

Needle in a Haystack: Tracking UAVs from Massive Noise in Real-World 5G-A Base Station Data

Chengzhen Meng^{†*}, Chenming He^{†*}, Yidong Jiang[†], Xiaoran Fan[‡], Dequan Wang[†],
Lingyu Wang[†], Jianmin Ji[†], Yanyong Zhang^{†◊}

[†]University of Science and Technology of China, [‡]Independent Researcher,

[◊]Institute of Artificial Intelligence, Hefei Comprehensive National Science Center

Abstract

The potential usage of UAVs in daily life has made monitoring them essential. However, existing systems for monitoring UAVs typically rely on cameras, LiDARs, or radars, whose limited sensing range or high deployment cost hinder large-scale adoption. In response, we develop BSense, the first system that tracks UAVs by leveraging point clouds from commercial 5G-A base stations. The key challenge lies in the dominant number of noise points that closely resemble true UAV points, resulting in a noise-to-UAV ratio over 100:1. Therefore, identifying UAVs from the raw point clouds is like finding a needle in a haystack. To overcome this, we propose a layered framework that filters noise at the point, object, and trajectory levels. At the raw point level, we observe that noise points from different spatial regions exhibit distinguishable and consistent signal fingerprints, which we can model to identify and remove them. At the object level, we design spatial and velocity consistency checks to identify false objects, and further compute confidence scores by aggregating these checks over multiple frames for more reliable discrimination. At the final trajectory level, we propose a Transformer-based network that captures multi-frame motion patterns to filter the few remaining false trajectories.

We evaluated BSense on a commercial 5G-A 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 95.56% 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 [2, 63, 65]. They are also emerging as integral components of smart city infrastructure, playing vital roles in traffic monitoring, emergency

response, and environmental surveillance [4, 37, 45]. Consequently, accurate wide-area tracking of UAVs is essential for these applications, ensuring safe flight navigation and reliable airspace intrusion detection.

Prior studies have explored the use of cameras, LiDARs, and radars for UAV perception [1, 48, 49, 60, 61]. However, cameras suffer from limited sensing range [28] and high sensitivity to lighting conditions [42], while LiDARs and radars are costly and impractical for large-scale deployment [40]. 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 [3, 31, 35, 53]. In addition to providing communication services, 5G-A base stations also offer sensing abilities [29, 54]. This enables low-cost, wide-area sensing of UAVs by leveraging existing base station infrastructure without requiring additional sensing equipment [58].

There have been previous studies [10, 27, 34, 38, 59, 64] that explore UAV tracking 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 tracking using point clouds from 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 challenge that emerge in such real-world deployment: *massive and persistent false points which closely resemble true UAV points*.

To substantiate this claim, Figure 1 shows a single frame from our dataset. The data were collected on a Huawei 5G-A base station [19, 20] deployed in Shanghai, which generates 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) [34, 59], the COTS base station we used provides a sensing range of up to 1,000 m. However, at such long ranges, UAV reflections become extremely weak and are difficult to

*Equal contribution.

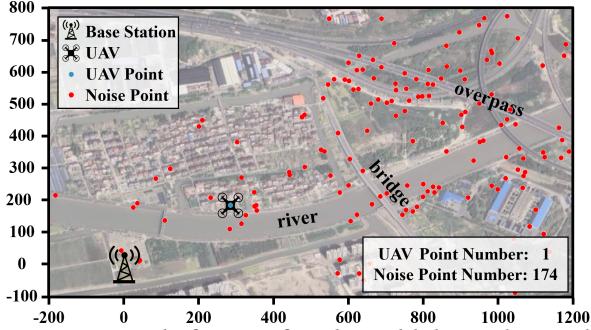


Figure 1: A single frame of real-world data. The number of noise points exceeds UAV points by a ratio of 174:1.

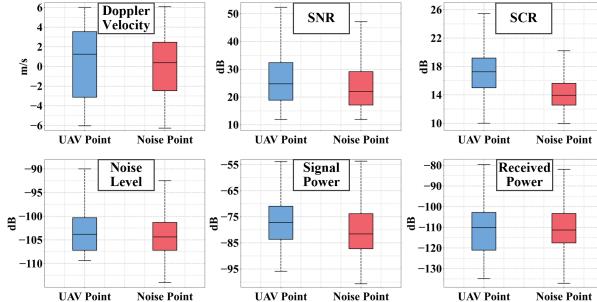


Figure 2: The value ranges of point features for UAV and noise exhibit substantial overlap.

distinguish from noise. Thus, identifying UAVs from raw point clouds is like finding a needle in a haystack.

Although signal-level noise suppression is applied before point cloud generation, a substantial number of noise points still persist. In this frame (Figure 1), 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 stationary structures (e.g., buildings and trees). Notably, small physical vibrations and Doppler leakage can make their returns exhibit non-zero velocities and thus difficult to filter [30]. (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 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 point cloud data from the COTS base station, 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.

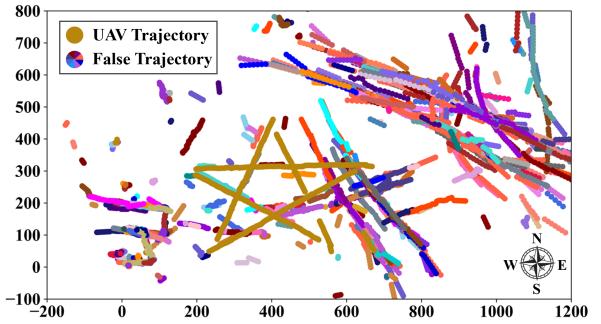


Figure 3: Persistent noise points give rise to massive long-term false trajectories.

This strong similarity eliminates any clear decision boundary, rendering traditional threshold-based filtering methods ineffective and leaving noise points difficult to distinguish from true UAV points.

- *Long-term False 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 raw points and visualize the trajectories in different colors. The true UAV trajectory appears as a dark gold \star , yet massive false trajectories also arise, some extending for hundreds of meters. Critically, these false trajectories often occur in plausible UAV locations, making accurate discrimination particularly challenging.

A straightforward approach is to use deep learning networks to classify and eliminate noise points and false trajectories. However, due to the overwhelming number of false points and their high similarity to true ones, our experiments in Section 3.2 show that even state-of-the-art learning methods fail to effectively eliminate the noise.

To overcome these challenges, we propose BSense, which achieves wide-area detection and tracking of UAVs using a COTS 5G-A base station. BSense employs a layered framework that filters noise at the point, object, and trajectory levels while preserving true UAV targets. At the point level, our key insight is that the background noise is strongly correlated with the surrounding environment and thus exhibits statistical patterns within local spatial regions. After partitioning the 3D sensing space into cubes, we observe that the noise points within each cube form a distinctive “fingerprint” consisting of multiple signal metrics. We therefore model each cube's noise fingerprint as a Gaussian distribution and estimate its mean vector and covariance matrix. Raw points are then evaluated by their similarity to the corresponding fingerprint, and those with a high degree of similarity are filtered out. Notably, only 10 minutes of data are sufficient for modeling, and due to the slow temporal variation, the model needs to be updated just once per day.

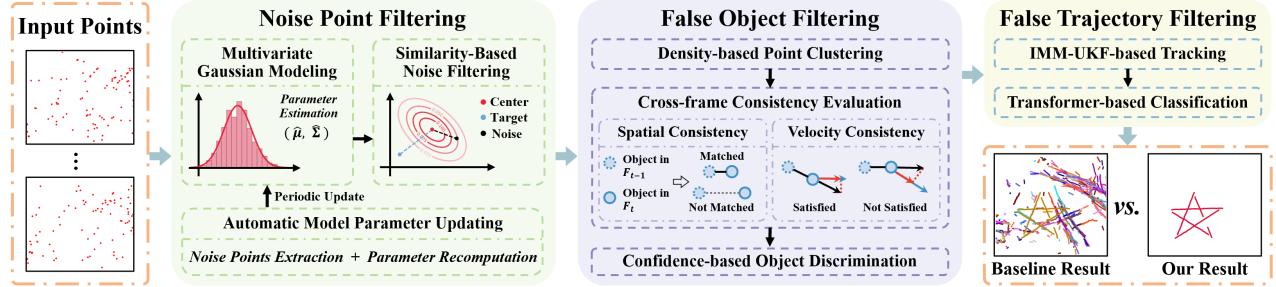


Figure 4: BSense adopts a layered framework consisting of (1) noise point filtering, (2) false object filtering, and (3) false trajectory filtering, which progressively suppress false detections to yield clean UAV trajectories.

After point-level filtering, we cluster the points into individual objects and further eliminate false objects. Specifically, we check object consistency between two consecutive frames based on intuitive motion principles. (1) *Spatial consistency*: True UAVs exhibit continuous spatial motion. Thus, a true object should have a corresponding object within a spatial neighborhood in the previous frame; if no such object exists, it is regarded as a false object. (2) *Velocity consistency*: A true object’s Doppler velocity measured by the base station should align with the radial velocity computed from the inter-frame spatial displacement; otherwise, it is regarded as a false object. Meanwhile, checks based on only two consecutive frames can be unreliable due to measurement errors. We thus aggregate these binary outcomes over time using an exponentially decaying weighting strategy, yielding spatial- and velocity-based confidence scores that emphasize more recent observations. This design produces robust indicators of object validity, allowing the system to reliably filter out objects that lack stable spatial or velocity behavior.

Finally, objects are tracked into individual trajectories; however, some false ones may still persist. To address this, we propose *TrajFormer*, a lightweight trajectory classification network inspired by the Transformer architecture [47], which leverages spatial motion and signal features across multiple frames to filter false 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 false ones.

In summary, we make the following contributions:

- We propose BSense, the first system that achieves accurate wide-area UAV detection and tracking using a commercial 5G-A base station in a real-world deployment. A demonstration video is available at <https://youtu.be/Bx668ORkAq4>.
- 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 fingerprint modeling method to remove points similar to the noise. Next, we design spatial and velocity consistency checks to distinguish true UAV targets from false objects. Finally, we employ

a lightweight classification Transformer-based network to eliminate the residual false 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, ∞ , \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 95.56%, with 96.73% precision, 94.41% 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. Our system takes as input 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 fingerprint, filtering points that are similar to the noise fingerprint. At the object level, the module clusters points into objects, performs spatial and velocity consistency checks, and computes confidence scores by aggregating these checks over multiple frames to distinguish true targets from

false objects. At the trajectory level, the module tracks objects and employs a lightweight Transformer-based classification network to eliminate false trajectories.

2.3 Noise Point Filtering through Noise Fingerprint 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 stationary structures such as buildings, terrain, and trees. However, these points may exhibit high signal strength and non-zero Doppler velocities [30], making them difficult to directly distinguish from UAV points based on raw point features.

To address this, we shift our focus from individual point-level features to the statistical patterns they form within local spatial regions. Our key idea is that background noise arises from echo signals reflected by the surrounding environment. Therefore, the signal metrics of noise points within each local region should follow a characteristic and consistent distribution, which we refer to as the **noise fingerprint**. Building on this insight, we first analyze and confirm that the noise fingerprint can be effectively characterized by a Gaussian distribution. We then estimate the distribution parameters across multiple frames and filter points based on their statistical deviation from the fitted distribution.

2.3.1 Analysis of Noise Fingerprint. We find that the signal metrics of noise points within localized spatial regions can be effectively modeled as a Gaussian distribution. This conclusion is supported by both theoretical reasoning and empirical evidence, as detailed below.

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 cubes with an edge length of 40 m, each containing a localized set of points. For each cube, we apply the Henze-Zirkler test [16] 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

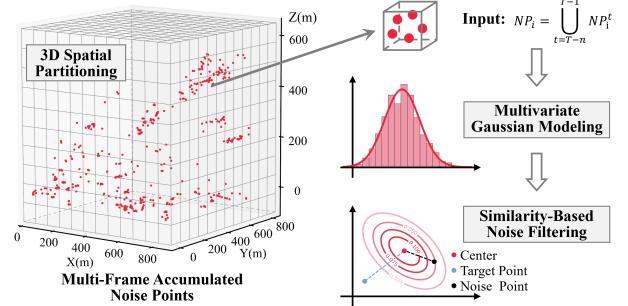


Figure 5: Noise point filtering through fingerprint modeling. The 3D space is partitioned, with noise in each cube modeled by a multivariate Gaussian; points are filtered based on their similarity to the corresponding distribution.

75% of cubes satisfy the Gaussianity assumption, confirming the feasibility of statistical modeling in our context. For the remaining cubes that fail the test, we observe that they often contain a higher proportion of complex ghost points, likely caused by multi-path effects or sidelobe interference, which leads to non-Gaussian signal patterns. These cubes are excluded from the modeling process, and the ghost points they contain are subsequently handled and filtered out by BSense's object-level and trajectory-level modules.

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 cubes (each with an edge length of 40 m) and model the noise distribution within each cube using a multivariate Gaussian. Due to space limitations, the rationale behind the cube size selection is discussed in Appendix A. 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 cube i , we define NP_i as the subset of noise points located within it: $NP_i = \{p \in NP \mid p \in \text{cube}_i\}$.

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 [23]. For cube i , we calculate the mean and standard deviation across all noise points:

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

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

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

Next, we estimate the parameters of noise distribution in cube i using Maximum Likelihood Estimation (MLE). For a

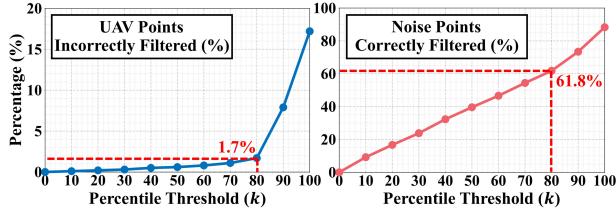


Figure 6: Percentage of filtered UAV points and noise points at different k values.

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:

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

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

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

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 [7]. 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 cube i , standardize its feature vector, and compute the Mahalanobis distance as:

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

where $z(p)$ 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 cube i , respectively.

We then classify a point as noise and remove it if its Mahalanobis distance falls below a predefined similarity threshold τ_{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_{upd} . Our experiments show that updating the model once per day is sufficient, and each update requires less than 900 frames (approximately 10 minutes) of data. The choice of optimal values for T and N will be discussed in Section 3.3. However, P_{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_{upd} . The remaining subset, denoted as NP_{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 False Object Filtering Using Spatial- and Velocity-based Confidences

After leveraging intrinsic point-level signal metrics, we further analyze the cross-frame motion patterns to eliminate false objects arising from multi-path effects and sidelobe interference. Our key idea is to perform a consistency check between consecutive frames based on intuitive motion principles. (1) *Spatial consistency*: A true object exhibits continuous spatial motion, so a corresponding object should appear within a local spatial neighborhood in the previous frame. (2) *Velocity consistency*: A true object's Doppler velocity measured by the base station should align with the radial velocity computed from the inter-frame spatial displacement.

Building on these consistency principles, we leverage the motion-pattern differences they reveal to eliminate false objects. First, we cluster points into individual objects. Then, we compute robust spatial- and velocity-based confidences by aggregating binary consistency outcomes over time using an exponentially decaying weighting strategy inspired by eligibility traces in reinforcement learning [46]. Finally, objects with low confidence scores are filtered out.

2.4.1 Point Clustering. Since motion characteristics are exhibited by the object as a whole, we first extend the DBSCAN

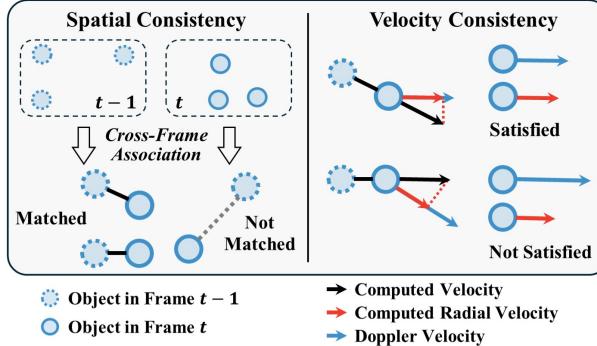


Figure 7: Spatial consistency is satisfied when an object is successfully matched across frames. **Velocity consistency** is satisfied when the radial velocity computed from spatial displacement is consistent with the Doppler velocity.

algorithm [11] to cluster points into individual objects more accurately. A cluster is formed when a point has sufficient neighboring points within a distance threshold τ_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 $\tau_d = 10$. Given the small size of UAVs, 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 true 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 [26] 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 [8] 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 [9]), 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.

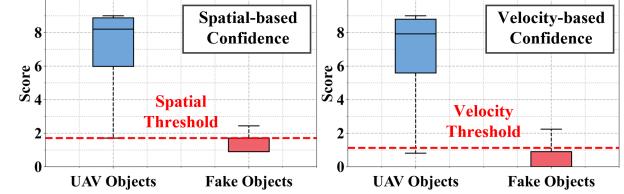


Figure 8: Statistical ranges of spatial- and velocity-based confidences. Thresholds are set at the 5th percentile of the UAV confidence scores, with $\tau_s = 1.7$ and $\tau_v = 1.1$.

Since true 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, true 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 γ_v is also empirically set to 0.9 following [9].

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 true 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 false 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 τ_s and τ_v are critical parameters that must be carefully chosen to balance the suppression of false objects with the retention of true targets.

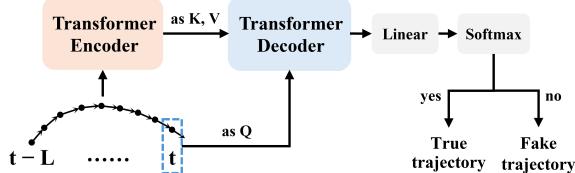


Figure 9: The overview of our Transformer-based trajectory classification network.

To determine appropriate thresholds, we analyze the confidence score distributions of both UAVs and false objects. As shown in Figure 8, both of the proposed confidences provide clear separation between UAVs and false 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 false objects.

Key Result: After this module, the average number of false 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 False Trajectory Filtering

After obtaining the object results, we track their positions over time while simultaneously classifying the generated trajectories. Notably, the vast majority of false detections have already been removed by the point-level and object-level filtering methods. We therefore propose a lightweight trajectory classification network that operates in parallel with tracking, continuously classifying and filtering out the few 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) [39], 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 10: The 5G-A base station Active Antenna Unit (AAU) and the UAV deployed in our field experiments.

with obj across L consecutive frames, where we set $L = 6$, a value determined through experiments in Section 3.5, balancing accuracy and output latency.

2.5.2 Transformer-based Trajectory Classification. During tracking, false trajectories may still emerge. To address this, we propose TrajFormer, a lightweight Transformer-based trajectory classification 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 [47] 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 [47], 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 false objects per frame decreases from **2.58** to **0.04**, and BSense achieves an **F1 score of 95.56%**, as detailed in Section 3.5.

3 EVALUATION

3.1 Experimental Setup

System Setup. We collected data using a Huawei COTS 5G-A base station [19, 20] operating at 4.9 GHz with a 100 MHz 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 [8] used in our field trials. The AAU was positioned at a height of 23 m, providing

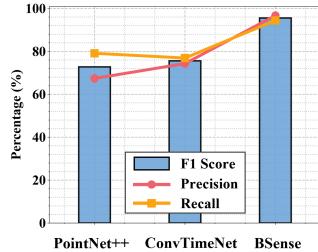


Figure 11: Performance comparison with baselines.

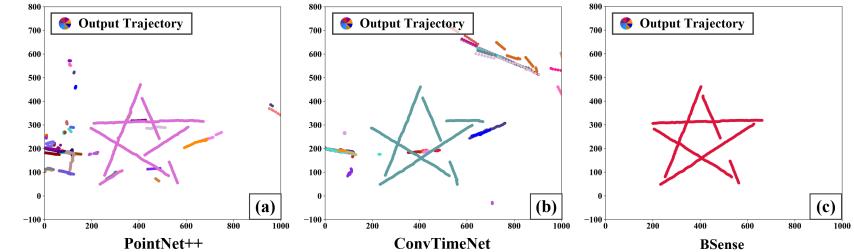


Figure 12: Visual comparison with baselines. Compared with (a) and (b), our system (c) filters false trajectories and achieves accurate UAV detection and tracking.

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

Data Acquisition. The UAV was instructed to follow a set of shaped flight paths (\circlearrowleft , S , ∞ , \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° , -75° , ..., 75° , and 85° . 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 fingerprint 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.

Train-Test Split. Since the TrajFormer introduced in Section 2.5.2 requires training, evaluation was performed using 5-fold cross-validation [55]. 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.

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 noise fingerprint 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 10 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 false 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 10 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 false objects per frame (N_f) is defined as $N_f = \frac{FP}{\text{total frames}}$.

- **Trajectory-Level Metrics:** Finally, using the object tracking and false 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_{loc} is calculated as $E_{\text{loc}} = \frac{1}{N} \sum_{i=1}^N \|\hat{\mathbf{pos}}_i - \mathbf{pos}_i\|_2$, 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

Comparison with Baselines. Since no existing method targets UAV tracking using point clouds from COTS 5G-A base stations, a direct comparison is not feasible. We therefore compare BSense with two representative baselines, which filter noise at the point and trajectory levels, respectively.

- (1) **PointNet++** [41]: A widely used network for classifying radar points [12, 25]. It learns both local and global features from raw point clouds to distinguish and remove false points. The remaining points are then used for clustering and object tracking.
- (2) **ConvTimeNet** [5]: A state-of-the-art time-series classification model. Since trajectories can naturally be treated as time-series data, we first perform point clustering and object tracking, and then apply ConvTimeNet to distinguish true UAV trajectories from false ones.

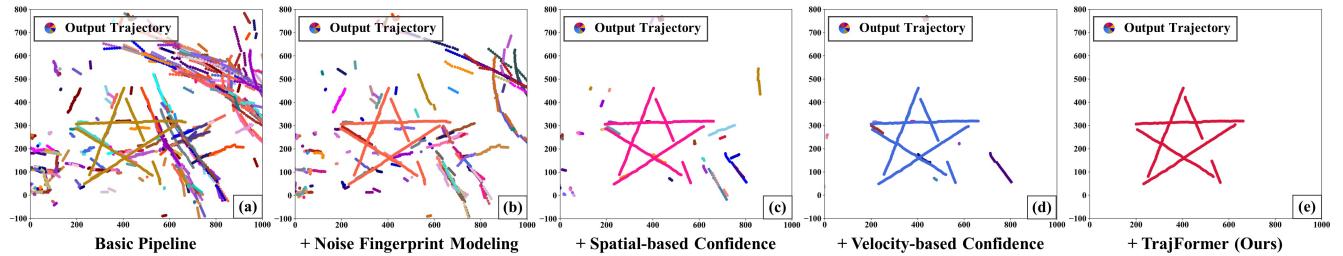


Figure 13: Trajectory visualizations in the ablation study. From left to right, the symbol “+” indicates that the methods are constructed through the incremental addition of our proposed modules.

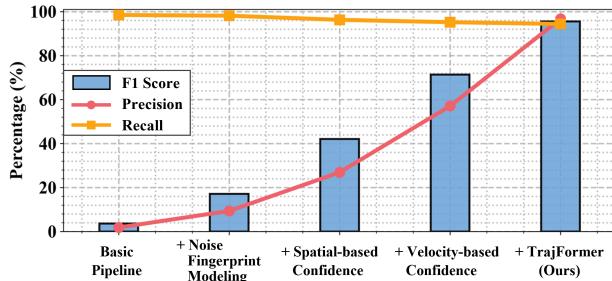


Figure 14: Ablation study. The “+” indicates stepwise integration of our proposed modules.

As shown in Figure 11, BSense achieves the highest F1 score of 95.56%, which significantly outperforms PointNet++ (72.78%) and ConvTimeNet (75.59%). Figure 12 further presents a visual comparison in the \star -trajectory case, where our system effectively removes false trajectories and achieves accurate UAV tracking, outperforming both baselines.

Ablation Study. We perform an ablation study by incrementally adding our modules to the basic pipeline and evaluating their contributions on the full dataset:

- (1) *Basic Pipeline*: The raw points are directly clustered with DBSCAN [11] and the resulting objects are tracked with an IMM-UKF [39].
- (2) *+ Noise Fingerprint Modeling*: Building on *method (1)*, we incorporate noise point filtering based on noise fingerprint modeling, as described in Section 2.3.
- (3) *+ Spatial-based Confidence*: Extending *method (2)*, we further introduce spatial-based confidence for filtering false objects, as discussed in Section 2.4.
- (4) *+ Velocity-based Confidence*: Extending *method (3)*, we further introduce velocity-based confidence for filtering false objects, as discussed in Section 2.4.
- (5) *+ TrajFormer (Ours)*: Finally, building on *method (4)*, we employ the proposed TrajFormer network to filter false trajectories, as detailed in Section 2.5.

Notably, methods (1)–(5) demonstrate the stepwise integration of our proposed modules, with the corresponding results shown in Figure 14. The basic pipeline performs poorly, achieving an F1 score of only 3.65% due to numerous noise-induced false positives, which drive the precision down to

1.86%. Building on this basic pipeline, we progressively incorporate noise fingerprint modeling, spatial-based confidence, velocity-based confidence, and TrajFormer. Each module further suppresses false positives, ultimately increasing the F1 score from 3.65% to 95.56%. This progression clearly demonstrates the effectiveness of each module in our system.

In addition, Figures 13 (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, false trajectories are progressively eliminated, while the UAV trajectory is fully preserved. These visualizations further demonstrate the effectiveness of our system in achieving accurate UAV detection and tracking.

Visualization Results. The system’s per-frame outputs are saved and visualized. Figure 15 presents a comparison between the basic pipeline and our BSense across different flight paths (–, \circ , S, ∞ , \square , \diamond , M, \star). The first and third rows show the basic pipeline results, while the second and fourth rows display the corresponding outputs of BSense. As illustrated, compared with the basic pipeline, BSense effectively eliminates false trajectories while accurately detecting and tracking the UAV. For the ∞ , 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).

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 [22]. The results show that the three sequential stages—noise point filtering, false object filtering, and false 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.

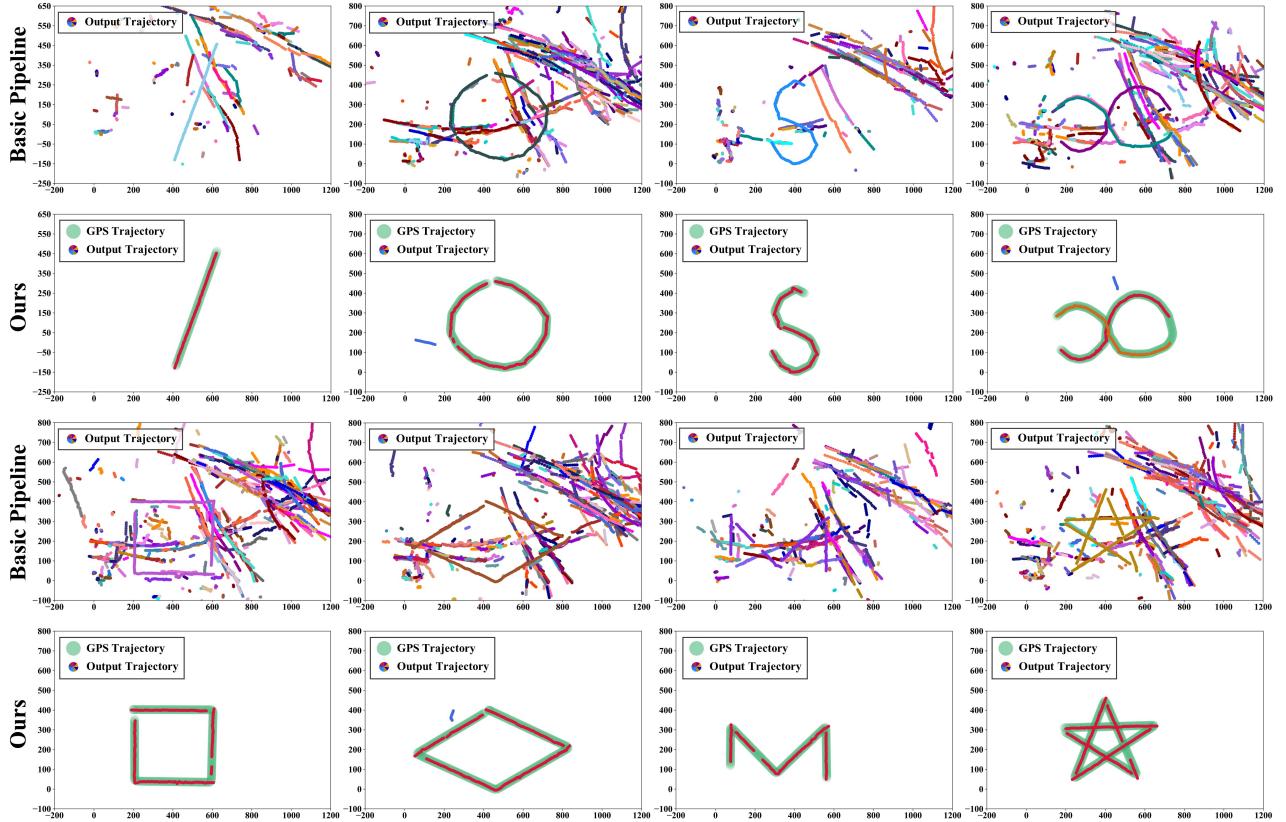


Figure 15: Visual comparison of the basic pipeline and our BSense across different flight paths. Our system effectively eliminates false trajectories and achieves accurate UAV detection and tracking.

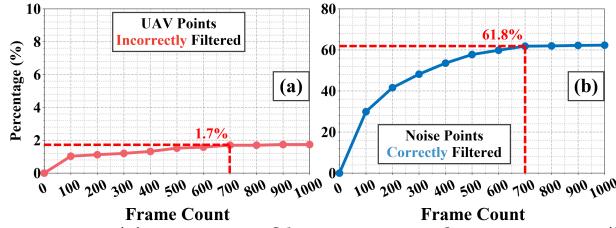


Figure 16: (a) Incorrect filtering rate of UAV points; (b) Correct filtering rate of noise points under different data frame counts for noise fingerprint modeling.

3.3 Performance of Noise Point Filtering

Data Frame Count for Noise Fingerprint Modeling. The system adaptively models the noise fingerprint 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 16 presents the filtering performance under different values of N . As shown in Figure 16(b), the correct filtering rate of noise points improves with larger N , but the incremental gains diminish substantially once N exceeds 700 frames (7.5 minutes). Figure 16(a) shows that the incorrect filtering rate of UAV points consistently remains below 2%. Considering both filtering effectiveness and storage overhead, we set $N =$

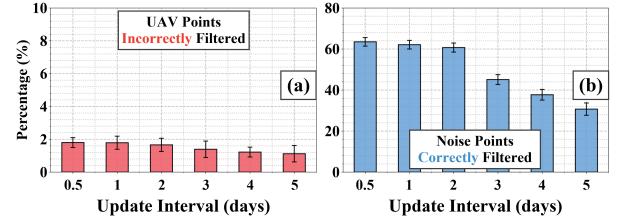


Figure 17: (a) Incorrect filtering rate of UAV points; (b) Correct filtering rate of noise points under different update intervals for model parameter updates.

700 for noise fingerprint modeling and parameter updates. Under this configuration, our noise fingerprint 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.

Model Parameter Update Interval. To ensure the robustness of noise point cloud filtering over time, the system periodically updates the parameters of the noise fingerprint model with an interval of T , as described in Section 2.3.4. Figure 17 reports the filtering performance under different values of T . As shown in Figure 17(b), the correct filtering rate of noise points decreases with increasing T , with a pronounced drop once T exceeds 2. Meanwhile, Figure 17(a)

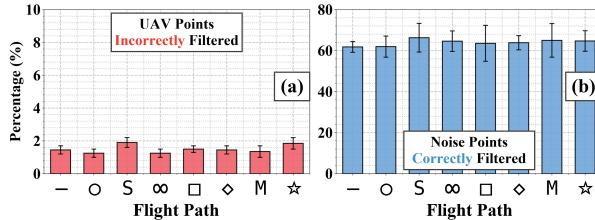


Figure 18: (a) Incorrect filtering rate of UAV points; (b) Correct filtering rate of noise points across different UAV flight paths.

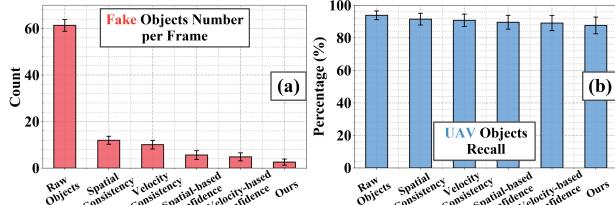


Figure 19: (a) Number of false objects per frame; (b) Recall of UAV objects under varying filtering methods.

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.

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 18 illustrates the performance of the proposed noise point filtering method across different UAV flight paths (–, ○, S, ∞, □, ◊, M, ☆). As shown in Figure 18(b), the correct filtering rate of noise points consistently exceeds 60%, while Figure 18(a) 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 False Object Filtering

Different Filtering Methods. We evaluate the effectiveness of different methods for filtering false objects:

- (1) *Raw Objects*: object results obtained directly from DBSCAN clustering [11].
- (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.

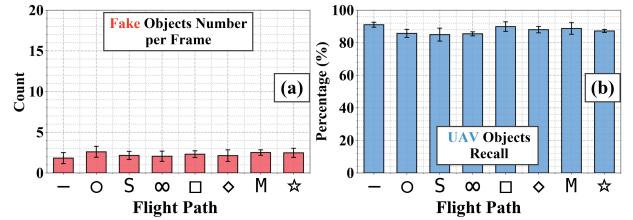


Figure 20: (a) Number of false objects per frame; (b) Recall of UAV objects across different UAV flight paths.

Figure 19 compares the performance of these methods. As shown in Figure 19(a), applying spatial consistency, velocity consistency, and the derived spatial- and velocity-based confidence significantly reduces the number of false objects. Meanwhile, Figure 19(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 false 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.

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 20 illustrates the performance of the proposed false object filtering method across different UAV flight paths (–, ○, S, ∞, □, ◊, M, ☆). As shown in Figure 20(a), the number of false objects per frame consistently remains below 3, while Figure 20(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

Trajectory Classification Window Size. The system determines whether a trajectory is a true target or a false 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 classification performance. However, increasing L also introduces output latency, as the system must accumulate more observations before making a decision. Figure 21 presents the tracking performance across different window sizes. Figure 21(b) shows that when L exceeds 6, the improvements in F1 score become marginal, while Figure 21(a) demonstrates that the mean UAV localization error consistently remains below 6 m, satisfying the requirement for kilometer-scale sensing. Therefore, we set $L = 6$ in our experiments. Under this configuration, the system achieves strong detection performance with an F1 score of 95.56% (precision 96.73%, recall 94.41%) and maintains a mean localization error of 4.9 m.

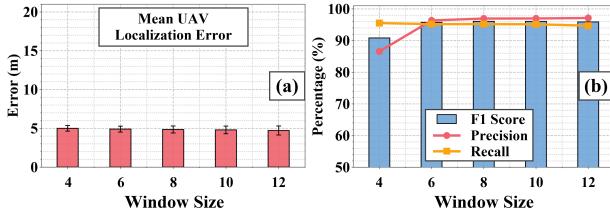


Figure 21: UAV tracking performance under varying window sizes: (a) mean localization error; (b) F1 score.

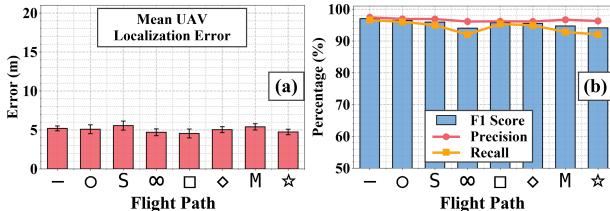


Figure 22: UAV tracking performance across different flight paths: (a) mean localization error; (b) F1 score.

Different UAV Flight Paths. For flight paths ranging from simple (–) to complex (\star), obtaining clean and reliable UAV trajectories is crucial. Figure 22 presents the tracking performance across different flight paths (–, \circ , S, ∞ , \square , \diamond , M, \star). Precision consistently exceeds 96% across all paths, highlighting the effectiveness of our system. However, the recall for the ∞ , M, and \star trajectories is comparatively lower, primarily due to the continuous absence of target raw points during certain periods (Figure 15). 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 observe that the UAV trajectories output by the system occasionally exhibit interruptions of 50–100 m, as illustrated in the ∞ and \star examples in Figure 15. These interruptions are not caused by algorithmic misses, but rather by prolonged target absences in the raw point cloud data. They typically occur when the UAV is occluded by tall urban buildings or when its flight direction remains tangential to the base station. In future work, we plan to leverage deep learning to extract motion features from both preceding and succeeding trajectory segments, enabling seamless trajectory completion by generating the missing portions of interrupted trajectories.

Limited Localization Accuracy. Our system achieves accurate UAV detection with a mean localization error of 4.9 m. Although meter-level accuracy at kilometer-scale sensing is

nontrivial, further improvement is needed for broader applications. This work primarily addresses the challenge of filtering massive false detections, while the remaining localization error mostly arises from measurement noise inherent in base-station point clouds; thus, enhancing localization accuracy is beyond our current scope. Future improvements may include increasing signal bandwidth or expanding the antenna array to boost ranging accuracy and angular resolution, thereby yielding higher-quality point clouds.

5 RELATED WORK

Sensor-based UAV Sensing. Cameras, LiDARs, and radars are widely used for object detection [6, 14, 15, 36, 50, 52], and recent studies have also explored their use in UAV sensing [1, 13, 18, 21, 24, 44, 48, 49, 56, 57, 60, 61]. Despite the strong perception performance achieved by these sensor-based approaches, they still suffer from several limitations. Camera-based systems struggle to obtain reliable distance information and therefore typically provide only 2D detection. They are also highly sensitive to lighting conditions [42], making nighttime operation difficult. LiDAR and radar systems, while more robust, tend to be expensive and thus unsuitable for large-scale deployment [40], limiting their use to specific scenarios such as airports or UAV shows.

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 [29, 35, 53, 54]. Recent studies [10, 17, 27, 32, 34, 38, 43, 51, 59, 62, 64] 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. [33], 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.

6 CONCLUSION

We present BSense, the first practical system that tracks UAVs using commercial 5G-A 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] Tasnim Azad Abir, Vu Le, Endrowednes Kuantama, Pranjol Sen Gupta, Austin Copley, Judith Dawes, Mohammad Islam, Richard Han, and Phuc Nguyen. 2025. Detection and Tracking of Drone Swarms using LiDAR. In *Proceedings of the 23rd Annual International Conference on Mobile Systems, Applications and Services*. 500–514.
- [2] Yunes Alqudsi and Murat Makaraci. 2025. UAV swarms: research, challenges, and future directions. *Journal of Engineering and Applied Science* 72, 1 (2025), 12.
- [3] 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.
- [4] 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.
- [5] Mingyue Cheng, Jiqian Yang, Tingyue 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.
- [6] Xiaomeng Chu, Jiajun Deng, Guoliang You, Yifan Duan, Houqiang Li, and Yanyong Zhang. 2025. RaCFormer: Towards High-Quality 3D Object Detection via Query-based Radar-Camera Fusion. In *Proceedings of the Computer Vision and Pattern Recognition Conference*. 17081–17091.
- [7] Roy De Maesschalck, Delphine Jouan-Rimbaud, and Désiré L Massart. 2000. The mahalanobis distance. *Chemometrics and intelligent laboratory systems* 50, 1 (2000), 1–18.
- [8] DJI. 2022. DJI Mavic 3 Enterprise Series. [Online]. <https://enterprise.dji.com/mavic-3-enterprise>.
- [9] Yifan Duan, Jie Peng, Yu Zhang, Jianmin Ji, and Yanyong Zhang. 2022. Pffilter: 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.
- [10] Mohamed Elfatoure, 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).
- [11] 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.
- [12] Thomas Griebel, Dominik Authaler, Markus Horn, Matti Henning, Michael Buchholz, and Klaus Dietmayer. 2021. Anomaly detection in radar data using PointNets. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2667–2673.
- [13] Marcus Hammer, Marcus Hebel, Björn Borgmann, Martin Laurenzis, and Michael Arens. 2018. Potential of lidar sensors for the detection of UAVs. In *Laser Radar Technology and Applications XXIII*, Vol. 10636. SPIE, 39–45.
- [14] Chenming He, Chengzhen Meng, Chunwang He, Xiaoran Fan, Beibei Wang, Yubo Yan, and Yanyong Zhang. 2024. See Through Vehicles: Fully Occluded Vehicle Detection with Millimeter Wave Radar. In *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*. 740–754.
- [15] Chenming He, Rui Xia, Chengzhen Meng, Xiaoran Fan, Dequan Wang, Haojie Ren, Jianmin Ji, and Yanyong Zhang. 2025. Ghost Points Matter: Far-Range Vehicle Detection with a Single mmWave Radar in Tunnel. In *Proceedings of the 31st Annual International Conference on Mobile Computing and Networking*. 650–666.
- [16] 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.
- [17] 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.
- [18] 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.
- [19] Huawei. 2024. *Intelligence + 5G-A: Elevating Connectivity Beyond Boundaries*. Technical Report. <https://www-file.huawei.com/admin/asset/v1/pro/view/d862bd7c8ce943d09e48dd56acd4cac5.pdf>
- [20] Huawei. 2025. Cloud Core Network. [Online]. <https://carrier.huawei.com/en/products/core-network>.
- [21] 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.
- [22] 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>.
- [23] Anil Jain, Karthik Nandakumar, and Arun Ross. 2005. Score normalization in multimodal biometric systems. *Pattern recognition* 38, 12 (2005), 2270–2285.
- [24] Jens Klare, Oliver Biallawons, and Delphine Cerutti-Maori. 2017. UAV detection with MIMO radar. In *2017 18th International Radar Symposium (IRS)*. IEEE, 1–8.
- [25] Johannes Kopp, Dominik Kellner, Aldi Piroli, and Klaus Dietmayer. 2023. Tackling Clutter in Radar Data-Label Generation and Detection Using PointNet++. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 1493–1499.
- [26] Harold W Kuhn. 1955. The Hungarian method for the assignment problem. *Naval research logistics quarterly* 2, 1-2 (1955), 83–97.
- [27] 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).
- [28] Yao Li, Yingjie Wang, Chengzhen Meng, Yifan Duan, Jianmin Ji, Yu Zhang, and Yanyong Zhang. 2024. FARFusion: A practical roadside radar-camera fusion system for far-range perception. *IEEE Robotics and Automation Letters* 9, 6 (2024), 5433–5440.
- [29] 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.
- [30] 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.
- [31] 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).
- [32] 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).

- [33] Fangkai Lu, Zhen Huang, Yongqiang Li, Xiaofeng Song, Xutao Shi, and Yanfei Wang. 2024. Intergrated Sensing and Communications (ISAC) in 5G-Advanced for High-Precison Localization and Tracking of Vessels at Sea. In *2024 10th International Conference on Computer and Communications (ICCC)*. IEEE, 1288–1292.
- [34] 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.
- [35] 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).
- [36] Chengzhen Meng, Chenming He, Dequan Wang, Yuxuan Xiao, Lingyu Wang, Xiaoran Fan, Lu Zhang, and Yanyong Zhang. 2025. Gr-fall: A fall detection system with gait recognition for indoor environments using siso mmwave radar. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 9, 3 (2025), 1–26.
- [37] 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.
- [38] 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.
- [39] 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.
- [40] 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.
- [41] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. 2017. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems* 30 (2017).
- [42] Mahdi Rezaei, Mutsuhiro 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.
- [43] 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.
- [44] Ulzhalgas Seidaliyeva, 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.
- [45] 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.
- [46] Richard S Sutton and Andrew G Barto. 1998. Reinforcement learning: an introduction MIT Press. *Cambridge, MA* 22447, 10 (1998).
- [47] 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).
- [48] 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.
- [49] 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.
- [50] Dequan Wang, Xinran Zhang, Kai Wang, Lingyu Wang, Xiaoran Fan, and Yanyong Zhang. 2024. Rdgait: A mmwave based gait user recognition system for complex indoor environments using single-chip radar. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 8, 3 (2024), 1–31.
- [51] Huafei Wang, Liangtian Wan, Mianxiong Dong, Kaoru Ota, and Xianapeng 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.
- [52] Yingjie Wang, Jiajun Deng, Yao Li, Jinshui Hu, Cong Liu, Yu Zhang, Jianmin Ji, Wanli Ouyang, and Yanyong Zhang. 2023. Bi-lrfusion: Bi-directional lidar-radar fusion for 3d dynamic object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 13394–13403.
- [53] 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.
- [54] 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.
- [55] 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.
- [56] 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.
- [57] 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.
- [58] 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.
- [59] 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).
- [60] 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.
- [61] 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.
- [62] 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).
- [63] 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.
- [64] 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).
- [65] 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 Cube Size for Space Partitioning

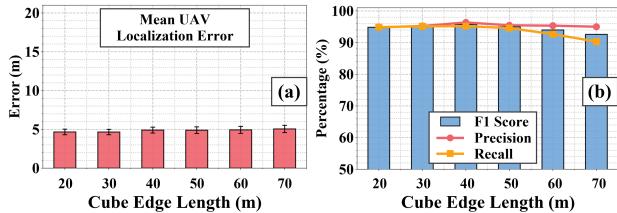


Figure 23: UAV tracking performance across different cube edge lengths.

We partition the 3D space into cubes and model the fingerprint of noise points within each cube using a multivariate

Gaussian, as described in Section 2.3.2. In theory, the cube size should not be set too large; otherwise, a single cube may encompass multiple reflectors with significantly different scattering properties, causing the statistical characteristics of the noise points to deviate from a Gaussian distribution.

Therefore, we vary the cube sizes and evaluate the resulting UAV tracking performance on our dataset, as illustrated in Figure 23. We find that when the cube edge length is kept within 50 m, the system performance remains stable, with the F1 score consistently above 95%. However, once the edge length exceeds 60 m, the performance begins to degrade, primarily reflected in a noticeable drop in recall.

This trend is expected: larger cubes are more likely to mix reflections from different types of surfaces, reducing the local statistical consistency required for reliable Gaussian modeling. As the fitted distribution becomes less representative, more true UAV points may be incorrectly filtered out, leading to reduced recall. In summary, we recommend keeping the cube edge length below 50 m when partitioning the space. In our system, we ultimately set it to 40 m.