MATBAK: TARGETED-BEHAVIOR ADVERSARIAL ATTACKS ON MULTIPLE REINFORCEMENT AGENTS VIA PREFERENCE-BASED INTENTION POLICIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Adversarial attacks provide a crucial lens for evaluating the robustness of multi-agent reinforcement learning (MARL) systems, revealing worst-case failures and security risks in multi-agent systems (MAS) driven by deep reinforcement learning methods. However, existing attacks on MARL systems primarily aim to degrade cumulative returns, capturing generic performance failures while overlooking attacker-specified collective behaviors that are critical in safety-sensitive MAS. In this work, we study targeted-behavior adversarial attacks against MARL systems, where an attacker seeks to induce preferred team-level behaviors via bounded observation perturbations. Unlike standard reward-degradation attacks, targeted-behavior attacks require translating high-level attacker intentions into coordinated behavioral changes among interacting agents. We propose Multi-Agent Targeted-Behavior Adversarial Attack (MatBak), a preference-based framework for eliciting attacker-preferred collective behaviors in MARL systems. MatBak first learns an intention policy from attacker preferences. It employs a victim-observation encoder to evaluate attacker utility, then optimizes an observation-perturbation adversary via a dual-weighted bi-level mechanism measuring both agent-level engagement and state-level occupancy, enabling adaptive victim selection, targeted credit assignment, and fine-grained behavior manipulation. Across SMAC and MetaDrive tasks with MAT, MAPPO, and QMIX agents, MatBak achieves the highest attacker return in 26 of 27 settings; when used for adversarial training, it reduces average attacker return by 22.1% and improves average environment return by 11.4% over the strongest non-MatBak baselines.

1 INTRODUCTION

Evaluating the vulnerabilities of multi-agent reinforcement learning (MARL) (Tan, 1993; Zhang et al., 2021b; Wong et al., 2023) systems under adversarial attacks (Huang et al., 2017; Pattanaik et al., 2017; Sun et al.) is essential for safety-critical deployments (Yan et al., 2024). Specifically, when an adversary injects observation perturbations (Zhang et al., 2020; 2021a) into a subset of agents within the victim multi-agent system (MAS) (Dorri et al., 2018), the perturbations can propagate through the system to impair other agents (Castelfranchi et al., 1992; Hannoun et al., 1998) and trigger sub-optimal collective performance (Teo et al., 2013; Altmann et al., 2024). This propagation effect is particularly critical under targeted-behavior attacks (Sayyed et al., 2024; Bai et al., 2025), where strategically crafted perturbations can mislead the coordinated policy and induce cascading failures beyond generic reward degradation (Yan et al., 2024). As illustrated in Figure 1, perturbing one or a few vehicles in an MARL-driven traffic MAS can induce a collective traffic standstill that blocks a time-sensitive ambulance en route to a hospital. Therefore, understanding such vulnerabilities (Li et al., 2025) and developing effective countermeasures have become key research priorities.

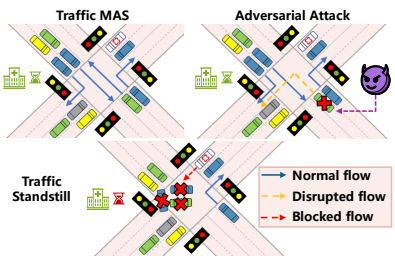

Figure 1: Targeted-behavior adversarial attacks on MARL agents.

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.

In the single-agent domain, targeted-behavior adversarial attacks have been addressed by learning preference-aligned intention policies that capture attacker intent and training adversaries to imitate them (Bai et al., 2025). However, extending such efforts to multi-agent settings is non-trivial. On the one hand, complex inter-agent interactions (Castelfranchi et al., 1992; Hannoun et al., 1998) obscure the effects of collective behavior manipulation, and the state-occupancy measure established by existing single-agent methods (Bai et al., 2025) proves insufficient to provide fine-grained, agent-level learning signals for adversary training. On the other hand, the emergent nature of MAS (Teo et al., 2013; Altmann et al., 2024) complicates the assignment of attacker-specified utilities to individual victims when targeting only a subset of agents, thereby posing challenges for adversary victim selection. Consequently, the vulnerabilities of MARL systems under targeted-behavior adversarial attacks remain inadequately evaluated, and effective countermeasures are largely underexplored.

In this work, we aim to evaluate the vulnerabilities of MARL systems under targeted-behavior adversarial attacks and enhance system robustness against such attacks. We propose Multi-Agent Targeted-Behavior Adversarial Attack (MatBak), a novel observation-adversarial attack method designed to elicit targeted behaviors in MARL systems via preference-based reinforcement learning (PbRL) Christiano et al. (2017); Ibarz et al. (2018). MatBak first incorporates a victim-observation encoder that estimates the per-agent attacker utility to enable adaptive victim selection. Subsequently, it optimizes the adversarial policy to align with the learned intention policy through bi-level optimization, employing an efficient dual-weighting mechanism that estimates both the victim-specific individual engagement measure and the state occupancy measure. To further strengthen the robustness of MARL agents against targeted-behavior attacks, we also develop a variant of MatBak to facilitate adversarial training (Pinto et al., 2017; Zhang et al., 2021a). To the best of our knowledge, this is the first work to study targeted-behavior adversarial attacks in multi-agent settings and systematically evaluate the vulnerabilities of various types of MARL systems. Extensive experiments across multiple MARL benchmarks have validated the effectiveness of our method.

2 Related Works

Observation-adversarial attacks on DRL agents. Observation-adversarial attacks deceive DRL agents by perturbing their observations (Huang et al., 2017). One line of research generates such perturbations from policy gradients: (Huang et al., 2017) applies the Fast Gradient Sign Method (FGSM) (Goodfellow et al., 2015), and (Pattanaik et al., 2017) back-propagates the worst-action cross-entropy loss. Another line of research formulates adversary training as a *Markov Decision Process* (MDP) (Puterman, 1990) and incorporates DRL: SA-RL (Zhang et al., 2021a) trains an adversary via end-to-end RL, while PA-AD (Sun et al.) learns state perturbations via director-actor collaboration to determine the perturbation direction. In multi-agent settings, (Lin et al., 2020) crafts adversarial perturbations to identify and enforce the most damaging action for a victim agent, while (Zan et al., 2023) dynamically clusters victim agents by their features and crafts perturbations on one group for transfer to others. Although these methods are highly effective and widely used for worst-case security analysis (Ilahi et al., 2021), the reliance on victims' internal information such as training reward, predetermined target state, or target policy limits their practical applicability.

To overcome this limitation, prior research in the single-agent domain has focused on methods that mainly rely on the feedback from deployed DRL models rather than the victims' internal information. Representative approaches include exploiting the transferability of adversarial examples via surrogate models (Huang et al., 2017), training adversarial policies that manipulate victim observations through natural actions in a shared environment (Gleave et al., 2020), and generating state perturbations with preference-aligned intention policies to induce targeted behaviors (Bai et al., 2025). However, extending such methods to the multi-agent setting remains largely underexplored. Following this practical setting, the proposed MatBak method performs observation-adversarial attacks **solely utilizing deployed model feedback** (e.g., victims' action distributions under perturbed observations). Without requiring access to victims' internal information (e.g., training-phase gradients or rewards), MatBak enjoys practical feasibility of attacking deployed MARL systems.

Preference-based reinforcement learning. PbRL (Christiano et al., 2017; Ibarz et al., 2018; Lee et al., 2021; Park et al., 2022) enables the integration of human preferences into the learning dynamics of RL agents. (Christiano et al., 2017) fits a neural reward model by querying pairwise human preferences between pairs of trajectory segments. (Ibarz et al., 2018) leverages expert demonstra-

tions and learns reward from pairwise human trajectory preferences to train DQN (Mnih et al., 2015) agents. (Lee et al., 2021) reduces the cost of extensive human feedback via unsupervised exploration and reward relabeling, and (Park et al., 2022) further enhances human feedback efficiency through semi-supervised reward learning and data augmentation. In this study, we employ PbRL to extract attacker intent and train the intention policy which serves as a learning target of the adversary.

3 PRELIMINARIES

3.1 MARL PROBLEM FORMULATION

Formulating the victim policy. We formulate the learning model of the cooperative MARL victims as *Markov Games* (MG) (Littman, 1994) represented by a tuple $\langle \mathcal{N}, \boldsymbol{\mathcal{O}}, \boldsymbol{\mathcal{A}}, R, P, \gamma \rangle$, which has been widely adopted in the literature (Wen et al., 2022; Hu et al., 2025). Here, $\mathcal{N} = \{1, \ldots, n\}$ denotes the agent set, $\boldsymbol{\mathcal{O}} = \prod_{i=1}^{n} \mathcal{O}^i$ denotes the joint observation space, $\boldsymbol{\mathcal{A}} = \prod_{i=1}^{n} \mathcal{A}^i$ denotes the joint action space, and $\gamma \in [0, 1)$ denotes the discount factor. The shared reward is described by the reward function $R : \boldsymbol{\mathcal{O}} \times \boldsymbol{\mathcal{A}} \longrightarrow \mathbb{R}$, and the system dynamics are captured by the transition function $P : \boldsymbol{\mathcal{O}} \times \boldsymbol{\mathcal{A}} \times \boldsymbol{\mathcal{O}} \longrightarrow [0, 1]$. Notations and descriptions are summarized in Appendix A.

Analysis of MG. In MG, each agent $i \in \mathcal{N}$ perceives $o_t^i \in \mathcal{O}^i$ at time step t and selects $a_t^i \in \mathcal{A}^i$ by its own policy $\pi_i(\cdot|o_t^i)$. The (unnormalized) marginal observation distribution emerges from the transition kernel P in conjunction with the joint policy $\boldsymbol{\pi}$, which is defined as $\rho_{\boldsymbol{\pi}}(\cdot) \triangleq \sum_0^\infty \gamma^t \Pr(\boldsymbol{o}_t = \boldsymbol{o}|\boldsymbol{\pi})$. After interacting with the environment, the whole team obtains a shared reward $R_t(\boldsymbol{o}_t, \boldsymbol{a}_t)$ and perceives a subsequent joint observation $\boldsymbol{o}_{t+1}$ sampled by $P(\cdot|\boldsymbol{o}_t, \boldsymbol{a}_t)$. The objective of MG is to maximize the infinite-horizon discounted cumulative return given by $R^\gamma \triangleq \sum_0^\infty \gamma^t R_t(\boldsymbol{o}_t, \boldsymbol{a}_t)$.

3.2 REWARDED OBSERVATION-ADVERSARIAL MARKOV GAMES

Formulating the threat model. To study observation-adversarial attacks in multi-agent settings, inspired by the RSA-MDP model (Bai et al., 2025), we introduce a centralized adversary and formalize *Rewarded Observation-Adversarial Markov Games* (ROA-MG). Specifically, an ROA-MG is defined by a tuple $\langle \mathcal{N}, \boldsymbol{\mathcal{O}}, \boldsymbol{\mathcal{A}}, \mathcal{B}, \hat{R}, P, \gamma \rangle$, where $\mathcal{N} = \{1, \ldots, n\}$ denotes the set of n victim agents, $\boldsymbol{\mathcal{O}} = \prod_{i=1}^{n} \mathcal{O}^i$ denotes victim agents' joint observation space, $\boldsymbol{\mathcal{A}} = \prod_{i=1}^{n} \mathcal{A}^i$ denotes victim agents' joint action space, $\hat{R} : \boldsymbol{\mathcal{O}} \times \boldsymbol{\mathcal{A}} \longrightarrow \mathbb{R}$ denotes the reward function evaluating the performance of victim agents under a preference-based adversarial perspective, $P : \boldsymbol{\mathcal{O}} \times \boldsymbol{\mathcal{A}} \times \boldsymbol{\mathcal{O}} \longrightarrow [0, 1]$ denotes victim agents' observation transition dynamics, and $\gamma \in [0, 1)$ denotes the discount factor.

In this research, we study observation-adversarial attacks against deployed victim MAS that follow a set of fixed, stationary policies $\boldsymbol{\pi}_\nu = \{\pi^1, \ldots, \pi^n\}$, where each $\pi_\nu^i : \mathcal{O}^i \to \mathcal{P}(\mathcal{A}^i)$ is implemented via a function approximator parameterized by ν. **We assume that the adversary has access to feedback from deployed victim models** (e.g., action distributions under perturbed observations). This assumption does not compromise algorithmic practicality, as such feedback can be estimated via black-box probing (e.g., conducting repeated queries and applying probability regression analysis) without relying on white-box access to internal information such as gradients or rewards. Specifically, the adversary α is equipped with a stochastic policy $\pi_\alpha : \boldsymbol{\mathcal{O}} \to \mathcal{P}(\widehat{\mathcal{A}})$, who perceives victim agents' joint observation $\boldsymbol{o} \in \boldsymbol{\mathcal{O}}$ and takes action $\widehat{a} \in \widehat{\mathcal{A}}$ to perturb $\boldsymbol{o}$ into $\tilde{\boldsymbol{o}}$ constrained by $\mathcal{B}(\boldsymbol{o})$. Here, $\mathcal{B}(\boldsymbol{o})$ is defined as $\{\tilde{\boldsymbol{o}} \in \boldsymbol{\mathcal{O}} : \|\boldsymbol{o} - \tilde{\boldsymbol{o}}\| \leq \epsilon\}$, where $\epsilon = \varepsilon \cdot \max_{o_i, o_j \in \mathcal{O}} |o_i - o_j|$ enforces a bound on the adversary's attack capacity and ε denotes the perturbation budget. The adversary is trained to output bounded Gaussian noises $\boldsymbol{\delta} \in \Delta$ (where Δ is the perturbation space) satisfying $\|\boldsymbol{\delta}\|_\infty \leq 1$ and perturb victim agents' joint observation via $\tilde{\boldsymbol{o}} = \boldsymbol{o} + \epsilon\boldsymbol{\delta}$. The victim agents then take actions according to the perturbed joint observation $\tilde{\boldsymbol{o}}$. We use $\boldsymbol{\pi}_{\nu \circ \alpha}$ to represent the perturbed policy induced by the adversary policy π_α, i.e., $\boldsymbol{\pi}_{\nu \circ \alpha}(\cdot \mid \boldsymbol{o}) = \boldsymbol{\pi}_\nu(\cdot \mid \pi_\alpha(\boldsymbol{o})), \forall \boldsymbol{o} \in \boldsymbol{\mathcal{O}}$.

Analysis of ROA-MG. ROA-MG learns an optimal adversary π_α^* which guides the victim agents to maximize the cumulative preference-based reward measured by $\hat{R}^\gamma \triangleq \sum_0^\infty \gamma^t \hat{R}_t(\boldsymbol{o}_t, \boldsymbol{a}_t)$. We provide theoretical analysis to show that learning the optimal adversary in an ROA-MG is equivalent to solving for an optimal policy in a standard MDP, which is available in Appendix C.

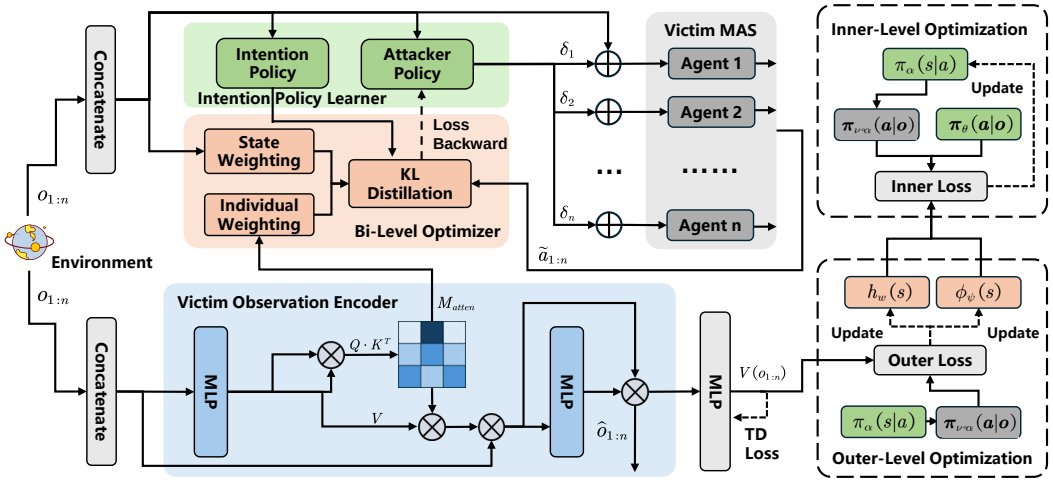

Figure 2: The overall framework of MatBak, which comprises three core modules: (i) the intention policy learner; (ii) the victim-observation encoder; and (iii) the bi-level optimizer. Specifically, the intention policy learner fits a preference-based reward model and trains an intention policy via RL; the victim-observation encoder encodes victims' local observations and derives attacker utility (i.e., per-agent importance scores); and the bi-level optimizer trains the adversary to align the attacked victim policy with the intention policy following a dually-weighted KL objective.

4 METHODOLOGY

In this section, we present a detailed overview of the proposed Multi-Agent Targeted-Behavior Adversarial Attack (MatBak) framework. As illustrated in Figure 2, MatBak comprises three core modules that are trained alternately: (i) the intention policy learner; (ii) the victim-observation encoder; and (iii) the bi-level optimizer, which are detailed as follows.

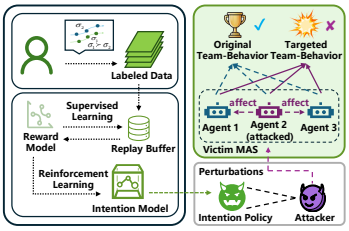

Figure 3: Learning the intention policy via PbRL for targeted-behavior adversarial attacks.

4.1 LEARNING THE INTENTION POLICY VIA PBRL

To induce an attacker-specified team-level targeted behavior within the victim MAS, as illustrated in Figure 3, MatBak learns an **intention policy** π_θ following the general principle of PbRL, which bridges the attacker's abstract intentions and concrete action-level victim behaviors. Specifically, human experts first label preferences over the victim trajectory segments consisting of observation-action pairs. Notably, we employ a scripted teacher here for simplicity, which is a common practice (Liu et al., 2022; Bai et al., 2025). The scripted teacher annotates preferences based on the degree to which each trajectory segment exhibits the attacker-specified targeted behavior. Let a length-k trajectory segment be $\sigma = \{(\boldsymbol{o}_t, \boldsymbol{a}_t), (\boldsymbol{o}_{t+1}, \boldsymbol{a}_{t+1}), \ldots, (\boldsymbol{o}_{t+k-1}, \boldsymbol{a}_{t+k-1})\}$, where $\boldsymbol{o}_t \in \boldsymbol{\mathcal{O}}$ and $\boldsymbol{a}_t \in \boldsymbol{\mathcal{A}}$ are generated under adversarial attacks by $\boldsymbol{a}_t \sim \boldsymbol{\pi}_\nu(\cdot \mid \pi_\alpha(\boldsymbol{o}_t))$. Given a segment pair (σ^0, σ^1), the scripted teacher annotates a preference label $y \in \{(1,0), (0,1), (0.5, 0.5)\}$, which represents a distribution specifying the probability of a segment being preferred. Following the Bradley-Terry (Bradley & Terry, 1952) model, we fit a reward model $\hat{r}_\zeta(\boldsymbol{o}, \boldsymbol{a})$ via

$$P_\zeta[\sigma^0 \succ \sigma^1] = \frac{\exp \sum_t \hat{r}_\zeta(\boldsymbol{o}_t^0, \boldsymbol{a}_t^0)}{\sum_{i\in\{0,1\}} \exp \sum_t \hat{r}_\zeta(\boldsymbol{o}_t^i, \boldsymbol{a}_t^i)}, \tag{1}$$

and optimize ζ by minimizing the cross-entropy loss

$$\mathcal{L}(\zeta) = -\mathbb{E}_{(\sigma^0,\sigma^1,y)\sim\mathcal{D}_\sigma}\Big[\sum_{i=0}^{1} y(i) \log P_\zeta[\sigma^i \succ \sigma^{1-i}]\Big], \tag{2}$$

where $\mathcal{D}_\sigma$ is a dataset of triplets (σ^0, σ^1, y) containing segment pairs and corresponding preference labels. The learned reward model $\hat{r}_\zeta$ then serves as an intent-aligned target-behavior detector that provides reward signals for training the intention policy by MARL algorithms.

4.2 Extracting Mutual Attention via Victim-Observation Encoder

To enable precise and efficient attacks, MatBak employs a **victim-observation encoder** E_ξ that estimates victim-specific attacker utility, i.e., the individual significance of each victim agent from the attacker's perspective. Given a joint observation $\boldsymbol{o} = (o^1, \ldots, o^n)$, the encoder applies stacked self-attention blocks and outputs hidden observation representations $H^{(\ell)}(\boldsymbol{o}) = [h^1, \ldots, h^n] \in \mathbb{R}^{n \times d}$, where d denotes the embedding dimension. For each self-attention layer ℓ, the attention matrix is calculated by

$$M_{\text{atten}}^{(\ell)}(\boldsymbol{o}) = \text{softmax}\Big(\frac{Q^{(\ell)} {K^{(\ell)}}^\top}{\sqrt{d_k}}\Big) \in \mathbb{R}^{n \times n}, \tag{3}$$

where $Q^{(\ell)} = H^{(\ell-1)} W_Q$, $K^{(\ell)} = H^{(\ell-1)} W_K$. We use $M_{\text{atten}}^{(-1)}$ (i.e., the last-layer output) as mutual-attention evidence. The encoder is trained to fit the joint observation value function $V_\xi(\boldsymbol{o})$ by minimizing *Temporal Difference* (TD) loss on the attacked rollouts:

$$\mathcal{L}_{\text{TD}}(\xi) = \mathbb{E}_{(\boldsymbol{o}_t, \boldsymbol{a}_t, \hat{r}_t, \boldsymbol{o}_{t+1}) \sim \mathcal{D}} \Big[\big(\hat{r}_t + \gamma V_{\bar{\xi}}(\boldsymbol{o}_{t+1}) - V_\xi(\boldsymbol{o}_t) \big)^2 \Big], \tag{4}$$

where $\hat{r}_t = \hat{r}_\zeta(\boldsymbol{o}_t, \boldsymbol{a}_t)$, $\bar{\xi}$ denotes the target network parameters, and $\mathcal{D}$ denotes the replay buffer. Attacker utility scores are computed by column-wise averaging and linear normalization:

$$u_i(\boldsymbol{o}) = \frac{1}{n} \sum_{j=1}^{n} M_{\text{atten}}^{ji}(\boldsymbol{o}), \quad w_i(\boldsymbol{o}) = \frac{u_i(\boldsymbol{o})}{\sum_{k=1}^{n} u_k(\boldsymbol{o})}. \tag{5}$$

The core insight is that high attention indicates a key contribution to shaping the joint value, which measures the effectiveness of inducing targeted behavior. At each time step t, the target set with capacity of k is determined by $\mathcal{I}_k(\boldsymbol{o}_t) = \text{Top-}k(\{w_i(\boldsymbol{o}_t)\}_{i=1}^n)$, enabling implementations of precise perturbations on the identified crucial agents within the victim MAS (i.e., $\tilde{o}_t^i = o_t^i$ for $i \notin \mathcal{I}_k(\boldsymbol{o}_t)$).

4.3 Learning the Adversary via Bi-Level Optimization

To enhance the effectiveness of targeted-behavior adversarial attacks and stabilize training processes, MatBak learns a centralized adversary π_α with a dual-weighting mechanism comprising (i) an individual-weighting module ϕ_ψ that evaluates the victim-specific individual engagement measure; and (ii) a state-weighting module h_ω that adjusts the state occupancy measure. As illustrated in Figure 2, we utilize a **bi-level optimizer** to orchestrate updating the adversary policy and the weighting functions through both inner and outer optimization processes. With a fixed intention policy π_θ, the overall optimization objective of MatBak can be formulated as:

$$\min_{\omega, \psi} \ \mathcal{J}_\pi(\hat{\alpha}(\omega, \psi)) \quad \text{s.t. } \hat{\alpha}(\omega, \psi) = \arg\min_{\alpha} \ \mathcal{L}(\alpha; \omega, \theta, \psi). \tag{6}$$

Inner-level optimization. The inner loop aims to learn an optimal adversary. Specifically, the inner-level objective $\mathcal{L}$ aligns the perturbed victim policy $\boldsymbol{\pi}_{\nu \circ \alpha}$ with the intention policy $\boldsymbol{\pi}_\theta$ through the following dually-weighted *Kullback-Leibler* (KL) divergence loss:

$$\mathcal{L}(\alpha; \omega, \theta, \psi) = \mathbb{E}_{\boldsymbol{o} \sim \mathcal{D}} \Big[h_\omega(\boldsymbol{o}) \sum_{i=1}^{n} \phi_\psi^i(\boldsymbol{o}) \, D_{\text{KL}} \Big(\pi_{\nu \circ \alpha}^{(i)}(\cdot \mid \boldsymbol{o}) \,\big\|\, \pi_\theta^{(i)}(\cdot \mid \boldsymbol{o}) \Big) \Big], \tag{7}$$

where $\boldsymbol{o} \in \boldsymbol{\mathcal{O}}$, $\pi_{\nu \circ \alpha}^{(i)}(\cdot \mid \boldsymbol{o}) \triangleq \pi_\nu^{(i)}(\cdot \mid \pi_\alpha(\boldsymbol{o}))$, and $\mathcal{D}$ denotes the replay buffer sampled by a mixture behavior policy combining $\boldsymbol{\pi}_{\nu \circ \alpha}$ and $\boldsymbol{\pi}_\theta$.

Outer-level optimization. The outer loop aims to learn an efficient dual-weighting mechanism which identifies the importance of victim individuals (i.e., attacker utility) and states. Specifically, the weighting functions are trained by the outer-level loss $\mathcal{J}_\pi$, which uses the joint observation-value

estimator learned within the victim-observation encoder E_ξ to compute a clipped proximal policy optimization (PPO) (Schulman et al., 2017) objective:

$$\mathcal{J}_\pi(\alpha(\omega,\psi)) = \mathop{\mathbb{E}}_{\substack{\boldsymbol{o}_t\sim\mathcal{D},\\ \boldsymbol{a}_t\sim\boldsymbol{\pi}_{\nu\circ\alpha(\omega,\psi)}}}\left[-\sum_{t=0}^{T-1}\min\left(\Lambda_t(\alpha)\hat{A}_t, \mathrm{clip}(\Lambda_t(\alpha),1\pm\epsilon_{\mathrm{clip}})\hat{A}_t\right)\right]. \tag{8}$$

In Eq. (8), $\Lambda_t(\alpha) = \frac{\boldsymbol{\pi}_{\nu\circ\alpha}(\boldsymbol{a}_t|\boldsymbol{o}_t)}{\boldsymbol{\pi}^{\mathrm{old}}_{\nu\circ\alpha}(\boldsymbol{a}_t|\boldsymbol{o}_t)}$ denotes the importance ratio, ϵ_{clip} denotes the PPO clipping parameter, and $\hat{A}_t$ estimates the joint advantage value via *Generalized Advantage Estimation* (GAE) (Schulman et al., 2016) as

$$\hat{A}_t = \sum_{l=0}^{\infty}(\gamma\lambda)^l\delta^{\mathrm{TD}}_{t+l}, \quad \delta^{\mathrm{TD}}_t = \hat{r}_t + \gamma V_\xi(\boldsymbol{o}_{t+1}) - V_\xi(\boldsymbol{o}_t), \tag{9}$$

where λ regulates the residual decay. For algorithmic implementation, we adopt a one-step approximation for the inner-level optimization:

$$\hat{\alpha}(\omega,\psi) \approx \alpha_t - \eta_t\,\nabla_\alpha\mathcal{L}(\alpha;\omega,\theta,\psi)|_{\alpha_t}, \tag{10}$$

which induces an explicit dependence of $\hat{\alpha}$ on ω and ψ and enables gradient-based outer-level updates. We explicitly derive the outer-loop gradients w.r.t. ω and ψ in Appendix D.

Leveraging the bi-level optimizer, MatBak efficiently orchestrates updates between (i) descending $\nabla_\alpha\mathcal{L}$ to learn α and (ii) descending $\nabla_\omega\mathcal{J}_\pi$ and $\nabla_\psi\mathcal{J}_\pi$ to learn (ω,ψ), thereby yielding an adversary that induces a preferred joint victim policy aligned with $\boldsymbol{\pi}_\theta$ while adaptively emphasizing critical joint states and crucial victim individuals. The pseudocode of MatBak is provided in Appendix B.

Theoretical analysis. We provide theoretical analysis for the convergence of bi-level optimization in Appendix E. Specifically, under assumptions of L-smoothness, bounded gradients, and Lipschitz-continuous inner gradients etc., we show that the outer-level gradients w.r.t. ω and ψ converge to zero using Theorems E.3 and E.4, and that the expected squared gradient norm of the inner-level optimization process converges to zero under step-size assumptions using Theorem E.5.

4.4 Adversarial Training for Robust MARL under Targeted-Behavior Attacks

A constructive application of adversarial attack methods lies in enhancing the robustness of learning algorithms via adversarial training. For example, (Zhang et al., 2021a) establishes a generic framework which alternates training with learned adversaries (ATLA). In addition, (Liang et al., 2022) investigates worst-case-aware robust RL (WocaR-RL), which directly estimates and optimizes the worst-case reward of victim agents under bounded adversarial attacks.

Drawing inspiration from these works, in this section, we explore the potential of MatBak in facilitating robust MARL training by orchestrating updates of the (joint) victim policy $\boldsymbol{\pi}_\nu$ and the MatBak attacker policy π_α. Specifically, we formulate adversarial training objective $\mathcal{G}$ as the following min-max problem on ROA-MG rollouts:

$$\mathcal{G}: \min_\nu \max_\alpha \mathbb{E}_{\tau\sim\boldsymbol{\pi}_{\nu\circ\alpha}}\Big[\sum_{t\geq 0}\gamma^t\big(\lambda_{\mathrm{atk}}\cdot\hat{r}_\zeta(\boldsymbol{o}_t,\boldsymbol{a}_t)-\bar{r}_t\big)\Big], \tag{11}$$

where λ_{atk} is the adversary-training coefficient, $\tilde{\boldsymbol{o}}_t = \pi_\alpha(\boldsymbol{o}_t)$, $\boldsymbol{a}_t \sim \boldsymbol{\pi}_\nu(\cdot \mid \tilde{\boldsymbol{o}}_t)$, and $\boldsymbol{o}_{t+1} \sim P(\cdot \mid \boldsymbol{o}_t, \boldsymbol{a}_t)$. In each iteration, we fix α and train ν following Eq. (11) using MARL algorithms on the attacked trajectories and introduce two practical instantiations to strengthen the robustness of MARL: (i) ***MatBak-ATLA***, which updates $\boldsymbol{\pi}_\nu$ with $\bar{r}_t = r_{\mathrm{env}}(\boldsymbol{o}_t,\boldsymbol{a}_t)$, and (ii) ***MatBak-WocaR***, which updates $\boldsymbol{\pi}_\nu$ with $\bar{r}_t = r_{\mathrm{env}}(\boldsymbol{o}_t,\boldsymbol{a}_t) - \gamma V_\theta(\boldsymbol{o}_{t+1})$ to optimize the worst-case targeted-behavior return estimated by the value estimator of the adversary intention policy $\boldsymbol{\pi}_\theta$.

5 Experiments

We design experiments to address four research questions (RQs): **RQ1:** How does MatBak compare with baselines? **RQ2:** Can MatBak induce targeted behavior in the victim MAS? **RQ3:** Does MatBak enhance MARL robustness? **RQ4:** What are the contributions of its essential components?

5.1 EXPERIMENTAL SETUP

Environments. We evaluate MatBak on two widely used MARL benchmarks: StarCraft Multi-Agent Challenge (SMAC) (Samvelyan et al., 2019) and MetaDrive (Li et al., 2022), which encompass open-world cooperation tasks with heterogeneous agents and partial observability. Detailed benchmark descriptions are available in Appendix F.2. Regarding **attacker intention**, on SMAC, we characterize it as inducing a targeted dispersed-formation behavior that hinders coordination and leaves agents vulnerable to being defeated individually; on MetaDrive, we characterize it as inducing a targeted traffic-standstill behavior. A detailed description is provided in Appendix F.3.

Compared methods. We compare MatBak (ours) with advanced baselines including Adv-ZS (Gleave et al., 2020), Adv-TP (Guo et al., 2021), Adv-xAI (Wu et al., 2021), SA-RL (Zhang et al., 2021a), PA-AD (Sun et al.), RAT (Bai et al., 2025), and AMI (Li et al., 2025). Following (Li et al., 2025), we extend single-agent baselines to multi-agent settings by replacing the original single victim with a group of cooperative victims, and train attackers with both MAPPO (Yu et al., 2022) (Adv-ZS, Adv-TP, Adv-xAI, AMI) and PPO (Schulman et al., 2017) (SA-RL, PA-AD, RAT, MatBak) to compare the effectiveness of multi-agent and single-agent PPO attackers. For fair comparison, all methods share the same threat model setup and attack a limited set of 40% victim agents.

Implementation settings. Since targeted behaviors are typically not expressible as explicit rewards in practice, we employ both **ground-truth targeted-behavior alignment rewards** (on SMAC) and **reward signals derived from a learned reward model** $\hat{r}_\zeta$ to measure the alignment of victim behavior with the attacker's intention. All methods share the same reward learning setup and follow the principle of PEBBLE (Lee et al., 2021) to learn reward models with the same number of preference labels. We follow the foundational settings of PbRL to expedite the training process, using a scripted teacher that provides accurate preference labels to simulate human feedback (Liu et al., 2022; Bai et al., 2025). Unless explicitly specified, we employ oracle versions of the non-PbRL baselines, which are enhanced by using the ground-truth targeted-behavior alignment reward during adversary training. We adopt the **average attacker return** (i.e., the ground-truth targeted-behavior alignment rewards) as the evaluation metric, which directly measures the targeted-behavior-oriented attacker efficacy. Further implementation details are available in Appendix F.1.

Table 1: Adversarial attack results on SMAC (reported as average attacker return ± standard deviation). Column headers show different maps and MARL algorithms used to train the victims.

	Method	**8m** (homogeneous)			**MMM** (heterogeneous)			**3s5z** (heterogeneous, Hard)		
		MAT	MAPPO	QMIX	MAT	MAPPO	QMIX	MAT	MAPPO	QMIX
Oracle	Adv-ZS (Gleave et al., 2020)	47.66 ±0.76	59.69 ±1.71	58.69 ±7.05	131.78 ±9.23	128.25 ±4.24	144.42 ±24.45	200.61 ±2.86	190.48 ±4.69	249.57 ±7.26
	Adv-TP (Guo et al., 2021)	48.32 ±1.12	58.99 ±1.70	57.52 ±2.66	127.38 ±7.70	128.37 ±2.19	133.52 ±15.14	207.28 ±5.14	192.77 ±5.35	243.80 ±11.23
	Adv-xAI (Wu et al., 2021)	48.26 ±1.03	57.76 ±0.57	59.80 ±5.40	131.87 ±6.73	128.22 ±3.76	143.35 ±17.42	202.01 ±3.85	191.29 ±5.45	241.68 ±16.35
	AMI (Li et al., 2025)	48.18 ±1.18	59.61 ±4.54	56.09 ±4.00	125.31 ±1.79	129.65 ±2.92	140.12 ±17.09	203.30 ±5.67	190.12 ±2.96	246.86 ±11.99
	SA-RL (Zhang et al., 2021a)	49.00 ±0.62	254.22 ±2.97	74.41 ±3.15	130.76 ±3.26	264.23 ±2.96	202.06 ±12.33	228.10 ±8.91	306.64 ±1.02	224.69 ±1.85
	PA-AD (Sun et al.)	49.32 ±0.21	248.27 ±4.15	56.50 ±1.19	139.34 ±3.36	265.90 ±1.79	211.22 ±14.85	234.37 ±3.28	307.36 ±6.12	248.14 ±8.32
PbRL	Adv-ZS (Gleave et al., 2020)	49.21 ±1.73	62.82 ±1.50	58.49 ±3.82	124.13 ±4.67	126.65 ±6.17	124.18 ±3.09	200.32 ±1.10	189.47 ±3.49	248.28 ±9.36
	Adv-TP (Guo et al., 2021)	48.88 ±2.09	60.92 ±2.59	58.42 ±3.42	124.13 ±4.67	125.78 ±3.74	126.10 ±4.74	201.30 ±5.70	187.93 ±3.24	248.28 ±9.36
	Adv-xAI (Wu et al., 2021)	49.18 ±1.54	60.62 ±4.31	58.40 ±3.31	125.80 ±4.58	127.35 ±4.49	127.61 ±2.06	199.73 ±2.69	189.81 ±3.24	250.41 ±2.96
	AMI (Li et al., 2025)	48.76 ±1.18	61.59 ±2.13	58.51 ±3.73	123.89 ±4.91	126.56 ±5.17	127.61 ±4.27	203.35 ±3.16	186.18 ±3.08	253.92 ±7.24
	SA-RL (Zhang et al., 2021a)	48.18 ±0.53	252.53 ±2.49	72.94 ±3.07	128.54 ±2.04	264.47 ±1.33	194.47 ±15.86	219.96 ±4.56	306.01 ±2.31	218.60 ±0.75
	PA-AD (Sun et al.)	49.09 ±0.33	252.61 ±3.23	56.90 ±0.94	139.84 ±2.84	266.69 ±1.83	215.67 ±14.63	236.41 ±2.13	310.34 ±4.59	246.19 ±7.77
	RAT (Bai et al., 2025)	284.68 ±30.68	359.52 ±24.69	114.07 ±42.20	488.35 ±60.08	438.40 ±39.25	489.10 ±101.60	718.30 ±28.99	441.64 ±13.28	541.72 ±131.33
	MatBak (Ours)	**379.60 ±74.63**	**553.38 ±135.80**	**585.99 ±229.58**	**517.20 ±43.73**	**565.38 ±106.38**	**623.80 ±263.78**	**742.93 ±73.58**	**630.85 ±65.62**	**731.40 ±188.58**

5.2 PERFORMANCE COMPARISON

Manipulation of targeted-behavior adversarial attacks on multiple SMAC agents. We evaluate MatBak and baselines by attacking MARL agents trained by MAT (CTCE, policy-based, sequential decision-making), MAPPO (CTDE, policy-based, simultaneous decision-making), and QMIX (CTDE, value-based, simultaneous decision-making) on 8m (homogeneous), MMM (heterogeneous), and 3s5z (heterogeneous, hard) maps of SMAC. As illustrated in Table 1, we provide both oracle and PbRL results of the baseline methods that are not originally designed for PbRL settings.

Table 1 shows that MatBak outperforms all baselines, with evident gains over RAT. Since RAT directly adapts existing single-agent targeted-behavior adversarial attacks to MARL settings and leverages similar preference-aligned intention policy with state-occupancy adjustment, this result provides direct evidence for the efficacy of the essential mechanisms of MatBak. The higher standard deviation alongside higher attacker returns arises from the discretized nature of targeted-behavior

alignment feedback, where marginal trajectory differences near the targeted behavior thresholds cause abrupt reward changes. Further results on SMAC including the MMM2 map and **attack success rates** defined on dispersed-formation behavior are reported in Appendix F.4.1, which show that MatBak effectively elicits the targeted behavior pre-defined by the attacker in the victim MAS, ruling out confounds (e.g., reward hacking) that could inflate the attacker return.

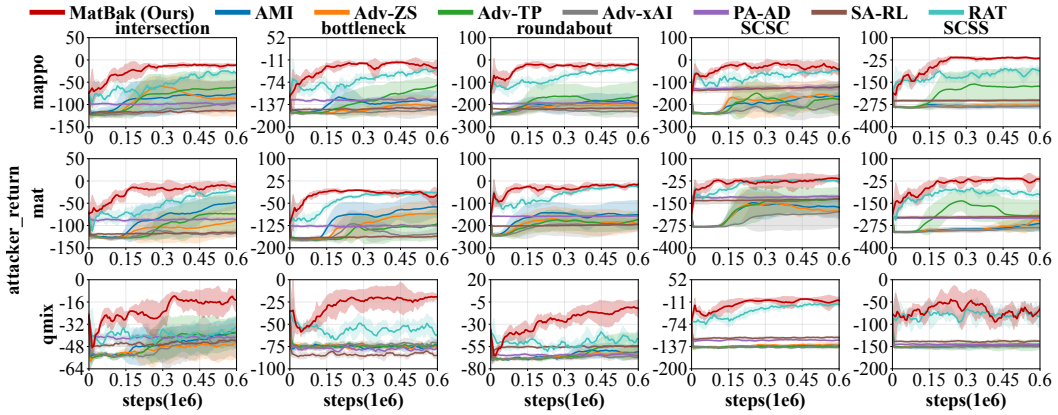

Figure 4: Adversarial attack results on MetaDrive (reported as average attacker return ± standard deviation). Figure headers show different scenarios and MARL algorithms used to train the victims.

Manipulation of targeted-behavior adversarial attacks on multiple MetaDrive agents. We attack MARL agents trained by MAT, MAPPO, and QMIX in intersection, bottleneck, roundabout, SCSC, and SCSS scenarios on MetaDrive. As shown in Figure 4, MatBak consistently outperforms all baselines in attacker return. While RAT achieves comparable performance in some scenarios, its lack of explicit utility modeling and fine-grained dual weighting leads to lower sample efficiency (e.g., mat_intersection, mappo_roundabout) and greater learning instability (e.g., mappo_SCSC). Further results on MetaDrive including **attack success rates** defined on traffic standstill behavior and **robust attack performance under observation noises** are reported in Appendix F.4.1.

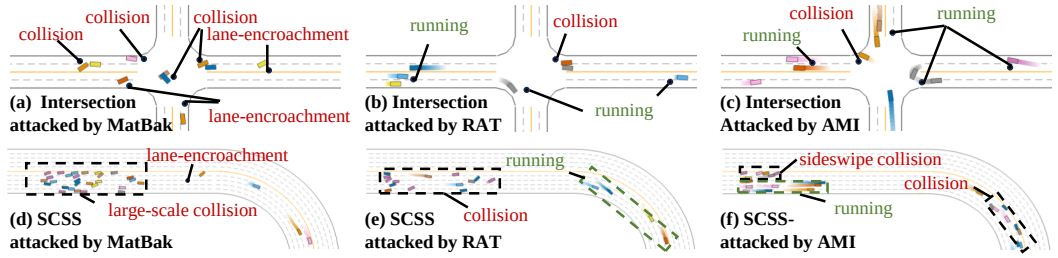

Figure 5: A visualization of targeted-behavior attacks conducted by different methods on MetaDrive.

5.3 Case Study

Figure 5 presents a fine-grained case study visualization in the intersection and SCSS scenarios on MetaDrive. The victims are trained by MAT. It can be observed that, under the same condition of attacking 40% victims, MatBak induces larger-scale traffic standstill than AMI and RAT, thus validating its effectiveness and indicating potential safety risks in real-world traffic systems. Further visualization results on MetaDrive including the SCSC scenario are reported in Appendix F.4.3.

5.4 Robust Adversarial Training and Evaluation

We conduct cross-validation experiments to validate the effectiveness of MatBak-ATLA and MatBak-WocaR in enhancing MARL robustness against targeted-behavior attacks. We train MAT victims on the 3s5z map of SMAC following ATLA and WocaR paradigms against the attacks of

Table 2: Performance of ATLA and WocaR models trained by different methods on the 3s5z map of SMAC. The victim models are trained by MAT, and the results are averaged across 100 episodes.

Model	Adv-ZS	Adv-TP	Adv-xAI	AMI	SA-RL	PA-AD	RAT	MatBak	Avg Env R	Avg Attack R
Adv-ZS-ATLA (Gleave et al., 2020)	9.38±4.64	8.90±4.43	8.94±4.24	8.18±4.10	9.41±3.89	10.18±2.51	7.58±2.15	5.16±1.96	8.47±3.49	395.88±156.11
Adv-TP-ATLA (Guo et al., 2021)	11.75±5.21	**13.29±5.27**	**13.33±5.52**	13.43±5.15	10.31±4.22	11.42±3.70	7.20±1.30	8.86±2.69	11.20±4.13	340.87±100.93
Adv-xAI-ATLA (Wu et al., 2021)	10.29±5.26	10.66±4.26	9.98±5.03	9.72±4.72	10.65±3.88	11.01±3.43	6.15±1.64	8.76±2.83	9.65±3.88	419.95±152.79
AMI-ATLA (Li et al., 2025)	11.15±4.08	11.44±3.85	10.18±4.12	10.29±3.98	9.59±3.42	9.98±2.99	8.60±2.46	8.23±3.09	9.93±3.50	394.25±169.82
SA-RL-ATLA (Zhang et al., 2021a)	6.88±1.35	13.00±1.52	10.03±1.56	7.73±1.50	8.91±1.30	10.98±2.01	10.30±1.84	9.04±1.64	9.61±1.59	326.91±114.64
PA-AD-ATLA (Sun et al.)	11.19±3.99	11.52±3.60	10.45±3.47	11.26±3.59	12.06±2.95	10.54±4.34	10.76±3.43	8.45±3.63	10.78±3.63	328.71±116.30
RAT-ATLA (Bai et al., 2025)	11.21±4.23	10.13±4.41	11.66±3.72	11.72±3.51	9.17±4.58	9.68±3.32	8.71±4.22	11.66±2.20	10.49±3.78	323.26±116.45
RAT-WocaR (Bai et al., 2025)	13.05±4.48	10.83±5.14	11.36±4.89	12.85±4.27	11.90±4.35	10.16±3.31	**14.21±4.67**	10.69±3.86	11.88±4.37	392.18±156.53
MatBak-ATLA (Ours)	**13.67±3.87**	12.77±3.78	13.03±4.30	**13.61±3.95**	**13.03±4.13**	**11.46±2.71**	13.14±4.77	15.10±4.43	**13.23±3.99**	**251.83±93.64**
MatBak-WocaR (Ours)	13.00±3.69	11.42±3.83	12.38±3.83	11.98±3.85	12.21±4.20	10.33±2.05	10.15±2.75	**15.71±4.87**	12.15±3.63	264.36±108.06

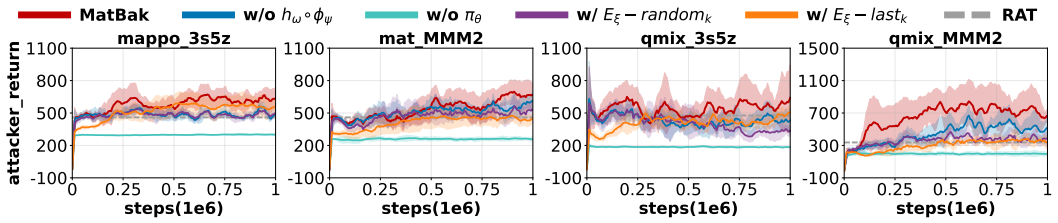

Figure 6: Comparison results of MatBak and its variants ablating $h_\omega \circ \phi_\psi$, $\boldsymbol{\pi}_\theta$, and E_ξ on SMAC.

all methods, and then evaluate the victims' performance under the attacks of these methods. Table 2 shows that MatBak-ATLA and MatBak-WocaR achieve the highest average environment return (Avg Env R) and the lowest average attacker return (Avg Attack R), indicating stronger robustness. Further details as well as **attack-resistance visualizations** are available in Appendix F.4.2.

5.5 Ablation Studies

We ablate three essential components of MatBak: the victim-observation encoder E_ξ, the dual-weighting controller $h_\omega \circ \phi_\psi$, and the intention policy $\boldsymbol{\pi}_\theta$. We utilize two SMAC maps: 3s5z and MMM2, with victim agents trained by MAT, MAPPO, and QMIX. To ablate E_ξ, we evaluate two variants: $E_\xi - \text{random}_k$, which selects k victims randomly, and $E_\xi - \text{last}_k$, which selects k victims with the lowest attention scores. MatBak w/o $h_\omega \circ \phi_\psi$ replaces the dual-weighting with uniform weights, and MatBak w/o $\boldsymbol{\pi}_\theta$ directly uses reward-model feedback as the attacker training objective.

As illustrated in Figure 6, MatBak outperforms all ablated variants, confirming the necessity of its essential components. Notably, MatBak w/o $\boldsymbol{\pi}_\theta$ exhibits the most pronounced performance degradation, emphasizing the significance of intention learning and alignment. Meanwhile, MatBak w/ $E_\xi - \text{last}_k$ outperforms MatBak w/ $E_\xi - \text{random}_k$ in most cases. We attribute this to target consistency, since the last_k strategy allows the adversary to learn specialized perturbations for the less-attended agents, whereas random_k forces a generic strategy due to constantly shifting targets. Crucially, MatBak (i.e., w/ $E_\xi - \text{top}_k$) consistently outperforms both variants (w/ $E_\xi - \text{random}_k$ and $E_\xi - \text{last}_k$), confirming the effectiveness of the attention-based utility assignment mechanism.

6 Conclusion

In this work, we aim to explore the vulnerabilities of MARL systems under targeted-behavior adversarial attacks and strengthen their robustness against such attacks. We propose MatBak, the first framework that elicits attacker-preferred targeted collective behaviors by perturbing victim observations in MARL systems. MatBak explicitly aligns adversarial attacks with attacker preference via PbRL and enhances attack efficiency by estimating both the (victim's) individual-engagement measure and the state-occupancy measure. With MatBak, we systematically evaluate the vulnerabilities of diverse MARL systems trained by distinct types of MARL methods. Extensive experiments demonstrate MatBak's effectiveness, and its adversarial training variants strengthen MARL robustness against such attacks. Since targeted-behavior attacks can induce harmful collective outcomes such as traffic congestion, responsible deployment is essential. We demonstrate MatBak's positive utility for adversarial training, and advocate for further research on detection mechanisms and defensive strategies to mitigate the dual-use risks of such attacks in real-world MARL systems.

AI USE STATEMENT

Generative AI tools were used solely for grammar checking and language polishing during the preparation of this manuscript, and the authors take full responsibility for its entire content.

ETHICS STATEMENT

This work investigates adversarial attack methods on MARL systems to expose their vulnerabilities and motivate the development of effective defenses. We acknowledge the dual-use risk of the proposed method (MatBak), as it can effectively induce preference-guided targeted collective behaviors in such systems for both beneficial and malicious purposes. To mitigate this risk, we develop adversarial training methods (MatBak-ATLA and MatBak-WocaR) that demonstrably strengthen MARL robustness against such attacks. Additionally, all experiments in this work are conducted on publicly available simulated benchmarks and do not involve any human subjects or sensitive data.

REFERENCES

Philipp Altmann, Julian Schönberger, Steffen Illium, Maximilian Zorn, Fabian Ritz, Tom Haider, Simon Burton, and Thomas Gabor. Emergence in multi-agent systems: A safety perspective. In *International Symposium on Leveraging Applications of Formal Methods*, pp. 104–120. Springer, 2024.

Fengshuo Bai, Runze Liu, Yali Du, Ying Wen, and Yaodong Yang. Rat: Adversarial attacks on deep reinforcement agents for targeted behaviors. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 15453–15461, 2025.

Ralph Allan Bradley and Milton E Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

Cristiano Castelfranchi, Maria Miceli, and Amedeo Cesta. Dependence relations among autonomous agents. *Decentralized AI*, 3:215–227, 1992.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.

Ali Dorri, Salil S Kanhere, and Raja Jurdak. Multi-agent systems: A survey. *Ieee Access*, 6:28573–28593, 2018.

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

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *International Conference on Learning Representations*, 2015.

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.

Mahdi Hannoun, Jaime Simao Sichman, Olivier Boissier, and Claudette Sayettat. Dependence relations between roles in a multi-agent system: Towards the detection of inconsistencies in organization. In *International Workshop on Multi-Agent Systems and Agent-Based Simulation*, pp. 169–182. Springer, 1998.

Jian Hu, Siyang Jiang, Seth Austin Harding, Haibin Wu, and Shih-wei Liao. Rethinking the implementation tricks and monotonicity constraint in cooperative multi-agent reinforcement learning. *arXiv preprint arXiv:2102.03479*, 2021.

Kun Hu, Muning Wen, Xihuai Wang, Shao Zhang, Yiwei Shi, Minne Li, Minglong Li, and Ying Wen. Pmat: Optimizing action generation order in multi-agent reinforcement learning. In *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems*, pp. 997–1005, 2025.

Sandy Huang, Nicolas Papernot, Ian Goodfellow, Yan Duan, and Pieter Abbeel. Adversarial attacks on neural network policies. *arXiv preprint arXiv:1702.02284*, 2017.

Borja Ibarz, Jan Leike, Tobias Pohlen, Geoffrey Irving, Shane Legg, and Dario Amodei. Reward learning from human preferences and demonstrations in atari. *Advances in neural information processing systems*, 31, 2018.

Inaam Ilahi, Muhammad Usama, Junaid Qadir, Muhammad Umar Janjua, Ala Al-Fuqaha, Dinh Thai Hoang, and Dusit Niyato. Challenges and countermeasures for adversarial attacks on deep reinforcement learning. *IEEE Transactions on Artificial Intelligence*, 3(2):90–109, 2021.

Kimin Lee, Laura M Smith, and Pieter Abbeel. Pebble: Feedback-efficient interactive reinforcement learning via relabeling experience and unsupervised pre-training. In *International Conference on Machine Learning*, pp. 6152–6163. PMLR, 2021.

Quanyi Li, Zhenghao Peng, Lan Feng, Qihang Zhang, Zhenghai Xue, and Bolei Zhou. Metadrive: Composing diverse driving scenarios for generalizable reinforcement learning. *IEEE transactions on pattern analysis and machine intelligence*, 45(3):3461–3475, 2022.

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.

Yongyuan Liang, Yanchao Sun, Ruijie Zheng, and Furong Huang. Efficient adversarial training without attacking: Worst-case-aware robust reinforcement learning. *Advances in neural information processing systems*, 35:22547–22561, 2022.

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.

Michael L Littman. Markov games as a framework for multi-agent reinforcement learning. In *Machine learning proceedings 1994*, pp. 157–163. Elsevier, 1994.

Runze Liu, Fengshuo Bai, Yali Du, and Yaodong Yang. Meta-reward-net: Implicitly differentiable reward learning for preference-based reinforcement learning. *Advances in Neural Information Processing Systems*, 35:22270–22284, 2022.

Julien Mairal. Stochastic majorization-minimization algorithms for large-scale optimization. *Advances in Neural Information Processing Systems*, 26, 2013.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

Yurii Nesterov. Introductory lectures on convex programming volume i: Basic course. *Lecture notes*, 3(4):5, 1998.

Jongjin Park, Younggyo Seo, Jinwoo Shin, Honglak Lee, Pieter Abbeel, and Kimin Lee. Surf: Semi-supervised reward learning with data augmentation for feedback-efficient preference-based reinforcement learning. In *10th International Conference on Learning Representations, ICLR 2022*, 2022.

Anay Pattanaik, Zhenyi Tang, Shuijing Liu, Gautham Bommannan, and Girish Chowdhary. Robust deep reinforcement learning with adversarial attacks. *arXiv preprint arXiv:1712.03632*, 2017.

Lerrel Pinto, James Davidson, Rahul Sukthankar, and Abhinav Gupta. Robust adversarial reinforcement learning. In *International conference on machine learning*, pp. 2817–2826. PMLR, 2017.

Martin L Puterman. Markov decision processes. *Handbooks in operations research and management science*, 2:331–434, 1990.

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.

Mikayel Samvelyan, Tabish Rashid, Christian Schroeder De Witt, Gregory Farquhar, Nantas Nardelli, Tim GJ Rudner, Chia-Man Hung, Philip HS Torr, Jakob Foerster, and Shimon Whiteson. The starcraft multi-agent challenge. *arXiv preprint arXiv:1902.04043*, 2019.

Sazzad Sayyed, Milin Zhang, Shahriar Rifat, Ananthram Swami, Michael De Lucia, and Francesco Restuccia. Resilience and security of deep neural networks against intentional and unintentional perturbations: Survey and research challenges. *arXiv preprint arXiv:2408.00193*, 2024.

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

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Yanchao Sun, Ruijie Zheng, Yongyuan Liang, and Furong Huang. Who is the strongest enemy? towards optimal and efficient evasion attacks in deep rl. In *Deep RL Workshop NeurIPS 2021*.

Ming Tan. Multi-agent reinforcement learning: Independent vs. cooperative agents. In *Proceedings of the tenth international conference on machine learning*, pp. 330–337, 1993.

Yong Meng Teo, Ba Linh Luong, and Claudia Szabo. Formalization of emergence in multi-agent systems. In *Proceedings of the 1st ACM SIGSIM Conference on Principles of Advanced Discrete Simulation*, pp. 231–240, 2013.

Muning Wen, Jakub Kuba, Runji Lin, Weinan Zhang, Ying Wen, Jun Wang, and Yaodong Yang. Multi-agent reinforcement learning is a sequence modeling problem. *Advances in Neural Information Processing Systems*, 35:16509–16521, 2022.

Annie Wong, Thomas Bäck, Anna V Kononova, and Aske Plaat. Deep multiagent reinforcement learning: Challenges and directions. *Artificial Intelligence Review*, 56(6):5023–5056, 2023.

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.

Bing Yan, Peng Shi, Chee Peng Lim, Yuan Sun, and Ramesh K Agarwal. Security and safety-critical learning-based collaborative control for multiagent systems. *IEEE Transactions on Neural Networks and Learning Systems*, 2024.

Chao Yu, Akash Velu, Eugene Vinitsky, Jiaxuan Gao, Yu Wang, Alexandre Bayen, and Yi Wu. The surprising effectiveness of ppo in cooperative multi-agent games. *Advances in neural information processing systems*, 35:24611–24624, 2022.

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.

Huan Zhang, Hongge Chen, Chaowei Xiao, Bo Li, Mingyan Liu, Duane Boning, and Cho-Jui Hsieh. Robust deep reinforcement learning against adversarial perturbations on state observations. *Advances in neural information processing systems*, 33:21024–21037, 2020.

Huan Zhang, Hongge Chen, Duane S Boning, and Cho-Jui Hsieh. Robust reinforcement learning on state observations with learned optimal adversary. In *International Conference on Learning Representations*, 2021a.

Kaiqing Zhang, Zhuoran Yang, and Tamer Başar. Multi-agent reinforcement learning: A selective overview of theories and algorithms. *Handbook of reinforcement learning and control*, pp. 321–384, 2021b.

APPENDIX

A Notations and Descriptions **14**

B Algorithmic Pseudocode **15**

C Connection between ROA-MG and Equivalent MDP **16**

D Derivation of the Gradients in the Outer-Level Loss **16**

D.1 Gradient w.r.t. ω (State-Weighting) 17

D.2 Gradient w.r.t. ψ (Individual-Weighting) 17

E Theoretical Analysis and Proofs **17**

E.1 Preliminaries and Useful Lemmas 18

E.2 Convergence of Outer-Level Optimization with Dual Weighting Functions 19

E.2.1 Convergence of Outer-Level Optimization w.r.t. State-Weighting 19

E.2.2 Convergence of Outer-Level Optimization w.r.t. Individual-Weighting . . . 19

E.3 Convergence of Adversary Policy in the Inner-Level Optimization 20

F Experimental Details **22**

F.1 Implementation Details 22

F.2 Benchmark Descriptions 22

F.3 Intention Illustration 23

F.4 Additional Experimental Results 23

F.4.1 Additional Adversarial Attack Results 23

F.4.2 Additional Adversarial Training Results 26

F.4.3 Additional Case Study and Visualizations 28

F.5 Hyperparameter Settings 28

A NOTATIONS AND DESCRIPTIONS

Several key notations and descriptions used in this paper are summarized in Table 3 as follows:

Table 3: Key notations and descriptions.

Notation	Description
$\mathcal{N}$	The set of victim agents
$\mathcal{O}$	The joint observation space of the victim agents
$\mathcal{A}, \widehat{\mathcal{A}}$	The joint action space of the victim agents, the action space of the adversary
$\mathcal{S}$	The state space of the adversary
$R, \hat{R}$	The joint reward function and the joint reward function driven by attacker preference
$P, \widehat{P}$	The observation transition probability function, the transition probability function of the adversary
γ	The discount factor
$\rho_{\boldsymbol{\pi}}$	The marginal observation distribution
t, T	The current timestep and the maximum timesteps within each episode
$\boldsymbol{\pi}, \pi^i$	The joint policy function and the policy function of agent i
$\boldsymbol{\pi}_\nu, \boldsymbol{\pi}_{\nu \circ \alpha}$	The joint policy function of the victim agents and the joint policy function of the victim agents under targeted-behavior adversarial attacks
$\pi_\alpha, \boldsymbol{\pi}_\theta$	The adversary policy and the (joint) intention policy
$\boldsymbol{o}, o^i$	The joint observation of the victim agents and the local observation of victim agent i
$\tilde{\boldsymbol{o}}, \tilde{o}^i$	The perturbed joint observation of the victim agents and the perturbed local observation of victim agent i
$\boldsymbol{a}, a^i$	The joint action of the victim agents and the action of victim agent i
σ, y	The trajectory segment and the preference label
$\mathcal{D}_\sigma, \mathcal{D}$	The dataset of trajectory segments and preference labels; the replay buffer
$V(\boldsymbol{o})$	The multi-agent joint observation value function
$Q(\boldsymbol{o}, \boldsymbol{a})$	The multi-agent joint observation-action value function
$A(\boldsymbol{o}, \boldsymbol{a})$	The multi-agent joint advantage function
$\mathbb{E}(\cdot)$	The expectation function
$D_{\mathrm{KL}}(\cdot)$	The *Kullback-Leibler* (KL) divergence function
$\mathcal{B}(\cdot)$	The adversary perturbation constraint function
ϵ	The adversary perturbation budget
δ, Δ	The adversary perturbation and the adversary perturbation space
ν, α, θ	Neural network parameters for the victim policy, adversary policy, and intention policy
ζ, ξ	Neural network parameters for the reward model and the victim-observation encoder
ω, ψ	Neural network parameters for the state-weighting module and individual-weighting module
h_ω	The state-weighting function (parameterized by ω)
ϕ_ψ	The individual-weighting function (parameterized by ψ)
$\hat{r}_\zeta, r_{\mathrm{env}}$	The learned reward model (parameterized by ζ) and the original environment return
E_ξ	The victim-observation encoder (parameterized by ξ)
M_{atten}	The mutual-attention matrix of the victim-observation encoder
$u_i(\boldsymbol{o}), w_i(\boldsymbol{o})$	The attacker utility score and the normalized attacker utility score for victim agent i
$\mathcal{I}_k(\boldsymbol{o}_t)$	The target set of specific victim agents selected for attack at timestep t
$\mathcal{J}_\pi$	The outer-level loss function
$\mathcal{L}$	The inner-level loss function

B ALGORITHMIC PSEUDOCODE

An algorithmic pseudocode of MatBak is summarized in Algorithm 1.

Algorithm 1 Multi-Agent Targeted-Behavior Adversarial Attack (MatBak) Algorithm

Input: Fixed joint victim policy $\boldsymbol{\pi}_\nu$, scripted-preference feedback frequency K, outer-loss update frequency M, task horizon H

Initialize parameters of V_ξ, E_ξ, $\boldsymbol{\pi}_\theta$, $\hat{r}_\zeta$, π_α, ϕ_ψ and h_ω
Initialize replay buffer $\mathcal{D}$ and intention policy $\boldsymbol{\pi}_\theta$ via maximum entropy RL
Initialize preference dataset $\mathcal{D}_\sigma \leftarrow \varnothing$
for each iteration **do**
  ▷ Step 1: Construct the combined behavior policy.
  if episode is done **then**
    Sample switch point $h \sim \mathcal{U}(0, H)$
    Set $\boldsymbol{\pi}_{1:h}^{\text{combined}} = \boldsymbol{\pi}_{1:h}^{\nu \circ \alpha}$, $\boldsymbol{\pi}_{h+1:H}^{\text{combined}} = \boldsymbol{\pi}_{h+1:H}^{\theta}$
  end if
  Execute $\boldsymbol{a}_t \sim \boldsymbol{\pi}$; observe $s_{t+1} \leftarrow \boldsymbol{o}_{t+1}$
  Store transition: $\mathcal{D} \leftarrow \mathcal{D} \cup \{(s_t, a_t, \hat{r}_\zeta(s_t, a_t), s_{t+1})\}$
  ▷ Step 2: Query preference and train the reward model.
  if iteration number % $K = 0$ **then**
    Sample trajectory segment pair (σ^0, σ^1)
    Query preference label y from the scripted teacher (which simulates human preference)
    $\mathcal{D}_\sigma \leftarrow \mathcal{D}_\sigma \cup \{(\sigma^0, \sigma^1, y)\}$
    Sample mini-batch from $\mathcal{D}_\sigma$ and update $\hat{r}_\zeta$ via Eq. (2)
  end if
  ▷ Step 3: Inner-level optimization.
  for each gradient step **do**
    Sample mini-batch from $\mathcal{D}$
    Update π_α by minimizing Eq. (7) w.r.t. α
    Update E_ξ, V_ξ via Eq. (4)
  end for
  ▷ Step 4: Outer-level optimization.
  if iteration number % $M = 0$ **then**
    Sample mini-batch from $\mathcal{D}$
    Update h_ω, ϕ_ψ by minimizing Eq. (8) w.r.t. ω, ψ respectively
  end if
  ▷ Step 5: Update the intention policy.
  Update $\boldsymbol{\pi}_\theta$ via minimizing the policy loss of specific MARL methods (e.g., MAT, MAPPO, QMIX, etc.)
end for

Output: Trained adversary π_α

C Connection between ROA-MG and Equivalent MDP

Lemma C.1. ***ROA-MG to MDP Equivalence.*** *Given an ROA-MG $\langle \mathcal{N}, \boldsymbol{\mathcal{O}}, \boldsymbol{\mathcal{A}}, \mathcal{B}, \hat{R}, P, \gamma \rangle$ with a fixed joint victim policy $\boldsymbol{\pi}_\nu$, learning an optimal adversary α^* is equivalent to solving for an optimal policy in a standard MDP $\langle \mathcal{S}, \widehat{\mathcal{A}}, \widehat{R}, \widehat{P}, \gamma \rangle$, where $\mathcal{S} = \boldsymbol{\mathcal{O}}$, $\widehat{\mathcal{A}}(s) = \mathcal{B}(s)$,*

$$\widehat{P}(s' \mid s, \widehat{a}) = \sum_{\boldsymbol{a} \in \boldsymbol{\mathcal{A}}} \left(\prod_{i=1}^{n} \pi_\nu^i(a^i \mid \widehat{a}^i) \right) P(\boldsymbol{o}' \mid \boldsymbol{o}, \boldsymbol{a}), \tag{12}$$

where $(s, s') \equiv (\boldsymbol{o}, \boldsymbol{o}') \in \boldsymbol{\mathcal{O}}$, $\widehat{a} \in \widehat{\mathcal{A}}(s)$, $\widehat{a}^i$ is the i-th component of $\widehat{a}$, and

$$\widehat{R}(s, \widehat{a}) \triangleq \sum_{\boldsymbol{a} \in \boldsymbol{\mathcal{A}}} \left(\prod_{i=1}^{n} \pi_\nu^i(a^i \mid \widehat{a}^i) \right) \hat{R}(s, \boldsymbol{a}). \tag{13}$$

Proof. We construct an MDP whose state is the victims' joint observation and whose action is the (admissible) perturbed joint observation crafted by the adversary, and demonstrate the equivalence from the following perspectives:

Equivalence of the policy space. We define the adversary policy in an ROA-MG as

$$\mathcal{F}_\alpha \triangleq \{\alpha : \boldsymbol{\mathcal{O}} \to \boldsymbol{\mathcal{O}} \,|\, \alpha(\boldsymbol{o}) \in \mathcal{B}(\boldsymbol{o}),\ \forall \boldsymbol{o} \in \boldsymbol{\mathcal{O}}\}. \tag{14}$$

In the constructed MDP, an adversary policy is a mapping $\pi_\alpha : \mathcal{S} \to \widehat{\mathcal{A}}(\cdot)$ with $\pi_\alpha(s) \in \widehat{\mathcal{A}}(s)$. Since $\mathcal{S} = \boldsymbol{\mathcal{O}}$ and $\widehat{\mathcal{A}}(s) = \mathcal{B}(s)$, the two adversary policy classes coincide under the identification $\pi_\alpha(\boldsymbol{o}) \equiv \alpha(\boldsymbol{o})$.

Equivalence of the transition dynamics. Fix any $\alpha \in \mathcal{F}_\alpha$ and let $\widehat{a} = \alpha(\boldsymbol{o})$ be the perturbed joint observation fed to the victims. Given $\boldsymbol{o}_t = \boldsymbol{o}$ and $\widehat{a}_t = \widehat{a}$, the victims sample a joint action $\boldsymbol{a}_t \sim \boldsymbol{\pi}_\nu(\cdot \mid \widehat{a})$, i.e., $\Pr(\boldsymbol{a}_t = \boldsymbol{a} \mid \widehat{a}) = \prod_{i=1}^n \pi_\nu^i(a^i \mid \widehat{a}^i)$, and the environment transitions according to $P(\boldsymbol{o}' \mid \boldsymbol{o}, \boldsymbol{a})$. Therefore the induced one-step transition kernel over $\boldsymbol{o}$ under the adversary action $\widehat{a}$ is exactly

$$\Pr(\boldsymbol{o}_{t+1} = \boldsymbol{o}' \mid \boldsymbol{o}_t = \boldsymbol{o}, \widehat{a}_t = \widehat{a}) = \sum_{\boldsymbol{a} \in \boldsymbol{\mathcal{A}}} \left(\prod_{i=1}^{n} \pi_\nu^i(a^i \mid \widehat{a}^i) \right) P(\boldsymbol{o}' \mid \boldsymbol{o}, \boldsymbol{a}) = \widehat{P}(s' \mid s, \widehat{a}). \tag{15}$$

Equivalence of the optimization objective. For any $\alpha \in \mathcal{F}_\alpha$, define the MDP policy π_α by $\pi_\alpha(\boldsymbol{o}) = \alpha(\boldsymbol{o})$. At time t, conditioned on $(s_t, \widehat{a}_t) = (\boldsymbol{o}_t, \alpha(\boldsymbol{o}_t))$, the expected immediate return equals

$$\mathbb{E}\Big[\hat{R}(\boldsymbol{o}_t, \boldsymbol{a}_t) \mid \boldsymbol{o}_t, \widehat{a}_t\Big] = \sum_{\boldsymbol{a} \in \boldsymbol{\mathcal{A}}} \left(\prod_{i=1}^{n} \pi_\nu^i(a^i \mid \widehat{a}_t^i) \right) \hat{R}(\boldsymbol{o}_t, \boldsymbol{a}) = \widehat{R}(s_t, \widehat{a}_t). \tag{16}$$

Taking expectation over trajectories and summing over time step t yields

$$\mathcal{J}_{\text{ROA-MG}}(\alpha) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \hat{R}(\boldsymbol{o}_t, \boldsymbol{a}_t)\right] = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \widehat{R}(s_t, \widehat{a}_t)\right] = \mathcal{J}_{\text{MDP}}(\pi_\alpha). \tag{17}$$

Thus $\arg\max_{\alpha \in \mathcal{F}_\alpha} \mathcal{J}_{\text{ROA-MG}}(\alpha)$ is equivalent to $\arg\max_{\pi_\alpha} \mathcal{J}_{\text{MDP}}(\pi_\alpha)$, which completes the proof. □

D Derivation of the Gradients in the Outer-Level Loss

In this section, we derive the gradients w.r.t. ω and ψ in the outer-level loss. Note that the actual input of the individual-weighting module ϕ_ψ is the attention matrix M_{atten}, which is obtained by applying the victim-observation encoder operator $E_\xi(\cdot)$ (independent of ψ) to the state s (i.e., original joint victim observations). For simplicity, we directly denote it as $\phi_\psi(s)$ in Appendix.

D.1 Gradient w.r.t. ω (State-Weighting)

By the chain rule,

$$\nabla_\omega \mathcal{J}_\pi(\hat{\alpha}(\omega,\psi))\big|_{\omega_t} = \frac{\partial \mathcal{J}_\pi}{\partial \hat{\alpha}}\Big|_{\hat{\alpha}_t} \frac{\partial \hat{\alpha}(\omega,\psi)}{\partial \omega}\Big|_{\omega_t} \tag{18}$$

$$= \frac{\partial \mathcal{J}_\pi}{\partial \hat{\alpha}}\Big|_{\hat{\alpha}_t} \sum_{s\sim\mathcal{D}} \frac{\partial \hat{\alpha}(\omega,\psi)}{\partial h_\omega(s)}\Big|_{\alpha_t} \frac{\partial h_\omega(s)}{\partial \omega}\Big|_{\omega_t}. \tag{19}$$

Using the one-step inner-level approximation,

$$\hat{\alpha}(\omega,\psi) \approx \alpha_t - \eta_t \nabla_\alpha \mathcal{L}(\alpha_t;\omega,\theta,\psi), \tag{20}$$

and the inner-level loss

$$\mathcal{L}(\alpha;\omega,\theta,\psi) = \sum_{s\sim\mathcal{D}} h_\omega(s) \sum_{i=1}^{n} \phi_\psi^i(s)\, D_{\mathrm{KL}}\Big(\pi_{\nu\circ\alpha}^{(i)}(\cdot|s) \,\|\, \pi_\theta^{(i)}(\cdot|s)\Big), \tag{21}$$

we have, for each s,

$$\frac{\partial \hat{\alpha}(\omega,\psi)}{\partial h_\omega(s)}\Big|_{\alpha_t} = -\eta_t \sum_{i=1}^{n} \phi_\psi^i(s)\, \frac{\partial}{\partial \alpha} D_{\mathrm{KL}}\Big(\pi_{\nu\circ\alpha}^{(i)}(\cdot|s) \,\|\, \pi_\theta^{(i)}(\cdot|s)\Big)\Big|_{\alpha_t}. \tag{22}$$

Substituting yields

$$\nabla_\omega \mathcal{J}_\pi(\hat{\alpha}(\omega,\psi))\big|_{\omega_t}$$
$$= -\eta_t \sum_{s\sim\mathcal{D}} \left(\frac{\partial \mathcal{J}_\pi}{\partial \hat{\alpha}}\Big|_{\hat{\alpha}_t}\right)^\top \left[\sum_{i=1}^{n} \phi_\psi^i(s)\, \frac{\partial}{\partial \alpha} D_{\mathrm{KL}}\Big(\pi_{\nu\circ\alpha}^{(i)}(\cdot|s) \,\|\, \pi_\theta^{(i)}(\cdot|s)\Big)\Big|_{\alpha_t}\right] \frac{\partial h_\omega(s)}{\partial \omega}\Big|_{\omega_t}. \tag{23}$$

D.2 Gradient w.r.t. ψ (Individual-Weighting)

Similarly,

$$\nabla_\psi \mathcal{J}_\pi(\hat{\alpha}(\omega,\psi))\big|_{\psi_t} = \frac{\partial \mathcal{J}_\pi}{\partial \hat{\alpha}}\Big|_{\hat{\alpha}_t} \frac{\partial \hat{\alpha}(\omega,\psi)}{\partial \psi}\Big|_{\psi_t} \tag{24}$$

$$= \frac{\partial \mathcal{J}_\pi}{\partial \hat{\alpha}}\Big|_{\hat{\alpha}_t} \sum_{s\sim\mathcal{D}} \sum_{i=1}^{n} \frac{\partial \hat{\alpha}(\omega,\psi)}{\partial \phi_\psi^i(s)}\Big|_{\alpha_t} \frac{\partial \phi_\psi^i(s)}{\partial \psi}\Big|_{\psi_t}. \tag{25}$$

From the one-step inner update and the inner-level loss above, for each (s,i),

$$\frac{\partial \hat{\alpha}(\omega,\psi)}{\partial \phi_\psi^i(s)}\Big|_{\alpha_t} = -\eta_t\, h_\omega(s)\, \frac{\partial}{\partial \alpha} D_{\mathrm{KL}}\Big(\pi_{\nu\circ\alpha}^{(i)}(\cdot|s) \,\|\, \pi_\theta^{(i)}(\cdot|s)\Big)\Big|_{\alpha_t}. \tag{26}$$

Thus,

$$\nabla_\psi \mathcal{J}_\pi(\hat{\alpha}(\omega,\psi))\big|_{\psi_t}$$
$$= -\eta_t \sum_{s\sim\mathcal{D}} \sum_{i=1}^{n} \left(\frac{\partial \mathcal{J}_\pi}{\partial \hat{\alpha}}\Big|_{\hat{\alpha}_t}\right)^\top \left[h_\omega(s)\, \frac{\partial}{\partial \alpha} D_{\mathrm{KL}}\Big(\pi_{\nu\circ\alpha}^{(i)}(\cdot|s) \,\|\, \pi_\theta^{(i)}(\cdot|s)\Big)\Big|_{\alpha_t}\right] \frac{\partial \phi_\psi^i(s)}{\partial \psi}\Big|_{\psi_t}. \tag{27}$$

E Theoretical Analysis and Proofs

In this section, we provide convergence analysis and proofs for the bi-level optimization process in the proposed MatBak framework. We assume that: (i) $\mathcal{J}_\pi$ and $\mathcal{L}$ are Lipschitz-smooth (L-smooth), and their gradients are bounded, i.e., $\|\nabla \mathcal{J}_\pi\| \le \rho$, $\|\nabla \mathcal{L}\| \le \rho$; (ii) the modular learning step sizes η_t (adversary), β_t (state-weighting), and γ_t (individual-weighting) are positive.

E.1 Preliminaries and Useful Lemmas

Lemma E.1 (**Lipschitz Smoothness** (Lemma 1.2.3 in (Nesterov, 1998))). *If $f(x)$ is L-smooth on $\mathbb{R}^n$, then for any $x, y \in \mathbb{R}^n$:*

$$f(y) \leq f(x) + \nabla f(x)^\top (y - x) + \frac{L}{2}\|y - x\|^2. \tag{28}$$

Proof. By the fundamental theorem of calculus,

$$\begin{aligned}
f(y) &= f(x) + \int_0^1 \nabla f(x + \tau(y - x))^\top (y - x)\, d\tau && (29)\\
&= f(x) + \nabla f(x)^\top (y - x) + \int_0^1 \left(\nabla f(x + \tau(y - x)) - \nabla f(x)\right)^\top (y - x)\, d\tau && (30)\\
&\leq f(x) + \nabla f(x)^\top (y - x) + \int_0^1 \|\nabla f(x + \tau(y - x)) - \nabla f(x)\|\, \|y - x\|\, d\tau && (31)\\
&\leq f(x) + \nabla f(x)^\top (y - x) + \int_0^1 \tau L \|y - x\|^2\, d\tau && (32)\\
&= f(x) + \nabla f(x)^\top (y - x) + \frac{L}{2}\|y - x\|^2. && (33)
\end{aligned}$$

□

Lemma E.2 (**Sequence Convergence** (Lemma A.5 in (Mairal, 2013))). *Let $(a_n)_{n\geq 1}$ and $(b_n)_{n\geq 1}$ be two non-negative real sequences such that (1) the series $\sum_{n=1}^{\infty} a_n$ diverges, (2) the series $\sum_{n=1}^{\infty} a_n b_n$ converges, (3) there exists a constant $C > 0$ such that $|b_{n+1} - b_n| \leq C a_n$. Then, the sequence $(b_n)_{n\geq 1}$ converges to 0.*

Proof. Assume that $\limsup_{n\to+\infty} b_n = \lambda > 0$. Then we can build two sequences of indices $(n_j)_{j\geq 1}$ and $(m_j)_{j\geq 1}$ such that

$$m_j < n_j < m_{j+1}, \quad \frac{\lambda}{3} < b_k \quad \text{for} \quad m_j \leq k < n_j, \quad b_k \leq \frac{\lambda}{3} \quad \text{for} \quad n_j \leq k < m_{j+1}. \tag{34}$$

Let $\varepsilon = \frac{\lambda^2}{9C}$ and $\tilde{j}$ be large enough such that

$$\sum_{n=m_{\tilde{j}}}^{\infty} a_n b_n < \varepsilon. \tag{35}$$

Then, we have for all $j \geq \tilde{j}$ and all m with $m_j \leq m \leq n_j - 1$,

$$\begin{aligned}
\left|b_{n_j} - b_m\right| &\leq \sum_{k=m}^{n_j-1} |b_{k+1} - b_k| \leq \frac{3C}{\lambda} \sum_{k=m}^{n_j-1} a_k \frac{\lambda}{3} \leq \frac{3C}{\lambda} \sum_{k=m}^{n_j-1} a_k b_k \leq \frac{3C}{\lambda} \sum_{k=m}^{+\infty} a_k b_k && (36)\\
&\leq \frac{3C\varepsilon}{\lambda} \leq \frac{\lambda}{3}. && (37)
\end{aligned}$$

Therefore, using the triangle inequality,

$$b_m \leq b_{n_j} + \frac{\lambda}{3} \leq \frac{2\lambda}{3}, \tag{38}$$

and finally, for all m with $n_j \leq m < m_{j+1}$,

$$b_m \leq \frac{2\lambda}{3}, \tag{39}$$

which contradicts $\limsup_{n\to+\infty} b_n = \lambda > 0$. Therefore, by contradiction, we prove that b_n converges to zero. □

E.2 Convergence of Outer-Level Optimization with Dual Weighting Functions

E.2.1 Convergence of Outer-Level Optimization w.r.t. State-Weighting

Theorem E.3 (Convergence w.r.t. State-Weighting). *Fix ψ. Suppose $\mathcal{J}_\pi(\hat{\alpha}(\omega,\psi))$ is L-smooth w.r.t. ω and is lower-bounded by $\mathcal{J}_\pi^\star$. Let $\beta = \min\{\frac{1}{L}, \frac{c_2}{\sqrt{T}}\}$ for some $c_2 > 0$ satisfying $c_2 \le \frac{\sqrt{T}}{L}$, and update*

$$\omega_{t+1} = \omega_t - \beta\,\nabla_\omega \mathcal{J}_\pi\big(\hat{\alpha}(\omega_t,\psi)\big), \qquad t = 1,\dots,T. \tag{40}$$

Then,

$$\min_{1\le t\le T} \mathbb{E}\left[\left\|\nabla_\omega \mathcal{J}_\pi\big(\hat{\alpha}(\omega_t,\psi)\big)\right\|^2\right] \le \frac{2\left(\mathbb{E}\left[\mathcal{J}_\pi\big(\hat{\alpha}(\omega_1,\psi)\big)\right] - \mathcal{J}_\pi^\star\right)}{T\beta} = O\left(\frac{1}{\sqrt{T}}\right). \tag{41}$$

Proof. Apply Lemma E.1 to $f(\omega) = \mathcal{J}_\pi(\hat{\alpha}(\omega,\psi))$ with $y = \omega_{t+1}$ and $x = \omega_t$:

$$\mathcal{J}_\pi(\hat{\alpha}(\omega_{t+1},\psi)) - \mathcal{J}_\pi(\hat{\alpha}(\omega_t,\psi)) \le \nabla_\omega \mathcal{J}_\pi(\hat{\alpha}(\omega_t,\psi))^\top(\omega_{t+1}-\omega_t) + \frac{L}{2}\|\omega_{t+1}-\omega_t\|^2. \tag{42}$$

Using equation 40, $\omega_{t+1} - \omega_t = -\beta\nabla_\omega \mathcal{J}_\pi(\hat{\alpha}(\omega_t,\psi))$, hence

$$\mathcal{J}_\pi(\hat{\alpha}(\omega_{t+1},\psi)) - \mathcal{J}_\pi(\hat{\alpha}(\omega_t,\psi)) \le -\left(\beta - \frac{L\beta^2}{2}\right)\left\|\nabla_\omega \mathcal{J}_\pi(\hat{\alpha}(\omega_t,\psi))\right\|^2. \tag{43}$$

Since $\beta \le \frac{1}{L}$, we have $\beta - \frac{L\beta^2}{2} \ge \frac{\beta}{2}$, thus

$$\frac{\beta}{2}\left\|\nabla_\omega \mathcal{J}_\pi(\hat{\alpha}(\omega_t,\psi))\right\|^2 \le \mathcal{J}_\pi(\hat{\alpha}(\omega_t,\psi)) - \mathcal{J}_\pi(\hat{\alpha}(\omega_{t+1},\psi)). \tag{44}$$

Summing over $t = 1,\dots,T$ and taking expectation yields

$$\frac{\beta}{2}\sum_{t=1}^{T}\mathbb{E}\left[\left\|\nabla_\omega \mathcal{J}_\pi(\hat{\alpha}(\omega_t,\psi))\right\|^2\right] \le \mathbb{E}\left[\mathcal{J}_\pi(\hat{\alpha}(\omega_1,\psi))\right] - \mathbb{E}\left[\mathcal{J}_\pi(\hat{\alpha}(\omega_{T+1},\psi))\right] \tag{45}$$

$$\le \mathbb{E}\left[\mathcal{J}_\pi(\hat{\alpha}(\omega_1,\psi))\right] - \mathcal{J}_\pi^\star. \tag{46}$$

Therefore,

$$\min_{1\le t\le T}\mathbb{E}\left[\left\|\nabla_\omega \mathcal{J}_\pi(\hat{\alpha}(\omega_t,\psi))\right\|^2\right] \le \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\left\|\nabla_\omega \mathcal{J}_\pi(\hat{\alpha}(\omega_t,\psi))\right\|^2\right] \tag{47}$$

$$\le \frac{2\left(\mathbb{E}\left[\mathcal{J}_\pi(\hat{\alpha}(\omega_1,\psi))\right] - \mathcal{J}_\pi^\star\right)}{T\beta} = O\left(\frac{1}{\sqrt{T}}\right), \tag{48}$$

which finishes the proof. □

E.2.2 Convergence of Outer-Level Optimization w.r.t. Individual-Weighting

Theorem E.4 (Convergence w.r.t. Individual-Weighting). *Fix ω. Suppose $\mathcal{J}_\pi(\hat{\alpha}(\omega,\psi))$ is L-smooth w.r.t. ψ and is lower-bounded by $\mathcal{J}_\pi^\star$. Let $\gamma = \min\{\frac{1}{L}, \frac{c_3}{\sqrt{T}}\}$ for some $c_3 > 0$ satisfying $c_3 \le \frac{\sqrt{T}}{L}$, and update*

$$\psi_{t+1} = \psi_t - \gamma\,\nabla_\psi \mathcal{J}_\pi\big(\hat{\alpha}(\omega,\psi_t)\big), \qquad t = 1,\dots,T. \tag{49}$$

Then,

$$\min_{1\le t\le T}\mathbb{E}\left[\left\|\nabla_\psi \mathcal{J}_\pi\big(\hat{\alpha}(\omega,\psi_t)\big)\right\|^2\right] \le \frac{2\left(\mathbb{E}\left[\mathcal{J}_\pi\big(\hat{\alpha}(\omega,\psi_1)\big)\right] - \mathcal{J}_\pi^\star\right)}{T\gamma} = O\left(\frac{1}{\sqrt{T}}\right). \tag{50}$$

Proof. Apply Lemma E.1 to $g(\psi) = \mathcal{J}_\pi(\hat{\alpha}(\omega,\psi))$ with $y = \psi_{t+1}$ and $x = \psi_t$:

$$\mathcal{J}_\pi(\hat{\alpha}(\omega,\psi_{t+1})) - \mathcal{J}_\pi(\hat{\alpha}(\omega,\psi_t)) \le \nabla_\psi \mathcal{J}_\pi(\hat{\alpha}(\omega,\psi_t))^\top(\psi_{t+1}-\psi_t) + \frac{L}{2}\|\psi_{t+1}-\psi_t\|^2. \tag{51}$$

Using equation 49, $\psi_{t+1} - \psi_t = -\gamma \nabla_\psi \mathcal{J}_\pi(\hat{\alpha}(\omega, \psi_t))$, hence

$$\mathcal{J}_\pi(\hat{\alpha}(\omega, \psi_{t+1})) - \mathcal{J}_\pi(\hat{\alpha}(\omega, \psi_t)) \leq -\left(\gamma - \frac{L\gamma^2}{2}\right) \left\|\nabla_\psi \mathcal{J}_\pi(\hat{\alpha}(\omega, \psi_t))\right\|^2. \tag{52}$$

Since $\gamma \leq \frac{1}{L}$, we have $\gamma - \frac{L\gamma^2}{2} \geq \frac{\gamma}{2}$, thus

$$\frac{\gamma}{2} \left\|\nabla_\psi \mathcal{J}_\pi(\hat{\alpha}(\omega, \psi_t))\right\|^2 \leq \mathcal{J}_\pi(\hat{\alpha}(\omega, \psi_t)) - \mathcal{J}_\pi(\hat{\alpha}(\omega, \psi_{t+1})). \tag{53}$$

Summing over $t = 1, \ldots, T$ and taking expectation yields

$$\frac{\gamma}{2} \sum_{t=1}^{T} \mathbb{E}\left[\left\|\nabla_\psi \mathcal{J}_\pi(\hat{\alpha}(\omega, \psi_t))\right\|^2\right] \leq \mathbb{E}\left[\mathcal{J}_\pi(\hat{\alpha}(\omega, \psi_1))\right] - \mathbb{E}\left[\mathcal{J}_\pi(\hat{\alpha}(\omega, \psi_{T+1}))\right] \tag{54}$$

$$\leq \mathbb{E}\left[\mathcal{J}_\pi(\hat{\alpha}(\omega, \psi_1))\right] - \mathcal{J}_\pi^\star. \tag{55}$$

Therefore,

$$\min_{1 \leq t \leq T} \mathbb{E}\left[\left\|\nabla_\psi \mathcal{J}_\pi(\hat{\alpha}(\omega, \psi_t))\right\|^2\right] \leq \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}\left[\left\|\nabla_\psi \mathcal{J}_\pi(\hat{\alpha}(\omega, \psi_t))\right\|^2\right] \tag{56}$$

$$\leq \frac{2\left(\mathbb{E}\left[\mathcal{J}_\pi(\hat{\alpha}(\omega, \psi_1))\right] - \mathcal{J}_\pi^\star\right)}{T\gamma} = O\left(\frac{1}{\sqrt{T}}\right), \tag{57}$$

which finishes the proof. □

E.3 Convergence of Adversary Policy in the Inner-Level Optimization

Theorem E.5 (Convergence of the Adversary Policy). *Suppose for any (ω, ψ) the inner loss $\mathcal{L}(\alpha; \omega, \theta, \psi)$ is lower-bounded and L-smooth w.r.t. α, ω, and ψ. The gradients are bounded by:*

$$\left\|\nabla_\alpha \mathcal{L}(\alpha; \omega, \theta, \psi)\right\| \leq \rho, \qquad \left\|\nabla_\omega \mathcal{L}(\alpha; \omega, \theta, \psi)\right\| \leq \rho, \qquad \left\|\nabla_\psi \mathcal{L}(\alpha; \omega, \theta, \psi)\right\| \leq \rho, \tag{58}$$

and $\|\nabla_\omega \mathcal{J}_\pi\| \leq \rho$, $\|\nabla_\psi \mathcal{J}_\pi\| \leq \rho$. Assume that the inner-gradient is Lipschitz continuous and there exist constants $L_{\alpha\alpha}, L_{\alpha\omega}, L_{\alpha\psi} \geq 0$ such that for all t we have

$$\left\|\nabla_\alpha \mathcal{L}(\alpha_{t+1}; \omega_{t+1}, \theta, \psi_{t+1}) - \nabla_\alpha \mathcal{L}(\alpha_t; \omega_t, \theta, \psi_t)\right\| \leq L_{\alpha\alpha}\|\alpha_{t+1} - \alpha_t\| + L_{\alpha\omega}\|\omega_{t+1} - \omega_t\| + L_{\alpha\psi}\|\psi_{t+1} - \psi_t\|. \tag{59}$$

Assume the step sizes satisfy

$$\sum_{t=1}^{\infty} \eta_t = \infty, \qquad \sum_{t=1}^{\infty} \eta_t^2 < \infty, \qquad \eta_t \leq \frac{1}{L}, \tag{60}$$

and there exist constants $c_\beta, c_\gamma \geq 0$ such that

$$\beta_t \leq c_\beta \eta_t^2, \qquad \gamma_t \leq c_\gamma \eta_t^2. \tag{61}$$

Then, we have

$$\lim_{t \to \infty} \mathbb{E}\left[\left\|\nabla_\alpha \mathcal{L}(\alpha_t; \omega_t, \theta, \psi_t)\right\|^2\right] = 0. \tag{62}$$

Proof. Recall the inner-level update

$$\alpha_{t+1} = \alpha_t - \eta_t \nabla_\alpha \mathcal{L}(\alpha_t; \omega_t, \theta, \psi_t), \tag{63}$$

where

$$\mathcal{L}(\alpha; \omega_t, \theta, \psi_t) = \mathbb{E}_{s \sim \mathcal{D}}\left[h_{\omega_t}(s) \sum_{i=1}^{n} \phi_{\psi_t}^i(s) \cdot D_{\mathrm{KL}}^{(i)}(\alpha; s)\right]. \tag{64}$$

By Lemma E.1 and the Lipschitz-smoothness of $\mathcal{L}(\cdot; \omega_t, \theta, \psi_t)$ in α,

$$\mathcal{L}(\alpha_{t+1}; \omega_t, \theta, \psi_t) - \mathcal{L}(\alpha_t; \omega_t, \theta, \psi_t) \leq \nabla_\alpha \mathcal{L}(\alpha_t; \omega_t, \theta, \psi_t)^\top (\alpha_{t+1} - \alpha_t) + \frac{L}{2}\|\alpha_{t+1} - \alpha_t\|^2. \tag{65}$$

Using equation 63, $\alpha_{t+1} - \alpha_t = -\eta_t \nabla_\alpha \mathcal{L}(\alpha_t; \omega_t, \theta, \psi_t)$, we obtain

$$\mathcal{L}(\alpha_{t+1}; \omega_t, \theta, \psi_t) - \mathcal{L}(\alpha_t; \omega_t, \theta, \psi_t) \leq -\eta_t \|\nabla_\alpha \mathcal{L}(\alpha_t; \omega_t, \theta, \psi_t)\|^2 + \frac{L\eta_t^2}{2} \|\nabla_\alpha \mathcal{L}(\alpha_t; \omega_t, \theta, \psi_t)\|^2. \tag{66}$$

Since $\eta_t \leq \frac{1}{L}$, we have $\eta_t - \frac{L\eta_t^2}{2} \geq \frac{\eta_t}{2}$, hence

$$\frac{\eta_t}{2} \|\nabla_\alpha \mathcal{L}(\alpha_t; \omega_t, \theta, \psi_t)\|^2 \leq \mathcal{L}(\alpha_t; \omega_t, \theta, \psi_t) - \mathcal{L}(\alpha_{t+1}; \omega_t, \theta, \psi_t). \tag{67}$$

Next, we bound the drift caused by updating (ω, ψ):

$$\begin{aligned}
&\mathcal{L}(\alpha_{t+1}; \omega_t, \theta, \psi_t) - \mathcal{L}(\alpha_{t+1}; \omega_{t+1}, \theta, \psi_{t+1}) \\
=&\big(\mathcal{L}(\alpha_{t+1}; \omega_t, \theta, \psi_t) - \mathcal{L}(\alpha_{t+1}; \omega_{t+1}, \theta, \psi_t)\big) \\
&\quad + \big(\mathcal{L}(\alpha_{t+1}; \omega_{t+1}, \theta, \psi_t) - \mathcal{L}(\alpha_{t+1}; \omega_{t+1}, \theta, \psi_{t+1})\big) && (68) \\
\leq& \sup_{\omega_t \leq \omega \leq \omega_{t+1}} \|\nabla_\omega \mathcal{L}(\alpha_{t+1}; \omega, \theta, \psi_t)\| \cdot \|\omega_{t+1} - \omega_t\| \\
&\quad + \sup_{\psi_t \leq \psi \leq \psi_{t+1}} \|\nabla_\psi \mathcal{L}(\alpha_{t+1}; \omega_{t+1}, \theta, \psi)\| \cdot \|\psi_{t+1} - \psi_t\| && (69) \\
\leq& \rho(\|\omega_{t+1} - \omega_t\| + \|\psi_{t+1} - \psi_t\|). && (70)
\end{aligned}$$

By the bounded gradients and the outer updates,

$$\|\omega_{t+1} - \omega_t\| = \beta_t \|\nabla_\omega \mathcal{J}_\pi(\hat{\alpha}(\omega_t, \psi_t))\| \leq \rho\beta_t, \qquad \|\psi_{t+1} - \psi_t\| = \gamma_t \|\nabla_\psi \mathcal{J}_\pi(\hat{\alpha}(\omega_t, \psi_t))\| \leq \rho\gamma_t, \tag{71}$$

thus

$$\mathcal{L}(\alpha_{t+1}; \omega_t, \theta, \psi_t) - \mathcal{L}(\alpha_{t+1}; \omega_{t+1}, \theta, \psi_{t+1}) \leq \rho^2(\beta_t + \gamma_t). \tag{72}$$

Combining equation 67 and equation 72, we obtain

$$\frac{\eta_t}{2} \|\nabla_\alpha \mathcal{L}(\alpha_t; \omega_t, \theta, \psi_t)\|^2 \leq \mathcal{L}(\alpha_t; \omega_t, \theta, \psi_t) - \mathcal{L}(\alpha_{t+1}; \omega_{t+1}, \theta, \psi_{t+1}) + \rho^2(\beta_t + \gamma_t). \tag{73}$$

Summing equation 73 from $t = 1$ to T and taking expectation yields

$$\begin{aligned}
\sum_{t=1}^{T} \eta_t \, \mathbb{E}\big[\|\nabla_\alpha \mathcal{L}(\alpha_t; \omega_t, \theta, \psi_t)\|^2\big] \leq\, & 2\,\mathbb{E}[\mathcal{L}(\alpha_1; \omega_1, \theta, \psi_1)] - 2\,\mathbb{E}[\mathcal{L}(\alpha_{T+1}; \omega_{T+1}, \theta, \psi_{T+1})] \\
&+ 2\rho^2 \sum_{t=1}^{T} (\beta_t + \gamma_t) && (74) \\
\leq\, & 2\Big(\mathbb{E}[\mathcal{L}(\alpha_1; \omega_1, \theta, \psi_1)] - \inf_{\alpha, \omega, \psi} \mathcal{L}(\alpha; \omega, \theta, \psi)\Big) \\
&+ 2\rho^2 (c_\beta + c_\gamma) \sum_{t=1}^{T} \eta_t^2 && (75) \\
\leq\, & 2\Big(\mathbb{E}[\mathcal{L}(\alpha_1; \omega_1, \theta, \psi_1)] - \inf \mathcal{L}\Big) + 2\rho^2 (c_\beta + c_\gamma) \sum_{t=1}^{\infty} \eta_t^2 < \infty. \\
& && (76)
\end{aligned}$$

Therefore,

$$\sum_{t=1}^{\infty} \eta_t \, \mathbb{E}\big[\|\nabla_\alpha \mathcal{L}(\alpha_t; \omega_t, \theta, \psi_t)\|^2\big] < \infty. \tag{77}$$

Let $b_t = \mathbb{E}\big[\|\nabla_\alpha \mathcal{L}(\alpha_t; \omega_t, \theta, \psi_t)\|^2\big]$, $a_t = \eta_t$, $g_t := \nabla_\alpha \mathcal{L}(\alpha_t; \omega_t, \theta, \psi_t)$. Then

$$\begin{aligned}
|b_{t+1} - b_t| &= \Big|\mathbb{E}\big[\|g_{t+1}\|^2 - \|g_t\|^2\big]\Big| \leq \mathbb{E}\Big[\big|\|g_{t+1}\|^2 - \|g_t\|^2\big|\Big] && (78) \\
&= \mathbb{E}\Big[\big|(g_{t+1} - g_t)^\top (g_{t+1} + g_t)\big|\Big] \leq \mathbb{E}\Big[\|g_{t+1} - g_t\|(\|g_{t+1}\| + \|g_t\|)\Big]. && (79)
\end{aligned}$$

By the bounded-gradient assumption, $\|g_t\| \le \rho$ for all t, hence $\|g_{t+1}\| + \|g_t\| \le 2\rho$ and

$$|b_{t+1} - b_t| \le 2\rho\,\mathbb{E}\big[\|g_{t+1} - g_t\|\big]. \tag{80}$$

Since

$$\begin{aligned}\big\|\nabla_\alpha \mathcal{L}(\alpha_{t+1};\omega_{t+1},\theta,\psi_{t+1}) - \nabla_\alpha \mathcal{L}(\alpha_t;\omega_t,\theta,\psi_t)\big\| \le\ & L_{\alpha\alpha}\|\alpha_{t+1}-\alpha_t\| \\ & + L_{\alpha\omega}\|\omega_{t+1}-\omega_t\| + L_{\alpha\psi}\|\psi_{t+1}-\psi_t\|,\end{aligned} \tag{81}$$

and the update magnitude bounds

$$\|\alpha_{t+1} - \alpha_t\| \le \eta_t\rho, \quad \|\omega_{t+1} - \omega_t\| \le \rho\beta_t, \quad \|\psi_{t+1} - \psi_t\| \le \rho\gamma_t, \tag{82}$$

we obtain

$$\|g_{t+1} - g_t\| \le \rho\big(L_{\alpha\alpha}\eta_t + L_{\alpha\omega}\beta_t + L_{\alpha\psi}\gamma_t\big). \tag{83}$$

Combining equation 80 and equation 83 yields

$$|b_{t+1} - b_t| \le 2\rho^2\big(L_{\alpha\alpha}\eta_t + L_{\alpha\omega}\beta_t + L_{\alpha\psi}\gamma_t\big) \tag{84}$$

$$\le 2\rho^2\big(L_{\alpha\alpha}\eta_t + L_{\alpha\omega}c_\beta\eta_t^2 + L_{\alpha\psi}c_\gamma\eta_t^2\big) \tag{85}$$

$$\le 2\rho^2\big(L_{\alpha\alpha} + L_{\alpha\omega}c_\beta + L_{\alpha\psi}c_\gamma\big)\eta_t. \tag{86}$$

Therefore, there exists a constant $C := 2\rho^2(L_{\alpha\alpha} + c_\beta L_{\alpha\omega} + c_\gamma L_{\alpha\psi}) > 0$ such that

$$|b_{t+1} - b_t| \le C\eta_t = Ca_t. \tag{87}$$

By equation 60, equation 77, and Lemma E.2, we have

$$\lim_{t\to\infty} \mathbb{E}\big[\|\nabla_\alpha \mathcal{L}(\alpha_t;\omega_t,\theta,\psi_t)\|^2\big] = 0, \tag{88}$$

which finishes the proof. □

F Experimental Details

This section presents experimental details for Section 5.

F.1 Implementation Details

We conduct the experiments using NVIDIA 5090 GPUs (for SMAC) and NVIDIA A6000 GPUs (for MetaDrive) in this study. The programming environment is established with Python 3.10.19 and PyTorch 2.9.0. To ensure robustness, each experiment is repeated five times with distinct random seeds, and the experimental results are reported in the format of mean metric (e.g., attacker return, environment reward) values $\pm$ standard deviation for both benchmarks.

F.2 Benchmark Descriptions

StarCraft Multi-Agent Challenge (SMAC). StarCraft Multi-Agent Challenge (SMAC) (Samvelyan et al., 2019) provides a specialized evaluation testbed for decentralized micromanagement tasks, using the StarCraft II game engine integrated with Blizzard's API and DeepMind's PySC2 environment. In contrast to full-game simulations, SMAC specifically models constrained tactical engagements where a team of allied units, operating under partial observability and decentralized control, must defeat an opposing force controlled by scripted AI. This design creates a suite of demanding combat micro-scenarios that facilitates reproducible benchmarking of collaborative MARL algorithms under conditions of limited information.

MetaDrive. MetaDrive (Li et al., 2022) is a compositional driving simulation platform designed for generalizable reinforcement learning research. Built on the Panda3D graphics engine and the Bullet physics engine, it prioritizes scenario compositionality and procedural generation over photorealistic rendering, unlike conventional driving simulators. By composing elementary road blocks and integrating real-world traffic datasets such as Waymo and Argoverse, MetaDrive synthesizes an infinite variety of diverse driving scenes. This design supports a range of interactive tasks, including single-agent navigation, multi-agent coordination, and safety-critical exploration. It thereby enables reproducible benchmarking of reinforcement learning algorithms under conditions of environmental heterogeneity and distributional shifts.

F.3 Intention Illustration

We present an illustration of the attacker's intention on SMAC and MetaDrive benchmarks in Figure 7. On SMAC, we characterize the attacker's intention as inducing a targeted dispersed-formation behavior, i.e., guiding agents to move away from one another, thereby hindering coordination and making them more vulnerable to being defeated individually in combat. On MetaDrive, we characterize the attacker's intention as inducing a team-level targeted behavior of traffic standstill, i.e., reducing vehicle speed and travel distance through collisions or lane encroachment.



Figure 7: An illustration of the attacker's intention on SMAC (left) and MetaDrive (right) benchmarks. The attacked agents are marked with boxes.

F.4 Additional Experimental Results

This section presents additional experimental results to further validate the effectiveness of our method in both inducing targeted-behavior via observation-adversarial attacks and enhancing MARL robustness against such attacks.

F.4.1 Additional Adversarial Attack Results

Additional adversarial attack results on SMAC. Full comparison results of all methods on the 8m, MMM, 3s5z, and MMM2 maps of SMAC are summarized in Table 4 (reported as average attacker return $\pm$ standard deviation). Victim agents are trained by different MARL algorithms including MAT, MAPPO, and QMIX. The experimental results show that MatBak consistently outperforms all the baseline methods, which validates its effectiveness.

We also report attack success rates (ASRs) of different methods in inducing targeted behavior on SMAC. Specifically, an attack is considered successful if it increases the average spatial dispersion of victim agents beyond the no-attack baseline level (i.e., compare the mean spatial dispersion value) throughout an episode. The experimental results are summarized in Table 5 (in oracle settings) and Table 6 (in PbRL settings). It can be observed that MatBak achieves the highest ASR in 10 out of 12 oracle settings (including ties) and 10 out of 12 PbRL settings. Notably, MatBak demonstrates particularly strong performance against QMIX-trained victims, achieving 85.00–100.00% ASR in PbRL settings compared to 46.67–93.33% for the strongest non-MatBak baseline RAT. This advantage is especially pronounced on the 8m map (90.00% vs. 46.67%) and 3s5z map (91.67% vs. 55.00%), indicating MatBak's effectiveness in manipulating value-based MARL systems toward targeted dispersed-formation behaviors on SMAC.

Additional adversarial attack results on MetaDrive. The numerical comparison results of all methods on the MetaDrive benchmark are summarized in Table 7. The experimental results show that MatBak outperforms all baseline methods in attacker return, which validates its effectiveness in inducing targeted behaviors in victim MAS.

We also report attack success rates (ASRs) of different methods in inducing targeted behavior on MetaDrive. Specifically, an attack is considered successful if it effectively increases the number of additionally stopped vehicles compared to the no-attack baseline (i.e., compare the number of

Table 4: Experimental results of adversarial attacks on SMAC tasks for 8m, MMM, 3s5z, and MMM2 maps (displayed in average attacker return ± standard deviation). The column headers represent different maps, as well as the MARL algorithms used to train the victim models, including MAT (Wen et al., 2022), MAPPO (Yu et al., 2022), and QMIX (Rashid et al., 2020).

	Method	**8m** (homogeneous)			**MMM** (heterogeneous)		
		MAT	MAPPO	QMIX	MAT	MAPPO	QMIX
Oracle	Adv-ZS (Gleave et al., 2020)	47.66 ±0.76	59.69 ±1.71	58.69 ±7.05	131.78 ±9.23	128.25 ±4.24	144.42 ±24.45
	Adv-TP (Guo et al., 2021)	48.32 ±1.12	58.99 ±1.70	57.52 ±2.66	127.38 ±7.70	128.37 ±2.19	133.52 ±15.14
	Adv-xAI (Wu et al., 2021)	48.26 ±1.03	57.76 ±0.57	59.80 ±5.40	131.87 ±6.73	128.22 ±3.76	143.35 ±17.42
	AMI (Li et al., 2025)	48.18 ±1.18	59.61 ±4.54	56.09 ±4.00	125.31 ±1.79	129.65 ±2.92	140.12 ±17.09
	SA-RL (Zhang et al., 2021a)	49.00 ±0.62	254.22 ±2.97	74.41 ±3.15	130.76 ±3.26	264.23 ±2.96	202.06 ±12.33
	PA-AD (Sun et al.)	49.32 ±0.21	248.27 ±4.15	56.50 ±1.19	139.34 ±3.36	265.90 ±1.79	211.22 ±14.85
PbRL	Adv-ZS (Gleave et al., 2020)	49.21 ±1.73	62.82 ±1.50	58.49 ±3.82	124.13 ±4.67	126.65 ±6.17	124.18 ±3.09
	Adv-TP (Guo et al., 2021)	48.88 ±2.09	60.92 ±2.59	58.42 ±3.42	124.13 ±4.67	125.78 ±3.74	126.10 ±4.74
	Adv-xAI (Wu et al., 2021)	49.18 ±1.54	60.62 ±4.31	58.40 ±3.31	125.80 ±4.58	127.35 ±4.49	127.61 ±2.06
	AMI (Li et al., 2025)	48.76 ±1.18	61.59 ±2.13	58.51 ±3.73	123.89 ±4.91	126.56 ±5.17	127.61 ±4.27
	SA-RL (Zhang et al., 2021a)	48.18 ±0.53	252.53 ±2.49	72.94 ±3.07	128.54 ±2.04	264.47 ±1.33	194.47 ±15.86
	PA-AD (Sun et al.)	49.09 ±0.33	252.61 ±3.23	56.90 ±0.94	139.84 ±2.84	266.69 ±1.83	215.67 ±14.63
	RAT (Bai et al., 2025)	284.68 ±30.68	359.52 ±24.69	114.07 ±42.20	488.35 ±60.08	438.40 ±39.25	489.10 ±101.60
	MatBak (Ours)	**379.60 ±74.63**	**553.38 ±135.80**	**585.99 ±229.58**	**517.20 ±43.73**	**565.38 ±106.38**	**623.80 ±263.78**
	Method	**3s5z** (heterogeneous, Hard)			**MMM2** (heterogeneous, Hard)		
		MAT	MAPPO	QMIX	MAT	MAPPO	QMIX
Oracle	Adv-ZS (Gleave et al., 2020)	200.61 ±2.86	190.48 ±4.69	249.57 ±7.26	144.46 ±2.86	131.44 ±5.97	178.65 ±6.23
	Adv-TP (Guo et al., 2021)	207.28 ±5.14	192.77 ±5.35	243.80 ±11.23	144.11 ±3.33	134.23 ±2.96	170.79 ±14.90
	Adv-xAI (Wu et al., 2021)	202.01 ±3.85	191.29 ±5.45	241.68 ±16.35	151.59 ±23.21	129.38 ±9.85	179.31 ±17.52
	AMI (Li et al., 2025)	203.30 ±5.67	190.12 ±2.96	246.86 ±11.99	141.08 ±2.52	130.72 ±2.68	182.56 ±6.58
	SA-RL (Zhang et al., 2021a)	228.10 ±8.91	306.64 ±1.02	224.69 ±1.85	161.19 ±4.87	251.75 ±2.79	197.30 ±8.26
	PA-AD (Sun et al.)	234.37 ±3.28	307.36 ±6.12	248.14 ±8.32	140.23 ±2.42	254.69 ±4.93	227.51 ±22.44
PbRL	Adv-ZS (Gleave et al., 2020)	200.32 ±1.10	189.47 ±3.49	248.28 ±9.36	141.38 ±3.63	133.06 ±1.06	176.16 ±12.12
	Adv-TP (Guo et al., 2021)	201.30 ±5.70	187.93 ±3.24	248.28 ±9.36	141.57 ±3.83	133.10 ±0.95	176.16 ±12.12
	Adv-xAI (Wu et al., 2021)	199.73 ±2.69	189.81 ±3.24	250.41 ±2.96	145.10 ±6.31	134.42 ±1.35	182.22 ±10.06
	AMI (Li et al., 2025)	203.35 ±3.16	186.18 ±3.08	253.92 ±7.24	145.87 ±4.67	134.82 ±2.55	184.56 ±13.30
	SA-RL (Zhang et al., 2021a)	219.96 ±4.56	306.01 ±2.31	218.60 ±0.75	151.75 ±2.75	255.86 ±2.73	193.09 ±5.51
	PA-AD (Sun et al.)	236.41 ±2.13	310.34 ±4.59	246.19 ±7.77	137.71 ±2.50	262.33 ±6.56	223.69 ±14.06
	RAT (Bai et al., 2025)	718.30 ±28.99	441.64 ±13.28	541.72 ±131.33	465.10 ±57.58	463.47 ±29.14	343.96 ±70.67
	MatBak (Ours)	**742.93 ±73.58**	**630.85 ±65.62**	**731.40 ±188.58**	**570.43 ±64.45**	**530.75 ±93.86**	**518.14 ±220.69**

Table 5: Attack success rates of different methods on SMAC. Attacker agents are trained by the ground-truth targeted-behavior alignment reward in oracle settings. Each result is computed across 60 evaluation episodes.

Victim	Map	Adv-ZS	Adv-TP	Adv-xAI	AMI	SA-RL	PA-AD	RAT	MatBak (Ours)
MAT	8m	56.67%	50.00%	43.33%	50.00%	48.33%	61.67%	83.33%	**96.67%**
	3s5z	51.67%	56.67%	56.67%	55.00%	68.33%	68.33%	**100.00%**	91.67%
	MMM	56.67%	43.33%	58.33%	48.33%	48.33%	65.00%	**100.00%**	91.67%
	MMM2	41.67%	36.67%	46.67%	36.67%	68.33%	86.67%	81.67%	**100.00%**
MAPPO	8m	45.00%	25.00%	33.33%	41.67%	80.00%	76.67%	**96.67%**	**96.67%**
	3s5z	43.33%	26.67%	38.33%	43.33%	48.33%	78.33%	80.00%	**96.67%**
	MMM	60.00%	66.67%	65.00%	66.67%	53.33%	70.00%	80.00%	**100.00%**
	MMM2	48.33%	48.33%	51.67%	46.67%	61.67%	85.00%	**98.33%**	**98.33%**
QMIX	8m	31.67%	31.67%	31.67%	31.67%	36.67%	33.33%	50.00%	**70.00%**
	3s5z	51.67%	35.00%	53.33%	38.33%	58.33%	36.67%	51.67%	**80.00%**
	MMM	48.33%	25.00%	55.00%	46.67%	55.00%	76.67%	86.67%	**95.00%**
	MMM2	66.67%	56.67%	73.33%	65.00%	53.33%	80.00%	86.67%	**100.00%**

Table 6: Attack success rates of different methods on SMAC. Attacker agents are trained using the feedback signal from the reward model in PbRL settings. Each result is computed across 60 evaluation episodes.

Victim	Map	Adv-ZS	Adv-TP	Adv-xAI	AMI	SA-RL	PA-AD	RAT	MatBak (Ours)
MAT	8m	50.00%	55.00%	51.67%	51.67%	53.33%	51.67%	80.00%	**95.00%**
	3s5z	40.00%	55.00%	46.67%	53.33%	56.67%	50.00%	**98.33%**	96.67%
	MMM	38.33%	30.00%	46.67%	31.67%	45.00%	68.33%	95.00%	**98.33%**
	MMM2	48.33%	51.67%	46.67%	60.00%	48.33%	90.00%	90.00%	**100.00%**
MAPPO	8m	68.33%	61.67%	66.67%	63.33%	73.33%	71.67%	78.33%	**91.67%**
	3s5z	46.67%	43.33%	40.00%	40.00%	50.00%	63.33%	60.00%	**98.33%**
	MMM	53.33%	50.00%	63.33%	46.67%	60.00%	76.67%	91.67%	**100.00%**
	MMM2	43.33%	41.67%	45.00%	58.33%	46.67%	90.00%	**100.00%**	98.33%
QMIX	8m	33.33%	38.33%	36.67%	25.00%	45.00%	35.00%	46.67%	**90.00%**
	3s5z	53.33%	55.00%	55.00%	55.00%	55.00%	36.67%	55.00%	**91.67%**
	MMM	60.00%	38.33%	51.67%	40.00%	45.00%	56.67%	93.33%	**100.00%**
	MMM2	56.67%	55.00%	50.00%	58.33%	61.67%	71.67%	81.67%	**85.00%**

Table 7: Numerical comparison results in intersection, bottleneck, roundabout, SCSC, and SCSS scenarios of MetaDrive, reported as mean attacker return $\pm$ standard deviation. Victim models are trained with MAT, MAPPO, and QMIX.

	Method	intersection	bottleneck	roundabout	SCSC	SCSS
MAT	Adv-ZS (Gleave et al., 2020)	-87.95 ±34.37	-85.77 ±42.89	-196.03 ±25.51	-198.92 ±29.18	-251.59 ±46.38
	Adv-TP (Guo et al., 2021)	-73.20 ±43.00	-119.48 ±52.78	-173.42 ±39.33	-128.79 ±63.33	-217.90 ±84.66
	Adv-xAI (Wu et al., 2021)	-116.51 ±9.29	-152.91 ±18.82	-195.22 ±29.63	-196.72 ±85.37	-287.34 ±20.00
	AMI (Li et al., 2025)	-48.76 ±12.58	-61.78 ±34.75	-151.65 ±64.02	-174.05 ±64.92	-263.41 ±36.74
	SA-RL (Zhang et al., 2021a)	-115.74 ±2.97	-160.41 ±1.80	-194.96 ±1.13	-125.26 ±16.86	-225.52 ±6.17
	PA-AD (Sun et al.)	-86.22 ±4.03	-125.00 ±2.24	-157.08 ±1.37	-113.28 ±6.47	-233.23 ±0.91
	RAT (Bai et al., 2025)	-22.73 ±3.60	**-14.20 ±2.43**	-24.56 ±4.87	-28.91 ±13.82	-102.36 ±18.24
	MatBak (Ours)	**-14.38 ±11.20**	-20.29 ±20.29	**-15.39 ±7.65**	**-13.80 ±25.28**	**-13.92 ±11.18**
MAPPO	Adv-ZS (Gleave et al., 2020)	-85.56 ±39.72	-139.26 ±24.81	-201.69 ±37.07	-171.02 ±71.06	-279.81 ±12.96
	Adv-TP (Guo et al., 2021)	-61.31 ±33.23	-81.92 ±15.70	-160.85 ±64.58	-177.53 ±66.48	-172.25 ±101.90
	Adv-xAI (Wu et al., 2021)	-96.91 ±18.38	-151.69 ±16.66	-232.93 ±21.43	-198.71 ±52.71	-285.05 ±8.54
	AMI (Li et al., 2025)	-75.65 ±30.56	-127.36 ±28.01	-194.15 ±30.26	-162.67 ±63.83	-283.12 ±17.31
	SA-RL (Zhang et al., 2021a)	-113.74 ±1.72	-143.29 ±5.11	-214.82 ±3.39	-120.25 ±11.21	-251.05 ±4.89
	PA-AD (Sun et al.)	-96.25 ±2.87	-121.71 ±2.63	-193.50 ±4.37	-120.06 ±11.46	-253.02 ±0.94
	RAT (Bai et al., 2025)	-27.89 ±3.51	-35.88 ±6.76	-37.66 ±11.88	-48.63 ±7.16	-87.23 ±16.56
	MatBak (Ours)	**-11.70 ±2.77**	**-32.23 ±19.43**	**-23.10 ±5.00**	**-40.79 ±22.14**	**-16.34 ±4.52**
QMIX	Adv-ZS (Gleave et al., 2020)	-45.42 ±11.29	-75.16 ±1.16	-62.37 ±5.92	-132.85 ±7.98	-147.66 ±3.41
	Adv-TP (Guo et al., 2021)	-37.80 ±11.67	-74.26 ±1.00	-54.43 ±14.27	-135.36 ±4.84	-146.59 ±9.54
	Adv-xAI (Wu et al., 2021)	-44.78 ±11.98	-70.50 ±3.83	-67.18 ±2.14	-138.04 ±3.08	-150.23 ±1.91
	AMI (Li et al., 2025)	-37.70 ±10.94	-73.50 ±5.29	-61.04 ±7.68	-134.61 ±4.01	-149.07 ±2.41
	SA-RL (Zhang et al., 2021a)	-43.91 ±0.83	-84.75 ±4.33	-54.77 ±0.81	-111.35 ±3.53	-138.07 ±1.32
	PA-AD (Sun et al.)	-41.10 ±1.77	-77.11 ±2.79	-65.23 ±1.04	-119.17 ±1.36	-144.53 ±1.24
	RAT (Bai et al., 2025)	-34.28 ±6.96	-61.57 ±9.99	-46.59 ±8.46	-18.90 ±8.27	-81.63 ±24.02
	MatBak (Ours)	**-14.31 ±6.85**	**-19.51 ±2.57**	**-12.43 ±9.69**	**-6.96 ±10.51**	**-64.75 ±12.75**

additionally stopped vehicles before and after attacks). The experimental results are summarized in Table 8. It can be observed that MatBak achieves the highest ASR in 12 out of 15 scenarios. MatBak demonstrates particularly strong performance against the MAT-trained and the MAPPO-trained victims, achieving 95.00–100.00% ASR in most scenarios. In contrast, QMIX-trained victims exhibit lower susceptibility to targeted-behavior attacks across all methods (e.g., 0.00–26.67% in bottleneck) in MetaDrive scenarios. Notably, while PA-AD achieves the highest ASR in the MAT-SCSC scenario (71.67%), it shows limited effectiveness in other scenarios, whereas MatBak maintains consistently high success rates across diverse traffic configurations.

Table 8: Attack success rates of different methods on MetaDrive. Attacker agents are trained using the feedback signal from the reward model in PbRL settings. Each result is computed across 60 evaluation episodes.

Victim	Map	Adv-ZS	Adv-TP	Adv-xAI	AMI	SA-RL	PA-AD	RAT	MatBak (Ours)
MAT	intersection	33.33%	30.00%	38.33%	20.00%	36.67%	66.67%	**100.00%**	**100.00%**
	bottleneck	21.67%	48.33%	36.67%	28.33%	20.00%	78.33%	**100.00%**	**100.00%**
	roundabout	5.00%	25.00%	21.67%	15.00%	5.00%	63.33%	93.33%	**100.00%**
	SCSC	28.33%	40.00%	33.33%	26.67%	23.33%	**71.67%**	68.33%	18.33%
	SCSS	30.00%	55.00%	33.33%	45.00%	0.00%	93.33%	**100.00%**	**100.00%**
MAPPO	intersection	18.33%	33.33%	41.67%	30.00%	13.33%	31.67%	95.00%	**98.33%**
	bottleneck	31.67%	18.33%	25.00%	13.33%	18.33%	36.67%	**95.00%**	90.00%
	roundabout	1.67%	8.33%	5.00%	1.67%	0.00%	25.00%	83.33%	**95.00%**
	SCSC	30.00%	33.33%	36.67%	28.33%	33.33%	21.67%	46.67%	**55.00%**
	SCSS	28.33%	33.33%	41.67%	23.33%	16.67%	81.67%	**98.33%**	96.67%
QMIX	intersection	23.33%	30.00%	21.67%	26.67%	16.67%	30.00%	41.67%	**68.33%**
	bottleneck	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1.67%	**26.67%**
	roundabout	23.33%	38.33%	43.33%	36.67%	11.67%	46.67%	10.00%	**86.67%**
	SCSC	13.33%	13.33%	13.33%	18.33%	21.67%	8.33%	11.67%	**41.67%**
	SCSS	10.00%	16.67%	16.67%	10.00%	6.67%	50.00%	56.67%	**80.00%**

To further validate the performance of different methods in noise-perturbed scenarios, we add Gaussian noise with a magnitude of 5% of the observation scope to all victim observations during both training and evaluation. We summarize the experimental results in Figure 8. It can be observed that MatBak achieves the highest attacker return in 14 out of 15 noise-perturbed scenarios, demonstrating superior attack robustness to observation noise. In contrast, baseline methods such as Adv-ZS, Adv-TP, Adv-xAI, and AMI exhibit severe performance degradation under noise, with many settings collapsing to near-zero or identical low returns, which suggests that the baseline methods lack robustness when observation noise disrupts the perturbation structure, whereas MatBak's essential mechanisms enable more resilient targeted-behavior manipulation.

We also report attack success rates (ASRs) of different methods on MetaDrive in noise-perturbed settings, with experimental results summarized in Table 9. The results indicate that MatBak maintains strong attack effectiveness under noise, achieving the highest ASR in 14 out of 15 scenarios (including ties). Notably, MatBak achieves 95.00–100.00% ASR on most MAT and MAPPO scenarios, whereas baseline methods show dramatic performance degradation (e.g., Adv-ZS, Adv-TP, Adv-xAI, and AMI collapse to 0.00% on QMIX-bottleneck and 5.00% on MAT-roundabout). The performance gap is particularly pronounced on QMIX victims, where MatBak achieves 15.00–98.33% ASR compared to 0.00–48.33% for the strongest non-MatBak baseline PA-AD, further validating the robustness of MatBak's bi-level optimization framework against observation perturbations.

F.4.2 Additional Adversarial Training Results

We summarize the attacker return (Avg Attack R, i.e., the targeted behavior attacker reward) of the ATLA and WocaR models trained by MatBak and baseline methods on the 3s5z map of SMAC in Table 10. The results show that MatBak-ATLA and MatBak-WocaR models achieve the lowest average attacker return, indicating stronger robustness.

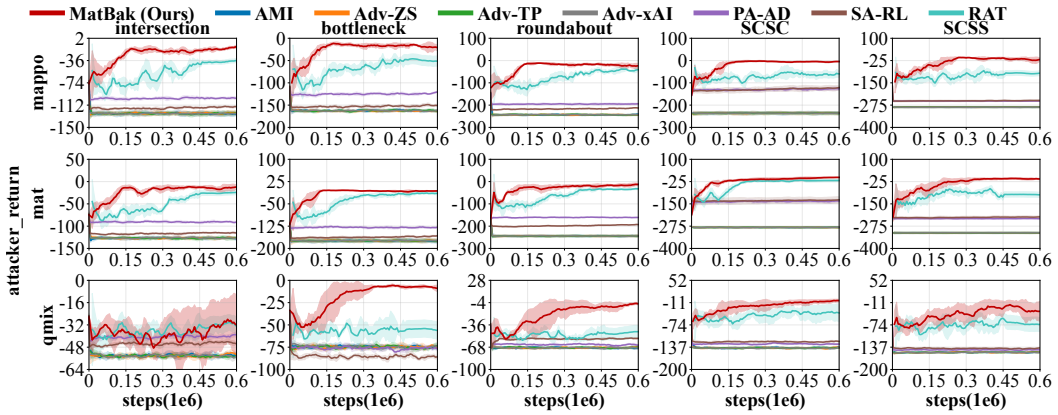

Figure 8: Adversarial attack results in noise-perturbed scenarios on MetaDrive (reported as average attacker return ± standard deviation). The figure labels show different scenarios and MARL algorithms used to train the victims.

Table 9: Attack success rates of different methods on MetaDrive in noise-perturbed settings. Attacker agents are trained using the feedback signal from the reward model in PbRL settings. Each result is computed across 60 evaluation episodes.

Victim	Map	Adv-ZS	Adv-TP	Adv-xAI	AMI	SA-RL	PA-AD	RAT	MatBak (Ours)
MAT	intersection	26.67%	26.67%	26.67%	26.67%	48.33%	50.00%	**98.33%**	**98.33%**
	bottleneck	16.67%	16.67%	16.67%	16.67%	11.67%	81.67%	96.67%	**100.00%**
	roundabout	5.00%	5.00%	5.00%	5.00%	5.00%	58.33%	58.33%	**95.00%**
	SCSC	20.00%	20.00%	20.00%	20.00%	40.00%	61.67%	**78.33%**	36.67%
	SCSS	0.00%	0.00%	0.00%	0.00%	0.00%	93.33%	**100.00%**	**100.00%**
MAPPO	intersection	26.67%	26.67%	26.67%	26.67%	33.33%	30.00%	78.33%	**96.67%**
	bottleneck	8.33%	8.33%	8.33%	8.33%	10.00%	30.00%	71.67%	**98.33%**
	roundabout	0.00%	0.00%	0.00%	0.00%	1.67%	21.67%	45.00%	**100.00%**
	SCSC	31.67%	31.67%	31.67%	31.67%	35.00%	26.67%	30.00%	**36.67%**
	SCSS	16.67%	16.67%	16.67%	16.67%	13.33%	85.00%	**95.00%**	**95.00%**
QMIX	intersection	11.67%	11.67%	11.67%	11.67%	23.33%	31.67%	35.00%	**63.33%**
	bottleneck	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	**15.00%**
	roundabout	13.33%	13.33%	13.33%	13.33%	15.00%	46.67%	11.67%	**98.33%**
	SCSC	15.00%	15.00%	15.00%	15.00%	20.00%	16.67%	10.00%	**51.67%**
	SCSS	11.67%	11.67%	11.67%	11.67%	6.67%	48.33%	31.67%	**73.33%**

Table 10: Attacker return of ATLA and WocaR models trained by different adversarial attack methods on the 3s5z map of SMAC. The average attacker return (Avg Attack R) column reports mean attacker return ±standard deviation. The victim models are trained by the MAT algorithm (Wen et al., 2022), and the results are averaged across 100 episodes.

Model	Adv-ZS	Adv-TP	Adv-xAI	AMI	SA-RL	PA-AD	RAT	MatBak	Avg Attack R
Adv-ZS-ATLA (Gleave et al., 2020)	378.52	385.58	391.07	411.43	442.90	265.22	357.91	534.40	395.88±156.11
Adv-TP-ATLA (Guo et al., 2021)	363.05	357.90	347.85	339.28	395.07	296.65	263.06	364.07	340.87±100.93
Adv-xAI-ATLA (Wu et al., 2021)	451.36	383.18	436.26	467.68	450.39	313.73	317.98	539.04	419.95±152.79
AMI-ATLA (Li et al., 2025)	377.58	323.79	397.93	359.47	420.75	310.77	458.34	505.40	394.25±169.82
SA-RL-ATLA (Zhang et al., 2021a)	291.05	296.89	**257.31**	**245.72**	365.71	416.28	440.87	301.46	326.91±114.64
PA-AD-ATLA (Sun et al.)	322.43	307.27	314.29	299.79	353.12	317.42	308.22	407.10	328.71±116.30
RAT-ATLA (Bai et al., 2025)	291.92	307.22	306.99	287.56	391.35	293.55	401.58	305.89	323.26±116.45
RAT-WocaR (Bai et al., 2025)	400.05	431.65	411.64	391.70	416.72	261.40	404.15	420.15	392.18±156.53
MatBak-ATLA (Ours)	**251.31**	255.92	258.13	256.31	**251.88**	199.92	274.46	**266.69**	**251.83±93.64**
MatBak-WocaR (Ours)	299.66	**252.84**	258.20	277.70	282.28	**164.18**	**258.92**	321.11	264.36±108.06

We also report visualization results of ATLA models trained by different methods under cross-attacks on SMAC in Figure 9. Specifically, we visualize the spatial formations of four MAT-trained models on the 3s5z map: the base model (no adversarial training), ATLA-AMI, ATLA-RAT, and ATLA-MatBak, each evaluated under attacks from AMI, RAT, and MatBak. The experimental results show that models trained with ATLA using AMI or RAT adversaries still exhibit dispersed formations when attacked by methods different from their training adversary (e.g., ATLA-AMI disperses under AMI, RAT and MatBak attacks; ATLA-RAT disperses under AMI and MatBak attacks), indicating limited generalization to unseen attack strategies. In contrast, the ATLA-MatBak model maintains compact formations under all three attack methods, demonstrating that MatBak's preference-based targeted-behavior attacks provide more comprehensive adversarial training signals that enhance robustness against diverse dispersed-formation induction strategies.

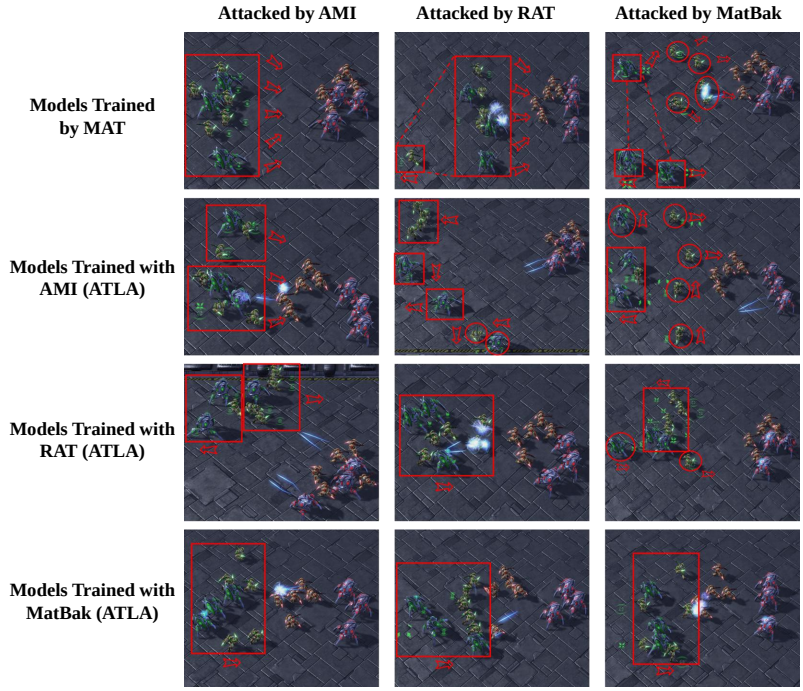

Figure 9: A visualization of ATLA models trained by AMI, RAT, and MatBak (ours) under targeted-behavior adversarial attacks of models trained by AMI, RAT, and MatBak (ours).

F.4.3 Additional Case Study and Visualizations

We summarize the full visualization results of adversarial attacks by MatBak (ours), RAT, and AMI on MetaDrive in Figure 10. The attacker's intention for all methods is designed to induce a team-level targeted behavior of traffic standstill (i.e., to reduce vehicle speed and travel distance). Under the same condition of attacking 40% of the victim agents, MatBak induces larger-scale traffic standstill in the victim MAS, which validates its effectiveness and indicates potential safety risks in real-world traffic systems.

F.5 Hyperparameter Settings

This section presents the detailed hyperparameter settings for all algorithms adopted in experiments. Specifically:

Victim training. Table 11 shows the hyperparameters we used for MAT in the SMAC experiments. Table 12 shows the map-specific hyperparameters we used for MAT in the SMAC experiments. Table 13 shows the hyperparameters we used for MAPPO in the SMAC experiments. Table 14 shows the hyperparameters we used for QMIX in the SMAC experiments.

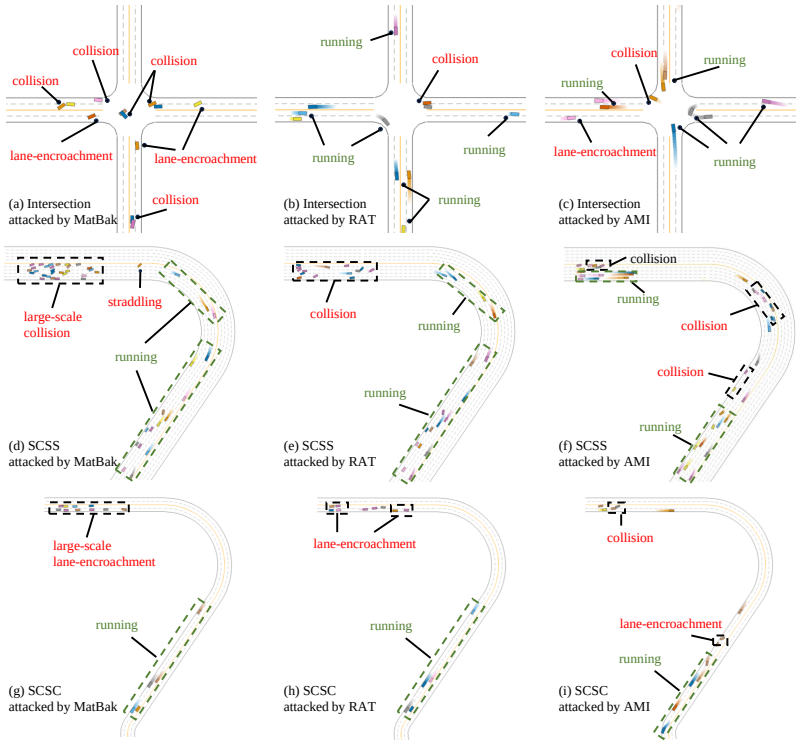

Figure 10: Full visualization results of adversarial attacks by MatBak (ours), RAT, and AMI in intersection (subfigures (a)-(c)), SCSS (subfigures (d)-(f)), and SCSC (subfigures (g)-(i)) scenarios on MetaDrive. The victim agents are trained by the MAT algorithm.

Table 15 shows the common hyperparameters we used for MAT, MAPPO, and QMIX in the MetaDrive experiments. Table 16 shows the hyperparameters we used for MAT in the MetaDrive experiments. Table 17 shows the hyperparameters we used for MAPPO in the MetaDrive experiments. Table 18 shows the hyperparameters we used for QMIX in the MetaDrive experiments.

Table 11: Hyperparameters used for MAT on the SMAC benchmark. The parameter settings are identical to those provided in the MAT paper (Wen et al., 2022).

Hyperparameters	Value	Hyperparameters	Value	Hyperparameters	Value
critic lr	5e-4	actor lr	5e-4	optim eps	1e-5
gain	0.01	batch size	3200	num mini-batch	1
use gae	True	use huber loss	True	optimizer	Adam
max grad norm	10	entropy coef	0.01	stacked frames	1
training threads	16	rollout threads	32	episode length	100
num blocks	1	num heads	1		

Table 12: Map-specific hyperparameters used for MAT on the SMAC benchmark. The parameter settings are identical to those provided in the MAT paper (Wen et al., 2022).

maps	ppo epochs	ppo clip	steps	γ
8m	15	0.2	1e6	0.99
3s5z	10	0.05	3e6	0.99
MMM	15	0.2	2e6	0.99
MMM2	5	0.05	1e7	0.99

Table 13: Hyperparameters used for MAPPO on the SMAC benchmark. The parameter settings are identical to those provided in the MAPPO paper (Yu et al., 2022).

Hyperparameters	Value	Hyperparameters	Value	Hyperparameters	Value
critic lr	5e-4	actor lr	5e-4	gae lambda	0.95
gain	0.01	use gae	True	use popart	False
optim eps	1e-5	num mini-batch	1	entropy coef	0.01
stacked frames	1	ppo epochs	5	ppo clip	0.05
training threads	16	rollout threads	32	episode length	100
optimizer	Adam	weight decay	0	network initialization	Orthogonal
gamma	0.99	epsilon anneal time	5e4	critic loss	mse loss
buffer size	5000	batch size	32	use reward normalization	True
buffer length	400	num envs	8	use feature normalization	True
num GRU layers	1	RNN hidden state dim	64	fc layer dim	64
num fc	2	num fc after	1	steps	1e7

Table 14: Hyperparameters used for QMIX on the SMAC benchmark. The parameter settings are identical to those provided in the PyMARL2 paper (Hu et al., 2021).

Hyperparameters	Value	Hyperparameters	Value
epsilon start	1.0	learning rate	1e-3
epsilon finish	0.05	TD lambda	0.6
epsilon anneal time	100000	hypernet embedding	64
batch size	128	mixing embedding dim	32
optimizer	Adam	target update interval	200
buffer size	5000	batch size run	8
gamma	0.99	steps	4e6

Table 15: Common hyperparameters used for MAT, MAPPO, and QMIX on the MetaDrive benchmark.

maps	env type	num agents	lane num	exit length	rollout threads	batch size
bottleneck	bottleneck	10	4	60	20	4000
intersection	intersection	10	2	60	20	4000
roundabout	roundabout	10	4	60	32	6400
SCSS	pg	30	5	80	20	4000
SCSC	pg	10	2	50	20	4000

Table 16: Hyperparameters used for MAT on the MetaDrive benchmark.

Hyperparameters	Value	Hyperparameters	Value	Hyperparameters	Value
critic lr	5e-4	actor lr	5e-4	optim eps	1e-5
gain	0.01	γ	0.99	num mini-batch	1
use gae	True	use huber loss	True	optimizer	Adam
max grad norm	10	entropy coef	0.01	episode length	200
training threads	16	num embedding	64	ppo clip	0.05
ppo epochs	10	gae lambda	0.95	num blocks	1
num heads	1				

Table 17: Hyperparameters used for MAPPO on the MetaDrive benchmark.

Hyperparameters	Value	Hyperparameters	Value	Hyperparameters	Value
critic lr	5e-4	actor lr	5e-4	optim eps	1e-5
gain	0.01	recurrent policy	True	num mini-batch	1
use gae	True	use huber loss	True	optimizer	Adam
max grad norm	10	entropy coef	0.01	episode length	200
training threads	16	num recurrents	1	ppo clip	0.05
ppo epochs	10	steps	5e6	gamma	0.99
gae lambda	0.95	hidden size	64	n layer	1
use value action masks	False				

Table 18: Hyperparameters used for QMIX on the MetaDrive benchmark.

Hyperparameters	Value	Hyperparameters	Value
epsilon start	1.0	learning rate	1e-3
epsilon finish	0.05	TD lambda	0.6
epsilon anneal time	100000	batch size	128
episode length	200	hypernet embedding	64
mixing embedding dim	32	RNN hidden dim	64
grad norm clip	10	q lambda	False
gamma	0.99	target update interval	200
buffer size	5000	action selector	epsilon-greedy
optimizer	Adam	prioritized replay	False

Attacker training. Table 19 shows the common hyperparameters we used for training the MAPPO attacker of Adv-ZS, Adv-TP, Adv-xAI, and AMI. Table 20 shows the common hyperparameters we used for training the PPO attacker of SA-RL, PA-AD, RAT, and MatBak.

Table 21 shows the hyperparameters we used for Adv-ZS. Table 22 shows the hyperparameters we used for Adv-TP. Table 23 shows the hyperparameters we used for Adv-xAI. Table 24 shows the hyperparameters we used for AMI. Table 25 shows the hyperparameters we used for SA-RL. Table 26 shows the hyperparameters used for PA-AD. Table 27 shows the hyperparameters we used for RAT. Table 28 shows the hyperparameters used for MatBak. Table 29 shows the benchmark-specific hyperparameters we used for RAT and MatBak.

Table 19: Common hyperparameters used for training the MAPPO attacker of Adv-ZS, Adv-TP, Adv-xAI, and AMI. The parameter settings are identical to those provided in the AMI paper (Li et al., 2025).

Hyperparameters	Value	Hyperparameters	Value	Hyperparameters	Value
critic lr	1e-4	actor lr	1e-4	gae lambda	0.95
Q nstep	5	ppo epochs	4	ppo clip	0.2
optim eps	1e-5	num mini-batch	1	entropy coef	0.01
training threads	16	rollout threads	16	episode length	100
optimizer	Adam	activation	ReLU	hidden layer	1
γ	0.99	eval episode	20	standardise rewards	True
standardise returns	False	ε	0.8	norm clipping	10
hidden units	64	stacked frames	1	(β_1, β_2)	(0.9, 0.999)

Table 20: Common hyperparameters used for training the PPO attacker of SA-RL, PA-AD, RAT, and MatBak.

Hyperparameters	Value	Hyperparameters	Value
Number of layers	4	Hidden units of each layer	128
actor lr	5e-5	critic lr	1e-4
γ	0.95	ε	0.8
ppo clip	0.2	ppo epochs	20
mini-batches	4	batch size	32
buffer size	16 $\times$ episode length	(β_1, β_2)	(0.9, 0.999)
rollout threads	16		

Table 21: Hyperparameters used for training the MAPPO attacker of Adv-ZS.

Hyperparameters	Value
Victim critic learning rate	1e-4

Table 22: Hyperparameters used for training the MAPPO attacker of Adv-TP.

Hyperparameters	Value
Victim critic learning rate	1e-4

Table 23: Hyperparameters used for training the MAPPO attacker of Adv-xAI.

Hyperparameters	Value	Hyperparameters	Value
ϵ_{state}	0.1	lr_{state}	1e-4
ϵ_{action}	0.1	$\text{lr}_{\text{action}}$	1e-4

Table 24: Hyperparameters used for training the MAPPO attacker of AMI. The parameter settings are identical to those provided in the AMI paper (Li et al., 2025).

Hyperparameters	Value	Hyperparameters	Value
p_ϕ learning rate	= base learning rate	TAO critic learning rate	= base learning rate
AMI λ	0.1	sensitivity metric	L1
use sampled MAA	True	MAA entropy regularization	0
use target sensitivity reward	True	use position mask	False
TAO learning rate	= base learning rate	AMI λ	[0.03, 0.05, 0.1]

Table 25: Hyperparameters used for training the PPO attacker of SA-RL.

Hyperparameters	Value	Hyperparameters	Value
attacked-agent selection	random	attack mask hold steps	10

Table 26: Hyperparameters used for training the PPO attacker of PA-AD.

Hyperparameters	Value	Hyperparameters	Value
attack method	FGSM	attacked-agent selection	random
attack mask hold steps	10		

Table 27: Hyperparameters used for training the PPO attacker of RAT.

Hyperparameters	Value	Hyperparameters	Value
Intention blocks	1	Intention embedding	64
Intention heads	1	Intention rollout threads	16
Intention learning rate	5e-4	Intention update frequency	50
Intention PPO epochs	10	Intention mini-batches	1
Intention clip	0.05	Intention entropy coefficient	0.01
Intention discount	0.99	Intention GAE lambda	0.95
Intention reward source	Reward model	Reward model hidden layers	3
Reward model hidden units	256	Reward model learning rate	3e-4
Reward ensemble size	3	Segment length	10
Feedback batch	64	Feedback interval	1000
Preference buffer capacity	5000	Reward model train batch size	128
Reward model activation	`tanh`	Reward update epochs	200
Max feedback	15000	Preference sampling strategy	Disagreement
Use state controller h_ω	True	State-controller hidden units	64
State-controller hidden depth	2	State-controller learning rate	1e-3
Outer PPO clip parameter	0.2	ε	0.8
Outer optimization frequency	20	Use outer optimization	True
Random target selection	True	Attack mask hold steps	10
Inner updates	4	Inner batch size	32

Table 28: Hyperparameters used for training the PPO attacker of MatBak.

Hyperparameters	Value	Hyperparameters	Value
Intention blocks	1	Intention embedding	64
Intention heads	1	Intention rollout threads	16
Intention learning rate	5×10^{-4}	Intention update frequency	50
Intention PPO epochs	10	Intention mini-batches	1
Intention clip	0.05	Intention entropy coefficient	0.01
Intention discount	0.99	Intention GAE lambda	0.95
Intention reward source	Reward model	Reward model hidden layers	3
Reward model hidden units	256	Reward model learning rate	3e-4
Reward ensemble size	3	Segment length	10
Feedback batch	64	Feedback interval	1000
Preference buffer capacity	5000	Reward model train batch size	128
Reward model activation	`tanh`	Reward update epochs	200
Use victim-observation encoder	True	Encoder blocks	2
Encoder embedding	128	Encoder heads	4
Encoder learning rate	5e-4	Encoder target update frequency	2
Encoder TD discount	0.95	Use state controller h_ω	True
Use victim controller ϕ_ψ	True	State-controller hidden units	64
State-controller hidden depth	2	State-controller learning rate	1e-3
Victim-controller hidden units	64	Victim-controller hidden depth	2
Victim-controller learning rate	1e-3	Inner updates	4
Max feedback	15000	ε	0.8
Inner batch size	32	Outer optimization frequency	20
Use outer optimization	True	Outer PPO clip parameter	0.2
Random target selection	False	Attack mask hold steps	10

Table 29: Benchmark-specific hyperparameters used for RAT and MatBak on SMAC and MetaDrive.

Benchmark	Victim	Map	Controller freeze steps
SMAC	MAT	8m	2.5e5
		3s5z	7e5
		MMM	1.5e5
		MMM2	7e5
	MAPPO	8m	7e5
		3s5z	7e5
		MMM	2.5e5
		MMM2	7e5
	QMIX	8m	7e5
		3s5z	3.5e5
		MMM	7e5
		MMM2	7e5
MetaDrive	MAT	Bottleneck	7e5
		Intersection	7e5
		Roundabout	7e5
		SCSC	2e5
		SCSS	7e5
	MAPPO	Bottleneck	2e5
		Intersection	7e5
		Roundabout	7e5
		SCSC	2e5
		SCSS	7e5
	QMIX	Bottleneck	7e5
		Intersection	2e5
		Roundabout	7e5
		SCSC	2e5
		SCSS	2e5