HiGM-mixs: A Lightweight and Efficient Method for Orbital Motion Intention Recognition of Non-Cooperative Space Targets

Yuhang Wang[a], Yuhan Liu[a], Pengyu Wang[a,*], Yanning Guo[a], Robert Fonod[b]

[a] *Department of Control Science and Engineering, Harbin Institute of Technology, Harbin, 150001, China*
[b] *School of Architecture, Civil and Environmental Engineering, École Polytechnique Fédérale de Lausanne, Lausanne, CH-1015, Switzerland*

Abstract

Accurate intention recognition of non-cooperative space targets is critical for spacecraft on-orbit safety, yet existing methods struggle to balance accuracy and model efficiency under onboard computational constraints. This paper proposes a lightweight method for orbital motion intention recognition based on short-window relative-motion observations. First, a physics-guided grouped feature encoder organizes relative-motion variables according to their roles in configuration evolution. Based on this encoder, a lightweight grouped mixer model, termed GM-mixs, is developed to capture temporal dependencies and feature interactions with low computational complexity. Furthermore, a hierarchical extension, termed HiGM-mixs, employs a dual-branch design to jointly exploit raw local motion details and physics-trend information, thereby enhancing local feature extraction and robustness. In addition, a phase-consistent sampling strategy is proposed to improve cross-orbit generalization, while TH relative-motion dynamics are used to evaluate the applicability of HiGM-mixs under eccentric reference-orbit conditions. Finally, extensive experiments demonstrate that HiGM-mixs achieves the highest recognition accuracy of 99.49% among all compared models while

*Corresponding author.

Email addresses: wangyh_hit@163.com (Yuhang Wang), yhliu@hit.edu.cn (Yuhan Liu), wangpy_hit@163.com (Pengyu Wang), guoyn@hit.edu.cn (Yanning Guo), robert.fonod@ieee.org (Robert Fonod)

using only 25.19% of the parameters of the smallest recurrent or attention-based baseline model.

Keywords: Non-cooperative space targets, Intention recognition, Deep learning, Hierarchical time-series architecture

1. Introduction

The rapid advancement of aerospace technology in recent years has accelerated the exploitation of space resources, while also increasing the complexity of the near-Earth orbital environment [1, 2, 3]. Consequently, Space Situational Awareness (SSA) has evolved from basic orbit determination to a comprehensive intelligence framework required to ensure space flight safety [4]. Modern SSA encompasses not only the tracking of space debris and active payloads but also the autonomous perception of target behaviors during complex rendezvous and proximity operations [5]. The evolution rate of the space security environment has surpassed the defense thresholds of traditional orbital early warning systems. Existing early warning mechanisms based primarily on collision probability assessment provide limited information about the orbital motion intentions of unknown targets [6], motivating the development of enhanced behavior-awareness capabilities. Therefore, accurate orbital motion intention recognition of non-cooperative space targets is crucial for ensuring the safe on-orbit operation of spacecraft and can provide essential motion evidence for subsequent task intention inference.

Intention recognition has attracted considerable research attention in the field of artificial intelligence. Conventional intention recognition methods include evidence theory [7, 8], expert systems [9], Bayesian networks [10, 11], and template matching [12]. These methods require substantial prior knowledge and are highly dependent on manually designed features, thereby limiting their generalization capability. With the rise of deep learning technology [13], intention recognition research has increasingly shifted toward neural network-based methods that learn representations directly from data. Recent studies have explored sequence-oriented architectures (e.g., recurrent networks and Transformers) and data augmentation strategies to capture temporal dynamics in intention-related behaviors [14, 15].

Intention recognition of non-cooperative space targets can be naturally treated as a time series classification task, in which intentions are inferred from sequential observations of the target states. Accordingly, recent stud-

ies on space non-cooperative target intention recognition have increasingly explored data-driven sequence modeling methods [16, 17].

Recent studies have investigated intention recognition for non-cooperative space targets, covering both traditional machine learning approaches and deep neural network methods. In [18], an HMM-variant probabilistic sequence model (GMM–HMM) is employed for intention recognition. In [19], a random-forest ensemble method is applied using motion features. More recently, deep learning has been increasingly adopted for intention recognition [13]. In [20], a geometry-based representation of relative motion is obtained based on the Clohessy–Wiltshire (CW) equations and fed into a deep neural network (DNN) classifier for intention recognition. In [21], motion data are converted into image-based representations and recognized using a convolutional neural network (CNN). Motivated by the success of attention-based Transformers in time-series modeling [22, 23], subsequent studies have incorporated self-attention mechanisms and Transformer components into models for intention recognition of non-cooperative space targets. Specifically, some works treat the task as sequence modeling and combine temporal modules such as gated recurrent units (GRUs) and long short-term memory (LSTM) networks with attention mechanisms, including a bidirectional GRU (Bi-GRU) with self-attention [24, 25, 26], a CNN–LSTM with self-attention under multi-constraint settings [27], and hybrid architectures integrated with the Transformer [28, 29]. In addition, large language model (LLM)-based frameworks have been proposed to support intention recognition using multi-source information and prompt-driven inference [30]. Related recent studies have further investigated long-term orbital maneuver intention recognition for non-cooperative satellites [31] and intention recognition for maneuvering spacecraft clusters [32].

It can be observed that many recent models for intention recognition of non-cooperative space targets seek to improve accuracy by stacking multiple components (e.g., convolutional/recurrent blocks with self-attention or Transformer modules), which often increases architectural complexity and computational cost. Particularly, some models employing attention mechanisms and Transformer may introduce more prior constraints due to the design of positional encoding or attention heads, thereby potentially introducing redundant components that limit the model's generalization ability in complex scenarios. Studies [33, 34] have shown that Transformer-based models do not consistently outperform simpler models on widely used time-series forecasting benchmarks.

To reduce the architectural complexity and computational cost of existing methods while improving recognition performance and generalization, this paper proposes a lightweight method for orbital motion intention recognition of non-cooperative space targets from short observation windows. First, a physics-guided grouped feature encoder is designed to organize different orbital variables according to their physical relationships during orbital configuration evolution. Multilayer perceptron (MLP)-based mixing has shown promising performance in time series modeling [35, 36]. Building on this encoding mechanism and MLP-based mixing, a lightweight grouped mixer model, termed GM-mixs, is developed for orbital motion intention recognition from short observation windows.

Inspired by the effectiveness of hierarchical modeling in sequential learning [37, 38], we introduce a dual-branch design into the low-level module and incorporate hierarchical modeling into the proposed architecture. Based on this design, a hierarchical extension of GM-mixs, termed HiGM-mixs, is developed. It enhances local feature extraction and robustness under short-window setting. Specifically, each local chunk consists of a raw motion sequence branch and a physics-trend sequence branch, so that raw local motion details and stable physics-trend information can be jointly exploited. The resulting chunk-level features are then aggregated at a higher level for orbital motion intention recognition. Furthermore, the generalization capability of HiGM-mixs is investigated across different orbital regimes and under eccentric reference-orbit conditions. The main contributions of this work are as follows:

(1) A physics-guided lightweight grouped mixer model, termed GM-mixs, is proposed for orbital motion intention recognition from short observation windows. It organizes the relative-position variables into physically related groups according to their roles in configuration evolution and performs temporal and cross-group interaction modeling with simple MLP-based modules, thereby maintaining a compact architecture with low computational cost.

(2) By incorporating a hierarchical dual-branch chunk modeling architecture, HiGM-mixs is further developed. It jointly models raw local motion information and physics-trend information to strengthen local representation and robustness under short-window conditions while preserving a lightweight architecture.

(3) A phase-consistent sampling strategy is proposed to improve cross-orbit generalization without additional training. Furthermore, the TH

relative-motion dynamics for eccentric reference orbits are used to analyze the eccentricity-induced configuration evolution and recognition results, thereby evaluating the applicability and limitations of HiGM-mixs under eccentric reference-orbit conditions.

The rest of this paper is organized as follows. Section 2 formulates the orbital motion intention recognition problem and defines the associated configuration label space. Section 3 presents the proposed HiGM-mixs model for orbital motion intention recognition. Section 4 presents the experimental setup and evaluates the proposed models through comparative experiments. Section 5 summarizes the methodology and results.

2. Problem Statement

In this section, we formulate the orbital motion intention recognition problem and define the intention space adopted in this work, following existing studies on space target intention recognition [26, 28, 30]. Specifically, fifteen in-plane relative-motion geometric configurations are adopted to define the configuration label space for this task. These labels represent the target's orbital motion intention considered in this study.

2.1. Problem Formulation

This study addresses orbital motion intention recognition of a single non-cooperative space target from short observation windows in an orbital rendezvous scenario. The rendezvous described here refers to scenarios where non-cooperative targets exhibiting anomalous orbital behavior approach the observer spacecraft. This scenario setting follows the rendezvous scenario considered in [26].

The scenario considered in this study is shown in Fig. 1. In this scenario, the observing spacecraft S follows its nominal orbit and does not maneuver under normal conditions. The spacecraft S is assumed to carry onboard sensors (e.g., optical or radar tracking) that provide target relative measurements within a finite detection range. The required relative information is obtained from these measurements and standard state estimation. As the non-cooperative target T approaches S, a sequence of observations is collected over time for subsequent intention recognition. In practical scenarios, the target T may approach the observer spacecraft S through natural drift or maneuvers. In this paper, the term non-cooperative means that the target does not share its state, objective, or control information with the

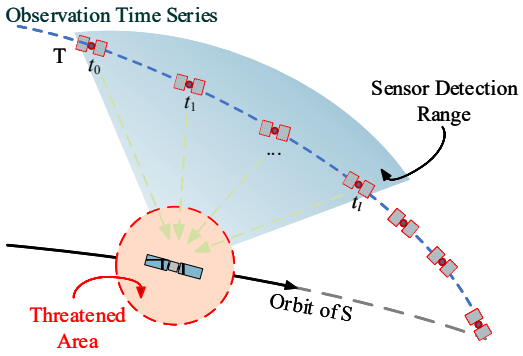

Figure 1: Observation scenario for short-window orbital motion intention recognition of a non-cooperative space target.

observer spacecraft. The recognition task considered in this work focuses on non-maneuvering short observation windows, where the relative trajectory is mainly governed by orbital dynamics. If a maneuver occurs within an observation window, the recognition result can be updated using a new rolling observation window after the maneuver-induced trajectory change is observed.

The recognition problem can be formulated as a mapping from short-window observation time series to predefined relative-motion geometric configuration labels. Let the relative-motion geometric configuration label space be $G_R = \{p_i \mid i \in \mathbb{I}_1^{15}\}$. Given the observation time series $\boldsymbol{\mathcal{X}}$, the recognition model outputs the predicted relative-motion geometric configuration label:

$$\tilde{p} = f(\boldsymbol{\mathcal{X}}), \tilde{p} \in G_R \tag{1}$$

In a broader intention-analysis framework, the predicted configuration label may be combined with complementary multi-source information through an interpretation mapping or inference model $\Gamma(\cdot)$ to infer a higher-level task-intention hypothesis:

$$\hat{I} = \Gamma\left(\tilde{p}, \mathbf{c}_m\right), \hat{I} \in S_{IG} \tag{2}$$

where $\tilde{p}$ denotes the predicted relative-motion geometric configuration label, $\mathbf{c}_m$ denotes complementary multi-source information, and $\hat{I}$ denotes the higher-level intention obtained from this configuration label. The development and validation of $\Gamma(\cdot)$ are beyond the scope of this study; the present

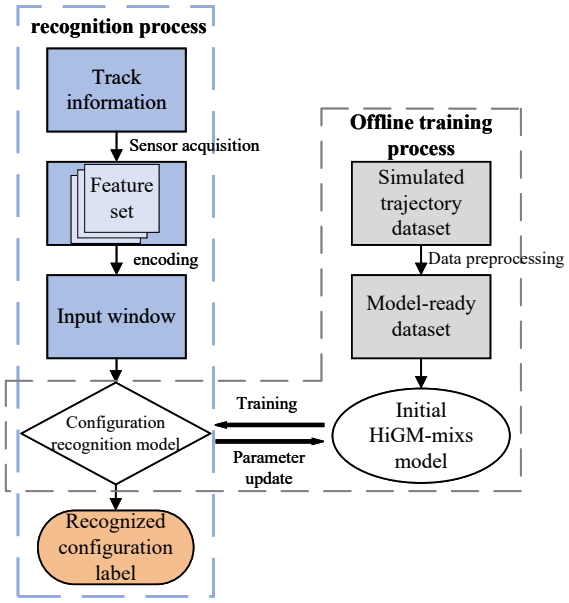

Figure 2: Workflow of the proposed short-window orbital configuration recognition method.

work focuses on learning $f(\cdot)$ for orbital motion intention recognition from short observation windows. The overall recognition workflow is shown in Fig. 2.

2.2. Intention Space

Before constructing the recognition model, the configuration label space used in this task should be specified. In the present work, this definition follows prior studies on space target intention modeling [26, 28, 30]. For non-cooperative space targets, related-motion configuration definitions have been reported in [39]. In the present work, the adopted in-plane label space with 15 types is built on the existing 19 in-plane types reported in the literature. The 15-type label space is obtained by merging several geometrically similar types.

The recognition task considered here focuses on the relative motion of the target within a short observation window. Within this window, the target relative trajectory is governed by natural orbital dynamics and does not include controlled maneuvers. Under this setting, the short-window observation time series corresponds to a relatively stable relative-motion geometric

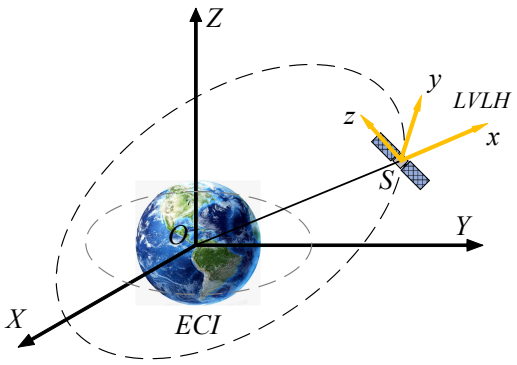

Figure 3: Schematic diagram of the ECI and LVLH coordinate systems.

configuration. If a maneuver occurs within the window, the current observation sequence will no longer satisfy the non-maneuvering assumption. Therefore, the recognition result needs to be updated using subsequent observation windows.

To facilitate the description of the relative motion between spacecraft S and target T, two coordinate systems are introduced here, as shown in Fig. 3. To describe the inertial position of a spacecraft, the Earth-Centered Inertial (ECI) coordinate system is introduced. To describe the relative motion between the target and the spacecraft, the Local-Vertical Local-Horizontal (LVLH) coordinate system is introduced. These two coordinate systems jointly establish the description framework for spacecraft motion in space. The Earth and both spacecraft are treated as point masses, and the separation between S and T is assumed to be much smaller than the distance from S to the center of the Earth. The ECI coordinate system is used as an inertial reference for dynamics analysis. The LVLH coordinate system establishes a local reference system for specific satellite orbits.

The mean motion of spacecraft S in this scenario is $n = \sqrt{\mu/a^3}$, where μ is Earth's standard gravitational parameter and a is the spacecraft orbital radius. The relative position of T is defined as (x, y, z) in the LVLH coordinate system centered at spacecraft S. The governing equations of relative

motion are given by [40]:

$$\begin{cases} \ddot{x} - 2n\dot{y} - 3n^2 x = 0 \\ \ddot{y} + 2n\dot{x} = 0 \\ \ddot{z} + n^2 z = 0 \end{cases} \tag{3}$$

The analytical solution of the CW equations is as follows:

$$\begin{cases} x(t) = A \cdot \sin nt - B \cdot \cos nt + C \\ y(t) = 2B \cdot \sin nt + 2A \cdot \cos nt + D \cdot t + E \\ z(t) = F \cdot \sin nt + G \cdot \cos nt \\ \dot{x}(t) = nB \cdot \sin nt + nA \cdot \cos nt \\ \dot{y}(t) = -2nA \cdot \sin nt + 2nB \cdot \cos nt + D \\ \dot{z}(t) = -nG \cdot \sin nt + nF \cdot \cos nt \end{cases} \tag{4}$$

where

$$\begin{cases} A = \frac{\dot{x}_0}{n} \\ B = 2\frac{\dot{y}_0}{n} + 3x_0 \\ C = 2\left(2x_0 + \frac{\dot{y}_0}{n}\right) \\ D = -3\left(\dot{y}_0 + 2nx_0\right) \\ E = y_0 - \frac{2}{n}\dot{x}_0 \\ F = \frac{\dot{z}_0}{n} \\ G = z_0 \end{cases} \tag{5}$$

and $[x_0, y_0, z_0, \dot{x}_0, \dot{y}_0, \dot{z}_0]^\top$ denotes the state of the non-cooperative target T in the LVLH coordinate system at time t_0.

It should be noted that the CW equations are derived under the assumption of a circular reference orbit. Therefore, the CW-based relative-motion representation used in this work is primarily formulated for short-window relative motion around near-circular reference orbits. As the eccentricity of the reference orbit increases, variations in the orbital angular rate become more pronounced. In this case, model mismatch between the CW-based representation and the actual eccentric-orbit dynamics may cause the observed relative-motion geometry to deviate from the nominal configuration associated with the same initial condition. For high-eccentricity scenarios, model predictions should therefore be evaluated against the actual relative-motion geometry rather than solely against the nominal CW-based labels. Further

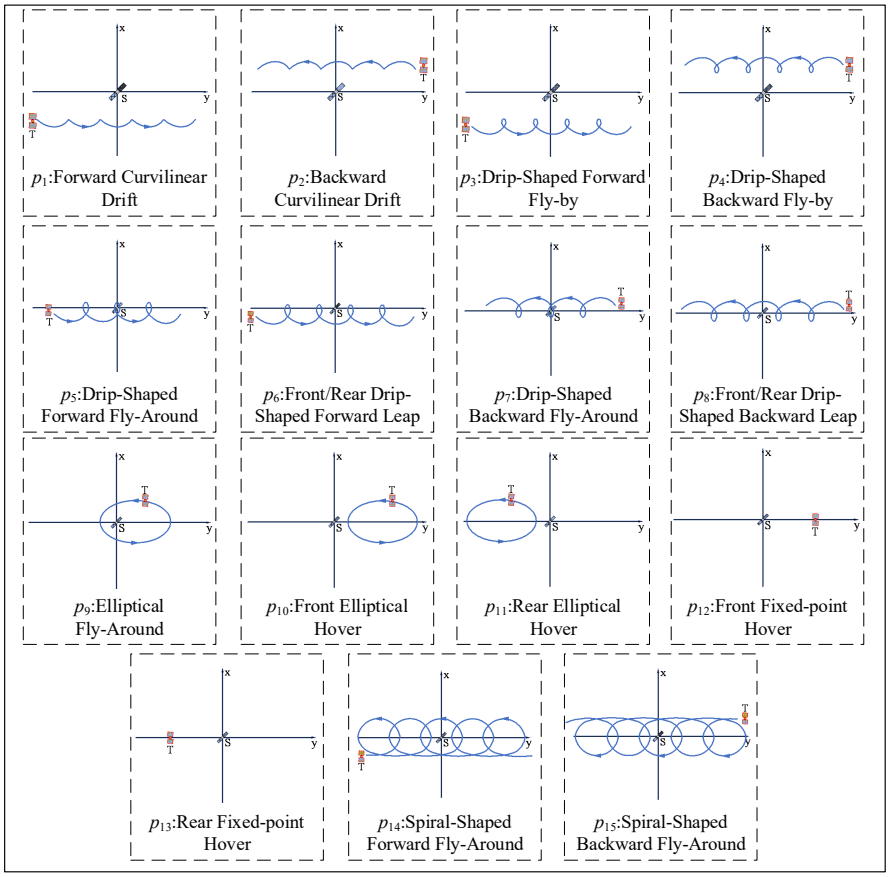

Figure 4: Geometric configurations of relative motion for non-cooperative space targets, summarized from [26].

analysis of eccentricity-induced configuration evolution and the corresponding recognition results is provided in Appendix A.

According to the spacecraft relative motion equations, the relative motion of a space target is a three-dimensional motion with multiple coupled variables. In this work, we focus on the in-plane relative motion. This is because in the CW model, the motion along the z axis is decoupled from the motion in the $x-y$ plane in the relative motion model. Accordingly, the orbital motion intention recognition task considered in this work is formulated primarily in terms of the geometric patterns in the $x-y$ plane.

Following the feature-point-based configuration characterization in [26],

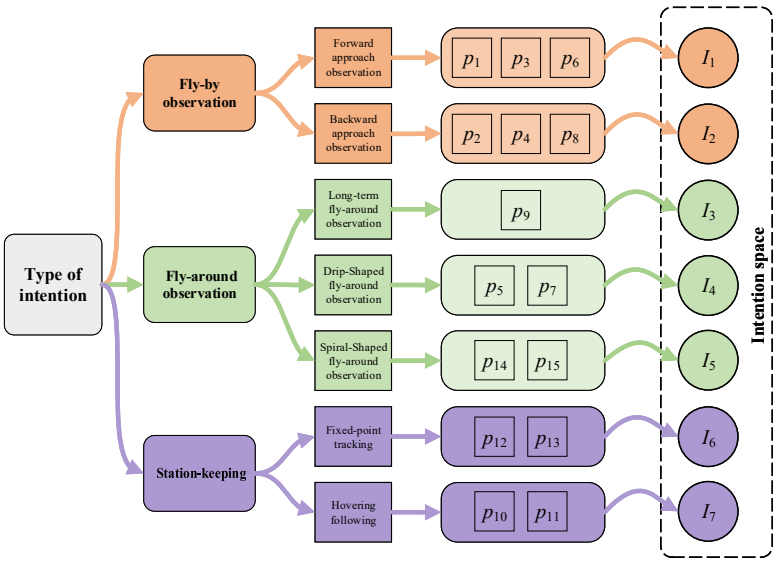

Figure 5: Mapping from geometric configuration labels to higher-level intention.

we adopt the same approach to distinguish different in-plane relative motion configurations. See [26] for details. The resulting relative motion configurations used in this work are shown in Fig. 4. To illustrate the potential role of the recognized configurations in subsequent task-intention inference, a conceptual interpretation of the geometric configuration labels is provided below. This mapping summarizes the task intention into three categories: fly-by observation, fly-around observation, and station-keeping. Each task-intention category may be associated with several relative-motion geometric configurations. The mapping is as follows:

(1) **Fly-by observation**: Encompasses two intention types: forward approach observation (p_1, p_3, p_6) and backward approach observation (p_2, p_4, p_8), which may be associated with tasks such as orbit analysis.

(2) **Fly-around observation**: Includes three intention types: long-term fly-around observation (p_9), drip-shaped fly-around observation (p_5, p_7), and spiral-shaped fly-around observation (p_{14}, p_{15}), primarily serving tasks such as persistent monitoring.

(3) **Station-keeping**: Covers two intention types: fixed-point tracking (p_{12}, p_{13}) and hovering following (p_{10}, p_{11}), which may be associated with tasks such as stationary assessment and fixed-point observation.

The above relative motion trajectory shapes are only considered in the orbital plane, without accounting for motion in the z-axis direction. Therefore, when motion in the z-axis direction is considered, a complete fly-around motion may not be achievable. However, given the complexity of three-dimensional motion and the conservatism in defensive strategy decision-making, we do not delve into this scenario in detail. The conceptual task-intention space for target T is shown in Fig. 5 and is defined as follows:

$$S_{IG} = \{I_i \mid i \in \mathbb{I}_1^7\} \tag{6}$$

where I_i denotes the i^{th} task-intention category in S_{IG}.

Remark 1. *The seven major categories and their associated intention descriptions are introduced as a higher-level taxonomy to facilitate interpretation in the non-cooperative context. In our implementation and experiments, the recognition task is performed at the geometric configuration level: the model predicts one of the predefined* 15 *relative-motion geometric configuration labels* $\{p_i\}$*. The taxonomy is only used to map these configuration-level outputs to higher-level intention hypotheses when needed.*

3. Orbital Motion Intention Recognition Model

3.1. Overall Architecture

To achieve lightweight and efficient orbital motion intention recognition from orbital time series data, this section presents a physics-guided grouped mixer architecture, as shown in Fig. 6. The proposed method consists of three components: a physics-guided grouped feature encoder, a stack of global mixer layers, and a classification head. The resulting flat model is referred to as GM-mixs. Let the input sequence be denoted by:

$$\boldsymbol{\mathcal{X}} = [\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_l]^{\mathrm{T}} \in \mathbb{R}^{l \times o} \tag{7}$$

where l is the sequence length and o is the feature dimension at each time step. In batched form, the input is denoted by $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{b \times l \times o}$, where b denotes the batch size. In GM-mixs, the input at each time step consists only of the relative position variables in the LVLH frame:

$$\boldsymbol{X}_t = [x_t, y_t, z_t]^{\mathrm{T}} \in \mathbb{R}^3 \tag{8}$$

thus $o = 3$. In particular, the in-plane motion components (x_t, y_t) mainly characterize the planar geometric evolution of the relative trajectory, whereas the out-of-plane component z_t reflects the cross-track deviation. The motion configurations to be recognized in this work are primarily determined by the relative motion pattern in the $x - y$ plane and are only weakly related to the variation along the z-direction. Therefore, treating these variables in a fully uniform manner at the input stage may weaken the representation of physically meaningful correlations. We first propose a physics-guided grouped feature encoder to transform the raw input sequence into a structured latent sequence. The resulting latent sequence is then processed by a stack of global mixer layers to model temporal dependencies within the short observation window in a shared hidden space. Finally, a classification head maps the learned representation to the geometric configuration label space.

To improve model robustness and recognition performance, we construct a hierarchical model, termed HiGM-mixs, as shown in Fig. 6. Unlike GM-mixs, HiGM-mixs employs a dual-branch low-level module. Specifically, each local chunk is represented by a raw chunk and a physics-trend chunk. The resulting branch-specific representations are fused into a chunk-level feature, which is then aggregated by a global mixer for final classification. The components of HiGM-mixs are detailed in the following subsections.

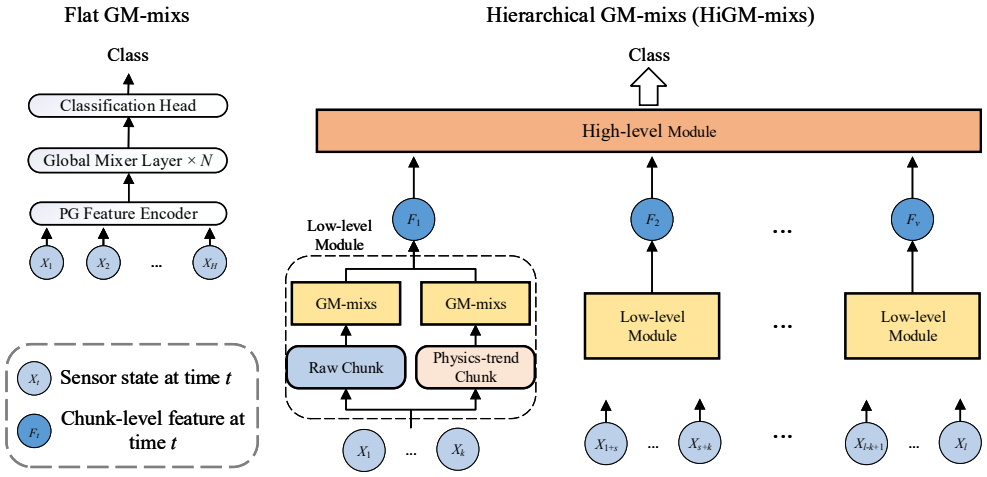

Figure 6: Schematic diagram of GM-mixs and HiGM-mixs.

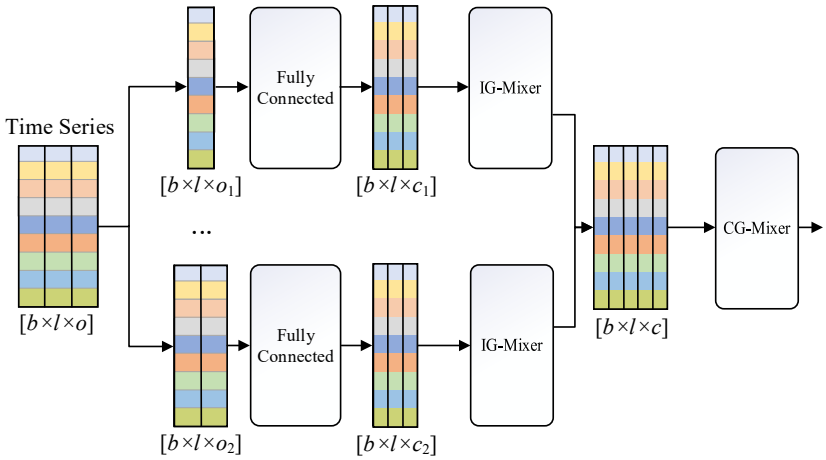

Figure 7: Schematic diagram of the physics-guided grouped feature encoder, where the input feature dimension and hidden feature dimension satisfy $o = o_1 + o_2$ and $c = c_1 + c_2$, respectively.

3.2. Physics-Guided Grouped Feature Encoder

For convenience in the following description, the basic mixing layer adopted in this paper is uniformly defined as:

$$\mathcal{M}(X; f) = X + f\left(\mathrm{LN}(X)\right) \tag{9}$$

where X and $\mathcal{M}(X; f)$ denote the input and output of the layer, $\mathrm{LN}(\cdot)$ denotes layer normalization, and $f(\cdot)$ denotes an MLP transformation. This sublayer consists of three components: layer normalization [41], MLP-based transformation, and skip connections [42]. The MLP uses GELU [43] as the activation function.

In the GM-mixs architecture, we propose a physics-guided grouped feature encoder to organize the input physical variables before temporal modeling, as shown in Fig. 7. Instead of mixing all input variables in a fully uniform manner, the encoder first partitions the input feature vector into several groups with distinct physical associations. It then performs projection, in-group mixing, and cross-group mixing in sequence to generate a unified latent representation.

Taking the relative position input as an example, for each time step $t \in \{1, 2, \ldots, l\}$, the raw input vector is defined as shown in (8). According to the physical associations among different features, the input can be partitioned into the following groups:

$$\boldsymbol{g}_{t,1} = [x_t, y_t]^{\mathrm{T}} \in \mathbb{R}^2, \boldsymbol{g}_{t,2} = [z_t]^{\mathrm{T}} \in \mathbb{R}^1 \tag{10}$$

where $\boldsymbol{g}_{t,1}$ denotes the in-plane motion group, and $\boldsymbol{g}_{t,2}$ denotes the out-of-plane motion group. For the j-th group, a linear projection layer is first applied to map the input features from the group specific input dimension o_j to the hidden dimension c_j. After projection, an in-group mixer is applied to each group:

$$\boldsymbol{u}_{t,j} = \mathcal{M}(\boldsymbol{g}'_{t,j}; f_{\mathrm{IG}}^{(j)}), j = 1, 2 \tag{11}$$

where $f_{\mathrm{IG}}^{(j)}(\cdot)$ denotes the MLP transformation of the in-group mixer for the j-th group, and $\boldsymbol{g}'_{t,j}$ denotes the projected representation of $\boldsymbol{g}_{t,j}$. The in-group mixer is used to capture the correlations within each group. Specifically, for $\boldsymbol{g}_{t,1}$, it captures the coupled geometric relationship between x_t and y_t; for $\boldsymbol{g}_{t,2}$, it extracts a latent representation of the z_t component.

The outputs of all groups are then concatenated as:

$$\boldsymbol{u}_t = [\boldsymbol{u}_{t,1}, \boldsymbol{u}_{t,2}] \in \mathbb{R}^c, c = c_1 + c_2 \tag{12}$$

A cross-group mixer is subsequently applied:

$$\boldsymbol{q}_t = \mathcal{M}(\boldsymbol{u}_t; f_{\mathrm{CG}}) \tag{13}$$

where $f_{\mathrm{CG}}(\cdot)$ denotes the MLP transformation of the cross-group mixer. This step is used to capture the interactions between different groups, thereby producing a unified latent representation at time step t. The obtained sequence is:

$$\boldsymbol{V} = [\boldsymbol{q}_1, \boldsymbol{q}_2, \ldots, \boldsymbol{q}_l]^{\mathrm{T}} \in \mathbb{R}^{b \times l \times c} \tag{14}$$

After the cross-group mixer, the encoder further employs a shallow temporal mixer:

$$\boldsymbol{H} = \mathcal{M}(\boldsymbol{V}; f_{\mathrm{PT}}) \tag{15}$$

where $f_{\mathrm{PT}}(\cdot)$ denotes the MLP transformation used in the shallow temporal mixer. In this encoder, the physically meaningful differences among the raw variables are first captured in the learned representation, and preliminary temporal dependencies are then encoded before the sequence is processed by deeper global mixer layers.

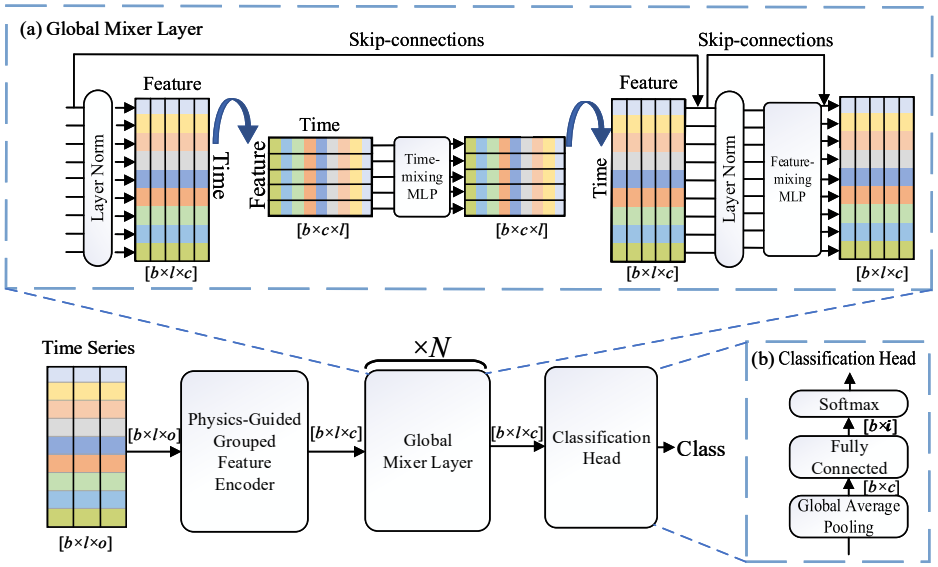

Figure 8: Schematic diagram of the proposed GM-mixs architecture.

3.3. Global Mixer Layer

Multiple global mixer layers are employed to perform deep feature extraction and modeling over both the feature dimension and the temporal dimension of the feature sequence $\boldsymbol{H}$ obtained in the previous step, as shown in Fig. 8(a). For the m-th global mixer layer, the operation can be expressed as:

$$\widetilde{\boldsymbol{H}}^{(m)} = \mathcal{M}(\boldsymbol{H}^{(m-1)}; f_{\mathrm{GT}}^{(m)}) \tag{16}$$

$$\boldsymbol{H}^{(m)} = \mathcal{M}(\widetilde{\boldsymbol{H}}^{(m)}; f_{\mathrm{GF}}^{(m)}) \tag{17}$$

where $f_{\mathrm{GT}}^{(m)}(\cdot)$ denotes the MLP transformation of the temporal mixer in the m-th layer, and $f_{\mathrm{GF}}^{(m)}(\cdot)$ denotes the MLP transformation of the feature mixer in the m-th layer. The final output of the N global mixer layers is given by:

$$\boldsymbol{H}_{\mathrm{out}} = \boldsymbol{H}^{(N)} \in \mathbb{R}^{b\times l\times c} \tag{18}$$

3.4. Classification Head

Finally, a pooling layer followed by a fully connected layer is employed to map the output to the dimensionality of the geometric configuration label

space, as shown in Fig. 8(b). The resulting features are then fed into a multi-class *softmax* function to compute the probabilities corresponding to distinct geometric configuration labels. The *softmax* function is defined as follows:

$$\hat{p}_i = \text{softmax}(\mathcal{H}_i) = \frac{e^{\mathcal{H}_i}}{\sum_{j=1}^{15} e^{\mathcal{H}_j}} \tag{19}$$

$$\hat{s} = \arg\max(P) \tag{20}$$

$$\arg\max(P) = \left\{ i \mid i \in 1, 2, \ldots, 15, \hat{p}_i = \max_{j \in \{1,2,\ldots,15\}} \hat{p}_j \right\} \tag{21}$$

where $\mathcal{H}_i$ denotes the input to the *softmax* function, $\hat{p}_i$ denotes the probability of geometric configuration label i, P denotes the predicted probability distribution over all geometric configuration labels, and $\hat{s}$ denotes the predicted geometric configuration label. The general architecture of GM-mixs is shown in Fig. 8.

Remark 2. *Compared with existing intention recognition models, the proposed physics-guided grouped feature encoder and GM-mixs have the following advantages.*

(1) The input variables are organized according to their physical roles in orbital configuration evolution. The in-plane components (x_t, y_t) are grouped together, while the out-of-plane component z_t is treated separately. This design reduces unnecessary feature mixing at the input stage and provides a more structured representation for orbital motion intention recognition.

(2) Based on this encoding, GM-mixs uses simple MLP-based modules for temporal mixing and cross-group interaction modeling. It avoids recurrent units and self-attention mechanisms. Therefore, it has a simple structure and low computational complexity, while still preserving the main physical relations in the input sequence.

3.5. Hierarchical Architecture

To further enhance local feature extraction capability and robustness, we develop a hierarchical version of GM-mixs, termed HiGM-mixs, by incorporating a hierarchical time series modeling architecture, as shown in Fig. 9. The hierarchical model first partitions the original time series into overlapping chunks. The low-level module then captures local features from these

chunks, and the resulting chunk-level feature sequence is subsequently aggregated at the high-level module. Unlike the flat GM-mixs, the low-level module in the hierarchical model adopts a dual-branch design to jointly exploit raw local trajectory details and smoothed physics-trend information.

Given the original sequence $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{b\times l\times o}$, a sliding-window strategy is adopted to generate v local chunks:

$$\boldsymbol{W}_i \in \mathbb{R}^{b\times k\times o}, i = 1, 2, \ldots, v \tag{22}$$

where k denotes the chunk length, and v denotes the number of extracted chunks. For an input sequence of length l, the number of chunks is given by:

$$v = \left\lfloor \frac{l-k}{s} \right\rfloor + 1 \tag{23}$$

where s denotes the stride between two adjacent chunks. To generate overlapping local chunks, the stride satisfies $0 < s < k$.

Unlike GM-mixs, HiGM-mixs adopts a dual-branch design in its low-level module for each local chunk. Specifically, for the i-th chunk, two branch specific chunk representations are constructed:

$$\boldsymbol{W}_i^{(r)} \in \mathbb{R}^{b\times k\times 3}, \quad \boldsymbol{W}_i^{(p)} \in \mathbb{R}^{b\times k\times d_p} \tag{24}$$

where $\boldsymbol{W}_i^{(r)}$ denotes the raw chunk composed of the original relative position sequence (x, y, z) within the chunk, $\boldsymbol{W}_i^{(p)}$ denotes the physics-trend chunk

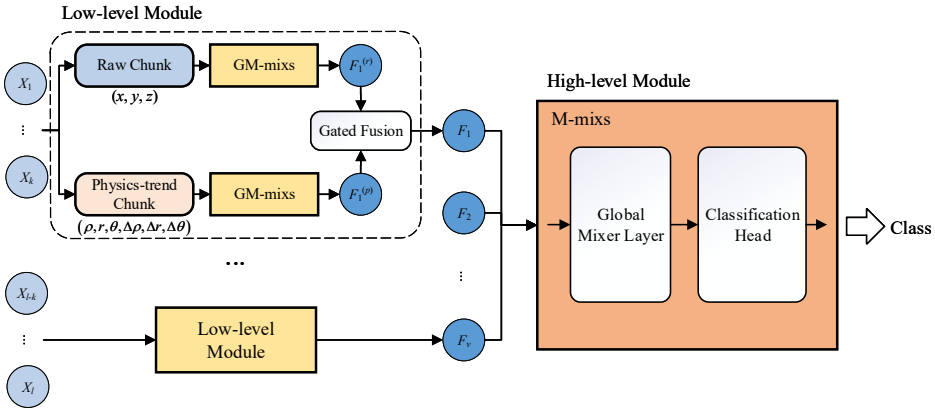

Figure 9: Schematic diagram of the proposed HiGM-mixs architecture.

constructed from physics-based geometric descriptors and their smoothed forms, and d_p denotes the feature dimension of the physics-trend chunk. In the present implementation, $d_p = 6$. The raw chunk is used to preserve local motion details and geometric variations. The physics-trend chunk is used to emphasize stable local geometric trends while suppressing high frequency noise. In this paper, the following physical-trend descriptors are selected:

$$\rho_t = \sqrt{x_t^2 + y_t^2}, \quad r_t = \sqrt{x_t^2 + y_t^2 + z_t^2}, \quad \theta_t = \operatorname{atan2}(y_t, x_t) \tag{25}$$

$$\Delta\rho_t = \rho_t - \rho_{t-1}, \quad \Delta r_t = r_t - r_{t-1}, \quad \Delta\theta_t = \theta_t - \theta_{t-1} \tag{26}$$

where ρ_t denotes the planar projected relative distance, r_t denotes the three dimensional relative distance, θ_t denotes the in-plane azimuth angle, and $\Delta\rho_t$, Δr_t, and $\Delta\theta_t$ denote their corresponding first-order differences. Due to the short length of local chunks, the physical-trend descriptors may be affected by local fluctuations. Therefore, before the above six variables are fed into the physics-trend branch, a simple moving-average smoothing operation is applied along the temporal dimension. Let the physical-trend vector be $\boldsymbol{\phi}_t = [\rho_t, r_t, \theta_t, \Delta\rho_t, \Delta r_t, \Delta\theta_t]$. For the j-th component of $\boldsymbol{\phi}_t$, the smoothed value is computed as:

$$\bar{\phi}_t^{(j)} = \frac{1}{K_s} \sum_{q=-\kappa}^{\kappa} \phi_{\mathrm{clip}(t+q,1,k)}^{(j)}, \quad K_s = 2\kappa + 1 \tag{27}$$

Here, clip$(\cdot)$ denotes replicate padding at the sequence boundaries. In this work, the smoothing window is set to $K_s = 3$, namely $\kappa = 1$. Thus, the smoothed physics-trend chunk is obtained as $\boldsymbol{W}_i^{(p)} \in \mathbb{R}^{b\times k\times 6}$.

These two chunk representations are then encoded by two branch-specific low-level GM-mixs:

$$\boldsymbol{F}_i^{(r)} = \mathcal{F}_r\left(\boldsymbol{W}_i^{(r)}\right), \quad \boldsymbol{F}_i^{(p)} = \mathcal{F}_p\left(\boldsymbol{W}_i^{(p)}\right) \tag{28}$$

where $\mathcal{F}_r(\cdot)$ and $\mathcal{F}_p(\cdot)$ denote the GM-mixs modules in the raw branch and the physics-trend branch, respectively, and $\boldsymbol{F}_i^{(r)}$ and $\boldsymbol{F}_i^{(p)}$ denote the local feature sequences extracted by the corresponding branches, respectively.

In the raw-branch GM-mixs $\mathcal{F}_r(\cdot)$, each chunk is partitioned into an in-plane group containing the (x, y) sequence and an out-of-plane group containing the z sequence. In the physics-trend-branch GM-mixs $\mathcal{F}_p(\cdot)$, each chunk is partitioned into a geometric-state group containing (ρ, r, θ) and a trend-variation group containing $(\Delta\rho, \Delta r, \Delta\theta)$.

To fuse the two branch outputs, $\boldsymbol{F}_i^{(r)}$ and $\boldsymbol{F}_i^{(p)}$, a gated fusion mechanism is employed to generate the chunk-level feature $\boldsymbol{F}_i$. By stacking all chunk-level features, the high-level feature sequence is obtained as $\boldsymbol{F} = [\boldsymbol{F}_1, \boldsymbol{F}_2, \ldots, \boldsymbol{F}_v]$. The high-level feature sequence is then fed into the global mixer. Since the high-level input is a sequence of fused chunk-level features rather than the original physical variables, the global mixer is implemented by M-mixs without the physics-guided grouped feature encoder, and outputs the final class probability distribution, as shown in Fig. 9.

Remark 3. *Compared with GM-mixs, HiGM-mixs introduces a hierarchical architecture to strengthen local feature extraction under short observation windows. In each local chunk, the raw chunk preserves local motion details, while the physics-trend chunk provides more stable local trend information. Their combination helps the model capture complementary information within each local chunk. The resulting chunk-level feature is then aggregated by the high-level module for final classification. Therefore, HiGM-mixs improves representation capability and robustness, while still preserving a lightweight design.*

3.6. Training Details

Before training the model, the dataset must first be subjected to normalization processing. The purpose of normalization is to eliminate dimensional disparities among data points, prevent gradient explosion, optimize gradient descent, and enhance both training efficacy and stability of the model. Among prevalent normalization methods, min-max normalization is commonly employed. However, considering the characteristics of the sample orbital data in this study, such as its wide range, processing with the min-max normalization method may generate outliers that affect model training. Therefore, we adopt the “scale compression normalization" method to preprocess the data, compressing it into the range of [-1, 1]. (x, y, z) is the set of sample relative position data, and the normalization processing formula is as follows:

$$L(m) = \sqrt{x_m^2 + y_m^2 + z_m^2} \tag{29}$$

$$[\bar{x}_m, \bar{y}_m, \bar{z}_m] = \left[\frac{x_m}{L(m)}, \frac{y_m}{L(m)}, \frac{z_m}{L(m)}\right] \tag{30}$$

where $L(m)$ is the magnitude of the relative position vector for $m \in \{t_1, \ldots, t_H\}$, and $[\bar{x}_m, \bar{y}_m, \bar{z}_m]$ represents the normalized relative position data.

Table 1: Orbital parameters of the GEO target spacecraft.

Orbit elements	Value(unit)
Semi-major axis (a)	42164 km
Eccentricity (e)	0.00046
Inclination (i)	0.7°
Right ascension of ascending node (Ω)	30°
Argument of perigee (w)	45°
True anomaly (f)	22°

Based on the above, the model was trained using backpropagation and optimized using the adaptive moment estimation (Adam [44]) algorithm. Since this problem is a multi-class classification task, we choose the cross-entropy function as the loss function. The expression of the cross-entropy function is as follows:

$$\mathcal{L}(\omega) = -\frac{1}{M}\sum_{j=1}^{M}(P^{(j)})^{\top} \cdot \log \hat{P}(\omega)^{(j)} \tag{31}$$

where ω denotes all weight parameters in the intention recognition model, M denotes the number of samples in the batch, P denotes the actual probability distribution of the sample data, and $\hat{P}$ denotes the probability distribution predicted by the model. All our models are trained end-to-end, and the core of training the model is to find a set of parameters ω that minimizes the total loss of all samples.

4. Simulation results

In this section, the performance of the models in the orbital motion intention recognition task will be evaluated through experiments, and the advantages and limitations of these models will be analyzed. Focus on the following questions:

(1) How does GM-mixs compare with existing models?
(2) Can HiGM-mixs offer advantages over flat models?
(3) What is the complexity of the proposed model?
(4) How does the proposed model perform in cross orbit transfer and under noisy observations?

Table 2: Hardware and software specifications.

Hardware and software	Specifications
CPU	Intel Xeon Platinum 8352V
GPU	NVIDIA GeForce RTX 4090
RAM	45 G
Python version	3.12
Deep learning framework	PyTorch 2.2.2
CUDA version	11.8

4.1. Experimental Setting and Dataset Generation

Our experimental framework begins with dataset generation focusing on GEO targets. Following the initial training phase, cross-regime validation is performed by adopting targets from various orbital environments to demonstrate the versatility and reliability of the proposed recognition model. The six elements of the orbit of spacecraft S are shown in Table 1. Assume that the radius of the threatened area centered on the spacecraft S is set to 10 km, and the radius of the rendezvous determination region is set to 0.5 km.

We generated a comprehensive dataset of various relative-motion geometric configuration categories by numerically propagating orbits with a dynamics model that includes J2 and J4 perturbations. Gaussian white noise was injected into the simulated observations to replicate measurement uncertainties. Each simulation sample is generated by forward integration over 10,800 s with a sampling interval of 120 s, resulting a time-series comprising 90 time steps. Then, a sliding window with a length of 20 was applied along the time axis to construct supervised learning samples. Each window segment was treated as one sample and assigned the corresponding geometric configuration label.

This study selects the position information time series as the input, employing a sliding window size of 20. Each time series includes relative position coordinates (x, y, z) at 20 consecutive time steps. In orbital relative-motion dynamics, the relative position sequence is determined by the local relative state through the state-transition relation. Based on this relation, the observability analysis in [26] shows that adjacent position observations can

Source code and data will be released at https://github.com/Yitu-W/HiGM-mixs.git upon acceptance.

constrain the local relative velocity when the corresponding transition submatrix is nonsingular. Therefore, when $l > 2$, the relative position sequence can already provide the basic dynamic information required for configuration recognition. The same study further reports that $l = 10$ provides a practical trade-off between recognition accuracy and observation redundancy. For the proposed HiGM-mixs, a longer input window is also useful for constructing overlapping local chunks. This helps the model better extract local motion trends and aggregate full-window information in the high-level module. Based on these considerations, the input length is set to $l = 20$. This setting provides additional temporal redundancy for noisy observations and slowly evolving configurations, while preserving the short-window recognition setting.

To mitigate the weak cross-orbit generalization caused by different orbital time scales, this paper proposes a phase-consistent sampling strategy for cross-orbit sample construction. The mean motion $n = \sqrt{\mu/a^3}$ determines the temporal evolution rate of the relative motion and is directly related to the orbital semi-major axis. Therefore, due to the different orbital radii of GEO, MEO, and LEO, the same physical sampling interval does not correspond to the same relative-motion phase in different orbital regimes. This inconsistency changes the apparent temporal scale of the relative-motion sequence and further increases the cross-orbit distribution shift.

To reduce this mismatch, GEO is used as the reference orbit. Let n_{G} denote the mean motion of the GEO reference orbit, and let Δt_{G} denote the physical sampling interval used for GEO sample generation. In this study, Δt_{G} is set to 120 s. The reference phase interval $\Delta\tau_{\mathrm{ref}}$ is defined as:

$$\Delta\tau_{\mathrm{ref}} = n_{\mathrm{G}}\Delta t_{\mathrm{G}} \tag{32}$$

For a target orbit (o), let a_o and $n_o = \sqrt{\mu/a_o^3}$ denote its semi-major axis and mean motion, respectively. The sampling interval of this target orbit is then set as:

$$\Delta t_o = \frac{\Delta\tau_{\mathrm{ref}}}{n_o} \tag{33}$$

Based on this strategy, the MEO and LEO samples are sampled with adjusted intervals, so that their input windows cover the same nondimensional orbital phase span as the GEO samples. This strategy only changes the sampling interval used for cross-orbit sample construction. It does not change the model structure or introduce additional trainable parameters.

The hardware platform configuration and experimental environment parameters used in this experiment are shown in Table 2. A total of 16,200,000 sample groups were generated to form the training dataset. Additionally, 4,050,000 sample groups were generated for the validation set to facilitate hyperparameter tuning, and another 4,050,000 sample groups constituted the test set to evaluate the model's identification performance. The three subsets were independently generated using disjoint initial conditions. The batch size was set to 2048 with an initial learning rate of 10^{-3}. A validation-based learning rate schedule was adopted: the learning rate was multiplied by 0.1 when the validation loss did not decrease for 5 consecutive epochs, with a minimum learning rate of 10^{-6}. No fixed number of training epochs was specified; instead, training was terminated using early stopping when the validation loss failed to improve for 10 consecutive epochs, ensuring that each model reached its best validation performance. The test set was used exclusively for final evaluation.

4.2. Model Evaluation Metrics

The intention recognition problem in this paper is a classification task, so this experiment adopts a classic evaluation metric system for classification tasks in the field of machine learning. The specific evaluation metrics include: Accuracy, Precision, Recall, and F1-Score. The calculation formulas are as follows:

$$Acc = \frac{T_p + T_n}{T_p + T_n + F_p + F_n} \tag{34}$$

$$\mathcal{P} = \frac{T_p}{T_p + F_p} \tag{35}$$

$$R = \frac{T_p}{T_p + F_n} \tag{36}$$

$$F_1 = 2 \cdot \frac{\mathcal{P} \times R}{\mathcal{P} + R} \tag{37}$$

where Acc denotes Accuracy, $\mathcal{P}$ denotes Precision, R denotes Recall, F_1 denotes the F1-score, T_p denotes the number of true positive samples, T_n denotes the number of true negative samples, F_p denotes the number of false positive samples, and F_n denotes the number of false negative samples. Accuracy is defined as the proportion of correctly classified samples to the total samples, which directly demonstrates the classification capability of model. Precision is defined as the ratio of true positive predictions to all positive

predictions made by the model. Recall is defined as the proportion of true positives identified by the model relative to all actual positive samples in the dataset. The F1-score is the harmonic mean of precision and recall.

4.3. Experimental Results and Analysis

The experimental dataset was preprocessed, and model training was conducted on a Linux system using the configuration listed in Table 2. In the following experiments, models without hierarchical modeling are referred to as “flat models", while models with hierarchical modeling are referred to as “hierarchical models".

4.3.1. Performance of Flat Models

Prior to model training, the hyperparameters must be tuned to achieve better performance. To this end, we evaluate models with different hyperparameter settings on the validation set and select the configuration that yields the best validation accuracy. First, we vary the feature embedding dimension while keeping all other hyperparameters fixed. Specifically, the feature embedding dimension is set to 32, 64, 128, and 256, and the corresponding validation accuracies are shown in Fig. 10. As the feature embedding dimension increases from 32 to 128, the validation accuracy improves markedly; however, further increasing it to 256 leads to saturation and a slight degradation. This indicates that a higher-dimensional embedding space provides richer representations for subsequent modules, whereas an excessively large embedding dimension increases computational cost and may introduce overfitting. Therefore, to balance accuracy and efficiency, we set the feature embedding dimension to 128.

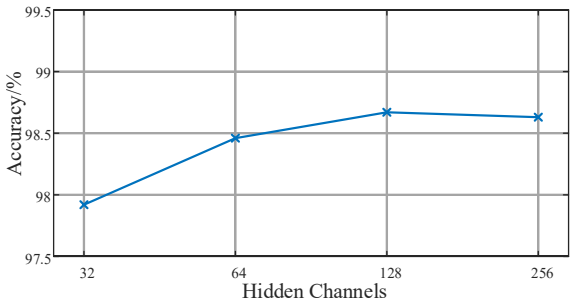

Figure 10: Validation Accuracy of the model for different numbers of hidden channels (feature embedding dimension c).

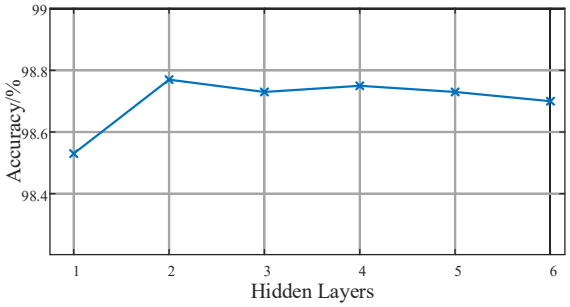

Figure 11: Validation Accuracy of the model for different numbers of hidden layers (N).

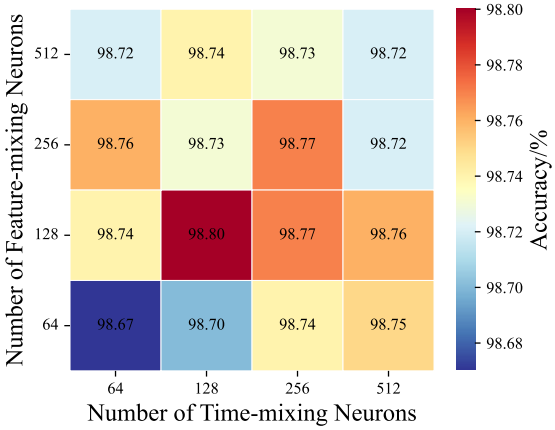

Figure 12: Validation accuracy heatmap for different numbers of time-mixing and feature-mixing neurons.

Based on the above setting, we further tune the number of hidden layers. We set the number of hidden layers to 1, 2, 3, 4, 5, and 6 while keeping the other hyperparameters unchanged, and the validation accuracy curves are reported in Fig. 11. The accuracy consistently increases when the number of layers grows from 1 to 2, suggesting that a deeper network enhances the modeling capability for temporal patterns. When the depth is further increased to 3–6 layers, the accuracy no longer improves and even slightly decreases, possibly due to increased optimization difficulty and degraded generalization. Consequently, we set the number of hidden layers to 2.

With the above hyperparameters fixed, we perform a grid search over the numbers of neurons in the time-mixing MLP and the feature-mixing

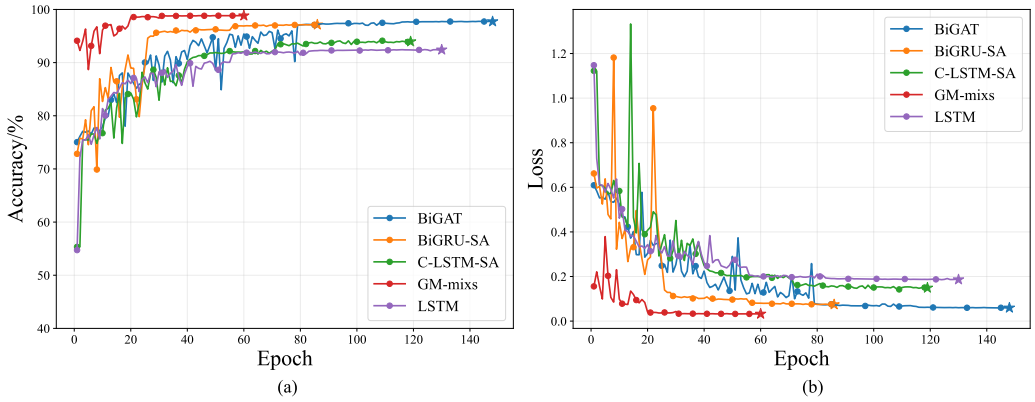

Figure 13: Comparison of training performance between the GM-mixs and other flat models under identical conditions. (a) shows the accuracy curve, while (b) depicts the loss curve.

MLP to examine their effects on performance. The neuron numbers for both MLPs are chosen from 64, 128, 256, 512, and the resulting validation-accuracy heatmap is shown in Fig. 12. When the MLP widths are small, the model capacity is insufficient to capture discriminative patterns, resulting in lower accuracy. Increasing the widths to (128, 128) yields the better performance. Further enlarging the MLPs increases the parameter count but does not provide additional accuracy gains. The highest validation accuracy is achieved at time-mixing = 128 and feature-mixing = 128, which also offers a favorable accuracy–complexity trade-off. Therefore, we adopt (128, 128) as the final neuron configuration.

To validate the advantages of the proposed GM-mixs, we compare it with four representative flat baselines: LSTM, BiGRU-SA [25], C-LSTM-SA [27], and BiGAT [28]. All models are trained on the same dataset with identical optimization settings, and an early-stopping strategy is adopted based on the validation loss to avoid overfitting. For each model, the best test accuracy and the corresponding loss at the early-stopping epoch are taken as its final performance.

Fig. 13(a) and Fig. 13(b) show the evolution of test accuracy and loss with respect to training epochs. It can be observed that GM-mixs converges much faster than the other models: its accuracy rises above the 97% level within only a few epochs, while the loss rapidly drops to a very low value and remains stable thereafter. In contrast, BiGAT, BiGRU-SA, C-LSTM-

SA, and LSTM require more epochs to approach their best performance, and their loss curves settle at noticeably higher levels. With early stopping, GM-mixs achieves the highest test accuracy and the lowest loss among all flat models, with a test accuracy of 98.78% and a loss of 0.0335, clearly outperforming the remaining baselines. These results show that, under the same training configuration, GM-mixs converges faster and achieves better generalization than the compared models.

4.3.2. Validation of Hierarchical Architecture Gains

First, we tune the chunk length, which is a key hyperparameter in the hierarchical model, on the validation set. In this experiment, the stride is fixed to s=3, and only the chunk length k is varied. We consider four chunk length settings (5, 8, 11, and 14); the corresponding training curves and validation set results are reported in Fig. 14. The results show that the model achieves the best recognition performance when $k = 11$, with an accuracy of 99.42%. Analysis suggests that the chunk length $k = 11$ provides a better balance between local feature extraction and global information preservation:

(1) Too small chunks might cause the model to over-focus on extremely short-term local fluctuations, failing to effectively model key patterns over longer ranges and potentially introducing more noise.

(2) Conversely, overly large chunks could lead to excessive aggregation of information within chunks, losing important fine grained temporal dynamic features and reducing the model sensitivity and discriminative ability to local pattern changes.

Therefore, we set the chunk length to $k = 11$ and fix it in all subsequent

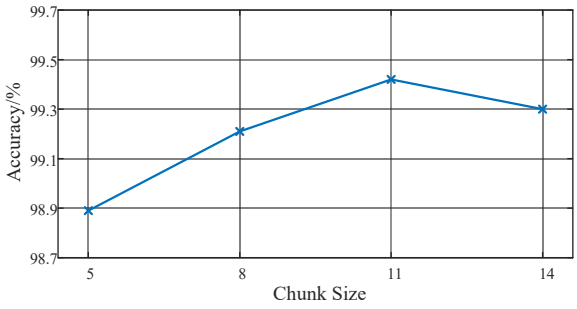

Figure 14: Recognition performance of HiGM-mixs with different chunk lengths under a fixed stride. The chunk length k is set to 5, 8, 11, and 14, and the best performance is obtained at $k = 11$.

Table 3: Test set performance of flat models and hierarchical variants formed by different combinations of low-level and high-level models. Each result is reported as the mean and standard deviation over 10 repeated runs with different random seeds. The compared BiGRU-SA, C-LSTM-SA, and BiGAT baselines follow [25], [27], and [28], respectively.

Model type	**Model Architecture**		**Accuracy/%**	**Precision/%**	**Recall/%**	**F1-Score/%**
Flat	LSTM		92.517±0.047	92.546±0.045	92.517±0.047	92.501±0.047
	C-LSTM-SA		93.996±0.040	94.029±0.039	93.996±0.040	93.998±0.040
	BiGRU-SA		97.137±0.029	97.146±0.028	97.137±0.029	97.138±0.029
	BiGAT		98.246±0.022	98.249±0.022	98.246±0.022	98.246±0.022
	GM-mixs		98.852±0.022	98.854±0.022	98.852±0.022	98.852±0.022
Hierarchical	Low-level	High-level				
	LSTM	LSTM	97.904±0.024	97.909±0.024	97.904±0.024	97.905±0.024
		C-LSTM-SA	98.253±0.021	98.258±0.021	98.253±0.021	98.254±0.021
		BiGRU-SA	98.204±0.023	98.208±0.023	98.204±0.023	98.205±0.023
		BiGAT	98.372±0.023	98.377±0.023	98.372±0.023	98.373±0.023
		M-mixs	98.752±0.019	98.757±0.019	98.752±0.019	98.753±0.019
	C-LSTM-SA	LSTM	98.408±0.029	98.415±0.027	98.408±0.029	98.409±0.028
		C-LSTM-SA	98.104±0.028	98.109±0.028	98.104±0.028	98.104±0.028
		BiGRU-SA	98.295±0.024	98.304±0.023	98.295±0.024	98.296±0.023
		BiGAT	98.862±0.020	98.869±0.019	98.862±0.020	98.863±0.020
		M-mixs	99.099±0.012	99.102±0.011	99.099±0.012	99.099±0.012
	BiGRU-SA	LSTM	98.873±0.017	98.880±0.016	98.873±0.017	98.874±0.017
		C-LSTM-SA	98.627±0.020	98.636±0.019	98.627±0.020	98.629±0.019
		BiGRU-SA	99.038±0.015	99.045±0.015	99.038±0.015	99.039±0.015
		BiGAT	98.918±0.016	98.926±0.015	98.918±0.016	98.919±0.016
		M-mixs	99.273±0.011	99.277±0.011	99.273±0.011	99.274±0.011
	BiGAT	LSTM	99.160±0.013	99.162±0.013	99.160±0.013	99.160±0.013
		C-LSTM-SA	99.106±0.014	99.108±0.014	99.106±0.014	99.107±0.014
		BiGRU-SA	99.114±0.015	99.115±0.015	99.114±0.015	99.114±0.015
		BiGAT	99.208±0.015	99.212±0.015	99.208±0.015	99.208±0.015
		M-mixs	99.165±0.016	99.165±0.016	99.165±0.016	99.165±0.016
	GM-mixs	LSTM	99.297±0.010	99.298±0.010	99.297±0.010	99.297±0.010
		C-LSTM-SA	99.235±0.008	99.237±0.008	99.235±0.008	99.235±0.008
		BiGRU-SA	99.290±0.012	99.291±0.012	99.290±0.012	99.290±0.012
		BiGAT	99.217±0.013	99.219±0.013	99.217±0.013	99.217±0.013
		M-mixs	99.494±0.007	99.494±0.007	99.494±0.007	99.494±0.007
	HiGM-mixs		+3.344 accuracy gain			

experiments. The stride is fixed to $s = 3$. Since the input sequence length is $l = 20$, the resulting number of chunks is $v = 4$. Under this configuration, we then evaluate the effectiveness of the hierarchical modeling architecture by comparing it with several flat models.

By replacing the low-level module and high-level module model in the HiGM-mixs with other flat models, various hierarchical variants can be obtained. All hierarchical variants and flat models are trained and evaluated under the same experimental setup. The results are summarized in Table 3, where each metric is reported by its mean and standard deviation over 10 repeated runs with different random seeds. According to the test results, the

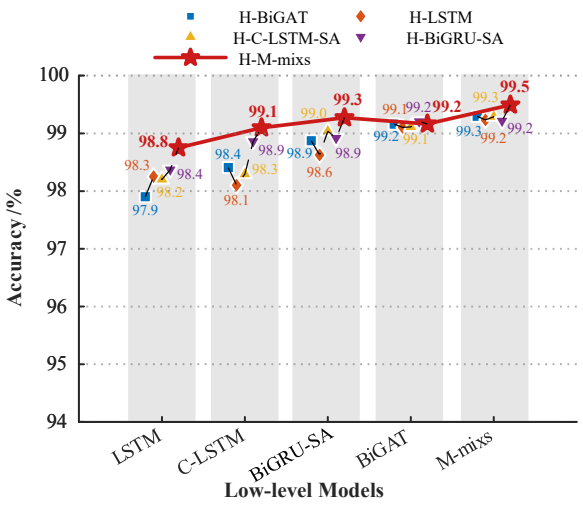

Figure 15: Recognition accuracy comparison of models with different hierarchical levels. HiGM-mixs achieves the best performance among the hierarchical models.

HiGM-mixs model outperforms the flat model, with the recognition accuracy in the task being further improved by 3.344 percentage points. Meanwhile, among hierarchical models, the HiGM-mixs model also outperformed other hierarchical models, achieving an approximately 0.669 percentage points average improvement in F1-Score.

As shown in Fig. 15, most hierarchical models achieve higher recognition accuracy than their corresponding flat models. This result shows that the hierarchical time series modeling architecture can improve feature extraction for orbital configuration recognition. For example, HiLSTM improves the accuracy of LSTM by 5.387%. These results indicate that hierarchical modeling is generally effective when the low-level model and the high-level model are properly matched.

However, the performance gain of hierarchical modeling is not uniform across different module combinations. For example, when BiGAT is used as the low-level model and LSTM is used as the high-level model, the accuracy reaches 99.160%, which is 0.914 percentage points higher than that of the flat BiGAT model. This result indicates that hierarchical aggregation can also improve strong attention-based flat models. Nevertheless, this variant is still lower than the proposed HiGM-mixs, whose accuracy reaches 99.494%. In addition, different high-level modules lead to different performance gains under the same low-level module. For example, with C-LSTM-SA as the low-level

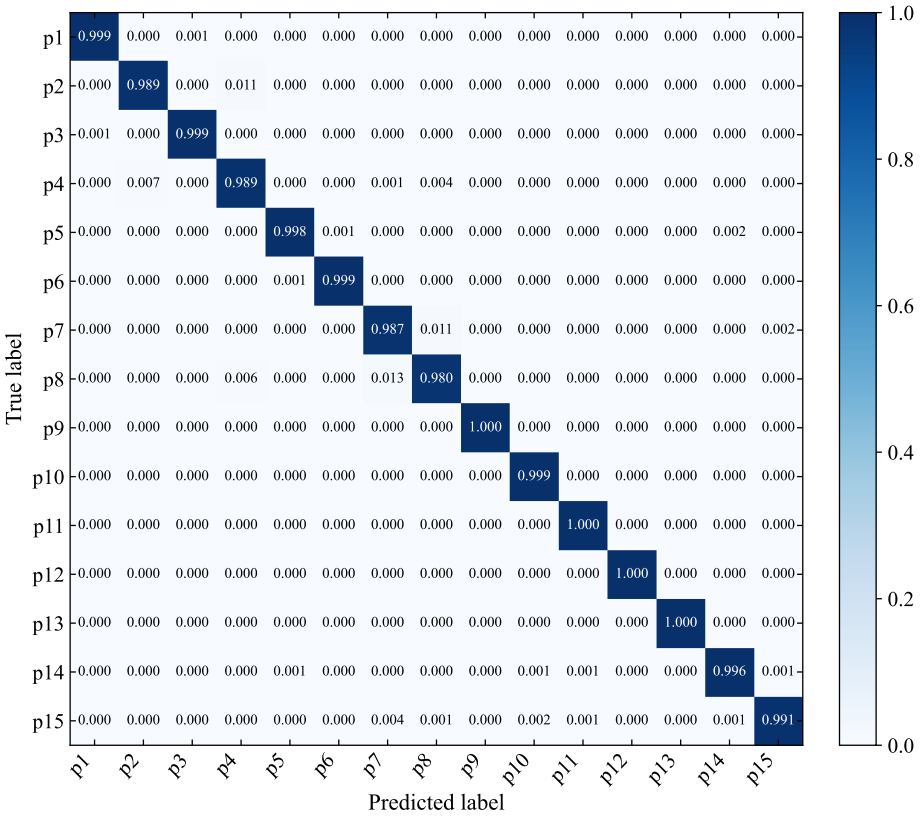

Figure 16: Normalized confusion matrix of HiGM-mixs for the 15 geometric configuration classes.

model, the BiGAT high-level module achieves 98.862% accuracy, whereas the C-LSTM-SA high-level module achieves 98.104% accuracy. These results suggest that the effectiveness of hierarchical modeling depends on the compatibility between the low-level feature extractor and the high-level aggregation module. The proposed HiGM-mixs achieves the best performance because the GM-mixs low-level module and the M-mixs high-level module are well matched for local-to-global temporal aggregation. This further supports the effectiveness of the proposed HiGM-mixs architecture.

Although HiGM-mixs achieves the highest mean accuracy in Table 3, the difference from the strongest hierarchical variants is relatively small. Therefore, we further conduct statistical significance tests under the same random seeds and test settings. The paired comparison is performed between HiGM-

Table 4: Complexity of the models.

Module	Module Size	Parameter Count	Computational Complexity	Accuracy
LSTM	7.06 MB	1,849,870	73.53 MFLOPs	92.517%
C-LSTM-SA	8.47 MB	2,217,615	45.04 MFLOPs	93.996%
BiGRU-SA	16.79 MB	4,197,711	167.56 MFLOPs	97.137%
BiGAT	22.43 MB	5,059,055	226.03 MFLOPs	98.246%
S-CNN	0.78 MB	202,639	6.78 MFLOPs	86.352%
HF-MLP	1.06 MB	277,519	10.93 MFLOPs	97.461%
GM-mixs	0.54 MB	126,365	8.53 MFLOPs	98.852%
HiGM-mixs	1.82 MB	466,042	40.36 MFLOPs	99.494%

mixs and the two strongest hierarchical variants, namely GM-mixs+LSTM and GM-mixs+BiGRU-SA. Compared with GM-mixs+LSTM, HiGM-mixs improves the mean accuracy by 0.1962 percentage points, with a 95% confidence interval of [0.1907, 0.2016]. The paired (t)-test gives $p = 3.31 \times 10^{-14}$, and the Wilcoxon signed-rank test gives (p=0.00195). Compared with GM-mixs+BiGRU-SA, HiGM-mixs improves the mean accuracy by 0.2040 percentage points, with a 95% confidence interval of [0.1997, 0.2082]. The paired (t)-test gives $p = 2.48 \times 10^{-15}$, and the Wilcoxon signed-rank test gives (p=0.00195). These results confirm that the improvement of HiGM-mixs is statistically significant.

To further evaluate the discrimination ability of the proposed model on similar classes, Fig. 16 shows the normalized confusion matrix for the 15 geometric configuration classes. Most classes are recognized with high accuracy. The main confusions occur between geometrically similar classes, especially p_7 and p_8, and p_2 and p_4. These classes share similar local trajectory patterns under short observation windows, which makes them more difficult to distinguish. In contrast, p_9, p_{12}, and p_{13} are recognized almost perfectly.

4.3.3. Model Complexity Analysis

Using a fixed input size (a sequence of length 20 with 3 features), we quantify model complexity in terms of model size, the number of trainable parameters, and FLOPs per forward pass. The results are summarized in Table 4. To provide a fairer lightweight comparison, we add two lightweight baselines in the revised manuscript: a shallow 1D convolutional neural network (S-CNN) and a handcrafted feature based MLP (HF-MLP).

Analysis results show that the flat model GM-mixs exhibits remarkable lightweight characteristics. It has only 0.126 M parameters, a model size of only 0.54 MB, and an inference complexity of 8.53 MFLOPs. Compared with

the recurrent and attention-based baseline models, GM-mixs uses only about 6.83% of the parameters of the smallest baseline model, while achieving an average accuracy improvement of about 3.38 percentage points. Compared with the newly added lightweight baselines, GM-mixs also remains compact. It uses fewer parameters than both S-CNN and HF-MLP, while achieving a higher accuracy of 98.852%.

The proposed HiGM-mixs model employing the hierarchical modeling architecture has 0.466 M parameters, a model size of 1.82 MB, and an inference complexity of 40.36 MFLOPs. Compared with the GM-mixs (proposed herein), HiGM-mixs exhibits elevated model complexity, including increased parameter size, larger model size, and higher computational complexity. This complexity arises from the enhanced modeling capability enabled by its hierarchical structure. Compared with the recurrent and attention-based baselines, HiGM-mixs uses only about 25.19% of the parameters of the smallest recurrent and attention-based baseline model, while achieving an average accuracy improvement of about 5.21 percentage points. Although HiGM-mixs is larger than the two newly added lightweight baselines, it achieves higher accuracy. It improves the accuracy by 13.14 percentage points over S-CNN and by 2.03 percentage points over HF-MLP.

Overall, the proposed models achieve a favorable trade-off between recognition accuracy and model complexity. GM-mixs is the most compact proposed model and still achieves high recognition accuracy. HiGM-mixs further improves the recognition accuracy by using the hierarchical dual-branch design. Although HiGM-mixs has higher complexity than the two newly added lightweight baselines, it achieves better recognition accuracy. Meanwhile, its model complexity remains much lower than that of the recurrent and attention-based baselines. These results indicate that HiGM-mixs is an effective lightweight model for short-window orbital configuration recognition

Table 5: Orbital parameters of the MEO target spacecraft.

Orbit elements	Value(unit)
Semi-major axis (a)	26560.4 km
Eccentricity (e)	0.009659
Inclination (i)	55.9°
Right ascension of ascending node (Ω)	110.7°
Argument of perigee(w)	55.5°
True anomaly (f)	100.0°

Table 6: Orbital parameters of the LEO target spacecraft.

Orbit elements	Value(unit)
Semi-major axis (a)	7218.24 km
Eccentricity (e)	0.001563
Inclination (i)	98.5°
Right ascension of ascending node (Ω)	228.6°
Argument of perigee(w)	127.4°
True anomaly (f)	45°

of non-cooperative space targets.

4.3.4. Evaluation of Noise Robustness and Cross-Orbit Transfer

To evaluate the model's noise tolerance and cross-orbit generalization, we conduct additional tests on two representative non-cooperative targets in different orbital regimes, namely one MEO satellite and one LEO satellite, in addition to the original GEO scenario. The MEO and LEO targets are selected from real on-orbit spacecraft, and their classical Keplerian orbital elements at a reference epoch are obtained from the public CelesTrak database [45]. The corresponding nominal orbital elements of the three targets (GEO, MEO, and LEO) are summarized in Table 1, Table 5, and Table 6. To separate the effect of measurement noise from the effect of cross-orbit distribution shift, the noise robustness experiment is conducted only in the GEO

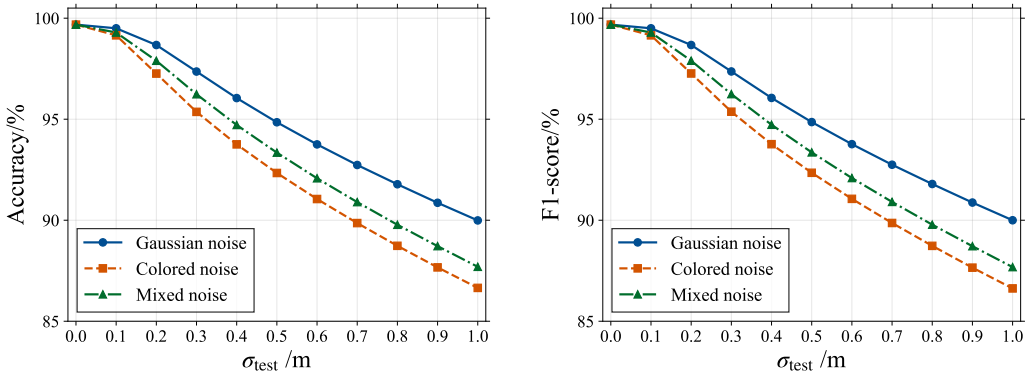

Figure 17: Test-set accuracy (left) and macro F1-score (right) of the original HiGM-mixs under different measurement-disturbance levels σ (m), including Gaussian noise, colored noise, and mixed noise, for the GEO orbit.

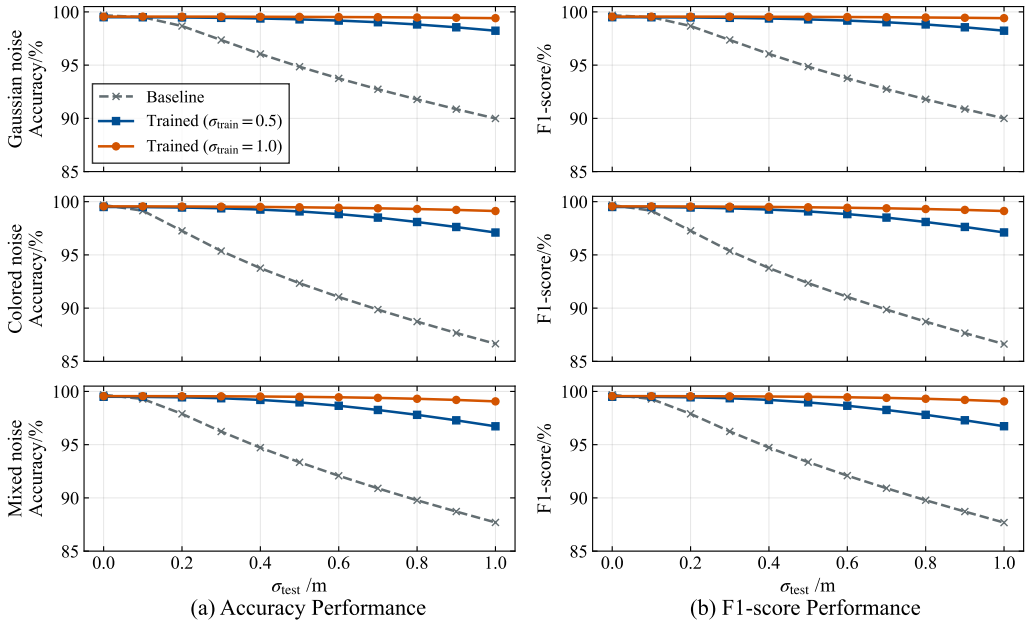

Figure 18: Test-set accuracy (left) and macro F1-score (right) of HiGM-mixs under different measurement-disturbance levels σ (m), including the training noise levels of $\sigma_{\text{train}} = 0.5$ m and $\sigma_{\text{train}} = 1.0$ m, for the GEO orbit.

case. Three types of measurement noise are considered, including zero-mean additive Gaussian noise, colored noise, and mixed noise with sparse outliers. For all three noise types, the nominal noise standard deviation σ_{test} is swept from 0 m to 1 m in steps of 0.1 m. For colored noise, an AR(1) process is used, and the temporal correlation coefficient is set to 0.7. For mixed noise, sparse outliers are randomly added to the Gaussian-noise measurements with a probability of 5%, and the outlier amplitude is set to five times σ_{test}.

The test results of the original HiGM-mixs model under different noise levels are shown in Fig. 17. When $\sigma_{\text{test}} = 0$ m, the model achieves 99.6816% accuracy and 99.682% macro F1-score. As σ_{test} increases, the performance decreases smoothly under all three noise types. At $\sigma_{\text{test}} = 1.0$ m, the accuracy under Gaussian noise, colored noise, and mixed noise decreases to 89.991%, 86.651%, and 87.695%, respectively. The corresponding macro F1-scores are 89.999%, 86.625%, and 87.680%, respectively. Compared with Gaussian noise, colored noise and mixed noise lead to stronger performance degradation. Nevertheless, the performance curves remain stable, which in-

Table 7: GEO-to-MEO and GEO-to-LEO transfer results of HiGM-mixs under zero-shot, lightweight fine-tuning and phase-consistent sampling settings.

Setting	Accuracy/%	F1-Score/%
GEO test	99.494	99.494
GEO→MEO zero shot	36.060	29.685
GEO→LEO zero shot	21.972	17.665
GEO→MEO fine-tuning	87.662	87.351
GEO→LEO fine-tuning	83.985	83.516
GEO→MEO phase-consistent sampling	99.269	99.268
GEO→LEO phase-consistent sampling	98.993	98.952

dicates that HiGM-mixs has a certain robustness to different measurement noise types in the GEO case.

To further enhance robustness in the high-noise regime, we additionally retrain the model with increased training noise levels $\sigma_{\text{train}} = 0.5$ m and $\sigma_{\text{train}} = 1.0$ m, while keeping the same test protocol with $\sigma_{\text{test}} \in [0, 1.0]$ m. The corresponding results are shown in Fig. 18. Compared with the baseline model in Fig. 17, the retrained models show flatter degradation curves. When $\sigma_{\text{train}} = 0.5$ m, the accuracy at $\sigma_{\text{test}} = 1.0$ m reaches 98.233%, 97.101%, and 96.725% under Gaussian noise, colored noise, and mixed noise, respectively. When $\sigma_{\text{train}} = 1.0$ m, the accuracy at $\sigma_{\text{test}} = 1.0$ m further increases to 99.412%, 99.121%, and 99.065% under the three noise types. These results show that noise-aware retraining significantly improves the robustness of HiGM-mixs under strong measurement noise. The clean-condition accuracy only changes slightly, from 99.682% for the original model to 99.500% and 99.557% for the two retrained models. This indicates that the retraining strategy improves high-noise robustness while maintaining high performance under low-noise conditions.

To distinguish cross orbit adaptability from transfer generalization, we further evaluate the final HiGM-mixs model under GEO-to-MEO and GEO-to-LEO transfer settings. The results are summarized in Table 7. The zero-shot results show that direct cross-orbit transfer is difficult, and the recognition accuracy drops to 36.060% for GEO-to-MEO transfer and 21.972% for GEO-to-LEO transfer. In the lightweight fine-tuning setting, the low level parameters are fixed, and only the high-level classification module is updated. The learning rate is set to 1×10^{-4}. Only 20% of the target orbit training data is used. The model is fine-tuned with early stopping based on

the validation loss. After lightweight fine-tuning, the recognition accuracy increases to over 80%. These results show that strict zero-shot generalization across orbital regimes remains difficult.

To address this issue, this paper proposes a lightweight phase-consistent sampling strategy. In this setting, the MEO and LEO samples are constructed using the strategy described in Section 4.1. Their input windows cover the same nondimensional orbital phase span as the GEO samples. This setting does not update the model parameters or change the model structure. With the proposed phase consistent sampling strategy, the accuracy reaches 99.269% for GEO-to-MEO and 98.993% for GEO-to-LEO. Compared with zero-shot transfer, the accuracy increases by 63.209 percentage points for GEO-to-MEO and 77.021 percentage points for GEO-to-LEO. These results show that the weak zero-shot performance is largely caused by the orbital time scale mismatch in the input sequences. After this mismatch is reduced, HiGM-mixs shows strong cross-orbit generalization without additional training.

4.3.5. Recognition in a Continuous Multi-Maneuver Scenario

To further evaluate the applicability of HiGM-mixs under maneuvering-target conditions, a continuous relative-motion trajectory containing two impulsive maneuvers is constructed. In this scenario, the T initially approaches the observing spacecraft S under orbital motion intention p_1. The target subsequently performs two impulsive maneuvers, $\mathbf{M}_1 = [0.706,\ 0.509,\ -0.019]$ m/s, and $\mathbf{M}_2 = [-0.990,\ 0.245,\ -0.049]$ m/s, resulting in the orbital motion intention sequence $p_1 \rightarrow p_6 \rightarrow p_{11}$. The complete trajectory is propagated using the J_4 orbital model. The initial orbital conditions, sampling interval of 120 s, measurement-noise setting, and data preprocessing procedure are kept consistent with those used in the original experiments.

The actual relative trajectory is shown in Fig. 19(a). The solid curves represent the three orbital-motion stages corresponding to p_1, p_6, and p_{11}. The dashed curves denote the corresponding no-maneuver continuations from the selected states and illustrate the trajectory changes produced by the two impulsive maneuvers. The continuously acquired relative-position sequence is processed using a 20-sample rolling observation window with a sliding step of one sample. Each rolling window is fed directly into the pretrained HiGM-mixs model without retraining or parameter updating. The resulting configuration recognition sequence is shown in Fig. 19(b).

Among the 660 rolling observation windows, 631 are correctly recognized,

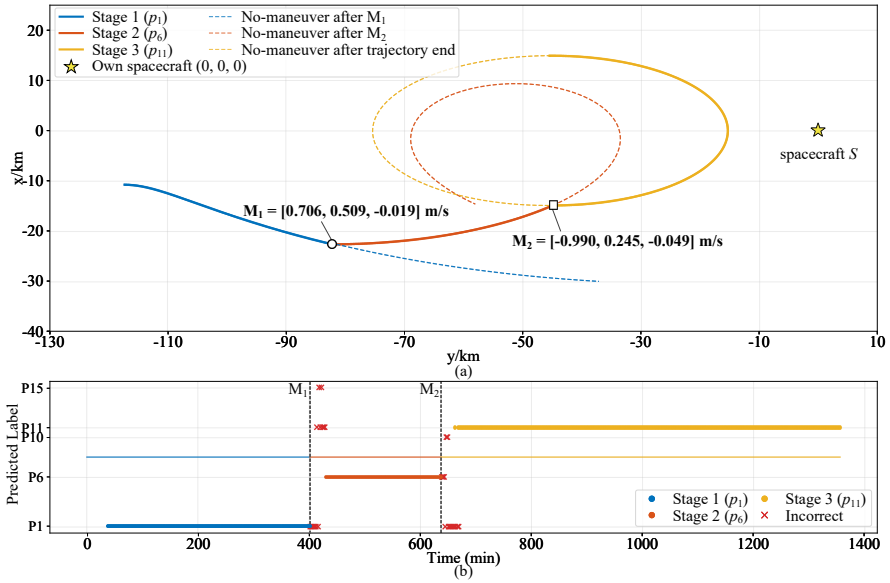

Figure 19: Validation of orbital motion intention recognition under a continuous multi-maneuver scenario. (a) Relative trajectory evolution with two impulsive maneuvers; (b) rolling-window recognition results.

yielding an overall accuracy of 95.61%. The incorrect predictions mainly occur near the maneuver instants because the corresponding windows contain both pre-maneuver and post-maneuver observations. As the rolling windows are updated with post-maneuver observations, the recognition results stabilize at p_6 and p_{11}, respectively. These results demonstrate that the proposed method can effectively identify maneuver-induced orbital motion intention transitions in continuous multi-maneuver scenarios.

5. Conclusion

This paper studies orbital motion intention recognition of non-cooperative space targets based on short-window observation time series. To address the problems of complex model structures and high computational cost in existing methods, this paper proposes a lightweight grouped mixer model, termed GM-mixs. The proposed model improves the balance between recognition performance and model efficiency. To further improve local feature extraction and robustness, a hierarchical extension, termed HiGM-mixs, is developed by introducing a hierarchical dual-branch chunk modeling archi-

tecture. It preserves a simple structure and remains suitable for resource constrained onboard scenarios. The proposed GM-mixs and HiGM-mixs can serve as efficient and lightweight models for orbital motion intention recognition of non-cooperative space targets. The experimental results show that HiGM-mixs achieves an accuracy of 99.494% on the test set. Compared with the recurrent and attention-based baseline models, HiGM-mixs uses only about 25.19% of the parameters of the smallest recurrent and attention-based baseline model, while achieving higher recognition accuracy. The cross-orbit experiments show that the proposed phase-consistent sampling strategy improves GEO-to-MEO and GEO-to-LEO generalization without additional training. Additional experiments demonstrate that HiGM-mixs can track maneuver-induced orbital motion intention transitions and retains a certain degree of generalization under nonzero-eccentricity conditions. Future work will investigate more general eccentric-orbit, strong-interference, and multi-target scenarios.

Acknowledgements

This work is supported by National Key Laboratory of Space Intelligent Control Foundation under Grant No. HTKJ2025KL502014, and Shanghai Academy of Spaceflight Technology (SAST) Foundation under Grant No. SAST2025-012.

Declaration of generative AI and AI-assisted technologies in the manuscript preparation process

During the preparation of this work, the authors used ChatGPT, only to assist with language polishing and grammar refinement. The authors reviewed and edited the output as needed and take full responsibility for the content of the published article.

Appendix A. Configuration Evolution and Recognition under Eccentric Reference-Orbit Conditions

To evaluate the applicability of HiGM-mixs beyond near-circular reference-orbit conditions, the TH/YA analytical solution [46, 47] is used to analyze eccentricity-induced configuration evolution. To separate the influence of orbital eccentricity from that of the initial orbital phase, the feature points

Table A.8: Configuration evolution and model predictions for different orbital motion intentions on eccentric reference orbits. For each intention, 2,000 trajectories are propagated for 36 h from their initial states and segmented into observation windows of 20 time steps.

Label	Nominal CW configuration	Possible configurations	Configuration-evolution trend	Top-3 ($> 1\%$) predictions
p_1	Forward Curvilinear Drift	p_1, p_3, p_6	**Radial:** Increasing excursion with possible y-axis crossing. **Along-track:** Accumulating positive-y drift with possible drip-shaped deformation.	$\boldsymbol{e = 0.02}$: p_1 92.38%, p_3 6.22%. $\boldsymbol{e = 0.10}$: p_1 62.21%, p_6 25.66%, p_3 8.35%.
p_2	Backward Curvilinear Drift	p_2, p_4, p_8	**Radial:** Increasing excursion with possible y-axis crossing. **Along-track:** Accumulating negative-y drift with possible drip-shaped deformation.	$\boldsymbol{e = 0.02}$: p_2 99.48%. $\boldsymbol{e = 0.10}$: p_2 60.66%, p_8 17.06%, p_6 12.08%.
p_3	Drip-Shaped Forward Fly-by	p_1, p_3, p_6	**Radial:** Increasing excursion with possible y-axis crossing. **Along-track:** Accumulating positive-y drift with enhanced drip-shaped motion.	$\boldsymbol{e = 0.02}$: p_3 61.81%, p_1 24.50%, p_6 11.58%. $\boldsymbol{e = 0.10}$: p_6 36.12%, p_3 30.57%, p_1 26.89%.
p_4	Drip-Shaped Backward Fly-by	p_2, p_4, p_8	**Radial:** Increasing excursion with possible y-axis crossing. **Along-track:** Accumulating negative-y drift with enhanced drip-shaped motion.	$\boldsymbol{e = 0.02}$: p_2 89.44%, p_4 5.24%, p_1 5.01%. $\boldsymbol{e = 0.10}$: p_2 63.06%, p_8 16.43%, p_6 9.30%.
p_5	Drip-Shaped Forward Fly-Around	p_1, p_5, p_6	**Radial:** Repeated y-axis crossings with increasing excursion. **Along-track:** Drip-shaped oscillation with accumulating positive-y drift.	$\boldsymbol{e = 0.02}$: p_5 49.39%, p_6 32.38%, p_{14} 7.96%. $\boldsymbol{e = 0.10}$: p_6 30.87%, p_5 21.48%, p_1 17.74%.
p_6	Front/Rear Drip-Shaped Forward Leap	p_1, p_5, p_6	**Radial:** Repeated y-axis crossings with increasing excursion. **Along-track:** Drip-shaped oscillation with accumulating positive-y drift.	$\boldsymbol{e = 0.02}$: p_6 81.81%, p_{14} 6.93%, p_1 4.89%. $\boldsymbol{e = 0.10}$: p_6 52.62%, p_1 15.67%, p_{14} 13.76%.
p_7	Drip-Shaped Backward Fly-Around	p_2, p_7, p_8	**Radial:** Repeated y-axis crossings with increasing excursion. **Along-track:** Drip-shaped oscillation with accumulating negative-y drift.	$\boldsymbol{e = 0.02}$: p_8 47.47%, p_4 18.82%, p_7 15.73%. $\boldsymbol{e = 0.10}$: p_2 67.51%, p_8 15.09%, p_6 6.61%.
p_8	Front/Rear Drip-Shaped Backward Leap	p_2, p_7, p_8	**Radial:** Repeated y-axis crossings with increasing excursion. **Along-track:** Drip-shaped oscillation with accumulating negative-y drift.	$\boldsymbol{e = 0.02}$: p_8 60.19%, p_4 17.38%, p_2 14.20%. $\boldsymbol{e = 0.10}$: p_2 66.35%, p_8 16.25%, p_6 6.29%.
p_9	Elliptical Fly-Around	p_1, p_6, p_9, p_{14}	**Radial:** Increasing excursion with possible delayed y-axis crossing. **Along-track:** Accumulating positive-y drift with loss of trajectory closure.	$\boldsymbol{e = 0.02}$: p_{14} 88.94%, p_9 6.50%, p_{10} 3.76%. $\boldsymbol{e = 0.10}$: p_1 64.21%, p_6 17.89%, p_{14} 8.39%.
p_{10}	Front Elliptical Hover	p_8, p_2, p_{10}, p_{15}	**Radial:** Increasing excursion with possible delayed y-axis crossing. **Along-track:** Accumulating negative-y drift with loss of trajectory closure.	$\boldsymbol{e = 0.02}$: p_{15} 82.32%, p_{10} 16.08%, p_9 1.58%. $\boldsymbol{e = 0.10}$: p_2 39.51%, p_4 23.85%, p_8 18.80%.
p_{11}	Rear Elliptical Hover	p_8, p_2, p_{11}, p_{15}	**Radial:** Increasing excursion with possible delayed y-axis crossing. **Along-track:** Accumulating negative-y drift with loss of trajectory closure.	$\boldsymbol{e = 0.02}$: p_{15} 70.75%, p_8 14.34%, p_{11} 11.23%. $\boldsymbol{e = 0.10}$: p_2 61.32%, p_8 18.52%, p_{15} 6.20%.
p_{12}	Front Fixed-point Hover	p_2, p_{12}	**Radial:** No radial motion, $x(f) = 0$. **Along-track:** Bounded periodic motion on the positive-y side.	$\boldsymbol{e = 0.02}$: p_{12} 88.70%, p_3 9.40%, p_{15} 1.82%. $\boldsymbol{e = 0.10}$: p_{12} 79.72%, p_3 13.97%, p_4 3.12%.
p_{13}	Rear Fixed-point Hover	p_1, p_{13}	**Radial:** No radial motion, $x(f) = 0$. **Along-track:** Bounded periodic motion on the negative-y side.	$\boldsymbol{e = 0.02}$: p_{13} 63.88%, p_1 36.03%. $\boldsymbol{e = 0.10}$: p_{13} 56.16%, p_1 42.01%, p_2 1.26%.
p_{14}	Spiral-Shaped Forward Fly-Around	p_9, p_{14}, p_{15}	**Radial:** Increasing excursion with possible y-axis crossing. **Along-track:** Non-closed motion with possible drift-direction changes.	$\boldsymbol{e = 0.02}$: p_{14} 70.16%, p_{15} 17.25%, p_9 5.17%. $\boldsymbol{e = 0.10}$: p_8 28.64%, p_4 24.82%, p_2 14.16%.
p_{15}	Spiral-Shaped Backward Fly-Around	p_2, p_4, p_7, p_8, p_{15}	**Radial:** Increasing excursion with possible y-axis crossing. **Along-track:** Non-closed motion with accumulating negative-y drift.	$\boldsymbol{e = 0.02}$: p_{15} 60.17%, p_8 22.01%, p_7 16.67%. $\boldsymbol{e = 0.10}$: p_2 57.79%, p_8 17.28%, p_4 8.90%.

Note: Higher e generally leads to faster and stronger configuration deformation; for p_{12} and p_{13}, it mainly increases the along-track variation range.

are placed at the periapsis of the reference orbit, i.e., $f_p = 0$, and the corresponding constant coefficients of the TH solution are determined from the same initial relative states. The resulting evolution trends are numerically validated using the J_4 orbital model. The complete results are summarized in Table A.8.

The results show that orbital eccentricity generally increases the radial excursion and introduces nonuniform along-track drift. Consequently, several drift, fly-by, fly-around, and leap configurations gradually evolve toward geometrically related classes, while the elliptical configurations may lose trajectory closure. In contrast, the two fixed-point configurations remain bounded because their ideal TH solutions contain no $(K(f))$-dependent secular term. Although increasing eccentricity makes the boundaries between similar configurations less distinct, most predictions within the tested range remain consistent with the evolved geometric configurations. These results indicate that HiGM-mixs retains a certain degree of configuration recognition and generalization capability under nonzero-eccentricity conditions.

References

[1] J.-C. Liou, N. L. Johnson, Risks in space from orbiting debris, Science 311 (5759) (2006) 340–341.

[2] C. Mu, S. Liu, M. Lu, Z. Liu, L. Cui, K. Wang, Autonomous spacecraft collision avoidance with a variable number of space debris based on safe reinforcement learning, Aerospace Science and Technology 149 (2024) 109131.

[3] C. Pardini, L. Anselmo, Evaluating the impact of space activities in low earth orbit, Acta Astronautica 184 (2021) 11–22.

[4] B. Wang, S. Li, J. Mu, X. Hao, W. Zhu, J. Hu, Research advancements in key technologies for space-based situational awareness, Space: Science & Technology 2022 (2022) 9802793.

[5] B. D. Little, C. E. Frueh, Space situational awareness sensor tasking: comparison of machine learning with classical optimization methods, Journal of Guidance, Control, and Dynamics 43 (2) (2020) 262–273.

[6] C. Han, Y. Gu, X. Sun, S. Liu, Rapid algorithm for covariance ellipsoid model based collision warning of space objects, Aerospace Science and Technology 117 (2021) 106960.

[7] Z. Liao, S. Wang, J. Shi, S. Zhiyong, Y. Zhang, M. B. Sial, Cooperative situational awareness of multi-uav system based on improved d-s evidence theory, Aerospace Science and Technology 142 (2023) 108605.

[8] S. Le Hegarat-Mascle, I. Bloch, D. Vidal-Madjar, Application of dempster-shafer evidence theory to unsupervised classification in multisource remote sensing, IEEE Transactions on Geoscience and Remote Sensing 35 (4) (1997) 1018–1031.

[9] O. C. Schrempf, D. Albrecht, U. D. Hanebeck, Tractable probabilistic models for intention recognition based on expert knowledge, in: 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2007, pp. 1429–1434.

[10] B. I. Ahmad, J. K. Murphy, S. Godsill, P. M. Langdon, R. Hardy, Intelligent interactive displays in vehicles with intent prediction: A bayesian framework, IEEE Signal Processing Magazine 34 (2) (2017) 82–94.

[11] A. Koenig, T. Rehder, S. Hohmann, Exact inference and learning in hybrid Bayesian Networks for lane change intention classification, in: 2017 IEEE Intelligent Vehicles Symposium (IV), 2017, pp. 1535–1540.

[12] J. Margarito, R. Helaoui, A. M. Bianchi, F. Sartor, A. G. Bonomi, User-independent recognition of sports activities from a single wrist-worn accelerometer: A template-matching-based approach, IEEE Transactions on Biomedical Engineering 63 (4) (2015) 788–796.

[13] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444.

[14] H. Zhang, J. Zhang, Z. Zhang, X. Ru, M. Shen, Intention inference for spacecraft proximity maneuver using cascaded neural network, Aerospace Science and Technology 168 (2026) 111128.

[15] C. Li, K. Wang, Y. Song, P. Wang, L. Li, Air target intent recognition method combining graphing time series and diffusion models, Chinese Journal of Aeronautics 38 (1) (2025) 103177.

[16] N. Mohammadi Foumani, L. Miller, C. W. Tan, G. I. Webb, G. Forestier, M. Salehi, Deep learning for time series classification and extrinsic regression: A current survey, ACM Computing Surveys 56 (9) (2024) 1–45.

[17] J. F. Torres, D. Hadjout, A. Sebaa, F. Martínez-Álvarez, A. Troncoso, Deep learning for time series forecasting: a survey, Big Data 9 (1) (2021) 3–21.

[18] Y. Zhang, P. Lu, Y. Wang, Intention recognition of non-cooperative space targets based on GMM-HMM, IFAC-PapersOnLine 59 (20) (2025) 1279–1284.

[19] R. Zhao, Intention recognition method for spatial non-cooperative target based on improved random forest, Advances in Space Research 77 (1) (2026) 714–728.

[20] Q. Sun, Z. Dang, Deep neural network for non-cooperative space target intention recognition, Aerospace Science and Technology 142 (2023) 108681.

[21] J. Li, Z. Yang, Y. Luo, Intention inference for space targets using deep convolutional neural network, Advances in Space Research 75 (2) (2025) 2184–2200.

[22] Q. Wen, T. Zhou, C. Zhang, C. Weiqi, Z. Ma, J. Yan, L. Sun, Transformers in time series: A survey, in: Proceedings of the 32nd International Joint Conference on Artificial Intelligence (IJCAI-23), International Joint Conferences on Artificial Intelligence Organization, 2023, pp. 6778–6786.

[23] G. Zerveas, S. Jayaraman, D. Patel, A. Bhamidipaty, C. Eickhoff, A transformer-based framework for multivariate time series representation learning, in: Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining, 2021, pp. 2114–2124.

[24] X. Tong, H. Cai, A. Hu, J. Zhang, Intention recognition of maneuvering spacecraft cluster using mic-net, IEEE Transactions on Aerospace and Electronic Systems 61 (5) (2025) 13286–13304.

[25] H. Zhang, J. Luo, Y. Gao, W. Ma, An intention inference method for the space non-cooperative target based on BiGRU-Self Attention, Advances in Space Research 72 (5) (2023) 1815–1828.

[26] Q. Sun, L. Zhao, S. Tang, Z. Dang, Orbital motion intention recognition for space non-cooperative targets based on incomplete time series data, Aerospace Science and Technology 158 (2025) 109912.

[27] Y. Chen, X. Zhang, W. Liao, G. Wei, S. Fan, Orbital behavior intention recognition for space non-cooperative targets under multiple constraints, Aerospace 12 (6) (2025) 520.

[28] Q. Sun, L. Zhao, Z. Dang, BiGAT: A model for recognizing motion intentions of space non-cooperative targets, IEEE Transactions on Aerospace and Electronic Systems (2024).

[29] Y. Yu, Y. Guo, P. Wang, H. Zhang, Intention recognition for space non-cooperative targets using kolmogorov-arnold transformer, Aerospace Science and Technology 170 (2026) 111497.

[30] H. Jing, Q. Sun, Z. Dang, H. Wang, Intention recognition of space non-cooperative targets using large language models, Space: Science & Technology 5 (2025) 0271.

[31] W. Yuan, Q. Xia, H. Qian, B. Qiao, J. Xu, B. Xiao, An intelligent hierarchical recognition method for long-term orbital maneuvering intention of non-cooperative satellites, Advances in Space Research 75 (6) (2025) 5037–5050.

[32] X. Tong, H. Cai, A. Hu, J. Zhang, Intention recognition of maneuvering spacecraft cluster using mic-net, IEEE Transactions on Aerospace and Electronic Systems 61 (5) (2025) 13286–13304.

[33] A. Zeng, M. Chen, L. Zhang, Q. Xu, Are transformers effective for time series forecasting?, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 37, 2023, pp. 11121–11128.

[34] V. Ekambaram, A. Jati, N. Nguyen, P. Sinthong, J. Kalagnanam, Tsmixer: Lightweight mlp-mixer model for multivariate time series forecasting, in: Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2023, pp. 459–469.

[35] S.-A. Chen, C.-L. Li, S. O. Arik, N. C. Yoder, T. Pfister, TSMixer: An all-MLP architecture for time series forecast-ing, Transactions on Machine Learning Research (2023).

[36] F. Fusco, D. Pascual, P. Staar, D. Antognini, pNLP-mixer: an efficient all-MLP architecture for language, in: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 5: Industry Track), 2023, pp. 53–60.

[37] J. You, Y. Wang, A. Pal, P. Eksombatchai, C. Rosenburg, J. Leskovec, Hierarchical temporal convolutional networks for dynamic recommender systems, in: The World Wide Web Conference, 2019, pp. 2236–2246.

[38] T. Wang, N. Strodthoff, S4Sleep: Elucidating the design space of deep-learning-based sleep stage classification models, Computers in Biology and Medicine 187 (2025) 109735.

[39] Q. Sun, L. Zhao, Z. Dang, Comprehensive classification of non-periodic relative motion styles of spacecraft: A geometric approach and applications, Astrodynamics 9 (6) (2025) 969–992.

[40] W. Clohessy, R. Wiltshire, Terminal guidance system for satellite rendezvous, Journal of the Aerospace Sciences 27 (9) (1960) 653–658.

[41] J. L. Ba, J. R. Kiros, G. E. Hinton, Layer normalization, arXiv preprint arXiv:1607.06450 (2016).

[42] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[43] D. Hendrycks, K. Gimpel, Gaussian error linear units (GELUs), arXiv preprint arXiv:1606.08415 (2016).

[44] D. P. Kingma, J. L. Ba, Adam: A method for stochastic optimization, in: Proceedings of the International Conference on Learning Representations (ICLR), 2015. doi:10.48550/arXiv.1412.6980.

[45] T. S. Kelso, Celestrak: Satellite orbital data, https://celestrak.org/satcat/, accessed: 2025-12-11 (2025).

[46] J. Tschauner, P. Hempel, Rendezvous zu einem in elliptischer Bahn umlaufenden Ziel, Astronautica Acta 11 (2) (1965) 104–109.

[47] K. Yamanaka, F. Ankersen, New State Transition Matrix for Relative Motion on an Arbitrary Elliptical Orbit, Journal of Guidance, Control, and Dynamics 25 (1) (2002) 60–66.