Value Functions Factorization With Latent State Information Sharing in Decentralized Multi-Agent Policy Gradients

Hanhan Zhou, Tian Lan, *Senior Member, IEEE*, and Vaneet Aggarwal, *Senior Member, IEEE*

***Abstract*—The use of centralized training and decentralized execution for value function factorization demonstrates the potential for addressing cooperative multi-agent reinforcement tasks. QMIX, one of the methods in this field, has emerged as the leading approach and showed superior performance on the StarCraft II micromanagement benchmark. Nonetheless, its monotonic mixing method of combining per-agent estimates in QMIX has limitations in representing joint action Q-values and may not provide enough global state information for accurately estimating single-agent value function, which can lead to suboptimal results. To this end, we present LSF-SAC, a novel framework that features a variational inference-based information-sharing mechanism as extra state information to assist individual agents in the value function factorization. We demonstrate that such latent individual state information sharing can significantly expand the power of value function factorization, while fully decentralized execution can still be maintained in LSF-SAC through a soft-actor-critic design. We evaluate LSF-SAC on the StarCraft II micromanagement challenge and demonstrate that it outperforms several state-of-the-art methods in challenging collaborative tasks. We further set extensive ablation studies for locating the key factors accounting for its performance improvements. We believe that this new insight can lead to new local value estimation methods and variational deep learning algorithms.**

***Index Terms*—Machine learning, reinforcement learning, multi-agent systems.**

I. Introduction

REINFORCEMENT learning has been shown to match or surpass human performance in multiple domains, including various Atari games [3], [25], [26], Go [22], and StarCraft II [45]. Many real-world problems, like autonomous vehicles coordination [17] and network packet delivery [50] often involve multiple agents' decision making, which can be modeled as multi-agent reinforcement learning (MARL). Even though multi-agent cooperative problems could be solved by single-agent algorithms, joint state, and action space imply limited scalability [29], [41]. Further, partial observability and communication constraints give rise to additional challenges to MARL problems. One approach to deal with such issues is the paradigm of centralized training and decentralized execution (CTDE) [21]. The approaches for CTDE mainly include value function decomposition [34], [39] and multi-agent policy gradient [4].

Value decomposition based approaches like QMIX [34] represent the joint action values using a monotonic mixing function of per-agent estimates. The algorithms recorded the best performance on many StarCraft II micromanagement challenge maps [24]. Further, it is demonstrated [31] that multi-agent policy gradient is substantially outperformed by QMIX on both multi-agent particle world environment (MPE) [27] and StarCraft multi-agent challenge (SMAC) [35]. Despite recent attempts for combining policy gradient methods and value decomposition, e.g., VDAC [38], and mSAC [32], the achieved improvements over QMIX are limited. One of the fundamental challenges is that the restricted function class permitted by QMIX limits the joint action Q-values it can represent, leading to suboptimal value approximations and inefficient explorations [24]. A number of proposals have been made to refine the value function factorization of QMIX, e.g., QTRAN [37] and weighted QMIX [33]. However, solving tasks that require significant coordination remains a key challenge.

To this end, we propose *LSF-SAC* a Latent State information sharing assisted value function factorization under multi-agent Soft-Actor-Critic paradigm. In particular, we introduce a novel peer-assisted information-sharing mechanism to enable effective value function factorization by sharing the latent individual states, which can be considered extra state information for more accurate individual Q-value estimation by each agent. While global information sharing or communications in MARL - e.g., TarMAC [2] - typically prevents fully distributed decision-making, we show that by leveraging the design of soft-actor-critic, LSF-SAC is able to retain fully decentralized execution while enjoying the benefits of latent individual states sharing. It also incorporates the entropy measure of the policy into the reward to encourage exploration.

Manuscript received 4 January 2022; revised 11 September 2022, 4 December 2022, 21 February 2023, and 3 April 2023; accepted 15 May 2023. Date of publication 17 July 2023; date of current version 25 September 2023. This research was supported by CISCO and Meta. *(Corresponding author: Hanhan Zhou.)*

Hanhan Zhou and Tian Lan are with the Department of Electrical and Computer Engineering, George Washington University, Washington, DC 20052 USA (e-mail: hanhan@gwu.edu; tlan@gwu.edu).

Vaneet Aggarwal is with the School of Industrial Engineering and the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907 USA, and also with the CS Department, KAUST, Thuwal 23955, Saudi Arabia (e-mail: vaneet@purdue.edu).

A demo video and code of implementation can be found at https://sites.google.com/view/sacmm.

This article has supplementary downloadable material available at https://doi.org/10.1109/TETCI.2023.3293193, provided by the authors.

Digital Object Identifier 10.1109/TETCI.2023.3293193

The key insight of LSF-SAC is that existing approaches of value function factorization mainly use the joint state information only in the mixing network, which yet is restricted by the function class it can represent. We show an accurate independent value function estimation requires not only the state information of one specific agent but also a proper representation of all individual state information. We propose a way to extract and utilize the extra state information for individual, per-agent value function estimation through a variational inference method, serving as latent individual state information, since it's impossible and unnecessary to feed the whole state information to individual value function estimations. It is shown to significantly improve the power of value function factorization. Since we utilize such latent state information sharing only in centralized critic, the CTDE assumptions are preserved without affecting fully decentralized decision making, unlike previous work introducing global communications [47]. Further, we note that combining actor-critic framework with value decomposition in LSF-SAC offers a way to decouple the decision-making of individual agents (through separate policy networks) from value function networks, while also allowing the maximization of entropy to enhance its stability and exploration.

Our key contributions are summarized as follows:

- Our novel approach, LSF-SAC, introduces a unique method for value function factorization that incorporates additional individual latent state information to enhance per-agent value function estimation. Our study demonstrates that the inclusion of latent state information can substantially enhance the efficacy of monotonic factorization operators, representing the first framework for value function factorization to leverage this technique.
- The soft-actor-critic design in LSF-SAC enables the segregation of policy networks and value function networks for individual agents, allowing a completely decentralized execution while still maintaining the advantages of peer-assisted value function factorization. Additionally, LSF-SAC promotes an entropy maximization approach for multi-agent reinforcement learning, resulting in a more effective exploration.
- Our results showcase the efficacy of LSF-SAC and highlight its superior performance compared to several state-of-the-art baselines on the StarCraft II micromanagement challenge, by achieving better outcomes and faster convergence.

II. Background

A. Value Function Decomposition

Value function decomposition methods [34], [37], [39], [48] learn a joint Q functions $Q^{\text{tot}}(\tau, \mathbf{a})$ as a function of combined individual Q functions, conditioning individual local observation history,then these local Q values are combined with a learnable mixing neural network to produce joint Q values [36].

$$Q^{\text{tot}}(\tau, \mathbf{a}) = q^{\text{mix}}\left(\boldsymbol{s}, \left[q^i\left(\tau^i, a^i\right)\right]\right) \tag{1}$$

Under the principle of guaranteed consistency between global optimal joint actions and local optimal actions, a global argmax performed on Q^{tot} yields the same result as a set of individual argmax operations performed on each local q^i, also known as Individual Global Maximum (IGM):

$$\arg\max_{\boldsymbol{u}} Q^{\text{tot}} = \left(\arg\max_{u_1} Q_1, \ldots, \arg\max_{u_N} Q_N\right) \tag{2}$$

VDN [39] takes the joint value function as a summation of local action-value:

$$Q^{\text{tot}}(\boldsymbol{\tau}, \boldsymbol{u}) = \sum_{i=1}^{N} Q_i(\tau_i, u_i) \tag{3}$$

while QMIX proposed a more general case of VDN by approximating a broader class of monotonic functions to represent joint action-value functions rather than a summation of the local action values.

$$\frac{\partial Q^{\text{tot}}(\boldsymbol{\tau}, \boldsymbol{u})}{\partial Q_i\left(\tau_i, u_i\right)} > 0, \forall i \in \mathcal{N}. \tag{4}$$

QPLEX [46] provides IGM consistency by taking advantage of the duplex dueling architecture,

$$Q^{\text{tot}}(\boldsymbol{\tau}, \boldsymbol{u}) = \sum_{i=1}^{N} Q_i\left(\boldsymbol{\tau}, u_i\right) + \sum_{i=1}^{N} \left(\lambda_i(\boldsymbol{\tau}, \boldsymbol{u}) - 1\right) A_i\left(\boldsymbol{\tau}, u_i\right) \tag{5}$$

where

$$\begin{aligned} A_i\left(\boldsymbol{\tau}, u_i\right) &= w_i(\boldsymbol{\tau})\left[Q_i\left(\tau_i, u_i\right) - V_i\left(\tau_i\right)\right], V_i\left(\tau_i\right) \\ &= \max_{u_i} Q_i\left(\tau_i, u_i\right), \end{aligned} \tag{6}$$

$w_i(\boldsymbol{\tau})$ is a positive weight, yet its operator still limits it to only discrete action space [51].

B. Maximum Entropy Deep Reinforcement Learning

In a maximum entropy reinforcement learning framework, also known as soft-actor-critic [10], the objective is to maximize not only the cumulative expected total reward, but also the expected entropy of the policy:

$$J(\pi) = \sum_{t=0}^{T} \mathbf{E}_{\left(\mathbf{s}_t, \mathbf{a}_t\right) \sim \rho_\pi}\left[r\left(\mathbf{s}_t, \mathbf{a}_t\right) + \alpha \mathcal{H}\left(\pi\left(\cdot|\mathbf{s}_t\right)\right)\right] \tag{7}$$

where $\rho_\pi(\mathbf{s}_t, \mathbf{a}_t)$ denotes the state-action marginal distribution of the trajectory induced by the policy $\pi(\mathbf{a}_t|\mathbf{s}_t)$. Soft actor-critic utilized actor-critic architecture with independent policy and value networks and an off-policy paradigm for efficient data collection and entropy maximization for effective exploration. It is considered a state-of-the-art baseline for many RL problems with continuous actions due to its stability and capability.

C. Multi-Agent Policy Gradient Method

Multi-agent policy gradient (MAPG) methods are extensions to policy gradient algorithms, with policy $\pi_{\theta_a}(u^a|o^a)$. Compared with single-agent policy gradient methods, MAPG usually faces the issues of high variance gradient estimates [41] and

credit assignment [5]. A general multi-agent policy gradient can be written as:

$$\nabla_\theta J = \mathbb{E}_\pi \left[\sum_u \nabla_\theta \log \pi_\theta \left(u^a | o^a\right) Q_\pi(s, \mathbf{u}) \right]$$

Current literature on multi-agent policy gradients often leverages centralized training with a decentralized execution (CTDE) approach. This involves using a central critic to obtain additional state information s, and helps avoid the high variance associated with vanilla multi-agent policy gradients. For instance, [41] utilize a central critic to estimate $Q(s, (a_1, \ldots, a_n))$ and optimize parameters in actors by following a multi-agent DDPG gradient, which is derived from:

$$\nabla_{\theta_\alpha} J = \mathbb{E}_\pi \left[\nabla_{\theta_a} \pi \left(u^a | o^a\right) \nabla_u \cdot Q_{u^a}(s, \mathbf{u})|_{u^\alpha = \pi(o^\alpha)} \right]$$

COMA [4] proposes to apply the following counterfactual policy gradients to solve the credit assignment issue by as: where $A^a(s, \mathbf{u}) = \sum_{u^-} \pi_\theta(u^a|\tau^a) Q_\pi^a(s, (\mathbf{u}^{-a}, u^a))$ is the counterfactual advantage for agent a.

D. Variational Autoencoders

For variables $X \in \mathcal{X}$ which are generated from unknown random variable z based on a generative distribution $p_u(x|z)$ with unknown parameter $\boldsymbol{u}$ and a prior distribution on the latent variables, of which we assume is a Gaussian with 0 mean and unit variance $p(z) = \mathcal{N}(\boldsymbol{z}; \mathbf{0}, \boldsymbol{I})$. To approximate the true posterior $p(z|x)$ with a variational distribution $q_w(z|\vec{x}) = \mathcal{N}(\boldsymbol{z}; \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{w})$. [19], [20], [30] proposed Variational Autoencoders (VAE) to learn this distribution by using the Kullback-Leibler (KL) divergence from the approximate to the true posterior $D_{\mathrm{KL}}(q_w(z|x) \| p(z|x))$, the lower bound on the evidence $\log p(\boldsymbol{x})$ is derived as $\log p(\boldsymbol{x}) \geq \mathbb{E}_{\mathbf{z} \sim q_w(z|x)}[\log p_u(x|z)] - D_{KL}(q_w(z|x) \| p(z))$. [15] proposed β-VAE, where a parameter $\beta \geq 0$ is used to control the trade-off between the reconstruction loss and the KL-divergence.

E. Information Bottleneck Method

Information bottleneck method [43] is a technique in information theory which introduced as the principle of extracting the relevant information with random input variable $X \in \mathcal{X}$ and output random variable $Y \in \mathcal{Y}$, while finding the proper tradeoff between extraction accuracy and complexity. Given the joint distribution $p(x, y)$, their relevant information is defined as their mutual information $I(X; Y)$. This problem can also be seen as a rate-distortion problem [44] with non-fixed distortion measure conditioning the optimal map, defined as

$$d_{IB} = D_{\mathrm{KL}}(p(y|x) \| p(y|\hat{x}))$$

where D_{KL} is the Kullback-Leibler divergence. Then the expected IB distortion $E[d_{IB}(x, \hat{x})] = D_{IB} = I(X; Y|\hat{X})$, with the variational principle as

$$\mathcal{L}[p(\hat{x}|x)] = I(X; \hat{X}) - \beta I(X; Y|\hat{X})$$

where β is a positive Lagrange multiplier operates as a tradeoff parameter between accuracy and complexity. [1] further proposed a variational approximation to the information bottleneck using deep neural networks.

III. Related Works

Cooperative multi-agent decision-making confronts the situation of exponentially growing joint state and action spaces, which can pose significant challenges [40]. While various strategies such as independent Q-learning and mean field games have been explored in the literature, they often struggle to perform well on complex tasks or require agents with homogenous capabilities [38]. Recently, a centralized training and decentralized execution (CTDE) paradigm has been proposed to tackle these challenges for scalable decision-making [21]. Key approaches within the CTDE framework include value function decomposition and multi-agent policy gradient methods.

Compared to value-based methods, Policy Gradient methods are generally considered to have more stable convergence and can be extended more easily to continuous action problems [8]. One representative approach in the multi-agent Policy Gradient category is COMA [4], which employs a centralized critic module to estimate an individual agent's counterfactual advantage. However, as highlighted in recent studies [31], [54], value-based methods still outperform multi-agent policy-based methods like MADDPG [41] in the StarCraft multi-agent challenge (SMAC) [35].

To address the limitations of centralized critic modules, decomposed actor-critic methods that combine value function decomposition and policy gradient methods with decomposed critics have been introduced to guide policy gradients. VDAC [38] utilizes a structure similar to QMIX to estimate the joint state-value function, while DOP [48] uses a network similar to Qatten [49] for policy gradients with off-policy tree backup and on-policy TD. However, the authors of [48] note that decomposed critics are constrained by their limited expressive capability and may not converge to global optima, even if individual policies converge to local optima [51]. Although extensions of the monotonic mixing function, such as QTRAN [37], and weighted QMIX [33], have been explored, significant challenges remain when tackling tasks that require substantial coordination.

Another related topic is representational learning for reinforcement learning, and various methods have been proposed to learn effective state representations. For instance, [9] proposed a VAE-based forward model to learn state representations in the environment. [7] developed a technique to learn Gaussian embedding representations of different tasks during meta-testing. [18] introduced a recurrent VAE model that encodes observation and action history and learns a variational distribution of the task.

As also analyzed and suggested in MAVEN [24] and QTRAN [37], the representational constraints on the joint action-values introduced by the monotonic mixing network in QMIX [33] and similar methods will lead to provably poor

exploration and sub-optimal behavior policies. To solve this issue, one of the directions is to release the restriction of the joint action-value functions, e.g., QTRAN uses a linear summation over the utility functions and an additional value estimation, WQMIX [33] uses an unrestricted joint action-value function estimator as the weighted projection of a wider class of joint action-value functions; another direction is to promote a more committed exploration algorithm to recover the poor exploration introduced by the monotonic constraints, e.g., MAVEN combines value and policy-based methods with agents conditioning their behavior on a variable controlled policy for a temporally extended exploration. In this work, our proposed Decomposed Soft-actor-critic will promote the exploration through entropy maximization, while providing additional information from latent state information as assisted information for value function factorization.

Another topic related is communication-based MARL methods. Although the requirement of communication abilities might limit the actual use case of the proposed algorithm, with communications enabled, MARL agents will have a better understanding of the environment (or the other agents), and are therefore able to coordinate their behaviors and potentially better performances. Most works leverage local information to generate encoded messages. The messages may contain individual observations [11], [12], or intended actions (or plans) [13], [47] . A close paper to our work is NDQ [47], which also utilizes latent variables to represent the information as the communication messages during the decentralized agents' execution. Although we both consider information extraction as an information bottleneck problem, there are several key differences between our work and NDQ: (I) NDQ is a value-based method, while our work is a policy-based method under the soft-actor-critic framework. (II) NDQ requires communication between agents during decentralized execution, which limits its use cases, while we only utilize the latent extra state information during the central critics so that CTDE is maintained. (III) NDQ requires one-to-one communication during the execution stage, while in this work, we introduce a latent information-sharing mechanism that can be considered as an all-to-all message-sharing method. By enabling the latent information sharing mechanism in our work as a communication method, this work could potentially be transformed to a communication-based method, and many communication-based methods can be transformed into a framework where communication is only used for centralized training and restricted during execution, nevertheless, their performance and the actual use case may vary a lot.

The proposed LSF-SAC method leverages an actor-critic design with latent state information for value function factorization. We introduce a novel way to utilize the extra state information, as inspired from β-VAE [15], by using variational inference in a decomposed critic as latent state information for better individual value estimation. Despite information sharing, CTDE is still maintained due to the use of actor-critic structure. We also utilize the entropy and expected return maximization for better exploration through soft actor-critic with separate actor and critic networks.

IV. System Model

We approach the problem as a fully cooperative multi-agent environment with a decentralized partially observable Markov decision process (DEC-POMDP) [28]. The DEC-POMDP is defined as given by a tuple $G = \langle I, S, U, P, r, Z, O, n, \gamma \rangle$, where $I \equiv \{1, 2, \ldots, n\}$ is the finite set of agents. The state of the system is defined as a finite set of global states $s \in S$, from which each agent draws its own observation from the observation function $o_i \in O(s, i) : S \times A \rightarrow O$. At each timestamp t, each agent i chooses an action $u_i \in U$ where U is a set of actions available, forming a joint action selection $\boldsymbol{u}$. A shared reward is then given as $r = R(s, \boldsymbol{a}) : S \times \mathbf{U} \rightarrow \mathbb{R}$, and each agent transitions to a new state s' based on the transition probability function $P(s'|s, \mathbf{u}) : S \times U \rightarrow [0, 1]$. Each agent maintains its own action-observation history $\tau_i \in \mathrm{T} \equiv (O \times U)^*$. Then a joint action value function $Q^{\pi}_{tot}(\boldsymbol{\tau}, \boldsymbol{u}) = \mathbb{E}_{s_{0:\infty}, \boldsymbol{u}_{0:\infty}}[\sum_{t=0}^{\infty} \gamma^t r_t | s_0 = s, \boldsymbol{u}_0 = \boldsymbol{u}, \boldsymbol{\pi}]$ is proposed with policy π, and $\gamma \in [0, 1)$ is the discount factor. Notation in bold represents joint quantities across all agents, and quantities with superscript i are specific to agent i.

V. Proposed Approach

In this section, we first introduce the main structure of our proposed method, LSF-SAC, then we discuss the detailed implementation of the key designs, namely soft actor-critic framework for multi-agent reinforcement learning and value decomposition with latent information-sharing mechanism, and their corresponding optimizing strategies.

A. Framework Overview

In our learning framework (Fig. 1), each individual actor (Green part) outputs $\pi_\theta(a^i|\tau^i)$ only conditioned on its own local observation history. The centralized mixing network (Orange Part) approximates the joint action-value function from individual value functions (Blue part). A latent information-sharing mechanism (Purple part) is proposed to encode the extracted extra state information to assist individual agents in local action-value estimation. Function approximators (neural networks) are used for both actor and critic networks and optimized with stochastic gradient descent.

The centralized critic network consists of (i) a local Q-network for each agent, (ii) a mixing network that takes all individual action-values with their weights and biases generated by a separate hyper-network, and (iii) an extra state information encoder to generate latent state information for facilitating individual Q-value estimation. For each agent i, the local Q network represents its local Q value function $q_i(\tau_i, a_i, m_i)$ where m_i is the extra state information for agent i drawn from the global information sharing pool. More precisely, the information for agent i is generated from the messages of all other agents following a multivariate Gaussian distribution, denoted as $m_i =< m_1^{out} \cdots m_i^{out} \cdots m_n^{out} >$ with $m_i^{out} \sim \boldsymbol{N}(f_m(\tau_i; \theta_m), \boldsymbol{I}))$, where τ_i is the local observation history,

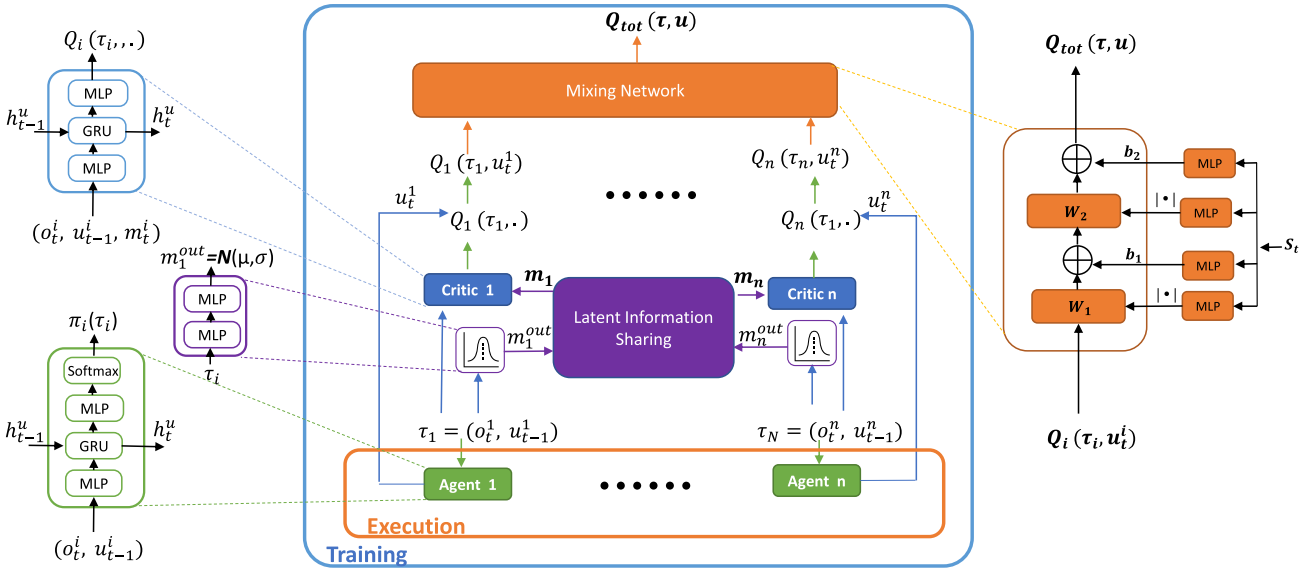


Fig. 1. Overview of LSF-SAC Approach. Best viewed in color.

θ_m is the parameters of encoder f_m and $\boldsymbol{I}$ is an identity matrix.

The mixing network is a feed-forward network, following the approach in QMIX, which mixes all local Q values to produce an estimate Q^{tot}. The weights and biases of the mixing network are generated by a hypernetwork that takes joint state information s. To enforce monotonicity, the weights generated from the hypernetworks are followed by an absolute function to create non-negative values. The decentralized actor-network is similar to the individual Q network, except it only conditions on its own observation and action history, and a softmax layer is added to the end of the network to convert logits into categorical distribution. The overall goal is to minimize:

$$\mathcal{L}(\boldsymbol{\theta}) = \mathcal{L}_{TD}(\boldsymbol{\theta_{TD}}) + \lambda_1 \mathcal{L}_m(\boldsymbol{\theta}_m) + \lambda_2 \mathcal{L}_\pi(\boldsymbol{\theta}_\pi) \quad (8)$$

where $\mathcal{L}_{TD}(\boldsymbol{\theta_{TD}})$ is the TD loss, of which we show it can also be used as the center critic loss, $\mathcal{L}_m(\boldsymbol{\theta}_m)$ is the message encoding loss, and $\mathcal{L}_\pi(\boldsymbol{\theta}_\pi)$ is the joint actor (policy) loss. λ_1 and λ_2 are the weighting terms. The details about latent state information generation and soft-actor-critic framework along with how to optimize them will be discussed in the following section.

B. Variational Approach Based Latent State Information

One of the key advantages of multi-agent policy gradients under the CTDE assumption is the effective utilization of extra state information. In our design, not only is the extra state information accessible to the mixing network but also to the individual agents' value networks (through information sharing). Due to the partial observability and uncertainty of the multi-agent environments, the individual value estimation conditioned on its own observation and action history can be volatile and unreliable. Intuitively, introducing extra information from other agents helps remove the ambiguity and uncertainty of current observation to enable effective individual value estimation.

However, it remains a crucial problem how to efficiently and effectively encode such extra state information. In most scenarios, even during the centralized training stage, it is impossible to directly feed the whole state information as input for individual value functions, as it consists of other agents' observation and unseen state information, without a carefully designed algorithm it is hard for a local agent to utilize them; at the same time, the input size of global state information is significantly larger than local observations, which would make the training longer to converge. We consider this as an information bottleneck problem [43], specifically, for agent i, we maximize the mutual information between other agents' encoded information and their actions while minimizing the mutual information between its own encoded information and action selection, so that only the necessary information is chosen and then efficiently encoded.

To encode additional state information for estimating individual values in an efficient and effective manner, we approach this problem as an information bottleneck problem [43], and the objective for each agent i can be written as:

$$J_m(\boldsymbol{\theta}_m) = \sum_{j=1}^{n} \left[I_{\theta_m}(A_j; M_i | \mathrm{T}_j, M_j) - \beta I_{\boldsymbol{\theta}_m}(M_i; T_i) \right] \quad (9)$$

where A_j is agent j's action selection, M_i is a random variable of m_i^{out}, T_j is a random variable of τ_j, and a parameter $\beta \geq 0$ is used to control the trade-off between the mutual information of its own and other agents. However, since the mutual information is intractable, this does not result in a model that can be learned. To overcome this challenge, we utilize variational approximation techniques, specifically the deep variational information bottleneck approach [1]. By parameterizing our model with a neural network, we can derive and optimize a variational lower bound for the first term of our objective function, as follows. Detailed derivations and proofs can be found in Appendix **A.1**.

Lemma 1: A lower bound of mutual information $I_{\theta_m}(A_j; M_i|\mathrm{T}_j, M_j)$ is

$$\mathbb{E}_{\mathbf{T}\sim\mathcal{D},M_j\sim f_m}[-\mathcal{H}[p(A_j|\mathbf{T}), q_\psi(A_j|\mathrm{T}_j, \boldsymbol{M})]]$$

where q_ψ is a variational Gaussian distribution with parameters ψ to approximate the unknown posterior $p(A_j|\mathrm{T}_j, M_j)$, $\mathbf{T} = \{T_1, T_2, \ldots, T_n\}$, $\mathbf{M} = \{M_1, M_2, \ldots, M_n\}$.

Proof: We provide a proof outline as follows.

$$I_{\theta_c}(A_j; M_i|T_j, M_j) = \int da_j d\tau_j dm_j p(a_j, \tau_j, m_j) \log \frac{p(a_j|\tau_j, m_j)}{p(a_j|\tau_j, m_j^{out})}$$

where $p(a_j|\tau_j, m_j)$ is fully defined by our decoder f_m and Markov Chain [23]. Note this is intractable in our case, let $q_\psi(a_j|\tau_j, m_j)$ be a variational approximation to $p(a_j|\tau_j, m_j)$. Since the KL-divergence is always positive,

hence

$$\begin{aligned} I_{\theta_c}(A_j; M_i|T_j, M_j) &\geq \int da_j d\tau_j dm_j p(a_j, \tau_j, m_j) \log \frac{q_\psi(a_j|\tau_j, m_j)}{p(a_j|\tau_j, m_j^{out})} \\ &= \mathbb{E}_{\mathbf{T}\sim\mathcal{D},M_j\sim f_m}[-\mathcal{H}[p(A_j|\mathbf{T}), q_\psi(A_j|\mathrm{T}_j, \boldsymbol{M})]] \\ &\quad + \mathcal{H}(A_j|T_j, M_j^{out}) \end{aligned}$$

Consider $\mathcal{H}(A_j|T_j, M_j^{out})$ is a positive term that is independent of our optimization procedure and can be ignored, then we have

$$I_{\theta_m}(A_j; M_i|\mathrm{T}_j, M_j) \geq \mathbb{E}_{\mathbf{T}\sim\mathcal{D},M_j\sim f_m}[-\mathcal{H}[p(A_j|\mathbf{T}), q_\psi(A_j|\mathrm{T}_j, \boldsymbol{M})]] \quad (10)$$

□

Similarly, by introducing another variational approximator q_ϕ, we have

$$\begin{aligned} I_{\boldsymbol{\theta}_m}(M_i; T_i) &= \mathbb{E}_{\mathrm{T}_i\sim D,M_j\sim f_m}[D_{\mathrm{KL}}(p(M_i|\mathrm{T}_i)\|p(M_i))] \\ &\leq \mathbb{E}_{\mathrm{T}_i\sim D,M_j\sim f_m}[D_{\mathrm{KL}}(p(M_i|\mathrm{T}_i)\|q_\phi(M_i))] \end{aligned} \quad (11)$$

where D_{KL} denotes the Kullback-Leibler divergence operator and $q_\phi(M_i)$ is a variational posterior estimator of $p(M_i)$ with parameters ϕ (see Appendix **A.1** for details). Then with the evidence lower bound derived above we optimize this bound for the message encoding objective which is to minimize

$$\begin{aligned} \mathcal{L}_m(\boldsymbol{\theta}_m) = \mathbb{E}_{\mathbf{T}\sim\mathcal{D},M_j\sim f_m}[&-\mathcal{H}[p(A_j|\mathbf{T}), q_\psi(A_j|\mathrm{T}_j, M_j)] \\ &+ \beta D_{\mathrm{KL}}(p(M_i|\mathrm{T}_i)\|q_\phi(M_i))]. \end{aligned} \quad (12)$$

C. Factorizing Multi Agent Maximum Entropy RL

In this section, we present one possible implementation of expanding soft actor-critic to the multi-agent domain with latent state information assisted value function decomposition, its objective extended to the multi-agent domain can be defined as

$$J(\pi) = \sum_t \mathbb{E}[r(\mathbf{s}_t, \mathbf{a}_t) + \alpha\mathcal{H}(\pi(\cdot|\mathbf{s}_t))] \quad (13)$$

Algorithm 1: LSF-SAC.

```
for k = 0 to train_steps_limits do
  Reset environment
  for t = 0 to max_episode do
    For each agent i, choose action a_i ~ π_i
    Execute joint action a, record reward r,
    save state-action history τ, next state s_{t+1}
  Store (τ, a, r, τ') in replay buffer D
  end for
  for t = 1 to T do
    Sample minibatch B from D
  Generate latent state information
  m_i^out ~ N(f_m(τ_i; θ_m), I)), for i = 0 to n
  Update critic network
  θ_D ← η∇̂L_TD(θ_D) w.r.t (9)
  Update policy network
  π ← η∇̂L(π) w.r.t (7)
  Update encoding network
    θ_m ← η∇̂L_m(θ_m) w.r.t (5)
  Update temperature parameter
  α ← η∇̂α w.r.t (8)
  if time_to_update_target_network then
    θ⁻ ← θ
  end if
  end for
end for
Return π
```

where the temperature α is the hyper-parameter to control the trade-off between maximizing the expected return and maximizing the entropy for better exploration.

Following the previous research on value decomposition, to maximize both the expected return and the entropy, we find the soft policy loss of LSF-SAC as:

$$\begin{aligned} \mathcal{L}_{LP}(\pi) &= \mathbb{E}_{\mathcal{D}}[\alpha\log\boldsymbol{\pi}(\boldsymbol{u}_t|\boldsymbol{\tau}_t) - Q_{tot}^{\pi}(\boldsymbol{s_t}, \boldsymbol{\tau_t}, \boldsymbol{u}_t, \boldsymbol{m}_t)] \\ &= -q^{\mathrm{mixing}}\big(\boldsymbol{s}_t, \mathbb{E}_{\pi^i}\big[q^i\left(\tau_t^i, u_t^i, m_t^i\right) \\ &\quad - \alpha\log\pi^i\left(u_t^i|\tau_t^i\right)\big]\big) \end{aligned} \quad (14)$$

where q^{mixing} is the value decomposition operator with $u_i \sim \pi_i(o_i)$, and $\mathcal{D}$ is the replay buffer used to sample training data (state-action history and reward, etc.).

Then, we can tune the temperature α as proposed in [10] by optimizing the following:

$$J(\alpha) = \mathbb{E}_{a_t\sim\pi_t}[-\alpha\log\pi_i(a_t|s_t) - \alpha\mathcal{H}_0] \quad (15)$$

Unlike VDAC which shares the same network for actor networks and local Q value estimations, we use a separate network for policy networks and train them independently from critic networks. Latent state information is used for individual critics for joint action value function factorization. We propose a latent state information-assisted soft value decomposition design as

$$Q_{tot}(\boldsymbol{\tau}, \boldsymbol{a}, \boldsymbol{m}; \boldsymbol{\theta}) = q^{\mathrm{mixing}}(\boldsymbol{s}_t, \mathbb{E}_{\pi^i}[q^i(\tau_t^i, a_t^i, m_t^i); \boldsymbol{\theta}])$$

TABLE I
PAYOFF MATRIX OF THE ONE-STEP MATRIX GAME, Q_1, Q_2 AND RECONSTRUCTED Q_{tot} OF SELECTED ALGORITHMS

u_1 \ u_2	A	B	C
A	**8.0**	-12.0	-12.0
B	-12.0	0.0	0.0
C	-12.0	0.0	0.0

(a) Payoff of matrix game

Q_1 \ Q_2	**4.2**(A)	2.3(B)	2.3(C)
3.8(A)	**8.0**	6.13	6.1
-2.1(B)	2.1	0.2	0.2
-2.3(C)	1.9	0.0	0.0

(b) QTRAN

Q_1 \ Q_2	**1.7**(A)	-11.5(B)	-12.7(C)
0.4(A)	**8.1**	-6.2	-6.0
-9.9(B)	-6.0	-5.9	-6.1
-9.5(C)	-5.9	-6.0	-6.0

(c) LSF-SAC

Q_1 \ Q_2	3.1(A)	**-2.3**(B)	-2.4(C)
-2.3(A)	-5.4	-4.6	-4.7
-1.2(B)	-4.4	-3.5	-3.6
-0.7(C)	-3.9	**-3.0**	-3.1

(d) VDN

Q_1 \ Q_2	-0.9(A)	0.0(B)	**0.0**(C)
-1.0(A)	-8.1	-8.1	-8.1
0.1(B)	-8.1	0.0	**0.0**
0.1(C)	-8.1	0.0	0.0

(e) QMIX

Q_1 \ Q_2	-2.5(A)	-1.3(B)	**0.0**(C)
-1.0(A)	-7.8	-6.0	-4.2
0.1(B)	-6.1	-4.4	**-2.6**
0.1(C)	-4.2	-2.4	-0.7

(f) DOP

Boldface denotes optimal/greedy actions from state-action value. The use of variational information can significantly improve the power of the function factorization operators.

We then use TD advantage with latent information sharing the design as the critic loss, i.e.,

$$\begin{aligned}\mathcal{L}_{TD}(\boldsymbol{\theta}) &= [r+\gamma \max_{\boldsymbol{a}^*} Q_{tot}\left(\boldsymbol{\tau}', \boldsymbol{a}', \boldsymbol{m}'; \boldsymbol{\theta}^-\right) - Q^{\pi}_{tot}(\boldsymbol{\tau}, \boldsymbol{a}, \boldsymbol{m}; \boldsymbol{\theta})]^2 \\ &= [r+\gamma \max_{\boldsymbol{a}^*} q^{\text{mixing}}(\boldsymbol{s}_t, \mathbb{E}_{\pi^i}[q^i(\tau^i_{t+1}, a^i_{t+1}, m^i_{t+1}); \boldsymbol{\theta}^-]) \\ &\quad - q^{\text{mixing}}(\boldsymbol{s}_t, \mathbb{E}_{\pi^i}[q^i(\tau^i_t, a^i_t, m^i_t); \boldsymbol{\theta}])]^2 \end{aligned} \tag{16}$$

where $a_i \sim \pi_i(o_i)$, $\boldsymbol{\theta}^-$ is the parameters of the target network that are periodically updated. Detailed derivations can be found in Appendix **A.2**.

VI. EXPERIMENTS

In this section, we first empirically study the improvements of power in value function factorization achieved by LSF-SAC through a non-monotonic matrix game. We compare the results with several existing value function factorization methods. Then in StarCraft II, we compare LSF-SAC with several state-of-the-art baselines. Finally, we perform several ablation studies to analyze the factors that contribute to the performance.

A. Single-State Matrix Game

Proposed in QTRAN [37], the non-monotonic matrix game, as illustrated in Table I(a), consists of two agents with three available actions and a shared reward. We show the value function factorization results of QTRAN, LSF-SAC, VDN, QMIX, and DOP [48].

Table I(b)–(f) shows the learning results of selected algorithms, QTRAN and LSF-SAC can learn a policy that each agent jointly takes the optimal action conditioning only on their local observations, meaning successful factorization. DOP falls into the sub-optimum caused by miscoordination penalties, similar to VDN and QMIX, which are limited by additivity and monotonicity constraints. Although QTRAN managed to address such limitations with more general value decomposition, as pointed out in later works [24] that it poses computationally intractable constraints that can lead to poor empirical performance on complex MARL domains. It is also worth noting that LSF-SAC can find the optimal joint action under the monotonic constraints by providing variational information, however, its joint action value estimation will still be restricted by such limitation; this indicates that the multi-agent entropy maximization design and the utilization of latent state information can significantly enhance the exploration policies and improve the power of the monotonic factorization operators in a mixing network like QMIX.

Besides the single-state matrix game example shown in Table I, we can also consider a multi-state problem with two agents, A and B. Let $(o_1^{(A)}, o_1^{(B)})$ and $(o_2^{(A)}, o_2^{(B)})$ be the two agents' observations in two different states s_1 and s_2. Providing latent information m_B conditioned on $o_1^{(B)}$ and $o_2^{(B)}$ will enable Agent A to better estimate its local utility $Q_A(o^{(A)}, m_B)$ in the two states s_1 and s_2. Thus, with the latent information m_A and m_B, the joint action-value function estimate with a mixing network f is given by $Q_{tot} = f(Q_A(o^{(A)}, m_B), Q_B(o^{(B)}, m_A))$, which is able to represent a larger class of functions than $Q_{tot} = f(Q_A(o^{(A)}), Q_B(o^{(B)}))$, for the goal of estimating $Q^*(o^{(A)}, o^{(B)})$.

B. Predator-Prey Environments

We first evaluate the performance of our baseline algorithms on a partially-observable multi-agent environment Predator-Prey environment as described in [53]. This environment involves 8 predators cooperating to catch 8 AI-controlled prey units on a 10×10 grid, with successful captures requiring at least two predators to surround and capture a prey unit simultaneously. Our aim is to test the algorithms' ability to handle relative over-generalization and monotonicity constraints. More details are provided in the Appendix on this environment. In this relatively easy testing environment, we observe satisfying final results compared to SOTA works. Although, at the beginning of the training, a larger shaded area indicates a more volatile training procedure, this could be due to the insufficient training of the information generation module at its earlier stage demonstrating the effect of the overhead from the information sharing mechanism.

C. Decentralised Starcraft II Micromanagement Benchmark

To further assess the effectiveness of our approach, we benchmark its performance against various state-of-the-art multi-agent reinforcement learning (MARL) methods on selected scenarios from the StarCraft Multi-Agent Challenge (SMAC) [35].

We then perform several ablation studies to analyze the factors that contribute to the performance. It is worth noting that the StarCraft Multi-Agent Challenges (SMAC) are affected by several code-level optimizations techniques, i.e., hyper-parameter tuning, as also found by [16], some works are relying on heavy hyper-parameters tuning to achieve results that they otherwise cannot. Consistent with previous work, we carry out the test with the same hyperparameters settings across all algorithms. More details about the algorithm implementation and settings can be found in Appendix **C**.

Fig. 2. Illustration of SMAC benchmark on map `5m_vs_6 m`, where the testing algorithm is to control the 5 marines on the left (marked green), combating with 6 marines controlled by the game built-in AI on the right (marked red).

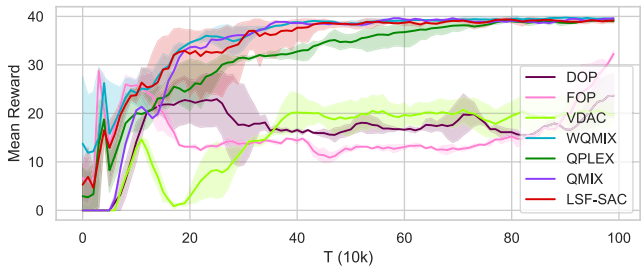


Fig. 3. Results on Predator-Prey Environments.

In the SMAC benchmark[1], each agent is responsible for controlling a unit that collaborates with other friendly units in combat against the game's built-in AI-controlled units. The combat can take on a symmetric form, where both parties have access to the same units, or it can be asymmetric. Our testing is conducted on 10 different maps that cover all difficulty levels, including 4 easy maps (`3 m, 3s5z, 8 m, 1c3s5z`), 3 hard maps (`3s_vs_5z, 5m_vs_6 m, 27m_vs_30 m,`), and 3 super-hard maps (`6h_vs_8z, corridor, MMM2, 27m_vs_30 m`). We selected these maps based on criteria such as the size of the action space (`27m-vs-30 m`), the need for advanced exploration strategies (corridor), and the requirement for a high level of coordination between agents (`6h_vs_8z`). The same default environment setting was used for all benchmark algorithms in our testing, and each baseline algorithm was trained using 4 random seeds and evaluated every 10,000 training steps with 32 testing episodes. Further details on the environment setup and hyperparameter settings can be found in Appendix A.3.3. We compare LSF-SAC with several state-of-the-art MARL algorithms as baselines. We choose two decomposed actor-critic methods: FOP [51] and DOP [52], one decomposed policy gradient method: VDAC [38], three decomposed value-based method: WQMIX [33], QPLEX [46] and QMIX [34],[2] and finally a communication-based value-based method: NDQ [47].

D. General Results

Following the practice of previous works [35], for every map result, we compare the winning rate and plot the median with the shaded area representing the highest and lowest range from testing results in Fig. 2. In general, we observe LSF-SAC achieves strong performance on all selected SMAC maps, notably it outperforms the state-of-the-art algorithms or achieves faster and more stable convergence at a higher win rate. Note that LSF-SAC performs exceptionally well on testing maps with challenging tasks that require more state information or substantial cooperation. Previous research has shown that there exists a performance gap between state-of-the-art (SOTA) value-based methods and policy gradient methods, particularly on maps that require the use of extensive exploration techniques.

In easy scenarios, almost all algorithms perform well. As the built-in AI would tend to attack the nearest enemy, by pulling back the friendly unit with a lower health value is a simple strategy to learn for winning. No significant performance gap was observed except for the training converging speed.

Within hard maps, LSF-SAC is able to train a usable policy that outperforms all baseline algorithms. On `27m_vs_30 m` and `MMM2`, LSF-SAC performs exceptionally in terms of the convergence speed and the final performance. On `corridor`, LSF-SAC and the selected two value-based methods are able to learn a model, with our method converging faster with slightly better performance, while policy-based methods suffer from this map as it requires more exploration to find the specific trick in winning this challenging scenario. On `5m_vs_6 m`, although within a similar performance range, LSF-SAC converges to a policy with lower variance and slightly better performance in the end. Finally, on `6h_vs_8z`, which is a super hard map that requires extensive exploration techniques, LSF-SAC achieves both faster convergence and better performance by a large margin as compared to the selected baselines.

It is also worth noting that the performance gap between value-based and policy-based methods still exists even for the state-of-the-art methods, while LSF-SAC as a policy-based method not only narrows such gap but also achieves remarkable performance.

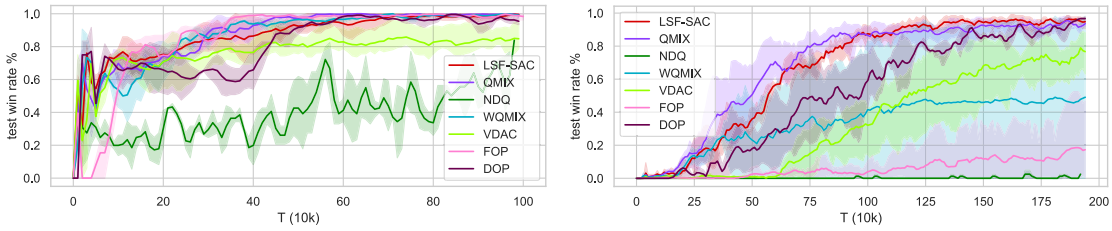
E. Ablation Study

In this section, we perform a comparison between LSF-SAC and several modified algorithms to understand the contribution of different modules in LSF-SAC. We choose one of the previously tested SMAC maps: `MMM2`. Each experiment is repeated with three independent runs with random seeds with their median results presented.

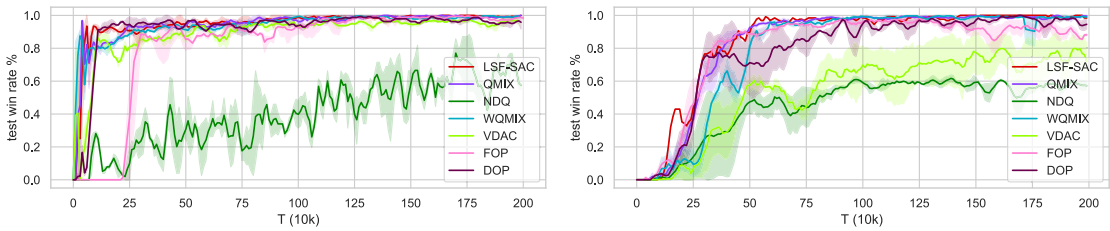
1) Ablation 1: First, we consider the setting of LSF-SAC without the extra state info encoding (Purple part in Fig. 1) as MASAC. This demonstrates how multi-agent soft-actor-critic works alone. It highlights the importance of latent state information by comparing the results of MASAC against the original LSF-SAC.

[1]In this article, all SMAC experiments are carried out utilizing the latest `SC2.4.10`, performance is always not comparable across versions. We implemented our algorithm based on an open-sourced codebase [16].

[2]In this section we refer WQMIX to ow-qmix as it shows a generally better performance than cw-qmix.

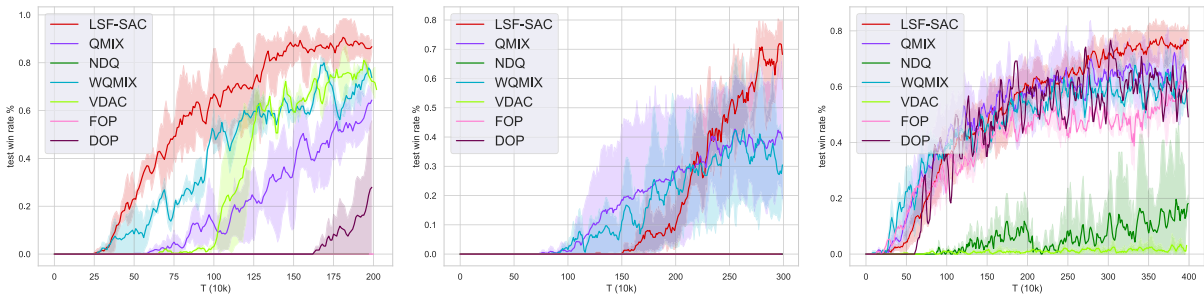


(a) 3m (easy) (b) 3s5z (easy)

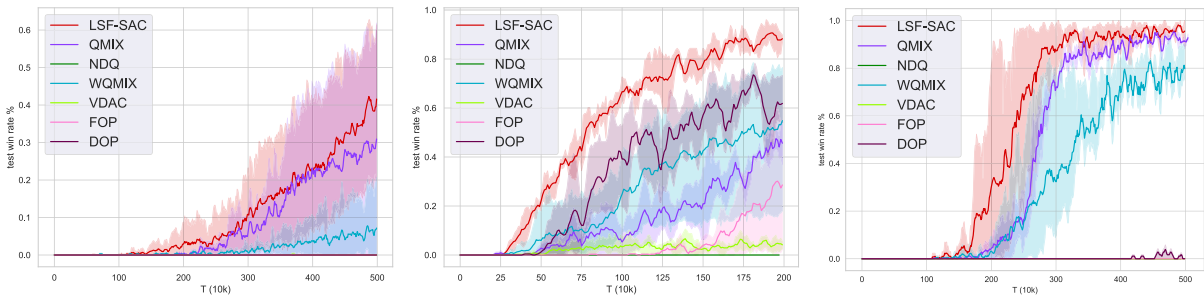


(c) 8m (easy) (d) 1c3s5z (easy)

Fig. 4. Results of 4 easy maps on the SMAC benchmark.



(a) 27m_vs_30m (hard) (b) 3s_vs_5z (hard) (c) 5m_vs_6m (hard)



(d) 6h_vs_8z (super hard) (e) MMM2 (super hard) (f) corridor (super hard)

Fig. 5. Results of hard and super hard maps on the SMAC benchmark.

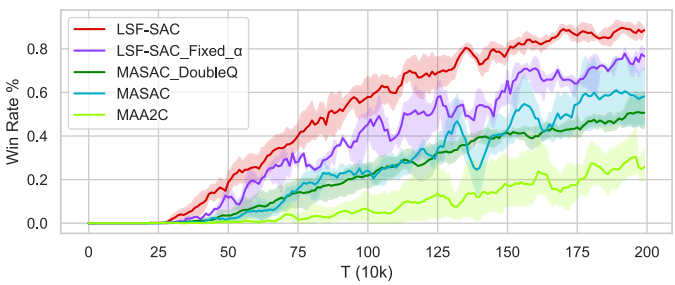


Fig. 6. Ablation Results on MMM2.

2) Ablation 2: We also consider a fixed temperature design as LSF-SAC_Fixed_α with fixed $\alpha = 1.0$ (MASAC $\alpha = 1.0$); this is to understand the effectiveness of the design in automatically updating the temperature α.

3) Ablation 3: We then consider the implementation of a multi-agent soft-actor-critic with value decomposition as MASAC, and the implementation of multi-agent advantage actor-critic with value decomposition as MAA2C, which can be considered as QMIX under a SAC and A2C setting, respectively [38]. This is to find the contribution of soft-actor-critic in enhancing exploration.

4) Ablation 4: Finally we note that the original (single-agent) soft-actor-critic algorithm [10] and several other works use two independently trained soft Q-functions and use the minimum of the two as the policy for optimizing, as [6], [14] points out that policy steps are known to degrade the performance of value-based methods, e.g. in [32] they train with $\mathcal{L}(\theta) = [(r_t + \gamma \min_{j\in 1,2} Q_{tot}((\boldsymbol{s}'_t, \boldsymbol{\tau}'_t, \boldsymbol{a}'_t; \theta_j^-))) - Q_{tot}(\boldsymbol{s}_t, \boldsymbol{\tau}_t, \boldsymbol{a}_t; \theta))^2]$. Their performance comparison can be found in the ablation studies as MASAC_DoubleQ [32]. This is to find if TD advantage with double Q learning is more stable under MARL when combined with value function decomposition.

F. Ablation Results

By comparing the results of MASAC and LSF-SAC, we observe an improvement in both maps regarding the performance of LSF-SAC, which confirms the contribution of the latent state information assisted value decomposition design.

Also, LSF-SAC with $\alpha = 1.0$ is able to achieve a higher winning rate and faster convergence than MASAC. The performance gap between LSF-SAC and MASAC demonstrates the importance of the proposed latent assistive information and our design of entropy maximization specialized for value decomposition methods. The performance gap between LSF-SAC and LSF-SAC with fixed α indicates the necessity of self-updating temperature term in balancing the trade-off between promoting exploration and maximizing the expected rewards.

Finally, although MSAC_DoubleQ delivers a learnable policy at a plodding pace, this could potentially be the result of a complex model and relatively continuous reward in this specific environment. Also, due to its redundant network size, we find that MSAC_DoubleQ, with its double value function design, takes a significantly longer time for training. This proves TD advantage with a single value function might be sufficient to optimize multi-agent actor critics within value decomposition methods. Nevertheless, we observe the design of the double Q network demonstrated the most stable training process with the lowest variance among all ablated baselines.

VII. Conclusion

In this article, we propose LSF-SAC, a novel framework that combines latent state information assisted individual value estimation for joint value function factorization and multi-agent entropy maximization, for collaborative multi-agent reinforcement learning under the CTDE paradigm. We introduce an information-theoretical regularization method for optimizing the latent state information assisted latent information generator to efficiently and effectively utilize extra state information in individual value estimation, while CTDE can still be maintained through a soft-actor-critic design. We also propose one possible implementation of expanding the off-policy maximum entropy deep reinforcement learning to the multi-agent domain with latent state information. Empirical results show that our framework significantly outperforms the baseline methods in the SMAC environment. We further analyze the key factors contributing to the performance in our framework by a set of ablation studies. In future works, we plan to focus on expanding the proposed method with better generation and utilization of the extra state information with theoretical demonstrations of its assisting benefits.

References

[1] A. Alemi, I. Fischer, J. Dillon, and K. Murphy, "Deep variational information bottleneck," in *Proc. Int. Conf. Learn. Representations*, 2019.

[2] A. Das et al., "Targeted multi-agent communication," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 1538–1546.

[3] Y. Duan, X. Chen, R. Houthooft, J. Schulman, and P. Abbeel, "Benchmarking deep reinforcement learning for continuous control," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1329–1338.

[4] J. Foerster, G. Farquhar, T. Afouras, N. Nardelli, and S. Whiteson, "Counterfactual multi-agent policy gradients," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 2974–2982.

[5] J. Foerster et al., "Stabilising experience replay for deep multi-agent reinforcement learning," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 1146–1155.

[6] S. Fujimoto, H. Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 1587–1596.

[7] A. Grover, M. Al-Shedivat, J. Gupta, Y. Burda, and H. Edwards, "Learning policy representations in multiagent systems," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 1802–1811.

[8] J. K. Gupta, M. Egorov, and M. Kochenderfer, "Cooperative multi-agent control using deep reinforcement learning," in *Proc. Int. Conf. Auton. Agents Multiagent Syst.*, 2017, pp. 66–83.

[9] D. Ha and J. Schmidhuber, "World models," 2018, *arXiv:1803.10122*.

[10] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 1861–1870.

[11] Y. Niu, R. Paleja, and M. Gombolay, "Multi-agent graph-attention communication and teaming," in *Proc. AAMAS*, 2021, pp. 964–973.

[12] Y. Du et al., "Learning correlated communication topology in multi-agent reinforcement learning," in *Proc. 20th Int. Conf. Auton. Agents MultiAgent Syst.*, 2021, pp. 456–464.

[13] J. Jiang and Z. Lu, "Learning attentional communication for multi-agent cooperation," in *Proc. Adv. In Neural Inf. Process. Syst.*, 2018, pp. 7265–7275.

[14] H. Hasselt, "Double Q-learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2010, pp. 2613–2621.

[15] I. Higgins et al., "Beta-vae: Learning basic visual concepts with a constrained variational framework," in *Proc. Int. Conf. Learn. Representations*, 2016.

[16] J. Hu, H. Wu, S. Harding, S. Jiang, and S. Liao, "Riit: Rethinking the importance of implementation tricks in multi-agent reinforcement learning," 2021, *arXiv:2102.03479*.

[17] Y. Hu, A. Nakhaei, M. Tomizuka, and K. Fujimura, "Interaction-aware decision making with adaptive strategies under merging scenarios," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2019, pp. 151–158.

[18] M. Igl, L. Zintgraf, T. A. Le, F. Wood, and S. Whiteson,, "Deep variational reinforcement learning for POMDPs," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 2117–2126.

[19] D. P. Kingma, S. Mohamed, D. J. Rezende, and M. Welling, "Semi-supervised learning with deep generative models," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 3581–3589.

[20] D. Kingma and M. Welling, "Auto-encoding variational bayes," 2013, *arXiv:1312.6114*.

[21] L. Kraemer and B. Banerjee, "Multi-agent reinforcement learning as a rehearsal for decentralized planning," *Neurocomputing*, vol. 190, pp. 82–94, 2016.

[22] T. Lillicrap et al., "Continuous control with deep reinforcement learning," 2015, *arXiv:1509.02971*.

[23] M. Littman, "Markov games as a framework for multi-agent reinforcement learning," *Mach. Learn. Proc.,* vol. 1994, pp. 157–163, 1994.

[24] A. Mahajan, T. Rashid, M. Samvelyan, and S. Whiteson, "Maven: Multi-agent variational exploration," in *Proc. 33rd Adv. In Neural Inf. Process. Syst.* 2019, pp. 7613–7624.
[25] V. Mnih et al., "Playing atari with deep reinforcement learning," 2013, *arXiv:1312.5602*.
[26] V. Mnih et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, pp. 529–533, 2015.
[27] I. Mordatch and P. Abbeel, "Emergence of grounded compositional language in multi-agent populations," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 1495–1502.
[28] F. Oliehoek and C. A. Amato, *A Concise Introduction to Decentralized POMDPs.* Berlin, Germany: Springer, 2016.
[29] L. Panait and S. Luke, "Cooperative multi-agent learning: The state of the art," *Auton. Agents Multi-Agent Syst.*, vol. 11, no. 3, pp. 387–434, 2005.
[30] G. Papoudakis and S. Albrecht, "Variational autoencoders for opponent modeling in multi-agent systems," 2020, *arXiv:2001.10829*.
[31] G. Papoudakis, F. Christianos, L. Schüfer, and S. Albrecht, "Comparative evaluation of cooperative multi-agent deep reinforcement learning algorithms," 2020, *arXiv:2006.07869*.
[32] Y. Pu, S. Wang, R. Yang, X. Yao, and B. Li, "Decomposed soft actor-critic method for cooperative multi-agent reinforcement learning," 2021, *arXiv:2104.06655*.
[33] T. Rashid, G. Farquhar, B. Peng, and S. Whiteson, "Weighted qmix: Expanding monotonic value function factorisation for deep multi-agent reinforcement learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020 pp. 10199–10210.
[34] T. Rashid et al., "Monotonic value function factorisation for deep multi-agent reinforcement learning," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 4295–4304.
[35] M. Samvelyan et al., "The starcraft multi-agent challenge," in *Proc. 18th Int. Conf. Auton. Agents MultiAgent Syst.*, 2019, pp. 2186–2188.
[36] J. Shao, H. Zhang, Y. Jiang, S. He, and X. Ji, "Credit assignment with meta-policy gradient for multi-agent reinforcement learning," 2021, *arXiv:2102.12957*.
[37] K. Son, D. Kim, W. Kang, D. Hostallero, and Y. Yi, "Qtran: Learning to factorize with transformation for cooperative multi-agent reinforcement learning," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 5887–5896.
[38] J. Su, S. Adams, and P. Beling, "Value-decomposition multi-agent actor-critics," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 11352–11360.
[39] P. Sunehag et al., "Value-decomposition networks for cooperative multi-agent learning," 2017, *arXiv:1706.05296*.
[40] A. Tampuu et al., "Multiagent cooperation and competition with deep reinforcement learning," *PLoS One.*, vol. 12, 2017, Art. no. e0172395.
[41] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and O. I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," in *Proc. Adv. In Neural Inf. Process. Syst.*, 2017, pp. 6382–6392.
[42] M. Tan, "Multi-agent reinforcement learning: Independent vs. cooperative agents," in *Proc. 10th Int. Conf. Mach. Learn.*, 1993, pp. 330–337.
[43] N. Tishby, F. Pereira, and W. Bialek, "The information bottleneck method," 2000, *Physics/0004057*.
[44] N. Tishby and N. Zaslavsky, "Deep learning and the information bottleneck principle," in *Proc. IEEE Inf. Theory Workshop*, 2015, pp. 1–5.
[45] O. Vinyals et al., "Grandmaster level in StarCraft II using multi-agent reinforcement learning," *Nature*, vol. 575, pp. 350–354, 2019.
[46] J. Wang, Z. Ren, T. Liu, Y. Yu, and C. Zhang, "Qplex: Duplex dueling multi-agent q-learning," 2020, *arXiv:2008.01062*.
[47] T. Wang, J. Wang, C. Zheng, and C. Zhang, "Learning nearly decomposable value functions via communication minimization," in *Proc. Int. Conf. Learn. Representations*, 2019.
[48] Y. Wang, B. Han, T. Wang, H. Dong, and C. Zhang, "Off-policy multi-agent decomposed policy gradients," 2020, *arXiv:2007.12322*.
[49] Y. Yang et al., "Qatten: A general framework for cooperative multiagent reinforcement learning," 2020, *arXiv:2002.03939*.
[50] D. Ye, M. Zhang, and Y. Yang, "A multi-agent framework for packet routing in wireless sensor networks," *Sensors*, vol. 15, pp. 10026–10047, 2015.
[51] T. Zhang, Y. Li, C. Wang, G. Xie, and Z. Lu FOP, "Factorizing optimal joint policy of maximum-entropy multi-agent reinforcement learning," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 12491–12500.
[52] Y. Wang, B. Han, T. Wang, H. Dong, and C. Zhang Dop, "Off-policy multi-agent decomposed policy gradients," in *Proc. Int. Conf. Learn. Representations*, 2020.
[53] W. Böhmer, V. Kurin, and S. Whiteson, "Deep coordination graphs," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 980–991.
[54] K. Son, S. Ahn, R. Reyes, J. Shin, and Y. Yi, "QTRAN++: Improved value transformation for cooperative multi-agent reinforcement learning," 2020, *arXiv:2006.12010*.