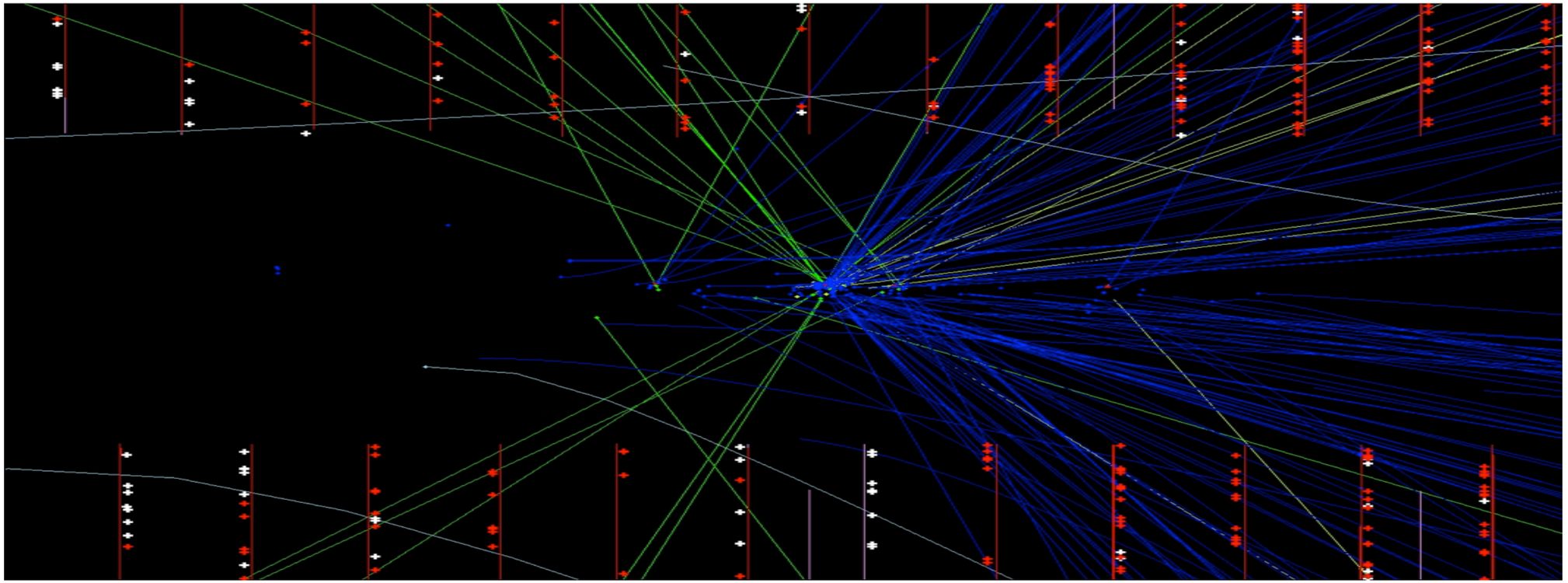
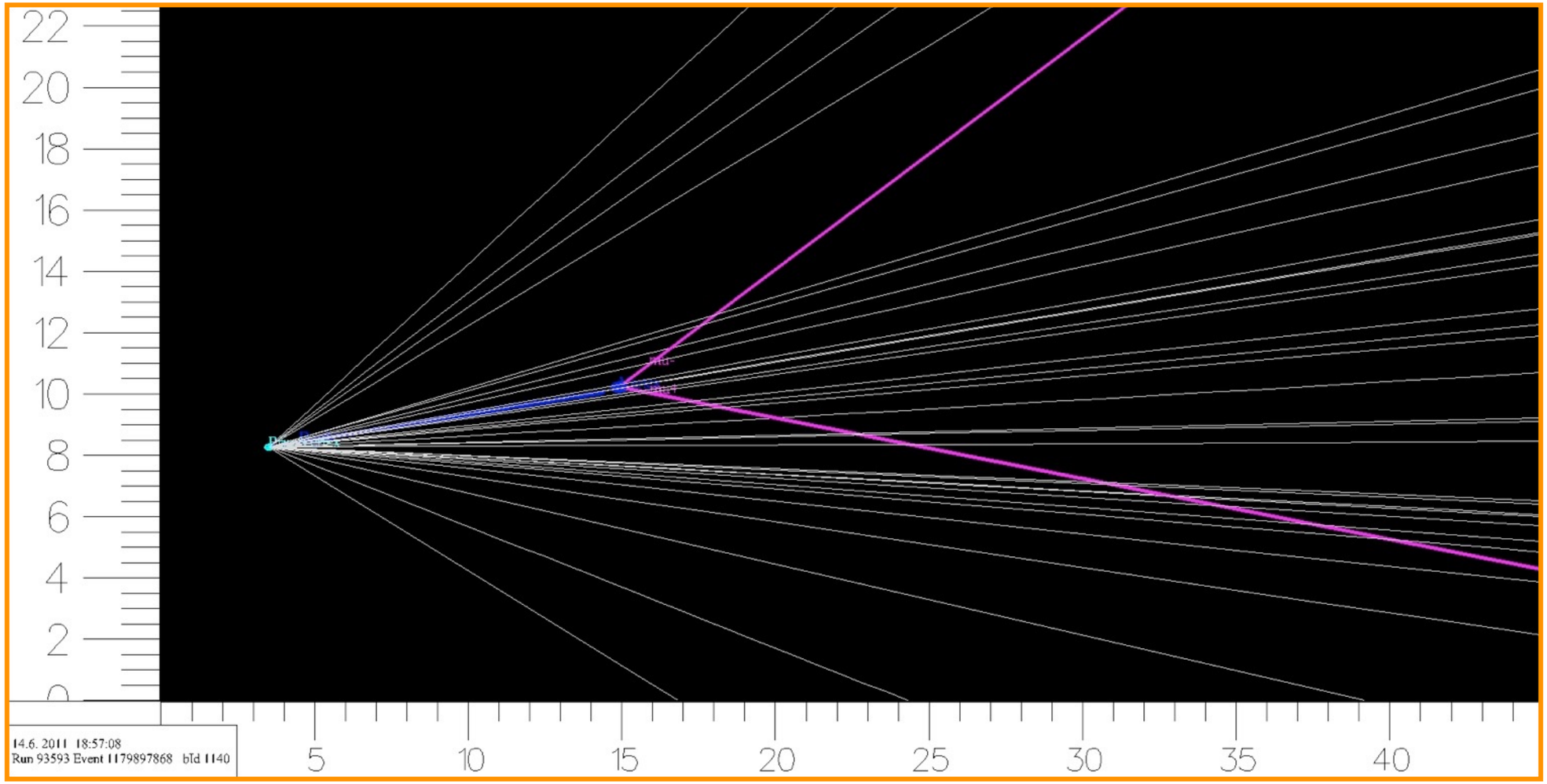
Why is vertex finding at 5 TB/s an interesting problem?



Vladimir V. Gligorov, CNRS/LPNHE

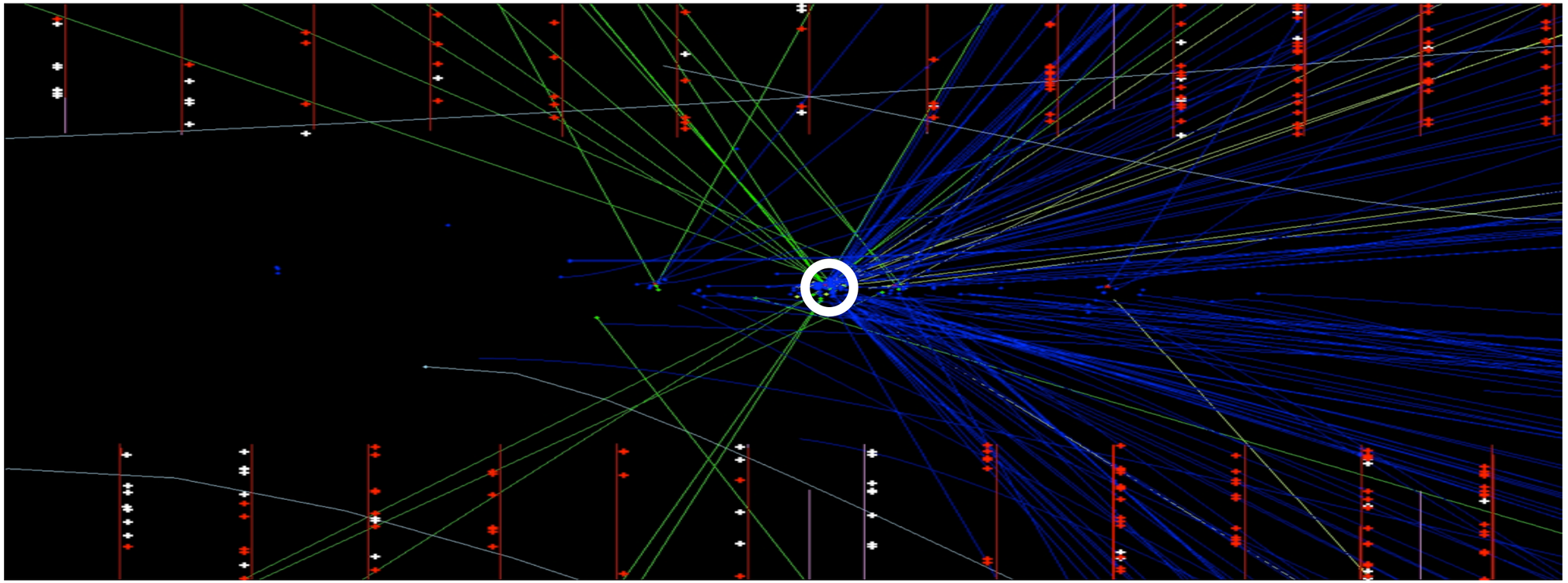
RAPID 2018, Dortmund, 19.11.2018

Why are we trying to find vertices?



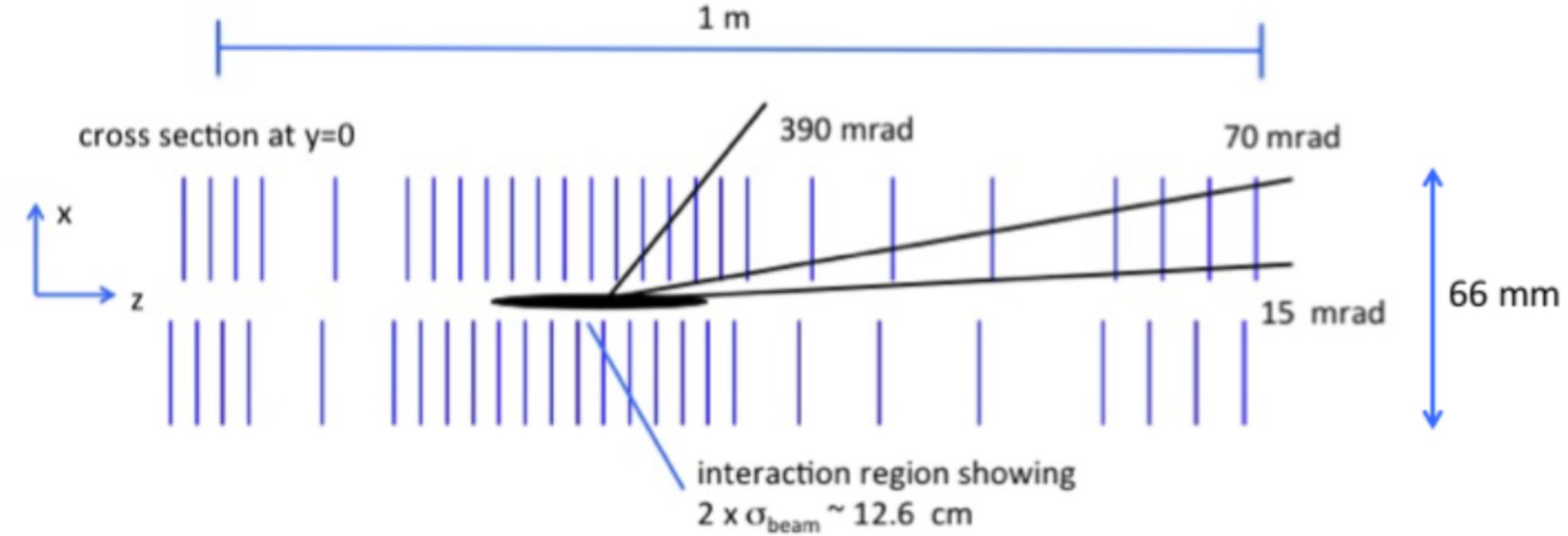
Important for measuring particle lifetimes, separating long-lived particles from short-lived ones

So do we need to find all vertices?



Why isn’t it enough to just find the primary vertex my signal came from? Good/open question…

Where are we trying to find vertices?

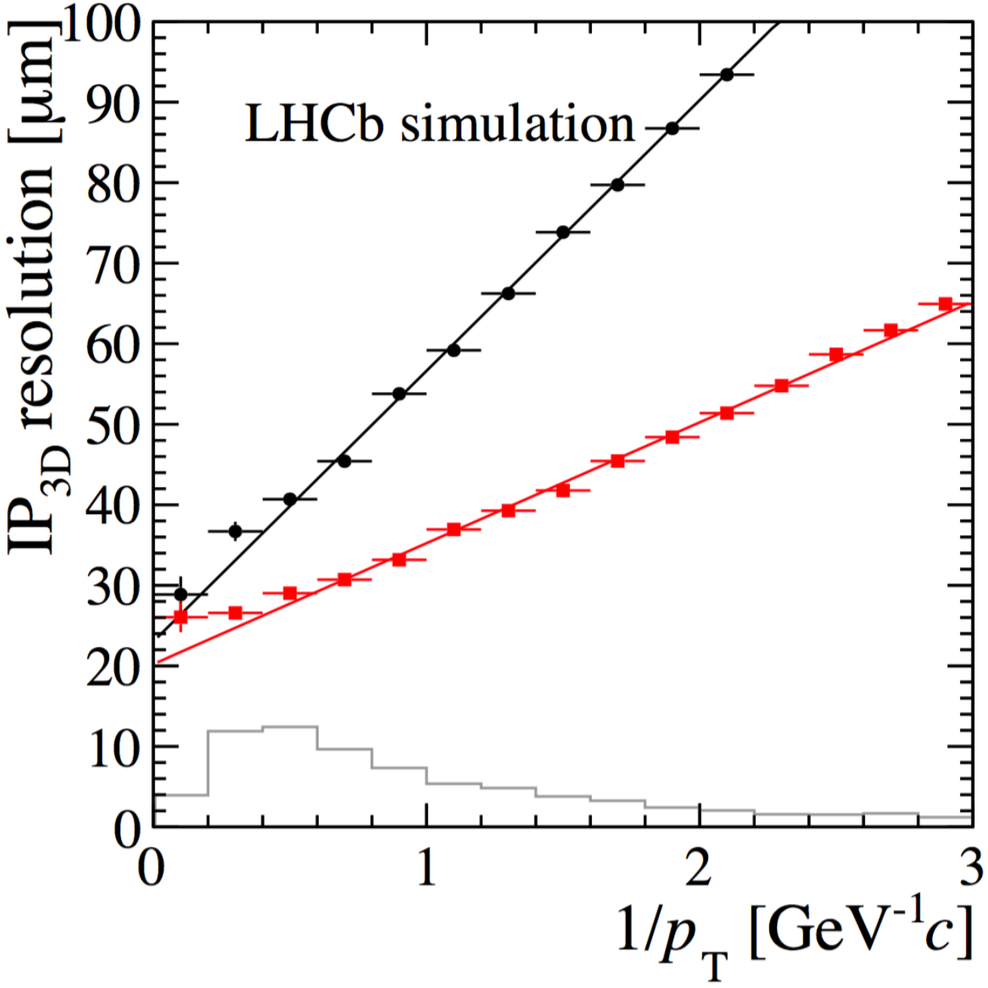


Characteristics of interaction region

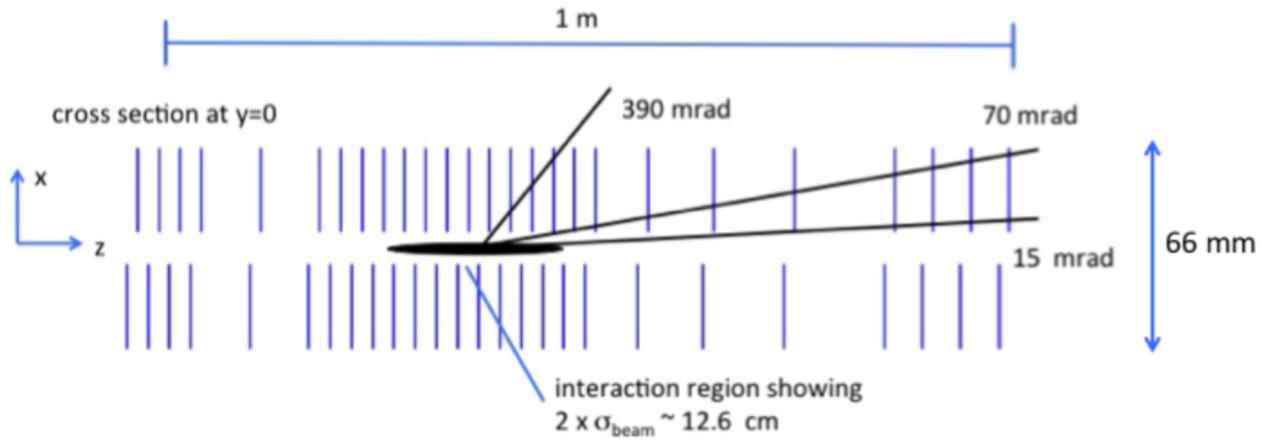
Spread of ~6 cm in z, tens of microns in x,y

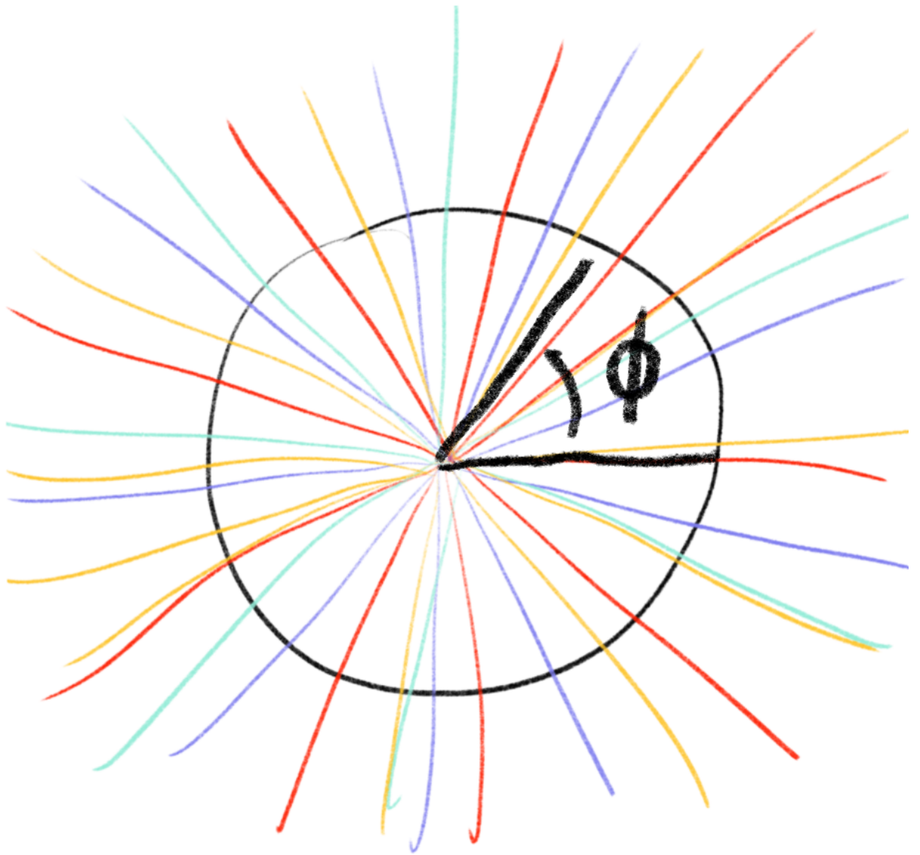
Average of ~6 pp collisions per event

Particle resolution on distance of closest approach to a point is on the order of tens of microns as well

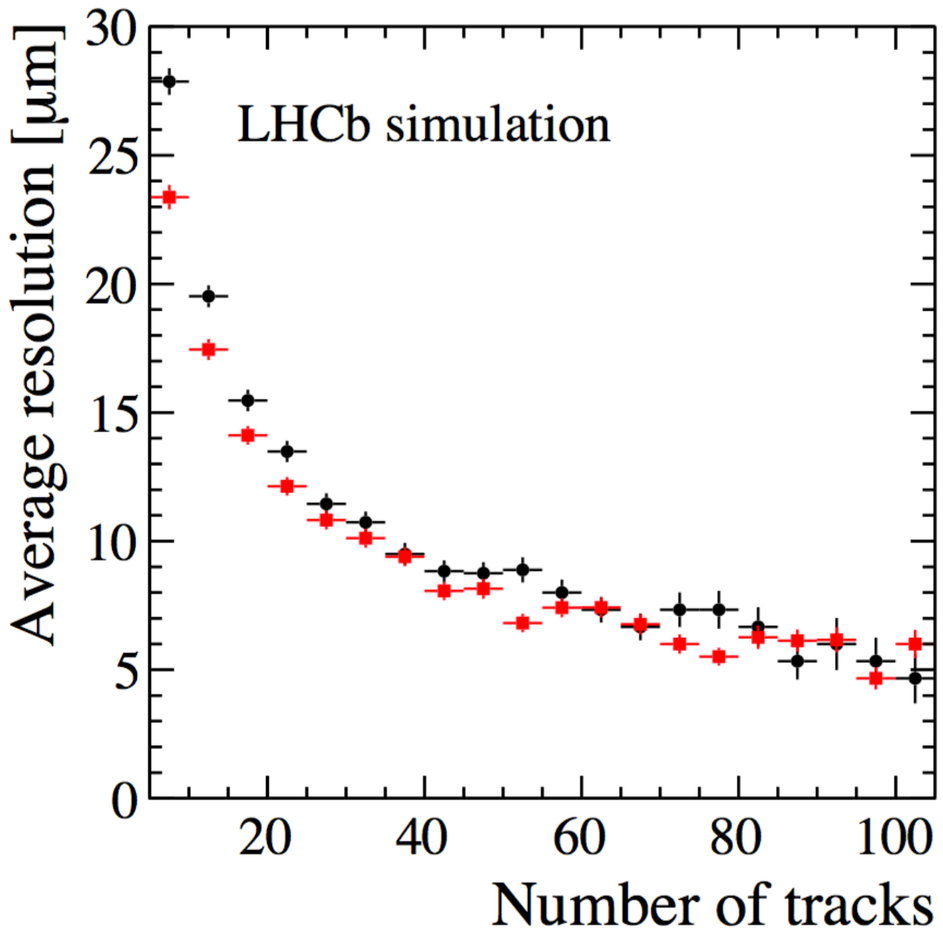


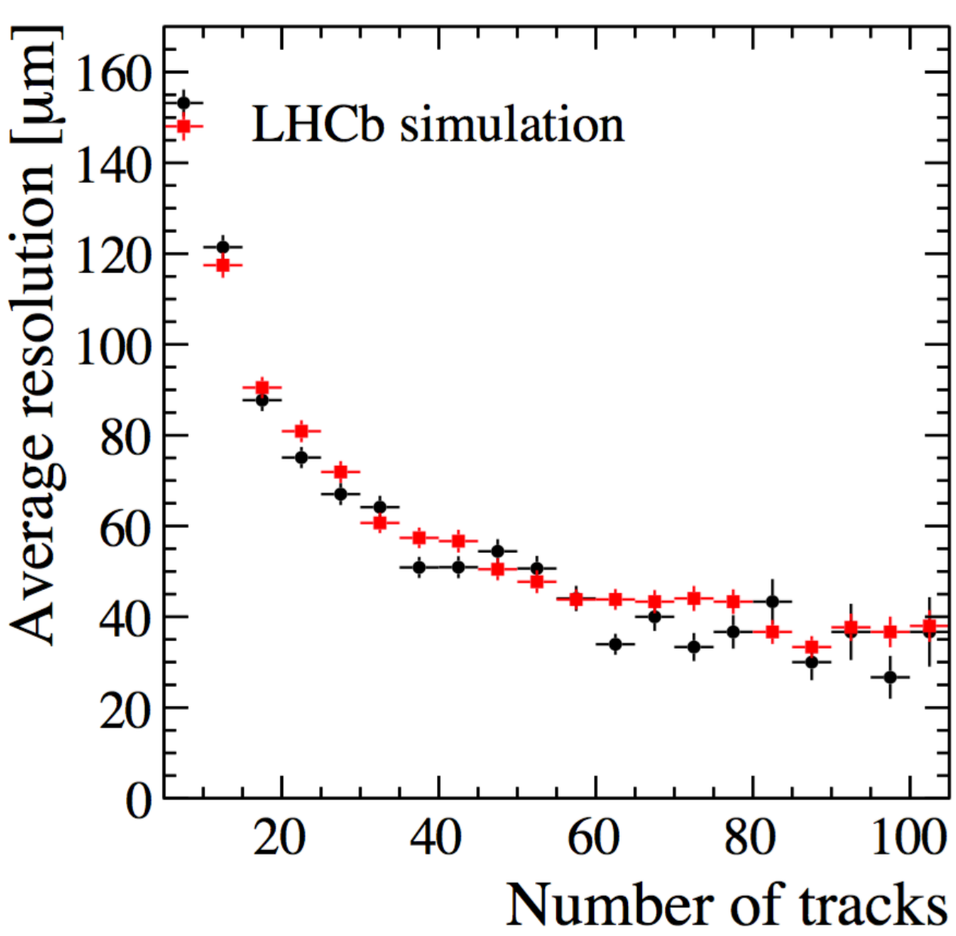
Symmetries of detector





Resolutions of “baseline” algorithm





Left is x, right is z. Sharp dependence on number of tracks in vertex

Some conceptual points to think about

Traditional algorithm associates tracks to a primary vertex

Important for doing the vertex fit and getting a reliable covariance matrix

However for many tracks the correct assignment might be ambiguous, especially if two primary vertices sit on top of each other. With a 6 cm interaction region this happens a non-negligible fraction of the time.

Could we have a more probabilistic assignment of tracks to primary vertices without spoiling the properties of the covariance matrix?

Allow ambiguous tracks to contribute to more than one vertex?

Associate only for resolutions, not for PV finding as such?

More data