Machine Intelligence Research
www.mi-research.net

20(2), April 2023, 233-248
DOI: 10.1007/s11633-022-1383-7

Offline Pre-trained Multi-agent Decision Transformer

Linghui Meng[1,2*] Muning Wen[3*] Chenyang Le[3] Xiyun Li[1,4]
Dengpeng Xing[1,2] Weinan Zhang[3] Ying Wen[3] Haifeng Zhang[1,2]
Jun Wang[6] Yaodong Yang[5] Bo Xu[1,2]

[1] Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China

[2] School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China

[3] Shanghai Jiao Tong University, Shanghai 200240, China

[4] School of Future Technology, University of Chinese Academy of Sciences, Beijing 100049, China

[5] Institute for AI, Peking University, Beijing 100871, China

[6] Department of Computer Science, University College London, London WC1E 6BT, UK

Abstract: Offline reinforcement learning leverages previously collected offline datasets to learn optimal policies with no necessity to access the real environment. Such a paradigm is also desirable for multi-agent reinforcement learning (MARL) tasks, given the combinatorially increased interactions among agents and with the environment. However, in MARL, the paradigm of offline pre-training with online fine-tuning has not been studied, nor even datasets or benchmarks for offline MARL research are available. In this paper, we facilitate the research by providing large-scale datasets and using them to examine the usage of the decision transformer in the context of MARL. We investigate the generalization of MARL offline pre-training in the following three aspects: 1) between single agents and multiple agents, 2) from offline pretraining to online fine tuning, and 3) to that of multiple downstream tasks with few-shot and zero-shot capabilities. We start by introducing the first offline MARL dataset with diverse quality levels based on the StarCraftII environment, and then propose the novel architecture of multi-agent decision transformer (MADT) for effective offline learning. MADT leverages the transformer′s modelling ability for sequence modelling and integrates it seamlessly with both offline and online MARL tasks. A significant benefit of MADT is that it learns generalizable policies that can transfer between different types of agents under different task scenarios. On the StarCraft II offline dataset, MADT outperforms the state-of-the-art offline reinforcement learning (RL) baselines, including BCQ and CQL. When applied to online tasks, the pre-trained MADT significantly improves sample efficiency and enjoys strong performance in both few-short and zero-shot cases. To the best of our knowledge, this is the first work that studies and demonstrates the effectiveness of offline pre-trained models in terms of sample efficiency and generalizability enhancements for MARL.

Keywords: Pre-training model, multi-agent reinforcement learning (MARL), decision making, transformer, offline reinforcement learning.

Citation: L. Meng, M. Wen, C. Le, X. Li, D. Xing, W. Zhang, Y. Wen, H. Zhang, J. Wang, Y. Yang, B. Xu. Offline pre-trained multi-agent decision transformer. *Machine Intelligence Research*, vol.20, no.2, pp.233–248, 2023. http://doi.org/10.1007/s11633-022-1383-7

1 Introduction

Multi-agent reinforcement learning (MARL) algorithms[1] play an essential role in solving complex decision-making tasks by learning from the interaction data between computerised agents and (simulated) physical environments. It has been typically applied to self-driving[2–4], order dispatching[5, 6], modelling population dynamics[7], and gaming AIs[8, 9]. However, the scheme of learning policy from experience requires the algorithms with high computational complexity[10] and sample efficiency due to the limited computing resources and high cost resulting from the data collection[11–14]. Furthermore, even in domains where the online environment is feasible, we might still prefer to utilize previously-collected data instead; for example, if the domain′s complexity requires large datasets for effective generalization. In addition, a policy trained on one scenario usually cannot perform well on another even under the same task. Therefore, a universal policy is critical for saving the training time of general reinforcement learning (RL) tasks.

Notably, the recent advance of supervised learning has shown that the effectiveness of learning methods can be maximized when they are provided with very large modelling capacity, trained on very large and diverse datasets[15–17]. The surprising effectiveness of large, generic models supplied with large amounts of training data, such as GPT-3[18], spurs the community to search for ways to scale up thus boosting the performance of RL models. Towards this end, Decision transformer[19] is one

Research Article
Special Issue on Large-scale Pre-training: Data, Models, and Fine-tuning
Manuscript received July 6, 2022; accepted October 18, 2022
Recommended by Associate Editor Zhi-Yuan Liu
*These authors contribute equally to this work

of the first models that verifies the possibility of solving conventional (offline) RL problems by generative trajectory modelling, i.e., modelling the joint distribution of the sequence of states, actions, and rewards without temporal difference learning.

The technique of transforming decision-making problems into sequence modelling problems has opened a new gate for solving RL tasks. Crucially, this activates a novel pathway toward training RL systems on diverse datasets[20–22] in much the same manner as in supervised learning, which is often instantiated by offline RL techniques[23]. Offline RL methods have recently attracted tremendous attention since they enable agents to apply self-supervised or unsupervised RL methods in settings where online collection is infeasible. We, thus, argue that this is particularly important for MARL problems since online exploration in multi-agent settings may not be feasible in many settings[24], but learning with unsupervised or meta-learned[25] outcome-driven objectives via offline data is still possible. However, it is unclear yet whether the effectiveness of sequence modelling through transformer architecture also applies to MARL problems.

In this paper, we propose multi-agent decision transformers (MADT), an architecture that casts the problem of MARL as conditional sequence modelling. Our mandate is to understand if the proposed MADT can learn through pre-training a generalized policy on offline datasets, which can then be effectively used to other downstream environments (known or unknown). As a study example, we specifically focus on the well-known challenge for MARL tasks: the StarCraft multi-agent challenge (SMAC)[26], and demonstrate the possibility of solving multiple SMAC tasks with one big sequence model. Our contribution is as follows: We propose a series of transformer variants for offline MARL by leveraging the sequential modelling of the attention mechanism. In particular, we validate our pre-trained sequential model in the challenging multi-agent environment for its sample efficiency and transferability. We built a dataset with different skill levels covering different variations of SMAC scenarios. Experimental results on SMAC tasks show that MADT enjoys fast adaptation and superior performance via learning one big sequence model.

The main challenges in our offline pre-training and online fine-tuning problems are the out-of-distribution and training paradigm mismatch problems. We tackle these two problems with the sequential model and pre-train the global critic model offline.

2 Related work

Offline deep reinforcement learning. Recent works have successfully applied RL in robotics control[27, 28] and gaming AIs[29] online. However, many works attempt to reduce the cost resulting from online interactions by learning with neural networks from an offline dataset named offline RL methods[23]. There are two classes to divide the offline RL methods: constraint-based and sequential model-based methods. For the constraint-based methods, a straightforward method is to adopt the off-policy algorithm and regard the offline datasets as a replay buffer to learn a policy with promising performance. However, experience existing in offline datasets and interaction with online environments have different distributions, which causes the overestimation in the off-policy (value-based) method[30]. Substantial works presented in offline RL aim at resolving the distribution shift between the static offline datasets and the online environment interactions[30–32]. In addition, depending on the dynamic planning ability of the transition model, Matsushima et al.[33, 34] learn different models offline and regularize the policy efficiently. In particular, Yang et al.[35, 36] constrain off-policy algorithms in the multi-agent field. Related to our work for the improvement of sample efficiency, Nair et al.[37] derive the Karush Kuhn-Tucker (KKT) conditions of the online objective, generating an advantage weight to avoid the out-of-distribution (OOD) problem. For the sequential model-based methods, Decision transformer outperforms many state-of-the-art offline RL algorithms by regarding the offline policy training process as a sequential modelling and testing it online[19, 38]. In contrast, we show a transformer-based method in the multi-agent field, attempting to transfer across many scenarios without extra constraints. By sharing the sequential model across agents and learning a global critic network offline, we conduct a pre-trained multi-agent policy that can be continuously fine-tuned online.

Multi-agent reinforcement learning. As a natural extension from single-agent RL, MARL[1] attracts much attention to solve more complex problems under Markov games. Classic algorithms often assume multiple agents to interact with the environment online and collect the experience to train the joint policy from scratch. Many empirical successes have been demonstrated in solving zero-sum games through MARL methods[8, 39]. When solving decentralized partially observable Markov decision processes (Dec-POMDPs) or potential games[40], the framework of centralized training and decentralized execution (CTDE) is often employed[41–45] where a centralized critic is trained to gather all agents′ local observations and assign credits. While CTDE methods rely on the individual-global-max assumption[46], another thread of work is built on the so-called advantage decomposition lemma[47], which holds in general for any cooperative game; such a lemma leads to provably convergent multi-agent trust-region methods[48] and constrained policy optimization methods[49].

Transformer. Transformer[50] has achieved a great breakthrough to model relations between the input and output sequence with variable length, for the sequence-to-sequence problems[51], especially in machine translation[52] and speech recognition[53]. Recent works even reorganize the vision problems as the sequential modelling process and construct the state-of-the-art (SOTA) model with pretraining, named vision transformers (ViT)[16, 54, 55].

Due to the Markovian property of trajectories in offline datasets, we can utilize Transformer as that in language modelling. Therefore, Transformer can bridge the gap between supervised learning in the offline setting and reinforcement learning in online interaction because of the representation capability. We claim that the components in Markov games are sequential, then utilise the transformer for each agent to fit a transferable MARL policy. Furthermore, we fine-tune the learned policy via trial-and-error.

3 Methodology

In this section, we demonstrate how the transformer is applied to our offline pre-training MARL framework. First, we introduce the typical paradigm and computation process for the multi-agent reinforcement learning and attention-based model. Then, we introduce an offline MARL method, in which the transformer sequentially maps between the local observations and actions of each agent in the offline dataset via parameter sharing. Then we leverage the hidden representation as the input of the MADT to minimize the cross-entropy loss. Furthermore, we introduce how to integrate the online MARL with MADT in constructing our whole framework to train a universal MARL policy. To accelerate the online learning, we load the pre-trained model as a part of the MARL algorithms and learn the policy based on experience in the latest buffer stored from the online environment. To train a universal MARL policy quickly adapting to other tasks, we bridge the gap between different scenarios from observations, actions, and available actions, respectively. Fig. 1 overviews our method from the perspective of offline pre-training with supervised learning and online fine-tuning with MARL algorithms. The main contributions of this work are summarized as follows: 1) We conducted an offline dataset for multi-agent offline pre-training on the well-known challenging task, SMAC; 2) To improve the sample efficiency online, we propose fine-tuning the pre-trained multi-agent policy instantiated with the sequence model by sharing policy among agents and show the strong capacity of sequence modelling for multi-agent reinforcement learning in the few-shot and zero-shot settings; 3) We propose pre-training an actor and a critic to fine-tune with the policy-based network. In contrast to the imitation learning that only fits a policy network offline, MADT trains the actor and critic offline together and fine-tunes them online in the RL-style training scheme. We also give some empirical conclusions, such as the effect of reward-to-go in the online fine-tuning stage and the multi-task padding method on SMAC.

Multi-agent reinforcement learning. For the Markov game, which is a multi-agent extension of the Markov decision process (MDP), there is a tuple representing the essential elements $\langle S, A, R, P, n, \gamma \rangle$, where S denotes the state space of n agents, $S_1 \times S_2 \cdots \times S_n \to S$. A_i is the action space of each agent i, $P : S_i \times A_i \to PD(S_i)$ denotes the transition function emitting the distribution over the state space and A is the joint action space, $R_i : S \times A_i \to \mathbf{R}$ is the reward function of each agent and takes action following their policies $\pi(a|s) \in \Pi_i : S \to PD(A)$ from the policy space Π_i, where Π_i denotes the policy space of agent i, $a \in A_i$, and $s \in S_i$. Each agent aims to maximize its long-term reward $\sum_t \gamma^t r_i^t$, where $r_i^t \in R_i$ denotes the reward of agent i in time t and γ denotes the discount factor. In the cooperative setting, we also denote the r_i with r shared among agents for the simplification.

Attention-based model. The attention-based model has shown its stable and strong representation capability. The scale dot-production attention uses the self-attention mechanism demonstrated in [50]. Let $\boldsymbol{Q} \in \mathbf{R}^{t_q \times d_q}$ be the quries, $\boldsymbol{K} \in \mathbf{R}^{t_k \times d_k}$ be the keys, and $\boldsymbol{V} \in \mathbf{R}^{t_v \times d_v}$ be the values, where t_* are the element numbers of different inputs and d_* are the corresponding element dimensions. Normally, $t_k = t_v$ and $d_q = d_k$. The outputs of self-attention are computed as

$$\text{Attention}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \text{softmax}(\frac{\boldsymbol{Q}\boldsymbol{K}^{\mathrm{T}}}{\sqrt{d_k}})\boldsymbol{V} \tag{1}$$

where the scalar $1/\sqrt{d_k}$ is used to prevent the softmax function from entering regions that have very small gradients. Then, we introduce the multi-head attention process as follows:

$$\text{MultiHead}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \text{Concat}(\text{head}_1, \cdots, \text{head}_h)\boldsymbol{W}^O \tag{2}$$

$$\text{head}_i = \text{Attention}(\boldsymbol{Q}\boldsymbol{W}_i^Q, \boldsymbol{K}\boldsymbol{W}_i^K, \boldsymbol{V}\boldsymbol{W}_i^V). \tag{3}$$

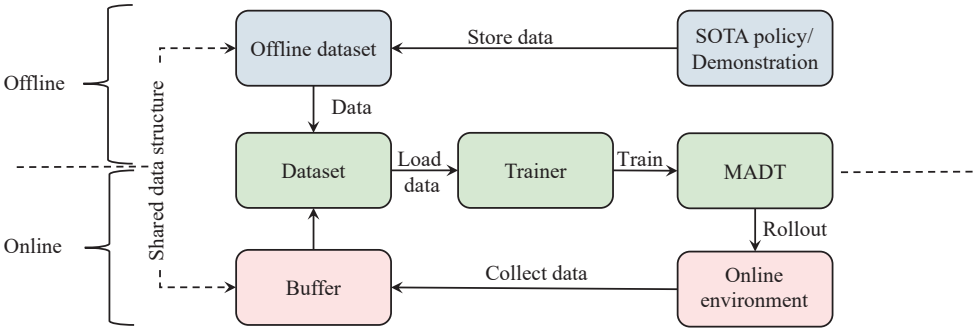


Fig. 1 Overview of the pipeline for pre-training the general policy and fine-tuning it online

The position-wise feed-forward network is another core module of the transformer. It consists of two linear transformations with a ReLU activation function. The dimensionality of inputs and outputs is d_{model}, and that of the feed forward layer is d_{ff}. Specially,

$$\mathrm{FFN}(x) = \max(0, x\boldsymbol{W}_1 + \boldsymbol{b}_1)\boldsymbol{W}_2 + \boldsymbol{b}_2 \tag{4}$$

where $\boldsymbol{W}_1 \in \mathbf{R}^{d_{model} \times d_{ff}}$ and $\boldsymbol{W}_2 \in \mathbf{R}^{d_{ff} \times d_{model}}$ are the weights, and $\boldsymbol{b}_1 \in \mathbf{R}^{d_{ff}}$ and $\boldsymbol{b}_2 \in \mathbf{R}^{d_{model}}$ are the biases. Across different positions are the same linear transformations. Note that the position encoding for leveraging the order of the sequence is as follows:

$$\begin{aligned} \mathrm{PE}(pos, 2i) &= \sin(pos/10\,000^{2i/d_{model}}) \\ \mathrm{PE}(pos, 2i+1) &= \cos(pos/10\,000^{2i/d_{model}}). \end{aligned} \tag{5}$$

3.1 Multi-agent decision transformer

Algorithm 1 shows the offline training process for a single-task of MADT, in which we autoregressively encode the trajectories from the offline datasets in offline pre-trained MARL and train the transformer-based network with supervised learning. We carefully reformulate the trajectories as the inputs of the causal transformer that are different from those in the Decision transformer[19]. We deprecate the reward-to-go and actions that are encoded with states together in the single-agent DT. We will interpret the reason for this in the next section. Similar to the seq2seq models, MADT is based on the autoregressive architecture with reformulated sequential inputs across timescales. The left part of Fig. 2 shows the architecture. The causal transformer encodes the agent i's trajectory sequence τ_i^t at the time step t to a hidden representation $\boldsymbol{h}_i^t = (h_1, h_2, \cdots, h_l)$ with a dynamic mask. Given $\boldsymbol{h}_t$, the output at the time step t is based on the previous data and then consumes the previously emitted actions as additional inputs when predicting a new action.

Algorithm 1. MADT-Offline: Multi-agent decision transformer

1) **Input:** Offline dataset $\mathcal{D}: \{\tau_i : \langle s_i^t, o_i^t, a_i^t, v_i^t, d_i^t, r_i^t \rangle_{t=1}^{T}\}_{i=1}^{n}$, v_i^t denotes the available action
2) **Initialize** θ for the Causal transformer
3) **Initialize** α as the learning rate, C as the context length, and n as the maximum agent number
4) **for** $\tau = \{\tau_1, \cdots, \tau_i, \cdots, \tau_n\}$ in $\mathcal{D}$ **do**
5) Chunk the trajectory into $\tau_i = \{r_i^t, s_i^t, a_i^t\}_{t \in 1:C}$ as the ground truth samples, where C is the context length, and mask the trajectory when d_i^t is true
6) **for** $\tau_i^t = \{\tau_i^1 \cdots \tau_i^t \cdots \tau_i^C\}$ in τ_i **do**
7) Mask illegal actions via $P(a_i^t | \tau_i^{<t}; \theta') = 0$ if v_i^t is true
8) Predict the action $\hat{a}_i^t = \arg\max_{a_i} P(a_i | \tau_i^{<t}; \theta')$
9) Update θ with $\theta = \arg\max_{\theta'} \frac{1}{C} \sum_{t=1}^{C} P(a_i^t) \times \log P(\hat{a}_i^t | \tau_i^{<t}; \theta')$
10) **end for**
11) **end for**

Trajectories reformulation as input. We model the lowest granularity at each time step as a modelling unit x_t from the static offline dataset for the concise representation. MARL has many elements, such as $\langle \mathrm{global\_state}, \mathrm{local\_observation} \rangle$, different from the single agent. It is reasonable for sequential modelling methods to model them in a MDP. Therefore, we formulate the trajectory as follows:

$$\tau^i = (x_1, \cdots, x_t, \cdots, x_C), \quad \text{where } x_t = (s_t, o_t^i, a_t^i)$$

where s_t^i denotes the global shared state, o_t^i denotes the individual observation for agent i at time step t, and a_t^i denotes the action. We regard x_t as a token and process the whole sequence similar to the scheme in the language modelling.

Output sequence construction. To bridge the gap between training with the whole context trajectory and

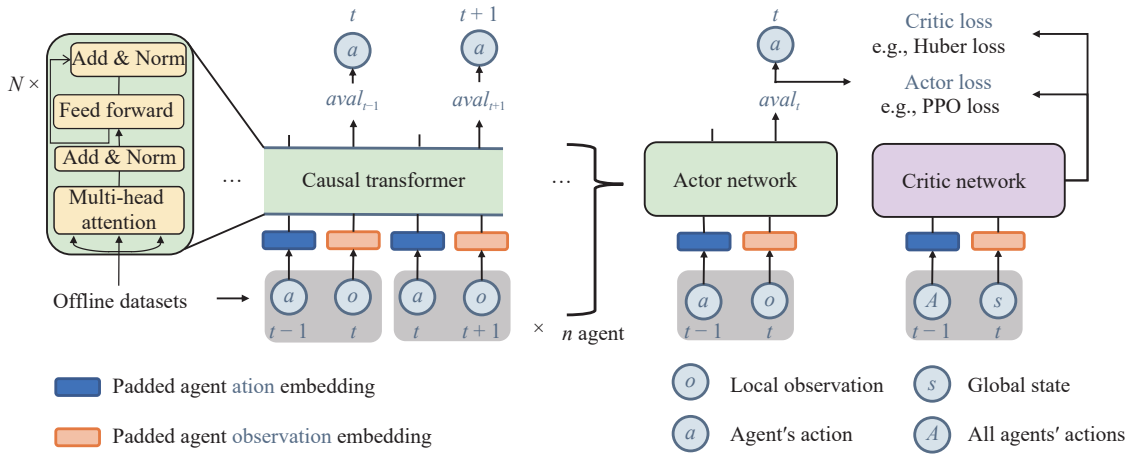


Fig. 2 Detailed model structure for offline and online MADT

testing with only previous data, we mask the context data to autoregressively output in the time step t with previous data in $\langle 1, \cdots, t-1 \rangle$. Therefore, MADT predicts the sequential actions at each time step using the decoder as follows:

$$y = a_t = \arg\max_a p_\theta(a|\tau, a_1, \cdots, a_{t-1}) \tag{6}$$

where θ denotes the parameters of MADT and τ denotes the trajectory including the global state s, local observation o before the time step t, p_θ is the distribution over the legal action space under the available action v.

Core module description. MADT differs from the transformers in conventional sequence modelling tasks that take inputs with position encoding and decode the encoded hidden representation autoregressively. We use the masking mechanism with a lower triangular matrix to compute the attention:

$$\text{Attention}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \text{softmax}\Big(\frac{\boldsymbol{Q}\boldsymbol{K}^{\mathrm{T}}}{\sqrt{d_k}} + M\Big)\boldsymbol{V} \tag{7}$$

where M is the mask matrix that ensures that the input at the time step t can only correlate with the input from $\langle 1, \cdots, t-1 \rangle$. We employ the cross-entropy (CE) as the total sequential prediction loss and utilize the available action v to ensure agents take those illegal actions with a probability of zero. The CE loss can be represented as follows:

$$L_{CE}(\theta) = \frac{1}{C}\sum_{t=1}^{C} P(a_t) \log P(\hat{a}_t|\tau_t, \hat{a}_{<t}; \theta) \tag{8}$$

where C is the context length, a_t is the ground truth action, τ_t includes $\{s_{1:t}, o_{1:t}\}$. $\hat{a}$ denotes the output of MADT. The cross-entropy loss shown above aims to minimize the distribution distance between the prediction and the ground truth.

3.2 Multi-agent decision transformer with PPO

The method above can fit the data distribution well, resulting from the sequential modelling capacity of the transformer. However, it fails to work well when pre-training on the offline datasets and improves continually by interacting with the online environment. The reason is the mismatch between the objectives of the offline and online phase. In the offline stage, the imitation-based objective conforms to a supervised learning style in MADT and ignores measuring each action with a value model. When the pre-trained model is loaded to interact with the online environment, the buffer will only collect actions conforming to the distributions of the offline datasets rather than those corresponding to high reward at this state. That means the pre-trained policy is encouraged to choose an action to be identical to the distribution in the offline dataset, even though it leads to a low reward. Therefore, we need to design another paradigm, MADT-PPO, to integrate RL and supervised learning for fine-tuning in Algorithm 2. Fig. 2 shows the pre-training and fine-tuning framework. A direct method is to share the pre-trained model across each agent and implement the REINFORCE algorithm[56]. However, only actors result in higher variance, and the employment of a critic to assess state values is necessary. Therefore, in online MARL, we leverage an extension of PPO, the state-of-the-art algorithm on tasks of StarCraft, multi-agent particle environment (MPE), and even the return-based game Hanabi[57]. In the offline stage, we adopt the strategy mentioned before to pre-train an offline policy for each agent $\pi(o_i, a_i)$ and additionally use the global state to pre-train a centralized critic $V_\phi(s)$. In the fine-tuning stage, we first load the offline pre-trained sharing policy as each agent′s online initial policy $\pi_i(o_i, a_i)$. When the critic is pre-trained, we instantiate the centralized critic with the pre-trained model as $V_\phi(s)$. To fine-tune the pre-trained multi-agent policy and critic model, multiple agents clear the buffer and interact with the environment to learn the policy via maximizing the following PPO objective:

$$\sum_{i=1}^{n} E_{s\sim\rho_{\theta_{\text{old}}}, a\sim\pi_{\theta_{\text{old}}}}[\min(\omega\hat{A}(s,a),\ \text{clip}(\omega, 1-\epsilon, 1+\epsilon)\hat{A}(s,a))] \tag{9}$$

where $\omega = \pi_\theta(o_i, a_i)/\pi_{\theta_{\text{old}}}(o_i, a_i)$ denotes the importance weight using in PPO-style algorithms, $\hat{A}$ denotes the advantage function computed by reward and the critic model and ϵ denotes the clip parameter. The detailed fine-tuning pipeline can be found in Algorithm 2.

Algorithm 2. MADT-Online: Multi-agent decision transformer with PPO

1) **Input:** Offline dataloader $\mathcal{D}$, Pretrained MADT policy with parameter θ
2) **Initialize** θ and ϕ are the parameters of an actor $\pi_\theta(a_i|o_i)$ and critic $V_\phi(s)$ respectively, which could be inherited directly from pre-trained models
3) **Initialize** n as the agent number, γ as the discount factor, and ϵ as clip ratio.
4) **for** $\tau = \{\tau_1, \cdots, \tau_i, \cdots, \tau_n\}$ in $\mathcal{D}$ **do**
5) Sample $\tau_i = \{s^t, o_i^t, a_i^t\}_{t\in 1:C}$ as the ground truth, where C is the context length
6) Compute the advantage function $\hat{A}(s, a_i) = \sum_t \gamma^t r(s, a_i) - V_\phi(s)$
7) Compute the important weight $w = \pi_\theta(o_i, a_i)/\pi_{\theta_{old}}(o_i, a_i)$
8) Update θ_i for $i \in 1, \cdots, n$ via:
9) $\theta_i = \arg\max E_{s\sim\rho_{\theta_{old}}, a\sim\pi_{\theta_{old}}}[\text{clip}(w, 1-\epsilon, 1+\epsilon)\hat{A}(s, a_i)]$

10) Compute the MSE loss $L_\phi = \frac{1}{2}[\sum_t \gamma^t r^t - V_\phi(s)]^2$

11) Update the critic network via $\phi = \arg\min_\phi L_\phi$

12) **end for**

3.3 Universal model across scenarios

To train a universal policy for each of the scenarios in the SMAC which might vary with agent number in feature space, action space, and reward ranges, we consider the modification list below.

Parameters sharing across agents. When offline examples are collected from multiple tasks or the test phase owns the different agent numbers from the offline datasets, the difference in agent numbers across tasks is an intractable problem for deciding the number of actors. Thus, we consider sharing the parameters across all actors with one model as well as attaching one-hot agent IDs into observations for compatibility with a variable number of agents.

Feature encoding. When the policy needs to generalize to new scenarios that arise from different feature shapes, we propose encoding all features into a universal space by padding zero at the end and mapping them to a low-dimensional space with fully connected networks.

Action masking. Another issue is the different action spaces across scenarios. For example, fewer enemies in a scenario means fewer potential attack options as well as fewer available actions. Therefore, an extra vector is utilized to mute the unavailable actions so that their probabilities are always zero during both the learning and evaluating processes.

Reward scaling. Different scenarios might vary in reward ranges and lead to unbalanced models during multi-task offline learning. To balance the influence of examples from different scenarios, we scale their rewards to the same range to ensure that the output models have comparable performance across different tasks.

4 Experiments

We show three experimental settings: offline MARL, online MARL by loading the pre-trained model, and few-shot or zero-shot offline learning. For the offline MARL, we expect to verify the performance of our method by pre-training the policy and directly testing on the corresponding maps. In order to demonstrate the capacity of the pre-trained policy on the original or new scenarios, we aim to demonstrate the fine-tuning in the online environment. Experimental results in offline MARL show that our MADT-offline in Section 3.1 outperforms the state-of-the-art methods. Furthermore, MADT-online in Section 3.2 can improve the sample efficiency across multiple scenarios. Besides, the universal MADT trained from multi-task data with MADT-online generalizes well in each scenario in a few-shot or even zero-shot setting.

4.1 Offline datasets

The offline datasets are collected from the running policy, MAPPO[58], on the well-known SMAC task[26]. Each dataset contains a large number of trajectories: $\tau := (s_t, o_t, a_t, r_t, done_t, v_t)_{t=1}^T$. Different from D4RL[59], our datasets conform to the property of DecPOMDP, which owns local observations and available actions for each agent. In the appendix, we list the statistical properties of the offline datasets in Tables A1 and A2.

4.2 Offline multi-agent reinforcement learning

In this experiment, we aim to validate the effectiveness of the MADT offline version in Section 3.1 as a framework for offline MARL on the static offline datasets. We train a policy on the offline datasets with various qualities and then apply it to an online environment, StarCraft[26]. There are also baselines under this setting, such as behavior cloning (BC), as a kind of imitation learning method showing stable performance on single-agent offline RL. In addition, we employ the conventional effective single-agent offline RL algorithms, BCQ[32], CQL[31], and ICQ[35], and then use the extension method by simply mixing each agent value network proposed by [35] for multi-agent setting, denoting it as "xx-MA". We compare the performance of the MADT offline version with other abovementioned offline RL algorithms under online evaluation in the MARL environment. To verify the quality of our collected datasets, we chose data from different levels and trained the baselines as well as our MADT. Fig. 3 shows the overall performance on various quality datasets. The baseline methods enhance their performance stably, indicating the quality of our offline datasets. Furthermore, our MADT outperforms the offline MARL baselines and converges faster across easy, hard, and super hard maps (2s3z, 3s5z, 3s5z VS. 3s6z, corridor). From the initial performance in the evaluation period, our pretrained model gives a higher return than baselines in each task. Besides, our model can surpass the average performance in the offline dataset.

4.3 Offline pre-training and online fine-tuning

The experimental designed in this subsection intends to answer the question: Is the pre-training process necessary for online MARL? First, we compare the online version of MADT in Section 3.2 with and without loading the pre-trained model. If training MADT only by online experience, we can view it as a transformer-based MAPPO replacing the actor and critic backbone networks with the transformer. Furthermore, we validate that our framework MADT with the pre-trained model can improve sample efficiency on most easy, hard, and

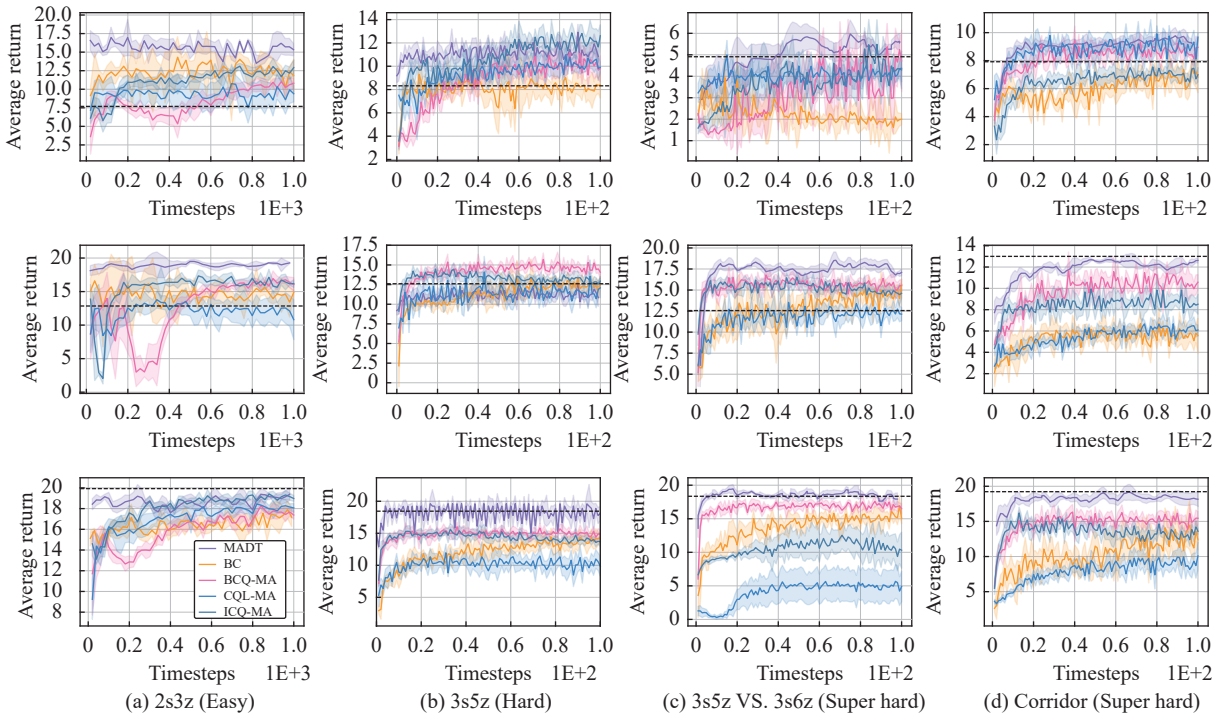


Fig. 3 Performance of offline MADT compared with baselines on four easy or (super-)hard SMAC maps. The dotted lines represent the mean values in the training set. Columns (a)–(d) are average returns from (poor, medium, good) datasets from top to the bottom.

super hard maps.

Necessity of the pretrained model. We train our MADT based on the datasets collected from a map and fine-tune it on the same map online with the MAPPO algorithm. For comparison fairness, we use the transformer as both actor and critic networks with and without the pre-trained model. Primarily, we choose three maps from easy, hard, and super hard maps to validate the effectiveness of the pre-trained model in Fig. 4. Experimental results show that the pre-trained model converges faster than the algorithm trained from scratch, especially in challenging maps.

Improving sample efficiency. To validate the sample efficiency improvement by loading our pre-trained MADT and fine-tuning it with MAPPO, we compare the overall framework with the state-of-the-art algorithm, MAPPO[58], without the pre-training phase. We measure the sample efficiency in terms of the time to threshold mentioned in [60], which denotes the number of online interactions (timesteps) to achieve a predefined threshold in Table 1, and our pre-trained model needs much less than the traditional MAPPO to achieve the same win rate.

4.4 Generalization with multi-task pre-training

Experiments in this section explore the transferability of the universal MADT mentioned in Section 3.3, which is pre-trained with mixed data from multiple tasks. Depending on whether the downstream tasks have been seen or not, the few-shot experiments are designed to validate the adaptability of the seen tasks. In contrast, the zero-shot experiments are designed for the held-out maps.

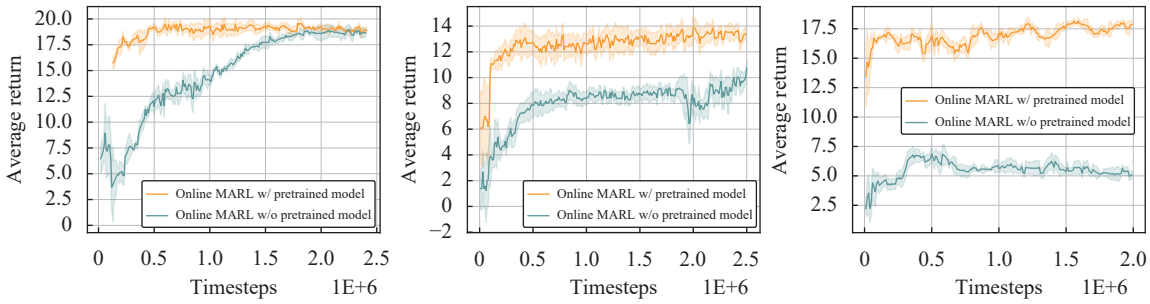
Few-shot learning. The results in Fig. 5(a) show that our method can utilize multi-task datasets to train a universal policy and generalize to all tasks well. Pre-trained MADT can achieve higher returns than the model trained from scratch when we limit the interactions with the environment.

Zero-shot learning. Fig. 5(b) shows that our universal MADT can surprisingly improve performance on downstream tasks even if it has not been seen before (3 stalkers VS. 4 zealots).

4.5 Ablation study

The experiments in this subsection are designed to answer the following research questions: RQ1: Why should we choose MAPPO for the online phase? RQ2: Which kind of input should be used to make the pre-trained model beneficial for the online MARL? RQ3: Why cannot the offline version of MADT be improved in the online fine-tuning period after pre-training?

Suitable online algorithm. Although the selection of the MARL algorithm for the online phase should be flexible according to specific tasks, we design experiments to answer RQ1 here. As discussed in Section 3, we can train Decision Transformer for each agent and fine-tune it online with an MARL algorithm. An intuitive method is to load the pre-trained transformer and take it as the policy network for fine-tuning with the policy gradient method, e.g., REINFORCE[56]. However, for the

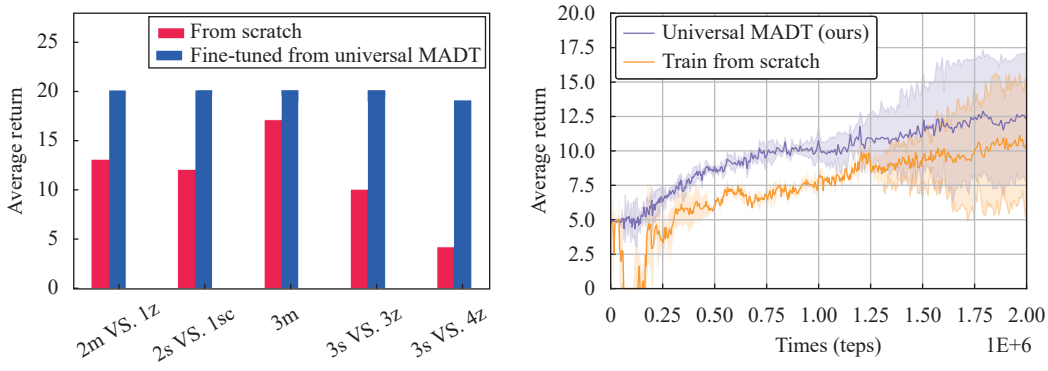


(a) So many baneling (Easy) (b) 5 m VS. 6 m (Hard) (c) 3s5z VS. 3s6z (Super hard)

Fig. 4 Average returns with and without the pre-trained model

Table 1 Number of interactions needed to achieve the win rate 20%, 40%, 60%, 80% and 100% for the training policy with (MAPPO/pre-trained MADT). "–" means no more samples are needed to reach the target win rate. "∞" represents that policies cannot reach the target win rate.

Maps	# Samples to achieve the win rate					Maps	# Samples to achieve the win rate				
	20%	40%	60%	80%	100%		20%	40%	60%	80%	100%
2m VS. 1z (Easy)	8E+4/–	1E+5/–	1.3E+5/–	1.5E+5/–	1.6E+5/–	3s VS. 5z (Hard)	8E+5/–	8.5E+5/–	8.7E+5/ 1.5E+4	9E+5/ 5E+4	2E+6/ 1.5E+5
3m (Easy)	3.2E+3/–	8.3E+4/–	3.2E+5/–	4E+5/–	7.2E+5/–	2c VS. 64zg (Hard)	2E+5/–	3E+5/–	4E+5/ 8E+4	5E+5/ 1E+5	1.8E+6/ 5E+5
2s VS. 1sc (Easy)	1E+4/–	2.5E+4/–	3E+4/–	8E+4/ 4E+4	3E+5/ 1.2E+5	8m VS. 9m (Hard)	3E+5/–	6E+5/–	1.4E+6/ 2E+4	2E+6/ 8E+4	∞/2.2E+6
3s VS. 3z (Easy)	2.5E+5/–	3E+5/–	6.2E+5/ 1E+4	7.3E+5/ 1.5E+5	8E+5/ 2.9E+5	5m VS. 6m (Hard)	1.5E+6/ 2E+5	2.5E+6/ 8E+5	5E+6/ 2E+6	∞/∞	∞/∞
3s VS. 4z (Easy)	3E+5/–	4E+5/–	5E+5/–	6.2E+5/–	1.5E+6/ 1.8E+5	3s5z (Hard)	8E+5/ 6.3E+4	1.3E+6/ 1E+5	1.5E+6/ 4E+5	1.9E+6/ 1E+6	2.5E+6/ 2E+6
So many baneling (Easy)	3.2E+4/ 8E+3	1E+5/ 4E+4	3.2E+5/ 7E+4	5E+5/ 8E+4	1E+6/ 6.4E+5	10m VS. 11m (Hard)	2E+5/–	3.5E+5/–	4E+5/ 2.8E+4	1.7E+6/ 1.2E+5	4E+6/ 2.5E+5
8m (Easy)	4E+5/–	5E+6/–	5.6E+5/–	5.6E+5E+ 5/1.6E+5	8.8E+5/ 2.4E+5	MMM2 (Super hard)	1E+6/	1.8E+6/	2.3E+6/	4E+6/	∞/∞
MMM (Easy)	5.2E+4/–	8E+4/–	3E+5/–	4.5E+5/–	1.8E+6/ 6E+5	3s5z VS. 3s6z (Super hard)	1.8E+6/–	2.5E+6/–	3E+6/ 8E+5	5E+6/ 1E+6	∞/∞
Bane VS. bane (Easy)	3.2E+3/–	3.2E+3/–	3.2E+5/–	4E+5/–	5.6E+5/–	Corridor (Super hard)	1.5E+6/–	1.8E+6/–	2E+6/–	2.8E+6/–	7.8E+6/ 4E+5



(a) (b)

Fig. 5 Few-shot and zero-shot validation results. (a) shows the average returns of the universal MADT pre-trained from all five tasks data and the policy trained from scratch, individually. We limit the environment interaction to 2.5 M steps. (b) shows the average returns of a held-out map (3s VS. 4z), where the universal MADT is trained from data on (2m VS. 1z, 2s VS. 1sc, 3m, 3s VS. 3z).

reason of the high variance mentioned in Section 3.2, we choose MAPPO as the online algorithm and compare its performance in improving the sample efficiency during the online period in Fig. 6(a).

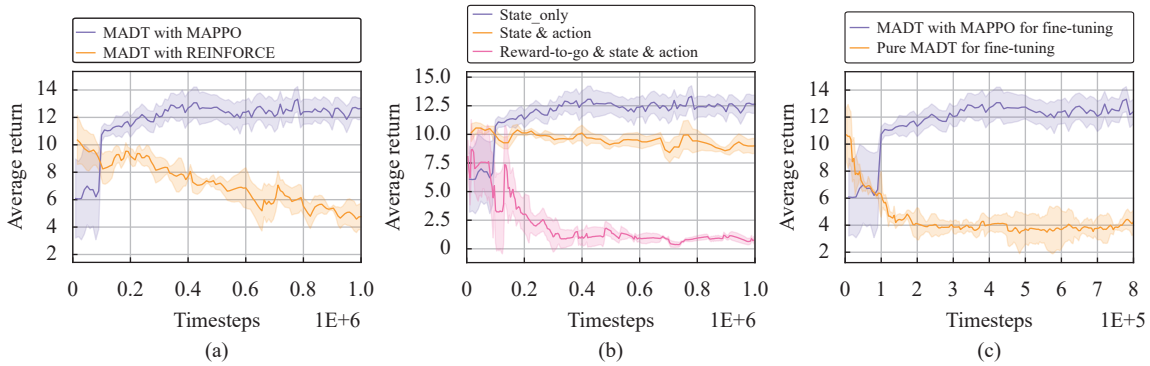


Fig. 6 Ablation results on a hard map, 5m_vs_6m, for validating the necessity of (a) MAPPO in MADT-online, (b) Input formulation, (c) online version of MADT.

Dropping reward-to-go in MADT. To answer RQ2, we compare different inputs embedded into the transformer, including the combination of state, reward-to-go, and action. We find reward-to-go harmful to online fine-tuning performance, as shown in Fig. 6(b). We suppose the distribution of reward-to-go is the mismatch between offline data and online samples. That is, the rewards of online samples are usually lower than those of offline data due to stochastic exploration at the beginning of the online phase. It deteriorates the fine-tuning capability of the pre-trained model, and based on Fig. 6(b), we only choose states as our inputs for pre-training and fine-tuning.

Integrating online MARL with MADT. To answer RQ3, we directly apply the offline version of MADT for pre-training and fine-tune it online. However, Fig. 6(c) shows that it cannot be improved during the online phase. We analyse the results caused by the absence of motivation for chasing higher rewards and conclude that offline MADT is supervised learning and tends to fit its collected experience even with unsatisfactory rewards.

5 Conclusions

In this work, we propose MADT, an offline pre-trained model for MARL, which integrates the transformer to improve sample efficiency and generalizability in tackling SMAC tasks. MADT learns a big sequence model that outperforms the state-of-the-art methods in offline settings, including BC, BCQ, CQL, and ICQ. When applied in online settings, the pre-trained MADT can drastically improve the sample efficiency. We applied MADT to train a generalizable policy over a series of SMAC tasks and then evaluated its performance under both few-shot and zero-shot settings. The results demonstrate that the pre-trained MADT policy adapts quickly to new tasks and improves performance on different downstream tasks. To the best of our knowledge, this is the first work that demonstrates the effectiveness of offline pre-training and the effectiveness of sequence modelling through transformer architectures in the context of MARL.

Appendix A Properties of datasets

We list the properties of our offline datasets in Tables A1 and A2.

Appendix B Detailsofhyper-parameters

Details of hyper-parameters used for MADT experiments are listed from Tables B1–B5.

Table A1 Properties for our offline dataset collected from the experience of multi-agent PPO on the easy maps of SMAC

Maps	Difficulty	Data quality	# Samples	Reward distribution (mean (±std))
3m	Easy	3m-poor	62 528	6.29 (± 2.17)
		3m-medium	–	–
		3m-good	1 775 159	19.99 (± 0.18)
8m	Easy	8m-poor	157 133	6.43 (± 2.41)
		8m-medium	297 439	11.95 (± 0.94)
		8m-good	2 781 145	20.00 (± 0.16)
2s3z	Easy	2s3z-poor	147 314	7.69 (± 1.77)
		2s3z-medium	441 474	12.85 (± 1.37)
		2s3z-good	4 177 846	19.93 (± 0.67)

Table A1 (continued) Properties for our offline dataset collected from the experience of multi-agent PPO on the easy maps of SMAC

Maps	Difficulty	Data quality	# Samples	Reward distribution (mean (±std))
2s_vs_1sc	Easy	2s_vs_1sc-poor	12 887	6.62 (± 2.74)
		2s_vs_1sc-medium	33 232	11.70 (± 0.73)
		2s_vs_1sc-good	1 972 972	20.23 (± 0.02)
3s_vs_4z	Easy	3s_vs_4z-poor	216 499	7.58 (± 1.45)
		3s_vs_4z-medium	335 580	12.13 (±m 1.38)
		3s_vs_4z-good	3 080 634	20.19 (± 0.40)
MMM	Easy	MMM-poor	326 516	7.64 (± 2.05)
		MMM-medium	648 115	12.23 (± 1.37)
		MMM-good	2 423 605	20.08 (± 1.67)
So_many_baneling	Easy	So_many_baneling-poor	1 542	9.08 (± 0.66)
		So_many_baneling-medium	59 659	13.31 (± 1.14)
		So_many_baneling-good	1 376 861	19.46 (± 1.29)
3s_vs_3z	Easy	3s_vs_3z-poor	52 807	8.10 (± 1.37)
		3s_vs_3z-medium	80 948	11.87 (± 1.19)
		3s_vs_3z-good	149 906	20.02 (± 0.09)
2m_vs_1z	Easy	2m_vs_1z-poor	25 333	5.20 (± 1.66)
		2m_vs_1z-medium	300	11.00 (± 0.01)
		2m_vs_1z-good	120 127	20.00 (± 0.01)
Bane_vs_bane	Easy	Bane_vs_bane-poor	63	1.59 (± 3.56)
		Bane_vs_bane-medium	3 507	14.00 (± 0.93)
		Bane_vs_bane-good	458 795	19.97 (± 0.36)
1c3s5z	Easy	1c3s5z-poor	52 988	8.10 (± 1.65)
		1c3s5z-medium	180 357	12.68 (± 1.42)
		1c3s5z-good	2 400 033	19.88 (± 0.69)

Table A2 Properties for our offline dataset collected from the experience of multi-agent PPO on the hard and super hard maps of SMAC

Maps	Difficulty	Data quality	# Samples	Reward distribution (mean (±std))
5m_vs_6m	Hard	5m_vs_6m-poor	1 324 213	8.53 (± 1.18)
		5m_vs_6m-medium	657 520	11.03 (± 0.58)
		5m_vs_6m-good	503 746	20 (± 0.01)
10m_vs_11m	Hard	10m_vs_11m-poor	140 522	7.64 (± 2.39)
		10m_vs_11m-medium	916 845	12.72 (± 1.25)
		10m_vs_11m-good	895 609	20 (± 0.01)
2c_vs_64zg	Hard	2c_vs_64zg-poor	10 830	8.91 (± 1.01)
		2c_vs_64zg-medium	97 702	13.05 (± 1.37)
		2c_vs_64zg-good	2 631 121	19.95 (± 1.24)
8m_vs_9m	Hard	8m_vs_9m-poor	184 285	8.18 (± 2.14)
		8m_vs_9m-medium	743 198	12.19 (± 1.14)
		8m_vs_9m-good	911 652	20 (± 0.01)

Table A2 (continued) Properties for our offline dataset collected from the experience of multi-agent PPO on the hard and super hard maps of SMAC

Maps	Difficulty	Data quality	# Samples	Reward distribution (mean (±std))
3s_vs_5z	Hard	3s_vs_5z-poor	423 780	6.85 (± 2.00)
		3s_vs_5z-medium	686 570	12.12 (± 1.39)
		3s_vs_5z-good	2 604 082	20.89 (± 1.38)
3s5z	Hard	3s5z-poor	365 389	8.32 (± 1.44)
		3s5z-medium	2 047 601	12.61 (± 1.32)
		3s5z-good	1 448 424	18.45 (± 2.03)
3s5z_vs_3s6z	Super hard	3s5z_vs_3s6z-poor	594 089	7.92 (± 1.77)
		3s5z_vs_3s6z-medium	2 214 201	12.56 (± 1.37)
		3s5z_vs_3s6z-good	1 542 571	18.35 (± 2.04)
27m_vs_30m	Super hard	27m_vs_30m-poor	102 003	7.18 (± 2.08)
		27m_vs_30m-medium	456 971	13.19 (± 1.25)
		27m_vs_30m-good	412 941	17.33 (± 1.97)
MMM2	Super hard	MMM2-poor	1 017 332	7.87 (± 1.74)
		MMM2-medium	1 117 508	11.79 (± 1.28)
		MMM2-good	541 873	18.64 (± 1.47)
Corridor	Super hard	Corridor-poor	362 553	4.91 (± 1.71)
		Corridor-medium	439 505	13.00 (± 1.32)
		Corridor-good	3 163 243	19.88 (± 0.99)

Table B1 Common hyper-parameters for all MADT experiments for pre-training on a map, taking 3m (easy) as an example

Hyper-parameter	Value	Hyper-parameter	Value	Hyper-parameter	Value
Offline_train_critic	True	Max_timestep	400	Eval_epochs	32
n_layer	2	n_head	2	n_embd	32
Online_buffer_size	64	Model_type	State_only	Mini_batch_size	128

Table B2 Hyper-parameters for MADT experiments in Fig. 3

Maps	offline_episode_num	offline_lr
2s3z	1 000	1E–4
3s5z	1 000	1E–4
3s5z_vs_3s6z	1 000	5E–4
Corridor	1 000	5E–4

Table B3 Hyper-parameters for MADT experiments in Figs. 4, 6 and Table 1

Maps	Offline_episode_num	Offline_lr	Online_lr	Online_ppo_epochs
2c_vs_64zg	1 000	5E–4	5E–4	10
10m_vs_11m	1 000	5E–4	5E–4	10
8m_vs_9m	1 000	1E–4	5E–4	10
3s_vs_5z	1 000	1E–4	5E–4	10
3s5z	1 000	1E–4	5E–4	10
3m	1 000	1E–4	5E–4	15

Table B3 (continued) Hyper-parameters for MADT experiments in Figs. 4, 6 and Table 1

Maps	Offline_episode_num	Offline_lr	Online_lr	Online_ppo_epochs
2s_vs_1sc	1 000	1E–4	5E–4	15
MMM	1 000	1E–4	1E–4	5
So_many_baneling	1 000	1E–4	1E–4	5
8m	1 000	1E–4	1E–4	5
3s_vs_3z	1 000	1E–4	1E–4	5
3s_vs_4z	1 000	1E–4	1E–4	5
Bane_vs_bane	1 000	1E–4	1E–4	5
2m_vs_1z	1 000	1E–4	1E–4	5
2c_vs_64zg	1 000	1E–4	1E–4	5
5m_vs_6m	1 000	1E–4	1E–4	10
Corridor	1 000	1E–4	1E–4	10
3s5z_vs_3s6z	1 000	1E–4	1E–4	10

Table B4 Hyper-parameters for MADT experiments in Fig. 5(a)

Hyper-parameter	Value
Offline_map_lists	[3s_vs_4z, 2m_vs_1z, 3m, 2s_vs_1sc, 3s_vs_3z]
Offline_episode_num	[200, 200, 200, 200, 200]
Offline_lr	5E–4
Online_lr	1E–4
Online_ppo_epochs	5

Table B5 Hyper-parameters for MADT experiments in Fig. 5(b)

Hyper-parameter	Value
Offline_map_lists	[2m_vs_1z, 3m, 2s_vs_1sc, 3s_vs_3z]
Offline_episode_num	[250, 250, 250, 250]
Offline_lr	5E–4
Online_lr	1E–4
Online_ppo_epochs	5

Acknowledgements

Linghui Meng was supported in part by the Strategic Priority Research Program of the Chinese Academy of Sciences (No. XDA27030300). Haifeng Zhang was supported in part by the National Natural Science Foundation of China (No. 62206289).

Open Access

References

[1] Y. D. Yang, J. Wang. An overview of multi-agent reinforcement learning from game theoretical perspective. [Online], Available: https://arxiv.org/abs/2011.00583, 2020.

[2] S. Shalev-Shwartz, S. Shammah, A. Shashua. Safe, multi-agent, reinforcement learning for autonomous driving. [Online], Available: https://arxiv.org/abs/1610.03295, 2016.

[3] M. Zhou, J. Luo, J. Villella, Y. D. Yang, D. Rusu, J. Y. Miao, W. N. Zhang, M. Alban, I. Fadakar, Z. Chen, A. C. Huang, Y. Wen, K. Hassanzadeh, D. Graves, D. Chen, Z. B. Zhu, N. Nguyen, M. Elsayed, K. Shao, S. Ahilan, B. K. Zhang, J. N. Wu, Z. G. Fu, K. Rezaee, P. Yadmellat, M. Rohani, N. P. Nieves, Y. H. Ni, S. Banijamali, A. C. Rivers, Z. Tian, D. Palenicek, H. bou Ammar, H. B. Zhang, W. L. Liu, J. Y. Hao, J. Wang. Smarts: Scalable multi-agent reinforcement learning training school for autonomous driving. [Online], Available: https://arxiv.org/abs/2010.09776, 2020.

[4] H. F. Zhang, W. Z. Chen, Z. R. Huang, M. N. Li, Y. D. Yang, W. N. Zhang, J. Wang. Bi-level actor-critic for multi-agent coordination. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, New York, USA, pp. 7325–7332, 2020.

[5] M. N. Li, Z. W. Qin, Y. Jiao, Y. D. Yang, J. Wang, C. X. Wang, G. B. Wu, J. P. Ye. Efficient ridesharing order dispatching with mean field multi-agent reinforcement learning. In *Proceedings of World Wide Web Conference*, ACM, San Francisco, USA, pp. 983–994, 2019. DOI: 10.1145/3308558.3313433.

[6] Y. D. Yang, R. Luo, M. N. Li, M. Zhou, W. N. Zhang, J. Wang. Mean field multi-agent reinforcement learning. In *Proceedings of the 35th International Conference on Machine Learning*, Stockholm, Sweden, pp. 5571–5580, 2018.

[7] Y. D. Yang, L. T. Yu, Y. W. Bai, Y. Wen, W. N. Zhang, J. Wang. A study of AI population dynamics with million-agent reinforcement learning. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, ACM, Stockholm, Sweden, pp. 2133–2135, 2018.

[8] P. Peng, Y. Wen, Y. D. Yang, Q. Yuan, Z. K. Tang, H. T. Long, J. Wang. Multiagent bidirectionally-coordinated nets: Emergence of human-level coordination in learning to play StarCraft combat games. [Online], Available: https://arxiv.org/abs/1703.10069, 2017.

[9] M. Zhou, Z. Y. Wan, H. J. Wang, M. N. Wen, R. Z. Wu, Y. Wen, Y. D. Yang, W. N. Zhang, J. Wang. MALib: A parallel framework for population-based multi-agent reinforcement learning. [Online], Available: https://arxiv.org/abs/2106.07551, 2021.

[10] X. T. Deng, Y. H. Li, D. H. Mguni, J. Wang, Y. D. Yang. On the complexity of computing Markov perfect equilibrium in general-sum stochastic games. [Online], Available: https://arxiv.org/abs/2109.01795, 2021.

[11] T. Haarnoja, A. Zhou, K. Hartikainen, G. Tucker, S. Ha, J. Tan, V. Kumar, H. Zhu, A. Gupta, P. Abbeel, S. Levine. Soft actor-critic algorithms and applications. [Online], Available: https://arxiv.org/abs/1812.05905, 2018.

[12] R. Munos, T. Stepleton, A. Harutyunyan, M. G. Bellemare. Safe and efficient off-policy reinforcement learning. In *Proceedings of the 30th Conference on Neural Information Processing Systems*, Barcelona, Spain, pp. 1054–1062, 2016.

[13] L. Espeholt, R. Marinier, P. Stanczyk, K. Wang, M. Michalski. SEED RL: Scalable and efficient deep-RL with accelerated central inference. In *Proceedings of the 8th International Conference on Learning Representations*, Addis Ababa, Ethiopia, 2020.

[14] L. Espeholt, H. Soyer, R. Munos, K. Simonyan, V. Mnih, T. Ward, Y. Doron, V. Firoiu, T. Harley, I. Dunning, S. Legg, K. Kavukcuoglu. IMPALA: Scalable distributed deep-RL with importance weighted actor-learner architectures. In *Proceedings of the 35th International Conference on Machine Learning*, Stockholm, Sweden, pp. 1407–1416, 2018.

[15] K. M. He, X. L. Chen, S. N. Xie, Y. H. Li, Dollár, R. Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, New Orleans, USA, pp. 15979–15988, 2021. DOI: 10.1109/CVPR52688.2022.01553.

[16] Z. Liu, Y. T. Lin, Y. Cao, H. Hu, Y. X. Wei, Z. Zhang, S. Lin, B. N. Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of IEEE/CVF International Conference on Computer Vision*, IEEE, Montreal, Canada, pp. 9992–10002, 2021. DOI: 10.1109/ICCV48922.2021.00986.

[17] S. Kim, J. Kim, H. W. Chun. Wave2Vec: Vectorizing electroencephalography bio-signal for prediction of brain disease. *International Journal of Environmental Research and Public Health*, vol. 15, no. 8, Article number 1750, 2018. DOI: 10.3390/ijerph15081750.

[18] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei. Language models are few-shot learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, Vancouver, Canada, Article number 159, 2020. DOI: 10.5555/3495724.3495883.

[19] L. L. Chen, K. Lu, A. Rajeswaran, K. Lee, A. Grover, M. Laskin, P. Abbeel, A. Srinivas, I. Mordatch. Decision transformer: Reinforcement learning via sequence modeling. [Online], Available: https://arxiv.org/abs/2106.01345, 2021.

[20] Y. D. Yang, J. Luo, Y. Wen, O. Slumbers, D. Graves, H. bou Ammar, J. Wang, M. E. Taylor. Diverse auto-curriculum is critical for successful real-world multiagent learning systems. In *Proceedings of the 20th International Conference on Autonomous Agents and Multi-agent Systems*, ACM, pp. 51–56, 2021.

[21] N. Perez-Nieves, Y. D. Yang, O. Slumbers, D. H. Mguni, Y. Wen, J. Wang. Modelling behavioural diversity for learning in open-ended games. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 8514–8524, 2021.

[22] X. Y. Liu, H. T. Jia, Y. Wen, Y. J. Hu, Y. F. Chen, C. J. Fan, Z. P. Hu, Y. D. Yang. Unifying behavioral and response diversity for open-ended learning in zero-sum games. In *Proceedings of the 35th Conference on Neural Information Processing Systems*, pp. 941–952, 2021.

[23] S. Levine, A. Kumar, G. Tucker, J. Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. [Online], Available: https://arxiv.org/abs/2005.01643, 2020.

[24] R. Sanjaya, J. Wang, Y. D. Yang. Measuring the non-transitivity in chess. *Algorithms*, vol. 15, no. 5, Article number 152, 2022. DOI: 10.3390/a15050152.

[25] X. D. Feng, O. Slumbers, Y. D. Yang, Z. Y. Wan, B. Liu, S. McAleer, Y. Wen, J. Wang. Discovering multi-agent auto-curricula in two-player zero-sum games. [Online], Available: https://arxiv.org/abs/2106.02745, 2021.

[26] M. Samvelyan, T. Rashid, C. S. de Witt, G. Farquhar, N. Nardelli, T. G. J. Rudner, C. M. Hung, P. H. S. Torr, J. Foerster, S. Whiteson. The StarCraft multi-agent challenge. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, Montreal, Canada, pp. 2186–2188, 2019.

[27] Z. Li, S. R. Xue, X. H. Yu, H. J Gao. Controller optimization for multirate systems based on reinforcement learning. *International Journal of Automation and Computing*, vol. 17, no. 3, pp. 417–427, 2020. DOI: 10.1007/s11633-020-1229-0.

[28] Y. Li, D. Xu. Skill learning for robotic insertion based on one-shot demonstration and reinforcement learning. *International Journal of Automation and Computing*, vol. 18, no. 3, pp. 457–467, 2021. DOI: 10.1007/s11633-021-1290-3.

[29] C. Berner, G. Brockman, B. Chan, V. Cheung, P. Debiak, C. Dennison, D. Farhi, Q. Fischer, S. Hashme, C. Hesse, et al. Dota 2 with large scale deep reinforcement learning. [Online], Available: https://arxiv.org/abs/1912.06680, 2019.

[30] A. Kumar, J. Fu, G. Tucker, S. Levine. Stabilizing off-policy Q-learning via bootstrapping error reduction. In *Proceedings of the 33rd Conference on Neural Information Processing Systems*, Vancouver, Canada, pp. 11761–11771, 2019.

[31] A. Kumar, A. Zhou, G. Tucker, S. Levine. Conservative Q-

learning for offline reinforcement learning. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, Vancouver, Canada, Article number 100, 2020. DOI: 10.5555/3495724.3495824.

[32] S. Fujimoto, D. Meger, D. Precup. Off-policy deep reinforcement learning without exploration. In *Proceedings of the 36th International Conference on Machine Learning*, Long Beach, USA, pp. 2052–2062, 2019.

[33] T. Matsushima, H. Furuta, Y. Matsuo, O. Nachum, S. X. Gu. Deployment-efficient reinforcement learning via model-based offline optimization. In *Proceedings of the 9th International Conference on Learning Representations*, 2021.

[34] D. J. Su, J. D. Lee, J. M. Mulvey, H. V. Poor. MUSBO: Model-based uncertainty regularized and sample efficient batch optimization for deployment constrained reinforcement learning. [Online], Available: https://arxiv.org/abs/2102.11448, 2021.

[35] Y. Q. Yang, X. T. Ma, C. H. Li, Z. W. Zheng, Q. Y. Zhang, G. Huang, J. Yang, Q. C. Zhao. Believe what you see: Implicit constraint approach for offline multi-agent reinforcement learning. [Online], Available: https://arxiv.org/abs/2106.03400, 2021.

[36] J. C. Jiang, Z. Q. Lu. Offline decentralized multi-agent reinforcement learning. [Online], Available: https://arxiv.org/abs/2108.01832, 2021.

[37] A. Nair, M. Dalal, A. Gupta, S. Levine. Accelerating online reinforcement learning with offline datasets. [Online], Available: https://arxiv.org/abs/2006.09359, 2020.

[38] M. Janner, Q. Y. Li, S. Levine. Offline reinforcement learning as one big sequence modeling problem. [Online], Available: https://arxiv.org/abs/2106.02039, 2021.

[39] L. C. Dinh, Y. D. Yang, S. McAleer, Z. Tian, N. P. Nieves, O. Slumbers, D. H. Mguni, H. bou Ammar, J. Wang. Online double oracle. [Online], Available: https://arxiv.org/abs/2103.07780, 2021.

[40] D. H. Mguni, Y. T. Wu, Y. L. Du, Y. D. Yang, Z. Y. Wang, M. N. Li, Y. Wen, J. Jennings, J. Wang. Learning in nonzero-sum stochastic games with potentials. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 7688–7699, 2021.

[41] Y. D. Yang, Y. Wen, J. Wang, L. H. Chen, K. Shao, D. Mguni, W. N. Zhang. Multi-agent determinantal Q-learning. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 10757–10766, 2020.

[42] T. Rashid, M. Samvelyan, C. Schroeder, G. Farquhar, J. Foerster, S. Whiteson. QMIX: Monotonic value function factorisation for deep multi-agent reinforcement learning. In *Proceedings of the 35th International Conference on Machine Learning*, Stockholm, Sweden, pp. 4295–4304, 2018.

[43] Y. Wen, Y. D. Yang, R. Luo, J. Wang, W. Pan. Probabilistic recursive reasoning for multi-agent reinforcement learning. In *Proceedings of the 7th International Conference on Learning Representations*, New Orleans, USA, 2019.

[44] Y. Wen, Y. D. Yang, J. Wang. Modelling bounded rationality in multi-agent interactions by generalized recursive reasoning. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence*, Yokohama, Japan, pp. 414–421, 2020. DOI: 10.24963/ijcai.2020/58.

[45] S. Hu, F. D. Zhu, X. J. Chang, X. D. Liang. UPDeT: Universal multi-agent reinforcement learning via policy decoupling with transformers. [Online], Available: https://arxiv.org/abs/2101.08001, 2021.

[46] K. Son, D. Kim, W. J. Kang, D. E. Hostallero, Y. Yi. QTRAN: Learning to factorize with transformation for cooperative multi-agent reinforcement learning. In *Proceedings of the 36th International Conference on Machine Learning*, Long Beach, USA, pp. 5887–5896, 2019.

[47] J. G. Kuba, M. N. Wen, L. H. Meng, S. D. Gu, H. F. Zhang, D. H. Mguni, J. Wang, Y. D. Yang. Settling the variance of multi-agent policy gradients. In *Proceedings of the 35th Conference on Neural Information Processing Systems*, pp. 13458–13470, 2021.

[48] J. G. Kuba, R. Q. Chen, M. N. Wen, Y. Wen, F. L. Sun, J. Wang, Y. D. Yang. Trust region policy optimisation in multi-agent reinforcement learning. In *Proceedings of the 10th International Conference on Learning Representations*, 2022.

[49] S. D. Gu, J. G. Kuba, M. N. Wen, R. Q. Chen, Z. Y. Wang, Z. Tian, J. Wang, A. Knoll, Y. D. Yang. Multi-agent constrained policy optimisation. [Online], Available: https://arxiv.org/abs/2110.02793, 2021.

[50] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, Long Beach, USA, pp. 6000–6010, 2017. DOI: 10.5555/3295222.3295349.

[51] I. Sutskever, O. Vinyals, Q. V. Le. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems*, Montreal, Canada, pp. 3104–3112, 2014. DOI: 10.5555/2969033.2969173.

[52] Q. Wang, B. Li, T. Xiao, J. B. Zhu, C. L. Li, D. F. Wong, L. S. Chao. Learning deep transformer models for machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy, pp. 1810–1822, 2019. DOI: 10.18653/v1/P19-1176.

[53] L. H. Dong, S. Xu, B. Xu. Speech-transformer: A no-recurrence sequence-to-sequence model for speech recognition. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, Calgary, Canada, pp. 5884–5888, 2018. DOI: 10.1109/ICASSP.2018.8462506.

[54] K. Han, Y. H. Wang, H. T. Chen, X. H. Chen, J. Y. Guo, Z. H. Liu, Y. H. Tang, A. Xiao, C. J. Xu, Y. X. Xu, Z. H. Yang, Y. M. Zhang, D. C. Tao. A survey on vision transformer. [Online], Available: https://arxiv.org/abs/2012.12556, 2020.

[55] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. H. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proceedings of the 9th International Conference on Learning Representations*, 2020.

[56] R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, vol. 8, no. 3, pp. 229–256, 1992. DOI: 10.1007/BF00992696.

[57] I. Mordatch, P. Abbeel. Emergence of grounded compositional language in multi-agent populations. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence*, New Orleans, USA, Article number 183, 2018. DOI: 10.5555/3504035.3504218.

[58] C. Yu, A. Velu, E. Vinitsky, J. X. Gao, Y. Wang, A. Bayen, Y. Wu. The surprising effectiveness of PPO in cooperative, multi-agent games. [Online], Available: https://arx-

iv.org/abs/2103.01955, 2021.

[59] J. Fu, A. Kumar, O. Nachum, G. Tucker, S. Levine. D4RL: Datasets for deep data-driven reinforcement learning. [Online], Available: https://arxiv.org/abs/2004.07219, 2020.

[60] Z. D. Zhu, K. X. Lin, A. K. Jain, J. Zhou. Transfer learning in deep reinforcement learning: A survey. [Online], Available: https://arxiv.org/abs/2009.07888, 2020.

Linghui Meng received the B.Sc. degree in rail traffic signal control from Beijing Jiao Tong University, China in 2019. He is currently a Ph.D. degree candidate in pattern recognition and intelligent system at both Institute of Automation, Chinese Academy of Sciences, China, and University of Chinese Academy of Sciences, China.

His interests include theoretical research on reinforcement learning, pre-training models, multi-agent system and speech recognition.

E-mail: menglinghui2019@ia.ac.cn

ORCID iD: 0000-0002-5826-8072

Muning Wen is currently a Ph.D. degree candidate in computer science and technology at Shanghai Jiao Tong University, China.

His research interests include reinforcement learning, deep learning and multi-agent system.

E-mail: muning.wen@sjtu.edu.cn

Chenyang Le is an undergraduate in computer science and technology at Shanghai Jiaotong University, China.

His research interests include reinforcement learning, automatic speech recognition and speech translation.

E-mail: nethermanpro@sjtu.edu.cn

Xiyun Li is currently a Ph.D. degree candidate in pattern recognition and intelligent system at Institute of Automation, Chinese Academy of Sciences, China.

His research interests include reinforcement learning and brain-inspired cognitive models.

E-mail: lixiyun2020@ia.ac.cn

Dengpeng Xing received the B.Sc. degree in mechanical electronics and the M.Sc. degree in mechanical manufacturing and automation from Tianjin University, China in 2002 and 2006, respectively, and the Ph.D. degree in control science and engineering from Shanghai Jiao Tong University, China in 2010. He is currently an associate professor in the Research Center for Brain-inspired Intelligence, Institute of Automation, Chinese Academy of Sciences, China.

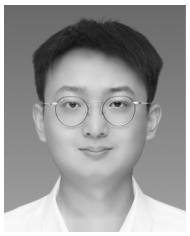
His research interests include reinforcement learning and brain-inspired robotics.

E-mail: dengpeng.xing@ia.ac.cn

Weinan Zhang received the B.Eng. in computer science and technology from ACM Class of Shanghai Jiao Tong University, China in 2011, and the Ph.D. degree in computer science and technology from University College London, UK in 2016. He is now a tenure-track associate professor at Shanghai Jiao Tong University, China. He has published over 150 research papers on international conferences and journals and has been serving as an area chair or (senior) PC member at ICML, NeurIPS, ICLR, KDD, AAAI, IJCAI, SIGIR, etc, and a reviewer at JMLR, TOIS, TKDE, TIST, etc.

His research interests include reinforcement learning, deep learning and data science with various real-world applications of recommender systems, search engines, text mining & generation, knowledge graphs and game AI.

E-mail: wnzhang@sjtu.edu.cn



Ying Wen received the B.Eng. degree from Beijing University of Posts and Telecommunications, China, the B.Eng. degree with first class honor from Queen Mary, University of London, UK in 2015, the M.Sc. degree with distinction honor from University College London, UK in 2016, and the Ph.D. degree from Department of Computer Science, University College London, UK. is currently a tenure-track Assistant Professor with the John Hopcroft Center for Computer Science, Shanghai Jiao Tong University, China. He has published over 20 research papers about machine learning on top-tier international conferences (ICML, NeurIPS, ICLR, IJCAI, and AAMAS). He has been serving as a PC member at ICML, NeurIPS, ICLR, AAAI, IJCAI, ICAPS and a reviewer at TIFS, *Operational Research*, etc. He was granted Best Paper Award in AAMAS 2021 Blue Sky Track and Best System Paper Award in CoRL 2020.

His research interests include machine learning, multi-agent systems and human-centered interactive systems.

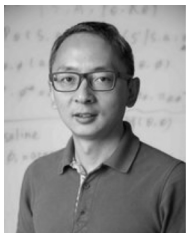
E-mail: ying.wen@sjtu.edu.cn

Haifeng Zhang received the B.Sc. degree in computer science and economics and the Ph.D. degree in computer science from Peking University, China in 2012 and 2018, respectively. He was a visiting scientist at Center on Frontiers of Computing Studies (CFCS), Peking University, China. Earlier, he was a research fellow at University College London, UK. He is an associate professor at Institute of Automation, Chinese Academy of Sciences (CASIA), China.

He has published research papers on international conferences ICML, NeurIPS, AAAI, IJCAI, AAMAS, etc. He has served as a Reviewer for AAAI, IJCAI, TNNLS, *Acta Automatica Sinica*, and Co-Chair for IJCAI competition, IJTCS, DAI Workshop, etc.

His research interests include reinforcement learning, game AI, game theory and computational advertising.

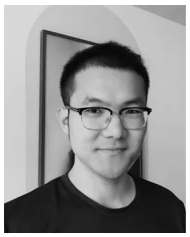
E-mail: haifeng.zhang@ia.ac.cn



Jun Wang received the B. Sc. degree from Southeast University, China in 1997, the M. Sc. degree from National University of Singapore, Singapore in 2003, the Ph. D. degree from Delft University of Technology, The Netherlands in 2007. is currently a professor with Computer Science Department, University College London, UK. He has published over 200 research articles. His team won the First Global Real-Time Bidding Algorithm Contest with more than 80 participants worldwide. He was the winner of multiple best paper awards. He was a recipient of the Beyond Search—Semantic Computing and Internet Economics Award by Microsoft Research and the Yahoo! FREP Faculty Award. He has served as an Area Chair for ACM CIKM and ACM SIGIR. His recent service includes the Co-Chair for Artificial Intelligence, Semantics, and Dialog in ACM SIGIR 2018.

His research interests are in the areas of AI and intelligent systems, covering (multiagent) reinforcement learning, deep generative models, and their diverse applications on information retrieval, recommender systems and personalization, data mining, smart cities, bot planning, and computational advertising.

E-mail: jun.wang@cs.ucl.ac.uk (Corresponding author)
ORCID iD: 0000-0001-9006-7951



Yaodong Yang received the B. Sc. degree in electronic engineering & information science from University of Science and Technology of China, China in 2013, the M.Sc. degree in science (Quant. Biology/ Biostatistics) degree from Imperial College London, UK 2014, and the Ph.D. degree in computer science from University College London, UK in 2021. He is a machine learning researcher with ten-year working experience in both academia and industry. Currently, he is an assistant professor at Peking University, China. Before joining Peking University, he was an assistant professor at King′s College London, UK. Before KCL, he was a principal research scientist at Huawei UK. Before Huawei, he was a senior research manager at AIG, working on AI applications in finance. He has maintained a track record of more than forty publications at top conferences and journals, along with the Best System Paper Award at CoRL 2020 and the Best Blue-sky Paper Award at AAMAS 2021.

His research interests include reinforcement learning and multi-agent systems.

E-mail: yaodong.yang@pku.edu.cn

Bo Xu received the B. Sc. degree in electrical engineering from Zhejiang University, China in 1988, and the M. Sc. and Ph. D. degrees in pattern recognition and intelligent system from Institute of Automation, Chinese Academy of Sciences, China in 1992 and 1997, respectively. He is a professor, the director of Institute of Automation, Chinese Academy of Sciences, China, and also deputy director of Center for Excellence in Brain Science and Intelligence Technology, Chinese Academy of Sciences, China.

His research interests include brain-inspired intelligence, brain-inspired cognitive models, natural language processing and understanding, and brain-inspired robotics.

E-mail: xubo@ia.ac.cn (Corresponding author)
ORCID iD: 0000-0002-1111-1529