A Proofs for LPA-Dec-POMDP

Lemma 1 (Bellman equation for fixed $\boldsymbol{\pi}$ and π_{adv})**.** *Given $\hat{\mathcal{M}} = \langle \mathcal{N}, \mathcal{S}, \mathcal{A}, P, K, \Omega, O, R, \gamma \rangle$, $\boldsymbol{\pi}$ and π_{adv}, we have*

$$\begin{aligned}\hat{V}_{\boldsymbol{\pi} \circ \pi_{adv}}(s,k) =& \mathbb{E}_{\boldsymbol{a}, \hat{\boldsymbol{a}} \sim \boldsymbol{\pi} \circ \pi_{adv}(s,k)}[\mathbb{E}_{s' \sim P(\cdot|s', \boldsymbol{a})}[R(s, \hat{\boldsymbol{a}}, s') \\ & + \gamma \hat{V}_{\boldsymbol{\pi} \circ \pi_{adv}}(s', k - \mathbb{I}(\hat{\boldsymbol{a}} \neq \boldsymbol{a}))]],\end{aligned} \tag{1}$$

where $\boldsymbol{\pi}(\boldsymbol{a}|s) = \prod_{i \in \mathcal{N}} \pi^i(a^i|O(s,i))$, $\mathbb{E}_{\boldsymbol{a}, \hat{\boldsymbol{a}} \sim \boldsymbol{\pi} \circ \pi_{adv}(s,k)}[f] = \sum_{\boldsymbol{a} \in \mathcal{A}} \boldsymbol{\pi}(\boldsymbol{a}|s) \sum_{\hat{\boldsymbol{a}} \in \mathcal{A}} \pi_{adv}(\hat{\boldsymbol{a}}|s, \boldsymbol{a}, k) f$.

Proof. Based on the definition of $\hat{V}_{\boldsymbol{\pi} \circ \pi_{adv}}(s,k)$:

$$\begin{aligned}&\hat{V}_{\boldsymbol{\pi} \circ \pi_{adv}}(s,k) \\ &= \mathbb{E}_{\boldsymbol{\pi} \circ \pi_{adv}}[\sum_{n=0}^{\infty} \gamma^n r_{t+n+1} | s_t = s, k_t = k] \\ &= \mathbb{E}_{\boldsymbol{\pi} \circ \pi_{adv}}[r_{t+1} + \sum_{n=0}^{\infty} \gamma^n r_{t+n+2} | s_t = s, k_t = k] \\ &= \sum_{\boldsymbol{a} \in \mathcal{A}} \boldsymbol{\pi}(\boldsymbol{a}|s) \sum_{\hat{\boldsymbol{a}} \in \mathcal{A}} \pi_{adv}(\hat{\boldsymbol{a}}|s, \boldsymbol{a}, k) \sum_{s' \in \mathcal{S}} P(s'|s, \hat{\boldsymbol{a}})[r_{t+1} + \\ &\gamma \mathbb{E}_{\boldsymbol{\pi} \circ \pi_{adv}}[\sum_{n=0}^{\infty} \gamma^n r_{t+n+2} | s_{t+1} = s', k_{t+1} = k - \mathbb{I}(\hat{\boldsymbol{a}} \neq \boldsymbol{a})]] \\ &= \mathbb{E}_{\boldsymbol{a}, \hat{\boldsymbol{a}} \sim \boldsymbol{\pi} \circ \pi_{adv}(s,k)}[\mathbb{E}_{s' \sim P(\cdot|s', \hat{\boldsymbol{a}})}[R(s, \hat{\boldsymbol{a}}, s') \\ &+ \gamma \hat{V}_{\boldsymbol{\pi} \circ \pi_{adv}}(s', k - \mathbb{I}(\hat{\boldsymbol{a}} \neq \boldsymbol{a}))]].\end{aligned} \tag{2}$$

The state-value function $\hat{Q}_{\boldsymbol{\pi} \circ \boldsymbol{\pi_{adv}}}(s, \boldsymbol{a}, k)$ can be derived in a similar way.

Theorem 1. *Given an LPA-Dec-POMDP $\hat{\mathcal{M}} = \langle \mathcal{N}, \mathcal{S}, \mathcal{A}, P, K, \Omega, O, R, \gamma \rangle$, a fixed joint policy $\boldsymbol{\pi}$ of the ego-system and a heuristic-based policy perturbation function g, there exists an MDP $\bar{\mathcal{M}} = (\bar{\mathcal{S}}, \bar{\mathcal{A}}, \bar{P}, \bar{R}, \gamma)$ such that the optimal adversarial attacker π^*_{adv} for LPA-Dec-POMDP is disentangled by an optimal policy v^* of $\bar{\mathcal{M}}$ and g, where $\bar{\mathcal{S}} = \mathcal{S} \times \mathbb{N}$, $\bar{s} = (s,k), \bar{s}' = (s', k'), k, k' \leq K$ indicates the remaining attack budget, $\bar{\mathcal{A}} = \mathcal{N} \cup \{null\}$, $\bar{R}(\bar{s}, \bar{a}, \bar{s}') = -R(s, \hat{\boldsymbol{a}}, s')$,*

$$\bar{P}(\bar{s}'|\bar{s}, \bar{a}) = \begin{cases} 0 & k - k' \notin \{0, 1\} \\ P(s'|s, \hat{\boldsymbol{a}}) \mathbb{I}(\hat{\boldsymbol{a}} = \boldsymbol{a}) & k - k' = 0 \\ P(s'|s, \hat{\boldsymbol{a}}) \mathbb{I}(\hat{\boldsymbol{a}} \neq \boldsymbol{a}) & k - k' = 1, \end{cases}$$

where $\bar{d}(\bar{s}_0) = d(s_0) \mathbb{I}(k_0 = K)$, d and $\bar{d}$ are distributions over initial state in $\hat{\mathcal{M}}$ and $\bar{\mathcal{M}}$, respectively, $\boldsymbol{a} = \boldsymbol{\pi}(s)$, $\hat{\boldsymbol{a}} = g(\bar{a}, \boldsymbol{\pi}(s), k)$ are original and forced action of ego-system, respectively.

Proof. To prove this theorem, we first prove that, given a fixed joint ego-agent policy $\boldsymbol{\pi}$ and a heuristic-based policy perturbation function g, $\forall v$, we have $\hat{V}_{\boldsymbol{\pi} \circ \pi_{adv}}(s,k) = -\bar{V}_v(\bar{s})$, where $\pi_{adv} = v \circ g$, $\bar{s} = (s,k)$.

Based on the Bellman equation for $\bar{V}$, we could derive:

$$\begin{aligned}&\bar{V}(\bar{s}) \\ &= \sum_{\bar{a} \sim \bar{\mathcal{A}}} v(\bar{a}|\bar{s}) \sum_{\bar{s}' \sim \bar{\mathcal{S}}} P(\bar{s}'|\bar{s}, \bar{a})[\bar{R}(\bar{s}, \bar{a}, \bar{s}') + \gamma \bar{V}(\bar{s}')] \\ &= \sum_{i \in \hat{\mathcal{N}}} v(i|s,k)[\sum_{s' \sim \mathcal{S}} P(s'|s, \hat{\boldsymbol{a}}) \mathbb{I}(\hat{\boldsymbol{a}} = \boldsymbol{v}(s))[-R(s, \hat{\boldsymbol{a}}, s') + \\ &\quad \gamma \bar{V}_v((s',k))] + \sum_{s' \sim \mathcal{S}} P(s'|s, \hat{\boldsymbol{a}}) \mathbb{I}(\hat{\boldsymbol{a}} \neq \boldsymbol{v}(s))[-R(s, \hat{\boldsymbol{a}}, s') \\ &\quad - \gamma \bar{V}_v((s', k-1))]] \\ &= \sum_{i \in \hat{\mathcal{N}}} v(i|s,k)[\sum_{s' \sim \mathcal{S}} P(s'|s, \hat{\boldsymbol{a}})[-R(s, \hat{\boldsymbol{a}}, s') + \\ &\quad \gamma \bar{V}_v((s', k - \mathbb{I}(\hat{\boldsymbol{a}} \neq \boldsymbol{a})))] \\ &= \sum_{\boldsymbol{a} \in \mathcal{A}} \boldsymbol{\pi}(\boldsymbol{a}|s) \sum_{\hat{\boldsymbol{a}} \in \mathcal{A}} \pi_{adv}(\hat{\boldsymbol{a}}|s, \boldsymbol{a}, k) \sum_{s' \in \mathcal{S}} P(s'|s, \hat{\boldsymbol{a}})[-R(s, \\ &\quad \hat{\boldsymbol{a}}, s') + \gamma \bar{V}((s', k - \mathbb{I}(\hat{\boldsymbol{a}} \neq \boldsymbol{a}))).\end{aligned} \tag{3}$$

Then, we discuss the case that $\forall s \in \mathcal{S}, k = 0$, where $\pi_{adv}(\hat{\boldsymbol{a}}|s, \boldsymbol{a}, 0) = \mathbb{I}(\hat{\boldsymbol{a}} = \boldsymbol{a})$:

$$\begin{aligned}&\hat{V}_{\boldsymbol{\pi} \circ \pi_{adv}}(s,0) \\ =& \sum_{\boldsymbol{a} \in \mathcal{A}} \boldsymbol{\pi}(\boldsymbol{a}|s) \sum_{s' \in \mathcal{S}} P(s'|s, \boldsymbol{a})[R(s, \hat{\boldsymbol{a}}, s') + \gamma \hat{V}_{\boldsymbol{\pi} \circ \pi_{adv}}(s', 0)] \\ =& V_{\boldsymbol{\pi}}(s) \\ &\bar{V}_v((s,0)) \\ =& \sum_{\bar{a} \sim \bar{\mathcal{A}}} v(\bar{a}|\bar{s}) g(\bar{a}, \boldsymbol{\pi}(s), 0) \sum_{s' \sim \mathcal{S}} P(s'|s, \hat{\boldsymbol{a}})[-R(s, \hat{\boldsymbol{a}}, s') + \\ &\quad \gamma \bar{V}_v((s,0))] \\ =& \sum_{\boldsymbol{a} \in \mathcal{A}} \boldsymbol{\pi}(\boldsymbol{a}|s) \sum_{s' \in \mathcal{S}} P(s'|s, \boldsymbol{a})[-R(s, \hat{\boldsymbol{a}}, s') + \gamma \bar{V}_v((s,0))] \\ =& - V_{\boldsymbol{\pi}}(s),\end{aligned} \tag{4}$$

where $V_{\boldsymbol{\pi}}(s)$ is the value function under the original Dec-POMDP.

Accordingly, we have that $k = 0$, $\hat{V}_{\boldsymbol{\pi} \circ \pi_{adv}}(s, 0) = -\bar{V}_v((s,0)), \forall s \in \mathcal{S}$. Then we could derive that $\hat{V}_{\boldsymbol{\pi} \circ \pi_{adv}}(s,k) = -\bar{V}_v((s,k)), , \forall s \in \mathcal{S}, \forall k \in \{0, 1, ..., K\}$ based on Eq.(2) and Eq.(3).

Besides, for $\bar{M}$, we have $\forall (\bar{s}, \bar{a}, \bar{s}')$,

$$-\max_{s, \hat{\boldsymbol{a}}, s'} R(s, \hat{\boldsymbol{a}}, s') \leq \bar{R}(\bar{s}, \bar{a}, \bar{s}') \leq -\min_{s, \hat{\boldsymbol{a}}, s'} R(s, \hat{\boldsymbol{a}}, s').$$

Based on the basic property of MDP, there exist an optimal policy v^* for $\bar{M}$, such that $\bar{V}_{v^*}(\bar{s}) \geq \bar{V}_v(\bar{s})$,$\forall \bar{s} \in \bar{\mathcal{S}}, \forall v$.

Now we have that the optimal adversarial attacker π^*_{adv} for LPA-Dec-POMDP is disentangled by an optimal policy v^* of $\bar{M}$ and g, that is $\pi^*_{adv} = v^* \circ g$. By optimizing v in $\bar{\mathcal{M}}$, we are able to optimize the adversarial attacker in $\hat{\mathcal{M}}$.

Theorem 2. *Given $\hat{\mathcal{M}} = \langle \mathcal{N}, \mathcal{S}, \mathcal{A}, P, K, \Omega, O, R, \gamma \rangle$, a fixed deterministic adversarial attacker policy π_{adv}, there* *exists a Dec-POMDP* $\tilde{\mathcal{M}} = \langle \mathcal{N}, \tilde{\mathcal{S}}, \mathcal{A}, \tilde{P}, \Omega, \tilde{O}, \tilde{R}, \gamma \rangle$ *such that the optimal policy of* $\tilde{\mathcal{M}}$ *is the optimal policy for LPA-Dec-POMDP given* π_{adv}*, where* $\tilde{\mathcal{S}} = \mathcal{S} \times \mathbb{N}$*,* $\tilde{d}(\tilde{s}_0) = d(s_0)\mathbb{I}(k_0 = K)$*,* $\tilde{O}(\tilde{s}, i) = O(s, i)$*,* $\tilde{R}(\tilde{s}, \boldsymbol{a}, \tilde{s}') = R(s, \hat{\boldsymbol{a}}, s')$*,*

$$\tilde{P}(\tilde{s}'|\tilde{s}, \boldsymbol{a}) = \begin{cases} 0 & k - k' \notin \{0, 1\} \\ P(s'|s, \hat{\boldsymbol{a}}) & otherwise, \end{cases}$$

where $\tilde{\mathrm{d}}$ *and* d *are distributions over initial state in* $\tilde{\mathcal{M}}$ *and* $\hat{\mathcal{M}}$*, respectively, and* $\tilde{s} = (s, k)$*,* $\tilde{s}' = (s', k')$*,* $\hat{\boldsymbol{a}} = \pi_{adv}(s, \boldsymbol{a}, k)$*.*

Proof. Following the same idea in the proof of Thm.1, we aim to prove that, given a deterministic π_{adv}, $\forall \boldsymbol{\pi}$,we have $\hat{V}_{\boldsymbol{\pi}\circ\pi_{adv}}(s, k) = \tilde{V}_{\boldsymbol{\pi}}(\tilde{s})$, $\forall s \in \mathcal{S}, k \in \{0, 1, ..., K\}$, where $\tilde{s} = (s, k)$.

For $\tilde{\mathcal{M}}$, we have Bellman Equation:

$$\begin{aligned} &\tilde{V}_{\boldsymbol{\pi}}(\tilde{s}) \\ &= \sum_{\boldsymbol{a}\sim\mathcal{A}} \boldsymbol{\pi}(\boldsymbol{a}|\tilde{s}) \sum_{\tilde{s}'\in\tilde{\mathcal{S}}} \tilde{P}(\tilde{s}'|\tilde{s}, \boldsymbol{a})[\tilde{R}(\tilde{s}, \boldsymbol{a}, \tilde{s}') + \gamma \tilde{V}_{\boldsymbol{\pi}}(\tilde{s}')] \\ &= \sum_{\boldsymbol{a}\sim\mathcal{A}} \boldsymbol{\pi}(\boldsymbol{a}|\tilde{s}) \sum_{s'\in\mathcal{S}} P(s'|s, \hat{\boldsymbol{a}})[R(s, \hat{\boldsymbol{a}}, s') + \gamma \tilde{V}_{\boldsymbol{\pi}}((s', k \\ &- \mathbb{I}(\hat{\boldsymbol{a}} \neq \boldsymbol{a})))], \end{aligned} \tag{5}$$

where $\hat{\boldsymbol{a}} = \pi_{adv}(s, \boldsymbol{a}, k)$.

By substituting $\pi_{adv}(\hat{\boldsymbol{a}}|s, \boldsymbol{a}, k)$ with a deterministic policy in Eq.(2), we have:

$$\begin{aligned} &\hat{V}_{\boldsymbol{\pi}\circ\pi_{adv}}(s, k) \\ &= \sum_{\boldsymbol{a}\sim\mathcal{A}} \boldsymbol{\pi}(\boldsymbol{a}|s) \sum_{s'\in\mathcal{S}} P(s'|s, \hat{\boldsymbol{a}})[R(s, \hat{\boldsymbol{a}}) + \gamma \hat{V}_{\boldsymbol{\pi}\circ\pi_{adv}}((s', k \\ &- \mathbb{I}(\hat{\boldsymbol{a}} \neq \boldsymbol{a})))]. \end{aligned} \tag{6}$$

We do not distinguish between $\boldsymbol{\pi}(\boldsymbol{a}|\tilde{s})$ and $\boldsymbol{\pi}(\boldsymbol{a}|s)$ since $\boldsymbol{\pi}(\boldsymbol{a}|\tilde{s}) = \prod_{i\in\mathcal{N}} \pi^i(a^i|\tilde{O}(\tilde{s}, i)) = \prod_{i\in\mathcal{N}} \pi^i(a^i|O(s, i)) = \boldsymbol{\pi}(\boldsymbol{a}|s)$.

Similar to the proof in Thm.(1), we could derive that $\forall \boldsymbol{\pi}$, $k = 0$, we have $\hat{V}_{\boldsymbol{\pi}\circ\pi_{adv}}(s, 0) = \tilde{V}_{\boldsymbol{\pi}}(\tilde{s})$, $\forall s \in \mathcal{S}$, where $\tilde{s} = (s, 0)$. Combined with Eq.(5) and Eq.(6),for $\forall \boldsymbol{\pi}$, we have $\hat{V}_{\boldsymbol{\pi}\circ\pi_{adv}}(s, k) = \tilde{V}_{\boldsymbol{\pi}}(\tilde{s})$, $\forall s \in \mathcal{S}, k \in \{0, 1, ..., K\}$, where $\tilde{s} = (s, k)$. The optimal policy $\boldsymbol{\pi}$ in $\tilde{M}$ is also the optimal policy in original LPA-Dec-POMDP $\hat{\mathcal{M}}$.

Theorem 3. *Given* $\hat{\mathcal{M}} = \langle \mathcal{N}, \mathcal{S}, \mathcal{A}, P, K, \Omega, O, R, \gamma \rangle$*, a stochastic adversarial attacker policy* π_{adv}*, there exists an Dec-POMDP* $\tilde{\mathcal{M}} = \langle \mathcal{N}, \tilde{\mathcal{S}}, \mathcal{A}, \tilde{P}, \Omega, \tilde{O}, \tilde{R}, \gamma \rangle$ *such that* $\forall \boldsymbol{\pi}$*, we have* $\tilde{V}_{\boldsymbol{\pi}}(\tilde{s}) \leq \hat{V}_{\boldsymbol{\pi}\circ\pi_{adv}}(s, k)$*, where* $\tilde{s} = (s, k)$*,* $\hat{V}_{\boldsymbol{\pi}\circ\pi_{adv}}(s, k)$ *denotes the state value function in the original LPA-Dec-POMDP,* $\forall s \in \mathcal{S}, \forall k \in \{0, 1, ..., K\}$*.*

Proof. Similar to Thm.(2), we define the reward and transition functions as follows:

$$\tilde{R}(\tilde{s}, \boldsymbol{a}, \tilde{s}') = \begin{cases} R(s, \boldsymbol{a}, s') & k - k' = 0 \\ \hat{R}(s, \boldsymbol{a}, s', k) & otherwise \end{cases}$$

$$\tilde{P}(\tilde{s}'|\tilde{s}, \boldsymbol{a}) = \begin{cases} 0 & k - k' \notin \{0, 1\} \\ P(s'|s, \boldsymbol{a})\pi_{adv}(\boldsymbol{a}|s, \boldsymbol{a}, k) & k - k' = 0 \\ \hat{P}(s'|s, \boldsymbol{a}, k) & k - k' = 1, \end{cases}$$

where $\hat{P}(s'|s, \boldsymbol{a}, k) = \sum_{\hat{\boldsymbol{a}}\sim\mathcal{A}, \hat{\boldsymbol{a}}\neq\boldsymbol{a}} P(s'|s, \hat{\boldsymbol{a}})$, $\hat{R}(s, \boldsymbol{a}, s') = \sum_{\hat{\boldsymbol{a}}\sim\mathcal{A}, \hat{\boldsymbol{a}}\neq\boldsymbol{a}} R(s, \hat{\boldsymbol{a}}, s')$.

By substituting π_{adv} with a stochastic version in Eq.(5), we have

$$\begin{aligned} &\tilde{V}_{\boldsymbol{\pi}}(\tilde{s}) \\ &= \sum_{\boldsymbol{a}\sim\mathcal{A}} \boldsymbol{\pi}(\boldsymbol{a}|\tilde{s})\pi_{adv}(\boldsymbol{a}|s, \boldsymbol{a}, k) \sum_{s'\in\mathcal{S}} P(s'|s, \boldsymbol{a})[R(s, \boldsymbol{a}, s') \\ &+ \gamma \tilde{V}_{\boldsymbol{\pi}}((s', k))] + \sum_{\boldsymbol{a}\sim\mathcal{A}} \boldsymbol{\pi}(\boldsymbol{a}|\tilde{s}) \sum_{s'\in\mathcal{S}} \sum_{\hat{\boldsymbol{a}}\neq\boldsymbol{a}} \pi_{adv}(\hat{\boldsymbol{a}}|s, \boldsymbol{a}, k) \\ &P(s'|s, \hat{\boldsymbol{a}})[\sum_{\hat{\boldsymbol{a}}'\neq\boldsymbol{a}} \pi_{adv}(\hat{\boldsymbol{a}}'|s, \boldsymbol{a}, k)R(s, \hat{\boldsymbol{a}}', s') + \\ &\quad \gamma \tilde{V}_{\boldsymbol{\pi}}((s', k - 1))]. \end{aligned} \tag{7}$$

By splitting Eq.(2) based on $\mathbb{I}(\hat{\boldsymbol{a}} \neq \boldsymbol{a})$, we would derive that:

$$\begin{aligned} &\hat{V}_{\boldsymbol{\pi}\circ\pi_{adv}}(s, k) \\ &= \sum_{\boldsymbol{a}\sim\mathcal{A}} \boldsymbol{\pi}(\boldsymbol{a}|\tilde{s})\pi_{adv}(\boldsymbol{a}|s, \boldsymbol{a}, k) \sum_{s'\in\mathcal{S}} P(s'|s, \boldsymbol{a})[R(s, \boldsymbol{a}, s') \\ &+ \gamma \hat{V}_{\boldsymbol{\pi}\circ\pi_{adv}}(s', k)] + \sum_{\boldsymbol{a}\sim\mathcal{A}} \boldsymbol{\pi}(\boldsymbol{a}|\tilde{s}) \sum_{s'\in\mathcal{S}} \sum_{\hat{\boldsymbol{a}}\neq\boldsymbol{a}} \pi_{adv}(\hat{\boldsymbol{a}}|s, \boldsymbol{a}, k) \\ &P(s'|s, \hat{\boldsymbol{a}})[R(s, \hat{\boldsymbol{a}}, s') + \gamma \hat{V}_{\boldsymbol{\pi}\circ\pi_{adv}}(s', k - 1)]. \end{aligned} \tag{8}$$

Similar to the proof in Thm.(1), we could derive that $\forall \boldsymbol{\pi}$, $k = 0$, and we have $\hat{V}_{\boldsymbol{\pi}\circ\pi_{adv}}(s, 0) = \tilde{V}_{\boldsymbol{\pi}}(\tilde{s})$, $\forall s \in \mathcal{S}$, where $\tilde{s} = (s, 0)$.

According to the basic property of expectation that $\mathbb{E}_X[f(X)g(X)] \geq \mathbb{E}_X[f(X)]\mathbb{E}_X[g(X)]$, we have $\forall \boldsymbol{\pi}$, and $\hat{V}_{\boldsymbol{\pi}\circ\pi_{adv}}(s, k) \geq \tilde{V}_{\boldsymbol{\pi}}(\tilde{s})$, $\forall s \in \mathcal{S}, k \in \{0, 1, ..., K\}$, where $\tilde{s} = (s, k)$. By optimizing $\boldsymbol{\pi}$ in Dec-POMDP $\tilde{\mathcal{M}}$, we are optimizing the lower bound of the state value function of $\boldsymbol{\pi}$ in the original LPA-Dec-POMDP.

B Additional Description of Algorithms

B.1 Description of related algorithms

As described in the main manuscript, our work mainly includes two parts: Attacker Population Generation and the whole process of ROMANCE, the detailed pseudocodes are shown in Algo. 1 and Algo. 2, respectively.

We describe the procedure of evolutionary generation of the attacker population in Algorithm 1. In each iteration, we first select n_p adversarial attackers from the archive as the current population based on their quality scores. Ego-system will interact with the attacker in the population alternately as is described in the function CollectTraj. The trajectories

Algorithm 1: Attacker Population Generation

```
Input: A joint ego-agent policy $\boldsymbol{\pi}$, archive $Arc_{adv}$,
       population size $n_p$, archive capacity $n_a$.
$P_{adv} \leftarrow$select$(Arc_{adv}, n_p)$;
for $\pi_{adv}^{\phi_j} \in P_{adv}$ do
    $\mathcal{D}_{\boldsymbol{\pi}}, \mathcal{D}_{adv}^j \leftarrow$ CollectTraj$(\boldsymbol{\pi}, \pi_{adv}^{\phi_j})$;
end
Optimize $\{\phi_j\}_{j=1}^{n_p}$ based on Eq.(6);
$Arc_{adv} \leftarrow$ Update$(Arc_{adv}, P_{adv})$;
Function Update ($Arc_{adv}$, $P_{adv}$):
    for $\pi_{adv}^{\phi_j} \in P_{adv}$ do
        for $\pi_{adv}^{\phi_i} \in Arc_{adv}$ do
            calculate dist$(i, j)$ based on Eq.(8);
        end
        if $\min_i$ dist$(i, j) \geq threshold$ then
            add$(\pi_{adv}^{\phi_j}, Arc_{adv})$;
        else
            reserve$(\pi_{adv}^{\arg\min_i \text{dist}(i,j)}, \pi_{adv}^j, Arc_{adv})$;
        end
    end
return $Arc_{adv}$
Function CollectTraj ($\boldsymbol{\pi}$,$\pi_{adv}^{\phi_j}$):
    $k \leftarrow K$;
    $\mathcal{D}_{\boldsymbol{\pi}}, \mathcal{D}_{adv}^j \leftarrow \{\}, \{\}$;
    for $t = 0$ to $T$ do
        $\{o_t^i\}_{i=1}^N = \{O(s_t, i)\}_{i=1}^N$;
        $\boldsymbol{a}_t = \boldsymbol{\pi}(\boldsymbol{\tau}_t)$;
        $\hat{i} \sim v^j(s_t, k)$;
        if $\hat{i} \neq$ null and $k > 0$ then
            $\hat{a}_t^{\hat{i}} = \arg\min_{a^{\hat{i}}} Q^i(\tau_t^{\hat{i}}, a^{\hat{i}})$;
            $\hat{\boldsymbol{a}}_t = (\hat{a}_t^{\hat{i}}, \boldsymbol{a}_t^{-\hat{i}})$;
            $k \leftarrow k - 1$;
        else
            $\hat{\boldsymbol{a}}_t = \boldsymbol{a}_t$
        end
        $s_{t+1}, r_t, done \leftarrow env.step(\hat{\boldsymbol{a}}_t)$;
        $\mathcal{D}_\pi \leftarrow \mathcal{D}_\pi \cup \{s_t, \boldsymbol{o}_t, \boldsymbol{a}_t, r_t, done\}$;
        $\mathcal{D}_{adv}^j \leftarrow \mathcal{D}_{adv}^j \cup \{s_t, \hat{i}, -r_t, done\}$;
        if done is True then
            break;
        end
    end
return $\mathcal{D}_{\boldsymbol{\pi}}$,$\mathcal{D}_{adv}^j$
```

collected are used to optimize the population and thus obtaining n_p new adversarial attackers. To keep the attackers' quality and diversity in the archive, the specialized updating mechanism in function Update is adopted.

In Algorithm 2, we present the alternating training paradigm ROMANCE, where the best response to a population of adversarial attackers are optimized based on the evolutionary generation of attackers in Algorithm 1.

Algorithm 2: ROMANCE

```
Input: Environment $\mathcal{E}$, population size $n_p$, archive
       capacity $n_a$, num of iterations $N_{gen}$.
Initialize the archive $Arc_{adv}$ with capacity $n_a$;
for $gen = 1$ to $N_{gen}$ do
    $P_{adv} \leftarrow$ select$(Arc_{adv}, n_p)$;
    for $n = 1$ to $N_{adv}$ do
        for $\pi_{adv}^{\phi_j} \in P_{adv}$ do
            $\mathcal{D}_{\boldsymbol{\pi}}, \mathcal{D}_{adv}^j \leftarrow$ CollectTraj$(\boldsymbol{\pi}^\theta, \pi_{adv}^{\phi_j})$;
        end
        Optimize $\boldsymbol{\phi}$ with $\mathcal{D}_{adv}$ based on Eq. (6);
    end
    for $n = 1$ to $N_{ego}$ do
        for $\pi_{adv}^{\phi_j} \in P_{adv}$ do
            $\mathcal{D}_{\boldsymbol{\pi}}, \mathcal{D}_{adv}^j \leftarrow$ CollectTraj$(\boldsymbol{\pi}^\theta, \pi_{adv}^{\phi_j})$;
        end
        Optimize $\theta$ with $\mathcal{D}_{\boldsymbol{\pi}}$ based on Eq. (9);
    end
    $Arc_{adv} \leftarrow$ Update$(Arc_{adv}, P_{adv})$;
end
```

Algorithm 3: RARL

```
Input: Environment $\mathcal{E}$, num of iterations $N_{gen}$.
Initialize an attacker policy $\pi_{adv}^{\phi}$ ;
for $gen = 1$ to $N_{gen}$ do
    for $n = 1$ to $N_{adv}$ do
        $\mathcal{D}_{\boldsymbol{\pi}}, \mathcal{D}_{adv}^j \leftarrow$ CollectTraj$(\boldsymbol{\pi}^\theta, \pi_{adv}^{\phi_j})$;
        Optimize $\boldsymbol{\phi}$ with $\mathcal{D}_{adv}$ based on Eq. (3);
    end
    for $n = 1$ to $N_{ego}$ do
        $\mathcal{D}_{\boldsymbol{\pi}}, \mathcal{D}_{adv}^j \leftarrow$ CollectTraj$(\boldsymbol{\pi}^\theta, \pi_{adv}^{\phi_j})$;
        Optimize $\theta$ with $\mathcal{D}_{\boldsymbol{\pi}}$ based on Eq. (9);
    end
end
```

As for the two subject baselines, RARL and RAP, we present the detailed pseudocodes in Algo. 3 and Algo.4, respectively.

C Detailed description of SMAC

SMAC (Samvelyan et al. 2019) is a combat scenario of StarCraft II unit micromanagement tasks. We consider a partial observation setting, where an agent can only see a circular area around it with a radius equal to the sight range, which is set to 9. We train the ally units with reinforcement learning algorithms to beat enemy units controlled by the built-in AI. At the beginning of each episode, allies and enemies are generated at specific regions on the map. Every agent takes action from the discrete action space at each timestep, including the following actions: no-op, move [direction], attack [enemy id], and stop. Under the control of these actions, agents can move and attack in continuous maps. MARL agents will get a global reward equal to the total damage done to enemy units

Algorithm 4: RAP

```
Input: Environment $\mathcal{E}$, population size $n_p$, num of
       iterations $N_{gen}$.
Initialize the population $P_{adv}$ with capacity $n_p$;
for $gen = 1$ to $N_{gen}$ do
    for $n = 1$ to $N_{adv}$ do
        for $\pi^{\phi_j}_{adv} \in P_{adv}$ do
            $\mathcal{D}_{\boldsymbol{\pi}}, \mathcal{D}^{j}_{adv} \leftarrow$ CollectTraj($\boldsymbol{\pi}^{\theta}, \pi^{\phi_j}_{adv}$);
        end
        Optimize $\phi$ with $\mathcal{D}_{adv}$ based on Eq. (3);
    end
    for $n = 1$ to $N_{ego}$ do
        for $\pi^{\phi_j}_{adv} \in P_{adv}$ do
            $\mathcal{D}_{\boldsymbol{\pi}}, \mathcal{D}^{j}_{adv} \leftarrow$ CollectTraj($\boldsymbol{\pi}^{\theta}, \pi^{\phi_j}_{adv}$);
        end
        Optimize $\theta$ with $\mathcal{D}_{\boldsymbol{\pi}}$ based on Eq. (9);
    end
end
```

at each timestep. Killing each enemy unit and winning the combat (killing all the enemies) will bring additional bonuses of 10 and 200, respectively. We briefly introduce the SMAC maps used in our paper in Tab. 1, and the snapshots of each map are shown in Figure 1.

Map	Ally Units	Enemy Units	Type
2s3z	2 Stalkers, 3 Zealots	2 Stalkers, 3 Zealots	Symmetric, Heterogeneous
3m	3 Marines	3 Marines	Symmetric, Homogeneous
3s_vs_3z	3 Stalkers	3 Zealots	micro-trick, kiting
8m	8 Marines	8 Marines	Symmetric, Homogeneous
MMM	1 Medivac, 2 Marauders, 7 Marines	1 Medivac, 2 Marauders, 7 Marines	Symmetric, Heterogeneous
1c3s5z	1 Colossi, 3 Stalkers, 5 Zealots	1 Colossi, 3 Stalkers, 5 Zealots	Symmetric, Heterogeneous

Table 1: Properties of 6 conducted SMAC scenarios.

D The Architecture, Infrastructure, and Hyperparameters Choices of ROMANCE

Since ROMANCE is built on top of QMIX in the main experiments (VDN and QPLEX in the Integrative Abilities part in the main manuscript), we here present specific settings, including network architectures and hyperparameters choices. The local agent network shares the same architecture with QMIX, having a GRU cell with a dimension of 64 to encode historical information and two fully connected layers to compute local Q values. Mixing networks are applied according to existing MARL methods. The adversarial attacker utilizes a multi-layer perceptron (MLP) with a hidden layer of 64 units as a victim selection function to choose the victim and force it to execute the local worst action according to the heuristic-based policy perturbation function. We adopt RMSProp as the optimizer with $\alpha = 0.99$, $\epsilon = 1 \times 10^{-5}$ for both ego-system and attackers. Specifically, the learning rate of the ego-system is set to be 4×10^{-4} for 2s3z, 3m, and 3s_vs_3z and 2×10^{-4} for others. $\delta = 5 \times 10^{-2}$ and $\lambda = 4 \times 10^{-2}$ are the parameters of reference distribution and regularized factor for SPRQ, respectively. Besides, we set a smoothing constant $b = 2 \times 10^{-2}$ over the action distribution in case of KL-divergence approaching infinity. The whole framework is trained end-to-end with collected episodic data on NVIDIA GeForce RTX 2080 Ti GPUs with 800 iterations (generations).

E Additional Experimental Results

Besides what has been presented in the main manuscript, we show the learning curve of different methods (built on QMIX) on other four maps in Fig. 2. We can observe that ROMANCE achieves the best performance under the strong adversarial attack on other maps. We can also observe that RARL even performs worse than RANDOM in map MMM and 1c3s5z. We believe that when ther exist many heterogeneous agents, the ego-system might be overfitting to a specific type of attackers easily. In Fig. 3, we present the learning curves of different methods implemented on QPLEX and VDN on map 2s3z. The curves show that ROMANCE can significantly enhance the robustness of value-based MARL algorithms when they are integrated.

In Fig. 4 we show the attackers' quality generated by different methods on all six maps. For EGA, EGA_w/o_sa and PBA that generate more than one attackers in a single run, we take the mean value of five best attackers as the quality of each run. The superiority of EGA over other methods in all six maps demonstrates its effectiveness.

In Tab. 2-4, we investigate how different types of hyperparameters affect the performance of ROMANCE. Respectively, we alter the size of the archive, size of the population, and the number of attack during training and then test the ego-system under three settings.

Archive Size. The archive is used to record the attackers with high-performance and diverse behaviors generated so far. It is crucial for ROMANCE, as new attackers are generated based on those selected from the archive. Too small size results in decrease in diversity, but too large size makes it inefficient to select high-performing attackers.

As shown in Tab. 2, we find that the slightly bigger archive size promotes the improvement of robustness as its larger capacity allows for maintaining more high-performing and diverse attackers. The performance might also decrease if it goes too large, with a decrease in efficiency, as explained above.

Population Size. Tab. 3 describes the robustness of ROMANCE when the size of the population changes. The population size refers to the number of attackers the ego-system will encounter in one generation. A larger population size promotes the robustness of the ego-system in the final stage, but might also generate attackers with similar behavior, thus harming the efficiency. Despite the same archive size, train-

(a) 2s3z

(b) 3m

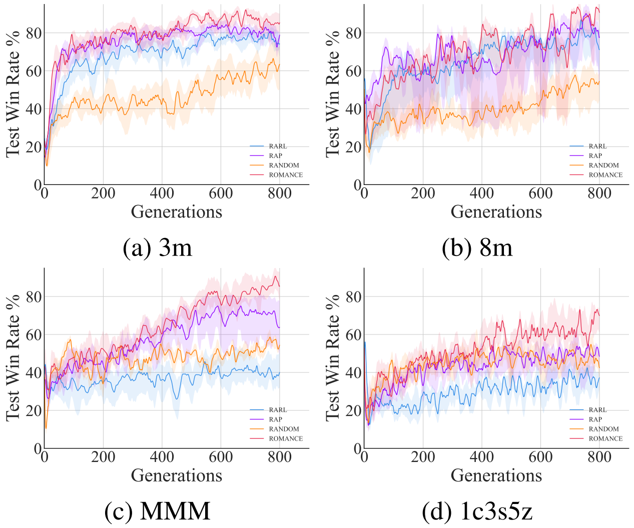
(c) 3s_vs_3z

(d) 8m

(e) MMM

(f) 1c3s5z

Figure 1: Snapshots of our selected StarCraft II scenarios.



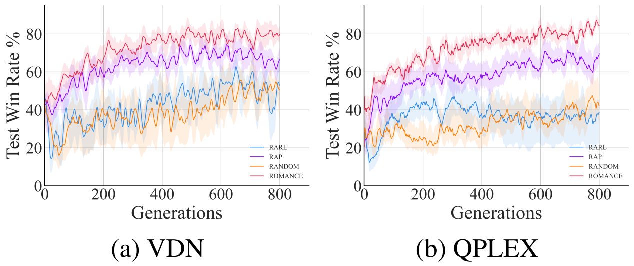
(a) 3m

(b) 8m

(c) MMM

(d) 1c3s5z

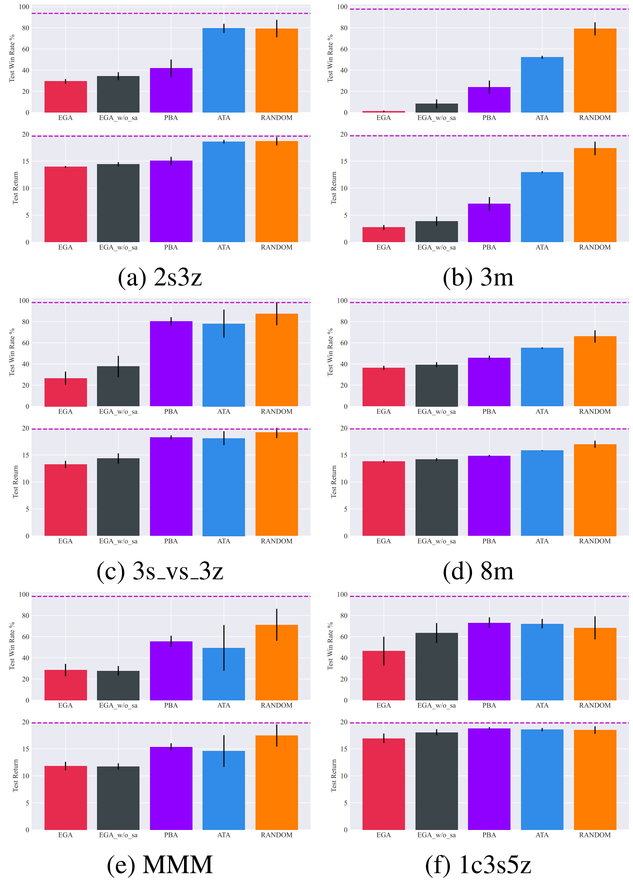
Figure 2: Average test win rates on four more maps.



(a) VDN

(b) QPLEX

Figure 3: Average test win rates of VDN and QPLEX on map 2s3z, the result of QMIX is shown in the main manuscript.



(a) 2s3z

(b) 3m

(c) 3s_vs_3z

(d) 8m

(e) MMM

(f) 1c3s5z

Figure 4: The quality of attackers generated by different methods on more six maps.

ing the best response to a small size population with only 2 or 3 attackers tends to result in overfitting, and thus weakens the generalization ability against diverse attackers. Results in Tab. 3 show that the default population size 4 is appropriate in the map 2s3z.

Training Attack Budget. We also present ROMANCE's

Archive Size	Natural	Random Attack	EGA
11	94.0 ± 9.06	83.7 ± 10.3	71.2 ± 12.1
13	98.5 ± 0.68	88.9 ± 1.59	76.5 ± 4.16
15	97.9 ± 1.34	89.1 ± 1.97	81.6 ± 0.84
17	98.1 ± 0.62	88.2 ± 0.94	83.6 ± 3.02
19	96.8 ± 1.14	86.6 ± 1.06	78.8 ± 7.40

Table 2: Average test win rate of ROMANCE on map 2s3z when archive size changes.

Population Size	Natural	Random Attack	EGA
2	94.7 ± 4.56	83.4 ± 7.40	66.2 ± 9.44
3	92.6 ± 5.47	82.0 ± 9.78	68.8 ± 12.9
4	97.9 ± 1.34	89.1 ± 1.97	81.6 ± 0.84
5	98.4 ± 0.83	88.7 ± 2.74	79.2 ± 5.25
6	98.8 ± 0.45	89.6 ± 1.07	80.9 ± 1.34

Table 3: Average test win rate of ROMANCE on map 2s3z when population size changes.

average test win rate under different budgetary training attack numbers in Tab. 4. When the training budget is less than the testing one, the generalization ability of our method guarantees their decent performance. However, it is still inferior to the ego-system whose training budgetary attack number is in accordance with the testing one. Excess training attack numbers might also cause performance degradation, as too strong attackers bring a conservative ego-system.

Attack Num	Natural	Random Attack	EGA
6	97.3 ± 1.42	87.2 ± 2.42	71.6 ± 5.59
7	98.0 ± 1.05	87.2 ± 3.28	76.2 ± 2.80
8	97.9 ± 1.34	89.1 ± 1.97	81.6 ± 0.84
9	97.8 ± 1.38	88.2 ± 3.29	84.4 ± 2.68
10	98.3 ± 0.86	87.4 ± 2.91	80.1 ± 1.99

Table 4: Average test win rate of ROMANCE on map 2s3z when attacker number changes.

References

Samvelyan, M.; Rashid, T.; Schroeder de Witt, C.; Farquhar, G.; Nardelli, N.; Rudner, T. G.; Hung, C.-M.; Torr, P. H.; Foerster, J.; and Whiteson, S. 2019. The StarCraft Multi-Agent Challenge. In *AAMAS*, 2186–2188.