

# BSense: Wide-Area UAV Detection and Tracking with 5G-Advanced Base Station



Chengzhen Meng<sup>†\*</sup>, Chenming He<sup>†\*</sup>, Yidong Jiang<sup>†</sup>, Xiaoran Fan<sup>‡</sup>, Dequan Wang<sup>†</sup>,  
Lingyu Wang<sup>†</sup>, Jianmin Ji<sup>†</sup>, Yanyong Zhang<sup>†◊</sup>

<sup>†</sup>University of Science and Technology of China, <sup>‡</sup>Independent Researcher,

<sup>◊</sup>Institute of Artificial Intelligence, Hefei Comprehensive National Science Center

## Abstract

The increasing prevalence of UAVs in daily life has made their consistent monitoring a critical necessity. However, existing UAV perception systems typically rely on cameras or radars, whose limited sensing range and deployment constraints hinder large-scale adoption. In response, we develop BSense, the first system that achieves accurate UAV detection and tracking by leveraging 5G-Advanced base stations in the real world. The key challenge here is discriminating false detections that closely resemble UAVs, with their number overwhelming true targets by more than 100 to 1. To overcome this, we propose a novel layered framework that operates across point, object, and trajectory levels. At the point level, we adaptively model the distribution of noise signal features and filter incoming points with high similarity to the noise. At the object level, we define confidence metrics to distinguish UAVs from fake objects, derived from spatial and velocity consistencies across frames and augmented with frequency-recency heuristics. At the trajectory level, we propose a Transformer-based network that captures multi-frame motion patterns to eliminate fake trajectories.

We evaluated BSense using a COTS 5G-Advanced base station deployed in an urban environment. The UAV was instructed to fly along 25 distinct trajectories across 51 cases over 7 days, yielding 135 minutes of data with more than 12,000 frames. On this dataset, our system reduces the number of false detections from an average of 168.05 per frame to 0.04, achieving an average F1 score of 95.77% and a mean localization error of 4.9 m at ranges up to 1,000 m.

## Keywords

5G-A Base Station Sensing, UAV Detection and Tracking

## 1 INTRODUCTION

Unmanned aerial vehicles (UAVs) are becoming increasingly prevalent, with applications ranging from aerial transportation to infrastructure inspection [1, 53, 55]. They are also emerging as integral components of smart city infrastructure, playing vital roles in traffic monitoring, emergency response, and environmental surveillance [3, 30, 37]. Consequently,

accurate wide-area UAV detection and tracking are critical to these applications, ensuring safe flight navigation and reliable airspace intrusion detection.

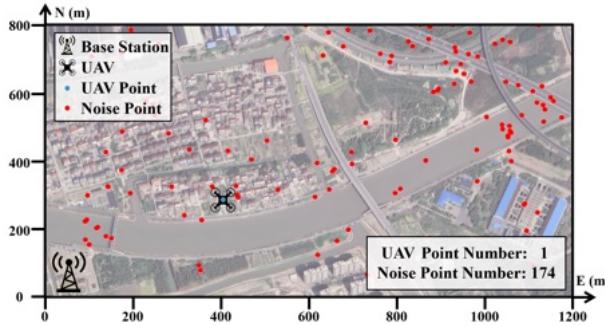
In recent years, some researchers have explored the use of cameras and radars for UAV perception [12, 20, 36, 41, 46, 51]. However, cameras are limited by their short sensing range and are highly sensitive to lighting conditions [34], whereas radars are expensive and impractical for large-scale deployment [33]. In response to these limitations, this study explores leveraging the base station to develop a low-cost UAV detection and tracking system with wide-area coverage.

Recently, 5G has been advancing toward 5G-Advanced (5G-A), which incorporates Integrated Sensing and Communication (ISAC) technology to enhance base station functionality [2, 18, 25, 29, 43]. In addition to providing reliable communication services, 5G-A base stations also offer radar-like sensing abilities [23, 44]. This enables low-cost UAV sensing by leveraging existing base station infrastructure without requiring additional sensing equipment [48].

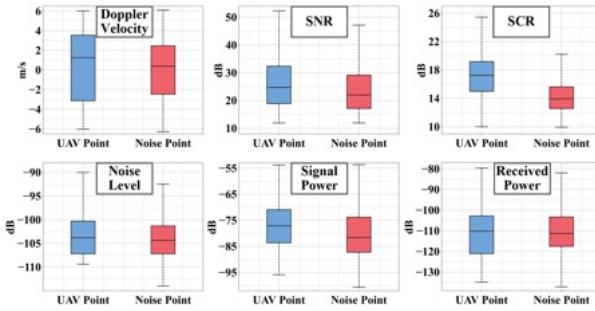
There have been previous studies [8, 22, 28, 49, 54] that explore UAV perception using 5G-A base stations. However, their evaluations are limited to theoretical analysis or idealized simulations, which fail to reflect the complexity of real-world physical environments. To move beyond these controlled settings, we investigate the feasibility of wide-area UAV detection and tracking using a commercial off-the-shelf (COTS) 5G-A base station deployed in an urban environment. To the best of our knowledge, this is the first work to reveal and address the key challenges that emerge in such real-world deployment: *numerous and persistent false detections which closely resemble true UAV targets*.

To substantiate this claim, Figure 1 shows a single frame from our dataset. The data were collected on a Huawei COTS 5G-A base station deployed in Shanghai, which generates per-frame point clouds through its internal signal-processing pipeline. For clearer visualization, we project the point cloud onto the corresponding satellite map. Notably, unlike prior systems that focus on UAVs within short distances (typically under 150 m) [28, 49], the COTS base station we used provides a sensing range of up to 1,000 m. However, at long ranges, UAV reflections become extremely weak and are difficult to distinguish from environmental noise.

\*Equal contribution.



**Figure 1: A single frame of real-world data.** The number of noise points exceeds UAV points by a ratio of 174:1.

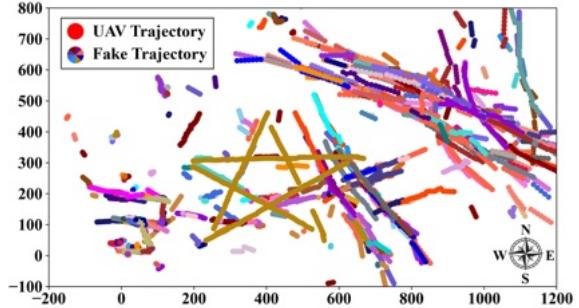


**Figure 2: The value ranges of point features for UAV and noise exhibit substantial overlap.**

As a result, even though signal-level noise suppression is applied before point cloud generation, a substantial number of noise points still persist in the output. In this frame, only a single point corresponds to the UAV, while 174 points are noise—resulting in a two-order-of-magnitude imbalance. Through theoretical analysis and experimental observations, we attribute these noise points to three sources: (1) *background clutter* from static structures (e.g., buildings and trees). However, small physical vibrations and Doppler leakage can make their returns exhibit non-zero velocities and thus difficult to filter [24]. (2) *Multi-path ghosts* caused by multiple reflections between moving objects (e.g., vehicles) and reflective surfaces (e.g., building facades). (3) *Sidelobe-induced ghosts* caused by sidelobe returns incorrectly interpreted as main-lobe signals. Therefore, our primary task is to accurately distinguish UAVs from these abundant false detections, which has not been investigated in prior works.

However, this isn't easy because the false detections are persistent and significantly similar to UAV targets:

- *Feature Overlap Between UAV and Noise Points.* As illustrated in Figure 2, a statistical analysis of real-world point cloud data, including Doppler velocity, SNR, SCR, and other signal metrics, shows substantial overlap between the feature distributions of UAV points and noise points across all dimensions. This strong similarity eliminates any clear decision boundary, rendering



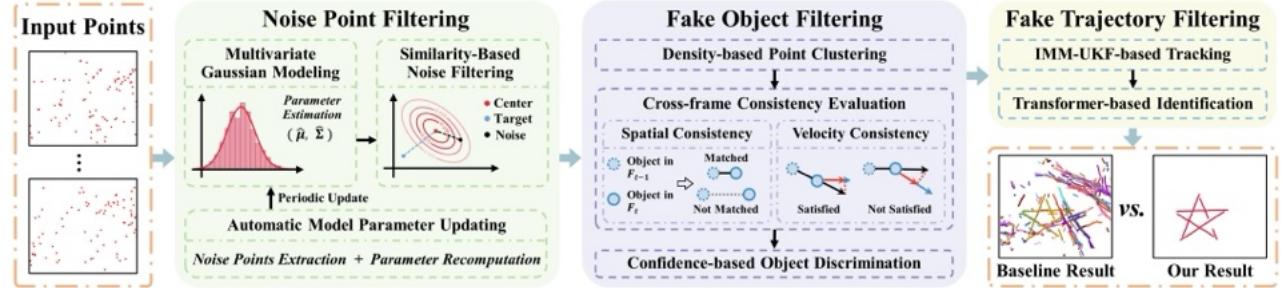
**Figure 3: Persistent noise points give rise to numerous long-term fake trajectories.**

traditional threshold-based filtering methods ineffective and leaving noise points particularly difficult to distinguish from true UAV points.

- *Long-term Fake Trajectories from Persistent Noise Points.* Beyond the similarity in point-level features, many noise points persist over long durations, forming stable trajectories that closely mimic those of UAVs. As shown in Figure 3, we cluster and track the detected point clouds and visualize the trajectories in different colors. The true UAV trajectory appears as a dark gold star, yet numerous fake trajectories also arise, some extending for hundreds of meters. Critically, these long-term fake trajectories often occur in plausible UAV locations, making accurate discrimination particularly challenging. Our experiments in Appendix A show that even state-of-the-art learning-based methods cannot fully eliminate such fake trajectories.

This paper presents BSense, a wide-area UAV perception system with the 5G-A base station. To tackle the challenge of numerous and persistent noise points, we propose a novel layered framework that filters noise at the point, object, and trajectory levels while preserving true UAV targets. At the point level, we exploit statistical modeling rather than traditional threshold-based filtering, inspired by the statistical decision theory in the field of signal processing [19]. Specifically, instead of relying on fixed thresholds, we assess each point by its degree of similarity to a fitting noise distribution. Through theoretical analysis and experimental validation, we show that within local spatial regions, the signal metrics of noise points are well approximated by a Gaussian distribution. Accordingly, we design an adaptive noise distribution modeling algorithm that continuously updates voxel-wise distribution parameters over time. Incoming points are then evaluated against the corresponding fitted distribution, and those with a high degree of similarity are filtered out.

After exploiting intrinsic point-level signal metrics, we further discriminate UAVs from fake objects at the object level by analyzing cross-frame motion patterns. The point clouds are first clustered into individual objects, among which many



**Figure 4:** BSense adopts a layered framework consisting of (1) noise point filtering, (2) fake object filtering, and (3) fake trajectory filtering, which progressively suppress false detections to yield clean UAV trajectories.

fake ones remain. We observe that fake objects exhibit unstable and irregular motions, whereas true targets maintain consistent behaviors across frames. To capture this distinction, we evaluate object consistency from two perspectives: (1) *Spatial consistency*—true targets move continuously through space and can be reliably matched across consecutive frames. (2) *Velocity consistency*—the Doppler velocity measured by the base station is consistent with the radial velocity computed from the real target’s spatial displacement. Building on these consistencies, we define spatial- and velocity-based confidences, inspired by the eligibility trace concept [38] in reinforcement learning. These confidences incorporate two heuristics: (1) *Frequency heuristic*—the more frequently a consistency is satisfied, the higher the confidence score. (2) *Recency heuristic*—the more recently a consistency is satisfied, the higher the confidence score. Consequently, the confidence scores accumulate evidence over time, allowing us to reliably discard objects with low confidences.

Finally, objects are tracked into individual trajectories; however, some fake ones may still persist. To address this, we propose TrajFormer, a lightweight trajectory identification network inspired by the Transformer architecture [39], which leverages spatial motion and signal features across multiple frames to eliminate fake trajectories. By capturing long-range temporal dependencies and motion dynamics with its encoder, and aligning the current observation with historical context through its decoder, TrajFormer effectively distinguishes true trajectories from fake ones.

In summary, we make the following contributions:

- We propose BSense, the first system that achieves reliable wide-area UAV detection and tracking using a commercial 5G-A base station in a real deployment. A demonstration video is available at [todo](#).
- We propose a layered framework that filters noise at the point, object, and trajectory levels while preserving true UAV targets. First, we introduce an adaptive noise-feature distribution modeling method to remove points similar to the modeled noise distribution. Next, we define spatial-based and velocity-based confidence metrics, incorporating consistency checks and frequency-recency heuristics, to distinguish true

UAV targets from fake objects. Finally, we design a lightweight Transformer-based trajectory identification network to eliminate fake trajectories.

• We utilized a COTS 5G-A base station deployed in an urban environment and commanded the UAV to follow diverse flight trajectories (–,  $\circlearrowleft$ , S,  $\infty$ ,  $\square$ ,  $\diamond$ , M,  $\star$ ). Across 7 days, we collected 51 flight cases, yielding approximately 135 minutes of valid data (over 12,000 frames). We evaluate BSense on this dataset and achieve an average F1 score of 95.77%, with 96.36% precision, 95.19% recall, and a mean localization error of 4.9 m, demonstrating its accuracy and robustness in real-world urban conditions.

## 2 BSENSE DESIGN

### 2.1 Problem Definition

In this work, our goal is to achieve accurate UAV detection and tracking while effectively mitigating false detections. The input to our system is a point cloud  $P$  captured by the 5G-A base station. The point cloud is directly generated by the base station’s built-in signal-processing pipeline, and our system operates solely on this output as the application-layer perception module. Each point  $p \in P$  encodes spatial coordinates, Doppler velocity, and a set of signal quality metrics, including signal-to-noise ratio (SNR), signal-to-clutter ratio (SCR), noise level, signal power, and received power:

$$p = (x, y, z, v_d, snr, scr, nl, sp, rp), \quad p \in P. \quad (1)$$

The output of our system is the estimated position, velocity, and identity of each tracked UAV, represented as:

$$u = (x, y, z, v_x, v_y, v_z, id), \quad u \in U. \quad (2)$$

### 2.2 System Overview

Figure 4 presents an overview of BSense, a layered framework operating at the point, object, and trajectory levels. At the point level, the module models and automatically updates the noise distribution, filtering incoming points that are similar to the modeled noise distribution. At the object level, the module clusters points into objects, evaluates their consistency, and computes spatial-based and velocity-based confidences with frequency-recency heuristics to distinguish true targets from fake objects. At the trajectory level, the module

tracks objects and employs a lightweight Transformer-based identification network to eliminate fake trajectories.

### 2.3 Noise Point Filtering through Noise Distribution Modeling

We start our noise filtering at the point cloud level, specifically targeting the persistent background noise. These are typically caused by direct reflections from surrounding structures such as buildings, terrain, and trees. Despite the advanced signal-level processing capabilities of 5G-A base stations, a significant number of such noise points remain in the output point cloud, highlighting a fundamental limitation of traditional signal-domain clutter suppression.

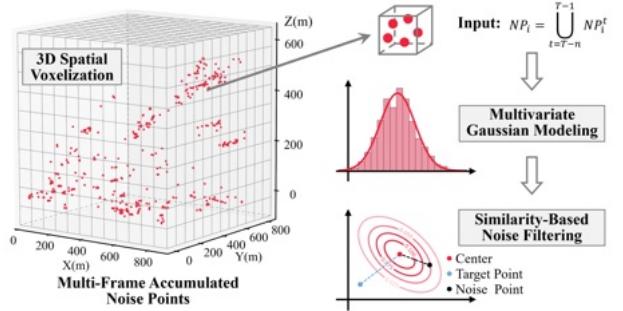
Conventional signal processing methods typically rely on the assumption that real targets can be distinguished from noise based on certain salient features, such as higher signal strength or different Doppler shifts. Accordingly, most systems adopt threshold-based filters to eliminate points with low signal strength. However, in our scenario, this assumption breaks down: UAVs are small, slow, and far from the base station. These factors lead to weak reflected signals and minimal Doppler shifts, making the UAVs' signal returns statistically similar to those of noise (as shown in Figure 2).

To address this, we shift the perspective from “*how strong is the signal*” to “*how similar is the signal to known noise distribution*.” Specifically, we hypothesize that within a local spatial region, the signal metrics (e.g., SNR, SCR) of noise points follow a consistent statistical distribution. Based on this, we propose a modeling approach that fits the local distribution of signal metrics for noise points across multiple frames, and filters incoming points by measuring their statistical deviation from the fitted distribution.

**2.3.1 Analysis of Noise Distribution.** We hypothesize that signal metrics of noise points within localized spatial regions follow a Gaussian distribution. This hypothesis is supported by both theoretical analysis and empirical evidence.

From a theoretical perspective, two key rationales underlie this assumption. First, the Central Limit Theorem (CLT) states that the sum or average of a large number of independent random variables tends toward a Gaussian distribution, regardless of their individual distributions. In our scenario, noise signals in a local area arise from the aggregation of reflections by numerous small, independent scatterers. Second, within a sufficiently small spatial region, the physical properties of the environment are typically homogeneous. This spatial consistency leads to stable scattering behavior, which supports the convergence of aggregated signal metrics toward a Gaussian distribution.

To empirically validate this hypothesis, we collect 10 minutes of noise data (900 frames) and accumulate the corresponding point clouds. The 3D space is then partitioned into



**Figure 5: Noise point filtering through distribution modeling.** The 3D space is voxelized, with noise in each voxel modeled by a multivariate Gaussian; points are filtered based on their similarity to the corresponding distribution.

voxels with an edge length of 40 m, each containing a localized set of points. For each voxel, we apply the Henze-Zirkler test [10] to assess whether the five signal metrics—SNR, SCR, noise level, signal power, and received power—jointly follow a multivariate Gaussian distribution. Results show that over 75% of voxels satisfy the Gaussianity assumption, confirming the feasibility of statistical modeling in our context. For the remaining voxels that fail the test, we observe that they often contain a higher proportion of complex ghost points, likely caused by multipath propagation or sidelobe interference, which leads to non-Gaussian signal patterns. These voxels are excluded from subsequent modeling.

In the following sections, we detail the procedures for Gaussian modeling, noise filtering, and parameter updating.

**2.3.2 Multivariate Gaussian Modeling.** As illustrated in Figure 5, we partition the 3D space into voxels (each with an edge length of 40 m) and model the noise distribution within each voxel using a multivariate Gaussian. Due to space limitations, the rationale behind the voxel size selection is discussed in Appendix B. To construct the initial noise model, we first collect point cloud data over a period during which no UAVs are present in the sensing area, denoted as  $NP$ . This ensures that all points in  $NP$  can be regarded as noise. For each voxel  $i$ , we define  $NP_i$  as the subset of noise points located within it:  $NP_i = \{p \in NP \mid p \in \text{voxel}_i\}$ .

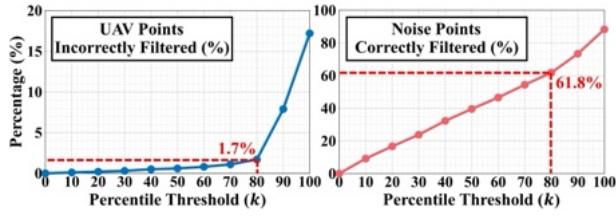
We represent each point  $p \in NP_i$  using a five-dimensional feature vector  $\mathbf{x}(p)$ , which consists of raw signal metrics:

$$\mathbf{x}(p) = [snr, scr, noise, sp, rp]^T, \quad \forall p \in NP_i. \quad (3)$$

To mitigate the effects of differing scales and units across signal metrics, we standardize each feature using z-score normalization [17]. For voxel  $i$ , we calculate the mean and standard deviation across all noise points:

$$\boldsymbol{\mu}_z^i = [\mu_{snr}^i, \mu_{scr}^i, \mu_{noise}^i, \mu_{sp}^i, \mu_{rp}^i]^T, \quad (4)$$

$$\boldsymbol{\sigma}_z^i = [\sigma_{snr}^i, \sigma_{scr}^i, \sigma_{noise}^i, \sigma_{sp}^i, \sigma_{rp}^i]^T. \quad (5)$$



**Figure 6: Percentage of filtered UAV points and noise points at different  $k$  values.**

The standardized vector  $\mathbf{z}(p)$  for each point  $p \in NP_i$  is then computed as:  $\mathbf{z}(p) = \frac{\mathbf{x}(p) - \hat{\mu}_i}{\sigma_z^i}$ .

Next, we estimate the parameters of noise distribution in voxel  $i$  using Maximum Likelihood Estimation (MLE). For a multivariate Gaussian distribution, the parameters consist of the mean vector ( $\hat{\mu}_i$ ) and the covariance matrix ( $\hat{\Sigma}_i$ ), which are computed as follows:

$$\hat{\mu}_i = \frac{1}{|NP_i|} \sum_{p \in NP_i} \mathbf{z}(p), \quad (6)$$

$$\hat{\Sigma}_i = \frac{1}{|NP_i|} \sum_{p \in NP_i} (\mathbf{z}(p) - \hat{\mu}_i)(\mathbf{z}(p) - \hat{\mu}_i)^\top. \quad (7)$$

Finally, for each voxel, we record the z-score standardization statistics and multivariate Gaussian parameters for subsequent noise filtering. The complete parameter set for voxel  $i$  is given by:  $\text{Param}_i = \{\boldsymbol{\mu}_z^i, \boldsymbol{\sigma}_z^i, \hat{\mu}_i, \hat{\Sigma}_i\}$ .

**2.3.3 Similarity-based Noise Filtering.** After modeling the noise distribution, we evaluate each point in the current frame for its similarity to the corresponding noise distribution using Mahalanobis distance [5]. This distance metric considers the correlations among features, making it well-suited for multivariate data. Points with high similarity to the noise distribution (i.e., small Mahalanobis distances) are identified as noise and subsequently removed.

Specifically, for each point  $p$  in the current frame, we locate its corresponding voxel  $i$ , standardize its feature vector, and compute the Mahalanobis distance as:

$$D(p; \text{Param}_i) = \sqrt{(\mathbf{z}(p) - \hat{\mu}_i)^\top \hat{\Sigma}_i^{-1} (\mathbf{z}(p) - \hat{\mu}_i)}, \quad (8)$$

where  $\mathbf{z}(p)$  is the z-score-normalized feature vector of point  $p$ , and  $\hat{\mu}_i$  and  $\hat{\Sigma}_i$  denote the estimated mean vector and covariance matrix of the noise distribution in voxel  $i$ , respectively.

We then classify a point as noise and remove it if its Mahalanobis distance falls below a predefined similarity threshold  $\tau_{sim}$ . This threshold is determined adaptively based on the distribution of Mahalanobis distances computed from the historical noise samples  $NP$ :

$$\tau_{sim} = \text{Percentile}_k(\{D(p; \text{Param}_i) \mid p \in NP\}), \quad (9)$$

where  $\text{Percentile}_k(\cdot)$  returns the  $k$ -th percentile of the computed distances. Empirically, we set  $k = 80$ , which means

80% of historical noise points in  $NP$  have a Mahalanobis distance below the threshold. This allows us to filter most of the noise points while mitigating the impact of outliers.

To evaluate the impact of the threshold parameter  $k$ , we plot in Figure 6 the percentages of filtered UAV points and noise points at different  $k$  values. As  $k$  increases, more noise points are removed, but the risk of incorrectly filtering UAV points also rises. At  $k = 80$ , we achieve a good balance—removing 61.8% of noise points while only filtering 1.7% of UAV points—making it a suitable choice for our system.

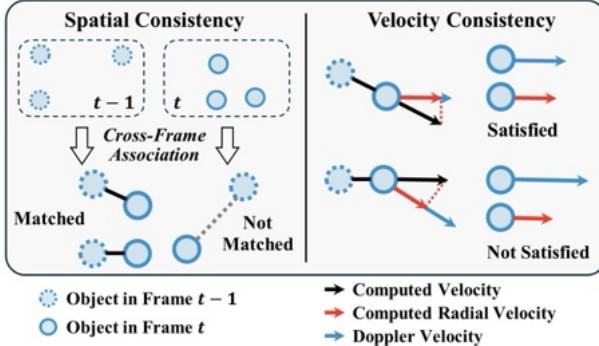
**2.3.4 Automatic Model Parameter Updating.** The last thing we need to consider is that over time, the signal reflection patterns in the environment may gradually shift, rendering the previously fitted Gaussian parameters inaccurate. To ensure the long-term robustness of our system, it is necessary to update the model parameters automatically.

We perform parameter updates at fixed intervals of  $T$ . At each update, we retrospectively collect the most recent  $N$  frames of point cloud data, denoted as  $P_{\text{upd}}$ . The choice of optimal values for  $T$  and  $N$  will be discussed in Section 3.3. However,  $P_{\text{upd}}$  may contain true target points from UAVs, which could contaminate the noise model. To address this, we leverage our system’s final detection results to identify and exclude true points from  $P_{\text{upd}}$ . The remaining subset, denoted as  $NP_{\text{upd}}$ , contains only noise points and is subsequently used to refit the multivariate Gaussian models and recompute the similarity threshold.

**Key Result:** After this module, the average number of noise points per frame reduces from **168.05** to **64.15**, achieving a **61.8%** reduction, as detailed in Section 3.3.

## 2.4 Fake Object Filtering Using Spatial- and Velocity-based Confidences

After leveraging intrinsic signal metrics at the point level, we further analyze the cross-frame motion patterns to filter out fake objects, which are mainly caused by multi-path effects and sidelobe interference. We observe that such fake objects typically exhibit disorganized and irregular motion patterns. In contrast, real targets tend to follow consistent motion behaviors across frames. To capture this distinction, we define two key forms of consistency: (1) *Spatial consistency*—real targets move continuously through space and can be reliably matched across consecutive frames. (2) *Velocity consistency*—the Doppler velocity measured by the base station is consistent with the radial velocity computed from the real target’s spatial displacement. These motion consistencies provide reliable cues for distinguishing true targets from fake objects. Building on this insight, we propose a confidence-based filtering method that leverages both spatial and velocity confidences to identify true targets and remove



**Figure 7: Spatial consistency** is satisfied when an object is successfully matched across frames. **Velocity consistency** is satisfied when the radial velocity computed from spatial displacement is consistent with the Doppler velocity.

fake objects. The design of these confidences is inspired by the eligibility trace concept [38] in reinforcement learning, which incorporates frequency and recency heuristics. We next formalize the computation of these scores.

**2.4.1 Point Clustering.** Since motion characteristics are exhibited by the object as a whole, we first extend the DBSCAN algorithm [9] to cluster points into individual objects more accurately. A cluster is formed when a point has sufficient neighboring points within a distance threshold  $\tau_d$ . To avoid the erroneous merging of spatially adjacent objects, we incorporate Doppler velocity into the distance metric. The resulting metric is defined as:

$$d = \|(a_x(x_i - x_j), a_y(y_i - y_j), a_z(z_i - z_j), a_v(v_i - v_j))\|_2. \quad (10)$$

Here, we empirically set the weights to  $(a_x, a_y, a_z, a_v) = (1, 1, 1, 3)$ , and the distance threshold to  $\tau_d = 10$ . Additionally, given the small size of the UAV, which may appear as a single point, we also retain clusters that contain only one point.

**2.4.2 Spatial-based Confidence.** We design this confidence based on the principle that real targets move continuously through space and exhibit spatial consistency. As illustrated in Figure 7, after obtaining the objects from clustering, we employ the Kuhn-Munkres matching algorithm [21] to establish spatial associations between objects across neighboring frames. Two objects are considered a match if their Euclidean distance is less than 15 m. Given the 0.64 s interval between consecutive frames, this threshold is sufficient to cover the inter-frame displacement of commercial UAVs [6] operating at typical maximum speeds (below 20 m/s).

For each object  $obj$  in the  $t$  th frame, we set  $I_s(obj, t) = 1$  if a match is identified; otherwise,  $I_s(obj, t) = 0$ . To quantify how consistently an object is matched over time, we define the *spatial-based confidence*  $C_s$  as:

$$C_s(obj, t) = \gamma_s \cdot (C_s(obj', t - 1) + I_s(obj, t)), \quad (11)$$

where  $\gamma_s \in [0, 1]$  is a decay factor that controls the forgetting rate of historical information (empirically set to 0.9 following [7]), and  $obj'$  denotes the object in frame  $(t - 1)$  matched to  $obj$ . This recursive formulation updates the confidence of each object over time, giving more weight to recent matches. As a result, the confidence score captures both the frequency and recency of successful associations, providing a robust measure of spatial consistency.

Since real targets move continuously in space and exhibit spatial consistency, a higher confidence score  $C_s$  indicates a higher probability that the object is a true target.

**2.4.3 Velocity-based Confidence.** In addition to spatial consistency, real targets are also expected to exhibit velocity consistency: the Doppler velocity measured by the base station should align with the radial velocity inferred from spatial displacement. Specifically, we first compute the radial velocity of the object relative to the base station based on the spatial association between two consecutive frames:

$$v_r = \frac{\text{pos}(obj, t) - \text{pos}(obj', t - 1)}{\Delta t} \cdot \frac{\text{pos}(obj, t)}{\|\text{pos}(obj, t)\|}, \quad (12)$$

where  $\text{pos}(obj, t)$  and  $\text{pos}(obj', t - 1)$  denote the positions of the matched object in frames  $t$  and  $(t - 1)$ , respectively.

We then evaluate the consistency between the computed radial velocity  $v_r$  and the Doppler velocity  $v_d$  measured by the base station, as illustrated in Figure 7. This consistency is quantified using the following formulation:

$$I_v(obj, t) = (|v_r - v_d| < \tau_{c1}) \wedge \left( \frac{\max(v_r, v_d)}{\min(v_r, v_d)} < \tau_{c2} \right) \wedge (v_r \cdot v_d > 0), \quad (13)$$

where we empirically set  $\tau_{c1} = 2$  and  $\tau_{c2} = 2$ . If the condition is satisfied,  $I_v(obj, t) = 1$ ; otherwise,  $I_v(obj, t) = 0$ .

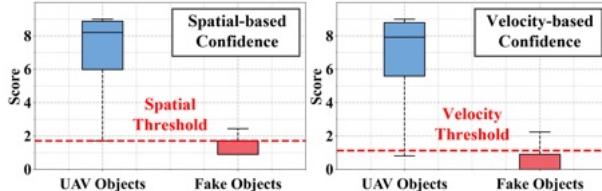
Following the same recursive formulation as the spatial confidence, we define the *velocity confidence*  $C_v$  as:

$$C_v(obj, t) = \gamma_v \cdot (C_v(obj', t - 1) + I_v(obj, t)), \quad (14)$$

where  $\gamma_v$  is also empirically set to 0.9 following [7].

Since the Doppler velocity measured by the base station is expected to be consistent with the radial velocity computed from the spatial displacement of a real target, a higher  $C_v$  score indicates a greater likelihood that the object corresponds to a true target.

**2.4.4 Confidence-based Object Discrimination.** After computing the spatial- and velocity-based confidence scores, we leverage these measures to discriminate between true and fake objects. Objects that fail to meet the spatial–velocity confidence criteria, specifically those with  $C_s < \tau_s$  and  $C_v < \tau_v$ , are filtered out. The thresholds  $\tau_s$  and  $\tau_v$  are critical parameters that must be carefully chosen to balance the suppression of fake objects with the retention of true targets. To determine appropriate thresholds, we analyze the confidence score



**Figure 8: Statistical ranges of spatial- and velocity-based confidences.** Thresholds are set at the 5th percentile of the UAV confidence scores, with  $\tau_s = 1.7$  and  $\tau_v = 1.1$ .

distributions of both UAVs and fake objects. As shown in Figure 8, both of the proposed confidences provide clear separation between UAVs and fake objects. Accordingly, we adopt a percentile-based thresholding strategy: to retain more UAVs, we set the thresholds at the 5th percentile of the UAV confidence scores ( $\tau_s = 1.7$  and  $\tau_v = 1.1$ ). This configuration ensures high recall for UAVs while maintaining effective filtering of fake objects.

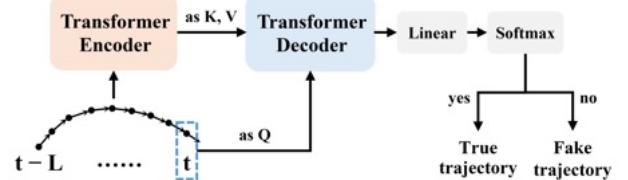
**Key Result:** After this module, the average number of fake objects per frame reduces from **61.34** to **2.58**, achieving a **95.79%** reduction, as detailed in Section 3.4.

## 2.5 Object Tracking and Fake Trajectory Filtering

After obtaining the object results, we track their positions over time while simultaneously identifying the generated trajectories. Notably, the vast majority of false detections have already been removed by the point-level and object-level filtering methods. We thus propose a lightweight trajectory identification network that operates in parallel with tracking, continuously identifying and filtering out the remaining false trajectories in real time.

**2.5.1 IMM-UKF-based Object Tracking.** To continuously and accurately track the UAV's position, we employ the Interacting Multiple Model Unscented Kalman Filter (IMM-UKF) [32], which offers robust tracking under diverse motion patterns. In this framework, the IMM dynamically switches among multiple motion models, while the UKF ensures accurate state estimation in nonlinear systems. In particular, we define the object state vector to include position, azimuth angle, elevation angle, velocity magnitude, and angular velocity, represented as  $S = [x, y, z, \phi, \theta, v, \omega_\phi]$ .

We consider two motion models in our IMM framework. The first is the linear Constant Velocity (CV) model, which assumes that an object moves at a constant speed. The second is the nonlinear Constant Turn Rate and Velocity (CTRV) model, which assumes that an object follows a constant turn rate in the horizontal plane while maintaining a constant velocity in the vertical direction. The tracking algorithm maintains each trajectory and updates it based on the matched object observation ( $obj$ ). A trajectory candidate is only confirmed and output after it has been consistently associated



**Figure 9: The overview of our Transformer-based trajectory identification network.**

with  $obj$  across  $L$  consecutive frames, where we set  $L = 6$ , a value determined through experiments in Section 3.5.1, balancing accuracy and output latency.

**2.5.2 Transformer-based Trajectory Identification.** During tracking, false trajectories may still emerge. To address this, we propose TrajFormer, a lightweight Transformer-based trajectory identification network that operates jointly with the tracker to continuously distinguish true trajectories from false ones. Each trajectory is evaluated in real time using the most recent  $L$  observations until it is verified as a true target and then output. Each  $obj$  comprises spatial motion and signal features:  $obj = (x, y, z, v_d, snr, scr, nl, sp, rp)$ .

As shown in Figure 9, TrajFormer first employs a Transformer encoder [39] to extract temporal features from multi-frame observations ( $t - L, \dots, t$ ). The current observation ( $t$ ) is then used as the query ( $Q$ ), while the encoder outputs serve as the key ( $K$ ) and value ( $V$ ) in a Transformer decoder [39], which integrates the current frame with its historical context. A linear layer followed by a softmax classifier then determines whether the trajectory corresponds to a true target or a false alarm. The encoder captures long-range temporal dependencies and motion dynamics across consecutive frames, producing a compact representation of trajectory evolution. The decoder conditions on the current observation and aligns it with the encoded historical context, thereby highlighting motion consistency or irregularities. This design enables the network to learn multi-frame motion patterns and effectively distinguish true trajectories from false ones.

**Key Result:** After this module, the average number of fake objects per frame decreases from **2.58** to **0.04**, and BSense achieves an **F1 score of 95.77%**, as detailed in Section 3.5.

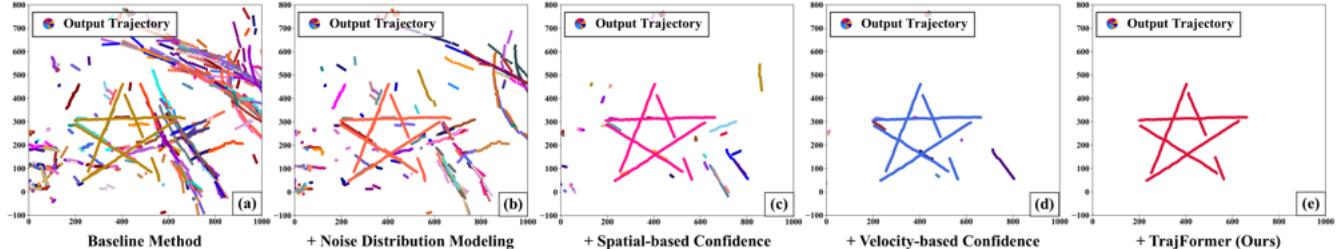
## 3 BSENSE EVALUATION

### 3.1 Experimental Setup

**3.1.1 System Setup.** We collected data using a Huawei COTS 5G-A base station [13, 14] operating at 4.9 GHz with a 100 MHz sensing bandwidth, deployed in an urban environment in Shanghai, China. Figure 10 shows the base station's Active Antenna Unit (AAU) and the DJI Mavic 3T UAV [6] used in our field trials. The AAU was positioned at a height of 23 m, providing a sensing range of up to 1,000 m, with a horizontal field of view (FOV) of 130° and a vertical FOV of 45°. As



**Figure 10: The 5G-A base station Active Antenna Unit (AAU) and the UAV deployed in our field experiments.**

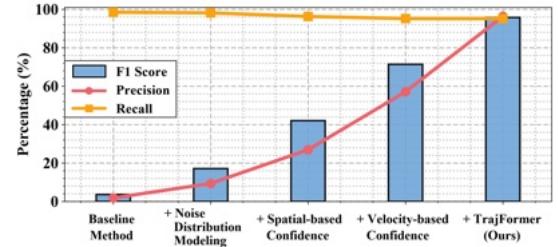


**Figure 12: Comparison of trajectory visualizations in the ablation study.** From left to right, the symbol “+” indicates that the methods are progressively constructed through the incremental addition of our proposed modules.

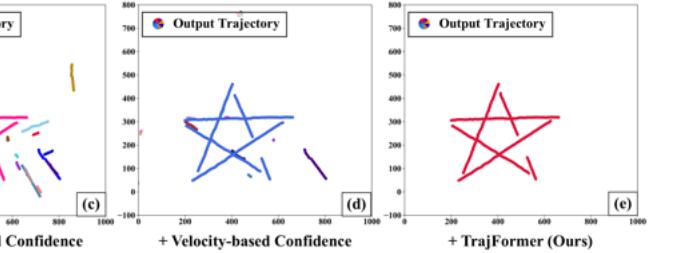
shown in Figure 1, the coverage area encompassed residential buildings, factories, overpasses, and a river, providing a complex environment for validating BSense.

**3.1.2 Data Acquisition.** The UAV was instructed to follow a set of shaped flight paths ( $\circlearrowleft$ ,  $S$ ,  $\infty$ ,  $\square$ ,  $\diamond$ ,  $M$ ,  $\star$ ) as well as straight-line paths (—) within the sensing area. For the straight-line flights, the trajectory-boresight angles were set to  $-85^\circ$ ,  $-75^\circ$ , ...,  $75^\circ$ , and  $85^\circ$ . Overall, these paths resulted in 25 distinct trajectories. The UAV was equipped with GPS, which provided the ground-truth locations for evaluation. The base station outputs point cloud data at a rate of one frame every 640 milliseconds. Data were collected continuously over 7 days, resulting in 51 flight cases with a total duration of approximately 135 minutes (over 12,000 frames). Among these, a 10-minute case without UAV flights, recorded at the beginning of the collection period, was used to initialize the noise distribution model, as described in Section 2.3.2. The remaining 50 cases were collected by flying each path type twice, with path lengths ranging from 600 m to 2,300 m.

**3.1.3 Train-Test Split.** Since the TrajFormer introduced in Section 2.5.2 requires training, evaluation was performed using 5-fold cross-validation [45]. To ensure fair evaluation and to verify generalization, the split was performed at the level of entire flight cases, and cases of the same trajectory type were never placed in both training and testing folds. During training, only the cases in the training folds were segmented using a fixed-size sliding window to produce positive and negative samples, following ground-truth labels. Testing was performed sequentially on full trajectories.



**Figure 11: Ablation study performance comparison.** The “+” indicates stepwise integration of proposed modules.

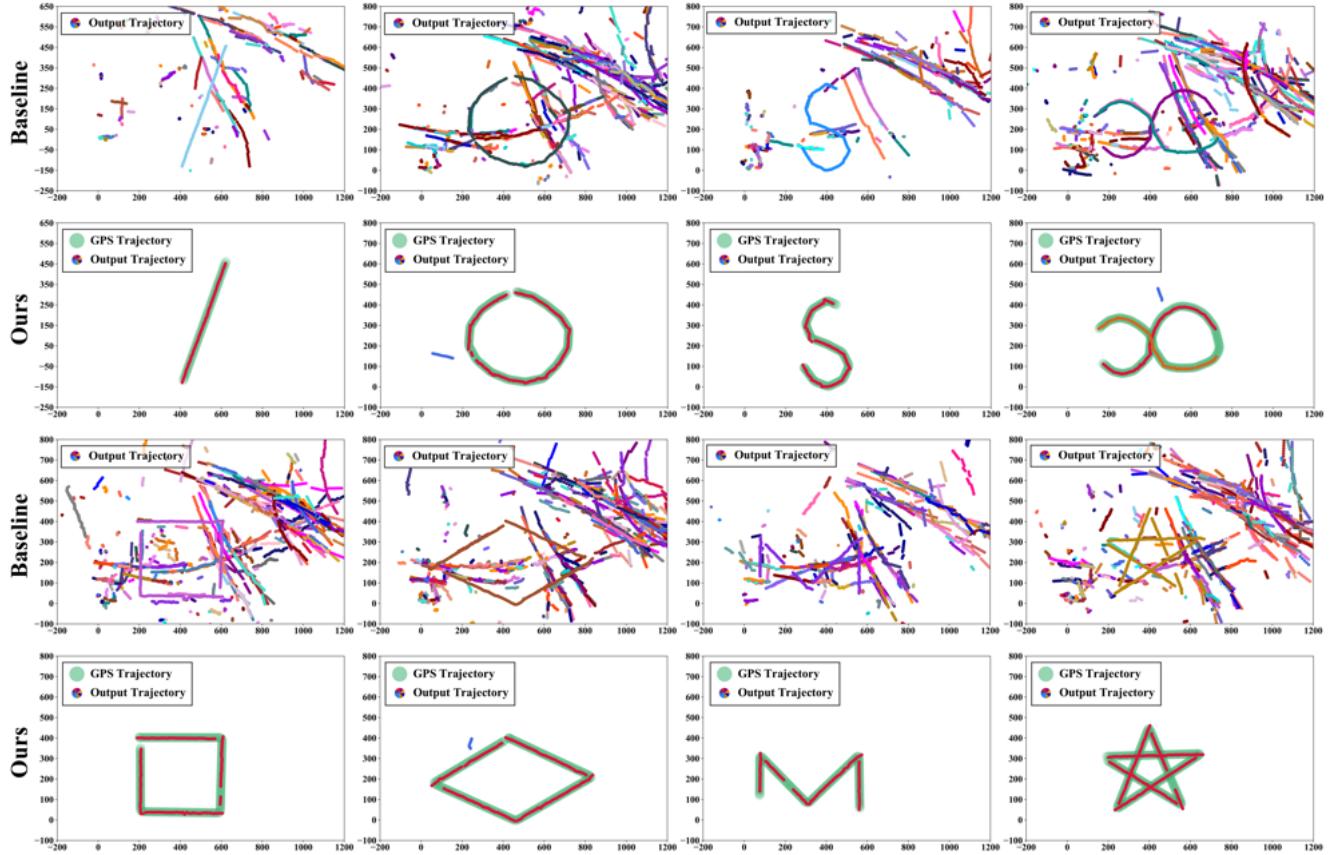


**3.1.4 Performance Metrics.** We use the following metrics to evaluate BSense, focusing on accurate UAV detection and tracking as well as effective filtering of false targets:

- **Point-Level Metrics:** Noise points are first filtered based on signal distribution modeling, as described in Section 2.3. To evaluate this module, we define point-level metrics as follows. A point is classified as a UAV point if its Euclidean distance to the ground truth is within 20 m; otherwise, it is treated as noise. Let  $c_u$  and  $c_n$  denote the numbers of UAV and noise points before filtering, and  $c'_u$  and  $c'_n$  the corresponding numbers after filtering. The incorrect filtering rate of UAV points is defined as  $F_u = \frac{1}{c_u}(c_u - c'_u)$ , and the correct filtering rate of noise points as  $F_n = \frac{1}{c_n}(c_n - c'_n)$ .

- **Object-Level Metrics:** Then, we obtain object results from the point clustering and fake-object filtering described in Section 2.4. For evaluation, we perform minimum-distance binary matching between the detected objects and the ground truth. A detected object is classified as a true positive ( $TP$ ) if the matched distance is within 20 m; otherwise, it is counted as a false positive ( $FP$ ). Ground-truth objects without a match are treated as false negatives ( $FN$ ). The UAV recall ( $R$ ) is computed as  $R = \frac{TP}{TP+FN}$ , while the average number of fake objects per frame ( $N_f$ ) is defined as  $N_f = \frac{FP}{\text{total frames}}$ .

- **Trajectory-Level Metrics:** Finally, using the object tracking and fake trajectory filtering described in Section 2.5, we evaluate the overall tracking performance, characterized by the F1 score and the mean localization error. The F1 score is computed as the harmonic mean of precision  $P$  and recall  $R$ , where  $P = \frac{TP}{TP+FP}$ ,  $R = \frac{TP}{TP+FN}$ , and  $F1 = \frac{2PR}{P+R}$ . The mean localization error  $E_{\text{loc}}$  is calculated as  $E_{\text{loc}} = \frac{1}{N} \sum_{i=1}^N \|\hat{\mathbf{pos}}_i -$



**Figure 13: Visual comparison of baseline and our BSense across different flight paths.** Compared with the baseline, our system effectively eliminates fake trajectories and achieves accurate UAV detection and tracking.

$\text{pos}_i\|_2$ , measuring the average Euclidean distance between the system-output and ground-truth UAV positions. Since the primary challenge of this study lies in numerous false detections, we focus particularly on the F1 score.

## 3.2 Overall Performance

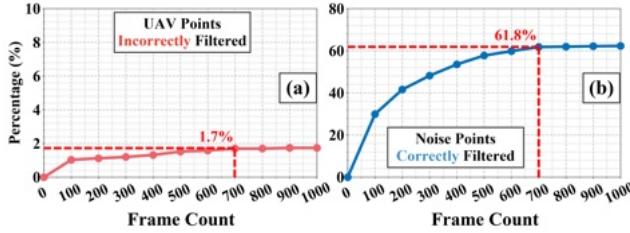
**3.2.1 Ablation Study.** We perform an ablation study by incrementally adding our modules to the baseline method and evaluating their contributions on the full dataset:

- (1) *Baseline Method*: The raw point cloud is directly clustered with DBSCAN [9] and the resulting objects are tracked with an IMM-UKF [32].
- (2) + *Noise Distribution Modeling*: Building on *method (1)*, we incorporate noise point filtering based on noise distribution modeling, as described in Section 2.3.
- (3) + *Spatial-based Confidence*: Extending *method (2)*, we further introduce spatial-based confidence for filtering fake objects, as discussed in Section 2.4.
- (4) + *Velocity-based Confidence*: Extending *method (3)*, we further introduce velocity-based confidence for filtering fake objects, as discussed in Section 2.4.

(5) + *TrajFormer (Ours)*: Finally, building on *method (4)*, we employ the proposed TrajFormer network to filter fake trajectories, as detailed in Section 2.5.

Notably, methods (1)–(5) demonstrate the stepwise integration of our proposed modules, with the corresponding results presented in Figure 11. The baseline exhibits the lowest performance, with an F1 score of only 3.65%. This poor result is caused by the substantial number of noise points (Figure 1), which generate numerous false positives and lead to an extremely low precision of 1.86%. Building on this baseline, we progressively incorporate noise distribution modeling, spatial-based confidence, velocity-based confidence, and TrajFormer. Each module further suppresses false positives, yielding cumulative improvements in F1 and ultimately increasing the score from 3.65% to 95.77%. This progression demonstrates the effectiveness of each module in our system.

In addition, Figures 12 (a)–(e) present trajectory visualizations of methods (1)–(5) on the  $\star$ -trajectory case, with different colors representing different trajectories. As shown, with the incremental integration of our proposed modules, fake trajectories are progressively eliminated, while the UAV



**Figure 14:** (a) Incorrect filtering rate of UAV points; (b) Correct filtering rate of noise points under different data frame counts for noise distribution modeling.

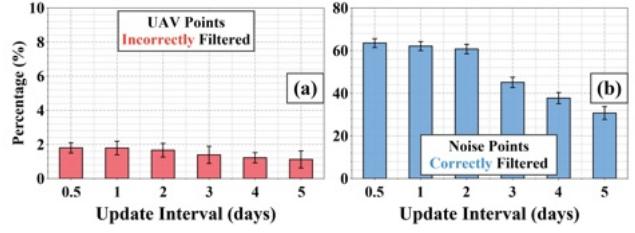
trajectory is fully preserved. These visualizations further demonstrate the effectiveness of our system in achieving accurate UAV detection and tracking.

**3.2.2 Visualization Results.** The system's per-frame outputs are saved and visualized. Figure 13 presents a comparison between the baseline and our BSense across different flight paths ( $-$ ,  $\circ$ ,  $S$ ,  $\infty$ ,  $\square$ ,  $\diamond$ ,  $M$ ,  $\star$ ). The first and third rows show the baseline results, while the second and fourth rows display the corresponding outputs of BSense. As illustrated, compared with the baseline, BSense effectively eliminates fake trajectories while accurately detecting and tracking the UAV. For the  $\infty$ ,  $M$ , and  $\star$  trajectories, partial gaps appear due to the continuous absence of target raw points during certain periods. Overall, these results highlight the strong performance of BSense, which suppresses false detections and produces clean UAV trajectories across flight paths ranging from simple ( $-$ ) to complex ( $\star$ ).

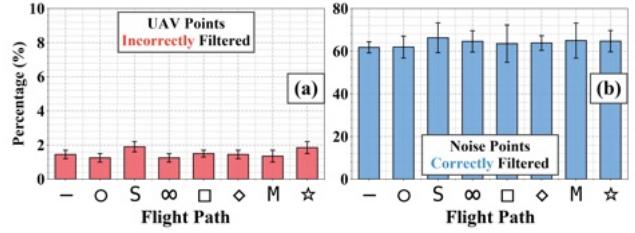
**3.2.3 Processing Speed.** In addition to accurate detection and tracking, processing speed is crucial for the practicality of a real-time system. To evaluate this, we executed our system on an Intel Core i5-11500 CPU [16]. The results show that the three sequential stages—noise point filtering, fake object filtering, and fake trajectory filtering—achieve average per-frame processing times of 6.91 ms, 15.12 ms, and 16.34 ms, respectively. In total, the system requires 38.37 ms per frame, which corresponds to approximately 25 frames per second (FPS). This runtime is substantially shorter than the base station's raw point cloud output interval (640 ms per frame), demonstrating that the proposed system can operate in real time and is well-suited for practical deployment.

### 3.3 Performance of Noise Point Filtering

**3.3.1 Data Frame Count for Noise Distribution Modeling.** The system adaptively models the noise distribution using the most recent  $N$  point cloud frames, as described in Section 2.3. Increasing  $N$  enhances the accuracy of parameter estimation, but it also introduces additional storage overhead. Figure 14 presents the filtering performance under different values of  $N$ . As shown in Figure 14(b), the correct filtering



**Figure 15:** (a) Incorrect filtering rate of UAV points; (b) Correct filtering rate of noise points under different update intervals for model parameter updates.



**Figure 16:** (a) Incorrect filtering rate of UAV points; (b) Correct filtering rate of noise points across different UAV flight paths.

rate of noise points improves with larger  $N$ , but the incremental gains diminish substantially once  $N$  exceeds 700 frames. Meanwhile, Figure 14(a) shows that the incorrect filtering rate of UAV points consistently remains below 2%. Considering both filtering effectiveness and storage overhead, we set  $N = 700$  for noise distribution modeling and parameter updates. Under this configuration, our noise distribution modeling-based filtering method reduces the average number of noise points per frame from 168.05 to 64.15, achieving a 61.8% reduction while only filtering 1.7% of UAV points.

**3.3.2 Model Parameter Update Interval.** To ensure the robustness of noise point cloud filtering over time, the system periodically updates the parameters of the noise distribution model with an interval of  $T$ , as described in Section 2.3.4. Figure 15 reports the filtering performance under different values of  $T$ . As shown in Figure 15(b), the correct filtering rate of noise points decreases with increasing  $T$ , with a pronounced drop once  $T$  exceeds 2. Meanwhile, Figure 15(a) shows that the incorrect filtering rate of UAV points consistently remains below 2%. Based on these results, we set the update interval to  $T = 2$ , which balances update frequency and ensures stable filtering performance.

**3.3.3 Different UAV Flight Paths.** For UAVs that may appear at any location within the sensing area, it is essential to ensure they are not mistakenly filtered out. Figure 16 illustrates the performance of the proposed noise point filtering method across different UAV flight paths (−,  $\circ$ ,  $S$ ,  $\infty$ ,  $\square$ ,  $\diamond$ ,  $M$ ,  $\star$ ). As shown in Figure 16(b), the correct filtering rate of noise points consistently exceeds 60%, while Figure 16(a)

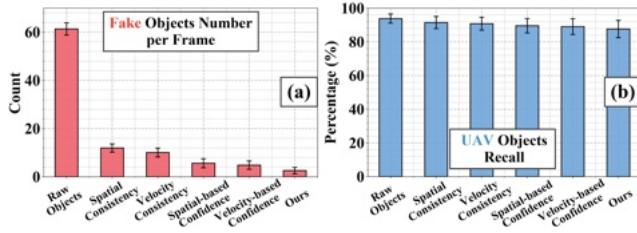


Figure 17: (a) Number of fake objects per frame; (b) Recall of UAV objects under varying filtering methods.

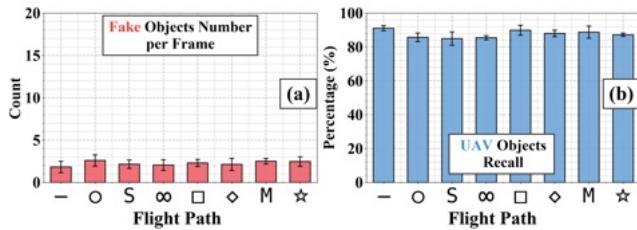


Figure 18: (a) Number of fake objects per frame; (b) Recall of UAV objects across different UAV flight paths.

shows that the incorrect filtering rate of UAV points remains below 2% across all flight paths. These results demonstrate the robustness and generalizability of the proposed method, which effectively filters out over 60% of noise points while preserving more than 98% of UAV points.

### 3.4 Performance of Fake Object Filtering

**3.4.1 Different Filtering Methods.** We evaluate the effectiveness of different methods for filtering fake objects:

- (1) *Raw Objects*: object results obtained directly from DBSCAN clustering [9].
- (2) *Spatial Consistency*: filtering out objects that do not satisfy spatial consistency.
- (3) *Velocity Consistency*: filtering out objects that do not satisfy velocity consistency.
- (4) *Spatial-based Confidence*: filtering out objects with spatial-based confidence below the threshold.
- (5) *Velocity-based Confidence*: filtering out objects with velocity-based confidence below the threshold.
- (6) *Ours*: filtering out objects with both spatial- and velocity-based confidences below their thresholds.

Figure 17 compares the performance of these methods. As shown in Figure 17(a), applying spatial consistency, velocity consistency, and the derived spatial- and velocity-based confidence significantly reduces the number of fake objects. Meanwhile, Figure 17(b) shows that UAV recall remains above 85% across all methods. Overall, compared with raw objects, our filtering method using spatial-based and velocity-based confidence reduces the average number of fake objects from 61.34 to 2.58 per frame, achieving a 95.79% reduction while maintaining an 87.64% UAV recall. Moreover, UAV recall can be further improved through subsequent object tracking.

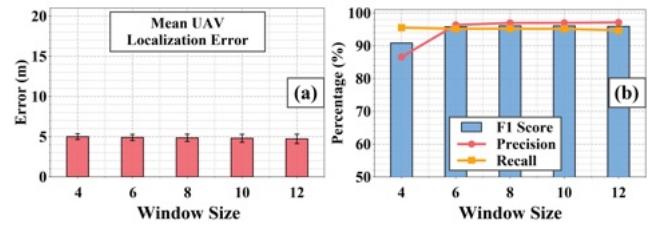


Figure 19: UAV tracking performance under varying window sizes: (a) mean localization error; (b) F1 score.

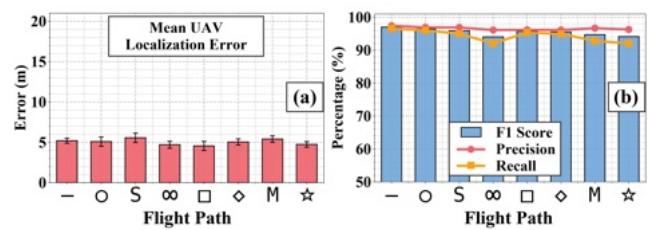


Figure 20: UAV tracking performance across different flight paths: (a) mean localization error; (b) F1 score.

**3.4.2 Different UAV Flight Paths.** For UAVs that may appear anywhere within the sensing area, it is critical to ensure they are not incorrectly discarded. Figure 18 illustrates the performance of the proposed fake object filtering method across different UAV flight paths (-, ○, S, ∞, □, ◇, M, ☆). As shown in Figure 18(a), the number of fake objects per frame consistently remains below 3, while Figure 18(b) indicates that the recall of UAV objects consistently exceeds 85% across all flight paths. These results highlight the robustness and adaptability of the proposed method, demonstrating reliable performance under diverse flight scenarios.

### 3.5 Performance of UAV Tracking

**3.5.1 Trajectory Identification Window Size.** The system determines whether a trajectory is a true target or a fake using TrajFormer, which processes the most recent  $L$  observations, as described in Section 2.5.2. A larger window size  $L$  enables TrajFormer to capture richer temporal information, thereby improving trajectory identification. However, increasing  $L$  also introduces output latency, as the system must accumulate more observations before making a decision.

Figure 19 presents the tracking performance across different window sizes. Figure 19(b) shows that when  $L$  exceeds 6, the improvements in F1 score become marginal, while Figure 19(a) demonstrates that the mean UAV localization error consistently remains below 6 m, satisfying the requirement for kilometer-scale sensing. To balance detection performance and output latency, we set  $L = 6$  in our experiments. Under this configuration, the system achieves strong detection performance with an F1 score of 95.77% (precision 96.36%, recall 95.19%) and maintains a mean localization error of 4.9 m.

**3.5.2 Different UAV Flight Paths.** For flight paths ranging from simple (–) to complex (☆), obtaining clean and reliable UAV trajectories is crucial. Figure 20 presents the tracking performance across different flight paths (–, ○, S, ∞, □, ◇, M, ☆). Precision consistently exceeds 96% across all paths, highlighting the effectiveness of our system. However, the recall for the ∞, M, and ☆ trajectories is comparatively lower, primarily due to the continuous absence of target raw points during certain periods (Figure 13). The corresponding F1 scores for these trajectories are 94.03%, 94.72%, and 94.13%, respectively, whereas all other paths achieve scores above 95%. This limitation, arising from sustained target point absence, will be further analyzed in Section 4 as part of our discussion on system constraints.

## 4 DISCUSSION

While BSense has achieved high performance in complex real-world environments, it is important to acknowledge its limitations and potential research directions in the future.

**Trajectory Interruption.** In our experiments, we observed that the UAV trajectories output by the system occasionally exhibit interruptions lasting for 50–100 m, as illustrated in the ∞ and ☆ examples in Figure 13. These interruptions are not caused by algorithmic misses, but rather by prolonged target absences in the raw point cloud data. We found that such cases typically arise when the UAV is occluded by tall urban buildings or when its flight direction remains tangential to the base station. Traditional tracking and re-identification algorithms are ineffective in addressing this issue. In future work, we plan to leverage deep learning to extract motion features from both preceding and succeeding trajectory segments, enabling automatic re-identification and seamless completion of interrupted trajectories. This will provide more robust and continuous UAV detection and tracking.

**Limited Localization Accuracy.** Our system achieves accurate UAV detection with a mean localization error of 4.9 m. While attaining meter-level accuracy at a kilometer-scale sensing range is already nontrivial, this precision still leaves room for improvement to support broader applications. It is worth noting that this work primarily focuses on filtering massive false detections in the raw data, whereas the remaining localization error mainly stems from measurement noise inherent in the base station point clouds. Optimizing localization accuracy is thus beyond the scope of this paper. Potential directions for future improvement include increasing the signal bandwidth or the number of antennas at the base station to enhance ranging accuracy and angular resolution, thereby producing higher-quality point clouds.

## 5 RELATED WORK

**Sensor-based UAV Perception.** Recent research on UAV detection and tracking has investigated the use of different

sensors, primarily cameras and radars [12, 15, 20, 36, 40, 41, 46, 47, 50, 51]. For instance, Zhao et al.[51] developed a vision-based UAV detection and tracking framework that integrates a YOLOv5 backbone with a Kalman filter for anti-UAV applications. Similarly, Wang et al.[41] proposed a radar-based UAV detection method using pulse-Doppler radar and a CNN-BiLSTM model to capture spatio-temporal features in complex environments. Despite these advances, sensor-based approaches remain constrained: camera systems have limited sensing ranges and are highly sensitive to lighting conditions [34], while radar systems are often costly and less suitable for large-scale deployment [33].

In this paper, we leverage 5G-A base stations for precise UAV detection and tracking. Unlike prior sensor-based systems, our system directly reuses existing communication base station infrastructure, enabling large-scale deployment without additional sensing hardware.

**Base Station-based Sensing.** With the advancement of Integrated Sensing and Communication (ISAC), 5G-A base stations have been explored as sensing platforms in addition to providing communication services [23, 29, 43, 44]. Recent studies [8, 11, 22, 26, 28, 31, 35, 42, 49, 52, 54] have explored the use of 5G-A base stations for vehicle detection, UAV detection, and environment reconstruction. However, these efforts primarily focus on ISAC architecture design and signal processing optimization, with evaluations limited to theoretical analyses or idealized simulations. To the best of our knowledge, the only real-world deployment reported so far is by Lu et al. [27], which leverages an ISAC base station for maritime vessel detection and tracking. Yet, their experiments were conducted in open environments where the large vessels are relatively easy to detect.

In this paper, we use a 5G-A base station in a complex urban environment for UAV sensing. Unlike vessels, small and slow-moving UAVs produce weak reflections, resulting in numerous false detections in the raw data. For the first time, we reveal and address this challenge in the wild—marking a concrete step toward practical ISAC sensing.

## 6 CONCLUSION

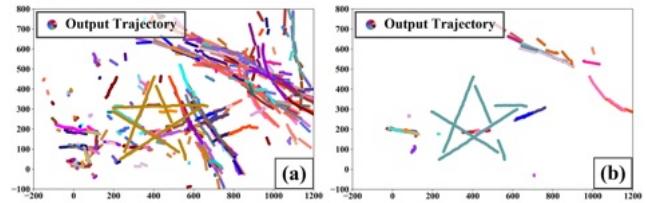
We have presented BSense, the first practical system that achieves accurate UAV detection and tracking using commercial 5G-Advanced base stations. Our system introduces a layered framework that progressively filters noise at the point, object, and trajectory levels. Through deployment in a complex urban environment, we demonstrate that BSense reduces hundreds of false detections per frame to nearly zero, while achieving meter-level localization accuracy at kilometer-scale sensing ranges. We believe that the design, implementation, and evaluation of BSense mark an important step in advancing ISAC from theoretical exploration to real-world UAV detection and tracking at scale.

## References

- [1] Yunes Alqudsi and Murat Makaraci. 2025. UAV swarms: research, challenges, and future directions. *Journal of Engineering and Applied Science* 72, 1 (2025), 12.
- [2] Jiabin Chen, Wupeng Xie, Chaowei Wang, Mingliang Pang, Fan Jiang, and Lexi Xu. 2024. Resilient Massive Access assisted ISAC in Space-Air-Ground Integrated Networks. In *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*. 2203–2208.
- [3] Zhicheng Chen, Zhenzhe Qu, Nicholas Xiong, Anfeng Liu, Mianxiong Dong, Tian Wang, and Shaobo Zhang. 2024. UITDE: A UAV-assisted intelligent true data evaluation method for ubiquitous IoT systems in intelligent transportation of smart city. *IEEE Transactions on Intelligent Transportation Systems* 25, 8 (2024), 9597–9607.
- [4] Mingyu Cheng, Jiqian Yang, Tingye Pan, Qi Liu, Zhi Li, and Shijin Wang. 2025. Convtimeenet: A deep hierarchical fully convolutional model for multivariate time series analysis. In *Companion Proceedings of the ACM on Web Conference 2025*. 171–180.
- [5] Roy De Maesschalck, Delphine Jouan-Rimbaud, and Désiré L Massart. 2000. The mahalanobis distance. *Chemometrics and intelligent laboratory systems* 50, 1 (2000), 1–18.
- [6] DJI. 2022. DJI Mavic 3 Enterprise Series. [Online]. <https://enterprise.dji.com/mavic-3-enterprise>.
- [7] Yifan Duan, Jie Peng, Yu Zhang, Jianmin Ji, and Yanyong Zhang. 2022. Pfilter: Building persistent maps through feature filtering for fast and accurate lidar-based slam. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 11087–11093.
- [8] Mohamed Elfiatoure, Mohammadali Mohammadi, Hien Quoc Ngo, Hyundong Shin, and Michail Matthaiou. 2025. Multiple-target detection in cell-free massive MIMO-assisted ISAC. *IEEE Transactions on Wireless Communications* (2025).
- [9] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, Vol. 96. 226–231.
- [10] Norbert Henze and Bernd Zirkler. 1990. A class of invariant consistent tests for multivariate normality. *Communications in statistics-Theory and Methods* 19, 10 (1990), 3595–3617.
- [11] Zelin Hu, Qibin Ye, Yixuan Huang, Su Hu, and Gang Yang. 2024. Joint range-velocity-azimuth estimation for OFDM-based integrated sensing and communication. *IEEE Transactions on Wireless Communications* 23, 10 (2024), 12933–12948.
- [12] Bo Huang, Jianan Li, Junjie Chen, Gang Wang, Jian Zhao, and Tingfa Xu. 2023. Anti-UAV410: A thermal infrared benchmark and customized scheme for tracking drones in the wild. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 46, 5 (2023), 2852–2865.
- [13] Huawei. 2024. *Intelligence + 5G-A: Elevating Connectivity Beyond Boundaries*. Technical Report. <https://www-file.huawei.com/admin/asset/v1/pro/view/d862bd7c8ce943d09e48dd56acd4cac5.pdf>
- [14] Huawei. 2025. Cloud Core Network. [Online]. <https://carrier.huawei.com/en/products/core-network>.
- [15] Tatsuya Iizuka, Takuya Sasatani, Toru Nakamura, Naoko Kosaka, Masaki Hisada, and Yoshihiro Kawahara. 2023. Millisign: mmwave-based passive signs for guiding uavs in poor visibility conditions. In *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking*. 1–15.
- [16] Intel. 2021. i5-11500. [Online]. <https://www.intel.com/content/www/us/en/products/sku/21227/intel-core-i511500-processor-12m-cache-up-to-4-60-ghz/specifications.html>.
- [17] Anil Jain, Karthik Nandakumar, and Arun Ross. 2005. Score normalization in multimodal biometric systems. *Pattern recognition* 38, 12 (2005), 2270–2285.
- [18] Jiamo Jiang, Mingfeng Xu, Zhongyuan Zhao, Kaifeng Han, Yang Li, Ying Du, and Zhiqin Wang. 2022. Rethinking the performance of isac system: from efficiency and utility perspectives. In *Proceedings of the 1st ACM MobiCom Workshop on Integrated Sensing and Communications Systems*. 19–24.
- [19] Stephen M Kay. 1998. Fundamentals of statistical signal processing: detection theory Prentice-Hall PTR. *Upper Saddle River, New Jersey* (1998).
- [20] Jens Klare, Oliver Biallawons, and Delphine Cerutti-Maori. 2017. UAV detection with MIMO radar. In *2017 18th International Radar Symposium (IRS)*. IEEE, 1–8.
- [21] Harold W Kuhn. 1955. The Hungarian method for the assignment problem. *Naval research logistics quarterly* 2, 1-2 (1955), 83–97.
- [22] Ruotong Li, Qixun Zhang, Dingyou Ma, Kan Yu, and Yuzhen Huang. 2025. Joint Target Assignment and Resource Allocation for Multi-Base Station Cooperative ISAC in UAV Detection. *IEEE Transactions on Vehicular Technology* (2025).
- [23] Xu Lin, Zhaopeng Du, Lin Mei, and Ruoyu Zhang. 2024. Integrated Sensing and Communications: A Survey of Recent Progress Toward 5G-A and 6G. In *International Conference on Wireless and Satellite Systems*. Springer, 245–262.
- [24] Binyue Liu, Jianqiang Yang, Bo Xu, Bolei Wang, and Hua Cai. 2025. Key Technologies for Low-Altitude Sensing in 5G-A Integrated Communication and Sensing Networks. *Journal of Signal Processing* 41, 5 (2025), 787–806.
- [25] Haotian Liu, Zhiqing Wei, Jinghui Piao, Huici Wu, Xingwang Li, and Zhiyong Feng. 2025. Carrier aggregation enabled MIMO-OFDM integrated sensing and communication. *IEEE Transactions on Wireless Communications* (2025).
- [26] Bohao Lu, Zhiqing Wei, Huici Wu, Xinrui Zeng, Lin Wang, Xi Lu, Dongyang Mei, and Zhiyong Feng. 2024. Deep learning based multi-node ISAC 4D environmental reconstruction with uplink-downlink cooperation. *IEEE Internet of Things Journal* (2024).
- [27] Fangkai Lu, Zhen Huang, Yongjiang Li, Xiaofeng Song, Xutao Shi, and Yanfei Wang. 2024. Intergrated Sensing and Communications (ISAC) in 5G-Advanced for High-Precision Localization and Tracking of Vessels at Sea. In *2024 10th International Conference on Computer and Communications (ICCC)*. IEEE, 1288–1292.
- [28] Xi Lu, Zhiqing Wei, Ruizhong Xu, Lin Wang, Bohao Lu, and Jinghui Piao. 2024. Integrated sensing and communication enabled multiple base stations cooperative UAV detection. In *2024 IEEE International Conference on Communications Workshops (ICC Workshops)*. IEEE, 1882–1887.
- [29] Xuewen Luo, Qingfeng Lin, Ruoyu Zhang, Hsiao-Hwa Chen, Xingwei Wang, and Min Huang. 2025. ISAC—A Survey on Its Layered Architecture, Technologies, Standardizations, Prototypes and Testbeds. *IEEE Communications Surveys & Tutorials* (2025).
- [30] Nader Mohamed, Jameela Al-Jaroodi, Imad Jawhar, Ahmed Idries, and Farhan Mohammed. 2020. Unmanned aerial vehicles applications in future smart cities. *Technological forecasting and social change* 153 (2020), 119293.
- [31] Samaneh Motie, Hadi Zayyani, Mohammad Salman, and Mehdi Bekrani. 2024. Self UAV localization using multiple base stations based on TDoA measurements. *IEEE Wireless Communications Letters* 13, 9 (2024), 2432–2436.
- [32] Chang Nie, Zhiyang Ju, Zhifeng Sun, and Hui Zhang. 2023. 3D object detection and tracking based on lidar-camera fusion and IMM-UKF algorithm towards highway driving. *IEEE Transactions on Emerging Topics in Computational Intelligence* 7, 4 (2023), 1242–1252.
- [33] Pierre Poitevin, Michel Pelletier, and Patrick Lamontagne. 2017. Challenges in detecting UAS with radar. In *2017 International Carnahan Conference on Security Technology (ICCST)*. IEEE, 1–6.

- [34] Mahdi Rezaei, Mutsuhiko Terauchi, and Reinhard Klette. 2015. Robust vehicle detection and distance estimation under challenging lighting conditions. *IEEE transactions on intelligent transportation systems* 16, 5 (2015), 2723–2743.
- [35] Prajwalita Saikia, Keshav Singh, Wan-Jen Huang, and Trung Q Duong. 2024. Hybrid deep reinforcement learning for enhancing localization and communication efficiency in RIS-aided cooperative ISAC systems. *IEEE internet of things journal* 11, 18 (2024), 29494–29510.
- [36] Ulzhalgas Seidalyieva, Lyazzat Ilipbayeva, Kyrmyzy Taissariyeva, Nurzhigit Smailov, and Eric T Matson. 2023. Advances and challenges in drone detection and classification techniques: A state-of-the-art review. *Sensors* 24, 1 (2023), 125.
- [37] Rohit Sharma and Rajeev Arya. 2022. UAV based long range environment monitoring system with Industry 5.0 perspectives for smart city infrastructure. *Computers & Industrial Engineering* 168 (2022), 108066.
- [38] Richard S Sutton and Andrew G Barto. 1998. Reinforcement learning: an introduction MIT Press. Cambridge, MA 22447, 10 (1998).
- [39] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).
- [40] Bingshu Wang, Qiang Li, Qianchen Mao, Jinbao Wang, CL Philip Chen, Aihong Shangguan, and Haosu Zhang. 2024. A survey on vision-based anti unmanned aerial vehicles methods. *Drones* 8, 9 (2024), 518.
- [41] Chenxing Wang, Jiangmin Tian, Jiuwen Cao, and Xiaohong Wang. 2021. Deep learning-based UAV detection in pulse-Doppler radar. *IEEE Transactions on Geoscience and Remote Sensing* 60 (2021), 1–12.
- [42] Huafei Wang, Liangtian Wan, Mianxiong Dong, Kaoru Ota, and Xianpeng Wang. 2019. Assistant vehicle localization based on three collaborative base stations via SBL-based robust DOA estimation. *IEEE Internet of Things Journal* 6, 3 (2019), 5766–5777.
- [43] Zhiqing Wei, Fengyun Li, Haotian Liu, Xu Chen, Huici Wu, Kaifeng Han, and Zhiyong Feng. 2024. Multiple reference signals collaborative sensing for integrated sensing and communication system towards 5G-A and 6G. *IEEE Transactions on Vehicular Technology* 73, 10 (2024), 15185–15199.
- [44] Zhiqing Wei, Hanyang Qu, Yuan Wang, Xin Yuan, Huici Wu, Ying Du, Kaifeng Han, Ning Zhang, and Zhiyong Feng. 2023. Integrated sensing and communication signals toward 5G-A and 6G: A survey. *IEEE Internet of Things Journal* 10, 13 (2023), 11068–11092.
- [45] Tzu-Tsung Wong and Po-Yang Yeh. 2019. Reliable accuracy estimates from k-fold cross validation. *IEEE Transactions on Knowledge and Data Engineering* 32, 8 (2019), 1586–1594.
- [46] Guangyu Wu, Fuhui Zhou, Chengzhen Meng, and Xiang-Yang Li. 2023. Precise UAV MMW-vision positioning: A modal-oriented self-tuning fusion framework. *IEEE Journal on Selected Areas in Communications* 42, 1 (2023), 6–20.
- [47] Guangyu Wu, Fuhui Zhou, Kai Kit Wong, and Xiang-Yang Li. 2024. A vehicle-mounted radar-vision system for precisely positioning clustering uavs. *IEEE Journal on Selected Areas in Communications* 42, 10 (2024), 2688–2703.
- [48] Qingqing Wu, Jie Xu, Yong Zeng, Derrick Wing Kwan Ng, Naofal Al-Dhahir, Robert Schober, and A Lee Swindlehurst. 2021. A comprehensive overview on 5G-and-beyond networks with UAVs: From communications to sensing and intelligence. *IEEE Journal on Selected Areas in Communications* 39, 10 (2021), 2912–2945.
- [49] Ruming Yang, Xingkang Li, Yongming Huang, Luxi Yang, and Wei Zhang. 2025. Hierarchical Reinforcement Learning-Based Beam Selection for Integrated Sensing and Communication Systems. *IEEE Transactions on Wireless Communications* (2025).
- [50] Jia Zhang, Xin Na, Rui Xi, Yimiao Sun, and Yuan He. 2023. mmHawk-eye: Passive UAV detection with a COTS mmWave radar. In *2023 20th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*. IEEE, 267–275.
- [51] Jie Zhao, Jingshu Zhang, Dongdong Li, and Dong Wang. 2022. Vision-based anti-uav detection and tracking. *IEEE Transactions on Intelligent Transportation Systems* 23, 12 (2022), 25323–25334.
- [52] Na Zhao, Qing Chang, Xiao Shen, Yunlong Wang, and Yuan Shen. 2024. Joint Target Localization and Data Detection in Bistatic ISAC Networks. *IEEE Transactions on Communications* (2024).
- [53] Tianya Zhao, Ningning Wang, Shiwen Mao, and Xuyu Wang. 2024. Few-shot learning and data augmentation for cross-domain uav fingerprinting. In *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*. 2389–2394.
- [54] Lei Zhou, Jisheng Dai, Weichao Xu, and Chunqi Chang. 2024. Joint Target Detection and Channel Estimation for Distributed Massive MIMO ISAC Systems. *IEEE Transactions on Cognitive Communications and Networking* (2024).
- [55] Yuekuan Zhou. 2025. Unmanned aerial vehicles based low-altitude economy with lifecycle techno-economic-environmental analysis for sustainable and smart cities. *Journal of Cleaner Production* (2025), 145050.

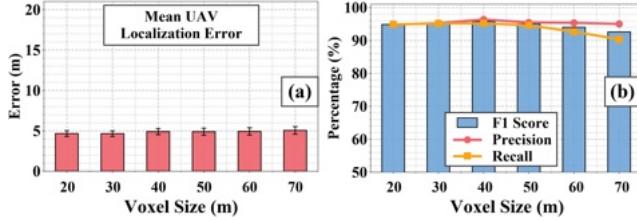
## A Limitations of Purely Learning-Based Fake Trajectory Filtering



**Figure 21: Visual comparison:** (a) results after point clustering and object tracking; (b) the same results after additional ConvTimeNet-based fake trajectory filtering.

Before conducting this work, we explored whether learning-based methods could directly eliminate all false trajectories after conventional point clustering and object tracking. Since trajectories are inherently sequential data, we adopted ConvTimeNet [4], a state-of-the-art time-series classification model, to distinguish true UAV trajectories from fake ones. Using its default training configuration, we conducted 5-fold cross-validation [45], which yielded an F1 score of only 75.59% (precision 74.32%, recall 76.90%). Figure 21 provides a visual comparison. Although ConvTimeNet reduces the number of fake trajectories to some extent, many false trajectories still persist, and parts of the true UAV trajectory are incorrectly removed. These results reveal that even advanced time-series models struggle to suppress the large number of long-term fake trajectories. This highlights the necessity of our layered approach: by leveraging point-level signal modeling and object-level motion consistency in addition to trajectory analysis, our system can achieve far more reliable UAV detection and tracking.

## B Voxel Size for Space Partitioning



**Figure 22: UAV tracking performance across different voxel sizes: (a) mean localization error; (b) F1 score.**

We partition the 3D space into voxels and model the distribution of noise points within each voxel using a multivariate

Gaussian, as described in Section 2.3.2. A large voxel size yields stable distribution estimates by aggregating sufficient points, whereas a small voxel size captures fine-grained spatial variations with higher spatial resolution. Thus, selecting an appropriate voxel size is crucial to balance spatial resolution and statistical stability. Figure 22 illustrates UAV tracking performance under different voxel sizes. As shown in Figure 22(a), the mean localization error consistently remains below 6 m, while Figure 22(b) demonstrates that a voxel size of 40 m yields the best tracking performance. Accordingly, we adopt a voxel size of 40 m, where the system effectively filters noise and reliably detects UAVs, achieving an F1 score of 95.77%.