

Article

LAKF-Based 3D Multi-Object Tracking: Hybrid KalmanNet for Nonlinear Motion Estimation

Yucheng Long ¹, Jian Song ¹, Yunfeng Xu ^{1,2}, Peng He ¹, Lina Bu ¹ and Qiang Fu ^{1,*}

¹ The Shijiazhuang Campus, Army Engineering University of PLA, Shijiazhuang 050003, China; longyc0010@163.com (Y.L.); songj9507@outlook.com (J.S.); hbkd_xyf@hebust.edu.cn (Y.X.); hep0168@aeu.edu.cn (P.H.); bulina90@aeu.edu.cn (L.B.)

² School of Information Science and Engineering, Hebei University of Science and Technology, Shijiazhuang 050018, China

* Correspondence: Fu_Qiang@aeu.edu.cn

Abstract

Reliable 3D multi-object tracking (MOT) is essential for autonomous driving. The model-based Kalman filter (KF) is the standard motion estimator in 3D MOT, but its linear-dynamics assumption and constant noise covariances limit its adaptability to nonlinear motion and time-varying noise. The learning-aided Kalman filter (LAKF) addresses these issues by learning the Kalman gain (KG) through recurrent neural networks. We adopt CNPE-KalmanNet, a variant that integrates complex-number position encoding to capture semantic correlations among heterogeneous state components. However, the analytical KG is derived under the assumption that the process and measurement noise covariance matrices are positive definite, whereas the learned KG from CNPE-KalmanNet lacks structural constraints. The unconstrained learned KG tends to produce a dense matrix, leading to cross-dimensional coupling, particularly across different scales. This paper proposes a hybrid KG that fuses the model-based analytical KG with the data-driven learned KG. Compared to the AB3DMOT baseline on KITTI and the P3DMOT baseline on nuScenes, our approach improves average recall by 13% and 22%, and IoU by 6% and 7% in state estimation, and AMOTA by 0.7% and 0.5% for overall tracking accuracy. The hybrid design retains the numerical stability of analytical filtering while incorporating the strong nonlinear fitting capacity from deep learning.

Keywords: Kalman filter; recurrent neural networks; tracking-by-detection



Academic Editors: Felipe Jiménez and José María López

Received: 25 June 2026

Revised: 29 July 2026

Accepted: 30 July 2026

Published: 2 August 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

3D multi-object tracking (MOT) [1,2] is critical to autonomous driving, as it provides persistent state estimates of nearby dynamic agents. This allows the control stack to forecast object trajectories more accurately and boost overall vehicle control stability. The major 3D MOT paradigms are tracking-by-detection (TBD) [3], joint detection and tracking (JDT) [4], and tracking-by-attention (TBA) [5]. TBD has become the dominant framework for 3D MOT, owing to its high efficiency and excellent compatibility with 3D object detection models that separate detection, estimation, and association modules. The estimation module employs a Kalman filter (KF) [6] for motion prediction, while the association module performs data association between predictions and detections. This design provides greater interpretability compared to end-to-end methods. In contrast, JDT and TBA adopt end-to-end architectures to simultaneously train detection and association directly from sensor inputs. This generally demands larger training datasets and complicates the joint optimization process.

With the rapid development of 3D sensors and detection technologies, TBD-based 3D MOT methods, such as AB3DMOT [7] and P3DMOT [8], have been widely applied in autonomous driving and intelligent monitoring, achieving reliable tracking performance. Prior research on 3D TBD methods focuses on the data association module, including the design of novel similarity metrics and the optimization of the matching algorithm to reduce association errors. Recent studies on TBD-based 3D multi-object tracking employ multi-feature cascaded association [9], fast hierarchical association [10], and robust two-stage multimodal association [11]. For JDT paradigms, ADA-Track++ [12] designs edge-augmented graph attention learnable association integrated within an alternating detection-association transformer decoder. Under TBA paradigms, the GMM-based LiDAR-camera tracker [13] refines probabilistic data association. In contrast, improvements to the estimation module, which is the core of motion prediction and underpins data association, have been less explored. Most existing TBD methods still rely on the linear KF as the estimator, which cannot effectively handle nonlinear dynamics, leading to prediction and association errors that cause trajectory drift. Furthermore, its process and measurement noise covariances remain static throughout tracking, failing to adapt to dynamic scene changes, such as object occlusion and sensor noise fluctuations.

Driven by deep learning, the learning-aided Kalman filter (LAKF) is proposed to overcome the KF's limitations in nonlinear dynamics and noise adaptation. The LAKF architectures that integrate Long Short-Term Memory (LSTM) [14] or the Gated Recurrent Unit (GRU) [15] networks with KF can be categorized according to their primary learning objectives. Certain approaches [16,17] aim to learn time-varying noise covariances $\mathbf{Q}_t$ and $\mathbf{R}_t$, while others aim to optimize the Kalman gain (KG) [18] to enhance state estimation performance.

As a representative GRU-based LAKF architecture, KalmanNet (KNet) [18,19] directly obtains the KG in a data-driven manner. Owing to its compatibility with traditional KF pipelines, KNet and its variants have been successfully applied to various estimation tasks, including integrated navigation [20] and 2D MOT [21,22]. Other variants are designed to process high-dimensional inputs such as image and graph data [23,24], and to improve robustness to model mismatch [25]. Recent advances further propose CNPE-KNet [26], a variant that integrates complex-number position encoding (CNPE) to characterize semantic correlations among heterogeneous components.

Despite its promising results in various estimation tasks, existing KNet-based methods still suffer from limitations when applied to 3D MOT. In 3D MOT, the height h and yaw angle ψ are weakly coupled, so measurement noise in one should not drive large corrections in the other. The recurrent neural network (RNN) imposes no structural constraints on the KG, leading to dense connections and cross-dimensional noise crosstalk. The analytical KG computed by the standard KF naturally maintains a block structure that respects the weak-coupling prior. Motivated by this, we propose a hybrid KG that fuses the model-based analytical KG with the data-driven learned KG, using the former as a structural prior to suppress cross-dimensional noise crosstalk.

To clarify the advantages of the hybrid KG, we contrast it with two alternative strategies adopted in LAKF architectures. Methods that learn $\mathbf{Q}_t$ and $\mathbf{R}_t$ covariances only adapt to time-varying noise terms and cannot resolve persistent trajectory errors resulting from the fixed linear motion model. In contrast, learning KG implicitly learns nonlinear dynamics and noise covariances, mitigating trajectory drift caused by fixed linear assumptions. Hard-mask methods require a criterion to determine which entries to suppress at each time step, and no such criterion exists under nonlinear dynamics. A hard mask derived from the analytical KG inherits the limitations of the standard KF, which is optimal only under linear-Gaussian assumptions. Its block structure neglects weak but meaningful cross-state

correlations, and the fixed mask based on this structure eliminates valid coupling information under nonlinear dynamics. In contrast, the proposed hybrid KG approach applies soft constraints derived from the analytical KF to suppress noise crosstalk while preserving the model's capacity to learn nonlinear relationships.

The main contributions of this paper are summarized as follows:

1. We propose a time-varying hybrid KG that fuses analytical model KG and data-driven CNPE-KNet KG. It adopts the stable block structural prior of the standard KF while retaining the nonlinear learning capacity of CNPE-KNet, thereby suppressing undesired cross-dimensional noise crosstalk.
2. We integrate the hybrid estimator into the 3D TBD framework as a plug-and-play module for the traditional KF. Unlike adaptive $\mathbf{Q}_t/\mathbf{R}_t$ filters that require retraining when switching detectors, our model undergoes one-time training on synthetic data and can be deployed directly on any KF-based tracker without extra fine-tuning.

The remainder of this paper is arranged as follows. In Section 2, we introduce preliminaries on commonly used filtering algorithms and the TBD method. Section 3 details the proposed LAKF-based 3D MOT framework, which adopts a hybrid KG to suppress undesired cross-dimensional noise crosstalk. Section 4 evaluates performance on the KITTI [27] and nuScenes [28] datasets.

2. Materials and Methods

2.1. Model-Based Kalman Filter

The model-based KF propagates the posterior state mean and covariance across time steps through two sequential stages: prediction and update.

In the prediction stage of the proposed filter, the prior state estimate and prior error covariance matrix are propagated forward through the state transition matrix $\mathbf{F}$.

2.1.1. Prediction Stage

$$\begin{aligned}\hat{\mathbf{x}}_{t|t-1} &= \mathbf{F}\hat{\mathbf{x}}_{t-1|t-1}, \\ \mathbf{P}_{t|t-1} &= \mathbf{F}\mathbf{P}_{t-1|t-1}\mathbf{F}^\top + \mathbf{Q}_t.\end{aligned}\quad (1)$$

In the update stage, the prior state estimate is corrected by fusing the current measurement through the KG, which optimally balances the predicted and measured information.

2.1.2. Update Stage

$$\begin{aligned}\hat{\mathbf{y}}_{t|t-1} &= \mathbf{H}\hat{\mathbf{x}}_{t|t-1}, \\ \mathbf{S}_t &= \mathbf{H}\mathbf{P}_{t|t-1}\mathbf{H}^\top + \mathbf{R}_t, \\ \mathbf{K}_t &= \mathbf{P}_{t|t-1}\mathbf{H}^\top\mathbf{S}_t^{-1}, \\ \hat{\mathbf{x}}_{t|t} &= \hat{\mathbf{x}}_{t|t-1} + \mathbf{K}_t(\mathbf{y}_t - \hat{\mathbf{y}}_{t|t-1}), \\ \mathbf{P}_{t|t} &= (\mathbf{I} - \mathbf{K}_t\mathbf{H})\mathbf{P}_{t|t-1}.\end{aligned}\quad (2)$$

$\hat{\mathbf{x}}_{t-1|t-1} \in \mathbb{R}^m$ and $\mathbf{P}_{t-1|t-1} \in \mathbb{R}^{m \times m}$ are the posterior state mean and covariance at time $t-1$. $\hat{\mathbf{x}}_{t|t-1}$ and $\mathbf{P}_{t|t-1}$ are the predicted state mean and covariance at time t . $\hat{\mathbf{y}}_{t|t-1} \in \mathbb{R}^n$ and $\mathbf{S}_t \in \mathbb{R}^{n \times n}$ are the predicted measurement mean and covariance at time t . The state transition matrix $\mathbf{F}$ and the measurement matrix $\mathbf{H}$ respectively characterize the system dynamics and measurement model, while $\mathbf{Q}_t$ and $\mathbf{R}_t$ are the process and measurement noise covariance matrices.

2.2. Learning-Aided Kalman Filter

KNet is a representative LAKF that combines RNNs with the recursive framework of the traditional KF. It implicitly captures complex nonlinear dynamics and time-varying

uncertainties without requiring explicit knowledge of the noise statistics. At each time step t , the input feature vector is constructed as $\mathbf{Z}_t = [F1, F2, F3, F4]^\top$:

$$\begin{aligned} F1 : \Delta \tilde{\mathbf{y}}_t &= \mathbf{y}_t - \mathbf{y}_{t-1}, & F2 : \Delta \mathbf{y}_t &= \mathbf{y}_t - \hat{\mathbf{y}}_{t|t-1}, \\ F3 : \Delta \tilde{\mathbf{x}}_t &= \hat{\mathbf{x}}_{t|t} - \hat{\mathbf{x}}_{t-1|t-1}, & F4 : \Delta \hat{\mathbf{x}}_t &= \hat{\mathbf{x}}_{t|t} - \hat{\mathbf{x}}_{t|t-1}, \end{aligned} \quad (3)$$

where $\Delta \tilde{\mathbf{y}}_t$, $\Delta \mathbf{y}_t$, $\Delta \tilde{\mathbf{x}}_t$, and $\Delta \hat{\mathbf{x}}_t$ are the measurement difference, the innovation, the state difference, and the state evolution difference, respectively.

The concatenated $\mathbf{Z}_t$ is fed into the RNNs to learn a KG $\mathbf{K}_t(\Theta)$, where Θ denotes the trainable parameters of the RNNs:

$$\hat{\mathbf{x}}_{t|t} = \hat{\mathbf{x}}_{t|t-1} + \mathbf{K}_t(\Theta)(\mathbf{y}_t - \hat{\mathbf{y}}_{t|t-1}). \quad (4)$$

To improve KNet's robustness against model mismatch, existing variants are primarily developed from three perspectives: network structure modifications, optimized training algorithms, and the characterization of correlations across heterogeneous state components. Split-KNet [25] employs two separate recurrent networks to learn the prior state and innovation covariances, improving robustness to model mismatch. For integrated navigation sensor fusion, Song [20] developed a modified sliding-window TBPTT training algorithm that leverages the first-order Markov property of system states to mitigate gradient explosion. Several variants [21,22] employ different feature modeling to decouple state semantics, overcoming the limitations of the standard KNet, which directly concatenates heterogeneous state components. CNPE-KNet [26], which serves as the backbone of our method, integrates CNPE to characterize semantic correlations among these components.

2.3. Tracking-by-Detection

TBD is the predominant paradigm in 3D MOT and has been widely deployed in industrial applications. As noted in Section 1, TBD decomposes tracking into detection, estimation, and association modules. The complete implementation of TBD is summarized in Algorithm 1.

In the detection module, the detector generates 3D bounding boxes for all potential objects in each frame. The observation vector is defined as

$$\mathbf{y} = [x, y, z, l, w, h, \psi], \quad (5)$$

where (x, y, z) denote the 3D center coordinates, (l, w, h) denote the object dimensions, and ψ is the heading angle.

The estimation module employs a constant velocity (CV) model-based KF for each trajectory, with the object state represented as

$$\mathbf{x} = [x, vx, y, vy, z, vz, l, w, h, \psi]^\top. \quad (6)$$

The filter predicts the state for the subsequent frame, where (vx, vy, vz) denote the velocities along the three axes. The predicted observation $\mathbf{y}_{\text{pred}}^{(i)}$ is computed through the observation model. The state transition matrix $\mathbf{F}$ and the observation matrix $\mathbf{H}$ are defined as

$$\begin{aligned} \mathbf{F} &= \text{diag}[\mathbf{F}_{cv}, \mathbf{F}_{cv}, \mathbf{F}_{cv}, 1, 1, 1], \mathbf{F}_{cv} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}, \\ \mathbf{H} &= \text{diag}[\mathbf{H}_1, \mathbf{H}_1, \mathbf{H}_1, 1, 1, 1], \mathbf{H}_1 = \begin{bmatrix} 1 & 0 \end{bmatrix}, \end{aligned} \quad (7)$$

where Δt is the time interval. The states x, y, z follow the CV model, which is characterized by the $\mathbf{F}_{cv}$ matrix. The states l, w, h, ψ are modeled as constant. $\mathbf{H}$ is the measurement

matrix that extracts the observable components while ignoring velocities. In most 3D MOT systems, the process noise matrix $\mathbf{Q}_t$ and the measurement noise matrix $\mathbf{R}_t$ are predefined, constant matrices that reflect statistical sensor and motion errors.

Algorithm 1 Tracking-by-detection with Kalman filter

Input: Video sequence V , total frame count T , detection threshold τ_{score} , initial covariance $\mathbf{P}_{\text{init}}$
Output: Tracks $\mathcal{T}$;
Initialization: $\mathcal{T} \leftarrow \emptyset$ $\triangleright t = 1$, all detections will directly initialize new tracks via the unmatched detections branch.
for each frame _{t} from 1 to T **do**
 for each track $i \in \mathcal{T}_{t-1}^i$ **do**
 $\mathbf{y}_{\text{pred}}^{(i)} \leftarrow \text{KF.Predict}()$
 end for
 $\mathcal{D}_t \leftarrow \text{Detector}(\text{frame}_t)$
 $\mathcal{D}_t \leftarrow \{\text{det} \in \mathcal{D}_t \mid \text{det.score} \geq \text{score}_{\text{thre}}\}$
 Associate $\{\mathbf{y}_{\text{pred}}^{(i)}\}$ with $\mathcal{D}_t$ using similarity matrix
 for each matched pair $(\mathbf{y}_{\text{pred}}^{(i)}, \text{det})$ **do**
 $\mathbf{x}_{t|t}^{(i)}, \mathbf{P}_{t|t}^{(i)} \leftarrow \text{KF.Update}()$
 Add $\mathbf{x}_{t|t}^{(i)}$ to $\mathcal{T}^i$
 end for
 for unmatched detections $\mathbf{y}_t \in \mathcal{D}_t$ **do**
 Initialize a new track: $\mathbf{x}_{t|t}^{(n)} = \mathbf{y}_t, \mathbf{P}_{t|t}^{(n)} = \mathbf{P}_{\text{init}}$
 Add $\mathbf{x}_{t|t}^{(n)}$ to $\mathcal{T}^n$
 end for
 $\mathcal{T}_{t-1} \leftarrow \mathcal{T}_t$
end for
Return: $\mathcal{T}$

The association module constructs a cost matrix by computing the 3D Intersection over Union (IoU) between each predicted observation $\mathbf{y}_{\text{pred}}^{(i)}$ and all detections in the current frame. Bipartite matching at each frame is performed using the Hungarian algorithm [29]. For each matched pair, the KF updates the state estimate using the associated detection. If a trajectory's predicted observation does not match any detection, the filter skips the update step and retains the predicted state as the posterior estimate. After several consecutive frames without a matching detection, the trajectory is terminated. Unmatched detections initialize new KF instances.

2.4. Limitations

While KNet avoids manual tuning of the noise covariance matrices and learns unknown system dynamics end-to-end, it concatenates physically disparate components into a single feature vector $\mathbf{Z}_t$. Aggregating these heterogeneous components mixes information with distinct physical meanings, breaking their semantic independence and obscuring the correlations among them. The recently proposed CNPE-KNet mitigates this issue through CNPE, which maps real-valued components into the complex domain and applies component-wise rotary transformations. Under this strategy, homogeneous components share the same rotation priors, whereas heterogeneous components have distinct rotation priors. This design preserves the original norm of each component, thereby maintaining semantic independence, while complex-valued Multi-Layer Perceptrons (MLPs) model the semantic correlations among components.

However, CNPE-KNet optimizes component encoding and learns the KG in a data-driven manner without physical structural constraints. This undesired cross-dimensional noise crosstalk primarily arises from the unconstrained, dense structure of the learned $\mathbf{K}_t(\Theta)$. To compensate for time-varying dynamics and model mismatch, the network captures correlations via the off-diagonal elements of the dense $\mathbf{K}_t(\Theta)$. Consequently, the off-diagonal entries couple dimensions with substantially different noise magnitudes. For instance, the noise magnitude of h is on the order of 10^{-1} , while the noise magnitude of the ψ is on the order of 10^{-2} . As a result, the larger noise magnitude of h contaminates the correction of the ψ angle. Without explicit physics-driven structural constraints, these redundant off-diagonal connections cannot be suppressed. To suppress undesired cross-dimensional noise crosstalk, we design a hybrid KG module equipped with explicit physics-driven constraints.

2.5. Hybrid Kalman Gain for State Estimation

To combine the advantages of the model-based KF and the data-driven CNPE-KNet, we propose a time-varying hybrid KG:

$$\mathbf{K}_t^{\text{Hybrid}} = \lambda_t \mathbf{K}_t + (1 - \lambda_t) \tilde{\mathbf{K}}_t(\Theta), \quad (8)$$

where $\mathbf{K}_t$ denotes the model-based KG computed at each time step t using predefined noise covariances $\mathbf{Q}_t$ and $\mathbf{R}_t$, and $\tilde{\mathbf{K}}_t(\Theta)$ represents the data-driven KG learned by CNPE-KNet from the input features $\mathbf{Z}_t$. λ_t is a learned time-varying parameter that balances the contributions of the model-based $\mathbf{K}_t$ and data-driven $\tilde{\mathbf{K}}_t(\Theta)$.

Following the standard KF prediction and KG computation detailed in Equations (1) and (2), our filtering pipeline computes the prior state covariance $\mathbf{P}_{t|t-1}$ and model-based KG $\mathbf{K}_t$ for each time step t . Given the hybrid KG derived in Equation (8), the KG is embedded within the update recursion to obtain the posterior state estimate:

$$\hat{\mathbf{x}}_{t|t} = \hat{\mathbf{x}}_{t|t-1} + \mathbf{K}_t^{\text{Hybrid}} (\mathbf{y}_t - \hat{\mathbf{y}}_{t|t-1}). \quad (9)$$

To avoid indefinite covariance matrices induced by unconstrained $\mathbf{K}_t(\Theta)$, the Joseph stabilized covariance update is adopted:

$$\mathbf{P}_{t|t} = (\mathbf{I} - \mathbf{K}_t^{\text{Hybrid}} \mathbf{H}) \mathbf{P}_{t|t-1} (\mathbf{I} - \mathbf{K}_t^{\text{Hybrid}} \mathbf{H})^\top + \mathbf{K}_t^{\text{Hybrid}} \mathbf{R}_t (\mathbf{K}_t^{\text{Hybrid}})^\top. \quad (10)$$

The Joseph covariance update provides a strict theoretical guarantee for the positive semi-definiteness (PSD) of posterior covariance. Since the initial state covariance $\mathbf{P}_{\text{init}}$, noise covariance matrices $\mathbf{Q}_t$, and $\mathbf{R}_t$ are predefined as PSD diagonal matrices, the prior covariance $\mathbf{P}_{t|t-1}$ propagated through the prediction stage satisfies $\mathbf{P}_{t|t-1} \geq 0$. Combined with the Joseph recursion, the posterior covariance $\mathbf{P}_{t|t}$ retains the PSD property for any hybrid KG $\mathbf{K}_t^{\text{Hybrid}}$ in mathematical theory.

The fusion weight $\lambda_t \in [0, 1]$ makes $\mathbf{K}_t^{\text{Hybrid}}$ a convex combination of analytical KG $\mathbf{K}_t$ and learned KG $\tilde{\mathbf{K}}_t(\Theta)$. From the triangle inequality of the spectral norm,

$$\|\mathbf{K}_t^{\text{Hybrid}}\| \leq \lambda_t \|\mathbf{K}_t\| + (1 - \lambda_t) \|\tilde{\mathbf{K}}_t(\Theta)\|. \quad (11)$$

For the controllable and observable CV model, the model-based KG $\mathbf{K}_t$ is uniformly bounded. The data-driven KG $\tilde{\mathbf{K}}_t(\Theta)$ is strictly bounded via tanh activation:

$$\|\tilde{\mathbf{K}}_t(\Theta)\| = \|\tanh(\mathbf{K}_t(\Theta))\| \leq 1. \quad (12)$$

Combining the two bounds, a global constant $K_{\max}$ can be established:

$$K_{\max} = \max\{\sup_t \|\mathbf{K}_t\|, \sup_t \|\tilde{\mathbf{K}}_t(\Theta)\|\}. \quad (13)$$

With bounded $\mathbf{K}_t^{\text{Hybrid}}$, $\mathbf{F}$, $\mathbf{H}$, $\mathbf{Q}_t$ and $\mathbf{R}_t$, the Joseph stabilized covariance update guarantees that the posterior error covariance $\mathbf{P}_{t|t}$ remains bounded, which effectively eliminates long-term filtering divergence.

Unlike the data-driven architecture that trains unconstrained $\mathbf{K}_t(\Theta)$, our design employs a shared RNN backbone equipped with two output branches. At each time step t , the RNN retains a shared hidden state $\mathbf{h}_{t-1}$. Leveraging this shared $\mathbf{h}_{t-1}$, one branch applies a linear projection followed by a sigmoid activation to produce the scalar weight λ_t , while the other produces the data-driven $\tilde{\mathbf{K}}_t(\Theta)$. λ_t is optimized end-to-end with the network parameters, achieving temporal adaptation to match system dynamics.

Given the block structure of $\mathbf{F}$ and $\mathbf{H}$ in Equation (7) and diagonal $\mathbf{Q}_t$ and $\mathbf{R}_t$, the analytical KG $\mathbf{K}_t$ inherits a block structure. The sub-blocks (x, vx) , (y, vy) , (z, vz) , l , w , h , and ψ are mutually decoupled in $\mathbf{K}_t$. This block structure serves as a structural prior in the hybrid KG. Since $\mathbf{K}_t$ has zero entries between weakly coupled dimensions, the hybrid KG scales those entries of $\tilde{\mathbf{K}}_t(\Theta)$ by $1 - \lambda_t$, directly suppressing cross-dimensional noise crosstalk. The state estimate follows the same recursive form as Equation (4).

End-to-end training of the estimator is implemented with backpropagation through time (BPTT). The training objective minimizes the deviation between the state estimate $\hat{\mathbf{x}}_{t|t}$ and the ground-truth state $\mathbf{x}_t^{\text{gt}}$. Rather than relying on standard mean squared error (MSE), we adopt the SmoothL1 loss for more robust training. The SmoothL1 loss has been shown to improve training stability in the Split-KNet framework [25]:

$$\mathcal{L} = \frac{1}{T} \sum_{t=1}^T \text{SmoothL1}(\hat{\mathbf{x}}_{t|t} - \mathbf{x}_t^{\text{gt}}), \quad (14)$$

where T denotes the sequence length.

$$\text{SmoothL1}(d) = \begin{cases} 0.5d^2, & \text{if } |d| < 1, \\ |d| - 0.5, & \text{otherwise,} \end{cases} \quad (15)$$

where d is $\hat{\mathbf{x}}_{t|t} - \mathbf{x}_t^{\text{gt}}$. Combining the advantages of L2 and L1 losses, the SmoothL1 loss behaves like MSE for $|d| < 1$ and produces smooth gradient updates. For large estimation errors, it switches to L1 loss to constrain gradient magnitudes and mitigate the risk of gradient explosion.

2.6. Data Preparation for Estimator Training

Existing 3D MOT datasets contain only frame-wise annotations and noisy detection measurements, lacking explicitly paired ground-truth states. As an LAKF trained under supervised learning, KNet relies on matched pairs of ground-truth states and noisy measurements to model system dynamics. To obtain a paired trajectory of ground-truth states and noisy measurements, we generate synthetic detection boxes by adding zero-mean Gaussian noise to each ground-truth bounding box $[x, y, z, l, w, h, \psi]$. The diagonal entries of the measurement covariance matrix are adopted from offline statistics. For KITTI, the diagonal elements are

$$[0.01970, 0.01179, 0.04190, 0.52534, 0.11816, 0.00983, 0.01602],$$

and nuScenes uses

$$[0.08900, 0.09412, 0.03265, 1.00536, 0.10913, 0.02359, 0.02455].$$

In the TBD framework, new detections are associated with existing trajectories via IoU computation between the detection box and the predicted bounding box. We adopt an IoU threshold of 0.7 for the KITTI dataset and 0.5 for the nuScenes dataset. Accordingly, only measurements exceeding this IoU threshold are used for track matching. To ensure the synthetic detections reflect real association conditions, we enforce that each synthetic detection box attains an IoU above the matching threshold with its corresponding ground-truth box. We permit up to 100 resampling attempts to generate a synthetic detection box that satisfies the IoU threshold. If the IoU condition remains unsatisfied after all attempts, the sample with the highest IoU among all attempts is retained. This strategy ensures that the generated detections match the IoU threshold required for association. For ground-truth labels that indicate occlusion, the corresponding synthetic detection is set to NaN. This simulates partial-observation scenarios and drives the estimator to learn robust inference under incomplete data.

2.7. Overall Framework and System Integration

The complete pipeline of the proposed method consists of two stages, as illustrated in Figure 1:

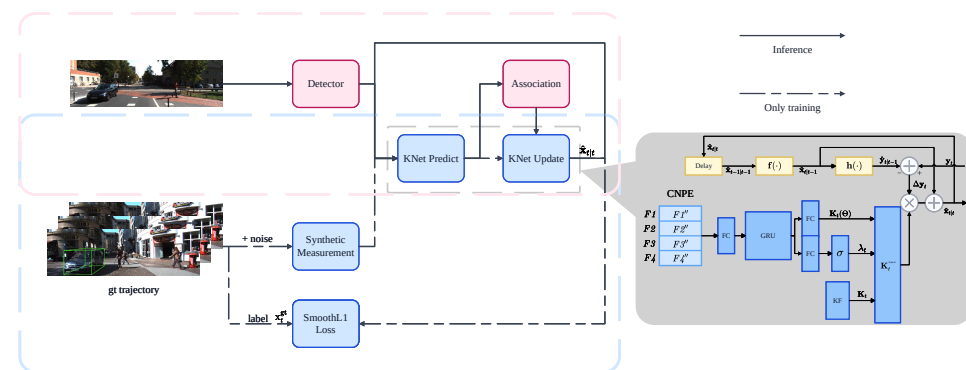


Figure 1. Overall framework of the proposed hybrid KalmanNet for 3D MOT, including offline training and online inference pipelines, where the blue dashed box denotes the offline estimator training stage and the red dashed box denotes the online tracking stage.

In the offline training stage, the GRUs within KNet leverage their temporal memory to implicitly compensate for the unknown noise covariance matrices and adapt to nonlinear dynamics and time-varying uncertainties. Under ground-truth supervision, we optimize the trainable parameters Θ and hidden states $\mathbf{h}_t$ to yield the trained hybrid KNet for online 3D tracking.

In the online tracking stage, the trained model directly replaces the linear KF in the 3D TBD framework as a plug-and-play module, requiring no modifications to the detection and association modules. A separate KNet instance is maintained for each trajectory $\mathcal{T}$, and the tracking pipeline follows the standard TBD paradigm in Algorithm 1.

3. Results

3.1. Dataset

The KITTI dataset contains 21 training sequences and 29 test sequences. The training sequences include 8008 frames, with an average of 3.8 detections per frame. Point cloud data is collected at 10 Hz using a Velodyne HDL-64E LiDAR sensor (Velodyne Lidar, Inc., San

Jose, CA, USA). The same training–validation split and evaluation metrics as AB3DMOT are adopted, with 10 sequences used for training and the remaining 11 for validation.

The nuScenes dataset consists of 1000 annotated driving scenes collected in Boston and Singapore. Each scene lasts for 20 s, with 3D bounding boxes annotated at 2 Hz, yielding 40,000 keyframes in total. Point cloud data is captured using a Velodyne HDL-32E LiDAR sensor (Velodyne Lidar, Inc., San Jose, CA, USA) operating at 20 Hz. The dataset is officially split into a train/val set of 850 scenes, with 700 for training and 150 for validation.

To guarantee fair evaluations of the estimation module, we utilize the PointRCNN [30] detector on KITTI and the Megvii [31] detector on nuScenes.

3.2. Evaluation Metrics

We evaluate the proposed hybrid KG strategy from two critical perspectives: state accuracy and MOT capability. The standard metrics are adopted for 3D MOT evaluation:

Recall@50: Recall@50 [27] denotes the recall calculated with a 3D IoU matching threshold of 0.5:

$$\text{Recall@50} = \frac{\text{TP}_{\text{IoU} \geq 0.5}}{\text{TP}_{\text{IoU} \geq 0.5} + \text{FN}}. \quad (16)$$

AR: The average recall metric [32] computes the mean recall across multiple 3D IoU thresholds from 0.5 to 0.95 in increments of 0.05.

$$\text{AR} = \frac{1}{10} \sum_{\tau=0.5}^{0.95} \text{Recall}(\tau), \quad (17)$$

where $\text{Recall}(\tau)$ denotes the recall at 3D IoU threshold τ .

3D IoU: 3D IoU [27] measures the degree of overlap between a predicted 3D bounding box and a ground-truth 3D bounding box.

$$3\text{D IoU} = \frac{|\mathcal{B}_{\text{pred}} \cap \mathcal{B}_{\text{gt}}|}{|\mathcal{B}_{\text{pred}} \cup \mathcal{B}_{\text{gt}}|}, \quad (18)$$

where $\mathcal{B}_{\text{pred}}$ and $\mathcal{B}_{\text{gt}}$ are the predicted and ground-truth 3D boxes.

AMOTA: Average Multi-Object Tracking Accuracy [7] averages MOTA scores across all detection confidence thresholds by integrating over the MOTA-Recall curve, serving as the primary overall metric for 3D MOT.

TP, FP, FN: True Positive denotes predicted boxes successfully matched with ground truths, False Positive denotes unmatched predictions, and False Negative denotes ground-truth objects without any matched predicted boxes [33].

For the state estimation task, we use 3D IoU and AR rather than root-mean-square error (RMSE) as the primary metric. Directly computing RMSE is inappropriate because position, dimension, and yaw orientation are measured in different units. A common approach is to normalize each component to a common range, such as $[0, 1]$. However, normalizing each component would cause the LAKF to assign equal importance to all components. Given that the heterogeneous components have different semantics and noise attributes, treating them as equivalent would require introducing additional hyperparameters to weight each component. Computing RMSE for a single component cannot reflect overall estimation performance among heterogeneous components. In the TBD framework, the quality of state estimation directly affects the IoU-based data association. IoU serves as a more relevant metric, capturing geometric overlap as a scalar, reflecting the estimator's impact on tracking, and avoiding the unit-mixing problem of RMSE.

3.3. Compared Models

To evaluate the effectiveness of the proposed estimation module, we integrate it into two representative TBD-based 3D MOT frameworks. AB3DMOT [7] is a real-time 3D MOT framework that combines a linear KF with the Hungarian algorithm for data association and serves as the baseline tracker on the KITTI dataset. P3DMOT [8] combines the Mahalanobis distance with multimodal appearance features to form a joint matching cost for the Hungarian algorithm, improving robustness to occlusion and sensor noise, and is used as the baseline tracker on the nuScenes dataset. We compare the following five estimation methods: (1) the standard linear KF, (2) standard KNet, (3) the LSTM-based estimator, (4) LSTM-KF, (5) CNPE-KNet, and (6) the proposed hybrid KG method. We adopt the 3D IoU association module with the Hungarian algorithm. All tracking evaluations for the car category are conducted on the validation set. All proposed methods are implemented using PyTorch 2.3.1. Experiments are conducted on a workstation equipped with an Intel Xeon Platinum 8352V CPU, an NVIDIA GeForce RTX 3080 GPU, and 64 GB RAM.

3.4. Implementation Details

The GRU backbone follows the architecture established in KNet [18]. This architecture employs three single-layer GRUs with hidden dimensions of 50, 50, and 24, arranged in a sequentially dependent cascade. The cascade models the $Q \rightarrow P \rightarrow S$ dependency chain inherent to the analytical KF.

Two output heads are attached to this shared backbone. The first produces the learned KG $\hat{\mathbf{K}}_t(\Theta) \in \mathbb{R}^{10 \times 7}$ using a linear projection with tanh activation. The second generates the fusion weight $\lambda_t \in [0, 1]$ through a linear layer followed by a sigmoid. The CNPE module, input feature construction $\mathbf{Z}_t$ (dimension 34), and KG output structure (dimension 70) remain identical to those in CNPE-KNet [26].

All models are trained using the Adam optimizer with a learning rate of 1×10^{-4} and weight decay of 1×10^{-4} for 30 epochs and a batch size of 256. The implementation is based on PyTorch 2.3.1 with CUDA 11.8.

3.5. Quantitative and Qualitative Results

We compare the proposed method with the standard KF, standard KNet, CNPE-KNet, an LSTM-based estimator, and an LSTM-KF that learns adaptive process and observation noise covariances. Owing to space constraints, Figure 2 presents only representative methods in the evolutionary sequence from KF to KNet and to the proposed method. Comprehensive quantitative results for all baseline algorithms are provided in Tables 1 and 2. Figure 2 shows that when an object vehicle is close to the ego vehicle or is making a turn, the estimation box generated by the traditional KF exhibits rotational errors and positional deviations, limiting accurate tracking of the motion trajectory. This limitation originates from the KF's assumptions of linear motion and constant noise covariances, which make it inadequate for representing nonlinear movement trajectories. In contrast, KNet implicitly learns nonlinear motion dynamics and the covariances of process and measurement noise from data. The estimated bounding box aligns more closely with the ground truth, reflecting better adaptability. Furthermore, the proposed hybrid KG incorporates physical constraints and mitigates cross-dimensional noise crosstalk. Using this design, our approach yields more accurate estimates than other baselines.



Figure 2. Qualitative comparison of different state estimators. Green, blue, red, and yellow bounding boxes represent ground truth, standard KF, KNet, and our method, respectively. As the estimator evolves from KF to our method, the estimated box aligns progressively closer to the ground truth.

Table 1. State estimation results on the KITTI and nuScenes datasets. The proposed hybrid KG demonstrates superior performance compared to all baseline methods across all three evaluation metrics on both datasets. The arrow $\uparrow$ denotes that higher values indicate better performance. Bold values indicate the best performance in each column.

Dataset (Tracker)	Method	Recall@50% $\uparrow$	AR% $\uparrow$	3D IoU% $\uparrow$
KITTI (AB3DMOT)	KF [6]	87.32	40.73	64.07
	KNet [18]	87.90	44.84	65.89
	LSTM [14]	87.43	42.45	64.96
	LSTM-KF [16]	87.58	43.71	65.44
	CNPEKNet [26]	88.62	45.22	66.47
	Ours	89.41	46.27	68.20
nuScenes (P3DMOT)	KF [6]	96.57	43.22	68.86
	KNet [18]	96.87	48.32	71.39
	LSTM [14]	96.62	44.48	69.34
	LSTM-KF [16]	96.69	46.71	70.55
	CNPEKNet [26]	97.84	50.74	72.67
	Ours	98.41	52.95	73.90

Table 2. Summary of 3D MOT results on the KITTI and nuScenes datasets. The proposed hybrid KG achieves the highest AMOTA and TP, and the lowest FP on both datasets, with an increase in inference latency compared to the standard KF. The arrows $\uparrow$ and $\downarrow$ denote that higher and lower values, respectively, indicate better performance. Bold values indicate the best performance in each column.

Dataset (Tracker)	Method	AMOTA% $\uparrow$	TP $\uparrow$	FP $\downarrow$	FN $\downarrow$	ms/track/frame $\downarrow$
KITTI (AB3DMOT)	KF [6]	30.6	7769	1065	2079	2.64
	KNet [18]	30.9	7674	965	2148	4.49
	LSTM [14]	30.7	7653	969	2182	3.57
	LSTM-KF [16]	30.8	7665	988	2160	6.88
	CNPEKNet [26]	31.2	7700	922	2116	5.38
	Ours	31.3	7783	916	2133	6.78
nuScenes (P3DMOT)	KF [6]	69.4	41,129	5271	16,701	4.35
	KNet [18]	69.6	41,121	5246	16,742	6.83
	LSTM [14]	69.4	41,098	5331	16,771	5.46
	LSTM-KF [16]	69.5	41,126	5250	16,758	10.03
	CNPEKNet [26]	69.7	41,114	5184	16,733	7.99
	Ours	69.9	41,165	5067	16,655	9.97

4. Discussion

Table 1 presents the state estimation results for various state estimators on the KITTI dataset. As shown in Figure 3a, the IoU values of the KF and its variants are higher than those of the raw detection results, validating the effectiveness of filtering algorithms in improving state estimation. The standard KF achieves an AR of 40.73% and an IoU of 64.07%, serving as the baseline. The LSTM estimator discards the KF structure and learns the state update purely from data, achieving 42.45% AR and 64.96% IoU. The LSTM-KF, which retains the KF structure and learns adaptive noise covariances $\mathbf{Q}_t$ and $\mathbf{R}_t$, improves the AR to 43.71% and IoU to 65.44%, surpassing the KF baseline. However, adapting noise covariances cannot compensate for errors introduced by the fixed linear motion model, limiting its gains relative to KG learning approaches. KNet raises the AR to 44.84% and IoU to 65.89%, demonstrating that RNNs can learn nonlinear dynamics and time-varying noise from data. CNPE-KNet further improves these metrics to 45.22% AR and 66.47% IoU by capturing semantic correlations among heterogeneous components. Building on CNPE-KNet, our hybrid KG incorporates a block structural prior and reaches 46.27% AR and 68.20% IoU. After integrating the estimator into the complete 3D MOT framework, the overall tracking results are summarized in Table 2. The baseline KF achieves an AMOTA of 30.6%. This value rises to 31.2% with CNPE-KNet, and our proposed method further improves it to 31.3%. Regarding computational efficiency, the average latency in milliseconds (ms) per track per frame is reported. The KF achieves the lowest latency of 2.64 ms per track per frame due to its simple linear structure. KNet, which learns the KG, and LSTM-KF, which learns process and observation noise parameters, achieve latencies of 4.49 and 6.88 ms per track per frame, respectively. The proposed method, which incorporates a structural prior and feature-correlation modeling, operates at 6.78 ms per track per frame.

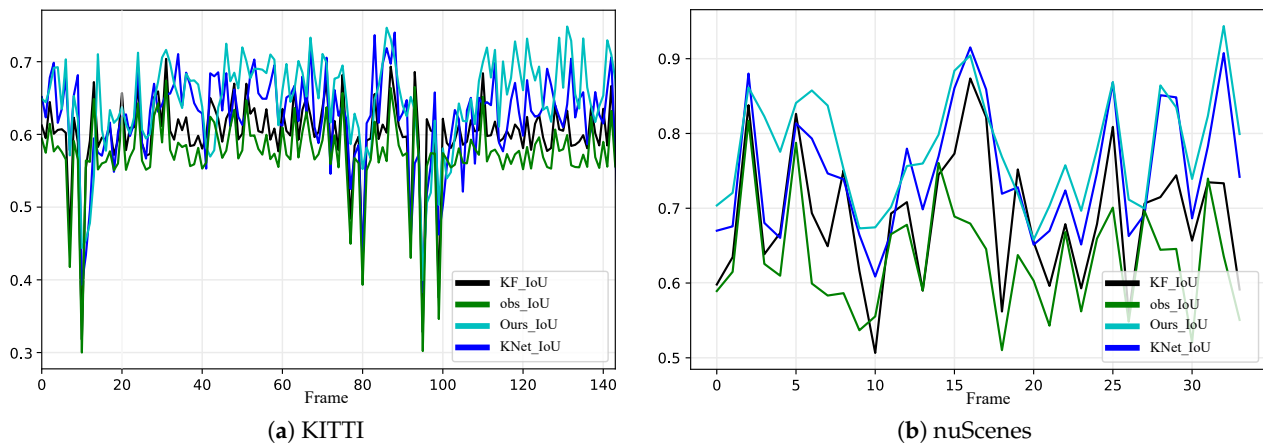


Figure 3. Per-frame 3D IoU curves of different state estimators on the validation sets. The proposed hybrid KG achieves superior performance and consistently outperforms other competing methods in the majority of frames.

On the nuScenes dataset, Table 1 and Figure 3b confirm that all filtering-based estimators outperform raw detections and that the performance ranking remains consistent with KITTI. The larger, more diverse nuScenes dataset yields more pronounced performance differences among estimators. The KF baseline achieves 43.22% AR and 68.86% IoU. The LSTM estimator outperforms the KF, achieving 44.48% AR and 69.34% IoU. The LSTM-KF achieves 46.71% AR and 70.55% IoU, remaining inferior to KNet, confirming that learning the KG is more effective than learning noise covariances for handling nonlinear motion. KNet improves AR to 48.32% and IoU to 71.39%, while CNPE-KNet further improves AR to 50.74% and IoU to 72.67%. Our hybrid KG advances the AR to 52.95% and IoU to 73.90%. Although the state estimation gains are substantial, the AMOTA improvement from 69.4% to 69.9% is relatively modest, since the association module is a critical component of the tracking pipeline and remained at its baseline configuration in our controlled experiment. This is consistent with the design of a controlled baseline experiment, in which all performance gains arise solely from the estimation module. The measurable AMOTA improvement confirms that enhanced state estimation directly contributes to more accurate tracking. Table 2 presents the average latency of all estimators evaluated on the nuScenes datasets. The KF achieves the lowest latency of 4.35 ms per track per frame, while KNet and LSTM-KF operate at 6.83 and 10.03 ms per track per frame, respectively. The hybrid estimator runs at 9.97 ms per track per frame.

Table 3 summarizes the ablation study conducted on λ_t . Setting $\lambda_t = 0$ eliminates the analytical KG, reducing the method to CNPE-KNet, which yields 45.22% AR and 66.47% IoU on the KITTI dataset, and 50.74% AR and 72.67% IoU on the nuScenes dataset. In contrast, $\lambda_t = 1$ discards the learned KG and retains the standard analytical KG, resulting in 40.73% AR and 64.07% IoU on the KITTI dataset, and 43.22% AR and 68.86% IoU on the nuScenes dataset. Among all fixed scalar values of λ_t , setting $\lambda_t = 0.5$ achieves the optimal trade-off between the analytical KG and the learned KG. It obtains 45.81% AR and 67.45% IoU on the KITTI dataset, and 52.06% AR and 73.43% IoU on the nuScenes dataset, surpassing the performance obtained at both $\lambda_t = 0$ and $\lambda_t = 1$. Our learned λ_t adapts the blending ratio for each frame based on historical information, achieving 46.27% AR and 68.20% IoU on the KITTI dataset, and 52.95% AR and 73.90% IoU on the nuScenes dataset, surpassing all fixed alternatives.

Table 3. Ablation study on different λ_t configurations. $\lambda_t = 0$ and $\lambda_t = 1$ correspond to learned KG (CNPE-KNet) and model-based KG (KF), respectively. The learned time-varying λ_t consistently outperforms all fixed alternatives, demonstrating the benefit of frame-wise adaptive fusion. The arrows $\uparrow$ denote that higher and lower values, respectively, indicate better performance. Bold values indicate the best performance in each column.

Dataset	Method	AR% $\uparrow$	3D IoU% $\uparrow$	AMOTA% $\uparrow$
KITTI	$\lambda_t = 0$	45.22	66.47	31.2
	$\lambda_t = 0.3$	44.96	66.08	31.0
	$\lambda_t = 0.5$	45.81	67.45	31.2
	$\lambda_t = 0.7$	42.34	64.83	30.7
	$\lambda_t = 1$	40.73	64.07	30.6
	Ours	46.27	68.20	31.3
nuScenes	$\lambda_t = 0$	50.74	72.67	69.7
	$\lambda_t = 0.3$	49.29	71.74	69.6
	$\lambda_t = 0.5$	52.06	73.43	69.8
	$\lambda_t = 0.7$	47.15	70.61	69.5
	$\lambda_t = 1$	43.22	68.86	69.4
	Ours	52.95	73.90	69.9

Figure 4 illustrates the temporal variation of the learned λ_t along a typical target trajectory. From frame 1 to frame 3, the target moves at an approximately constant speed along a straight path, while the yaw angle remains nearly unchanged. In this phase, λ_t ranges from 0.5103 to 0.5000, indicating a slight preference for the analytical KG, which is well suited to constant velocity motion. From frame 4 to frame 7, the target continues along the straight path, but the distance between consecutive frames varies, indicating variable-speed motion. λ_t decreases from 0.4961 to 0.4938, indicating a shift toward the learned CNPE-KNet to compensate for the limitations of the CV assumption under varying speeds. After frame 8, the yaw angle begins to change, and the inter-frame distance initially decreases and then increases, suggesting that the target decelerates into a turn and subsequently accelerates upon exiting the turn. In this phase, λ_t decreases to the range of 0.4882 to 0.4958, relying more on the learned KG to capture the nonlinear turning dynamics. The dynamic adjustment of λ_t throughout the sequence confirms that the learned coefficient can steadily adapt to various motion patterns.

Although the hybrid KG improves estimation in most frames, its data-driven component remains dependent on the accuracy of incoming observations. Consequently, severe observation errors remain the primary cause of failure cases. In Figure 5a, large position drifts in the noisy observations at frames 21 and 26 bias the estimated state, resulting in bounding boxes that deviate from the ground truth. At frames 23 and 29, inaccurate observation angles adversely affect the yaw estimate. A similar effect occurs in Figure 5b, where the observation angle error at frame 28 degrades the heading estimation. These cases demonstrate that the learned component remains sensitive to severe observation outliers.

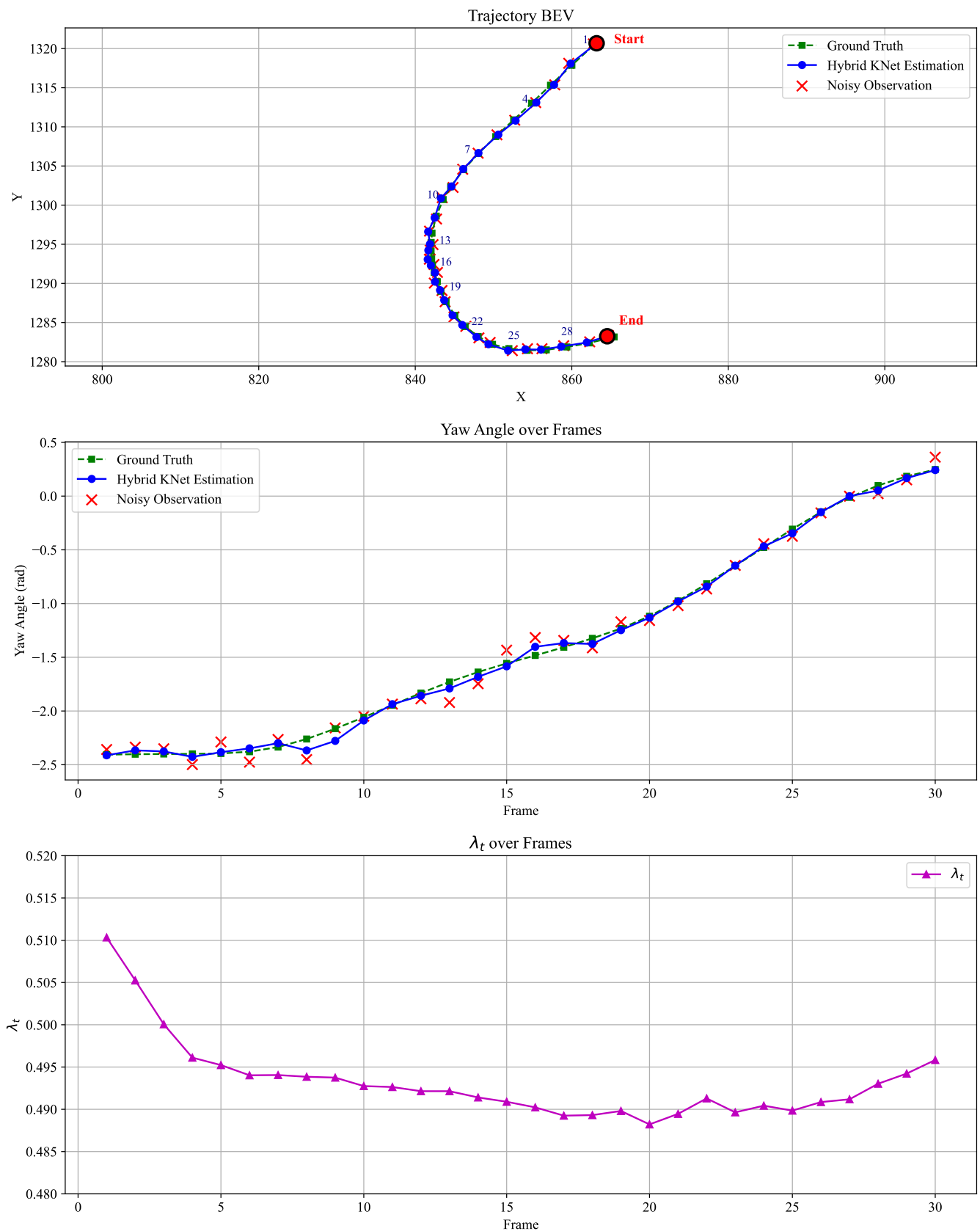


Figure 4. The single-target bird's-eye view (BEV) trajectory, frame-wise yaw angle estimation, and the time-varying fusion weight λ_t are obtained using the proposed hybrid KNet, where the numbers in the first subfigure denote the corresponding frame indices.

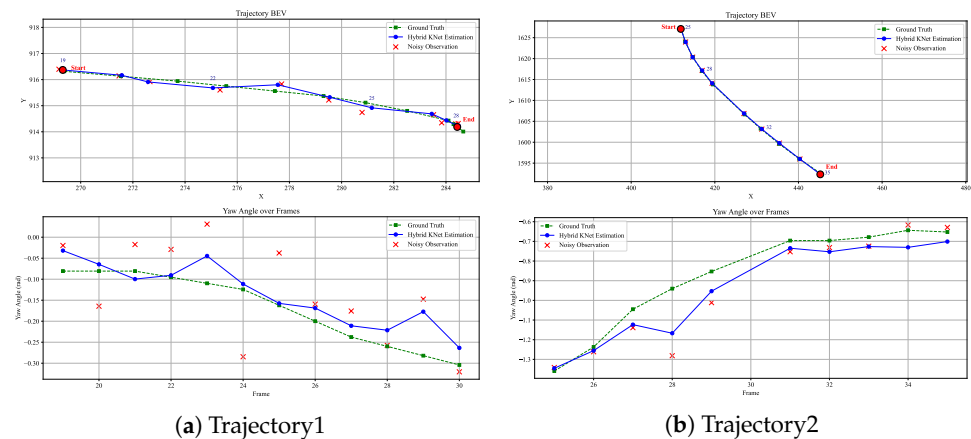


Figure 5. Qualitative failure cases. In Trajectory 1, positional drifts at frames 21 and 26 result in state-estimation bias, while the observation angle error at frames 23 and 29 reduce yaw estimation. In Trajectory 2, an inaccurate observation angle at frame 28 degrades estimation. These cases demonstrate the learned gain’s sensitivity to severe observation outliers, where the numbers denote the corresponding frame indices.

5. Conclusions

To mitigate the KF’s poor adaptability to nonlinear motion and the cross-dimensional noise crosstalk of KNet in 3D MOT, this paper presents a hybrid KG strategy integrated with CNPE-KNet. We embed the LAKF into the 3D TBD framework, using RNNs to learn the system dynamics and time-varying noise. We construct a hybrid KG by combining the model-based analytical KG and the data-driven KG and introducing physical structural constraints to reduce noise crosstalk. The proposed method replaces the traditional KF in the 3D TBD framework as a plug-and-play module, requiring no modifications to the detection and association modules. Experimental results on the KITTI and nuScenes datasets show that our method outperforms the KF, KNet, LSTM, LSTM-KF, and CNPE-KNet in state estimation and overall tracking accuracy. Both qualitative visual results and quantitative metrics demonstrate the effectiveness of the hybrid design.

Author Contributions: Conceptualization, Y.L., Q.F. and P.H.; methodology, Y.L., J.S. and Y.X.; software, Y.L. and J.S.; validation, Q.F.; formal analysis, L.B.; investigation, Y.X.; resources, J.S. and P.H.; data curation, J.S. and P.H.; writing—original draft preparation, Y.L.; writing—review and editing, Q.F.; visualization, Y.X. and L.B.; supervision, Q.F.; project administration, Q.F. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All experimental evaluations in this paper are conducted on the public KITTI (<https://www.cvlibs.net/datasets/kitti/> (accessed on 25 June 2026)) and nuScenes (<https://www.nuscenes.org/> (accessed on 25 June 2026)) datasets, which can be accessed freely for academic use after registration. No original research data were produced during this study.

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

References

- Peng, Z.; Liu, W.; Ning, Z.; Zhao, Q.; Cheng, S.; Hu, J. 3D Multi-Object Tracking in Autonomous Driving: A Survey. In *Proceedings of the 2024 36th Chinese Control and Decision Conference (CCDC)*; IEEE: New York, NY, USA, 2024; pp. 4964–4971. [\[CrossRef\]](#)
- Pang, Z.; Li, Z.; Wang, N. SimpleTrack: Understanding and Rethinking 3D Multi-Object Tracking. In *Proceedings of the Computer Vision—ECCV 2022 Workshops*; Springer: Cham, Switzerland, 2022; pp. 680–696. [\[CrossRef\]](#)
- Wojke, N.; Bewley, A.; Paulus, D. Simple Online and Realtime Tracking with a Deep Association Metric. In *Proceedings of the 2017 IEEE International Conference on Image Processing (ICIP)*; IEEE: New York, NY, USA, 2017; pp. 3645–3649. [\[CrossRef\]](#)
- Feichtenhofer, C.; Pinz, A.; Zisserman, A. Detect to Track and Track to Detect. In *Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV)*; IEEE: New York, NY, USA, 2017; pp. 3057–3065. [\[CrossRef\]](#)
- Meinhardt, T.; Kirillov, A.; Leal-Taixe, L.; Feichtenhofer, C. TrackFormer: Multi-Object Tracking with Transformers. In *Proceedings of the 2022 IEEE Conference on Computer Vision and Pattern Recognition*; IEEE: New York, NY, USA, 2022; pp. 8834–8844. [\[CrossRef\]](#)
- Kalman, R.E. A New Approach to Linear Filtering and Prediction Problems. *J. Basic Eng.* **1960**, *82D*, 35–45. [\[CrossRef\]](#)
- Weng, X.; Wang, J.; Held, D.; Kitani, K. 3D Multi-Object Tracking: A Baseline and New Evaluation Metrics. In *Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*; IEEE: New York, NY, USA, 2020; pp. 10359–10366. [\[CrossRef\]](#)
- Chiu, H.K.; Li, J.; Ambrus, R.; Bohg, J. Probabilistic 3D Multi-Modal, Multi-Object Tracking for Autonomous Driving. In *Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA)*; IEEE: New York, NY, USA, 2021; pp. 14227–14233. [\[CrossRef\]](#)
- Jiang, C.; Zhang, M.; Wang, Y.; Zhang, A. AHMOT: Adaptive Kalman Filtering and Hierarchical Data Association for 3-D Multiobject Tracking in IoT-enabled Autonomous Vehicles. *IEEE Internet Things J.* **2025**, *12*, 21290–21303. [\[CrossRef\]](#)
- Zhang, H.; Guo, B.; Jin, L.; Yue, C.; Liu, X.; Shi, Y.; Huang, T.; He, Y. A Dual-Adaptive Framework for 3D Multi-Object Tracking with Fast Hierarchical Association in IoT Intelligent Vehicles. *Internet Things* **2026**, *37*, 101942. [\[CrossRef\]](#)
- Xie, C.; Lin, C.; Zheng, X.; Gong, B.; M. López, A. Multi-Modal 3D Multi-Object Tracking with Robust Association and Track Drift Compensation. *Neurocomputing* **2025**, *657*, 131687. [\[CrossRef\]](#)
- Ding, S.; Schneider, L.; Cordts, M.; Gall, J. ADA-track++: End-to-End Multi-Camera 3D Multi-Object Tracking with Alternating Detection and Association. *IEEE Trans. Pattern Anal. Mach. Intell.* **2026**, *48*, 482–499. [\[CrossRef\]](#) [\[PubMed\]](#)
- Altaf, M.A.; Kim, M.Y. Gaussian Mixture Model-Based Data Association Incorporating a Deep Learning Network for Multivehicle Tracking and Detection in Autonomous Driving Systems. *Adv. Intell. Syst.* **2026**, *8*, e202500100. [\[CrossRef\]](#)
- Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [\[CrossRef\]](#) [\[PubMed\]](#)
- 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)*; Moschitti, A., Pang, B., Daelemans, W., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2014; pp. 1724–1734. [\[CrossRef\]](#)
- Coskun, H.; Achilles, F.; DiPietro, R.; Navab, N.; Tombari, F. Long Short-Term Memory Kalman Filters: Recurrent Neural Estimators for Pose Regularization. In *Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV)*; IEEE: New York, NY, USA, 2017; pp. 5525–5533. [\[CrossRef\]](#)
- Jia, C.; Ma, J.; Kou, W.M. Multiple Variational Kalman-GRU for Ship Trajectory Prediction with Uncertainty. *IEEE Trans. Aerosp. Electron. Syst.* **2025**, *61*, 3654–3667. [\[CrossRef\]](#)
- Revach, G.; Shlezinger, N.; Ni, X.; Escoriza, A.L.; Van Sloun, R.J.G.; Eldar, Y.C. KalmanNet: Neural Network Aided Kalman Filtering for Partially Known Dynamics. *IEEE Trans. Signal Process.* **2022**, *70*, 1532–1547. [\[CrossRef\]](#)
- Shlezinger, N.; Revach, G.; Ghosh, A.; Chatterjee, S.; Tang, S.; Imbiriba, T.; Dunik, J.; Straka, O.; Closas, P.; Eldar, Y.C. Artificial Intelligence-Aided Kalman Filters: AI-Augmented Designs for Kalman-Type Algorithms. *IEEE Signal Process. Mag.* **2025**, *42*, 52–76. [\[CrossRef\]](#)
- Song, J.; Mei, W.; Xu, Y.; Fu, Q.; Bu, L. Practical Implementation of KalmanNet for Accurate Data Fusion in Integrated Navigation. *IEEE Signal Process. Lett.* **2024**, *31*, 1890–1894. [\[CrossRef\]](#)
- Song, J.; Mei, W.; Xu, Y.; Fu, Q.; Kou, R.; Bu, L.; Long, Y. LAKALMANTracker: Robust Learning-Aided Kalman Filtering for Multi-Object Tracking. In *Proceedings of the ICASSP 2026—2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*; IEEE: New York, NY, USA, 2026; pp. 3821–3825. [\[CrossRef\]](#)
- Song, J.; Mei, W.; Xu, Y.; Fu, Q.; Kou, R.; Bu, L.; Long, Y. Motion Estimation for Multi-Object Tracking Using KalmanNet with Semantic-Independent Encoding. *Inf. Fusion* **2026**, *136*, 104513. [\[CrossRef\]](#)
- Buchnik, I.; Revach, G.; Steger, D.; Van Sloun, R.J.G.; Routtenberg, T.; Shlezinger, N. Latent-KalmanNet: Learned Kalman Filtering for Tracking From High-Dimensional Signals. *IEEE Trans. Signal Process.* **2023**, *72*, 352–367. [\[CrossRef\]](#)
- Buchnik, I.; Sagi, G.; Leinwand, N.; Loya, Y.; Shlezinger, N.; Routtenberg, T. GSP-KalmanNet: Tracking Graph Signals via Neural-Aided Kalman Filtering. *IEEE Trans. Signal Process.* **2024**, *72*, 3700–3716. [\[CrossRef\]](#)

25. Choi, G.; Park, J.; Shlezinger, N.; Eldar, Y.C.; Lee, N. Split-KalmanNet: A Robust Model-Based Deep Learning Approach for State Estimation. *IEEE Trans. Veh. Technol.* **2023**, *72*, 12326–12331. [[CrossRef](#)]
26. Long, Y.; Mei, W.; Song, J.; Xu, Y.; Fu, Q.; Bu, L. CNPE-KalmanNet: Learning-Aided Kalman Filter With Complex Number Position Encoding. *IEEE Signal Process. Lett.* **2026**, *33*, 1556–1560. [[CrossRef](#)]
27. Geiger, A.; Lenz, P.; Urtasun, R. Are We Ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition*; IEEE: New York, NY, USA, 2012; pp. 3354–3361. [[CrossRef](#)]
28. Caesar, H.; Bankiti, V.; Lang, A.H.; Vora, S.; Liong, V.E.; Xu, Q.; Krishnan, A.; Pan, Y.; Baldan, G.; Beijbom, O. nuScenes: A Multimodal Dataset for Autonomous Driving. In *Proceedings of the 2020 IEEE Conference on Computer Vision and Pattern Recognition*; IEEE: New York, NY, USA, 2020; pp. 11618–11628. [[CrossRef](#)]
29. Kuhn, H.W. The Hungarian method for the assignment problem. *Nav. Res. Logist. Q.* **1955**, *2*, 83–97. [[CrossRef](#)]
30. Shi, S.; Wang, X.; Li, H. PointRCNN: 3D Object Proposal Generation and Detection from Point Cloud. In *Proceedings of the 2019 IEEE Conference on Computer Vision and Pattern Recognition*; IEEE: New York, NY, USA, 2019; pp. 770–779. [[CrossRef](#)]
31. Li, Y.; Qi, X.; Chen, Y.; Wang, L.; Li, Z.; Sun, J.; Jia, J. Voxel Field Fusion for 3D Object Detection. In *Proceedings of the 2022 IEEE Conference on Computer Vision and Pattern Recognition*; IEEE: New York, NY, USA, 2022; pp. 1110–1119. [[CrossRef](#)]
32. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft COCO: Common Objects in Context. In *Proceedings of the Computer Vision—ECCV 2014*; Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T., Eds.; Springer International Publishing: Cham, Switzerland, 2014; pp. 740–755. [[CrossRef](#)]
33. Bernardin, K.; Stiefelhausen, R. Evaluating Multiple Object Tracking Performance: The CLEAR MOT Metrics. *EURASIP J. Image Video Process.* **2008**, *2008*, 246309. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.