A Proofs

A.1 Proof of Theorem 4.1

Proof. For any two global state value functions V_{tot}^1 and V_{tot}^2, let π_{tot}^1 be the optimal global policy under $\mathcal{T}_f^* V_{tot}^1$, and we can get:

$$\begin{aligned}
&\left(\mathcal{T}_f^* V_{tot}^1\right)(\boldsymbol{o}) - \left(\mathcal{T}_f^* V_{tot}^2\right)(\boldsymbol{o}) \\
&= \sum_{\boldsymbol{a}} \pi_{tot}^1(\boldsymbol{a}|\boldsymbol{o}) \left[r + \gamma \mathbb{E}_{\boldsymbol{o}'}\left[V_{tot}^1\left(\boldsymbol{o}'\right)\right] - \alpha \log\left(\frac{\pi_{tot}^1(\boldsymbol{a}|\boldsymbol{o})}{\mu_{tot}(\boldsymbol{a}|\boldsymbol{o})}\right)\right] - \max_{\pi_{tot}} \sum_{\boldsymbol{a}} \pi_{tot}(\boldsymbol{a}|\boldsymbol{o}) \left[r + \gamma \mathbb{E}_{\boldsymbol{o}'}\left[V_{tot}^2\left(\boldsymbol{o}'\right)\right] - \alpha \log\left(\frac{\pi_{tot}(\boldsymbol{a}|\boldsymbol{o})}{\mu_{tot}(\boldsymbol{a}|\boldsymbol{o})}\right)\right] \\
&\leq \sum_{\boldsymbol{a}} \pi_{tot}^1(\boldsymbol{a}|\boldsymbol{o}) \left[r + \gamma \mathbb{E}_{\boldsymbol{o}'}\left[V_{tot}^1\left(\boldsymbol{o}'\right)\right] - \alpha \log\left(\frac{\pi_{tot}^1(\boldsymbol{a}|\boldsymbol{o})}{\mu_{tot}(\boldsymbol{a}|\boldsymbol{o})}\right)\right] - \sum_{\boldsymbol{a}} \pi_{tot}^1(\boldsymbol{a}|\boldsymbol{o}) \left[r + \gamma \mathbb{E}_{\boldsymbol{o}'}\left[V_{tot}^2\left(\boldsymbol{o}'\right)\right] - \alpha \log\left(\frac{\pi_{tot}^1(\boldsymbol{a}|\boldsymbol{o})}{\mu_{tot}(\boldsymbol{a}|\boldsymbol{o})}\right)\right] \\
&= \gamma \sum \pi_{tot}^1(\boldsymbol{a}|\boldsymbol{o}) \mathbb{E}_{\boldsymbol{o}'}\left[V_{tot}^1\left(\boldsymbol{o}'\right) - V_{tot}^2\left(\boldsymbol{o}'\right)\right] \\
&\leq \gamma \left\|V_{tot}^1 - V_{tot}^2\right\|_\infty
\end{aligned}$$

Therefore, it follows that:

$$\left\|\mathcal{T}_f^* V_{tot}^1 - \mathcal{T}_f^* V_{tot}^2\right\|_\infty \leq \gamma \left\|V_{tot}^1 - V_{tot}^2\right\|_\infty$$

□

A.2 Proof of Proposition 4.2

Proof. For a behavior-regularized Dec-POMDP with $f(\pi_{tot}, \mu_{tot}) = \log(\pi_{tot}/\mu_{tot})$, the learning objective can be written as $\max_{\pi_{tot}} \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \left(r\left(\boldsymbol{o}_t, \boldsymbol{a}_t\right) - \alpha log\left(\pi_{tot}\left(\boldsymbol{a}_t|\boldsymbol{o}_t\right)/\mu_{tot}\left(\boldsymbol{a}_t|\boldsymbol{o}_t\right)\right)\right)\right]$. Its Lagrangian function can obtain when the optimal global policy is written as follows:

$$\begin{aligned}
L(\pi_{tot}, \beta, u) = &\sum_{\boldsymbol{o}} d_{\pi_{tot}}(\boldsymbol{o}) \sum_{\boldsymbol{a}} \pi_{tot}(\boldsymbol{a}|\boldsymbol{o}) \left(Q_{tot}(\boldsymbol{o}, \boldsymbol{a}) - \alpha \log\left(\frac{\pi_{tot}(\boldsymbol{a}|\boldsymbol{o})}{\mu_{tot}(\boldsymbol{a}|\boldsymbol{o})}\right)\right) \\
&- \sum_{\boldsymbol{o}} d_{\pi_{tot}}(\boldsymbol{o}) \left[u(\boldsymbol{o}) \left(\sum_{\boldsymbol{a}} \pi_{tot}(\boldsymbol{a}|\boldsymbol{o}) - 1\right) + \sum_{\boldsymbol{a}} \beta(\boldsymbol{a}|\boldsymbol{o}) \pi_{tot}(\boldsymbol{a}|\boldsymbol{o})\right],
\end{aligned}$$

where $d_{\pi_{tot}}$ is the stationary joint observation distribution of the global policy π_{tot}. u and β are Lagrangian multipliers for the equality and inequality constraints.

According to the Karush-Kuhn-Tucker (KKT) conditions where the derivative of the Lagrangian objective function with respect to the global policy is zero at the optimal solution, it follows that:

$$\begin{aligned}
&Q_{tot}(\boldsymbol{o}, \boldsymbol{a}) - \alpha\left(\log\left(\frac{\pi_{tot}(\boldsymbol{a}|\boldsymbol{o})}{\mu_{tot}(\boldsymbol{a}|\boldsymbol{o})}\right) + 1\right) - u(\boldsymbol{o}) + \beta(\boldsymbol{a}|\boldsymbol{o}) = 0 \qquad (16) \\
&\sum_{\boldsymbol{a}} \pi_{tot}(\boldsymbol{a}|\boldsymbol{o}) = 1 \\
&\beta(\boldsymbol{a}|\boldsymbol{o}) \pi_{tot}(\boldsymbol{a}|\boldsymbol{o}) = 0 \\
&0 \leq \pi_{tot}(\boldsymbol{a}|\boldsymbol{o}) \leq 1 \text{ and } 0 \leq \beta(\boldsymbol{a}|\boldsymbol{o})
\end{aligned}$$

From Eq. (16), we can further solve the optimal global policy as:

$$\pi_{tot}(\boldsymbol{a}|\boldsymbol{o}) = \mu_{tot}(\boldsymbol{a}|\boldsymbol{o}) \cdot \exp\left(\frac{Q_{tot}(\boldsymbol{o}, \boldsymbol{a}) - u(\boldsymbol{o}) + \beta(\boldsymbol{a}|\boldsymbol{o})}{\alpha} - 1\right)$$

The above formula can be further simplified. β is the Lagrangian multiplier, and meets complementary slackness $\beta(\boldsymbol{a}|\boldsymbol{o})\pi_{tot}(\boldsymbol{a}|\boldsymbol{o}) = 0$. Considering the joint observation $\boldsymbol{o}$ is fixed, $\exp\left(\frac{Q_{tot}(\boldsymbol{o},\boldsymbol{a}) - u(\boldsymbol{o}) + \beta(\boldsymbol{a}|\boldsymbol{o})}{\alpha} - 1\right)$ is always larger than 0. Therefore, for any positive probability action, its corresponding Lagrangian multiplier $\beta(a|s)$ is 0. Therefore, $\pi_{tot}(\boldsymbol{a}|\boldsymbol{o})$ can be reformulated as:

$$\pi_{tot}(\boldsymbol{a}|\boldsymbol{o}) = \mu_{tot}(\boldsymbol{a}|\boldsymbol{o}) \cdot \exp\left(\frac{Q_{tot}(\boldsymbol{o}, \boldsymbol{a}) - u(\boldsymbol{o})}{\alpha} - 1\right) \qquad (17)$$

Bringing Eq. (17) into $\sum_{\boldsymbol{a}} \pi_{tot}(\boldsymbol{a}|\boldsymbol{o}) = 1$, we have:

$$\mathbb{E}_{\boldsymbol{a}\sim\mu_{tot}}\left[\exp\left(\frac{Q_{tot}(\boldsymbol{o},\boldsymbol{a}) - u(\boldsymbol{o})}{\alpha} - 1\right)\right] = 1 \tag{18}$$

The left side of Eq. (18) can be seen as a continuous and monotonic function of u, so it has only one solution denoted as u^*, and we denote the corresponding policy π_{tot} as π^*_{tot}.

Integrating Eq. (17) into the expression of optimal global state value, we can get:

$$\begin{aligned}
V^*_{tot}(\boldsymbol{o}) &= \mathcal{T}^*_f V^*_{tot}(\boldsymbol{o}) \\
&= \sum_{\boldsymbol{a}} \pi^*_{tot}(\boldsymbol{a}|\boldsymbol{o}) \left(Q^*_{tot}(\boldsymbol{o},\boldsymbol{a}) - \alpha \log\left(\frac{\pi^*_{tot}(\boldsymbol{a}|\boldsymbol{o})}{\mu_{tot}(\boldsymbol{a}|\boldsymbol{o})}\right)\right) \\
&= \sum_{\boldsymbol{a}} \pi^*_{tot}(\boldsymbol{a}|\boldsymbol{o}) \left(u^*(\boldsymbol{o}) + \alpha\right) \\
&= u^*(\boldsymbol{o}) + \alpha
\end{aligned}$$

To summarize, we obtain the optimality condition of the behavior regularized MDP with Reverse KL divergence as follows:

$$\begin{aligned}
&Q^*_{tot}(\boldsymbol{o},\boldsymbol{a}) = r(\boldsymbol{o},\boldsymbol{a}) + \gamma \mathbb{E}_{\boldsymbol{o}'|\boldsymbol{o},\boldsymbol{a}}\left[V^*_{tot}\left(\boldsymbol{o}'\right)\right] \\
&V^*_{tot}(\boldsymbol{o}) = u^*(\boldsymbol{o}) + \alpha \\
&\pi^*_{tot}(\boldsymbol{a}|\boldsymbol{o}) = \mu_{tot}(\boldsymbol{a}|\boldsymbol{o}) \cdot \exp\left(\frac{Q^*_{tot}(\boldsymbol{o},\boldsymbol{a}) - u^*(\boldsymbol{o})}{\alpha} - 1\right)
\end{aligned}$$

where $u(\boldsymbol{o})$ is a normalization term and has a optimal value u^* that makes the corresponding optimal policy π^*_{tot} satisfy $\sum_{\boldsymbol{a}\in\mathcal{A}^n} \pi^*_{tot}(\boldsymbol{a}|\boldsymbol{o}) = 1$.

□

B Experiment Settings

B.1 Multi-Agent MuJoCo

Multi-agent Mujoco [7] is a benchmark framework developed for assessing and comparing the effectiveness of algorithms in continuous multi-agent robotic control. Within this framework, a robotic system is partitioned into independent agents, each tasked with controlling a specific set of joints. The agents collaborate harmoniously to accomplish shared objectives, such as acquiring the ability to walk through an environment, with the ultimate goal of maximizing the cumulative reward. Multi-agent MuJoCo environment consists of multiple different robot configurations, and it is often used for the study of novel MARL algorithms for decentralized coordination in isolation.

To generate the dataset transitions, we captured the interactions between the environment and trained online MARL algorithms. Specifically, we use HAPPO [16] algorithm to collect data. The expert dataset is generated by employing the converged HAPPO algorithm. This involves training the algorithm until it reaches a state of convergence, where the agents have learned optimal policies. The medium dataset is generated by first training a policy online using HAPPO, early-stopping the training, and collecting samples from this partially-trained policy. The medium-replay dataset consists of recording all samples in the replay buffer observed during training until the policy reaches the medium level of performance. The medium-expert dataset by mixing equal amounts of expert demonstrations and suboptimal data. For all datasets, the hyperparameter env_args.agent_obsk (determines up to which connection distance agents will be able to form observations) is set to 1. The reward distribution of our datasets is listed in Table 2.

B.2 The StarCraft Multi-Agent Challenge

The StarCraft Multi-Agent Challenge (SMAC) benchmark is chosen as our testing environment. Due to its high control complexity, SMAC is a popular multi-agent cooperative control environment for evaluating advanced MARL methods. It consists of a collection of StarCraft II microscenarios in

Table 2: The multi-agent MuJoCo dataset

Scenario	Quality	Reward Distribution
2-Agent Ant	expert	2.06±0.35
	medium	1.42±0.37
	medium-expert	1.74±0.48
	medium-replay	1.03±0.21
3-Agent Hopper	expert	3.64±0.79
	medium	3.16±1.00
	medium-expert	3.41±0.93
	medium-replay	2.37±0.69
6-Agent HalfCheetah	expert	2.79±1.76
	medium	1.43±1.36
	medium-expert	2.11±1.71
	medium-replay	0.66±1.09

which two groups of units engage in combat. Agents based on the MARL algorithm control the first group's units, while a built-in heuristic game AI bot with different difficulties controls the second group's units. Scenarios vary in terms of the initial location, number and type of units, and elevated or impassable terrain. The available actions for each agent include no operation, move[direction], attack [enemy id], and stop. The reward that each agent receives is the same. The hit-point damage dealt and received determines the agents' share of the reward. SMAC consists of several StarCraft II multi-agent micromanagement maps. We consider 4 representative battle maps, including 2 hard map (5m_vs_6m, 2c_vs_64zg), and 2 super hard maps (6h_vs_8z, corridor), as our experiment tasks. The task type and other details of the maps are listed in the Table 3.

Table 3: SMAC maps for experiments.

Map Name	Ally Units	Enemy Units	Type
5m_vs_6m	5 Marines	6 Marines	homogeneous & asymmetric
2c_vs_64zg	2 Colossi	64 Zerglings	micro-trick: positioning
6h_vs_8z	6 Hydralisks	8 Zealots	micro-trick: focus fire
corridor	6 Zealots	24 Zerglings	micro-trick: wall off

The offline SMAC dataset used in this study is provided by [22], which is the largest open offline dataset on SMAC. Different from single-agent offline datasets, it considers the property of Dec-POMDP, which owns local observations and available actions for each agent. The dataset is collected from the trained MAPPO agent, and includes three quality levels: good, medium, and poor. For each original large dataset, we randomly sample 1000 episodes as our dataset.

C Implementation Details

C.1 Details of OMIGA

The local Q-value, state value networks and policy networks of OMIGA are represented by 3-layers ReLU activated MLPs with 256 units for each hidden layer. For the weight network, we use 2-layer ReLU activated MLPs with 64 units for each hidden layer. All the networks are optimized by Adam optimizer.

C.2 Details of baselines

We compare OMIGA against four recent offline MARL algorithms: ICQ [40], OMAR [25], BCQ-MA and CQL-MA. For the ICQ and OMAR, we implement them based on the algorithm description in their papers. BCQ-MA is the multi-agent version of BCQ, and CQL-MA is the multi-agent version of CQL. BCQ-MA and CQL-MA use linear weighted value decomposition structure as $Q_{tot} = \sum_{i=1}^{n} w_i(\boldsymbol{o})Q_i\left(o_i, a_i\right) + b(\boldsymbol{o}), w^i \geq 0$ for the multi-agent setting. The policy constrain of BCQ-MA and the value regularization of CQL-MA are both imposed on the local Q-value.

In this paper, all experiments are implemented with pytorch and executed on NVIDIA V100 GPUs.

C.3 Hyperparameters

For multi-agent MuJoCo, the hyperparameters of OMIGA and baselines are listed in Table 4. An important hyperparameter of OMIGA is the regularization hyperparameter α. The higher α encourages OMIGA staying near the behavioral distribution, and lower α makes OMIGA more optimistic. On most tasks, we use $\alpha = 10$ to ensure good regularization effect. On the medium quality dataset of HalfCheetah task, we choose $\alpha = 1$.

Table 4: Hyperparameters of OMIGA and baselines for multi-agent MuJoCo

Hyperparameter	Value
Shared parameters	
Q-value network learning rate	5e-4
Policy network learning rate	5e-4
Optimizer	Adam
Target update rate	0.005
Batch size	128
Discount factor	0.99
Hidden dimension	256
Weight network hidden dimension	64
OMIGA	
State value network learning rate	5e-4
Regularization parameter α	1 or 10
Others	
Lagrangian coefficient (ICQ)	10
Tradeoff factor α (OMAR, CQL-MA)	1

For SMAC, the hyperparameters of OMIGA and baselines are listed in Table 5. On most tasks, we use $\alpha = 10$. On the poor dataset of 6h_vs_8z map, the quality of the dataset is relatively poor. It does not make much sense to make the policy close to the behavioral policy, so we choose $\alpha = 2$ to make the algorithm more radical.

On the most SMAC maps, the learning rate of all networks is set to 5e-4. The exception is the map 2c_vs_64zg, on this map, the learning rate of all networks is set to 1e-4.

Table 5: Hyperparameters of OMIGA and baselines for SMAC

Hyperparameter	Value
Shared parameters	
Q-value network learning rate	5e-4 or 1e-4
Policy network learning rate	5e-4 or 1e-4
Optimizer	Adam
Target update rate	0.005
Batch size	128
Discount factor	0.99
Hidden dimension	256
Weight network hidden dimension	64
OMIGA	
State value network learning rate	5e-4 or 1e-4
Regularization parameter α	2 or 10
Others	
Lagrangian coefficient (ICQ)	10
Threshold (BCQ-MA)	0.3
Tradeoff factor α (OMAR, CQL-MA)	1

Table 6: Average scores and standard deviations over 5 random seeds on the mixed offline SMAC datasets

Map	Dataset	BCQ-MA	CQL-MA	ICQ	OMAR	OMIGA(ours)
6h_vs_8z	good-poor	11.41±0.44	9.56±0.25	11.00±0.36	9.17±0.19	**11.88±0.27**
6h_vs_8z	good-medium	11.79±0.29	10.08±0.26	11.18±0.25	10.02±0.16	**12.05±0.47**
6h_vs_8z	medium-poor	11.18±0.41	10.73±0.38	11.25±0.35	10.42±0.19	**11.85±0.35**
corridor	good-poor	12.37±1.36	4.88±0.35	11.78±1.53	5.54±0.75	**13.01±0.89**
corridor	good-medium	13.32±0.71	5.77±1.30	12.98±0.62	6.63±0.74	**14.02±1.04**
corridor	medium-poor	8.11±0.35	6.18±0.59	8.27±0.48	6.25 ±0.48	**9.70±1.40**

D Additional Results

D.1 Results on mixed datasets

We want to investigate whether OMIGA has superior performance when the datasets are mixed. Unlike BCQ-MA and OMAR, OMIGA doesn't need to learn a behavior policy. We choose two original datasets on the SMAC super hard maps 6h_vs_8z and corridor, and make mixed datasets by combining these SMAC datasets of different quality, including good-poor, good-medium, medium-poor datasets. Each mixed dataset is blended by 50% of each of the two original datasets. On these mixed suboptimal datasets, the behavior policy is heterogeneous. Therefore, it is more difficult for algorithms such as BCQ-MA and OMAR to learn an accurate behavior policy, making implicit value learning with the regularization framework of OMIGA more appealing.

Table 6 shows the results that OMIGA consistently outperforms other offline MARL baselines under all different mixed dataset experiments. Compared with the results on the original datasets, the performance of OMIGA has become more leading, indicating the benefits of implicit value regularization of OMIGA.