Action Selection Advantage Estimation in Multi-Agent Reinforcement Learning

Xiaoning Zhao
Waseda University
Tokyo, Japan
z.xiaoning@isl.cs.waseda.ac.jp

Toshiharu Sugawara
Waseda University
Tokyo, Japan
sugawara@waseda.jp

***Abstract*—On-policy Multi-Agent Reinforcement Learning (MARL) methods, such as Multi-Agent Proximal Policy Optimization (MAPPO), Independent Proximal Policy Optimization, or Multi-Agent Deep Deterministic Policy Gradient, are widely used because of their perfect balance between simplicity and stability. But they often collect substantial amounts of data from the environment. In real-world scenarios, collecting environmental samples is usually more expensive than training agents, since each interaction often requires costly data acquisition from the environment or human-involved processes. One way to solve this issue is to increase the sample efficiency of the current training paradigm. To tackle this issue, we present Selective Advantage Estimation (SAE), which is able to increase the impact of better-than-baseline actions to avoid overly conservative updates under a clipped policy. SAE is designed to be a drop-in replacement for Generalized Advantage Estimation (GAE) and can be used directly with the current MARL training paradigm that uses advantage estimation without additional work. Our method is tested in two classic MARL environments, Multi-Agent Particle Environments and StarCraft Multi-Agent Challenge (SMAC). In SMAC, compared with GAE, SAE accelerates training by up to 71% in the time required to reach win-rate milestones (50%-70%). Our results indicate that selectively amplifying the signal from high-quality actions is an effective and practical way to improve MAPPO training efficiency.**

***Index Terms*—Multi-agent system, Training efficiency, Multi-agent reinforcement learning, Homogeneous agent**

I. Introduction

Reinforcement Learning (RL) has emerged as a important research focus within machine learning. Its decision-making framework is particularly effective for tasks requiring complex, human-like behaviors. Notably, Reinforcement Learning with human feedback [1] is algorithm in the training of Large Language Models (LLMs) to ensure alignment with human intent. Among current RL paradigms, Proximal Policy Optimization (PPO) [2] is widely adopted for on-policy training due to its balance of stability and simplicity. Despite its popularity, PPO is inherently sample-inefficient. This limitation is worsened in Multi-Agent Reinforcement Learning (MARL) scenarios, where the simultaneous exploration of multiple agents leads to a significant increase in sample complexity. Consequently, this inefficiency hinders the deployment of PPO in large-scale multi-agent domains, such as path-finding, traffic management, and gaming AI [3]–[5].

One effective approach to improve sampling efficiency in on-policy reinforcement learning is the utilization of advantage functions. These functions quantify the relative quality of actions, guiding the agent to update its strategy more effectively. For instance, Generalized Advantage Estimation (GAE) [6], one of the most widely used methods, balances bias and variance to enhance PPO's sample efficiency. This capability facilitates faster learning and broader applicability in multi-agent systems, high-dimensional continuous control, and real-world decision-making domains.

However, studies focusing on improving the sample efficiency of MARL have been limited. Although the future seems very promising, like co-operation between LLM-driven robotic agent to do collective behaviors [7], Smart Society [8], etc. Those scenarios need large amount of agent cooperatively working together seamlessly to achieve object. The interaction between the agents, leading to exponentially growing sample complexity and computational burden. While prior studies have explored advantage estimation improvement [9], [10], off-policy corrections [11], [12], or optimize training paradigm techniques [12] to address inefficiency in single-agent settings, improving the sampling efficiency of MAPPO remains under-explored.

In this paper, we proposed *Selective Advantage Estimation* (SAE) to address the sample inefficiency issue of MAPPO. SAE is a simple drop-in replacement for the GAE that selectively amplifies better-than-baseline action while reducing the unwanted variance and bias. SAE uses a simple indicator function to select an action that is considered a good action, depending on the baseline, which helps the model focus on useful information while avoiding unwanted noise. We apply this method within the MAPPO paradigm under the Centralized Training with Decentralized Execution (CTDE) framework, using parallel rollout actors. The results on Multi-agent Particle Environments (MPE) and SMAC show that our method improves training speed with equal final performance compared to the baseline model, especially in simple homogeneous agent coordination settings.

Our main contribution in this paper is a selective advantage estimator that leverages ideas from the Self-Imitation Learning (SIL) and the Upgoing Policy Update (UPGO) within the currently popular on-policy training paradigm, MAPPO.

II. Related Work

1) Multi-Agent Reinforcement learning: Improving sample efficiency has been a central challenge in MARL. The Parallel Environment Data-Mixing Actor-Critic method [13] improves efficiency by mixing trajectories from parallel environments, though it relies heavily on careful implementation. Efficient Multi-agent Reinforcement Learning by Planning [14] uses centralized models and planning with optimistic search to reduce environment interaction costs, but faces scalability and model bias issues. Sample-Efficient Multi-agent Reinforcement Learning with Reset Replay [15] enhances off-policy learning by resetting replay buffers and augmenting data, achieving high reuse without forgetting. Together, these works illustrate complementary strategies data mixing, model-based planning, and replay augmentation for tackling the sample inefficiency problem in MARL.

2) MAPPO based advantage estimation: In MARL, improving sample efficiency often requires more effective credit assignment and advantage estimation strategies. Several recent studies have focused on modifying or augmenting the advantage function to address this challenge. One line of work introduced a noisy advantage regularization method in cooperative MARL, where Gaussian perturbations are applied to advantage values to mitigate overfitting and improve policy robustness in MAPPO-like frameworks [9]. Another approach proposed Difference Advantage Estimation, which refines the estimation of agent-specific advantages by leveraging difference-based baselines, leading to more accurate policy updates in multi-agent settings [16]. Beyond on-policy adaptations, some researchers explored multi-step counterfactual advantage estimation in an off-policy setup. This approach enables more efficient reuse of past experiences while preserving stable policy optimization, demonstrating gains in sample efficiency for cooperative tasks [10]. Complementary to this, another study proposed partial reward decoupling, which restructures the reward signal to provide agents with clearer and more targeted advantage information, thereby enhancing both credit assignment and learning efficiency [17].

III. Background

A. Multi-agent Reinforcement learning

The MARL problem is commonly formulated as a Markov game:

$$\mathcal{G} = \langle \mathcal{N}, \mathcal{S}, \{\mathcal{A}_i\}_{i\in\mathcal{N}}, \{r_i\}_{i\in\mathcal{N}}, P, \gamma \rangle$$

where $\mathcal{N} = \{1, \dots, N\}$ denotes the set of agents, $\mathcal{S}$ is the state space, $\mathcal{A}_i$ is the action space of agent i, $r_i(s, a)$ is the reward function for agent i, $P(s'|s, a)$ is the state transition probability, and $\gamma \in [0, 1)$ is the discount factor.

At each timestep t, all agents select actions $a_t = (a_t^1, \dots, a_t^N)$, the environment transitions to $s_{t+1} \sim P(\cdot|s_t, a_t)$, and each agent i receives reward $r_i^t = r_i(s_t, a_t)$.

For a joint policy $\pi = (\pi_1, \dots, \pi_N)$, the value functions for agent i are defined as:

$$V_i^\pi(s) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r_i(s_t, a_t) \,\middle|\, s_0 = s \right]$$

$$Q_i^\pi(o_i, a) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t \, r_i(s_t, a_t) \middle| o_{i,0} = o_i, a_0 = a \right]$$

In cooperative MARL, all agents share the same reward function, i.e., $r_i(s, a) = r_j(s, a), \forall i, j$, and the objective is to maximize the joint return. For homogeneous agents, the policies are shared, such that $\pi_i = \pi_j, \forall i, j$. In this paper, our method focuses on optimizing performance for homogeneous agents within a cooperative environment.

B. Multi-agent Proximal Policy Optimization

PPO is naturally extended to the multi-agent setting via the CTDE paradigm, resulting in MAPPO. In cooperative MARL with homogeneous agents, parameter sharing is commonly adopted, where all agents utilize a shared policy π_θ while maintaining decentralized execution. During execution, each agent i selects actions based solely on its local observation o_t^i.

During training, however, a centralized critic is employed to leverage global state information to stabilize learning. Specifically, MAPPO utilizes a centralized value function $V_\phi(s)$, where $s \in \mathcal{S}$ denotes the global state and ϕ represents the critic parameters.

The advantage function for agent i at timestep t is defined as:

$$A_t^i = Q_i^\pi(o_t^i, a_t^i) - V_\phi(s_t) \tag{1}$$

which is commonly estimated using GAE:

$$\hat{A}_t^i = \sum_{l=0}^{\infty} (\gamma\lambda)^l (r_{t+l} + \gamma V_\phi(s_{t+l+1}) - V_\phi(s_{t+l})) \tag{2}$$

where $\lambda \in [0, 1]$ controls the bias–variance trade-off.

MAPPO optimizes a clipped surrogate objective for each agent:

$$\begin{aligned} \mathcal{L}_{\text{PPO}}^i(\theta) = \mathbb{E}_t \Big[\min \Big(& r_t^i(\theta) \hat{A}_t^i, \\ & \text{clip}\big(r_t^i(\theta), 1-\epsilon, 1+\epsilon\big) \, \hat{A}_t^i \Big) \Big] \end{aligned} \tag{3}$$

where $r_t^i(\theta) = \frac{\pi_\theta(a_t^i|o_t^i)}{\pi_{\theta_{\text{old}}}(a_t^i|o_t^i)}$ is the importance sampling ratio with local observation, and ϵ is the clipping coefficient.

The overall policy objective is obtained by averaging over all agents:

$$\mathcal{L}_{\text{actor}}(\theta) = \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}_{\text{PPO}}^i(\theta)$$

The centralized critic is trained by minimizing the mean-squared error:

$$\mathcal{L}_{\text{critic}}(\phi) = \mathbb{E}_t \left[\left(V_\phi(s_t) - \hat{R}_t \right)^2 \right]$$

where $\hat{R}_t = \sum_{l=0}^{\infty} \gamma^l r_{t+l}$ denotes the empirical return.

The final MAPPO objective combines the actor loss, critic loss, and an entropy bonus:

$$\begin{aligned} \mathcal{L}_{\text{obj}}(\theta, \phi) = & \mathcal{L}_{\text{actor}}(\theta) - c_v \mathcal{L}_{\text{critic}}(\phi) \\ & + c_e \mathbb{E}_t \big[\mathcal{H}\big(\pi_\theta(\cdot \mid o_i^t)\big) \big] \end{aligned} \tag{4}$$

where c_v and c_e are weighting coefficients, and $\mathcal{H}(\cdot)$ denotes policy entropy. Under this CTDE framework, MAPPO effectively mitigates non-stationarity during training while preserving decentralized execution, making it a strong baseline for cooperative MARL.

IV. Proposed Method

A. Selective Advantage Estimation

At present, the MAPPO paradigm shares a fundamental issue with its single-agent counterpart: sample inefficiency. Although PPO brings significant improvements in stability in high-dimensional spaces especially for continuous action control and overall training stability these gains are substantial compared with earlier models. As an on-policy method, PPO requires fresh data trajectory after every policy update. In complex, high-dimensional environments, the sampling cost can be significantly higher than the training cost itself. For example, if an environment requires a large amount of computation, even with parallel environment, the GPU spends most of its time waiting for actor rollouts rather than computing gradients. This is exactly the issue we observed in our experiments: in the SMAC environment, the time spent on training versus trajectory collection is approximately 1:3. For current PPO-based LLM training paradigms, which often require months of training, improving sample efficiency would constitute a significant improvement.

The most commonly used PPO training paradigm, PPO-Clip, may suppress a beneficial action update when the update step is too large. The method we propose is designed specifically to address this issue. We incorporate fundamental principles from SIL [18] and UPGO [19] to overcome this limitation. Our approach uses the advantage function to identify high-quality actions and selectively increase their importance within trajectories, amplifying their influence by boosting their advantage values.

The selection process is implemented by modifying the bootstrapping mechanism, defined by the following formula:

$$m_t \triangleq \mathbb{I}[Q_\pi(o_{t+1}, a_{t+1}) \geq V_\phi(s_{t+1})] \tag{5}$$

where $\mathbb{I}[\cdot]$ is the indicator function. This selects actions that perform better than the baseline. Consequently, we define the selected Temporal Difference (TD) residual as:

$$\delta_t^{sel} = r_t + \gamma[\omega\, m_t\, \delta_{t+1}^{sel} + V_\phi(s_{t+1})] - V_\phi(s_t) \tag{6}$$

where $m_t \in \{0, 1\}$is a binary selection mask indicating whether the action at time *t* is considered beneficial, and ω is a hyperparameter that will be explained later.

When $m_t = 0$, the recursive term eliminate, and the selected TD residual reduces to the standard TD residual:

$$\delta_t^{\text{sel}} = r_t + \gamma V_\phi(s_{t+1}) - V_\phi(s_t) \tag{7}$$

When $m_t = 1$, the selected TD residual can be decomposed into a standard TD component and an additional recursive correction term:

$$\delta_t^{\text{sel}} = \big(r_t + \gamma V_\phi(s_{t+1}) - V_\phi(s_t)\big) + \gamma\omega\delta_{t+1} \tag{8}$$

This formulation ensures that temporal-difference bootstrap is always exist, while the action better than the average identified by the indicator function (5) are allowed to propagate their future TD errors backward through time. The hyperparameter ω controls the strength of this backward propagation, enabling selective amplification of high-quality action without uniformly increasing variance.

Based on the above, we define the Selective Advantage Estimation (SAE) in recursive form:

$$A_t^{\text{sel}(\gamma,\lambda,\omega)} = \delta_t^{sel} + \gamma A_{t+1}^{\text{sel}(\gamma,\lambda,\omega)} \tag{9}$$

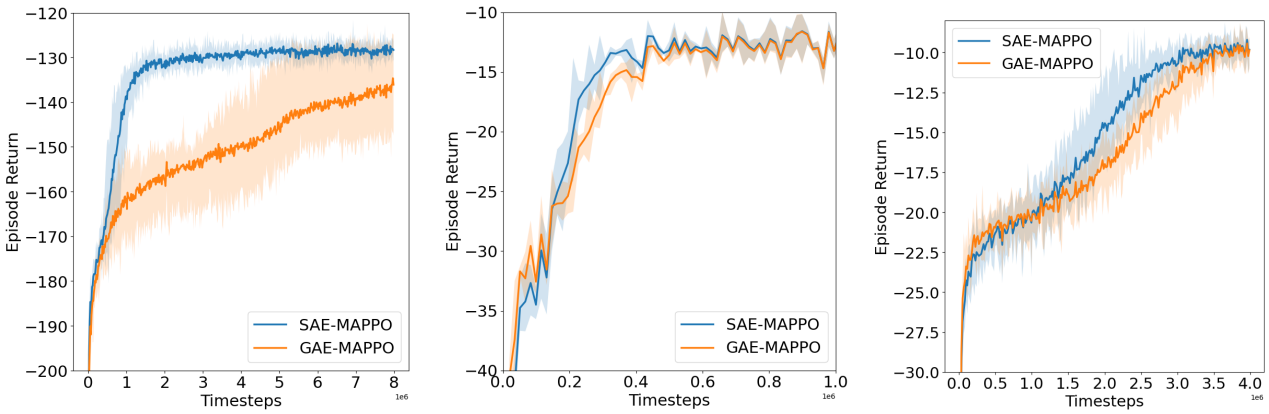
B. Algorithm

Based on the (9) We propose SAE-MAPPO, a MARL method built upon current MAPPO-clip paradigm. The algorithm follows the CTDE paradigm and improves sample efficiency by introducing Selective Advantage Estimation.

Algorithm 1 SAE-MAPPO

Initialize shared actor (decentralized policy) $\pi_\theta(a_i|o_i)$ and centralized critic $V_\phi(s)$ with random weights
Set PPO clip ϵ, discount γ, GAE parameter λ, actor lr α, critic lr β
Set number of rollout workers N_{actor}, rollout horizon $B_{\max}$, PPO epochs K, mini-batches M
while true **do**
  {repeat until time limit / convergence}
  for each parallel actor $j = 1, \ldots, N_{\text{actor}}$ **do**
    Collect a trajectory $\mathcal{T}_j$
    Store old log-probabilities $\log \pi_{\theta_{old}}(a_t^i|o_t^i)$ for all agents $i = 1, ..., N$
  end for
  $\mathcal{T} \leftarrow \bigcup_{j=1}^{N_{\text{actor}}} \mathcal{T}_j$
  for each timestep t in $\mathcal{T}$ and each agent $i = 1, ..., N$ **do**
    Compute selective advantage estimation $\hat{A}_t^{sel}$
  end for
  for $k = 1, \ldots, K$ **do**
    {PPO update epochs}
    Shuffle $\mathcal{T}$ and split into M mini-batches
    for each mini-batch $\mathcal{B}$ **do**
      For all $(t, i) \in \mathcal{B}$, compute ratio $r_t^i(\theta)$
      Compute objective function $\mathcal{L}_{obj}(\theta, \phi)$
      Update actor and critic parameters
    end for
  end for
  $\mathcal{T} \leftarrow \emptyset$ {clear buffer for next update}
end while=0

All agents share a decentralized policy $\pi_\theta(a_i \mid o_i)$. Each agent selects actions using only its local observation. During training, a centralized critic $V_\phi(s)$ is used to estimate the global state value. Access to global information allows the critic to provide more accurate value estimates and improves training stability.



(a) Simple spread (b) Simple speak listener (c) Simple reference

Fig. 1: Result of MPE environment

To improve data collection efficiency, SAE-MAPPO employs multiple parallel actors. Each actor interacts with an independent environment instance using the current policy and collects a trajectory. For every agent and timestep, we store the action and the corresponding log-probability under the behavior policy. All trajectories are merged into a single training buffer.

For each timestep t and each agent i, we compute selective advantage $\hat{A}_t^{\text{sel}}$. This estimator replaces standard GAE. This allows the policy to focus more on useful experiences while remaining fully on-policy. Policy optimization is performed using clip-PPO. The collected data are shuffled and divided into mini-batches. For each agent-timestep pair, we compute the probability ratio $r_t^i(\theta)$. The actor is updated using the clipped PPO objective weighted by the selective advantages, while the centralized critic is trained by minimizing the value prediction error. This update is repeated for multiple epochs. The above procedure is repeated until convergence or a predefined training limit is reached. SAE-MAPPO preserves the simplicity and stability of MAPPO while improving learning efficiency by emphasizing better-than-average actions.

V. Experimental Evaluation

In this section, we present a comprehensive empirical evaluation of the proposed Selective Advantage Estimation method. To verify its effectiveness in improving sample efficiency and accelerating convergence, we compare SAE-MAPPO against the standard GAE-MAPPO baseline.

A. Experiment Setup

We evaluate the effectiveness of our proposed method SAE directly with current state-of-the-art GAE-MAPPO training paradigm. Experiments are conducted on two widely used multi-agent reinforcement learning benchmarks: the Multi-Agent Particle Environment (MPE) and the StarCraft Multi-Agent Challenge (SMAC).

In the MPE benchmark, we focus on cooperative tasks and consider both homogeneous and heterogeneous agent settings. All results on MPE are averaged over 7 random seeds, and training is performed until the agents reach performance convergence.

In SMAC, we evaluate our method by competing against the built-in AI scripts base enemy, testing both homogeneous and heterogeneous agent under symmetric and asymmetric combat scenarios. Results are averaged over 10 random seeds, with all agents trained for 10 million time steps, which is sufficient for convergence across all evaluated scenarios. For each environment, hyperparameter are tuned to achieve the best performance, following common practices in prior work. To ensure a fair comparison, the same hyperparameter settings are used consistently across all compared methods.

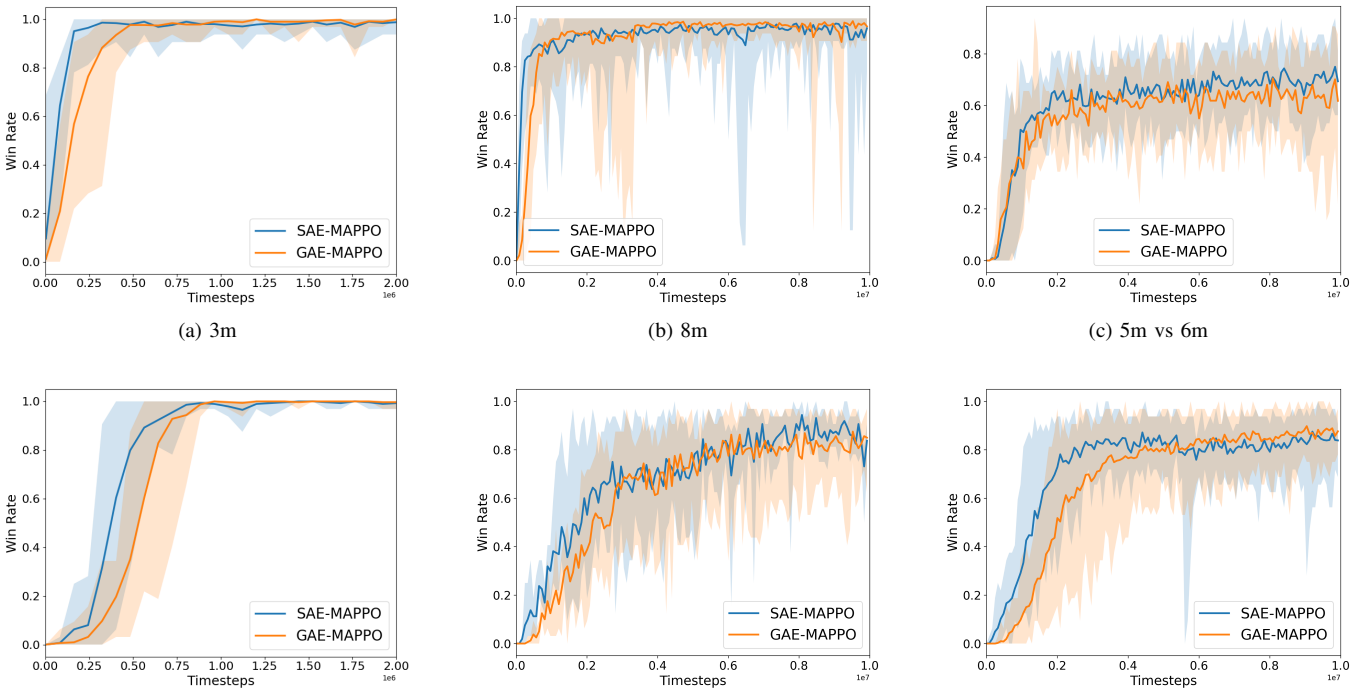
B. MPE

The MPE is a widely used benchmark for cooperative and competitive MARL. It consists of a set of simple particle-based agents interacting in a continuous 2D world, where agents must coordinate their actions based on local observations to achieve collective targets.

We evaluate SAE on three standard tasks from the MPE, which are designed to assess cooperative control under partial observability. The tasks include homogeneous cooperation (Simple Spread), communication only (Simple Speaker–Listener), and heterogeneous cooperation (Simple Reference). Fig. 1 reports the mean episode return averaged over multiple random seeds, with shaded regions indicating Min-Max range.

The Simple Spread task requires multiple homogeneous agents to jointly cover all landmarks while avoiding collisions. As shown in Fig.1a, our method converges faster than baseline, achieving higher episode returns in the early stage of training. Moreover, our method reaches a better final performance, indicating more effective cooperation among agents. These results suggest that the proposed advantage function enhancement reduces overly conservative policy updates and accelerates the training process.

In the Simple Speaker-Listener task, a speaker agent sends a message to guide a listener under partial observability,



(a) 3m (b) 8m (c) 5m vs 6m

(d) 3s vs 3z (e) 10m vs 11m (f) 8m vs 9m

Fig. 2: Results on SMAC environments

requiring effective communication between agents. As shown in Fig. 1b, both methods converge quickly, but our method reaches near-optimal performance earlier.

The Simple Reference task involves heterogeneous agents with different observations, requiring accurate coordination and work assignment. As shown in Fig. 1c, our method consistently achieves higher returns than baseline throughout training. The performance gap is most noticeable in the middle stage of training, indicating that is our method more effectively utilizing high-quality trajectories in heterogeneous cooperative settings.

Overall, our method improves sample efficiency and convergence speed across all three MPE tasks while maintaining stable training behavior. The performance gains are better in tasks that require stronger coordination and credit assignment, demonstrating the effectiveness of selectively amplifying advantageous action in multi-agent scenarios.

C. SMAC

We further evaluate SAE-MAPPO on several scenarios from the SMAC, which focuses on decentralized control with partial observability and varying degrees of coordination difficulty. Fig. 2 reports the win rate during training for homogeneous agents under symmetric and asymmetric combat scenarios, with Min-Max range.

The *3m* and *8m* scenarios involve symmetric battles with homogeneous agents, serving as relatively simple coordination benchmarks. As shown in Fig. 2a and Fig. 2b , both methods quickly achieve near-perfect win rates. our method converges slightly faster in the early stage of training, while both methods reach comparable final performance. This indicates that our method does not sacrifice performance in simple settings. Although it may introduce slight instability to the model.

The *5m vs 6m*, *8m vs 9m*, and *10m vs 11m* scenarios evaluate asymmetric combat settings with increasing coordination difficulty. Among them, *5m vs 6m* is the most challenging due to the high ratio of enemies to allies, which significantly increases the difficulty of coordination and decision making. As shown in Fig. 2c, our method consistently achieves a higher win rate than baseline throughout training. For the *8m vs 9m* and *10m vs 11m* scenarios, which involve tighter coordination constraints and larger team sizes, our method demonstrates faster convergence and higher win rates during the early stages of training, as shown in Fig. 2f and Fig. 2e. Although the win-rate gap narrows in later stages in training process, our method still maintains a strong learning speed.

The *3s vs 3z* scenario tests micro-management skills, particularly kiting, in which agents must coordinate movement and attack timing to avoid damage. As shown in Fig. 2d, our method reaches a high win rate faster than baseline, while both methods achieve near-optimal performance after convergence.

Table I shows that SAE consistently reaches early and mid-stage performance milestones (50%, 70%) faster than GAE across most scenarios, with speedups up to 71%. This demonstrates that our method improves sample efficiency during early stage of training. Suggesting our method do improved optimization efficiency rather than delayed convergence. Overall, our method consistently improves convergence speed by

TABLE I: Training speed comparison between SAE and GAE at different win-rate milestones. Speedup is defined as $(T^{\text{GAE}} - T^{\text{SAE}})/T^{\text{GAE}}$. Positive values indicate that SAE reaches the milestone earlier.

Scenario	50% of Max Win Rate			70% of Max Win Rate			90% of Max Win Rate		
	$T^{\text{GAE}}(1e^6)$	$T^{\text{SAE}}(1e^6)$	Speedup (%)	$T^{\text{GAE}}(1e^6)$	$T^{\text{SAE}}(1e^6)$	Speedup (%)	$T^{\text{GAE}}(1e^6)$	$T^{\text{SAE}}(1e^6)$	Speedup (%)
3m	0.16	0.08	**49.0**	0.24	0.16	**32.9**	0.40	0.16	**59.5**
3s vs 3z	0.56	0.40	**28.4**	0.64	0.48	**24.9**	0.72	0.64	**11.1**
5m vs 6m	0.88	0.96	-9.1	1.12	1.12	0.0	2.40	2.64	-10.0
8m	0.40	0.16	**59.5**	0.56	0.16	**71.0**	0.80	0.64	**19.9**
8m vs 9m	2.00	1.20	**39.9**	2.56	1.60	**37.5**	4.48	2.48	**44.6**
10m vs 11m	2.16	1.28	**40.7**	2.80	2.48	**11.4**	4.96	6.56	-32.2

increasing sample efficiency across SMAC scenarios. These results indicate that it can maintain its improvement in a more challenging environment. Furthermore, the real-world training time has been reduced about 2% for each task in the SMAC environment.

D. Discussion

Our method is conceptually related to Self-Imitation Learning and Upgoing Policy Update, both of which utilize high-quality experiences. However, unlike these approaches, SAE is designed to operate within an on-policy framework and does not rely on replay buffers, off-policy corrections, or trajectory-level filtering. Compared to prior methods that modify the advantage function, our easily integrable method can be directly apply into the training paradigm without introducing additional neural networks, auxiliary objectives, or data-gathering processes. SAE serves as a method to improve sample efficiency while maintaining the simplicity inherent in on-policy models.

The core motivation of SAE is to address the uniform treatment of action in standard GAE. In conventional GAE, all action are treated equally, regardless of whether the corresponding actions are beneficial to the training process or not. This uniform treatment may dilute informative learning signals and suppress large but meaningful policy updates, particularly under any clipping mechanism. As a result, the policy may become overly conservative even when encountering high-quality actions.

SAE addresses this issue by selectively amplifying the impact of actions only when they outperform the baseline. By introducing the indicator function, SAE can amplify the influence of good actions while reducing unnecessary variance and bias amplification from suboptimal actions. This process improves sample efficiency by prioritizing informative experiences rather than uniformly reweighting all on-policy samples across the trajectory.

Despite its effectiveness in improving sample efficiency, SAE has several limitations. First, the selection mechanism relies on a fixed binary threshold dependent on the value function, which may lead to instability in the late stages of training. Additionally, the amplification strength is controlled by a single static hyperparameter; thus, SAE shares a common drawback with other hyperparameter-based methods, requiring tuning for optimal performance. Furthermore, regarding heterogeneous agent cooperation, the effectiveness of our method can be significantly affected within the current CTDE paradigm. Since actions vary across trajectories, a beneficial action for one type of agent may not be suitable for another, yet both are evaluated by the same critic network that treats all agents identically.

VI. Conclusion

We proposed the SAE method to address the sample inefficiency issue in the current MAPPO training paradigm by incorporating an action selection mechanism that emphasizes actions leading to better states. By introducing an indicator function into the bootstrapping process, SAE selectively amplifies actions that improve the agent's training trajectory. Experiments on MPE and SMAC demonstrate that our method achieves faster learning speeds through enhanced sample efficiency while maintaining comparable final performance. Despite these gains, SAE relies on a fixed threshold and a static amplification hyperparameter, which can introduce late-stage instability and necessitate hyperparameter tuning, particularly in heterogeneous agent tasks.

Future work will explore adaptive or probabilistic selection mechanisms, agent-type-specific baselines, and extensions to mixed cooperative-competitive settings. Furthermore, integrating SAE with multiple baselines for heterogeneous agents and applying it to large-scale real-world systems—such as PPO-based Large Language Model training—represent promising directions for improving sample efficiency in high-cost training regimes.

References

[1] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, and e. Agarwal, "Training language models to follow instructions with human feedback," in *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, Eds., vol. 35. Curran Associates, Inc., 2022, pp. 27 730–27 744.

[2] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms," 2017. [Online]. Available: https://arxiv.org/abs/1707.06347

[3] K. Zhang, D. Zhu, Q. Xu, H. Zhou, and C. Zheng, "Pps-qmix: Periodically parameter sharing for accelerating convergence of multi-agent reinforcement learning," 2024. [Online]. Available: https://arxiv.org/abs/2403.02635

[4] A. L. Bazzan, "Opportunities for multiagent systems and multiagent reinforcement learning in traffic control," *Autonomous Agents and Multi-Agent Systems*, vol. 18, no. 3, pp. 342–375, 2009.

[5] O. Vinyals, T. Ewalds, S. Bartunov, P. Georgiev, A. S. Vezhnevets, M. Yeo, A. Makhzani, H. Küttler, J. Agapiou, J. Schrittwieser *et al.*, "Starcraft ii: A new challenge for reinforcement learning," *arXiv preprint arXiv:1708.04782*, 2017.

[6] J. Schulman, P. Moritz, S. Levine, M. I. Jordan, and P. Abbeel, "High-dimensional continuous control using generalized advantage estimation," in *International Conference on Learning Representations (ICLR)*, 2016.

[7] J. De Curtò and I. De Zarzà, "Llm-driven social influence for cooperative behavior in multi-agent systems," *IEEE Access*, 2025.

[8] W. Chen, Y. Su, J. Zuo, C. Yang, C. Yuan, C.-M. Chan, H. Yu, Y. Lu, Y.-H. Hung, C. Qian *et al.*, "Agentverse: Facilitating multi-agent collaboration and exploring emergent behaviors," in *The Twelfth International Conference on Learning Representations*, 2023.

[9] J. Hu, S. Hu, and S.-w. Liao, "Policy regularization via noisy advantage values for cooperative multi-agent actor-critic methods," *arXiv preprint arXiv:2106.14334*, 2021.

[10] S. Kim, W. Kim, J. Jeon, Y. Sung, and S. Han, "Off-policy multi-agent policy optimization with multi-step counterfactual advantage estimation," in *The 22nd International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2023*. AAMAS Workshop, 2023.

[11] M. Zawalski, B. Osiński, H. Michalewski, and P. Miłoń, "Off-policy correction for multi-agent reinforcement learning," in *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*, ser. AAMAS '22. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2022, p. 1774–1776.

[12] Y. Zhao, Z. Yang, Z. Wang, and J. D. Lee, "Local optimization achieves global optimality in multi-agent reinforcement learning," in *International Conference on Machine Learning*. PMLR, 2023, pp. 42 200–42 226.

[13] Z. Ye, Y. Chen, X. Jiang, G. Song, B. Yang, and S. Fan, "Improving sample efficiency in multi-agent actor-critic methods," *Applied Intelligence*, vol. 52, no. 4, pp. 3691–3704, 2022.

[14] Q. Liu, J. Ye, X. Ma, J. Yang, B. Liang, and C. Zhang, "Efficient multi-agent reinforcement learning by planning," in *12th International Conference on Learning Representations, ICLR 2024*, 2024.

[15] Y. Yang, G. Chen, P.-A. Heng *et al.*, "Sample-efficient multiagent reinforcement learning with reset replay," in *Forty-first international conference on machine learning*, 2024.

[16] Y. Li, G. Xie, and Z. Lu, "Difference advantage estimation for multi-agent policy gradients," in *International Conference on Machine Learning*. PMLR, 2022, pp. 13 066–13 085.

[17] A. Kapoor, B. Freed, J. Schneider, and H. Choset, "Assigning credit with partial reward decoupling in multi-agent proximal policy optimization," in *Reinforcement Learning Conference*, 2024.

[18] J. Oh, Y. Guo, S. Singh, and H. Lee, "Self-Imitation Learning," in *International conference on machine learning*. PMLR, 2018, pp. 3878–3887.

[19] O. Vinyals, I. Babuschkin, W. M. Czarnecki, M. Mathieu, A. Dudzik, J. Chung, D. H. Choi, R. Powell, T. Ewalds, P. Georgiev *et al.*, "Grandmaster level in StarCraft II using multi-agent reinforcement learning," *nature*, vol. 575, no. 7782, pp. 350–354, 2019.