Fast Teammate Adaptation in the Presence of Sudden Policy Change

Abstract

In cooperative multi-agent reinforcement learning (MARL), where an agent coordinates with teammate(s) for a shared goal, it may sustain non-stationary caused by the policy change of teammates. Prior works mainly concentrate on the policy change during the training phase or teammates altering cross episodes, ignoring the fact that teammates may suffer from policy change suddenly within an episode, which might lead to miscoordination and poor performance as a result. We formulate the problem as an open Dec-POMDP, where we control some agents to coordinate with uncontrolled teammates, whose policies could be changed within one episode. Then we develop a new framework ***Fast t**eammates **a**da**p**tation (**Fastap**)* to address the problem. Concretely, we first train versatile teammates' policies and assign them to different clusters via the Chinese Restaurant Process (CRP). Then, we train the controlled agent(s) to coordinate with the sampled uncontrolled teammates by capturing their identifications as context for fast adaptation. Finally, each agent applies its local information to anticipate the teammates' context for decision-making accordingly. This process proceeds alternately, leading to a robust policy that can adapt to any teammates during the decentralized execution phase. We show in multiple multi-agent benchmarks that Fastap can achieve superior performance than multiple baselines in stationary and non-stationary scenarios.

1 INTRODUCTION

Cooperative Multi-agent Reinforcement Learning (MARL) has shown great promise in recent years, where multiple agents coordinate to complete a specific task with a shared goal [Oroojlooy and Hajinezhad, 2022], achieving great progress in various domains (e.g., path finding [Sartoretti et al., 2019], active voltage control [Wang et al., 2021], and dynamic algorithm configuration [Xue et al., 2022]). Various methods emerge as promising solutions, including policy-based ones [Lowe et al., 2017, Yu et al., 2022], value-based series [Sunehag et al., 2018, Rashid et al., 2018], and many variants like transformer [Wen et al., 2022], showing remarkable coordination ability in a wide range of tasks like StarCraft multi-agent challenge (SMAC), Google Research Football (GRF) [Gorsane et al., 2022], etc. Other works investigate different aspects, including communication among agents [Zhu et al., 2022], model learning [Wang et al., 2022], policy robustness [Guo et al., 2022], ad hoc teamwork [Mirsky et al., 2022], etc.

However, one issue that can arise in MARL is non-stationarity [Papoudakis et al., 2019] caused by changes in teammates' policies. Non-stationary is a hazardous issue for reinforcement learning, either in single-agent reinforcement learning (SARL) [Padakandla et al., 2019], or MARL [Papoudakis et al., 2019] settings, where the environment dynamic (e.g., transition or reward functions) of a learning system may change over time (inter- or intra-episodes). Many solutions have been developed in SARL to relieve this problem, including meta-reinforcement learning [Beck et al., 2023], strategic retreat [Dastider and Lin, 2022], sticky Hierarchical Dirichlet Process (HDP) prior [Ren et al., 2022], etc. The non-stationary in MARL is, however, much more complex, as we should consider the policy change caused by multiple teammates rather than the single environment dynamic change in SARL. The majority of works in MARL mainly focus on the non-stationary during the training phase [Albrecht and Stone, 2018, Kim et al., 2021], the teammates' policy change across episodes [Qin et al., 2022, Hu et al., 2020], or when perturbations happen [Guo et al., 2022] (See related work in App. A). However, the sudden policy change of teammates when deployed within an episode is never explored to the best of our knowledge, neither in problem formulation nor efficient algorithm design. Ignoring

Submitted to the 39th Conference on Uncertainty in Artificial Intelligence (UAI 2023). **To be used for reviewing only**.

this issue would result in policy shift and even catastrophic miscoordination as agents' policies depend on other teammates in MARL Zhang et al. [2021]. On the other hand, the successful approaches used in SARL are unsuitable for the MARL setting because of the MARL's inherent characteristic (e.g., partial observability). This begs the question: Can we acquire a robust policy that can handle such changes and adapt to the new teammates' polices rapidly?

In this work, we aim to develop a robust coordination policy for the mentioned issue. Concretely, we formulate the problem as an Open Dec-POMDP, where we control multiple agents to coordinate with some uncontrolled teammates, whose policies could be altered unpredictably within one episode. Subsequently, we develop a new training framework Fastap, with which an agent can anticipate the teammates' identification via its local information. Specifically, as similar teammates might possess similarities in their identifications, learning a specific context for each teammate but ignoring the relationships among them could lead to trivial encodings. We thus assign them to different clusters via the Chinese Restaurant Process (CRP) to shrink the context search space. For the controlled coordinating policy training, we sample representative teammates to coordinate with by capturing their identifications into distinguishing contexts to augment the joint policy during the centralized training phase. Each agent then utilizes its local information to approximate the global context information. The mentioned processes proceed alternately, and we can finally obtain a robust policy to adapt to any teammates gradually during the decentralized execution phase.

For evaluation, we conduct experiments on different MARL benchmarks where the teammates' policy alter within one episode, including level-based foraging (LBF) [Papoudakis et al., 2021b], Predator-prey (PP), Cooperative navigation (CN) from MPE [Lowe et al., 2017], and a map created from StarCraft Multi-Agent Challenge (SMAC) [Samvelyan et al., 2019]. Experimental results show that the proposed Fastap can cluster teammates to distinguishing groups, learn meaningful context to capture teammates' identification, and achieve outstanding performance in stationary and non-stationary scenarios compared with multiple baselines.

2 PROBLEM FORMULATION

The aim of this work is to train multiple controllable agents to interact with other teammates that might suddenly change their policies at any time step within one episode. Therefore we formalize the problem by extending the framework of Dec-POMDP [Oliehoek and Amato, 2016] to an Open Dec-POMDP $\mathcal{M} = \langle \mathcal{N}, \bar{\mathcal{N}}, \mathcal{S}, \mathcal{A}, \bar{\mathcal{A}}, P, \Omega, O, R, \mathcal{U}, \gamma \rangle$. Here $\mathcal{N} = \{1, ..., n\}, \bar{\mathcal{N}} = \{\bar{1}, ..., \bar{m}\}$ are the sets of controllable agents and uncontrollable teammates, respectively, $\mathcal{S}$ stands for the set of state, $\mathcal{A} = \mathcal{A}^1 \times ... \times \mathcal{A}^n$ and $\bar{\mathcal{A}} = \mathcal{A}^{\bar{1}} \times ... \times \mathcal{A}^{\bar{m}}$ are the corresponding sets of joint actions for $\mathcal{N}$ and $\bar{\mathcal{N}}$, P, O, R denote the corresponding transition, observation, and reward functions, Ω is the set of observations, $\gamma \in [0, 1)$ is the discounted factor, and $\mathcal{U}$ is a probability distribution used to control the frequency of sudden change.

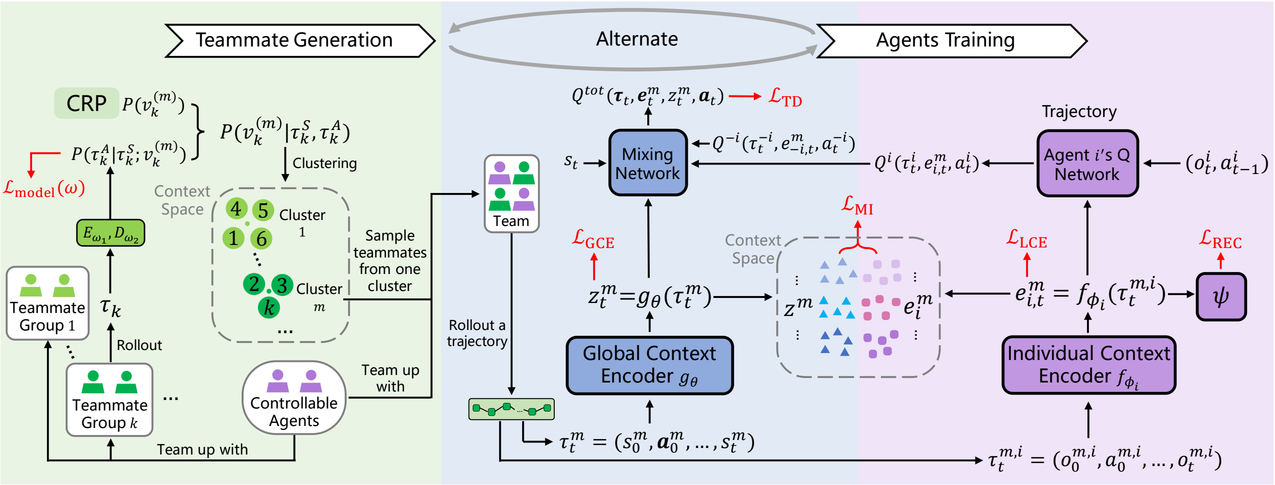
At the beginning of each episode, the set of uncontrollable teammates that participate in the cooperation at the very start is denoted by $\bar{\mathcal{N}}_0 \in \mathcal{P}(\bar{\mathcal{N}})$, where $\mathcal{P}(\cdot)$ stands for the power set, and the waiting time is represented by $u_0 \sim \mathcal{U}$. At each time step t, $u_t = u_{t-1} - 1$ and $\bar{\mathcal{N}}_t = \bar{\mathcal{N}}_{t-1}$ are updated. If $u_t \leq 0$, it will be resampled from $\mathcal{U}$, and a brand new set of uncontrollable teammates $\bar{\mathcal{N}}_t \in \mathcal{P}(\bar{\mathcal{N}})$ will replace the previous one. Meanwhile, controllable agent i receives the observation $o^i = O(s, i)$ and outputs action $a^i \in \mathcal{A}^i$, and so do the uncontrollable teammates. Notice that the number of uncontrollable teammates is changeable in one episode. The joint action $(\boldsymbol{a}, \bar{\boldsymbol{a}})$ leads to the next state $s' \sim P(\cdot|s, (\boldsymbol{a}, \bar{\boldsymbol{a}}))$ and a shared reward $R(s, (\boldsymbol{a}, \bar{\boldsymbol{a}}))$, where $\boldsymbol{a} = (a^1, ..., a^n) \in \mathcal{A}$ and $\bar{\boldsymbol{a}} \in \{(a^{\bar{i}})_{\bar{i} \in \bar{N}} | a^{\bar{i}} \in \mathcal{A}^{\bar{i}}, \bar{N} \in \mathcal{P}(\bar{\mathcal{N}})\}$. To relieve the partial observability, the trajectory history $(o^i_1, a^i_1, ...o^i_{t-1}, a^i_{t-1}, o^i_t)$ of agent i until time step t is encoded into τ^i_t by GRU [Cho et al., 2014]. Under an Open Dec-POMDP, we aim to find an optimal policy when uncontrollable teammates suffer from a sudden change. Then, with $\boldsymbol{\tau}_t = \langle \tau^1_t, ..., \tau^n_t \rangle$, the formal objective is to find a joint policy $\boldsymbol{\pi}(\boldsymbol{\tau}_t, \boldsymbol{a})$ for controllable agents, which maximizes the global value function $Q^{\boldsymbol{\pi}}_{\text{tot}}(\boldsymbol{\tau}, \boldsymbol{a}) = \mathbb{E}_{s,\boldsymbol{a},\bar{\boldsymbol{a}}}[\sum_{t=0}^{\infty} \gamma^t R(s, (\boldsymbol{a}, \bar{\boldsymbol{a}}))| s_0 = s, \boldsymbol{a}_0 = \boldsymbol{a}, \boldsymbol{\pi}, \bar{\boldsymbol{\pi}}]$, where $\bar{\boldsymbol{\pi}}$ is the unknown joint policy of uncontrollable teammates.

3 METHOD

In this section, we will present the detailed design of Fastap (see Fig. 1), a novel multi-agent policy learning approach that enables controllable agents to handle the sudden change of teammates' polices and adapt to new teammates rapidly. First, we design an infinite mixture model that formulates the distribution of continually increasing teammate clusters based on the Chinese Restaurant Process (CRP) [Blei and Frazier, 2010] (Sec 3.1 and Fig. 1(a)). Next, we introduce the centralized context encoder learning objective for fast adaption (Sec 3.2 and Fig. 1(b)). Finally, considering the popular CTDE paradigm in cooperative MARL, we train each controllable agent to recognize and adapt to the teammate situation rapidly according to its local information (Sec 3.3 and Fig. 1(c)).

3.1 CRP-BASED INFINITE MIXTURE FOR DYNAMIC TEAMMATE GENERATION

To adapt to the sudden change in teammates with diverse behaviors in one episode rapidly during evaluation, we expect to maintain a set of diverse policies to simulate the possibly encountered teammates in the training phase. Nevertheless, it is unreasonable and inefficient to consider every newly



Teammate Generation
Alternate
Agents Training
CRP
$P(v_k^{(m)})$
$P(v_k^{(m)}|\tau_k^S, \tau_k^A)$
$P(\tau_k^A|\tau_k^S; v_k^{(m)})$
Clustering
$\mathcal{L}_{\text{model}}(\omega)$
Context Space
4 5 Cluster 1
1 6
2 3 Cluster m
k
...
$E_{\omega_1}, D_{\omega_2}$
Sample teammates from one cluster
Teammate Group 1
τ_k
Rollout
Teammate Group k
...
Controllable Agents
Team up with
Team up with
Team
Rollout a trajectory
$Q^{tot}(\boldsymbol{\tau}_t, \boldsymbol{e}_t^m, z_t^m, \boldsymbol{a}_t)$
$\mathcal{L}_{\text{TD}}$
s_t
Mixing Network
$Q^{-i}(\tau_t^{-i}, e_{-i,t}^m, a_t^{-i})$
$Q^i(\tau_t^i, e_{i,t}^m, a_t^i)$
Trajectory
Agent i's Q Network
(o_t^i, a_{t-1}^i)
$\mathcal{L}_{\text{MI}}$
$\mathcal{L}_{\text{GCE}}$
$\mathcal{L}_{\text{LCE}}$
$\mathcal{L}_{\text{REC}}$
Context Space
$z_t^m = g_\theta(\tau_t^m)$
z^m
e_i^m
$e_{i,t}^m = f_{\phi_i}(\tau_t^{m,i})$
ψ
Global Context Encoder g_θ
Individual Context Encoder f_{ϕ_i}
$\tau_t^m = (s_0^m, \boldsymbol{a}_0^m, ..., s_t^m)$
$\tau_t^{m,i} = (o_0^{m,i}, a_0^{m,i}, ..., o_t^{m,i})$

(a) CRP-Based Infinite Mixture for Dynamic Teammate Generation

(b) Centralized Contextualization Learning for Fast Adaptation

(c) Decentralized Team Situation Recognition

Figure 1: The overall framework of Fastap.

generated group of teammates as a novel type while ignoring the similarities among them. This approach lacks scalability in a learning process where teammates are generated incrementally, and it may lead to reduced training effectiveness if teammates with similar behavior are generated. Accordingly, we expect to acquire clearly distinguishable boundaries of teammates' behaviors by applying a behavior-detecting module to assign teammate groups with similar behaviors to the same cluster. To tackle the issue, an infinite Dirichlet Process Mixture (DPM) model [Lee et al., 2020] could be applied due to its scalability and flexibility in the number of clusters. Concretely, we can formulate the teammate generation process as a stream of teammate groups with different trajectory batch $\mathcal{D}_1, \mathcal{D}_2, ...$ where each batch $\mathcal{D}_k$ is a set of trajectories $\tau = (s_0, \boldsymbol{a}_0 ..., s_T)$ sampled from the interactions between the k^{th} teammate group and the environment, and T is the horizon length. Considering the difficulty of trajectory representation due to its high dimension, we utilize a trajectory encoder E_{ω_1} parameterized by ω_1 to encode τ into a latent space. Specifically, we partition the trajectory τ into $\tau^S = (s_0, ...s_{T-1}, s_T)$ and $\tau^A = (\boldsymbol{a}_0, ..., \boldsymbol{a}_{T-1})$, and a transformer architecture is applied to extract features from the trajectory and represent it as $v = E_{\omega_1}(\tau)$. For the k^{th} teammate group generated so far, $v_k = \mathbb{E}_{\tau_k \sim \mathcal{D}_k}[E_{\omega_1}(\tau_k)]$ will be used to represent its behavioral type, and $\bar{v}^m$ is the mean value of the m^{th} cluster.

If M clusters are instantiated so far, the cluster that the k^{th} teammate group belongs to will be inferred from the assignment $P(v_k^{(m)}|\tau_k) = P(v_k^{(m)}|\tau_k^S, \tau_k^A), m = 1, ..., M, M+1$, where $v_k^{(m)}$ denotes that the k^{th} group belongs to the m^{th} cluster based on its representation v_k. The posterior distribution can be written as:

$$P(v_k^{(m)}|\tau_k^S, \tau_k^A) \propto P(v_k^{(m)}) P(\tau_k^A|\tau_k^S; v_k^{(m)}), \qquad (1)$$

we apply CRP [Blei and Frazier, 2010] to instantiate the DPM model as the prior. Specifically, for a sequence of teammate groups whose representations are $[v_1, v_2, ...v_k, ...]$, the prior $P(v_k^{(m)})$ is set to be:

$$P(v_k^{(m)}) = \begin{cases} \frac{n^{(m)}}{k-1+\alpha}, & m \leq M \\ \frac{\alpha}{k-1+\alpha}, & m = M+1, \end{cases} \qquad (2)$$

where $n^{(m)}$ denotes the number of teammate groups belonging to the m^{th} cluster, M is the number of clusters instantiated so far, $\sum_{m=1}^{M} n^{(m)} = k-1$, and $\alpha > 0$ is a concentration hyperparameter that controls the probability of the instantiation of a new cluster.

To estimate the predictive likelihood $P(\tau_k^A|\tau_k^S; v_k^{(m)})$, we use an RNN-based decoder D_{ω_2} that takes $\tau_k^S, v_k^{(m)}$ as input and predicts τ_k^A. The decoder represents each sample as an Gaussian distribution $\mathcal{N}(\mu(\tau_t^S, v), \sigma^2(\tau_t^S, v))$ where $\tau_t^S = (s_0, ..., s_t)$, such that

$$\begin{aligned} P(\tau_k^A|\tau_k^S; v_k^{(m)}) =& D_{\omega_2}(\tau_k^A|\tau_k^S; v_k^{(m)}) \\ =& \prod_{t=1}^{T} D_{\omega_2}(\boldsymbol{a}_t^k|\tau_{k,t}^S, v_k^{(m)}), \\ \text{where } v_k^{(m)} =& \begin{cases} \frac{n^{(m)}\bar{v}^m + v_k}{n^{(m)}+1} & m \leq M \\ v_k & m = M+1. \end{cases} \end{aligned} \qquad (3)$$

Combing the estimated prior Eq. (2) and predictive likelihood Eq. (3), we are able to decide which cluster the k^{th} teammate group belongs to and thus acquire clearly distinguishable boundaries of teammates' behavior. After the assignment, the mean value of the m^{th} cluster will also be updated. Meanwhile, to force the learned representation v to capture the behavioral information of each teammate group and estimate the predictive likelihood more precisely, the encoder $E_{\omega_1}$and decoder D_{ω_2} are optimized as:

$$\mathcal{L}_{\text{model}}(\boldsymbol{\omega}) = -\log \mathbb{E}_{\tau\sim\cup_{k=1}^{K}\mathcal{D}_k}[D_{\omega_2}(\tau^A|\tau^S; E_{\omega_1}(\tau))], \tag{4}$$

where K is the number of teammate groups generated so far, $\boldsymbol{\omega} = (\omega_1, \omega_2)$. The encoder and decoder are optimized while generating teammate groups (see details in App. B.1).

3.2 CENTRALIZED CONTEXTUALIZATION LEARNING FOR FAST ADAPTATION

After gaining the generated teammates divided into different clusters, this part aims to train a robust policy to handle sudden teammate change and rapidly adapt to the new teammates via conditioning the controllable agents' policies on other teammates' behavior. Despite the diversity and complexity that unknown teammates' behavior exhibits, the CRP formalized before helps acquire clearly distinguishable boundaries based on teammates' behavioral types with regard to high-level semantics. Inspired by Environment Sensitive Contextual Policy Learning (ESCP) [Luo et al., 2022], which aims to guide the context encoder to identify and track the sudden change of the environment rapidly, we expect to utilize a global context encoder g_θ and local context encoder $\{f_{\phi_i}\}_{i=1}^n$ to embed the historical interactions into a compact but informative representation space. The encoders are supposed to identify a new type of teammate fast so as to recognize the sudden change in time, and we can optimize the encoder by proposing an objective that helps the encoder's output coverage to the oracle rapidly at an early time and keep consistent for the remaining steps.

During centralized training phase, we set $z_t^m = g_\theta(\tau_t^m)$, where $\tau_t^m = (s_0^m, \boldsymbol{a}_0^m, ..., s_t^m)$ is generated based on the interactions between the paired joint policy $(\boldsymbol{\pi}, \bar{\boldsymbol{\pi}}^m)$ and the environment, and $\bar{\boldsymbol{\pi}}^m$ is the joint policy of uncontrollable teammates belonging to the m^{th} cluster. Notice that the cluster of teammates is chosen at the beginning of each episode and will not change during training, and sudden change of teammates only happens during evaluation. We can acquire the empirical optimization objective of g_θ as:

$$\mathcal{L}_{\text{GCE}} = \sum_{m=1}^{M} \mathbb{E}[||z_t^m - \bar{z}^m||_2^2] - \log\det(R_{\{\bar{z}^m\}}), \tag{5}$$

where $\bar{z}^m$ is the moving average of all past context vectors used for stabilizing the training process, θ is the parameter of the global context encoder g_θ, $\det(\cdot)$ denotes the matrix determinant, and $R_{\{\bar{z}^m\}}$ is a relational matrix. Intuitively, the objective expects to help the encoder's output coverage rapidly at an early time and keep it consistent for the remaining steps. Specifically, the former part forces z_t^m to converge fast and stably in one episode, and the latter pushes the expectation of z_t^m to a set of separable but representative latent vectors. The full derivation can be found in App. B.2. In practice, a recurrent neural network is applied to instantiate g_θ, which takes $\tau_t^m = (s_0^m, \boldsymbol{a}_0^m, ..., s_t^m)$ as input and outputs a multivariate Gaussian distribution $\mathcal{N}(\mu_\theta(\tau_t^m), \sigma_\theta^2(\tau_t^m))$. Thus the teammates context is obtained from the Gaussian distribution with the reparameterization trick by $z_t^m \sim g_\theta(\tau_t^m)$. As we can apply Fastap to any value-based methods, the global embedding z_t^m could also be integrated into the centralized network. Similarly, the local embedding $e_t^{m,i}$ and local trajectory $\tau_t^{m,i}$ will also be concatenated to calculate the local Q-value $Q^i(\tau_t^{m,i}, e_t^{m,i}, \cdot)$, where the optimization of the local context encoder will be explained in detail in the next part. Therefore, the TD loss $\mathcal{L}_{\text{TD}} = [r_t^m + \gamma \max_{\boldsymbol{a}_{t+1}^m} \bar{Q}_{\text{tot}}(s_{t+1}^m, \boldsymbol{e}_{t+1}^m, z_{t+1}^m, \boldsymbol{a}_{t+1}^m) - Q_{\text{tot}}(s_t^m, \boldsymbol{e}_t^m, z_t^m, \boldsymbol{a}_t^m)]$ is utilized to accelerate the centralized contextualization learning, where $\bar{Q}_{\text{tot}}$ is periodically updated target Q network, and $\boldsymbol{e}_t^m = (\boldsymbol{e}_t^{m,i})_{i=1}^n$. The overall optimization objective of g_θ can thus be derived:

$$\mathcal{L}_{\text{ADAP}} = \mathcal{L}_{\text{TD}} + \alpha_{\text{GCE}}\mathcal{L}_{\text{GCE}}, \tag{6}$$

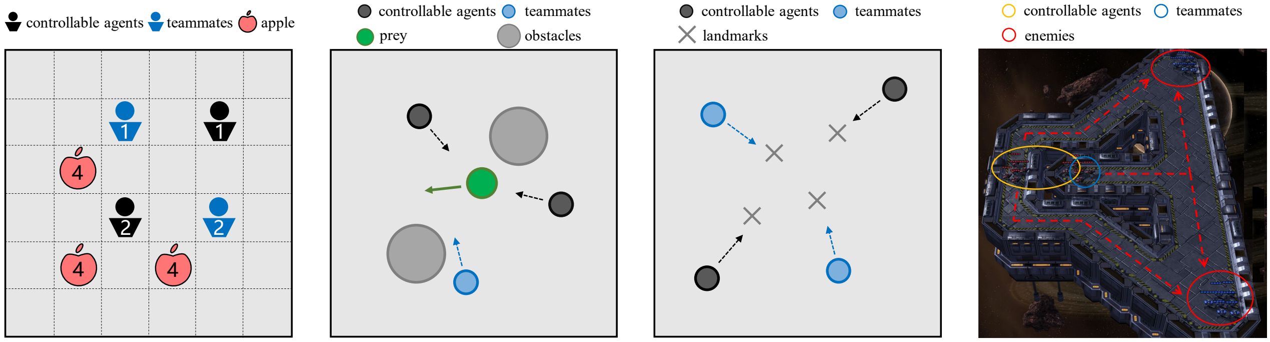
where α_{GCE} is an adjustable hyper-parameter to balance the two optimization objective.

3.3 DECENTRALIZED TEAM SITUATION RECOGNITION AND OPTIMIZATION

Despite the fact that optimizing Eq. (6) helps obtain compact and representative representations z_t^m that could guide individual policies to adapt to teammate sudden change rapidly, partial observability of MARL will not allow agents that execute in a decentralized manner to obtain z_t^m encoded from the global state-action trajectory. Thus, we equip each agent i with a local encoder f_{ϕ_i} to recognize the team situation. Concretely, the network architecture of f_{ϕ_i} is similar to g_θ, f_{ϕ_i} takes local trajectory $\tau_t^{m,i} = (o_0^{m,i}, a_0^{m,i}, ..., o_t^{m,i})$ as input and outputs $e_t^{m,i} \sim \mathcal{N}(\mu_{\phi_i}(\tau_t^{m,i}), \sigma_{\phi_i}^2(\tau_t^{m,i}))$. To make $e_t^{m,i}$ informatively consistent with z_t^m, we introduce a mutual information (MI) objective by maximizing the MI $\mathcal{I}(e_t^{m,i}; z_t^m|\tau_t^{m,i})$ between $e_t^{m,i}$ and z_t^m conditioned on the agent i's local trajectory $\tau_t^{m,i}$. Due to the difficulty and feasibility of estimating the conditional distribution directly, variational distribution $q_\xi(e_t^{m,i}|z_t^m, \tau_t^{m,i})$ is used to approximate the conditional distribution $p(e_t^{m,i}|z_t^m, \tau_t^{m,i})$. Inspired by the information bottleneck [Alemi et al., 2017], we would derive a tractable lower bound of MI objective:

$$\begin{aligned}&\mathcal{I}(e_t^{m,i}; z_t^m|\tau_t^{m,i}) \geq \\ &\qquad \mathbb{E}_{\mathcal{D}}[\log q_\xi(e_t^{m,i}|z_t^m, \tau_t^{m,i})] + \mathcal{H}(e_t^{m,i}|\tau_t^{m,i}),\end{aligned} \tag{7}$$

where $\mathcal{H}(\cdot)$ denotes the entropy, and variables of the distributions are sampled from the experience replay buffer $\mathcal{D}$. We defer the full derivation to App. B.3. We can now rewrite



(a) LBF (b) PP (c) CN (d) 10m_vs_14m

Figure 2: Experimental environments used in this paper.

the MI objective as:

$$\mathcal{L}_{\text{MI}} = \sum_{m=1}^{M}\sum_{i=1}^{n} \mathbb{E}_{\mathcal{D}}[\log q_{\xi}(e_t^{m,i}|z_t^m, \tau_t^{m,i})] + \mathcal{H}(e_t^{m,i}|\tau_t^{m,i}), \tag{8}$$

the mentioned symbols are defined similarly as Eq. (5). To facilitate the learning process, two local auxiliary optimization objectives are further designed. On the one hand, we expect $e_t^{m,i}$ to recognize the team situation and adapt to new teammates that change suddenly as z_t^m does:

$$\mathcal{L}_{\text{LCE}} = \sum_{m=1}^{M}\sum_{n=1}^{n} \mathbb{E}[||e_t^{m,i} - \bar{e}^{m,i}||_2^2] - \log\det(R_{\{\bar{e}^{m,i}\}}). \tag{9}$$

On the other hand, to derive the descriptive representation $e_t^{m,i}$ of the specific team situation, we hope $e_t^{m,i}$ can learn the relationship between controllable agents and the teammates. Therefore, we expect $e_t^{m,i}$ to reconstruct the observations and actions taken by teammates:

$$\mathcal{L}_{\text{REC}} = \sum_{m=1}^{M}\sum_{n=1}^{n} \mathbb{E}_{\mathcal{D}}[-\log h_{\psi_i}(\bar{\boldsymbol{o}}_t^m, \bar{\boldsymbol{a}}_t^m|e_t^{m,i})], \tag{10}$$

where h is parameterized by ψ_i for each agent i. As $e_t^{m,i}$ and $\tau_t^{m,i}$ will be concatenated into the input of individual Q network $Q^i(\tau_t^{m,i}, e_t^{m,i}, \cdot)$, the TD loss $\mathcal{L}_{\text{TD}}$ is also utilized to promote the learning of local context encoder. Thus, the optimization objective becomes:

$$\mathcal{L}_{\text{DEC}} = \mathcal{L}_{\text{TD}} + \alpha_{\text{MI}}\mathcal{L}_{\text{MI}} + \alpha_{\text{LCE}}\mathcal{L}_{\text{LCE}} + \alpha_{\text{REC}}\mathcal{L}_{\text{REC}}, \tag{11}$$

where $\alpha_{\text{MI}}, \alpha_{\text{LCE}}, \alpha_{\text{REC}}$ are the corresponding adjustable hyperparameters of the three objectives.

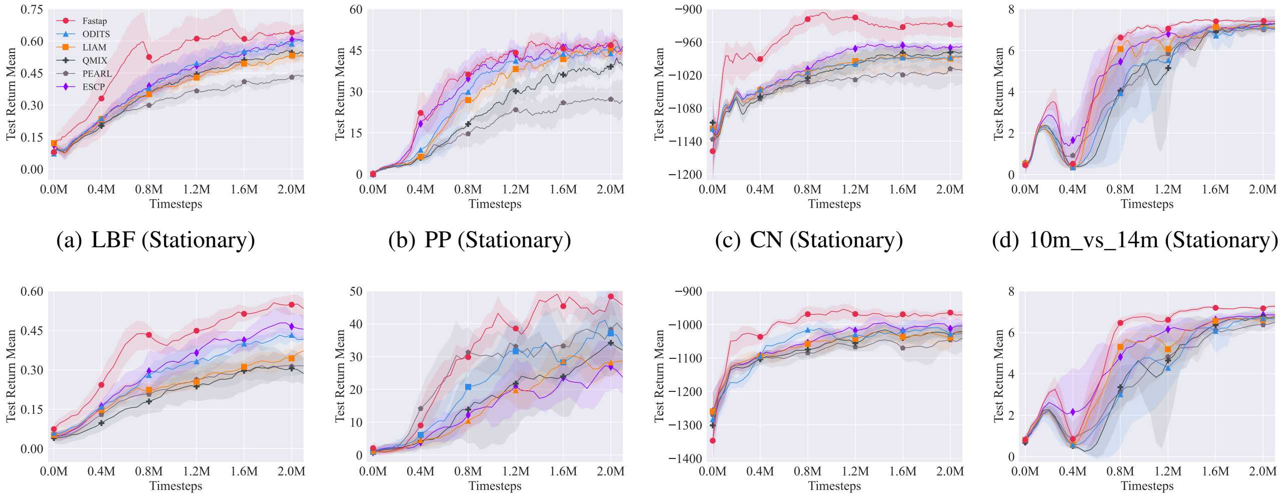
4 EXPERIMENTS

In this section, we design extensive experiments for the following questions: 1) Can Fastap achieve high adaptability and generalization ability when encountering teammate sudden change compared to other baselines in different scenarios, and how each component influences its performance? (Sec. 4.2) ? 2) Can CRP help acquire distinguishable boundaries of teammates' behaviors, and what team situation representation is learned by Fastap (Sec. 4.3)? 3) What transfer ability Fastap reveals, and how does each hyperparameter influence its coordination capability (Sec. 4.4)?

4.1 ENVIRONMENTS AND BASELINES

We select four multi-agent tasks as our environments, as shown in Fig. 2. Level Based Foraging (LBF) [Papoudakis et al., 2021b] is a cooperative grid world game with agents that are rewarded if they concurrently navigate to the food and collect it. Predator-prey (PP) and Cooperative navigation (CN) are two scenarios coming from the MPE environment [Lowe et al., 2017], where multiple agents (predators) need to chase and encounter the adversary agent (prey) to win the game in PP, and in CN, multiple agents are trained to move towards landmarks while avoiding collisions with each other. We also create a map 10m_vs_14m from SMAC [Samvelyan et al., 2019], where 10 allies are spawned at different points to attack 14 enemies to win.

For baselines, we consider multiple ones and implement them to a popular valued-based method QMIX [Rashid et al., 2018] for comparisons, including (1) the vanilla QMIX without any extra design; (2) Meta-learning SARL methods: PEARL [Rakelly et al., 2019] uses recently collected context to infer a probabilistic variable describing the task; ESCP [Luo et al., 2022] copes with the sudden change in the environment by learning a context-sensitive policy; (3) Context-based MARL approaches: LIAM [Papoudakis et al., 2021a] predicts teammates' current behaviors based on local observation history to relieve non-stationary in the training phase; ODITS [Gu et al., 2022] applies a centralized "teamwork situation encoder" for end-to-end learning to adapt to arbitrary teammates across episodes. More details about the environments and baselines, and Fastap are illustrated in App. C, and App. D, respectively.



(a) LBF (Stationary) (b) PP (Stationary) (c) CN (Stationary) (d) 10m_vs_14m (Stationary)

(e) LBF (Non-Sta.) (f) PP (Non-Sta.) (g) CN (Non-Sta.) (h) 10m_vs_14m (Non-Sta.)

Figure 3: Performance comparison with baselines on multiple benchmarks.

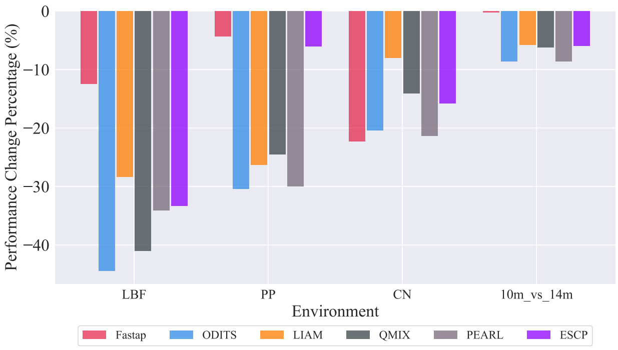
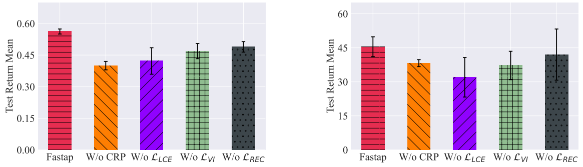


Figure 4: Performance difference in stationary and non-stationary conditions. The value is the difference in the performance under non-stationary and stationary settings w.r.t. the best return.

4.2 COMPETITIVE RESULTS AND ABLATIONS

Coordination Ability in Stationary and Non-stationary Settings At first glance, we compare Fastap against the mentioned baselines to investigate the coordination ability under stationary and non-stationary conditions, as shown in Fig. 3. We can find all algorithms suffer from coordination ability degradation when teammates are in a non-stationary manner, indicating a specific consideration of teammates' policy sudden change in a non-stationary environment is needed. When only using local information to obtain a context to capture the teammates' information, methods like PERAL and LIAM show indistinctive coordination improvement in stationary and non-stationary settings, PEARL performs even worse than vanilla QMIX, demonstrating that successful meta-learning approaches in SARL cannot be implemented without modification in the MARL setting.



(a) LBF (b) PP

Figure 5: Ablation Studies.

Furthermore, when learning a teammate's behavior context extraction model in both global and local ways, ODITS shows superior performance in the two mentioned conditions, manifesting the necessity of utilizing global states to improve training efficiency. Besides, ESCP also reveals a relatively better coordination capability, demonstrating the effectiveness of optimizing a context encoder with fast adaptability. Fastap achieves the best performance on all benchmarks both in stationary and non-stationary conditions, and suffers from the least performance degradation when tested in a non-stationary condition in most environments (see Fig. 4), showing the effectiveness and high efficiency of the proposed method.

Ablation Studies As Fastap is composed of multiple components, we here design ablation studies on benchmarks LBF and PP to investigate how they impact the coordination performance of Fastap under non-stationary settings. First, for the infinite mixture model of dynamic teammate generation, we derive *W/o CRP* by removing the CRP process and taking each newly generated teammate group as a new cluster. Next, to explore whether a teammate-behavior-sensitive encoder helps improve adaptability, we introduce *W/o LCE* by removing $\mathcal{L}_{\text{LCE}}$ of local encoders. Furthermore,

$\mathcal{U}$	Fastap	Fastap_wo_CRP	ODITS	LIAM	QMIX	PEARL	ESCP
stationary	$\mathbf{0.642 \pm 0.008}$	0.594 ± 0.015	0.637 ± 0.008	0.597 ± 0.029	0.569 ± 0.033	0.507 ± 0.021	0.618 ± 0.040
$U[5,8]$	$\mathbf{0.562 \pm 0.012}$	0.400 ± 0.020	0.352 ± 0.002	0.415 ± 0.026	0.306 ± 0.038	0.288 ± 0.019	0.404 ± 0.026
$U[6,7]$	$\mathbf{0.567 \pm 0.001}$	0.444 ± 0.314	0.487 ± 0.022	0.454 ± 0.157	0.444 ± 0.221	0.333 ± 0.000	0.556 ± 0.125
$U[2,9]$	0.484 ± 0.285	0.222 ± 0.133	0.416 ± 0.182	0.401 ± 0.078	0.443 ± 0.205	0.205 ± 0.114	$\mathbf{0.514 \pm 0.314}$
$U[3,6]$	$\mathbf{0.518 \pm 0.136}$	0.366 ± 0.217	0.444 ± 0.314	0.388 ± 0.283	0.353 ± 0.272	0.264 ± 0.066	0.502 ± 0.120
$U[3,3]$	$\mathbf{0.384 \pm 0.272}$	0.246 ± 0.141	0.342 ± 0.118	0.362 ± 0.208	0.222 ± 0.314	0.243 ± 0.172	0.271 ± 0.157

Table 1: The final average return ± std in LBF, where $\mathcal{U}$ is the sudden change probability distribution of open Dec-POMDP that controls the frequency of sudden change, and $U[m,n]$ denotes a discrete uniform distribution parameterized by m and n. The row of the original training sudden change distribution $\mathcal{U} = U[5,8]$ is highlighted as gray.

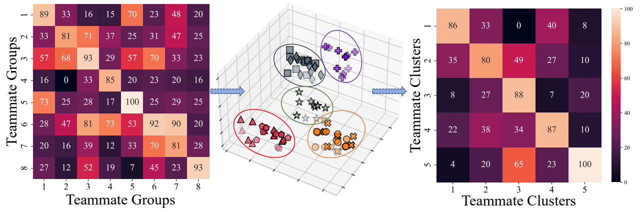


Figure 6: Cross-Play performance before and after CRP and teammate behavior embeddings.

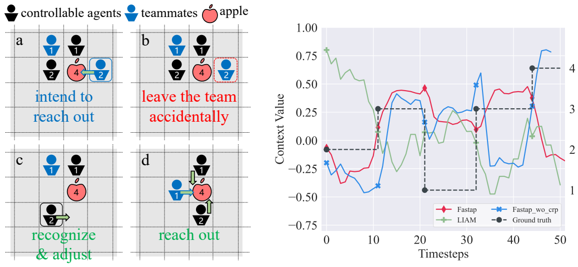
we pick up *W/o MI* to investigate how maximizing mutual information between global and local contexts accelerates learning efficiency. Finally, *W/o REC* is introduced to check the impact of the auxiliary optimization objective that involves agent modeling. As is shown in Fig. 5, *W/o CRP* and *W/o MI* suffer the most severe performance degradation in LBF and PP, respectively, manifesting the benefit of the introduction of CRP model and that teammate-behavior-sensitive encoders do help agents adapt to sudden change of teammates rapidly. Besides, when removing $\mathcal{L}_{\text{MI}}$, the performance gap *W/o MI* shows in two benchmarks demonstrate the necessity of utilizing global information to facilitate the learning of local context encoders. Finally, we also find agent modeling helps learn more informative context and brings about a slight coordination improvement.

Comparisons in (OOD) Non-stationary Setting. As this study considers a setting where the frequency of uncontrolled teammates' sudden change follows a fixed probability distribution $\mathcal{U}$, which is set to be a uniform distribution, we evaluate the generalization ability when altering the changing frequency during testing. The experiments on LBF are conducted with the distribution $\mathcal{U} = U[5,8]$ during training. As shown in Tab. 1, we compare the final returns of different learned policies in LBF by altering the distribution $\mathcal{U}$. Although different approaches obtain similar coordination ability in stationary conditions, they suffer from strong performance degradation when altering teammates' policy-changing frequency (e.g., ODITS suffer from close to half performance degradation in sudden change[3, 3]). On the other hand, Fastap and ESCP achieve outstanding generalization ability in both in-distribution and OOD settings mostly. More specifically, in the stationary setting, Fastap outperforms the best baseline ODITS by 0.005, while in the original non-stationary setting, the gap increases to 0.147. We also find Fastap shows inferiority to ESCP in setting sudden change[2, 9], we believe that both methods fail to perform well under the 2-timestep sudden change interval, while Fastap sacrifices a part of the performance under large timestep sudden change interval that might happen in $U[2,9]$. A more robust policy in diverse conditions would be developed in the future.

4.3 TEAMMATE ADAPTATION ANALYSIS

Here we conduct experiments to investigate the CRP model and teammate adaptation progress. We first verify whether CRP helps acquire distinguishable boundaries of teammates' behaviors by performing Cross-Play [Hu et al., 2020] experiments on LBF before and after CRP. As shown in the left part of Fig. 6, for generation process of 8 teammate groups, we find that the values on the diagonal from the top left to the bottom right are relatively larger. However, several high performances of other points (e.g., Teammate groups 2 and 3) indicate that the generated teammate groups might share similar behavior. To help relieve the negative influence caused by taking teammate groups with similar behavior as two different types, CRP is applied to learn the behavior type and assign teammates with similar behavior to the same cluster. Further, we sample latent variables generated by $E_{\omega_1}(\tau_k)$ and reduce the dimensionality by principal component analysis (PCA) [Wold et al., 1987]. We find that latent variables assigned to the same cluster (the ellipse) are distributed in the adjacent areas. Cross-Play experiments are also conducted on the teammate clusters after CRP, and we find from the right part of Fig. 6 that teammates belonging to different clusters achieve low performance when paired together, indicating the effectiveness of CRP.

To investigate how teammate-behavior-sensitive encoders help adapt to teammates' sudden change rapidly, we also visualize the fragment snapshot of an episode during testing as shown in Fig. 7(a). When a teammate and two controlled agents are trying to reach out for an apple and win the score as they were intended, the teammate accidentally leaves out the team, and they fail to get the reward provisionally. However, the controlled agents learned by Fastap recognize



(a) Snapshot

(b) Context Curve

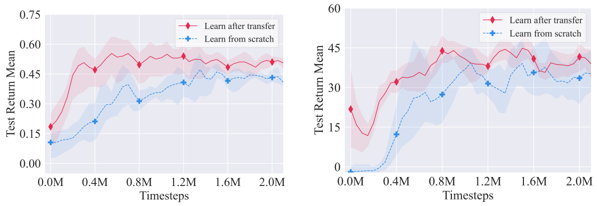
Figure 7: Teammate adaptation visualization.

the situation and switch out the policy rapidly by moving downward and coordinating with the other teammate to attain the reward. Meanwhile, we record the latent context vector in different timesteps of one episode. Fastap encodes the context to four-dimensional vectors in LBF, and we reduce the dimensionality to one-dimensional scalars by PCA. We scatter the points in Fig. 7(b) together with the contexts learned by LIAM and ablation Fastap_wo_CRP. The results imply that the contexts learned by Fastap are sensitive to the sudden change of teammates, and when the teammates are stable, the latent context is stable and flat. Despite the fact that agent modeling helps recognize the teammates' behavior, the context curve of LIAM is still hysteretic and unstable. Meanwhile, the ablation Fastap_wo_CRP can also adapt to new teammates rapidly, but it fails to recognize the teammates with similar behavior and results in the unstable latent context (e.g., Teammate Cluster 3).

4.4 TRANSFER AND SENSITIVE STUDIES

Our Fastap learns teammates recognition module to cope with teammates that might change suddenly in one episode. The sudden change distribution $\mathcal{U}$ that controls the frequency of changing is fixed, and a more frequent change or a larger gap of waiting interval tends to make the training more difficult. Here, we investigate the policy transfer ability of Fastap by comparing the performance after fine-tuning and learning from scratch. Concretely, we train Fastap agents under the sudden change distribution $\mathcal{U}_{\text{source}} = U[5, 8]$ for 0.6M timesteps and initialize the trained network with the saved checkpoint under the target setting with $\mathcal{U} = \mathcal{U}_{\text{target}} = U[3, 6]$. The learning curves demonstrated in Fig. 8 show that agents trained under $\mathcal{U}_{\text{source}}$ possess a jumpstart compared with the random initialization, and we hope it could accelerate the learning in a new environment by reusing previously learned knowledge.

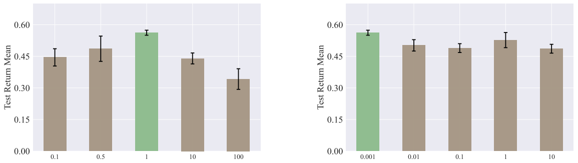
As Fastap includes multiple hyperparameters, here we conduct experiments on benchmark LBF to investigate how each one influences the coordination ability. First, α_{GCE} balances the trade-off between the TD-loss and the global context optimization object. If it is too small, agents may



(a) LBF

(b) PP

Figure 8: Policy Transfer Ability.



(a) Sensitivity of α_{GCE}

(b) Sensitivity of α_{MI}

Figure 9: Sensitivity Studies on LBF.

coordinate in stationary environment excessively, ignoring the extraction of teammates context information. On the other hand, if it is too large, agents pay much attention to teammates identification with risk of overfitting to specific teammates types. We thus find each hyperparameter via grid-search. As shown in Fig. 9(a), we can find that $\alpha_{\text{GCE}} = 1$ is the best choice in this benchmark. α_{MI} influences the optimization of local encoder f_{ϕ_i} to recognize the team situation. Fig. 9(b) shows that $\alpha_{\text{cont}_g} = 0.001$ performs the best. $\alpha_{\text{LCE}}, \alpha_{\text{REC}}$ can be found in a similar way (see App. E).

5 FINAL REMARKS

In this work, we study the teammates' adaptation problem when some coordinators suffer from the sudden policy change. We first formalize this problem as an open Dec-POMDP, where some coordinators from a team may sustain policy changes unpredictably within one episode, and we train multiple controlled agents to adapt to this change rapidly. For this goal, we propose Fastap, an efficient approach to learn a robust multi-agent coordination policy by capturing the teammates' policy-changing information. Extensive experimental results on stationary and non-stationary conditions from different benchmarks verify the effectiveness of Fastap, and more analysis results also confirm it from multiple aspects. Our method can be seen as a primary attempt for the open-environment setting [Zhou, 2022] in cooperative MARL, and we sincerely hope it can be a solid foothold for applying MARL to practical applications. For future work, researches on the changing of action/observation space of the MARL system or utilizing techniques like transformer [Vaswani et al., 2017] to obtain a generalist coordination policy for non-stationary from diverse sources and degrees is of great value.

References

Stefano V. Albrecht and Peter Stone. Autonomous agents modelling other agents: A comprehensive survey and open problems. *Artificial Intelligence*, 258:66–95, 2018.

Alexander A. Alemi, Ian Fischer, Joshua V. Dillon, and Kevin Murphy. Deep variational information bottleneck. In *ICLR*, 2017.

Jacob Beck, Risto Vuorio, Evan Zheran Liu, Zheng Xiong, Luisa Zintgraf, Chelsea Finn, and Shimon Whiteson. A survey of meta-reinforcement learning. *preprint arXiv:2301.08028*, 2023.

David M. Blei and Peter I. Frazier. Distance dependent chinese restaurant processes. In *ICML*, pages 87–94, 2010.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *EMNLP*, pages 1724–1734, 2014.

Apan Dastider and Mingjie Lin. Non-parametric stochastic policy gradient with strategic retreat for non-stationary environment. In *CASE*, pages 1377–1384. IEEE, 2022.

Rihab Gorsane, Omayma Mahjoub, Ruan John de Kock, Roland Dubb, Siddarth Singh, and Arnu Pretorius. Towards a standardised performance evaluation protocol for cooperative MARL. In *NeurIPS*, 2022.

Pengjie Gu, Mengchen Zhao, Jianye Hao, and Bo An. Online ad hoc teamwork under partial observability. In *ICLR*, 2022.

Jun Guo, Yonghong Chen, Yihang Hao, Zixin Yin, Yin Yu, and Simin Li. Towards comprehensive testing on the robustness of cooperative multi-agent reinforcement learning. *preprint arXiv:2204.07932*, 2022.

Hengyuan Hu, Adam Lerer, Alex Peysakhovich, and Jakob N. Foerster. "other-play" for zero-shot coordination. In *ICML*, pages 4399–4410, 2020.

Dong-Ki Kim, Miao Liu, Matthew Riemer, Chuangchuang Sun, Marwa Abdulhai, Golnaz Habibi, Sebastian Lopez-Cot, Gerald Tesauro, and Jonathan P. How. A policy gradient algorithm for learning to learn in multiagent reinforcement learning. In *ICML*, pages 5541–5550, 2021.

Soochan Lee, Junsoo Ha, Dongsu Zhang, and Gunhee Kim. A neural dirichlet process mixture model for task-free continual learning. In *ICLR*, 2020.

Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. In *NIPS*, pages 6379–6390, 2017.

Fan-Ming Luo, Shengyi Jiang, Yang Yu, Zongzhang Zhang, and Yi-Feng Zhang. Adapt to environment sudden changes by learning a context sensitive policy. In *AAAI*, pages 7637–7646, 2022.

Reuth Mirsky, Ignacio Carlucho, Arrasy Rahman, Elliot Fosong, William Macke, Mohan Sridharan, Peter Stone, and Stefano V Albrecht. A survey of ad hoc teamwork: Definitions, methods, and open problems. *preprint arXiv:2202.10450*, 2022.

Frans A Oliehoek and Christopher Amato. *A Concise Introduction to Decentralized POMDPs*. Springer, 2016.

Afshin Oroojlooy and Davood Hajinezhad. A review of cooperative multi-agent deep reinforcement learning. *Applied Intelligence*, pages 1–46, 2022.

Sindhu Padakandla, Prabuchandran K. J., and Shalabh Bhatnagar. Reinforcement learning algorithm for non-stationary environments. *Applied Intelligence*, pages 1–17, 2019.

Georgios Papoudakis, Filippos Christianos, Arrasy Rahman, and Stefano V Albrecht. Dealing with non-stationarity in multi-agent deep reinforcement learning. *preprint arXiv:1906.04737*, 2019.

Georgios Papoudakis, Filippos Christianos, and Stefano Albrecht. Agent modelling under partial observability for deep reinforcement learning. In *NeurIPS*, pages 19210–19222, 2021a.

Georgios Papoudakis, Filippos Christianos, Lukas Schäfer, and Stefano V Albrecht. Benchmarking multi-agent deep reinforcement learning algorithms in cooperative tasks. In *NeurIPS*, 2021b.

Rongjun Qin, Feng Chen, Tonghan Wang, Lei Yuan, Xiaoran Wu, Zongzhang Zhang, Chongjie Zhang, and Yang Yu. Multi-agent policy transfer via task relationship modeling. *preprint arXiv:2203.04482*, 2022.

Kate Rakelly, Aurick Zhou, Chelsea Finn, Sergey Levine, and Deirdre Quillen. Efficient off-policy meta-reinforcement learning via probabilistic context variables. In *ICML*, pages 5331–5340, 2019.

Tabish Rashid, Mikayel Samvelyan, Christian Schroeder, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning. In *ICML*, pages 4295–4304, 2018.

Hang Ren, Aivar Sootla, Taher Jafferjee, Junxiao Shen, Jun Wang, and Haitham Bou-Ammar. Reinforcement learning in presence of discrete markovian context evolution. In *ICLR*, 2022.

Mikayel Samvelyan, Tabish Rashid, Christian Schröder de Witt, Gregory Farquhar, Nantas Nardelli, Tim G. J. Rudner, Chia-Man Hung, Philip H. S. Torr, Jakob N. Foerster, and Shimon Whiteson. The Starcraft multi-agent challenge. In *AAMAS*, pages 2186–2188, 2019.

Guillaume Sartoretti, Justin Kerr, Yunfei Shi, Glenn Wagner, TK Satish Kumar, Sven Koenig, and Howie Choset. Primal: Pathfinding via reinforcement and imitation multi-agent learning. *IEEE Robotics and Automation Letters*, 4 (3):2378–2385, 2019.

Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinícius Flores Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z Leibo, Karl Tuyls, and Thore Graepel. Value-decomposition networks for cooperative multi-agent learning based on team reward. In *AAMAS*, pages 2085–2087, 2018.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, pages 5998–6008, 2017.

Jianhong Wang, Wangkun Xu, Yunjie Gu, Wenbin Song, and Tim C. Green. Multi-agent reinforcement learning for active voltage control on power distribution networks. In *NeurIPS*, pages 3271–3284, 2021.

Xihuai Wang, Zhicheng Zhang, and Weinan Zhang. Model-based multi-agent reinforcement learning: Recent progress and prospects. *preprint arXiv:2203.10603*, 2022.

Muning Wen, Jakub Grudzien Kuba, Runji Lin, Weinan Zhang, Ying Wen, Jun Wang, and Yaodong Yang. Multi-agent reinforcement learning is a sequence modeling problem. In *NeurIPS*, 2022.

Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987.

Ke Xue, Jiacheng Xu, Lei Yuan, Miqing Li, Chao Qian, Zongzhang Zhang, and Yang Yu. Multi-agent dynamic algorithm configuration. In *NeurIPS*, 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. In *NeurIPS*, 2022.

Kaiqing Zhang, Zhuoran Yang, and Tamer Başar. Multi-agent reinforcement learning: A selective overview of theories and algorithms. *Handbook of Reinforcement Learning and Control*, pages 321–384, 2021.

Zhi-Hua Zhou. Open-environment machine learning. *National Science Review*, 9(8), 2022.

Changxi Zhu, Mehdi Dastani, and Shihan Wang. A survey of multi-agent reinforcement learning with communication. *preprint arXiv:2203.08975*, 2022.