CLOSED-LOOP SATELLITE COLLISION AVOIDANCE: A BENCHMARK AND PLANNER-GUIDED LEARNING FRAMEWORK

Anonymous authors
Paper under double-blind review

ABSTRACT

Satellite collision avoidance (SCA) is a closed-loop sequential decision problem: maneuvers must reduce conjunction risk, limit delivered velocity increment, and recover the spacecraft toward its designated orbit. Existing studies use incompatible assumptions that obscure this trade-off. We introduce SCA-Bench, combining catalog-derived states and historical encounter geometries with finite-burn control, online re-screening, recovery, and single- and multi-object scenarios. Decoupled warning and execution models include measured prediction–execution discrepancy in warning covariance. We also present SCA-Planner, which behavior-clones offline finite-burn planner trajectories before warning-gated proximal policy optimization. On a fixed 128-scenario test set, OrbitZoo obtains the highest collision-probability compliance (99.21%), whereas SCA-Planner uses less cumulative maneuver effort (0.1850 vs. 0.4300 m/s), recovers closer to the designated orbit (0.5800 vs. 0.4600), and attains a higher composite score. Ablations expose distinct operating points; no method dominates safety, maneuver effort, and recovery.

1 INTRODUCTION

The continued expansion of satellite constellations, together with the accumulation of orbital debris, has increased congestion and collision risk in low Earth orbit (Pardini & Anselmo, 2020; Foreman et al.). A collision can generate additional fragments that further increase the likelihood of future conjunctions and, in the long term, contribute to a self-sustaining collision cascade (Kessler & Cour-Palais, 1978; Liou & Johnson, 2006; 2008). Reliable threat assessment and collision avoidance are therefore increasingly important for the safe and sustainable operation of large satellite populations (Krage, 2026). Post-maneuver recovery is additionally important for returning a spacecraft toward its designated mission orbit.

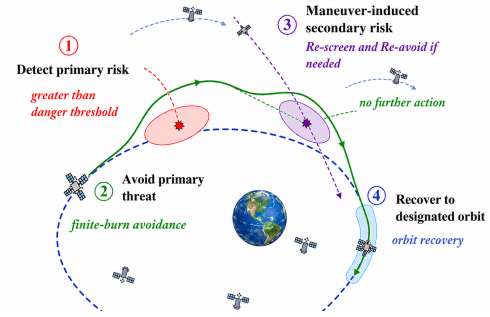


Figure 1: **Closed-loop SCA in SCA-Bench:** detect, avoid, re-screen, and recover.

Satellite collision avoidance (SCA) is not simply a one-shot maneuver problem. An avoidance maneuver consumes limited capability, displaces the spacecraft from its designated orbit, and changes the state from which subsequent threats are assessed. Warning estimates also evolve, and the executed trajectory may deviate from the prediction. Resolving one conjunction can therefore affect both recovery and later threats, making SCA a closed-loop sequential decision problem.

A useful controller must jointly reduce collision risk, limit cumulative delivered velocity increment, and recover toward the designated orbit. These objectives are coupled: aggressive maneuvering may improve compliance but spend more capability and hinder recovery, whereas conservative control may preserve the orbit while leaving residual risk. In multi-object scenes, a maneuver may also change which object becomes the next threshold-exceeding threat. Different SCA methods may

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

therefore realize different safety–maneuver–recovery operating points rather than uniformly dominate one another.

Meaningful comparison requires more than common initial states. Existing resources address historical risk prediction, learned low-thrust avoidance and return, or reusable orbital-control simulation (Uriot et al., 2022; Solomon & Paduraru, 2025; Oliveira et al., 2025), but use different task, prediction, execution, intervention, and evaluation assumptions. Direct controller-level comparison therefore requires a common closed-loop protocol while leaving the controller replaceable.

We introduce SCA-Bench as this common evaluation setting. Catalog-derived states and historical encounter geometries are evaluated with continuous finite-burn execution, repeated screening, recovery, and single- and multi-object scenes. Warning prediction is separated from numerical execution, with measured discrepancy included in warning covariance. Fixed test sets report collision-probability compliance (CPCR), cumulative maneuver effort (CME), orbit recovery (ORS), and a composite score alongside its constituents so that distinct operating points remain visible.

The benchmark also exposes sparse continuous control: recorded conjunction data are dominated by low-risk events (Uriot et al., 2022), while uninformed safety-critical exploration can be unsafe and wasteful (García et al., 2015). We therefore develop SCA-Planner, a reference learner in which an offline finite-burn planner supplies demonstrations, behavior cloning initializes the continuous action policy and recurrent Intervention Gate, and PPO (Schulman et al., 2017) refines the policy in the same closed-loop environment. The benchmark itself remains independent of this policy class.

On Test128, OrbitZoo (Oliveira et al., 2025) achieves higher compliance, whereas SCA-Planner uses less maneuver effort and recovers closer to the designated orbit. Pipeline and reward ablations likewise shift the operating point rather than yield uniform gains.

Our contributions are:

- **SCA-Bench**, a closed-loop SCA evaluation setting that standardizes scenario construction, warning, finite-burn execution, re-screening, recovery, and method-independent metrics across single- and multi-object scenarios.
- **SCA-Planner**, a planner-guided reference learner that combines behavior-cloning initialization with PPO in the same closed-loop environment.
- **A trade-off-centered empirical study** comprising fixed single- and multi-object comparisons and ablations that expose distinct safety–maneuver–recovery operating points.

Together, these components provide a controlled closed-loop setting for comparing safety–maneuver–recovery trade-offs across controller classes.

2 Related Work

Maneuver optimization and collision-risk assessment. Classical collision-avoidance maneuver design optimizes maneuver magnitude, direction, or timing to increase miss distance or reduce collision probability while limiting maneuver cost (Patera & Peterson, 2003; Bombardelli & Hernando-Ayuso, 2015). Multiple-impulse convex formulations extend this setting and can approximate finite-burn arcs associated with short warning times or low-thrust propulsion (Armellin, 2021). Collision risk is commonly quantified from the relative state and combined uncertainty of the two objects, often through a two-dimensional representation of the encounter geometry (Patera, 2001; Akella & Alfriend, 2000). Recent surveys further organize collision-avoidance maneuver methods across analytical, semi-analytical, and numerical approaches, as well as impulsive and continuous-thrust formulations (Pavanello et al., 2025). These methods provide the physical structure behind several baselines in SCA-Bench. Our focus is complementary: rather than proposing another task-specific maneuver optimizer, we evaluate optimization-based and learned controllers under the same warning, execution, re-screening, and recovery protocol.

Learning for conjunction management. Learning has been applied to several stages of conjunction management. The ESA Spacecraft Collision Avoidance Challenge studies prediction of final collision risk from histories of conjunction data messages (Uriot et al., 2022). For maneuver decision making, Bourriez et al. (Bourriez et al., 2023) formulate autonomous collision avoidance as a partially

observable decision problem, while Kazemi et al. (Kazemi et al., 2024) train PPO policies for collision-avoidance maneuver planning. Mu et al. (Mu et al., 2024) formulate autonomous spacecraft collision avoidance with a variable number of debris objects as a constrained Markov decision process and apply safe reinforcement learning. Solomon and Paduraru (Solomon & Paduraru, 2025) jointly study collision-avoidance and return maneuvers for low-thrust satellites, and OrbitZoo (Oliveira et al., 2025) provides a high-fidelity reinforcement learning environment spanning collision avoidance and other orbital-control tasks. Collectively, these works cover risk prediction, direct maneuver-policy learning, constrained control, orbital return, and reusable simulation infrastructure. Because they target different problems and adopt different scenario, action, dynamics, intervention, and evaluation setups, however, their reported safety, maneuver-effort, and recovery outcomes are difficult to compare directly. SCA-Bench addresses this comparison problem by fixing the closed-loop evaluation protocol while leaving the controller itself replaceable.

Guided policy learning and safe control. Imitation learning provides a general mechanism for acquiring sequential policies from expert demonstrations (Ross et al., 2011), while PPO provides a practical on-policy approach for continuous policy optimization (Schulman et al., 2017). More broadly, safe reinforcement learning incorporates risk or operational constraints into learning and exploration (García et al., 2015), including policy-optimization methods with explicit cost constraints (Achiam et al., 2017). Our reference method, SCA-Planner, combines optimization-generated demonstrations with behavior-cloning initialization, warning-gated intervention, and PPO refinement inside the same closed-loop environment used for evaluation. This provides a learning-based reference point for the benchmark while preserving a common safety–maneuver–recovery evaluation protocol across controller classes.

3 SCA-BENCH

SCA-Bench fixes prediction, execution, scenario construction, and evaluation independently of controller design.

3.1 PROBLEM FORMULATION

Let $\mathbf{r}(t)$ and $\mathbf{v}(t)$ denote the position and velocity of the controlled spacecraft, and $\widetilde{\mathbf{r}}(t)$ and $\widetilde{\mathbf{v}}(t)$ those of its designated orbit. For each candidate object $j \in \mathcal{J}$, let $\mathbf{r}_j(t)$ and $\mathbf{v}_j(t)$ denote its position and velocity. Assuming no further maneuver, the warning model predicts the TCA over a look-ahead window $\mathcal{T}$ as

$$\widehat{\tau}_j(t) = \arg\min_{\tau \in \mathcal{T}} \left\| \widehat{\mathbf{r}}_j(\tau \mid t) - \widehat{\mathbf{r}}(\tau \mid t) \right\|_2 . \tag{1}$$

The warning model then computes the corresponding collision probability $\widehat{P}_{\mathrm{c}}^{j}(t)$ at $\widehat{\tau}_j(t)$. A warning is active when $\widehat{P}_{\mathrm{c}}^{j}(t) > \widetilde{P}_{\mathrm{c}}$, with $\widetilde{P}_{\mathrm{c}} = 3 \times 10^{-7}$.

At each 30-second control interval, a policy receives the current spacecraft state, designated-orbit deviation, and a warning summary for the earliest threshold-exceeding conjunction. It outputs a normalized command in the local radial–transverse–normal (RTN) frame,

$$\widehat{\mathbf{a}}_t = \pi(\mathbf{o}_{1:t}) \in [-1, 1]^3. \tag{2}$$

The propulsion and execution models convert this command into a finite burn. One episode contains 240 intervals (two hours), covering avoidance and recovery.

3.2 CLOSED-LOOP PREDICTION AND EXECUTION

The execution model numerically integrates every object with fourth-order Runge–Kutta and supplies realized trajectories for evaluation. Independently, the SGP4 warning model (Vallado et al., 2006) reconstructs elements from the current numerical state, propagates a fixed window, screens pairs, refines local minima, and computes encounter-plane collision probability at every interval. The resulting controlled prediction–execution mismatch is measured and added to event covariance; Appendix A.1 reports its percentile profiles. This design evaluates fresh predictions after each finite burn but does not claim a complete operational model.

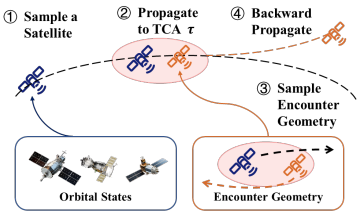


Figure 2: **Scenario construction:** sample encounter geometry at TCA and propagate both objects backward to initialize the episode.

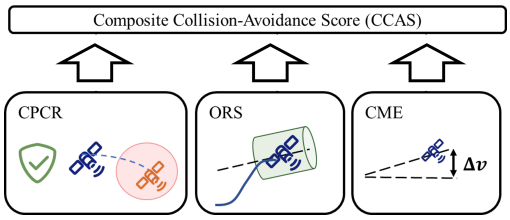


Figure 3: **Evaluation metrics:** CPCR, CME, and ORS measure safety, effort, and recovery, respectively; CCAS summarizes the trade-off.

3.3 Scenario Construction

Because high-risk conjunctions are scarce, controlled episodes combine a Starlink Supplemental GP ego state (cel) with historical relative miss and velocity geometry from the ESA challenge (esa). Bounded perturbations vary direction and speed, a scaled miss distance sets the provisional risk regime, and backward propagation from the sampled TCA yields the initial state (Figure 2). Candidates must have admissible states and orbital elements, and the numerical no-maneuver rollout and warning model must agree on the hazardous/safe label. Accepted cases are deduplicated; train and test are disjoint; and test cases never enter the demonstration set.

3.4 Benchmark Statistics

The benchmark contains 1,408 training and 128 test scenarios, respectively 70.0% and 70.3% hazardous; Test128 has 90 hazardous and 38 safe cases. The multi-object extension adds 2,280 background objects per base test scene. In all 90 hazardous scenes, at least one of six fixed-magnitude RTN probes changes the selected warning target (278 probes in total). These are dataset acceptance tests, not policy actions or evidence of resolving secondary conjunctions. Figure 4 summarizes the resulting risk and probe-induced target-switch distributions; these probe statistics characterize scenario sensitivity rather than policy performance.

3.5 Evaluation Protocol and Metrics

For episode i, let $P_{\mathrm{c}}^{j,(i)}$ be the collision probability computed from the realized numerical trajectory for object j. CPCR is the fraction of episodes whose worst realized event is compliant:

$$\mathrm{CPCR} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}\left[\max_{j\in\mathcal{J}^{(i)}} P_{\mathrm{c}}^{j,(i)} \le \widetilde{P}_{\mathrm{c}}\right]. \tag{3}$$

We report CPCR as a percentage in tables.

The cumulative delivered velocity increment for episode i is $\Delta V_T^{(i)} = \sum_{t=1}^{T} \|\Delta \mathbf{v}_t^{(i)}\|_2$. Its test-set mean defines cumulative maneuver effort,

$$\mathrm{CME} = \frac{1}{N}\sum_{i=1}^{N} \Delta V_T^{(i)}. \tag{4}$$

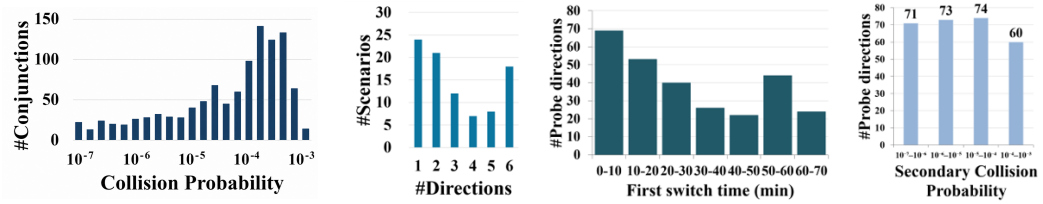
CME has units of m/s and is a proxy for propellant expenditure; it is not an energy measure.

Recovery compares the terminal state with the designated orbit. In the designated-orbit RTN frame, with orbital-tube scale vector $\mathbf{d} \in \mathbb{R}^3$ and designated-orbit mean motion $n^{(i)}$, define

$$\begin{aligned} \epsilon_r^{(i)} &= \left\| \left(\mathbf{r}^{(i)}(T) - \widetilde{\mathbf{r}}^{(i)}(T)\right)/\mathbf{d} \right\|_2, \\ \epsilon_v^{(i)} &= \left\| \left(\mathbf{v}^{(i)}(T) - \widetilde{\mathbf{v}}^{(i)}(T)\right)/(n^{(i)}\mathbf{d}) \right\|_2. \end{aligned} \tag{5}$$

The orbit recovery score is

$$\mathrm{ORS} = \frac{1}{N}\sum_{i=1}^{N} \frac{1}{1 + w_r \epsilon_r^{(i)} + w_v \epsilon_v^{(i)}}, \tag{6}$$



(a) Risk distribution (b) Triggering (c) First-switch time (d) First-switch P_c

Figure 4: **Benchmark statistics:** (a) no-maneuver risk distribution, (b) triggering directions per scenario, (c) first target-switch time, and (d) first target-switch collision probability.

where $w_r = w_v = 0.5$ weight the position and velocity deviations equally.

CPCR, CME, and ORS are primary. For a compact composite view, we also report

$$\mathrm{CCAS} = \frac{\mathrm{CPCR}}{1 + (1 - \mathrm{ORS})/(1 - r^\star) + \mathrm{CME}/v^\star}, \tag{7}$$

with policy-independent calibration $r^\star \approx 0.34$ and $v^\star \approx 0.24$ m/s. CPCR is a fraction in Equation 7; CCAS summarizes the reported trade-off but does not establish universal superiority. Appendix C gives the calibration.

3.6 Benchmark scalability

The warning pipeline averages 0.0301 s for 1,000 objects and 0.5545 s for 10,000 objects on reported Starlink catalog states; full implementation-specific results are in Appendix A.1.

4 SCA-Planner

SCA-Planner is a planner-guided reference framework for sparse continuous control. An offline finite-burn teacher supplies intervention trajectories; behavior cloning (BC) initializes when and how to maneuver; and closed-loop PPO adapts that policy under the benchmark objectives. This task-specific pipeline does not treat BC plus PPO as a new reinforcement-learning algorithm.

4.1 Sparse-intervention policy

The policy separates what maneuver to execute from whether to intervene. The feed-forward action branch $\mathcal{A}$ proposes a maneuver from the current observation, while the recurrent Gate $\mathcal{G}$ uses observation history to determine whether that maneuver should be executed. During training,

$$\begin{aligned} \boldsymbol{\mu}_t^a &= \mathcal{A}(\mathbf{o}_t), & p_t^g &= \mathcal{G}(\mathbf{o}_{1:t}), \\ \mathbf{z}_t &\sim \mathcal{N}\big(\boldsymbol{\mu}_t^a, \mathrm{diag}(\boldsymbol{\sigma}_a^2)\big), & b_t &\sim \mathrm{Bernoulli}(p_t^g), \\ \widehat{\mathbf{a}}_t &= b_t \tanh(\mathbf{z}_t), \end{aligned} \tag{8}$$

where $\boldsymbol{\sigma}_a$ is a trainable standard deviation. During deterministic inference,

$$\mathbf{z}_t = \boldsymbol{\mu}_t^a, \qquad b_t = \mathbf{1}[p_t^g \geq \widetilde{p}^g], \tag{9}$$

where $\widetilde{p}^g$ is the gating threshold.

The 20-dimensional observation contains threat RTN state, warning and risk dynamics, TCA, cumulative maneuver, elapsed time, and designated-orbit deviation; continuous features use fixed physical normalization and clipping.

The two branches answer different questions. The feed-forward action branch maps the current geometry to a bounded burn, while the recurrent Gate can use the evolution of warning status and risk to distinguish a transient safe prediction from sustained clearance. This separation is important because a good action direction need not imply that a burn should be executed at every subsequent control step.

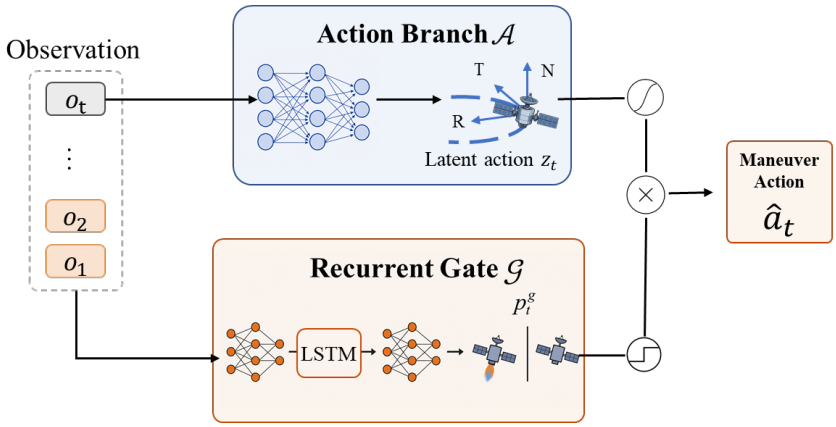


Figure 5: **Sparse-intervention policy.** The action branch proposes a maneuver, while the recurrent Gate determines whether to execute it.

4.2 Planner-guided behavior cloning

For each unsafe scenario, the teacher searches plans with at most two finite-burn intervals at times tied to first warning, predicted TCA, and the episode horizon. Candidates are replayed in the closed-loop environment, ranked by safety, recovery, and cumulative ΔV, optionally refined, and retained only after replay-based safety and consistency checks. Safe scenarios retain the no-op trajectory.

The demonstration set consists of sequences $\mathcal{D} = \{(\mathbf{o}_t, b_t^\star, \mathbf{a}_t^\star)\}_{t=1}^T$, where $b_t^\star$ indicates whether the teacher intervenes and $\mathbf{a}_t^\star$ is the corresponding maneuver command when $b_t^\star = 1$.

Crucially, candidates become demonstrations only after execution in the same finite-burn, re-screening loop used for learning. The labels therefore describe realized intervention timing and commands rather than the teacher's internal prediction alone. Including safe no-op trajectories teaches the Gate not to manufacture maneuvers when the initial warning state is already compliant, but also creates the severe class imbalance addressed below.

BC uses a class-balanced binary loss for the Gate so that abundant no-op steps do not suppress intervention. For teacher maneuver label $b_t^\star$, let $\mathcal{T}^+$ and $\mathcal{T}^-$ contain intervention and no-intervention steps. The Gate objective is

$$\mathcal{L}_{\text{gate}} = -\frac{1}{|\mathcal{T}^+|}\sum_{t\in\mathcal{T}^+}\log p_t^g - \frac{1}{|\mathcal{T}^-|}\sum_{t\in\mathcal{T}^-}\log(1-p_t^g). \tag{10}$$

The action branch is supervised only on $\mathcal{T}^+$. For teacher commands $\mathbf{a}_t^\star \in (-1,1)^3$, define $\mathbf{z}_t^\star = \operatorname{atanh}(\mathbf{a}_t^\star)$ and use

$$\begin{aligned}\mathcal{L}_{\text{action}} &= \frac{1}{|\mathcal{T}^+|}\sum_{t\in\mathcal{T}^+}\text{SmoothL1}(\mathbf{z}_t, \mathbf{z}_t^\star),\\ \mathcal{L}_{\text{exec}} &= \frac{1}{|\mathcal{T}^+|}\sum_{t\in\mathcal{T}^+}\|\tanh(\mathbf{z}_t) - \mathbf{a}_t^\star\|_2^2.\end{aligned} \tag{11}$$

Thus, latent supervision preserves the policy parameterization while executed action supervision matches the bounded command seen by the simulator. BC first trains the Gate with $\mathcal{L}_{\text{gate}}$ and the action branch with $\mathcal{L}_{\text{action}} + \mathcal{L}_{\text{exec}}$; the two branches are then jointly fine-tuned with the combined objective.

4.3 Reinforcement-learning fine-tuning

PPO (Schulman et al., 2017) refines the cloned actor in the closed-loop benchmark. Its hybrid likelihood is

$$\pi(b_t, \mathbf{z}_t \mid \mathbf{o}_{1:t}) = \pi^g(b_t \mid \mathbf{o}_{1:t})\,[\pi^a(\mathbf{z}_t \mid \mathbf{o}_t)]^{b_t}. \tag{12}$$

The Gate likelihood always contributes, whereas the continuous likelihood contributes only when an intervention is sampled. The critic starts from scratch because demonstrations contain no PPO value targets. The Gate is frozen for the first 5×10^5 environment steps to preserve the learned intervention pattern while PPO first adapts action magnitude and direction. Training samples the 986 initial-warning cases among 1,408 training scenarios, preventing deterministic no-op episodes from dominating batches; evaluation still uses all Test128 cases.

BC and PPO consequently serve complementary roles. BC places the actor near a sparse intervention distribution that random exploration may rarely discover; PPO then evaluates those actions through complete closed-loop rollouts, where post-maneuver warnings and recovery outcomes enter the return. The refinement stage is therefore not a second imitation pass: it can move away from the teacher's selected operating point while retaining its intervention prior.

4.4 PHASE-AWARE MULTI-OBJECTIVE LEARNING

Four phases route learning signals. NOMINAL covers warning-free operation; a warning enters HAZARD-ACTIVE; ten consecutive clear steps move through CLEAR-CONFIRMING to RECOVERY; and any new warning returns to HAZARD-ACTIVE. Risk progress is rewarded only during an active hazard. With $x_t = \ln(\widehat{P}_{\mathrm{c}}(t)/\widetilde{P}_{\mathrm{c}})$, its progress-and-residual term is

$$R_t^{\mathrm{risk}} = x_{t-1} - \gamma x_t - \lambda^{\mathrm{risk}} x_t. \tag{13}$$

Incremental maneuver effort and its soft cumulative budget use

$$\begin{aligned} R_t^{\Delta V} &= \Delta V_{t-1} - \Delta V_t, \\ R_t^{\mathrm{budget}} &= [\Delta V_{t-1}/\widetilde{\Delta V} - 1]_+^2 - [\Delta V_t/\widetilde{\Delta V} - 1]_+^2, \end{aligned} \tag{14}$$

where $[x]_+ = \max(x, 0)$. The incremental term applies throughout and the budget term outside HAZARD-ACTIVE. After clearance, recovery reward tracks progress in the time-indexed score and penalizes deficits below target $\widetilde{\mathrm{ORS}}$:

$$R_t^{\mathrm{rec}} = \gamma \mathrm{ORS}_t - \mathrm{ORS}_{t-1} - \lambda^{\mathrm{rec}} \left[\frac{\widetilde{\mathrm{ORS}} - \mathrm{ORS}_t}{\widetilde{\mathrm{ORS}}} \right]_+^2. \tag{15}$$

Finally, the complete simulator-propagated trajectory supplies closest-event probability $P_c^\star$ and terminal reward:

$$R_T^{\mathrm{term}} = \begin{cases} -\widetilde{R}_1, & P_c^\star > \widetilde{P}_c^{\mathrm{col}}, \\ -\widetilde{R}_2 - \lambda^{\mathrm{term}} \ln(P_c^\star/\widetilde{P}_{\mathrm{c}}), & \widetilde{P}_{\mathrm{c}} < P_c^\star \le \widetilde{P}_c^{\mathrm{col}}, \\ \widetilde{R}_3, & P_c^\star \le \widetilde{P}_{\mathrm{c}}. \end{cases} \tag{16}$$

where $\widetilde{R}_1$, $\widetilde{R}_2$, and $\widetilde{R}_3$ denote the hard-collision penalty, unsafe-trajectory penalty, and safe-trajectory bonus, respectively. Routing prevents recovery or budget shaping from competing with immediate risk reduction during an active warning. Section 5.4 tests how this routing changes the operating point; Appendix A.7 gives the coefficients.

5 EXPERIMENTS

We evaluate controller operating points, multi-object behavior, the BC–PPO pipeline, intervention gating, objective routing, and an illustrative closed-loop episode.

5.1 EXPERIMENTAL SETUP

All methods run deterministically for one complete episode on each Test128 case (90 hazardous, 38 safe). Learned policies use one NVIDIA GeForce RTX 4090 and seed 42; each configuration is therefore one trained policy on a fixed suite, without confidence intervals or significance tests. We compare no-op, four optimization baselines (Armellin, 2021; Patera & Peterson, 2003; Bombardelli & Hernando-Ayuso, 2015; alf), OrbitZoo (Oliveira et al., 2025) adapted to the common protocol, and SCA-Planner. Appendix A.8 details implementations and runtime.

5.2 BASELINE COMPARISON

Table 1: Test128 baseline comparison; runtime is mean per-episode evaluation time.

Method	CPCR (%) ↑	CME (m/s) ↓	ORS ↑	CCAS ↑	Runtime (s) ↓
No-op	29.69	0.0000	1.0000	0.2969	6.82
Armellin (2021)	29.69	0.0545	0.9416	0.2260	33.45
Patera (2003)	50.00	0.0993	0.7372	0.2765	69.74
Bombardelli (2015)	57.03	0.0905	0.7188	0.3168	84.95
Alfano '05 (alf)	60.94	0.1800	0.7740	0.2921	33.31
OrbitZoo (2025)	99.21	0.4300	0.4600	0.2760	14.57
SCA-Planner	83.59	0.1850	0.5800	0.3482	10.52

OrbitZoo has the highest CPCR (99.21%), whereas SCA-Planner uses less CME (0.1850 versus 0.4300 m/s) and has higher ORS (0.5800 versus 0.4600). The optimization baselines occupy intermediate points; Armellin in particular shows why low CME is meaningful only with its safety outcome. All commands are executed in the same finite-burn environment, but these differences describe a fixed suite rather than a population estimate.

Mean runtime ranges from 6.82 s for no-op to 84.95 s for Bombardelli '15; SCA-Planner and OrbitZoo require 10.52 s and 14.57 s. Optimization rows include online planning and execution, while learned-policy rows exclude offline training, so these are not total training costs. Although SCA-Planner has the highest CCAS under the fixed calibration, the composite must be read with CPCR, CME, and ORS; it is neither a safety ranking nor evidence of universal superiority.

5.3 MULTI-OBJECT EVALUATION

The 2,280-object backgrounds test whether a maneuver selected for the current threat changes what the warning system sees next. Fixed-direction acceptance probes establish that target changes are possible in every hazardous base scene; they do not show that a policy recognizes or resolves the new event. Table 2 asks the policy-level question using aggregate episode outcomes.

Table 2: Single- and multi-object Test128 outcomes.

Method	Setting	CPCR (%)	CME (m/s)	ORS	CCAS
OrbitZoo	Single	99.21	0.4300	0.4600	0.2760
	Multiple	87.50	0.4736	0.4681	0.2329
SCA-Planner	Single	83.59	0.1850	0.5800	0.3482
	Multiple	82.81	0.1851	0.5818	0.3453

OrbitZoo's CPCR changes from 99.21% to 87.50%, whereas SCA-Planner changes from 83.59% to 82.81% with nearly unchanged CME. The latter changes less here, but the measurements neither attribute failures to particular secondary objects nor establish general multi-object robustness.

5.4 ABLATION STUDY

Planner-guided learning pipeline. Table 3 first isolates the learning-pipeline contrast. Random-init PPO is no-op-like, while lowering its Gate threshold obtains compliance only with excessive CME; BC supplies a sparse prior and PPO improves its CPCR with modest additional effort. This clarifies stage roles, not general training stability or seed variation.

Intervention Gate. The feed-forward and always-active variants raise CPCR but spend more CME and recover less, showing that the Gate controls timing and sparsity, not only compliance.

Objective and reward routing. The corresponding block in Table 3 shows that removing risk lowers effort and improves recovery but sharply reduces compliance; removing ΔV or recovery, or

Table 3: Ablations of the learning pipeline, intervention Gate, and reward routing on Test128.

Configuration	CPCR (%)	CME (m/s)	ORS	CCAS
SCA-Planner (reference)	83.59	0.1850	0.5800	0.3482
Learning pipeline variants				
Random-init PPO	29.69	0.0000	1.0000	0.2969
Random-init PPO (low Gate)	80.47	6.6219	0.4366	0.0277
BC only	75.00	0.1632	0.6154	0.3322
Gate variants				
Feed-forward Gate	95.31	0.3653	0.4995	0.2919
Always active	100.00	64.1421	0.3153	0.0038
Objective and reward routing				
All rewards in all phases	89.84	0.2601	0.5379	0.3239
No risk objective	66.41	0.1373	0.6548	0.3176
No ΔV objective	90.63	0.2700	0.5348	0.3214
No recovery objective	92.19	0.2654	0.5266	0.3277
Uniform ΔV shaping	85.94	0.2219	0.5580	0.3323
No soft budgets	87.50	0.2229	0.5540	0.3370

activating all dense rewards, shifts the trade-off oppositely. Uniform shaping and no soft budgets are intermediate. These fixed-suite points show temporal arbitration, not uniform metric gains; Appendix B reports the secondary setting.

5.5 Qualitative case study

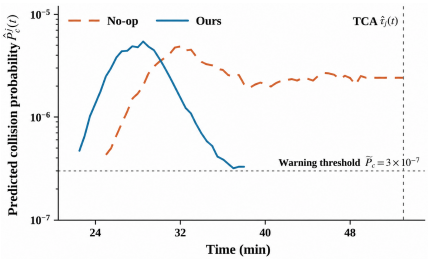


Figure 6: **Illustrative risk evolution:** SCA-Planner crosses the warning threshold before TCA; no-op remains above it.

Figure 6 illustrates one hazardous Test128 episode. SCA-Planner risk first rises to a peak, then reverses and reaches the warning threshold well before TCA, whereas the no-op trajectory remains above the threshold over the corresponding post-peak interval. The difference is temporal rather than purely terminal: it shows how compliance develops during the encounter, complementing aggregate CPCR, CME, and ORS, which summarize episode-level outcomes. This single episode is illustrative and does not establish general causal or robustness claims.

6 Conclusion

SCA-Bench provides a unified closed-loop benchmark for satellite collision avoidance, jointly evaluating collision-probability compliance, maneuver effort, and orbit recovery under finite-burn execution, re-screening, recovery, and multi-object scenarios. The benchmark highlights an important characteristic of sequential collision avoidance: improving one objective can alter the subsequent orbital state and warning context, making performance inherently dependent on the trade-off among safety, control effort, and recovery quality. Within this setting, SCA-Planner combines planner-guided behavior cloning with closed-loop PPO refinement to learn sparse continuous interventions. On Test128, OrbitZoo and SCA-Planner exhibit distinct operating points: OrbitZoo favors collision-probability compliance, while SCA-Planner achieves lower maneuver effort, closer orbit recovery, and higher CCAS. Together with the pipeline and objective ablations, these results show that SCA-Bench can differentiate meaningful control trade-offs rather than reducing evaluation to a single success metric. We hope this benchmark provides a useful foundation for studying broader policy classes, richer multi-object interactions, and more diverse orbital and dynamical conditions.

AI USE STATEMENT

Generative AI tools were used to help restructure and copyedit the manuscript and update its LaTeX formatting. They were not used to generate experiments or experimental values. Numerical statements were checked against the existing paper, supplementary material, or explicit result tables, and bibliographic claims were checked against the cited sources. The authors remain responsible for reviewing and approving the final submission.

REPRODUCIBILITY STATEMENT

The fixed data splits, evaluation metrics, model components, and experimental protocol are specified in Sections 3–5. Appendix A records the warning and execution models, training hyperparameters, reward coefficients, baseline protocols, and secondary ablations used to interpret the reported results.

REFERENCES

Review of conjunction probability methods for short-term encounters | request PDF. *ResearchGate*. URL https://www.researchgate.net/publication/289311065_Review_of_conjunction_probability_methods_for_short-term_encounters.

CelesTrak: current supplemental GP element sets. URL https://celestrak.org/NORAD/elements/supplemental/.

Kelvins - collision avoidance challenge - data. URL https://kelvins.esa.int/collision-avoidance-challenge/data/.

Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization, May 2017. URL http://arxiv.org/abs/1705.10528. arXiv:1705.10528 [cs.LG].

Maruthi R. Akella and Kyle T. Alfriend. Probability of collision between space objects. *Journal Of Guidance, Control, And Dynamics*, 23(5):769–772, 2000. ISSN 0731-5090. doi: 10.2514/2.4611. URL https://doi.org/10.2514/2.4611.

Roberto Armellin. Collision avoidance maneuver optimization with a multiple-impulse convex formulation. *Acta Astronautica*, 186:347–362, September 2021. ISSN 0094-5765. doi: 10.1016/j.actaastro.2021.05.046. URL https://www.sciencedirect.com/science/article/pii/S0094576521002836.

Claudio Bombardelli and Javier Hernando-Ayuso. Optimal impulsive collision avoidance in low earth orbit. *Journal Of Guidance, Control, And Dynamics*, 38(2):217–225, February 2015. ISSN 0731-5090. doi: 10.2514/1.G000742. URL https://arc.aiaa.org/doi/10.2514/1.g000742.

Nicolas Bourriez, Adrien Loizeau, and Adam F. Abdin. Spacecraft autonomous decision-planning for collision avoidance: a reinforcement learning approach, October 2023. URL http://arxiv.org/abs/2310.18966. arXiv:2310.18966 [cs.RO].

Veronica L. Foreman, Afreen Siddiqi, and Olivier De Weck. Large satellite constellation orbital debris impacts: case studies of OneWeb and SpaceX proposals. In *AIAA SPACE and astronautics forum and exposition*. American Institute of Aeronautics and Astronautics. doi: 10.2514/6.2017-5200. URL https://arc.aiaa.org/doi/abs/10.2514/6.2017-5200.

Javier García, Fern, and O Fernández. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research*, 16(42):1437–1480, 2015. URL http://jmlr.org/papers/v16/garcia15a.html.

Sajjad Kazemi, Nasser L. Azad, K Andrea Scott, Haroon B. Oqab, and George B. Dietrich. Satellite collision avoidance maneuver planning in low earth orbit using proximal policy optimization. In *2024 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1–9, June 2024. doi: 10.1109/CEC60901.2024.10611892. URL https://ieeexplore.ieee.org/document/10611892.

Donald J. Kessler and Burton G. Cour-Palais. Collision frequency of artificial satellites: the creation of a debris belt. *Journal of Geophysical Research: Space Physics*, 83(A6):2637–2646, 1978. ISSN 2156-2202. doi: 10.1029/JA083iA06p02637. URL https://onlinelibrary.wiley.com/doi/abs/10.1029/JA083iA06p02637.

Frederic J. Krage. NASA spacecraft conjunction assessment and collision avoidance best practices handbook, volume 1. Technical report, National Aeronautics and Space Administration, January 2026. URL https://ntrs.nasa.gov/citations/20260000453. Number: CA2 Handbook Rev 2.

J.-C. Liou and N. L. Johnson. Risks in space from orbiting debris. *Science*, 311(5759):340–341, January 2006. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.1121337. URL https://www.science.org/doi/10.1126/science.1121337.

J. C. Liou and N. L. Johnson. Instability of the present LEO satellite populations. *Advances in Space Research*, 41(7):1046–1053, January 2008. ISSN 0273-1177. doi: 10.1016/j.asr.2007.04.081. URL https://www.sciencedirect.com/science/article/pii/S0273117707004097.

Chaoxu Mu, Shuo Liu, Ming Lu, Zhaoyang Liu, Lei Cui, and Ke Wang. Autonomous spacecraft collision avoidance with a variable number of space debris based on safe reinforcement learning. *Aerospace Science and Technology*, 149:109131, June 2024. ISSN 1270-9638. doi: 10.1016/j.ast.2024.109131. URL https://www.sciencedirect.com/science/article/pii/S1270963824002645.

Alexandre Oliveira, Katarina Dyreby, Francisco Caldas, and Cláudia Soares. OrbitZoo: real orbital systems challenges for reinforcement learning, October 2025. URL http://arxiv.org/abs/2504.04160. arXiv:2504.04160 [cs.LG].

Carmen Pardini and Luciano Anselmo. Environmental sustainability of large satellite constellations in low earth orbit. *Acta Astronautica*, 170:27–36, May 2020. ISSN 0094-5765. doi: 10.1016/j.actaastro.2020.01.016. URL https://www.sciencedirect.com/science/article/pii/S0094576520300187.

Russell P. Patera. General method for calculating satellite collision probability. *Journal Of Guidance, Control, And Dynamics*, 24(4):716–722, 2001. ISSN 0731-5090. doi: 10.2514/2.4771. URL https://doi.org/10.2514/2.4771.

Russell P. Patera and Glenn E. Peterson. Space vehicle maneuver method to lower collision risk to an acceptable level. *Journal Of Guidance, Control, And Dynamics*, 26(2):233–237, March 2003. ISSN 0731-5090. doi: 10.2514/2.5063. URL https://arc.aiaa.org/doi/10.2514/2.5063.

Zeno Pavanello, Luigi De Maria, Andrea De Vittori, Michele Maestrini, Pierluigi Di Lizia, and Roberto Armellin. CAMmary: a review of spacecraft collision avoidance manoeuvre design methods. *Acta Astronautica*, 236:770–789, November 2025. ISSN 0094-5765. doi: 10.1016/j.actaastro.2025.07.027. URL https://www.sciencedirect.com/science/article/pii/S0094576525004576.

Stephane Ross, Geoffrey J. Gordon, and J. Andrew Bagnell. A reduction of imitation learning and structured prediction to No-regret online learning, March 2011. URL http://arxiv.org/abs/1011.0686. arXiv:1011.0686 [cs.LG].

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, August 2017. URL http://arxiv.org/abs/1707.06347. arXiv:1707.06347 [cs.LG].

Alexandru Solomon and Ciprian Paduraru. Collision avoidance and return manoeuvre optimisation for low-thrust satellites using reinforcement learning. volume 2, pp. 1009–1016. SCITEPRESS, February 2025. ISBN 978-989-758-737-5. doi: 10.5220/0013249000003890. URL http://www.scitepress.org/Papers/2025/132490.

Thomas Uriot, Dario Izzo, Luís F. Simões, Rasit Abay, Nils Einecke, Sven Rebhan, Jose Martinez-Heras, Francesca Letizia, Jan Siminski, and Klaus Merz. Spacecraft collision avoidance challenge: design and results of a machine learning competition. *Astrodynamics*, 6(2):121–140, June 2022. ISSN 2522-0098. doi: 10.1007/s42064-021-0101-5. URL https://doi.org/10.1007/s42064-021-0101-5.

David Vallado, Paul Crawford, Ricahrd Hujsak, and T.s. Kelso. Revisiting spacetrack report #3. In *AIAA/AAS astrodynamics specialist conference and exhibit*, Guidance, Navigation, and Control and Co-located Conferences. American Institute of Aeronautics and Astronautics, August 2006. doi: 10.2514/6.2006-6753. URL https://arc.aiaa.org/doi/10.2514/6.2006-6753.

A IMPLEMENTATION AND PROTOCOL DETAILS

This appendix records implementation details, evaluation protocol, calibration, and secondary results that support the main text. It introduces no new experiment.

A.1 WARNING PIPELINE AND MEASURED MISMATCH

At each control step, the warning pipeline receives the current epoch and ECI states for the controlled spacecraft and candidate objects. It converts ECI to TEME using IAU-76 precession and the dominant IAU-80 nutation terms. A nonsingular initial representation $[a, e_x, e_y, h_x, h_y, \lambda]$, with explicit handling of circular, equatorial, and circular–equatorial cases, is converted to the classical elements required by SGP4. Two fixed-point inverse-SGP4 iterations compare the reconstructed state with the input TEME state, wrap angular residuals, and correct the elements. The reconstructed objects are then propagated over the fixed warning window and converted back to ECI.

Coarse screening samples every second propagation point, explicitly includes the final point, and retains pairs whose sampled separation is at most 500 km. Only upper-triangular pairs are processed. Full-grid screening keeps every local minimum, rather than only the global minimum, so one pair may produce multiple events in a warning window. Interior minima are refined by quadratic interpolation of three neighboring squared separations and retained inside an RTN box with half-widths 2, 44, and 51 km in radial, along-track, and cross-track directions. The box test uses each object in turn as the RTN reference to reduce ordering dependence.

The encounter-plane calculation uses the supplied covariance plus a look-ahead-dependent model-error uncertainty term measured from the discrepancy below. The prediction-error term is the pooled RTN second-moment matrix $\mathbf{Q}(\tau)$ at look-ahead time τ. The two object-centered matrices are rotated separately to ECI and summed; this model-error uncertainty term is then added to the supplied position covariances. The profile is indexed by look-ahead time, with linear interpolation between recorded times, before encounter-plane projection. The projected covariance is transformed to principal-axis coordinates and the Alfano integration order is selected adaptively for numerical stability.

Table 4: Position error after two inverse-SGP4 iterations (km).

Horizon	p50	p90	p95	p99	Max
0 min	0.0033	0.0077	0.0094	0.0134	0.0301
15 min	0.0266	0.0429	0.0484	0.0583	0.0787
30 min	0.0839	0.1462	0.1647	0.1940	0.2790
60 min	0.2722	0.4895	0.5399	0.6361	0.9060
90 min	0.3490	0.6282	0.6940	0.8523	1.0596
120 min	0.3886	0.7062	0.7847	0.9260	1.2678

Table 5: Velocity error after two inverse-SGP4 iterations (m/s).

Horizon	p50	p90	p95	p99	Max
0 min	0.0037	0.0086	0.0105	0.0149	0.0329
15 min	0.0531	0.0899	0.1032	0.1269	0.1751
30 min	0.1049	0.1842	0.2033	0.2433	0.3578
60 min	0.2838	0.5056	0.5591	0.6654	0.9011
90 min	0.3757	0.6737	0.7407	0.9053	1.1584
120 min	0.4644	0.8307	0.9175	1.0989	1.5269

When several events exceed $\widetilde{P}_{\mathrm{c}}$, the warning summary selects the earliest predicted TCA, then decreasing collision probability and object ID as deterministic tie-breaks. Table 6 reports the complete pipeline, including reconstruction, propagation, screening, refinement, and risk evaluation.

Table 6: Warning-pipeline runtime on real Starlink catalog states.

Objects	Pairs	Mean (s)	p50 (s)	p95 (s)
1	0	0.0110	0.0110	0.0113
10	45	0.0122	0.0122	0.0124
100	4,950	0.0186	0.0186	0.0187
1,000	499,500	0.0301	0.0298	0.0313
10,000	49,995,000	0.5545	0.5544	0.5548

A.2 Numerical execution and closed-loop update

The executor clips the normalized RTN command and maps it to a finite-burn acceleration under the propulsion constraints. For state $\mathbf{x} = [\mathbf{r}^\top, \mathbf{v}^\top]^\top$, the numerical truth model is

$$\dot{\mathbf{x}} = f(t, \mathbf{x}, \mathbf{a}^{\mathrm{RTN}}) = \begin{bmatrix} \mathbf{v} \\ \mathbf{a}_{\mathrm{grav}} + \mathbf{a}_{J_2} + \mathbf{a}_{\mathrm{thrust}} \end{bmatrix}, \tag{17}$$

where the RTN thrust is transformed to ECI before evaluating the dynamics. Fourth-order Runge–Kutta integration advances every object by one 30-second control interval. The realized next state, rather than an independently continued prediction, reinitializes the warning model. The resulting update is

$$\mathbf{s}_t \to \widehat{P}_{c,t} \to \widehat{\mathbf{a}}_t \to \mathbf{a}_t \to \mathbf{s}_{t+1},$$

followed by fresh reconstruction, propagation, screening, and risk evaluation. Thus post-burn warnings and target changes enter the next policy observation.

A.3 Scenario generation and acceptance

Ego state and encounter geometry. The ego state is sampled from Starlink Supplemental GP data, including epoch, ECI position and velocity, and position covariance. Invalid uncertainty or states outside the intended low-Earth-orbit regime are removed. A target TCA is sampled on the discrete control grid and the ego spacecraft is propagated to that epoch. An ESA encounter template supplies relative miss direction, relative-velocity direction, miss distance, and relative speed in RTN. The template is instantiated around the propagated ego state, transformed to ECI, and both objects are propagated backward to the episode start. Small joint rotations of the miss and relative-velocity directions and bounded perturbation of relative speed preserve the historical encounter structure while increasing diversity. Scaling the template miss distance proposes hazardous or safe risk levels; these labels remain provisional until both models are evaluated.

Validity, consistency, and split integrity. Candidates are rejected for non-finite states, implausible altitude or velocity, non-elliptic energy, excessive eccentricity, or perigee, apogee, and period outside the benchmark regime. A hazardous candidate is accepted only when both the numerical no-maneuver trajectory and the warning model exceed $\widetilde{P}_{\mathrm{c}}$ for the original threat; a safe candidate is accepted only when both remain below threshold. Exact TCA and probability agreement is not required. Accepted cases are deduplicated by scenario identity, training and test splits are disjoint, and Test128 cases are excluded from behavior-cloning demonstrations.

Multi-object construction. For safe scenes, catalog background objects are propagated to the base epoch and must pass orbital-validity and minimum-separation checks without creating an initial warning. For hazardous scenes, the initial warning must still select the original threat. Additional local objects are generated through bounded perturbations of the ego orbit and checked against the ego spacecraft, original threat, and all backgrounds. During dataset construction only, fixed-magnitude probes in six RTN directions test whether the selected warning can switch to a distinct local object above $\widetilde{P}_{\mathrm{c}}$. These probes are acceptance tests, not policy actions, teacher actions, demonstrations, or evaluation maneuvers.

A.4 Observation and teacher construction

The 20-dimensional policy observation contains the conjunction object's RTN relative position and velocity; current and previous warning indicators; current and previous collision-risk levels and

their temporal difference; TCA; cumulative maneuver; elapsed episode time; and RTN position and velocity deviation from the designated orbit. Collision probabilities are represented as log ratios to $\widetilde{P}_{\mathrm{c}}$. Continuous quantities use fixed physical scales and every feature is clipped to a bounded range.

The offline teacher first replays the no-op plan. A safe rollout is retained as the demonstration; an unsafe rollout triggers search over plans with at most two finite-burn intervals. Candidate start times lie on a grid tied to the first warning, predicted TCA, and episode horizon, while durations and bounded RTN commands are enumerated from predefined candidate sets. Promising plans are replayed in the full closed loop, ranked by safety, orbital recovery, and cumulative ΔV, and optionally refined locally. Only replay-safe plans that pass consistency checks become demonstrations.

A.5 Training configuration

Table 7: Principal SCA-Planner hyperparameters.

Item	Value
Episode horizon / interval / steps	7,200 s / 30 s / 240
Warning threshold / update	3×10^{-7} / every step
Observation / architecture	20-D / A1.5 recurrent Gate actor
Behavior-cloning demonstrations	1,352
PPO scenario pool	986 initial-warning cases
Parallel environments / rollout length	16 / 240 steps
PPO epochs / optimizer / clip / target KL	3 / Adam / 0.1 / 0.003
Discount γ / GAE λ	0.997 / 0.98
Gate / action / critic learning rates	5×10^{-7} / 5×10^{-6} / 3×10^{-5}
Gate freeze	first 5×10^5 environment steps
Evaluation	deterministic, one episode per Test128 case

A.6 Behavior-cloning and PPO objectives

For demonstration steps t, let $b_t^\star$ be the teacher's maneuver label and $\mathbf{a}_t^\star$ its command; $\mathcal{T}^+$ and $\mathcal{T}^-$ contain maneuver and no-maneuver steps. The class-balanced Gate loss is

$$\mathcal{L}_{\mathrm{gate}} = -\frac{1}{|\mathcal{T}^+|}\sum_{t\in\mathcal{T}^+}\log p_t^g - \frac{1}{|\mathcal{T}^-|}\sum_{t\in\mathcal{T}^-}\log(1-p_t^g). \tag{18}$$

For $\mathbf{z}_t^\star = \operatorname{atanh}(\mathbf{a}_t^\star)$, the action branch losses on maneuver steps are

$$\mathcal{L}_{\mathrm{action}} = \frac{1}{|\mathcal{T}^+|}\sum_{t\in\mathcal{T}^+}\mathrm{SmoothL1}(\mathbf{z}_t, \mathbf{z}_t^\star), \tag{19}$$

$$\mathcal{L}_{\mathrm{exec}} = \frac{1}{|\mathcal{T}^+|}\sum_{t\in\mathcal{T}^+}\|\tanh(\mathbf{z}_t) - \mathbf{a}_t^\star\|_2^2. \tag{20}$$

PPO uses the hybrid likelihood

$$\pi(b_t, \mathbf{z}_t \mid \mathbf{o}_{1:t}) = \pi^g(b_t \mid \mathbf{o}_{1:t})\left[\pi^a(\mathbf{z}_t \mid \mathbf{o}_t)\right]^{b_t}. \tag{21}$$

A.7 Reward coefficients

Let $x_t = \ln(\widehat{P}_{\mathrm{c}}(t)/\widetilde{P}_{\mathrm{c}})$ and ΔV_t be cumulative delivered maneuver. The hazard-active risk reward and incremental maneuver rewards are

$$R_t^{\mathrm{risk}} = x_{t-1} - \gamma x_t - \lambda^{\mathrm{risk}} x_t, \tag{22}$$

$$R_t^{\Delta V} = \Delta V_{t-1} - \Delta V_t, \tag{23}$$

$$R_t^{\mathrm{budget}} = [\Delta V_{t-1}/\widetilde{\Delta V} - 1]_+^2 - [\Delta V_t/\widetilde{\Delta V} - 1]_+^2. \tag{24}$$

With time-indexed recovery score ORS_t, the recovery-phase reward is

$$R_t^{\mathrm{rec}} = \gamma \mathrm{ORS}_t - \mathrm{ORS}_{t-1} - \lambda^{\mathrm{rec}}\left[\frac{\widetilde{\mathrm{ORS}} - \mathrm{ORS}_t}{\widetilde{\mathrm{ORS}}}\right]_+^2. \tag{25}$$

For closest-event probability $P_c^\star$, the terminal reward is

$$R_T^{\mathrm{term}} = \begin{cases} -\widetilde{R}_1, & P_c^\star > \widetilde{P}_c^{\mathrm{col}}, \\ -\widetilde{R}_2 - \lambda^{\mathrm{term}} \ln(P_c^\star / \widetilde{P}_{\mathrm{c}}), & \widetilde{P}_{\mathrm{c}} < P_c^\star \le \widetilde{P}_c^{\mathrm{col}}, \\ \widetilde{R}_3, & P_c^\star \le \widetilde{P}_{\mathrm{c}}. \end{cases} \tag{26}$$

Table 8: Reference reward coefficients.

Component	Value	Component	Value
Risk progress	20	Residual risk	4
Hazard ΔV	0	Confirming ΔV	0.5
Recovery ΔV	2	Soft ΔV budget	0.30 m/s
Soft budget weight	8	Recovery progress	30
Post-TCA recovery deficit	120	Safe terminal bonus	60
Terminal truth-risk weight	40	Terminal recovery	60
Terminal recovery deficit	100	Unsafe terminal penalty	-160
Collision terminal penalty	-220	Terminal ΔV budget	0.60 m/s
Terminal ΔV coefficient	5		

A.8 Baseline execution and runtime

All optimization baselines output commands that are executed by the same finite-burn environment. Their CPCR, CME, and ORS are therefore computed from realized numerical trajectories rather than directly from internal prediction models. No-op applies the zero command throughout the episode.

The Armellin baseline plans candidate impulses at fixed maneuver nodes using a linearized, budget-aware multi-impulse formulation; its selected impulse sequence is converted to finite-burn commands. The Patera baseline searches candidate maneuver times and magnitudes until the predicted post-maneuver risk satisfies $\widetilde{P}_{\mathrm{c}}$, then passes the selected maneuver to the same executor. The Bombardelli baseline uses a B-plane-style geometric search: it selects candidate epochs, estimates local encounter-geometry sensitivity by finite differences, and converts the selected correction to a finite burn. The Alfano baseline enumerates finite-burn candidate plans and ranks them with the causal warning model, including SGP4 prediction of the post-maneuver conjunction, before execution in the benchmark environment.

OrbitZoo uses a feed-forward Gaussian policy in which one output controls burn activation and the remaining outputs parameterize thrust. Its reward scale, decision horizon, and safety threshold are adapted to SCA-Bench, and the learned policy is evaluated deterministically with the same finite-burn numerical execution as SCA-Planner. Optimization-baseline runtime includes maneuver planning and environment execution. The learned policies incur their principal cost offline and require network inference during evaluation.

Table 1 reports the measured evaluation runtimes. Learned- policy times include inference and execution but not training, so they are not training-cost comparisons.

B Additional ablations

The main paper reports the learning-pipeline, Gate, and objective-routing ablations. This appendix retains a secondary recovery-oriented operating point.

Table 9: Recovery-oriented operating points on Test128.

Method	CPCR (%)	CME (m/s)	ORS	CCAS
BC only	75.00	0.1632	0.6154	0.3322
Recovery-favored PPO	77.34	0.5476	0.6533	0.2047
SCA-Planner	83.59	0.1850	0.5800	0.3482

The recovery-favored checkpoint improves both CPCR and ORS over BC, but expends 0.5476 m/s and lowers CCAS to 0.2047. It shows that PPO can move toward a recovery-favored operating point; it does not remove the trade-off.

Relative to the reference configuration, this variant removes the initial Gate-freezing stage, increases the actor-side learning rates, raises the recovery target from 0.67 to 0.77, and doubles the recovery reward weight. Its higher ORS is therefore an objective-selected operating point rather than a new architecture.

With random initialization and the standard deterministic Gate threshold, PPO converges to the no-op-like point reported in the main ablation. Lowering the threshold raises CPCR to 80.47% but activates burns during 41.80% of control steps, averaging 100.31 burns per episode and 6.6219 m/s CME. This distinguishes failure to activate from learning a sparse, coordinated intervention policy.

Table 10: Behavior-cloning activation-function ablation.

Activation	CPCR (%)	CME (m/s)	ORS	CCAS
Tanh	86.67	1.7128	0.4377	0.0974
ReLU	93.33	1.3127	0.4468	0.1290

This behavior-cloning-stage comparison is separate from the final PPO evaluation. ReLU improves all four reported quantities at this stage, including a 0.4001 m/s reduction in CME, and is therefore used in the cloned actor that initializes PPO.

C CCAS CALIBRATION

The policy-independent calibration is a head-on encounter with an 800 m miss distance along RTN normal and TCA at the midpoint of the two-hour episode. The spacecraft performs one transverse finite burn at step 0 and then no-ops. The no-op rollout has realized $P_{\mathrm{c}} = 1.7912 \times 10^{-6}$. The lowest-magnitude compliant burn found by the reported search has $P_{\mathrm{c}} = 2.9285 \times 10^{-7}$, below $\widetilde{P}_{\mathrm{c}} = 3 \times 10^{-7}$, with $v^\star \approx 0.24$ m/s and terminal $r^\star \approx 0.34$. The search retains the lowest compliant evaluated burn because realized P_{c} is not strictly monotonic in action magnitude. These constants scale one representative unsafe-to-safe transition; they do not imply that CPCR, recovery loss, and CME have identical operational value.

The search evaluates a coarse magnitude grid and then refines each observed unsafe-to-safe bracket. It uses the same truth propagation, finite-burn model, closest-approach refinement, realized collision probability, and terminal ORS as benchmark evaluation. If the other $N - 1$ episodes are held fixed, replacing one no-op outcome by this calibrated compliant maneuver changes CPCR by $1/N$ and contributes

$$\frac{(1 - r^\star)/N}{1 - r^\star} = \frac{v^\star/N}{v^\star} = \frac{1}{N} \tag{27}$$

to each normalized denominator term. The calibration aligns numerical scales; it does not equate their operational value or marginal influence. Because the CCAS denominator is at least one, CCAS is upper-bounded by CPCR.

D LIMITATIONS

SCA-Bench currently adopts a controlled evaluation setting with one controlled spacecraft, prescribed surrounding-object trajectories, and a fixed two-hour horizon, enabling consistent comparison of closed-loop controllers under a common protocol. Natural extensions include richer uncertainty models, longer-horizon conjunction sequences, and coordinated multi-spacecraft interactions. Learned policies are evaluated deterministically on the fixed Test128 suite; broader evaluation across training seeds and scenario variations is left for future work.