Agent-Aware Training for Agent-Agnostic Action Advising in Deep Reinforcement Learning

Yaoquan Wei, Shunyu Liu*, Jie Song, Tongya Zheng, Kaixuan Chen, Mingli Song

State Key Laboratory of Blockchain and Data Security, Zhejiang University
{yaoquanwei,liushunyu,sjie,chenkx,brooksong}@zju.edu.cn, doujiang_zheng@163.com

Abstract

Action advising endeavors to leverage supplementary guidance from expert teachers to alleviate the issue of sampling inefficiency in Deep Reinforcement Learning (DRL). Previous agent-specific action advising methods are hindered by imperfections in the agent itself, while agent-agnostic approaches exhibit limited adaptability to the learning agent. In this study, we propose a novel framework called *Agent-Aware trAining yet Agent-Agnostic Action Advising* (A7) to strike a balance between the two. The underlying concept of A7 revolves around utilizing the similarity of state features as an indicator for soliciting advice. However, unlike prior methodologies, the measurement of state feature similarity is performed by neither the error-prone learning agent nor the agent-agnostic advisor. Instead, we employ a proxy model to extract state features that are both discriminative (adaptive to the agent) and generally applicable (robust to agent noise). Furthermore, we utilize behavior cloning to train a model for reusing advice and introduce an intrinsic reward for the advised samples to incentivize the utilization of expert guidance. Experiments are conducted on the GridWorld, LunarLander, and six prominent scenarios from Atari games. The results demonstrate that A7 significantly accelerates the learning process and surpasses existing methods (both agent-specific and agent-agnostic) by a substantial margin.

Introduction

Deep Reinforcement Learning (DRL) has emerged as a well-established paradigm for addressing sequential decision-making tasks (Mnih et al. 2013; Barto, Sutton, and Anderson 2020; Jiang, Xie, and Yang 2021; Jiang et al. 2022; Liu et al. 2024b) spanning across diverse practical domains, including video games (Vinyals et al. 2019; Ye et al. 2020), robotics (Sangiovanni et al. 2018; Andrychowicz et al. 2020), auto-driving (Chen, Yuan, and Tomizuka 2019; Kiran et al. 2021), industrial control (Zhou et al. 2020; Yang et al. 2020; Sharma et al. 2021; Liu et al. 2024a) *etc*. DRL necessitates the agent's acquisition of knowledge through trial and error, enabling them to adapt and enhance their performance by interacting with the environment. However, a formidable challenge within the realm of DRL lies in sampling inefficiency (Yarats et al. 2021; Ye et al. 2022), as the agent must engage in numerous interactions with the environment in order to acquire a promising policy.

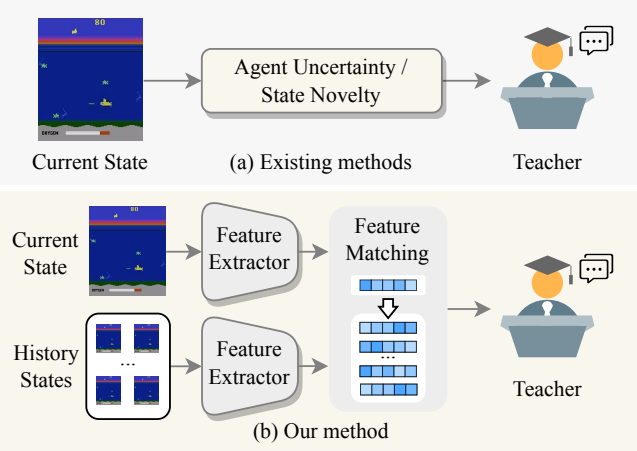


Figure 1: Comparing our method with the existing methods for action advising. (a) Existing methods rely on estimating the uncertainty or novelty for the current state to seek teacher advice. (b) Our method conducts feature matching to seek teacher advice, which considers the relationship between the current state and history states.

To this date, there has been a remarkable amount of research effort to overcome the sampling inefficiency with the aid of online expert feedback, including *action-based advice* (Torrey and Taylor 2013; Silva et al. 2020; Liu et al. 2023), *preference-based evaluation* (Christiano et al. 2017; Lee, Smith, and Abbeel 2021; OpenAI 2023), and *language-based instruction* (Goyal, Niekum, and Mooney 2019; Zhou and Small 2021). Among the diverse approaches, action advising is recently gaining increasing attention as a straightforward yet compelling solution for its more accurate guidance on the policy. Nevertheless, the inherent nature of continuous interactions in action advising inevitably burdens the expert with substantial communications. Hence, the agent must judiciously determine when to seek guidance and effectively leverage the limited resources of expert advice.

Existing action advising approaches determine whether or not acquire action advice from the expert by evaluating the agent uncertainty or novelty of the current state, as depicted in Figure 1(a), which can be broadly classified into

*Corresponding author.

two categories: *agent-specific* methods and *agent-agnostic* methods. Agent-specific methods (Silva et al. 2020; İlhan et al. 2022) hinge upon the agent's inherent uncertainty on the current state to solicit advice from teachers, thus exhibiting enhanced adaptability to the agent policy. Albeit effective in certain scenarios, the uncertainty estimation is easily misled by the agent's own imperfections, consequently resulting in inadequate coverage of the advised state space. In contrast, agent-agnostic methods (Ilhan, Gow, and Perez-Liebana 2019; Ilhan, Gow, and Perez 2021) assess the novelty of the state from the viewpoint of the demonstrator or others, irrespective of the agent's policy. This circumvents the issues stemming from an imperfect agent (especially during the early stage of training) yet leads to wastage of advice in states where the agent has already gained sufficient experience.

In this work, we endeavor to amalgamate the advantages of both approaches. We present an innovative framework called *Agent-Aware trAining yet Agent-Agnostic Action Advising* (A7) for predicting state novelty, as depicted in Figure 1(b). The fundamental concept of A7 revolves around utilizing the similarity of state features as an indicator for seeking advice. However, unlike previous methodologies, the measurement of state feature similarity is not performed by the error-prone learning agent or the agent-agnostic advisor. We employ a proxy model (Feature Extractor) to extract state features that are both discriminative (i.e., adaptive to the agent) and generally applicable (i.e., robust to agent noise). Taking inspiration from the contrastive method BYOL (Grill et al. 2020), we tailor a contrastive learning approach called action-BYOL to train the proxy feature extractor by contrasting the current state with the subsequent state following the agent's policy (in which sense we call it *agent-aware training*). Upon encountering a new state, action-BYOL extracts its features and conducts feature matching (compares them with those from historical states), based on which an advice query is sent to an external expert (in which sense we call it *agent-agnostic action advising*). Additionally, we employ behavior cloning to train a model for reusing advice and introduce an intrinsic reward for the advised samples to incentivize the exploitation of expert guidance. To summarize, A7 offers several advantages over state-of-the-art approaches in action advising:

- In contrast to prior agent-specific methods such as RCMP (Silva et al. 2020) and SUA-AIR (İlhan et al. 2022), A7 employs a self-supervised learning strategy to acquire generally applicable state features. This agent-agnostic approach reduces sensitivity to imperfections in the learning agent.
- In comparison to prior agent-agnostic methods like SNA (Ilhan, Gow, and Perez-Liebana 2019) and ANA (Ilhan, Gow, and Perez 2021), A7 leverages the proposed action-BYOL to extract state features. This method trains the feature extractor by contrasting the current state with the next state following the agent's policy, resulting in more discriminative features for identifying novel states.

Experiments conducted on various scenarios demonstrate that the proposed A7 framework significantly accelerates the learning process and surpasses existing methods.

Related Work

To overcome the sampling inefficiency problem in DRL, learning from human feedback has attracted much attention in the academic field in recent years, where human feedback can be roughly divided into *action-based advice* (Arora and Doshi 2021; Liu et al. 2023), *preference-based evaluation* (Christiano et al. 2017; OpenAI 2023), and *language-based instruction* (Goyal, Niekum, and Mooney 2019; Zhou and Small 2021). Christiano et al. first scaled preference-based learning to utilize modern deep learning techniques while Lee et al. proposed a feedback-efficient RL algorithm by utilizing off-policy learning and pre-training for preference-based methods recently. Toro Icarte et al. (Toro Icarte et al. 2018) utilize natural language advice (e.g., "Turn out the lights before you leave the office" or "Always alleviate potholes in the road"), which can recommend regarding behavior to guide the exploration of the RL agent.

Compared with the low discriminability of preference-based evaluation and the semantic ambiguity of language-based instruction, we are interested in action-based advice methods, also called action advising, which provides much more accurate guidance to the agent. At the heart of action-advising methods is how to determine the optimal timing for the student agent to solicit action advice from the teacher model (a pre-trained model or an expert). *Agent-specific* and *agent-agnostic* methods have dominated the two mainstream branches of action advising, which both assess the advice significance based on the uncertainty of the current state. On the one hand, agent-specific methods evaluate the agent-level uncertainty based on the current state from the agent network. Torrey and Taylor (2013) initially estimated the uncertainty of the teacher agent by considering its Q-value and sought advice upon high uncertainty. In contrast, Silva et al. (2020) calculated the uncertainty from the viewpoint of the student agent based on an multi-head attention network employed by Bootstrapped DQN (Osband et al. 2016). Liu et al. (2023) additionally employed the value loss as a measure of state uncertainty. İlhan et al. (2022) further calculated the uncertainty by utilizing a twin network with dropout to mitigate interference from the original network.

On the other hand, agent-agnostic methods (Ilhan, Gow, and Perez-Liebana 2019; Ilhan, Gow, and Perez 2021) evaluate state-level uncertainty based on the global states beyond the limited viewpoint of a single agent. Ilhan, Gow, and Perez (2021) measured the novelty of a piece of advice based on Random Network Distillation (RND) and only updated RND for the advised states. Albeit effective of existing action-advising approaches, we are motivated to bridge the advantages of agent-specific and agent-agnostic methods in this paper to advance the utility of expert feedback. Torrey and Taylor (2013) introduced a teacher uncertainty method that uses the Q-value of the teacher agent to decide when to get the advice.

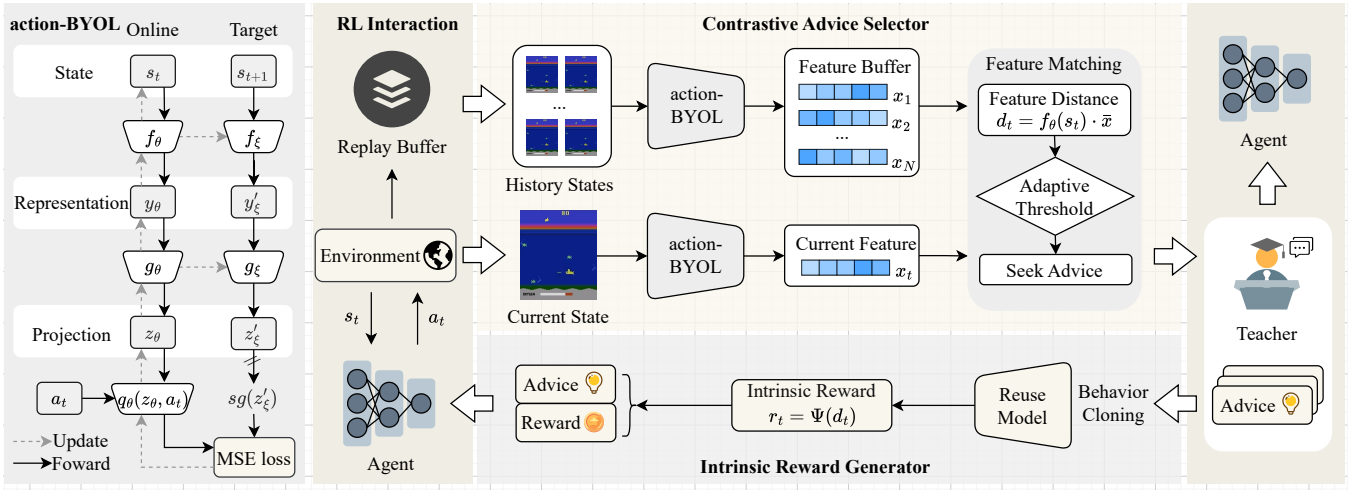


Figure 2: **Left:** an illustrative diagram of action-BYOL, which minimizes a similarity loss between $q_\theta(z_\theta, a_t)$ and $sg(z'_\xi)$. $sg(\cdot)$ means stop-gradient operation. **Right:** an illustrative diagram of the proposed A7 framework, comprising two key components: the contrastive advice selector and the intrinsic reward generator.

Method

In this work, we focus on the action advising problem for the control tasks with the discrete action space under the Markov Decision Process (MDP). In the framework of action advising in DRL, a student agent π_S can seek action advice from the expert teacher π_T to learn an effective policy. Then the expert teacher will return the action advice $\tilde{a}_t = \pi_T(s_t)$ based on the current state s_t. Specifically, the advice budget N is often limited due to resource constraints. We adopt Dueling DQN (Wang et al. 2016) as the backbone of all compared methods to ensure comparability.

In what follows, we detail the proposed A7 framework. As shown in Figure 2, A7 comprises two complementary components: the contrastive advice selector and the intrinsic reward generator. The contrastive advice selector employs a proxy model called action-BYOL, which is trained with the states experienced by the agent (agent-aware) to extract relevant state features. Then, the selector only uses the similarity between state features to identify an appropriate state for seeking advice, regardless of the agent (agent-agnostic). Moreover, the intrinsic reward generator collects the state-advice pairs chosen by the selector for reuse. It also introduces additional intrinsic rewards for advised samples (advised by the teacher and reuse model) to incentivize the exploitation of expert guidance. With these two components, A7 can accelerate the agent learning process and improve sampling efficiency.

Contrastive Advice Selector

To integrate the benefits of existing methods, encompassing adaptability to agent behavior and robustness to agent noise, we adopt the similarity among state features as an indicator for seeking action advice. This necessitates the effective extraction of state features in our approach. To achieve this, we employ the contrastive learning method, BYOL (Grill et al. 2020), to train the feature extractor using states experienced by the agent, which can be referred to as agent-aware training. However, the similarity calculation for action advising is performed without considering the specific agent, which can be referred to as agent-agnostic. Moreover, considering the temporal relationships between states, we introduce modifications to the BYOL method. When an agent encounters a state and takes an action, it transitions to the next state. Two consecutive states are usually similar. Also, the selected action can provide transitional information between the current state and the subsequent state. To fully leverage this information, we incorporate the current state, selected action, and next state into the contrastive learning process instead of employing simple augmentations. We term this modified model as action-BYOL. It is also worth noting that the hidden layer of the agent network also has the potential for feature extraction. Nonetheless, the continuous update of network parameters, along with incomplete initial network training, limits its effectiveness in representing the relationship between states. Therefore, we adopt a separate pre-trained action-BYOL as a feature extractor.

The action-BYOL model consists of the *online* and *target* networks, as depicted in the left part of Figure 2. The online network with parameters θ takes the current state s_t as input and outputs the representation $x_\theta \triangleq f_\theta(s_t)$, as well as the projection $z_\theta \triangleq g_\theta(x_\theta)$. The target network with parameters ξ takes the next state s_{t+1} as input and outputs the target representation $x'_\xi \triangleq f_\xi(s_{t+1})$, as well as the target projection $z'_\xi \triangleq g_\xi(x'_\xi)$. Moreover, we further output a predictor $q_\theta(z_\theta, a_t)$, which takes the selected action a_t and the projection z_θ as input. Note that the predictor is only applied to the online network. We normalize $q_\theta(z_\theta, a_t)$ and z'_ξ to $\bar{q}_\theta(z_\theta, a_t) \triangleq q_\theta(z_\theta, a_t)/\|q_\theta(z_\theta, a_t)\|_2$ and $\bar{z}'_\xi \triangleq z'_\xi/\|z'_\xi\|_2$. Finally, the similarity loss function (Grill et al. 2020) be-

tween the predictions and target projections is defined as:

$$\mathcal{L}_C = \|\bar{q_\theta}(z_\theta, a_t) - \bar{z}'_\xi\|_2^2 = 1 - \frac{\langle q_\theta(z_\theta, a_t), z'_\xi \rangle}{\left\|q_\theta(z_\theta, a_t)\right\|_2 \cdot \left\|z'_\xi\right\|_2}. \quad (1)$$

The optimization is performed to minimize $\mathcal{L}_C$ with respect to the online parameters θ only, while the target parameter ξ is updated slowly by the online parameters via $\xi \leftarrow \tau\xi + (1-\tau)\theta$ with the target decay rate $\tau \in [0, 1]$.

To facilitate the learning of sample features, we train the action-BYOL model periodically until the budget is exhausted. However, the agent often fails within a few steps during the early stage of training, resulting in the collection of similar samples that hinder the learning process of action-BYOL. Therefore, to acquire diverse samples, we allow the agent to continuously seek advice from the teacher at the beginning. After each training stage of action-BYOL, we only retain the encoder f_θ and use the representation output as the state feature. The features of all experienced states are stored in a feature buffer $\mathcal{D}_f$. At each time step t, we extract the agent-agnostic feature from the current state and calculate the average cosine distance between the current feature and the stored features as the indicator for seeking advice:

$$d_t = \Phi(s_t, \mathcal{D}_f) = \frac{\sum_{j=1}^{M} x_j \cdot f_\theta(s_t)}{M}, \quad (2)$$

where s_t is the current state, $x_j \in \mathcal{D}_f$ is the stored feature in the feature buffer, and M is the buffer size. Training an agent involves an extensive process of interactive learning with substantial samples. Hence, it is essential to evaluate the overall relationships among the samples. However, it is not practical to store all state features in a buffer due to the memory overhead and computational speed limitations. Therefore, we transform Equation (2) into:

$$d_t = f_\theta(s_t) \cdot \bar{x}, \text{ where } \bar{x} = \frac{\sum_{j=1}^{M} x_j}{M} \text{ and } x_j \in \mathcal{D}_f. \quad (3)$$

It is easy to devise incremental formulas for updating the average feature $\bar{x}$ with low computational cost. Thus, it suffices to store a single average feature $\bar{x}$, eliminating the need to store all individual state features.

During training, the agent will seek the expert teacher for advice if the feature distance d_t of the current state s_t exceeds a threshold σ. Additionally, the current state feature $x_t = f_\theta(s_t)$ will be used for updating the average feature $\bar{x}$. However, it is challenging to determine a fixed distance threshold σ for different environments with various feature spaces. Therefore, to address the necessity of tuning the distance threshold for each environment, we employ an adaptive distance threshold. At each time step t, we add the distance d_t to a fixed-length queue, denoted as $\mathcal{H}$. If the queue is not yet full, the agent can continuously seek advice from the teacher. Once the queue $\mathcal{H}$ reaches its maximum length, we sort the queue $\mathcal{H}$ incrementally and use the percentile value of $\mathcal{H}$ as our adaptive threshold for subsequent steps. Although the length of the queue and the percentile value still require configuration, they can be universally applied across all environments.

Intrinsic Reward Generator

To encourage the exploitation of expert guidance, we employ behavior cloning to train a reuse model to imitate the expert teacher for action advising. Moreover, additional intrinsic rewards are introduced for each advised sample to train the agent. Specifically, we collect the state-advice pairs generated by the contrastive advice selector. These pairs are then used to train a neural network known as the reuse model using behavior cloning. Behavior cloning approximates the conditional distributions of actions based on the associated state. The reuse model is trained to minimize the negative log-likelihood loss function as:

$$\mathcal{L}_G = \sum_{(s,a)\in\mathcal{T}} -\log G(a|s;\phi), \quad (4)$$

where $\mathcal{T}$ denotes the collected state-advice pairs and ϕ represents the parameter of the reuse model G. Subsequently, by taking the current state as input, the uncertainty of the reuse model G can be calculated. Then we determine whether to provide its output as re-advice to the student based on the level of uncertainty. A smaller level of uncertainty indicates a high alignment between the current state and the training states of the reuse model, enabling the reuse model to deliver the teacher action of that specific state. The uncertainties of all trained state-advice pairs are computed, and the threshold u_r is set as the lower 90% of these uncertainties. When using the reuse model, the first step is to calculate the uncertainty u_s for the current state. If the u_s is lower below the threshold u_r. The resulting output with a deactivated dropout layer is subsequently utilized as advice for the agent. The calculation of u_s have been provided in the appendix. In this way, the student can seek advice from the reuse model G when encountering states that are similar to the advised samples.

Although the agent can directly execute the re-advised actions from the reuse model to the environment for guidance, the standard rewards from the environment are not sufficient for the agent to learn these expert behaviors effectively. To further encourage the exploitation of these re-advised samples, we propose to assign intrinsic rewards to each re-advised sample based on its distance from the feature buffer. Specifically, it is necessary for the agent to learn from hard samples. This implies that samples with greater feature distance require a larger intrinsic reward. Thus, the additional intrinsic reward is designed as follows:

$$\hat{r}_t = \Psi(d_t) = \lambda_t \cdot \tanh\left(\frac{d_t}{d_m}\right), \quad (5)$$

where d_t denotes the feature distance between the current state and the stored features. d_m denotes the average feature distance in the feature buffer, serving as a regularization term. The time decay coefficient λ_t controls the effect of intrinsic rewards and decays over time. The speed of decaying λ_t determines how long the intrinsic rewards will continue to influence the agent policy. Choosing the appropriate decay speed of λ_t can accelerate learning while preventing substantial biases in the policy. In this paper, we adopt a linear decay regime to gradually reduce the value of λ_t from

the initial value λ_0. For advice directly obtained from the teacher, we keep this initial value unchanged. By incorporating advice reuse and leveraging intrinsic rewards, the intrinsic reward generator can enhance the effective utilization of teacher advice and expedite the learning process.

Experiments

To demonstrate the effectiveness of the proposed A7 framework for action advising in DRL, we conduct experiments on the GridWorld (Ilhan, Gow, and Perez 2021), LunarLander (Towers et al. 2023) and six popular scenarios from Atari games in line with the previous works (Silva et al. 2020; Ilhan, Gow, and Perez 2021; Ilhan, Gow, and Liebana 2021; İlhan et al. 2022). In this section, we first introduce the compared methods and the special hyperparameter settings. Then the comparison results are reported and analyzed. Moreover, ablation studies are conducted to investigate the advantages of our A7.

Experimental Settings

We compare A7 with various baselines, including:

1. **Heuristic methods**: *No Advising (NA)*, where the student agent follows its own policy without advice; *Early Advising (EA)*, where the student agent always requests advice until the advice budget is exhausted;*Random Advising (RA)*, where the student agent requests advice with a probability of 50% at every step.
2. **Agent-specific methods**: *Importance-base Action Advising (IAA)* (Torrey and Taylor 2013), where the student agent uses the difference between the maximum and minimum values of the Q-values to calculate uncertainty and requests advice based on a predefined threshold; *Requesting Confident Moderated Policy Advice (RCMP)* (Silva et al. 2020), where the student agent uses multi-head DQN to calculate uncertainty and requests advice based on a predefined threshold; *Student Uncertainty-driven Advising with Advice Imitation & Reuse (SUA-AIR)* (Ilhan, Gow, and Liebana 2021; İlhan et al. 2022), where the student agent requests advice based on an adaptive uncertainty estimation, paired with an imitation model that is using uncertainty thresholds for advice reuse.
3. **Agent-agnostic methods**: *Advice Novelty-Based Advising (ANA)* (Ilhan, Gow, and Perez 2021), where the student agent adopts random network distillation (Burda et al. 2019) to calculate state novelty for action advising.

We adopt Double DQN (Van Hasselt, Guez, and Silver 2018) as the backbone. The decay of λ_t is linear. For GridWorld and LunarLander, it takes 20k steps with inital value of 0.1, while for Atari scenarios, it takes 1M steps with inital value of 0.5. To carry out sufficient experiments, we follow the same teacher setting as previous works (Silva et al. 2020; İlhan et al. 2022) to use a pre-trained model as a teacher. Please refer to Appendix for more details.

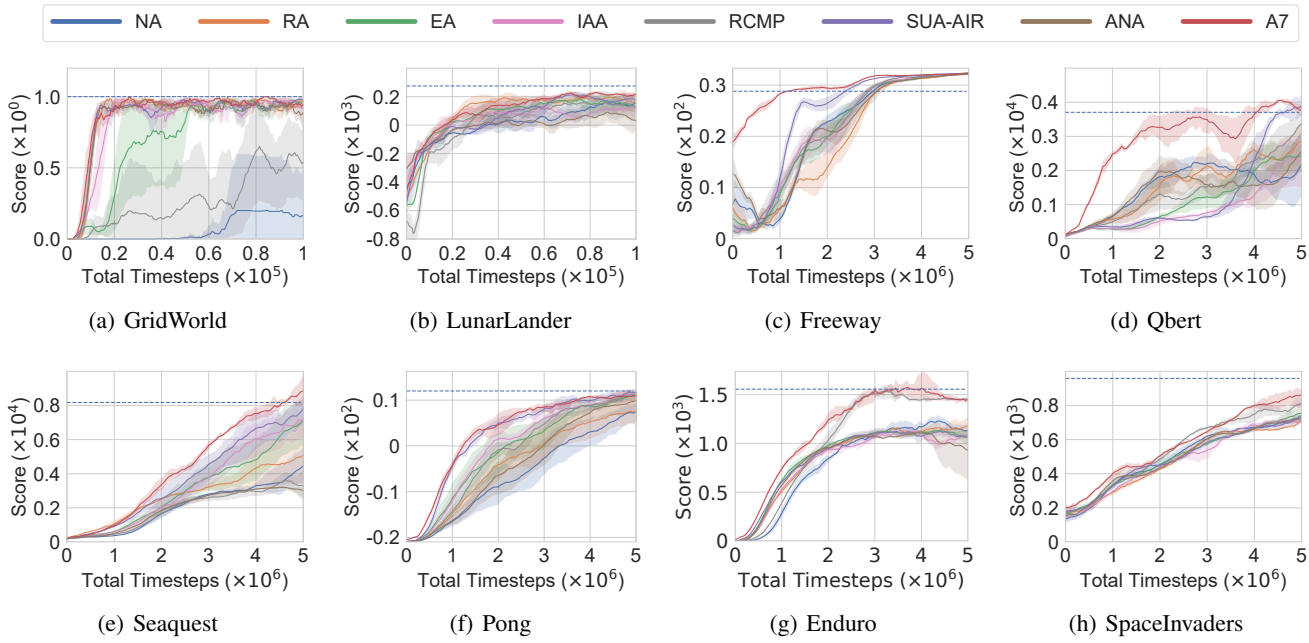
Results and Analysis

The experimental results in various environments compared with the state-of-the-art methods are shown in Figure 3 and Table 1. Specifically, since the curves in Figure 3 represent the scores of the agents at different time steps, we adopt the area under the learning curve (AUC) as an important metric to evaluate the sampling efficiency of different methods, which provides an overall measure of the agent's learning efficiency. In the easy environments (GridWorld and LunarLander), NA often performs poorly, while our proposed A7 can achieve superior performance. Similarly, several baselines, including EA and SUA-AIR, also exhibit promising results in these two environments. Especially in the Freeway, Qbert, and Seaquest scenarios, our proposed A7 method consistently outperforms baselines by a large margin during training. Moreover, A7 has also demonstrated a substantial performance advantage compared to other methods when evaluated using the AUC metric. In the Freeway and Qbert, our method demonstrates a powerful capability to expedite agent learning, achieving scores that are significantly higher than those obtained by other methods during the early stages of training. In the Seaquest, A7 consistently outperforms other methods in terms of scores throughout the entire duration. In the Pong, Enduro, and SpaceInvaders scenarios, A7 maintained its leading position in the overall learning curves. Additionally, when considering the overall AUC, A7 remained the best. In addition, we also listed the best evaluation scores achieved by all methods throughout the training phase in Table 2. It is evident that, in all scenarios except for Pong, our method attained the highest scores, which also validates A7 improves the scores of the agent compared to existing methods. To sum up, the experimental results suggest that our novel framework A7 amalgamates the advantages of agent-specific and agent-agnostic approaches, improving the sampling efficiency and accelerating the agent learning process to achieve non-trivial performance.

Ablation Studies

The contribution of different components To understand the superior performance of A7, we carry out ablation studies to test the contribution of its two main components: contrastive advice selector and intrinsic reward generator. The results are shown in Figure 4. By comparing A7 without contrastive advice selector (replace the advice selection strategy of A7 with EA, and set the intrinsic reward to a fixed value) and without the intrinsic reward generator, we can conclude that neither of them alone can achieve the level of A7. This comparison highlights the effectiveness of our excellent advice selection strategy and the benefits obtained from combining it with the design of intrinsic rewards in achieving excellent results. Additionally, the AUC of both components exceeds that of NA, which demonstrates the effectiveness of both components in accelerating agent training and improving sampling efficiency.

The impact of different advice budgets Moreover, to study the impact of different advice budgets on the performance of A7, we conduct an ablation study as shown in Figure 5. The performance benefit of A7 experiences a substantial increase when the number of advice budgets rises from 5k to 25k. Conversely, the performance of SUA-AIR has shown minimal improvement. This can be attributed to



(a) GridWorld (b) LunarLander (c) Freeway (d) Qbert

(e) Seaquest (f) Pong (g) Enduro (h) SpaceInvaders

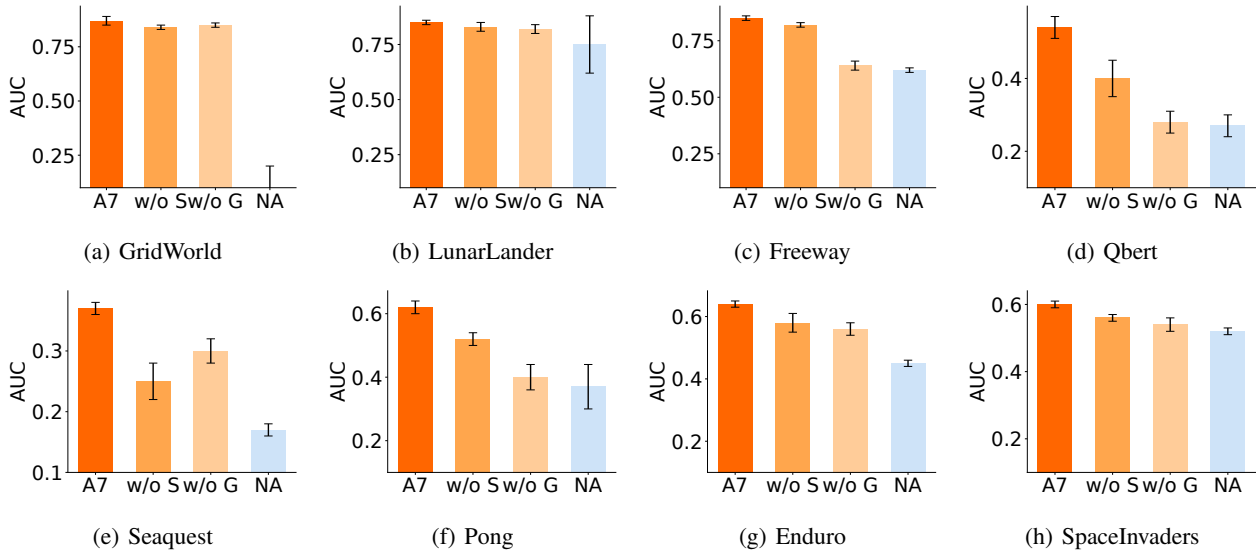
Figure 3: Learning curves of our proposed A7 and baselines on the GridWorld, LunarLander, and six Atari scenarios. All experimental results are illustrated with the mean and the standard deviation of the performance over five random seeds for a fair comparison. The score represents the cumulative reward for a game during evaluation. To make the results in figures clearer for readers, we adopt a 95% confidence interval to plot the error region. Dashed lines represent the operation level of the teachers in different environments.

Method	GridWorld	LunarLander	Freeway	Qbert	Seaquest	Pong	Enduro	SpaceInvaders
NA	0.07 ± 0.13	0.75 ± 0.03	0.62 ± 0.01	0.27 ± 0.03	0.17 ± 0.01	0.37 ± 0.07	0.45 ± 0.01	0.52 ± 0.01
RA	0.68 ± 0.11	0.83 ± 0.03	0.56 ± 0.03	0.28 ± 0.04	0.23 ± 0.04	0.43 ± 0.07	0.47 ± 0.01	0.50 ± 0.01
EA	0.85 ± 0.02	0.80 ± 0.03	0.62 ± 0.03	0.20 ± 0.04	0.25 ± 0.07	0.50 ± 0.07	0.48 ± 0.01	0.53 ± 0.01
IAA	0.81 ± 0.01	0.78 ± 0.04	0.62 ± 0.03	0.15 ± 0.04	0.25 ± 0.07	0.52 ± 0.03	0.48 ± 0.01	0.50 ± 0.01
RCMP	0.27 ± 0.22	0.75 ± 0.06	0.62 ± 0.04	0.26 ± 0.06	0.25 ± 0.07	0.49 ± 0.03	0.56 ± 0.01	0.57 ± 0.01
SUA-AIR	0.85 ± 0.01	0.80 ± 0.02	0.69 ± 0.01	0.22 ± 0.03	0.31 ± 0.06	0.61 ± 0.02	0.48 ± 0.01	0.53 ± 0.01
ANA	0.84 ± 0.01	0.73 ± 0.04	0.62 ± 0.03	0.25 ± 0.03	0.16 ± 0.01	0.43 ± 0.06	0.46 ± 0.03	0.52 ± 0.01
A7	$\mathbf{0.87 \pm 0.02}$	$\mathbf{0.85 \pm 0.01}$	$\mathbf{0.85 \pm 0.01}$	$\mathbf{0.54 \pm 0.03}$	$\mathbf{0.37 \pm 0.01}$	$\mathbf{0.62 \pm 0.03}$	$\mathbf{0.64 \pm 0.01}$	$\mathbf{0.60 \pm 0.01}$

Table 1: Area under the learning curve (AUC) of all compared methods in different environments. $\pm$ corresponds to one standard deviation of the average score over five random seeds. **Bold** indicates the best performance in each environment.

Method	GridWorld	LunarLander	Freeway	Qbert	Seaquest	Pong	Enduro	SpaceInvaders
NA	0.18 ± 0.35	168.52 ± 25.65	32.20 ± 0.11	1992.40 ± 1170.26	4461.56 ± 1234.05	7.61 ± 2.33	1135.19 ± 94.63	727.40 ± 69.29
RA	0.91 ± 0.01	198.45 ± 35.37	32.25 ± 0.08	2930.50 ± 702.73	5145.38 ± 2514.32	7.75 ± 3.64	1179.58 ± 86.84	687.05 ± 35.77
EA	0.89 ± 0.03	185.00 ± 41.75	32.26 ± 0.15	2563.35 ± 491.27	6695.36 ± 1601.95	11.41 ± 1.04	1066.53 ± 105.43	750.70 ± 53.33
IAA	0.86 ± 0.05	105.00 ± 13.26	31.41 ± 0.21	2015.00 ± 573.78	6760.80 ± 1264.99	4.64 ± 2.85	1062.61 ± 168.85	715.50 ± 35.75
RCMP	0.61 ± 0.02	182.12 ± 19.41	32.14 ± 0.25	3233.00 ± 1079.40	2150.80 ± 1166.01	11.48 ± 2.03	1504.75 ± 268.45	875.35 ± 56.83
SUA-AIR	0.94 ± 0.01	224.95 ± 63.25	32.31 ± 0.12	4024.95 ± 606.63	7865.12 ± 1536.11	$\mathbf{12.19 \pm 1.56}$	1077.28 ± 92.76	772.95 ± 76.71
ANA	0.91 ± 0.01	76.32 ± 15.07	32.09 ± 0.05	2382.35 ± 1015.09	2950.78 ± 586.52	9.76 ± 2.37	920.67 ± 324.39	730.60 ± 18.21
A7	$\mathbf{0.95 \pm 0.01}$	$\mathbf{266.35 \pm 34.12}$	$\mathbf{32.36 \pm 0.15}$	$\mathbf{4096.45 \pm 344.81}$	$\mathbf{8692.98 \pm 1009.86}$	11.13 ± 1.67	$\mathbf{1544.83 \pm 77.01}$	$\mathbf{893.18 \pm 59.24}$

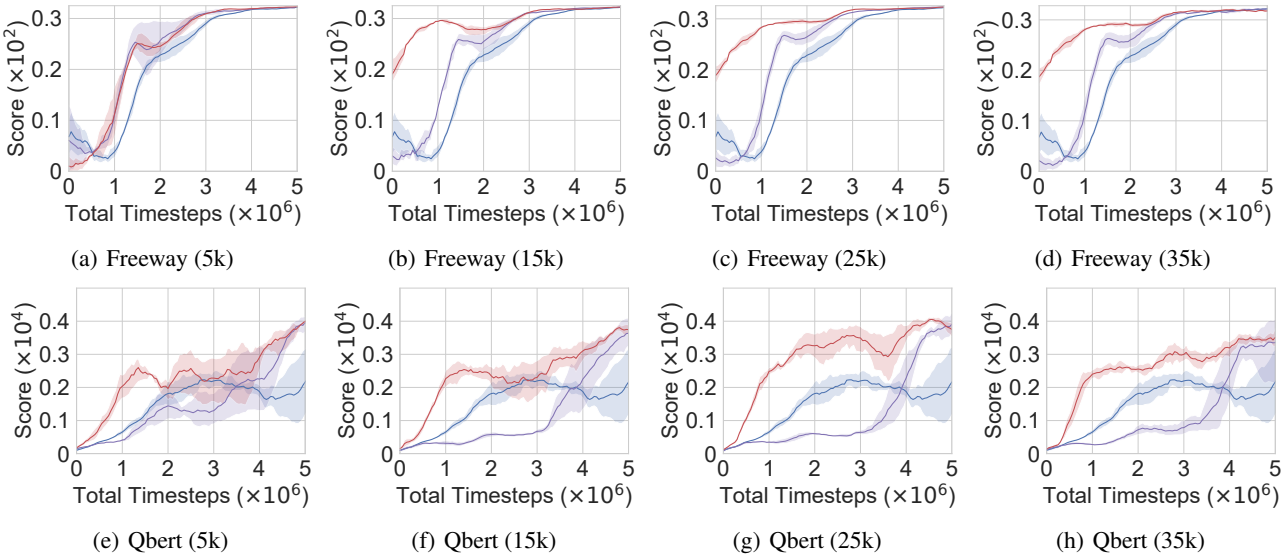
Table 2: Test evaluation scores of all compared methods in different environments. $\pm$ corresponds to one standard deviation of the average score over five random seeds. **Bold** indicates the best performance in each environment.



(a) GridWorld (b) LunarLander (c) Freeway (d) Qbert

(e) Seaquest (f) Pong (g) Enduro (h) SpaceInvaders

Figure 4: Ablation study on the contrastive advice selector (S) and the intrinsic reward generator (G) for six Atari scenarios.



(a) Freeway (5k) (b) Freeway (15k) (c) Freeway (25k) (d) Freeway (35k)

(e) Qbert (5k) (f) Qbert (15k) (g) Qbert (25k) (h) Qbert (35k)

Figure 5: The performance comparison of our proposed A7 (red line) , SUA-AIR (purple line) and No Advising baseline (blue line) under different advice budgets in the Freeway scenario and the Qbert scenario.

the fact that the states selected by A7 for seeking advice can better represent the entire sample space. However, the states selected by SUA-AIR are quite similar, resulting in a lack of improvement in terms of performance. It is also noteworthy that the budget for A7 increased from 15k to 35k in Freeway and 25k to 35k in Qbert, but the consequent growth in performance was minimal. This suggests that once a tipping point is reached, an increase in the number of advice results in a gradual decline in the growth of benefits.

Conclusion

In this work, we propose a novel framework called A7 to alleviate the sampling inefficiency in DRL. A7 amalgamates the advantages of agent-specific and agent-agnostic methods, making it the first dedicated attempt to explicitly build the similarity of state features as the indicator for action advising. Experimental results on different environments show that A7 accelerates the training of agents more effectively and yields significantly high sampling efficiency compared with state-of-the-art competitors. Action advising methods are limited to environments with discrete action spaces. Due to the expansive range of continuous actions, human teachers often struggle to provide precise continuous actions, which can result in unfavorable outcomes when sub-optimal advice is given. Thus, an important future direction lies in the development of action advising methods designed for environments with continuous action spaces, broadening their applications in various practical domains.

Acknowledgments

This work is supported by the Science and Technology Project of SGCC: Hybrid enhancement intelligence with human-AI coordination and its application in reliability analysis of regional power system (5700-202217190A-1-1-ZN).

References

Andrychowicz, O. M.; Baker, B.; Chociej, M.; Jozefowicz, R.; McGrew, B.; Pachocki, J.; Petron, A.; Plappert, M.; Powell, G.; Ray, A.; et al. 2020. Learning dexterous in-hand manipulation. *The International Journal of Robotics Research*.

Arora, S.; and Doshi, P. 2021. A survey of inverse reinforcement learning: Challenges, methods and progress. *Artificial Intelligence*.

Barto, A. G.; Sutton, R. S.; and Anderson, C. W. 2020. Looking back on the actor–critic architecture. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*.

Burda, Y.; Edwards, H.; Storkey, A.; and Klimov, O. 2019. Exploration by random network distillation. In *International Conference on Learning Representations*.

Chen, J.; Yuan, B.; and Tomizuka, M. 2019. Model-free deep reinforcement learning for urban autonomous driving. In *IEEE intelligent transportation systems conference*.

Christiano, P. F.; Leike, J.; Brown, T.; Martic, M.; Legg, S.; and Amodei, D. 2017. Deep reinforcement learning from human preferences. *Conference on Neural Information Processing Systems*, 30.

Goyal, P.; Niekum, S.; and Mooney, R. J. 2019. Using natural language for reward shaping in reinforcement learning. *arXiv preprint arXiv:1903.02020*.

Grill, J.-B.; Strub, F.; Altché, F.; Tallec, C.; Richemond, P.; Buchatskaya, E.; Doersch, C.; Avila Pires, B.; Guo, Z.; Gheshlaghi Azar, M.; et al. 2020. Bootstrap your own latent-a new approach to self-supervised learning. *Annual Conference on Neural Information Processing Systems*.

İlhan, E.; Das, S.; Taylor, M. E.; et al. 2022. Methodical Advice Collection and Reuse in Deep Reinforcement Learning. *arXiv preprint arXiv:2204.07254*.

Ilhan, E.; Gow, J.; and Liebana, D. P. 2021. Action Advising with Advice Imitation in Deep Reinforcement Learning. In *International Conference on Autonomous Agents and Multiagent Systems*.

Ilhan, E.; Gow, J.; and Perez, D. 2021. Student-initiated action advising via advice novelty. *IEEE Transactions on Games*.

Ilhan, E.; Gow, J.; and Perez-Liebana, D. 2019. Teaching on a budget in multi-agent deep reinforcement learning. In *IEEE Conference on Games*.

Jiang, H.; Li, G.; Xie, J.; and Yang, J. 2022. Action Candidate Driven Clipped Double Q-Learning for Discrete and Continuous Action Tasks. *IEEE Transactions on Neural Networks and Learning Systems*.

Jiang, H.; Xie, J.; and Yang, J. 2021. Action candidate based clipped double q-learning for discrete and continuous action tasks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, 7979–7986.

Kiran, B. R.; Sobh, I.; Talpaert, V.; Mannion, P.; Al Sallab, A. A.; Yogamani, S.; and Pérez, P. 2021. Deep reinforcement learning for autonomous driving: A survey. *IEEE Transactions on Intelligent Transportation Systems*.

Lee, K.; Smith, L.; and Abbeel, P. 2021. Pebble: Feedback-efficient interactive reinforcement learning via relabeling experience and unsupervised pre-training. *arXiv preprint arXiv:2106.05091*.

Lee, K.; Smith, L.; Dragan, A.; and Abbeel, P. 2021. B-pref: Benchmarking preference-based reinforcement learning. *arXiv preprint arXiv:2111.03026*.

Liu, S.; Chen, K.; Yu, N.; Song, J.; Feng, Z.; and Song, M. 2023. Ask-ac: An initiative advisor-in-the-loop actor–critic framework. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*.

Liu, S.; Luo, W.; Zhou, Y.; Chen, K.; Zhang, Q.; Xu, H.; Guo, Q.; and Song, M. 2024a. Transmission Interface Power Flow Adjustment: A Deep Reinforcement Learning Approach Based on Multi-Task Attribution Map. *IEEE Transactions on Power Systems*, 39(2): 3324–3335.

Liu, S.; Song, J.; Zhou, Y.; Yu, N.; Chen, K.; Feng, Z.; and Song, M. 2024b. Interaction Pattern Disentangling for Multi-Agent Reinforcement Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12): 8157–8172.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Graves, A.; Antonoglou, I.; Wierstra, D.; and Riedmiller, M. 2013. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.

OpenAI. 2023. GPT-4 Technical Report. *arXiv preprint arXiv.2303.08774*.

Osband, I.; Blundell, C.; Pritzel, A.; and Van Roy, B. 2016. Deep exploration via bootstrapped DQN. In *Conference on Neural Information Processing Systems*.

Sangiovanni, B.; Rendiniello, A.; Incremona, G. P.; Ferrara, A.; and Piastra, M. 2018. Deep reinforcement learning for collision avoidance of robotic manipulators. In *European Control Conference*.

Sharma, J.; Andersen, P.-A.; Granmo, O.-C.; and Goodwin, M. 2021. Deep Q-Learning With Q-Matrix Transfer Learning for Novel Fire Evacuation Environment. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*.

Silva, F. L. D.; Hernandez-Leal, P.; Kartal, B.; and Taylor, M. E. 2020. Uncertainty-Aware Action Advising for Deep Reinforcement Learning Agents. In *AAAI Conference on Artificial Intelligence*.

Toro Icarte, R.; Klassen, T. Q.; Valenzano, R. A.; and McIlraith, S. A. 2018. Advice-based exploration in model-based reinforcement learning. In *Canadian Conference on Artificial Intelligence*.

Torrey, L.; and Taylor, M. E. 2013. Teaching on a budget: agents advising agents in reinforcement learning. In *International conference on Autonomous Agents and Multi-Agent Systems*.

Towers, M.; Terry, J. K.; Kwiatkowski, A.; Balis, J. U.; Cola, G. d.; Deleu, T.; Goulão, M.; Kallinteris, A.; KG, A.; Krimmel, M.; Perez-Vicente, R.; Pierré, A.; Schulhoff, S.; Tai, J. J.; Shen, A. T. J.; and Younis, O. G. 2023. Gymnasium.

Van Hasselt, H.; Guez, A.; and Silver, D. 2018. Deep reinforcement learning with double q-learning. In *AAAI Conference on Artificial Intelligence*.

Vinyals, O.; Babuschkin, I.; Czarnecki, W. M.; Mathieu, M.; Dudzik, A.; Chung, J.; Choi, D. H.; Powell, R.; Ewalds, T.; Georgiev, P.; et al. 2019. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature*.

Wang, Z.; Schaul, T.; Hessel, M.; Hasselt, H.; Lanctot, M.; and Freitas, N. 2016. Dueling network architectures for deep reinforcement learning. In *International Conference on Machine Learning*.

Yang, L.; Sun, Q.; Zhang, N.; and Liu, Z. 2020. Optimal energy operation strategy for we-energy of energy internet based on hybrid reinforcement learning with human-in-the-loop. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*.

Yarats, D.; Zhang, A.; Kostrikov, I.; Amos, B.; Pineau, J.; and Fergus, R. 2021. Improving sample efficiency in model-free reinforcement learning from images. In *AAAI Conference on Artificial Intelligence*.

Ye, D.; Liu, Z.; Sun, M.; Shi, B.; Zhao, P.; Wu, H.; Yu, H.; Yang, S.; Wu, X.; Guo, Q.; et al. 2020. Mastering complex control in moba games with deep reinforcement learning. In *AAAI Conference on Artificial Intelligence*.

Ye, Z.; Chen, Y.; Jiang, X.; Song, G.; Yang, B.; and Fan, S. 2022. Improving sample efficiency in Multi-Agent Actor-Critic methods. *Applied Intelligence*.

Zhou, K.; Song, S.; Xue, A.; You, K.; and Wu, H. 2020. Smart train operation algorithms based on expert knowledge and reinforcement learning. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*.

Zhou, L.; and Small, K. 2021. Inverse reinforcement learning with natural language goals. In *AAAI Conference on Artificial Intelligence*.