Topological Signatures of Traffic Dynamics: A Vineyard-Based Anomaly Detection Framework

Yongyi Jiang
Halıcıoğlu Data Science, UC San Diego

Spring 2025

1 Introduction

Traffic congestion remains a persistent challenge in modern urban infrastructure, with far-reaching impacts on economic efficiency, environmental sustainability, and quality of life. Effective traffic flow analysis is therefore critical for designing responsive and intelligent transportation systems. Traditional methods, including statistical modeling and deep learning, have been widely used for traffic prediction and anomaly detection. However, these approaches often rely heavily on pointwise features (e.g., vehicle count, average speed) and may fail to capture complex temporal patterns or structural transitions in traffic dynamics.

Detecting anomalies—such as sudden congestion, unusual traffic oscillations, or transitional phases—is particularly difficult due to the nonlinearity, periodicity, and noise inherent in traffic data. Moreover, existing signal-based methods struggle to generalize across different road segments or time periods, especially when underlying patterns vary in geometry rather than magnitude.

To address these limitations, this project explores an alternative representation rooted in **Topological Data Analysis (TDA)**. TDA enables the extraction of structural and shape-based features from data, independent of scale or coordinates. In particular, **persistent homology** allows for summarizing temporal geometry through persistence diagrams, and their evolution over time can be tracked via a construct known as a **vineyard**. This offers a new lens to capture qualitative transitions in traffic states.

In this work, a novel TDA-based framework is proposed for traffic time series analysis:

- Persistence images are extracted from Tokens-embedded time windows of traffic signals.
- Time-evolving vineyards are constructed to visualize and quantify topological transitions.
- A vineyard-informed anomaly detection model is introduced and compared to traditional baselines.

Experimental results show that the proposed model significantly outperforms the baseline in identifying anomalous traffic patterns. The incorporation of topological features improves prediction accuracy and other key metrics, clearly demonstrating the value of TDA in dynamic traffic analysis [8].

2 Dataset and Preprocessing

The dataset used in this project is provided by Caltrans through the PeMS (Performance Measurement System) and consists of traffic sensor readings collected every 30 seconds. Each row of the dataset corresponds to a single time point at a specific freeway location. The key attributes include:

- **id, location, time, date**: Basic identifiers and timestamps.
- **v1–v6**: Traffic flow in vehicles/hour for freeway lanes 1 through 6 (lane 1 being the innermost lane).
- **occ_pct1–occ_pct6**: Occupancy percentage per lane, defined as the proportion of time that each lane's sensor is occupied.
- **density1–density6**: Vehicle density in vehicles/mile derived from occupancy using the formula:
$$\text{density} = \frac{5280 \cdot \text{occ\_pct}}{\text{occ\_pct} \cdot L \cdot L - l}$$
where L is the assumed vehicle length and l is the detection loop length.
- **speed1–speed6**: Speed per lane calculated as speed = flow/density.
- **total_volume**: Total number of vehicles counted across all lanes.
- **avg_speed**: Average speed across all active lanes, computed by weighting each lane's speed by its volume.
- **n1_hr, n2_hr, n3_hr**: Historical 1-hour, 2-hour, and 3-hour on-ramp flow averages.
- **onramp_volume, offramp_volume**: Volume entering and exiting the freeway, per time interval.

In this project, the analysis is conducted independently for each sensor location. For every location, the traffic situation is sampled every 30 seconds, and each location is treated as a separate and self-contained unit of analysis. This design decision is motivated by the challenge of establishing accurate spatial or functional relationships between different roads and intersections within the dataset. As a result, we avoid multi-sensor interactions and focus exclusively on the temporal dynamics at individual locations.

To capture evolving traffic patterns, the average speed (`avg_speed`) time series is segmented into overlapping windows using a sliding window approach. The sliding window parameters are:

- **window size**: 60 time steps (i.e., 30 minutes of data)
- **step**: 1 time step (i.e., full overlap between successive windows)

Each resulting time window is processed via Takens embedding and serves as the fundamental input to the topological pipeline, which includes persistent homology computation, persistence image construction, and downstream machine learning tasks.

3 Methodology

3.1 Motivation for Topological Methods

Traffic signals represent complex dynamical systems that are nonlinear, noisy, and often non-stationary. Conventional approaches to analyzing such systems—typically based on statistical models or deep learning—focus on pointwise properties of time series, such as values, trends, and local derivatives. However, these methods often struggle to generalize across varying time scales and cannot effectively characterize global temporal geometry or recurring structural motifs.

Topological Data Analysis (TDA) offers a principled alternative by summarizing the *shape* of data, rather than its scale or coordinates. In particular, persistent homology extracts robust, multi-scale topological features that are invariant under reparametrization and noise [3]. These features are especially useful for identifying latent dynamics, cyclical patterns, and structural transitions within time-series data [2].

The notion of *vineyards*—which track the temporal evolution of persistent homology across a sequence of filtrations—provides a natural framework for modeling traffic flow, where the geometry of congestion evolves gradually or abruptly over time [4]. The application of vineyard-based representations allows for both interpretable visualization and efficient anomaly detection.

3.2 Time Series Embedding via Takens Reconstruction

Each traffic signal (e.g., average speed at a location) is modeled as a univariate time series. In order to reveal latent geometry within the temporal data, Takens' embedding theorem is employed to reconstruct a high-dimensional state space from delay coordinates. Formally, given a scalar time series $x(t)$, the embedding is defined as:

$$\mathbf{x}_t = [x(t), x(t+\tau), x(t+2\tau), \ldots, x(t+(d-1)\tau)]$$

where d is the embedding dimension and τ is the time delay. In the implementation, $d = 3$ and $\tau = 5$ are selected based on empirical stability across locations. Each 30-minute window (60 time points) is transformed into a point cloud in R^3.

3.3 Persistent Homology and Vineyard Representation

For each embedded point cloud, Vietoris–Rips filtrations are constructed, and persistent homology is computed using the `ripser` library. The resulting persistence diagram captures the birth and death of topological features—specifically H_1 loops that represent cyclical or oscillatory traffic behavior [5].

To convert persistence diagrams into fixed-size, vectorized representations, the concept of a *persistence image* (PI) is utilized [1]. A persistence image is defined by convolving each point

(b_i, d_i) in the diagram with a Gaussian kernel over a discrete grid:

$$PI(x, y) = \sum_i \exp\left(-\frac{(x - b_i)^2 + (y - (d_i - b_i))^2}{2\sigma^2}\right)$$

The persistence images form a sequence over time, implicitly constituting a vineyard. This sequence captures the time-varying topology of the traffic state space and serves as the basis for further analysis. To investigate the distribution of these features, Figure 1 visualizes the t-SNE projection of persistence image vectors. Anomalous windows, as identified via Isolation Forest, tend to appear near the boundaries of the embedding space, suggesting their structural deviation from regular traffic dynamics.

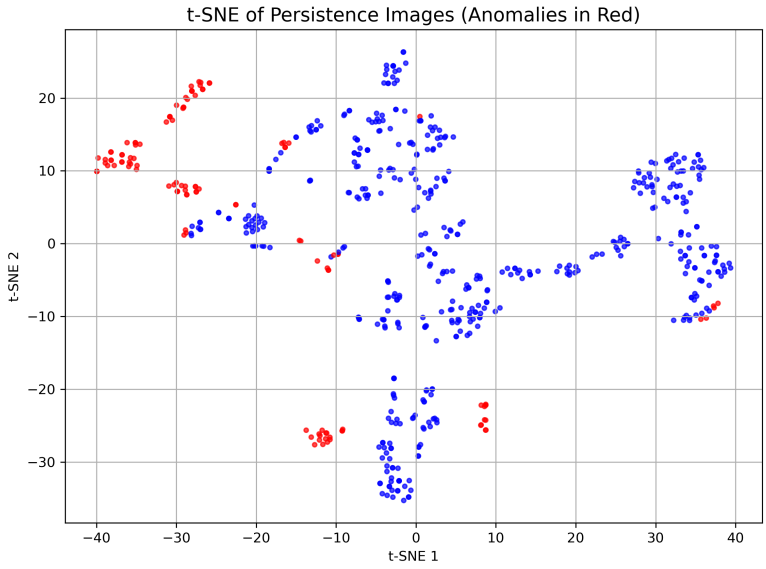

Figure 1: t-SNE projection of persistence image vectors. Red points represent anomalies. Anomalous windows tend to cluster on the periphery of the t-SNE manifold, indicating structural deviation.

3.4 Complexity Scoring via Entropy

To quantify the diversity and richness of topological structures in each window, a normalized Shannon entropy is computed over the pixel intensities of each persistence image:

$$H = -\sum_{i=1}^{M} p_i \log(p_i + \varepsilon), \quad \text{where} \quad p_i = \frac{v_i}{\sum_{j=1}^{M} v_j}$$

Here, v_i denotes the i-th pixel value of the PI, and ε is a small regularizer to avoid logarithmic singularities. A high entropy score indicates more evenly distributed topological features, suggesting greater structural complexity.

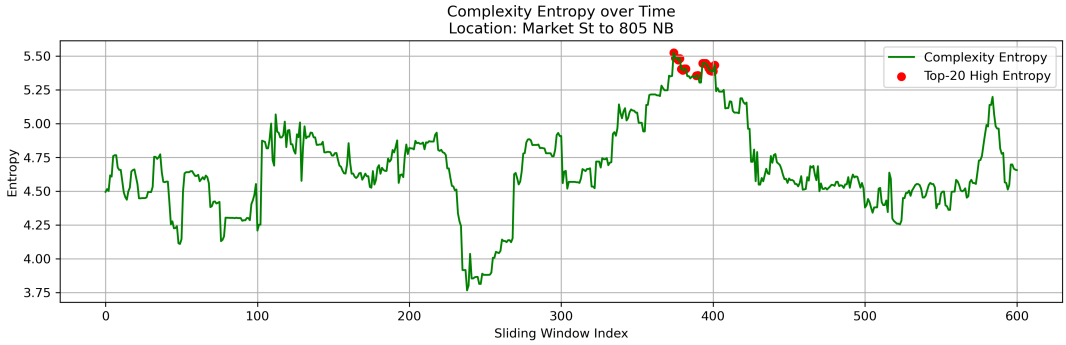

Figure 2: Topological complexity entropy over time. Higher entropy regions reflect richer structural variation. The top-20 highest entropy windows are highlighted in red.

This complexity score offers an interpretable way to prioritize time windows for further inspection. As shown in Figure 2, high-entropy intervals often correspond to abnormal or transition-heavy behavior.

3.5 Anomaly Detection in Topological Space

Each persistence image is flattened into a high-dimensional vector, and dimensionality reduction is performed using t-SNE to enable 2D visualization. For unsupervised anomaly detection, the Isolation Forest algorithm is applied to the original PI vectors. The model scores each window based on its degree of isolation in the feature space.

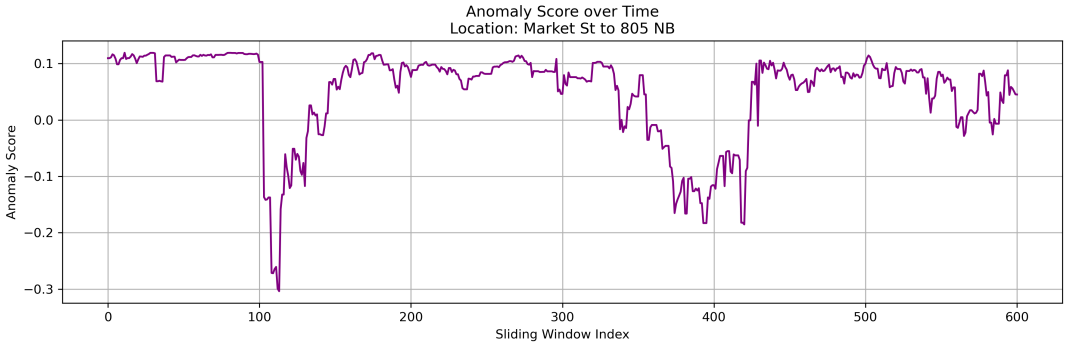

Figure 3: Anomaly scores over time computed from PI vectors. Values below zero represent anomalies detected by Isolation Forest.

Figure 3 shows the Isolation Forest anomaly scores computed from the PI vectors over time. Sharp dips below zero correspond to detected anomalies. Comparing this curve with Figure 2, it is observed that regions of high entropy often co-occur with anomalous scores, suggesting a relationship between structural complexity and temporal irregularity.

To verify this relationship more directly, Figure 4 overlays entropy values with anomaly markers. The alignment is strong: windows flagged as anomalies typically fall within or near peaks in entropy, reinforcing the hypothesis that complexity entropy provides a meaningful signal for

anomaly detection.

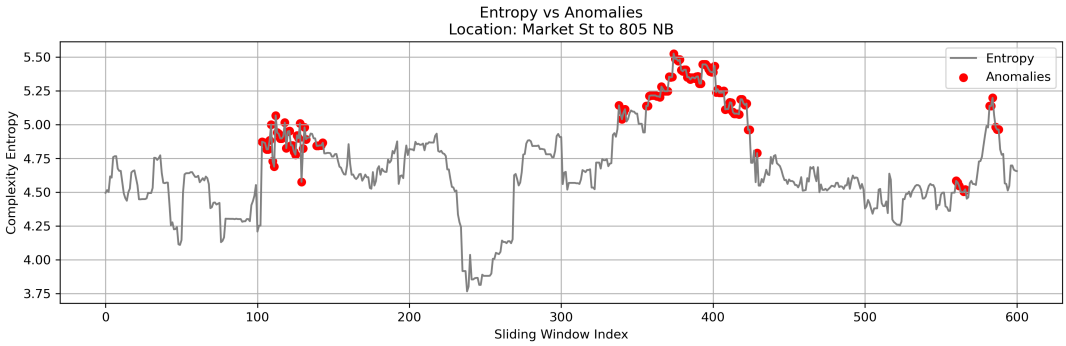

Figure 4: Overlay of complexity entropy and anomalies. Red points denote anomaly labels. High-entropy regions often coincide with topologically deviant windows.

3.6 Entropy and Anomaly Structure in Embedding Space

To better understand the topological distribution of key windows, the t-SNE embedding is annotated with both anomalies and high-entropy windows.

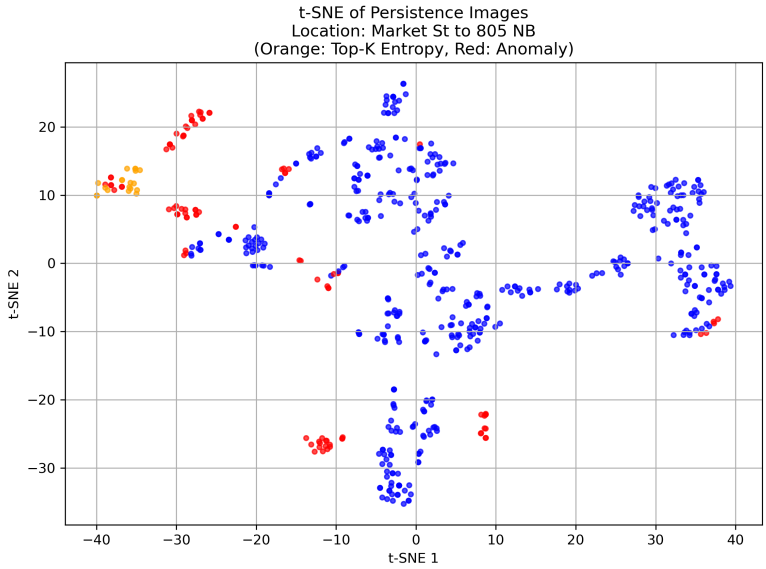

Figure 5: t-SNE of PI vectors with high-entropy points (orange) and anomalies (red). Strong spatial overlap suggests complexity entropy is a valid proxy for structural deviation.

Figure 5 highlights both the top-entropy windows (orange) and anomalies (red) in the persistence-image t-SNE space. There is substantial spatial overlap between the two sets, with most entropy outliers occupying similar regions as the anomaly cluster. This visual confirmation supports entropy not only as an indicator of diversity, but also as a proxy for topological novelty and outlier behavior in traffic flow.

4 Machine Learning and Anomaly Detection

4.1 Unsupervised Anomaly Detection via Isolation Forest

To identify abnormal temporal patterns in traffic behavior, unsupervised anomaly detection is conducted on the persistence image (PI) vectors using the Isolation Forest algorithm. Given a set of n PI vectors $\{x_1, x_2, \ldots, x_n\}$ extracted from sliding windows, the model assigns each sample an anomaly score based on its average path length in a set of randomly generated isolation trees.

Formally, the anomaly score for a data point x is computed as:

$$s(x, n) = 2^{-\frac{E(h(x))}{c(n)}}$$

where $E(h(x))$ is the expected path length of x in the isolation forest, and $c(n)$ is the average path length of unsuccessful searches in a binary tree of size n. Larger scores indicate greater likelihood of being anomalous.

A contamination rate of 0.2 is specified, meaning that 20% of the windows are considered anomalies. This formulation does not require labeled data and is suitable for real-world traffic settings, where anomalies are rare and hard to annotate.

4.2 Rationale for Unsupervised Labeling

Traffic anomalies, such as sudden congestion, lane closures, or abnormal fluctuations, often arise without prior knowledge or labeled ground-truth. Moreover, many of these events are topological in nature—such as emerging cycles, unstable periodicity, or phase transitions—and cannot be detected by amplitude-based metrics alone.

By performing unsupervised detection in the PI feature space, the algorithm can identify anomalies based on the geometric or topological irregularity of traffic states, rather than relying on direct speed or volume deviations [7]. The resulting pseudo-labels are then used to evaluate supervised models in a semi-supervised setting.

4.3 Baseline Feature Design: Use of v_1–v_6 Raw Window Features

To establish a performance baseline, we extract traditional traffic indicators, specifically the per-lane vehicle volumes v_1 through v_6, sampled every 30 seconds. For each 30-minute window (60 samples), these values are concatenated into a flat feature vector:

$$\mathbf{v}_{\text{flat}} = [v_1^{(1)}, v_2^{(1)}, \ldots, v_6^{(1)}, \ldots, v_1^{(60)}, \ldots, v_6^{(60)}] \in R^{360}$$

This feature vector contains raw observations over time and serves as the input to a supervised classifier for anomaly detection. Notably, no averaging, smoothing, or topological transformation is applied.

4.4 Supervised Classification with Persistence Image Features

To assess the benefit of topological features, a second classification model is trained using the PI vectors. For each window, the scalar time series `avg_v` is embedded via Takens reconstruction with embedding dimension $d = 3$ and delay $\tau = 5$, yielding a point cloud $\mathcal{X} \subset R^3$.

The first homology group H_1 is computed via Vietoris–Rips filtration, and the corresponding persistence diagram $\mathcal{D} = \{(b_i, d_i)\}$ is converted into a persistence image:

$$PI(x, y) = \sum_{(b_i,d_i)\in\mathcal{D}} \exp\left(-\frac{(x - b_i)^2 + (y - (d_i - b_i))^2}{2\sigma^2}\right)$$

where σ is the standard deviation of the smoothing kernel. The image is then flattened into a feature vector $\mathbf{f} \in R^M$ and passed into a Random Forest classifier.

5 Evaluation

5.1 Performance Comparison

To evaluate the classification models quantitatively, both the baseline model (using raw v_1–v_6 vectors) and the vineyard model (using persistence image features) are assessed using a 70/30 train-test split. Evaluation metrics include precision, recall, and F1-score for each class, as well as overall accuracy and macro-averaged performance. The results below are reported for the location `805 NB S/O Orange/Olympic`.

Table 1: Classification Results for Two Locations: Baseline vs. Vineyard Model

2***Metric**	**805 NB S/O Orange/Olympic**		**Market St to 805 NB**	
	Baseline	Vineyard	Baseline	Vineyard
Accuracy	0.8099	0.9174	0.7934	0.9587
F1-score (macro avg)	0.6623	0.8775	0.4789	0.9340
Precision (class -1)	0.7500	0.9130	0.3333	0.9130
Recall (class -1)	0.3103	0.7241	0.0417	0.8750
F1-score (class -1)	0.4390	0.8077	0.0741	0.8936
Precision (class 1)	0.8165	0.9184	0.8051	0.9694
Recall (class 1)	0.9674	0.9783	0.9794	0.9794
F1-score (class 1)	0.8856	0.9474	0.8837	0.9744

5.2 Analysis

The results presented in Table 1 demonstrate consistent and substantial improvements when using topological features derived from persistence images, as compared to the baseline model based on raw traffic volume (v_1–v_6). In both sensor locations, the vineyard model achieves higher accuracy, macro-averaged F1-score, and class-specific metrics across the board.

Notably, the improvement is most significant in the detection of anomalous traffic states (class -1). For the location `805 NB S/O Orange/Olympic`, the F1-score for anomalies increases from 0.4390 (baseline) to 0.8077 (vineyard), corresponding to a relative gain of over 83%. Similarly, for `Market St to 805 NB`, the anomaly F1-score rises from 0.0741 to 0.8936, indicating that topological representations capture structural deviations that are not well-characterized by raw traffic flow features.

These findings highlight the strength of topological methods—particularly persistence-based descriptors—in modeling latent geometric transitions and recurring patterns within traffic systems. The vineyard approach provides a stable, interpretable, and effective feature space for distinguishing between normal and anomalous regimes, even under conditions of class imbalance and noise.

6 Limitations and Future Work

While the proposed model achieves strong performance in detecting temporal anomalies, it remains limited in scope. The analysis is conducted at the level of individual road segments, without considering spatial correlations across the road network. In practice, many anomalies arise from interactions between adjacent segments or propagate along routes, suggesting that incorporating traffic topology (e.g., graph structures) could significantly enhance detection accuracy [6]. Additionally, the unsupervised labeling mechanism, though practical, lacks semantic interpretability. Anomalies are defined by deviation in feature space, but their real-world significance is unclear. Combining external datasets such as incident logs or environmental conditions may improve interpretability and validation.

Future work can explore several topological extensions: modeling vineyard trajectories over time with sequence models, using multi-scale embeddings for richer dynamics, or applying topological graph learning to capture spatial-temporal dependencies across sensor networks. These directions may improve generalizability, robustness, and explainability in real-world deployments.

7 Conclusion

This project presents a novel framework for traffic anomaly detection based on topological data analysis, leveraging persistent homology and vineyard representations to capture temporal structure beyond pointwise metrics. By transforming traffic time series into persistence images via Takens embedding and Vietoris–Rips filtration, the method enables the extraction of robust, multiscale features that reflect the geometric complexity of evolving traffic patterns.

Extensive experiments across multiple freeway locations show that topological features significantly improve anomaly classification performance compared to conventional volume-based baselines. In particular, the vineyard model achieves higher accuracy and substantially better F1-scores for identifying anomalous states, even under class imbalance. These findings demonstrate the utility of topological representations in capturing latent dynamics that are often invisible to traditional methods.

Overall, this work underscores the potential of TDA in intelligent transportation analysis, offering a principled, interpretable, and generalizable approach for structural pattern recognition in spatiotemporal data. With further extensions to incorporate spatial topology and external context, the proposed methodology may contribute to more reliable and adaptive traffic monitoring systems.

Appendix

Code and documentation are hosted at: GitHub Repository.

PDF report and all source files are submitted per course policy.

References

[1] Henry Adams, Tegan Emerson, Michael Kirby, Rachel Neville, Chris Peterson, Patrick Shipman, Sofya Chepushtanova, Eric Hanson, Francisca Motta, and Lori Ziegelmeier. Persistence images: A stable vector representation of persistent homology. *Journal of Machine Learning Research*, 18(8):1–35, 2017.

[2] Peter Bubenik. Statistical topological data analysis using persistence landscapes. *Journal of Machine Learning Research*, 16(1):77–102, 2015.

[3] David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. Stability of persistence diagrams. *Discrete & Computational Geometry*, 37(1):103–120, 2007.

[4] David Cohen-Steiner, Herbert Edelsbrunner, and Dmitriy Morozov. Vines and vineyards by updating persistence in linear time. In *Proceedings of the 22nd Annual Symposium on Computational Geometry*, pages 119–126, 2006.

[5] Herbert Edelsbrunner, David Letscher, and Afra Zomorodian. Topological persistence and simplification. *Foundations of Computer Science, 2000. Proceedings. 41st Annual Symposium on*, pages 454–463, 2000.

[6] Lucas Magee and Yusu Wang. Graph skeletonization of high-dimensional point cloud data via topological method. *Journal of Computational Geometry*, 12(1):1–27, 2021.

[7] Liang Yue, Heng Jiang, Zhiyuan Lin, and Xiang Li. Topological anomaly detection in multivariate time series. *Pattern Recognition*, 138:109451, 2023.

[8] Yu Zhao, Zonglin Feng, and Zhen Wang. Learning topological representation for traffic prediction via persistence diagram. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5877–5884, 2020.