Critic-over-Actor-Critic Modeling: Finding Optimal Strategy in ICU Environments

Riazat Ryan
Department of Computer and Information Science
University of Massachusetts Dartmouth
North Dartmouth, USA
rryan2@umassd.edu

Ming Shao
Department of Computer and Information Science
University of Massachusetts Dartmouth
North Dartmouth, USA
mshao@umassd.edu

2022 IEEE International Conference on Big Data (Big Data) | 978-1-6654-8045-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/BigData55660.2022.10021125

***Abstract*—Reinforcement learning (RL) is mechanized to learn from experience. It solves the problem in sequential decisions by optimizing reward-punishment through experimentation of the distinct actions in an environment. Unlike supervised learning models, RL lacks static input-output mappings and the objective of minimization of a vector error. However, to find out an optimal strategy, it is crucial to learn both continuous feedback from training data and the offline rules of the experiences with no explicit dependence on online samples. In this paper, we present a study of a multi-agent RL framework which involves a Critic in semi-offline mode criticizing over an online Actor-Critic network, namely, Critic-over-Actor-Critic (CoAC) model, in finding optimal treatment plan of ICU patients as well as optimal strategy in a combative battle game. For further validation, we also examine the model in the adversarial assignment.**

***Index Terms*—Actor-critic, healthcare trajectory, adversarial teaming, multi-agent RL, optimal treatment recommendation.**

I. Introduction

Reinforcement Learning (RL) is a model where an agent learns through optimizing its behavior while interacting with the environment. Agent receives rewards or punishment as feedback for a taken action, and thus, the model quantifies the action. A policy in the network learns through these steps and decides an action at certain states. The value function approximates the next return. The main goal of the agent in the learning is to find an optimal policy which can maximize the total reward until the terminal state. Among two major paradigms of RL, single agent problems are mostly used. Single agent RL model behaves the same way without REINFORCE as in a group [1].

However, in a practical scenario, REINFORCE [2] takes place whether it is competitive or cooperative while other multiple agents present in an interactive environment. More than one agent creates a dynamic grid where agents will be learning at the same time on the same or different objective [2]. A Multi-Agent Reinforcement Learning (MARL) model potentially leads to more realistic and complex non-stationary scenarios.

Deep Reinforcement Learning (DRL), including both learning styles, i.e., Single and Multi-Agent, has lately made exciting advancement in diverse domains. With Atari games [2], Go [3], continuous control tasks [4], it has shown human-like performance. The work in [5] solved a multi-agent challenge with a cooperative model. In terms of policy, MARL showed a different point of view by changing other agents policy to learn a new task [6]. MARL is also consistent with discrete and continuous actions of the agents [7]. In the mixed environment, like competitive-cooperative, MARL achieved delicate admission. In adversarial learning, too, multi-agent model has committed established work [8].

Despite numerous accomplishments, MARL suffers from distinct setbacks. In MARL, all agents apparently learn inter-collectively with no external guidance [9]. In addition, each agent learns, and its policy optimizes as training progresses. But at the simulation, it faces non-stationary environment challenges, which creates a learning instability [10]. MARL also lacks an explore-exploit duo action together at a time for greater rewards, i.e., an overall observer for the network's policy [11]. Traditional RL also prevents the straightforward usage of exploration of new actions because of its greedy character [12].

Our approach extends prior works in a number of ways. The main idea is to learn a centralized actor with two critic units where one critic, semi-offline in nature, has no rewards on its way, and another one criticizes actor's performance through value error. We name this new approach *Critic-over-Actor-Critic (CoAC)*. The perception behind our idea comes from the fact that, in many real-world environments, agents may have to interact with both online and offline environments with or without any prior experience. Also, the major concern of our study is to find a way to make better use of RL models in health-care where the circumstance demands to learn the collective rules along with experiences. Investigating optimal treatment trajectory in the clinical environment involves modeling patient-level temporal healthcare processes in state-action space and learning a generic way to understand clinical knowledge. The current practice of standard multi-agent reinforcement learning hardly takes these dynamics into account. Our CoAC approach is able to dynamically select which critic network to attend at a time during the simulation.

Our proposed CoAC approach offers:

- Actor-Critic couple where a supplemental critic explores action set and can guide the actor in semi-offline mode;
- a composite policy of agents' interaction in online-semi offline environment;

978-1-6654-8045-1/22/$31.00 ©2022 IEEE

- presence of an adversarial entity in the environment;
- an environment occupied with cooperative and competitive agents;

and they are applicable and beneficial to a multi-agent scenario in finding episode-wise optimal strategies. We have validated our proposed model on two different simulated environments and tasks and compared it with prior works.

II. Recent Works

A. Multi-Agent and Actor-Critic Models

Multi-Agent Reinforcement Learning (MARL) is an extensively studied model. It covers a big spectrum of distinctive problems, ranging from cooperative task [7], self-organizing swarm robots [13] to high dimensional continuous state-space model [14]. MARL promises to operate in dynamic environment [6], in loosely coordinated cooperative settings [7], in group rewards [5], also in learning other agents' policies [15].

The heart of our proposed model is an Actor-Critic model. Actor-Critic (AC) method is one of the well-practiced techniques in RL [16]. In general, reinforcement learning algorithms either optimize on learning a value function [12], like value iteration and TD-learning [16], or learning a policy directly [16]. AC methods learn both simultaneously - the actor being the policy and the critic being the value function. Recent practices of Actor-Critic framework comprise of centralized learning [6], attention in centralized [10], mixed reward [7].

In sequential decision making, among Deep RL models, Actor-Critic comes as a distinct instrument. [17] presented a model which learns from limited experiences domain using a value-based decisive mechanism. [4] used AC itself as an optimizer to a recurrent model in synthesizing a multi-label sequence. Actor-Critic has also been utilized adversarial unit. [8] implemented Actor and Critic in an adversarial manner to a learning module. Current focus of RL has deeply become autonomous driving. An Actor-Critic model [18] exhibited how it can be feasible in making decision for short time periods. In our case, the multi-agent comes with an incentive to manifest an optimal plan for multi-disease problems from two points of view.

B. RL in Clinical Environments

As of targeting to find an optimal strategy in a multi-agent setup, our primary objective is to bring a simulated ICU environment based on MIMIC III [19] to the RL agents to figure out the optimal treatments of multi-disease patients. With the impression of usage of RL models in healthcare, model-based and model-free algorithms have been applied in diverse clinical problems. [20], [21] proposed simple Q-learning models to find out cancer treatment therapy. Despite the fact of missing the temporal information in state space, [22] has successfully shown the advantage of Actor-Critic model and how the temporal difference loss of the model can help get a better AI treatment plan.

Single agent reinforcement learning models are also capable in finding the optimal plan. Work in [23] showed a single agent could plan for the treatment of sepsis patients in a model-based DQN network. With different rewards and clinical goals, [24] studied a Deep-RL model for sepsis patients. With the continuous state-action plan, [25] brought a Q learning based RL model exploring the nature of the treatment of sepsis ICU patients. One of the prominent works of off-policy learning in clinical environment adopted the Actor-Critic method to determine continuous medical decisions [26]. Multi-agent models are also adherent to clinical conditions. [27] proposed a context-ware policy gradient based multi-agent model which learns through joint reward actions. With the sub-speciality domain information, several agents interacting in a single environment can find precise treatment for oncology patients [28] and have drawn much attention lately. Typically in clinical problems, existing RL methods handle just one disease environment with the online rule. The proposed architecture plans to find the clinical rule as well as RL strategies for optimal treatment in multi-disease space.

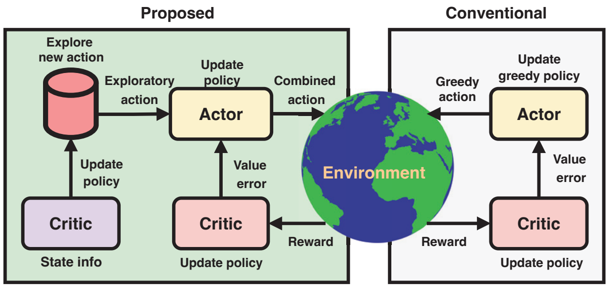

Fig. 1: Left: the proposed model where the new critic explores the new action set. Right: existing Actor-Critic model.

III. Methodology

Our proposed network is comparatively more flexible than the prior approaches of MARL. Our algorithm is able to train policies in ICU and simulated environments with group rewards and greedy rewards. Agent can adopt different action spaces. Duo critics attend the actor individually. Thus, our approach achieves scalability in terms of the number of agents and can adapt to different types of environments with ease.

Figure 1 visualizes our proposed Critic-over-Actor-Critic (CoAC) architecture against the conventional Actor-Critic model. The core learning part includes an Actor-Critic network. The Actor makes a decision of a greedy action based on the value error from the generic Critic. The Critic on the far left has exploratory nature to find an action. This Critic behaves the same way as a critic from an Actor-Critic. We have modeled the value error to the corresponding actions through an exploratory model parameter. Therefore, the Actor and the top Critic together decide a combined action to take to perform in the environment.

A. MDP Formulation

The environment also provides an evaluation of the action called the immediate reward r. The agents and states are modeled using MDP [16], and the formulation follows:

- S: a continuous state space to describe the user state. The state of a user at timestamp t can be represented as $s_t \in S$ ($t \in T$ and $t > 0$).
- a: represents discrete and continuous actions space. The action $a \in A$ of the agent is to recommend the selected item. In training, action a_t at timestamp $t, a_{t+1} \in A_{x+1}$ where x is the agents' observation space.
- P: $S \times A$ states transition probability.
- R: $S \times A \rightarrow R$ is the reward function where (s, a) denotes the immediate reward r by taking action a at state s.
- γ: discount factor in the learning.

B. Value Error, Action and Rewards

The job of the generic critic with the environment is to observe states and rewards and to build a value function, $V^{\pi}(s)$ that accounts for both immediate and future rewards received under the deterministic policy π. This value function [7] is defined as:

$$V^{\pi}(s) = \sum_{s \in S} Pr(s|s_0, a)R(s) + \gamma V^{\pi}(s_0), \tag{1}$$

where $R(s)$ is the expected value of r, $\gamma \in [0, 1]$ is a factor that discounts the value of the next state, and $Pr(s|s_0, a)$ is the probability of transitioning to state s after taking action $a = \pi(s)$ [10]. We adopt Temporal Difference (TD) [7], a common approach to updating the state-value estimates $V(s)$, by an amount proportional to the TD error [7]. This is defined as:

$$\delta = r + \gamma V(s) - V(s_0). \tag{2}$$

For deterministic policy and valued actions, a model parameter composes the action as a weighted sum of the actions given by the actor's policy and the semi-offline critic's exploratory policy's [11] action:

$$a = ba^G + (1 - b)a^E, \tag{3}$$

where a^E is the critic's exploratory action and, a^G is the actor's greedy action, as given by policies π^E and π^G, respectively. b is a model parameter values between $[0, 1]$.

C. Policy Update, Other Parameters and Learning

The semi-offline critic optimizes its policy by targeting the variance between the current and terminal state through the probability of using the exploratory actions. The network starts with a random action and then converges to the action which has more exploratory rewards:

$$\pi^E(a^E|s) = \Gamma \pi^{rand}(a^E_s|T^{ae}). \tag{4}$$

Here, a^E_s and T^{ae} are exploratory actions in the current and terminal states, respectively. π^{rand} initially starts with an indiscriminate value and gradually converges into exploratory policy. Γ is the number of agents in the current state s.

From the generic critic side, the actor receives a direction through its own policy and a guided error estimation:

$$\eta = \frac{1}{2}[\pi(s) - \pi^G(s)]^2. \tag{5}$$

Algorithm 1: Critic-over-Actor-Critic (CoAC).

```
Input:
$V(s)$: critic value function
$\pi^G(s)$: actor policy
$\sigma$: exploration factor
$\alpha$: actor steps; $\beta$: critics step size
$\gamma \in [0, 1]$: discount factor
$\lambda$: decay factor
Output: optimal trajectory of the agent
repeat for each episode
    $e \leftarrow 0$, reset agents path
    $s \leftarrow$ initial state
    repeat for each step of the episodes
        $a^G \leftarrow$ by the actor using policy $\pi^G$
        $a^E \leftarrow a^E + N(0, \sigma)$ exploratory action by the critic
          using policy $\pi^E$
        $a = ba^G + (1 - b)a^E$, combined action
        update $e \leftarrow \gamma\ \lambda\ e + V(s)$
        take action $a$, observe reward $r$ and next state be $s'$
        $\delta \leftarrow r + \gamma\ V(s') - V(s)$
        $s \leftarrow s'$
        critic policy $\pi$ updates
    till $s$ is the terminal state
end episode
```

Actor's policy is basically a greedy policy [10]. But, it learns the joint policy π^A, from both the critics. Therefore, we have:

$$\pi^A = \begin{cases} \frac{c}{a^g}, & \text{if } a^g \in A \text{ is a greedy action} \\ 1 - c + \frac{c}{a^g}, & \text{otherwise.} \end{cases} \tag{6}$$

The parameter c here is arbitrarily small and tends to optimize into greedy policy. We assume that π^G is computed by a model function approximator with the parameter vector w, and after each state transition, those parameters are updated according to a rule [9]:

$$\begin{aligned} & w_{s+1} = w_s + w_s * \eta + k\Delta w^{RL} + (1 - k)\Delta w^{SL}, \\ & w^{SL} = \beta \Delta E(s), \end{aligned} \tag{7}$$

where Δw^{RL} and Δw^{SL} are the individual updates based on RL and critic random action, respectively. k is a trade-off of two actions which apparently interpolates between two policies of learning. The actor updates its equation to make the reinforcement-based adjustment to the parameters of its policy π^A, which is computed as [9]:

$$\Delta w^{RL} = \alpha \delta (a^E - a^G) \Delta w \pi^A(s), \tag{8}$$

where α and β are step-size-parameter and is updated by the proposed CoAC algorithm detailed in Algorithm 1.

IV. Experiment

Our experiments are outlined in two environments. The first one is an ICU environment built up as the miniature version from the MIMIC III ICU dataset [19]. The second environment is a simulation in functionality where we have picked a 2-team battle game from PettingZoo library [29].

A. MIMIC III ICU Environment

The first environment is a multi-disease state space, namely, Sepsis, Kidney and Heart disease critical patients and their physiological conditions. We have imputed the missing information as per their way [30]. Our ICU environment is custom-built (with two more clusters of patients) using a sepsis environment [31]. Details of this environment are elaborated as follows:

State: A patient's state is composed of features [24], including respiration rate, heart rate, arterial pH, positive end-expiratory pressure, oxygen saturation, inspired oxygen fraction, arterial oxygen partial pressure, plateau pressure, average airway pressure, mean non-invasive blood pressure, SP02, Fe. Each state consists of 14 normalized features from the Sepsis, Kidney, and Heart patients list of MIMIC III. We extract about 1200 admissions from adult patients for the state-action space.

Action: The actions are normalized into two treatment mediums: Vasopressor (VI) and IV fluids (IV). Each set has 12 individual treatments. Actions are continuous between 0 and 1, indicating possible vasopressor and IV fluid interventions across 5 dosage quantiles [24]. The actor network and the critic in Figure 1 have ReLU activation function and in the output layer, it is a Softmax function. The critic on the top uses the ReLU activation function, but has no activation function in the output layer [7].

Target: The end goal is to leverage this clinical information to find a treatment action for each time step based on the information given that any patient at a particular time step should survive until the terminal state. The agents will keep trying with actions until they reach the terminal state.

Reward: The reward function r_{t+1} is defined as $r_{t+1} = r_{vital(t+1)} + r_{ventoff(t+1)} + r_{venton(t+1)}$ in which $r_{vital(t+1)}$ evaluates the effect of the actions on the states of the patients. $r_{ventoff(t+1)}$ estimates the performance of ventilation being stopped at time $t+1$. Also, rewards are based on patients survivability and the equal-balanced doses of VI and IV over the time period. For each disease, we have set a threshold to measure the severity of the patients' conditions and the agents of the corresponding reward take that into account. In addition, rewards segregated the experiments into two parts: (1) group rewards; (2) greedy rewards. First, when using group rewards, agents in the same group help each other to reach the terminal state T using the factor r_{groups}, which tends to check the maximum rewards for the entire team as a single unit in the MDP. Second, regarding greedy rewards, it is the default setup of MDP to crave for the highest rewards as an individual agent.

Agents: For each disease, there are separate agents: Sepsis agent, Kidney agent, and Heart agent. They are defined separately with their own rewards but with the same action set. Since the rewards, discount factor γ, and continuous action values are different, they behave distinctively in the grid. At the terminal state T, three agents will have three separate action recommendations for each group of patients.

Episode Simulation and Testing Simulation: The grid is defined as such to detect episode transitions. The transaction between states takes place based on the binary condition of VI or IV [25]. The testing grid has the same features and architecture as the episode simulation. However, it may have expired or dispatched from hospital info. This model was used in the environment to decide the reward values at the end of each episode [25].

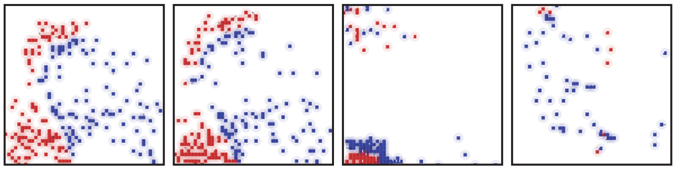

Fig. 2: 2-team battle game with no adversarial agents using our model. Red has to tag blue agents, and blue has to do the opposite. Cooperative is nature. This environment brings more survivability and more agents presented until the end.

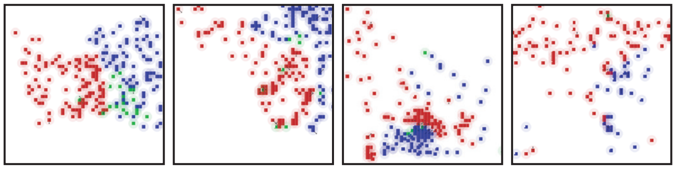

Fig. 3: In this battle, the green agents try to tag both red and blue agents. In that too, our RL agents could eliminate the green agents while tagging opponents through decent reward and survival rate.

Adversarial Environment: For further validation of the proposed study, we experimented on an adversarial environment setting, too. With the ICU grid, we have doubled the size of the no-transaction states in the simulation and let the agents decide within the same action lists.

B. PettingZoo Environment

Our proposed model has performed on PettingZoo [29] environment, too. PettingZoo is a Python-based library of many diverse multi-agent reinforcement learning environments. For the sake of the experiments, we have adopted its *Adversarial Pursuit* environment from the MPE library and then modified it as per our need. Our changes took place and converted the grid as shown in Figure 2.

Action: It is a discrete action tagging game. One team has to tag the opponent team agents. Agents can move high-handedly in the grid. Once one team tags all agents of the opponent team, the game is over.

Reward: Predator's reward is: 1 reward for tagging a prey; -0.2 reward for tagging anywhere. Prey's reward is: -1 reward for being tagged.

Observation space: The observation space is a 10×10 map for pursuers and a 9×9 map for the pursued.

Adversarial Agent: In the adversarial nature simulation, we have introduced an adversarial agent (green agents as shown in Figure 3) in the environment whose action is to tag both teams.

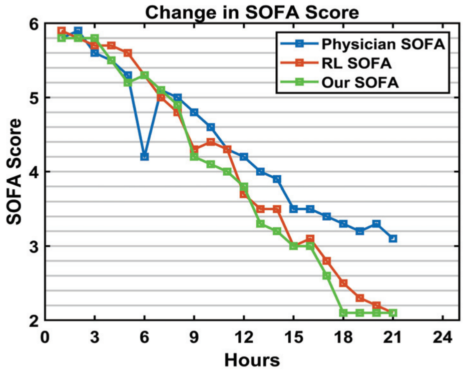

Fig. 4: Comparison of changes of SOFA score among Physician, RL and Our model in the ICU Environment. Over the progress of time, all manage to reduce the score. The proposed model could reach recovery (SOFA < 3) before Physicians and RL agents could.

TABLE I: Our Model vs. Baseline Models. CoAC is compared to other multi-agent models with two discrete learning methods where the proposed model demonstrated high competence

PettingZoo				
Metric	MA-DQN	MA-SAC	Greedy-Ours	Group-Ours
Rewards	0.5903	0.5874	0.6837	**0.6928**
Survival Rate	**0.7244**	0.6317	0.6745	0.6601
% of Agents at T	**0.8864**	0.8251	0.8819	0.8720

Sepsis-Kidney-Heart				
Metric	MA-DQN	MA-SAC	Greedy-Ours	Group-Ours
Rewards	0.7383	0.7751	0.7992	**0.8014**
Survival Rate	0.8041	**0.8625**	0.8227	0.8608
% of Agents at T	0.8035	0.8270	0.8596	**0.8629**

TABLE II: Adversary: Our Model vs Baseline Models. With adversarial property in hand, CoAC holds the performance in both environments.

Adversarial-Sepsis-Kidney-Heart				
Metric	MA-DQN	MA-SAC	Greedy-Ours	Group-Ours
Rewards	0.5351	0.5373	0.6528	**0.6845**
Survival Rate	**0.7891**	0.7255	0.7818	0.7730
% of Agents at T	0.7017	0.6250	**0.7933**	0.7453

Adversarial-PettingZoo				
Metric	MA-DQN	MA-SAC	Greedy-Ours	Group-Ours
Rewards	0.5125	0.5208	**0.6533**	0.6187
Survival Rate	0.6593	**0.6817**	0.6152	0.6046
% of Agents at T	0.7377	0.7689	**0.7943**	0.7315

V. Results and Discussion

To examine the instrumental suitability of the proposed model, we quantified (1) rewards, (2) survival rate, (3) number of agents at the terminal state, (4) SOFA Score against baseline multi-agent RL models and physician.

A. ICU Environment

The goal of the agents in the ICU environment is to maximize survivability while keeping a balance of equal doses between IV and VI [25] (IV and VI balance the blood pressure of a patient and prevent organ failure [24]). Among three stages of sepsis progression (Progression is determined by SOFA Score that assesses the severity of sepsis [24]), our proposed framework balanced the applied doses of IV and VI through low, mid, and high SOFA score levels.

Episodic change of SOFA: In the episodic change of SOFA score in Figure 4, CoAC has achieved consistently better performance than the physician guidance and other RL agents. Before reaching the terminal state, RL agents gained minimal discharge (i.e., SOFA<3) SOFA score. In the interim, agents suffer through the trade-off between exploration and exploitation although with the presence of semi-offline critic. However, greedy actions from the actor network eventually credit on the SOFA points. Early hours experience slow progress of patients clinical condition in terms of SOFA for slow policy gradient [9].

Rewards, Survival Rate, and Agents: Overall advantages of the proposed model over baseline RL algorithms contemplate in total rewards, survival rate, and how many agents can reach the terminal state, as shown in Table I. Sepsis-Kidney-Heart ICU environment has been tested against Multi-Agent DQN [12] and Multi-Agent Soft Actor-Critic model (SAC) [11]. With our model, we have examined the environment with two reward setup.

DQN and SAC total reward collection for all the agents are the same overall. In the group reward, Critic-Actor-Critic gains the maximum reward among all four simulations. Agents interaction with each other with the nature of exploration from the extra critic might have added the privilege here [9]. However, in case of survival rate, DQN surpasses others. The simplest nature of Q learning overtook the benefit of exploration nature of the critic [13]. Now, agents reached the terminal state seem to be a major favor from the semi-offline Critic. The new Critic exploration nature arranges a way to move most of the agents until they get to the terminal state [9]. Our model could arrive at the end state with 86% of agents in group rewards and 85% with greedy rewards approach.

Adversary: For the adversarial demonstration and validation in Table II, the simulation-grid size for the agents has been doubled with more no-transition states and less survival stages. The investigation flows the same path. However, the adversarial nature decreases the performance of the matrices in a slight margin compared to the traditional model-based RL networks. Even with the adversary, the model carried on the same trend. Compared to other multi-agent models, agents in the proposed network could achieve more than 10% of rewards. In survivability, Critic-over-Actor-Critic took the lead with a greedy approach at par with DQN. However, at the end, greedy, like the previous test, reached with the highest number of agents.

B. PettingZoo Environment

The game environment of our experiments also attained remarks with the proposed model. In this evaluation, our model is compared to the same baseline models. With predator-prey (red and blue agents in Figure 2), DQN and SAC have significant (around 15%) less performance in terms of rewards than our framework. In survival rate, because of the straightforward MDP formulation [12], DQN again achieves the best score here. Both group and greedy reward of our model performed equally strong in the survivability scale. PettingZoo, too, has more agents landed at the terminal state through our CoAC model.

Adversary: For the adversary in PettingZoo, we have the green agents tagging both red and blue agents in Figure 3. Green agents follow the game rule only. With the presence of adversarial agent, the proposed model could hold the same effects. In rewards, the agents in the model profess the same as without the adversary. With survival rate, our Critic-over-Actor-Critic is nearly at par with SAC model in Table II. Greedy reached the terminal state with the highest number of agents. Group rewards model compensated its team members with co-operation [12].

VI. Conclusions

The crucial setback of an Actor-Critic model is the lack of freedom to have a non-supervisory action which can support the Actor network to decide on an behavior with no feedback. The semi-offline critic in this proposed model strengthened the Actor-Critic network, saved the actor from over-doing and normalized the final action with a curious look into the environment. Our Critic-over-Actor-Critic model showed meaningful advantages in finding optimal strategy in both cooperative and combative reinforce domains.

Acknowledgement

This work is supported in part by the UMass Dartmouth's Marine and Undersea Technology (MUST) Research Program funded by the Office of Naval Research (ONR) under Grant No. N00014-20-1-2170 and National Science Foundation under Grant No. 2144772.

References

[1] M. Zinkevich and T. Balch, "Symmetry in markov decision processes and its implications for single agent and multi agent learning," in *In Proceedings of the 18th International Conference on Machine Learning*. Citeseer, 2001.

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

[3] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel *et al.*, "A general reinforcement learning algorithm that masters chess, shogi, and go through self-play," *Science*, vol. 362, no. 6419, pp. 1140–1144, 2018.

[4] S. Najafi, C. Cherry, and G. Kondrak, "Efficient sequence labeling with actor-critic training," in *Canadian Conference on AI*. Springer, 2019, pp. 466–471.

[5] M. Samvelyan, T. Rashid, C. S. De Witt, G. Farquhar, N. Nardelli, T. G. Rudner, C.-M. Hung, P. H. Torr, J. Foerster, and S. Whiteson, "The starcraft multi-agent challenge," *arXiv preprint arXiv:1902.04043*, 2019.

[6] J. Foerster, G. Farquhar, T. Afouras, N. Nardelli, and S. Whiteson, "Counterfactual multi-agent policy gradients," in *AAAI*, vol. 32, no. 1, 2018.

[7] 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 NeurIPS*, vol. 30, 2017.

[8] T. Oikarinen, W. Zhang, A. Megretski, L. Daniel, and T.-W. Weng, "Robust deep reinforcement learning through adversarial loss," *Advances in NeurIPS*, vol. 34, 2021.

[9] I. Kostrikov, R. Fergus, J. Tompson, and O. Nachum, "Offline reinforcement learning with fisher divergence critic regularization," in *ICML*. PMLR, 2021, pp. 5774–5783.

[10] S. Iqbal and F. Sha, "Actor-attention-critic for multi-agent reinforcement learning," in *ICML*. PMLR, 2019, pp. 2961–2970.

[11] T. Haarnoja, A. Zhou, K. Hartikainen, G. Tucker, S. Ha, J. Tan, V. Kumar, H. Zhu, A. Gupta, P. Abbeel *et al.*, "Soft actor-critic algorithms and applications," *arXiv preprint arXiv:1812.05905*, 2018.

[12] M. Lapan, *Deep Reinforcement Learning Hands-On: Apply modern RL methods, with deep Q-networks, value iteration, policy gradients, TRPO, AlphaGo Zero and more*. Packt Publishing Ltd, 2018.

[13] S. Abdallah and V. Lesser, "Multiagent reinforcement learning and self-organization in a network of agents," in *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, 2007, pp. 1–8.

[14] J. K. Gupta, M. Egorov, and M. Kochenderfer, "Cooperative multi-agent control using deep reinforcement learning," in *International conference on autonomous agents and multiagent systems*. Springer, 2017, pp. 66–83.

[15] H. He, J. Boyd-Graber, K. Kwok, and H. Daumé III, "Opponent modeling in deep reinforcement learning," in *ICML*. PMLR, 2016, pp. 1804–1813.

[16] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," *Advances in NeurIPS*, vol. 12, 1999.

[17] Â. G. Lovatto, T. P. Bueno, D. D. Mauá, and L. N. Barros, "Decision-aware model learning for actor-critic methods: when theory does not meet practice," 2020.

[18] W. Song, S. Liu, Y. Li, Y. Yang, and C. Xiang, "Smooth actor-critic algorithm for end-to-end autonomous driving," in *2020 ACC*. IEEE, 2020, pp. 3242–3248.

[19] A. E. Johnson, T. J. Pollard, L. Shen, L.-w. H. Lehman, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. Anthony Celi, and R. G. Mark, "Mimic-iii, a freely accessible critical care database," *Scientific data*, vol. 3, no. 1, pp. 1–9, 2016.

[20] Y. Zhao, M. R. Kosorok, and D. Zeng, "Reinforcement learning design for cancer clinical trials," *Statistics in medicine*, vol. 28, no. 26, pp. 3294–3315, 2009.

[21] A. Hassani *et al.*, "Reinforcement learning based control of tumor growth with chemotherapy," in *2010 ICSSE*. IEEE, 2010, pp. 185–189.

[22] I. Ahn and J. Park, "Drug scheduling of cancer chemotherapy based on natural actor-critic approach," *BioSystems*, vol. 106, no. 2-3, pp. 121–129, 2011.

[23] M. Komorowski, L. A. Celi, O. Badawi, A. C. Gordon, and A. A. Faisal, "The artificial intelligence clinician learns optimal treatment strategies for sepsis in intensive care," *Nature medicine*, vol. 24, no. 11, pp. 1716–1720, 2018.

[24] A. Raghu, M. Komorowski, I. Ahmed, L. Celi, P. Szolovits, and M. Ghassemi, "Deep reinforcement learning for sepsis treatment," *arXiv preprint arXiv:1711.09602*, 2017.

[25] A. Raghu, M. Komorowski, L. A. Celi, P. Szolovits, and M. Ghassemi, "Continuous state-space models for optimal sepsis treatment: a deep reinforcement learning approach," in *Machine Learning for Healthcare Conference*. PMLR, 2017, pp. 147–163.

[26] L. Li, M. Komorowski, and A. A. Faisal, "The actor search tree critic (astc) for off-policy pomdp learning in medical decision making," *arXiv preprint arXiv:1805.11548*, 2018.

[27] R. E. Allen, J. K. Gupta, J. Pena, Y. Zhou, J. W. Bear, and M. J. Kochenderfer, "Health-informed policy gradients for multi-agent reinforcement learning," *arXiv preprint arXiv:1908.01022*, 2019.

[28] J.-N. Eckardt, K. Wendt, M. Bornhäuser, and J. M. Middeke, "Reinforcement learning for precision oncology," *Cancers*, vol. 13, no. 18, p. 4624, 2021.

[29] J. K. Terry, B. Black, N. Grammel, M. Jayakumar, A. Hari, R. Sullivan, L. Santos, C. Dieffendahl, C. Horsch, R. Perez-Vicente *et al.*, "Pettingzoo: Gym for multi-agent reinforcement learning," *Advances in NeurIPS*, vol. 34, 2021.

[30] R. Ryan, H. Zhao, and M. Shao, "Ctc-attention based non-parametric inference modeling for clinical state progression," in *2019 IEEE International Conference on Big Data (Big Data)*. IEEE, 2019, pp. 145–154.

[31] A. Kiani, C. Wang, and A. Xu, "Sepsis world model: A mimic-based openai gym" world model" simulator for sepsis treatment," *arXiv preprint arXiv:1912.07127*, 2019.