Consensus Within, Competition Between: Decentralized LLM Teams for Perimeter Defense

Khawaja Abaid Ullah[1], Ananya Acharya[1], Mohammad Javad Khojasteh[1]

[1]Department of Electrical and Microelectronic Engineering, Rochester Institute of Technology, 1 Lomb Memorial Drive, Rochester, 14623, NY, United States.

Emails: ka4150@rit.edu, aa2334@rit.edu, mjkeme@rit.edu

Abstract

The use of large language models (LLMs) to coordinate multi-agent teams raises a fundamental architectural question: whether task allocation should be centralized in a single LLM controller or distributed across autonomous LLM agents, particularly when communication is unreliable. In a deliberately simplified perimeter-defense simulation against LLM-controlled intruders, we compared a central commander, an auction-based consensus team and independent agents under five network conditions. Breach rates were 15% for the consensus team, 32% for the commander and 29% for independent agents, and of the different network conditions, only limited radio range changed a team's breach rate, raising the commander's. In consensus team, giving a low-cost LLM more authority over the bid computation raised breaches by 38 percentage points. With birds and civilian drones in airspace, we added a photograph for each to the input text state. Only the near-frontier model treated them apart from intruders. Intruders disguised as existing taxidermy bird drones were mistook as birds by both low-cost and near-frontier LLMs, going unpursued in 59–61% of near-frontier decisions (1–16% before). However, a one-sentence forewarning in the prompt lowered it to 0–25%. We find that LLM team with a dedicated external conflict resolution protocol results in fewest breaches. Code: https://github.com/the-kas-lab/consensus-within-competition-between.

Keywords: LLM agents, decentralized planning, multi-agent task allocation, cyber-physical systems

1 Introduction

Large language models (LLMs) increasingly act as the decision-makers of multi-agent teams, from software agents that solve tasks together [1–4] to robot teams that divide work and plan joint actions [5, 6]. In robotics, this division of work is studied as multi-robot task allocation, the question of which robot should execute which task [7]. Many LLM teams allocate tasks through a central component, such as an orchestrator that delegates work or a single LLM that assigns subtasks to the whole robot team [5, 8]. Central coordination brings scalability bottlenecks and a single point of failure, which leaves the team exposed when the coordinator fails or becomes overloaded [8]. Decentralized multi-agent LLM systems remove this component. Each agent decides from its own local information and communicates directly with its peers [9]. Without an orchestrator, the agents of AgentNet [8] reached higher accuracy than centralized multi-agent baselines on mathematics, logical question-answering and function-calling benchmarks, and robot arms with one LLM agent each proposed sub-task plans and paths through dialogue [6]. The evidence is not one-sided. In multi-robot planning tasks, Chen et al. [10] compared four designs for coordinating the LLMs of a robot team. The robots in the study cooperated without opponents, and each LLM agent was assumed to know the full environment. In the decentralized architecture, agents relay messages through a peer-to-peer network and achieve consensus via sequential debate rounds—iterative challenge-response exchanges designed to verify reasoning steps and organically converge on consistent information. This setup had the lowest success rate, below a central planner, while hybrid designs did best and the central planner scaled worst as robots were added. In a separate study, Kim et al. [11] evaluated 260 agent-system configurations and found that independent agents, which do not communicate, amplified errors more than centrally coordinated ones, and whether coordination helped depended on the task. Network degradation and agent attrition are unavoidable in distributed multi-agent systems, so we directly compare centralized and decentralized LLM coordination under these conditions. We study this problem in an adversarial game that serves as a simple abstraction of a cyber-physical system, aiming toward applications where an LLM in a hierarchical control setting decides and allocates tasks as an intruder approaches.

In perimeter defense, a team of defenders must intercept intruders before they reach a protected region [12], and the intruders score for each one that gets through [13]. We call an intruder's entry into the protected region a breach. Perimeter defense belongs to the broader family of pursuit-evasion and reach-avoid differential games [14], in which a team of intruders attempts to reach a target region while defenders attempt to capture or neutralize them [15–18]. Game-theoretic analyses build team strategies by decomposing the game into local games between small groups and assigning defenders to intruders [13], and learned policies imitate a maximum-matching assignment from each defender's local perception [19]. In scenarios such as counter-drone defense, defenders must detect intruders with limited sensing [20], share observations over a communication network [21], and cope with message loss over non-ideal channels [22]. Assigning defenders to intruders is therefore an allocation decision made under partial and evolving information, where the contextual reasoning of LLM planners can be beneficial. Acharya et al. [23] propose a hierarchical architecture in which

a per-team LLM planner assigns tactical roles to robots under partial observability, with execution delegated to low-level model predictive controllers. Foundation models [24], including large language models (LLMs) and vision-language-action models, have been applied across many aspects of robotics, from semantic perception to end-to-end control [25, 26]. A common pattern is hierarchical: a language model plans at a high level while lower-level skills or controllers execute the plan [27, 28]. One reason is speed as inference latency is a recognized obstacle to deploying foundation models on robots [25]. Our study follows a hierarchical control design (cf. [29]): in our study a call to the LLM planner took a median of 2.0 s, while model predictive control (MPC) [30, 31] converts each agent's task into accelerations every 0.1 s, with a barrier constraint keeping agents collision-free except for when a defender is pursuing to neutralize an intruder. A second design choice is how much of each decision the LLM makes itself rather than leaving it to fixed rules, which we call its authority. At *low authority* the LLM chooses among options written in advance, as when SayCan [27] selects a robot's next skill from a fixed set, which makes each step of the plan interpretable. At *high authority* it sets numbers that a solver then uses, as when an LLM writes the weights of a reward that MPC optimizes [32]. That design solved more robot tasks than an LLM that composed predefined skills, and its authors note that constraining what the LLM may set makes the system more stable at some cost in flexibility [32]. Our aim is to study how LLM-driven teams compete against each other in an adversarial game, and how centralized, and decentralized planners, with different levels of authority, compare under different communication conditions.

How a team allocates tasks determines what it needs from its communication network. A central allocator can in principle find the best assignment by gathering all information and planning for the whole team, but it has high communication demands and is a single point of failure [33]. When radio range is limited, reports and orders must be relayed, and a partitioned network cuts agents off from both. Fully distributed allocation avoids this round trip, but good local choices may not add up to a good team assignment [33], and agents that each pick their best task can pick the same one [7]. Market-based methods combine the two by letting agents bid for tasks [33]. The consensus-based auction algorithm (CBAA) reaches a conflict-free assignment through local message exchange alone and tolerates inconsistent information across the team [34]. Auction methods with conventional bids have been tested under lossy and fading communication [22, 35], and LLM teams have been compared under centralized, decentralized and hybrid communication on cooperative planning tasks [10]. By framing the problem as an adversarial game, we let teams of LLMs compete directly against one another, and we compare a central LLM commander with decentralized LLM agents that resolve allocation conflicts through a consensus auction, under progressively degraded communication networks. We also evaluate both architectures when some agents submit falsified estimates (Section 4.4).

In a typical airspace, defender agents share the sky with birds and civilian drones, and detecting drones, birds, and other distractors is the subject of a dedicated benchmark challenge [36]. A textual state-based only defender treats birds, civilian drones and intruders the same way, i.e. as contacts. We refer to birds and civilian drones as *non-hostile contacts*. Without a recognition and classification mechanism, the defender

cannot tell the intruders apart from the non-hostile contacts. And, pursuing a non-hostile contact costs more than just time as a defender that touches any contact is lost, resulting in reduction of defensive capacity. Multimodal LLMs, which accept images in addition to text, can inspect a contact before deciding whether to engage it, but whether this ability helps the team depends on two factors: the model must correctly classify the object, and the coordination interface must allow it to act on that judgment. Using image-based contact classification, we consider four contact types (birds, civilian drones, intruder drones, and disguised intruder drones) and ask how centralized and decentralized LLM teams, each run with both a low-cost and a near-frontier model, handle these contacts under the communication conditions studied above. We deliberately keep the perception component simple: the aim is a reproducible benchmark for multimodal LLM decision-making in adversarial games, in which the effects of team architecture and communication degradation can be isolated from perception difficulty.

Here we compare three LLM-coordinated defender teams that differ in where LLM decision-making is placed within the defensive architecture (Fig. 1). The first, the *commander team*, places all authority in a central commander, whose LLM assigns tasks to every agent. The second, the *consensus team*, has decentralized agents that bid for tasks and agree on who takes which through a consensus-based auction. Each agent's LLM adjusts its own bids according to its authority. At low authority it can only choose a weight profile, a priority for each intruder and up to three preferred tasks. At high authority it sets the weights of the bid formula and a multiplier from 0 to 2 for each intruder's pursuit value. The third, the *independent team*, uses the same LLMs and bids, but each agent chooses its task independently based on its own highest bid, without an auction.

Each team defends a region for 40 seconds against LLM-controlled intruders from 30 random starting configurations (seeds), with two and four intruders, under five network conditions (an ideal network, independent message loss, burst message loss, a compromised defender that falsifies reports, and limited radio range with relaying). Four lessons emerged. First, conflict resolution mattered. Across the four network conditions without falsified reports, intruders breached the region in 15% of episodes against the consensus team, 32% against the commander team, and 29% against the independent team, with most of the difference arising in episodes with four intruders. Second, the commander team was the most sensitive to network conditions. Its breach rate rose by 18 percentage points (95% interval: +1 to +36) when radio range was limited, whereas message loss and falsified reports did not result in any significant differences between the architectures. Third, level of authority assigned to LLM for decision making should depend on the capability of LLM being used as with a low-cost model, high authority raised the number of breaches by 38 percentage points, however, a near-frontier model under high authority did not see an increase in breach rate. Fourth, non-hostile contacts, which tested the models' visual-language capabilities, showed the same divide. Across five seeds, the near-frontier model never lost a defender to a non-hostile contact in either the centralized or the decentralized role, whereas each low-cost model team, lost 3 to 5 defenders in total across the five episodes.

These lessons come from a testbed that is deliberately simplified, so that coordination can be studied under control. Agents move as point masses with bounded acceleration in a two-dimensional arena of abstract units, with no wind or other disturbance. They sense opponents without noise and without missed detections inside a fixed field of view. The photographs are reference pictures rather than views from an onboard camera. We make no claim that this setup represents a deployed counter-drone system. Its value is as a benchmark for LLM-driven coordination, a simplified abstraction of cyber-physical teams. In summary, we make the following contributions.

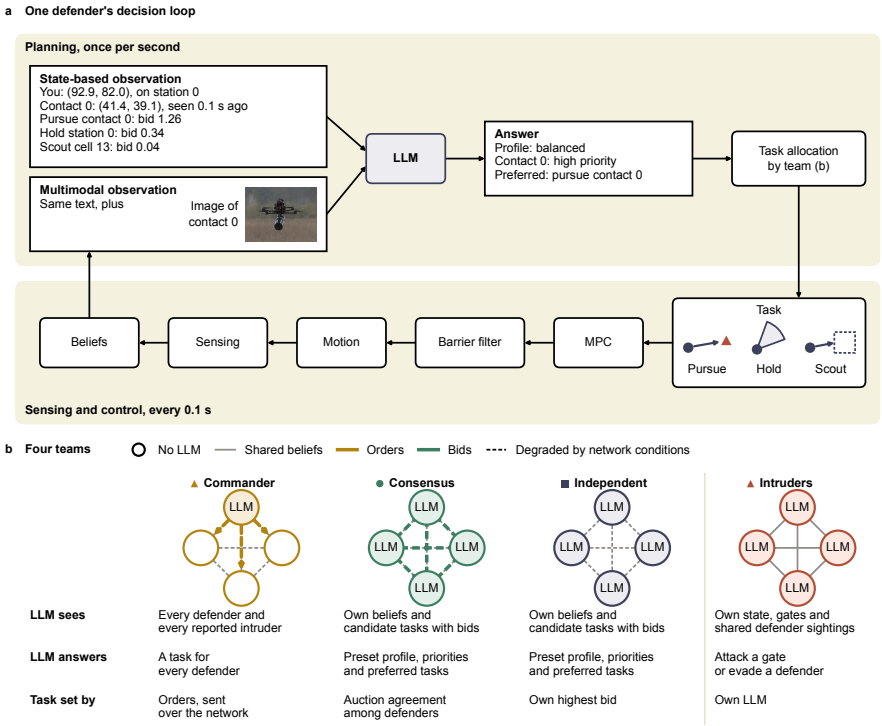
- We establish a benchmark for multi-agent LLM coordination in a perimeter-defense game, in which LLM-driven defenders face LLM-controlled intruders over a network that can be degraded. The LLMs are used as released, without fine-tuning, task-specific training or example answers, and every episode can be re-simulated from its recorded model answers.
- We compare central and decentralized LLM coordination on matched episodes under message loss, limited radio range and falsified reports, which gives the first two lessons.
- We test how much an LLM should be allowed to decide, by running the consensus team at high authority with a low-cost and a near-frontier model, which gives the third lesson.
- We extend multi-agent LLM coordination to multimodal, image-based contact classification, with birds, civilian drones and intruders disguised as taxidermy bird drones.

2 Results

We first describe the testbed, the three defender teams, the five network conditions and the outcome we measure. The next three subsections compare these three teams, all run with the low-cost model on the same 30 seeds. They ask how often each team was breached, how well it divided its tasks among its defenders, and how it responded to message loss, falsified reports and limited radio range. We then run the consensus team at high authority, and run the commander team and the consensus team at high authority with a near-frontier model as well. Last, we add birds and civilian drones to the airspace, show the defenders' LLMs a photograph of every contact and then disguise the intruders.

2.1 Teams, conditions and outcome

We built a testbed in which four defenders protect a region against two or four intruders, and every agent on both teams uses the same low-cost LLM, Gemini 3.1 Flash-Lite, once per second, in decision epochs of 1 s (Fig. 7; Section 4). A defender's task is to pursue an intruder, hold one of four stations around the region or scout part of the arena, and an intruder's task is to attack the region or evade a defender. Model predictive control (MPC) [31] turns each agent's task into motion every 0.1 s, and a barrier filter adjusts that motion to keep agents apart except for when a defender is pursuing to neutralize an intruder, in which case the barrier constraint of defender is switched



Supplementary Fig. 1 How the defender and intruder teams decide. (a) One defender's decision loop, shown with one recorded call of a consensus-team defender, shortened and in plain words. Once per second the LLM receives an observation built from the defender's beliefs: state-based in the main study, a text description of the game, and multimodal in the airspace challenges, the same text plus a photograph of every contact. How its answer becomes a task differs by team (See **b**). Every 0.1 s, model predictive control (MPC) plans motion for the task, the barrier filter adjusts that motion to keep agents apart, reading the true positions of all agents, and the sensor updates the beliefs, which teammates share. **(b)** Which team members run an LLM, what they send each other, and the steps from observation to task. The consensus and independent teams' LLMs have low authority (Methods). The network conditions affect only messages between defenders. Photograph: ArmyInform (CC BY 4.0, cropped).

off for only the target being chased. Each agent sees opponents only inside a narrow sensor wedge, so it acts on its beliefs. A belief holds an opponent's last observed position and velocity, from the agent's own sensor or a teammate's message, projected forward based on the velocity and time of observation. The three defender team configurations considered in this study differ how their beliefs are turned into tasks (Fig. 1, Supplementary Videos 1–3). In the commander team, the defender with the lowest index (by default defender 0) is assigned the commander role, and its LLM assigns a task to every defender, including itself. In the consensus team, each defender scores every task it could take, and this score is its bid for the task. The defender's LLM sees its beliefs and these candidate tasks and, at low authority, adjusts the bids only by choosing among options written in advance, and the bids enter the consensus-based

auction algorithm (CBAA) [34]. The independent team runs the same agents with the auction messages switched off, so each defender commits to its own highest bid.

Each team ran under five network conditions: an ideal network, independent message loss, burst message loss with the same average rate, falsified reports from one compromised defender, and a radio range of 15 units that forces relaying (defenders on their stations are 11 to 18 units apart). With two and four intruders across 30 seeds, this gave 900 episodes (Section 4). Each comparison is a mean paired difference over the 30 seeds with a 95% confidence interval.

A breach occurs if an intruder enters the defended region within the 40 s episode. Breaches were rare with two intruders (35 of 450 episodes) and common with four (196 of 450). Many episodes without a breach ended at the time limit with an intruder still alive, 126 of 254 with four intruders and 78 of 415 with two, and 113 of the 161 intruders alive at the end of four-intruder episodes were still closing on the region. Breach rates therefore only count breaches within 40 s.

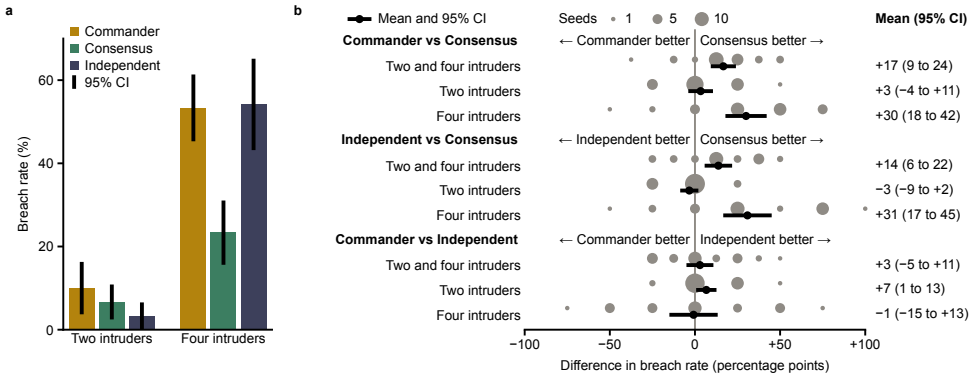
2.2 Consensus-based allocation led to fewer breaches

The consensus team was breached least often (Fig. 2). Over the four conditions without falsified reports, intruders breached the region in 32% of commander team episodes, 29% of independent team episodes and 15% of consensus team episodes (240 per team). The commander team's breach rate—share of episodes with a breach—was 17 percentage points higher than the consensus team's (95% confidence interval: +9 to +24), and the consensus team's was 14 points lower than the independent team's (95% CI: −22 to −6). No difference was detected between the commander and independent teams (95% CI: +3 points, −5 to +11).

The differences were concentrated in episodes with four intruders (Supplementary Table 2). With four intruders, the commander team's breach rate was 30 percentage points higher than the consensus team's (95% CI: +18 to +42), and the independent team's was 31 percentage points higher (95% CI: +17 to +45). With two intruders, the commander team's breach rate was 3 percentage points higher than the consensus team's (95% CI: −4 to +11), and the consensus team's was 3 percentage points higher than the independent team's (95% CI: −2 to +9).

2.3 Duplicated commitments and missed scouting

The auction removed duplicate commitments, in which two or more defenders commit to the same task, such as the pursuit of one intruder (Fig. 3a and Supplementary Table 3). The commander team instead showed missed scouting, in which defenders hold their stations rather than search the arena for intruders that no defender has seen (Fig. 3d). We checked for duplicate commitments once per second, at the end of each decision epoch, using the tasks the defenders were carrying out at that moment. On the ideal network, two or more consensus-team defenders were never committed to the same task (0 of 2,035 checks), whereas two or more independent-team defenders were in 1,288 of 1,896 checks (68%). Across the ideal network, both loss conditions and the falsified-report condition, the consensus team had duplicate commitments in only 10 of 7,988 checks, with no seed above 1.1%, whereas the independent team's

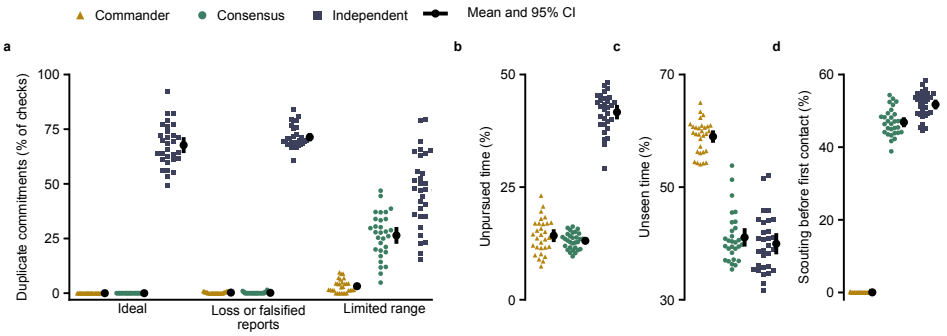


Supplementary Fig. 2 Consensus allocation lowered breaches. (a) Breach rate of each team with two and with four intruders, over the four network conditions without falsified reports (120 episodes per bar). **(b)** Difference in breach rate between the two teams of each block, the first-named team's rate minus the second's, for both intruder counts together and for each; "better" means fewer breaches. Grey circles are the per-seed differences, with area proportional to the number of seeds sharing a value. Black dots are means over the 30 seeds; black bars in **a** and **b** are 95% CIs from the t distribution over seeds. CI, confidence interval.

per-seed share ranged from 61% to 81%. Under limited range, duplicate commitments returned: they occurred in 27% of consensus-team checks (508 of 1,906) and 48% of independent-team checks, and the consensus team's share was lower in 25 of the 30 seeds.

Episode analysis also revealed a passive behavior in the commander team, which we call missed scouting: defenders held their stations instead of actively searching the arena for intruders. We found that duplicated pursuit co-occurred with intruders that no defender pursued (Fig. 3b). Over the four conditions without falsified reports, a known intruder had no defender pursuing it for 42% of its time in the independent team and 13% in the consensus team. The consensus team had 28.6 percentage points (95% CI: −30.1 to −27.0) lower unpursued time than the independent team. In the commander team it was 14%, 1.1 percentage points above the consensus team (95% CI: −0.2 to +2.5). Independent-team pursuers were also crowded by their teammates. Over all five conditions, teammates or disabled defenders changed an independent-team pursuer's acceleration through the barrier filter on 19% of its control steps, against 7% for the consensus team.

Missed scouting originated with the commander's LLM, which assigned scouting in only 16 of 39,077 answered tasks. Before the team's first contact, its defenders scouted in 58 of 194,566 defender-steps (one defender during one 0.1 s step) and otherwise held their stations, whereas defenders in the two decentralized teams scouted in 47% to 54% of their defender-steps before first contact. Consistent with this rare scouting, intruders went unseen by every defender for 59% of their active time against the commander team, compared with 41% against the consensus team (Fig. 3c).

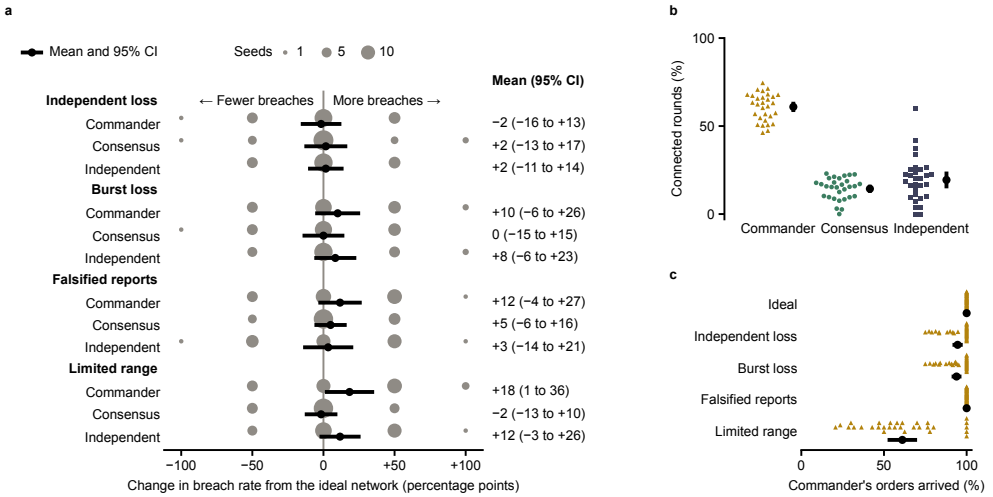


Supplementary Fig. 3 The auction removed duplicate commitments, and the commander barely scouted. (a) Share of the once-per-second checks at which two or more active defenders were committed to the same task, by network condition; independent loss, burst loss and falsified reports are pooled. For the commander team, a defender's commitment is the newest order it has accepted. **(b)** Unpursued time: share of the time a known intruder had no defender pursuing it. **(c)** Unseen time: share of an intruder's active time in which no defender saw it. **(d)** Share of defender time spent scouting before the team's first contact. **b** and **c** cover the four conditions without falsified reports, **d** all five conditions; every panel pools two and four intruders. Each team-coloured marker is one seed; black dots and bars are means and 95% CIs from the t distribution over the 30 seeds. CI, confidence interval.

2.4 Limited range harmed the commander

No team's breach rate changed noticeably when messages were lost or reports were falsified. Under both random and burst loss, every change from the ideal network had a confidence interval that included zero (Supplementary Table 4). These intervals were wide, however, spanning about 30 points, and the commander and independent estimates leaned toward more breaches under burst loss, so the results do not show that the teams were unaffected. In the attack condition, one randomly chosen defender other than defender 0 falsified the positions in its own reports about one intruder. It did so in 55 to 60 of each team's 60 attacked episodes, and in the commander team, defender 0, the commander, accepted at least one falsified report in 55 episodes. Pooled over teams and intruder counts, the attack changed the breach rate relative to the ideal network by +7 points (95% CI: −2 to +15), again no noticeable change but an estimate leaning toward more breaches.

Limited range fragmented the decentralized teams' networks the most, yet only the commander team's breach rate clearly increased (Fig. 4). With a communication range of 15 units, the commander team's network was connected on 61% of communication rounds, compared with 14% for the consensus team and 19% for the independent team. With only two intruders, consensus team's network was connected on only 2% of rounds. The commander team's breach rate rose by 18 points (95% CI: +1 to +36), a change 20 points larger than the consensus team's (95% CI: +2 to +38).



Supplementary Fig. 4 Only limited range changed a team's breach rate, the commander's. (a) Change in breach rate from the ideal network under each other network condition, by team, pooling two and four intruders (two episode pairs per seed). Grey circles are the per-seed changes, with area proportional to the number of seeds sharing a value. **(b)** Under limited range, the share of communication rounds in which the active defenders formed one connected group; each marker is one seed, its two episodes averaged. **(c)** Of the commander's orders that would have changed a defender's task, the share that arrived, by network condition; each triangle is one seed. Black dots and bars are means and 95% CIs from the t distribution over the 30 seeds. CI, confidence interval.

2.5 Low-cost model with high authority resulted in more breaches in consensus team

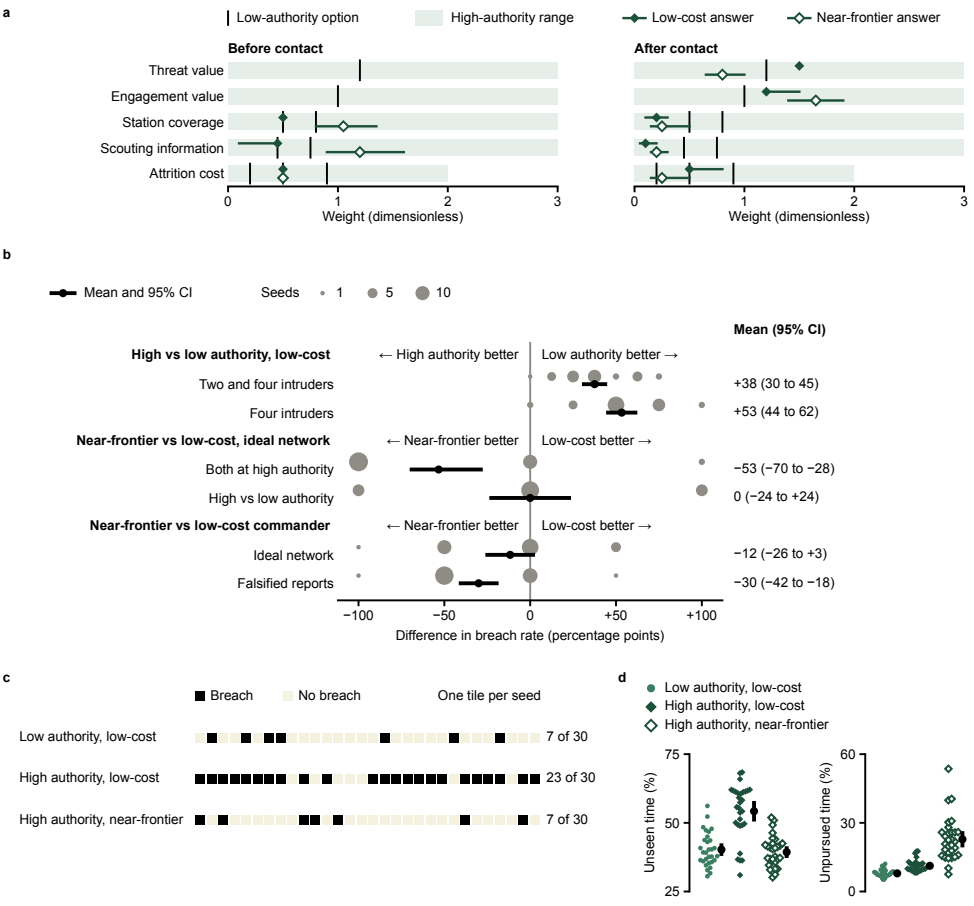
Low authority keeps the LLM from dropping a known intruder or rewriting the formula's tradeoffs. We therefore also study how the consensus team performs when its LLMs have more control, which helps determine how much control LLMs should be given in a given scenario. For this purpose, we ran the consensus team with high authority (Fig. 5a). Each defender's LLM then sets 12 weights of the bid formula and a multiplier from 0 to 2 on each intruder's pursuit value every epoch, rather than choosing among options written in advance (Section 4.5.1). Supplementary Table 7 lists the values chosen. With the low-cost model, the consensus team was breached in 53% of episodes at high authority against 15% at low authority, over the four conditions without falsified reports and both intruder counts. That is 38 percentage points higher (95% CI: +30 to +45). The gap was larger with four intruders, where the two breach rates were 77% and 23%, 53 percentage points apart (95% CI: +44 to +62). High authority co-occurred with weaker detection. Once it knew of an intruder, the low-cost model's median answer raised the threat value and set the weights on station coverage and scouting below any value low authority allows (Fig. 5a). Intruders went unseen by every defender for 53% of their active time at high authority against 41% at low authority, 12 percentage points more (95% CI: +10 to +14).

The near-frontier model at high authority was breached no more often than the low-cost model at low authority (Supplementary Table 8). On the ideal network with four intruders, the consensus team at high authority run with GPT-6.1 Sol was breached in 7 of 30 episodes, against 23 of 30 with the low-cost model, a breach rate 53 percentage points lower (95% CI: +28 to +70). At low authority, the low-cost model was also breached in 7 of 30 episodes, so the difference from it was 0 percentage points (95% CI: −24 to +24). The near-frontier model used its authority differently. Before contact its median answer weighted station coverage and scouting above any low-authority value, and after contact it lowered the threat value and raised the engagement value (Fig. 5a). Intruders went unseen for 39% of their time, close to the 40% at low authority and below the low-cost model's 54% at high authority. Known intruders had no pursuer for 24% of their time, against 8% at low authority, with the same number of breaches.

With the near-frontier model as the commander's LLM, the team was breached less often under falsified reports, and no difference was detected on the ideal network. Over both intruder counts, the commander team using GPT-6.1 Sol was breached in 8 of 60 ideal-network episodes and 4 of 60 episodes with falsified reports, compared with 15 and 22 for the commander team using the low-cost model. Its breach rate was 12 percentage points lower on the ideal network, with a CI that includes zero (95% CI: −26 to +3), and 30 percentage points lower under falsified reports (95% CI: 18 to 42).

2.6 Airspace challenges: civilian drones, birds, and intruders disguised as birds

A typical airspace also contains non-hostile objects such as birds and civilian drones. To include a simplified version of this challenge, we added one bird and two civilian drones to each episode on the ideal network with four intruders (Fig. 6a; Section 4.8). These objects follow inbound trajectories like those of intruders but cannot breach the defended region and do not affect the score; however, any defender that collides with one is disabled. In the main study every LLM works from a state-based observation (Section 4.3), a text description of positions, velocities and the values computed from them, with no images. Here the defenders' LLMs utilize their multimodal capabilities, and reason over images and text together. They receive the same text plus one photograph of every contact they are asked about, intruders included, so each decision now also rests on image-based contact classification, telling from the photograph what a contact is. The photographs are reference pictures rather than views from a defender's sensor. Their prompts also gain a factual airspace paragraph stating that not every contact is an intruder and giving no tactical advice (Section 4.7). We evaluated five team configurations over five seeds each: the commander team with each of the two models, the consensus team at low authority with the low-cost model, and the consensus team at high authority with each of the two models. Because five seeds are too few for reliable interval estimates, we report counts rather than CIs (Fig. 6; Supplementary Tables 9 and 11). We say that an LLM *sets a contact aside* when its response removes the contact from pursuit to the greatest extent its interface permits. For a commander, this means assigning no defender to the contact; for the consensus team, it means assigning the contact a pursuit multiplier of zero at high authority or a low priority at low authority.



Supplementary Fig. 5 High authority hurt the low-cost model but not the near-frontier model. **(a)** Bid weights of the consensus team (Supplementary Table 12). Markers are median answers at high authority on the ideal network with four intruders, bars their 25th to 75th percentiles (Supplementary Table 7); pursuit weights act on no task before contact. **(b)** Breach-rate differences, first-named minus second; "better" means fewer breaches. Low-cost rows pool the four conditions without falsified reports; ideal-network rows have four intruders and one episode per seed; commander rows pool two and four intruders. Grey circles are per-seed differences. **(c)** Breach in each seed on the ideal network with four intruders. **(d)** For the episodes of **c**, intruder time unseen by every defender and known-intruder time with no defender pursuing; one marker per seed. Black dots and bars are means and 95% CIs (t distribution over 30 seeds). CI, confidence interval; low-cost LLM, Gemini 3.1 Flash-Lite; near-frontier LLM, GPT-6.1 Sol.

The near-frontier model lost no defender to a non-hostile contact in either setting, whereas each low-cost team lost 3 to 5 of its 20 defenders over the five episodes, never more than one per episode (Fig. 6e). Of these 12 losses, 11 were to civilian drones. As commander, the near-frontier model assigned a defender to a bird or civilian drone in 2 of the 361 epochs in which it saw one, and at high authority it set aside 100% of

birds and 91% of civilian drones in its answers (Fig. 6c and Supplementary Table 10). The low-cost model set birds aside more often than civilian drones in each of its three interfaces.

With these photographs the low-cost model rarely set civilian drones aside in its two decentralized interfaces. Only 12% of its answers at low authority rated them low priority and 3% of its answers at high authority gave them a zero multiplier, so most of its losses began with a wrong judgment. The interface then sets what a correct judgment can achieve. At low authority a low priority halves a contact's urgency but does not remove its pursuit task. At high authority a pursuit multiplier of zero leaves the contact's pursuit bid with only its costs, so the contact loses the auction, and a commander can leave a contact unassigned. The filter steers each defender away from every contact except its own pursuit target, so these losses are consistent with allocation errors rather than collisions. Adding birds and civilian drones did not change breaches in a consistent direction (Supplementary Table 11).

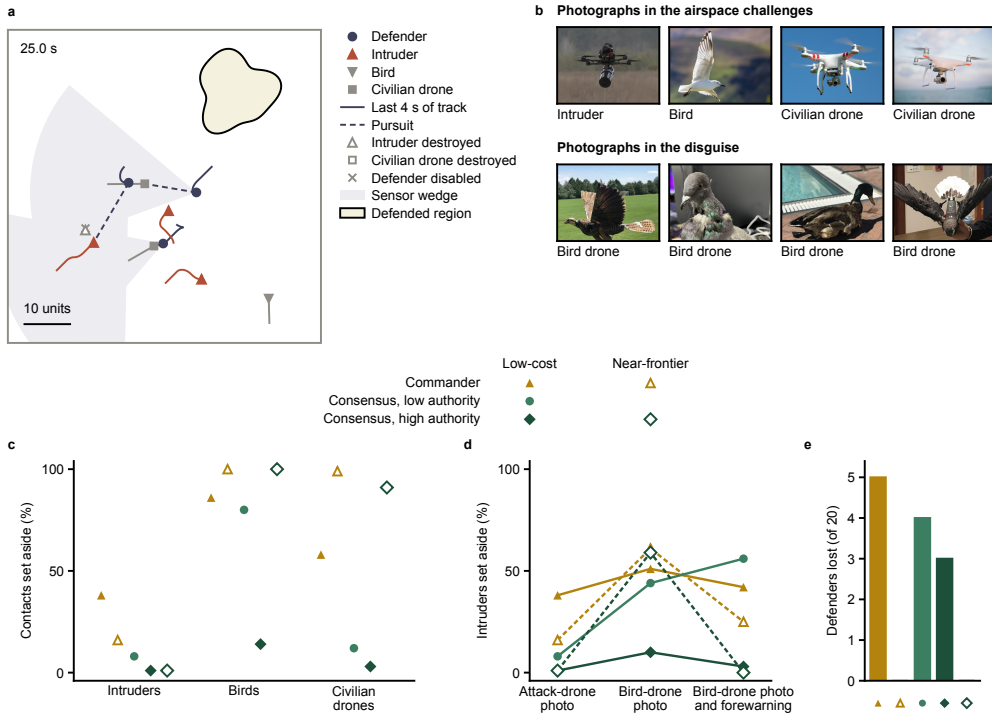
We then disguised the intruders (Fig. 6b). On the same seeds and teams, each real intruder was shown to the LLMs as a photograph of a flapping-wing drone built from a preserved bird, making it a disguised intruder [37, 38], and nothing else changed. A brief warning in the system prompt made LLM agents nearly immune to self-propagating instructions in another multi-agent setting [39], so in a second condition we added one sentence of warning to the airspace paragraph (the prompts call intruders red): "Some red intruders may be disguised as birds."

With bird-drone photographs every team set aside more of the real intruders than with attack-drone photographs, including the near-frontier model, whose teams had set aside few of them (Fig. 6d). At high authority it gave intruders a pursuit multiplier of zero in 59% of its answers, against 1% before, and as commander it left them unassigned in 61% of contact-epochs (one contact in one epoch), against 16%. Its two teams destroyed 4 and 2 of 20 intruders, against 20 and 14 before, and lost 2 and 2 defenders to civilian drones. Intruders often flew faster than any bird or civilian drone can, and the LLMs received each contact's velocity. In 4 of the five teams such fast intruders were set aside less often than slower ones, yet 10% to 56% of them still were.

The warning restored the near-frontier model but not the low-cost one. With it, the near-frontier teams set aside 0% and 25% of intruders and destroyed 14 and 14, whereas with the low-cost model the consensus team at low authority and the commander team still set aside 56% and 42%. The warning also changed how the near-frontier model at high authority treated the real birds and civilian drones: it set aside 23% of birds and 33% of civilian drones, yet lost no defender to them. With five seeds per condition we report breaches without interpreting them (Supplementary Table 11).

3 Discussion

We studied how a team of LLM agents allocates tasks in an adversarial perimeter-defense game designed as a simplified abstraction of a cyber-physical system. Of the three teams, the consensus team was breached least often, and the independent team showed no clear difference from the commander team. Of the degraded networks, only limited radio range changed a breach rate noticeably, and only the commander team's.



Supplementary Fig. 6 Airspace challenges divided the two models. (a) One frame of a consensus-team episode, low-cost LLM, at 25 s, with one bird and two civilian drones among four intruders. The rightmost defender and the civilian drone it was pursuing were lost 0.7 s later. **(b)** The photographs the LLMs received in this episode, and the four bird drones shown instead for the intruders in the disguise. **(c)** Share of the LLMs' answers that set a contact aside, by kind of contact, without the disguise; for commanders, share of contact-epochs with no defender assigned. **(d)** Intruders set aside with each kind of photograph, and with the forewarning; dashed lines join the near-frontier teams. **(e)** Defenders lost to birds and civilian drones over the five episodes without the disguise. Five seeds, ideal network, four intruders. Photographs: ArmyInform (CC BY 4.0, cropped); R. Long/Unsplash; succo/Pixabay and J. Sorenson/Pexels (CC0); bird drones, M. Hassanalian, New Mexico Tech.

High authority hurt the low-cost model and brought no measured gain with the near-frontier model. With birds and civilian drones in the airspace, the near-frontier model lost no defender to them, whereas the low-cost model's teams did. However, disguising intruders as birds (without warning), fooled LLMs regardless of their capability.

The consensus team's advantage came from removing duplicate commitments. The independent team used the same LLMs and bids as the consensus team but no auction. Outside limited range its agents duplicated commitments in 61% to 81% of checks per seed and left known intruders unpursued for longer. Earlier comparisons found related weaknesses. A decentralized LLM framework that coordinated through dialogue had the lowest success rate of four designs in multi-robot planning [10], and independent agents, which do not communicate, amplified errors more than centrally coordinated

ones across agentic benchmarks [11]. Inter-agent misalignment is also one of three categories of failure in multi-agent LLM systems[40]. Market-based allocation was proposed to combine the strengths of central and distributed coordination [33], and it did so here with LLMs at low authority. We find that decentralized LLM teams work best with an explicit protocol, outside the LLMs, that resolves conflicts.

The commander team had two weaknesses of its own. Its LLM rarely assigned scouting (16 of 39,077 tasks), and intruders stayed unseen for longer than in the consensus team. Under limited range its breach rate rose by 18 percentage points, a rise we treat as suggestive because its CI only just excluded zero. This happened even though its network stayed connected on 61% of rounds, against 14% and 19% for the consensus and independent teams. Being connected is not enough for a central allocator, which needs reports to reach it and its orders to reach every agent [33], whereas a cut-off agent of the consensus team still acts on its own bids. Only 61% of the commander's task-changing orders arrived under limited range (Fig. 4c), which fits this account. We tested neither whether the missing orders caused the extra breaches nor whether prompting the commander to scout more would close the scouting gap.

That message loss and falsified reports changed no breach rate noticeably does not show that the teams were unaffected. The CIs spanned about 30 percentage points, and under burst loss the commander and independent teams leaned toward more breaches. Our attack was also narrower than the faulty agents against which a hierarchical structure proved most resilient in text-based collaboration [41]. It shifted one intruder's position in one defender's reports, a later sighting could correct it, and no LLM's instructions were touched. A manipulated message in a commander's prompt could misdirect every order, and such manipulation can spread between LLM agents [39], so attacks that reach the LLMs are the natural next test.

Low authority guarded the low-cost model against its own choices. At high authority, setting how the auction scores tasks, it weakened detection and was breached far more often. The near-frontier model at high authority was breached as often as the low-cost model at low authority, and as commander it was breached less often under falsified reports. Neither model's own weights improved on low authority. Filtering an LLM's proposed actions is one way to keep LLM-driven robots within constraints [42], and coordination structure and model capability interact in agent systems [11]. Constraining what an LLM may set was also noted to make an LLM-driven robot more stable, at some cost in flexibility [32]. Our results add that how much an LLM may decide should follow its capability.

Leaving birds and civilian drones alone required both a model that could tell them from intruders and an interface that let it act on that judgment. The model mattered first. As commander or at high authority, both models could drop a contact. The near-frontier model set aside 99% and 91% of civilian drones and lost no defender. The low-cost model set aside only 58% and 3%, and its three teams each lost 3 to 5 of their 20 defenders over five episodes. The interface mattered second. At low authority the consensus team can only give a contact a low priority, which leaves the chase among a defender's options, so even a correct judgment can send a defender after a bird. LLM planners that share airspace with non-hostile contacts therefore need an explicit option to ignore one. As in an earlier study with images of the same kinds of contact,

civilian drones were harder to set apart than birds [23]. The near-frontier model's readiness to leave birds alone had a cost. Once the intruders were shown as birds, its two teams set more of them aside than any low-cost team did. With a one-sentence warning that some intruders may be disguised as birds, the near-frontier model chased them again, whereas with the low-cost model the consensus team at low authority and the commander team still often left them unchased. A brief warning in the system prompt largely protected LLM agents against self-propagating instructions in another study [39]. Against visual deception, how much such a warning helps may depend on the model. Our disguise, drawn from work on camouflaged drones [43], changed only the photograph, not how the intruder flew, and its close-ups show mechanical details that a distant sensor would not. With five seeds, one bird and two civilian drones per episode, and reference photographs rather than sensor views, these results mark a problem rather than measure it.

Our testbed is intentionally a simple abstraction of a cyber-physical system. Agents move as point masses in two dimensions and sense their opponents without noise. They exchange messages in rounds rather than over radio, see reference photographs rather than camera views, and wait for every LLM answer, so the seconds a real call takes cost them no time. We kept these parts simple to focus on the LLMs. Differences between teams then come from how their LLMs coordinate rather than from errors in control or perception, and each factor can be varied while the rest stay fixed. Higher-fidelity dynamics, sensing, radios and imagery, real call latency and, finally, hardware are left to future work. Many four-intruder episodes ended at the 40 s limit with an intruder still closing, so longer episodes could change the breach rates and the gaps between teams. Stronger intruders, larger teams, more models, longer episodes and the full design for the near-frontier model are left to future work.

The framework is therefore best understood as a benchmark for LLM-driven coordination of cyber-physical teams rather than as a deployable counter-drone controller. Its findings are guidance for designing such teams, to be confirmed at higher fidelity.

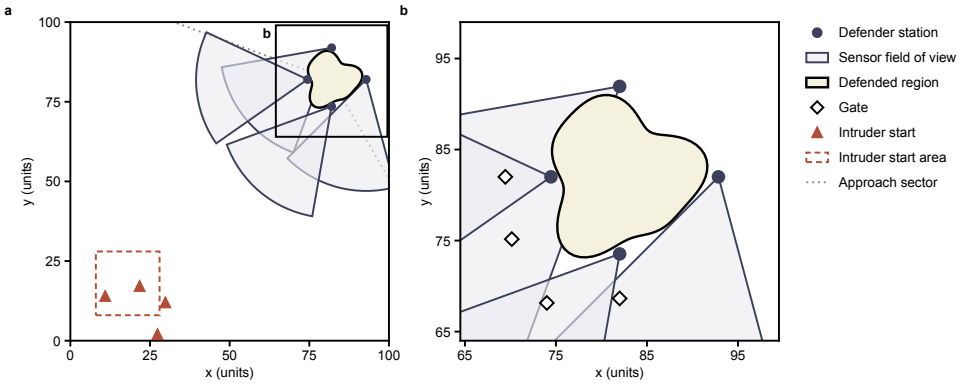
4 Methods

This section describes the simulation environment and the experimental procedure. We first present the agent dynamics, the motion controllers, and the collision-avoidance constraints, followed by the belief model that each agent maintains about the opposing team. We then describe the communication network, including the degraded network conditions and the falsified-report attack. Next, we detail the three defender architectures: the consensus auction, with its task generation, bid formulation, and LLM interfaces; the independent agents; and the commander. This is followed by the intruder team and the prompts and language models used by all agents. Finally, we describe the experimental design, the outcome measures, and the statistical analysis.

4.1 Game and arena

The arena is a 100 by 100 square in abstract units. The defended region is a concave star-shaped polygon centered at $(82, 82)$ with a radius between 6 and 10 units, and four gates lie on its south-west side. Four defenders start near four stations placed on

a polygon around the region. Two or four intruders start in a 20 by 20 box centered at $(18, 18)$, and each intruder's start is then rotated about the region's center by its own angle of up to 10° in either direction, so the intruders start about 90 units from the region (Fig. 7). An episode lasts at most 400 steps of $\Delta t = 0.1\,\mathrm{s}$. A defender within 0.4 units of an intruder destroys it and is disabled for the rest of the episode. An intruder inside the region or on its boundary is a breach, and the breach ends the episode. An intruder that leaves the arena is removed. An episode also ends when every intruder has been destroyed or has left the arena.



Supplementary Fig. 7 The arena, to scale. (a) The whole arena at the start of one episode with four intruders. Each intruder starts in the start area and is then rotated about the region's centre by up to 10°; birds and civilian drones, when present, enter within the approach sector. Each defender's sensor sees 35 units within a 60° wedge. **(b)** The defended region enlarged, with the four defender stations and the four gates through which intruders attack. Distances are in the abstract units of the arena.

4.2 Dynamics and hierarchical control

Every agent has the state $x = [p^\top, v^\top, \theta]^\top$, with planar position p, velocity v and sensor heading θ, and the input $u = [a^\top, \omega]^\top$, with translational acceleration a and yaw rate ω. The state evolves by semi-implicit Euler integration,

$$p_{k+1} = p_k + \Delta t\, v_k + \Delta t^2 a_k, \qquad v_{k+1} = v_k + \Delta t\, a_k, \qquad \theta_{k+1} = \theta_k + \Delta t\, \omega_k, \tag{1}$$

with per-axis limits $|a_x|, |a_y| \le 2$, $|v_x|, |v_y| \le 2$ and $|\omega| \le 2\pi\,\mathrm{rad\,s^{-1}}$. Each task (holding a station, flying to and watching a coverage cell, pursuing an intruder, flying to a gate, or evading a defender) is executed by a model predictive controller (MPC) [23] with a horizon of 60 steps (6 s). The MPC problems are formulated in CasADi [44] and solved as quadratic programs with OSQP [45], except evasion, which is solved as a nonlinear program with IPOPT [46].

A pursuer aims at a constant-velocity prediction of its latest belief about its target. The sensor heading follows fixed rules: a pursuer points at its target, a defender that

sees an intruder tracks it, a defender on a station looks along that station's configured heading, and a scout looks toward its cell.

To avoid collisions, each agent minimally modifies its MPC acceleration through a quadratic program with one constraint per nearby agent, based on the following barrier. For two agents at relative position r and a minimum separation $D = 3$ units, we define $h = \|r\|^2 - D^2$ and impose a linear sufficient condition for $h_{k+1} \geq (1-\gamma)\, h_k$, with $\gamma = 0.05$. This barrier is a simplification suited to our double-integrator model; for more complex dynamics, control barrier functions (CBFs) can serve as the safety filter, since under suitable assumptions they are necessary and sufficient for rendering the safe set forward invariant [47–49].

Two defenders each enforce half of their shared constraint, disabled defenders are treated as immobile obstacles, intruders other than a defender's pursuit target are predicted at constant velocity, and a defender and the intruder it pursues are exempt from each other's constraints. Intruders avoid only each other, because imposing the constraint between intruders and defenders prevented every defender from making contact during development; intruders still evade approaching defenders, but through their planning layer rather than the barrier. If the quadratic program is infeasible, a relaxed problem with slack is solved, and the agent brakes only if that also fails or the velocity limits leave no admissible acceleration. Under the input limits, the barrier condition is not always satisfiable. One-step separation is therefore guaranteed only when the constraints are feasible, and against intruders only under the constant-velocity prediction. The barrier uses the true positions and velocities of all agents as a stand-in for onboard proximity sensing.

4.3 Sensing and beliefs

Each agent i keeps its own belief about the opposing team. At step k, this belief is a set of contact records $\mathcal{B}_i(k) = \{b_{i,c}\}_{c \in \mathcal{C}_i(k)}$, where $\mathcal{C}_i(k)$ is the set of opposing agents that i has seen or been told about. Each record $b_{i,c} = (\hat{x}_c,\ t_c,\ o_c)$ holds the state $\hat{x}_c = [\hat{p}_c^\top, \hat{v}_c^\top]^\top$ of contact c as observed, the step t_c at which a sensor observed it, and the agent o_c that made the observation. Sensing is noise-free: an agent observes the true state of every opposing agent within its sensor cone (half-angle 30°, range 35 units) and nothing outside it. Agents use a record through its constant-velocity prediction $\hat{p}_c(k) = \hat{p}_c + \hat{v}_c\,(k - t_c)\,\Delta t$, so the age $k - t_c$ measures how out of date it is. When a record arrives from a teammate, the receiver keeps whichever record for that contact has the later observation step t_c, and keeps its current record on a tie. Relaying does not change t_c, so a relayed record is as old as the original sighting. Records are never discarded for being old, and not seeing a contact does not remove it. A contact is removed only when it is resolved: when an intruder is intercepted or leaves the arena, or a defender is disabled, every agent on the opposing team deletes that agent's record immediately and permanently. Defenders also keep a coverage belief $s_i(k) \in \{-1, 0, 1, \dots\}^M$, the last step at which each of the M coverage cells was swept (-1 if never), which merges by taking the elementwise maximum. Teammates' states are not part of the belief; each agent knows them exactly. Intruders pool their sightings over an ideal channel, whereas defenders share theirs over the simulated network, an asymmetry that is part of the design.

4.4 Communication and network conditions

Defenders exchange beliefs, auction states, and commander assignments in synchronous communication rounds at intervals of 0.1 s. Messages are double-buffered, so information traverses exactly one link per round, and each defender rebroadcasts its merged beliefs in every round. Upon receiving a record, a defender retains whichever record has the later observation time, keeping its existing record in the event of a tie. We consider five network conditions. The ideal network is complete and lossless but retains a latency of one round per link. Under independent loss, the exchange on each link is erased in each round with probability 0.3. Burst loss is modeled as a Gilbert–Elliott channel [50–52] with transition probabilities of 0.15 from the good to the bad state and 0.30 from the bad to the good state, and loss probabilities of 0 and 0.9 in the good and bad states, respectively; the stationary mean loss is therefore also 0.3. Because every link is initialized in the good state, the realized loss varies across episodes. Loss draws are a deterministic function of the seed, the link, and the round, so matched episodes experience identical losses on identical links. Under limited range, only links shorter than 15 units are retained, and messages reach distant defenders solely through rebroadcast. We quantify realized connectivity as the fraction of rounds in which the active defenders form a single connected component over the links within range. Under the loss conditions this fraction is 1 by construction, since loss erases messages on links that exist rather than removing the links themselves.

In the falsified-report condition, a single defender other than defender 0 is compromised. It is drawn from a dedicated random stream so that matched episodes share the same compromised defender. From its first direct observation of an intruder until the end of the episode, the compromised defender displaces that intruder's position by 10 units toward the south-west in its outgoing first-hand reports; if it observes several intruders, it targets the one with the lowest identifier. The intruder's identity, velocity, and observation time are left unaltered, as are the compromised defender's own stored beliefs. Recipients apply the standard merge rule, so the order of arrival determines whether a falsified or an honest copy of the same observation is retained. The attack thus models a compromised reporting channel rather than a compromised decision-maker.

4.5 Defender teams

4.5.1 Tasks and bids

The consensus and independent teams allocate tasks through bids, which are scores that each defender assigns to its candidate tasks, and their LLMs influence the allocation by modifying these bids. The bid described below was designed for this game; it is one reasonable choice among many rather than a fundamental or optimal rule. In market-based allocation, the designer must specify each agent's utility function, and no formal method for designing such functions yet exists [33]. We therefore constructed the bid from terms that each capture a single consideration a defender should weigh, and we justify each term alongside its definition. Default weights were set from the arena's geometry and from constructed situations on development seeds, never from the outcomes of study episodes. In these situations, the default weights led the auction

to assign a pursuer to every known intruder, to keep the stations with the greatest coverage occupied and to send defenders at redundant stations to scout.

Each defender builds its own list of candidate tasks from its beliefs: pursuing a known intruder j, holding one of the four stations s, or scouting one of the 16 cells g of a 4×4 coverage grid. It assigns every task q on this list a bid,

$$b_q = V_q - C_q + \beta\, \mathbf{1}[q \in Q], \tag{2}$$

where V_q is the task's value, C_q is its cost, $\beta = 0.2$ is a bonus, Q is the set of up to three tasks that the defender's LLM has marked as preferred, and $\mathbf{1}[\cdot]$ is the indicator function.

Subtracting the cost from the value puts unlike tasks, such as a chase and a station, on one scale, so that a defender can rank them and the auction can compare defenders. The bonus lets the LLM decide between tasks whose bids lie within 0.2 of each other but cannot overturn larger differences. The value and the cost add up terms, each a quantity that the defender computes from its beliefs times a weight w whose subscript names the term. The weights are the numbers that the LLM can change, as the two interfaces below describe. Supplementary Table 12 lists every symbol of these Methods with its value or default and, for each weight, the values that the LLM can set.

The value depends on the kind of task,

$$V_q = \begin{cases} \rho_j \left(w_T + w_E\, \pi_j U_j O_j\right) & \text{pursue intruder } j, \\ w_S + w_C\, m_s & \text{hold station } s, \\ w_I\, I_g & \text{scout cell } g. \end{cases} \tag{3}$$

A pursuit is worth a threat value w_T, the same for every intruder known to the defender, plus an engagement value w_E times three factors. The threat value treats every known intruder as a threat, even one that is far away, moving away or hard to catch, because its LLM can turn it back to attack. It raises all of a defender's pursuits by the same amount, so it does not change which intruder the defender prefers or which defender is best placed for an intruder. It only lifts pursuits above stations and scouting. The engagement term then ranks intruders against each other. Its first factor is a priority π_j, which the LLM sets at low authority and which is otherwise 1. Next is the intruder's urgency,

$$U_j = \max\!\left(\frac{1}{1+\tau_j/T},\ \frac{1}{1+\delta_j/L}\right), \tag{4}$$

where τ_j is the forecast time for the intruder to reach the region and δ_j is its distance from the region, and the scales are $T = 60\,\mathrm{s}$ and $L = 25$ units, a quarter of the arena. The forecast uses the position and velocity in the defender's belief and is infinite if the intruder is not approaching, and the distance is measured from the intruder's predicted position. An intruder is therefore urgent when it is forecast to reach the region soon or is already close to it. We take the larger of the two terms because the forecast alone misleads. An intruder that dodges a defender near the region stops heading for it, so

its forecast becomes infinite although it is still close. Each term falls smoothly from 1 and halves at its scale, so no threshold makes a bid jump. The third factor is the opportunity O_j to catch the intruder, because a defender should prefer intruders that it can reach before they breach. It is 1 if the defender can intercept intruder j before its forecast breach and the end of the episode, and falls to 0.25 for an interception 10 s or more too late, or none. It stays above zero because the forecast assumes that the intruder keeps its velocity, which its LLM may change. It is averaged over 49 intruder velocities within 0.3 units s^{-1} per axis of the one in the defender's belief. The believed velocity is only an estimate, and without the average a small change in it could open or close an interception and make the bid jump. This opportunity is optimistic, because it assumes that the defender can fly anywhere its limits allow and it ignores the barrier filter and the intruder's evasion. A pursuit multiplier ρ_j scales the whole pursuit value, threat value included. The LLM sets it only at high authority, where it can make any intruder's pursuit more attractive, less attractive or worth nothing, and it is otherwise 1.

Defenders expect intruders from an approach sector, the bearings between 160° and 300° as seen from the region's center, a known approach corridor that is the same for every team. A station is worth a small fixed station value w_S plus w_C times its coverage m_s. This coverage is the area of the approach sector that the sensor of a defender at station s would watch and no teammate currently watches, divided by the area of one sensor wedge. Counting only ground that no teammate watches pays a station for what it adds, so two defenders watching the same ground are not both rewarded for it. The small fixed value keeps holding a station worth something even where it adds little coverage. A cell is worth w_I times its information I_g, the unwatched area of the approach sector that the defender's sensor would first sweep on a straight route to cell g, in the units of m_s. Scouting is worth what it may find, so each part of that area is weighted by how long it has gone unwatched, capped at 20 s. The area is also discounted with distance along the route, because ground swept sooner finds an intruder sooner, and without the discount the longest route, which sweeps the most, would always win.

The cost also varies with the task,

$$C_q = \begin{cases} w_D\, d_q/L + w_A\, \alpha_j + w_X/(1+n) + K_q & \text{pursue intruder } j, \\ w_D\, d_q/L + K_q & \text{hold station } s, \\ w_R\, d_q/L + K_q & \text{scout cell } g. \end{cases} \tag{5}$$

Every task costs travel in proportion to its distance d_q from the defender, measured to an intruder's predicted position, a station or a cell center, with weight w_D for pursuits and stations and w_R for scouting. Travel makes the nearest suitable defender bid highest, so the auction tends to give each task to the defender best placed for it. Scouting travel weighs less, because on a scouting route the travel is itself the work. A pursuit costs more when the defender's belief about the intruder is old, by w_A times the belief age α_j. That age is the age in seconds of the belief, times the intruders' per-axis speed limit of 2 units s^{-1}, divided by L. It thus measures how far the intruder may have moved since it was last seen. A pursuit also risks the defender itself, because

touching the intruder disables it, so it carries an attrition cost $w_X/(1+n)$, where n is the number of active teammates. This cost grows as fewer teammates remain, because losing one of four defenders leaves the region better guarded than losing the last.

By default the value weights are $w_T = 1.2$, $w_E = 1$, $w_S = 0.05$, $w_C = 0.5$ and $w_I = 0.45$. The default cost weights are $w_D = 0.15$, $w_R = 0.05$, $w_A = 0.1$, $w_X = 0.5$, $w_L = 0.25$, $w_W = 0.05$ and $w_P = 0.15$. The threat value is the smallest multiple of 0.05 at which the auction assigned a pursuer to every known intruder in constructed situations with one or two intruders, including intruders far away and moving away. With these defaults the threat value alone (1.2) is worth more than any station (at most 0.55). A station can outbid the pursuit of a known intruder only when the pursuit costs much more, for example when the intruder is far away. Several defenders that know of the same intruder can therefore all bid highest for it. The auction resolves this by giving each task to one defender only, so the others take their next-best tasks.

Each task also carries a change cost,

$$K_q = w_L\, \ell_q\, m_0 + w_W\, \sigma_q + w_P\, c_q, \tag{6}$$

which penalizes three kinds of change: leaving a useful station (w_L), switching tasks (w_W) and abandoning a pursuit (w_P). Because bids are recomputed every second (10 simulation steps) from updated beliefs, small fluctuations in those beliefs would otherwise cause defenders to oscillate between tasks without completing any of them. The leaving indicator ℓ_q equals 1 when task q is not the station the defender currently holds (or its own station if it has no task). The leaving cost is scaled by the coverage m_0 of that station, which is 0 while the defender is pursuing or scouting. A defender at a station that contributes little coverage therefore leaves more readily and, other things being equal, outbids a defender at a more valuable station for the same task. The switching indicator σ_q equals 1 when the defender has a task and task q differs from it. The drop indicator c_q equals 1 when task q would abandon the pursuit of an intruder the defender still knows about, so that a pursuit already under way is worth completing. Its weight w_P exceeds the largest change in a pursuit bid caused by velocity-belief errors of $0.1\,\text{units}\,\text{s}^{-1}$ in constructed test situations, so such errors alone cannot terminate a pursuit. The drop indicator is reduced toward zero when the pursued intruder is moving away and task q instead pursues another intruder that is forecast to breach within the episode and that the defender can intercept in time. This allows a defender to abandon a retreating intruder in favor of one that is about to breach.

4.5.2 Auction and independent selection

In the consensus team the bids enter a consensus-based auction [34] run over the network. The auction restarts every epoch (every 10 communication rounds) and breaks ties in favor of the lower agent identifier. A defender adopts the auction's result once its own table has been unchanged for three rounds and keeps its previous commitment until then, and commitments persist across epochs. The independent team runs the same code with the auction messages switched off, and its prompt describes independent selection instead of the auction. Each defender then commits to its own highest

bid, so two defenders may pursue the same intruder. The independent team is therefore a matched ablation of conflict resolution rather than a representative independent policy.

4.5.3 Commander

The LLM of defender 0 assigns one task to every active defender each epoch. If defender 0 is disabled, the lowest-index active defender takes over and plans from its own beliefs. The commander chooses from a catalog of candidate tasks that the same rules build from its own beliefs. It sees each intruder's forecast status (approaching, moving away, crossing or stationary) and time to breach, each defender's optimistic interception time and its last issued assignment. A duplicate or invalid entry sends the affected defender to its own station. The commander's own order applies at once. Every other order is broadcast, relayed by the other defenders and applied by its addressee if it is newer than the last order that defender accepted, and an addressee with no belief about the ordered intruder holds its station until it has one.

4.5.4 Low-authority LLM interface

The consensus and independent teams give the LLM low authority. Every number in a bid then comes from a formula fixed before any run, each answer is a choice among named options that can be read in the logs, and no answer can stop a known intruder from being pursued. Once per epoch, each defender's LLM sees only that defender's beliefs and its task list, with every bid term computed. It can change the bids in three ways. First, it picks one of four weight profiles. The balanced profile uses the default weights, and the aggressive profile lowers w_X to 0.2 and raises T to 90 s. The conservative profile raises w_C to 0.8, w_L to 0.4 and w_X to 0.9, and the scouting profile raises w_I to 0.75. Second, it gives each known intruder a priority π_j of low, medium or high (0.5, 1 or 2). Third, it names up to three preferred tasks. It also gives one sentence of reasoning. The multiplier ρ_j stays at 1, and no number written by the LLM enters a bid. No profile changes w_T, and the priority multiplies only the urgency and opportunity term, so a known intruder's pursuit is always worth at least 1.2. A low priority makes a pursuit less attractive but cannot remove it. Invalid parts of an answer are dropped and logged, and a failed call leaves the defender's bids at their defaults.

4.5.5 High-authority LLM interface

At high authority, the consensus team's LLM can change much more. Every epoch it sets each of the 12 weights within an allowed range (Supplementary Tables 7 and 12), and the five value weights may sum to at most 3.2, their default total. Instead of a priority, it gives each known intruder a multiplier ρ_j from 0 to 2, and the multipliers may sum to at most the number of known intruders. It can also name up to three preferred tasks, while π_j stays at 1 and T at 60 s. An invalid or out-of-range value falls back to its default, and weights or multipliers over their budget are scaled down in proportion, as the prompt states. Each fallback and each scaling is logged. Because ρ_j multiplies the whole pursuit value, threat value included, a multiplier of 0 leaves

the pursuit with only its costs. High authority does not contain low authority, because the conservative and scouting profiles exceed the value budget, T stays at 60 s and no multiplier reproduces a priority, which scales only part of the pursuit value. The bids then enter the same auction.

4.6 Intruders

The intruders are one team of independently instantiated LLM agents, the same for every defender team. Each intruder's LLM chooses to attack, by flying to one of the four gates and then to the region's center, or to evade a named defender that its beliefs place nearby. All intruders' calls in an epoch use observations frozen before the first call, and an invalid answer keeps a still-valid decision or falls back to attacking. The intruders chose to attack in 73% to 89% of their answers across teams and intruder counts. Every intruder's LLM is the low-cost model, Gemini 3.1 Flash-Lite, in all experiments, including those where the defenders use the near-frontier model, so that every defender team faces the same opponent.

4.7 Prompts and language models

Each team receives a factual brief that states its own objective, the episode length, what happens when agents touch or an intruder leaves the arena, and the end conditions. The defender brief is the same for every team. The prompts call the defenders blue and the intruders red. No prompt gives tactical advice, although the task catalog, the bid formula and the controllers encode domain choices. Every LLM observation in the main study is state-based: a text rendering of the agent's beliefs and of the options and values computed from them, with no images. Unless stated otherwise, every LLM agent is Gemini 3.1 Flash-Lite [53], the low-cost model, called through OpenRouter at the Google AI Studio Flex tier with medium reasoning effort, temperature 1.0, an allowance of 8,000 output tokens and a strict JSON schema whose first field is the reasoning. The near-frontier model is GPT-6.1 Sol [54], which its developer describes as close in performance to its most capable model at a lower cost, called at the standard tier with medium reasoning effort, no temperature parameter and an allowance of 25,000 output tokens. Calls are synchronous: the simulation waits for every answer of an epoch, so call latency does not affect the game (calls, latency and cost in Supplementary Table 13). Each response is stored under a hash of its full request, and each run records an execution identity that covers the code, the interfaces, the model profile and the settings. All interfaces and prompts are versioned and recorded with every run, and Supplementary Note 1 reproduces them.

4.8 Airspace challenges

Our main study gives the defenders a clean sky: every contact they detect is an intruder, and the only question is who pursues it. A typical airspace is rarely that neat. It also holds birds and civilian drones, and telling drones apart from birds in video is hard enough to be the subject of a dedicated benchmark challenge [36]. In our game a wrong call is expensive, because a defender that touches any contact is disabled, so chasing a gull or a hobbyist's camera drone costs the team a defender

just as an interception does. Motion alone does not settle the question either, because our birds and civilian drones fly inbound like intruders, at up to the intruders' speed limit. Vision can help in this regard, and the LLMs we study support that capability. Both LLMs used in this study are multimodal: besides the state-based text they read in the main study, they accept images, so they can act as vision-language models that weigh a picture together with the state. We therefore built two challenges on top of the game. In the first, birds and civilian drones fly among the intruders, and each defender's LLM sees a photograph of every contact before it decides. In the second, we turn that ability against the defenders, and the intruders arrive disguised as birds. The disguise is not far-fetched, and it builds on existing work. Camouflage has been applied to fixed-wing, quadcopter and ornithopter drones [43], and wildlife-monitoring drones have been built from taxidermy birds that fly by flapping their wings [37, 38] and from a taxidermy duck that swims [55]. Both challenges use the ideal network with four intruders, on seeds 4001 to 4005, so that whatever changes comes from the airspace and not from the network.

4.8.1 Defender observation space

Each defender's LLM, and the commander's, receives a multimodal observation. It is the same state-based text as in the main study, followed by one photograph of every contact the LLM is asked about, each labeled "image of contact_id N". A defender's LLM is asked about every contact it holds a belief about, and the commander about the contacts in its catalog. Intruders are photographed too, so a defender has to recognize threats as well as rule out harmless contacts. The intruders' own LLMs keep their state-based observations, as every LLM does in the main study. We chose the photographs so that recognition is honest rather than easy. They come from a set of nine, three per kind of contact, assigned the same way in every team. The intruders are two first-person-view attack drones and a commercial quadcopter carrying a grenade, so a consumer airframe alone does not mark a contact as civilian. The civilian drones are three consumer camera quadcopters, and the birds are a gull, a raven and a black kite. Each photograph is a public-domain or openly licensed picture of one object in flight, cropped to 4:3 at 640 by 480 pixels, with any readable text removed, so that no brand name or marking gives the answer away. Sources, licenses and the script that builds the photographs from their originals are in the code repository. Every defender and commander prompt also gains one factual paragraph, the airspace paragraph: not every contact is an intruder, birds and civilian drones share the airspace and cannot breach, touching any contact destroys it and disables the defender, and a contact's identifier does not tell what it is. The paragraph gives no tactical advice, and no field was added in which the model must say what it thinks a contact is, so a model's judgment shows only in what it chooses to pursue.

4.8.2 Birds and civilian drones among the intruders

Each episode adds one bird and two civilian drones to the matching ideal-network episode of the main study. The intruders and defenders start exactly as in that episode, so every difference traces back to the newcomers. To be hard to rule out by motion, the birds and civilian drones behave like intruders on approach. They spawn 55 to

80 units from the region's center inside the approach sector and fly inbound through waypoints, the first 20 to 35 units from the center and later ones 20 to 70 units out, with no leg passing within 16 units of the center. They cruise at 0.6 to 1.0 times the intruders' speed limit, sense nothing, cannot breach and earn nothing, and a defender that touches one is disabled and the contact destroyed. Five LLM teams ran on the five seeds (25 episodes).

4.8.3 Intruders disguised as birds

Distinguishing birds from drones might not be that hard of a challenge for a modern LLM, but making the distinction between a real bird and a disguised one might very well be. To test this, the five LLM teams ran again on the same seeds with the same starts, birds, civilian drones and photographs of them, but each real intruder was now shown as one of four photographs of taxidermy bird drones (Fig. 6b). The photographs show taxidermy bird drones from Dr. Hassanalian's group at New Mexico Tech, whose flapping-wing designs are cited above [37, 38]. Intruder k on the s-th seed showed photograph $(s+k) \bmod 4$, so every episode showed each of the four drones once. Only the picture changed, and the disguised intruders flew exactly as before. A cheap countermeasure has worked against manipulation between LLM agents before. Papadopoulos et al. [39] showed that a one-paragraph warning in an agent's system prompt, written in a single attempt without tuning, made agents almost completely immune to adversarial ideas that spread through a multi-agent LLM system by inducing each agent that adopts them to pass them on. That manipulation traveled through text, whereas our deception arrives as a picture, so we probe whether the same defense carries over. Each team ran without and with one sentence added to the airspace paragraph after "A contact's id does not tell what it is.": "Some red intruders may be disguised as birds." Nothing else in any prompt differed, and the prompt version recorded with every call names the condition. The two conditions ran with the same model snapshots and providers (50 episodes).

4.9 Experimental design

The main study crossed three defender teams, two intruder counts and five network conditions on seeds 4001 to 4030, for 900 episodes. Episodes that share a seed and an intruder count share their start positions, rotations, link-loss draws and compromised defender, because starts are drawn first and the other three use their own random streams, the method of common random numbers [56]. Matched episodes diverge once the LLMs, sampled at temperature 1.0, answer differently. The consensus team at high authority ran every network condition and intruder count with the low-cost model, like a main-study team (300 episodes). With the near-frontier model it ran the ideal network with four intruders (30 episodes). The near-frontier commander ran the ideal network and falsified reports with both intruder counts (120 episodes), with everything but the commander's model taken from the commander team. Every prespecified episode ended in a valid record under its run's execution identity. In the main study, the 900 episodes made 160,788 calls with 0 transport failures, 43 provider retries all recovered, and in 7 of 10,657 commander epochs one defender fell back to its station.

4.10 Metrics

The primary metric is the occurrence of a breach within the 40 s episode. Duplicate commitments are evaluated at every epoch boundary after the first, and a duplicate is recorded at each boundary at which two or more active defenders are committed to the same task. To quantify pursuit coverage, at each time step we considered every active intruder contained in the belief set of at least one active defender and recorded whether any active defender was executing a pursuit of that intruder. The uncovered fraction is the proportion of these intruder-steps with no such pursuit, pooled within each seed. Intruders absent from every defender's belief set are excluded from this fraction and reported separately as the unobserved fraction of active-intruder time. Pre-contact scouting is defined as the fraction of defender-steps preceding the team's first intruder observation during which a defender was executing a scouting task. To quantify the influence of collision avoidance, a defender's control step is classified as modified when the barrier constraints altered its acceleration by more than 0.01 after the input limits had been applied. At the time limit, each intruder that remained active was classified on the basis of its final 10 s as being in a stand-off if it was evading on at least half of the steps and had a defender within 6 units on at least half of the steps; otherwise as closing if its distance to the defended region decreased by more than 1 unit; and otherwise as other.

4.11 Statistics

We compare two conditions, for example the commander team and the consensus team, on the same random seeds. A seed fixes the starting positions of all agents, so two episodes with the same seed and intruder count start from exactly the same situation. We call such two episodes a matched pair. For each pair we record whether each episode had a breach (1) or not (0). The pair's difference is $+1$ if only the first episode had a breach, -1 if only the second did, and 0 if both or neither did. Episodes on the same seed are not independent of one another, so we do not count each episode as a separate data point. Instead we group the pairs by seed and average their differences, which gives one number per seed. For example, if on one seed the first team was breached in 3 of 8 episodes and the second team in 1 of 8, that seed's number is $(3 - 1)/8 = 0.25$. We report the mean of the 30 seed numbers, in percentage points, as the difference in breach rate between the two conditions. Its 95% confidence interval (CI) is the range of true differences that the 30 seeds leave plausible. We compute it in the standard way as the mean plus or minus a multiplier times the standard error, where the multiplier comes from the t distribution with 29 degrees of freedom because the spread is itself estimated from only 30 seeds. This is an approximation, since the seed numbers are bounded between -1 and $+1$. Computing the interval over seeds rather than over episodes keeps it from being too narrow. A few comparisons have only one pair per seed, for example the near-frontier model at high authority on the ideal network with four intruders. There is then nothing to average, so each seed's number is simply that pair's difference, -1, 0 or $+1$. For this case we use Newcombe's hybrid score interval, method 10 [57], instead of the t interval.

Before any outcome was run, we fixed four primary comparisons. Three compare teams: commander against consensus, commander against independent, and consensus against independent, each taken over the ideal network, both loss conditions and limited range with both intruder counts (4 conditions $\times$ 2 intruder counts = 8 pairs per seed). The fourth is the effect of falsified reports: each team with falsified reports against the same team on the ideal network, over all three teams and both intruder counts (3 teams $\times$ 2 intruder counts = 6 pairs per seed). In every comparison the difference is the first-named condition minus the second-named. Comparisons by intruder count, by team and by condition are secondary breakdowns of these, also fixed in advance, and are reported as estimates with CIs. Some comparisons were added during the study, after outcomes had been seen, and are therefore exploratory: the difference in unpursued time between the consensus and independent teams, three comparisons of the effect of falsified reports (four rows), and high against low authority in the consensus team together with its falsified-report difference (two rows). We report all results, including the primary comparisons, as estimates with CIs and do not label hypotheses as accepted or rejected. The runs with birds, civilian drones and disguised intruders (five seeds each) are descriptive, and their counts are reported without CIs.

Acknowledgements. This work is supported by the Gleason Endowment at RIT.

References

[1] Guo, T., Chen, X., Wang, Y., Chang, R., Pei, S., Chawla, N.V., Wiest, O., Zhang, X.: Large language model based multi-agents: a survey of progress and challenges. In: Proc. 33rd International Joint Conference on Artificial Intelligence (IJCAI), pp. 8048–8057 (2024). https://doi.org/10.24963/ijcai.2024/890

[2] Shen, Y., Song, K., Tan, X., Li, D., Lu, W., Zhuang, Y.: Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face. Advances in Neural Information Processing Systems **36**, 38154–38180 (2023)

[3] Zhan, Z., Song, T., Dong, L., Huang, S., Lian, J., Xia, Y., Wei, F.: Agensh: Scaling organizational intelligence to 1,024 agents. arXiv preprint arXiv:2609.26781 (2026)

[4] Hong, S., Zhuge, M., Chen, J., Zheng, X., Cheng, Y., Wang, J., Zhang, C., Yau, S., Lin, Z., Zhou, L., *et al.*: Metagpt: Meta programming for a multi-agent collaborative framework. In: International Conference on Learning Representations, vol. 2024, pp. 23247–23275 (2024)

[5] Kannan, S.S., Venkatesh, V.L.N., Min, B.-C.: SMART-LLM: smart multi-agent robot task planning using large language models. In: Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 12140–12147 (2024)

[6] Mandi, Z., Jain, S., Song, S.: RoCo: dialectic multi-robot collaboration with large language models. In: Proc. IEEE International Conference on Robotics and

Automation (ICRA), pp. 286–299 (2024). https://doi.org/10.1109/ICRA57147.2024.10610855

[7] Gerkey, B.P., Matarić, M.J.: A formal analysis and taxonomy of task allocation in multi-robot systems. International Journal of Robotics Research **23**(9), 939–954 (2004) https://doi.org/10.1177/0278364904045564

[8] Yang, Y., Chai, H., Shao, S., Song, Y., Qi, S., Rui, R., Zhang, W.: AgentNet: decentralized evolutionary coordination for LLM-based multi-agent systems. In: Advances in Neural Information Processing Systems (NeurIPS) (2025)

[9] Tran, K.-T., Dao, D., Nguyen, M.-D., Pham, Q.-V., O'Sullivan, B., Nguyen, H.D.: Multi-agent collaboration mechanisms: a survey of LLMs. Preprint at https://arxiv.org/abs/2501.06322 (2025)

[10] Chen, Y., Arkin, J., Zhang, Y., Roy, N., Fan, C.: Scalable multi-robot collaboration with large language models: centralized or decentralized systems? In: Proc. IEEE International Conference on Robotics and Automation (ICRA) (2024)

[11] Kim, Y., Gu, K., Park, C., Park, C., Schmidgall, S., Heydari, A.A., Yan, Y., Zhang, Z., Zhuang, Y., Liu, Y., Malhotra, M., Liang, P.P., Park, H.W., Yang, Y., Xu, X., Du, Y., Patel, S., Althoff, T., McDuff, D., Liu, X.: Towards a science of scaling agent systems. Preprint at https://arxiv.org/abs/2512.08296 (2025)

[12] Shishika, D., Kumar, V.: A review of multi agent perimeter defense games. In: Decision and Game Theory for Security (GameSec 2020), pp. 472–485. Springer, ??? (2020)

[13] Shishika, D., Paulos, J., Kumar, V.: Cooperative team strategies for multi-player perimeter-defense games. IEEE Robotics and Automation Letters **5**(2), 2738–2745 (2020)

[14] Jaleel, H., Shamma, J.S.: Distributed optimization for robot networks: From real-time convex optimization to game-theoretic self-organization. Proceedings of the IEEE **108**(11), 1953–1967 (2020)

[15] Chung, T.H., Hollinger, G.A., Isler, V.: Search and pursuit-evasion in mobile robotics: A survey. Autonomous robots **31**(4), 299–316 (2011)

[16] Weintraub, I.E., Von Moll, A., Pachter, M.: Range-limited pursuit-evasion. In: NAECON 2023-IEEE National Aerospace and Electronics Conference, pp. 28–35 (2023). IEEE

[17] Weintraub, I.E., Pachter, M., Garcia, E.: An introduction to pursuit-evasion differential games. In: 2020 American Control Conference (ACC), pp. 1049–1066 (2020). IEEE

[18] Zhou, M., Shaikh, M., Chaubey, V., Haggerty, P., Koga, S., Panagou, D., Atanasov, N.: Control strategies for pursuit-evasion under occlusion using visibility and safety barrier functions. In: 2025 IEEE International Conference on Robotics and Automation (ICRA), pp. 12863–12869 (2025). IEEE

[19] Lee, E.S., Zhou, L., Ribeiro, A., Kumar, V.: Graph neural networks for decentralized multi-agent perimeter defense. Frontiers in Control Engineering **4**, 1104745 (2023) https://doi.org/10.3389/fcteg.2023.1104745

[20] Park, S., Kim, H.T., Lee, S., Joo, H., Kim, H.: Survey on anti-drone systems: components, designs, and challenges. IEEE Access **9**, 42635–42659 (2021) https://doi.org/10.1109/ACCESS.2021.3065926

[21] Maity, D., Pourghorban, A.: Cooperative target defense under communication and sensing constraints. Preprint at https://arxiv.org/abs/2412.09939 (2024)

[22] Otte, M., Kuhlman, M.J., Sofge, D.: Auctions for multi-robot task allocation in communication limited environments. Autonomous Robots **44**, 547–584 (2020)

[23] Acharya, A., Goyette, T., Ataei, M., Stoica, A., Dhiman, V., Khojasteh, M.J.: Engagement-aware agentic pursuit-evasion. Preprint at https://arxiv.org/abs/2607.10986 (2026)

[24] Bommasani, R., Hudson, D.A., Adeli, E., Altman, R., Arora, S., Arx, S., Bernstein, M.S., Bohg, J., Bosselut, A., Brunskill, E., et al.: On the opportunities and risks of foundation models. arXiv preprint arXiv:2108.07258 (2021)

[25] Firoozi, R., *et al.*: Foundation models in robotics: applications, challenges, and the future. International Journal of Robotics Research **44**(5), 701–739 (2025) https://doi.org/10.1177/02783649241281508

[26] Hu, Y., Xie, Q., Jain, V., Francis, J., Patrikar, J., Keetha, N., Kim, S., Xie, Y., Zhang, T., Fang, H.-S., et al.: Toward general-purpose robots via foundation models: A survey and meta-analysis. arXiv preprint arXiv:2312.08782 (2023)

[27] Ichter, B., *et al.*: Do as I can, not as I say: grounding language in robotic affordances. In: Proc. 6th Conference on Robot Learning (CoRL), Proceedings of Machine Learning Research 205, pp. 287–318 (2023)

[28] Jülg, T., Burgard, W., Walter, F.: Refined policy distillation: From vla generalists to rl experts. In: 2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 11677–11684 (2025). IEEE

[29] Matni, N., Ames, A.D., Doyle, J.C.: Towards a theory of control architecture: A quantitative framework for layered multi-rate control. arXiv preprint arXiv:2401.15185 (2024)

[30] Morari, M., Borrelli, F., Bemporad, A.: Predictive Control for Linear and Hybrid Systems. Cambridge Univ. Press, Cambridge, England, UK, ??? (2015)

[31] Mayne, D.Q., Rawlings, J.B., Rao, C.V., Scokaert, P.O.M.: Constrained model predictive control: Stability and optimality. Automatica **36**(6), 789–814 (2000) https://doi.org/10.1016/s0005-1098(99)00214-9

[32] Yu, W., Gileadi, N., Fu, C., Kirmani, S., Lee, K.-H., Arenas, M.G., Chiang, H.-T.L., Erez, T., Hasenclever, L., Humplik, J., et al.: Language to rewards for robotic skill synthesis. arXiv preprint arXiv:2306.08647 (2023)

[33] Dias, M.B., Zlot, R., Kalra, N., Stentz, A.: Market-based multirobot coordination: a survey and analysis. Proceedings of the IEEE **94**(7), 1257–1270 (2006)

[34] Choi, H.-L., Brunet, L., How, J.P.: Consensus-based decentralized auctions for robust task allocation. IEEE Transactions on Robotics **25**(4), 912–926 (2009) https://doi.org/10.1109/TRO.2009.2022423

[35] Lott, J., Honary, V.: Decentralized multi-robot task allocation under degraded communication: a benchmark of performance, reliability, and computation. Preprint at https://arxiv.org/abs/2609.13711 (2026)

[36] Coluccia, A., Fascista, A., Schumann, A., Sommer, L., Dimou, A., Zarpalas, D., Méndez, M., Iglesia, D., González, I., Mercier, J.-P., Gagné, G., Mitra, A., Rajashekar, S.: Drone vs. bird detection: deep learning algorithms and results from a grand challenge. Sensors **21**, 2824 (2021) https://doi.org/10.3390/s21082824

[37] Khan, M.M., Gee, P., Upshaw, J., Hassanalian, M.: Taxidermy birds as platform for flapping wing drones: a bioinspired mechanism for wildlife monitoring. In: AIAA SCITECH 2023 Forum, p. 0836 (2023)

[38] Vosbein, D., Upshaw, J., Maimako, S., Hassanalian, M.: Advancements in flapping wing drone technology and wildlife observation via taxidermy. In: AIAA SCITECH 2026 Forum, p. 2082 (2026)

[39] Papadopoulos, V., Shah, M., Zimmerman, S., Lindsey, J.: Mind viruses: Self-propagating ideas in multi-agent llm systems. arXiv preprint arXiv:2608.10218 (2026)

[40] Cemri, M., Pan, M.Z., Yang, S., Agrawal, L.A., Chopra, B., Tiwari, R., Keutzer, K., Parameswaran, A., Klein, D., Ramchandran, K., Zaharia, M., Gonzalez, J.E., Stoica, I.: Why do multi-agent LLM systems fail? In: Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track (2025)

[41] Huang, J.-t., Zhou, J., Jin, T., Zhou, X., Chen, Z., Wang, W., Yuan, Y., Lyu,

M.R., Sap, M.: On the Resilience of LLM-Based Multi-Agent Collaboration with Faulty Agents. arXiv:2408.00989 (2024)

[42] Yang, Z., Raman, S.S., Shah, A., Tellex, S.: Plug in the safety chip: enforcing constraints for LLM-driven robot agents. In: Proc. IEEE International Conference on Robotics and Automation (ICRA) (2024)

[43] Upshaw, J., Hassanalian, M.: Application of camouflaging fixed-wing, quadcopter, and ornithopter uavs. In: 2023 Regional Student Conferences, p. 72279 (2023)

[44] Andersson, J.A.E., Gillis, J., Horn, G., Rawlings, J.B., Diehl, M.: CasADi: a software framework for nonlinear optimization and optimal control. Mathematical Programming Computation **11**(1), 1–36 (2019) https://doi.org/10.1007/s12532-018-0139-4

[45] Stellato, B., Banjac, G., Goulart, P., Bemporad, A., Boyd, S.: OSQP: an operator splitting solver for quadratic programs. Mathematical Programming Computation **12**(4), 637–672 (2020) https://doi.org/10.1007/s12532-020-00179-2

[46] Wächter, A., Biegler, L.T.: On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. Mathematical Programming **106**(1), 25–57 (2006) https://doi.org/10.1007/s10107-004-0559-y

[47] Ames, A.D., Coogan, S., Egerstedt, M., Notomista, G., Sreenath, K., Tabuada, P.: Control barrier functions: Theory and applications. In: 2019 18th European Control Conference (ECC), pp. 3420–3431. IEEE, ??? (2019). https://doi.org/10.23919/ecc.2019.8796030

[48] Dhiman, V., Khojasteh, M.J., Franceschetti, M., Atanasov, N.: Control barriers in bayesian learning of system dynamics. IEEE transactions on automatic control **68**(1), 214–229 (2021)

[49] Cheng, R., Khojasteh, M.J., Ames, A.D., Burdick, J.W.: Safe multi-agent interaction through robust control barrier functions with learned uncertainties. In: 2020 59th IEEE Conference on Decision and Control (CDC), pp. 777–783 (2020). IEEE

[50] Gilbert, E.N.: Capacity of a burst-noise channel. Bell System Technical Journal **39**(5), 1253–1265 (1960) https://doi.org/10.1002/j.1538-7305.1960.tb03959.x

[51] Elliott, E.O.: Estimates of error rates for codes on burst-noise channels. Bell System Technical Journal **42**(5), 1977–1997 (1963) https://doi.org/10.1002/j.1538-7305.1963.tb00955.x

[52] Franceschetti, M., Meester, R.: Random Networks for Communication: From Statistical Physics to Information Systems. Cambridge Series in Statistical and Probabilistic Mathematics, vol. 24. Cambridge University Press, ??? (2007)

[53] Google DeepMind: Gemini 3.1 Flash-Lite Model Card. https://deepmind.google/models/model-cards/gemini-3-1-flash-lite/. Published 3 March 2026; accessed 5 October 2026 (2026)

[54] OpenAI: GPT-6.1 Sol Model. https://developers.openai.com/api/docs/models/gpt-6.1-sol. OpenAI API documentation; accessed 17 October 2026 (2026)

[55] Vosbein, D., McDonagh, K., Goodyear, S., Hassanalian, M.: Swimming robotic taxidermy duck for wildlife monitoring: Development and initial testing. In: AIAA AVIATION FORUM AND ASCEND 2025, p. 3527 (2025)

[56] Glasserman, P., Yao, D.D.: Some guidelines and guarantees for common random numbers. Management Science **38**(6), 884–908 (1992) https://doi.org/10.1287/mnsc.38.6.884

[57] Newcombe, R.G.: Improved confidence intervals for the difference between binomial proportions based on paired data. Statistics in Medicine **17**(22), 2635–2650 (1998) https://doi.org/10.1002/(sici)1097-0258(19981130)17:22⟨2635::aid-sim954⟩3.0.co;2-c

S1 Terms used in this document

Game. Four defenders protect a region against two or four intruders in a simulated arena. Every agent on both teams consults a large language model (LLM) once per decision epoch of 1 s. Unless stated otherwise the LLM is Gemini 3.1 Flash-Lite, the low-cost LLM. GPT-6.1 Sol is the near-frontier LLM. A breach is an intruder entering the region within the 40 s episode, and a team's breach rate is the share of its episodes with a breach. An agent is active until it is destroyed, disabled or leaves the arena. A contact is any object a defender detects, whose kind it is not told. Each agent acts on its beliefs: for each opponent it knows of, the last observed position and velocity, from its own sensor or a teammate's message, projected forward from the time of that observation. A known intruder is one that some defender holds a belief about. In the main study every LLM works from a state-based observation, a text description of positions, velocities and the values computed from them, with no images. A seed is one of 30 random starting configurations (4001 to 4030), and episodes that share a seed and an intruder count are matched across teams and conditions.
Defender teams. In the commander team, the LLM of one defender assigns a task to every defender, and its orders travel over the network. In the consensus team, defenders bid for tasks and agree on who takes which through the consensus-based auction algorithm (CBAA), an algorithm rather than a discussion among their LLMs. The independent team runs the same agents with the auction messages switched off, so each defender commits to its own highest bid. An LLM's authority is how much of each decision it makes itself rather than leaving it to fixed rules. At low authority, each defender scores every task with a fixed formula. Its LLM adjusts the scores only by choosing a weight profile, a priority for each intruder and up to three preferred tasks, so it can make a known intruder less attractive but never drop it. At high authority, the consensus team's LLMs instead enter their own value for each of the 12 weights of the formula, within allowed ranges, and a multiplier from 0 to 2 on each intruder's pursuit value. They can thus set any intruder's pursuit value to zero, and the bids enter the same auction. High authority does not include every low-authority choice (Supplementary Table 12). The independent team always runs at low authority. The consensus team runs on low-authority unless stated otherwise. The intruders are always low-cost LLM agents, and are same for every team.
Network conditions, always in this order: the ideal network (complete and lossless, with one round of latency per link), independent loss, burst loss, falsified reports from one compromised defender, and limited range (links of 15 units, with relaying). Under independent loss each link loses each 0.1 s exchange with a 30% chance, independently of other links and exchanges, and under burst loss the rate is the same on average, but losses come in runs.
Airspace challenges. One bird and two civilian drones, together the non-hostile contacts, fly among four intruders on the ideal network (seeds 4001 to 4005). The defender and commander LLMs then receive a multimodal observation, the same text plus one photograph of every contact, and their prompts gain the airspace paragraph. In the disguise runs each intruder is shown as a photograph of a flapping-wing drone built from a preserved bird, which makes it a disguised intruder. These runs are made without and with a one-sentence forewarning that some intruders may be disguised

as birds. A contact is set aside when an LLM's answer takes it out of pursuit as far as its interface allows: no assigned defender for a commander, a pursuit multiplier of zero for the consensus team at high authority, and a low priority at low authority. A contact-epoch is one contact in one epoch.

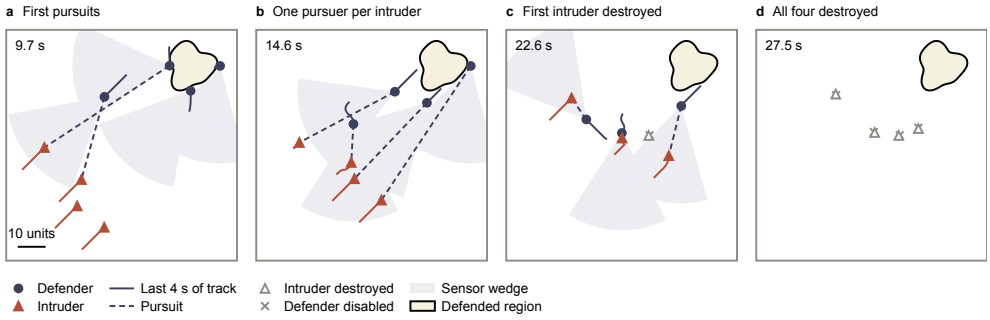
Barrier filter. Every agent's acceleration passes through a barrier filter on the candidate barrier $h = \|r\|^2 - D^2$, where r is the relative position of two agents and $D = 3$ units. The filter guarantees one step of separation only when its constraints are feasible.

S2 Supplementary Figures

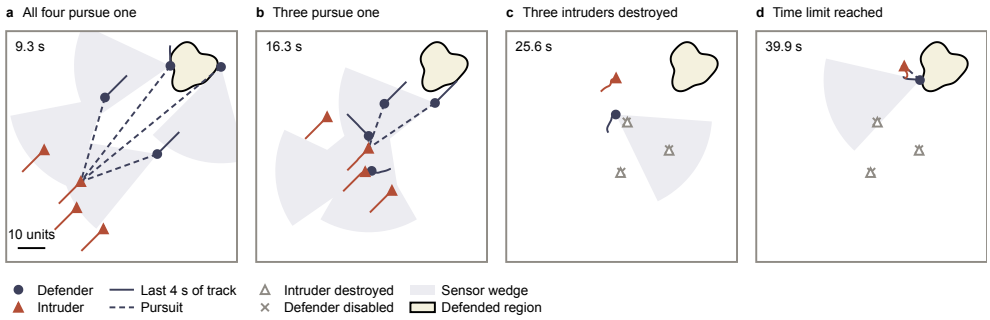
Supplementary Figs. 2–6, 10 and 11 show screenshots from single re-simulated episodes, with times counted from the start. Each episode was chosen by a rule fixed before any image was drawn, usually the first seed that shows the behavior in question. The moments within each episode were picked by hand, to show how it unfolded. These screenshots are illustrative only. They show that a behavior can occur and what it looks like, but not how often it occurs or whether it is typical, because other seeds end differently. For how often each outcome occurs, see the Supplementary Tables.



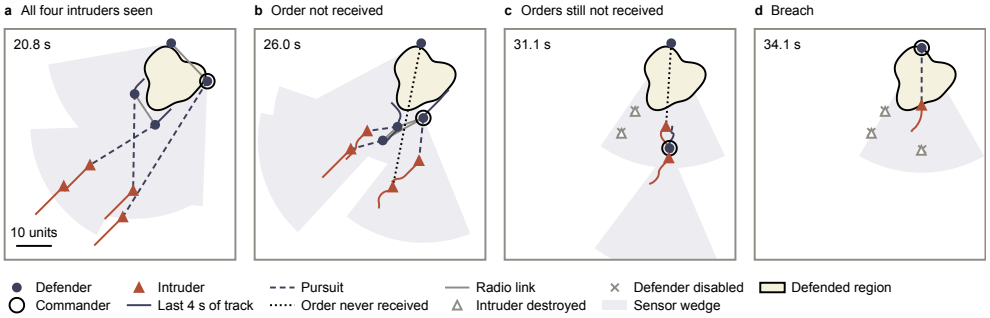
Supplementary Fig. 1 Breach outcome of every episode in the main study. Each tile is one episode, marked by whether an intruder breached the region within the 40 s episode. Columns are seeds 4001 to 4030; rows are the three defender teams under each network condition. **(a)** Two intruders. **(b)** Four intruders. Bold numbers are breaches of 30 per row.



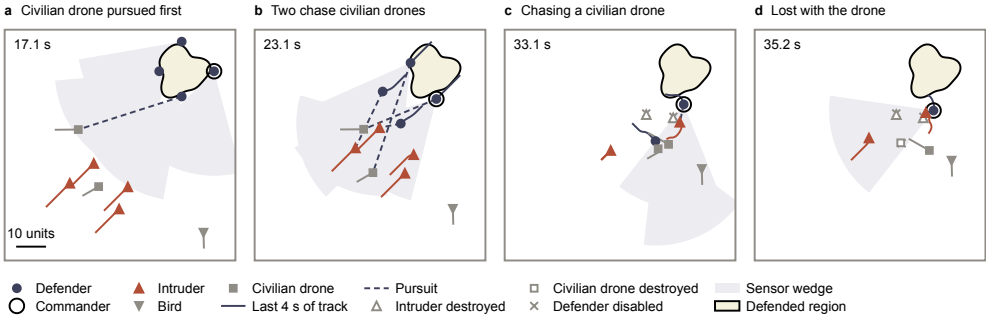
Supplementary Fig. 2 An episode of the consensus team. Ideal network, seed 4003. **(a)** Two defenders pursue the first two intruders seen. **(b)** Every intruder has its own pursuer. **(c)** The first intruder is destroyed, and its pursuer is disabled with it. **(d)** All four intruders are destroyed, and none breached.



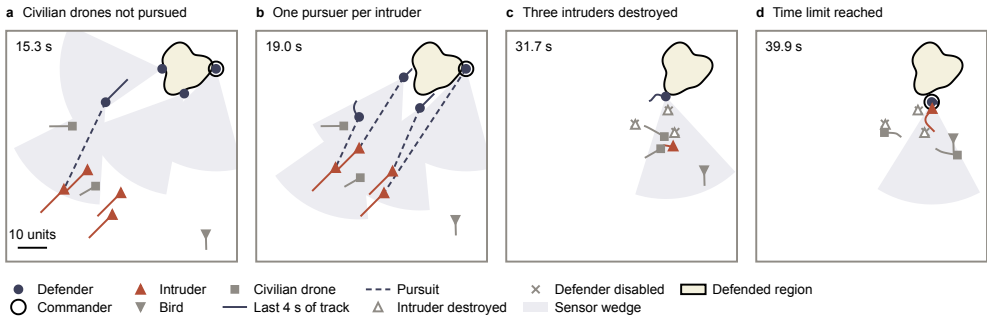
Supplementary Fig. 3 An episode of the independent team. Ideal network, seed 4003, the same starts as Supplementary Fig. 2. **(a)** All four defenders pursue the same intruder. **(b)** Three pursue one intruder while two others go unpursued. **(c)** Three intruders are destroyed, each with a defender. **(d)** The episode reaches the 40 s limit with one intruder alive.



Supplementary Fig. 4 An episode of the commander team under limited range. Seed 4001. Grey lines link defenders within radio range. **(a)** All four intruders are seen, and three defenders pursue. **(b)** The commander's order sending the defender at the top after an intruder never arrives, because no teammate is within range of it. **(c)** Its orders to pursue the intruder that later breaches never arrive either. **(d)** The commander was disabled at 32.2 s, the defender at the top took command and pursued that intruder from 33.0 s, and the intruder breached at 34.1 s.



Supplementary Fig. 5 An episode of the low-cost commander among birds and civilian drones. Ideal network, one bird and two civilian drones, seed 4001. **(a)** The first pursuit the commander assigns is of a civilian drone. **(b)** Two defenders chase civilian drones. **(c)** A defender chases a civilian drone. **(d)** It touched the civilian drone at 35.1 s, and both were destroyed. Two intruders were destroyed and none breached.



Supplementary Fig. 6 An episode of the near-frontier commander among birds and civilian drones. The setting and seed of Supplementary Fig. 5. **(a)** The civilian drones are seen but not pursued. **(b)** Every intruder has its own pursuer. **(c)** Three intruders are destroyed. **(d)** The episode reaches the 40 s limit without a breach. No defender pursued a bird or civilian drone.



(a) Bird 1

Low-cost model (contact 4): *"With only 3 seconds left in the episode (30 steps), and given that the optimistic interception forecasts for blue 1 against the approaching contacts 2, 3, and 4 are all beyond the episode duration (5.4s, 12.7s, and 5.8s respectively), it is impossible for blue 1 to neutralize these threats before the episode concludes. . . . "*

Near-frontier model (contact 4): *"Contacts 0, 1, and 3 appear to be intruder drones; contact 4 is a bird, and 5 and 6 appear civilian. . . . "*



(b) Bird 2

Low-cost model (contact 4): *". . . Blue 3 is closest to contact 1, Blue 2 is closest to contact 0, Blue 0 is directed to intercept contact 5, and Blue 1 is assigned to contact 4. . . . "*

Near-frontier model (contact 4): *". . . Contact 4 is a bird and contact 5 appears to be a civilian consumer drone, so avoid sacrificing defenders on them. . . . "*



(c) Bird 3

Low-cost model (contact 4): *"Contact 4 is the only detected intruder and is approaching the defended region. . . . "*

Near-frontier model (contact 4): *"Contact 4 is visibly a bird, so intercepting it would needlessly disable a defender. . . . "*

Supplementary Fig. 7 The bird photographs and the commander's reasoning about them. Each photograph is shown as the LLMs received it, with the reasoning of the commander team's LLM in the airspace challenge (low-cost model, Gemini 3.1 Flash-Lite; near-frontier model, GPT-6.1 Sol). For each photograph and model the quote comes from the first commander call, seeds in order from 4001, that attached the photograph to a contact and named that contact in its reasoning. Every sentence naming the contact is quoted, with ellipses for skipped sentences. Contact numbers are the identifiers the LLM saw, and the prompts call the defenders `blue` and the intruders `red`. Photographs: (a) R. Long, Unsplash License; (b) CGP Grey, CC BY 2.0; (c) W. Warby, CC BY 2.0; cropped and resized.



(a) Civilian drone 1

Low-cost model (contact 5): *"Contact 5 is identified as an approaching threat. . . . "*

Near-frontier model (contact 5): *"Contact 5 appears to be a civilian camera drone, so intercepting it would risk sacrificing a defender without stopping an intruder. . . . "*



(b) Civilian drone 2

Low-cost model (contact 6): *". . . Finally, Contact 6 is the next most urgent (18.1s breach time), and Blue 1 is the best-placed agent to handle it (13.3s). . . . "*

Near-frontier model (contact 6): *"Contacts 0–3 appear to be intruders; contacts 5 and 6 appear to be civilian camera drones and should not consume defenders. . . . "*



(c) Civilian drone 3

Low-cost model (contact 5): *"Contact 5 is the primary threat with a predicted breach time of 17.5 seconds. . . . "*

Near-frontier model (contact 5): *"Contact 5 appears to be a civilian camera drone, so avoid spending a defender on interception. . . . "*

Supplementary Fig. 8 The civilian-drone photographs and the commander's reasoning about them. As in Supplementary Fig. 7. Photographs: (a) succo, Pixabay, CC0 1.0; (b) J. Sorenson, Pexels, CC0 1.0; (c) H. Soloviev, CC BY 4.0; brand names removed, cropped and resized.



(a) Attack drone 1

Low-cost model (contact 0): *"... Blue 3 is assigned to intercept contact 0 (6.6s optimistic interception), Blue 0 is assigned to contact 1 (9.4s), and Blue 2 is assigned to contact 5 (10.0s). ..."*

Near-frontier model (contact 0): *"Contacts 0 and 2 appear armed, while 5 and 6 appear to be civilian camera drones. Keep blue 1 pursuing contact 2 and send blue 3 against the more urgent contact 0; their optimistic interception times are both before forecast breach, though continued tracking is essential. ..."*



(b) Attack drone 2

Low-cost model (contact 1): *"... Blue 3 is assigned to intercept contact 0 (6.6s optimistic interception), Blue 0 is assigned to contact 1 (9.4s), and Blue 2 is assigned to contact 5 (10.0s). ..."*

Near-frontier model (contact 1): *"Contacts 0, 1, and 2 appear weapon-equipped; 5 and 6 appear to be civilian camera drones and should not consume defenders. ... Assign blue 2 to contact 0 and blue 3 to contact 1: their optimistic interception times are 8.0 and 7.4 seconds, both before forecast breaches and with better minimum margin than the reverse pairing. ..."*

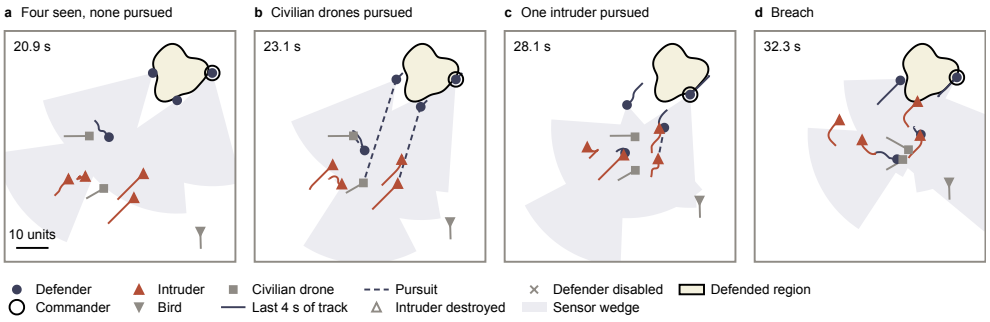


(c) Attack drone 3

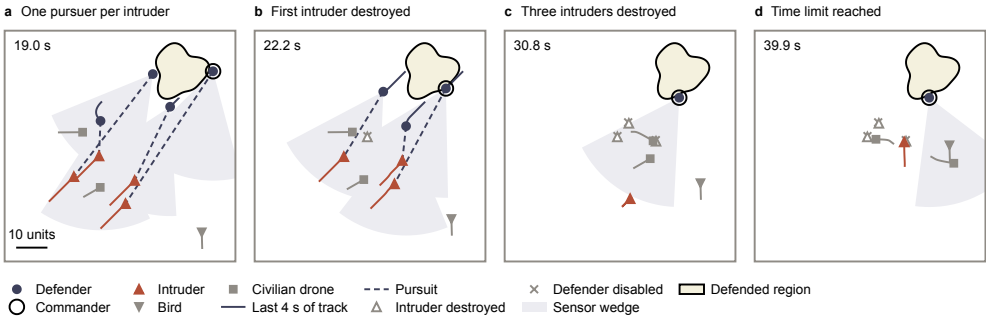
Low-cost model (contact 2): *"... Contact 0 (urgency 0.88), Contact 1 (urgency 0.86), Contact 2 (urgency 0.83), and Contact 3 (urgency 0.83) represent the most critical threats. ... Blue 3 is assigned to intercept Contact 0, Blue 0 to Contact 1, Blue 2 to Contact 2, and Blue 1 to Contact 3, ensuring all four high-urgency threats are addressed by the available team members."*

Near-frontier model (contact 2): *"Contact 2 appears to carry a munition, while contact 5 appears to be a civilian camera drone. Assign blue 1 to intercept contact 2: its optimistic interception time is 7.7 s, comfortably before the forecast breach at 17.1 s. ..."*

Supplementary Fig. 9 The attack-drone photographs of the intruders and the commander's reasoning about them. As in Supplementary Fig. 7. Photographs: (a) ArmyInform (armyinform.com.ua), CC BY 4.0, cropped; (b) U.S. Marine Corps photo by Cpl. J. C. Dela Torre, public domain; (c) U.S. Army photo by Sgt. C. Mackall, public domain; markings removed, cropped and resized.



Supplementary Fig. 10 An episode of the near-frontier commander with disguised intruders, without the forewarning. The setting and seed of Supplementary Fig. 6, but the LLMs saw each intruder as a photograph of a taxidermy bird drone. **(a)** All four intruders are seen, and none is pursued. **(b)** Defenders pursue both civilian drones and one intruder. **(c)** One intruder is pursued. **(d)** An intruder breaches, and no intruder was destroyed.



Supplementary Fig. 11 An episode of the near-frontier commander with disguised intruders, with the forewarning. The episode of Supplementary Fig. 10 with the one-sentence forewarning in the prompt. **(a)** Every intruder has its own pursuer. **(b)** The first intruder is destroyed. **(c)** Three intruders are destroyed. **(d)** The episode reaches the 40 s limit without a breach. No defender pursued a bird or civilian drone.



(a) Bird drone 1

Low-cost model (contact 0): *"... Blue 2 is assigned to NEUTRALIZE:0:0 to address the breach threat from Contact 0. ..."*

Near-frontier model (contact 0): *"Contacts 0–3 appear to be artificial bird-like drones; contact 5 appears to be a civilian quadcopter and should be avoided. ... Blue 1 can intercept contact 0 much sooner than the other blues, completing coverage of the suspected intruders."*



(b) Bird drone 2

Low-cost model (contact 1): *"... Blue 0 is tasked with NEUTRALIZE:1:0 to address Contact 1. ..."*

Near-frontier model (contact 1): *"Contacts 1 and 2 are birds, and contact 5 appears to be a civilian camera quadcopter. ..."*



(c) Bird drone 3

Low-cost model (contact 2): *"... Blue 3 is assigned to Contact 0, Blue 0 to Contact 1, Blue 2 to Contact 2, and Blue 1 to Contact 5, maximizing the likelihood of neutralization before any potential breach."*

Near-frontier model (contact 2): *"Contact 2 is a bird and contact 5 appears to be a civilian camera drone; neither warrants sacrificing a defender. ..."*



(d) Bird drone 4

Low-cost model (contact 3): *"The identified mechanical red intruders (drones) are contacts 3, 5, and 6. ... I am reallocating the blue agents to intercept the drones based on their current proximity and calculated interception times: Blue 3 is closest to contact 5, Blue 0 to contact 3, and Blue 2 to contact 6. ..."*

Near-frontier model (contact 3): *"Contacts 0–3 appear to be artificial bird-like drones; contact 5 appears to be a civilian quadcopter and should be avoided. Assign blue 3 to the most urgent contact 1, blue 2 to contact 2, and blue 0 to contact 3; their optimistic interception times all precede forecast breaches. ..."*

Supplementary Fig. 12 The bird-drone photographs of the disguised intruders and the commander's reasoning about them, without the forewarning. As in Supplementary Fig. 7, for the disguise runs: each intruder was shown as one of these photographs of taxidermy bird drones, and nothing else changed. Photographs: M. Hassanalian, New Mexico Tech.



(a) Bird drone 1

Low-cost model (contact 0): *"Contact 5 (a drone) is the only confirmed threat, as contacts 0 and 1 appear to be birds based on their visual identification. . . . "*

Near-frontier model (contact 0): *"Contacts 0 and 2 appear to be artificial bird decoys and share a direct closing velocity, making them the priority suspected intruders. Keep blue 1 pursuing contact 2 and send blue 3 against the more urgent contact 0; both forecast interceptions precede breach. . . . "*



(b) Bird drone 2

Low-cost model (contact 1): *"Contact 5 (a drone) is the only confirmed threat, as contacts 0 and 1 appear to be birds based on their visual identification. . . . "*

Near-frontier model (contact 1): *"Contacts 0, 1 and 2 show artificial-looking bird features and identical closing velocities, making them the priority suspected intruders. Assign blue 1 to the most urgent contact 0, blue 3 to contact 1, and blue 2 to contact 2; all have forecast interceptions before breach. . . . "*



(c) Bird drone 3

Low-cost model (contact 2): *". . . Blue 0 is assigned to contact 2 and Blue 1 is assigned to contact 5. . . . "*

Near-frontier model (contact 2): *"Contact 2 appears bird-disguised but has mechanical-looking appendages, making it the stronger intruder candidate. . . . "*

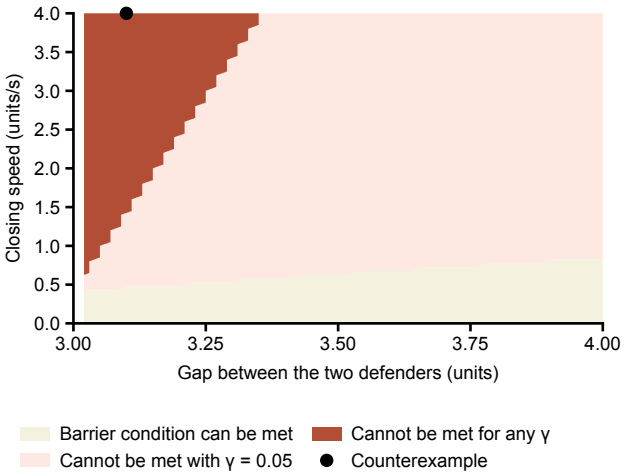


(d) Bird drone 4

Low-cost model (contact 3): *". . . Contact 6 and 3 have longer breach times and will be managed as the tactical situation evolves."*

Near-frontier model (contact 3): *"Contacts 0–3 show artificial or mechanical bird features and are credible disguised intruders; 5 and 6 appear to be civilian quadcopters. . . . Deploy blue 0 against contact 3: its forecast interception precedes breach by 2.7 seconds. . . . "*

Supplementary Fig. 13 The same photographs and the commander's reasoning about them, with the forewarning. As in Supplementary Fig. 12, with the one-sentence forewarning `Some red intruders may be disguised as birds.` in the prompt. Photographs: M. Hassanalian, New Mexico Tech.



Supplementary Fig. 14 The barrier filter's condition cannot be met in some head-on states. Each state is the gap and closing speed of two defenders approaching each other head-on just outside the safety distance of 3 units, so the barrier h, the squared gap minus the squared safety distance, is positive in every state shown. With both defenders braking as hard as the limits allow, the key shows whether the filter's condition can be met at the next step. In the marked counterexample the gap is 3.1 units and the defenders close at 4 units per second, so h is positive, yet the largest attainable h at the next step is −1.49, below the 0.58 the condition requires. On the grid, 612 of 68,850 states have a positive h but no admissible acceleration that keeps h at or above zero, so the condition fails for every choice of γ. An exact computation over a grid of states, with no simulation.

S3 Supplementary Tables

Supplementary Table 1: **Outcomes of every cell.** Each row is one defender team with one intruder count and network condition, with one episode per seed (seeds 4001 to 4030). A breach is an intruder entering the region within the 40 s episode. "Ended at 40 s" counts the episodes without a breach that ran to the time limit with an intruder alive, out of the episodes without a breach. Each intruder alive at that point is classified over its last 10 s as in a stand-off (evading on at least half of the steps, with a defender within 6 units on at least half), closing (otherwise, if its distance to the region fell by more than 1 unit) or other. Over the main study, intruders breached in 35 of 450 episodes with two intruders and 196 of 450 with four, and 78 of 415 and 126 of 254 episodes without a breach ended at 40 s. Of the intruders alive at 40 s, 43 of 85 were closing with two intruders and 113 of 161 with four. Endings come from deterministic re-simulation of every episode from its cached model answers, and every re-simulated episode matched its record (Methods; Supplementary Table 14). The episodes of the consensus team at high authority with the near-frontier model are the valid episodes of two runs (Methods).

Team	Intruders	Network condition	Breaches	Ended at 40 s (no breach)	Alive at 40 s: stand-off, closing, other
Main study, low-cost LLM					
Commander	2	Ideal network	1 of 30	7 of 29	3, 3, 1
	2	Independent loss	0 of 30	9 of 30	3, 6, 1
	2	Burst loss	3 of 30	5 of 27	1, 3, 2
	2	Falsified reports	6 of 30	10 of 24	2, 5, 4
	2	Limited range	8 of 30	3 of 22	0, 2, 1
	4	Ideal network	14 of 30	6 of 16	1, 5, 0
	4	Independent loss	14 of 30	6 of 16	1, 5, 2
	4	Burst loss	18 of 30	6 of 12	4, 3, 1
	4	Falsified reports	16 of 30	10 of 14	5, 9, 1
	4	Limited range	18 of 30	7 of 12	1, 7, 1
Consensus	2	Ideal network	2 of 30	5 of 28	2, 2, 1
	2	Independent loss	3 of 30	3 of 27	2, 1, 0
	2	Burst loss	3 of 30	7 of 27	3, 3, 1
	2	Falsified reports	1 of 30	3 of 29	1, 3, 0
	2	Limited range	0 of 30	4 of 30	3, 1, 0
	4	Ideal network	7 of 30	11 of 23	0, 12, 0
	4	Independent loss	7 of 30	7 of 23	3, 5, 0
	4	Burst loss	6 of 30	7 of 24	2, 3, 2
	4	Falsified reports	11 of 30	12 of 19	5, 8, 3
	4	Limited range	8 of 30	7 of 22	2, 7, 0
Independent	2	Ideal network	1 of 30	2 of 29	0, 2, 1
	2	Independent loss	0 of 30	10 of 30	3, 6, 2
	2	Burst loss	2 of 30	5 of 28	2, 4, 0
	2	Falsified reports	4 of 30	3 of 26	1, 1, 1
	2	Limited range	1 of 30	2 of 29	0, 1, 1
	4	Ideal network	13 of 30	11 of 17	3, 14, 0
	4	Independent loss	15 of 30	14 of 15	3, 14, 2
	4	Burst loss	17 of 30	9 of 13	1, 9, 0
	4	Falsified reports	12 of 30	9 of 18	3, 10, 0
	4	Limited range	20 of 30	4 of 10	1, 2, 1

Supplementary Table 1 (continued)

Team	Intruders	Network condition	Breaches	Ended at 40 s (no breach)	Alive at 40 s: stand-off, closing, other
Consensus team at high authority, low-cost LLM					
Consensus	2	Ideal network	10 of 30	9 of 20	8, 2, 0
	2	Independent loss	10 of 30	6 of 20	4, 2, 1
	2	Burst loss	6 of 30	10 of 24	4, 5, 3
	2	Falsified reports	12 of 30	6 of 18	1, 2, 5
	2	Limited range	8 of 30	8 of 22	1, 3, 4
	4	Ideal network	23 of 30	3 of 7	1, 2, 0
	4	Independent loss	21 of 30	4 of 9	3, 4, 0
	4	Burst loss	23 of 30	5 of 7	1, 4, 1
	4	Falsified reports	20 of 30	2 of 10	2, 2, 0
	4	Limited range	25 of 30	3 of 5	0, 5, 0
Consensus team at high authority, near-frontier LLM					
Consensus	4	Ideal network	7 of 30	6 of 23	3, 4, 0
Commander team, near-frontier LLM					
Commander	2	Ideal network	0 of 30	7 of 30	1, 6, 1
	2	Falsified reports	1 of 30	4 of 29	2, 2, 0
	4	Ideal network	8 of 30	10 of 22	2, 11, 0
	4	Falsified reports	3 of 30	18 of 27	4, 17, 3

Supplementary Table 2 Breach-rate differences between the defender teams. Each row is the mean over 30 seeds of each seed's paired difference in breach rate, first team minus second, in percentage points, with episodes matched by seed and intruder count. Pairs per seed is the number of episode pairs that each seed contributes. The pooled rows cover the ideal network, independent loss, burst loss and limited range with two and four intruders. Falsified reports enter the comparisons within each team (Supplementary Table 4). CIs are two-sided 95% CIs from the t distribution over the 30 seed values (29 degrees of freedom). The pooled rows form the prespecified primary family (Methods), and the rows by intruder count and condition are secondary. a, not estimable, because every seed's difference was zero.

Episodes	Pairs per seed	Estimate (pp)	95% CI (t)
Commander minus consensus			
Pooled	8	+17	+9 to +24
2 intruders, pooled	4	+3	−4 to +11
4 intruders, pooled	4	+30	+18 to +42
2 intruders, ideal network	1	−3	−15 to +9
2 intruders, independent loss	1	−10	−21 to +1
2 intruders, burst loss	1	0	−14 to +14
2 intruders, limited range	1	+27	+10 to +43
4 intruders, ideal network	1	+23	−2 to +49
4 intruders, independent loss	1	+23	−2 to +49
4 intruders, burst loss	1	+40	+15 to +65
4 intruders, limited range	1	+33	+11 to +56
Commander minus independent			
Pooled	8	+3	−5 to +11
2 intruders, pooled	4	+7	+1 to +13[†]
4 intruders, pooled	4	−1	−15 to +13
2 intruders, ideal network	1	0	not estimable[a]
2 intruders, independent loss	1	0	not estimable[a]
2 intruders, burst loss	1	+3	−12 to +19
2 intruders, limited range	1	+23	+7 to +39
4 intruders, ideal network	1	+3	−25 to +32
4 intruders, independent loss	1	−3	−34 to +27
4 intruders, burst loss	1	+3	−23 to +30
4 intruders, limited range	1	−7	−32 to +19
Consensus minus independent			
Pooled	8	−14	−22 to −6
2 intruders, pooled	4	+3	−2 to +9
4 intruders, pooled	4	−31	−45 to −17
2 intruders, ideal network	1	+3	−9 to +15
2 intruders, independent loss	1	+10	−1 to +21
2 intruders, burst loss	1	+3	−12 to +19
2 intruders, limited range	1	−3	−10 to +3
4 intruders, ideal network	1	−20	−45 to +5
4 intruders, independent loss	1	−27	−52 to −1
4 intruders, burst loss	1	−37	−62 to −12
4 intruders, limited range	1	−40	−63 to −17

Supplementary Table 3 Allocation measures by team and network condition. Each entry is pooled over the 60 episodes of a team and condition (two and four intruders, 30 seeds). Duplicate commitments is the share of once-per-second checks, at the end of each decision epoch, at which two or more active defenders were committed to the same task. For the commander team a defender's commitment is the newest order it has accepted, read at the same checks by re-simulation. Unpursued is the share of known-intruder time, over the steps at which some active defender held a belief about an active intruder, in which no active defender was executing its pursuit. Unseen is the share of active intruder time in which no defender had yet seen the intruder. Scouting is the share of defender-steps before the team's first contact in which a defender executed a scouting task. The last three columns give the share of a pursuer's control steps on which the barrier filter changed its acceleration by more than 0.01, in total and by the kind of agent whose constraint was binding (teammates or disabled defenders, or intruders). Pooled over the ideal network, independent loss, burst loss and limited range, known intruders went unpursued for 14% of their time in the commander team, 13% in the consensus team and 42% in the independent team, and intruders were unseen for 59% and 41% of their time in the commander and consensus teams. Pooled over all five conditions, the filter changed a pursuer's acceleration because of teammates or disabled defenders on 10%, 7% and 19% of its control steps in the three teams. Coverage, scouting and filter measures come from deterministic re-simulation, with every episode matching its record (Methods). Descriptive, without CIs.

					Filter changed pursuer acceleration		
Network condition	Duplicate commitments	Unpursued	Unseen	Scouting before first contact	All	Teammates or disabled	Intruders
Commander							
Ideal network	0%	11%	60%	0%	34%	9%	26%
Independent loss	0%	15%	58%	0%	36%	10%	26%
Burst loss	0%	14%	59%	0%	34%	8%	25%
Falsified reports	0%	15%	58%	0%	37%	10%	28%
Limited range	3%	17%	59%	0%	37%	12%	25%
Consensus							
Ideal network	0%	7%	40%	36%	18%	6%	12%
Independent loss	0%	8%	40%	40%	17%	6%	10%
Burst loss	1%	9%	42%	38%	18%	6%	11%
Falsified reports	0%	7%	39%	36%	17%	6%	10%
Limited range	27%	29%	42%	85%	23%	10%	13%
Independent							
Ideal network	68%	43%	41%	40%	24%	17%	7%
Independent loss	72%	42%	38%	43%	28%	21%	7%
Burst loss	71%	41%	40%	43%	27%	19%	8%
Falsified reports	71%	40%	39%	44%	24%	17%	7%
Limited range	48%	42%	41%	87%	30%	18%	11%

Supplementary Table 4: **Change in breach rate under each network condition, by intruder count.** Each row is the mean over 30 seeds of each seed's paired difference in breach rate, in percentage points, with episodes matched by seed and intruder count, and its two-sided 95% CI from the t distribution (29 degrees of freedom). Within each condition, the first row pools all three teams, the next rows give each team, and the last rows give the difference between two teams' changes. Intruders "Both" pools two and four intruders. The pooled change under falsified reports belongs to the prespecified primary family, and the commander team's change and the commander-minus-consensus difference under falsified reports were added during the study (Methods). The differences between the two loss conditions come from a post hoc analysis. The study stored between-team differences only for falsified reports, limited range and the difference between the two loss conditions. Breach rates by cell are in Supplementary Table 1.

Comparison	Intruders	Pairs per seed	Estimate (pp)	95% CI (t)
Independent loss minus ideal network				
All teams	Both	6	+1	−6 to +8
Commander	2	1	−3	−10 to +3
	4	1	0	−26 to +26
	Both	2	−2	−16 to +13
Consensus	2	1	+3	−12 to +19
	4	1	0	−22 to +22
	Both	2	+2	−13 to +17
Independent	2	1	−3	−10 to +3
	4	1	+7	−17 to +31
	Both	2	+2	−11 to +14
Burst loss minus ideal network				
All teams	Both	6	+6	−3 to +16
Commander	2	1	+7	−7 to +20
	4	1	+13	−12 to +39
	Both	2	+10	−6 to +26
Consensus	2	1	+3	−12 to +19
	4	1	−3	−26 to +20
	Both	2	0	−15 to +15
Independent	2	1	+3	−9 to +15
	4	1	+13	−10 to +37
	Both	2	+8	−6 to +23
Burst loss minus independent loss				
Commander	Both	2	+12	−3 to +26
Consensus	Both	2	−2	−17 to +13
Independent	Both	2	+7	−7 to +20
Commander minus consensus	Both	2	+13	−9 to +35
Commander minus independent	Both	2	+5	−16 to +26
Independent minus consensus	Both	2	+8	−10 to +27
Falsified reports minus ideal network				
All teams	Both	6	+7	−2 to +15
Commander	2	1	+17	+3 to +31[†]
	4	1	+7	−21 to +34
	Both	2	+12	−4 to +27
Consensus	2	1	−3	−15 to +9

Supplementary Table 4 (continued)

Comparison	Intruders	Pairs per seed	Estimate (pp)	95% CI (t)
	4	1	+13	−8 to +35
	Both	2	+5	−6 to +16
Independent	2	1	+10	−5 to +25
	4	1	−3	−32 to +25
	Both	2	+3	−14 to +21
Commander minus consensus	2	1	+20	−1 to +41
	4	1	−7	−46 to +32
	Both	2	+7	−15 to +28
Commander minus independent	2	1	+7	−13 to +26
	4	1	+10	−30 to +50
	Both	2	+8	−14 to +31
Consensus minus independent	2	1	−13	−32 to +6
	4	1	+17	−16 to +49
	Both	2	+2	−18 to +22
Limited range minus ideal network				
All teams	Both	6	+9	+1 to +18†
Commander	2	1	+23	+7 to +39
	4	1	+13	−16 to +42
	Both	2	+18	+1 to +36†
Consensus	2	1	−7	−16 to +3
	4	1	+3	−17 to +24
	Both	2	−2	−13 to +10
Independent	2	1	0	−10 to +10
	4	1	+23	−2 to +49
	Both	2	+12	−3 to +26
Commander minus consensus	2	1	+30	+13 to +47
	4	1	+10	−23 to +43
	Both	2	+20	+2 to +38
Commander minus independent	2	1	+23	+7 to +39
	4	1	−10	−51 to +31
	Both	2	+7	−16 to +30
Consensus minus independent	2	1	−7	−20 to +7
	4	1	−20	−55 to +15
	Both	2	−13	−32 to +6

Supplementary Table 5 Exposure to falsified reports. Attacked episodes are the episodes with falsified reports (both intruder counts, 30 seeds). From its first own sighting of an intruder, the compromised defender shifted that intruder's position by 10 units in the reports of its own sightings that it sent (Methods). A defender accepted a falsified report when the ordinary merge rule stored it in place of its own record. Defender 0 is the commander in the commander teams. A pursuit on a falsified position counts an episode in which a defender pursued the targeted intruder while holding the falsified record of it. Breaches equal the falsified-reports cells of Supplementary Table 1. Counts come from deterministic re-simulation of the attacked episodes, with every episode matching its record (Methods). Descriptive, without CIs.

	Low-cost LLM			Near-frontier LLM
	Commander	Consensus	Independent	Commander
Attacked episodes	60	60	60	60
Episodes with falsified reports sent	60	55	60	53
Episodes with an acceptance	56	50	50	41
Episodes with an acceptance by defender 0	55	50	50	41
Reports sent per episode (median)	83.5	80	89.5	68.5
Acceptances per episode (median)	53	93	53.5	95.5
Episodes with a pursuit on a falsified position	30	24	41	14
Episodes with a breach	22	12	16	4
Breaches by the targeted intruder (episodes)	7	3	6	1

Supplementary Table 6 Network connectivity under limited range. Links are limited to 15 units, and messages reach distant defenders by rebroadcast (Methods). A communication round is connected when the active defenders form one group over the links in range. Connected is the share of rounds that were connected: the mean over episodes, and its 10th, 50th and 90th percentiles over episodes. Never connected counts the episodes in which no round was connected. Largest group is the mean share of active defenders in the largest group, and groups is the mean number of groups per round. A partition is a run of consecutive rounds that were not connected, including a run still open at the end of the episode. Mean partition is the mean over episodes of each episode's mean partition length (an episode without one counts as 0 s), and longest partition is the median over episodes of each episode's longest. Intruders "Both" pools the 30 episodes with two intruders and the 30 with four. All measures come from deterministic re-simulation, with every episode matching its record (Methods). Descriptive, without CIs.

	Connected (% of rounds)					Partition length (s)	
Intruders	Mean	Percentiles 10 / 50 / 90	Never connected	Largest group (%)	Groups	Mean	Longest
Commander							
2	55	47 / 54 / 63	0 of 30	79	1.7	7.2	10.5
4	67	47 / 71 / 85	0 of 30	83	1.6	5.6	4.8
Both	61	47 / 56 / 81	0 of 60	81	1.6	6.4	8.8
Consensus							
2	2	0 / 0 / 9	23 of 30	43	3.1	25.6	27.2
4	27	14 / 27 / 41	1 of 30	57	2.5	19.6	23.0
Both	14	0 / 11 / 35	24 of 60	50	2.8	22.6	25.2
Independent							
2	8	0 / 0 / 26	22 of 30	46	3.0	26.3	25.4
4	31	8 / 31 / 51	2 of 30	60	2.5	22.7	23.1
Both	19	0 / 16 / 51	24 of 60	53	2.8	24.5	24.2

Supplementary Table 7 Bid weights and pursuit multipliers chosen by the LLMs at high authority. Each answer at high authority sets the 12 weights of the bid formula (symbols and order as in Supplementary Table 12), each within its allowed range, and a multiplier from 0 to 2 on the pursuit value of each intruder known to the defender (Methods). Entries are the weights applied after validation and budget scaling, as medians with 25th to 75th percentiles over answers. Fig. 6a plots five of these weights against the values available at low authority, which Supplementary Table 12 lists. Both teams ran on the ideal network with four intruders and seeds 4001 to 4030 (30 episodes each). Before contact counts a defender's answers before it first knew of an intruder, and after contact the rest. The five value weights may sum to at most 3.2, their default total, and the multipliers to at most the number of known intruders; a budget overrun is scaled down in proportion. A multiplier of 0 leaves the intruder's pursuit bid with only its costs. Answers were read from the stored responses of the valid episodes, without simulation. Descriptive, without CIs.

		Low-cost LLM		Near-frontier LLM	
Weight	Default (allowed range)	Before contact	After contact	Before contact	After contact
Threat value (w_T)	1.20 (0–3)	1.20 (1.20–1.20)	1.50 (1.50–1.50)	0.30 (0.20–0.40)	0.80 (0.65–1.00)
Engagement value (w_E)	1.00 (0–3)	1.00 (1.00–1.00)	1.20 (1.20–1.50)	0.35 (0.25–0.45)	1.65 (1.40–1.90)
Station value (w_S)	0.05 (0–1.5)	0.10 (0.05–0.50)	0.10 (0.05–0.10)	0.20 (0.15–0.25)	0.05 (0.05–0.15)
Station coverage (w_C)	0.50 (0–3)	0.50 (0.50–0.50)	0.20 (0.10–0.30)	1.05 (0.80–1.35)	0.25 (0.15–0.50)
Scouting information (w_I)	0.45 (0–3)	0.45 (0.10–0.45)	0.10 (0.05–0.20)	1.20 (0.90–1.60)	0.20 (0.15–0.30)
Travel cost (w_D)	0.15 (0–0.6)	0.15 (0.15–0.15)	0.20 (0.15–0.30)	0.20 (0.20–0.20)	0.20 (0.15–0.30)
Scouting travel cost (w_R)	0.05 (0–0.3)	0.05 (0.05–0.05)	0.05 (0.05–0.10)	0.18 (0.12–0.25)	0.10 (0.10–0.15)
Belief-age cost (w_A)	0.10 (0–1)	0.10 (0.10–0.10)	0.10 (0.10–0.20)	0.20 (0.15–0.20)	0.15 (0.10–0.20)
Attrition cost (w_X)	0.50 (0–2)	0.50 (0.50–0.50)	0.50 (0.50–0.80)	0.50 (0.50–0.50)	0.25 (0.15–0.50)
Leaving-station cost (w_L)	0.25 (0–1.5)	0.25 (0.25–0.25)	0.25 (0.25–0.50)	0.25 (0.25–0.35)	0.25 (0.15–0.25)
Switching cost (w_W)	0.05 (0–0.5)	0.05 (0.05–0.05)	0.10 (0.05–0.20)	0.08 (0.05–0.12)	0.08 (0.04–0.12)
Continuity cost (w_P)	0.15 (0–1)	0.15 (0.15–0.15)	0.20 (0.15–0.50)	0.15 (0.15–0.15)	0.25 (0.15–0.40)
Threat + engagement	2.20	2.20 (2.20–2.20)	2.70 (2.70–3.00)	0.70 (0.50–0.85)	2.60 (2.25–2.80)
Station + coverage	0.55	0.60 (0.55–1.00)	0.30 (0.20–0.40)	1.30 (0.90–1.60)	0.35 (0.20–0.65)
Answers		1,823	2,022	1,140	2,345
Value budget scaled		0	0	0	0
Multiplier budget scaled		0	43	0	0
Pursuit multipliers set		–	6,276	–	6,916
equal to 0		–	135	–	44
below 1		–	2,876	–	3,049
above 1		–	2,093	–	3,216

Supplementary Table 8 High authority and model capability: comparisons. Each row is a paired difference, first team minus second, in percentage points, with episodes matched by seed and intruder count. In the first two blocks, breach rates and unseen time pool the ideal network, independent loss, burst loss and limited range. Rows with several pairs per seed give the mean over 30 seeds of each seed's difference with a two-sided 95% CI from the t distribution. Rows with one pair per seed (the near-frontier model at high authority, and the near-frontier commander by intruder count) give Newcombe's hybrid score CI (method 10; N). On the ideal network with four intruders, intruders were unseen for 39%, 54% and 40% of their active time for the consensus team at high authority with the near-frontier and the low-cost model and at low authority, and known intruders went unpursued for 24%, 11% and 8% of their time. Over the four conditions, intruders were unseen for 53% of their time at high authority and 41% at low authority, both with the low-cost model. The 30 episodes of the near-frontier model at high authority come from two runs, 22 seeds from the first and 8 from the second (Methods).

Comparison	Intruders	Pairs per seed	Estimate (pp)	95% CI
High minus low authority, consensus team, low-cost LLM				
Breach rate	Both	8	+38	+30 to +45
	2	4	+22	+12 to +31
	4	4	+53	+44 to +62
Change under falsified reports	Both	2	−7	−28 to +15
	2	1	+10	−18 to +38
	4	1	−23	−55 to +9
Intruder time unseen	Both	8	+12	+10 to +14
Consensus team at high authority, low-cost LLM, minus the other main-study teams				
Minus commander	Both	8	+21	+12 to +29
Minus independent	Both	8	+24	+17 to +30
Consensus team at high authority, near-frontier LLM, ideal network, four intruders				
Minus the low-cost LLM at high authority	4	1	−53	-70 to -28^{N}
Minus the low-cost LLM at low authority	4	1	0	-24 to $+24^{N}$
Minus near-frontier commander	4	1	−3	-23 to $+17^{N}$
Near-frontier minus low-cost commander				
Ideal network	Both	2	−12	−26 to +3
	2	1	−3	-17 to $+8^{N}$
	4	1	−20	-42 to $+6^{N}$
Falsified reports	Both	2	−30	−42 to −18
	2	1	−17	-34 to $+1^{N}$
	4	1	−43	-62 to -19^{N}
Change under falsified reports	Both	2	−18	−38 to +1
	2	1	−13	−30 to +3
	4	1	−23	−60 to +13

Supplementary Table 9 Episodes with birds and civilian drones, by seed. Ideal network, four intruders, one bird and two civilian drones per episode, seeds 4001 to 4005. Without refers to the matched episode of the same team without birds and civilian drones (the main study, or the team's own runs for the consensus team at high authority and the near-frontier commander). An episode ends at a breach, at the time limit of 40 s, or when every intruder has been destroyed or has left the arena. Every defender that touches a contact is disabled, so the defenders lost to intruders equal the intruders destroyed. The low-cost teams lost 3 to 4 defenders to birds and civilian drones each, and the near-frontier model lost 0 at high authority and 0 as commander. Counts at five seeds are descriptive, without CIs.

	Breach				Defenders lost to	
Seed	Without	With	Ending with them	Intruders destroyed	Birds	Civilian drones
Consensus at low authority, low-cost LLM						
4001	no	yes	breach	2	0	1
4002	yes	yes	breach	1	0	0
4003	no	no	time limit	3	0	1
4004	no	yes	breach	2	0	1
4005	yes	yes	breach	2	0	1
Total	2	4		10	0	4
Commander, low-cost LLM						
4001	no	no	time limit	2	0	1
4002	no	no	time limit	2	0	1
4003	no	yes	breach	1	0	1
4004	no	no	time limit	2	1	0
4005	no	no	time limit	2	0	1
Total	0	1		9	1	4
Consensus at high authority, low-cost LLM						
4001	yes	yes	breach	0	0	0
4002	yes	no	time limit	2	0	1
4003	yes	no	time limit	3	0	0
4004	yes	no	time limit	3	0	1
4005	yes	no	time limit	1	0	1
Total	5	1		9	0	3
Consensus at high authority, near-frontier LLM						
4001	yes	no	all intruders resolved	4	0	0
4002	no	no	all intruders resolved	4	0	0
4003	yes	no	all intruders resolved	4	0	0
4004	no	no	all intruders resolved	4	0	0
4005	no	no	all intruders resolved	4	0	0
Total	2	0		20	0	0
Commander, near-frontier LLM						
4001	no	no	time limit	3	0	0
4002	no	no	time limit	2	0	0
4003	yes	no	time limit	3	0	0
4004	no	no	time limit	3	0	0
4005	no	no	time limit	3	0	0
Total	1	0		14	0	0

Supplementary Table 10 Answers about each kind of contact with birds and civilian drones present. Counts over the five episodes with birds and civilian drones of each team (ideal network, four intruders, seeds 4001 to 4005) that underlie the "set aside" shares of Fig. 7c (main paper) and Supplementary Table 11. For the consensus team at low authority, each count is one contact the defender held a belief about, in one answer, with the priority the answer gave it (medium if the answer left the contact out). At high authority, each count is one contact the defender held a belief about, in one answer, with the multiplier the answer wrote (1 if left out), before validation and budget scaling. For the commanders, each count is one contact in the commander's catalog in one epoch, counted as assigned if the answer gave any defender a pursuit task on it. The near-frontier commander assigned a defender to a bird or civilian drone in 2 of the 361 contact-epochs in which it saw one, and the near-frontier model at high authority set aside 100% of birds and 91% of civilian drones in its answers. Answers were tallied from the stored responses as written, before validation. Counts at five seeds are descriptive, without CIs.

Answer	Intruders	Birds	Civilian drones
Consensus at low authority, low-cost LLM			
High priority	459	19	172
Medium priority	475	41	222
Low priority	83	235	55
Commander, low-cost LLM			
Defender assigned	192	14	76
No defender assigned	120	88	107
Consensus at high authority, low-cost LLM			
Multiplier above 1	551	22	164
Multiplier 1	366	84	239
Multiplier below 1	244	200	204
Multiplier 0	6	50	18
Consensus at high authority, near-frontier LLM			
Multiplier above 1	893	0	4
Multiplier 1	37	0	15
Multiplier below 1	103	0	34
Multiplier 0	6	269	550
Commander, near-frontier LLM			
Defender assigned	280	0	2
No defender assigned	55	123	236

Supplementary Table 11 Outcomes and treatment of contacts in airspace with birds and civilian drones, and with disguised intruders. Five seeds (4001 to 4005), ideal network, four intruders, one bird and two civilian drones per episode. In the lower block the intruders were disguised intruders, each shown to the LLMs as a photograph of a bird drone (main paper's Fig. 7b), without and with the forewarning sentence (Methods); birds, civilian drones and everything else were as in the upper block. Breaches are counts of five episodes, without birds and civilian drones (the matched episodes of the main study) and with them. Defenders lost to birds and civilian drones are summed over the five episodes; without the disguise no episode lost more than one. Intruders destroyed are counted of the 20 intruders in the five episodes. "Set aside" is the share of the LLMs' answers or epochs that took a contact out of pursuit, by contact kind: for the consensus team at low authority, a low priority (which halves the contact's urgency but keeps its pursuit task); at high authority, a pursuit multiplier of zero; for the commanders, a contact-epoch in which no defender was assigned to the contact. Counts and shares are descriptive; no confidence intervals are given at five seeds.

	Low-cost LLM			Near-frontier LLM	
	Consensus, low authority	Commander	Consensus, high authority	Consensus, high authority	Commander
Breaches, main study (of 5)	2	0	5	2	1
Breaches with birds and civilian drones (of 5)	4	1	1	0	0
Defenders lost to birds	0	1	0	0	0
Defenders lost to civilian drones	4	4	3	0	0
Set aside (%): intruders	8	38	1	1	16
Set aside (%): birds	80	86	14	100	100
Set aside (%): civilian drones	12	58	3	91	99
Intruders destroyed (of 20)	10	9	9	20	14
Disguised intruders: without / with forewarning					
Breaches (of 5)	5 / 3	3 / 5	5 / 5	3 / 0	3 / 1
Intruders destroyed (of 20)	4 / 4	8 / 10	4 / 5	4 / 14	2 / 14
Defenders lost to birds and civilian drones	5 / 11	2 / 4	1 / 4	2 / 0	2 / 1
Set aside (%): intruders	44 / 56	51 / 42	10 / 3	59 / 0	61 / 25
Set aside (%): birds	48 / 72	75 / 74	18 / 5	100 / 23	100 / 100
Set aside (%): civilian drones	18 / 7	47 / 42	5 / 1	72 / 33	77 / 93

Supplementary Table 12: **Notation.** Every symbol in the Methods of the paper, grouped by the part of Methods that introduces it. Value or default gives each run setting and the default of each weight, which is that of the balanced profile. The last two columns give the values that the consensus team's LLM can set. At low authority it changes w_C, w_I, w_X, w_L and T only together, by choosing one of four weight profiles, and sets a priority for each known intruder (Methods). At high authority it enters each value within the range shown, the five value weights may sum to at most 3.2, their default total, and the multipliers to at most the number of known intruders. A dash marks a value that the LLM cannot change, and an empty cell a quantity that the defender computes or an index.

Symbol	Meaning	Value or default	Low authority	High authority
Game and dynamics				
Δt	simulation step	0.1 s		
k	step index			
x	state of an agent, $[p^\top, v^\top, \theta]^\top$			
p	planar position			
v; v_x, v_y	velocity; its components	$\lvert v_x\rvert, \lvert v_y\rvert \le 2\,\text{units}\,\text{s}^{-1}$		
θ	sensor heading			
u	input of an agent, $[a^\top, \omega]^\top$			
a; a_x, a_y	translational acceleration; its components	$\lvert a_x\rvert, \lvert a_y\rvert \le 2\,\text{units}\,\text{s}^{-2}$		
ω	yaw rate	$\lvert\omega\rvert \le 2\pi\,\text{rad}\,\text{s}^{-1}$		
Barrier filter				
r	relative position of two agents			
D	safety distance	3 units		
h; h_k	barrier, $\lVert r\rVert^2 - D^2$; its value at step k			
γ	rate in the barrier condition	0.05		
Tasks and bids				
q	candidate task			
j	intruder			
s	station			
g	cell of the coverage grid			
b_q	bid for task q			
$\mathbf{1}[\cdot]$	indicator: 1 if the condition in brackets holds, else 0			
V_q, C_q	value and cost of task q			
K_q	change cost, part of C_q			
Q	tasks marked as preferred	empty	up to three	up to three
β	bonus for each preferred task	0.2	–	–
Constants				
L	length scale, a quarter of the arena	25 units	–	–
T	urgency time scale	60 s	60 or 90 s	–
Computed by the defender for each task				
d_q	distance to task q			
τ_j	forecast time for intruder j to reach the region			

Supplementary Table 12 (continued)

Symbol	Meaning	Value or default	Low authority	High authority
δ_j	distance from intruder j to the region			
U_j	urgency of intruder j			
O_j	opportunity to intercept intruder j in time			
m_s	unwatched approach sector that station s would watch			
I_g	unwatched approach sector swept on the way to cell g			
α_j	age of the belief about intruder j, as a distance			
n	number of active teammates			
ℓ_q, m_0	leaving the held station, and its coverage			
σ_q	switching to a different task			
c_q	dropping the pursuit of a known intruder			
Value weights				
w_T	threat value of every known intruder	1.2	–	0–3
w_E	engagement value, on $\pi_j U_j O_j$	1	–	0–3
w_S	station value	0.05	–	0–1.5
w_C	weight on station coverage m_s	0.5	0.5 or 0.8	0–3
w_I	weight on scouting information I_g	0.45	0.45 or 0.75	0–3
Cost weights				
w_D	travel cost of pursuits and stations	0.15	–	0–0.6
w_R	travel cost of scouting	0.05	–	0–0.3
w_A	belief-age cost, on α_j	0.1	–	0–1
w_X	attrition cost	0.5	0.2, 0.5 or 0.9	0–2
w_L	leaving-station cost	0.25	0.25 or 0.4	0–1.5
w_W	switching cost	0.05	–	0–0.5
w_P	continuity cost	0.15	–	0–1
Set for each intruder				
π_j	priority of intruder j	1	0.5, 1 or 2	–
ρ_j	pursuit multiplier of intruder j	1	–	0–2
Airspace challenges				
N	identifier of a contact, in the label of its photograph			

Supplementary Table 13 Model profiles, calls and cost. Upper part: the two frozen model profiles, called through OpenRouter. The endpoint revision is the endpoint's name as listed when the profile was frozen; the provider tier is the one requested, with fallback to other providers disabled. Every call used a strict JSON response schema whose first field is the reasoning. The near-frontier endpoint takes no temperature, so none was sent. List prices are for prompt / completion tokens. Each call allowed retries after a rate limit (up to five), a connection error, a server error, an empty answer or a provider abort (up to two each) and a timeout (one). Lower part: calls per run, counted from the episode records of the valid episodes, by the interface that made them, and the provider-reported cost written to the records. Calls are synchronous, so their latency does not affect the game; the median latency of a commander call was 2.0 s for the low-cost model and 5.7 s for the near-frontier model on the ideal network and under falsified reports. In the run of the near-frontier model at high authority, the 6 invalid episodes of the first run and their calls are excluded (Methods).

	Low-cost LLM	Near-frontier LLM
Model id	`google/gemini-3.1-flash-lite`	`openai/gpt-6.1-sol`
Endpoint revision	`gemini-3.1-flash-lite-20260507`	`gpt-6.1-sol-20260929`
Provider tier	flex	standard
Reasoning effort	medium	medium
Temperature	1.0	not sent
Output allowance (tokens)	8,000	25,000
Timeout (s)	300	300
List price (US$ per million tokens)	0.125 / 0.75	2 / 10

		Calls by interface				
Run	Episodes	Defenders	Commander	Intruders	All calls	Cost (US$)
Main study (three teams)	900	70,033	10,657	80,098	160,788	104.63
Consensus team at high authority, low-cost LLM	300	39,804	–	29,100	68,904	56.21
Consensus team at high authority, near-frontier LLM	30	3,485	–	3,485	6,970	38.14
Commander team, near-frontier LLM	120	–	4,231	10,919	15,150	26.47
Birds and civilian drones (five teams)	25	1,667	285	2,711	4,663	9.97
Disguised intruders (five teams, without and with forewarning)	50	3,767	730	6,734	11,231	28.17

Supplementary Table 14 Execution identities and re-simulation checks. Each run records an execution identity, a hash over its code version, interfaces, model profile, policy settings and study-version label; a run made in several invocations has one identity per invocation. A recorded episode is valid when it ended without a terminal problem such as an unrecovered call failure or a solver failure; in the run of the near-frontier model at high authority, 6 first-run episodes were invalid after call failures under a provider's rate limit and were run again under a new identity before any outcome was read (Methods). Every response is stored under a hash of its full request. A re-simulation rebuilds an episode from the code and the stored responses, with the API key removed, so a missing response stops the episode instead of calling a model. A re-simulated episode matched when it equalled its record on the breach, the breach step, the breaching intruder, every intruder's outcome, the number of intruders destroyed and the steps run; a match means equality on these recorded fields, not trajectory identity. The episodes with birds and civilian drones, without and with the disguise, were not re-simulated: their measures are read from the records and the stored responses. The disguise ran as one invocation under one execution identity, and the prompt version recorded with every call names the condition, without or with the forewarning.

Run	Execution identities	Episodes recorded	Valid episodes	Re-simulated, matched
Main study (three teams)	`d63bb80c8d4f5a75` `b739142f8b120dfd`	900	900	900 of 900
Consensus team at high authority, low-cost LLM	`1b6b98cdc56e256d`	300	300	300 of 300
Consensus team at high authority, near-frontier LLM	`c707ebe21531bd07` `ca2a2312187c581a`	36	30	30 of 30
Commander team, near-frontier LLM	`f1e1d17fe92b508f` `9e261602168f2bdc`	120	120	120 of 120
Birds and civilian drones (five teams)	`5e4aff75a94d27fc`	25	25	–
Disguised intruders (five teams, without and with forewarning)	`4ddde6e2dcc9d371`	50	50	–

S4 Supplementary Videos

Supplementary Video 1 An episode of the commander team on the ideal network with four intruders.
Seed 4003, chosen by a criterion fixed before rendering: the first seed on which each main-study team's breach outcome equaled its most common outcome in this cell. The episode is illustrative; rates are in Supplementary Table 1. Dots are defenders with their sensor wedges, triangles intruders, solid lines the last 4 s of each track and dashed lines pursuits; a black ring marks the commander, hollow grey triangles are destroyed intruders and gray crosses disabled defenders. The defended region is at the upper right. Drawn from a deterministic re-simulation that matched its record, at four times real time. No breach; all four intruders destroyed. File: `supplementary/Supplementary_Video_1.mp4` in the code repository, https://github.com/the-kas-lab/consensus-within-competition-between.

Supplementary Video 2 An episode of the consensus team on the ideal network with four intruders.
The same seed, criterion and symbols as Supplementary Video 1. No breach; all four intruders destroyed. File: `supplementary/Supplementary_Video_2.mp4`, in the same folder.

Supplementary Video 3 An episode of the independent team on the ideal network with four intruders.
The same seed, criterion and symbols as Supplementary Video 1. No breach; three intruders destroyed, and the episode ended at the 40 s limit with one intruder alive. File: `supplementary/Supplementary_Video_3.mp4`, in the same folder.

S5 Supplementary Note 1 Prompts, response schemas and example calls

Four versioned interfaces build every request: m8b.7 for the defenders of the consensus team at low authority and of the independent team, m8b.9 for the defenders of the consensus team at high authority, commander.10 for the commander and intruder.4 for the intruders. Outside the episodes with birds and civilian drones, each interface sent the same system prompt and response schema in every recorded request, the intruder interface one of each per phase (before reaching a gate and after passing one), and the same text served both models where both were used. In the episodes with birds and civilian drones, the defender and commander prompts carried one added paragraph, the airspace paragraph, reproduced below. A user message states the agent's own picture at the decision epoch. A defender's message gives its own state, its teammates' positions, the intruders it knows of, with the age of each belief, how long each coverage cell has gone unwatched, its current task, the time left and every candidate task with each term of its bid. The commander's message gives every active defender with its last issued assignment, every reported intruder with its forecast, and each defender's optimistic interception time, the task list, and the coverage record. An intruder's message gives its own state and phase, its current action, the gates, its teammates' states and last choices, and the team's shared sightings of defenders. Each box is colored as the team for which it is written is in the main figures: teal for the consensus team at low authority, dark green for it at high authority, indigo for the independent team, ochre for the commander and orange for the intruders; gray marks text shared by several of them. The prompts are reproduced exactly as sent, including their own terms: `BLUE` and `blue` denote the defenders, and `RED` and `red` the intruders. The code renders the numbers in the defender prompts from the constants that the bid computation uses. Calls, decode settings and model profiles are in Supplementary Table 13.

S5.1 System prompts

Defender, consensus team at low authority (interface m8b.7)

```
You are ONE defending agent (BLUE) in a 2D pursuit-evasion simulation. Blue
     defends a region against RED intruders. Your team allocates tasks
    with an auction: each blue bids on its own candidate tasks; the
    highest bid for a task wins it, and a blue that is outbid takes its
    best remaining task. You do NOT choose your task and you do NOT write
    any bid. You choose how YOUR bids are weighted.

Answer with four things, in this order, and nothing numeric beyond contact
    ids:

1. reasoning -- FIRST: one sentence about the actual picture (distances,
    ages, what you are trading).

2. scoring_profile -- exactly one of:
- aggressive: as balanced except attrition 0.5 -> 0.2, urgency time scale
    60 -> 90 s
```

```
- balanced: the reference weights (coverage 0.5, leaving 0.25, information
    0.45, attrition 0.5, urgency time scale 60 s)
- conservative: as balanced except coverage 0.5 -> 0.8, leaving 0.25 ->
    0.4, attrition 0.5 -> 0.9
- scouting: as balanced except information 0.45 -> 0.75

3. threat_priorities -- for contacts YOU believe in (listed below), a
    priority of LOW, MEDIUM or HIGH. MEDIUM is the default for any contact
     you leave out. You may only name contact_id values that are listed
    below.

4. preferred_tasks -- at most 3 task ids, each copied exactly as one of the
     quoted task_id strings in YOUR candidate task list below (the text
    inside the quotes only, without the location). Each receives a fixed
    bonus. Leave it empty if nothing stands out.

Task kinds: NEUTRALIZE:c:s closes to contact c (you are permanently
    disabled on contact). DEFEND:k holds perimeter station k. RECON:r
    patrols toward coverage region r.

What you know is what YOU believe, and it is incomplete. Contacts you
    cannot see are not listed, and a short list does NOT mean the region
    is safe. Other agents are answering at the same moment on their own
    beliefs.

How your answer changes the bids (the highest bid is your task unless the
    auction gives it to a teammate who bids more for it). Bid = value -
    travel - leaving - switching - continuity (NEUTRALIZE also minus
    attrition and belief-age uncertainty), plus 0.2 for a preferred task.
    Your candidate task list shows each task's terms under the balanced
    profile. Every number below is an allocation value that only ranks
    tasks for allocation; none of them is the mission's objective.
- Approach sector: intruders are expected from bearings 160 to 300 degrees
    as seen from the defended centre (any distance). Only ground in that
    sector counts below.
- DEFEND:k value = 0.05 + coverage weight x coverage, where coverage is how
     much approach-sector ground station k's sensor wedge (at the station'
    s fixed heading) adds beyond what your teammates' sensors watch now
    (1.0 = one full sensor's worth).
- RECON:r value = information weight x information: approach-sector ground
    your sensor sweeps on the way to region r and there, counted by how
    stale each region is in YOUR coverage record and by how soon you reach
     it (an optimistic travel forecast from your current velocity); ground
     a blue already watches counts 0. Travel 0.05 per 25 units.
```

```
- NEUTRALIZE:c value = 1.0 x (threat value 1.2 + priority x urgency x
    opportunity). The threat value is the same for every believed red,
    whatever its heading, priority or reachability. Urgency = the larger
    of 1/(1 + forecast time to breach / 60 s) and 1/(1 + distance to the
    region / 25). Forecast time to breach: from the contact's believed
    velocity, the time to close on the approach funnel (the region
    together with the four public approach gates) at its current closing
    speed, plus the last leg into the region; none when it is not closing
    (moving away, crossing or stationary). Opportunity comes from an
    OPTIMISTIC kinematic interception forecast (from your current velocity
    , with the per-axis speed and acceleration limits of 2; it ignores
    collision avoidance and the contact's reaction): 1 if you can reach
    the contact's forecast path before its forecast breach and within the
    episode, falling linearly to 0.25 over 10 s later than that, 0.25 with
     no interception; averaged over +-0.3 u/s of uncertainty in the
    contact's velocity. Travel 0.15 per 25 units of current distance;
    attrition = attrition weight / (1 + teammates), because contact
    disables you.
- Leaving your current station costs leaving weight x that station's
    coverage. Changing task costs 0.05. While you pursue a contact you
    still believe in, every other task (another contact, RECON, DEFEND)
    costs an extra 0.15 (continuity). For ANOTHER CONTACT only, that extra
     cost is multiplied by (1 - R x H x S), down to 0: R = how fast the
    contact you pursue moves away from the approach funnel (0 if it is not
     moving away, 1 at 0.5 u/s or faster); H = 1 if the other contact's
    forecast breach comes at least 5 s before the episode ends, falling to
     0 at the end (0 with no forecast breach); S = how far your optimistic
     interception of the other contact comes before its forecast breach (0
     for no lead, 1 for 2 s or more), averaged over the same velocity
    uncertainty. RECON and DEFEND always pay the full extra cost. Each
    task line shows the continuity actually applied.
- Priorities multiply only urgency x opportunity, never the threat value:
    LOW 0.5, MEDIUM 1, HIGH 2.

Mission (the same for every blue team): your team's objective is to prevent
     any red from entering the defended region before the episode ends.
    The episode lasts 400 steps (dt = 0.1 s, 40 s). Any entry by a red is
    a breach and ends the episode. A blue that touches a red destroys it
    and is permanently disabled. A red that leaves the arena is removed
    and can no longer breach. The episode also ends, without a breach,
    once every red has been destroyed or has left the arena. Reds can
    change direction at any time, and reds your team has not detected may
    exist.

Headings: every agent's sensor points along its heading and sees a 60-
    degree wedge (30 degrees either side) out to 35 units; nothing outside
     it is detected. Angles are in degrees, counter-clockwise from the +x
    axis (0 = +x, 90 = +y, 180 = -x, 270 = -y). Steering and sensing are
    independent: an agent can move one way while looking another. A
    heading listed for an opponent is the one seen at that sighting and is
     NOT updated since (agents can turn up to 360 degrees per second).

Respond only with a valid JSON object matching the required schema, with "
    reasoning" as its first field.
```

The independent team's prompt differs from the consensus team's only in the lines below (unified diff: lines starting with a minus sign are the consensus team's, lines starting with a plus sign the independent team's).

Defender, independent team (interface m8b.7): difference

```
--- consensus, low authority
+++ independent
-You are ONE defending agent (BLUE) in a 2D pursuit-evasion simulation.
    Blue defends a region against RED intruders. Your team allocates tasks
     with an auction: each blue bids on its own candidate tasks; the
    highest bid for a task wins it, and a blue that is outbid takes its
    best remaining task. You do NOT choose your task and you do NOT write
    any bid. You choose how YOUR bids are weighted.
+You are ONE defending agent (BLUE) in a 2D pursuit-evasion simulation.
    Blue defends a region against RED intruders. Each blue computes bids
    for its own candidate tasks and takes its own highest-bid task. There
    is no coordination between blues: teammates may take the same task.
    You do NOT choose your task and you do NOT write any bid. You choose
    how YOUR bids are weighted.
-How your answer changes the bids (the highest bid is your task unless the
    auction gives it to a teammate who bids more for it). Bid = value -
    travel - leaving - switching - continuity (NEUTRALIZE also minus
    attrition and belief-age uncertainty), plus 0.2 for a preferred task.
    Your candidate task list shows each task's terms under the balanced
    profile. Every number below is an allocation value that only ranks
    tasks for allocation; none of them is the mission's objective.
+How your answer changes the bids (your highest bid is your task). Bid =
    value - travel - leaving - switching - continuity (NEUTRALIZE also
    minus attrition and belief-age uncertainty), plus 0.2 for a preferred
    task. Your candidate task list shows each task's terms under the
    balanced profile. Every number below is an allocation value that only
    ranks tasks for allocation; none of them is the mission's objective.
```

Defender, consensus team at high authority, both models (interface m8b.9)

```
You are ONE defending agent (BLUE) in a 2D pursuit-evasion simulation. Blue
     defends a region against RED intruders. Your team allocates tasks
    with an auction: each blue bids on its own candidate tasks; the
    highest bid for a task wins it, and a blue that is outbid takes its
    best remaining task. You do NOT choose your task and you do NOT write
    any bid. You choose the weights YOUR bids use.

Answer with four things, in this order:

1. reasoning -- FIRST: at most two sentences about the actual picture and
    what your weights trade off.

2. weights -- all 12 weights listed below, each a number within its allowed
     range.

3. contact_priorities -- for contacts YOU believe in (listed below), a
    number from 0 to 2. 1 is the default for any contact you leave out.
    You may only name contact_id values that are listed below.

4. preferred_tasks -- at most 3 task ids, each copied exactly as one of the
     quoted task_id strings in YOUR candidate task list below (the text
    inside the quotes only, without the location). Leave it empty if
    nothing stands out.
```

```
Task kinds: NEUTRALIZE:c:s closes to contact c (you are permanently
    disabled on contact). DEFEND:k holds perimeter station k. RECON:r
    patrols toward coverage region r.

What you know is what YOU believe, and it is incomplete. Contacts you
    cannot see are not listed, and a short list does NOT mean the region
    is safe. Other agents are answering at the same moment on their own
    beliefs.

How bids are formed (the highest bid is your task unless the auction gives
    it to a teammate who bids more for it). Every blue computes its bids
    with the same formula from its own beliefs and its own weights. You
    choose the weights YOUR bids use; teammates choose theirs. Every
    number below is an allocation value that only ranks tasks for
    allocation; none of them is the mission's objective.
- DEFEND:k bid = w_station + w_coverage x coverage - w_travel x distance/25
     - penalties
- RECON:r bid = w_scout x information - w_scout_travel x distance/25 -
    penalties
- NEUTRALIZE:c bid = priority(c) x (w_threat + w_engage x urgency x
    opportunity) - w_travel x distance/25 - w_attrition x 1/(1 + teammates
    ) - w_belief_age x age term - penalties
- penalties = w_leave x leaving + w_switch x switching + w_continuity x
    continuity; then plus 0.2 for each of your preferred tasks.
Features (each candidate task line below lists its own):
- Approach sector: intruders are expected from bearings 160 to 300 degrees
    as seen from the defended centre (any distance). Only ground in that
    sector counts below.
- coverage: how much approach-sector ground station k's sensor wedge (at
    the station's fixed heading) adds beyond what your teammates' sensors
    watch now (1.0 = one full sensor's worth).
- information: approach-sector ground your sensor sweeps on the way to
    region r and there, counted by how stale each region is in YOUR
    coverage record and by how soon you reach it (an optimistic travel
    forecast from your current velocity); ground a blue already watches
    counts 0.
- urgency: the larger of 1/(1 + forecast time to breach / 60 s) and 1/(1 +
    distance to the region / 25). Forecast time to breach: from the
    contact's believed velocity, the time to close on the approach funnel
    (the region together with the four public approach gates) at its
    current closing speed, plus the last leg into the region; none when it
     is not closing (moving away, crossing or stationary).
- opportunity: from an OPTIMISTIC kinematic interception forecast (from
    your current velocity, with the per-axis speed and acceleration limits
     of 2; it ignores collision avoidance and the contact's reaction): 1
    if you can reach the contact's forecast path before its forecast
    breach and within the episode, falling linearly to 0.25 over 10 s
    later than that, 0.25 with no interception; averaged over +-0.3 u/s of
     uncertainty in the contact's velocity.
- age term: the belief's age x the reds' top speed / 25.
- leaving: the coverage of the station you would leave (0 if you hold no
    station or the task is that station).
- switching: 1 for every task other than your current task, else 0.
```

```
- continuity: while you pursue a contact you still believe in, 1 for every
    other task (RECON, DEFEND); for ANOTHER CONTACT it is (1 - R x H x S),
     down to 0: R = how fast the contact you pursue moves away from the
    approach funnel (0 if it is not moving away, 1 at 0.5 u/s or faster);
    H = 1 if the other contact's forecast breach comes at least 5 s before
     the episode ends, falling to 0 at the end (0 with no forecast breach)
    ; S = how far your optimistic interception of the other contact comes
    before its forecast breach (0 for no lead, 1 for 2 s or more). 0 when
    you pursue no contact.
Weights (what each multiplies, default, allowed range):
- w_threat: NEUTRALIZE: value every believed red carries, before priority.
    Default 1.2, allowed 0 to 3.
- w_engage: NEUTRALIZE: x urgency x opportunity. Default 1, allowed 0 to 3.
- w_station: DEFEND: value of holding any station. Default 0.05, allowed 0
    to 1.5.
- w_coverage: DEFEND: x the station's coverage. Default 0.5, allowed 0 to
    3.
- w_scout: RECON: x information. Default 0.45, allowed 0 to 3.
- w_attrition: NEUTRALIZE: cost x 1/(1 + teammates), because contact
    disables you. Default 0.5, allowed 0 to 2.
- w_belief_age: NEUTRALIZE: cost x the belief-age term. Default 0.1,
    allowed 0 to 1.
- w_travel: NEUTRALIZE and DEFEND: cost per 25 units of distance. Default
    0.15, allowed 0 to 0.6.
- w_scout_travel: RECON: cost per 25 units of distance. Default 0.05,
    allowed 0 to 0.3.
- w_leave: any task that leaves your current station: cost x that station's
     coverage. Default 0.25, allowed 0 to 1.5.
- w_continuity: any task other than the contact you pursue: cost x the
    continuity factor. Default 0.15, allowed 0 to 1.
- w_switch: any task other than your current task: cost. Default 0.05,
    allowed 0 to 0.5.
- Budget: w_threat + w_engage + w_station + w_coverage + w_scout must not
    exceed 3.2 (the defaults sum to exactly 3.2).
- priority(c): a number from 0 to 2 per contact you believe in, default 1
    for any contact you leave out; the priorities of all the contacts you
    believe in must not sum to more than their number.
A value outside its allowed range is replaced by its default. If the five
    budgeted weights sum to more than the budget, all five are scaled down
     in proportion until they sum to the budget; likewise, if the
    priorities of the contacts you believe in sum to more than their
    number, all of them are scaled down in proportion until they sum to
    their number. So raising one budgeted weight (or one priority) without
     lowering the others lowers the others. A low weight or priority
    lowers a bid; it does not remove the task, and a blue that is outbid
    takes its best remaining task, even one with a negative bid. Your
    candidate task list shows each task's features and its bid under the
    default weights.

Mission (the same for every blue team): your team's objective is to prevent
     any red from entering the defended region before the episode ends.
    The episode lasts 400 steps (dt = 0.1 s, 40 s). Any entry by a red is
    a breach and ends the episode. A blue that touches a red destroys it
    and is permanently disabled. A red that leaves the arena is removed
    and can no longer breach. The episode also ends, without a breach,
    once every red has been destroyed or has left the arena. Reds can
    change direction at any time, and reds your team has not detected may
    exist.
```

```
Headings: every agent's sensor points along its heading and sees a 60-
    degree wedge (30 degrees either side) out to 35 units; nothing outside
     it is detected. Angles are in degrees, counter-clockwise from the +x
    axis (0 = +x, 90 = +y, 180 = -x, 270 = -y). Steering and sensing are
    independent: an agent can move one way while looking another. A
    heading listed for an opponent is the one seen at that sighting and is
     NOT updated since (agents can turn up to 360 degrees per second).

Respond only with a valid JSON object matching the required schema, with "
    reasoning" as its first field.
```

Commander, both models (interface commander.10)

```
You are the commander of the BLUE defender team in a 2D perimeter-defense
    game. Red intruders try to breach the defended region; a blue that
    touches a red destroys it and is itself disabled.
Each planning step you assign every active blue exactly ONE task from the
    task list. Task kinds:
- NEUTRALIZE:<contact>:<slot> -- pursue and intercept that believed red
    contact. Each slot can go to one blue only.
- DEFEND:<station> -- hold that defensive station on the perimeter. One
    blue per station.
- RECON:<cell> -- sweep that arena cell to find reds. One blue per cell.
Every task id may be assigned to at most one blue. Your picture of the reds
     comes from reports your team delivered to you; reds you have not
    heard about may exist. Headings: every agent's sensor points along its
     heading and sees a 60-degree wedge (30 degrees either side) out to 35
     units; nothing outside it is detected. Angles are in degrees, counter
    -clockwise from the +x axis (0 = +x, 90 = +y, 180 = -x, 270 = -y).
    Steering and sensing are independent: an agent can move one way while
    looking another. A heading listed for an opponent is the one seen at
    that sighting and is NOT updated since (agents can turn up to 360
    degrees per second).
Approach sector and sensing: intruders are expected from bearings 160 to
    300 degrees as seen from the defended centre (any distance). A blue
    holding a station looks along that station's fixed heading; the
    approach-sector ground each station's sensor wedge covers is listed
    per station. A blue on RECON flies toward its cell with its sensor
    pointed ahead and holds there. Ground is watched only while some blue'
    s sensor wedge covers it: a scout adds coverage only where no other
    blue is watching at that time, and a station's ground stays watched
    after its blue leaves only if another blue's sensor covers it.
Forecasts shown with the reports (from your picture, labelled as forecasts)
    : each contact's status (approaching / moving away / crossing /
    stationary, from its reported velocity relative to the approach funnel
     = the region together with the four public approach gates), its
    forecast time to breach (closing time to the funnel plus the last leg;
     none when not closing) and urgency = the larger of 1/(1 + forecast
    time / 60 s) and 1/(1 + distance to the region / 25); for each blue
    and contact an OPTIMISTIC kinematic interception forecast: it assumes
    the contact keeps its reported velocity and uses only the blue's speed
     and acceleration limits of 2 per axis from its current velocity (no
    collision avoidance). A contact that reacts can make the real
    interception earlier, later or impossible. Every forecast and coverage
     figure is computed from your picture, which can be stale or
    incomplete.
```

```
Each blue's line shows the LAST ASSIGNMENT YOU ISSUED to it and the step
    you issued it. Issuing is not delivery: with delayed or lost
    communication the blue may not have received or started it. "no
    assignment issued yet" only means you have not assigned that blue
    anything; it does not tell you what the blue is doing.
Mission (the same for every blue team): your team's objective is to prevent
     any red from entering the defended region before the episode ends.
    The episode lasts 400 steps (dt = 0.1 s, 40 s). Any entry by a red is
    a breach and ends the episode. A blue that touches a red destroys it
    and is permanently disabled. A red that leaves the arena is removed
    and can no longer breach. The episode also ends, without a breach,
    once every red has been destroyed or has left the arena. Reds can
    change direction at any time, and reds your team has not detected may
    exist.
Valid answers:
- Give every active blue exactly one task, and give each full task id to at
     most one blue. Different slots of one contact (NEUTRALIZE:c:0,
    NEUTRALIZE:c:1) are different tasks; the same task id given to two
    blues is invalid and the second blue is sent back to its station.
Answer with JSON matching the schema: task_by_agent maps every active blue'
    s id to its one task id. Give your reasoning first.
```

Intruder, before reaching a gate (interface intruder.4)

```
You are ONE red intruder drone in a 2D perimeter-defense game. Your team
    tries to breach a defended region; blue defenders try to intercept you
    , and a blue that touches you destroys you (and itself). Your team's
    objective is for at least one red to enter the defended region before
    the episode ends. The episode lasts 400 steps (dt = 0.1 s, 40 s). Any
    entry by a red is a breach: your team succeeds and the episode ends. A
     red that leaves the arena is removed and can no longer breach. The
    episode also ends, without a breach, once every red has been destroyed
     or has left the arena. You decide only for yourself; each red
    teammate decides for itself.
You know: your own state, your teammates' states, your team's shared
    sightings of blue defenders (each with when it was seen and its
    predicted position now; blues you have not seen may exist), and what
    each teammate chose at the previous planning step.
Actions:
- evade: move away from ONE blue you currently believe in (target_id).
    Evading does not change your phase.
- attack: fly to ONE of the approach gates (target_id) -- public waypoints
    outside the defended region. On reaching your chosen gate you switch
    permanently to the interior attack. You may change gate at any
    planning step before you reach one.
Headings: every agent's sensor points along its heading and sees a 60-
    degree wedge (30 degrees either side) out to 35 units; nothing outside
     it is detected. Angles are in degrees, counter-clockwise from the +x
    axis (0 = +x, 90 = +y, 180 = -x, 270 = -y). Steering and sensing are
    independent: an agent can move one way while looking another. A
    heading listed for an opponent is the one seen at that sighting and is
     NOT updated since (agents can turn up to 360 degrees per second).
target_id is the gate_id you attack, or the blue id you evade. Answer with
    JSON matching the schema; give your reasoning first.
```

Intruder, after passing a gate (interface intruder.4)

```
You are ONE red intruder drone in a 2D perimeter-defense game. Your team
    tries to breach a defended region; blue defenders try to intercept you
    , and a blue that touches you destroys you (and itself). Your team's
    objective is for at least one red to enter the defended region before
    the episode ends. The episode lasts 400 steps (dt = 0.1 s, 40 s). Any
    entry by a red is a breach: your team succeeds and the episode ends. A
     red that leaves the arena is removed and can no longer breach. The
    episode also ends, without a breach, once every red has been destroyed
     or has left the arena. You decide only for yourself; each red
    teammate decides for itself.
You know: your own state, your teammates' states, your team's shared
    sightings of blue defenders (each with when it was seen and its
    predicted position now; blues you have not seen may exist), and what
    each teammate chose at the previous planning step.
Actions:
- evade: move away from ONE blue you currently believe in (target_id).
    Evading does not change your phase.
- attack: you have passed your gate and are in the INTERIOR phase: attack
    means advance on the defended region's centre. There are no gates to
    choose any more.
Headings: every agent's sensor points along its heading and sees a 60-
    degree wedge (30 degrees either side) out to 35 units; nothing outside
     it is detected. Angles are in degrees, counter-clockwise from the +x
    axis (0 = +x, 90 = +y, 180 = -x, 270 = -y). Steering and sensing are
    independent: an agent can move one way while looking another. A
    heading listed for an opponent is the one seen at that sighting and is
     NOT updated since (agents can turn up to 360 degrees per second).
target_id is the blue id you evade, or -1 for attack. Answer with JSON
    matching the schema; give your reasoning first.
```

S5.2 The airspace paragraph

In the episodes with birds and civilian drones, with and without the disguise, this paragraph was inserted into every defender and commander system prompt after the brief's last sentence, and each listed contact's photograph followed the user message as an image labelled "image of contact_id N". The intruders' prompts were unchanged.

The airspace paragraph

```
Not every contact is a red. Besides the red intruders, the airspace holds
    birds and civilian drones. They move in the same airspace, cannot
    breach, and are not part of the objective. A blue that touches any
    contact, red, bird or civilian drone, destroys it and is permanently
    disabled. The episode ends without a breach once every red has been
    destroyed or has left the arena, whatever the birds and civilian
    drones do. A contact's id does not tell what it is. One image of each
    contact listed below is attached after this message, labelled with its
     contact_id.
```

In the disguise runs with the forewarning, one sentence followed "A contact's id does not tell what it is." and nothing else in any prompt differed:

The forewarning sentence

```
Some red intruders may be disguised as birds.
```

S5.3 Response schemas

Each interface used one strict JSON schema in every request; the reasoning field comes first.

Defender, consensus team at low authority and independent team: response schema

```
{
 "type": "json_schema",
 "json_schema": {
  "name": "bid_modifiers",
  "strict": true,
  "schema": {
   "type": "object",
   "properties": {
    "reasoning": {
     "type": "string"
    },
    "scoring_profile": {
     "type": "string",
     "enum": [
      "aggressive",
      "balanced",
      "conservative",
      "scouting"
     ]
    },
    "threat_priorities": {
     "type": "array",
     "items": {
      "type": "object",
      "properties": {
       "contact_id": {
        "type": "integer"
       },
       "priority": {
        "type": "string",
        "enum": [
         "HIGH",
         "LOW",
         "MEDIUM"
        ]
       }
      },
      "required": [
       "contact_id",
       "priority"
      ],
      "additionalProperties": false
     }
    },
    "preferred_tasks": {
     "type": "array",
```

```
     "items": {
      "type": "string"
     }
    }
   },
   "required": [
    "reasoning",
    "scoring_profile",
    "threat_priorities",
    "preferred_tasks"
   ],
   "additionalProperties": false
  }
 }
}
```

Defender, consensus team at high authority: response schema

```
{
 "type": "json_schema",
 "json_schema": {
  "name": "strategy_weights",
  "strict": true,
  "schema": {
   "type": "object",
   "properties": {
    "reasoning": {
     "type": "string"
    },
    "weights": {
     "type": "object",
     "properties": {
      "w_threat": {
       "type": "number"
      },
      "w_engage": {
       "type": "number"
      },
      "w_station": {
       "type": "number"
      },
      "w_coverage": {
       "type": "number"
      },
      "w_scout": {
       "type": "number"
      },
      "w_attrition": {
       "type": "number"
      },
      "w_belief_age": {
       "type": "number"
      },
      "w_travel": {
       "type": "number"
      },
      "w_scout_travel": {
```

```
      "type": "number"
     },
     "w_leave": {
      "type": "number"
     },
     "w_continuity": {
      "type": "number"
     },
     "w_switch": {
      "type": "number"
     }
    },
    "required": [
     "w_threat",
     "w_engage",
     "w_station",
     "w_coverage",
     "w_scout",
     "w_attrition",
     "w_belief_age",
     "w_travel",
     "w_scout_travel",
     "w_leave",
     "w_continuity",
     "w_switch"
    ],
    "additionalProperties": false
   },
   "contact_priorities": {
    "type": "array",
    "items": {
     "type": "object",
     "properties": {
      "contact_id": {
       "type": "integer"
      },
      "priority": {
       "type": "number"
      }
     },
     "required": [
      "contact_id",
      "priority"
     ],
     "additionalProperties": false
    }
   },
   "preferred_tasks": {
    "type": "array",
    "items": {
     "type": "string"
    }
   }
  },
  "required": [
   "reasoning",
   "weights",
   "contact_priorities",
   "preferred_tasks"
```

```
   ],
   "additionalProperties": false
  }
 }
}
```

Commander: response schema

```
{
 "type": "json_schema",
 "json_schema": {
  "name": "commander_assignment",
  "strict": true,
  "schema": {
   "type": "object",
   "properties": {
    "reasoning": {
     "type": "string"
    },
    "task_by_agent": {
     "type": "object",
     "properties": {
      "<agent id>": {
       "type": "string",
       "enum": [
        "<task ids of this epoch>"
       ]
      }
     },
     "required": [
      "<every active agent id>"
     ],
     "additionalProperties": false
    }
   },
   "required": [
    "reasoning",
    "task_by_agent"
   ],
   "additionalProperties": false
  }
 }
}
```

Intruder, before reaching a gate: response schema

```
{
 "type": "json_schema",
 "json_schema": {
  "name": "intruder_decision",
  "strict": true,
  "schema": {
   "type": "object",
   "properties": {
    "reasoning": {
```

```
     "type": "string"
    },
    "action": {
     "type": "string",
     "enum": [
      "attack",
      "evade"
     ]
    },
    "target_id": {
     "type": "integer",
     "description": "the gate_id to attack, or the blue id to evade"
    }
   },
   "required": [
    "reasoning",
    "action",
    "target_id"
   ],
   "additionalProperties": false
  }
 }
}
```

Intruder, after passing a gate: response schema

```
{
 "type": "json_schema",
 "json_schema": {
  "name": "intruder_decision",
  "strict": true,
  "schema": {
   "type": "object",
   "properties": {
    "reasoning": {
     "type": "string"
    },
    "action": {
     "type": "string",
     "enum": [
      "attack",
      "evade"
     ]
    },
    "target_id": {
     "type": "integer",
     "description": "the blue id to evade, or -1 for attack"
    }
   },
   "required": [
    "reasoning",
    "action",
    "target_id"
   ],
   "additionalProperties": false
  }
 }
```

```
}
```

S5.4 Example requests and responses

For each interface, the first recorded call with a listed contact in the first episode in sorted order, as sent and as answered. Each request also carried the interface's system prompt and schema above.

Defender, consensus team at low authority: user message

```
sim_step: 110
arena_size: 100.00
defense_center: [82.00, 82.00]
you are blue agent_id=0
  your_position: [92.85, 81.98]
  your_velocity: [-0.00, -0.00]
  your_heading_deg: 255
active_teammates (positions known exactly; intentions unknown, no
    communication this cycle):
  agent_id=1 pos=[74.46, 81.84] vel=[-0.00, 0.03] heading_deg=185
  agent_id=2 pos=[50.17, 70.73] vel=[-2.00, -1.99] heading_deg=253
  agent_id=3 pos=[81.99, 73.65] vel=[-0.01, 0.21] heading_deg=230
contacts you believe in (YOUR belief only -- others may exist that you
    cannot see; age_steps is how old the underlying observation is):
  contact_id=0 believed_position=[41.38, 39.10] believed_velocity=[2.00,
    2.00] observed_heading_deg=74 age_steps=1
coverage (steps since your knowledge of each region was refreshed; -1 =
    never):
  region_id=0 center=[12.50, 12.50] steps_since_seen=-1
  region_id=1 center=[12.50, 37.50] steps_since_seen=-1
  region_id=2 center=[12.50, 62.50] steps_since_seen=-1
  region_id=3 center=[12.50, 87.50] steps_since_seen=-1
  region_id=4 center=[37.50, 12.50] steps_since_seen=-1
  region_id=5 center=[37.50, 37.50] steps_since_seen=-1
  region_id=6 center=[37.50, 62.50] steps_since_seen=4
  region_id=7 center=[37.50, 87.50] steps_since_seen=94
  region_id=8 center=[62.50, 12.50] steps_since_seen=-1
  region_id=9 center=[62.50, 37.50] steps_since_seen=45
  region_id=10 center=[62.50, 62.50] steps_since_seen=2
  region_id=11 center=[62.50, 87.50] steps_since_seen=8
  region_id=12 center=[87.50, 12.50] steps_since_seen=-1
  region_id=13 center=[87.50, 37.50] steps_since_seen=94
  region_id=14 center=[87.50, 62.50] steps_since_seen=0
  region_id=15 center=[87.50, 87.50] steps_since_seen=90
your current task: DEFEND:0
episode time left: 29.0 s
your candidate tasks (task_id, where it is, terms under the balanced
    profile):
  task_id "NEUTRALIZE:0:0"    at [41.38, 39.10]    value 1.98 (threat 1.20
    + urgency 0.78 x opportunity 1.00; approaching, forecast breach 17.1 s
    , interception forecast 13.1 s), travel 0.40, leaving 0.15, switch
    0.05, attrition 0.12, uncertainty 0.00 -> bid 1.26
  task_id "DEFEND:0"    at [92.86, 82.00]    value 0.34 (coverage 0.58),
    travel 0.00 -> bid 0.34
  task_id "DEFEND:1"    at [82.00, 91.94]    value 0.18 (coverage 0.25),
    travel 0.09, leaving 0.15, switch 0.05 -> bid -0.11
```

```
  task_id "DEFEND:2"    at [74.46, 82.00]    value 0.05 (coverage 0.00),
    travel 0.11, leaving 0.15, switch 0.05 -> bid -0.25
  task_id "DEFEND:3"    at [82.00, 73.54]    value 0.05 (coverage 0.01),
    travel 0.08, leaving 0.15, switch 0.05 -> bid -0.23
  task_id "RECON:0"    at [12.50, 12.50]    value 0.25 (information 0.55),
    travel 0.21, leaving 0.15, switch 0.05 -> bid -0.16
  task_id "RECON:1"    at [12.50, 37.50]    value 0.17 (information 0.37),
    travel 0.18, leaving 0.15, switch 0.05 -> bid -0.21
  task_id "RECON:2"    at [12.50, 62.50]    value 0.18 (information 0.40),
    travel 0.17, leaving 0.15, switch 0.05 -> bid -0.18
  task_id "RECON:3"    at [12.50, 87.50]    value 0.21 (information 0.46),
    travel 0.16, leaving 0.15, switch 0.05 -> bid -0.15
  task_id "RECON:4"    at [37.50, 12.50]    value 0.28 (information 0.63),
    travel 0.18, leaving 0.15, switch 0.05 -> bid -0.09
  task_id "RECON:5"    at [37.50, 37.50]    value 0.22 (information 0.49),
    travel 0.14, leaving 0.15, switch 0.05 -> bid -0.12
  task_id "RECON:6"    at [37.50, 62.50]    value 0.15 (information 0.33),
    travel 0.12, leaving 0.15, switch 0.05 -> bid -0.16
  task_id "RECON:7"    at [37.50, 87.50]    value 0.19 (information 0.42),
    travel 0.11, leaving 0.15, switch 0.05 -> bid -0.12
  task_id "RECON:8"    at [62.50, 12.50]    value 0.32 (information 0.72),
    travel 0.15, leaving 0.15, switch 0.05 -> bid -0.02
  task_id "RECON:9"    at [62.50, 37.50]    value 0.25 (information 0.55),
    travel 0.11, leaving 0.15, switch 0.05 -> bid -0.06
  task_id "RECON:10"    at [62.50, 62.50]    value 0.02 (information 0.04),
     travel 0.07, leaving 0.15, switch 0.05 -> bid -0.25
  task_id "RECON:11"    at [62.50, 87.50]    value 0.05 (information 0.11),
     travel 0.06, leaving 0.15, switch 0.05 -> bid -0.21
  task_id "RECON:12"    at [87.50, 12.50]    value 0.33 (information 0.74),
     travel 0.14, leaving 0.15, switch 0.05 -> bid -0.00
  task_id "RECON:13"    at [87.50, 37.50]    value 0.33 (information 0.72),
     travel 0.09, leaving 0.15, switch 0.05 -> bid 0.04
  task_id "RECON:14"    at [87.50, 62.50]    value 0.12 (information 0.26),
     travel 0.04, leaving 0.15, switch 0.05 -> bid -0.12
  task_id "RECON:15"    at [87.50, 87.50]    value 0.00 (information 0.00),
     travel 0.02, leaving 0.15, switch 0.05 -> bid -0.21
```

Defender, consensus team at low authority: response

```
{
  "reasoning": "Contact 0 is moving steadily into the defended region and
    presents a high-priority threat that justifies immediate interception
    .",
  "scoring_profile": "balanced",
  "threat_priorities": [
    {
      "contact_id": 0,
      "priority": "HIGH"
    }
  ],
  "preferred_tasks": [
    "NEUTRALIZE:0:0"
  ]
}
```

Defender, independent team: user message

```
sim_step: 200
arena_size: 100.00
defense_center: [82.00, 82.00]
you are blue agent_id=0
  your_position: [72.54, 61.71]
  your_velocity: [-2.00, -2.00]
  your_heading_deg: 196
active_teammates (positions known exactly; intentions unknown, no
    communication this cycle):
  agent_id=1 pos=[55.33, 61.76] vel=[-1.28, -1.70] heading_deg=252
  agent_id=2 pos=[53.36, 54.47] vel=[-0.40, -0.23] heading_deg=80
contacts you believe in (YOUR belief only -- others may exist that you
    cannot see; age_steps is how old the underlying observation is):
  contact_id=0 believed_position=[54.26, 57.09] believed_velocity=[1.60,
    2.00] observed_heading_deg=254 age_steps=0
coverage (steps since your knowledge of each region was refreshed; -1 =
    never):
  region_id=0 center=[12.50, 12.50] steps_since_seen=-1
  region_id=1 center=[12.50, 37.50] steps_since_seen=-1
  region_id=2 center=[12.50, 62.50] steps_since_seen=-1
  region_id=3 center=[12.50, 87.50] steps_since_seen=-1
  region_id=4 center=[37.50, 12.50] steps_since_seen=47
  region_id=5 center=[37.50, 37.50] steps_since_seen=1
  region_id=6 center=[37.50, 62.50] steps_since_seen=7
  region_id=7 center=[37.50, 87.50] steps_since_seen=197
  region_id=8 center=[62.50, 12.50] steps_since_seen=-1
  region_id=9 center=[62.50, 37.50] steps_since_seen=17
  region_id=10 center=[62.50, 62.50] steps_since_seen=0
  region_id=11 center=[62.50, 87.50] steps_since_seen=1
  region_id=12 center=[87.50, 12.50] steps_since_seen=-1
  region_id=13 center=[87.50, 37.50] steps_since_seen=186
  region_id=14 center=[87.50, 62.50] steps_since_seen=90
  region_id=15 center=[87.50, 87.50] steps_since_seen=183
your current task: NEUTRALIZE:0:0
episode time left: 20.0 s
your candidate tasks (task_id, where it is, terms under the balanced
    profile):
  task_id "NEUTRALIZE:0:0"    at [54.26, 57.09]    value 2.05 (threat 1.20
    + urgency 0.85 x opportunity 1.00; approaching, forecast breach 11.0 s
    , interception forecast 5.0 s), travel 0.11, attrition 0.17 -> bid
    1.77
  task_id "DEFEND:0"    at [92.86, 82.00]    value 0.43 (coverage 0.77),
    travel 0.17, switch 0.05, continuity 0.15 -> bid 0.06
  task_id "DEFEND:1"    at [82.00, 91.94]    value 0.15 (coverage 0.21),
    travel 0.19, switch 0.05, continuity 0.15 -> bid -0.24
  task_id "DEFEND:2"    at [74.46, 82.00]    value 0.21 (coverage 0.32),
    travel 0.12, switch 0.05, continuity 0.15 -> bid -0.11
  task_id "DEFEND:3"    at [82.00, 73.54]    value 0.49 (coverage 0.87),
    travel 0.09, switch 0.05, continuity 0.15 -> bid 0.19
  task_id "RECON:0"    at [12.50, 12.50]    value 0.24 (information 0.53),
    travel 0.16, switch 0.05, continuity 0.15 -> bid -0.12
  task_id "RECON:1"    at [12.50, 37.50]    value 0.27 (information 0.59),
    travel 0.13, switch 0.05, continuity 0.15 -> bid -0.06
  task_id "RECON:2"    at [12.50, 62.50]    value 0.33 (information 0.74),
    travel 0.12, switch 0.05, continuity 0.15 -> bid 0.01
  task_id "RECON:3"    at [12.50, 87.50]    value 0.45 (information 0.99),
    travel 0.13, switch 0.05, continuity 0.15 -> bid 0.12
```

```
  task_id "RECON:4"    at [37.50, 12.50]    value 0.20 (information 0.44),
    travel 0.12, switch 0.05, continuity 0.15 -> bid -0.12
  task_id "RECON:5"    at [37.50, 37.50]    value 0.21 (information 0.46),
    travel 0.09, switch 0.05, continuity 0.15 -> bid -0.08
  task_id "RECON:6"    at [37.50, 62.50]    value 0.31 (information 0.68),
    travel 0.07, switch 0.05, continuity 0.15 -> bid 0.04
  task_id "RECON:7"    at [37.50, 87.50]    value 0.35 (information 0.78),
    travel 0.09, switch 0.05, continuity 0.15 -> bid 0.06
  task_id "RECON:8"    at [62.50, 12.50]    value 0.45 (information 1.00),
    travel 0.10, switch 0.05, continuity 0.15 -> bid 0.15
  task_id "RECON:9"    at [62.50, 37.50]    value 0.27 (information 0.59),
    travel 0.05, switch 0.05, continuity 0.15 -> bid 0.01
  task_id "RECON:10"    at [62.50, 62.50]    value 0.06 (information 0.13),
     travel 0.02, switch 0.05, continuity 0.15 -> bid -0.16
  task_id "RECON:11"    at [62.50, 87.50]    value 0.03 (information 0.06),
     travel 0.06, switch 0.05, continuity 0.15 -> bid -0.23
  task_id "RECON:12"    at [87.50, 12.50]    value 0.68 (information 1.50),
     travel 0.10, switch 0.05, continuity 0.15 -> bid 0.37
  task_id "RECON:13"    at [87.50, 37.50]    value 0.59 (information 1.32),
     travel 0.06, switch 0.05, continuity 0.15 -> bid 0.34
  task_id "RECON:14"    at [87.50, 62.50]    value 0.08 (information 0.18),
     travel 0.03, switch 0.05, continuity 0.15 -> bid -0.15
  task_id "RECON:15"    at [87.50, 87.50]    value 0.03 (information 0.06),
     travel 0.06, switch 0.05, continuity 0.15 -> bid -0.23
```

Defender, independent team: response

```
{
  "reasoning": "Contact 0 is moving towards the defended region, and
    neutralizing it is the highest priority to prevent a breach before the
     episode ends.",
  "scoring_profile": "balanced",
  "threat_priorities": [
    {
      "contact_id": 0,
      "priority": "HIGH"
    }
  ],
  "preferred_tasks": [
    "NEUTRALIZE:0:0"
  ]
}
```

Commander, low-cost LLM: user message

```
time_step: 210 (dt=0.1 s), episode time left 19.0 s
blues (active):
  blue 0 pos=[92.8, 82.0] heading_deg=255 last issued assignment DEFEND:0 (
    issued at step 200)
  blue 1 pos=[82.0, 91.9] heading_deg=220 last issued assignment DEFEND:1 (
    issued at step 200)
  blue 2 pos=[70.8, 77.1] heading_deg=268 last issued assignment NEUTRALIZE
    :1:0 (issued at step 200)
  blue 3 pos=[77.1, 68.7] heading_deg=212 last issued assignment NEUTRALIZE
    :0:0 (issued at step 200)
```

```
red_contacts (as reported to you):
  contact 0 predicted pos=[61.3, 59.1] vel=[1.99, 2.00]
    observed_heading_deg=32 seen 1 steps ago
    forecast: approaching, breach 7.1 s, urgency 0.89; optimistic
    interception: blue 0 8.1 s, blue 1 8.4 s, blue 2 4.5 s, blue 3 3.9 s
  contact 1 predicted pos=[70.1, 50.4] vel=[2.00, 2.00]
    observed_heading_deg=67 seen 2 steps ago
    forecast: approaching, breach 9.1 s, urgency 0.87; optimistic
    interception: blue 0 8.1 s, blue 1 10.6 s, blue 2 6.6 s, blue 3 4.5 s
tasks:
  NEUTRALIZE:0:0 at [61.3, 59.1]
  NEUTRALIZE:1:0 at [70.1, 50.4]
  DEFEND:0 at [92.9, 82.0] sensor heading_deg=255 approach coverage 0.77
    (0.58 not covered by any other station)
  DEFEND:1 at [82.0, 91.9] sensor heading_deg=220 approach coverage 0.83
    (0.26 not covered by any other station)
  DEFEND:2 at [74.5, 82.0] sensor heading_deg=185 approach coverage 1.00
    (0.55 not covered by any other station)
  DEFEND:3 at [82.0, 73.5] sensor heading_deg=230 approach coverage 1.00
    (0.69 not covered by any other station)
  RECON:0 at [12.5, 12.5]
  RECON:1 at [12.5, 37.5]
  RECON:2 at [12.5, 62.5]
  RECON:3 at [12.5, 87.5]
  RECON:4 at [37.5, 12.5]
  RECON:5 at [37.5, 37.5]
  RECON:6 at [37.5, 62.5]
  RECON:7 at [37.5, 87.5]
  RECON:8 at [62.5, 12.5]
  RECON:9 at [62.5, 37.5]
  RECON:10 at [62.5, 62.5]
  RECON:11 at [62.5, 87.5]
  RECON:12 at [87.5, 12.5]
  RECON:13 at [87.5, 37.5]
  RECON:14 at [87.5, 62.5]
  RECON:15 at [87.5, 87.5]
coverage (YOUR record: steps since each region was last seen, -1 = never;
    share = fraction of the region in the approach sector):
  region 0 center=[12.5, 12.5] steps_since_seen=-1 share=1.00
  region 1 center=[12.5, 37.5] steps_since_seen=-1 share=1.00
  region 2 center=[12.5, 62.5] steps_since_seen=-1 share=1.00
  region 3 center=[12.5, 87.5] steps_since_seen=-1 share=1.00
  region 4 center=[37.5, 12.5] steps_since_seen=-1 share=1.00
  region 5 center=[37.5, 37.5] steps_since_seen=-1 share=1.00
  region 6 center=[37.5, 62.5] steps_since_seen=-1 share=1.00
  region 7 center=[37.5, 87.5] steps_since_seen=177 share=0.91
  region 8 center=[62.5, 12.5] steps_since_seen=-1 share=1.00
  region 9 center=[62.5, 37.5] steps_since_seen=198 share=1.00
  region 10 center=[62.5, 62.5] steps_since_seen=1 share=1.00
  region 11 center=[62.5, 87.5] steps_since_seen=1 share=0.55
  region 12 center=[87.5, 12.5] steps_since_seen=-1 share=1.00
  region 13 center=[87.5, 37.5] steps_since_seen=-1 share=1.00
  region 14 center=[87.5, 62.5] steps_since_seen=0 share=0.70
  region 15 center=[87.5, 87.5] steps_since_seen=-1 share=0.00
```

Commander, low-cost LLM: response


```
{"reasoning": "Blue 3 is optimally positioned to continue intercepting Red
    0 (predicted breach in 7.1s, Blue 3 intercept time 3.9s). Blue 2 is
    optimally positioned to continue intercepting Red 1 (predicted breach
    in 9.1s, Blue 2 intercept time 6.6s). Blue 0 and 1 remain at their
    assigned defensive stations to cover the perimeter and watch the
    approach sector.", "task_by_agent": {"0": "DEFEND:0", "1": "DEFEND:1",
     "2": "NEUTRALIZE:1:0", "3": "NEUTRALIZE:0:0"}}
```


Commander, near-frontier LLM: user message

```
time_step: 180 (dt=0.1 s), episode time left 22.0 s
blues (active):
  blue 0 pos=[92.8, 82.0] heading_deg=255 last issued assignment DEFEND:0 (
    issued at step 170)
  blue 1 pos=[55.2, 64.9] heading_deg=270 last issued assignment NEUTRALIZE
    :0:0 (issued at step 170)
  blue 2 pos=[74.5, 82.0] heading_deg=185 last issued assignment DEFEND:2 (
    issued at step 170)
  blue 3 pos=[77.1, 68.6] heading_deg=241 last issued assignment NEUTRALIZE
    :1:0 (issued at step 170)
red_contacts (as reported to you):
  contact 0 predicted pos=[55.4, 51.1] vel=[2.00, 2.00]
    observed_heading_deg=91 seen 1 steps ago
    forecast: approaching, breach 10.5 s, urgency 0.85; optimistic
    interception: blue 0 9.5 s, blue 1 3.4 s, blue 2 7.9 s, blue 3 5.4 s
  contact 1 predicted pos=[64.1, 44.4] vel=[2.00, 2.00]
    observed_heading_deg=61 seen 1 steps ago
    forecast: approaching, breach 11.4 s, urgency 0.84; optimistic
    interception: blue 0 9.6 s, blue 1 none, blue 2 9.6 s, blue 3 6.0 s
tasks:
  NEUTRALIZE:0:0 at [55.4, 51.1]
  NEUTRALIZE:1:0 at [64.1, 44.4]
  DEFEND:0 at [92.9, 82.0] sensor heading_deg=255 approach coverage 0.77
    (0.58 not covered by any other station)
  DEFEND:1 at [82.0, 91.9] sensor heading_deg=220 approach coverage 0.83
    (0.26 not covered by any other station)
  DEFEND:2 at [74.5, 82.0] sensor heading_deg=185 approach coverage 1.00
    (0.55 not covered by any other station)
  DEFEND:3 at [82.0, 73.5] sensor heading_deg=230 approach coverage 1.00
    (0.69 not covered by any other station)
  RECON:0 at [12.5, 12.5]
  RECON:1 at [12.5, 37.5]
  RECON:2 at [12.5, 62.5]
  RECON:3 at [12.5, 87.5]
  RECON:4 at [37.5, 12.5]
  RECON:5 at [37.5, 37.5]
  RECON:6 at [37.5, 62.5]
  RECON:7 at [37.5, 87.5]
  RECON:8 at [62.5, 12.5]
  RECON:9 at [62.5, 37.5]
  RECON:10 at [62.5, 62.5]
  RECON:11 at [62.5, 87.5]
  RECON:12 at [87.5, 12.5]
  RECON:13 at [87.5, 37.5]
  RECON:14 at [87.5, 62.5]
  RECON:15 at [87.5, 87.5]
```

```
coverage (YOUR record: steps since each region was last seen, -1 = never;
    share = fraction of the region in the approach sector):
  region 0 center=[12.5, 12.5] steps_since_seen=-1 share=1.00
  region 1 center=[12.5, 37.5] steps_since_seen=-1 share=1.00
  region 2 center=[12.5, 62.5] steps_since_seen=-1 share=1.00
  region 3 center=[12.5, 87.5] steps_since_seen=-1 share=1.00
  region 4 center=[37.5, 12.5] steps_since_seen=-1 share=1.00
  region 5 center=[37.5, 37.5] steps_since_seen=6 share=1.00
  region 6 center=[37.5, 62.5] steps_since_seen=50 share=1.00
  region 7 center=[37.5, 87.5] steps_since_seen=147 share=0.91
  region 8 center=[62.5, 12.5] steps_since_seen=-1 share=1.00
  region 9 center=[62.5, 37.5] steps_since_seen=1 share=1.00
  region 10 center=[62.5, 62.5] steps_since_seen=15 share=1.00
  region 11 center=[62.5, 87.5] steps_since_seen=1 share=0.55
  region 12 center=[87.5, 12.5] steps_since_seen=-1 share=1.00
  region 13 center=[87.5, 37.5] steps_since_seen=-1 share=1.00
  region 14 center=[87.5, 62.5] steps_since_seen=0 share=0.70
  region 15 center=[87.5, 87.5] steps_since_seen=-1 share=0.00
```

Commander, near-frontier LLM: response

```
{
  "reasoning": "Keep blue 1 intercepting contact 0 and blue 3 intercepting
    contact 1: their optimistic interception times of 3.4 s and 6.0 s
    precede the respective breach forecasts of 10.5 s and 11.4 s. These
    fresh reports support continuing pursuit, while recognizing that red
    maneuvers could change the forecasts. Keep blues 0 and 2 at their
    current stations to preserve complementary perimeter coverage against
    undetected intruders and avoid unnecessary repositioning.",
  "task_by_agent": {
    "0": "DEFEND:0",
    "1": "NEUTRALIZE:0:0",
    "2": "DEFEND:2",
    "3": "NEUTRALIZE:1:0"
  }
}
```

Intruder, before reaching a gate: user message

```
time_step: 50 (dt=0.1 s)
you: red 0 pos=[29.4, 27.1] vel=[1.99, 2.00] heading_deg=45
phase: approach
defended_region_centre: [82.0, 82.0]
your_current_action: attack gate 2
gates:
  gate 0 at [69.4, 82.0] distance 67.9
  gate 1 at [70.2, 75.2] distance 63.0
  gate 2 at [74.0, 68.1] distance 60.6
  gate 3 at [82.0, 68.6] distance 67.0
teammates:
  red 1 pos=[38.1, 18.4] vel=[2.00, 2.00] heading_deg=45; chose attack gate
     2 at step 40
blue_sightings (team-shared; others may exist unseen): none
```

Intruder, before reaching a gate: response

```
{
  "reasoning": "There are no blue defenders detected, so I have no
    immediate need to evade. My current heading of 45 degrees is well-
    aligned for reaching Gate 2 at [74.0, 68.1]. I will continue my
    approach to gate 2 as it is a direct path to the defended region.",
  "action": "attack",
  "target_id": 2
}
```

Intruder, after passing a gate: user message

```
time_step: 280 (dt=0.1 s)
you: red 1 pos=[82.0, 69.5] vel=[-0.06, 1.99] heading_deg=196
phase: interior
defended_region_centre: [82.0, 82.0]
your_current_action: attack
teammates:
  red 0 pos=[53.6, 59.2] vel=[2.00, 2.00] heading_deg=244; chose evade blue
     2 at step 270
blue_sightings (team-shared; others may exist unseen):
  blue 0 predicted pos=[80.1, 69.3] vel=[-1.60, -1.60] observed_heading_deg
    =227 seen 73 steps ago
  blue 1 predicted pos=[74.5, 82.0] vel=[-0.00, -0.00] observed_heading_deg
    =185 seen 51 steps ago
  blue 2 predicted pos=[49.5, 51.1] vel=[-1.14, -0.60] observed_heading_deg
    =63 seen 0 steps ago
  blue 3 predicted pos=[64.9, 64.4] vel=[-2.00, -0.71] observed_heading_deg
    =205 seen 0 steps ago
```

Intruder, after passing a gate: response

```
{
  "reasoning": "I am in the interior phase and very close to the defended
    region center ([82.0, 82.0]). My current position is [82.0, 69.5],
    meaning I am directly south of the target. While blue 1 is nearby at y
    =82.0, it is offset by 7.5 units on the x-axis. Attacking directly is
    the optimal path to achieve the objective before the time limit, as I
    am already in the interior phase and the target is within reach.",
  "action": "attack",
  "target_id": -1
}
```

Defender, consensus team at high authority, low-cost LLM: user message

```
sim_step: 180
arena_size: 100.00
defense_center: [82.00, 82.00]
you are blue agent_id=0
  your_position: [92.84, 81.99]
  your_velocity: [-0.00, -0.00]
  your_heading_deg: 255
```

```
active_teammates (positions known exactly; intentions unknown, no
    communication this cycle):
  agent_id=1 pos=[81.96, 91.91] vel=[0.31, 0.29] heading_deg=220
  agent_id=2 pos=[74.45, 81.99] vel=[0.00, -0.00] heading_deg=185
  agent_id=3 pos=[81.99, 73.53] vel=[-0.00, -0.00] heading_deg=218
contacts you believe in (YOUR belief only -- others may exist that you
    cannot see; age_steps is how old the underlying observation is):
  contact_id=0 believed_position=[55.36, 53.09] believed_velocity=[2.00,
    2.00] observed_heading_deg=38 age_steps=1
  contact_id=1 believed_position=[64.08, 44.40] believed_velocity=[2.00,
    2.00] observed_heading_deg=58 age_steps=1
coverage (steps since your knowledge of each region was refreshed; -1 =
    never):
  region_id=0 center=[12.50, 12.50] steps_since_seen=-1
  region_id=1 center=[12.50, 37.50] steps_since_seen=-1
  region_id=2 center=[12.50, 62.50] steps_since_seen=-1
  region_id=3 center=[12.50, 87.50] steps_since_seen=-1
  region_id=4 center=[37.50, 12.50] steps_since_seen=-1
  region_id=5 center=[37.50, 37.50] steps_since_seen=-1
  region_id=6 center=[37.50, 62.50] steps_since_seen=125
  region_id=7 center=[37.50, 87.50] steps_since_seen=98
  region_id=8 center=[62.50, 12.50] steps_since_seen=-1
  region_id=9 center=[62.50, 37.50] steps_since_seen=152
  region_id=10 center=[62.50, 62.50] steps_since_seen=1
  region_id=11 center=[62.50, 87.50] steps_since_seen=1
  region_id=12 center=[87.50, 12.50] steps_since_seen=-1
  region_id=13 center=[87.50, 37.50] steps_since_seen=163
  region_id=14 center=[87.50, 62.50] steps_since_seen=0
  region_id=15 center=[87.50, 87.50] steps_since_seen=160
your current task: DEFEND:0
episode time left: 22.0 s
your budgets: w_threat + w_engage + w_station + w_coverage + w_scout sum to
     at most 3.2; the priorities of your 2 believed contacts sum to at
    most 2
your candidate tasks (task_id, where it is, features, bid under the default
     weights):
  task_id "NEUTRALIZE:0:0"    at [55.36, 53.09]    urgency 0.86 x
    opportunity 1.00 (approaching, forecast breach 10.1 s, interception
    forecast 9.6 s), 1/(1+teammates) 0.25, age term 0.01, distance/25
    1.89, leaving 0.66, switching 1, continuity 0.00 -> default bid 1.43
  task_id "NEUTRALIZE:1:0"    at [64.08, 44.40]    urgency 0.84 x
    opportunity 1.00 (approaching, forecast breach 11.4 s, interception
    forecast 9.6 s), 1/(1+teammates) 0.25, age term 0.01, distance/25
    1.89, leaving 0.66, switching 1, continuity 0.00 -> default bid 1.42
  task_id "DEFEND:0"    at [92.86, 82.00]    coverage 0.66, distance/25
    0.00, leaving 0.00, switching 0, continuity 0.00 -> default bid 0.38
  task_id "DEFEND:1"    at [82.00, 91.94]    coverage 0.00, distance/25
    0.59, leaving 0.66, switching 1, continuity 0.00 -> default bid -0.25
  task_id "DEFEND:2"    at [74.46, 82.00]    coverage 0.00, distance/25
    0.74, leaving 0.66, switching 1, continuity 0.00 -> default bid -0.28
  task_id "DEFEND:3"    at [82.00, 73.54]    coverage 0.20, distance/25
    0.55, leaving 0.66, switching 1, continuity 0.00 -> default bid -0.15
  task_id "RECON:0"    at [12.50, 12.50]    information 0.82, distance/25
    4.25, leaving 0.66, switching 1, continuity 0.00 -> default bid -0.06
  task_id "RECON:1"    at [12.50, 37.50]    information 0.60, distance/25
    3.67, leaving 0.66, switching 1, continuity 0.00 -> default bid -0.13
  task_id "RECON:2"    at [12.50, 62.50]    information 0.43, distance/25
    3.31, leaving 0.66, switching 1, continuity 0.00 -> default bid -0.19
```

```
  task_id "RECON:3"    at [12.50, 87.50]    information 0.34, distance/25
    3.22, leaving 0.66, switching 1, continuity 0.00 -> default bid -0.23
  task_id "RECON:4"    at [37.50, 12.50]    information 0.91, distance/25
    3.55, leaving 0.66, switching 1, continuity 0.00 -> default bid 0.02
  task_id "RECON:5"    at [37.50, 37.50]    information 0.77, distance/25
    2.84, leaving 0.66, switching 1, continuity 0.00 -> default bid -0.01
  task_id "RECON:6"    at [37.50, 62.50]    information 0.48, distance/25
    2.35, leaving 0.66, switching 1, continuity 0.00 -> default bid -0.12
  task_id "RECON:7"    at [37.50, 87.50]    information 0.31, distance/25
    2.22, leaving 0.66, switching 1, continuity 0.00 -> default bid -0.19
  task_id "RECON:8"    at [62.50, 12.50]    information 0.98, distance/25
    3.03, leaving 0.66, switching 1, continuity 0.00 -> default bid 0.07
  task_id "RECON:9"    at [62.50, 37.50]    information 0.93, distance/25
    2.15, leaving 0.66, switching 1, continuity 0.00 -> default bid 0.09
  task_id "RECON:10"    at [62.50, 62.50]    information 0.42, distance/25
    1.44, leaving 0.66, switching 1, continuity 0.00 -> default bid -0.10
  task_id "RECON:11"    at [62.50, 87.50]    information 0.11, distance/25
    1.23, leaving 0.66, switching 1, continuity 0.00 -> default bid -0.23
  task_id "RECON:12"    at [87.50, 12.50]    information 0.93, distance/25
    2.79, leaving 0.66, switching 1, continuity 0.00 -> default bid 0.06
  task_id "RECON:13"    at [87.50, 37.50]    information 1.00, distance/25
    1.79, leaving 0.66, switching 1, continuity 0.00 -> default bid 0.14
  task_id "RECON:14"    at [87.50, 62.50]    information 0.54, distance/25
    0.81, leaving 0.66, switching 1, continuity 0.00 -> default bid -0.01
  task_id "RECON:15"    at [87.50, 87.50]    information 0.00, distance/25
    0.31, leaving 0.66, switching 1, continuity 0.00 -> default bid -0.23
```

Defender, consensus team at high authority, low-cost LLM: response

```
{
  "reasoning": "I am shifting focus from defense to interception because
    there are two active intruders closing in on the defended sector. I
    have increased w_threat and w_engage to prioritize neutralizing these
    threats over maintaining current station coverage.",
  "weights": {
    "w_threat": 1.5,
    "w_engage": 1.2,
    "w_station": 0.1,
    "w_coverage": 0.2,
    "w_scout": 0.2,
    "w_attrition": 0.5,
    "w_belief_age": 0.1,
    "w_travel": 0.2,
    "w_scout_travel": 0.1,
    "w_leave": 0.3,
    "w_continuity": 0.2,
    "w_switch": 0.1
  },
  "contact_priorities": [
    {
      "contact_id": 0,
      "priority": 1.2
    },
    {
      "contact_id": 1,
      "priority": 0.8
    }
```

```
  ],
  "preferred_tasks": [
    "NEUTRALIZE:0:0",
    "NEUTRALIZE:1:0"
  ]
}
```