Cluster Scanning: a novel approach to resonance searches

Ivan Oleksiyuk*, John Raine, Tobias Golling, Slava Voloshynovskiy

University of Geneva

Michael Krämer

RWTH Aachen

*ivan.oleksiyuk@unige.ch

ML4Jets 2023

Problem formulation

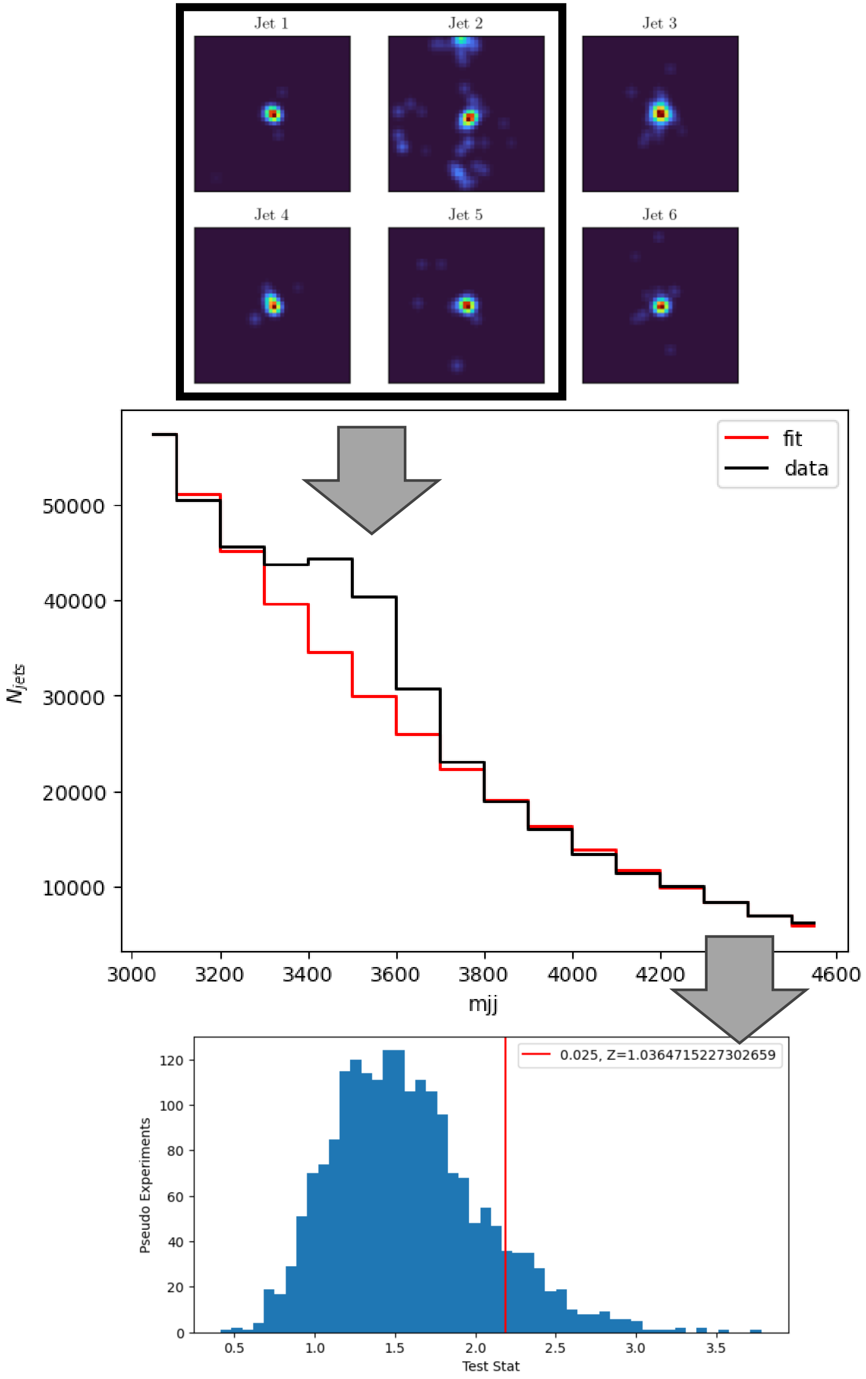
Bump hunting:

1. Select a signal rich subspace
   ⇒ based on signal model

2. Find a way to estimate background
   ⇒ Usually n-parameter fit or SWIFT

3. Define and calibrate test statistic

4. Unblind and find significance/limits

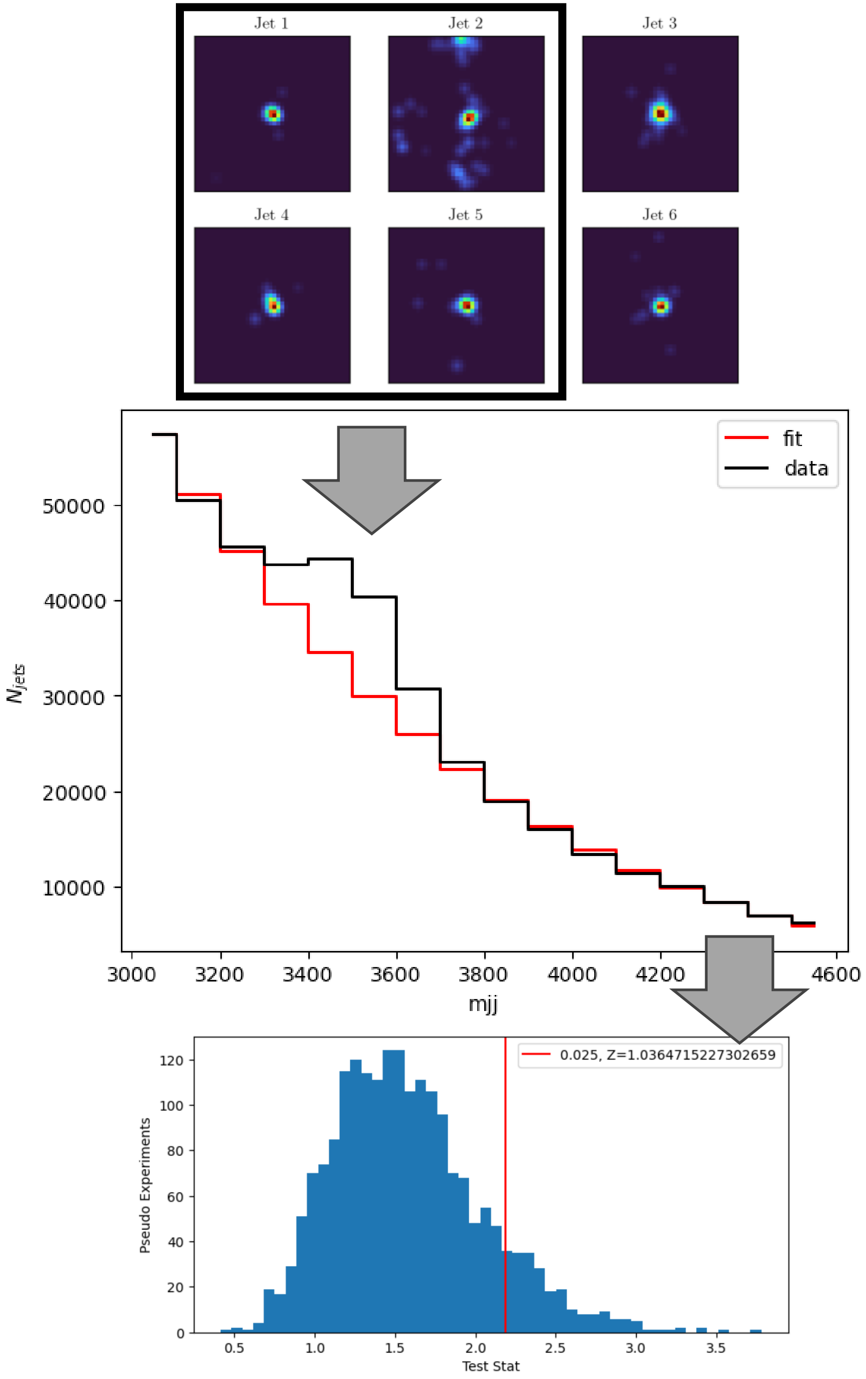


Problem formulation

Bump hunting:

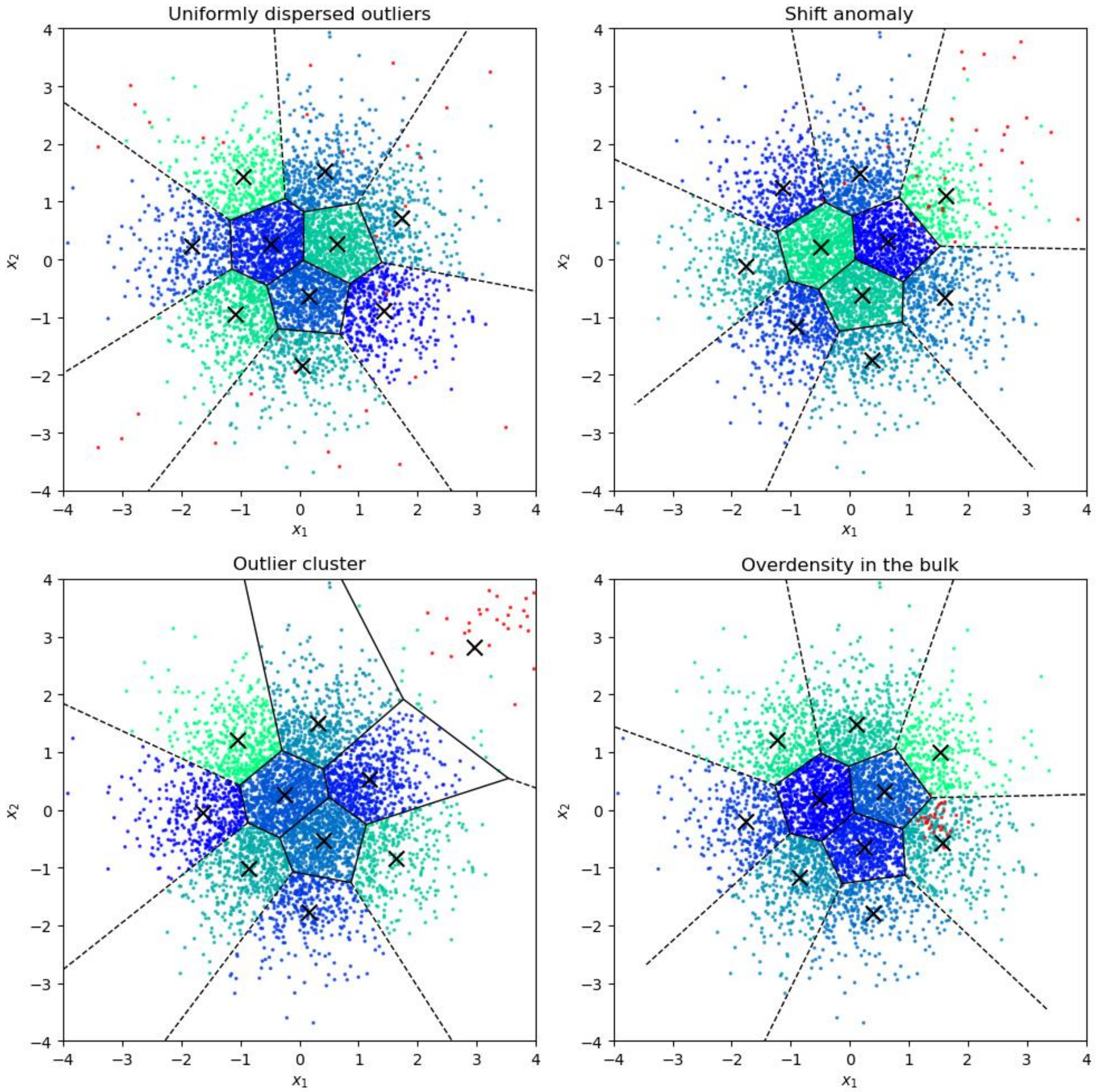
1. Select a signal rich subspace
   ⇒ based on signal model
   Can we do it in a model agnostic way? Use unsupervised ML?
2. Find a way to estimate background
   ⇒ Usually n-parameter fit or SWIFT
   Can we do this without assumptions on functional form or smoothness?
3. Define and calibrate test statistic
   Need fast methods for calibration
4. Unblind and find significance/limits

Answer: Cluster Scanning!



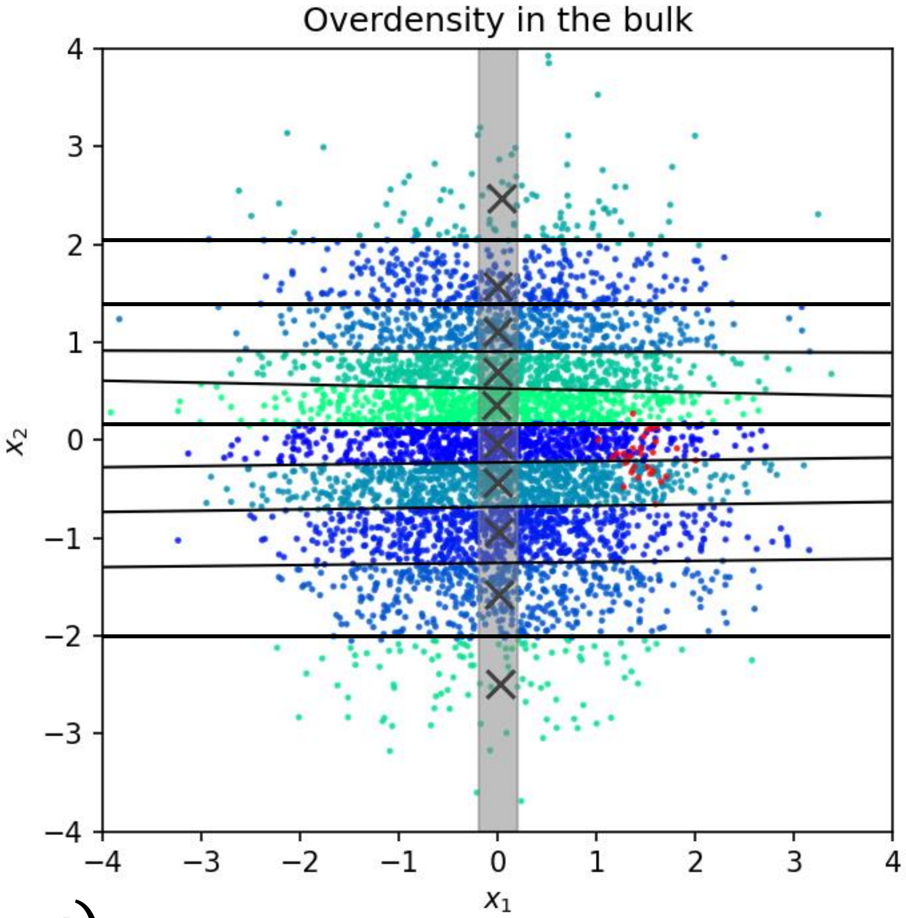
Outliers or Overdensities?

- Usual assumption:
  anomalies = outliers
- In HEP: signal is produced by the same process
- Anomalies localised⇒Use clustering
- **Small number of clusters contain several times more signal than the rest**



Smoothness or Independence?

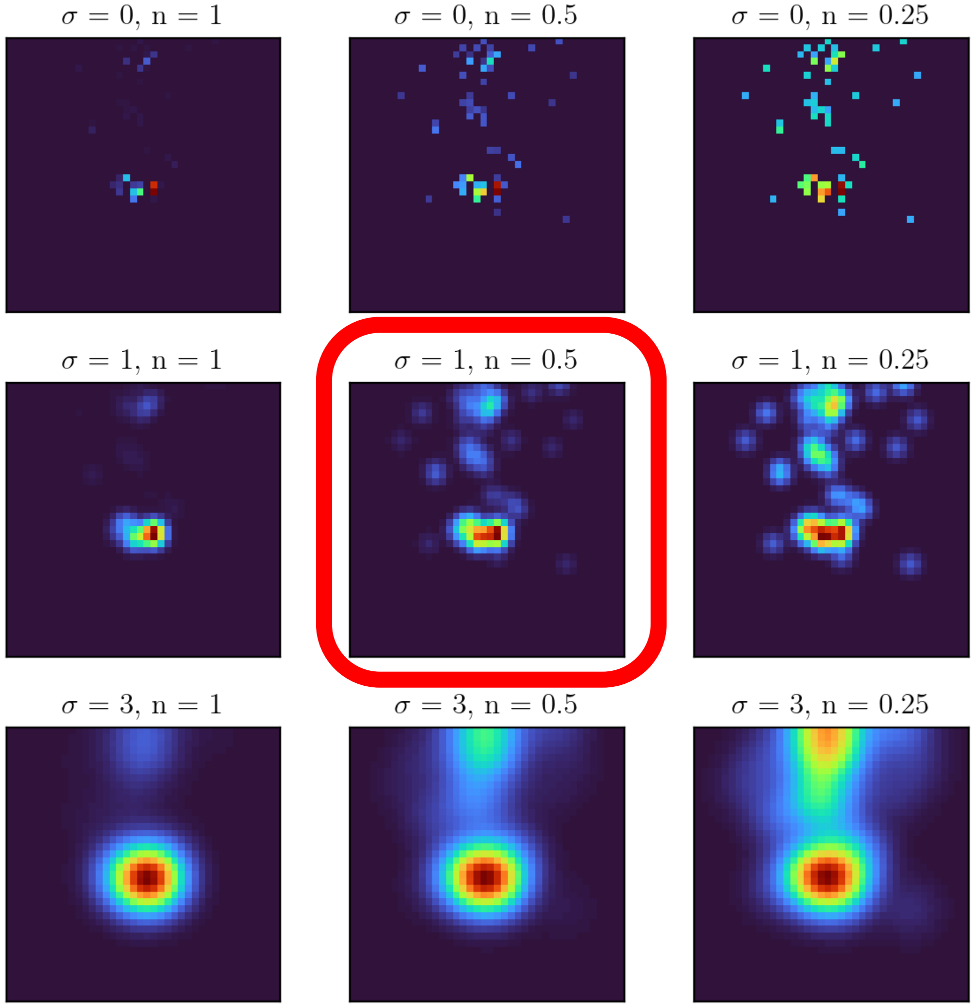
- Usual assumption:
  Background is smooth/parametrizable with
  $f(x) = p_1(1-x)^{p_2} x^{p_3 + p_4 \ln(x) + p_5 \ln(x^2)}$
  But it is just a good guess
- Our assumption:
  Clustering jets in a narrow m_{jj} window will make m_{jj} and cluster index independent variables



$$p(m_{jj} | jet\ in\ cluster\ i) \approx p(m_{jj} | jet\ in\ cluster\ j)$$

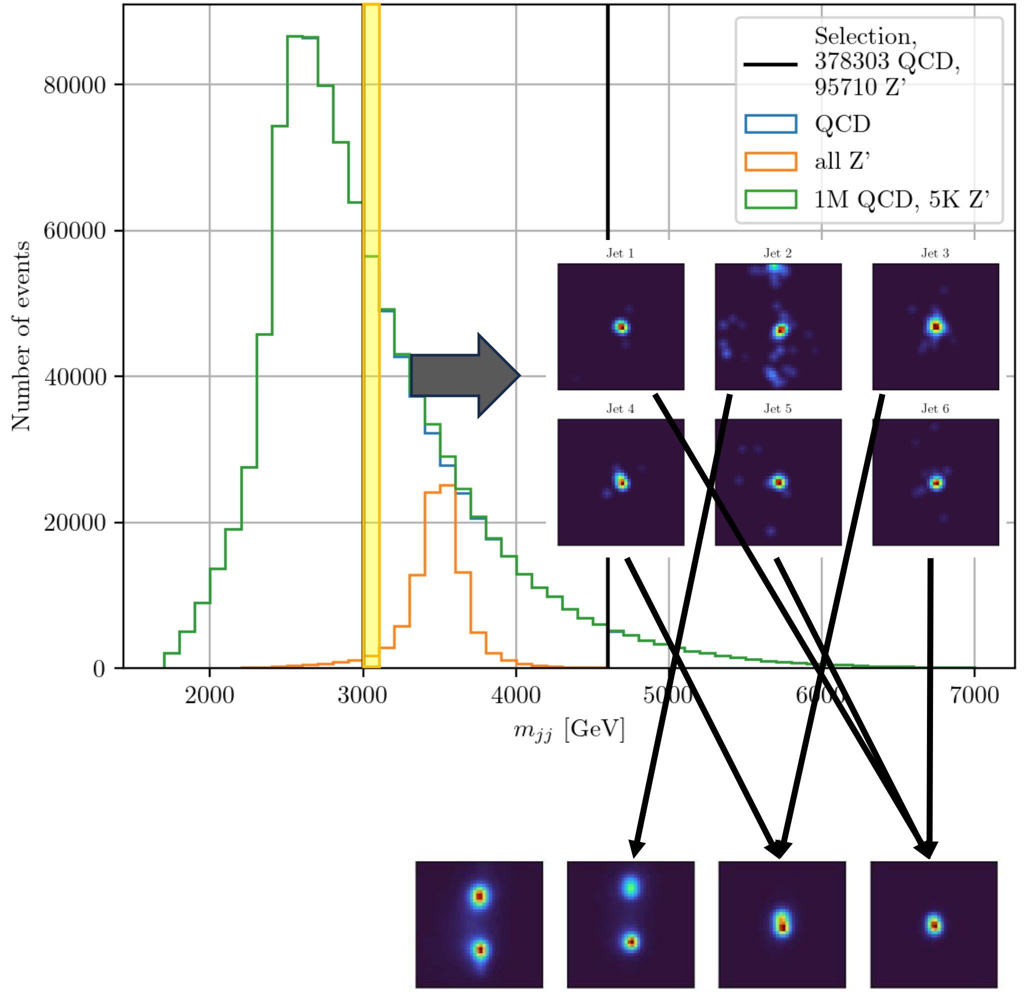
Data and Preprocessing

- Use LHCO R&D dataset with QCD background and Z’ signal
- Low-level features
  ⇒ jet-images
- Images are very sparce
  ⇒ smearing with a gaussian kernel
- Pixel intensities span several orders of magnitude
  ⇒ Apply power function with $n = 0.5$



Method

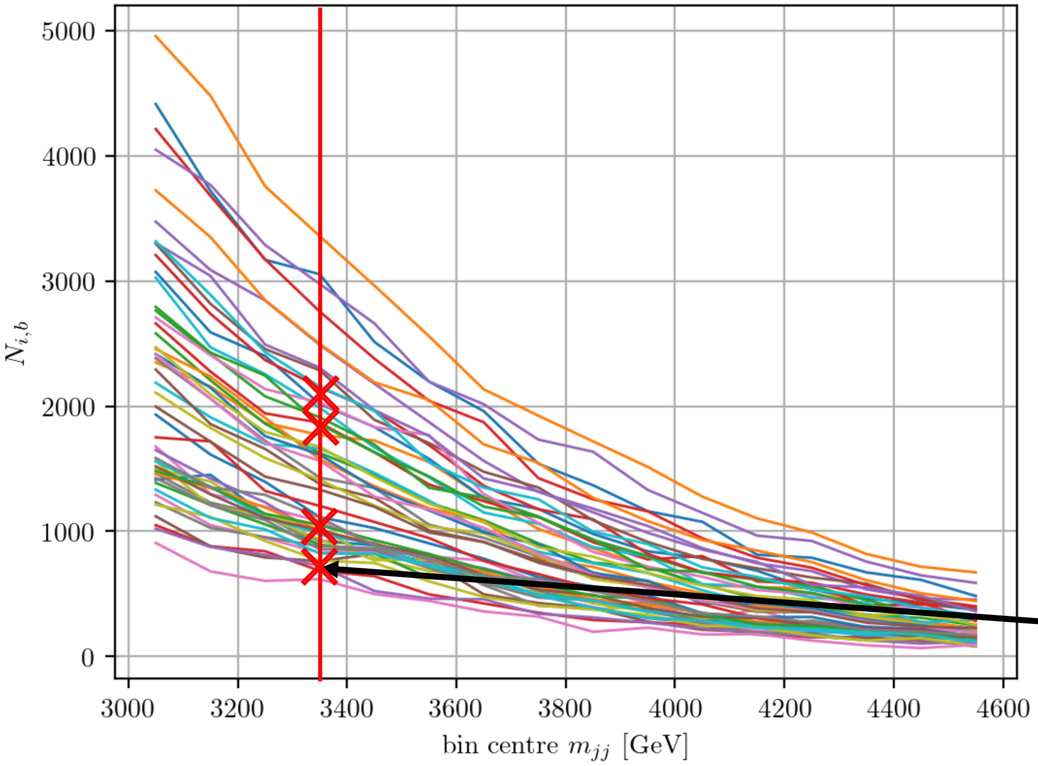
1. Take all jets from narrow m_{jj} window.
2. Use **mini-batch k-means** to cluster jet images into 50 clusters.

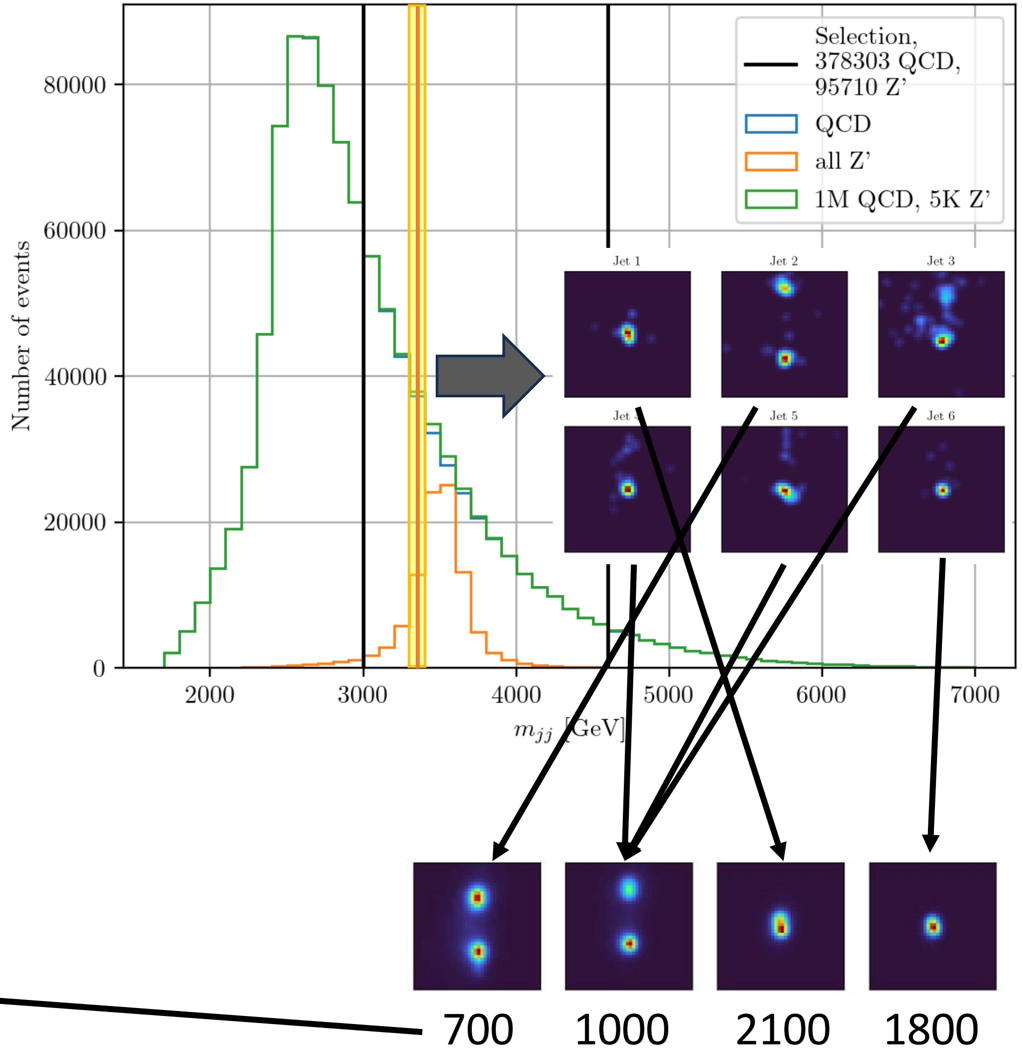


Cluster centroids

Method

3. For each m_{jj} bin assigning jets to the cluster they are closest to
4. Scan through all bins to get cluster distributions

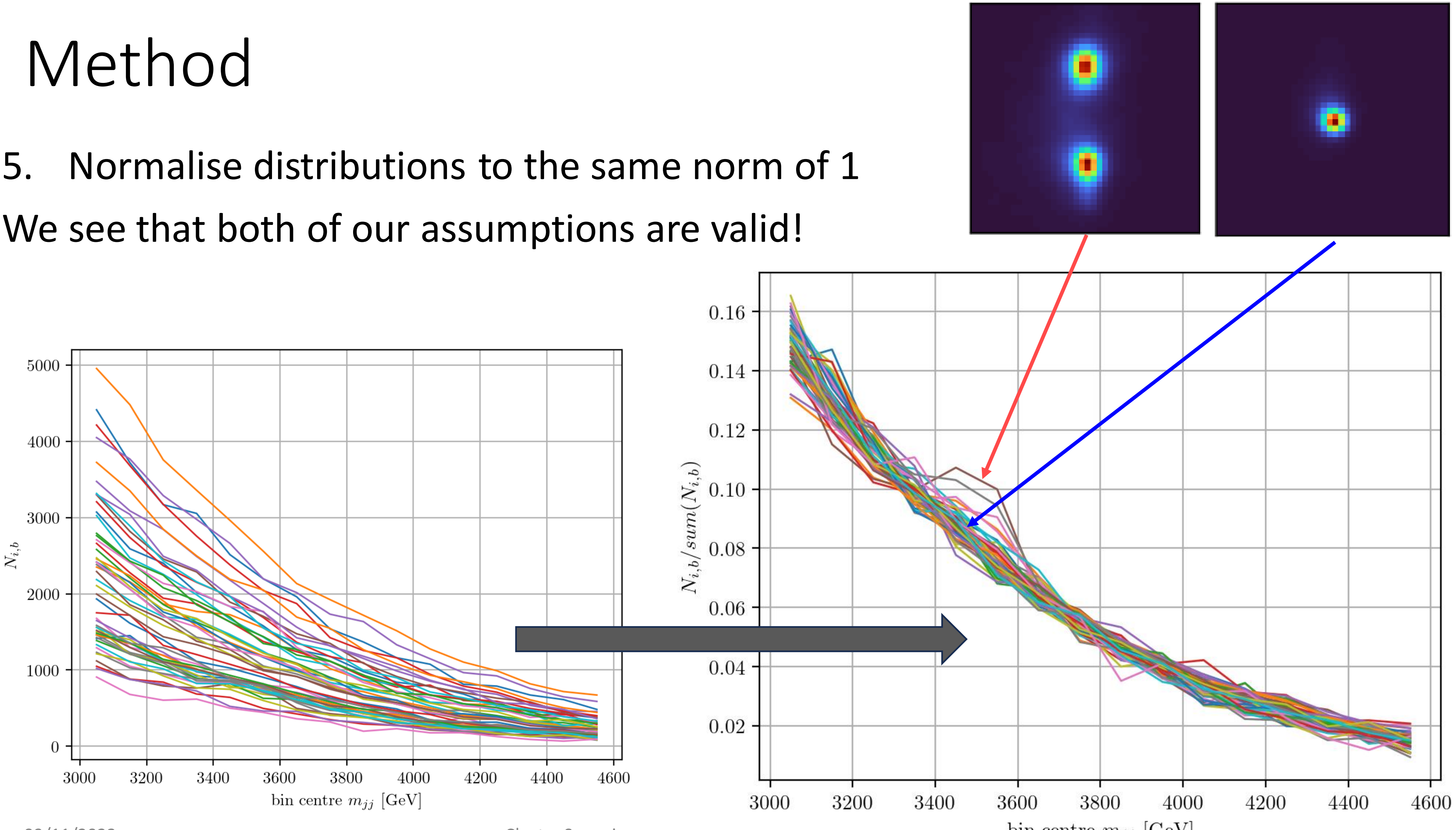




Method

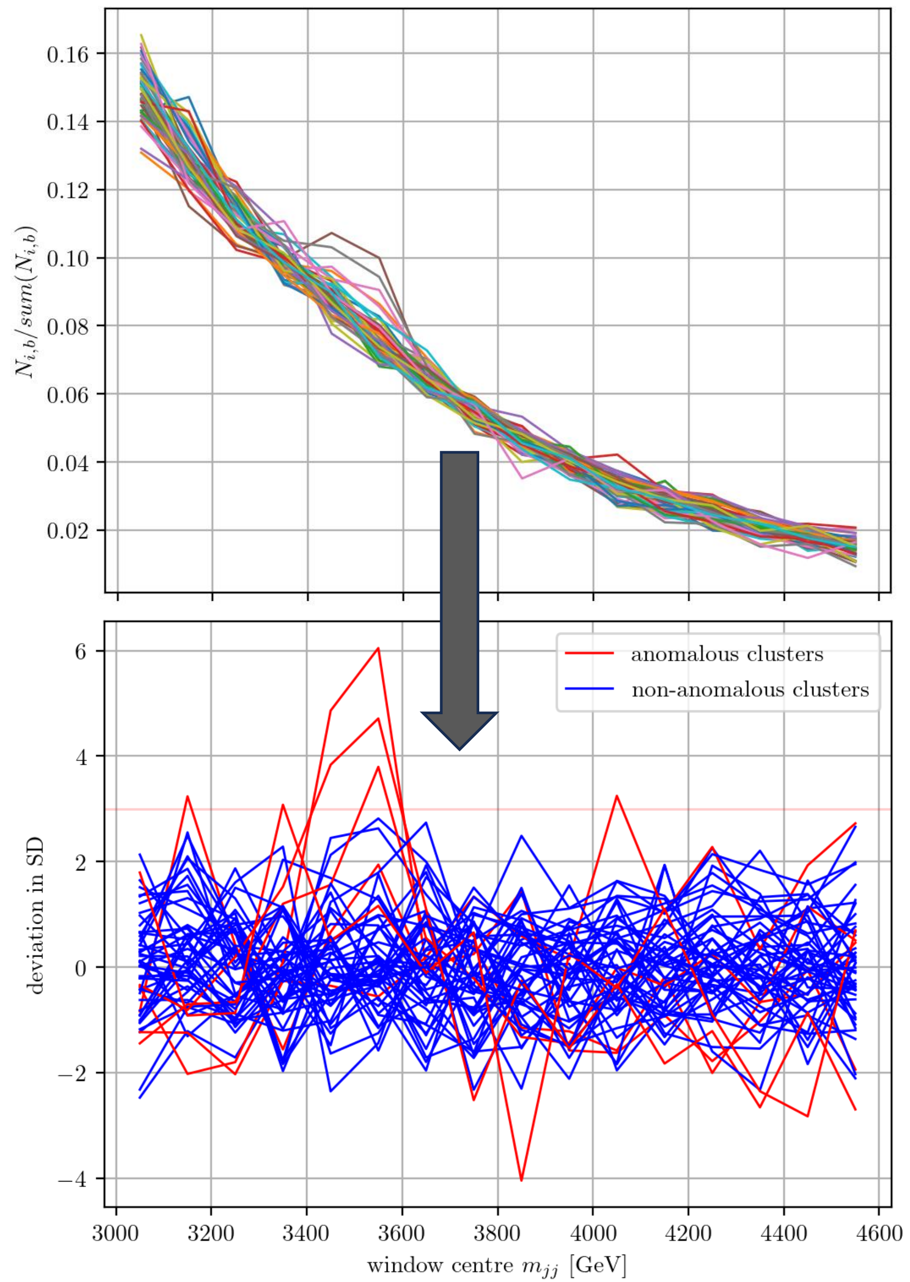
5. Normalise distributions to the same norm of 1

We see that both of our assumptions are valid!



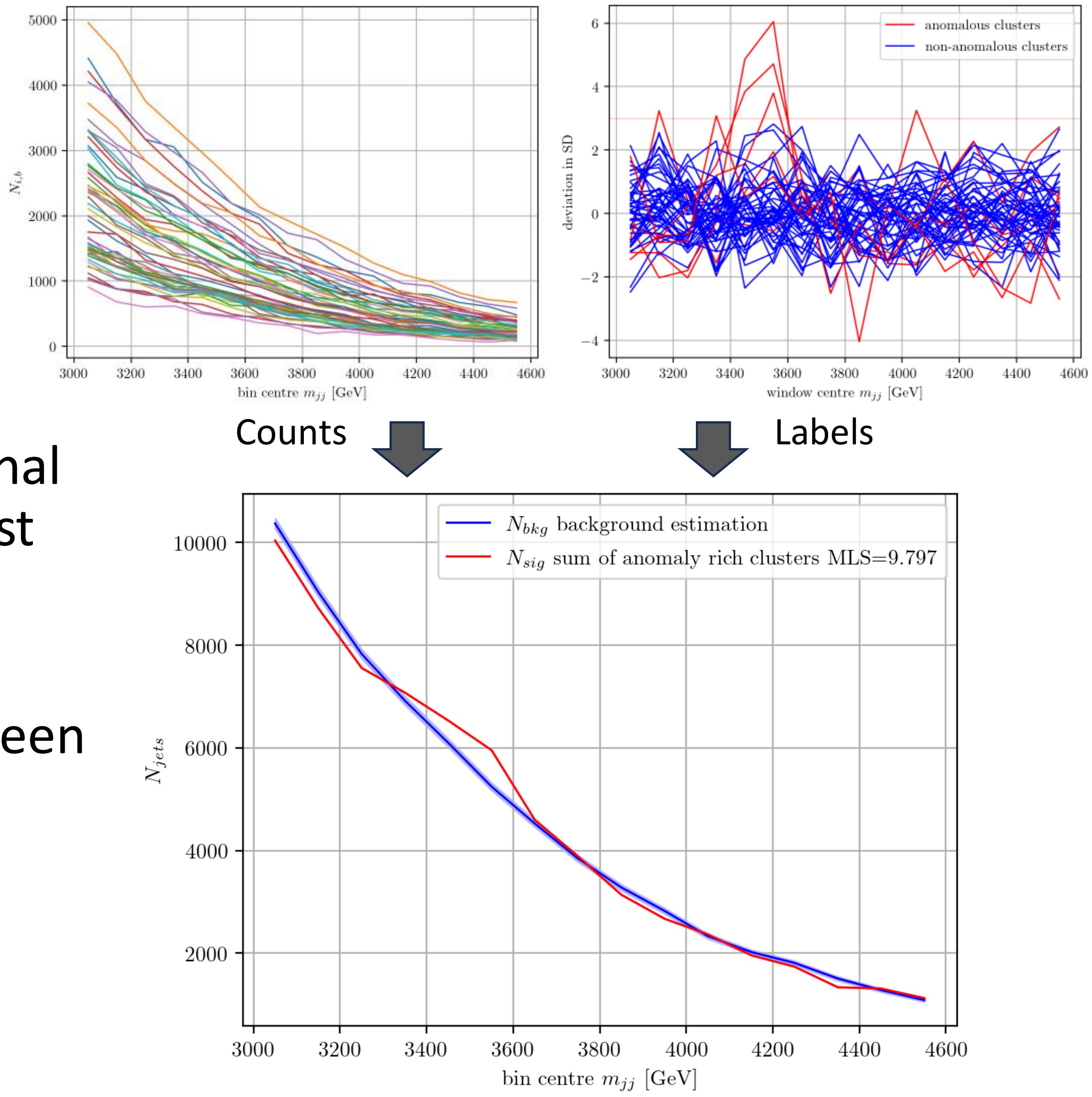
Method

6. Standardize in each bin using outlier robust mean and standard deviations
7. Label all the clusters that deviate more than 3 robust standard deviations as signal rich and the rest signal depleted



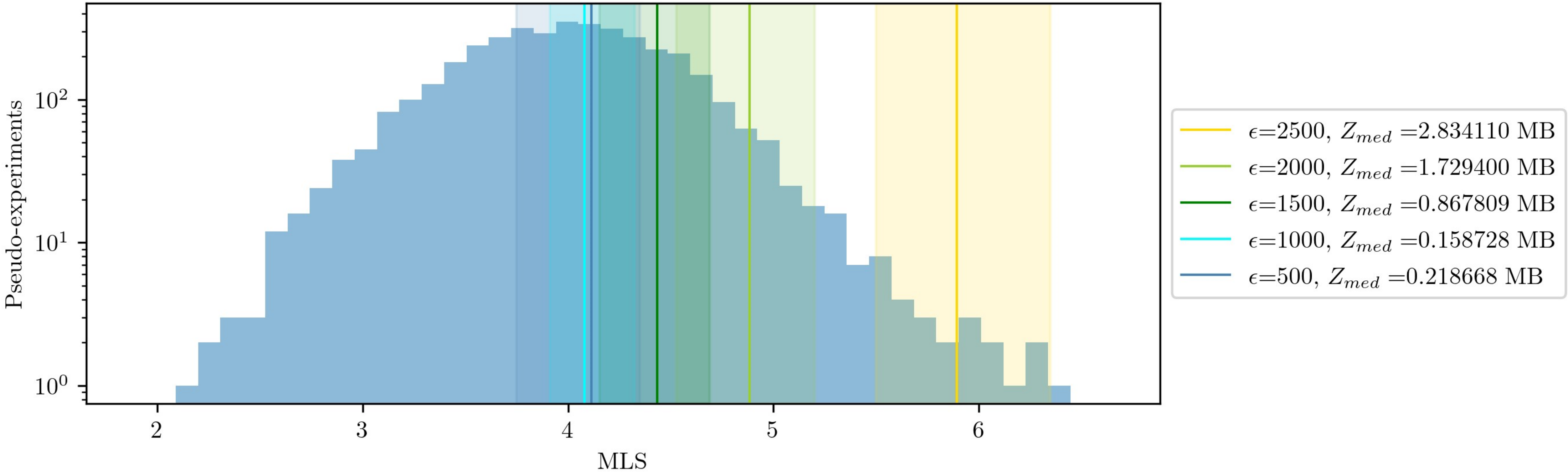
Method

8. Combine selected clusters into signal rich spectrum with a bump and rest into background estimate
9. Find test statistic (maximum local significance) from difference between them



Method

10. Ensemble by averaging several test statistics of several k-means initialisations
11. Calibrate using bootstrap resampling
12. Evaluate p-value for signal contaminated pseudo-experiments



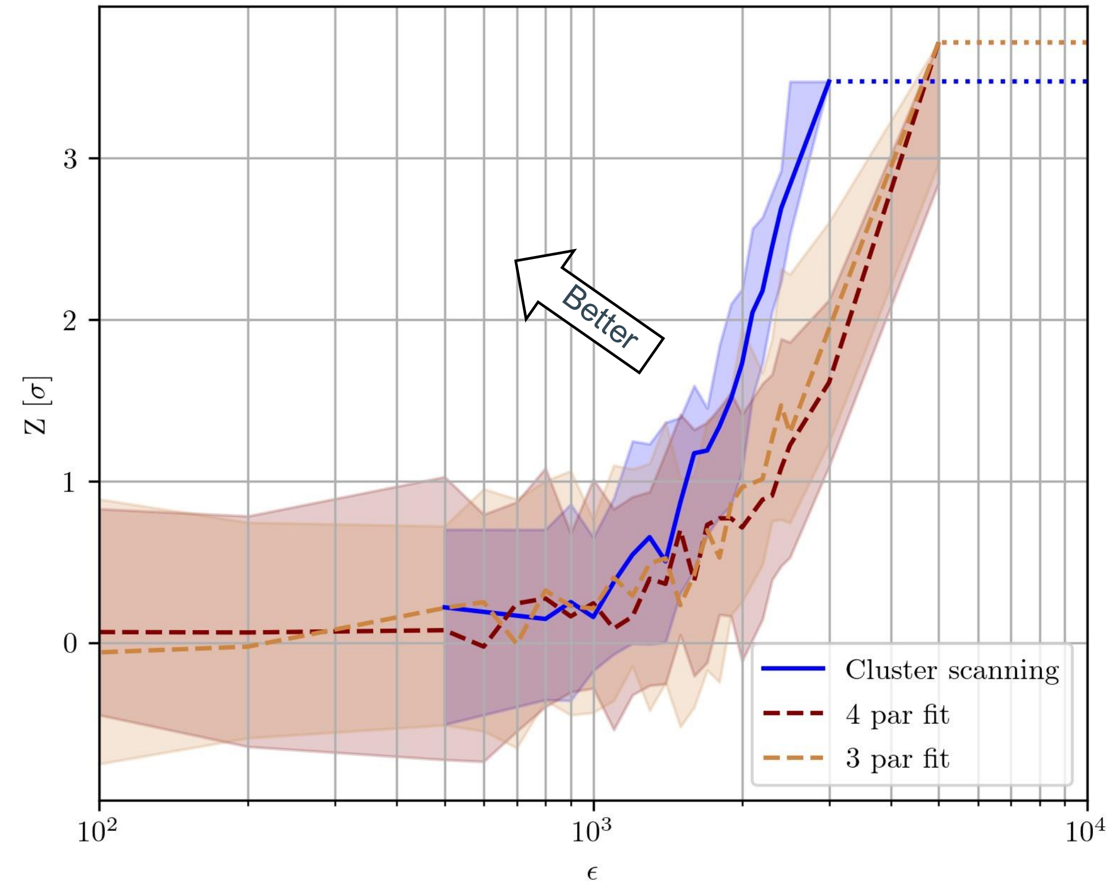
Method

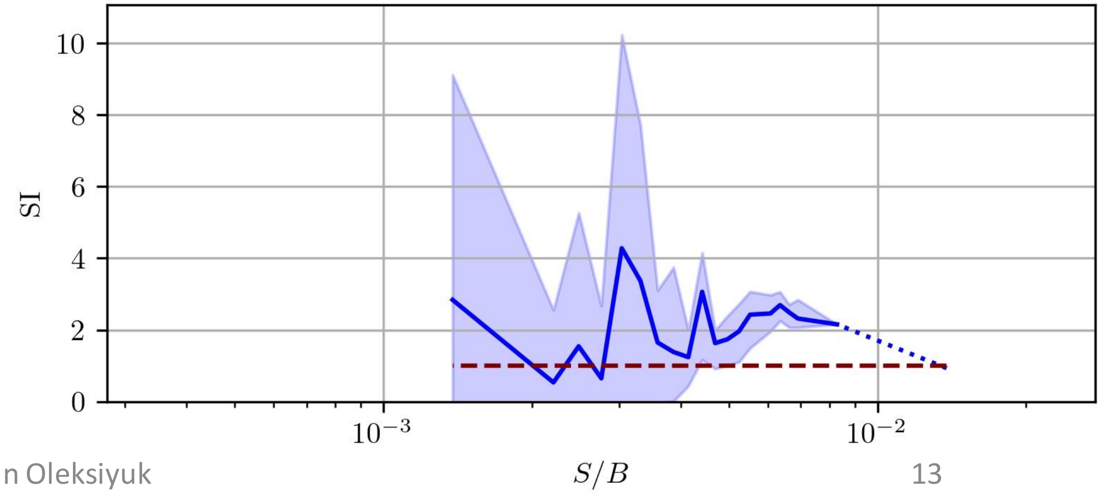
- Benchmark: global fit with n-parameter dijet fit function.

$$f(x) = p_1(1-x)^{p_2} x^{p_3 + p_4 \ln(x) + p_5 \ln(x^2)}$$

- Can detect 3-sigma evidence with only **61%** of the signal needed for the fit-based method.

<u>Cluster Scanning works for narrow resonance searches!</u>



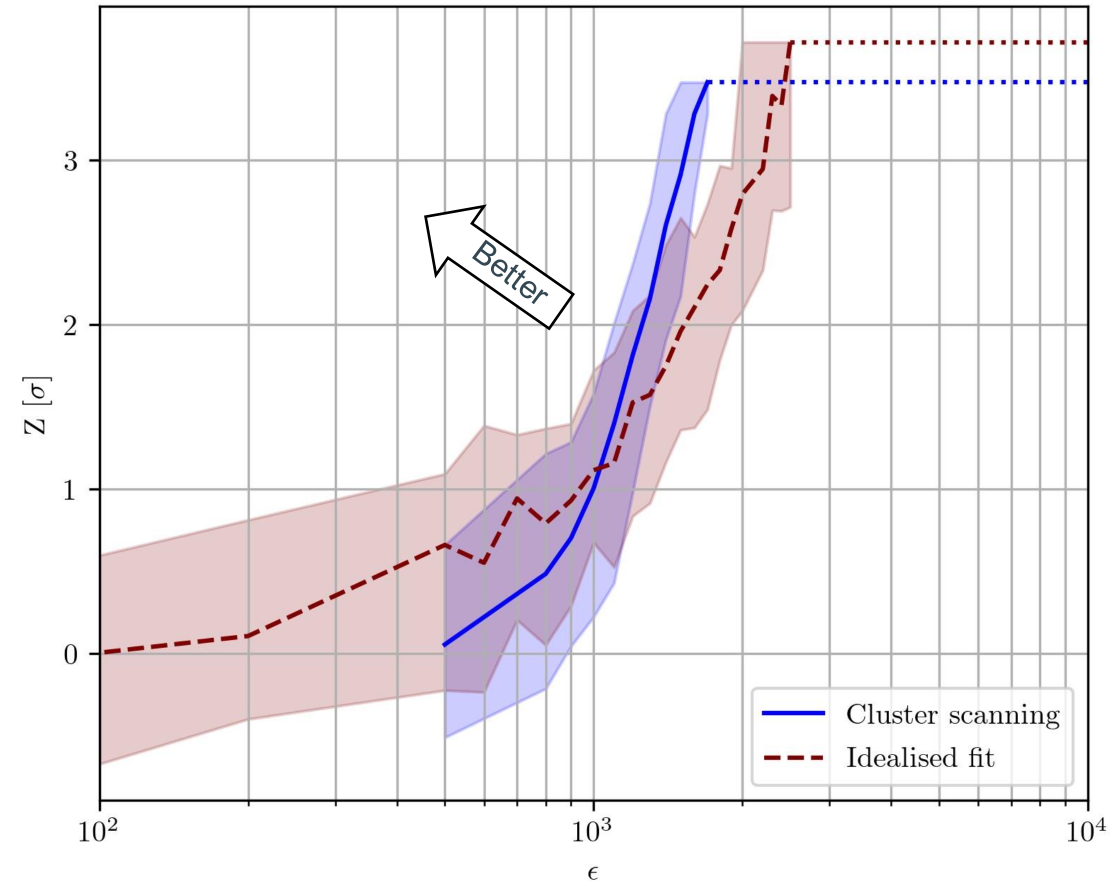


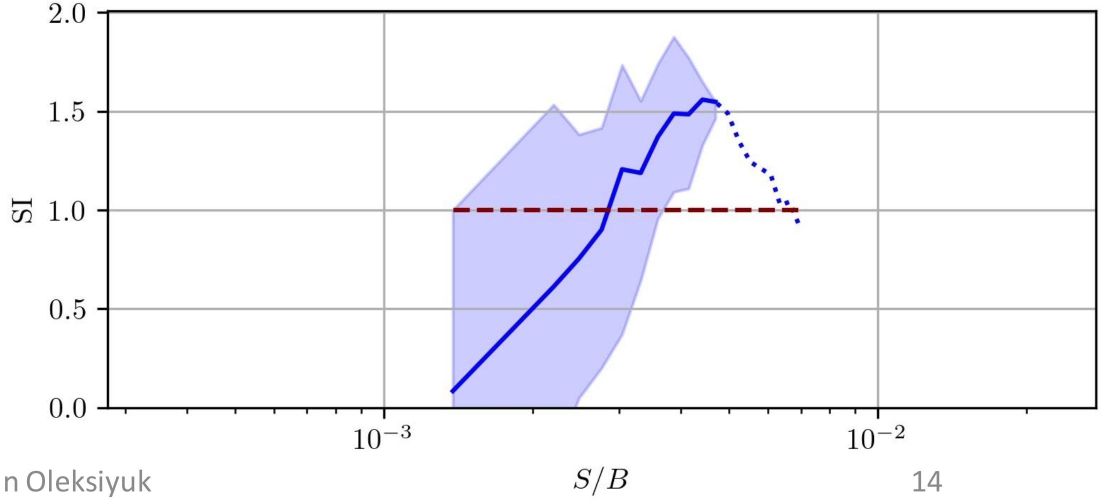
Idealised performance

Case: model for background is known

- **Idealised fit:**
  Fit = background expectation
  Analysed sample = expectation + statistic fluctuation fluctuations
- **Idealised CS:**
  $$p\left(m_{jj}|Cluster\ i\right) = p\left(m_{jj}|Cluster\ j\right)$$

<u>$\boldsymbol{m_{jj}}$ **+ low-level features > only** $\boldsymbol{m_{jj}}$</u>





Conclusion

Cluster scanning is:

- **Useful:** improves significance compared to global functional fit
- **Versatile:** background estimate without fitting + model agnostic
- **Complementary:** different set of assumptions
- **Fast:** ensembling and calibration

Potential further applications - Synergy with Deep Learning:

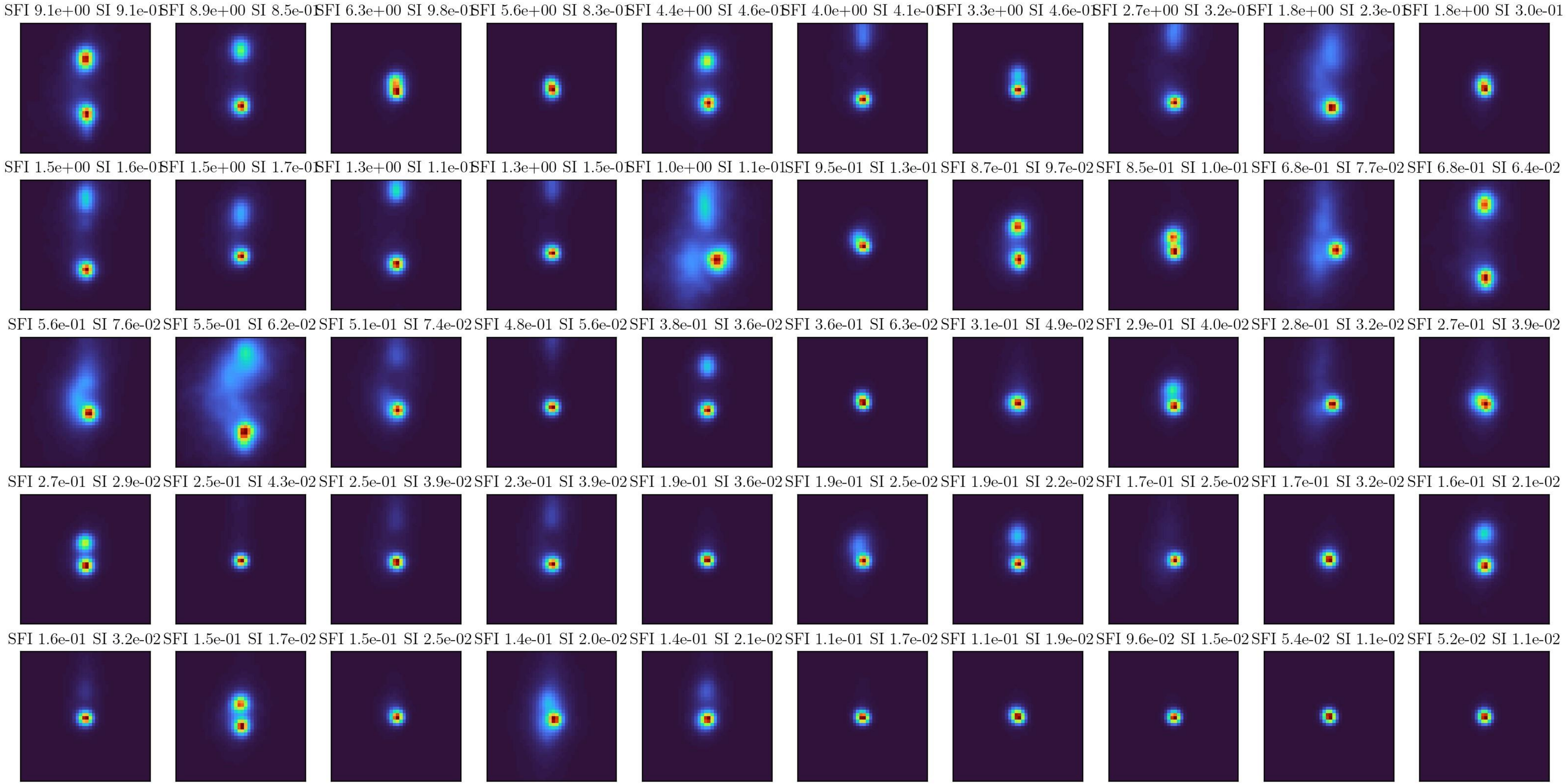
- Apply to features extracted by supervised/unsupervised/SSL deep learning
- Apply after a cut on the anomaly score in anomaly detection methods work in mass sculpting regime

Thank you for attention

Please ask your questions

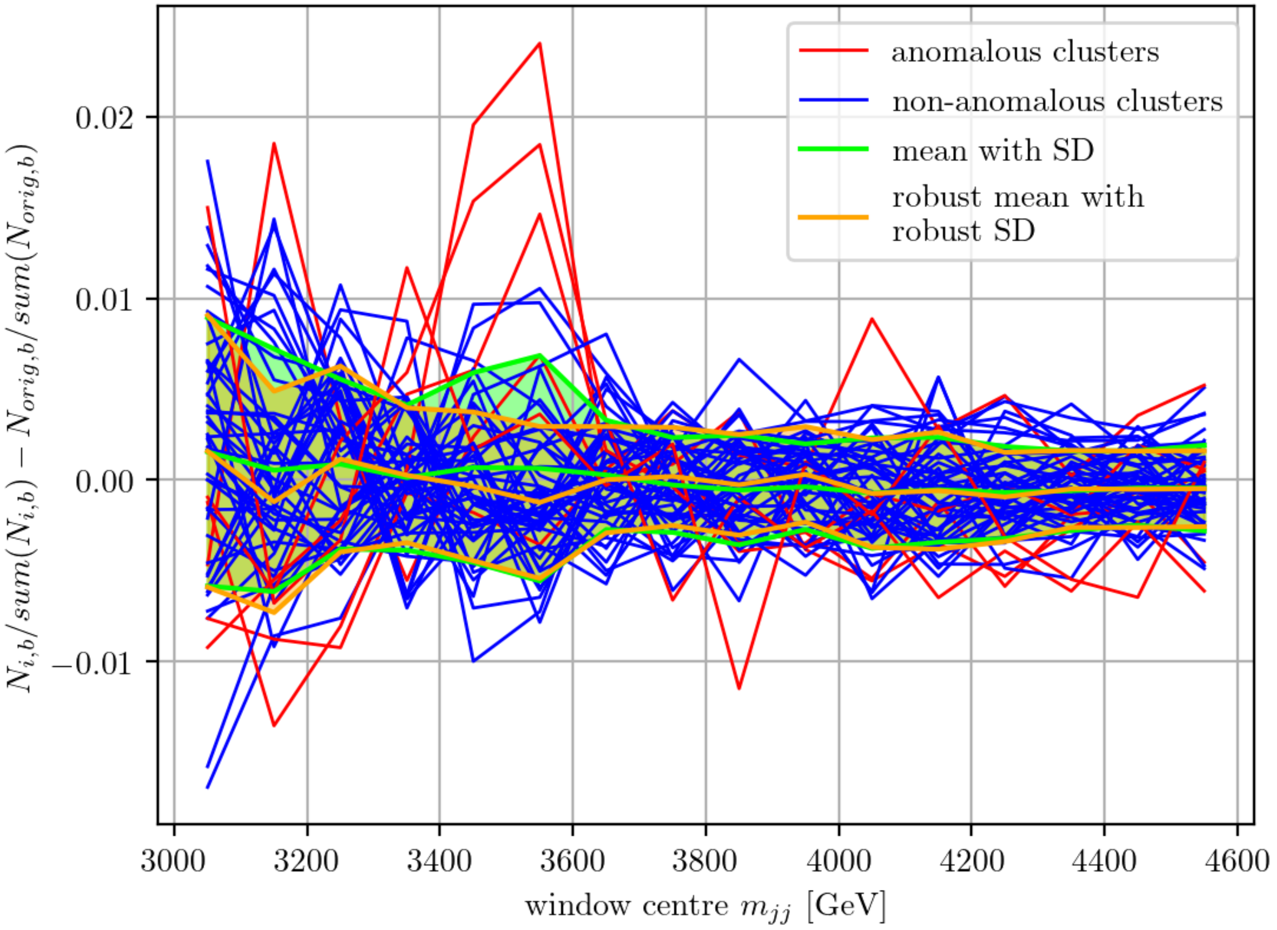
Arxive coming soon!

Backup: clusters

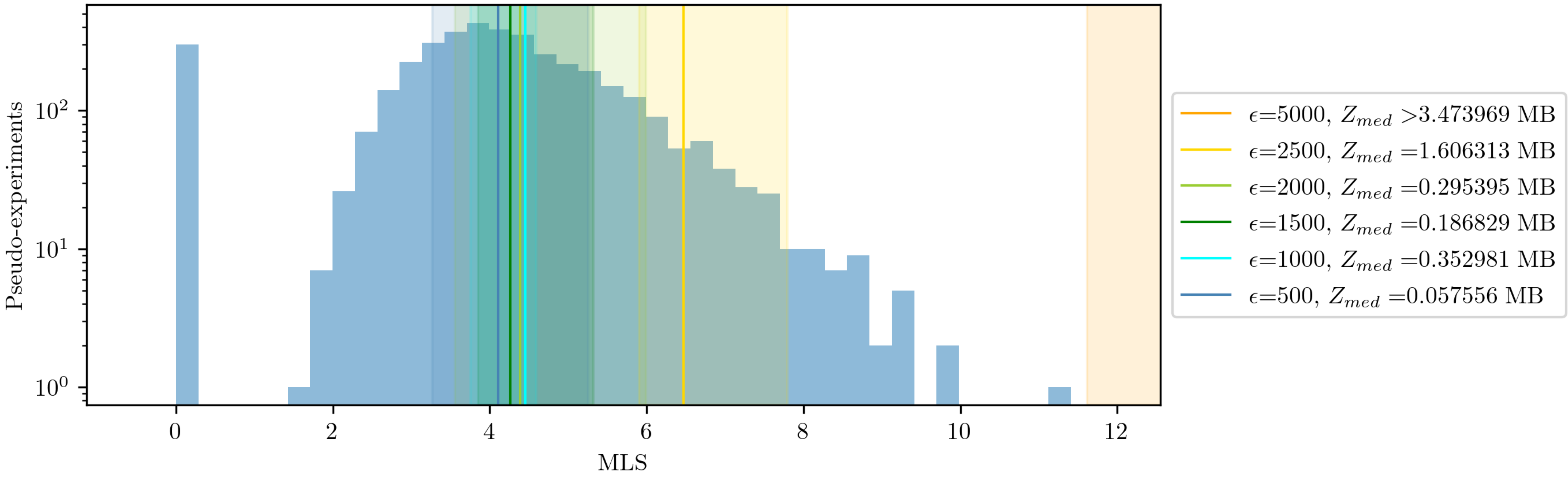


Backup: outlier robust measures

While searching for outliers, it is preferred to use outlier robust estimators for standard deviation (SD) and mean. We define them as follows: given a sample of observations $S = \{\vec{x_1}, \vec{x_2}, \ldots \vec{x_n}\}$ we find a median $med(\vec{x})$ (which is itself an outlier robust estimator) of this sample and take a subsample $\tilde{S}_f$ that is constructed from S by discarding a fraction $0 < f < 1$ of all samples that have largest absolute distance to this median. In this way we have discarded the outliers. After that we construct estimators $\tilde{\mu}_f = mean(\tilde{S}_f)$ and $\tilde{\sigma}_f = SD(\tilde{S}_f) \cdot g(f)$. If S is a sample from $\mathcal{N}(\mu, \sigma)$ it is obvious that with $\lim\limits_{n\to\infty} \tilde{\mu}_f = \lim\limits_{n\to\infty} mean(S) = \mu$. If one takes S from $\mathcal{N}(0,1)$ and rescales $\vec{x_i} \to \sigma\vec{x_i}$, then both estimators transform $\tilde{\sigma}_f \to \sigma\tilde{\sigma}_f$ and $SD(S) \to \sigma SD(S)$ by definition, so both estimators $\tilde{\sigma}_f$ and $SD(S)$ are proportional to a true σ of the Gaussian distribution.



Backup: no ensambling



Backup: training in the signal rich region