DEVELOPMENT OF TOPOLOGICAL DATA ANALYSIS TECHNIQUES FOR NOISY AND SPARSE 2D TIME-DEPENDENT DATASETS

Jamison Hines[1], Saskia Mordijck[1], Sarah Day[2]

[1]Department of Physics, William & Mary, Williamsburg, VA 23185, USA

[2]Department of Mathematics, William & Mary, Williamsburg, VA 23185, USA

April 3, 2026

Acknowledgements

I would like to thank multiple individuals for their support in this research. I am thankful to Professor Saskia Mordijck and Professor Sarah Day for providing insight and opportunities to build my knowledge and skills in pursuing this project. A share of gratitude goes to the William & Mary Plasma Physics Group for their feedback and suggestions. I am grateful to conduct research alongside Michael Campagna and Luke Payne for their coordination of the application portions and data retrieval. Lastly, a special thanks goes to Sage Stanish and Julius Kiewel for building the foundation of this research, currently and beyond.

Abstract

This project leverages Topological Data Analysis (TDA) to identify turbulence transitions on the Large Plasma Device (LAPD), using a high-speed visible camera dataset. TDA is an analysis technique that extracts topological features from digital data. Plasmas are nonlinear systems that evolve through instabilities or transitions to new equilibria, so TDA methods support predictions of plasma turbulence, which are relevant to solar and atmospheric phenomena. A TDA algorithm is developed with reliance on cubical persistent homology and statistical metrics. Noise is added to digital data to recover lost topological information, and noise alleviators are allocated to enhance noise resistance for applied images. Image processing, such as foreground detection, visualizes shifts in digital data to identify transitional observations in conjunction with TDA results. The TDA findings display topological transitions that indicate possible transitional occurrences in magnetically confined plasma. Future work includes the reduction of spatial resolution for TDA confirmation and the extension of these techniques toward datasets in solar investigations to yield more precise transitions in scientific experiments.

Keywords: Topological data analysis, turbulent transitions, digital data, magnetically confined plasma

1. Introduction

This research seeks to develop Topological Data Analysis (TDA) techniques for two-dimensional, time-dependent datasets with varying levels of noise and sparsity. These techniques are developed for dataset applications by extracting information from digital data for synthetic analysis. This is directly relevant to understanding plasmas, planetary atmospheres, and oceanic flows, as all are non-linear dynamical systems that can undergo unexpected changes due to instabilities or transitions to new equilibria. Although detecting and tracking systems undergoing transitions are an essential focus for scientists hoping to understand such systems, indicating the occurrence of transitions is an obstacle in a variety of systems for empirical research and databases. Likewise, digital data deals with noise and sparsity, excessive variables that result in invalidity when interpreting scientific findings. The research builds on recent foundations, including the counting of holes in physical systems using computational homology and resistive drift-wave turbulence (Stanish, 2022) and the identification of transitions via topological data analysis in plasma with noisy turbulence (Kiewel, 2024).

The major challenge is to find precise occurrences of plasma turbulent transitions in camera datasets while mitigating noise and sparsity, which presents a greater challenge when analyzing the TDA findings. Therefore, the research aims to develop TDA methods to identify potential transitional events in magnetically confined plasmas. The methods revolve around homological feature extraction, cubical persistent homology, noise/sparsity alleviators, statistical topology metrics, and image processing methods such as foreground detection. There is a review of the TDA methods with multidimensional datasets corresponding to magnetically confined plasma and nonlinear transitions, primarily extracted from the Large Plasma Device (LAPD) camera dataset at high resolution and greyscale levels. Interpreting the plasma turbulent behaviors from the camera dataset application brings engineering solutions for solving turbulence causations in magnetically confined plasma experiments.

2. Topological Data Analysis Methodology

TDA is a mathematical tool based on topology for extracting topological features from point clouds (discrete sets of points in Cartesian coordinates) and other digital data, thereby enabling the identification of transitions in two-dimensional datasets (Kiewel, 2024). The tool is gaining popularity in applied mathematics and modeling for its potential to extract topological and geometric information for further data analysis. The GUDHI library is an open-source library utilized for topological data analysis and interpretation of higher-dimensional datasets (The GUDHI Project, 2025).

One of the primary tools under TDA is known as persistent homology (PH), a method to define and compute qualitative features from datasets based on characteristics from geometry and topology (Otter, 2017). PH characterizes the features as Betti numbers that represent the connectivity of sets of points, lines, and their n-dimensional counterparts in higher dimensions. For points in a nested sequence of space, notated as X ⊂ R^2, Betti numbers depict the number of components (β0), the number of 1-dimensional holes (β1), and voids scripted as holes in multidimensional spaces. For application purposes, the Betti numbers are noted as β0, the number of components/objects of the sets, and β1, the number of 1-dimensional holes in the nested spaces.

Figure 1: Two topological shapes with gray structures representing a set with β0 = 0 (left) and β1 = 1 (right). Figure from Kiewel (2022).

The digital data is viewed as a grayscale image as f : P → {0, 1, ..., 255}, where f(x) provides the value of pixel x. The greyscale images are two-dimensional and structured cubically, providing leverage to the cubical complexes for further filtration in PH. In terms of topological structures, cubical complexes are topological spaces whose simplices include vertices, edges, squares, cubes, and higher-dimensional hypercubes (Otter, 2017).

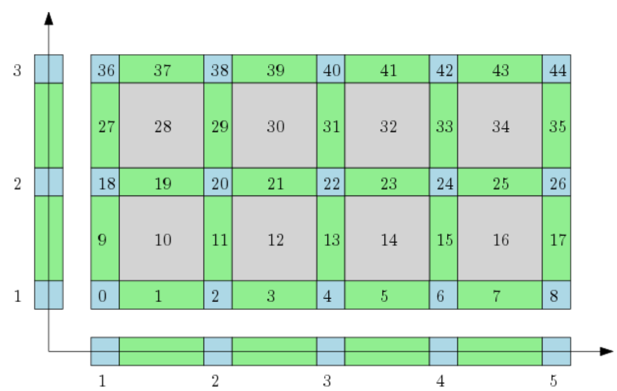

Figure 2: Image displaying the product of one-dimensional cubical complexes. The quantity of every cube in each direction is equivalent to 2n + 1, where n is the number of maximal cubes in the considered direction. Figure from the GUDHI Project (2025).

Each pixel in the 2D digital image is allocated to a vertex, and adjacent vertices are linked together as an edge, and the remaining squares are filled in for the image. Vertices are then named as the integer of the pixel value, while edges are named based on the maximum value of adjacent vertices. The cubical complex combines the labels into nested sequences of cubical complexes C for C ∈ {0, 1, …, 256}, where each C holds all characteristics if the integer values are less than or equal to I for I ∈ {0, 1, …, 256}; this complex process is known as a filtered cubical complex.

PH is used next to determine the structure of the greyscale data based on the extracted features. A distance parameter ϵ is defined as a threshold of the evolving topological structure. As

ϵ increases continuously to the highest possible value, the topological structure alters in the sublevel sets where the greyscale data is stored. Also known as sublevel set filtration, the sublevel set is formed as $ft := \{x \in P \mid f(x) \leq t\}$, and the moving threshold results in stationary points over the nested sets. The local maxima reveal features known as Birth and Death rates: birth rates occur when a Betti feature first emerges in the set, whereas death rates occur when two Betti features appear consecutively, with the lower-lifespan feature removed.

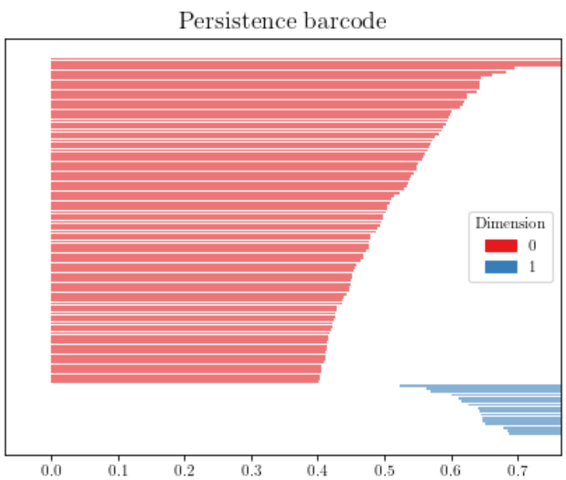

Figure 3: The persistence barcode from a list of collected persistence values, representing a diagram in a single homology dimension or a range of homology dimensions. Figure from the GUDHI Project (2025).

Once the filtration is complete, a persistent diagram depicts the Betti features remaining in the sublevel set (Fugacci, 2016). This diagram is an extensive plotting system that calculates the difference in the homology of the sub-level homology sets (Cohen-Steiner, 2005). Every point in the given set relates to a feature and quantifies its importance by the absolute difference between the point's two coordinates. The farther the feature is from the threshold defined as the line y=x, the longer the lifespan of the feature is in the diagram. If the feature is located at the infinite line, these features are the longest features to remain in the filtration process.

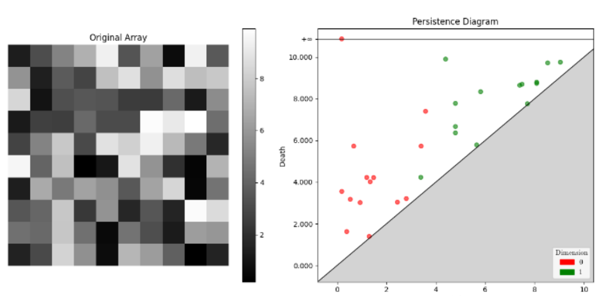

Figure 4: Two figures that display cubical persistent homology. The left image is a 10x10 greyscale figure with randomized pixel values in the range ∈ {0, 10}. The diagram to the right represents the lasting features for the number of components (Dimension = 0) and number of holes (Dimension=1) in the image.

Multidimensional datasets are large and detailed for human consumption, and quantity errors and discretization obstacles add unnecessary data complexity when related to any acquisition process (Cohen-Steiner, 2005).

2.1 Noise & Sparsity Alleviators

TDA processes the digital data to interpret the topological nature of the data and possible transitions. In contrast, the presence of noise and sparsity hinders the retrieval of the necessary data, which disrupts the interpretation behind TDA. To counter these obstacles, algorithms are introduced to supervise the removal of noise and sparsity. Gaussian Blur and Opening/Closing algorithms are primarily utilized to mitigate noise at various scales. Gaussian Blur computes the weighted average of the neighboring values for pixels based on the proportional size (kernel) and thus averages and smooths out noise throughout the image. The proportional scale is ~10 for the Gaussian Blur. The Opening/Closing algorithm utilizes morphological operations to remove pixel noise and restore noisy images.

Figure 5: Images with morphological operations. The original image is compared with morphologically affected images, correlated with the process behind the Opening/Closing Algorithm. The kernel size for each operation is ~10.

Opening is the process of eroding and then dilating the image, and closing is the process of dilating followed by erosion of the image. Eroding selects the minimum pixel value with respect to the kernel size, while dilating chooses the maximum pixel value in the neighborhood of pixels. The proportional scaling for the Opening/Closing algorithm is also ~10. Before undergoing the noise removal methods for the research data, the digital data resolution is first altered to only concentrate on the visible plasma. This is noted as altered data resolution for the applied image set. The alteration directly incorporates the necessary pixels in the digital data while reducing the processing time in the TDA algorithm.

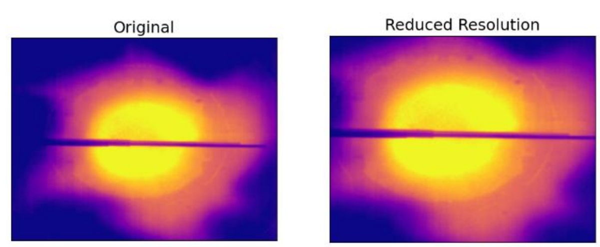

Figure 6: The altered resolution of the digital data. This is a cropping of each boundary to only focus on the plasma turbulent behavior and edge interactions.

Gaussian noise is next applied to the digital data, as this perturbation can clean the data and recover the underlying topology independent from the data dimensionality (Adler, 2013). Gaussian noise is introduced using the defined continuous Gaussian distribution:

$$f = \frac{1}{(2\pi\sigma)^2} e^{-\frac{(x-\bar{x})^T(x-\bar{x})}{2\sigma}} \quad (1)$$

The substantial addition of noise is effective enough for sample points to appear and uncover extraneous homology elements. Gaussian noise does not lead to unnecessary elements with proper sample sizes, and noise outliers thus do not interrupt homology recovery of the image data. If a substantial proportion of additional noise is added to the greyscale images, then sample points are visualized throughout the images, revealing unnecessary homological elements. The amount of noise scaling to noise-removal scaling ratio is therefore crucial to reveal the necessary image data for proper TDA production. The two processes are to ensure the data is not sparse before accessing the noise removal algorithms.

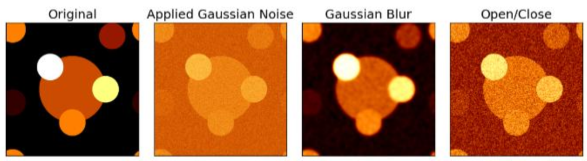

Figure 7: The figure contains four images, each with noise properties. The left image is the original with 2D uniform circles and various intensities between 0 and 1. The other images consist of an applied Gaussian noise of ~1 proportional scale, followed by independent noise removals. Gaussian Blur is applied to the second image with ~10 proportional scaling, and Opening/Closing is applied to the second image with ~10 proportional scaling. These applied proportional sizes are the same for the TDA applications.

When processing the digital data through these alleviators and then the TDA, the developed PDs are known as processed PDs in this case.

2.2 Topological Metrics

To determine the topological difference between the original and processed PDs, two topological metrics are utilized; These are Full Wasserstein Distance (WD) and Bottleneck Distance.

The Full WD metric is defined as the following:

$$W_p(\mu, v) = \left(\underbrace{inf}_{\gamma \in \Gamma(\mu, v)} \int_{X \times X} \|x - y\|^p d\gamma(x, y) \right)^{\frac{1}{p}} \quad (2)$$

Full WD is a metric that tracks the minimum distance of Betti features between original and processed PDs. The characteristics of the Full WD incorporate the geometry of the hidden space X, where X and Y are degenerate at points x, y ∈ X, and therefore W(X, Y) is correlated to the distance between x and y in space X (Panaretos, 2018). This concept allows WD to locate persistent pairs above the persistence threshold and depict the similarity between images. The metric operates with complex topological stability, but there is more susceptibility to small noise because the metric relies on the number of persistent pairs appearing from noise.

When given two persistence diagrams, the bottleneck distance between the plots is defined to be

$$W_\infty(X, Y) = \underbrace{inf}_{\eta: X \to Y} \underbrace{sup}_{x \epsilon X} \|x - \eta(x)\|_\infty \quad (3)$$

Where η is the range of all bijections (Kerber, 2017). The Bottleneck metric calculates the longest edge length, and in this case, the maximum lifespan of Betti features between original and processed PDs. Bottleneck uses a simplified calculation of the matching pairs, which leads to more topological stability and less sensitivity to noise. By measuring the absolute scaled distance for the homological features represented in the image datasets, the plots containing topological distances provide analysis for the metrics and topological information embedded in the digital data.

2.3 Foreground Detection

Observing a visual difference in the original frames poses an issue because there is no consistency in finding the transitional differences. The research uses foreground detection, an image processing tactic to detect objects in motion, and extracts the information to the foreground of the new image. Foreground detection is applied on each transitional frame to visualize the occurrence of turbulent transitions between plasma images in the LAPD Camera Dataset. To view the effect of the foreground detection, background subtraction removes the background by applying a pixel threshold, and a threshold of ~3 is used in this project. The pixels quantitatively above this threshold are labeled as the foreground, while pixels quantitatively under the threshold are labeled as the background. A foreground mask is then inserted as a detection mask to assign white pixels to the foreground (plasma transitions) and black pixels to the background (extraneous content).

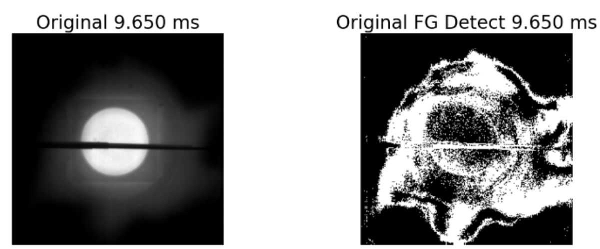

Figure 8: An image of the plasma from the LAPD Camera Data Run ID 69, particle transport shot 104 from Time 3750800907 in November 2022. On the left is the original image of the plasma before the second clear transition recorded by the topological metrics at ~9.65ms. On the right is the image with foreground (FG) detection applied to contrast the transitioning regions with the other sources. The white regions are where the plasma has edge interactions with the device, thus depicting possible regions to interpret signs of turbulent transitions.

The image processor is applied for clearer visualization of the detected foreground in the results. These visualizations are shown on transitional images, images before and after each topological transition on the metric data.

3. Fast Camera Plasma Measurements

Plasma is the fourth state of matter with non-linear properties and dynamics, and the key contributor is turbulence that occurs naturally and spontaneously. This phenomenon generates prominent instabilities, resulting in high variations of plasma behavior and therefore motivates the discovery of new modification strategies. Recently, there have been reviews on flow shear and magnetic shear to supervise transport barriers and limit turbulent behaviors. Transport barriers are regions where plasma turbulence is suppressed or quenched, and these barriers are frequently produced and maintained in tokamaks for low turbulent transport of confined plasma (Garbet, 2006). Flow shear and magnetic shear are essential in stimulating formation and sustainment of these barriers, and regimes are defined by a critical heating power that should be minimized. Although there is documentation of turbulent transport and flow shear analysis, there is low predictability behind the occurrences of turbulent transport for confined plasma. Understanding these occurrences leads to engineering solutions for the suppression of plasma turbulence and thus contributes to the development of the discussed techniques.

The TDA methods are leveraged to identify plasma turbulence transitions on the Large Plasma Device (LAPD), a high-speed visible camera dataset of magnetically confined plasma. The camera captures plasma frame-by-frame. Images are sized at 256 pixels by 256 pixels, and the camera frame rate is ~44,000 frames per second for each data run. The camera data originated from experiments in November 2022, which captured details of plasma interactions, contributing to localized basic plasma physics at the University of California (UCLA) (Gekelman, 2016).

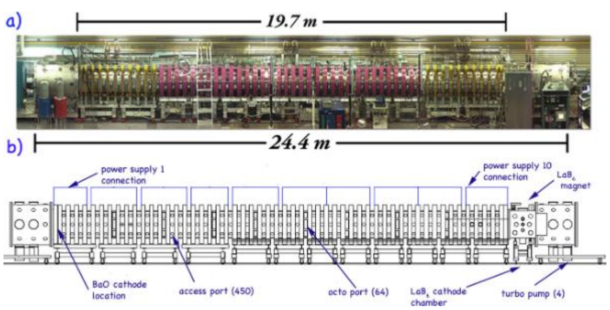

Figure 9: (a) A physical layout and design of the LAPD device. On the left is a pump-out chamber that contains the BaO cathode and two 2200 l/s turbo-molecular pumps. The magnets and supports are attached throughout the device, and the chamber containing the LaB_6 cathode is shown to the right. (b) Machine drawing showing two types of ports and connections to the power supplies. Yellow magnets are from the CCT tokamak at UCLA, and purple magnets were constructed at UCLA. An additional magnet delivers a magnetic field at the second LaB_6 cathode. Figure from Gekelman (Gekelman, 2016).

4. TDA Application Results

The LAPD camera dataset consists of images stacked over a time series, projecting the progression of plasma at a reference point. Each image is processed through the TDA methods in the following order: homology feature extraction, sublevel set filtration, cubical persistent homology, altered resolution of digital data, addition of Gaussian noise, noise removers to balance noise-scale ratios, computing of statistical metrics for comparison of the original image and processed image, and visualization of major digital transitions using foreground detection. The images from each data run are converted into a time series based on the experimental frame rate and total number of frames per data run. LAPD data run 65, data run 69, and data run 73 are studied and represented in the following results.

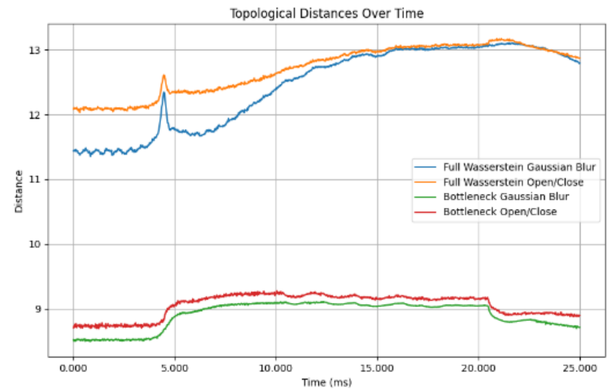

Figure 10: Topological distances over the time series from the LAPD camera data run ID 65, particle transport shot 104 from time 3750800907 in November 2022. Clear transitions are recorded at ~4.5ms, ~15ms, and ~21ms, based on the topological structure of the given images, suggesting plasma transition regions within the device.

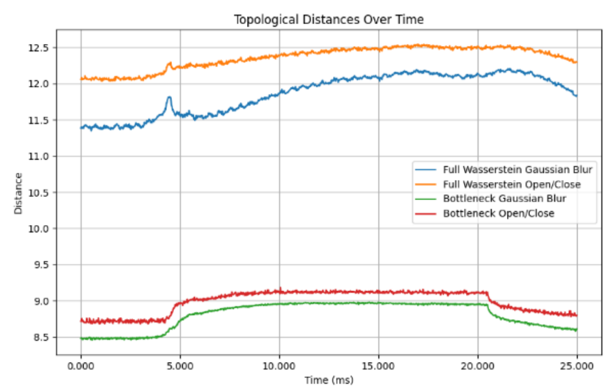

Figure 11: Topological distances over the time series from the LAPD camera data run ID 69, particle transport shot 104 from time 3750867143 in November 2022. Clear transitions are recorded at ~4.5ms, ~15ms, and ~21ms, based on the topological structure of the given images, suggesting plasma transition regions within the device.

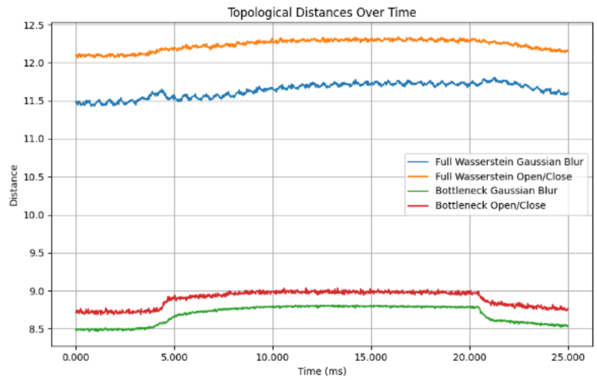

Figure 12: Topological distances over the time series from the LAPD camera data run ID 73, particle transport shot 104 from time 3750878858 in November 2022. Clear transitions are recorded at ~4.5ms and ~21ms, based on the topological structure of the given images, suggesting plasma transition regions within the device.

5. Discussion

In the topological metric plots, the metric value at each time is the distance between the original and processed image. The metric is labeled based on which noise algorithm is utilized for the specified result. Each distance independently intersects with Gaussian Blur and Opening/Closing for comparisons of results between the TDA techniques. While the bottleneck metric has less susceptibility to small noise fluctuations, the metric only calculates the maximum distance of the feature lifespan, meaning the results have fewer calculations of homological features and lower scaled values in comparison to the Wasserstein metric. Opening/Closing removes noise more precisely in each kernel location, whereas Gaussian blur removes noise at a macroscale. This means that Gaussian blur introduces less digital complexity in the images and therefore reduces image disparity, which is why Gaussian blur has evidently lower scaled values compared to Opening/Closing operations. The shift of plasma behavior is substantial in the data runs between time ~ 5ms and time ~ 21ms, implying the metrics are more susceptible to the homological features collected in the TDA methods. The plasma behavior can be depicted by foreground detection, and the visuals depict the exterior edge boundaries between the first substantial gradient (time ~ 5ms) and the last substantial gradient (time ~ 21ms). Understanding the location and intensity of plasma behavior leads to the depicted image structure at certain regions for more prominent topological features. In other words, observing more plasma interactions within the digital data correlates to higher counting of homological features and greater scaled values of the statistical metrics, which influence the findings of transitions in the dataset. The image complexity for data run 65 is known as “strong” due to the significantly extended size of the plasma edge boundaries interacting with the exterior fields. The metric plot results in higher ascents and descents of the time series, along with oversensitivity of the Wasserstein distance and minimal fluctuation from the bottleneck distance. The images in data run 69 are labeled as “small” for the small plasma edge boundaries when interacting with the exterior field. There are lower-scale ascents and descents with fewer fluctuations from the bottleneck distance. Data run 73 has images labeled as “none” for the limited plasma edge boundary size against the external magnetic field. There are lower-scale ascents and descents with even less fluctuations for the bottleneck distance. Based on this analysis, one can conclude that a higher image complexity of these images from magnetically confined plasma interactions is proportional to a higher number of essential topological features, which increases scaled metric quantities and therefore suggests more frequent transitional occurrences to review. Despite not having any specifications of the underlying physical profiles or probe measurements, the statistical metrics measure similar reactions for the timestamps of sharp topological transitions of time ~ 5ms and time ~ 21ms, indicating possible areas of interest for the specified data runs within the LAPD camera dataset. There is a metric spike around time ~ 5ms for all data runs, indicating oversensitivity of the Wasserstein metric for the noise variation when the device initializes the plasma transport. This spike could be a future investigation for Wasserstein metric theoretics. With the results shown and transitions indicated by subtle fluctuations and correlating gradients of the metric distances over every digital time series, the TDA methods identified topological variations that imply noticeable transitional occurrences in the camera dataset.

6. Conclusion and Future Directions

With the construction of the TDA techniques, the initial metric results reveal the possibility of finding possible plasma turbulent transitions given the topological changes for various November 2022 experiments in the LAPD camera dataset. While finding transitions, noise is controlled on the digital data with the addition of noise/sparsity alleviators, including Gaussian Blur and Opening/Closing operations. Sparse data is reduced because of additive Gaussian noise and altered digital data resolution with respect to image boundaries. Implementation of foreground detection on transitional images provides visual discernment of the recorded topological transitions in the LAPD experiments. The next directions are to reduce the spatial resolution of digital images for TDA confirmation and extend these techniques to datasets in solar investigations in order to yield more TDA contributions in scientific fields.

Bibliography

Adler, R. J., Bobrowski, O., & Weinberger, S. (2013). Crackle: The persistent homology of noise. arXiv preprint arXiv:1301.1466.

Cohen-Steiner, D., Edelsbrunner, H., & Harer, J. (2005). Stability of persistence diagrams. *Discrete & Computational Geometry*, 37(1), 263-271.

Fugacci, U., Scaramuccia, S., & Iuricich, F. (2016). Persistent homology: a step-by-step introduction for newcomers. *The Eurographics Association*.

Garbet, X. (2006). Introduction to turbulent transport in fusion plasmas. *Comptes Rendus Physique*, 7(6), 573-583.

Gekelman, W., Pribyl, P., & Lucky, Z. (2016). The upgraded large plasma device, a machine for studying frontier basic plasma physics. *Rev. Sci. Instrum*. 87(2), 87, 025105.

Kerber, M., Morozov, D., & Nigmetov, A. (2017). Geometry helps to compare persistence diagrams. *Journal of Experimental Algorithmics*, 22(1), 1-20.

Kiewel, Julius, "Identifying Transitions in Plasma with Topological Data Analysis of Noisy Turbulence" (2024). *Undergraduate Honors Theses*. William & Mary. Paper 2233.

Otter, N., Porter, M. A., & Tillmann, U. (2017). A roadmap for the computation of persistent homology. arXiv preprint arXiv:1506.0890.

Panaretos, V. M. & Zemel, Y. (2018). Statistical aspects of Wasserstein distance. *Annual Review of Statistics and its Applications*, 6, 405-431.

Stanish, Sage, "Counting Holes in Physical Systems - Applications of Computational Homology to Systems in Physics -" (2022). *Undergraduate Honors Theses.* William & Mary. Paper 1785.

The GUDHI Project (2025). GUDHI user and reference manual. (3.11.0). *GUDHI Editorial Board*.

Appendix

Computational Code

A.1 Computational Interpreter and Packages for Python Code

```
import numpy as np

import scipy as sp

import gudhi as gd

import gudhi.wasserstein as gdwas

import matplotlib.pyplot as plt

import cv2

import math

from gudhi.wasserstein import wasserstein_distance

import os

import matplotlib.ticker as mticker

# Python Interpreter Version : 12.5 (All python packages can be available for this version)
```

A.2 Topological Data Analysis: Cubical Persistent Homology, Sublevel Set Filtration, Persistent Homology, and Persistent Diagram

```
#Finds the betti persistence data of 2D image data

def genperscalc(img1):

 ccimg1 = gd.PeriodicCubicalComplex(top_dimensional_cells=img1, periodic_dimensions=[False,True])

 ccimg1.compute_persistence()

 pimg1 = ccimg1.persistence()

 return pimg1

# Find persistence of homological features

def get_betti_pers(p): # Input: Gudhi output .persistence()

 b0=[];y1=[];b1=[];y2=[];b2=[];y3=[]

 for i in range(len(p)):

  if p[i][0]==0:

   b0.append(p[i][1][0])

   y1.append(p[i][1][1])

  elif p[i][0]==1:

   b1.append(p[i][1][0])
```

```
      y2.append(p[i][1][1])
    if not isinstance(p[i][1], (list, tuple, np.ndarray)):
        # skip malformed entries
        continue
    if len(p[i][1]) < 2:
        continue
    b0.append(p[i][1][0])
    b1.append(p[i][1][1])
  return(b0,y1,b1,y2)
# Construct persistent diagram of homological features
def persinfo(tempdata): # Input: 2D data array
  cc = gd.PeriodicCubicalComplex(top_dimensional_cells = tempdata, periodic_dimensions=[False,True])
  cc.compute_persistence()
  p=cc.persistence()
  b0,y1,b1,y2 = get_betti_pers(p)
  lims = [np.amin(b0)-0.05,np.amax(b1)+0.05]
  fig=plt.figure()
  plt.xlabel("Birth",fontsize=20)
  plt.ylabel("Death",fontsize=20)
  plt.plot(b0,y1,'gx',label="Betti 0 Features")
  plt.plot(b1,y2,'rx',label="Betti 1 Features")
  plt.legend()
  plt.fill_between(lims,lims,y2=lims[0],color='#d3d3d3')
  return fig # Output: matplotlib.pyplot.figure() object
```

A.3 Noise/Sparsity Alleviators

```
# Adds gaussian noise to the image data fed in. This is from the random package, with mean equal to pixel value, and sigma being proportional to the max of the data
def crop_frame(frame, x_start, x_end, y_start, y_end):
    '''Crops images (frames) for displaying the essential image data.'''
    return frame[y_start:y_end, x_start:x_end]
def noisegauss(tempdata, size): # Input: 2D array, proportionality constant for sigma
   '''Applies a proportion of gaussian noise to the 2D array or digital data.'''
```

```
  noise = np.random.normal(loc=tempdata, scale=size)
  return tempdata + noise # Output: 2D array
def gaussblur(tempdata, sigma): # Input: 2D data array, sigma
  '''Applies Gaussian blur to the 2D array or digital data.'''
  return sp.ndimage.gaussian_filter(tempdata,sigma) # Output: 2D data array
#Opening/Closing smoothing algorithm
def imopen(tempdata, d):
  '''Applies opening as a morphological operation.'''
  return sp.ndimage.grey_opening(tempdata, size=(3,3))#structure=np.ones((3,3)))
def imclose(tempdata, d):
  '''Applies closing as a morphological operation.'''
  return sp.ndimage.grey_closing(tempdata, size=(3,3))#structure=np.ones((3,3)))
def opencloseblur(tempdata,itera):
  '''Combines opening and closing operations for 2D array or digital data.'''
  d = (len(tempdata), len(tempdata[0]))
  for i in range(itera):
    tempdata = imopen(tempdata, d)
    tempdata = imclose(tempdata, d)
  return tempdata
def process_frame(frame, noise_level, gblurred_level, openclose_level):
    '''Applies gaussian noise, gaussian blur, and opening/closing algorithm.'''
    noisy = noisegauss(frame, noise_level)
    gblurred = gaussblur(noisy, gblurred_level)
    openclose = opencloseblur(noisy, openclose_level)
    return gblurred, openclose
```

A.4 Statistical Metrics for Topological Structures

```
#Wasserstein comparison of two different image arrays
def wasserbetti(temppersistence):# Input: Gudhi output .persistence()
  '''Computes Wassertein comparison between two images.'''
  b=[];y=[]
  for i in range(len(temppersistence)):
```

```
    b.append(temppersistence[i][1][0])
    y.append(temppersistence[i][1][1])
  return np.vstack((b,y)).T
def wasserbetti2(temppersistence):
  b=[]; y=[]
  for i in range(len(temppersistence)):
    b.append(temppersistence[i][1][0])
    y.append(temppersistence[i][1][1])
  return np.vstack((b, y)).T
#Wasserstein comparison of two different image arrays
def WassersteinCompare(img1, img2): # Input: Two 2D arrays to compare
  '''Computes Wasserstein Distance of persistent diagrams from 2 images.'''
  ccimg1 = gd.PeriodicCubicalComplex(
    top_dimensional_cells=img1,
    periodic_dimensions=[False,True]); ccimg2=gd.PeriodicCubicalComplex(
      top_dimensional_cells=img2,
      periodic_dimensions=[False,True])
  ccimg1.compute_persistence(); ccimg2.compute_persistence()
  pimg1 = ccimg1.persistence(); pimg2 = ccimg2.persistence()
  cimg1 = wasserbetti(pimg1); cimg2 = wasserbetti(pimg2)
  return gdwas.wasserstein_distance(cimg2,cimg1,order=1.,internal_p=2.) # Wasserstein Distance Between Images
#Bottleneck comparison of two different image arrays
def BottleneckCompare(img1,img2): # Input: Two 2D arrays to compare
  '''Computes Wasserstein Distance of persistent diagrams from 2 images.'''
  pimg1 = genperscalc(img1); pimg2=genperscalc(img2)
  cimg1 = wasserbetti2(pimg1); cimg2=wasserbetti2(pimg2)
  return gd.bottleneck_distance(cimg1, cimg2) # Bottleneck Distance Between Images
def compute_metrics(original, processed):
    '''Computes Full Wasserstein and Bottleneck distances.'''
    wd = WassersteinCompare(original, processed)
    bn = BottleneckCompare(original, processed)
    return wd, bn
```

A.5 Image Processors

```
# Process foreground detection to applied images with plotting results
def process_pipeline(img, fgbg, out_mask, label1, label2, start_frame, transition_frames, end_frame, fps, frame_index,
data_run_id):
    """Apply foreground detection to transitional images."""
    import cv2
    import numpy as np
    import matplotlib.pyplot as plt
    import os
    # Ensure correct format
    if img.dtype != np.uint8:
        img = np.uint8(img / img.max() * 255)
    if len(img.shape) == 2:
        img = cv2.cvtColor(img, cv2.COLOR_GRAY2BGR)
    # Background Subtraction
    fgmask = fgbg.apply(img)
    # Write mask to video
    out_mask.write(fgmask)
    # Checkpoint frames
    checkpoints = {start_frame, *transition_frames, end_frame}
    if frame_index in checkpoints:
        time_value = (frame_index / fps) * 1000  # ms
        plt.figure(figsize=(15, 5))
        # Original image
        plt.subplot(1, 2, 1)
        plt.title(f'{label1} {time_value:.3f} ms', fontsize=24)
        plt.imshow(img)
        plt.axis('off')
        # Foreground mask
        plt.subplot(1, 2, 2)
        plt.title(f'{label2} FG Detect {time_value:.3f} ms', fontsize=24)
        plt.imshow(fgmask, cmap='gray')
        plt.axis('off')
```

```
    # Save images
    os.makedirs("Output_Frames", exist_ok=True)
    cv2.imwrite(os.path.join("Output_Frames", f"Original_{time_value:.3f}.png"), img)
    cv2.imwrite(os.path.join("Output_Frames", f"Mask_{time_value:.3f}.png"), fgmask)
    # Save figure
    plt.savefig(f'{data_run_id}_FG_Detect_Original_Image_{frame_index}.png')
    plt.close()
```

A.6 Plotting of Statistical Metrics for TDA Results

```
# Metrics over time
def plot_td_summary_with_time_c(metrics, frame_to_time, number_of_frames_to_read, end_frame, time_axis):
  '''Plots summary of metrics over time.'''
  if number_of_frames_to_read == end_frame:
   wd_gauss = [math.log(row[1]) for row in metrics]
   wd_openclose = [math.log(row[2]) for row in metrics]
   bn_gauss = [math.log(row[3]) for row in metrics]
   bn_openclose = [math.log(row[4]) for row in metrics]
   plt.figure(figsize=(10, 6))
   plt.plot(time_axis, wd_gauss, label="Full Wasserstein Gaussian Blur")
   plt.plot(time_axis, wd_openclose, label="Full Wasserstein Open/Close")
   plt.plot(time_axis, bn_gauss, label="Bottleneck Gaussian Blur")
   plt.plot(time_axis, bn_openclose, label="Bottleneck Open/Close")
   plt.xlabel("Time (ms)")
   plt.ylabel("Distance")
   plt.title("Topological Distances Over Time")
   plt.legend(loc="center right")
   plt.gca().xaxis.set_major_formatter(mticker.FormatStrFormatter('%.3f'))
   plt.grid(True)
   plt.savefig(f'./{'Topological Distances Over Time'}.png')
   plt.close()
  else:
   times = [(row[0])/(frame_to_time) for row in metrics]
```

```python
wd_gauss = [math.log(row[1]) for row in metrics]
wd_openclose = [math.log(row[2]) for row in metrics]
bn_gauss = [math.log(row[3]) for row in metrics]
bn_openclose = [math.log(row[4]) for row in metrics]
plt.figure(figsize=(10, 6))
plt.plot(times, wd_gauss, label="Full Wasserstein Gaussian Blur")
plt.plot(times, wd_openclose, label="Full Wasserstein Open/Close")
plt.plot(times, bn_gauss, label="Bottleneck Gaussian Blur")
plt.plot(times, bn_openclose, label="Bottleneck Open/Close")
plt.xlabel("Time (ms)")
plt.ylabel("Distance")
plt.title("Topological Distances Over Time")
plt.legend(loc="center right")
plt.gca().xaxis.set_major_formatter(mticker.FormatStrFormatter('%.3f'))
plt.grid(True)
plt.savefig(f'./{'Topological Distances Over Time'}.png')
plt.close()
```