PORTAL: Automatic Curricula Generation for Multiagent Reinforcement Learning

Jizhou Wu[1], **Jianye Hao**[1*], **Tianpei Yang**[2*], **Xiaotian Hao**[1], **Yan Zheng**[1], **Weixun Wang**[3], **Matthew E. Taylor**[2]
[1]College of Intelligence and Computing, Tianjin University
[2]University of Alberta and Alberta Machine Intelligence Institute
[3]Netease Fuxi AI Lab
{research5,jianye.hao,tpyang,xiaotianhao,yanzheng,wxwang}@tju.edu.cn
matthew.e.taylor@ualberta.ca

Abstract

Despite many breakthroughs in recent years, it is still hard for MultiAgent Reinforcement Learning (MARL) algorithms to directly solve complex tasks in MultiAgent Systems (MASs) from scratch. In this work, we study how to use Automatic Curriculum Learning (ACL) to reduce the number of environmental interactions required to learn a good policy. In order to solve a difficult task, ACL methods automatically select a sequence of tasks (i.e., curricula). The idea is to obtain maximum learning progress towards the final task by continuously learning on tasks that match the current capabilities of the learners. The key question is how to measure the learning progress of the learner for better curriculum selection. We propose a novel ACL framework, ***PrOgRessive mulTiagent Automatic curricuLum*** (**PORTAL**), for MASs. PORTAL selects curricula according to two criteria: 1) How difficult is a task, relative to the learners' current abilities? 2) How similar is a task, relative to the final task? By learning a shared feature space between tasks, PORTAL can characterize different tasks based on the distribution of features and select those that are similar to the final task. Also, the shared feature space can effectively facilitate policy transfer between curricula. Experimental results show that PORTAL can train agents to master extremely hard cooperative tasks, which can not be achieved with previous state-of-the-art MARL algorithms.

1 Introduction

Although reinforcement learning (RL) agents can learn sophisticated behaviors by continuously interacting with the environment, they suffer from the notorious sample inefficiency problem (Taylor and Stone 2009; Yang et al. 2020). In MultiAgent Systems (MASs), this problem is more severe since the agents need to learn under partially observable and non-stationary environments, which makes it difficult for agents to achieve cooperation and even leads to algorithmic failures (Hao et al. 2023; Yang et al. 2021; Claus and Boutilier 1998; Yang et al. 2023). One potential way to address difficult MARL problems is to use curriculum learning (CL) (Narvekar et al. 2020) to generate a sequence of tasks from easy to hard to improve the agents' learning process. The key idea is that by continuously learning tasks that match the current capabilities of the agents, we can obtain more learning progress than direct learning on the (final) target task. Based on this idea, we study how to generate the task sequence to achieve maximum learning progress toward the final task in this paper.

Current MARL CL works have many limitations. Some works (Wang et al. 2020; Long et al. 2020; Chen et al. 2021) address the challenges in large-scale MASs by starting from learning scenarios with few agents and progressively increasing the number of agents to learn the final task. These approaches require the assumption that the learning difficulty of a task increases as the number of agents increases; however, this assumption may be incorrect in complex domains. Moreover, the above works mainly focus on how to maximize the learner's learning speed on the curricula, which implicitly assumes that all tasks selected will contribute to the learning of the final task, which may be untrue in practice. Some works consider the learning relevance of a task to the final task, but they also have limitations. The approach in (Fang et al. 2021) can only work with the assumption that the tasks have the same state and action space, and thus could not be applied in complex MASs. The approaches in (da Silva and Costa 2018; Zhao and Pajarinen 2022) need to work with task *contexts*, like object type and number, goal position, etc. In this setting, searching for a suitable task as curriculum is relatively easy, since they measure the similarity of tasks by contextual variables which relies on human prior knowledge. The above reasons limit the ability of these approaches to generalize to complex domains. Please refer to the appendix for more information on related work.[1]

In this paper, we propose a novel ACL framework ***PrOgRessive mulTiagent Automatic curricuLum*** (**PORTAL**) to facilitate MARL algorithms. We study 1) how to measure the learning progress of learners for better curriculum selection and 2) how to design efficient transfer mechanisms for better curriculum transfer, two critical issues in multiagent ACL. The main contributions are:

1. We propose a novel MARL curriculum selection criterion that considers both the difficulty of tasks for learners and the relevance of tasks to the final task.

*Corresponding Author

[1]Code and appendix: https://github.com/TJU-DRL-LAB/transfer-and-multi-task-reinforcement-learning

2. To facilitate curriculum transfer, we propose a shared semantic feature space to align observations of different tasks, enabling efficient policy transfer between tasks.
3. Experimental results show that PORTAL outperforms state-of-the-art (SOTA) MARL curriculum methods and can master extremely hard (cooperative) tasks, which cannot be achieved with prior MARL algorithms.

2 Background

This section provides an introduction to the problem and background knowledge relevant to our method.

2.1 Dec-POMDPs

A Decentralized Partially Observable Markov Decision Process (Dec-POMDP) is a natural multiagent extension of Markov Decision Processes, which can be represented as a tuple with 8 elements $\langle \mathcal{N}, \mathcal{S}, \mathcal{O}, \mathcal{A}, R, T, Z, \gamma \rangle$. $\mathcal{N}$ is a set of agents with $|\mathcal{N}| = n$ controled by us. In some environments, there are n' **opponents** that are not controlled by us. We collectively refer to agents and opponents as entities, such as the number of entities $E = n + n'$. The observation $\mathcal{O} \equiv \{\mathcal{O}_1, \dots \mathcal{O}_n\}$ contains the observations of n agents. With observation function $Z : \mathcal{S} \to \mathcal{O}$, each agent i receives its own observation $o_i \in \mathcal{O}_i$ from the global state s, and then the game transits to the next state s' according to the transition function T, and rewards the agents according to the reward function R. Since the number of entities is different among environments, the observation and action dimensions between environments are all different. Note that we focus on fully cooperative settings in this paper; however, this method can also be naturally extended to other settings.

2.2 Value Decomposition Network (VDN)

VDN (Sunehag et al. 2018) is a popular MARL algorithm that uses the Centralized Training Decentralized Execution (CTDE) paradigm. Due to the global shared reward function, the key issue is how to make each agent optimize its objective function based on its contribution to the team. VDN approximates the decomposition of the joint Q value function into a summation of n individual Q functions, corresponding to n different agents, and the input of each sub-Q function is the local observation and action of the corresponding agents: $Q_{total}(\boldsymbol{o_{t+1}}, \boldsymbol{a_{t+1}}) = \sum_{i=1}^{n} Q_i(o_t^i, a_t^i)$.

3 PORTAL

We begin this section with an overview of the PORTAL framework (Section 3.1). Then, we introduce the proposed curriculum selection criterion (considering both the task relevance and the learning difficulty) and the automatic curriculum sequence generation method in Section 3.2. Finally, we describe our efficient curriculum transfer mechanism in Section 3.3, fully defining PORTAL.

3.1 MultiAgent Curriculum Learning

The multiagent curriculum learning problem is formally defined as follows: given an initial task, T_0, and a set of candidate tasks, $\mathcal{T} = \{T_m\}_{m=1}^{M}$, where T_M is the final task.

Algorithm 1: PORTAL

Initialize: policy π_θ, current task $c = T_0$, current candidate task set $\mathcal{T} = \{T_m\}_{m=1}^{M}$, τ, R_{min}, R_{max}
repeat
  // Curriculum learning
  Train π_θ on task c until it converges
  // Calculate curriculum selection criterion
  Collect data on final task $\{\langle \boldsymbol{o}_t, \boldsymbol{a}_t, r_t, \boldsymbol{o}_{t+1} \rangle\}_M$ and other candidate tasks $\{\langle \boldsymbol{o}_{t,m}, \boldsymbol{a}_{t,m}, r_{t,m}, \boldsymbol{o}_{t+1,m} \rangle\}_{m=1}^{M-1}$ using π_θ.
  for *each task* $T_m \in \mathcal{T}$ **do**
    Calculate η_m^d using $\{\langle r_{t,m} \rangle\}$
    Calculate η_m^s using $\{\langle \boldsymbol{o}_{t,m} \rangle\}$, $\{\langle \boldsymbol{o}_t \rangle\}_M$ ▷ see Eq. (1)
  // Curriculum selection
  Generate moderately difficult task set $\{T_{\hat{m}}\}_{\hat{m}=1}^{\hat{M}}$ using η_m^d ▷ see Eq. (2)
  Set $c \leftarrow T_j$, where $j = \arg\max_{\hat{m}} \{\eta_{\hat{m}}^s\}_{\hat{m}=1}^{\hat{M}}$
  Update $\mathcal{T} = \mathcal{T} \setminus \left\{ T_m \in \mathcal{T} \mid m : \eta_d^m < \eta_d^{(t)} \right\}$
until *final task* T_M *is learned*;

The goal is to learn a policy π_θ that maximizes the accumulated return $R_t = \sum_{i=0}^{\infty} \gamma^i r_{t+i}$ on the final task T_M. To achieve that, the curriculum learning algorithm starts from T_0 and selects several tasks from candidate tasks until T_M is selected. At any time, the policy π_{θ_i} is trained on one task T_i, expressed as the current task $c = T_i$, using any MARL algorithm until it converges.

We present the framework overview of PORTAL in Figure 1, which involves multiple curriculum selections and transfers during the learning process. For a given task T_i, a single-step example consists of three parts:

1. Overall process: In Figure 1.b, when we finish the previous task, select the next task from the set of candidate tasks, and then learn the newly selected task.
2. Curriculum Selection: In Figure 1.a, the policy π_{θ_i} trained on T_i, generates trajectories $\{\langle \boldsymbol{o}_{t,m}, \boldsymbol{a}_{t,m}, r_{t,m}, \boldsymbol{o}_{t+1,m} \rangle\}_{m=1}^{M-1}$ on candidate tasks and $\{\langle \boldsymbol{o}_t, \boldsymbol{a}_t, r_t, \boldsymbol{o}_{t+1} \rangle\}_M$ on the final task T_M. The task difficulty criterion η^d and task similarity criterion η^s are calculated using $\{\langle r_{t,m} \rangle\}$ and $\{\langle \boldsymbol{o}_{t,m} \rangle\}$, $\{\langle \boldsymbol{o}_t \rangle\}$ respectively.
3. Curriculum Transfer: In Figure 1.c, transfer learning allows an agent to learn the current task T_j by starting from the previous policy π_{θ_i}.

With $c = T_j$ and $\pi = \pi_{\theta_j}$, we repeat the above procedure multiple times until learning $\pi = \pi_{\theta_M}$ on task $c = T_M$, as shown in Algorithm 1.

3.2 Curriculum Selection

Many ACL methods (Matiisen et al. 2020; Wu, Zhang, and Song 2018) assume that the final task includes the elements

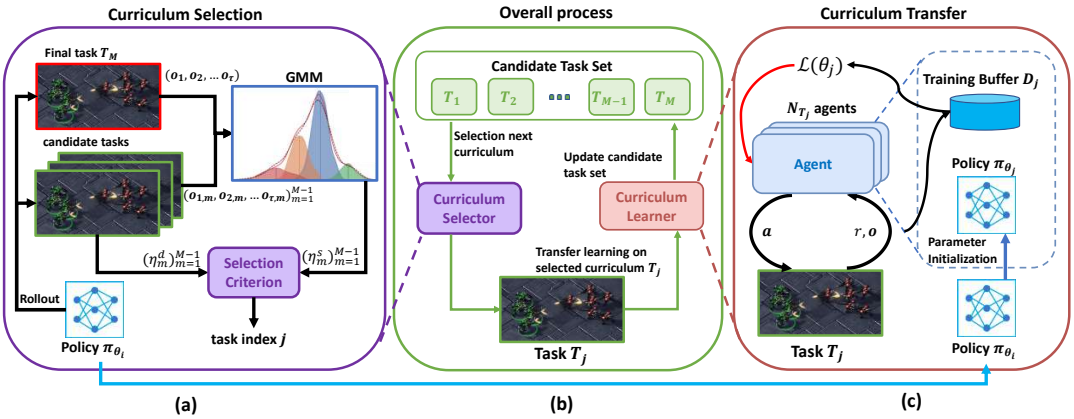


Figure 1: PORTAL Framework: (a) Curriculum Selection: at the time we finished training on the current task, we collected data on all candidate tasks and calculated the curriculum selection criterion. (b) Overall process: we select task T_j as the next task and then train the agent and update the candidate task set. (c) Curriculum Transfer: We reload previous policy π_{θ_i} as an initialization for the policy π_{θ_j} on the new task. A black arrow denotes data and a red arrow denotes gradients.

of all candidate tasks, therefore learning on any candidate task would contribute to the learning on the final task. This assumption may not hold in practice, such as when some candidate tasks are irrelevant to the final task. In this case, learning on a task with appropriate difficulty can only ensure the efficiency on the current task, but cannot guarantee the contribution to the final task. To avoid this problem, our curriculum selection is based on two criteria: difficulty relative to the current policy and learning relevance to the final task.

Task Similarity Criterion To the best of our knowledge, there is still no solution to measure the contribution of learning one task to another. We thus transform the problem of measuring task learning relevance into a more solvable one: measuring task similarity. Measuring task similarity is a substitute: the more similar a task is to the final task, the more likely it is to improve the performance on the final task.

Intuitively, the task similarity is estimated by the distribution differences of some key states explored by the pre-trained policy, which already contains information about the observation space and transition functions. Suppose we train a policy on task A and reload it to task B. If tasks B and A are very similar, then the decisions made by the policy on B will produce similar trajectories to those on A. Conversely, if A and B are not similar, then even if the policy gives the same action in the same state on A and B, different transition functions will lead to different next states, accumulating over time to produce different trajectories. We further validate this in Section 4.3.

Inspired by (Kanervisto, Kinnunen, and Hautamäki 2020), we consider using observation distributions to characterize tasks and measure the similarity between these distributions. Considering the true distribution of observations can be complex for a multiagent task, we choose to use Gaussian Mixture Models (GMMs) (Xuan, Zhang, and Chai 2001) to model complex distributions of observations. Note that the original observation spaces of different tasks may not be comparable, especially in MASs, the observation dimensions vary in tasks with different numbers of agents. We first need to unify the observation space, to be discussed in Section 3.3. For now, we base our calculations on assuming that we have a unified observation space.

To calculate the task similarity between the final task and the candidate tasks, we first use the policy trained on the previous task T_i to collect observation data. Then, each task's model parameters can be adapted with Maximum A Posteriori (MAP) (Murphy 2012) adaptation of GMM components using the data as follows:

1. Given the observation data $X = \{x_h\}_{h=1}^{H}$ from task T_i, H is the total number of samples. Each x_h can be approximated by a combination of K different Gaussian components $N(\boldsymbol{X} \mid \boldsymbol{\mu}_k, \Sigma_k)$ weighted by different weights w_k. With $p_k(x_h) = N(x_h \mid \boldsymbol{\mu}_k, \Sigma_k)$, we calculate the k_{th} Gaussian components of x_h as:
$$\Pr(k \mid \boldsymbol{x}_h) = \frac{w_k p_k(\boldsymbol{x}_h)}{\sum_{k=1}^{K} w_k p_k(\boldsymbol{x}_h)}$$
2. The expectation (E_k) corresponding to component k can be calculated using the EM algorithm to update parameters μ_k until convergence, where α_k is a hyperparameter that regulates the ratio of old to new parameters.
$$E_k(\boldsymbol{x}) = \frac{\sum_{h=1}^{H} \Pr(k \mid \boldsymbol{x}_h)\, \boldsymbol{x}_h}{\sum_{h=1}^{H} \Pr(k \mid \boldsymbol{x}_h)}$$
$$\hat{\boldsymbol{\mu}}_k = \alpha_k E_k(\boldsymbol{X}) + (1 - \alpha_k)\, \boldsymbol{\mu}_k$$
3. The distance between observation distributions of two tasks can be calculated by their GMM parameters μ_i and μ_j, which are approximated using the KL-Divergence upper-bound (Campbell, Sturim, and Reynolds 2006):
$$d_{\mathrm{KL}}(\boldsymbol{\mu}_i, \boldsymbol{\mu}_j) \leq \frac{1}{2} \sum_{k=1}^{K} w^k \left(\boldsymbol{\mu}_i^k - \boldsymbol{\mu}_j^k\right) \boldsymbol{\Sigma}^{-1} \left(\boldsymbol{\mu}_i^k - \boldsymbol{\mu}_j^k\right) \quad (1)$$

The task similarity criterion η_i^s for each task T_i is the negation of its distance to the final task T_M, defined as $-d_{\mathrm{KL}}(\boldsymbol{\mu}_i, \boldsymbol{\mu}_M)$.

Curriculum Difficulty Criterion Previous ACL studies (Florensa et al. 2018, 2017; Zhang, Abbeel, and Pinto 2020) have found that selecting intermediate tasks of moderate difficulty yields the largest learning progress. Therefore, we follow this principle to select a moderately difficult task. After training π_i on task T_i, we need to measure the difficulties of candidate tasks in $\mathcal{T}$. Many works (Narvekar et al. 2020; Visús, García, and Fernández 2021) have used policy transfer gain to evaluate task difficulty (Zhu, Lin, and Zhou 2020), e.g., Time to Threshold (TT), Jumpstart Performance (JP), and Asymptotic Performance (AP). To obtain TT and AP, we need to train the policy until it converges, which is time-consuming. Therefore, we choose to use JP as the measure of difficulty, since the original intention of curriculum learning is to reduce the training cost. For each candidate task $T_m \in \mathcal{T}$, we roll out several trajectories using π_i and calculate the average return of episodes as the task difficulty criterion η_m^d.[2] Then, we define tasks of moderate difficulty:

$$\hat{\mathcal{T}} = \left\{ T \in \mathcal{T} \mid R_{min} \leq \eta_{(T)}^d \leq R_{max} \right\} \tag{2}$$

where $\eta_{(T)}^d$ is the corresponding difficulty criterion of task T, and R_{min} and R_{max} are two threshold values.

In practice, given candidate task set $\mathcal{T} = \{T_m\}_{m=1}^M$, we select next task from $\mathcal{T}$ with task difficulty criterion η_m^d and task similarity criterion η_m^s following a two-step selecting mechanism as shown in Algorithm 1. Firstly, selecting moderately difficult tasks following the approach in 3.2, we get $\hat{\mathcal{T}} = \{T_{\hat{m}}\}_{\hat{m}=1}^{\hat{M}}$ (line 11). Then we select the one that is most similar to the final task T_M from $\hat{\mathcal{T}}$ as the next task, i.e., $\arg\max_{\hat{m}} \{\eta_{\hat{m}}^s\}_{\hat{m}=1}^{\hat{M}}$, where $\hat{m}$ is the index of task in $\hat{\mathcal{T}}$ (line 12). We select the moderately difficult tasks first because tasks similar to the final task could be very difficult, learning on these tasks will be time-consuming, which impairs the learning efficiency. After task selection, we also need to remove tasks that are easier than T_j in the candidate task set (used in the next curriculum selection step) $\mathcal{T}$, which is $\mathcal{T} = \mathcal{T} \setminus \left\{ T_m \in \mathcal{T} \mid m : \eta_d^m < \eta_d^{(t)} \right\}$ (line 13).

3.3 Curriculum Transfer

A common transfer mechanism is to directly initialize an agent with the policy trained on the previous task (i.e., value-function transfer (Taylor and Stone 2009)) Unfortunately, this could be non-trivial due to two reasons: 1) the observation space of different tasks has different input dimensions so that the model cannot be reloaded directly. 2) More importantly, the semantics of original observations are different among tasks, which hinders better transfer performance. So, it is necessary to map the observations of different tasks into a shared semantic feature space first to achieve more effective transfer learning.

[2]The rewards of all tasks are normalized to the same scale so that a policy's return can estimate the difficulty of a task.

Next, we introduce a property prevalent in MASs — Sparse Interaction (SI) to show that the observations in different tasks have similar semantics. In a Dec-POMDP, each agent receives its observation, which contains nearby entities' information with different importance. For example, in the predator-prey environment (Yang et al. 2018), each predator needs to cooperate with its neighbor predators without considering other predators of larger distance. Also, at one time, each agent only interacts with some of the entities and the interactions do not happen all the time. The example indicates that agents should mainly consider the information most relevant to themselves while ignoring the irrelevant information when making decisions. Thus, we propose a **Entity Relation Feature Network (ERFN)** to extract semantic features shared between tasks considering the SI property to facilitate curriculum transfer. Another benefit of the shared feature space is that the features can be used to calculate the task similarity criterion, which solves the problem of inconsistent observation dimensions between tasks.

Entity Relation Feature Network In MASs, the observation of an agent i at time step t contains the information about itself ($o_t^{i,0}$) and E entities, $o_t^i = \{o_t^{i,0}, o_t^{i,1}, \ldots, o_t^{i,E}\}$, for example, entity observations may include relative distance, relative coordinates, and so on. In our setting, entities are enemies and allies. To extract informative features from the observation, we design a feature network that can generalize across tasks with different entity numbers. Specifically, we split the observation and use a shared network ϕ to extract feature $\phi(o_t^{i,e})$, splitting observations by entities is a common approach in MAS, as seen in works like PIC (Liu, Yeh, and Schwing 2019) and DYMA (Wang et al. 2020), where the features are aggregated using a summation function, so the extracted observation feature of an agent at time step t can be represented as: $f(o_t^i) = \sum_{e=0}^{E}(\phi(o_t^{i,e}))$. However, this may hinder curriculum transfer because the *sum* operation is sensitive to the number of agents, i.e., the value scale of the aggregated features will be correlated with the number of entities in a task. Furthermore, when we transfer between tasks, only those observations with the same number of agents would be recognized, potentially discarding useful information. Based on the property we discussed before, we learn semantic features where the most relevant information is the most important, thus weakening the effect of the number of entities without losing important information. In practice, we propose a feature aggregation function based on relations between an agent and other entities in its observation. In addition to the feature network ϕ, our model contains an additional relation network ψ, which is used to calculate the relational weight. The final feature with relational weight can be calculated as:

$$f(o_t^i) = \sum_{e=0}^{E} \phi(o_t^{i,e}) \psi([o_t^{i,e}, o_t^{i,0}])$$

The input of ψ is the concatenation of $o_t^{i,e}$ and $o_t^{i,0}$ since we need to use an agent's own information and entity information to calculate the relation weight. The weighted

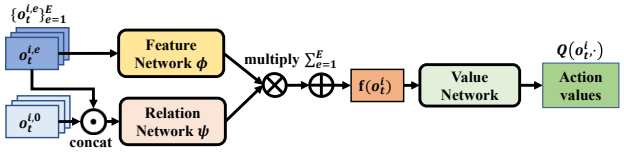


Figure 2: The input observation is divided and the feature and relation weight of each entity are calculated, respectively, using a shared feature network and a relation network.

summed feature will then be fed into the value network to output the Q values, shown in Figure 2.

In practice, we use HPN-VDN (Hao et al. 2022) as our backbone model (detailed model structure and hyperparameter settings are in Appendix B), but PORTAL is not restricted to value-based methods. The policy is trained end-to-end to minimize the loss as follows:

$$\mathcal{L}(\theta) = \left[\left(y - Q_{total}(\boldsymbol{o_t}, \boldsymbol{a_t}; \theta)\right)^2\right] \tag{3}$$

where $\boldsymbol{o}, \boldsymbol{a}$ are joint observations and joint actions, and $y = r_t + \gamma \max_{\boldsymbol{a_{t+1}}} Q_{total}(\boldsymbol{o_{t+1}}, \boldsymbol{a_{t+1}}; \theta^-)$ is the target, where θ^- parameterizes the target value network. The global Q value is calculated by Eq. 4, where n is the number of agents.

$$Q_{total}(\boldsymbol{o_t}, \boldsymbol{a_t}) = \sum_{i=1}^{n} Q_i(o_t^i, a_t^i) \tag{4}$$

We share the parameters among all agents regarding the feature and relation networks (Figure 2). The input length of the feature network, i.e., $\text{len}(o_t^{i,e})$ remains consistent across tasks, as well as $\text{len}([o_t^{i,e}, o_t^{i,0}])$, ensuring transferability.

4 Experiments

In this section, we study the following research questions (RQs) via comprehensive experiments.

RQ1. (Performance) Does PORTAL improve the convergence rate and asymptotic performance over MARL non-curriculum baselines and MARL curriculum baselines?

RQ2. (Necessity of task similarity criterion) Does the introduction of the task similarity criterion lead to better curriculum selection than using only the task difficulty criterion? Can the task similarity criterion serve as a valid proxy to measure the facilitation of tasks to the final task?

RQ3. (Feature aggregation functions) How do aggregation methods perform in training and curriculum transfer?

4.1 Experiment Setups

We focus on the widely used MARL benchmark Starcraft MultiAgent Challenge (SMAC) (Samvelyan et al. 2019). More details about the environment's observations, actions, etc. can be found in Appendix A.1. Since the previous SOTA algorithms (Hu et al. 2021; Hao et al. 2022) have achieved an almost 100% win rate on most original scenarios in SMAC, we constructed more difficult scenarios to test our method. Environment configurations are detailed in Appendix A.2.

The experiments are carried out on three different task series, they are Marines:*5m (initial), 5m_vs_6m, 6m, 7m, 8m_vs_9m, 8m_vs_10m, 10m, 15m, 20m, 25m, 7m_vs_9m (final)*, Stalkers & Zealots (S & Z):*2s3z (initial), 2s4z, 2s5z, 3s4z, 2s3z_vs_2s4z, 2s3z_vs_2s5z, 1s4z_vs_4s1z, 1s4z_vs_5s1z, 3s5z, 1s4z_vs_6s1z, 3s5z_vs_3s6z, 3s5z_vs_3s7z, 3s5z_vs_8s2z, 3s5z_vs_4s6z, 3s5z_vs_4s7z, 3s5z_vs_4s8z (final)* and Medivac & Marauders & Marines (MMM):*MMM0 (initial), MMM1, MMM2, MMM3, MMM4, MMM5, MMM6, MMM7, MMM8, MMM9, MMM10 (final)*. For symmetric tasks, we do not show the enemy configuration in name, like *5m_vs_5m* is abbreviated as *5m*. The final tasks are all extremely hard. We consider the diversity of tasks when constructing the task set, which is closer to the reality that not all tasks in the task set are relevant to the final task's learning objectives. For example, 5m_vs_6m (asymmetric), 25m (large scale, symmetric), and 8m_vs_10m (extremely asymmetric) for homogeneous scenarios. For heterogeneous scenarios like S & Z and MMM, tasks with different unit types are involved, resulting in different winning strategies between tasks even with the same number of agents.

4.2 Comparison With Related Baselines (RQ1)

Here, we compare with four baselines: (1)the SOTA MARL non-curriculum algorithm HPN-QMIX (Hao et al. 2022) (a stronger baseline than our backbone model HPN-VDN) and three MARL curriculum baselines:(2)DYMA (Wang et al. 2020), an algorithm that generates task sequences in increments of the number of agents. Two modified versions of PORTAL (3)ERFN-OO and (4)ERFN-MAACL, since the network of OO (da Silva and Costa 2018) and MAACL (Zhang et al. 2022) cannot be directly applied in our task scenarios, we only use their task selection criterion and other parts are the same as PORTAL.

Although our goal is to achieve better performance on the final task, we also provide results on intermediate tasks to illustrate that PORTAL could improve the learning on all curricula in sequence, since good performance on intermediate tasks means there is more useful knowledge that could be transferred to the final task. For all algorithms, the initial and final tasks are the same, but they may generate different intermediate curriculum sequences as shown in Table **??**(More sequence generation details are in Appendix B), we only plot the learning curves of tasks that are same as PORTAL's tasks in sequence (the learning curves of all tasks of ERFN-OO and ERFN-MAACL are in Appendix C). Figure 3 shows the test win rate on the curriculum sequence generated by PORTAL for all three series. We can see that PORTAL achieves the best performance over other baselines on intermediate tasks and the final task. Although some baselines have curriculum mechanisms, they did not select tasks that are most relevant to the final task. In practice, we control that all algorithms use the same number of training steps on the curriculum sequence to ensure fairness For task similarity, PORTAL spends 10,000 steps on each task to collect enough data to make the task selection stable. This is a small cost compared to the millions of steps required for training.

Algorithm	*Marine(initial task: 5m)*	*S & Z*(initial task: 2s3z)	*MMM*(initial task: MMM0)
PORTAL	5m_vs_6m,8m_vs_10m,7m_vs_9m	3s5z_vs_3s6z,3s5z_vs_4s7z,3s5z_vs_4s8z	MMM4,MMM7,MMM10
DYMA	5m_vs_6m,7m,7m_vs_9m	3s5z_vs_3s6z,3s5z_vs_4s7z,3s5z_vs_4s8z	MMM4,MMM7,MMM10
OO	5m, 7m_vs_9m	1s4z_vs_4s1z,3s5z_vs_4s6z,3s5z_vs_4s8z	MMM8,MMM4,MMM10
MAACL	20m, 8m_vs_10m, 7m_vs_9m	1s4z_vs_5s1z,3s5z_vs_8s2z,3s5z_vs_4s8z	MMM4,MMM8,MMM10

Table 1: The curriculum sequences generated by each algorithm.

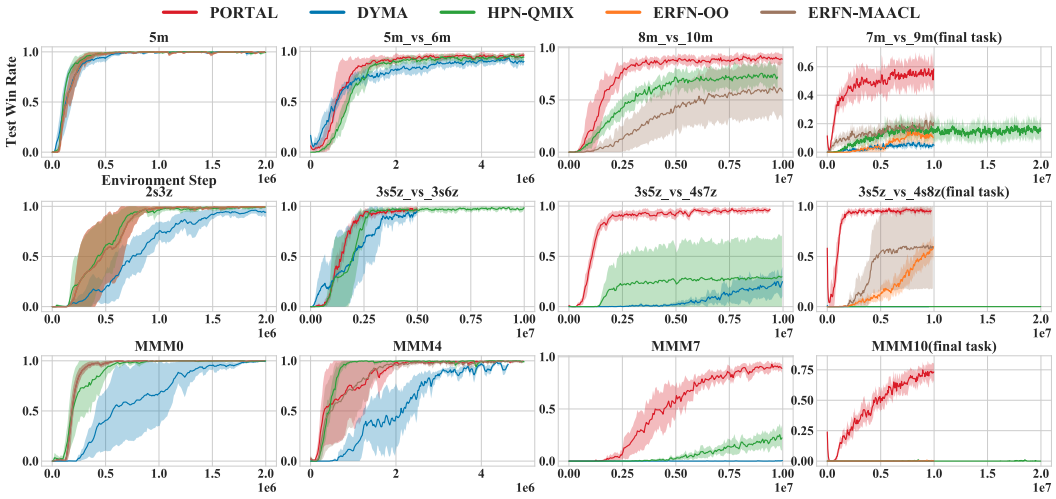


Figure 3: The three rows show the result of Marines, S & Z, MMM respectively. In each row, graphs are arranged from left to right according to the curriculum sequence, and the policy starts learning on the next task after the previous task is completed. The learning curve uses the environment steps as the x-axis and the test win rate as the y-axis. The darker lines are the means and the lighter areas are the variances, using 95% confidence intervals. Each curve shows the performance over 3 random seeds.

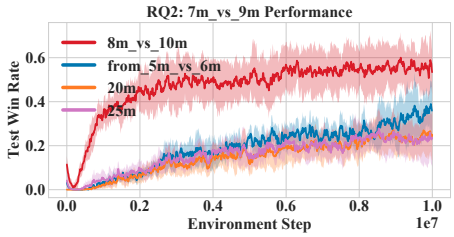


Figure 4: Performance comparison of different task sequences: *5m, 5m_vs_6m, x, 7m_vs_9m* where x is one of the task in *5m_vs_6m, 8m_vs_10m, 20m, 25m*.

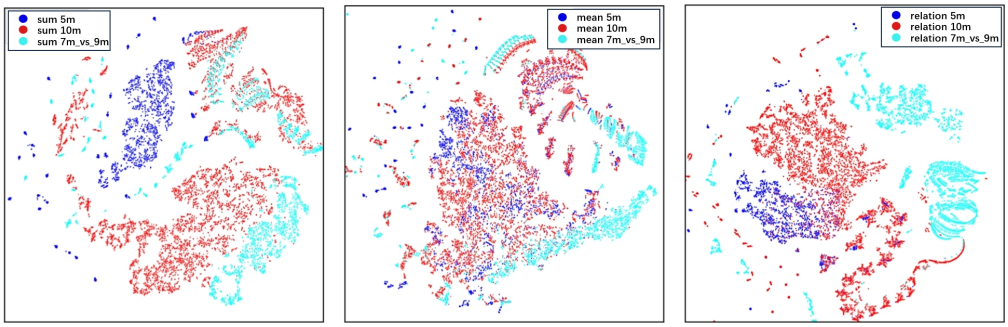
Method	**8m_vs_10m**	**7m_vs_9m**
Sum train	0.8540(±0.0814)	0.2255(±0.0553)
Mean train	0.7869(±0.0742)	0.1532(±0.0434)
Relation train	0.8455(±0.0689)	0.2186(±0.0592)
Sum transfer	0.8422(±0.0811)	0.2182(±0.0702)
Mean transfer	0.8207(±0.1239)	0.1726(±0.0504)
Relation transfer	**0.9074(±0.0736)**	**0.5587(±0.0953)**

Table 2: Performance comparison of different aggregation functions on *8m_vs_10m* and *7m_vs_9m* .

4.3 Necessity Of Task Similarity Measure (RQ2)

In RQ1, ERFN-MAACL can be seen as a baseline that only measures the difficulty of the task, the results have shown that using the task difficulty criterion alone is not good enough. Here, we carried out more analysis to evaluate whether adding our task similarity criterion leads to better curriculum selection. For example, consider the curriculum selection process for marines, after the policy has learned on *5m_vs_6m*, the tasks with moderate difficulty are *8m_vs_10m, 20m, 25m*. Without considering the task similarity, *20m* or *25m* may be selected as the next task instead of *8m_vs_10m*. We investigate the influence of selecting different intermediate tasks on the performance of *7m_vs_9m*. In addition to *20m, 25m*, we also add *5m_vs_6m* to compare with.

The results are shown in Figure 4 (results for S & Z and MMM are in Appendix C). We can see that *8m_vs_10m* is the best intermediate task over all other tasks. Intuitively, *7m_vs_9m* and *8m_vs_10m* are similar since both tasks have fewer agents than *20m, 25m* and the difference between the number of enemies and allies is the largest (i.e., two). In this way, reloading the policy learned on *8m_vs_10m* provides more benefits for the final task, thus achieving higher performance.



(a) Sum (b) Mean (c) Relation

Figure 5: The visualization result of different aggregation functions:*sum*, *mean*, *relation* on tasks:*5m, 7m_vs_9m, 10m.*

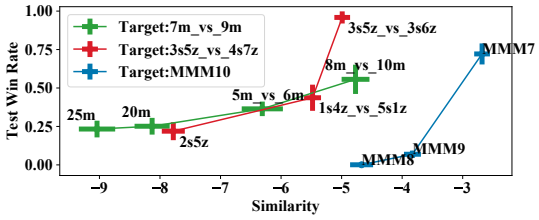


Figure 6: The influence of the similarity between each candidate task and 7m_vs_9m on the performance of transferring to 7m_vs_9m with standard deviation error bars.

We further validate our hypothesis that task similarity is an appropriate substitute for task relevance. Figure 6 shows the influence of the similarity between each candidate task and the final task on the performance of transferring to the final task. We can see that the performance on the final task increases with increasing similarity between the candidate task and the final task. This positive correlation indicates that our task similarity is a valid substitute for measuring the learning contribution of one task to the final task.

4.4 Feature Aggregation Function (RQ3)

This section highlights experiments that show feature aggregation matters in curriculum transfer. We evaluate the three aggregation functions *sum, mean, relation (ours)* (as mentioned in Section 3.3) from two perspectives: learning from scratch (train) and curriculum learning (transfer). Comparison experiments are carried out on two tasks *8m_vs_10m* and *7m_vs_9m*. [3] The convergence test win rate of the methods is shown in Table **??**. As we can see from the train results, *relation* has comparable performance with *sum* and they both outperform *mean*. This is due to the weak representational ability of *mean*, which completely ignores information about the entity number. Although *relation* weakens the impact of entity number, it still has good representation capability. Then we compare transfer capabilities. We can see that *relation* significantly outperforms the other two methods. From the experiment results, we can see that *relation* is the best aggregation function among the three because it characterizes information about entities that are highly relevant and learns a shared semantic feature space between tasks for better transfer learning.

To further illustrate that *relation* learns a better semantic feature space than the other two methods, we train a policy on 5m and reload it on *5m, 7m_vs_9m, 10m* to collect observation features. We use t-SNE (Van der Maaten and Hinton 2008) to visualize the result in Figure 5. We can see that *relation* separates features of *7m_vs_9m* and *5m, 10m*, since *7m_vs_9m* is an asymmetric task with different semantics. While *5m* and *10m* have many overlaps, this is consistent with our claim: *relation* weakens the influence of entity number on the features of different tasks. In contrast, the features from *sum* are more scattered and it can not distinguish between *7m_vs_9m* and *10m*, and *mean* completely mixes the features of different tasks.

5 Conclusion and Future Work

In this paper, we propose a novel MARL ACL framework to solve tasks that are hard to learn from scratch using existing MARL algorithms. For automatic curriculum selection, we study how to measure the learning progress of the agents and propose a curriculum selection criterion from two perspectives: learning difficulty and learning relevance. In practice, we calculate observation distributions to measure task similarity, which serves as a proxy for learning relevance. We also learn a semantic feature space shared across tasks to facilitate policy transfer between curricula. With the automatic curriculum selection and transfer mechanism, our approach significantly outperforms existing MARL algorithms.

In the course of the study, we also found that there is still much room for improvement in our approach. Our automated curriculum learning method is still mainly embodied in automatic curriculum selection and thus requires a given set of tasks. A subsequent improvement is to extend the method so that it can be applied to situations where there is no given set of tasks, such that the method needs to further enhance automation by generating tasks on its own.

[3] For curriculum learning, we use the curriculum sequence generated by PORTAL: {*5m_vs_6m, 8m_vs_10m, 7m_vs_9m*}.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Grant No. 92370132, 62106172), the National Key R&D Program of China (Grant No. 2022ZD0116402). Part of this work has taken place in the Intelligent Robot Learning (IRL) Lab at the University of Alberta, which is supported in part by research grants from the Alberta Machine Intelligence Institute (Amii); a Canada CIFAR AI Chair, Amii; Compute Canada; Huawei; Mitacs; and NSERC.

References

Campbell, W. M.; Sturim, D. E.; and Reynolds, D. A. 2006. Support vector machines using GMM supervectors for speaker verification. *IEEE Signal Process. Lett.*, 13(5): 308–311.

Chen, J.; Zhang, Y.; Xu, Y.; Ma, H.; Yang, H.; Song, J.; Wang, Y.; and Wu, Y. 2021. Variational Automatic Curriculum Learning for Sparse-Reward Cooperative Multi-Agent Problems. In *Advances in Neural Information Processing Systems 34*, 9681–9693.

Claus, C.; and Boutilier, C. 1998. The Dynamics of Reinforcement Learning in Cooperative Multiagent Systems. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence and Tenth Innovative Applications of Artificial Intelligence Conference*, 746–752. AAAI Press / The MIT Press.

da Silva, F. L.; and Costa, A. H. R. 2018. Object-Oriented Curriculum Generation for Reinforcement Learning. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2018, Stockholm, Sweden, July 10-15, 2018*, 1026–1034. International Foundation for Autonomous Agents and Multiagent Systems Richland, SC, USA / ACM.

Fang, K.; Zhu, Y.; Savarese, S.; and Fei-Fei, L. 2021. Adaptive Procedural Task Generation for Hard-Exploration Problems. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Florensa, C.; Held, D.; Geng, X.; and Abbeel, P. 2018. Automatic Goal Generation for Reinforcement Learning Agents. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, 1514–1523. PMLR.

Florensa, C.; Held, D.; Wulfmeier, M.; Zhang, M.; and Abbeel, P. 2017. Reverse Curriculum Generation for Reinforcement Learning. In *Proceedings of The 1st Annual Conference on Robot Learning*, volume 78 of *Proceedings of Machine Learning Research*, 482–495. PMLR.

Hao, J.; Hao, X.; Mao, H.; Wang, W.; Yang, Y.; Li, D.; Zheng, Y.; and Wang, Z. 2022. Boosting Multiagent Reinforcement Learning via Permutation Invariant and Permutation Equivariant Networks. In *The Eleventh International Conference on Learning Representations*.

Hao, J.; Yang, T.; Tang, H.; Bai, C.; Liu, J.; Meng, Z.; Liu, P.; and Wang, Z. 2023. Exploration in deep reinforcement learning: From single-agent to multiagent domain. *IEEE Transactions on Neural Networks and Learning Systems*.

Hu, J.; Wu, H.; Harding, S. A.; Jiang, S.; and Liao, S. 2021. RIIT: Rethinking the Importance of Implementation Tricks in Multi-Agent Reinforcement Learning. *CoRR*, abs/2102.03479.

Kanervisto, A.; Kinnunen, T.; and Hautamäki, V. 2020. General Characterization of Agents by States they Visit. *arXiv preprint arXiv:2012.01244*.

Liu, I.; Yeh, R. A.; and Schwing, A. G. 2019. PIC: Permutation Invariant Critic for Multi-Agent Deep Reinforcement Learning. In *Proceedings of The 3rd Annual Conference on Robot Learning*, volume 100 of *Proceedings of Machine Learning Research*, 590–602. PMLR.

Long, Q.; Zhou, Z.; Gupta, A.; Fang, F.; Wu, Y.; and Wang, X. 2020. Evolutionary Population Curriculum for Scaling Multi-Agent Reinforcement Learning. In *Proceedings of The 8th International Conference on Learning Representations, ICLR 2020*. OpenReview.net.

Matiisen, T.; Oliver, A.; Cohen, T.; and Schulman, J. 2020. Teacher-Student Curriculum Learning. *IEEE Trans. Neural Networks Learn. Syst.*, 31(9): 3732–3740.

Murphy, K. P. 2012. *Machine learning: a probabilistic perspective*. MIT press.

Narvekar, S.; Peng, B.; Leonetti, M.; Sinapov, J.; Taylor, M. E.; and Stone, P. 2020. Curriculum Learning for Reinforcement Learning Domains: A Framework and Survey. *CoRR*, abs/2003.04960.

Samvelyan, M.; Rashid, T.; de Witt, C. S.; Farquhar, G.; Nardelli, N.; Rudner, T. G. J.; Hung, C.; Torr, P. H. S.; Foerster, J. N.; and Whiteson, S. 2019. The StarCraft Multi-Agent Challenge. *CoRR*, abs/1902.04043.

Sunehag, P.; Lever, G.; Gruslys, A.; Czarnecki, W. M.; Zambaldi, V. F.; Jaderberg, M.; Lanctot, M.; Sonnerat, N.; Leibo, J. Z.; Tuyls, K.; and Graepel, T. 2018. 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*, 2085–2087. International Foundation for Autonomous Agents and Multiagent Systems Richland, SC, USA / ACM.

Taylor, M. E.; and Stone, P. 2009. Transfer Learning for Reinforcement Learning Domains: A Survey. *J. Mach. Learn. Res.*, 10: 1633–1685.

Van der Maaten, L.; and Hinton, G. 2008. Visualizing data using t-SNE. *Journal of machine learning research*, 9(11).

Visús, Á.; García, J.; and Fernández, F. 2021. A Taxonomy of Similarity Metrics for Markov Decision Processes. *CoRR*, abs/2103.04706.

Wang, W.; Yang, T.; Liu, Y.; Hao, J.; Hao, X.; Hu, Y.; Chen, Y.; Fan, C.; and Gao, Y. 2020. From Few to More: Large-Scale Dynamic Multiagent Curriculum Learning. In *Proceedings of The Thirty-Fourth AAAI Conference on Artificial Intelligence*, 7293–7300. AAAI Press.

Wu, Y.; Zhang, W.; and Song, K. 2018. Master-Slave Curriculum Design for Reinforcement Learning. In *Proceedings*

of the Twenty-Seventh International Joint Conference on Artificial Intelligence, 1523–1529. ijcai.org.

Xuan, G.; Zhang, W.; and Chai, P. 2001. EM algorithms of Gaussian mixture model and hidden Markov model. In *Proceedings 2001 international conference on image processing (Cat. No. 01CH37205)*, volume 1, 145–148. IEEE.

Yang, T.; Hao, J.; Meng, Z.; Zhang, Z.; Hu, Y.; Chen, Y.; Fan, C.; Wang, W.; Liu, W.; Wang, Z.; and Peng, J. 2020. Efficient Deep Reinforcement Learning via Adaptive Policy Transfer. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*.

Yang, T.; Wang, W.; Hao, J.; Taylor, M. E.; Liu, Y.; Hao, X.; Hu, Y.; Chen, Y.; Fan, C.; Ren, C.; et al. 2023. ASN: action semantics network for multiagent reinforcement learning. *Autonomous Agents and Multi-Agent Systems*, 37(2).

Yang, T.; Wang, W.; Tang, H.; Hao, J.; Meng, Z.; Mao, H.; Li, D.; Liu, W.; Chen, Y.; Hu, Y.; Fan, C.; and Zhang, C. 2021. An Efficient Transfer Learning Framework for Multiagent Reinforcement Learning. In *Advances in Neural Information Processing Systems 34*.

Yang, Y.; Yu, L.; Bai, Y.; Wen, Y.; Zhang, W.; and Wang, J. 2018. A Study of AI Population Dynamics with Million-agent Reinforcement Learning. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, 2133–2135.

Zhang, T.; Liu, Z.; Pu, Z.; and Yi, J. 2022. Automatic Curriculum Learning for Large-Scale Cooperative Multiagent Systems. *IEEE Transactions on Emerging Topics in Computational Intelligence*.

Zhang, Y.; Abbeel, P.; and Pinto, L. 2020. Automatic Curriculum Learning through Value Disagreement. In *Advances in Neural Information Processing Systems 33*.

Zhao, W.; and Pajarinen, J. 2022. Self-Paced Multi-Agent Reinforcement Learning. *CoRR*, abs/2205.10016.

Zhu, Z.; Lin, K.; and Zhou, J. 2020. Transfer Learning in Deep Reinforcement Learning: A Survey. *CoRR*, abs/2009.07888.