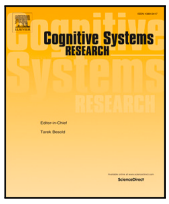
Cognitive Systems Research 89 (2025) 101306

Contents lists available at ScienceDirect

Cognitive Systems Research

journal homepage: www.elsevier.com/locate/cogsys



Applying opponent and environment modelling in decentralised multi-agent reinforcement learning

Alexander Chernyavskiy [a,*], Alexey Skrynnik [b,c], Aleksandr Panov [a,b,c]

[a] Moscow Institute of Physics and Technology, Dolgoprudny, Russia
[b] AIRI, Moscow, Russia
[c] Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences, Moscow, Russia

ARTICLE INFO

Action editor: Serge Thill

Keywords:
Multi-agent systems
Model-based reinforcement learning
Opponent modelling
Multi-agent reinforcement learning

ABSTRACT

Multi-agent reinforcement learning (MARL) has recently gained popularity and achieved much success in different kind of games such as zero-sum, cooperative or general-sum games. Nevertheless, the vast majority of modern algorithms assume information sharing during training and, hence, could not be utilised in decentralised applications as well as leverage high-dimensional scenarios and be applied to applications with general or sophisticated reward structure. Thus, due to collecting expenses and sparsity of data in real-world applications it becomes necessary to use world models to model the environment dynamics, using latent variables — i.e. use world model to generate synthetic data for training of MARL algorithms. Therefore, focusing on the paradigm of decentralised training and decentralised execution, we propose an extension to the model-based reinforcement learning approaches leveraging fully decentralised training with planning conditioned on neighbouring co-players' latent representations. Our approach is inspired by the idea of opponent modelling. The method makes the agent learn in joint latent space without need to interact with the environment. We suggest the approach as proof of concept that decentralised model-based algorithms are able to emerge collective behaviour with limited communication during planning, and demonstrate its necessity on iterated matrix games and modified versions of StarCraft Multi-Agent Challenge (SMAC).

1. Introduction

A problem of modelling interactions between autonomous learning agents has been challenging for a long time (Brown, 1951). In multi-agent systems there could emerge behaviour patterns that could not be observed in single-agent systems — for example, "tit-a-tat" strategy in iterated prisoner's dilemma (IPD) (Axelrod & Hamilton, 1981) or intended cooperation (Davydov, Liusko, & Panov, 2021; Yi, Li, Wang, & Lu, 2021). Since then, multi-agent learning has been applied to a lot of real-life scenarios, such as autonomous driving (Albrecht & Stone, 2018; Dinneweth, Boubezoul, Mandiau, & Espié, 2022), multi-robotic systems (Oroojlooy & Hajinezhad, 2023; Singh, Kumar, & Singh, 2022), gaming system competing against humans (Berner et al., 2019; Silver et al., 2018; Vinyals et al., 2019) and others. In essence, multi-agent learning can be represented as either a multitude of agents simultaneously learning to interact with each other or an agent playing against copies of itself from previous learning steps. Thus, each individual agent, while learning, might affect learning of other agents, making the process extremely non-stationary (Dinneweth et al., 2022; Lowe et al., 2017). Commonly, this issue can be mitigated by reinforcement learning algorithms, however, it requires numerous environmental steps, and that highlights the importance of sample efficient training.

World models (Ha & Schmidhuber, 2018) — i.e. models of approximated environment dynamics learned from repeated interactions with the environment — have been a relatively new way to reach sample efficiency during training of reinforcement learning agents. Despite diversity of those methods (Moerland, Broekens, Plaat, Jonker, et al., 2023), most of modern approaches utilise latent variables to compress spatial and temporal representations of the environment in low-dimensional discrete or continuous variables (Hafner, Pasukonis, Ba, & Lillicrap, 2023; Ye, Liu, Kurutach, Abbeel, & Gao, 2021); The world models employ planning (Sutton, 1990) to collect synthetic data samples and improve their actor–critic model that utilises adjusting the latent policy acting in the environment using various reinforcement learning algorithms. Being much more sample efficient, model-based reinforcement learning has plenty of challenges yet to be fully resolved like dealing with stochasticity, uncertainty, partial observability, compounding errors during planning and others (Moerland et al.,

* Corresponding author.
E-mail address: chernyavskij.as@phystech.edu (A. Chernyavskiy).

https://doi.org/10.1016/j.cogsys.2024.101306
Received 6 August 2024; Accepted 18 November 2024
Available online 2 December 2024

2023). Therefore, they were only recently brought to MARL (Egorov & Shpilman, 2022; Skrynnik, Andreychuk, Yakovlev and Panov, 2024; Venugopal, Milani, Fang, & Ravindran, 2023; Wang, Zhang, & Zhang, 2022) and have their own limitations.

In multi-agent systems, there should be a key distinction in what information learning agents possess during the training and execution (or inference) procedure. One of the most popular approaches is centralised training and decentralised execution scheme (Du, Leibo, Islam, Willis, & Sunehag, 2023; Lowe et al., 2017). One restricts communication (direct information flow from observations of other agents, and indirect, or using special communication blocks, like attention mechanism Egorov & Shpilman, 2022) between the agents during the execution phase but lets them do so during the training.

However, in most practical applications — due to the essential requirement of constant adaptation and extreme sparsity, restrictions on communication (Lyu, Baisero, Xiao, Daley, & Amato, 2023; Skrynnik, Andreychuk, Nesterova, Yakovlev and Panov, 2024), necessity to scale with number of agents (Gupta, Egorov, & Kochenderfer, 2017; Zhang, Sun, Barth, & Ma, 2020) — the right choice may be a model of decentralised training with decentralised execution when each agent acts in the environment independently on the others and treats it as stationary at each time step. Despite having its own issues (Amato, 2024), ones can be mitigated with introducing communication protocols (Jiang & Lu, 2018) to avoid full centralisation, or utilise opponent modelling (Albrecht & Stone, 2018; Kiselev & Panov, 2017) as anticipation of other agents' behaviour or using meta-learning methods (Beck et al., 2023) to enforce generalisation for the agents.

Modelling co-players' or behavioural policies could help us infer "mental" states of a learning agent and its co-players — i.e. their probabilistic beliefs, intentions or action preferences, also known as Theory of Mind (ToM, Baker, Saxe, and Tenenbaum (2011)). This process of inferring beliefs of the other agents is analogous to cognitive abilities of humans to build models of others and utilise them to anticipate how other people can influence their own behaviour and lives, predict potential social interactions. ToM has been studied from perspectives of psychology, cognitive science and social sciences for a long time. In reinforcement learning, there were a lot of works on modelling agents' incentives, for example, machine theory of mind (Cross, Xiang, Bhatia, Yamins, & Haber, 2024; Rabinowitz et al., 2018), utilising meta-learning to build models of the encountered agents, from observations of their behaviour alone. The method targets learning RL policies, explicitly taking into account semantically meaningful beliefs of the agents' co-players. As decentralised agents learn independently, for emergence of collective behaviour it is essential to build beliefs about the opponents. That can be viewed as an opportunity to apply machine ToM.

We attempt to make a sample-efficient model-based algorithm learning and acting in a decentralised manner, but aware of other agents to enforce collective behaviour. We list our contributions as follows:

(1) We modify planning process of decentralised multi-agent baseline based on (Egorov & Shpilman, 2022) in a way that lets agents incorporate necessary information relevant to their closest co-players. This allows to train policies purely using synthetic trajectories of latent states generated by the model of the agent and its co-players or opponents;
(2) We demonstrate the proof of the concept on several multi-agent scenarios, including various matrix games and modified versions of StarCraft Multi-Agent Challenge (SMAC) (Rutherford et al., 2023; Samvelyan et al., 2019) targeting extensive forms of cooperation.

2. Background

2.1. Multi-agent reinforcement learning

One of the most common ways to mathematically describe interaction between autonomous agents in a shared environment is a partially observable stochastic game, which captures a sequential decision-making process when the agents have limited ability to observe the game state. Formally, a partially observed stochastic game (POSG) (Shoham & Leyton-Brown, 2008) could be defined as a tuple

$$G = \langle \mathcal{I}, \mathcal{S}, \{\mathcal{A}\}_{i=1}^{N}, P, \rho_0, \{R^i\}_{i=1}^{N}, \{\mathcal{O}^i\}_{i=1}^{N}, \{O^i\}_{i=1}^{N}, \gamma \rangle, \quad (1)$$

where $\mathcal{S}$ is a finite set of the environment states, $\mathbf{A} = \prod_{i\in\mathcal{I}} \mathcal{A}^i$ — a finite set of joint actions, $\mathbf{O} = \prod_{i\in\mathcal{I}} \mathcal{O}^i$ — a joint observation space, $O^i : \mathcal{S} \times \mathbf{A} \times \mathcal{O}^i \to \Delta(\mathcal{O}^i)$ – an observation function of the ith agent, $\rho_0 \in \Delta(\mathcal{S})$ — initial state distribution.

The game proceeds as follows. Starting from an initial state $s_0 \sim \rho$, at time t, an agent $i \in \mathcal{I}$ receives a private observation o_t^i governed by the observation function, $O^i(o_t^i|s_{t+1}, a_t)$, and chooses an action $a_t^i \in \mathcal{A}^i$m that gets simultaneously executed with all other agents. Given the state s_t and agents' joint action $\mathbf{a}_t = \{a_t^i\}$, the environment transitions to the next state according to the state transition function $P(s_{t+1}|s_t, \mathbf{a}_t) : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \Delta(\mathcal{S})$ and gets reward according to the reward function $r_t^i = R^i(s_{t+1}, \mathbf{a}_t, s_t) : \mathcal{S}\times\mathcal{A}\times\mathcal{S} \to \mathbb{R}$. Each agent aims to find a behavioural policy $\pi^i(a_t^i|h_t^i) \in \Pi^i : \mathcal{T} \to \Delta(\mathcal{A}^i)$ conditioned on action-observation history $h_t = (s_0, \mathbf{o}_0, \mathbf{a}_0, s_1, \mathbf{o}_1, \mathbf{a}_1, \dots, s_t, \mathbf{o}_t) \in \mathcal{T} = (\mathcal{S} \times \mathbf{A} \times \mathbf{O})^*$ that can guide the agent to take sequential actions such that the discounted cumulative reward of the joint policy $\pi = (\pi^i, \pi^{-i})$ is maximised.

It is worth noting, that for cooperative games it is true that $\forall i, j \in \mathcal{I}\ \mathcal{R}^i = \mathcal{R}^j = \mathcal{R}$ — i.e. all the agents share the same reward function. This setting is called decentralised partially observable Markov decision process (Dec-POMDP) in the literature (Amato, 2024; Oliehoek, 2012). Moreover, the framework also covers repetitive matrix games: they could be considered as POSGs with a single environment state ($|\mathcal{S}| = 1$). As we focused on a fully decentralised setting, agents utilise only their own observations to make decisions, any communication in the training process is prohibited.

2.2. World models

Model-based reinforcement learning (Sutton, 1990) could be defined as an indirect form of behaviour learning by learning a model of the environment — i.e. transition and reward functions of POSG — by observed experience. The learned model is used to improve an agent's policy with decision-time planning (using the learned model to roll out the optimal action at a given state by looking forward during execution). From all the diverse family of single-agent model-based algorithms, state-of-the-art performance in discrete (like Atari games) and continuous control (like robotic manipulation) tends to be demonstrated by the world models using latent variable (EfficientZero and Dreamer models) (Hafner et al., 2023; Lee, Nagabandi, Abbeel, & Levine, 2020; Ye et al., 2021) that follow the structure of Sutton (1990) using supervised and self-supervised learning to repeatedly update model from interactions with the environment.

Conventionally, a latent variable model is implemented as a sequential variational autoencoder (Kingma & Welling, 2013) or the other type of probabilistic inference model that squeezes high-dimensional environment data into a latent variable. Experimentally shown (Hafner et al., 2023; Xie, Losey, Tolsma, Finn, & Sadigh, 2021), models with the discrete latent variables perform better than continuous. Given a set of interactions from a single agent POSG also called a partially observable Markov decision process (POMDP), $\{o_t, a_{t-1}\}_{t=0}^{T}$, the model employs a latent variable z_t to estimate the probability of observed data $p(o_{1:T}|a_{0:T-1})$, and evidence lower bound (ELBO) (Kingma & Welling, 2013) is used for the log-likelihood of the sequence of the data. The

objective for distributions parameterised by a neural network with set of parameters φ can be denoted as follows:

$$\max_{\varphi} \log p(o_{1:T}|a_{0:T-1}) \geq -\max_{\varphi} \mathbb{E}_{z_{1:T}\sim q_{\varphi}} \sum_{t=1}^{T} \log p_{\varphi}(o_t|z_t) \\ -KL(q_{\varphi}(z_t|o_t, a_{t-1}) \parallel p_{\varphi}(\hat{z}_t|a_{t-1})) \tag{2}$$

where $p_{\varphi}(o_t|z_t)$ denotes an encoder model and the second term describes KL-divergence (Kingma & Welling, 2013) between the prior and posterior transition distributions.

In our work, we are focused on model-based architecture called Dreamer (Hafner, Lillicrap, Ba, & Norouzi, 2019). It could be defined as follows: its core is based on a recurrent state–space model (RSSM) that has two components — stochastic and deterministic — and each one is represented with its own latent variable z_t and h_t, correspondingly. The latter is done with a recurrent neural network and the former — with variational autoencoder. The model is parameterised with a set of parameters ϕ.

$$\text{RSSM} \begin{cases} \text{Recurrent model:} & h_t = f_{\phi}(h_{t-1}, z_{t-1}, a_{t-1}), \\ \text{Representation model:} & z_t \sim q_{\phi}(z_t|h_t, x_t), \\ \text{Transition predictor:} & \hat{z}_t \sim p_{\phi}(\hat{z}_t|h_t), \end{cases} \tag{3}$$

$$\text{Auxiliary components} \begin{cases} \text{Image predictor:} & \hat{x}_t \sim p_{\phi}(\hat{x}_t|h_t, z_t), \\ \text{Action predictor:} & \hat{a}_t \sim p_{\phi}(\hat{a}_t|h_t, z_t), \\ \text{Reward predictor:} & \hat{r}_t \sim p_{\phi}(\hat{r}_t|h_t, z_t), \\ \text{Discount predictor:} & \hat{\gamma}_t \sim p_{\phi}(\hat{\gamma}_t|h_t, z_t), \end{cases} \tag{4}$$

The model accepts observations of the learning agent x_t as input. The transition predictor attempts to reconstruct the next model state only from the current model state about any information about transition. This means that the dynamics is learned implicitly, making it unnecessary to generate and observe in future for planning. The discount predictor estimates the probability of an episode ending when learning behaviours from model predictions, and reward predictions let the model to reconstruct the reward function.

Loss function to achieve the objective is defined in the following way:

$$\max_{\varphi} \mathbb{E}_{q_{\phi}(z_{1:T}|x_{1:T}, a_{1:T})} - \log p_{\phi}(x_t|h_t, z_t) - \log p_{\phi}(r_t|h_t, z_t) - \log p_{\phi}(\gamma_t|h_t, z_t) \\ +\beta \cdot KL\left[q_{\phi}(z_t|h_t, x_t) \parallel p_{\phi}(z_t|h_t)\right] \tag{5}$$

In multi-agent reinforcement learning, there were proposed several extensions of the aforementioned model to make the world models feasible in multi-agent setting. One of them Egorov and Shpilman (2022) used an additional communication block implemented as transformer encoder (Vaswani, 2017) and optimising mutual information $\log p_{\phi}(\hat{a}_{t-1}|h_t, z_t)$ for better performance. Additionally, the model used multi-agent PPO algorithm (MAPPO, Yu et al. (2022)) to improve the agents' policies. Different from the standard approach of Dreamer using advantage actor–critic, the authors (Egorov & Shpilman, 2022) empirically found that MAPPO performs better in multi-agent scenarios. Nonetheless, it was noted (Venugopal et al., 2023) that the communication mechanism and parameter sharing (the model employs one set of parameters for all the agents) could violate what data the agents are able to access during the training and the execution. To mitigate that, they proposed a two-level hierarchical method that allows to decompose global information used in training from local observations of the agents. Although, both method demonstrate state-of-the-art performance on multi-agent benchmarks, they appear to be centralised at least during training. We aim to build a model-based algorithm following a scheme of decentralised training and decentralised execution.

3. Method

Here, we propose and explain details of our method based on latent conditioning during the planning procedure. Our method is based upon the powerful baseline MAMBA (Egorov & Shpilman, 2022). We base our explanation on the StarCraft multi-agent challenge because it nicely approaches multi-agent cooperation and has visual observations, however, the general principle could be applied to any environment with homogeneous agents and cooperative reward function.

3.1. World model training

We build our approach on top of aforementioned Dreamer architecture (Eq. (3)) optimising its corresponding objective. Moreover, we incorporate an additional loss term from the baseline model: information loss $-\log p_{\phi}(\hat{a}_{t-1}|h_t, z_t)$ from Egorov and Shpilman (2022) to encourage the latent state to depend mostly on their own actions, thus making it less dependent on others. Done this way, each agent learns its own world model based on egocentric interactions with the environment. Crucially, we omit all communication between the agents.

3.2. Co-player based planning

Following the line of work (Venugopal et al., 2023), there are four stages of the model-based training: (1) collecting data by interaction with a learning environment, (2) training the world model, (3) planning in the latent space of the model to produce latent trajectories, (4) training a reinforcement agent on these trajectories to improve the environment collection policy. As the first stage requires interaction with the environment and each agent relies only on its own observation and act independently, we decided to concentrate on improving planning stage because it does not have to utilise environment data and thus, can be done effectively using world models of the agents.

During the planning, we adopt "conditioning in imagination" as follows (see Fig. 2). Each agent gets its posterior estimation from training samples employing corresponding replay memories. So, let us select two agents ith (a modelled agent) and jth (ith co-player). We choose co-players based on visibility (visual presence, or in matrix games we always choose the opponent) the jth agent on the ith agent observation. Specifically, we aim to predict the ith agent viewpoint with the jth agent observation prediction model if the jth agent in a visibility range of ith. This is done to make sure that the modelled agent start planning from using the co-player visibility range as a suitable prior of the environment. Notably, this does not mean that the agent now has complete information about the environment state: the agent does not widen its point of view because of the conditioning but makes the agent aware of his co-player might have seen. It can as widen the viewpoint as well as mislead the agent, however, we hypothesise that, on average, additional information is better for the agent.

Furthermore, both agents plan using their policies, but the modelled agent takes aggregated own latent states and co-players' latent states to condition on the transition dynamics The behaviour of the agents should remain symmetrical, as depicted in Fig. 1 (right). We use aggregation via summation of the latent vectors. Furthermore, each agent's behavioural policy is learned independently by the MAPPO algorithm.

We experiment with learnable and with heuristic co-players. For the latter, we replace the RL algorithm with recovered from the world model action and learn the model in fully supervised way.

4. Experiments

To demonstrate sustainability of our approach, we test it on two groups of methods: iterated matrix games and cooperative high-dimensional environments (i.e. StarCraft multi-agent environment). The choice is dictated by the necessity to test the effect of the proposed method on transition and reward approximation of the world models.

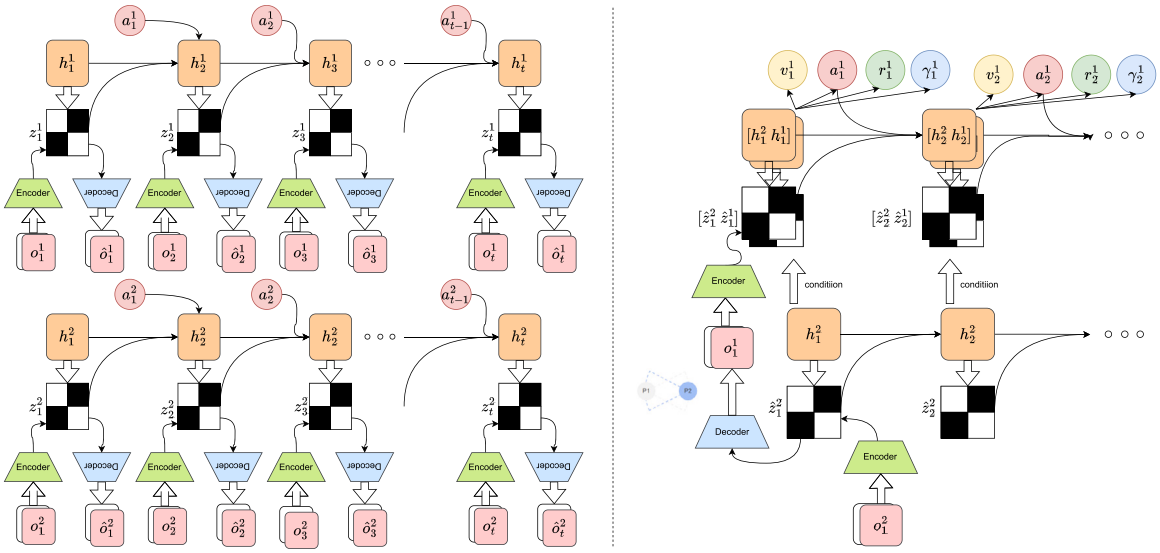


Fig. 1. A visualisation of the proposed method. During the training (left) all agents learn their world models independently, whereas during the planning (right) phase each and every learning agent conditions own trajectories on planning of its co-players based on their proximity during the interaction.

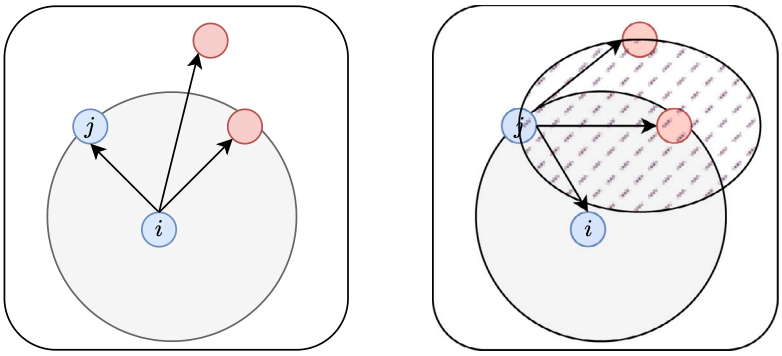


Fig. 2. Visual description of observation conditioning. On the left picture there is a viewpoint of the ith agent, and jth is visible. On the right, there is a new viewpoint of the modelled agent, predicted from the jth agent viewpoint.

Table 1
Structure of the used neural network.

Component	Parameters (M)
World model	6.8
Actor	0.4
Critic	1.25

Table 2
Payoff matrix of prisoner's dilemma.

A1/A2	C	D
C	(2, 2)	(0, 3)
D	(3, 0)	(1, 1)

Matrix games do not have transition dynamics due to only game state but utilising them as test scenarios can give an insight into reward structure and give direct mapping between policy structure and outcomes of the game (Axelrod & Hamilton, 1981). On the contrary, environments like SMAC do not focus much on reward structure (one models a simple cooperation game), however, could give some interpretability of the transition dynamics. For experiments, we use a neural network with a single set of parameters, the number of those is shown in Table 1. We use the same structure of the world model (i.e. RSSM and auxiliary components) for all experiments. Even though matrix games do not have explicit transition dynamics, so some RSSM components may look excessive, we decided to keep the architecture identical for all the cases. Moreover, we observed that performance of the model with "excessive" parts is not worse than a model specifically corrected for the matrix game scenario. All the experiments were run on a single RTX Titan and reported as an average of five random seeds.

4.1. Matrix games

Prisoners dilemma. The game could be considered a benchmark of reciprocal cooperation (Foerster et al., 2017). Moreover, a lot of applications of human interactions could be modelled using the game. The dilemma appears an iterated two-player matrix game with payoff matrix defined as shown in Table 2.

Taking into account episodic structure of the game, the winning strategy is either unconditionally defect until the end of the episode or defect only the other player defects and cooperate otherwise and all the variants of the strategy where a player cooperates until the other does the same (see Fig. 3).

During training, we observed that the agents not utilising modelling tend to defect most of the time and getting +1 reward per step. Meanwhile, with opponent modelling, agents employ a mixed strategy with probability to defect around $\frac{1}{2}$ at each time step, and one of the agents had learnt how to win against the other most of the time. This

Fig. 3. Reward per step plots of two learning agents playing iterated prisoner's dilemma. The dashed line refers to a baseline, or fully decentralised case.

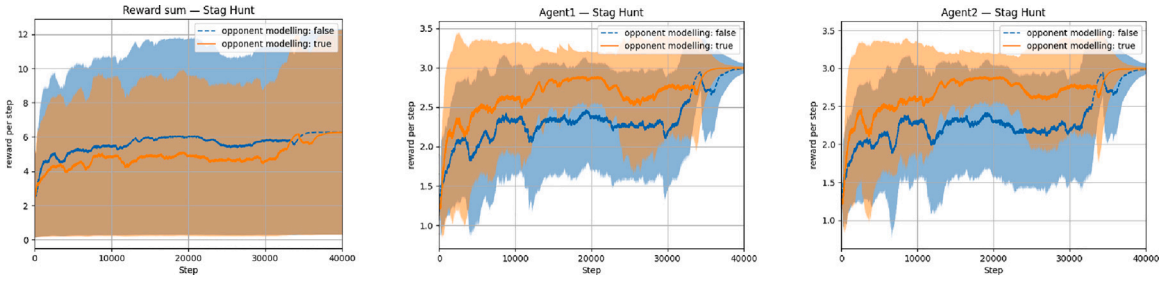


Fig. 4. Reward per step plots of two learning agents playing iterated stag hunt game. The dashed line refers to a baseline, or fully decentralised case.

Table 3
Payoff matrix of stag hunt.

A1/A2	S	H
S	(3, 3)	(0, 1)
H	(1, 0)	(1, 1)

Table 4
Payoff matrix of "battle of sexes".

A1/A2	F	S
F	(2, 1)	(0, 0)
S	(0, 0)	(1, 2)

experiment shows that a learning agent could be conditioned on a particular non-deterministic policy to obtain higher returns. Nonetheless, the pattern of such emergence (i.e. a way to control the behaviour) is left for future studies.

Stag hunt. The game appears to be a trust dilemma, or common interest game with payoff matrix is defined as shown in Table 3.

The stag hunt differs from the prisoner's dilemma in that there are two pure-strategy Nash equilibria, one where both players cooperate, and one where both players defect. In the Prisoner's Dilemma, in contrast, despite the fact that both players cooperating is Pareto efficient, the only pure Nash equilibrium is when both players choose to defect (see Fig. 4).

We observed that in both settings, with modelling, tend to converge to the payoff-centred equilibrium that gives +3 reward to each agent, with both of them playing mixed strategies to hunt stag around 90% of the time.

"Battle of sexes" game. This game models full (non-interchangeable) cooperation between subjects. In this game, that payoff matrix is shown in Table 4, there is a pure Nash equilibrium: mimic the other player's actions or coordinate around half of the time.

We observed that the baseline method converged to pure Nash equilibrium of the one-step game, i.e. to copy the opponent's action. However, in the setting with opponent modelling the agents employed a mixed strategy to deviate from the opponent's action around half of the time, converging to a more "fair" reward per step for the episode. Notably, the method using opponent modelling reaches higher rewards at earlier stages (see Fig. 5).

Overall, from the experiments with the games of different reward and equilibrium, it can be observed that opponent modelling tends to prioritise mixed over independent strategies as well as higher returns for an episode over single step rewards. It is worth noticing that convergence depends on the equilibrium structure of the game. For example, in prisoners' dilemma both agent choose to mutually defect, but do it differently: agents with opponent modelling defect using stochastic strategy, whereas naive agents just choose to defect as the opponent defects. The same we can see in "battle of sexes" game: agents with modelling utilise mixed strategies to confuse the opponent, diverging from the one-step equilibrium. That allow them to potentially reach higher rewards with less learning steps. We consider a direction of investigating, how different equilibrium scenarios can influence stochastic policies of the learning agents, as a direction for future work.

4.2. StarCraft multi-agent challenge

To facilitate the development process and modify the environment, we decided to stick with an alternative to StarCraft multi-agent challenge called SMAX (Rutherford et al., 2023). It varies from the traditional environment with the ability to set flexible configurations, i.e. making different populations of agent learnable and manually adjust heuristic behaviour. We divide our experiments in two groups: in one population, targeting micromanagement scenarios, and between populations, targeting macro-management scenarios. The environment allows two settings: sparse reward setting, when agents get rewarded by the outcome of an episode, and dense reward — based on damage done to/from other agents at the time step. To facilitate the learning process, we allow agents to get dense rewards (see Fig. 6).

Alternating fire scenario. We utilise 2m_vs_1z (a.k.a. Marine Double Team), where two Terran Marines need to defeat an enemy Zealot. In this setting, the Marines must design a cooperative policy which does not allow the Zealot to reach them, otherwise they will be defeated (see Fig. 7).

As the scenario follows the fully cooperative game, independent learning agents and agents with opponent modelling do exactly the same, showing how the world model itself is able to learn simple cooperative behaviour from transition dynamics of the environment.

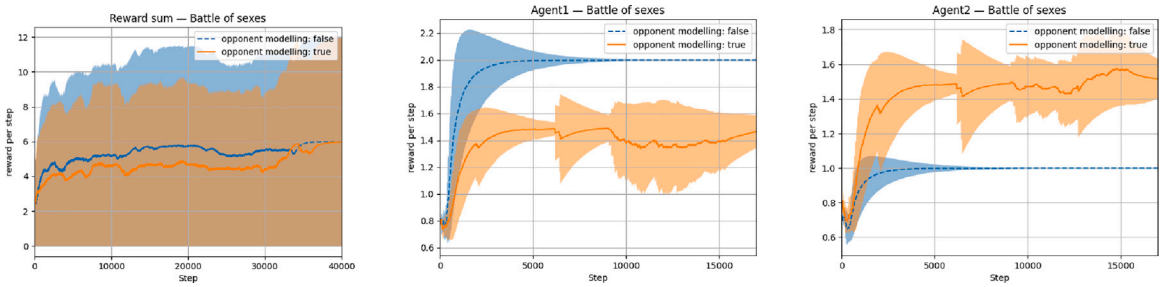


Fig. 5. Reward per step plots of two learning agents playing iterated "battle of sexes" game. The dashed line refers to a baseline, or fully decentralised case.

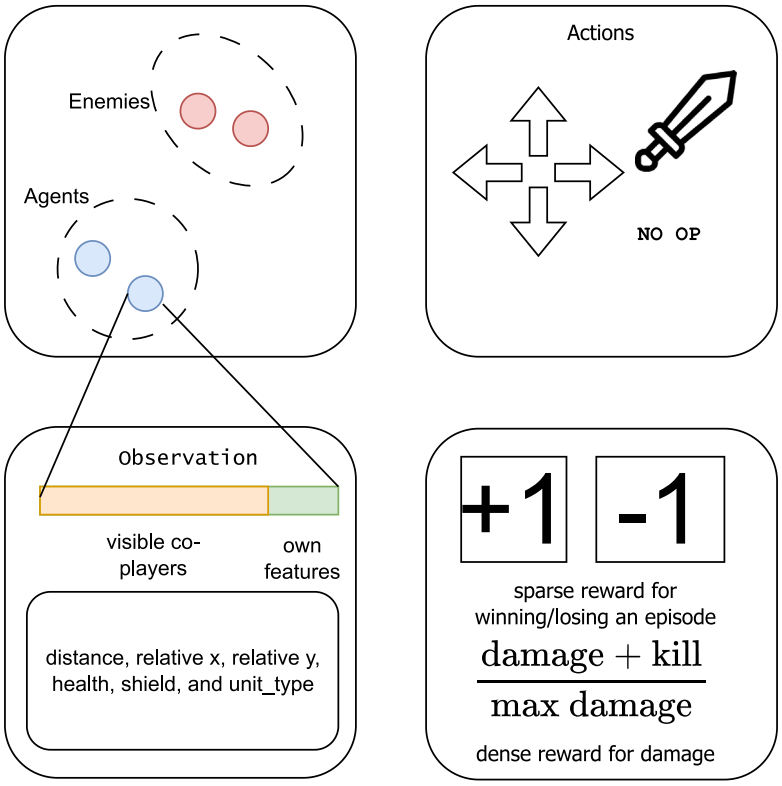


Fig. 6. Simplified version of StarCraft learning environment, SMAX, depicted for four homogeneous agents, with descriptions of observation space, action space and reward structure of the environment.

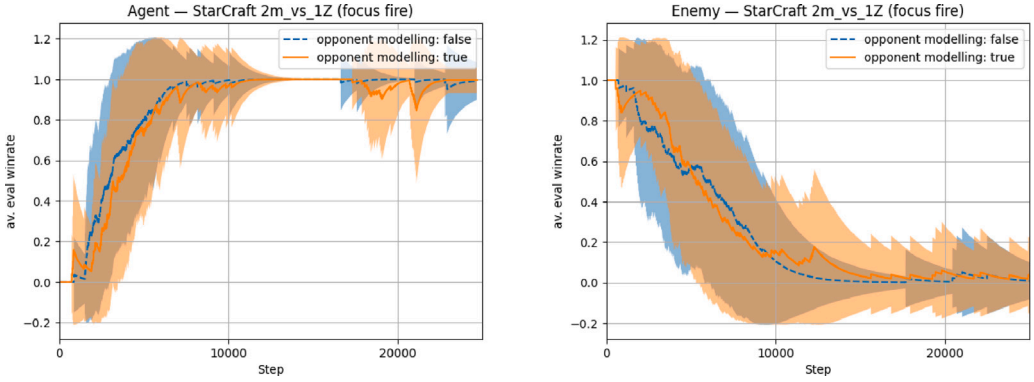


Fig. 7. Average win rate plots for two learning agents playing an iterated "alternating fire scenario" game. The dashed line refers to a baseline, or fully decentralised case.

Focus fire scenario. We came up with a 2m_vs_2 m cross-like scenario where one of the agents is learnable and one is a bot with a randomised target selection. The goal of the learning agent is either support him and isolate 2v1 battle or make a series of 1v1 interactions (see Fig. 9).

There we get similar results, however, with help of modelling the agents tend to better isolate 2v1 duels and 1v1 duels (Fig. 8) whereas learning independently we did not observe 2v1 scenarios, each agent either tries no avoid its fight or have a 1v1 duel.

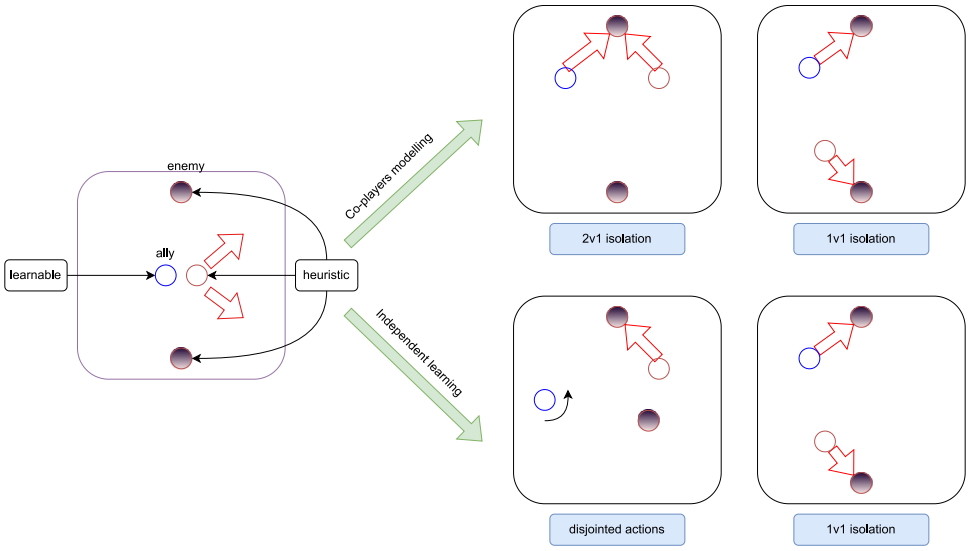


Fig. 8. Visual observations of strategies adopted by the agents in the "focus fire" scenario. With opponent modelling, the agents tend to be more prone to isolate 2v1 duels than act entirely by themselves.

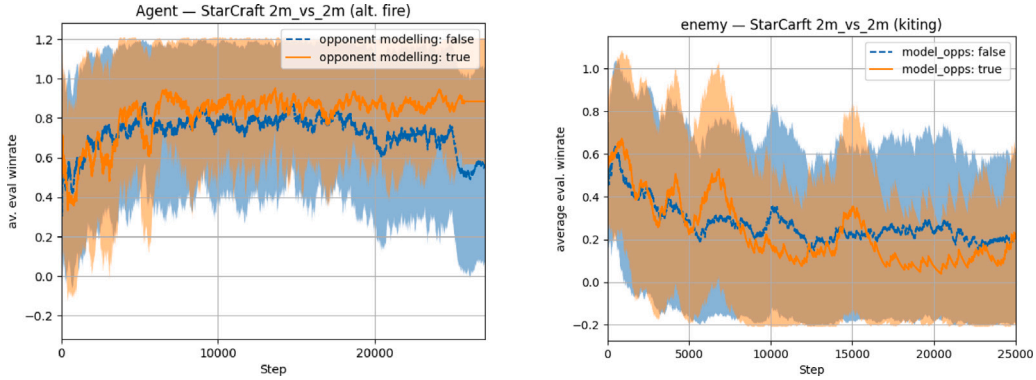


Fig. 9. Average win rate plots for two learning agents playing an iterated "focus fire scenario" game. The dashed line refers to a baseline, or fully decentralised case.

In conclusion, a lot of coordination behaviour in the environment comes from the world model learning on transitions, however, with assistance of modelling teammates' behaviour we could visually observe teamwork-like behavioural patterns but not that well-defined in numbers.

5. Conclusion

We presented an extension to the fully decentralised model-based MARL algorithm that allows to learn cooperative multi-agent behaviours using planning capabilities of the learned world models. Despite the latter are able to handle simple scenarios themselves, they are still unstable in cases when reward structure of the environment is sophisticated, for example, it has several equilibria. We showed that including conditioning to the planning improves convergence and stability of the model-based baseline and lets it handle more difficult cases of conditional cooperation, more relying on the opponent's latent policy. Moreover, we tested and visually recognised, using SMAC, that there could be emerged more teamwork-based behavioural patterns, with this simple team-based conditioning.

However, in our experiments with did not find a satisfying us explanation of how to control conditioning to get desired behaviours: for instance, we got a lot of draws and passive behaviours in StarCraft that were rewarded highly but were not desired from the point of the game. Furthermore, there is an open question about a way how to do the conditioning procedure for the method to perform more efficiently in various environments.

CRediT authorship contribution statement

Alexander Chernyavskiy: Conceptualization, Data curation, Methodology, Supervision, Writing – original draft. **Alexey Skrynnik:** Formal analysis, Investigation, Methodology, Resources, Software, Supervision. **Aleksandr Panov:** Conceptualization, Formal analysis, Methodology, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This work was supported by Russian Science Foundation, Russia, grant No. 20-71-10116, https://rscf.ru/en/project/20-71-10116/.

Data availability

No data was used for the research described in the article.

References

Albrecht, S. V., & Stone, P. (2018). Autonomous agents modelling other agents: A comprehensive survey and open problems. *Artificial Intelligence*, *258*, 66–95.

Amato, C. (2024). (A partial survey of) decentralized, cooperative multi-agent reinforcement learning. arXiv preprint arXiv:2405.06161.

Axelrod, R., & Hamilton, W. D. (1981). The evolution of cooperation. *Science*, *211*(4489), 1390–1396.

Baker, C., Saxe, R., & Tenenbaum, J. (2011). Bayesian theory of mind: Modeling joint belief–desire attribution. *vol. 33*, In *Proceedings of the annual meeting of the cognitive science society*.

Beck, J., Vuorio, R., Liu, E. Z., Xiong, Z., Zintgraf, L., Finn, C., et al. (2023). A survey of meta-reinforcement learning. arXiv preprint arXiv:2301.08028.

Berner, C., Brockman, G., Chan, B., Cheung, V., Dębiak, P., Dennison, C., et al. (2019). Dota 2 with large scale deep reinforcement learning. arXiv preprint arXiv:1912.06680.

Brown, G. W. (1951). Iterative solution of games by fictitious play. *Activity Analysis of Production and Allocation*, *13*(1), 374.

Cross, L., Xiang, V., Bhatia, A., Yamins, D. L., & Haber, N. (2024). Hypothetical minds: Scaffolding theory of mind for multi-agent tasks with large language models. arXiv preprint arXiv:2407.07086.

Davydov, V., Liusko, T., & Panov, A. I. (2021). Self and other modelling in cooperative resource gathering with multi-agent reinforcement learning. In *Advances in intelligent systems and computing*: *vol. 1310*, *Brain-inspired cognitive architectures for artificial intelligence: BICA*AI 2020* (pp. 69–77). Springer International Publishing.

Dinneweth, J., Boubezoul, A., Mandiau, R., & Espié, S. (2022). Multi-agent reinforcement learning for autonomous vehicles: A survey. *Autonomous Intelligent Systems*, *2*(1), 27.

Du, Y., Leibo, J. Z., Islam, U., Willis, R., & Sunehag, P. (2023). A review of cooperation in multi-agent learning. arXiv preprint arXiv:2312.05162.

Egorov, V., & Shpilman, A. (2022). Scalable multi-agent model-based reinforcement learning. arXiv preprint arXiv:2205.15023.

Foerster, J. N., Chen, R. Y., Al-Shedivat, M., Whiteson, S., Abbeel, P., & Mordatch, I. (2017). Learning with opponent-learning awareness. arXiv preprint arXiv:1709.04326.

Gupta, J. K., Egorov, M., & Kochenderfer, M. (2017). Cooperative multi-agent control using deep reinforcement learning. In *Autonomous agents and multiagent systems: AAMAS 2017 workshops, best papers, São Paulo, Brazil, May 8-12, 2017, revised selected papers 16* (pp. 66–83). Springer.

Ha, D., & Schmidhuber, J. (2018). Recurrent world models facilitate policy evolution. *Advances in Neural Information Processing Systems*, *31*.

Hafner, D., Lillicrap, T., Ba, J., & Norouzi, M. (2019). Dream to control: Learning behaviors by latent imagination. arXiv preprint arXiv:1912.01603.

Hafner, D., Pasukonis, J., Ba, J., & Lillicrap, T. (2023). Mastering diverse domains through world models. arXiv preprint arXiv:2301.04104.

Jiang, J., & Lu, Z. (2018). Learning attentional communication for multi-agent cooperation. *Advances in Neural Information Processing Systems*, *31*.

Kingma, D. P., & Welling, M. (2013). Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114.

Kiselev, G. A., & Panov, A. I. (2017). Synthesis of the behavior plan for group of robots with sign based world model. *vol. 10459*, In *Interactive collaborative robotics*: *Lecture Notes in Computer Science*, (pp. 83–94). Springer.

Lee, A. X., Nagabandi, A., Abbeel, P., & Levine, S. (2020). Stochastic latent actor-critic: Deep reinforcement learning with a latent variable model. *Advances in Neural Information Processing Systems*, *33*, 741–752.

Lowe, R., Wu, Y. I., Tamar, A., Harb, J., Pieter Abbeel, O., & Mordatch, I. (2017). Multi-agent actor-critic for mixed cooperative-competitive environments. *Advances in Neural Information Processing Systems*, *30*.

Lyu, X., Baisero, A., Xiao, Y., Daley, B., & Amato, C. (2023). On centralized critics in multi-agent reinforcement learning. *Journal of Artificial Intelligence Research*, *77*, 295–354.

Moerland, T. M., Broekens, J., Plaat, A., Jonker, C. M., et al. (2023). Model-based reinforcement learning: A survey. *Foundations and Trends® in Machine Learning*, *16*(1), 1–118.

Oliehoek, F. A. (2012). Decentralized pomdps. In *Reinforcement learning: State-of-the-art* (pp. 471–503). Springer.

Oroojlooy, A., & Hajinezhad, D. (2023). A review of cooperative multi-agent deep reinforcement learning. *Applied Intelligence: The International Journal of Artificial Intelligence, Neural Networks, and Complex Problem-Solving Technologies*, *53*(11), 13677–13722.

Rabinowitz, N., Perbet, F., Song, F., Zhang, C., Eslami, S. A., & Botvinick, M. (2018). Machine theory of mind. In *International conference on machine learning* (pp. 4218–4227). PMLR.

Rutherford, A., Ellis, B., Gallici, M., Cook, J., Lupu, A., Ingvarsson, G., et al. (2023). Jaxmarl: Multi-agent rl environments in jax. arXiv preprint arXiv:2311.10090.

Samvelyan, M., Rashid, T., De Witt, C. S., Farquhar, G., Nardelli, N., Rudner, T. G., et al. (2019). The starcraft multi-agent challenge. arXiv preprint arXiv:1902.04043.

Shoham, Y., & Leyton-Brown, K. (2008). *Multiagent systems: Algorithmic, game-theoretic, and logical foundations*. Cambridge University Press.

Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., et al. (2018). A general reinforcement learning algorithm that masters chess, Shogi, and go through self-play. *Science*, *362*(6419), 1140–1144.

Singh, B., Kumar, R., & Singh, V. P. (2022). Reinforcement learning in robotic applications: A comprehensive survey. *Artificial Intelligence Review*, *55*(2), 945–990.

Skrynnik, A., Andreychuk, A., Nesterova, M., Yakovlev, K., & Panov, A. (2024). Learn to follow: Decentralized lifelong multi-agent pathfinding via planning and learning. *vol. 38*, In *Proceedings of the AAAI conference on artificial intelligence* (pp. 17541–17549).

Skrynnik, A., Andreychuk, A., Yakovlev, K., & Panov, A. (2024). Decentralized Monte Carlo tree search for partially observable multi-agent pathfinding. *vol. 38*, In *Proceedings of the AAAI conference on artificial intelligence* (pp. 17531–17540).

Sutton, R. S. (1990). Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. *vol. 1990*, In *Machine learning proceedings* (pp. 216–224). Elsevier.

Vaswani, A. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*.

Venugopal, A., Milani, S., Fang, F., & Ravindran, B. (2023). Bi-level latent variable model for sample-efficient multi-agent reinforcement learning. arXiv preprint arXiv:2304.06011.

Vinyals, O., Babuschkin, I., Czarnecki, W. M., Mathieu, M., Dudzik, A., Chung, J., et al. (2019). Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, *575*(7782), 350–354.

Wang, X., Zhang, Z., & Zhang, W. (2022). Model-based multi-agent reinforcement learning: Recent progress and prospects. arXiv preprint arXiv:2203.10603.

Xie, A., Losey, D., Tolsma, R., Finn, C., & Sadigh, D. (2021). Learning latent representations to influence multi-agent interaction. In *Conference on robot learning* (pp. 575–588). PMLR.

Ye, W., Liu, S., Kurutach, T., Abbeel, P., & Gao, Y. (2021). Mastering atari games with limited data. *Advances in Neural Information Processing Systems*, *34*, 25476–25488.

Yi, Y., Li, G., Wang, Y., & Lu, Z. (2021). Learning to share in multi-agent reinforcement learning. arXiv preprint arXiv:2112.08702.

Yu, C., Velu, A., Vinitsky, E., Gao, J., Wang, Y., Bayen, A., et al. (2022). The surprising effectiveness of ppo in cooperative multi-agent games. *Advances in Neural Information Processing Systems*, *35*, 24611–24624.

Zhang, L., Sun, Y., Barth, A., & Ma, O. (2020). Decentralized control of multi-robot system in cooperative object transportation using deep reinforcement learning. *IEEE Access*, *8*, 184109–184119.