# STAR: Spatio-Temporal State Compression for Multi-Agent Tasks with Rich Observations

**Chao Li**[1], **Yujing Hu**[2], **Shangdong Yang**[3,1], **Tangjie Lv**[2], **Changjie Fan**[2], **Wenbin Li**[1*], **Chongjie Zhang**[4] and **Yang Gao**[1*]
[1]State Key Laboratory for Novel Software Technology, Nanjing University
[2]NetEase Fuxi AI Lab
[3]School of Computer Science, Nanjing University of Posts and Telecommunications
[4]Department of Computer Science & Engineering, Washington University in St. Louis
chaoli1996@smail.nju.edu.cn, liwenbin@nju.edu.cn, gaoy@nju.edu.cn

## Abstract

This paper focuses on the problem of learning compressed state representations for multi-agent tasks. Under the assumption of rich observation, we pinpoint that the state representations should be compressed both spatially and temporally to enable efficient prioritization of task-relevant features, while existing works typically fail. To overcome this limitation, we propose a novel method named Spatio-Temporal stAte compRession (STAR) that explicitly defines both spatial and temporal compression operations on the learned state representations to encode per-agent task-relevant features. Specifically, we first formalize this problem by introducing Task Informed Partially Observable Stochastic Game (TI-POSG). Then, we identify the spatial representation compression in it as encoding the latent states from the joint observations of all agents, and achieve this by learning representations that approximate the latent states based on the information theoretical principle. After that, we further extract the task-relevant features of each agent from these representations by aligning them based on their reward similarities, which is regarded as the temporal representation compression. Structurally, we implement these two compression by learning a set of agent-specific decoding functions and incorporate them into a critic shared by agents for scalable learning. We evaluate our method by developing decentralized policies on 12 maps of the StarCraft Multi-Agent Challenge benchmark, and the superior performance demonstrates its effectiveness.

## 1 Introduction

A multitude of real-world problems can be naturally modeled as multi-agent tasks and tentatively solved using multi-agent reinforcement learning (MARL) technology. Although recent works have achieved significant progress in both algorithms (*e.g.,* value decomposition methods [Sunehag *et al.*, 2017; Rashid *et al.*, 2020; Wang *et al.*, 2020a] and multi-agent policy gradient methods [Lowe *et al.*, 2017; Foerster *et al.*, 2018; Yu *et al.*, 2022]) and applications (*e.g.,* robot swarm control [Huang *et al.*, 2020], autonomous vehicles [Cao *et al.*, 2012], traffic management [Wang *et al.*, 2020b] and sensor networks [Zhang and Lesser, 2011]), they heavily depend on direct access to the states of multi-agent tasks to ensure efficient policy learning. However, such direct access to states is frequently unattainable in real-world domains, where agents typically possess partial observations, thereby posing challenges to the efficient learning of policies in these situations.

One promising approach to this problem involves leveraging inherent structures within tasks to identify efficient state representations from the observations of agents. This necessitates an inductive bias towards the underlying structures of tasks. In this study, we assume the characteristic of rich observation. Diverging from its single-agent counterpart [Du *et al.*, 2019], we posit that the joint observation of all agents can uniquely determine the underlying state and the reward function of each agent depends on a small subset of the state (referred to as a sub-state). This structural characteristic is pervasive in many real-world multi-agent scenarios. For example, in the context of traffic signal control, the joint observation of traffic conditions at all intersections determine the state, while the reward function exclusively relies on the average waiting time of current vehicles — a small subset of the complete state. This principle holds true in medical treatment scenarios as well, where the diagnoses of all medical robots collectively determine the patient's true condition. Their reward functions, in turn, hinge on several key factors related to the true illness. Clearly, in tasks characterized by rich observations, we can construct compressed state representations based on the observations and reward functions of agents, thereby enhancing the efficiency of agents' policy learning.

Motivated by this insight, this paper proposes the problem of learning compressed state representations for multi-agent tasks with rich observations, and identifies two sub-processes to solve it. We refer to the first process as the spatial representation compression, which aims to encode the latent states from the joint observations of all agents. The second process aims to further extract task-relevant features (*i.e.,* sub-states relevant to per-agent reward function) from the states, which

*Corresponding authors: Wenbin Li, Yang Gao

needs to compress the learned representations alongside the temporal dimension based on their reward similarities and thus is regarded as the temporal representation compression.

Accordingly, we divide existing MARL methods into two major categories. The first category of works [Rashid *et al.*, 2020; Wang *et al.*, 2020a; Foerster *et al.*, 2018; Yu *et al.*, 2022] enjoys direct access to latent states and implicitly accomplishes temporal representation compression by learning value functions conditioned on the states as supervision signals. However, the necessity for access to latent states and the inefficiency arising from bootstrapping value updates [Fu *et al.*, 2021] limit these methods in practice. On the contrary, the second category of works constructs state representations based on the local observations of agents [Lowe *et al.*, 2017; Iqbal and Sha, 2019] and also uses value functions to achieve implicit representation compression. Nevertheless, they utilize the value function as sole supervision signal for both spatial and temporal representation compression, rendering them less efficient due to the laborious bootstrapping value update.

To overcome above limitations, we propose a novel method named Spatio-Temporal stAte compRession (STAR), which defines explicit spatial and temporal compression operations on the learned representations to encode task-relevant features in the states. Initially, we introduce the Task Informed Partially Observable Stochastic Game (TI-POSG) to formalize the learning of compressed state representations for tasks with rich observations. Subsequently, leveraging the information theoretical principle, we achieve spatial representation compression by compressing the joint observation of all agents into representations approximating the latent states for each agent individually. Afterwards, we further align these representations based on their reward similarities by using the bisimulation metric [Ferns *et al.*, 2004]. This alignment ensures that only task-relevant features are encoded, thereby achieving temporal representation compression. Structurally, we implement these two compression by learning a set of agent-specific decoding functions and incorporate them into a critic shared by agents to facilitate scalable learning.

Experimentally, we evaluate our method on 12 maps of the StarCraft Multi-Agent Challenge (SMAC) [Samvelyan *et al.*, 2019]. The results demonstrate that STAR achieves superior performance across all maps, highlighting its effectiveness.

## 2 Related Work

In this section, we divide existing MARL methods into two major categories according to their ways of compressing state representations and give a brief introduction to them.

The first category of approaches assumes direct access to the states of multi-agent tasks and utilizes value functions as supervision signals to achieve temporal representation compression. Specifically, value decomposition methods, such as QMIX [Rashid *et al.*, 2020] and QPLEX [Wang *et al.*, 2020a], entail learning a factorized global action value function conditioned on the utility functions of all agents and the states. Meanwhile, multi-agent policy gradient methods, exemplified by COMA [Foerster *et al.*, 2018] and MAPPO [Yu *et al.*, 2022], learn value functions (critics) conditioned on the states to optimize agents' decentralized policies (actors). In extending the value decomposition structure into the multi-agent policy gradient paradigm, DOP [Wang *et al.*, 2020c] and FACMAC [Peng *et al.*, 2021] adopt similar critic structures as above value decomposition methods. Despite exhibiting promise in certain tasks, the necessity for direct access to states and the inefficiency arising from bootstrapping value updates limit these methods to more domains in practice.

In contrast, the second category of approaches constructs state representations based on the observations of agents and employs value functions to achieve representation compression. Independent MARL methods such as IQL [Tan, 1993], IPPO [de Witt *et al.*, 2020], and IAC [Christianos *et al.*, 2020] typically fall into this category by learning value functions conditioned on per-agent local observations. However, these local observations inadequately characterize the states, exacerbating the non-stationarity problem and resulting in poor performance. To overcome this limitation, MADDPG [Lowe *et al.*, 2017] concatenates the observations of all agents as proxy states and learns value functions conditioned on them. MAAC [Iqbal and Sha, 2019] further extends MADDPG by incorporating the attention module [Vaswani *et al.*, 2017] to selectively identify relevant features for each agent under the supervision of value functions. However, both methods rely solely on the value function as the supervision signal for both spatial and temporal representation compression, introducing inefficiencies due to the laborious bootstrapping value update.

In summary, current MARL methods exhibit inefficiencies in learning compressed state representations, often overlooking this crucial problem. To address this limitation, this paper first formalizes the learning of compressed state representations for multi-agent tasks with rich observations and then proposes explicit compression operations for the learned representations to encode only task-relevant features. As a result, our work is complementary to these MARL methods through efficiently constructing compressed state representations and thereby accelerating the learning of multi-agent policy.

## 3 Preliminary

In this section, we review some basis concepts about Partially Observable Stochastic Game (POSG) and MARL methods.

### 3.1 POSG

A multi-agent task where agents receive partial observations is usually modeled as a partially observable stochastic game (POSG) $\langle \mathcal{N}, S, \boldsymbol{A}, \boldsymbol{R}, P, \boldsymbol{O}, \boldsymbol{Z}, \gamma \rangle$, where $\mathcal{N} = \{1, 2, \ldots, n\}$ is the agent set and $S$ denotes the state space. $\boldsymbol{A} = A^1 \times A^2 \times \ldots \times A^n$ represents the joint action space of all agents and $A^i$ is the local action space of agent $i$. At each time step $t$, each agent $i$ receives its local observation $o_t^i \in Z^i \in \boldsymbol{Z}$ from its own observation space $Z^i$ according to its observation function $O^i(o_t^i|s_t) \in \boldsymbol{O}$, and selects local action $a_t^i$ by its policy $\pi^i$. Based on agents' joint action $\boldsymbol{a_t} = (a_t^1, a_t^2, \ldots, a_t^n)$, the environment transits to the next state $s_{t+1}$ according to the state transition function $P(s_{t+1}|s_t, \boldsymbol{a_t})$ and provides each agent $i$ with its local reward $r_t^i$ according to its reward function $R^i(s_t, \boldsymbol{a_t}) \in \boldsymbol{R}$. And $\gamma$ is a discount factor. Furthermore, each agent $i$ conditions its policy $\pi^i$ on its local action observation history $\tau_t^i = (o_0^i, a_o^i, \ldots, o_{t-1}^i, a_{t-1}^i, o_t^i)$ to deal with the partial observability challenge.

### 3.2 MARL

We focus on multi-agent policy gradient methods that parameterize each agent $i$'s policy $\pi^i_{\omega^i}$ by $\omega^i$, and directly optimize it to maximize per-agent cumulative rewards using the policy gradient $\nabla_{\omega^i}\mathcal{J}(\omega^i) = \mathbb{E}[\nabla_{\omega^i}\log\pi^i_{\omega^i}(a^i_t|\tau^i_t)G^i]$, where $G^i$ is usually set as the action value function $Q^i$ or the advantage function $A^i$ of agent $i$. To mitigate the non-stationarity problem, $Q^i$ and $A^i$ are conditioned on the states and joint actions of all agents to guarantee stationary policy optimization.

## 4 Problem Formalization

In this section, we begin by illustrating the assumptions inherent in rich observation. Then we introduce the TI-POSG to formalize the learning of compressed state representations.

### 4.1 Assumptions

To establish direct mappings between agents' observations and latent states, we introduce the following assumption:

**Assumption 4.1** (State identification). *For each agent $i$, the joint observation $\hat{o}^i = (o^i, o^{-i})$ comprised by its local observation $o^i$ and other agents $-i$'s observations $o^{-i}$ uniquely determines the underlying state $s$ that generates them.*

The state identification implies the existence of a set of perfect decoding functions $\boldsymbol{q} = (q^1, q^2, \ldots, q^n)$, where $q^i : \hat{\mathcal{Z}}^i \to S$ maps the joint observations $\hat{o}^i \in \hat{\mathcal{Z}}^i$ shaped by agent $i$ to their generating states $s \in S$. Here, $\hat{\mathcal{Z}}^i$ denotes the joint observation space shaped by agent $i$. By approximating the decoding function $q^i$ for each agent $i$, we can recover the latent states from the high dimensional joint observations, which enables tractable learning on the smaller state space.

However, the state space of multi-agent task usually grows exponentially with the number of agents, which presents challenges for efficient learning. Motivated by the fact that each agent's reward function typically depends on a small subset of the state (*i.e.,* sub-state) in many real-world domains, we further make the following assumption about the latent state:

**Assumption 4.2** (Reward relevance). *For each agent $i$, its reward function $R^i$ only depends on a subset $s^{i,+}$ of the full state $s$, while other components $s^{i,-}$ are reward-irrelevant.*

The reward relevance explicitly divides the latent state into two components: task-relevant and task-irrelevant features. For each agent $i$, its reward function $R^i$ is entirely determined by the task-relevant features $s^{i,+}$, and the task-irrelevant features $s^{i,-}$ carry no information about it. This introduces a promising approach to further accelerate the policy learning: compress the state to contain only these task-relevant features for each agent and learn value functions conditioned on them.

### 4.2 TI-POSG

Building upon these assumptions, we introduce the Task Informed Partially Observable Stochastic Game (TI-POSG). To enhance clarity, we present a two-agent TI-POSG example. Fig. 1 (a) illustrates this example from the global perspective, where agents $i$ and $j$ respectively possess their local rewards $r^i_t$ and $r^j_t$, and receive their local observations $o^i_t$ and $o^j_t$ from the latent state $s_t$. This resembles a normal POSG except the state identification property that all agents' joint observations uniquely determine the latent states that generate them.

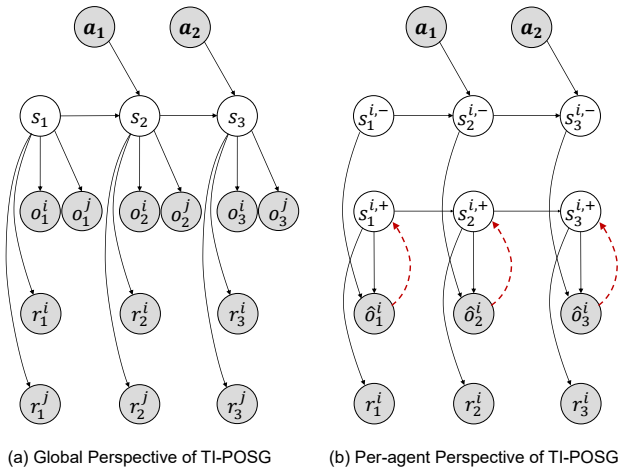


Figure 1: The graphical models of a two-agent TI-POSG from the global perspective (a) and the per-agent perspective (b). The subscripts of all variables indicate time steps $1, 2, 3$ and the superscripts refer to agents $i$ and $j$. Empty and solid circles respectively denote unobservable and observable stochastic variables, while solid lines represent the generative process. For agent $i$, the latent state $s_t$ is divided into two components: task-relevant features $s^{i,+}_t$ and task-irrelevant features $s^{i,-}_t$. A similar division is applied to agent $j$. The objective is to learn state representations for each agent that solely capture its own task-relevant features within the latent states, as indicated by the red arrows with dashed lines. For brevity, we omit the dependency between per-agent rewards and agents' joint actions $\boldsymbol{a_t}$.

We further describe the TI-POSG from a per-agent perspective. As shown in Fig. 1 (b), for agent $i$, its shaped joint observation $\hat{o}^i_t$ is emitted from the latent state $s_t$, which is divided into two components: task-relevant features $s^{i,+}_t$ that wholly determine its local reward $r^i_t$ and task-irrelevant features $s^{i,-}_t$. Our objective is to learn state representations that exclusively capture these task-relevant features based on the joint observation while disregarding other irrelevant features.

Accordingly, we formalize the task of learning compressed state representations for multi-agent tasks with rich observations as learning state representations that characterize only the task-relevant features for each agent, while discarding other irrelevant features. We identify two processes to address this problem. The first process, spatial representation compression, is dedicated to recovering latent states from the joint observations of all agents. The second process, termed temporal representation compression, further extracts only task-relevant features from the states by compressing the learned state representations based on their reward similarities. Building upon this formalization, we propose a practical method in the subsequent section to tackle this problem.

## 5 Methodology

This section gives a comprehensive introduction to our proposed method Spatio-Temporal stAte compRession (STAR).

### 5.1 Spatial Representation Compression

To approximate the unknown decoding function $q^i : \hat{\mathcal{Z}}^i \to S$ for each agent $i$, we consider the information bottleneck (IB)

principle [Tishby *et al.*, 2000]. Let $\tilde{X}$ denote the source message variable and $X$ represent the compressed representation variable of it. In contrast to classic lossless compression where all aspects of $\tilde{X}$ are enforced to retain in $X$, IB seeks to preserve only critical information in $\tilde{X}$ that is relevant to another variable $Y$. Specifically, we formulate the optimization objective of IB by the following equation:

$$\min_{p(x|\tilde{x})} I(X;\tilde{X}) - \beta I(X;Y),$$

where $p(x|\tilde{x})$ denotes the compression function that encodes the original message $\tilde{x} \in \tilde{X}$ into the representation $x \in X$. $I(\cdot\,;\,\cdot)$ represents the mutual information between two random variables and $\beta$ is a Lagrange multiplier that quantifies the amount of information encoded by $X$ from $\tilde{X}$ with respect to identifying $Y$. This trade-off is achieved by minimizing the encoding rate $I(X;\tilde{X})$ and simultaneously maximizing the mutual information $I(X;Y)$ between the compressed representation variable $X$ and the target variable $Y$.

To recover the latent states $s \in S$ from the joint observations $\hat{o}^i \in \hat{\mathcal{Z}}^i$ shaped by agent $i$, we propose learning an encoding function $f^i(x^i|\hat{o}^i) : \hat{\mathcal{Z}}^i \to X^i$ encoding the joint observation $\hat{o}^i \in \hat{\mathcal{Z}}^i$ into the representation $x^i \in X^i$. The goal is for the representation space $X^i$ to be close to the true state space $S$, making the learned encoding function $f^i$ an accurate approximation to the unknown decoding function $q^i$.

Motivated by the fact that the true states strictly adhere to the task dynamics, we enforce that our learned representations $x^i$ encapsulate the most relevant information with respect to the dynamics of current tasks. This ensures consistent optimal policies [Gelada *et al.*, 2019]. Therefore, we set the source message variable as $\hat{\mathcal{Z}}^i$ and the representation variable as $X^i$. [1] The target variable with respect to the task dynamics is set as $\{X^i_{t-1}, \boldsymbol{A}_{t-1}\}$, where $X^i_{t-1}$ and $\boldsymbol{A}_{t-1}$ respectively denote the representation variable and joint action variable at the last time step $t-1$. [2] This leads to the following objective:

$$\min_{f^i(x^i|\hat{o}^i)} I(X^i;\hat{\mathcal{Z}}^i) - \beta I(X^i;\{X^i_{t-1}, \boldsymbol{A}_{t-1}\}). \tag{1}$$

We first minimize the encoding rate $I(X^i;\hat{\mathcal{Z}}^i)$. The encoding rate measures the amount of bits transmitted per message $\hat{\mathcal{Z}}^i$, and the representation dimension resembles the number of bits per message. Thus, minimizing the encoding rate can be achieved by selecting small dimension for the representations [Tao *et al.*, 2020]. We follow this by choosing the smallest dimension for the representations with guarantees that information about task dynamics can still be recorded.

For the second term, we rewrite it as follows:

$$\begin{aligned} &I(X^i;\{X^i_{t-1}, \boldsymbol{A}_{t-1}\}) \\ &\quad= \mathbb{E}_{x^i_{t-1},\boldsymbol{a}_{t-1},x^i} \log \frac{p(x^i|x^i_{t-1},\boldsymbol{a}_{t-1})}{p(x^i)} \\ &\quad= H(X^i) + \mathbb{E}_{x^i_{t-1},\boldsymbol{a}_{t-1},x^i} \log p(x^i|x^i_{t-1},\boldsymbol{a}_{t-1}) \\ &\quad\geq H(X^i) + \mathbb{E}_{x^i_{t-1},\boldsymbol{a}_{t-1},x^i} \log q^\phi(x^i|x^i_{t-1},\boldsymbol{a}_{t-1}), \end{aligned} \tag{2}$$

where $H(X^i)$ denotes the entropy of our learned representations and $q^\phi(x^i|x^i_{t-1},\boldsymbol{a}_{t-1})$ is a variational distribution used to approximate the unknown distribution $p(x^i|x^i_{t-1},\boldsymbol{a}_{t-1})$. Based on Eq. (2), the second term of Eq. (1) can be optimized by maximizing both $H(X^i)$ and the likelihood on dynamics prediction (detailed derivation can be found in Appendix A). To maximize $H(X^i)$, we can choose various techniques, such as state entropy maximization [Seo *et al.*, 2021], random value functions [Osband *et al.*, 2019], or other exploration techniques, which depends on the tasks being solved.

For maximizing $\mathbb{E}\log q^\phi(x^i|x^i_{t-1},\boldsymbol{a}_{t-1})$, we learn a transition function $f_{\mathrm{P}} : X^i_{t-1} \times \boldsymbol{A}_{t-1} \to X^i$ parameterized by $\phi_{\mathrm{P}}$ and a reward function $f_{\mathrm{R}} : X^i_{t-1} \times \boldsymbol{A}_{t-1} \to \mathbb{R}$ parameterized by $\phi_{\mathrm{R}}$. The respective objectives are defined as follows:

$$\begin{aligned} \mathcal{L}_{\mathrm{P}}(\theta^i,\phi_{\mathrm{P}}) &= \mathbb{E}_{(x^i_{t-1},\boldsymbol{a}_{t-1},x^i)\sim\mathcal{D}}[(\bar{x}^i - f_{\mathrm{P}}(x^i_{t-1},\boldsymbol{a}_{t-1}))^2] \\ \mathcal{L}_{\mathrm{R}}(\theta^i,\phi_{\mathrm{R}}) &= \mathbb{E}_{(x^i_{t-1},\boldsymbol{a}_{t-1},r^i_{t-1})\sim\mathcal{D}}[(r^i_{t-1} - f_{\mathrm{R}}(x^i_{t-1},\boldsymbol{a}_{t-1}))^2], \end{aligned} \tag{3}$$

where the encoding function $f^i$ is parameterized by $\theta^i$ and we sample data from a replay buffer $\mathcal{D}$ to approximate the expectation. During training the transition function, we detach the gradient of $x^i$ (we denote it as $\bar{x}^i$) to prevent both $x^i$ and $x^i_{t-1}$ being mapped to zero variables, avoiding representational collapse. To address this issue, we complement the encoding rate loss in Eq. (1) with a relaxed reconstruction task that compels $x^i$ to reconstruct the local observation $o^i$ of agent $i$. By learning a reconstruction function $f_{\mathrm{C}} : X^i \to Z^i$ parameterized by $\phi_{\mathrm{C}}$, we define its loss function as follows:

$$\mathcal{L}_{\mathrm{C}}(\theta^i,\phi_{\mathrm{C}}) = \mathbb{E}_{(x^i,o^i)\sim\mathcal{D}}[(o^i - f_{\mathrm{C}}(x^i))^2]. \tag{4}$$

In summary, the maximization of $I(X^i;\{X^i_{t-1}, \boldsymbol{A}_{t-1}\})$ based on Eq. (3) enables the learned representations to capture information crucial to task dynamics, ensuring accurate approximation of latent states. Simultaneously, we minimize $I(X^i;\hat{\mathcal{Z}}^i)$ by employing small representation dimensions and a relaxed reconstruction. This ensures that the learned representations are compressed concerning the joint observations while also preventing them from collapsing to zero variables.

## 5.2 Temporal Representation Compression

After learning per-agent representations that approximate the latent states, we aim to further extract task-relevant features from these representations while discarding other irrelevance. We propose achieving this extraction by explicitly aligning *similar* state representations that exhibit *similar* task-relevant features, regardless of other task-irrelevant features in them.

Consider the scenario where $s_t$ is divided into $(s^{i,+}_t, s^{i,-}_t)$ from the perspective of agent $i$, and $s_k$ is decomposed into $(s^{j,+}_k, s^{j,-}_k)$ for agent $j$. If we identify that their task-relevant features, $s^{i,+}_t$ and $s^{j,+}_k$, are similar, we should make the corresponding state representations of agents $i$ and $j$ close in the representation space. By doing so, the learned state representations of agents encode only task-relevant features, leading to more compressed representations.

The key is to introduce a specific similarity function. For all agents, this function should be *task-relevant*, solely characterizing similarities between task-relevant features in states.

[1] In this paper we employ the same notation to denote both variable and space for enhancing readability and simplifying notations.

[2] We again overload notations here for clarity.

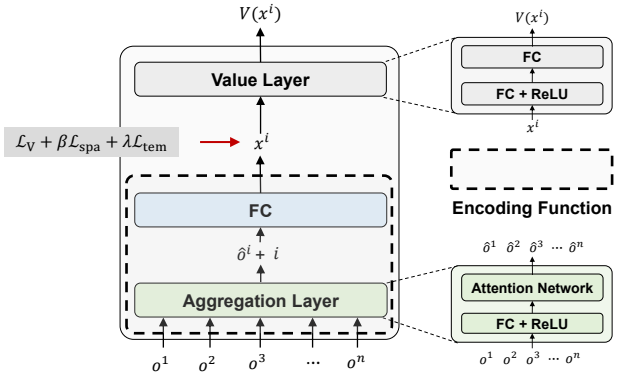


Figure 2: Architecture of the critic shared by agents.

Under the reward relevance assumption that task-relevant features entirely determine per-agent reward function, the cumulative rewards of each agent depend solely on these task-relevant features. Therefore, we propose employing the bisimulation metric [Ferns *et al.*, 2004] as the similarity function, measuring the distance between any two states based on the *task-relevant* element: agents' cumulative rewards. The distance is formalized as the difference in both immediate rewards and one-step transition probabilities.

Specifically, we use $\pi^*$-bisimulation metric [Zhang *et al.*, 2020] to quantify the similarities between task-relevant features in states and update the encoding function to generate similar representations for states with akin task-relevant features. The loss function of it is defined as follows:

$$\mathcal{L}_{\mathrm{tem}}(\theta^i) = \mathbb{E}_{(x_t^i, \boldsymbol{a_t}, r_t^i),(x_k^j, \boldsymbol{a_k}, r_k^j)\sim\mathcal{D}}[(||x_t^i - x_k^j||_1 - \underbrace{|r_t^i - r_k^j| - \gamma W_2(f_{\mathrm{P}}(\bar{x}_t^i, \boldsymbol{a}_t), f_{\mathrm{P}}(\bar{x}_k^j, \boldsymbol{a}_k)))^2}_{\text{the bisimulation metric}}], \quad (5)$$

where $(x_t^i, \boldsymbol{a_t}, r_t^i)$ and $(x_k^j, \boldsymbol{a_k}, r_k^j)$ are two randomly sampled batches from the replay buffer $\mathcal{D}$, and $W_2$ denotes the 2-Wasserstein metric. For our learned deterministic transition function $f_{\mathrm{P}}$, $W_2(f_{\mathrm{P}}(\bar{x}_t^i, \boldsymbol{a}_t), f_{\mathrm{P}}(\bar{x}_k^j, \boldsymbol{a}_k))^2 = ||f_{\mathrm{P}}(\bar{x}_t^i, \boldsymbol{a}_t) - f_{\mathrm{P}}(\bar{x}_k^j, \boldsymbol{a}_k)||_2^2$. Eq. (5) ensures that the L1 distance between any two state representations equals the corresponding bisimulation metric, compelling agent representations to be close when possessing similar task-relevant features and thus resulting in a more compressed state representation space.

### 5.3 Overall Learning Objective

Based on the learned state representations $x^i$, we further learn a state value function $V(x^i) : X^i \to \mathbb{R}$ parameterized by $\phi_{\mathrm{V}}$ and use the value function loss to update them as follows:

$$\mathcal{L}_{\mathrm{V}}(\theta^i, \phi_{\mathrm{V}}) = \mathbb{E}_{(x^i, \hat{R}^i)\sim\mathcal{D}}[(V(x^i) - \hat{R}^i)^2], \quad (6)$$

where $\hat{R}^i$ denotes the discounted reward-to-go of agent $i$.

For scalable learning, we incorporate all agents' encoding functions and the state value function into a critic shared by all agents. As depicted in Fig. 2, the critic is comprised by two primary components: (i) the encoding function $f^i$ shared among agents (it consists of an aggregation layer that shapes the individual joint observation $\hat{o}^i$ for each agent, and a fully connected layer outputting per-agent state representations), and (ii) the value layer responsible for outputting the state values. The overall learning objective is defined as follows:

$$\mathcal{L} = \mathcal{L}_{\mathrm{V}} + \beta\mathcal{L}_{\mathrm{spa}} + \lambda\mathcal{L}_{\mathrm{tem}}, \quad (7)$$

where $\beta$ and $\lambda$ are two weighting factors. $\mathcal{L}_{\mathrm{spa}} = \mathcal{L}_{\mathrm{P}} + \mathcal{L}_{\mathrm{R}} + \mathcal{L}_{\mathrm{C}}$ denotes the spatial compression loss. And $\mathcal{L}_{\mathrm{tem}}$ represents the temporal compression loss, as defined by Eq. (5).

Structurally, we implement the aggregation layer by an attention module [Vaswani *et al.*, 2017], facilitating the efficient aggregation of local observations from all agents. In the subsequent fully connected layer, we introduce one-hot agent labels as additional inputs, augmenting the per-agent joint observations for agent discrimination. The value layer is implemented through a two-layer MLP. More structural and training details regarding our method can be found in Appendix B.

## 6 Experiment

In this section, we evaluate our method by developing decentralized policies for agents on 12 maps of the SMAC benchmark. We carry out our experiments to answer the following questions: (i) Can STAR improve the learning efficiency of these decentralized policies in comparison to other baselines? (see Sec. 6.2) (ii) If so, which component contributes the most to its performance gain? (see Sec. 6.3) (iii) Can the learned representations succeed in characterizing task-relevant features? (See Sec. 6.4). All experimental results are illustrated with the median performance and the standard error over five random seeds. More details are provided in Appendix C.

### 6.1 Settings

We begin by introducing basic settings of our method. Specifically, we instantiate STAR by following the same paradigm as MAPPO, where agents' decentralized policies undergo updates based on advantage functions derived from the state value function. Furthermore, we maximize the representation entropy ($H(X^i)$ in Eq. (2)) by proposing a combined value function that guides agents' policies, defined as follows:

$$V^i = \alpha V_l(o^i) + (1-\alpha)V(x^i),$$

where $V^i$ represents the ultimate value function for agent $i$, and $V_l(o^i)$ denotes a local value function conditioned on the local observation $o^i$ besides $V(x^i)$. $\alpha$ is a diminishing factor that progressively reduces the influence of $V_l(o^i)$ on per-agent policy optimization. The intuition behind this is that the local value function $V_l$ can provide a stable supervision signal for agents' policies during the initial learning process and thus agents are able to generate informative transitions. Then we can use these transitions to update the state representations efficiently, which serves as a warm-up phase. Empirically we find that this combination proves effective for the SMAC task.

### 6.2 Evaluation Performance

To address the first question, we compare our method against various common baseline techniques used to shape state representations. The selected representative methods include:

- IPPO [de Witt *et al.*, 2020]. This method directly uses the local observations as state representations and learns value functions conditioned on them to update policies.

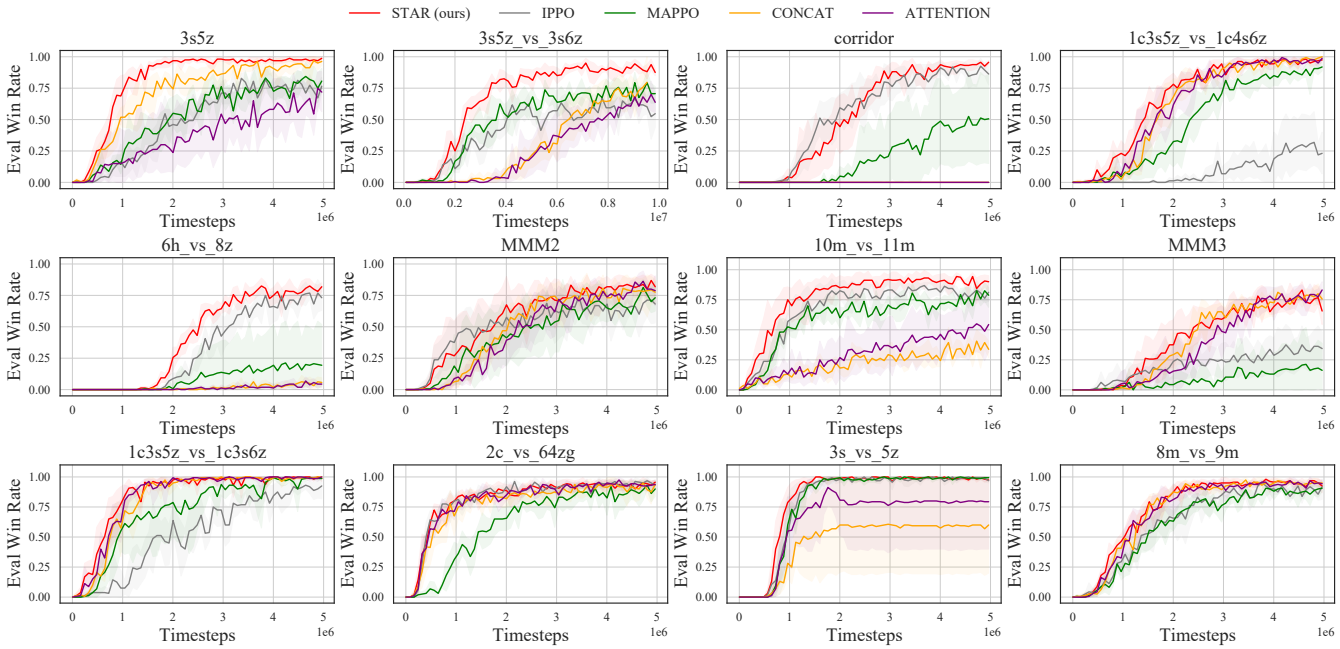


Figure 3: Evaluation performance of our method against multiple baselines on 12 maps.

- CONCAT [Lowe *et al.*, 2017]. In contrast, CONCAT concatenates all agents' local observations as state representations and conditions the value functions on them.
- ATTENTION [Iqbal and Sha, 2019]. ATTENTION uses an attention module within the value function to identify useful components from the observations of all agents.
- MAPPO [Yu *et al.*, 2022]. Unlike approximating state representations based on agents' observations, MAPPO benefits from having access to latent states of tasks and learns value functions conditioned on them.

Fig. 3 shows the comparative results of STAR against these baselines on 12 maps of the SMAC benchmark. One can observe that STAR achieves superior performance on almost all maps, notably outperforming other baselines by a significant margin on challenging maps 3s5z_vs_3s6z (super hard), corridor (super hard), MMM3 (super hard), 6h_vs_8z (super hard), 1c3s5z_vs_1c4s6z (super hard), 3s5z (hard) and 10m_vs_11m (hard). These maps are featured by a large number of units, leading to considerable overlapping information between agents' local observations. Therefore, constructing compressed state representations from these observations proves beneficial for multi-agent policy learning.

In addition to these maps, we find that STAR exhibits slight performance advantages on other maps, including MMM2, 2c_vs_64zg, 3s_vs_5z, 1c3s5z_vs_1c3s6z and 8m_vs_9m. We hypothesize that this is because all methods above are able to quickly learn effective policies for allied agents to defeat the enemies on these maps and generate informative transitions, as demonstrated by their learning curves. Based on these experiences, the state representations learned by these methods can be efficiently compressed by their value functions. Thus, the representational benefits brought by STAR is not obvious.

On the contrary, the baselines CONCAT and ATTENTION are stuck in the redundant joint observations of all agents and their performances are limited by the powerless supervision signals from their value functions. Although they may overcome this limitation by continued training, the learning efficiency may be too terrible. We also notice that IPPO is capable of achieving competitive performance on several maps, and outperforms its counterpart MAPPO. This reveals that the local observations of agents in the SMAC benchmark may include relatively sufficient information about the latent states.

### 6.3 Ablation Study

For the second question, we carry out ablation studies to assess the contributions of STAR's major components: (a) the constraints related to spatial and temporal compression, and (b) the combined value function used to maximize the representation entropy. Additionally, we examine the impact of (c) the scale of the attention module in the aggregation layer.

**Ablation of component (a).** We compare STAR with additional baselines to verify the effectiveness of each constraint. Specifically, these baselines include:

- STAR-No-Temporal: We remove the temporal compression loss $\mathcal{L}_{\text{tem}}$ from Eq. (7) to validate the effect of it.
- STAR-No-Reconstruct: We remove the relaxed reconstruction loss $\mathcal{L}_{\text{C}}$ from Eq. (7) to validate whether representational collapse may occur without this constraint.
- STAR-No-Spatial: The ablation of spatial compression is a bit different because $\mathcal{L}_{\text{P}}$ is used to calculate $\mathcal{L}_{\text{tem}}$. Therefore, we retain only $\mathcal{L}_{\text{P}}$ and $\mathcal{L}_{\text{tem}}$ in Eq. (7) to test the effect caused by the spatial compression loss $\mathcal{L}_{\text{spa}}$.

The results on maps 3s5z and 6h_vs_8z are shown in Fig. 4 (a) and (b). STAR-No-Temporal shows a slight performance

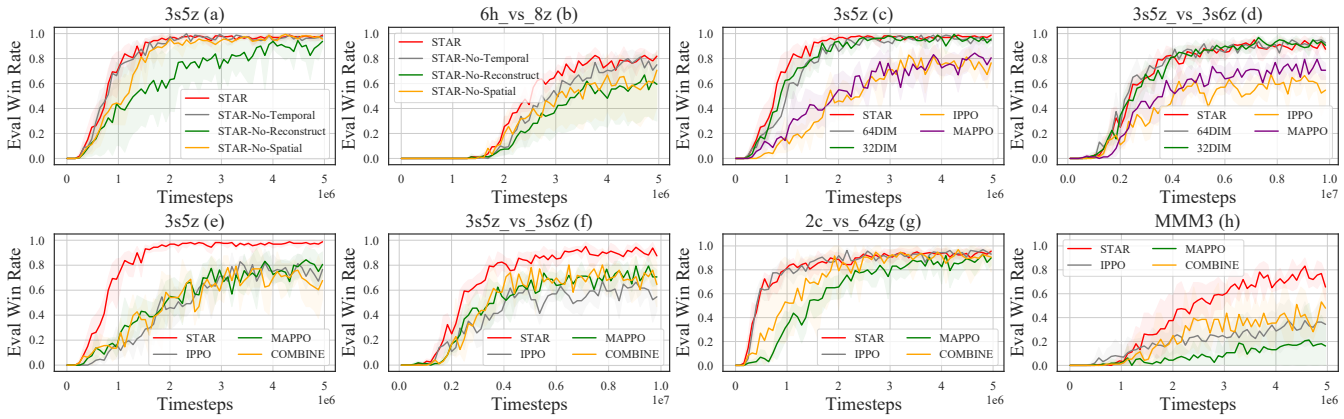


Figure 4: Ablation study with respect to major components of our method.

decrease compared to STAR. This demonstrates that the temporal representation compression can still be achieved by using the value function as a supervision signal, but with low efficiency as evidenced by the relatively slow convergence rate of STAR-No-Temporal on the map 6h_vs_8z.

STAR-No-Reconstruct performs the worst on both maps, which demonstrates that the relaxed reconstruction task protects the state representations from the representation collapse by encoding per-agent local observation into it.

The performance of STAR-No-Spatial is between the performance of the above two baselines. Compared to STAR-No-Reconstruct, STAR-No-Spatial additionally removes the reward prediction loss $\mathcal{L}_{\mathrm{R}}$ besides the reconstruction loss $\mathcal{L}_{\mathrm{C}}$. During the initial learning process where rewards are sparse, $\mathcal{L}_{\mathrm{R}}$ forces the representations to predict these sparse rewards. This prediction of zero vectors may increase the risk of representational collapse occurring and lead to poor performance. The better performance of STAR-No-Spatial over STAR-No-Reconstruct further demonstrates this insight.

**Ablation of component (b).** To validate the effectiveness of maximizing the representation entropy with the combined value function, we compare STAR with a new baseline:

- COMBINE: The COMBINE method similarly employs a combined value function $V^i$, which is comprised by a local value function $V_l(o^i)$ and a state value function $V(s)$ conditioned on the latent state $s$. Except for the state value function, other components of COMBINE, such as the decreasing factor $\alpha$, are the same as STAR.

The comparison results are shown in Fig. 4 (e) - (h). We select two major categories of maps as our testbeds. On the first category, where both IPPO and MAPPO perform well (3s5z and 3s5z_vs_3s6z), we observe that the combined value function has no discernible effect on COMBINE's performance.

In contrast, on the second category of maps where IPPO performs better than MAPPO (2c_vs_64zg), we can find that the warm-up facilitated by the local value function $V_l$ significantly improves the performance of COMBINE in comparison to MAPPO, which solely uses the state value function $V(s)$ for policy updates. This demonstrates that our suggested combined value function can offer useful supervision signals in the initial phase, fostering effective policy learning.

However, the performance improvement attributed to this combined value function is limited by the learned state representations. On the map MMM3, although COMBINE outperforms the baseline MAPPO, there still exist a large margin on the performance between it and STAR. This underscores that STAR primarily benefits from compressed state representations, with the combined value function serving as an auxiliary component in the learning process.

**Ablation of component (c).** We introduce two extra baselines 64DIM and 32DIM to evaluate the effect caused by the scale of attention module in the aggregation layer. These two methods set the units of the attention module to 64 and 32, respectively, with the number of heads set to 1. As depicted in Fig. 4 (c) and (d), their performances are close to that of STAR on maps 3s5z and 3s5z_vs_3s6z but are still significantly better than other baselines. This demonstrates that STAR are robust to the scale of this attention module.

### 6.4 Visualization of the Representation

We give a t-SNE plot of the learned state representations of all agents throughout an entire episode on the map 3s5z. As illustrated in Appendix C, representations with similar cumulative rewards are positioned close to each other, indicating their ability to capture task-relevant features within the states.

## 7 Conclusion

Learning compressed state representations facilitates efficient policy learning for multi-agent tasks with rich observations. This paper formalizes this problem by introducing TI-POSG, and presents a novel method STAR that identifies spatial and temporal representation compression to solve it. Extensive experiments further verify the effectiveness of this method.

**Limitations and Future Work.** Our method depends on learning accurate transition functions of multi-agent tasks to enable efficient representation compression. However, learning precise transition functions for tasks characterized by intricate dynamics proves to be a challenging endeavor. To alleviate this issue, we intend to simplify tasks through the game abstraction. Additionally, we plan to explore the generalization capabilities of the representations learned by our method across similar tasks. We leave them as our future researches.

## Acknowledgments

This work was supported by National Science and Technology Major Project (No.2021ZD0113303), National Natural Science Foundation of China (No.62192783, No.62106100, No.62206133), Primary Research & Development Plan of Jiangsu Province (No.BE2021028), Jiangsu Natural Science Foundation (No.BK20221441), State Key Laboratory of Novel Software Technology Project (No.KFKT2022B12), and Young Elite Scientists Sponsorship Program by CAST (No.2023QNRC001). What's more, the authors greatly thank all anonymous reviewers for their valuable comments.

## References

[Cao *et al.*, 2012] Yongcan Cao, Wenwu Yu, Wei Ren, and Guanrong Chen. An overview of recent progress in the study of distributed multi-agent coordination. *IEEE Transactions on Industrial informatics*, 9(1):427–438, 2012.

[Christianos *et al.*, 2020] Filippos Christianos, Lukas Schäfer, and Stefano Albrecht. Shared experience actor-critic for multi-agent reinforcement learning. *Advances in neural information processing systems*, 33:10707–10717, 2020.

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

[Du *et al.*, 2019] Simon Du, Akshay Krishnamurthy, Nan Jiang, Alekh Agarwal, Miroslav Dudik, and John Langford. Provably efficient rl with rich observations via latent state decoding. In *International Conference on Machine Learning*, pages 1665–1674. PMLR, 2019.

[Ferns *et al.*, 2004] Norm Ferns, Prakash Panangaden, and Doina Precup. Metrics for finite markov decision processes. In *UAI*, volume 4, pages 162–169, 2004.

[Foerster *et al.*, 2018] Jakob Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. Counterfactual multi-agent policy gradients. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[Fu *et al.*, 2021] Xiang Fu, Ge Yang, Pulkit Agrawal, and Tommi Jaakkola. Learning task informed abstractions. In *International Conference on Machine Learning*, pages 3480–3491. PMLR, 2021.

[Gelada *et al.*, 2019] Carles Gelada, Saurabh Kumar, Jacob Buckman, Ofir Nachum, and Marc G Bellemare. Deepmdp: Learning continuous latent space models for representation learning. In *International Conference on Machine Learning*, pages 2170–2179. PMLR, 2019.

[Huang *et al.*, 2020] Yixin Huang, Shufan Wu, Zhongcheng Mu, Xiangyu Long, Sunhao Chu, and Guohong Zhao. A multi-agent reinforcement learning method for swarm robots in space collaborative exploration. In *2020 6th international conference on control, automation and robotics (ICCAR)*, pages 139–144. IEEE, 2020.

[Iqbal and Sha, 2019] Shariq Iqbal and Fei Sha. Actor-attention-critic for multi-agent reinforcement learning. In *International conference on machine learning*, pages 2961–2970. PMLR, 2019.

[Lowe *et al.*, 2017] Ryan Lowe, Yi I Wu, Aviv Tamar, Jean Harb, OpenAI Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. *Advances in Neural Information Processing Systems*, 30, 2017.

[Osband *et al.*, 2019] Ian Osband, Benjamin Van Roy, Daniel J Russo, and Zheng Wen. Deep exploration via randomized value functions. *Journal of Machine Learning Research*, 20:1–62, 2019.

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

[Rashid *et al.*, 2020] Tabish Rashid, Mikayel Samvelyan, Christian Schroeder De Witt, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. Monotonic value function factorisation for deep multi-agent reinforcement learning. *The Journal of Machine Learning Research*, 21(1):7234–7284, 2020.

[Samvelyan *et al.*, 2019] 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. In *International Conference on Autonomous Agents and MultiAgent Systems*, pages 2186–2188, 2019.

[Seo *et al.*, 2021] Younggyo Seo, Lili Chen, Jinwoo Shin, Honglak Lee, Pieter Abbeel, and Kimin Lee. State entropy maximization with random encoders for efficient exploration. In *International Conference on Machine Learning*, pages 9443–9454. PMLR, 2021.

[Sunehag *et al.*, 2017] 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.

[Tan, 1993] Ming Tan. Multi-agent reinforcement learning: independent versus cooperative agents. In *International Conference on International Conference on Machine Learning*, pages 330–337, 1993.

[Tao *et al.*, 2020] Ruo Yu Tao, Vincent François-Lavet, and Joelle Pineau. Novelty search in representational space for sample efficient exploration. *Advances in Neural Information Processing Systems*, 33:8114–8126, 2020.

[Tishby *et al.*, 2000] Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

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

[Wang *et al.*, 2020b] Xiaoqiang Wang, Liangjun Ke, Zhimin Qiao, and Xinghua Chai. Large-scale traffic signal control using a novel multiagent reinforcement learning. *IEEE transactions on cybernetics*, 51(1):174–187, 2020.

[Wang *et al.*, 2020c] Yihan Wang, Beining Han, Tonghan Wang, Heng Dong, and Chongjie Zhang. Dop: Off-policy multi-agent decomposed policy gradients. In *International conference on learning representations*, 2020.

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

[Zhang and Lesser, 2011] Chongjie Zhang and Victor Lesser. Coordinated multi-agent reinforcement learning in networked distributed pomdps. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 25, pages 764–770, 2011.

[Zhang *et al.*, 2020] Amy Zhang, Rowan McAllister, Roberto Calandra, Yarin Gal, and Sergey Levine. Learning invariant representations for reinforcement learning without reconstruction. *arXiv preprint arXiv:2006.10742*, 2020.