AI sensors

Article

Deep Learning for Space Debris Tracking: One-Step Tracklet Filtering with a Hybrid GRU-CNN Architecture

Alessandro Cabras [1,*,†], Niccolò Pilloni [2,†], Victor Mustieles-Perez [3,4], Jan Siminski [5], Marco Alessandrini [5] and Davide Bacciu [6]

[1] National Institute for Astrophysics (INAF), Cagliari Astronomical Observatory, Via della Scienza 5, 09047 Selargius, Italy
[2] European Organisation for the Exploitation of Meteorological Satellites (EUMETSAT), Eumetsat Allee 1, 64295 Darmstadt, Germany; niccolo.pilloni4@gmail.com
[3] German Aerospace Center (DLR), Microwaves and Radar Institute, 82234 Weßling, Germany; victor.mustieles@dlr.de
[4] Institute of Microwaves and Photonics, Friedrich-Alexander-Universität (FAU), 91058 Erlangen, Germany
[5] European Space Operations Centre (ESOC), European Space Agency (ESA), Robert-Bosch-Straße 5, 64293 Darmstadt, Germany; jan.siminski@esa.int (J.S.); marco.alessandrini@esa.int (M.A.)
[6] Dipartimento di Informatica, University of Pisa, L.go B. Pontecorvo 3, 56127 Pisa, Italy; davide.bacciu@unipi.it
* Correspondence: alessandro.cabras@inaf.it
† These authors contributed equally to this work.

Abstract

The proliferation of space debris in Low Earth Orbit (LEO) poses a growing threat to operational satellites, requiring robust surveillance and tracking systems. Radar is a primary technology for monitoring these objects; however, standard tracking algorithms often degrade when measurements are sparse, corrupted by non-Gaussian noise, or available only as short tracklets. Traditional methods, such as the Unscented Kalman Filter (UKF), rely on explicit physical and statistical models that may struggle to converge under highly nonlinear dynamics, uncertain initialization, and non-ideal sensor perturbations. In this study, we propose a hybrid deep learning framework for learned one-step tracklet filtering of radar measurements. The architecture consists of a stateful Gated Recurrent Unit (GRU) layer followed by one-dimensional Convolutional Neural Network (1D-CNN) layers, complemented by variable-specific preprocessing strategies, including residual learning for range and relative pivoting for azimuth, to handle the scale disparities and heterogeneous behavior of radar observables. This design combines the ability of GRUs to model temporal dependencies with the effectiveness of CNNs in extracting local features for signal denoising. The method is validated on synthetically generated LEO trajectories with realistic orbital perturbations and tunable radar noise profiles, including Gaussian noise, impulsive spikes, transient degradation, and state-dependent perturbations. Compared with EKF- and UKF-based analytical baselines, the proposed model achieves lower filtering error and improved robustness under severe non-Gaussian disturbances. Stress testing further shows that the network can reject non-physical sensor anomalies without requiring long initialization warm-up phases, making it suitable for sparse short-tracklet processing in synthetic SST benchmark scenarios.

Keywords: one-step filtering; radar measurement denoising; tracklet filtering; deep learning; hybrid neural networks (GRU-CNN); space debris tracking; unscented kalman filter

Academic Editors: Chengkuo Lee and Zixuan Zhang

Received: 25 March 2026
Revised: 7 July 2026
Accepted: 10 July 2026
Published: 28 July 2026

1. Introduction

In recent decades, a large number of satellites have been launched for industrial and scientific applications. Depending on the service to be provided, satellites are placed in Low Earth Orbit (LEO), Medium Earth Orbit (MEO), and Geostationary Earth Orbit (GEO). In particular, scientific missions such as Earth Observation missions, which are indispensable to monitor climate change (glacial melting, alterations in biodiversity) [1], are mostly LEO. These missions are threatened by space debris, which significantly increased due to the escalation of space exploitation [2]. A reliable and efficient system for space debris detection and a methodology for trajectory prediction are therefore mandatory.

Space debris tracking is widely performed with space tracking and surveillance systems based on optical and radar sensors. While optical sensors can provide better position accuracy, radar can still achieve high-quality measurements regardless of the atmospheric conditions [3]. For this reason, radar technology has emerged as a prevailing instrument for tracking targets in LEO [4]. However, radar measurements in the context of space surveillance face significant challenges. As noted by Levanon [5], radar signals are inherently compromised by thermal noise and clutter. Furthermore, tracking LEO objects is complicated by nonlinear orbital dynamics, including atmospheric drag variations and gravitational perturbations (J_2 effects) [6]. Precise trajectory prediction and smoothing algorithms are therefore required to enhance the detections provided by the system and generate the corresponding trajectories, thus averting potential collisions between debris and satellites.

In modern operational scenarios, space situational awareness relies on highly specialized data acquisition equipment. State-of-the-art facilities, such as advanced active Space Surveillance and Tracking (SST) radars or space-debris laser-ranging stations—which employ sophisticated monostatic or bistatic optical configurations and active tracking control loops—exemplify the complex hardware required to capture precise observations of uncooperative LEO targets [7,8]. For example, Steindorfer et al. describe a bistatic MHz laser-ranging configuration in which the transmitting and receiving optical paths are physically separated, allowing the system to mitigate atmospheric backscatter while acquiring high-rate range measurements from both cooperative satellites and diffusely reflecting debris objects. Such examples show that the measurement process is strongly shaped by the physical architecture of the sensor, including aperture size, transmitter–receiver geometry, repetition rate, detector behavior, range gating, filtering, and illumination-dependent noise.

These hardware-dependent characteristics are directly relevant when designing simulation-based datasets for tracking and anomaly-detection algorithms. In a practical operational workflow, real acquisition systems provide the nominal observation geometry, sampling cadence, measurement variables, and uncertainty bounds, while high-fidelity simulation allows these conditions to be reproduced under controlled perturbations. This makes it possible to generate synthetic tracklets that preserve realistic sensor constraints while systematically injecting anomalies such as clutter, missed detections, biased measurements, hardware-induced spikes, or signal degradation. The combination of real equipment specifications and simulation therefore provides a controlled bridge between operational data acquisition and data-driven model development, enabling deep learning models to be trained and stress-tested under realistic but reproducible conditions before deployment on real SST observations.

Figure 1 provides a schematic and conceptual overview of this operational SST sensing chain and illustrates how real equipment specifications can be translated into simulation parameters for controlled tracklet generation and learned filtering.

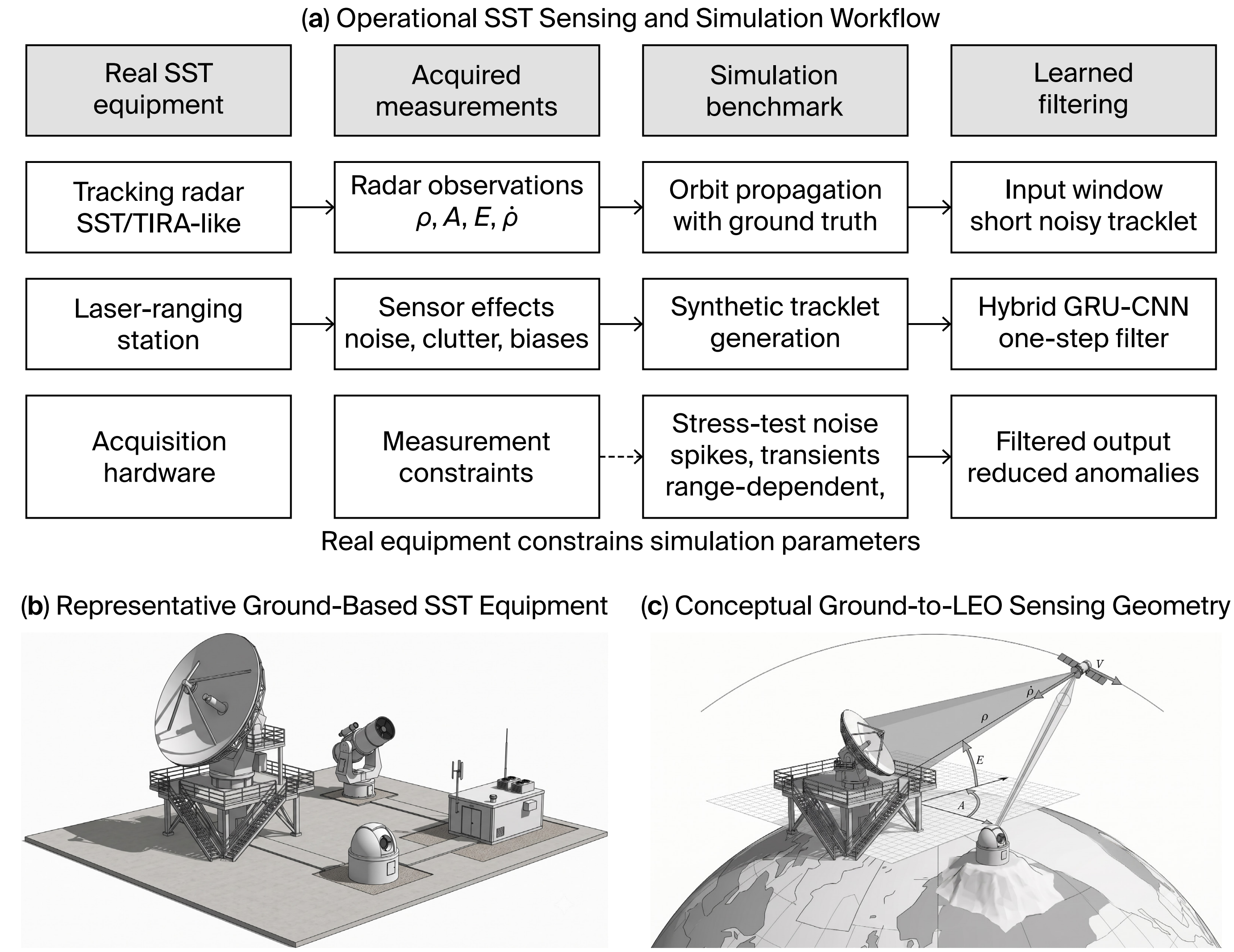

Figure 1. Overview of the operational Space Surveillance and Tracking (SST) context considered in this work. (**a**) Schematic workflow linking real SST equipment, acquired measurements, simulation-based benchmark generation, and the proposed GRU-CNN one-step filtering framework. (**b**) Conceptual 3D illustration of representative ground-based SST equipment, including radar, optical, and laser-ranging sensing assets. (**c**) Conceptual 3D ground-to-LEO sensing geometry, highlighting the line-of-sight observation process and the acquisition of radar-frame quantities such as range, azimuth, elevation, and range rate.

The Kalman filter is the most widely used method for correlation, filtering, and smoothing of the measurements to generate the target trajectories. The seminal works by Blackman and Popoli [9] and Bar-Shalom et al. [10] established the foundation for modern tracking systems. Other solutions include the alpha-beta filter, which is a simpler solution, or Particle Filters (PF), which allow for dealing with nonlinear and non-Gaussian problems but incur a prohibitive computational cost for real-time tracking of rapidly moving LEO debris [11,12]. In the specific domain of orbital determination, variations such as the Extended Kalman Filter (EKF) and the Unscented Kalman Filter (UKF) have become the standard [13]. While the EKF heavily suffers from Jacobian linearization errors under severe orbital nonlinearities, the UKF [14] leverages the unscented transform to handle them by propagating a set of sigma points through the system dynamics. Recent studies have applied UKF variants to improve reentry prediction and debris tracking [15], motivating its use as a strong conventional baseline for our comparative analysis. Nevertheless, these model-based approaches suffer from known limitations: they rely on explicit physical models that may not capture unmodeled perturbations (e.g., varying drag coefficients)

https://doi.org/10.3390/aisens2030010

and typically assume Gaussian noise distributions, which is often an oversimplification in complex radar environments [16].

To address the limitations of model-based filters, data-driven approaches based on Deep Learning (DL) have gained significant traction in recent years. Unlike traditional filters, Neural Networks (NNs) can learn complex dynamics and noise distributions directly from data. In the context of time series forecasting, Recurrent Neural Networks (RNNs), specifically Long Short-Term Memory (LSTM) networks [17] and Gated Recurrent Units (GRU) [18], have shown state-of-the-art performance.

Several recent works have started applying these techniques to aerospace tracking. For instance, Peng and Bai [19] investigated an artificial neural network-based machine learning approach to improve orbit prediction accuracy for resident space objects using historical orbit determination and prediction data. Similarly, Zhang et al. [20] investigated the use of LSTM neural networks for short-term LEO orbit prediction, showing improved prediction accuracy over conventional propagation-based approaches. In the radar domain, convolutional neural networks have been successfully employed for radar signal denoising and interference mitigation [21], while recurrent deep learning models have also been explored for multi-target tracking and data association in cluttered radar scenarios [22]. More recently, research has further explored the application of advanced recurrent models, transformer-based architectures, and hybrid deep learning techniques for resident space object tracking and radar data processing [23,24].

Despite these advancements, a significant gap remains in the literature regarding the simultaneous handling of measurement noise and temporal dynamics for sparse radar data. Most existing works address these challenges separately: recurrent neural network approaches for orbit prediction mainly focus on propagating or correcting orbital states over short horizons [19,20], while CNN-based radar methods typically target signal-level denoising or interference mitigation without explicitly modeling the orbital temporal evolution of sparse SST tracklets [21]. Similarly, recurrent data-association models have been explored for multi-target tracking in cluttered scenarios [22], but they address the association problem rather than learned one-step filtering of individual space-debris tracklets. Furthermore, standard sequential models often struggle to generalize when observations are extremely short (sparse tracklets) and heavily corrupted by non-Gaussian noise, a scenario where physical filters like UKF frequently diverge.

While recent developments have explored hybrid frameworks that integrate nonlinear neural networks (such as AC-LSTM) with parallel Kalman filters through confidence-weighted fusion mechanisms [25], most existing deep learning approaches still treat tracking either as a pure time series forecasting problem or a pure denoising task. In contrast, our standalone hybrid architecture performs both simultaneously on highly corrupted, sparse tracklets, eliminating the need for parallel model fusion or reliance on conventional filters during inference. Furthermore, a key innovation of our approach with respect to the existing deep learning literature is the integration of domain-aware preprocessing: we utilize Residual Learning to handle the vast distance scales of Range measurements, and Relative Pivoting to enforce local stationarity in unbounded angular domains (Azimuth).

The objective of this work is to design and validate this framework specifically to:

1. Bridge the gap between denoising and forecasting: by proposing a unified architecture that outperforms traditional filtering (UKF) specifically in scenarios with scarce and non-Gaussian noisy observations.
2. Validate Hybrid Deep Learning: assessing the effectiveness of a GRU+CNN architecture in extracting local measurement correlations (via CNN) to aid long-term trajectory prediction (via GRU).

3. Real-time feasibility: demonstrate the framework's flexibility for variable prediction horizons, while specifically validating its performance on next-step prediction ($m = 1$) with computational efficiency suitable for real-time tracking systems.

The paper is organized as follows: Section 2 details the materials and methods, providing the formal problem definition, the description of the synthetic dataset generation, the preprocessing strategy, and the proposed hybrid neural network architecture. Section 3 presents the experimental results and discussion, evaluating the training dynamics and comparing the trajectory reconstruction performance against the Unscented Kalman Filter. Finally, Section 4 summarizes the conclusions and outlines directions for future research.

2. Materials and Methods

2.1. Problem Definition

Let $\{x_t\}_{t=1}^{T}$ denote a sequence of radar measurements affected by noise, where each x_t represents the azimuth, elevation, range, and Doppler components at time step t. The goal is to estimate the future state $\hat{x}_{t+m}$ over a prediction horizon m, given only a limited sequence of past observations $\{x_{t-n+1}, \ldots, x_t\}$.

Formally, we seek to learn a function $f_\theta : \mathbb{R}^{n \times d} \to \mathbb{R}^d$, parameterized by θ, such that

$$\hat{x}_{t+m} = f_\theta(x_{t-n+1}, \ldots, x_t), \tag{1}$$

where d is the dimensionality of the feature space. The objective is to minimize the prediction error with respect to the true (noise-free) trajectory x^*_{t+m}:

$$\min_\theta \ \mathbb{E}\left[\|\hat{x}_{t+m} - x^*_{t+m}\|^2 \right]. \tag{2}$$

In the main experimental setting of this work, we use a short-horizon configuration with $m = 1$. Therefore, the task should be interpreted primarily as learned one-step filtering or radar measurement denoising, rather than long-horizon orbit propagation. Multi-step forecasting is evaluated separately as a sensitivity analysis, while full orbit determination, tracklet association, and collision-risk prediction are outside the direct scope of this study.

We distinguish here between four related but different tasks. One-step filtering estimates the next clean measurement from a short noisy observation window. Multi-step forecasting predicts several future samples without claiming full orbital-state determination. Orbit determination estimates a physically consistent orbital state and covariance from observations. Collision-risk prediction further requires conjunction geometry, uncertainty propagation, and downstream probability-of-collision metrics. The proposed method addresses the first task and provides preliminary evidence for short-horizon forecasting.

In this study, we assume a single target following a specific trajectory without classification of target type or maneuvering behavior. The dataset used for training the neural network is composed of simulated orbital trajectories that are converted into synthetic radar measurements, enabling a realistic yet tunable tracking scenario. The simulator allows the adjustment of different noise behaviors, making it possible to stress-test the model under multiple operating conditions.

2.2. Trajectory Simulator

To train the neural network, we generated a synthetic dataset using a custom-built orbital trajectory simulator. The use of synthetic data was necessary not only due to the scarcity of open-source radar datasets but primarily to ensure access to a precise ground truth for error calculation, which is often unavailable in real-world measurements. While analytical propagators like SGP4 applied to Two-Line Elements (TLEs) are standard for macroscopic catalog maintenance, they introduce inherent kilometric-level inaccuracies. Such errors would contaminate the ground truth, making it impossible to rigorously evaluate the network's ability to filter out meter-level radar noise. Therefore, a controlled numerical integration was preferred to establish a mathematically pristine baseline. Furthermore, obtaining actual raw radar logs containing severe, uncalibrated operational anomalies (e.g., clutter, hardware spikes) is currently restricted by institutional data policies. Therefore, carefully modeled synthetic data serves as the most rigorous alternative to establish a quantitative mathematical benchmark. Figure 2 illustrates the ground-based observation geometry and the corresponding local radar-frame projection used to define the simulated measurements. The radar sensing assumptions used in the synthetic tracklet generation are summarized in Table 1.

Table 1. Radar sensing assumptions used in the synthetic tracklet generation.

Parameter	**Value/Description**
Sensor type	Monostatic ground-based tracking radar
Observed coordinates	Range, azimuth, elevation, range-rate
Coordinate frame	ECI propagation frame; local topocentric radar frame for measurements
Sampling interval	$\Delta t = 0.188$ s
Range noise std.	5 m
Range-rate noise std.	2 m/s
Angular noise std.	0.5° (0.0087 rad)
Noise profiles	Standard, Spikes, Transient, State-Dependent
Target regime	Synthetic LEO debris-like objects

These assumptions should therefore be interpreted as an abstracted sensor model: the simulator does not reproduce the full electromagnetic radar processing chain, but it preserves the measurement variables, temporal cadence, uncertainty levels, and anomaly classes that directly affect the downstream tracking and filtering task.

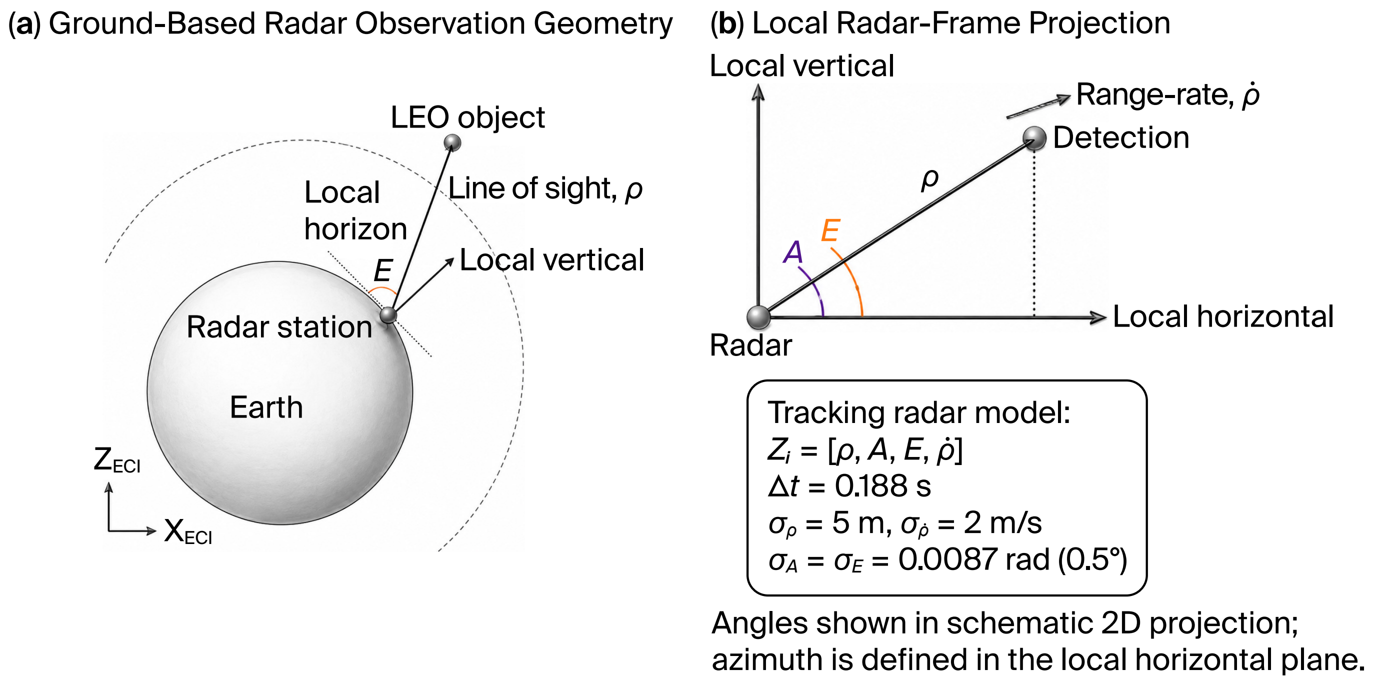

Figure 2. Schematic radar sensing geometry used to generate the synthetic tracklets. (**a**) A ground-based SST radar observes a LEO object through line-of-sight measurements. The clean target trajectory is propagated in the ECI frame and transformed into the radar local frame. (**b**) Two-dimensional schematic projection of the local radar frame. Each simulated detection is represented as $z_t = [\rho, A, E, \dot{\rho}]$, where ρ is range, A azimuth, E elevation, and $\dot{\rho}$ range-rate. Measurement noise is injected directly in radar coordinates according to the assumed sensor accuracy model.

Simulated radar measurements were initially generated in the local spherical coordinate system of a ground-based radar station, comprising azimuth, elevation, range, and range rate. These measurements were then converted into Cartesian Earth-centered Inertial (ECI) coordinates (x, y, z) by computing the corresponding positions and velocities, which served as additional features for training.

From an operational perspective, this simulation pipeline reproduces the main stages of a real SST acquisition chain. A ground-based tracking sensor first observes the object along its line of sight and produces time-stamped measurements in the local radar frame: Range, Azimuth, Elevation, and Range Rate. These measurements are then converted into the coordinate representations required by the tracking algorithm, while the simulated clean trajectory provides the reference state that is usually unavailable in real operational data. Sensor-dependent effects are subsequently introduced at the measurement level by imposing the sampling cadence, nominal covariance bounds, and non-ideal perturbations such as missed detections, clutter-like outliers, transient signal degradation, biased samples, and hardware-induced spikes. In this way, the generated tracklets emulate the imperfect outputs of a real acquisition system rather than ideal noiseless orbital trajectories.

The initial state of each simulated trajectory was determined based on a chosen orbital radius and initial orientation. From this starting configuration, the trajectory was propagated by integrating the equations of motion, including the J_2 zonal harmonic perturbation to account for the Earth's oblateness, which represents the dominant gravitational perturbation for LEO objects:

$$\frac{d^2\vec{r}}{dt^2} = -\frac{\mu_\oplus}{r^3}\vec{r} + \vec{a}_{J_2} \tag{3}$$

where:

- $\mu_\oplus = 3.986004418 \times 10^{14}\ \mathrm{m}^3/\mathrm{s}^2$ is the Earth's gravitational parameter;
- $r = R_\oplus + h$, where $R_\oplus$ is the Earth's radius and h is the altitude of the target above the surface;
- $\vec{a}_{J_2}$ represents the acceleration due to the J_2 zonal harmonic.

Since radar tracking of LEO objects typically involves short observation windows (tracklets), this perturbed model provides a highly accurate baseline for short-term smoothing tasks [26–28]. Each simulated observation window spans approximately 60 s. While atmospheric drag is often mandatory for multi-day orbit prediction, over a brief 60 s window, the spatial displacement caused by drag is strictly sub-meter. In stark contrast, the radar measurement noise fluctuates by tens or hundreds of meters. Consequently, within the boundary of a single tracklet, the dominant error source is overwhelmingly the sensor noise, rendering the impact of neglecting atmospheric drag negligible for the specific task of signal denoising.

After propagation, the orbital positions were transformed back into the radar's local spherical frame. Radar noise was then simulated by adding white Gaussian noise to the measurements, with parameters aligned to the estimated accuracy of the radar system. Simulations included a range of orbital altitudes and azimuth entry angles, generating diverse scenarios to test the model's generalization capability.

To ground the synthetic dataset in realistic physical sensor characteristics, the simulated measurements emulate the operational envelope of a ground-based tracking radar dedicated to Space Surveillance and Tracking (SST), analogous to prominent European systems like TIRA [29]. Such systems typically operate in the UHF, L, or S-bands, providing the necessary electronic or mechanical beam steering capabilities to acquire and track high-speed LEO targets over short observation windows. The specific baseline noise parameters adopted in our simulator (e.g., 5 m standard deviation for Range and 0.0087 rad for angular coordinates) were selected to reflect the typical measurement covariance of an active tracking radar under nominal atmospheric propagation conditions. By explicitly defining these sensor boundaries, we ensure that the neural network is trained to handle the exact physical constraints of modern space surveillance hardware.

2.2.1. Advanced Noise Modeling for Stress-Testing

While baseline Gaussian noise accurately models thermal sensor characteristics, real-world radar surveillance of Low Earth Orbit (LEO) objects is frequently corrupted by severe, non-ideal artifacts such as clutter, false echoes, temporary obscuration, and signal fading. To rigorously evaluate the robustness of the proposed hybrid architecture against these operational anomalies, the synthetic dataset was augmented with specialized adversarial noise profiles.

Rather than relying solely on uniform measurement errors, the trajectory generation pipeline was configured to inject extreme, non-stationary perturbations into the observations. A visual representation of the resulting simulated measurements, compared with the true orbital paths across all spatial coordinates, is illustrated in Figure 3.

The dataset, comprising a total of 10,000 simulated trajectories, was proportionally partitioned to include the following degradation scenarios (visually showcased in Figure 4):

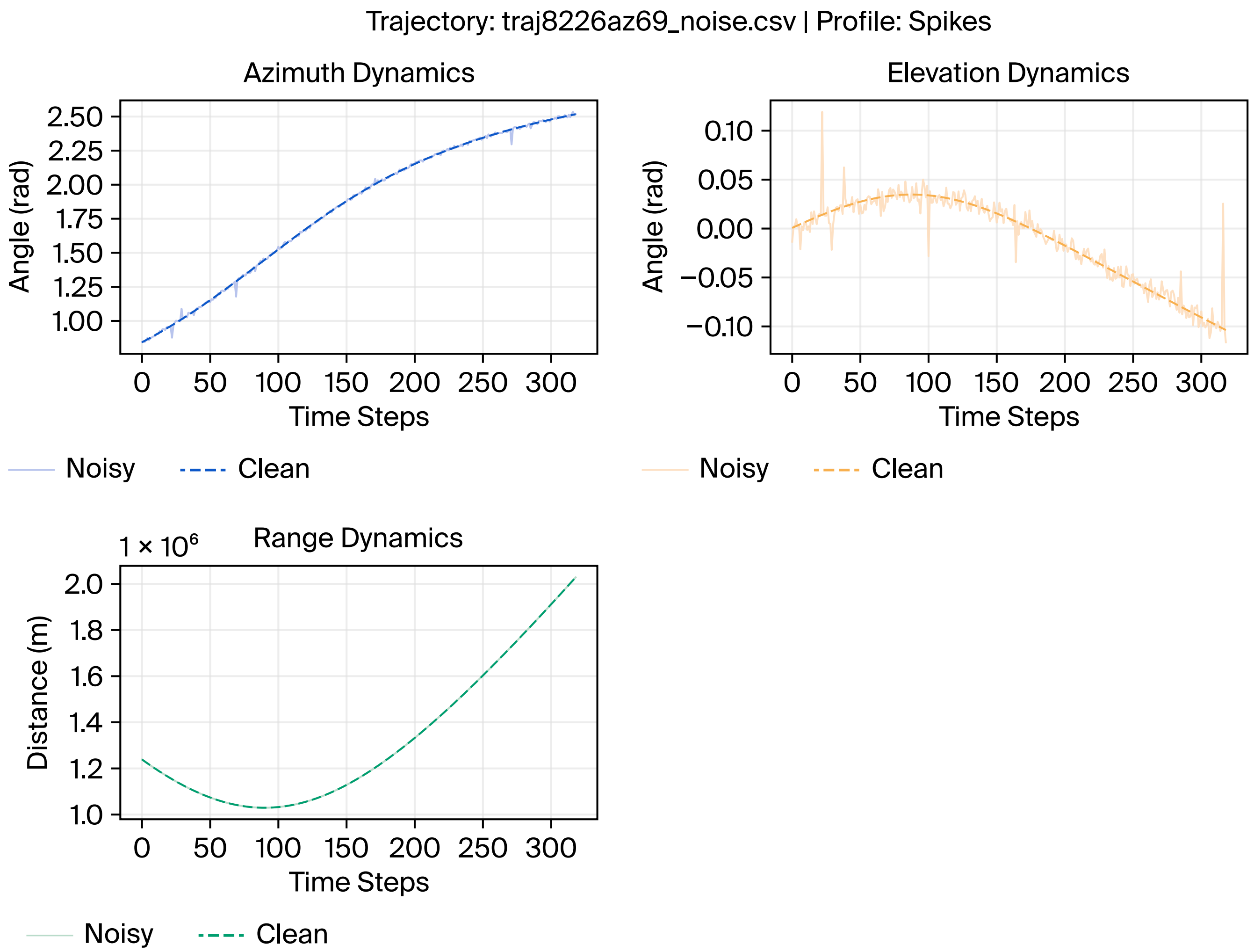

Figure 3. Visual representation of the synthetic dataset generation, illustrating the perturbation of a sample orbital trajectory. The subplots display the Azimuth, Elevation, and Range coordinates over the observation window, comparing the true orbital path (ground truth, dashed lines) against the simulated radar measurements (noisy, solid lines). The overall dataset comprises 10,000 trajectories distributed across four specialized noise profiles (Standard, Spikes, Transient, and State-Dependent).

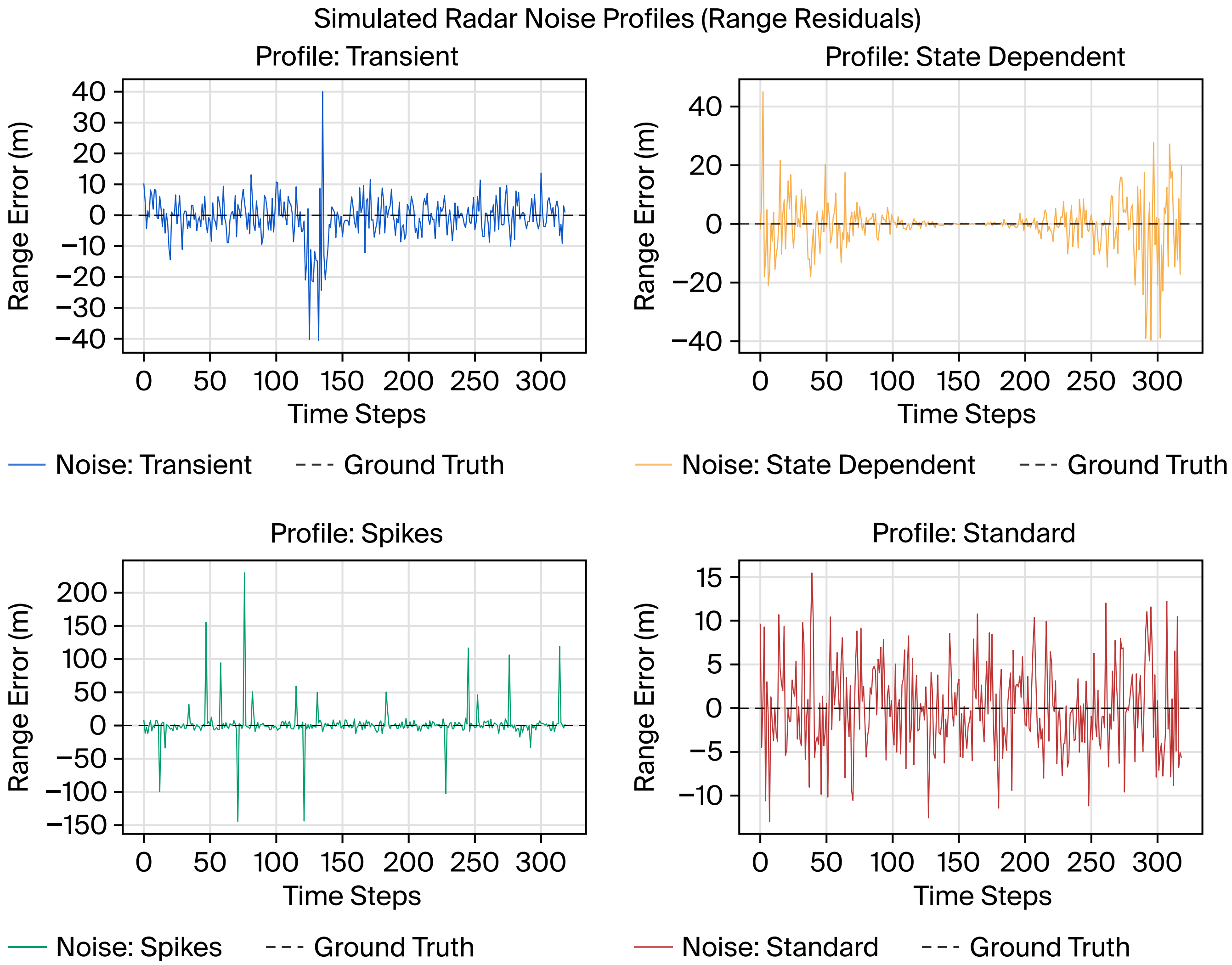

Figure 4. Showcase of the four distinct radar noise profiles (Range error residuals) implemented in the simulator. (**Top-Left**): Transient Jamming profile simulating temporary signal obscuration. (**Top-Right**): State-Dependent profile where noise variance scales quadratically with target distance. (**Bottom-Left**): Spikes profile simulating severe, isolated sensor faults. (**Bottom-Right**: Standard Gaussian thermal noise.

- Standard Profile: The baseline zero-mean Gaussian noise representing nominal sensor operation, serving as the control group for the performance evaluation.
- Spikes Profile (Impulsive Noise): Designed to simulate sporadic false detections or transient sensor faults. Severe, instantaneous errors—often exceeding the nominal standard deviation by an order of magnitude—were injected at random temporal indices. This profile explicitly challenges the filter's ability to reject isolated, non-physical anomalies without diverging.
- Transient Profile (Jamming/Obscuration): Emulates temporary sensor blinding, dense clutter patches, or electronic interference. In this scenario, the noise variance is significantly amplified over a continuous sequence of consecutive time steps, testing the model's capacity to maintain trajectory lock and preserve the orbital memory during prolonged signal degradation.
- State-Dependent Profile: Models the fundamental physics of the radar equation, where the Signal-to-Noise Ratio (SNR) intrinsically degrades with distance. The applied noise variance was scaled quadratically with respect to the target's instantaneous range. This generates a heteroscedastic noise distribution, reproducing the realistic behavior of tracking distant objects and severely testing algorithms that rely on constant error covariance matrices.

By exposing the network to these extreme, unpredictable noise conditions during the evaluation phase, this study aims to demonstrate the invariant feature extraction capabilities of the CNN-GRU framework and its superiority over standard analytical filters, which typically assume strict Gaussian error distributions and may diverge when facing severe measurement spikes.

2.2.2. Dataset Preparation

In time series forecasting using neural networks, specific data formatting is required to learn temporal patterns. We employed the "sliding window" approach to construct input-target pairs.

Given a trajectory time series $\{x_1, x_2, \ldots, x_T\}$, the sliding window technique with a window length n and a forecasting horizon m is formalized as follows:

- Input: For each time step t, the input vector X_t contains the values of the n preceding steps:

$$X_t = \{x_t, x_{t-1}, \ldots, x_{t-n+1}\} \tag{4}$$

- Target: The target y_t is the value at the forecasting horizon m:

$$y_t = x_{t+m} \tag{5}$$

In this study, we selected a window length of $n = 25$ points with a prediction horizon of $m = 1$ point. This configuration was chosen to evaluate the model's capability to perform immediate next-step smoothing. The windowing process is repeated for all trajectories. Given $T = 319$ points per trajectory, each trajectory yields $k = T - n - m + 1 = 294$ samples. With 10,000 simulated trajectories, the total dataset comprises approximately 2.94×10^6 input-target pairs.

2.3. Input Strategy and Data Preprocessing

Given the diverse physical characteristics of the tracked coordinates, a "one-size-fits-all" approach was deemed suboptimal. We designed a unified preprocessing pipeline based on an initial Exploratory Data Analysis (EDA) of the raw signals.

Figure 5 presents the probability density functions of the three target variables. Two critical observations drove our preprocessing choices:

1. Scale Disparity: The Range and Cartesian coordinates (r_x, r_y, r_z) operate in the order of 10^6 meters, whereas Azimuth and Elevation are in radians (10^{-1}). Without normalization, the gradient descent would be entirely dominated by the range dynamics, preventing the model from learning angular corrections.
2. Heterogeneous Distributions: The distributions exhibit varied shapes multimodal for Azimuth, highly peaked for Elevation and Doppler, and skewed for Range. Rather than forcing these distributions into a Gaussian shape via heavy transformation (e.g., Yeo–Johnson), we opted to preserve their physical structure using linear scaling and address the complexity via variable-specific target definitions.

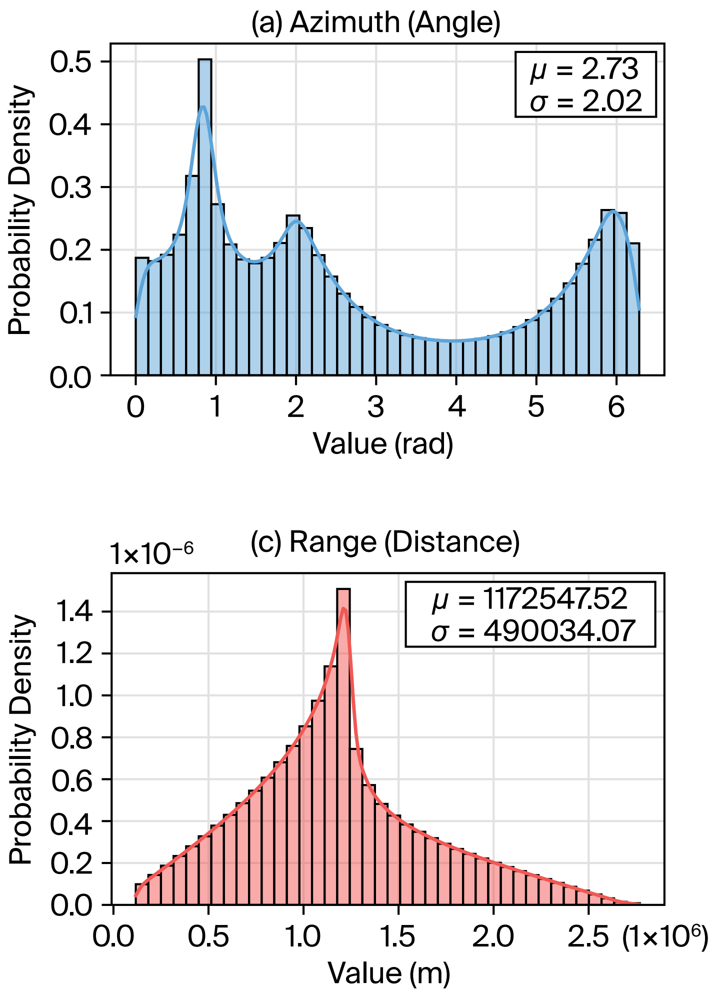

Figure 5. The plots illustrate the distinct physical characteristics of the predicted signals. (**a**): Azimuth exhibits a multimodal distribution due to orbital geometry. (**b**): Elevation shows a highly peaked distribution centered around the orbital plane. (**c**): Range spans a significantly larger magnitude (10^6 m) compared with angular variables, justifying the need for specific residual learning strategies.

2.3.1. Feature Engineering and Selection

To strictly respect temporal causality, we computed a set of "Delta" features representing the discrete backward difference: $\Delta x_t = x_t - x_{t-1}$. Feature selection was performed to identify the most relevant variables for each prediction task, ensuring the network receives the necessary geometric context:

- For Angular Coordinates (Elevation, Azimuth): The input vector includes the complementary angular measurement and the full 3D spatial coordinates (r_x, r_y, r_z). This combination captures the 3D spatial geometry required to fully resolve the angular position relative to the observer.
- For Range: The input prioritizes radial distance and Doppler shift. As shown in the distributions, Doppler provides a direct measurement of the range rate, which is critical for estimating the drift of the sensor error.

Additionally, the relative timestamp of each observation was included in the input vector to explicitly model the temporal cadence of the sequential data. By ingesting the time deltas between consecutive measurements as an explicit feature, the recurrent layers natively account for variable time intervals, irregular sensor sampling rates, and missing data points, which are common occurrences in real-world radar operations.

2.3.2. Normalization and Target Strategies

We applied a StandardScaler to all inputs to enforce zero mean and unit variance.

$$z = \frac{x - \mu}{\sigma} \tag{6}$$

This step is crucial for noise handling: by standardizing the input, the variance of the thermal radar noise is scaled to a consistent range across all trajectories. This allows the subsequent 1D CNN layers to learn universal, scale-invariant convolutional filters to detect and smooth out high-frequency noise patterns, regardless of the target's absolute distance or angular position. Furthermore, the operational limits of this framework are inherently tied to the physical radar equation. For extremely small debris (e.g., <10 cm) at higher LEO altitudes (>1000 km), the Signal-to-Noise Ratio (SNR) rapidly degrades below the sensor's detection threshold, preventing the formation of continuous tracklets. Our State-Dependent noise profile (Section 2.2.1) was specifically designed to simulate this performance boundary.

The learning objective and the reconstruction pipeline were then differentiated based on the variable type, as illustrated in Figure 6.

- Elevation (Direct Prediction): Since the variable is bounded and stable, the network directly approximates the function $f(x_t) \approx y_{t+1}$. The final estimate is obtained by simply inverting the standard scaling.
- Range (Residual Learning): Predicting absolute values in the order of 10^6 m is numerically unstable. Inspired by residual learning frameworks [30], we adopted a formulation where the network predicts the sensor error $\hat{e}$. The final trajectory $\hat{y}$ is reconstructed by adding this estimated correction to the noisy measurement:

$$\hat{y}_{t+1} = y_{meas}[t+1] + \mathcal{N}^{-1}(\mathrm{NN}(x_t)) \tag{7}$$

This approach simplifies the optimization landscape, as the residuals are typically sparse and centered around zero, unlike the unbounded absolute trajectory.

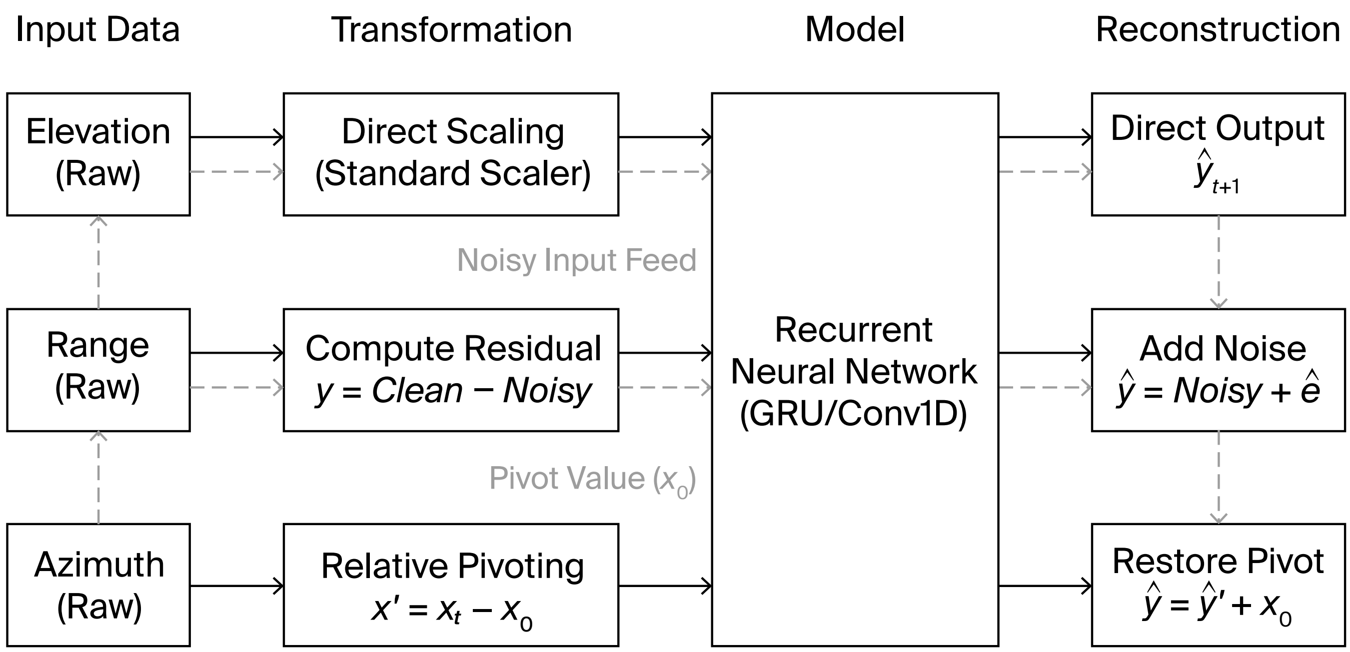

Figure 6. Overview of the variable-specific processing pipelines. While Elevation relies on direct prediction, Range and Azimuth utilize specialized transformations (Residual Learning and Pivoting, respectively) to handle scale and unbounded domains. The dashed lines indicate information from the input preserved for the reconstruction phase.

- Azimuth (Relative Pivoting): Azimuth trajectories can exhibit non-stationary behavior and unbounded growth (phase wrapping). To enforce local stationarity [31], we implemented a window-relative transformation. Given an input window W, the first element w_0 serves as the pivot and is subtracted from the sequence ($W' = W - w_0$). The absolute prediction is reconstructed by restoring the pivot:

$$\hat{y}_{abs} = \hat{y}_{rel} + w_0 \tag{8}$$

  This invariance property allows the model to generalize across different orbital orientations by focusing solely on the local trajectory morphology.

2.4. Machine Learning Architecture

To address the problem of trajectory smoothing from noisy radar data, we designed a hybrid Deep Learning architecture. Deep learning models such as Convolutional Neural Networks (CNNs) [32] and Recurrent Neural Networks (RNNs) [33] offer powerful alternatives to traditional filtering. In particular, Gated Recurrent Units (GRUs) [34] address the vanishing gradient problem of standard RNNs using a gating mechanism:

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t] + b_z) \tag{9}$$

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t] + b_r) \tag{10}$$

$$\tilde{h}_t = \tanh(W \cdot [r_t \odot h_{t-1}, x_t] + b) \tag{11}$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \tag{12}$$

Our proposed model combines GRUs with 1D Convolutional layers. Hybrid architectures combining convolutional and recurrent layers have recently shown superior performance in complex sensor tasks, such as GNSS monitoring [35] and visual-inertial odometry [36]. This hybrid approach leverages the GRU's ability to maintain a memory of long-term orbital dynamics (via the hidden state h_t) and the CNN's effectiveness in extracting local features and smoothing high-frequency noise from the input signal.

Network Design

The specific architecture, illustrated in Figure 7, consists of:

- Stateful GRU Layer: A single GRU layer with 256 units. We implemented this layer as stateful, meaning the hidden state is preserved across batches. This is crucial for maintaining orbital context over long sequences that exceed the single batch window.
- 1D Convolutional Layers: The GRU output sequence is passed through four 1D convolutional layers.
  – Layers 1 and 3: 32 filters, kernel size 5, Leaky ReLU activation.
  – Layer 2: 64 filters, kernel size 5, Leaky ReLU activation.
  – Output Layer: 1 filter (kernel size 5), linear activation to generate the continuous prediction.

This design was chosen after comparative experiments against stand-alone CNNs and GRUs. The results demonstrate that the hybrid model provides a consistent improvement, exploiting the complementary strengths of recurrent modeling for dynamics and convolution for noise filtering.

Since the dataset comprises multiple independent trajectories, maintaining the GRU state indefinitely would introduce errors when switching from one trajectory to the next. To prevent "state contamination" between unrelated samples, we implemented a custom training loop (overriding `train_step`). This logic resets the network's internal states solely at the end of each trajectory sequence, ensuring that the orbital context is preserved only within relevant observation windows.

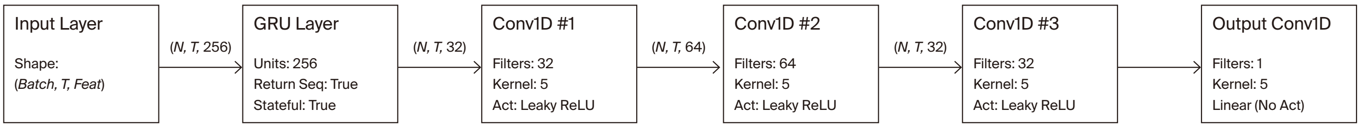

Figure 7. Architecture of the proposed hybrid neural network. The model processes the input sequence through a GRU layer (256 units) to capture temporal dependencies, followed by a stack of 1D Convolutional layers (filters: 32-64-32) for local feature refinement. The final convolution projects the features into the target prediction space. All intermediate layers utilize Leaky ReLU activation.

Model performance was assessed using standard regression metrics:

- Mean Squared Error (MSE): Used as the training loss function to penalize large outliers.
- Root Mean Squared Error (RMSE): Provides error estimation in the original unit of measurement (e.g., meters or degrees).
- Mean Absolute Error (MAE): Offers a robust performance indicator less sensitive to extreme outliers [37].

To validate the results, we reconstructed full trajectories from the predicted sliding windows, allowing for a direct point-by-point comparison against the ground truth (noise-free) simulated paths.

2.5. Training Strategy and Hyperparameter Tuning

The training process utilized teacher forcing [38], where the ground truth from the previous time step is used as input for the current step during training, simulating the availability of continuous radar measurements. Additionally, data shuffling was applied at the trajectory level to prevent order-dependent bias.

To optimize the architecture, we performed hyperparameter tuning using both Random Search [39] and Hyperband [40]. The bounds of the search space were explicitly constrained to balance the model's nonlinear approximation capacity with the strict computational latency required for real-time tracking (ensuring an inference time significantly lower than the radar pulse repetition interval). We explored the following search space:

- GRU Units: [128, 256, 512]
- Batch Size: [8, 16, 32, 64]
- Conv Filters: [32, 64, 128]
- Kernel Size: [3, 5, 7]
- Optimizer: [Adam, RMSprop, SGD]

The optimal configuration identified and used for the final results consists of: a GRU layer with 256 units, followed by convolutional layers with filter sizes of 32, 64, 32, and 1, respectively. The Adam optimizer was selected with an initial learning rate of 10^{-3}. A batch size of 64 was employed, utilizing early stopping with a patience of 5 epochs to prevent overfitting.

3. Results and Discussion

All the results presented in this section were obtained by running the neural network on dual AMD EPYC 7402 24-core processors, totaling 96 logical CPUs, with 256 GB of RAM and two NVIDIA GeForce RTX 4090 GPUs, each with 24 GB of video memory. The software was executed in a TensorFlow containerized environment [41], and as a result, the network achieved an average inference time of 25 milliseconds per input window and the training time is approximately 3 h.

3.1. Training Results

The training process was conducted using the Adam optimizer with an initial learning rate of 10^{-3}. To ensure optimal convergence and prevent overfitting, we implemented a robust callback strategy including Early Stopping (patience = 10) and a dynamic Learning Rate Scheduler ('ReduceLROnPlateau'), which halves the learning rate upon detecting a plateau in the validation loss (patience = 3).

Figure 8 illustrates the evolution of the Loss function (MSE) and the evaluation metrics (RMSE, MAE) over the training epochs for the Elevation variable. As shown in the left panel, the loss decreases rapidly during the first 20 epochs, indicating that the network quickly learns the primary trajectory dynamics. Subsequently, the curve flattens as the learning rate is automatically reduced, allowing the optimizer to fine-tune the weights in the local minimum.

The training was automatically halted by the early stopping mechanism when no further improvement in validation loss was observed, restoring the best model weights obtained at Epoch 87. Crucially, the Training and Validation curves remain closely aligned throughout the process (right panel), with the validation RMSE stabilizing around 2.0×10^{-3}. This close correspondence indicates a high generalization capability and confirms that the model is not suffering from overfitting, effectively capturing the underlying physical laws of the debris motion rather than memorizing the noise.

Table 2 summarizes the RMSE values for Elevation, Azimuth, and Range, obtained from the various models analyzed. For simplicity, only the RMSE values are reported in the table, as the MAE exhibits a similar trend. The Perceptron exhibited the worst performance, while the Convolutional model and RNNs led to significant improvements. The integration of RNNs and CNNs further reduced the error. Finally, hyperparameter tuning on the last model produced the best results, demonstrating the effectiveness of this combination in optimizing the predictions.

Table 2. RMSE values on the test set for Elevation, Azimuth, and Range obtained using different models. The RNN model combined with a convolutional network, optimized through hyperparameter tuning, delivered the best performance.

	Elevation	**Azimuth**	**Range**
MLP	8.0000×10^{-3}	1.5500×10^{-2}	6.7697×10^{-4}
CNN	2.3000×10^{-3}	9.2300×10^{-3}	5.0834×10^{-4}
RNN	2.4000×10^{-3}	7.1700×10^{-3}	2.4315×10^{-4}
RNN + CNN	2.0000×10^{-3}	4.9900×10^{-3}	1.8424×10^{-4}
RNN + CNN (Tuned)	1.9000×10^{-3}	4.4500×10^{-3}	1.7367×10^{-4}

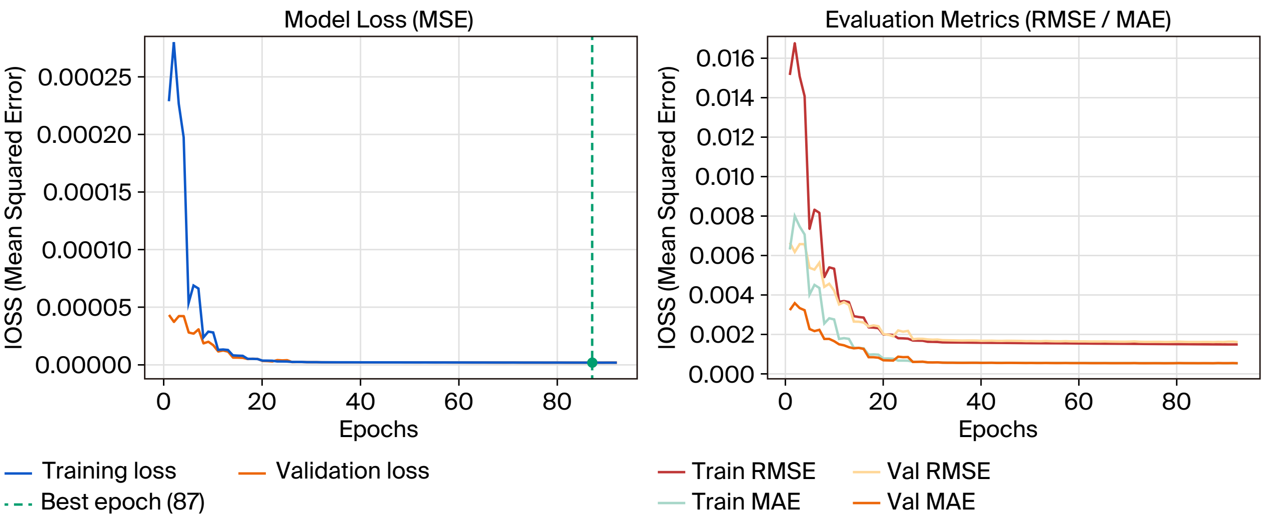

Figure 8. Training dynamics for the Elevation model. (**Left**): The Mean Squared Error (MSE) loss for training (blue) and validation (orange) sets. (**Right**): Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) trends. The vertical green dotted line marks the best epoch (87), where the validation loss reached its global minimum before early stopping.

3.2. One-Step Tracklet Filtering Results

This section evaluates the proposed model in the main operating regime used throughout the paper: one-step filtering with an input window of 25 radar measurements and prediction horizon $m = 1$. The objective is to reconstruct the clean next measurement from corrupted radar tracklets. Therefore, the reported results quantify short-horizon denoising and filtering accuracy rather than long-term orbit prediction.

The analysis is conducted separately for the three coordinates: Range, Azimuth, and Elevation. For each variable, two complementary visualizations are provided to assess the reconstruction performance.

Figure 9 first presents the extended qualitative comparison with the additional analytical baselines.

The first type of plot (Figures 10–12) displays the trajectory reconstruction. It shows the trajectory predicted by the network, the original trajectory with simulated noise, the ideal (noise-free) trajectory, and the estimation provided by the Unscented Kalman Filter (UKF). To enhance the visualization of the denoising capabilities, a zoomed inset is included in each graph. This inset highlights a specific segment of the trajectory, allowing for a closer comparison between the smooth curve generated by the Neural Network (NN), the UKF output, and the noisy input. In all three cases, the model clearly improves the quality of the radar measurements, reducing the deviation from the ideal trajectory.

The second type of plot (Figures 13–15) presents a detailed error analysis. These figures combine the time-evolution of the absolute errors with a probability density function (histogram) of the residuals. This dual representation demonstrates that the NN error distribution is narrower and more centered around zero compared with the UKF. Furthermore, quantitative metrics, including the Root Mean Square Error (RMSE) and the percentage improvement over the UKF, are reported directly within the figures. This confirms that the neural network achieves better accuracy across all variables considered, specifically for Range, Azimuth, and Elevation.

The UKF was initialized using the first three radar detections: the initial position was estimated as their average, and the initial velocity was computed from these detections combined with the known orbital height. The UKF parameters were tuned for the test scenario. The process and measurement noise were modeled as Gaussian, with standard deviations set according to the radar's accuracy in each coordinate. The associated covariance matrices were defined accordingly.

It was observed that the UKF requires approximately 50 warm-up points before producing reliable estimates, often exhibiting high error peaks during initialization. This behavior is consistent with the known sensitivities of the Unscented Transform to initial state uncertainty and its potential for suboptimal performance in the presence of heavy non-Gaussian noise or high nonlinearities [12,16]. Consequently, to ensure a fair comparison, the reported metrics and the error visualizations exclude this initial warm-up phase, focusing on the steady-state performance of both filters.

To address the sensitivity of Kalman-type filters to initialization, covariance tuning, and process-noise assumptions, we also report an extended set of conventional baselines. In addition to the UKF, we evaluate an EKF under the same constant-velocity measurement model, an adaptive UKF with innovation-based measurement-noise adaptation, and a robust Huber-weighted UKF designed to reduce the effect of impulsive outliers. A qualitative comparison of the one-step elevation filtering performance, including the newly introduced analytical baselines, is visually presented in Figure 9. All recursive filters are evaluated after discarding the first 50 samples as warm-up. The neural network output is evaluated without temporal smoothing or post-processing.

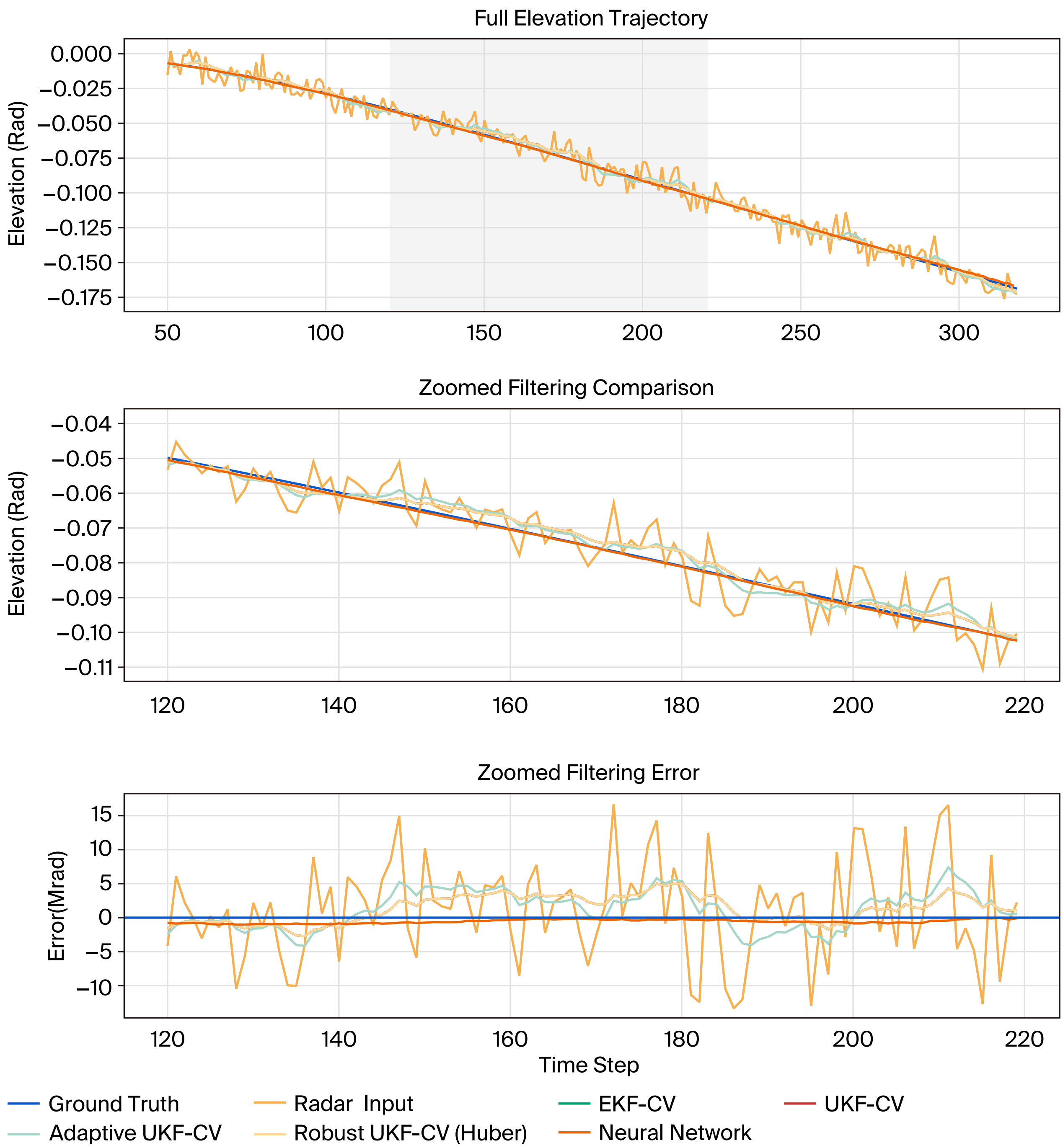

Figure 9. Qualitative comparison of one-step elevation filtering on a representative trajectory. The (**top panel**) shows the full trajectory after the recursive-filter warm-up region, with the shaded area indicating the zoomed interval. The (**middle panel**) compares noisy radar measurements, ground truth, Kalman-type baselines, and the neural network output within the zoomed segment. The (**bottom panel**) reports the corresponding filtering error in milliradians. The neural network prediction is shown without temporal smoothing or post-processing.

The EKF-CV and UKF-CV baselines achieve nearly identical performance because the elevation-only constant-velocity model is linear in both the state transition and measurement functions. The Huber robustification does not significantly alter the result in this representative segment, indicating that the selected Kalman tuning already suppresses most impulsive deviations. The adaptive UKF performs worse in this case, likely due to over-adaptation of the measurement covariance under non-stationary radar noise. Even after re-tuning the Kalman-type filters to avoid an overly noisy response, the proposed GRU-CNN architecture consistently outperforms all analytical baselines, effectively halving the RMSE and severely restricting the peak error magnitude.

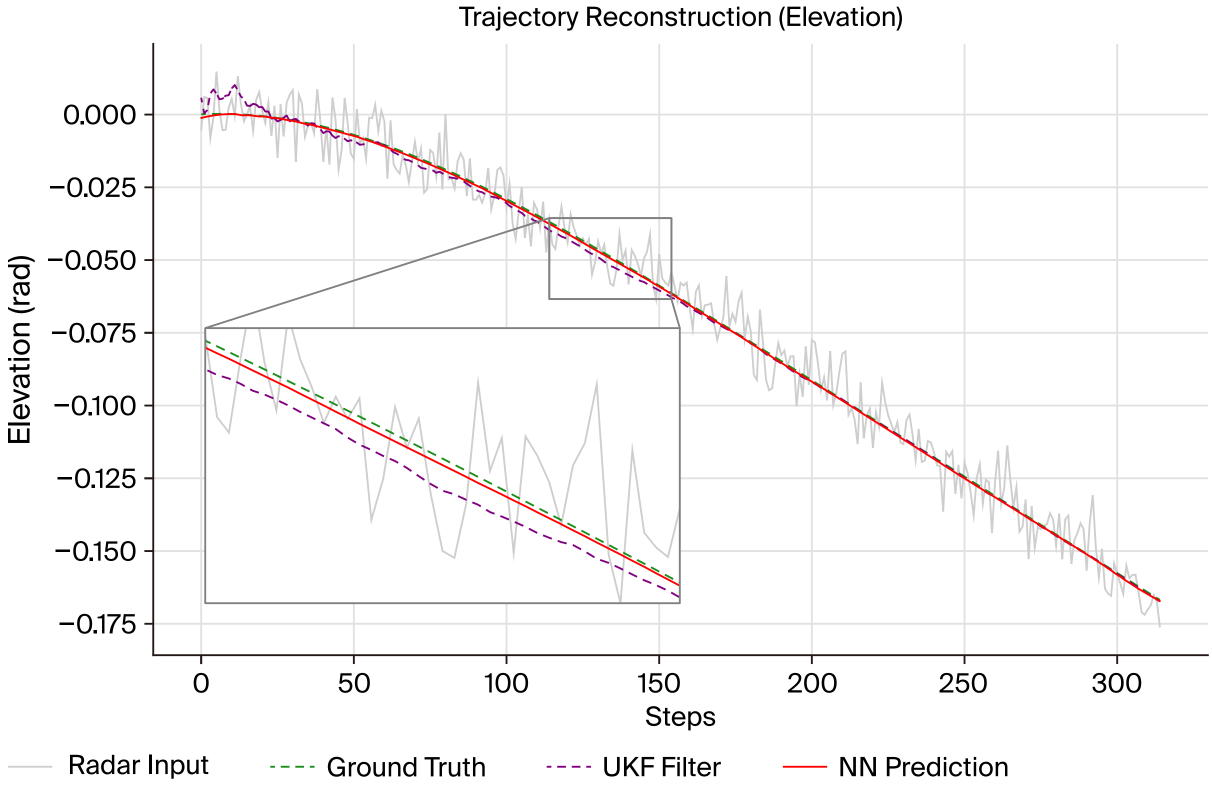

Figure 10. Elevation trajectory reconstruction. The main plot shows the comparison between the noisy radar input (gray), the UKF estimation (purple), the Neural Network prediction (red), and the ground truth (green). The zoomed inset highlights the superior smoothing capability of the NN compared with the UKF in a specific trajectory segment.

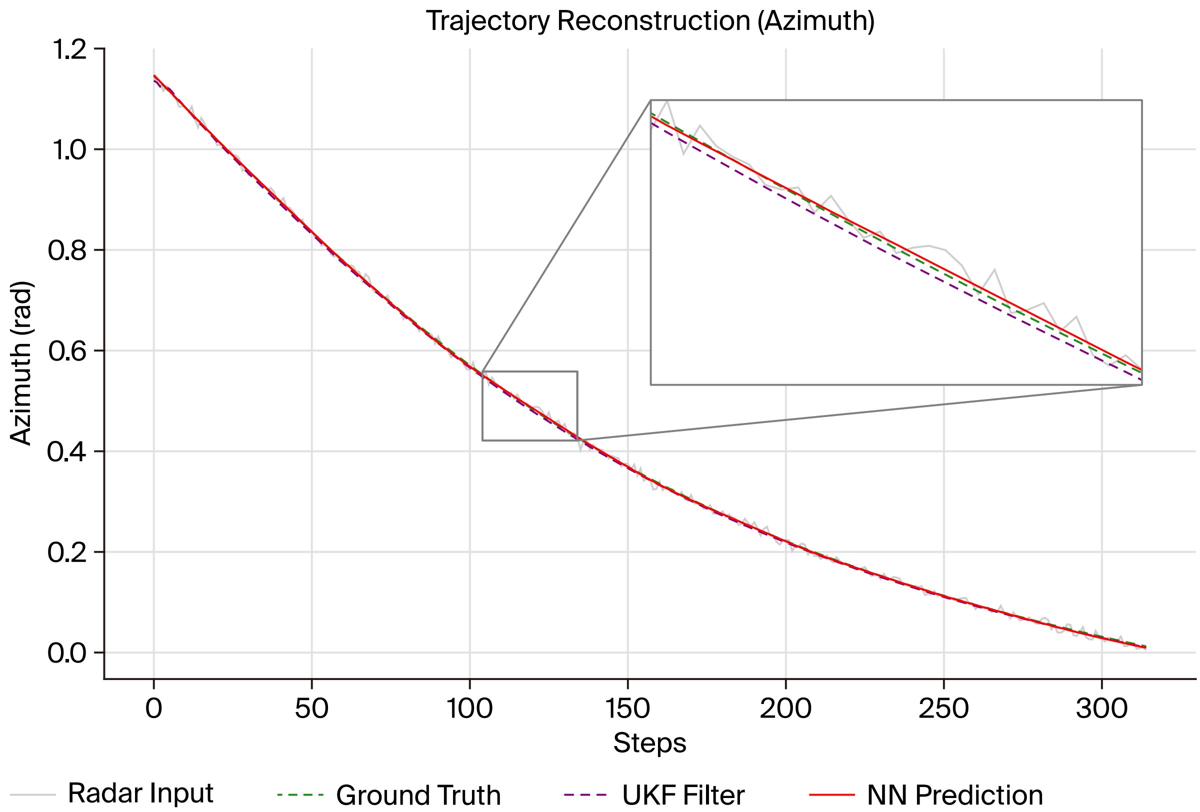

Figure 11. Azimuth trajectory reconstruction. Comparison of the denoised azimuth trajectory against the UKF and the original noisy signal. The inset demonstrates the model's ability to recover the true path even in the presence of significant noise variance.

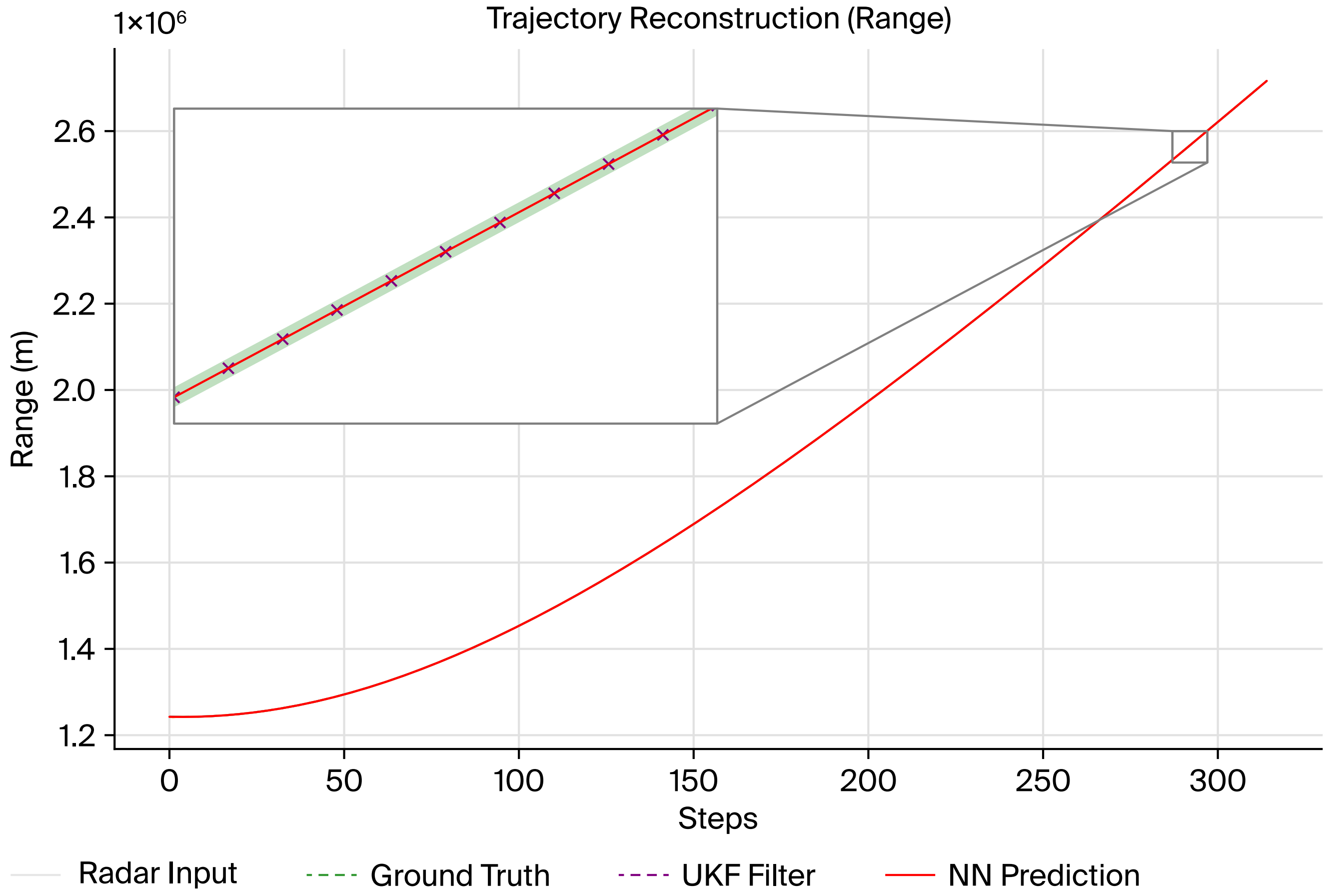

Figure 12. Range trajectory reconstruction. The plot illustrates the effectiveness of the proposed model in filtering range measurements. Due to the massive absolute scale of the target distance (order of 10^6 m), the subtle filtering deviations between the UKF and the Neural Network are inevitably compressed in this absolute reference frame. For a highly distinguishable, fine-grained comparison of the tracking performance, please refer to the residual error analysis presented in Figure 15.

The corresponding quantitative metrics are summarized in Table 3.

Table 3. One-step elevation filtering performance on a representative trajectory after discarding the first 50 samples as recursive-filter warm-up. The neural network output is evaluated without temporal smoothing or post-processing. Kalman-type filters use the selected tuning parameters $q_{\text{scale}} = 0.001$ and $r_{\text{scale}} = 20.0$.

Method	RMSE (mrad)	MAE (mrad)	Max Error (mrad)
Radar input	7.11	5.81	20.86
EKF-CV	1.88	1.51	5.02
Robust EKF-CV (Huber)	1.88	1.51	5.02
UKF-CV	1.88	1.51	5.02
Robust UKF-CV (Huber)	1.88	1.51	5.02
Adaptive UKF-CV	2.61	2.15	7.39
Neural Network	0.88	0.80	1.38

As specified in Section 2.2.2, for the comparison between the various models, input sequences of 25 points were used, with the prediction of a single point in the output. However, it is important to assess how the network's performance changes when these requirements become more stringent. Figure 16 shows the RMSE trend on the test set as the input sequence length decreases from 25 to 5 points. Each test was repeated twice: once with the prediction of one point, and once with five points. The results show that, across all three variables (Azimuth, Elevation, and Range), the performance remains stable with minimal degradation when predicting multiple points. Specifically, for Azimuth, the RMSE increases slightly at longer input sequence lengths for five detections, but the overall trend does not worsen significantly. For Elevation, there is a modest increase in RMSE when predicting five points compared with one, but the values remain close, particularly for shorter input sequences. Similarly, for Range, the performance shows only slight differences, with a small increase in RMSE for five points, yet overall, the performance remains low and stable.

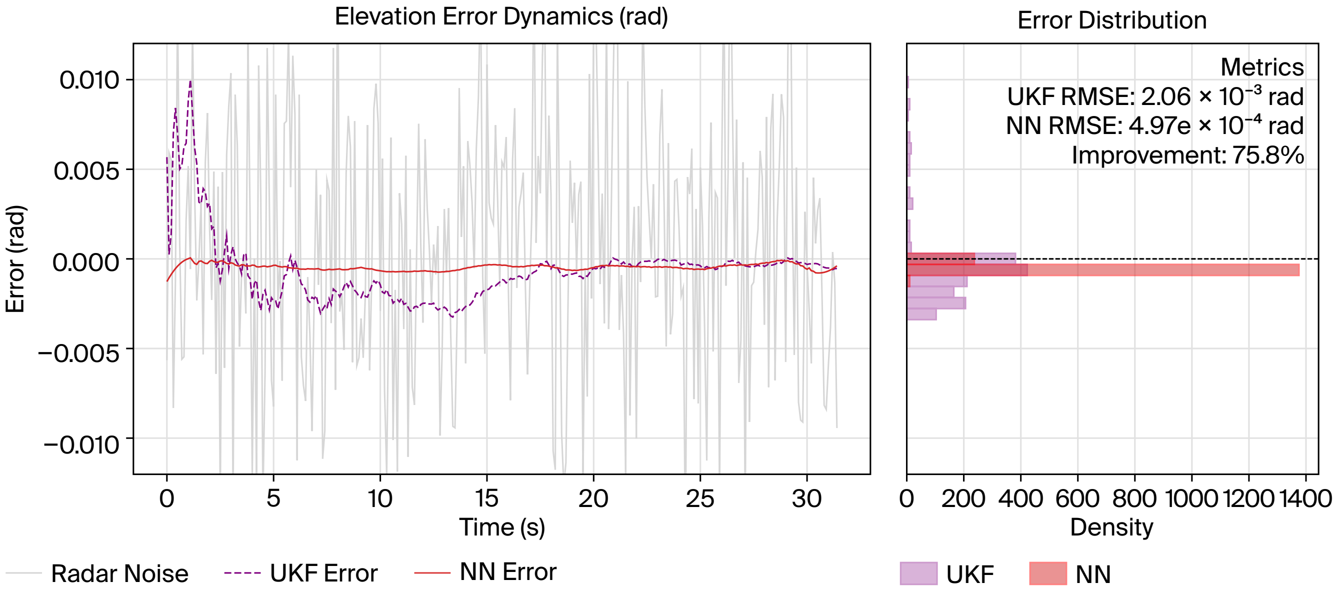

Figure 13. Elevation error analysis. The (**left panel**) shows the error dynamics over time for both the UKF (purple) and the Neural Network (red), excluding the warm-up phase. The (**right panel**) displays the error distribution (histogram), highlighting a tighter concentration around zero for the NN, which corresponds to a lower RMSE.

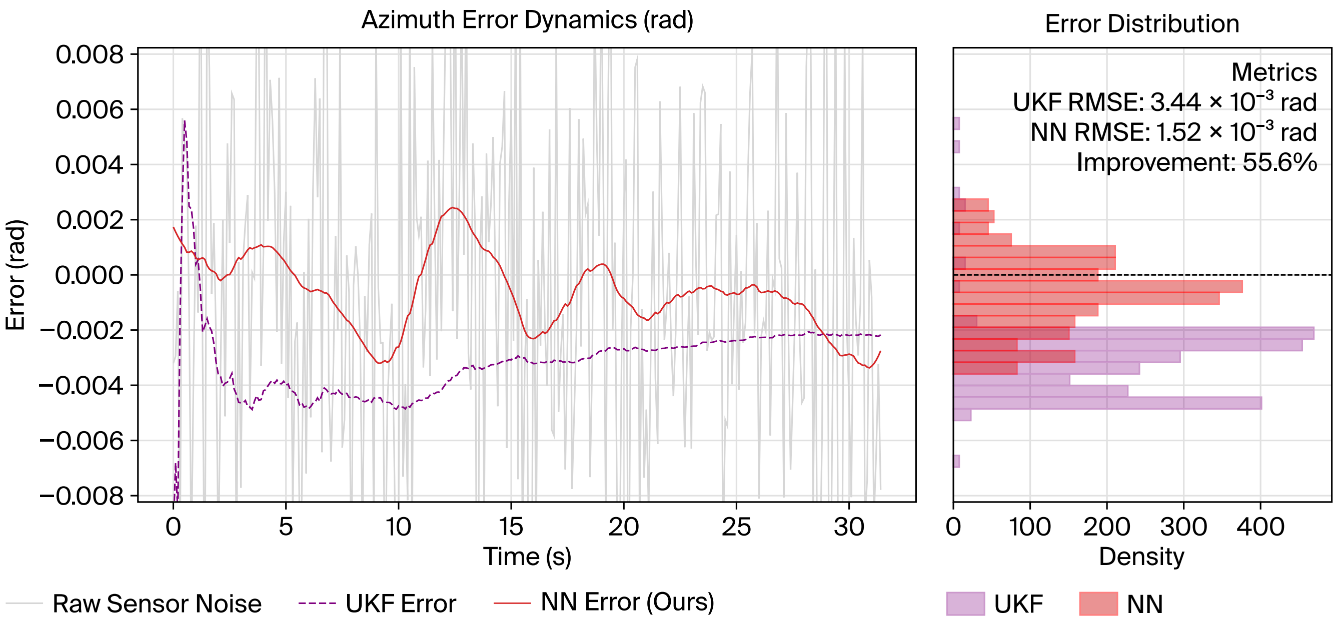

Figure 14. Azimuth error analysis. Comparison of residual errors between the UKF and the NN. The probability density function on the right confirms the reduction in variance achieved by the neural network model.

Beyond the numerical improvement in RMSE and MAE, the practical implications of the proposed model are significant. In our experiments, the UKF requires approximately 50 radar measurements before converging to a reliable estimate, whereas our network achieves stable accuracy already after the first observation window. This capability is crucial in realistic space surveillance scenarios, where observation windows are often short due to orbital dynamics and sensor limitations.

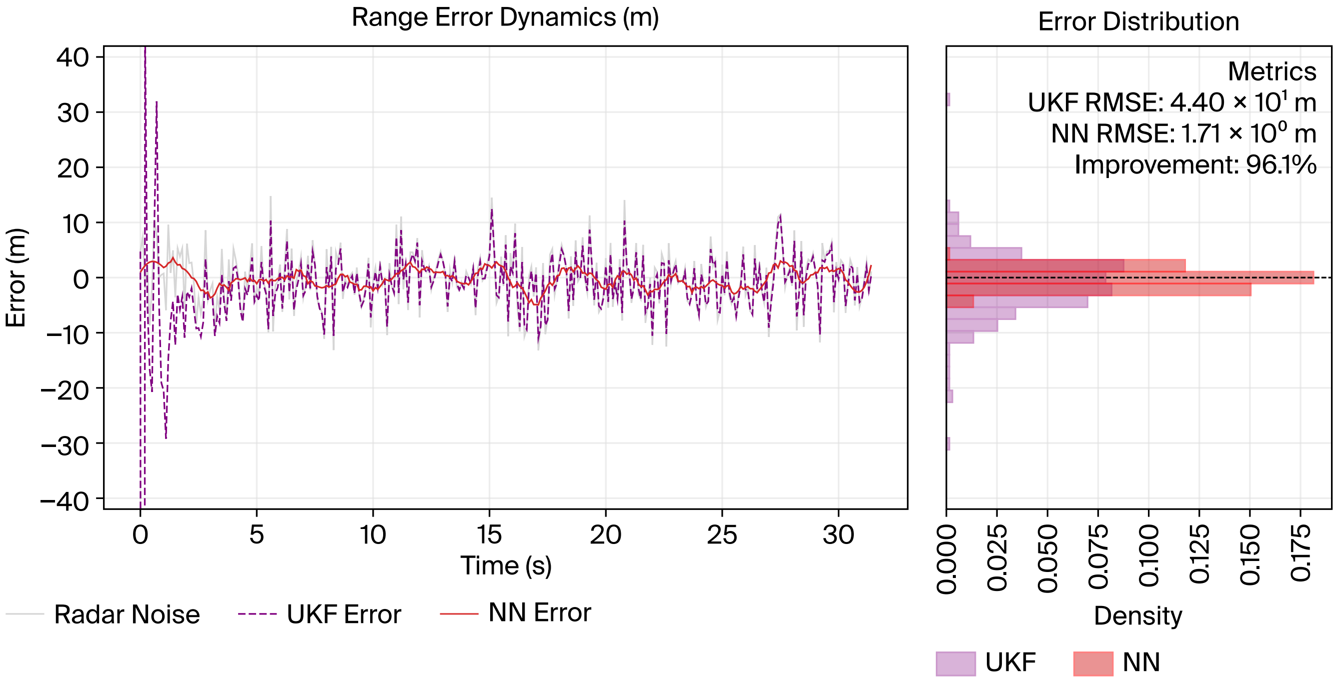

Figure 15. Range error analysis. The time series plot (**left**) and the error histogram (**right**) demonstrate the quantitative improvement of the proposed method over the UKF. Key metrics, including RMSE and percentage improvement, are reported in the figure.

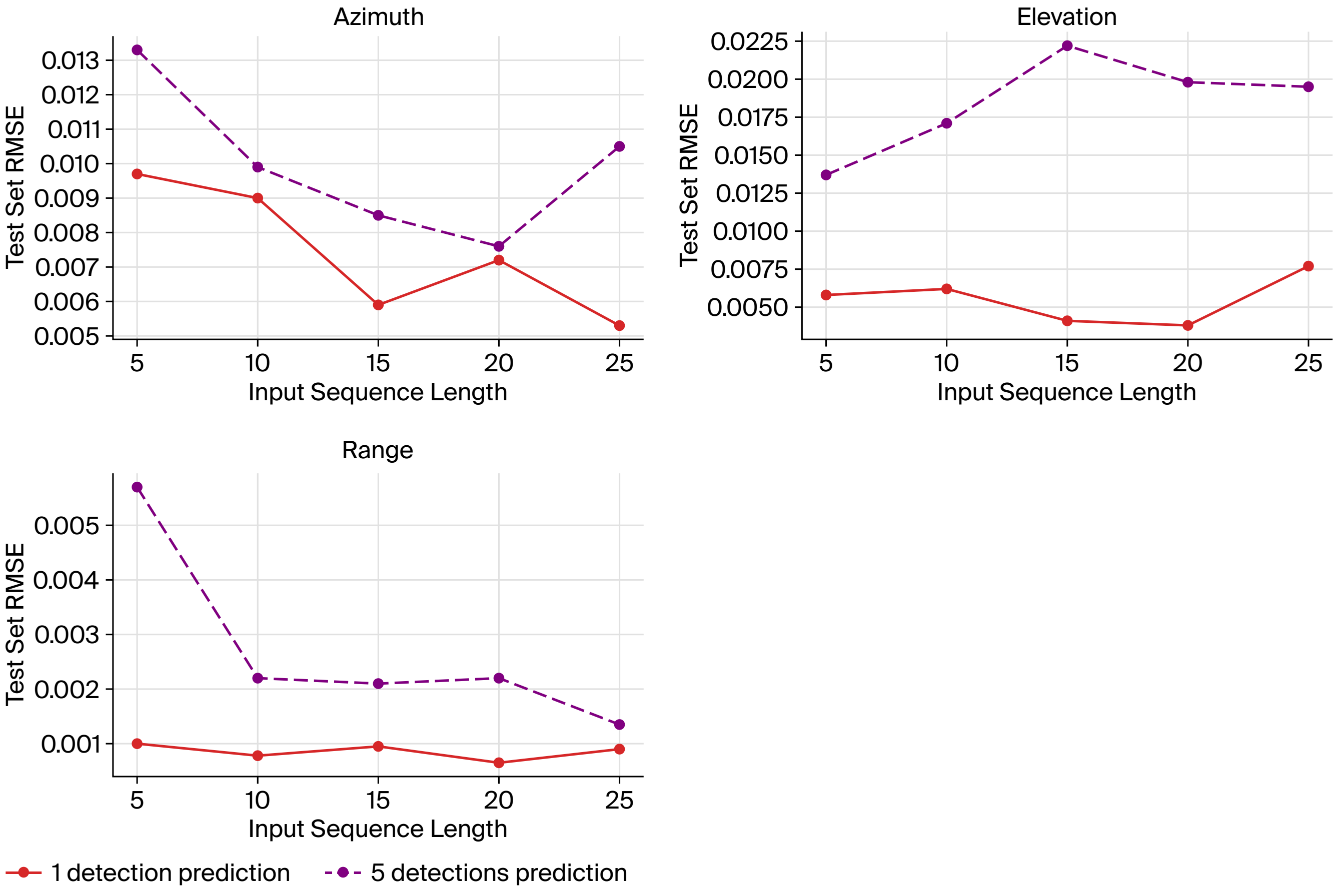

Figure 16. The three plots above show the test set results, varying the input sequence length and output sequence length. The x-axis represents the input sequence length, while the y-axis shows the RMSE values. Red line corresponds to predictions with only 1 point, and violet line to predictions with 5 points.

Moreover, the reduction in trajectory estimation error translates into narrower uncertainty envelopes, which in turn improves downstream tasks such as collision probability assessment and radar resource allocation. In operational terms, this means that fewer false alarms are generated and tracking can be maintained even when only sparse or noisy measurements are available.

Thus, the proposed method not only improves accuracy in a statistical sense but also provides tangible benefits for space situational awareness systems.

3.3. Robustness Analysis Under Extreme Noise Conditions

To rigorously address the operational challenges of space surveillance, the proposed framework was subjected to stress testing using the adversarial noise profiles defined in Section 2.2.1. While standard Gaussian noise provides a baseline for thermal sensor characteristics, assessing the model's resilience to unpredictable artifacts—such as clutter, false echoes, and signal jamming—is critical for real-world deployment.

3.3.1. Qualitative Evaluation of Anomaly Rejection

Figure 17 provides a comprehensive visual analysis of the model's behavior when subjected to severe measurement anomalies (Spikes profile). The top panels illustrate macroscopic segments of the Azimuth and Elevation trajectories, demonstrating how the neural network maintains a stable trajectory lock. Despite the continuous injection of extreme noise spikes into the raw radar signal, the continuous prediction remains perfectly aligned with the ground truth, exhibiting no signs of divergence.

To further dissect this invariant behavior, the bottom panel of Figure 17 isolates a microscopic view of a single, catastrophic sensor fault in the Elevation coordinate. In this scenario, the radar registers a sudden, non-physical jump (reaching approximately 0.18 rad), deviating significantly from the nominal orbital path.

Traditional filtering techniques that rely strictly on sequential updates are highly susceptible to such outliers, often requiring multiple subsequent measurements to correct the induced state deviation. In contrast, the hybrid GRU-CNN architecture successfully identifies the measurement as a non-physical artifact. By leveraging the temporal memory of the preceding trajectory segment (encoded within the GRU hidden state h_t) and the local spatial filtering of the CNN, the network effectively rejects the anomaly. The resulting prediction ($\approx$0.012 rad) remains tightly aligned with the ground truth ($\approx$0.002 rad), achieving a substantial reduction in the noise impact without diverging.

3.3.2. Quantitative Performance Across Noise Profiles

To statistically validate this robustness, the model's Root Mean Square Error (RMSE) was evaluated across a representative subset of the test dataset, categorized by the applied noise profile. The results, summarized in Table 4, demonstrate the framework's exceptional stability across all tracked coordinates.

Focusing on the angular coordinates, the network exhibited remarkable invariance. Under the Standard Gaussian profile, the Elevation prediction achieved a baseline error of 45.19 mrad. When subjected to severe isolated anomalies (Spikes) or prolonged signal corruption (Transient noise), the error did not degrade, but rather stabilized at 40.70 mrad and 41.31 mrad, respectively. This resilience provides quantitative evidence that the recurrent layers successfully filter out non-physical impulsive noise without overfitting, maintaining a robust representation of the underlying orbital kinematics.

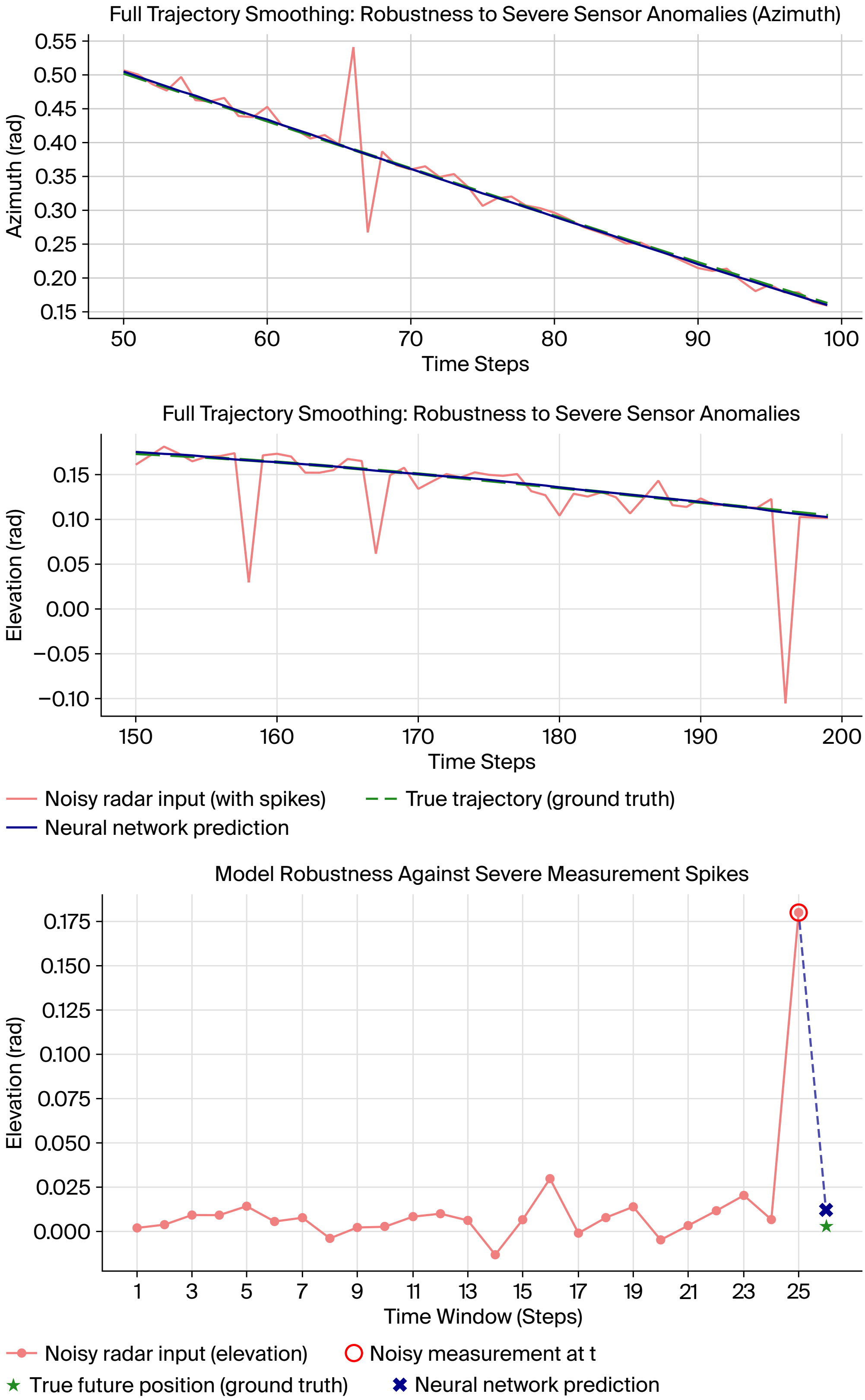

Figure 17. Qualitative robustness analysis of the hybrid GRU-CNN framework against severe sensor anomalies. (**Top**) Zoomed-in segments of the Azimuth and Elevation trajectory reconstructions. The neural network (solid dark blue line) consistently maintains trajectory lock with the ground truth (dashed green line), effectively filtering out the extreme, non-physical noise spikes injected into the raw radar signal. (**Bottom**) A microscopic view of a single, isolated measurement spike in the Elevation coordinate. While the raw radar input (red line) diverges abruptly due to a simulated sensor fault, the network's temporal memory correctly rejects the outlier, keeping the prediction (blue cross) firmly anchored to the true future position (green star).

While the Azimuth and Range coordinates experienced a physiological increase in error during the most severe Spikes scenarios (reaching 65.58 mrad and 9.74 m, respectively), these values remain remarkably low considering the massive magnitude of the injected perturbations. Crucially, the network avoids the catastrophic divergence typically observed in standard model-based filters under such extreme non-Gaussian conditions.

Furthermore, under the State-Dependent degradation profile—where noise variance scales quadratically with the target's range to emulate the real radar equation—the performance penalty was minimal across all variables (e.g., 45.68 mrad for Elevation and 6.26 m for Range). This confirms the model's capacity to generalize effectively even when the signal-to-noise ratio naturally deteriorates at greater observation distances.

Table 4. Quantitative performance across different noise profiles. The table compares the mean prediction RMSE across the four stress-testing scenarios for Elevation, Azimuth, and Range. The stability of the error metrics confirms the model's robustness to various sensor perturbations.

Applied Noise Profile	Elevation (mrad)	Azimuth (mrad)	Range (m)
Standard (Baseline)	45.19	47.66	5.02
Spikes (Impulsive)	40.70	65.58	9.74
Transient	41.31	56.71	7.40
State-Dependent	45.68	51.86	6.26

3.4. Scope and Limitations of the Synthetic Benchmark

A major limitation of this study is the absence of validation on real radar measurements. Access to raw operational SST radar data, especially data containing uncalibrated anomalies such as false echoes, clutter, jamming-like transients, or hardware-induced spikes, is typically restricted by institutional and security constraints. Therefore, the present work should be interpreted as a controlled simulation benchmark rather than a complete operational validation.

The use of synthetic data nevertheless provides two important advantages: first, it gives access to an exact noise-free ground truth, which is generally unavailable in real radar observations; second, it allows controlled stress testing under repeatable non-Gaussian perturbations. However, synthetic noise profiles cannot fully reproduce all effects present in operational radar chains, including calibration errors, missed detections, target scintillation, atmospheric propagation, beam-shape effects, false alarms, and data-association ambiguities.

For this reason, the proposed GRU-CNN model is best interpreted as a learned one-step tracklet filter validated under physically motivated simulated radar conditions. Demonstrating transferability to operational radar measurements remains an essential next step and will require either access to real radar campaigns or high-fidelity sensor-in-the-loop simulation.

4. Conclusions and Future Works

In this work, we presented a hybrid deep learning framework for space debris trajectory reconstruction, designed to overcome the limitations of traditional filtering techniques in processing sparse and noisy radar tracklets. The proposed architecture, integrating Gated Recurrent Units (GRUs) with 1D Convolutional Neural Networks (CNNs), effectively learns nonlinear orbital dynamics while filtering heavy non-Gaussian noise. Crucially, the adoption of physically aware preprocessing strategies specifically Residual Learning for Range and Relative Pivoting for Azimuth proved decisive in handling the extreme scale disparities and unbounded nature of orbital measurements.

Experimental results on synthetic LEO datasets validate the effectiveness of the approach. Specifically, the system achieves:

- Superior Accuracy: The model significantly outperforms the Unscented Kalman Filter (UKF), reducing the Root Mean Square Error (RMSE) by over 96% in Range estimation (from 44.0 m to 1.7 m) and demonstrating superior precision in angular tracking.
- Immediate Convergence: Unlike the UKF, which exhibits a substantial initialization transient (warm-up) to minimize error covariance, the neural network provides stable and accurate predictions from the very first output window, a critical advantage for handling short and fragmented radar tracklets.
- Robustness: The modular design, dedicating separate networks to Azimuth, Elevation, and Range, ensured consistent performance across varying noise levels and orbital geometries without the need for complex multivariate coupling.
- Invariance to Anomalies: The network demonstrated exceptional robustness against severe, unpredictable sensor perturbations. Stress tests involving simulated radar spikes and transient jamming resulted in negligible error degradation across all tracked coordinates, proving the model's ability to effectively reject non-physical artifacts.

While the current study focused on next-step prediction ($m = 1$) using a sliding window of 25 points, the architecture's recurrent nature suggests strong potential for extended forecasting.

Future developments will focus on three main directions. First, we aim to validate the model on real radar measurement data to assess its transferability from synthetic environments to operational sensors with real-world artifacts (e.g., clutter, false alarms). Second, we plan to explore multi-step forecasting horizons to predict longer trajectory segments, which is essential for re-acquisition and collision avoidance maneuvers. Finally, we intend to investigate advanced architectures such as Transformers or attention-based mechanisms to further capture long-term dependencies and improve computational efficiency for real-time onboard implementation.

In conclusion, this study highlights that domain-aware hybrid deep learning models offer a promising, flexible, and high-performance alternative to classical model-based filters for the challenging task of space debris surveillance.

Author Contributions: Conceptualization, A.C. and N.P.; methodology, N.P.; software, A.C.; validation, A.C. and N.P.; formal analysis, N.P. and V.M.-P.; investigation, N.P.; resources, N.P.; data curation, N.P. and V.M.-P.; writing—original draft preparation, A.C. and N.P.; writing—review and editing, A.C., N.P. and D.B.; visualization, A.C.; supervision, D.B., J.S. and M.A.; project administration, A.C. and N.P.; funding acquisition, A.C. All authors have read and agreed to the published version of the manuscript.

Funding: This project was partially funded by the National Institute for Astrophysics (INAF), Astronomical Observatory of Cagliari, and by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) GRK 2680–Project-ID 437847244.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

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

Abbreviations

The following abbreviations are used in this manuscript:

AI	Artificial Intelligence
CNN	Convolutional Neural Network

CPU	Central Processing Unit
DL	Deep Learning
ECI	Earth-Centered Inertial
EDA	Exploratory Data Analysis
EKF	Extended Kalman Filter
ESA	European Space Agency
GEO	Geostationary Earth Orbit
GNSS	Global Navigation Satellite System
GPU	Graphics Processing Unit
GRU	Gated Recurrent Units
LEO	Low Earth Orbit
LR	Learning Rate
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error
MEO	Medium Earth Orbit
MLP	Multi Layer Perceptron
MSE	Mean Squared Error
NN	Neural Network
PDF	Probability Density Function
PF	Particle Filters
RAM	Random Access Memory
ReLU	Rectified Linear Unit
RMSE	Root Mean Square Error
RNN	Recurrent Neural Network
SGD	Stochastic Gradient Descent
SGP4	Simplified General Perturbations No. 4
SNR	Signal-to-Noise Ratio
SST	Space Surveillance and Tracking
TIRA	Tracking & Imaging Radar
TLE	Two-Line Elements
UHF	Ultra High Frequency
UKF	Unscented Kalman Filter

References

1. O'Neill, A.; Barber, D.; Bauer, P.; Dahlin, H.; Diament, M.; Hauglustaine, D.; Le Traon, P.-Y.; Mattia, F.; Mauser, W.; Merchant, C.; et al. Earth observation science strategy for ESA: A new era for scientific advances and societal benefits. In *Earth Observation Science Strategy*; European Space Agency: Noordwijk, NL, USA, 2015; Volume SP-1329.
2. ESA Space Debris Office. *ESA's Annual Space Environment Report*; Technical Report; European Space Agency: Paris, France, 2023.
3. Mehrholz, D.; Leushacke, L.; Flury, W.; Jehn, R.; Klinkrad, H.; Landgraf, M. Detecting, tracking and imaging space debris. *ESA Bull.* **2002**, *109*, 128–134.
4. Muntoni, G.; Montisci, G.; Pisanu, T.; Andronico, P.; Valente, G. Crowded Space: A Review on Radar Measurements for Space Debris Monitoring and Tracking. *Appl. Sci.* **2021**, *11*, 1364. [CrossRef]
5. Levanon, N. *Radar Principles*; Wiley-Interscience: Hoboken, NJ, USA, 1988.
6. Vallado, D.A. *Fundamentals of Astrodynamics and Applications*, 4th ed.; Microcosm Press: Portland, OR, USA, 2013.
7. Steindorfer, M.A.; Wang, P.; Koidl, F.; Kirchner, G. Space debris and satellite laser ranging combined using a megahertz system. *Nat. Commun.* **2025**, *16*, 575. [CrossRef] [PubMed]
8. Suchodolski, T.; Lejba, P.; Michałek, P.; Schillak, S.; Wnuk, E. Active Control Loop of the BOROWIEC SLR Space Debris Tracking System. *Sensors* **2022**, *22*, 2231. [CrossRef] [PubMed]
9. Blackman, S.S.; Popoli, R. *Design and Analysis of Modern Tracking Systems*; Artech House: Norwood, MA, USA, 1999.
10. Bar-Shalom, Y.; Li, X.R.; Kirubarajan, T. *Estimation with Applications to Tracking and Navigation*; Wiley: Hoboken, NJ, USA, 2001.
11. Lawton, J.; Jesionowski, R.; Zarchan, P. Comparison of Four Filtering Options for a Radar Tracking Problem. *J. Guid. Control Dyn.* **1998**, *21*, 618–623. [CrossRef] [PubMed]
12. Julier, S.; Uhlmann, J. Unscented filtering and nonlinear estimation. *Proc. IEEE* **2004**, *92*, 401–422. [CrossRef]
13. Simon, D. *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*; Wiley-Interscience: Hoboken, NJ, USA, 2006.

14. Wan, E.; Van Der Merwe, R. The unscented Kalman filter for nonlinear estimation. In Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium, Lake Louise, AB, Canada, 4 October 2000; pp. 153–158. [CrossRef]
15. Jones, B.; Doostan, A.; Born, G.H. Nonlinear Propagation of Uncertainties in Orbit Determination Using the Unscented Transformation. *J. Guid. Control Dyn.* **2018**, *41*, 1–15. [CrossRef]
16. Arulampalam, M.S.; Maskell, S.; Gordon, N.; Clapp, T. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Trans. Signal Process.* **2002**, *50*, 174–188. [CrossRef]
17. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
18. Cho, K.; van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1724–1734. [CrossRef]
19. Peng, H.; Bai, X. Artificial Neural Network–Based Machine Learning Approach to Improve Orbit Prediction Accuracy. *J. Spacecr. Rockets* **2018**, *55*, 1248–1260. [CrossRef]
20. Zhang, W.; Zhang, K.; Li, X.; Huang, J. Improving LEO short-term orbit prediction using LSTM neural network. *Adv. Space Res.* **2025**, *76*, 481–496. [CrossRef]
21. Rock, J.; Toth, M.; Messner, E.; Meissner, P.; Pernkopf, F. Complex Signal Denoising and Interference Mitigation for Automotive Radar Using Convolutional Neural Networks. In Proceedings of the 22nd International Conference on Information Fusion (FUSION), Ottawa, ON, Canada, 2–5 July 2019; pp. 1–8. [CrossRef]
22. Liu, H.; Zhang, H.; Mertz, C. DeepDA: LSTM-Based Deep Data Association Network for Multi-Targets Tracking in Clutter. In Proceedings of the 22nd International Conference on Information Fusion (FUSION), Ottawa, ON, Canada, 2–5 July 2019; pp. 1–8. [CrossRef]
23. Zhang, Z.; Zhang, Y.; Liu, W.; Liu, C.; Liu, T. SD-DETR: Space Debris Detection Transformer Based on Dynamic Convolutional Network and Cross-Scale Collaborative Attention. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2025**, *18*, 25845–25862. [CrossRef]
24. Raimondi, M.; Ciattaglia, G.; Nocera, A.; Gardano, M.; Senigagliesi, L.; Spinsante, S.; Gambi, E. mmTracking: A DL-Based mmWave RADAR Data Processing Algorithm for Indoor People Tracking. *IEEE Sens. J.* **2025**, *25*, 45071–45083. [CrossRef]
25. Wang, C.; Zhang, J.; Wang, J.; Wu, Y. Confidence-Based Fusion of AC-LSTM and Kalman Filter for Accurate Space Target Trajectory Prediction. *Aerospace* **2025**, *12*, 347. [CrossRef]
26. Saccon, C.; Pilloni, N. On an analytical method for the satellite problem revisited. *arXiv* **2022**, arXiv:2212.14243.
27. Miller, J.K. *Planetary Spacecraft Navigation*; Springer Nature: Berlin/Heidelberg, Germany, 2019.
28. Givental, A. Kepler's Laws and Conic Sections. *Arnold Math. J.* **2015**, *2*, 139–148. [CrossRef]
29. Klare, J.; Behner, F.; Carloni, C.; Cerutti-Maori, D.; Fuhrmann, L.; Hoppenau, C.; Karamanavis, V.; Laubach, M.; Marek, A.; Perkuhn, R.; et al. The Future of Radar Space Observation in Europe—Major Upgrade of the Tracking and Imaging Radar (TIRA). *Remote Sens.* **2024**, *16*, 4197. [CrossRef]
30. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778. [CrossRef]
31. Hyndman, R.J.; Athanasopoulos, G. *Forecasting: Principles and Practice*, 2nd ed.; OTexts: Melbourne, Australia, 2018.
32. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef] [PubMed]
33. Rumelhart, D.E.; Hinton, G.E.; Williams, R.J. Learning Internal Representations by Error Propagation. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume 1: Foundations*; MIT Press: Cambridge, MA, USA, 1986; pp. 318–362.
34. Chung, J.; Gulcehre, C.; Cho, K.; Bengio, Y. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *arXiv* **2014**, arXiv:1412.3555.
35. Xie, Y.; Wang, J.; Li, H.; Dong, A.; Kang, Y.; Zhu, J.; Wang, Y.; Yang, Y. Deep Learning CNN-GRU Method for GNSS Deformation Monitoring Prediction. *Appl. Sci.* **2024**, *14*, 4004. [CrossRef]
36. Tabassum, T.E.; Xu, Z.; Petrunin, I.; Rana, Z.A. Integrating GRU with a Kalman Filter to Enhance Visual Inertial Odometry Performance in Complex Environments. *Aerospace* **2023**, *10*, 923. [CrossRef]
37. Willmott, C.; Matsuura, K. Advantages of the Mean Absolute Error (MAE) over the Root Mean Square Error (RMSE) in Assessing Average Model Performance. *Clim. Res.* **2005**, *30*, 79. [CrossRef]
38. Williams, R.J.; Zipser, D. A Learning Algorithm for Continually Running Fully Recurrent Neural Networks. *Neural Comput.* **1989**, *1*, 270–280. [CrossRef]
39. Bergstra, J.; Bengio, Y. Random search for hyper-parameter optimization. *J. Mach. Learn. Res.* **2012**, *13*, 281–305.

40. Li, L.; Jamieson, K.; DeSalvo, G.; Rostamizadeh, A.; Talwalkar, A. Hyperband: A novel bandit-based approach to hyperparameter optimization. *J. Mach. Learn. Res.* **2017**, *18*, 6765–6816.
41. Cabras, A. *Creating a Docker Environment for Jupyter Notebook-Based Machine Learning Projects*; Italian National Institute for Astrophysics (INAF): Rome, Italy, 2024. [CrossRef]