NEVER REVISIT: CONTINUOUS EXPLORATION IN MULTI-AGENT REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Recently, intrinsic motivations are wildly used for exploration in multi-agent reinforcement learning. We discover that coming with intrinsic rewards is the issue of revisitation – the relative values of intrinsic rewards fluctuate, causing a sub-space visited before becomes attractive after a period of exploration to other areas. Consequently, agents risk exploring some sub-spaces repeatedly. In this paper, we formally define the concept of revisitation, based on which we propose an observation-distribution matching approach to detect the appearance of revisitation. To avoid it, we add branches to agents' local Q-networks and the mixing network to separate sub-spaces which have already been revisited. Furthermore, to prevent adding branches excessively, we design intrinsic rewards to reduce the probability of and penalize the occurrence of revisitation. By virtue of these advances, our method achieves superior performance on three challenging Google Research Football (GRF) scenarios with sparse rewards.

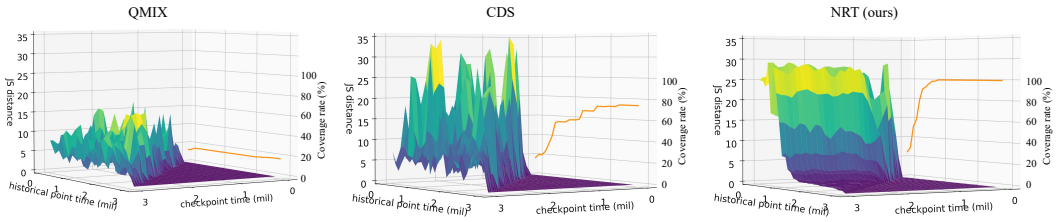


Figure 1: The issue of revisitation. JS distance between joint-observation distribution induced by policies at different training time steps is shown. Experiments are carried out on a 6×12 maze task introduced in Sec. 5. **Left**: Basic exploration scheme (QMIX (Rashid et al., 2018) with ϵ-greedy exploration) achieves a coverage rate less than 20%. **Middle**: Adding CDS (Li et al., 2021) intrinsic incentives improves the coverage rate to 80%. However, the fluctuating JS distances indicate that agents are periodically revisiting some sub-spaces. Consequently, most samples are wasted on revisitation. **Right**: Revisitation is avoided based on our method, and the JS distances are large and stable before converging to an optimal strategy.

1 INTRODUCTION

Multi-agent cooperation is ubiquitous in real-world problems, such as sensor networks (Zhang & Lesser, 2013) and traffic light control (Zhang et al., 2019). To introduce intelligence into multi-agent systems and achieve sophisticated cooperative behavior, multi-agent reinforcement learning (MARL) has been gaining increasing interest in recent years. Advanced MARL methods have largely pushed forward the performance of machine learning algorithms on tasks such as StarCraft II micromanagement (Rashid et al., 2018; Wang et al., 2021b), Hanabi (Bard et al., 2020; Foerster et al., 2019), and robotic control (Kurin et al., 2020).

Despite these achievements, a problem persists and prevents MARL from extending successfully to more complex problems. The action-observation space grows exponentially with the number of agents, and the efficiency of exploration strategies in such search spaces largely limits the learning efficiency of MARL algorithms. Basic exploration (Wang et al., 2020b) schemes, like ϵ-greedy,

adopted by many previous works (Wang et al., 2021a; Yu et al., 2021; de Witt et al., 2020) appear to struggle even in tasks with a moderate number of agents. For example, in a 6×12 maze game with two agents (Fig. 4 (b)), QMIX (Rashid et al., 2018) using independent search can only cover less than 20% of the state space (Fig. 1 left) and struggles to find any rewards.

Various intrinsic motivations are introduced into MARL algorithms to enhance their exploration ability by encouraging interaction among agents (Wang et al., 2020b), maximizing a measurements of behavioral randomness (Houthooft et al., 2016; Mahajan et al., 2019; Gupta et al., 2021), and spurring individuality (Jiang & Lu, 2021) and diversity (Li et al., 2021). These methods significantly enlarge the sub-space that can be explored. For example, the coverage rate increases from $\sim 20\%$ to $\sim 80\%$ using the CDS (Li et al., 2021) diversity-encouragement incentives (Fig. 1 middle). However, in this paper, we find that coming with the augmented exploration is a ***revisitation*** problem that severely hurts the expected exploration ability and prevents learning efficiently.

Revisitation refers to the situation where, with an enlarged exploration space, agents forget the areas they have visited before after exploring a recently sub-space for a while, causing them to return back and re-explore. Revisitation may also occur when the intrinsic motivation in previously explored sub-spaces becomes relatively high after exploring other sub-spaces. This issue becomes more detrimental when revisitation in some sub-spaces happens repeatedly. For example, in Fig. 1 (middle), we present the empirical joint observation distribution induced by CDS (Li et al., 2021) policies at different training time steps. It can be observed that similar distributions occur periodically during learning, and the algorithm wastes most training samples on unvalued, revisited experiences.

In this paper, we give a formal definition for revisitation in multi-agent settings, based on which we propose a framework for solving this issue. Our idea is to add branches to agents' local Q-networks and the mixing network to separate sub-spaces that have been revisited. A branch selection scheme is proposed so that upon receiving an observation, agents can decide how to quickly reach under-explored sub-spaces instead of wasting samples on revisitation. Moreover, to avoid an excessive number of branches in the network, we design intrinsic rewards to prevent revisitation from occurring and penalize detected revisitation.

Based on the above merits for detecting and preventing revisitation, our approach NRT (**N**ever **R**evisi**t**) achieves nearly 100% coverage rate on the maze task (Fig. 1 right). Furthermore, we benchmark our approach on Google Research Football (GRF, Kurach et al. (2020)), and find that NRT significantly pushes forward state of the art. Ablation studies show that adding branches is the most important component in revisitation avoidance and performance improvement in challenging tasks. At the same time, the proposed intrinsic incentives are critical in achieving stable learning performance.

2 BACKGROUND

In this paper, we formulate multi-agent coordination tasks as Dec-POMDP Oliehoek et al. (2016), which can be defined as a tuple $\mathcal{G} = \langle I, S, A, P, R, O, n, \gamma \rangle$, where I is the set of agents, S is the state space, A is the action space, P is the transition function, R is the reward function, O is the observation space, n is the number of agents, and $\gamma \in [0, 1)$ is the discount factor. During sampling, each agent $i \in I$ observes its unique information o_i and selects an action $a_i \in A$ independently. According to the joint action $\boldsymbol{a}$ and environment transition function $P(s'|s, \boldsymbol{a})$, the environment transfers to a new state s' and provides a reward that is shared across all agents.

2.1 CENTRALIZED TRAINING WITH DECENTRALIZED EXECUTION

Our approach follows a recently advanced multi-agent optimization framework of centralized training with decentralized execution (CTDE) Lowe et al. (2017); Foerster et al. (2018); Sunehag et al. (2018); Rashid et al. (2018). In this framework, each agent executes only based on its local action-observation history. This distributed decision-making tackles the exponentially growing joint action space. For stable optimization, agents are trained in a centralized manner with access to global information to achieve reasonable credit assignment across agents. The individual-global-max (IGM) principle Son et al. (2019) further guarantees the consistency between the local and global Q-function based on value factorization. With the above advantages, some algorithms Son et al. (2020); Wang et al. (2021a); Ma et al. (2021) have achieved remarkable progress in challenging MARL cooperation tasks.

2.2 CDS

Although the CTDE framework has been proven to accelerate training with value factorization, it still has to face a giant search space. Diversity in exploration is required for sophisticated cooperation. Li et al. (2021) introduced CDS, which achieves diversity when necessary via shared experiences. Some of the most salient novelties are: (1) Introduce an identity-aware diversity based on $I^{\pi}(\tau_T; id)$ to encourage specialty of individual trajectories. (2) Provide an independent Q-network for each agent to estimate local Q-value after combining with one shared Q-network. (3) Add an L1 regularization on independent Q-values to encourage sharing crucial knowledge while keeping enough policy representation capacity of necessary diversity.

3 METHOD

When the search space is large in the multi-agent settings, revisitation to previously explored sub-spaces is likely to happen when agents learn in a recently discovered sub-space and forget the earlier experience. Our idea to solve this problem is first to detect the appearance of revisitation. If revisitation is detected, we will augment each agent's local Q-network following a new credit assignment method so that agents can keep continuous exploration. Correspondingly, in this section, we introduction our algorithm for revisitation detection (Sec. 3.1), augmentation of the training framework (Sec. 3.2), and auxiliary incentive design supporting these new features (Sec. 3.3). Fig. 2 and Fig. 3 provide a quick overview of our method.

3.1 REVISITATION DETECTION

We first define the concept of ***revisitation*** in multi-agent settings. Our definition is based on the empirical joint observation distribution induced by the joint policy $\boldsymbol{\pi}$: $\rho_{\boldsymbol{\pi}}(\boldsymbol{o}) = \mathbb{E}_{\boldsymbol{\pi}}[\frac{1}{T}\sum_{t=0}^{T}\mathbf{1}_{\boldsymbol{o}_t=\boldsymbol{o}}]$. Intuitively, when $\rho_{\boldsymbol{\pi}}(\boldsymbol{o})$ changes significantly, the agent team starts to explore a new sub-space. Still, if, after a significant change, $\rho_{\boldsymbol{\pi}}(\boldsymbol{o})$ degenerates to a distribution that has occurred before, revisitation happens. Formally, we have the following definition:

Definition 1. *A δ-revisitation occurs at time step t if there exists a time step $t'' < t$ so that*

1. $\max_{t' \in (t'', t)} D_{\text{JS}}\left[\rho_{\boldsymbol{\pi}_{t''}}(\boldsymbol{o}) \| \rho_{\boldsymbol{\pi}_{t'}}(\boldsymbol{o})\right] > \delta$;
2. $D_{\text{JS}}\left[\rho_{\boldsymbol{\pi}_{t''}}(\boldsymbol{o}) \| \rho_{\boldsymbol{\pi}_{t}}(\boldsymbol{o})\right] < \delta$.

Here, D_{JS} is the Jensen-Shannon distance between two distributions. The first condition describes the ability of many multi-agent exploration algorithms. As the learning progresses, agents are encouraged to explore new sub-spaces, leading to a shift in the empirical joint observation distribution. The second condition depicts the occurrence of revisitation. After exploring a new sub-space for a while, the agents may forget previously learned knowledge, possibly due to limited network representative capacity, and start to re-explore previously visited sub-spaces.

In practice, we estimate the joint observation distribution $\rho_{\boldsymbol{\pi}}(\boldsymbol{o})$ using an a variational auto-encoder parameterized by ξ, by minimizing the evidence lower bound:

$$\mathcal{L}_{\rho}(\xi) = \mathbb{E}_{\text{q}(z|\boldsymbol{o};\xi)}[\log p(\boldsymbol{o}, z; \xi) - \log q(z \mid \boldsymbol{o}; \xi)], \tag{1}$$

where z is a latent variable to reconstruct $\boldsymbol{o}$.

During training, for every $\bar{T}$ training steps, we store the current joint observation distribution estimator p_{ξ_t} into a buffer, and check for each previous version $p_{\xi_{t''}}$ whether they satisfy the two conditions listed in Definition 1. If the conditions are satisfied, we call (t'', t', t) a ***revisitation tuple*** and store it in a set $\mathcal{R}$. The set $\mathcal{R}$ thereby stores all the revisitation encountered during the training process.

3.2 AUGMENTATION OF THE TRAINING FRAMEWORK

After the detection of revisitation, the next step is to avoid repeated exploration in a sub-space. Intuitively, we augment the representation of policies to separate revisited regions for continuous exploration. Practically, when a revisitation (t'', t', t) is detected, we copy the latest Q-function

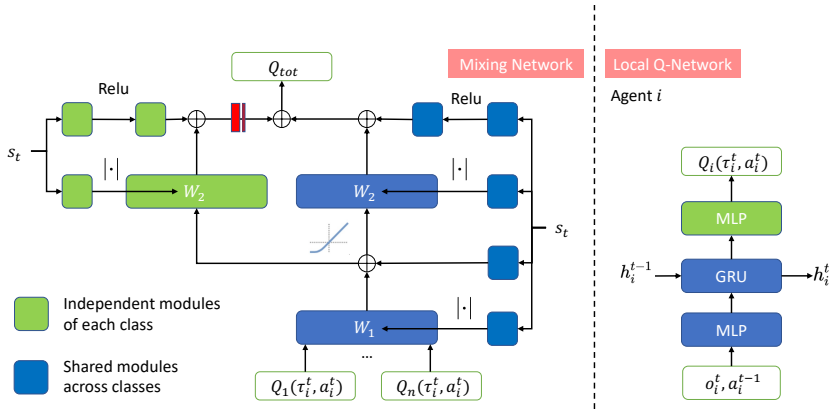


Figure 2: NRT (**N**ever **R**evisi**t**)'s network structure.

version as a new branch. After that, We further augment the mixing network in the CTDE framework to achieve diverse credit assignments across agents. For memory efficiency, the branches only include the last layer of the mixing network and the local utility network, as shown in Fig. 2. We call each branch a ***class***. For each class C except the earliest one, we denote the corresponding revisitation tuple as $(t_C^{''}, t_C^{'}, t_C)$. Inspired by CDS's partially shared local Q-network structure, we introduce a partially shared mixing network for each class to achieve diverse agent credit assignments while maintaining sufficient experience sharing. The red item added to the independent part of each class is an L1 regularization term for filtering out useless diversity.

Each policy class is created to avoid revisitation in a subset of the joint observation space. A question is how to identify this subset for each class. The answer to this question also determines which branch should be used during learning and execution when joint observation $\boldsymbol{o}$ is encountered.

Formally, for each class C except the earliest one, the associated joint observation set is defined as:

$$\Omega_C = \{\boldsymbol{o} || \rho_{\boldsymbol{\pi}_{t''}}(\boldsymbol{o}) - \rho_{\boldsymbol{\pi}_{t'}}(\boldsymbol{o})\}| > \eta\} \cap \{\boldsymbol{o} || \rho_{\boldsymbol{\pi}_{t''}}(\boldsymbol{o}) - \rho_{\boldsymbol{\pi}_{t}}(\boldsymbol{o})\}| < \eta\}, \tag{2}$$

where η is one hyper-parameter related to δ in Definition 1.

When encountering joint observation $\boldsymbol{o}$ during training and execution, we check for each class in the increasing order in which they are created. According to each agent's independent observation, the first class satisfying the marginal condition in Eq. 2 is selected in the decentralized execution manner. The earliest branch is activated if no augmented classes based on revisitation are chosen. More clearly, the earliest branch created at the beginning of learning can be called the default branch or the latest version.

3.3 Auxiliary Supporting Incentives

A possible shortcoming of the methods introduced in previous sections is that when revisitation occurs frequently, there may be too many classes. To prevent the number of classes from increasing quickly, we design incentives to (1) prevent the appearance of revisitation in advance, and (2) penalize the occurrence of revisitation.

To prevent revisiting, we punish agents for visiting observations similar to historical ones. Upon receiving a new observation $\boldsymbol{o}_t$, we sample several (10 for all our experiments) trajectories from the replay buffer, and penalize if they are close to $\boldsymbol{o}_t$. This intrinsic motivation is defined as

$$\alpha_t = \frac{1}{\sqrt{\sum_{\boldsymbol{o} \in \mathcal{N}_k(\boldsymbol{o}_t)} K(\boldsymbol{o}_t, \boldsymbol{o})} + \epsilon}, \tag{3}$$

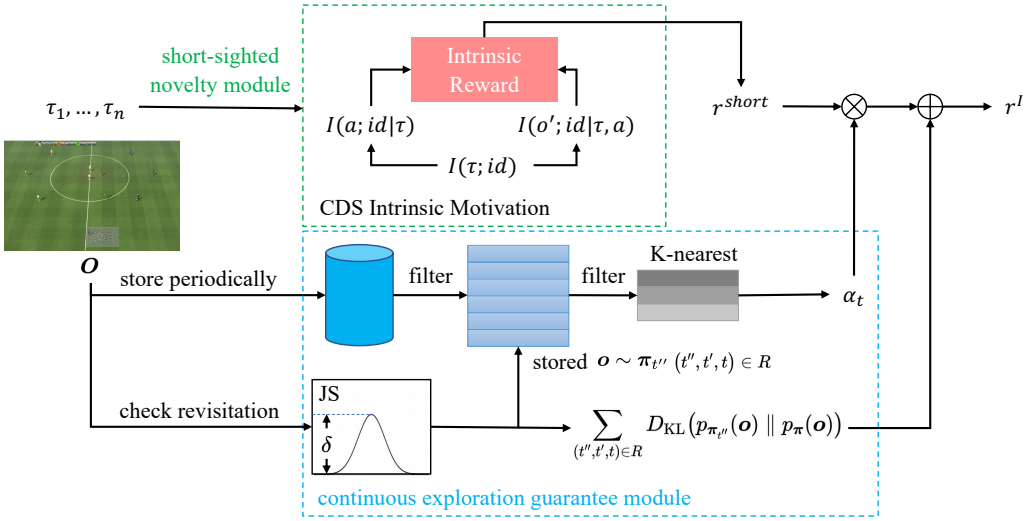


Figure 3: NRT (**N**ever **R**evisi**t**)'s intrinsic reward generator.

where $\mathcal{N}_k(\boldsymbol{o}_t)$ is the k nearest neighbors of $\boldsymbol{o}_t$ in the sampled trajectories, and ϵ is a small constant (0.001) for numeric stability. The kernel function $K(\cdot, \cdot)$ is defined as

$$K(x, y) = \frac{\epsilon}{d^2(x, y) + \epsilon}, \tag{4}$$

where d is the Euclidean distance.

For penalizing revisitation, we will guarantee trajectories related to $\boldsymbol{\pi}_{t''}$ in each revisitation tuple $(t'', t', t) \in \mathcal{R}$ are sampled to calculated α_t in Eq. 3. Besides that, we further apply the following intrinsic rewards based on all revisitation tuples:

$$r^p = \sum_{(t'',t',t)\in R} D_{\mathrm{KL}}\left(p_{\boldsymbol{\pi}_{t''}}(\boldsymbol{o}) \| p_{\boldsymbol{\pi}}(\boldsymbol{o})\right), \tag{5}$$

where $p_{\boldsymbol{\pi}}$ is the recent joint policy.

Overall, the intrinsic reward used in our paper is

$$r^i = r^{\text{short}} \cdot \texttt{clip}(1 + \bar{\alpha}, L_1, L_2) + \beta_p r^p, \tag{6}$$

where $\bar{\alpha} = \frac{\alpha - \mu_e}{\sigma_e}$ is the running average of α, `clip` means we clip the running average so that it is in the range $[L_1, L_2]$, and r_t^{short} can be based on any intrinsic motivations. In this paper, we use the intrinsic rewards introduced in CDS Li et al. (2021). β_p is a hyper-parameter tuning the weight of r^p.

4 Related Work

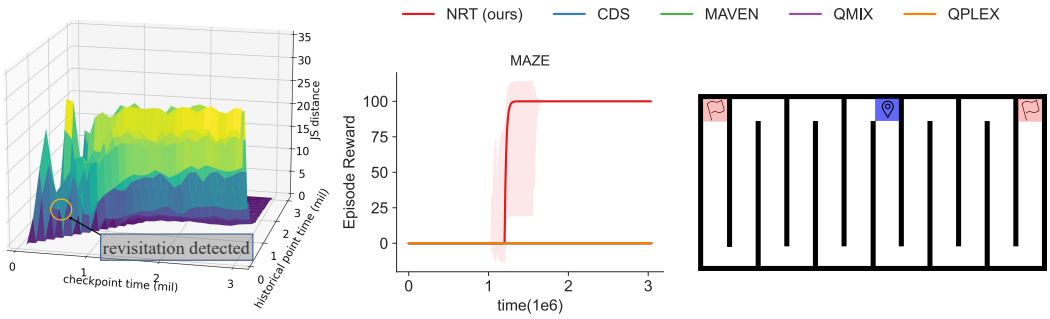
Exploration Bonuses. Exploration is one key factor that determines the efficiency of deep reinforcement learning (Houthooft et al., 2016; Trott et al., 2019; Raileanu & Rocktäschel, 2020; Zintgraf et al., 2021; Chen et al., 2021). Starting from the classic ϵ-greedy action selector, numerous advanced algorithms are proposed to enhance the efficiency of exploration. Count-based exploration methods directly use visit counts to guide an agent's behavior towards reducing uncertainty (Strehl & Littman, 2008; Bellemare et al., 2016; Martin et al., 2017; Ostrovski et al., 2017; Tang et al., 2017; Machado et al., 2020). A representative method is Random Network Distillation (RND), using prediction errors to estimate count-based rewards (Burda et al., 2018b; Osband et al., 2018; 2019; Ciosek et al., 2019). The prediction errors are also used in Pathak et al. (2017); Burda et al. (2018a); Pathak et al. (2019); Dean et al. (2020) to achieve self-organized intrinsic motivations.

Exploration in the Multi-Agent Setting. Exploration is more crucial in the multi-agent setting, because the search space grows exponentially with the number of agents. Besides exploring the environment, we also need to consider diversity across agents to achieve sophisticated coordination. In the multi-agent setting, agents' exploration can be jointly encouraged (Mahajan et al., 2019; Gupta et al., 2021; Zheng et al., 2021; Liu et al., 2021), encouraged in pairs (Wang et al., 2020b; Ndousse et al., 2021), or independently encouraged for being special from the whole group (Jiang & Lu, 2021; Li et al., 2021). Exploration can also be encouraged through the factorization of roles (Wang et al., 2020a; 2021b). In this paper, we choose CDS (Li et al., 2021) as the intrinsic motivation for exploration because it is the current state-of-the-art algorithm in many challenging benchmarks.

Forgetting. Revisitation discussed in our paper also relates to the forgetting of previous policies, which is wildly discussed in lifelong learning with the multi-task setting (Rebuffi et al., 2017; Li & Hoiem, 2017; Rolnick et al., 2019; Von Oswald et al., 2019). Some advanced algorithms have studied this topic by extending the supervised regularization (Kirkpatrick et al., 2017) or replay (Isele & Cosgun, 2018; Yan et al., 2022) paradigms, exacerbating the stability-plasticity tension. Others, on the other hand, have proposed multi-stage processes in which the agent first applies existing knowledge to the current task and then incorporates newly acquired knowledge into a shared repository (Schwarz et al., 2018; Mendez et al., 2020). However, forgetting is rarely discussed in single-task and multi-agent reinforcement learning, which also has the possibility of forgetting during exploration. The question is how to detect and forbid it automatically without any information about the stages agents are in during learning.

In this paper, we attempt to answer this question to enhance exploration efficiency. Due to the fluctuation of intrinsic motivations, agents typically consider some areas that have already been explored are worth exploring again rather than exploring further based on those that are already explored. This phenomenon heavily influences exploration efficiency, but is rarely discussed. Ecoffet et al. (2019) and Ecoffet et al. (2021) intuitively show a hypothetical example of revisitation caused by detachment in intrinsic motivations. They solve it by storing all visited states and related policies to access. Before sampling, they will first check which state to return from the whole state space, which is challenging to execute in a large continuous search space, especially in multi-agent settings. In this paper, we first show policies' repeated revisitation with experiments in the multi-agent setting and solve it with sample strategies based on fewer memories. Our approach maintains active detection and prevention of revisitation. For each revisitation, we augment the learning framework with auxiliary supporting incentives for preventing it from occurring again. We hope this paper can stimulate a rethinking of intrinsic motivations for more effective exploration in single-task multi-agent learning.

5 Didactic Example



(a) Another perspective on the 3D plot of JS distance changing

(b) Comparison against baselines

(c) Environment

Figure 4: With revisit forbidding, our approach achieves remarkable exploration efficiency.

In this section, we present a maze task (shown in Fig. 4 (c)) to demonstrate the existence of revisitation, its influence on exploration, and how our approach improves learning performance by avoiding revisitation. In this task, two agents are initialized at the blue point and are expected to simultaneously reach two red points located in different corners of a 6×12 maze. Each agent can observe its own and teammate's coordinates, and the concatenation of two agents' positions is provided as state

information. The action set of both agents includes one-step movement in four directions and an idle action for doing nothing. The horizon is set to 50 steps. If agents reach the two red points simultaneously, the episode will terminate immediately with a team reward of 100. For any other cases, the environmental reward is set to 0. The sparsity of environmental rewards enhances the difficulty of learning a successful policy in this environment. Fig. 4 (b) shows the comparison of our approach against baseline algorithms, including state-of-the-art value function factorization learning methods (QMIX (Rashid et al., 2018) and QPLEX (Wang et al., 2021a)) and multi-agent exploration algorithms (CDS (Li et al., 2021) and MAVEN (Mahajan et al., 2019)). Experimental results demonstrate the advantage of our approach in exploration, as NRT is the only algorithm that obtains the episode reward of 100.

To understand the effectiveness of our method, we store joint observation distribution estimator $\rho_{\boldsymbol{\pi}}(\boldsymbol{o})$ periodically (every 100k time steps in this experiment). Each time a distribution estimator is stored, we compare the latest joint observation distribution with the historical ones by measuring their JS distances. A small JS distance indicates that the agents are exploring a similar sub-space. The results are shown in Fig. 1. It can be observed that the JS distances for QMIX and CDS fluctuate, indicating that they are repeatedly exploring some sub-spaces. Moreover, QMIX can only cover less than 20% of the joint observation space. By contrast, our approach achieves nearly 100% coverage rate via successfully handling the issue of revisitation. Fig. 1-right shows most JS distances are large before the convergence to the strategy.

In Fig. 4 (a), we further show another perspective of the 3D plot of JS distance plot. The circle in Fig. 4 (a) highlights a time step when a revisitation is detected. At this time, our approach (1) adds a new branch to the Q-network, which is activated as described in Sec. 3.2; (2) ensures that the joint observations where the revisitation happens are sampled when calculating the intrinsic reward (Eq. 3); and (3) adds punishment for revisitation according to Eq. 5. In this way, our approach effectively prevents the reappearance of revisitation, leading to remarkable exploration efficiency.

6 EXPERIMENT

In Sec. 5, we introduce a toy maze environment to illustrate our approach's efficiency of exploration while showing the significant harm of repeat revisitation learned by current advanced multi-agent algorithms. In this section, we will compare our approach against baselines on Google Research Football (GRF) Kurach et al. (2020), which is currently the most challenging benchmark for testing cooperation between agents. We show the median and variance of every evaluation's average performance of our approach, baselines, and ablations tested with three random seeds.

6.1 ENVIRONMENT SETTING

The GRF benchmark provides numerous scenarios varying in agent numbers and difficulty. In each scenario, algorithms typically control agents on the left side and play against the built-in AI. In this paper, we use three representative scenarios to demonstrate the significance of considering revisitation, which are `Academy_3_vs_1_with_Keeper`, `Academy_Counterattack_Easy`, and `Academy_Counterattack_Hard`. Agents' initial locations for each scenario are shown in Appendix. Following the convention of Kurach et al. (2020) and Li et al. (2021), the state and each agent's observation are based on the official simple 115 representation vector, while observations are established according to relative positions. Each agent has a discrete action space of 19 dimensions, including moving and several basic actions, such as passing and shooting. Environmental rewards are only provided at the end of one episode. Agents will get $+100$ if they score, else get -1. In this paper, we test two different experimental settings about environment random seeds, which can be easily switched by one officially provided interface. They are fixed in the deterministic setting while randomly sampled from 0 to $2 * 10^9$ every episode in the stochastic setting.

6.2 PERFORMANCE

Fig. 5 demonstrates the comparison between our approach with baseline algorithms in the deterministic and stochastic settings. Here we first analyze experiment results in the deterministic setting. In `Academy_3_vs_1_with_Keeper`, QPLEX can achieve similar learning efficiency with our approach, while CDS and QMIX can also achieve a 100% winning rate. However, in more complex

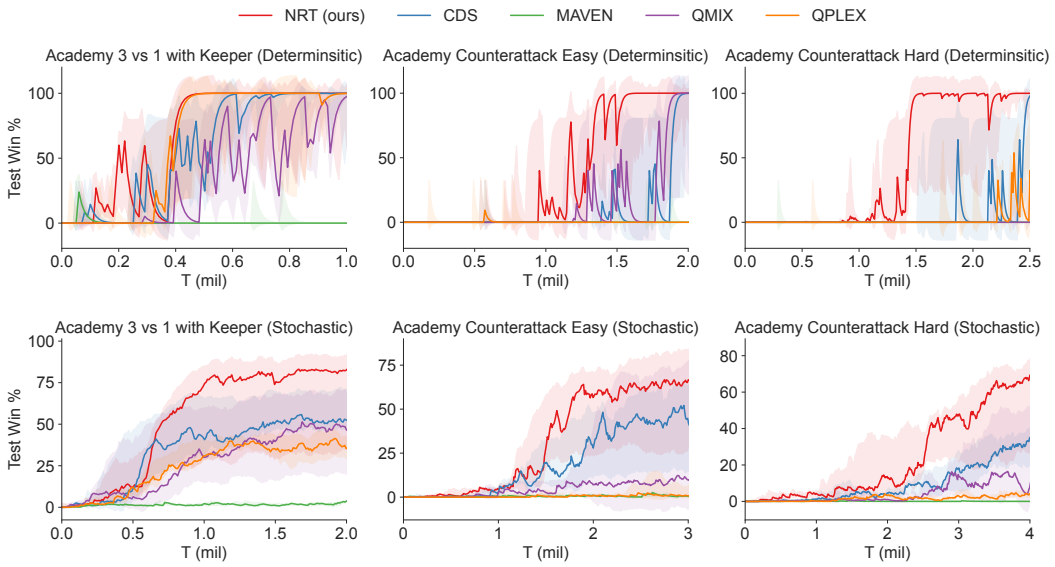


Figure 5: Comparison of our approach against baseline algorithms on GRF with deterministic environment seeds or stochastic environment seeds. We smooth all curves based on the running average. In the deterministic setting, the weight is 0.6, while it is 0.9 in the stochastic counterpart.

scenarios, `Academy_Counterattack_Hard` in particular, the outperformance of our approach is obvious, showing the advantages of considering revisitation. In `Academy_3_vs_1_with_Keeper` and `Academy_Counterattack_Easy`, CDS performs similar with QMIX. With our additional supporting incentives based on Q-function restoration, NRT achieves more efficient exploration.

Like the deterministic counterpart, our approach maintains its outperformance in the stochastic setting, as shown in Fig. 5. Because opposing players' behavior might differ across various random environment seeds, the average winning rates during evaluation are nearly impossible to be 100%. In this case, our approach still shows clear superiority in all three scenarios. In challenging scenarios, such as `Academy_Counterattack_Easy`, CDS outperforms QMIX. Following the guidance of our life-long novelty guarantee module, the performance is further raised in all three scenarios. Experiment results illustrate that our approach can tune intrinsic motivations to lead them to more sophisticated policies in complex tasks.

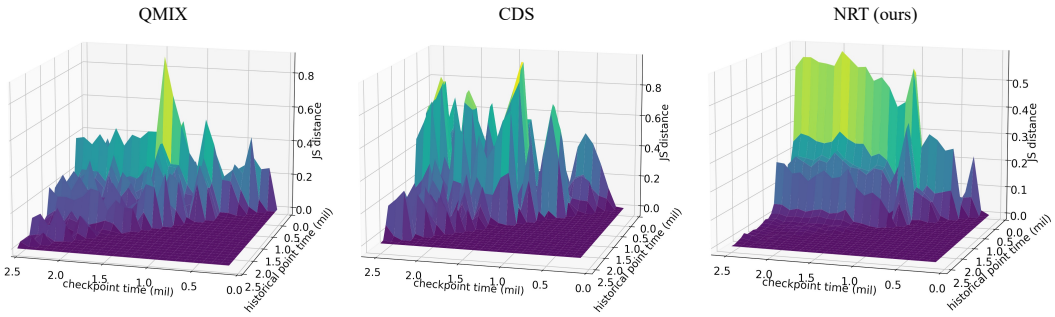


Figure 6: JS distances calculated during training in `Academy_Counterattack_Hard` under the deterministic setting

Dynamic changes of $\rho_{\boldsymbol{\pi}}(\boldsymbol{o})$ during training . We further show JS distances calculated during training in `Academy_Counterattack_Hard` under the deterministic setting. The definition of the x-y axis is the same as that introduced in Sec. 5. Similar to our toy maze environment, JS distances between historical points and checkpoints periodically fluctuate under the training of QMIX and CDS, as shown in Fig. 6 left and middle. Meanwhile, under our approach's guidance of preventing

revisitation, policies learned in complex environments achieve stable improvement, as shown in Fig. 6 right. Taking advantage from stable improved exploration, our approach achieves outstanding performance in challenging GRF scenarios.

6.3 Ablation Study

To understand the contribution of each component in the proposed NRT framework, we conduct ablation studies to test the contribution of its three main components for preventing revisitation: (1) Augmented branches based on Eq. 2. (2) Auxiliary dynamic weight of intrinsic motivations based on Eq. 3. (3) Auxiliary punishment after the detection of revisitation based on Eq. 5. We respectively ablate the three factors and name related ablation algorithms as NRT-ab-new-branch, NRT-ab-alpha, and NRT-ab-KL.

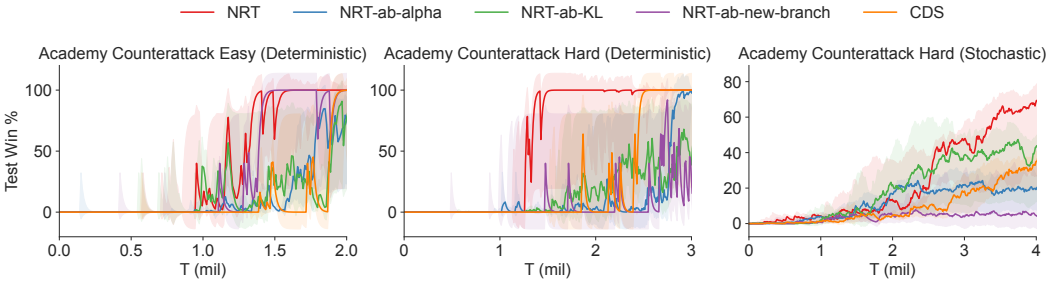


Figure 7: Ablation studies on three GRF tasks.

We compare NRT with its ablations on three representative tasks: `Academy_Counterattack_Easy` under the deterministic setting and `Academy_Counterattack_Hard` under both the deterministic and stochastic environment, as shown in Fig. 7. As for NRT-ab-new-branch, we notice an interesting phenomenon. In `Academy_Counterattack_Easy`, NRT-ab-new-branch performs similarly with NRT. However, in `Academy_Counterattack_Hard` under the deterministic setting, a more complex task, the performance of NRT-ab-new-branch is worse than NRT but still can win sometimes. In `Academy_Counterattack_Hard` under the stochastic setting, the most challenging task, the median winning rate of NRT-ab-new-branch during evaluation is almost zero. This phenomenon reveals the importance of constructing dynamically scalable representations and credit assignments according to the degree of exploration in complex environments. The performance of NRT-ab-alpha and NRT-ab-KL in three tasks illustrate that the combined effect of our auxiliary supporting incentives makes our approach outperform vanilla CDS.

7 Closing Remarks

In this paper, we investigate the intrinsic motivation methods for multi-agent exploration, and find the revisitation issue that prevents these methods from achieving efficient exploration in complex tasks. For future work, we plan to study the interplay between the number of new branches (classes) and the scale of the proposed intrinsic rewards. As intrinsic rewards can control the number of new branches, it is an important problem to determine the best choices for scaling intrinsic rewards. Moreover, we find that adding new branches may not help performance improvement on some simple tasks. This problem is also important and deserves further investigation because intuitively, more classes mean more powerful representational capacity.

References

Nolan Bard, Jakob N Foerster, Sarath Chandar, Neil Burch, Marc Lanctot, H Francis Song, Emilio Parisotto, Vincent Dumoulin, Subhodeep Moitra, Edward Hughes, et al. The hanabi challenge: A new frontier for ai research. *Artificial Intelligence*, 280:103216, 2020.

Marc Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Remi Munos. Unifying count-based exploration and intrinsic motivation. In *Advances in Neural Information Processing Systems*, pp. 1471–1479, 2016.

Yuri Burda, Harri Edwards, Deepak Pathak, Amos Storkey, Trevor Darrell, and Alexei A Efros. Large-scale study of curiosity-driven learning. *arXiv preprint arXiv:1808.04355*, 2018a.

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

Haipeng Chen, Wei Qiu, Han-Ching Ou, Bo An, and Milind Tambe. Contingency-aware influence maximization: A reinforcement learning approach. In *Uncertainty in Artificial Intelligence*, pp. 1535–1545. PMLR, 2021.

Kamil Ciosek, Vincent Fortuin, Ryota Tomioka, Katja Hofmann, and Richard Turner. Conservative uncertainty estimation by fitting prior networks. In *International Conference on Learning Representations*, 2019.

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.

Victoria Dean, Shubham Tulsiani, and Abhinav Gupta. See, hear, explore: Curiosity via audio-visual association. *Advances in Neural Information Processing Systems*, 33:14961–14972, 2020.

Adrien Ecoffet, Joost Huizinga, Joel Lehman, Kenneth O Stanley, and Jeff Clune. Go-explore: a new approach for hard-exploration problems. *arXiv preprint arXiv:1901.10995*, 2019.

Adrien Ecoffet, Joost Huizinga, Joel Lehman, Kenneth O Stanley, and Jeff Clune. First return, then explore. *Nature*, 590(7847):580–586, 2021.

Jakob Foerster, Francis Song, Edward Hughes, Neil Burch, Iain Dunning, Shimon Whiteson, Matthew Botvinick, and Michael Bowling. Bayesian action decoder for deep multi-agent reinforcement learning. In *International Conference on Machine Learning*, pp. 1942–1951, 2019.

Jakob N Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. Counterfactual multi-agent policy gradients. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Tarun Gupta, Anuj Mahajan, Bei Peng, Wendelin Böhmer, and Shimon Whiteson. Uneven: Universal value exploration for multi-agent reinforcement learning. In *International Conference on Machine Learning*, pp. 3930–3941. PMLR, 2021.

Rein Houthooft, Xi Chen, Yan Duan, John Schulman, Filip De Turck, and Pieter Abbeel. Vime: Variational information maximizing exploration. In *Advances in Neural Information Processing Systems*, pp. 1109–1117, 2016.

David Isele and Akansel Cosgun. Selective experience replay for lifelong learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Jiechuan Jiang and Zongqing Lu. The emergence of individuality. In *International Conference on Machine Learning*, pp. 4992–5001. PMLR, 2021.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114 (13):3521–3526, 2017.

Karol Kurach, Anton Raichuk, Piotr Stańczyk, Michał Zajac, Olivier Bachem, Lasse Espeholt, Carlos Riquelme, Damien Vincent, Marcin Michalski, Olivier Bousquet, et al. Google research football: A novel reinforcement learning environment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 4501–4510, 2020.

Vitaly Kurin, Maximilian Igl, Tim Rocktäschel, Wendelin Boehmer, and Shimon Whiteson. My body is a cage: the role of morphology in graph-based incompatible control. In *International Conference on Learning Representations*, 2020.

Chenghao Li, Tonghan Wang, Chengjie Wu, Qianchuan Zhao, Jun Yang, and Chongjie Zhang. Celebrating diversity in shared multi-agent reinforcement learning. *arXiv e-prints*, pp. arXiv–2106, 2021.

Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017.

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

Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, OpenAI Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Advances in Neural Information Processing Systems*, pp. 6379–6390, 2017.

Xiaoteng Ma, Yiqin Yang, Chenghao Li, Yiwen Lu, Qianchuan Zhao, and Jun Yang. Modeling the interaction between agents in cooperative multi-agent reinforcement learning. In *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems*, pp. 853–861, 2021.

Marlos C Machado, Marc G Bellemare, and Michael Bowling. Count-based exploration with the successor representation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 5125–5133, 2020.

Anuj Mahajan, Tabish Rashid, Mikayel Samvelyan, and Shimon Whiteson. Maven: Multi-agent variational exploration. In *Advances in Neural Information Processing Systems*, pp. 7611–7622, 2019.

Jarryd Martin, Suraj Narayanan Sasikumar, Tom Everitt, and Marcus Hutter. Count-based exploration in feature space for reinforcement learning. *arXiv preprint arXiv:1706.08090*, 2017.

Jorge Mendez, Boyu Wang, and Eric Eaton. Lifelong policy gradient learning of factored policies for faster training without forgetting. *Advances in Neural Information Processing Systems*, 33: 14398–14409, 2020.

Kamal K Ndousse, Douglas Eck, Sergey Levine, and Natasha Jaques. Emergent social learning via multi-agent reinforcement learning. In *International Conference on Machine Learning*, pp. 7991–8004. PMLR, 2021.

Frans A Oliehoek, Christopher Amato, et al. *A concise introduction to decentralized POMDPs*, volume 1. Springer, 2016.

Ian Osband, John Aslanides, and Albin Cassirer. Randomized prior functions for deep reinforcement learning. *Advances in Neural Information Processing Systems*, 31, 2018.

Ian Osband, Benjamin Van Roy, Daniel J Russo, Zheng Wen, et al. Deep exploration via randomized value functions. *J. Mach. Learn. Res.*, 20(124):1–62, 2019.

Georg Ostrovski, Marc G Bellemare, Aäron van den Oord, and Rémi Munos. Count-based exploration with neural density models. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 2721–2730. JMLR. org, 2017.

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

Deepak Pathak, Dhiraj Gandhi, and Abhinav Gupta. Self-supervised exploration via disagreement. In *International conference on machine learning*, pp. 5062–5071. PMLR, 2019.

Roberta Raileanu and Tim Rocktäschel. Ride: Rewarding impact-driven exploration for procedurally-generated environments. *arXiv preprint arXiv:2002.12292*, 2020.

Tabish Rashid, Mikayel Samvelyan, Christian Schroeder Witt, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning. In *International Conference on Machine Learning*, pp. 4292–4301, 2018.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 2001–2010, 2017.

David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. Experience replay for continual learning. *Advances in Neural Information Processing Systems*, 32, 2019.

Jonathan Schwarz, Wojciech Czarnecki, Jelena Luketina, Agnieszka Grabska-Barwinska, Yee Whye Teh, Razvan Pascanu, and Raia Hadsell. Progress & compress: A scalable framework for continual learning. In *International Conference on Machine Learning*, pp. 4528–4537. PMLR, 2018.

Kyunghwan Son, Daewoo Kim, Wan Ju Kang, David Earl Hostallero, and Yung Yi. Qtran: Learning to factorize with transformation for cooperative multi-agent reinforcement learning. In *International Conference on Machine Learning*, pp. 5887–5896, 2019.

Kyunghwan Son, Sungsoo Ahn, Roben Delos Reyes, Jinwoo Shin, and Yung Yi. Qtran++: Improved value transformation for cooperative multi-agent reinforcement learning, 2020.

Alexander L Strehl and Michael L Littman. An analysis of model-based interval estimation for markov decision processes. *Journal of Computer and System Sciences*, 74(8):1309–1331, 2008.

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 based on team reward. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pp. 2085–2087. International Foundation for Autonomous Agents and Multiagent Systems, 2018.

Haoran Tang, Rein Houthooft, Davis Foote, Adam Stooke, OpenAI Xi Chen, Yan Duan, John Schulman, Filip DeTurck, and Pieter Abbeel. # exploration: A study of count-based exploration for deep reinforcement learning. In *Advances in neural information processing systems*, pp. 2753–2762, 2017.

Alexander Trott, Stephan Zheng, Caiming Xiong, and Richard Socher. Keeping your distance: Solving sparse reward tasks using self-balancing shaped rewards. *Advances in Neural Information Processing Systems*, 32, 2019.

Johannes Von Oswald, Christian Henning, João Sacramento, and Benjamin F Grewe. Continual learning with hypernetworks. *arXiv preprint arXiv:1906.00695*, 2019.

Jianhao Wang, Zhizhou Ren, Terry Liu, Yang Yu, and Chongjie Zhang. Qplex: Duplex dueling multi-agent q-learning. *International Conference on Learning Representations (ICLR)*, 2021a.

Tonghan Wang, Heng Dong, Victor Lesser, and Chongjie Zhang. Roma: Multi-agent reinforcement learning with emergent roles. In *Proceedings of the 37th International Conference on Machine Learning*, 2020a.

Tonghan Wang, Jianhao Wang, Wu Yi, and Chongjie Zhang. Influence-based multi-agent exploration. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020b.

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

Qingsen Yan, Dong Gong, Yuhang Liu, Anton van den Hengel, and Javen Qinfeng Shi. Learning bayesian sparse networks with full experience replay for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 109–118, 2022.

Chao Yu, Akash Velu, Eugene Vinitsky, Yu Wang, Alexandre Bayen, and Yi Wu. The surprising effectiveness of ppo in cooperative, multi-agent games. *arXiv preprint arXiv:2103.01955*, 2021.

Chongjie Zhang and Victor Lesser. Coordinating multi-agent reinforcement learning with limited communication. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, pp. 1101–1108. International Foundation for Autonomous Agents and Multiagent Systems, 2013.

Zhi Zhang, Jiachen Yang, and Hongyuan Zha. Integrating independent and centralized multi-agent reinforcement learning for traffic signal network optimization. *arXiv preprint arXiv:1909.10651*, 2019.

Lulu Zheng, Jiarui Chen, Jianhao Wang, Jiamin He, Yujing Hu, Yingfeng Chen, Changjie Fan, Yang Gao, and Chongjie Zhang. Episodic multi-agent reinforcement learning with curiosity-driven exploration. *Advances in Neural Information Processing Systems*, 34:3757–3769, 2021.

Luisa M Zintgraf, Leo Feng, Cong Lu, Maximilian Igl, Kristian Hartikainen, Katja Hofmann, and Shimon Whiteson. Exploration in approximate hyper-state space for meta reinforcement learning. In *International Conference on Machine Learning*, pp. 12991–13001. PMLR, 2021.