A. Mathematical Formulation

Let $\boldsymbol{\theta}$ and ϕ denote the parameters of the joint policies and the probabilistic classifier, respectively. Then, the whole learning process corresponds to the following bi-level optimization:

$$\begin{aligned} \max_{\boldsymbol{\theta}\in\Theta} \quad & J(\boldsymbol{\theta}, \phi^*(\boldsymbol{\theta})) \\ s.t. \quad & \phi^*(\boldsymbol{\theta}) = \arg\min_{\phi'\in\Phi} \mathcal{L}(\phi', \boldsymbol{\theta}), \end{aligned}$$

where J is the RL objective with intrinsic reward, $\mathcal{L}$ is the loss function of the probabilistic classifier, and ϕ is an implicit function of $\boldsymbol{\theta}$. Therefore, to solve this optimization, we can iteratively update $\boldsymbol{\theta}$ by

$$\frac{\mathrm{d}J(\boldsymbol{\theta}, \phi^*(\boldsymbol{\theta}))}{\mathrm{d}\boldsymbol{\theta}} = \frac{\partial J(\boldsymbol{\theta}, \phi)}{\partial \boldsymbol{\theta}}\bigg|_{\phi=\phi^*(\boldsymbol{\theta})} + \frac{\mathrm{d}\phi^*(\boldsymbol{\theta})}{d\boldsymbol{\theta}} \frac{\partial J(\boldsymbol{\theta}, \phi)}{\partial \phi}\bigg|_{\phi=\phi^*(\boldsymbol{\theta})}$$

where

$$\frac{\mathrm{d}\phi^*(\boldsymbol{\theta})}{d\boldsymbol{\theta}} = -\left(\frac{\partial^2 \mathcal{L}(\phi, \boldsymbol{\theta})}{\partial\phi\partial\phi^T}\right)^{-1} \left(\frac{\partial^2 \mathcal{L}(\phi, \boldsymbol{\theta})}{\partial\phi\partial\boldsymbol{\theta}^T}\right)\bigg|_{\phi=\phi^*(\boldsymbol{\theta})}$$

which is obtained by the implicit function theorem. In practice, the second-order term is neglected due to high computational complexity, without incurring significant performance drop, such as in meta-learning and GANs. Therefore, we can solve the bi-level optimization by the first-order approximation with iterative updates:

$$\begin{aligned} \phi_{k+1} &\approx \arg\min_{\phi} \mathcal{L}(\phi, \mathcal{B}_k) \\ \boldsymbol{\theta}_{k+1} &= \boldsymbol{\theta}_k + \zeta_k \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}, \phi_{k+1}). \end{aligned}$$

B. Hyperparameters

The hyperparameters of EOI and the baselines in each scenario are summarized in Table 1. Since QMIX and MAAC are off-policy algorithms with replay buffer, we do not need to maintain the buffer $\mathcal{B}$ but build the training data from the replay buffer $\mathcal{D}$. For EDTI, ROMA, and HC, we use their default settings.

Table 1. Hyperparameters

Hyperparameter	Pac-man	Windy Maze	Firefighters	Battle	10_vs_10
runs with different seeds	5	10	5	5	5
horizon (T)	30	15	20	100	100
discount (γ)		0.98		0.96	0.995
replay buffer size		2×10^4			1×10^4
actor learning rate		1×10^{-3}		-	3×10^{-4}
critic learning rate		1×10^{-4}		-	1×10^{-4}
QMIX learning rate		1×10^{-4}			-
# MLP units		$(128, 128)$			
batch size		128			
MLP activation		ReLU			
optimizer		Adam			
ϕ learning rate		1×10^{-3}		1×10^{-4}	1×10^{-4}
α in QMIX		0.05		0.02	-
α in MAAC		0.2		-	0.04
β_1		0.04		0.05	0.05
β_2		0.1		0.05	0.05
Δt			4		