aerospace

MDPI

Article

Pre-Collision-Avoidance for LEO Spacecraft Using PACC-QMIX with an Orbital-Corridor Reward

Zhibo Su [1,2], Ruxuan Ding [1,2], Wenlin Tang [1,*], Wenlong Niu [1], Xiaodong Peng [1,3,4,5,*] and Jiafeng Zhang [1,*]

[1] Key Laboratory of Electronics and Information Technology for Space System, National Space Science Center, Chinese Academy of Sciences, Beijing 100190, China; suzhibo25@mails.ucas.ac.cn (Z.S.); dingruxuan25@mails.ucas.ac.cn (R.D.); niuwenlong@nssc.ac.cn (W.N.)
[2] University of Chinese Academy of Sciences, Beijing 100049, China
[3] School of Fundamental Physics and Mathematical Sciences, Hangzhou Institute for Advanced Study, UCAS, Hangzhou 310024, China
[4] Taiji Laboratory for Gravitational Wave Universe, Hangzhou 310024, China
[5] Key Laboratory of Gravitational Wave Precision Measurement of Zhejiang Province, Hangzhou 310024, China
* Correspondence: tangwenlin@nssc.ac.cn (W.T.); pxd@nssc.ac.cn (X.P.); zhangjiafeng@nssc.ac.cn (J.Z.)

Abstract

Low Earth orbit is becoming increasingly congested, and hazardous conjunctions are creating a growing collision risk. Conventional collision avoidance relies on emergency maneuvers during the final warning window, leaving little time for decision-making, limiting maneuver opportunities, and consuming more fuel. To address these problems, this paper proposes a pre-emptive collision-avoidance framework that uses the longer period before an avoidance warning to plan small maneuvers in advance. These maneuvers move the spacecraft into a safe orbital corridor with a lower overall collision threat. We formulate pre-collision-avoidance as a multistage sequential decision-making problem and propose pre-action-component cooperative QMIX (PACC-QMIX). The maneuver decision is divided among three agents that determine the action components and one agent that determines the maneuver time. The shaped reward combines a normalized environment reward, an orbital-corridor reward, a maneuver-count penalty, and a separate collision-probability penalty. The collision-avoidance window is set to 1 day, while the pre-collision-avoidance planning windows are set to 7 and 14 days. Across ten randomized scenarios, the 7-day window increases the mean reward by 60.7% relative to the 1-day window, demonstrating the effectiveness of pre-collision-avoidance maneuvers. Under the 1-day and 7-day windows, PACC-QMIX achieves mean rewards of −0.0840 and −0.0330, respectively. Compared with MAPPO, Double DQN, and cross-entropy optimization, PACC-QMIX has the highest mean reward among the four methods in both windows. Five configurations are considered in the sensitivity experiments on the maneuver window and collision-probability threshold: <14-day, 1×10^{-8}>, <14-day, 1×10^{-6}>, <14-day, 1×10^{-4}>, <7-day, 1×10^{-4}>, and <1-day, 1×10^{-4}>. The rewards for the 14-day window range from −0.0325 to −0.0276, while the reward for the seven-day window is −0.0330. These results show that the <7-day, 1×10^{-4}> configuration provides a favorable balance among collision-risk control, fuel consumption, and pre-collision-avoidance lead time. For ten selected 7-day pre-collision-avoidance scenarios, stored maneuvers obtained in a two-body-only orbital environment are tested in an orbital environment including J_2 perturbation and atmospheric drag. The experimental results show that the collision probabilities of all ten cases remain far below the threshold of 1×10^{-5} when the corresponding strategies are applied. In summary, PACC-QMIX achieves effective pre-collision-avoidance. By planning maneuvers in advance, it reduces fuel consumption, extends the orbital lifetime of satellites, and offers a new approach to the long-term stable operation of large-scale constellations.

Check for updates

Academic Editors: Mingying Huo, Gang Zhang and Zichen Fan

Received: 27 August 2026
Revised: 1 October 2026
Accepted: 8 October 2026
Published: 27 October 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

Keywords: low Earth orbit; pre-collision-avoidance; action decomposition; multi-agent reinforcement learning; orbital-corridor reward

1. Introduction

Large satellite constellations and continued debris accumulation have increased congestion in low Earth orbit (LEO) [1–3]. At typical orbital relative velocities, even a small fragment can severely damage an operational spacecraft. A collision may also generate additional debris and contribute to the Kessler syndrome [4]. Collision-avoidance planning must reduce risk while limiting fuel consumption, operational disruption, and deviation from the mission orbit. In operational conjunction assessment, a per-event collision probability of 10^{-4} is widely accepted as a reference threshold above which mitigation is recommended [5]. Unlike the probability threshold, the timing of a mitigation decision is not governed by a universal lead time. Operators instead define a mission-specific maneuver commitment point, which is the latest time at which a decision must be made for the maneuver to be implemented before the time of closest approach (TCA) [5]. Pre-collision-avoidance planning spans several orbital revolutions and provides additional candidate maneuver epochs. For favorable encounter geometries, a small velocity increment applied earlier can accumulate sufficient separation by conjunction, potentially reducing fuel consumption and deviation from the mission orbit.

Conventional collision-avoidance formulations include impulsive velocity-increment minimization [6], low-thrust maneuver design [7–9], convex optimization for multiple impulses or encounters [10,11], and constrained cross-entropy (CE) optimization [12]. These approaches provide established tools for individual encounters. Long-horizon multistage planning is more difficult because each maneuver changes the protected orbit, encounter geometry, collision probability, remaining fuel, and subsequent control options. Joint selection of maneuver vectors and execution times consequently forms a high-dimensional, potentially nonconvex problem with temporal coupling. A serial planning method must account for these state transitions rather than optimize each maneuver stage in isolation. These computational and temporal-coupling challenges have motivated the exploration of learning-based methods for collision-risk estimation and sequential maneuver planning.

Learning-based research on collision-avoidance maneuver planning spans several related directions. Data-driven models support collision-risk estimation, orbit prediction, maneuver detection, and sensor tasking for space situational awareness [13–19]. Reinforcement learning has also been applied to cooperative avoidance and single-spacecraft planning with continuous actions, multiple debris objects, uncertainty, and mission constraints [20–25]. In multi-agent reinforcement learning (MARL), multi-agent deep deterministic policy gradients (MADDPG) and counterfactual multi-agent policy gradients (COMA) address coordination and credit assignment [26,27]. Value-decomposition networks (VDN) and QMIX instead represent a joint action value through local utility functions [28,29]. Subsequent studies explore improved monotonic mixing, attention-based value decomposition, multi-agent proximal policy optimization (MAPPO), and value-decomposition actor–critic methods [30–33]. Action branching offers a related approach to representing combinatorial actions through smaller decision branches [34].

For long-horizon collision avoidance, the learning architecture must accommodate coordination among maneuver components and temporal coupling across successive stages. Within each stage, the velocity-increment components and relative waiting time jointly determine the resulting trajectory. Discretizing these components creates a joint action space whose size grows multiplicatively with the number of choices per component. Across

stages, orbital propagation changes the state and the effectiveness of subsequent maneuvers. These features call for a method that coordinates action components within each stage while carrying the effects of earlier maneuvers into later decisions.

Several multi-agent reinforcement learning methods are relevant to this formulation, including MADDPG, COMA, VDN, QMIX, and MAPPO [26–29,32]. MADDPG was designed primarily for continuous control [26], whereas COMA and MAPPO can accommodate cooperative discrete decisions [27,32]. Unlike VDN's additive decomposition [28], QMIX uses a state-conditioned nonlinear mixing network to represent a broader class of joint action values [29]. The monotonicity constraint allows component-wise greedy choices to maximize the represented joint value without enumerating all joint actions [29].

Pre-collision-avoidance is formulated as a multistage serial decision process with soft constraint penalties, and pre-action-component cooperative QMIX (PACC-QMIX) is proposed to solve the resulting planning problem. The method applies QMIX to the control dimensions of one spacecraft rather than coordinating multiple maneuvering spacecraft. At each stage, four agents select three velocity-increment components and a relative waiting time. A state-conditioned monotonic mixing network combines their utilities to estimate the joint stage-action value. After each maneuver, nonlinear orbital propagation updates the state for the next decision.

The shaped reward combines a normalized environment reward, an orbital-corridor reward, a maneuver-count penalty and a separate piecewise collision-probability penalty. The corridor term rewards maneuver sequences that retain a larger local margin when velocity-increment magnitudes are scaled. The evaluation covered one-day short-horizon collision avoidance and pre-collision-avoidance over seven-day and 14-day planning horizons [5,35,36]. The study comprised four experiments. First, PACC-QMIX was compared with CE, Double DQN and MAPPO under 1-day and 7-day planning horizons across ten randomized scenarios. Second, sensitivity to the planning horizon and collision-probability threshold was examined. Third, reward ablation and coefficient sensitivity were evaluated in Scenario 5 using five independently trained PACC-QMIX seeds. Fourth, stored 7-day action sequences from ten selected scenarios were tested unchanged in an orbital environment including J_2 perturbation and atmospheric drag, without retraining or new action selection.

The remainder of this paper is organized as follows. Section 2 formulates the pre-collision-avoidance problem and describes the orbital dynamics, maneuver model and optimization objective. Section 3 presents the simulation environment, serial action-selection mechanism, PACC-QMIX architecture, reward design and training procedure. Section 4 describes the experimental design and results. Section 5 summarizes the main findings and limitations.

2. Problem Formulation

This study considers collision avoidance for one maneuverable protected satellite and several debris objects in LEO. Only the protected satellite can execute the actions selected by the policy. The debris trajectories follow from their initial states and the orbital-dynamics model. Because the planning horizon may contain several predicted conjunctions, the protected satellite must select a sequence of maneuvers before the hazardous encounters occur. The aim is to keep the joint collision probability below a prescribed threshold while limiting fuel consumption and departure from the mission orbit.

2.1. Pre-Collision-Avoidance Problem

This study considers collision avoidance over a multi-orbit planning horizon rather than an emergency correction shortly before conjunction. Earlier planning provides more

candidate maneuver epochs and permits the required orbital correction to be distributed across several stages.

Let the protected satellite be P, and let the system contain N_d debris objects. The i-th object is denoted by D_i, where $i = 1, 2, ..., N_d$. The predicted conjunction time between the protected satellite and the i-th object is

$$t_{c,i}, i = 1, 2, \cdots, N_d \,, \tag{1}$$

where $t_{c,i}$ is the predicted hazardous-conjunction time between the protected satellite and the i-th debris object. If several conjunctions occur within the planning horizon, the policy considers their combined collision risk.

The pre-collision-avoidance planning window is defined as

$$[t_0 \,, t_f], \tag{2}$$

where t_0 and t_f are the start and end times, respectively. In general, t_f does not exceed the principal conjunction time. If the window contains K maneuver stages, stage j contains one candidate maneuver, with $j = 1, 2, ..., K$.

For favorable conjunction geometries, earlier planning allows a small velocity change to accumulate separation before conjunction, potentially reducing maneuver effort and orbital deviation relative to a late correction.

The maneuver stages are executed serially. After the orbit is propagated and the stage j action is executed, stage $j + 1$ observes the updated environment. The stages are therefore neither independent nor determined simultaneously at the initial epoch. Their temporal coupling can be written as

$$\boldsymbol{s}_1 \overset{\boldsymbol{u}_1}{\rightarrow} \boldsymbol{s}_2 \overset{\boldsymbol{u}_2}{\rightarrow} \cdots \overset{\boldsymbol{u}_K}{\rightarrow} \boldsymbol{s}_f \,, \tag{3}$$

where $\boldsymbol{S}_j$ is the environment state at the start of stage j, and $\boldsymbol{u}_j$ is the corresponding maneuver control. The state $\boldsymbol{s}_j$ is obtained after executing all preceding stages and propagating the system to the start epoch of stage j.

The problem is therefore a long-horizon serial decision process. Each maneuver changes the protected orbit, encounter geometry, collision probability, remaining fuel, and actions that remain useful at later stages.

2.2. Coordinate System

Positions and velocities are expressed in an Earth-centered inertial (ECI) frame. The protected satellite's orbit is described by classical orbital elements [37].

$$\boldsymbol{OE} = (a, e, i, \Omega, \omega, M), \tag{4}$$

where $\boldsymbol{OE}$ is the orbital-element vector. The elements a, e, i, Ω, ω, and M denote semimajor axis, eccentricity, inclination, right ascension of the ascending node, argument of perigee, and mean anomaly, respectively.

During propagation, the protected satellite's state at time t is

$$\mathbf{x}_p\,(t) = [\mathbf{r}_p\,(t), \boldsymbol{v}_p\,(t)]^T, \tag{5}$$

where $\boldsymbol{r}_p(t) = [x_p(t), y_p(t), z_p(t)]^T$ and $\boldsymbol{v}_p(t) = [v_{x,p}(t), v_{y,p}(t), v_{z,p}(t)]^T$ are the ECI position and velocity vectors of the protected satellite, respectively.

Debris objects are initialized from Cartesian position and velocity at their hazardous-conjunction epochs rather than from six orbital elements. The state of object i at the corresponding conjunction time $t_{c,i}$ is

$$\boldsymbol{x}_{d,i}\,(t_{c,i}\,) = [\boldsymbol{r}_{d,i}\,(t_{c,i}\,), \boldsymbol{v}_{d,i}\,(t_{c,i}\,)]^T, i = 1, 2, \cdots, N_d \,, \tag{6}$$

where $\boldsymbol{r}_{d,i}(t_{c,i})$ and $\boldsymbol{v}_{d,i}(t_{c,i})$ are the ECI position and velocity vectors of debris object i at the corresponding conjunction time.

The state of debris object i at a later time t is obtained by propagating the reference-epoch state from $t_{c,i}$ to t.

$$\boldsymbol{x}_{d,i}(t) = \mathcal{F}_{t,t_{c,i}}\boldsymbol{x}_{d,i}(t_{c,i}), \tag{7}$$

where $\mathcal{F}_{t,t_{c,i}}$ denotes the nonlinear two-body Keplerian flow map from $t_{c,i}$ to t.

2.3. Dynamics Model

The primary training and evaluation experiments use a two-body-only orbital environment for the protected satellite and debris objects. This model includes Earth's central gravity but omits nonspherical gravity, atmospheric drag, solar radiation pressure, and third-body gravity. Section 4.5 separately tests the stored maneuver sequences in an orbital environment including J_2 perturbation and atmospheric drag.

Under central gravity, the equation of motion is

$$\ddot{\boldsymbol{r}} = -\frac{\mu}{r^3}\boldsymbol{r}, \tag{8}$$

where $\boldsymbol{r} = [x,y,z]^T$ is the geocentric position vector, $r = \|\boldsymbol{r}\|$ denotes the magnitude of the geocentric position vector, and μ is Earth's standard gravitational parameter.

In the absence of a maneuver, the protected satellite propagates according to

$$\dot{\boldsymbol{x}}_p(t) = f(\boldsymbol{x}_p(t)). \tag{9}$$

The i-th debris object propagates according to

$$\dot{\boldsymbol{x}}_{d,i}(t) = \mathcal{F}\big(\boldsymbol{x}_{d,i}(t), \Delta t\big), \quad i = 1, \ldots, N_d. \tag{10}$$

where $\mathcal{F}$ denotes the nonlinear two-body Keplerian flow map over the propagation interval Δt.

2.4. Maneuver Model

The protected satellite applies impulsive velocity changes to reduce collision risk at future conjunctions. Under the impulsive approximation, maneuver duration is negligible relative to the orbital period, and each impulse changes the spacecraft velocity instantaneously [6,38]. The total correction can be distributed among serial stages separated by orbital propagation. This setting follows the multiple-impulse collision-avoidance concept [10] and assigns a velocity increment and a waiting time to each stage.

The serial impulsive model contains at most K decision stages. At stage j, the control vector contains three ECI velocity-increment components and a relative waiting time:

$$\boldsymbol{u}_j = [\Delta v_{x,j}, \Delta \mathrm{v}_{y,j}, \Delta \mathrm{v}_{z,j}, \tau_j]^T, \quad j = 1, 2, \ldots, K, \tag{11}$$

where $\Delta v_{x,j}$, $\Delta v_{y,j}$, and $\Delta v_{z,j}$ are the velocity-increment components at stage j, and τ_j is the waiting time from the preceding stage epoch. The velocity-increment components are expressed in m s^{-1}, τ_j is expressed in days, and all epochs use the MJD2000 scale. Including τ_j in the stage control determines both the maneuver vector and the maneuver execution epoch.

The velocity-increment vector at stage j is

$$\Delta \boldsymbol{v}_j = [\Delta v_{x,j}, \Delta v_{y,j}, \Delta v_{z,j}]^T. \tag{12}$$

The effectiveness of an impulsive maneuver depends on both the applied velocity increment and the available lead time before the predicted conjunction [6,39]. Let t_{j-1}

denote the epoch preceding stage j, and let t_f denote the endpoint of the planning horizon. The execution epoch of stage j is

$$t_j = \min(t_{j-1} + \tau_j, t_f), \tag{13}$$

The protected satellite is first propagated from t_{j-1} to t_j, after which $\Delta \boldsymbol{v}_j$ is applied. Because t_j depends on the preceding waiting times, later maneuver epochs are not specified independently at the beginning of the planning horizon.

Under the standard impulsive-maneuver assumption, spacecraft position remains continuous across the maneuver epoch, whereas velocity changes instantaneously [6,38]. The state transition at stage j is therefore

$$\begin{aligned} \boldsymbol{r}_p(t_j^+) &= \boldsymbol{r}_p(t_j^-), \\ \boldsymbol{v}_p(t_j^+) &= \boldsymbol{v}_p(t_j^-) + \Delta \boldsymbol{v}_j, \end{aligned} \tag{14}$$

where $t_j{}^-$ and $t_j{}^+$ denote the instants immediately before and after the stage j impulse. The vectors $\boldsymbol{r}_p$ and $\boldsymbol{v}_p$ are the corresponding position and velocity of the protected satellite. Each impulse changes the state from which subsequent stages begin, affecting later propagation, conjunction geometry, and collision probability. These dependencies make the maneuver sequence a serially coupled multistage decision problem.

When $\|\Delta \mathbf{v}_j\|_2 = 0$, no impulse is applied at stage j. The waiting time τ_j remains active, so orbital propagation continues to the next decision epoch. This zero-maneuver option preserves temporal progression without requiring an unnecessary impulse.

The cumulative velocity increment measures the total maneuver effort of the serial sequence [10,39]:

$$\Delta V_{sum} = \sum_{j=1}^{K} \left\| \Delta \boldsymbol{v}_j \right\|_2, \tag{15}$$

Equation (15) sums the Euclidean norms of the velocity increments applied at each stage. We report fuel consumption using the linear proxy $m_{f,proxy} = c_f \Delta V_{sum}$, where $c_f = 1$ kg/(m s^{-1}). The reported value in kilograms is therefore numerically equal to ΔV_{sum} in m s^{-1}. The initial fuel value, fuel-use limit, and all reported fuel-consumption results follow this convention. Note that this proxy is a measure of maneuver effort, not propellant mass calculated from spacecraft mass, specific impulse, or the rocket equation.

2.5. Constrained Optimization Model

The pre-collision-avoidance task is formulated as a long-horizon constrained optimization problem. Let the planning window contain K serial decision stages. With the stage control $\boldsymbol{u}_j$ defined by Equation (11), the complete maneuver sequence is

$$\boldsymbol{U} = \left\{ \boldsymbol{u}_j \right\}_{j=1}^{K}. \tag{16}$$

where K is the maximum number of decision stages. The controls in $\boldsymbol{U}$ are executed serially, and each stage begins from the orbital state produced by the preceding stages.

The execution epoch at stage j depends on the preceding epoch and the relative waiting time:

$$t_j = \min\left(t_{j-1} + \tau_j, t_f\right), \quad j = 1, 2, \ldots, K. \tag{17}$$

Orbital propagation yields the protected satellite's state history and the joint collision probability, cumulative velocity increment, and orbital deviation. The cumulative velocity increment is calculated using Equation (15).

The optimization objective combines collision risk, maneuver effort, and departure from the reference orbit:

$$\min_{\boldsymbol{U}} J(\boldsymbol{U}) = w_p \frac{P_c(\boldsymbol{U})}{P_{th}} + w_v \frac{\Delta V_{sum}(\boldsymbol{U})}{\Delta V_{sum,max}} + w_o D_{OE}(\boldsymbol{U})$$
$$s.t. \begin{cases} P_c(\boldsymbol{U}) \le P_{th}, \\ \Delta V_{sum}(\boldsymbol{U}) \le \Delta V_{sum,max}, \\ D_{OE}(\boldsymbol{U}) \le 1, \\ \| \Delta \boldsymbol{v}_j \| \le \Delta V_{single,max}, j = 1, 2, \cdots, K, \\ 0 \le \tau_j \le \tau_{\max}, j = 1, 2, \cdots, K, \\ t_j = \min\left(t_{j-1} + \tau_j, t_f\right), j = 1, 2, \cdots, K. \end{cases} \tag{18}$$

Equation (18) defines $J(\boldsymbol{U})$ for the complete maneuver sequence. The first objective term is the joint collision probability, normalized by the prescribed threshold P_{thr}. The second term is the cumulative velocity increment $\Delta V_{sum}(\boldsymbol{U})$, normalized by the upper limit $\Delta V_{sum,max}$. The orbital-element deviation vector $\Delta \boldsymbol{OE}(\boldsymbol{U})$ is normalized by the allowable deviation vector $\Delta \boldsymbol{OE}(\boldsymbol{U})_{\max}$ to obtain $D_{OE}(\boldsymbol{U})$. Orbital preservation requires $D_{OE}(\boldsymbol{U}) \le 1$. At stage j, the velocity increment is bounded by $\|\Delta \boldsymbol{v}_j\|_2 \le \Delta V_{single,max}$ and the relative waiting time satisfies $0 \le \tau_j \le \tau_{max}$. The maneuver epochs follow the recursive timing relation defined in Equation (17).

Direct optimization of Equation (18) is difficult because a K stage sequence contains $4K$ coupled control variables. Each maneuver changes the orbital state from which all subsequent stages begin, while each waiting time shifts the execution epochs of the remaining controls. Consequently, modifying one stage can alter all later trajectories, encounter geometries, collision probabilities, and orbital deviations. Nonlinear Keplerian propagation maps the complete control sequence to the objective and constraint values, which can be evaluated only after the sequence has been propagated. Equation (18) therefore defines a temporally coupled and potentially nonconvex constrained optimization problem whose solution method is presented in Section 3.

3. Materials and Methods

This section presents PACC-QMIX and the orbital-corridor margin calculation for the optimization problem defined in Section 2. The planning window is divided into K serial maneuver stages. At each stage, action-component agents select discrete controls from the current orbital state. An action mapper combines these controls into a velocity increment and a relative waiting time. The environment advances the decision epoch, executes the maneuver, propagates the orbital states, and returns the next state and stage reward.

Within each stage, PACC-QMIX coordinates the action components. Orbital propagation links decisions across stages. Three utility networks evaluate the Cartesian velocity-increment components, and a fourth evaluates the relative waiting time. A state-conditioned mixing network combines their selected utilities to estimate the joint action value. Since each maneuver changes the protected orbit, conjunction geometry, fuel state, and collision risk, the next stage begins from a state shaped by all preceding actions.

At each stage, the environment maps the selected controls to a maneuver and propagates the resulting state. The orbital-corridor margin and shaped reward then evaluate collision risk, maneuver effort, and orbital deviation. These rewards provide feedback for selecting actions over the planning window.

3.1. Orbital Environment and Sequential Decision Formulation

3.1.1. Orbital Simulation Environment

The simulation environment provides the orbital dynamics and risk-evaluation interface required for policy training and evaluation. Each scenario contains one protected

satellite and N_d debris objects. The environment comprises five functional modules: object initialization, orbital propagation, collision-probability estimation, maneuver execution, and reward calculation.

During initialization, the protected satellite and debris objects are constructed from their reference-epoch states. The protected satellite can change the protected satellite's orbit through the maneuver actions selected by the policy. The debris objects do not perform active maneuvers and evolve according to the adopted orbital dynamics.

The propagation module computes the position and velocity of each object over the planning window. Debris trajectories are propagated directly from their initial states. For the protected satellite, the selected velocity increment is applied at each maneuver epoch before propagation continues. This procedure produces the maneuvered trajectory and the relative encounter geometry between the protected satellite and each debris object.

The collision module detects hazardous approaches and estimates the corresponding collision probabilities using the encounter geometry and position uncertainty. The implementation adopts the collision-probability approximation described in Ref. [40]. Under the approximation that the detected conjunction events are statistically independent, their joint collision probability is

$$P_c = 1 - \prod_{i=1}^{N_c} (1 - P_i), \tag{19}$$

where P_i is the collision probability of conjunction i, N_c is the number of detected conjunctions, and P_c is the joint collision probability over the planning window.

The maneuver module maps the discrete action components to continuous controls and executes the waiting and impulsive-maneuver operations in serial order. After each stage, the environment updates the protected orbit, joint collision probability, remaining fuel, and orbital-element deviation. These quantities are then used to construct the next state and calculate the stage reward.

3.1.2. State Space

All action-component agents receive the same environment state and use one-hot identities to distinguish control dimensions. The state contains normalized time and stage progress, remaining fuel, collision risk, protected-satellite state, debris states, and orbital deviation.

For one protected satellite and N_d debris objects, the environment state is

$$\boldsymbol{s}_j = \left[\eta_t, \eta_s, \eta_f, \widetilde{P}_c, \boldsymbol{x}_p(t_j), \boldsymbol{x}_{d,1}(t_j), \ldots, \boldsymbol{x}_{d,N_d}(t_j), \Delta\boldsymbol{OE}(t_j)\right]. \tag{20}$$

The quantities η_t, η_s, η_f, denote normalized simulation time, maneuver-stage progress, and remaining fuel, respectively. The normalized collision-risk feature used in the implementation is $\widetilde{P}_c = clip(P_c/P_{thr}, 0, 10)/10$. Position and velocity components are divided by 10^7 m and 10^4 m s^{-1}, respectively. Each constrained orbital-element deviation is divided by the allowable magnitude of the corresponding deviation and subsequently clipped to control extreme values. These transformations place state variables with different physical units on comparable numerical scales.

The protected satellite state at t_j is denoted by $\boldsymbol{x}_p(t_j)$, following the definition in Equation (5).

For each debris object i, the initial state at the corresponding conjunction epoch $t_{c,i}$ follows the Cartesian state definition in Equation (6).

During RL interaction, the state contains each debris object after propagation to the current decision time:

$$\boldsymbol{x}_{d,i}(t_j) = \varphi\left(t_j, t_{c,i}; \boldsymbol{x}_{d,i}(t_{c,i})\right), \tag{21}$$

where $\boldsymbol{x}_{d,i}(t_{c,i})$ is the reference-epoch state of debris object i, and $\varphi(\cdot)$ denotes the nonlinear two-body orbital flow associated with the dynamics model in Equation (8). The flow map is nonlinear and does not constitute a fixed linear state-transition matrix. Because the debris objects do not maneuver, their states depend only on their reference-epoch states, the elapsed propagation time, and the adopted orbital dynamics.

After the action at stage j, the environment advances the decision epoch, executes the selected velocity increment, and propagates all orbital states. The environment then recalculates the joint collision probability, remaining fuel, and orbital-element deviation. These quantities form the next state $\boldsymbol{s}_{j+1}$ and provide the global information required to coordinate subsequent action components.

3.1.3. Serial Stages and Action-Component Agents

Here, an agent is a computational action component rather than an independently maneuvering spacecraft.

Pre-collision-avoidance requires several temporally coupled decisions over a long planning horizon. The planning horizon is divided into K serial maneuver stages. The stages are not treated as separate agents. Instead, their dependence arises from the orbital transition $\boldsymbol{s}_j \rightarrow \boldsymbol{s}_{j+1}$. The maneuver selected at stage j changes the orbit, conjunction geometry, fuel state, and orbital deviation observed at stage $j+1$.

To reduce the joint search dimension, each stage control is decomposed into cooperative action components. Four agents select the three velocity-increment components and relative waiting time:

$$\mathcal{G} = \{g_x, g_y, g_z, g_\tau\}. \tag{22}$$

where $\mathcal{G}$ is the set of four action-component agents. The agents g_x, g_y, g_z, and g_τ select ΔV_x, ΔV_y, ΔV_z, and the relative waiting time τ, respectively.

At stage j, each agent selects a discrete action from the current state. Their joint discrete action is

$$\boldsymbol{a}_j = \left(a_{x,j}, a_{y,j}, a_{z,j}, a_{\tau,j}\right). \tag{23}$$

The action mapper then produces the continuous stage-control vector $\boldsymbol{u}_j$ defined in Equation (11).

Within each stage, the PACC-QMIX mixer coordinates the local action values associated with the four control components. Across stages, the transition $\boldsymbol{s}_j \rightarrow \boldsymbol{s}_{j+1}$ carries the effects of earlier maneuvers into subsequent decisions. The resulting formulation therefore combines component-level coordination within a stage with serial decision making across stages.

3.1.4. Discrete Action Mapping

QMIX operates on discrete action spaces, whereas orbital maneuvers are defined by continuous velocity increments and waiting times. PACC-QMIX therefore discretizes each component of the stage control while retaining the continuous orbital propagation model.

At stage j, the four agents select actions from their respective discrete sets:

$$\begin{aligned} a_{x,j} &\in \mathcal{A}_x, \quad a_{y,j} \in \mathcal{A}_y, \\ a_{z,j} &\in \mathcal{A}_z, \quad a_{\tau,j} \in \mathcal{A}_\tau, \end{aligned} \tag{24}$$

The sets $\mathcal{A}_x$, $\mathcal{A}_y$, $\mathcal{A}_z$ contain the discrete levels of the three ECI velocity-increment components, while $\mathcal{A}_\tau$ contains the discrete waiting-time levels. Each velocity component has N_v admissible levels, and the waiting time has N_τ levels.

The joint discrete action at stage j is therefore

$$\boldsymbol{a}_j = \left(a_{x,j}, a_{y,j}, a_{z,j}, a_{\tau,j}\right) \in \mathcal{A}_x \times \mathcal{A}_y \times \mathcal{A}_z \times \mathcal{A}_\tau. \tag{25}$$

The mapping function converts the discrete components into a continuous maneuver control:

$$\boldsymbol{u}_j = \mathcal{M}(\boldsymbol{a}_j) = \left[\Delta V_{x,j}, \Delta V_{y,j}, \Delta V_{z,j}, \tau_j\right]^{\mathrm{T}}. \tag{26}$$

where $\mathcal{M}(\cdot)$ maps the discrete action-component combination to the continuous maneuver control.

When all three velocity-increment components are zero,

$$\Delta V_{x,j} = \Delta V_{y,j} = \Delta V_{z,j} = 0 \tag{27}$$

Equation (27) represents a no-maneuver stage. The selected waiting time τ_j remains active, so the environment continues to propagate to the next decision epoch. A separate four-dimensional no-maneuver action is therefore unnecessary.

This decomposition preserves the four physical control variables while replacing a single joint action space of size ${N_v}^3 N_\tau$ with four smaller component spaces. The mixer subsequently evaluates their joint effect on the stage-level action value.

The resulting action table specifies a two-stage maneuver sequence for the simulated conjunction. At each stage, the action table records a waiting time and three ECI velocity-increment components drawn from the discrete sets in Table 1. The simulator applies the selected velocity increment instantaneously at the designated maneuver epoch. Then the simulator propagates the orbit to the next decision stage, where the updated state informs selection of the second maneuver. After both stages, the resulting trajectory is used to assess collision risk. The reported outcomes therefore reflect an instantaneous-velocity-change approximation. The action table records the velocity increments applied at the selected maneuver times.

Table 1. Environment, action-space, and physical-constraint parameters used in the PACC-QMIX experiments.

Category	Parameter	Value
Environment	Protected satellites per scenario	1
Environment	Debris objects per scenario	5
Experimental protocol	Policy-training protocol	One separate PACC-QMIX policy per scenario
Maneuver and action space	Serial decision stages	2
Maneuver and action space	Action-component agents	4
Maneuver and action space	Action-space dimensions	[11, 11, 11, 3]
Maneuver and action space	Velocity-increment components	$\{0, \pm 0.002, \pm 0.004, \pm 0.006, \pm 0.008, \pm 0.010\}$ m s^{-1}
Maneuver and action space	Waiting-time actions	{0, 3, 6} days
Physical constraints	Initial fuel-proxy value	10 kg
Physical constraints	Cumulative fuel-consumption limit	10 kg
Physical constraints	Orbital-element-deviation limits	(100, 0.01, 0.01, 0.01, 0.01) sixth component unconstrained
State representation	Environmental-state dimension	46

3.2. Orbital-Corridor and Reward Design

3.2.1. Orbital-Corridor Reward

A collision-avoidance maneuver should reduce future collision risk without producing excessive fuel consumption or orbital deviation. If the policy is trained only to reduce collision probability, the policy may favor unnecessarily large velocity increments. The orbital-corridor margin calculation is therefore introduced to estimate the local maneuver-magnitude margin around the selected maneuver sequence. The normalized margin is then used to construct the orbital-corridor reward.

For each nonzero maneuver, the orbital-corridor margin calculation scales the velocity-increment magnitude while preserving the direction of the velocity-increment vector. The complete maneuver table is then re-evaluated through orbital propagation, collision-probability calculation, fuel calculation, and orbital-deviation checks. The interval of feasible scaling factors is converted into a local maneuver-magnitude margin.

The local maneuver-magnitude margin is incorporated as a constraint-informed reward term. This reward term favors maneuver sequences that remain feasible over a wider range of local magnitude scaling.

For each stage j, the velocity-increment vector $\Delta \boldsymbol{v}_j$ is defined in Equation (12). A positive scaling factor s perturbs the maneuver magnitude to probe the admissible scaling interval used to determine the local maneuver-magnitude margin.

$$\Delta \boldsymbol{v}_j' = s\Delta \boldsymbol{v}_j, \tag{28}$$

Values $s > 1$ and $0 < s < 1$ increase and decrease the maneuver magnitude, respectively. A scaled maneuver contributes to the local maneuver-magnitude margin when the scaled maneuver satisfies the collision-probability, fuel-consumption, and orbital-preservation constraints. Scaled maneuvers that fail to satisfy these constraints fall outside the admissible scaling interval.

Because the scaling factor s must remain positive and may vary over several orders of magnitude, the boundary search is performed in logarithmic coordinates by defining

$$s = \exp(\alpha), \tag{29}$$

where $\alpha = ln(s)$ is the logarithmic maneuver-scaling variable. Searching over s is therefore equivalent to searching over α. At stage j, separate bisection searches are conducted below and above $\alpha = 0$ to determine the feasible scaling bounds ${s_j}^{min}$ and ${s_j}^{max}$. For each candidate scaling factor, the velocity increment at stage j is rescaled in the complete maneuver table. The resulting maneuver sequence is then re-evaluated through orbital propagation, collision-probability calculation, fuel-consumption calculation, and orbital-deviation checks.

For effective maneuver j, the local maneuver-magnitude margin is

$$W_j = (s_j^{max} - s_j^{min})\left\|\Delta \boldsymbol{v}_j\right\| \tag{30}$$

where $\|\Delta \boldsymbol{v}_j\|$ is the magnitude of the velocity increment at stage j, $[{s_j}^{min}, {s_j}^{max}]$ is the admissible scaling interval for stage j, and W_j is the local maneuver-magnitude margin for stage j.

If $\mathrm{N_{man}}$ nonzero maneuvers are executed, their mean local maneuver-magnitude margin is

$$W = \frac{1}{N_{\mathrm{man}}} \sum_{j \in \mathcal{J}_{\mathrm{man}}} W_j, \tag{31}$$

where $\mathcal{J}_{\mathrm{man}}$ is the index set of stages that execute nonzero velocity increments.

A stage with three zero velocity components is excluded from the orbital-corridor margin calculation. If every stage selects zero, the orbital-corridor reward is zero so that an ineffective sequence receives no additional reward.

Because local maneuver-magnitude margins can differ substantially across scenarios, the mean local maneuver-magnitude margin is mapped to a bounded interval:

$$W_{norm} = \frac{\overline{W}}{\overline{W} + W_0}, \quad W_0 > 0. \tag{32}$$

where W_0 is the reference value used to normalize the local maneuver-magnitude margin. In the present implementation, $W_0 = 1$ in the same velocity-increment unit as $\overline{W}$. The normalized local maneuver-magnitude margin W_{norm} lies in $[0, 1)$ and increases monotonically with the mean local maneuver-magnitude margin $\overline{W}$. The normalized quantity remains a local maneuver-magnitude margin rather than a general robustness measure.

The normalized local maneuver-magnitude margin is used as the orbital-corridor reward. The orbital-corridor reward encourages maneuvers that reduce risk while retaining a larger local maneuver-magnitude margin with respect to the prescribed collision-probability, fuel, and orbital-deviation boundaries.

3.2.2. Reward Design

The shaped reward combines a normalized environment reward, an orbital-corridor reward, a maneuver-count term, and a separate collision-probability penalty.

$$R_{final} = \mathrm{clip}\left(\lambda_{env} R_{env} + \lambda_{corr} R_{corr} + \lambda_{man} R_{man} - \lambda_p R_p, R_{\min}, R_{\max}\right) \tag{33}$$

Here, R_{env} denotes the normalized environment reward, R_{corr} the orbital-corridor reward, and R_{man} the maneuver-count term. The non-negative R_p denotes the collision-probability penalty. Each coefficient is obtained by dividing the corresponding original weight by the sum of all four original weights. The normalized coefficients therefore satisfy $\lambda_{env} + \lambda_{corr} + \lambda_{man} + \lambda_p = 1$. A larger collision-probability penalty lowers the shaped reward through the term $-\lambda_p R_p$. Section 4.1 reports the coefficient values, probability breakpoints, normalization scale, and clipping bounds.

The raw environment reward aggregates penalties associated with collision probability, the fuel-use proxy, and orbital-element deviation. To reduce differences in reward magnitude across scenarios, the raw environment reward is normalized as follows.

$$R_{\mathrm{env}} = \tanh\left(\frac{r_{\mathrm{env}}}{\kappa_{\mathrm{env}}}\right), \tag{34}$$

where r_{env} is the raw environment reward and $\kappa_{\mathrm{env}} > 0$ is a scale parameter. The transformation reduces the effect of extreme values without changing their ordering.

The orbital-corridor reward is the normalized local maneuver-magnitude margin defined in Section 3.2.1:

$$R_{corr} = W_{norm} = \frac{\overline{W}}{\overline{W} + W_0}, \tag{35}$$

where $\overline{W}$ is the mean local maneuver-magnitude margin and W_0 serves as the reference scale for the mean local maneuver-magnitude margin. A larger R_{corr} indicates that the selected maneuver remains feasible over a wider local range of velocity-increment scaling. The term favors solutions with a larger local margin.

The maneuver-count penalty is

$$R_{man} = -\frac{N_{man}}{K}, \tag{36}$$

where N_{man} is the number of stages containing a nonzero velocity increment and K is the number of serial decision stages. A stage for which $\|\Delta \boldsymbol{v}_j\| > 0$ counts as one maneuver, whereas a stage with three zero velocity-increment components does not.

An additional piecewise penalty is applied to the joint collision probability. Let $P_1 < ... < P_L$ denote the probability breakpoints and $c_1 < ... < c_L$ the corresponding penalty magnitudes. The penalty can be written in the general form

$$R_p(P_c) = \begin{cases} 0, & P_c \le P_1, \\ c_l, & P_l < P_c \le P_{l+1}, \\ c_L, & P_c > P_L, \end{cases} \tag{37}$$

where P_c is the joint collision probability. The penalty is zero below the first breakpoint and increases as P_c crosses successive risk levels. This mechanism gives collision risk additional priority beyond the contribution of P_c to the raw environment reward.

The coefficient ordering reflects the priorities of the maneuver problem. These coefficients are empirical and may need to change for a different mission. In this paper, their effects are tested through matched ablations in Section 4.4.

To provide feedback before the terminal stage, the environment component is distributed across the serial transitions. Let

$$C_j = \lambda_{\text{env}} \tanh\left(\frac{r_{\text{env},j}}{\kappa_{\text{env}}}\right) \tag{38}$$

denote the weighted environment component evaluated from the partial maneuver sequence after stage j. For a nonterminal stage,

$$r_j = C_j - C_{j-1} - \frac{\lambda_{\text{man}}}{K} I_j, \tag{39}$$

where I_j equals one if stage j contains a nonzero maneuver and zero otherwise. At the final stage, the remaining reward is assigned as

$$r_K = R_{\text{final}} - \sum_{j=1}^{K-1} r_j. \tag{40}$$

A discount factor γ occurs in the temporal-difference target used by PACC-QMIX. Consequently, the discounted training return is

$$G = \sum_{j=1}^{K} \gamma^{j-1} r_j \tag{41}$$

G is not generally equal to R_{final}. The decomposition supplies intermediate feedback and preserves the undiscounted sum of the stage rewards.

3.3. PACC-QMIX Architecture and Training

PACC-QMIX learns the joint value of the action components after the state, action mapping, local maneuver-magnitude margin, and reward have been defined. Each utility network evaluates one control component, and the state-conditioned mixer combines the selected local values. The agents therefore correspond to action components rather than maneuver stages. Temporal dependence between stages is represented by the serial state transitions and the temporal-difference target.

Training uses online utility networks, an online mixer, corresponding target networks, and an experience-replay buffer. The online networks select actions and receive gradient updates. The target networks construct the temporal-difference targets, while replay reduces the temporal correlation between sampled transitions. Figure 1 shows this architecture.

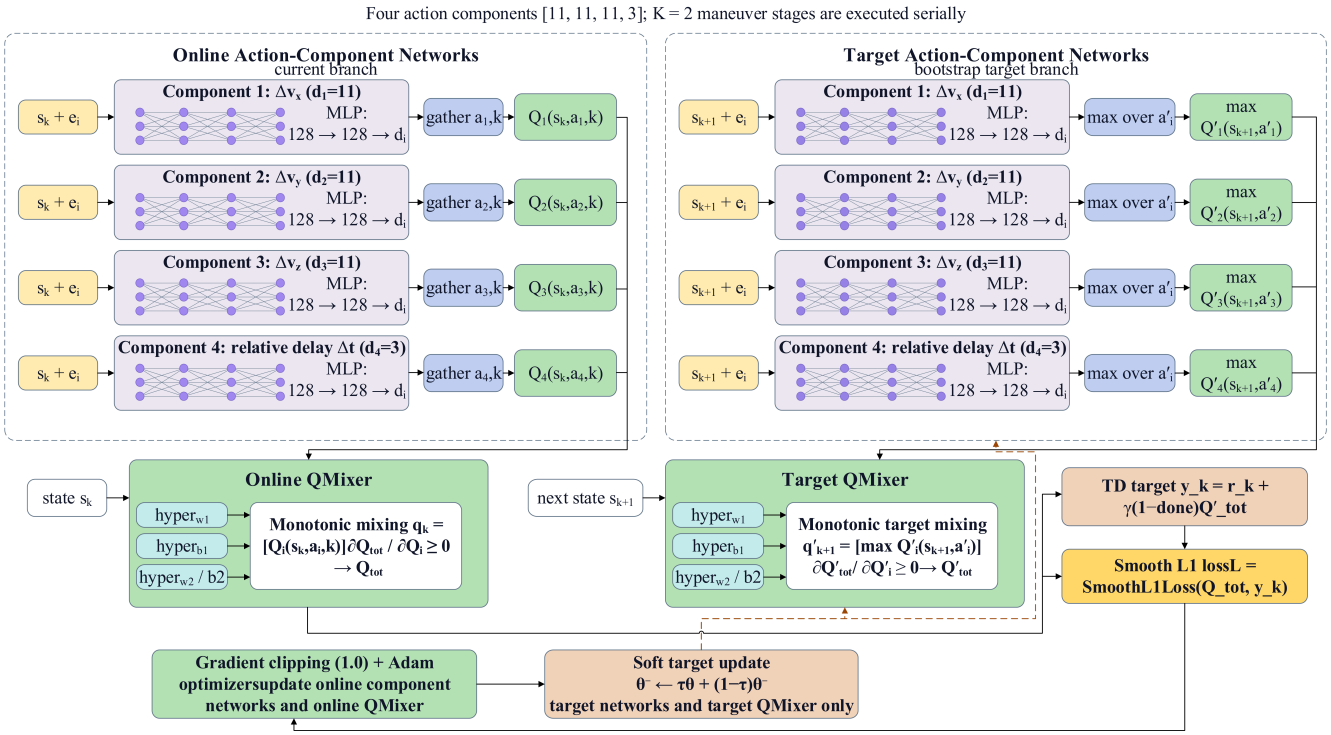


Figure 1. PACC-QMIX training architecture with online utility and mixing networks, target networks, and experience replay.

For action-component agent i at serial stage j, the local action value associated with the selected action is

$$Q_i^{(j)} = Q_i\left(\boldsymbol{o}_{i,j}, a_{i.j}; \boldsymbol{\theta}_i\right),\ i \in \{x, y, z, \tau\}. \tag{42}$$

where $i \in x,y,z,\tau$ identifies the action-component agent, and $j \in 1, \ldots, K$ denotes the serial maneuver stage. The vector $\boldsymbol{o}_{i,j}$ is the input to agent i at stage j, $a_{i,j}$ is the discrete action selected by agent i, and $\boldsymbol{\theta}_i$ denotes the parameters of agent i's utility network. The scalar $Q_i^{(j)}$ is the local action value assigned to $a_{i,j}$ under the input $\boldsymbol{o}_{i,j}$.

At stage j, all agents share the orbital state and use identities to distinguish the control dimensions ΔV_x, ΔV_y, ΔV_z, and waiting time τ. Stage progress is already included in the global state. The input to agent i at stage j is

$$\boldsymbol{o}_{i,j} = [\boldsymbol{s}_j, \boldsymbol{e}_i]. \tag{43}$$

where $\boldsymbol{s}_j$ is the global state at stage j, and $\boldsymbol{e}_i$ is the identity encoding of agent i. The identity allows shared-state agents to learn utilities specific to their control dimensions.

At stage j, each utility network outputs local action values for the discrete action space assigned to the corresponding agent.

$$\boldsymbol{Q}_i^{(j)} = \left[Q_i(\boldsymbol{o}_{i,j}, a_i = 0), Q_i(\boldsymbol{o}_{i,j}, a_i = 1), \ldots, Q_i(\boldsymbol{o}_{i,j}, a_i = |\mathcal{A}_i| - 1)\right], \tag{44}$$

where $\mathcal{A}_i$ is the discrete action space of agent i, and $|\mathcal{A}_i|$ gives the size of that action space. Action-space sizes may differ between components.

Each agent uses an ε-greedy policy to balance exploration and exploitation:

$$a_{i,j} = \begin{cases} \text{Random}(\mathcal{A}_i), & \textit{with probability } \varepsilon, \\ \underset{(a_i \in \mathcal{A}_i)}{argmax} Q_i(o_{i,j}, a_i), & \textit{with probability } 1 - \varepsilon. \end{cases} \tag{45}$$

The action Random($\mathcal{A}_i$) is sampled uniformly from the discrete space of component i, and ε is the exploration rate. A large initial ε promotes diverse combinations. As ε decays, action selection relies increasingly on the learned local values.

The four selected components form the joint discrete action $\boldsymbol{a}_j$ defined in Equation (23).

The action mapper converts the joint discrete action $\boldsymbol{a}_j$ into the continuous stage-control vector $\boldsymbol{u}_j$ defined in Equation (11).

$$\boldsymbol{u}_j = \mathcal{M}(\boldsymbol{a}_j) = \left[\Delta v_{x,j}, \Delta v_{y,j}, \Delta v_{z,j}, \tau_j\right]^{\mathrm{T}}. \tag{46}$$

The environment propagates for the relative waiting time, executes the velocity increment, and returns stage reward r_j and next state $\boldsymbol{s}_{j+1}$. Repeating this process produces the complete multistage maneuver sequence.

The mixing network combines the selected local action values into a stage-level global action value:

$$Q_{\mathrm{tot}}^{(j)} = f_{\mathrm{mix}}\left(Q_x^{(j)}, Q_y^{(j)}, Q_z^{(j)}, Q_\tau^{(j)}; \boldsymbol{s}_j\right). \tag{47}$$

Here, f_{mix} combines the four selected local action values into the global action value $Q_{tot}{}^{(j)}$, conditioned on the global state $\boldsymbol{s}_j$.

Following the original QMIX formulation, PACC-QMIX imposes a monotonicity constraint so that the globally greedy action can be decomposed into locally greedy component actions:

$$\frac{\partial Q_{\mathrm{tot}}^{(j)}}{\partial Q_i^{(j)}} \geq 0, \quad i \in \{x, y, z, \tau\}. \tag{48}$$

The constraint makes global value $Q_{tot}{}^{(j)}$ nondecreasing in every local value $Q_i{}^{(j)}$. Combining the locally greedy component actions therefore maximizes the stage-level global value.

The network contains action-component utility networks, an online mixer, target utility networks, a target mixer, and an update module. Utilities provide local values, and the online mixer produces global value Q for the selected joint action. The target networks construct stable temporal-difference targets Q_{tot}. Figure 2 shows the network and update relationships.

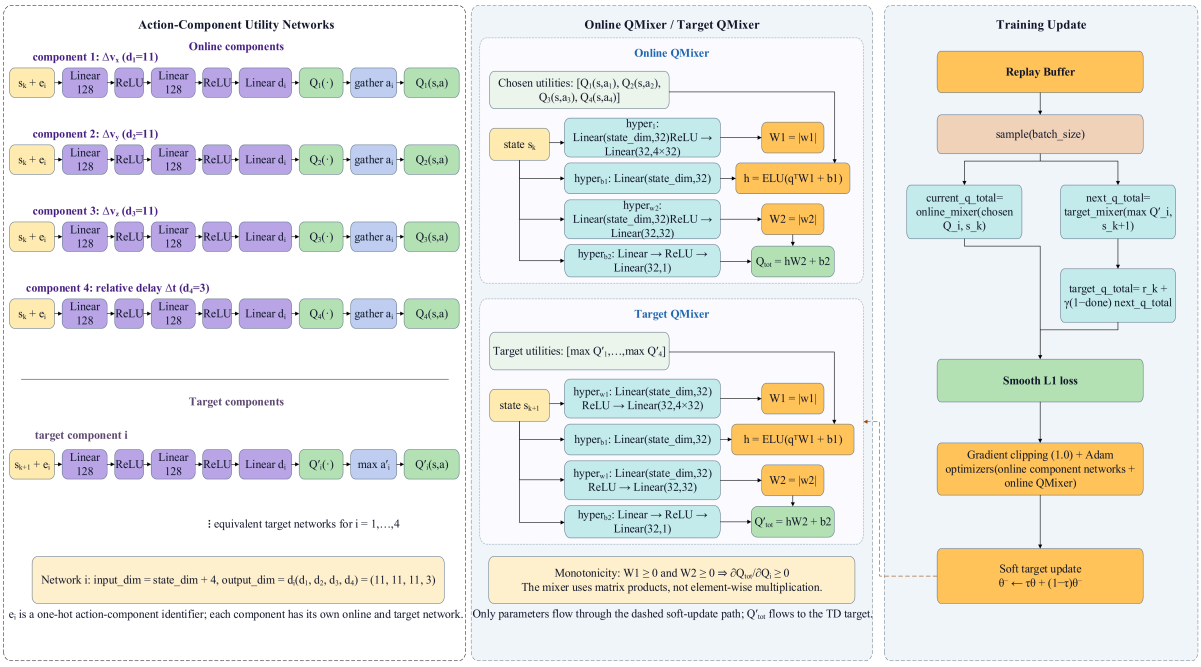


Figure 2. PACC-QMIX utility networks, hypernetwork-conditioned mixing networks, and parameter-update pathway.

As shown in Figure 2, the global state conditions the weights and biases of the online and target mixers. The state input includes temporal progress, fuel, collision risk, satellite

and debris states, and orbital deviation. Absolute-value transformations are applied to the generated mixing weights, but not to the biases, ensuring that the weights remain non-negative and satisfy the monotonicity condition in Equation (48).

Each serial stage contributes one replay transition:

$$\left(\boldsymbol{s}_j, \boldsymbol{a}_j, r_j, \boldsymbol{s}_{j+1}, d_j\right). \tag{49}$$

where $\boldsymbol{s}_j$ is the current state, $\boldsymbol{a}_j$ is the joint discrete action, r_j is the stage reward, $\boldsymbol{s}_{j+1}$ is the next state, and d_j indicates termination.

For a replay minibatch and next state $\boldsymbol{s}_{j+1}$, the target utilities and mixer calculate the target global value. The temporal-difference target is

$$y_j = r_j + \gamma(1 - d_j)\max_{a'} Q_{\text{tot}}^{-}\left(\boldsymbol{s}_{j+1}, \boldsymbol{a}'\right). \tag{50}$$

where γ is the discount factor, $Q_{tot}{}^{-}$ is the global action value produced by the target utility and mixing networks, and $\boldsymbol{a}'$ is a candidate joint action at the next state $\boldsymbol{s}_{j+1}$. The binary variable d_j equals one for a terminal transition and zero otherwise. The factor $1 - d_j$ therefore removes the future-return term at the terminal stage. Under the monotonic mixing constraint, each target utility network can select the locally greedy action before the selected local values are combined by the target mixer.

The online global action value is

$$Q_{\text{tot}}^{(j)} = Q_{\text{tot}}\left(\boldsymbol{s}_j, \boldsymbol{a}_j; \boldsymbol{\theta}\right). \tag{51}$$

where $\boldsymbol{\theta}$ denotes all online utility, mixing, and hypernetwork parameters. A smooth L1 loss is applied to the difference between target y and estimate $Q_{tot}(\boldsymbol{s}, \boldsymbol{a}; \boldsymbol{\theta})$:

$$L(\boldsymbol{\theta}) = \mathbb{E}_{(\boldsymbol{s}_j, \boldsymbol{a}_j, r_j, \boldsymbol{s}_{j+1}, d_j) \sim \mathcal{D}}\left[\text{SmoothL1}\left(Q_{\text{tot}}(\boldsymbol{s}_j, \boldsymbol{a}_j; \boldsymbol{\theta}) - y_j\right)\right]. \tag{52}$$

where $L(\boldsymbol{\theta})$ is the training loss, and $\boldsymbol{\theta}$ denotes all trainable parameters of the online utility networks, hypernetworks, and mixing network. The vectors $\boldsymbol{s}_j$ and $\boldsymbol{a}_j$ are the global state and joint discrete action at stage j, respectively. The scalar $Q_{tot}(\boldsymbol{s}_j, \boldsymbol{a}_j; \boldsymbol{\theta})$ is the global action-value estimate produced by the online networks, and y_j is the temporal-difference target defined in Equation (50). The expectation is approximated by the mean over a minibatch sampled from the replay buffer. Smooth L1 is a piecewise loss applied to the temporal-difference residual between Q_{tot} and y_j. With the implemented transition parameter $\beta = 1$, the loss is quadratic when the absolute residual is below 1 and linear in the residual magnitude otherwise. This behavior preserves sensitivity to small estimation errors while reducing the influence of large temporal-difference errors compared with mean-squared-error loss.

Gradient descent updates the online utility and mixing networks. Target parameters do not receive gradients and instead track the online parameters through a soft update:

$$\boldsymbol{\theta}^{-} \leftarrow \tau_{\text{soft}}\boldsymbol{\theta} + (1 - \tau_{\text{soft}})\boldsymbol{\theta}^{-}, \ (0 < \tau_{\text{soft}} < 1) \tag{53}$$

where $\boldsymbol{\theta}^{-}$ and $\boldsymbol{\theta}$ denote the target and online parameters, respectively, and τ_{soft} is the soft-update coefficient. A small τ_{soft} produces a slowly varying target network, which reduces rapid changes in the temporal-difference target.

The complete training and inference procedure of PACC-QMIX is summarized in Algorithm 1.

Algorithm 1. Training and inference procedure of PACC-QMIX.

Input: conjunction environment E, number of training episodes N_{ep}, maximum number of maneuver stages K, discount factor γ, target-network soft-update coefficient τ_{soft}, exploration rate ε, replay-buffer capacity C, and minibatch size B.
Output: trained PACC-QMIX policy and serial maneuver-action table.

Initialize the online utility networks, online mixing network, target utility networks, target mixing network, and replay buffer D.
Copy the online-network parameters to the corresponding target networks.
for episode = 1, ..., N_{ep} **do**
    Reset the conjunction environment and obtain the initial global state $\boldsymbol{s}_1$.
    for maneuver stage $j = 1, ..., K$ do
        Construct the global state $\boldsymbol{s}_j$ from temporal, fuel, risk, orbital-state, and orbital-deviation information.
        for action component i in x,y,z,τ do
            Evaluate $Q_i(\boldsymbol{o}_{i,j}, a_i, \boldsymbol{\theta}_i)$, and select $a_{i,j}$ using the ε-greedy policy.
        end for
        Assemble the joint discrete action: $\boldsymbol{a}_j = (a_{x,j}, a_{y,j}, a_{z,j}, a_{\tau,j})$. Map the joint discrete action to the continuous stage control: $\boldsymbol{u}_j = [\Delta v_{x,j}, \Delta v_{y,j}, \Delta v_{z,j}, \tau_j]^T$
        Propagate for τ_j, execute the selected nonzero velocity increment, and obtain s_{j+1} and $r_{env,j}$. Set $d_j = \mathbb{I}(j = K)$.
        Compute r_j from the change in the environment-return component minus the stage maneuver penalty. At $j = K$, replace r_j with $R_{\text{final}} - \sum_{l=1}^{j-1} r_l$.
        Store $(\boldsymbol{s}_j, \boldsymbol{a}_j, r_j, \boldsymbol{s}_{j+1}, d_j)$ in D.
        if $|D| \geq B$ **then**
            Sample a minibatch from D, and compute the temporal-difference target y_j and online global action-value estimate $Q_{tot}(\boldsymbol{s}_j, \boldsymbol{a}_j; \boldsymbol{\theta})$.
            Minimize the smooth L1 loss and update the online utility networks, hypernetworks, and mixing network.
            Softly update the target utility networks and target mixing network using τ_{soft}.
        end if
        Update the current state: $\boldsymbol{s}_j \leftarrow \boldsymbol{s}_{j+1}$.
    end for
    Decay ε after the episode.
end for
During inference, set $\varepsilon = 0$, execute the maneuver stages serially, and record the resulting maneuver-action table.

At each maneuver stage, four action-component agents select three velocity-increment components and a relative waiting time. A state-conditioned monotonic mixer combines their local action values, and orbital propagation updates the environment state before the next decision. The policy thus produces serial maneuver sequences while accounting for collision risk, fuel consumption, orbital deviation, and the local maneuver-magnitude margin.

Training uses the orbital-corridor reward and shaped stage rewards with experience replay, temporal-difference learning, and softly updated target networks. Section 4 compares PACC-QMIX with baseline methods, examines planning-horizon and threshold sensitivity, tests reward ablations, and replays stored maneuvers in an orbital environment including J_2 perturbation and atmospheric drag.

4. Results

4.1. Experimental Setup

Each scenario contained one protected satellite and five debris objects. Only the protected satellite could maneuver. The study comprised four experiments. The first compared

PACC-QMIX with CE, Double DQN, and MAPPO under 1-day and 7-day planning horizons. The second examined sensitivity to the planning horizon and collision-probability threshold. The third evaluated reward ablation and coefficient sensitivity in Scenario 5 using five independent PACC-QMIX training seeds. The fourth tested stored 7-day action sequences in an orbital environment including J_2 perturbation and atmospheric drag, without retraining or new action selection. The first three experiments used a two-body-only orbital environment. The reported metrics included final shaped reward, collision probability, threshold exceedance, fuel consumption, and orbital deviation.

In both the first and second experiments, 10 independent scenarios were used for experimentation and analysis. A separate PACC-QMIX policy was trained for each scenario and tested configuration. These results therefore assess scenario-specific planning only, and generalization to unseen scenarios was not evaluated.

Table 1 summarizes the environment, action-space, and physical-constraint settings used in the experiments.

Collision risk contributes to both the environment reward and the separate piecewise penalty. The corridor term has a smaller weight and rewards a wider local maneuver-magnitude margin. The maneuver-count penalty has the smallest weight and discourages unnecessary maneuvers. The weights were chosen for the tested setup, and their effects were examined in the ablation and sensitivity analyses in Section 4.4. Table 2 lists the baseline reward parameters.

Table 2. Numerical reward parameters used in the experiments.

Category	Parameter	Value
Environment reward	Normalization scale κ_{env}	5
Environment reward	λ_{env}	0.8333
Orbital-corridor reward	Reference margin W_0	1
Orbital-corridor reward	λ_{corr}	0.0521
Maneuver-count term	Definition	$R_{man} = -N_{man}/K$
Maneuver-count term	λ_{man}	0.0104
Collision penalty	λ_p	0.1042
Collision penalty	Probability breakpoints	10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}
Collision penalty	Penalty magnitudes	0, 1, 2, 5, 8
Stage-reward decomposition	Initial reference component C_0	0
Final shaped reward	Clipping interval	$[-10, 10]$

Table 3 summarizes the network architecture, training hyperparameters, and evaluation settings. These settings were held constant across scenarios and planning horizons unless stated otherwise. Section 4.2 compares the baseline methods under the 1-day and 7-day planning horizons.

Table 3. PACC-QMIX architecture and training hyperparameters.

Category	Parameter	Value
Component Q networks	Network type	Four independent MLPs; parameters not shared
Component Q networks	Hidden layers	Two layers of 128 units

Table 3. *Cont.*

Category	Parameter	Value
Component Q networks	Activation function	ReLU
Mixing network	Hidden dimension	32
Optimization	Optimizer	Adam
Optimization	Learning rate	10^{-4}
Experience replay	Buffer capacity	10,000 transitions
Experience replay	Batch size	32
TD learning	Discount factor γ	0.95
TD learning	Soft-update coefficient τ	0.01
Exploration	Initial/minimum ε	1.0/0.01
Training budget	Episodes	3000 per scenario

4.2. Comparison with Baseline Methods in Short-Horizon and Pre-Collision-Avoidance Scenarios

The 1-day planning horizon represents short-horizon collision avoidance, for which maneuver selection is restricted to the final day before the nominal conjunction. The 7-day horizon represents pre-collision-avoidance, in which planning begins earlier and the two serial maneuver stages can be distributed over a longer interval. These labels describe the two experimental configurations considered here rather than universal operational categories.

PACC-QMIX, CE, Double DQN, and MAPPO were evaluated on the same orbital scenarios in the same two-body-only orbital environment, collision-probability calculation, and final shaped-reward formulation. A collision-probability threshold of 10^{-4} was applied to both planning horizons. NASA identifies 10^{-4} as a widely accepted per-event mitigation threshold for conjunction assessment and recommends mitigation above this value [5].

PACC-QMIX, Double DQN, and MAPPO were trained for 3000 episodes, while CE was run for 3000 optimization iterations. All values reported in Table 4 are final shaped rewards calculated using the same reward formulation. Higher, or less-negative, values indicate better aggregate performance with respect to collision risk, maneuver cost, orbital deviation, and the local maneuver-magnitude margin.

Table 4. Scenario-level final shaped rewards obtained by PACC-QMIX, CE, Double DQN, and MAPPO under the 7-day and 1-day planning configurations with $P_{thr} = 10^{-4}$. The final row reports the mean and standard deviation (SD) across the ten scenarios.

Experiment	PACC-QMIX, 7 d	CE, 7 d	Double DQN, 7 d	MAPPO, 7 d	PACC-QMIX, 1 d	CE, 1 d	Double DQN, 1 d	MAPPO, 1 d
1	−0.0391	−0.5554	−0.0296	−0.022	−0.1381	−0.2098	−1.3646	−0.1523
2	−0.0490	−0.8370	−0.0270	−0.039	−0.091	−0.0934	−0.098	−0.1243
3	−0.0187	−0.8072	−1.3646	−0.0327	−0.1016	−0.8361	−0.0764	−0.0911
4	−0.0290	−0.8437	−1.6771	−0.0389	−0.0723	−0.3301	−1.0359	−0.0697
5	−0.0235	−0.8437	−0.1063	−0.0692	−0.0870	−0.8370	−1.6771	−0.0798
6	−0.0615	−0.8438	−1.3646	−0.0331	−0.0604	−0.2672	−1.6771	−0.0941
7	−0.0245	−0.8433	−0.0339	−0.0333	−0.0433	−0.8191	−1.6771	−0.0702
8	−0.0262	−0.1782	−0.0841	−0.0366	−0.1102	−0.8128	−1.3646	−0.0996
9	−0.0296	−0.0766	−0.0676	−0.0103	−0.0385	−0.8434	−1.3646	−0.0512
10	−0.0290	−0.8371	−0.0534	−0.0427	−0.0977	−0.8358	−0.0987	−0.0962
Mean ± SD	−0.0330 ± 0.0132	−0.6666 ± 0.2986	−0.4808 ± 0.6875	−0.0358 ± 0.0151	−0.0840 ± 0.0309	−0.5885 ± 0.3182	−1.0434 ± 0.6860	−0.0929 ± 0.0290

At the common 10^{-4} threshold, PACC-QMIX achieved the highest mean final shaped reward in the 7-day configuration (−0.0330), followed by MAPPO, CE, and Double DQN. The mean-reward differences between PACC-QMIX and MAPPO, Double DQN, and CE

were 0.0028, 0.4478, and 0.6336. Collision probabilities remained below the threshold for PACC-QMIX, CE and MAPPO in all ten scenarios. Double DQN exceeded the threshold in scenarios 3, 4 and 6. At 1 day, the mean final shaped rewards were −0.0840 for PACC-QMIX, −0.0929 for MAPPO, −0.5885 for CE, and −1.0434 for Double DQN. PACC-QMIX exceeded the MAPPO, CE, and Double DQN means by 0.0089, 0.5045, and 0.9594 reward units, respectively. Collision probability did not exceed the 10^{-4} threshold in any of the ten PACC-QMIX, MAPPO, or CE scenarios. For Double DQN, collision probability did not exceed the threshold only in scenarios 2, 3 and 10. Figure 3 illustrates these scenario-level reward comparisons.

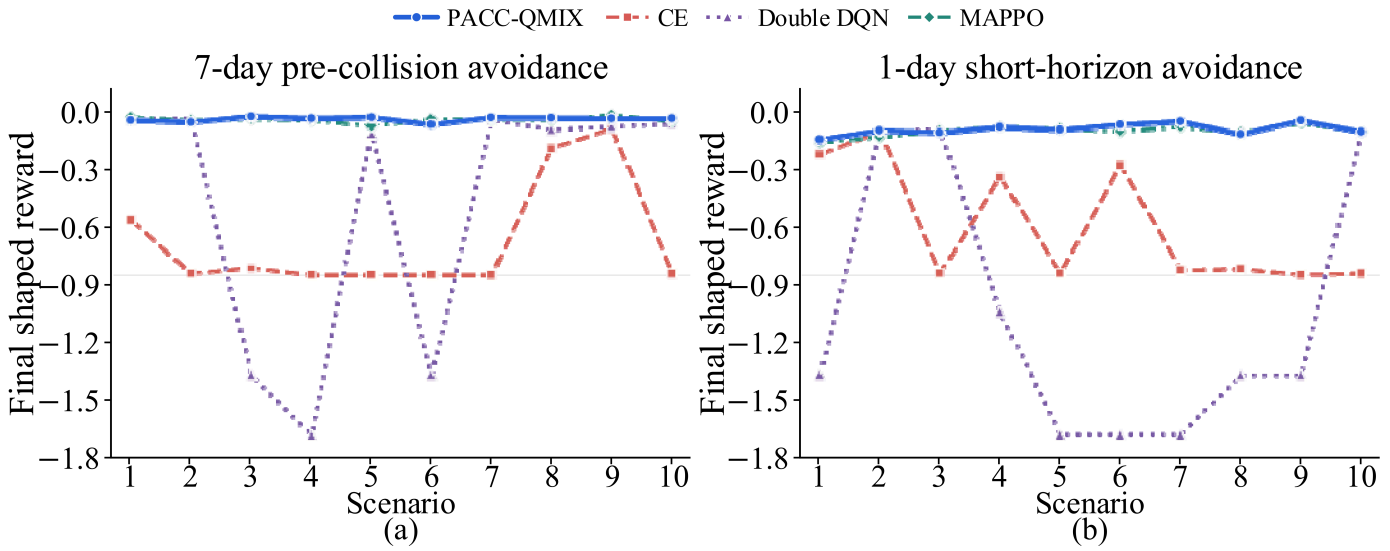


Figure 3. Scenario-level final shaped rewards of PACC-QMIX, CE, Double DQN, and MAPPO at $P_{thr} = 10^{-4}$: (**a**) the 7-day pre-collision-avoidance configuration and (**b**) the 1-day short-horizon collision-avoidance configuration.

Double DQN showed the greatest variation across scenarios, with a reward SD of 0.6875 at 7 days and 0.6860 at 1-day. MAPPO's mean rewards remained close to those of PACC-QMIX, whereas the gaps to CE and Double DQN were larger.

In these scenarios, the 7-day comparison with CE showed a smaller environment penalty for PACC-QMIX, reflected by the less negative environment contribution in Table 5. This means that PACC-QMIX had lower collision probabilities, lower fuel consumption, and smaller aggregate orbital-element-deviation penalties. The lower maneuver-effort penalty corresponded to a smaller cumulative velocity increment.

Table 5. Reward decomposition for PACC-QMIX and CE across the ten 7-day scenarios in Table 4. Values are mean ± sample standard deviation across scenarios.

Reward Quantity	PACC-QMIX, Mean ± SD	CE, Mean ± SD
Environment contribution	−0.024735 ± 0.013179	−0.660727 ± 0.289093
Orbital-corridor contribution	+0.002124 ± 0.000450	+0.004546 ± 0.010723
Maneuver-count contribution	−0.010417 ± 0.000000	−0.010417 ± 0.000000
Additional collision-probability penalty	0.000000 ± 0.000000	0.000000 ± 0.000000
Final shaped reward	−0.033027 ± 0.013174	−0.666597 ± 0.298559

Then, a run-to-run variation experiment was assessed in a 1-day scenario using five independent random seeds, numbered 0–4, for each method. PACC-QMIX was trained for 3000 episodes per seed and evaluated greedily every 100 episodes. CE was allowed up to 3000 search iterations, with early stopping available after iteration 200. Both methods used

the same scenario, collision-probability threshold, and final reward evaluator. Five final evaluations were averaged to produce one normalized reward per seed. Table 6 summarizes the seed-level rewards, collision probabilities, and reward statistics for both methods.

Table 6. Five-seed results for PACC-QMIX and early-stopped CE in the first 1-day scenario in Table 4. A collision probability of 0 is the numerical value returned by the evaluator.

Seed or Statistic	PACC-QMIX Reward	PACC-QMIX Collision Probability	CE Reward	CE Collision Probability
0	−0.1378	3.55×10^{-11}	−0.2046	8.51×10^{-6}
1	−0.1484	0	−0.2104	7.79×10^{-6}
2	−0.0937	1.76×10^{-6}	−0.2084	7.59×10^{-6}
3	−0.0986	1.27×10^{-7}	−0.2231	8.35×10^{-6}
4	−0.1501	0	−0.2086	7.59×10^{-6}
Mean ± sample SD	-0.1257 ± 0.0274	—	-0.2110 ± 0.0071	—
95% confidence interval for the mean reward	$[-0.1598, -0.0917]$	—	$[-0.2198, -0.2023]$	—

For each method, the 95% confidence interval for the mean reward across seeds was calculated as

$$\bar{x} \pm t_{1-\alpha/2,n-1} \frac{s}{\sqrt{n}}, \quad n = 5, \alpha = 0.05. \tag{54}$$

where $\bar{x}$ and s are the mean and sample SD of that method's five seed-level rewards.

PACC-QMIX achieved a mean normalized reward 0.0853 reward units higher than CE. The 95% confidence interval for the difference in mean normalized rewards, calculated using Welch's method, was [0.0517, 0.1189]. Welch's two-sided t-test yielded $t(4.53) = 6.74$ and $p = 0.0016$. An exact two-sided Mann–Whitney U test yielded $U = 25$ and $p = 0.0079$. Greedy-evaluation rewards generally stabilized during later training across the five-seed experiment, and all final policies from both methods remained below the 10^{-4} collision-probability threshold.

Note that on a platform equipped with an Intel Core Ultra 9 275HX CPU and 24 processing threads, offline training for 3000 episodes required 425.73 s for PACC-QMIX, 519.14 s for Double DQN and 871.39 s for MAPPO. One 3000-iteration CE search required 21,285.67 s (5.91 h).

4.3. Effects of Planning Horizon and Collision-Probability Threshold

Following the operational definitions introduced in Section 4.2, the 1-day horizon represented short-horizon collision avoidance, whereas the 7-day and 14-day horizons represented pre-collision-avoidance. The first comparison held the collision-probability threshold at 10^{-4} across all three horizons to evaluate the effect of earlier planning. A separate 14-day experiment used thresholds of 10^{-8}, 10^{-6}, and 10^{-4} to examine increasingly stringent risk requirements. Each configuration contained ten scenarios and used the same policy architecture and evaluation metrics.

Table 7 reports the final shaped reward values for the ten scenarios under the tested planning horizons and collision-probability thresholds. Table 8 reports the corresponding scenario-level fuel consumption.

Table 7. Scenario-level PACC-QMIX final shaped rewards under different planning horizons and collision-probability thresholds.

Experiment	14 d, 10^{-8}	14 d, 10^{-6}	14 d, 10^{-4}	7 d, 10^{-4}	1 d, 10^{-4}
1	−0.0420	−0.0206	−0.0243	−0.0391	−0.1381
2	−0.0473	−0.0187	−0.0421	−0.0490	−0.0910
3	−0.0200	−0.0322	−0.0340	−0.0187	−0.1016
4	−0.0249	−0.0375	−0.0173	−0.0290	−0.0723
5	−0.0193	−0.0259	−0.0142	−0.0235	−0.087
6	−0.0352	−0.0157	−0.0132	−0.0615	−0.0604
7	−0.0102	−0.0449	−0.1260	−0.0245	−0.0433
8	−0.0278	−0.0501	−0.0212	−0.0262	−0.1102
9	−0.0306	−0.0136	−0.0159	−0.0296	−0.0385
10	−0.0192	−0.0376	−0.0168	−0.0290	−0.0977
Mean	−0.0276	−0.0297	−0.0325	−0.0330	−0.0840

Table 8. Scenario-level PACC-QMIX fuel consumption (kg) under different planning horizons and collision-probability thresholds.

Experiment	14 d, 10^{-8}	14 d, 10^{-6}	14 d, 10^{-4}	7 d, 10^{-4}	1 d, 10^{-4}
1	0.0200	0.0190	0.0131	0.0127	0.0259
2	0.0185	0.0095	0.0225	0.0200	0.0254
3	0.0143	0.0173	0.0115	0.0255	0.0249
4	0.0174	0.0085	0.0177	0.0196	0.0180
5	0.0281	0.0218	0.0236	0.0274	0.0243
6	0.0228	0.0136	0.0060	0.0246	0.0233
7	0.0040	0.0094	0.0138	0.0243	0.0197
8	0.0152	0.0170	0.0149	0.0171	0.0233
9	0.0101	0.0101	0.0103	0.0210	0.0226
10	0.0151	0.0157	0.0220	0.0160	0.0181
Mean	0.0165	0.0142	0.0156	0.0208	0.0226

At the common 10^{-4} threshold, extending the planning horizon from 1 to 7 days raised the mean final shaped reward from −0.0840 to −0.0330, a difference of 0.0510 reward units. Mean fuel consumption decreased from 0.0226 to 0.0208 kg. No scenario exceeded the threshold in either configuration. The 14-day and 7-day configurations had very similar mean rewards (−0.0325 and −0.0330, respectively). The 14-day configuration had lower mean fuel consumption (0.0156 versus 0.0208 kg), but increased the mean reward by only 0.0005 reward units. The 7-day configuration with a 10^{-4} threshold was therefore selected for subsequent experiments as the most suitable overall balance among the settings tested in this study. Tightening the threshold from 10^{-4} to 10^{-6} or 10^{-8} within the 14-day horizon produced small changes in mean reward, no threshold exceedances, and no monotonic change in mean fuel consumption. The largest mean-reward difference among the tested settings remained the change from 1 to 7 days.

Figure 4 summarizes the scenario-level final shaped rewards across the tested planning horizons and collision-probability thresholds. Figure 5 illustrates one maneuver sequence under the 7-day planning horizon and a collision-probability threshold of 10^{-4}.

The original trajectory represents propagation without avoidance maneuvers, whereas the maneuvered trajectory follows the action sequence generated by PACC-QMIX. The colored markers and trajectories distinguish the protected satellite from the five debris objects. The translucent green region indicates the local magnitude-scaling interval used to calculate the orbital-corridor reward. Varying maneuver magnitudes while holding their directions fixed defines this interval, which quantifies local tolerance to maneuver-

magnitude variations. M1 and M2 identify the two serial maneuver stages and show how the maneuver sequence changes the propagated trajectory before conjunction.

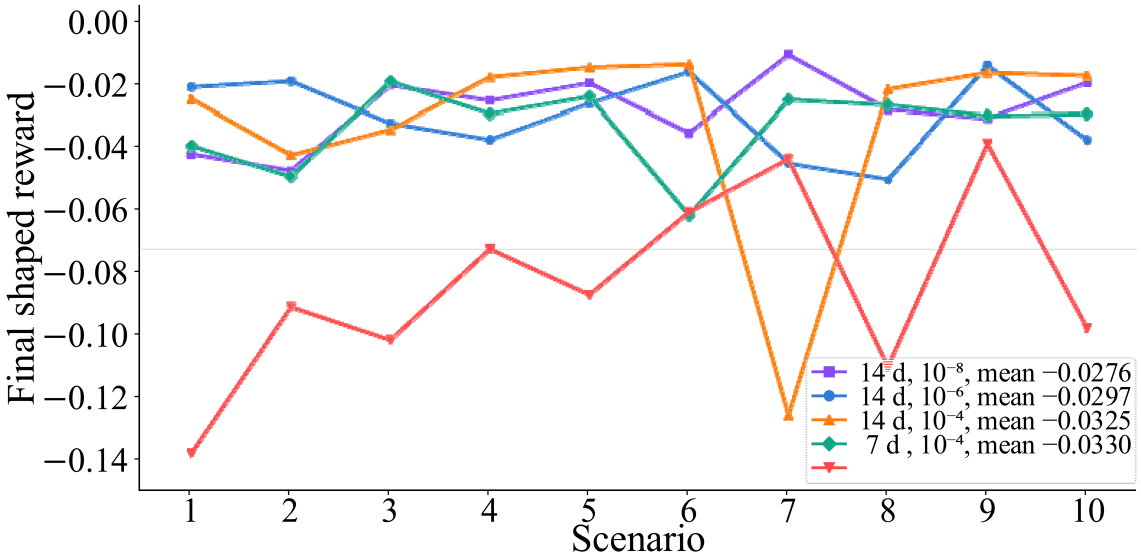


Figure 4. Scenario-level PACC-QMIX final shaped rewards under the tested planning horizons and collision-probability thresholds.

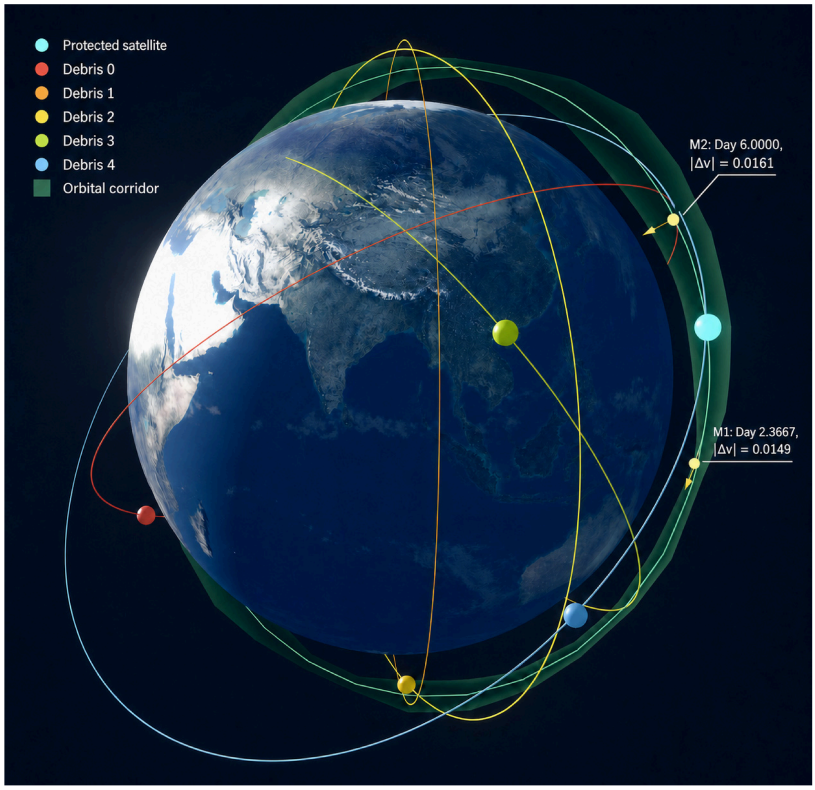


Figure 5. Representative multi-object pre-collision maneuver generated by PACC-QMIX under the 7-day planning horizon and a collision-probability threshold of 10^{-4}. The visualization shows the protected satellite, the debris objects, the original and maneuvered trajectories, the orbital-corridor scaling region, and stages M1 and M2.

Figure 6 shows a representative PACC-QMIX training run under the 1-day planning horizon and a collision-probability threshold of 10^{-4}.

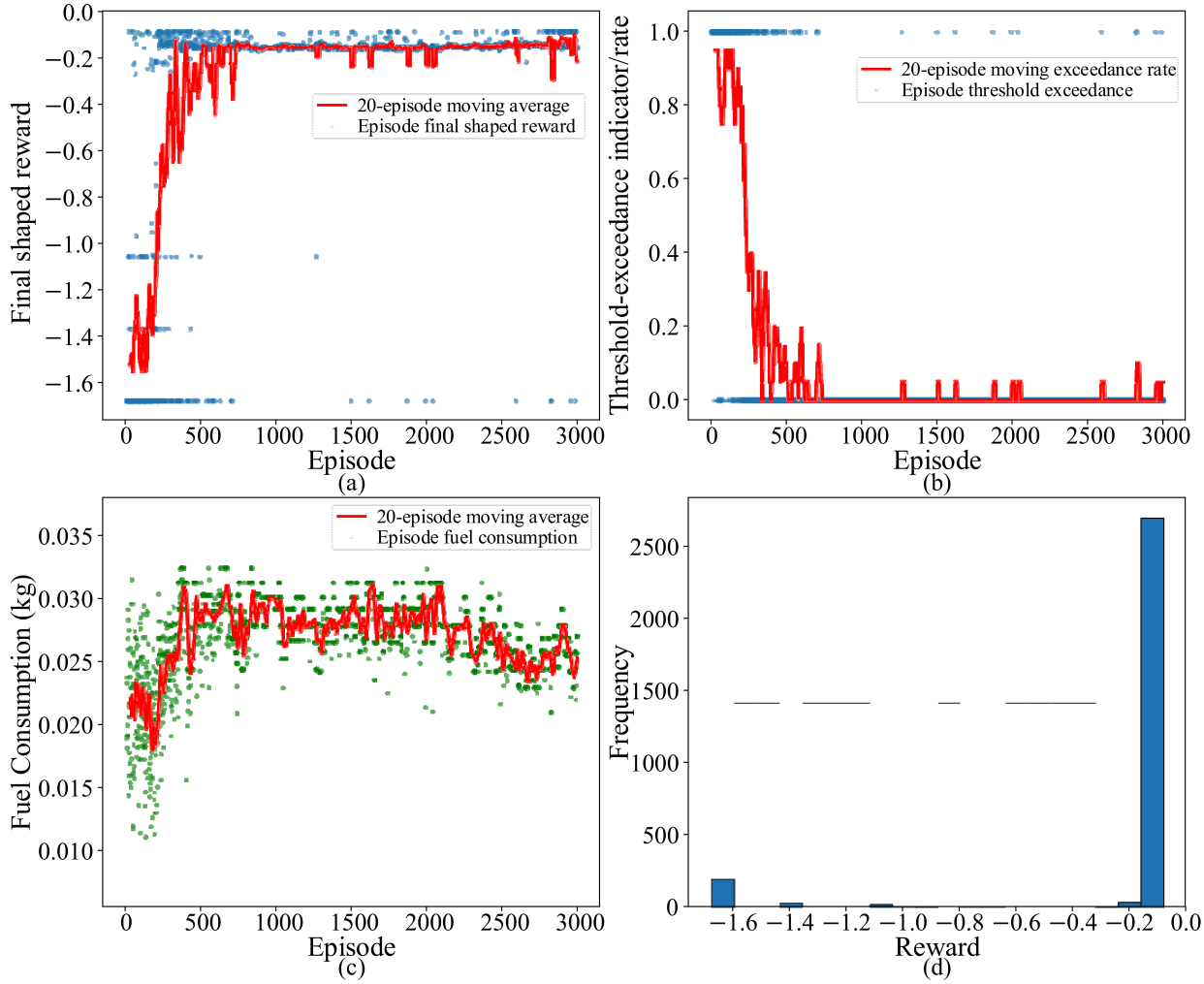


Figure 6. PACC-QMIX training dynamics over 3000 episodes under the 1-day planning horizon and a collision-probability threshold of 10^{-4}. (**a**) Episode final shaped reward and the corresponding 20-episode moving average. (**b**) Threshold-exceedance indicator and its 20-episode moving rate. (**c**) Episode fuel consumption and its 20-episode moving average. (**d**) Distribution of episode final shaped rewards.

Figure 6 follows one 3000-episode training run. Figure 6a reports the final shaped reward defined in Section 3.2.2, with values closer to zero indicating better performance on this scale. Figure 6b presents threshold exceedances, and Figure 6c reports fuel consumption in kilograms under the convention defined in Section 2.4.

During the first 500 episodes, the 20-episode moving-average reward rose from about −1.5 to −0.2, while the moving exceedance rate fell from near one to almost zero. The moving-average reward stayed near −0.15 for much of the remaining run, despite occasional low-reward episodes and threshold exceedances. Mean fuel consumption rose early to about 0.03 kg, then fluctuated around 0.025–0.030 kg before declining modestly near the end.

4.4. Reward Ablation Experiments and Coefficient Sensitivity Analysis Experiments

To examine the orbital-corridor and maneuver-count terms, five reward configurations were tested in Scenario 5 under the 1-day planning horizon. The full reward was compared with variants that removed either term or tripled the corresponding coefficient. Table 9 summarizes the orbital-corridor margins, reference shaped rewards, and nominally feasible seed counts for the five configurations.

Table 9. Reward ablation in Scenario 5. Values are the mean ± sample standard deviation across five independently trained seeds. The orbital-corridor margin is dimensionless. Reference shaped rewards were calculated using the baseline coefficients for every configuration.

Reward Configuration	Normalized Orbital-Corridor Margin	Reference Shaped Reward	Nominally Feasible Seeds
Full reward	0.047728 ± 0.003821	−0.077979 ± 0.022348	5/5
No orbital-corridor term	0.047107 ± 0.005972	−0.074332 ± 0.033075	5/5
Threefold corridor weight	0.041149 ± 0.005719	−0.110534 ± 0.034790	5/5
No maneuver-count term	0.047428 ± 0.001322	−0.079003 ± 0.014284	5/5
Threefold maneuver-count weight	0.045966 ± 0.002175	−0.077859 ± 0.016536	5/5

Scenario 5 used the baseline coefficients listed in Table 2. None of the 25 final policies across the five ablation configurations exceeded the 10^{-4} collision-probability threshold. The full reward yielded a slightly higher mean orbital-corridor margin than removing the corridor term (0.047728 versus 0.047107). Tripling the corridor weight reduced the mean margin to 0.041149. Removing the corridor term gave a slightly higher mean reference shaped reward than the full reward (−0.074332 versus −0.077979). Removing the maneuver-count term slightly reduced the mean corridor margin (0.047428) and reference shaped reward (−0.079003 versus −0.077979). The full reward had the highest observed mean corridor margin among the five configurations.

Then, the coefficient sensitivity analysis scaled one training coefficient at a time by 0.5, 2, or 3. All final policies were evaluated using the baseline coefficients, and the results are shown in Table 10.

Table 10. Coefficient sensitivity results in Scenario 5. Values are the mean ± sample SD of the reference shaped reward across five independent training seeds. Only the coefficient named in each row was varied.

Coefficient Varied	×0.5	Baseline	×2	×3
Environment	−0.113006 ± 0.031662	−0.077979 ± 0.022348	−0.086614 ± 0.047380	−0.060995 ± 0.003610
Orbital corridor	−0.087988 ± 0.034315	−0.077979 ± 0.022348	−0.085274 ± 0.032630	−0.110534 ± 0.034790
Maneuver count	−0.096765 ± 0.046093	−0.077979 ± 0.022348	−0.089234 ± 0.033264	−0.077859 ± 0.016536
Collision penalty	−0.086488 ± 0.024821	−0.077979 ± 0.022348	−0.097096 ± 0.033124	−0.107186 ± 0.031530

The baseline corridor weight gave the highest mean among corridor settings. Tripling the environment weight yielded the highest mean overall (−0.060995), whereas maneuver-count scaling showed no consistent improvement. Collision-penalty scaling also showed no monotonic trend, with both increased settings yielding lower means than baseline (Table 10). These results indicate that the baseline corridor weight and increased environment weight were favorable among the tested settings.

4.5. Fixed-Maneuver Test Experiment

For this experiment, the scenario-specific PACC-QMIX policies were trained in a two-body-only orbital environment and tested in an orbital environment including J_2 perturbation and atmospheric drag. A no-maneuver trajectory from the same initial Cartesian states provided a paired baseline for each scenario.

The test used an orbital environment including central gravity, the EIGEN-6S degree-two, order-zero J_2 term, and NRLMSISE-00 atmospheric drag driven by historical CSSI space-weather inputs. The protected satellite had a mass of 1000 kg, an area of 10 m^2, and a drag coefficient of 2.2. Each debris object had a mass of 10 kg, an area of 0.2 m^2, and the same drag coefficient. Solar radiation pressure, third-body perturbation, and relativity were excluded.

The ten scenarios were selected because the no-maneuver collision probability exceeded 10^{-5} under both two-body propagation and propagation incorporating J_2 perturbation and atmospheric drag. This selection provided a hazardous no-maneuver baseline in both orbital environments.

For each debris object, collision probability was evaluated at that object's closest-approach epoch during the test. Table 11 reports the largest object-level probability and the smallest closest-approach distance across the five debris objects. These two metrics can come from different objects. The probability threshold was 10^{-5}, one order of magnitude stricter than the 10^{-4} reference used in the planning-horizon study.

Table 11. Fixed-maneuver test experiment result. Probability is the maximum object-level collision probability, and distance is the minimum closest-approach distance across five debris objects.

Experiment	No Maneuver: Probability	No Maneuver: Minimum Distance (m)	Best Action: Probability	Best Action: Minimum Distance (m)
1	1.19×10^{-2}	25.7	0	17,310.00
2	1.24×10^{-2}	6.7	1.05×10^{-93}	6228.60
3	3.42×10^{-4}	22.3	1.95×10^{-38}	3819.40
4	2.73×10^{-5}	4.1	2.43×10^{-22}	2695.00
5	8.58×10^{-4}	3.5	1.24×10^{-48}	4370.40
6	1.05×10^{-2}	7.4	3.44×10^{-42}	4101.30
7	1.32×10^{-2}	7.2	8.98×10^{-30}	3401.80
8	3.07×10^{-3}	14.7	2.70×10^{-30}	3395.00
9	3.27×10^{-3}	30.5	7.09×10^{-56}	4736.50
10	7.30×10^{-4}	13.7	2.29×10^{-29}	3296.00

All ten no-maneuver trajectories exceeded the 10^{-5} threshold under both propagation models. When the stored best-action tables were tested under an orbital environment with J_2 perturbation and atmospheric drag, all ten scenarios remained below the object-level threshold, with no threshold exceedances. In this experiment, the no-maneuver minimum distances ranged from 3.5 to 30.5 m. After maneuvering, they ranged from 2695 to 17,310 m. The minimum distance increased in every paired scenario. These values describe modeled closest approaches. The collision probability displayed as zero for Scenario 1 is a numerical evaluator output and does not imply zero physical risk.

These results show that maneuver sequences obtained from policies trained in the two-body-only orbital environment retained threshold compliance in all ten selected tests incorporating J_2 perturbation and atmospheric drag, without retraining or new action selection.

5. Conclusions

PACC-QMIX models pre-collision-avoidance for one protected satellite and multiple debris objects as a sequence of maneuver decisions. At each stage, four action-component agents select three ECI velocity-increment components and a relative waiting time. A state-conditioned monotonic mixer combines their local values, while propagation in a two-body-only orbital environment updates the state for the next decision. The shaped reward includes a normalized environment reward, an orbital-corridor reward, a maneuver-count penalty, and a separate collision-probability penalty. The corridor term favors a larger local maneuver-magnitude margin.

At the 10^{-4} collision-probability threshold, PACC-QMIX achieved the highest mean final shaped reward among the four methods under both the 1-day and 7-day configurations. The PACC-QMIX mean reward improved from -0.0840 at 1 day to -0.0330 at 7 days, a difference of 0.0510 reward units. All PACC-QMIX scenarios remained below the threshold,

whereas Double DQN exceeded the threshold in several scenarios. In the ten-scenario 7-day reward decomposition, PACC-QMIX had a smaller environment penalty than CE. The maneuver-count contributions and additional collision-probability penalties were equal for both methods. MAPPO produced a similar 7-day mean reward of −0.0358. For 3000 episodes of offline training, MAPPO required 871.39 s versus 425.73 s for PACC-QMIX, a difference of 445.66 s.

At the same threshold, extending the planning horizon from 7 to 14 days raised the mean reward from −0.0330 to −0.0325. The gain was only 0.0005 reward units, although the 14-day configuration also had lower mean fuel consumption. Given the small reward gain and the potential operational burden of a longer planning window, the 7-day configuration was retained for subsequent analysis. The preferred planning horizon may vary with scenario-specific conjunction geometry and mission constraints.

Then, to assess the reward design, ablation and coefficient-sensitivity analyses were conducted in Scenario 5 under the 1-day planning horizon. Each tested reward configuration used five independently trained PACC-QMIX seeds. The full reward produced the highest observed mean normalized orbital-corridor margin of 0.047728 ± 0.003821. Removing the corridor term yielded 0.047107 ± 0.005972, indicating that the orbital-corridor reward did have an effect on the normalized orbital-corridor margin, although the effect was limited in this setting. In the sensitivity analysis, the baseline corridor weight gave the highest mean reference shaped reward among the tested corridor weights. Tripling the environment weight gave the highest observed mean across all tested settings, at −0.060995 versus −0.077979 for the baseline.

Finally, stored maneuver sequences from ten selected 7-day scenarios were tested with J_2 perturbation and atmospheric drag. The scenario-specific policies were trained under two-body-only orbital environment, and the test involved no retraining or new action selection. All ten no-maneuver trajectories exceeded the 10^{-5} object-level collision-probability threshold under both propagation models. All ten stored maneuver sequences remained below that threshold in the environment incorporating J_2 perturbation and atmospheric drag. The minimum closest-approach distance increased in every paired scenario, from 3.5–30.5 m without maneuvers to 2695–17,310 m with maneuvers. These results indicate that the learned maneuver policies retained their collision-avoidance effectiveness under the more realistic perturbed propagation model, despite being trained solely with two-body dynamics.

The action space is limited to discrete ECI velocity increments and relative waiting times over at most two stages. The model does not account for spacecraft-specific propulsion and attitude limits, finite burns, minimum impulse, command timelines, execution errors, or updated conjunction assessments. The effect of action discretization on solution quality also remains unquantified. Future work should evaluate these constraints with a spacecraft-specific maneuver model.

Author Contributions: Conceptualization, Z.S., X.P. and J.Z.; methodology, Z.S., R.D. and J.Z.; software, Z.S. and R.D.; validation, Z.S., R.D. and W.N.; formal analysis, Z.S., R.D. and J.Z.; investigation, Z.S., R.D. and W.N.; resources, W.T., X.P. and J.Z.; data curation, Z.S. and R.D.; writing—original draft preparation, Z.S. and J.Z.; writing—review and editing, R.D., W.T., W.N., X.P. and J.Z.; visualization, Z.S. and R.D.; supervision, W.T., X.P. and J.Z.; project administration, W.T., X.P. and J.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data underlying the results presented in this paper are not publicly available at this time but may be obtained from the authors upon reasonable request.

Acknowledgments: During preparation of this manuscript, the authors used OpenAI Codex [26.721.41059] for Chinese-to-English translation and language editing. The authors reviewed and edited all output and take full responsibility for the publication's content.

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

References

1. McDowell, J.C. The Low Earth Orbit Satellite Population and Impacts of the SpaceX Starlink Constellation. *ApJL* **2020**, *892*, L36. [CrossRef]
2. European Space Agency. ESA Space Environment Report. 2026. Available online: https://www.sdo.esoc.esa.int/environment_report/Space_Environment_Report_latest.pdf (accessed on 9 June 2026).
3. Starlink. Progress 2025. Available online: https://starlink.com/public-files/starlinkProgressReport_2025.pdf (accessed on 21 July 2026).
4. Kessler, D.J.; Johnson, N.L.; Liou, J.-C.; Matney, M. The Kessler Syndrome: Implications to Future Space Operations. *Adv. Astronaut. Sci.* **2010**, *137*, 47–61.
5. Krage, F.J. *NASA Spacecraft Conjunction Assessment and Collision Avoidance Best Practices Handbook*; NASA/SP-20230002470, Revision 1; National Aeronautics and Space Administration: Washington, DC, USA, 2023. Available online: https://ntrs.nasa.gov/citations/20230002470 (accessed on 29 September 2026).
6. Bombardelli, C. Optimal Impulsive Collision Avoidance. *Adv. Astronaut. Sci.* **2013**, *148*, 3663–3679.
7. Hernando-Ayuso, J.; Bombardelli, C. Low-Thrust Collision Avoidance in Circular Orbits. *J. Guid. Control Dyn.* **2021**, *44*, 983–995. [CrossRef]
8. De Vittori, A.; Dani, G.; Di Lizia, P.; Armellin, R. Numerically Efficient Low Thrust Collision Avoidance Maneuver Design in GEO Regime with Equinoctial Orbital Elements. In *Proceedings of the 2nd NEO and Debris Detection Conference, Darmstadt, Germany, 24–26 January 2023*; ESA Space Debris Office: Darmstadt, Germany, 2023. Available online: https://conference.sdo.esoc.esa.int/proceedings/neosst2/paper/80 (accessed on 29 September 2026).
9. Gonzalo, J.L.; Colombo, C.; Di Lizia, P. A Semi-Analytical Approach to Low-Thrust Collision Avoidance Manoeuvre Design. In Proceedings of the 70th International Astronautical Congress (IAC 2019), Washington, DC, USA, 21–25 October 2019.
10. Armellin, R. Collision Avoidance Maneuver Optimization with a Multiple-Impulse Convex Formulation. *Acta Astronaut.* **2021**, *186*, 347–362. [CrossRef]
11. Pavanello, Z.; Pirovano, L.; Armellin, R.; De Vittori, A.; Di Lizia, P. A Convex Optimization Method for Multiple Encounters Collision Avoidance Maneuvers. In *Proceedings of the AIAA SCITECH 2024 Forum, Orlando, FL, USA, 8–12 January 2024*; American Institute of Aeronautics and Astronautics: Reston, VA, USA, 2024. [CrossRef]
12. Wen, M.; Topcu, U. Constrained Cross-Entropy Method for Safe Reinforcement Learning. In *Advances in Neural Information Processing Systems 31*; Curran Associates, Inc.: Red Hook, NY, USA, 2018.
13. Kabir, M.R.; Faysal, T.I.; Hossain, M.S.; Shorno, J.N.; Siddique, S. A Satellite Collision Avoidance System Based on General Regression Neural Network. In *Proceedings of the 2020 IEEE/ACM International Conference on Big Data Computing, Applications and Technologies (BDCAT), Leicester, UK, 7–10 December 2020*; IEEE: New York, NY, USA, 2020; pp. 154–160. [CrossRef]
14. Jang, D.; Siew, P.M.; Gondelach, D.; Linares, R. Space Situational Awareness Tasking for Narrow Field of View Sensors: A Deep Reinforcement Learning Approach. In Proceedings of the 71st International Astronautical Congress (IAC 2020), Virtual Conference, 12–14 October 2020.
15. Peng, H.; Bai, X. Gaussian Processes for Improving Orbit Prediction Accuracy. *Acta Astronaut.* **2019**, *161*, 44–56. [CrossRef]
16. Peng, H.; Bai, X. Artificial Neural Network–Based Machine Learning Approach to Improve Orbit Prediction Accuracy. *J. Spacecr. Rockets* **2018**, *55*, 1248–1260. [CrossRef]
17. Salleh, N.; Yuhaniz, S.S.; Mohd Azmi, N.F. Modeling Orbital Propagation Using Regression Technique and Artificial Neural Network. *Int. J. Adv. Sci. Eng. Inf. Technol.* **2022**, *12*, 1279–1286. [CrossRef]
18. Liao, J.; Xu, H.; Luo, Y.; Meng, Y. A Patch-Transformer-Based Method for Orbit Prediction of LEO Satellites. *Aerosp. Sci. Technol.* **2026**, *176*, 112652. [CrossRef]
19. Cipollone, R.; Frueh, C.; Di Lizia, P. LSTM-Based Maneuver Detection for Resident Space Object Catalog Maintenance. *Neural Comput. Appl.* **2025**, *37*, 17341–17362. [CrossRef]
20. Kazemi, S.; Azad, N.L.; Scott, K.A.; Oqab, H.B.; Dietrich, G.B. Satellite Collision Avoidance Maneuver Planning in Low Earth Orbit Using Proximal Policy Optimization. In *Proceedings of the 2024 IEEE Congress on Evolutionary Computation (CEC), Yokohama, Japan, 30 June–5 July 2024*; IEEE: New York, NY, USA, 2024; pp. 1–9. [CrossRef]

21. Li, Z.; Li, H.; Li, C. ELVO-Based Autonomous Satellite Collision Avoidance with Multiple Debris. *Aerospace* **2025**, *12*, 402. [CrossRef]
22. Bourriez, N.; Loizeau, A.; Abdin, A.F. Spacecraft Autonomous Decision-Planning for Collision Avoidance: A Reinforcement Learning Approach. *arXiv* **2023**, arXiv:2310.18966.
23. Chen, X.; Wang, T.; Qiu, J.; Feng, J. Mission Planning on Autonomous Avoidance for Spacecraft Confronting Orbital Debris. *IEEE Trans. Aerosp. Electron. Syst.* **2025**, *61*, 4115–4126. [CrossRef]
24. Mu, C.; Liu, S.; Lu, M.; Liu, Z.; Cui, L.; Wang, K. Autonomous Spacecraft Collision Avoidance with a Variable Number of Space Debris Based on Safe Reinforcement Learning. *Aerosp. Sci. Technol.* **2024**, *149*, 109131. [CrossRef]
25. Wang, Y.; Chang, S.; Tang, H.; Wang, C. Automating Satellite Collision Avoidance via Multi-Agent Reinforcement Learning. In *Proceedings of the 2025 6th International Conference on Computer Information and Big Data Applications, Wuhan, China, 14–16 March 2025*; Association for Computing Machinery: New York, NY, USA, 2025; pp. 1284–1290. [CrossRef]
26. Lowe, R.; Wu, Y.; Tamar, A.; Harb, J.; Abbeel, P.; Mordatch, I. Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments. In *Advances in Neural Information Processing Systems 30*; Curran Associates, Inc.: Red Hook, NY, USA, 2017.
27. Foerster, J.; Farquhar, G.; Afouras, T.; Nardelli, N.; Whiteson, S. Counterfactual Multi-Agent Policy Gradients. *Proc. AAAI Conf. Artif. Intell.* **2018**, *32*, 2974–2982. [CrossRef]
28. Sunehag, P.; Lever, G.; Gruslys, A.; Czarnecki, W.M.; Zambaldi, V.; Jaderberg, M.; Lanctot, M.; Sonnerat, N.; Leibo, J.Z.; Tuyls, K.; et al. Value-Decomposition Networks for Cooperative Multi-Agent Learning. *arXiv* **2017**, arXiv:1706.05296.
29. Rashid, T.; Samvelyan, M.; de Witt, C.S.; Farquhar, G.; Foerster, J.; Whiteson, S. Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning. *J. Mach. Learn. Res.* **2020**, *21*, 1–51.
30. Hu, J.; Jiang, S.; Harding, S.A.; Wu, H.; Liao, S. Rethinking the Implementation Tricks and Monotonicity Constraint in Cooperative Multi-Agent Reinforcement Learning. *arXiv* **2021**, arXiv:2102.03479.
31. Yang, Y.; Hao, J.; Liao, B.; Shao, K.; Chen, G.; Liu, W.; Tang, H. Qatten: A General Framework for Cooperative Multiagent Reinforcement Learning. *arXiv* **2020**, arXiv:2002.03939.
32. Yu, C.; Velu, A.; Vinitsky, E.; Gao, J.; Wang, Y.; Bayen, A.; Wu, Y. The Surprising Effectiveness of PPO in Cooperative Multi-Agent Games. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 24611–24624. [CrossRef]
33. Su, J.; Adams, S.; Beling, P. Value-Decomposition Multi-Agent Actor-Critics. *Proc. AAAI Conf. Artif. Intell.* **2021**, *35*, 11352–11360. [CrossRef]
34. Tavakoli, A.; Pardo, F.; Kormushev, P. Action Branching Architectures for Deep Reinforcement Learning. *Proc. AAAI Conf. Artif. Intell.* **2018**, *32*, 4131–4138. [CrossRef]
35. Parker, W.E.; Freeman, M.; Chisham, G.; Kavanagh, A.; Mun Siew, P.; Rodriguez-Fernandez, V.; Linares, R. Influences of Space Weather Forecasting Uncertainty on Satellite Conjunction Assessment. *Space Weather* **2024**, *22*, e2023SW003818. [CrossRef]
36. Burgis, S.; Rübberdt, H.; Gaedigk, C.; Keuper, L.; Naufal, G.; Paetzold, J.; Oikonomidou, X.; Bastida Virgili, B. MAS—A Mission Analysis Software for Collision Risk Quantification and Impact Assessment of Rule-Based Decision-Making for Collision Avoidance. *J. Space Saf. Eng.* **2025**, *12*, 338–356. [CrossRef]
37. Vallado, D.A. *Fundamentals of Astrodynamics and Applications*, 5th ed.; Microcosm Press: Portland, OR, USA, 2022.
38. Bombardelli, C. Analytical Formulation of Impulsive Collision Avoidance Dynamics. *Celest. Mech. Dyn. Astron.* **2014**, *118*, 99–114. [CrossRef]
39. Gonzalo, J.L.; Colombo, C.; Di Lizia, P. Analytical Framework for Space Debris Collision Avoidance Maneuver Design. *J. Guid. Control Dyn.* **2021**, *44*, 469–487. [CrossRef]
40. Chen, L.; Bai, X.-Z.; Liang, Y.-G.; Li, K.-B. *Orbital Data Applications for Space Objects: Conjunction Assessment and Situation Analysis*; Springer: Singapore, 2017. [CrossRef]

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