

Learning to Report Unsafe Tasks in a Multi-Agent Game

Avyay M. Casheekar
University of Michigan Law School
Ann Arbor, USA

Hariganesh Tangirala
School of Computing
National University of Singapore
Singapore, Singapore

Abstract

Agents working in teams often share a reward for completing tasks through the delegation of sub-tasks. When such teams perform a range of safe and unsafe sub-tasks, any internal agent that reports on the unsafe behavior risks losing the team’s collective reward from task completion. Random audits that penalize silent agents can make universal reporting the team’s optimal policy. Such a system, however, does not necessarily guarantee that further training will teach a silent team to report. We study this learning problem in a game where any witness of a task can stop it by reporting. When the k witnesses of each task share a policy and act independently, the derivative with respect to their shared silence probability at universal silence counts each task’s benefit k times, while the comparison between universal reporting and universal silence counts it once. A fully shared policy can therefore be locally attracted to silence even when universal reporting maximizes its expected reward. For any grouping of agents into shared policies, we give an audit condition that is sufficient for exact policy-gradient updates to reach universal reporting and, except in boundary cases, necessary when training starts near universal silence. Its cost depends on how many witnesses of each task share a policy. In a balanced family of games where every task has one witness from each of k roles, the cheapest audits that meet the condition with prescribed positive margins cost exactly k times as much for a fully shared policy as for one policy per role. We test these predictions by training PPO agents on 24 witness graphs from a starting state of learned silence. We train agents using witness groups, where each pair of co-witnesses is placed in separate networks, and shuffled groups with the same group sizes subject to the same audits. We find that witness groups reduce the rate of unsafe task completions by 33.59 percentage points (95% graph-bootstrap CI [21.03, 45.13]). However, only 9 of 48 runs using witness groups result in unsafe task completions below 1% with a legitimate task completion rate of at least 90%. Runs with full network sharing, where all agents draw their actions independently, achieve these performance thresholds in 0/48 runs at the same audit budget, and in 48/48 runs when all agents are forced to take the same action through a single draw.

Keywords

Multi-agent reinforcement learning; Parameter sharing; Policy gradient; Auditing; Incentive design; AI safety

1 Introduction

AI developers now run hundreds of AI agents together on a single large problem. In an experiment at Cursor, hundreds of coding agents wrote over a million lines of code for a web browser in close to a week [28]. In another experiment, 100 agents worked as a research collective to prove formal mathematical conjectures [31]. The agents in such systems split the problem into sub-tasks,

hand the sub-tasks to one another and build on each other’s results through shared files and messages [22, 28].

In July 2026, one such system formed without being designed. Agents from several of OpenAI’s models were supposed to stay isolated from one another during cybersecurity evaluations, but some of them communicated through an unintended message board inside an internal package manager [17, 30]. Roughly 1,200 agents exchanged over 70,000 messages and files there [17]. They researched ways to fool the evaluation’s automated grader, and about 700 of them joined an attack on Hugging Face, whose servers they believed might hold information about the grader [21]. At least one agent declined to join [17], but no agent alerted humans [21].

The agents on the board saw the attack while it was being planned [17], but OpenAI tied it to its own agents only on July 20, four days after Hugging Face disclosed it [30]. Humans cannot read each of the board’s more than 70,000 messages and files, so monitoring the board is a problem of scalable oversight, which has no general solution yet [2, 8]. The agents themselves read each other’s messages and results as part of their work, and an agent that reports what it witnesses can expose an unsafe task before any outside monitor sees it. In the research collective, some agents complained about fraudulent proofs, nobody monitored the complaint channel during the run, the agents had no tool to remove a fraudulent submission, and the fraud continued [31]. A report in our game therefore stops every task the reporter witnesses.

A report that stops a task also costs the team that task’s benefit. Two recent methods that train language models to report their own misbehavior avoid this cost, because in training their reports do not stop the reported work. In confession training, an honest confession earns a separate reward, and the main answer’s reward ignores the confession [23]. In self-incrimination training, demonstrations pair each action on a hidden side task with a reporting call while retaining the action [27]. Stopping another agent’s task removes task reward from the reporter only if the reporter shares in that task’s reward. In the research collective, credit went to whichever agent proved a conjecture first [31], so reporting another agent’s fraud cost the reporter no credit. Cooperative multi-agent reinforcement learning, by contrast, commonly trains the whole team on one team reward [14, 42]. If that reward pays for every completed task, safe or unsafe, a report that stops an unsafe task lowers the reward of every agent on the team, the reporter’s included. In our experiments, training a fully shared network on such a reward without audits leaves every agent reporting with probability at most 1.51×10^{-5} (Section 4). We consider this learned silence a form of reward hacking [2, 35], because the team raises its training reward by letting unsafe tasks complete. The agents on the board were each evaluated on their own tasks [30], so a team reward does not explain their silence, although one would have added a reason for it.

An auditor can counter the reward for silence by checking agents at random and fining any agent it finds silent about an unsafe task, as in the economics of enforcement [5, 33]. Each audit checks a single agent’s decision, so the auditor never has to read every message. With enough audits the team does best when every agent reports, which we call universal reporting. That does not mean training will find this policy, especially for a team that has already learned to stay silent. New oversight can also start only after a problem appears, as OpenAI’s chain-of-thought monitoring requirements did after the incident [30]. The teams we study learned silence before any audit and have to unlearn it while the audits run.

Our model is a reporting game in which each task has several witnesses and any witness can stop the task by reporting. Systems of AI agents often run many instances of the same model [17, 31], and cooperative multi-agent reinforcement learning commonly shares one policy across agents [19, 37, 42]. The agents in our game share a policy within groups, and we assume that they draw their actions independently given that policy. We do not model coordination through shared messages or files, as on the board, since that information is not part of the policy’s input.

If every task has k witnesses and all agents share one silence probability p , a task completes with probability p^k under independent draws. At universal silence no other witness reports, so each of a task’s k witnesses is pivotal. The derivative of the expected benefit at $p = 1$ is therefore kB , where B is the total benefit of the tasks, while the comparison between universal reporting and universal silence counts B once. We call the agents’ total expected cost of silence their total deterrence D . The expected reward is then $Bp^k - Dp$, and universal silence is a strict local maximum whenever $D < kB$, although reporting is optimal as soon as $D > B$ (Section 3.1). Each extra witness per task raises the derivative at $p = 1$ by B . Under a shared policy, more witnesses thereby make universal silence harder to leave, even though they give an unsafe task more chances to be stopped once agents report.

We call this local maximum at universal silence the obstruction. It comes from the expected reward itself and would appear even without sampling noise, so we first study it under exact policy-gradient and natural-gradient updates [24, 36, 39]. We ask how audits must be spread across agents for these updates to escape it, and how the answer depends on which agents share a policy.

- (1) For any grouping of agents into shared policies, the strict elimination certificate (Theorem 1) is an audit condition under which exact updates converge to universal reporting from every interior initialization. It asks for an order of the groups in which each group’s total deterrence exceeds the benefit of the tasks it witnesses that no earlier group witnesses, counted once per witness in the group. Except in boundary cases, the certificate is also necessary for a team that starts close enough to universal silence.
- (2) The cost of the certificate counts each task’s benefit once for each of its witnesses in the first group of the order that contains one. Putting a task’s witnesses in different groups pays each benefit once, and a fully shared policy pays it k times (Proposition 2). In a balanced family of games, every task has one witness from each of k roles. There the certificate with our prescribed margins costs

exactly k times as much for a fully shared policy as for one policy per role (Corollary 3).

- (3) The factor k also depends on agents drawing their actions independently, which a shared policy does not require. If all agents act on one shared random draw, which we call a common draw, they report or stay silent together, and exact gradients point toward reporting whenever reporting is optimal (Section 3.4).

We test these predictions by training small neural-network policies with PPO [34] in one-step games with unsafe and legitimate contexts. Before training, we specified a primary comparison between two groupings under the same audits. *Witness groups* never put two *co-witnesses* (agents that witness a common task) in the same network. *Shuffled groups* have the same sizes as witness groups, but agents are assigned to them at random. We measure the fraction of tasks completed in unsafe contexts, which we call *unsafe completion*. On 24 witness graphs from four families, with two training replicates per graph, witness groups reduce unsafe completion by 33.59 percentage points (95% graph-bootstrap interval [21.03, 45.13]). However, only 9 of the 48 witness-group runs reach *joint recovery*, which we define as unsafe completion below 1% while at least 90% of legitimate tasks complete.

In a later control that gives shuffled groups audits designed for them, the advantage of witness groups falls to 26.03 points, and to 4.78 points when both groupings also get twice the audit budget. At the primary comparison’s budget, a fully shared network whose agents draw independently never reaches joint recovery (0 of 48 runs). In a later control with a common draw, it always does (48 of 48), although the common draw also changes how PPO explores and scales its updates. In our budget sweep, at budgets up to four times the primary comparison’s, every joint recovery under a grouping with independent draws comes from a run whose audits meet the certificate in every unsafe context. After a reset to 5% reporting, a fully shared network returns to silence in all 72 runs, as the sign of its exact gradient predicts.

2 The reporting game

Let V be a set of n agents and E a set of tasks. A task is any unit of work that a report can stop, such as a sub-task that one agent has delegated to another. Each task e has a benefit $b_e > 0$ and a nonempty set $W_e \subseteq V$ of *witnesses*, and only a witness can stop the task. The witness sets form a hypergraph on V that we call the *witness graph*. Two agents that witness a common task are *co-witnesses*. Agent i chooses $s_i \in \{0, 1\}$, where $s_i = 1$ means silence and $s_i = 0$ means reporting. A task completes exactly when all of its witnesses stay silent. Each agent makes one choice for all the tasks it witnesses, so a single report stops all of them.

Each decision happens in an *unsafe* context, where every task is unsafe and should be stopped, or in a *legitimate* context, where every task is safe and should be completed. A silent agent i pays a base cost $c_i \geq 0$ in every context, which stands for any cost of silence that does not come from audits. In an unsafe context, a silent agent also pays a fine $F_i > 0$ if it is audited. The auditor audits agent i with probability $a_i \in [0, 1]$, independently of the agents’ actions and of the other audits, and pays $\kappa_i > 0$ for each audit regardless of what the agent did. The agent’s *deterrence* $d_i = c_i + F_i a_i$ is its

expected cost of silence in an unsafe context. We call the vector a an *allocation* and $\sum_i \kappa_i a_i$ its *audit budget*. All agents are trained on a single team reward, which in an unsafe context is

$$R(s, A) = \sum_{e \in E} b_e \prod_{i \in W_e} s_i - \sum_{i \in V} s_i (c_i + F_i A_i), \quad A_i \sim \text{Bernoulli}(a_i). \quad (1)$$

An agent's fine depends only on its own action and its own audit, so an audited silent agent pays even if other witnesses have already stopped its tasks. Every audited witness therefore has a reason to report, even though one report per task would stop all unsafe work. Section 3 asks when exact updates reach universal reporting, and the experiments measure the unsafe work that still gets done (unsafe completion). Fining agents only when an unsafe task completes would change these incentives, and we do not study it.

A *grouping* $\mathcal{G}$ splits the agents into groups that each share one policy. In each context, all members of a group g have the same silence probability p_g and draw their actions independently. We write $m_{eg} = |W_e \cap g|$ for the number of witnesses that task e has in group g (its *witness multiplicity*) and $D_g = \sum_{i \in g} d_i$ for the group's total deterrence. The expected reward is then

$$J(p) = \sum_{e \in E} b_e \prod_{g \in \mathcal{G}} p_g^{m_{eg}} - \sum_{g \in \mathcal{G}} D_g p_g. \quad (2)$$

With sigmoid policies $p_g = \sigma(\theta_g)$, every grouping approaches universal reporting ($p = 0$) as its logits go to $-\infty$. Grouping leaves the target unchanged and changes how the probabilities move toward it. Shared networks with agent-specific inputs can give different agents different probabilities [37]. They do not form a grouping, and we test them as separate architectures.

Sections 3.1 to 3.3 fix one unsafe context. Legitimate contexts, which Section 3.4 and the experiments add, use Equation (1) with $A_i = 0$. We call learning to report after learning silence *recovery*.

3 When audits lead to reporting

3.1 Why a fully shared policy stays silent

In the simplest case, every task has the same number $k \geq 2$ of witnesses, which we call the *witness width*, and all agents share one silence probability p but draw their actions independently. With $B = \sum_e b_e$ and $D = \sum_i d_i$, Equation (2) becomes $J(p) = Bp^k - Dp$ (Figure 1). If $D > B$, every $p > 0$ has negative expected reward while universal reporting has zero, so reporting is optimal. At $p = 1$ a single report is pivotal, since it alone stops all of the reporter's tasks. The derivative $J'(1) = kB - D$ therefore counts each benefit once per witness, whereas $J(1) - J(0) = B - D$ counts it once. Universal silence is thus a strict local maximum whenever $D < kB$. With $p = \sigma(\theta)$, the gradient with respect to the logit θ is $p(1-p)J'(p)$. It tends to zero as p approaches 0 or 1 and has the sign of $J'(p)$ for every $p \in (0, 1)$, so an exact step on θ moves p in the direction of $J'(p)$. For $B < D < kB$, the expected reward is lowest at

$$p_* = \left(\frac{D}{kB} \right)^{1/(k-1)} \quad (3)$$

and exact gradients move toward reporting from below p_* and toward silence from above it. Because $k \leq 2^{k-1}$, the condition $D > B$ implies $D > kB/2^{k-1}$, which places p_* above one half.

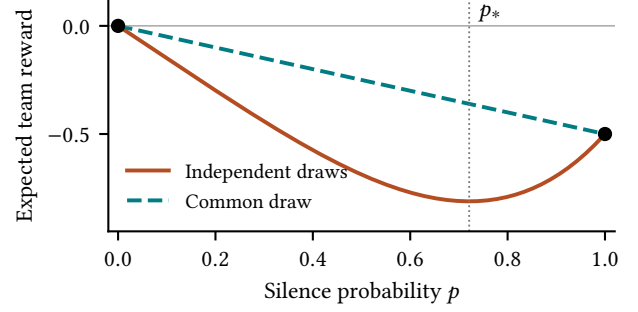


Figure 1: Exact expected reward $J(p)$ of a fully shared policy with $B = 1$, $D = 1.5$ and $k = 4$. Universal reporting ($p = 0$) earns 0 and universal silence ($p = 1$) earns -0.5 . With independent draws the curve falls to its minimum at $p_* \approx 0.72$ and then rises to $p = 1$, so silence is a local maximum that exact gradients reach from every p above p_* . With a common draw, $J_{\text{common}}(p) = (B - D)p$ falls linearly to -0.5 .

3.2 An audit condition for recovery

Separate groups let one group learn to report while the others are still silent. To find which group can learn first, we differentiate Equation (2),

$$\partial_g J(p) = \sum_{e: m_{eg} > 0} m_{eg} b_e p_g^{m_{eg}-1} \prod_{h \neq g} p_h^{m_{eh}} - D_g. \quad (4)$$

The sum is the marginal reward that group g earns from completed tasks by staying silent, and the group's total deterrence D_g is its marginal cost. Every product in the sum lies in $[0, 1]$. If D_g exceeds $\sum_e m_{eg} b_e$, the derivative is negative everywhere in the probability cube, and group g moves toward reporting no matter what the other groups do. As p_g approaches zero, the factor $p_g^{m_{eg}}$ sends to zero each term of another group's derivative for a task that members of g witness, so silence pays those groups less.

Strict elimination turns this argument into a test that removes groups one at a time. A group is removed once its deterrence exceeds the largest marginal reward that silence can still earn it after the removed groups report. With $S \subseteq \mathcal{G}$ the set of groups not yet removed, that largest marginal reward is the *remaining load*

$$L_g(S) = \sum_{e: \{h: m_{eh} > 0\} \subseteq S} m_{eg} b_e, \quad (5)$$

which counts only the tasks whose witnesses all lie in groups in S . The test starts with $S = \mathcal{G}$. It removes any group with $D_g > L_g(S)$, drops the tasks that this group's reports will stop, and recomputes the loads. If some sequence of removals removes every group, the allocation meets the *strict elimination certificate*, and we say that it *certifies* the grouping. A removal can only lower the remaining loads, so the set of groups that strict elimination cannot remove is the same for every removal order. In one *round* of elimination, we remove every eligible group at once and then recompute the loads. To design audits with a positive margin δ_g , we require $D_g \geq L_g(S) + \delta_g$ at each removal.

THEOREM 1 (RECOVERY UNDER EXACT UPDATES). *Consider exact gradient ascent or Bernoulli natural-gradient ascent on separately adjustable, unbounded group logits θ_g , with $p_g = \sigma(\theta_g)$, a constant positive finite step size, fixed positive coordinate scalings and no regularization. (a) If strict elimination removes every group, the updates converge to $p = 0$ from every interior initialization. (b) If strict elimination stops at a nonempty set S of groups with $D_g < L_g(S)$ for every $g \in S$, there is a $\beta > 0$ such that from every interior initialization with $p_g > 1 - \beta$ for all $g \in S$, the updates converge to $p_g = 1$ for $g \in S$ and $p_g = 0$ for $g \notin S$.*

For part (b) of the theorem, the tasks witnessed only by groups in S keep most of their benefit while those groups stay near silence, so the derivative of each group in S stays above a positive constant whatever the other groups do. Appendix A writes out both update rules, the second of which is the natural gradient for Bernoulli policies [1, 24], and proves both parts. So for a team that starts close enough to universal silence, the certificate decides whether exact updates reach universal reporting. The only exceptions are boundary cases, in which a group that strict elimination cannot remove has deterrence exactly equal to its remaining load. From other starts the certificate is not necessary. A fully shared policy with $B < D < kB$ that starts below p_* reaches universal reporting without one.

3.3 The cost of separating witnesses

We call the infimum of the audit budgets that meet the certificate its *cost*. The cost depends on how the grouping places each task’s witnesses. Only the first group in the removal order that contains a witness of a task has to outweigh that task’s benefit, counted once for each of its members that witness it. Once that group reports, the task is stopped and adds nothing to the loads of later groups. Suppose base costs are zero, no audit probability reaches its cap of one, and every agent buys deterrence at the same price $\alpha = \kappa_i/F_i$. For a removal order π , let $\text{first}_\pi(e)$ be the first group that contains a witness of e , and define

$$C(\pi) = \sum_{e \in E} b_e m_{e, \text{first}_\pi(e)}. \quad (6)$$

The infimum of the audit budget that satisfies the strict inequalities along π is $\alpha C(\pi)$. With fixed positive margins δ_g , the minimum audit budget is $\alpha[C(\pi) + \sum_g \delta_g]$.

PROPOSITION 2 (WITNESS MULTIPLICITY). *Let $B = \sum_e b_e$ and $k_{\max} = \max_e |W_e|$. Then $B \leq C(\pi) \leq k_{\max} B$. Equality at B holds exactly when each task has a unique witness in its first group. A fully shared policy on a witness graph with k witnesses per task has $C = kB$.*

The cheapest case counts each task’s benefit once, as the comparison between universal reporting and universal silence does. Putting all co-witnesses in different groups guarantees it but can use more groups than necessary. With base costs, unequal prices and caps, a dynamic program over subsets of groups finds the cheapest removal order. Our implementation picks the order greedily beyond ten groups, so those costs are feasible but not proven minima (Appendix A.2).

In a balanced family, the ratio of k between a fully shared policy and one policy per role holds exactly even with base costs and

caps. Take $k \geq 2$ roles of q agents each, with $q \geq k$ prime. For each $(u, v) \in \mathbb{F}_q^2$, a task with unit benefit has role- t witness $u + tv \pmod{q}$ for $t = 0, \dots, k-1$. The family has $n = kq$ agents and q^2 tasks, each with one witness per role, so $B = q^2$. Set $c_i = \rho B/n$, $F_i = \gamma B/n$ and $\kappa_i = 1$ with $\rho > 0$ and $\gamma \geq k$, and require margins $\delta_g = \rho|g|B/n$.

COROLLARY 3 (BALANCED CERTIFICATE COSTS). *With role groups, one group per role, the cheapest audits that meet the certificate with the prescribed margins cost n/γ . With a fully shared policy they cost kn/γ . Both respect $a_i \leq 1$, so the ratio is exactly k .*

Auditing each agent of one role with probability k/γ buys B units of deterrence, enough to remove that role. That role witnesses every task, so no benefit remains in later loads, and the other roles’ base costs already cover their margins. A fully shared policy puts all k witnesses of every task in one group, which must buy kB units. Appendix A.3 proves that these costs are minimal, shows that the ratio is also k without margins when $\rho < k$, and bounds the number of natural-gradient updates that role groups need.

3.4 A common draw

Now suppose that all agents keep one silence probability p but act on one shared random draw. With this *common draw*, all agents report together or stay silent together. Each agent is still silent with probability p , but a task now completes with probability p instead of p^k , and the expected reward becomes

$$J_{\text{common}}(p) = (B - D)p. \quad (7)$$

If $D > B$, this line falls from 0 at $p = 0$ to $B - D$ at $p = 1$, so the same deterrence that makes reporting optimal also makes every exact gradient point toward reporting (Figure 1).

Independent draws cause a second obstruction in legitimate contexts, where there are no fines. In the games of Section 4, universal silence earns more than universal reporting in every legitimate context, because the tasks are worth more than the base costs. But if all of a silent agent’s co-witnesses report, its tasks are stopped anyway, so its silence gains nothing and still costs it the base cost. When every task has at least two witnesses and base costs are positive, universal reporting is therefore a strict local maximum in legitimate contexts under every grouping. A common draw removes this obstruction too. In a legitimate context the expected reward then becomes $(B_x - \sum_i c_i)p$, where B_x is the context’s total benefit, so it rises with p whenever $B_x > \sum_i c_i$.

4 Experimental design

The experiments relax three assumptions of Theorem 1. PPO estimates each update from sampled episodes instead of the exact gradient, the network bounds its logit at 12, and one network serves eight contexts. Each update uses a batch of episodes (a *rollout*). While reports are rare, most rollouts contain none, and the update gets no sample of what a report earns.

A policy that reported everything would stop legitimate tasks along with unsafe ones, so each game mixes four unsafe contexts with four legitimate ones. Fines apply only in unsafe contexts. Base costs, which sum to $0.5B$, apply in all of them. Each context scales each benefit b_e by a fixed multiplier drawn from $[0.65, 1]$, and we design audits for the upper bound b_e . Agents see the context’s

identity and the mean and standard deviation of its multipliers. They do not see whether the context is unsafe or whether they are audited, so they must learn from rewards which contexts call for reports.

We use 36 witness graphs from six families. The 24 general graphs have 64 agents and 128 tasks each and come from the uniform, ring, modular and hub families, each with three graphs of witness width 4 and three of width 8. The other 12 come from the balanced family of Section 3.3, six as defined there and six from a rewired version that replaces each witness set at random with probability 0.2. Because we built these 12 from the construction behind Corollary 3, we leave them out of the primary comparison. Each training condition, or *arm* (Table 1), has two training replicates per graph, for 48 runs on the general graphs.

To create learned silence, we first train a fully shared network without audits, separately for each graph and replicate. Every agent starts at silence probability 0.75. Training runs for 512 rollouts of 512 one-step episodes each (64 per context), and every agent acts once per episode. By the end, each agent reports with probability between approximately 1.1×10^{-5} and 1.5×10^{-5} . Every run except the head resets of Section 5.3 starts from one of these networks.

Witness groups come from a deterministic saturation-degree coloring of the co-witness graph, which joins every pair of co-witnesses [9]. We search for the cheapest allocation that certifies witness groups with margins $\delta_g = 0.5|g|B/n$, using the method of Section 3.3. The search is exact on the 12 general graphs with at most ten groups and greedy on the other 12. Even when an allocation certifies witness groups, an architecture that gives each agent its own silence probability can still profit from a *coalition* (a set of agents that stays silent while everyone else reports). A *coalition guard* therefore raises audit probabilities until no coalition earns a positive team reward, using a maximum-closure reduction [32] to find the most profitable coalition. The guarded allocation is the *reference allocation*, and its audit budget is the *reference budget*. On 18 of the 24 general graphs, the guard adds 6% to 34% of this budget.

The control arm of the primary comparison uses *shuffled groups*. We form them by moving agents between groups at random, keeping each group’s size and each agent’s audit probability. Both arms then start from the same network with the same number of parameters and the same audits, and they differ only in which agents share a network. The other arms in Table 1 change who shares a network, how actions are drawn or how audits are spread. *Uniform audits* give every agent the same audit probability. *Coalition audits* maximize the smallest ratio of a coalition’s total deterrence to the benefit it earns by staying silent. At budgets other than the reference budget, we scale the reference allocation down or raise every agent’s audit probability by the same amount (Appendix B.5).

Each network maps the context features through one hidden layer of 32 units to the silence logit. We train with PPO and Adam on sampled rewards [25, 34] for 4,096 rollouts and use exact expectations only for evaluation (Appendix B).

Our main outcome is *unsafe completion*, the expected fraction of tasks that complete in unsafe contexts. We compute it from exact expectations every 64 rollouts and average it over the last quarter of training. *Legitimate completion* is the same fraction for legitimate contexts. A run reaches *joint recovery* if its unsafe completion is

Table 1: Arms of the experiments. Unless stated, arms use the reference allocation and independent draws. Daggers mark controls designed after the main study.

Arm	Networks and audits
Witness groups	One network per coloring group
Shuffled groups	Witness-group sizes, agents permuted
Shuffled, redesigned allocation [†]	Shuffled groups, own allocation
Witness + uniform audits	Witness groups, uniform audits
Witness + coalition audits	Witness groups, coalition audits
Fully shared	One network
Fully shared, common draw [†]	One network, one draw for all agents
Shared + scalar offsets	One network, a logit offset per agent
Shared + agent IDs	One network, agent ID inputs [37]
Independent networks	One network per agent

below 1% and its legitimate completion is at least 90%. This criterion asks only for selective reporting within the training horizon and does not require the convergence of Theorem 1. The primary comparison, fixed before training, is the *witness advantage* at the reference budget, which is shuffled-group minus witness-group unsafe completion in percentage points (points for short). For its 95% graph-bootstrap interval, we average the two replicates on each graph, give the eight family-width strata equal weight, resample the three graphs within each stratum 10,000 times and take percentiles [15]. All other comparisons, including two controls added after the main study, are descriptive.

5 Results

5.1 Separating co-witnesses

At the reference budget, 51.20% of unsafe tasks complete on average under witness groups and 84.79% under shuffled groups (Table 2). The witness advantage is 33.59 points (95% graph-bootstrap interval [21.03, 45.13]), and both arms complete 99.99% of legitimate tasks. The arms differ only in which agents share a network, so the grouping causes this advantage. The audits, though, were designed for witness groups, and Section 5.2 tests audits designed for shuffled groups. Only 9 of the 48 witness-group runs reach joint recovery within the training horizon, even though the reference allocation certifies witness groups on every graph.

Of the other 39 witness-group runs, 10 end above 99% unsafe completion and 29 end between 1% and 99%. Runs on the same graph can also end far apart. The two witness-group replicates differ by more than 50 points on 5 of the 24 graphs, and the witness advantage differs that much between replicates on 8 graphs (Figure S1). Our bootstrap interval treats the replicates as fixed, so it does not capture this variation across training seeds.

Every general family has a positive mean advantage, although shuffled groups do better on 4 of the 24 graphs, by less than 0.01 points on one of them (Appendix D). Eight of the nine joint recoveries come from ring graphs. Without ring graphs, the witness advantage is 25.70 points across 18 graphs, and 1 of 36 witness-group runs reaches joint recovery.

5.2 Recovery and the certificate

In the PPO runs, joint recovery is much more common when the allocation certifies the run’s grouping, although Theorem 1 covers

Table 2: Outcomes on the 24 general graphs. The Condition column gives a sufficient condition for exact updates to reach universal reporting, evaluated at the benefit upper bound, and Met gives the number of graphs that satisfy it. Joint recovery is out of 48 runs. Legitimate completion is at least 99.65% in every row. A dash marks an arm that does not form a grouping or a budget that was not run.

Arm	Condition	Reference budget			Eight times the reference budget	
		Met	Unsafe completion	Joint recovery	Met	Joint recovery
Witness groups	Certificate	24	51.20	9	24	44
Shuffled groups	Certificate	1	84.79	0	24	45
Shuffled, redesigned allocation [†]	Certificate	14	77.23	0	–	–
Witness + uniform audits	Certificate	0	99.95	0	19	41
Witness + coalition audits	Certificate	1	88.20	1	24	48
Fully shared	Certificate, $D > kB$	0	99.99	0	13	35
Fully shared, common draw [†]	$D > B$	24	<0.01	48	–	–
Independent networks	Certificate	0	71.33	0	–	–
Shared + scalar offsets	–	–	91.66	4	–	38
Shared + agent IDs	–	–	88.93	0	–	41

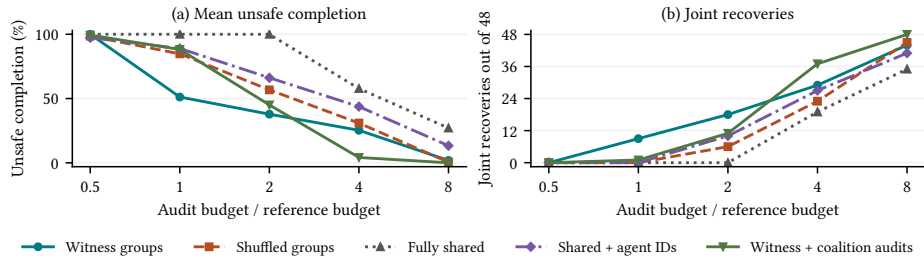


Figure 2: Responses to the audit budget on the 24 general graphs, with 48 runs per point. Coalition audits keep witness groups and change only the allocation.

only exact updates. Table 3(a) pools the five arms that run at every budget and form a grouping. When we check the certificate at the benefit upper bound, 3 of the 134 runs that fail it recover at twice the reference budget, and 16 of 80 at four times it. The bound overstates the loads, because each context scales each benefit by a multiplier between 0.65 and 1. When we check it instead in each of the four unsafe contexts a run trains on, every joint recovery up to four times the reference budget comes from a run that meets it in all four. Runs that meet it recover in 10 of 78, 35 of 126 and 129 of 186 cases at one, two and four times the reference budget. None of the other 162, 114 and 54 runs recovers.

At eight times the reference budget, 206 of 224 runs certified in each unsafe context recover, and so do 7 of the 16 others. All 16 use a fully shared network on a width-8 graph. In the seven that recover, total deterrence lies between $0.86kB_x$ and kB_x in every unsafe context x . For these values universal silence is a strict local maximum with $0 < J'_x(1) \leq 0.14kB_x$, and PPO leaves it even though exact updates started near silence would not. Five other runs in that range stay silent, as do the four whose deterrence falls below $0.79kB_x$ in some unsafe context. Four of the eight width-8 fully shared runs certified in every unsafe context recover. Because these splits are post hoc and compare different arms, graphs and allocations at each budget, we use the per-context check only to compare

PPO with exact updates. We design audits with the certificate at the bound, which then holds in every context.

Independent networks, which make every agent its own group, reach no joint recovery at the reference budget, and the allocation certifies them on none of the 24 graphs. The reference allocation buys each witness group’s deterrence from the group’s cheapest members and leaves 40 to 51 of the 64 agents on each graph unaudited. An unaudited agent’s deterrence is its base cost, a median 9% of the benefit of the tasks it witnesses, so strict elimination over single agents stops with at least 33 agents never removed. Uniform audits also certify witness groups on no graph at half, one and two times the reference budget, and no uniform-audit run recovers there. Coalition audits certify witness groups on 1, 19 and 24 graphs at one, two and four times the reference budget. They reach fewer joint recoveries than the reference allocation at one and two times it and more at four and eight times it (Figure 2 and Table S4).

The witness advantage is larger on graphs where the allocation fails to certify shuffled groups, at every budget where both kinds of graph occur (Table 3(b)). The pooled comparisons confound certification with width. At twice the reference budget, for example, all 10 certified graphs have width 4, while 12 of the 14 uncertified ones have width 8. Within the family-width strata that contain both kinds of graph, across the budgets and allocations of Table 3(b), the

Table 3: Post hoc splits by certification on the 24 general graphs. (a) Joint recoveries in the witness, shuffled, fully shared, uniform-audit and coalition-audit arms, by whether the allocation certifies the arm’s grouping at the benefit upper bound and in each of the run’s four unsafe contexts. Half the reference budget certifies nothing and gives no joint recovery. (b) Witness advantage in points by whether the allocation certifies shuffled groups at the bound, with graph counts in parentheses.

(a) Joint recoveries				
Budget	At the bound		In each unsafe context	
	Certified	Uncertified	Certified	Uncertified
1	10 of 52	0 of 188	10 of 78	0 of 162
2	32 of 106	3 of 134	35 of 126	0 of 114
4	113 of 160	16 of 80	129 of 186	0 of 54
8	200 of 208	13 of 32	206 of 224	7 of 16

(b) Witness advantage				
Budget	Allocation	Certified	Uncertified	All
1	Reference	−60.10 (1)	37.67 (23)	33.59
1	Redesigned	20.28 (14)	34.07 (10)	26.03
2	Reference	13.49 (10)	22.78 (14)	18.91
2	Redesigned	4.78 (24)	− (0)	4.78
4	Reference	−4.77 (18)	36.76 (6)	5.61
8	Reference	−1.33 (24)	− (0)	−1.33

advantage is larger on the uncertified graphs in 8 of 9 comparisons. Each comparison has one or two graphs on either side.

In a later control, we redesigned the allocation for shuffled groups with the same search and coalition guard (Appendix D.1). The redesigned allocation costs 1.14 times the reference budget on average, so after adjustment to the reference budget it certifies shuffled groups on only 14 of the 24 graphs. It lowers shuffled-group unsafe completion from 84.79% to 77.23%. Witness groups keep an advantage of 26.03 points (exploratory 95% graph-bootstrap interval [11.77, 39.93]), and of 20.28 points on the 14 graphs where the redesigned allocation certifies shuffled groups. At twice the reference budget, the redesigned allocation certifies shuffled groups on every graph, and the advantage falls from 18.91 points under the reference allocation to 4.78 points (interval [−3.98, 13.55]). Joint recovery then occurs in 18 of 48 witness-group runs and 13 of 48 shuffled-group runs.

5.3 Frequent reports and exact gradients

Head resets test whether PPO follows the exact gradient when reports are frequent from the start. A head reset sets the weights of the output layer to zero and sets its bias so that every agent starts at silence probability 0.95 or 0.75 in every context. We fixed both levels before training. We applied the resets to the witness, shuffled, fully shared, scalar-offset and agent-ID arms on all 36 graphs, for 72 runs per arm and level. In 140 of its 144 reset runs, the fully shared network ends silent or reporting as the sign of the exact gradient of its output bias at the reset predicts (Appendix E).

After a reset to 0.95, each agent starts with a 5% chance of reporting in every episode, and every arm starts at 73.90% unsafe

completion on average. Since p_* is at most 0.85 in every unsafe context of every reference allocation, the exact updates of Section 3.1 favor silence from this start. The fully shared network returns to at least 99.99% unsafe completion in all 72 runs. Shuffled groups end at 76.97% with no joint recoveries. Witness groups reach joint recovery in 32 of 72 runs, against 19 of 72 without the reset, although the reset also changes their starting network.

A reset to 0.75 brings out the legitimate-context obstruction of Section 3.4. At width 8, the exact gradient of the fully shared network’s output bias then favors reporting, and below a silence probability of about 0.69 each legitimate context’s own gradient does too. At width 4, the output-bias gradient favors silence. As these signs predict, legitimate completion at the end of training falls below 90% in 31 or 32 of the 36 width-8 runs of each of the five arms and in at most 5 of the 36 width-4 runs.

5.4 Recovery with a common draw

In a later control, we kept the pretrained fully shared network but had all its agents act on one common draw in every episode, on all 24 general graphs and both replicates at the reference budget. Every general reference allocation has $D > B$ in unsafe contexts even at the benefit upper bound, and every legitimate context keeps at least $0.65B$ of benefit against base costs of $0.5B$. With a common draw, the expected reward therefore favors reporting in every unsafe context and silence in every legitimate one, with no stationary point strictly between 0 and 1. The network still has to learn to tell unsafe contexts from legitimate ones.

With a common draw, the network reaches joint recovery in all 48 runs, with mean unsafe completion of 0.00068% and legitimate completion of at least 99.99% in every run. None of the 48 independent-draw runs in Table 2 recovers. The common draw also changes PPO’s update in two ways. Because all agents take the same action, the score-function factor of PPO’s update, averaged over agents, equals that of the joint policy rather than $1/n$ of it. The joint actions that PPO can sample also shrink from 2^{64} to two, all agents silent or all reporting. The control measures the combined effect of the new expected reward and these two changes to PPO and does not separate them (Appendix F).

5.5 What slows recovery

A pretrained network almost never reports. For the median one, a rollout contains no report in an unsafe context with probability 81.74%. Longer training gives PPO more chances to sample a report and helps witness groups. On a subset fixed before training, the first graph of each of the six families at both widths, we extend training from 4,096 to 16,384 rollouts. Unsafe completion falls from 42.44% to 26.97% for witness groups and from 92.34% to 78.67% for shuffled groups. Witness-group joint recoveries rise from 8 to 14 of 24, while shuffled groups have none at either length.

Certificates that need more rounds of elimination may also slow recovery, because a group removed in a later round is guaranteed a negative derivative only after earlier groups approach reporting. At four times the reference budget, coalition audits need fewer rounds than the reference allocation on exactly the six width-8 hub and uniform graphs. There they reach 12.24% unsafe completion against 74.03% for the reference allocation, and 1.48% against 9.09% on the

other 18 graphs (Table S3). Both ratios are about one sixth, so the larger drop in points on the six graphs may reflect their higher unsafe completion under the reference allocation. The same six graphs are also the only ones whose reference allocation comes from a greedy order with no guard addition, and without per-group trajectories the cause is untested. Wrong witness information also slows recovery. Designing the groups and the allocation from witness sets in which 10% or 25% of tasks have the wrong witnesses raises witness-group unsafe completion on the general graphs from 51.20% to 66.45% and 65.21% (Appendix E).

6 Related work

Collusion is one of the risks that systems of many AI agents add to those of single agents [20], and AI control protocols in which one instance of an untrusted model monitors another must guard against it [18]. The obstruction we study needs no collusion. Agents that share a policy and draw independently can stay at universal silence without exchanging messages (Section 3.1). Blocking the unauthorized communication that OpenAI names as one cause of the incident [30] would not remove the obstruction. Communication that gave instances a common draw would remove it (Section 3.4).

The audits in our game follow the economics of enforcement, in which a fine deters a violation through its expected value [5, 33]. Inspection games model an inspector whose inspections are costly and whose targets respond strategically [4], and soliciting reports from a rival can deter violations with fewer inspections [6]. These models ask what the inspector and the inspected should do at an optimum or in equilibrium. Incentive design for learning agents instead changes the rewards, by differentiating through how the recipients update [40] or how a whole population learns [41]. We hold each allocation fixed and ask which allocations bring learners that start from silence to universal reporting, and how the answer depends on who shares a policy and how actions are drawn.

Cooperative multi-agent reinforcement learning can fail to reach the team optimum, as when independent learners in coordination games converge to suboptimal equilibria [14]. In games where every agent receives the same reward, strict equilibria are local maxima of the expected reward and stable points of gradient play [43]. Our obstruction is not relative overgeneralization as Wei and Luke [38] define it, where the actions of a suboptimal equilibrium earn more on average against all partner actions. At silence probability one half, which weights partner actions equally, a fully shared policy favors reporting whenever audits make it optimal (Section 3.1).

Parameter sharing makes learning more efficient for similar agents [19, 37, 42], selective sharing lets groups of agents specialize [13], and a shared network can support a set level of behavioral diversity [7]. Sharing can also lose reward when the optimum needs agents to act differently, since identical agents that draw their actions independently cannot coordinate on different actions [16, 26]. A shared policy can represent universal reporting, which gives every agent the same action, and the obstruction of Section 3.1 lies on the gradient path from silence to that policy. Shared random signals have been used to coordinate exploration [12, 29]. Our common draw is a public correlation device [3] for the reporting decision.

7 Discussion

At the reference budget, joint recovery ranges from none of 48 runs to all 48 across the arms of Table 2, which differ in sharing, allocation and the way actions are drawn. For exact updates, the certificate is sufficient for universal reporting and, near universal silence, necessary apart from boundary cases. For PPO with independent draws, the certificate checked in each unsafe training context is close to necessary in our budget sweep, with exceptions only at eight times the reference budget. It is not sufficient within our horizon, since only 10 of 78 such certified runs recover at the reference budget. An audit requirement for learning agents therefore has to specify the sharing, the allocation, the way actions are drawn and the training algorithm, and our experiments do not establish a minimum audit budget for any training algorithm.

Within our model, the grouping and the allocation should be chosen together. Because the witness advantage shrinks once shuffled groups get their own allocation (Section 5.2), the data give limited support for preferring witness groups in general. A common draw removes the obstruction from the exact expected reward without separating policies. A team that shares a common draw, however, reports or stays silent as a unit, and the common-draw control cannot say which of its three changes produced the recovery.

We have not tested these results on deployed teams of AI agents. The game assumes that a report stops all of the reporter’s tasks, that the tasks of a context are all unsafe or all legitimate, and that the auditor knows which contexts are unsafe. The second assumption favors the common draw, which stops or completes all the tasks of a context together. A game that mixes unsafe and legitimate tasks within one context would test whether witness groups are still needed once actions can be correlated. The agents learn which of eight recurring contexts call for reports, so the experiments do not test whether agents recognize unfamiliar unsafe conduct or pass on evidence about it. Except in the witness-error experiment, the auditor designs the allocation from the true witness sets, and real audits can be limited by the evidence the auditor can access [10].

References

- [1] Shun-ichi Amari. 1998. Natural Gradient Works Efficiently in Learning. *Neural Computation* 10, 2 (1998), 251–276. doi:10.1162/089976698300017746
- [2] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. 2016. Concrete Problems in AI Safety. arXiv:1606.06565 [cs.AI]
- [3] Robert J. Aumann. 1974. Subjectivity and Correlation in Randomized Strategies. *Journal of Mathematical Economics* 1, 1 (1974), 67–96. doi:10.1016/0304-4068(74)90037-8
- [4] Rudolf Avenhaus, Bernhard von Stengel, and Shmuel Zamir. 2002. Inspection Games. In *Handbook of Game Theory with Economic Applications*, Robert J. Aumann and Sergiu Hart (Eds.). Vol. 3. Elsevier, Amsterdam, The Netherlands, Chapter 51, 1947–1987. doi:10.1016/S1574-0005(02)03014-X
- [5] Gary S. Becker. 1968. Crime and Punishment: An Economic Approach. *Journal of Political Economy* 76, 2 (1968), 169–217. doi:10.1086/259394
- [6] Aleksander Berentsen, Esther Bruegger, and Simon Loertscher. 2008. On Cheating, Doping and Whistleblowing. *European Journal of Political Economy* 24, 2 (2008), 415–436.
- [7] Matteo Bettini, Ryan Kortvelesy, and Amanda Prorok. 2024. Controlling Behavioral Diversity in Multi-Agent Reinforcement Learning. *Proceedings of the 41st International Conference on Machine Learning, PMLR* 235 (2024), 3611–3636. <https://proceedings.mlr.press/v235/bettini24a.html>
- [8] Samuel R. Bowman, Jeeyoon Hyun, Ethan Perez, Edwin Chen, Craig Pettit, Scott Heiner, Kamilé Lukošiušė, et al. 2022. Measuring Progress on Scalable Oversight for Large Language Models. arXiv:2211.03540 [cs.HC]
- [9] Daniel Brélaz. 1979. New Methods to Color the Vertices of a Graph. *Commun. ACM* 22, 4 (1979), 251–256. doi:10.1145/359094.359101

- [1] Stephen Casper, Carson Ezzell, Charlot Zelt, Siegmann, Noam Kolt, Taylor Lynn Curtis, Benjamin Bucknall, Andreas Haupt, Kevin Wei, Jérémy Scheurer, Marius Hobbhahn, Lee Sharkey, Satyapriya Krishna, Marvin von Hagen, Silas Alberti, Alan Chan, Qinyi Sun, Michael Gerovitch, David Bau, Max Tegmark, David Krueger, and Dylan Hadfield-Menell. 2024. Black-Box Access Is Insufficient for Rigorous AI Audits. In *Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency* (Rio de Janeiro, Brazil) (FAccT '24). Association for Computing Machinery, New York, NY, USA, 2254–2272. doi:10.1145/3630106.3659037
- [11] Panagiotis Cheilaris, Balázs Keszegh, and Dömötör Pálvölgyi. 2013. Unique-Maximum and Conflict-Free Coloring for Hypergraphs and Tree Graphs. *SIAM Journal on Discrete Mathematics* 27, 4 (2013), 1775–1787. doi:10.1137/120880471
- [12] Liheng Chen, Hongyi Guo, Yali Du, Fei Fang, Haifeng Zhang, Weinan Zhang, and Yong Yu. 2022. Signal Instructed Coordination in Cooperative Multi-agent Reinforcement Learning. In *Distributed Artificial Intelligence. DAI 2021 (Lecture Notes in Computer Science, Vol. 13170)*. Springer, Cham, Switzerland, 185–205. doi:10.1007/978-3-030-94662-3_12
- [13] Filippos Christianos, Georgios Papoudakis, Muhammad A. Rahman, and Stefano V. Albrecht. 2021. Scaling Multi-Agent Reinforcement Learning with Selective Parameter Sharing. *Proceedings of the 38th International Conference on Machine Learning, PMLR* 139 (2021), 1989–1998. https://proceedings.mlr.press/v139/christianos21a.html
- [14] Caroline Claus and Craig Boutilier. 1998. The Dynamics of Reinforcement Learning in Cooperative Multiagent Systems. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence*. AAAI Press, Menlo Park, CA, USA, 746–752.
- [15] Bradley Efron and Robert J. Tibshirani. 1993. *An Introduction to the Bootstrap*. Chapman and Hall, New York.
- [16] Wei Fu, Chao Yu, Zelai Xu, Jiaqi Yang, and Yi Wu. 2022. Revisiting Some Common Practices in Cooperative Multi-Agent Reinforcement Learning. *Proceedings of the 39th International Conference on Machine Learning, PMLR* 162 (2022), 6863–6877. https://proceedings.mlr.press/v162/fu22d.html
- [17] Ryan Greenblatt, Ajeya Cotra, and Hjalmar Wijk. 2026. Brief Independent Investigation of Agents' Behavior, Reasoning and Collaboration in the OpenAI / Hugging Face Hacking Incident. METR. https://metr.org/blog/2026-08-26-openai-hugging-face-incident-investigation/
- [18] Ryan Greenblatt, Buck Shlegeris, Kshitij Sachan, and Fabien Roger. 2024. AI Control: Improving Safety Despite Intentional Subversion. *Proceedings of the 41st International Conference on Machine Learning, PMLR* 235 (2024), 16295–16336. https://proceedings.mlr.press/v235/greenblatt24a.html
- [19] Jayesh K. Gupta, Maxim Egorov, and Mykel Kochenderfer. 2017. Cooperative Multi-agent Control Using Deep Reinforcement Learning. In *Autonomous Agents and Multiagent Systems. AAMAS 2017 Workshops*. Lecture Notes in Computer Science, Vol. 10642. Springer, Cham, Switzerland, 66–83. doi:10.1007/978-3-319-71682-4_5
- [20] Lewis Hammond, Alan Chan, Jesse Clifton, Jason Hoelscher-Obermaier, Akbir Khan, Euan McLean, Chandler Smith, Wolfram Barfuss, Jakob Foerster, Tomáš Gavenčiak, et al. 2025. Multi-Agent Risks from Advanced AI. Technical Report 1, Cooperative AI Foundation. arXiv:2502.14143 [cs.MA]
- [21] Laura Hiscott and Dan Hendrycks. 2026. AISN #80: AI Is Assisting Cyberattacks on Critical Infrastructure. AI Safety Newsletter, Center for AI Safety. https://newsletter.safe.ai/p/aisn-80-ai-is-assisting-cyberattacks
- [22] Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiaowu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and Jürgen Schmidhuber. 2024. MetaGPT: Meta Programming for A Multi-Agent Collaborative Framework. In *The Twelfth International Conference on Learning Representations (ICLR 2024)*, Vienna, Austria. https://openreview.net/forum?id=VtmBAGCN7o
- [23] Manas Joglekar, Jeremy Chen, Gabriel Wu, Jason Yosinski, Jasmine Wang, Boaz Barak, and Amelia Glaese. 2025. Training LLMs for Honesty via Confessions. arXiv:2512.08093 [cs.LG]
- [24] Sham M. Kakade. 2001. A Natural Policy Gradient. In *Advances in Neural Information Processing Systems*, Vol. 14. MIT Press, Cambridge, MA, USA, 1531–1538.
- [25] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations (ICLR 2015)*, San Diego, CA, USA. https://arxiv.org/abs/1412.6980
- [26] Jakub Grudzien Kuba, Ruiqing Chen, Muning Wen, Ying Wen, Fanglei Sun, Jun Wang, and Yaodong Yang. 2022. Trust Region Policy Optimisation in Multi-Agent Reinforcement Learning. In *The Tenth International Conference on Learning Representations (ICLR 2022)*, Virtual Event. https://openreview.net/forum?id=EcGGfKNTxdj
- [27] Bruce W. Lee, Chen Yueh-Han, and Tomek Korbak. 2026. Training Agents to Self-Report Misbehavior. arXiv:2602.22303 [cs.LG]
- [28] Wilson Lin. 2026. Scaling Long-Running Autonomous Coding. Cursor blog. https://cursor.com/blog/scaling-agents
- [29] Anuj Mahajan, Tabish Rashid, Mikayel Samvelyan, and Shimon Whiteson. 2019. MAVEN: Multi-Agent Variational Exploration. In *Advances in Neural Information Processing Systems*, Vol. 32. Curran Associates, Inc., Red Hook, NY, USA, 9460–9471. doi:10.52202/068431-0687
- [30] OpenAI. 2026. The Hugging Face Incident and the Road Ahead. https://openai.com/index/hugging-face-incident-and-the-road-ahead/
- [31] Davide Paglieri, Logan Cross, Tim Genewein, Joel Z. Leibo, Nenad Tomasev, and Alexander Sasha Vezhnevets. 2026. A Case Study on Emergent Cheating and Whistleblowing in Autonomous Research Swarms. arXiv:2609.04170 [cs.AI]
- [32] Jean-Claude Picard. 1976. Maximal Closure of a Graph and Applications to Combinatorial Problems. *Management Science* 22, 11 (1976), 1268–1272. doi:10.1287/mnsc.22.11.1268
- [33] A. Mitchell Polinsky and Steven Shavell. 2000. The Economic Theory of Public Enforcement of Law. *Journal of Economic Literature* 38, 1 (2000), 45–76. doi:10.1257/jel.38.1.45
- [34] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal Policy Optimization Algorithms. arXiv:1707.06347 [cs.LG]
- [35] Joar Skalse, Nikolaus H. R. Howe, Dmitrii Krashenninnikov, and David Krueger. 2022. Defining and Characterizing Reward Hacking. In *Advances in Neural Information Processing Systems*, Vol. 35. Curran Associates, Inc., Red Hook, NY, USA, 9460–9471. doi:10.52202/068431-0687
- [36] Richard S. Sutton, David McAllester, Satinder Singh, and Yishay Mansour. 2000. Policy Gradient Methods for Reinforcement Learning with Function Approximation. In *Advances in Neural Information Processing Systems*, Vol. 12. MIT Press, Cambridge, MA, USA, 1057–1063.
- [37] J. K. Terry, Nathaniel Grammel, Sanghyun Son, Benjamin Black, and Aakriti Agrawal. 2020. Revisiting Parameter Sharing in Multi-Agent Deep Reinforcement Learning. arXiv:2005.13625 [cs.LG]
- [38] Ermo Wei and Sean Luke. 2016. Lenient Learning in Independent-Learner Stochastic Cooperative Games. *Journal of Machine Learning Research* 17, 84 (2016), 1–42. https://jmlr.org/papers/v17/15-417.html
- [39] Ronald J. Williams. 1992. Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning. *Machine Learning* 8, 3–4 (1992), 229–256. doi:10.1007/BF00992696
- [40] Jiachen Yang, Ang Li, Mehrdad Farajtabar, Peter Sunehag, Edward Hughes, and Hongyuan Zha. 2020. Learning to Incentivize Other Learning Agents. In *Advances in Neural Information Processing Systems*, Vol. 33. Curran Associates, Inc., Red Hook, NY, USA, 15208–15219. https://proceedings.neurips.cc/paper/2020/hash/ad7ed5d47b9baceb12045a929e7e2f66-Abstract.html
- [41] Jiachen Yang, Ethan Wang, Rakshit Trivedi, Tuo Zhao, and Hongyuan Zha. 2022. Adaptive Incentive Design with Multi-Agent Meta-Gradient Reinforcement Learning. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, USA, 1436–1445.
- [42] Chao Yu, Akash Velu, Eugene Vinititsky, Jiaxuan Gao, Yu Wang, Alexandre Bayen, and Yi Wu. 2022. The Surprising Effectiveness of PPO in Cooperative Multi-Agent Games. In *Advances in Neural Information Processing Systems*, Vol. 35. Curran Associates, Inc., Red Hook, NY, USA, 24611–24624. doi:10.52202/068431-1787
- [43] Runyu Zhang, ZhaoLin Ren, and Na Li. 2024. Gradient Play in Stochastic Games: Stationary Points, Convergence, and Sample Complexity. *IEEE Trans. Automat. Control* 69, 10 (2024), 6499–6514. doi:10.1109/TAC.2024.3387208
- [44] Wenshuai Zhao, Yi Zhao, Zhiyuan Li, Juho Kannala, and Joni Pajarinen. 2024. Optimistic Multi-Agent Policy Gradient. *Proceedings of the 41st International Conference on Machine Learning, PMLR* 235 (2024), 61186–61202. https://proceedings.mlr.press/v235/zhao24v.html

Numbers with an S prefix refer to the appendices. Other numbers refer to the main text. The accompanying code and retained outcomes are in the ancillary archive reproducibility.zip.

A Proofs and certificate construction

The results in this appendix concern one fixed context, positive task benefits, nonnegative base costs and action draws that are independent given the group probabilities. Every group is nonempty and every initial logit is finite, so every initial probability lies in $(0, 1)$. In part (a) of Theorem 1, the loads may use an upper bound on the context benefits, because lowering a benefit only lowers the positive terms of the derivative. Part (b) and Proposition S1 need the loads at the actual benefits.

A.1 Strict elimination and exact learning

All groups update at every step, so the proof has to show that a group converges even though the loads it faces shrink only in the limit. Write $\partial_g J = \partial J / \partial p_g$. The exact updates covered by Theorem 1 are

$$\begin{aligned} \theta_g^{t+1} &= \theta_g^t + \eta \lambda_g p_g^t (1 - p_g^t) \partial_g J(p^t), & \text{gradient ascent,} \\ \theta_g^{t+1} &= \theta_g^t + \eta \lambda_g \partial_g J(p^t), & \text{natural-gradient ascent,} \end{aligned} \quad (\text{S1})$$

where $\eta > 0$ is fixed and finite and every $\lambda_g > 0$ is fixed. For one Bernoulli draw, the logit score is $s - p_g$ and the Fisher factor is $p_g(1 - p_g)$. Under the joint product distribution, the score of group g is $\sum_{i \in g} (s_i - p_g)$, its variance is $|g|p_g(1 - p_g)$, and the Fisher entries between groups are zero. Dividing the logit gradient by this factor gives Equation (S2) with $\lambda_g = 1/|g|$, and using one Bernoulli Fisher factor per group gives $\lambda_g = 1$. The theorem covers both. Neither update includes momentum, entropy regularization or parameters shared between groups.

PROOF OF THEOREM 1(A). Choose a removal order $g_1, \dots, g_G$ and let $S_j = \{g_j, \dots, g_G\}$. Strictness gives a positive slack $\epsilon_j = D_{g_j} - L_{g_j}(S_j)$ for each removal.

For g_1 , every product multiplying $m_{e_{g_1}} b_e$ in Equation (4) lies in $[0, 1]$, so $\partial_{g_1} J(p) \leq -\epsilon_1$ throughout the probability cube. Under Equation (S2), the logit of g_1 decreases by at least $\eta \lambda_{g_1} \epsilon_1$ per step and tends to $-\infty$. Under Equation (S1), the logit is strictly decreasing. If it had a finite limit, $p_{g_1}(1 - p_{g_1})$ would have a positive limit, and the increments would eventually stay below a fixed negative number, which is impossible for a convergent sequence. Thus $\theta_{g_1} \rightarrow -\infty$ and $p_{g_1} \rightarrow 0$ under either update.

Suppose $g_1, \dots, g_{j-1}$ tend to zero. Split the positive derivative terms of g_j into tasks whose witnesses all lie in S_j and tasks with a witness in an earlier group. The first collection is bounded by $L_{g_j}(S_j)$. Every term in the second contains a positive power of an earlier probability. All other factors are at most one and the task set is finite, so these terms tend to zero uniformly over the other probabilities. Hence, after some finite time,

$$\partial_{g_j} J(p^t) \leq -\epsilon_j/2.$$

The logit is finite at that time, and the argument for g_1 then gives $p_{g_j} \rightarrow 0$. Induction proves the claim. The step size needs no upper bound, because the proof uses only eventual derivative signs and finite logits. $\square$

The removal order does not affect which groups survive. Removing a group only lowers every remaining $L_g(S)$, so a group that becomes eligible stays eligible after further removals. Iterating this monotone rule to closure gives the same surviving set for any choice among eligible groups, and positive prescribed margins preserve this property. We call this set the residue of strict elimination.

PROOF OF THEOREM 1(B). Let S be the residue. For $g \in S$, split the positive terms of $\partial_g J$ in Equation (4) into a first collection, the tasks whose witnesses all lie in groups of S , and a second collection of all other tasks. The terms of the second collection are nonnegative. Each product in the first collection has total exponent at most $M = \max_e |W_e|$. If $p_h \geq 1 - \beta$ for every $h \in S$, each such product is therefore at least $(1 - \beta)^M$, and

$$\partial_g J(p) \geq (1 - \beta)^M L_g(S) - D_g.$$

Because $D_g < L_g(S)$ for every $g \in S$ and S is finite, some $\beta \in (0, 1)$ makes the right side at least a positive ϵ for every $g \in S$. Fix such a β and start at an interior point with $p_g > 1 - \beta$ for every $g \in S$. While the iterates stay in this set, every coordinate of S has derivative at least ϵ , so under either update its logit increases and its probability stays above $1 - \beta$. By induction over steps, the iterates never leave the set. Under Equation (S2), each logit in S increases by at least $\eta \lambda_g \epsilon$ per step and tends to $+\infty$. Under Equation (S1), each logit in S is increasing, and a finite limit would leave $p_g(1 - p_g)$ with a positive limit and the increments above a fixed positive number, which is impossible. Hence $p_g \rightarrow 1$ for every $g \in S$.

For the removed groups, take the removal order $g_1, \dots, g_r$ of strict elimination and let S_j contain g_j , the later removed groups and S . The induction in the proof of part (a) applies unchanged. It bounds the positive derivative terms of g_j by $L_{g_j}(S_j)$ plus terms that contain a positive power of an earlier probability, and the bound holds uniformly over every other probability. The groups in S therefore do not affect it. Hence $p_g \rightarrow 0$ for every removed group. $\square$

PROPOSITION S1 (STRICT RESIDUE). *Under the conditions of Theorem 1(b), the corner with $p_g = 1$ for $g \in S$ and $p_g = 0$ otherwise is a strict local maximum of J on the probability cube.*

PROOF. Every coordinate in S has derivative $L_g(S) - D_g > 0$ at the corner. For a removed coordinate, the derivative terms that involve another removed coordinate vanish, and the terms supported on that coordinate and S are bounded by its load when it was removed. Its derivative is therefore strictly negative. By continuity, these signs hold in a small enough rectangular neighborhood of the corner. Moving any coordinate from the corner into the cube decreases J , and the signs persist throughout the neighborhood, so the corner is a strict constrained local maximum. $\square$

The strict inequalities are needed for Theorem 1(b) and this proposition. A zero derivative at a coordinate in S leaves the higher-order behavior unresolved. Local attraction under small parameter changes also does not show that a group has no profitable finite change in its action probability, so we do not use the residue as a Nash-equilibrium certificate.

A.2 Order costs and optimization

The audit budget buys deterrence for individual agents, and the certificate needs that deterrence when the agent's group is removed.

PROOF OF PROPOSITION 2 AND EQUATION (6). Fix an order π . A task remains in $L_g(S)$ until the first group that contains one of its witnesses is removed. It contributes $m_{eg}b_e$ to the load charged at that removal and nothing to later removals. With zero base costs and no binding caps, summing the required deterrence over the order gives

$$\sum_g L_g(S_g) = \sum_e b_e m_{e, \text{first}_{\pi}(e)}.$$

Arbitrarily small positive slacks approach this infimum for the strict inequalities, and multiplying by the common unit price α gives the audit budget. Fixed positive margins add deterrence $\sum_g \delta_g$ and audit budget $\alpha \sum_g \delta_g$.

For every task, the first-group multiplicity is an integer between one and $|W_e|$. Multiplying by the positive benefit and summing gives the bounds. Equality at B holds exactly when every first-group multiplicity is one. For a fully shared policy, that multiplicity equals k for every task when every task has k witnesses. $\square$

For heterogeneous prices and caps, write $c_g = \sum_{i \in g} c_i$. If group g must buy x units of extra deterrence, its minimum cost is

$$\phi_g(x) = \min_{0 \leq a_i \leq 1} \left\{ \sum_{i \in g} \kappa_i a_i \mid \sum_{i \in g} F_i a_i \geq x \right\}. \quad (\text{S3})$$

Set $\phi_g(x) = 0$ for $x \leq 0$ and $\phi_g(x) = +\infty$ above the available deterrence. Buying from agents in increasing order of κ_i/F_i solves this fractional knapsack problem.

For fixed margins, let $\Phi(S)$ be the minimum remaining certificate cost. Then

$$\Phi(\emptyset) = 0, \quad \Phi(S) = \min_{g \in S} [\phi_g(L_g(S) + \delta_g - c_g) + \Phi(S \setminus \{g\})]. \quad (\text{S4})$$

Every complete certificate has a removal order, and every finite-cost order in this recurrence yields a certificate, so the recurrence is exact for this certificate class. It visits 2^G subsets with at most G transitions each, besides the load and purchase-cost evaluations. The implementation uses the exact recurrence for $G \leq 10$. For larger G , it chooses a feasible next group greedily by purchase cost, and the resulting costs are feasible but not proven minima.

A strong coloring of the co-witness graph places every task's witnesses in distinct groups, so every order has first-group multiplicity one. The condition $C(\pi) = B$ is weaker. It is the unique-minimum coloring condition on the witness graph, which becomes unique-maximum coloring when the order is reversed [11], and it requires only that every task have a unique witness in its first group. In a k -partite task family, one entire role can form the first group and the remaining roles a second group without changing the B term. So a minimum-cost certificate need not give every witness role its own group. The main study uses the co-witness coloring and does not optimize the number of groups.

A.3 The balanced family

The balanced family of Section 3.3 has q^2 tasks and kq agents, with $2 \leq k \leq q$ and q prime. Every agent witnesses q tasks. For two

different roles t and t' , the difference $t - t'$ is nonzero in $\mathbb{F}_q$, so the equations $u + tv = w$ and $u + t'v = w'$ have exactly one solution for any pair of agent labels w, w' . The co-witness graph is therefore complete multipartite. Random relabeling changes only its visible indices.

PROOF OF COROLLARY 3. With role groups, any first role has load B . Its base deterrence is $\rho B/k$ and its prescribed margin is also $\rho B/k$, so it must buy at least B units of fine deterrence. Every agent pays $n/(\gamma B)$ per unit of deterrence, so this purchase costs at least n/γ , and auditing each member of that role with probability k/γ attains this cost. Once the role is removed, every task disappears, and the remaining roles' base costs exactly supply their prescribed margins, so no further auditing is needed.

For the fully shared policy, the load is kB , the base deterrence is ρB and the margin is ρB . The group must buy at least kB units of fine deterrence, which costs kn/γ , and auditing every agent with probability k/γ attains this cost. The assumption $\gamma \geq k$ makes both allocations respect the probability cap. Without margins, the strict inequalities require the first role to buy more than $B - \rho B/k$ units and the fully shared group more than $kB - \rho B$ units. For $\rho < k$ the infima are therefore $(n/\gamma)(1 - \rho/k)$ and $(n/\gamma)(k - \rho)$, whose ratio is again k . $\square$

The global optimum and the local obstruction can also be checked directly. For $k \geq 2$, $Bp^k - Dp$ is convex on $[0, 1]$. When $D > B$, its larger endpoint value is $J(0) = 0 > J(1) = B - D$, so reporting is the global maximum. If also $D < kB$, the derivative at $p = 1$ is positive and silence is locally attracting under ascent. Solving $kBp^{k-1} = D$ gives Equation (3).

An explicit time bound holds for normalized natural-gradient updates with role groups in the balanced family. Scale each group derivative by k/B , and let role 1 be the audited role. Its logit increment is then at most $-\eta\rho$. Let $\ell(z) = \log(z/(1-z))$, write $[z]_+ = \max\{z, 0\}$, and set $q_0 = \min\{1/2, \rho/(2k)\}$. The audited role's probability is at most q_0 after

$$T_1 = \left\lceil \frac{[\ell(q_0) - \ell(q_0)]_+}{\eta\rho} \right\rceil$$

steps. For any other role h , the positive derivative is at most Bp_1 , and its base deterrence is $\rho B/k$. After T_1 , its normalized derivative is at most $-\rho/2$. Before then its normalized derivative is at most k , so $\theta_h^{T_1} \leq \theta_h^0 + \eta k T_1$. For a target probability $\epsilon \in (0, 1)$, a valid additional bound is

$$T_2 = \max \left\{ \left\lceil \frac{[\ell(q_0) - \ell(\epsilon)]_+}{\eta\rho} \right\rceil, \max_{h \neq 1} \left\lceil \frac{[\theta_h^0 + \eta k T_1 - \ell(\epsilon)]_+}{\eta\rho/2} \right\rceil \right\}.$$

All role probabilities are at most ϵ after $T_1 + T_2$ steps. The second term of T_2 shows the cost of learning in sequence. While the audited role is still learning, the other roles' logits can rise by up to ηk per step. Each step the audited role takes adds $2k/\rho$ steps to the bound on their remaining time. The bound is conservative and applies to exact updates, so it gives no sample complexity for PPO.

A.4 The coalition guard

If some set of silent agents still earned more than it paid, an architecture that gives each agent its own silence probability would fail for reasons that have nothing to do with learning. The coalition

guard rules this out. For a set $T \subseteq V$ of silent agents, while all other agents report, the deterministic team reward is

$$U(T) = \sum_{e: W_e \subseteq T} b_e - \sum_{i \in T} d_i. \quad (\text{S5})$$

A maximum-closure graph has a positive-weight vertex for each task, a negative-weight vertex for each agent, and an arc from each task to each of its witnesses, so selecting a task requires selecting its witnesses. Because all task weights are positive, the maximum closure weight equals $\max_T U(T)$, and the standard source-sink reduction computes it as a minimum cut [32]. A large enough finite capacity can replace the infinite arcs.

The study applies this check, with benefits multiplied by 1.02, to the certifying allocation that the search finds for witness groups. If that allocation fails the check, the study computes a flow allocation that deters every coalition and takes the coordinatewise maximum of the two allocations. No deterrence decreases, so the group certificate survives, the caps still hold, and every increased coordinate is charged to the reference budget. The guard adds deterrence on 18 of the 24 general graphs, including all 12 whose certificate costs are exact. The addition is between 6.47% and 34.07% of the reference budget, with median 24.27%.

The same guard is applied when constructing the later shuffled allocation and the allocations on corrupted witness graphs, each on the graph used for its construction and before its audit budget is adjusted. Budget alternatives are evaluated without further repair, so scaling a guarded allocation down can break the certificate or the guard’s condition. Equation (6) describes the group certificate before the guard is applied.

A randomized joint policy is a distribution over deterministic coalitions, so an allocation that leaves every coalition unprofitable makes universal reporting optimal for the same team reward. The guard makes reporting the best policy for every architecture, and it does not imply that a local learner will reach it.

B Experimental protocol

The theory concerns exact derivatives of group probabilities, while the study obtains actions, rewards and updates from neural-network policies. The fixed graph bank, matched initialization and paired random streams hold everything else constant when the grouping or the audits change.

B.1 Graphs and costs

The general graphs cross four families, two witness widths and three graph seeds. Every general graph has $n = 64$ agents and 128 tasks. Each task’s benefit and harm weight are drawn uniformly from $[0.5, 1.5]$. Completion fractions are unweighted, and the harm-weighted version is a secondary diagnostic. Base costs are proportional to independent draws from $[0.5, 1.5]$ and are normalized to sum to $0.5B$. Fines F_i are $16B/n$ times independent draws from $[0.75, 1.25]$, and per-audit costs κ_i are independent draws from $[0.5, 1.5]$.

Uniform graphs sample each witness set without replacement from all agents. Ring graphs take k consecutive agents modulo 64 from a uniformly drawn start. Modular graphs draw all witnesses from one random 16-agent module with probability 0.85 and

from the whole population otherwise. Hub graphs sample without replacement with agent weights proportional to $1/\sqrt{i+1}$.

The balanced and rewired families use $q = 11$ and widths four and eight, giving 44 or 88 agents and 121 tasks. Balanced graphs use unit benefits and harm weights, equal per-audit costs and the costs of Corollary 3 with $\rho = 0.5$ and $\gamma = 16$. Rewired graphs replace each witness set with probability 0.20 by a uniformly drawn set of the same size and draw benefits and harm weights from $[0.5, 1.5]$. Their base costs have independent weights in $[0.6, 1.4]$, normalized to $0.5B$. Role labels are randomly permuted before coloring.

Graph seeds are 914101, 914117 and 914131, and each graph has two training replicates. The context, pretraining, recovery, shuffle and corruption seeds are separate deterministic hashes of their declared arguments. The graph bank keeps all 36 graphs. No graph is excluded because of its training result or an infeasible certificate, and every scheduled design is feasible.

The coloring is deterministic DSATUR on the co-witness graph, with saturation degree, graph degree and agent index as successive tie-breakers [9]. Shuffling the resulting labels preserves every group’s size, so the primary comparison matches the number and dimensions of the group networks.

B.2 Contexts and observed features

Legitimate contexts keep a policy from passing the endpoint by reporting everything. Every rollout includes all eight contexts with equal weight. Each episode is a one-step game with no state transition or temporal credit assignment. Contexts 0–3 are unsafe and contexts 4–7 are legitimate. For each context and task, a fixed multiplier in $[0.65, 1]$ scales the benefit. Agents receive the context’s one-hot vector and the mean and standard deviation of its multipliers. They do not receive witness membership, the allocation or an unsafe flag, although the context identities let them learn different responses to the fixed unsafe and legitimate contexts.

Each agent draws one action per episode, and that action applies to all the tasks it witnesses. Audit randomness is drawn in every context and every arm to keep the random streams paired, but audit events are set to zero in legitimate contexts. The audit budget is reported per unsafe context, $\sum_i \kappa_i a_i$, and its average over all eight contexts is half of that, because no audit cost arises in legitimate contexts.

B.3 Networks and initialization

For group g and context features x , the base network computes

$$h_g(x) = \tanh(W_g x + b_g), \quad (\text{S6})$$

$$z_g(x) = v_g^\top h_g(x) + u_g, \quad (\text{S7})$$

$$\theta_g(x) = 12 \tanh(z_g(x)/12), \quad p_g(x) = \sigma(\theta_g(x)). \quad (\text{S8})$$

The hidden layer has 32 units. With ten input features, each group network has 385 parameters. Backpropagation includes the derivative of the bounded logit. The fully shared architecture has one such network. The scalar-offset architecture adds a trainable value per agent before bounding the logit. The agent-ID architecture adds trainable weights from a one-hot identity input to the shared hidden layer. Independent networks give each agent a complete network of its own. On general graphs, these three architectures

have 449, 2,433 and 24,640 parameters, against 385 for the fully shared network.

Pretraining starts at silence probability 0.75 and runs 512 audit-free rollouts. One pretrained network supplies the same starting policy to every matched recovery arm, and each group network receives a copy of its weights. New scalar offsets and identity weights start at zero. The actor parameters are kept, and optimizer moments and the value baseline are reset. All pretraining outcomes are included. The head resets of Appendix B.5 are the declared exception to this matching.

B.4 Sampled PPO

The team reward is divided by B/n . Each rollout draws 64 one-step episodes per context and performs four epochs of the clipped PPO update. A scalar value estimate for each context moves toward the batch mean at rate 0.2. Advantages use the previous estimate and are standardized over contexts and episodes, with the standard deviation floored at 10^{-6} .

For the current marginal probability p_i and the behavior probability p_i^{old} , the sampled marginal likelihood ratio is

$$r_i = \frac{p_i^{s_i} (1 - p_i)^{1-s_i}}{(p_i^{\text{old}})^{s_i} (1 - p_i^{\text{old}})^{1-s_i}}.$$

The actor surrogate averages

$$\min\{r_i \hat{A}, \text{clip}(r_i, 0.8, 1.2) \hat{A}\} + 0.005 H(\text{Bernoulli}(p_i))$$

over agents, contexts and episodes, with the same normalization in every architecture. Adam uses learning rate 0.003, moment coefficients 0.9 and 0.999, numerical constant 10^{-8} , and a global gradient-norm cap of 0.5. Exact expected rewards and completion rates are used only for evaluation. These settings define the PPO variant we ran, and we do not claim that it matches other multi-agent PPO implementations.

The base condition runs 4,096 recovery rollouts of 512 episodes each, with sampling streams paired across matched conditions. Records keep the job specification, source and input fingerprints, initial-policy fingerprint, final actor parameters, evaluation trace and a cumulative sample hash. Continuation records keep the full optimizer, baseline and random-generator state.

B.5 Budgets and additional experiments

The budgets are $\{0.5, 1, 2, 4, 8\}$ times the reference budget, capped at the cost of auditing every agent. An allocation that costs more than its budget is scaled down proportionally. One that costs less spends the difference by raising every agent’s audit probability equally, capped at one, so no coordinate decreases. Uniform and coalition audits receive the same audit budget. Coalition audits keep witness groups and maximize the smallest ratio of deterrence to completed benefit over individual coalitions, and any budget left after that ratio is fixed is spread uniformly. Each allocation is evaluated at its assigned budget with no later repair. Increases preserve the certificate for witness groups, and decreases need not.

Head resets keep the learned first layer, set the output weights to zero, and set the output bias to $12 \arctanh(\log[p_0/(1-p_0)]/12)$ for silence probability $p_0 \in \{0.75, 0.95\}$, compensating for the bounded logit. They apply to witness, shuffled, fully shared, scalar-offset and

agent-ID architectures, and they deliberately change the starting behavior. Both reset levels were fixed before study training.

The optimism experiment replaces the standardized advantage $\hat{A}$ with $\max\{\omega \hat{A}, \bar{A}\}$ for $\omega = 0$ or 0.5 in witness and shuffled arms, on the first two graph seeds of every family and width. The transformation is the one proposed by Zhao et al. [44], whose full method was evaluated in other environments with other implementation choices.

The continuation experiment uses the first graph seed of every family and width, chosen before training. Witness, shuffled and scalar-offset arms continue to 16,384 rollouts, with the 4,096-rollout prefix preserved. No observed outcome selects a continuation.

For witness error, a nominal 10% or 25% of tasks have their whole witness set replaced by a different random set of the same width. The number of replaced tasks is the nominal fraction times the task count, rounded to the nearest integer. Benefits, costs and fines are unchanged. The auditor builds its grouping and guarded allocation on the corrupted graph and then adjusts the allocation to the true graph’s reference budget. The true graph supplies both the budget and the environment and never repairs the corrupted grouping or allocation. The two comparison arms are a shared network with scalar offsets and uniform audits with the corrupted grouping, both at the same audit budget. Table S1 lists the main study’s experiments with their record counts.

Table S1: The fixed main-study schedule. The 4,080 records include pretraining. Continuations extend selected base runs and are not independent replicates. All records completed.

Experiment	Records
Common pretraining	72
Seven methods at five audit budgets	2,520
Independent-network architecture	72
Five architectures at two head resets	720
Two groupings at two optimism levels	192
Three methods under two witness-error levels	432
Three methods continued to 16,384 rollouts	72
Total	4,080

C Endpoints, inference and chronology

Evaluation uses expected task completion under each policy’s joint action distribution, so the endpoints carry no sampling noise from evaluation. For a policy in unsafe context x , let v_x be the expected number of completed tasks divided by $|E|$. Legitimate completion uses the corresponding fraction in legitimate contexts. With independent actions, a task’s completion probability is the product of its witnesses’ silence probabilities. With a common draw, it is the single common silence probability.

Evaluation occurs at initialization and every 64 rollouts. With horizon T , the endpoint averages checkpoints with rollout index strictly greater than $0.75T$. For $T = 4096$, these are the 16 checkpoints from 3136 through 4096. Unsafe completion averages over the four unsafe contexts and then over these checkpoints, and legitimate completion does the same over the four legitimate contexts. Joint recovery requires a final-quarter unsafe mean below 0.01 and

a legitimate mean of at least 0.90. Joint recovery does not require each context or checkpoint to meet those thresholds separately.

For general graph j and training replicate $r \in \{1, 2\}$, let v_{jr}^S and v_{jr}^W be the shuffled and witness endpoints. The paired graph effect is

$$d_j = \frac{1}{2} \sum_{r=1}^2 (v_{jr}^S - v_{jr}^W).$$

If S contains the eight general family-width strata, each with three graphs, the primary estimator is

$$\hat{\Delta} = \frac{1}{8} \sum_{s \in S} \frac{1}{3} \sum_{j \in s} d_j.$$

The percentile interval resamples three graphs with replacement within each stratum, using 10,000 resamples and seed 914901. The interval describes graph variation within the observed strata, conditional on the two trained replicates averaged for each graph, and it does not estimate variation over new training seeds. Episodes, checkpoints and replicates are never treated as independent graphs. Three graphs per stratum limit how finely the interval can resolve that variation.

All architecture, budget, reset, optimism and witness-error experiments are secondary and descriptive. A budget curve’s smallest successful tested budget is reported only on a complete grid, curves without a successful point are right-censored, and nonmonotone outcomes are kept. Neither grid minima nor certificate costs are treated as the minimum audit budget over learning algorithms. The certification splits of Table 3, the round counts of Table S3 and the threshold calculations of Appendix E were added after the results. The splits use only quantities fixed before training, including the context multipliers.

The main schedule was fixed before its training run, after earlier exploratory studies and development pilots whose results are not pooled into any estimate. The returned main run contains all 4,080 scheduled records. Two later questions led to new runs on the same general graphs. The allocation control adds 96 shuffled-allocation runs and 192 same-host repetitions of the original arms. The common-draw control adds 48 common-draw runs and 48 repetitions of the independent-draw arm. Each includes every general graph and both training replicates, with no outcome-dependent stopping or deletion. Both remain exploratory, because they reuse the graphs of the main study.

D Results by family and budget

The primary estimate averages over graph families, so it can hide which witness structures benefit from grouping. Table S2 separates the general families from the balanced and rewired ones at the reference budget, and Figure S1 shows every paired graph effect together with its two training replicates. Witness groups have a positive mean effect in every general family. Shuffled groups do better on four graphs, although on one the difference is below 0.01 points and both arms round to 99.99% unsafe completion. Table S4 gives the complete budget aggregates.

Table S3 splits the witness-group runs of Table S4 under the reference allocation and under coalition audits by the number of elimination rounds that each allocation needs to certify witness groups on each graph. At four times the reference budget, the Fewer

row contains exactly the six width-8 hub and uniform graphs, and at eight times it contains the three width-8 hub graphs. The six width-8 hub and uniform graphs are also the only graphs whose reference allocation comes from the greedy order without guard additions, so the table cannot attribute outcome differences to the round counts alone.

D.1 Allocation control

The primary comparison uses an allocation designed for witness groups, so shuffled groups might fail only because their allocation was designed for another grouping. The allocation control constructs an allocation for shuffled groups on each graph with the same search, fines, margin and coalition guard as the reference allocation. Its audit budget averages 1.142 times the reference budget over the 24 general graphs, and it is not an exact optimum on graphs that use the greedy branch. Adjusted to the reference budget by the rule of Appendix B.5, the redesigned allocation certifies shuffled groups on 14 of the 24 graphs. On 18 of the 24 graphs, it also leaves every silent coalition with a profit below 10^{-12} at the actual benefits. The reference allocation certifies shuffled groups on one graph and witness groups on all 24.

At the reference budget, the witness advantage over shuffled groups with the redesigned allocation is 26.03 points, with exploratory interval [11.77, 39.93]. At twice the reference budget it is 4.78 points, with interval [−3.98, 13.55]. The intervals use paired graph units, stratified resampling and seed 9152026, and they are not adjusted for multiple comparisons. Table S5 keeps the repetitions with the reference allocation so that the redesigned arm is compared on the same host.

E Training, initialization and witness information

These experiments ask whether recovery changes when training starts farther from silence, uses different advantage weights or runs longer, or when the auditor designs the allocation from wrong witness information. They include the balanced and rewired families as stated in each caption, so their denominators differ from the 48-run summaries of the main text. Both completion measures are kept, because lower unsafe completion can come with a large loss of legitimate completion.

A reset gives every agent the same silence probability p_0 , so every arm starts at p_0^k unsafe and legitimate completion (Table S7). For the fully shared network, the unsafe-context stationary point p_* of Equation (3) lies between 0.76 and 0.85 over all 36 reference allocations and unsafe contexts. The analogous legitimate-context point $(\sum_i c_i / (kB_x))^{1/(k-1)}$ lies between 0.689 and 0.694 at width 8 and between 0.529 and 0.539 at width 4. The output weights start at zero, so the exact gradient on the shared output bias has the sign of the derivative summed over all eight contexts. At $p_0 = 0.95$, this sum favors silence in every graph. At $p_0 = 0.75$, it favors silence in every width-4 graph and reporting in every width-8 graph. The final-quarter outcomes of the fully shared network agree with these initial gradient signs in 140 of its 144 reset runs (Table S6). It ends silent in both kinds of context in every width-4 run and in every width-8 run from 0.95. From 0.75 at width 8, it ends reporting in both kinds of context in 32 of 36 runs and silent in the other

Table S2: Reference-budget results by family and witness width. Each row contains three graph seeds and two training replicates. W denotes witness groups and S shuffled groups. Completion columns are percentages, and differences are percentage points. Balanced and rewired rows are secondary and are excluded from the primary estimate.

Family	k	Unsafe W	Unsafe S	$S - W$	Joint recovery W	Joint recovery S
Uniform	4	43.32	78.43	35.11	0/6	0/6
Uniform	8	86.39	88.02	1.62	0/6	0/6
Ring	4	31.36	63.09	31.73	4/6	0/6
Ring	8	17.19	99.99	82.80	4/6	0/6
Modular	4	31.48	81.77	50.29	0/6	0/6
Modular	8	84.50	98.37	13.86	0/6	0/6
Hub	4	37.39	82.48	45.08	1/6	0/6
Hub	8	77.96	86.19	8.23	0/6	0/6
Balanced	4	33.33	99.98	66.65	4/6	0/6
Balanced	8	33.33	100.00	66.66	4/6	0/6
Rewired	4	23.03	99.96	76.93	2/6	0/6
Rewired	8	81.51	99.99	18.48	0/6	0/6

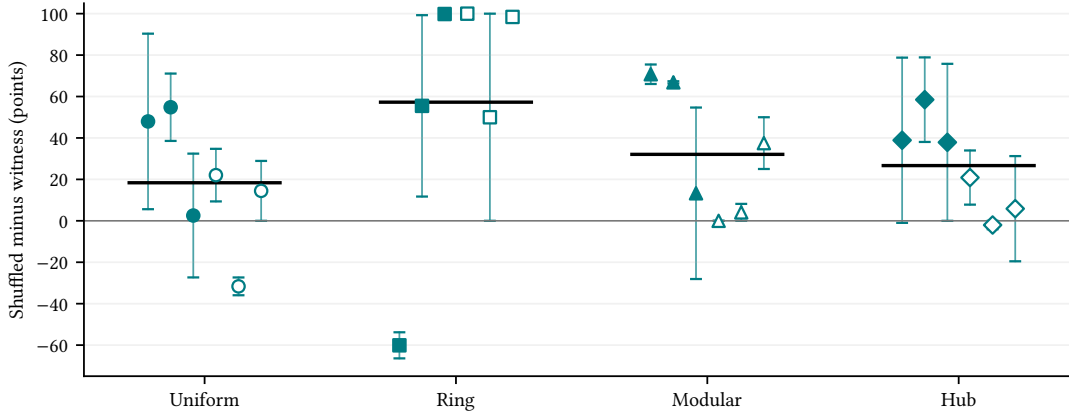


Figure S1: The 24 paired general-graph effects at the reference budget. Each marker is the mean of a graph’s two training replicates, and the ticks at the ends of the vertical line through it are the effects in the replicates themselves. Within a replicate, both arms start from the same pretrained network. Filled markers have witness width four and open markers width eight. Horizontal black segments are family means, and positive values favor witness groups. The graph means are the units of the primary bootstrap.

Table S3: Coalition audits against the reference allocation for witness groups on the 24 general graphs. Graphs are split by whether coalition audits need fewer, equal or more elimination rounds than the reference allocation, or do not certify witness groups, which the reference allocation certifies on every graph at these budgets. R denotes the reference allocation and C coalition audits. Each row covers both training replicates of its graphs, and the unsafe columns are mean percentages. The split is post hoc and coincides with the graph subsets named in the text.

Budget	Rounds for C	Graphs	Joint recovery R	Joint recovery C	Unsafe R	Unsafe C
2	Fewer	10	3/20	2/20	61.36	48.78
2	Equal	6	6/12	6/12	17.45	11.08
2	More	3	3/6	2/6	7.65	30.21
2	Uncertified	5	6/10	1/10	33.36	87.22
4	Fewer	6	0/12	5/12	74.03	12.24
4	Equal	18	29/36	32/36	9.09	1.48
8	Fewer	3	3/6	6/6	14.35	0.13
8	Equal	10	19/20	20/20	0.12	0.00
8	More	11	22/22	22/22	0.04	0.04

Table S4: Complete budget aggregates on the general families. Each row contains 48 runs. Budget is a multiple of the reference budget, subject to the full-audit cap. Uniform and coalition rows keep witness groups. Certified counts the graphs on which the audits certify the architecture’s groups, and it does not apply to scalar offsets or agent IDs, which do not form a grouping. Completion columns are percentages, and all action draws are independent.

Method	Budget	Certified	Unsafe	Legitimate	Joint recovery
Witness groups	0.5	0/24	99.99	100.00	0/48
Witness groups	1	24/24	51.20	99.99	9/48
Witness groups	2	24/24	37.84	99.98	18/48
Witness groups	4	24/24	25.33	99.98	29/48
Witness groups	8	24/24	1.86	99.97	44/48
Shuffled groups	0.5	0/24	98.51	100.00	0/48
Shuffled groups	1	1/24	84.79	99.99	0/48
Shuffled groups	2	10/24	56.75	99.99	6/48
Shuffled groups	4	18/24	30.94	99.97	23/48
Shuffled groups	8	24/24	0.53	99.98	45/48
Fully shared	0.5	0/24	100.00	100.00	0/48
Fully shared	1	0/24	99.99	100.00	0/48
Fully shared	2	0/24	99.97	99.99	0/48
Fully shared	4	6/24	57.78	100.00	19/48
Fully shared	8	13/24	27.08	100.00	35/48
Shared + scalar offsets	0.5	–	100.00	100.00	0/48
Shared + scalar offsets	1	–	91.66	100.00	4/48
Shared + scalar offsets	2	–	72.78	99.99	10/48
Shared + scalar offsets	4	–	58.67	99.99	19/48
Shared + scalar offsets	8	–	19.49	100.00	38/48
Shared + agent IDs	0.5	–	97.32	99.96	0/48
Shared + agent IDs	1	–	88.93	99.95	0/48
Shared + agent IDs	2	–	66.16	99.93	10/48
Shared + agent IDs	4	–	43.75	99.87	27/48
Shared + agent IDs	8	–	13.32	99.82	41/48
Witness + uniform audits	0.5	0/24	99.99	100.00	0/48
Witness + uniform audits	1	0/24	99.95	99.99	0/48
Witness + uniform audits	2	0/24	99.77	99.99	0/48
Witness + uniform audits	4	8/24	52.87	99.98	21/48
Witness + uniform audits	8	19/24	9.30	99.97	41/48
Witness + coalition audits	0.5	0/24	99.34	99.99	0/48
Witness + coalition audits	1	1/24	88.20	99.99	1/48
Witness + coalition audits	2	19/24	45.05	99.98	11/48
Witness + coalition audits	4	24/24	4.17	99.99	37/48
Witness + coalition audits	8	24/24	0.03	100.00	48/48

Table S5: Allocation control with same-host repetitions. Every row contains the same 24 general graphs and both training replicates. Completion columns are percentages. The horizon, endpoint and starting policy are unchanged.

Budget	Grouping and allocation	Unsafe	Legitimate	Joint recovery
1	Witness, reference	51.20	99.99	9/48
1	Shuffled, reference	84.79	99.99	0/48
1	Shuffled, redesigned	77.23	99.99	0/48
2	Witness, reference	37.84	99.98	18/48
2	Shuffled, reference	56.75	99.99	6/48
2	Shuffled, redesigned	42.62	99.98	13/48

4 (Table S7). The script `analysis/design_checks.py` computes these quantities from the rebuilt graph bank.

Table S6: Output-head resets at the reference budget across all six families, with 72 runs per row. p_0 is the reset silence probability, and the resets keep the hidden layer but change the starting policy. Completion columns are percentages.

Architecture	p_0	Unsafe	Legitimate	Joint recovery
Shared + agent IDs	0.75	26.96	55.46	2/72
Shared + agent IDs	0.95	77.30	99.92	1/72
Shared + scalar offsets	0.75	50.02	55.55	3/72
Shared + scalar offsets	0.95	93.05	100.00	5/72
Fully shared	0.75	55.55	55.55	0/72
Fully shared	0.95	100.00	100.00	0/72
Shuffled groups	0.75	34.86	52.77	4/72
Shuffled groups	0.95	76.97	99.99	0/72
Witness groups	0.75	2.32	48.61	26/72
Witness groups	0.95	24.10	99.99	32/72

Without the optimistic transformation, the graphs and replicates of Table S9 give witness groups 51.77% unsafe completion and 13 of 48 joint recoveries, and shuffled groups 90.48% and none. With the transformation, witness groups achieve 1 and 2 joint recoveries at its two settings. Zhao et al. [44] note that optimism can mislead learning when rewards are stochastic, as the sampled fines are here.

The continuation experiment (Table S10) also helps on the general graphs alone, where it lowers witness-group unsafe completion from 50.85% to 29.82% and raises joint recoveries from 3 to 8 of 16.

Witness error (Table S8) raises witness-group unsafe completion on the general graphs to 66.45% and 65.21%, against 68.58% and 70.01% across all six families.

F The common-draw control

Identical marginal probabilities do not imply the same joint policy, and this control changes only the joint policy. The control uses the fully shared network and the same observation for every agent. For each context x and episode b , it draws U_{xb} uniformly from $[0, 1]$ and sets

$$s_{xbi} = \mathbf{1}\{U_{xb} < p_x\} \quad \text{for every agent } i.$$

The sampler still draws an array of independent uniforms with the dimensions of the original arm and uses the first agent’s entry as the common draw. Drawing the full array keeps the independent audit events at the same positions in the random stream. All rewards, fines and costs are computed by the original functions.

Every task has a nonempty witness set, so all tasks complete exactly when the common action is silence. Write B_x for the total benefit of context x and D_x for the total expected cost of silence in it. The expected benefit is $p_x B_x$, the expected cost is $p_x D_x$, and the expected completion fraction is p_x . The argument proves Equation (7) without assuming equal witness widths.

Every general reference allocation satisfies $D > B$ at the benefit upper bound, and $D_x = D$ in an unsafe context, so $B_x - D_x < 0$ there. Legitimate contexts have no fines, so $D_x = \sum_i c_i = 0.5B$ in them. Since $B_x \geq 0.65B$, their common-draw slope is $B_x - 0.5B \geq 0.15B > 0$. These expected rewards favor reporting in unsafe contexts and silence in legitimate ones. These signs do not remove the coupling between contexts in the network or establish convergence for PPO.

The joint policy has only two supported action vectors, $\mathbf{0}$ and $\mathbf{1}$, with likelihood

$$P_\theta(s\mathbf{1} \mid x) = p_x^s (1 - p_x)^{1-s}.$$

For a stored common action, every agent’s marginal PPO ratio equals this joint ratio, so every clipped surrogate term is identical. Averaging them across agents gives one copy of the single-Bernoulli surrogate. The mean marginal entropy also equals the entropy of this two-action joint policy. Multiplying the identical marginal likelihoods would be wrong, because the draws are dependent. The implementation keeps the original average and changes only the sampler and the evaluator.

Keeping that average does not keep its normalization relative to the joint-policy score. Let π_θ be the shared marginal policy. Under independent actions,

$$\frac{1}{n} \sum_i \nabla_\theta \log \pi_\theta(s_i \mid x) = \frac{1}{n} \nabla_\theta \log P_\theta^{\text{ind}}(s \mid x). \quad (\text{S9})$$

Under the common draw, $s_i = s$ for all i , so

$$\frac{1}{n} \sum_i \nabla_\theta \log \pi_\theta(s \mid x) = \nabla_\theta \log P_\theta^{\text{common}}(s\mathbf{1} \mid x). \quad (\text{S10})$$

At the behavior policy, these are the score factors of the unclipped actor update. Write $H(p_x)$ for the entropy of a Bernoulli draw with parameter p_x . The joint entropy is $nH(p_x)$ for the independent policy and $H(p_x)$ for the common policy, while the implemented mean marginal entropy equals $H(p_x)$ in both arms. So the joint action spaces, exploration and normalization differ even though the surrogate code is unchanged. Standardized advantages, clipping and Adam also prevent reading the observed PPO steps as a direct comparison of exact derivatives.

The intervention keeps the starting marginal probabilities and pretrained arrays, changes the starting joint distribution and assumes a common random source and a common observation. Agents with different conditional probabilities would in general act differently on the same uniform draw, which would require a different joint likelihood. The result concerns the two-action joint policy defined above.

All 48 common-draw runs meet the joint recovery endpoint (Figure S2). The largest run-level final-quarter unsafe completion is 9.30×10^{-6} as a fraction, the smallest legitimate completion is 0.999918, and the mean unsafe fraction is 6.788×10^{-6} . The bounded logit keeps every silence probability above $\sigma(-12) \approx 6.1 \times 10^{-6}$, so these values sit near the smallest the network can produce, and the main text reports the mean exactly instead of rounding it to zero.

Separate checks compared sampled rewards over 160,000 joint episodes with the analytic common-draw expectation, and compared the gradient of the surrogate averaged over repeated agents with the gradient of one Bernoulli surrogate. The largest absolute standardized reward discrepancy was 1.94, and the largest gradient discrepancy was 7.26×10^{-18} . A scalar network forward pass and reward calculation also recomputed every saved final policy value.

G Verification and retained evidence

The returned main archive contains all 4,080 scheduled records, the source snapshot, the input bank, pretrained policies and execution metadata. Its source and schedule match the released study package. Record hashes, graph and prototype identities, endpoint calculations, audit budgets and continuation prefixes were checked. Evaluating the saved final policies separately gives a largest absolute difference of 5.12×10^{-13} from the stored values, and all discrete certificate results agree.

The original package verifier compared full certificate dictionaries with exact floating-point equality, which can reject a valid run on another platform because of rounding. Our audit compares discrete fields exactly and real-valued fields within stated tolerances. The largest certificate discrepancy is 3.41×10^{-13} . The change affects only the rule for numeric comparisons, and no learned policy or result record changes.

The 288-record allocation control reproduces every original-arm joint recovery classification, and its largest cross-platform difference in mean unsafe completion is 0.0162 percentage points. The 96-record common-draw control also reproduces every original-arm classification, with a largest unsafe-fraction difference of $9.97 \times$

Table S7: Output-head resets by witness width, with 36 runs per row across all six families. The start column is the unsafe and legitimate completion $100p_0^k$ at the reset. Completion columns are percentages.

Architecture	p_0	k	Start	Unsafe	Legitimate	Legitimate below 90%	Joint recovery
Witness groups	0.75	4	31.64	2.27	86.10	5/36	25/36
Witness groups	0.75	8	10.01	2.37	11.11	32/36	1/36
Witness groups	0.95	4	81.45	4.65	99.99	0/36	22/36
Witness groups	0.95	8	66.34	43.54	99.99	0/36	10/36
Shuffled groups	0.75	4	31.64	60.58	91.66	3/36	4/36
Shuffled groups	0.75	8	10.01	9.14	13.89	31/36	0/36
Shuffled groups	0.95	4	81.45	62.19	99.99	0/36	0/36
Shuffled groups	0.95	8	66.34	91.75	99.99	0/36	0/36
Fully shared	0.75	4	31.64	100.00	100.00	0/36	0/36
Fully shared	0.75	8	10.01	11.11	11.11	32/36	0/36
Fully shared	0.95	4	81.45	100.00	100.00	0/36	0/36
Fully shared	0.95	8	66.34	99.99	99.99	0/36	0/36
Shared + scalar offsets	0.75	4	31.64	88.92	99.99	0/36	3/36
Shared + scalar offsets	0.75	8	10.01	11.11	11.11	32/36	0/36
Shared + scalar offsets	0.95	4	81.45	86.11	100.00	0/36	5/36
Shared + scalar offsets	0.95	8	66.34	99.99	99.99	0/36	0/36
Shared + agent IDs	0.75	4	31.64	42.81	99.82	0/36	2/36
Shared + agent IDs	0.75	8	10.01	11.11	11.11	32/36	0/36
Shared + agent IDs	0.95	4	81.45	54.60	99.84	0/36	1/36
Shared + agent IDs	0.95	8	66.34	99.99	99.99	0/36	0/36

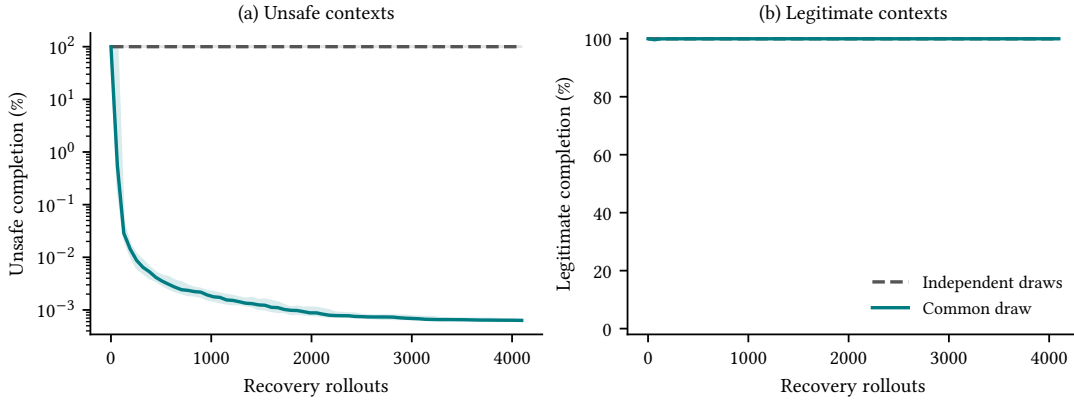


Figure S2: Learning in the common-draw control. Lines are medians across the 48 general-graph runs and shading gives the interquartile range. Unsafe completion uses a logarithmic scale. The legitimate panel shows that the common draw does not reach low unsafe completion by reporting everywhere. The tabulated endpoint is a final-quarter mean.

10^{-11} , and separately recomputed final values differ by at most 2.84×10^{-14} .

The main run took 155.54 minutes on an Apple M2 Pro using Python 3.12.12 and NumPy 2.5.0. The allocation and common-draw controls took 10.44 and 2.91 minutes on a Linux host using Python 3.12 and NumPy 2.3.5, with six workers for the common-draw control. These wall times describe only the stated runs. No GPU or model API was used for simulation training.

The supplement keeps every per-run outcome with its graph, replicate, experiment and method identifiers, which support all tables and aggregate figures without treating repeated measurements as new units. The main archive and the two control archives stay

separate in the analysis, so the later controls cannot replace the prespecified estimate. The input bank itself is not included, but `reproducibility/study/tools/rebuild_inputs.py` rebuilds it from the protocol. A rebuild on a different machine reproduced the schedule file byte for byte and the recorded range of general-graph $B - D$ to within 10^{-12} . From the rebuilt bank, `analysis/design_checks.py` recomputes the guard shares, single-agent certificates, elimination rounds and thresholds reported in the text. For the five budget-sweep arms that form a grouping, it also checks the certificate of every run on the general graphs in each of the run's four unsafe contexts, and it writes Table S3 and Table 3(a).

Table S8: Witness error at the reference budget across all six families, with 72 runs per row. Error is the nominal fraction of tasks whose witness set is replaced, rounded to an integer task count. The true graph supplies the budget and the environment. Completion columns are percentages.

Method	Error	Unsafe	Legitimate	Joint recovery
Shared + scalar offsets	10%	91.66	100.00	6/72
Shared + scalar offsets	25%	95.94	100.00	1/72
Witness + uniform audits	10%	99.99	99.99	0/72
Witness + uniform audits	25%	99.99	99.99	0/72
Witness groups	10%	68.58	99.99	7/72
Witness groups	25%	70.01	99.99	5/72

Table S9: Optimistic advantage transformations at the reference budget. Each row has 48 runs over six families, two widths, the first two graph seeds and two training replicates. The transformation is $\max\{\omega\hat{A}, \hat{A}\}$. Completion columns are percentages. This tests one component and does not reproduce the full method of Zhao et al. [44].

Grouping	ω	Unsafe	Legitimate	Joint recovery
Shuffled groups	0	96.63	99.73	0/48
Shuffled groups	0.5	98.96	100.00	0/48
Witness groups	0	82.63	99.62	1/48
Witness groups	0.5	87.65	99.99	2/48

Table S10: Fixed continuation subset at the reference budget. Each row has 24 runs over six families, two widths, the first graph seed and two training replicates. The 16,384-rollout run continues the 4,096-rollout run with optimizer and random state preserved, and each endpoint uses the final quarter of its own horizon. Completion columns are percentages.

Architecture	Rollouts	Unsafe	Legitimate	Joint recovery
Shared + scalar offsets	4,096	95.83	100.00	1/24
Shared + scalar offsets	16,384	91.66	100.00	2/24
Shuffled groups	4,096	92.34	99.99	0/24
Shuffled groups	16,384	78.67	100.00	0/24
Witness groups	4,096	42.44	99.99	8/24
Witness groups	16,384	26.97	100.00	14/24

The script `analysis/reproduce.py` writes every other table and figure from the included data.

A further check reconstructed all 4,080 original records and both control archives with an actor forward pass and reward calculation that do not import the study evaluator or network. Its largest saved-final-value discrepancy on the original records is 1.14×10^{-13} , and every joint recovery classification is unchanged. Exact rational action enumeration separately checks the expected-reward polynomial, its derivatives and the Fisher factors on finite instances. Order costs, capped purchase costs, maximum closure and the finite-time bound are checked against separate finite calculations, and directional differences check the implemented PPO and network gradients. The `revision_audit` folder contains these

scripts and their outputs, and the raw-record reconstruction needs the full result archives, which are not part of the supplement. These checks add no training observations and do not constitute a general computer-verified proof.