Kieran Maguire[(1)], *Henry Day-Hall*[(2)], Srinandan Dasmahapatra[(1)], Stefano Moretti[(1)]

(1) University of Southampton, (2) Czech Technical University in Prague

10th November 2023

Introduction

Principles of jets

A good jet clustering algorithm;

1. Reveals the kinematics of the hard scattering.
2. IR and collinear safe.
3. Simple to compute for a theory.
4. Fast to compute in practice.

Universally adopted Anti-k_{T} algorithm meets all these criteria. Difficult to improve on that.

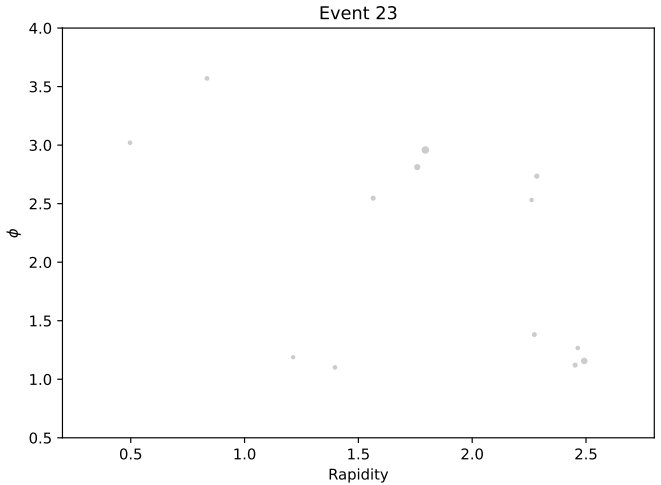
Anti-k_{T} is a greedy algorithm; it makes the optimum move at the current step, but cannot consider all possible end points.

Could we imagine a non-greedy algorithm?

The NCut objective

A good cluster

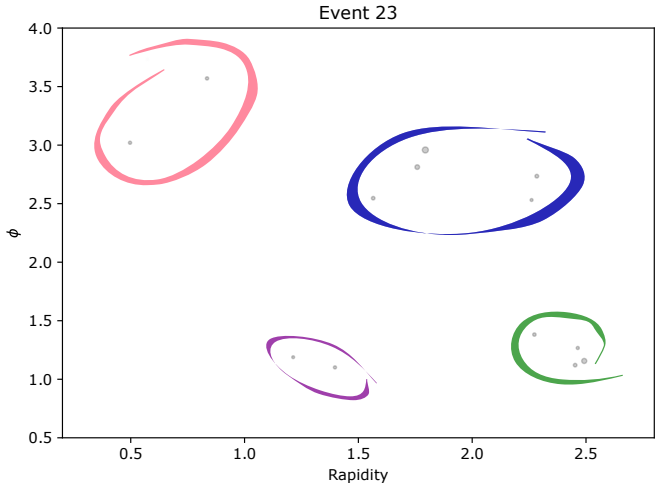
What makes something a "cluster" or jet? Consider a very small event;



The NCut objective

by eye

Small enough to visualise easily. We can guess what should go into which jet.



The NCut objective

Affinity

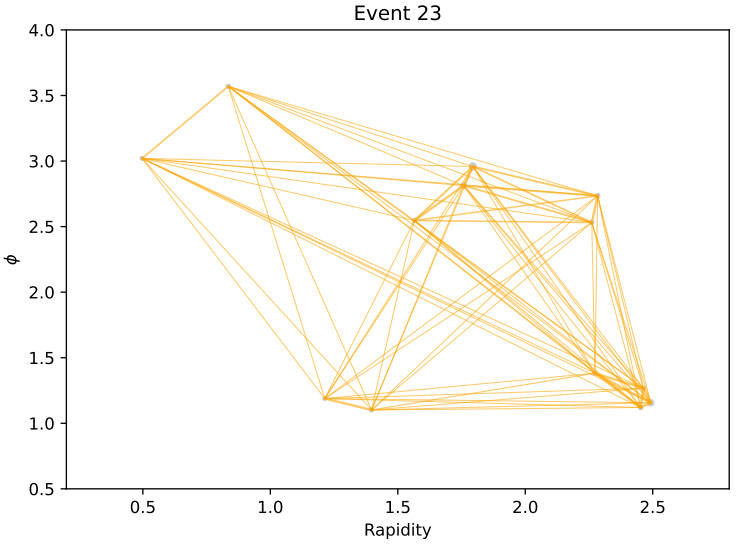
We need to express our objective algorithmically.

Affinity is a measure of how likely two particles are to be in the same jet.

What if it was;

$$A_{ij} = \mathrm{e}^{-d_{ij}^2/2\sigma^2}$$

(d_{ij} is the Cambridge-Aachen distance between particles i and j)



The NCut objective

Weight

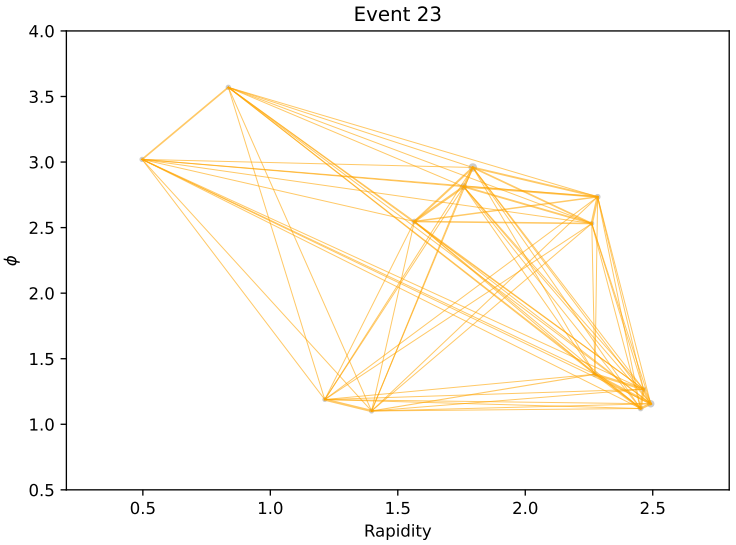
Let all particles be exclusively grouped in jets, J. Simply minimising affinity between jets;

$$\min_J \sum_{K \in J} \sum_{i \in K, j \notin K} A_{ij}$$

creates a problem, optimal solutions tend to isolate single particles.

Balance the jets by assigning each particle a weight, w_i.

$$\min_J \sum_{K \in J} \frac{\sum_{i \in K, j \notin K} A_{ij}}{\sum_{i \in K} w_i}$$



The NCut objective

Degree as weight

The degree of each point is a common choice for weight.

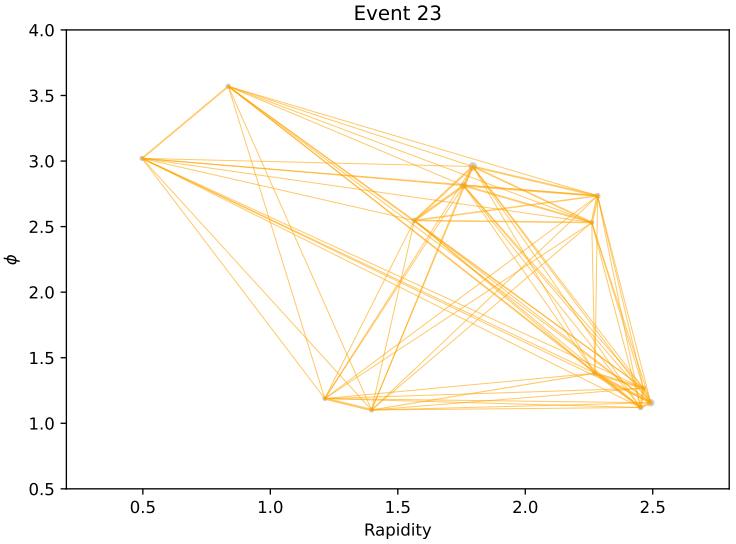
$$w_i = \sum_j A_{ij}$$

Particles contribute to a cluster by home much they are connected to other particles.

The minimisation problem becomes;

$$\min_J \sum_{K \in J} \frac{\sum_{i \in K, j \notin K} \mathrm{e}^{-d_{ij}^2/2\sigma^2}}{\sum_{i \in K} \sum_j \mathrm{e}^{-d_{ij}^2/2\sigma^2}}$$

Prohibitive expensive to compute.



Relaxation to obtain a solution

Spectral clustering

Let us form a graph Laplacian.

Let $Z_{i,j} = \delta_{i,j} w_i$ and $D_{i,j} = \delta_{i,j} \sum_a A_{i,a}$, then our Laplacian is;

$$L = Z^{-1/2}(D - A)Z^{-1/2}.$$

- Each eigenvector of L has as many elements as there are particles.
- Perfect case; affinity between jets is zero.
- In this case, the eigenvectors with highest eigenvalue are piecewise-constant.
- Particle groups are denoted by their value in the eigenvectors.

If we apply this solution to real (imperfect) cases, it is a relaxation

There is an elegant proof of this, too long for this talk (see backup slides).

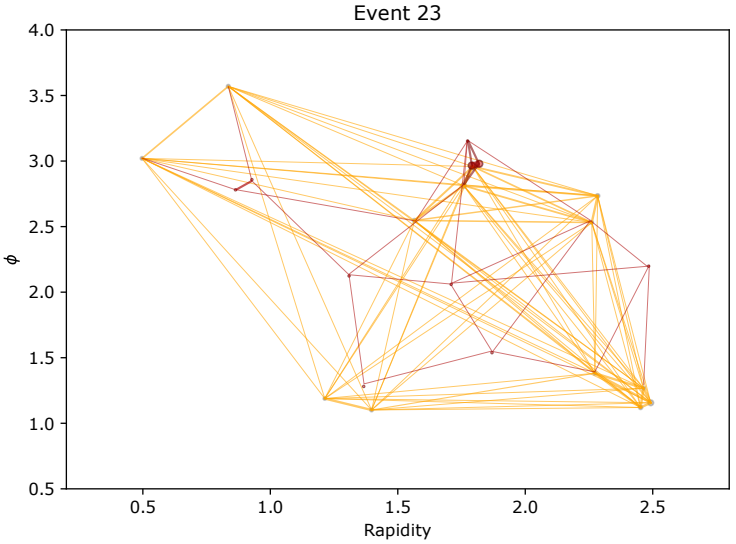
IR and collinear safety

Problem

$$\min_J \sum_{K \in J} \frac{\sum_{i \in K, j \notin K} \mathrm{e}^{-d_{ij}^2/2\sigma^2}}{\sum_{i \in K} \sum_j \mathrm{e}^{-d_{ij}^2/2\sigma^2}}$$

This isn't IRC safe.

- Collinear splitting in a jet will add new connections between jets, and massively modify the weight.
- Soft emissions will be just as impactful as every other particle, modifying connections and weights.

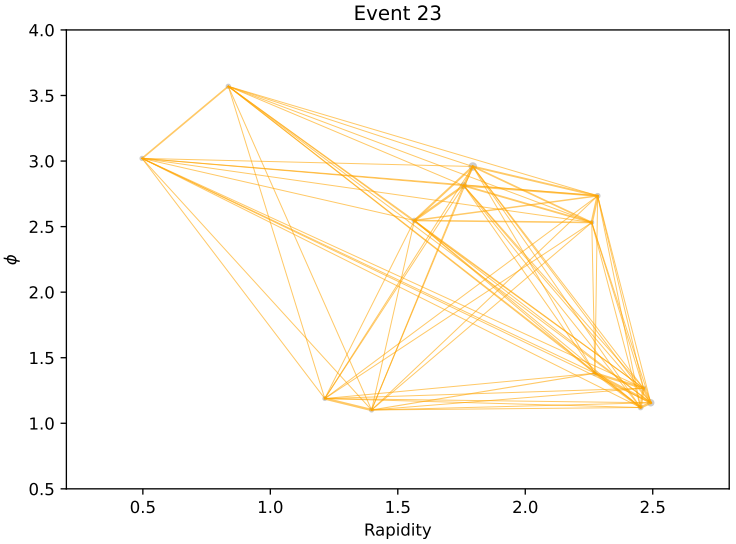


IR and collinear safety

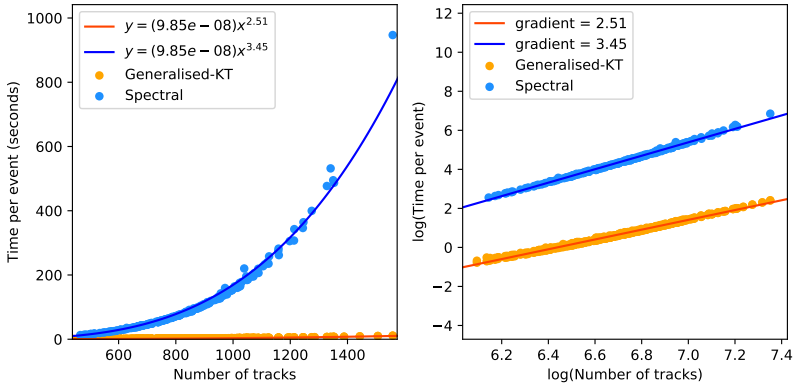
An ugly solution

Return to a greedy agglomerative algorithm?

1. Modify the weight, to scale with degree at larger p_{T}, and with p_{T} at smaller p_{T}.
2. Use the relaxed NCut objective to get an alternative distance metric.
3. Modify this distance so that at low angular separation it goes to zero.
4. Merge the closest pair.
5. Repeat.



Ugly computational complexity



Measurements of the runtime would indicate that this is approximately $O(N^3)$.
That's actually optimistic.

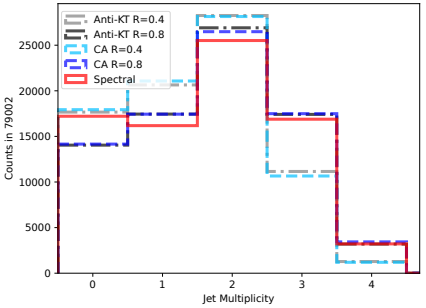
- The most expensive element is the eigenvalue calculation, in theory $O(N^3)$.
- But this method repeats the eigenvalue calculation up to N times.
- So run time could be as bad as $O(N^4)$.

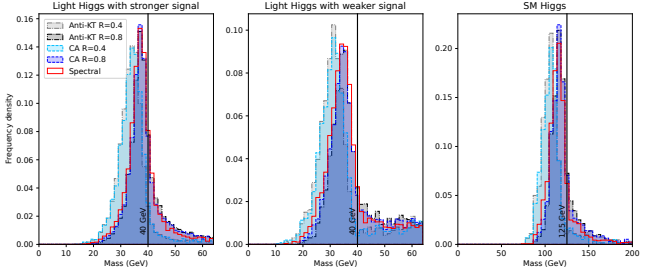
This is not a tractable in realistic HEP applications.

Ugly solution; good results

$H_{125\mathrm{GeV}} \to h_{40\mathrm{GeV}} h_{40\mathrm{GeV}} \to b\bar{b}b\bar{b}$

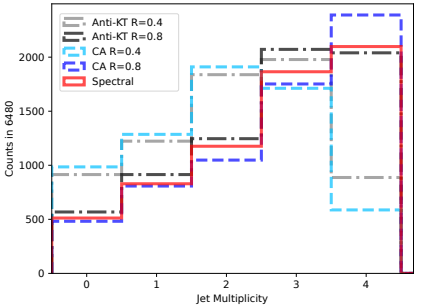
Light Higgs

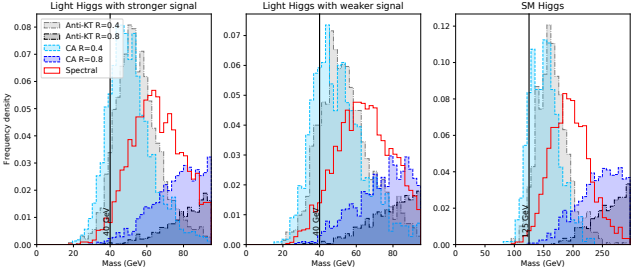




With pileup

Light Higgs





Computational complexity

Chebyshev approximation

There is a second trick that we can use Chebyshev approximation of the eigenvectors, developed by arXiv:0912.3848.
Very roughly;

- A matrix multiplied onto a vector can only return the same vector if that vector is an eigenvector.
- We can approximate the eigenvectors by repeatedly applying the matrix to a random vector, it must converge to an eigenvector. (This is the QR algorithm.)
- Eigenvectors must be orthonormal.
- In a localised area, an orthonormal basis can be approximated by a set of Chebyshev polynomials.
- This gives us access to subsequent eigenvectors.

This brings the eigenvector calculation down to $\mathcal{O}(N^2)$. If we could avoid the agglomerative step, this could create a $\mathcal{O}(N^2)$ clustering.
Which gives us the acronym Chebyshev Approximated Laplacian Eigenvectors.

IR and collinear safety

Affinities

Focus on the numerator;

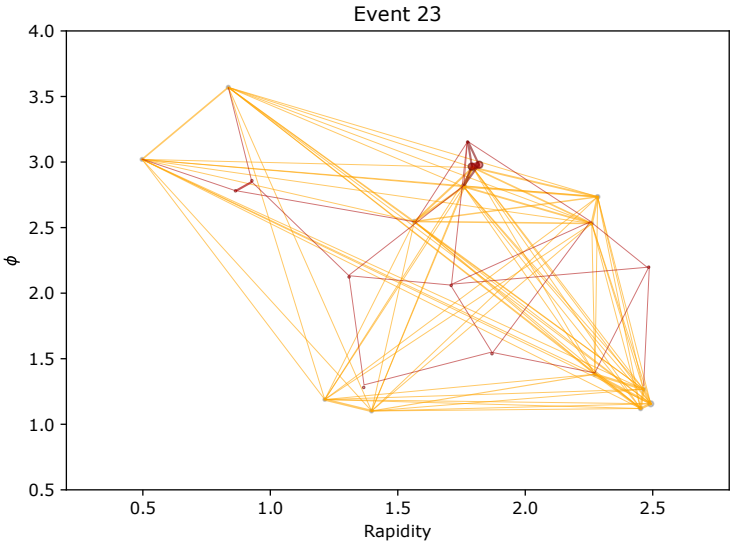
$$\min_J \sum_{K \in J} \frac{\sum_{i \in K, j \notin K} A_{ij}}{\sum_{i \in K} w_i}$$

For equality under the collinear splitting of $j \to a, b$; $A_{ij} = A_{ia} + A_{ib}$ for any i.

This is achieved by;

$$A_{ij} = p_{\mathrm{T}i} p_{\mathrm{T}j} \mathrm{e}^{-d_{ij}^2/2\sigma^2}$$

which also makes the affinities of any soft emission vanish.



IR and collinear safety

Weights

Then the denominator;

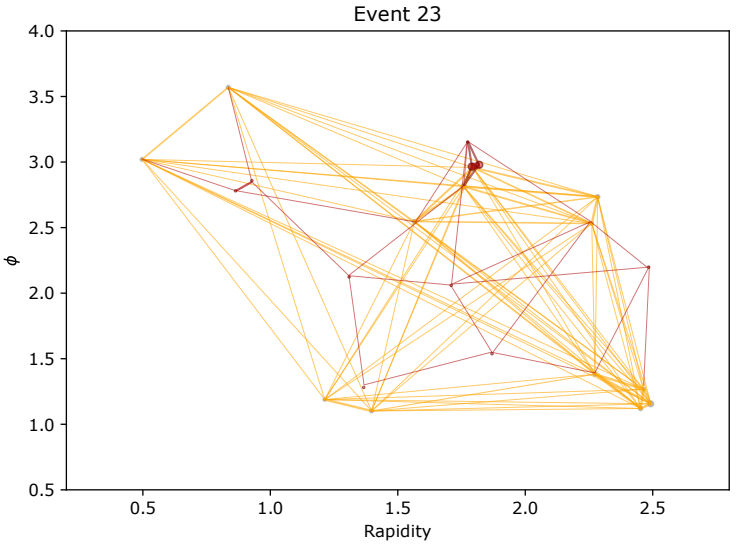
$$\min_J \sum_{K \in J} \frac{\sum_{i \in K, j \notin K} A_{ij}}{\sum_{i \in K} w_i}$$

For equality under the collinear splitting of $j \to a, b$; $w_j = w_a + w_b$.

This is achieved by;

$$w_j = p_{\mathrm{T}j}$$

which also makes the affinities of any soft emission vanish.



IR and collinear safety

Better solution

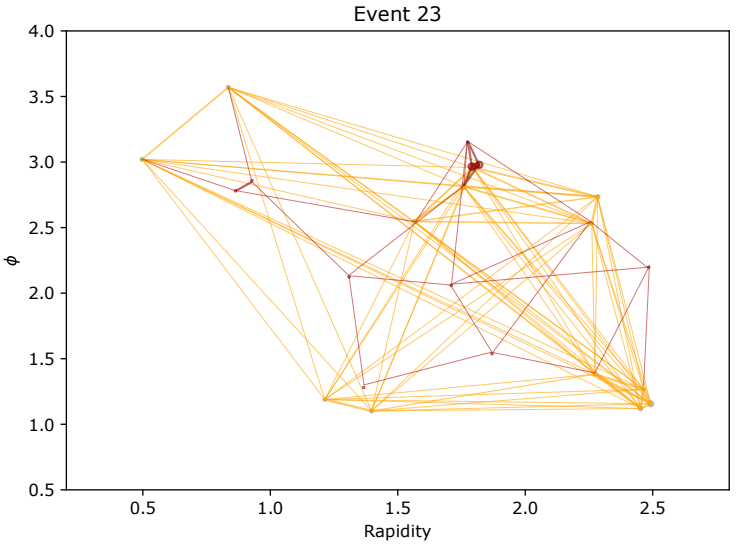
Our objective is

$$\min_J \sum_{K \in J} \frac{\sum_{i \in K, j \notin K} A_{ij}}{\sum_{i \in K} p_{\mathrm{T}\,i}}$$

with

$$A_{ij} = p_{\mathrm{T}\,i} p_{\mathrm{T}\,j} \mathrm{e}^{-d_{ij}^2/2\sigma^2}$$

This is IR and collinear safe.

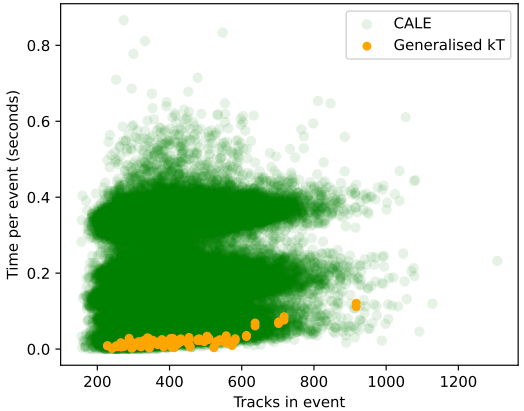


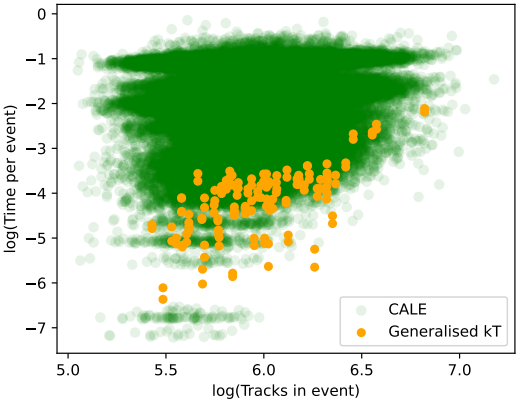
Computational complexity

Elegant solution

Timing now goes as $\mathcal{O}(N^2)$ for the whole jet formation.

- There is a significant setup overhead, a more careful implementation would be needed to determine if this could be avoided.
- This is in line with a naive implementation of the anti-k_{T} algorithm.
- Like the anti-k_{T} algorithm, this could be taken to $\mathcal{O}(N \log N)$ with appropriate localisation.

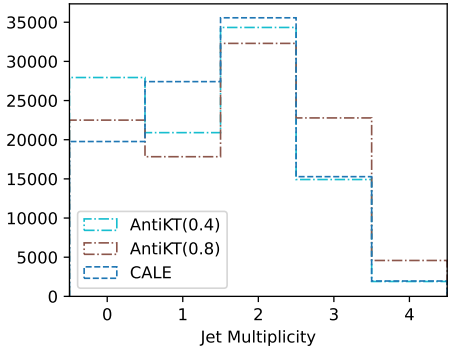


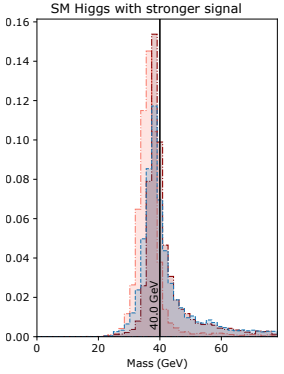


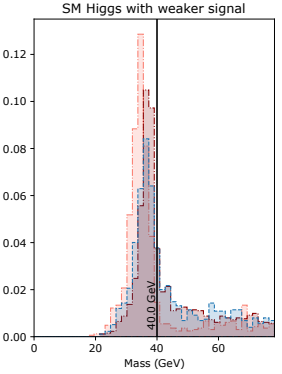
Current results

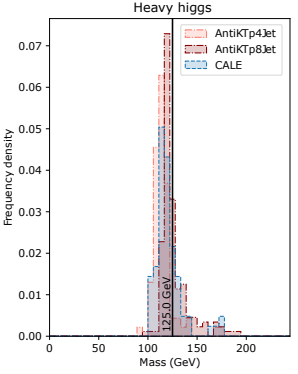
Requires work

$H_{125\mathrm{GeV}} \to h_{40\mathrm{GeV}} h_{40\mathrm{GeV}} \to b\bar{b}b\bar{b}$









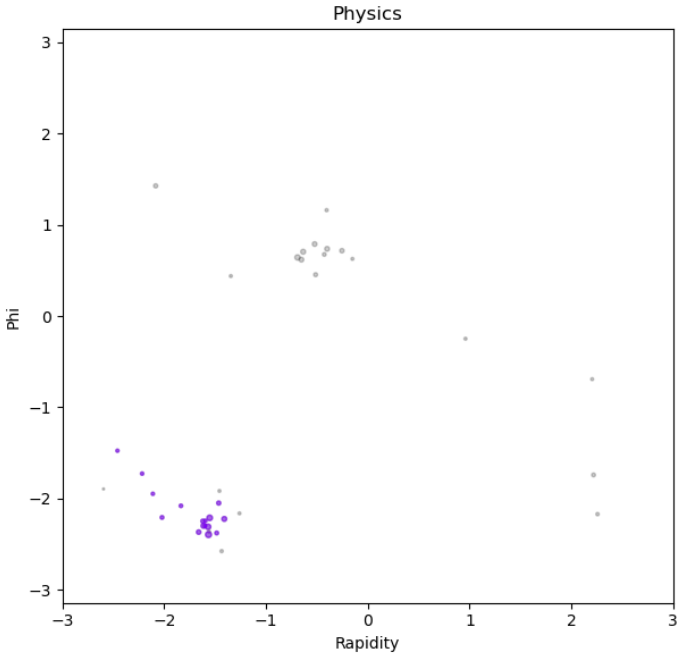
Mass peak is no longer an improvement on the anti-k_{T} algorithm, and also seen to be more fragile in the parameter ranges.

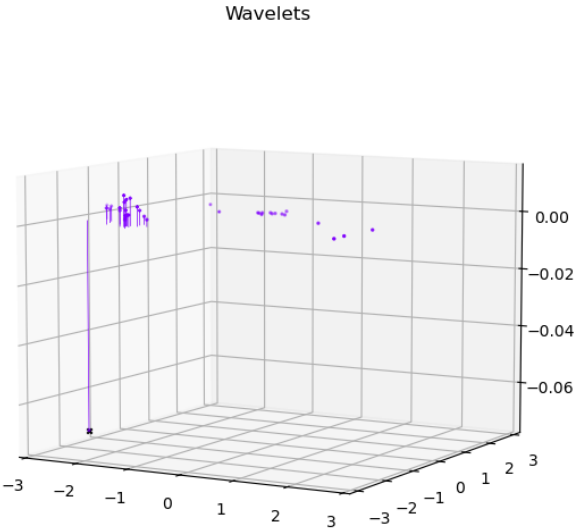
Conclusion

- If it could be efficiently implemented, the NCut objective would offer a nice improvement to jet definitions.
- It offers an explicit objective, and good signal selection, even in the presence of pileup.
- Improving efficiency is challenging, but tools are available, and we are making progress in this direction.

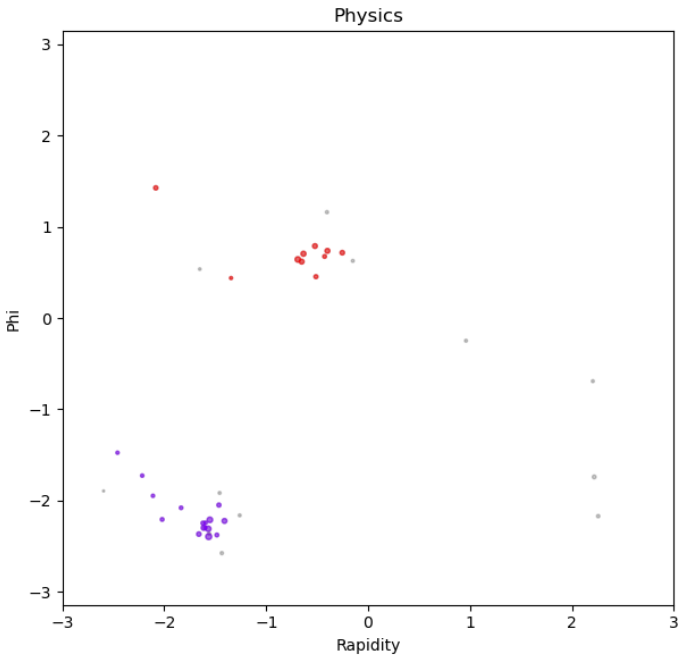
Thank you for your attention!

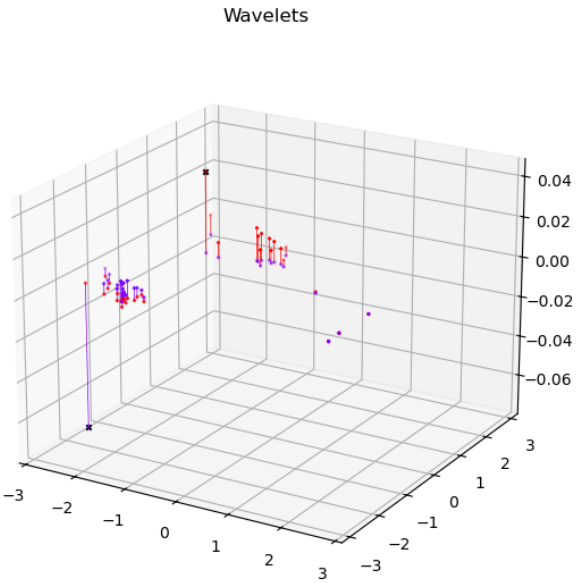
Backup; Chebyshev Wavelets



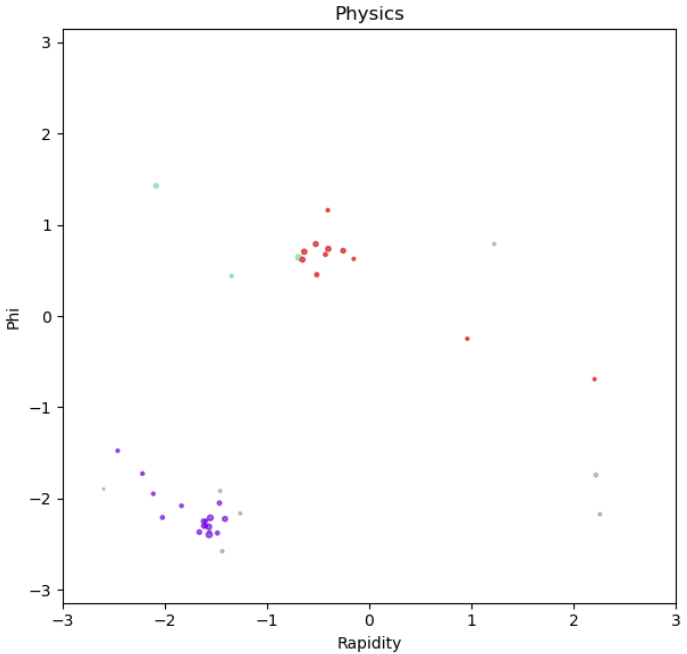


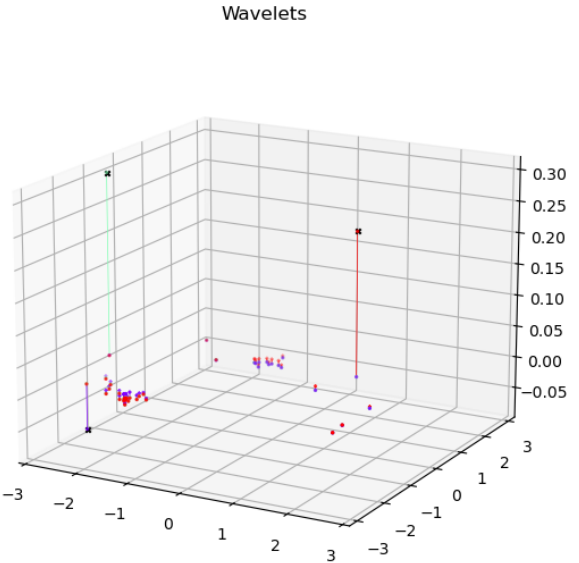
Backup; Chebyshev Wavelets



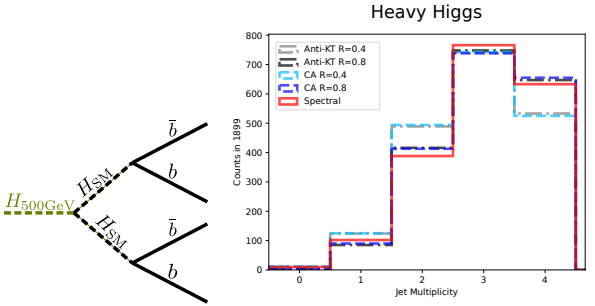


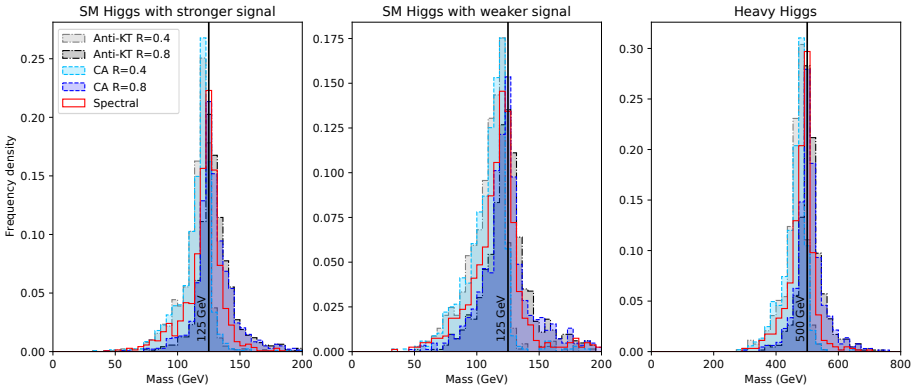
Backup; Chebyshev Wavelets



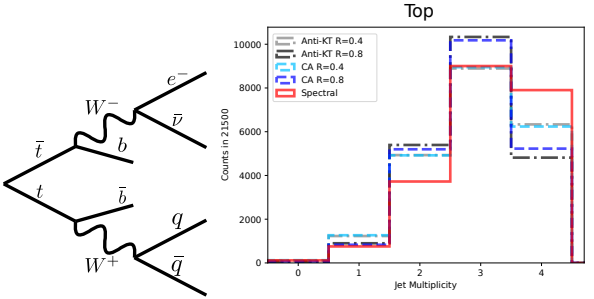


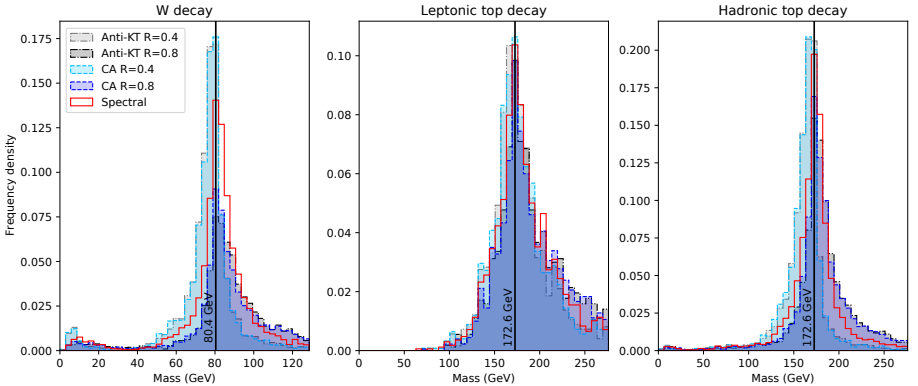
Backup; performance on heavier Higgs





Backup; performance on semileptonic top





Backup; Spectral Clustering

Relaxation and proof

Theory behind spectral clustering; `https://arxiv.org/abs/0711.0189`
Points to be clustered are considered as nodes of a graph. Label $j = 1 \dots n$.
Between, each pair, an 'affinity' is defined. Larger affinities for points that should be allocated to the same group. This results in a square matrix;

$$A = \underbrace{\left.\begin{bmatrix} 0 & a_{1,2} & a_{1,3} & \cdots & a_{1,n} \\ a_{2,1} & 0 & a_{2,3} & \cdots & a_{2,n} \\ a_{3,1} & a_{3,2} & 0 & \cdots & a_{3,n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & a_{n,3} & \cdots & 0 \end{bmatrix}\right\} n}_{n}$$

Backup; Spectral Clustering

Relaxation and proof

Theory behind spectral clustering; `https://arxiv.org/abs/0711.0189`
Points to be clustered are considered as nodes of a graph. Label $j = 1 \dots n$.
Between, each pair, an 'affinity' is defined. Larger affinities for points that should be allocated to the same group. This results in a square matrix;

$$A = \underbrace{\left.\begin{bmatrix} 0 & a_{1,2} & a_{1,3} & \cdots & a_{1,n} \\ a_{2,1} & 0 & a_{2,3} & \cdots & a_{2,n} \\ a_{3,1} & a_{3,2} & 0 & \cdots & a_{3,n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & a_{n,3} & \cdots & 0 \end{bmatrix}\right\} n}_{n}$$

For problems involving spatially distributed points, it is conventional to use a Gaussian kernel to define the affinity from the distance;

$$a_{i,j} = \exp\left(\frac{-d_{i,j}^2}{\sigma_v}\right)$$

Backup; Spectral Clustering

Relaxation and proof

- Denote a choice of clusters $G_{\mathbf{k}}$, where the index $\mathbf{k} = 1 \dots s$ is an index over the clusters.
- The set of all points outside cluster $G_{\mathbf{k}}$ is denoted $\bar{G}_{\mathbf{k}}$.
- $W(G_{\mathbf{k}}, \bar{G}_{\mathbf{k}})$ is all the affinities severed by separating $G_{\mathbf{k}}$ from the rest of the graph.

$$W(G_{\mathbf{k}}, \bar{G}_{\mathbf{k}}) = \sum_{i \in G_{\mathbf{k}}, j \in \bar{G}_{\mathbf{k}}} a_{i,j}$$

As stated earlier, minimising $\sum_{\mathbf{k}} W(G_{\mathbf{k}}, \bar{G}_{\mathbf{k}})$ tends to lead to uneven groups. The solution is to assign each cluster a weight $\mathrm{vol}(\mathrm{G}_{\mathbf{k}})$, indicating how much of the graph it contains. One possible choice is the sum of all affinities connecting to points in the cluster;

$$\mathrm{vol}(G_{\mathbf{k}}) = \sum_{i \in G_{\mathbf{k}}, j} a_{i,j}$$

In the new objective function, the cost of creating each group is divided by it's weight;

$$\mathrm{NCut} = \sum_{\mathbf{k}} \frac{W(G_{\mathbf{k}}, \bar{G}_{\mathbf{k}})}{\mathrm{vol}(G_{\mathbf{k}})}$$

Backup; Spectral Clustering

Relaxation and proof

Unfortunately, actually minimising this objective is NP hard (computationally intractable).

$$\mathrm{NCut} = \sum_{\mathbf{k}} \frac{W(G_{\mathbf{k}}, \bar{G}_{\mathbf{k}})}{\mathrm{vol}(G_{\mathbf{k}})}$$

However, there is a relaxed version, which is solvable in $\mathcal{O}(n^2)$.
The clusters could be fully determined by s indicator vectors;

$$h_{\mathbf{k},i} = \begin{cases} \frac{1}{\sqrt{\mathrm{vol}(G_k)}} & \text{if } i \in G_k \\ 0 & \text{otherwise} \end{cases}$$

Let D be a square, diagonal matrix, where $D_{i,i} = \sum_j a_{i,j}$. The unnormalised Laplacian can then be written as;

$$L = D - A = \underbrace{\begin{bmatrix} \sum_j a_{1,j} & -a_{1,2} & -a_{1,3} & \cdots & -a_{1,n} \\ -a_{2,1} & \sum_j a_{2,j} & -a_{2,3} & \cdots & -a_{2,n} \\ -a_{3,1} & -a_{3,2} & \sum_j a_{3,j} & \cdots & -a_{3,n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -a_{n,1} & -a_{n,2} - & a_{n,3} & \cdots & \sum_j a_{n,j} \end{bmatrix}}_{n} \Bigg\} n$$

Backup; Spectral Clustering

Relaxation and proof

Multiplying this Laplacian by a matching pair of indicator vectors;

$$
\begin{aligned}
h'_{\mathbf{k}} L h_{\mathbf{k}} &= \sum_{i,j} h_{\mathbf{k},i} L_{i,j} h_{\mathbf{k},j} \\
&= \sum_{i,j} h_{\mathbf{k},i} \left(\delta_{i,j} \sum_{p} a_{i,p} - a_{i,j} \right) h_{\mathbf{k},j} \\
&= \sum_{i} \left(h^2_{\mathbf{k},i} \sum_{p} a_{i,p} - \sum_{j} h_{\mathbf{k},i} h_{\mathbf{k},j} a_{i,j} \right) \\
&= \sum_{i,j} a_{i,j} \left(h^2_{\mathbf{k},i} - h_{\mathbf{k},i} h_{\mathbf{k},j} \right) \\
&= \frac{1}{2} \sum_{i,j} a_{i,j} \left(h_{\mathbf{k},i} - h_{\mathbf{k},j} \right)^2
\end{aligned}
$$

Backup; Spectral Clustering

Relaxation and proof

Multiplying this Laplacian by a matching pair of indicator vectors;

$$\begin{aligned} h'_{\mathbf{k}} L h_{\mathbf{k}} &= \sum_{i,j} h_{\mathbf{k},i} L_{i,j} h_{\mathbf{k},j} \\ &= \sum_{i,j} h_{\mathbf{k},i} \left(\delta_{i,j} \sum_{p} a_{i,p} - a_{i,j} \right) h_{\mathbf{k},j} \\ &= \sum_{i} \left(h^2_{\mathbf{k},i} \sum_{p} a_{i,p} - \sum_{j} h_{\mathbf{k},i} h_{\mathbf{k},j} a_{i,j} \right) \\ &= \sum_{i,j} a_{i,j} \left(h^2_{\mathbf{k},i} - h_{\mathbf{k},i} h_{\mathbf{k},j} \right) \\ &= \frac{1}{2} \sum_{i,j} a_{i,j} \left(h_{\mathbf{k},i} - h_{\mathbf{k},j} \right)^2 \end{aligned}$$

Using the definition of the indicator vector;
If both i and j are outside $G_{\mathbf{k}}$ then $h_{\mathbf{k},i} = h_{\mathbf{k},j} = 0$, so the last term vanishes.
If both i and j are inside $G_{\mathbf{k}}$ then $h_{\mathbf{k},i} = h_{\mathbf{k},j}$, so the last still term vanishes.
So only the cross terms remain.

Backup; Spectral Clustering

Relaxation and proof

Condensing this down;

$$\begin{aligned} h'_{\mathbf{k}} L h_{\mathbf{k}} =& \frac{1}{2} \sum_{i \in G_{\mathbf{k}}, j \in \bar{G}_{\mathbf{k}}} a_{i,j} \left(h_{\mathbf{k},i} - h_{\mathbf{k},j} \right)^2 \\ =& \frac{1}{2} \sum_{i \in G_{\mathbf{k}}, j \in \bar{G}_{\mathbf{k}}} \frac{a_{i,j}}{\sqrt{\mathrm{vol}(G_{\mathbf{k}})}^2} \end{aligned}$$

Which looks is exactly what we wanted to minimise.

Backup; Spectral Clustering

Relaxation and proof

The objective has been rephrased as;

$$h_{\mathbf{k}}' L h_{\mathbf{k}} = \frac{1}{2} \sum_{i \in G_{\mathbf{k}}, j \in \bar{G}_{\mathbf{k}}} \frac{a_{i,j}}{\mathrm{vol}(G_{\mathbf{k}})}$$

Now recall the Rayleigh quotient, and the min-max theorem, which states that;

given a Hermitian matrix M
the vector x (with $||x|| = 1$) that minimises $x'Mx$
is the eigenvector of M corresponding to the smallest eigenvector.

This is almost what we need. Two problems;

1. Our $h_{\mathbf{k}}$ are not normalised, $||h_{\mathbf{k}}|| \neq 1$.
   Solvable; make the normalisation, then absorb it into the definition of the Laplacian.

   $$h_{\mathbf{k}}' L h_{\mathbf{k}} \rightarrow h_{\mathbf{k}}' D^{-1/2} L D^{-1/2} h_{\mathbf{k}}$$

   so define

   $$L_{\mathrm{symm}} = D^{-1/2} L D^{-1/2}$$

   .
2. The min-max theorem does not in general produce piecewise-constant x, so the x will not have the form defined for the $h_{\mathbf{k}}$. Not solvable; this is the relaxation.

Backup; Spectral Clustering

Relaxation and proof

To summarise;
The objective is to find clusters that minimise

$$\mathrm{NCut} = \sum_{\mathbf{k}} \frac{W(G_{\mathbf{k}}, \bar{G}_{\mathbf{k}})}{\mathrm{vol}(G_{\mathbf{k}})}$$

.
This is equivalent to finding h_{k} that minimise

$$h'_{\mathbf{k}} L_{\mathrm{symm}} h_{\mathbf{k}}$$

where $L_{\mathrm{symm}} = D^{-1/2}(D - A)D^{-1/2}$.
Solving that directly is NP hard, but if $h_{\mathbf{k}}$ is exchanged for a vector, x, whose values are only required to be normalised, then

$$x' L_{\mathrm{symm}} x$$

is minimised by the eigenvectors of L_{symm} corresponding to the smallest eigenvalue.