2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

The Effectiveness of State Representation Model in Multi-Agent Proximal Policy Optimization for Multi-Agent Path Finding

Jaehoon Chung[1], Jamil Fayyad[1], Mehran Ghafarian Tamizi[1] and Homayoun Najjaran[1,*]

Abstract— **Multi-agent pathfinding plays a crucial role in various robot applications. Recently, deep reinforcement learning methods have been adopted to solve large-scale planning problems in a decentralized manner. Nonetheless, such approaches pose challenges such as non-stationarity and partial observability. In this paper, we address these challenges by integrating a state representation model into a multi-agent proximal policy optimization framework. To do so, we propose to utilize a state representation model which extracts representation features from the global map and leverages this information to enhance the training process. Our approach involves decoupling the feature extractor from the agent training process, enabling a more accurate representation of the global state that remains unbiased by the actions of the agents. Furthermore, our modularized approach offers the flexibility to replace the representation model with another model or modify tasks within the global map, without the retraining of the agents. We demonstrated the effectiveness of our approach by comparing three multi-agent proximal policy optimization frameworks. Our experimental results demonstrate that our approach improves the average episode reward compared to the other approaches.**

I. INTRODUCTION

Deep reinforcement learning (DRL) is a framework where a reinforcement learning (RL) agent learns optimal actions by iteratively exploring and exploiting an interactive environment, leveraging deep learning (DL) models. Notably, DRL has made substantial strides in mastering complex domains in sequential decision-making tasks and control challenges [1]. These advancements have spurred considerable interest within research communities, particularly in developing autonomous models for scenarios involving multiple agents, often mirroring real-world industrial complexities. It has led multi-agent deep reinforcement learning (MADRL) to become a focal point of rigorous research area. In many cases, multi-agent pathfinding (MAPF) emerges as a crucial initial step towards solving collective tasks [2], establishing itself as a thriving subfield within MADRL research. MAPF addresses the challenge of coordinating multiple agents to navigate collision-free paths towards their respective goal points [3]

One of the primary challenges in solving MAPF is the scalability issues, particularly concerning large maps and a substantial number of agents [4]. A growing number of agents normally entails an exponential increase in computational memory and time, resulting in resource-intensive models and practical hurdles for real-world applications. Recent studies have sought to address this issue through the adoption of distributed or decentralized approaches, wherein agents make decisions based on individual observations [5]–[7]. The decentralized execution, inherent to MADRL approaches, facilitates the dynamic scaling of the number of agents – an advantage lacking in centralized executions with fixed input-output dimensions. However, decentralized approaches inevitably bring about non-stationarity issues and limitations in accessing complete environmental information due to partial observability. Non-stationarity arises from the dynamic nature of the environment, where updates from other agents constantly change the environment [8]. Hence, the predictions based on previous experiences lie less reliable over time. On the other hand, partial observability refers to each agent's limited observation range, restricting direct access to all relevant environmental states necessary for decision-making [9].

One of the efforts to mitigate non-stationarity in MAPF is fostering communication among agents. DHC [6] achieved this by integrating graph convolutional communication blocks and guided RL, successfully enhancing the stationary. Their subsequent work further improved communication by minimizing redundant information in broadcast communication, ensuring agents access only pertinent data for decision-making. Chen et al [7] addressed non-stationarity in MAPF without explicit communication but by capturing implicit collaborative information between agents. Similarly, PICO [10] integrated learned implicit planning priorities with a communication learning scheme and demonstrated its effectiveness.

Another challenge in MAPF posed by partial observability has been addressed through the centralized training and decentralized execution (CTDE) scheme [11]. In this approach, agents access the environmental state representation and team rewards during the training phase, whereas the learned policies rely solely on each agent's local observation during the execution phase. This framework reduces the likelihood of agents making undesirable selfish behaviors, fostering agents to align with team objectives or collaborative rewards. However, since the full state of the environment is normally not known in most industrial settings, the environmental state is represented by aggregation of local observations from the agents. Hence, the effectiveness of the CTDE scheme depends on the accurate representation of the environmental state.

Specifically, the state-representation studies in multi-agent proximal policy optimization (MAPPO) have empirically

* Corresponding Author: Homayoun Najjaran, email: (najjaran@uvic.ca)
[1] Department of Mechanical Engineering, University of Victoria, 800 Finnerty Road, Victoria, V8P 5C2, BC, Canada

demonstrated the efficacy of training the critic network with agent-specific global state (ASGS) representations [12]. Inspired by this research, we propose a state representation model where it reconstructs a global map image from the set of local observations and leverages it to extract spatial context and inform it in the CTDE scheme of the MAPPO framework. Given that the entire dimension of aggregated local observations varies with the number of agents, model-driven feature extraction can maintain a consistent dimensional state irrespective of varying numbers of agents. We investigate the impact of introducing this model and outline our main contributions as follows.

- We proposed decoupling the feature extractor network from the agent training framework. This allows updating the feature extractor or modifying the task without retraining the actor-critic networks.
- We generated a global map dataset by utilizing local observations of agents in a partially observable environment. The dataset is collected by rolling out diverse scenarios with a varying number of agents.
- We evaluated the effectiveness of our approach through a wide range of experiments, including a comparison with two other MAPPO frameworks.

II. BACKGROUND AND RELATED WORK

A. Deep Learning-based Multi-Agent Path Finding

Numerous approaches in the domain of MADRL or DRL have been deployed to effectively tackle the challenges posed by large-scale MAPF problems by enabling communication or coordination among agents in MAPF. One of the pioneering works in learning-based MAPF is PRIMAL [13], which has inspired many subsequent studies aiming to develop solutions for large-scale MAPF. PRIMAL was improved by designing agents to learn conventional behaviors that enhance implicit agent coordination and was extended to address lifelong MAPF (LMAPF) scenarios [14]. Wang et al [15] introduced a transformer-based communication learning mechanism into PRIMAL to mitigate conflicting messages in highly-scaled MAPF and foster agent cooperation. Another noteworthy extension of PRIMAL is ALPHA [16], where a graph transformer framework allows agents to access fuzzy global information, thus facilitating proactive policies for agents. This approach demonstrated enhanced agent coordination when global information is made accessible to agents, resulting in significantly reduced episode lengths.

B. Proximal Policy Optimization

Proximal Policy Optimization (PPO) [17] stands out as an advanced policy-gradient method, serving as an on-policy DRL algorithm that learns from actions taken within the current policy. It has demonstrated its empirical competitiveness against other state-of-the-art DRL methods, coupled with its simplicity of implementation. Instead of imposing a hard constraint, PPO utilizes clipping of the objective function and introduces importance sampling to evaluate a new policy using samples collected from the original policy. Subsequently, after a certain number of iterations, it updates the original policy to the new policy. Algorithm 1 presents the pseudocode details of PPO.

Algorithm 1 PPO with Clipped Objective Function

Require: initial actor parameters ω_0, initial critic parameters θ_0, clipping threshold ϵ,

for $k = 0, 1, 2, \ldots$ **do**

Collect set of partial trajectories $\mathcal{D}_k = \{\tau_i\}$ on policy $\pi_k = \pi(\omega_k)$

Derive reward R_t

Estimate advantages $\mathrm{A}^{(t)}_{\omega_k}$ using any advantage estimation algorithm based on the current value function $V_{\theta k}$

Compute policy update:

$\omega_{k+1} = \mathrm{argmax}_\omega \; \mathcal{L}_{\omega_k}(\omega) - \beta_k \cdot D_{KL}(\omega||\omega_k)$

by taking K steps of minibatch SGD (via Adam), where

$$\mathcal{L}_{\omega_k}(\omega) = \mathbb{E}_{\tau\sim\pi_{\omega_k}}\Bigg[\sum_{t=0}^{T}\Bigg[\min\Big(\frac{\pi_\omega(a_t|s_t)}{\pi_{\omega_k}(a_t|s_t)}\cdot \mathrm{A}^{(t)}_{\omega_k}, clip\Big(\frac{\pi_\omega(a_t|s_t)}{\pi_{\omega_k}(a_t|s_t)}, 1-\epsilon, 1+\epsilon\Big)\cdot \mathrm{A}^{(t)}_{\omega_k}\Big)\Bigg]\Bigg]$$

Compute value estimate function update:

$\theta_{k+1} = \mathrm{argmin}_\theta \; \mathbb{E}_{\tau\sim\pi_{\omega_k}}\big(V_\theta(s_t) - R_t\big)^2$

end for

PPO can be effectively extended to MAPPO through CTDE paradigm with improved performance in multi-agent settings. Here, the critic-network V_θ maps the global state to the return: $S \to \mathbb{R}$. The actor-network π_ω, on the other hand, maps agent observation $o_t^{(a)}$ to a distribution over actions in action spaces. In discrete spaces, the network outputs a categorical distribution among the action spaces. In continuous space, the network outputs the mean and standard deviation vectors of a Multivariate Gaussian Distribution, allowing actions to be sampled from it.

C. State Representation Model

Model-based methods have been designed to address sample inefficiency issues in DRL. $World\ model$ is one of the prominent model-based approaches where a visual model learns image features in an unsupervised manner to aid the decision-making of a controller by informing extracted features. Inspired by the human internal perception model's impact on decision-making, Ha et al [18] introduced a Variational Autoencoder (VAE) that simulates learning in imagination, acting as an agent internal perception model of the RL environment. This VAE converts the state provided by the environment into a compressed spatial and temporal latent context. The concept of learning in imagination through a $world\ model$ has led to successful extensions in various domains, such as SimPLe [19], IRIS [20], and DreamerV3 [1].

In our work, we propose to encode the global map where the spatial context of the global information is trained in an unsupervised manner. As a result, our approach removes the need for heuristic algorithms, manually driven feature extraction, or topological representation.

III. PROBLEM DESCRIPTION

A. Environmental Setup

While many other MAPF studies utilize the common 2D discrete grid world setup, we depart from convention by employing the Vectorized Multi-Agent Simulator (VMAS) [21]. Designed specifically for a PyTorch-based vectorized 2D physics engine, VMAS offers a continuous world setup that better reflects real-world dynamics. The grid world setup, while widely used, imposes limitations by confining agents' movements to 4 or 8 connected adjacent grids with fixed speeds. Additionally, it does not reflect the law of inertia in robot motion dynamics. Although opting for a continuous environment with physics dynamics poses the agents more challenging tasks of MAPF, we enable agents to exhibit versatile mobility and more realistically simulate real-world scenarios.

VMAS encompasses a variety of challenging multi-robot scenarios, including MAPF, which is accessible through its $navigation$ scenario mode. In our study, we focus on this mode. During each episode, agents are randomly spawned in a 2D continuous space without obstacles, with the map bounded by $(-1, 1)$ on both the x-axis and y-axis. each agent is assigned a radius of 0.03 and equipped with sensory capabilities, enabling it to perceive its current position, velocity, relative displacement to the goal, and the local observations from a lidar sensor. An episode terminates either when all agents successfully reach their respective goals at the end of a time step (success) or when the number of time steps reaches a predefined limit (failure). Agents involved in collisions incur penalties per collision.

B. Observation Representation

In VMAS, the environment is assumed to be partially observable, restricting agents to accessing only local observations. Each agent's lidar has a range of 0.3 and provides 12-ray data representing distances to surrounding objects. If a lidar ray detects no object within its sensing range, the agent receives a reading of 0 for that ray. This partially observable setting of VMAS makes agents susceptible to long-horizontal planning capability beyond immediate surroundings. To address this limitation, we propose providing agents with access to a certain representation form of the global state. Within the original setting, a global state can be represented as the concatenation of local observations. However, the state dimensionality of this representation varies with the number of agents and may overlook crucial global information unobserved by individual agents. Conversely, providing agents with the global state alone lacks agent-specific information, potentially leading to homogeneous agent behavior of all agents in the absence of local observations. Thus, it is essential to integrate global information with local observations.

1) Global Map Representation: In our study, we represent global information as a 2D image map with 64×64 pixels. We believe image representation of the global map can more effectively capture spatial information compared to representing it as coordinate locations. Additionally, image-based representation adapts well to scalability challenges posed by the varying number of agents without altering dimensionality. The 2D global image map encompasses agents' current locations and goal points.

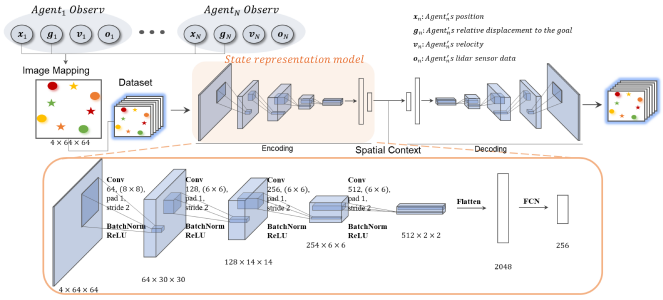

Fig. 1. The training framework of the state representation model. The model is the encoder part with 4 convolutional layers and 2 fully connected networks of the ConvAE that learns the global map image dataset. The global map includes information on the agents' location and goal points. The role of the state representation model is to abstract the spatial context of the global map to inform agents with global state representation with reduced dimension.

C. Action Space and Reward

In VMAS, agent actions are 2D forces within the physics engine, enabling holonomic motion. After each action step, individual agents receive rewards based on their respective statuses. The reward structure is outlined in Table I. After each time step, individual rewards are accumulated to form the team reward.

TABLE I
REWARD STRUCTURE OF VMAS $Navigation$ MODE

Status	Reward
Positional Reward	$d(\boldsymbol{x}_{cur}, g) - d(\boldsymbol{x}_{prev}, g)$
Collision	-1
Final	0.01

IV. LEARNING WORLD MODEL

The training framework of the state representation model is illustrated in Fig 1. The primary role of the model is to extract an abstract, compressed representation from the global map generated from agents' observations and inform agents of the spatial context. Here, we employ a simple convolutional autoencoder (ConvAE) as our state representation model. We collect training data comprising 5 episodes for varying numbers of agents, ranging from 2 agents to 100 agents. Each episode consists of 50 frames, resulting in a total of 24750 frames for the training dataset. The episodes are rolled out using the MAPPO model from BenchMARL [22]. The ConvAE takes a $4 \times 64 \times 64$ global map image input tensor and processes it through 4 convolutional layers and 2 fully connected networks (FCN) layers to encode it into a latent vector with 256 dimensions. The decoder then reconstructs the original image frame from the latent vector.

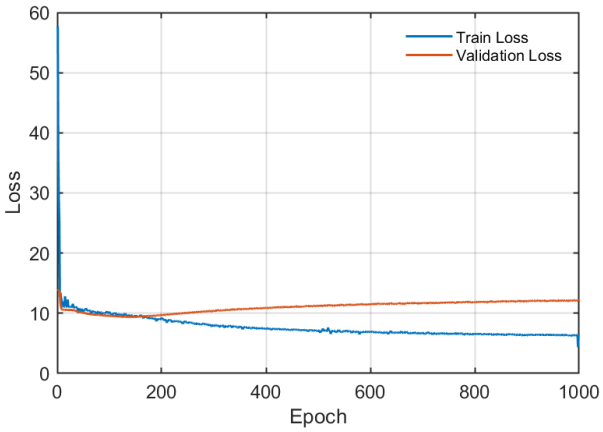

Fig. 2. The training slope of the state representation model. The model trained for 162 epochs is selected as it overfits the training dataset when trained with further epochs.

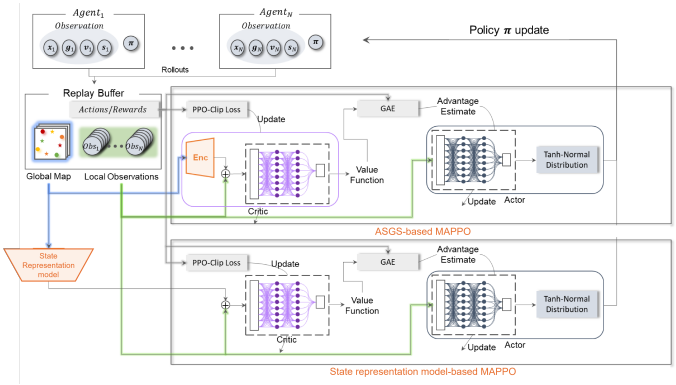

Fig. 3. The framework of ASGS-based MAPPO and the state representation model-based MAPPO. The main difference between the two frameworks is the location of the global map feature extractor. In ASGS-based MAPPO, the encoder is inside the critic making the parameters updated simultaneously with the value function networks. This incurs the encoder to extract global map features based on value function which can be biased by the agent training process. Whereas, the encoder in the state representation model-based MAPPO extracts global map features based on the global map itself via pre-trained model.

Each convolution and deconvolution layer uses a stride of 2 and padding of 1. The first convolutional layer employs an (8×8) kernel and outputs 64 channels, resulting in a $64 \times 30 \times 30$ tensor. Subsequently, the tensor is passed to the second convolutional layer with a (6×6) kernel and 128 channels, outputting a $128 \times 14 \times 14$ tensor. The third and last convolutional layers have the same kernel size as the second layer but with 256 and 512 channels, respectively, outputting $256 \times 6 \times 6$ and $512 \times 2 \times 2$ tensors. Finally, the tensor goes through two FCN layers with 256 nodes to generate the latent vector. The decoder reconstructs the original $4 \times 64 \times 64$ image by passing the latent vector to an exactly inverted network architecture to the encoder. The decoder learns to minimize the mean-squared error (MSE) between the reconstructed image and the original image. We train the model for 1000 epochs, saving the encoder model every 10 epochs. The model closest to convergence of the loss function without overfitting the training dataset is selected as the state representation model to avoid overfitting. Fig 2 shows the result of the training slope of the state representation model.

V. State Representation Model-Based MAPPO for MAPF

A distinctive feature of this work from previous studies utilizing MAPPO [12], [17] is that the actor-critic network not only depends on individual observations but also the features from the global map itself extracted from the state representation model. While the study [12] empirically shows that the ASGS facilitates value learning and enhances MAPPO's performance, the critic-network parameters are updated based solely on value function. This may force the critic-network to update ASGS based on value function, which may be biased by the training process and not fully reflect the spatial context of the global state. Our approach intends to inform agents with a better representation of the environmental state and explores the potential of incorporating the state representation model into the MAPPO framework. The state representation model can also effectively reduce the input dimension to the MAPPO network, lightening the size of the network parameters. This may allow more compact agents in real applications.

In this paper, we conduct a comparison study between ASGS-based MAPPO and state representation model-based MAPPO for MAPF. The differences between the two MAPPO frameworks are depicted in Fig 2. The critic-network of ASGS-based MAPPO consists of an encoder with the same architecture as our proposed framework described in section IV and two subsequent FCN layers. The encoder extracts 256-dimensional image features from the global map and those features are concatenated with 18-dimensional local observations to constitute 274-dimensional ASGS. Then, the ASGS is passed through the FCN layers which are comprised of 274 nodes to estimate the team objective function which is the expected reward of the entire rewards at the end of the scenario. On the other hand, the critic-network of the state representation model-based MAPPO consists of only two FCN layers with 274 nodes. It also takes the input with the concatenation of the environmental states and local observations, but the environmental state this turn is extracted from the model.

The actor-network of both frameworks only takes input from the agent's local observations and passes them through 2 FCN layers with 128 nodes. Since the agents' action space is continuous, we utilize a probabilistic actor with a stochastic policy. The actor-network outputs the mean and scale parameters of a distribution and the action is derived by passing those parameters to the Tanh-normal distribution with the action space boundaries. Both frameworks adopt common practices in implementing PPO, including Generalized Advantage Estimation (GAE) [23] with value-clipping. The hyperparameter details for GAE and PPO loss are listed in Table II. ϵ_{clip} refers to the value-clipping threshold for

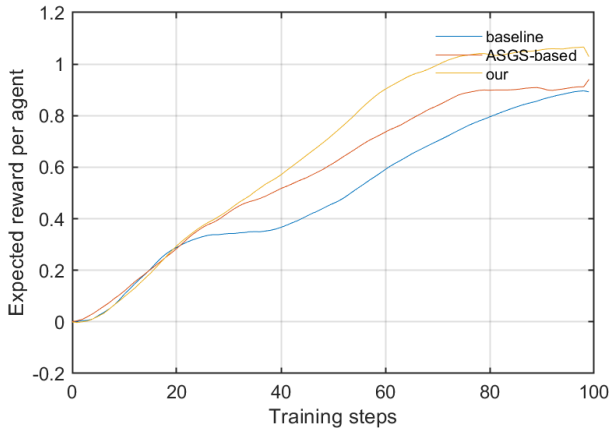

Fig. 4. The training curve of baseline MAPPO, ASGS-based MAPPO, and state representation model-based MAPPO.

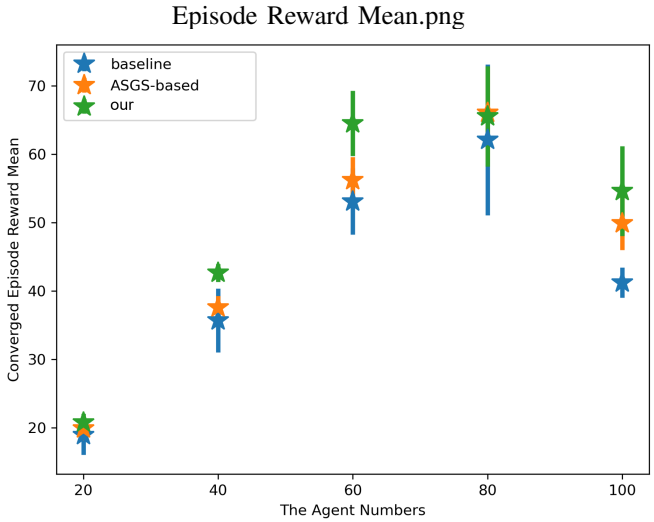

Fig. 5. The average episode reward of training baseline MAPPO, ASGS-based MAPPO, and state representation model-based MAPPO. The experiment results were derived across 20, 40, 60, 80, and 100 agents. The error bars denote the deviation of the episode reward means.

PPO loss, γ to discount factor, λ to GAE coefficient, and ϵ_{etp} to entropy multiplier for computing the total loss.

According to the study in [12], the networks are trained using 10 epochs per update to improve the stability of policy and value learning. Besides, we split the training data into two mini-batches to improve practical performance by using more data to estimate gradients as suggested in [24]. 500 VMAS scenarios are chosen as the batch size for training the frameworks and both MAPPOs are trained until convergence.

TABLE II
HYPERPARAMETERS FOR TRAINING MAPPO

Hyperparameters	***Value***
heightϵ_{clip}	0.2
γ	0.9
λ	0.9
ϵ_{etp}	0.0001

VI. EXPERIMENT RESULTS

We train several baseline MAPPO, ASGS-based MAPPO, and state representation model-based MAPPO with different numbers of agents. The baseline MAPPO trains the critic network with the concatenation of local observations from agents. According to the training results, our proposed framework tend to better learn the problem than other two frameworks. Fig 4 shows the training curve on the frameworks with 20 agents settings and Fig 5 shows the training results of the converged value of the episode reward means over training the environments with different numbers of agents. The episode reward mean tends to drop when the number of agents exceeds 80, which might caused by the congestion of a large number of agents blocking each other's path within a limited square box. Still, the experimental result shows that the MAPPO framework tends to extract more relevant information for mapping actions with rewards from state representation model than a simple concatenation of local observations from entire agents, or ASGS.

VII. CONCLUSIONS AND FUTURE STUDIES

This work demonstrates that MAPPO trained with better state representations achieves improved results in value learning and policies for MAPF. Specifically, we propose a state representation model-based MAPPO framework where the model is introduced to extract the spatial context of the global map in the MAPF environment and provide the environmental information to the MAPPO framework. ConvAE is trained with global map samples and the encoder part of the trained ConvAE is extracted as the state representation model. We conduct a comparison study between three MAPPO frameworks: baseline MAPPO, ASGS-based MAPPO, and proposed MAPPO framework. The experimental result shows that our proposed framework provides a better representation of the global state for improved episode reward means.

Our study still shows that there is a limitation in our framework when applied to a congested environment. However, we believe it builds the foundation of future work and has the potential to address the issue and to provide further improvements. Firstly, the state representation model can provide global information not only to the critic but also to the actor as it abstracts the high-dimensional environmental state to a size-invariant low-dimensional latent vector. This would allow the agents' policy networks to have a compact memory size with low computational costs for the policy network. We also expect this would reduce both non-stationarity and partial observability of the agents in the MAPF environment, without formulating a communication algorithm between the agents.

Moreover, we only study the simplest ConvAE model for the state representation model in this work. Since it only captures a single frame of the global map to extract contextual features, spatial information is extracted to represent the environmental state in this study. However,

spatio-temporal information can be extracted by adopting sequence models. This might improve the state representation of global information and facilitate the agents to become more proactive.

REFERENCES

[1] D. Hafner, J. Pasukonis, J. Ba, and T. Lillicrap, "Mastering diverse domains through world models," *arXiv preprint arXiv:2301.04104*, 2023.

[2] J. Chung, J. Fayyad, Y. A. Younes, and H. Najjaran, "Learning team-based navigation: a review of deep reinforcement learning techniques for multi-agent pathfinding," *Artificial Intelligence Review*, vol. 57, no. 2, p. 41, 2024.

[3] R. Stern, N. R. Sturtevant, A. Felner, S. Koenig, H. Ma, T. T. Walker, J. Li, D. Atzmon, L. Cohen, T. S. Kumar, *et al.*, "Multi-agent pathfinding: Definitions, variants, and benchmarks," in *Twelfth Annual Symposium on Combinatorial Search*, 2019.

[4] P. Friedrich, Y. Zhang, M. Curry, L. Dierks, S. McAleer, J. Li, T. Sandholm, and S. Seuken, "Scalable mechanism design for multi-agent path finding," *arXiv preprint arXiv:2401.17044*, 2024.

[5] B. Wang, Z. Liu, Q. Li, and A. Prorok, "Mobile robot path planning in dynamic environments through globally guided reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6932–6939, 2020.

[6] Z. Ma, Y. Luo, and H. Ma, "Distributed heuristic multi-agent path finding with communication," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 8699–8705.

[7] L. Chen, Y. Wang, Y. Mo, Z. Miao, H. Wang, M. Feng, and S. Wang, "Multiagent path finding using deep reinforcement learning coupled with hot supervision contrastive loss," *IEEE Transactions on Industrial Electronics*, vol. 70, no. 7, pp. 7032–7040, 2022.

[8] Y. Wang, M. Damani, P. Wang, Y. Cao, and G. Sartoretti, "Distributed reinforcement learning for robot teams: A review," *Current Robotics Reports*, vol. 3, no. 4, pp. 239–257, 2022.

[9] A. Skrynnik, A. Andreychuk, K. Yakovlev, and A. I. Panov, "When to switch: planning and learning for partially observable multi-agent pathfinding," *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[10] W. Li, H. Chen, B. Jin, W. Tan, H. Zha, and X. Wang, "Multi-agent path finding with prioritized communication learning," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 10 695–10 701.

[11] R. Lowe, Y. I. Wu, A. Tamar, J. Harb, O. Pieter Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," *Advances in neural information processing systems*, vol. 30, 2017.

[12] C. Yu, A. Velu, E. Vinitsky, J. Gao, Y. Wang, A. Bayen, and Y. Wu, "The surprising effectiveness of ppo in cooperative multi-agent games," *Advances in Neural Information Processing Systems*, vol. 35, pp. 24 611–24 624, 2022.

[13] G. Sartoretti, J. Kerr, Y. Shi, G. Wagner, T. S. Kumar, S. Koenig, and H. Choset, "Primal: Pathfinding via reinforcement and imitation multi-agent learning," *IEEE Robotics and Automation Letters*, vol. 4, no. 3, pp. 2378–2385, 2019.

[14] M. Damani, Z. Luo, E. Wenzel, and G. Sartoretti, "Primal _2: Pathfinding via reinforcement and imitation multi-agent learning-lifelong," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2666–2673, 2021.

[15] Y. Wang, B. Xiang, S. Huang, and G. Sartoretti, "Scrimp: Scalable communication for reinforcement-and imitation-learning-based multi-agent pathfinding," *arXiv preprint arXiv:2303.00605*, 2023.

[16] C. He, T. Yang, T. Duhan, Y. Wang, and G. Sartoretti, "Alpha: Attention-based long-horizon pathfinding in highly-structured areas," *arXiv preprint arXiv:2310.08350*, 2023.

[17] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[18] D. Ha and J. Schmidhuber, "World models," *arXiv preprint arXiv:1803.10122*, 2018.

[19] L. Kaiser, M. Babaeizadeh, P. Milos, B. Osinski, R. H. Campbell, K. Czechowski, D. Erhan, C. Finn, P. Kozakowski, S. Levine, *et al.*, "Model-based reinforcement learning for atari," *arXiv preprint arXiv:1903.00374*, 2019.

[20] V. Micheli, E. Alonso, and F. Fleuret, "Transformers are sample efficient world models," *arXiv preprint arXiv:2209.00588*, 2022.

[21] M. Bettini, R. Kortvelesy, J. Blumenkamp, and A. Prorok, "Vmas: a vectorized multi-agent simulator for collective robot learning," *arXiv preprint arXiv:2207.03530*, 2022.

[22] M. Bettini, A. Prorok, and V. Moens, "Benchmarl: Benchmarking multi-agent reinforcement learning," *arXiv preprint arXiv:2312.01472*, 2023.

[23] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel, "High-dimensional continuous control using generalized advantage estimation," *arXiv preprint arXiv:1506.02438*, 2015.

[24] A. Ilyas, L. Engstrom, S. Santurkar, D. Tsipras, F. Janoos, L. Rudolph, and A. Madry, "A closer look at deep policy gradients," *arXiv preprint arXiv:1811.02553*, 2018.