A. Individual-Global-Optimal

Definition 1 (IGO). *For an optimal joint policy $\pi^*_{jt}(\boldsymbol{u}|\boldsymbol{\tau}) : \boldsymbol{\mathcal{T}} \times \boldsymbol{\mathcal{U}} \rightarrow [0,1]$, where $\boldsymbol{\tau} \in \boldsymbol{\mathcal{T}}$ is a joint trajectory, if there exist individual optimal policies $[\pi^*_i(u_i|\tau_i) : \mathcal{T} \times \mathcal{U} \rightarrow [0,1]]^N_{i=1}$, such that the following holds:*

$$\pi^*_{jt}(\boldsymbol{u}|\boldsymbol{\tau}) = \prod_{i=1}^{N} \pi^*_i(u_i|\tau_i),$$

*then, we say that $[\pi_i]$ satisfy **IGO** for π_{jt} under $\boldsymbol{\tau}$.*

A.1. Policy Improvement under IGO

Under the IGO constraint, if individual policies are locally updated towards optimal, the joint policy improvement will be obtained. Formally, it can be proved by using the distance measure of KL-divergence, as shown bellowing:

$$\begin{aligned}
D_{\text{KL}}(\pi_{\text{jt}}(\boldsymbol{u}|\boldsymbol{\tau})||\pi^*_{\text{jt}}(\boldsymbol{u}|\boldsymbol{\tau})) &= \int_{\boldsymbol{\mathcal{U}}} \pi_{\text{jt}}(\boldsymbol{u}|\boldsymbol{\tau}) \log \frac{\pi_{\text{jt}}(\boldsymbol{u}|\boldsymbol{\tau})}{\pi^*_{\text{jt}}(\boldsymbol{u}|\boldsymbol{\tau})} d\boldsymbol{u} \\
&= \int_{\boldsymbol{\mathcal{U}}} \prod_{i=1}^{N} \pi_i(u_i|\tau_i) \sum_{i=1}^{N} \log \frac{\pi_i(u_i|\tau_i)}{\pi^*_i(u_i|\tau_i)} d\boldsymbol{u} \\
&= \sum_{i=1}^{N} \int_{\mathcal{U}_i} \int_{\mathcal{U}_{-i}} \pi_i(u_i|\tau_i)\pi_{-i}(u_{-i}|\tau_{-i}) \log \frac{\pi_i(u_i|\tau_i)}{\pi^*_i(u_i|\tau_i)} du_i du_{-i} \\
&= \sum_{i=1}^{N} D_{\text{KL}}(\pi_i(u_i|\tau_i)||\pi^*_i(u_i|\tau_i)) \int_{\mathcal{U}_{-i}} \pi_{-i}(u_{-i}|\tau_{-i}) du_{-i} \\
&= \sum_{i=1}^{N} D_{\text{KL}}(\pi_i(u_i|\tau_i)||\pi^*_i(u_i|\tau_i)),
\end{aligned}$$

where π_{-i} inducing action u_{-i} represents the joint policy without agent i. This implies that, under the IGO constraint, the distance of joint policy to the optimal joint policy can be shortened by the improvement of individual policies.

A.2. Relationship between IGO and IGM

In fact, IGO is an extension of existing Individual-Global-Maximum (IGM) (Son et al., 2019) constraint, which guarantees the consistency between individual optimal actions and optimal joint action, *i.e.*,

Definition 2 (IGM). *For a joint action-value function $Q_{jt} : \boldsymbol{\mathcal{T}} \times \boldsymbol{\mathcal{U}} \rightarrow \mathbb{R}$, where $\boldsymbol{\tau} \in \boldsymbol{\mathcal{T}}$ is a joint trajectory, if there exist individual action-value functions $[Q_i : \mathcal{T} \times \mathcal{U} \rightarrow \mathbb{R}]^N_{i=1}$, such that the following holds*

$$\arg\max_{\boldsymbol{u}} Q_{jt} = \big(\arg\max_{u_1} Q_1, \ldots, \arg\max_{u_N} Q_N\big).$$

*then, we say that $[Q_i]$ satisfy **IGM** for Q_{jt} under $\boldsymbol{\tau}$. In this case, we also say that $Q_{jt}(\boldsymbol{\tau}, \mathbf{u})$ is factorized by $[Q_i(\tau_i, u_i)]$, or that $[Q_i]$ are factors of Q_{jt}.*

When the joint policy and individual policy are as following:

$$\pi_{\text{jt}}(\boldsymbol{u}|\boldsymbol{\tau}) = \begin{cases} 1, & \boldsymbol{u} = \arg\max\limits_{\boldsymbol{u}} Q_{\text{jt}}(\boldsymbol{\tau}, \boldsymbol{u}) \\ 0, & \text{otherwise} \end{cases}$$

$$\pi_i(u_i|\tau_i) = \begin{cases} 1, & u_i = \arg\max\limits_{u_i} Q_i(\tau_i, u_i) \\ 0, & \text{otherwise} \end{cases},$$

the joint action $\boldsymbol{u} = \arg\max\limits_{\boldsymbol{u}} Q_{\text{jt}}(\boldsymbol{\tau}, \boldsymbol{u})$ if and only if the individual action $u_i = \arg\max\limits_{u_i} Q_i(\tau_i, u_i)$ for any agent i. Therefore, we say IGM can be seen as a special case of IGO if we specialize in the greedy policy, or IGO is an extension of IGM.

B. Factorized Soft Policy Iteration

B.1. Joint Soft Policy Evaluation

Lemma 1 (Joint Soft Policy Evaluation). *Consider the soft Bellman operator $\Gamma_{\pi_{jt}}$ and a mapping $Q^0_{jt} : \boldsymbol{\mathcal{T}} \times \boldsymbol{\mathcal{U}} \to \mathbb{R}$ with $|\boldsymbol{\mathcal{U}}| < \infty$, and define $Q^{k+1}_{jt} = \Gamma_{\pi_{jt}} Q^k_{jt}$. Then, the sequence Q^k_{jt} will converge to the joint soft-Q-function of π_{jt} as $k \to \infty$.*

Proof. First, define the entropy augmented reward as:

$$r_{\pi_{\text{jt}}}(\boldsymbol{\tau}_t, \boldsymbol{u}_t) := r(\boldsymbol{\tau}_t, \boldsymbol{u}_t) + \mathbb{E}_{\boldsymbol{\tau}_{t+1}}[\mathcal{H}(\pi_{\text{jt}}(\cdot|\boldsymbol{\tau}_{t+1}))].$$

Then, rewrite the update rule as:

$$Q_{\text{jt}}(\boldsymbol{\tau}_t, \boldsymbol{u}_t) \leftarrow r_{\pi_{\text{jt}}}(\boldsymbol{\tau}_t, \boldsymbol{u}_t) + \gamma \mathbb{E}_{\boldsymbol{\tau}_{t+1}, \boldsymbol{u}_{t+1} \sim \pi_{\text{jt}}}[Q_{\text{jt}}(\boldsymbol{\tau}_{t+1}, \boldsymbol{u}_{t+1})].$$

Last, apply the standard convergence results for policy evaluation (Sutton & Barto, 2018). □

B.2. Individual Soft Policy Improvement

We restrict the individual soft policy π_i of each agent i to some set of policies Π_i and update the individual soft policy according to:

$$\begin{aligned}\pi_i^{\text{new}} &= \underset{\pi_i' \in \Pi_i}{\arg\min} D_{\text{KL}}\left(\pi_i'(\cdot|\tau_i) \middle|\middle| \exp\left(\frac{1}{\alpha_i}\left(Q_i^{\pi_i^{\text{old}}}(\tau_i, \cdot) - V_i^{\pi_i^{\text{old}}}(\tau_i)\right)\right)\right) \\ &= \underset{\pi_i' \in \Pi_i}{\arg\min} J_{\pi_i^{\text{old}}}(\pi_i'(\cdot|\tau_i)).\end{aligned} \tag{12}$$

Lemma 2 (Individual Soft Policy Improvement). *Let $\pi_i^{old} \in \Pi_i$ and π_i^{new} be the optimizer of the minimization problem defined in (8). Then, we have $Q_{jt}^{\pi_{jt}^{new}}(\boldsymbol{\tau}_t, \boldsymbol{u}_t) \geq Q_{jt}^{\pi_{jt}^{old}}(\boldsymbol{\tau}_t, \boldsymbol{u}_t)$ for all $(\boldsymbol{\tau}_t, \boldsymbol{u}_t) \in \boldsymbol{\mathcal{T}} \times \boldsymbol{\mathcal{U}}$ with $|\boldsymbol{\mathcal{U}}| < \infty$, where $\pi_{jt}^{old} = \prod_{i=1}^N \pi_i^{old}$ and $\pi_{jt}^{new} = \prod_{i=1}^N \pi_i^{new}$.*

Proof. Let $Q_i^{\pi_i^{\text{old}}}$ and $V_i^{\pi_i^{\text{old}}}$ be the corresponding soft state-action value and soft state value of individual policy π_i^{old}. First, considering that $J_{\pi_i^{\text{old}}}(\pi_i^{\text{new}}(\cdot|\tau_i)) \leq J_{\pi_i^{\text{old}}}(\pi_i^{\text{old}}(\cdot|\tau_i))$, we have

$$\mathbb{E}_{u_i \sim \pi_i^{\text{new}}}[\alpha_i \log \pi_i^{\text{new}}(u_i|\tau_i) - Q_i^{\pi_i^{\text{old}}}(\tau_i, u_i) + V_i^{\pi_i^{\text{old}}}(\tau_i)] \leq \mathbb{E}_{u_i \sim \pi_i^{\text{old}}}[\alpha_i \log \pi_i^{\text{old}}(u_i|\tau_i) - Q_i^{\pi_i^{\text{old}}}(\tau_i, u_i) + V_i^{\pi_i^{\text{old}}}(\tau_i)]. \tag{13}$$

Since $\log$ partition function $V_i^{\pi_i^{\text{old}}}$ depends only on the state, where

$$V_i^{\pi_i^{\text{old}}}(\tau_i) = \mathbb{E}_{u_i \sim \pi_i^{\text{old}}}[Q_i^{\pi_i^{\text{old}}}(\tau_i, u_i) - \alpha_i \log \pi_i^{\text{old}}(u_i|\tau_i)],$$

the inequality (13) can be reduced to

$$\mathbb{E}_{u_i \sim \pi_i^{\text{new}}}[Q_i^{\pi_i^{\text{old}}}(\tau_i, u_i) - \alpha_i \log \pi_i^{\text{new}}(u_i|\tau_i)] \geq V_i^{\pi_i^{\text{old}}}(\tau_i). \tag{14}$$

Then, according to Section 4.1, for π_i^{old}, the individual soft Q-values and the joint soft Q-value satisfy:

$$\begin{aligned} Q_{\text{jt}}(\boldsymbol{\tau}, \boldsymbol{u}) &= \sum_{i=1}^N \frac{\alpha}{\alpha_i}\left[Q_i(\tau_i, u_i) - V_i(\tau_i)\right] + V_{\text{jt}}(\boldsymbol{\tau}), \\ \text{where } V_{\text{jt}}(\boldsymbol{\tau}) &:= \alpha \int_{\boldsymbol{\mathcal{U}}} \exp(\frac{1}{\alpha} Q_{\text{jt}}(\boldsymbol{\tau}, \boldsymbol{u})) d\boldsymbol{u} \\ V_i(\tau_i) &:= \alpha_i \log \int_{\mathcal{U}_i} \exp(\frac{1}{\alpha_i} Q_i(\tau_i, u)) du. \end{aligned} \tag{15}$$

Considering $\pi_{\text{jt}}^{\text{new}} = \prod_{i=1}^{n}\pi_i^{\text{new}}$, we have:

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{u}\sim\pi_{\text{jt}}^{\text{new}}}\left[Q_{\text{jt}}^{\pi_{\text{jt}}^{\text{old}}}(\boldsymbol{\tau},\boldsymbol{u}) - \alpha\log\pi_{\text{jt}}^{\text{new}}(\boldsymbol{u}|\boldsymbol{\tau})\right]\\
&= \mathbb{E}_{\boldsymbol{u}\sim\pi_{\text{jt}}^{\text{new}}}\left[\sum_{i=1}^{N}\frac{\alpha}{\alpha_i}\left[Q_i^{\pi_i^{\text{old}}}(\tau_i,u_i) - V_i^{\pi_i^{\text{old}}}(\tau_i)\right] + V_{\text{jt}}^{\pi_{\text{jt}}^{\text{old}}}(\boldsymbol{\tau}) - \alpha\log\pi_{\text{jt}}^{\text{new}}(\boldsymbol{u}|\boldsymbol{\tau})\right] \qquad \text{(plugging in (15))}\\
&= \sum_{i=1}^{N}\mathbb{E}_{u_i\sim\pi_i^{\text{new}}}\left[\frac{\alpha}{\alpha_i}\left(Q_i^{\pi_i^{\text{old}}}(\tau_i,u_i) - V_i^{\pi_i^{\text{old}}}(\tau_i) - \alpha_i\log\pi_i^{\text{new}}(u_i|\tau_i)\right)\right] + V_{\text{jt}}^{\pi_{\text{jt}}^{\text{old}}}(\boldsymbol{\tau})\\
&\geq V_{\text{jt}}^{\pi_{\text{jt}}^{\text{old}}}(\boldsymbol{\tau}) \qquad \text{(plugging in (14)).}
\end{aligned} \tag{16}$$

Last, considering the soft Bellman equation, the following holds:

$$\begin{aligned}
Q_{\text{jt}}^{\pi_{\text{jt}}^{\text{old}}}(\boldsymbol{\tau}_t,\boldsymbol{u}_t) &= r_t + \gamma\mathbb{E}_{\boldsymbol{\tau}_{t+1}}[V^{\pi_{\text{jt}}^{\text{old}}}(\boldsymbol{\tau}_{t+1})]\\
&\leq r_t + \gamma\mathbb{E}_{\boldsymbol{\tau}_{t+1}}\left[\mathbb{E}_{\boldsymbol{u}_{t+1}\sim\pi_{\text{jt}}^{\text{new}}}\left[Q_{\text{jt}}^{\pi_{\text{jt}}^{\text{old}}}(\boldsymbol{\tau}_{t+1},\boldsymbol{u}_{t+1}) - \alpha\log\pi_{\text{jt}}^{\text{new}}(\boldsymbol{u}_{t+1}|\boldsymbol{\tau}_{t+1})\right]\right] \qquad \text{(plugging in (16))}\\
&\;\;\vdots\\
&\leq Q_{\text{jt}}^{\pi_{\text{jt}}^{\text{new}}}(\boldsymbol{\tau}_t,\boldsymbol{u}_t),
\end{aligned}$$

where we have repeatedly expanded $Q_{\text{jt}}^{\pi_{\text{jt}}^{\text{old}}}$ on the RHS by applying the soft Bellman equation and the bound in (16). Convergence to $Q_{\text{jt}}^{\pi_{\text{jt}}^{\text{new}}}$ follows from Lemma 1. □

B.3. Factorized Soft Policy Iteration

Factorized soft policy iteration alternates between joint soft policy evaluation and individual soft policy improvement, and provably converges to the global optimum among the policies in $[\Pi_i]_{i=1}^{N}$.

Theorem 1 (Factorized Soft Policy Iteration). *Considering joint soft policy can be factorized as $\pi_{jt} = \prod_{i=1}^{N}\pi_i$, repeated application of joint soft policy evaluation and individual soft policy improvement from $\pi_i \in \Pi_i, \forall i \in \mathcal{N}$ converges to a policy π_{jt}^* such that $Q_{jt}^{\pi_{jt}^*}(\boldsymbol{\tau},\boldsymbol{u}) \geq Q_{jt}^{\pi_{jt}}(\boldsymbol{\tau},\boldsymbol{u})$ for all $[\pi_i \in \Pi_i]_{i=1}^{N}$ and $(\boldsymbol{\tau},\boldsymbol{u}) \in \boldsymbol{\mathcal{T}} \times \boldsymbol{\mathcal{U}}$, assuming $|\boldsymbol{\mathcal{U}}| < \infty$.*

Proof. Let π_{jt}^{k} be the soft joint policy at iteration k.

First, by Lemma 2, the sequence $Q_{\text{jt}}^{\pi_{\text{jt}}^{k}}$ is monotonically increases. Since $Q_{\text{jt}}^{\pi_{\text{jt}}}$ is bounded above for all $\pi_{\text{jt}} = \prod_{i=1}^{N}\pi_i$, where $[\pi_i \in \Pi_i]_{i=1}^{N}$, that is, both the reward and entropy are bounded, the sequence converges to some π_{jt}^*.

Then, at convergence, it must be the case that

$$J_{\pi_{\text{jt}}^*}(\pi_{\text{jt}}^*(\cdot|\boldsymbol{\tau})) \leq J_{\pi_{\text{jt}}^*}(\pi_{\text{jt}}(\cdot|\boldsymbol{\tau})), \forall\pi_{\text{jt}} \neq \pi_{\text{jt}}^*.$$

Using the same iterative argument as in the proof of Lemma 2, we get $Q_{\text{jt}}^{\pi_{\text{jt}}^*}(\boldsymbol{\tau},\boldsymbol{u}) > Q_{\text{jt}}^{\pi_{\text{jt}}}(\boldsymbol{\tau},\boldsymbol{u})$ for all $(\boldsymbol{\tau},\boldsymbol{u}) \in \boldsymbol{\mathcal{T}} \times \boldsymbol{\mathcal{U}}$. That is, the soft value of any other policy π_{jt} is lower than that of converged policy π_{jt}^*. Therefore, π_{jt}^* is optimal in $[\Pi_i]_{i=1}^{N}$. □

C. Temperature Adjustment

In experiments, unless otherwise specified, the team temperature α is adjusted through automating entropy adjustment in Haarnoja et al. (2018b), *i.e.*,

$$J_\alpha = \mathbb{E}_{\boldsymbol{u}\sim\pi_{\text{jt}}}[-\alpha\log\pi_{\text{jt}}(\boldsymbol{u}|\boldsymbol{\tau}) - \alpha\bar{\mathcal{H}}],$$

where $\bar{\mathcal{H}}$ is the target entropy.

For the adjustment of individual temperature parameter α_i, as each agent's specific contribution to the team is unknown, we introduce a weight network $\lambda^{\Psi}(\boldsymbol{\tau}, \boldsymbol{u})$, parameterized by Ψ to learn $[\alpha/\alpha_i]_{i=1}^{N}$, *i.e.*,

$$\alpha_i = \alpha \frac{\mathbb{E}_{u_i \sim \pi_i}[Q_i(\tau_i, u_i) - V_i(\tau_i)]}{\mathbb{E}_{\boldsymbol{u} \sim \pi_{\text{jt}}}[\lambda_i^{\Psi}(\boldsymbol{\tau}, \boldsymbol{u})\big(Q_i(\tau_i, u_i) - V_i(\tau_i)\big)]}. \tag{17}$$

In this case, (15) can be rewritten as

$$Q_{\text{jt}}(\boldsymbol{\tau}, \boldsymbol{u}) = \sum_{i=1}^{N} \lambda_i(\boldsymbol{\tau}, \boldsymbol{u})\big[Q_i(\tau_i, u_i) - V_i(\tau_i)\big] + V_{\text{jt}}(\boldsymbol{\tau}), \tag{18}$$

where $V_{\text{jt}}(\boldsymbol{\tau})$ and $V_i(\tau_i)$ are the same as in (15).

Next, we will analyze the convergence of individual policy with λ_i. Considering that α_i is related only to $\boldsymbol{\tau}$ in (17), the inequality (13) and (14) still holds since the expectation in (13) and (14) is only related to actions. Considering $\pi_{\text{jt}}^{\text{new}} = \prod_{i=1}^{N} \pi_i^{\text{new}}$, we have:

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{u} \sim \pi_{\text{jt}}^{\text{new}}}\left[Q_{\text{jt}}^{\pi_{\text{jt}}^{\text{old}}}(\boldsymbol{\tau}, \boldsymbol{u}) - \alpha \log \pi_{\text{jt}}^{\text{new}}(\boldsymbol{u}|\boldsymbol{\tau})\right] \\
&= \mathbb{E}_{\boldsymbol{u} \sim \pi_{\text{jt}}^{\text{new}}}\left[\sum_{i=1}^{N} \lambda_i(\boldsymbol{\tau}, \boldsymbol{u})\big[Q_i^{\pi_i^{\text{old}}}(\tau_i, u_i) - V_i^{\pi_i^{\text{old}}}(\tau_i)\big] + V_{\text{jt}}^{\pi_{\text{jt}}^{\text{old}}}(\boldsymbol{\tau}) - \alpha \log \pi_{\text{jt}}^{\text{new}}(\boldsymbol{u}|\boldsymbol{\tau})\right] \qquad \text{(plugging (18))} \\
&= \mathbb{E}_{\boldsymbol{u} \sim \pi_{\text{jt}}^{\text{new}}}\left[\sum_{i=1}^{N} \lambda_i(\boldsymbol{\tau}, \boldsymbol{u})\big[Q_i^{\pi_i^{\text{old}}}(\tau_i, u_i) - V_i^{\pi_i^{\text{old}}}(\tau_i)\big] + V_{\text{jt}}^{\pi_{\text{jt}}^{\text{old}}}(\boldsymbol{\tau}) - \sum_{i=1}^{N} \alpha \log \pi_i^{\text{new}}(u_i|\tau_i)\right] \\
&= \sum_{i=1}^{N} \mathbb{E}_{u_i \sim \pi_i^{\text{new}}}\left[\frac{\alpha}{\alpha_i}\Big(Q_i^{\pi_i^{\text{old}}}(\tau_i, u_i) - V_i^{\pi_i^{\text{old}}}(\tau_i) - \alpha_i \log \pi_i^{\text{new}}(u_i|\tau_i)\Big)\right] + V_{\text{jt}}^{\pi_{\text{jt}}^{\text{old}}}(\boldsymbol{\tau}) \qquad \text{(plugging (17))} \\
&\geq V_{\text{jt}}^{\pi_{\text{jt}}^{\text{old}}}(\boldsymbol{\tau}) \qquad \text{(plugging (14))}.
\end{aligned} \tag{19}$$

Therefore, $\lambda_i^{\Psi}(\boldsymbol{\tau}, \boldsymbol{u})$ dose not violate the convergence of the factorized soft policy iteration theorem.

The objective of α_i can be written as

$$J(\alpha_i) = \mathbb{E}_{\boldsymbol{u} \sim \pi_{\text{jt}}}[\alpha_i - \alpha/\lambda_i(\boldsymbol{\tau}, \boldsymbol{u})]^2$$

In practice, we can also adjust α_i smoothly, *i.e.*,

$$\alpha_i \leftarrow (1 - \epsilon) \cdot \alpha_i + \epsilon \cdot \alpha \frac{\mathbb{E}_{u_i \sim \pi_i}[Q_i(\tau_i, u_i) - V_i(\tau_i)]}{\mathbb{E}_{\boldsymbol{u} \sim \pi_{\text{jt}}}[\lambda_i(\boldsymbol{\tau}, \boldsymbol{u})(Q_i(\tau_i, u_i) - V_i(\tau_i))]},$$

where $\epsilon = 1 \times 10^{-4}$.

D. Experiment Settings and Implementation Details

In the well-known matrix game (Son et al., 2019) and differential game (Wei et al., 2018), FOP can converge to the global optimum for both discrete and continuous action space. We also evaluate FOP on a set of StarCraft II micromanagement tasks (Samvelyan et al., 2019), and show that FOP substantially outperforms existing decomposed value-based and actor-critic methods.

D.1. Matrix Game and Differential Game

In matrix game and differential game, we use a replay buffer of size 10^6. The learning rate of all experiments are 3×10^{-4}. All the networks in FOP consist of two hidden layers of 64 units with ReLU non-linearity. The last layer of the weight network has absolute activation while that of other networks has no activation function.

The decomposed actor-critic methods including DOP (Wang et al., 2020) and FacMADDPG (de Witt et al., 2020) are also evaluated in both the matrix and differential game for comparison. The network producing weight k and bias b of DOP

Table 3. The non-monotonic cooperative matrix game. Boldface means the optimal action from individual policies.

(a) VDN: Q_1, Q_2, Q_{jt}

Q_2 Q_1	-6.7(A)	**-0.2(B)**	-0.8(C)
-6.5(A)	-13.2	-6.7	-7.3
-0.2(B)	-6.9	**-0.4**	-1.0
-0.7(C)	-7.4	-0.9	-1.5

(b) QMIX: Q_1, Q_2, Q_{jt}

Q_2 Q_1	-5.6(A)	0.1(B)	**0.0(C)**
-6.6(A)	-8.1	-8.1	-8.1
0.2(B)	-8.1	0.0	0.0
0.1(C)	-8.1	0.0	**0.0**

(c) QTRAN-alt: Q_1, Q_2, Q_{jt}

u_2 u_1	**3.3(A)**	0.1(B)	0.1(C)
4.7(A)	**8.0**	-12.0	-12.0
-0.1(B)	-12.0	0.0	0.0
-0.1(C)	-12.0	0.0	0.0

(d) QPLEX: Q_1, Q_2, Q_{jt}

Q_2 Q_1	**5.5(A)**	0.7(B)	-0.1(C)
2.4(A)	**8.0**	-12.1	-12.1
0.1(B)	-12.2	0.0	0.0
0.2(C)	-12.2	0.0	0.0

consists of two hidden layers of 64 units with ReLU non-linearity. The mixing network in FacMADDPG consists of a single hidden layer of 32 units, utilizing an ELU non-linearity. The hyper-networks consist of a feed-forward network with a single hidden layer of 64 units with the ReLU non-linearity.

Matrix Game

In the matrix game, the team temperature parameter α is adjusted by linearly annealing from 1 to 0.01 over $10k$ learning steps. As shown in Section 5.1, FOP achieves the optimal, while FacMADDPG and DOP fail into the sub-optimal.

We additionally evaluate the performance of decomposed value-based methods including VDN (Sunehag et al., 2018), QMIX (Rashid et al., 2018), QTRAN (Son et al., 2019), and QPLEX (Wang et al., 2021), with linearly decayed ϵ-greedy from 1 to 0.01 over $10k$ learning steps. Results are shown in Table 3. Obviously, FOP, QPLEX, and QTRAN are the only three algorithms that can successfully converge to the optimal, and FOP is the *first* decomposed actor-critic method.

Since the state and action space of the matrix game is quite small, each state-action pair can be easily explored. Thus, the learning mainly replies on the expressive ability. Since the decomposition structure of the centralized critic in FacMADDPG and DOP is QMIX and Qatten, respectively, which has limited expressive ability (Wang et al., 2021), they cannot find the optimal solution.

Differential Game

We also include the classic multi-agent actor-critic methods, MADDPG (Lowe et al., 2017) and MAAC (Iqbal & Sha, 2019), as the baselines for comparison. We show the learning paths where the scattered red dots are the exploration trails before convergence in Figure 2. By comparing the Figure 2d with Figure 2e, we can see that MAAC with the maximum-entropy objective has a wider range of exploration trails than MADDPG. More interestingly, by comparing the Figure 2f with Figure 2d, we can see that FacMADDPG (de Witt et al., 2020) also has a wider range of exploration trails than MADDPG.

DOP (Wang et al., 2020) analyzes that classic multi-agent actor-critic methods suffer from the exploration or sub-optimality of other agents' policies, causing a large variance of policy gradients. Considering that both MAAC and FOP utilize the maximum-entropy reinforcement learning, MAAC updates individual policies based on the centralized critic while FOP is based on the decomposed critic. Thus, we show the gradient variance of MAAC and FOP in the differential game. From Figure 5, we can see that the variance of FOP is smaller than MAAC, and all the decomposed actor-critic methods have a smaller gradient than the classic multi-agent actor-critic methods.

Although the decomposed critic can reduce the gradient variance, both FacMADDPG and DOP cannot converge to the global optimum. In both the matrix game and the differential game, we show that FacMADDPG and DOP cannot estimate the centralized critic accurately. We claim that the imprecise estimate may mislead individual policies. However, we cannot conclude that an accurate estimate of the centralized critic is a necessary and sufficient condition for the global optimal convergence of individual policies since the MADDPG and MAAC with the accurate estimate cannot converge to the global optimum. The pathology of finding a sub-optimal solution is also called *relative overgeneralization*, which is discussed in the next.

D.2. Relative Overgeneralization

The matrix game and the differential game are dramatically challenging since they face the game-theoretic pathology called ***relative overgeneralization*** (RO): during exploration, other agents act randomly, and punishment caused by uncooperative agents may outweigh rewards that would be achievable with coordinated actions, leading to policies sub-optimal (Wei & Luke, 2016; Castellini et al., 2019).

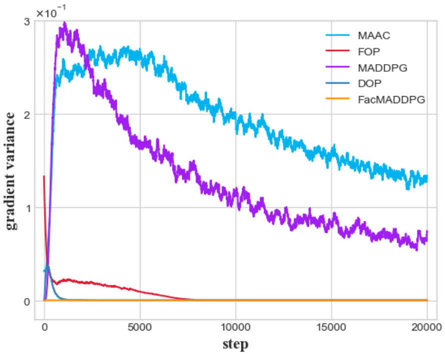

Figure 5. Gradient variance in the differential game.

Relative overgeneralization occurs when a sub-optimal solution in the joint space of actions is preferred over a global optimal solution. For instance, consider the matrix game in Section 5.1. During the random exploration, the expected return of action A (related to the global optimal solution (A, A)) is $-16/3$ while the expected return of action B and C is $-12/3$. Thus, the sub-optimal solution is more attractive than the global optimal solution. Similarly, in the differential game, the sub-optimal solution is more attractive than the global optimal solution. Thus, even if the classic multi-agent actor-critic including MADDPG (Lowe et al., 2017) and MAAC (Iqbal & Sha, 2019) estimate the centralized critic accurately, they never converge to the global optimum.

Considering that RO occurs when the largest average return weighted by exploration comes from sub-optimal, the policy's exploration ability makes sense to solve the RO. MAVEN (Mahajan et al., 2019) introduces a hierarchical model to coordinate diverse ϵ explorations among agents. In this way, MAVEN helps QMIX find the global optimal solution in matrix game (Mahajan et al., 2019). Unfortunately, such an approach limits to discrete actions due to the intractability of latent variables.

Some studies utilize the Boltzmann exploration policy for a more exploration ability than ϵ-greedy policy (Wei & Luke, 2016). FMQ (Rashid et al., 2020) with Boltzmann exploration policy in Q-learning converges almost always to the global optimum in the nonmonotonic matrix game. Using the Boltzmann policy with the maximum-entropy objective, MASQL (Wei et al., 2018) improves the global convergence of MADDPG in the differential game from 0% to 72% of the time, which shows that the maximum-entropy objective is helpful to solve the dramatic challenge, *i.e.*, relative overgeneralization.

Therefore, in this paper, we chose to factorize the maximum-entropy MARL under the IGO constraint. Empirical experiments show that our proposed method solves the relative overgeneralization problem in the two games for discrete and continuous action spaces.

D.3. Ablation Study

Castellini et al. (2019) illustrates that the learning outcomes of MASQL are extremely sensitive to the way of temperature annealing, *i.e.*, when and how to anneal the temperature to a small value during training is non-trivial. Therefore, we launched ablation experiments to explore the sensitivity of FOP to the temperature parameter.

Since the state-action space is small in the matrix game, we utilized the simple approach to adjust, *i.e.*, linearly anneal α form 1 to different $\hat{\alpha}$ over $10k$ steps. Results are shown in Table 2. We can see that, FOP can always find the optimal solution, even if the policy is not greedy ($\hat{\alpha}$ is not 0).

In the differential game, we utilized the different annealing approaches and different annealing rates. Learning curves are shown in Figure 3a. The annealing approaches are as:

$$\begin{array}{ll}
\text{(purple line)} & \log\alpha \longleftarrow \log\alpha - 1\times 10^{-3} \\
\text{(red line)} & \log\alpha \longleftarrow \log\alpha - 5\times 10^{-4} \\
\text{(light blue line)} & \alpha \longleftarrow \alpha - 1\times 10^{-4} \\
\text{(orange line)} & \alpha \longleftarrow \alpha - 5\times 10^{-5} \\
\text{(dark blue line)} & \text{Automating Entropy Adjustment.}
\end{array}$$

Table 4. The brief introduction of three typical scenarios in SMAC.

Map Name	Ally Units	Enemy Units
2c_vs_64zg	2 Colossi	64 Zerglings
3s_vs_3z	3 Stalkers	3 Zealots
MMM	1 Medivac, 2 Marauders & 7 Marines	1 Medivac, 2 Marauders & 7 Marines
MMM2	1 Medivac, 2 Marauders & 7 Marines	1 Medivac, 2 Marauders & 8 Marines

(a) 2c_vs_64zg (b) 3s_vs_3z (c) MMM (d) MMM2

Figure 6. Visualization of four maps of StarCraft II

We can see that FOP always converges to the optimal, and the only difference under these approaches is the convergence rate. Therefore, we say that FOP is robust with the temperature parameter adjustment.

Next, we changed FOP's factorization to DOP's linear decomposition to analyze the importance of IGO constraint. Figure 3b shows that, without the factorization based on the IGO constraint, even if the joint action space is well explored, the biased estimate of the joint Q-value leads the individual policies to the sub-optimal, which reveals the significance of the consistency of optimal joint policy and individual policies. Even if we changed the way of temperature annealing, FOP with DOP's decomposition cannot converge to the global optimum.

Last, we changed FOP's Boltzmann policy to the greedy policy by fixing α and α_i to 0. Note that the $\lambda(\boldsymbol{\tau}, \boldsymbol{u})$ in (18) still exists. Figure 3c illustrates that the lack of effective exploration biases the estimate of the joint Q-value and leads the policies to the sub-optimal, which reveals the importance of the soft policy.

D.4. StarCraft II

In StarCraft II, we utilized the default settings of the baselines including VDN, QMIX, QPLEX, and DOP. The network of FOP consists of three layers, a fully-connected layer with 64 units activated by ReLU, followed by a 64 bit GRU, and followed by another fully-connected layer. We adopt the multi-head attention module to construct the weight network with 4 heads. The temperature parameters α and α_i are annealed from 0.5 to 0.05 over $200k$ time steps. The learning rate is set to 5×10^{-4}. The target networks are updated after every 200 training episodes.

We considered four maps including the 2c_vs_64zg, 3s_vs_3z, MMM and the MMM2, whose brief introduction is illustrated in Table 4. The Visualization of the four scenarios are shown in Figure 6. The enemy units are controlled by the built-in AI, and each ally unit is controlled by the reinforcement learning agent. The units of the two groups in 2c_vs_64zg, 3s_vs_3z and MMM2 scenarios are asymmetric. All experiments on StarCraft II utilize the default reward and observation settings of the SMAC benchmark (Samvelyan et al., 2019). We pause the training every episode and evaluate 32 episodes with individual policies to measure *win rate* of each algorithm.