

A Target Tracking Model for Weak Radar Targets Based on Track-Before-Detect

Ben Zhang

University of Chinese Academy of Sciences

Ying Liu

yingliu@ucas.ac.cn

University of Chinese Academy of Sciences

Xiaoying Chen

University of Chinese Academy of Sciences

Cheng Wang

Beijing Raying Technology, Co., Ltd.

Research Article

Keywords: radar target tracking, low-SNR, deep learning, track-before-detect, ConvLSTM, U-Net

Posted Date: March 25th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-9211452/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: The authors declare no competing interests.

A Target Tracking Model for Weak Radar Targets Based on Track-Before-Detect

Ben Zhang, Ying Liu^{*}, and Xiaoying Chen
University of Chinese Academy of Sciences
Beijing 100049, China
zhangben23@mails.ucas.ac.cn

Cheng Wang
Beijing Raying Technology, Co., Ltd.
Beijing 100080, China
jason_wang@rayinglab.com

Abstract—Radar target tracking under low signal-to-noise ratio (low-SNR) conditions remains a critical challenge. For targets such as stealth aircraft and low-altitude unmanned aerial vehicles (UAVs), radar echoes are often extremely weak and submerged in noise, rendering conventional detection and tracking ineffective. To address this issue, we propose a method that combines a Convolutional Long Short-Term Memory (ConvLSTM) network, U-Net, and the dynamic-programming track-before-detect (DP-TBD) algorithm. Specifically, DP-TBD generates multi-frame accumulation maps, which are fed into the network. U-Net extracts spatial features from each accumulation map, ConvLSTM models temporal dependencies over the accumulation-map sequence, and a Squeeze-and-Excitation (SE) module suppresses noise-dominated channels. The network outputs a final-frame confidence map for thresholding and track extraction. Experiments show that the proposed method improves tracking performance in low-SNR environments, achieving a 90% tracking rate with a false alarm rate below 10^{-5} at an RD-map SNR of 3 dB.

Index Terms—radar target tracking, low-SNR, deep learning, track-before-detect, ConvLSTM, U-Net

I. INTRODUCTION

In surveillance scenarios involving stealth aircraft and low-flying unmanned aerial vehicles (UAVs), targets usually have very small radar cross sections (RCS), resulting in echo signals much weaker than that of the background noise and clutter. It creates a typical low-SNR environment, where the SNR, defined as the ratio of the target signal power to the background noise power, is low. Under such conditions, traditional radar signal processing algorithms struggle to distinguish targets from noise, where a low detection threshold causes many false alarms, while a high threshold leads to missed detections. The root of this problem is that traditional radar target tracking typically follows a Detect-Before-Track (DBT) approach, which first detects potential targets in each frame (e.g., using CFAR) and then links these detections over time using tracking algorithms like Probabilistic Data Association (PDA) and Kalman filtering (KF). However, in low-SNR conditions, single-frame detection often fails, producing either too many false alarms or missing the target entirely, which makes the subsequent tracking stage fail. To overcome this problem, Track-Before-Detect (TBD) technology was developed [1],

[2]. TBD accumulates signal energy over multiple frames based on potential target trajectories. Since the signals from targets tend to be consistent over time while noise is random, accumulation enhances the signal of the target relative to the noise. Classic TBD methods, such as Dynamic Programming TBD (DP-TBD), are effective but have limitations. They rely on specific prior assumptions about target motion patterns and clutter/noise distributions. If the actual scenario differs from the predefined assumptions (e.g., unexpected maneuvers or complex clutter), their performance drops significantly [3], [4]. Moreover, their computational complexity increases rapidly with the complexity of scenarios and model assumptions [5].

Recently, deep learning has been introduced to address these limitations. Deep learning models can learn features directly from data, making them more robust to complex environments [6]. Therefore, this paper proposes a deep learning-enhanced TBD model called CL-UNet-TBD. Experiments show that our proposed model performs well in low-SNR tracking tasks. Specifically, under 3 dB Range-Doppler (RD) map conditions, it maintains a tracking rate of over 90% while keeping the false alarm rate below 10^{-5} .

The main contributions of this paper are summarized as follows:

- A novel deep learning-enhanced TBD framework is proposed. Instead of replacing the traditional TBD process entirely or using raw data directly, we utilize the multi-frame accumulation maps generated by DP-TBD as the input for the deep network. This strategy effectively combines the interpretability of classical methods with the feature learning capability of deep learning.
- A specialized spatiotemporal network architecture is designed for weak target tracking. We integrate U-Net with Convolutional Long Short-Term Memory (ConvLSTM), where U-Net is employed to extract multi-scale spatial features from the accumulation maps, and ConvLSTM is used to capture the temporal motion patterns of targets across frames, ensuring robust tracking of moving targets.
- A Squeeze-and-Excitation (SE) module is embedded into the network to adaptively recalibrate channel-wise feature responses. This allows the model to focus on target-related features while significantly suppressing noise and clutter, thereby improving the detection performance in low-SNR environments.

This work was supported in part by the National Natural Science Foundation of China under Grant #62176247 and the Fundamental Research Funds for the Central Universities. (Corresponding author: Ying Liu, email: yingliu@ucas.ac.cn)

The rest of this paper is organized as follows: Section II reviews related work. Section III describes the proposed CL-UNet-TBD model. Section IV presents the experimental setup and results. Section V concludes the paper.

II. RELATED WORK

A. Model-Based Track-Before-Detect Algorithms

While the DBT approach is reliable in high-SNR environments, it struggles with weak targets. To address this limitation, various TBD algorithms have been proposed.

Hough Transform TBD (HT-TBD) uses the Hough transform to accumulate energy. It maps detection points, obtained by applying a low threshold detection, to a parameter space. Points conforming to a specific motion model intersect at a single point in the parameter space, forming detectable peaks [1]. While robust to noise and data gaps, it is computationally expensive for complex motion models and relies heavily on prior motion assumptions, limiting its performance for maneuvering targets (e.g., aircraft executing sharp turns).

Particle Filter TBD (PF-TBD) represents the target state using a set of weighted particles, where target state evolution is reflected by weight updates. The algorithm follows a “predict-update-resample” sequence: it predicts the state of each particle based on the motion model, updates weights according to the likelihood of radar observations, and resamples to retain high-weight particles [7]. Particles matching the true trajectory accumulate higher weights over time, enabling detection. Although PF-TBD handles non-linear and non-Gaussian scenarios effectively, it requires a massive number of particles to avoid particle degeneracy in low-SNR or multi-target environments, leading to high computational costs.

Dynamic Programming TBD (DP-TBD) is a widely used method that constructs a frame-recursive value function on discrete grids (e.g., Range-Doppler cells). Under specific state transition constraints, the method updates the value function frame-by-frame using dynamic programming. Finally, a threshold is applied to the accumulated value function at the last frame, and target trajectories are retrieved by backtracking [2], [8]. DP-TBD is computationally efficient and easy to implement but depends heavily on the correctness of the target motion and signal-noise-clutter models.

Other notable TBD methods include Random Finite Set (RFS) based TBD [9], 3D Matched Filter TBD [10], [11], and Information Geometry based TBD [12].

B. Deep Learning in Radar Target Tracking

Existing deep learning applications in radar target tracking are mainly divided into three categories:

Deep Learning Assisted DBT primarily employs deep neural networks to optimize specific stages within the DBT pipeline. In the detection stage, DAROD [13] adapts the Faster R-CNN framework to replace the traditional CFAR detector, mitigating the information loss of hard thresholding to enhance robustness in clutter-rich environments. In the tracking stage, Long Short-Term Memory (LSTM) networks [14] are employed to predict and smooth target states, substituting the

function of Kalman filters. Deep networks have also been integrated with classical filtering: DeepMTT [15] uses a bi-directional LSTM to predict the residuals between Unscented Kalman Filter (UKF) estimates and true states, correcting these residuals to obtain more precise estimates for maneuvering targets; NNAKF [16] employs RNNs to dynamically predict the process noise covariance of the Kalman filter; and Kalman-Net [17] learns the Kalman gain directly from data, bypassing the complex gain calculations. While these methods enhance the tracking robustness in moderately low-SNR scenarios, they remain fundamentally constrained by frontend single-frame thresholding.

Deep Learning Enhanced TBD addresses the limitations of traditional TBD methods, such as heavy reliance on prior assumptions and high computational complexity in complex scenarios. The DP-TBD framework, known for its clear structure and extensibility, has become an ideal candidate for deep learning integration. This integration is typically applied to two key stages of the pipeline. The first stage involves estimating the confidence of target presence at each position using single-frame observations, aiming to maximize the distinction between targets and clutter/noise. Instead of classical Bayesian or maximum likelihood estimation, deep learning methods such as deep neural networks [4], Convolutional Denoising Autoencoders (CDA) [18], and 3D U-Net [19] are employed to generate confidence maps. The second stage focuses on determining the transition range or value function update rules based on target motion. Unlike classical methods that assume predefined motion models, deep learning approaches learn motion patterns directly from data to narrow the search space. For instance, Generative Adversarial Networks (GANs) have been used to learn potential target regions in RD maps [6]; Transformer-based networks have been applied to predict future target ranges from historical trajectories [5]. However, recent research has primarily focused on maneuvering target tracking, where limited attention has been given to tracking in lower SNR environments.

End-to-End Tracking attempts to learn tracking rules directly from raw radar echoes, unlike DBT and TBD algorithms, which typically use preprocessed Range-Doppler or Range-Azimuth maps as network inputs. For example, a cross-attention Transformer framework [20] has been proposed to achieve end-to-end coherent detection under low-SNR conditions. While this paradigm simplifies the traditional processing pipeline and avoids extensive manual feature design, it typically suffers from weak interpretability, strong dependence on large-scale high-quality labeled data, and difficulties in transferring across different radar systems or scenarios.

III. METHOD

The proposed CL-UNet-TBD model falls within the paradigm of deep learning-enhanced TBD. As shown in Fig. 1, the model consists of three modules: the multi-frame accumulation module, the ConvLSTM-UNet network, and the track extraction module. A sequence of radar RD maps serves as the input. Firstly, the multi-frame accumulation module generates

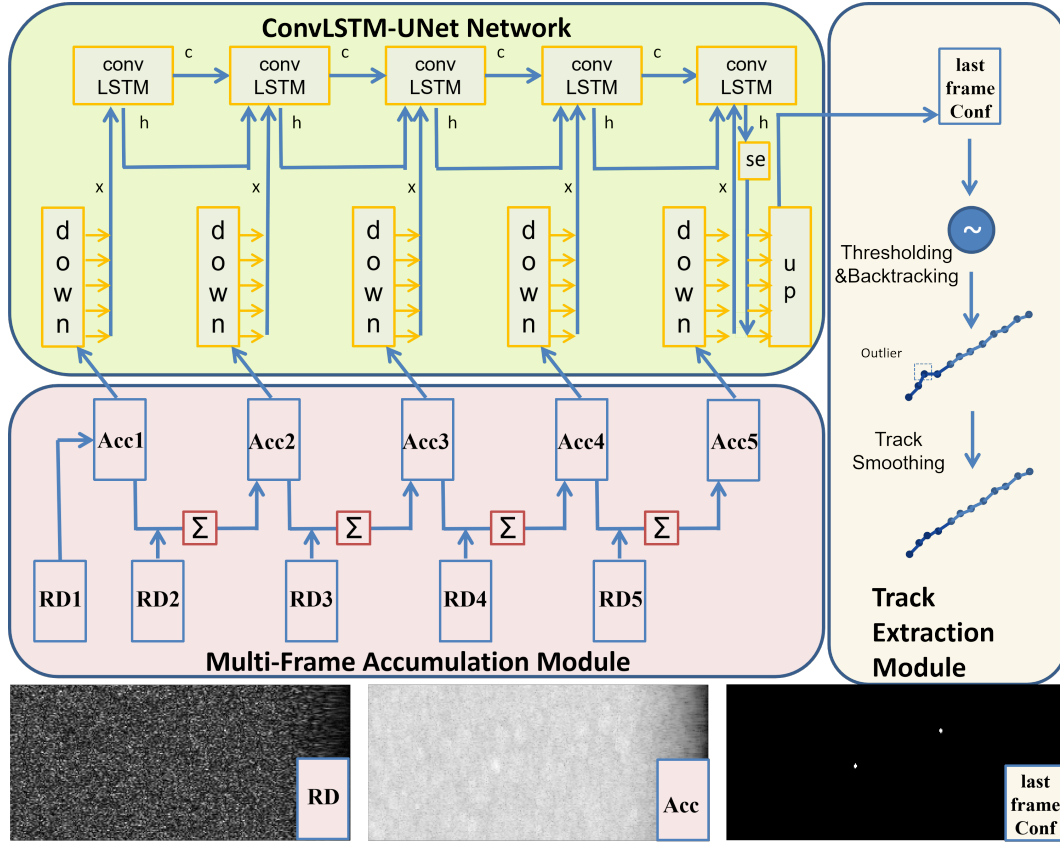


Fig. 1. Overall architecture of the CL-UNet-TBD method.

per-frame accumulation maps, which are then processed by the ConvLSTM-UNet network to extract spatiotemporal features and produce a final-frame confidence map indicating the potential target locations. Finally, the track extraction module performs thresholding on the given map and retrieves the target trajectory through backtracking.

This design effectively leverages deep learning’s powerful data-driven denoising capability and spatiotemporal feature extraction capability. The sequence of accumulation maps contains rich spatiotemporal features essential for detection. Spatially, unlike point targets that occupy few pixels in raw RD maps, accumulation maps exhibit distinct distribution patterns distinguishable from noise. Temporally, target motion creates consistent sequential correlations across the accumulation map sequence. The classical DP-TBD model, relying on thresholds and local maxima, often fails to exploit these features, especially in low-SNR environments where strong noise peaks cause false alarms and at the same time reduce the contrast between accumulated target signals and background noise. In contrast, the proposed CL-UNet-TBD network achieves superior performance by effectively extracting these spatiotemporal features to isolate targets from background noise.

A. Multi-Frame Accumulation Module

This module enhances target energy by accumulating energy across multiple frames, allowing the ordered target signals to

emerge from the disordered noise. In this module, a simplified version of the DP-TBD algorithm is implemented [2], using the amplitude of the RD map as the per-state observation score that serves as the likelihood proxy which is recursively accumulated to form the dynamic-programming value function, without relying on complex priors of noise and clutter distributions.

Target state is defined as $x_k = [r_k, v_k]$, where r_k and v_k denote the radial range and velocity at frame k . The observation at frame k , denoted as Z_k , is an $m \times n$ RD map where each pixel $z_{i,j}$ denotes the signal intensity at the corresponding range and Doppler bin.

Our goal is to recover target trajectories from the observation sequence $Z_{1:K}$. A common probabilistic interpretation of DP-TBD is to view the problem as a MAP estimation under a first-order Markov model, which yields a recursion of the form

$$P(x_k | Z_{1:k}) \propto P(Z_k | x_k) \int P(x_k | x_{k-1}) P(x_{k-1} | Z_{1:k-1}) dx_{k-1}. \quad (1)$$

However, in our implementation we do not explicitly model these probabilities. Instead, we adopt a score-based dynamic programming accumulation on the discrete RD grid. We define an accumulated score $I_k(x)$ for each state x at frame k , approximate the integral by a maximization over a constrained

transition neighborhood, and obtain

$$I_k(x) = Z_k(x) + \max_{x' \in \Omega(x)} I_{k-1}(x') \quad (2)$$

where $Z_k(x)$ is an empirical observation score, defined as the observed RD-map amplitude at state x in frame k in our experiments. $\Omega(x)$ is the transition neighborhood defined by the maximum possible displacement between frames: for $x = (r, v)$, $\Omega(x) = \{x' = (r', v') : |r - r'| \leq \Delta R, |v - v'| \leq \Delta V\}$. This recursion produces a sequence of accumulation maps $I_{1:K}$ that serve as the input to the ConvLSTM-UNet network.

The pseudo-code for the multi-frame accumulation module is presented in Algorithm 1.

Algorithm 1: Multi-Frame Accumulation Algorithm

Input: Sequence of radar measurements

$$Z_{1:K} = \{Z_1, Z_2, \dots, Z_K\}$$

Output: Sequence of accumulated value functions

$$I_{1:K} = \{I_1, I_2, \dots, I_K\}$$

```

1 Initialize:  $I_1(x) = Z_1(x)$  for all states  $x$ ;
2 for  $k = 2$  to  $K$  do
3   for all states  $x$  do
4     Compute maximum value from previous frame
       within transition neighborhood
        $V_{prev}(x) = \max_{x' \in \Omega(x)} I_{k-1}(x')$ ;
5     Update value function for the current frame
        $I_k(x) = Z_k(x) + V_{prev}(x)$ ;
6 return  $I_{1:K}$ 

```

B. ConvLSTM-UNet Network

The input to the ConvLSTM-UNet network is the sequence of the accumulation maps generated by the previous module, and the output is the confidence map for the final frame. By integrating U-Net [21], a dominant model in image segmentation, with ConvLSTM [22], which has proven powerful for spatiotemporal modeling, our proposed network possesses a strong capability to mine small moving targets from the sequence of accumulation maps. The network consists of a series of cascaded ConvLSTM-UNet units, as shown in Fig. 2. The unit comprises four components: an encoder, ConvLSTM modules, SE modules, and a decoder. However, for non-final time steps which are solely responsible for feature state propagation, the SE and decoder modules are omitted.

1) *U-Net with ConvLSTM*: The encoder (downsampling path) consists of four blocks, each containing a 2×2 max pooling layer followed by two 3×3 convolutions with padding, each followed by Batch Normalization and ReLU activation. The decoder (upsampling path) also comprises four blocks. Firstly it upsamples features using bilinear interpolation, concatenates them with the corresponding skip features, and then fuses them through two 3×3 convolutions, each followed by Batch Normalization and ReLU activation.

A key design element is the integration of ConvLSTM into both of the skip connections and the bottleneck layer. At

each scale and time step, the ConvLSTM module receives the downsampled features of the current frame and the hidden state from the previous time step as the inputs. Its output hidden features are propagated to the ConvLSTM of the next time step. At the final time step, instead of further propagation, these features serve as the skip features for the current scale's decoder and as the bottleneck features at the coarsest scale. Such a design enables the network to leverage the target signal features accumulated from history to enhance the features of the target and suppress strong noise.

2) *SE Channel Attention*: As the depth of U-Net increases, the feature channels may include not only the target-related information but also the features from the noise. In our architecture, noise channels can impact the final tracking performance negatively. Therefore, an SE module is introduced after the ConvLSTM of the final time step to recalibrate the channel weights. Specifically, the ConvLSTM output undergoes adaptive global average pooling, followed by a dimensionality reduction and an expansion via two 1×1 convolutions. Next, a Sigmoid function generates channel weights to perform channel-wise weighting on the features. Such a mechanism ensures that at every scale, output of ConvLSTM is recalibrated to emphasize target-dominant channels and suppress the noise-dominated ones.

C. Track Extraction Module

This module takes the final-frame confidence map P_K from the ConvLSTM-UNet network as the input to identify track endpoints via thresholding and recover the full trajectories using backtracking.

1) *Adaptive Thresholding and Peak Extraction*: Let $\mathcal{S}$ denote the set of detected track endpoints. To ensure robustness against varying noise levels, we first compute a quantile-based baseline threshold $q = \text{Quantile}(P_K, 0.9995)$. Then, we construct a set of candidate thresholds $\mathcal{T} = \{\max(\tau_i, q) \mid \tau_i \in \{0.5, 0.3, 0.2, 0.1, 0.05\}\}$ and iteratively test them from strict to loose until a non-empty set of peaks is detected. A pixel (r, v) is considered a candidate peak if $P_K(r, v) > \tau$ ($\tau \in \mathcal{T}$) and it is a local maximum within a neighborhood of size d . To handle flat peaks, we aggregate candidate pixels into connected components and retain only the coordinate with the maximum confidence in each component as a valid endpoint in $\mathcal{S}$.

2) *Backtracking*: During the accumulation process, the multi-frame accumulation module records the optimal transition path in a matrix $\mathbf{S}$, where $\mathbf{S}[k, r_k, v_k] = (r_{k-1}, v_{k-1})$ represents the optimal predecessor state of the state (r_k, v_k) . For each detected endpoint $(r_K, v_K) \in \mathcal{S}$, the full trajectory is reconstructed recursively:

$$(r_{k-1}, v_{k-1}) = \mathbf{S}[k, r_k, v_k], \quad k = K, K-1, \dots, 2 \quad (3)$$

It yields a set of discrete trajectories $\mathcal{X} = \{\{(r_k^{(m)}, v_k^{(m)})\}_{k=1}^K\}_{m=1}^M$, where M is the number of detected trajectories.

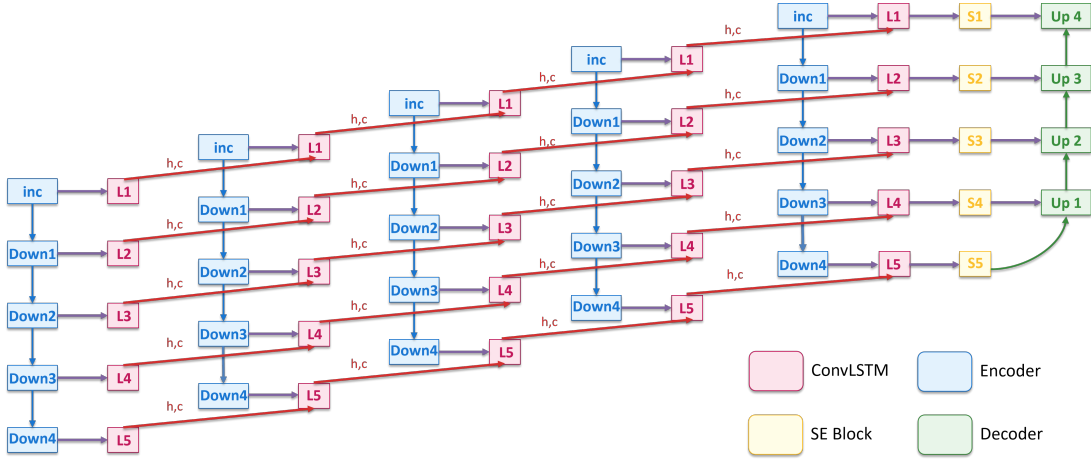


Fig. 2. Structure of the ConvLSTM-UNet network.

3) *Smoothing*: Finally, to correct minor errors caused by noise or sidelobes, a sliding window mode filter is applied to the extracted tracks. For each time step, the range and Doppler indices are updated to the most frequent values within a window of size N centered at the given step. If there are multiple modes, the value closest to the current time step is selected.

IV. EXPERIMENTS

A. Dataset

Our proposed model is evaluated on a simulated dataset designed for low-SNR radar weak target tracking. We control the SNR level by adding complex Gaussian white noise to the clean echoes. The raw echoes are processed via Pulse Compression (PC) and Moving Target Detection (MTD) to generate the Range-Doppler (RD) maps used as the input. We generated datasets at 10 distinct SNR levels: RD-map SNR $\in \{15, 13, \dots, -3\}$ dB, corresponding to raw echo SNR $\in \{-26, -28, \dots, -44\}$ dB. Each SNR level contains 20 scenarios, with each scenario consisting of 100 consecutive RD frames. Each scenario contains two targets with initial ranges in $[600, 1800]$ m and radial velocities in $[-300, 300]$ m/s. The simulation parameters are detailed in Table I.

TABLE I
SIMULATION PARAMETERS

Parameter	Value
Sampling Rate	20 MHz
Pulse Repetition Frequency	20 kHz
Carrier Frequency	15 GHz
Coherent Processing Interval (CPI) Pulses	128
Gate Width	16 μ s
RD Map Size	320 $\times$ 128
Range Resolution	7.5 m
Velocity Resolution	1.5625 m/s

B. Evaluation Metrics

Four metrics are employed to assess tracking performance: Detection Probability (P_d), Track Detection Probability

(P_{dtrack}), Multiple Object Tracking Accuracy (MOTA) and Probability of False Alarm (PFA). A prediction ($\hat{r}, \hat{v}$) is considered a successful match to a ground truth (r, v) (where r and v denote the range and Doppler indices on the RD map) if the error in both dimensions is within one cell ($|\hat{r} - r| \leq 1 \wedge |\hat{v} - v| \leq 1$).

1) *Detection Probability (P_d)*: P_d is the ratio of successfully matched detections to the total number of ground truth targets: $P_d = N_{TP}/N_{GT}$, where N_{TP} is the total true positives across all frames and N_{GT} is the total number of ground truth.

2) *Track Detection Probability (P_{dtrack})*: A ground truth track Γ_u is considered successfully detected if at least 80% of its points are matched by a single predicted track ($N_{TP}(\Gamma_u)/N(\Gamma_u) \geq 0.8$). P_{dtrack} is the ratio of successfully detected tracks over the total number of ground truth tracks.

3) *MOTA*: A comprehensive measure of the tracking accuracy:

$$MOTA = 1 - \frac{N_{miss} + N_{fp} + N_{idsw}}{N_{GT}} \quad (4)$$

where N_{miss} , N_{fp} , and N_{idsw} denote the number of misses, false positives, and ID switches, respectively.

4) *Probability of False Alarm (PFA)*: PFA is defined as the probability that a detection cell is misclassified as a target, as shown in (5):

$$PFA = \frac{N_{fp}}{N_{total_cells}}, \quad (5)$$

where N_{fp} is the number of false positives, and N_{total_cells} is the total number of range-Doppler cells across all frames.

C. Ablation Study

We evaluate the contribution of each component by comparing our Proposed Model with four variants. Fig. 3 illustrates the performance curves across different SNR levels, and the detailed results at 3 dB RD-map SNR (corresponding to -38 dB raw echo SNR) are summarized in Table II.

Necessity of Temporal Propagation: First, removing cross-frame propagation (*No Cross-Frame Prop.*) leads to a catastrophic performance drop across all metrics. For instance, at

TABLE II
ABLATION RESULTS AT LOW-SNR (3 dB RD-MAP SNR,
CORRESPONDING TO -38 dB RAW ECHO SNR)

Setting	$P_d(\%) \uparrow$	$P_{dtrack}(\%) \uparrow$	PFA(10^{-6}) $\downarrow$	MOTA $\uparrow$
Skip+Bottleneck+SE	93.65	90.00	3.96	0.855
Skip Only+SE	91.18	87.50	2.48	0.861
Bottleneck Only+SE	81.00	67.50	3.05	0.747
Skip+Bottleneck (No SE)	91.00	87.50	2.69	0.855
No Cross-Frame Prop.+SE	84.28	72.50	7.80	0.683

3 dB, P_{dtrack} drops significantly by 17.5% (from 90.00% to 72.50%). Simultaneously, PFA nearly doubles, rising from 3.96×10^{-6} to 7.80×10^{-6} . This confirms that learning temporal consistency from historical accumulation frames is the cornerstone of our method, crucial for both improving detection recall and suppressing false alarms in noise.

Trade-off in Propagation Paths: Second, we analyze the propagation paths. The results indicate that removing either the Skip or Bottleneck branch degrades P_{dtrack} . Specifically, at 3 dB, eliminating the Skip connection (*Bottleneck Only*) leads to a drastic performance drop (from 90.00% to 67.50%), while removing the Bottleneck (*Skip Only*) causes a smaller decline (from 90.00% to 87.50%). This confirms that both paths contribute to tracking performance under low SNR, but the Skip connection, which transmits multi-level features containing high-resolution low-level semantics, plays a more dominant role than the Bottleneck path, which conveys only the deepest abstract semantics. Consequently, the Proposed Model retains both paths. Although this integration results in a slightly higher PFA (3.96×10^{-6}) compared to the single-branch variants (e.g., 2.48×10^{-6} for *Skip Only* and 3.05×10^{-6} for *Bottleneck Only*), we accept this trade-off to achieve the maximal detection rate.

Role of SE Module: Finally, we assess the effect of the SE channel attention mechanism. As summarized in Table II, removing SE (*No SE*) reduces P_{dtrack} from 90.00% to 87.50% at 3 dB. This gain is achieved at the cost of a higher false alarm probability: PFA increases from 2.69×10^{-6} (*No SE*) to 3.96×10^{-6} (Proposed Model). Meanwhile, the comprehensive metric MOTA remains at 0.855 for both variants. The curves in Fig. 3 further indicate that SE becomes more beneficial as SNR decreases: at 2 dB RD-map SNR (-40 dB raw echo SNR), it improves P_{dtrack} from 25.0% to 27.5% and also reduces PFA from 1.72×10^{-5} to 1.39×10^{-5} , underscoring its capability to improve the distinction between weak targets and background noise under severe noise conditions.

D. Comparison Study

To validate the effectiveness of our approach, we compare it against four representative methods. Fig. 4 illustrates the performance curves across different SNR levels, and the

detailed results at 3 dB RD-map SNR (corresponding to -38 dB raw echo SNR) are summarized in Table III.

TABLE III
COMPARISON RESULTS AT LOW-SNR (3 dB RD-MAP SNR,
CORRESPONDING TO -38 dB RAW ECHO SNR)

Method	$P_d(\%) \uparrow$	$P_{dtrack}(\%) \uparrow$	PFA(10^{-6}) $\downarrow$	MOTA $\uparrow$
CL-UNet-TBD	92.88	90.00	5.19	0.823
CA-CFAR+EKF	0.70	0.00	0.10	0.005
DP-TBD	70.50	42.50	14.41	0.406
MKCF-TBD	69.00	12.50	15.14	0.360
TRO-TBD	85.45	62.50	6.71	0.714

The comparison models include:

- **CA-CFAR + EKF:** A standard industrial DBT pipeline using Cell-Averaging CFAR for detection and Extended Kalman Filter for tracking [23], [24].
- **DP-TBD:** A classical Track-Before-Detect method using dynamic programming to accumulate energy across frames [2], [8].
- **MKCF-TBD:** A pseudo-spectrum-based TBD method that accumulates energy by predicting pixel locations in the final frame using a speed-squared filter [25].
- **TRO-TBD:** A recent deep learning-based TBD method utilizing a Transformer network to adaptively optimize the transition ranges [5].

Except for CA-CFAR+EKF, which operates as a single-frame detector, all the other methods are multi-frame detectors with the accumulation frame count set at 10. In terms of P_d and PFA, CL-UNet-TBD consistently outperforms the comparison methods across all SNR levels, with a particularly pronounced advantage in the low-SNR regime of 3 dB and below. At 3 dB, it achieves a P_d of 92.88%, significantly surpassing TRO-TBD (85.45%), DP-TBD (70.50%), and MKCF-TBD (69.00%), whereas CA-CFAR+EKF fails almost completely (0.70%) due to single-frame thresholding limitations. Simultaneously, CL-UNet-TBD achieves a PFA of only 5.19×10^{-6} at 3 dB, which is significantly lower than that of TRO-TBD (6.71×10^{-6}).

Regarding P_{dtrack} and the comprehensive tracking metric MOTA, CL-UNet-TBD achieves over 95% track detection rate and maintains a high MOTA in the medium-to-high SNR range (7 dB to 15 dB). At 3 dB, it still retains a 90% successful tracking rate, whereas DP-TBD, MKCF-TBD, and TRO-TBD yield P_{dtrack} of only 42.50%, 12.50%, and 62.50%, respectively. Moreover, CL-UNet-TBD achieves a MOTA of 0.823 at 3 dB, significantly outperforming the highest MOTA among competing methods (TRO-TBD at 0.714). Overall, CL-UNet-TBD demonstrates superior tracking performance in low-SNR environments which stably extract weak targets while suppressing false alarms.

Practical deployment of the algorithm requires both accuracy and efficiency. We visualize the efficiency-accuracy trade-off at 3 dB RD-map SNR (-38 dB raw echo SNR) by plotting P_{dtrack} versus the average processing time per frame in Fig. 5. Compared with other TBD baselines (DP-TBD,

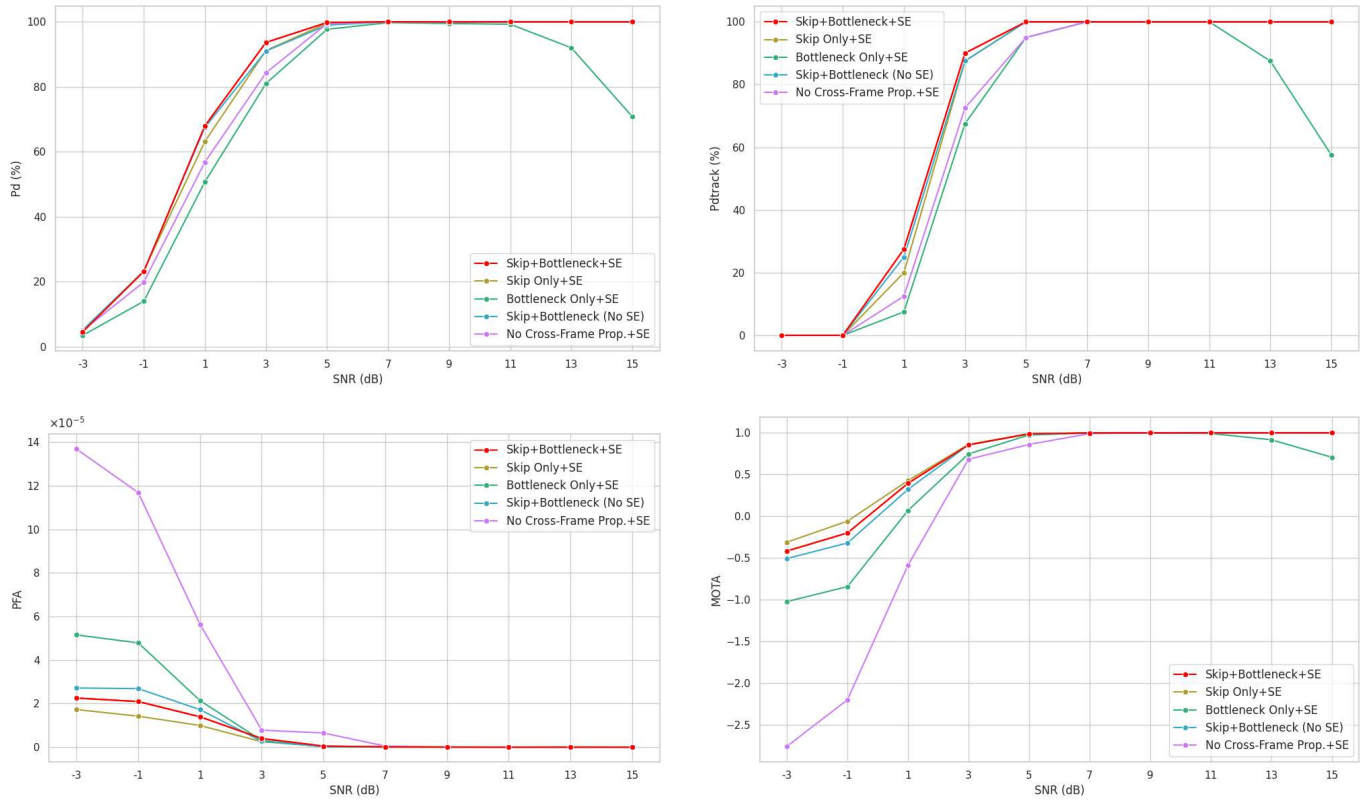


Fig. 3. Ablation study results: P_d , P_{drack} , PFA, and MOTA vs. SNR.

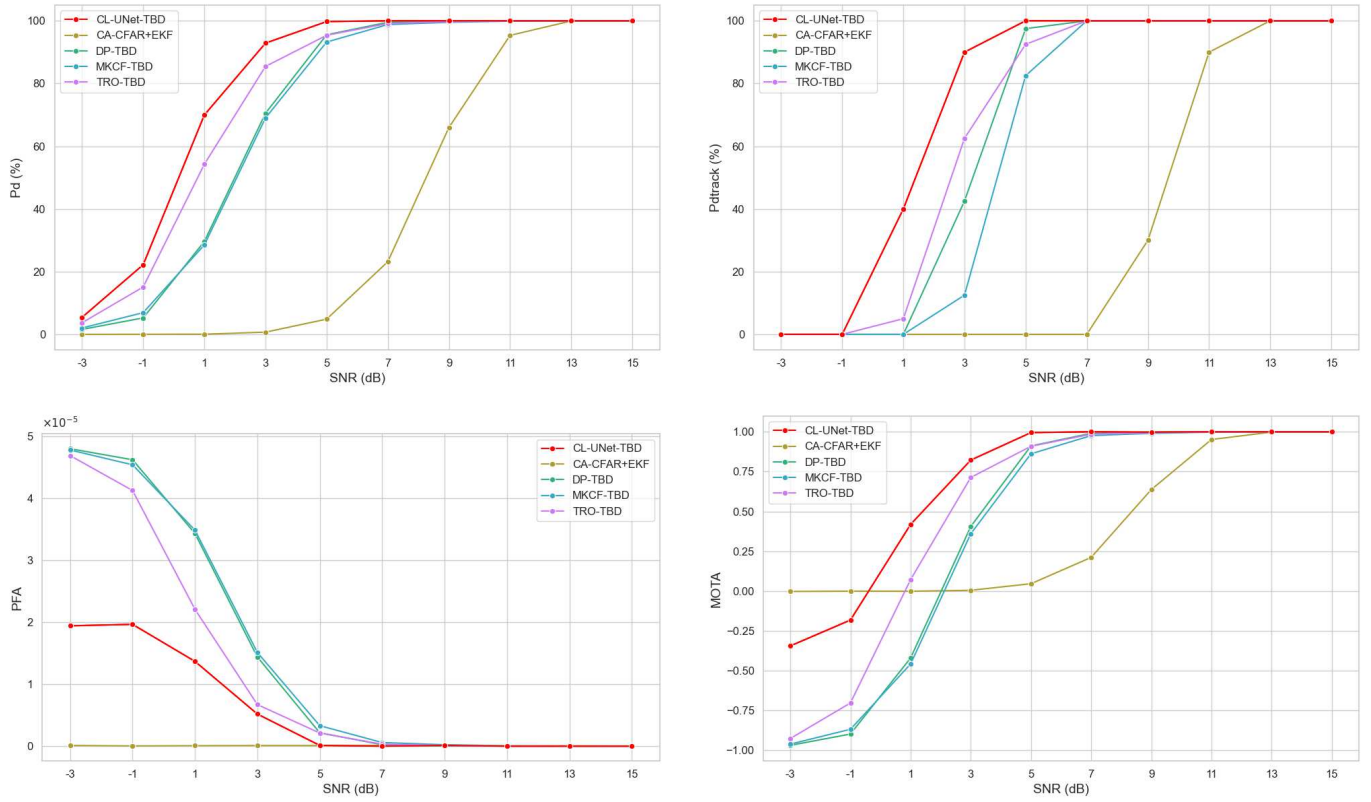


Fig. 4. Comparison study results: P_d , P_{drack} , PFA, and MOTA vs. SNR.

MKCF-TBD, and TRO-TBD), CL-UNet-TBD achieves the highest $P_{d\text{track}}$ while also being the fastest TBD method (37.9 ms/frame). Although CA-CFAR+EKF is faster (1.7 ms/frame), it fails to track at this SNR ($P_{d\text{track}} = 0\%$). Overall, CL-UNet-TBD provides the most favorable accuracy–efficiency trade-off.

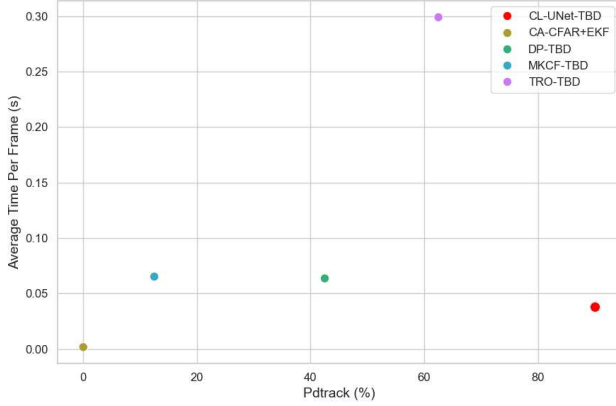


Fig. 5. Efficiency–accuracy trade-off at 3 dB RD-map SNR (-38 dB raw echo SNR): $P_{d\text{track}}$ versus average time per frame.

V. CONCLUSION

This paper proposes CL-UNet-TBD, a novel radar target tracking method for low-SNR environments. By feeding the accumulation maps from the classical DP-TBD algorithm into a deep network equipped with U-Net, ConvLSTM, and SE modules, the benefits of physical models and data-driven learning are expressed effectively. Our model significantly outperforms the existing approaches in detection rate, tracking accuracy, and false alarm rate at very low-SNR levels (e.g., 3 dB RD-map SNR, corresponding to -38 dB raw echo SNR), while maintaining superior computational efficiency. It offers a robust solution for detecting weak targets like stealth aircraft and small drones. Future work will extend the model in more complex scenarios with non-Gaussian clutter and maneuvering dense swarm targets.

REFERENCES

- [1] B. D. Carlson, E. D. Evans, and S. L. Wilson, “Search radar detection and track with the Hough transform. I. System concept,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 30, no. 1, pp. 102–108, 1994.
- [2] S. M. Tonissen and R. J. Evans, “Performance of dynamic programming techniques for Track-before-detect,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 32, no. 4, pp. 1440–1451, 1996.
- [3] X. Lu, “Track-before-detect algorithms for airborne radar in K-distributed clutter,” Master’s thesis, Nanjing University of Science and Technology, 2022.
- [4] E. Fishel, N. Tsarov, T. Tapiro, I. Nuri, and N. Shlezinger, “Learn to track-before-detect via neural dynamic programming,” in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024, pp. 9586–9590.

- [5] X. Liu, Y. Chen, and S. Zhang, “Track-before-detect with transformer-based transition range optimization for maneuvering target detection,” SSRN, 2025, posted: 24 Jun 2025. [Online]. Available: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=5318594
- [6] C. Carr, B. Dang, and J. G. Metcalf, “RADGAN: Applying adversarial machine learning to track-before-detect radar,” in *2021 IEEE Radar Conference (RadarConf21)*. IEEE, 2021, pp. 1–6.
- [7] D. J. Salmond and H. Birch, “A particle filter for Track-before-detect,” in *Proceedings of the 2001 American Control Conference*, 2001, pp. 3755–3760.
- [8] J. Arnold, S. W. Liu, and K. Ward, “Track-before-detect processing for an airborne type radar,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 42, no. 4, pp. 1234–1246, 2006.
- [9] B.-N. Vo, S. S. Singh, and A. Doucet, “Sequential Monte Carlo methods for multitarget filtering with random finite sets,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, no. 4, pp. 1224–1245, 2005.
- [10] M. Dragovic, “Velocity filtering for target detection and track initiation,” DSTO Systems Sciences Laboratory, Defence Science and Technology Organisation, Edinburgh, SA, Australia, Technical Report ADA415922, 2003.
- [11] G. Zhou, L. Wang, and T. Kirubarajan, “A pseudo-spectrum approach for weak target detection and tracking,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 55, no. 6, pp. 3394–3412, dec 2019.
- [12] J. Liu, H. Wu, Z. Yang, X. Hua, and Y. Cheng, “An information geometry-based track-before-detect algorithm for range-azimuth measurements in radar systems,” *Entropy*, vol. 27, no. 6, p. 637, 2025.
- [13] C. Decourt, R. VanRullen, D. Salle, and T. Oberlin, “DAROD: A deep automotive radar object detector on Range-Doppler maps,” in *2022 IEEE Intelligent Vehicles Symposium (IV)*. Aachen, Germany: IEEE, 2022, pp. 112–118.
- [14] V. Nimbaragi, S. Sasi, and V. Jithesh, “LSTM-RNN for radar data processing,” *International Research Journal of Engineering and Technology (IRJET)*, vol. 7, no. 5, 2020.
- [15] J. Liu, Z. Wang, and M. Xu, “DeepMTT: A deep learning maneuvering target-tracking algorithm based on bidirectional LSTM network,” *Information Fusion*, vol. 53, pp. 289–299, 2020.
- [16] S. Jouaber, S. Bonnabel, S. Velasco-Forero, and M. Pilté, “NNAKF: A neural network adapted Kalman filter for target tracking,” in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 4075–4079.
- [17] G. Revach, N. Shlezinger, X. Ni, A. L. Escoriza, R. J. G. van Sloun, and Y. C. Eldar, “KalmanNet: Neural network aided Kalman filtering for partially known dynamics,” *IEEE Transactions on Signal Processing*, vol. 70, pp. 1532–1547, 2022.
- [18] A. Testolin and R. Diamant, “Combining denoising autoencoders and dynamic programming for acoustic detection and tracking of underwater moving targets,” *Sensors*, vol. 20, no. 10, p. 2945, 2020.
- [19] C. Zhu, “Radar multi-frame track-before-detect algorithms based on supervised learning,” Master’s thesis, University of Electronic Science and Technology of China, Sichuan, China, 2024.
- [20] X. Lu, Z. Pan, and H. Zhou, “Cross-attention transformer for coherent detection in radar under low-SNR conditions,” *Sensors*, vol. 25, no. 24, p. 7588, 2025. [Online]. Available: <https://doi.org/10.3390/s25247588>
- [21] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional networks for biomedical image segmentation,” in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, ser. Lecture Notes in Computer Science, vol. 9351. Springer, 2015, pp. 234–241.
- [22] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo, “Convolutional LSTM network: A machine learning approach for precipitation nowcasting,” in *Advances in Neural Information Processing Systems*, vol. 28, 2015, pp. 802–810.
- [23] H. Rohling, “Radar CFAR thresholding in clutter and multiple target situations,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-19, no. 4, pp. 608–621, 1983.
- [24] R. E. Kalman, “A new approach to linear filtering and prediction problems,” *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.
- [25] G. Zhou and L. Wang, “Pseudo-spectrum based speed square filter for Track-before-detect in Range-Doppler domain,” *IEEE Transactions on Signal Processing*, vol. 67, no. 21, pp. 5596–5610, 2019.