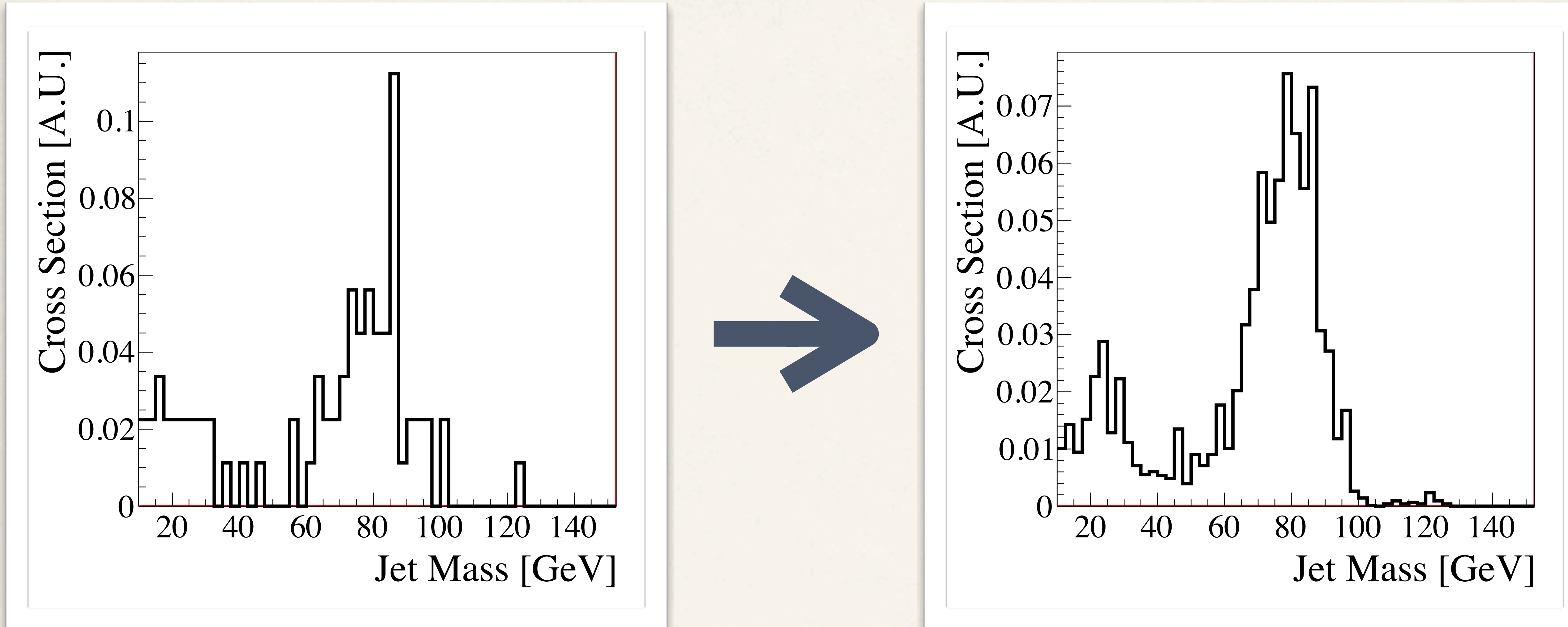


Qjets

David Krohn (Harvard)

NPKI 2/28/12

Based on work with S. Ellis, A. Hornig, T. Roy, and M. Schwartz: arXiv:1201.1914, accepted by PRL and work in progress with above plus D. Kahawala, A. Thalapillil, and L.-T. Wang

Outline

- Quick Review of Jet Algorithms
- Introduction to Qjets
- Implementation
- Example: Jet Pruning
- Future Directions & Conclusions

Takeaway

- Many jet substructure analyses employ trees
- But, more than one tree can plausibly be associated with a jet
  - Typically, we use k_T or C/A to chose the “best” tree
- However, if we force ourselves to only consider a single tree for each jet, we make ourself more susceptible arbitrary choices of the jet algorithm
- By looking at many trees for each jet, we can decrease random fluctuations and create a more powerful analysis

Review of Jets & Jet Substructure

Types of Algorithms

- There are two main classes of jet algorithm
- Sequential recombinations
  - Combine four-momenta one by one
- Cone algorithms
  - Stamp out jets as with a cookie cutter

Focus on these

Sequential Recombination

- Define a distance measure between every pair of four-momenta in an event (jet-jet distances)

$$d_{ij}$$

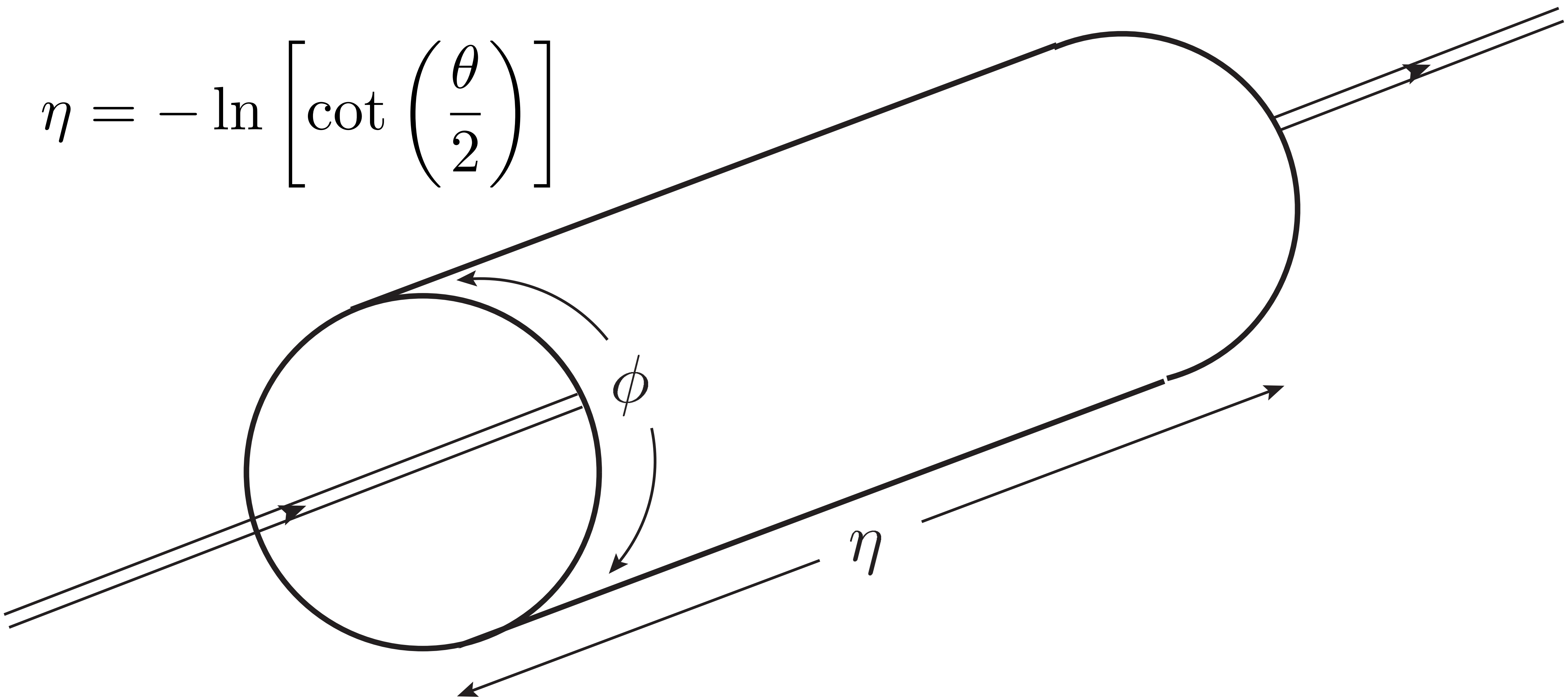
- Define a distance measure for each four-momenta individually (jet-beam distances)

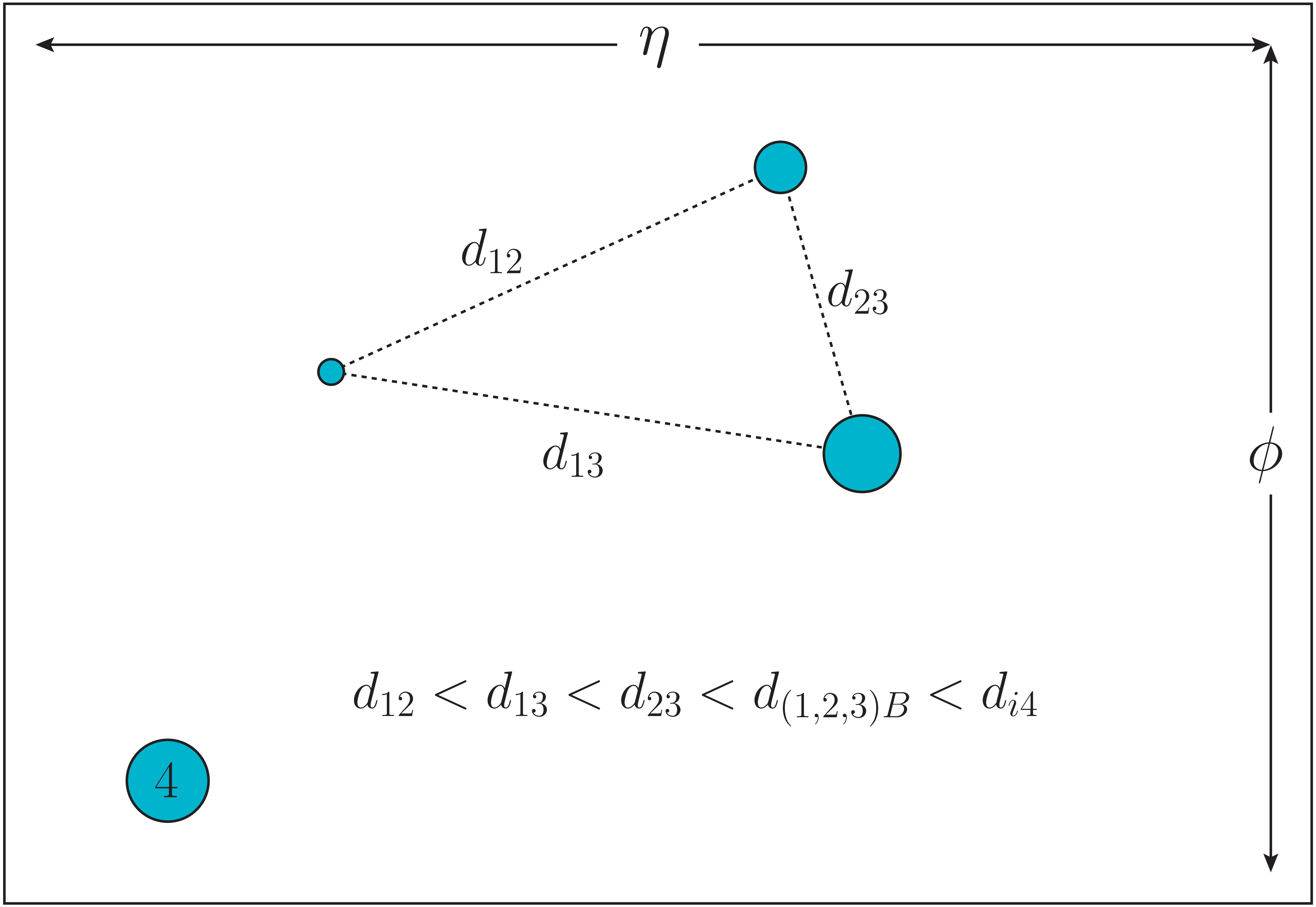
$$d_{iB}$$

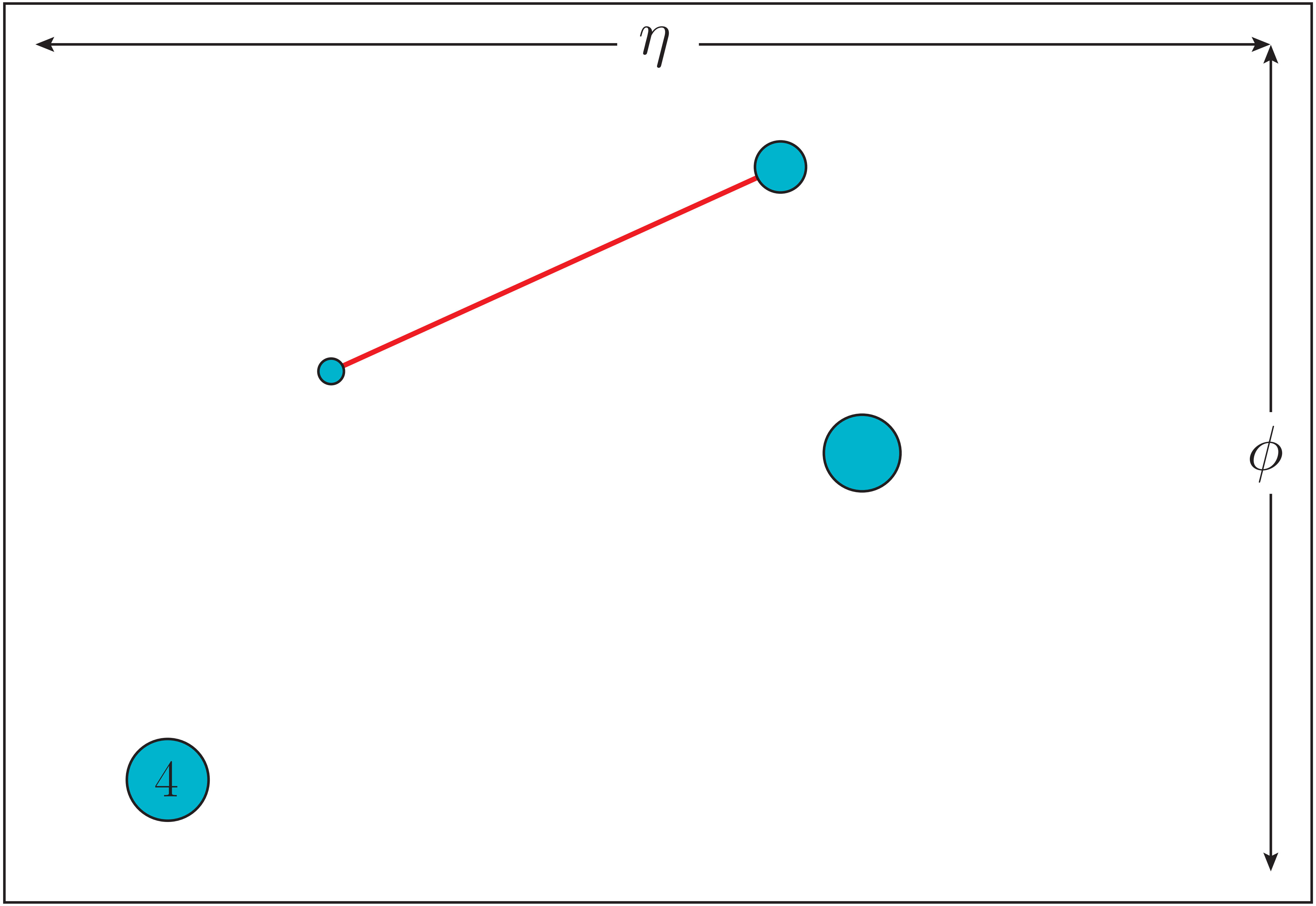
- If smallest distance at any stage in clustering is jet-jet, add together corresponding four-momenta
    - Otherwise take jet with smallest jet-beam distance and set it aside
- Repeat till all jets are set aside
- In this way, jets are constructed by pairwise recombinations - get a tree-like sequence at the end.

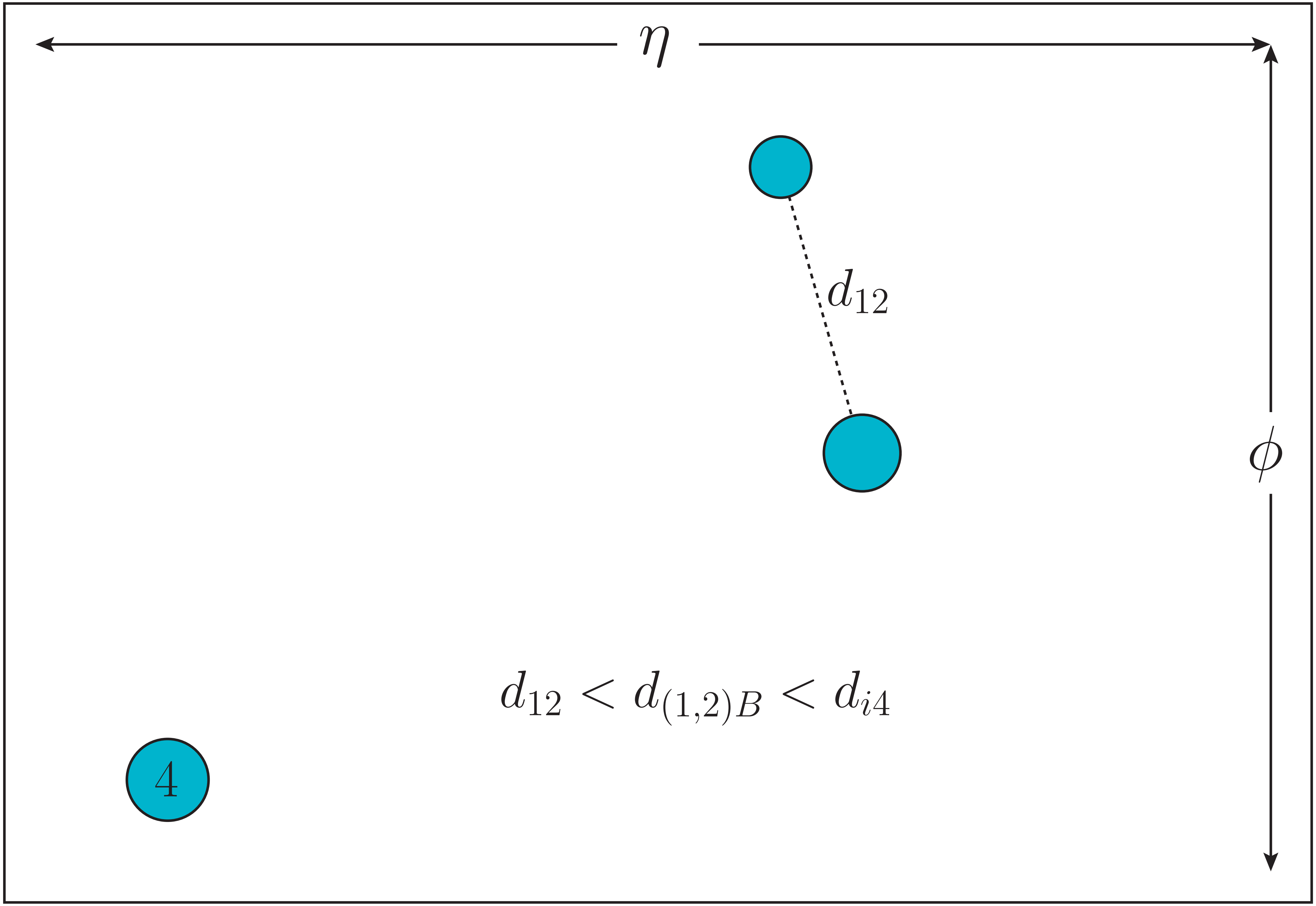
Coordinate System

$$\eta = -\ln\left[\cot\left(\frac{\theta}{2}\right)\right]$$

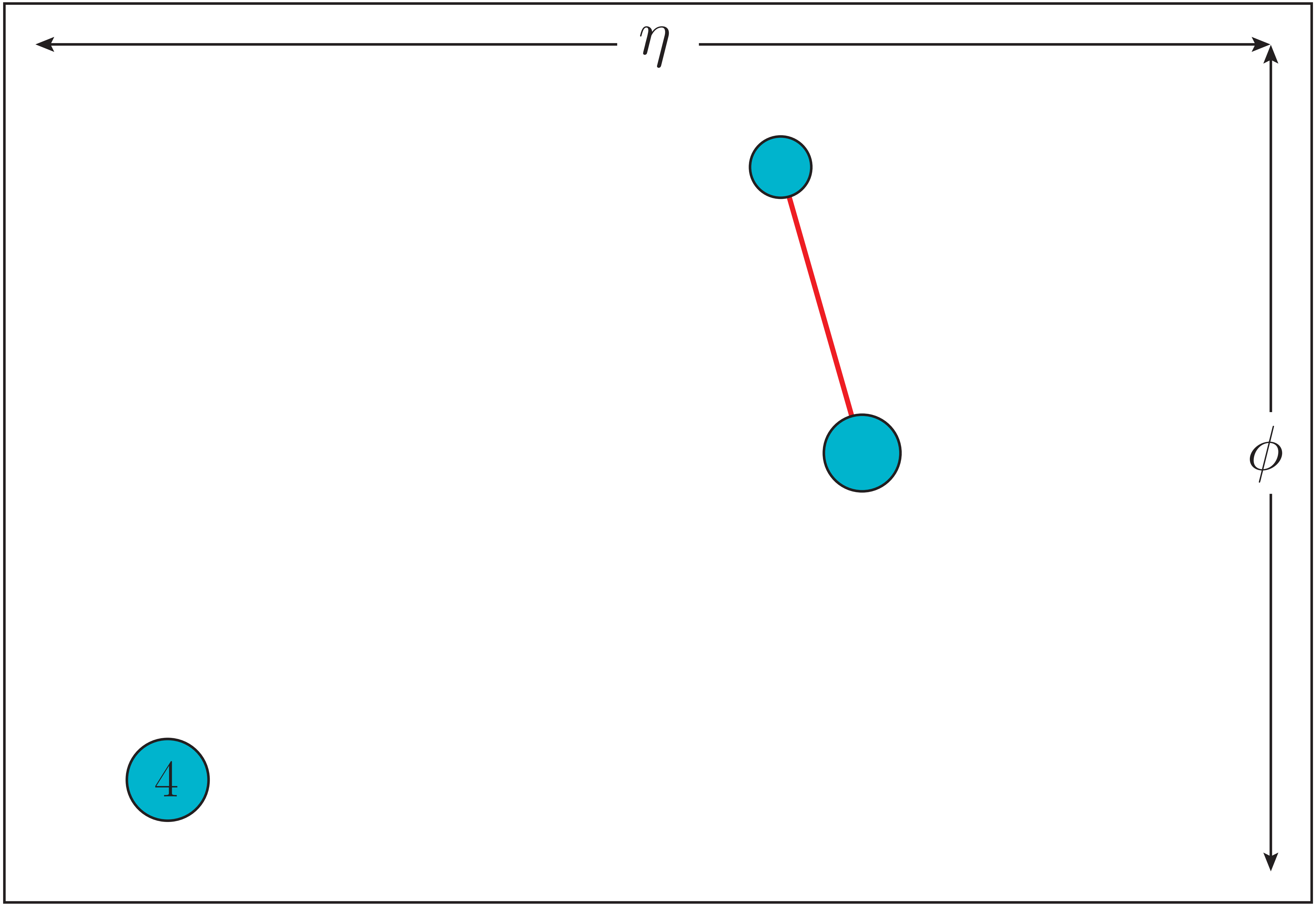



η
ϕ
d_{12}
d_{23}
d_{13}
$d_{12} < d_{13} < d_{23} < d_{(1,2,3)B} < d_{i4}$
4

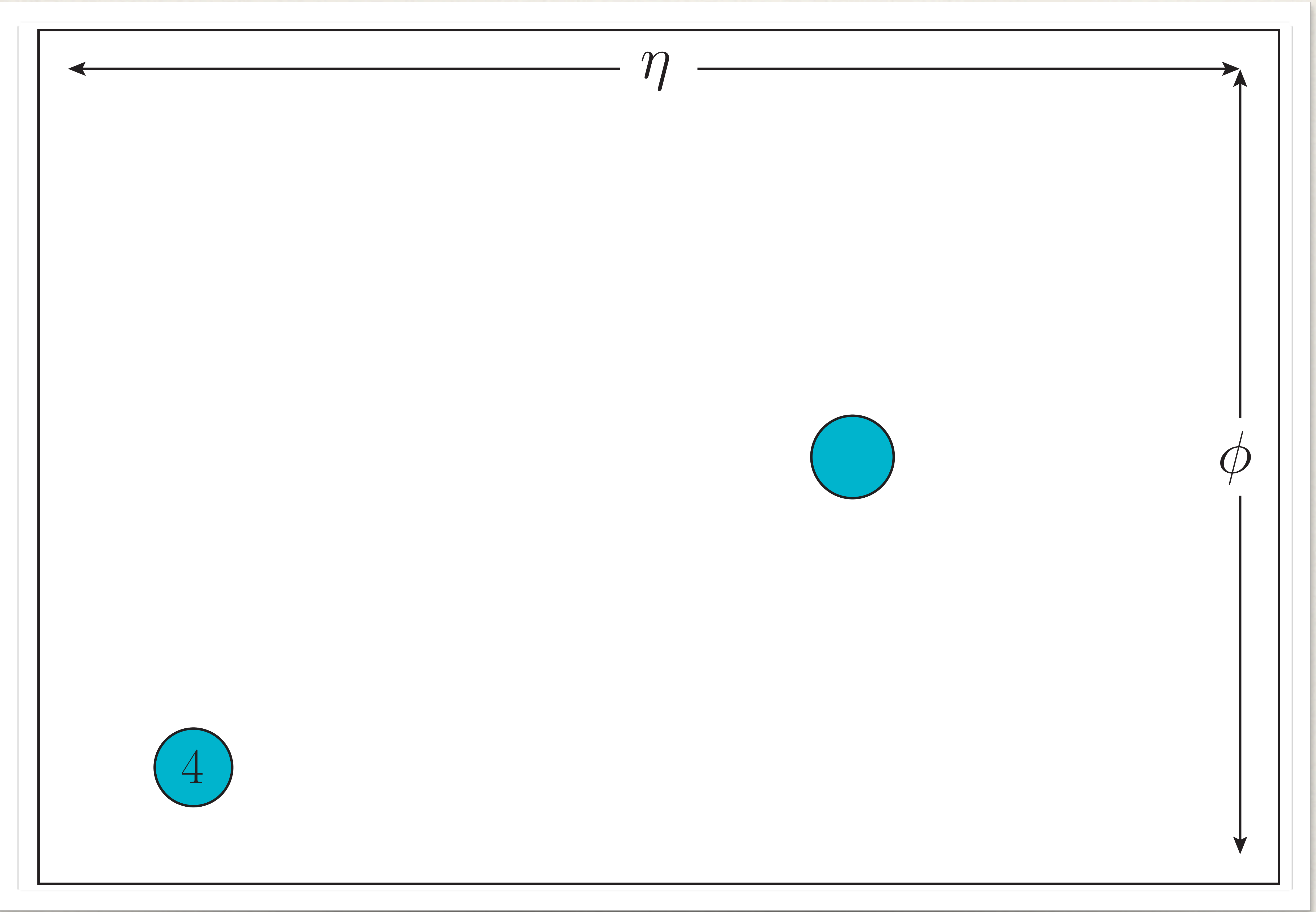

η
ϕ
4



η
ϕ
d_{12}
$d_{12} < d_{(1,2)B} < d_{i4}$
4



η
ϕ
4



Done!

Standard Recombination Algorithms

- k_T algorithm

$$d_{ij} = \min(p_{Ti}^2, p_{Tj}^2)\left(\frac{\Delta R}{R_0}\right)^2, \; d_{iB} = p_{Ti}^2$$

- C/A algorithm

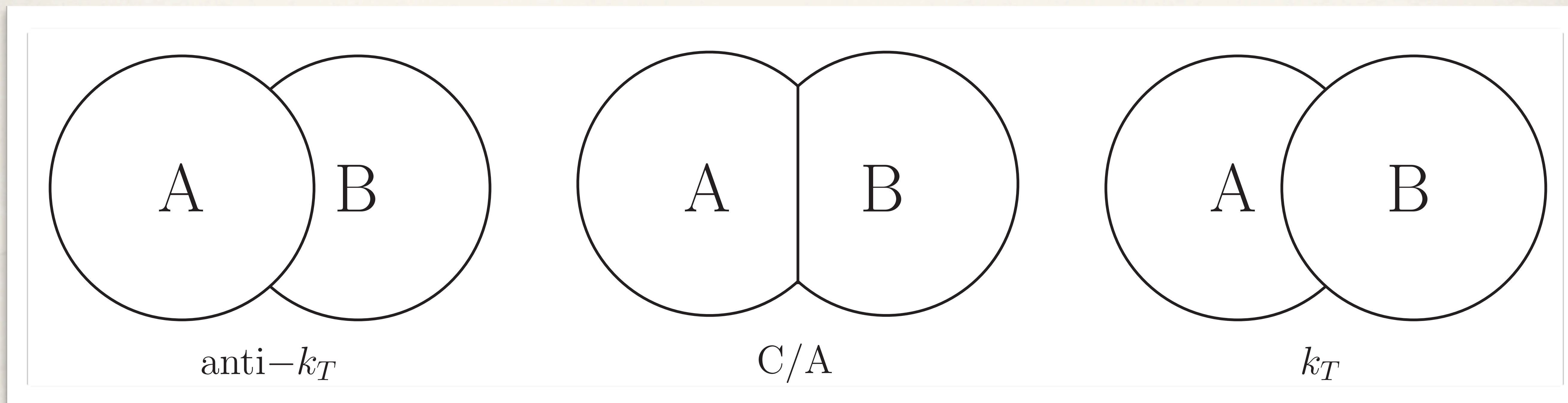
$$d_{ij} = \left(\frac{\Delta R}{R_0}\right)^2, \; d_{iB} = 1$$

- anti-k_T algorithm

$$d_{ij} = \min(p_{Ti}^{-2}, p_{Tj}^{-2})\left(\frac{\Delta R}{R_0}\right)^2, \; d_{iB} = p_{Ti}^{-2}$$

Approximate Jet Behavior:

$$p_{TA} > p_{TB}$$



Hard to Soft

Near to Far

Soft to Hard

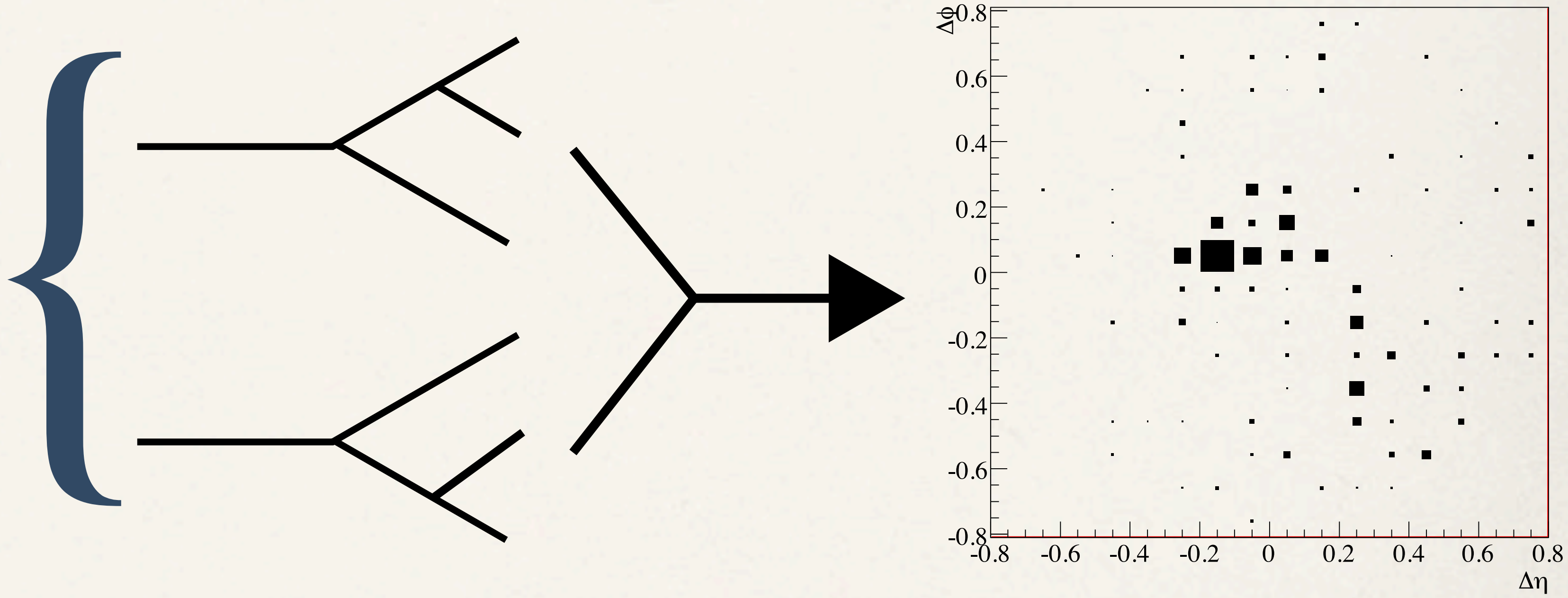
Qjets

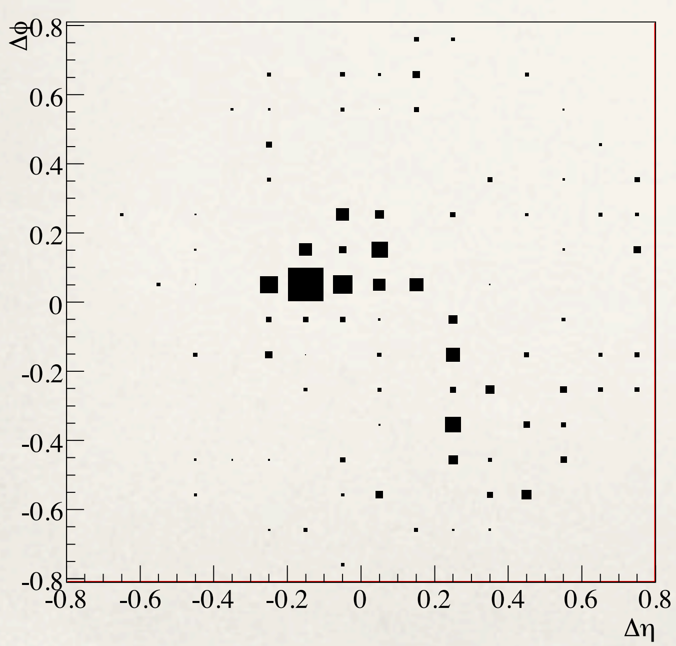
Two Basic Approaches to Substructure

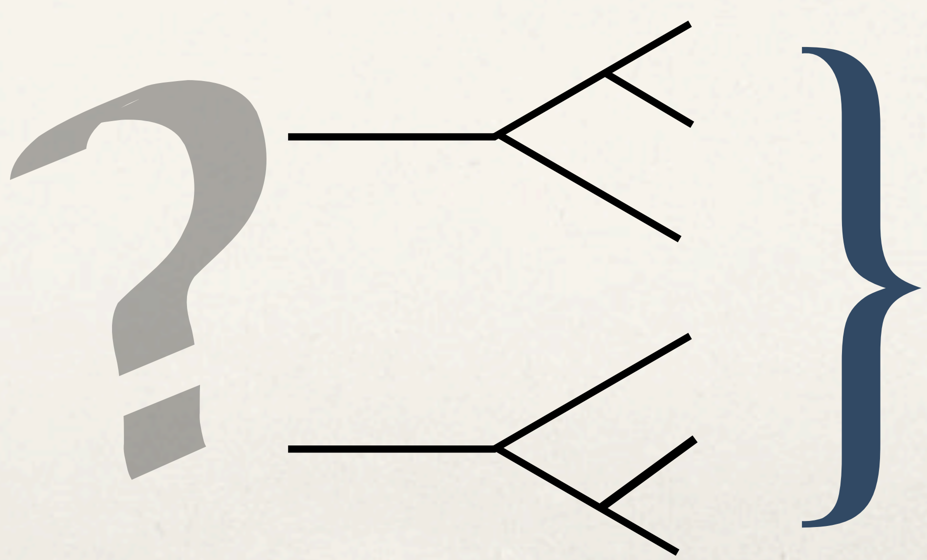
1. Consider only the two-dimensional distribution of energy in a jet
   - Examples: Trimming & Filtering, N-Subjettiness, Jet substructure w/o trees
2. Try to associate a tree structure with a jet
   - Allows one to use heuristic pictures of parton shower & decay chains.
   - Examples: Pruning, energy sharing variables, mass drop
   - However, the current procedure for constructing a tree is not ideal.

Mapping Jets to Trees

The energy distribution for a particular tree is unambiguous



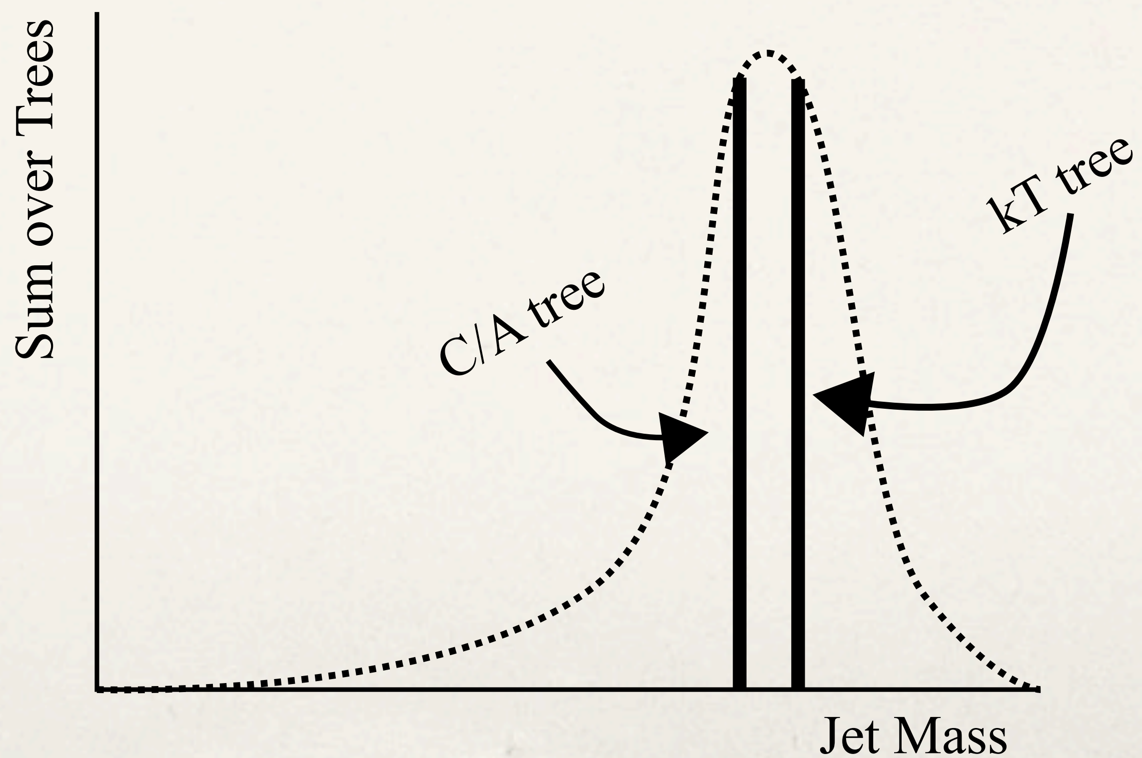




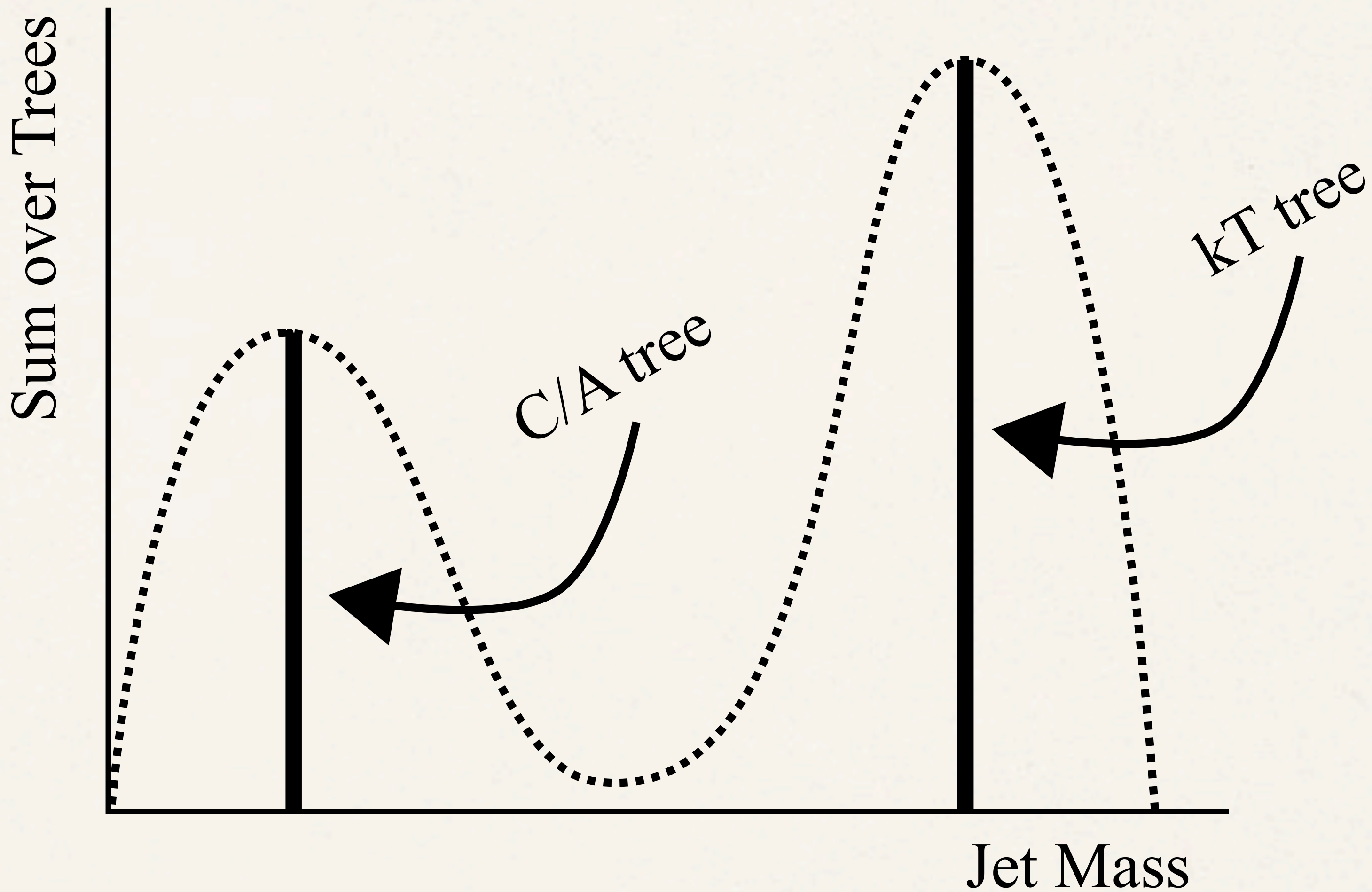
But, more than one tree can correspond to the same energy distribution

Unnecessary Choices

- How do we assign a particular tree to an energy distribution?
- Standard answer: Use a well motivated algorithm like C/A or kT
- Ideally, since both are well motivated algorithms they'll give the same answer:



- However, sometimes the answers are very different.

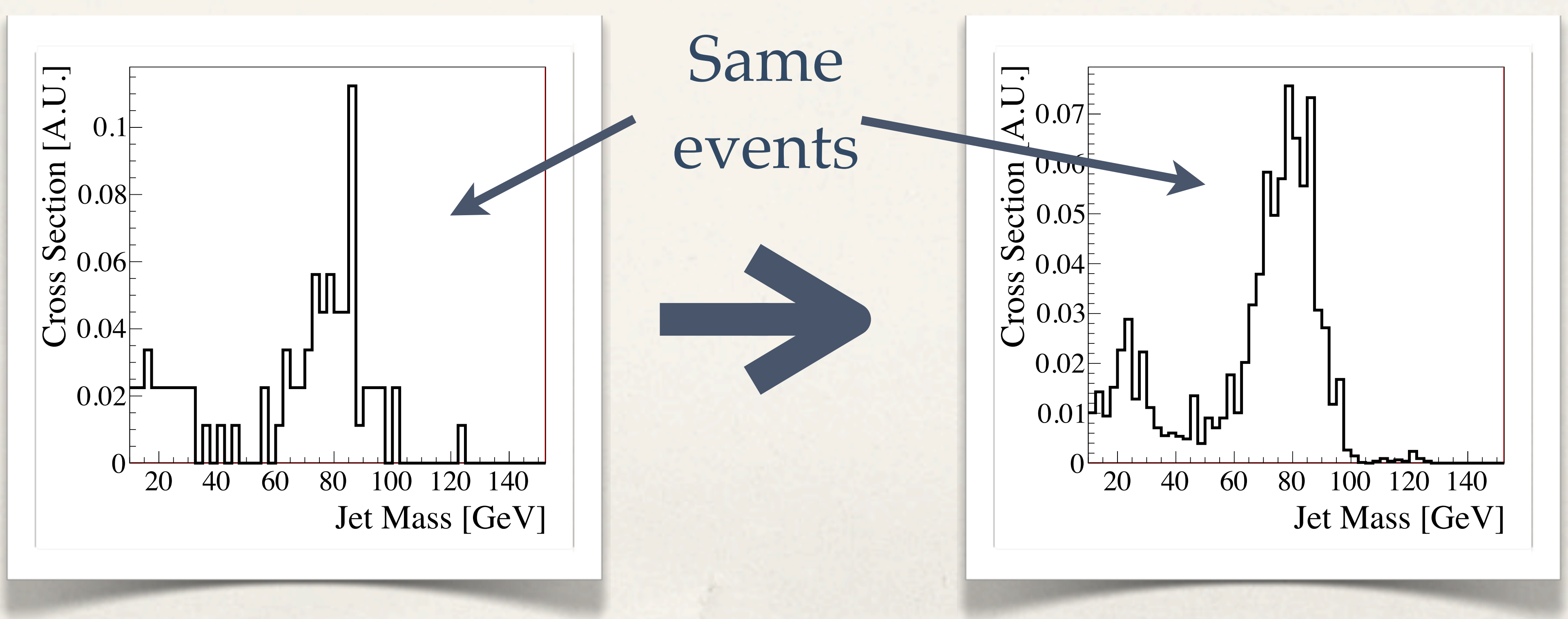


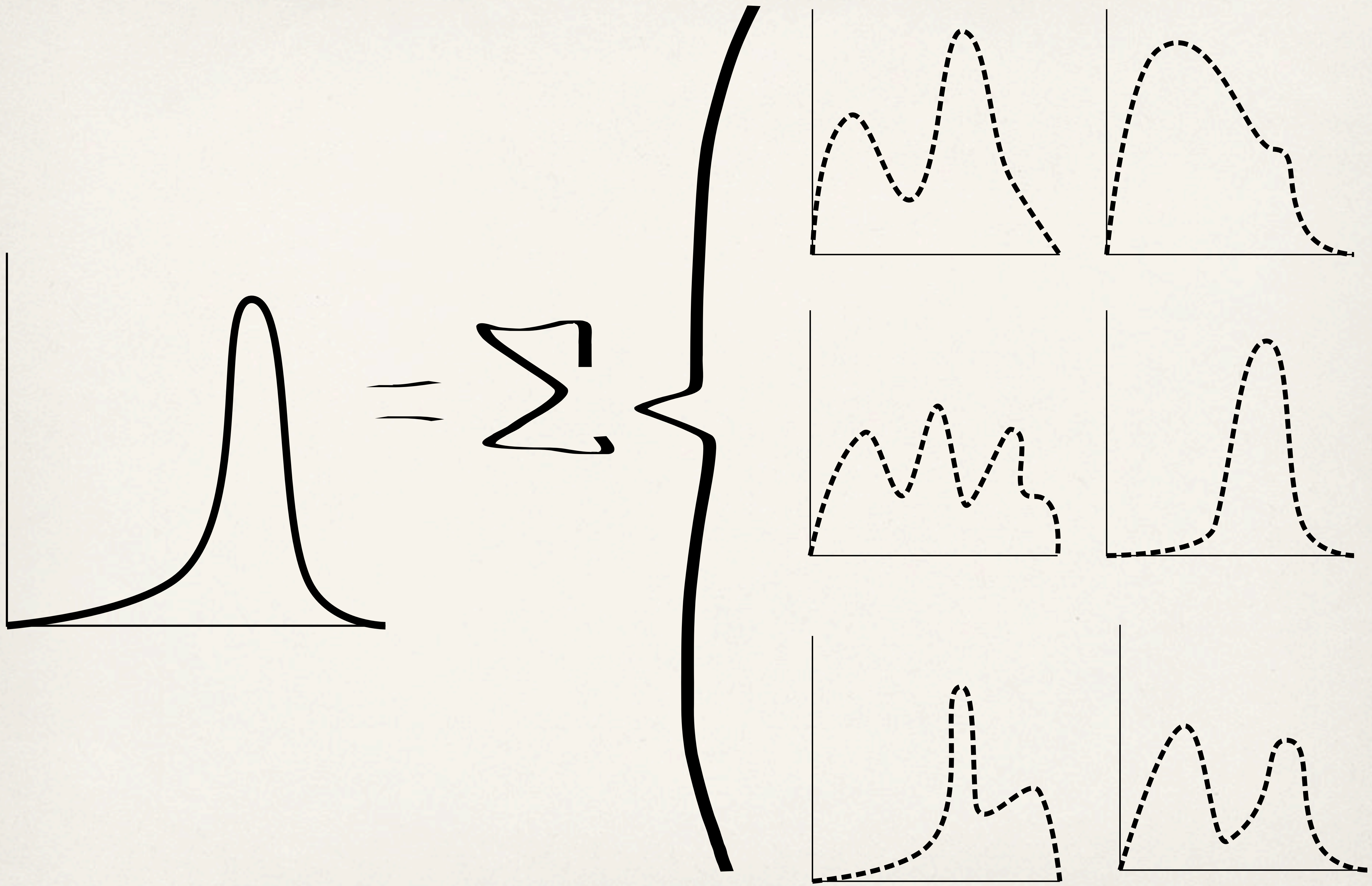
- Considering only the kT or C/A tree introduces an element of randomness into this process, resulting in unnecessary fluctuations in the final state observable.

- Intuitively it makes sense that defining an observable in a way which reflects the ambiguity of this clustering should yield better results.

Solution: Sum over Trees

- We propose that rather than assigning a single number to each event, instead each event should contribute a distribution obtained by summing the observable over many trees.
- When we sum these together, the result is much more stable than the histogram we would have had if we just considered one number per event.





Weights

- The only question is: when we add together the result obtained from different trees, how should we weight each tree's contribution?
- Surely they should not all count equally. If they did, then why would we use kT or C/A to find our trees in the first place?
- In theory, one could weight each tree by the product of splitting functions and Sudakovs one would obtain from a parton shower.
  - Work in progress.

Implementation

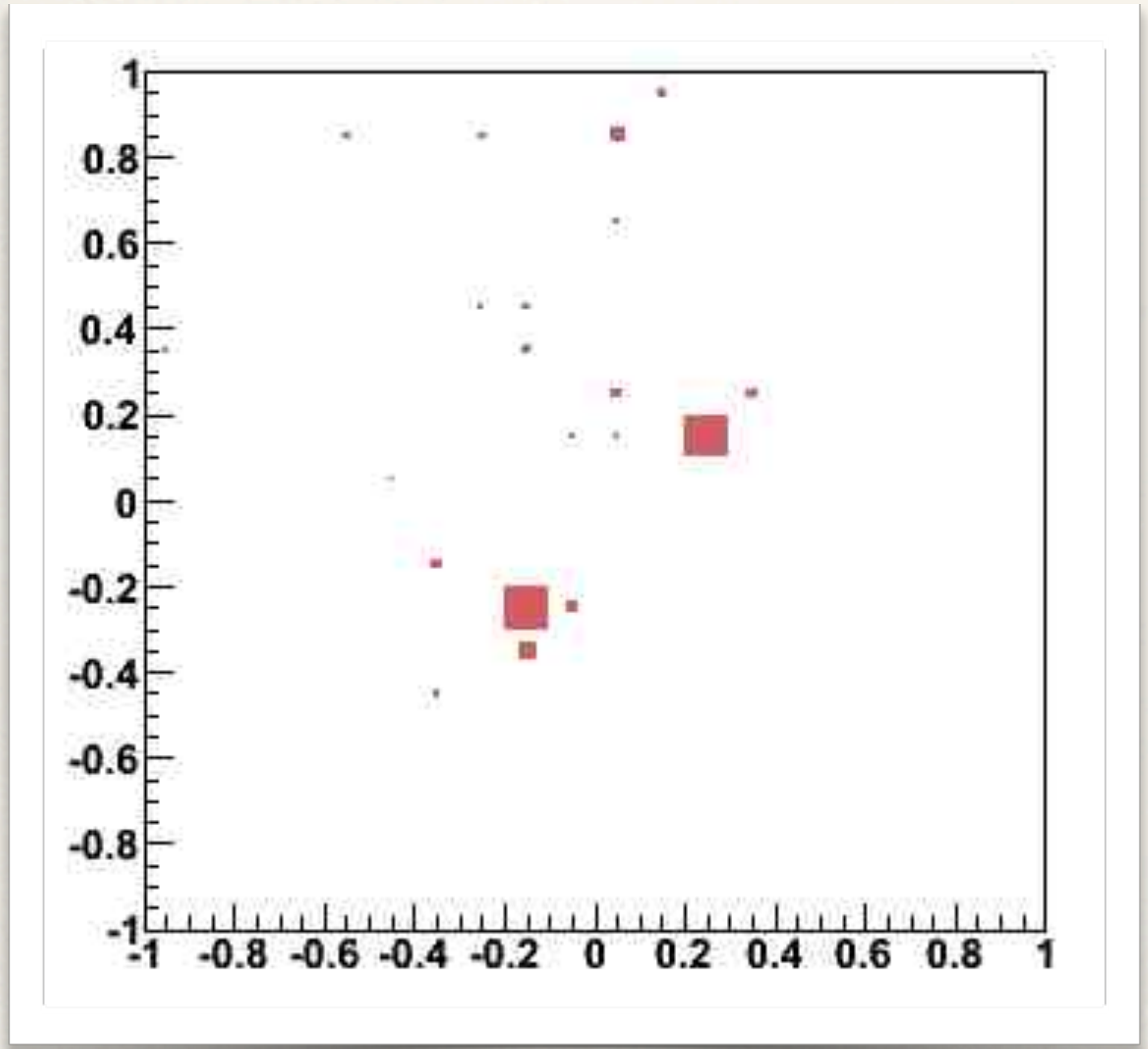
- Instead, we find a simpler Monte-Carlo procedure works quite well.
  - As in a sequential recombination algorithm, assign every pair of proto-jets a distance measure d_{ij}.
  - However, unlike a normal sequential algorithm (where the pair with the smallest measure is selected clustered), here we suggest that a given pair be randomly selected for merging with probability

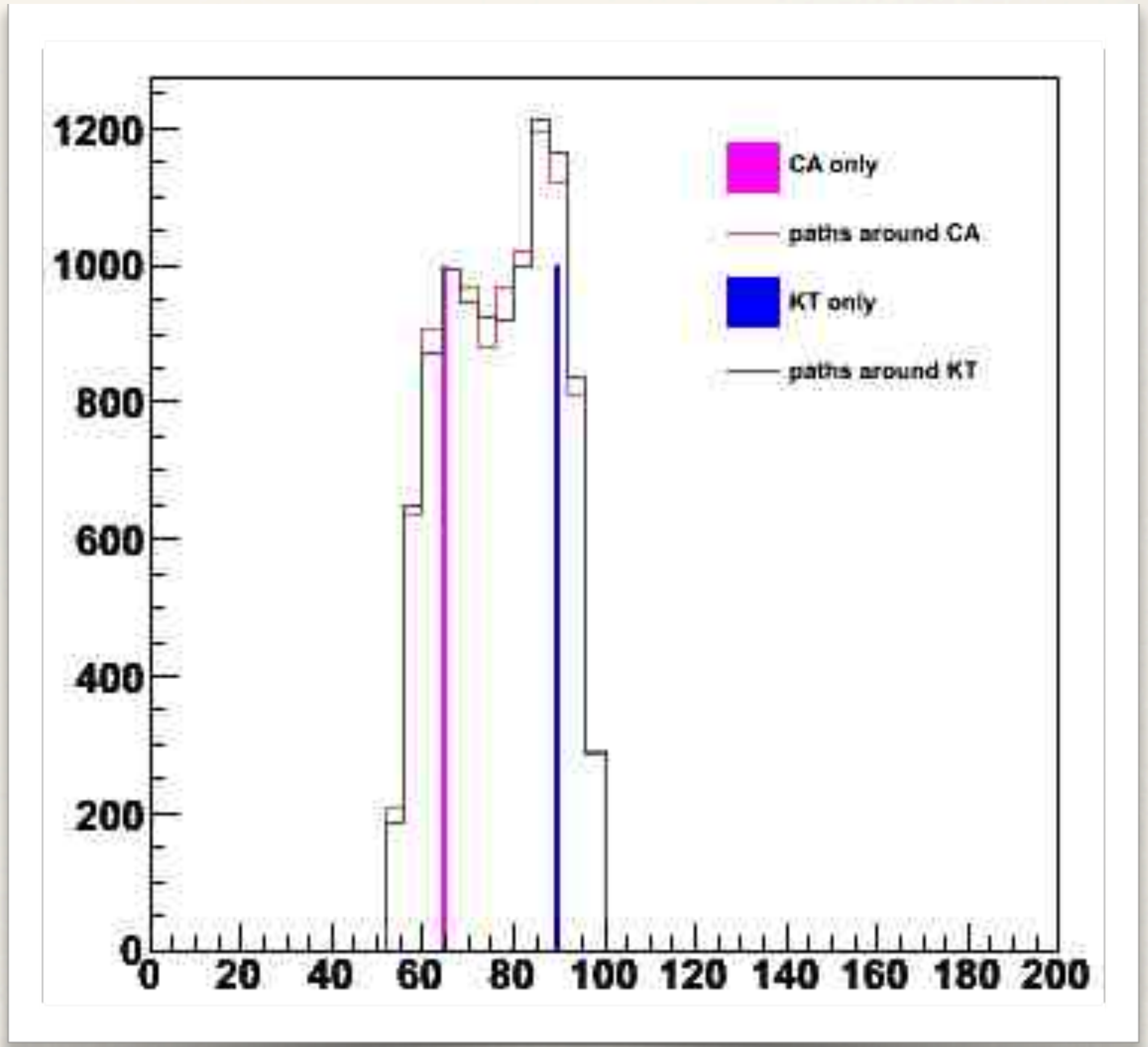
$$\Omega_{ij} \equiv \frac{1}{\Omega} \exp\left(-\alpha \frac{d_{ij}}{d_{ij}^{\min}}\right), \ \alpha = \text{rigidity parameter}$$

  - Thus, paths which deviate from the CA or kT behavior are less likely to occur
  - Repeat many (~100) times, till the distribution stabilizes

- The result is that you get many trees
- The probability of finding a given tree decreases as it becomes less k_T or C/A like

Example: Boosted W-Jets with Pruning



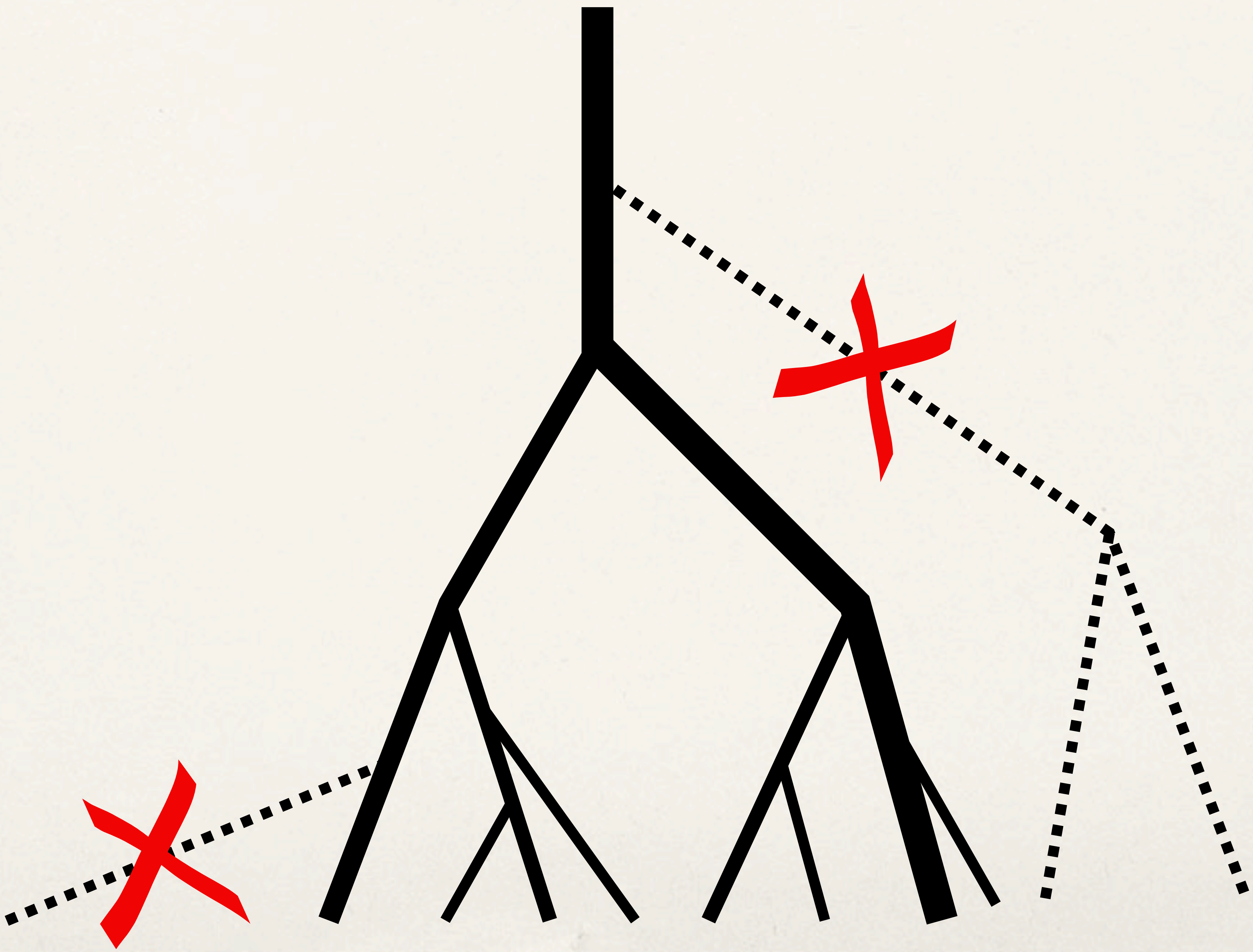


Pruning

- Pruning was introduced to look for boosted heavy objects (e.g., tops, higgses, W's, etc) by cleaning up their mass.
- Intuition: QCD has soft/collinear singularities. Wide-angle emissions should come from hard decays.
  - Remove all parts of the jet which are *both* soft and wide angle.
- Two main advantages:
  - Boosted objects see their mass reconstruction improved
  - Massive QCD jets (a large background) see their mass substantially decreased -> lower backgrounds

Pruning (Ellis, Vermilion, Walsh - 0903.5081, 0912.0033)

A Pruned Tree



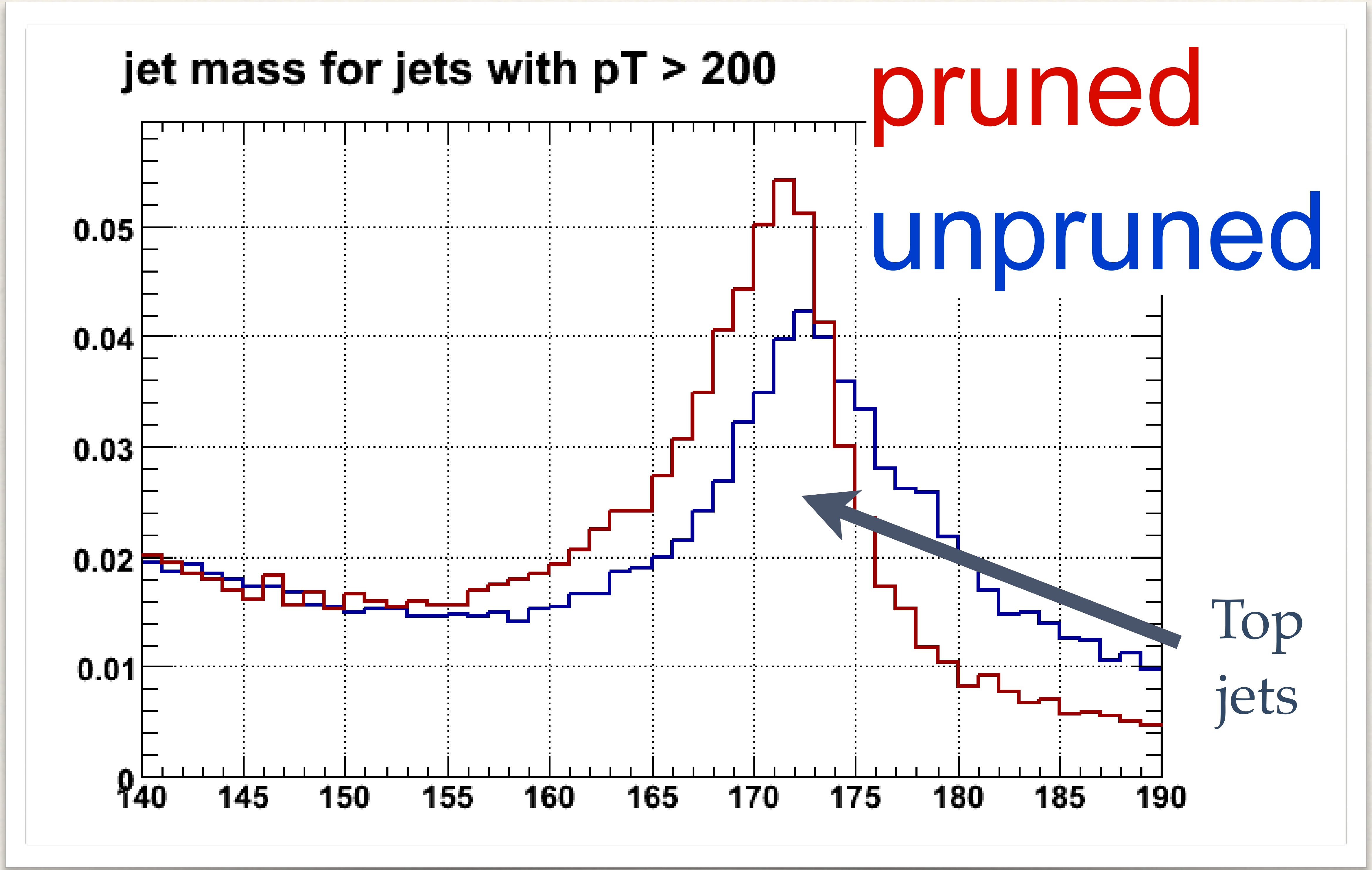


Figure source: http://www.phys.washington.edu/users/ellis/USATLAS.pdf

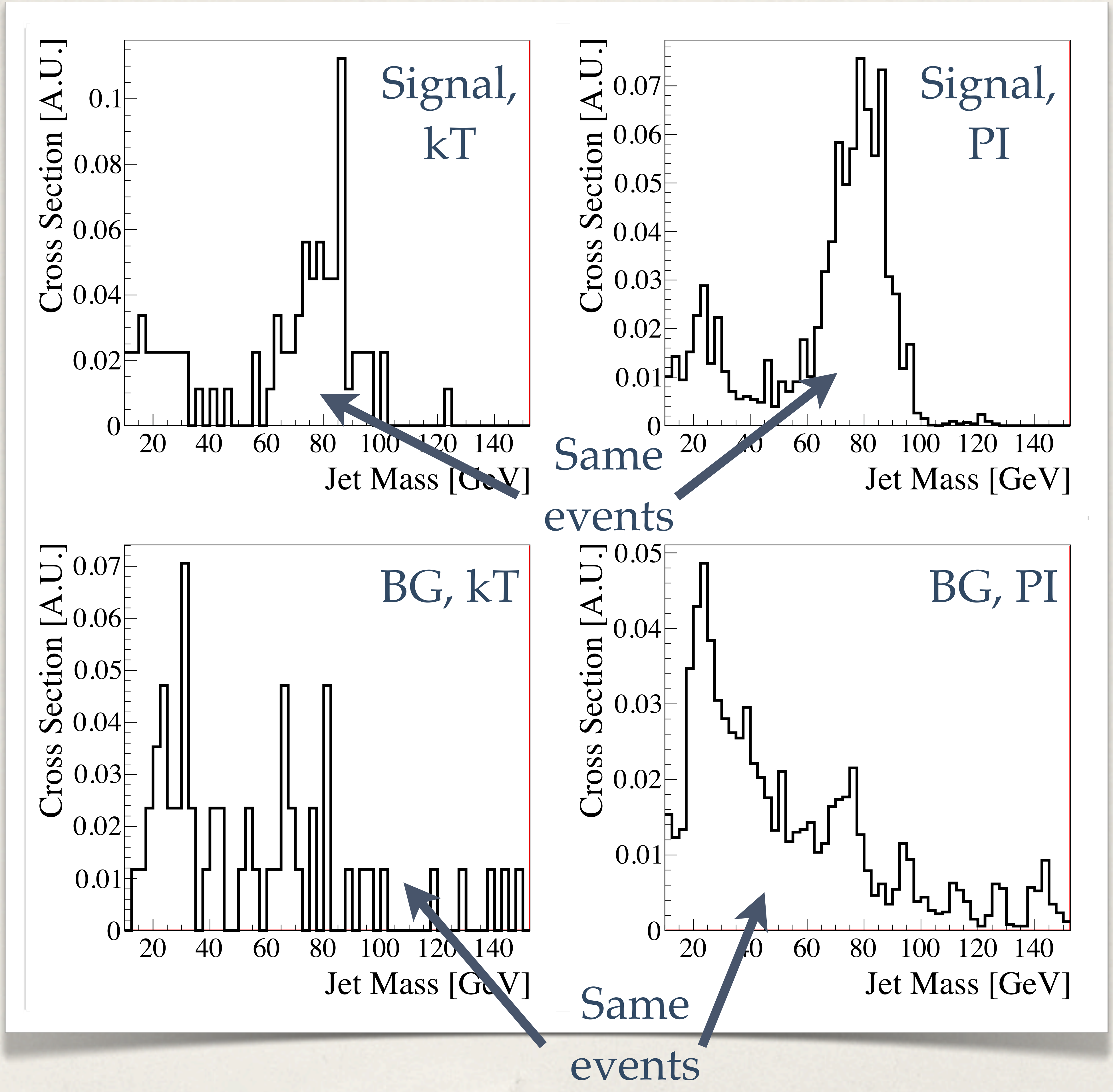
- Let’s see what happens when we modify pruning so that it runs over trees generated via the Qjet procedure.

Signal Discovery & Exclusion

- Signal = boosted W-jets, pT > 500
- BG = light QCD jets, pT > 500
- Measure the signal size in a bin (here 70-90 GeV) and compare it to the size of the BG fluctuations (Poisson stats included)
- Need only ~70% the luminosity to have the same significance

$$S/\delta B \propto \sqrt{N}$$

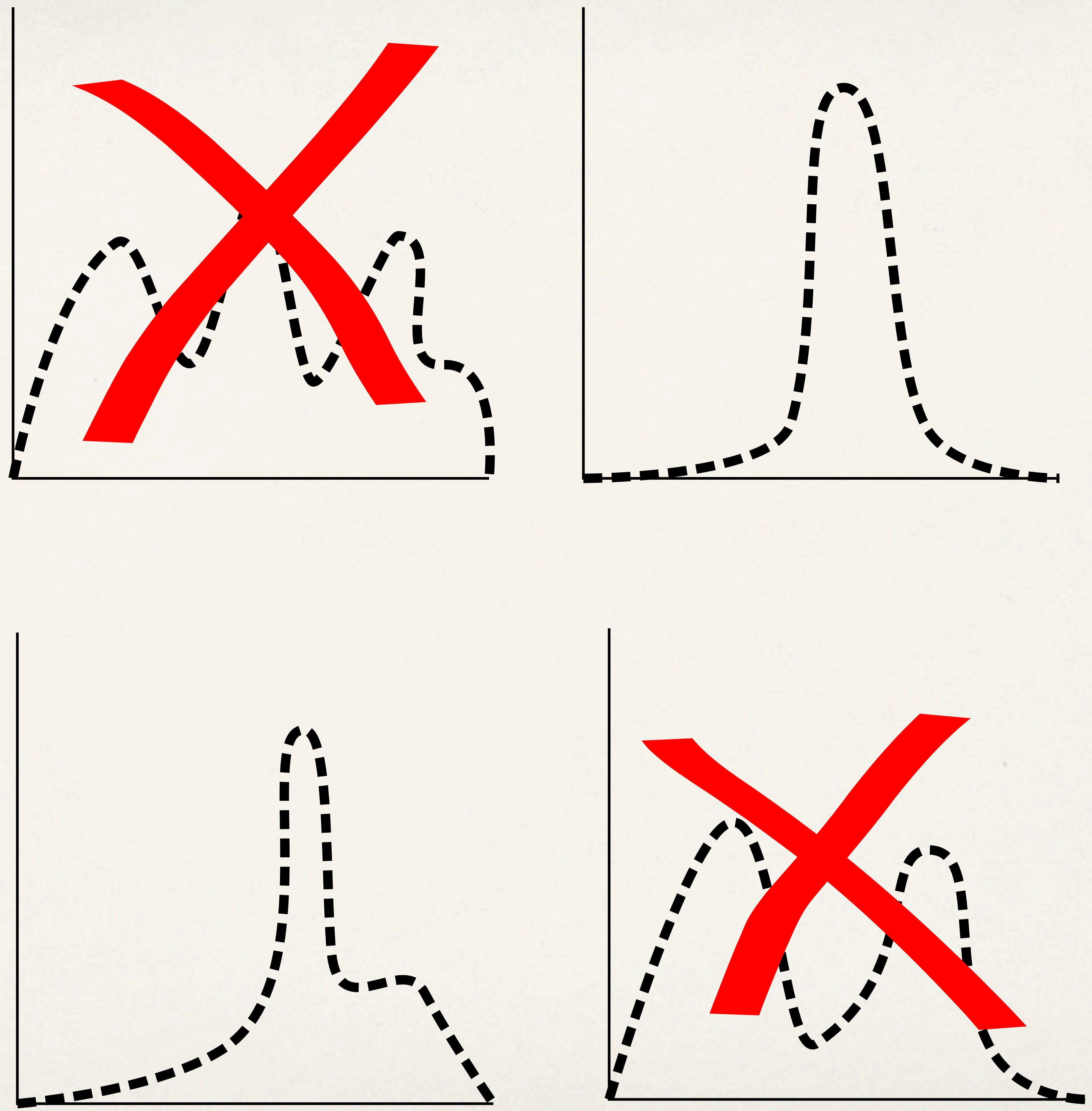
Algorithm	S/delta(B)	Relative Lumi Required
kT	4.9	1.00
Flat (α =0)	6.0	0.83
Qjets (α =10^{-1})	6.3	0.69



Signal, kT
Signal, PI
BG, kT
BG, PI
Cross Section [A.U.]
Jet Mass [GeV]
Same events
Same events

Signal vs. Background Discriminant

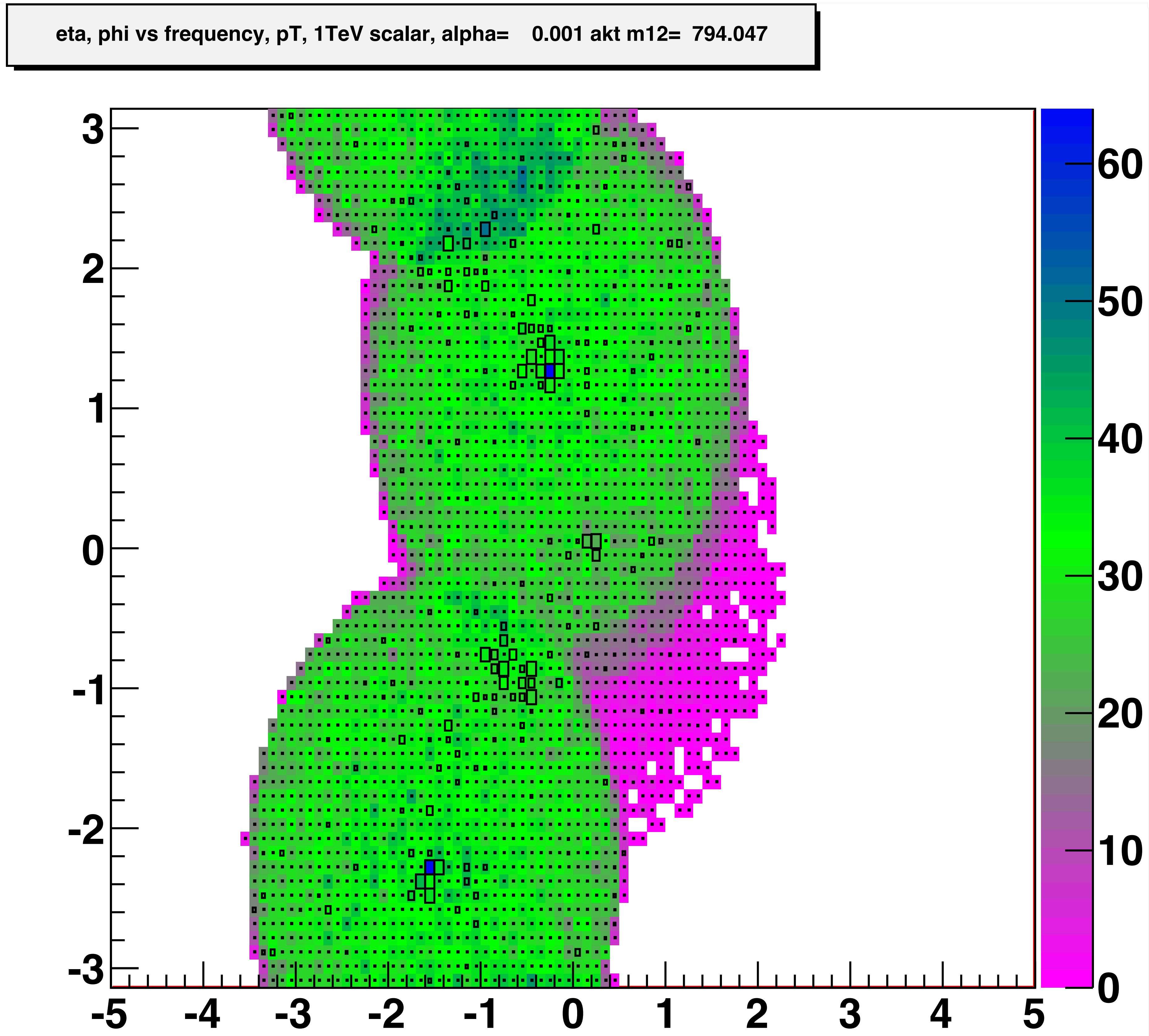
- When there's a "right answer" for a jet's mass, most of the trees tend to center around that value.
  - There's a "right answer" for the pruned mass of a boosted particle's jet, but not for a background QCD jet
- The width of a mass distribution serves as a good signal to background discriminant!



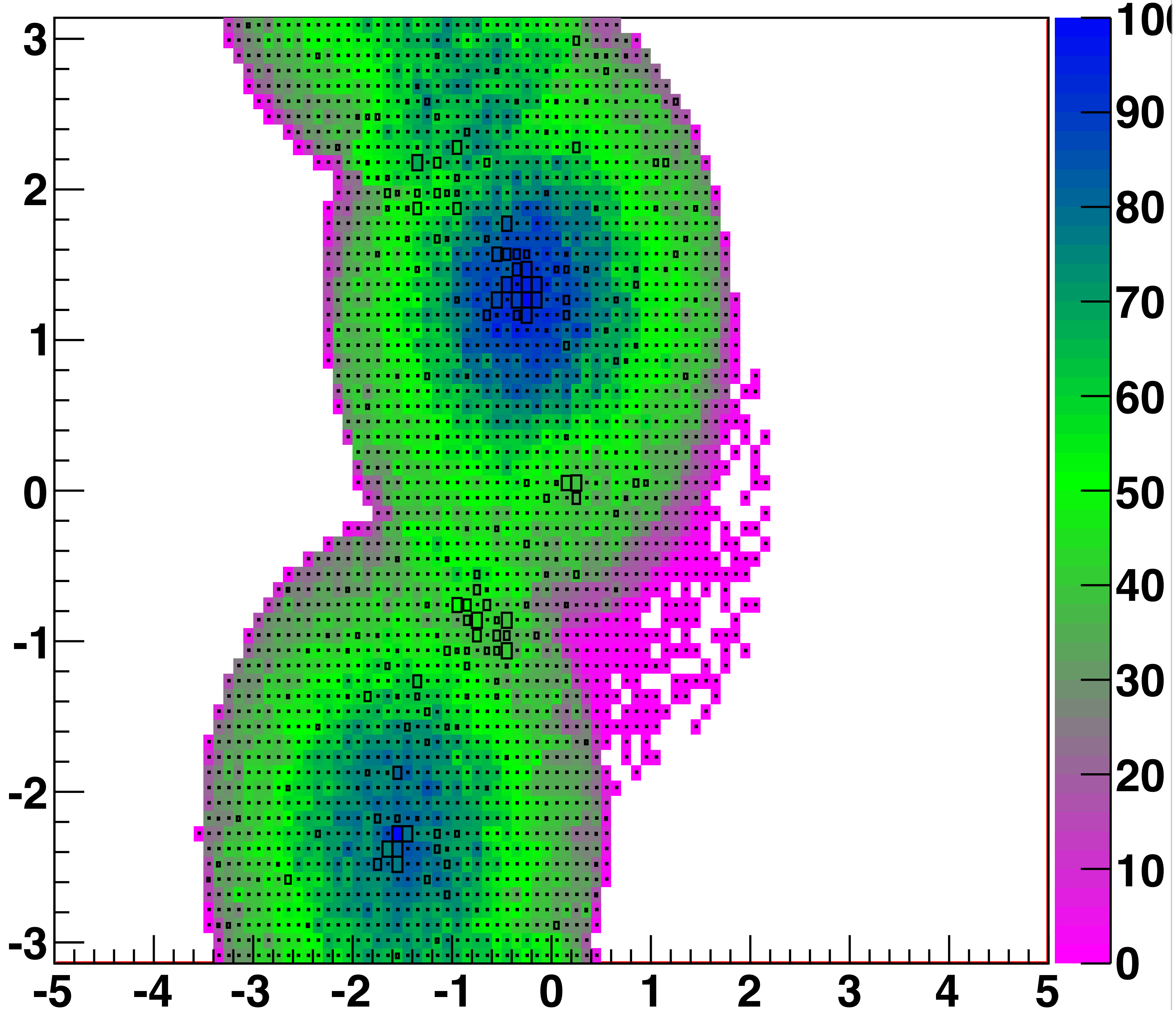
Generalize to a Jet Algorithm

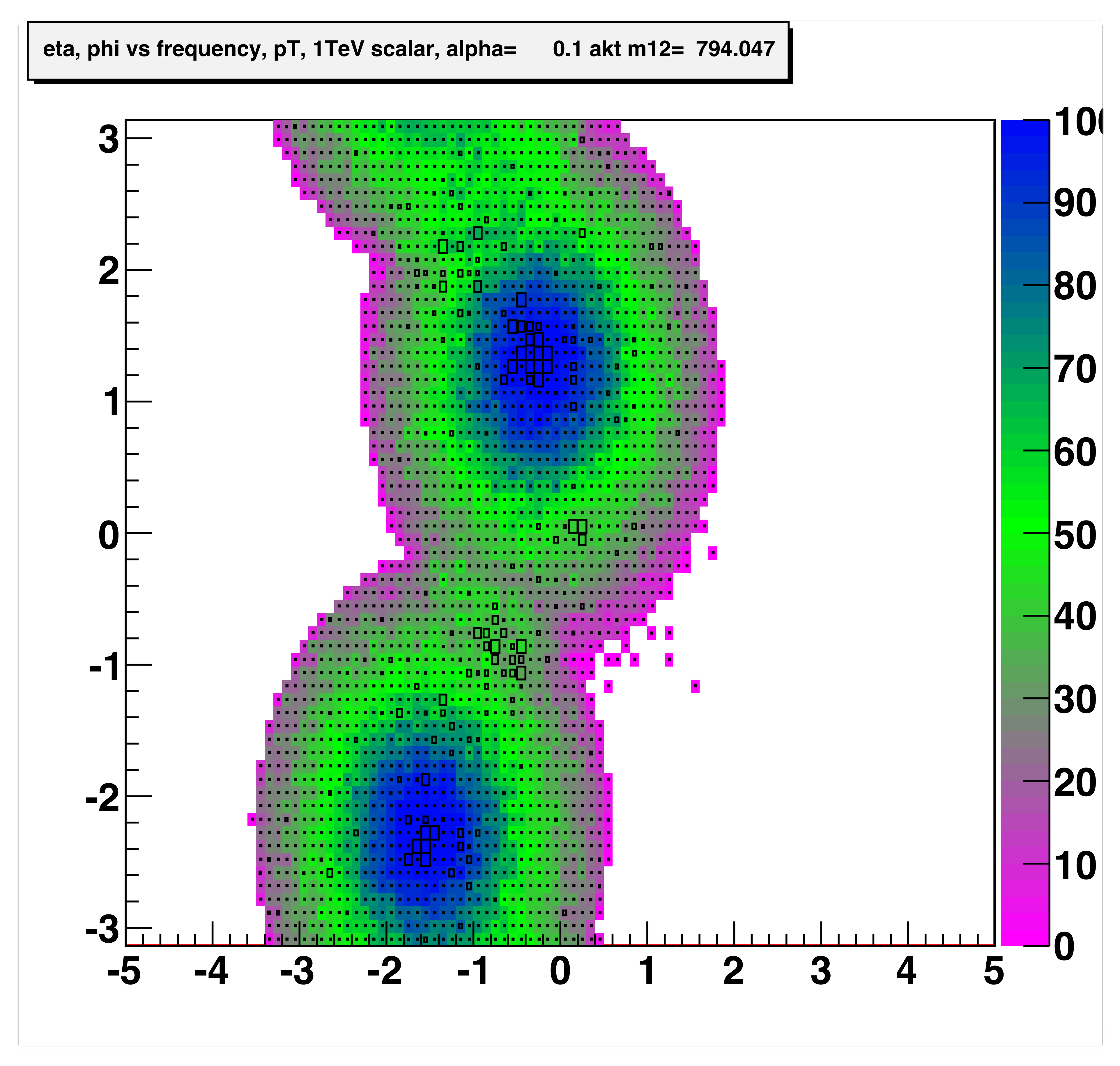
Qanti-kT

- Work in progress
- Take anti-kT and perturb around it as with Qjets
- Final state is now different
  - Different jet four-momenta
  - Different jet multiplicities

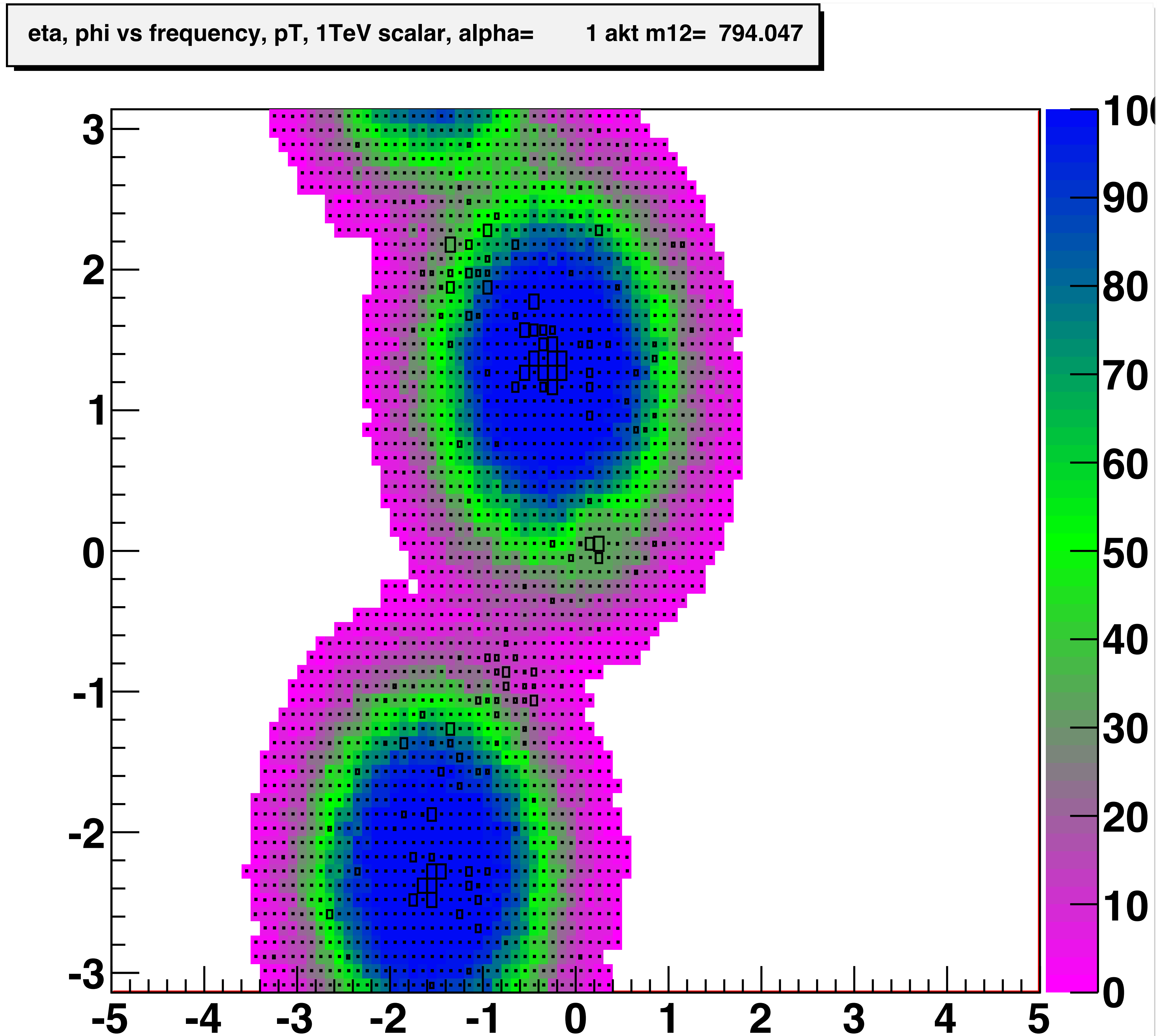

eta, phi vs frequency, pT, 1TeV scalar, alpha= 0.001 akt m12= 794.047
3
2
1
0
-1
-2
-3
-5
-4
-3
-2
-1
0
1
2
3
4
5
60
50
40
30
20
10
0

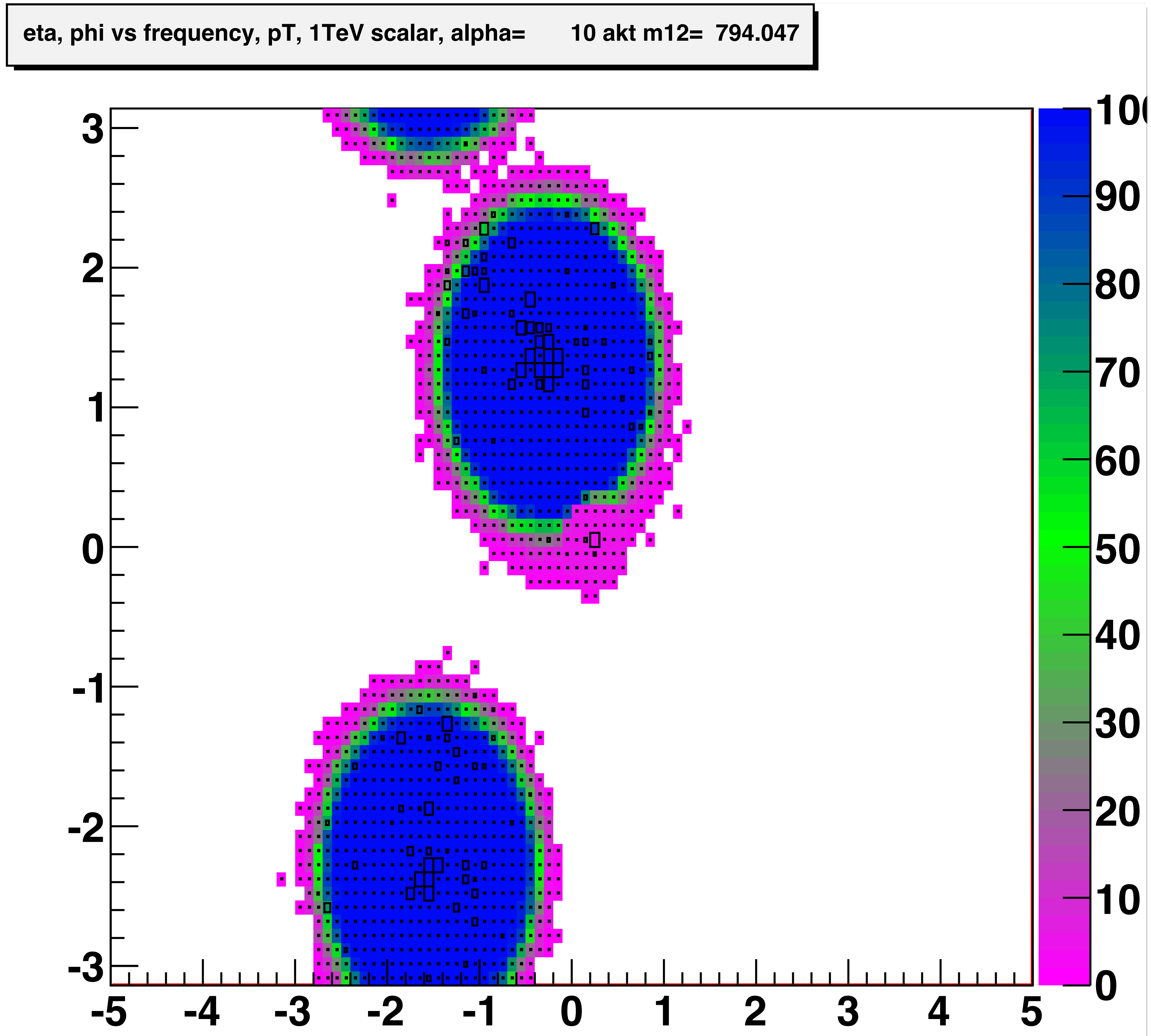
eta, phi vs frequency, pT, 1TeV scalar, alpha= 0.01 akt m12= 794.047

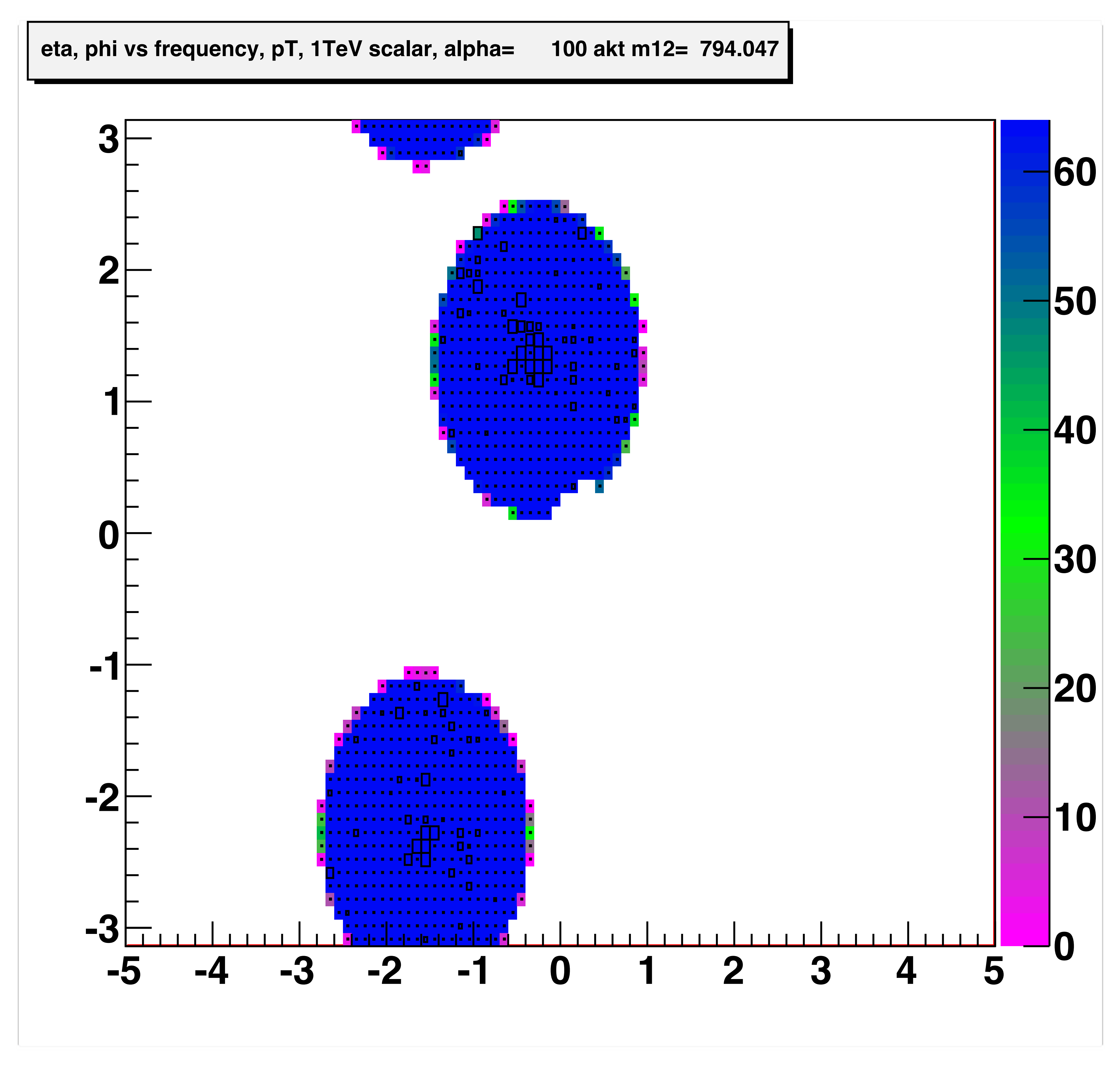




eta, phi vs frequency, pT, 1TeV scalar, alpha= 0.1 akt m12= 794.047
3
2
1
0
-1
-2
-3
-5
-4
-3
-2
-1
0
1
2
3
4
5
90
80
70
60
50
40
30
20
10
0


eta, phi vs frequency, pT, 1TeV scalar, alpha= 1 akt m12= 794.047
3
2
1
0
-1
-2
-3
-5
-4
-3
-2
-1
0
1
2
3
4
5
0
10
20
30
40
50
60
70
80
90


eta, phi vs frequency, pT, 1TeV scalar, alpha= 10 akt m12= 794.047
3
2
1
0
-1
-2
-3
-5
-4
-3
-2
-1
0
1
2
3
4
5
0
10
20
30
40
50
60
70
80
90



eta, phi vs frequency, pT, 1TeV scalar, alpha= 100 akt m12= 794.047
3
2
1
0
-1
-2
-3
-5
-4
-3
-2
-1
0
1
2
3
4
5
60
50
40
30
20
10
0

Conclusion

- When we use C/A or k_T to associate a tree with a jet this is really just our "best guess" for the showering history.
- Sometimes these two algorithms return very different answers for the event at hand.
  - By choosing, e.g. the k_T answer over the C/A one, we introduce randomness into the picture, and the statistics are degraded.
- We propose that all trees be considered, each with a set weight, and a distribution obtained for each event (rather than a single number).
  - The results obtained from this are much less susceptible to unwanted fluctuations: equivalent to a ~2x increase in luminosity.