A Self-Evolving Default Action for Cooperative Tasks with Continuous Action Space

Shuangyao Huang[1*†]

[1]Xi'an Jiaotong-Liverpool University
School of Internet of Things
shuangyao.huang@xjtlu.edu.cn

arXiv:2607.18597v1 [cs.LG] 21 Jul 2026

Abstract

Counterfactual credit assignment has proven effective in multi-agent reinforcement learning (MARL) for discrete action spaces, yet its extension to continuous-action cooperative tasks remains challenging. Existing methods that approximate the counterfactual baseline via Monte Carlo sampling often introduce bias into policy gradients and fail to guarantee convergence to local optima, as the sampled actions may not have been sufficiently trained. To address these limitations, we propose SAFE, a novel MARL framework that employs a counterfactual baseline conditioned on a self-evolving default action sampled from each agent's experience buffer. This design naturally extends to continuous action spaces without relying on additional simulations, reward models, or environment-specific prior knowledge. The baseline accurately quantifies each agent's contribution, and introduces no bias into the deterministic policy gradient, ensuring convergence to local optima. Extensive experiments on cooperative vehicular tasks demonstrate that SAFE consistently outperforms state-of-the-art models.

Introduction

Multi-agent reinforcement learning (MARL) has achieved notable success in simulated decentralized cooperative games, such as StarCraft II (Vinyals et al. 2017), yet its deployment in real-world continuous-control tasks, such as autonomous vehicular networking, remains limited. These applications present three critical challenges. First, these applications usually feature continuous control, precluding credit assignment schemes that are designed around discrete action space. Second, these applications typically require fully decentralized policies for safety and prompt decision-making, since reliable communication is not guaranteed under all circumstances due to environmental limitations. Last, these applications present large joint state and action spaces, requiring credit assignment schemes to be scalable under the centralized training with decentralized execution (CTDE) (Oliehoek, Spaan, and Vlassis 2008) paradigm.

Decentralized cooperative models that work with continuous action spaces have been largely limited to multi-agent deep deterministic policy gradient (MADDPG) (Lowe et al. 2017) and MAPPO (Yu et al. 2022), but they do not scale well under the CTDE paradigm, as it requires the joint observation and action of all agents as input. Credit assignment schemes bypass the scaling issues associated with joint observation and action space, through counterfactual baseline or value factorization. The value factorization schemes (Sunehag et al. 2017; Rashid et al. 2020) have limited representation capability as they impose restrictive monotonicity assumptions and do not extend naturally to continuous action space as agents select actions greedily by taking $\arg\max$ on their policy. On the other hand, counterfactual credit assignment (COMA) (Foerster et al. 2018) offers an alternative by quantifying each agent's contribution with a baseline that marginalizes out the agent's action from a centralized critic. However, COMA is inherently limited to discrete action spaces, and naive Monte Carlo extensions to continuous spaces introduce bias into policy gradients and lack convergence guarantees, because the sampled actions may be undertrained and the baseline expectation is not zero under the joint policy.

This raises a central research question: *How can we design a counterfactual baseline for continuous action spaces that accurately quantifies each agent's contribution while preserving unbiased policy gradients and convergence guarantees?* The success of counterfactual credit assignment (Foerster et al. 2018) suggests that the optimal baseline is derived by taking the expectation of the Q value of all available actions. In this way, the baseline is learned directly from agents' experiences, rather than reward models or user selected task-dependent default actions. However, this approach is limited to discrete action spaces, since the expectation presumes a finite set of available actions for each agent. One possible extension to continuous action spaces is to approximate the expectation with Monte Carlo sampling. However, unlike counterfactual values in discrete action spaces, the sampled actions may not have been sufficiently trained to the centralized critic. As a result, the values conditioned on these actions do not accurately quantify the contribution of an agent. Moreover, these undertrained sampled actions introduce biases to the joint policy gradients, since the expectation of the Q values conditioned on the sampled actions is not zero.

We answer this question by introducing SAFE (*S*elf-evolving default *A*ction *F*rom *E*xperiences), a novel MARL framework whose core innovation is a counterfactual baseline

*Corresponding author.

†Supported by XJTLU Research Development Fund: RDF-23-02-026.

Copyright © 2027, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

conditioned on a self-evolving default action sampled from each agent's experience replay buffer. Unlike prior work, this baseline naturally extends to continuous action space. As we proved mathematically, this baseline introduces zero bias into the deterministic policy gradient, ensuring convergence to local optima. Moreover, our experiments demonstrate that this baseline adapts dynamically as the agent's behavior evolves during training, with the sampled default action increasingly reflecting the agent's average performance. Empirically, on cooperative autonomous driving tasks with varying numbers of vehicles and obstacles, SAFE consistently outperforms state-of-the-art baselines including VDN, QMIX, and COMA. Ablation studies further confirm that the performance gains stem directly from the credit assignment mechanism, not merely from continuous action adaptation. Parameter analyses reveal that a single sampled default action forms a more effective baseline than averaging multiple samples, validating our design principle.

Related Work

Independent Q-learning (IQL) (Tan 1993) is a strong benchmark in MARL tasks with discrete action spaces. However, IQL has no convergence guarantees, as agents independently explore and update their policies, the environment becomes non-stationary from each agent's viewpoint. MADDPG (Lowe et al. 2017) exploits the CTDE paradigm that allows agents to access all local information and global state in training, addressing the non-stationary problem. However, it does not scale well with the number of agents as the input dimension grows exponentially.

Credit assignment schemes provide better solutions for large scale multi-agent tasks. Explicit credit assignment schemes such as counterfactual multi-agent policy gradients (COMA) (Foerster et al. 2018) and Shapley Q-learning (Wang et al. 2020; Li et al. 2021) explicitly derive an advanced function for each agent that evaluates the contribution of the agent's action in the cooperative task. COMA estimates an agent's contribution with a baseline defined by the expectation of counterfactual Q values. The agent's policy is then updated to maximize the difference rewards (Wolpert and Tumer 2001) between the actual Q value and this baseline. However, they show poor performance in practice due to the sample inefficiency of the on-policy training approach of COMA and the high computational complexity in deriving the baseline with Shapley Q values.

On the other hand, implicit credit assignment schemes such as value-decomposition networks (VDN) (Sunehag et al. 2017) and QMIX (Rashid et al. 2020) factorize the joint action values as a monotonic combination of individual action values based on the individual-global-max (IGM) condition, which guarantees consistency between the joint and individual action values. However, value factorization leads to unbounded divergence of individual value functions and unpredictable agent behaviors (Wang et al. 2021). Moreover, the representation capability of the mixing network is limited as it strongly depends on the assumption of monotonic relations. Most importantly, existing credit assignment schemes are specifically designed for discrete actions spaces and do not extend well to multi-agent tasks involving continuous action spaces.

Background

We consider a fully cooperative multi-agent setting in which a team of agents select sequential actions within a stochastic and partially observable environment. This problem can be formulated as a decentralized partially observable Markov decision process (Dec-POMDP). A Dec-POMDP is defined by a tuple $(\mathcal{N}, \mathcal{S}, \mathcal{A}, r, \mathcal{O}, \mathcal{P})$, where $\mathcal{N} \coloneqq \{1, \cdots, N\}$ denotes the set of N agents. $\mathcal{S}$, $\mathcal{O}$ and $\mathcal{A}$ are state, observation and action space of the environment, respectively. Lastly, r is a reward signal and $\mathcal{P}$ is the state transition probability of the environment.

At each step, each agent makes an observation $o_i \in \mathcal{O}$, $i \in \mathcal{N}$ on the environment, and selects an action $a_i \in \mathcal{A}$, $i \in \mathcal{N}$ based on its observation, yielding the joint action $\boldsymbol{a} \coloneqq \{a_i\}_{i=1}^{N} \in \mathcal{A}^N$. After all agents execute their actions $\boldsymbol{a}$ in state $s \in \mathcal{S}$ simultaneously, the environment transits from state s to s' with probability $P(s'|s, \boldsymbol{a}) : \mathcal{S} \times \mathcal{A}^N \times \mathcal{S} \to [0, 1]$, and returns a numerical reward $r(s, \boldsymbol{a}) : \mathcal{S} \times \mathcal{A}^N \to \mathbb{R}$. In addition, the history of each agent's observations and actions is denoted $\tau_i \coloneqq \{(o_i, a_i)_t\}_{t=1}^{T}$.

In fully cooperative tasks, all agents within the team receive one team reward. Eventually, each agent optimizes its policy $a_i = \pi(\tau_i)$ by maximizing a long-term team return defined by the accumulation of the team reward over time, which induces a joint action-value that follows the Bellman equation:

$$Q_{tot}(s_t, \boldsymbol{a}_t) = \mathbb{E}[\sum_{l=0}^{\infty} \gamma^l r_{t+l}] = r_t + \gamma \mathbb{E}[Q_{tot}(s_{t+1}, \boldsymbol{a}_{t+1})], \tag{1}$$

where $\gamma \in [0, 1)$ is a discount factor.

Deterministic Policy Gradient

Deterministic policy gradient (DPG) (Silver et al. 2014; Lillicrap et al. 2015) is an actor-critic method specifically adapted for continuous action space. Similar to Q-learning (Mnih et al. 2015), the actions are selected maximizing the optimal action-value function at each step. When the action space is continuous, the action-value function is presumed to be differentiable with respect to the action. Assuming the actions are selected by a deterministic policy $a_t = \pi(s_t)$ which maximizes the Q function, the optimal Q value can be approximated with:

$$\max_a Q(s_t, a) \approx Q(s_t, \pi(s_t)), \tag{2}$$

where $Q(s_t, \pi(s_t))$ is subject to Eq. (1).

With this approximation, the critic can be updated with gradient descent by solving:

$$\min_{\phi} \mathbb{E}_{(s,a,r,s') \sim \mathcal{D}} \left[\left(Q_\phi(s, a) - \left(r + \gamma Q_{\bar{\phi}}(s', \pi_{\bar{\theta}}(s')) \right) \right)^2 \right], \tag{3}$$

and the actor is optimized with gradient ascent by maximizing the critic's output:

$$\max_{\theta} \mathbb{E}_{s \sim \mathcal{D}}[Q_\phi(s, \pi_\theta(s))], \tag{4}$$

where $\mathcal{D}$ is a reply buffer, ϕ and θ are parameters of the critic and actor, and $\bar{\phi}$ and $\bar{\theta}$ are parameters of the target networks.

Counterfactual Credit Assignment

Counterfactual credit assignment (Foerster et al. 2018; Wang et al. 2020) derives a counterfactual baseline from the centralize critic for each agent. An advantage function is derived by subtracting the counterfactual baseline from the joint action value following the concept of difference rewards:

$$A_i(s, \boldsymbol{a}) = Q(s, \boldsymbol{a}) - b_i, \tag{5}$$

where $i \in \mathcal{N}$. The counterfactual baseline b_i is derived by marginalizing out an agent or its action from the centralized critic, and replacing with another agent or default actions. In COMA, b_i is obtained by replacing an agent's action with every possible individual action available to that agent, and then taking the expectation of all resulting critic values:

$$b_i = \sum_{\bar{a}_i \in \mathcal{A}} \pi(s, \bar{a}_i) Q(s, \boldsymbol{a}_{-i}, \bar{a}_i), \tag{6}$$

where $\pi(s, a)$ is a stochastic policy defined for discrete action space and $\boldsymbol{a}_{-i}$ denotes the joint action except the action taken by the current agent.

Lastly, the policy gradient is given by:

$$g_\theta = \mathbb{E}_\pi \left[\sum_{i \in \mathcal{N}} \nabla_\theta \log \pi_i(s, a_i) A_i(s, \boldsymbol{a}) \right], \tag{7}$$

with θ being the parameters of actors.

Given the fact that $Q(s, \boldsymbol{a})$ evaluates the global Q-value of the taken action and the baseline reflects the average result that can be obtained under all other possible actions of the agent, the advantage function in Eq. (5), therefore, reflects the advantage of the taken action over the average result.

It is worth noting that the effectiveness of the baseline in Eq. (6) is built on the assumption of a discrete action space $\mathcal{A}$. With this assumption, each action $\bar{a}_i \in \mathcal{A}$ has been executed a sufficient number of times by agent i for state s during training. As a result, the estimation of counterfactual Q value $Q(s, \boldsymbol{a}_{-i}, \bar{a}_i)$ is sufficiently trained. Consequently, the baseline b_i accurately quantifies the contribution of agent i.

Algorithm

Counterfactual credit assignment has gain considerable success in multi-agent settings with discrete action spaces. However, it leads to poor performances in continuous-action tasks, especially for real-world tasks that require fine-grained control where a subtle change in action can lead to drastically different outcomes. Approximating the expectation in Eq. (6) with Monte Carlo sampling does not guarantee performance as the sampled actions may not have been frequently executed during training. Moreover, the baseline approximated via Monte Carlo sampling introduces bias into the policy gradient and does not guarantee convergence to local optima of policies. This highlights the importance of developing a counterfactual baseline that quantifies the contribution of continuous actions. The remaining part of this section is dedicated to addressing the above difficulty.

First, this paper adopts a centralized critic conditioned on the true global state s and the joint action-observation histories $\boldsymbol{\tau}$. The centralized critic is feasible as it is used only during training and only the actor is needed for execution. Each actor conditions on its own action-observation histories τ_i. Parameter sharing is used among actors as agents have the same role in most cooperative applications, such as vehicular control. The network architectures are illustrated in Fig. 1a.

For credit assignment, this paper adopts a counterfactual baseline inspired by difference rewards, in which each agent learns from a shaped reward:

$$D_i = r(s, \boldsymbol{a}) - r(s, (\boldsymbol{a}_{-i}, \bar{a}_i)). \tag{8}$$

Eq. (8) compares the global true reward received after each step of an MDP process and a reward conditioned on a default action $\bar{a}_i$ of agent i. Any action a_i that improves D_i also improves the team reward $r(s, \boldsymbol{a})$ as $r(s, (\boldsymbol{a}_{-i}, \bar{a}_i))$ does not depend on a_i. However, Eq. (8) requires repeated access to a training simulator to estimate the value of actions $\boldsymbol{a}$ and $(\boldsymbol{a}_{-i}, \bar{a}_i)$ for the same given state s. This is infeasible when the state space is continuous as it dramatically increases the number of simulations. Moreover, it requires a carefully selected user-defined default action, which relies on in-depth knowledge of the task and must be modified on a per-task basis. Additionally, the default action should be selected such that the expectation of critic values with regard to the policy conditioned on the default action is zero, bringing no bias to the policy gradient. COMA (Foerster et al. 2018) implements difference rewards using a centralized critic, which avoids the difficulties associated with simulation overload and default action selection while ensuring policy convergence. However, its critic architecture supports only discrete action spaces.

A key idea of this paper is that difference rewards can be implemented using a centralized critic with a default action that is selected from the agent's experience. Specifically, the default action is chosen based on the probability density distribution of all previously executed actions and evolves adaptively as the agent learns from the environment. In this way, the approximation of difference rewards can be extended to continuous action space while remaining applicable to discrete action space. Moreover, the selection of default action does not rely on additional simulations, reward models, or environment dependent knowledge.

This paper trains a centralized critic that estimates the global Q value for the joint action $\boldsymbol{a}$ under a state s. For each agent i, a counterfactual baseline is derived from the centralized by replacing the a_i with a default action $\bar{a}_i(t)$ at training step t, while maintaining other agents' inputs unchanged:

$$b_i = Q(s, \boldsymbol{a}_{-i}, \bar{a}_i(t)),\ \bar{a}_i(t) \sim \mathcal{D}_t. \tag{9}$$

The default action $\bar{a}_i(t)$ is sampled uniformly from the agent's experience buffer $\mathcal{D}_t$ at training step t. Consequently, b_i estimates the average performance of agent i with increasing accuracy, as $\bar{a}_i(t)$ approaches the action that is most frequently executed by the agent as t increases. As a result, the values conditioned on these actions will be sufficiently trained and the advantage function A_i in Eq. (5) accurately quantify the contribution of agent i. The credit assignment with the self-evolving default action selected from the agents' experience buffer is illustrated in Fig. 1b.

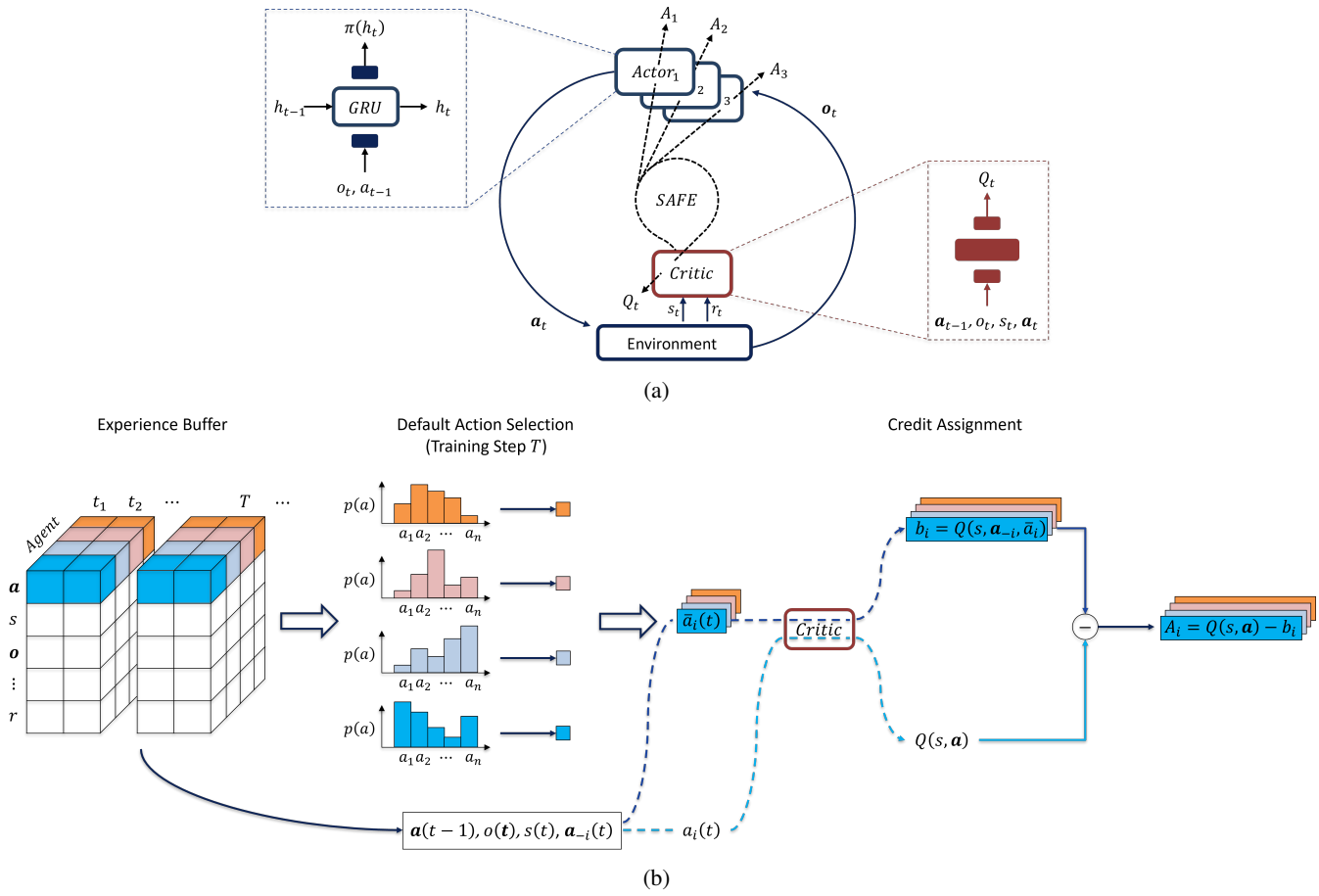

Figure 1: (a). Architectures of the centralized actors and the centralized critic, as while as the information flow between them for credit assignment in training. (b). Credit assignment with the self-evolving default action selected from the agents' experience buffer.

Next, we demonstrate that a policy gradient using our baseline converges to a local optimum in **Lemma** 0.1. In other words, the baseline does not affect the convergence of the deterministic policy gradient. Previous work on single agent actor-critic (Sutton et al. 2000) has shown that a baseline b does not affect the convergence of stochastic policy gradient if b is action-independent. COMA (Foerster et al. 2018) further proves that in multi-agent scenarios, a stochastic policy gradient is not affected if baseline b_i is a_i independent. A natural question arises here: *Can this conclusion be generalized to a deterministic policy gradient?* The authors in (Silver et al. 2014) have illustrated that deterministic policy π_θ is a special case of stochastic policy by writing the stochastic policy as $\mu_{\pi_\theta,\sigma}$, where σ represents a variance, such that

$$\lim_{\sigma \to 0} \mu_{\pi_\theta,\sigma} = \pi_\theta. \tag{10}$$

Let set $\mathcal{A}_\sigma = \{a_1, a_2, \cdots, a_n\}$ denote a subset of $\mathcal{A}$: $a_i \in \mathcal{A}$, for $i \in [1, n]$. By the definition of stochastic policy, $\sum_{a \in \mathcal{A}_\sigma} \mu_{\pi_\theta,\sigma}(a|\tau) = 1$. We have **Lemma** 0.1 that demonstrates our baseline ensures convergence of stochastic policy gradients and generalizes to deterministic policy gradients.

Lemma 0.1. *For a policy gradient at step t:*

$$g_t = \mathbb{E}_\mu \left[\sum_i \nabla_\theta \log \mu_{\theta,\sigma}(a_i|\tau_i) \cdot A_i \right], \qquad A_i = Q(s, \boldsymbol{a}) - b_i(s, \boldsymbol{a}_{-i}). \tag{11}$$

where b_i is an agent-specific baseline defined in Eq. (9)*, we have*

$$\lim_{t \to \infty} ||g_t|| = 0. \tag{12}$$

Proof. We need to only look at the expectation of the baseline b_i with regard to the current policy. Since $\mu_{\theta,\sigma}(\cdot|\tau)$ is a stochastic policy, we can write the joint policy of all agents as $\boldsymbol{\mu}(\boldsymbol{a}|s) = \prod_i \mu_i(a_i|\tau_i)$. Let $d^{\boldsymbol{\mu}}(s)$ be the discounted state distribution probability: $d^{\boldsymbol{\mu}}(s) = \sum_{t=0}^{\infty} \gamma^t Pr(s_t = s|s_0, \boldsymbol{\mu})$,

Figure 2: Highway cooperative autonomous driving tasks. Controlled vehicles are shown in green, while obstacle vehicles are depicted with blue.

the expectation of the baseline is written as:

$$
\begin{aligned}
g^{(1)} &= -\mathbb{E}_{\mu}\left[\sum_i \nabla_\theta \log \mu_i(a_i|\tau_i)\cdot b_i\right] \\
&= -\sum_s d^{\boldsymbol{\mu}}(s)\sum_i\sum_{\boldsymbol{a}^{-i}}\boldsymbol{\mu}^{-i}\cdot\sum_{a_i\in\mathcal{A}_\sigma}\mu_i(a_i|\tau_i)\nabla_\theta\log\mu_i(a_i|\tau_i)b_i \\
&= -\sum_s d^{\boldsymbol{\mu}}(s)\sum_i\sum_{\boldsymbol{a}^{-i}}\boldsymbol{\mu}^{-i}\cdot\underbrace{b_i\sum_{a_i\in\mathcal{A}_\sigma}\mu_i(a_i|\tau_i)\nabla_\theta\log\mu_i(a_i|\tau_i)}_{(b_i \text{ is not a function of } a_i)} \\
&= -\sum_s d^{\boldsymbol{\mu}}(s)\sum_i\sum_{\boldsymbol{a}^{-i}}\boldsymbol{\mu}^{-i}\cdot b_i\nabla_\theta\sum_{a_i\in\mathcal{A}_\sigma}\mu_i(a_i|\tau_i) \\
&= -\sum_s d^{\boldsymbol{\mu}}(s)\sum_i\sum_{\boldsymbol{a}^{-i}}\boldsymbol{\mu}^{-i}\cdot b_i\nabla_\theta 1 \\
&= \ 0.
\end{aligned}
$$

The reminder of Eq. (11) is simply the standard single-agent actor-critic policy gradient:

$$
g_t^{(2)} = \mathbb{E}_{\mu}\left[\nabla_\theta \log \boldsymbol{\mu}(\boldsymbol{a}|s)Q(s,\boldsymbol{a})\right]. \tag{13}
$$

It is shown in (Konda and Tsitsiklis 1999) that an actor-critic following this gradient converges to a local maximum of the objective function:

$$
\lim_{t\to\infty}||g_t^{(2)}|| = 0. \tag{14}
$$

Hence, b_i does not introduce bias in the policy gradient of μ, and therefore π, as $\mu \equiv \pi$ when $\sigma = 0$. □

Experimental Setup

Highway-Env (Leurent 2018) is a *Gym* compatible environment designed for cooperative vehicular tasks, including highway, intersection, and merging traffic scenarios. We focus on cooperative autonomous driving in highway scenario which is illustrated in Fig. 2, where the controlled vehicles (green) are driving on a multi-lane highway populated with obstacle vehicles (blue). In this task, the controlled vehicles travel from the left of the monitor to the right, while the obstacle vehicles travel in the opposite direction toward them. The controlled vehicles are positioned in close proximity to each other, such that even a minor change in heading may result in a collision.

During training, each agent observes a set of kinematic features for itself and its neighbors, including position, velocity, and heading. The action of an agent directly controls its steering angle in a range of $[-\pi/4, \pi/4]$ to maintain proximity to neighbors. The reward function for an agent includes collision penalty, and components encouraging formation and efficiency. The team reward is the sum of all agent rewards. More details can be found in the environment's documentation.

Experiments are conducted under four scenarios: 2-Vehicle-1-Obstacle (2V1O), 3-Vehicle-2-Obstacle (3V2O), 5-Vehicle-2-Obstacle (5V2O), and 7-Vehicle-2-Obstacle (7V2O). The benchmark methods compared are VDN, QMIX, IQL, COMA, and MAPPO. For models designed for discrete action spaces, the agents use discrete meta-actions, namely {0: *LEFT LANE*, 1: *IDLE*, 2: *RIGHT LANE*, 3: *FASTER*, 4: *SLOWER*}. All experiments are conducted with three independent runs on an NVIDIA RTX 5060Ti GPU with 16GB memory.

Network Structure and Training

The actor consists of 128-neuron gated recurrent units (GRU) (Cho et al. 2014) that use fully connected layers to process the input and to produce the output values from the hidden state. The agent's observation is fed into the input layer. Actions are produced from the final layer $\boldsymbol{z}$ via a $\tanh$ activation added by a random value μ controlled by parameter ϵ: $a_i = (1-\epsilon)\tanh(\boldsymbol{z})_i + \epsilon\mu$, where μ is sampled from the action space uniformly. Parameter μ is annealed exponentially from 1.0 to 0.05 across 50,000 training episodes. The centralized critic consists of multiple fully connected layers with ReLU activation functions. The global state, the agent's observation, and the all agents' joint actions are fed into the input layer. The output layer generates the global action value Q.

Parameter Study

In this experiment, we examine impact of the number of sampled actions in Eq. (9). In other words, we try to answer the question *Which one serves a better baseline, a single default action or the expectation of a group actions*? Mathematically, we examine the impact of K for agent i in:

$$
b_K = \mathbb{E}_{k=1}^{K} Q(s, \boldsymbol{a}_{-i}, \bar{a}_i^k),\ \bar{a}_i^k \sim \mathcal{D}. \tag{15}
$$

The key idea of SAFE is to estimates the average performance of an agent through a counterfactual baseline, which is derived from the centralized critic by replacing the agent's action with another action sampled from the agent's experience buffer. As the training goes on, the sampled action approaches the action that is most frequently executed by the agent. Therefore, the baseline gradually approximates the average performance of that agent in the task. To answer the above question, we compare SAFE_1 SAFE_{30}, SAFE_{50}, SAFE_{80}, and SAFE_{100}, where SAFE_K is deviced by baseline b_K and SAFE_1 is the default setting of SAFE.

Ablations

We conduct a number of ablations to show the effectiveness of the counterfactual baseline scheme in SAFE. In other words, we try to answer the question *Is the superior performance of SAFE achieved by its credit assignment scheme, or its adaptation to continuous action space*? To answer this question, we compare SAFE_1, the default setting, with the following baselines that also adapt to continuous action space but either adopt a different credit assignment scheme or a centralized critic:

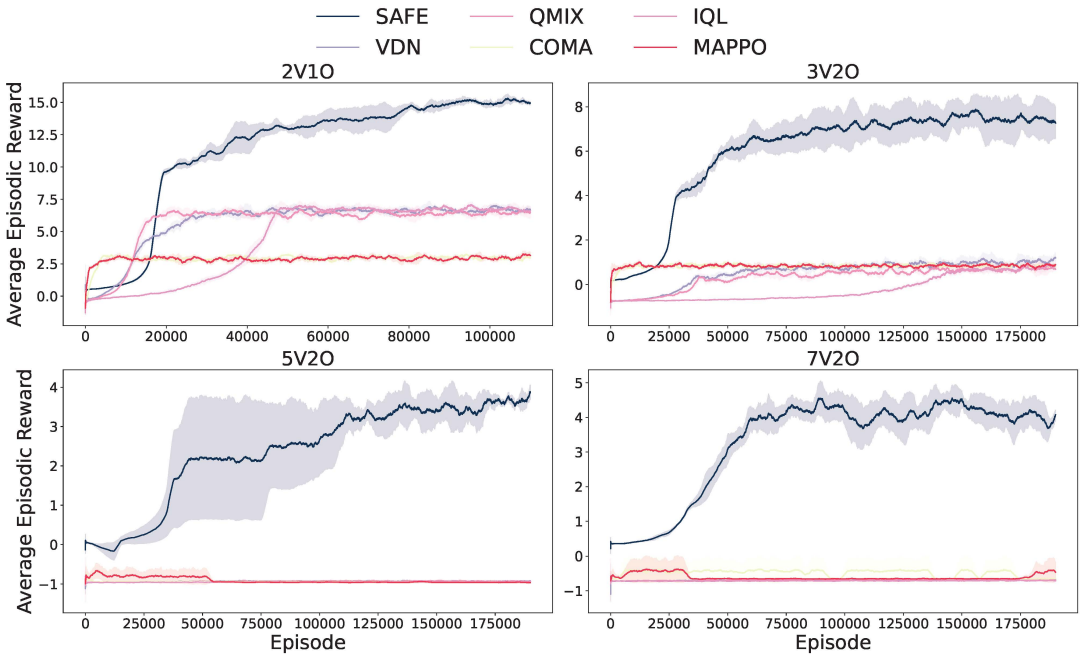

Figure 3: Comparison of SAFE against benchmark models on highway task, under four scenarios: 2-Vehicle-1-Obstacle (2V1O), 3-Vehicle-2-Obstacle (3V2O), 5-Vehicle-2-Obstacle (5V2O), and 7-Vehicle-2-Obstacle (7V2O).

- SAFE_a=0: Adopts the same baseline as in Eq. (9), except that the replacing action is set to a default action $a_i = 0$, rather than sampled from experience buffer.
- SAFE_batch_mean: Adopts the same baseline as in Eq. (9), except that the replacing action is set to the mean of a batch of 32 actions sampled uniformly from the experience buffer: $\bar{a}_i = \mathbb{E}_{k=1}^{32} \bar{a}_i^k,\ \bar{a}_i^k \sim \mathcal{D}$.
- COMA_cont: Continuous COMA, which adapts to continuous action space by approximating the baseline with Monte Carlo sampling.
- Centralized_critic: Multiple agent maintain a centralized critic and decentralized actors. The centralized critic is directly trained by the team reward, while the decentralized actors are updated by maximizing the centralized Q.

Experimental Results

Fig. 3 shows the performance of SAFE in the four scenarios in comparison against the benchmark models. Because agents spawn in close proximity, even minor actions can cause collisions. Thus, the task demands highly precise and sensitive adjustments to succeed. Consequently, all benchmarks with discrete action spaces fail to completely avoid collisions, achieving poor performance across all five scenarios. In contrast, SAFE achieves outstanding performance across all scenarios, owing to its compatibility with continuous action spaces. Only in scenario 2V1O (the simplest of the four) did VDN, QMIX, and IQL converge to stable policies. In this case, the two controlled vehicles can avoid collisions by simply executing the discrete actions *LEFT LANE* and *RIGHT LANE*. However, these models fail in all other scenarios, where more subtle and finely tuned actions are required to prevent collisions. The coarse granularity of discrete meta-actions quickly traps the policies in local minima. It is also worth noting that COMA and MAPPO fail to converge to optimal policies across all four scenarios, largely due to the sample inefficiency of on-policy training.

The self-evolving default action in our counterfactual baseline can be further examined through Fig. 4, which presents the probability density of default actions sampled from each agent's experience replay buffer. During the first 100 training episodes, agents 0 and 1 most frequently sample actions from the bin $[-0.04, +0.04)$. However, after 100,000 episodes, their sampled actions evolves to the bin $[-0.23, -0.14)$. In contrast, agent 2 initially concentrates its sampled actions in $[-0.23, -0.14)$ within the first 100 episodes, but evolves to $[-0.04, +0.04)$ after 100,000 episodes. The rationale behind this evolution is as follows: as training progresses, agents 0 and 1 gradually learn to steer leftward (taking actions $a \in [-0.23, -0.14)$) to avoid obstacle vehicles, while agent 2 tends to maintain its course (taking actions $a \in [-0.04, +0.04)$) to cooperatively avoid obstacles, as illustrated in Fig. 4a. Consequently, actions in $a \in [-0.23, -0.14)$ become dominant in the replay buffers of agents 0 and 1, whereas actions in $a \in [-0.04, +0.04)$ come to dominate agent 2's buffer. The default action sampled from the replay buffer effectively reflects the agents' average performance, as it corresponds to the most frequently executed action. In this regard, the distribution shown in Fig. 4b is consistent with a stochastic policy for a discrete action space.

Additionally, Fig. 5a demonstrates that *a single default action sampled from the agent's experience buffer forms a*

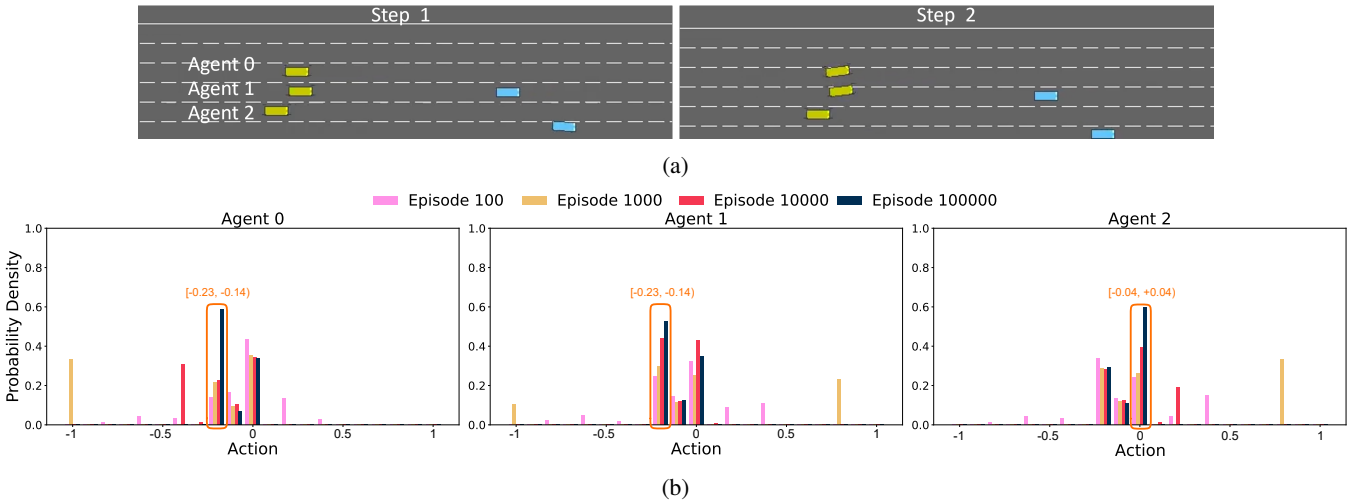

Figure 4: (a). Trajectories of vehicles after 150,000 training episodes. (b). Probability density distributions of each agent's default actions at 100, 1,000, 10,000, and 100,000 training episodes.

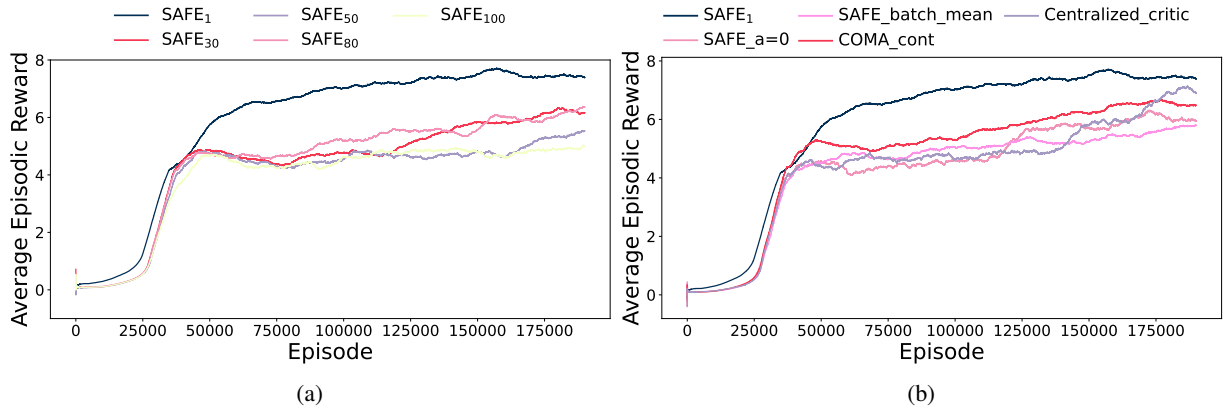

Figure 5: (a). Parameter study on the number of sampled actions in constructing our baseline. (b). Ablation study on the effectiveness of our counterfactual credit assignment scheme.

better baseline compared to the expectation of a group actions. Increasing the number of samples does not help with the estimation accuracy of the baseline in Eq. 9, and introduces additional biases yielding unsatisfactory results. On the other hand, Fig. 5b further demonstrates that *the superior performance of SAFE is achieved by its credit assignment scheme, rather than its adaptation to continuous action space*. While all the baseline models adapt to continuous action space of agents, only SAFE yields superior performance, owning to the accuracy of its credit assignment scheme. The experiments are conducted in scenario 3V2O, because the task is neither too simple nor too complex, making it suited for studying the subtle impact of parameters and baselines on model performance.

Conclusive Remarks

This paper presents SAFE, a novel MARL framework for cooperative tasks with continuous action spaces. The key innovation is a counterfactual baseline conditioned on a self-evolving default action sampled from each agent's experience buffer, which naturally extends credit assignment to continuous action space without additional simulations or task-specific priors. We prove theoretically that this baseline introduces no bias into the deterministic policy gradient, guaranteeing convergence to local optima. Experiments on cooperative autonomous driving tasks demonstrate that SAFE consistently outperforms state-of-the-art methods including VDN, QMIX, IQL, COMA, and MAPPO across scenarios of varying complexity. Ablation studies confirm that the performance gains stem from the credit assignment scheme itself rather than mere adaptation to continuous actions. Parameter analysis shows that a single sampled default action forms a better baseline than averaging multiple samples. These results establish SAFE as an effective solution for real-world multi-agent systems requiring continuous control.

References

Cho, K.; Van Merriënboer, B.; Bahdanau, D.; and Bengio, Y. 2014. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*.

Foerster, J.; Farquhar, G.; Afouras, T.; Nardelli, N.; and Whiteson, S. 2018. Counterfactual multi-agent policy gradients. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Konda, V.; and Tsitsiklis, J. 1999. Actor-critic algorithms. *Advances in neural information processing systems*, 12.

Leurent, E. 2018. An Environment for Autonomous Driving Decision-Making. https://github.com/eleurent/highway-env.

Li, J.; Kuang, K.; Wang, B.; Liu, F.; Chen, L.; Wu, F.; and Xiao, J. 2021. Shapley Counterfactual Credits for Multi-Agent Reinforcement Learning. *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

Lillicrap, T. P.; Hunt, J. J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; and Wierstra, D. 2015. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*.

Lowe, R.; Wu, Y. I.; Tamar, A.; Harb, J.; Pieter Abbeel, O.; and Mordatch, I. 2017. Multi-agent actor-critic for mixed cooperative-competitive environments. *Advances in neural information processing systems*, 30.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; Ostrovski, G.; et al. 2015. Human-level control through deep reinforcement learning. *nature*, 518(7540): 529–533.

Oliehoek, F. A.; Spaan, M. T.; and Vlassis, N. 2008. Optimal and approximate Q-value functions for decentralized POMDPs. *Journal of Artificial Intelligence Research*, 32: 289–353.

Rashid, T.; Samvelyan, M.; De Witt, C. S.; Farquhar, G.; Foerster, J.; and Whiteson, S. 2020. Monotonic value function factorisation for deep multi-agent reinforcement learning. *Journal of Machine Learning Research*, 21(178): 1–51.

Silver, D.; Lever, G.; Heess, N.; Degris, T.; Wierstra, D.; and Riedmiller, M. 2014. Deterministic policy gradient algorithms. In *International conference on machine learning*, 387–395. Pmlr.

Sunehag, P.; Lever, G.; Gruslys, A.; Czarnecki, W. M.; Zambaldi, V.; Jaderberg, M.; Lanctot, M.; Sonnerat, N.; Leibo, J. Z.; Tuyls, K.; et al. 2017. Value-decomposition networks for cooperative multi-agent learning. *arXiv preprint arXiv:1706.05296*.

Sutton, R. S.; McAllester, D. A.; Singh, S. P.; and Mansour, Y. 2000. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, 1057–1063.

Tan, M. 1993. Multi-agent reinforcement learning: Independent vs. cooperative agents. In *Proceedings of the tenth international conference on machine learning*, 330–337.

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

Wang, J.; Ren, Z.; Han, B.; Ye, J.; and Zhang, C. 2021. Towards understanding cooperative multi-agent q-learning with value factorization. *Advances in Neural Information Processing Systems*, 34: 29142–29155.

Wang, J.; Zhang, Y.; Kim, T.-K.; and Gu, Y. 2020. Shapley Q-Value: A Local Reward Approach to Solve Global Reward Games. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05): 7285–7292.

Wolpert, D. H.; and Tumer, K. 2001. Optimal payoff functions for members of collectives. *Advances in Complex Systems*, 4(02n03): 265–279.

Yu, C.; Velu, A.; Vinitsky, E.; Gao, J.; Wang, Y.; Bayen, A.; and Wu, Y. 2022. The surprising effectiveness of ppo in cooperative multi-agent games. *Advances in neural information processing systems*, 35: 24611–24624.