WHO GETS BLAMED? ATTRIBUTION-EVASIVE ADVERSARIAL ATTACKS IN COOPERATIVE MULTI-AGENT REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Existing adversarial attacks in cooperative multi-agent reinforcement learning (c-MARL) mainly focus on degrading team performance, treating a lower return or win rate as the primary measure of attack success. However, after a team failure, post-hoc attribution analysis may assign responsibility among agents by estimating their relative contributions to the observed attacked trajectory. An attack that successfully disrupts the team but makes the compromised agent highly attributable therefore leaves a strong diagnostic footprint. This motivates a distinct adversarial objective: a compromised agent should not only induce team failure, but also avoid becoming the most attributable cause of that failure. We study attribution-evasive adversarial attacks in cooperative multi-agent reinforcement learning and propose **AE-APL** (Attribution-Evasive Adversarial Policy Learning). The proposed framework augments a standard adversarial policy objective with an exposure-aware penalty, using a counterfactual Joint-Q attribution critic to estimate a local value-based surrogate of each agent's harmful contribution. The resulting attribution scores are converted into a relative blame exposure signal, which penalizes the compromised agent when it becomes disproportionately responsible under attribution while preserving a controllable attack-stealth trade-off. Experiments on MaMuJoCo and SMAC show that **AE-APL** improves attribution stealth over strong attack baselines while maintaining competitive attack effectiveness.

1 INTRODUCTION

Cooperative multi-agent reinforcement learning relies on coordination dependencies among agents to achieve shared objectives in tasks such as robotic coordination (Zhu et al., 2024), autonomous driving (Dinneweth et al., 2022), traffic control (Liang et al., 2025), and strategic team games (Ellis et al., 2023). These dependencies also create a security vulnerability: a single compromised participant can steer the joint dynamics toward failure while acting only through its own admissible behavior. This paper focuses on such policy-based attacks, where the adversary does not perturb observations, hijack executed actions, or modify communication channels, but instead controls one participating agent and influences the team through normal environment dynamics. This threat model is especially difficult to audit because the attack is expressed as ordinary interaction rather than as an externally visible manipulation.

Existing attacks against cooperative MARL show that compromised agents can severely degrade team performance by exploiting coordination dependencies. However, most attack objectives and evaluations are disruption-oriented: they measure return drop, win-rate reduction, or convergence to poor joint outcomes. These metrics capture whether an attack succeeds, but they ignore a distinct question that arises in audited cooperative systems: whether the resulting failure leaves counterfactual evidence that makes the compromised agent blameworthy. In such settings, failures may trigger execution-log analysis (Zhang et al., 2025), simulator replay (Choi et al., 2022), or counterfactual diagnosis (Kalech & Natan, 2022). An auditor may ask not only whether an agent's behavior looked abnormal, but whether replacing that agent's realized action with a feasible counterfactual reference would have prevented or mitigated the failure. For an adversary that aims to remain embedded, this attribution exposure is part of the attack risk, not merely a post-hoc evaluation artifact.

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

This post-hoc risk is not captured by existing notions of stealth, which mainly concern detectable perturbations, behavioral abnormality, or attack frequency. Attribution stealth instead concerns whether counterfactual replacement of the attacker's realized actions would substantially improve the team outcome. An attack may therefore appear behaviorally plausible or sparse while remaining highly attributable.

This reveals an objective-level gap in current policy-based c-MARL attack formulations. Existing adversarial policies primarily optimize damage, whereas an embedded adversary may need to jointly optimize damage and the counterfactual evidence of responsibility. The challenge is that attribution evasion is meaningful only under controlled attack effectiveness: an attacker can trivially reduce blame by causing less harm. A useful attribution-evasive attack must therefore reduce exposure at matched or comparable levels of team degradation. This requires incorporating attribution exposure into the adversarial objective and estimating how team value would change under feasible counterfactual action replacements along the realized trajectory. It also requires an exposure signal that is local enough for policy optimization but aligned with the post-hoc attribution rule used to assess responsibility.

We address this problem by formulating *attribution-evasive adversarial policy learning*, in which a compromised agent learns to degrade team performance while reducing post-hoc attribution exposure. Building on this formulation, we introduce Attribution-Evasive Adversarial Policy Learning, referred to as AE-APL, an exposure-aware adversarial policy learning algorithm. AE-APL estimates exposure as a counterfactual value gap: the increase in team value that would result from replacing the attacker's action with a feasible counterfactual reference at the same state. A counterfactual Joint-Q attribution critic provides this signal, which is then used during adversarial policy optimization. The resulting objective penalizes behavior that makes the attacker disproportionately attributable while retaining pressure to reduce team performance; a single exposure weight controls the attack–stealth trade-off. This design turns counterfactual attribution from a passive diagnostic quantity into a training signal for adversarial policy learning.

We evaluate AE-APL on MaMuJoCo and SMAC. Relative to its disruption-only counterpart and existing attack baselines, AE-APL reduces attribution exposure while retaining competitive attack effectiveness. The reduction further transfers to independent post-training auditors with benign-policy references, alternative attribution functionals, and simulator replay. This paper makes the following contributions:

- We introduce attribution exposure as a new objective for policy-based attacks against cooperative MARL, measuring whether the compromised agent is assigned a disproportionately large share of harmful attribution under post-hoc analysis rather than whether its behavior appears unusual or infrequent.
- We propose AE-APL, an exposure-aware adversarial policy learning method that constructs a counterfactual value-based surrogate of local harmful contribution and uses the resulting relative exposure as an optimization signal to balance team degradation against attribution risk.
- Experiments show that AE-APL transfers from benign-policy-free training to stronger post-hoc auditing with benign-policy references, independent models, alternative attribution functionals, and simulator replay.

2 Related Work

2.1 Adversarial Attacks in MARL

Adversarial attacks in RL and MARL may perturb observations (Lin et al., 2020), manipulate executed actions (Lee et al., 2025; Yuan et al., 2023; Jiang et al., 2026), or corrupt communication channels (Tu et al., 2021). The threat model closest to ours is the adversarial policy attack, in which a compromised agent learns its own policy and influences the team only through normal environment dynamics (Gleave et al., 2020; Guo et al., 2021; Wu et al., 2021; Liu et al., 2024; Li et al., 2025). Existing methods primarily optimize team disruption, such as return drop or win-rate reduction. AE-APL retains this environment-mediated threat model but additionally minimizes the post-hoc evidence that attributes the resulting failure to the compromised agent.

2.2 Stealthy Attacks

Existing stealthy attacks mainly reduce detectability during execution, for example by restricting interventions to critical states or timesteps (Lin et al., 2017; Yang et al., 2020; Sun et al., 2020; Cheng et al., 2023), or by constraining attack frequency, action deviation, and behavioral abnormality (Li et al., 2022; Liu et al., 2024). These notions differ from attribution stealth, which concerns whether counterfactual replacement of an attacker's realized actions would substantially improve the team outcome after failure. An attack may therefore appear sparse or behaviorally plausible while remaining highly attributable.

2.3 Multi-Agent Attribution

Multi-agent attribution commonly estimates an agent's contribution through counterfactual comparisons. Representative approaches include difference rewards (Wolpert & Tumer, 2001), counterfactual advantages (Foerster et al., 2018), value decomposition (Sunehag et al., 2018; Rashid et al., 2020), and Shapley-based credit assignment (Wang et al., 2020; Li et al., 2021). Related work on blame attribution and causal responsibility similarly uses counterfactual evidence for explanation and failure diagnosis (Triantafyllou et al., 2021; 2022; Zhang et al., 2025). Whereas these methods identify responsible agents, AE-APL studies the inverse problem: how a strategic adversary can optimize its policy to reduce the attribution signals used by post-hoc auditors.

3 Method

3.1 Problem Formulation and Threat Model

We model cooperative multi-agent decision making as a partially observable Markov game

$$\mathcal{G} = \left\langle \mathcal{N}, \mathcal{S}, \{\mathcal{O}^i\}_{i=1}^N, \{\mathcal{A}^i\}_{i=1}^N, P, r^{\text{team}}, \gamma \right\rangle, \tag{1}$$

where $\mathcal{N} = \{1, \dots, N\}$ is the agent set, $\mathcal{S}$ is the global state space, $\mathcal{O}^i$ and $\mathcal{A}^i$ are the observation and action spaces of agent i, P is the transition function, r^{team} is the shared team reward, and $\gamma \in [0, 1)$ is the discount factor. At timestep t, agent i acts according to $\pi^i(\cdot \mid h_t^i)$, where $h_t^i = (o_0^i, a_0^i, \dots, o_{t-1}^i, a_{t-1}^i, o_t^i)$ is its local observation–action history.

Under nominal operation, each agent follows a benign policy π_{B}^i. The attacker compromises agent $k \in \mathcal{N}$ and replaces π_{B}^k with an adversarial policy π_θ^k. The resulting joint policy is $\Pi_\theta = \left(\pi_\theta^k, \Pi_{\text{B}}^{-k}\right)$, with

$$\Pi_\theta(\mathbf{a}_t \mid \mathbf{h}_t) = \pi_\theta^k(a_t^k \mid h_t^k) \prod_{i \neq k} \pi_{\text{B}}^i(a_t^i \mid h_t^i), \tag{2}$$

where $\mathbf{a}_t = (a_t^1, \dots, a_t^N)$ and $\mathbf{h}_t = (h_t^1, \dots, h_t^N)$.

During training, the attacker has access to global states, joint actions, and team rewards for learning the adversarial policy and the attribution critic, but does not access or query the original benign policies. At execution time, π_θ^k observes only h_t^k and cannot directly modify any other agent or communication channel. Under this threat model, the attacker seeks to degrade team performance while reducing its post-hoc attribution exposure. We express this goal as

$$\max_\theta \; J_{\text{atk}}(\theta) - \beta J_{\text{exp}}^k(\theta), \tag{3}$$

where J_{exp}^k denotes the attribution exposure of the compromised agent and $\beta \geq 0$ controls the trade-off. The attack return is

$$J_{\text{atk}}(\theta) = \mathbb{E}_{\tau \sim \Pi_\theta}\left[\sum_{t=0}^{T-1} \gamma^t r_t^{\text{atk}}\right], \tag{4}$$

where $r_t^{\text{atk}} = -r_t^{\text{team}}$. We next construct a tractable surrogate for J_{exp}^k and incorporate it into adversarial policy learning.

3.2 COUNTERFACTUAL ATTRIBUTION SURROGATE

AE-APL uses a centralized Joint-Q critic to construct a local value-based attribution surrogate. Under the attacked joint policy Π_θ, the team action-value function is

$$Q^{\Pi_\theta}_{\text{team}}(s, \mathbf{a}) = \mathbb{E}_{\Pi_\theta}\left[\sum_{\ell=0}^{\infty} \gamma^\ell r^{\text{team}}_\ell \;\middle|\; s_0 = s, \mathbf{a}_0 = \mathbf{a}\right]. \tag{5}$$

We approximate this function with a centralized critic Q_ϕ trained on trajectories generated by Π_θ:

$$\mathcal{L}_Q(\phi) = \mathbb{E}_{(s_t, \mathbf{a}_t)\sim\mathcal{D}_Q}\left[\left(Q_\phi(s_t, \mathbf{a}_t) - y_t\right)^2\right], \tag{6}$$

where $y_t = r^{\text{team}}_t + \gamma(1 - d_t)Q_{\phi^-}(s_{t+1}, \mathbf{a}_{t+1})$. Here, d_t is the termination indicator, $\mathbf{a}_{t+1}$ is the next joint action in the attacked rollout, and Q_{ϕ^-} is the target critic.

Continuous-action proposal. Let $\mathcal{B}_u$ denote the valid transitions in the attacked rollout batch collected before policy update u. For each agent i, we compute the component-wise empirical mean action

$$\overline{\mathbf{a}}^i_u = \frac{1}{|\mathcal{B}_u|}\sum_{(e,t)\in\mathcal{B}_u} \mathbf{a}^i_{e,t}. \tag{7}$$

At each update, we sample $\alpha_u \sim \text{Uniform}(0, 1)$, and share the sampled value across all agents and transitions in that update. The reference action is

$$\widetilde{\mathbf{a}}^i_{t,u} = (1 - \alpha_u)\mathbf{a}^i_t + \alpha_u \overline{\mathbf{a}}^i_u. \tag{8}$$

The corresponding proposal is the conditionally deterministic Dirac distribution

$$\mu^i_{\text{cf}}\left(\widetilde{\mathbf{a}}^i \mid h^i_t, \mathbf{a}^i_t; \mathcal{B}_u, \alpha_u\right) = \delta_{\widetilde{\mathbf{a}}^i_{t,u}}\left(\widetilde{\mathbf{a}}^i\right). \tag{9}$$

Since the continuous action space is convex, the interpolation of two feasible actions remains feasible and requires no additional clipping.

Discrete-action proposal. For discrete-action domains, let $m^i_t(a) \in \{0, 1\}$ denote the environment-provided action-availability mask and define $\mathcal{A}^i_{\text{feas}}(h^i_t) = \left\{a \in \mathcal{A}^i : m^i_t(a) = 1\right\}$. We enumerate every feasible replacement while holding the other agents' actions fixed, setting $q^i_t(a) = Q_\phi\left(s_t, \mathbf{a}^{-i}_t \oplus a\right)$ for $a \in \mathcal{A}^i_{\text{feas}}(h^i_t)$. Let $K^i_t = \min\left\{3, \left|\mathcal{A}^i_{\text{feas}}(h^i_t)\right|\right\}$ and let $a^i_{t,(1)}, \ldots, a^i_{t,(K^i_t)}$ be the actions with the largest values, ordered such that

$$q^i_t(a^i_{t,(1)}) \geq \cdots \geq q^i_t(a^i_{t,(K^i_t)}). \tag{10}$$

Given fixed nonnegative Top-3 base weights $\bar\omega_1, \bar\omega_2, \bar\omega_3$, we normalize the available prefix as $\omega^{(K^i_t)}_r = \frac{\bar\omega_r}{\sum_{j=1}^{K^i_t}\bar\omega_j}$ for $r = 1, \ldots, K^i_t$, and define the categorical proposal

$$\mu^i_{\text{cf}}\left(a \mid h^i_t, \mathbf{a}_t\right) = \sum_{r=1}^{K^i_t} \omega^{(K^i_t)}_r \mathbb{I}\left[a = a^i_{t,(r)}\right]. \tag{11}$$

The executed action remains a candidate whenever it is feasible. Unavailable actions and transitions after agent death are excluded.

Counterfactual contribution. For both action-space types, the counterfactual team value is

$$Q^i_{\text{cf},\phi}\left(s_t, \mathbf{a}^{-i}_t\right) = \mathbb{E}_{\widetilde{a}^i_t\sim\mu^i_{\text{cf}}}\left[Q_\phi\left(s_t, \mathbf{a}^{-i}_t \oplus \widetilde{a}^i_t\right)\right]. \tag{12}$$

For continuous actions, this expectation reduces to the single critic evaluation induced by Eq. (8). For discrete actions, it becomes the weighted Top-3 aggregate $Q^i_{\text{cf},\phi} = \sum_{r=1}^{K^i_t} \omega^{(K^i_t)}_r q^i_t(a^i_{t,(r)})$.

The resulting local harmful contribution is

$$c^i_t = m^i_{t,\text{valid}}\left[Q^i_{\text{cf},\phi}\left(s_t, \mathbf{a}^{-i}_t\right) - Q_\phi(s_t, \mathbf{a}_t)\right]_+, \tag{13}$$

where $[x]_+ = \max(x, 0)$ and $m^i_{t,\text{valid}}$ excludes invalid transitions, including timesteps after agent death in discrete combat domains. A large c^i_t indicates that replacing the realized action with feasible reference alternatives is predicted to increase team value substantially. If no feasible replacement yields a better counterfactual value, the positive gap is zero.

The intervention changes only agent i's action at the current timestep while retaining the attacked continuation policy represented by Q_ϕ. c^i_t measures one-step local harm.

3.3 Relative Attribution Exposure

To compare the contributions of different agents, we normalize $\{c^i_t\}_{i=1}^N$ into attribution shares:

$$p^i_t = \frac{c^i_t + \epsilon/N}{\sum_{j=1}^N c^j_t + \epsilon}, \tag{14}$$

where $\epsilon > 0$ prevents division by zero. The resulting shares satisfy $\sum_{i=1}^N p^i_t = 1$ and reduce to the uniform distribution when all contributions are zero.

We first measure how much the compromised agent's attribution share exceeds the uniform baseline:

$$e^k_{t,\text{share}} = \left[\frac{p^k_t - \frac{1}{N}}{1 - \frac{1}{N}} \right]_+ . \tag{15}$$

The term penalizes the compromised agent only when its attribution share exceeds $1/N$, with a larger penalty for a larger excess.

We additionally use a pairwise soft-rank term:

$$e^k_{t,\text{rank}} = \frac{2}{N-1} \sum_{j \neq k} \left[\sigma\left(\frac{c^k_t - c^j_t}{\tau_r} \right) - \frac{1}{2} \right]_+ , \tag{16}$$

where $\sigma(\cdot)$ is the sigmoid function and $\tau_r > 0$ controls its scale. This term is zero under contribution ties and increases only when the attacker receives a larger contribution than another agent.

The step-level exposure is

$$\mathcal{E}^k_t = \lambda_\text{s} e^k_{t,\text{share}} + (1 - \lambda_\text{s}) e^k_{t,\text{rank}}, \tag{17}$$

where $\lambda_\text{s} \in [0, 1]$. Both terms are bounded in $[0, 1]$. The first term measures the compromised agent's attribution share, while the second term compares its contribution with those of the other agents.

AE-APL optimizes a dense step-level exposure signal, whereas evaluation is based on trajectory-level attribution measures computed with multiple post-training attribution methods.

3.4 Attack-Preserving Policy Optimization

Minimizing attribution exposure alone may simply weaken the attack, since causing less damage also reduces responsibility. To avoid this trivial behavior, AE-APL activates the exposure penalty only after the adversarial policy achieves a sufficient level of team degradation.

Let u index policy updates, and let $\widehat{J}^{(u)}_\text{team}$ denote the average team return over recent attacked episodes. Given a threshold J_floor, we define the attack-capability gate as

$$g^{(u)} = \mathbb{I}\left[\widehat{J}^{(u)}_\text{team} \leq J_\text{floor} \right]. \tag{18}$$

The gate is evaluated once before each PPO update and held fixed throughout its PPO epochs. It activates the exposure penalty only when the recent team return falls below the threshold. Thus, it serves as an update-level switch rather than a trajectory-level condition or a hard constraint on attack strength.

Table 1: Comparison with existing attacks at two AE-APL operating points, $\beta = 0.25$ and $\beta = 1$ (default). JQ-NR and Top-1 are computed by an independently fitted post-training Joint-Q auditor; higher JQ-NR and lower Top-1 indicate lower attribution exposure. Higher MaMuJoCo return drop and lower SMAC win rate indicate stronger attacks.

Method	MaMuJoCo JQ-NR (↑)	MaMuJoCo Top-1 (↓)	MaMuJoCo Return Drop (%, ↑)	SMAC JQ-NR (↑)	SMAC Top-1 (↓)	SMAC Win Rate (↓)
AMI	0.284 ± 0.109	0.377 ± 0.118	59.53 ± 14.21	0.389 ± 0.145	0.200 ± 0.118	0.200 ± 0.092
GMA	0.278 ± 0.137	0.185 ± 0.063	78.84 ± 9.53	0.301 ± 0.159	0.400 ± 0.191	0.181 ± 0.086
ICLR	0.214 ± 0.131	0.244 ± 0.090	78.89 ± 9.45	0.253 ± 0.138	0.492 ± 0.181	$\mathbf{0.172 \pm 0.076}$
ICML	0.265 ± 0.127	0.152 ± 0.060	79.03 ± 9.74	0.173 ± 0.054	0.467 ± 0.073	0.200 ± 0.096
USENIX	0.272 ± 0.092	0.296 ± 0.098	$\mathbf{79.33 \pm 10.87}$	0.124 ± 0.036	0.583 ± 0.097	0.180 ± 0.067
AE-APL ($\beta = 0.25$)	0.427 ± 0.057	0.091 ± 0.031	78.17 ± 9.47	0.527 ± 0.036	0.009 ± 0.008	0.186 ± 0.044
AE-APL ($\beta = 1$)	$\mathbf{0.692 \pm 0.133}$	$\mathbf{0.021 \pm 0.014}$	76.41 ± 9.30	$\mathbf{0.890 \pm 0.028}$	$\mathbf{0.000 \pm 0.000}$	0.213 ± 0.082

The exposure-aware reward is

$$r_t^{\mathrm{AE}} = r_t^{\mathrm{atk}} - \beta g^{(u)} \, \mathrm{sg}\left(\mathcal{E}_t^k\right), \tag{19}$$

where $\beta \geq 0$ controls the attack–stealth trade-off and $\mathrm{sg}(\cdot)$ denotes stop-gradient. The exposure-aware rewards are used to compute generalized advantage estimates, after which π_θ^k and V_ψ are updated using the clipped PPO objective. The two critics serve different purposes. The Joint-Q critic Q_ϕ is trained on team rewards and is used to estimate each agent's counterfactual contribution. In contrast, V_ψ estimates the return under the exposure-aware reward and provides the value baseline for optimizing π_θ^k.

The exposure penalty is applied at the policy-update level rather than conditioned on the outcome of each trajectory. It is active only when $g^{(u)} = 1$. When $\beta = 0$ or $g^{(u)} = 0$, the exposure term vanishes and AE-APL reduces to the base adversarial-policy objective. Implementation details and pseudocode are provided in Appendix; the additional Joint-Q evaluations are used only during training, while execution retains only the adversarial policy with no additional inference cost.

4 Experiments

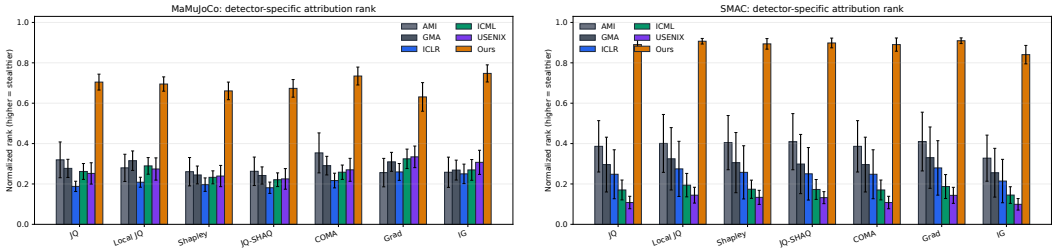
4.1 Experimental Setup

We evaluate AE-APL on five MaMuJoCo tasks and three SMAC scenarios, covering continuous-action and discrete-action cooperative control. We report results over five independent runs with different random seeds. Unless otherwise stated, each auditor constructs the evaluated agent's counterfactual reference using its original benign policy, which is never accessed during AE-APL training. Full environment configurations and training details are provided in Appendix.

Baselines. We compare AE-APL with AMI (Li et al., 2025), GMA (Zan et al., 2023), and three adversarial-policy methods denoted by venue. ICLR trains an adversarial policy against fixed victim policies (Gleave et al., 2020). ICML uses a surrogate attack objective (Guo et al., 2021), while USENIX incorporates explanation-based guidance into PPO (Wu et al., 2021). AMI trains one compromised agent to steer benign teammates toward harmful joint behavior. GMA uses group-based interventions and serves as a broader reference because it does not strictly match our single-agent policy-replacement threat model. Further details are provided in Appendix.

Attribution methods. We evaluate each attack using seven post-training attribution functionals: Joint-Q (JQ), Local Joint-Q (Local JQ), COMA (Foerster et al., 2018), Shapley (Shapley, 1953; Wang et al., 2020), JQ-SHAQ, action gradients (Grad) (Simonyan et al., 2014), and integrated gradients (IG) (Sundararajan et al., 2017). These methods cover counterfactual value comparison, coalition-based contribution, and action-sensitivity attribution. Larger trajectory-level scores indicate greater responsibility for team degradation. Full definitions and implementation details are provided in Appendix.

Evaluation metrics. Attack effectiveness is measured by team return in MaMuJoCo and win rate in SMAC, with lower values indicating stronger attacks. Each attribution method produces



(a) MaMuJoCo (b) SMAC

Figure 1: Normalized rank of the compromised agent under seven independently fitted post-training attribution auditors. "Ours" denotes AE-APL with $\beta = 1$. Higher values indicate lower attribution exposure. The auditors use separate parameters and trajectory sets and are unavailable during attack training.

a trajectory-level score $\tilde{z}_i$, oriented so that larger values indicate greater responsibility. Agents are ranked in descending order of $\tilde{z}_i$, and the compromised agent's normalized rank is

$$\mathrm{NR}_k = \frac{\mathrm{rank}(k) - 1}{N - 1}, \tag{20}$$

where larger values indicate lower exposure. We also report the Top-1 rate, i.e., the fraction of evaluations in which agent k ranks first.

For share-based analysis, we retain the harmful component $h_i = [\tilde{z}_i]_+$ and define

$$\bar{p}_i = \begin{cases} \dfrac{h_i}{\sum_j h_j}, & \sum_j h_j > \epsilon, \\ \dfrac{1}{N}, & \text{otherwise}, \end{cases} \tag{21}$$

where $\epsilon = 10^{-6}$. The median-relative exposure is

$$\mathrm{MedRelExp}_k = \frac{\bar{p}_k + \epsilon}{\mathrm{median}_{j \neq k}\, \bar{p}_j + \epsilon}. \tag{22}$$

Values above one indicate greater harmful attribution than the median benign teammate.

4.2 Comparison with Existing Attacks

Table 1 compares AE-APL at two exposure weights with existing attacks. At $\beta = 0.25$, AE-APL operates at an attack-strength level comparable to the strongest baselines. On MaMuJoCo, its return drop of 78.17% differs from those of GMA, ICLR, and ICML by only 0.67–0.86 percentage points. On SMAC, its win rate of 0.186 falls within the range achieved by the competing methods. Despite the similar aggregate attack effectiveness, AE-APL achieves JQ-NR values of 0.427 and 0.527 on MaMuJoCo and SMAC, exceeding the strongest baseline attribution results by 0.143 and 0.138, respectively. Its Top-1 rates also decrease to 0.091 and 0.009, compared with the best baseline rates of 0.152 and 0.200. These results show that the attribution improvement at low β is achieved without a large aggregate reduction in attack effectiveness.

Increasing the exposure weight to $\beta = 1$ further improves attribution stealth. JQ-NR increases to 0.692 on MaMuJoCo and 0.890 on SMAC, while the corresponding Top-1 rates decrease to 0.021 and 0.000. This improvement is accompanied by an effectiveness cost: the MaMuJoCo return drop decreases to 76.41%, and the SMAC win rate increases to 0.213. Thus, $\beta = 0.25$ provides a near-matched aggregate-effectiveness operating point, whereas $\beta = 1$ represents a more stealth-oriented operating point.

To characterize the trade-off beyond these two operating points, we sweep β on MaMuJoCo. As shown in Appendix Figure 2, increasing β initially produces a substantial improvement in attribution stealth. The mean normalized rank reaches 0.692 at $\beta = 1$ and then increases only marginally, to 0.730 at $\beta = 8$. In contrast, the attacked-return increase relative to the disruption-only baseline

Table 2: Environment-grounded validation on MaMuJoCo. The same disruption-only policy ($\beta = 0$) and AE-APL policy ($\beta = 1$) are evaluated using both the independently fitted Joint-Q auditor and simulator replay. Lower attack-effectiveness change indicates stronger attacks, while higher normalized rank indicates lower attribution exposure.

Task	Δ Eff.($\downarrow$)	Δ JQ-NR($\uparrow$)	Replay-NR($\beta = 0$)	Replay-NR($\beta = 1$)	Δ Replay-NR($\uparrow$)
HalfCheetah-6x1	$-3.55\pm7.44\%$	$+0.63\pm0.05$	0.30	0.84	$+0.54$
Humanoid-17x1	$+2.68\pm0.64\%$	$+0.42\pm0.06$	0.33	0.91	$+0.58$
Humanoid-9xbody	$+1.27\pm1.34\%$	$+0.45\pm0.02$	0.40	0.77	$+0.37$
HumanoidStandup-17x1	$+7.42\pm3.61\%$	$+0.43\pm0.07$	0.29	0.59	$+0.30$
Walker2d-6x1	$+10.38\pm6.63\%$	$+0.48\pm0.15$	0.16	0.59	$+0.43$
Mean $\pm$ *SD*	$+3.64\pm5.43\%$	$+0.48\pm0.09$	0.30 ± 0.09	0.74 ± 0.15	$+0.44\pm0.12$

Table 3: Ablation of the exposure objective, its share and rank components, and the attack-capability gate. All exposure-aware variants use $\beta = 1$. Higher return drop and lower win rate indicate stronger attacks; higher JQ-NR and lower Top-1 indicate lower attribution exposure.

Variant	MaMuJoCo Return Drop (%, $\uparrow$)	MaMuJoCo JQ-NR ($\uparrow$)	MaMuJoCo Top-1 ($\downarrow$)	SMAC Win Rate ($\downarrow$)	SMAC JQ-NR ($\uparrow$)	SMAC Top-1 ($\downarrow$)
Disruption only ($\beta = 0$)	78.20 ± 9.57	0.188 ± 0.026	0.244 ± 0.090	0.182 ± 0.076	0.248 ± 0.133	0.470 ± 0.126
Share only ($\lambda_s = 1$)	76.31 ± 9.42	0.618 ± 0.121	0.041 ± 0.020	0.201 ± 0.081	0.851 ± 0.041	0.013 ± 0.011
Rank only ($\lambda_s = 0$)	76.82 ± 9.46	0.573 ± 0.118	0.056 ± 0.025	0.197 ± 0.080	0.823 ± 0.052	0.021 ± 0.016
No gate	71.84 ± 9.88	0.701 ± 0.095	0.019 ± 0.009	0.375 ± 0.085	0.904 ± 0.021	0.000 ± 0.000
Full AE-APL ($\beta = 1$)	76.41 ± 9.30	0.692 ± 0.133	0.021 ± 0.014	0.213 ± 0.082	0.890 ± 0.028	0.000 ± 0.000

($\beta = 0$) grows from 3.643% at $\beta = 1$ to 27.239% at $\beta = 8$. Thus, most of the attribution gain is obtained by $\beta = 1$, whereas larger exposure weights incur substantially greater effectiveness costs for limited additional stealth.

Based on this trade-off, we use $\beta = 0.25$ as the near-matched-effectiveness setting in the comparison with existing attacks and $\beta = 1$ as the default stealth-oriented setting in the subsequent analyses.

4.3 Transfer Across Attribution Methods

AE-APL is trained without benign-policy access. We test whether its exposure reduction transfers to independent post-training auditors with benign-policy references and alternative attribution functionals.

After training each attack policy, we freeze its parameters and construct seven post-training attribution auditors. For each attribution functional, we independently collect an auditor-training trajectory set, initialize and train the required model, and evaluate it on a disjoint trajectory set. The auditors use benign-policy references but share neither parameters nor trajectories with the training-time Joint-Q critic or with one another; all benign-policy queries and auditor outputs are unavailable during attack optimization. They cover three distinct attribution families: counterfactual value comparison (Joint-Q, Local Joint-Q, and COMA), coalition-based marginal contribution (Shapley and JQ-SHAQ), and action sensitivity (action gradients and integrated gradients).

As shown in Figure 1, AE-APL achieves substantially higher normalized ranks under all seven independently fitted auditors. The improvement is consistent across the three attribution families, despite their different mechanisms for assigning agent-level responsibility. Because training uses neither benign-policy references nor access to the post-training auditors, these results demonstrate transfer from a weaker training signal to stronger, unseen attribution analyses. The consistent gains under stronger, unseen auditors provide evidence against critic-specific or evaluation-sample-specific adaptation.

4.4 Relative Attribution Exposure

Normalized rank captures only the compromised agent's position in the attribution ordering and does not reflect the magnitude of its attribution relative to its teammates. We therefore use MaMuJoCo as a representative case and analyze the median-relative exposure defined in Equation 22.

We compare AE-APL with $\beta = 1$ against its disruption-only counterpart with $\beta = 0$. The median paired log ratio

$$\log\left(\frac{\mathrm{MedRelExp}_{\beta=1}}{\mathrm{MedRelExp}_{\beta=0}}\right)$$

is -0.633, corresponding to an approximate 46.9% reduction in relative exposure, with a cluster-bootstrap 95% confidence interval of $[-0.857, -0.383]$. On the original scale, the median paired decrease is 0.763, with an interval of $[0.313, 1.039]$. Among records for which the compromised agent is initially attributed more harm than the median benign teammate, 83.0% fall below this reference under AE-APL. Mean MedRelExp likewise decreases from 1.795 to 0.841. These results indicate that the improvement in normalized rank reflects a substantial reduction in the compromised agent's attribution relative to its teammates, rather than merely an increase in near ties.

As shown in Appendix Table 12, the compromised agent's mean attribution share decreases from 0.150 to 0.081, whereas benign shares and concentration increase only modestly and normalized attribution entropy remains largely unchanged. This indicates that AE-APL reduces the attacker's relative prominence without concentrating attribution on a single benign teammate. Consistently, the Joint-Q raw harm of the attacker decreases from 3.144 to 1.063, while both the aggregate and maximum benign raw harm also decline. Thus, the reduced exposure is not achieved by shifting absolute harmful attribution to benign agents.

4.5 Environment-Grounded Attribution Validation

Simulator replay further strengthens the auditor by evaluating counterfactual effects directly through environment dynamics rather than a learned value model. From the same simulator snapshot, we replace one agent's factual action with its benign-policy action while holding the others fixed, and compare the resulting counterfactual rollout return with the factual return under frozen policies. Replay-NR is computed from these environment-level return differences and therefore does not rely on a learned attribution model.

As shown in Table 2, AE-APL improves Replay-NR on all five tasks, increasing the mean from 0.30 to 0.74 with an average gain of 0.44 ± 0.12. This closely matches the JQ-NR gain of 0.48 ± 0.09, while the attack-effectiveness change averages $+3.64 \pm 5.43\%$ and varies across tasks. The transfer persists under direct environment-level counterfactual evaluation. The consistent task-level improvements under both measures reduce the likelihood that AE-APL's attribution gains arise solely from the learned Joint-Q proxy.

4.6 Ablation Studies

We ablate the share and rank terms of the exposure objective and the attack-capability gate. All exposure-aware variants use $\beta = 1$. The share and rank variants set $\lambda_s = 1$ and $\lambda_s = 0$, respectively, while the variant without the gate applies the exposure penalty at every policy update.

As shown in Table 3, both exposure terms individually improve attribution stealth, while their combination provides the best overall balance between attack effectiveness and attribution exposure. Full AE-APL substantially improves JQ-NR and Top-1 on both benchmark suites with only limited effectiveness loss. Removing the gate retains comparable attribution stealth but noticeably weakens the attack, particularly on SMAC, where the win rate increases from 0.213 to 0.375. These results confirm that the gate prevents exposure reduction from being achieved primarily by weakening the attack. Appendix reports sensitivity to J_{floor}.

5 Conclusion

We introduced AE-APL to study post-hoc attribution exposure as an adaptive adversarial objective in cooperative MARL. Across multiple environments and attribution methods, AE-APL reduces the compromised agent's relative exposure while exhibiting a controllable trade-off with attack effectiveness. These results highlight the need to evaluate attribution mechanisms against policies that explicitly adapt to their diagnostic signals.

AI USE STATEMENT

Generative AI tools were used to assist with language polishing and code development and debugging. All AI-assisted text and code were carefully reviewed, edited, and verified by the authors. The authors take full responsibility for the correctness and content of the final manuscript.

ETHICS STATEMENT

This work studies attribution-evasive attacks in cooperative multi-agent reinforcement learning. The proposed method could be misused to make a compromised agent less attributable after a team failure, which may make failure diagnosis more difficult. Our experiments are conducted only in simulated MaMuJoCo and SMAC environments. We consider a single compromised agent that acts through its own valid environment actions and does not modify other agents, observations, or communication channels. The study does not involve human participants or personal data. We study this problem to better understand the limits of attribution methods under adaptive attacks and to support the development of more reliable failure analysis methods.

REPRODUCIBILITY STATEMENT

Experimental settings, hyperparameters, random seeds, evaluation protocols, and implementation details are documented in the appendix. The appendix also provides the AE-APL training procedure and pseudocode, together with the benchmark and evaluation configurations. The implementation code is available at `https://anonymous.4open.science/r/who_iclr-DEB1/`.

REFERENCES

Zelei Cheng, Xian Wu, Jiahao Yu, Wenhai Sun, Wenbo Guo, and Xinyu Xing. Statemask: Explaining deep reinforcement learning through state mask. *Advances in Neural Information Processing Systems*, 36:62457–62487, 2023.

Hongjun Choi, Zhiyuan Cheng, and Xiangyu Zhang. Rvplayer: Robotic vehicle forensics by replay with what-if reasoning. In *29th Annual Network and Distributed System Security Symposium, NDSS 2022*. The Internet Society, 2022.

Joris Dinneweth, Abderrahmane Boubezoul, René Mandiau, and Stéphane Espié. Multi-agent reinforcement learning for autonomous vehicles: A survey. *Autonomous Intelligent Systems*, 2(1):27, 2022.

Benjamin Ellis, Jonathan Cook, Skander Moalla, Mikayel Samvelyan, Mingfei Sun, Anuj Mahajan, Jakob Foerster, and Shimon Whiteson. Smacv2: An improved benchmark for cooperative multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 36:37567–37593, 2023.

Jakob Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. Counterfactual multi-agent policy gradients. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), 2018. doi: 10.1609/aaai.v32i1.11794.

Adam Gleave, Michael Dennis, Cody Wild, Neel Kant, Sergey Levine, and Stuart Russell. Adversarial policies: Attacking deep reinforcement learning. In *International Conference on Learning Representations*, 2020.

Wenbo Guo, Xian Wu, Sui Huang, and Xinyu Xing. Adversarial policy learning in two-player competitive games. In *International conference on machine learning*, pp. 3910–3919. PMLR, 2021.

Junyong Jiang, Xin Yuan, Longhe Lin, Songze Li, and Lu Dong. Budget-efficient attacks and robustness training for cooperative MARL. In *Forty-third International Conference on Machine Learning*, 2026. URL `https://openreview.net/forum?id=k3T8Cwv1ly`.

Meir Kalech and Avraham Natan. Model-based diagnosis of multi-agent systems: A survey. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(11):12334–12341, 2022. doi: 10.1609/aaai.v36i11.21498.

Sunwoo Lee, Jaebak Hwang, Yonghyeon Jo, and Seungyul Han. Wolfpack adversarial attack for robust multi-agent reinforcement learning. In *Forty-second International Conference on Machine Learning*, 2025.

Jiahui Li, Kun Kuang, Baoxiang Wang, Furui Liu, Long Chen, Fei Wu, and Jun Xiao. Shapley counterfactual credits for multi-agent reinforcement learning. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 934–942, 2021.

Simin Li, Jun Guo, Jingqiao Xiu, Yuwei Zheng, Pu Feng, Xin Yu, Jiakai Wang, Aishan Liu, Yaodong Yang, Bo An, et al. Attacking cooperative multi-agent reinforcement learning by adversarial minority influence. *Neural Networks*, pp. 107747, 2025.

Yang Li, Quan Pan, and Erik Cambria. Deep-attack over the deep reinforcement learning. *Knowledge-Based Systems*, 250:108965, 2022.

Jintao Liang, Xinqi Du, Yilin Liu, Ziyue Li, Haoyuan Jiang, Zhishuai Li, Cheng Long, Hua Wei, and Wolfgang Ketter. A survey on multi-agent reinforcement learning for adaptive transportation solutions. *SN Computer Science*, 6(8):955, 2025.

Jieyu Lin, Kristina Dzeparoska, Sai Qian Zhang, Alberto Leon-Garcia, and Nicolas Papernot. On the robustness of cooperative multi-agent reinforcement learning. In *2020 IEEE Security and Privacy Workshops (SPW)*, pp. 62–68. IEEE, 2020.

Yen-Chen Lin, Zhang-Wei Hong, Yuan-Hong Liao, Meng-Li Shih, Ming-Yu Liu, and Min Sun. Tactics of adversarial attack on deep reinforcement learning agents. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, IJCAI'17, pp. 3756–3762. AAAI Press, 2017. ISBN 9780999241103.

Xiangyu Liu, Souradip Chakraborty, Yanchao Sun, and Furong Huang. Rethinking adversarial policies: A generalized attack formulation and provable defense in rl. In *The Twelfth International Conference on Learning Representations*, 2024.

Tabish Rashid, Mikayel Samvelyan, Christian Schroeder De Witt, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. Monotonic value function factorisation for deep multi-agent reinforcement learning. *Journal of Machine Learning Research*, 21(178):1–51, 2020.

Lloyd S. Shapley. A value for n-person games. In Harold William Kuhn and Albert William Tucker (eds.), *Contributions to the Theory of Games, Volume II*, pp. 307–317. Princeton University Press, 1953.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *2nd International Conference on Learning Representations, ICLR*, 2014.

Jianwen Sun, Tianwei Zhang, Xiaofei Xie, L. Ma, Yan Zheng, Kangjie Chen, and Yang Liu. Stealthy and efficient adversarial attacks against deep reinforcement learning. In *AAAI Conference on Artificial Intelligence*, 2020.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International conference on machine learning*, pp. 3319–3328. PMLR, 2017.

Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinicius Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z. Leibo, Karl Tuyls, and Thore Graepel. Value-decomposition networks for cooperative multi-agent learning based on team reward. In *Proceedings of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018)*, AAMAS '18, pp. 2085–2087. International Foundation for Autonomous Agents and Multiagent Systems, 2018.

Stelios Triantafyllou, Adish Singla, and Goran Radanovic. On blame attribution for accountable multi-agent sequential decision making. *Advances in Neural Information Processing Systems*, 34:15774–15786, 2021.

Stelios Triantafyllou, Adish Singla, and Goran Radanovic. Actual causality and responsibility attribution in decentralized partially observable markov decision processes. In *Proceedings of the 2022 AAAI/ACM Conference on AI, Ethics, and Society*, pp. 739–752, 2022.

James Tu, Tsunhsuan Wang, Jingkang Wang, Sivabalan Manivasagam, Mengye Ren, and Raquel Urtasun. Adversarial attacks on multi-agent communication. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7768–7777, 2021.

Jianhong Wang, Yuan Zhang, Tae-Kyun Kim, and Yunjie Gu. Shapley q-value: A local reward approach to solve global reward games. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):7285–7292, 2020. doi: 10.1609/aaai.v34i05.6220.

David H Wolpert and Kagan Tumer. Optimal payoff functions for members of collectives. *Advances in Complex Systems*, 4(02n03):265–279, 2001.

Xian Wu, Wenbo Guo, Hua Wei, and Xinyu Xing. Adversarial policy training against deep reinforcement learning. In *30th USENIX Security Symposium (USENIX Security 21)*, pp. 1883–1900, 2021.

Chao-Han Huck Yang, Jun Qi, Pin-Yu Chen, Yi Ouyang, I-Te Danny Hung, Chin-Hui Lee, and Xiaoli Ma. Enhanced adversarial strategically-timed attacks against deep reinforcement learning. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3407–3411. IEEE, 2020.

Lei Yuan, Ziqian Zhang, Ke Xue, Hao Yin, Feng Chen, Cong Guan, Lihe Li, Chao Qian, and Yang Yu. Robust multi-agent coordination via evolutionary generation of auxiliary adversarial attackers. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(10):11753–11762, 2023. doi: 10.1609/aaai.v37i10.26388.

Lixia Zan, Xiangbin Zhu, and Zhao-Long Hu. Adversarial attacks on cooperative multi-agent deep reinforcement learning: a dynamic group-based adversarial example transferability method. *Complex & Intelligent Systems*, 9(6):7439–7450, 2023.

Shaokun Zhang, Ming Yin, Jieyu Zhang, Jiale Liu, Zhiguang Han, Jingyang Zhang, Beibin Li, Chi Wang, Huazheng Wang, Yiran Chen, and Qingyun Wu. Which agent causes task failures and when? on automated failure attribution of LLM multi-agent systems. In *Forty-second International Conference on Machine Learning*, 2025.

Changxi Zhu, Mehdi Dastani, and Shihan Wang. A survey of multi-agent deep reinforcement learning with communication. *Autonomous Agents and Multi-Agent Systems*, 38(1):4, 2024.

A Attribution Detector Definitions

This section defines the post-hoc attribution auditors used in evaluation. All model-based auditors are trained only after the corresponding attack policy has been frozen. For each attribution method d, we independently collect an auditor-training trajectory set $\mathcal{D}_d^{\text{train}}$, initialize a detector-specific attribution model, and fit that model without modifying the frozen attack policy. The auditor-training trajectories are disjoint from both the attack-training trajectories and the final auditor-evaluation trajectories $\mathcal{D}_d^{\text{eval}}$.

Auditor parameters, benign-policy queries, counterfactual references, and attribution outputs are unavailable during attack-policy optimization. Unless otherwise stated, the counterfactual reference for agent i is constructed using its original benign policy $\pi_B^i(\cdot \mid h_t^i)$. In discrete-action environments, the benign-policy distribution is restricted and renormalized over the actions permitted by the environment-provided availability mask. Transitions after agent death are excluded from that agent's attribution score.

We distinguish the training-time attribution critic Q_ϕ^{train} used by AE-APL from the independently fitted post-training auditor models $\{Q_{\omega_d}^{\text{aud}}\}_d$. The training-time critic is discarded after attack training and is never reused as a post-training attribution auditor.

A.1 JOINT-Q DETECTOR

Let $Q^{\text{aud}}_{\omega_{\text{JQ}}}(s_t, \mathbf{a}_t)$ denote the independently fitted post-training Joint-Q auditor, where $\mathbf{a}_t = (a_t^1, \ldots, a_t^N)$ is the executed joint action. Let $\bar{a}_t^{i,B}$ denote a feasible counterfactual action generated from the original benign policy of agent i, and let $\mathbf{a}_t^{-i} \oplus \bar{a}_t^{i,B}$ denote the joint action in which only agent i is replaced. The step-level harmfulness score is defined as

$$s^i_{t,\text{JQ}} = m^i_{t,\text{valid}} \left[Q^{\text{aud}}_{\omega_{\text{JQ}}} \left(s_t, \mathbf{a}_t^{-i} \oplus \bar{a}_t^{i,B} \right) - Q^{\text{aud}}_{\omega_{\text{JQ}}} \left(s_t, \mathbf{a}_t \right) \right]_+, \tag{23}$$

where $[x]_+ = \max(x, 0)$ and $m^i_{t,\text{valid}}$ excludes invalid transitions.

For discrete-action environments, the benign-policy distribution is masked and renormalized over available actions before constructing the counterfactual reference. For dead agents, $m^i_{t,\text{valid}} = 0$, and no post-death score is accumulated.

A.2 LOCAL JOINT-Q DETECTOR

The Local Joint-Q detector uses an independently trained post-training auditor $Q^{\text{aud}}_{\omega_{\text{localJQ}}}$. It does not share parameters or auditor-training trajectories with the standard Joint-Q detector.

For one-hot discrete actions, we construct a local counterfactual action by interpolating between the executed action and its benign-policy reference:

$$\widetilde{a}^i_t(\alpha) = (1 - \alpha)\, \text{onehot}(a^i_t) + \alpha\, \text{onehot}(\bar{a}^{i,B}_t), \tag{24}$$

where $\alpha = 0.2$ in our evaluation. The value of the resulting locally perturbed joint action is

$$V^{i,\text{local}}_t = Q^{\text{aud}}_{\omega_{\text{localJQ}}} \left(s_t,\, \mathbf{a}_t^{-i} \oplus \widetilde{a}^i_t(\alpha) \right). \tag{25}$$

The corresponding harmfulness score is

$$s^i_{t,\text{localJQ}} = m^i_{t,\text{valid}} \left[V^{i,\text{local}}_t - Q^{\text{aud}}_{\omega_{\text{localJQ}}} \left(s_t, \mathbf{a}_t \right) \right]_+. \tag{26}$$

For continuous actions, we use the same construction but interpolate directly in the continuous action space from a^i_t toward $\bar{a}^{i,B}_t$.

A.3 SHAPLEY DETECTOR

The Shapley detector estimates each agent's marginal harmful contribution using sampled permutations. For a coalition $C \subseteq \mathcal{N}$, agents in C use their executed actions, whereas agents outside C use their benign-policy counterfactual actions. We denote the auditor value of the resulting joint action by

$$V^{\text{Shapley}}_t(C) = Q^{\text{aud}}_{\omega_{\text{Shapley}}} \left(s_t, \mathbf{a}_t(C) \right), \tag{27}$$

where $\mathbf{a}_t(C)$ denotes the coalition-constructed joint action.

For a sampled permutation π, let C^i_π be the set of agents preceding agent i. We define its permutation-specific harmful contribution as

$$\Delta^i_{t,\pi} = V^{\text{Shapley}}_t(C^i_\pi) - V^{\text{Shapley}}_t(C^i_\pi \cup \{i\}). \tag{28}$$

The step-level Shapley score is then

$$s^i_{t,\text{Shapley}} = m^i_{t,\text{valid}} \left[\mathbb{E}_\pi \left[\Delta^i_{t,\pi} \right] \right]_+. \tag{29}$$

This sign convention converts the standard value-oriented marginal contribution into a harmfulness score. A larger score indicates that replacing agent i's benign-policy reference with its executed action causes a larger decrease in predicted team value. We approximate the expectation using sampled permutations rather than enumerating all coalitions.

A.4 JQ-SHAQ DETECTOR

JQ-SHAQ uses a detector-specific per-agent credit model trained only after the corresponding attack policy has been frozen. The model is fitted on an independently collected JQ-SHAQ auditor-training split and evaluated on a disjoint trajectory split. Its output is oriented so that a larger score indicates greater harmful responsibility for the observed team failure.

The JQ-SHAQ model, its training trajectories, its benign-policy queries, and its attribution outputs are unavailable during attack-policy optimization. No JQ-SHAQ model is trained or fine-tuned on the final auditor-evaluation trajectories. If the detector cannot be successfully fitted for a particular setting, its result is reported as unavailable rather than being obtained by training on the evaluation split.

A.5 COMA-STYLE DETECTOR

The COMA-style detector uses an independently fitted post-training auditor $Q^{\mathrm{aud}}_{\omega_{\mathrm{COMA}}}$ to estimate the harmfulness of agent i's executed action. Let $\pi^i_B(\cdot \mid h^i_t)$ denote the original benign-policy distribution of agent i. We first define the benign-policy-weighted counterfactual baseline as

$$V^{i,B}_{t,\mathrm{COMA}} = \sum_{a' \in \mathcal{A}^i_{\mathrm{avail}}(h^i_t)} \widehat{\pi}^i_B(a' \mid h^i_t) \cdot Q^{\mathrm{aud}}_{\omega_{\mathrm{COMA}}}\left(s_t,\, \mathbf{a}_t^{-i} \oplus a'\right). \tag{30}$$

The corresponding step-level harmfulness score is

$$s^i_{t,\mathrm{COMA}} = m^i_{t,\mathrm{valid}} \left[V^{i,B}_{t,\mathrm{COMA}} - Q^{\mathrm{aud}}_{\omega_{\mathrm{COMA}}}(s_t, \mathbf{a}_t) \right]_+ . \tag{31}$$

The masked benign-policy distribution is defined as

$$\widehat{\pi}^i_B(a \mid h^i_t) = \frac{\pi^i_B(a \mid h^i_t) m^i_t(a)}{\sum_{a' \in \mathcal{A}^i} \pi^i_B(a' \mid h^i_t) m^i_t(a')}. \tag{32}$$

Here, $m^i_t(a)$ indicates whether action a is available, and $\widehat{\pi}^i_B$ is therefore the benign-policy distribution renormalized over valid actions. The reported experiments do not use a uniform-action fallback. A larger score indicates that the executed action is predicted to be more harmful than the benign-policy-weighted counterfactual baseline.

Implementation and fairness controls. For each environment, all policy-based attacks use the same pretrained victim checkpoint, compromised-agent identity, environment interface, attack-training budget, and policy-evaluation protocol. The benign teammate policies remain fixed throughout attack training. Whenever permitted by the original formulations, baseline methods use matched policy and value-network architectures. Shared optimization parameters are kept consistent across methods, whereas method-specific coefficients follow the corresponding official implementations and are selected using the same validation budget.

After each attack policy has been trained and frozen, all methods are evaluated using the same number of episodes and the same post-training attribution protocol. The attribution auditors are trained independently after attack training, using auditor-training trajectories that are disjoint from both attack-training trajectories and final evaluation trajectories. No post-training auditor, benign-policy counterfactual query, or auditor output is available to AE-APL or any baseline during attack-policy optimization.

A.6 ACTION-GRADIENT DETECTOR.

The action-gradient detector uses an independently fitted auditor $Q^{\mathrm{aud}}_{\omega_{\mathrm{Grad}}}$. For agent i, we define

$$s^i_{t,\mathrm{Grad}} = m^i_{t,\mathrm{valid}} \left\| \nabla_{a^i_t} Q^{\mathrm{aud}}_{\omega_{\mathrm{Grad}}}(s_t, \mathbf{a}_t) \right\|_1 . \tag{33}$$

For discrete actions, the gradient is taken with respect to the continuous one-hot action representation before action selection.

A.7 Integrated-Gradients Detector

The integrated-gradients detector uses an independently fitted post-training auditor $Q^{\mathrm{aud}}_{\omega_{\mathrm{IG}}}$ to estimate the contribution of agent i's action to the predicted joint return. We use the benign-policy counterfactual action $\bar{a}_t^{i,B}$ as the integration baseline and approximate the integration path using K_{IG} uniformly spaced steps. At step k, the interpolated action is

$$\tilde{a}_t^{i,(k)} = \bar{a}_t^{i,B} + \frac{k}{K_{\mathrm{IG}}} \left(a_t^i - \bar{a}_t^{i,B} \right). \tag{34}$$

The integrated-gradients attribution is then computed as

$$\begin{aligned} \mathrm{IG}_t^i = \left(a_t^i - \bar{a}_t^{i,B} \right) \odot \frac{1}{K_{\mathrm{IG}}} \sum_{k=1}^{K_{\mathrm{IG}}} \\ \nabla_{a^i} Q^{\mathrm{aud}}_{\omega_{\mathrm{IG}}} \left(s_t,\, \mathbf{a}_t^{-i} \oplus \tilde{a}_t^{i,(k)} \right), \end{aligned} \tag{35}$$

where $\odot$ denotes element-wise multiplication and $\mathbf{a}_t^{-i} \oplus \tilde{a}_t^{i,(k)}$ denotes the joint action obtained by replacing agent i's action with the interpolated action while keeping the actions of all other agents fixed.

The corresponding step-level detection score is

$$s_{t,\mathrm{IG}}^i = m_{t,\mathrm{valid}}^i \left\| \mathrm{IG}_t^i \right\|_1, \tag{36}$$

where $m_{t,\mathrm{valid}}^i$ masks invalid or unavailable agent–time pairs. A larger score indicates that the deviation from the benign counterfactual action has a stronger influence on the auditor's predicted joint return. We set $K_{\mathrm{IG}} = 20$ in all experiments.

A.8 Trajectory-Level Attribution Aggregation

Each detector first produces a nonnegative step-level score $s_{t,d}^i$ for agent i under detector d. We aggregate these scores over the valid timesteps of trajectory τ as

$$\widetilde{z}_d^i(\tau) = \frac{\sum_{t=0}^{T(\tau)-1} m_{t,\mathrm{valid}}^i s_{t,d}^i}{\sum_{t=0}^{T(\tau)-1} m_{t,\mathrm{valid}}^i + \epsilon_{\mathrm{eval}}}, \tag{37}$$

where $\epsilon_{\mathrm{eval}} > 0$ prevents division by zero.

For each trajectory, agents are ranked in descending order of $\widetilde{z}_d^i(\tau)$. The compromised agent's normalized rank is

$$\mathrm{NR}_{k,d}(\tau) = \frac{\mathrm{rank}_d(k;\tau) - 1}{N - 1}, \tag{38}$$

where a larger value indicates lower attribution exposure. The detector-specific Top-1 rate is the fraction of evaluated trajectories for which the compromised agent obtains rank one.

B Compared Attack Baselines

We compare AE-APL with five representative attack baselines, covering standard adversarial-policy learning, improved adversarial-policy objectives, and attacks developed specifically for cooperative multi-agent systems. For brevity and consistency with the main paper, we denote three adversarial-policy baselines by their publication venues: ICLR, ICML, and USENIX.

ICLR. ICLR denotes the adversarial-policy method of Gleave et al. Gleave et al. (2020). The method replaces one agent with a trainable adversarial policy while keeping the victim policies fixed. The adversarial agent interacts with the victims only through admissible environment actions and learns to minimize the performance of the victim team. Originally introduced for two-player environments, we extend the same policy-replacement formulation to cooperative MARL by treating the remaining benign agents as a fixed joint victim policy. This method also serves as the disruption-oriented reference used to configure the capability floor in AE-APL.

ICML. ICML denotes the adversarial-policy learning method of Guo et al. Guo et al. (2021). It improves the optimization of an adversarial policy using a surrogate attack objective that provides a more effective learning signal than directly relying on the environment outcome alone. We adapt the method to the cooperative setting by replacing one benign agent with the adversarial policy and defining its attack objective using the degradation of the shared team performance. All benign teammate policies remain fixed during adversarial-policy training.

USENIX. USENIX denotes the adversarial policy training method of Wu et al. Wu et al. (2021). The method incorporates explanation-based guidance into adversarial-policy optimization to identify actions that more effectively influence the victim policy. We retain its guided adversarial-training objective while extending the victim from a single policy to the fixed joint policy of the benign teammates. The adversarial agent is trained using the same observation and action interface as AE-APL.

AMI. Adversarial Minority Influence (AMI) Li et al. (2025) is a policy-based attack designed specifically for cooperative MARL. AMI characterizes the unilateral influence of the adversarial agent on its benign teammates and trains a targeted adversarial oracle to identify jointly harmful target behaviors. The adversarial policy then attempts to steer the benign agents toward these targets through normal environment interaction. AMI shares our single-compromised-agent and fixed-benign-policy setting, but optimizes influence and team disruption without explicitly considering post-training attribution exposure.

GMA. GMA Zan et al. (2023) is a dynamic group-based attack for cooperative MARL. It selects and attacks groups of agents to exploit cross-agent transferability and degrade collective performance. We include GMA as a broader multi-agent attack reference because it directly targets coordination dependencies among multiple agents. However, unlike AE-APL and the adversarial-policy baselines above, GMA is not strictly matched to the single-agent policy-replacement threat model and may intervene in the behavior of multiple victim agents. Its results should therefore be interpreted as a comparison with a broader and less restrictive attack setting.

Implementation and fairness controls. For each environment, all policy-based methods use the same pretrained victim checkpoint, compromised-agent identity, environment interface, training budget, and evaluation protocol. The benign teammate policies remain fixed throughout attack training. We implement the policy-based baselines within the same training pipeline and use matched policy and value-network architectures whenever their original formulations permit. Shared optimization parameters are kept consistent across methods, while method-specific coefficients are selected according to the corresponding original implementations and tuned using the same validation budget.

All trained attack policies are evaluated using the same number of episodes and the same post-training attribution protocol. In particular, the attribution auditors are fitted only after the attack policies have been frozen and are not available to any baseline during training. Consequently, the comparison measures both the disruption produced by each original attack objective and the attribution footprint that it leaves under a common set of independently fitted auditors.

C Implementation Details of AE-APL

This section provides the implementation procedure of AE-APL. The overall training framework contains three learnable components: the adversarial policy π_θ^k, the PPO value function V_ψ, and the centralized Joint-Q critic Q_ϕ. The adversarial policy controls only the compromised agent k, while the policies of all benign teammates remain fixed throughout training. The Joint-Q critic estimates counterfactual harmful contributions, whereas the PPO value function estimates the return induced by the exposure-aware reward. These two value functions are trained with different targets and are not shared.

C.1 Overall Training Procedure

At each policy update, AE-APL first collects a rollout batch under the current attacked joint policy

$$\Pi_\theta = \left(\pi_\theta^k, \Pi_{\mathrm{B}}^{-k}\right).$$

The resulting trajectories contain global states, joint actions, team rewards, termination indicators, local histories, and, for discrete-action environments, action-availability masks. These trajectories are used both to train the Joint-Q critic and to update the adversarial policy.

Before policy optimization, we compute the average team return over the recent attacked episodes and evaluate the attack-capability gate $g^{(u)}$ using Eq. (18). The gate is evaluated once for each policy update u and remains fixed throughout all PPO epochs associated with that update. Consequently, the attribution penalty is activated only after the current adversarial policy reaches the prescribed level of attack effectiveness. This update-level design prevents individual high-return trajectories from switching the exposure penalty on and off within the same PPO update.

The Joint-Q critic Q_ϕ is then trained on the collected attacked trajectories using the original team reward. Its target critic Q_{ϕ^-} is updated according to the target-network update rule used in the corresponding implementation. After the critic update, Q_ϕ is held fixed while all counterfactual values, harmful contributions, attribution shares, and exposure scores are computed. In particular, the exposure score used to construct the policy reward is detached from the Joint-Q computation graph. Therefore, the policy update does not backpropagate through Q_ϕ or through the counterfactual action-selection procedure.

For every valid transition, AE-APL evaluates the harmful contribution of all agents. The realized action of agent i is replaced by a feasible counterfactual reference while the actions of all other agents remain fixed. The resulting counterfactual Joint-Q value is compared with the factual Joint-Q value to obtain the nonnegative local contribution c_t^i. The contributions of all agents are subsequently normalized into attribution shares, from which the share-based and rank-based exposure terms of the compromised agent are calculated.

Finally, AE-APL constructs the exposure-aware reward r_t^{AE} using Eq. (19). Generalized advantage estimation is applied to this reward sequence, and the adversarial policy π_θ^k and PPO value function V_ψ are updated with the standard clipped PPO objective. The Joint-Q critic is used only to produce the training-time exposure signal and is discarded after training. Hence, AE-APL introduces no additional attribution computation during execution.

C.2 Domain-Specific Counterfactual Construction

AE-APL uses different feasible counterfactual proposals for continuous- and discrete-action environments.

Continuous-action environments. For each policy update u, we compute the empirical mean action $\bar{\mathbf{a}}_u^i$ of every agent i from the valid transitions in the current rollout batch. A single interpolation coefficient $\alpha_u \sim \mathrm{Uniform}(0,1)$ is sampled once at the beginning of the update and shared across agents and transitions. The counterfactual action is then constructed by interpolating between the realized action and the batch-level mean action according to Eq. (9). Sharing α_u within an update ensures that all counterfactual contributions are evaluated under the same interpolation strength.

The interpolation is an operationally feasible reference rather than an estimate of the unavailable benign action. Since the continuous action spaces used in MaMuJoCo are convex after action normalization, interpolating between two feasible actions preserves feasibility and does not require additional clipping.

Discrete-action environments. For discrete-action environments, AE-APL uses the environment-provided action-availability mask to enumerate the feasible replacement actions of each agent. Holding the actions of all other agents fixed, the Joint-Q critic evaluates every available replacement. The feasible actions are ranked by their predicted team value, and the highest-valued actions are aggregated using the normalized Top-3 weighting rule in Eq. (11). When fewer than three actions are available, only the available prefix of the weight vector is retained and renormalized.

The factual action remains among the candidates whenever it is available. Unavailable actions are never evaluated as counterfactual references. Moreover, transitions after an agent has died are excluded from that agent's contribution and exposure computation. These masks ensure that the counterfactual comparison does not assign attribution using actions that could not have been executed in the corresponding environment state.

Algorithm 1 AE-APL Training

Require: Fixed benign policies Π_{B}; attacker index k; exposure weight β; share weight λ_{s}; return threshold J_{floor}

Initialize adversarial policy π_θ^k, PPO value function V_ψ, Joint-Q critic Q_ϕ, and target critic Q_{ϕ^-}
for policy update $u = 1, 2, \ldots$ **do**
  Collect rollout batch $\mathcal{B}_u$ under $\Pi_\theta = (\pi_\theta^k, \Pi_{\mathrm{B}}^{-k})$
  Compute recent team performance $\widehat{J}_{\mathrm{team}}^{(u)}$
  Compute the update-level gate $g^{(u)} = \mathbb{I}\Big[\widehat{J}_{\mathrm{team}}^{(u)} \le J_{\mathrm{floor}}\Big]$ using Eq. (18)
  Update Q_ϕ on $\mathcal{B}_u$ using team rewards, and update the target critic Q_{ϕ^-}
  Hold Q_ϕ fixed during attribution computation
  if the action space is continuous **then**
    Compute the batch-mean action $\bar{\mathbf{a}}_u^i$ for each agent i from valid transitions in $\mathcal{B}_u$
    Sample one shared interpolation coefficient $\alpha_u \sim \mathrm{Uniform}(0, 1)$
  end if
  for each valid transition $(s_t, \mathbf{a}_t)$ in $\mathcal{B}_u$ **do**
    for each agent $i \in \mathcal{N}$ **do**
      if the action space is continuous **then**
        Construct the feasible interpolated action $\widetilde{a}_{t,u}^i$ using Eq. (9)
      else
        Obtain the feasible action set from the action-availability mask
        Evaluate feasible replacements with Q_ϕ while holding $\mathbf{a}_t^{-i}$ fixed
        Construct the normalized Top-3 counterfactual proposal using Eq. (11)
      end if
      Compute the counterfactual value $Q_{\mathrm{cf},\phi}^i(s_t, \mathbf{a}_t^{-i})$ using Eq. (12)
      Compute the local harmful contribution $c_t^i = m_{t,\mathrm{valid}}^i \left[Q_{\mathrm{cf},\phi}^i(s_t, \mathbf{a}_t^{-i}) - Q_\phi(s_t, \mathbf{a}_t)\right]_+$ using Eq. (13)
    end for
    Compute attribution shares $\{p_t^i\}_{i=1}^N$ using Eq. (14)
    Compute the compromised agent's share exposure
    Compute its soft-rank exposure using Eq. (16)
    Combine the share and rank terms to obtain $\mathcal{E}_t^k$ using Eq. (17)
    Detach $\mathcal{E}_t^k$ from the Joint-Q computation graph
  end for
  Construct exposure-aware rewards $r_t^{\mathrm{AE}} = r_t^{\mathrm{atk}} - \beta g^{(u)} \,\mathrm{sg}\big(\mathcal{E}_t^k\big)$ using Eq. (19)
  Compute generalized advantage estimates from $\{r_t^{\mathrm{AE}}\}$ using V_ψ
  Update π_θ^k and V_ψ using the clipped PPO objective
end for
return Trained adversarial policy π_θ^k

C.3 PSEUDOCODE

Algorithm 1 summarizes the complete implementation procedure. The Joint-Q update, attribution computation, and PPO update are executed sequentially within each policy update. The attribution quantities are computed with the updated Joint-Q critic but are treated as fixed reward signals during policy optimization.

Gradient and parameter updates. The three optimization components are updated separately. The Joint-Q critic Q_ϕ is optimized only with the team-reward temporal-difference target. The PPO value function V_ψ is optimized only with returns derived from r_t^{AE}. The adversarial policy receives the exposure signal through the exposure-aware advantages, but no gradient is propagated through the attribution critic, the counterfactual proposal, or the exposure normalization. This separation avoids directly optimizing the policy through the learned auditor and treats attribution exposure as a dense, critic-derived reward signal.

Execution. After training, only the adversarial policy π_θ^k is retained. Neither the Joint-Q critic, the counterfactual proposal mechanism, nor the exposure computation is evaluated during deployment. Consequently, AE-APL has the same execution-time policy architecture and inference cost as its disruption-only adversarial-policy counterpart.

Table 4: Capability-threshold configuration. The threshold permits attacked-team performance up to $0.2|J^e_{\mathrm{ICLR}}|$ above the reference attack result, with a sign-safe definition for potentially negative MaMuJoCo returns.

Benchmark	Definition of J	Capability floor
MaMuJoCo	Attacked-team return	$J^e_{\mathrm{floor}} = J^e_{\mathrm{ICLR}} + 0.2\,\lvert J^e_{\mathrm{ICLR}}\rvert$
SMAC	Attacked-team win rate	$J^e_{\mathrm{floor}} = J^e_{\mathrm{ICLR}} + 0.2\,\lvert J^e_{\mathrm{ICLR}}\rvert$

Table 5: Sensitivity of AE-APL to the capability-floor multiplier ρ on MaMuJoCo. The environment-specific threshold is $J^e_{\mathrm{floor}}(\rho) = J^e_{\mathrm{ICLR}} + (\rho - 1)|J^e_{\mathrm{ICLR}}|$.

ρ	Return Drop (%, ↑)	JQ-NR (↑)	Top-1 (↓)
0.8	78.72 ± 9.42	0.621 ± 0.126	0.041 ± 0.019
1.0	77.58 ± 9.35	0.661 ± 0.129	0.030 ± 0.016
1.2	76.41 ± 9.30	0.692 ± 0.133	0.021 ± 0.014
1.4	75.32 ± 9.37	0.708 ± 0.136	0.019 ± 0.013
1.6	73.94 ± 9.45	0.717 ± 0.139	0.017 ± 0.012

D Complete Task-Level Results

This section reports the complete task-level results for the default AE-APL operating point $\beta = 1$. These results underlie the corresponding default-setting benchmark averages reported in the main paper. We separately report attack effectiveness and attribution exposure on five MaMuJoCo tasks and three SMAC scenarios. All values are reported as means $\pm$ sample standard errors over five independent runs.

D.1 Task-Level Comparison with Existing Attacks

Tables 6 and 7 present the complete task-level comparison with existing attacks. The results separately report attack effectiveness and attribution exposure, thereby showing whether the benchmark-level improvements are consistently supported across individual environments.

On MaMuJoCo, AE-APL with $\beta = 1$ achieves JQ-NR values between 0.552 and 0.839 and limits the Top-1 attribution rate to at most 0.100 across all five tasks. The attribution reduction is particularly pronounced on Humanoid-17x1, Humanoid-9xbody, and HumanoidStandup-17x1. Walker2d-6x1 remains the most challenging continuous-control setting, exhibiting the lowest JQ-NR and the largest residual Top-1 rate among the five tasks.

On SMAC, AE-APL with $\beta = 1$ achieves JQ-NR values of 0.889, 0.964, and 0.817 on 3s6z, 7h_vs_8z, and 1m2m8m_vs_1m2m7m, respectively. The compromised agent is never ranked first by the post-training Joint-Q auditor in any of the three reported scenarios. The effectiveness–exposure trade-off is most visible on 1m2m8m_vs_1m2m7m, where AE-APL attains substantially lower attribution exposure but a higher attacked-team win rate than several disruption-oriented baselines.

Table 6: Task-level comparison with existing attacks on MaMuJoCo. Attack effectiveness is measured by return drop, where higher values indicate stronger attacks. Higher JQ-NR and lower Top-1 indicate lower attribution exposure. Values are reported as means ± sample SEM.

Task / Scenario	Method	Attack Effectiveness	JQ-NR (↑)	Top-1 (↓)
HalfCheetah-6x1	AMI	97.69 ± 5.54	0.877 ± 0.071	0.000 ± 0.000
	GMA	121.55 ± 0.19	0.245 ± 0.034	0.000 ± 0.000
	ICLR	121.07 ± 1.34	0.283 ± 0.047	0.000 ± 0.000
	ICML	121.86 ± 0.57	0.341 ± 0.067	0.000 ± 0.000
	USENIX	139.70 ± 10.62	0.425 ± 0.108	0.000 ± 0.000
	AE-APL ($\beta = 1$)	121.62 ± 0.28	0.839 ± 0.047	0.000 ± 0.000
Humanoid-17x1	AMI	26.90 ± 1.98	0.000 ± 0.000	0.993 ± 0.007
	GMA	51.31 ± 0.22	0.158 ± 0.020	0.047 ± 0.047
	ICLR	51.97 ± 0.32	0.284 ± 0.061	0.000 ± 0.000
	ICML	51.17 ± 0.41	0.150 ± 0.016	0.027 ± 0.013
	USENIX	44.39 ± 0.42	0.404 ± 0.106	0.000 ± 0.000
	AE-APL ($\beta = 1$)	50.68 ± 0.26	0.699 ± 0.054	0.000 ± 0.000
Humanoid-9xbody	AMI	6.10 ± 1.44	0.188 ± 0.015	0.053 ± 0.007
	GMA	51.73 ± 0.92	0.073 ± 0.012	0.620 ± 0.046
	ICLR	51.58 ± 0.42	0.210 ± 0.057	0.340 ± 0.129
	ICML	51.55 ± 0.89	0.150 ± 0.093	0.500 ± 0.186
	USENIX	50.78 ± 0.78	0.105 ± 0.023	0.567 ± 0.077
	AE-APL ($\beta = 1$)	50.96 ± 0.92	0.688 ± 0.064	0.000 ± 0.000
HumanoidStandup-17x1	AMI	46.59 ± 0.21	0.318 ± 0.153	0.013 ± 0.013
	GMA	46.32 ± 0.36	0.522 ± 0.078	0.060 ± 0.023
	ICLR	46.76 ± 0.75	0.222 ± 0.004	0.033 ± 0.033
	ICML	45.37 ± 0.59	0.241 ± 0.092	0.047 ± 0.037
	USENIX	44.31 ± 0.68	0.348 ± 0.150	0.047 ± 0.013
	AE-APL ($\beta = 1$)	42.80 ± 2.25	0.682 ± 0.099	0.007 ± 0.007
Walker2d-6x1	AMI	120.37 ± 0.74	0.037 ± 0.031	0.827 ± 0.059
	GMA	123.29 ± 1.06	0.392 ± 0.051	0.200 ± 0.070
	ICLR	123.08 ± 1.83	0.071 ± 0.012	0.847 ± 0.024
	ICML	125.22 ± 0.81	0.443 ± 0.062	0.187 ± 0.074
	USENIX	117.47 ± 1.46	0.078 ± 0.053	0.867 ± 0.103
	AE-APL ($\beta = 1$)	115.97 ± 0.36	0.552 ± 0.136	0.100 ± 0.053

Table 7: Task-level comparison with existing attacks on SMAC. Attack effectiveness is measured by win rate, where lower values indicate stronger attacks. Higher JQ-NR and lower Top-1 indicate lower attribution exposure. Values are reported as means ± sample SEM.

Task / Scenario	Method	Attack Effectiveness	JQ-NR (↑)	Top-1 (↓)
3s6z	AMI	0.200 ± 0.011	0.394 ± 0.027	0.200 ± 0.073
	GMA	0.252 ± 0.017	0.401 ± 0.015	0.250 ± 0.117
	ICLR	0.233 ± 0.013	0.263 ± 0.017	0.493 ± 0.117
	ICML	0.200 ± 0.015	0.178 ± 0.006	0.468 ± 0.050
	USENIX	0.240 ± 0.013	0.156 ± 0.006	0.583 ± 0.044
	AE-APL ($\beta = 1$)	0.172 ± 0.017	0.889 ± 0.049	0.000 ± 0.000
7h_vs_8z	AMI	0.000 ± 0.000	0.128 ± 0.046	0.400 ± 0.173
	GMA	0.000 ± 0.000	0.025 ± 0.017	0.829 ± 0.100
	ICLR	0.000 ± 0.000	0.019 ± 0.011	0.883 ± 0.067
	ICML	0.000 ± 0.000	0.106 ± 0.032	0.583 ± 0.083
	USENIX	0.000 ± 0.000	0.036 ± 0.014	0.783 ± 0.083
	AE-APL ($\beta = 1$)	0.050 ± 0.000	0.964 ± 0.007	0.000 ± 0.000
1m2m8m_vs_1m2m7m	AMI	0.400 ± 0.050	0.645 ± 0.168	0.000 ± 0.000
	GMA	0.291 ± 0.123	0.477 ± 0.142	0.121 ± 0.026
	ICLR	0.283 ± 0.093	0.477 ± 0.189	0.100 ± 0.076
	ICML	0.400 ± 0.076	0.235 ± 0.097	0.350 ± 0.076
	USENIX	0.300 ± 0.010	0.180 ± 0.019	0.383 ± 0.017
	AE-APL ($\beta = 1$)	0.417 ± 0.017	0.817 ± 0.031	0.000 ± 0.000

D.2 Task-Level Transfer across Attribution Auditors

Tables 8 and 9 report the compromised agent's normalized rank under seven independently fitted post-training attribution auditors. These auditors cover counterfactual value comparison through JQ, Local JQ, and COMA; coalition-based marginal attribution through Shapley and JQ-SHAQ; and action-sensitivity attribution through Grad and IG. All auditor scores are oriented such that a higher normalized rank indicates lower attribution exposure.

On MaMuJoCo, AE-APL with $\beta = 1$ achieves the highest or a near-highest normalized rank for most task–auditor combinations. The improvements are particularly consistent on Humanoid-17x1, Humanoid-9xbody, and HumanoidStandup-17x1, where AE-APL substantially exceeds the disruption-oriented baselines across value-based, coalition-based, and gradient-based attribution. On HalfCheetah-6x1, AE-APL remains highly ranked under all seven auditors, although AMI attains a higher JQ and COMA rank. Walker2d-6x1 is again the most difficult task: AE-APL improves the rank under all seven views relative to most baselines, but its gradient-based rank is lower than on the other tasks.

On SMAC, AE-APL achieves the highest normalized rank for every scenario–auditor combination reported in Table 9. The improvement therefore extends beyond the independently fitted Joint-Q auditor used for the primary evaluation and transfers consistently to local counterfactual, coalition, and gradient-based attribution methods. This cross-auditor consistency supports the conclusion that AE-APL reduces the compromised agent's broader attribution prominence rather than overfitting to a single post-training auditor.

Table 8: Task-level normalized rank of the compromised agent under seven independently fitted post-training attribution auditors on MaMuJoCo. Values are means ± sample SEM. Higher values indicate lower attribution exposure.

Task	Method	JQ	Local JQ	Shapley	JQ-SHAQ	COMA	Grad	IG
HalfCheetah-6x1	AMI	0.877 ± 0.071	0.708 ± 0.022	0.701 ± 0.030	0.703 ± 0.041	0.996 ± 0.004	0.723 ± 0.077	0.717 ± 0.055
	GMA	0.245 ± 0.034	0.207 ± 0.007	0.212 ± 0.012	0.207 ± 0.007	0.387 ± 0.039	0.380 ± 0.019	0.280 ± 0.070
	ICLR	0.283 ± 0.047	0.203 ± 0.003	0.212 ± 0.010	0.200 ± 0.005	0.352 ± 0.084	0.269 ± 0.065	0.233 ± 0.026
	ICML	0.341 ± 0.067	0.253 ± 0.024	0.235 ± 0.018	0.212 ± 0.006	0.424 ± 0.076	0.336 ± 0.064	0.285 ± 0.057
	USENIX	0.425 ± 0.108	0.364 ± 0.119	0.341 ± 0.129	0.340 ± 0.130	0.499 ± 0.170	0.469 ± 0.174	0.380 ± 0.132
	AE-APL ($\beta = 1$)	0.839 ± 0.047	0.869 ± 0.051	0.808 ± 0.024	0.816 ± 0.022	0.920 ± 0.049	0.929 ± 0.067	0.911 ± 0.051
Humanoid-17x1	AMI	0.000 ± 0.000	0.006 ± 0.003	0.001 ± 0.000	0.001 ± 0.001	0.000 ± 0.000	0.032 ± 0.006	0.006 ± 0.002
	GMA	0.158 ± 0.020	0.195 ± 0.031	0.144 ± 0.027	0.145 ± 0.020	0.145 ± 0.022	0.253 ± 0.069	0.175 ± 0.065
	ICLR	0.284 ± 0.061	0.316 ± 0.069	0.300 ± 0.128	0.272 ± 0.100	0.217 ± 0.046	0.330 ± 0.096	0.303 ± 0.156
	ICML	0.150 ± 0.016	0.203 ± 0.027	0.147 ± 0.018	0.144 ± 0.019	0.144 ± 0.021	0.249 ± 0.054	0.133 ± 0.044
	USENIX	0.404 ± 0.106	0.471 ± 0.112	0.415 ± 0.103	0.379 ± 0.100	0.340 ± 0.081	0.450 ± 0.078	0.444 ± 0.105
	AE-APL ($\beta = 1$)	0.699 ± 0.054	0.682 ± 0.023	0.629 ± 0.074	0.675 ± 0.077	0.752 ± 0.057	0.674 ± 0.045	0.787 ± 0.005
Humanoid-9xbody	AMI	0.188 ± 0.015	0.234 ± 0.062	0.196 ± 0.047	0.216 ± 0.042	0.284 ± 0.109	0.244 ± 0.013	0.233 ± 0.047
	GMA	0.073 ± 0.012	0.131 ± 0.010	0.070 ± 0.024	0.060 ± 0.010	0.076 ± 0.012	0.383 ± 0.004	0.333 ± 0.042
	ICLR	0.210 ± 0.057	0.175 ± 0.053	0.147 ± 0.045	0.134 ± 0.043	0.196 ± 0.068	0.439 ± 0.033	0.464 ± 0.073
	ICML	0.150 ± 0.093	0.216 ± 0.129	0.128 ± 0.067	0.115 ± 0.066	0.163 ± 0.073	0.422 ± 0.020	0.378 ± 0.030
	USENIX	0.105 ± 0.023	0.122 ± 0.010	0.071 ± 0.010	0.063 ± 0.013	0.095 ± 0.025	0.426 ± 0.020	0.376 ± 0.051
	AE-APL ($\beta = 1$)	0.688 ± 0.064	0.675 ± 0.055	0.720 ± 0.041	0.725 ± 0.060	0.703 ± 0.052	0.662 ± 0.061	0.669 ± 0.047
HumanoidStandup-17x1	AMI	0.318 ± 0.153	0.345 ± 0.133	0.336 ± 0.165	0.333 ± 0.166	0.427 ± 0.196	0.281 ± 0.115	0.333 ± 0.177
	GMA	0.522 ± 0.078	0.560 ± 0.062	0.516 ± 0.100	0.493 ± 0.083	0.542 ± 0.072	0.495 ± 0.129	0.528 ± 0.103
	ICLR	0.222 ± 0.004	0.252 ± 0.009	0.279 ± 0.039	0.253 ± 0.024	0.310 ± 0.029	0.253 ± 0.010	0.248 ± 0.038
	ICML	0.241 ± 0.092	0.282 ± 0.105	0.260 ± 0.078	0.226 ± 0.075	0.303 ± 0.091	0.573 ± 0.034	0.517 ± 0.139
	USENIX	0.348 ± 0.150	0.361 ± 0.149	0.322 ± 0.135	0.293 ± 0.139	0.318 ± 0.138	0.300 ± 0.043	0.321 ± 0.096
	AE-APL ($\beta = 1$)	0.682 ± 0.099	0.665 ± 0.082	0.593 ± 0.126	0.596 ± 0.109	0.750 ± 0.083	0.693 ± 0.060	0.776 ± 0.061
Walker2d-6x1	AMI	0.037 ± 0.031	0.105 ± 0.025	0.069 ± 0.022	0.061 ± 0.022	0.061 ± 0.028	0.001 ± 0.001	0.000 ± 0.000
	GMA	0.392 ± 0.051	0.484 ± 0.090	0.281 ± 0.061	0.305 ± 0.072	0.304 ± 0.047	0.037 ± 0.012	0.027 ± 0.013
	ICLR	0.071 ± 0.012	0.097 ± 0.042	0.045 ± 0.014	0.047 ± 0.009	0.011 ± 0.009	0.007 ± 0.007	0.001 ± 0.001
	ICML	0.443 ± 0.062	0.493 ± 0.083	0.395 ± 0.070	0.408 ± 0.081	0.257 ± 0.043	0.043 ± 0.007	0.035 ± 0.009
	USENIX	0.078 ± 0.053	0.053 ± 0.042	0.047 ± 0.043	0.049 ± 0.045	0.099 ± 0.079	0.027 ± 0.027	0.015 ± 0.013
	AE-APL ($\beta = 1$)	0.552 ± 0.136	0.584 ± 0.110	0.555 ± 0.164	0.556 ± 0.168	0.548 ± 0.152	0.197 ± 0.120	0.595 ± 0.172

E Experimental Details and Hyperparameters

This section provides the experimental configurations required to reproduce the reported results. We summarize the benchmark settings, adversarial-policy optimization parameters, Joint-Q critic configuration, exposure-related hyperparameters, and evaluation protocol.

Table 9: Task-level normalized rank of the compromised agent under seven independently fitted post-training attribution auditors on SMAC. Values are means ± sample SEM. Higher values indicate lower attribution exposure.

Scenario	Method	JQ	Local JQ	Shapley	JQ-SHAQ	COMA	Grad	IG
3s6z	AMI	0.394 ± 0.027	0.402 ± 0.027	0.406 ± 0.047	0.408 ± 0.045	0.385 ± 0.027	0.410 ± 0.027	0.329 ± 0.037
	GMA	0.401 ± 0.015	0.428 ± 0.013	0.407 ± 0.008	0.398 ± 0.010	0.386 ± 0.015	0.432 ± 0.013	0.341 ± 0.009
	ICLR	0.263 ± 0.017	0.276 ± 0.016	0.259 ± 0.014	0.249 ± 0.010	0.248 ± 0.017	0.278 ± 0.015	0.213 ± 0.013
	ICML	0.178 ± 0.006	0.193 ± 0.010	0.175 ± 0.017	0.173 ± 0.019	0.169 ± 0.006	0.187 ± 0.004	0.145 ± 0.004
	USENIX	0.156 ± 0.006	0.143 ± 0.008	0.135 ± 0.019	0.132 ± 0.022	0.108 ± 0.006	0.145 ± 0.008	0.099 ± 0.006
	AE-APL ($\beta = 1$)	0.889 ± 0.049	0.907 ± 0.036	0.894 ± 0.075	0.897 ± 0.079	0.889 ± 0.049	0.908 ± 0.046	0.842 ± 0.053
7h_vs_8z	AMI	0.128 ± 0.046	0.108 ± 0.032	0.119 ± 0.018	0.114 ± 0.025	0.128 ± 0.046	0.083 ± 0.022	0.097 ± 0.032
	GMA	0.025 ± 0.017	0.017 ± 0.009	0.004 ± 0.004	0.004 ± 0.004	0.025 ± 0.017	0.021 ± 0.004	0.004 ± 0.004
	ICLR	0.019 ± 0.011	0.019 ± 0.003	0.008 ± 0.005	0.006 ± 0.003	0.019 ± 0.011	0.022 ± 0.003	0.006 ± 0.003
	ICML	0.106 ± 0.032	0.117 ± 0.035	0.094 ± 0.011	0.089 ± 0.006	0.106 ± 0.032	0.094 ± 0.024	0.083 ± 0.005
	USENIX	0.036 ± 0.014	0.064 ± 0.024	0.064 ± 0.040	0.069 ± 0.034	0.036 ± 0.014	0.064 ± 0.024	0.033 ± 0.022
	AE-APL ($\beta = 1$)	0.964 ± 0.007	0.931 ± 0.019	0.953 ± 0.015	0.950 ± 0.022	0.964 ± 0.007	0.931 ± 0.020	0.944 ± 0.015
1m2m8m_vs_1m2m7m	AMI	0.645 ± 0.168	0.693 ± 0.192	0.690 ± 0.163	0.705 ± 0.168	0.645 ± 0.168	0.737 ± 0.141	0.558 ± 0.155
	GMA	0.477 ± 0.142	0.491 ± 0.171	0.401 ± 0.175	0.467 ± 0.119	0.621 ± 0.124	0.621 ± 0.178	0.363 ± 0.121
	ICLR	0.477 ± 0.189	0.530 ± 0.116	0.507 ± 0.205	0.495 ± 0.204	0.477 ± 0.189	0.537 ± 0.207	0.423 ± 0.160
	ICML	0.235 ± 0.097	0.272 ± 0.114	0.253 ± 0.076	0.257 ± 0.087	0.235 ± 0.097	0.280 ± 0.109	0.207 ± 0.080
	USENIX	0.180 ± 0.019	0.225 ± 0.046	0.203 ± 0.029	0.195 ± 0.015	0.180 ± 0.019	0.223 ± 0.050	0.165 ± 0.008
	AE-APL ($\beta = 1$)	0.817 ± 0.031	0.883 ± 0.015	0.835 ± 0.021	0.847 ± 0.019	0.817 ± 0.031	0.888 ± 0.019	0.737 ± 0.039

Table 10: Task-specific environment and training configurations. Agent identities in this table follow the zero-based indexing used by the implementation, whereas the mathematical exposition uses one-based indices. Each method is trained with seeds $\{0, 1, 2, 3, 4\}$. All attack methods use the same fixed MAPPO victim checkpoint and compromised-agent identity within each task.

Task / Scenario	Compromised Agent	Episode Horizon	Training Budget	Evaluation Episodes
HalfCheetah-6x1	2	1000	10M environment steps	300
Humanoid-17x1	0	1000	10M environment steps	300
Humanoid-9xbody	4	1000	10M environment steps	300
HumanoidStandup-17x1	0	1000	10M environment steps	300
Walker2d-6x1	5	1000	10M environment steps	300
3s6z	5	170	2M environment steps	300
7h_vs_8z	3	150	2M environment steps	300
1m2m8m_vs_1m2m7m	3	150	2M environment steps	300

E.1 Benchmark and Training Configurations

We evaluate AE-APL on five continuous-action MaMuJoCo tasks: HalfCheetah-6x1, Humanoid-17x1, Humanoid-9xbody, HumanoidStandup-17x1, and Walker2d-6x1. We additionally evaluate it on three discrete-action SMAC scenarios: 3s6z, 7h_vs_8z, and 1m2m8m_vs_1m2m7m.

For each task, we use a fixed pretrained benign team and replace the policy of one designated agent with the adversarial policy. The remaining benign policies are frozen throughout attack training. All compared methods use the same victim checkpoint, compromised-agent identity, training budget, and evaluation protocol within each task.

Table 10 reports the task-specific configurations. For MaMuJoCo, the training budget is measured in environment steps; for SMAC, it is measured in environment steps or episodes according to the implementation.

E.2 Optimization and Attribution Hyperparameters

The adversarial policy and its value function are optimized using PPO. The PPO value function estimates returns under the exposure-aware reward, whereas the centralized Joint-Q critic is trained separately using the original team reward. The Joint-Q critic is held fixed when counterfactual contributions and exposure scores are computed.

The main optimization and attribution hyperparameters are summarized in Table 11. Unless otherwise specified, the same configuration is used across tasks within each benchmark.

Table 11: Core AE-APL optimization and attribution hyperparameters.

Hyperparameter	MaMuJoCo	SMAC
Adversarial-policy architecture	two-layer MLP $(64, 64)$; ReLU–LayerNorm centralized MLP $(64, 64)$;	two-layer MLP $(64, 64)$; ReLU, masked softmax
PPO value-function architecture	two value branches	centralized MLP $(64, 64)$
Joint-Q critic architecture	state/action encoders: 256 each; fusion head: 256	state/action encoders: 256 each; alive-mask encoder: 128; head: 256
Policy learning rate	5×10^{-4}	1×10^{-4}
PPO value learning rate	5×10^{-3}	1×10^{-4}
Joint-Q learning rate	5×10^{-4}	3×10^{-4}
Discount factor γ	0.99	0.99
GAE parameter λ_{GAE}	0.95	0.95
PPO clipping coefficient	0.2	0.2
Entropy coefficient	0.01	0.01
PPO minibatches / minibatch size	40 / 800 transitions	1 / 32 episodes
PPO epochs per update	5	4
Rollout size per update	32,000 transitions	32 episodes
Joint-Q updates per policy update	12	2
Joint-Q target update	EMA, $\tau_Q = 0.01$	EMA, $\tau_Q = 0.01$
Maximum gradient norm	10	10
Default exposure weight β	1	1
Relative-exposure weight λ_{s}	0.4	0.4
Soft-rank temperature τ_{r}	0.50	0.50
Numerical stabilizer ϵ	10^{-6}	10^{-6}
Top-3 base weights $\bar{\boldsymbol{\omega}}$	$--$	$(0.5, 0.3, 0.2)$

E.3 Capability-Floor Configuration

Let J^e_{ICLR} denote the attacked-team performance achieved by the ICLR baseline in environment e. Here, J denotes the attacked-team return for MaMuJoCo and the attacked-team win rate for SMAC. Lower values indicate stronger attacks in both benchmarks.

Because MaMuJoCo attacked-team returns may be negative, directly multiplying J^e_{ICLR} by a factor larger than one does not always relax the capability requirement. We therefore use the sign-safe capability floor

$$J^e_{\mathrm{floor}} = J^e_{\mathrm{ICLR}} + (\rho - 1)\,|J^e_{\mathrm{ICLR}}|\,, \qquad \rho = 1.2. \tag{39}$$

This definition allows the current attacked-team performance to be at most 20% higher than the corresponding ICLR result, irrespective of whether J^e_{ICLR} is positive or negative. For nonnegative metrics, including SMAC win rates, Equation (39) reduces to $J^e_{\mathrm{floor}} = \rho J^e_{\mathrm{ICLR}}$.

At policy update u, the capability gate is

$$g^{(u,e)} = \mathbb{I}\left[\widehat{J}^{(u,e)}_{\mathrm{team}} \le J^e_{\mathrm{floor}}\right]. \tag{40}$$

The exposure penalty is activated only when the current adversarial policy satisfies this capability requirement. Increasing ρ weakly increases $J^e_{\mathrm{floor}}(\rho)$ for every environment and therefore consistently relaxes the lower-is-better capability requirement, including in tasks where the attacked-team return is negative.

E.4 Evaluation Protocol

For MaMuJoCo, attack effectiveness is measured by the percentage decrease in team return relative to the corresponding clean victim team:

$$\mathrm{ReturnDrop} = 100\frac{J_{\mathrm{clean}} - J_{\mathrm{attack}}}{|J_{\mathrm{clean}}|}. \tag{41}$$

A return drop above 100% can occur when the attacked team obtains a negative return while the clean-team return is positive. For SMAC, attack effectiveness is measured directly using the attacked-team win rate.

Each attack policy is trained using 5 independent runs. The final policy from each run is evaluated for 300 episodes in MaMuJoCo and 300 episodes in SMAC. Task-level results are reported as the mean and sample standard error:

$$\mathrm{SEM} = \frac{s}{\sqrt{n}}, \tag{42}$$

where s is the sample standard deviation across independent runs and n is the number of runs.

For attribution evaluation, each trained attack policy is frozen before the post-training auditors are fitted. The auditors use trajectory data collected independently of the attack-training trajectories. JQ-NR and Top-1 are computed using the independently fitted post-training Joint-Q auditor, while the cross-auditor analysis additionally evaluates Local JQ, Shapley, JQ-SHAQ, COMA, Grad, and IG.

All compared attacks use the same victim checkpoint and evaluation episodes within each task. Baseline hyperparameters follow their official implementations where available, while the training budget and compromised-agent identity are matched to AE-APL.

F Sensitivity to the Exposure Weight

The exposure weight β controls the relative importance of attribution evasion in the AE-APL objective. When $\beta = 0$, the exposure penalty vanishes and the method reduces to its disruption-only counterpart. As β increases, the adversarial policy places greater emphasis on reducing the compromised agent's attribution exposure, potentially at the cost of attack effectiveness.

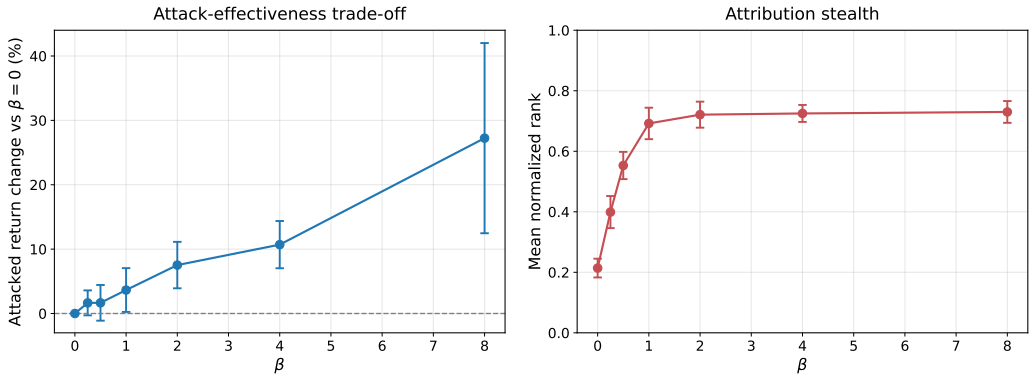


Figure 2: Sensitivity to the exposure weight β on MaMuJoCo. The left panel reports the attacked-return change relative to the disruption-only counterpart ($\beta = 0$), where a larger positive change indicates a weaker attack. The right panel reports the compromised agent's mean normalized attribution rank, where a higher value indicates lower attribution exposure. Error bars denote sample SEM.

Figure 2 shows that increasing β produces a clear trade-off between attack effectiveness and attribution exposure. Starting from $\beta = 0$, the mean normalized rank improves rapidly as the exposure penalty is introduced. In particular, increasing β to 1 raises the normalized rank from approximately 0.21 to 0.69, while the corresponding attacked-return change remains below 5%. This indicates that a moderate exposure penalty can substantially reduce the compromised agent's relative attribution prominence without severely weakening the attack.

Further increasing β continues to improve attribution stealth, but with diminishing returns. The normalized-rank gain becomes comparatively small beyond $\beta = 2$ and largely saturates around $\beta = 4$, whereas the attack-effectiveness cost continues to increase. At $\beta = 8$, the normalized rank remains close to that obtained with $\beta = 4$, but the attacked return increases substantially relative

to the disruption-only counterpart. Therefore, excessively large exposure weights primarily weaken the attack without providing a commensurate reduction in attribution exposure.

We use $\beta = 1$ as the default setting because it lies near the knee of the trade-off: it captures a large fraction of the attainable attribution-stealth improvement while incurring only a limited average effectiveness cost. We additionally report $\beta = 0.25$ in the main comparison as a lower-penalty operating point with attack effectiveness more closely matched to the strongest disruption-oriented baselines.

G Attribution Redistribution

Table 12: Attribution redistribution on MaMuJoCo. Share statistics aggregate seven post-training attribution views, while raw harmful contributions use the Joint-Q auditor. Benign HHI measures attribution concentration among benign agents. Paired changes from $\beta = 0$ to $\beta = 1$ are reported with 95% cluster-bootstrap intervals.

Metric	$\beta = 0$	$\beta = 1$	Paired change
Scale-free metrics (seven attribution views)			
Attacker share p_k	0.150	0.081	$-0.070\ [-0.102, -0.041]$
Median benign share	0.087	0.096	$+0.009\ [0.003, 0.016]$
Maximum benign share	0.220	0.251	$+0.031\ [0.012, 0.052]$
Normalized attribution entropy	0.939	0.932	$-0.007\ [-0.028, 0.012]$
Normalized benign HHI	0.045	0.055	$+0.010\ [0.001, 0.020]$
Unnormalized harmful contribution (primary JQ only)			
Attacker raw harm $\bar{c}_k$	3.144	1.063	$-2.081\ [-4.228, -0.211]$
Benign raw-harm sum $\sum_{j \neq k} \bar{c}_j$	20.711	18.608	$-2.103\ [-4.181, -0.307]$
Maximum benign raw harm $\max_{j \neq k} \bar{c}_j$	9.744	8.978	$-0.766\ [-1.508, -0.203]$

H Sensitivity to the Capability-Floor Multiplier

We study the sensitivity of AE-APL to the multiplier ρ used to determine the capability floor:

$$J^e_{\text{floor}}(\rho) = J^e_{\text{ICLR}} + (\rho - 1)\,|J^e_{\text{ICLR}}|\,. \tag{43}$$

where J^e_{ICLR} denotes the attacked-team performance achieved by ICLR in environment e. We evaluate $\rho \in \{0.8, 1.0, 1.2, 1.4, 1.6\}$ on MaMuJoCo while fixing $\beta = 1$ and keeping all other training and evaluation settings unchanged. Each value of ρ is trained independently using the same victim checkpoints, compromised-agent identities, training budgets, and random seeds as in the main experiments.

Table 5 shows that AE-APL is relatively stable over a moderate range of capability-floor multipliers. Increasing ρ relaxes the capability requirement and allows the exposure objective to be introduced under a less restrictive attack-effectiveness constraint. Consequently, the compromised agent's normalized rank generally increases and its Top-1 attribution rate decreases, while attack effectiveness gradually declines.

The most substantial attribution improvement occurs when ρ increases from 0.8 to 1.2. Further increasing ρ yields only limited gains in JQ-NR and Top-1, whereas the return drop continues to decrease. In particular, the results between $\rho = 1.0$ and $\rho = 1.4$ remain broadly consistent, indicating that AE-APL is not highly sensitive to small changes in the capability floor. We use $\rho = 1.2$ as the default because it provides a favorable balance between maintaining attack effectiveness and reducing attribution exposure.

H.1 Qualitative Analysis in 7h_vs_8z

Figure 3 presents selected frames extracted from Supplementary Video 1 to illustrate the learned attack behaviors in the SMAC scenario 7h_vs_8z. Under the disruption-oriented baseline, the

compromised agent largely disengages from combat and contributes little to the ongoing engagement. Such behavior can directly reduce team effectiveness, but it also leaves a visually conspicuous behavioral footprint because the compromised agent acts differently from its actively engaged teammates.

In contrast, AE-APL remains involved in the combat process. The compromised agent stays close to its teammates, continues attacking enemy units, and participates in enemy eliminations, while the team nevertheless fails to complete the scenario successfully. The resulting failure is therefore not associated with an immediately conspicuous withdrawal from combat. Instead, the compromised agent maintains apparently task-relevant behavior while influencing the joint coordination dynamics toward failure.

This qualitative observation is consistent with the post-training attribution results. On `7h_vs_8z`, AE-APL with $\beta = 1$ achieves a Joint-Q normalized rank of 0.964 ± 0.007 and a Top-1 attribution rate of 0.000 ± 0.000, while the attacked-team win rate remains 0.050 ± 0.000. Thus, the compromised agent is consistently ranked below the most responsible position by the post-training Joint-Q auditor, even though the attack continues to induce team failures.

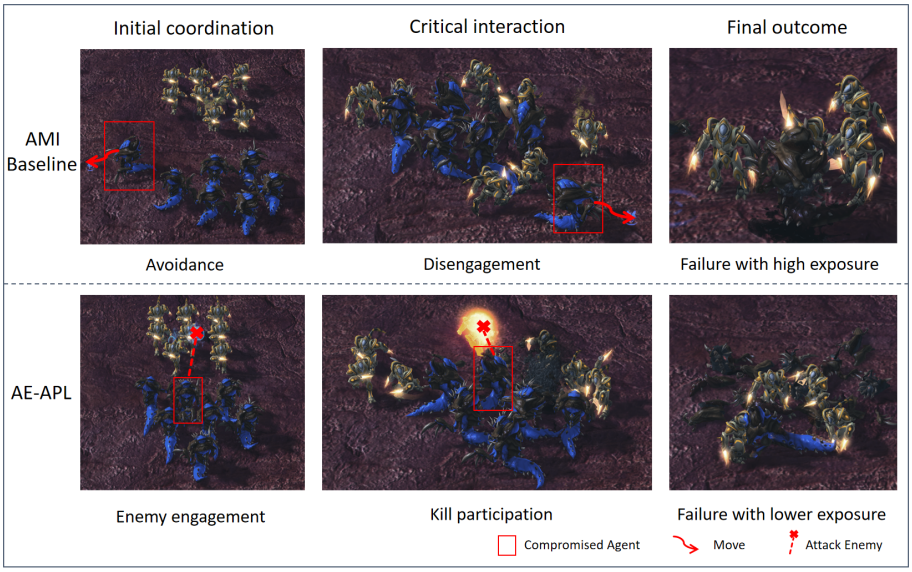


Figure 3: Qualitative comparison in the SMAC scenario `7h_vs_8z`, using selected frames extracted from Supplementary Video 1. The disruption-oriented baseline largely disengages from combat, producing an immediately conspicuous behavioral pattern. In contrast, AE-APL remains involved in enemy engagement and participates in enemy eliminations while the team still fails. The frames provide a qualitative illustration of the learned behaviors; the reported attribution metrics are computed separately using the post-training auditors.

Supplementary video. Supplementary Video 1 provides the complete evaluation trajectories corresponding to the qualitative behaviors summarized in Figure 3. The video is included only as a qualitative visualization and is not used to compute any of the attack effectiveness or attribution metrics reported in the paper.