Framing secondary vertexing as a clustering problem: possible heuristics

Diptaparna Biswas

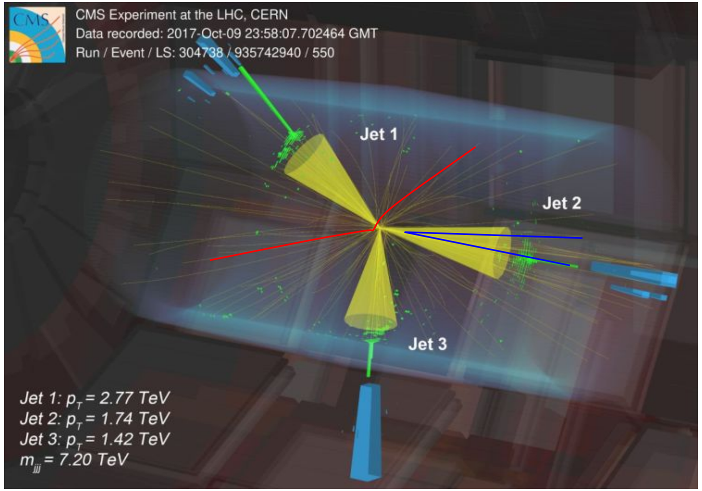
Universität Siegen / HEP Experimental High Energy Physics / CPPS Center for Particle Physics Siegen

STAR Sensing and Sensibility Transcending Disciplines for a Responsible Future

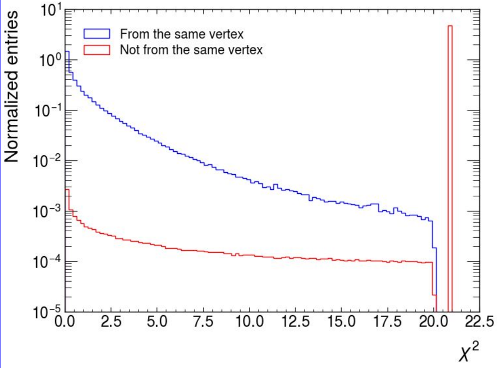
TrackOpt Meeting
7th November 2025

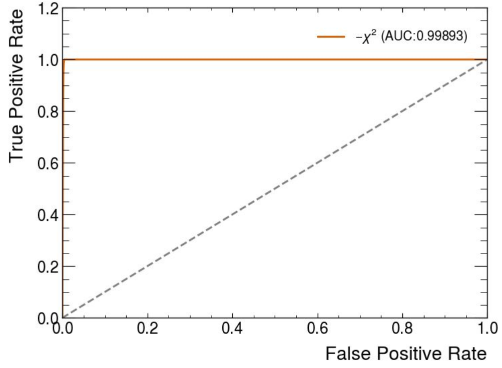
Looking for a heuristic

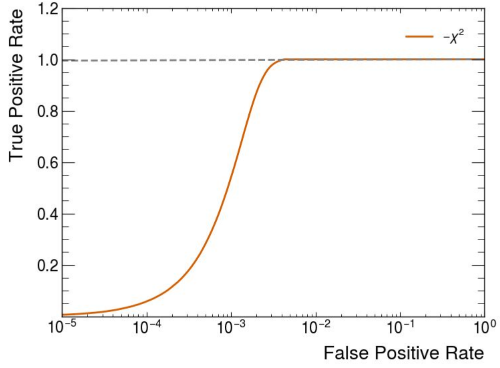
- For vertex finding inside a jet, the jet cone itself works as a bounding box.
  - ➢ We need to consider combinations only among the tracks (~20) inside the cone.
- However, for an arbitrary secondary vertex, tracks can go in all directions.
  - ➢ The track parameters themselves don't provide a measure of *closeness*.
    - ■ May introduce bias.
  - ➢ How to tackle combinatorial complexity?

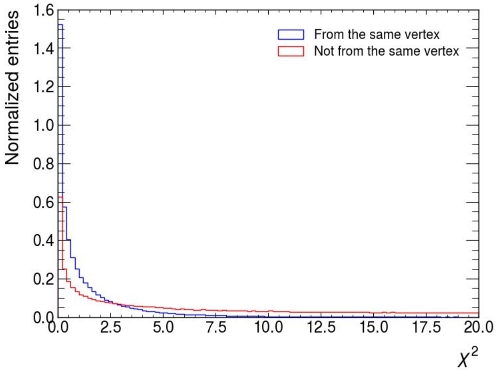


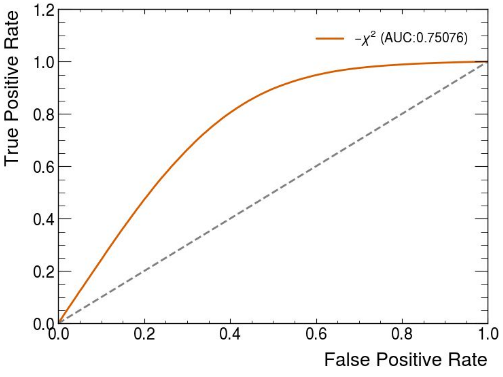
Utilizing χ^2 from Billoir fits of track-pairs

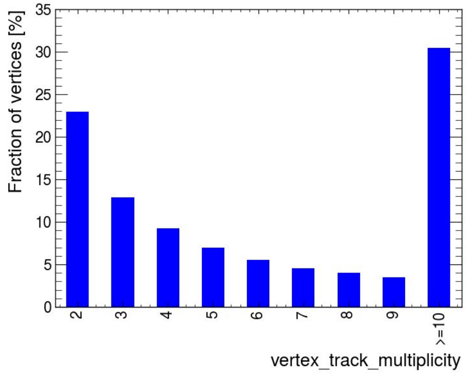










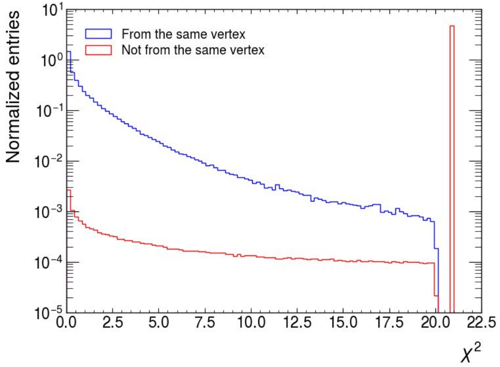


Utilizing χ^2 from Billoir fits of track-pairs

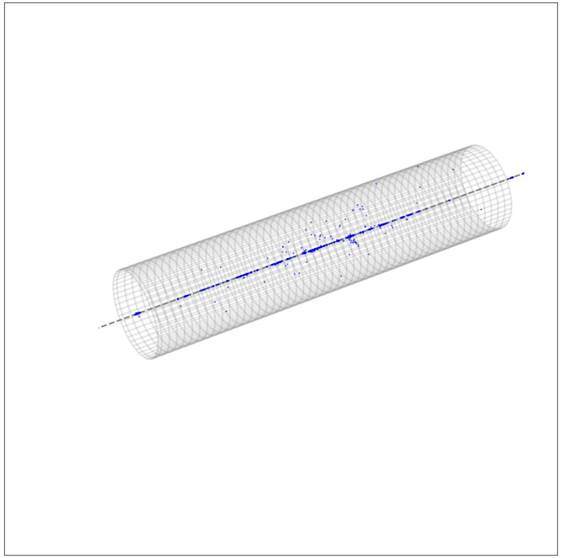
- Observation: If two tracks are from the same vertex, the fit always converges.

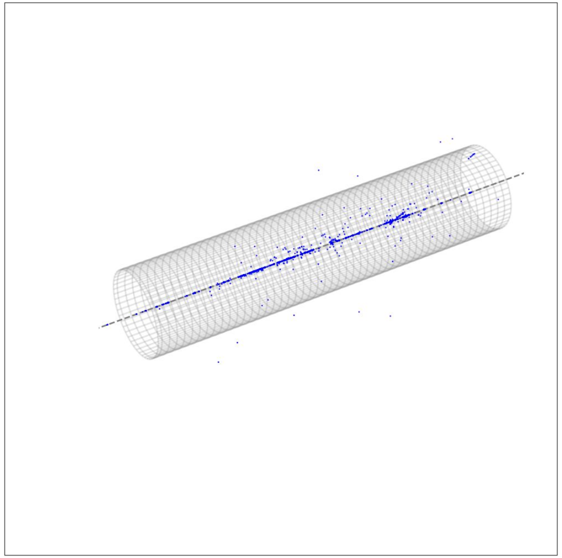
- Strategy: Create (overlapping) subsets of tracks.
  1. Start with an arbitrary track.
  2. Add all the tracks compatible with it to the set.
  3. Iterate over the tracks in set.
     - For each existing track, add all the compatible tracks.
  4. If any new track was added in step 3, repeat step 3.

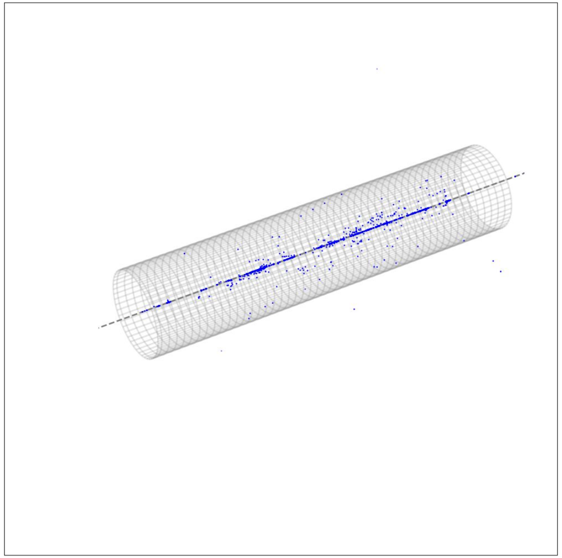
- These subsets can be taken as “bounding boxes”, which can also be overlapped.



Visualizing track-pair vertices in ODD



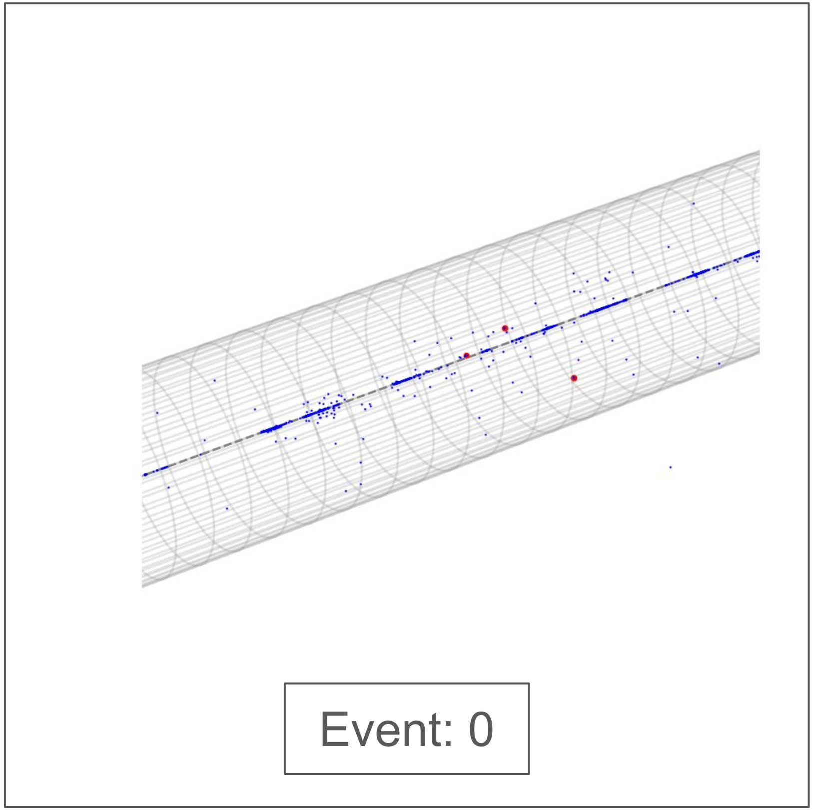


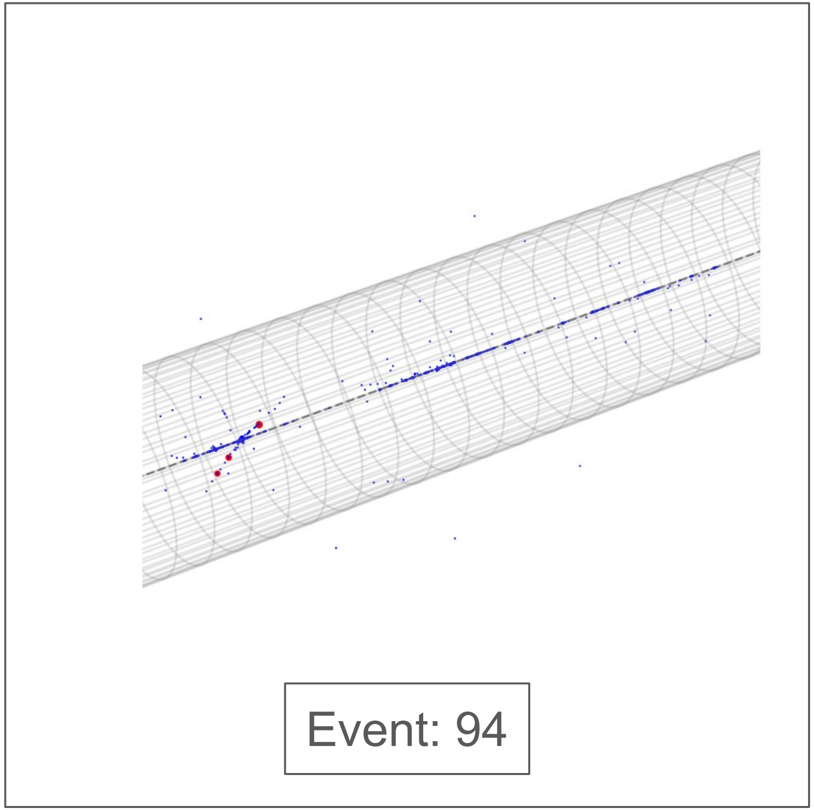


Finding multi-track vertices

- The baseline: Naive clustering solely based on position in 3D space.
  - Trivial to implement using `sklearn.cluster.DBSCAN`
    - `eps = 0.1`
      - Maximum distance between two samples to be considered as neighbours.
    - `min_samples = 1`
      - A "cluster" will be created even for a lonely two-track vertex.
  - Extremely fast.
- Visualizing the clusters:
  - Clusters with at least 2 two-track vertices has been shown in red on the next slide.
  - Transverse distance from the beam axis is required to be greater than 1 mm to eliminate the primary (including pile-up) vertices.

Visualizing vertices clusters in ODD





Performance metric for vertex finding

- Strategy for truth-matching:
  - ➢ At least 50% of the tracks from a truth vertex need to be associated with the reco vertex.
  - ➢ If multiple truth vertices match with a single reco vertex, the one with higher number of *common* tracks is chosen.
    - ■ If ambiguity still persists, the one closest to the cluster-centre is chosen.
- For measuring the performance of the clustering method:
  - ➢ Only the truth vertices with 3 or more tracks have been considered.
    - ■ The fraction of such vertices matched with a reco one is a metric for clustering.
- For measuring the performance of “Billoir fast vertex fit” itself:
  - ➢ Only the two-track truth vertices have been considered.
    - ■ For truth matching, **both** tracks are required to be associated with the reco one.

Framing vertex finding as a clustering problem

- Most obvious approach: A vertex is a cluster of tracks.
  - ➢ Consider the tracks as nodes.
    - ■ Track params are node features.
    - ■ Different quantities from track-pair Billoir fit can be taken as edge features.
  - ➢ This is the chosen strategy for MaskFormer as well as the previous primary-vertexing work.
- My proposal: A vertex is a cluster of one or more two-track vertices.
  - ➢ Each track-pair satisfying Billoir fit is a node.
    - ■ Different quantities from track-pair Billoir fit are node features.
    - ■ No edge feature. The Euclidean distance might be taken as one.
  - ➢ Immediately offers us a bounding box heuristic.
  - ➢ For two-track vertices, the performance is guaranteed to be at least as good as Billoir fit.

Next steps

- Calculate the performance of naive clustering (using DBSCAN), using the truth-matching strategy described before, to establish a baseline.
- Investigate the similarities between clustering of pixels (in an image) and track-pair vertices (from Billoir fits).
    - ➢ Can we use the lifted multicuts algorithm out of the box?
- Utilize hypergraphs to cluster tracks (without using any heuristics).
    - ➢ Known to be extremely parameter-efficient and fast as compared to attention-based models.
        - Example: HyPER
    - ➢ Each predicted hyperedge will correspond to a multi-track vertex.