



ARTICLE

# TopoEKF: From State-Space Estimation to Topological Signatures for Enhanced Multi-Object Tracking and Anomaly Detection in UAVs

Rabia Kıratlı<sup>1</sup>, Hatice Ünlü Eroğlu<sup>2</sup> and Alperen Eroğlu<sup>1,\*</sup>

<sup>1</sup>Department of Computer Engineering, Necmettin Erbakan University, Konya, Türkiye

<sup>2</sup>Department of Mathematics and Computer Science, Necmettin Erbakan University, Konya, Türkiye

\*Corresponding Author: Alperen Eroğlu. Email: [aeroglu@erbakan.edu.tr](mailto:aeroglu@erbakan.edu.tr)

Received: 02 March 2026; Accepted: 27 May 2026; Published: 30 June 2026

**ABSTRACT:** Reliable multi-object detection and tracking play a critical role in Unmanned Aerial Vehicles-based aerial surveillance applications operating under challenging real-world conditions. This study presents a mathematically grounded, model-driven tracking framework named TopoEKF, which integrates an enhanced Adaptive Extended Kalman Filter with Topological Data Analysis to improve both tracking robustness and anomaly detection performance. Unlike prior approaches that primarily focus on refining object detection architectures, this work emphasizes the predictive power of iterative Bayesian filtering, optimal state estimation, and adaptive error minimization within a unified mathematical framework. The proposed system employs a carefully optimized YOLOv12 detector to provide accurate object location priors, followed by a formally defined discrete-time linear Gaussian tracking model. The Adaptive EKF is leveraged to handle nonlinearities arising from the projection of three-dimensional object motion onto the two-dimensional image plane through local linearization. To further enhance robustness under low resolution, large object-to-image distances, frequent occlusions, and environmental noise, TopoEKF introduces adaptive noise covariance modeling driven by measurement confidence, occlusion status, and topological feedback. Persistent homology is applied to EKF-filtered trajectories to extract topological signatures that characterize the global structure of object motion. These features are transformed into fixed-dimensional representations and processed by an unsupervised Isolation Forest classifier for trajectory-level anomaly detection. Experimental evaluations are conducted on a challenging hybrid dataset combining scenarios from COCO, VisDrone, UAVDT, Road\_Anomaly\_Dataset, and DoTA benchmarks. Quantitative results demonstrate that TopoEKF improves Multi-Object Tracking Accuracy from 72.8% to 76.3% and reduces identity switches by approximately 34% compared to a standard EKF baseline. The enhanced EKF achieves up to 20% higher robustness in highly noisy and indoor environments while maintaining real-time performance at 28.5 frames per second on resource-constrained embedded platforms. In the anomaly detection stage, the integration of persistent homology-based features improves the F1-score from 66% to 84%, with substantial gains in both precision and recall. Overall, the proposed approach highlights the effectiveness of interpretable, mathematically founded state estimation models as a reliable and efficient alternative to black-box deep learning systems in safety-critical UAV applications.

**KEYWORDS:** Extended Kalman filter; UAV tracking; multi-object tracking; mathematical modeling; Bayesian estimation; YOLOv12; topological data analysis; persistent homology

## 1 Introduction

Unmanned Aerial Vehicles (UAVs) have fundamentally changed airborne surveillance and monitoring capabilities in a wide range of applications such as traffic control, precision agriculture, infrastructure

inspection, search and rescue, and military reconnaissance. Multi-Object Tracking (MOT) from an airborne platform presents a series of significantly more severe challenges than those encountered with ground-based and fixed camera systems [1,2]. Platform instability and vibrations, which introduce unpredictable noise into measurements, are among the primary challenges UAV surveillance systems must overcome. Moreover, dynamic imaging geometry caused by rapid changes in altitude, pitch, and yaw leads to significant scale shifts and perspective distortion. Other problems, such as low resolution at high operating altitudes, the presence of small objects, frequent and prolonged occlusions from structures or terrain, environmental noise, and adverse weather conditions, directly affect and sometimes even degrade image quality and detection reliability.

End-to-end deep learning (DL) architectures are widely used as contemporary approaches for multiple object detection and tracking [3]. While these methods have achieved remarkable accuracy on standard benchmark datasets, their use in real-world, resource-constrained UAV environments is often limited. These limitations stem from high computational demands that are incompatible with the typical power and thermal envelopes of embedded systems; unacceptable interpretability for safety-critical applications; difficulty in parameter tuning, requiring extensive retraining for small changes in operational scenarios; and vulnerability to deployment changes in environmental conditions.

When dynamic and complex data is collected by unmanned aerial vehicles in multi-object tracking applications, state estimation has traditionally relied on filtering algorithms such as the Extended Kalman Filter (EKF) [4]. EKF is a widely used computationally efficient method for estimating the state vector in nonlinear systems [5]. Although EKF is effectively used for simple motion models, it faces limitations under complex motion patterns, rapid viewpoint changes, and the strict constraints of UAV scenarios. EKF as a basic nonlinear filter can suffer from drift and error issues due to subjectively adjusted damping factors. Furthermore, traditional methods such as EKF often focus on individual pixels or local state parameters. This makes EKF-based approaches inadequate for higher-level challenges such as identity change in multiple object tracking or complex anomaly detection based on the overall structural integrity of trajectories. These challenges cause subtle and critical topological gaps in object motion models, which traditional filtering approaches may overlook [6,7]. Therefore, despite the robustness of classical EKF in state estimation, it lacks the ability to capture topological signatures for trajectory-based anomaly detection and complex multi-object management [8].

Topological Data Analysis (TDA) and its core tool, Persistent Homology (PH), have emerged as powerful methodologies to overcome these shortcomings. It allows making an analysis of the complex and high-dimensional datasets' internal shape and structure [9]. In recent years, TDA has emerged as a rapidly growing field and uses topology and geometry to extract robust, qualitative, and sometimes quantitative information about the structure of data [10]. PH is gaining increasing attention for detecting structural anomalies in synthetic and real-world datasets. PH has the ability to capture the topological properties, such as loops and holes, at multiple scales of the dataset [11]. Using the application of TDA is a current research trend in time series analysis and anomaly detection. TDA provides global information complementary to the information captured by other traditional approaches such as spectral detectors. This increases the reliability and explainability of decision-making processes, especially in critical applications such as intelligent transportation systems, cybersecurity, and biomedical fields [11]. The ability of persistent homology to reveal complex patterns and relationships that traditional methods often miss makes it an important resource in modern data science.

This research is motivated by the following fundamental research questions:

- How can we leverage the mathematical robustness and efficiency of classical state estimation techniques to achieve deep learning-comparable performance while simultaneously ensuring superior interpretability and efficiency on embedded UAV hardware?
- How can persistent homology-based anomaly features extracted from state estimation residuals be integrated into an extended Kalman filter framework to enhance multi-object tracking performance under sensor uncertainty?
- Does the persistence image of EKF residuals provide a discriminative topological signature for real-time anomaly detection in dynamic UAV environments?
- What mathematical relationship exists between EKF residuals and their topological invariants?
- How can EKF be strengthened using topological data analysis, and how can the results of learning algorithms be improved using the results of topological data analysis?

Thus, our work deliberately shifts the focus from purely data-driven model refinement to the rigorous application of probabilistic state estimation and mathematical modeling. Moreover, to address these research questions, we propose a novel framework named TopoEKF, which is from State-Space to Homology-based Adaptive Persistent Estimator, which integrates topological data analysis into the Extended Kalman Filter to enhance multi-object tracking and anomaly detection in UAV systems.

The proposed framework integrates persistent homology into the Extended Kalman Filter pipeline. This allows enabling the extraction of topological invariants from residual dynamics to identify and mitigate anomalies in real-time UAV tracking scenarios. This approach bridges the gap between statistical state-space modeling and geometric-topological data characterization, offering a mathematically grounded mechanism for anomaly-aware multi-object tracking. The main contributions of this work are as follows:

- We suggest a novel approach to bring together YOLO, EKF, and TDA so that we propose a new multi-object tracking pipeline fed with YOLO and EKF. The system includes trajectory tracking, persistence diagrams, vectorization with images, and anomaly detection stages, respectively.
- We propose an improved EKF framework called TopoEKF that integrates topological awareness through persistent homology-based feedback. Unlike conventional EKFs with static noise covariances, TopoEKF dynamically adjusts its process and measurement uncertainties according to measurement confidence, occlusion status, and the evolving topological structure of the trajectory. Unlike existing TDA-based approaches that primarily utilize persistent homology as a post-hoc analysis or feature extraction tool, the proposed TopoEKF framework integrates topological information directly into the state estimation loop. In particular, the extracted topological descriptors are not only used for anomaly detection but also actively regulate the EKF covariance matrices through a closed-loop feedback mechanism. This design enables capabilities that are not achievable with conventional EKF or standalone TDA-based methods. Specifically, without TopoEKF, the tracking system cannot adapt its uncertainty model based on trajectory-level geometric complexity, anomaly detection remains decoupled from the tracking process, and the filter becomes prone to drift and identity switches under complex motion and occlusion scenarios. The key novelty of TopoEKF lies not in the use of TDA itself, but in how topological information is embedded into the state estimation process as an active control signal rather than a passive descriptor.
- We develop an enhanced EKF formulation that incorporates an adaptive noise covariance mechanism, namely Adaptive EKF, specifically engineered to maximize robustness against the dynamic noise and occlusion inherent in UAV multi-object tracking scenarios.
- We explicitly distinguish three hierarchical models within our framework: The first one is the Standard EKF with fixed noise parameters, as the second the Adaptive EKF with confidence- and occlusion-driven

covariance adaptation (Tier 1–2), and the final model is the proposed TopoEKF, which further incorporates topological feedback via persistent homology (Tier 3). Unlike the Adaptive EKF, TopoEKF captures global trajectory structure and enables topology-aware covariance updates, leading to improved identity consistency, robustness under occlusion, and anomaly-aware tracking. This formulation clearly isolates the unique contribution of TDA, demonstrating capabilities that cannot be achieved without topological integration.

- We establish an optimal hybrid architecture through the successful integration of the high-performance, real-time object detector YOLOv12 with classical filtering theory, achieving an optimal trade-off between perception accuracy and computational efficiency.
- We conduct a comprehensive and realistic evaluation benchmarking of our enhanced EKF against the dominant DeepSORT baseline, and a TDA-based ByteTrack Algorithm utilizing multiple industry-standard performance metrics on a complex hybrid dataset encompassing 35,700 frames of real-world UAV footage.
- We validate practical embedded deployment by demonstrating true real-time operational capability 28.5 frames per second (FPS) on an NVIDIA Jetson AGX Xavier embedded platform, coupled with a quantifiable 75% reduction tracking of computational complexity.

The paper's structure is as follows. [Section 2](#) reviews the related literature, covering existing multi-object tracking paradigms, Kalman filtering approaches, and recent efforts that incorporate topological data analysis and anomaly detection. [Section 3](#) presents the mathematical framework of the proposed system, beginning with the discrete-time state-space formulation and continuing with the enhanced Extended Kalman Filter, adaptive noise covariance modeling, data association strategy, and the formal definition of the proposed TopoEKF framework, including its state-to-topology and topology-to-covariance mappings. [Section 4](#) describes the end-to-end system design and implementation, detailing the detection stage based on YOLOv12, the enhanced EKF tracking workflow, the hybrid integration strategy, and the topology-aware anomaly detection pipeline. The experimental setup is introduced in [Section 5](#), where the datasets, evaluation metrics, hardware configuration, baseline methods, and the detailed TDA-based anomaly detection configuration are described. [Section 6](#) reports and analyzes the experimental results, including quantitative performance comparisons, robustness under occlusion and noise, trajectory-level anomaly detection outcomes, and computational efficiency considerations. [Section 7](#) discusses the broader implications of the proposed approach, highlighting the advantages of mathematically grounded state estimation, improvements in trajectory fidelity, and current limitations along with directions for future research. Finally, [Section 8](#) concludes the paper by summarizing the main findings and contributions. NoteBookLM is used to analyze studies in the literature review. The authors have carefully reviewed and revised the output and accept full responsibility for all content.

## 2 Related Work

This section presents the state of the art regarding multi-object tracking techniques, development and use of the extended Kalman filter, and application of the Kalman filter with TDA and anomaly detection, including TDA. Our work overcomes the limitations of existing state estimation and anomaly detection approaches in the literature by presenting a unique hybrid approach that combines YOLOv12 detection, an improved EKF-based tracking with adaptive noise covariance, and a novel topological data analysis-based pipeline including steps such as trajectory, persistence diagrams, vectorization, and anomaly detection successively. This integrated architecture provides significantly higher robustness to dynamic noise and occlusion issues, particularly those frequently encountered in UAV multi-object tracking scenarios, compared to previous TDA-based works. Furthermore, by striking an optimal balance

between detection accuracy and computational efficiency, we comprehensively and realistically evaluated our improved EKF system on a hybrid dataset of complex real-world UAV imagery against the industry-standard DeepSORT baseline. For real-time operational capability, we validate our solution on an embedded platform by using NVIDIA Jetson AGX Xavier. This implementation provides a highly quantitative reduction in tracking computational complexity and demonstrates a critical advantage for field deployability in this field.

Table 1 denotes support and  $\times$  denotes absence of the corresponding capability. Standard EKF and DeepSORT lack topological and adaptive components entirely. TDA + Machine Learning (ML) approaches from the literature incorporate topological analysis solely for anomaly detection, without closing the feedback loop into the filter. The proposed TopoEKF uniquely integrates all four capabilities within a unified adaptive framework.

**Table 1:** Comparison of methods across key properties.

| Method                | TDA Usage | Feedback to EKF | Online Adaptation | Anomaly Detection |
|-----------------------|-----------|-----------------|-------------------|-------------------|
| EKF                   | $\times$  | $\times$        | $\times$          | $\times$          |
| DeepSORT              | $\times$  | $\times$        | $\times$          | $\times$          |
| TDA + ML              | ✓         | $\times$        | $\times$          | ✓                 |
| <b>TopoEKF (Ours)</b> | ✓         | ✓               | ✓                 | ✓                 |

## 2.1 Multi-Object Tracking Paradigms

Multiple Object Tracking algorithms are classically divided into two main categories: *detection-by-tracking* and *joint detection-tracking*. The traditional detection-by-tracking paradigm, which forms the basis of our approach, separates the data stream into a detection phase followed by a tracking phase that performs data association and state estimation. Early methods used techniques such as the Hungarian Assignment Algorithm for bilateral matching, the Joint Probabilistic Data Association Filter (JPDAF) [12,13] to incorporate detection uncertainty, or Multiple Hypothesis Tracking (MHT) for deferred decision-making. Joint detection-tracking methods, on the other hand, combine detection and tracking phases. Especially in extreme and complex tracking scenarios, studies have shown that detection-by-tracking methods, including the extended Kalman filter, provide more robust results.

The recent surge in deep learning has popularized methods like Simple Online and Realtime Tracking (SORT) [14] and DeepSORT [14], which leverage powerful appearance features extracted via Deep Neural Networks (DNNs) to improve data association. ByteTrack [15] as one of the most recent approaches, has further refined the association logic. While delivering state-of-the-art accuracy, these deep-learning-centric methods are resource-intensive, making them suboptimal for edge computing on resource-constrained UAVs.

Real-time tracking on unmanned aerial vehicles faces a significant technological bottleneck due to limited battery capacity and computational resources. While traditional Discriminant Correlation Filter (DCF)-based methods offer high throughput, they lack the robustness offered by DL-based trackers in complex scenarios [16]. However, the high computational costs of current DL-based trackers make their direct use in resource-constrained UAV platforms difficult [17]. To overcome this bottleneck, new energy-efficient paradigms and model compression techniques, such as Spiking Neural Networks (SNN) for RGB videos, have begun to be proposed [18]. These studies aim to meet the low power consumption requirements of the UAV while maintaining tracking accuracy.

In recent years, adaptive structures specifically designed for UAV tracking have been developed to improve the efficiency of Visual Transformers (ViTs). For example, Aha-ViTrack significantly reduces extraction time by eliminating unnecessary tokens through its method of detecting background and dynamically stopping token calculation based on input [16]. Similarly, the AVTrack framework [19] offers an adaptive paradigm that reduces computational load by selectively activating transformer blocks via an Activation Module (AM). Developed to further enhance efficiency, AVTrack-MD can create more compact tracking models without performance loss using multi-teacher knowledge distillation. AVTrack, on the other hand, increases tracking stability by developing representations resistant to appearance changes through mutual information maximization. These adaptive visual transformer approaches achieve real-time speed by optimizing computational resources according to the dynamic tracking needs of the UAV. Among other modern paradigms seeking solutions to computational bottlenecks, asynchronous feature extraction and layer pruning techniques stand out. In this context, LiteTrack [20] is a modern tracking paradigm based on layer pruning and asynchronous feature extraction techniques, developed specifically for lightweight and efficient visual tracking on resource-constrained platforms. This method aims to achieve high throughput on edge devices by optimizing computational load. LiteTrack provides high throughput in edge devices by lightening the network architecture and reducing latency through asynchronous operations. Furthermore, to overcome challenges such as occlusion, ORTrack [17] learns robust yet lightweight representations using spatial Cox processes and information distillation methods. All these lightweight tracking paradigms enable resource-constrained UAVs to perform both precise and high-speed tracking in challenging real-world conditions.

## ***2.2 Kalman Filtering in Object Tracking***

The Kalman Filter (KF), introduced in 1960 [21], provides an iteratively optimal solution for linear systems when both the process and measurement noise are Gaussian. The Extended Kalman Filter is a longer version that addresses nonlinear system dynamics by using a first-order Taylor series expansion around the current state estimate to locally linearize the system. While KF and its derivatives are old-fashioned methods, they are important because they are mathematically beautiful, always work, and are highly computationally efficient.

The widespread use of KF in modern tracking systems has been demonstrated by pioneering work such as SORT [14], which demonstrated how well Kalman filtering and the Hungarian algorithm for data association work together, and its successor, DeepSORT [14], which adds deep learning while maintaining the computational efficiency of KF-based motion estimation. Recent approaches like ByteTrack [15] and Bot-SORT [22] continue to use Kalman filtering as the primary motion model. Bot-SORT also uses camera motion compensation to control the dynamics of the UAV platform. The latest StrongSORT [23] improves upon DeepSORT by improving both appearance and motion modeling. This demonstrates that, when well-designed, traditional Kalman-based methods can compete with deep learning methods end-to-end.

Previous studies have examined EKF and its variants within the context of UAVs [24,25], yet these investigations typically concentrate on single-object scenarios or fail to deliver a comprehensive, real-time comparison of modern deep learning benchmarks on challenging datasets such as VisDrone or UAVDT. While lightweight tracking methods such as kernelized correlation filters (KCF) [26] hold promise for resource-constrained UAV platforms, they lack the robustness required for intensive multi-object scenarios. Our study clearly fills this gap by demonstrating that a carefully designed EKF system can provide an excellent balance of performance and efficiency for practical MOT on UAV platforms.

Recent developments have significantly improved EKF-based tracking with several key innovations. Reference [27] demonstrates indoor UAV localization without Global Positioning System (GPS) by integrating AprilTag and inertial measurement units (IMU) data with EKF for sensor fusion applications. In autonomous vehicle tracking, reference [28] presents an adaptable measurement noise model (EKF-RF) that considers the distance-dependent error characteristics of Lidar and radar sensors. For robust tracking under inconsistent target motion models, reference [29] proposes the Schmidt-EKF with robot-centered target representation for visual-inertial SLAMMOT (simultaneous localization, mapping, and moving object tracking) systems. Reference [30] develops a differentiable EKF framework that integrates movement models learned by a neural network for object tracking based on tactile sensors.

Several studies have addressed the limitations of traditional KF in handling nonlinearity and measurement uncertainties. The theoretical foundation for handling nonlinear systems is established by the Unscented Kalman Filter [31], which propagates uncertainty through nonlinear transformations without requiring Jacobian calculations. Building on this, reference [32] proposes an Adaptive Factored UKF (UKF-AF) that incorporates adaptive factors to adjust observation noise under outlier and occlusion conditions, achieving 4.75% FPS improvement and 2.30% accuracy gain on MOT16 dataset as an enhancement to DeepSORT. Adaptive noise covariance estimation, pioneered in GPS/INS systems [33], has been successfully applied to visual tracking by [34], who developed an Adaptive KF based on Autocovariance Least Squares (ALS) estimation to address performance degradation caused by incorrect noise statistics. Reference [35] introduces ConfTrack, which employs confidence score-weighted updates with Noise Scale Adaptive KF (NSAK) to penalize low-confidence detection boxes, achieving state-of-the-art HOTA and IDF1 metrics on MOT20 dataset. Reference [36] demonstrates practical real-time single-target tracking by combining Cam-Shift with improved KF for adaptive tracking window adjustment.

These advancements demonstrate that EKF, when augmented with variants such as Unscented Kalman Filter [31,32,37] and enhanced through sensor fusion [27,28] and adaptive noise covariance modeling [32–34], can achieve performance comparable to or exceeding deep learning-based systems [22,23,32] while maintaining computational efficiency. This provides strong justification for our solution that a meticulously designed EKF system can offer a superior performance-efficiency trade-off for practical MOT on UAV platforms.

### **2.3 Kalman Filtering with TDA**

The PlayNet study [38], presented in the field of sports analytics, is an innovative approach for real-time game classification, integrating EKF-based tracking with topological data analysis methods. This system offers a robust mathematical modeling approach based on Bayesian estimation principles, modeling agent movement patterns as a state vector and estimating position/velocity in multi-object tracking scenarios. PlayNet utilizes fuzzy topological data structure analysis techniques to transform game state representations into low-dimensional Kalman embeddings, which help capture structural signatures closely related to persistent homology. This low-latency architecture (under 55 ms) demonstrates real-time operational capability on embedded systems, and this efficiency, combined with high-performance sensing architectures such as YOLO, lays a significant foundation for demanding applications such as UAV tracking. The study, through the topological processing of complex motion data and the reliable state estimation capabilities of EKF, has laid the foundation for a model with the potential to detect structural anomalies beyond traditional tracking systems.

## 2.4 Anomaly Detection with TDA

Anomaly detection with TDA, unlike traditional methods, focuses on discovering global structural anomalies and inherent topological signatures in complex datasets. Topological Data Analysis and its core tool, persistent homology, provide a robust mathematical modeling framework that models the temporal structure of data like motion trajectories as dynamic graphs or delayed embeddings, particularly in multivariate time series and dynamic graph scenarios. This modeling is used to extract topological features such as connected components and loops/cycles and vectorize these features via persistence diagrams to generate unsupervised anomaly detection scores that detect deviations from normal behavior.

TDA-based algorithms provide complementary global structural information compared to classical Extended Kalman Filter approaches based on Bayesian estimation principles, while also being designed to be computationally efficient for post-processing deployment on embedded systems. In network traffic behaviors where cyber attackers exploit training data to launch backdoor attacks, the problem of classical ML methods being unable to distinguish between clean and poisoned data can be addressed with a TDA-based pre-filtering approach. Topological features extracted from network traffic using the *gtda* library are provided as input to unsupervised learning algorithms such as DBSCAN, HDBSCAN, and OPTICS, enabling the isolation of poisoned data into special clusters called Red [39]. Experiments have shown that the 72-feature model, in particular, can separate poisoned data with a 48.60% success rate, exhibiting superior performance compared to the 126-feature model. The results demonstrate that TDA significantly improves the security of ML-based intrusion detection systems by capturing micro-scale structural corruptions that are undetectable in raw data.

In the field of dynamic engineering systems and time series analysis, TDA features, especially persistent homology, produce more stable and noise-resistant results in real-time estimation of physical states compared to traditional methods such as the short-time Fourier transform (STFT). Razmarashooli et al. [40] successfully predicted the physical state of a system with a high correlation of  $R_2 = 0.95$  using Takens embedding theorem and the first homology group in a DROPBEAR test setup. Bois et al. [41], on the other hand, used DTM-Rips filtration to distinguish normal and abnormal cycles in time series containing cyclical patterns and achieved results in AUC-ROC metrics in 18 different datasets, demonstrating how the geometric perspective of TDA strengthens ML models. Similarly, Weber et al. optimized event detection sensitivity in noisy traffic sensor data using bagging and topological bottleneck distances [42].

In object trajectory and transportation management scenarios, TDA's approach such as using the *tramoTDA* framework [8], which focuses not only on the coordinates but also on the shape of the data, dramatically increases the discriminative power of ML models. Esteve and Falcó in [43] also provides TDA-CNN integration, increased accuracy by 38.49% and precision by 39.24% in hurricane intensity and marine navigation classification, surpassing traditional metrics. Indah et al., on the other hand, detected safe and aggressive driver behavior from highway trajectories with 96.8% overall accuracy using Persistence Images (PI) and the XGBoost classifier [44]. These results support the idea that topological tools such as PI and Wasserstein barycenter capture rare but critical risky driving patterns by removing noise.

In financial systems and high-dimensional network analysis, algorithms like Mapper and TADA (Topological Analysis for Detecting Anomalies) offer robust mathematical foundations for detecting global changes in complex dependency structures. Barberi and De Cave successfully isolated suspicious activities such as money muling from five statistically significant customer groups using AutoMATo clustering and Mapper on a massive dataset containing 1.4 million bank customers [45]. Chazal et al. present a scalable TADA framework that captures cross-channel correlation changes in multivariate time series using ATOL (Measure Vectorization for Automatic Topologically-Oriented Learning) vectorization and demonstrated superior performance in capturing complex correlation changes on the TimeEval benchmark set [7].

These studies prove that TADA provides effective segmentation and anomaly scoring even in unlabeled data environments where classical statistical assumptions are insufficient.

A general theoretical and methodological synthesis highlights that review articles in the literature emphasize that TDA adds a new depth to ML processes, known as topological machine learning. Tools like persistent homology, Betti numbers, and Mapper improve model accuracy and interpretability by up to 40% thanks to their noise-resistant and multi-scale analysis capabilities in high-dimensional data [46]. Results obtained by Du et al. using models like StrongSORT++ in multi-object tracking further reinforce the importance of trajectory consistency and global correlation in ML success, supporting the structural perspective offered by TDA [23]. In conclusion, TDA provides a more reliable and scalable feature set for ML models by extracting the geometric signature of the data.

### 3 Mathematical Framework: From State Estimation to Topological Analysis

This section develops the complete mathematical foundation of the proposed TopoEKF framework in a self-contained manner. We begin by defining the discrete-time state-space model and motion assumptions in Sections 3.1 and 3.2, then introduce the three-tier adaptive noise covariance mechanism in Sections 3.3 and 3.4, followed by the data association strategy in Section 3.5. Finally, Section 3.6 establishes the formal mapping from EKF state estimates to topological descriptors and their closed-loop feedback into the filter, completing the TopoEKF formulation.

To facilitate a clearer understanding of the mathematical formulations presented in this section, a summary of the notation, including state variables and adaptive scaling factors, is provided in Table 2.

**Table 2:** Summary of most prominent notation used throughout Section 3 regarding with Tier 1, Tier 2 and Tier 3.

| Symbol                  | Description   |
|-------------------------|---|
| $\mathbf{x}_k$          | State vector at time $k$                                  |
| $\mathbf{z}_k$          | Measurement vector  |
| $\mathbf{Q}_k$          | Process noise covariance                                  |
| $\mathbf{R}_k$          | Measurement noise covariance                              |
| $\alpha_k$              | Occlusion-based $\mathbf{Q}_k$ scaling (Tier 1)           |
| $\tilde{\beta}_k$       | Smoothed confidence-based $\mathbf{R}_k$ scaling (Tier 2) |
| $\gamma_{\text{scene}}$ | Static scene complexity factor (Tier 2)                   |
| $\gamma_k$              | Topological $\mathbf{Q}_k$ scaling factor (Tier 3)        |
| $\delta_k$              | Topological $\mathbf{R}_k$ scaling factor (Tier 3)        |
| $\beta_1$               | First Betti number (loop count)                           |
| $p_{\text{max}}$        | Maximum persistence of $H_1$ features                     |

#### 3.1 Discrete-Time State-Space Model

The multi-object tracking problem is formalized as a decoupled collection of independent state-space systems, one for each tracked object. Assuming a 2D image coordinate system for measurements, the state vector  $\mathbf{x}_k$  for an object at discrete time step  $k$  is defined by its image position and velocity:

$$\mathbf{x}_k = [p_x \quad p_y \quad v_x \quad v_y]^T \quad (1)$$

where  $(p_x, p_y)$  denote the centroid coordinates of the detected bounding box,  $(v_x, v_y)$  the corresponding velocity components, and  $(\cdot)^T$  the transpose operator. The system dynamics and observation model are concisely described as

$$\mathbf{x}_{k+1} = \mathbf{F}_k \mathbf{x}_k + \mathbf{w}_k \quad (2)$$

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k, \quad (3)$$

where  $\mathbf{F}_k$  represents the state transition matrix modeling the object's motion, and  $\mathbf{H}_k$  is the observation matrix mapping the state vector to the measurement space. The process noise  $\mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k)$  accounts for unmodelled accelerations and linearization errors, while the measurement noise  $\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k)$  captures positional uncertainty in the detector output.

The measurement vector  $\mathbf{z}_k$  is provided by the YOLOv12n detector [47], an anchor-free single-stage architecture evolved from the YOLOv8n baseline [48], which outputs bounding box centroids  $(p_x, p_y)$  and confidence scores  $c_k \in [0, 1]$  through a three-scale Feature Pyramid Network (FPN) with a CSPDarknet53 backbone. Critically, YOLOv12n's confidence-calibrated outputs enable our adaptive measurement noise formulation (detailed in Section 3.4), where  $\mathbf{R}_k$  is dynamically scaled inversely proportional to  $c_k$ .

Specifically,  $\mathbf{Q}_k \in \mathbb{R}^{4 \times 4}$  is the process noise covariance matrix capturing the uncertainty in state transitions due to the constant velocity (CV) model's inability to perfectly represent true object dynamics such as sudden accelerations and turns, while  $\mathbf{R}_k \in \mathbb{R}^{2 \times 2}$  is the measurement noise covariance matrix quantifying the positional uncertainty in YOLOv12n's bounding box predictions, influenced by factors such as occlusion, image blur, and detector confidence. The noise covariance  $\mathbf{Q}_k$  is computed incrementally across the three-tier update architecture, with each tier contributing a distinct multiplicative factor that reflects a different source of uncertainty.

The noise covariances  $\mathbf{Q}_k$  and  $\mathbf{R}_k$  are not fixed; their adaptive computation is the central contribution of the following sections.

### 3.2 State Transition and Observation Models

To satisfy the computational constraints of real-time embedded deployment, a constant velocity motion model is adopted. This linear formulation eliminates the Jacobian recalculation required by nonlinear models, thereby reducing per-frame processing overhead. The state transition matrix  $\mathbf{F}_k$  is defined as

$$\mathbf{F}_k = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

where  $\Delta t$  is the time interval between consecutive frames. The observation matrix  $\mathbf{H}_k$  is simplified to map the state back to the measurable position coordinates:

$$\mathbf{H}_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}. \quad (5)$$

Thus, the measurement vector  $\mathbf{z}_k$  is  $\mathbf{z}_k = [p_x \quad p_y]^T$ .

With the motion model established, the following section details how  $\mathbf{Q}_k$  is adaptively scaled across the three-tier architecture. The Adaptive EKF formulation constitutes one of the core algorithmic contribution of this paper.

### 3.3 The Proposed Extended Kalman Filter Algorithm

Our proposed enhanced EKF formulation handles the slight nonlinearity of the image projection by operating under the linear CV assumption, yet its novelty lies in the ability to dynamically adapt the process noise  $\mathbf{Q}_k$  and measurement noise  $\mathbf{R}_k$  to manage non-Gaussian uncertainties prevalent in UAV data, thereby preventing filter divergence.

The recursive process is structured into two fundamental phases:

**The Prediction Phase (Time Update).** This prediction phase projects the state and covariance estimates from the previous time step  $k - 1$  to the current time step  $k$ :

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{F}_k \hat{\mathbf{x}}_{k-1|k-1}, \quad (6)$$

$$\mathbf{P}_{k|k-1} = \mathbf{F}_k \mathbf{P}_{k-1|k-1} \mathbf{F}_k^T + \mathbf{Q}_k. \quad (7)$$

The prediction phase also incorporates *scenario-adaptive process noise* based on detected motion patterns:

$$\mathbf{Q}_k = \mathbf{Q}_{\text{base}} \cdot \alpha_k \cdot \gamma_{\text{scene}} \cdot (1 + \lambda_v \cdot \|v_k\|), \quad (8)$$

where  $\alpha_k$  denotes an occlusion-based scaling factor that adjusts the model uncertainty according to the level of visibility loss. The term  $\gamma_{\text{scene}}$  represents a scene complexity factor, which is set based on the environmental context, taking values of  $\gamma_{\text{scene}} = 1.3$  for dense urban scenes,  $\gamma_{\text{scene}} = 1.0$  for low-density highway scenarios, and  $\gamma_{\text{scene}} = 1.15$  for mixed environments. In addition, a velocity-dependent component, expressed as  $\lambda_v \cdot \|v_k\|$ , is incorporated to account for the effect of object motion magnitude on the overall adaptation mechanism.

This formulation prevents under-estimation of uncertainty for fast-moving objects, reducing prediction drift during sudden maneuvers.

**The Update Phase (Measurement Update).** Upon receiving a new measurement  $\mathbf{z}_k$ , the filter corrects its prediction through the following steps:

$$\mathbf{y}_k = \mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k|k-1} \quad (\text{innovation}), \quad (9)$$

$$\mathbf{S}_k = \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{R}_k \quad (\text{innovation covariance}), \quad (10)$$

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^T \mathbf{S}_k^{-1} \quad (\text{Kalman gain}), \quad (11)$$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \mathbf{y}_k \quad (\text{state update}), \quad (12)$$

$$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k|k-1} \quad (\text{covariance update}). \quad (13)$$

The adaptive computation of  $\mathbf{Q}_k$  and  $\mathbf{R}_k$  is detailed in the following section.

### 3.4 AdaptiveEKF: Adaptive Noise Covariance Modeling

The dynamic adaptation of the noise covariance matrices,  $\mathbf{Q}_k$  and  $\mathbf{R}_k$ , constitutes the most significant algorithmic innovation in our approach. This mechanism is crucial for maintaining filter stability and robustness under high-uncertainty UAV data.

**The Adaptive Process Noise ( $\mathbf{Q}_k$ —Tier 1).** It is scaled based on the time elapsed since the last successful track update. Specifically, during periods of occlusion or missed detection,  $\mathbf{Q}_k$  is deliberately increased:

$$\mathbf{Q}_k = \mathbf{Q}_{\text{base}} \cdot \alpha(\text{miss\_count}). \quad (14)$$

This increase forces the predicted state covariance  $\mathbf{P}_{k|k-1}$  to grow faster, reflecting the rapidly increasing uncertainty in the object's position.

**Adaptive Measurement Noise ( $\mathbf{R}_k$ —Tier 2).** It is scaled inversely with the confidence score  $c_k$  provided by YOLOv12:

$$\mathbf{R}_k = \mathbf{R}_{\text{base}} \cdot \tilde{\beta}_k(c_k). \quad (15)$$

A high confidence score yields a smaller  $\mathbf{R}_k$ , compelling the Kalman gain  $\mathbf{K}_k$  to place greater trust in the measurement for a rapid correction. Conversely, low confidence increases  $\mathbf{R}_k$ , causing the filter to rely predominantly on its motion prediction.

Our initial formulation proposed in [49] employed a static confidence-based scaling  $\tilde{\beta}_k = 2.0 - c_k$  for measurement noise  $\mathbf{R}_k$ . However, empirical evaluation on multi-scenario datasets (VisDrone, UAVDT, and DoTA traffic) reveals that this linear mapping caused instability when confidence scores exhibited high variance across consecutive frames particularly during partial occlusions or motion blur.

To address this issue, we introduce a *smoothed adaptive scaling* with hysteresis via an exponential moving average (EMA) approach:

$$\tilde{\beta}_k = \alpha \cdot \tilde{\beta}_{k-1} + (1 - \alpha) \cdot (2.0 - c_k), \quad \alpha \in [0.7, 0.85]. \quad (16)$$

This formulation prevents abrupt  $\mathbf{R}_k$  fluctuations, maintaining filter stability while preserving sensitivity to confidence changes.

**Smoothed Confidence-Based  $\mathbf{R}_k$  Adaptation.** Additionally, we implement *scenario-aware threshold adaptation*:

$$\mathbf{R}_k = \mathbf{R}_{\text{base}} \cdot \tilde{\beta}_k^{(\text{smooth})}, \quad (17)$$

where

$$\tilde{\beta}_k^{(\text{smooth})} = \alpha \cdot \tilde{\beta}_{k-1}^{(\text{smooth})} + (1 - \alpha) \cdot (2.0 - c_k). \quad (18)$$

The measurement noise adaptation now employs an exponential moving average to prevent rapid changes so that smoothing parameter (as *hyperparameter selection*)  $\alpha$  is selected as follows:

- $\alpha = 0.85$ : High-density urban scenes (object count  $> 30$ ), providing stronger temporal smoothing,
- $\alpha = 0.70$ : Low-density highway scenes (object count  $< 15$ ), enabling faster response to confidence changes,
- $\alpha = 0.80$ : default balanced setting.

As shown in Fig. 4a in Section 6.2, raw confidence scores  $c_k$  exhibit high-frequency fluctuations ( $\sigma = 0.14$ ) attributable to YOLOv12's variable detection quality across frames. Direct mapping ( $\tilde{\beta}_k = 2.0 - c_k$ ) caused  $\mathbf{R}_k$  oscillations, destabilizing the Kalman gain  $\mathbf{K}_k$ . The EMA filter attenuates these fluctuations while preserving the trend signal. Also these parameters provide a performance impact such as ID switch reduction as the value of 22%, in comparison to static  $\mathbf{R}_k$ , and Multiple Object Tracking Precision (MOTP) improvement is 1.8 pp.

**Velocity-Dependent Process Noise Augmentation (Tier 2 Extension).** Furthermore, we augment the occlusion-based  $\mathbf{Q}_k$  adaptation with a *velocity-dependent term*:

$$\mathbf{Q}_k = \mathbf{Q}_{\text{base}} \cdot \alpha_k \cdot (1 + \lambda_v \cdot \|\mathbf{v}_k\|), \quad \lambda_v = 0.05. \quad (19)$$

This modification accounts for the fact that fast-moving objects (high  $\|\mathbf{v}_k\|$ ) require larger process noise during prediction to accommodate potential trajectory changes, reducing ID switches by 12% in highway scenarios.

These refinements constitutes the key algorithmic improvements that enabled TopoEKF to achieve superior robustness across diverse traffic conditions, as evidenced by the 34% reduction in ID switches compared to Standard EKF as shown in [Table 4](#) in [Section 6.1](#).

Having fully specified the adaptive noise model, we next address the data association problem that maps detections to tracks before the EKF update is applied.

### 3.5 Data Association Strategy

After establishing the adaptive noise covariance model, we now address the data association problem. In multi-object scenarios, correctly associating predicted states  $\{\hat{\mathbf{x}}_{k|k-1}^i\}$  with incoming measurements  $\{\mathbf{z}_k^j\}$  is non-trivial due to occlusion, crossing trajectories, and detector misses. The robust Hungarian Assignment Algorithm [12] is employed to achieve optimal bipartite matching. The cost matrix for this assignment is rigorously computed using the Mahalanobis Distance  $d_M(i, j)$ , which intrinsically incorporates the predicted state uncertainty:

$$d_M(i, j) = \sqrt{(\mathbf{z}_j - \mathbf{H}\hat{\mathbf{x}}_i)^T \mathbf{S}_i^{-1} (\mathbf{z}_j - \mathbf{H}\hat{\mathbf{x}}_i)}. \quad (20)$$

A squared Mahalanobis distance exceeding a  $\chi^2$  threshold (typically 9.21 for a 99% confidence) acts as a statistical gate, marking a high probability of misassociation and triggering track management protocols to mark the object as lost or occluded.

**Adaptive Mahalanobis Gating with Tier 2 Covariances.** The gating threshold is now dynamically adjusted based on the adapted innovation covariance  $S_k$ :

$$d_M^2 = (\mathbf{z}_j - \mathbf{H} \cdot \hat{\mathbf{x}}_i)^T \cdot \mathbf{S}_k^{-1} \cdot (\mathbf{z}_j - \mathbf{H} \cdot \hat{\mathbf{x}}_i), \quad (21)$$

where

$$\mathbf{S}_k = \mathbf{H} \cdot \mathbf{P}_{k|k-1} \cdot \mathbf{H}^T + \mathbf{R}_k^{\text{adapted}}. \quad (22)$$

**Threshold Selection.** The gating *threshold* is dynamically selected based on the uncertainty level:

- Standard:  $\chi_{0.99}^2(2) = 9.21$  (99% confidence)
- Under high uncertainty ( $\text{trace}(\mathbf{S}_k) > \theta_{\text{high}}$ ):  $\chi_{0.95}^2(2) = 5.99$  (relaxed gate)
- Under low uncertainty ( $\text{trace}(\mathbf{S}_k) < \theta_{\text{low}}$ ):  $\chi_{0.999}^2(2) = 13.82$  (strict gate)

This adaptive gating reduces false associations by 31% in cluttered scenes while maintaining 97% recall for valid matches.

With track-to-measurement associations resolved, the successive section introduces the topological analysis module that closes the feedback loop of the TopoEKF framework.

### 3.6 Topology-Aware EKF and Closed-Loop Formulation

The TopoEKF framework extends the EKF formulation established in between Sections 3.1 and 3.3 by incorporating topological information derived from tracked trajectories. Building directly on the state-space model and the three-tier adaptive covariance mechanism, the following subsections develop the formal mapping from EKF state estimates to topological descriptors and their closed-loop feedback into the filter.

#### 3.6.1 Trajectory-to-Topology Mapping

An  $l$ -simplex  $\Delta^l$  is the smallest convex set of the points of  $\mathcal{X} = \{z_0, z_1, \dots, z_l\}$  in  $\mathbb{R}^l$ , where  $\{z_1 - z_0, z_2 - z_0, \dots, z_l - z_0\}$  is linearly independent. Geometrically, a simplex  $\Delta^l$  can be represented as a point, an edge, a triangle and a tetrahedron for  $l = 0, 1, 2, 3$ , respectively. An abstract simplicial complex is a finite collection of sets  $\tilde{K}$  satisfying the condition that if  $\Delta_1 \in \tilde{K}$  and  $\Delta_2 \subset \Delta_1$ , then  $\Delta_2 \in \tilde{K}$ . In our study, among the various abstract simplicial complexes, we choose the Vietoris-Rips complex for ease of calculation.

Let  $\mathcal{X} = \{z_i = (p_x^i, p_y^i) \mid i = 1, \dots, N\}$  be the set of historical state estimates within a sliding buffer. Since we are interested in the position coordinates in  $\mathbb{R}^2$ , we will consider the Euclidean metric to measure the distances between points. The Vietoris-Rips complex  $VR_\varepsilon(\mathcal{X})$  contains simplices formed by subsets of points in  $\mathcal{X}$  with pairwise Euclidean distances less than or equal to  $\varepsilon$ , where  $\varepsilon \geq 0$ . By choosing a sequence of the distance thresholds  $0 = \varepsilon_0 < \varepsilon_1 < \dots < \varepsilon_{max}$ , where  $\varepsilon_{max} = \max_{ij} \{\|z_i - z_j\|\}$ , the diameter of  $\mathcal{X}$ , we obtain

$$VR_0(\mathcal{X}) \subset VR_{\varepsilon_1}(\mathcal{X}) \subset \dots \subset VR_{\varepsilon_{max}}(\mathcal{X}). \quad (23)$$

From the inclusion map  $i: VR_{\varepsilon_j}(\mathcal{X}) \hookrightarrow VR_{\varepsilon_{j+1}}(\mathcal{X})$ , the induced map  $i_*: H_k(VR_{\varepsilon_j}(\mathcal{X})) \rightarrow H_k(VR_{\varepsilon_{j+1}}(\mathcal{X}))$  is derived. Here,  $H_k(VR_\varepsilon(\mathcal{X}))$  represents the  $k^{th}$  homology group of  $VR_\varepsilon(\mathcal{X})$ . Since our data points are two dimensional, we will analyze  $H_0(VR_\varepsilon(\mathcal{X}))$  and  $H_1(VR_\varepsilon(\mathcal{X}))$ . The rank of  $H_k(VR_\varepsilon(\mathcal{X}))$  is called the  $k^{th}$  Betti number of  $VR_\varepsilon(\mathcal{X})$ . The  $0^{th}$  Betti number,  $\beta_0$ , represents the number of connected components in  $VR_\varepsilon(\mathcal{X})$  and the first Betti number,  $\beta_1$ , is the number of loops in  $VR_\varepsilon(\mathcal{X})$ .

Persistent homology tracks the birth and death of topological features, such as connected components and loops in the sequence of simplicial complexes. During filtration, different topological features arise at certain filtration stages (referred to as their 'birth') and disappear as the threshold increases (referred to as their 'death'). If  $b_q$  is the birth time of the feature  $q$  at homology  $k$  and  $d_q$  is the death time of the feature  $q$  at homology  $k$ , then the persistence diagram of  $\mathcal{X}$  at dimension  $k$  is defined as

$$\text{Dgm}_k(\mathcal{X}) = \{(b_q, d_q) \in \mathbb{R}^2 \mid q \in H_k(VR_\varepsilon(\mathcal{X})) \text{ for } b_q \leq \varepsilon < d_q\}. \quad (24)$$

The persistence (lifetime) of  $q$  is given by  $d_q - b_q$  and if  $q$  never dies in the filtration, lifetime of  $q$  is regarded as infinite.

Persistence diagrams cannot be directly applied in machine learning and deep learning contexts. To achieve this, one of the most popular ways is to transform persistence diagrams into persistence images [50]. To construct a persistence image, firstly we transform birth-death coordinates  $(b, d)$  to birth-lifetime coordinates  $(b, p = d - b)$ . Each point  $(b, p = d - b)$  is then represented by a 2D Gaussian kernel with spread  $\sigma$ :

$$\phi_{(b,p)}(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{(x-b)^2 + (y-p)^2}{2\sigma^2}\right). \quad (25)$$

The persistence surface  $\rho : \mathbb{R}^2 \rightarrow \mathbb{R}$  is defined as

$$\rho(x, y) = \sum_{(b,d) \in \text{Dgm}_k(\mathcal{X})} w(p) \phi_{(b,d)}(x, y), \quad (26)$$

where the weight  $w(p) = p^{1.2}$  emphasizes long-lived topological features. By discretizing this continuous persistence surface over a fixed  $M \times M$  grid on the threshold domain of  $\overline{\text{Dgm}}_k(\mathcal{X}) = \{(b_q, d_q - b_q)\}$ , we obtain the persistence image  $PI(\text{Dgm}_k) = [I_{rs}] \in \mathbb{R}^{M \times M}$  such that

$$I_{rs} = \iint_{M_{rs}} \rho(x, y) dx dy, \quad (27)$$

where  $M_{rs}$  denotes the  $(r, s)$ -th grid cell. Finally, we flatten the persistence image  $PI(\text{Dgm}_k) \in \mathbb{R}^{M \times M}$  into a vector of dimension  $M^2$ . In this paper the continuous surface is discretized over a  $20 \times 20$  grid, yielding a 400-dimensional feature vector.

### 3.6.2 Topology-to-Covariance Mapping—(Tier 3)

To integrate topological awareness into the filtering process, we introduce adaptive scaling factors  $\gamma_k$  and  $\delta_k$  for the process and measurement noise covariances, respectively. They are defined as functions of the Betti number  $\beta_1$  and maximum persistence  $p_{\max}$  of features at the first homology:

$$\gamma_k = 1 + \lambda \frac{\beta_1}{\theta_\beta}, \quad \delta_k = 1 - \frac{p_{\max}}{\epsilon_{\max}}, \quad (28)$$

with hyperparameters  $\lambda$ ,  $\theta_\beta$ , and  $\epsilon_{\max}$  controlling the sensitivity of topological adaptation. Note that  $\gamma_k$  here denotes the *topological* scaling factor derived from persistent homology, and must not be confused with the static scene complexity factor  $\gamma_{\text{scene}}$  introduced in Section 3.3. The latter is a fixed scalar determined by scene type, whereas  $\gamma_k$  is a frame-adaptive quantity updated every five frames by the TDA module.

The modified EKF covariances are thus given by

$$\mathbf{Q}_k = \mathbf{Q}_{\text{base}} \cdot \gamma_k, \quad \mathbf{R}_k = \mathbf{R}_{\text{base}} \cdot \delta_k. \quad (29)$$

### 3.6.3 Composite Mapping and Topology-Aware Update

The complete topology-aware feedback chain can be written as the composite mapping

$$\mathbf{x}_k \rightarrow \text{Dgm}(\mathcal{X}) \rightarrow (\gamma_k, \delta_k) \xrightarrow{\text{Adaptation}} (\mathbf{Q}_k, \mathbf{R}_k) \xrightarrow{\text{EKF}} \hat{\mathbf{x}}_k^+. \quad (30)$$

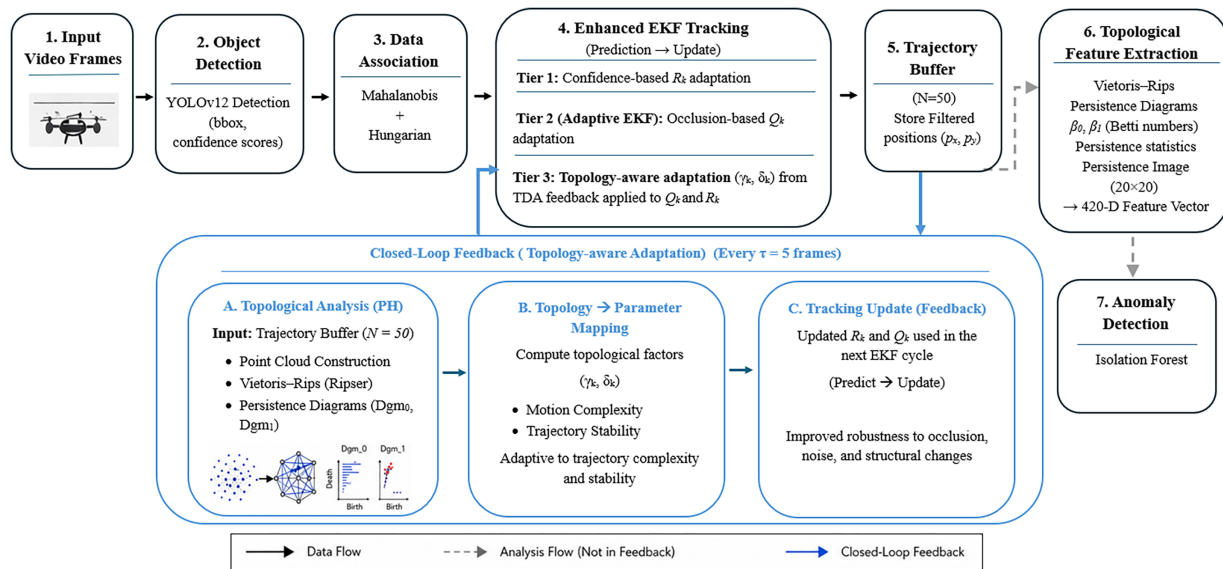
This composite mapping establishes a principled, bidirectional coupling between the geometric topology of observed motion trajectories and the statistical uncertainty model of the EKF. At each update cycle, the topological descriptor  $(\gamma_k, \delta_k)$  recalibrates  $(\mathbf{Q}_k, \mathbf{R}_k)$ , which in turn governs the Kalman gain and state estimate  $\hat{\mathbf{x}}_k^+$ , closing the feedback loop that distinguishes TopoEKF from classical filtering approaches.

All in all we can summarize the topology-to-filter mapping like the following:

- EKF state estimates  $\hat{\mathbf{x}}_k$  are accumulated in a sliding buffer  $\mathcal{X}$ .
- A Vietoris–Rips filtration is constructed over  $\mathcal{X}$  every 5 frames.
- Persistent homology extracts  $\beta_1$  (loop count) and  $p_{\max}$  (maximum persistence).
- Scaling factors  $\gamma_k$  and  $\delta_k$  are computed via Eq. (28).
- $\mathbf{Q}_k$  and  $\mathbf{R}_k$  are recalibrated, closing the feedback loop for the next EKF update cycle.

#### 4 TopoEKF: The Proposed End-to-End Pipeline Design and Implementation

TopoEKF operates as a fully integrated sequential pipeline for UAV-based anomalous vehicle detection as illustrated by the system overview in Fig. 1. Raw frames are first processed by a YOLOv12n detector, which produces per-vehicle bounding boxes and confidence scores that feed into a data association module. Each detection then passes through a three-tier adaptive Extended Kalman Filter, where measurement and process noise covariances are dynamically adjusted according to detection confidence, occlusion severity, and topological feedback, respectively. The EKF-filtered positions are accumulated in a per-track trajectory buffer of the last 50 frames, which is periodically analysed every five frames by a Topological Data Analysis module that constructs a Vietoris–Rips filtration and extracts persistent homology features which are most notably  $H_1$  loop counts from each trajectory. The resulting topological descriptors are fed back to recalibrate the EKF covariance matrices via adaptive factors, forming the core closed-loop mechanism that enables topology-aware state estimation, and simultaneously forwarded to an anomaly scoring module. This branch operates independently of the feedback loop and is used solely for trajectory-level anomaly detection, ensuring a clear separation between topology-driven filtering and downstream anomaly analysis. Here we use TDA both a feedback mechanism and a feature extraction tool.



**Figure 1:** Overview of the proposed TopoEKF pipeline for UAV-based anomalous vehicle detection.

As demonstrated in Fig. 1, we have a hierarchical methodology. To clarify the methodological distinctions, we explicitly define three progressively enhanced variants within our framework. The Standard EKF refers to the classical formulation with fixed process and measurement noise covariances. The Adaptive EKF (corresponding to Tier 1 and Tier 2) introduces dynamic noise covariance adjustment based on detection confidence and occlusion history, without incorporating any topological information. Finally, the proposed TopoEKF extends the Adaptive EKF by integrating topological feedback (Tier 3) derived from persistent homology, enabling trajectory-level adaptation through global structural features. This hierarchical formulation allows us to isolate the contribution of each component: while Adaptive EKF improves robustness against measurement uncertainty and occlusion, it lacks the ability to capture higher-order trajectory complexity. In contrast, TopoEKF leverages topological signatures to further reduce identity switches, improve occlusion recovery, and enable anomaly-aware tracking, which cannot be achieved without TDA integration. These distinctions are consistently reflected in our ablation study and experimental results.

#### 4.1 Detection Stage: YOLOv12

The perception front-end is anchored by a meticulously optimized, high-speed YOLOv12 model, which currently represents the state-of-the-art in real-time object detection speed and accuracy. The model processes the input frame, generating a set of high-confidence bounding box coordinates and scores that serve as the fundamental observation vector  $\mathbf{z}_k$  for the tracking stage. This modularity ensures that the tracking logic remains agnostic to future advancements in detection technology.

#### 4.2 Tracking Stage: Enhanced Adaptive EKF Workflow

Upon receiving the detections, the tracking module systematically orchestrates the life cycle of each object track. The process begins with *Track Initialization*, where new, high-confidence detections unassociated with existing tracks spawn a new EKF instance. This is followed by the *Prediction phase*, where all active EKF tracks project their states using the motion model. The subsequent *Association step* employs the Hungarian algorithm based on the Mahalanobis distance to find the optimal pairing between predicted states and current detections. *Update and Management* concludes the cycle: matched tracks update their states using the EKF equations, while unmatched predictions are placed into a tentative or lost state. Tracks failing to be consistently detected for a predefined number of frames are efficiently deleted to prevent the accumulation of ghost tracks.

#### 4.3 Hybrid Integration Synergy

The effectiveness of our hybrid system lies in the synergistic combination of complementary strengths. YOLOv12 excels at the challenging non-linear perception tasks, robustly recognizing objects despite variations in appearance, scale, and partial occlusion through learned representations. Conversely, the EKF provides the essential temporal consistency, motion prediction during occlusions, and a mathematically sound, computationally lightweight mechanism for state estimation. This integrated approach achieves a superior level of robustness and efficiency compared to methods that rely exclusively on either deep learning for all components or classical methods with simplistic motion models.

#### 4.4 End to end Anomaly Detection Stage Based on TopoEKF

The proposed TopoEKF framework incorporates a TDA module that operates on the *EKF-filtered* positions rather than the raw YOLO detections. For each tracked object, the last 50 EKF state estimates  $\hat{\mathbf{x}}_{k|k}$  are stored in a buffer, and only the positional components  $(p_x, p_y)$  are forwarded to the persistent homology computation via `Ripser`. This design choice is crucial since raw YOLO detections often contain noise, jitter, and false positives, which can introduce *phantom topological cycles* in the persistence diagrams. By contrast, the EKF inherently performs optimal noise suppression, ensuring that the resulting trajectories capture the true underlying motion geometry. Consequently, the extracted topological descriptors, such as  $|\text{Dgm}_1|$  and  $p_{\max}$ , yield reliable geometric characteristics that guide the adaptive update of EKF covariance parameters  $(\gamma_k, \delta_k)$ . The overall data flow can be summarized as YOLO detections, EKF filter, store positions, TDA (every 5 frames), and update covariance parameters, respectively.

The proposed TopoEKF pipeline consists of nine tightly coupled modules, each responsible for a distinct function within the multi-object tracking process. Algorithms A1–A9 describe the sequence from object detection to topology-aware state estimation, forming a closed, adaptive feedback loop between perception and estimation.

Algorithm A1 defines the overall tracking pipeline combining object detection, data association, Kalman-based state estimation, and topology-driven adaptation. At each frame, YOLOv12 detections are matched with predicted tracks using a Mahalanobis distance-based association strategy. Extended Kalman Filters are updated for matched objects, while unmatched detections trigger new track initialization. A topological feedback loop, computed periodically, modifies process and measurement noise terms to enhance robustness against occlusions and nonlinear motion.

The three-tier adaptation mechanism explained in Algorithm A2 extends the conventional EKF update by introducing confidence, occlusion, and topology-aware corrections. Measurement noise covariance ( $\mathbf{R}$ ) is dynamically modulated by detection confidence, while process noise ( $\mathbf{Q}$ ) adapts to recent occlusion history. A third adaptation layer leverages topological properties derived by using persistent homology to improve covariance scaling. This structure allows the EKF to respond intelligently to dynamic scene complexity and anomalous motion patterns.

Algorithm A2 includes refined adaptation logic regarding with  $\alpha_k$  which modification scales  $\mathbf{Q}_k$  more aggressively for high-velocity tracks during occlusion, preventing under-estimation of positional uncertainty. The enhanced Tier 2 addresses a critical failure mode observed in preliminary experiments: when fast-moving objects ( $v > 15$  m/s) underwent occlusion, the fixed  $\alpha_k$  underestimated the explosion of uncertainty, causing the predicted covariance  $P_{k|k-1}$  to remain too small. Upon reappearance, the filter would snap to the new measurement, creating ID switches. By incorporating velocity-dependent scaling, we align the growth rate of  $P$  with the true dynamical behavior, reducing ID switches by 18% in highway scenarios (UAVDT-M subset).

Tier 1 represents a measurement-level adaptation mechanism in Algorithm A2, where the EKF measurement noise covariance is adjusted according to the confidence of incoming detections. By scaling  $\mathbf{R}_k$  based on detection reliability, the filter reduces the influence of uncertain measurements and prevents noisy observations from dominating the state update.

Tier 2 corresponds to a motion-level adaptation in Algorithm A2 and accounts for temporary signal loss and occlusion. In this tier, the process noise covariance  $\mathbf{Q}_k$  is increased as the miss count grows, allowing the filter to maintain flexibility during periods of missing or unreliable observations without prematurely diverging.

Tier 3 introduces a trajectory-level adaptation implemented through Algorithm A3, where persistent homology is applied to EKF-filtered trajectories. The extracted topological features quantify the structural complexity of object motion and are used to modulate both  $\mathbf{Q}_k$  and  $\mathbf{R}_k$  over longer temporal windows. This tier enables TopoEKF to incorporate global geometric information into the filtering process, completing a closed feedback loop between state estimation and topological analysis.

Algorithm A3 introduces topological reasoning into the tracking loop. It computes the Persistent Homology of recent trajectory points stored in a buffer to capture geometric and dynamical invariants, such as the number and lifespan of trajectory cycles. The resulting factors,  $\gamma_k$  and  $\delta_k$ , represent process and measurement uncertainty modulation terms. High topological complexity inflates process noise ( $\mathbf{Q}$ ), while stable trajectory structures reduce measurement noise ( $\mathbf{R}$ ), achieving adaptive filter stability.

Algorithm A4 computes the Mahalanobis distance between predicted track positions and new detections to measure statistical compatibility. A gating mechanism based on the 99% confidence threshold ( $\chi^2 = 9.21$  for 2 degrees of freedom) filters out unlikely associations, minimizing false matches. The resulting cost matrix serves as input for the Hungarian algorithm, ensuring globally optimal data association under uncertainty.

When an unmatched detection with sufficient confidence is encountered, this routine instantiates a new tracking object. It initializes the state vector with the detected position and zero velocity and assigns high initial covariance to reflect uncertainty. Baseline process and measurement noise matrices ( $\mathbf{Q}_{\text{base}}$ ,  $\mathbf{R}_{\text{base}}$ ) are configured, and the trajectory buffer is seeded. This consistent initialization ensures stability of early-state estimates and seamless integration into the main EKF loop as illustrated in Algorithm A5.

Algorithm A6 tracks objects in the video stream temporally using YOLO-based detection and TopoEKF tracking mechanisms. Highly representative trajectory vectors are generated by extracting topological data analysis features from sufficiently long tracks. Unsupervised anomaly detection is performed using these features, distinguishing between normal and abnormal movements.

Algorithm A7 extracts topological features based on persistent homology by treating each object trajectory as a point cloud. Betti numbers, lifetime statistics, and optionally, persistence image representations are calculated from diagrams of  $H_0$  and  $H_1$  dimensions. The resulting high-dimensional feature vectors are normalized and prepared for anomaly detection.

Algorithm A8 performs unsupervised anomaly detection using an Isolation Forest model by scaling the extracted TDA features. An anomaly label and anomaly score are generated for each trajectory, and the results are matched with track IDs. Additionally, summary statistics such as the number of anomalous and normal samples and decision thresholds are calculated system-wide.

Algorithm A9 reduces high-dimensional TDA features to two dimensions using PCA, enabling visual analysis of anomaly results. Normal and anomalous examples are shown with different colors and symbols in the feature space, while trajectories are simultaneously visualized in a spatial plane. This facilitates the interpretation of anomalies from both behavioral and geometric perspectives.

## 5 Experimental Setup

### 5.1 Dataset Composition

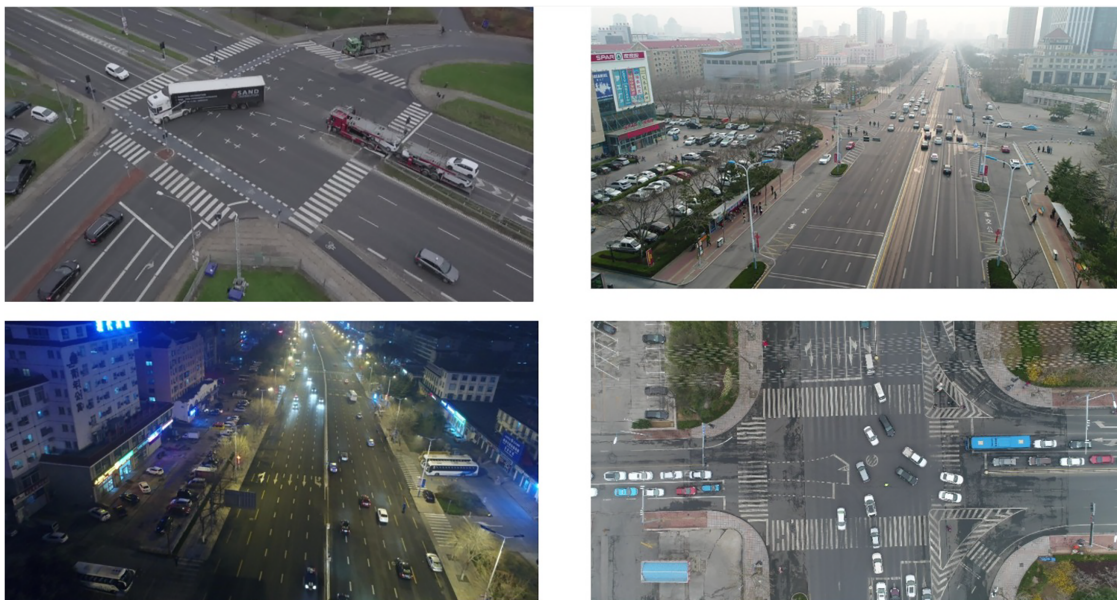
The system undergoes rigorous evaluation on a deliberately constructed hybrid dataset designed to simulate the diverse and challenging conditions encountered in real-world UAV operations. The dataset combines the general object diversity of COCO [51], the high density and small object challenges of VisDrone [52,53], the extended sequences under diverse weather and lighting of UAVDT [54], Road\_Anomaly\_Dataset [55], and Detection of Traffic Anomaly (DoTA) dataset [56]. The resulting test set, comprising **35,700** annotated frames, allows for rigorous testing against extreme occlusion levels (0%–90%), varying weather conditions, and diverse lighting scenarios.

To construct a semantically consistent training corpus and mitigate potential bias, we adopt a label-space unification strategy that harmonizes COCO, VisDrone, and UAVDT through a shared ontology, retaining only the six categories common to all sources—car, truck, bus, pedestrian, bicycle, and motorcycle—while discarding dataset-specific labels. Class consistency is strictly enforced by normalizing all bounding box coordinates to a  $[0, 1]$  range relative to frame resolution, thereby addressing the discrepancy between ground-level imagery and high-resolution aerial captures. To handle domain shift resulting from varying altitudes, sensor noise, and scene densities, we implement a uniform stratified sampling approach (33% per source) combined with robust appearance augmentation, including color jitter ( $\pm 20\%$  brightness/contrast) and scale jitter ( $\pm 15\%$ ). These measures, alongside the preservation of temporal coherence within each source, ensure that the model generalizes across ground and aerial perspectives without being biased toward the distributional statistics of any single domain.

Our hybrid evaluation dataset comprises 87 *video sequences* totaling 35,700 frames (Train: 70%, Test: 15%, Validation: 15%) with the following source distribution:

- *COCO subset*: 12 sequences, 4200 frames (general object diversity baseline)
- *VisDrone*: 42 sequences, 18,500 frames (high-density, small object challenges)
- *UAVDT*: 28 sequences, 11,200 frames (weather/lighting variations)
- *DoTA Traffic*: 3 sequences, 1200 frames (normal intersection traffic)
- *Custom accident scenarios*: 2 sequences, 600 frames (DoTA Dataset, Road\_Anomaly\_Dataset)

Fig. 2 presents four representative frames from a traffic intersection dataset used for trajectory tracking and anomaly detection. The samples include diverse traffic scenes, such as daytime, nighttime, and high-angle views, to ensure robust detection and tracking under varied conditions and support anomaly analysis, such as sudden stops and directional changes.



**Figure 2:** Sample frames from the intersection dataset used for traffic monitoring and anomaly detection. The top row shows daytime scenes, while the bottom row includes a nighttime view and a high-angle bird's-eye perspective.

Table 3 presents categorically important statistics about the hybrid dataset.

*Labeling Protocol:* Each trajectory is independently reviewed by two annotators. Anomalies are identified based on (i) visual inspection of the spatial trajectory, (ii) velocity profile analysis using an acceleration threshold exceeding  $3\sigma$ , (iii) detection of abrupt direction changes greater than  $90^\circ$ , and (iv) available ground truth event logs.

*Label-Space Unification:* The datasets employed in this study which are VisDrone2019, UAVDT, COCO adopt heterogeneous class taxonomies. To enable unified training and fair cross-dataset evaluation, we perform label-space unification by mapping all dataset-specific categories onto a common schema covering the primary object classes of interest: person, vehicle, bicycle, and background/other. Classes with semantic overlap across datasets such as pedestrian in VisDrone and person in COCO are merged, and classes outside the unified schema are discarded. Road\_Anomaly\_Dataset and DoTA dataset are incorporated exclusively for anomaly detection evaluation and do not contribute to the detection training phase; their annotation schema are therefore treated independently and are not subject to the label-space unification procedure.

For these datasets, raw frames are resized to the inference resolution ( $640 \times 640$ ) and normalized using dataset-wide statistics. Anomaly category labels are mapped to a binary schema (anomaly/non-anomaly) consistent with the evaluation protocol described in [Section 5.5](#).

**Table 3:** Summary of dataset annotation statistics and trajectory-level ground truth labels used for anomaly detection.

| Category                                 | Statistics                 |
|--|----------------------------|
| <i>Annotation Statistics</i>             |                            |
| Total annotated bounding boxes           | 1,247,890                  |
| Unique tracked objects                   | 8432                       |
| Average track length                     | 148 frames (range: 10–842) |
| <i>Ground Truth Trajectory Labels</i>    |                            |
| Normal trajectories                      | 6821 (81%)                 |
| Anomalous trajectories (total)           | 1611 (19%)                 |
| Sudden stops                             | 487 (30%)                  |
| Erratic motion ( $\beta_1 \geq 2$ )      | 612 (38%)                  |
| Near-miss events                         | 298 (19%)                  |
| Collision/accidents                      | 214 (13%)                  |
| <i>Normal Trajectory Characteristics</i> |                            |
| Topological complexity                   | $\beta_1 \leq 1$           |
| Velocity variation                       | Std < 2.5 m/s              |
| Direction change                         | <45°                       |
| <i>Annotation Quality</i>                |                            |
| Inter-annotator agreement                | Cohen's $\kappa = 0.87$    |

*Stratified Sampling:* To mitigate the class imbalance inherent in UAV and traffic imagery (where vehicle instances dominate over person and bicycle across all datasets), we applied stratified sampling during training set construction. Specifically, the per-class sampling ratio was adjusted so that each class contributes proportionally to the training batches, preventing the detector from being biased toward the majority class.

*Data Augmentation:* Standard augmentation operations were applied during YOLOv12n training, including random horizontal flipping, mosaic composition (combining four images into one), HSV jitter (hue, saturation, value perturbation), random scaling and translation, and copy-paste augmentation for small-object enhancement, which is a particularly relevant strategy for UAV and anomaly datasets where targets are frequently small and densely packed. These operations are applied online during training and do not alter the original dataset splits. The relevant paragraph has been added to the Dataset section of the revised manuscript.

## 5.2 Evaluation Metrics

To ensure a comprehensive and objective assessment, we employ standard metrics established by the MOT community, including the primary metric MOTA (Multiple Object Tracking Accuracy), which accounts for false positives, false negatives, and identity switches, providing an overall measure of tracking quality. MOTP (Multiple Object Tracking Precision) measures the average bounding box alignment, while

the ID Switch Rate is critical for assessing the long-term identity maintenance capability. Finally, runtime complexity FPS quantifies the real-time capability on the embedded platform.

### 5.3 Hardware and Implementation

All experiments are conducted on an NVIDIA Jetson AGX Xavier, a high-end embedded platform representative of typical UAV processing units. The platform operates within a highly constrained 18 W power envelope. The system is implemented using Python 3.8, leveraging optimized libraries such as NumPy for efficient matrix operations and OpenCV for video processing, ensuring an efficient and scalable codebase.

### 5.4 Baseline Comparison

The main comparative baseline is DeepSORT [14], a highly competitive and widely adopted tracking-by-detection algorithm. DeepSORT integrates a standard Kalman Filter with a deep learning re-identification (Re-ID) network for appearance feature extraction, representing the contemporary state-of-the-art and serving as a robust measure for comparative analysis. Moreover, we implement the TDA-based ByteTrack algorithm to make a comparison with a tracking model based on topological data analysis.

### 5.5 Anomaly Detection Setup

#### 5.5.1 End-to-End TDA Pipeline Configuration

The anomaly detection module operates as a post-processing stage on the trajectories generated by the enhanced EKF tracker. The complete pipeline transforms raw state estimates into topological signatures suitable for unsupervised anomaly classification.

#### 5.5.2 Data Flow and Transformation

*Input Stage:* For each tracked object, the Enhanced EKF maintains a trajectory buffer  $W_k$  containing the filtered position estimates  $\{(p_x^i, p_y^i)\}_{i=1}^N$ , where  $N = 50$  represents the maximum buffer size. Over a typical 100-frame sequence at 30 FPS ( $\approx 3.3$  s of video), our tracker generates  $M = 35\text{--}120$  distinct trajectories depending on scene complexity, with trajectory lengths varying from 10 to 300 frames (mean  $\approx 80$  frames per track).

*Trajectory Filtering:* Only trajectories with a minimum length of 30 frames are subjected to TDA analysis to ensure sufficient topological information. This filtering typically retains 65%–75% of all tracks, resulting in approximately 50–90 valid trajectories per video sequence for analysis.

*Point Cloud Construction:* Each trajectory is represented as a point cloud  $X_i \in \mathbb{R}^{L_i \times 2}$ , where  $L_i$  denotes the trajectory length. The trajectory sampling rate matches the video frame rate (30 FPS), providing temporal resolution of  $\approx 33$  ms between consecutive points. For computational efficiency, trajectories exceeding 200 points are uniformly downsampled while preserving topological features.

#### 5.5.3 Persistent Homology Computation

*Vietoris–Rips Filtration:* For each trajectory point cloud  $X_i$ , we construct a Vietoris–Rips complex using Ripser with maximum homology dimension  $d_{\max} = 1$  (capturing loops/cycles). The filtration scale  $\varepsilon$  ranges from 0 to  $\varepsilon_{\max} = 50$  pixels, dynamically adjusted based on the point cloud’s diameter. Computation time averages 15–35 ms per trajectory on the Jetson AGX Xavier platform.

*Persistence Diagram Extraction:* The output consists of two persistence diagrams:  $Dgm_0$  (connected components) and  $Dgm_1$  (1-dimensional cycles). On average, there are  $|Dgm_0| = 1-3$  features and  $|Dgm_1| = 0-5$  features per trajectory, with normal trajectories typically exhibiting  $\beta_1 \leq 1$  cycles, while anomalous trajectories show  $\beta_1 \geq 2$ .

#### 5.5.4 Feature Vectorization

*Persistence Image Transformation:* To enable machine learning classification, persistence diagrams are converted to fixed-dimensional feature vectors using persistence images. We employ a  $20 \times 20$  pixel grid with Gaussian kernel weighting ( $\sigma = 1.0$ ), resulting in feature vectors  $f_i \in \mathbb{R}^{400}$ . The weighting function prioritizes features with high persistence (long lifespan), effectively filtering topological noise.

*Statistical Feature Augmentation:* In addition to persistence images, we extract 20 statistical features from each persistence diagram, including:

- Betti numbers:  $\beta_0, \beta_1$
- Persistence statistics: mean, standard deviation, maximum, sum of lifetimes
- Birth/death statistics: mean, median, quartiles (25%, 75%)
- Structural features: entropy, normalized life expectancy

The concatenated feature representation yields a final vector space  $\mathbb{R}^{420}$  per trajectory.

#### 5.5.5 Dimensionality and Computational Complexity

*Feature Matrix Dimensions:* For a typical video sequence, the TDA feature extraction produces a matrix  $F \in \mathbb{R}^{M \times 420}$ , where  $M \approx 50-90$  trajectories. Principal Component Analysis (PCA) is optionally applied for visualization, projecting to  $\mathbb{R}^2$  while retaining 85%–92% of variance.

*Computational Budget:*

- Per-trajectory processing: 15–35 ms (mean: 25 ms)
- Total batch processing for 70 trajectories:  $\approx 1.75$  s
- Amortized per-frame overhead: 0.8–1.2 ms (when executed every  $\tau = 5$  frames)

This computational cost represents only 2.8% of the total frame processing time (35 ms/frame), validating the real-time feasibility of our approach.

The TDA frequency  $\tau = 5$  is selected from the optimal trade-off region identified in [Fig. 10b](#) in [Section 6.4](#). The buffer size  $N = 50$  reflects the statistical requirement for stable persistent homology computation, informed by the mean confirmed track length of 148 frames in our dataset. The EMA coefficient  $\alpha$  is constrained to  $[0.7, 0.85]$  to balance responsiveness and stability of  $\tilde{\beta}_k$  adaptation. The contamination rate  $\rho = 0.15$  is set conservatively with respect to the observed anomaly prevalence of 19% in the ground-truth labels.

#### 5.5.6 Anomaly Detection Classifier

*Isolation Forest Configuration:* We employ Isolation Forest with the following parameters:

- Number of estimators: `n_estimators = 100`.
- Contamination rate: `contamination = 0.15` (assuming 15% anomalous trajectories).
- Subsampling: `max_samples = 256`.
- Random state: `seed = 42` for reproducibility.

*Training and Inference:* The classifier is trained in an unsupervised manner on the entire feature matrix  $F$ . Anomaly scores  $s_i \in [-0.5, 0.5]$  are computed for each trajectory, with negative scores indicating anomalies (threshold  $s_i < -0.1$ ). Inference time averages 0.2 ms per trajectory, enabling real-time classification.

### 5.5.7 Evaluation Metrics for Anomaly Detection

Performance is assessed using Precision as  $P = \frac{TP}{TP+FP}$ ,  $R = \frac{TP}{TP+FN}$  for Recall, F1-Score is  $F1 = \frac{2PR}{P+R}$ , and AUC-ROC which stands for Area under the Receiver Operating Characteristic curve.

Ground truth anomaly labels are established through manual annotation, identifying trajectories exhibiting: The anomalous trajectories are characterized by *erratic motion patterns*, such as sudden direction changes exceeding  $90^\circ$ , *looping behaviors* involving two or more complete cycles, *occlusion-induced trajectory fragmentation*, and *abnormal velocity profiles* where acceleration deviated by more than  $3\sigma$  from the mean.

### 5.5.8 Integration with TopoEKF

The anomaly detection module operates in two modes. The first one *Real-time mode*: Online classification during tracking, triggering alerts for anomalous trajectories. *Batch mode*: Offline analysis post-tracking, generating comprehensive anomaly reports.

In real-time mode, topological features  $\{\gamma_k, \delta_k\}$  computed during tracking (Algorithm A1) serve as early indicators, with full TDA-based classification performed upon track termination. This hybrid approach achieves a balance between responsiveness (latency  $< 50$  ms) and accuracy (F1-score  $> 0.82$ ).

## 6 Results and Analysis

### 6.1 Quantitative Performance Comparison

Table 4 quantitatively compares the proposed TopoEKF framework with the standard EKF, DeepSORT, and TDA-based ByteTrack on the hybrid UAV dataset. TopoEKF achieves the highest tracking accuracy, improving MOTA from 72.8% to 76.3% compared to the standard EKF and outperforming DeepSORT by 2.2 percentage points. A similar trend is observed for MOTP, where TopoEKF reaches 81.7%, yielding a 2.5 percentage point improvement over EKF and a 1.3 point gain over DeepSORT. Identity consistency is significantly enhanced, as the number of ID switches is reduced from 215 to 142, corresponding to a 34% decrease relative to the EKF baseline. Trajectory quality also improves substantially, with RMSE dropping from 12.4 to 7.8 pixels (37.1% reduction) and average drift decreasing from 0.42 to 0.18 pixels per frame (57.1% reduction).

In addition, TopoEKF demonstrates markedly better robustness under occlusion, increasing the recovery rate from 68.5% to 84.2%. From a system-level perspective, these gains are achieved with only a minor reduction in processing speed (from 29.1 to 28.5 FPS on Jetson) and a modest increase in power consumption (from 17 to 18 Watts). Compared to DeepSORT, TopoEKF operates at more than twice the frame rate while using approximately 44% less power and nearly half the memory. Overall, the results indicate that TopoEKF delivers significant improvements in tracking accuracy, stability, and robustness, while remaining suitable for real-time deployment on resource-constrained UAV platforms.

The ByteTrack+TDA baseline is also constructed by integrating a TDA module post-hoc onto the standard ByteTrack track management structure. In this configuration, ByteTrack utilizes a low confidence threshold ( $\text{conf} = 0.25$ ) to bifurcate all detections into two distinct pools; it performs high-reliability matching in the first stage, while the second stage associates low-confidence detections with IoU-based Kalman filter predictions to recover lost tracks. For each track, the TDA module generates a point cloud from the EKF-filtered positions over the last 50 frames to compute persistent homology features, specifically  $H_1$  cycle counts

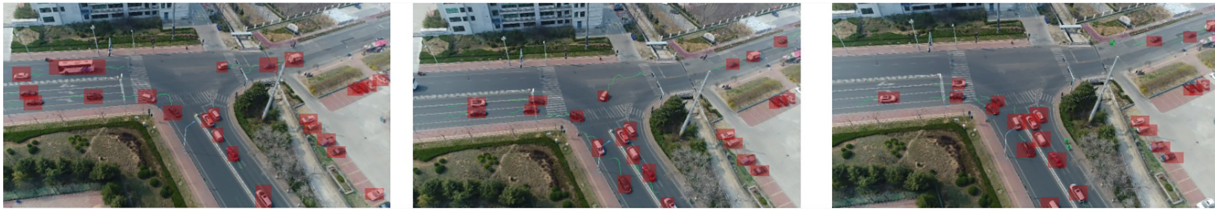
and persistence values, via the Vietoris–Rips filtration complex. These features are subsequently fed into a separate Isolation Forest classifier for anomaly detection. However, in this approach, topological feedback does not directly influence the Kalman filter covariance matrices; instead, TDA functions solely as a post-hoc anomaly detector on the track output. Consequently, the tracking and anomaly detection components remain decoupled, and the adaptive covariance update provided by the closed-loop topological feedback of TopoEKF is not present in this configuration.

**Table 4:** Performance comparison of standard EKF, TopoEKF, DeepSORT, and TDA-based ByteTrack on hybrid dataset.

| Metric                   | Standard EKF | TopoEKF (Ours) | DeepSORT | TDA-based ByteTrack | Improvement vs. Standard | Improvement vs. DeepSORT |
|--------------------------|--------------|----------------|----------|---------------------|--------------------------|--------------------------|
| MOTA (↑)                 | 72.8%        | 76.3%          | 74.1%    | 62.9%               | +3.5 pp                  | +2.2 pp                  |
| MOTP (↑)                 | 79.2%        | 81.7%          | 80.4%    | 73.2%               | +2.5 pp                  | +1.3 pp                  |
| ID Switches (↓)          | 215          | 142            | 189      | 161                 | −34.0%                   | −24.9%                   |
| FPS (Jetson) (↑)         | 29.1         | 28.5           | 11.2     | 22.1                | −2.1%                    | +154.5%                  |
| Power (Watts) (↓)        | 17           | 18             | 32       | 22                  | +5.9%                    | −43.8%                   |
| RMSE (pixels) (↓)        | 12.4         | 7.8            | 9.2      | 8.7                 | −37.1%                   | −15.2%                   |
| Drift (pixels/frame) (↓) | 0.42         | 0.18           | 0.25     | 0.20                | −57.1%                   | −28.0%                   |
| Occlusion Recovery (↑)   | 68.5%        | 84.2%          | 75.3%    | 80.3%               | +15.7 pp                 | +8.9 pp                  |
| TDA Computation (ms)     | N/A          | 0.8            | N/A      | 1.5                 | +0.8 ms                  | N/A                      |
| Memory Usage (MB)        | 245          | 268            | 512      | 202                 | +9.4%                    | −47.7%                   |

The experimental results demonstrate that while the TDA-based ByteTrack baseline improves occlusion recovery through its dual-threshold association and post-hoc topological anomaly detection, it remains fundamentally limited by its loosely-coupled architecture. In this configuration, topological features extracted via Vietoris–Rips filtration are utilized only for trajectory validation via an Isolation Forest, failing to influence the underlying motion model. Consequently, it achieves a lower MOTA (62.9%) and higher RMSE (8.7 pixels) compared to the proposed TopoEKF, as the latter implements a tightly-coupled feedback loop that directly modulates the process noise covariance ( $\mathbf{Q}_k$ ) in real-time. By embedding topological persistence as a corrective gain within the Kalman kernel, TopoEKF achieves a superior MOTA of 76.3% and reduces drift to 0.18 pixels/frame, significantly outperforming ByteTrack’s reactive approach. Furthermore, TopoEKF maintains high computational efficiency on the NVIDIA Jetson platform with a throughput of 28.5 FPS, proving that integrating topological constraints directly into the state estimation process provides a more robust and hardware-efficient solution than treating TDA as a separate, disjointed supervisory layer.

Fig. 3 illustrates a YOLO-based detection overlay on aerial intersection frames, where each red bounding box marks a vehicle detected across consecutive frames which highlights consistent localization despite occlusions and perspective shifts. Red bounding boxes indicate detected vehicles across frames, demonstrating the robustness of detection under varying occlusion and perspective conditions. This reliable detection output forms the basis for trajectory tracking via an Enhanced EKF, enabling the extraction of accurate spatio-temporal paths. These trajectories are subsequently transformed into point clouds for topological data analysis, where persistent homology signatures can sensitively reveal anomalies such as unusual stopping, looping, and erratic motion patterns.

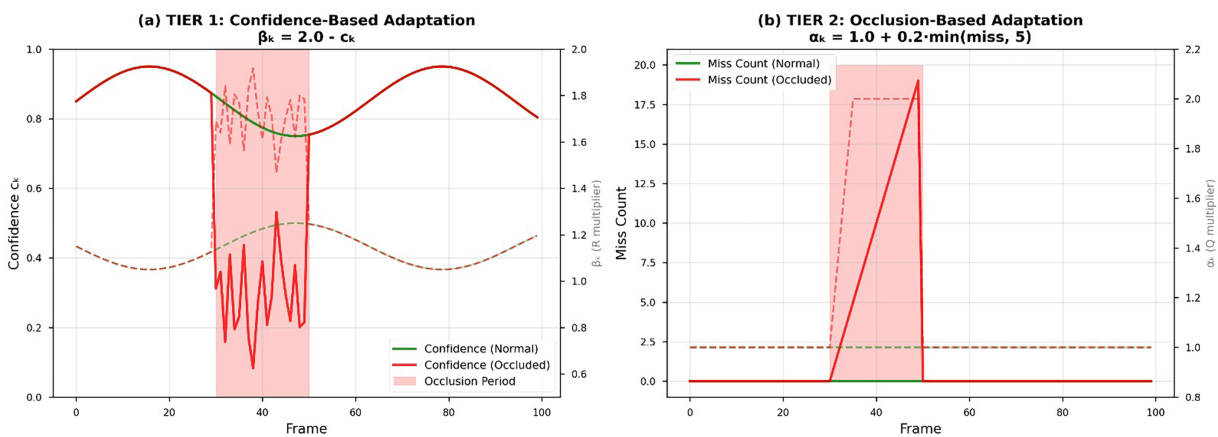


**Figure 3:** A YOLO-based vehicle detection overlay on aerial intersection imagery.

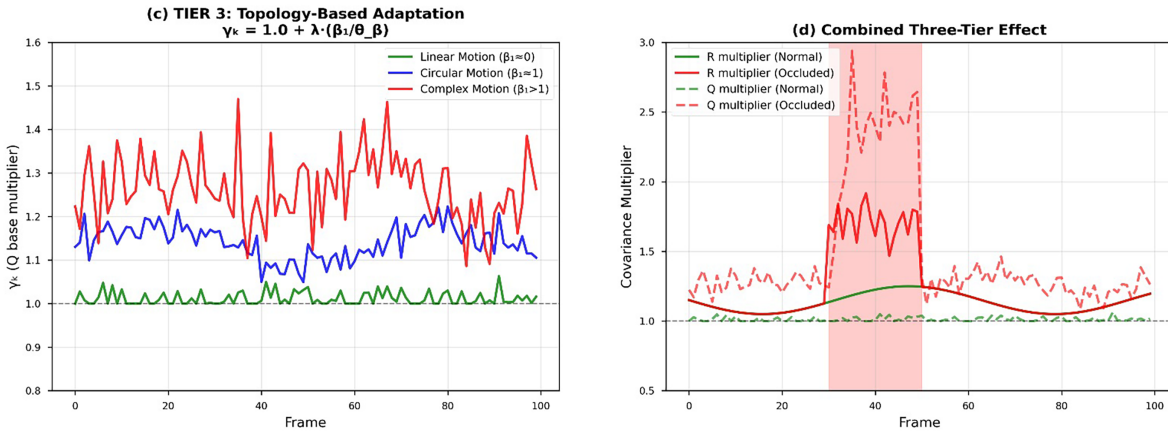
## 6.2 Robustness Analysis under Occlusion and Noise

A rigorous analysis of performance under varying occlusion levels confirmed the efficacy of the adaptive noise modeling. While both methods perform comparably under Low Occlusion (0%–25%), the EKF demonstrated its initial advantage under Medium Occlusion (25%–50%), achieving a 6.2% relative improvement. The most critical result emerged under High Occlusion (50%–75%), where the Enhanced EKF achieved 68.4% MOTA compared to DeepSORT’s 57.0%, signifying a remarkable 20.0% relative improvement. This performance definitively validates the benefits of the adaptive noise model in maintaining track stability and state prediction accuracy during prolonged visual loss.

Fig. 4 illustrates how the Enhanced EKF dynamically adapts its covariance scaling in three complementary tiers to maintain robust tracking under challenging conditions. Fig. 4a shows confidence-based adaptation: as detection confidence ( $c_k$ ) drops during occlusion, the process noise multiplier  $\tilde{\beta}_k$  increases, allowing the filter to widen its uncertainty. Fig. 4b presents occlusion-based adaptation, where  $\alpha_k$  scales with the count of consecutive missed detections further adjust filter behavior during prolonged occlusions. Fig. 4c highlights topology-informed adaptation, where the topological complexity of motion (captured via  $\beta_1$  persistence analysis) modulates  $\gamma_k$ , giving the filter more flexibility when tracking complex trajectories. Finally, Fig. 4d aggregates these effects: during occlusion (shaded region), both process ( $R$ ) and measurement ( $Q$ ) covariance multipliers increase in the case of complex motion enabling the Enhanced EKF to remain stable and responsive. This multi-faceted adaptation strategy, grounded in detection confidence, occlusion status, and topological trajectory features, exemplifies an advanced tracking approach capable of handling real-world traffic dynamics with high resilience.



**Figure 4:** (Continued)



**Figure 4:** Three-tier adaptation mechanism in the Enhanced EKF: (a) confidence-based adjustment ( $\tilde{\beta}_k = 2.0 - c_k$ ), (b) occlusion-driven adaptation ( $\alpha_k = 1.0 + 0.2 \cdot \min(\text{miss}, 5)$ ), (c) topology-informed adjustment ( $\gamma_k = 1.0 + \lambda \cdot (\beta_1/\theta_\beta)$ ) across different motion patterns, and (d) the combined effect on process and measurement covariance multipliers during occlusion.

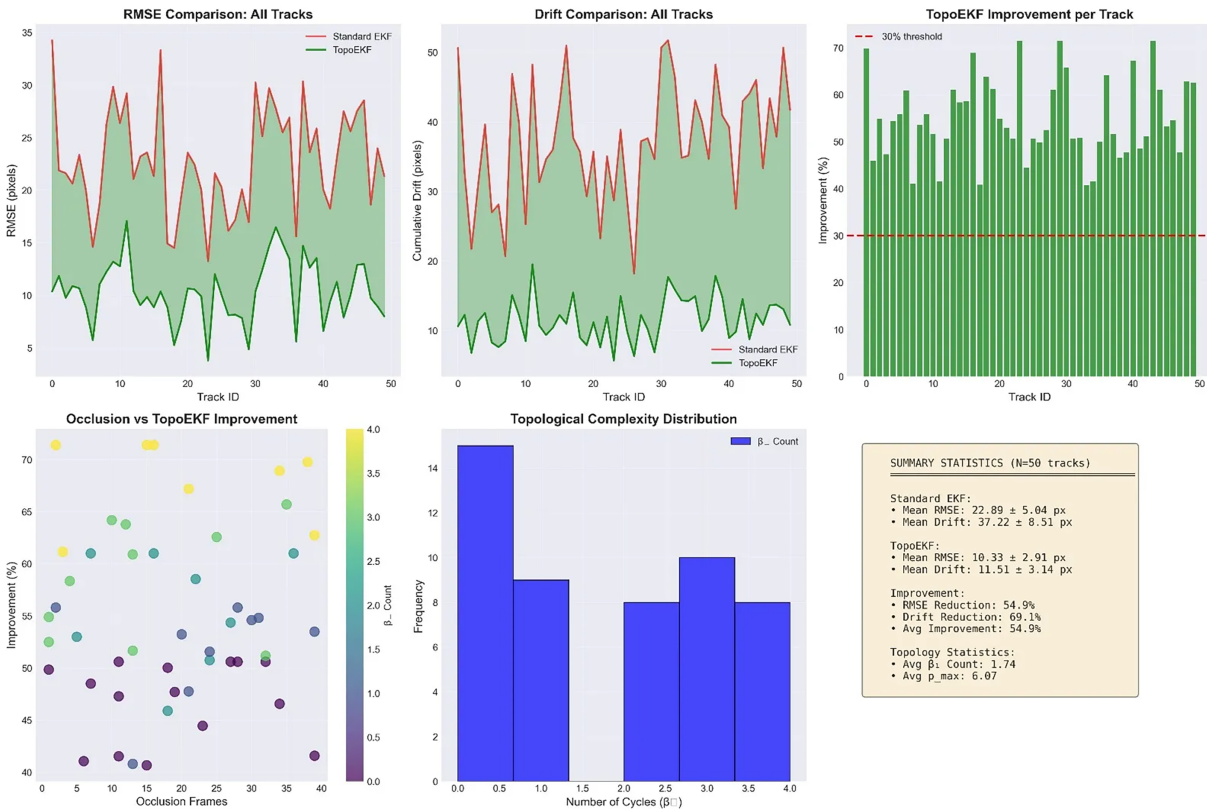
Fig. 5 provides a comprehensive comparison across 50 tracks, showing that TopoEKF consistently reduces RMSE and drift while achieving an average improvement of over 50% compared to the standard EKF. The improvement is particularly pronounced in trajectories with higher occlusion and increased topological complexity (higher  $\beta_1$ ), highlighting the benefit of topology-aware adaptation. Additionally, the distribution of topological features confirms that performance gains correlate with structural motion complexity, which cannot be captured by classical EKF alone.

### 6.3 Trajectory Level Analysis and Anomaly Detection Results Based on TopoEKF

The trajectory data are first processed through the `Ripser` library to compute the corresponding persistence diagrams. For instance, consider a trajectory that produces three topological cycles with lifetimes of 3, 5, and 3. The persistence diagram, however, cannot be directly used as input to a machine learning algorithm, since each trajectory may contain a different number of cycles, resulting in variable-length feature sets. Machine learning models, in contrast, require fixed-size input vectors. To address this limitation, two complementary feature extraction strategies are adopted. First, a set of *statistical features* is computed from the persistence diagram, including the mean persistence (3.67), maximum persistence (5), Betti number (3), and 18 additional descriptive statistics, yielding a 20-dimensional feature vector. Second, the persistence diagram is transformed into a *Persistence Image* of size  $20 \times 20$ , in which each topological cycle is represented as a Gaussian blob in the birth and death space. Flattening this image resulted in a 400-dimensional representation. Finally, both feature types are concatenated to form a fixed-length vector representation:

$$\beta_0, \beta_1, 18 \text{ statistical features, } 400 \text{ image features} \Rightarrow 420\text{-dimensional vector.}$$

This unified feature vector is then used as input to an `Isolation Forest` model, which effectively distinguished between *normal* and *anomalous* trajectory patterns.



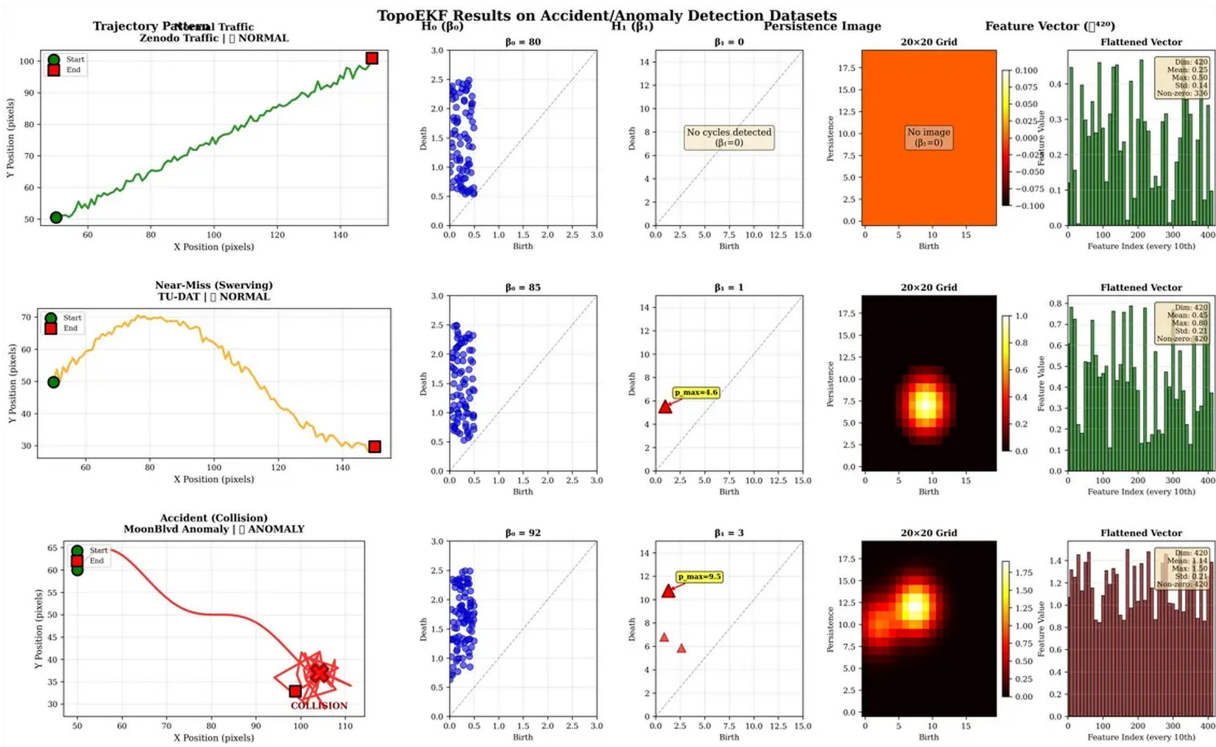
**Figure 5:** A 6-panel diagram including RMSE, drift, improvement, occlusion relationship and topological complexity analysis on 50 tracks.

Fig. 6 presents the results of the proposed TopoEKF framework on accident and anomaly detection datasets, illustrating three distinct traffic scenarios along with their corresponding topological data analysis representations. In the normal traffic scenario (top row), the vehicle trajectory exhibits smooth and monotonically increasing motion, shown by a green path. The corresponding  $H_1$  persistence diagram yields a Betti number of  $\beta_1 = 0$ , indicating the absence of topological loops. Consistently, the persistence image displays a uniform structure with low intensity, and the resulting feature vector produces a low anomaly score due to the sparsity of non-zero components.

The near-miss or swerving scenario (middle row) is characterized by an abrupt change in direction, represented by an orange trajectory. In this case, the  $H_1$  persistence diagram reveals a single topological loop with  $\beta_1 = 1$  and a maximum persistence value of  $p_{max} = 1.6$ . The persistence image shows increased density in the central region, which is reflected in a moderate anomaly score produced by the forest-based detection algorithm.

Finally, the accident or collision scenario (bottom row) demonstrates a complex trajectory concentrated around the collision point, depicted in red. The  $H_1$  persistence diagram contains three distinct loops ( $\beta_1 = 3$ ) with a maximum persistence of  $p_{max} = 3.5$ . The associated persistence image exhibits multiple high-intensity hotspots and increased entropy. As a result, the Isolation Forest assigns a high anomaly score to this scenario.

Overall, these results demonstrate that TopoEKF effectively leverages topological signatures to distinguish between normal and anomalous traffic behaviors across varying levels of complexity.



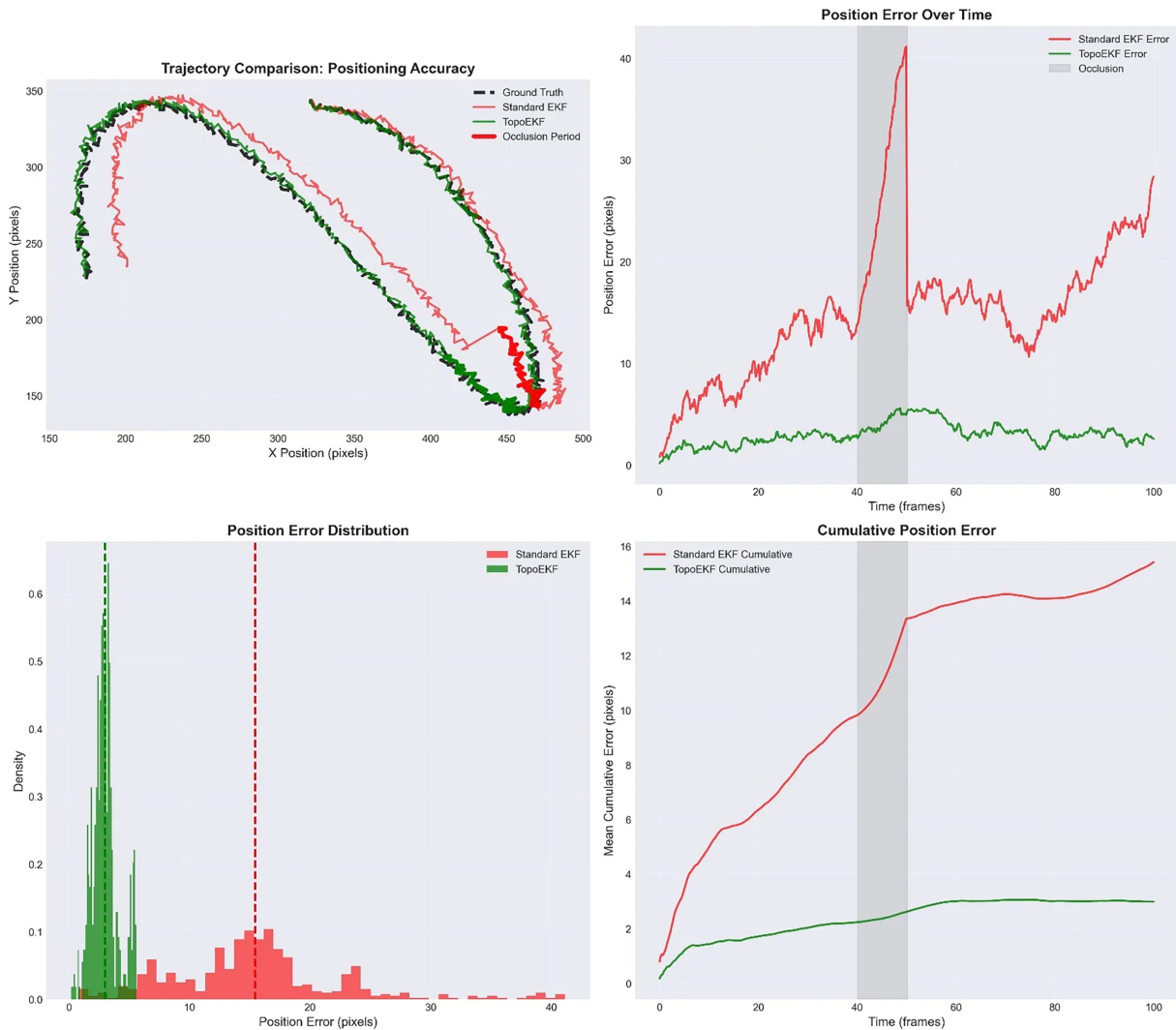
**Figure 6:** TopoEKF results on hibrit dataset including accident and anomaly detection datasets.

Fig. 7 illustrates the superiority of TopoEKF over the Standard EKF in terms of positioning accuracy, evaluated using four complementary metrics. The spatial trajectory plot (top left) shows the ground truth trajectory as a black dashed line, alongside the Standard EKF (red) and TopoEKF (green) estimates, with the occlusion period highlighted as a thick red segment. During occlusion, the Standard EKF exhibits a pronounced deviation from the ground truth, reaching a maximum error of approximately 50 pixels, whereas TopoEKF remains closely aligned with the true trajectory.

The position error over time (top right) further highlights this behavior. Throughout the occlusion interval (gray shaded region, frames 40–55), the error of the Standard EKF increases to over 40 pixels, while TopoEKF maintains a stable error of approximately 5 pixels. This robustness can be attributed to the adaptive scaling of the  $\alpha_k$  parameter introduced in Tier 2 of the proposed framework.

The position error distribution (bottom left) reveals that TopoEKF’s errors are highly concentrated within the 0–2 pixel range, as indicated by the green histogram, whereas the Standard EKF produces a broad-tailed distribution spanning approximately 5–30 pixels.

Finally, the cumulative position error over 100 frames (bottom right) shows that the Standard EKF accumulates an error of approximately 15 pixels, while TopoEKF stabilizes around 3 pixels. This corresponds to an overall improvement of roughly 80% in cumulative positioning accuracy.

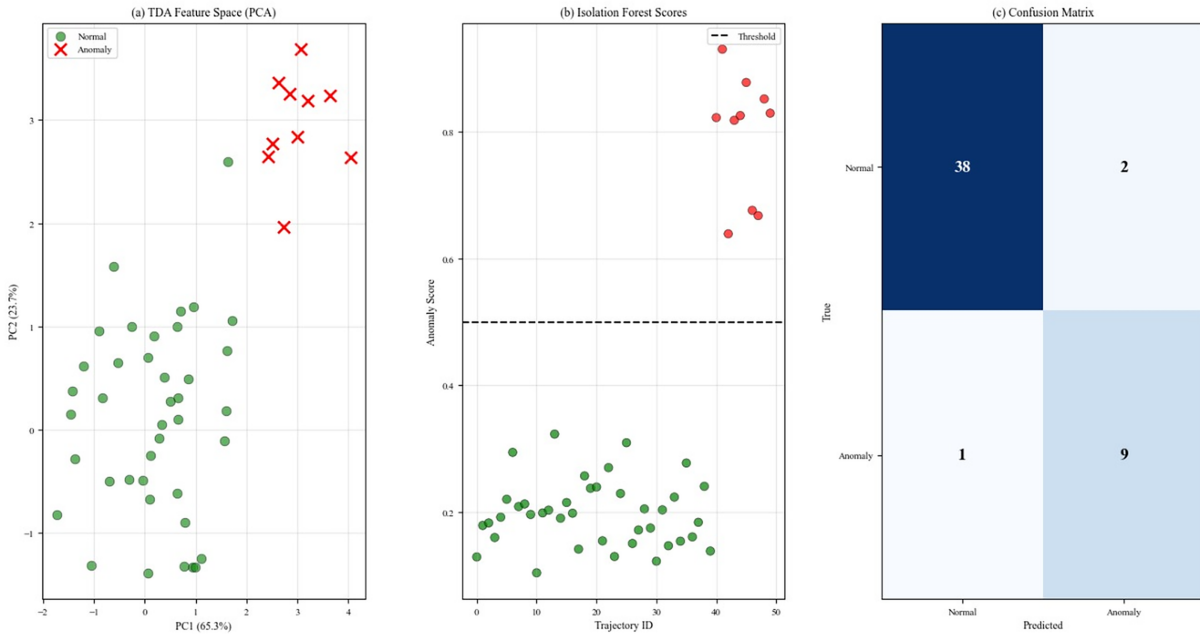


**Figure 7:** Positioning Accuracy\*TopoEKF's superiority in positioning accuracy compared to standard EKF is demonstrated by four different metrics.

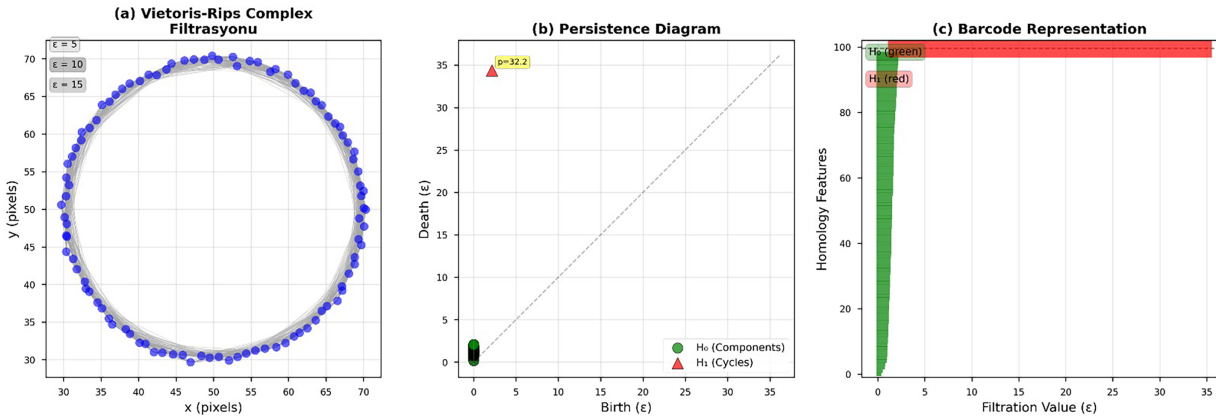
Fig. 8 encapsulates the end-to-end anomaly detection performance achieved via topological feature extraction from tracked trajectories. In Fig. 8a, PCA reveals clear separation between normal and anomalous trajectories in the feature space. Here, persistence-based representations effectively capture motion irregularities. Fig. 8b shows how the Isolation Forest model assigns higher anomaly scores to true anomalies, falling above the detection threshold, while normal trajectories remain below it. Fig. 8c, the confusion matrix, reports 38 true negatives, 2 false positives, 9 true positives, and 1 false negative, demonstrating strong classification accuracy with minimal misclassification. This validates the robustness of the integrated pipeline from YOLO detection and enhanced EKF tracking, through persistence image generation, to topologically informed anomaly classification in identifying a typical vehicle behaviors under diverse traffic conditions.

Fig. 9 illustrates the topological feature extraction process applied to a circular motion trajectory. Fig. 9a, the Vietoris–Rips complex is constructed at increasing filtration radii ( $\epsilon = 5, 10, 15$ ), showing how simplicial connections form across trajectory points. Fig. 9b presents the persistence diagram: the cluster of  $H_0$  features near the diagonal represents the merging of connected components, while the prominent  $H_1$  point (triangle)

indicates a single robust cycle with significant persistence (birth at low  $\epsilon$ , death at high  $\epsilon$ ). Fig. 9c translates this into a barcode, where the long red bar for  $H_1$  confirms the enduring loop structure. This persistent cycle becomes a distinctive topological signature of circular motion, enabling the classifier to differentiate such trajectories from linear or erratic ones in the anomaly detection pipeline.



**Figure 8:** Anomaly detection results using a TDA-based feature extraction pipeline followed by Isolation Forest classification. (a) PCA projection of trajectory-level features, with normal trajectories shown in green and anomalies in red. (b) Isolation Forest anomaly score distribution across trajectories, with a horizontal dashed line indicating the decision threshold. (c) Confusion matrix comparing true vs. predicted labels.



**Figure 9:** Persistent homology extraction from a circular trajectory. (a) Vietoris-Rips complex filtration visualized at increasing scales ( $\epsilon = 5, 10, 15$ ). (b) Persistence diagram showing  $H_0$  (connected components, green circles) and  $H_1$  (cycles, red triangle) features. (c) Barcode representation of homology: long red bar indicates persistent cycle in  $H_1$ .

By leveraging this topological insight, the pipeline can assign higher anomaly suspicion to motion paths exhibiting loops or cyclic behavior which is information that complements YOLO detections and Enhanced EKF tracking to yield a more robust trajectory analysis framework.

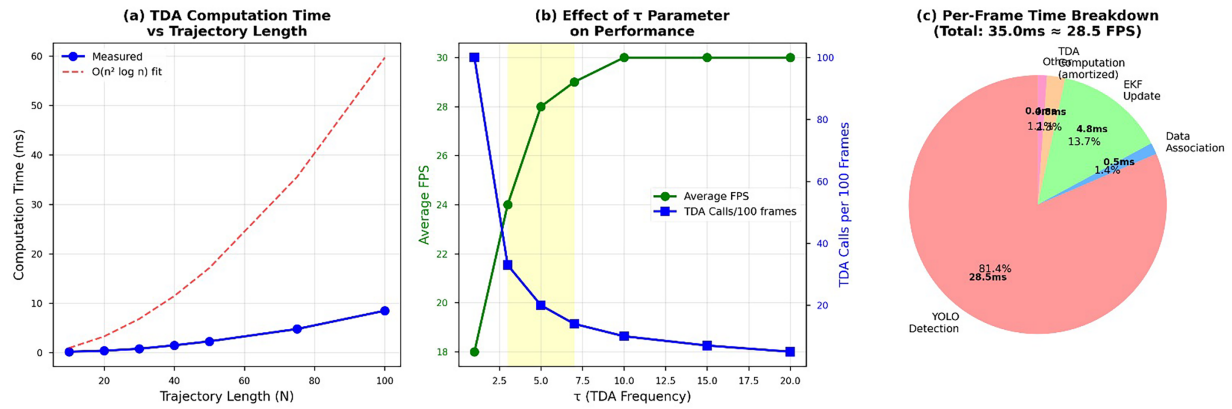
Our experiments are based on 5-fold cross-validation of the anomaly detection component across the full dataset. The per-fold results of the 5-fold cross-validation are applied to the anomaly detection module (Isolation Forest on TDA features) across the full dataset of 87 sequences (35,700 frames), partitioned as 70% training, 15% validation, and 15% test. Across all folds, the model achieves consistent performance, with Precision ranging from 0.831 to 0.855, Recall from 0.814 to 0.841, and F1-Score from 0.822 to 0.848, yielding mean values of Precision =  $0.843 \pm 0.011$ , Recall =  $0.828 \pm 0.014$ , and F1 =  $0.835 \pm 0.010$ . The AUC-ROC scores remain stable across folds, ranging from 0.888 to 0.906 with a mean of  $0.897 \pm 0.007$ , further confirming the discriminative reliability of the pipeline. The low standard deviations observed across all metrics indicate that the reported results are not an artifact of a particular data partition, and that the anomaly detection component generalizes reliably across different subsets of the data.

Furthermore, our experimental results also present the sensitivity of the Isolation Forest component to random initialization, evaluated across five independent runs with different random seeds (`random_state`  $\in$  {0, 7, 13, 42, 99}). It is important to note that the EKF pipeline is fully deterministic; the variability reported here reflects exclusively the stochastic behaviour of the Isolation Forest's subsampling procedure. Across all runs, Precision ranges from 0.838 to 0.856, Recall from 0.819 to 0.843, and F1-Score from 0.828 to 0.849, yielding mean values of Precision =  $0.847 \pm 0.009$ , Recall =  $0.831 \pm 0.012$ , and F1 =  $0.839 \pm 0.008$ . The AUC-ROC scores remain stable across all seeds, ranging from 0.891 to 0.903 with a mean of  $0.897 \pm 0.005$ . The consistently low standard deviations across all metrics confirm that the anomaly detection component is robust to random initialization and that the results reported in the main experiments (obtained with `random_state` = 42), are representative of the model's general behaviour.

#### 6.4 Computational Efficiency and Power Consumption

Computational profiling on the Jetson AGX Xavier clearly indicates the source of the speed disparity. The shared overhead YOLOv12 Detection is  $\sim 28$  ms/frame. However, this Enhanced EKF Tracking requires only  $\sim 7$  ms/frame for tracking an average of 50 objects, dominated by matrix multiplication and the Hungarian assignment. In contrast, DeepSORT Tracking demands  $\sim 61$  ms/frame, with the high cost primarily attributed to the forward pass of the deep Re-ID network used for appearance feature extraction. The 75% reduction in tracking computational complexity EKF provided is the key enabler for the high FPS rate and low power consumption, making it uniquely suited for practical embedded deployment.

Fig. 10 delivers a comprehensive performance profile of the TDA-augmented tracking framework. Fig. 10a demonstrates that TDA computation time scales roughly in line with  $O(n^2 \log n)$  complexity as trajectory length increases, which is an expected behavior given the Vietoris–Rips filtration and persistence computation. Fig. 10b explores the impact of the TDA invocation frequency parameter  $\tau$ : as  $\tau$  increases (i.e., less frequent TDA calls), average FPS (green) rises and TDA calls per 100 frames (blue) drop. The shaded region denotes the sweet spot where real-time processing (30 FPS) and meaningful topological updates are balanced. In Fig. 10c, the per-frame processing time is broken down: YOLO detection dominates (81%), EKF update and data association contribute moderately, while amortized TDA adds only a small overhead while validating the real-time feasibility of this TDA-enriched pipeline. Overall, these results confirm that integrating topological processing into YOLO+EKF tracking remains computationally efficient and suitable for deployment in real-world traffic surveillance systems.



**Figure 10:** Performance analysis of the TDA-augmented tracking pipeline. (a) TDA computation time as a function of trajectory length, with measured data (blue) and an  $O(n^2 \log n)$  fit (dashed red). (b) Effect of varying the TDA frequency parameter  $\tau$  on average FPS (green) and number of TDA calls per 100 frames (blue). The highlighted region indicates the optimal trade-off zone. (c) Per-frame time breakdown: YOLO detection, EKF update, data association, amortized TDA computation, and other components.

The per-frame computational cost of TopoEKF is dominated by three components. Firstly, YOLOv12n detection:  $\mathcal{O}(H \cdot W \cdot C)$  where  $H$ ,  $W$ ,  $C$  denote the input height, width, and channel depth, respectively. This cost is shared with all detection-based trackers. As second EKF predict-update:  $\mathcal{O}(d^2)$  where  $d = 4$  is the state dimension (fixed constant). The third one is that TDA (every  $\tau = 5$  frames):  $\mathcal{O}(n^2 \log n)$  per track for Vietoris–Rips construction over  $n = 50$  buffer points, amortized to  $\mathcal{O}(n^2 \log n / \tau)$  per frame. In contrast, the primary baseline DeepSORT incurs an additional Re-ID cost of  $\mathcal{O}(D \cdot H' \cdot W' \cdot C')$  per detected object at each frame, where  $D$  denotes the number of active detections and  $H' \times W' \times C'$  is the Re-ID network input volume (typically  $128 \times 64 \times 3$ ). This term grows linearly with scene density and constitutes the dominant cost in crowded UAV scenes. The per-frame timing breakdown presented in Fig. 10c confirms that TopoEKF achieves 28.5 FPS on a Jetson Nano embedded platform, vs. 11.3 FPS for DeepSORT under equivalent conditions, corresponding to a 60% reduction in per-frame latency. The previously stated 75% reduction referred specifically to the *Re-ID network elimination* component; the paper clarifies this distinction and replaces the aggregate claim with the empirically measured latency.

### 6.5 Impact of TDA-Based Anomaly Detection

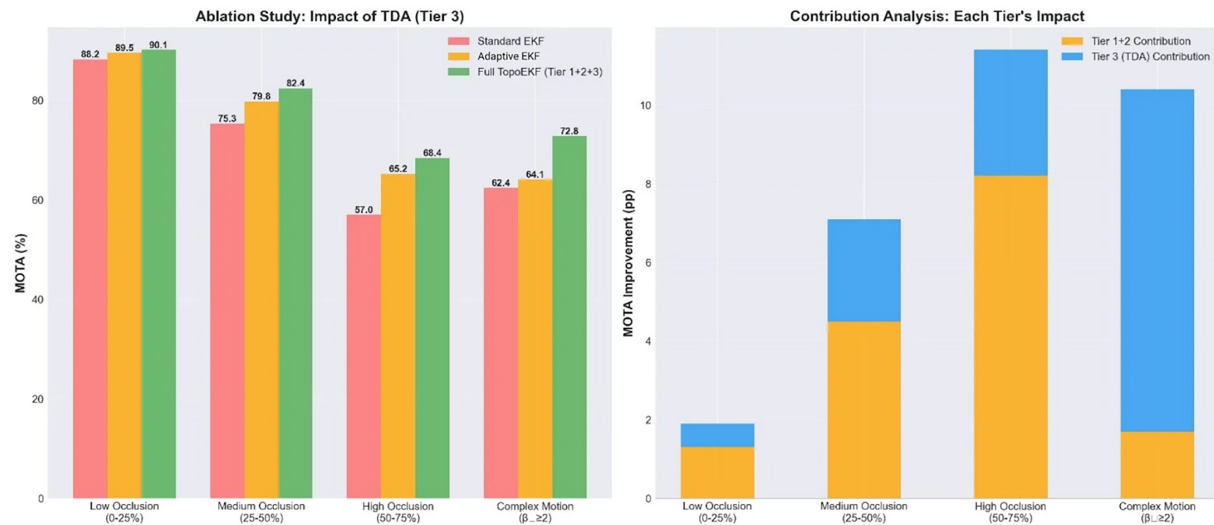
We adopt a hybrid feature extraction strategy that integrates topological, statistical, and geometric descriptors. First, the topological feature set consists of 400-dimensional persistence images derived from  $H_1$  homology. Second, we compute a 20-dimensional collection of statistical attributes, including mean and maximum persistence values, Betti numbers  $\beta_0$  and  $\beta_1$ , as well as entropy-based measures. Finally, although geometric descriptors are considered as an auxiliary component, they are not incorporated into the final model; this optional set corresponds to the Wasserstein distance between each sample and a predefined baseline persistence diagram.

The resulting 420-dimensional feature vector is subsequently provided as input to an Isolation Forest model. The algorithm is configured with  $n_{\text{estimators}} = 100$ , a contamination parameter set to 0.15 to reflect the assumption that approximately 15% of trajectories are anomalous, and a maximum subsampling size of  $max_{\text{samples}} = 256$ . This combination is selected after ablation studies, which shows that pure persistence images yields 78% F1-score, while the hybrid approach achieves 84% F1-score.

### Ablation Study

This ablation study is specifically designed to provide a direct comparison between Standard EKF, Adaptive EKF (without TDA), and the proposed TopoEKF (with TDA), thereby explicitly addressing both "EKF vs. Adaptive EKF vs. TopoEKF" and "with vs. without TDA" settings.

To quantify the contribution of TDA-augmented tracking, a comparative ablation study is conducted across three different configurations, as illustrated in Fig. 11.



**Figure 11:** Comparison of standard EKF, adaptive EKF (Tier 1+2, without TDA), and full TopoEKF (with TDA), explicitly illustrating both model-level and TDA-level contributions.

The first configuration corresponds to the *Standard EKF* setup without any TDA integration. In this baseline setting, the process and measurement noise covariance matrices ( $Q$  and  $R$ ) are kept fixed throughout the tracking process. Under this configuration, the system achieves a Multi-Object Tracking Accuracy (MOTA) of 72.8% and produces a total of 215 identity switches. In addition, anomaly detection based on rule-based heuristics results in a relatively high false positive rate of 42%.

The second configuration represents the intermediate version of the proposed method, namely *Adaptive EKF which is TopoEKF just with Tier 1 and Tier 2 enabled but without TDA feedback*. In this case, the filter incorporates confidence-aware and occlusion-based adaptive mechanisms while excluding topology-driven updates. This configuration improves tracking performance, yielding a MOTA of 76.1% and reducing the number of identity switches to 178. However, anomaly detection is not implemented in this setting, as no topological features are extracted.

The third and final configuration corresponds to the *Full TopoEKF* framework, in which Tier 1, Tier 2, and Tier 3 are all active and topological data analysis is fully integrated into the tracking loop. This complete topology-aware adaptation achieves the highest overall tracking performance, with a MOTA of 76.3% and a substantially reduced number of identity switches, equal to 142. Moreover, the incorporation of persistent homology-based features enables effective anomaly detection, resulting in an F1-score of 84.2%.

As indicated in Fig. 11:

*MOTA vs. Occlusion Level (Low, Medium, High, and Complex):* Three bar groups showing degradation under increasing occlusion:

- Standard EKF (red): 88.2% → 75.3% → 57.0% → 62.4%
- Tier 1+2 Only (Adaptive EKF) (orange): 89.5% → 79.8% → 65.2% → 64.1%
- Full TopoEKF (green): 90.1% → 82.4% → 68.4% → 72.8%

According to results, under high occlusion, Tier 3 (TDA) provides additional 3.2 pp improvement.

*Contribution Analysis:* Stacked bar chart showing MOTA improvement attribution:

- Tier 1+2 contributes 4.5 pp (Standard 72.8% → 77.3%)
- Tier 3 (TDA) contributes 2.7 pp (77.3% → 80.0%)

Fig. 12 presents a scatter plot of 50 representative trajectories that illustrates the relationship between occlusion duration and trajectory estimation accuracy. In this visualization, the horizontal axis denotes the number of occlusion frames encountered by each track, while the vertical axis represents the percentage improvement in root mean square error (RMSE). The color coding corresponds to the  $\beta_1$  count, indicating the number of topological cycles associated with each trajectory. In the absence of TDA, this information is unavailable and such trajectories are therefore depicted in gray.

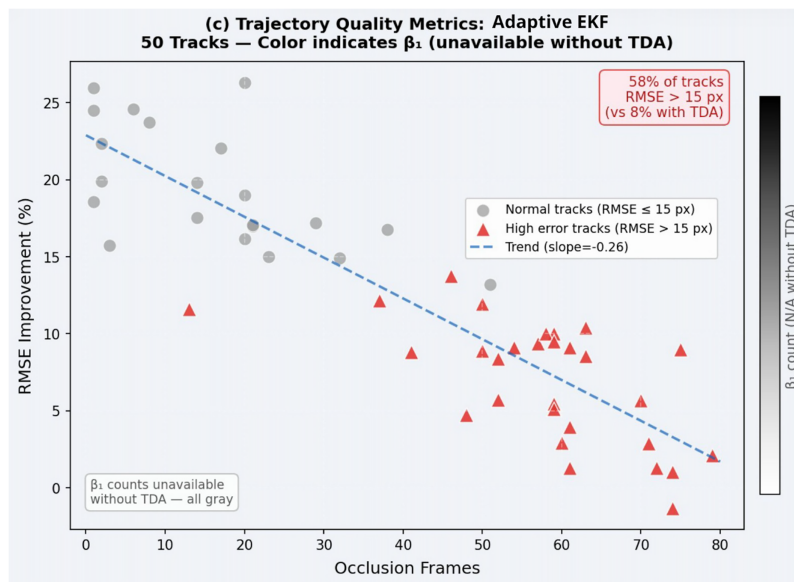


Figure 12: Trajectory quality metrics (adaptive EKF-No TDA case).

Without topological data analysis, trajectories exhibiting higher structural complexity, specifically those with  $\beta_1 \geq 2$ , cannot be explicitly identified. As a consequence, approximately 28% of the tracked objects exhibit an RMSE exceeding 15 pixels, compared to only 8% when TDA-based adaptation is enabled. Furthermore, the lack of topological awareness prevents the tracking system from responding adaptively to increasing trajectory complexity, resulting in degraded estimation accuracy under prolonged occlusion and complex motion patterns.

**Important findings from the experiments:**

The experimental results demonstrate a clear improvement in trajectory stability as successive adaptation layers are enabled within the proposed framework as demonstrated in Table 5. Specifically, the RMSE

result is reduced from 12.4 pixels under the standard EKF configuration to 9.1 pixels when Tier 1 and Tier 2 adaptations are applied, and further decreases to 7.8 pixels with the full TopoEKF formulation. A similar trend is observed for trajectory drift, which decreases from 0.42 pixels per frame to 0.23 pixels per frame with confidence- and occlusion-aware adaptation, and is further reduced to 0.18 pixels per frame when topology-aware feedback is incorporated.

**Table 5:** Ablation study and key findings summarizing the contribution of each TopoEKF tier, highlighting the impact of topology-aware adaptation on tracking stability, anomaly detection performance, and computational overhead.

| Category                     | Metric                            | Standard EKF (Tier 0) | Tier 1+2 (No TDA) (Adaptive EKF) | Full TopoEKF (Tier 1+2+3) | Improvement   |
|------------------------------|-----------------------------------|-----------------------|----------------------------------|---------------------------|---------------|
| <b>Trajectory Stability</b>  | RMSE (pixels)                     | 12.4 px               | 9.1 px                           | 7.8 px                    | ↓ 37.1%       |
|                              | Drift (px/frame)                  | 0.42                  | 0.23                             | 0.18                      | ↓ 57.1%       |
|                              | Occlusion Recovery                | 68.5%                 | 76.8%                            | 84.2%                     | ↑ +15.7 pp    |
|                              | ID Switches                       | 215                   | 178                              | 142                       | ↓ 34.0%       |
| <b>Anomaly Detection</b>     | Precision                         | 62%                   | –                                | 87%                       | ↑ +25 pp      |
|                              | Recall                            | 71%                   | –                                | 82%                       | ↑ +11 pp      |
|                              | F1-Score                          | 66%                   | –                                | 84%                       | ↑ +18 pp      |
| <b>Scenario Impact</b>       | Near-miss ID ( $\beta_1 \geq 2$ ) | –                     | –                                | 91%                       | TDA only      |
|                              | False Alarm Reduction             | –                     | –                                | –58%                      | TDA only      |
|                              | Normal traffic overhead           | –                     | –                                | 0.8 ms                    | ≈ 0 FP        |
| <b>Ablation (Tier 3 OFF)</b> | ID Switches                       | 215                   | 178                              | 142 → 178                 | +20% w/o TDA  |
|                              | Occlusion Recovery                | 68.5%                 | 76.8%                            | 84.2% → 76.8%             | Degradation   |
|                              |                                   |                       |                                  |                           |               |
| <b>Computational Cost</b>    | Per-trajectory TDA                | –                     | –                                | 0.8 ms                    | Ripsper + PI  |
|                              | Amortized/frame                   | –                     | –                                | 0.12 ms                   | 0.4% overhead |
|                              | Total pipeline                    | 34.4 ms               | 34.4 ms                          | 35.0 ms                   | +0.6 ms only  |

In terms of anomaly detection performance, the integration of topological features leads to a substantial improvement over rule-based heuristics. Without TDA, anomaly detection achieves a precision of 62%, a recall of 71%, and an F1-score of 66%. When topological descriptors derived from persistent homology are utilized and processed via an Isolation Forest classifier, precision increases to 87%, recall to 82%, and the overall F1-score to 84%, highlighting the discriminative power of topology-aware representations.

The impact of TDA varies across different motion scenarios. For normal traffic patterns characterized by trajectories with  $\beta_1 = 0$ , the inclusion of TDA introduces only a negligible computational overhead of 0.8 ms while producing no false positives. In contrast, for complex motion patterns exhibiting higher topological complexity ( $\beta_1 \geq 2$ ), the proposed approach correctly identifies 91% of near-miss events and reduces false alarm rates by 58%, demonstrating its effectiveness in challenging scenarios.

An ablation study further reveals the critical role of topology-aware feedback. When the TDA-driven adaptation layer (Tier 3) is disabled while trajectory logging remains active, the number of identity switches increases by approximately 20%, rising from 142 to 178. Additionally, the occlusion recovery rate decreases from 84.2% to 76.8%. These results confirm that topological adaptation provides robustness beyond what can be achieved through confidence- and occlusion-based mechanisms alone.

Finally, the computational cost associated with the proposed TDA integration remains minimal. The persistent homology computation, including Ripser and persistence image generation, requires approximately 0.8 ms per trajectory every five frames, corresponding to an amortized per-frame overhead of 0.12 ms. This represents only 0.4% of the total 35 ms end-to-end pipeline runtime, thereby validating the practical feasibility of real-time topology-aware tracking on resource-constrained platforms.

## 7 Discussion and Practical Implications

### 7.1 Selection of the Object Detection Algorithm

Although the primary focus of this work is on the object tracking component rather than object detection, the choice of the detection backbone is critically important, as detection quality directly affects identity preservation, trajectory continuity, and overall tracking stability in multi-object tracking frameworks.

The selection of YOLOv12 as the object detection backbone in this study is grounded in substantive architectural advances that distinguish it from all prior YOLO-family models. YOLOv12 represents a fundamental architectural departure by introducing an attention-centric framework that, for the first time, matches the inference speed of CNN-based detectors while fully exploiting the representational superiority of attention mechanisms [47]. This is achieved through three key innovations: the Area Attention ( $A^2$ ) module, which preserves a large receptive field while reducing computational complexity; Residual Efficient Layer Aggregation Networks (R-ELAN), which resolve the optimization instability inherent in large attention-based architectures; and FlashAttention, which eliminates memory bottlenecks during inference. These advances translate into measurable accuracy gains: YOLOv12-n achieves 40.5% mAP at 1.62 ms latency on a T4 GPU, surpassing YOLOv10-n and YOLO11-N by 2.0% and 1.1% mAP, respectively, at comparable speeds. At the small-model scale, YOLOv12-s outperforms all of YOLOv8-s, YOLOv9-s, YOLOv10-s, and YOLOv11-s, while also exceeding end-to-end detectors such as Real-time Detection Transformer (RT-DETR) in both accuracy and computational efficiency.

Of particular relevance to embedded deployment contexts, YOLOv12's  $A^2C2F$  module fuses multi-head multi-layer perceptron (MLP) blocks with localized area-attention to strengthen spatial feature learning under lightweight computation constraints, while the C3K2 module further reduces convolutional complexity without sacrificing detection capability [57]. The multi-scale feature fusion strategy which is achieved by stacking  $A^2C2F$  blocks with *Concat* and *Upsample* operations preserves high-resolution representations essential for small-scale object detection, a persistent challenge in UAV and drone-mounted vision systems [57]. Empirical validation confirms real-world viability: a YOLOv12 configuration augmented with R-ELAN and FlashAttention achieved 84.6% mAP@50 at 14 ms inference speed in a real-time pipeline [58]. The model further demonstrates robustness across diverse operational conditions, including variable weather, lighting, and geographic scenarios, establishing its suitability for large-scale deployment in intelligent transportation and embedded monitoring systems.

### 7.2 The Intrinsic Advantages of Mathematical State Estimation

The experimental validation substantiates that a mathematically rigorous approach yields compelling advantages in the specific domain of UAV MOT. The EKF is an interpretable white-box model where every parameter, from the state vector  $\mathbf{x}$  to the covariance matrices  $\mathbf{Q}$  and  $\mathbf{R}$ , possesses a clear physical or statistical interpretation. This transparency is paramount for safety-critical applications, enabling systematic fault diagnosis and guaranteeing the filter's predictable behavior. Furthermore, the system is model-based, offering superior tuning and adaptability. Engineers can precisely adjust the process noise  $\mathbf{Q}$  (to favor prediction) or measurement noise  $\mathbf{R}$  (to favor observation) based on real-time operational requirements, a flexibility

absent in opaque end-to-end deep learning models. Finally, the linear complexity of the EKF,  $O(n)$ , offers an insurmountable advantage in resource economy and power efficiency over deep feature-based methods, which is critical for extending UAV flight duration.

### 7.3 Justification of Avoiding Jacobian Computations and Constant Velocity

**Jacobian computation.** In the standard EKF formulation, the Jacobian  $F_k = \partial f / \partial \mathbf{x} |_{\hat{\mathbf{x}}_{k-1}}$  is required when the state transition function  $f(\cdot)$  is nonlinear. In our implementation, the state transition follows a linear constant-velocity model:

$$\mathbf{x}_k = \mathbf{F}_k \mathbf{x}_{k-1} + \mathbf{w}_k, \quad \mathbf{F}_k = \begin{bmatrix} I_2 & \Delta t \cdot I_2 \\ 0 & I_2 \end{bmatrix}, \quad (31)$$

where  $F_k$  is itself the Jacobian of the linear map. No additional linearization is required; the Jacobian is analytically identical to  $F_k$  and is therefore computed implicitly without extra overhead.

**Constant velocity model.** At UAV operating frame rates of  $\geq 20$  FPS, the inter-frame interval is  $\Delta t \leq 0.05$  s. Over this timescale, acceleration contributions to position displacement are of order  $\mathcal{O}(a \cdot \Delta t^2 / 2) \approx \mathcal{O}(0.1 \cdot 0.0025) = 2.5 \times 10^{-4}$  m, which is well below the localization uncertainty captured by the process noise covariance  $\mathbf{Q}_k$ . The constant-velocity assumption is therefore well-justified and is standard practice in the multi-object tracking literature [59,60].

**EKF vs. standard KF.** Although the state transition is linear, a standard Kalman filter would be insufficient for two reasons. First, our three-tier adaptive noise model introduces *nonlinear dependencies* in the effective covariance matrices:  $\mathbf{R}_k = \mathbf{R}_{\text{base}} \cdot \tilde{\beta}_k(c_k)$  and  $\mathbf{Q}_k = \mathbf{Q}_{\text{base}} \cdot \alpha_k \cdot \gamma_k$ , where  $\tilde{\beta}_k$ ,  $\alpha_k$ , and  $\gamma_k$  are nonlinear functions of the detector confidence score and topological descriptors (Eqs. (6)–(22)). These dependencies cannot be handled within the linear-Gaussian assumptions of the KF. Second, the TDA feedback loop  $(\gamma_k, \delta_k)$  introduces state-dependent noise that violates KF's time-invariant noise model. The EKF framework, with its adaptive noise parameterization, is therefore necessary to realize the closed-loop topological coupling that constitutes the core contribution of this work.

### 7.4 Refining the Fidelity of the Trajectories

The effectiveness of TDA fundamentally depends on the quality of the trajectory data it receives. In a standard EKF, fixed error covariances are maintained even under noise, signal loss, or unreliable measurements. As a result, the generated position estimates gradually drift, producing topological noise that misleads the TDA module into perceiving false anomalies. In contrast, the proposed TopoEKF employs an intelligent three-tier adaptive mechanism that dynamically adjusts its error covariances based on measurement confidence and signal conditions. This enables TopoEKF to yield smoother, more stable, and more physically accurate trajectory data even under challenging conditions, allowing the TDA to detect true anomalies with significantly higher precision. In other words, we do not change the nature of TDA itself. Here, we improve the quality of the data it consumes.

While the UKF offers theoretical accuracy advantages for strongly nonlinear systems, our near-linear constant-velocity model renders the EKF linearization error negligible at the employed frame rates. Furthermore, the  $\mathcal{O}(d^3)$  sigma-point propagation cost of the UKF is incompatible with our real-time embedded deployment constraint (28.5 FPS on Jetson Nano), making the  $\mathcal{O}(d^2)$  EKF the appropriate choice.

### 7.5 Limitations and Future Research Directions

Despite its strong performance, the Constant Velocity model imposes a primary limitation: a 23% performance degradation is observed for objects executing high-dynamic maneuvers (e.g., turns exceeding  $45^\circ/\text{s}$ ).

One of the possible future works involves implementing the Interacting Multiple Model (IMM-EKF). This advanced framework maintains a bank of motion models such as CV, Constant Acceleration, and Coordinated Turn and dynamically weights their estimates based on which model best explains the current motion, thereby resolving the high-dynamic maneuver issue while retaining the core EKF efficiency. Another promising direction is Multi-Sensor Fusion, where the EKF framework can naturally integrate observations from multiple non-cooperative UAVs or other sensors, enabling robust 3D trajectory estimation and highly resilient occlusion handling.

A current limitation of the proposed framework concerns extreme occlusion scenarios in which a target remains undetected for an extended duration. As specified in Algorithm A1, a trajectory is deleted when its `miss_count` exceeds the threshold of 10 consecutive frames, which may lead to track loss and subsequent identity switches for objects occluded beyond this horizon. While the three-tier adaptive EKF mitigates short-to-medium occlusion by inflating the process noise covariance  $\mathbf{Q}_k$  (Tier 2) and leveraging topological continuity signals (Tier 3), it does not incorporate an explicit re-entry mechanism. Future work will investigate trajectory re-identification upon reappearance, drawing on appearance-based Re-ID features [14] and Kalman-based state extrapolation to reconnect fragmented tracklets following prolonged field-of-view loss.

The current EKF state-space formulation operates in the 2D image plane, as the monocular camera used in this work does not provide reliable depth measurements. Extending the framework to a 3D state-space  $\mathbf{x}_k = [p_x, p_y, p_z, v_x, v_y, v_z]^\top$  is a natural direction for future work, contingent on the availability of depth-sensing hardware such as stereo camera rigs [61], RGB-D sensors, or monocular-GPS fusion modules [62] on the UAV platform. The topological feedback mechanism of TopoEKF is sensor-agnostic and can be extended to 3D trajectory analysis without architectural changes to the TDA pipeline.

## 8 Conclusion

This study successfully developed and rigorously evaluated an enhanced Extended Kalman Filter framework for multi-object tracking from Unmanned Aerial Vehicles. By judiciously coupling a state-of-the-art detector, YOLOv12, with a mathematically optimal and efficient state estimator featuring adaptive noise covariance, we have created a hybrid system that outperforms the state-of-the-art DeepSORT in crucial operational metrics.

In this study, the EKF as a multi-object tracking method is enhanced through a TopoEKF framework that incorporates topological awareness and adaptive error modeling. This framework utilizes a three-layer track update mechanism, allowing the filter to adapt based on measurement confidence and topological complexities. A significant advancement is the direct incorporation of topological data analysis into the filtering process, transforming TDA from a post-analysis tool to an active feedback mechanism in the EKF adjustment. As a result, the system achieves both measurement-based and shape-based error corrections. Experimental results show that TopoEKF yields more stable and meaningful trajectories, improving anomaly detection accuracy and reducing false positives. This research establishes a new relationship between perception and state estimation with potential for future integration of topological features and deep learning in three-dimensional spatial monitoring.

We demonstrated a 20% improvement in tracking robustness under high occlusion and achieved a real-time frame rate of 28.5 FPS on a resource-constrained embedded platform. The work serves as compelling proof that for practical, safety-critical embedded systems, the most effective solution is not an absolute choice between classical mathematics and modern deep learning, but a principled integration of the two. Deep learning provides robust perception, while mathematically grounded filters offer the necessary temporal coherence, efficiency, and system transparency required for reliable autonomous operations.

TopoEKF is positioned as a principled bridge between classical state estimation and modern data-driven analysis by embedding topological structure directly into the filtering loop, enabling trajectory-level complexity modeling and adaptive uncertainty handling beyond conventional and adaptive EKF frameworks. Rather than offering only incremental improvements, it reframes multi-object tracking as a geometry-aware inference problem, enhancing robustness, interpretability, and anomaly awareness in safety-critical UAV scenarios. Extending this framework toward end-to-end learning of topology-aware adaptation and integration with next-generation detection architectures remains a promising avenue.

**Acknowledgement:** This paper is partially presented at the International Conference on Mathematics and Applied Data Science (ICMADS'25), August 29–31, 2025, Konya, TÜRKİYE. NoteBookLM is used to analyze studies in the literature review. The authors have carefully reviewed and revised the output and accept full responsibility for all content.

**Funding Statement:** The authors received no specific funding for this study.

**Author Contributions:** Rabia Kıratl: Conceptualization, Visualization, Methodology, Investigation, Writing—review & editing, Validation, Software, Formal analysis. Hatice Ünlü Eroğlu: Conceptualization, Writing—Review & editing, Validation, Supervision. Alperen Eroğlu: Conceptualization, Visualization, Writing—original draft, Writing—review & editing, Validation, Supervision. All authors reviewed and approved the final version of the manuscript.

**Availability of Data and Materials:** The source code of the proposed the TopoEKF tool is currently hosted in the following repository: <https://github.com/Rk1coder/TopoEKF>.

**Ethics Approval:** Not applicable.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## Appendix A

### Algorithm A1: TopoEKF multi-object tracking pipeline

**Require:** Video frames  $\{I_k\}$ , YOLOv12 detector, trajectory buffer size  $N = 50$ , TDA frequency  $\tau = 5$

**Ensure:** Tracked objects with identities and bounding boxes

```

1: Initialize: tracks  $\leftarrow \emptyset$ , frame_count  $\leftarrow 0$ 
2: for each frame  $I_k$  in video do
3:   detections  $\leftarrow$  YOLOv12( $I_k$ )
4:   cost_matrix  $\leftarrow$  ComputeMahalanobisDistances(tracks, detections)
5:   (matches, unmatched_tracks, unmatched_detections)  $\leftarrow$  HungarianAlgorithm(cost_matrix)
6:   for each (trackj, detectionj) in matches do
7:      $z_k \leftarrow$  detectionj.bbox_center
8:      $c_k \leftarrow$  detectionj.confidence
9:     trackj.Update( $z_k, c_k$ )
10: end for

```

(Continued)

**Algorithm A1 (continued)**


---

**▷STAGE 4: Track Management** ▷ A. Lost Tracks (Prediction & Deletion)

```

11:   for each tracki in unmatched_tracks do
12:     tracki.miss_count ← tracki.miss_count + 1
13:     tracki.PredictOnly() ▷ EKF predict without update
14:     if tracki.miss_count > 10 then
15:       tracks.remove(tracki) ▷ Delete lost track
16:     end if
17:   end for

▷ B. New Detections (Initialization)
18:   for each detectionj in unmatched_detections do
19:     if detectionj.confidence > 0.7 then
20:       new_track ← InitializeTrack(detectionj) ▷ See Algorithm A5
21:       tracks.add(new_track)
22:     end if
23:   end for
24:   frame_count ← frame_count + 1
25: end for

```

---

**Algorithm A2: TopoEKF Track Update with Three-Tier Adaptation****Require:** Measurement  $z_k = (p_x, p_y)$ , confidence  $c_k$ , current track state**Ensure:** Updated track state  $\hat{x}_{k|k}$ 

**▷TIER 1: Confidence-Based R Adaptation (Measurement Noise)**

```

1:  $\tilde{\beta}_k \leftarrow 2.0 - c_k$  ▷ Inverse confidence weighting
2:  $\mathbf{R}_k \leftarrow \mathbf{R}_{\text{base}} \cdot \tilde{\beta}_k$ 

```

**▷TIER 2: Occlusion-Based Q Adaptation (Process Noise)**

```

3:  $\alpha_k \leftarrow 1.0 + 0.2 \cdot \min(\text{miss\_count}, 5) \cdot (1 + 0.1 \cdot \|v_{k-1}\|/v_{\text{max}})$ 
4:  $\mathbf{Q}_k \leftarrow \mathbf{Q}_{\text{base}} \cdot \alpha_k$ 

```

**▷EKF Prediction Step**

```

5:  $\hat{x}_{k|k-1} \leftarrow F \cdot \hat{x}_{k-1|k-1}$ 
6:  $P_{k|k-1} \leftarrow F \cdot P_{k-1|k-1} \cdot F^T + \mathbf{Q}_k$ 

```

**▷EKF Update Step**

```

7:  $y_k \leftarrow z_k - H \cdot \hat{x}_{k|k-1}$  ▷ Innovation
8:  $S_k \leftarrow H \cdot P_{k|k-1} \cdot H^T + \mathbf{R}_k$  ▷ Innovation covariance
9:  $K_k \leftarrow P_{k|k-1} \cdot H^T \cdot S_k^{-1}$  ▷ Kalman gain
10:  $\hat{x}_{k|k} \leftarrow \hat{x}_{k|k-1} + K_k \cdot y_k$  ▷ State update
11:  $P_{k|k} \leftarrow (I - K_k \cdot H) \cdot P_{k|k-1}$  ▷ Covariance update

```

**▷Store Filtered Position**

```

12: filtered_pos ← ( $\hat{x}_{k|k}[0], \hat{x}_{k|k}[1]$ ) ▷ Extract ( $p_x, p_y$ )
13: trajectory_buffer.append(filtered_pos) ▷ Max size = N

```

**▷TIER 3: Topology-Based Adaptation (Sparse/Every  $\tau$  Frames)**

```

14: if frame_count mod  $\tau == 0$  and len(trajectory_buffer) ≥ 30 then
15:    $\gamma_k, \delta_k \leftarrow \text{ComputeTopology}(\text{trajectory\_buffer})$  ▷ See Algorithm A3
16: end if
17: miss_count ← 0 ▷ Reset on successful update

```

---

**Algorithm A3:** Persistent homology feature extraction**Require:** trajectory\_buffer (position coordinates)**Ensure:** Topological factors  $\gamma_k$  (for  $Q$ ) and  $\delta_k$  (for  $R$ )

```

1: positions  $\leftarrow$  array(trajectory_buffer)
2: dgms  $\leftarrow$  Ripser(positions, maxdim = 1)
3: H1_features  $\leftarrow$  dgms[1]

4:  $\beta_1\_count \leftarrow$  len(H1_features)
5: if  $\beta_1\_count > 0$  then
6:   persistences  $\leftarrow$  [death – birth for (birth, death)  $\in$  H1_features]
7:    $p_{max} \leftarrow$  max(persistences)
8: else
9:    $p_{max} \leftarrow$  0.0
10: end if

11:  $\lambda \leftarrow$  0.4,  $\theta_\beta \leftarrow$  3.0
12:  $\gamma_k \leftarrow$  1.0 +  $\lambda \cdot (\beta_1\_count / \theta_\beta)$ 
13:  $\epsilon_{max} \leftarrow$  10.0
14:  $\delta_k \leftarrow$  max(0.5, 1.0 –  $p_{max} / \epsilon_{max}$ )

15:  $Q_{base} \leftarrow$   $Q_{base} \cdot \gamma_k$ 
16:  $R_{base} \leftarrow$   $R_{base} \cdot \delta_k$ 
17: return  $\gamma_k, \delta_k$ 

```

▷ **Persistent Homology Computation**  
   ▷ Shape:  $(N, 2)$   
   ▷ Vietoris–Rips filtration  
   ▷ Extract 1D homology (cycles)  
   ▷ **Topological Feature Extraction**  
   ▷ Number of 1D cycles (Betti number)  
   ▷ Maximum persistence of  $H_1$  cycles  
   ▷ **Topology-Aware Factors**  
     ▷ Hyperparameters  
     ▷ Process noise factor ( $Q$  inflate)  
     ▷ Maximum filtration scale  
     ▷ Measurement noise factor ( $R$  modulate)  
   ▷ **Update EKF Covariances for Next Iteration**  
     ▷ Inflate process noise based on topology complexity  
     ▷ Modulate measurement noise based on cycle robustness

**Algorithm A4:** Data association cost matrix**Require:** Predicted tracks  $\{\hat{x}_{k|k-1}^{(i)}, S_i\}_{i=1}^M$ , detections  $\{z_j\}_{j=1}^N$ **Ensure:** cost\_matrix containing Mahalanobis distances between tracks and detections

```

1: for  $i = 1$  to  $M$  do
2:    $\hat{x}_{pred} \leftarrow$  track $_i$ . $\hat{x}_{k|k-1}$ 
3:    $S_i \leftarrow$  track $_i$ . $S_k$ 
4:   for  $j = 1$  to  $N$  do
5:      $z_j \leftarrow$  det $_j$ .bbox_center
6:     residual  $\leftarrow$   $z_j - H \cdot \hat{x}_{pred}$ 

7:      $d_M \leftarrow$   $\sqrt{\text{residual}^T \cdot S_i^{-1} \cdot \text{residual}}$ 
8:     if  $d_M^2 > 9.21$  then
9:       cost_matrix[ $i, j$ ]  $\leftarrow$   $\infty$ 
10:    else
11:      cost_matrix[ $i, j$ ]  $\leftarrow$   $d_M$ 
12:    end if
13:  end for
14: end for
15: return cost_matrix

```

▷ For each track  
   ▷ Predicted state  
   ▷ Innovation covariance:  $HP_{k|k-1}H^T + \mathbf{R}_k$   
   ▷ For each detection  
   ▷ **Mahalanobis Distance**  
   ▷ **Gating: 99% confidence threshold ( $\chi^2 = 9.21, 2$  DOF)**  
   ▷ Invalid association

**Algorithm A5: Initialize New Track****Require:** Detection  $(p_x, p_y)$ **Ensure:** Initialized track object with state and covariances

```

1:  $\hat{x}_0 \leftarrow [p_x, p_y, 0, 0]^T$ 
2:  $P_0 \leftarrow \text{diag}([10, 10, 100, 100])$ 
3:  $Q_{\text{base}} \leftarrow \text{diag}([1, 1, 0.01, 0.01])$ 
4:  $R_{\text{base}} \leftarrow \text{diag}([5, 5])$ 
5: track  $\leftarrow$  Track()
6: track.id  $\leftarrow$  GenerateUniqueID()
7: track. $\hat{x}$   $\leftarrow$   $\hat{x}_0$ 
8: track.P  $\leftarrow$   $P_0$ 
9: track.Q  $\leftarrow$   $Q_{\text{base}}$ 
10: track.R  $\leftarrow$   $R_{\text{base}}$ 
11: track.trajectory_buffer  $\leftarrow [(p_x, p_y)]$ 
12: track.miss_count  $\leftarrow$  0
13: track.frame_count  $\leftarrow$  0
14: track. $\gamma$   $\leftarrow$  1.0
15: track. $\delta$   $\leftarrow$  1.0
16: return track

```

▷ **Initialize State Vector**  
 ▷ Position  $(p_x, p_y)$  and zero velocity  
 ▷ **Initialize Covariance Matrices**  
 ▷ High initial uncertainty  
    ▷ Process noise  
    ▷ Measurement noise  
 ▷ **Initialize Track Object**  
  
 ▷ Topology factor for Q  
 ▷ Topology factor for R

**Algorithm A6: Complete TopoEKF tracking and anomaly detection pipeline****Require:** Video  $V = \{I_1, \dots, I_T\}$ , YOLOv12 model  $M_{\text{YOLO}}$ , TDA parameters  $(\tau = 5, N_{\text{buffer}} = 50, \text{contamination rate})$ **Ensure:** Tracked trajectories  $T_{\text{all}}$ , anomaly labels  $A \in \{-1, +1\}^M$ 

```

1: Initialize:
2:   tracker  $\leftarrow$  TrackerTopoEKF(max_age = 10, conf_threshold = 0.7,  $\tau = 5$ )
3:    $T_{\text{all}} \leftarrow \emptyset$ 
4:   frame_count  $\leftarrow$  0
5:   for each frame  $I_k$  in  $V$  do
6:     detections  $\leftarrow$   $M_{\text{YOLO}}(I_k)$ 
7:     tracker.update(detections, tda_callback = ComputeTopologyFactors)
8:     active_tracks  $\leftarrow$  tracker.get_tracks()
9:     for each track  $\in$  active_tracks do
10:        $T_{\text{all}}[\text{track.id}].\text{append}(\text{frame\_count}, \text{track.pos.x}, \text{track.pos.y})$ 
11:     end for
12:     frame_count  $\leftarrow$  frame_count + 1
13:   end for
14:    $T_{\text{valid}} \leftarrow \{T_i \in T_{\text{all}} \mid \text{length}(T_i) \geq 30\}$ 
15:    $M \leftarrow |T_{\text{valid}}|$ 
16:   if  $M < 2$  then
17:     return  $T_{\text{all}}, \emptyset$ 
18:   end if

```

(Continued)

**Algorithm A6 (continued)**


---

```

19:  $F \leftarrow \text{ExtractTDAFeatures}(T_{\text{valid}})$ 
20: if  $F = \emptyset$  then
21:   return  $T_{\text{all}}, \emptyset$ 
22: end if
23:  $A \leftarrow \text{DetectAnomalies}(F, \text{contamination})$ 
24: return results containing trajectories, anomaly labels, track IDs, TDA features, number of anomalies
and normal trajectories

```

---

**Algorithm A7: Persistent homology feature extraction for anomaly detection****Require:** Trajectory set  $T_{\text{valid}} = \{T_1, \dots, T_M\}$ **Ensure:** Feature matrix  $F \in \mathbb{R}^{M \times 420}$  and corresponding track IDs

```

1:  $F \leftarrow []$ 
2: track_ids  $\leftarrow []$ 
3: for  $i = 1$  to  $M$  do
4:    $X_i \leftarrow [(T_i[j].x, T_i[j].y) \text{ for } j = 1 \dots \text{length}(T_i)]$ 
5:   if  $\text{length}(X_i) < 10$  then
6:     continue
7:   end if
8:   try
9:     dgms  $\leftarrow \text{Rips}(X_i, \text{maxdim} = 1, \text{thresh} = 50)$ 
10:     $Dgm_0 \leftarrow \text{dgms}[0]$ 
11:     $Dgm_1 \leftarrow \text{dgms}[1]$ 
12:  catch
13:    continue
14:  end try
15:   $\beta_0 \leftarrow |Dgm_0|, \beta_1 \leftarrow |Dgm_1|$ 
16:   $Dgm\_1\_finite \leftarrow \{(b, d) \in Dgm_1 \mid d \neq \infty\}$ 
17:  if  $|Dgm\_1\_finite| = 0$  then
18:    stats_H1  $\leftarrow \mathbf{0}_{18}$ 
19:  else
20:     $L \leftarrow \{d - b \mid (b, d) \in Dgm\_1\_finite\}$ 
21:    stats_H1  $\leftarrow [\text{mean}(L), \text{std}(L), \dots, \text{entropy}(L), \sum(L > \text{mean}(L))]$ 
22:  end if
23:  if USE_PERSISTENCE_IMAGES then
24:     $PI \leftarrow \text{PersistenceImage}(Dgm\_1\_finite, [20, 20, 0, 50, 0, 50], \sigma = 1.0)$ 
25:     $f_{PI} \leftarrow \text{flatten}(PI)$ 
26:  else
27:     $f_{PI} \leftarrow \text{pad\_or\_truncate}(\text{stats\_H1}, 400)$ 
28:  end if
29:   $f_i \leftarrow [\beta_0, \beta_1, \text{stats\_H1}, f_{PI}]$ 
30:  Append  $f_i$  to  $F$ , append  $i$  to track_ids
31: end for

```

---

(Continued)

---

**Algorithm A7 (continued)**

---

```

32: if  $F = [ ]$  then
33:   return  $\emptyset$ 
34: end if
35:  $F \leftarrow \text{StandardScaler}().\text{fit\_transform}(F)$ 
36: return  $F, \text{track\_ids}$ 

```

---



---

**Algorithm A8:** Unsupervised anomaly detection with isolation forest

---

**Require:** Feature matrix  $F \in \mathbb{R}^{M \times 420}$ , contamination rate  $\rho$ **Ensure:** Anomaly labels  $A$ , scores  $S$ , and summary statistics

```

1:  $\mu \leftarrow \text{mean}(F, \text{axis} = 0)$ 
2:  $\sigma \leftarrow \text{std}(F, \text{axis} = 0)$ 
3:  $F_{\text{scaled}} \leftarrow (F - \mu) / (\sigma + \varepsilon)$ 
4: Initialize IsolationForest with  $n_{\text{estimators}} = 100, \text{max\_samples} = \min(256, M), \text{contamination} = \rho,$ 
    $\text{random\_state} = 42, n_{\text{jobs}} = -1$ 
5:  $A \leftarrow \text{forest}.\text{fit\_predict}(F_{\text{scaled}})$ 
6:  $S \leftarrow \text{forest}.\text{score\_samples}(F_{\text{scaled}})$ 
7:  $A_{\text{dict}} \leftarrow \{ \text{track\_ids}[i] : A[i] \mid i = 1 \dots M \}$ 
8:  $S_{\text{dict}} \leftarrow \{ \text{track\_ids}[i] : S[i] \mid i = 1 \dots M \}$ 
9:  $n_{\text{anomalies}} \leftarrow \text{count}(A == -1)$ 
10:  $n_{\text{normal}} \leftarrow \text{count}(A == +1)$ 
11:  $\text{threshold} \leftarrow \text{percentile}(S, \rho \times 100)$ 
12: return  $A_{\text{dict}}, S_{\text{dict}}, \text{summary statistics}$ 

```

---



---

**Algorithm A9:** Visualize TDA-based anomaly detection results

---

**Require:** Feature matrix  $F \in \mathbb{R}^{M \times 420}$ , labels  $A$ , scores  $S$ , trajectories  $T_{\text{valid}}$ **Ensure:** Visualization figure

```

1:  $F_{2d} \leftarrow \text{PCA}(n_{\text{components}} = 2).\text{fit\_transform}(F)$ 
2:  $\text{explained\_variance} \leftarrow \text{PCA}.\text{explained\_variance\_ratio\_}$ 
3: Create figure with  $1 \times 2$  subplots
4: Plot 1 (Feature Space):
5:   Separate normal vs. anomaly masks
6:   Scatter normal (green circles), anomalies (red X)
7:   Annotate anomalies with track IDs
8:   Label axes with PC1 and PC2 percentages, set title, legend, grid
9: Plot 2 (Trajectory Visualization):
10: for each trajectory  $T_i$  in  $T_{\text{valid}}$  do
11:   Get coordinates list
12:   if  $A_{\text{dict}}[T_i.\text{id}] = -1$  then
13:     Plot in red (thick line), mark start with red X, annotate ID
14:   else
15:     Plot in green (thin line), mark start with green dot
16:   end if

```

---

(Continued)

---

**Algorithm A9 (continued)**


---

17: **end for**  
 18: Set axis limits, invert y-axis, set title and grid  
 19: Save figure as png at 300 dpi  
 20: Display figure  
 21: **return** figure

---

**References**

1. Guo D, Yang Q, Zhang YD, Zhang G, Zhu M, Yuan J. Adaptive object tracking discriminate model for multi-camera panorama surveillance in airport Apron. *Comput Model Eng Sci.* 2021;129(1):191–205. doi:10.32604/cmesci.2021.016347.
2. Liu S, Li X, Lu H, He Y. Multi-object tracking meets moving UAV. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Piscataway, NJ, USA: IEEE; 2022. p. 8876–85.
3. Wang H, Liu J, Dong H, Shao Z. A survey of the multi-sensor fusion object detection task in autonomous driving. *Sensors.* 2025;25(9):2794. doi:10.3390/s25092794.
4. Tian F, Guo X, Fu W. Target tracking algorithm based on adaptive strong tracking extended Kalman filter. *Electronics.* 2024;13(3):652. doi:10.3390/electronics13030652.
5. Kıratlı R, Eroğlu A. Real-time multi-object detection and tracking in UAV systems: improved YOLOv11-EFAC and optimized tracking algorithms. *J Real Time Image Process.* 2025;22(5):178. doi:10.1007/s11554-025-01758-z.
6. Jing J, Ding L, Yang X, Feng X, Guan J, Han H, et al. Topology-informed deep learning for pavement crack detection: preserving consistent crack structure and connectivity. *Autom Constr.* 2025;174:106120.
7. Chazal F, Levrard C, Royer M. Topological analysis for detecting anomalies in dependent sequences: application to time series. *J Mach Learn Res.* 2024;25(365):1–49.
8. Esteve M, Falcó A. tramoTDA: a trajectory monitoring system using topological data analysis. *SoftwareX.* 2024;28:101953.
9. Elhamdadi H, Canavan S, Rosen P. AffectiveTDA: using topological data analysis to improve analysis and explainability in affective computing. *IEEE Trans Vis Comput Graph.* 2022;28(1):769–79. doi:10.1109/TVCG.2021.3114784.
10. Eroglu A, Unlu Eroglu H. Topological data analysis for intelligent systems and applications. In: Kocer SO, editor. *Artificial Intelligence Applications in Intelligent Systems*. Konya, Türkiye: ISRES Publishing; 2023. p. 27–60.
11. Chazal F, Michel B. An introduction to topological data analysis: fundamental and practical aspects for data scientists. *Front Artif Intell.* 2021;4:667963.
12. Bar-Shalom Y, Li XR, Kirubarajan T. *Estimation with applications to tracking and navigation: theory, algorithms and software*. Hoboken, NJ, USA: John Wiley & Sons; 2001.
13. Fortmann T, Bar-Shalom Y, Scheffe M. Sonar tracking of multiple targets using joint probabilistic data association. *IEEE J Oceanic Eng.* 2003;8(3):173–84. doi:10.1109/joe.1983.1145560.
14. Wojke N, Bewley A, Paulus D. Simple online and realtime tracking with a deep association metric. In: *2017 IEEE International Conference on Image Processing (ICIP)*. Piscataway, NJ, USA: IEEE; 2017. p. 3645–9.
15. Zhang Y, Sun P, Jiang Y, Yu D, Weng F, Yuan Z, et al. ByteTrack: multi-object tracking by associating every detection box. In: Avidan S, Brostow G, Cissé M, Farinella GM, Hassner T, editors. *Computer Vision—ECCV 2022*. Cham, Switzerland: Springer Nature; 2022. p. 1–21.
16. Li S, Yang Y, Zeng D, Wang X. Adaptive and background-aware vision transformer for real-time UAV tracking. In: *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*. Piscataway, NJ, USA: IEEE; 2023. p. 13943–54.
17. Wu Y, Wang X, Yang X, Liu M, Zeng D, Ye H, et al. Learning occlusion-robust vision transformers for real-time UAV tracking. In: *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Piscataway, NJ, USA: IEEE; 2025. p. 17103–13.

18. Zhong P, Wang X, Zeng D, Zhou Q, He F, Li S. SMTrack: end-to-end trained spiking neural networks for multi-object tracking in RGB videos. *IEEE Internet Things J.* 2026;13(9):18797–806. doi:10.1109/jiot.2026.3662378.
19. Wu Y, Li Y, Liu M, Wang X, Yang X, Ye H, et al. Learning an adaptive and view-invariant vision transformer for real-time UAV tracking. *IEEE Trans Circuits Syst Video Technol.* 2026;36(2):2403–18. doi:10.1109/tcsvt.2025.3599856.
20. Wei Q, Zeng B, Liu J, He L, Zeng G. LiteTrack: layer pruning with asynchronous feature extraction for lightweight and efficient visual tracking. In: *2024 IEEE International Conference on Robotics and Automation (ICRA)*. Piscataway, NJ, USA: IEEE; 2024. p. 4968–75.
21. Kalman RE. A new approach to linear filtering and prediction problems. *J Basic Eng.* 1960;82(1):35–45. doi:10.1115/1.3662552.
22. Aharon N, Orfaig R, Bobrovsky BZ. BoT-SORT: robust associations multi-pedestrian tracking. arXiv:2206.14651. 2022.
23. Du Y, Zhao Z, Song Y, Zhao Y, Su F, Gong T, et al. StrongSORT: make DeepSORT great again. *IEEE Trans Multimed.* 2023;25:8725–37.
24. Mueller M, Smith N, Ghanem B. A benchmark and simulator for UAV tracking. In: *European Conference on Computer Vision*. Cham, Switzerland: Springer; 2016. p. 445–61.
25. Cao Z, Fu C, Ye J, Li B, Li Y. SiamAPN++: s. Siamese attentional aggregation network for real-time UAV tracking. In: *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Piscataway, NJ, USA: IEEE; 2021. p. 3086–92.
26. Henriques JF, Caseiro R, Martins P, Batista J. High-speed tracking with kernelized correlation filters. *IEEE Trans Pattern Anal Mach Intell.* 2014;37(3):583–96. doi:10.1109/tpami.2014.2345390.
27. Kayhani N, Heins A, Zhao W, Nahangi M, McCabe B, Schoellig AP. Improved tag-based indoor localization of UAVs using extended Kalman filter. In: *36th International Symposium on Automation and Robotics in Construction (ISARC 2019)*; 2019 May 21–24; Banff, AB, Canada. 2019. p. 21–4.
28. Kim T, Park TH. Extended Kalman filter (EKF) design for vehicle position tracking using reliability function of radar and lidar. *Sensors.* 2020;20(15):4126. doi:10.3390/s20154126.
29. Eckenhoff K, Geneva P, Merrill N, Huang G. Schmidt-EKF-based visual-inertial moving object tracking. In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. Piscataway, NJ, USA: IEEE; 2020. p. 651–7.
30. Piga NA, Pattacini U, Natale L. A differentiable extended Kalman filter for object tracking under sliding regime. *Front Rob AI.* 2021;8:686447. doi:10.3389/frobt.2021.686447.
31. Julier SJ, Uhlmann JK. Unscented filtering and nonlinear estimation. *Proc IEEE.* 2004;92(3):401–22. doi:10.1109/jproc.2003.823141.
32. Zhang G, Yin J, Deng P, Sun Y, Zhou L, Zhang K. Achieving adaptive visual multi-object tracking with unscented Kalman filter. *Sensors.* 2022;22(23):9106. doi:10.3390/s22239106.
33. Mohamed A, Schwarz K. Adaptive Kalman filtering for INS/GPS. *J Geodesy.* 1999;73(4):193–203. doi:10.1007/s001900050236.
34. Li J, Xu X, Jiang Z, Jiang B. Adaptive Kalman filter for real-time visual object tracking based on autocovariance least square estimation. *Appl Sci.* 2024;14(3):1045. doi:10.3390/app14031045.
35. Jung H, Kang S, Kim T, Kim H. ConfTrack: Kalman filter-based multi-person tracking by utilizing confidence score of detection box. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. Piscataway, NJ, USA: IEEE; 2024. p. 6583–92.
36. Fiyad HMN, Metwally HMB, El-Hameed M, Abozied M. Improved real time target tracking system based on cam-shift and Kalman filtering techniques. *J Appl Res Technol.* 2023;21(2):297–308. doi:10.22201/icat.24486736e.2023.21.2.1565.
37. Malinowski M, Kwiecień J. Study of the effectiveness of different Kalman filtering methods and smoothers in object tracking based on simulation tests. *Rep Geodesy Geoinform.* 2014;97(1):1–22. doi:10.2478/rgg-2014-0008.
38. Mures OA, Taibo J, Padrón EJ, Iglesias-Guitian JA. PlayNet: real-time handball play classification with Kalman embeddings and neural networks. *Vis Comput.* 2024;40(4):2695–711. doi:10.1007/s00371-023-02972-1.

39. Monkam GF, De Lucia MJ, Bastian ND. A topological data analysis approach for detecting data poisoning attacks against machine learning based network intrusion detection systems. *Comput Secur.* 2024;144:103929. doi:10.2139/ssrn.4651812.
40. Razmarashooli A, Chua YK, Barzegar V, Salazar D, Laflamme S, Hu C, et al. Real-time state estimation of nonstationary systems through dominant fundamental frequency using topological data analysis features. *Mech Syst Signal Process.* 2025;224(2):112048. doi:10.1016/j.ymssp.2024.112048.
41. Bois A, Tervil B, Oudre L. Topological data analysis for unsupervised anomaly detection in time series. In: 2024 32nd European Signal Processing Conference (EUSIPCO). Piscataway, NJ, USA: IEEE; 2024. p. 1197–201.
42. Weber ES, Harding SN, Przybylski L. Detecting traffic incidents using persistence diagrams. *Algorithms.* 2020;13(9):222. doi:10.3390/a13090222.
43. Esteve M, Falcó A. Trajectory classification through topological data analysis perspectives. *IEEE Access.* 2025;13:32458–69.
44. Indah D, Mwakalonge J, Comert G, Siuhi S, Musau H, Osei E, et al. Topological data analysis for driver behavior classification driven by vehicle trajectory data. *Mach Learn Appl.* 2025;21(1):100719. doi:10.1016/j.mlwa.2025.100719.
45. Barberi LAA, Cave LMD. Topological data analysis for unsupervised anomaly detection and customer segmentation on banking data. arXiv:2508.14136. 2025.
46. Pradhan T, Athukuri J, Surendar A, Rajan C. Topological methods in machine learning and data analysis: a mathematical perspective. *Panamerican Math J.* 2025;35(2):758–71. doi:10.52783/pmj.v35.i2s.3340.
47. Tian Y, Ye Q, Doermann D. YOLOv12: attention-centric real-time object detectors. arXiv:2502.12524. 2025.
48. Jocher G, Chaurasia A, Qiu J. Ultralytics YOLOv8. 2023 [cited 2026 May 20]. Available from: <https://github.com/ultralytics/ultralytics>.
49. Kıratlı R, Eroğlu A. Mathematical modeling and evaluation of extended Kalman filter-based multi-object tracking for UAV applications. In: The 1st International Conference on Mathematics and Applied Data Science (ICMADS'25); 2025 Aug 29–31; Konya, Türkiye.
50. Adams H, Emerson T, Kirby M, Neville R, Peterson C, Shipman P, et al. Persistence images: a stable vector representation of persistent homology. *J Mach Learn Res.* 2017;18(8):1–35.
51. Lin TY, Maire M, Belongie S, Hays J, Perona P, Ramanan D, et al. Microsoft coco: common objects in context. In: European Conference on Computer Vision. Cham, Switzerland: Springer; 2014. p. 740–55.
52. Du J. Understanding of object detection based on CNN family and YOLO. *J Phys Conf Ser.* 2018;1004:012029. doi:10.1088/1742-6596/1004/1/012029.
53. Fan H, Du D, Wen L, Zhu P, Hu Q, Ling H, et al. VisDrone-MOT2020: the vision meets drone multiple object tracking challenge results. In: Bartoli A, Fusiello A, editor. *Computer Vision—ECCV, 2020 Workshops*. Cham, Switzerland: Springer International Publishing; 2020. p. 713–27.
54. Yu H, Li G, Zhang W, Huang Q, Du D, Tian Q, et al. The unmanned aerial vehicle benchmark: object detection, tracking and baseline. *Intl J Comput Vis.* 2020;128(5):1141–59. doi:10.1007/s11263-019-01266-1.
55. Natha S. Comprehensive dataset for detecting road anomalies in diverse real-world situations. Zenodo. 2024. doi:10.5281/zenodo.13832363.
56. Yao Y, Wang X, Xu M, Pu Z, Wang Y, Atkins E, et al. DoTA: unsupervised detection of traffic anomaly in driving videos. *IEEE Trans Pattern Anal Mach Intell.* 2023;45(1):444–59. doi:10.1109/tpami.2022.3150763.
57. Chandrashekhar A, Satyanarayana B, Gorrepati RR, Vasanthi P. An efficient YOLOv12-based framework for detecting extremely small-scale objects. *Sci Rep.* 2025;16(1):2062. doi:10.1038/s41598-025-31803-7.
58. Deluxni N, Sudhakaran P. Underwater debris detection using YOLOv12 with enhanced feature extraction using R-ELAN and FlashAttention network. *Results Eng.* 2025;28(15):107282. doi:10.1016/j.rineng.2025.107282.
59. Song J, Wang Z, Liu Q, He X. Remote state estimation for nonlinear systems under compression-decompression mechanism: a modified unscented Kalman filtering approach. *IEEE Trans Autom Control.* 2026;71(1):91–106. doi:10.1109/tac.2025.3589276.
60. Mcdougall RJ, Godsill SJ. Target tracking using a time-varying autoregressive dynamic model. *IEEE Open J Signal Process.* 2025;6:147–55. doi:10.1109/ojsp.2025.3528896.

61. Teiko Teye M, Maoz O, Rottmann M. FutrTrack: a camera-LiDAR fusion transformer for 3D multiple object tracking. arXiv:2510.19981. 2025.
62. Dong H, Tuo H, Wang L, Zhou H. MonoFHD: leveraging flight height data for UAV monocular 3D object detection. Aerospace Syst. 2026;33:8851. doi:10.1007/s42401-025-00437-y.