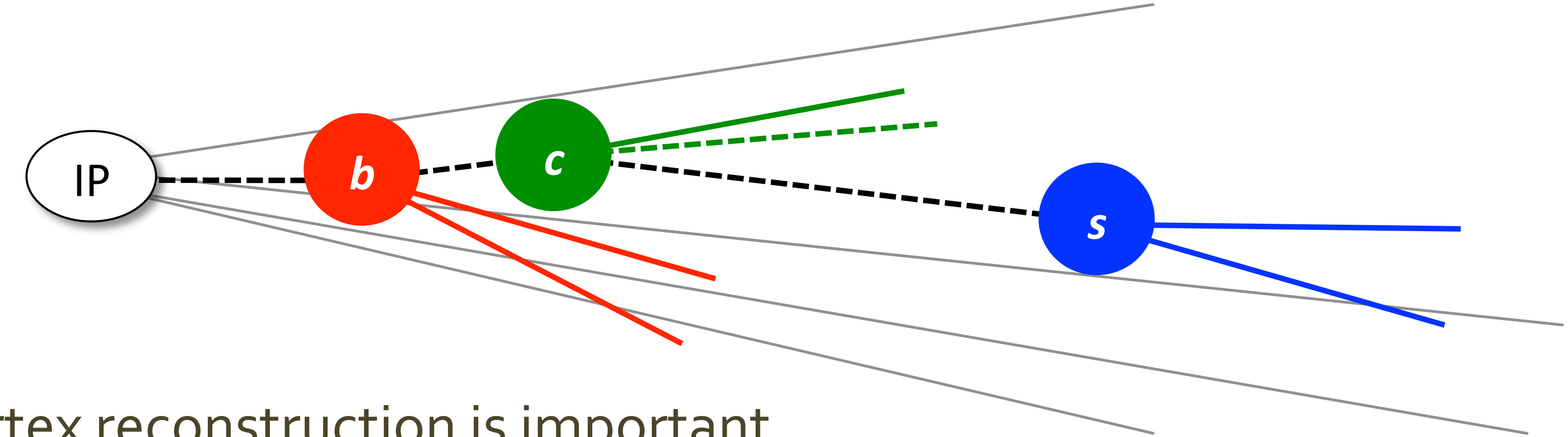
Status of LCFIVertex

Tomohiko Tanabe, Taikan Suehara
The Univ. of Tokyo

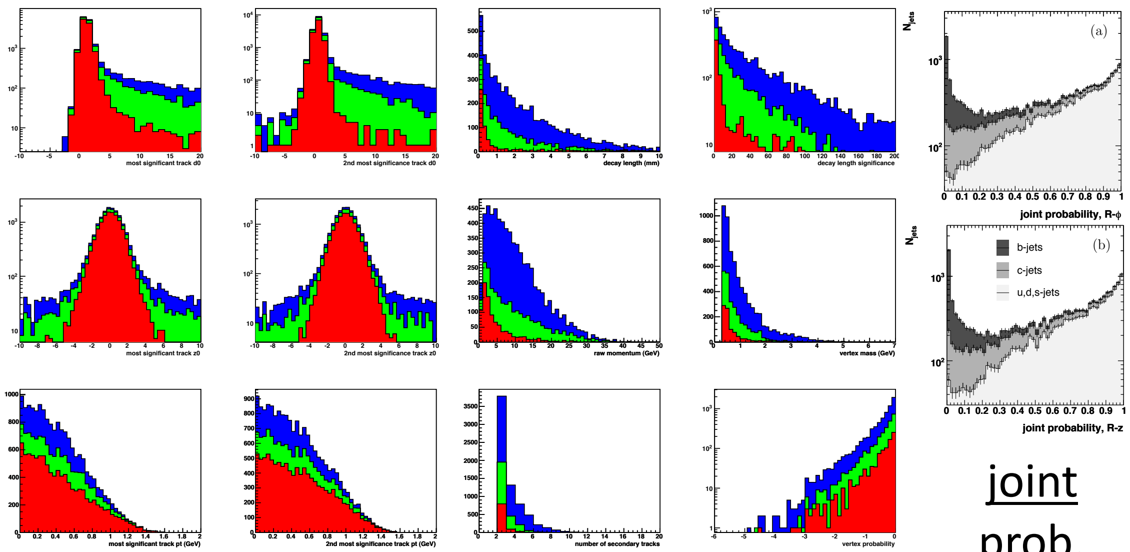
ideal flavor tagging

- reconstruct the entire decay chain (b -> c -> s) in a jet



  - vertex reconstruction is important
    - but vertices cannot be made from a single track
      - use track measurements (impact parameter)
  - presence of neutral particles
    - missing correction by using p_T
  - lepton ID: energetic/isolated leptons is a sign of heavy quark decays
- key is variable combination
  - likelihood, multivariate analysis (e.g. ANN)
  - event categorization (discrete variables)

LCFIVertex



input variables (tracks)

input variables (vertex)

joint prob.

neural net

b-likeness
c-likeness
parton charge

Tracking → Particle Flow (Pandora) → Jet Clustering (Durham) → Vertex Finder (ZVTOP) → Flavor Tagging with Neural Net → Charge Reconstruction

LCFIVertex framework

- improvements in **vertex finding, jet clustering, flavor tagging** in a unified way
  - creation of a new framework suited to this task
    - data types: event, track, neutral, mcparticle, jet, vertex
    - algorithm collections: old LCFIVertex kernels, new algorithms in vertex finding, jet clustering
    - LCFI's original neural net implementation is difficult to maintain/extend
      - replaced by TMVA (ANN and BDT)
        - » use of other frameworks possible with a bit of work
    - i/o integration as a marlin processor (through adapter objects)
      - i/o with other framework also possible
        - » standalone executable mode
        - » root macro invocation mode
    - modular design: algorithms can be turned on/off through XML files
  - **ROOT dependency**: use of TMVA requires the latest ROOT version

from the user's perspective

- prepare training and testing samples
- define categories (one classifier response for each category)
- define the input variables for each category
- write code implementing the input variables
  - the "default" variables are already implemented
    - a user can add any arbitrary variables with a bit of coding work
      - if proven to be effective in the general context, will be added to the official repository
- specify jet clustering and vertex finding algorithms to use
- invoke automated training process
  - normalization of input variables
  - creation of ntuples containing the variables
  - training of each sample (takes time)
  - produce intermediate plots, to be checked by the user
  - creation of weight files, to be used by the classifier in the next phase
- get final output, specifying the expected background rates

status of LCFI code

- implementation of new framework nearing completion; release expected soon
- for comparison purposes the original LCFIVertex implementation is left untouched
  - our new framework could be installed as a separate package, or inside a new directory within the old package (any preferences?)
- management of LCFI code is done through the GIT software
  - originally written for use with the Linux kernel source development, superior to cvs/svn in many ways
    - fast, distributed (central repository NOT needed)
  - we would like to propose this system for those interested in contributing to the project with code

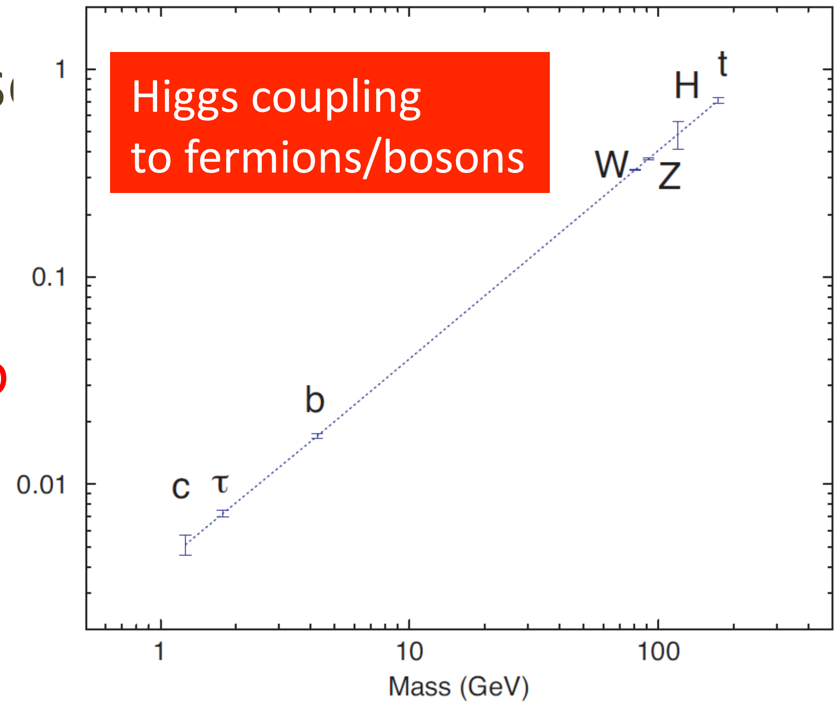
summary and future work

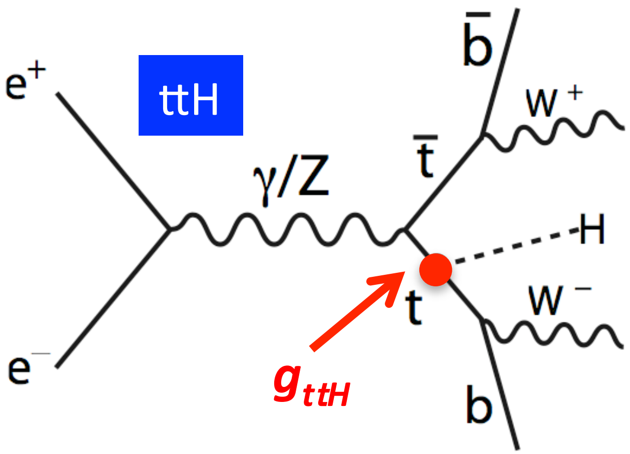
- release of LCFI new framework soon (1~2 weeks)
- establish common development model, expanding the developer base
  - with close connection with CLIC (Jan, Lucie)
- lepton ID inside jets is highly desired
  - need help from PFA expert (Mark?)
- application of new framework in the ZHH analysis
  - talk by Taikan in the main meeting

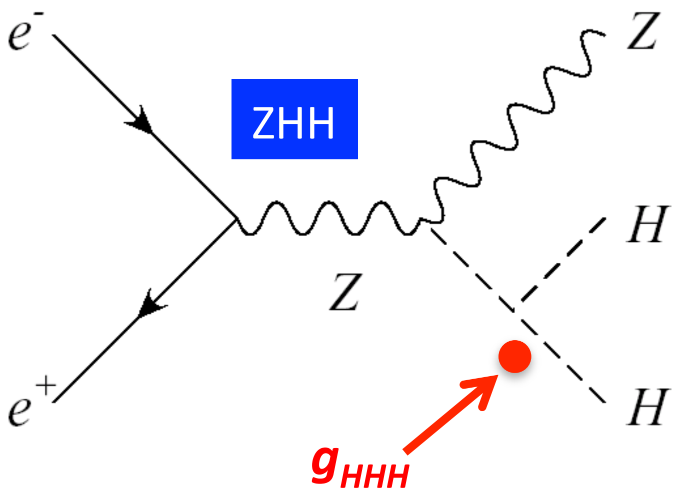
backup

motivation

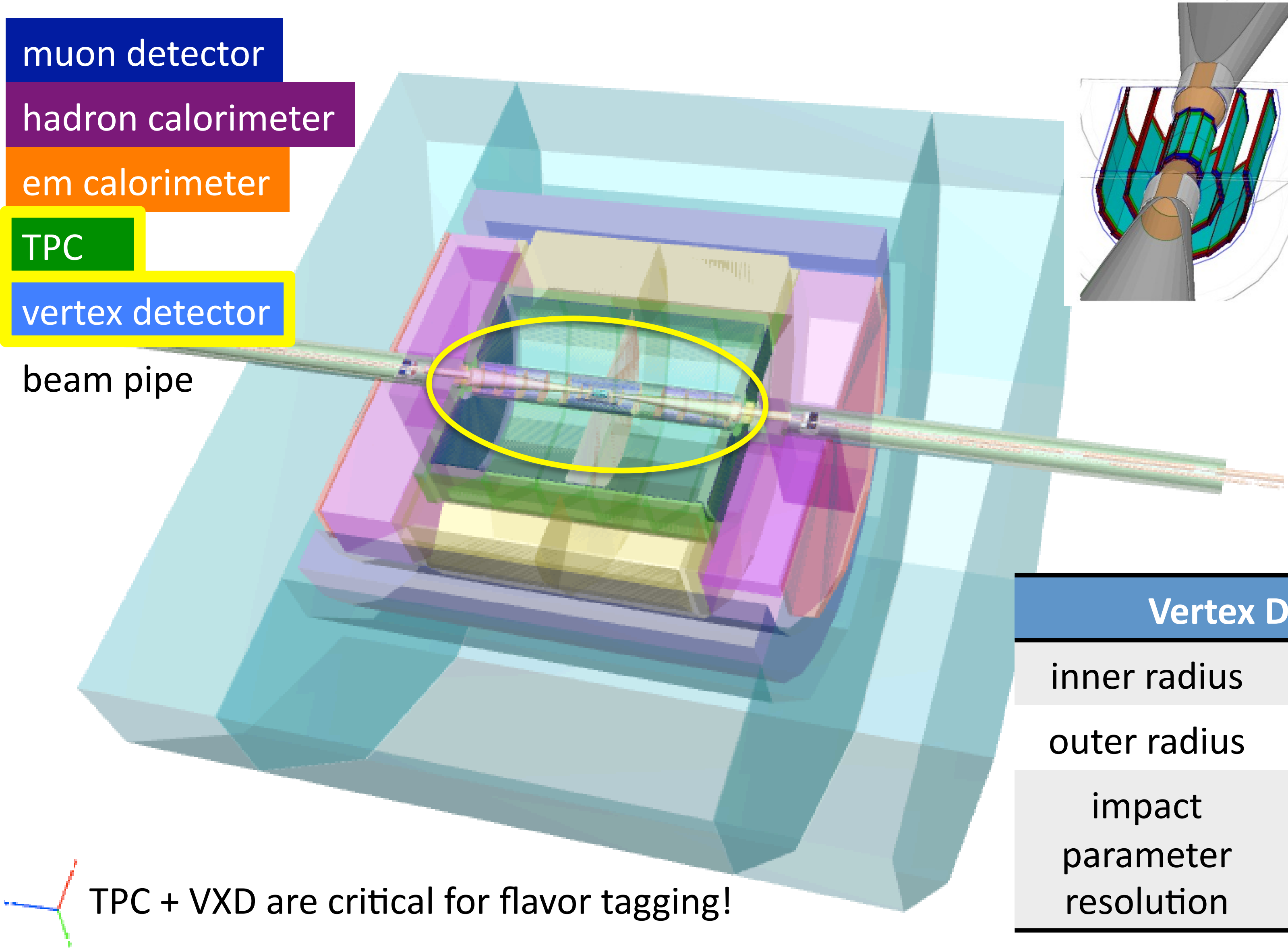
- Many important physics process
have multiple heavy flavor jets
  - Higgs BF : H -> bb, H -> cc
  - Higgs self-coupling : ZHH -> qqbbb
  - top-Yukawa : ttH -> bWbWbb
  - top physics : tt -> bWbW

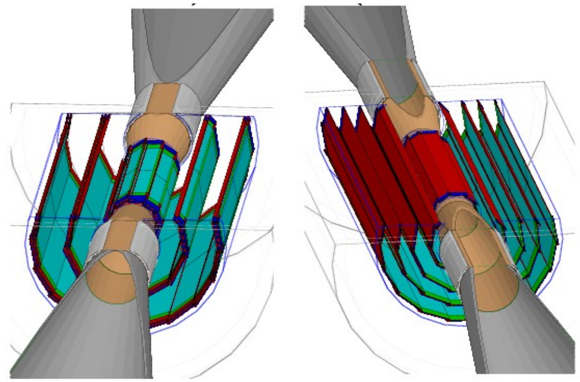






ILD Detector



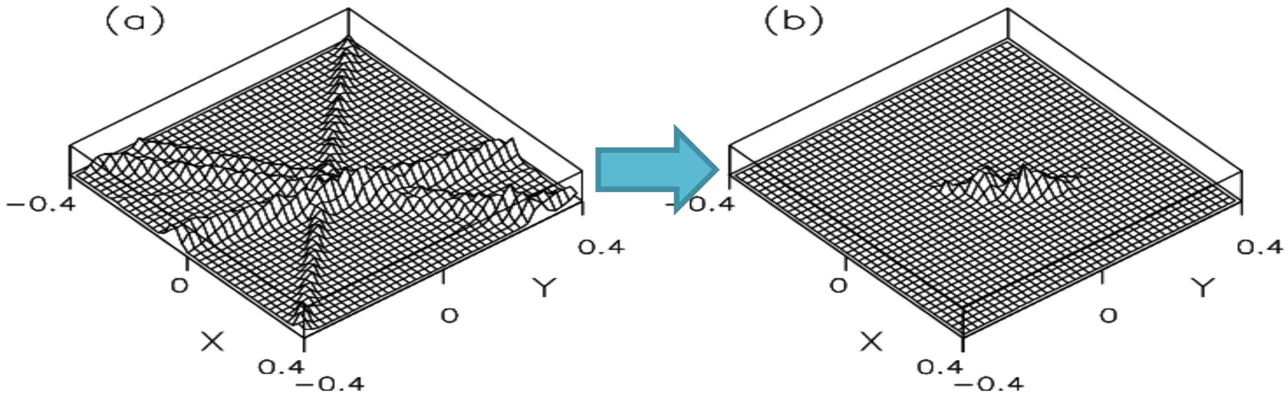


TPC + VXD are critical for flavor tagging!

Vertex Detector	
inner radius	15 mm
outer radius	60 mm
impact parameter resolution	< 5 mm (high momentum)

vertex finding algorithms

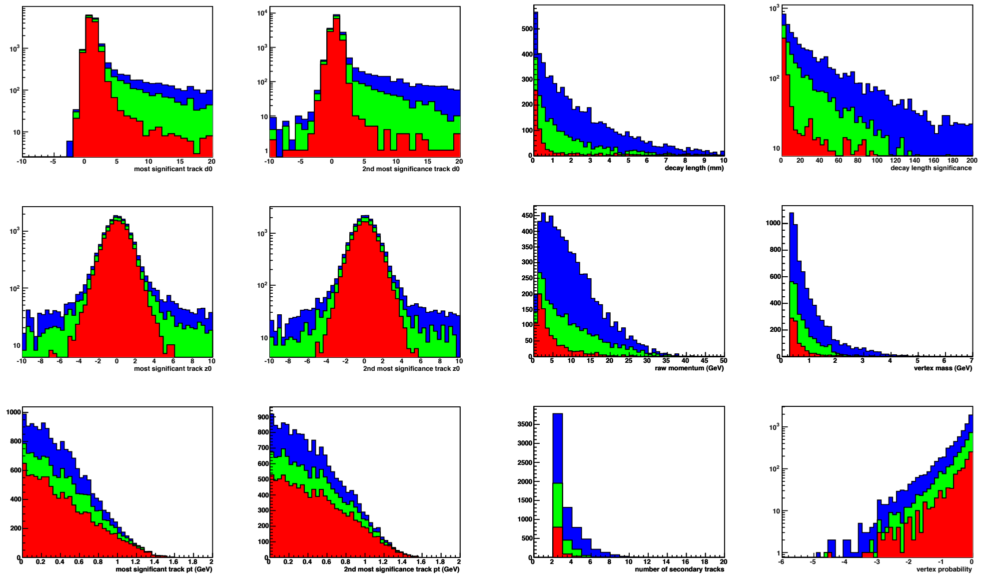
- topological vertex finder (ZVTOP)



  - can find vertices for arbitrary topology with any number of tracks（must be separated from the primary vertex）
  - it takes lots of CPU time due to having to evaluate the vertex function at many points in space
- tear-down
  - start from a set of tracks, remove tracks which are inconsistent (large chi-squared contribution)
  - if the primary tracks are properly removed, vertices can be found with high efficiency
- build-up
  - using track pairs as seed, attach other tracks
  - good seeds lead to good vertices

LCFI input variables

- previously in LCFI:
  - three categories, trained independently:
    - # vertex = 0
    - # vertex = 1
    - # vertex >= 2
  - for # vertex = 0 (8 variables):
    - d_0 impact parameter (1)
    - d_0 impact parameter (2)
    - z_0 impact parameter (1)
    - z_0 impact parameter (2)
    - track momentum (1)
    - track momentum (2)
    - d_0 joint probability
    - z_0 joint probability
  - for # vertex = 1, >=2 (8 variables):
    - d_0 joint probability
    - z_0 joint probability
    - vertex decay length
    - vertex decay length significance
    - vertex momentum
    - pt-corrected vertex mass
    - vertex multiplicity
    - vertex probability from the fitter



input variables

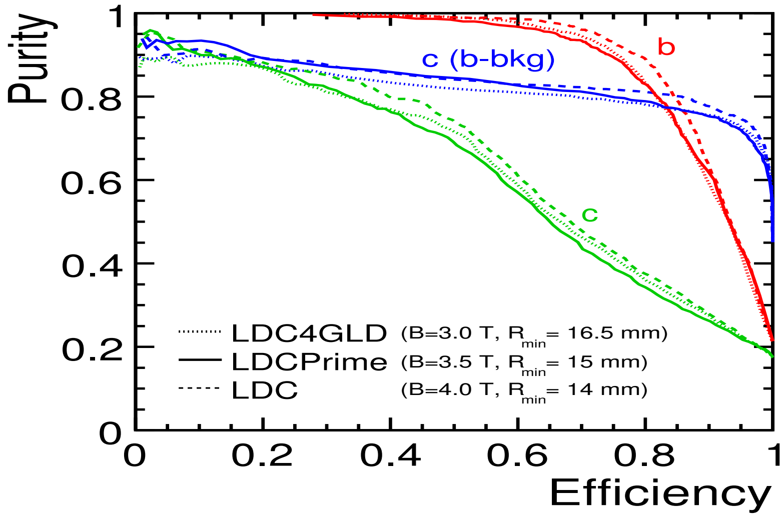
- additional variables
  - boosted sphericity
  - vertex displacement/momentum angle
  - vertex mass (not pt corrected)
  - mass, momentum, decay distance, decay significance for the first and second (if found) vertices
  - distance, significance, and displacement/momentum angle between the first and second vertices

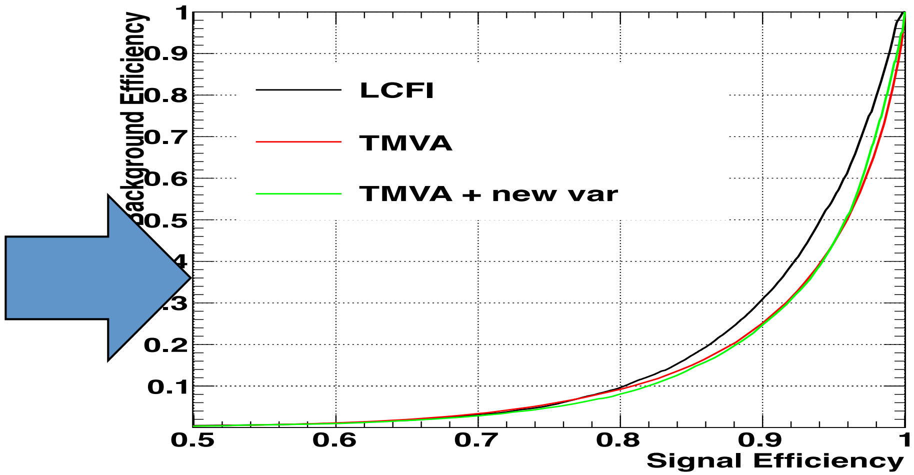
new input variables

- the new variables are trained by BDT
  - for # vertex = 0 (9 variables):
    - d_0 impact parameter (1)
    - d_0 impact parameter (2)
    - z_0 impact parameter (1)
    - z_0 impact parameter (2)
    - track momentum (1)
    - track momentum (2)
    - d_0 joint probability
    - z_0 joint probability
    - boosted sphericity
  - for # vertex = 1 (17 variables):
    - d_0 impact parameter (1)
    - d_0 impact parameter (2)
    - z_0 impact parameter (1)
    - z_0 impact parameter (2)
    - track momentum (1)
    - track momentum (2)
    - d_0 joint probability
    - z_0 joint probability
    - boosted sphericity
    - vertex decay length
    - vertex decay length significance
    - vertex momentum
    - vertex mass (pt-corrected)
    - vertex mass (not pt-corrected)
    - vertex multiplicity
    - vertex probability from the fitter
    - vertex disp/momentum angle
  - for # vertex >= 2 (29 variables):
    - d_0 impact parameter (1)
    - d_0 impact parameter (2)
    - z_0 impact parameter (1)
    - z_0 impact parameter (2)
    - track momentum (1)
    - track momentum (2)
    - d_0 joint probability
    - z_0 joint probability
    - boosted sphericity
    - vertex #1 decay length
    - vertex #2 decay length
    - distance between vertex #1 & #2
    - vertex #1 decay length significance
    - vertex #2 decay length significance
    - separation significance between vertex #1 & #2
    - vertex #1 momentum
    - vertex #2 momentum
    - vertex momentum (combined)
    - vertex #1 mass (not pt-corrected)
    - vertex #2 mass (not pt-corrected)
    - vertex mass (combined, pt-corrected)
    - vertex #1 multiplicity
    - vertex #2 multiplicity
    - vertex multiplicity (combined)
    - vertex probability from the fitter
    - vertex #1 disp/momentum angle
    - vertex #2 disp/momentum angle
    - vertex #1/#2 disp/momentum angle
    - vertex #1/#2 angle

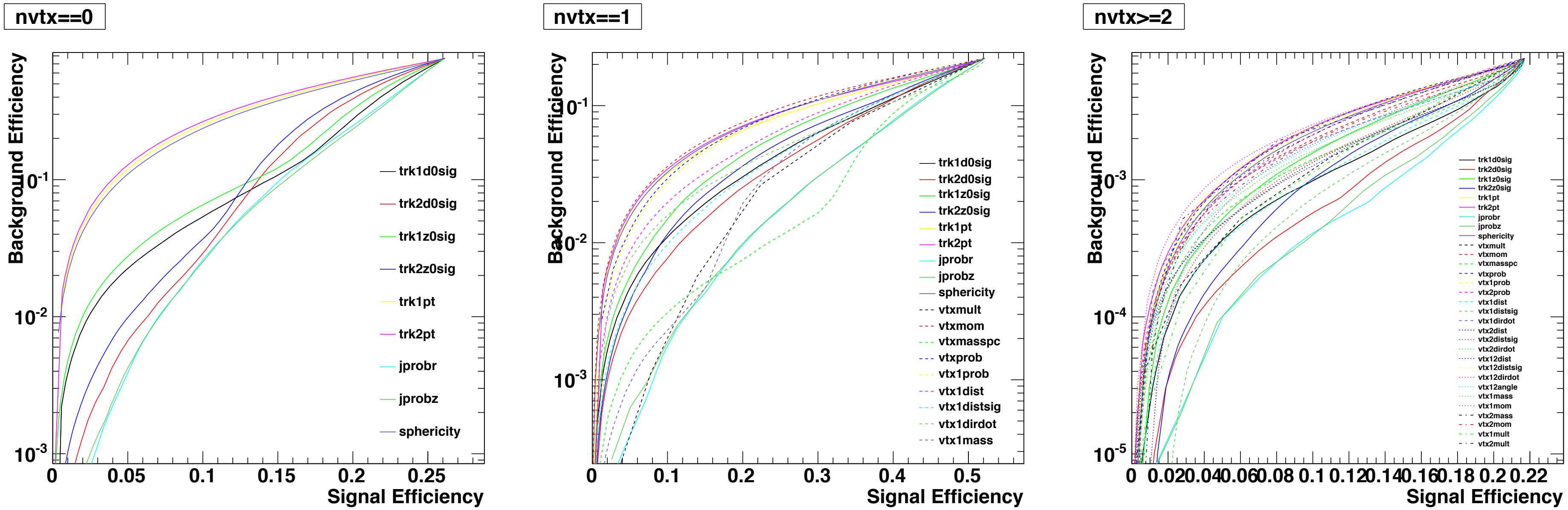
evaluating classifier response

- LoI flavor-tagging evaluation produced purity-efficiency plots
  - but this depends on the fraction of heavy jets, which changes from sample to sample
    - BF(Z->bb)=15%, BF($H_{120\ GeV}$->bb)=68%
- better to use a fraction-independent measure: evaluate using background efficiency versus signal efficiency instead





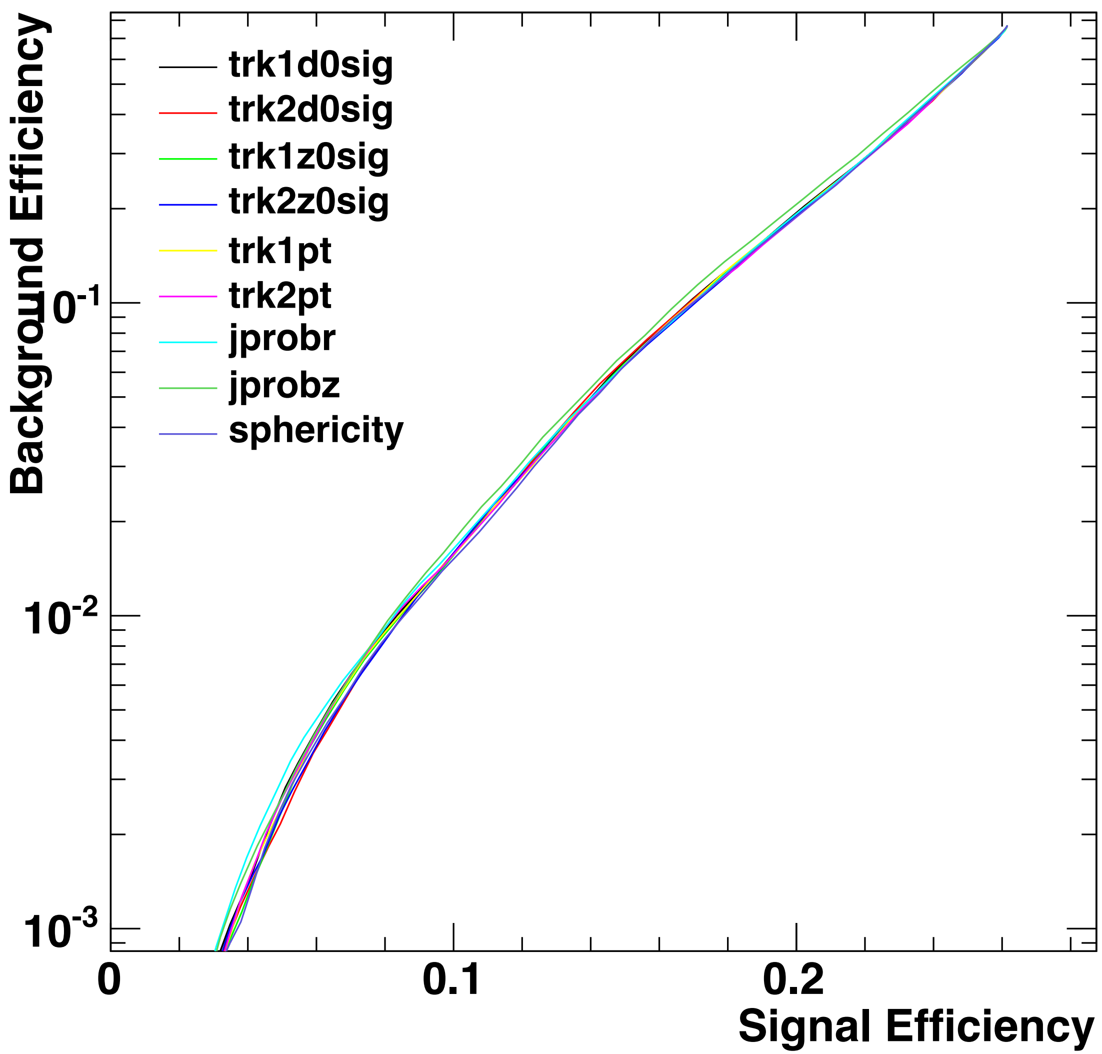
variable ranking



- single variable ranking shows the most useful variables (on their own)
  - joint probabilities, vertex mass
- however, any other uncorrelated variable can help; these plots do not show this effect

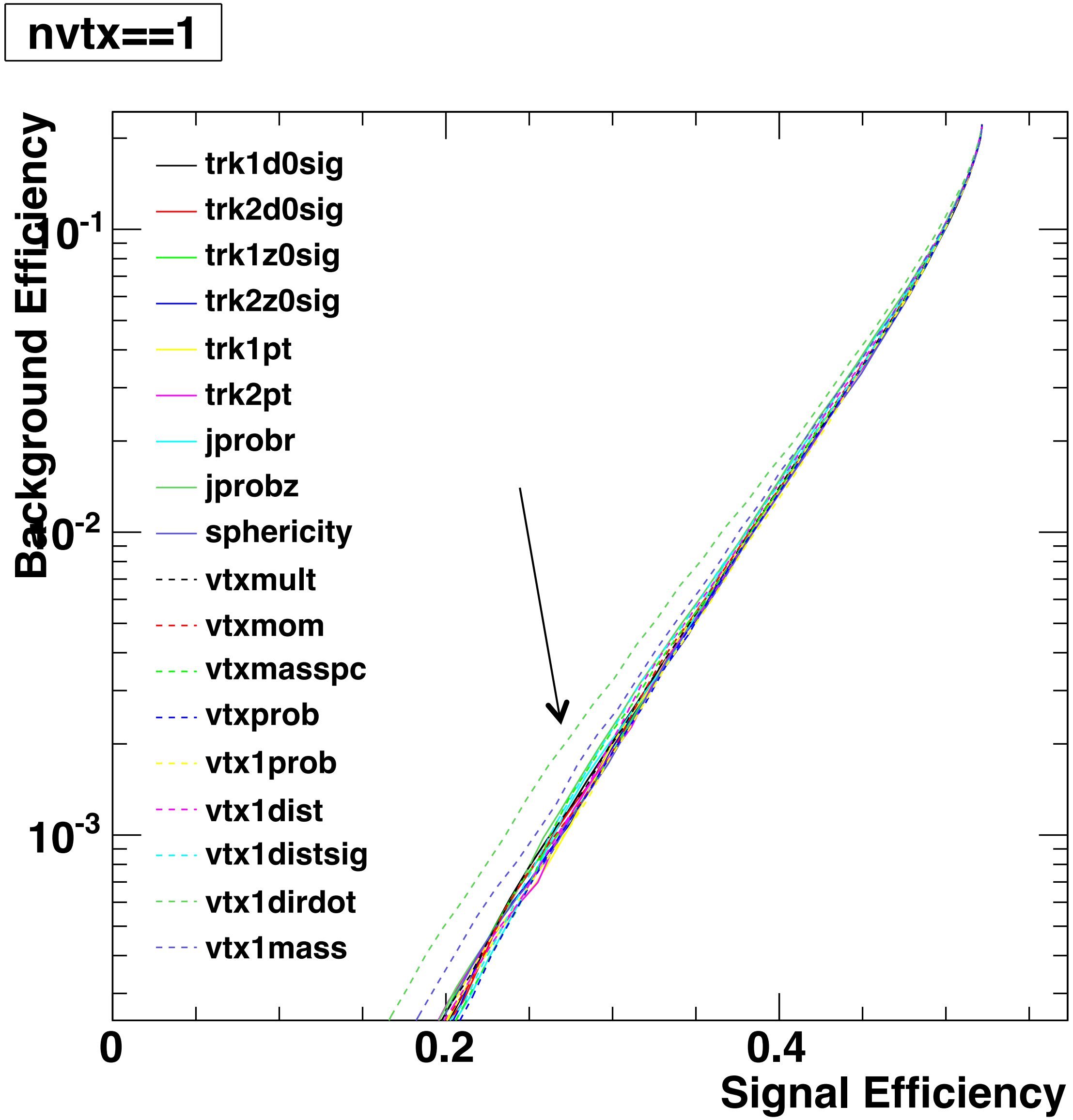
variable ranking (correlation)

nvtx==0



- result of training after removing a variable
- this shows how "unique" this variable is in terms of uncorrelated classifying power
- significantly worse performance after removing the variable shows that it's effective
- for nvtx=0, joint probabilities (both d_0 & z_0) are the most powerful as expected

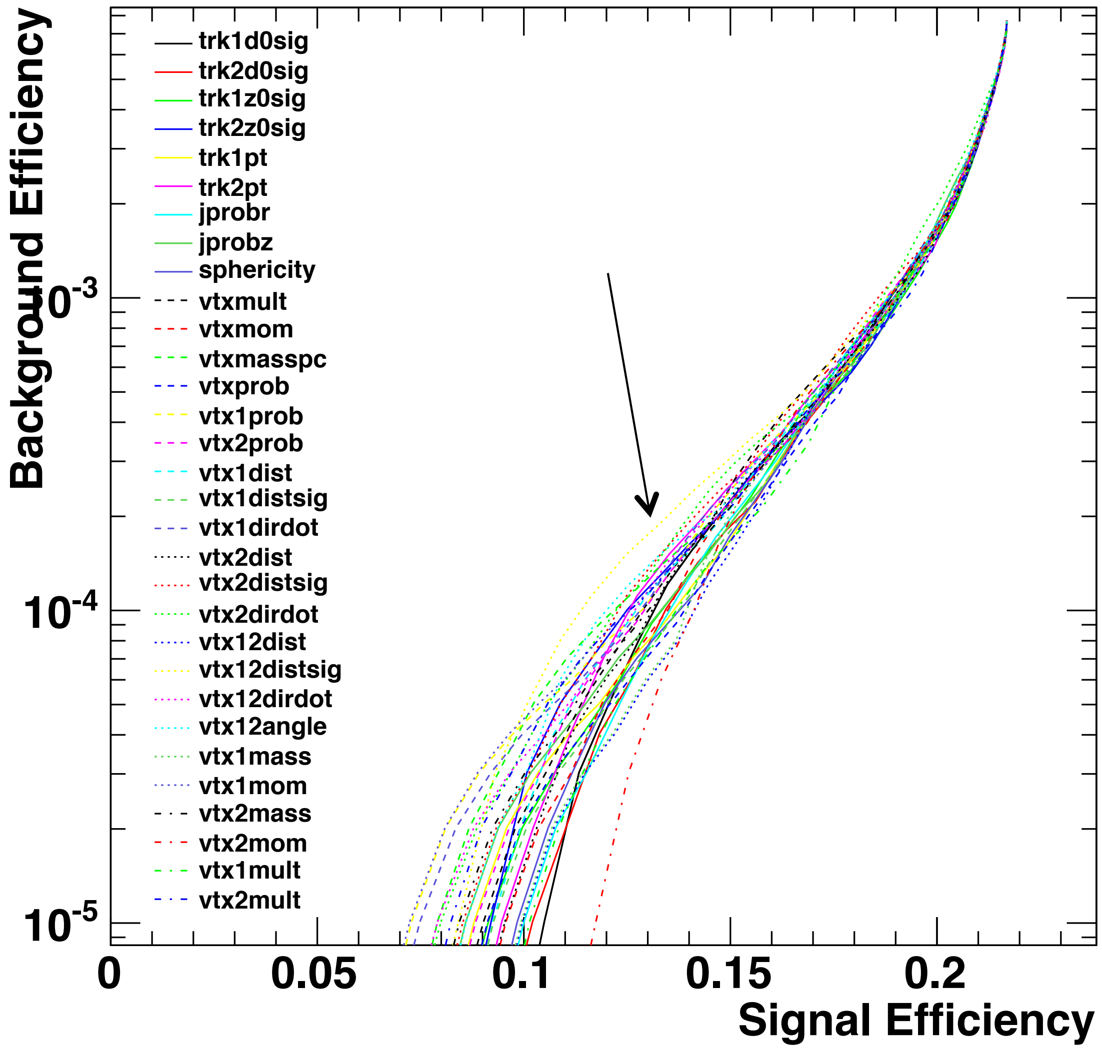
variable ranking (correlation)



- for nvtx=1, the most effective variables are:
    - displacement/ momentum angle of the vertex
    - uncorrected mass of the vertex
- <u>newly added variables are shown to be effective!!!</u>

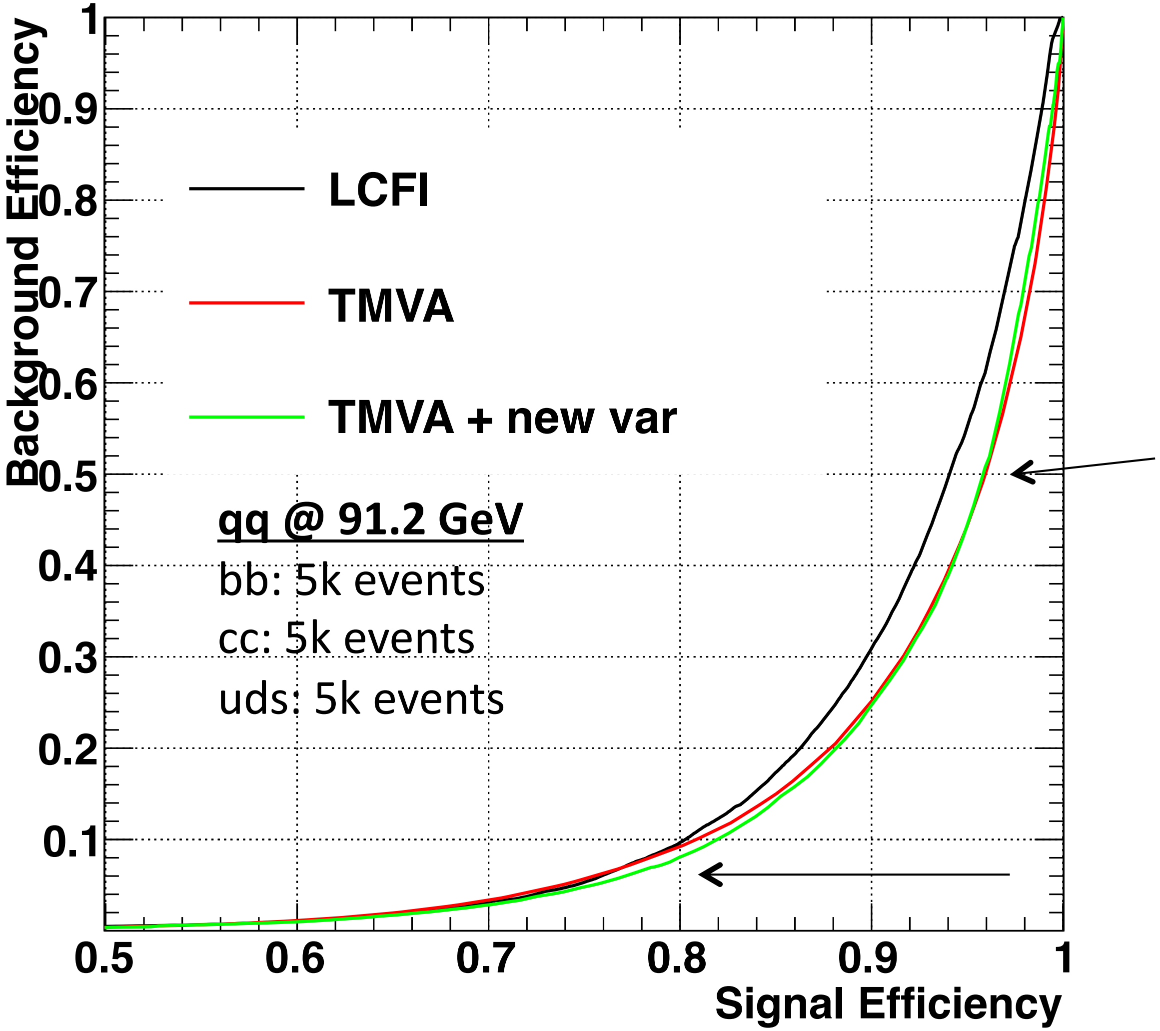
variable ranking (correlation)

nvtx>=2



- for nvtx>=2, the most effective variables are:
    - separation significance between the 1st and 2nd vertices
- AGAIN: newly added variables are shown to be effective!!!

results

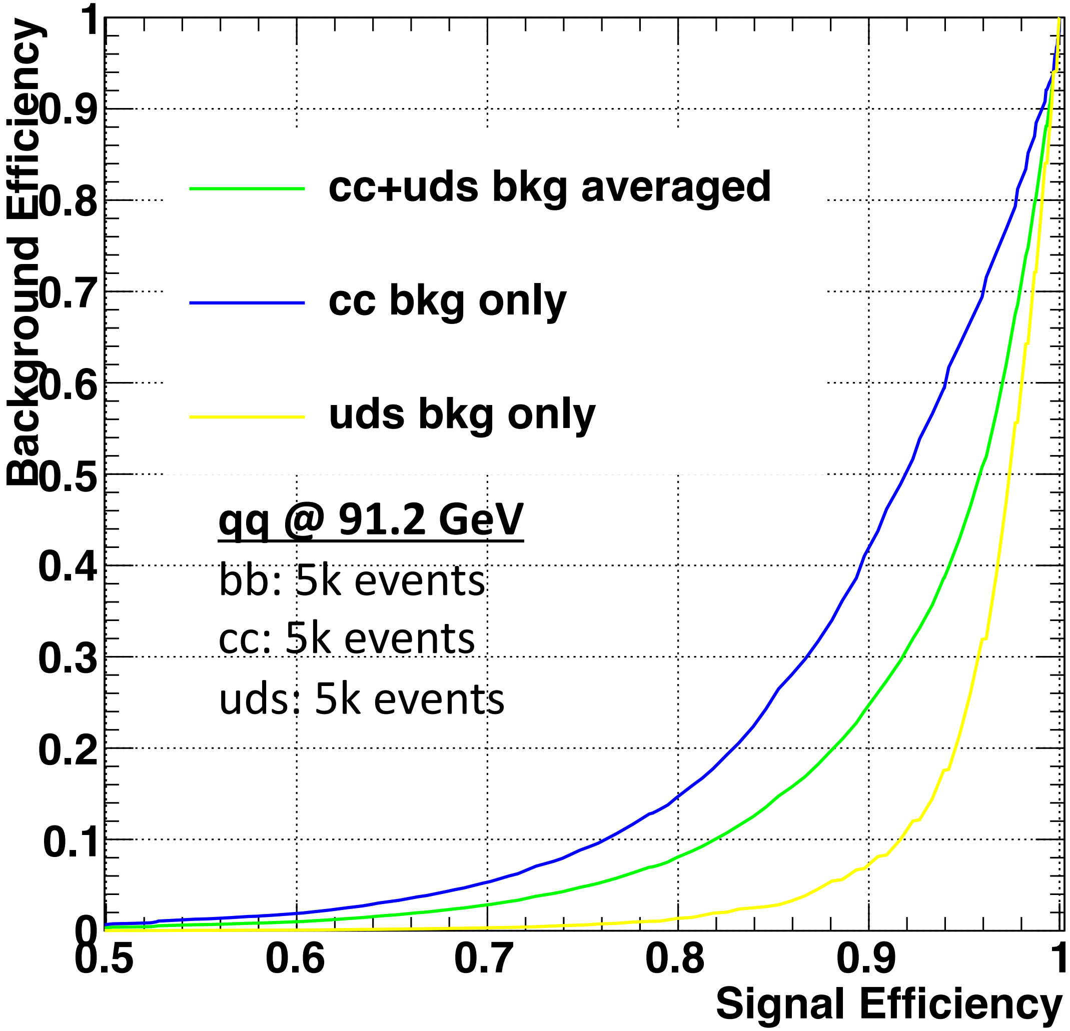


background = cc & uds mixed equally

there is already improvement merely by switching to TMVA (signal eff > 0.8)

more improvement by adding new variables (signal eff > 0.75)

results



uds rejection is pretty good

need more work on charm

summary (part I)

- we have a mature analysis framework to optimize flavor tagging
  - improvements over LCFI: ability to add arbitrary number of categories and input variables, choice of training method (via TMVA)
- performance for qq @ 91.2 GeV has been improved
- next:
  - leptons!
  - dedicated charm tagging
  - optimization at higher jet energies
    - ZHH jets, qq @ 500 GeV
    - qq @ CLIC energies
    - samples have been provided by A. Timoce and S. Posse (thanks, will look at them asap)

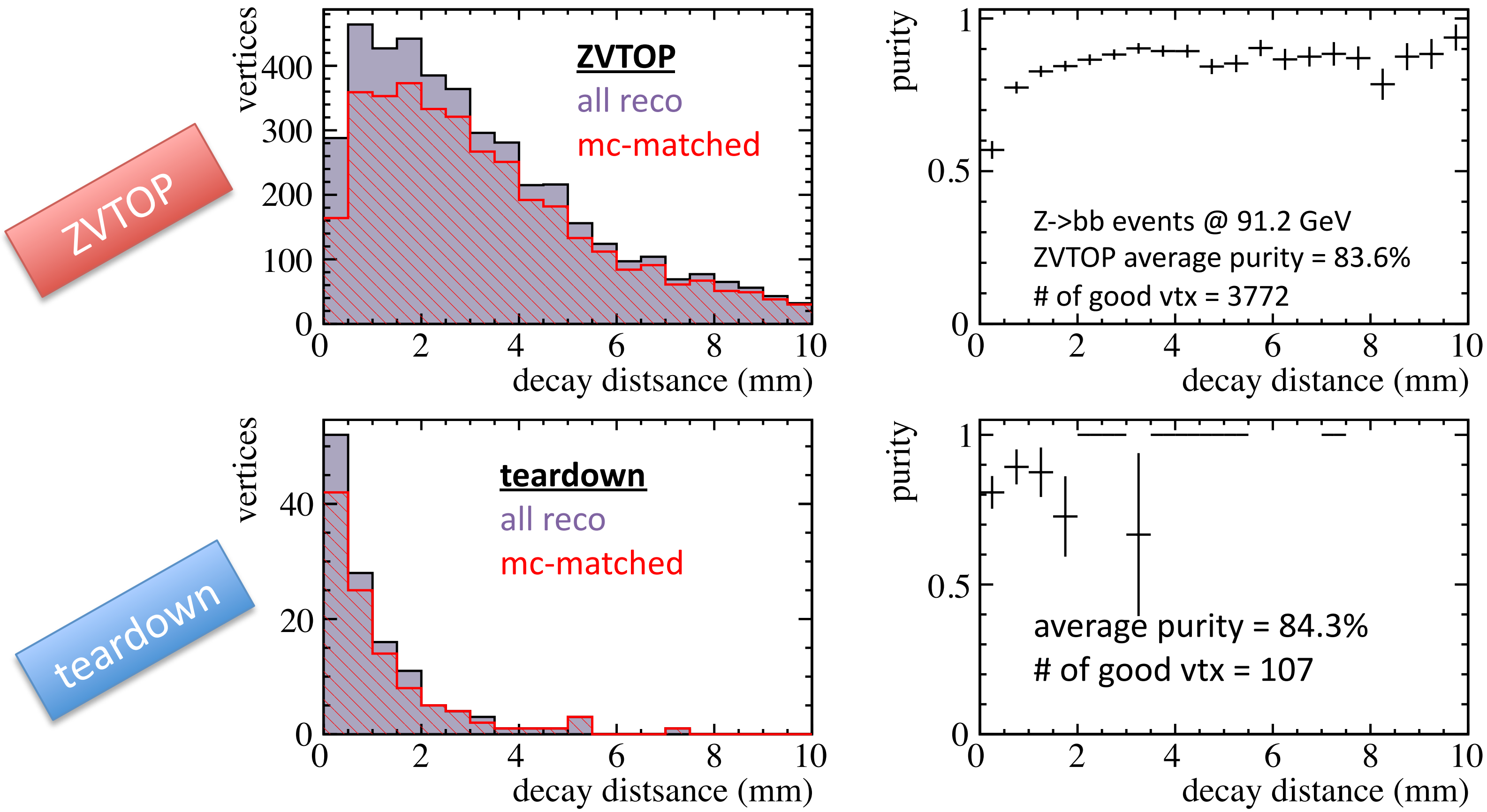
rejection of V° particles

- despite having V° taggers in the Marlin reconstruction chain, our vertex finders still find V°'s (K_S, Lambda, conversions) for two-track vertices
- we apply the following cuts to reject V°'s (reduce uds contamination):
  - cut on the angle θ between the vertex displacement from IP and the V° direction
  - mass requirements
  - K_S: $\cos\theta > 0.999$ & mass 15 MeV within PDG value
  - Lambda: $\cos\theta > 0.99995$ & mass 20 MeV within PDG value
  - conversions: $\cos\theta > 0.99995$ & less than 10 MeV for conversion mass, where the mass is geometrically corrected so that it is calculated using the track dip angles

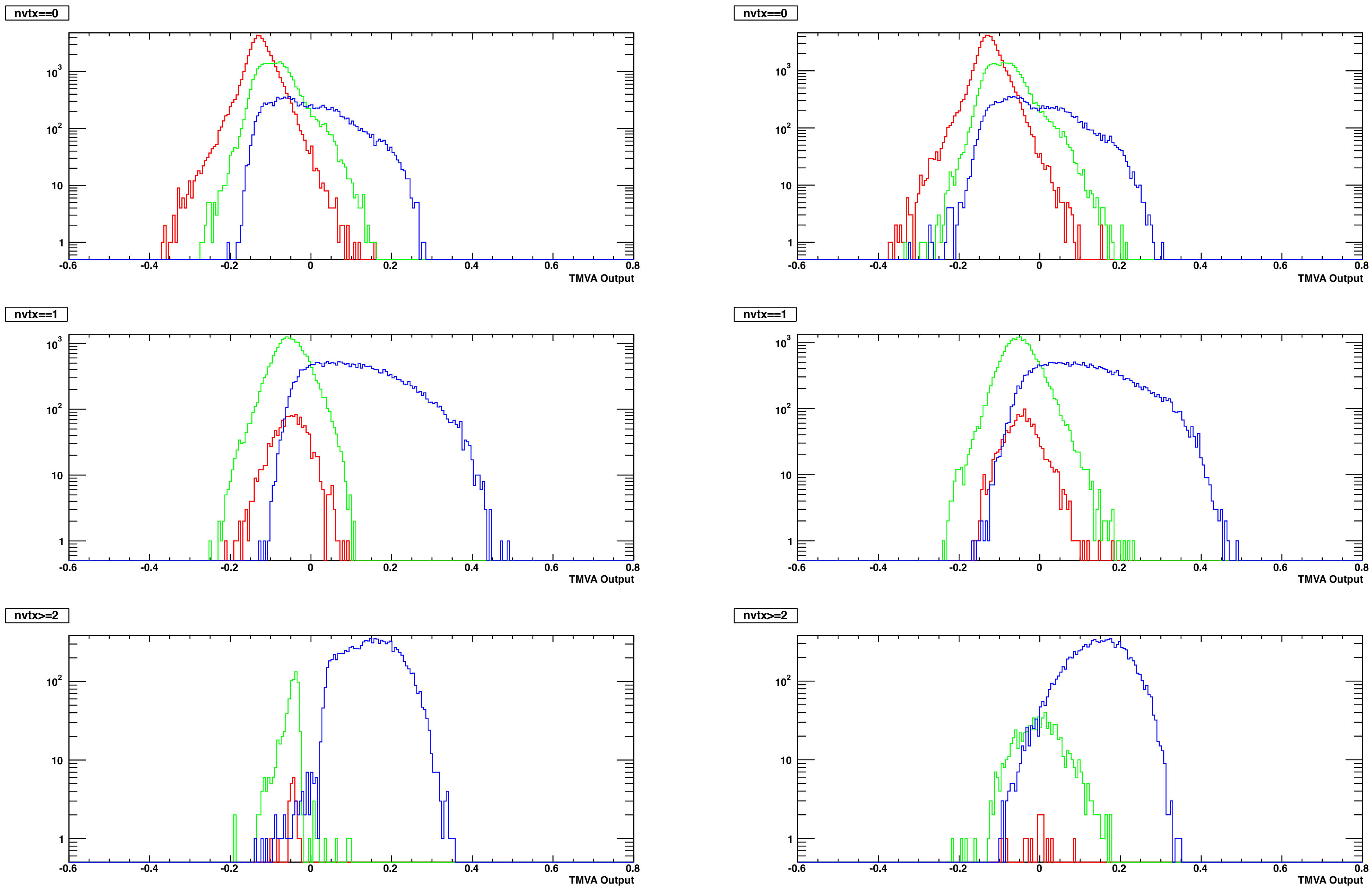
$$m^2_{\mathrm{conv}} = 2|\vec{p}_1||\vec{p}_2|(1 - \cos\Delta\lambda_{12})$$

	before cut	after cut
K_S	3205	623
Lambda	1482	371
conversions	2544	278
other two-track reco vertices	30747	30333

ZVTOP + teardown



ZVTOP is not good at vertices near the IP.
Teardown is performed using tracks not used by ZVTOP.
After quality cuts (dist, chi2, ntrks), we find +3% efficiency increase for the same purity.


nvtx==0
nvtx==0
nvtx==1
nvtx==1
nvtx>=2
nvtx>=2
TMVA Output
-0.6 -0.4 -0.2 0 0.2 0.4 0.6 0.8
1 10 10^2 10^3