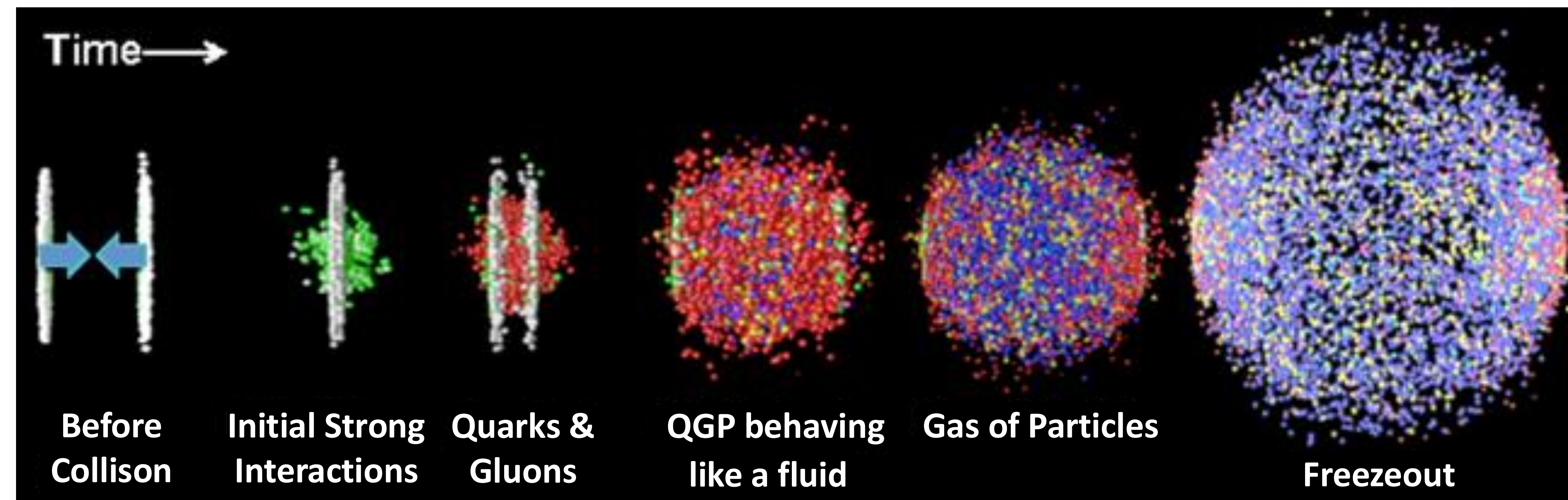


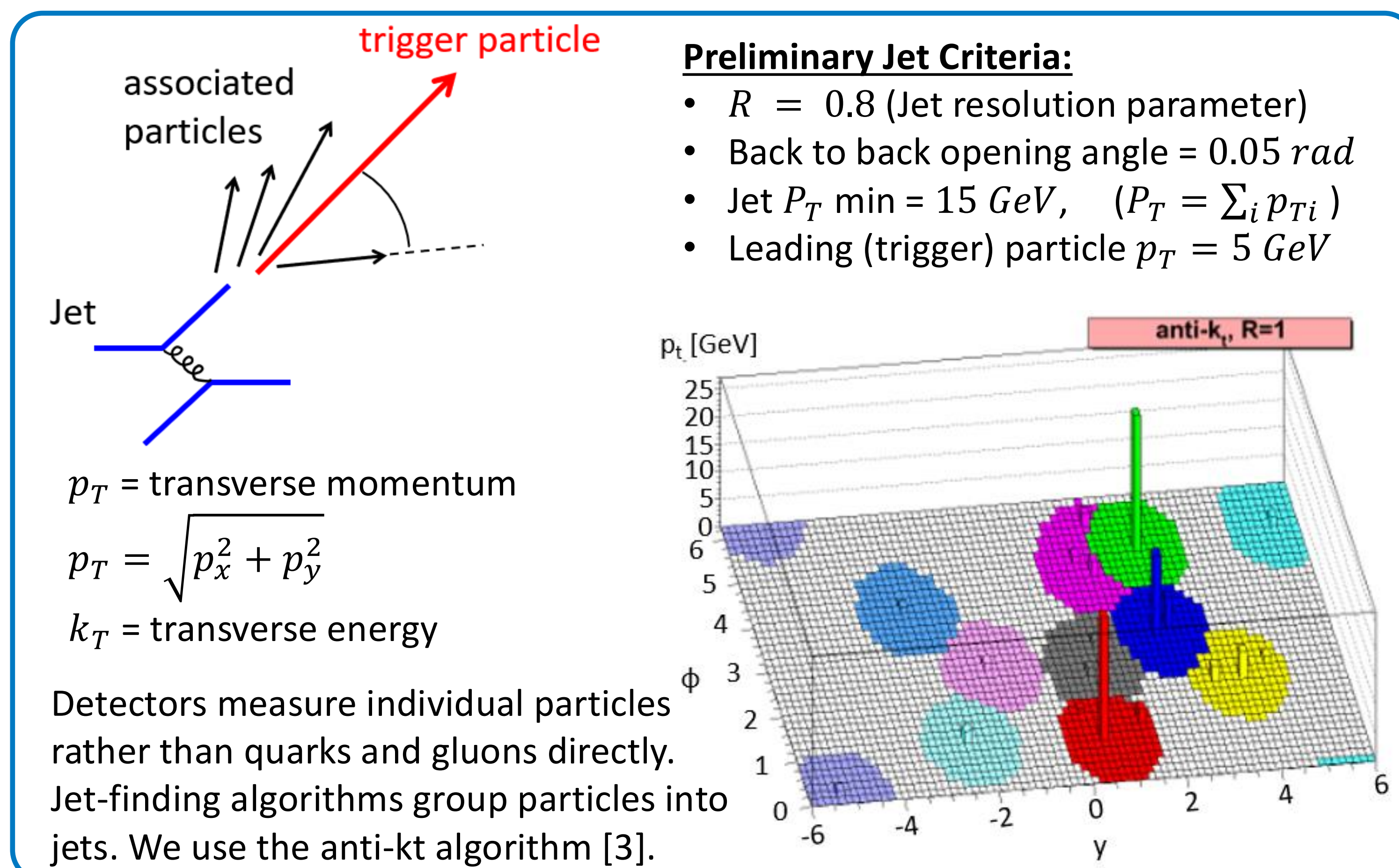
Abstract

Topological Data Analysis (TDA) and persistent homology are applied to relativistic proton–proton and nucleus–nucleus collision data to detect “jet” production from quark and gluon interactions. Quarks and gluons are the fundamental building blocks of the protons and neutrons that make atomic nuclei. The goal of relativistic nuclear collisions is to create and study a unique form of matter called quark-gluon plasma (QGP). TDA provides a framework for characterizing the geometric structure of high-dimensional data, and persistent homology quantifies features such as clusters and loops that remain distinctly identifiable in the data as a function of the distance between data points. In high-energy nuclear collisions, “jets” are sprays of particles emerging from high-momentum exchanges between quarks and gluons and they can provide insight into the properties of QGP. The primary objective of this project is to identify jets without relying on the traditional jet-finding algorithms. We generate simulated collision data with the PYTHIA Monte Carlo event generator [1] and analyze it using FastJet [2] and the anti-kt algorithm [3] to create training data. We use the Vietoris-Rips algorithm [4, 5] to build persistence diagrams from the particle transverse momenta. We plan to train machine learning (ML) models [6] to classify events using their topological features. To evaluate our TDA method, we will compare jet observables calculated with the training events to those of the TDA-ML classified events.

Relativistic Heavy Ion Collision

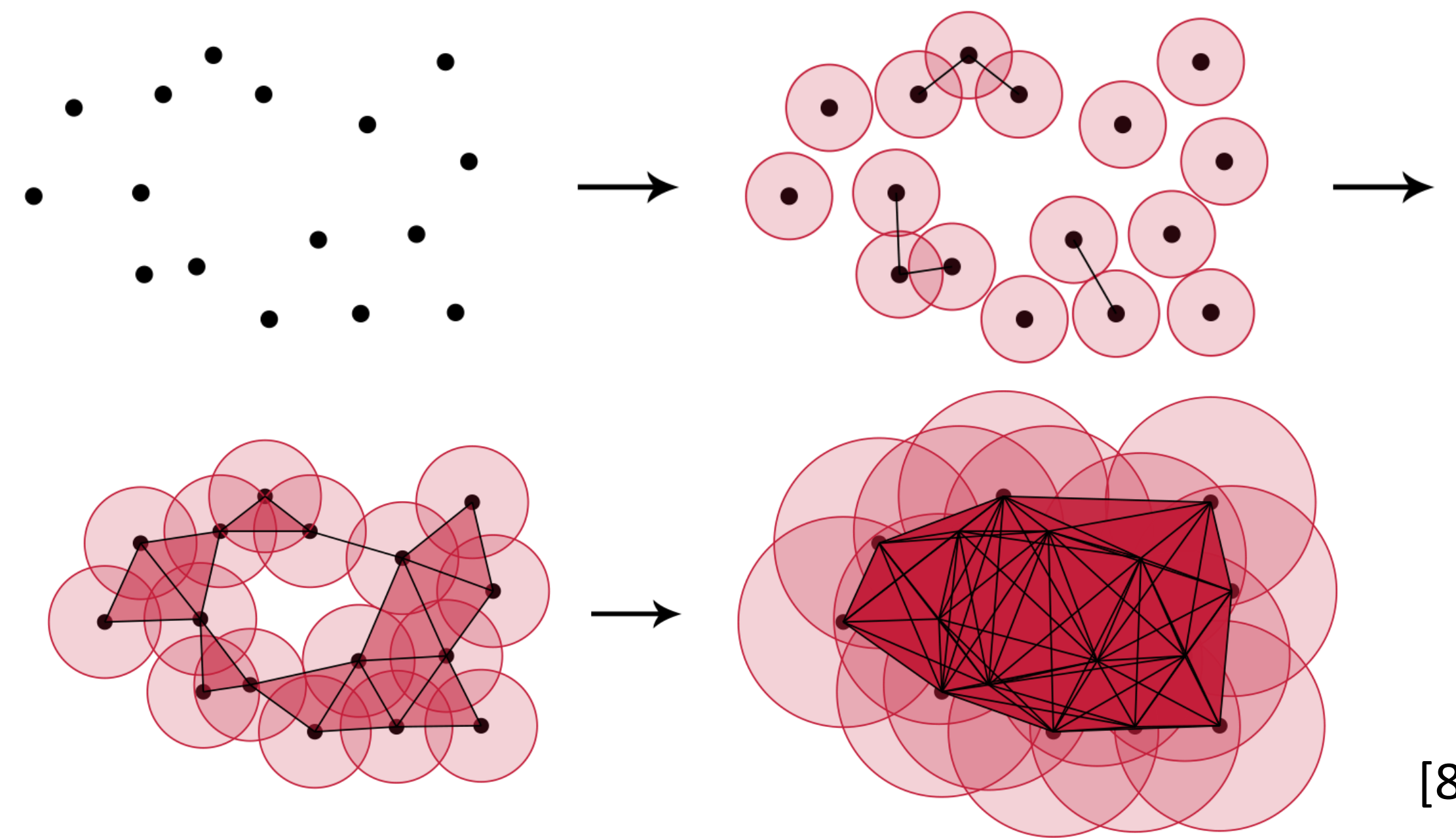


Jets



Persistence Homology

Vietoris-Rips is used to evaluate persistence homology [4]. Each point in the cloud represents a particle momentum. A growing “filtration radius” is applied around the points. When the radii intersect, edges are formed. A simplex is created when three edges form a triangle. A simplicial complex is when multiple simplexes and edges are connected.



Persistence Diagram

Features:

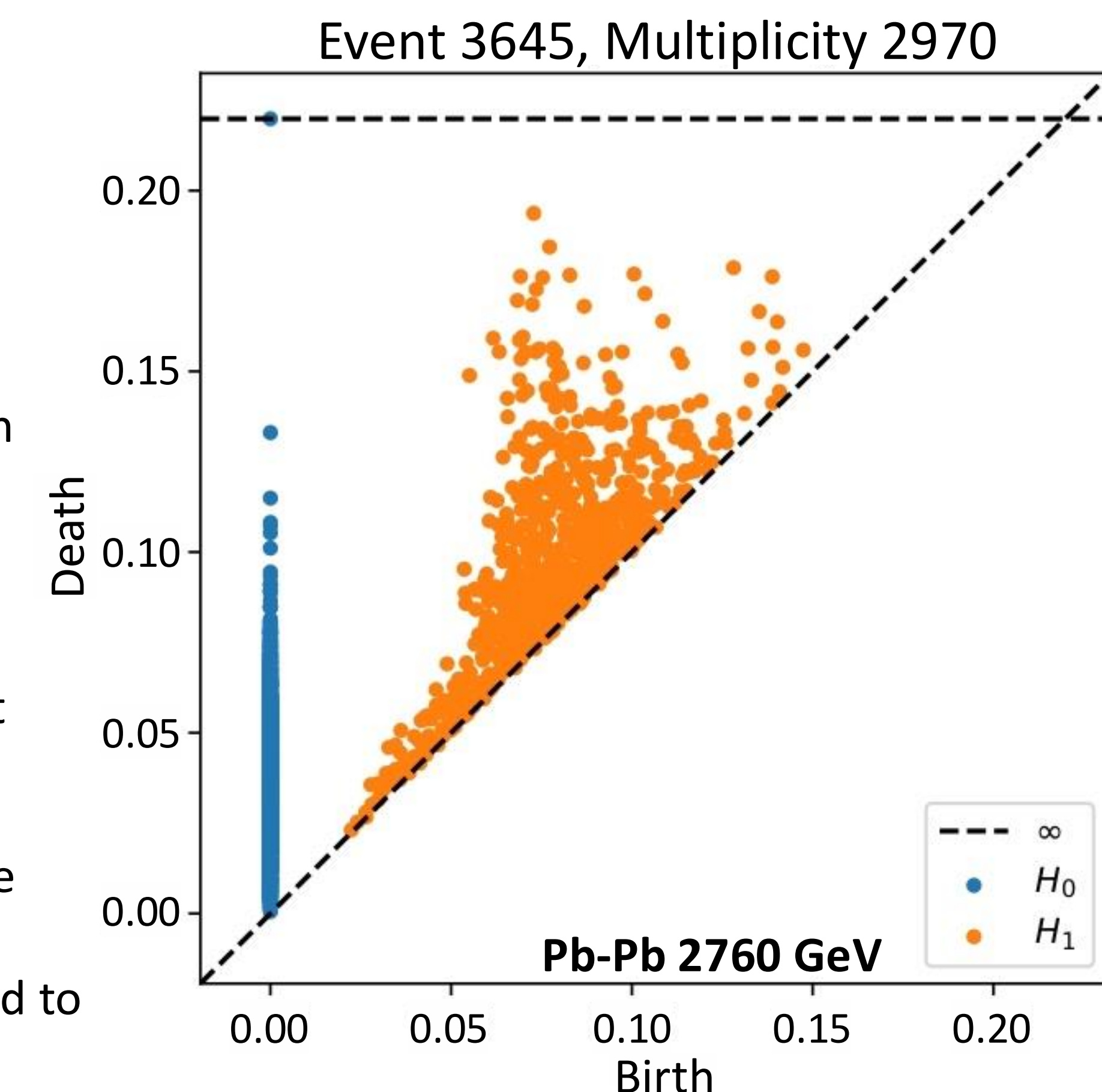
H0: Clusters
H1: Loops

Persistence: How long features “live”

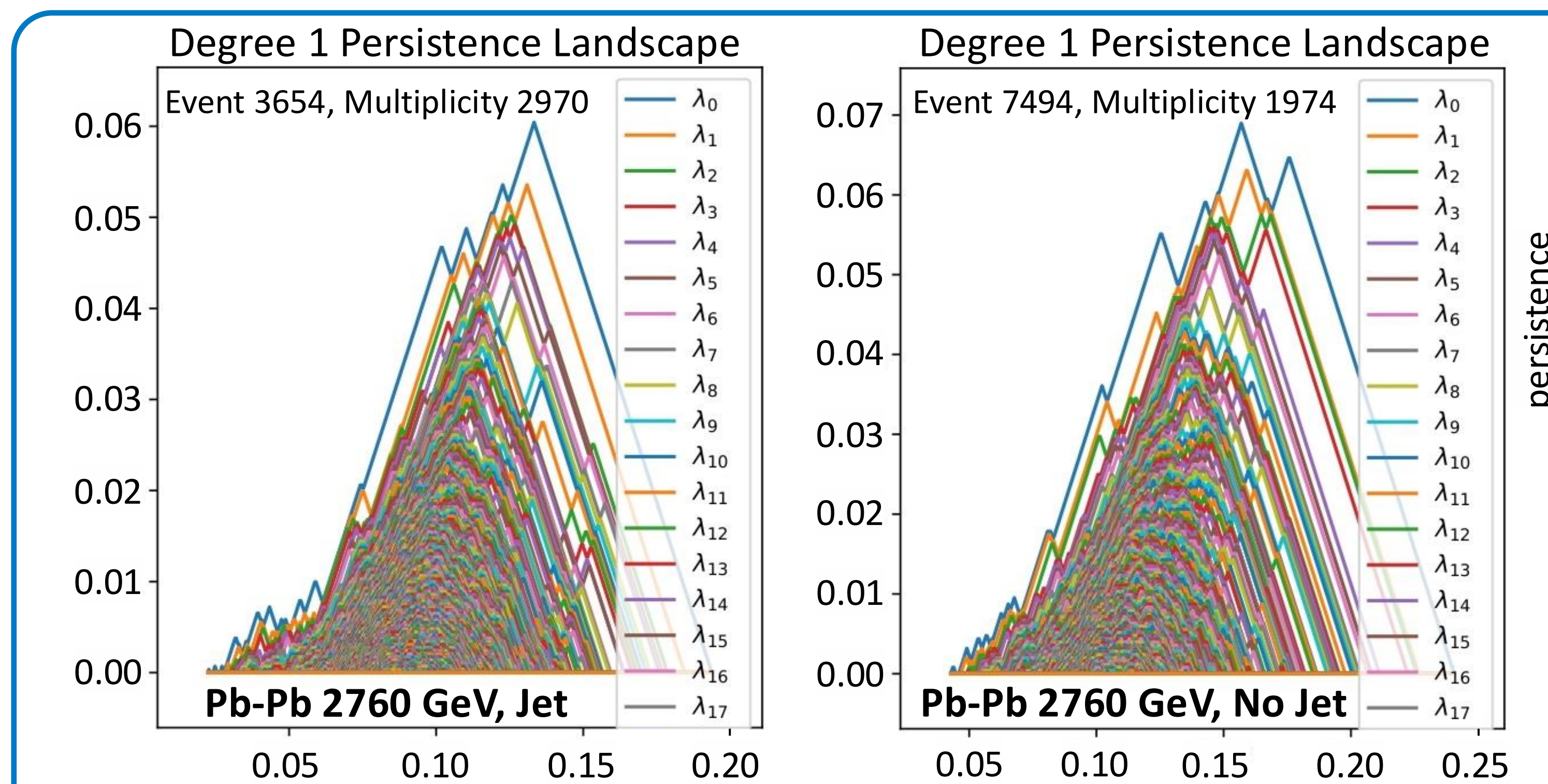
Birth: The value of the filtration radius when features form or change

Death: The value the filtration radius grows to before the next change

Persistence diagrams cannot be averaged directly. Persistence landscapes and images are used to compare diagram features.



Persistence Landscapes/Images



Persistence landscapes vectorize persistence diagrams to create features for machine learning [5]. Landscapes and images are computed for each event. Events are divided into jet and no-jet categories using anti-kt[3].

Goal: Train ML classifiers to identify jet events using images or landscapes.

Future: Test different ML classifiers and different experimental observable combinations.

Machine Learning

Preliminary Pipeline – Jet Event Classifier

Persistence Images → Standard Scalar → Support Vector Machine [6]

p-p 2760 GeV

	precision	recall	f1-score	support
no jets	1	1	1	276315
jets	0	0	0	157
accuracy				1 276472
macro avg	0.5	0.5	0.5	276472
wgtd avg	1	1	1	276472

Pb-Pb 2760 GeV

	precision	recall	f1-score	support
no jets	0.90	0.90	0.90	1671
jets	0.86	0.87	0.87	1271
accuracy			0.89	2942
macro avg	0.88	0.88	0.88	2942
wgtd avg	0.89	0.89	0.89	2942

Acknowledgements

Special thanks to Lawrence Technological University’s QUEST Program!

- [1] C. Bierlich et al., A comprehensive guide to the physics and usage of PYTHIA 8.3, SciPost Phys. Codeb. 2022, 8(2022), arXiv:2203.11601 [hep-ph].
- [2] M. Cacciari, G. P. Salam, and G. Soyez, FastJet User Manual, Eur. Phys. J. C 72, 1896 (2012), arXiv:1111.6097[hep-ph].
- [3] M. Cacciari, G. P. Salam, and G. Soyez, The anti-kt jet clustering algorithm, JHEP 04, 063, arXiv:0802.1189 [hep-ph].
- [4] U. Bauer, Ripser: efficient computation of Vietoris-Rips persistence barcodes, J. Appl. Comput. Topol. 5, 391(2021).
- [5] N. Saul and C. Tralie, Scikit-tda: Topological data analysis for python (2019).
- [6] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cour-napeau, M. Brucher, M. Perrot, and E. Duchesnay, Scikit-learn: Machine learning in Python, Journal of Machine Learning Research 12, 2825 (2011).
- [7] C. Tralie, N. Saul, and R. Bar-On, Ripser.py: A lean persistent homology library for python, Journal of Open Source Software 3, 925 (2018).
- [8] C. Bock, A gentle introduction to persistent homology, https://christian.bock.ml/posts/persistent_homology/ (2019).