COORDINATED STRATEGY IDENTIFICATION MULTI-AGENT REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

An agent's strategy can be considered as a subset of action spaces, specialized in certain goals. This paper introduces a coordinated Strategy Identification Multi-Agent reinforcement learning (MARL) with episodic memory, called SIMA. SIMA derives a new temporal difference (TD) target to increase the sample efficiency. The efficiency is achived by keeping the best returns and corresponding to the best joint strategies for given states. This TD target with an additive strategy mixer automatically switches between an episodic control and a conventional Q-learning according to the existence of similar memories. In addition, each agent needs to behave similarly according to its strategy trajectory for coordinated behaviors among agents and coherent evaluation of a group's joint strategies. To this end, SIMA introduces a theoretical regularization for action policies to maximize the mutual information between an agent's trajectory and its specified strategy. We demonstrate its significant performance improvement on the StarCraft Multi-Agent Challenge benchmark.

1 INTRODUCTION

Recently, cooperative multi-agent reinforcement learning (MARL) has drawn increasing interest, and cooperative MARL has been adopted to many applications including traffic control (Wiering et al., 2000), resource allocation (Dandanov et al., 2017), robot path planning (Wang et al., 2020a), and production systems (Dittrich & Fohlmeister, 2020), etc. In spite of these successful applications, cooperative MARL still has challenges in learning proper coordination among multiple agents.

The framework of centralized training and decentralized execution (CTDE) (Oliehoek et al., 2008; Gupta et al., 2017) is proposed to overcome the partial observability and non-stationarity due to simultaneous learning of other agents in MARL. CTDE enables a decentralized execution while fully utilizing a global information during a centralized training, so that the agents learn their policies efficiently by accessing the global information at the training stage. Especially, value factorization approaches (Sunehag et al., 2017; Rashid et al., 2018; Son et al., 2019; Wang et al., 2020b) assume the consistency between the greedy selection of individual agent and the joint greedy action selection as a group, and these assumption is generally accepted by achieving the state-of-the-art performance on difficulty multi-agent tasks, such as StarCraft II Multi-agent Challenge (SMAC) (Samvelyan et al., 2019). However, this assumption on consistent greedy selection tends to limit exploration during training, and subsequently, the trained models fall into local optima particularly in hard tasks. This limitation on exploration becomes detrimental when agents need to search through a large joint action-observation space.Hence, researchers provided committed exploration mechanism under this CTDE training practice (Mahajan et al., 2019; Wang et al., 2019; Liu et al., 2021).

Independent to devising mechanisms for explorations, another approach resolves the problem by decomposing the task into sub-tasks (Ghosh et al., 2018; Wang et al., 2021; Sun et al., 2020), so the large search space can be limited in the training. The task decomposition is effective in cooperative setting with clear separations of strategies among agents because the learning can be separated, i.e. across the strategy selection and the action policy learning in a hierarchical setting. Here, *Strategy* [1] can be viewed as a subset of action spaces, where each strategy is specialized in a certain functionality for an agent's performance. To dynamically determine a proper strategy of each agent, (Wang

[1] A strategy can be identically considered as a decomposed action space, a sub-task or a role. In our context, we use strategy as a representative terminology.

et al., 2021) adopts a hierarchical framework, where an upper-tier network selects an agent's strategy; and where a lower-tier network chooses agent's actions conditioned on a designated strategy. This hierarchical separation of learning objectives creates a learning hurdle because such separation will interacting two convergence trajectories, not a unified one(Mahajan et al., 2019). This hurdle becomes materialized as either longer convergence time or limited performances via immature parameter inference. Moreover, the learning of dual objectives becomes more complex if the cooperative MARL does not provide a clear agent-wise credit assignment.

To solve the inefficient hierarchical learning problem in cooperative MARL, we propose an efficient sample utilization method, a.k.a. coordinated Strategy Identification Multi-Agent reinforcement learning (SIMA). SIMA adopts episodic control (Lin et al., 2018; Zheng et al., 2021) to expedite the learning by increasing sample efficiency of key episodes with the best return. SIMA utilizes the episodic memory 1) for regularizing the joint Q-learning, and 2) for explicitly evaluating agent's strategy selection. To this end, we decompose a common reward into individual rewards by comparing with the best joint strategies in the episodic memory. Then, a new temporal difference (TD) target structure is designed to determine whether to utilize episodic control or a conventional Q-learning, according to the existence of similar memories in the episodic buffer. In conjunction to the new TD target, SIMA regularizes the action selection of the lower-tier network to strengthen the coherent behavior across agents with the same strategy. Subsequently, the problem of dual objectives can be limited because the action policy learning becomes conditional to the strategy selection.

We evaluate SIMA on StrarCraft II micromanagement tasks (Samvelyan et al., 2019). Empirical results demonstrate that the proposed method show the improved or comparable performance compared to the state-of-the-art baseline methods. Ablation studies and qualitative analysis provide the enablers of this strategy selection mechanism in the performance enhancement.

2 Related Works

Multi-agent Exploration Balancing exploration and exploitation in policy learning is a paramount issue in reinforcement learning. To encourage exploration, modified count-based methods (Bellemare et al., 2016; Ostrovski et al., 2017; Tang et al., 2017), prediction error-based methods (Stadie et al., 2015; Pathak et al., 2017; Burda et al., 2018; Kim et al., 2018), and information gain-based methods (Mohamed & Jimenez Rezende, 2015; Houthooft et al., 2016) have been proposed for a single agent reinforcement learning. In most cases, an incentive for exploration is introduced as an intrinsic reward of a TD target in Q-learning; or such incentive is added as a regularizer for overall loss functions. Recently, various aforementioned methods to encourage exploration have been adopted to the multi-agent setting (Mahajan et al., 2019; Wang et al., 2019; Jaques et al., 2019; Mguni et al., 2021), and still the past methods have shown their effectiveness. Mahajan et al. (2019) introduce a regularizer maximizing the mutual information between trajectories and latent variables to learn a diverse set of behaviors. LIIR (Du et al., 2019) learns a parameterized individual intrinsic reward function by maximizing a centralized critic. Chenghao et al. (2021) propose a novel information-theoretical objective to maximize the mutual information between agents' identities and trajectories to encourage diverse individualized behaviors. Our model, SIMA, proposes a theoretic regularization maximizing the mutual information between an agent's trajectory and its designated strategy as an incentive to encourage coordinated behavior.

Episodic Control Episodic control stores the best return of either given state or state-action pair to efficiently estimate its values or Q-values. This estimation on true values requires sample efficiency, given that the sample generation is often limited by simulation executions or real-world observations, (Blundell et al., 2016; Pritzel et al., 2017; Lin et al., 2018). NEC (Pritzel et al., 2017) uses a differentiable neural dictionary as an episodic memory to estimate the action value by the weighted sum of the values in the memory. EMDQN (Lin et al., 2018) utilizes a fixed random matrix to generate a state representation, which is used as a key to link between the state representation and the highest return of the state in the episodic memory. EMC (Zheng et al., 2021) extends the approach of EMDQN to a deep MARL with curiosity-driven exploration incentives. EMC shows performance improvement via episodic control in MARL tasks, but EMC requires a hyperparameter tuning to determine the level of importance of the episodic memory based-target during training, according to the difficulties of the tasks. The proposed model, SIMA, relaxes this hyperparameter selection by adaptively switching TD targets from either conventional Q-learning or episodic control.

Hierarchical MARL Hierarchical reinforcement learning is a long-standing subject in reinforcement learning to address the sparse reward problem (Sutton & Barto, 2018). The hierarchical RL has been also widely applied to MARL problems (Lee et al., 2019; Yang et al., 2019; Jin & Ma, 2018). Tang et al. (2018) propose a temporal abstraction to decompose the problem into a hierarchy of different time scales, encouraging agents to learn high-level coordination based on the independent skills learned at the low level. Mahajan et al. (2019) present a hierarchical structure, which creates a mixture of value and policy-based methods by introducing a latent space. Vezhnevets et al. (2020) presents a framework in which a top-level policy chooses strategy responses to opponents, yielding an efficient exploration of the strategy space. RODE (Wang et al., 2021) has a hierarchical structure 1) a role selector assigns a role to each agent every specific timestep, and 2) an agent explores to learn the policy conditioned on an assigned role. In these line of hierarchical RL, we have observed learning inefficiency originating from a differentiated RL components. Thus, SIMA presents a sample efficient method to the hierarchical framework. In addition, we observed another problem of learning instability due to differentiated RL components. To resolve this instability, SIMA presents a novel theoretic regularization to encourage correlation between strategy and action.

3 PRELIMINARY

Before detailed explanation on SIMA, we present the formulation on the multi-agent reinforcement learning, and the definition on *Strategy* in our context.

Decentralized POMDP A fully cooperative multi-agent task can be formalized by following the Decentralized Partially Observable Markov Decision Process (Dec-POMDP) Oliehoek & Amato (2016), $G = \langle I, S, A, P, R, \Omega, O, n, \gamma \rangle$, where I is the finite set of n agents; $s \in S$ is the true state of environment; $a_i \in A$ is the i-th agent's action forming the joint action $\boldsymbol{a} \in A^n$; $P(s'|s, \boldsymbol{a})$ is the state transition function; R is a reward function $r = R(s, \boldsymbol{a}, s') \in \mathbb{R}$; Ω is the observation space; O is the observation function generating an observation for each agent $o_i \in \Omega$; and finally, $\gamma \in [0, 1)$ is a discount factor. At each timestep, an agent has its own local observation o_i, and the agent selects an action $a_i \in A$. The current state s and the joint action of all agents $\boldsymbol{a}$ lead to a next state s' according to $P(s'|s, \boldsymbol{a})$. The joint variable of s, $\boldsymbol{a}$, and s' will determine the identical reward r across the multi-agent group. To overcome the partial observability in CTDE, each agent utilizes a local action-observation history $\tau_i \in T \equiv (\Omega \times A)$ for its policy $\pi_i(a|\tau_i)$ (Hausknecht & Stone, 2015). Also, we define the group trajectory as $\tau =< \tau_1, ..., \tau_n >$.

Strategy and Strategy Selector The fundamental gain of SIMA originates from the utilization of episodic memory in selecting an agent's strategy. Hence, SIMA requires a formal definition on *Strategy* to formulate the concept and its gain from selections.

Definition 1 (Strategy) Given a fully cooperative multi-agent task specified as Dec-POMDP, let $z \in Z$ be a strategy in a set of strategies Z. The strategy specifies a strategy-action space $A_z \subset A$ limiting available actions from the agent's policy, $\pi_z : T \times A_z \rightarrow [0, 1]$, where $\cup_z A_z = A$. It is feasible to have overlaps in action spaces across strategies, such as $|A_{z_j} \cap A_{z_k}| \geq 0$ for $z_j \neq z_k$.

Agents adopting the same strategy z share the policy π_z with a reduced action space, and such agents share their experience during training, leading to a faster policy learning. To determine a proper strategy-action space, A_z; we start from a framework, RODE, presented in Wang et al. (2021), which clusters actions according to an action's influence toward rewards and observation changes. First, each action, a_i, is embedded through an action encoder; so the corresponding representation vector is learned as h_{a_i}. Second, the representation vectors are clustered by K-means algorithm. Appendix D provides details on strategy-action space learning, corresponding strategy representation h_z[2] in our setting, and the structure of strategy selector. We denote a strategy selector as Q^s shared by all agents, and we simplify a i-th agent's result value of Eq.25 as Q^s_i. Let $z_{i,t}$ be the agent i's strategy at timestep t, then the joint strategy $z_{jt,t}$ is defined as $z_{jt,t} = [z_{i,t}]_{i=1}^n$.

[2] We abuse the notation of h_z without indicating the index of $z_i \in Z$, and we only utilize the subscript of z to emphasize the agent's strategy selection, i.e., z_i means an instance of z being selected as a strategy of agent i.

4 Methodology

This section presents SIMA (see Figure 1), a novel framework that finds coordinated strategies among agents with episodic memory. SIMA adopts a hierarchical learning framework for a strategy selector and an action policy. This hierarchical learning framework requires following innovations in the learning formulation.

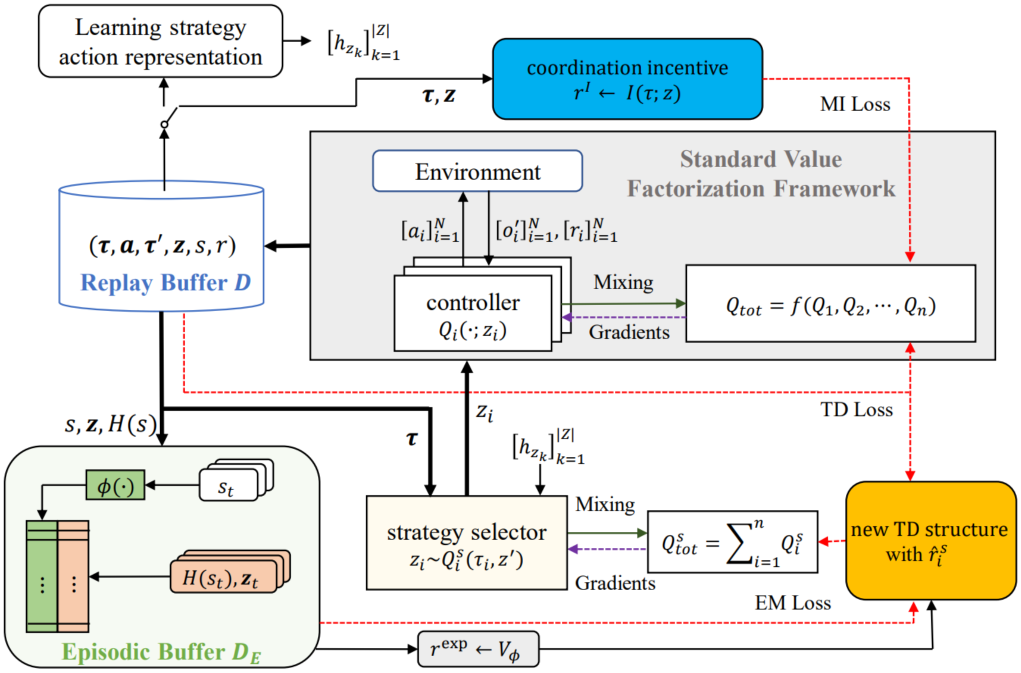

Learning strategy action representation
$[h_{z_k}]_{k=1}^{|Z|}$
$\boldsymbol{\tau}, \boldsymbol{z}$
coordination incentive
$r^I \leftarrow I(\tau; z)$
MI Loss
Standard Value Factorization Framework
Environment
$[a_i]_{i=1}^N$
$[o_i']_{i=1}^N, [r_i]_{i=1}^N$
Mixing
controller
$Q_i(\cdot; z_i)$
Gradients
$Q_{tot} = f(Q_1, Q_2, \cdots, Q_n)$
$(\boldsymbol{\tau}, \boldsymbol{a}, \boldsymbol{\tau}', \boldsymbol{z}, s, r)$
Replay Buffer D
TD Loss
$s, \boldsymbol{z}, H(s)$
$\boldsymbol{\tau}$
z_i
$[h_{z_k}]_{k=1}^{|Z|}$
$\phi(\cdot)$
s_t
$H(s_t), \boldsymbol{z}_t$
Episodic Buffer D_E
strategy selector
$z_i \sim Q_i^s(\tau_i, z')$
Mixing
Gradients
$Q_{tot}^s = \sum_{i=1}^n Q_i^s$
new TD structure with $\hat{r}_i^s$
EM Loss
$r^{\text{exp}} \leftarrow V_\phi$

Figure 1: Overview of SIMA framework

First, we need a new learning objective because our structure consists of the strategy selection and the action policy. Potentially, we are increasing the model complexity with introduced structure, so the sample efficiency needs to be improved. Therefore, for learning of a strategy selector, we derive a new temporal difference (TD) target for individual agents by utilizing episodic memory to boost the sample efficiency on multi-agent Q-learning. This TD target with a strategy mixer automatically switches between an episodic control and a conventional Q-learning according to the existence of similar memories. In addition, SIMA adopts a prediction-based intrinsic reward to encourage exploration on strategy selection to have a better trade-off between exploration and exploitation.

Second, SIMA designs a regularization to emphasize the hierarchical coordination. Once an agent selects a strategy, SIMA asks the agent to behave given the strategy boundary by limiting the action policy exploration. Meanwhile, SIMA promotes the strategy exploration by agents, so the exploration could be delegated to the strategy selector. Theoretic regularization is activated only after strategy is well defined via learning of strategy action representation as illustrated in Fig. 1.

4.1 Strategy Learning with Episodic Memory

The hierarchical framework simultaneously trains the strategy selector Q^s, being shared across agents; and an individual agent's action policy Q_i, being shared by the agents with the same strategy. This interaction between the agent group and the individual agents will require a long convergence time to find a proper joint strategy and an individual policy of agents.

Construction of Episodic Buffer To expedite learning by improving sample efficiency, SIMA memorizes $H(s_t)$, which is the highest return of a given global state s_t in episodic buffer $\mathcal{D}_E$ in Fig. 1. Besides of H, SIMA stores a joint strategy, $z^*_{jt,t} = [z^*_{i,t}]_{i=1}^n$, by paring it with H.

With a state embedding function $f_\phi(s) : S \rightarrow \mathbb{R}^k$ utilizing a fixed random matrix as a representation function to project states into a k-dimensional vector, a representation of global state s_t becomes $x_t = f_\phi(s_t)$. Here, x_t is used as a key to the highest return, $H(x_t)$, instead of s_t; and the corresponding best joint strategy $z^*_{jt,t}$. Similar to the episodic control in Lin et al. (2018), we update $H(x_t)$ with the following rules.

$$H(x_t) = \begin{cases} \max\{H(f_\phi(\hat{s}_t)), R_t(s_t, z_{jt,t})\}, & \text{if } ||\hat{x}_t - x_t|| < \delta \\ R_t(s_t, z_{jt,t}), & \text{otherwise} \end{cases} \tag{1}$$

where $R_t(s_t, z_{jt,t})$ is the return of a given $(s_t, z_{jt,t})$; and $\hat{x}_t = f_\phi(\hat{s}_t)$ is $x_t = f_\phi(s_t)$'s nearest neighbor in $\mathcal{D}_E$. Note that our episodic memory keeps the joint strategies instead of joint actions. If there is no similar projected state $\hat{x}_t$, such that $||\hat{x}_t - x_t|| < \delta$ in the memory; then $H(x_t)$ keeps the current $R_t(s_t, z_{jt,t})$.

Converting Group Reward into Individual Reward From the similar memories in the episodic memory between the current and next states, the corresponding best reward can be computed as

$$r^s(\hat{s}_t, z^*_{jt,t}) = H(f_\phi(\hat{s}_t)) - \gamma H(f_\phi(\hat{s}_{t+c})). \tag{2}$$

Here, $r^s_t = \Sigma_{t'=0}^{c-1} r_{t+t'}$ is a reward for a given strategy choice when an assigned strategy is maintained for c timesteps; and r_t is a common external reward from the environment without distinctions on either groups or agents. By comparing the current strategy $[z_{i,t}]_{i=1}^n$ and the best strategy $[z^*_{i,t}]_{i=1}^n$ in $\mathcal{D}_E$ for a given s_t, we can distinguish which agent's strategy results in a better or worse reward.

Additionally, r_t needs to be decomposed for group strategy. Therefore, SIMA generates $\hat{r}^s_i$ for an individual agent. If the current strategic reward is $r^s(s_t, z_{jt,t}) = \Sigma_{t'=0}^{c-1} r(s_{t+t'}, z_{jt,t})$, then we can derive the reward difference compared to the best reward of the current state from the episodic memory as

$$\Delta r^s(s_t, z_{jt,t}) = r^s(s_t, z_{jt,t}) - r^s(\hat{s}_t, z^*_{jt,t}). \tag{3}$$

Here, computing $r^s(\hat{s}_t, z^*_{jt,t})$ is possible only if both $\hat{s}_t$ and $\hat{s}_{t+c}$ exist. After that, the credit or the penalty of $\Delta r^s(s_t, z_{jt,t})$ is only given to agents whose current strategy $z_{i,t}$ is different from the best one $z^*_{i,t}$. To do so, we define a coefficient v_i to check whether an agent i's strategy has changed or not, by representing the change detection via the Dirac delta function, δ.

$$v_i = 1 - \delta\left(z_{i,t}, \hat{z}_{i,t}\right) \tag{4}$$

With this coefficient for the explicit credit assignment, we can derive an individual reward $\hat{r}^s_i$ as follows:

$$\hat{r}^s_i(s_t, z_{jt,t}) = \begin{cases} \frac{r^s(\hat{s}_t, z^*_{jt,t})}{n} + \frac{v_i}{\sum_i v_i} \Delta r^s(s_t, z_{jt,t}), & \text{if } \sum_i v_i \geq 1 \text{ and } \eta_t = 1 \\ \frac{r^s(\hat{s}_t, z^*_{jt,t})}{n}, & \text{elif } \sum_i v_i = 0 \text{ and } \eta_t = 1 \\ \frac{r^s(s_t, z_{jt,t})}{n}, & \text{otherwise} \end{cases} \tag{5}$$

where $\eta = 1$ if both $\hat{s}_t$ and $\hat{s}_{t+c}$ exist, $\eta = 0$ otherwise.

Formulating TD-Target with Individual Reward With the individual reward function $\hat{r}^s_i(s_t, z_{jt,t})$, we can derive a TD-target as $y_i = \hat{r}^s_i(s_t, z_{jt,t}) + \gamma \max_{z'} \bar{Q}^s_i(\tau_{i,t+1}, z')$. Here, $\bar{Q}^s$ is a target network of the shared strategy selector. By applying VDN mixer (Sunehag et al., 2017) for individual TD-targets, we can derive a following TD-target for a joint strategy Q^s_{tot} function.

$$y_{tot} = \sum_i^n \hat{r}^s_i(s_t, z_{jt,t}) + \gamma \max\left(\sum_i^n \max_{z'} \bar{Q}^s_i(\tau_{i,t+c}, z'), Q^*_{EC}\right) \tag{6}$$

Note that we set $Q^*_{EC} = H(f_\phi(\hat{s}_{t+c}))$ only if there exists $\hat{s}_{t+c}$ for a given s_{t+c} in the episodic memory and $z_{jt,t} = z^*_{jt,t}$. Otherwise, we set $Q^*_{EC} = \Sigma_i^n \max_{z'} \bar{Q}^s_i(\tau_{i,t+c}, z')$ by default. With the above TD target structure, Eq. 7 is the derived TD target for the strategy selector in SIMA.

$$y_{tot} = \begin{cases} r^s(s_t, z_{jt,t}) + \gamma \sum_i^n \max_{z'} \bar{Q}^s_i(\tau_{i,t+c}, z'), & \text{if } z_{jt,t} \neq z^*_{jt,t} \\ r^s(\hat{s}_t, z^*_{jt,t}) + \gamma H(f_\phi(\hat{s}_{t+c})), & \text{otherwise} \end{cases} \tag{7}$$

Here, y_{tot} becomes identical to a conventional TD target of Q-learning with VDN mixer when $z_{jt,t} \neq z^*_{jt,t}$. On the other hand, when $\Sigma_i^n \max_{z'} \bar{Q}_i^s(\tau_{i,t+c}, z') \leq H(\phi(\hat{s}_{t+c}))$ and $z_{jt,t} = z^*_{jt,t}$, y_{tot} becomes the TD-target of episodic control. The proposed TD-target structure (y_{tot}) stabilizes the learning of a strategy selector by maintaining its TD target value while action policy is being explored in the sample episode. In addition, with this target structure, we can automatically utilize an episodic control without introducing an additional hyperparameter to balance between sample and episodic TD errors. The final form of a loss function L_{s} for learning the strategy selector becomes

$$L_{\text{s}}(\theta_\tau, \theta_{emb}) = \mathbb{E}_{\tau, \boldsymbol{a}, \boldsymbol{z}, r, \tau' \in D}\left[\left(y_{tot} - \sum_i^N Q_i^s(\tau_{i,t}, z_{i,t})\right)^2\right]. \tag{8}$$

The TD-target presented in Eq.6, SIMA expedites and stabilizes the learning of a strategy selector when it is necessary as we show in Proposition 1. Appendix B presents the proof of Proposition 1.

Proposition 1. *Let $\nabla_{\theta_s}\tilde{L}_s$ be the optimal gradient of the loss in the entire training experience, toward true strategy value $Q^s_{tot}(s_t, z^*_{jt})$ with given s_t. Additionally, θ_s is a set of parameters from the strategy selector. Assuming 1) $z_{jt,t} = z^*_{jt,t}$; and 2) $\exists \hat{s}_t$ and $\exists \hat{s}_{t+1}$ given s_t and s_{t+1}; the proposed TD-target results in $\nabla_{\theta_s} L_s = \nabla_{\theta_s}\tilde{L}_s$ regardless of action policy π. On the other hand, a conventional TD-target deviates from optimal gradient, $\nabla_{\theta_s} L_s \neq \nabla_{\theta_s}\tilde{L}_s$.*

In addition, we implements a curiosity-driven incentive r^{exp} to balance between exploitation and exploration of the training on strategy selector. Appendix E provides the implementation details of r^{exp}.

4.2 Encouraging a coordinated behavior

The previous section derives a learning method for a strategy selector Q^s, which only utilizes a local information when selecting its strategy. During the learning of Q^s, the value of an individual strategy Q_i^s is evaluated assuming that other agents may behave in coordination according to their designated strategies. In other words, the evaluation of strategies becomes more accurate when agents act coherently with their given strategy trajectory $z_{0:T}$. To enforce this coherent agent behaviors with the same strategy, we introduce a novel theoretic regularization on policy learning to maximize the mutual information between the agent's trajectory and its designated strategy. We illustrate the concept of coordinated trajectory in Fig. 2, which depicts two agents' trajectories $(\tau_{i,0:T}, \tau_{j,0:T})$ with given strategy trajectories $(z_{i,0:T}, z_{j,0:T})$.

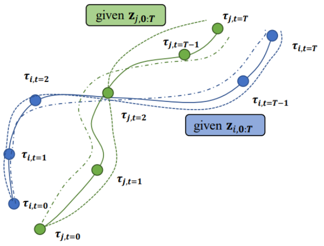


Figure 2: Coordinated trajectories with designated strategies

Let us consider the case of the agent i[3]. The mutual information of the agent's trajectory and its strategy trajectory can be expressed as

$$\begin{aligned} I^{\pi^s/\boldsymbol{\pi}}(\tau_T; z_{0:T}) &= \mathbb{E}_{z_{0:T}\sim\pi^s, \tau_T\sim\boldsymbol{\pi}}[\log \frac{p(\tau_T|z_{0:T})}{p(\tau_T)}] \\ &= \sum_{t=0}^{T-1} E_{z_t\sim\pi^s, \tau_t\sim\boldsymbol{\pi}}\left[\log \frac{p(a_t|\tau_t, z_t)}{p(a_t|\tau_t)}\right], \end{aligned} \tag{9}$$

where $p(\tau_T) = p(o_0)\prod_{t=0}^{T-1} p(a_t|\tau_t)p(o_{t+1}|\tau_t, a_t)$; π^s is a policy on strategy selection; and $\boldsymbol{\pi}$ is a joint action policy across all agents. Details of derivation are deferred to Appendix A.

Here $p(a_t|\tau_t, z_t)$ is the distribution determined by ε-greedy policy, and thus we adopt Boltzmann softmax of local value $Q(\cdot; z_t)$ to replace $p(a_t|\tau_t, z_t)$. Then, the lower bound of the summand of Eq.9 is derived as follows:

$$E_{z_t\sim\pi^s, \tau_t\sim\boldsymbol{\pi}}[\log \frac{p(a_t|\tau_t, z_t)}{p(a_t|\tau_t)}] \geq E_{z_t\sim\pi^s, \tau_t\sim\boldsymbol{\pi}}[\log \frac{\text{softmax}(\frac{1}{\beta_1}Q(a_t|\tau_t, z_t))}{p(a_t|\tau_t)}], \tag{10}$$

[3] We will remove the subscript i for simplicity in apparent local variables of agent i.

where β_1 is a control parameter. Note that since $D_{\mathrm{KL}}(p(a_t|\tau_t, z_t)||\mathrm{softmax}(\frac{1}{\beta_1}Q(a_t|\tau_t, z_t))) \geq 0$, the inequality holds. Hence, an introduced intrinsic reward at each step can be written as:

$$r^I = E_{z_t \sim \pi^s, \tau_t \sim \boldsymbol{\pi}}[D_{\mathrm{KL}}(\mathrm{softmax}(\frac{1}{\beta_1}Q(a_t|\tau_t, z_t))||p(a_t|\tau_t))]. \tag{11}$$

We can compute $\mathrm{softmax}(\frac{1}{\beta_1}Q(a_t|\tau_t, z_t))$ via local Q function, but $p(a_t|\tau_t)$ needs to be computed, as well. Similar to local Q, we can approximate $p(z'|\tau_t)$ via the strategy selector Q^s. Thus, the denominator of Eq.11 can be further expanded as

$$p(a_t|\tau_t) = \sum_{z' \in Z} p(z'|\tau_t)p(a_t|\tau_t, z') \simeq \sum_{z' \in Z} \mathrm{softmax}(\frac{1}{\beta_2}Q^s(z'|\tau_t))p(a_t|\tau_t, z'), \tag{12}$$

where β_2 is an additional control parameter. Again, since $p(a_t|\tau_t, z')$ can be approximated by $\mathrm{softmax}(\frac{1}{\beta_1}Q(a_t|\tau_t, z'))$, $p(a_t|\tau_t)$ can be expressed as

$$\begin{aligned} p(a_t|\tau_t) &= \sum_{z' \in Z} p(z'|\tau_t)p(a_t|\tau_t, z') \\ &\simeq \sum_{z' \in Z} \mathrm{softmax}(\frac{1}{\beta_2}Q^s(z'|\tau_t)) \cdot \mathrm{softmax}(\frac{1}{\beta_1}Q(a_t|\tau_t, z')). \end{aligned} \tag{13}$$

Hence, for a given trajectory τ_t, the intrinsic reward is determined as follows:

$$r^I = E_{z_t \sim \pi^s}\left[D_{\mathrm{KL}}\left(\mathrm{softmax}(\frac{1}{\beta_1}Q(\cdot|\tau_t, z_t))||\sum_{z' \in Z}\mathrm{softmax}(\frac{1}{\beta_2}Q^s(z'|\tau_t)) \cdot \mathrm{softmax}(\frac{1}{\beta_1}Q(\cdot|\tau_t, z')\right)\right]. \tag{14}$$

For the action policy Q, again we use a GRU to encode a local action-observation history τ into a vector h_τ. The local Q_i of the agent i is computed similar to Eq.25 in Appendix D with actions instead of a strategy. Subsequently, when generating the joint action-value function Q_{tot}, we adopt the mixer presented in QPLEX (Wang et al., 2020b), which guarantees a complete IGM condition. The final loss function for the action policy learning is determined as

$$L_{TD}(\theta) = \left(r + \beta_I \bar{r}^I + \gamma \mathrm{max}_{\boldsymbol{a}'}\bar{Q}_{tot}(s', \boldsymbol{a}') - Q_{tot}(s, \boldsymbol{a})\right)^2, \tag{15}$$

where β_I is a scale factor for the intrinsic reward encouraging a coordinated behavior; θ denotes parameters of networks related to action policy Q_i and the corresponding mixer network to generate Q_{tot}; $\bar{Q}_{tot}$ represents a target network; and $\bar{r}^I = 1/n\sum_{i=1}^{n} r_i^I$.

5 EXPERIMENTS

This section provides experimental results on SMAC (Samvelyan et al., 2019). The experiments compare SIMA against notable baselines, such as, value-based MARL methods (Rashid et al., 2018; Wang et al., 2020b), EMC adopting episodic control (Zheng et al., 2021), CDS encouraging individual diversity (Chenghao et al., 2021), and RODE, a role-based hierarchical approach, (Wang et al., 2021). We follow the practices of RODE in learning strategy action spaces with samples obtained during the first 50K timesteps.

5.1 PERFORMANCE ON STARCRAFT II

For a general performance evaluation, we test our methods on various maps, which require a different level of coordination according to the map difficulties. Winning rate is computed with 160 samples: 32 episodes for each training random seed, and 5 different random seeds. The median performance with the 25-75% percentiles are presented for all figures. Especially for a fair comparison, we set n_{circle}, the number of trainings per a sampled batch of 32 episodes during training, as 1 for all baselines since some of baselines increase $n_{\text{circle}} = 2$ as a default setting in their codes. Appendix C provides the further details of experiment settings.

Fig. 3 and Fig. 4 enumerate the performance on 1) easy and hard SMAC maps; and 2) super hard maps, respectively. This map categorization follows the practice of (Samvelyan et al., 2019). The proposed model reduces the performance variance and expedites the learning in easy and hard maps as in Fig. 3 compared to the previous hierarchical model (Wang et al., 2021). For super hard SMAC maps presented in Fig. 4, SIMA shows best or comparable performances against baseline methods. Moreover, SIMA achieves the best or the best-equivalent performances in all SMAC maps from the final convergence perspective.

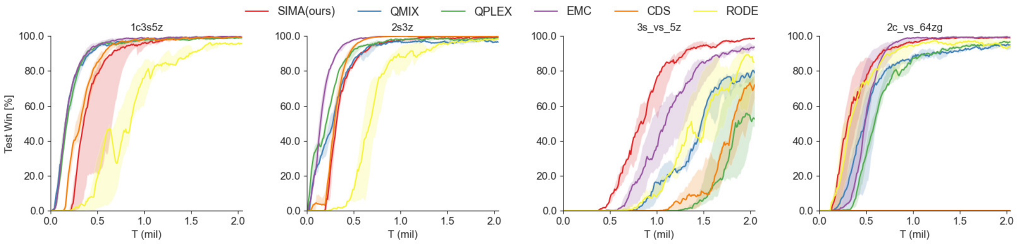


Figure 3: Performance comparison of SIMA against baseline algorithms on two **easy** SMAC maps: `1c3s5z` and `2s3z`, and two **hard** SMAC maps: `3s_vs_5z` and `2c_vs_64zg`.

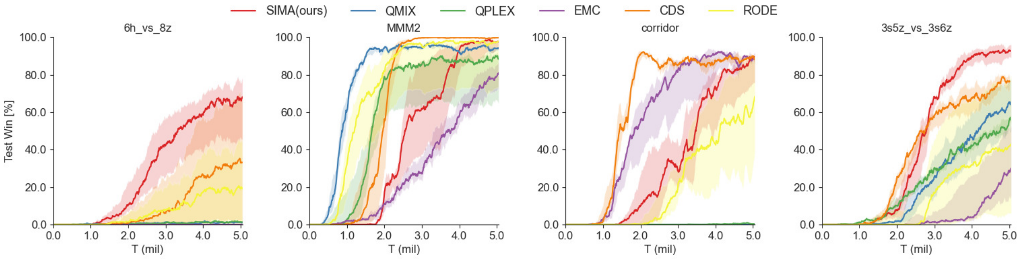
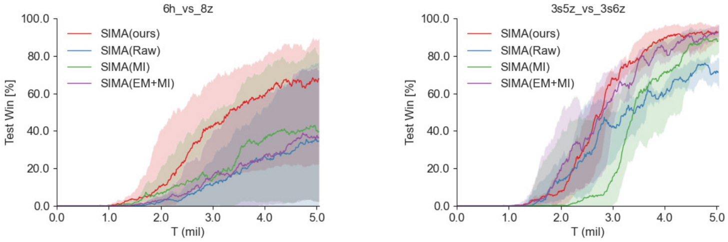


Figure 4: Performance comparison of SIMA against baseline algorithms on **super hard** SMAC maps.

5.2 Ablation Study and Qualitative Analysis

Ablation Study To understand the mechanism of SIMA by its components, we carried out ablation studies as illustrated in Fig. 5. SIMA(Raw) is the base structure by only utilizing 1) VND for strategy selector learning and 2) QPLEX for action policy learning. SIMA(MI) adopts a coordination incentive presented in Eq.14 for action policy learning. SIMA(EM+MI) additionally utilizes the loss function in Eq.8 for training on the strategy selector. SIMA(ours) contains all the proposed components including a curiosity incentive r^{exp}.

We identified different effects from components within SIMA through ablation studies. When strategy action spaces are not distinctively decomposed, such as a task in Fig. 5.(a), the learning of strategy selector via episodic memory becomes ineffective. On the other hand, training on the strategy selector with episodic memory shows its merit for the tasks with clear decomposition of strategy action spaces, such as a task in Fig. 5.(b).



(a) `6h_vs_8z`

(b) `3s5z_vs_3s6z`

Figure 5: Ablation studies on **super hard** SMAC maps

Qualitative Analysis Figure 6 describes the agents' behaviors with their designated strategies. At timestep 7, all Zealots were assigned to a strategy including attacking enemy Zealots, while all

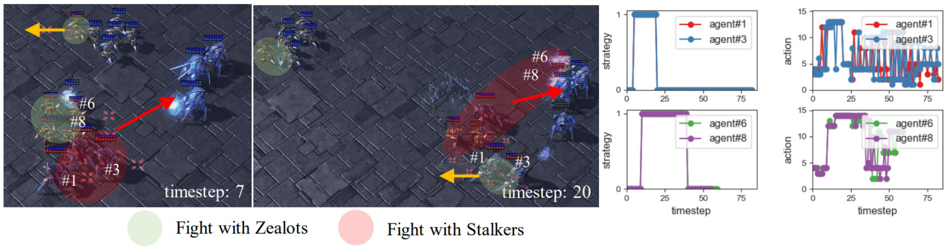


Figure 6: Visualization of assigned strategy on `3s5z_vs_3s6z` **super hard** SMAC map

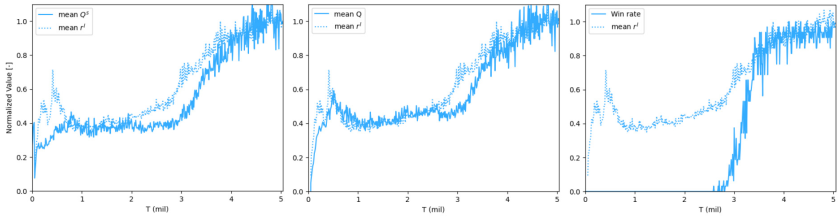


Figure 7: SIMA(MI): how does coordination incentive affect the performance on `3s5z_vs_3s6z` **super hard** SMAC map. All values except for winning rate are normalized with their final values for visibility.

Stalkers were ordered to fight against enemy Stalkers. At timestep 20, according to the strategy change, agent #1 and agent #3 moved toward enemy Zealots to prepare the engagement with them, as the ally Zealot attracting all enemy Zealots was about to die. As shown in right side of Fig. 6, agents with the same strategy trajectories show almost the same action trajectories, due to the coordination incentive. We measure the correlation coefficient between strategy trajectories $z_{t=0:T}$ and action trajectories $a_{t=0:T}$ of all agents from the 32 test samples on the `3s5z_vs_3s6z` super hard SMAC map with three different fully trained models of SIMA(Raw), SIMA(MI), and SIMA(ours). The correlation coefficient of each model is computed as $r_{\text{Raw}} = -0.15$, $r_{\text{MI}} = 0.61$, and $r_{\text{ours}} = 0.55$. Although all of them find their own winning strategies, models with a coordination incentive show much coordinated behaviors, according to their designated strategies.

Figure 7 illustrates the effect of coordination incentive towards the training performance. Until training timestep 2.5mil, no winning strategy were found. However, after a coordination incentive is given, some coordinated strategies are found, and the mean values of Q^s and Q start increasing. Finally, RL agents begin to find a winning strategy guided by the coordination incentive. This result suggests that the coordination incentive, r^I, proceeds learning under weak reward signal, consequently leading to find a better joint policy.

6 Conclusion

Hierarchical framework for MARL has drawn the attention because the real-world tasks has vast action space that should be regulated by roles and strategies. Given this hierarchical setting, SIMA provides an efficient sample utilization in training the strategy selector and the action policy since its structural characteristic yields instability and a decline in speed during training. Hence, we propose a new TD learning function to reduce the reward variance on strategy selector learning regardless of the exploration of the action policy. Finally, we introduce a new coordination incentive by regulating the agent's action policy to be coherent to the actions under the same strategy. Such an incentive helps agents to find coordinated behaviors with their designated strategies and the better joint policy as a result. We believe the proposed method can provide more insights about the coordination in multi-agent systems.

References

Marc Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Remi Munos. Unifying count-based exploration and intrinsic motivation. *Advances in neural information pro-*

cessing systems, 29, 2016.

Charles Blundell, Benigno Uria, Alexander Pritzel, Yazhe Li, Avraham Ruderman, Joel Z Leibo, Jack Rae, Daan Wierstra, and Demis Hassabis. Model-free episodic control. *arXiv preprint arXiv:1606.04460*, 2016.

Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation. *arXiv preprint arXiv:1810.12894*, 2018.

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.

Nikolay Dandanov, Hussein Al-Shatri, Anja Klein, and Vladimir Poulkov. Dynamic self-optimization of the antenna tilt for best trade-off between coverage and capacity in mobile networks. *Wireless Personal Communications*, 92(1):251–278, 2017.

Marc-André Dittrich and Silas Fohlmeister. Cooperative multi-agent system for production control using reinforcement learning. *CIRP Annals*, 69(1):389–392, 2020.

Yali Du, Lei Han, Meng Fang, Ji Liu, Tianhong Dai, and Dacheng Tao. Liir: Learning individual intrinsic reward in multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 32, 2019.

Dibya Ghosh, Abhishek Gupta, and Sergey Levine. Learning actionable representations with goal-conditioned policies. *arXiv preprint arXiv:1811.07819*, 2018.

Jayesh K Gupta, Maxim Egorov, and Mykel Kochenderfer. Cooperative multi-agent control using deep reinforcement learning. In *International conference on autonomous agents and multiagent systems*, pp. 66–83. Springer, 2017.

Matthew Hausknecht and Peter Stone. Deep recurrent q-learning for partially observable mdps. In *2015 aaai fall symposium series*, 2015.

Rein Houthooft, Xi Chen, Yan Duan, John Schulman, Filip De Turck, and Pieter Abbeel. Vime: Variational information maximizing exploration. *Advances in neural information processing systems*, 29, 2016.

Natasha Jaques, Angeliki Lazaridou, Edward Hughes, Caglar Gulcehre, Pedro Ortega, DJ Strouse, Joel Z Leibo, and Nando De Freitas. Social influence as intrinsic motivation for multi-agent deep reinforcement learning. In *International conference on machine learning*, pp. 3040–3049. PMLR, 2019.

Junchen Jin and Xiaoliang Ma. Hierarchical multi-agent control of traffic lights based on collective learning. *Engineering applications of artificial intelligence*, 68:236–248, 2018.

Hyoungseok Kim, Jaekyeom Kim, Yeonwoo Jeong, Sergey Levine, and Hyun Oh Song. Emi: Exploration with mutual information. *arXiv preprint arXiv:1810.01176*, 2018.

Youngwoon Lee, Jingyun Yang, and Joseph J Lim. Learning to coordinate manipulation skills via skill behavior diversification. In *International conference on learning representations*, 2019.

Zichuan Lin, Tianqi Zhao, Guangwen Yang, and Lintao Zhang. Episodic memory deep q-networks. *arXiv preprint arXiv:1805.07603*, 2018.

Iou-Jen Liu, Unnat Jain, Raymond A Yeh, and Alexander Schwing. Cooperative exploration for multi-agent deep reinforcement learning. In *International Conference on Machine Learning*, pp. 6826–6836. PMLR, 2021.

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

David Henry Mguni, Taher Jafferjee, Jianhong Wang, Nicolas Perez-Nieves, Oliver Slumbers, Feifei Tong, Yang Li, Jiangcheng Zhu, Yaodong Yang, and Jun Wang. Ligs: Learnable intrinsic-reward generation selection for multi-agent learning. *arXiv preprint arXiv:2112.02618*, 2021.

Shakir Mohamed and Danilo Jimenez Rezende. Variational information maximisation for intrinsically motivated reinforcement learning. *Advances in neural information processing systems*, 28, 2015.

Frans A Oliehoek and Christopher Amato. *A concise introduction to decentralized POMDPs*. Springer, 2016.

Frans A Oliehoek, Matthijs TJ Spaan, and Nikos Vlassis. Optimal and approximate q-value functions for decentralized pomdps. *Journal of Artificial Intelligence Research*, 32:289–353, 2008.

Georg Ostrovski, Marc G Bellemare, Aäron Oord, and Rémi Munos. Count-based exploration with neural density models. In *International conference on machine learning*, pp. 2721–2730. PMLR, 2017.

Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *International conference on machine learning*, pp. 2778–2787. PMLR, 2017.

Alexander Pritzel, Benigno Uria, Sriram Srinivasan, Adria Puigdomenech Badia, Oriol Vinyals, Demis Hassabis, Daan Wierstra, and Charles Blundell. Neural episodic control. In *International Conference on Machine Learning*, pp. 2827–2836. PMLR, 2017.

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*, pp. 4295–4304. PMLR, 2018.

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. The starcraft multi-agent challenge. *arXiv preprint arXiv:1902.04043*, 2019.

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*, pp. 5887–5896. PMLR, 2019.

Bradly C Stadie, Sergey Levine, and Pieter Abbeel. Incentivizing exploration in reinforcement learning with deep predictive models. *arXiv preprint arXiv:1507.00814*, 2015.

Changyin Sun, Wenzhang Liu, and Lu Dong. Reinforcement learning with task decomposition for cooperative multiagent systems. *IEEE transactions on neural networks and learning systems*, 32 (5):2054–2065, 2020.

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. *arXiv preprint arXiv:1706.05296*, 2017.

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

Haoran Tang, Rein Houthooft, Davis Foote, Adam Stooke, OpenAI Xi Chen, Yan Duan, John Schulman, Filip DeTurck, and Pieter Abbeel. # exploration: A study of count-based exploration for deep reinforcement learning. *Advances in neural information processing systems*, 30, 2017.

Hongyao Tang, Jianye Hao, Tangjie Lv, Yingfeng Chen, Zongzhang Zhang, Hangtian Jia, Chunxu Ren, Yan Zheng, Zhaopeng Meng, Changjie Fan, et al. Hierarchical deep multiagent reinforcement learning with temporal abstraction. *arXiv preprint arXiv:1809.09332*, 2018.

Alexander Vezhnevets, Yuhuai Wu, Maria Eckstein, Rémi Leblond, and Joel Z Leibo. Options as responses: Grounding behavioural hierarchies in multi-agent reinforcement learning. In *International Conference on Machine Learning*, pp. 9733–9742. PMLR, 2020.

Binyu Wang, Zhe Liu, Qingbiao Li, and Amanda Prorok. Mobile robot path planning in dynamic environments through globally guided reinforcement learning. *IEEE Robotics and Automation Letters*, 5(4):6932–6939, 2020a.

Jianhao Wang, Zhizhou Ren, Terry Liu, Yang Yu, and Chongjie Zhang. Qplex: Duplex dueling multi-agent q-learning. *arXiv preprint arXiv:2008.01062*, 2020b.

Tonghan Wang, Jianhao Wang, Yi Wu, and Chongjie Zhang. Influence-based multi-agent exploration. *arXiv preprint arXiv:1910.05512*, 2019.

Tonghan Wang, Tarun Gupta, Anuj Mahajan, Bei Peng, Shimon Whiteson, and Chongjie Zhang. Rode: Learning roles to decompose multi-agent tasks. *In Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

Marco A Wiering et al. Multi-agent reinforcement learning for traffic light control. In *Machine Learning: Proceedings of the Seventeenth International Conference (ICML'2000)*, pp. 1151–1158, 2000.

Jiachen Yang, Igor Borovikov, and Hongyuan Zha. Hierarchical cooperative multi-agent reinforcement learning with skill discovery. *arXiv preprint arXiv:1912.03558*, 2019.

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.

A Intrinsic reward for coordination

As $\tau_{0:t}$ is not dependent on the future strategy $z_{t+1:T}$, $p(\tau_T|z_{0:T})$ can be further expanded as

$$\begin{aligned} p(\tau_T|z_{0:T}) &= p(o_0|z_0)\prod\nolimits_{t=0}^{T-1} p(a_t|\tau_t, z_t)p(o_{t+1}|\tau_t, a_t, z_t) \\ &= p(o_0)\prod\nolimits_{t=0}^{T-1} p(a_t|\tau_t, z_t)p(o_{t+1}|\tau_t, a_t). \end{aligned} \tag{16}$$

The second equality comes from the fact that for a given action a_t, a strategy z_t at timestep t does not affect the probability of the observation transition since the transition function of the environment depends on a_t not on z_t itself. Then we can write Eq.9 as follows:

$$\begin{aligned} I^{\pi^s/\boldsymbol{\pi}}(\tau_T; z_{0:T}) &= E_{z_{0:T}\sim\pi^s,\tau_T\sim\boldsymbol{\pi}}[\log \frac{p(o_0)\prod_{t=0}^{T-1} p(a_t|\tau_t, z_t)p(o_{t+1}|\tau_t, a_t)}{p(o_0)\prod_{t=0}^{T-1} p(a_t|\tau_t)p(o_{t+1}|\tau_t, a_t)}] \\ &= E_{z_{0:T}\sim\pi^s,\tau_T\sim\boldsymbol{\pi}}[\log \frac{\prod_{t=0}^{T-1} p(a_t|\tau_t, z_t)}{\prod_{t=0}^{T-1} p(a_t|\tau_t)}] \\ &= E_{z_{0:T}\sim\pi^s,\tau_T\sim\boldsymbol{\pi}}[\sum_{t=0}^{T-1} \log \frac{p(a_t|\tau_t, z_t)}{p(a_t|\tau_t)}] = \sum_{t=0}^{T-1} E_{z_t\sim\pi^s,\tau_t\sim\boldsymbol{\pi}}\left[\log \frac{p(a_t|\tau_t, z_t)}{p(a_t|\tau_t)}\right]. \end{aligned} \tag{17}$$

B Mathematical proof

In this section, we provide the omitted proof of Proposition 1. Let us begin with the definition of some terminologies for the proof.

- $Q^s_{tot}(s_t, z^*_{jt,t})$: the optimal joint strategy-value of a given s_t.
- $\tilde{Q}^s_{tot}(s_t, z^*_{jt,t})$: the closest strategy-value compared to $Q^s_{tot}(s_t, z^*_{jt,t})$ within **the entire training experience**.
- $\hat{Q}^s_{tot}(s_t, z^*_{jt,t})$: the estimated strategy-value via the strategy selector.
- $\bar{Q}^s_{tot}(s_t, z^*_{jt,t})$: the estimated strategy-value via the target network of a strategy selector.

Then, the optimal gradient of the loss is defined as

$$\nabla_{\theta_s}\tilde{L}_s = \nabla_{\theta_s}\left(\tilde{Q}^s_{tot}(s_t, z^*_{jt,t}) - \hat{Q}^s_{tot}(s_t, z^*_{jt,t})\right)^2. \tag{18}$$

In this proof, we denote a state-strategy value as $Q^s_{tot}(s_t, z_{jt,t})$ instead of $Q^s_{tot}(\tau_t, z_{jt,t})$ for simplicity, and VDN mixer is assumed for a mixer to generate Q^s_{tot} from individual Q^s_i, such that $Q^s_{tot} = \sum_{i=1}^n Q^s_i$.

Proof. Let us assume that trajectory samples of c strategy interval timesteps, $\tau_{t:t+c-1} = (s_t, a_{jt,t}, z_{jt,t}, s_{t+1}, \cdots, s_{t+c-1}, a_{jt,t+c-1}, z_{jt,t+c-1}, s_{t+c})$ are attained from replay buffer $\mathcal{D}$ and the corresponding samples of $(\hat{s}_t, z^*_{jt,t}, H(f_\phi(s_t)), \hat{s}_{t+c}, z^*_{jt,t+c}, H(f_\phi(s_{t+c}))$ are found in episodic buffer $\mathcal{D}_E$. $\tilde{Q}^s_{tot}(s_t, z^*_{jt,t})$ can also be computed as

$$\tilde{Q}^s_{tot}(s_t, z^*_{jt,t}) = r^{s*} + \gamma \tilde{Q}^s_{tot}(s_{t+c}, z^*_{jt,t+c}). \tag{19}$$

We also assume that there exists sub-optimal joint actions within the trajectory samples, i.e., $^\exists a_{jt,t'} \in \tau_{t:t+c-1}$ such that $a_{jt,t'} \neq a^*_{jt,t'}$, due to the exploration of action policy, where $a^*_{jt,t'}$ is the optimal joint actions such that $\hat{Q}_{tot}(s_t, a^*_{jt,t}) \geq \hat{Q}_{tot}(s_t, a^{'}_{jt,t})$ for $^\forall a^{'}_{jt,t} \in A_{jt}$. The summed reward within the strategy interval is defined as $r^s_t \triangleq \sum_{t'=0}^{c-1} r(s_{t+t'}, a_{jt,t+t'})$.

(1) conventional TD-loss of Q-learning
Here, we denote the gradient of loss derived from the conventional TD-target of Q-learning as $\nabla_{\theta_s} L^c_s$ and corresponding TD-target as y^c_{tot}. The conventional TD-target of Q-learning is derived as

$$y^c_{tot} = r^s_t + \gamma \max_{z^{'}_{jt}} \bar{Q}^s_{tot}(s_{t+c}, z^{'}_{jt}). \tag{20}$$

(i) when $\max_{z^{'}_{jt}} \bar{Q}^s_{tot}(s_{t+c}, z^{'}_{jt}) < H(f_\phi(\hat{s}_{t+c}))$

This would be the most case of the samples from D. When the target network of a strategy selector is smaller than $H(f_\phi(\hat{s}_{t+c}))$, then $\max_{z^{'}_{jt}} \bar{Q}^s_{tot}(s_{t+c}, z^{'}_{jt}) \neq \tilde{Q}^s_{tot}(s_{t+c}, z^*_{jt,t+c})$ by the definition since $H(f_\phi(\hat{s}_{t+c})) = \tilde{Q}^s_{tot}(s_{t+c}, z^*_{jt,t+c})$. This yields $\nabla_{\theta_s} L^c_s \neq \nabla_{\theta_s} \tilde{L}_s$.

(ii) when $\max_{z^{'}_{jt}} \bar{Q}^s_{tot}(s_{t+c}, z^{'}_{jt}) = H(f_\phi(\hat{s}_{t+c}))$

In this case, $\max_{z^{'}_{jt}} \bar{Q}^s_{tot}(s_{t+c}, z^{'}_{jt})$ becomes $\tilde{Q}^s_{tot}(s_{t+c}, z^*_{jt,t+c})$. However, due to the sub-optimal joint action $a^{'}_{jt,t} \in \tau_{t:t+c-1}$, r^s is less than $(r^s)^*$, which yields $\nabla_{\theta_s} L^c_s \neq \nabla_{\theta_s} \tilde{L}_s$ as a result.

(2) the proposed TD-loss of Q-learning
The proposed TD-target is defined as

$$y_{tot} = r^{s*} + \gamma \max\left(\max_{z^{'}_{jt}} \bar{Q}^s_{tot}(s_{t+c}, z^{'}_{jt}), H(f_\phi(\hat{s}_{t+c}))\right). \tag{21}$$

When $z_{jt,t} = z^*_{jt,t}$, $\max_{z^{'}_{jt}} \bar{Q}^s_{tot}(s_{t+c}, z^{'}_{jt}) \leq H(f_\phi(\hat{s}_{t+c}))$ is always satisfied by the update rule of $H(f_\phi(\hat{s}_{t+c}))$. In addition, $\sum_i^n \hat{r}^s_i$ becomes $r^s(\hat{s}_t, z^*_{jt,t})$ by the definition of Eq.5. As a result, $\sum_i^n \hat{r}^s_i = r^{s*}$, since $r^{s*} = H(f_\phi(\hat{s}_t)) - \gamma H(f_\phi(\hat{s}_{t+c}))$. Then, the the proposed TD-target becomes

$$\begin{aligned} y_{tot} &= \sum_i^n \hat{r}^s_i + \gamma \max\left(\max_{z^{'}_{jt}} \bar{Q}^s_{tot}(s_{t+c}, z^{'}_{jt}), H(f_\phi(\hat{s}_{t+c}))\right) \\ &= r^{s*} + \gamma H(f_\phi(\hat{s}_{t+c})) = r^{s*} + \gamma \tilde{Q}_{tot}(s_{t+c}, z^*_{jt,t+c}) \\ &= \tilde{Q}_{tot}(s_t, z^*_{jt}), \end{aligned} \tag{22}$$

which yields $\nabla_{\theta_s} L_s = \nabla_{\theta_s} \tilde{L}_s$. □

When adopting episodic control for TD-target generation, it needs to be cautious whether the link between $\hat{s}_t$ and $\hat{s}_{t+c}$ is valid or not. For example, the value of $R_{t+c}(s_{t+c}, z^*_{jt,t+c}) = H(f_\phi(\hat{s}_{t+c}))$ is used for all $\hat{s}_{t+c}$ satisfying $||f_\phi(\hat{s}_{t+c}) - f_\phi(s_{t+c})||_2 < \delta$. Invalid link can refer the case of $H(f_\phi(\hat{s}_{t+c})) > H(f_\phi(\hat{s}_t))$, which implies minus rewards within $[t, t+c]$. Thus, we discard the an obvious-invalid link by checking $r^{s*} < 0$ in practice, when only a positive reward is obtainable from the environment.

In SIMA, a different strategy allows different action spaces on action policy. Thus, when $z_{jt,t} \neq z^*_{jt,t}$ and $^\exists \hat{s}_t, \hat{s}_{t+c}$, referring to the episodic memory can sample an invalid link. For example, $a_{jt,t}$ with a given $z_{jt,t}$ cannot generate a reward due to decomposed action spaces while $z^*_{jt,t}$ can, or a transition from s_t to $\hat{s}_{t+c}$ can be infeasible at all. Then, $\hat{Q}_{tot}(s_t, z_{jt,t}) \rightarrow r^{s*} + \gamma H(f_\phi(\hat{s}_{t+c}))$ is invalid and may cause learning instability.

Alternative approach of EMC (Zheng et al., 2021) where $r^s(s_t, z_{jt,t}) + \gamma H(f_\phi(\hat{s}_{t+c}))$ is used for a TD-target can also be vulnerable to this invalid link especially when it is adopted to hierarchical structure. Even though it refers to a valid link, its TD-target can result in a sub-optimal gradient $\nabla_{\theta_s} L_s$ when $z_{jt,t} = z^*_{jt,t}$ as shown in Proposition 1.

C Experiment Details

For action policy learning, we utilize a QPLEX mixing network with its default hyperparemeters from the original paper. Also, we set episodic memory capacity as 1M and episodic latent dimension as 4 by following the hyperparameters suggestion by EMC (Zheng et al., 2021). We adopt a usual ϵ-greedy setting with ϵ annealed linearly from its maximum value 1.0 to minimum value 0.05 over timestep T_ϵ presented in Table 1. For performance comparison with baseline methods, we use their own codes with fine-tuned algorithm configuration for hyperparameter settings if available. For hyperparameters related to a coordination incentive r^I, fixed common values of $\beta_I = 0.1$, $\beta_1 = 1.0$, and $\beta_2 = 1.0$ were used for all experiments.

For experiments on SMAC, we use the same version of starcraft.py for SMAC environment, which is based on the version used in RODE (Wang et al., 2021) adopting some modification for compatibility of QPLEX. All SMAC experiments were conducted on StraCraft II version 4.10.0 in Linux environment. Table 1 shows the hyperparameters regarding ϵ annealing time T_ϵ and a scale factor β_{exp} for a curiosity incentive r^{exp}.

Table 1: SIMA Hyperparameters.

category	map	T_ϵ	β_{exp}
easy/hard maps	all maps	70K	0.05
super hard maps	6h_vs_8z	500K	0.1
	MMM2	500K	0.001
	corridor	100K	0.2
	3s5z_vs_3s6z	500K	0.1

D Determining Strategy Action Spaces and Strategy Selection

Figure 8 illustrates a forward predictive model for training an action encoder, which makes one-hot action from a $|A|$ dimensional vector into a d-dimensional vector, $f_{a,e}(\cdot;\theta_e) : \mathbb{R}^{|A|} \rightarrow \mathbb{R}^d$. This predictive model is to learn an action representation h_a for a given action a that minimizes the prediction error of observation transition o' and a reward r. The loss function of the predictive model is defined as follows:

$$L_e(\theta_e, \xi_e) = \mathbb{E}_D[\sum_i^n ||p_o(h_{a_i}, o_i, \boldsymbol{a}_{-i}) - o'_i||_2 + \lambda_e \sum_i^n (p_r(h_{a_i}, o_i, \boldsymbol{a}_{-i}) - r)^2] \tag{23}$$

where λ_e is a scaling factor; p_o and p_r are prediction model parameterized with ξ_e for $\hat{o}_i^{'}$ and $\hat{r}$ as presented in Fig. 8. At the beginning, the action spaces of all strategies are initialized with the full action space. After training the predictive model for predetermined t_e timesteps, strategy action spaces are updated based on the similarity in action representation generated by $f_{a,e}(\cdot;\theta_e)$. There can be a variety of choices in clustering methods, we adopt a simple but effective K-means algorithm based on Euclidean distances between strategy action representations as in the original paper (Wang et al., 2021).

Since the representation vector of actions become similar if actions show similar reward and observation in the next timestep, a cluster of representation vectors will mean a coherent set of actions conveying a similar semantic meaning in observations and reward. In RODE, the strategy representation h_z of strategy z is defined as its mean value of strategy-action embedding $h_{a.}$ as follows:

$$h_z = \frac{1}{|A_z|} \sum_{a' \in A_z} h_{a'} \tag{24}$$

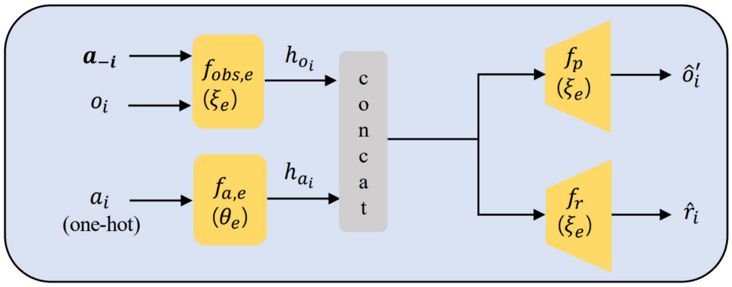


Figure 8: Strategy action representation learning

To select the strategy z for an agent i, the agent's action-observation history τ_i is encoded into h_{τ_i} by the GRU unit with parameters θ_τ, which is shared across all τs. Afterwards, a strategy selection is made by 1) turning τ_i to an embedding vector through $\bar{h}_{\tau_i} = f_{emb}(h_{\tau_i}; \theta_{emb})$, and 2) finding the most linearly aligned strategy representation h_z with the action-observation representation of $\bar{h}_{\tau_i}$ as the below.

$$Q^s(\tau_i, z) = f_{emb}(h_{\tau_i}; \theta_{emb})^T h_z = \bar{h}_{\tau_i}^T h_z \tag{25}$$

Figure 9 illustrates how strategy selector works with strategy representations h_z.

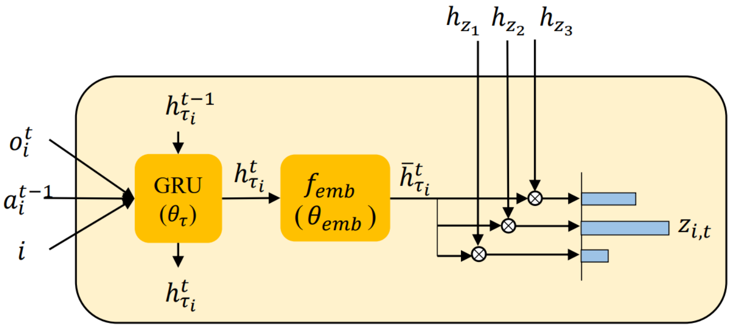


Figure 9: Strategy selector, Q^s

E Balancing Exploitation and Exploration

The learning method leveraging episodic control increases a sample efficiency (Lin et al., 2018; Zheng et al., 2021), as it references a TD target with the best return in the episodic memory. Therefore, this TD learning with best return encourages a greedy strategy selection in accordance with good memories that were found in an early learning phase. Whereas this might expedites learning, this can also result in a premature convergence to local optima, leading to a failure of finding a optimal policy in hard tasks.

Especially in a hierarchy structure, the premature convergence of a strategy selector Q^s on local optima can severely degrade learning performance, as it discards chance to find a better action $a \sim \pi_{z'}$ that can be derived from a currently sub-optimal strategy z'. To resolve this issue, we adopt a curiosity-driven exploration method, which uses prediction error as a curiosity; and SIMA provides incentive to the state with bigger prediction errors (Burda et al., 2018; Pathak et al., 2017).

As most MARL problems contain a large joint state space, we compute a prediction error based on the projected state space $x_t = f_\phi(s_t)$ that were used for episodic control. With a randomly initialized predictor network, $V_\phi(x_t)$, we can compute a prediction error as follows:

$$r^{\exp} = |H(\hat{x}_t) - V_\phi(x_t)| \tag{26}$$

With the intrinsic reward presented $r^{\exp}$ in Eq.26, the state of a new best return $H(\hat{x}_t)$ can still have an incentive to visit s_t even after the learning of $V_\phi(x_t)$. The final form of a loss function for the strategy selector L_{s} is developed from Eq.27 by including $r^{\exp}$ as follows:

$$L_{\mathrm{s}}(\theta_{\tau}, \theta_{emb}) = \mathbb{E}_{\tau,\boldsymbol{a},\boldsymbol{z},r,\tau' \in D}\left[\left(y_{tot} + \beta_{\exp} r^{\exp} - \sum_{i}^{N} Q_i^s(\tau_{i,t}, z_{i,t})\right)^2\right] \tag{27}$$

where $\beta_{\exp}$ is a scale factor for the intrinsic reward. With this intrinsic reward, we can achieve the balance between exploration and exploitation for strategy selector learning.