Adaptive Episode Length Adjustment for Multi-agent Reinforcement Learning

Byunghyun Yoo
Electronics and Telecommunications
Research Institute (ETRI)
Daejeon, South Korea
bhyoo@etri.re.kr

Younghwan Shin
Electronics and Telecommunications
Research Institute (ETRI)
Daejeon, South Korea
shinyh1115@etri.re.kr

Hyunwoo Kim
Electronics and Telecommunications
Research Institute (ETRI)
Daejeon, South Korea
kimhw@etri.re.kr

Euisok Chung
Electronics and Telecommunications
Research Institute (ETRI)
Daejeon, South Korea
eschung@etri.re.kr

Jeongmin Yang
Electronics and Telecommunications
Research Institute (ETRI)
Daejeon, South Korea
jmyang38@etri.re.kr

ABSTRACT

In standard reinforcement learning, an episode is defined as a sequence of interactions between agents and the environment, which terminates upon reaching a terminal state or a pre-defined episode length. Setting a shorter episode length enables the generation of multiple episodes with the same number of data samples, thereby facilitating an exploration of diverse states. While shorter episodes may limit the collection of long-term interactions, they may offer significant advantages when properly managed. For example, trajectory truncation in single-agent reinforcement learning has shown how the benefits of shorter episodes can be leveraged despite the trade-off of reduced long-term interaction experiences. However, this approach remains underexplored in MARL. This paper proposes a novel MARL approach, Adaptive Episode Length Adjustment (AELA), where the episode length is initially limited and gradually increased based on an entropy-based assessment of learning progress. By starting with shorter episodes, agents can focus on learning effective strategies for initial states and minimize time spent in dead-end states. The use of entropy as an assessment metric prevents premature convergence to suboptimal policies and ensures balanced training over varying episode lengths. We validate our approach using the StarCraft Multi-agent Challenge (SMAC) and a modified predator-prey environment, demonstrating significant improvements in both convergence speed and overall performance compared to existing methods. To the best of our knowledge, this is the first study to adaptively adjust episode length in MARL based on learning progress.

KEYWORDS

Multi-agent reinforcement learning, Episode length adjustment, Dead-end states

1 INTRODUCTION

Multi-agent reinforcement learning (MARL) has become an increasingly important area of research, especially in scenarios involving cooperative or competitive environments with multiple autonomous agents [1, 5, 28]. The ability of agents to make decisions based on their individual perspectives while interacting with other agents presents a complex yet realistic setting that resembles many natural and artificial systems. As MARL evolves, researchers have identified several key challenges, such as non-stationarity, coordination among agents, and the curse of dimensionality. Among these challenges, the impact of episode length or time limits on the learning performance of agents is of particular interest, especially in practical settings where time constraints are inherent.

In MARL, an episode is defined as a sequence of interactions between agents and the environment, which terminates upon reaching a terminal state or a pre-defined episode length. Typically, the episode length is set to a fixed value throughout the training process, and this value is sufficiently large to allow for finding an optimal policy to solve the given ask [24]. Determining an appropriate episode length is not straightforward. Setting a shorter episode length enables the generation of multiple episodes with the same number of data samples, thereby facilitating an exploration of diverse states. This increased exploration can help agents gather a wider variety of experiences, which is beneficial for generalization. However, it also has the drawback of potentially failing to collect experiences involving long-term interactions that exceed the restricted episode length.

Recent research in single-agent reinforcement learning has increasingly focused on techniques such as time limits or trajectory truncation, which utilize only a portion of an episode [3, 12–14]. These methods have been demonstrated to be beneficial both theoretically and empirically. Adjusting episode length is expected to be even more effective in MARL, where the state-action space is

significantly larger. Especially in multi-agent environments, the complexity arising from interactions between agents can lead to the existence of various types of dead-end states where agents cannot reach a desired goal within a reasonable number of steps. Dead-end states can severely hinder the learning process, especially if agents frequently encounter them in early training stages [2, 6, 30]. By adjusting episode length, agents can avoid spending excessive time in such unproductive states, focusing instead on learning effective strategies for early interactions. Despite the potential of these approaches, their application in MARL remains relatively underexplored.

In this paper, we propose a novel approach to MARL called Adaptive Episode Length Approach (AELA), where the episode length is initially limited during the early training stages and then gradually increased based on the learning situation. AELA begins by restricting agents to shorter episodes, allowing them to focus on learning effective strategies for initial states and to reduce a prolonged time in dead-end states. Then, our approach gradually increases the episode length as learning progresses, based on an entropy-based assessment of the agents. Entropy is used to prevent excessive convergence to experiences of a specific episode length, thereby ensuring adequate training over the originally defined episode length for the task. Also, this staged increase in episode length allows agents to incrementally expand their exploration without being overwhelmed by the complexity of the full task from the outset.

We validated our proposed method through experiments conducted on the StarCraft Multi-agent Challenge (SMAC)[18] and modified predator-prey (MPP) [20], which are widely used benchmarks for testing MARL algorithms. Our extensive evaluation reveals that our approach achieves significantly superior performance, particularly in terms of performance and convergence speed. The results demonstrate that limiting the episode length during the early training stages helps agents develop a stronger understanding of the initial state, ultimately improving final performance. To the best of our knowledge, this is the first study to adaptively adjust the episode length based on the learning situation in MARL.

2 RELATED WORK

Cooperative multi-agent reinforcement learning. Methods that decompose joint action values have been extensively studied and have shown notable success in cooperative multi-agent reinforcement learning (MARL) scenarios. The Value Decomposition Network (VDN), introduced by Sunehag et al. [23], models the joint action-value function as the sum of individual action-value functions, which promotes cooperative behavior among agents. QMIX, proposed by Rashid et al. [17], expanded on VDN by incorporating a mixing network that used monotonic utility functions under Individual-Global-Max (IGM) conditions, enhancing the flexibility of value decomposition while preserving the monotonicity constraint. To overcome the limitations of the monotonicity assumption, later approaches such as QTRAN [20] and Weighted QMIX (WQMIX) [16] were developed to relax these constraints by applying weights to the loss functions of suboptimal actions, thereby improving performance. Furthermore, QPLEX [25] employed a

duplex dueling architecture to encode the IGM condition via a neural network, resulting in a more sophisticated value factorization framework. Recent years have also seen increased interest in extending ideas from single-agent reinforcement learning, including distributional RL and risk-sensitive RL, to multi-agent environments [15, 19, 21, 22]. Additionally, role-based learning methods have been proposed to effectively decompose tasks in multi-agent systems [8, 26, 27, 29], and the concept of bidirectional dependencies has been explored to enhance cooperative behaviors among agents [7].

Episode length adjustment. The problem of efficiently managing episode lengths and their impact on reinforcement learning (RL) has gained increasing attention in recent years. Various approaches have been proposed to address issues related to time limits and trajectory truncation. Pardo et al. [12] explored time limits in RL and introduced methods for handling them effectively, showing that including the remaining time as part of the agent's input improves policy learning by avoiding state aliasing and instability. The advantages of truncating trajectories for enhancing Policy Optimization via Importance Sampling (POIS) accuracy were illustrated by [13]. Poiani et al. [14] proposed an adaptive trajectory truncation strategy called RIDO, which allocates interaction budgets dynamically to minimize variance in Monte Carlo policy evaluation. Their results indicate that adaptive schedules outperform fixed-length approaches across multiple RL domains. In addition to directly adjusting the episode length, there have also been studies addressing uncertain episode lengths. Mandal et al. [10] tackled online RL with uncertain episode lengths, proposing an algorithm that adapts to variable horizons by estimating the underlying distribution. They provided theoretical guarantees that highlight the effectiveness of adaptive horizon management in minimizing regret. Horizon Regularized Advantage (HRA) was proposed by [11] to improve the generalization of RL. By averaging advantage estimates from multiple discount factors, HRA improves policy generalization, particularly in environments with stochastic episode durations. As described so far, there have been studies related to episode length adjustment in single-agent environments; however, in multi-agent environments, there has been little related research.

3 BACKGROUND

3.1 Dec-POMDP

A decentralized partially observable Markov decision process (Dec-POMDP) is a framework used to model multi-agent decision-making in environments where agents must cooperate under conditions of uncertainty and limited observability. In a Dec-POMDP, multiple agents interact with a shared environment, but each agent has only partial information about the global state and must make decisions based on its local observations. Formally, a Dec-POMDP can be represented as a tuple $\langle I, S, A, P, O, Z, R, \gamma \rangle$, where I is the set of agents, S represents the set of global states, A is the set of joint actions across all agents, P is the state transition function, O is the set of joint observations, Z is the observation function, R is the reward function, and γ is the discount factor.

In this framework, each agent $i \in I := 1, \ldots, I$ selects actions $a_i \in A$ at each time step t. The state transition function provides

the probability of transitioning from one state to another given the joint action of all agents, as $P(s'|s, \mathbf{a}) : S \times A \times S \longmapsto [0, 1]$. Each agent has individual and partial observations $z \in Z$, according to the observation function $O(s, i) : S \times I \longmapsto Z$. The reward function $R(s, \mathbf{a}) : S \times A \longmapsto \mathbb{R}$ assigns a numerical value to each state-action pair, and $\gamma \in [0, 1)$ is the discount factor. The action-observation history for agent i is $\tau_i \in T := (Z \times \mathbf{A})^*$, on which its policy $\pi_i(a_i|\tau_i) : T \times \mathbf{A} \longmapsto [0, 1]$ is based. The formal goal is to find a joint policy π that optimizes a joint action-value function, which is represented as $Q_{jt}^{\boldsymbol{\pi}}(s, \mathbf{a}) = \mathbb{E}_{s_{0:\infty}, \mathbf{a}_{0:\infty}} [\sum_{t=0}^{\infty} \gamma^t r_t | s_0 = s, \mathbf{a}_0 = \mathbf{a}, \boldsymbol{\pi}]$.

3.2 Episode length adjustment

In reinforcement learning (RL), the agent interacts with the environment in a sequence of steps called an episode, which often has a fixed length E_L. However, adjusting episode length during training can significantly influence learning efficiency and policy performance, particularly in environments where the complexity or difficulty varies over time.

The goal of RL is to learn a policy π that maximizes the expected cumulative reward, also known as the return G_l, which is given by:

$$G_l = \sum_{k=0}^{E_L - 1} \gamma^k r(s_{l+k}, a_{l+k}), \tag{1}$$

where E_L is the length of the episode and l represents the time step within an episode. Note that the term 'time step' may cause some confusion with training steps or other terms, so the time step within an episode is referred to as an interaction step l. l is a natural number ranging from 1 to the episode length, E_L. Traditionally, the episode length E_L is a fixed value, but adjusting the episode length means setting it to some value smaller than E_L.

4 METHOD

In reinforcement learning, dead-end states refer to states from which the agent cannot reach the goal state or recover once reached. To efficiently collect experience in reinforcement learning, it is necessary to minimize the visits to dead-end states. In single-agent reinforcement learning, several techniques have been proposed to estimate and avoid such dead-ends [2, 6, 30]. However, in multi-agent reinforcement learning, the number of states and actions grows exponentially, and the states that lead to successful rewards are relatively few compared to the total number of states and actions. This makes it extremely challenging to directly estimate dead-ends.

Therefore, instead of precisely estimating and completely avoiding dead-ends, this paper focuses on efficiently collecting experience even in the presence of dead-ends. In multi-agent environments where dead-ends exist, effective experience collection often involves visiting states that occur before reaching a dead-end. This strategy is referred to as secure exploration, which emphasizes maximizing the exploration of states that precede dead-ends to ensure meaningful learning opportunities.

4.1 Theoretical background

In this section, we theoretically describe the relationship between reducing episode length and visiting secure states. First, we define a dead-end state and a secure state as follows:

Definition 1 (Dead-end State). *Once the agent reaches dead-end states s_d at time step t in a trajectory, no policy π can lead it to the goal state s_g for any $t' > t$.*

Definition 2 (Secure state). *A state that is not a dead-end state is referred to as a secure state s_s, as there exists at least one policy that guarantees reaching the goal state.*

According to these definitions, the properties of P_s with respect to the interaction step can be proven as follows.

Lemma 1. *Let E_L denote the episode length, and let l (where $l = 1, 2, \ldots, E_L$) denote the interaction step (i.e., the time step within each episode). Let $P_s(l)$ be the probability that the state is secure at interaction step l. Then, it holds that:*

$$P_s(l + 1) \le P_s(l). \tag{2}$$

Proof. Let the probability of falling into a dead-end state at time step l be denoted by $P_d(l)$. To remain in a secure state up to time step l, the agent must avoid falling into a dead-end state at every previous time step. Thus, we have:

$$P_s(l) = \prod_{k=1}^{l} (1 - P_d(k)). \tag{3}$$

To prove that $P_s(l)$ is not increasing with respect to l, we compare $P_s(l)$ and $P_s(l + 1)$:

$$P_s(l + 1) = P_s(l) \times (1 - P_d(l + 1)), \tag{4}$$

where $1 - P_d(l+1)$ represents the probability of avoiding a dead-end state at time step $l + 1$. Since $0 \le 1 - P_d(l + 1) \le 1$, it follows that:

$$P_s(l + 1) = P_s(l) \times (1 - P_d(l + 1)) \le P_s(l). \tag{5}$$

Hence, by comparing $P_s(l)$ and $P_s(l + 1)$, it can be concluded that $P_s(l)$ is a non-increasing function of l. Therefore, the probability of being in a secure state decreases monotonically or remains constant as the interaction step l increases. □

Based on Lemma 1, we prove that reducing the episode length increases or maintains the probability of being a secure state, thereby enabling efficient experience collection. The following two theorems build upon Lemma 1 to establish the theoretical foundation for our approach. Specifically, Theorem 1 provides a theoretical hint that effectively adjusting the episode length can yield benefits (proof in Appendix A). Therefore, in AELA, we implement adaptive control of the episode length based on this insight. Subsequently, Theorem 2 elucidates the impact of decreasing the probability of visiting dead-end states and the benefits in terms of regret minimization. These two theorems provide the theoretical foundation for the design of the AELA algorithm.

Theorem 1. *Let E_L be the fixed episode length, and N_s be the expected number of secure state visits over all episodes. Then, N_s remains constant or increases if E_L is reduced.*

Definition 3 (The probability of visiting dead-end states). *Let N_{total} denote the total number of collected data samples. When N_d denotes the expected number of dead-end state visits across all episodes, then N_d can be expressed as $N_{total} - N_s$. The probability P_d of visiting dead-end states can be defined as $\frac{N_d}{N_{total}}$.*

Algorithm 1: Adaptive Episode Length Adjustment (AELA)

---

1 Initialize each agent's policy or Q-value network parameters $\theta_1, \theta_2, \ldots, \theta_I$, the centralized critic network parameters ϕ, an empty replay buffer $\mathcal{D}$, the initial episode length E_{L_0}, an empty list $\mathcal{L}_H$ for storing entropy values, the window size w for collecting entropy values, and the batch size B

2 Set the current episode length as the initial episode length $E_L \leftarrow E_{L_0}$

3 Set target network parameters $\bar{\theta}_i \leftarrow \theta_i$ for $i = 1, 2, \ldots, I$, and $\bar{\phi} \leftarrow \phi$

4 Set target network update interval T_U

5 Set E_{max} to the episode length of the original task

6 Initialize both the time step t and the previous time step t_p to 0

7 **while** $t < t_{max}$ **do**

8   Initialize the interaction step l to 0

9   Initialize the state as the initial state s_0

10   **while** $l < E_L$ **do**

11   Each agent i takes action $a_{i,l} \sim \pi_{\theta_i}(\cdot|o_{i,l})$

12   Step into state s_{l+1}

13   Receive reward r_l and observe $o_{i,l+1}$

14   Add transition data to the replay buffer: $\mathcal{D} \leftarrow \mathcal{D} \cup \{(s_l, o_l, a_l, r_l, s_{l+1}, o_{l+1})\}$

15          $l = l + 1$, $t = t + 1$

16   **if** $|\mathcal{D}| > B$ **then**

17   Sample a random batch of episodes from $\mathcal{D}$

18   Calculate the total entropy of q-values H_{total}

19   Add the entropy H_{total} to $\mathcal{L}_H$

20   Update the parameters of the centralized critic ϕ with the sampled batch

21   Update each agent's policy or Q-value network parameters $\theta_1, \theta_2, \ldots, \theta_I$ with the sampled batch

22   **if** $|\mathcal{L}_H| \bmod w = 0$ **then**

23   Fit a linear model to the last w values in $\mathcal{L}_H$

24   **if** *(slope of linear model < 0)* **and** *($E_L < E_{max}$)* **then**

25                  $E_L \leftarrow E_L + 1$

26   **if** $(t - t_p) > T_U$ **then**

27   Update the target network of centralized critic $\bar{\phi} \leftarrow \phi$, and target networks of agents $\bar{\theta}_1 \leftarrow \theta_1, \ldots, \bar{\theta}_I \leftarrow \theta_I$

28              $t_p = t$

Definition 4 (Regret). *In the context of reinforcement learning, the regret after T time steps, denoted by $Regret(T)$, measures the difference in cumulative rewards between the optimal policy π^* and the agent's policy π:*

$$Regret(T) = R(\pi^*) - R(\pi), \quad (6)$$

where $R(\pi) = \sum_{t=1}^{T} r_t$ and $R(\pi^*) = \sum_{t=1}^{T} r_t^*$. Here, r_t represents the reward obtained by the agent's policy π at time step t, and r_t^* represents the reward obtained by the optimal policy π^* at the same time step.

Corollary 1. *According to Theorem 1, reducing the episode length E_L either increases or preserves the number of visits to secure states N_s. Consequently, reducing E_L will cause P_d to either decrease or at least remain unchanged.*

PROOF. From Theorem 1, we know that reducing E_L causes N_s to either increase or remain unchanged. Given that $P_d = \frac{N_d}{N_{\text{total}}} = 1 - \frac{N_s}{N_{\text{total}}}$ (defined in Definition 3), a non-decreasing change in N_s (i.e., an increase or no change) results in a non-increasing change in P_d (i.e., a decrease or no change). □

Based on Corollary 1, we establish the following Theorem 2 (proof is provided in Appendix B). To do so, we introduce an assumption that ensures the agent prioritizes reaching the goal state over accumulating intermediate rewards.

Assumption 1. *The reward r_g obtained at the goal state is sufficiently large compared to the rewards received over any interval of time steps within the episode. Formally, for any $1 \le k \le l \le T$,*

$$\sum_{t=k}^{l} r_t < r_g. \quad (7)$$

This assumption ensures that the agent prioritizes reaching the goal state over accumulating intermediate rewards.

Theorem 2. *Under Assumption 1, as P_d decreases, the difference in cumulative rewards between the optimal policy and the agent's policy also decreases, thereby reducing $Regret(T)$ (defined in Definition 4).*

Combining Theorem 1 and Theorem 2, we conclude that reducing the episode length decreases the probability of entering a

dead-end state P_d, which in turn reduces the regret term, except when the episode length reduction preserves P_d. Building upon the theoretical insights established in this Section 4.1, we introduce the Adaptive Episode Length Adjustment (AELA) algorithm in the following section.

4.2 Algorithm : AELA

AELA leverages the relationship between episode length and secure exploration, as demonstrated by Theorems 1 and 2, to optimize learning efficiency. By adaptively controlling the episode length based on the agent's learning progress, AELA effectively minimizes the probability of encountering dead-end states, thereby reducing regret and enhancing overall performance in multi-agent environments. To summarize, we realized the advantages of training with reduced episode lengths. However, starting with shorter episodes requires adjusting their length as the agent needs to gather experiences beyond the reduced episode lengths. To address this, we propose an adaptive method within the AELA framework that incrementally increases the episode length based on the agent's learning progress. Specifically, the algorithm monitors the entropy of the learned policy or Q-values; a decreasing trend in entropy indicates that the agent's policy is converging toward stability. When such convergence is detected, the episode length is extended to allow the agent to gather more experiences beyond the initial secure exploration phase.

Given a set of Q-values $Q(s, a)$ for each action a in a state s, the action probability distribution $P(a)$ can be defined using the softmax function:

$$P(a \mid s) = \frac{\exp(Q(s, a)/\tau)}{\sum_a \exp(Q(s, a)/\tau)}, \tag{8}$$

where τ is the temperature parameter that controls the smoothness of the action probability distribution. If a policy-based method is used, the action probability distribution represents the policy π. The entropy H of the action distribution can then be calculated as:

$$H = -\sum_a P(a \mid s) \log P(a \mid s). \tag{9}$$

Then, we sum all the entropy values over a mini-batch:

$$H_{\text{total}} = \sum_{b=1}^{B} \sum_{t=1}^{T} \sum_{i=1}^{I} \left(-\sum_a P(a_t^{(b,i)} \mid s_t^{(b,i)}) \log P(a_t^{(b,i)} \mid s_t^{(b,i)}) \right), \tag{10}$$

where, H_{total} represents the total entropy accumulated across all batches, time steps, and agents. The variable B refers to the batch size, which indicates the number of independent episodes being processed simultaneously. The variable T denotes the number of time steps within each episode, and I is the number of agents in the multi-agent system. The action taken by agent i at time step t in batch b is denoted by $a_t^{(b,i)}$, while the state observed by the same agent at the same time step and batch is represented by $s_t^{(b,i)}$. The probability of taking action a given the state s for agent i at time step t in batch b is given by $P(a_t^{(b,i)} \mid s_t^{(b,i)})$.

The entropy H_{total} is collected over a window size of w. If the entropy is decreasing, the episode length is increased by 1 as:

$$E_L \leftarrow E_L + f(H_{\text{trend}}), f(H_{\text{trend}}) = \begin{cases} 1 & \text{if } \alpha < 0, \\ 0 & \text{otherwise,} \end{cases} \tag{11}$$

where $H_{\text{trend}} = \alpha H_{total} + \beta$ represents the linear fitting function for the entropy values in a window. If the slope α is negative, $f(H_{\text{trend}})$ returns 1; otherwise, it returns 0. The detailed description of the algorithm is presented in Algorithm 1. This adaptive adjustment ensures that the agent benefits from reduced dead-end probabilities during the critical early learning stages while still being able to explore and learn from more complex scenarios in the latter parts of episodes.

5 EXPERIMENT

In this section, we present the experimental results of AELA under two well-known MARL test-beds: the StarCraft multi-agent challenge (SMAC) and modified predator-prey (MPP). We evaluated the effectiveness of our proposed algorithm by combining it with two widely known algorithms in MARL: VDN and QMIX. We named these combinations AELA-VDN and AELA-QMIX, respectively.
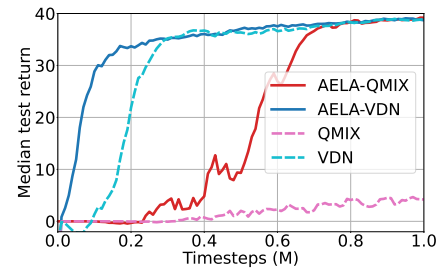
(a) $P = -2$
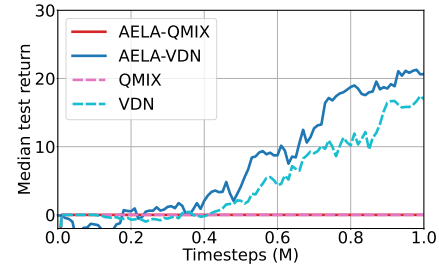
(b) $P = -4$

Figure 1: Median test return in the MPP tasks

5.1 Modified predator-prey

The modified predator-prey scenario is an extended version of the classical predator-prey problem, widely used in MARL research to evaluate agent coordination and learning capabilities. In this scenario, multiple predators collaborate to capture prey within an environment, emphasizing the need for cooperative learning. The modified predator-prey scenario employs a more intricate reward structure compared to the original. If two or more predators catch the prey, a reward of +10 is given, whereas if only one predator
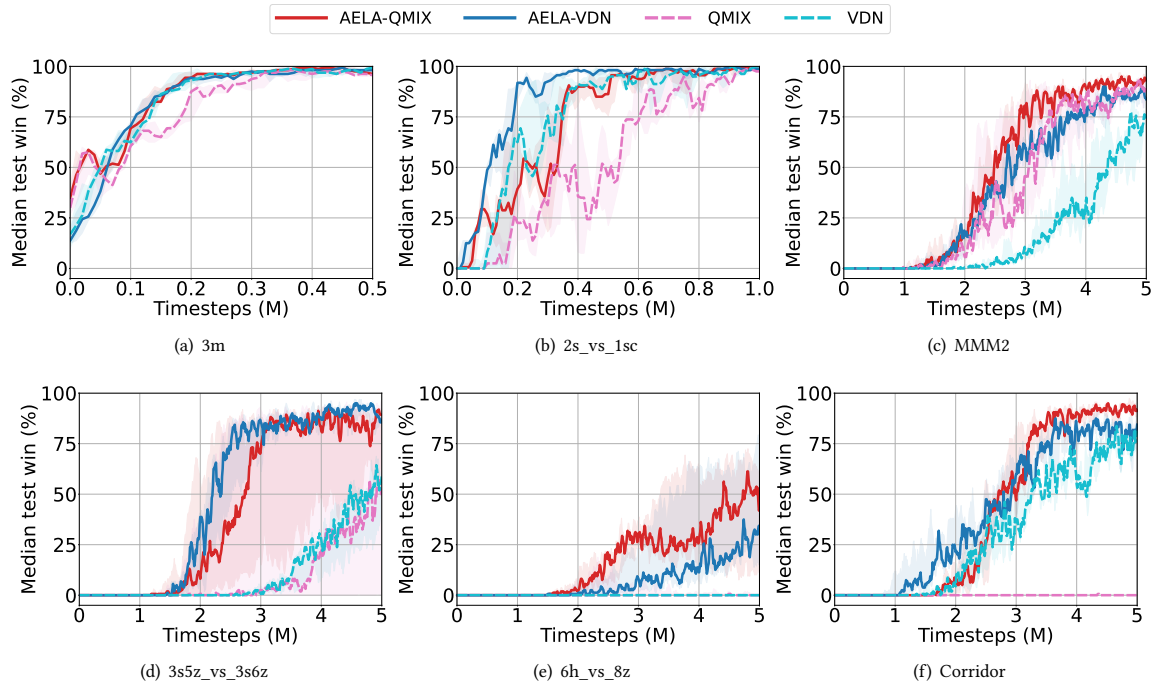
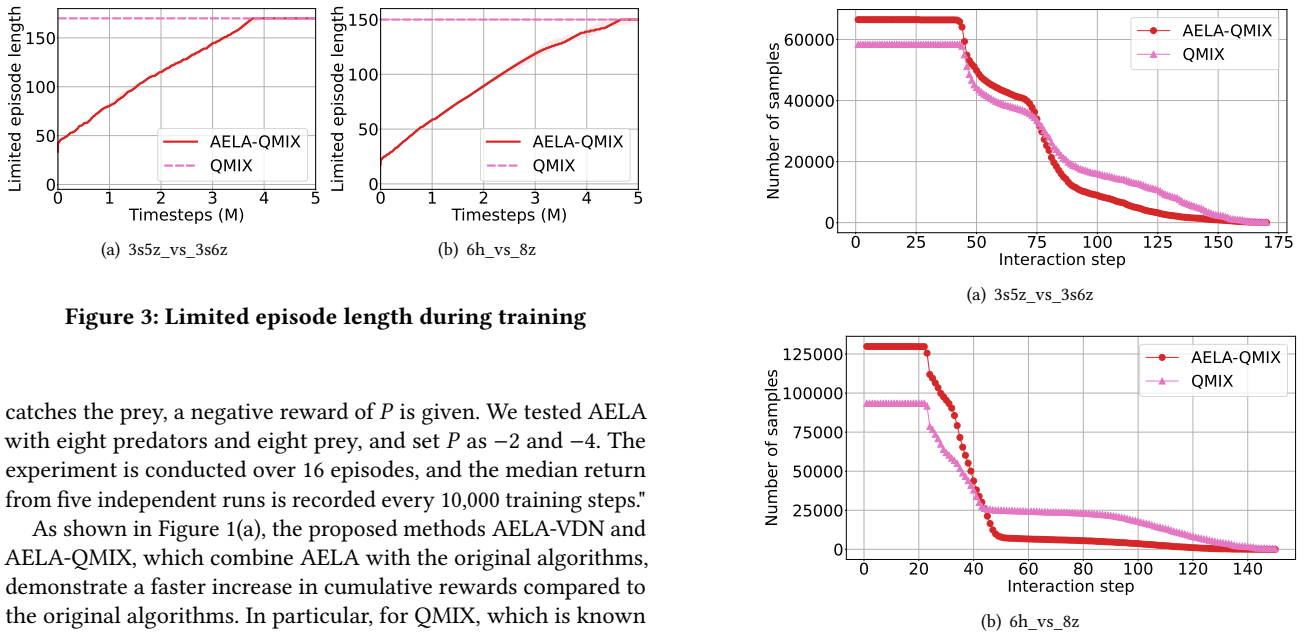Figure 2: Median test win rates with different SMAC scenarios

Figure 3: Limited episode length during training

Figure 4: Number of samples with interaction steps

catches the prey, a negative reward of P is given. We tested AELA with eight predators and eight prey, and set P as -2 and -4. The experiment is conducted over 16 episodes, and the median return from five independent runs is recorded every 10,000 training steps."

As shown in Figure 1(a), the proposed methods AELA-VDN and AELA-QMIX, which combine AELA with the original algorithms, demonstrate a faster increase in cumulative rewards compared to the original algorithms. In particular, for QMIX, which is known to be highly sensitive to the order in which states are visited [4, 9, 31], performance significantly improves after combining with AELA due to its ability to prevent the repetition of incorrect actions early in an episode by limiting the episode length. In the scenario with $P = -4$ depicted in Figure 1(b), only VDN found a policy that achieves high cumulative rewards, and even here, AELA-VDN exhibited a faster rise in cumulative rewards. Although there may be cases where a good policy is not found depending on the value

decomposition method, the results in Figure 1(b) are still meaningful as the experimental findings show no degradation in performance, consistent with the theoretical background discussed in Section 4.1.
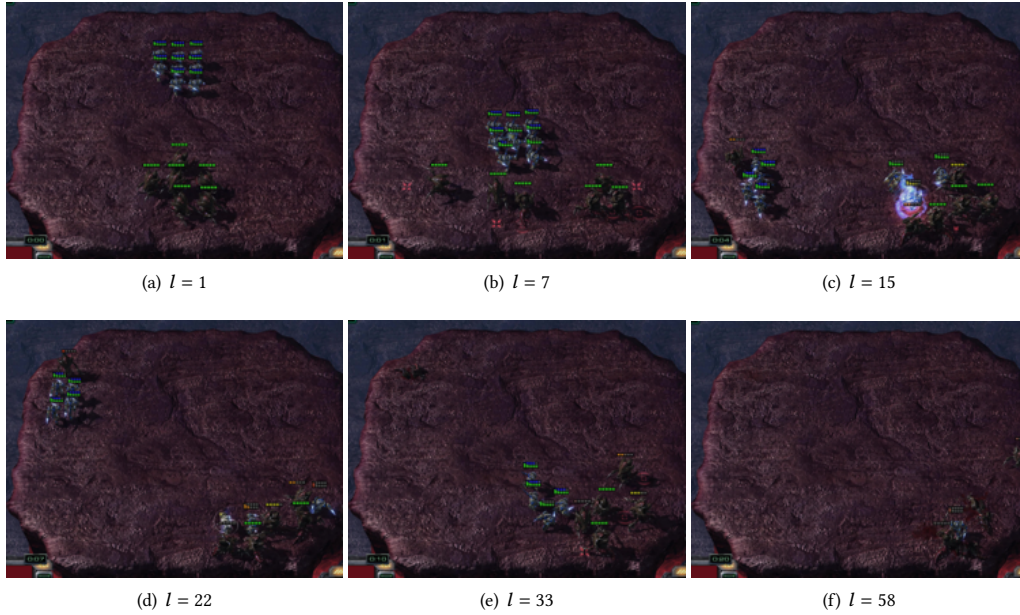
(a) $l = 1$ (b) $l = 7$ (c) $l = 15$

(d) $l = 22$ (e) $l = 33$ (f) $l = 58$

Figure 5: Snapshot of the final policy for AELA-QMIX in 6h_vs_8z with interaction steps l
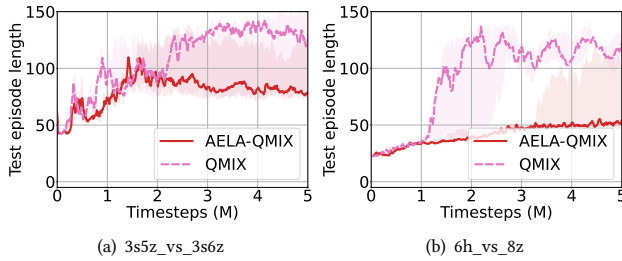

(a) 3s5z_vs_3s6z   (b) 6h_vs_8z

Figure 6: Interaction step at which an episode ends when following the current best policy during testing

5.2 StarCraft multi-agent challenge

In SMAC, agents are tasked with solving micromanagement scenarios where they must coordinate their actions to defeat opponent units controlled by the StarCraft II game engine. These scenarios vary in difficulty, unit types, and the number of agents involved, offering various challenges for evaluating MARL methods. The reward function in SMAC is typically defined based on damage dealt, units killed, and victory in the game, thus incentivizing effective coordination and combat tactics. Agents must make decisions with limited observability, requiring efficient exploration and accurate value estimation to achieve optimal performance. We performed five runs per scenario, and in each run, we conducted 32 tests every 10,000 training steps to calculate the win rate.

Figure 2 shows the median win rate over five runs for VDN, QMIX, AELA-VDN, and AELA-QMIX. In the relatively easier scenarios, such as *3m* and *2s_vs_1sc*, both methods reach a final win rate of nearly 100%, but the convergence speed is faster when AELA

is applied. In the remaining, even more challenging scenarios, AELA leads to improved final performance. Notably, in the *6h_vs_8z* and *Corridor* scenarios, the win rate with the original QMIX was close to zero, whereas AELA-QMIX achieved a significantly higher win rate.

Interestingly, performance improvements were achieved solely by adaptively limiting the episode length based on the learning situation without applying additional intrinsic rewards or other complicated value factorization mechanisms. We analyze that AELA improves performance by focusing on exploring diverse states early in the episode. Figure 3 shows the restricted episode length values applied as constraints during the training phase. With these restrictions applied, the number of samples in interaction steps was encouraged to be concentrated in the early part of the episode, as shown in Figure 4. Due to this intensive data collection early in the episode, AELA can reduce visitation to unnecessary states later in the episode, particularly dead-end states. We analyzed cases in SMAC where the presence of dead-end states prevents finding a winning policy. Figure 6 shows the episode length during testing. There is a tendency for QMIX to converge to a policy that has longer episode lengths, which is associated with dead-end states.

To examine this situation in more detail, a snapshot of the actual final policy is shown in Figure 5 and 7. As shown in Figures 5, in the *6h_vs_8z* scenario, an effective strategy involves having a few allied units lure some enemy units away. This prevents those enemy units from participating in the battle, allowing the remaining allied units to first attack the rest of the enemy units, and then finally engage the lured enemy units. However, as shown in Figure 7, if the strategy of luring enemy units is not properly learned, it can lead to an incorrect strategy where an allied unit moves away alone from the rest of the allied units, even when the enemy units are not
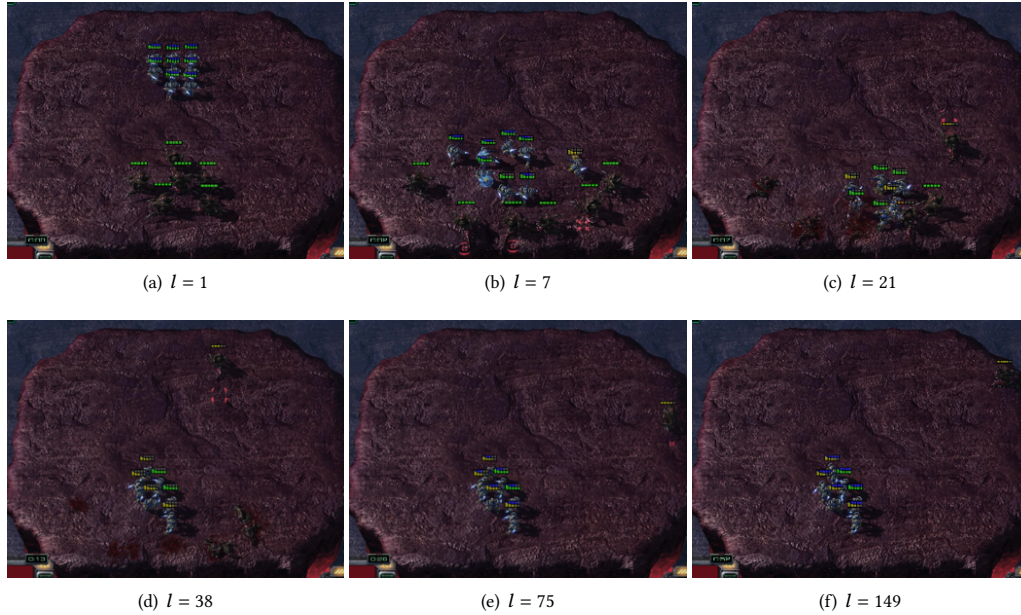
| (a) $l = 1$ | (b) $l = 7$ | (c) $l = 21$ |

| (d) $l = 38$ | (e) $l = 75$ | (f) $l = 149$ |

Figure 7: Snapshot of the final policy for QMIX in 6h_vs_8z with interaction steps l

following. Such a state, where all allied units die in combat against enemy units, leaving one isolated allied unit due to the failure of luring, can be considered a dead-end state. This is because a single allied unit cannot defeat all the remaining enemy units, making it impossible to transition to a winning state. Therefore, as shown in Figure 7, all states after $l = 38$ can be considered dead-end states. AELA mitigates the risk of entering dead-end states by terminating episodes at appropriate times and starting new episodes, which results in improved performance.

6 CONCLUSION

In this paper, we presented an adaptive approach to episode length adjustment for MARL. We theoretically demonstrated that limiting the episode length reduces the visitation probability of dead-end states, thereby allowing the learned value function to be closer to the optimal value function. Based on this background, we proposed a practical algorithm that initially limits the episode length and gradually extends it while assessing convergence through the entropy of Q-values. The proposed AELA algorithm was experimentally verified to outperform the traditional fixed episode length approach across various scenarios in multi-agent environments.

To the best of our knowledge, this paper is the first attempt to improve learning performance in multi-agent reinforcement learning by adjusting the episode length. Adjusting the episode length can be applied to any method, regardless of value decomposition techniques, additional intrinsic rewards, or network architecture, suggesting the potential for performance improvement from a new perspective. Future work could focus on extending this approach to more complex domains and incorporating additional metrics that could further enhance the adaptiveness of the episode length,

ensuring optimal policy convergence even in highly dynamic environments.

REFERENCES

[1] Joris Dinneweth, Abderrahmane Boubezoul, René Mandiau, and Stéphane Espié. 2022. Multi-agent reinforcement learning for autonomous vehicles: a survey. *Autonomous Intelligent Systems* 2, 1 (2022), 1–12.

[2] Mehdi Fatemi, Shikhar Sharma, Harm Van Seijen, and Samira Ebrahimi Kahou. 2019. Dead-ends and secure exploration in reinforcement learning. In *International Conference on Machine Learning*. PMLR, 1873–1881.

[3] Lior Fuks, Noor H Awad, Frank Hutter, and Marius Lindauer. 2019. An Evolution Strategy with Progressive Episode Lengths for Playing Games.. In *IJCAI*. 1234–1240.

[4] Tarun Gupta, Anuj Mahajan, Bei Peng, Wendelin Böhmer, and Shimon Whiteson. 2021. Uneven: Universal value exploration for multi-agent reinforcement learning. In *International Conference on Machine Learning*. PMLR, 3930–3941.

[5] Maximilian Hüttenrauch, Adrian Šošić, and Gerhard Neumann. 2017. Guided deep reinforcement learning for swarm systems. *arXiv preprint arXiv:1709.06011* (2017).

[6] Taylor W Killian, Sonali Parbhoo, and Marzyeh Ghassemi. 2023. Risk sensitive dead-end identification in safety-critical offline reinforcement learning. *arXiv preprint arXiv:2301.05664* (2023).

[7] Chuming Li, Jie Liu, Yinmin Zhang, Yuhong Wei, Yazhe Niu, Yaodong Yang, Yu Liu, and Wanli Ouyang. 2023. Ace: Cooperative multi-agent q-learning with bidirectional action-dependency. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 37. 8536–8544.

[8] Yuntao Liu, Yuan Li, Xinhai Xu, Donghong Liu, and Yong Dou. 2022. Rogc: Role-oriented graph convolution based multi-agent reinforcement learning. In

2022 IEEE International Conference on Multimedia and Expo (ICME). IEEE, 1–6.

[9] Anuj Mahajan, Tabish Rashid, Mikayel Samvelyan, and Shimon Whiteson. 2019. MAVEN: multi-agent variational exploration. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. 7613–7624.

[10] Debmalya Mandal, Goran Radanovic, Jiarui Gan, Adish Singla, and Rupak Majumdar. 2023. Online reinforcement learning with uncertain episode lengths. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37. 9064–9071.

[11] Nasik Muhammad Nafi, Raja Farrukh Ali, William Hsu, Kevin Duong, and Mason Vick. 2024. Policy Optimization using Horizon Regularized Advantage to Improve Generalization in Reinforcement Learning. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems*. 1427–1435.

[12] Fabio Pardo, Arash Tavakoli, Vitaly Levdik, and Petar Kormushev. 2018. Time limits in reinforcement learning. In *International Conference on Machine Learning*. PMLR, 4045–4054.

[13] Riccardo Poiani, Alberto Maria Metelli, and Marcello Restelli. 2023. Truncating trajectories in Monte Carlo reinforcement learning. In *International Conference on Machine Learning*. PMLR, 27994–28042.

[14] Riccardo Poiani, Nicole Nobili, Alberto Maria Metelli, and Marcello Restelli. 2024. Truncating trajectories in Monte Carlo policy evaluation: An adaptive approach. *Advances in Neural Information Processing Systems* 36 (2024).

[15] Wei Qiu, Xinrun Wang, Runsheng Yu, Rundong Wang, Xu He, Bo An, Svetlana Obraztsova, and Zinovi Rabinovich. 2021. RMIX: Learning risk-sensitive policies for cooperative reinforcement learning agents. In *Advances in Neural Information Processing Systems*, Vol. 34. 23049–23062.

[16] Tabish Rashid, Gregory Farquhar, Bei Peng, and Shimon Whiteson. 2020. Weighted qmix: Expanding monotonic value function factorisation for deep multi-agent reinforcement learning. In *Advances in neural information processing systems*, Vol. 33. 10199–10210.

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

[18] Mikayel Samvelyan, Tabish Rashid, Christian Schroeder de Witt, Gregory Farquhar, Nantas Nardelli, Tim GJ Rudner, Chia-Man Hung, Philip HS Torr, Jakob Foerster, and Shimon Whiteson. 2019. The StarCraft Multi-Agent Challenge. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*. 2186–2188.

[19] Siqi Shen, Chennan Ma, Chao Li, Weiquan Liu, Yongquan Fu, Songzhu Mei, Xinwang Liu, and Cheng Wang. 2023. RiskQ: risk-sensitive multi-agent reinforcement learning value factorization. *Advances in Neural Information Processing Systems* 36 (2023), 34791–34825.

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

[21] Kyunghwan Son, Junsu Kim, Sungsoo Ahn, Roben D Delos Reyes, Yung Yi, and Jinwoo Shin. 2022. Disentangling Sources of Risk for Distributional Multi-Agent Reinforcement Learning. In *International Conference on Machine Learning*. PMLR, 20347–20368.

[22] Wei-Fang Sun, Cheng-Kuang Lee, and Chun-Yi Lee. 2021. DFAC framework: Factorizing the value function via quantile mixture for multi-agent distributional Q-learning. In *International Conference on Machine Learning*. PMLR, 9945–9954.

[23] Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinicius Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z Leibo, Karl Tuyls, et al. 2018. 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*. 2085–2087.

[24] Richard S Sutton and Andrew G Barto. 2018. *Reinforcement learning: An introduction*. MIT press.

[25] Jianhao Wang, Zhizhou Ren, Terry Liu, Yang Yu, and Chongjie Zhang. 2020. QPLEX: Duplex Dueling Multi-Agent Q-Learning. In *International Conference on Learning Representations*.

[26] Tonghan Wang, Heng Dong, Victor Lesser, and Chongjie Zhang. 2020. ROMA: Multi-Agent Reinforcement Learning with Emergent Roles. In *International Conference on Machine Learning*. PMLR, 9876–9886.

[27] T Wang, T Gupta, B Peng, A Mahajan, S Whiteson, and C Zhang. 2021. RODE: learning roles to decompose multi- agent tasks. In *Proceedings of the International Conference on Learning Representations*.

[28] Dayon Ye, Minji Zhang, and Yu Yang. 2015. A multi-agent framework for packet routing in wireless sensor networks. *sensors* 15, 5 (2015), 10026–10047.

[29] Xianghua Zeng, Hao Peng, and Angsheng Li. 2023. Effective and stable role-based multi-agent collaboration by structural information principles. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 37. 11772–11780.

[30] Xiao Zhang, Hai Zhang, Hongtu Zhou, Chang Huang, Di Zhang, Chen Ye, and Junqiao Zhao. 2023. Safe Reinforcement Learning With Dead-Ends Avoidance and Recovery. *IEEE Robotics and Automation Letters* 9, 1 (2023), 491–498.

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