A. Regularizing with variational objective

We provide the full derivation of Equation (2) in the following:

$$\begin{aligned}
I(z_t^a;\zeta_t^a,\boldsymbol{s}_t) &= \mathbb{E}_{\boldsymbol{s}_t,z_t^a,\zeta_t^a}\left[\log\frac{p(z_t^a|\zeta_t^a,\boldsymbol{s}_t)}{p(z^a|\boldsymbol{s}_t)}\right] \quad \text{//by the definition of mutual information}\\
&= \mathbb{E}_{\boldsymbol{s}_t,z_t^a,\zeta_t^a}\left[\log\frac{q_\xi(z_t^a|\zeta_t^a,\boldsymbol{s}_t)}{p(z^a|\boldsymbol{s}_t)}\right] + \mathrm{KL}\Big(p(z_t^a|\zeta_t^a,\boldsymbol{s}_t), q_\xi(z_t^a|\zeta_t^a,\boldsymbol{s}_t)\Big)\\
&\geq \mathbb{E}_{\boldsymbol{s}_t,z_t^a,\zeta_t^a}\left[\log\frac{q_\xi(z_t^a|\zeta_t^a,\boldsymbol{s}_t)}{p(z^a|\boldsymbol{s}_t)}\right] \quad \text{//since } \mathrm{KL}(\cdot,\cdot)\geq 0\\
&= \mathbb{E}_{\boldsymbol{s}_t,z_t^a,\zeta_t^a}\Big[\log q_\xi(z_t^a|\zeta_t^a,\boldsymbol{s}_t)\Big] + H(z_t^a|\boldsymbol{s}_t).
\end{aligned} \tag{5}$$

B. Proof of Theorem 1

In this section we provide the proof for Theorem 1.

Theorem 1. *Denote the optimal action-value and value functions by Q_*^{tot} and V_*^{tot}. Denote the action-value and value functions corresponding to receiving new strategies every time from the coach by Q^{tot} and V^{tot}, and thos corresponding to following the strategies distributed according to (3) as $\tilde{Q}$ and $\tilde{V}$, i.e. $\tilde{V}(\boldsymbol{\tau}_t|\tilde{\boldsymbol{z}}_{\hat{t}};\boldsymbol{c}) = \max_{\boldsymbol{u}}\tilde{Q}(\boldsymbol{\tau}_{\hat{t}},\boldsymbol{u}|\tilde{\boldsymbol{z}}_t;\boldsymbol{c})$. Assume for any trajectory $\boldsymbol{\tau}_t$, actions $\boldsymbol{u}_t$, current state $\boldsymbol{s}_t$, the most recent state the coach distribute strategies $\boldsymbol{s}_{\hat{t}}$, and the players' characteristics $\boldsymbol{c}$, $||Q^{tot}(\boldsymbol{\tau}_t,\boldsymbol{u}_t,f(\boldsymbol{s}_{\hat{t}});\boldsymbol{c}) - Q_*^{tot}(\boldsymbol{s}_t,\boldsymbol{u}_t;\boldsymbol{c})||_2 \leq \kappa$, and for any strategies z_1^a, z_2^a, $|Q^{tot}(\boldsymbol{\tau}_t,\boldsymbol{u}_t|z_1^a,\boldsymbol{z}^{-a};\boldsymbol{c}) - Q^{tot}(\boldsymbol{\tau}_t,\boldsymbol{u}_t|z_2^a,\boldsymbol{z}^{-a};\boldsymbol{c})| \leq \eta||z_1^a - z_2^a||_2$. If the used team strategies $\tilde{\boldsymbol{z}}_t$ satisfies $\forall a,t,\ ||\tilde{z}_{\hat{t}}^a - z_{\hat{t}}^a||_2 \leq \beta$, then we have*

$$||V_*^{tot}(\boldsymbol{s}_t;\boldsymbol{c}) - \tilde{V}(\boldsymbol{\tau}_t|\tilde{\boldsymbol{z}}_{\hat{t}};\boldsymbol{c})||_\infty \leq \frac{2(n_a\eta\beta+\kappa)}{1-\gamma}, \tag{6}$$

where n_a is the number of agents and γ is the discount factor.

To summarize, the assumptions assume that the learned action-value function Q^{tot} approximates the optimal Q_*^{tot} well and has bounded Lipschitz constant with respect to individual action-value functions. Moreover, we assume the individual action-value functions also have bounded Lipschitz constant with respect to the strategies.

Proof. According to Assumption 2, if $||\tilde{z}_t^a - z_{\hat{t}}^a||_2 \leq \beta$ for all a, then

$$|Q^{\mathrm{tot}}(\boldsymbol{\tau}_t,\boldsymbol{u}_t|\tilde{\boldsymbol{z}}_t,\boldsymbol{c}) - Q^{\mathrm{tot}}(\boldsymbol{\tau}_t,\boldsymbol{u}_t|\boldsymbol{z}_{\hat{t}},\boldsymbol{c})| \leq \sum_{a_i,1\leq i\leq n_a}\eta_1\eta_2||\tilde{z_t^a} - z_{\hat{t}}^a||_2 \leq n_a\eta_1\eta_2\beta. \tag{7}$$

For notation convenience, we ignore the superscript of *tot* and the condition on $\boldsymbol{c}$. For a state $\boldsymbol{s}$, denote the action the learned policy take as $\boldsymbol{u}^\dagger$, i.e. $\boldsymbol{u}^\dagger \triangleq \mathrm{argmax}_{\boldsymbol{u}} Q(\boldsymbol{\tau},\boldsymbol{u})$. Similarly we can define $\boldsymbol{u}^*$ and $\tilde{\boldsymbol{u}}$ as the action one would take according to the optimal Q_* and the action-value $\tilde{Q}$ estimated using the old strategy. From Assumption 1, we know that

$$Q_*(\boldsymbol{s},\boldsymbol{u}^\dagger) \geq Q(\boldsymbol{\tau},\boldsymbol{u}^\dagger) - \kappa \geq Q(\boldsymbol{\tau},\boldsymbol{u}^*) - \kappa \geq Q_*(\boldsymbol{s},\boldsymbol{u}^*) - 2\kappa. \tag{8}$$

Therefore taking $\boldsymbol{u}^\dagger$ will result in at most 2κ performance drop at this single step. Similarly, denote $\epsilon_0 = n_a\eta_1\eta_2\beta$, then

$$Q(\boldsymbol{\tau},\tilde{\boldsymbol{u}}) \geq \tilde{Q}(\boldsymbol{\tau},\tilde{\boldsymbol{u}}) - \epsilon_0 \geq \tilde{Q}(\boldsymbol{\tau},\boldsymbol{u}^\dagger) - \epsilon_0 \geq Q(\boldsymbol{\tau},\boldsymbol{u}^\dagger) - 2\epsilon_0. \tag{9}$$

Hence $Q_*(\boldsymbol{s},\tilde{\boldsymbol{u}}) \geq Q_*(\boldsymbol{s},\boldsymbol{u}^*) - 2(\epsilon_0+\kappa)$. Note that this means taking the action $\tilde{\boldsymbol{u}}$ in the place of $\boldsymbol{u}^*$ at state $\boldsymbol{s}$ will result in at most $2(\epsilon_0+\kappa)$ performance drop. This conclusion generalizes to any step t. Therefore, if at each single step the performance is bounded within $2(\epsilon_0+\kappa)$, then overall the performance is within $2(\epsilon_0+\kappa)/(1-\gamma)$. □

C. Training Details

For both Resource Collection and Rescue Game, we set the max total number of training steps to 5 million. Then we use the exponentially decayed ϵ-greedy algorithm as our exploration policy, starting from $\epsilon_0 = 1.0$ to $\epsilon_n = 0.05$. We parallelize the

Name	Description	Value
$\|\mathcal{D}\|$	replay buffer size	100000
n_{head}	number of heads in multi-head attention	4
n_{thread}	number of parallel threads for running the environment	8
dh	the hidden dimension of all modules	128
γ	the discount factor	0.99
lr	learning rate	0.0003
	optimizer	RMSprop
α	α value in RMSprop	0.99
ϵ	ϵ value in RMSprop	0.00001
n_{batch}	batch size	256
grad clip	clipping value of gradient	10
target update frequency	how frequent do we update the target network	200 updates
λ_1	λ_1 in variational objective	0.001
λ_2	λ_2 in variational objective	0.0001

Table 3. Hyper-parameters in Resource Collection and Rescue Game.

environment with 8 threads for training. Experiments are run on the GeForce RTX 2080 GPUs. We provide the algorithm hyper-parameters in Table 3.

For StarCraft Micromanagement, we follow the same setup from (Iqbal et al., 2020) and train all methods on the 3-8sz and 3-8MMM maps for 12 millions steps. To regularize the learning, we use $\lambda_1 = 0.00005$ and $\lambda_2 = 0.000005$ for both maps. For all experiments, we set the default period before centralization to $T = 4$.