DIMM: Decoupled Multi-hierarchy Kalman Filtering via Reinforcement Learning

Jirong Zha [1], Yuxuan Fan[2], Kai Li[1], Han Li[1], Chen Gao[3]*, Xinlei Chen[1]*

[1]Shenzhen International Graduate School, Tsinghua University
[2]The Hong Kong University of Science and Technology (Guang Zhou)
[3]BNRist, Tsinghua University
zhajirong23@mails.tsinghua.edu.cn, yfan546@connect.hkust-gz.edu.cn, {likai24, h-li23}@mails.tsinghua.edu.cn,
chgao96@gmail.com, chen.xinlei@sz.tsinghua.edu.cn

Abstract

State estimation is challenging for target tracking with high maneuverability, as the target's state transition function changes rapidly, irregularly, and is unknown to the estimator. Existing work based on interacting multiple model (IMM) achieves more accurate estimation than single-filter approaches through model combination, aligning appropriate models for different motion modes of the target over time. However, two limitations of conventional IMM remain unsolved. First, the solution space of the model combination is constrained as the target's diverse kinematic properties in different directions are ignored. Second, the model combination weights calculated by the observation likelihood are not accurate enough due to the measurement uncertainty. In this paper, we propose a novel framework, DIMM, to effectively combine estimates from different motion models in each direction, thus increasing the target tracking accuracy. First, DIMM extends the model combination solution space of conventional IMM from a hyperplane to a hypercube by designing a 3D-decoupled multi-hierarchy filter bank, which describes the target's motion with various-order linear models. Second, DIMM generates more reliable combination weight matrices through a differentiable adaptive fusion network for importance allocation rather than solely relying on the observation likelihood; it contains an attention-based twin delayed deep deterministic policy gradient (TD3) method with a hierarchical reward. Experiments demonstrate that DIMM significantly improves the tracking accuracy of existing state estimation methods by $31.61\% \sim 99.23\%$.

Code — https://github.com/zhajirong/DIMM

1 Introduction

As a fundamental problem of perception and robotics (Jian et al. 2024), target tracking plays a critical role in a wide range of applications such as autonomous driving (Li and Jin 2022), urban surveillance (Liu et al. 2022; Ren et al. 2023) , robotic manipulation (Deng et al. 2020), target capture (Xie et al. 2024; Zha et al. 2024; Cheng et al. 2024), and so on (Sun et al. 2024) . However, in cases where the dynamic target is highly maneuverable, the state estimation issue becomes challenging due to the unknown switching of motion models and irregular system process noises (He et al. 2023; Luo, Zhou, and Bu 2024). Therefore, it remains less explored to tackle accurate state estimation for target tracking with unknown dynamics.

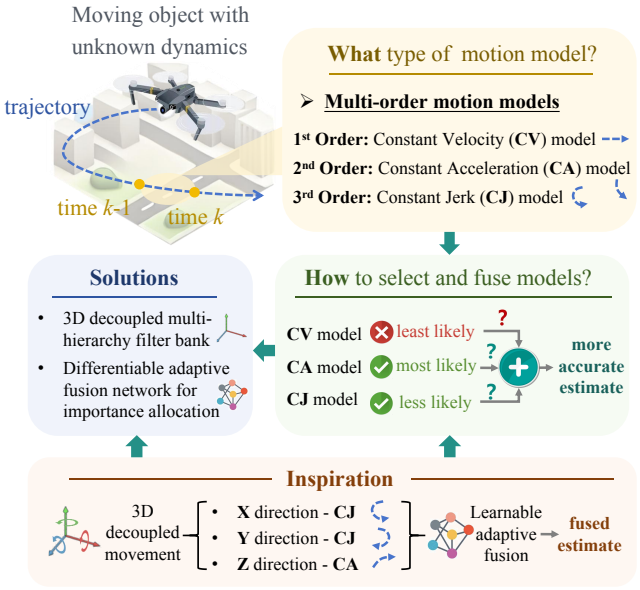


Figure 1: DIMM for target tracking with unknown dynamics. We aim to enhance estimation accuracy by determining the appropriate motion model and independently selecting and fusing models across dimensions.

Existing model-based works (Yang et al. 2025; Dingler 2022; Jiang and Huynh 2017) commonly utilize the Interacting Multiple Model (IMM) (Salvi et al. 2025; Mazor et al. 1998) to deal with targets' motion uncertainties by combining various motion models in a certain ratio (Lee and Park 2023). However, two major limitations of traditional IMM-based methods remain unsolved:

- Planar solution space constraint (**L1**). The traditional IMM algorithm uses direct weighting on the filters' 3D state estimate vectors, limiting the solution space of model combination as the target's kinematic properties may vary in different directions.
- Observation-dependent weight instability (**L2**). The im-

*Corresponding Author.

portance weights for model combination computed by observation likelihood are sensitive to measurement data's quality, since the weight values may be invalid with non-Gaussian distributed noises.

To address these two limitations, we propose a novel framework named Decoupled IMM (DIMM), to deal with accurate target tracking with unknown dynamics by expanding the combination solution space and generating adaptive combination weights. Compared to IMM, DIMM can better approximate the optimal estimate value with a more reasonable basis, *i.e.*, estimate variables from different filtering models and corresponding coefficients, *i.e.*, model combination weights, thus increasing the tracking accuracy.

Specifically, to overcome **L1**, we design a *decoupled multi-hierarchy filter bank* composed of motion models with various orders to realize the 3D decoupling of the state estimate vector, which is theoretically proven to expand the combination solution space and facilitates subsequent independent combination of the state variable in each direction. To overcome **L2**, we propose a *differentiable adaptive fusion network* with reinforcement learning for importance allocation of model fusion by learning the weight matrix from data. Specifically, we improve motion pattern recognition accuracy by independently weighting the estimated variable in each direction. Our contributions are threefold:

- We propose a novel target tracking framework, Decoupled IMM (DIMM), to improve the state estimation accuracy of dynamic targets with high maneuverability by adaptive learning-based fusion of state variables of each dimension independently.
- We design a 3D decoupled multi-hierarchy filter bank to realize the independent linear combination of models' states in different dimensions. We further propose a differentiable adaptive fusion network for importance allocation through attention-based TD3 with a hierarchical reward to generate more accurate combination weights.
- We evaluate DIMM's tracking performance on various collected 3D trajectory datasets, demonstrating its effectiveness in tracking accuracy improvement and excellent generalization.

2 Related Work

Existing work of state estimation for target tracking lies in three categories, model-based, data-driven, and hybrid ones, as illustrated in Fig. 2.

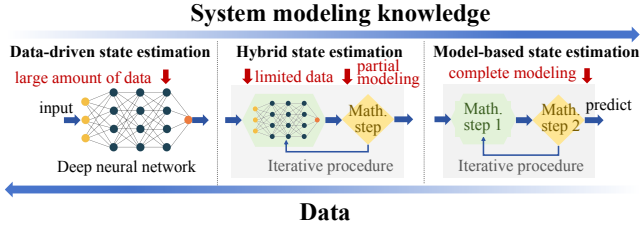


Figure 2: Categories of existing state estimation methods.

Model-based state estimation for target tracking relies on domain knowledge, such as prior understanding of the system's dynamics, including the target's movement function and measurement equation (Chen et al. 2020). The interacting multiple model (IMM) approach (Jilkov, Angelova, and Semerdjiev 1999), designed for highly maneuverable targets with unknown dynamics, improves estimation accuracy by combining multiple models (Akhtar and Habibi 2023). While model-based methods offer interpretability through explicit physical models, their performance can degrade with inaccurate models in complex systems (Chen et al. 2015; Shlezinger et al. 2023a; Yi et al. 2024).

Data-driven state estimation for target tracking with unknown dynamics has gained prominence with the maturation of deep learning, eliminating the need for system modeling knowledge. Recent advancements, including dynamical variational autoencoders (DVAEs) (Krishnan, Shalit, and Sontag 2017) and Kalman variational autoencoders (KVAE) (Fraccaro et al. 2017), enable unsupervised learning by combining VAEs with a linear Gaussian state-space model (Girin et al. 2020). Approaches like the Recurrent Kalman Network (RKN) (Becker et al. 2019) and DANSE (Ghosh, Honoré, and Chatterjee 2024) integrate neural networks with Bayesian techniques, balancing tractability and performance. Data-driven methods can extract features from measurements even when complex systems are hard to model (Shlezinger et al. 2023b). However, they require substantial data and computational resources, and neural networks often lack interpretability (Liu et al. 2023b,a; Tian et al. 2021).

Hybrid state estimation combines model-based and data-driven methods for target tracking with partially known dynamics. KalmanNet (Revach et al. 2022) uses a recurrent neural network (RNN) to model the Kalman gain, trained with supervised learning on true states and noisy measurements. Split-KalmanNet (Choi et al. 2023) addresses state and measurement model mismatches through parallel networks. The optimized KF (OKF) (Greenberg, Yannay, and Mannor 2024) improves estimation by optimizing noise covariance matrices, outperforming the Neural KF (NKF) (de Bézenac et al. 2020) with LSTM models. IMM with DNN has also been explored, such as the LSTM-based IMM (Deng, Li, and Li 2020) and XGBoost-based IMM (Chen and Guestrin 2016), which predict model interaction weights. Hybrid state estimation leverages both model knowledge and data learning, reducing data needs while improving accuracy (Shlezinger et al. 2023b). Our method balances data and model requirements by using a decoupled model-based IMM framework and an enhanced reinforcement learning (RL) module for adaptive model combination.

3 Problem Formulation

In target tracking, the noisy measurements from the sensor serve as input, and the estimated state of the target is the resulting output. A discrete-time target tracking system (Zha et al. 2023) with diverse dynamic models is formulated as

$$\begin{cases} \boldsymbol{x}_k = f^i\left(\boldsymbol{x}_{k-1}\right) + \boldsymbol{w}_{k-1}^i, \\ \boldsymbol{z}_k = h^i\left(\boldsymbol{x}_k\right) + \boldsymbol{v}_k^i, \ \forall i \in \boldsymbol{\mathcal{M}}, \end{cases} \tag{1}$$

where $\boldsymbol{x}_k \in \mathbb{R}^n$ represents the target's state at time step k, $\boldsymbol{z}_k \in \mathbb{R}^m$ denotes the measurement, and $\boldsymbol{\mathcal{M}} = \{m_1, m_2, ..., m_M\}$ is the model set. Function $f^i(\cdot) : \mathbb{R}^n \to \mathbb{R}^n$ specifies the target's state transition equation, and $h^i(\cdot) : \mathbb{R}^n \to \mathbb{R}^m$ is the sensor's measurement equation, both of which vary with the model $i \in \boldsymbol{\mathcal{M}}$[1]. Process noise $\boldsymbol{w}^i_{k-1} \in \mathbb{R}^n$ and measurement noise $\boldsymbol{v}^i_k \in \mathbb{R}^m$ are model-dependent with Gaussian distributions following $\boldsymbol{w}^i_{k-1} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q}^i_{k-1})$ and $\boldsymbol{v}^i_k \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{R}^i_k)$, respectively, where $\boldsymbol{Q}^i_{k-1}$ and $\boldsymbol{R}^i_k$ are the corresponding noise covariance matrices. Specifically, we denote the Markov transition probability of a model jump process from model i to j as π^{ij}. Commonly utilized motion models (Lv et al. 2025) in IMM include constant velocity (CV) model m_{cv} and constant acceleration (CA) model m_{ca} for linear movements, and constant turn rate (CT) model m_{ct} for nonlinear dynamics, which are detailed in the Appendix.

4 Methodology

4.1 Revisiting Interacting Multiple Model

As a popular yet effective way to track the target with high maneuverability, IMM algorithm combines multiple motion models, including CV, CA, and CT models, simultaneously to estimate the target's state, adapting to different movement patterns by weighting each model's predictions based on their likelihood with respect to measurement. Each iteration of the IMM algorithm includes four steps: interaction, filtering, weight generation, and combination, as shown in the Appendix[2]. However, as mentioned above, two critical limitations exist in the combination step of IMM, i.e., the planar solution space constraint (**L1**), and the observation-dependent weight instability (**L2**).

L1: Planar Solution Space Constraint. The conventional IMM algorithm implements direct weighting of the state estimates in all three directions obtained from M different motion models with an M-dimensional combination weight vector. Nonetheless, in cases where the target's motion model differs in each direction, such combination operation is no longer optimal. Actually, the multi-model state estimation can be regarded as a 3D convex optimization problem, while the traditional IMM algorithm restricts the feasible domain to a triangular planar region, extremely limiting the optimizable range of the solution space, as shown in Fig. 3.

Proposition 1. *The solution space of IMM's estimate combination is a hyperplane, while the solution space of 3D model combination weights is a hypercube.*

Proof. The proof is given in the Appendix. □

[1]The state dimension n and measurement dimension m vary with different motion and observation models. In this paper, all models use the target's noisy 3D position as the measurement, as the sensor's observation transformation is not the main focus.

[2]Only the simplest case using the Kalman Filter (KF) is considered in the IMM algorithm, with the linear state transition and measurement functions of model j denoted as $\boldsymbol{F}^j$ and $\boldsymbol{H}^j$, respectively. For details on IMM with more advanced filters like EKF and UKF for nonlinear systems, see (Mazor et al. 1998).

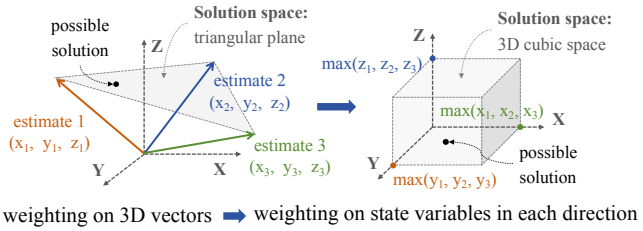


Figure 3: Extend the solution space by shifting the weighting target from state vectors to variables in each direction.

Solution for L1. Existing work based on IMM relies on nonlinear models to describe the target's complex movement, which results in interactions between different dimensional variables, thus preventing the independent model recognition and fusion for each direction. Therefore, we aim to design a multi-hierarchy linear filter bank for various motion models that can realize the 3D decoupling of the target's movements to facilitate the independent linear combination of the state's variable in each direction, which is addressed in Sec. 4.3. Moreover, to cater to the need for expanded 3D combination solution space, we consider a weight matrix rather than a weight vector for more reasonable estimate combination, as specified in Sec. 4.4.

L2: Observation-dependent Weight Instability. As a key component of IMM, the method of generating model combination weights significantly impacts estimation accuracy. Classic IMM algorithms (Mazor et al. 1998) compute weights based on observation likelihood under the Gaussian distribution assumption, which may be inaccurate when measurements are prone to errors, especially with frequent measurement loss, high observation noise, and non-Gaussian noise. Additionally, the manually predefined transition probability function in the IMM interaction step introduces uncertainty, causing instability in model switching at each time step.

Solution for L2. Since a learnable model recognition approach is needed for more accurate model selection and fusion to align with the target's current movement mode, we propose an adaptive fusion network with TD3 (AdaFuse-TD3) in Sec. 4.4 to decide the combination weight matrix rather than relying on the mathematical observation likelihood function for more accurate estimate combination.

4.2 Overview of DIMM

DIMM contains two main modules, a *decoupled multi-hierarchical filter bank* (DHFB) for multi-order local estimation, and a *differentiable adaptive fusion network* (DAFN) for multi-model estimate fusion, as shown in Fig. 4.

- **DHFB module** uses a multi-order motion model group to describe the target's movements, where each model runs a separate KF to generate its local state estimates. Our designed filter bank enables an independent linear combination of the filter's estimate variables in each spatial dimension, thus spanning a larger combination solution space for subsequent estimate fusion.

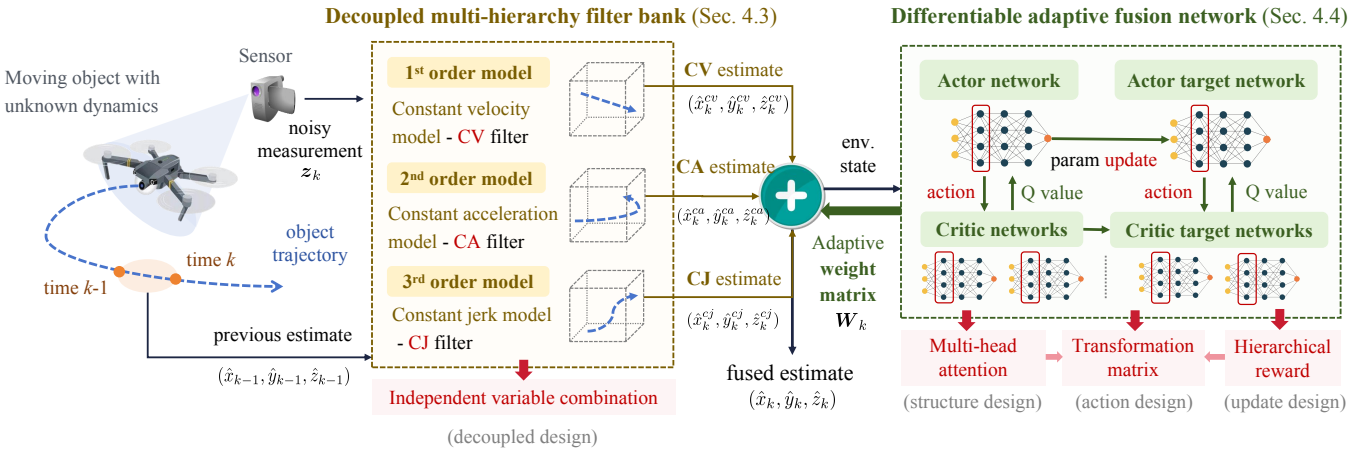


Figure 4: Overview of DIMM. The critical technical contributions are highlighted in red.

- **DAFN module** employs an attention-based TD3 architecture with a hierarchical reward to recognize motion patterns and assign importance weights to each model's estimates for subsequent fusion. Specifically, we take sequential measurements and multi-model estimates as the network input and obtain the transformation matrix of each model as the output to determine the model's combination weights in different dimensions.

Finally, the weighted combination of estimates is able to produce the fused target tracking result. The two modules' innovative design is displayed in Fig. 4. Our proposed DIMM algorithm is specified in the Appendix.

4.3 Decoupled Multi-hierarchical Filter Bank

Our method is built on a 3D-decoupled multi-hierarchy filter bank with a model group $\boldsymbol{\mathcal{M}}_D$, composed of the CV, CA, and constant jerk (CJ) model, to describe the target's movements. By considering various motion models with different orders, our filter bank not only facilitates the independent model combination of dimension-specific motion in each direction, but also provides a more rational representation for highly nonlinear 3D movements than existing methods based on CT model[3], thus improving estimation accuracy.

Specifically, the CV, CA, and CJ model considered in our method correspond to the first-order, second-order, and third-order motion model, respectively[4]. This multi-order model group is completely made up of linear models, which brings convenience for the subsequent separate linear weighted fusion of the state vector's 3D components, hence realizing the state decoupling in three directions.

Therefore, the DHFB module based on the basic linear KF is effective enough for state estimation, further simplifying the algorithm's computation complexity. Then, one obtains the posterior state estimate $\hat{\boldsymbol{x}}_k^i$ of each motion model as

$$\begin{aligned}\hat{\boldsymbol{x}}_k^i =& \hat{\boldsymbol{x}}_{k|k-1}^i + \boldsymbol{K}_k^i(\boldsymbol{z}_k - \hat{\boldsymbol{z}}_k^i) \\ =& f^i(\hat{\boldsymbol{x}}_{k-1}^i) + \boldsymbol{K}_k^i(\boldsymbol{z}_k - h^i(f^i(\hat{\boldsymbol{x}}_{k-1}^i))), \\ & i \in \boldsymbol{\mathcal{M}}_D,\end{aligned} \tag{2}$$

where the model group $\boldsymbol{\mathcal{M}}_D = \{m_{cv}, m_{ca}, m_{cj}\}$, $\hat{\boldsymbol{x}}_{k|k-1}^i$ represents the prior state estimate of model i at time step k, $\hat{\boldsymbol{z}}_k^i$ refers to the predicted measurement, and $\boldsymbol{K}_k^i$ denotes the Kalman gain.

Based on the 3D-decoupled multi-hierarchy filter bank, the task is to identify the optimal order of motion models and amplify their output impact during the combination step to better match the target's movement pattern in each direction.

4.4 Differentiable Adaptive Fusion Network

To address these challenges, instead of using weight vectors based on observation likelihood, we generate a transformation matrix for each model using our RL module, AdaFuse-TD3. This approach adapts the combination weights in three directions at each time step, supporting motion pattern recognition and dynamic adjustment. The fusion network learns from its interactions with the environment, generating transformation matrices that adapt to the target's unpredictable behavior. The transformation matrix serves as an importance metric for each motion model, determining the interaction weight in the model combination.

Environment Definition. The position estimation environment of our proposed AdaFuse-TD3 can be seen as a Markov decision process (MDP) represented by a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P}, \gamma)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{R}$ is the reward, $\mathcal{P}$ is the transition probability distribution, and $\gamma \in [0, 1)$ is the discount factor. We define the environment elements as follows.

State Space $\mathcal{S}$: $\boldsymbol{s}_k = [\boldsymbol{z}_{k-l:k}; \hat{\boldsymbol{p}}_k^{m_{cv}}; \hat{\boldsymbol{p}}_k^{m_{ca}}; \hat{\boldsymbol{p}}_k^{m_{cj}}; \hat{\boldsymbol{p}}_k] \in \mathbb{R}^{15}$. The state of our environment includes the l-length measurement sequence[5], filtered position estimates of the multi-

[3]The conventional model group, including CT models (Singer 2007), relies on idealized assumptions of circular motion, such as a fixed turning rate. These assumptions are unsuitable for highly nonlinear scenarios, like emergency stops or abrupt turns.

[4]The multi-order models' mathematical representation is detailed in the Appendix.

[5]For time step $k < l$, we apply zero-padding (Iwana 2022) to the missing measurement dimensions.

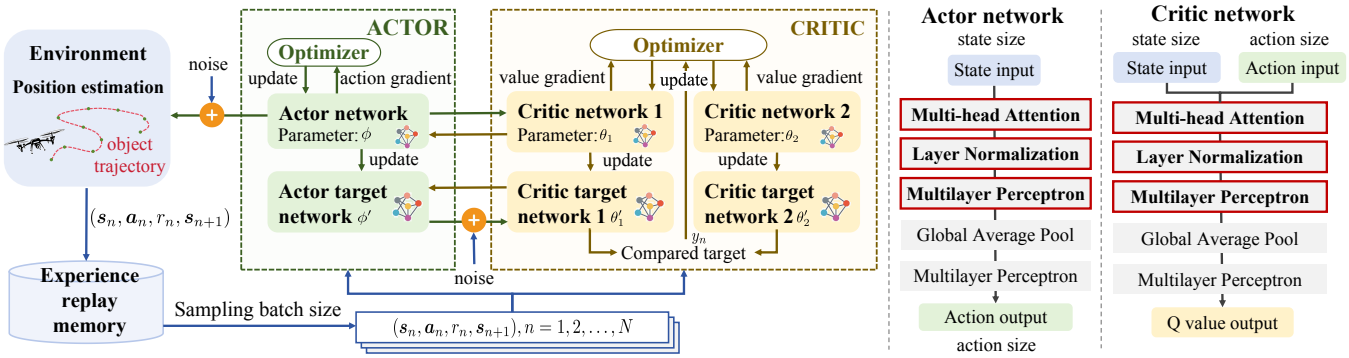


Figure 5: Network structure of the DAFN module.

hierarchy filter bank, and the fused position estimate.

Action Space $\mathcal{A}$: $\boldsymbol{a}_k = [\boldsymbol{a}_{k,x}; \boldsymbol{a}_{k,y}; \boldsymbol{a}_{k,z}] \in \mathbb{R}^9$, where $\boldsymbol{a}_{k,j} = [a_{k,j}^{m_{cv}}, a_{k,j}^{m_{ca}}, a_{k,j}^{m_{cj}}]^{\mathrm{T}}, j \in \{x, y, z\}$. We take the change of the importance weight value of the decoupled multi-hierarchy filter bank as the action and compute the corresponding transformation matrix of each model based on the action values, as given in Sec. 4.4.

Reward $\mathcal{R}$: $r_k \in \mathbb{R}$. We take the difference of the localization error between our algorithm and a benchmark filter as a hierarchical reward, which is detailed in Sec. 4.4.

Agent. We adopt a decision model inspired by TD3 (Fujimoto, Hoof, and Meger 2018) for weight values generation with continuous action space and design an improved network structure as specified in Sec. 4.4.

Attention-based Network Structure. Considering the input measurements are time-sequential and inter-correlated, we build the actor-critic network based on the multi-head attention mechanism (Vaswani 2017) to effectively capture long-range motion patterns. The network architecture is depicted in Fig. 5. Unlike LSTM networks (Hochreiter 1997) that process sequences sequentially, our attention-based structure enables parallel processing of temporal dependencies across the entire sequence. Then, the attention-encoded motion features are fed into subsequent multilayer perceptrons to generate the importance weight matrices.

Transformation Matrix Construction. To facilitate the decoupled combination of filters' position estimates in 3D space, we construct a diagonal transformation matrix for each model as

$$\boldsymbol{T}_k^i = \mathrm{diag}\{w_{k,x}^i, w_{k,y}^i, w_{k,z}^i\}, i \in \boldsymbol{\mathcal{M}}_D, \tag{3}$$

where the 3D importance weight matrix follows

$$\boldsymbol{W}_k = \begin{pmatrix} (\boldsymbol{w}_{k,x})^{\mathrm{T}} \\ (\boldsymbol{w}_{k,y})^{\mathrm{T}} \\ (\boldsymbol{w}_{k,z})^{\mathrm{T}} \end{pmatrix} = \begin{pmatrix} w_{k,x}^{m_{cv}} & w_{k,x}^{m_{ca}} & w_{k,x}^{m_{cj}} \\ w_{k,y}^{m_{cv}} & w_{k,y}^{m_{ca}} & w_{k,y}^{m_{cj}} \\ w_{k,z}^{m_{cv}} & w_{k,z}^{m_{ca}} & w_{k,z}^{m_{cj}} \end{pmatrix}. \tag{4}$$

Specifically, the weight value of model $i \in \boldsymbol{\mathcal{M}}_D$ in each direction $j \in \{x, y, z\}$ is generated as a constant between $[0, 1]$ through a softmax function according to

$$w_{k,j}^i = \frac{e^{a_{k,j}^i - \|\boldsymbol{a}_{k,j}\|_\infty}}{\sum_{j \in \{x,y,z\}} e^{a_{k,j}^i - \|\boldsymbol{a}_{k,j}\|_\infty}}, \tag{5}$$

where $\boldsymbol{a}_k = [\boldsymbol{a}_{k,x}; \boldsymbol{a}_{k,y}; \boldsymbol{a}_{k,z}] \in \mathbb{R}^9$ denotes the action vector as defined in Sec. 4.4. Then, we combine the estimate of each model based on its corresponding transformation matrix and obtain the fused position estimate as

$$\hat{\boldsymbol{p}}_k = \sum_{i \in \boldsymbol{\mathcal{M}}_D} \boldsymbol{T}_{k-1}^i \hat{\boldsymbol{p}}_k^i, \tag{6}$$

where $\hat{\boldsymbol{p}}_k^i$ is the position variable in estimate $\hat{\boldsymbol{x}}_k^i$. Note that only the communal position estimate of the multi-hierarchy filter bank is considered for combination in this paper to simplify the fusion process, as the state dimension varies with motion models[6].

Hierarchical Reward Design. Most existing RL-aided KF research (Gao et al. 2020; Tang et al. 2021) uses the negative localization error as a reward according to

$$r_k = - \parallel \boldsymbol{p}_k - \hat{\boldsymbol{p}}_k \parallel_2, \tag{7}$$

where $\boldsymbol{p}_k$ is ground truth of the target's position. However, the estimation error in Eq. 7 is highly susceptible to unknown environmental noises, which may cause instability to the convergence of reward. Therefore, we intend to reduce the reward variance by designing a hierarchical reward calculated from the difference between the filtering error of AdaFuse-TD3 and that of another benchmark filtering result and feeding it back as an advantageous signal into the network. Specifically, we define the hierarchical reward as:

$$r_k = - \parallel \boldsymbol{p}_k - \hat{\boldsymbol{p}}_{k,\text{AdaFuse-TD3}} \parallel_2 + \parallel \boldsymbol{p}_k - \hat{\boldsymbol{p}}_{k,\text{IMM}} \parallel_2, \tag{8}$$

where $\hat{\boldsymbol{p}}_{k,\text{AdaFuse-TD3}}$ denotes the position estimation results of our filtering method based on AdaFuse-TD3, and $\hat{\boldsymbol{p}}_{k,\text{IMM}}$ is the estimate obtained from the non-learning IMM approach. In this case, the larger the reward value, the higher the estimation accuracy of the filtering method based on AdaFuse-TD3. In summary, the hierarchical reward design weakens the effect of ambient noises, thus flattening the signal wave and improving the model convergence.

5 Experiment

5.1 Experimental Setup

This section introduces the datasets, baselines, and metrics utilized in this paper.

[6]For a more rigorous combination of state variables with unequal dimensions across motion models, see (Zubača et al. 2022).

OKF dataset (Greenberg, Yannay, and Mannor 2024) is a driving trajectory dataset consisting of segments with diverse accelerations and turn radius. **Multi-model dataset** is a self-built target trajectory dataset composed of random combinations of trajectory sequences generated by different motion models' dynamics, including CV, CA, CJ, and CT models.

Flightmare dataset is a drone trajectory dataset featuring randomly generated velocities in three directions, collected from Flightmare[7] (Song et al. 2021), a versatile and high-fidelity quadrotor platform for real-world validation. We select it as drone is an important application scenario (Wang et al. 2025).

KF (Kalman 1960) is a classic state estimation method.

IMM (Mazor et al. 1998) is a famous technique for target tracking with unknown switching dynamic models.

RKN (Becker et al. 2019) (Recurrent Kalman Network) is an end-to-end learning approach for KF.

DANSE (Ghosh, Honoré, and Chatterjee 2024) (Data-driven State Estimation) is the state-of-the-art model-free method.

LSTM-IMM (Deng, Li, and Li 2020) is a LSTM-based IMM method.

XGBoost-IMM (Li, Zhang, and Li 2021) is an IMM method based on XGBoost (Chen and Guestrin 2016).

OKF (Greenberg, Yannay, and Mannor 2024) is an optimized KF with parameter learning.

Mean squared error (MSE) is an evaluation metric suitable for undesirable large-error cases (Arulampalam et al. 2002).

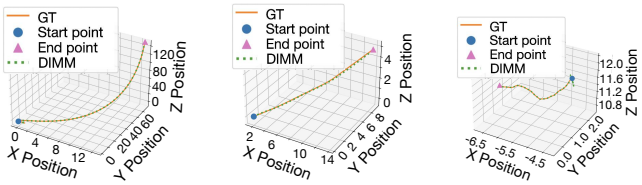
Mean absolute error (MAE) is an indicator of estimation accuracy preferable for required robustness to outliers.

5.2 Estimation Accuracy

Finding 1: DIMM outperforms existing state-of-the-art state estimation methods in terms of estimation accuracy. To quantify the algorithm's performance on target tracking accuracy, the MSE and MAE of estimates obtained from baselines and DIMM are compared in Tab. 1. Results show that the performance of model-based algorithms like KF and IMM varies with datasets. Specifically, the UKF-based IMM fails in the OKF dataset due to the numerical sensitivity, suggesting its significant reliance on the operating scenario (Seah and Hwang 2011). From Tab. 1, DIMM outperforms existing state-of-the-art estimation methods, confirming its effectiveness in accurate target tracking.

Finding 2: The estimate results of DIMM approximate the true values well. We compare the target's ground-truth and estimated trajectory of DIMM in Fig. 6. It can be seen that our algorithm effectively fits the complex 3D motion trajectory of the target with unknown dynamics. To further validate the feasibility of our algorithm's position estimates, Fig. 7 compares the true target position states with the estimated ones obtained from DIMM. As shown, the estimated position variables converge to the ground-truth values, indicating DIMM is applicable to nonlinear target tracking.

[7]The trajectory from Flightmare is considered realistic, as it incorporates practical factors like the drone's dynamic characteristics during motion generation.



(a) Multi-model. (b) OKF. (c) Flightmare.

Figure 6: Examples of comparison between the actual and estimated trajectories of DIMM for different datasets.

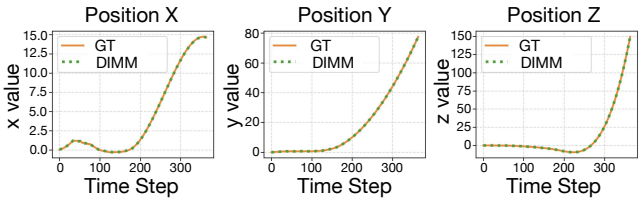


Figure 7: Examples of comparison between the actual and estimated position state variables of DIMM.

5.3 Study of Hierarchical Reward

Hierarchical reward design improves the estimation accuracy. To validate the effectiveness of the reward design illustrated in Sec. 4.4, we compare the tracking accuracy of our algorithm with and without the hierarchical term in Tab. 2. Specifically, we refer the reward defined by Eq. 7 as the simple reward, and our hierarchical reward is given in Eq. 8. It can be seen from Tab. 2 that the hierarchical reward design effectively improves the estimation accuracy.

5.4 Interpretable Analysis

Transformation matrix is for model importance allocation during combination. For more intuitive understanding of the transformation matrix $\boldsymbol{T}_k^i$ demonstrated in Sec. 4.4, we depict this diagonal matrix of each model $i \in \boldsymbol{\mathcal{M}}_D$ at given time steps in Fig. 8. As seen, the diagonal elements of each model's transition matrix correspond to the fusion weights of each filter's estimate, i.e. the values of $(w_{k,x}^i, w_{k,y}^i, w_{k,z}^i)$ in Eq. 3. According to Eq. 6, the greater the weight value of the filter with its corresponding model in one direction, the more significant its estimate is during model combination in that direction. Therefore, one can deduce the most appropriate model type of the moving target for each direction of the 3D space in the designed multi-hierarchy filter bank from the transformation matrix of each model at each time step, as analyzed in Fig. 8.

5.5 Inference Efficiency

DIMM demonstrates impressive inference efficiency. Running on a single A800 GPU, DIMM processes batches of 256 in just 22 ms with a 2000 MiB memory footprint. Its high throughput and low latency make it well-suited for real-time and large-scale tasks, particularly in autonomous systems. Overall, DIMM's efficiency in both speed and memory usage underscores its scalability and competitive advantage in applications demanding rapid and accurate tracking.

Datasets	Metrics	Model-based methods		Data-driven methods		Hybrid methods			
		KF	IMM	RKN	DANSE	LSTM-IMM	XGBoost-IMM	OKF	DIMM (ours)
OKF data	MSE	5.1771	-	0.6132	0.6408	0.9053	3.5254	3.9890	**0.4431**
	MAE	3.1713	-	0.1835	0.1687	0.2052	2.6929	1.6090	**0.1124**
Multi-model data	MSE	2.7535	2.0290	0.7442	0.0310	1.8879	3.4526	3.9509	**0.0041**
	MAE	2.1202	1.7635	0.1373	0.1430	4.6926	2.2531	1.5643	**0.0542**
Flightmare data	MSE	129.0360	129.5573	1.7353	1.6920	2.9830	5.5448	7.8423	**1.4934**
	MAE	101.5394	102.2903	1.1978	1.2630	4.0271	3.7056	3.2317	**1.0100**

Table 1: Comparison of estimation errors of DIMM with seven baselines, averaged over 100 randomized trials.

Reward design	Metrics	OKF data	Multi-model data	Flightmare data
DIMM (simple)	MSE	0.5389	0.1656	7.9213
	MAE	0.5281	0.3507	2.5474
DIMM (hierarchical)	MSE	**0.4431**	**0.0041**	**1.4934**
	MAE	**0.1124**	**0.0542**	**1.0100**

Table 2: Estimation errors of DIMM with different rewards.

Action space	OKF data		Multi-model data		Flightmare data	
	MSE	MAE	MSE	MAE	MSE	MAE
$(-5, 5)$	0.4622	0.1563	**0.0041**	**0.0542**	**1.4934**	**1.0100**
$(-4, 4)$	0.4512	0.1353	0.0065	0.0735	1.6404	1.0154
$(-3, 3)$	0.4478	0.1276	0.0188	0.1059	1.6134	1.0131
$(-2, 2)$	**0.4431**	**0.1124**	0.0082	0.0684	1.5890	1.0153
$(-1, 1)$	0.4519	0.1483	0.0229	0.1121	1.6400	1.0153

Table 3: Estimation errors for different action space.

Module setting	Metrics	OKF data	Multi-model data	Flightmare data
DIMM (w/o DAFN)	MSE	3.7969	1.9824	129.0346
	MAE	2.3349	1.7045	101.5391
DIMM (w/ DAFN)	MSE	**0.4431**	**0.0041**	**1.4934**
	MAE	**0.1124**	**0.0542**	**1.0100**

Table 4: Estimation errors of DIMM w/ and w/o DAFN.

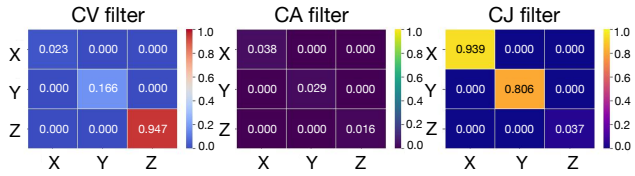
5.6 Ablation Study

The action space size affects estimation accuracy. In our case, the action space size corresponds to the range of combination weight values in the transformation matrices. It determines the granularity of available actions, impacting the accuracy of weight values. As shown in Tab. 3, a larger action space offers finer weight values but increases training complexity, while a smaller action space may limit the model's ability to learn nuanced behaviors.

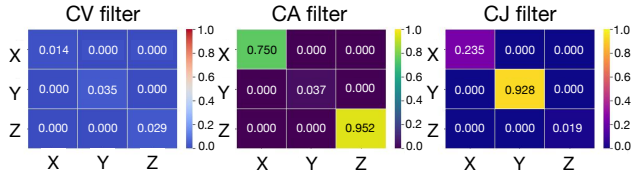
The DAFN module significantly improves estimation accuracy. The DAFN module in DIMM determines the combination weights for each model's estimate in each direction. From Tab. 4, incorporating the DAFN module improves performance across all datasets, with notable reductions in both MSE and MAE. Specifically, the MSE in DIMM with DAFN is reduced by 88.33%, 99.79%, and 98.84% compared to the version without DAFN, highlighting its crucial role in enhancing accuracy. This confirms the benefits of learning-based fusion weight over mathematical formula-based ones.

6 Conclusion and Future Work

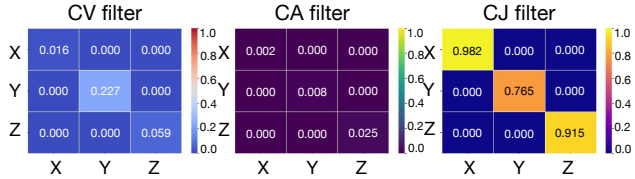
This paper proposes DIMM, a novel target tracking framework for accurate tracking with unknown dynamics. DIMM features a decoupled multi-hierarchy filter bank for multi-order local estimation, expanding the model combination solution space, and a differentiable adaptive fusion network for more accurate weight generation. Evaluation on multiple datasets shows significant improvements in tracking accuracy over SOTA approaches. Future work will focus on integrating the algorithm with advanced AI state evaluation methods like world models (Zhao et al. 2025).



(a) Transformation matrix of each model at one time step on the OKF dataset. It can be seen from the maximum diagonal elements of the transformation matrices that the best-fit models for X, Y, and Z directions are CJ, CJ, and CV model, respectively.



(b) Transformation matrices on the Flightmare dataset. The best-fit models for X, Y, and Z directions are CA, CJ, and CA model, respectively.



(c) Transformation matrices on our dataset. The best-fit models for X, Y, and Z directions are all CJ model, demonstrating our algorithm's applicability to isotropic single motion pattern.

Figure 8: Examples of the transformation matrix of each motion model's filter in the DHFB module.

Acknowledgments

This paper was supported by the Natural Science Foundation of China under Grant 62371269, Shenzhen Low-Altitude Airspace Strategic Program Portfolio Z253061 and Meituan Academy of Robotics Shenzhen. Sponsored by Tsinghua University-Toyota Research Center.

References

Akhtar, S.; and Habibi, S. 2023. The interacting multiple model smooth variable structure filter for trajectory prediction. *IEEE transactions on intelligent transportation systems*, 24(9): 9217–9239.

Arulampalam, M. S.; Maskell, S.; Gordon, N.; and Clapp, T. 2002. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on signal processing*, 50(2): 174–188.

Becker, P.; Pandya, H.; Gebhardt, G.; Zhao, C.; Taylor, C. J.; and Neumann, G. 2019. Recurrent Kalman networks: Factorized inference in high-dimensional deep feature spaces. In *International Conference on Machine Learning (ICML)*, 544–552.

Chen, T.; and Guestrin, C. 2016. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794.

Chen, X.; Purohit, A.; Dominguez, C. R.; Carpin, S.; and Zhang, P. 2015. Drunkwalk: Collaborative and adaptive planning for navigation of micro-aerial sensor swarms. In *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*, 295–308.

Chen, X.; Ruiz, C.; Zeng, S.; Gao, L.; Purohit, A.; Carpin, S.; and Zhang, P. 2020. H-DrunkWalk: Collaborative and adaptive navigation for heterogeneous MAV swarm. *ACM Transactions on Sensor Networks (TOSN)*, 16(2): 1–27.

Cheng, Y.; Zha, J.; Yang, R.; Sun, Z.; Xu, S.; and Chen, X. 2024. Multi-Agent Target Pursuit Using Perception Uncertainty-Aware Reinforcement Learning. In *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*, 1992–1997.

Choi, G.; Park, J.; Shlezinger, N.; Eldar, Y. C.; and Lee, N. 2023. Split-KalmanNet: A robust model-based deep learning approach for state estimation. *IEEE Transactions on Vehicular Technology*, 72(9): 12326–12331.

de Bézenac, E.; Rangapuram, S. S.; Benidis, K.; Bohlke-Schneider, M.; Kurle, R.; Stella, L.; Hasson, H.; Gallinari, P.; and Januschowski, T. 2020. Normalizing Kalman filters for multivariate time series analysis. *Advances in Neural Information Processing Systems (NeurIPS)*, 33: 2995–3007.

Deng, L.; Li, D.; and Li, R. 2020. Improved IMM algorithm based on RNNs. In *Journal of Physics: Conference Series*, volume 1518, 012055.

Deng, X.; Xiang, Y.; Mousavian, A.; Eppner, C.; Bretl, T.; and Fox, D. 2020. Self-supervised 6D object pose estimation for robot manipulation. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 3665–3671.

Dingler, S. 2022. State estimation with the Interacting Multiple Model (IMM) method. *arXiv preprint arXiv:2207.04875*.

Fraccaro, M.; Kamronn, S.; Paquet, U.; and Winther, O. 2017. A disentangled recognition and nonlinear dynamics model for unsupervised learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 30.

Fujimoto, S.; Hoof, H.; and Meger, D. 2018. Addressing function approximation error in actor-critic methods. In *International Conference on Machine Learning (ICML)*, 1587–1596.

Gao, X.; Luo, H.; Ning, B.; Zhao, F.; Bao, L.; Gong, Y.; Xiao, Y.; and Jiang, J. 2020. RL-AKF: An adaptive Kalman filter navigation algorithm based on reinforcement learning for ground vehicles. *Remote Sensing*, 12(11): 1704.

Ghosh, A.; Honoré, A.; and Chatterjee, S. 2024. DANSE: Data-driven non-linear state estimation of model-free process in unsupervised learning setup. *IEEE Transactions on Signal Processing*.

Girin, L.; Leglaive, S.; Bie, X.; Diard, J.; Hueber, T.; and Alameda-Pineda, X. 2020. Dynamical variational autoencoders: A comprehensive review. *arXiv preprint arXiv:2008.12595*.

Greenberg, I.; Yannay, N.; and Mannor, S. 2024. Optimization or architecture: How to hack Kalman filtering. *Advances in Neural Information Processing Systems (NeurIPS)*, 36.

He, S.; Wu, P.; Li, X.; Bo, Y.; and Yun, P. 2023. Adaptive modified unbiased minimum-variance estimation for highly maneuvering target tracking with model mismatch. *IEEE Transactions on Instrumentation and Measurement*, 72: 1–16.

Hochreiter, S. 1997. Long Short-term Memory. *Neural Computation MIT-Press*.

Iwana, B. K. 2022. On mini-batch training with varying length time series. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 4483–4487. IEEE.

Jian, Z.; Li, Q.; Zheng, S.; Wang, X.; and Chen, X. 2024. Lvcp: Lidar-vision tightly coupled collaborative real-time relative positioning. *arXiv preprint arXiv:2407.10782*.

Jiang, Z.; and Huynh, D. Q. 2017. Multiple pedestrian tracking from monocular videos in an interacting multiple model framework. *IEEE Transactions on Image Processing*, 27(3): 1361–1375.

Jilkov, V.; Angelova, D.; and Semerdjiev, T. A. 1999. Design and comparison of mode-set adaptive IMM algorithms for maneuvering target tracking. *IEEE Transactions on Aerospace and Electronic Systems*, 35(1): 343–350.

Kalman, R. E. 1960. A new approach to linear filtering and prediction problems.

Krishnan, R.; Shalit, U.; and Sontag, D. 2017. Structured inference networks for nonlinear state space models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.

Lee, I. H.; and Park, C. G. 2023. An improved interacting multiple model algorithm with adaptive transition probability matrix based on the situation. *International Journal of Control, Automation and Systems*, 21(10): 3299–3312.

Li, D.; Zhang, P.; and Li, R. 2021. Improved IMM algorithm based on XGBoost. In *Journal of Physics: Conference Series*, volume 1748, 032017.

Li, P.; and Jin, J. 2022. Time3D: End-to-end joint monocular 3D object detection and tracking for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 3885–3894.

Liu, D.; Zhu, X.; Bao, W.; Fei, B.; and Wu, J. 2022. SMART: Vision-based method of cooperative surveillance and tracking by multiple UAVs in the urban environment. *IEEE Transactions on Intelligent Transportation Systems*, 23(12): 24941–24956.

Liu, X.; Li, Q.; Wang, L.; Lin, M.; and Wu, J. 2023a. Data-driven state of charge estimation for power battery with improved extended Kalman filter. *IEEE Transactions on Instrumentation and Measurement*, 72: 1–10.

Liu, Z.-g.; Wang, Z.-k.; Yang, Y.-b.; and Lu, Y. 2023b. A data-driven maneuvering target tracking method aided with partial models. *IEEE Transactions on Vehicular Technology*, 73(1): 414–425.

Luo, C.; Zhou, C.; and Bu, X. 2024. Multi-missile phased cooperative interception strategy for high-speed and highly maneuverable targets. *IEEE Transactions on Aerospace and Electronic Systems*.

Lv, H.; Liu, M.; Liu, P.; Chang, K.; Li, M.; and Piao, C. 2025. Kalman Filter-Based High-Accuracy Indoor Positioning with NLoS Error Mitigation and Multi-Motion Model Switching. *IEEE Transactions on Vehicular Technology*.

Mazor, E.; Averbuch, A.; Bar-Shalom, Y.; and Dayan, J. 1998. Interacting multiple model methods in target tracking: a survey. *IEEE Transactions on Aerospace and Electronic Systems*, 34(1): 103–123.

Ren, J.; Xu, Y.; Li, Z.; Hong, C.; Zhang, X.-P.; and Chen, X. 2023. Scheduling uav swarm with attention-based graph reinforcement learning for ground-to-air heterogeneous data communication. In *Adjunct Proceedings of the 2023 ACM International Joint Conference on Pervasive and Ubiquitous Computing & the 2023 ACM International Symposium on Wearable Computing*, 670–675.

Revach, G.; Shlezinger, N.; Ni, X.; Escoriza, A. L.; Van Sloun, R. J.; and Eldar, Y. C. 2022. KalmanNet: Neural network aided Kalman filtering for partially known dynamics. *IEEE Transactions on Signal Processing*, 70: 1532–1547.

Salvi, A.; Ala, P. S. K.; Smereka, J. M.; Brudnak, M.; Gorsich, D.; Schmid, M.; and Krovi, V. 2025. Online identification of skidding modes with interactive multiple model estimation. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, 3139–3145. IEEE.

Seah, C. E.; and Hwang, I. 2011. Algorithm for performance analysis of the IMM algorithm. *IEEE Transactions on Aerospace and Electronic Systems*, 47(2): 1114–1124.

Shlezinger, N.; Whang, J.; Eldar, Y. C.; and Dimakis, A. G. 2023a. Model-Based Deep Learning. *Proceedings of the IEEE*, 111(5): 465–499.

Shlezinger, N.; Whang, J.; Eldar, Y. C.; and Dimakis, A. G. 2023b. Model-based deep learning. *Proceedings of the IEEE*, 111(5): 465–499.

Singer, R. A. 2007. Estimating optimal tracking filter performance for manned maneuvering targets. *IEEE Transactions on Aerospace and electronic systems*, (4): 473–483.

Song, Y.; Naji, S.; Kaufmann, E.; Loquercio, A.; and Scaramuzza, D. 2021. Flightmare: A flexible quadrotor simulator. In *Conference on Robot Learning*, 1147–1157. PMLR.

Sun, N.; Zhao, J.; Shi, Q.; Liu, C.; and Liu, P. 2024. Moving target tracking by unmanned aerial vehicle: A survey and taxonomy. *IEEE Transactions on Industrial Informatics*, 20(5): 7056–7068.

Tang, Y.; Hu, L.; Zhang, Q.; and Pan, W. 2021. Reinforcement learning compensated extended Kalman filter for attitude estimation. In *Proceedings of the International Conference on Intelligent Robots and Systems (IROS)*, 6854–6859. IEEE.

Tian, J.; Wang, B.; Wang, Z.; Cao, K.; Li, J.; and Ozay, M. 2021. Joint adversarial example and false data injection attacks for state estimation in power systems. *IEEE Transactions on Cybernetics*, 52(12): 13699–13713.

Vaswani, A. 2017. Attention is all you need. *Advances in Neural Information Processing Systems (NeurIPS)*.

Wang, H.; Xu, J.; Luo, X.; Chen, X.; Zhang, T.; Duan, R.; Liu, Y.; and Chen, X. 2025. Ultra-high-frequency harmony: mmwave radar and event camera orchestrate accurate drone landing. In *Proceedings of the 23rd ACM Conference on Embedded Networked Sensor Systems*, 15–29.

Xie, J.; Zhong, B.; Mo, Z.; Zhang, S.; Shi, L.; Song, S.; and Ji, R. 2024. Autoregressive queries for adaptive tracking with spatio-temporal transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 19300–19309.

Yang, F.; Zhong, J.; Luo, Y.; Zhang, Y.; Shen, X.; and Zhu, Y. 2025. Maneuvering target tracking based on a random motion model and integrated random interacting multiple model filtering. *Aerospace Science and Technology*, 110244.

Yi, K.; Luo, K.; Luo, X.; Huang, J.; Wu, H.; Hu, R.; and Hao, W. 2024. Ucmctrack: Multi-object tracking with uniform camera motion compensation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, 6702–6710.

Zha, J.; Han, L.; Dong, X.; and Ren, Z. 2023. Privacy-preserving push-sum distributed cubature information filter for nonlinear target tracking with switching directed topologies. *ISA Transactions*, 136: 16–30.

Zha, J.; Zhou, N.; Liu, Z.; Sun, T.; and Chen, X. 2024. Diffusion-based Filter for Fast and Accurate Collaborative Tracking with Low Data Transmission. *Authorea Preprints*.

Zhao, B.; Tang, R.; Jia, M.; Wang, Z.; Man, F.; Zhang, X.; Shang, Y.; Zhang, W.; Wu, W.; Gao, C.; et al. 2025. AirScape: An Aerial Generative World Model with Motion Controllability. In *Proceedings of the 33rd ACM International Conference on Multimedia*, 12519–12528.

Zubača, J.; Stolz, M.; Seeber, R.; Schratter, M.; and Watzenig, D. 2022. Innovative interaction approach in IMM filtering for vehicle motion models with unequal states dimension. *IEEE Transactions on Vehicular Technology*, 71(4): 3579–3594.