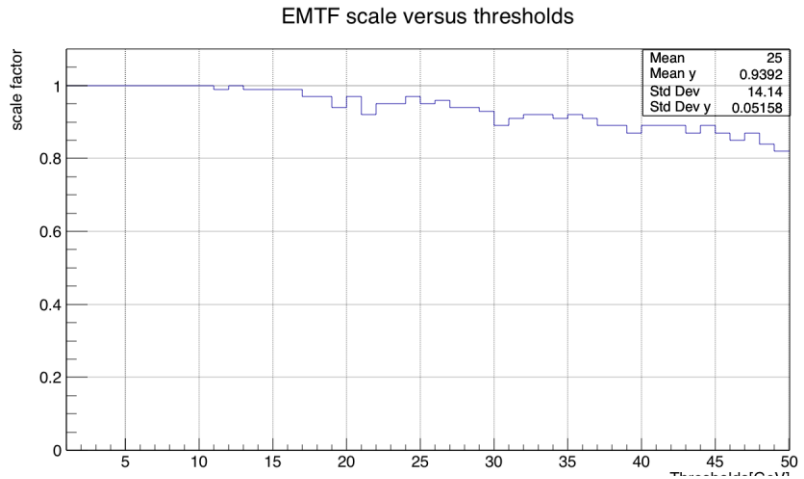
EMTF(Current was scaled to ~80%)

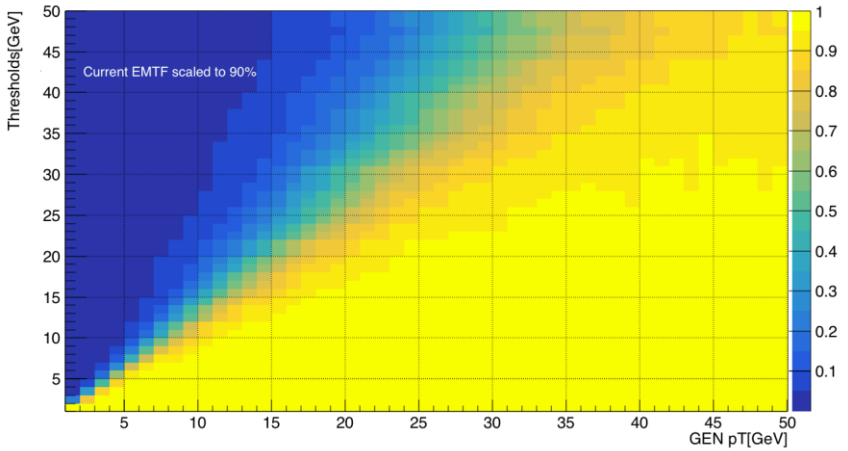
Trigger efficiency(k=75)





KNN (Scaled)

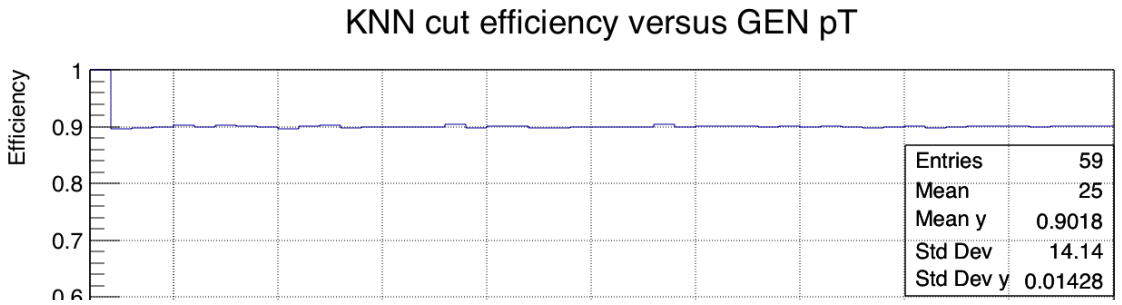


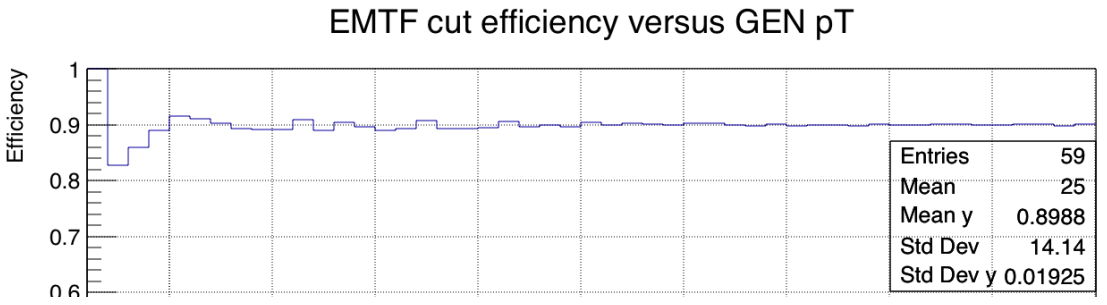


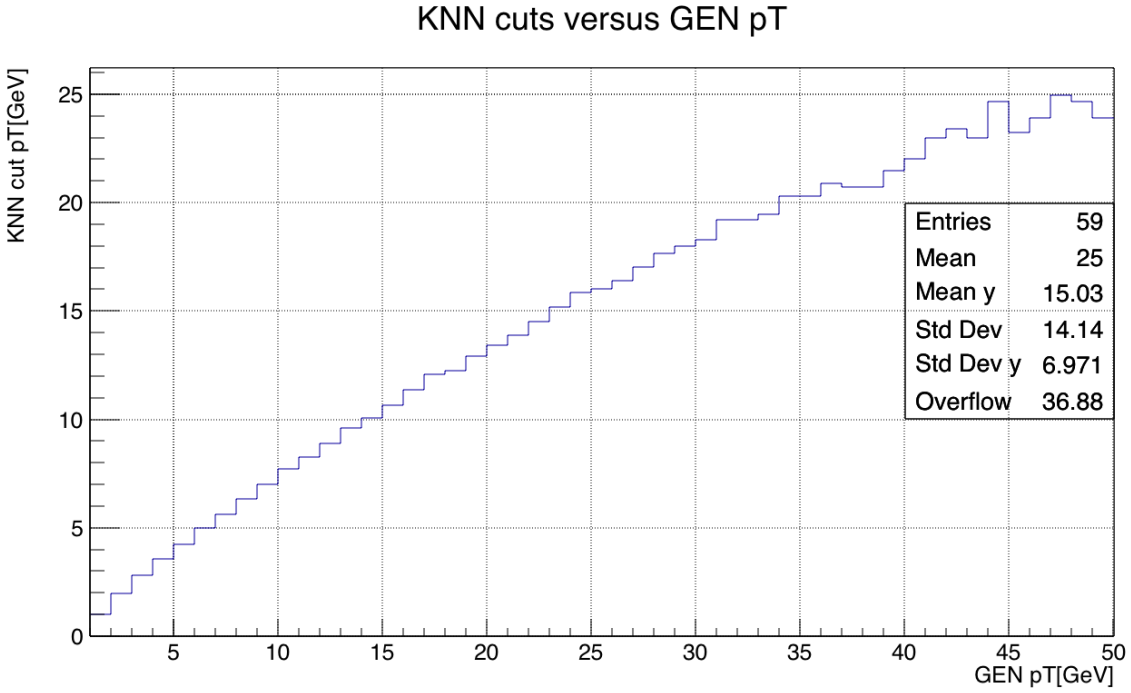
EMTF(Scaled)

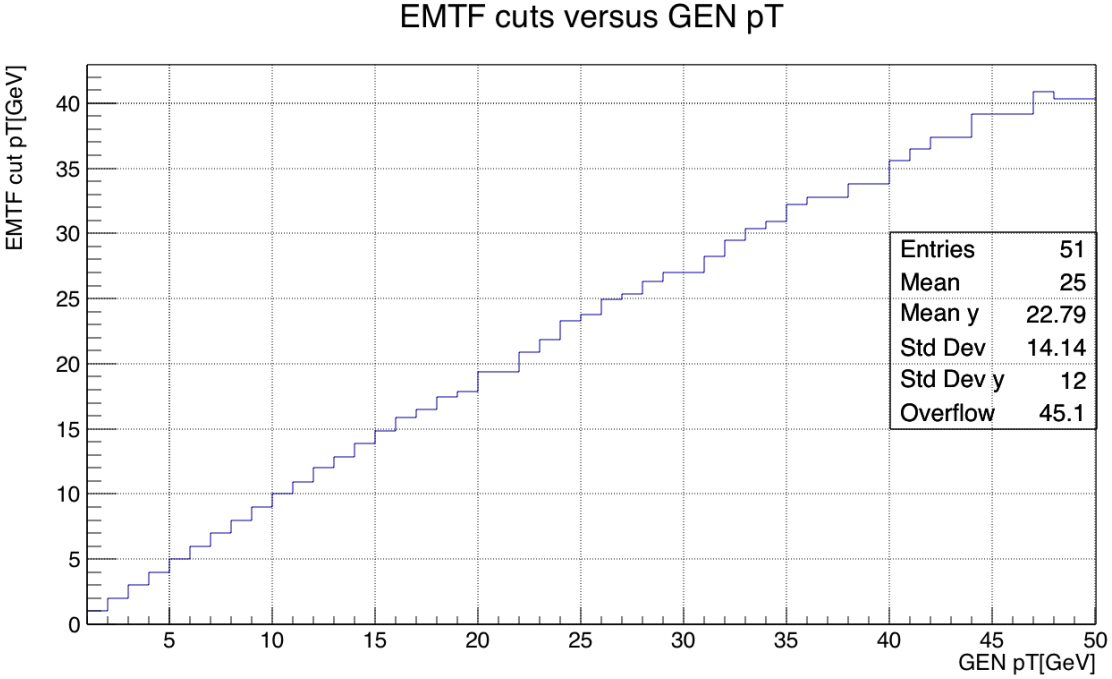
- Scale KNN efficiency to 90% at each threshold by multiplying a factor

Find 90% efficiency cut pT

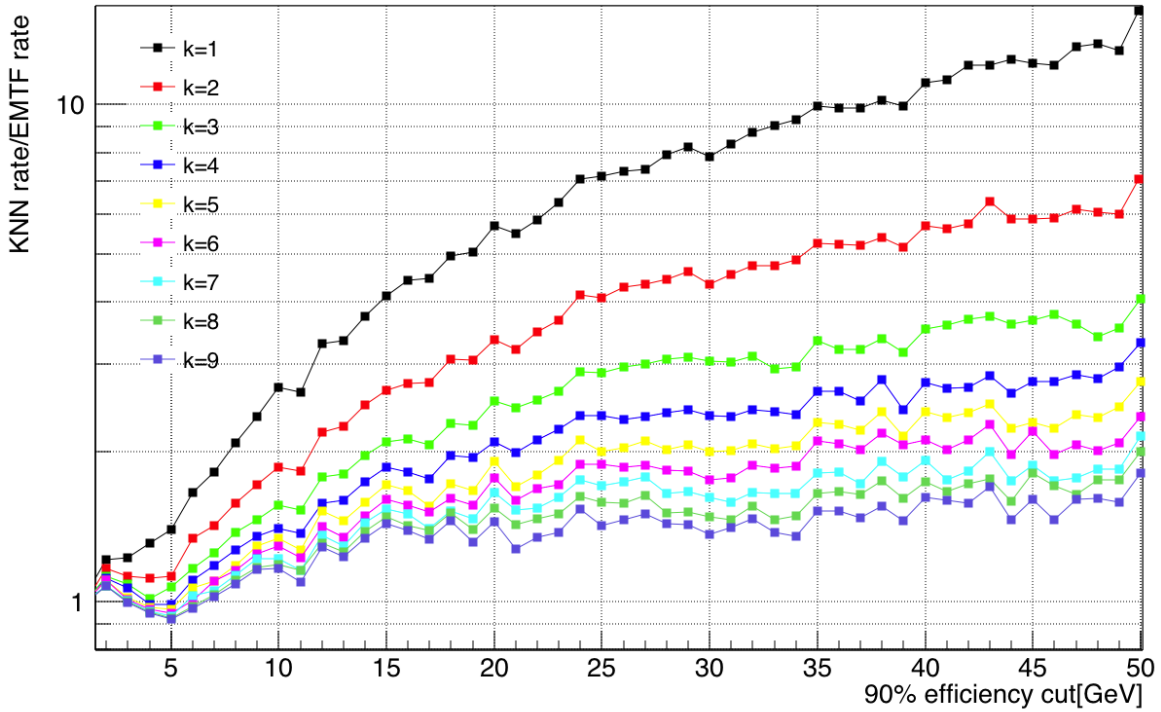


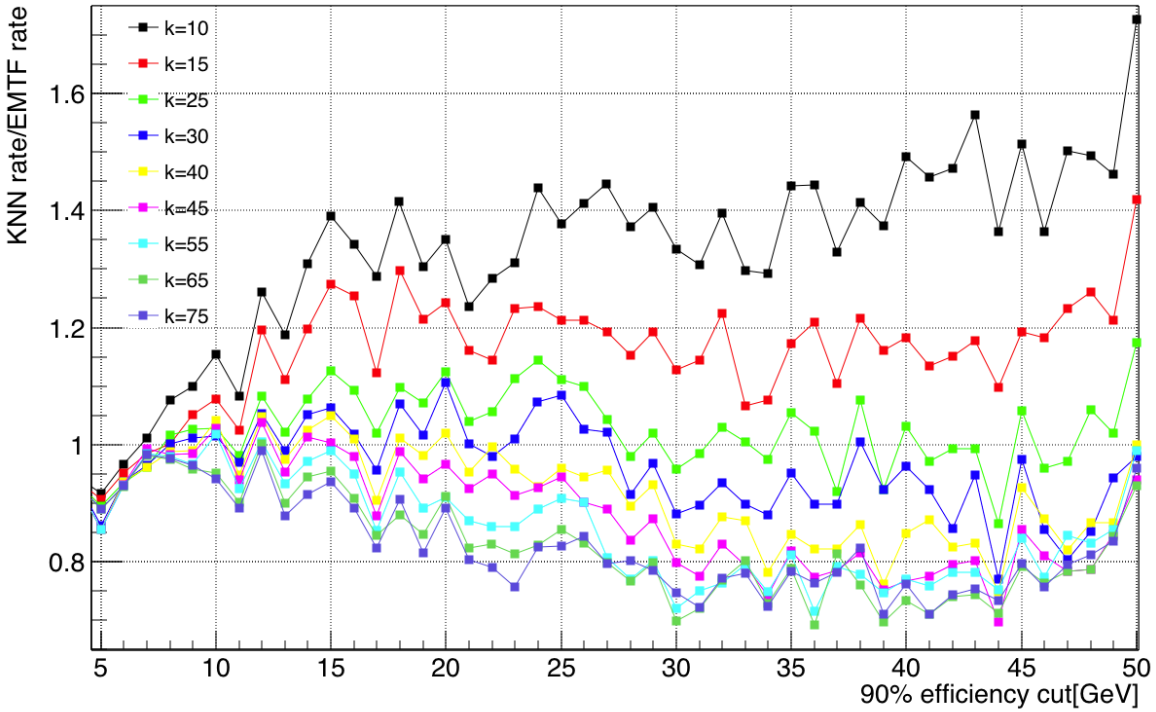






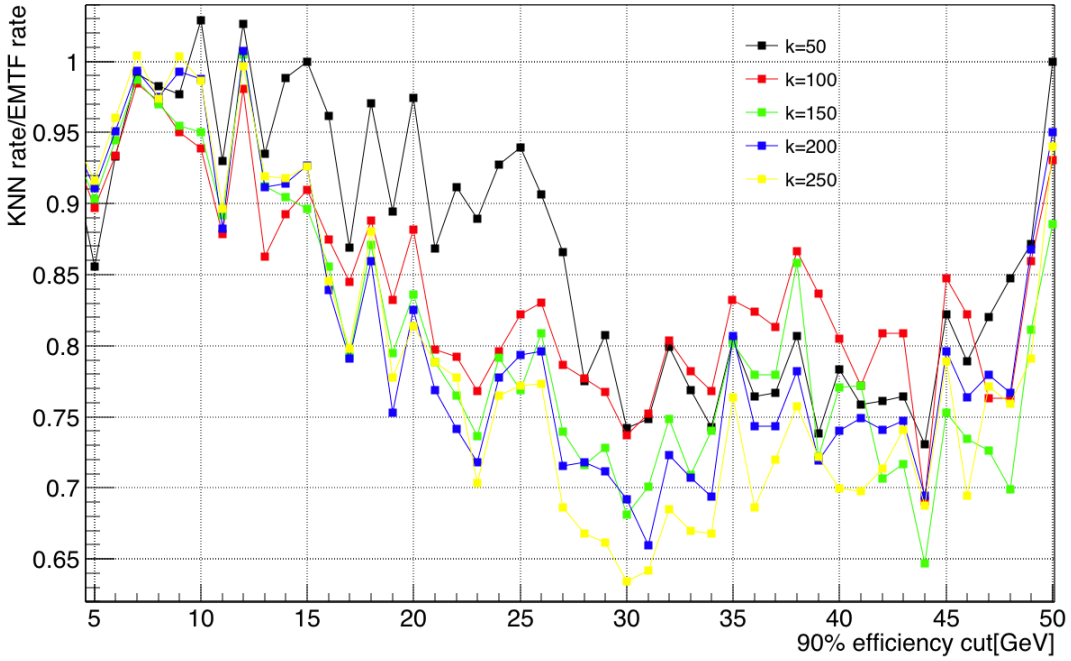
Plot ratio: k-NN rate/EMTF rate

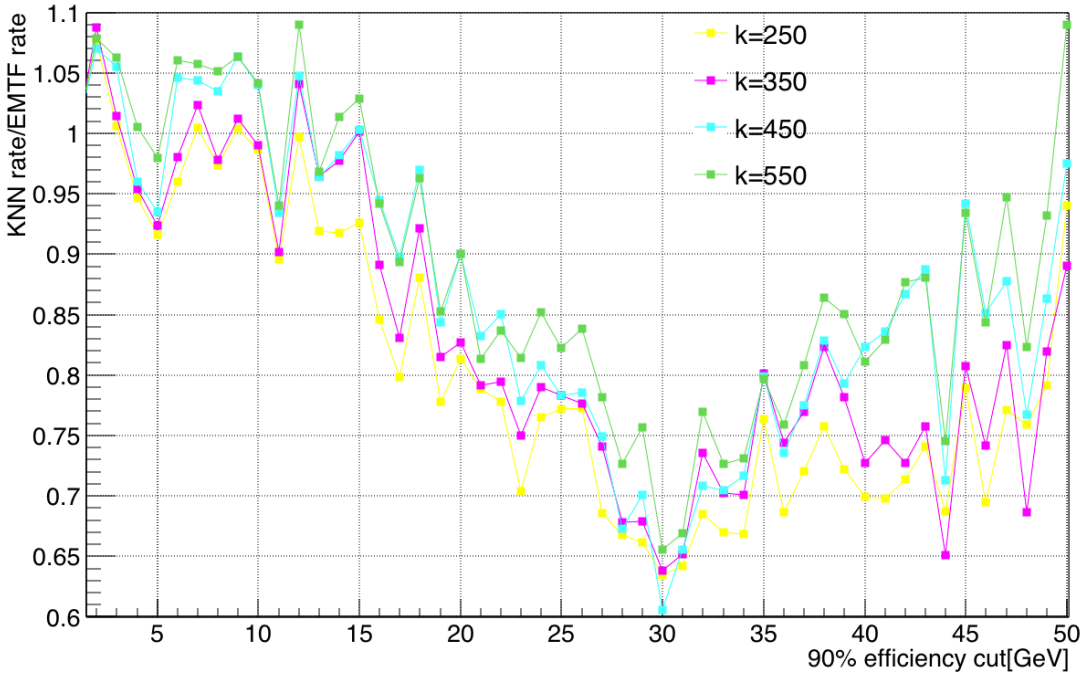




- 90% efficiency cut
- For k in [1, 75], performance goes better as k increases
- Approximate the current EMTF at k~25
- Good rate reduction performance in pT 5-50 GeV for k~75

Large scan step: 50 and 100



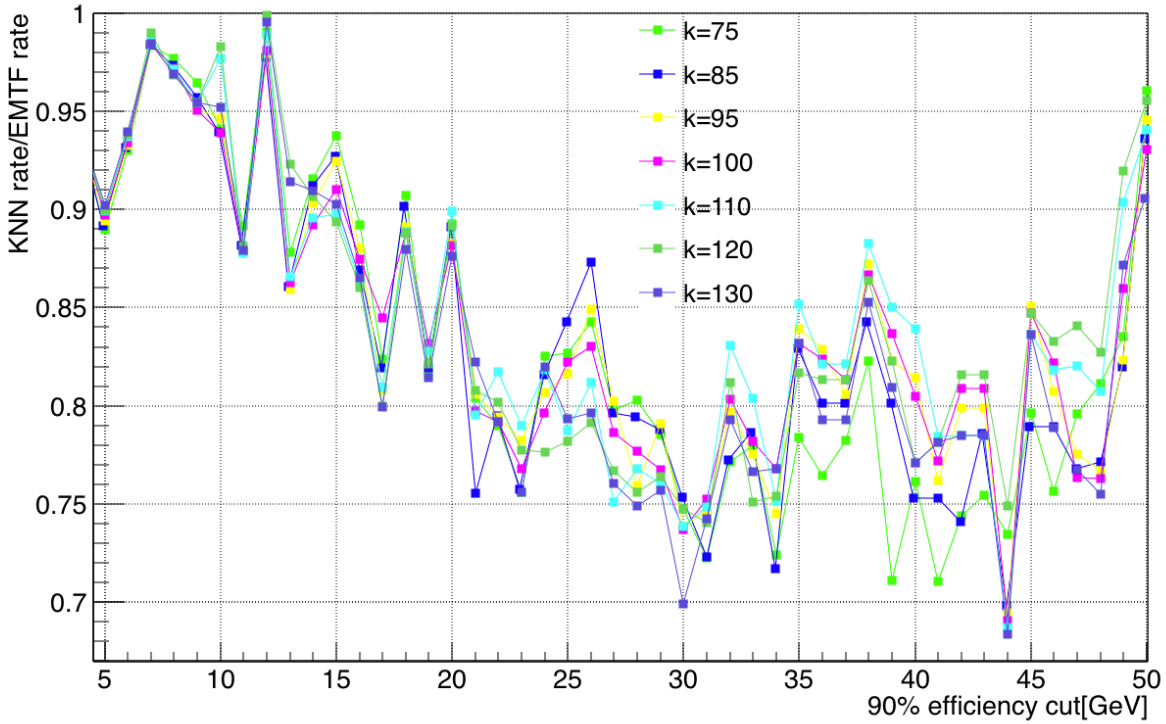


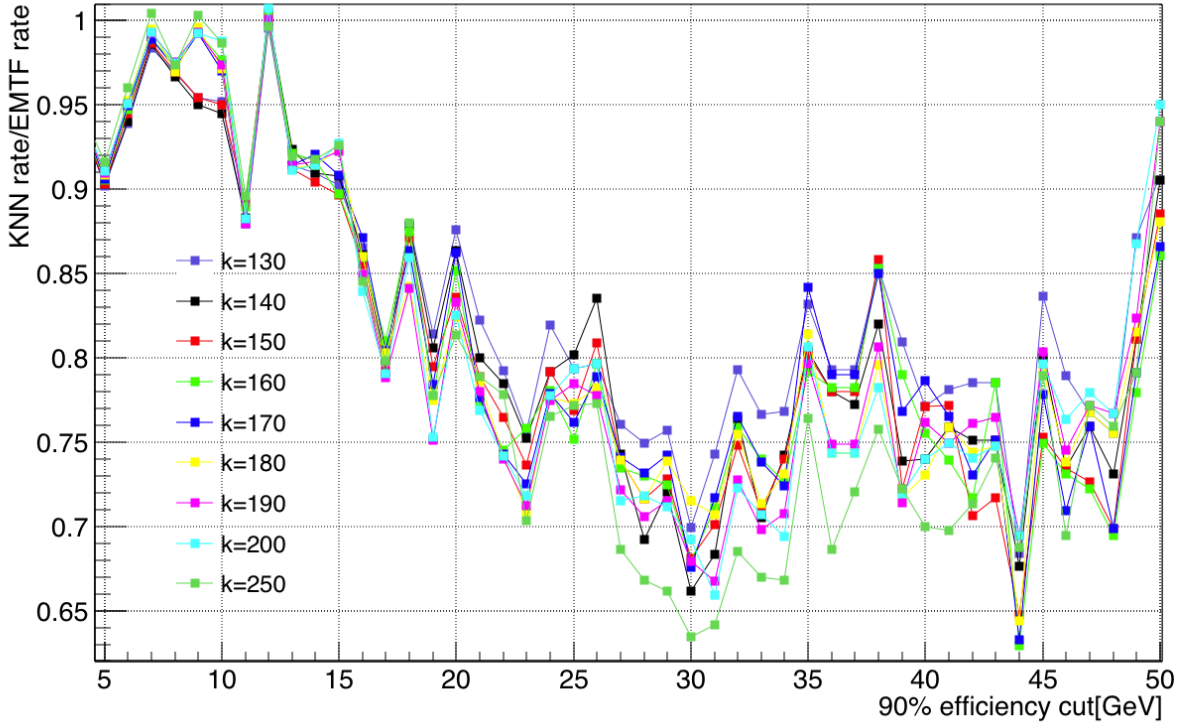
- Wide scan shows good rate reduction keeps until k~250
- Start to degrade the performance as k goes from 250 to 550
- A good stop at k~550, no need to go larger

Very long time for training/testing at k ~ O(10^3), time complexity of k-NN is O(k*logN) in TMVA, not feasible

Very large k is bad for high pT according to previous MSE results (backup)

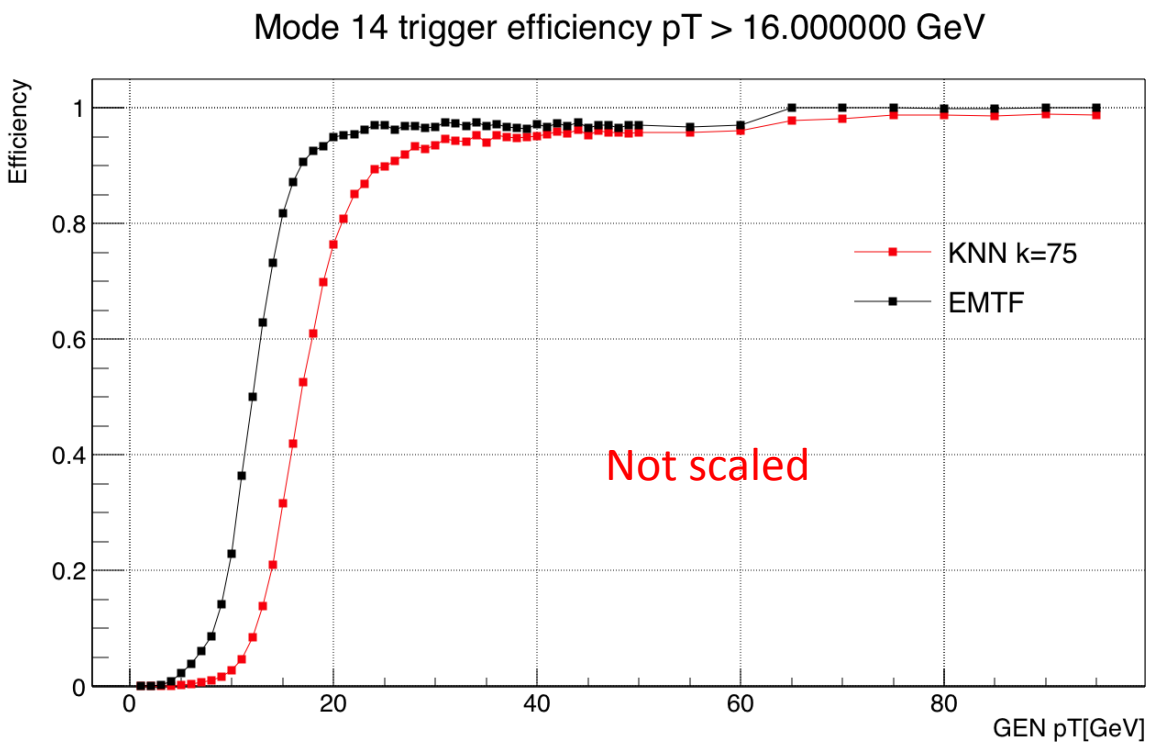
k values between 75 and 200

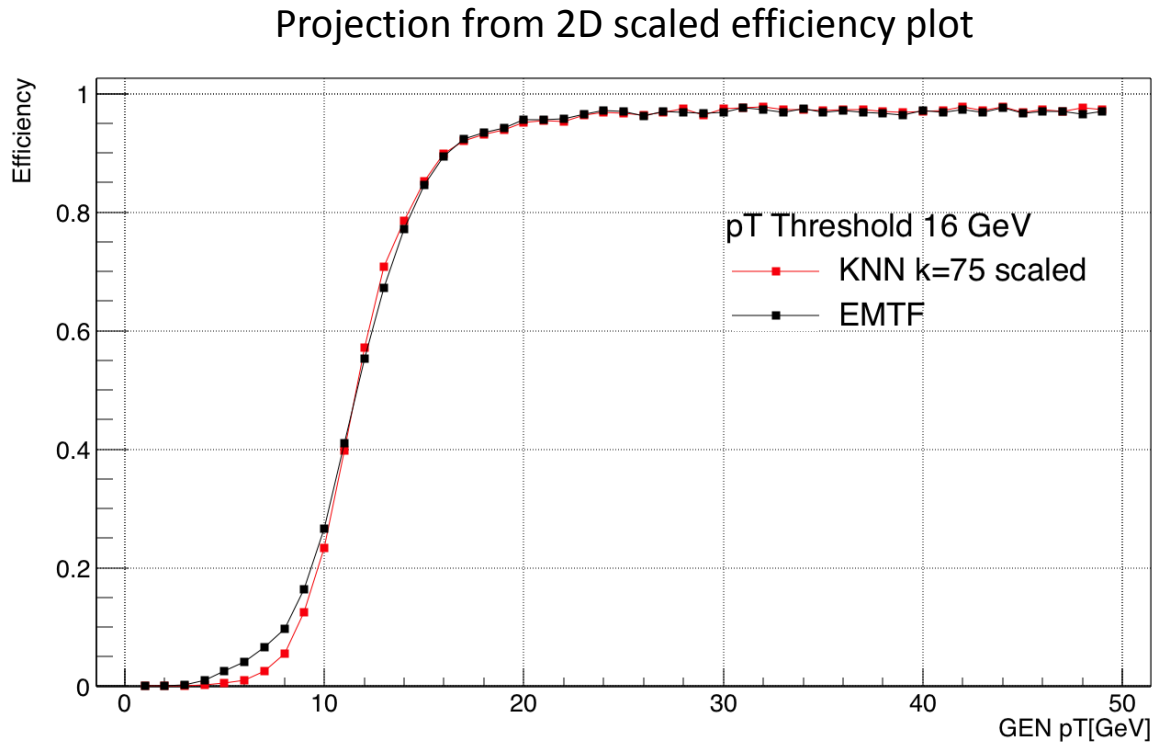




- General trend: performance is better as k goes from 75 to 250
- Depending on what the pT is, the best rate reduction comes from different k
- No significant or overall improvement on pT range 5-50 GeV
- Recommend using k~75, rate reduction 15%-29% more than EMTF at pT 20-40 GeV

Efficiency at k=75, threshold pT 16 GeV





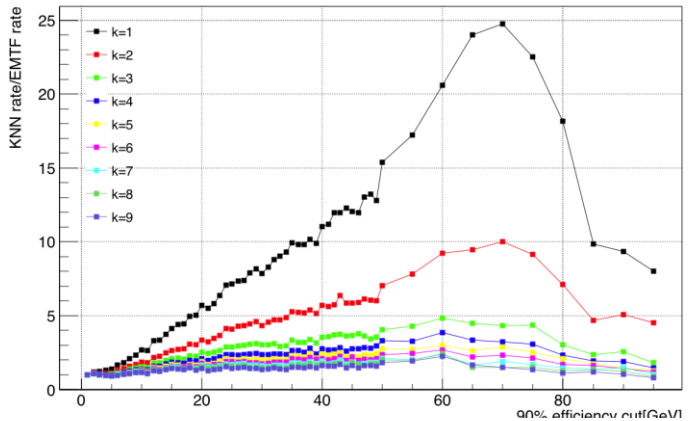
- k=75: 90% efficiency at 16 GeV, larger than 95% for pT >>16 GeV

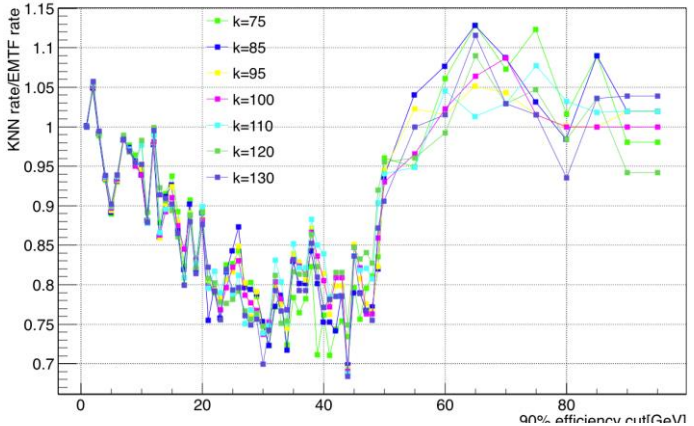
Summary

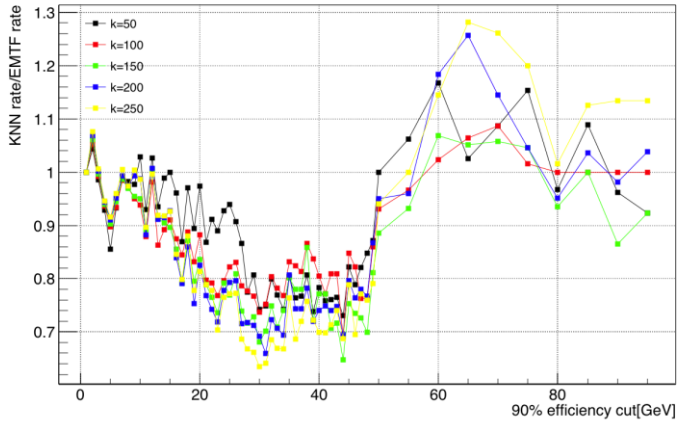
- KNN rate reduction starts to degrade after k~250
- k~75 is recommended for a good rate reduction for pT at 5-50 GeV
- k=75: 90% trigger efficiency at threshold 16 GeV, over 95% when pT >> 16 GeV

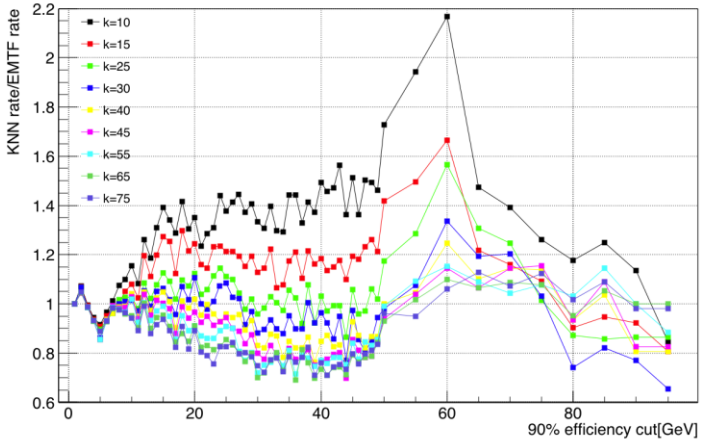
BACK UP

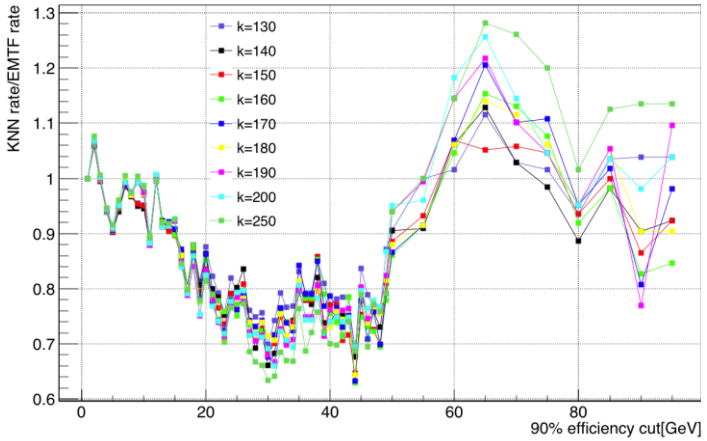
Rate tatio vs k, pT 1-99GeV

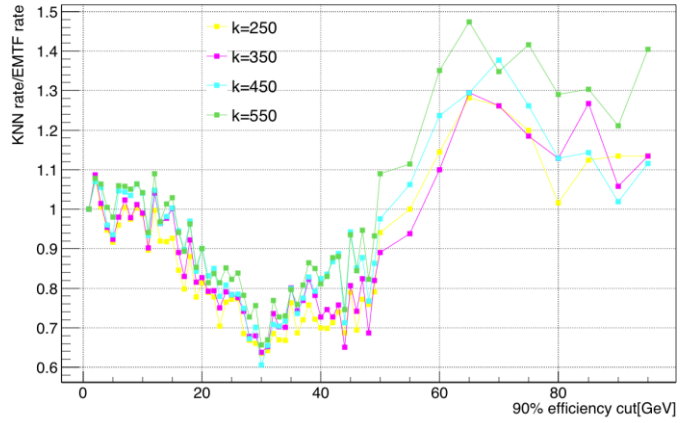




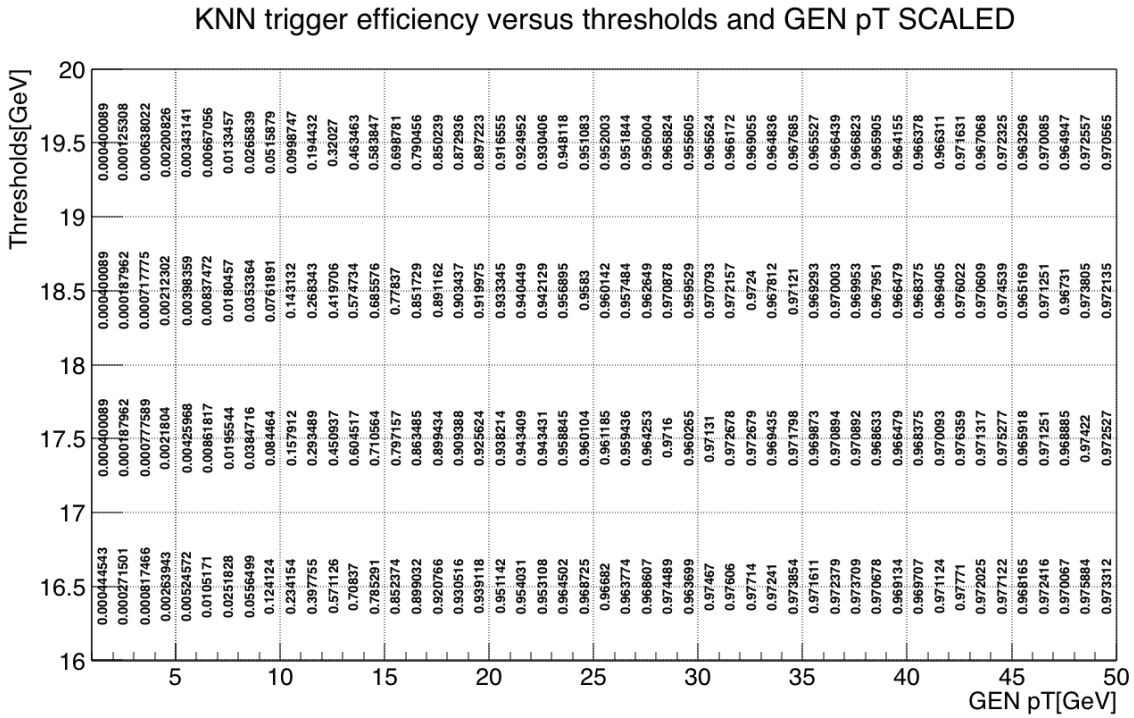


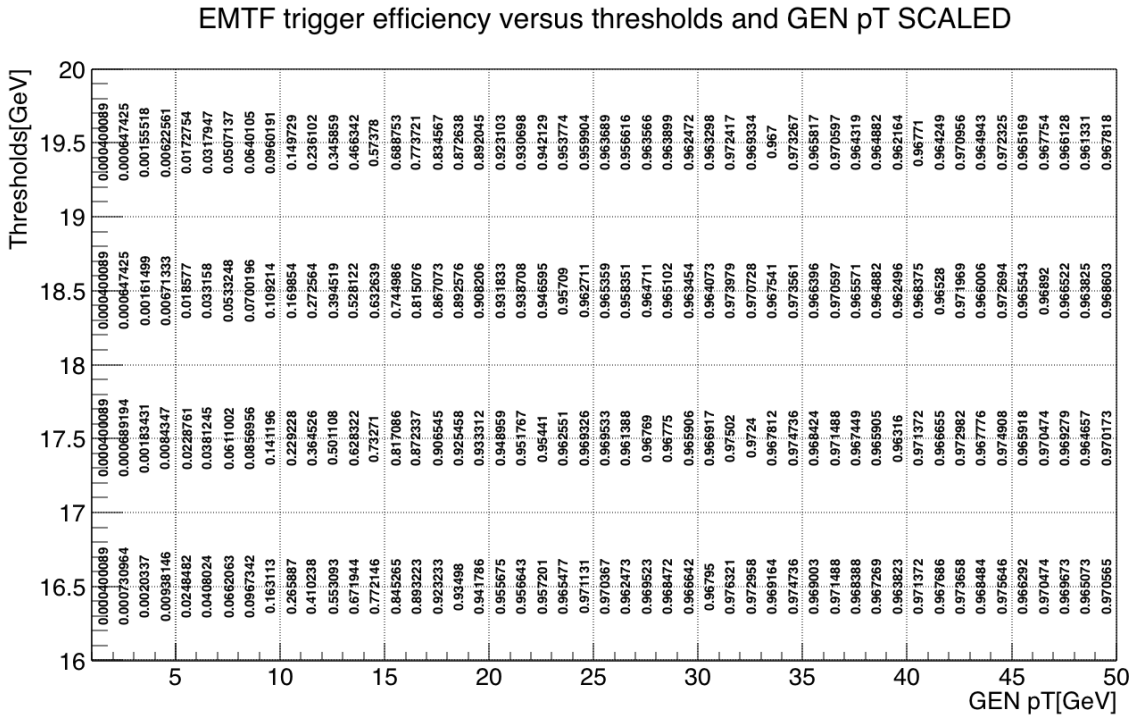






Efficiency at k=75, threshold pT 16-20 GeV





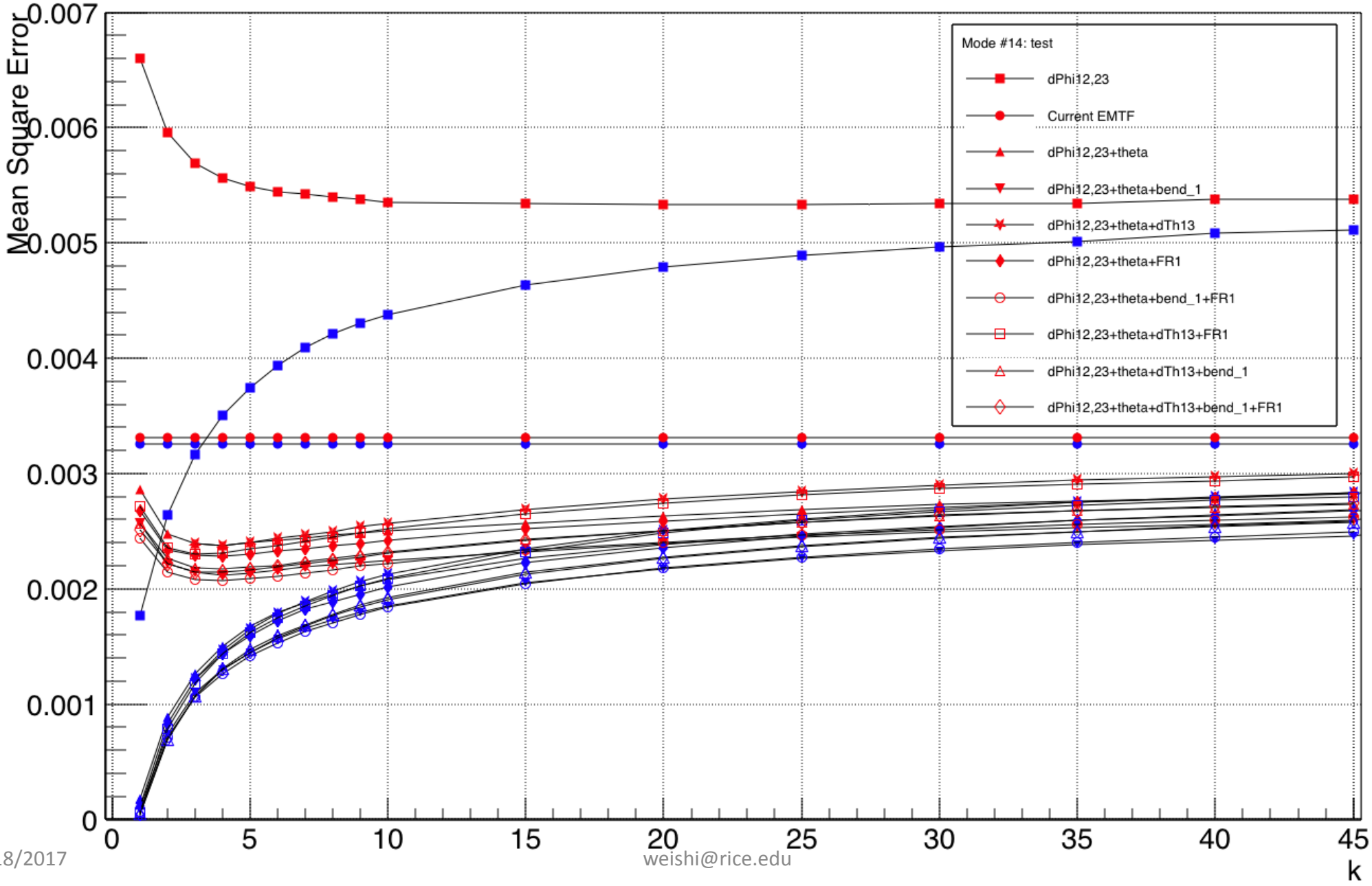
- 2016 EMTF scaled to 85% using a universal constant for all modes
- All possibilities of input variables are looped(this is done offline) and put into xml file to get a LUT(~2GB), and LUT is put into firmware, online assignment only use LUT, don't do calculation online
- K-NN might affect emulator in CMSSW, which use xml instead of LUT
- Include RPC, truncate bits →changes
- Compare different trigger algorithms(BDT, MLP, ANN) with k-NN using the 2017 LUT variables scheme for same mode 14(add FR2, ring # in station 1, # can be 1 or 2)
- Study all 3-station modes(mode 11 in EMTF may have bug)
- GEN→EMTF Emulator in CMSSW(pT resolution~25%)→RECO(pT resolution 1%, can be either MC or real data)~GEN
- Packing/unpacking: full precision dphi(10 bits needed for every single dphi)→nonlinear dphi, small dphi mapping is one to one, larger dphi start to reduce precisions/low resolution

Mode 14: Station 1-2-3

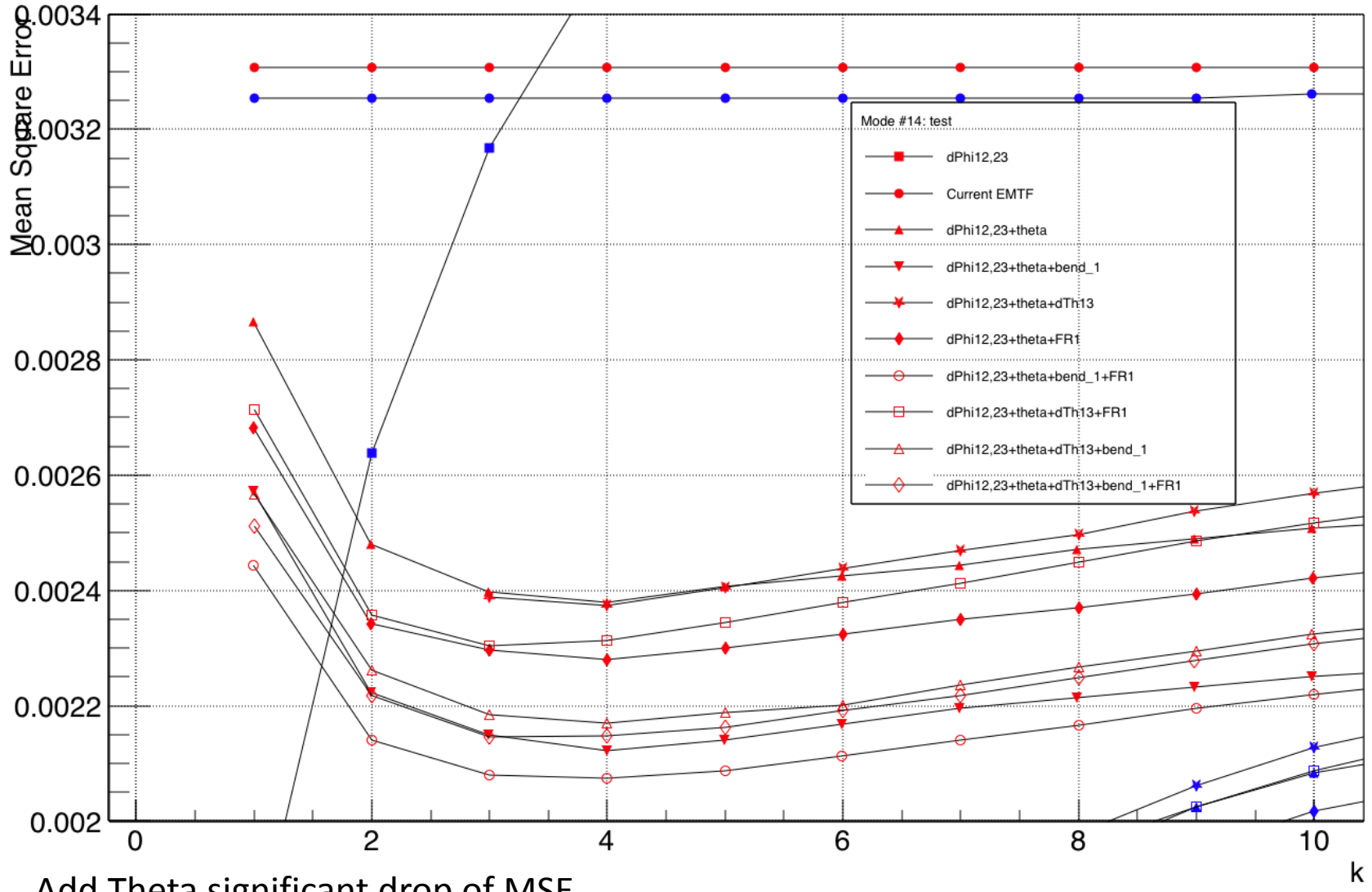
Compare to last time:

- Much higher statistics(10M events, mode 14 has 787858 train and 788544 test), 42 * last time: 18746 train)
- 2016 LUT(dPhi12, 23; theta; CLCT1; dTheta13; FR 1)(mainly tune value k)
- Low/high pT divide up
- Weight 1/pT(unweight not plotted yet)
- Tune KNN scale frac
- Doing classification instead of regression?

Whole pT range, weighted 1/pT

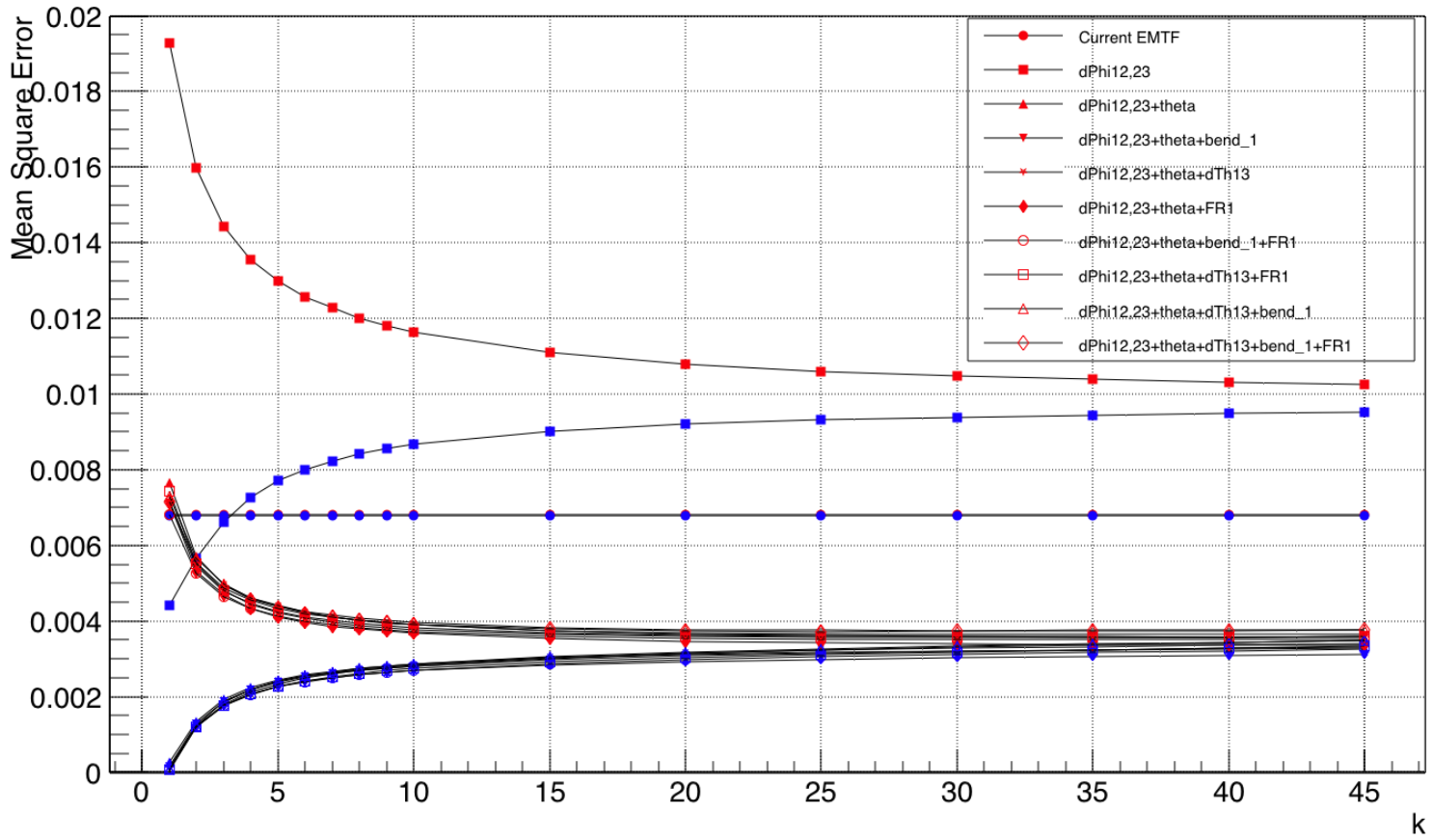


Zoom in



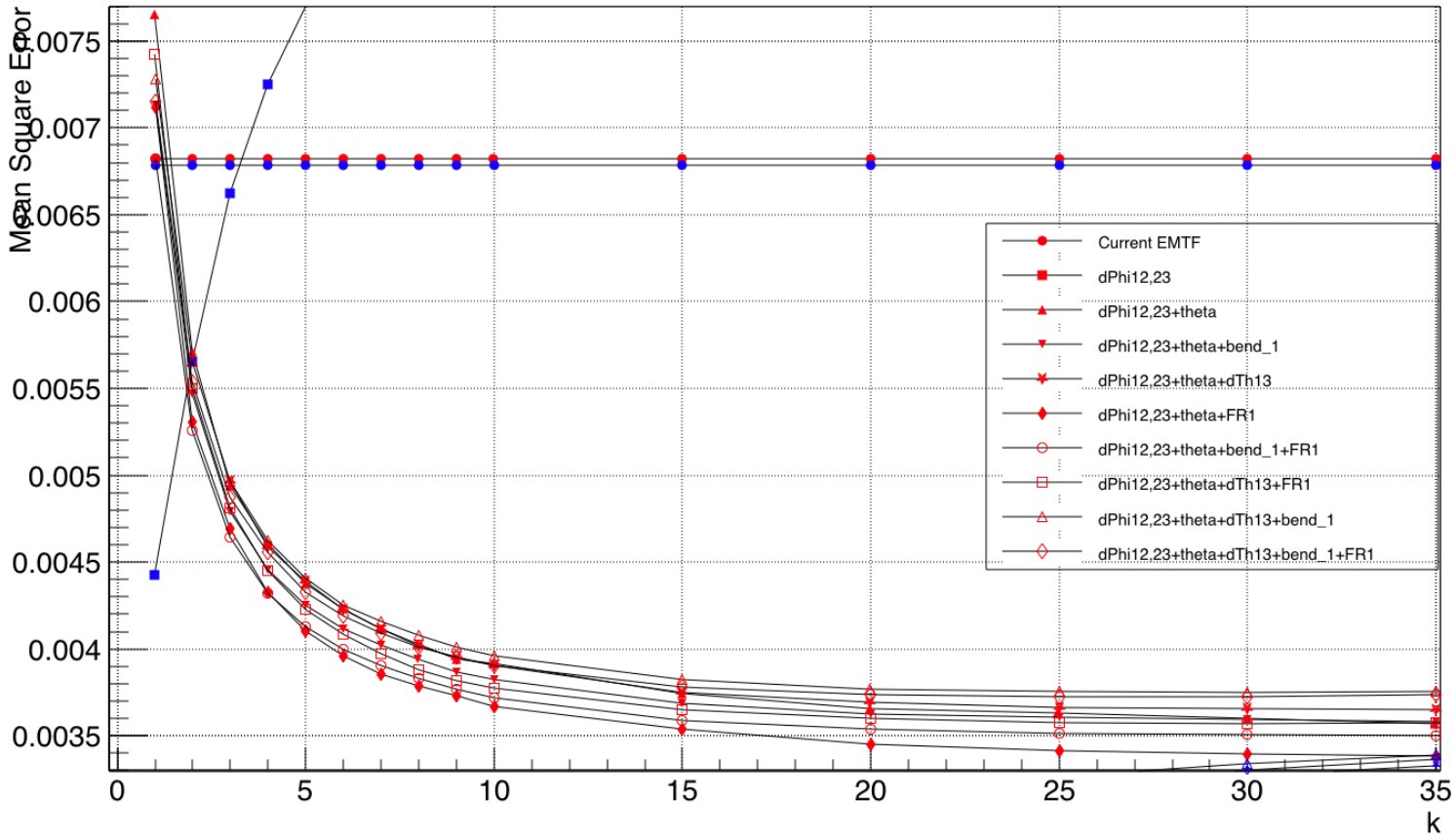
- Add Theta significant drop of MSE
- dPhis+theta+bend_1+FR1 give best performance

pT 1-8GeV

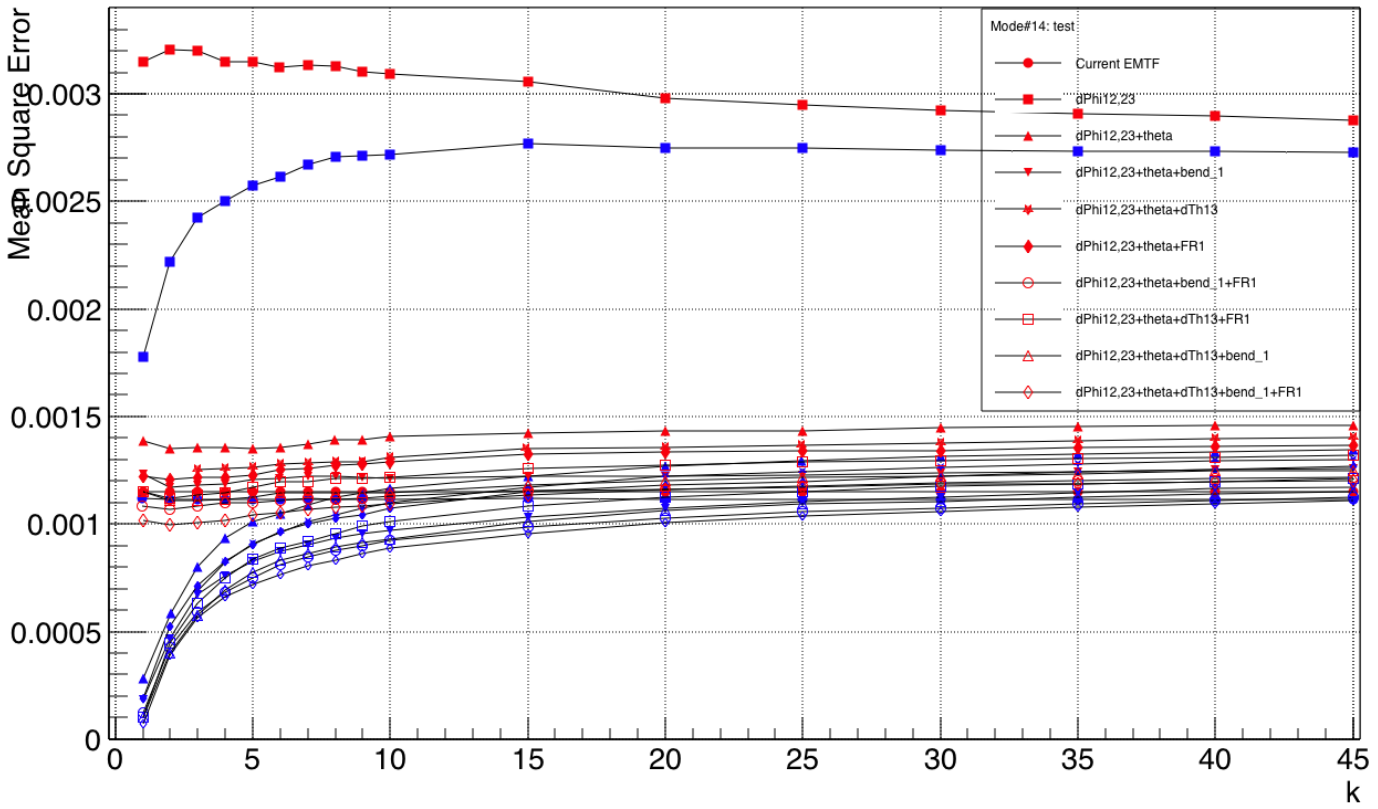


- Over fit starts when k<10, main contribution to overtrain in whole pT range plot
- Large k performance unchanged, still not enough statistics?

1-8GeV zoom in

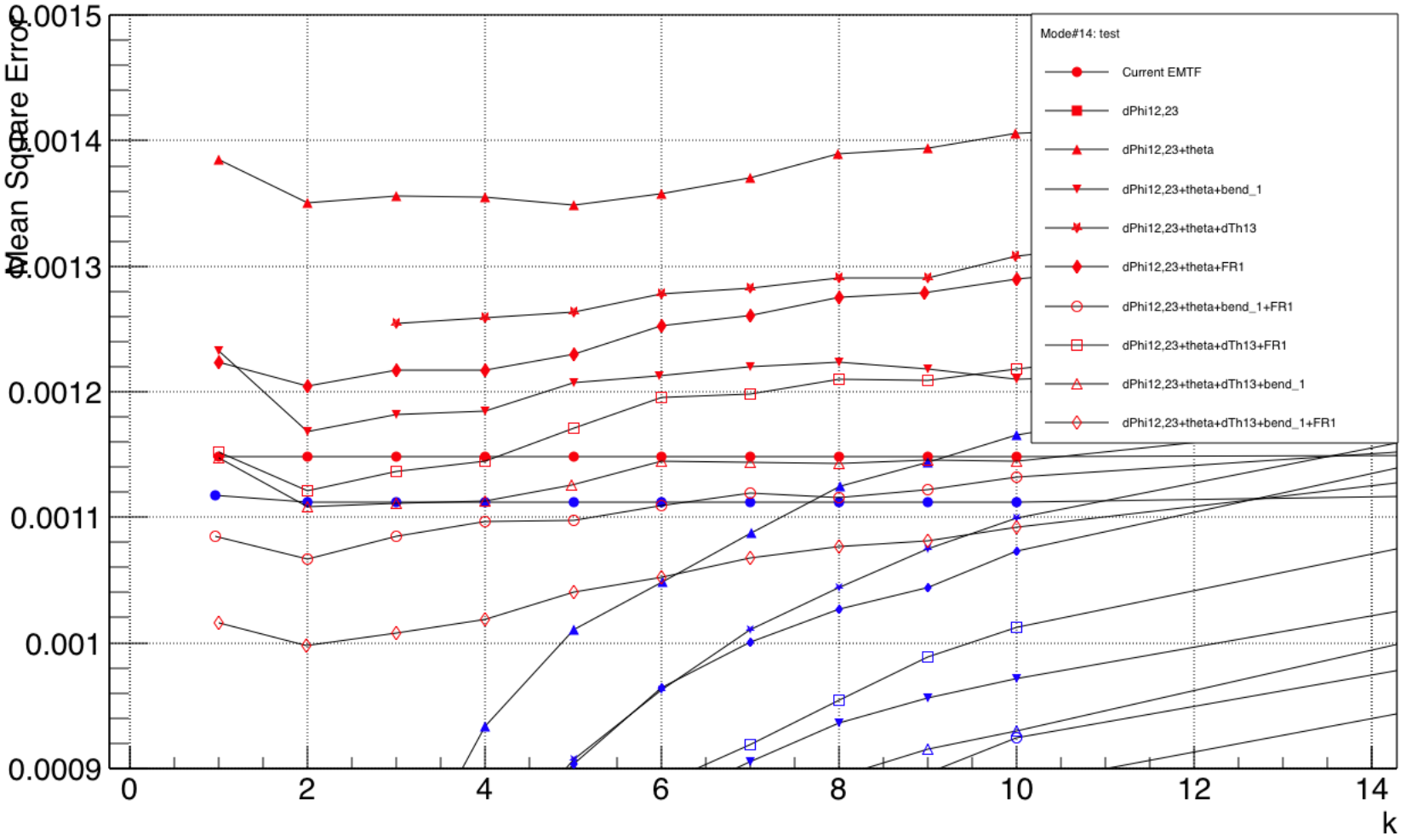


8-30GeV

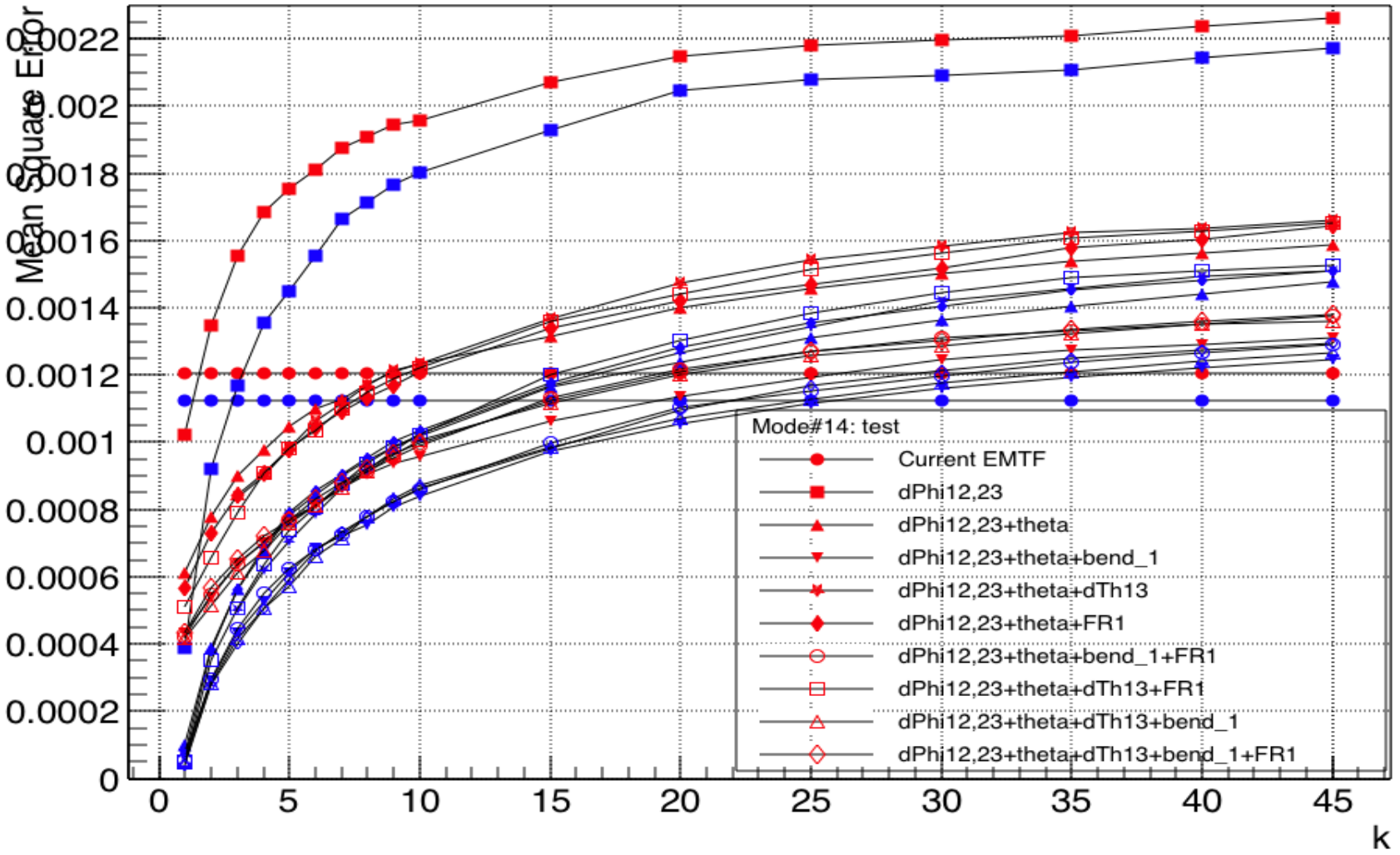


- Minima at k~2

8-30GeV zoom in

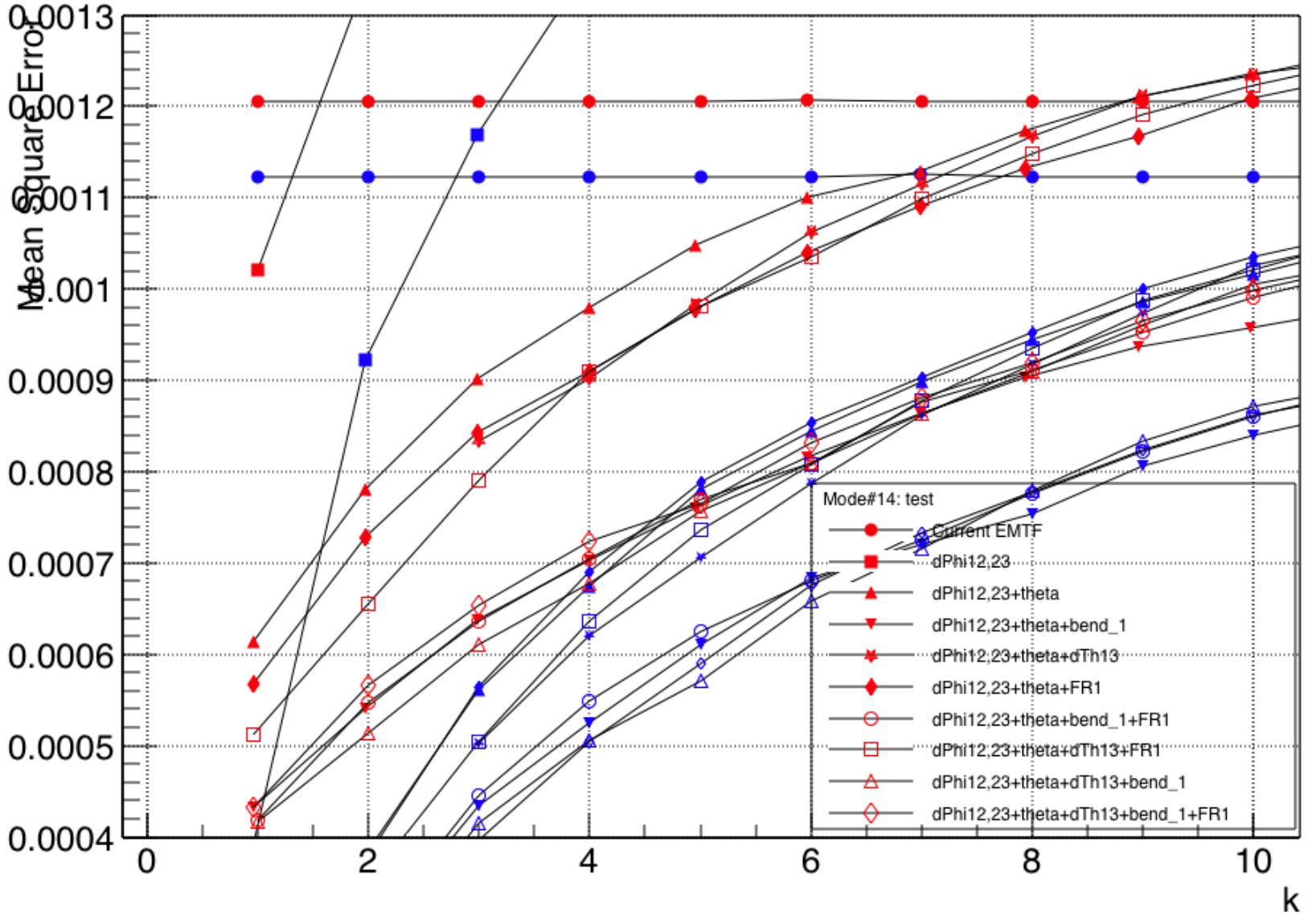


30-120GeV

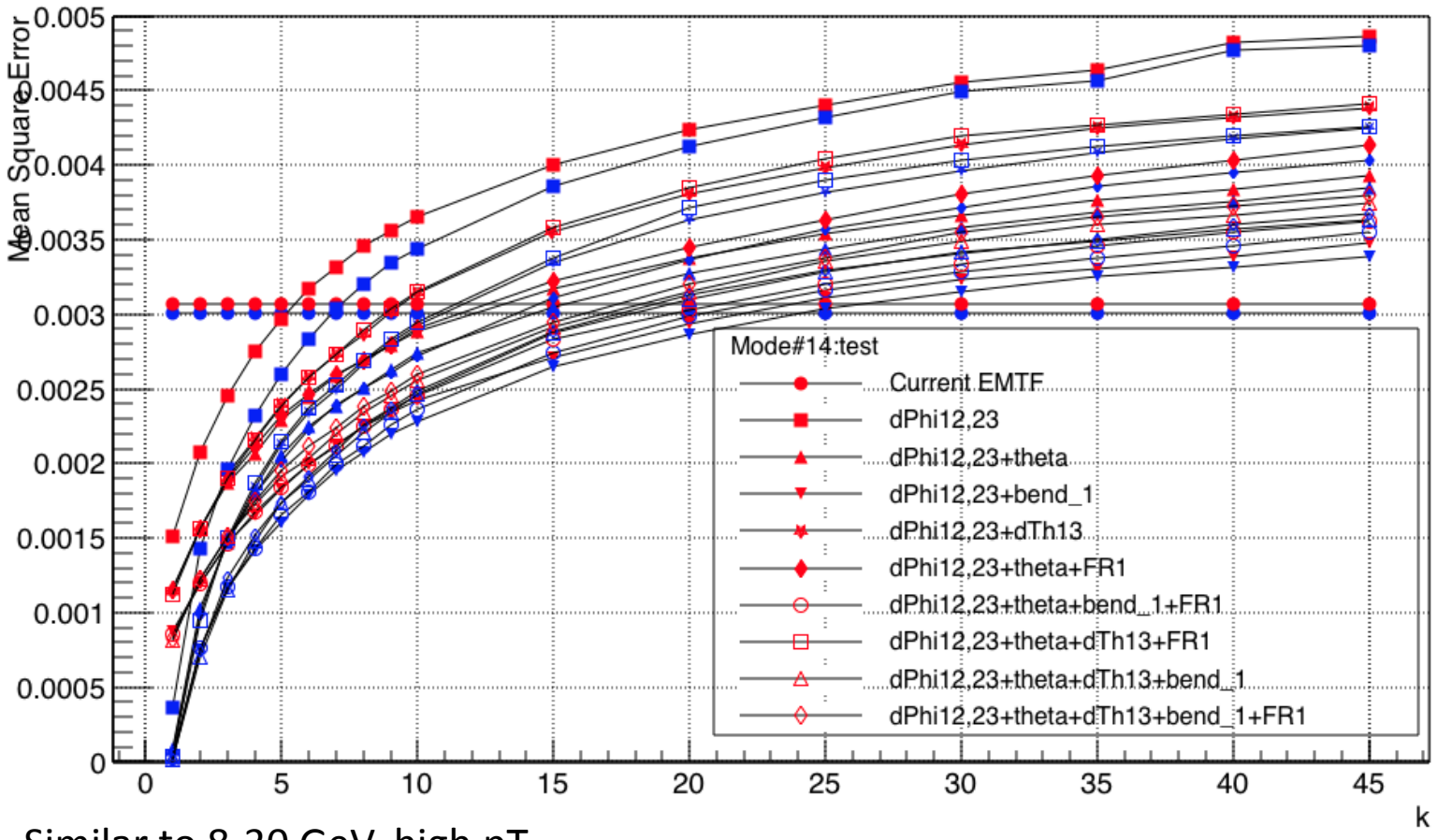


- MSE keeps decreasing even when k=1, too sparse input variable space?

30-120GeV zoom in

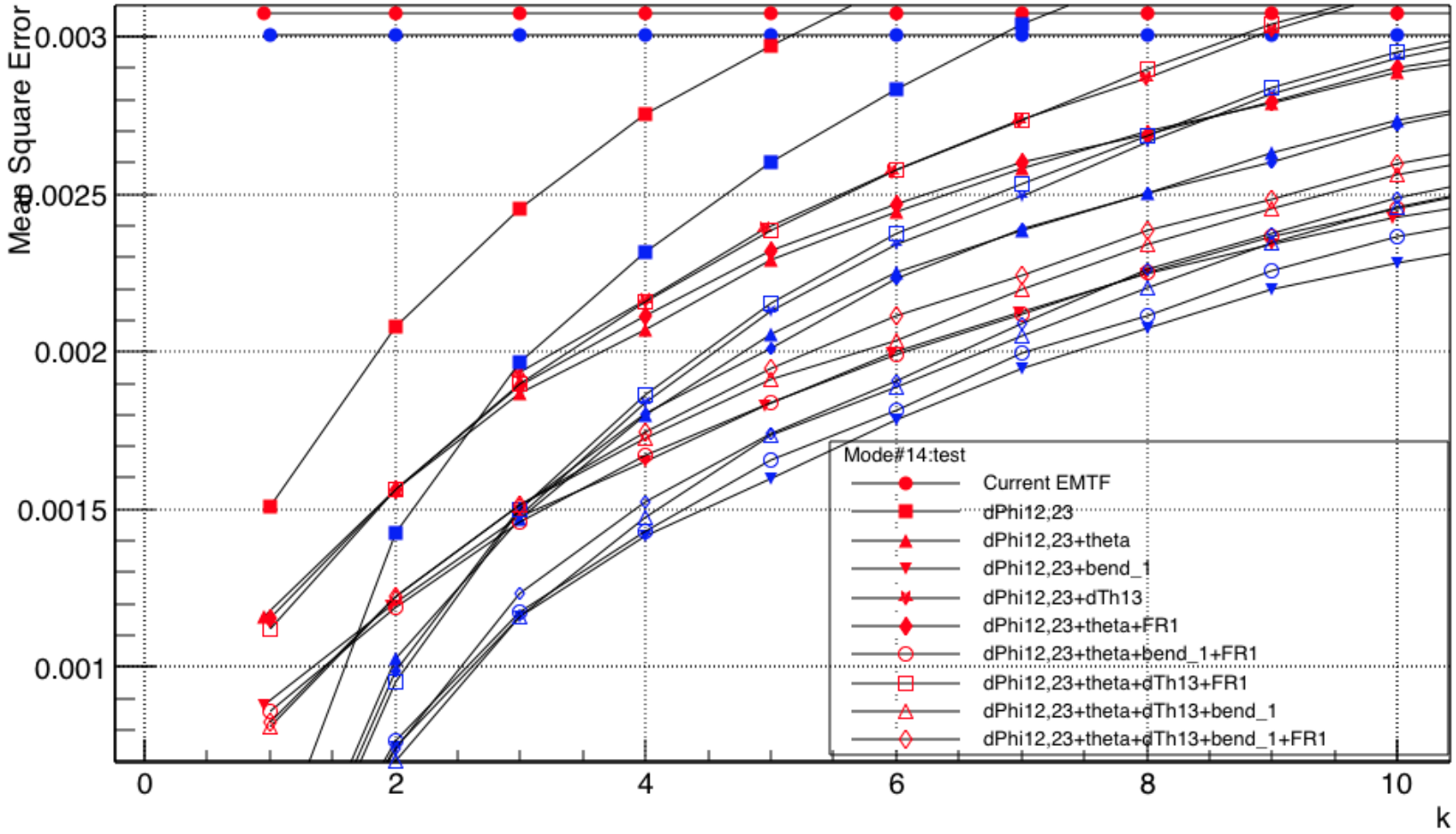


120-1000GeV



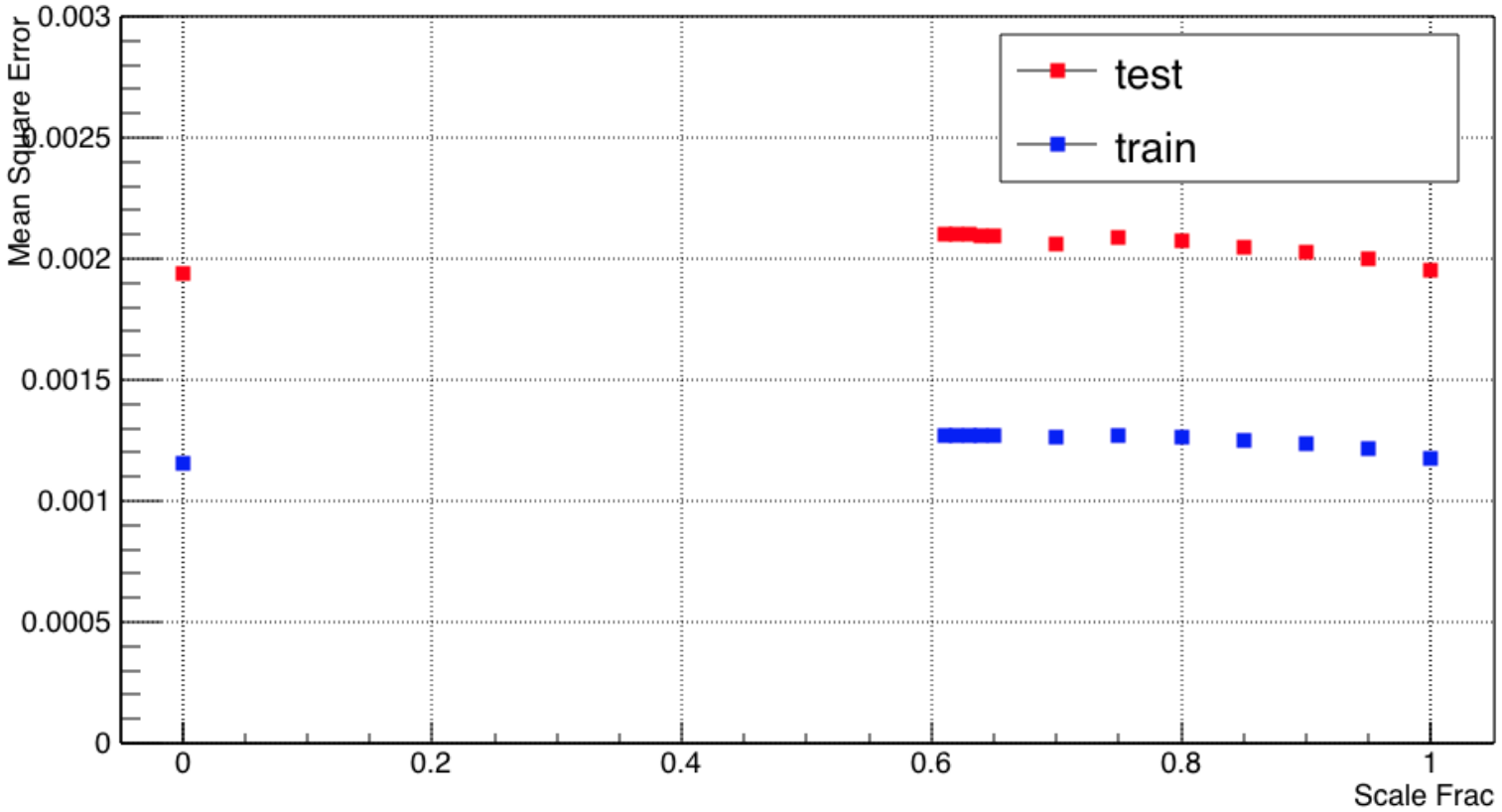
- Similar to 8-30 GeV, high pT

120-1000GeV


Mean Square Error
0.003
0.0025
0.002
0.0015
0.001
0
2
4
6
8
10
k
Mode#14:test
Current EMTF
dPhi12,23
dPhi12,23+theta
dPhi12,23+bend_1
dPhi12,23+dTh13
dPhi12,23+theta+FR1
dPhi12,23+theta+bend_1+FR1
dPhi12,23+theta+dTh13+FR1
dPhi12,23+theta+dTh13+bend_1
dPhi12,23+theta+dTh13+bend_1+FR1

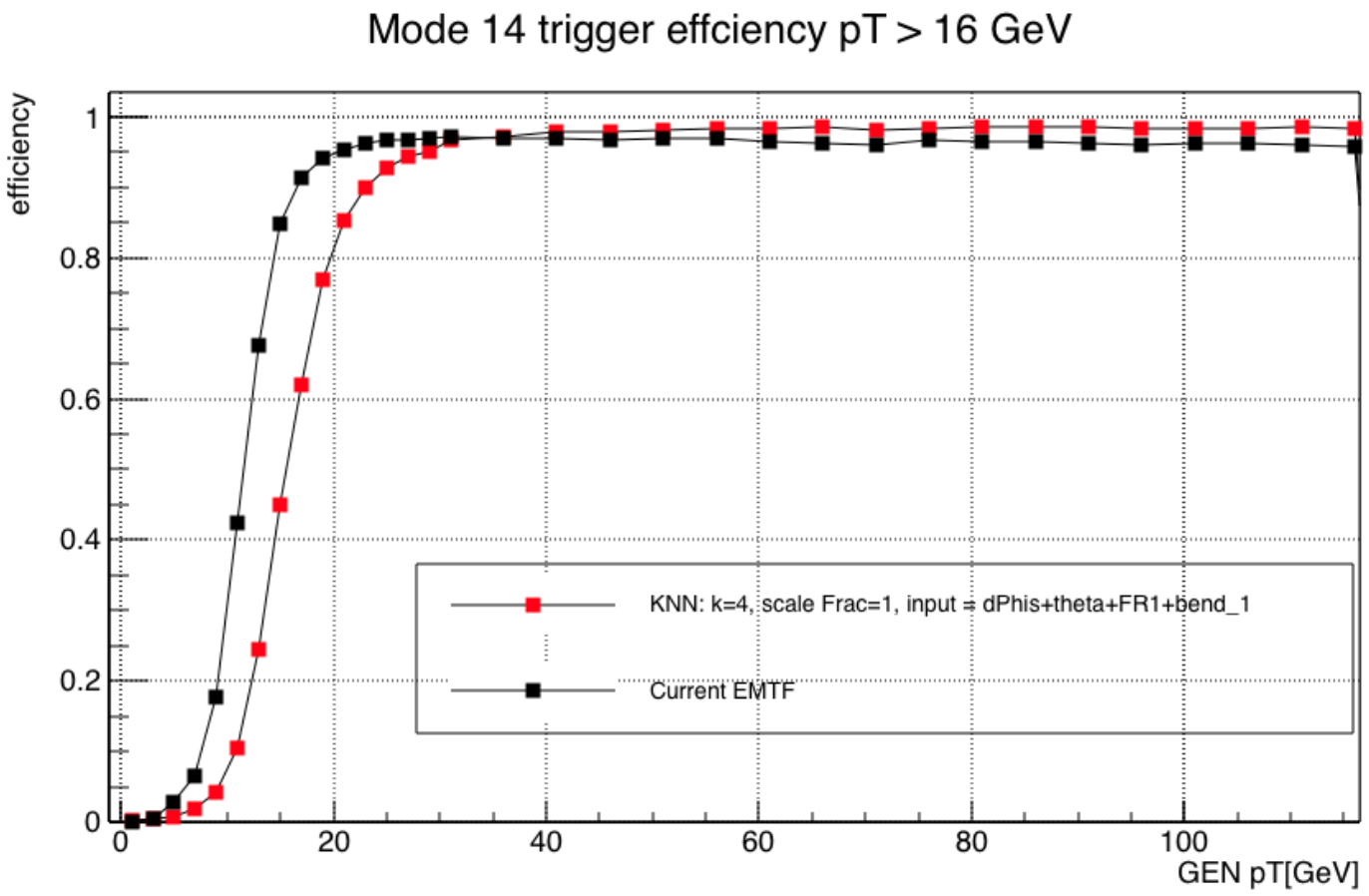
Scale the variables

- Input variables have different distribution
- e.g. theta has larger distribution than dphis
- Need to standardize variables when calculating distance



- Scale Frac = 0 means turning off scale
- Scale Frac = 1 gives best performance but no big difference

Efficiency on test sample(Not on zerobias!)



Conclusions

- Significant drop when adding theta
- Almost for all sets of input variables, k=4, Scale Frac=1, gives best performace
- dPhi(12,23),theta,bend_1,FR1 are best input variables
- Overfit at small k contributed from low pT 1-30GeV
- high momentum seems need more data(weight 1/pT)
- Flat curve at large k indicates statistics still not high enough(?)
- Non-parametric method/depend on local data structure/no mapping
- Large dataset/cross validation dataset

Steps

- Find optimized input/target/k/scalfrac for KNN for other modes
- Select the better method used for other modes: change inputs
- Include RPC hits and repeat 1-3 step
- Remove/truncate to fit in 29 bits
- Repeat for other modes
- Train charge assignment(classification)

Modes

Mode #	Definiition in code	Stations
15	1+2+4+8	1,2,3,4
14	2+4+8	1,2,3
13	1+4+8	1,2,4
12	4+8	1,2
11	1+2+8	1,3,4
10	2+8	1,3
9	1+8	1,4
8	8	1
7	1+2+4	2,3,4
6	2+4	2,3
5	1+4	2,4
4	4	2
3	1+2	3,4
2	2	3
1	1	4

Track mode	Δφ							Δθ						Bits
	1-2	1-3	1-4	2-3	2-4	3-4	+/-	1-2	1-3	1-4	2-3	2-4	3-4	
15	7			5		6	2**							20
14	7			5			2		3					17
13	7				5		2			3				17
12	9						1	3						13
11		7				5	2			3				17
10		9					1		3					13
9			9				1			3				13
7				7		6	2					3		18
6				9			1				3			13
5					9		1					3		13
3						9	1						3	13

Track mode	Bend (CLCT)					FR				θ	Md	Σ	Σ	Bits
	1	2	3	4	+/-	1	2	3	4					
15						1				5	4	10	20	30
14	2				1	1				5	4	13	17	30
13	2				1	1				5	4	13	17	30
12	2	2			2	1	1			5	4	17	13	30
11	2				1	1				5	4	13	17	30
10	2		2		2	1	1			5	4	17	13	30
9	2			2	2	1			1	5	4	17	13	30
7		2			1					5	4	12	18	30
6		2	2		2		1	1		5	4	17	13	30
5		2		2	2		1		1	5	4	17	13	30
3			2	2	2			1	1	5	4	17	13	30

Euclidean distance

x: input variables(dimension d) of train event
y: input variables of test event
f: polynomial kernel weight function
w_j: weight of j in train sample

$$R_{\text{rescaled}} = \left(\sum_{i=1}^{d} \frac{1}{w_i^2} |x_i - y_i|^2 \right)^{\frac{1}{2}}$$

$$\langle t(i,V) \rangle = \frac{\sum_{j \in V} w_j t_j f(\text{dis}(i,j))}{\sum_{j \in V} w_j f(\text{dis}(i,j))}$$

kNN

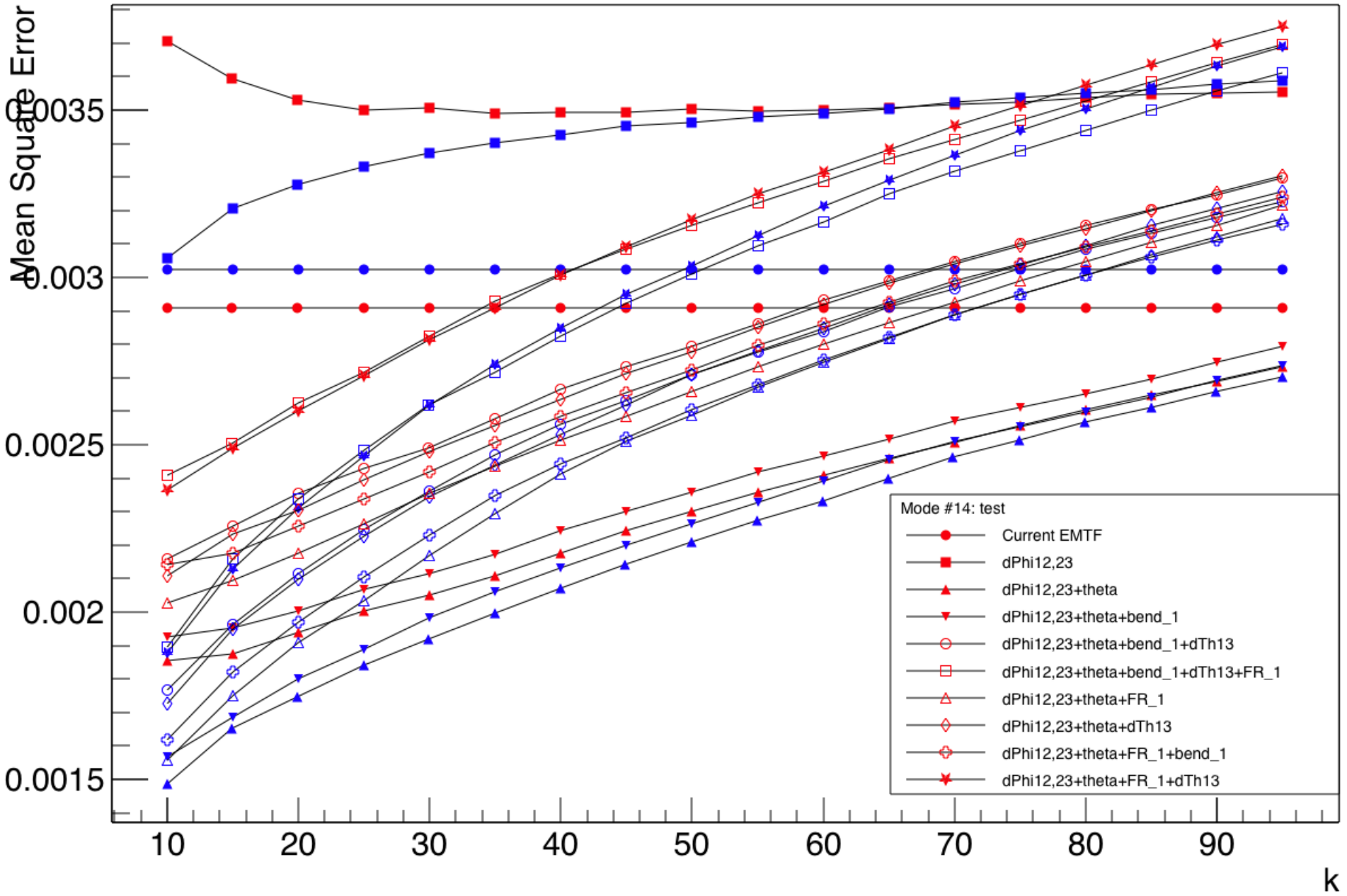
- Tune k and Scale frac
- Continuous/categorical variable: different metric(Euclidean/Hamming)
- When mixture of both kinds of variables, standardization or scale variables(called feature normalization)
- E.g. x'=(x-min)/(max-min); x'= (x-x_mean)/x_variance; assign weights to d(i,j)(TMVA adopts this)
- A non-parametric method
  Unlike other supervised learning algorithms, K-Nearest Neighbors doesn't learn an explicit mapping f from the training data
- Simply uses the training data at the test time to make predictions
- Need large dataset/cross validation dataset
- use kd-tree to store train data, O(log(n)) time/ one search

KNN Regression

- For a test event, the algorithm finds the k-nearest neighbours using the input variables, where each training event contains a regression value. The predicted regression value for the test event is the weighted average of the regression values of the k-nearest neighbours
- The choice of the metric governs the performance of the nearest neighbour algorithm. When input variables have different units a variable that has a wider distribution contributes with a greater weight to the Euclidean metric. This feature is compensated by rescaling the variables using a scaling fraction determined by the option ScaleFrac.

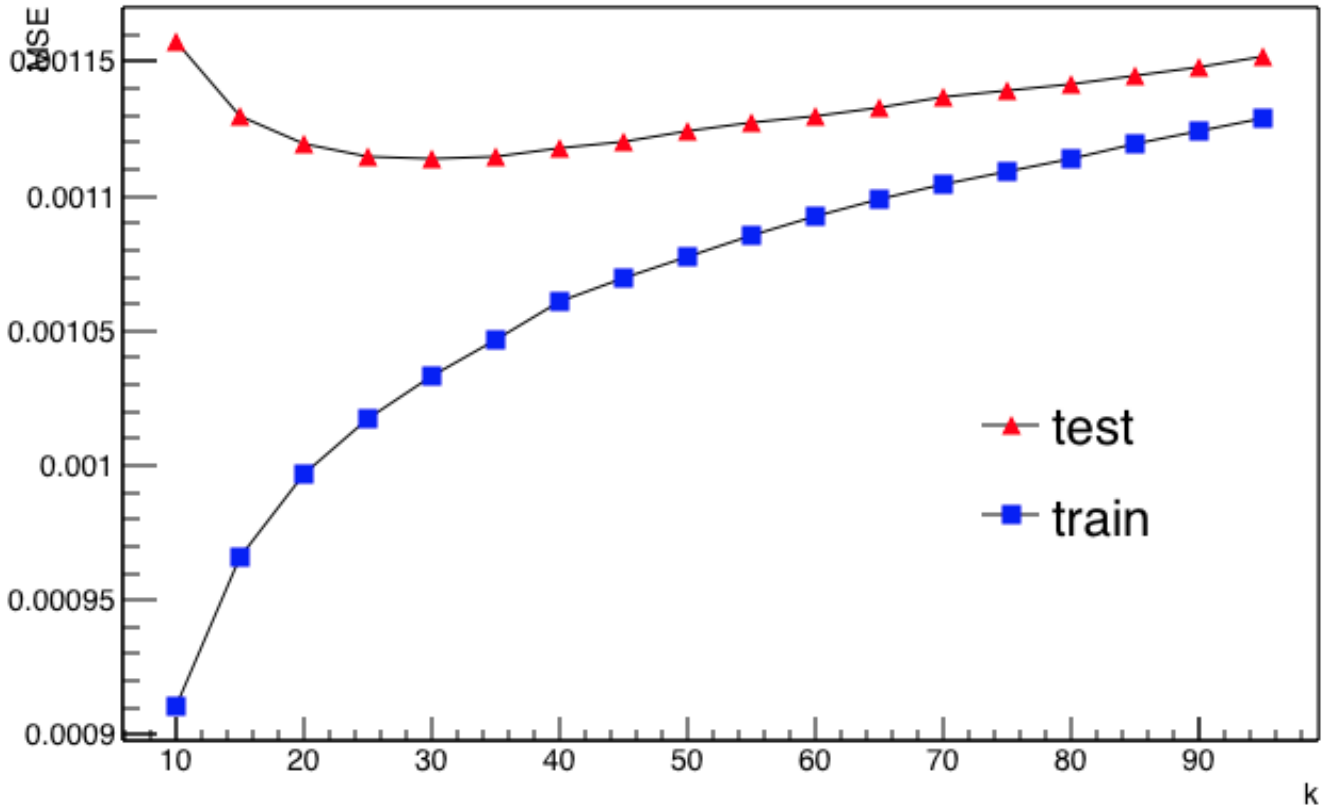
Comments

- Maybe we don’t need to choose the best model, instead, we can use weighted models for final pT LUT
- Maybe we can use different model for different input variables, some model may perform better than others in certain parameter region(eta)


Mean Square Error
0.0035
0.003
0.0025
0.002
0.0015
10
20
30
40
50
60
70
80
90
k
Mode #14: test
Current EMTF
dPhi12,23
dPhi12,23+theta
dPhi12,23+theta+bend_1
dPhi12,23+theta+bend_1+dTh13
dPhi12,23+theta+bend_1+dTh13+FR_1
dPhi12,23+theta+FR_1
dPhi12,23+theta+dTh13
dPhi12,23+theta+FR_1+bend_1
dPhi12,23+theta+FR_1+dTh13

"nkNN=30:ScaleFrac=0.8:SigmaFact=1.0:Kernel=Gaus:UseKernel=F:UseWeight=T:!Trim"

MSE = Deviation^2/N



Mode: 15
Input variables: dphi12, dphi23, dphi34
Target variable: 1/Gen pT

/afs/cern.ch/work/a/abrinke1/public/EMTF/Analyzer/ntuples/EMTF_MC_NTuple_SingleMu_noRPC_300k.root

BDT parameters(mode 15) from Andrew B.(check)

For mode 15

- 400 trees
- Depths: 5
- 1/pT weight(0-120 GeV), Log2(pT) is better target than 1/pT
- For very high pT, unweighted events better(>120 GeV)
- Input variables: FR bits bring significant improvement at low and high pT
- In addition to track theta, FR 1, and dPhi 1-2, 2-3, and 3-4(LUT v1), add combinations of dPhis, and ring number of station 1
- https://indico.cern.ch/event/608207/contributions/2451751/subcontributions/218758/attachments/1402616/2142649/2017_01_26_Mode_15_BDT.pdf

BDT

- Good: little tuning required(simple), robust

- Drawbacks:

1. will ignore non-discriminating variables as for each node splitting only the best discriminating variable is used
2. theoretically best performance on a given problem is generally inferior to other techniques like neural networks.

- See TMVA tutorial