Policy-Level Exploration for Coordinated Multi-Agent Reinforcement Learning

Yucong Zhang
Sun Yat-sen University
zhangyc8@mail2.sysu.edu.cn

Chao Yu
Sun Yat-sen University
yuchao3@mail.sysu.edu.cn

Abstract

In multi-agent reinforcement learning (MARL), agents must not only discover high-reward behaviours but also reliably reproduce coordinated behaviours over time. A fundamental challenge arises from a mismatch between action-level stochastic exploration and temporally extended coordination. To address this issue, we propose a policy-level exploration framework with Deterministic Action Execution (DAE) for MARL, which shifts locally random action exploration to temporally extended policy selection via an option-based exploration mechanism. Specifically, at each time step, each agent selects a policy from a shared policy set—comprising one exploitation policy and multiple exploration policies—using a learned option policy, and actions are executed deterministically conditioned on the selected policy. By maintaining deterministic execution under temporally extended policy selection, DAE facilitates temporally coordinated behaviours across multiple time steps. Comprehensive experiments on five multi-agent tasks—Predator Prey, StarCraft II micro management challenge (SMAC), SMACv2, Google Research Football, and Multi-Agent Coordination benchmark (MACO)—demonstrate that DAE outperforms mainstream baselines in terms of sample efficiency and final learning performance.

1 Introduction

In cooperative multi-agent reinforcement learning (MARL) [27, 40, 19], agents must not only discover high-reward behaviours but also reliably reproduce coordinated behaviours over time. Most existing MARL algorithms rely on action-level stochastic exploration strategies, such as ϵ-greedy [31, 32, 23], softmax exploration [41], Thompson Sampling [12], and action noise [29], where non-greedy actions are selected randomly at each decision step. While these strategies can encourage diverse behaviours, the randomness at each step prevents agents from reliably reproducing temporally coordinated behaviours. A mismatch thus arises between action-level stochastic exploration and temporally extended coordination, which greatly hinders the learning efficiency in common MARL algorithms.

To address this limitation, we propose a policy-level exploration framework with Deterministic Action Execution (DAE) for MARL, which shifts locally random action exploration to temporally extended policy selection for each agent. Specifically, DAE adopts an option-based exploration mechanism, in which each agent, at each time step, selects a policy from a shared policy set—comprising one exploitation policy and multiple exploration policies—using a learned option policy. The selected policy is then greedily executed using the agents' local observations. As for the policies in the shared set, the exploitation policy is optimized for maximizing the cumulative team return, while each exploration policy is optimized to maximize its divergence from both the exploitation policy and the other exploration policies, ensuring that different exploration policies explore toward distinct exploration directions. The option policy coordinates policy selection from the shared policy set across agents, thereby enabling dynamic balance between exploration and exploitation. By maintaining deterministic execution under temporally extended policy selection, DAE facilitates temporally

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

coordinated behaviours across multiple time steps. However, sparse reward feedback may cause the option policy to over-select policies that occasionally obtain rewards, thereby degrading exploration efficiency. To mitigate this, DAE incorporates stage-dependent auxiliary reward signals by accounting for the novelty of state-action pairs, the fluctuation of reward dynamics, and the correlation between states and the task goal.

To validate the effectiveness of DAE, we conduct empirical experiments on several challenging multi-agent tasks: Predator Prey [30], SMAC [36], SMACv2 [9], Google Research Football [18], and MACO [38]. Experimental results demonstrate that DAE can improve both sample efficiency and final performance compared to mainstream baselines. Further analyses demonstrate that policy-level exploration substantially facilitates temporally extended coordination, while the auxiliary exploration signals further enhance exploration efficiency.

2 Related Work

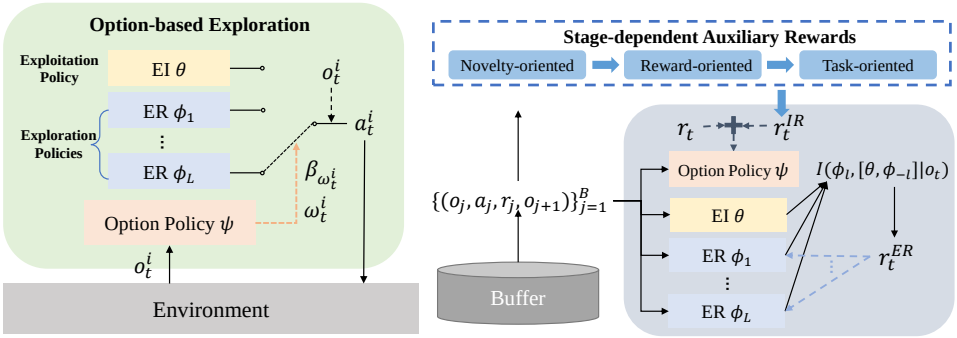
Effectively trading off exploration and exploitation is a challenging issue in MARL research [14, 25, 10] due to the exponential growth of the joint state-action space [39, 45, 5], and the necessity for coordinated exploration [35, 20] among agents. A large number of works focus on curiosity-driven exploration where agents are encouraged to explore novel states or dynamics [17, 22, 43, 15]. Several approaches further incorporate coordination into intrinsic rewards. EMC [44] leverages prediction errors to capture joint historical information, while MACE [16] coordinates exploration by communicating local novelty among the agents. More recently, diversity-driven exploration mechanisms [13, 34, 2, 11] have emerged as an effective strategy in MARL. Representative methods such as CDS [4], COIN [21], and MANGER [3] encourage diverse behaviours via intrinsic rewards or controlled updates. Furthermore, policy-based MARL methods [41, 6] introduce stochasticity through action sampling from learned policies and have demonstrated strong empirical performance in many cooperative tasks. Several randomized exploration strategies have been studied in reinforcement learning and extended to MARL, including upper confidence bound (UCB) [8] and Thompson sampling [12]. Although effective for improving exploration efficiency, most of them still rely on action-level stochasticity as the underlying exploration mechanism. In contrast, the proposed framework shifts exploration from the action level to the policy level via an option-based exploration mechanism, thereby enabling temporally coordinated exploration.

In addition, in single-agent reinforcement learning, Bootstrapped DQN [28] guides exploration via an ensemble of value functions. However, Bootstrapped DQN samples value functions from a uniform distribution, while the selected value function still relies on ϵ-greedy exploration to select actions during interaction with the environment, indicating that exploration remains stochastic at the action level. In contrast, DAE performs policy-level exploration by using a learned option policy to select from a policy set, while actions are executed deterministically conditioned on the selected policy. SMMAE [42] learns an agent-specific exploration policy and an adaptive exploration probability for each agent to encourage visits to underexplored states. However, SMMAE uses an -greedy-like exploration mechanism: at every timestep, each agent independently chooses whether to execute its exploration or cooperation policy. Such step-wise random switching may interrupt temporally extended coordinated behaviors. DAE instead selects a policy from a shared policy set and deterministically executes it across consecutive timesteps until a learned termination occurs. MADPO [7] promotes exploration and behavioral heterogeneity by maximizing policy divergence across agents and consecutive policy updates. However, each agent maintains only a single behavioral policy, and MADPO does not select among multiple exploration behaviors within a trajectory or learn how long a selected behavior should persist. In contrast, DAE explicitly learns both policy selection and termination, enabling a selected exploration behavior to be maintained over multiple timesteps.

3 Preliminary

3.1 Dec-POMDP

A cooperative multi-agent task can be formalized as a decentralized partially observable markov decision process (Dec-POMDP) [26], which consists of an eight-tuple $G = \langle I, S, A, P, R, O, n, \gamma \rangle$ where I is a set of agents, n is the maximum number of agents in the environment. S and A represent the joint state space and the joint action space for all agents, respectively. The transition function P



(a) The interaction process (b) The training process

Figure 1: The architecture of DAE. (a) At every timestep, each agent uses the option policy ψ to select a policy from the exploitation policy (EI, θ) or the exploration policies (ER, ϕ), and then executes an action a_t^i greedily based on its local observation o_t^i and the selected policy. (b) The upper panel illustrates the stage-dependent auxiliary rewards r_t^{IR}, which are incorporated into the learning of the option policy to promote the agents' exploration. The lower panel depicts the training of all policies, where the option policy is optimized to maximize the sum of environmental reward r_t and r_t^{IR}, the exploitation policy θ is trained to maximize the environmental reward, and each exploration policy ϕ_l is optimized to diverge from both the exploitation policy and the other exploration policies ϕ_{-l}.

describes the environment dynamics that define the state transition and reward transition. R is the reward function, which delivers a team reward for the agents in cooperative environments. At time step t, the environment supplies the local observation $o_t \in O$ to the agents. Then, agent i based on its own observation o_t^i selects an action a_t^i via its policy function $\pi(a_t^i|o_t^i)$. According to the state $s \in S$ and the joint action of all agents $(a_t^1, \cdots, a_t^n)$, the environment will move to the next time step $t+1$ through the transition function while both the team reward $r_t \in R$ and the next observation o_{t+1} will be sent to all agents. The learning goal of the agents is to maximize the cumulative discounted team reward $\sum_{t=0}^{T}[\gamma^t r_t]$ where T is the maximum time step.

3.2 The Option-critic Architecture

The option-critic architecture [1], which is proposed to achieve temporal abstraction over actions in deep RL, consists of three main modules: an option policy ψ, a termination function β, and a set of options I_ω. At time step t, the option policy ϕ selects a new option ω_t given the current state s_t while the termination function outputs a boolean value $\hat{\beta}_t \in \{0, 1\}$. The current option used by the agent to interact with the environment is the new option ω_t if $\hat{\beta}_t = 1$, and the previous option ω_{t-1} otherwise.

4 Method

This section describes the proposed DAE framework. As illustrated in Figure 1, the key idea is to perform exploration at the policy level via option selection, while stage-dependent auxiliary reward signals are incorporated to facilitate learning.

4.1 Policy-Level Exploration

In DAE, an option policy ψ is used to decide, at each time step, whether the agents interact with the environment using the exploitation policy θ or one of the L exploration policies $(\phi_1, \cdots, \phi_L)$. To enable temporally extended behaviours, the selected option persists across multiple time steps until termination.

The option policy is implemented via a neural network that takes the agent's local observation o_t^i as inputs and outputs both a policy identity $\omega_t^i \in \{ID_\theta, ID_{\phi_1}, \cdots, ID_{\phi_L}\}$ and a termination probability distribution $\beta_{\omega_t^i}$:

$$\omega_t^i, \beta_{\omega_t^i} = \pi_\psi(o_t^i), i = 1, \cdots, N. \tag{1}$$

To eliminate randomness at the action level, both the exploitation policy and the exploration policies adopt the greedy action selection:

$$a_t^i = \begin{cases} Greedy(\pi_\theta(o_t^i)), & \textit{Exploitation} \\ Greedy(\pi_{\phi_l}(o_t^i)), & \textit{Exploration}. \end{cases} \tag{2}$$

where $l \in [1, \cdots, L]$.

At each time step t, agent i first feeds its local observation o_t^i into the option policy network ψ, receiving both the policy identity ω_t^i and the termination probability distribution $\beta_{\omega_t^i}$. A termination value $\hat{\beta}_t^i$ is then sampled from the Bernoulli distribution $\beta_{\omega_t^i}$. The interaction action a_t^i of agent i is determined by the previous policy $\pi_{\omega_{t-1}^i}$ if $\hat{\beta}_t^i$ equals zero; otherwise, it follows the current interaction policy $\pi_{\omega_t^i}$:

$$a_t^i = \begin{cases} Greedy(\pi_\theta(o_t^i)), & \text{if}(\hat{\beta}_t^i = 0 \; and \; \omega_{t-1}^i = \textit{ID}_\theta)\text{or}(\hat{\beta}_t^i = 1 \; and \; \omega_t^i = \textit{ID}_\theta) \\ Greedy(\pi_{\phi_l}(o_t^i)), & \text{otherwise.} \end{cases} \tag{3}$$

4.2 Policy Learning

Training the Option Policy. Following the centralized training with decentralized execution paradigm, the option policy is trained using a centralized value function, while execution relies solely on local observations. The learning goal of the option policy is to maximize the cumulative discounted team reward, thereby achieving explicit coordinated exploration among the agents. The corresponding loss function for the option policy is defined as:

$$\mathcal{L}_\omega = \mathbb{E}_D \left[(r_t + \gamma Q_{\psi'}^{tot} - \sum_{i=1}^{N} Q_\psi(o_t^i, \omega_t^i))^2 \right], \tag{4}$$

$$Q_{\psi'}^{tot} = \sum_{i=1}^{N} \left[(1 - \hat{\beta}_{t+1}^i) Q_{\psi'}(o_{t+1}^i, \omega_{t+1}^i) + \hat{\beta}_{t+1}^i \max_{\omega_{t+1}^i} Q_{\psi'}(o_{t+1}^i, \cdot) \right], \tag{5}$$

where $\hat{\beta}_{t+1}^i = Bernoulli(\beta_{\omega_{t+1}^i})$, ψ' is the target network for ψ, D is a batch of data sampled from the replay buffer, and r_t is the reward. The loss function for $\beta_{\omega_t^i}$ is:

$$\mathcal{L}_\beta = \mathbb{E}_D \left[\sum_{i=1}^{N} \hat{\beta}_t^i (Q_{\psi'}(o_{t+1}^i, \omega_{t+1}^i) - \max_{\omega_{t+1}} Q_{\psi'}(o_{t+1}^i, \cdot)) \right]. \tag{6}$$

Note that, the learning goal of the exploration policies is to encourage agents to explore unknown states in the environment, which becomes progressively less beneficial for the agents' learning as the state space becomes sufficiently explored. Consequently, as the exploration process progresses, the option policy gradually increases the probability of selecting the exploitation policy, and thus the exploitation policy becomes the unique interaction policy for the agents at the end of the learning process. Unlike methods based on ensemble learning that require storing all policy models, DAE only reserves the exploitation policy model after learning, thus significantly conserving memory space. To more efficiently balance the exploration and exploitation in the option policy, we introduce a regularization term into the learning objective of the option policy:

$$\mathcal{L}_{re} = \mathbb{E}_D \left[\textit{CE}(\textit{ID}_\theta, \omega_t^i) \right], \tag{7}$$

where CE is the cross-entropy loss that reflects the option policy's tendency toward exploitation. Overall, the loss function of the option policy is defined as:

$$\mathcal{L}_\psi = \mathcal{L}_\omega + \mathcal{L}_\beta + \alpha_1 \mathcal{L}_{re}, \tag{8}$$

where α_1 is a hyperparameter.

Training the Exploitation Policy. The learning objective of the exploitation policy is to maximize the cumulative discounted team rewards as follows:

$$\mathcal{L}_\theta = \mathbb{E}_D \left[(r_t + \gamma \max_{\boldsymbol{a}_{t+1}} Q_{\theta'}^{tot}(\boldsymbol{o}_{t+1}, \cdot) - Q_\theta^{tot}(\boldsymbol{o}_t, \boldsymbol{a}_t))^2 \right], \tag{9}$$

where θ' is the target network for θ, $\boldsymbol{o}$ and $\boldsymbol{a}$ represent the joint observation and joint action, respectively, and Q^{tot} is the joint Q-value function.

Training the Exploration Policies. Each exploration policy ϕ_l aims to maximize its distance from both the exploitation policy θ and other exploration policies ϕ_{-l} via the mutual information maximization so that different exploration policies induce distinct exploration behaviours, thereby achieving a more efficient exploration:

$$max\ I(\phi_l; [\theta, \phi_{-l}] | \boldsymbol{o}_t). \tag{10}$$

In implementation, we maximize the mutual information between the actions produced by network ϕ_l and the identity of the networks $[\theta, \phi_{-l}]$ given $\boldsymbol{o}_t$:

$$I(\pi_{\phi_l}; ID | \boldsymbol{o}_t) \geq \mathbb{E}_{ID, \boldsymbol{o}, \boldsymbol{a}} \left[log \frac{q(\pi_{\phi_l} | ID, \boldsymbol{o}_t)}{p(\pi_{\phi_l} | \boldsymbol{o}_t)} \right], \tag{11}$$

where $ID = \{ID_\theta, ID_{\phi_{-l}}\}$. The lower bound of this mutual information is used as the reward r_t^{ER} for the learning of each exploration policy as follows:

$$\mathcal{L}_{\phi_l} = \mathbb{E}_D \left[(r_t^{ER} + \gamma \max_{\boldsymbol{a}_{t+1}} Q_{\phi_l'}^{tot}(\boldsymbol{o}_{t+1}, \cdot) - Q_{\phi_l}^{tot}(\boldsymbol{o}_t, \boldsymbol{a}_t))^2 \right], \tag{12}$$

where ϕ_l' is the target network for ϕ_l. Notably, we train only one exploration policy per training epoch to reduce instability in the network optimization of the exploration policies. Refer to Appendix A.3 for more details about r_t^{ER}. To ensure implementation simplicity and reduce the total number of trainable parameters, we employ shared network architectures for the option policy network, the exploitation policy network, and the exploration networks across all agents.

4.3 Stage-dependent Auxiliary Rewards

Although policy-level exploration facilitates temporally extended coordination, sparse reward feedback may cause the option policy to over-select policies that occasionally obtain rewards, thereby degrading exploration efficiency. To address this challenge, we incorporate stage-dependent auxiliary rewards that provide exploration guidance during different stages of learning. These rewards are designed to address different exploration requirements throughout learning, including broad coverage of uncertain state-action regions during early exploration, sensitivity to informative reward variations during policy refinement, and exploration toward behaviours correlated with task completion.

Specifically, a novelty-oriented signal is defined to encourage exploration of under-visited state-action pairs:

$$r_t^{nov} = Dist\big(\pi(a|o_t) \parallel \hat{\pi}(a|o_t)\big), \tag{13}$$

where $Dist$ is the l_2 distance between the true action distribution $\pi(a|o_t)$ and the predicted action distribution $\hat{\pi}(a|o_t)$ that is obtained from a prediction model.

To capture reward dynamics, a reward-oriented signal is defined based on the prediction error of the value function:

$$r_t^{rew} = max_t \ \left(Q_\theta(o_t, a_t) - Q_{\hat{\theta}}(o_t, a_t)\right)^2. \tag{14}$$

where $Q_{\hat{\theta}}$ is a predictor of Q_θ.

To reflect task relevance, a task-oriented signal is defined based on the mutual information between states and task completion signals $\mathbb{I}_{task}$:

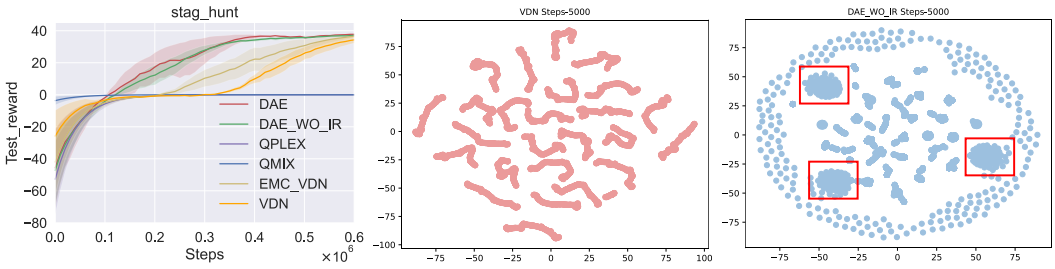
$$min\ -I(\mathbb{I}_{task}; Z) + I(Z; s_t)\ ,\ r_t^{ta} = D_{KL}\left(p(Z|s_t) || p'(Z)\right). \tag{15}$$

where s_t is the global state, Z is the latent representation, $p'(Z)$ is the prior distribution of Z.

Then, the overall reward is defined as:

$$r_t^{IR} = \begin{cases} \alpha_2 r_t^{nov} + \alpha_3 r_t^{ta}, & \text{if } \delta_t <= \xi \\ \alpha_2 r_t^{rew} + \alpha_3 r_t^{ta}, & \text{if } \delta_t > \xi, \end{cases} \tag{16}$$

where α_2 and α_3 are hyperparameters to control the magnitude of the exploration, ξ is a hyperparameter for distinguishing between different exploration stages, and $\delta_t = |r_t^{nov} - r_{t-1}^{nov}|$, which reflects the change in novelty prediction error between consecutive timesteps and is used to distinguish between different exploration stages. More details about stage-dependent auxiliary rewards can be found in Appendix A.4.



(a) Test reward curves (b) Visualization for VDN (c) Visualization for DAE_WO_IR

Figure 2: Numerical results and visualization on the Predator Prey domain

The resulting auxiliary reward r_t^{IR} is then incorporated into the optimization of the option policy. Therefore, Equation 4 is modified as follows:

$$\mathcal{L}_\omega = \mathbb{E}_D \left[(r_t^{IR} + r_t + \gamma Q_{\psi'}^{tot} - \sum_{i=1}^{N} Q_\psi(o_t^i, \omega_t^i))^2 \right]. \tag{17}$$

5 Experiments

To validate the effectiveness of policy-level exploration, we evaluate DAE across multiple cooperative MARL benchmarks with varying levels of coordination difficulty: (1) Predator Prey [30], (2) SMAC [36], (3) SMACv2 [9], (4) Google Research Football [18], and (5) MACO [38]. We compare DAE with baselines that use the $\epsilon - greedy$ exploration strategy, comprising three mainstream MARL baselines (VDN [33], QMIX [31], and QPLEX [37]) and five popular multi-agent exploration baselines (MAVEN [24], CDS [4], EMC [44], COIN [22], and MANGER [3]). Unless otherwise specified, we use QMIX [31] as the backbone for training the exploitation policy in DAE. Since both CDS and EMC are originally based on the QPLEX framework, we implement their QMIX-based variants, i.e., CDS_QMIX and EMC_QMIX, to ensure a fair evaluation. In addition, we compare DAE with two policy-based baselines (MAPPO [41] and IPPO [6]) using the softmax exploration. All experiments are conducted across five independent runs with different random seeds to ensure statistical reliability.

5.1 Predator Prey

In the Predator Prey task, 8 predators controlled by the learning agents must cooperatively capture 8 rule-based prey in a 10×10 grid environment. A positive reward is granted only when two or more predators simultaneously capture prey, while a penalty is imposed when a prey is captured by a single predator. Due to the limited representational capacity of the mixing network in QMIX, both DAE and EMC adopt VDN as the underlying value decomposition architecture. To illustrate the advantage of the policy-level exploration framework, we remove the three auxiliary rewards from DAE (denoted as DAE_WO_IR). As shown in Figure 2 (a), both DAE and DAE_WO_IR successfully complete the task within merely 400K training steps, exhibiting an approximately 30% faster learning convergence compared to most baselines.

To further analyze the influence of $\epsilon - greedy$ exploration versus the policy-level exploration on the agents' behaviour, we visualize the state distributions of VDN and DAE_WO_IR using t-SNE in the Predator Prey task, as shown in Figure 2 (b) and (c), respectively. Compared to VDN, which exhibits fragmented and disconnected state distributions, DAE_WO_IR produces more structured and coherent state clusters, with several prominent aggregation regions corresponding to successful capture strategies (highlighted by red boxes). Such structured state distributions indicate that policy-level exploration induces more consistent state transitions over time, which facilitates temporally extended coordinated behaviours under varying initial conditions. In contrast, the stochasticity introduced by ϵ-greedy exploration leads to dispersed state distributions, which may hinder the consistent coordination among agents.

Additionally, we also analyze the exploration dynamics induced by DAE by visualizing the selection frequency of each policy option during training. The results indicate that policy-level exploration

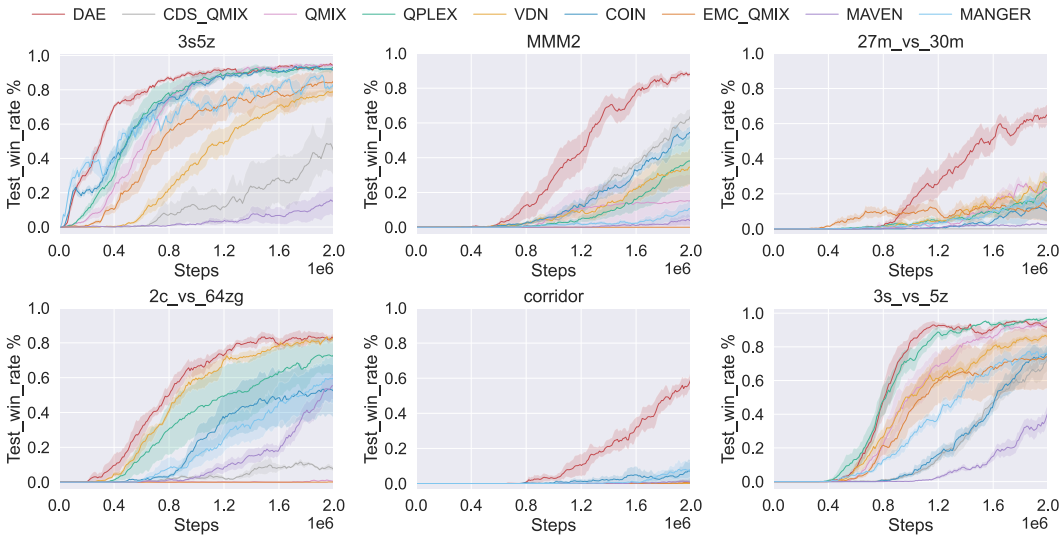


Figure 3: Learning curves on SMAC

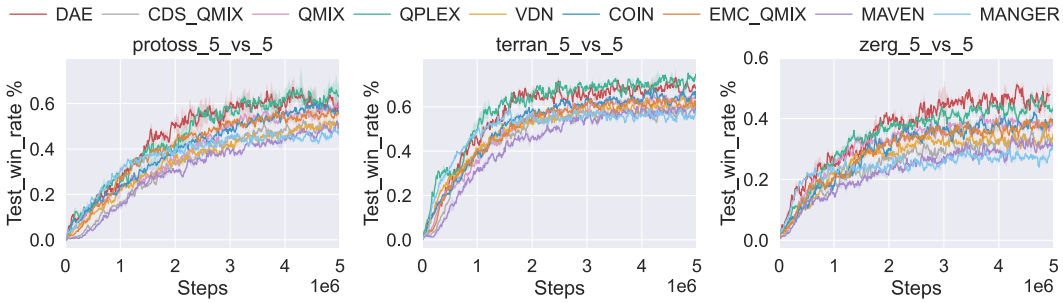


Figure 4: Learning curves on SMACv2

gradually reduces local exploration randomness by transitioning from frequent exploration-policy switching to persistent exploitation-policy execution. Detailed analyses are provided in Appendix D.

5.2 SMAC

We test DAE and baselines on six SMAC scenarios with varying difficulty levels, including an easy task ($3s5z$), two hard tasks ($3s_vs_5z$ and $2c_vs_64zg$), and three super hard tasks ($MMM2$, $corridor$, and $27m_vs_30m$). Figure 3 illustrates that DAE can improve learning performance across various SMAC scenarios. As exemplified in the $27m_vs_30m$ scenario, we can observe that most algorithms struggle to complete tasks successfully. The primary reason is that as the number of agents increases, coordinated exploration becomes increasingly challenging, potentially hindering the ability to reproduce coordinated behaviours. To further validate the robustness of DAE, we conduct experiments on stochastic SMACv2 tasks, including $protoss_5_vs_5$, $terran_5_vs_5$, and $zerg_5_vs_5$, as shown in Figure 4. DAE exhibits learning speed comparable to QPLEX, while outperforming the other QMIX-based methods. COIN attains a competitive final performance but exhibits slower convergence. QPLEX also achieves strong performance on several tasks, benefiting from its expressive value decomposition. Compared to these methods, DAE exhibits faster learning and competitive final performance.

5.3 Google Research Football

We evaluate the performance of all algorithms on three Google Research Football scenarios, i.e., $academy_counterattack_hard$, $academy_pass_and_shoot_with_keeper$, and $academy_3_vs_1_with_keeper$, as shown in Figure 5. In these sparse-reward tasks, DAE exhibits higher

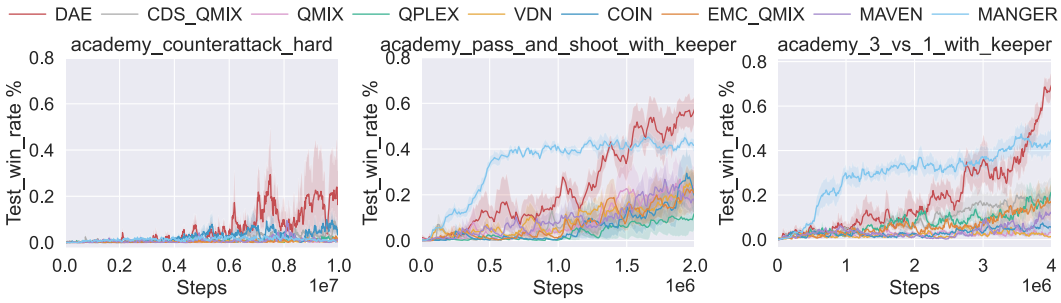


Figure 5: Learning curves on Google Research Football

win rates compared to baseline methods. These results highlight the effectiveness of the proposed exploration framework in sparse-reward settings. Although MANGER exhibits fast learning speed during the early-to-middle stages of the exploration process, it eventually falls into suboptimal behaviours due to the lack of task-oriented guidance in its learning.

5.4 MACO

We conduct experiments on a Multi-Agent Coordination benchmark (MACO) scenario $gather$ where all agents must coordinate by selecting identical actions at each timestep to achieve a shared goal specified at the beginning of each episode. The learning curves on this task can be found in Figure 6. Both VDN and MAVEN struggle to discover effective coordinated policies, converging to suboptimal solutions with an average win rate of approximately 35%. Although several baseline methods that incorporate enhanced exploration mechanisms, such as EMC_QMIX, achieve better performance, DAE exhibits significantly faster and more stable learning, demonstrating its superior ability to promote coordinated exploration in demanding cooperative settings.

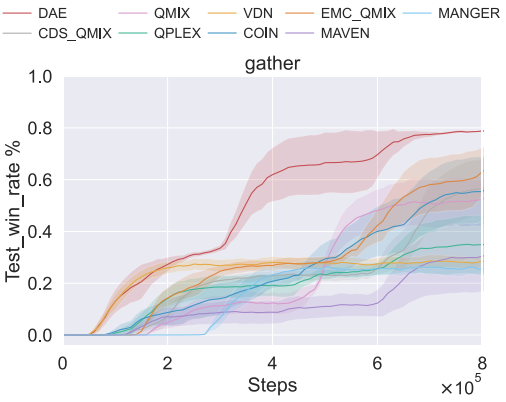


Figure 6: Learning curves on MACO

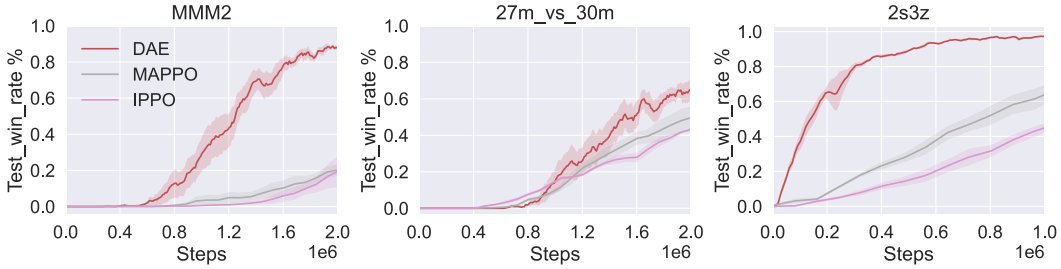


Figure 7: Comparison results with softmax exploration strategy

5.5 Comparison with Softmax Exploration

In this section, we compare DAE with two policy-based baselines, MAPPO and IPPO, both of which use the softmax exploration strategy. The evaluation results on three SMAC scenarios ($2s3z$, $27m_vs_30m$, and $MMM2$) are shown in Figure 7. Across all tasks, DAE demonstrates faster convergence and competitive or superior final performance than both MAPPO and IPPO. Policy-based methods introduce stochasticity through action sampling at each timestep. Although the action distributions are learned and evolve during training, stochasticity is still introduced at the action level,

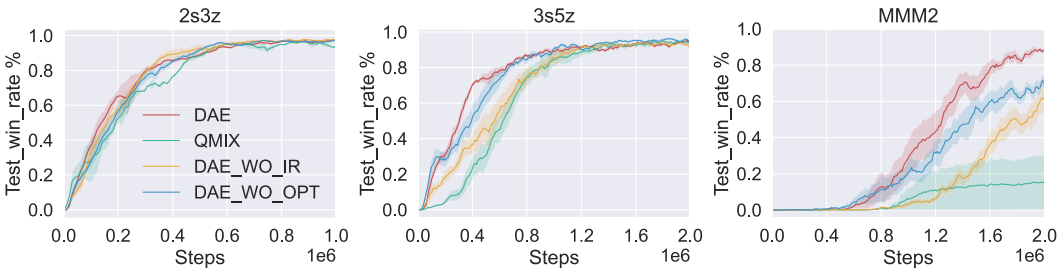


Figure 8: Ablation studies for each component

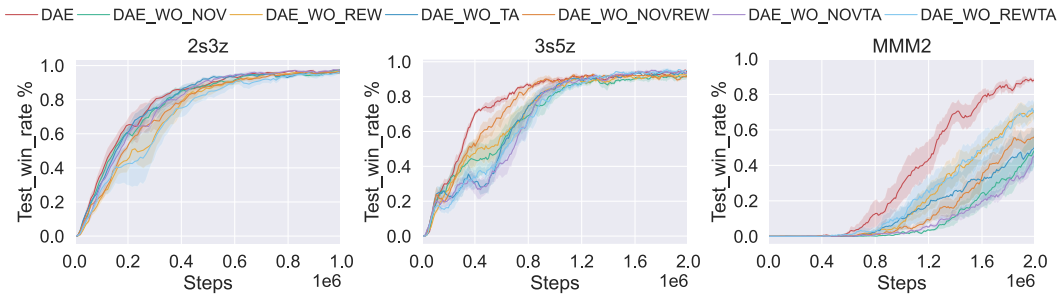


Figure 9: Ablation studies for different auxiliary reward components

which may lead to variability in temporally extended behaviours due to accumulated local randomness. In contrast, DAE performs exploration at the policy level via option selection, while maintaining deterministic action execution conditioned on the selected policy. Although options are selected at each timestep, the resulting behaviours are governed by a small set of underlying policies rather than independent stochasticity. This exploration reduces sensitivity to local randomness and facilitates temporally extended coordination over time. Importantly, our proposed policy-level exploration framework can be incorporated into policy-gradient frameworks.

5.6 Ablation Studies

To validate the contribution of each component in DAE, we conduct ablation studies on three SMAC scenarios: $2s3z$, $3s5z$, and $MMM2$. We consider three variants: (1) removing policy-level exploration (DAE_WO_OPT), (2) removing stage-dependent auxiliary rewards (DAE_WO_IR), and (3) removing both components (QMIX). As shown in Figure 8, the effects of removing individual components vary across scenarios. While all variants achieve comparable performance on simpler tasks, removing either component leads to more pronounced performance degradation on challenging tasks, demonstrating their contributions to DAE. DAE_WO_IR consistently outperforms QMIX across all tasks, demonstrating that policy-level exploration can facilitate more effective coordinated exploration than action-level stochasticity. We further evaluate the impact of different auxiliary reward components by removing individual signals, including novelty-oriented (DAE_WO_NOV), reward-oriented (DAE_WO_REW), and task-oriented (DAE_WO_TA), as well as their combinations (e.g., DAE_WO_NOVREW, DAE_WO_NOVTA, DAE_WO_REWTA). As shown in Figure 9, removing any of these components generally leads to reduced performance, indicating that these signals provide useful guidance during training. Additional ablation results are provided in Appendix C.2.

6 Conclusion

In this paper, we propose a policy-level exploration framework with deterministic action execution (DAE) for MARL. By shifting locally random action exploration to temporally extended policy selection for each agent, the proposed approach facilitates temporally extended coordinated behaviours. The stage-dependent auxiliary rewards are incorporated to provide additional guidance during training. Extensive experiments across diverse multi-agent benchmarks demonstrate that DAE improves both sample efficiency and final performance compared to mainstream baselines. Future work includes

extending the framework to large-scale and continuous-action environments, as well as investigating its applicability to real-world domains such as embodied robotics and swarm systems.

7 Limitations

While DAE demonstrates strong empirical performance across a range of cooperative MARL benchmarks, several limitations remain. First, the current framework is primarily evaluated in discrete-action settings with moderate numbers of agents, and extending it to continuous-action domains remains challenging. Second, DAE relies on a predefined set of policies, and how to automatically construct or adapt this policy set remains an open problem.

Acknowledgments and Disclosure of Funding

We gratefully acknowledge the support from the Distinguished Young Scholars Project funded by the Natural Science Foundation of Guangdong Province (No. 2025B1515020060), and the Basic and Applied Basic Research Program of the Guangzhou Science and Technology Plan (No. 2025A04J7141).

References

[1] Pierre-Luc Bacon, Jean Harb, and Doina Precup. The option-critic architecture. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.

[2] Rujikorn Charakorn, Poramate Manoonpong, and Nat Dilokthanakul. Generating diverse cooperative agents by learning incompatible policies. In *International Conference on Learning Representations*, 2023.

[3] Yangkun Chen, Kai Yang, Jian Tao, and Jiafei Lyu. Novelty-guided data reuse for efficient and diversified multi-agent reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 15930–15938, 2025.

[4] Li Chenghao, Tonghan Wang, Chengjie Wu, Qianchuan Zhao, Jun Yang, and Chongjie Zhang. Celebrating diversity in shared multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 34:3991–4002, 2021.

[5] Kai Cui, Anam Tahir, Gizem Ekinci, Ahmed Elshamanhory, Yannick Eich, Mengguang Li, and Heinz Koeppl. A survey on large-population systems and scalable multi-agent reinforcement learning. *arXiv preprint arXiv:2209.03859*, 2022.

[6] Christian Schroeder De Witt, Tarun Gupta, Denys Makoviichuk, Viktor Makoviychuk, Philip HS Torr, Mingfei Sun, and Shimon Whiteson. Is independent learning all you need in the starcraft multi-agent challenge? *arXiv preprint arXiv:2011.09533*, 2020.

[7] Haowen Dou, Lujuan Dang, Zhirong Luan, and Badong Chen. Measuring mutual policy divergence for multi-agent sequential exploration. *Advances in Neural Information Processing Systems*, 37:76265–76288, 2024.

[8] Abhimanyu Dubey and Alex Pentland. Provably efficient cooperative multi-agent reinforcement learning with function approximation. *arXiv preprint arXiv:2103.04972*, 2021.

[9] Benjamin Ellis, Jonathan Cook, Skander Moalla, Mikayel Samvelyan, Mingfei Sun, Anuj Mahajan, Jakob Foerster, and Shimon Whiteson. Smacv2: An improved benchmark for cooperative multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 36:37567–37593, 2023.

[10] Jianye Hao, Tianpei Yang, Hongyao Tang, Chenjia Bai, Jinyi Liu, Zhaopeng Meng, Peng Liu, and Zhen Wang. Exploration in deep reinforcement learning: From single-agent to multiagent domain. *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[11] Shuncheng He, Yuhang Jiang, Hongchang Zhang, Jianzhun Shao, and Xiangyang Ji. Wasserstein unsupervised reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 6884–6892, 2022.

[12] Hao-Lun Hsu, Weixin Wang, Miroslav Pajic, and Pan Xu. Randomized exploration in cooperative multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 37:74617–74689, 2024.

[13] Siyi Hu, Chuanlong Xie, Xiaodan Liang, and Xiaojun Chang. Policy diagnosis via measuring role diversity in cooperative multi-agent rl. In *International Conference on Machine Learning*, pages 9041–9071. PMLR, 2022.

[14] Dom Huh and Prasant Mohapatra. Multi-agent reinforcement learning: A comprehensive survey. *arXiv preprint arXiv:2312.10256*, 2023.

[15] Shariq Iqbal and Fei Sha. Coordinated exploration via intrinsic rewards for multi-agent reinforcement learning. *arXiv preprint arXiv:1905.12127*, 2019.

[16] Haobin Jiang, Ziluo Ding, and Zongqing Lu. Settling decentralized multi-agent coordinated exploration by novelty sharing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17444–17452, 2024.

[17] Yonghyeon Jo, Sunwoo Lee, Junghyuk Yeom, and Seungyul Han. Fox: Formation-aware exploration in multi-agent reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 12985–12994, 2024.

[18] Karol Kurach, Anton Raichuk, Piotr Stańczyk, Michał Zając, Olivier Bachem, Lasse Espeholt, Carlos Riquelme, Damien Vincent, Marcin Michalski, Olivier Bousquet, et al. Google research football: A novel reinforcement learning environment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 4501–4510, 2020.

[19] Pawel Ladosz, Lilian Weng, Minwoo Kim, and Hyondong Oh. Exploration in deep reinforcement learning: A survey. *Information Fusion*, 85:1–22, 2022.

[20] Chenghao Li, Tonghan Wang, Chongjie Zhang, and Qianchuan Zhao. Never explore repeatedly in multi-agent reinforcement learning. *arXiv preprint arXiv:2308.09909*, 2023.

[21] Jiahui Li, Kun Kuang, Baoxiang Wang, Xingchen Li, Fei Wu, Jun Xiao, and Long Chen. Two heads are better than one: A simple exploration framework for efficient multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 36:20038–20053, 2023.

[22] Jiong Li and Pratik Gajane. Curiosity-driven exploration in sparse-reward multi-agent reinforcement learning. *arXiv preprint arXiv:2302.10825*, 2023.

[23] Boyin Liu, Zhiqiang Pu, Yi Pan, Jianqiang Yi, Yanyan Liang, and Du Zhang. Lazy agents: A new perspective on solving sparse reward problem in multi-agent reinforcement learning. In *International Conference on Machine Learning*, pages 21937–21950. PMLR, 2023.

[24] Anuj Mahajan, Tabish Rashid, Mikayel Samvelyan, and Shimon Whiteson. Maven: Multi-agent variational exploration. *Advances in Neural Information Processing Systems*, 32, 2019.

[25] Zepeng Ning and Lihua Xie. A survey on multi-agent reinforcement learning and its application. *Journal of Automation and Intelligence*, 3(2):73–91, 2024.

[26] Frans A Oliehoek, Christopher Amato, et al. *A concise introduction to decentralized POMDPs*, volume 1. Springer, 2016.

[27] Afshin Oroojlooy and Davood Hajinezhad. A review of cooperative multi-agent deep reinforcement learning. *Applied Intelligence*, 53(11):13677–13722, 2023.

[28] Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. Deep exploration via bootstrapped dqn. *Advances in Neural Information Processing Systems*, 29, 2016.

[29] Bei Peng, Tabish Rashid, Christian Schroeder de Witt, Pierre-Alexandre Kamienny, Philip Torr, Wendelin Böhmer, and Shimon Whiteson. Facmac: Factored multi-agent centralised policy gradients. *Advances in Neural Information Processing Systems*, 34:12208–12221, 2021.

[30] Tabish Rashid, Gregory Farquhar, Bei Peng, and Shimon Whiteson. Weighted qmix: Expanding monotonic value function factorisation for deep multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 33:10199–10210, 2020.

[31] Tabish Rashid, Mikayel Samvelyan, Christian Schroeder, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning. In *International Conference on Machine Learning*, pages 4295–4304. PMLR, 2018.

[32] Kyunghwan Son, Daewoo Kim, Wan Ju Kang, David Earl Hostallero, and Yung Yi. Qtran: Learning to factorize with transformation for cooperative multi-agent reinforcement learning. In *International Conference on Machine Learning*, pages 5887–5896. PMLR, 2019.

[33] Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinicius Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z Leibo, Karl Tuyls, et al. Value-decomposition networks for cooperative multi-agent learning based on team reward. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pages 2085–2087, 2018.

[34] Zhenggang Tang, Chao Yu, Boyuan Chen, Huazhe Xu, Xiaolong Wang, Fei Fang, Simon Shaolei Du, Yu Wang, and Yi Wu. Discovering diverse multi-agent strategic behavior via reward randomization. In *International Conference on Learning Representations*, 2021.

[35] Maxime Toquebiau, Nicolas Bredeche, Faïz Benamar, and Jae-Yun Jun. Joint intrinsic motivation for coordinated exploration in multi-agent deep reinforcement learning. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems*, pages 2522–2524, 2024.

[36] Oriol Vinyals, Timo Ewalds, Sergey Bartunov, Petko Georgiev, Alexander Sasha Vezhnevets, Michelle Yeo, Alireza Makhzani, Heinrich Küttler, John Agapiou, Julian Schrittwieser, et al. Starcraft ii: A new challenge for reinforcement learning. *arXiv preprint arXiv:1708.04782*, 2017.

[37] Jianhao Wang, Zhizhou Ren, Terry Liu, Yang Yu, and Chongjie Zhang. Qplex: Duplex dueling multi-agent q-learning. In *International Conference on Learning Representations*, 2020.

[38] Tonghan Wang, Jianhao Wang, Chongyi Zheng, and Chongjie Zhang. Learning nearly decomposable value functions via communication minimization. In *International Conference on Learning Representations*, 2020.

[39] Annie Wong, Thomas Bäck, Anna V Kononova, and Aske Plaat. Deep multiagent reinforcement learning: Challenges and directions. *Artificial Intelligence Review*, 56(6):5023–5056, 2023.

[40] Tianpei Yang, Hongyao Tang, Chenjia Bai, Jinyi Liu, Jianye Hao, Zhaopeng Meng, and Peng Liu. Exploration in deep reinforcement learning: a comprehensive survey. *arXiv preprint arXiv:2109.06668*, 2021.

[41] Chao Yu, Akash Velu, Eugene Vinitsky, Jiaxuan Gao, Yu Wang, Alexandre Bayen, and Yi Wu. The surprising effectiveness of ppo in cooperative multi-agent games. *Advances in Neural Information Processing Systems*, 35:24611–24624, 2022.

[42] Shaowei Zhang, Jiahan Cao, Lei Yuan, Yang Yu, and De-Chuan Zhan. Self-motivated multi-agent exploration. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*, pages 476–484, 2023.

[43] Yucong Zhang and Chao Yu. Expode: Exploiting policy discrepancy for efficient exploration in multi-agent reinforcement learning. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*, pages 58–66, 2023.

[44] Lulu Zheng, Jiarui Chen, Jianhao Wang, Jiamin He, Yujing Hu, Yingfeng Chen, Changjie Fan, Yang Gao, and Chongjie Zhang. Episodic multi-agent reinforcement learning with curiosity-driven exploration. *Advances in Neural Information Processing Systems*, 34:3757–3769, 2021.

[45] Ziyuan Zhou, Guanjun Liu, and Ying Tang. Multi-agent reinforcement learning: Methods, applications, visionary prospects, and challenges. *arXiv preprint arXiv:2305.10091*, 2023.

A Detailed Explanation

A.1 Analysis of Action-Level Exploration

The following analysis is intended to provide intuition regarding how different exploration mechanisms influence the reproduction of coordinated trajectory segments in cooperative MARL. Rather than establishing formal guarantees for practical MARL systems, the analysis focuses on trajectory segments to illustrate how independently injected action-level stochasticity can affect temporally extended coordination behaviours.

A trajectory segment of length k is defined as

$$\tau_{t+k} = (s_t, \mathbf{a}_t), \ldots, (s_{t+k}, \mathbf{a}_{t+k}),$$

which represents a temporally extended coordinated behaviour among agents.

Theorem 1 *Consider an N-agent cooperative MARL problem with horizon T, where each agent follows independent ϵ-greedy exploration with action space size M. Let $\hat{\tau}_{t+k}$ be any previously observed trajectory segment of length k. Then the probability of reproducing the same action sequence in $\hat{\tau}_{t+k}$ under independent action-level exploration is bounded as*

$$\left(\frac{\epsilon}{M}\right)^{kN} \;\leq\; \Pr(\hat{\tau}_{t+k}) \;\leq\; \left(1 - \epsilon + \frac{\epsilon}{M}\right)^{kN}.$$

Proof Fix a previously observed joint trajectory $\hat{\tau}_{t+k} = \left(\hat{s}_t, \hat{\mathbf{a}}_t, \ldots, \hat{s}_{t+k}, \hat{\mathbf{a}}_{t+k}\right)$. To reproduce this segment, each agent $i \in \{1, \ldots, N\}$ must select the corresponding action $\hat{a}^i_{t'}$ at each timestep $t' \in \{t, \ldots, t+k\}$. Under independent ϵ-greedy exploration, for any agent i and timestep t', the probability of selecting the target action $\hat{a}^i_{t'}$ satisfies

$$\Pr(a^i_t = \hat{a}^i_t) = \begin{cases} 1 - \epsilon + \frac{\epsilon}{M}, & \text{If } \hat{a}^i_t \text{ is the greedy action} \\ \frac{\epsilon}{M}, & \text{otherwise.} \end{cases}$$

Therefore,

$$\frac{\epsilon}{M} \;\leq\; \Pr(a^i_{t'} = \hat{a}^i_{t'}) \;\leq\; 1 - \epsilon + \frac{\epsilon}{M}.$$

Under the simplifying assumption that exploratory action selections are conditionally independent across agents and timesteps,

$$\Pr(\hat{\tau}_{t+k}) = \prod_{t'=t}^{t+k} \prod_{i=1}^{N} \Pr(a^i_{t'} = \hat{a}^i_{t'}),$$

Applying the pointwise lower and upper bounds above yields

$$\left(\frac{\epsilon}{M}\right)^{kN} \;\leq\; \Pr(\hat{\tau}_{t+k}) \;\leq\; \left(1 - \epsilon + \frac{\epsilon}{M}\right)^{kN}.$$

Since $0 < 1 - \epsilon + \frac{\epsilon}{M} < 1$ whenever $\epsilon > 0$ and $M > 1$, the upper bound decays exponentially with kN, indicating that reproducing coordinated trajectory segments may become increasingly difficult as the number of agents or segment length grows.

A.2 Analysis of Policy-level Exploration

We analyze the probability of reproducing the same trajectory segment under policy-level exploration. In DAE, each agent selects a policy using an option policy and executes actions deterministically conditioned on the selected policy. Let $\omega^i_{t'}$ denote the policy identity selected by agent i at timestep t', and let $\hat{\omega}^i_{t'}$ be the selected policy identity corresponding to the previously observed trajectory segment $\hat{\tau}_{t+k}$.

To reproduce $\hat{\tau}_{t+k}$, it suffices that each agent selects the corresponding policy sequence $\{\hat{\omega}^i_{t'}\}^{t+k}_{t'=t}$, since actions are deterministically generated given the selected policies. Assume that agents follow

an option policy that selects policies with probability bounded below by $p_{\min} > 0$. Then, for any agent i and timestep t',

$$\Pr(\omega^i_{t'} = \hat{\omega}^i_{t'}) \geq p_{\min}.$$

Therefore, the probability of reproducing $\hat{\tau}_{t+k}$ satisfies

$$\Pr(\hat{\tau}_{t+k}) = \prod_{t'=t}^{t+k} \prod_{i=1}^{N} \Pr(\omega^i_{t'} = \hat{\omega}^i_{t'}) \geq (p_{\min})^{kN}.$$

In DAE, selected policies are temporally extended and can persist across multiple time steps. Let K_i denote the number of policy-selection events for agent i within the segment τ_{t+k}, where $K_i \leq k$. Since the selected policy remains unchanged between termination events, the probability of reproducing $\hat{\tau}_{t+k}$ can be more tightly bounded as

$$\Pr(\hat{\tau}_{t+k}) \geq (p_{\min})^{\sum_{i=1}^{N} K_i} \geq (p_{\min})^{\sum_{i=1}^{N} k}.$$

When selected policies persist over multiple time steps, $K_i \ll k$, and the exponential dependence shifts from the segment length k to the number of policy-selection events.

Comparing Theorem 1 with the above bound reveals a key difference: under action-level exploration, the probability of reproducing coordinated trajectory segments decays exponentially with kN, whereas under policy-level exploration, the decay depends on the number of policy-selection events. This analysis illustrates that reducing step-wise action stochasticity and introducing temporally extended policy execution can improve the likelihood of reproducing coordinated behaviours in cooperative multi-agent settings.

A.3 The Learning of Exploration Policies

In DAE, an option policy is used to guide the agents' coordinated exploration. Specifically, during the exploration process, each agent needs to select from a set of greedy policies, which consist of an exploitation policy and multiple exploration policies. The exploitation policy maximizes the cumulative discounted team reward:

$$\mathcal{L}_\theta = \mathbb{E}_D \left[\left(r_t + \gamma max_{\boldsymbol{a}_{t+1}} Q^{tot}_{\theta'}(\boldsymbol{o}_{t+1}, \cdot) - Q^{tot}_{\theta}(\boldsymbol{o}_t, \boldsymbol{a}_t) \right)^2 \right],$$

where θ is the network for the exploitation policy and θ' is the target network. Each exploration policy ϕ_l maximizes the distance from the exploitation policy θ and other exploration policies ϕ_{-l} via mutual information maximization, thus promoting diversified exploration among the agents:

$$max\ I(\phi_l; [\theta, \phi_{-l}] | \boldsymbol{o}_t).$$

In the implementation, we maximize the mutual information between the actions produced by network ϕ_l and the identity of the networks $[\theta, \phi_{-l}]$, given the observation $\boldsymbol{o}_t$:

$$I(\pi_{\phi_l}; ID | \boldsymbol{o}_t),$$

Where $ID = \{ID_\theta, ID_{\phi_{-l}}\}$. Then, we can derive the lower bound of this mutual information:

$$\begin{aligned}
&I(\pi_{\phi_l}; ID | \boldsymbol{o}_t) = \mathbb{E}_{ID,\boldsymbol{o},\boldsymbol{a}} \left[log \frac{p(\pi_{\phi_l} | ID, \boldsymbol{o}_t)}{p(\pi_{\phi_l} | \boldsymbol{o}_t)} \right] \\
&= \mathbb{E}_{ID,\boldsymbol{o},\boldsymbol{a}} \left[log \frac{q(\pi_{\phi_l} | ID, \boldsymbol{o}_t)}{p(\pi_{\phi_l} | \boldsymbol{o}_t)} * \frac{p(\pi_{\phi_l} | ID, \boldsymbol{o}_t)}{q(\pi_{\phi_l} | ID, \boldsymbol{o}_t)} \right] \\
&= \mathbb{E}_{ID,\boldsymbol{o},\boldsymbol{a}} \left[log \frac{q(\pi_{\phi_l} | ID, \boldsymbol{o}_t)}{p(\pi_{\phi_l} | \boldsymbol{o}_t)} \right] + \mathbb{E}_{ID,\boldsymbol{o},\boldsymbol{a}} \left[\frac{p(\pi_{\phi_l} | ID, \boldsymbol{o}_t)}{q(\pi_{\phi_l} | ID, \boldsymbol{o}_t)} \right] \\
&= \mathbb{E}_{ID,\boldsymbol{o},\boldsymbol{a}} \left[log \frac{q(\pi_{\phi_l} | ID, \boldsymbol{o}_t)}{p(\pi_{\phi_l} | \boldsymbol{o}_t)} \right] + D_{KL}(p(\pi_{\phi_l} | ID, \boldsymbol{o}_t) || q(\pi_{\phi_l} | ID, \boldsymbol{o}_t)) \\
&\geq \mathbb{E}_{ID,\boldsymbol{o},\boldsymbol{a}} \left[log \frac{q(\pi_{\phi_l} | ID, \boldsymbol{o}_t)}{p(\pi_{\phi_l} | \boldsymbol{o}_t)} \right].
\end{aligned}$$

The lower bound of this mutual information is considered as the reward r_t^{ER} used for the learning of each exploration policy. We train the exploration policies as follows:

$$\mathcal{L}_{\phi_i} = \mathbb{E}_D \left[(r_t^{ER} + \gamma \max_{\boldsymbol{a}_{t+1}} Q_{\phi_l'}^{tot}(\boldsymbol{o}_{t+1}, \cdot) - Q_{\phi_l}^{tot}(\boldsymbol{o}_t, \boldsymbol{a}_t))^2 \right],$$

where ϕ_l' is the target network for ϕ_l.

A.4 Stage-dependent Auxiliary Rewards

Although policy-level exploration facilitates temporally extended coordination, sparse reward feedback may cause the option policy to over-select policies that occasionally obtain rewards, thereby degrading exploration efficiency. To address this challenge, we incorporate stage-dependent auxiliary rewards that provide exploration guidance during different stages of learning. These rewards are designed to address different exploration requirements during learning, including broad coverage of uncertain state-action regions during early exploration, sensitivity to informative reward variations during policy refinement, and exploration toward behaviours correlated with task completion.

During the early stage of training, agents primarily require broad exploration over the state-action space to avoid premature convergence. To this end, we define a novelty-oriented reward as the discrepancy between the true action distribution and a predicted action distribution conditioned on the observation. Specifically, a prediction model is trained to estimate the action distribution given the observation, and the prediction error is used as the novelty signal:

$$r_t^{nov} = Dist\big(\pi(a|o_t) \parallel \hat{\pi}(a|o_t)\big),$$

where $Dist$ is the l_2 distance between the true action distribution $\pi(a|o_t)$ and the predicted action distribution $\hat{\pi}(a|o_t)$.

As the learning progresses, the novelty signal diminishes, making it less effective in guiding exploration toward informative states. Identifying transitions associated with significant reward variations thus becomes increasingly important for discovering effective behaviours. To this end, we encourage the agents to focus on the states where environmental feedback rewards fluctuate significantly. Specifically, we build a prediction network $Q_{\hat{\theta}}$ to predict the exploitation policy network of the agents Q_θ, and take the maximum prediction error in an episode as the reward-oriented reward:

$$r_t^{rew} = max_t \ \left(Q_\theta(o_t, a_t) - Q_{\hat{\theta}}(o_t, a_t)\right)^2,$$

where max_t is computed separately within each episode and $Q_{\hat{\theta}}$ has the same network architecture as Q_θ.

In the late stages of the exploration process, agents should focus on exploiting the state information crucial for task completion. We implement a task-oriented reward r_t^{ta} obtained by measuring the mutual information between states and the task completion indicators to improve the learning speed of the agents:

$$min \ -I(\mathbb{I}_{task}; Z) + I(Z; s_t),$$

where s_t is the global state, Z is the latent representation, and $\mathbb{I}_{task}$ equals 1 if the task is completed in the current episode and 0 otherwise. Then, we can derive the upper bound of this mutual information:

$$\begin{aligned}
-I(\mathbb{I}_{task}; Z) + I(Z; s_t) &= -\mathbb{E}_{\mathbb{I}_{task}, Z}\left[p(\mathbb{I}_{task}, Z) log \frac{p(\mathbb{I}_{task}|Z)}{p(\mathbb{I}_{task})}\right] + \mathbb{E}_{Z,s}\left[p(Z, s_t) log \frac{p(Z|s_t)}{p(Z)}\right] \\
&= -\mathbb{E}_{\mathbb{I}_{task}, Z}\left[p(\mathbb{I}_{task}, Z) log \frac{q(\mathbb{I}_{task}|Z)}{p(\mathbb{I}_{task})}\right] - D_{KL}(p(\mathbb{I}_{task}|Z)||q(\mathbb{I}_{task}|Z)) \\
&+ \mathbb{E}_{Z,s}\left[p(Z, s_t) log \frac{p(Z|s_t)}{p'(Z)}\right] - D_{KL}(p(Z)||p'(Z)) \\
&\leq -\mathbb{E}_{\mathbb{I}_{task}, Z}\left[p(\mathbb{I}_{task}, Z) log \frac{q(\mathbb{I}_{task}|Z)}{p(\mathbb{I}_{task})}\right] + \mathbb{E}_{Z,s}\left[p(Z, s_t) log \frac{p(Z|s_t)}{p'(Z)}\right] \\
&= \mathbb{E}_{\mathbb{I}_{task}, Z, s}\left[-log\ q(\mathbb{I}_{task}|Z)\right] - H(\mathbb{I}_{task}) + D_{KL}(p(Z|s_t)||p'(z)),
\end{aligned}$$

where $p'(Z)$ is the prior distribution of Z. The optimization objective of this mutual information is:

$$\mathcal{L} = \mathbb{E}[-log\ q(\mathbb{I}_{task}|Z) + D_{KL}\left(p(Z|s_t)||p'(Z)\right)],$$

We take $D_{KL}\left(p(Z|s_t)||p'(Z)\right)$ as the task-oriented reward r_t^{ta}. In tasks where such signals are absent, the task-oriented reward remains near zero, and the agents are driven by the novelty- and reward-oriented components.

Then, the overall reward is defined as:

$$r_t^{IR} = \begin{cases} \alpha_2 r_t^{nov} + \alpha_3 r_t^{ta}, & \text{if } \delta_t <= \xi \\ \alpha_2 r_t^{rew} + \alpha_3 r_t^{ta}, & \text{if } \delta_t > \xi, \end{cases}$$

where α_2 and α_3 are hyperparameters to control the magnitude of the exploration, ξ is a hyperparameter for distinguishing between different exploration stages, and $\delta_t = |r_t^{nov} - r_{t-1}^{nov}|$, which reflects the change in prediction error between timestep t and $t-1$. A small δ_t indicates that the novelty signal has stabilized, indicating that the agent has sufficiently explored the environment and should shift focus toward more informative regions. It is worth noting that the task-oriented reward is integrated with the other two rewards. In the early stages of exploration, agents barely complete the task successfully, resulting in a near-zero r_t^{ta}. As a result, exploration is primarily driven by the novelty-based reward or the reward-oriented reward. As agents gradually discover successful strategies, r_t^{ta} increases and eventually surpasses the other two rewards, naturally shifting the policy focus from exploration to task optimization. This design enables a smooth and adaptive transition from exploration to exploitation.

Table 1: Basic hyperparameters for all algorithms

Hyperparameter	**DAE**	**VDN**	**QPLEX**	**COIN**	**MANGER**
learning rate	5e-4	5e-4	5e-4	5e-4	1e-3
buffer size 5000	5000	5000	5000	5000	5000
batch size	32	32	32	32	128
training epochs	1	1	1	1	1
parallel threads	1	1	1	1	8
update interval	200	200	200	200	200
γ	0.99	0.99	0.99	0.99	0.99
rnn hidden dim	64	64	64	64	256
optimizer	RMSp	RMSp	RMSp	RMSp	Adam
mixing network	QMIX	VDN	QPLEX	QMIX	QMIX
prioritized buffer	False	False	False	False	False

Table 2: Five additional hyperparameters for DAE

Benchmark	Environment	L	α_1	α_2	α_3	ξ
Predator Prey	Predator Prey	3	0.1	0.001	0.001	0.0001
SMAC	3s5z	3	0.1	0.001	0.001	0.0001
	3s_vs_5z	5	0.1	0.001	0.05	0.0001
	27m_vs_30m	5	0.1	0.001	0.001	0.001
	2s3z	3	0.1	0.001	0.001	0.0001
	3s5z_vs_3s6z	10	0.1	0.1	0.005	0.001
	MMM2	3	0.1	0.0001	0.0001	0.0001
GRF	academy_pass_and_shoot_with_keeper	5	0.1	0.01	0.005	0.0001
	academy_3_vs_1_with_keeper	5	0.1	0.001	0.001	0.001
	academy_counterattack_hard	5	0.1	0.01	0.01	0.005
MACO	gather	3	0.1	0.001	0.001	0.001

B Experiment Settings

B.1 Hyperparameters

The exploitation policy in DAE is implemented using QMIX as the underlying network framework. All shared hyperparameters are consistent with those reported in the original paper, with no modifica-

Algorithm 1 DAE

Input: replay buffer $\mathcal{B}$, batch size B, update interval d, the number of exploration policies L
Output: Exploitation Policy π_θ

Initialize option policy ψ, exploitation policy θ, exploration policies ϕ_l, prediction network $\hat{\theta}$, encoder network φ, decoder network $\hat{\varphi}$, target policies ψ', θ', ϕ'_l
Initialize target network $\psi' \leftarrow \psi$, $\theta' \leftarrow \theta$, $\phi'_l \leftarrow \phi_l$
for each episode iteration **do**
  for each environment step t **do**
    for each agent i **do**
      Select an interaction policy by Eq.(1).
      Select an action by Eq.(2).
    end for
    $s_{t+1} \sim p(s_{t+1}|s_t, \boldsymbol{a}_t)$
    Store transition tuple $(s_t, \boldsymbol{o}_t, \boldsymbol{a}_t, r_t^{ext}, s_{t+1}, \boldsymbol{o}_{t+1})$ in $\mathcal{B}$
  end for
  Sample a mini-batch from $\mathcal{B}$
  for each gradient step **do**
    Evaluate the novelty-oriented reward in Eq.(13)
    Evaluate the reward-oriented reward in Eq.(14)
    Evaluate the task-oriented reward in Eq.(15)
    Evaluate the reward r^{IR} in Eq.(16)
    Evaluate the reward r^{ER} in Eq.(11)
    Using r_t and t_t^{IR} to update ψ in Eq.(8)
    Using r_t to update θ in Eq.(9)
    Using r_t^{ER} to update ϕ_l in Eq.(12)
    Update $\hat{\theta}$ by the prediction error in Eq.(13)
    Update φ and $\hat{\varphi}$ in Eq.(17)
  end for
  if t mod d **then**
    Update the target network:
    $\psi' \leftarrow \psi$, $\theta' \leftarrow \theta$, $\phi'_l \leftarrow \phi_l$
  end if
end for
return optimal exploitation policy

tions to ensure a fair comparison. The basic hyperparameters for all algorithms are summarized in Table 1. As DAE, QMIX, EMC, CDS, and MAVEN share identical hyperparameters, we show only the hyperparameters of DAE in the table for brevity.

Five additional hyperparameters are introduced for DAE to implement efficient exploration:

- L: the number of exploration policies,
- α_1: controls the learning of the option policy,
- α_2 and α_3: scaling factors for the auxiliary rewards,
- ξ: a threshold that determines transitions between exploration stages.

The values of these hyperparameters across different tasks are provided in Table 2.

B.2 Experimental Equipment

For all experiments, evaluations are conducted on a server equipped with an Ubuntu 18.04 operating system, an Intel Xeon Platinum 8269CY CPU, 10 Nvidia RTX 3090 GPUs, 512GB of DDR4 RAM, and 16TB of NVMe SSD storage. The training on all tasks is completed within 24 hours.

In practice, DAE has comparable GPU usage and training time to state-of-the-art methods such as EMC and CDS, with moderate overhead compared to QMIX ($\approx$ 1000MB memory and $\approx$ 2 hours additional training time). Importantly, only the exploitation policy is used during inference, incurring no additional runtime cost.

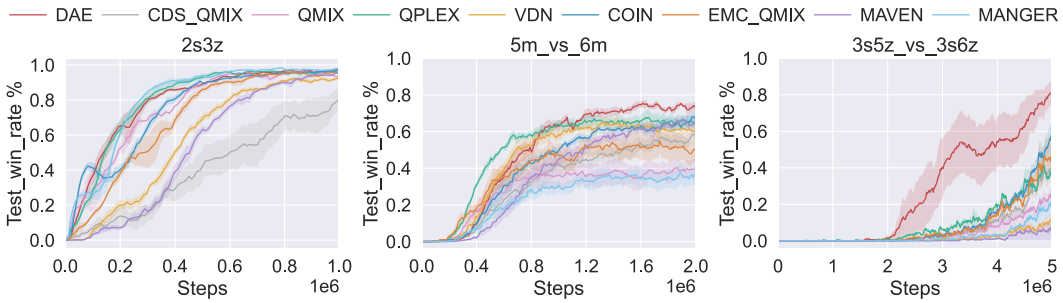


Figure 10: Evaluation results on SMAC

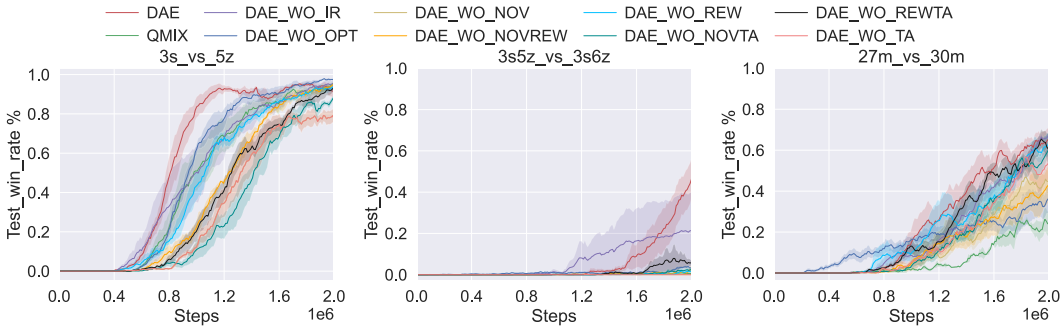


Figure 11: Ablations on SMAC

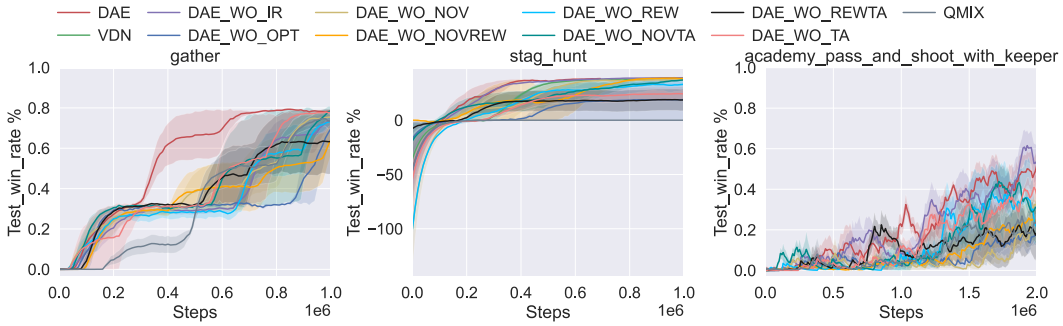


Figure 12: Ablations on Predator Prey, Google Research Football task, and MACO

B.3 The Whole Training Process

The whole training process of DAE is summarized in Algorithm 1.

C More Experimental Results

C.1 More Evaluations on SMAC

As shown in Figure 10, we perform experiments on three SMAC tasks, including $2s3z$ (easy), $5m_vs_6m$ (hard), and $3s5z_vs_3s6z$ (super hard). DAE can outperform the majority of baselines in terms of learning efficiency, achieving a higher test win rate during the training process. In the $2s3z$ task, QPLEX is able to achieve faster convergence than DAE, which is primarily attributed to the advantages of its dueling mixing network. MANGER also exhibits strong sample efficiency in the $2s3z$ task by controlling the update frequency of policies, but its performance degrades on the harder tasks.

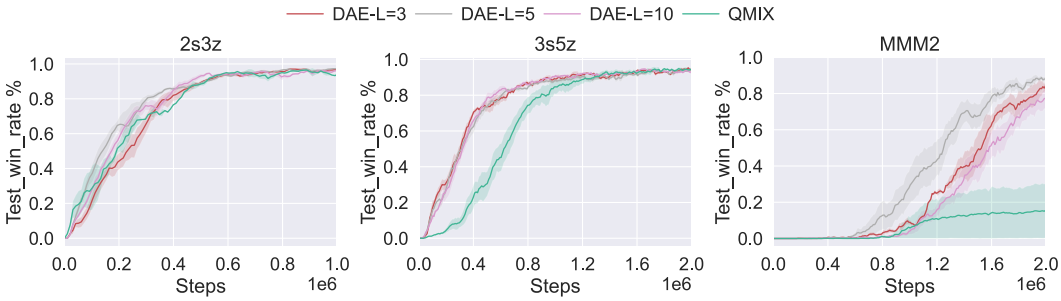


Figure 13: Ablation study of L.

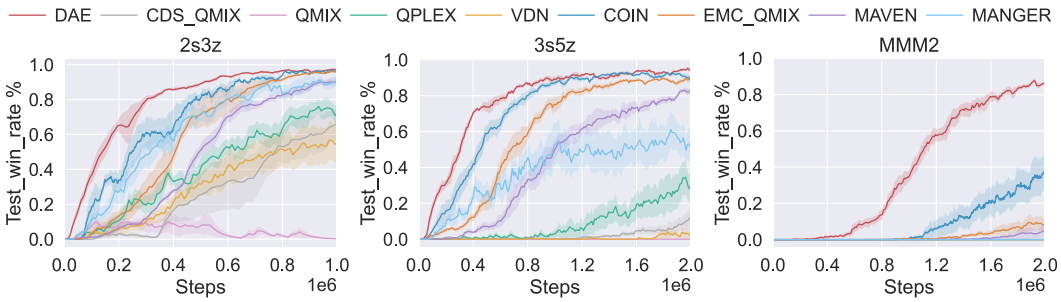


Figure 14: Results without $\epsilon - greedy$ exploration

C.2 More Ablation Studies

C.2.1 Ablation Studies on Various Benchmarks

As shown in Figure 11 and Figure 12, we conduct ablation studies on (1) Predator Prey, (2) three SMAC tasks including $27m_vs_30m$, $3s_vs_5z$, and $3s5z_vs_3s6z$, (3) a Google Research Football task, $academy_pass_and_shoot_with_keeper$, and (4) a MACO task, $gather$. DAE demonstrates efficient learning compared to baseline methods across multiple evaluations. Furthermore, it is obvious that in both the $27m_vs_30m$ and $3s5z_vs_3s6z$ tasks, two algorithms (DAE_WO_OPT and QMIX) that rely on $\epsilon - greedy$ exploration underperform compared to others adopting the policy-level exploration mechanism to guide the agents' exploration. The observed performance gap demonstrates that the random action-level selection significantly hampers the learning efficiency as the number of agents increases.

C.2.2 Ablation Studies on Hyperparameters

We evaluate the influence of the different numbers of exploration policies, L, on the robustness of DAE. Experiments are conducted on three SMAC tasks, including $2s3z$, $3s5z$, and $MMM2$. Values for L are selected from the set $\{3, 5, 8\}$. The results are shown in Figure 13. It is evident that DAE with different values of L outperforms QMIX, indicating that the performance of DAE is not sensitive to L for these tasks.

C.3 Evaluations without $\epsilon - greedy$ Exploration

In order to enable a direct comparison of exploration mechanisms, we remove $\epsilon - greedy$ exploration from all algorithms in the experiments. The evaluation results are shown in Figure 14. DAE shows a considerable improvement over the baselines in all tasks, highlighting the importance of both the policy-level exploration and the stage-specific exploration during the exploration process.

C.4 Comparison with Bootstrapped DQN

In single-agent reinforcement learning, Bootstrapped DQN [28] guides exploration via an ensemble of value functions. However, Bootstrapped DQN samples value functions from a uniform distribution,

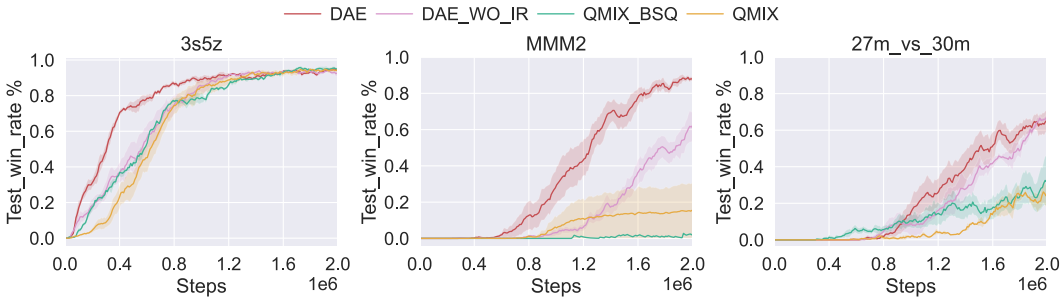


Figure 15: Comparison results.

while the selected value function still relies on ϵ-greedy exploration to select actions during interaction with the environment (as detailed in Appendix D of its original paper and code), indicating that exploration remains stochastic at the action level. In contrast, DAE performs policy-level exploration by using a learned option policy to select from a policy set, while actions are executed deterministically conditioned on the selected policy, thereby facilitating temporally extended exploration. We also carry out experiments to compare DAE with an extension of Bootstrapped DQN integrated into QMIX, denoted as QMIX_BSQ, as shown in Figure 15. The results show that DAE can outperform QMIX_BSQ.

D Selection Frequency of Each Option

To further understand the effect of the option selection mechanism on agent exploration, we visualize the selection frequency of each option in the $3s5z$ task, as shown in Figure 16- 19. During the early stage of training (first 1000 steps), agents frequently switch between exploration and exploitation policies, indicating active exploration guided by policy selection rather than by locally random actions. As learning progresses, the frequency of selecting exploration policies gradually decreases, while the exploitation policy becomes dominant, reflecting a natural transition from exploration to exploitation. Notably, most exploration behaviour in DAE occurs within the first 10000 steps, after which the policy stabilizes. These observations indicate that DAE gradually reduces exploration stochasticity during training by transitioning from frequent exploration-policy switching to persistent exploitation-policy execution. In contrast, baselines using ϵ-greedy exploration, such as QMIX, typically require a longer exploration phase (e.g., 50000 steps). To examine exploration efficiency under comparable budgets, we reduce the exploration phase of QMIX from 50000 to 10000 steps. The training result is shown in Figure 20. We can observe that this reduction leads to a noticeable performance drop, suggesting that DAE achieves more effective exploration within a shorter time horizon.

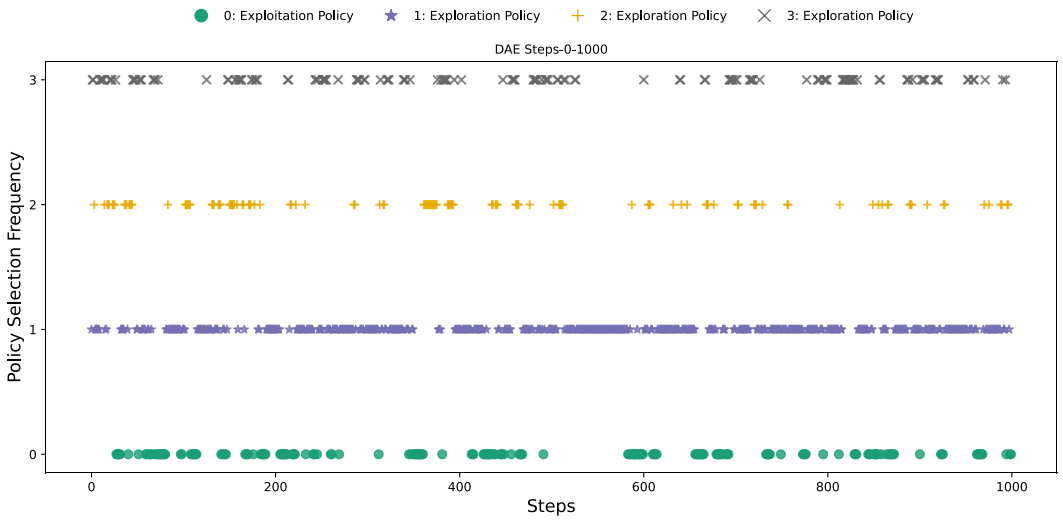


Figure 16: The selection frequency of each option within the first 1000 steps.

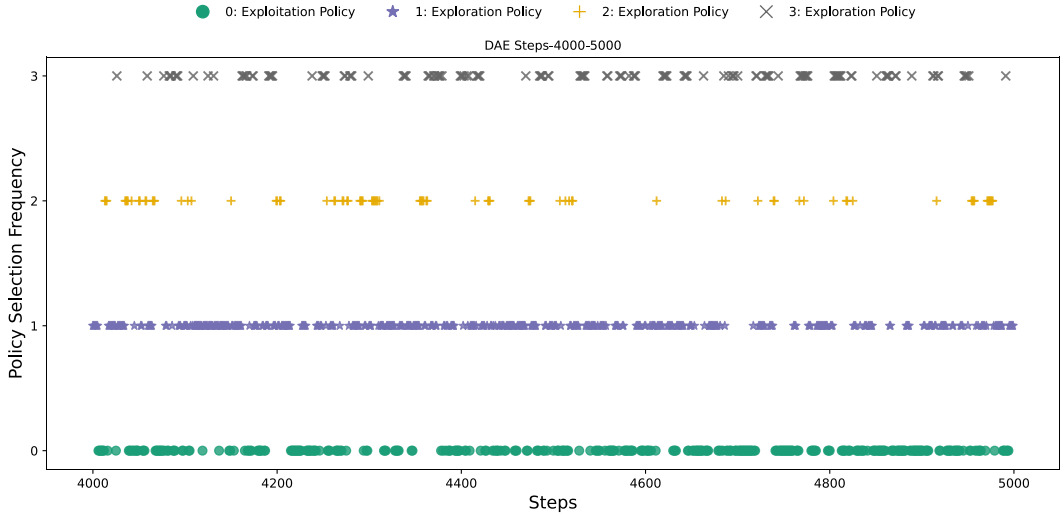


Figure 17: The selection frequency of each option within the range of 4000 to 5000 steps.

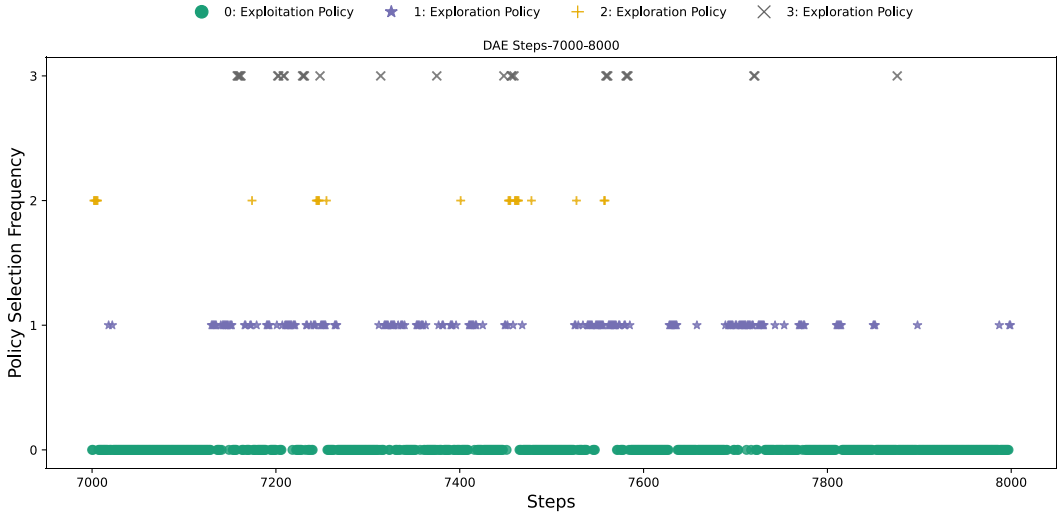


Figure 18: The selection frequency of each option within the range of 7000 to 8000 steps.

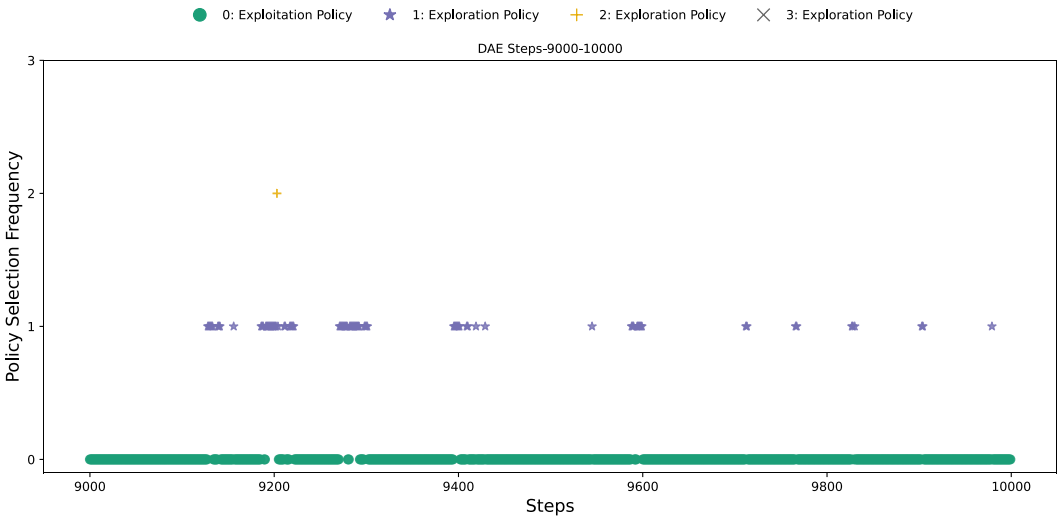


Figure 19: The selection frequency of each option within the range of 9000 to 10000 steps.

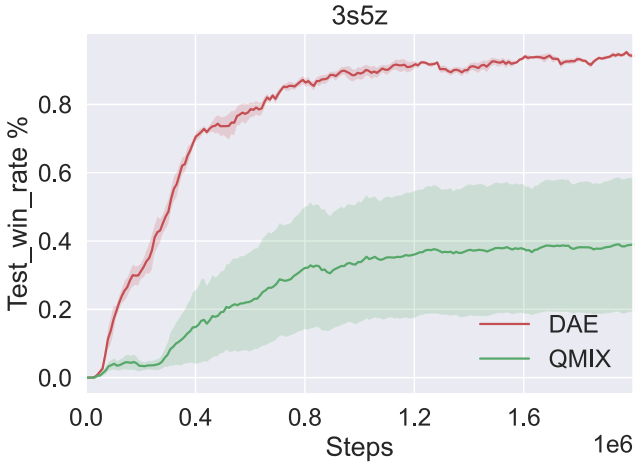


Figure 20: Exploration only within 10000 Steps.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We provide a detailed presentation of our method and the experiments in both the abstract and the introduction.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We present the limitations in Section 7

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: We provide the derivation of the formula in the Appendix.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide all the information.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We upload the source code in the supplementary material.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide detailed experimental settings in the Appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We state the error bars for experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide information about experimental equipment in the Appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This paper conforms to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: There is no negative social impact of our work.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification:

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite the original paper that produced the code package.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification:

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification:

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification:

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification:

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.