Mathematical Biosciences and Engineering

http://www.aimspress.com/journal/mbe

MBE, 19(8): 7933–7951
DOI: 10.3934/mbe.2022371
Received: 29 November 2021
Revised: 01 April 2022
Accepted: 01 April 2022
Published: 27 May 2022

Research article

An autonomous agent for negotiation with multiple communication channels using parametrized deep Q-network *

Siqi Chen*and Ran Su*

College of Intelligence and Computing, Tianjin University, Tianjin 300072, China

* **Correspondence:** Email: siqichen@tju.edu.cn, ran.su@tju.edu.cn.

Abstract: Agent-based negotiation aims at automating the negotiation process on behalf of humans to save time and effort. While successful, the current research considers communication between negotiation agents through offer exchange. In addition to the simple manner, many real-world settings tend to involve linguistic channels with which negotiators can express intentions, ask questions, and discuss plans. The information bandwidth of traditional negotiation is therefore restricted and grounded in the action space. Against this background, a negotiation agent called MCAN (multiple channel automated negotiation) is described that models the negotiation with multiple communication channels problem as a Markov decision problem with a hybrid action space. The agent employs a novel deep reinforcement learning technique to generate an efficient strategy, which can interact with different opponents, i.e., other negotiation agents or human players. Specifically, the agent leverages parametrized deep Q-networks (P-DQNs) that provides solutions for a hybrid discrete-continuous action space, thereby learning a comprehensive negotiation strategy that integrates linguistic communication skills and bidding strategies. The extensive experimental results show that the MCAN agent outperforms other agents as well as human players in terms of averaged utility. A high human perception evaluation is also reported based on a user study. Moreover, a comparative experiment shows how the P-DQNs algorithm promotes the performance of the MCAN agent.

Keywords: multi-agent systems; cooperative games; reinforcement learning; deep learning; human-agent interaction

* This article is a substantially extended version of our paper [1] presented at the IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2021). The extension concerns both the description and evaluation of the agent. As regards the description, all relevant aspects of our approach are shown in detail. As regards the evaluation, the experimental technicalities including negotiating opponents have been extended significantly. Moreover, a user study is conducted to evaluate how human players like the MCAN agent using the evaluation metric of [2] and a high human perception evaluation is reported based on a user study. Furthermore, a comparative analysis shows how the P-DQN algorithm promotes the performance of the MCAN agent.

1. Introduction

Negotiation provides a fundamental and powerful mechanism for managing interaction among computational agents [3], and it plays a central role in the field of distributed artificial intelligence. Effective communication is crucial in a negotiation [4–7], as the negotiators need to exchange information about their desires, infer their opponent's desires from communication, and balance between the two. To reach a successful negotiation, negotiation agents are thus required to employ a comprehensive strategy that integrates linguistic communication skills and bidding strategies, especially when interacting with humans. However, most work in the field of automated negotiation [8–11] typically uses offers/counter-offers to simplify the process, thereby ignoring linguistic communication in negotiation. While some work [12–15] considers negotiation with linguistic communication, those studies focus on the dialogue system that generates negotiation dialogues using natural language processing and incorporates the offer generation as a low-level strategy. This kind of work regards linguistic communication as a medium for conveying offer information and ignores the ability of linguistic communication to express other aspects, such as emotions and intentions. As a matter of fact, a high degree of linguistic communication skills can bring good negotiation experience and help express intentions better, thus speeding up the negotiation process.

Therefore, to model a negotiation that both supports offer and linguistic communication channels, this work follows a variant of the alternative offer protocol [16, 17] where the negotiators can simultaneously exchange offers and linguistic information (in terms of cheaptalk). Based on this protocol, we propose a novel framework called multiple channel automated negotiation (MCAN) that leverages parametrized deep Q-networks (P-DQNs) to learn a comprehensive negotiation strategy which integrates linguistic communication skills and bidding strategies at the same time. Combining the advantage of deep Q-networks (DQNs) and a deep deterministic policy gradient (DDPG), P-DQNs, which learns the optimal bidding strategy on each kind of linguistic communication skills and then chooses the co-optimal one, is suitable for this dual optimization problem. Unlike other algorithms that learn linguistic communication skills and bidding strategies separately, P-DQNs learns these two at the same time, which enables linguistic communication skills learning and bidding strategy learning to share all useful information without the need to exchange information between separate networks.

The framework is shown in Figure 1, which models the negotiation process as a Markov decision process (MDP). The MCAN agent takes the opponent's proposal (e.g., cheaptalk and offer) as input and sends it to the proposal resolver. After updating the state, the deep reinforcement learning (DRL) policy outputs an action to the proposal generator guiding the optimal proposal. To train this agent, we designed several rule-based negotiation agents depending on classic bidding strategies and different linguistic communication skills. The experimental results show that the MCAN agent outperforms those handcrafted agents according to the averaged utility. Furthermore, user studies are conducted in a GUI human-agent negotiation environment. We evaluate the MCAN agents and other rule-based negotiation agents based on the metrics introduced in [2]. The experimental results of negotiation with human negotiators show that the MCAN agent has successfully learned a comprehensive negotiation strategy that integrates linguistic communication skills and bidding strategies, even trained with simple rule-based agents.

The main contributions of the work are as follows:

- To the best of our knowledge, the first DRL-based negotiation agent MCAN is proposed which

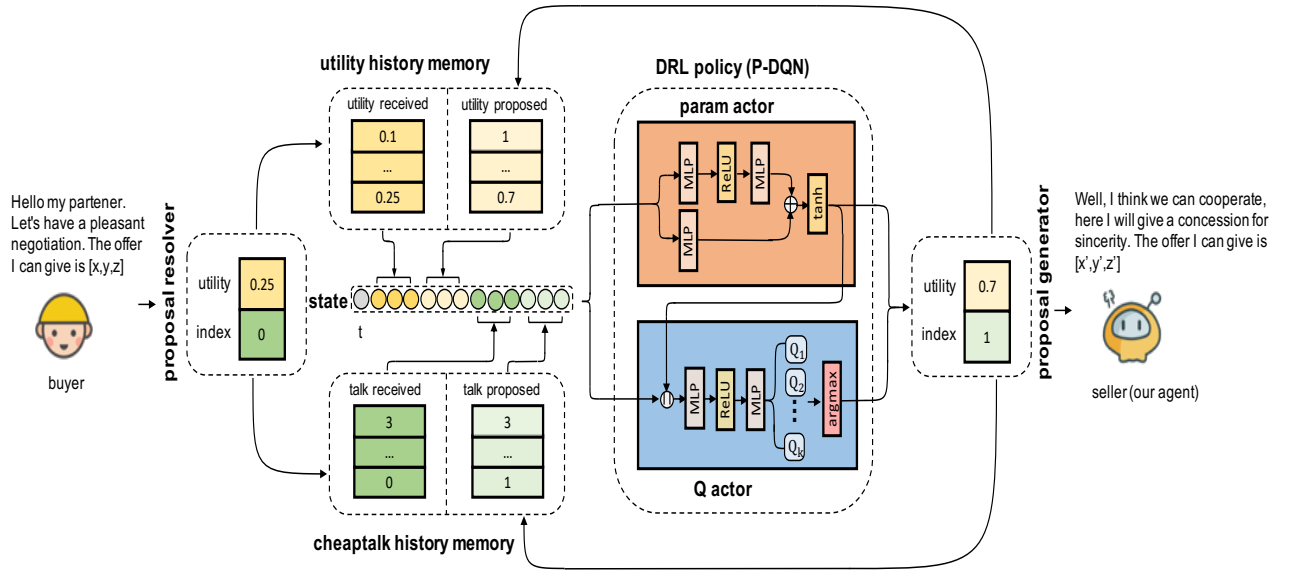


Figure 1. The overall architecture of the MCAN framework.

allows learning linguistic communication skills and bidding strategy simultaneously.

- We implements a negotiation environment where the human or agent negotiators can negotiate through both offer and linguistic channels.
- The extensive experiments show that even under the training of only simple rule-based agents, the MCAN agent can outperform human players, and the human perceptions of the MCAN agent are significantly higher than those of other agents.

2. Related work

Learning is important for automated negotiation and can significantly increase the performance of negotiation agents and their decision-making model. For the past decade, a variety of learning methods has been deployed in a range of negotiation scenarios. For example, Chen et al. [18] employed Gaussian process regression to approximate the opponent's model given no prior knowledge about their opponents' preferences and strategies. To tackle the problem of limited experience available in every single session, Chen et al. [19] developed a strategy that transfers knowledge efficiently from previous tasks on the basis of factored conditional restricted Boltzmann machines.

Reinforcement learning (RL) techniques, among others, have received increasingly more attention and been deployed in many negotiation tasks, like Q-learning [8]. More recently, deep learning has shown impressive performance in a number of areas [20–23]. RL has also been applied successfully in conjunction with deep learning, negotiation agents based on deep reinforcement learning (DRL) have gradually emerged to further improve agents' decision power. The authors of [9] have used an actor-critic algorithm for training both bidding and acceptance strategies in continuous state and action spaces. Sengupta et al. [11] use deep deterministic policy gradient (DDPG) [24] and soft actor-critic (SAC) [25] to learn a target utility for bidding strategy, which restricts the problem in continuous state and action spaces. These efforts, although successful, have considered negotiations in the form that information exchange between parties is only dependent on offers. As many real-world settings also

involve linguistic channels to express intentions, ask questions, discuss plans and so on, the information bandwidth is therefore restricted and grounded in the action space of negotiation.

There exists some research on linguistic communication in negotiation. Mell et al. [26] used effective conversation tactics to create value over repeated negotiations, and Oudah et al. [2] studied the impact of different talking styles on human opponents. These two methods used rule-based communication strategies which need to be pre-programmed by experts beforehand. Cao et al. [27] employed a deep neural network to negotiate on both proposal and linguistic channels, but the communication is implicit and cannot have support for human-machine interaction. Other studies focused on negotiation dialogues generation based on natural language processing [12, 13, 15]. He et al. [12] decoupled strategy and generation in negotiation dialogues using a two-layer architecture. Zhou [15] created a dynamic strategy coach to give tips and analyze during negotiation. Joshi et al. [13] incorporated graph neural networks as a strategy-graph network to predict the next optimal strategies. This kind of work only regards linguistic communication as a medium for conveying an offer and ignores the ability of linguistic communication to express important and useful information like emotions and intentions. The MACN agent can influence opponents and express its intentions by linguistic communication in terms of cheap talk, and learns linguistic communication skills and bidding strategy explicitly and it has generalization ability when facing different negotiators in different domains. To achieve that, the proposed agent must make decisions in a hybrid discrete-continuous action space, requiring a new algorithm for this problem. Therefore, this work employs parametrized deep Q-networks (P-DQNs) [28] to learn a target utility for a bidding strategy, and linguistic communication skills with the opponent. The P-DQNs is able to support hybrid discrete-continuous action space where every discrete action followed by a continuous parameter, by combining the advantages of DQN and DDPG.

3. Preliminaries

3.1. Negotiation settings

A negotiation setting contains a protocol, domain and agents. The negotiation protocol determines the rules of how agents interact with each other. The negotiation domain determines an outcome space Ω, where the agents negotiate for a win-win offer $\omega \in \Omega$. The negotiation domain can have single or multiple issues, where issues refer to the resources under contention, such as the price of an object or the level of the quality. The agents are the negotiators participating in the negotiation. The negotiation between two negotiators is called bilateral negotiation.

A large number of works [3,29,30] adopt the alternative offer protocal [17] for bilateral negotiation. During negotiation, the two negotiators in turn make offers and counter-offers until one accepts the other or the negotiation is terminated due to time-out. Another key aspect of a negotiation model is how an agent evaluates the utility of an offer. Formally, given an offer ω, which contains the value of each issue, let i denotes agent i, v_j denotes the value of issue j in the ω, and w_j^i denotes the weighting preference which agent i assigns to issue j (i.e., $\sum_{j=1}^{n} w_j^i = 1, 0 \leq w_j^i \leq 1$ where n denotes issue numbers). Therefore, the utility of the offer ω for agent i is defined as:

$$U^i(\omega) = \sum_{j=1}^{n} \left(w_j^i \cdot V_j^i \left(v_j \right) \right) \tag{3.1}$$

where V_j^i is the evaluation function of agent i, mapping the value of issue j to utility.

In order to get a good deal, agents are equipped with various bidding strategies [31]. At every round, the agent calculates its target expected utility, which determines whether it accepts an offer or not using these strategies. Among them, two strategies are commonly used: time-dependent strategies and behavior-dependent strategies [32]. Time-dependent strategies produce offers solely based on time. As time goes by, these strategies decrease the target expected utility gradually. Behavior-dependent strategies make offers depending on how opponent makes offers. The most well-known behavior-dependent strategy is tit-for-tat, which treats the opponent the same way it is treated.

3.2. Reinforcement learning

The reinforcement learning [33–35] problem can be formulated by a Markov decision process (MDP) defined as a 5-tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$, where $\mathcal{S}$ and $\mathcal{A}$ denote the sets of states and actions, respectively, $\mathcal{P}$ denotes the transition function: $\mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$, $\mathcal{R}$ denotes the reward function: $\mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$, and $\gamma \in [0, 1)$ denotes the discount factor. At the step t, the agent observes state $s_t \in \mathcal{S}$ and selects action a_t from the action space $\mathcal{A}$ following a policy π, which is a mapping from state space to action space. After taking the action, the agent receives reward r_t. The goal of the agent is to find a policy π that maximizes the expected discounted cumulative return,

$$G_t = \sum_{t=0}^{T} \gamma^t r_{t+1} \tag{3.2}$$

where T is the length of the whole episode.

Deep learning has been used widely recently [23, 36–38]. Combined with deep learning, deep Q-networks (DQNs) [39, 40] use the Bellman equation to approximates the optimal Q^* for discrete action space problems. For continuous action space problems, deep deterministic policy gradients (DDPGs) [24] consider the action-value function $Q_\pi(s_t, a_t)$ as a critic denotes how good it is when taking action a_t in s_t, where a_t is generated by deterministic policy $\mu_\theta : \mathcal{S} \to \mathcal{A}$. However, to deal with negotiation with offers and linguistic communication channels, the hybrid discrete-continuous action space is required to build the negotiation agent. In the following, the parametrized deep Q-networks (P-DQNs) is thus introduced.

3.3. Parametrized deep Q-networks

As for the hybrid discrete-continuous action space where every discrete action is followed by a continuous parameter, parametrized deep Q-networks (P-DQNs) combines the advantages of DQN and DDPG and gives a general solution to this problem. We focus on a discrete-continuous hybrid action space

$$\mathcal{A} = \{(k, x_k) \mid x_k \in \mathcal{X}_k \text{ for all } k \in K\} \tag{3.3}$$

where k is the k-th discrete action from discrete action set K, and x_k is the continuous parameter corresponding to discrete action k from k-th continuous action set $\mathcal{X}_k$. Therefore, the action value function is denoted as $Q(s, k, x_k)$, where $s \in \mathcal{S}, k \in K$, and $x_k \in \mathcal{X}_k$. Let k_t be the discrete action selected at time t and let x_t be the associated continuous parameter. Then, the Bellman equation becomes

$$Q(s, k, x_k) = \mathop{\mathbb{E}}_{r,s'}\left[r + \gamma \max_{k'} Q\left(s', k', x_{k'}^Q(s')\right) \mid s, k, x_k\right]. \tag{3.4}$$

P-DQNs first solve $x_k^* = \text{argsup}_{x_k \in \mathcal{X}_k} Q(s_{t+1}, k, x_k)$ using a deterministic policy network $x_k(s; \theta_x) : \mathcal{S} \to \mathcal{X}_k$, where θ_x denotes the network weights of the policy network, and then take the largest $Q(s_{t+1}, k, x_k^*)$ using a deep neural network $Q(s, k, x_k; \theta_Q)$, where θ_Q denotes the network weights. With this formulation, it is easy to apply the standard DQN approach of minimising the mean-square Bellman error to update the Q-network using mini-batches sampled from the replay buffer.

$$L_Q(\theta_Q) = \mathop{\mathbb{E}}_{(s,k,x_k,r,s') \sim D} \left[\frac{1}{2} \left(y - Q(s, k, x_k; \theta_Q) \right)^2 \right] \tag{3.5}$$

where $y = r + \gamma \max_{k' \in [K]} Q(s', k', x_{k'}(s'; \theta_x); \theta_Q)$ is the update target derived from Eq (3.4). Then the loss for the actor network in P-DQNs is given by the negative sum of Q-values:

$$L_x(\theta_x) = \mathop{\mathbb{E}}_{s \sim D} \left[- \sum_{k=1}^{K} Q(s, k, x_k(s; \theta_x); \theta_Q) \right]. \tag{3.6}$$

Through these two loss functions, we can calculate the policy gradient as

$$\nabla_{\theta_x} \mathbf{x}(s; \theta_x) = - \sum_{k=1}^{K} \nabla_{\mathbf{x}} Q(s, k, \mathbf{x}(s; \theta_x); \theta_Q) \nabla_{\theta_x} \mathbf{x}(s; \theta_x). \tag{3.7}$$

Using Eq (3.7), the policy network weight can be updated, and then the hybrid discrete-continuous action space problem can be solved. The P-DQN framework given in [28] can be found below.

Algorithm 1 Parametrized Deep Q-Network (P-DQN)

Require: Step sizes $\{\alpha_t, \beta_t\}_{t \geqslant 0}$, exploration parameter ϵ, minibatch size B, a probability distribution ξ.
Initialize network weights ω_1 and θ_1.
for t = 1,2,...,T **do**
Compute action parameters $x_k \leftarrow x_k(s_t, \theta_t)$.
Select action $a_t = (k_t, x_{k_t})$ according to the $\epsilon - greedy$ policy.

$$a_t = \begin{cases} \text{a sample from distribution } \xi & \text{with probability } \epsilon, \\ (k_t, x_{k_t}) \text{ such that } k_t = arg\, max_{k \in [K]} Q(s_t, k, x_k; \omega_t) & \text{with probability } 1 - \epsilon. \end{cases}$$

Take action a_t, and observe reward r_t and the next state s_{t+1}.
Store transition $[s_t, a_t, r_t, s_{t+1}]$ into $\mathcal{D}$.
Sample B transitions $\{s_b, a_b, r_b, s_{b+1}\}_{b \in [B]}$ randomly from $\mathcal{D}$.
Define the target y_b by

$$y_b = \begin{cases} r_b & \text{if } s_{b+1} \text{ is the terminal state}, \\ r_b + max_{k \in [K]} \gamma Q(s_{b+1}, k, x_k(s_{b+1}, \theta_t); \omega_t) & \text{if otherwise}. \end{cases}$$

Use data $\{y_b, s_b, a_b\}_{b \in [B]}$ to compute the stochastic gradients $\nabla_\omega \ell_t^Q(\omega)$ and $\nabla_\theta \ell_t^\Theta(\theta)$.
Update the weights by $\omega_{t+1} \leftarrow w_t - \alpha_t \nabla_\omega \ell_t^Q(\omega_t)$ and $\theta_{t+1} \leftarrow \theta_t - \beta_t \nabla_\theta \ell_t^\Theta(\theta_t)$.
end for

4. The design of the MCAN agent

4.1. Negotiation environment

We consider the bilateral multi-issue negotiation scenario where a buyer and a seller bargain for a piece of goods with communication. We follow the process of the alternative offer protocol except that we use both offer and cheaptalk while the origin protocol only uses offer. This protocol assumes an e-market environment where agents can communicate with opponents using cheaptalk while offering an offer. A buyer b always starts the negotiation by sending a proposal to seller s. During the negotiation both sides make proposals and counter-proposals in turn until one accepts the other, or the negotiation is terminated due to time-out. We are using the utility function defined in Eq (3.1). The cheaptalk refers to the views on negotiation that the agent wants to express to the opponent. In actual negotiations, linguistic communication can better convey the willingness to negotiate and speed up the negotiation process. Here we present a summary of the nine events that usually occur in nogotiation, as shown in Table 1.

Table 1. The negotiation events used in our framework.

Negotiation events	Category index
Accept the opponent's proposal	0
Promise to the opponent to make concessions	1
Request opponent to give a concessions	2
Show bottom line to opponent	3
Belief that both sides can get a win-win result	4
Greeting opponent for starting negotiation	5
Punish opponent for breaking its word	6
Forgive opponent for breaking its word	7
Threaten opponent for no conceding	8

Combining these events with different personas (nice, tough), we defines 18 different cheaptalk template for negotiators to choose. The nice persona seeks to avoid criticizing, complaining, or condemning its opponents, while respectfully building them up. On the contrary, the tough persona seeks to pull others down rather than build them up and advocates to be mean to others. The examples are given in Table 2. Besides, according to whether cheaptalk and offer are consistent, the negotiators can be honest or dishonest. For example, a dishonest negotiator will choose the event *promise to the opponent to make concessions* while giving an offer without concessions.

4.2. Proposed negotiation framework

We model this negotiation problem as an MDP, and the overall architecture is shown in Figure 1. To simplify and highlight the core of our work, cheaptalk is presented by its cheaptalk index in the predefined templates in Table 2. The MCAN agent receives the proposal from the opponent and uses the proposal resolver, which resolves the offer and cheaptalk index from the proposal. After calculating the utility of the offer, the information is added to the utility history memory and cheaptalk history

Table 2. The negotiation events used in our framework.

Category	Example cheaptalk for nice persona	Example cheaptalk for tough persona
0	Hi, I think this proposal is really nice. I accept it.	Oh, this barely satisfies me, but I accept it.
1	Well, I think we can cooperate, and here I will give a concession for sincerity.	...sigh. OK, for your pitiful sake, let me give you a little concession.
2	Hi, I know that everyone has difficulties, but it is a little unfair to me. Can you concede a bit?	Oh!!! You are such a mean person. Give me some concession.
3	My partner, this is the lowest bottom line I can accept. Let's understand each other.	This is my bottom line, deal or not deal!!!
4	Let's explore other options that may be better for us.	Well, you have to do what I said, and both sides can be good.
5	Hello my partener. Let's have a pleasant negotiation.	Hello.
6	I will punish you next turns, but I hope both sides can cooperate.	You selfish person, I will punish you.
7	What you did is totally understandable, though it will not benefit you in the long run.	You have been unimaginably selfish, but I will look past it for now.
8	I'm sorry that I can only start punishing you if you don't concede	You will regret it if you don't concede.

memory, respectively. The state is then generated by concatenating the current time, utility received history, utility proposed history, cheaptalk received history and cheaptalk proposed history. After that, the MCAN agent uses the DRL policy to generate an action which contains the best responding cheaptalk index and the corresponding target expected utility. After checking the acceptance condition, the counter-proposal is made by proposal generator.

4.2.1. Proposal resolver and proposal generator

The proposal resolver and the proposal generator are similar, since they're the reverse process of each other. The proposal resolver uses a sentence from the opponent as input and outputs the core of the proposal in the form of a dictionary while the proposal generator uses the proposal in dictionary form as input and outputs the responding sentence to the opponent. Using the pre-defined cheaptalk template, we can easily classify the sentence paradigm and extract the offer and cheaptalk index. The offer generator in the proposal generator is defined as

$$O^i\left(u_{target}\right) = \underset{\omega}{\operatorname{argmin}}\left(U^i(\omega) - u_{target}\right)^2 \tag{4.1}$$

where u_{target} denotes the target expected utility the agent wants to attain, and ω denotes any outcome in the outcome space Ω.

4.2.2. Utility history and cheaptalk history

Utility history and cheaptalk history are used to record the proposals proposed and received in the negotiation history. After resolving the proposal from the opponent, the utility of the offer will be calculated using Eq (3.1). It is then added to the utility received history. Meanwhile, the cheaptalk index will be added to the talk received history. After the MCAN agent makes a proposal, the utility proposed history and cheaptalk proposed history will be updated.

4.2.3. DRL policy

DRL policy is aiming to give the best cheaptalk index and the corresponding target expected utility for an offer at a certain state. We adopt the P-DQN algorithm here, as P-DQN gives a general solution to the hybrid discrete-continuous action space problem. It adapts well to our negotiation settings because it is able to calculate the target utility and cheaptalk index at the same time, which solves the problem of mutual influence between cheaptalk index action and target expected utility action.

The state is concatenated by the current time, utility received history, utility proposed history, cheaptalk received history and cheaptalk proposed history. For a negotiation session with time limit T, we use $u_{propose}[t-i,t]$ to denote the utility proposed history from $t-i$ to t, where $i \in [0, t-1]$. Similarly, $u_{receive}[t-i,t]$ denotes the utility received vector. $c_{propose}[t-i,t]$ denotes the cheaptalk index proposed from $t-i$ to t, and $c_{receive}[t-i,t]$ denotes the cheaptalk index received from $t-i$ to t. After defining these, the state and action can be defined as

$$\begin{aligned} s_t = \; & \{t_r, u_{propose}[t-2,t], u_{receive}[t-2,t], \\ & c_{propose}[t-2,t], c_{receive}[t-2,t]\} \\ a_t = \; & \left\{c, u_{target}\right\} \end{aligned}$$

where t_r denotes the relative time ($t_r = t/T$). The action outputs the cheaptalk index and the target expected utility. By checking the acceptance condition as (Eq (4.2)), the agent decides to use the proposal generator or accept the opponent's proposal.

$$\text{choice} = \begin{cases} \text{accept} & u_{target} <= u^t_{received} \\ \text{counter-proposal} & u_{target} > u^t_{received} \end{cases} \tag{4.2}$$

The goal of the DRL agent is to maximize the utility of the agreement offer, so the reward function is defined as

$$r = \begin{cases} U(\omega_a) & \text{if there is an agreement } \omega_a \\ 0 & \text{for no agreement} \end{cases}$$

where $U(\omega)$ means the self utility of the offer ω.

The main algorithm is shown in Algorithm 2. Before training, an opponent pool is prepared for the MCAN agent to negotiate with. In every episode, the algorithm randomly chooses an opponent from the opponent pool Π. Then, the MCAN agent and the chosen opponent are asked to start a

Algorithm 2 The training process of MCAN agent

Require: Episodes K, exploration parameter ϵ, replay buffer $\mathcal{D}$, opponent pool Π, max round T

for $k = 1, 2, .., K$ **do**
  Uniformly sample an opponent from opponent pool Π.
  for $t = 1, 2, ..., T$ **do**
    Update state s_t based on the history information.
    Select action $a_t = (c, u_{target})$ according to the ϵ-greedy policy.
    Check the acceptance condition according to (Eq (4.2)) and generate a choice.
    Received proposal from opponent and observe reward r_t, next state s_{t+1}.
    Store transition $[s_t, a_t, r_t, s_{t+1}]$ into $\mathcal{D}$.
    Update the history information based on action a_t and opponent's proposal.
    MCAN agent updates the Q actor network and param actor network weights according to 3.7.
  end for
end for

negotiation. During the negotiation, the MCAN agent collects the history information and updates the state s. Using the ϵ-greedy policy, the MCAN agent generates the current target expected utility and corresponding cheaptalk index. After checking the acceptance condition, the proposal is sent to the opponent. Through the alternative proposal, the MCAN collects the experience in the replay buffer and uses them to update the Q actor network and parameter actor network weights. After several rounds of training, the policy converged.

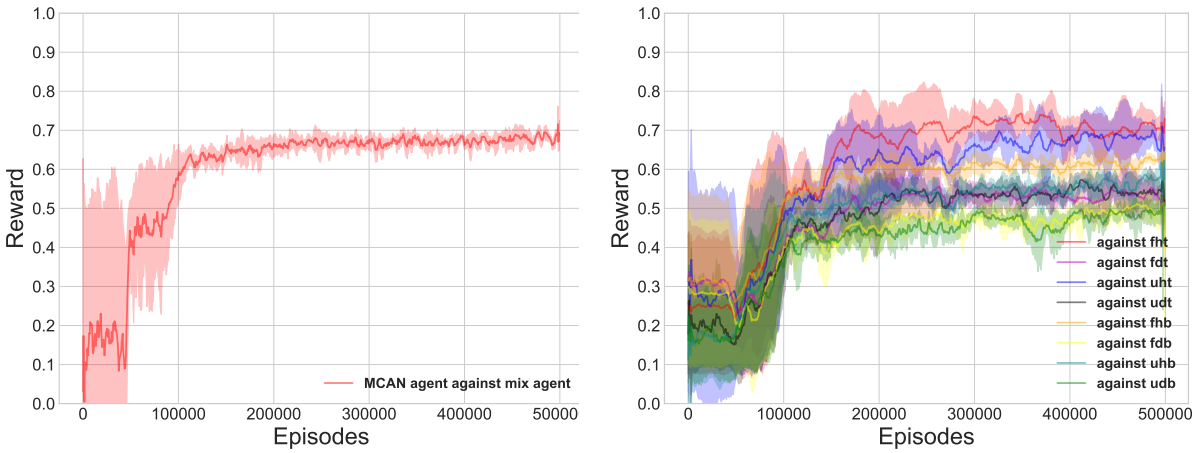
5. Experiments

In this section, we present the experimental settings and results of the MCAN agent against opponents. We begin with the experimental technicalities, which are given in Section 5.1. Section 5.2 compares the performance of our agent against the rule based negotiation agents. Then, a user study is conducted to obtain human evaluation in Section 5.3. A GUI human-agent negotiation environment is developed to facilitate interaction between agents with human players. Section 5.4 reports a comparative experiment to show how the P-DQN algorithm promotes the performance of the MCAN agent.

5.1. Experimental setup

We first design several rule-based negotiation agents and use them to train our MCAN agent. The rule-based negotiation agents are designed by integrating the linguistic communication skills of humans and classic bidding strategies. Here, we named the nice persona as friendly and the tough persona as unfriendly. Besides, according to whether the negotiator is honest, we divide linguistic communication skills into four types. The classic bidding strategies we used here are time-dependent Boulware agent and behavior-dependent tit-for-tat agent. We combine the linguistic communication skills and bidding strategy by using the target expected utility. Each round, the bidding strategy outputs a target expected utility based on utility history information and time, and the target expected utility is modified based on different linguistic communication skills. Besides, friendly agents will

choose the left column of Table 2, and unfriendly agents choose the right side. The honesty will decide whether the offer and cheaptalk have consistent performance. We named these 8 rule-based negotiation agents as friendly honest time-dependent Boulware (fht), friendly dishonest time-dependent Boulware (fdt), unfriendly honest time-dependent Boulware (uht), unfriendly dishonest time-dependent Boulware (udt), friendly honest behavior-dependent (fhb), friendly dishonest behavior (fdb), unfriendly honest behavior-dependent (uhb) and unfriendly dishonest behavior-dependent (udb), respectively. For



(a) The training process of MCAN agent **(b)** The evaluation process of the MCAN agent

Figure 2. In the training process, the MCAN agent is trained against the 8 rule-based agents by randomly choosing one opponent per episode. In the evaluation process, the MCAN agent is evaluated againsts the 8 rule-based agents, respectively.

the MCAN agent, we use the implement of the P-DQN from [28]. The only difference in the network architecture is that we added a layer activation function of tanh at the end of the actor parameter network to limit the output to between –1 and 1. The hyperparameters are set as follows: batch size = 128, discount = 0.99, Q-networks update parameter = 0.1, actor parameter networks update parameter = 0.001, replay buffer size = 100000. The optimizers for Q-networks and the actor parameter network are both Adam, with learning rates 0.001 and 0.0001, respectively.

All the experiments were conducted using our self-developed Python environment. Among the negotiation settings, the reservation price is set as 0.1,and the maximum number of round in one episode is 28–32 rounds to avoid possible exploitation. We used min-max normalisation for making the utility values between 0 and 1.

5.2. Performance against rule-based negotiation agents

We trained the MCAN agent for 1,000,000 episodes, and the learning curve and evaluation curve of the first 500,000 episodes are shown in Figure 2. In the training phase, in each episode the MCAN agent will face an opponent randomly sampled from 8 rule-based negotiation agents. In the evaluation phase, the MCAN agent will have negotiations with all 8 opponents one by one. We can find that the MCAN agent converged at around 200,000 episodes with a reward of 0.68. Through the evaluation result graph, we can find that there is an obvious advantage in the negotiation with fht and fhb. These

two agents are more likely to make concessions first, which makes the MCAN agent achieve a good deal easily. For negotiation with ufht and ufhb, the MCAN agent can gain a slight advantage by reasonable using of cheaptalk signals since they are honest. For the remaining four agents, the MCAN agent will try to reach a tie as much as possible. With the addition of the MCAN agent, there are a total of 9 agents. We let these 9 agents negotiate in pairs, each pair carries out 1000 evaluations. Then, we calculated the averaged agreement utility over 1000 evaluations as the evaluation metric for one pair. Fht and fub are equipped with the friendly and honest talk strategy. When facing other agents, they are likely to concede first. This leads them to reach negotiations with all agents, but the agreement utility of the negotiation is mostly below 0.5. The other six agents are either dishonest or tough, which leads them to be unable to reach a deal. We averaged the results of one agent against 8 other agents, and the result is shown in Figure 3.

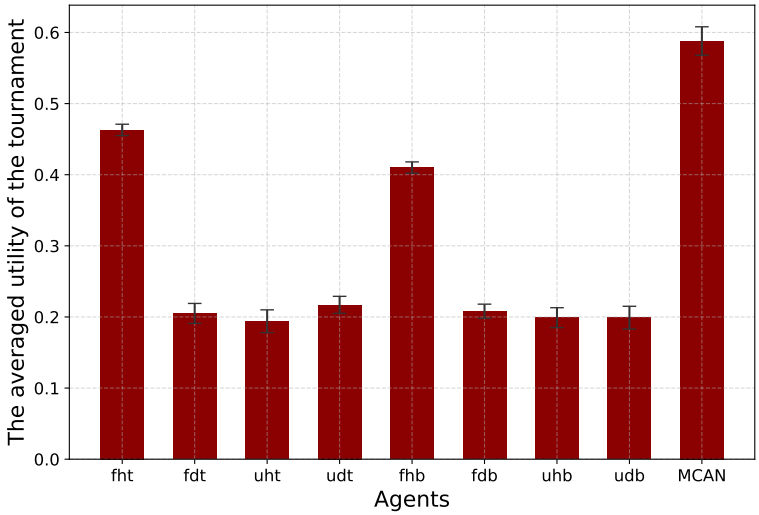
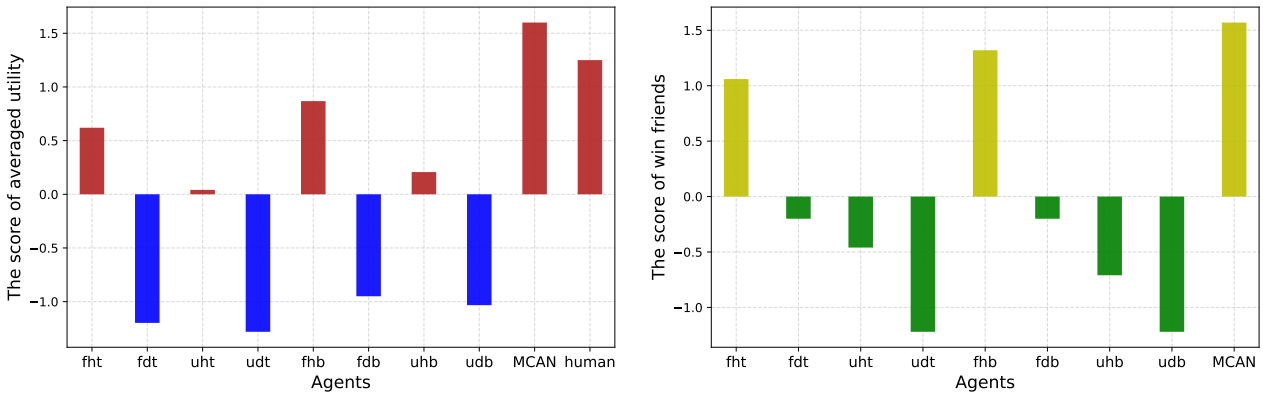


Figure 3. The averaged utility attained of 9 agents in the tournament.

It can be seen that the MCAN agent reached the highest average utility of 0.59, followed by fht and fhb, with 0.46 and 0.41, respectively. The remaining four agents are relatively close, all around 0.2. Through the above experiments, we can find that the MCAN agent can obtain a relatively good advantage when negotiating with the rule-based negotiation agents.

5.3. User study

To evaluate the negotiating performance against human participants, a GUI human-agent negotiation with talk environment was developed to enable agents to negotiate with humans. In this environment, we invited 200 volunteers from the College of Intelligence and Computing, Tianjin University, to conduct a human-agent negotiation test. Human volunteers are invited to be a party to the negotiation (buyer or seller), and their tasks are to get the highest possible agreement utility of negotiation. For the convenience of the volunteers, we used single-issue negotiation instead. Each volunteer participated 30 tests. For every 10 tests, they faced one agent randomly sampled from those 9 agents to fill out a questionnaire. We used the two evaluation metrics introduced by Oudah [2]. The first one is the



(a) The standard z-score of utility attained by each player **(b)** The standard z-score of human evaluation attained by each agent

Figure 4. The results in the user study. (a) denotes the quality of bidding strategy, and (b) denotes the ability of agents' human-agent friendless.

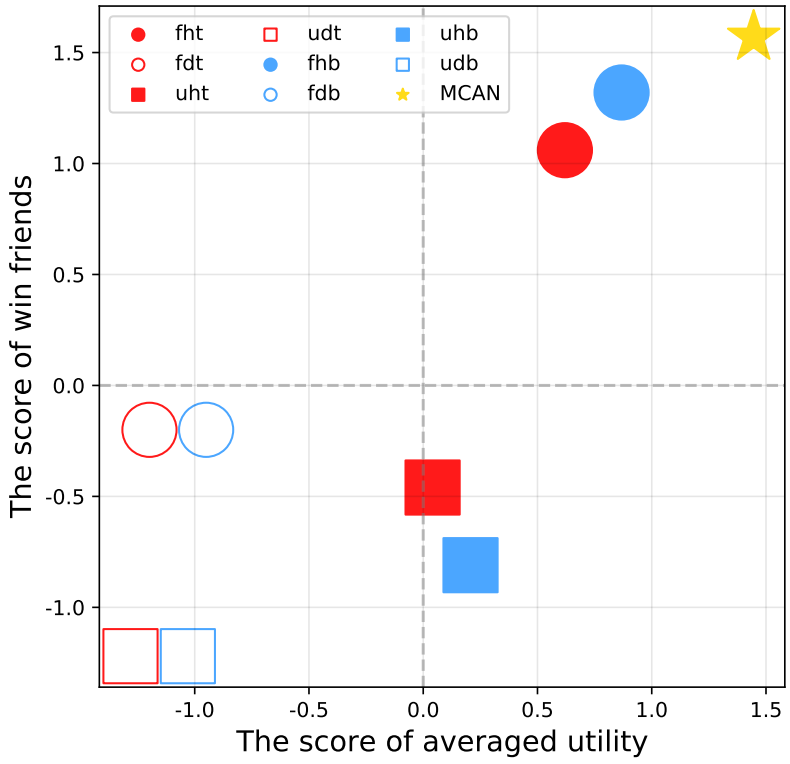


Figure 5. A summary of the user study.

average utility made between a volunteer and an agent, and the second one is the degree to winning friends, which is quantified through the questionnaire. The volunteers are asked (using a 5-point Likert scale) the degree to which they understand their opponent's intentions to what degree it is likable, intelligent, cooperative, friendy and willing to negotiate again. To compare the relative performances of the agents, we used the standardized z-score for each metric.

The results are shown in Figure 4. From Figure 4(a) we can find that the averaged utility obtained by the MCAN is the highest among the 9 agents, and it even slightly surpassed the performance of humans. We have found that maintaining honesty is an important reason for obtaining higher averaged utility since the fht, ufht, fhb and ufhb will maintain good credit in the negotiation. Besides, the performances of fht and fhb far exceeds other agents, which shows that keeping a nice persona will better than a tough persona. The results of Figure 4(b) are quite similar to Figure 4(a). The MCAN obtained the highest score among all agents, which shows that it is more capable of winning friends compared to the other rule-based agents. In the Figure 4(b), an interesting result is that the score of ufht and ufhb is not as positive as in Figure 4(a), which implies that the friendliness and honesty are both important. The Figure 5 summarizes the results of the study by showing the relative performance of the 9 agents in terms of averaged utility and score of winning friends.

According to Figure 5, it can be found that the MCAN agent learned the linguistic communication skills and bidding strategy successfully. To further analyze the behaviors learned by the MCAN agent, we visualized the cheaptalk MCAN has used in the negotiation with human volunteers, which is shown in Figure 6. The horizontal axis denotes the cheaptalk classes. Among them, the first 9 are the cheaptalk under the nice persona in the order of Table 2, while the last 9 are based on the tough persona. The results shows that the MCAN agent learns to behavior based on the the nice persona and is more likely to use indices 1, 2 and 3. This phenomenon shows that MCAN agent knows to show sincerity by expressing concessions to opponents in order to build trust between both parties and express its needs politely. These three cheaptalk actions are also worthy of attention, and they can be used to provide references for future linguistic design.

5.4. Analysis of P-DQN

In this section we will study the effect of the P-DQN by conducting a comparative experiment. The P-DQN takes the utility history and cheaptalk history as input and outputs the best cheaptalk index and the corresponding target expected utility. To evaluate the P-DQN's output, we made two variants of the P-DQN by replacing either the cheaptalk index side or the target expected utility side with random output, called the TP-DQN and OP-DQN, respectively. Then we carried out two other tournaments like we did in experiment 1 by replacing the P-DQN with our two variants. Then, we calculated the self utility attained against the 8 rule-based agents and the opponent average utility attained. The results are shown in Figure 7, and it can be found that the performances of the TP-DQN and OP-DQN plummet, which implies that the P-DQN's output is indivisible. The combined effect of cheaptalk index and target expected utility gives the P-DQN a great advantage.

6. Conclusion and future work

This work proposes a novel framework MCAN that leverages P-DQNs to learn a comprehensive negotiation strategy which integrates linguistic communication skills and bidding strategies. The P-

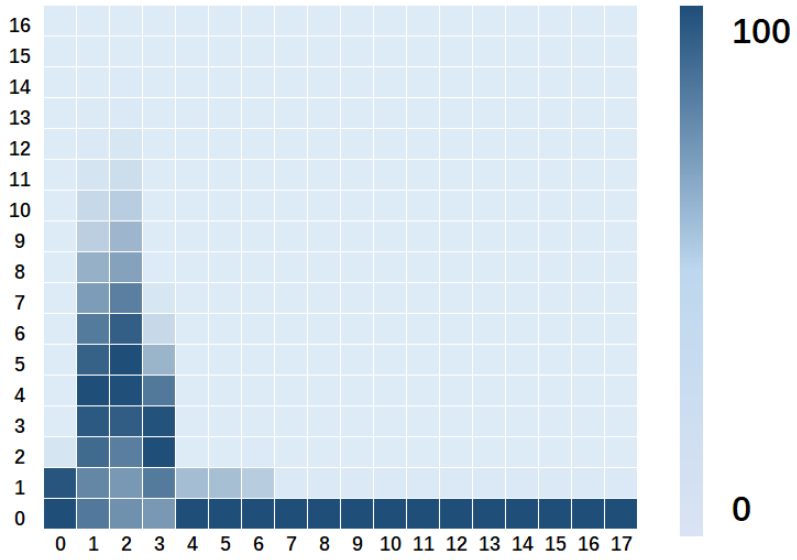


Figure 6. The visualization of the policy learned by the MCAN agent. The results shows that the MCAN agent learns to use the nice persona and is more likely to use indices 1, 2 and 3. From this phenomenon, we can infer that the MCAN agent knows to show sincerity by expressing concessions to opponents in order to build trust between both parties and express its needs politely.

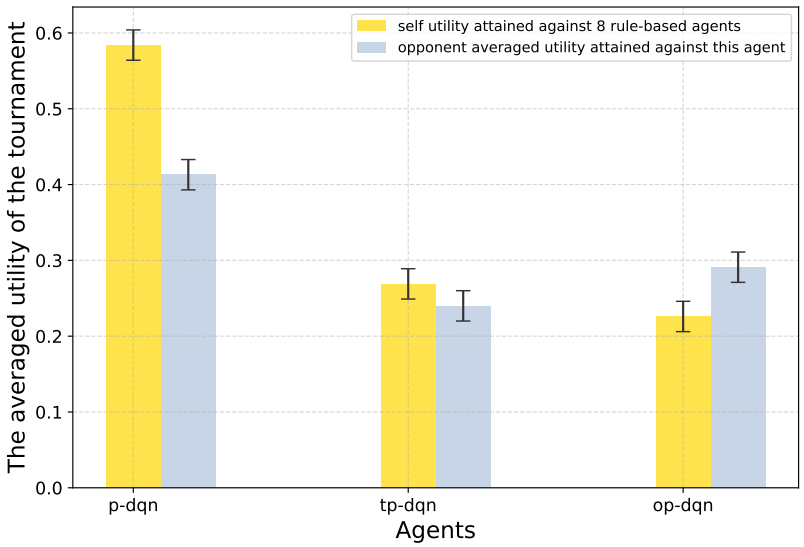


Figure 7. The comparative experimental results with TP-DQN and OP-DQN.

DQNs algorithm, which learns the optimal bidding strategy on each kind of linguistic communication skills and then chooses the co-optimal one, is suitable for this dual optimization problem. The experimental result shows that the MCAN agent outperforms the rule-based agent according to the averaged utility of negotiation. According to the user study, we also evaluated the MCAN agent and rule-based agents based on the metrics of winning friends in addition to utility metrics. The user study on negotiation with humans depicts that the MCAN agent has learned a comprehensive negotiation strategy which integrates linguistic communication skills and bidding strategies.

The exceptional results justify investment in further research efforts into this approach. In the future, we plan to explore how to learn more sophisticated linguistic communication skills to achieve a more application-oriented architecture. Second, the extension of the framework to other negotiation settings, such as concurrent negotiation or multi-lateral negotiation, is another interesting avenue to exploit. It is also of great interest to investigate how to train negotiation skills of human participants based on the framework. Last but not least, to further aid the learning performance in a different negotiation domain, incorporating transfer learning techniques into the proposed approach is believed to lead to improvement in negotiation power.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Grant No. 61602391).

Conflict of interest

The authors declare there is no conflict of interest.

References

1. X. Gao, S. Chen, Y. Zheng, J. Hao, A deep reinforcement learning-based agent for negotiation with multiple communication channels, in *2021 IEEE 33nd International Conference on Tools with Artificial Intelligence (ICTAI)*, IEEE, (2021), 868–872. https://doi.org/10.1109/ICTAI52525.2021.00139
2. M. Oudah, T. Rahwan, T. Crandall, J. Crandall, How AI wins friends and influences people in repeated games with cheap talk, in *Proceedings of the AAAI Conference on Artificial Intelligence*, (2018).
3. N. R. Jennings, P. Faratin, A. R. Lomuscio, S. Parsons, C. Sierra, M. Wooldridge, Automated negotiation: Prospects, methods and challenges, *Int. J. Group Decis. Negot.*, **10** (2001), 199–215. https://doi.org/10.1023/A:1008746126376
4. S. Chen, Y. Cui, C. Shang, J. Hao, G. Weiss, ONECG: Online negotiation environment for coalitional games, in *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS '19, Montreal, QC, Canada, May 13-17, 2019*, (2019), 2348–2350.
5. S. Chen and G. Weiss, An approach to complex agent-based negotiations via effectively modeling unknown opponents. *Expert Syst. Appl.*, 42 (2015), 2287–2304. https://doi.org/10.1016/j.eswa.2014.10.048

6. R. M. Coehoorn, N. R. Jennings, Learning on opponent's preferences to make effective multi-issue negotiation trade-offs, in *Proceedings of the 6th International Conference on Electronic Commerce*, (2004), 59–68. https://doi.org/10.1145/1052220.1052229
7. R. Lin, S. Kraus, J. Wilkenfeld, J. Barry, Negotiating with bounded rational agents in environments with incomplete information using an automated agent, *Artif. Intell.*, **172** (2008), 823–851. https://doi.org/10.1016/j.artint.2007.09.007
8. J. Bakker, A. Hammond, D. Bloembergen, T. Baarslag, Rlboa: A modular reinforcement learning framework for autonomous negotiating agents, in *Proceedings of the 18th international conference on Autonomous Agents and Multiagent Systems*, (2019), 260–268.
9. H. C. H. Chang, Multi-issue bargaining with deep reinforcement learning, preprint, arXiv: 2002.07788.
10. C. Jonker, R. Aydogan, T. Baarslag, K. Fujita, T. Ito, K. Hindriks, Automated negotiating agents competition (anac), in *Proceedings of the AAAI Conference on Artificial Intelligence*, (2017).
11. A. Sengupta, Y. Mohammad, S. Nakadai, An autonomous negotiating agent framework with reinforcement learning based strategies and adaptive strategy switching mechanism, preprint, arXiv: 2102.03588.
12. H. He, D. Chen, A. Balakrishnan, P. Liang, Decoupling strategy and generation in negotiation dialogues, in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, (2018), 2333–2343.
13. R. Joshi, V. Balachandran, S. Vashishth, A. Black, Y. Tsvetkov, Dialograph: Incorporating interpretable strategy-graph networks into negotiation dialogues, preprint, arXiv: 2106.00920.
14. S. Chen, Y. Yang, R. Su, Deep reinforcement learning with emergent communication for coalitional negotiation games, *Math. Biosci. Eng.*, **19** (2022), 4592–4609. https://doi.org/10.3934/mbe.2022212
15. Y. Zhou, H. He, A. W. Black, Y. Tsvetkov, A dynamic strategy coach for effective negotiation, in *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, (2019), 367–378. https://doi.org/10.18653/v1/W19-5943
16. R. Aydoğan, D. Festen, K. V. Hindriks, C. M. Jonker, Alternating offers protocols for multilateral negotiation, in *Modern Approaches to Agent-Based Complex Automated Negotiation*, Springer, (2017), 153–167. https://doi.org/10.1007/978-3-319-51563-2_10
17. A. Rubinstein, Perfect equilibrium in a bargaining model, *Econometric Soc.*, **50** (1982), 97–109. https://doi.org/10.2307/1912531
18. S. Chen, G. Weiss, An intelligent agent for bilateral negotiation with unknown opponents in continuous-time domains, *ACM Trans. Auton. Adapt. Sys.*, **9** (2014), 1–24. https://doi.org/10.1145/2629577
19. S. Chen, H. B. Ammar, K. Tuyls, G. Weiss, Using conditional restricted Boltzmann machine for highly competitive negotiation tasks, in *Proceedings of the 23th International Joint Conference on Artificial Intelligence*, (2013), 69–75.

20. Q. Jin, H. Cui, C. Sun, Z. Meng, R. Su, Free-form tumor synthesis in computed tomography images via richer generative adversarial network, *Knowl.-Based Syst.*, **218** (2021), 106753. https://doi.org/10.1016/j.knosys.2021.106753
21. J. Liu, R. Su, J. Zhang, L. Wei, Classification and gene selection of triple-negative breast cancer subtype embedding gene connectivity matrix in deep neural network, *Brief. Bioinf.*, **22**, (2021). https://doi.org/10.1093/bib/bbaa395
22. Q. Jin, Z. Meng, T. D. Pham, Q. Chen, L. Wei, R. Su, DUNet: A deformable network for retinal vessel segmentation, *Knowl.-Based Syst.*, **178**, (2019), 149–162. https://doi.org/10.1016/j.knosys.2019.04.025
23. R. Su, X. Liu, L. Wei, Q. Zou, Deep-Resp-Forest: A deep forest model to predict anti-cancer drug response, *Methods*, **166** (2019), 91–102. https://doi.org/10.1016/j.ymeth.2019.02.009
24. T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, et al., Continuous control with deep reinforcement learning, preprint, arXiv: 1509.02971.
25. T. Haarnoja, A. Zhou, P. Abbeel, S. Levine, Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor, in *International Conference on Machine Learning*, (2018), 1861–1870.
26. J. Mell, G. M. Lucas, J. Gratch, An effective conversation tactic for creating value over repeated negotiations., in *AAMAS*, **15**, (2015), 1567–1576.
27. K. Cao, A. Lazaridou, M. Lanctot, J. Z. Leibo, K. Tuyls, S. Clark, Emergent communication through negotiation, in *6th International Conference on Learning Representations*, (2018).
28. J. Xiong, Q. Wang, Z. Yang, P. Sun, L. Han, Y. Zheng, et al., Parametrized deep q-networks learning: Reinforcement learning with discrete-continuous hybrid action space, preprint, arXiv: 1810.06394.
29. T. Baarslag, K. Fujita, E. H. Gerding, K. Hindriks, T. Ito, N. R. Jennings, et al., Evaluating practical negotiating agents: Results and analysis of the 2011 international competition, *Artif. Intell.*, **198** (2013), 73–103. https://doi.org/10.1016/j.artint.2012.09.004
30. S. Chen, H. B. Ammar, K. Tuyls, G. Weiss, Optimizing complex automated negotiation using sparse pseudo-input gaussian processes, in *Proceedings of the 2013 International Conference on Autonomous Agents and Multi-agent Systems*, (2013), 707–714.
31. L. Ilany, Y. Gal, Algorithm selection in bilateral negotiation, *Auton. Agents Multi-Agent Syst.*, **30** (2016), 697–723. https://doi.org/10.1007/s10458-015-9302-8
32. P. Faratin, C. Sierra, N. R. Jennings, Negotiation decision functions for autonomous agents, *Robot. Auton. Syst.*, **24** (1998), 159–182. https://doi.org/10.1016/S0921-8890(98)00029-3
33. R. S. Sutton, A. G. Barto, *Reinforcement Learning: An Introduction*, MIT press, 2018.
34. M. A. Wiering, M. Van Otterlo, Reinforcement learning, in *Adaptation, Learning, and Optimization*, (2012).
35. C. Szepesvári, Algorithms for reinforcement learning, in *Synthesis Lectures on Artificial Intelligence and Machine Learning*, **4** (2010), 1–103. https://doi.org/10.2200/S00268ED1V01Y201005AIM009

36. B. Song, F. Li, Y. Liu, X. Zeng, Deep learning methods for biomedical named entity recognition: a survey and qualitative comparison, *Brief. Bioinf.*, **22** (2021). https://doi.org/10.1093/bib/bbab282

37. A. Lin, W. Kong, S. Wang, Identifying genetic related neuroimaging biomarkers of Alzheimer's disease via diagnosis-guided group sparse multitask learning method, *Curr. Bioinf.*, **16** (2021), 1–1. https://doi.org/10.2174/1574893616012103011058 59

38. J. Dong, M. Zhao, Y. Liu, Y. Su, X. Zeng, Deep learning in retrosynthesis planning: Datasets, models and tools, *Brief. Bioinf.*, **23** (2022), Bbab391.

39. V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstraey at al., Playing Atari with deep reinforcement learning, preprint, arXiv: 1312.5602.

40. V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, et al., Human-level control through deep reinforcement learning, *Nature*, **518** (2015), 529–533. https://doi.org/10.1038/nature14236

AIMS Press