HAD^{2}RL: Heterogeneous-Agent Dense Reinforcement Learning for Adversarial Failure Search of Detect-and-Avoid Systems

Anonymous authors
Paper under double-blind review

Abstract

Low-altitude unmanned aircraft are being deployed at scale for logistics and urban air mobility, and their detect-and-avoid (DAA) systems are designed to keep them separated from other aircraft and prevent near mid-air collisions (NMACs). Validating DAA safety relies on simulation, but NMACs are extremely rare, so conventional validation requires expert-designed test scenarios and large-scale simulation before failures are observed, which is costly. Recent work uses adversarial reinforcement learning to search for critical failures, but the sparse learning signal is often drowned out by gradient noise from the many non-critical states. Moreover, DAA failures typically arise from heterogeneous factors acting jointly, such as other aircraft and the environment, whose critical moments differ in time, so the search space grows and failures become sparser still. We therefore propose Heterogeneous-Agent Dense Reinforcement Learning (HAD^{2}RL). HAD^{2}RL estimates, for each agent, how much each state matters from the value of the failure event. It then replaces the hard mask with long-tailed weighting and GAE reconnection, so that credit follows this criticality and misjudged states are down-weighted rather than discarded. We prove that soft densification introduces only a bounded bias into the expected policy gradient while reducing gradient noise by one to two orders of magnitude. To evaluate HAD^{2}RL, we build DAABench, the first benchmark for adversarial failure search on rule-based DAA systems. On its 11 encounter families, HAD^{2}RL raises the NMAC discovery rate of three base algorithms by 15.4% on average and by up to 28$\times$ in rare families, and uncovers three previously unknown failure modes.

1 Introduction

Low-altitude unmanned aircraft are being deployed at scale for logistics, inspection, and urban air mobility, and their detect-and-avoid (DAA) systems are designed to keep them separated from other aircraft and prevent near mid-air collisions (NMACs). DAA safety is validated through simulation-based testing, in which encounters are simulated and the aircraft is checked for NMACs. NMACs are extremely rare, and the avoidance logic itself suppresses them further. Conventional validation therefore relies on safety experts to design test scenarios and on large-scale simulation to expose even a single failure, at substantial cost (RTCA, 2017; Kochenderfer et al., 2010; Liu & Feng, 2024; O'Kelly et al., 2018).

Adversarial failure search replaces exhaustive simulation with an adversarial agent trained to induce failures. Adaptive stress testing (AST), for example, formulates failure search as a Markov decision process and has evolved from tree search to deep reinforcement learning, amortizing per-scenario search into a reusable policy (Lee et al., 2015; 2020; Lipkis et al., 2022), with parallel progress in autonomous driving (Feng et al., 2023; Nie et al., 2026). The learning signal, however, is sparse. The informative reward appears only when an NMAC occurs, and the vast majority of steps in an episode are non-critical states that contribute nothing to the expected policy gradient but account for most of its variance (Feng et al., 2023). Dense reward shaping is the common remedy, but it requires careful tuning, spreads credit over non-critical steps, and optimizes a proxy of the true objective, which invites over-adversarial behavior that trades plausibility for failures (Ng et al., 1999; Lipkis

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.

et al., 2022; Nie et al., 2026). Dense reinforcement learning (D2RL) instead learns only at critical states, removing this noise without introducing gradient bias (Feng et al., 2023; 2026).

Failure search for DAA is harder still. An NMAC is typically induced jointly by heterogeneous factors such as other aircraft and the external environment, so exposing closed-loop blind spots requires two heterogeneous adversarial agents, an intruder that steers the encounter geometry and a perturber that corrupts what the DAA system perceives. The joint action space grows while NMACs remain rare, and the failure signal becomes even sparser (Liu & Feng, 2024). Dense reinforcement learning is the natural remedy, but existing methods are designed for a single agent and show two limitations when carried over to heterogeneous agents (Feng et al., 2023; 2026). **(1) Temporal criticality differs across agents but is treated as shared.** D2RL, for example, defines a state as critical if some action can change the event value, without distinguishing which agent the action comes from, so all agents share a single critical set. The intruder, however, is decisive only in a short window before the closest approach, whereas the perturber must act early to mislead the alerting logic. A shared critical set forces each agent to learn from states that are critical only for the other, so the noise of those states remains in its gradient while the credit for its own critical moments is diluted. **(2) Critical states differ in causal contribution but are treated alike.** The importance of a critical state depends on how much its action changes the probability of failure. The intruder's last few steps before the closest approach, for example, largely determine the miss distance, whereas an early deviation can still be corrected by the avoidance logic. Existing hard densification with a 0/1 mask treats all critical states equally and removes the rest, so the few steps that decide the failure receive no more gradient than any other critical step, and states misjudged as non-critical no longer contribute to the gradient at all.

To address these limitations, we propose Heterogeneous-Agent Dense Reinforcement Learning (HAD^{2}RL) and redefine criticality for heterogeneous agents. First, we define an event value for each agent, namely the value of the NMAC event under that agent's own action while the other agent follows its policy. The magnitude of the corresponding event advantage serves as the agent's continuous per-step criticality and is estimated by an event critic trained only on the NMAC indicator. Second, we propose soft densification, which multiplies this criticality, the causal contribution of an action to failure, into each agent's advantage through long-tailed weighting with a positive floor, so that misjudged states are kept, and cuts far-field bootstrap noise through GAE reconnection. To evaluate HAD^{2}RL, we further build DAABench, which wraps NASA DAIDALUS, the rule-based reference implementation of the DAA standard (Consiglio et al., 2015; RTCA, 2017), in closed loop and lets an intruder and a perturber attack simultaneously as two heterogeneous adversarial agents. Our main contributions are as follows:

- We define heterogeneous criticality for multiple agents, using per-agent event values to capture how criticality differs across agents and over time.
- We propose soft densification, which replaces the hard mask with long-tailed weighting and GAE reconnection, and prove that it has bounded gradient bias and one to two orders of magnitude less gradient noise.
- We build DAABench, the first benchmark for adversarial failure search on rule-based DAA systems, which stratifies encounter scenarios into 11 families by failure rarity.
- On DAABench, HAD^{2}RL raises the NMAC discovery rate of baselines by 15.4% on average and by up to $28\times$ in rare families, and uncovers three previously unknown failure modes.

2 Related Work

Failure search for collision avoidance systems. Formal verification certifies reachability and produces counterexamples for neural controllers such as ACAS Xu (Wu et al., 2024; Shi et al., 2025; Brix et al., 2024), but it does not cover rule-based closed-loop systems with hysteresis and memory such as DAIDALUS. Among simulation-based methods (Corso et al., 2021), Monte Carlo and importance sampling estimate failure probabilities (Kochenderfer et al., 2010; Zhao et al., 2016; O'Kelly et al., 2018), and archive-based search such as Go-Explore concentrates computation on high-risk regions (Ecoffet et al., 2021), but both optimize per instance. Adaptive stress testing (AST) amortizes the search into a reusable policy and has evolved in airborne collision avoidance from tree search (Lee et al., 2015) to differential testing (Lee et al., 2018), deep reinforcement learning (Lipkis et al., 2022), and unbiased estimation (Lipkis & Agogino, 2023). In driving, FREA bounds

the adversarial objective by the largest feasible region to curb over-adversarial and unavoidable collisions (Chen et al., 2024). All of these pit a single-agent or joint-action policy against the system under test, and none releases a benchmark with fixed tasks, metrics, and baselines for DAA. The closest benchmark, AVOIDDS, evaluates vision-based intruder detection in closed loop but performs no adversarial search (Smyers et al., 2023). DAABench fills this gap and casts failure search as a multi-agent problem over a heterogeneous dual attack surface.

Dense reinforcement learning and rare-event learning. The extremely low probability of safety-critical events leaves training data dominated by uninformative states, a fundamental barrier to sample efficiency (Liu & Feng, 2024). Credit assignment under sparse rewards has been approached from several angles. Potential-based reward shaping preserves the optimal policy but not the variance structure of the gradient (Ng et al., 1999), hindsight experience replay relies on goal relabeling (Andrychowicz et al., 2017), and return decomposition requires additional sequence modeling (Arjona-Medina et al., 2019). None delivers a credit estimate focused on safety-critical states. D2RL edits the Markov decision process to learn only at critical states, lowering policy-gradient variance by orders of magnitude without loss of unbiasedness (Feng et al., 2023), and has been extended to closed-loop training (Feng et al., 2026). Coupled with naturalistic behavior modeling, it approaches rare failures under plausibility constraints (Feng et al., 2021; Yan et al., 2023). DMARL writes only critical states into a dense buffer under CTDE, but defines criticality by a communication-coverage task and hard-filters the rest, addressing neither failure search nor heterogeneous criticality (Fu et al., 2024). These schemes either target a single agent with a binary hard mask or use buffer filtering to accelerate coverage optimization. We instead generalize densification to heterogeneous multi-agent failure search, defining criticality per agent and replacing the hard mask with soft densification that weights critical states by their causal contribution under a bounded bias.

Heterogeneous multi-agent reinforcement learning. Multi-agent RL has matured around centralized training with decentralized execution (CTDE) (Lowe et al., 2017). MAPPO extends PPO to the shared-return setting (Yu et al., 2022), and HATRPO and HAPPO add sequential updates and heterogeneous trust regions with monotonic-improvement guarantees (Kuba et al., 2022; Zhong et al., 2024). All impose homogeneous credit under a shared return, neither identifying safety-critical states nor distinguishing them per agent, so the sparse failure signal cannot be attributed to the temporally misaligned critical decisions of different agents. We model the intruder and the perturber as heterogeneous adversarial agents with distinct action spaces and failure pathways, and embed soft densification into this family. The weights act only on advantage estimation and leave the clipping or trust-region surrogate intact, densifying credit while preserving the update geometry.

3 Heterogeneous-Agent Dense Reinforcement Learning

HAD^{2}RL densifies learning for heterogeneous adversarial agents through two parts (Fig. 1). Heterogeneous criticality scores how much each state matters to each agent at each step, estimated by an event critic trained on the NMAC indicator alone; soft densification turns the hard mask into a long-tailed weight on the dense advantage and reconnects the GAE chain across inert steps. Both act on the advantage, so HAD^{2}RL plugs into any heterogeneous MARL algorithm without altering its objective. §3.1 formulates the problem, §3.2 and §3.3 present the two modules and the training pipeline, and §3.4 characterizes the bias incurred and the noise removed.

3.1 Problem Formulation

Safety validation searches for scenarios in which the system under test (SUT), here NASA DAIDALUS under DO-365, fails by an NMAC. We close the loop around DAIDALUS and train two heterogeneous adversaries to find the most likely NMAC within a natural-behavior envelope.

Heterogeneous adversarial Markov game. We model failure search as a Markov game with two heterogeneous agents $i \in \{\mathrm{I}, \mathrm{P}\}$:

$$\mathcal{M} = \big(\mathcal{S}, \{\mathcal{A}^i\}_i,\, P,\, r,\, \gamma,\, \rho_0\big). \tag{1}$$

The intruder I reshapes the conflict geometry through kinematic actions, and the perturber P corrupts DAIDALUS's sensor inputs and applies wind to the ownship. Their safety-critical moments are misaligned in time, since the intruder acts in a narrow window near the closest point of approach

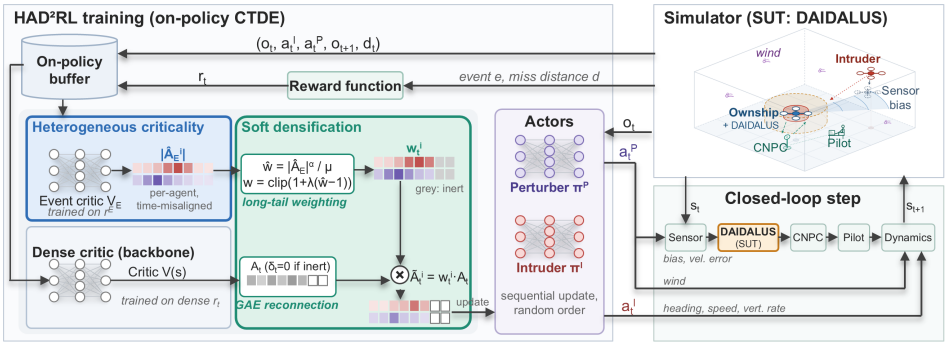

Figure 1: Overview of HAD^{2}RL. **Left:** CTDE training, with the two contributions boxed. An event critic V_E on r^E yields per-agent criticality $|\hat{A}_E^i|$, misaligned across agents, and a long-tailed weight rescales the reconnected advantage, $\tilde{A}_t^i = w_t^i A_t$. **Right:** the DAIDALUS closed loop, where the perturber a_t^P injects sensor bias and wind and the intruder a_t^I sets heading, speed, and vertical rate.

(CPA) whereas the perturber must seed the alerting chain well in advance. Both act on local observations under CTDE with joint policy $\pi = (\pi^{\mathrm{I}}, \pi^{\mathrm{P}})$ (Appendix B.1).

Failure objective under the natural-behavior envelope. The adversary maximizes DAIDALUS's failure probability subject to a natural-behavior envelope:

$$\max_{\pi} \Pr_{\tau\sim\pi}\big(\mathrm{NMAC}(\tau)\big) \quad \text{s.t.} \quad \tau \text{ within the natural-behavior envelope.} \tag{2}$$

An NMAC occurs when horizontal and vertical separations fall below 500 ft and 100 ft (DO-365). The envelope is a per-agent diagonal-Gaussian action prior p_0^i (Koren et al., 2018; Lee et al., 2020), and naturalness is measured by the relative log-likelihood (RLL) of

$$\mathrm{RLL}^i(\tau) = \sum_{t=1}^{T}\big(\log p_0^i(a_t^i) - \mathbb{E}_{p_0^i}[\log p_0^i]\big) = \sum_{t=1}^{T}\Big(\frac{d^i}{2} - \frac{1}{2}\sum_j (a_{t,j}^i/\sigma_j)^2\Big). \tag{3}$$

We train with a likelihood penalty and report $\Pr(\mathrm{NMAC})$ and RLL (Appendix B.4).

3.2 Heterogeneous Criticality

This subsection defines heterogeneous criticality, how much each state matters to each agent at each step, and §3.3 concentrates the learning budget accordingly.

Criticality for a single agent. D2RL (Feng et al., 2023) calls a state s *uncritical* if no action alters its event value, $q_\pi(s,a) = v_\pi(s)$ for all a, and *safety-critical* otherwise, $\mathcal{S}_c = \{s : q_\pi(s,\cdot) \not\equiv v_\pi(s)\}$. A global 0/1 mask on $\mathcal{S}_c$ reduces policy-gradient variance by a factor of $1/\rho_c$ (ρ_c: critical-state fraction) without introducing bias. Because this criterion ignores whose action changes the event value, we define the event value per agent.

Per-agent event value. The event value is defined solely with respect to the event reward

$$r_t^E = R_E \cdot 1\,[\text{NMAC occurs at } t]\,, \tag{4}$$

that is, an MDP carrying only the NMAC indicator. Let Q_π^{event} be the joint action-value under r^E alone. For agent i, its event action-value and state-value marginalize over the other agent's policy π^{-i}:

$$q^i_{\mathrm{event},\pi}(s,a^i) = \mathbb{E}_{a^{-i}\sim\pi^{-i}}\big[Q_\pi^{\mathrm{event}}(s,a^i,a^{-i})\big], \qquad v^i_{\mathrm{event},\pi}(s) = \mathbb{E}_{a^i\sim\pi^i}\big[q^i_{\mathrm{event},\pi}(s,a^i)\big]. \tag{5}$$

The event advantage $A^i_{\mathrm{event},\pi}(s,a^i) := q^i_{\mathrm{event},\pi}(s,a^i) - v^i_{\mathrm{event},\pi}(s)$ is the change in the NMAC event value caused by agent i's own action while the other agent follows its policy, that is, the causal contribution of the action to failure.

Definition 1 (Per-agent safety-critical state set). A state s is **non-critical** for agent i iff $A^i_{\mathrm{event},\pi}(s,\cdot) \equiv 0$, that is, no action of i changes its event value. Other states are **safety-critical** for

i and form

$$\mathcal{S}_c^i := \{\, s \in \mathcal{S} :\ A^i_{\text{event},\pi}(s,\cdot) \not\equiv 0 \,\}. \tag{6}$$

The two agents own separate, generally misaligned critical sets $\mathcal{S}_c^{\mathrm{I}}$ and $\mathcal{S}_c^{\mathrm{P}}$, and with a single agent Definition 1 recovers $\mathcal{S}_c$ of D2RL. Each $\mathcal{S}_c^i$ is induced by π and evolves during training, so Definition 1 serves the analysis and a learnable proxy replaces it in practice.

Event critic. The dense advantage $\hat{A}^i_{\text{dense}}$ sets the update direction, and $|\hat{A}^i_{\text{event}}|$ sets where it is strengthened. An event critic trained on r^E alone (Eq. 4) with standard GAE (Schulman et al., 2016) provides $\hat{A}^i_{\text{event}}$ (Appendix A.1). Because r^E is nonzero only at the terminal step, $|\hat{A}^i_{\text{event},t}|$ vanishes in far-field cruise and peaks at critical maneuvers, which makes it a learnable per-agent proxy for $\mathcal{S}_c^i$. The proxy is one-sided ($|\hat{A}^i_{\text{event}}| \neq 0 \Rightarrow s \in \mathcal{S}_c^i$, not conversely), and misses are retained by the floor $w_{\min} > 0$ (Proposition 1).

3.3 Soft Densification

Soft densification replaces the hard mask with two operators on the advantage. Long-tailed weighting injects criticality into the update, and GAE reconnection removes far-field bootstrap noise.

Long-tailed weighting. From the event-advantage magnitude we build a long-tailed weight and multiply it into the direction-carrying dense advantage:

$$w_t^i = \mathrm{clip}\big(1 + \lambda(\hat{w}_t^i - 1),\, w_{\min},\, w_{\max}\big), \qquad \hat{w}_t^i = \frac{|\hat{A}^i_{\text{event},t}|^\alpha}{\mu^i + \epsilon}. \tag{7}$$

$$\bar{A}^i_{\text{dense},t} = \frac{\hat{A}^i_{\text{dense},t} - \bar{\mu}_A}{\bar{\sigma}_A + \epsilon}, \qquad \tilde{A}_t^i = \bar{A}^i_{\text{dense},t} \cdot w_t^i. \tag{8}$$

In Eq. 8, $\hat{A}^i_{\text{dense}}$ sets the sign and w^i the magnitude, so critical states are weighted by their contribution to failure rather than kept or dropped. The floor $w_{\min} > 0$ down-weights misjudged states instead of discarding them, $w^i \in \{0,1\}$ recovers the hard mask of D2RL, and $\lambda \in [0,1]$ tunes from uniform weights ($\lambda = 0$) to the full long tail ($\lambda = 1$, Appendix C.2).

GAE reconnection. Under sparse rewards, bootstrapping carries noisy far-field values into critical returns. Following the edited Markov process of D2RL (Feng et al., 2023), we zero δ at *inert* steps (past the CPA with no alert, Appendix C.3), reconnecting the GAE chain, while the critic still trains on standard returns:

$$\hat{A}^i_{\text{dense},t} = \sum_{l \geq 0} (\gamma\lambda_{\text{gae}})^l \left(1 - m^{\text{inert}}_{t+l}\right) \delta^i_{t+l}, \qquad \delta_t^i = r_t + \gamma V^i_{\text{dense}}(s_{t+1}) - V^i_{\text{dense}}(s_t). \tag{9}$$

Eq. 9 omits the episode-termination mask and is conditionally unbiased (Proposition 3). Reconnection is enabled only when $\lambda > 0$ and otherwise falls back to standard GAE.

Training pipeline. HAD^{2}RL keeps the base algorithm's actor objective and replaces only the advantage inside it. For a clipping-based base algorithm, agent i maximizes

$$L^i(\theta) = \mathbb{E}_t\Big[\min\big(\rho_t^i \tilde{A}_t^i,\ \mathrm{clip}(\rho_t^i, 1-\varepsilon, 1+\varepsilon)\, \tilde{A}_t^i\big)\Big], \qquad \rho_t^i = \frac{\pi^i_\theta(a_t^i \mid o_t^i)}{\pi^i_{\theta_{\text{old}}}(a_t^i \mid o_t^i)}, \tag{10}$$

where $\tilde{A}_t^i$ is the weighted, reconnected advantage (Eq. 8); trust-region base algorithms use it in their constrained step alike. Sequential updates are unchanged. Each agent trains two critics on standard returns: V^i_{dense} on r_t and V^i_{event} on r_t^E alone. Setting $\lambda = 0$ recovers the base algorithm exactly (Algorithm 1, Appendix C.4).

3.4 Theoretical Guarantees

Propositions 1–3 bound the bias of long-tailed weighting, quantify its noise reduction, and show that GAE reconnection adds no bias. Assumptions and proofs are in Appendix A.

Proposition 1 (Bounded bias of weighting; Appendix A.3) *Let g_w^E be the weighted event policy gradient, g_1^E its unweighted counterpart, and $\bar{w}$ the mean weight over critical states. Then $g_w^E = \bar{w}\, g_1^E + b_w$ with $\|b_w\| \leq (w_{\max} - w_{\min})\, \mathbb{E}_{s\sim d^\pi}\big[\mathbf{1}[s \in \mathcal{S}_c^i]\, \|h^i(s)\|\big]$, where $h^i(s)$ is the expected event gradient at s (Appendix A.1).*

Down-weighting non-critical states introduces no bias, since they contribute nothing to the expected gradient. The only bias comes from uneven weights among critical states and is capped by the weight range, and the hard mask of D2RL is the zero-bias special case. The same range-controlled bound holds for the dense gradient, where w acts as a deliberate budget shift (Appendix A.6).

Proposition 2 (Noise reduction; Appendix A.4) *With weights w_C on critical states and κw_C elsewhere ($\kappa \in [0,1]$), the noise-to-signal ratio of an N-sample event-gradient estimator satisfies ($\kappa \in [0,1]$), the noise-to-signal ratio of an N-sample event-gradient estimator satisfies*

$$\frac{\mathrm{NSR}(w) + 1/N}{\mathrm{NSR}(\mathbf{1}) + 1/N} \;=\; (1 - p_c)\,\kappa^2 + p_c, \tag{11}$$

where p_c is the critical-state fraction.

The reduction is governed by how rare critical states are. At $\kappa = 0$ it recovers the hard-mask reduction p_c of D2RL, and for rare families with $p_c \ll 1$ and $\kappa \in [0.1, 0.3]$ as in our experiments, it amounts to one to two orders of magnitude.

Proposition 3 (Reconnection adds no bias; Appendix A.5) *If the dense value is unbiased on inert states, $\mathbb{E}\big[\delta_t^i \,\big|\, s_t,\, m_t^{\mathrm{inert}} = 1\big] = 0$, then the reconnected GAE of Eq. equation 9 has the same conditional expectation as standard GAE, and it removes the bootstrap variance injected by the dropped inert terms whenever these are weakly correlated with the retained ones.*

Propositions 1 and 2 describe a bias-variance trade-off whose balance depends on how rare failures are. When failures are rare ($p_c \ll 1$), the variance reduction dominates the bounded bias and densification should help most. When failures are frequent ($p_c \to 1$), the reduction vanishes, only the bias remains, and the gain should shrink. This rarity dependence is the prediction that §4 tests.

4 Experiments

We evaluate HAD^{2}RL on DAABench around four questions. Does it find more failures within the natural-behavior envelope (§4.2)? Does the gain follow the rarity dependence predicted in §3.4, and does the learned criticality mark outcome-deciding states (§4.3)? Which failure modes does it expose (§4.4)? What do long-tailed weighting and GAE reconnection each contribute (§4.5)?

4.1 Benchmark and Setup

Benchmark. We build DAABench as a closed-loop benchmark for adversarial failure search on a rule-based DAA system (Fig. 1). The system under test is NASA DAIDALUS v2.0.4 (Consiglio et al., 2015; RTCA, 2017), chosen because it is the deterministic reference implementation of DO-365, so every failure is reproducible. We expose two heterogeneous attack surfaces, 3-D kinematics for the intruder and perception and wind for the perturber, since DAA failures arise from both. Failure is an NMAC under DO-365, horizontal separation below 500 ft and vertical below 100 ft at the same time. From the encounter taxonomy of DO-365 Appendix H we construct 11 encounter families covering head-on, overtaking and overtaken, oblique, and multi-angle converging geometries, and stratify them by how rarely baselines reach an NMAC (Appendix B.2). Table B.1 positions DAABench against existing frameworks, none of which combines adversarial search, a dual attack surface, DO-365 metrics, and a rarity-stratified task suite.

Setup. Episodes run 180 steps at Δt=1 s, and all methods are evaluated on a fixed held-out set of 55,000 episodes, 5,000 per family Appendix B.2, reporting $P(\mathrm{NMAC})$, median miss distances (HMD, VMD), and per-step RLL for each agent, with full DO-365 metrics in Appendix D.1. We compare against nine baselines, four search methods (MCTS-AST, MCBEST, CEM-AST, Go-Explore-AST), two single-agent learners (PPO-AST, TRPO-AST), and three multi-agent learners (MAPPO, HAPPO, HATRPO), and apply HAD^{2}RL to each multi-agent learner as a densification module. Learning methods train for 10^7 steps with the hyperparameters listed in Appendix C.1.

4.2 Main Results

Gains concentrate where failures are rarest. HAD^{2}RL raises aggregate $P(\mathrm{NMAC})$ on every backbone, by 22.7%, 14.6%, and 9.0% on HAPPO, MAPPO, and HATRPO in relative terms, 15.4%

Table 1: Higher P(NMAC) is better (↑); lower HMD/VMD means tighter CPA (↓); RLL nearer 0 is more natural (→ 0). Bold = best learning method.

Method	P(NMAC)↑ (%)	HMD (ft, med.)↓	VMD (ft, med.)↓	RLL^I/step (→ 0)	RLL^P/step (→ 0)
MCTS-AST	2.9 ± 0.1	377	62	-11.63	-2.48
MCBEST	2.9 ± 0.1	374	61	-11.80	-2.48
CEM-AST	5.0 ± 0.2	377	59	-10.42	-3.04
Go-Explore-AST	1.1 ± 0.1	366	65	-11.64	-2.47
PPO-AST	19.4 ± 0.3	377	53	-0.07	$+2.50$
TRPO-AST	8.4 ± 0.2	349	36	-9.77	-3.56
MAPPO	57.4 ± 0.4	312	45	$+0.15$	-0.62
HATRPO	47.7 ± 0.4	306	57	$+0.42$	-0.31
HAPPO	56.4 ± 0.4	292	50	-0.23	-0.95
HAD^{2}RL-MAPPO	65.8 ± 0.4	309	47	$+0.04$	-1.23
HAD^{2}RL-HATRPO	52.0 ± 0.4	321	65	$+0.19$	-2.15
HAD^{2}RL-HAPPO	$\mathbf{69.2 \pm 0.4}$	326	59	-0.24	$\mathbf{-0.18}$

on average. On the rare families HAD^{2}RL-HAPPO finds 28 times as many NMACs as HAPPO, whereas on the saturated families HAPPO already reaches 96% and there is little left to find (§4.3).

Failures are found by assigning credit. Table 1 falls into three tiers, each removing one source of wasted failure signal. Search finds the fewest failures despite pushing actions far outside the envelope (RLL of -10 to -12 per step; Table 1), because what it learns about one encounter is discarded before the next. Learned policies retain it: single-agent learners find 1.7 to 3.9 times as many as the best search method. Yet the joint-policy TRPO-AST finds less than half as many as the intruder-only PPO-AST, since one terminal reward cannot tell which agent's action mattered, or when. Per-agent policies resolve the first ambiguity, and heterogeneous MARL finds 2.5 to 3 times as many as the best single-agent learner. HAD^{2}RL resolves the second, letting each agent learn mainly from steps where its action can still change the outcome, which adds another 9 to 23%.

Failures stay within the natural-behavior envelope. On the intruder channel, HAD^{2}RL stays within 0.25 RLL of its backbone on all three base algorithms (Table 1). On the perturber channel, HAD^{2}RL-HAPPO moves closer to the prior, whereas the MAPPO and HATRPO variants move away; HATRPO shows the largest shift, yet stays within the envelope.

Densification widens the failure set. Multi-agent adversaries penetrate 40 to 85 ft deeper into the NMAC cylinder than search. On HAPPO and HATRPO, HAD^{2}RL has a larger median HMD than its backbone, which reads as a weaker attack but is not: densification adds many marginal failures near the 500-ft boundary that the backbone missed, so the median moves outward as the failure count grows. VMD barely varies within the narrow 100-ft band.

4.3 Analysis by Failure Rarity

Following the encounter-geometry classes of DO-365 Appendix H (bearing, relative heading, and unmitigated HMD), we group the 11 families into three levels by how rarely the base algorithms reach an NMAC: **Rare** (overtaking and overtaken, low closure), **Medium** (head-on, oblique, 60° converging), and **Saturated** (120° and 135° converging).

Gain versus rarity. On HAPPO the gain falls monotonically as failures become common (Fig. 2), and MAPPO shows the same rare-dominated pattern. This is the prediction of Proposition 2. The noise reduction scales with how small the critical-state fraction is, and the three levels differ in exactly this quantity: inert steps dominate an episode in the rare families and almost disappear in the saturated ones (Fig. 3(c)). Where failures are rare, removing that noise is what makes learning possible. Where they are already frequent, there is little noise to remove, only the bounded bias remains, and the gain vanishes within measurement noise. HATRPO deviates from this ordering in a way that confirms rather than contradicts it. Its gain concentrates in the 120° converging families, where its own baseline is near zero, so these families are rare for HATRPO even though they are saturated for HAPPO. In the overtaking families HATRPO never reaches an NMAC, so the event critic has no failure to learn from and densification yields no gain. Densification therefore needs the base algorithm to reach failures occasionally, a limitation we return to in §5.

Validation of criticality. HAD^{2}RL infers criticality from $|\hat{A}^i_{\text{event}}|$ and the inert test rather than calibrating it on naturalistic data as D2RL does, so we check that the states it marks are the ones

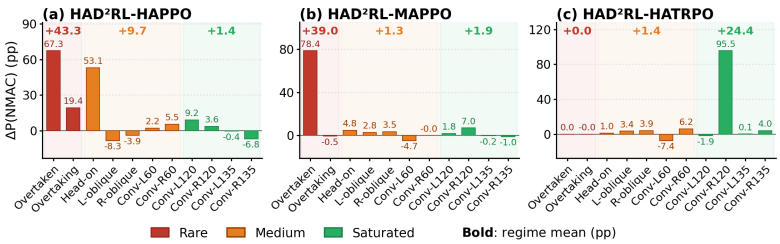

Figure 2: P(NMAC) gain by encounter family for (a) HAD^{2}RL-HAPPO, (b) HAD^{2}RL-MAPPO, and (c) HAD^{2}RL-HATRPO.

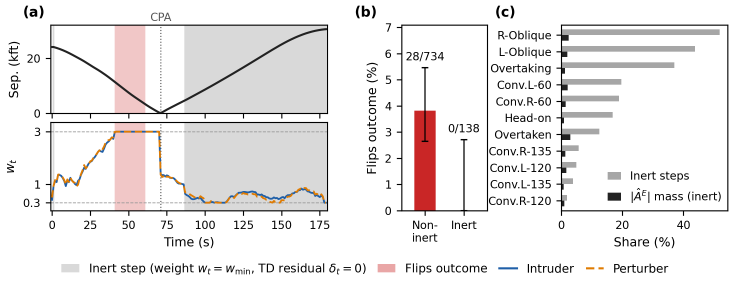

Figure 3: Criticality identified by HAD^{2}RL. (a) Weights w_t in a near-miss. (b) Outcome-flip rate by state type. (c) Inert-step share versus advantage mass.

that decide the outcome. At each replayed state we resample one agent's action, hold everything else fixed, and test whether the NMAC outcome flips. The weight w_t peaks exactly where actions flip the outcome and vanishes where nothing does, no inert state ever flips it, and the flip rate among non-inert states is significantly higher (Fig. 3(a,b)). Throughout training, inert steps can make up half of all steps yet carry a negligible share of the criticality mass, which concentrates on a few percent of steps (Fig. 3(c), Appendix D). The learning budget therefore goes to a small set of steps, and those are the steps whose actions change the outcome.

4.4 Failure-Mode Discovery

The DAIDALUS chain has three links, perception, alerting, and guidance. Grouping the discovered NMACs by the link that breaks first yields the three failure modes below, which complement the kinematic categories of Lipkis et al. (2022). Fig. 4 shows one episode per mode, full trajectories and per-family analysis are in Appendix D.5, and two of the three modes are out of reach for an intruder alone. **(i) Mode A, perception-induced alerting delay.** The perturber blurs the track estimate without leaving the sensor envelope, and the whole alerting chain slides late. In head-on encounters the Warning fires after the DO-365 threshold (AAT $= -8.5$ s, Fig. 4(a), where the green circle marks the unattacked Warning). Without a perception attack surface this mode does not exist. **(ii) Mode B, under-alerting in low-closure encounters.** In overtaking geometries the Warning fires on time, yet closure is so slow that too little separation remains for guidance (Fig. 4(b)). The base algorithms almost never reach an NMAC here, so these geometries would pass as safe, whereas HAD^{2}RL-HAPPO finds NMACs in 22 to 67% of episodes. **(iii) Mode C, guidance defeat under adequate alerting.** DAIDALUS warns on time and the ownship maneuvers, yet the NMAC still occurs (Fig. 4(c)). A late hard intruder maneuver makes the quasi-static guidance stale, while the perturber drifts the ownship's inputs through the closest approach via sensor bias or wind. Only the two agents acting together defeat guidance.

4.5 Ablation Study

Table 2 removes long-tailed weighting (noW) and GAE reconnection (noRC) individually and jointly on the HAPPO backbone, and Fig. D.1 in Appendix D.4 shows the corresponding training dynamics.

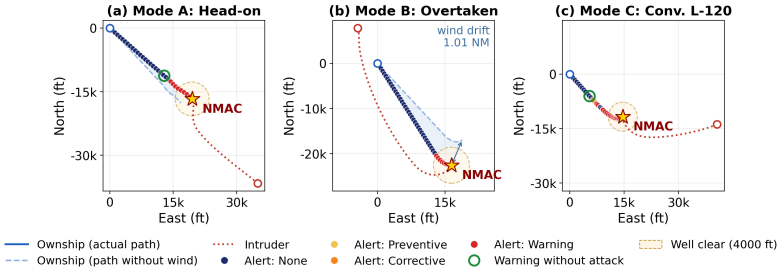

Figure 4: Failure trajectories, one episode per mode: (a) Head-on, (b) Overtaken, (c) Conv. L-120.

Reconnection makes the advantage usable. Removing reconnection costs the most (-17.7 pp in Rare and -63.7 pp in Saturated), and in Saturated it falls far below the plain backbone (33.7% against 96.0%). Early in training the critic is near noise on far-field steps. Standard GAE propagates this noise into every advantage before the closest approach, and weighting then amplifies it wherever the still-noisy event critic fires. Under noRC the intruder's entropy collapses while its actor gradient stays near zero (Fig. D.1(c,e)), so the policy sharpens without receiving a useful signal. The full method is the only one whose collision proxy rises, and it does so with a large actor gradient (Fig. D.1(a,e)), that is, with a gradient that carries signal rather than noise. Reconnection thus supplies the clean advantage that weighting requires (Proposition 3).

Weighting adds on top of a clean advantage. Given reconnection, weighting adds little in Rare ($+2.2$ pp), where most noise is already gone, and more in Saturated ($+9.5$ pp), where the long active window leaves many weakly critical steps to down-weight. In Saturated each operator alone falls below the backbone (87.9% and 33.7% against 96.0%) and only the pair recovers it (97.4%). With almost every step critical, little variance remains and any bias shows up as a loss, which is the $p_c \to 1$ limit of Propositions 1 and 2.

Table 2: Ablation on the HAPPO backbone (held-out $P(\text{NMAC})$ in %, per-step $\text{RLL}^I/\text{RLL}^P$).

	Rare		Saturated	
	$P(\text{NMAC})$	$\text{RLL}^I/\text{RLL}^P$	$P(\text{NMAC})$	$\text{RLL}^I/\text{RLL}^P$
HAD^{2}RL	**44.9±1.0**	$-0.43/-0.07$	**97.4±0.2**	$+0.65/-0.56$
noW	42.7±1.0	$+0.13/-1.26$	87.9±0.5	$+0.69/-1.15$
noRC	27.2±0.9	$+0.09/-1.50$	33.7±0.7	$+0.34/-1.30$
HAPPO	1.6±0.2	$-0.53/-0.16$	96.0±0.3	$+0.89/-1.30$

The two operators interact. Removing both costs far more than the sum of removing each, since weighting without reconnection only reshuffles a corrupted advantage. The effect concentrates on the intruder, whose entropy and gradient norm respond several times more strongly than the perturber's (Fig. D.1(c–f)). Its critical window is narrower (§3.1), so it depends more on where the learning budget goes.

5 Conclusion

We studied adversarial failure search for detect-and-avoid systems as a credit-assignment problem between heterogeneous agents. We introduced heterogeneous criticality, which specifies for each agent and each step how much a state matters, and realized it through soft densification, where long-tailed weighting and GAE reconnection replace the hard mask on the advantage. The method changes only the advantage estimate and applies to existing heterogeneous multi-agent reinforcement learning algorithms. On DAABench it raises failure discovery on every backbone, most sharply in rare encounter families, stays within the natural-behavior envelope, and exposes three failure modes, two of which require both adversaries acting together. One limitation is that densification needs the base algorithm to reach failures occasionally, since the event critic has nothing to learn from otherwise. Future work includes bootstrapping from search when no failure is found, replacing the hand-specified inert test and behavior prior with learned counterparts, isolating continuous weights and per-agent critical sets against hard-mask and shared-set variants, and extending the framework to more agents and other collision-avoidance logics.

AI Use Statement

Generative AI tools were used only to polish the writing, such as editing passages for readability and length. They were not used to develop the method, the theory, the experiments, or the analysis, nor to generate data or figures. All AI-edited text was reviewed by the authors, who take full responsibility for the content of this work.

Ethics Statement

This work involves no human subjects and no personal data. All encounters are generated in closed-loop simulation, and the system under test is the publicly available NASA DAIDALUS v2.0.4, run at its default DO-365 parameters without tuning. The adversarial agents are intended for pre-certification stress testing. They expose failure modes of a detect-and-avoid implementation so that the modes can be analyzed and fixed. The paper reports encounter geometry, alert timing, and aggregate failure rates rather than an operational procedure for attacking deployed aircraft, and the perturbations act only on simulated sensor inputs and wind. The authors declare no competing interests.

Reproducibility Statement

The problem formulation, reward, and algorithm are given in §3, with training pseudocode in Algorithm 1 and complete assumptions and proofs of Propositions 1–3 in Appendix A. The benchmark, including the simulation environment, the 11 encounter families, the held-out protocol, metric definitions, and baseline implementations, is described in Appendix B. Network architectures, hyperparameters, the λ-schedule, and the inert-step criterion are in Appendix C. Full per-family results, RLL breakdowns, and trajectory replays are in Appendix D. All methods are evaluated on the same fixed set of $11 \times 5{,}000$ held-out episodes with fixed evaluation seeds. Anonymous source code for the environment, HAD^{2}RL, and the evaluation scripts is provided as supplementary material.

References

Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, OpenAI Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. *Advances in neural information processing systems*, 30, 2017.

Jose A Arjona-Medina, Michael Gillhofer, Michael Widrich, Thomas Unterthiner, Johannes Brandstetter, and Sepp Hochreiter. Rudder: Return decomposition for delayed rewards. *Advances in Neural Information Processing Systems*, 32, 2019.

Christopher Brix, Stanley Bak, Taylor T. Johnson, and Haoze Wu. The fifth international verification of neural networks competition (VNN-COMP 2024): Summary and results. *arXiv preprint arXiv:2412.19985*, 2024.

Keyu Chen, Yuheng Lei, Hao Cheng, Haoran Wu, Wenchao Sun, and Sifa Zheng. FREA: Feasibility-guided generation of safety-critical scenarios with reasonable adversariality. In *Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 566–586, 2024.

Maria Consiglio, Cesar Muñoz, George Hagen, Anthony Narkawicz, Jason Upchurch, James Comstock, Rania Ghatas, Michael Vincent, and James Chamberlain. Human-in-the-loop experimental research for detect and avoid. In *2015 IEEE/AIAA 34th Digital Avionics Systems Conference (DASC)*, pp. 5A3–1, 2015.

Anthony Corso, Robert Moss, Mark Koren, Ritchie Lee, and Mykel Kochenderfer. A survey of algorithms for black-box safety validation of cyber-physical systems. *Journal of Artificial Intelligence Research*, 72:377–428, 2021.

Adrien Ecoffet, Joost Huizinga, Joel Lehman, Kenneth O. Stanley, and Jeff Clune. First return, then explore. *Nature*, 590(7847):580–586, 2021.

Shuo Feng, Xintao Yan, Haowei Sun, Yiheng Feng, and Henry X. Liu. Intelligent driving intelligence test for autonomous vehicles with naturalistic and adversarial environment. *Nature Communications*, 12:748, 2021.

Shuo Feng, Haowei Sun, Xintao Yan, Haojie Zhu, Zhengxia Zou, Shengyin Shen, and Henry X. Liu. Dense reinforcement learning for safety validation of autonomous vehicles. *Nature*, 615(7953): 620–627, 2023.

Shuo Feng, Haojie Zhu, Haowei Sun, Xintao Yan, Linxuan He, Jingxuan Yang, Guangzhen Su, Boqi Li, Shu Li, Ling Wang, Shengyin Shen, and Henry X. Liu. Breaking through safety performance stagnation in autonomous vehicles with dense learning. *Nature Communications*, 17(1):3163, 2026.

Hang Fu, Jingjing Wang, Jianrui Chen, Pengfei Ren, Zheng Zhang, and Guodong Zhao. Dense multiagent reinforcement learning aided multi-UAV information coverage for vehicular networks. *IEEE Internet of Things Journal*, 11(12):21274–21286, 2024.

Mykel J. Kochenderfer, Matthew W. M. Edwards, Leo P. Espindle, James K. Kuchar, and J. Daniel Griffith. Airspace encounter models for estimating collision risk. *Journal of Guidance, Control, and Dynamics*, 33(2):487–499, 2010.

Mark Koren, Saud Alsaif, Ritchie Lee, and Mykel J. Kochenderfer. Adaptive stress testing for autonomous vehicles. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1–7, 2018.

Jakub Grudzien Kuba, Ruiqing Chen, Muning Wen, Ying Wen, Fanglei Sun, Jun Wang, and Yaodong Yang. Trust region policy optimisation in multi-agent reinforcement learning. In *International Conference on Learning Representations*, 2022.

Ritchie Lee, Mykel J. Kochenderfer, Ole J. Mengshoel, Guillaume P. Brat, and Michael P. Owen. Adaptive stress testing of airborne collision avoidance systems. In *2015 IEEE/AIAA 34th Digital Avionics Systems Conference (DASC)*, pp. 6C2–1, 2015.

Ritchie Lee, Ole J. Mengshoel, Anshu Saksena, Ryan W. Gardner, Daniel Genin, Jeffrey Brush, and Mykel J. Kochenderfer. Differential adaptive stress testing of airborne collision avoidance systems. In *AIAA Modeling and Simulation Technologies Conference*, 2018. AIAA 2018-1923.

Ritchie Lee, Ole J. Mengshoel, Anshu Saksena, Ryan W. Gardner, Daniel Genin, Joshua Silbermann, Michael Owen, and Mykel J. Kochenderfer. Adaptive stress testing: Finding likely failure events with reinforcement learning. *Journal of Artificial Intelligence Research*, 69:1165–1201, 2020.

Rory Lipkis and Adrian Agogino. Discovery and analysis of rare high-impact failure modes using adversarial rl-informed sampling. In *International Conference on Autonomous Agents and Multiagent Systems*, pp. 123–140. Springer, 2023.

Rory Lipkis, Ritchie Lee, Joshua Silbermann, and Tyler Young. Adaptive stress testing of collision avoidance systems for small uass with deep reinforcement learning. In *AIAA SciTech 2022 Forum*, pp. 1854, 2022.

Henry X. Liu and Shuo Feng. Curse of rarity for autonomous vehicles. *Nature Communications*, 15: 4808, 2024.

Ryan Lowe, Yi I Wu, Aviv Tamar, Jean Harb, OpenAI Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. *Advances in neural information processing systems*, 30, 2017.

Andrew Y. Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *International Conference on Machine Learning*, volume 99, pp. 278–287, 1999.

Tong Nie, Yuewen Mei, Yihong Tang, Junlin He, Jie Sun, Haotian Shi, Wei Ma, and Jian Sun. Steerable adversarial scenario generation through test-time preference alignment. In *International Conference on Learning Representations*, volume 2026, pp. 120964–121003, 2026.

Matthew O'Kelly, Aman Sinha, Hongseok Namkoong, John C. Duchi, and Russ Tedrake. Scalable end-to-end autonomous vehicle testing via rare-event simulation. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

RTCA. DO-365: Minimum operational performance standards (MOPS) for detect and avoid (DAA) systems. Technical report, RTCA, Inc., 2017.

Leopoldo Rodriguez Salazar, Jose A. Cobano, and Anibal Ollero. Small UAS-based wind feature identification system part 1: Integration and validation. *Sensors*, 17(1):8, 2017. doi: 10.3390/s17010008.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. In *International Conference on Learning Representations*, 2016.

Zhouxing Shi, Qirui Jin, Zico Kolter, Suman Jana, Cho-Jui Hsieh, and Huan Zhang. Neural network verification with branch-and-bound for general nonlinearities. In *Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, volume 15696 of *Lecture Notes in Computer Science*, pp. 315–335, 2025.

Elysia Smyers, Sydney Katz, Anthony Corso, and Mykel J Kochenderfer. Avoidds: Aircraft vision-based intruder detection dataset and simulator. *Advances in Neural Information Processing Systems*, 36:8076–8091, 2023.

Haoze Wu, Omri Isac, Aleksandar Zeljić, Teruhiro Tagomori, Matthew Daggitt, Wen Kokke, Idan Refaeli, Guy Amir, Kyle Julian, Shahaf Bassan, Pei Huang, Ori Lahav, Min Wu, Min Zhang, Ekaterina Komendantskaya, Guy Katz, and Clark Barrett. Marabou 2.0: A versatile formal analyzer of neural networks. In *Computer Aided Verification (CAV)*, volume 14682 of *Lecture Notes in Computer Science*, pp. 249–264, 2024.

Xintao Yan, Zhengxia Zou, Shuo Feng, Haojie Zhu, Haowei Sun, and Henry X. Liu. Learning naturalistic driving environment with statistical realism. *Nature Communications*, 14(1):2037, 2023.

Chao Yu, Akash Velu, Eugene Vinitsky, Jiaxuan Gao, Yu Wang, Alexandre Bayen, and Yi Wu. The surprising effectiveness of PPO in cooperative multi-agent games. In *Advances in Neural Information Processing Systems*, volume 35, pp. 24611–24624, 2022.

Ding Zhao, Henry Lam, Huei Peng, Shan Bao, David J. LeBlanc, Kazutoshi Nobukawa, and Christopher S. Pan. Accelerated evaluation of automated vehicles safety in lane-change scenarios based on importance sampling techniques. *IEEE Transactions on Intelligent Transportation Systems*, 18(3):595–607, 2016.

Yifan Zhong, Jakub Grudzien Kuba, Xidong Feng, Siyi Hu, Jiaming Ji, and Yaodong Yang. Heterogeneous-agent reinforcement learning. *Journal of Machine Learning Research*, 25(32): 1–67, 2024.

A Bias and Variance Analysis of Soft Densification

This appendix gives precise statements and complete proofs of the propositions in §3.4. Propositions A.2 to A.4 are the precise versions of Propositions 1 to 3 in the main text, and Proposition A.1 covers the hard-mask limit. Section A.1 introduces notation and assumptions, and Section A.6 collects three remarks.

A.1 Notation and Assumptions

Consider a single agent (superscript i suppressed). Let π_θ denote the policy and d^π the discounted state-visitation distribution. Throughout this appendix, A^π refers to the advantage under the **event MDP**, which contains only the sparse event reward r^E (Eq. equation 4):

$$A^\pi(s,a) = Q_\pi^{\text{event}}(s,a) - V_\pi^{\text{event}}(s).$$

By the policy gradient theorem, $\nabla J(\theta) = \mathbb{E}_{s\sim d^\pi, a\sim\pi}[A^\pi(s,a)\nabla \log \pi_\theta(a \mid s)]$. Define

$$h(s) \triangleq \mathbb{E}_{a\sim\pi}[A^\pi(s,a)\nabla \log \pi(a \mid s)],$$

so that $\nabla J = \mathbb{E}_{s\sim d^\pi}[h(s)]$. The **critical-state set** (Feng et al., 2023; Definition 1 in the main text with the agent index suppressed) is $\mathcal{S}_c \triangleq \{s : A^\pi(s,\cdot) \not\equiv 0\}$, which satisfies

$$s \notin \mathcal{S}_c \implies h(s) = 0. \tag{12}$$

The **weighted policy-gradient estimator** of HAD^{2}RL is

$$\hat{g}_w = \frac{1}{N}\sum_t w_t\, \hat{A}_t\, \nabla \log \pi(a_t \mid s_t),$$

which reduces to the standard estimator $\hat{g}$ when $w_t \equiv 1$. The analysis relies on two assumptions:

- **(A1)** $\mathbb{E}[\hat{A}_t \mid s_t, a_t] = A^\pi(s_t, a_t)$ (unbiased advantage estimate).
- **(A2)** $w_t = w(s_t) > 0$ is a function of the state alone and, given s_t, is conditionally independent of a_t and of the estimation noise.

All propositions below hold independently for each agent.

A.2 Hard-Mask Limit

Proposition A.1 (Hard-mask limit). Let $w(s) = \mathbf{1}[s \in \mathcal{S}_c]$. Under (A1)–(A2): (1) $\mathbb{E}[\hat{g}_w] = \nabla J(\theta)$; (2) $\mathrm{Var}[\hat{g}_w] \le \mathrm{Var}[\hat{g}]$.

Proof. By (A1)–(A2) and the tower property,

$$\mathbb{E}[\hat{g}_w] = \mathbb{E}_s[w(s)\, h(s)]. \tag{13}$$

Substituting $w = \mathbf{1}_{\mathcal{S}_c}$ and applying Eq. equation 12 gives $\mathbb{E}_s[\mathbf{1}_{\mathcal{S}_c} h] = \mathbb{E}_s[h] = \nabla J$. For the variance, the zeroed terms $\mathbf{1}[s \notin \mathcal{S}_c]\hat{A}\nabla \log \pi$ have zero conditional mean, and removing zero-mean additive noise does not increase the variance (Theorem 1, property (2), of Feng et al., 2023). ■

Proposition A.1 shows that D2RL, which uses the hard mask, is the special case of HAD^{2}RL with $\kappa = 0$ in the notation of Section A.4.

A.3 Bias of Continuous Weighting

Proposition A.2 (Bounded bias of weighting). Under (A1)–(A2):

(a) $\mathbb{E}[\hat{g}_w] = \mathbb{E}_{s\sim d^\pi}[\mathbf{1}_{\mathcal{S}_c}\, w(s)\, h(s)] =: g_w$. That is, $\hat{g}_w$ is an unbiased estimator of the policy gradient under the densified state occupancy $d_w^\pi \propto w \cdot d^\pi$; weights on non-critical states do not affect the expectation.

(b) If $w|_{\mathcal{S}_c} \equiv c > 0$, then $\mathbb{E}[\hat{g}_w] = c\nabla J$ (directionally unbiased).

(c) Let $\bar{w}$ be the d^π-weighted mean of w over $\mathcal{S}_c$. Then

$$\mathbb{E}[\hat{g}_w] = \bar{w}\nabla J + b_w, \qquad \|b_w\| \le \mathrm{osc}_{\mathcal{S}_c}(w) \cdot \mathbb{E}_s[\mathbf{1}_{\mathcal{S}_c}\|h(s)\|],$$

where $\mathrm{osc}_{\mathcal{S}_c}(w) = \sup_{\mathcal{S}_c} w - \inf_{\mathcal{S}_c} w \le w_{\max} - w_{\min}$.

Proof. (a) By Eqs. equation 13 and equation 12, terms with $s \notin \mathcal{S}_c$ contribute nothing to the expectation. (b) Substitute $w \equiv c$ on $\mathcal{S}_c$. (c) Decompose $w = \bar{w} + (w - \bar{w})$ on $\mathcal{S}_c$; the first term yields $\bar{w}\nabla J$, and the remainder is bounded by $|w - \bar{w}| \le \mathrm{osc}(w)$. ■

In the notation of the main text, $g_w^E := \mathbb{E}[\hat{g}_w]$ and $g_1^E = \nabla J$, so part (c) with the coarser range $w_{\max} - w_{\min}$ in place of $\mathrm{osc}_{\mathcal{S}_c}(w)$ is exactly Proposition 1:

$$g_w^E = \bar{w}\, g_1^E + b_w, \qquad \|b_w\| \le (w_{\max} - w_{\min})\, \mathbb{E}_{s\sim d^\pi}\big[\mathbf{1}[s \in \mathcal{S}_c]\, \|h(s)\|\big], \tag{14}$$

where $w_{\min} \le w \le w_{\max}$ holds by the clipping in Eq. equation 7.

Interpretation. The bias arises solely from the non-uniformity of w *within* $\mathcal{S}_c$, as measured by $\mathrm{osc}_{\mathcal{S}_c}(w)$. Down-weighting non-critical states introduces no bias; non-uniform weights within the critical set steer the gradient away from ∇J, which is a deliberate reshaping of the objective.

A.4 Noise-to-Signal Reduction

Since $\mathbb{E}[\hat{g}_w] \neq \mathbb{E}[\hat{g}]$, we compare estimators through the scale-invariant **noise-to-signal ratio** $\mathrm{NSR}(w) \triangleq \mathrm{Var}[\hat{g}_w]/\|\mathbb{E}[\hat{g}_w]\|^2$. We introduce an additional assumption:

- **(A3)** $\mathbb{E}[\|\hat{A}\nabla \log \pi\|^2 \mid s] = \sigma^2$ is identical for all s (homogeneous conditional second moment; Feng et al. (2023) note that this holds approximately early in training).

Continuous weights are treated block-wise: we consider $w = w_C$ on $\mathcal{S}_c$ and $w = w_U$ on $\mathcal{S} \setminus \mathcal{S}_c$, with $\kappa = w_U/w_C \in [0, 1]$ and $p_c = \Pr_{d^\pi}(s \in \mathcal{S}_c)$.

Proposition A.3 (Noise reduction). Under (A1)–(A3),

$$\frac{\mathrm{NSR}(w) + 1/N}{\mathrm{NSR}(\mathbf{1}) + 1/N} = (1 - p_c)\kappa^2 + p_c.$$

Hence the NSR is strictly increasing in κ and attains its minimum p_c at $\kappa = 0$ (hard mask).

Proof. Let $X = w(s)\hat{A}\nabla \log \pi$. By Proposition A.2(b), $\mathbb{E}[X] = w_C\nabla J$. By (A3), $\mathbb{E}\|X\|^2 = \sigma^2[(1 - p_c)w_U^2 + p_c w_C^2]$. Substituting into $\mathrm{NSR}(w) + 1/N = \mathbb{E}\|X\|^2/(N\|\mathbb{E}X\|^2)$ and dividing by the $\kappa = 1$ case yields the result. ■

Interpretation. When $p_c \ll 1$ (rare failure families), the reduction factor is approximately κ^2. A moderate $\kappa \approx 0.1$–0.3 already improves the NSR by one to two orders of magnitude, which is consistent with the pronounced gains of HAD^{2}RL in rare families.

A.5 GAE Reconnection

The standard and reconnected GAE are

$$\hat{A}_t = \sum_{l\ge 0}(\gamma\lambda_{\mathrm{gae}})^l \delta_{t+l}, \qquad \hat{A}_t^{\mathrm{rc}} = \sum_{l\ge 0}(\gamma\lambda_{\mathrm{gae}})^l (1 - m_{t+l}^{\mathrm{inert}})\delta_{t+l},$$

where $\delta_t = r_t + \gamma V(s_{t+1}) - V(s_t)$. We introduce one additional assumption, of which the condition stated in Proposition 3 is the consequence:

- **(A4)** Every inert step u ($m_u^{\mathrm{inert}} = 1$) satisfies $s_u \notin \mathcal{S}_c$, and the critic is exact there, $V(s_u) = V^\pi(s_u)$, so that $\mathbb{E}[\delta_u \mid s_u, m_u^{\mathrm{inert}} = 1] = 0$.

Proposition A.4 (Reconnection adds no bias). Under (A4): (1) $\mathbb{E}[\hat{A}_t^{\mathrm{rc}} \mid s_t] = \mathbb{E}[\hat{A}_t \mid s_t]$; (2) if the dropped δ terms are weakly correlated with the retained ones, then $\mathrm{Var}[\hat{A}_t^{\mathrm{rc}}] \le \mathrm{Var}[\hat{A}_t]$.

Proof. The difference is $\hat{A}_t - \hat{A}_t^{\mathrm{rc}} = \sum_l (\gamma\lambda_{\mathrm{gae}})^l m_{t+l}^{\mathrm{inert}}\delta_{t+l}$. By (A4) and the tower property, each inert term satisfies $\mathbb{E}[m_u^{\mathrm{inert}}\delta_u \mid s_t] = 0$, so the difference has zero conditional expectation. For the variance, removing zero-mean, weakly correlated additive terms does not increase it. ■

The inert criterion uses a four-way conjunction (Appendix C.3), biased toward under-labeling so as not to discard critical steps. The critic is trained on standard (unreconnected) returns, so value estimates are unaffected.

A.6 Remarks

(1) Dependence on failure rarity. Propositions A.2(c) and A.3 together show that the net benefit of densification depends on p_c. In rare families ($p_c \ll 1$), the NSR improvement is substantial while the bias cost remains small; in saturated families the opposite holds. This is consistent with the experimental findings in §4.

(2) Effect on the dense gradient. The propositions above concern the event advantage. In practice, the actor uses the dense shared advantage, whose $h_{\text{dense}}(s)$ need not vanish off $\mathcal{S}_c$, so down-weighting non-critical states ($w < 1$) does change the expected gradient there, by design. The range still controls the size of that change: under the dense-advantage analogues of (A1)–(A2), decomposing $w = \bar{w}_{\mathcal{S}} + (w - \bar{w}_{\mathcal{S}})$ with $\bar{w}_{\mathcal{S}}$ the d^π-mean of w over *all* states gives

$$\mathbb{E}[\hat{g}_w^{\text{dense}}] = \bar{w}_{\mathcal{S}}\, \nabla J_{\text{dense}} + b, \qquad \|b\| \le (w_{\max} - w_{\min})\, \mathbb{E}_{s\sim d^\pi} \|h_{\text{dense}}(s)\|,$$

where b is the deliberate shift of the advantage budget toward critical states. The floor $w_{\min} > 0$ preserves the minimum signal needed for far-field shaping.

(3) Approximate nature of the assumptions. In (A2), w_t in fact depends on actions (through rollout averaging and clipping); (A1) and (A4) hold only approximately under finite training; and (A3) holds early in training, after which Proposition A.3 gives a qualitative trend. These assumptions hold to the same degree of approximation as in D2RL.

B DAABench Details

This appendix provides the full technical details of DAABench introduced in §4 and Table B.1. It covers the simulation environment and interface (B.1), encounter families and instance generation (B.2), DO-365 Appendix L metrics (B.3), the reward formulation (B.4), and the baseline suite (B.5).

B.1 Environment and Interface

Table B.1 compares DAABench with existing testbeds. This section specifies the full configuration of the environment.

System under test (SUT). NASA DAIDALUS v2.0.4 (Consiglio et al., 2015; RTCA, 2017) is the reference implementation of the DO-365 DAA alerting and guidance functions. It is embedded in the closed loop with the default parameters of DO-365 Table 2-21, and no parameter tuning is applied. At each step, DAIDALUS receives possibly corrupted ownship and intruder state estimates and outputs a four-level alert (None, Preventive, Corrective, Warning) together with heading, speed, and vertical-rate guidance bands. Hysteresis and M-of-N persistence logic are fully retained. Well-clear volume parameters: $\tau^*_{\text{mod}} = 35$ s, $\text{DMOD} = \text{HMD}^* = 4{,}000$ ft, $h^* = 450$ ft.

Ownship flight dynamics. The ownship is a deterministic guidance follower that executes avoidance maneuvers along three axes according to the DAIDALUS guidance bands. We model a pilot response delay of 3 s and a communication delay drawn from $\text{Unif}[0, 2]$ s. Kinematic constraints: maximum turn rate 3 °/s (consistent with the DO-365 Phase I dynamics model), ground-speed range $[40, 200]$ kt, vertical rate ± 500 fpm, and vertical acceleration $\le 0.25\, g$. Directional persistence is enforced, so once an avoidance side is selected it is maintained until the conflict is resolved.

Dual attack surface. Two heterogeneous adversarial agents stress the same alerting chain through distinct causal pathways, summarized in Table B.2.

All actions are bounded to their physical ranges via squashed-Gaussian outputs. The communication delay τ is not part of the perturber's policy. It is sampled uniformly from $\{0, 1, 2\}$ s at episode start and held constant throughout, and all methods share the same τ sampling protocol. The perturber's five policy dimensions are step-wise increments superimposed on an Ornstein–Uhlenbeck natural-disturbance prior (Appendix C, Table C.1). Background wind speed follows a truncated Weibull

Table B.1: Comparison of DAABench with existing safety testing frameworks. ✓: present; ×: absent; (·): partial; —: not applicable.

Framework	SUT	Adv. search	DO-365	Dual attack surface	RLL	Tasks	Baselines
DO-365 Monte Carlo	DAIDALUS	×	✓	×	—	(·)	—
ACAS X AST	TCAS / ACAS X	✓	×	×	×	×	×
ACAS sXu AST	ACAS sXu	✓	×	×	✓ (single)	×	×
AVOIDDS	Vision-based DAA	×	×	×	×	✓	✓
NADE / D2RL	AV driving system	✓	—	×	×	×	×
SafeBench	RL driving policy	✓	—	×	×	×	✓
DAABench (Ours)	**DAIDALUS**	✓	✓	✓	✓ (dual)	✓	✓

Table B.2: Dual attack surface. Two heterogeneous adversarial agents stress the same alerting chain through distinct causal pathways.

Agent	Dim.	Action components	Causal pathway
Intruder (I)	3	Turn rate $\pm 3^\circ$/s, forward acceleration $\pm 2\,\mathrm{m/s}^2$, vertical rate ± 1500 fpm	Kinematic: directly alters the encounter geometry
Perturber (P)	5	Wind increments (E/N, ± 15 kt each); sensor position bias ($[0, 1]$); horizontal- and vertical-velocity estimation errors ($[0, 1]$)	Perception and execution: corrupts SUT state inputs and drifts the ownship track

distribution $f(W) = (\kappa/\nu)(W/\nu)^{\kappa-1} \exp[-(W/\nu)^\kappa]$ with $\kappa = 2.0$ and $\nu = 17$ kt, truncated to $[3, 40]$ kt (Salazar et al., 2017).

Simulation specification. Time step $\Delta t = 1$ s and fixed episode length 180 s (180 steps). An episode terminates with an NMAC if, at the closest point of approach (CPA), horizontal separation is below 500 ft and vertical separation below 100 ft, following the DO-365 definition.

Observation space. The local observation o_t^i encodes ownship and intruder kinematics, current wind and sensor states, the DAIDALUS alert level and guidance bands, and relative geometry (bearing, range, closure rate, time to CPA). The global state s_t used by the critic is a superset of both agents' observations. Dimensions: $\dim o^{\mathrm{I}} = 25$, $\dim o^{\mathrm{P}} = 21$, $\dim s = 48$.

Interface. The environment exposes a `gym.Env`-style `reset(instance_id, seed)` / `step(a_I, a_P)` API with a PettingZoo parallel wrapper. The `info` dict returns per-step alert levels, clean and corrupted dual-channel state estimates, and geometric quantities.

B.2 Encounter Families and Instance Generation

Eleven encounter families. DO-365 Appendix H defines seven encounter geometries by right-of-way rules (head-on, overtaken, overtaking, left and right oblique overtaking, converging from left and from right) and prescribes the ownship's right of way and maneuver direction for each. We adopt this taxonomy as the classification framework. The two converging geometries are each subdivided by convergence angle $\{60, 120, 135\}$ into three families, and the remaining five geometries each form one family, yielding 11 families in total. Encounters are initialized following the Appendix H geometries so that both aircraft reach their nominal CPA under zero intervention.

Instance parameterization. Each instance is sampled uniformly within family-specific bounds over initial relative range, bearing, altitude offset, initial speeds, and nominal time to CPA, with a global $\mathrm{Unif}[0, 2\pi)$ rotation augmentation applied to the encounter geometry.

Natural prior. The intruder's natural prior is the zero action (constant-track straight flight), and the perturber's natural prior is zero disturbance (no wind estimation error and no sensor bias). RLL (Eq. 3) is computed relative to this prior, consistent with the zero-reference baseline of the likelihood cost in B.4.

Train and evaluation split. Training instances are sampled online from training seeds. Evaluation uses a fixed held-out set of 5,000 instances per family with deterministic seeds released with the benchmark, and all methods share the same set. Held-out and training instances are drawn from the same family distributions but do not overlap.

Table B.3: DO-365 Appendix L metrics reported by DAABench.

#	Metric	Definition
1	$P(\mathrm{NMAC})$	$\lvert\mathcal{E}_{\mathrm{NMAC}}\rvert/\lvert\mathcal{E}\rvert$
2	Decision success rate	$1 - P(\mathrm{NMAC})$
3	Min. separation HMD/VMD	Median horizontal/vertical distance at CPA over NMAC episodes (ft)
4	Correct alert rate $P(\mathrm{CRA})$	Fraction of hazard episodes with a correct alert before the threshold
5	Late alert rate $P(\mathrm{LA})$	Fraction of hazard episodes whose first alert arrives after the threshold
6	Average alert time (AAT)	Lead of the first alert relative to the 25 s threshold (s)
7	Reaction budget	Median time from first alert to CPA over NMAC episodes (s)
8	Alert coverage	Mean alert duration and fraction of episodes with an alert
9	Robust recovery rate	Fraction of surviving episodes that entered a hazard zone and recovered safe separation
10	Maximum deviation	Median lateral/vertical deviation of the ownship from its nominal track (m)

Rarity labeling. The 11 families are labeled rare ($< 5\%$), medium (5–90%), or saturated ($> 90\%$) by the held-out $P(\mathrm{NMAC})$ of the HAPPO baseline, for the analysis in §4.3. Per-family $P(\mathrm{NMAC})$ values and labels are listed in Table D.2.

B.3 DO-365 Appendix L Metrics

DAABench scores search methods with the operational metrics of DO-365 Appendix L (Table B.3). Let $\mathcal{E}$ denote the evaluation episode set, $\mathcal{E}_{\mathrm{NMAC}}$ the NMAC episodes, $\mathcal{E}_{\mathrm{HAZ}}$ the hazard episodes (those that entered the well-clear volume defined in B.1 and triggered an alert $\geq$ Preventive), and $\mathcal{E}_{\mathrm{surv}} = \mathcal{E} \setminus \mathcal{E}_{\mathrm{NMAC}}$ the surviving episodes. Metrics 1–3, 7, and 9 characterize collision outcomes, 4–6 and 8 characterize alerting-chain failures, and 10 captures avoidance cost. RLL (Eq. 3) is computed per episode and reported as a distribution (median and IQR). The benchmark leaderboard ranks methods by the pair $(P(\mathrm{NMAC}), \mathrm{RLL})$.

B.4 Reward Formulation

This section gives the complete per-step shared reward r_t referred to in §3.1. Both agents receive the same team reward

$$r_t \;=\; \underbrace{r_t^I}_{\text{intruder component}} \;+\; \underbrace{r_t^P}_{\text{perturber component}} \;,$$

under which the sequential advantage decomposition theorem of HAPPO holds. The two components are detailed below.

B.4.1 Intruder Component

Event term. R_E is awarded only at the NMAC step (DO-365 dual threshold, horizontal < 500 ft and vertical < 100 ft). **Minimum-separation ratchet.** Progress is measured by the NMAC-cylinder normalized distance

$$u_t = \min\Big(\frac{\mathrm{miss}_{\mathrm{cyl},t}}{c_{\mathrm{ref}}},\, 1\Big),$$
$$\mathrm{miss}_{\mathrm{cyl},t} = \max\Big(\frac{h_t}{500\,\mathrm{ft}},\, \frac{v_t}{100\,\mathrm{ft}}\Big),$$
$$r_t^{\mathrm{dmin}} = \beta \cdot \max(u_{t-1}^* - u_t,\, 0),$$

where $c_{\mathrm{ref}} = 16$, $u_t^* = \min_{\tau \leq t} u_\tau$, and $\beta = 100$. After telescoping, the cumulative episode reward is $\sum_t r_t^{\mathrm{dmin}} = -\beta\, u_T^*$. **Potential shaping.** $\Phi^I(s) = -w_{\mathrm{pot}} \cdot d_{3D}(s)/d_{\mathrm{ref}}$ ($w_{\mathrm{pot}} = 30$, $d_{\mathrm{ref}} = 20{,}000$ m) provides a step-wise approach gradient in the ratchet plateau region. **Likelihood**

Table B.4: Baseline suite. All methods share the DAABench environment, the held-out set, and the reward of Appendix B.4. PPO-AST uses only the 3-D intruder, and every other method uses the full 8-D attack surface.

Method	Category	Core mechanism	Configuration difference
MCTS-AST	Search	Action-prefix tree with progressive UCT widening	100 iterations per instance
MCBEST	Search	100 full samples from the natural prior; keep the best	No tree and no distribution update
CEM-AST	Search	Per-step proposal distribution, refit on elites	30 trajectories per generation; top 20% elites
Go-Explore-AST	Search	Cell-archive coverage with deterministic replay	Phase 1 only
PPO-AST	Learning	3-D intruder pursuit only	Disturbances sampled from the prior (not learned)
TRPO-AST	Learning	Trust-region policy optimization	Target KL $= 0.01$
HAPPO	Learning	Heterogeneous sequential policy optimization	Base algorithm
MAPPO	Learning	Parameter-sharing proximal policy optimization	Base algorithm
HATRPO	Learning	Heterogeneous trust-region policy optimization	Base algorithm
HAD^{2}RL-HAPPO	Learning	HAPPO + long-tailed weighting + GAE reconnection (§3)	Ours
HAD^{2}RL-MAPPO	Learning	MAPPO + long-tailed weighting + GAE reconnection (§3)	Ours
HAD^{2}RL-HATRPO	Learning	HATRPO + long-tailed weighting + GAE reconnection (§3)	Ours

cost. $\ell_t^I = \lambda_I \cdot \frac{1}{2}\sum_j (a_{t,j}^I/\sigma_j^I)^2$, where $\sigma^I = (0.5, 0.3, 0.07)$ are the nominal standard deviations for turn rate, forward acceleration, and vertical rate. The zero-action reference prevents no-action exploitation.

B.4.2 Perturber Component

The perturber receives no event term. Its reward consists of dense shaping and a likelihood cost, $r_t^P = S(\Phi_t^P) - \ell_t^P$, where the potential Φ_t^P has three components:

- **Alert degradation**: $w_{\text{perc}} \cdot c_t \cdot \Delta\alpha_t/3$, where $\Delta\alpha_t = \alpha_{\text{clean}} - \alpha_{\text{perturbed}} \in \{0, \dots, 3\}$, normalized to $[0, 1]$.
- **Perception separation gap**: $w_{\text{gap}} \cdot c_t \cdot g_t$. Sensor bias causes DAIDALUS to overestimate separation, and g_t is channel-normalized.
- **Execution-layer effect**: $w_{\text{exec}} \cdot c_t \cdot \text{clip}(\theta_t/\theta_{\text{ref}}, -1, 1)$, where θ_t is the wind-induced drift angle and $\theta_{\text{ref}} = 15$.

The gate $c_t = \mathbf{1}[\alpha_{\text{clean},t} > 0]$ ensures that far-field steps with no alert receive zero shaping. Weights: $w_{\text{perc}} = 0.15$, $w_{\text{gap}} = 0.05$, $w_{\text{exec}} = 0.15$. Φ^P is delivered via potential-based reward shaping, so its net episode contribution is zero.

Perturber likelihood cost. $\ell_t^P = \lambda_P \cdot \frac{1}{2}\sum_j (f_j/\sigma_P)^2 + \ell_t^{\text{wind-smooth}} + \ell_t^{\text{sensor-smooth}}$, with $\lambda_P = 0.03$ and $\sigma_P = 0.5$. The last two terms penalize deviations from the OU increment prior to preserve physical continuity.

B.5 Baselines

All baselines share the same environment, the same held-out set, the same reward (B.4), and the same 8-dimensional attack surface (except PPO-AST). Table B.4 summarizes categories and key differences. **Search baselines** run per instance with 100 iterations. **Single-agent learning baselines** (PPO-AST, TRPO-AST) learn cross-instance policies. PPO-AST controls only the 3-D intruder, and disturbances are sampled from the prior rather than learned.

Table C.1: Training hyperparameters shared by all base algorithms.

Category	Parameter	Value
Environment	Parallel environments	16
	Episode length (steps)	180
	Time step Δt	1 s
Optimization	Total env. steps	10^7
	Actor / critic learning rate	10^{-4} / 10^{-4}
	LR decay	None
	Gradient norm clip	2.0
	Actor / critic epochs	4 / 4
	ϵ_{clip} (HAPPO / MAPPO)	0.1
	KL threshold (HATRPO)	0.01
	Entropy coefficient	0.02
Network	Hidden layers	64×64, ReLU
	Recurrent layers	GRU $\times 2$, chunk 15
	Action head	Tanh-squashed Gaussian
	Parameter sharing	No (per agent)
GAE	γ / λ_{gae}	0.99 / 0.95
	Value loss	Huber (δ=2) + ValueNorm

Multi-agent base algorithms and ours. HAPPO, MAPPO, and HATRPO share the 8-D attack surface and the team reward (B.4). HAD^{2}RL (§3) adds densification to each, yielding three variants. Setting $\lambda = 0$ recovers the base algorithm bitwise (Appendix C.2), so every gain is attributable to densification alone. Hyperparameters are listed in Appendix C.

C Implementation Details

This appendix covers shared hyperparameters (C.1), densification parameters and the λ schedule (C.2), weight construction details (C.3), and the full training loop (C.4). All three base algorithms share Table C.1, and HAD^{2}RL adds only Table C.2. Setting $\lambda = 0$ recovers the base algorithm bitwise (C.2).

C.1 Training Hyperparameters

Table C.1 lists the full training configuration shared by all three base algorithms.

C.2 Densification Parameters and λ Schedule

Table C.2 lists the densification parameters of Eqs. 7 to 9. Weighting parameters are shared, with the only exception of MAPPO's higher floor $w_{\min} = 0.5$, and schedule parameters differ by base algorithm. λ **schedule.** Three gates in $[0, 1]$ are multiplied and EMA-smoothed once per rollout:

$$\lambda_k = \beta\, \lambda_{k-1} + (1 - \beta)\, g_k^{\text{on}}\, g_k^{\text{ev}}\, g_k^{\text{den}}, \qquad \lambda_0 = 0.$$

- g^{on} holds $\lambda = 0$ until the K/W condition is met, then ramps to 1 over T_{w} rollouts.
- g^{ev} is 1 for event rollouts and otherwise decays via $\text{clip}(\overline{|\hat{A}_{\text{event}}|}/\bar{a},\, 0,\, 1)$.
- g^{den} (optional) maps the event-density EMA to $[1, 0]$ over $[\rho_{\text{lo}}, \rho_{\text{hi}}]$, pushing $\lambda \to 0$ once failures are no longer rare (Proposition A.3).

GAE reconnection activates only when $\lambda_{k-1} \geq \lambda_{\text{rc}}$.

$\lambda = 0$ **recovers the base algorithm.** Eq. 7 yields $w \equiv 1$ and reconnection is off. The event critic uses an isolated RNG stream, so $\lambda = 0$ trajectories are bitwise identical to the base-algorithm baseline, and every difference in §4 is attributable to densification alone.

Table C.2: HAD^{2}RL densification parameters. Weighting parameters are shared except for MAPPO's floor, and schedule parameters differ by base algorithm.

Symbol	Meaning	HAPPO	HATRPO	MAPPO
α	Long-tail exponent (Eq. 7)	0.5	0.5	0.5
$[w_{\min}, w_{\max}]$	Weight clip range (Eq. 7)	$[0.3, 3]$	$[0.3, 3]$	$[0.5, 3]$
V_{event} input	Same architecture as the dense critic, with agent id	Yes	Yes	Yes
Event advantage	Standard GAE (no reconnection) with ValueNorm	Yes	Yes	Yes
Inert criterion	Past CPA and receding; both alerts off; not NMAC; not done	Yes	Yes	Yes
$n_{\min}$	Minimum NMAC count for an event rollout	1	1	1
K/W	Enable if $\geq K$ of the last W rollouts contain an event (W=0: consecutive)	8/0	5/20	8/0
T_{w}	Rollouts over which λ ramps linearly to 1 after enable	100	50	100
β	EMA coefficient of λ (0 = off)	0	0.8	0
$[\rho_{\text{lo}}, \rho_{\text{hi}}]$	Event-density gate (off if unset)	Off	$[0.1, 0.4]$	Off
λ_{rc}	Minimum λ required to enable GAE reconnection	0	0.05	0
Seq. inert neutralization	Set the inert-step ratio to 1 before multiplying M^i	Off	Off	—
RNG isolation	Event critic uses a private random stream	Yes	Yes	Yes

C.3 Long-Tailed Weight Construction

The main text (§3.3) omits the following conventions for brevity.

- μ^i is the batch mean of $|\hat{A}^i_{\text{event}}|^\alpha$, which normalizes $\hat{w}^i$ to unit mean before clipping. α controls tail sharpness (Table C.2).
- $\bar{\mu}_A$ and $\bar{\sigma}_A$ are reused from the base algorithm's running advantage normalization, so no additional estimation is needed.
- Ordering: normalize first, then multiply by w^i. Because $w^i > 0$, the per-step sign matches $\hat{A}^i_{\text{dense}}$. Clipping shifts the batch mean away from 1, redistributing the advantage budget across time steps.
- Notation: λ (focusing strength, Eq. 7), λ_I (likelihood coefficient), and λ_{gae} (GAE discount) are three distinct quantities.
- Far field: $|\hat{A}^i_{\text{event}}| \to 0$ implies $\hat{w}^i \to 0$, so $w^i \to \max(1-\lambda,\, w_{\min})$.

Inert-step conjunction. $m_t^{\text{inert}} = 1$ if and only if four conditions hold: (i) past CPA and receding; (ii) both corrupted and clean-twin alerts equal 0; (iii) not an NMAC step; (iv) environment not terminated. The conjunction favors under-labeling to avoid discarding critical steps. The critic trains on standard (unreconnected) returns.

C.4 Training Loop

Algorithm 1 gives the full HAD^{2}RL training loop, including the λ schedule and the gating of reconnection. The base algorithm $\mathcal{B}$ can be any heterogeneous multi-agent policy optimizer. When $\lambda = 0$ the loop degenerates bitwise to $\mathcal{B}$.

D Supplementary Evaluation Results

This appendix reports the full evaluation of all 12 methods on the held-out set of 55,000 episodes (5,000 per family). The held-out set shifts absolute conditions (airspace location, encounter orientation, altitude band, and disturbance bounds) relative to training while preserving the relative encounter geometry (Appendix B.2). All methods share the same instance set, the same DAIDALUS configuration, and the same evaluation protocol (Appendix B.5). Methods fall into four groups: *search* (MCTS, MCBEST, CEM, Go-Explore, 100 iterations per instance), *single-agent RL* (PPO-AST with the 3-D intruder only, TRPO-AST with a joint 8-D trust-region policy), *multi-agent base-*

Algorithm 1 HAD^{2}RL training loop (base algorithm $\mathcal{B}$; hyperparameters in Tables C.1 and C.2)

Initialize policies π_θ^I (intruder, 3-D) and π_θ^P (perturber, 5-D), dense critic V_ϕ, event critic V_ψ^E trained on r^E only, and $\lambda \leftarrow 0$.
for rollout $k = 1, 2, \ldots$ **do**
  Collect $\mathcal{D}_k \leftarrow \mathrm{Rollout}(\{\pi^i\}, \mathrm{env})$ with N parallel environments.
  Compute dense advantage $\hat{A}^i_{\mathrm{dense}}$ via GAE on V_ϕ and event advantage $\hat{A}^i_{\mathrm{event}}$ via GAE on V_ψ^E using r^E only.
  Update gates g_k^{on}, g_k^{ev}, g_k^{den} from $\mathcal{D}_k$ and set $\lambda_k \leftarrow \beta\,\lambda_{k-1} + (1-\beta)\, g_k^{\mathrm{on}}\, g_k^{\mathrm{ev}}\, g_k^{\mathrm{den}}$.
  if $\lambda_k > 0$ **then**
    $\hat{w}_t^i \leftarrow |\hat{A}^i_{\mathrm{event},t}|^\alpha / \mu^i$ and $w_t^i \leftarrow \mathrm{clip}\big((1-\lambda_k) + \lambda_k \hat{w}_t^i,\ w_{\min},\ w_{\max}\big)$ (Eq. 7).
    if $\lambda_k \geq \lambda_{\mathrm{rc}}$ **then**
      $m_t^{\mathrm{inert}} \leftarrow \mathbf{1}[\text{past CPA} \wedge \text{no alert} \wedge \text{no NMAC} \wedge \text{not done}]$ and recompute $\hat{A}^i_{\mathrm{dense}}$ with inert TD errors zeroed (Eq. 9).
    end if
    $\tilde{A}_t^i \leftarrow w_t^i \cdot \hat{A}^i_{\mathrm{dense},t}$ (Eq. 8).
  else
    $\tilde{A}_t^i \leftarrow \hat{A}^i_{\mathrm{dense},t}$.
  end if
  Update actors π^I, π^P via $\mathcal{B}.\mathrm{PolicyUpdate}(\tilde{A}^I, \tilde{A}^P)$.
  Update V_ϕ on standard (unreconnected) returns and V_ψ^E on event returns (isolated RNG stream).
end for
return $\{\pi_\theta^I, \pi_\theta^P\}$.

Table D.1: Aggregate metrics pooled over 55,000 held-out episodes. $P(\mathrm{NMAC})$ is the point estimate $\pm$ half-width of the Wilson 95% CI. HMD/VMD are NMAC-episode medians (ft). $P(\mathrm{CRA})$ and $P(\mathrm{LA})$ use hazard episodes as denominator. AAT baseline is 25 s. Reaction budget is the median first-alert-to-CPA time in NMAC episodes. Robust recovery denominator is episodes with an executed avoidance maneuver. RLL is the 11-family mean of per-family NMAC-episode median per-step values.

Method	P(NMAC) (%)	HMD/VMD	P(CRA)	P(LA)	AAT	React.	Alert	Recov.	RLL I/P
MCTS	2.9±0.2	377/62	3.1	82.7	−0.1	37.8	40.4	1.29	−11.6/−2.5
MCBEST	2.9±0.2	374/61	3.0	82.5	−0.1	38.0	40.1	1.34	−11.8/−2.5
CEM	5.0±0.2	377/59	3.2	83.6	+0.7	37.4	44.0	1.17	−10.4/−3.0
Go-Explore	1.1±0.1	366/65	3.6	75.5	+2.2	37.9	30.5	0.95	−11.6/−2.5
PPO-AST	19.4±0.3	377/53	2.3	91.6	−3.8	37.2	52.6	1.21	−0.1/+2.5
TRPO-AST	8.4±0.2	349/36	2.0	77.2	−5.5	38.0	24.1	1.11	−9.8/−3.6
HAPPO	56.4±0.4	292/50	9.0	85.1	−0.0	39.7	42.5	0.44	−0.2/−0.9
MAPPO	57.4±0.4	312/45	15.0	81.5	+1.5	41.2	45.7	0.33	+0.2/−0.6
HATRPO	47.7±0.5	306/57	1.0	94.6	−2.5	41.4	43.8	0.74	+0.4/−0.3
HAD^{2}RL-HAPPO	**69.2±0.4**	326/59	7.1	91.3	+1.3	38.4	45.2	0.33	−0.2/−0.2
HAD^{2}RL-MAPPO	65.8±0.4	309/47	11.4	86.7	+4.8	41.6	46.3	0.29	+0.0/−1.2
HAD^{2}RL-HATRPO	52.0±0.5	321/65	6.1	91.9	+0.6	37.2	47.9	0.67	+0.2/−2.2

lines (HAPPO, MAPPO, HATRPO), and *ours* (HAD^{2}RL on each multi-agent backbone). Metric definitions are in Appendix B.3.

D.1 Aggregate Leaderboard

Collision rate. HAD^{2}RL-HAPPO reaches 69.2% P(NMAC), 12.8 pp above HAPPO (56.4%). All three variants improve on their backbone, with a gain that varies by backbone, and the multi-agent methods lead the best search method (CEM, 5.0%) by 10 to 14 times.

Late alerts do not guarantee collisions. PPO-AST reaches a late-alert rate of 91.6% and TRPO-AST an AAT of -5.5 s, yet their P(NMAC) is only 19.4% and 8.4%, because DAIDALUS guidance still rescues most encounters once alerted, even late.

Table D.2: Per-family $P(\text{NMAC})$ (%). Rarity levels follow the HAPPO baseline (Appendix B.2). Bold: best per row; underlined: second best. Bottom rows: level and pooled means, then the number of families on which each method is best.

		Search				Single-agent RL		MARL			HAD^{2}RL (ours)		
	Family	MCTS	MCBEST	CEM	Go-Exp.	PPO	TRPO	HAPPO	MAPPO	HATRPO	HAPPO	MAPPO	HATRPO
Rare	Overtaken	0.3±0.2	0.3±0.2	0.6±0.2	0.2±0.1	0.0±0.0	0.1±0.1	0.0±0.0	0.0±0.0	0.0±0.0	67.3±1.3	**78.4±1.1**	0.0±0.0
	Overtaking	0.0±0.1	0.0±0.0	0.0±0.1	0.0±0.0	3.2±0.5	0.0±0.0	3.1±0.5	0.9±0.3	0.0±0.1	**22.5±1.2**	0.3±0.2	0.0±0.0
Medium	Head-on	6.0±0.7	5.2±0.6	10.5±0.8	2.2±0.4	65.3±1.3	86.4±1.0	15.8±1.0	7.0±0.7	98.1±0.4	68.9±1.3	11.7±0.9	**99.1±0.3**
	L-oblique	0.1±0.1	0.0±0.1	0.0±0.1	0.0±0.0	0.0±0.0	0.0±0.0	26.1±1.2	0.0±0.0	**5.94±1.4**	17.8±1.1	2.8±0.5	5.6±0.6
	R-oblique	0.3±0.1	0.2±0.1	0.3±0.2	0.1±0.1	0.0±0.0	0.1±0.1	29.4±1.3	65.1±1.3	3.9±0.5	25.5±1.2	**68.7±1.3**	7.7±0.7
	Conv.L-60	0.1±0.1	0.1±0.1	0.3±0.2	0.0±0.1	81.2±1.1	0.0±0.0	83.3±1.0	**87.7±0.9**	77.2±1.2	85.4±1.0	83.0±1.0	69.7±1.3
	Conv.R-60	0.1±0.1	0.1±0.1	0.3±0.2	0.1±0.1	57.6±1.4	0.0±0.0	78.8±1.1	81.7±1.1	0.0±0.0	**84.4±1.0**	81.7±1.1	6.2±0.7
Saturated	Conv.L-120	6.2±0.7	6.4±0.7	10.5±0.9	2.3±0.4	0.0±0.0	3.5±0.5	90.4±0.8	97.3±0.4	94.6±0.6	**99.5±0.2**	99.1±0.3	92.7±0.7
	Conv.R-120	6.3±0.7	6.3±0.7	10.9±0.9	2.5±0.4	0.0±0.0	2.4±0.4	96.1±0.5	92.2±0.7	0.0±0.0	**99.7±0.2**	99.2±0.2	95.5±0.6
	Conv.L-135	6.1±0.7	6.4±0.7	10.8±0.9	2.1±0.4	4.6±0.6	0.0±0.0	99.0±0.3	**99.9±0.1**	95.8±0.6	98.6±0.3	99.7±0.2	95.9±0.6
	Conv.R-135	6.7±0.7	6.3±0.7	10.9±0.9	2.3±0.4	1.8±0.4	0.0±0.0	98.5±0.3	**99.7±0.2**	95.1±0.6	91.7±0.8	98.7±0.3	99.1±0.3
Mean	Rare	0.2±0.1	0.1±0.1	0.3±0.1	0.1±0.1	1.6±0.2	0.0±0.0	1.6±0.2	0.4±0.1	0.0±0.0	**44.9±1.0**	39.4±1.0	0.0±0.0
	Medium	1.3±0.1	1.2±0.1	2.3±0.2	0.5±0.1	40.8±0.6	17.3±0.5	46.7±0.6	48.3±0.6	47.7±0.6	**56.4±0.6**	49.6±0.6	37.7±0.6
	Saturated	6.3±0.3	6.4±0.3	10.8±0.4	2.3±0.2	1.6±0.2	1.5±0.2	96.0±0.3	97.3±0.2	71.4±0.6	97.4±0.2	**99.2±0.1**	95.8±0.3
	All	2.9±0.1	2.9±0.1	5.0±0.2	1.1±0.1	19.4±0.3	8.4±0.2	56.4±0.4	57.4±0.4	47.7±0.4	**69.2±0.4**	65.8±0.4	52.0±0.4
	# best	0	0	0	0	0	0	0	3	1	4	2	1

Naturalness. Search methods produce $\text{RLL}^I \approx -11$, far outside the natural-behavior envelope. Multi-agent baselines and HAD^{2}RL variants stay within $(-1, +1)$, one order of magnitude more natural. HAD^{2}RL-HAPPO ($\text{RLL}^I = -0.2$) matches HAPPO (-0.2) on the intruder channel and moves the perturber channel from -0.9 to -0.2, so densification does not sacrifice naturalness.

Robust recovery. All three HAD^{2}RL variants achieve lower recovery rates than their backbones (0.29 to 0.67% against 0.33 to 0.74%), so the discovered failure trajectories leave DAIDALUS less room to recover, indirect evidence of higher attack quality.

D.2 Per-Family Collision-Rate Breakdown

Rare families. In Overtaken and Overtaking, all baselines achieve $P(\text{NMAC}) \leq 3.1\%$. HAD^{2}RL-HAPPO raises Overtaken from 0% to 67.3% and Overtaking from 3.1% to 22.5%. This is where densification matters most, since the noise reduction of Proposition 2 is largest where critical states are rarest.

Backbone dependence. HATRPO excels in Head-on (98.1%) and L-oblique (59.4%) but fails entirely in Conv.R-60 and Conv.R-120 (0%), where its trust-region constraint traps the policy in a local optimum. HAD^{2}RL-HATRPO inherits the weakness in Conv.R-60 but repairs Conv.R-120 (95.5%), which is rare for HATRPO even though it is saturated for HAPPO (§4.3). In the two overtaking families HATRPO never reaches an NMAC, so the event critic has no failure to learn from and HAD^{2}RL-HATRPO stays at 0%.

Saturated families. In the four 120° and 135° converging families, HAPPO, MAPPO, and the HAD^{2}RL variants all exceed 90%, a systematic vulnerability of the SUT under high-closure geometry.

TRPO-AST and agent decomposition. TRPO-AST controls the full 8-D attack surface with a single joint policy, yet achieves only 8.4% overall, below PPO-AST (19.4%) with its 3-D surface. It learns a head-on pursuit mode (86.4%) but reaches 0% in 8 of 11 families, so single-agent optimization over the joint action space is trapped in a local optimum. The weakest multi-agent backbone (HATRPO, 47.7%) outperforms TRPO-AST by nearly six times, which supports the heterogeneous CTDE decomposition.

D.3 Per-Family RLL Breakdown

Table D.3 reports per-family median per-step RLL for NMAC episodes across the three backbone pairs. $\text{RLL} = 0$ corresponds to the natural prior, and negative values indicate more aggressive deviations.

Table D.3: Per-family median per-step RLL for NMAC episodes across the three backbone pairs. RLL = 0: natural prior; negative: more aggressive; positive: more passive. "—": zero NMACs (RLL undefined). I/P: intruder/perturber channel.

		HAPPO		HAD²RL-HAPPO		MAPPO		HAD²RL-MAPPO		HATRPO		HAD²RL-HATRPO	
Family	Rarity	I	P	I	P	I	P	I	P	I	P	I	P
Head-on	Med	+0.93	−0.06	+0.75	+0.39	+0.74	−0.47	+0.83	−0.62	+1.22	+0.45	+1.24	−1.38
Overtaken	Rare	—	—	+0.24	−0.56	—	—	+0.02	−1.84	—	—	—	—
Overtaking	Rare	−0.53	−0.16	−1.10	+0.43	−0.28	−1.73	−0.29	−2.50	+0.60	+0.47	—	—
L-oblique	Med	−3.06	−2.10	−2.67	−2.18	—	—	−0.50	−1.34	−0.18	−0.10	−0.29	+0.43
R-oblique	Med	−1.22	−0.99	−1.13	+1.08	−2.04	−0.54	−0.72	−2.77	−0.86	−0.63	+0.76	−2.52
Conv.L-60	Med	−0.45	−0.54	−0.49	−0.68	−0.44	+0.25	−1.42	−0.15	−1.07	−1.66	+0.05	−4.36
Conv.R-60	Med	−1.51	−0.43	−0.87	+1.75	+0.15	−1.50	−0.04	−1.34	—	—	−2.11	−2.88
Conv.L-120	Sat	+0.84	−1.42	+0.92	+1.59	+0.27	−0.23	−0.56	−1.80	+1.24	−2.09	−0.48	−3.64
Conv.R-120	Sat	+0.44	−0.88	+0.96	−0.53	+0.87	−0.79	+0.74	−0.91	—	—	+0.66	−2.45
Conv.L-135	Sat	+0.98	−3.30	−0.12	−0.74	+0.92	+0.26	+1.24	+0.89	+1.22	+1.91	+0.74	−1.40
Conv.R-135	Sat	+1.27	+0.39	+0.83	−2.57	+1.19	−0.82	+1.17	−1.20	+1.20	−0.82	+1.12	−1.19

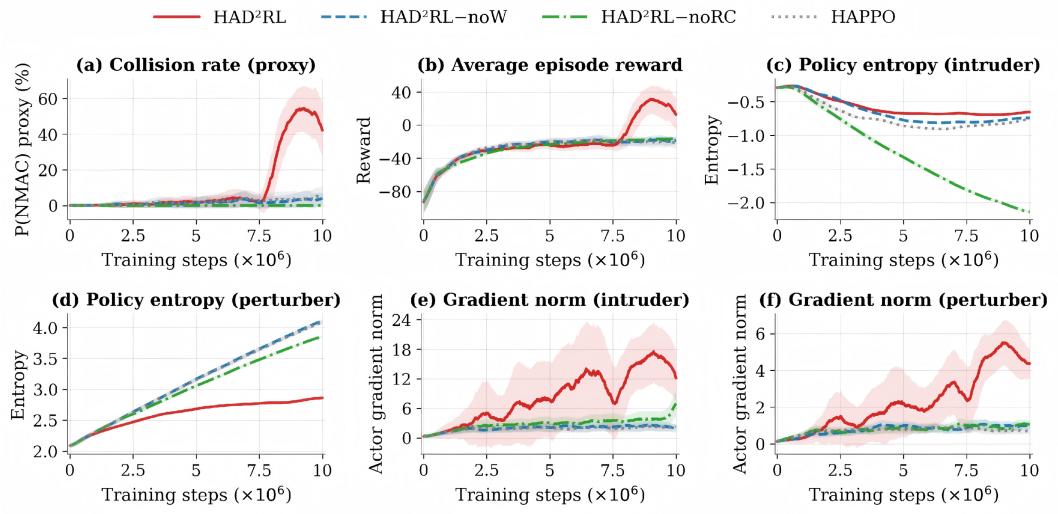

Figure D.1: Training dynamics of the ablations on the HAPPO backbone: (a) collision proxy, (b) average episode reward, (c,d) policy entropy of the intruder and the perturber, (e,f) actor gradient norm of the intruder and the perturber.

Intruder channel. Across all three backbone pairs the mean RLL^I stays near zero: HAPPO -0.23 against HAD^{2}RL-HAPPO -0.24 (10/11 families), MAPPO $+0.15$ against HAD^{2}RL-MAPPO $+0.04$ (9/11), and HATRPO $+0.42$ against HAD^{2}RL-HATRPO $+0.19$ (8/9). Densification does not push the intruder outside the natural-behavior envelope on any backbone.

Perturber channel. In several families HAD^{2}RL flips the perturber RLL from negative (aggressive) to positive (passive), most prominently from HAPPO to HAD^{2}RL-HAPPO in Head-on, Overtaking, R-oblique, Conv.R-60, and Conv.L-120. Critical-state weighting reallocates the disturbance budget, so the perturber no longer applies sustained pressure but concentrates its effect on the few critical steps.

Rare families. In Overtaken all three baselines produce zero NMACs, so their RLL is undefined. HAD^{2}RL-HAPPO reaches RLL^I $= +0.24$ and HAD^{2}RL-MAPPO $+0.02$ there, both within the envelope, so densification triggers failures without resorting to unnatural actions.

D.4 Training Dynamics of the Ablations

Fig. D.1 shows the training curves behind Table 2 in §4.5.

Table D.4: Representative episodes for all 11 families (the four families shown in Figs. D.2 to D.5 and seven more). Method columns: HMD/VMD (ft) at NMAC; *italic*†: no NMAC, values at CPA; ‡: bit-identical trajectories. Δt_W: first-Warning delay under attack relative to the clean twin (s). t_{lead}: first Warning under attack to NMAC (s), DO-365 threshold 25 s. Mode: dominant failure mode of the family (§4.4).

		HMD/VMD (ft)							Δt_W (s)		t_{lead} (s)
Family	Mode	MCTS	MCB	CEM	Go-E	TRPO	HAPPO	**H-H**	HAPPO	**H-H**	**H-H**
Overtaken	B	463/93	430/80	479/9	472/30	497/69	*2623/37*†	421/14	44†	0	32
Overtaking	B	479/78	*650/144*†	487/95	*505/156*†	*4056/6*†	341/91	478/88	0	21	38
Head-on	A	423/78	171/70‡	171/70‡	495/97	150/34	329/88	243/88	9	7	25
L-oblique	C	378/93‡	378/93‡	378/93‡	*265/130*†	*2101/425*†	431/42	252/84	4	3	32
R-oblique	C	470/29	448/73	444/37	438/68	475/89	351/68	281/63	5	4	32
Conv.L-60	C	500/97	394/81‡	394/81‡	443/99	*4950/86*†	245/25	461/71	0	0	51
Conv.R-60	C	460/76	473/7	444/27	447/83	*2512/157*†	107/98	450/79	0	0	45
Conv.L-120	C	399/78	400/61	462/69	444/25	117/87	243/73	155/63	3	32	28
Conv.R-120	C	269/24	399/85	221/97	461/58	499/54	85/56	285/41	0	25	35
Conv.L-135	C	472/94	471/18	253/7	314/95	*7632/221*†	339/27	194/75	24	32	27
Conv.R-135	C	313/82	345/23	272/27	463/74	*5047/21*†	402/44	386/77	25	30	29

D.5 Representative NMAC Trajectories

This subsection complements the three episodes of §4.4. For the three learning-based attackers (HAD^{2}RL-HAPPO, HAPPO, TRPO-AST) we show trajectory figures for four representative families, two rare (Overtaken, Overtaking) and two medium (Head-on, L-oblique), in Figs. D.2 to D.5. Table D.4 characterizes all 11 families and the four search baselines (MCTS, MCBEST, CEM, Go-Explore) numerically, since search trajectories lack the clean-twin alert channel and are qualitatively similar across families. Families are labeled with the dominant failure mode of §4.4.

Episode selection. All episodes are deterministic replays of held-out instances (Appendix B.2). HAPPO and HAD^{2}RL-HAPPO share the same instance in each family, except Overtaking, where the shared instance yields no HAPPO NMAC and HAPPO's instance is found by a 100-seed scan. For Head-on, Overtaken, and Conv.L-120 the HAD^{2}RL-HAPPO episode is the one shown in Figs. 4 to 4. Each search baseline and TRPO-AST replays the first instance on which it reached NMAC in its own held-out evaluation (search re-run with 100 iterations and the evaluation RNG seed). Cells in which a method found no NMAC show a non-NMAC episode, marked by its closest point of approach (CPA). The episodes therefore illustrate how each method fails when it does, and how often it fails is given by Table D.2, where 0.0% still admits a few NMACs ($< 0.05\%$).

Reading guide. Each row has four panels: (1) horizontal track, with the ownship's actual and still-air paths; (2) altitude; (3) horizontal separation with the well-clear (4,000 ft) and NMAC (500 ft) thresholds; (4) DAIDALUS alert level under attack (red). For the two multi-agent attackers (HAPPO, HAD^{2}RL-HAPPO) panel (4) additionally shows the clean twin (green dashed), a second DAIDALUS instance fed the true, uncorrupted states of the same trajectory, with a red band marking the interval in which the attack suppresses the alert. The title reports the NMAC (star) or CPA time with HMD/VMD. We write Δt_W for the delay of the first Warning under attack relative to the clean twin, and t_{lead} for the interval from the first Warning under attack to NMAC, for which DO-365 requires $t_{\text{lead}} \geq 25$ s. As Δt_W requires the clean twin, Table D.4 reports it only for the two multi-agent attackers.

In three cells the search-baseline trajectories are bit-identical (Head-on: MCBEST = CEM; L-oblique: MCTS = MCBEST = CEM; Conv.L-60: MCBEST = CEM), marked ‡ in Table D.4. All four search methods share the nominal proposal prior and seed their RNG with the instance seed, so early rollouts coincide, and when the highest-return NMAC lies in this shared prefix the replayed failure is identical.

D.5.1 Cross-Method Summary

Alerting is delayed but rarely late. HAD^{2}RL-HAPPO delays the first Warning in 8 of 11 families ($\Delta t_W = 3$ to 32 s) and HAPPO in 7 of 11 (3 to 44 s). Yet in all 11 HAD^{2}RL-HAPPO episodes

the Warning still precedes NMAC by at least the 25 s threshold ($t_{\text{lead}} = 25$ to 51 s). The attack thus consumes alerting margin rather than causing an outright late alert. The NMAC is completed by the low-closure geometry (Mode B) or by guidance defeat (Mode C), and only in Head-on does the delay reach the threshold itself (Mode A).

Multi-agent attackers fail the SUT faster. In the four non-head-on Medium families, HAPPO and HAD^{2}RL-HAPPO reach NMAC at $t = 45$ to 82 s, whereas the search baselines and TRPO-AST need $t = 74$ to 167 s. The search failures arise late in the horizon and rely on intruder deviations far outside the nominal envelope ($\text{RLL}^I \approx -11$, Table D.1).

TRPO-AST. TRPO-AST reaches NMAC in 5 of 11 cells. Its Head-on and Conv.L-120 episodes are the only NMACs in the set whose Warning trails the 25 s threshold ($t_{\text{lead}} = 12$ and 17 s), consistent with the single intruder-pursuit mode noted in Appendix D.2. In Conv.L-135 and Conv.R-135 it never triggers a Warning.

D.5.2 Rare Families (Mode B)

Overtaken (Fig. D.2). All search and TRPO-AST episodes reach NMAC late ($t = 112$ to 136 s), although each method does so on at most 0.6% of instances. HAPPO delays the first Warning by 44 s yet never closes below 2,623 ft, matching its 0.0% rate. HAD^{2}RL-HAPPO (the episode of Fig. 4) leaves the Warning undelayed and 32 s ahead of NMAC, above the 25 s threshold, and the low closure rate nonetheless leaves guidance too little separation to resolve the encounter (Mode B).

Overtaking (Fig. D.3). Among the search baselines only MCTS and CEM reach NMAC (on $< 0.05\%$ of instances). MCBEST and Go-Explore stop just outside the NMAC cylinder (650/144 and 505/156 ft), and TRPO-AST never comes within 4,000 ft. HAPPO and HAD^{2}RL-HAPPO both reach NMAC at $t = 64$ s. In the HAD^{2}RL episode the attacked alert oscillates between Corrective and None while the clean twin already warns, delaying the Warning by 21 s, so the low-closure blind spot of Mode B is compounded by perception-induced suppression.

D.5.3 Medium Families (Modes A and C)

Head-on (Fig. D.4). All seven methods reach NMAC at $t = 96$ to 99 s with HMD 150 to 495 ft, and MCBEST and CEM return the same trajectory. HAPPO and HAD^{2}RL-HAPPO delay the Warning by 9 s and 7 s, compressing the Warning-to-NMAC interval to exactly the 25 s threshold, the Mode A signature of Fig. 4, which shows the same HAD^{2}RL episode. TRPO-AST, whose 86.4% rate stems from a single intruder-pursuit policy (Appendix D.2), yields a Warning only 12 s before NMAC.

L-oblique (Fig. D.5). MCTS, MCBEST, and CEM return one identical trajectory that reaches NMAC only at $t = 167$ s, near the end of the 180 s horizon. Go-Explore misses vertically (VMD 130 ft) and TRPO-AST ends 2,101 ft apart. HAPPO and HAD^{2}RL-HAPPO reach NMAC at $t = 45$ and 50 s. Their Warning is delayed by only 4 s and 3 s and still precedes NMAC by 35 s and 32 s, so the failure lies downstream of alerting (Mode C).

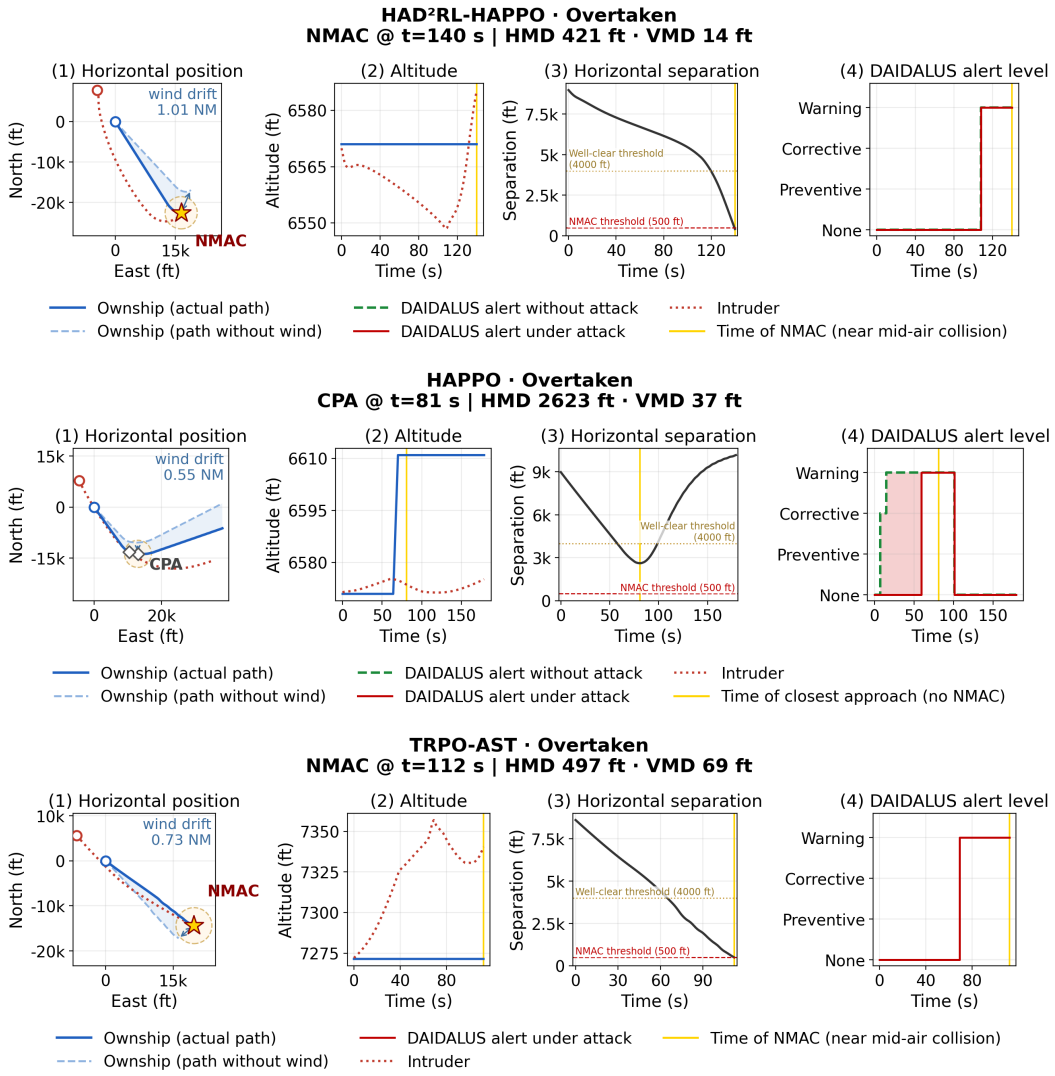

Figure D.2: Overtaken (Rare, Mode B), learning-based attackers. Top to bottom: HAD^{2}RL-HAPPO (67.3%), HAPPO (0.0%, no NMAC), TRPO-AST (0.1%). Percentages: held-out $P(\text{NMAC})$ (Table D.2). Panels as in Fig. 4.

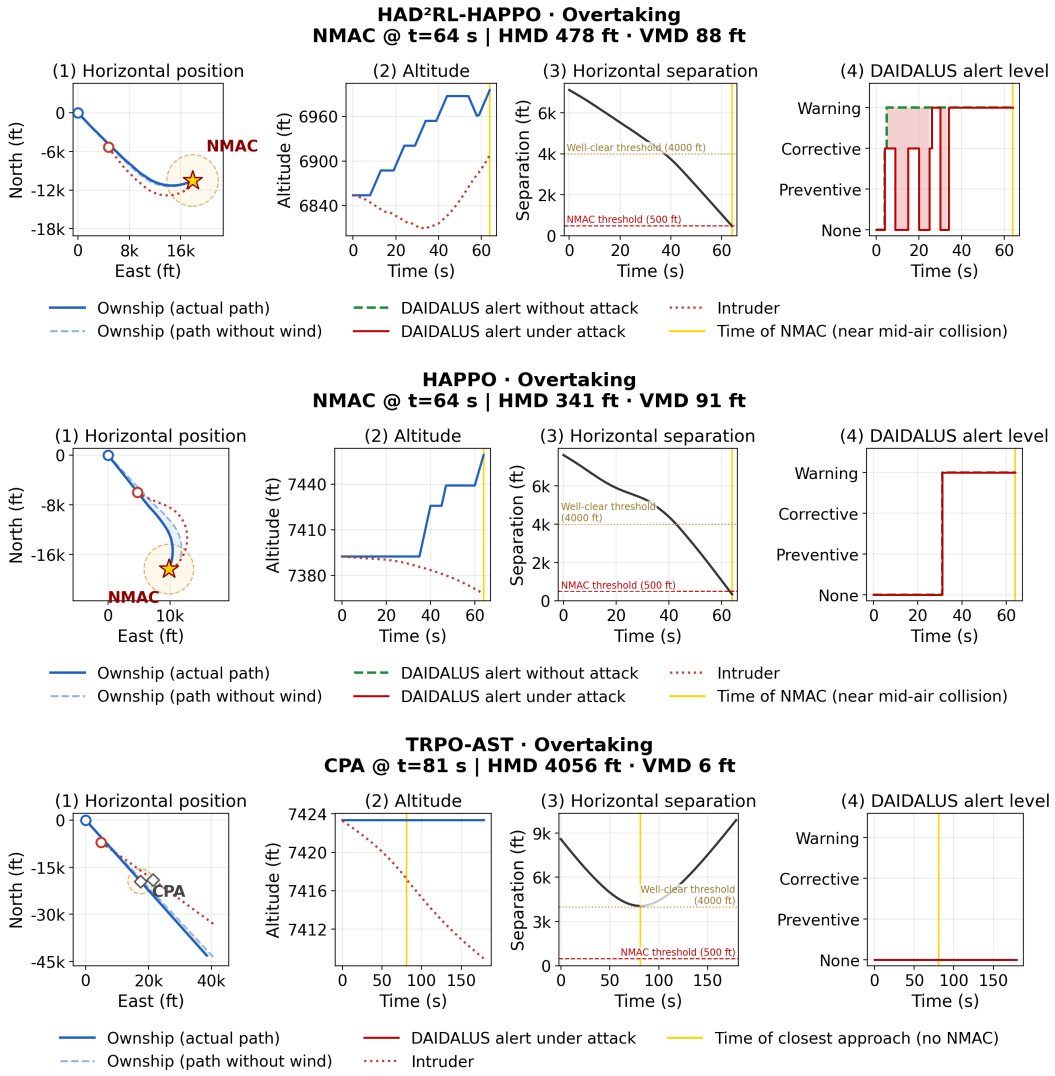

Figure D.3: Overtaking (Rare, Mode B), learning-based attackers. Top to bottom: HAD^{2}RL-HAPPO (22.5%), HAPPO (3.1%), TRPO-AST (0.0%, no NMAC). Percentages: held-out $P(\text{NMAC})$ (Table D.2). Panels as in Fig. 4.

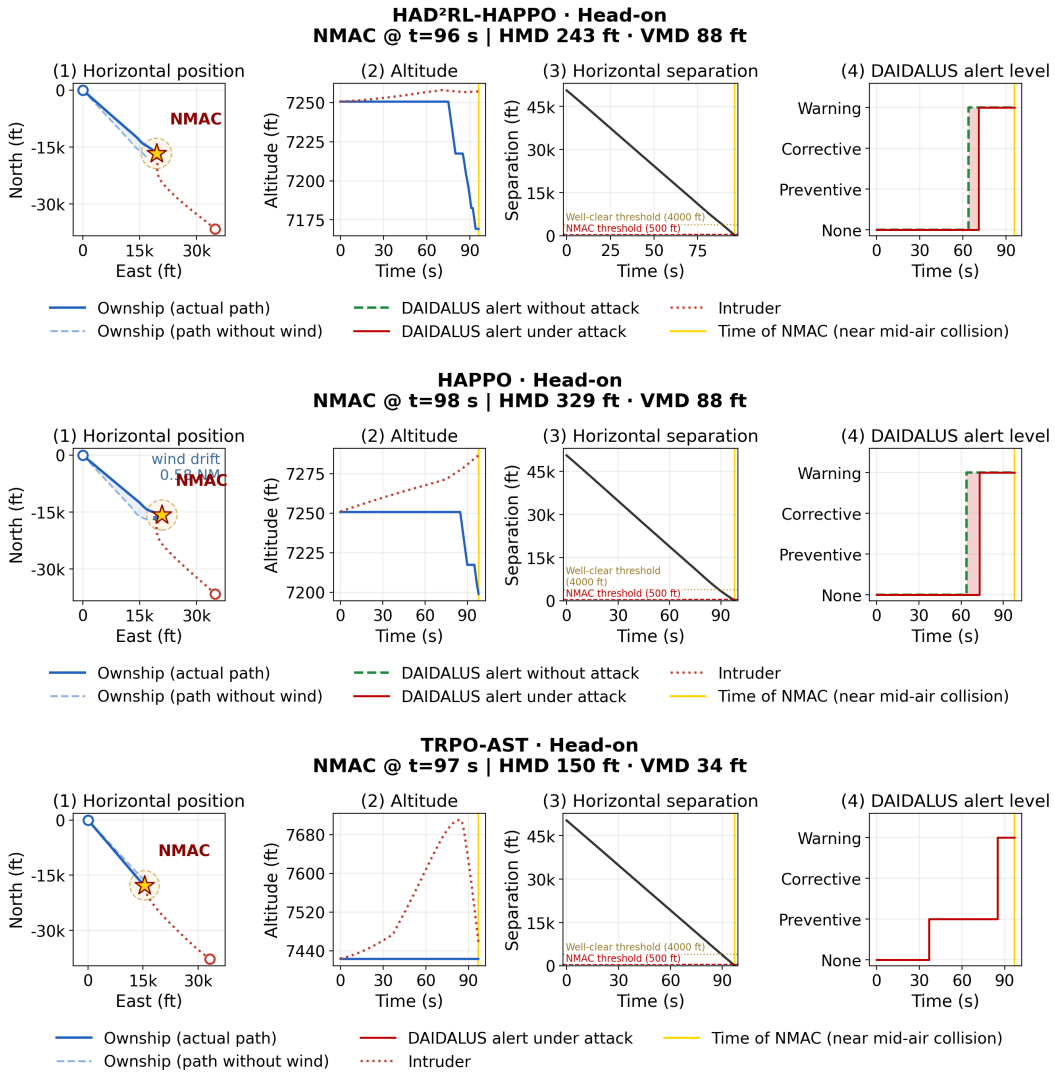

Figure D.4: Head-on (Medium, Mode A), learning-based attackers. Top to bottom: HAD^{2}RL-HAPPO (68.9%), HAPPO (15.8%), TRPO-AST (86.4%). Percentages: held-out $P(\text{NMAC})$ (Table D.2). Panels as in Fig. 4.

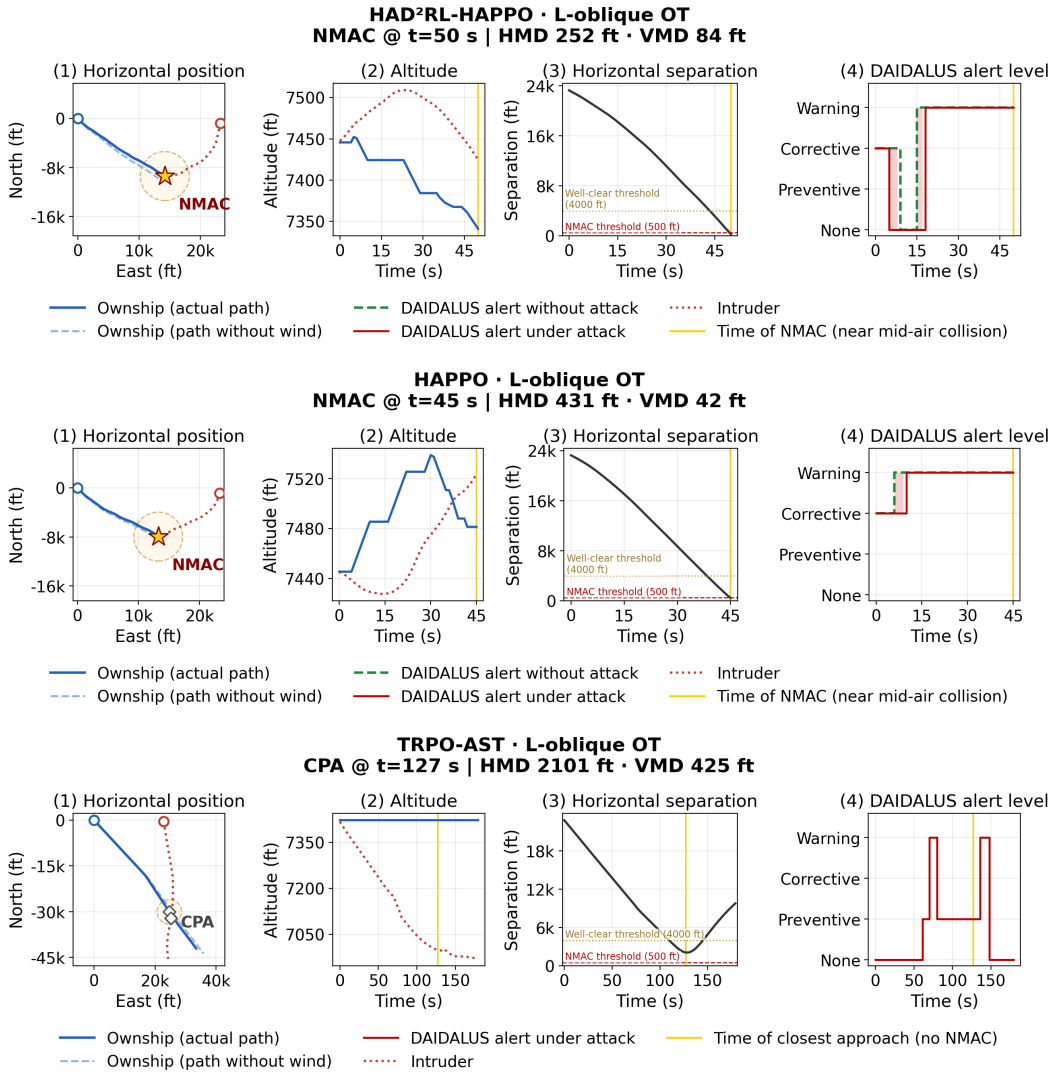

Figure D.5: L-oblique (Medium, Mode C), learning-based attackers. Top to bottom: HAD^{2}RL-HAPPO (17.8%), HAPPO (26.1%), TRPO-AST (0.0%, no NMAC). Percentages: held-out P(NMAC) (Table D.2). Panels as in Fig. 4.