DIFFER: Decomposing Individual Reward for Fair Experience Replay in Multi-Agent Reinforcement Learning
– Supplementary Material –

A The Proof of Proposition 1

Proposition 1. *The invariance of gradient is satisfied when individual reward of agent i is given by:*

$$r_i = (R + \gamma \widetilde{Q}_{\text{tot}} - Q_{\text{tot}}) \frac{\partial Q_{\text{tot}}}{\partial Q_i} - \gamma \widetilde{Q}_i + Q_i. \tag{S1}$$

for any agent $i \in I$.

Proof. According to the chain rule,

$$\frac{\partial L_{\text{tot}}}{\partial \theta_p} = \frac{\partial L_{\text{tot}}}{\partial Q_{\text{tot}}} \sum_{i \in I} \frac{\partial Q_{\text{tot}}}{\partial Q_i} \frac{\partial Q_i}{\partial \theta_p} = 2(R + \gamma \widetilde{Q}_{\text{tot}} - Q_{\text{tot}}) \sum_{i \in I} \frac{\partial Q_{\text{tot}}}{\partial Q_i} \frac{\partial Q_i}{\partial \theta_p}; \tag{S2}$$

$$\sum_{i \in I} \frac{\partial L_i}{\partial \theta_p} = \sum_{i \in I} \frac{\partial L_i}{\partial Q_i} \frac{\partial Q_i}{\partial \theta_p} = \sum_{i \in I} 2(r_i + \gamma \widetilde{Q}_i - Q_i) \frac{\partial Q_i}{\partial \theta_p}. \tag{S3}$$

Therefore,

$$\begin{aligned}
&\frac{\partial L_{\text{tot}}}{\partial \theta_p} = \sum_{i \in I} \frac{\partial L_i}{\partial \theta_p} \\
\iff\ & 2(R + \gamma \widetilde{Q}_{\text{tot}} - Q_{\text{tot}}) \sum_{i \in I} \frac{\partial Q_{\text{tot}}}{\partial Q_i} \frac{\partial Q_i}{\partial \theta_p} = \sum_{i \in I} 2(r_i + \gamma \widetilde{Q}_i - Q_i) \frac{\partial Q_i}{\partial \theta_p} \\
\iff\ & 2 \sum_{i \in I} [(R + \gamma \widetilde{Q}_{\text{tot}} - Q_{\text{tot}}) \frac{\partial Q_{\text{tot}}}{\partial Q_i}] \frac{\partial Q_i}{\partial \theta_p} = 2 \sum_{i \in I} (r_i + \gamma \widetilde{Q}_i - Q_i) \frac{\partial Q_i}{\partial \theta_p} \\
\Leftarrow\ & (R + \gamma \widetilde{Q}_{\text{tot}} - Q_{\text{tot}}) \frac{\partial Q_{\text{tot}}}{\partial Q_i} = r_i + \gamma \widetilde{Q}_i - Q_i, \forall i \in I \\
\iff\ & r_i = (R + \gamma \widetilde{Q}_{\text{tot}} - Q_{\text{tot}}) \frac{\partial Q_{\text{tot}}}{\partial Q_i} - \gamma \widetilde{Q}_i + Q_i, \forall i \in I.
\end{aligned} \tag{S4}$$

In conclusion, a sufficient condition for invariance of gradients is as follow: individual reward of agent i is calculated as Equ. (S1) for any $i \in I$. □

B The Hyper-parameters Setting

In DIFFER, the replay buffer stores the most recent 5000 episodes, and mini-batches of size 32 are sampled from it. The target network is updated every 200 episodes. RMSProp is utilized as the

Preprint. Under review.

optimizer for both the mixing network and agent network, with a learning rate of 0.0005, α (decay rate) set to 0.99, and ϵ (small constant) set to 0.00001. Gradients are clipped within the range of [-10, 10]. The discount factor for the expected reward (return) is 0.99.

Regarding the fair experience replay, we apply the "warm-up" technique to the sample ratio η. It linearly increases from 0.8 to 1.0 over 60% of the time steps and remains constant thereafter. The regulation parameter α, which determines the prioritization degree in fair experience replay, is set to 0.8. The regulation parameter β of the sampling probability anneals linearly from 0.6 to 1.0.

For exploration, we employ an ϵ-greedy strategy with ϵ linearly annealed from 1.0 to 0.05 over 50K time steps and then held constant for the remainder of the training. The maximum time step during training is set to 2005000 for all experiments. Each scenario is run with 5 different random seeds.

The agent network architecture follows a DRQN[1] structure, consisting of a GRU layer for the recurrent layer with a 64-dimensional hidden layer. Before and after the GRU layer, there are fully-connected layers with 64 dimensions each. The mixing network is implemented using open-source code. All experiments on the SMAC benchmark adopt the default reward and observation settings provided by the benchmark. For the baseline algorithms, we use the authors' code with hyper-parameters fine-tuned specifically for the SMAC benchmark.

The supplementary materials contain the code for our DIFFER framework. The main codes of individual reward calculation and fair experience replay are shown in DIFFER_code/learners/q_learner_divide.py and DIFFER_code/ER/PER/prioritized_memory.py, respectively.

The specific hyper-parameters are listed below.

Table S1: The hyper-parameters that used in our DIFFER framework.

Hyper parameters	Meaning	Value
epsilon_start	Start value of ϵ anneal	1.0
epsilon_end	End value of ϵ anneal	0.05
epsilon_anneal_time	Duration step of ϵ anneal	50000
mempool_size	Memory pool size in learner	5000
batch_size	Batch size per training	32
target_update_interval	Interval steps between 2 updates of target network	200
lr	Learning rate	0.0005
regularization	Weights of L1-regularizer	0.0005
tot_optimizer	Optimizer in training	RMSProp
tot_optim_alpha	Alpha of optimizer	0.99
tot_optim_eps	Epsilon of optimizer	0.00001
ind_optimizer	Optimizer in training	RMSProp
ind_optim_alpha	Alpha of optimizer	0.99
ind_optim_eps	Epsilon of optimizer	0.00001
grad_norm_clip	Clip range of gradient normalization	10
η_{start}	Start value of η warm up	0.8
η_{end}	End value of η warm up	1.0
p	Duration step ratio of η warm up	0.6
α	Regulation parameter of sampling probability	0.8
ϵ	Regulation parameter of sampling probability	0.01
β_{start}	Start value of β warm up	0.6
β_{end}	End value of β warm up	1.0

C Ablation Studies for warm_up Technique of Sample Ratio

To highlight the significance of the $warm\_up$ trick for the individual experience sample ratio η, we conduct ablation experiments on SMAC maps `MMM2`. Figure S1 showcases the performance of QMIX-DIFFER with and without the $warm\_up$ sample ratio. In this context, the model **W-warm_up (ours)**

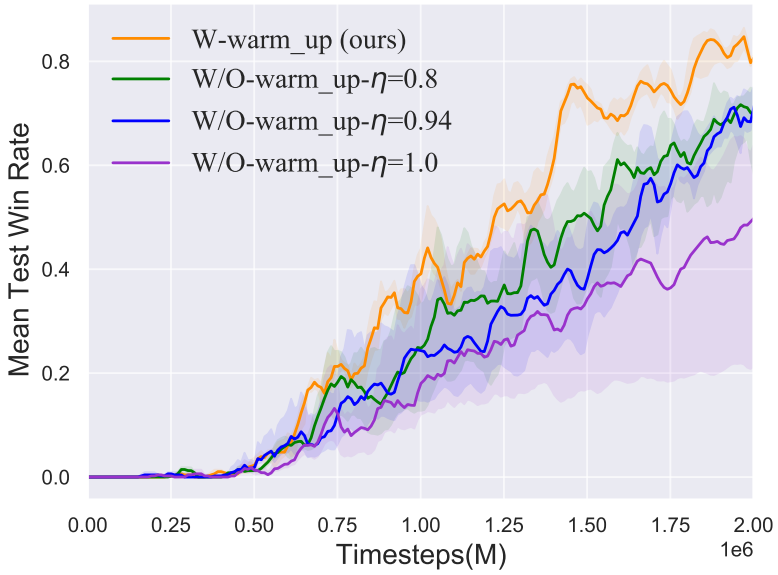

Figure S1: Ablation studies regarding the *warm up* trick of individual transition sample ratio on `MMM2`. "W-warm_up (ours)" represents the sample ratio increases as the training step increases until 1.0. "W/O-warm_up" represents the sample ratio η is fixed during training.

refers to the setting where the sample ratio linearly increases as the training progresses. On the other hand, **W/O-warm_up** denotes the scenario where the sample ratio remains fixed throughout the training phase. To ensure a fair comparison, we conduct experiments with different fixed values of η, while keeping the batch-size of all three models in Figure S1 consistent at 32. Let us assume that the overall quantity of training data is K for W/O-warm_up-η=1.0. By employing the hyper-parameter setting detailed in Section B, we can calculate that the overall training data amounts to $0.94K$ for W-warm_up. Consequently, we select η from the set $\{0.8, 0.94, 1.0\}$. The results clearly demonstrate that the $warm\_up$ trick leads to performance improvements, even when using a smaller amount of training data. This highlights the importance of incorporating the $warm\_up$ strategy into the training process.

D Analysis of Sampling Percentage

In this section, we present the distribution of individual experience sampling percentages across different agent classes within the `MMM2` scenario from the SMAC environment[2]. The team composition consists of three distinct unit classes: 1 Medivac unit responsible for healing teammates, 2 Marauder units specialized in offensive actions, and 7 Marine units dedicated to attacking. Throughout the training process, DIFFER selects important individual experiences from a diverse set of 10 agents to update the agent networks. We calculate the sampling percentages of individual experiences for each agent and derive the average sampling percentage for each unit class. The corresponding curves, focusing on the later stages of training for improved stability, are depicted in Figure S2.

The graph showcases significant variations in the average sampling percentages among the unit classes, reflecting their distinct learning difficulties. As the learning difficulty of a particular unit class intensifies, a larger proportion of individual experiences from that class are sampled during training. This adaptive behavior of the DIFFER method highlights its ability to dynamically adjust the sampling percentages to address the unique learning challenges encountered by each unit class. Notably, achieving effective strategies for the Medivac unit proves particularly challenging due to its distinctive sub-task and limited availability. Consequently, the average sampling percentage of the Medivac unit surpasses that of the other two unit classes significantly. Although the number of Marine units exceeds that of Marauder units, the Marines are encouraged to adopt a specialized positioning strategy. This strategy requires the Marines to spread out in a fan-like formation, aiming to minimize the damage inflicted upon them while maximizing their effectiveness in dealing damage

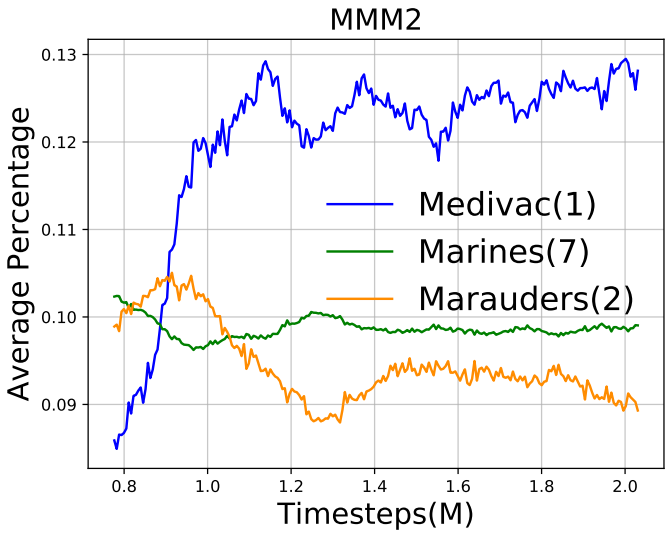

Figure S2: Average percentage of sampled individual experiences from each unit class on SMAC scenario `MMM2`. The number in the label represents the quantity of unit. The sampling percentage of a unit class is high when it poses a learning challenge.

to the enemy. Consequently, the Marines face higher learning difficulties compared to the Marauders. Reflecting this distinction, the average sampling percentage of the Marine unit class surpasses that of the Marauder unit class.

E Environment Introduction

In this section, we provide a brief overview of the multi-agent environment discussed in this paper.

(a) SMAC (b) GRF (c) MAMujoco

Figure S3: Visualisation of three experimental environments.

E.1 SMAC

The Starcraft Multi-Agent Challenge (SMAC)[2] (as shown in Figure S3 (a)) is a prominent cooperative multi-agent environment known for its partial observability. In this paper, we utilize the default environment setting of SMAC, specifically version SC2.4.10. SMAC offers a variety of scenarios, each featuring a confrontation between two teams of units. One team is controlled by a trained model, while the other team is governed by a built-in strategy. The scenarios differ in terms of the initial unit positions, the number and types of units in each team, as well as the characteristics of the map terrain. In the SMAC environment, a team is deemed victorious when all units belonging to the opposing team are eliminated. The primary objective of the policy is to maximize the win rate across all scenarios. To facilitate training, the environment provides a shaped reward function that takes into

account factors such as hit-point damage inflicted and received by agents, the number of units killed, and the outcome of the battle. Learning a coordinated micromanagement strategy in diverse maps poses a significant challenge for the agents in this environment. Such a strategy aims to maximize the damage inflicted upon enemies while minimizing the damage sustained. Overall, the SMAC environment demands agents to acquire sophisticated coordination and decision-making skills in order to excel in diverse scenarios and effectively apply micromanagement strategies.

E.2 GRF

Google Research Football (GRF) environment [3] (shown in Figure S3 (b)) presents a challenging multi-agent reinforcement learning (MARL) setting where a team of agents must learn to pass the ball amongst themselves and overcome their opponents' defense to score goals. Similar to the SMAC environment, the opposing team in GRF is controlled by a built-in strategy. In GRF, each agent has 19 different actions at their disposal, including standard move actions in eight directions, as well as various ball-kicking techniques such as short and long passes, shooting, and high passes that are difficult for opponents to intercept. The agents' observations encompass information about their own position, movement direction, the positions of other agents, and the ball. An episode in GRF terminates either after a fixed number of steps or when one of the teams successfully scores a goal. The winning team receives a reward of +100, while the opposing team is rewarded with -1. In this study, we specifically focus on two official scenarios: `academy_3_vs_1_with_keeper` and `academy_counterattack_hard`.

E.3 MAMujoco

The Multi-Agent Mujoco (MAMujoco) environment[4] (shown in Figure S3 (c)) is a highly versatile and realistic simulation platform designed for multi-agent robotic control tasks. It is based on the Mujoco physics engine and offers continuous action spaces, allowing for smooth and precise control of robotic agents. Each agent in the environment represents a specific component or a complete robot, enabling the simulation of complex multi-robot systems. All of the tasks mentioned in this work are configured according to their default configuration. We set maximum observation distance to $k = 0$ for $\texttt{Humanoid} - \texttt{v2}$ and $k = 1$ for `manyagent_swimmer`.

We also provide the description of agent types in each scenario in Table S2. It aims to enhance the reader's understanding of the distinct performance improvements brought about by DIFFER.

Table S2: The agent type distribution in each scenario.

Env.	Scenario Name	#Agent	#Type	Type Distribution
SMAC	27m_vs_30m	27	1	27
	2c_vs_64zg	2	1	2
	MMM2	10	3	1-2-7
	3s5z_vs_3s6z	8	2	3-5
GRF	academy_3_vs_1_with_keeper	3	2	1-2
	academy_counterattack_hard	4	2	2-2
MAMujoco	Humanoid-v2_0	2	2	1-1
	manyagent_swimmer	20	10	2-2-2-2-2-2-2-2-2-2

References

[1] M. Hausknecht and P. Stone, "Deep recurrent q-learning for partially observable mdps," *arXiv preprint arXiv:1507.06527*, 2015.

[2] M. Samvelyan, T. Rashid, C. S. de Witt, G. Farquhar, N. Nardelli, T. G. J. Rudner, C.-M. Hung, P. H. S. Torr, J. Foerster, and S. Whiteson, "The StarCraft Multi-Agent Challenge," *CoRR*, vol. abs/1902.04043, 2019.

[3] K. Kurach, A. Raichuk, P. Stanczyk, M. Zajac, O. Bachem, L. Espeholt, C. Riquelme, D. Vincent, M. Michalski, O. Bousquet, *et al.*, "Google research football: A novel reinforcement learning environment," in *AAAI Conference on Artificial Intelligence (AAAI)*, vol. 34, pp. 4501–4510, 2020.

[4] B. Peng, T. Rashid, C. Schroeder de Witt, P.-A. Kamienny, P. Torr, W. Böhmer, and S. Whiteson, "Facmac: Factored multi-agent centralised policy gradients," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 34, pp. 12208–12221, 2021.