An Effective Negotiating Agent Framework based on Deep Offline Reinforcement Learning

Abstract

Learning is crucial for automated negotiation, and recent years have witnessed a remarkable achievement in application of reinforcement learning (RL) for various negotiation tasks. Conventional RL methods focus generally on learning from active interactions with opposing negotiators. However, collecting online data is expensive in many realistic negotiation scenarios. While previous studies partially mitigate this problem through the use of opponent simulators (i.e., agents following known strategies), in reality it is usually hard to fully capture an opponent's negotiation strategy. Moreover, a further challenge lies in an agent's capability of adapting to dynamic variations of an opponent's preferences or strategy, which may happen from time to time for different reasons in subsequent negotiations. In response to these challenges, this article proposes a novel Deep Offline Reinforcement learning Negotiating Agent (DOREA) framework that allows to learn an effective strategy using previously collected negotiation datasets without requiring interaction with an opponent. This is in contrast to existing RL-based negotiation approaches that all rely on active interaction with opponents. Furthermore, the strategy fine-tuning mechanism is included to adjust the learned strategy in response to the preferences or strategy changes of the opponent. The performance of the DOREA framework is evaluated based on a diverse set of state-of-the-art baselines under different settings. Experimental results show that the proposed framework allows to learn effective strategies exclusively with offline datasets, and is also capable of effectively adapting to changes of an opponent's negotiation preferences or strategy.

1 INTRODUCTION

Negotiation is a process where parties of different interests exchange offers to mutually explore the likelihoods of achieving mutual benefit, resolving conflicts or finding mutually acceptable solutions. With that, negotiation can serve as a fundamental and powerful mechanism for managing conflicts [Jennings et al., 2001]. This mechanism, however, can be time-consuming and costly for humans [Fatima et al., 2004]. Automated negotiation has therefore become a subject of central interest in multi-agent systems over the past decade due to its advantages over non-computerized negotiation, such as alleviating the efforts of human negotiators, reaching better outcomes by compensating limitations of human computational and reasoning abilities, and so on.

Reinforcement learning (RL) is a powerful learning paradigm for control tasks. Specifically, RL can be utilized to automatically acquire near-optimal behavioral skills (represented by policies) for given tasks. The successful application of RL algorithms in diverse fields (e.g., natural language processing, computer vision and complex games [Devlin et al., 2019, Silver et al., 2017]) has also led to the exploration of RL in automated negotiation [Bakker et al., 2019, Bagga et al., 2020, Chang, 2021, Sengupta et al., 2021, Wu et al., 2021, Yang et al., 2021, Higa et al., 2023, Chen et al., 2023]. Despite the remarkable progress that has been achieved so far, conventional RL methods for negotiation typically focus on online learning from active interactions with the environment (i.e., everything in the negotiation scenario including the opponent and the domain) to iteratively collect data to be used for policy improvement. However, this kind of online learning is of limited value and often impractical for negotiation, mainly because data collection based on online interactions is very expensive. For example, training a RL agent from scratch in an e-commence scenario against a negotiation partner is likely to lead to a large number of unacceptable results and low-quality customer experience. While previous approaches partially mitigate this problem by using opponent simulators (i.e., agents ap-

Submitted to the 39th Conference on Uncertainty in Artificial Intelligence (UAI 2023). **To be used for reviewing only**.

plying known strategies) for training, in realistic settings it is usually hard to fully capture an opponent's negotiation strategy due to uncertain user states and actions, noisy environments, and the fact that negotiators aim at hiding information related to their strategies in order to hamper exploitation through their opponents.

Due to the limited value of online RL for automated negotiation, a key question is whether data collected during previous negotiation sessions can be effectively utilized by an agent to learn its negotiation skills. In particular, can an agent do so, and moreover adapt its negotiation strategy when there are changes in opponent preference or strategy due to various reasons (e.g., different social motives of users, market demand). Therefore, a novel Deep Offline Reinforcement learning Negotiating Agent (DOREA) framework is proposed, which can learn an effective strategy from offline datasets of previously collected negotiation experiences. The DOREA framework does also enable an agent to fine-tune a learned strategy and to adapt it to changes of the opponent preferences or strategy.

The remainder of this paper is structured as follows. Section 2 overviews important related work. Section 3 provides the reader with background knowledge that is relevant for the remaining sections. The technicalities of the DOREA framework are presented in Section 4. An in-depth analysis is given in Section 5. Lastly, Section 6 concludes and identifies interesting future research directions.

2 RELATED WORK

Recently, RL-based negotiating agents have attracted considerable research attention. For example, Bakker et al. [2019] propose a RL framework (RLBOA) built on the BOA architecture for automated negotiation. The Tabular Q-learning algorithm is used to train the bidding strategy. To have a compact state representation, RLBOA maps the offers to the utility space and discretizes the utility space into a number of equal bins. A problem with such discretization is that it can lead to loss of information conveyed in the offers, e.g., the state/action domain structure. Moreover, the Q-learning approach suffers from large state space and over-estimation of Q value problem. Bagga et al. [2020] pre-train a negotiation strategy through supervised learning (SL) with synthetic data in order to accelerate the learning process. Initialized by the learned SL strategy, the negotiation agent evolved using a model-free Deep RL method called Deep Deterministic Policy Gradient (DDPG) [Lillicrap et al., 2016] with additional negotiation experience. A limitation of this approach is that it only addresses negotiations of a single issue. Wu et al. [2021] considered the negotiation scenarios where the opponent may change its strategy at times. They proposed a negotiating agent based on Bayesian policy reuse to detect an opponent strategy and respond with the best learned RL policy from existing policies. Higa et al. [2023] proposed a reward-based negotiating agent strategy through a multi-issue policy network. The policy network was trained to predict the optimal policy in policy-based RL without incorporating utility functions. Although the existing work has advanced the field of automated negotiation, it still suffers from one common limitation, that is, the requirement for a large number of online interactions with the environment in order to train the policy. The work most closely related to ours is Sengupta et al. [2022]. There a negotiation framework is proposed that trains a base model with negotiation history for its bidding strategy. A binary classifier enables the detection of changes in utility functions, and then the adapted model is provided to automatically adapt to such changes by using parameter sharing based transfer learning technique with newly collected datasets during negotiation. A drawback of this framework is that the opponent must keep its strategy fixed all the time, otherwise both the classifier and adapted model will be ineffective (as they are trained by the negotiation traces produced by the opponent strategy). In contrast, our approach is considerably broader in its applicable range of negotiation tasks because it is not restricted to any opponent strategy, utility function or the quality of previously collected datasets (as we will show later in Section 5.2, a dataset collected by even a simple strategy can produce a negotiating agent based on DOREA framework whose performance is still acceptable.).

Offline RL is a new RL paradigm concerned with learning exclusively from datasets of previously-collected experiences [Levine et al., 2020]. This learning pattern is very valuable in environments where online interaction is impractical or expensive, and has achieved remarkable successes in robotics [Chen et al., 2022, Yu et al., 2021], autonomous driving [Tennenholtz and Mannor, 2022], healthcare [Fatemi et al., 2021], and other fields [Prudencio et al., 2022]. Although recently much research effort has been devoted to learning useful negotiation strategies with RL, to the best of our knowledge, our work is the first attempt to use (1) offline RL for learning an effective negotiation strategy and (2) offline-to-online techniques for fine-tuning the learned strategy and adapting it to changes of opponent preferences or strategies.

3 PRELIMINARIES

3.1 NEGOTIATION SETTINGS

This work adopts a bilateral multi-issue negotiation environment widely used in the automated negotiation field (e.g., [Hao et al., 2014, Sengupta et al., 2021, Wu et al., 2021]). A negotiation scenario consists of a domain description and preference profiles of both parties. The preference profiles of a domain determine the utility functions (as shown below). Let I be the set of negotiation agents, with i representing a specific agent ($i \in \{o, s\}$ where s refers to the agent and o to its opponent). J is the set of issues under negotiation,

with j being a particular issue ($j \in \{1, ..., n\}$ where n is the number of issues). The utility function of agent i maps any negotiation outcome ω from outcome space Ω to a real-valued number in the range of [0, 1], and is defined as:

$$U_i(\omega) = \sum_{j=1}^{n} (w_j^i \cdot V_j^i(v_{jk})) \tag{1}$$

where ω is an outcome represented as a vector of values, with one value for each issue; v_{jk} is the k-th possible choice of issue j; and V_j^i is the evaluation function of agent i for issue j that maps a choice of issue j (e.g., v_{jk}) to a real number in the interval of [0, 1]; and w_j^i ($j \in \{1, \ldots, n\}$) the weighting preference which agent i ascribes to issue j.

During negotiation, both parties exchange offers in each round to express their demands, relying on the stacked alternating offers protocol [Aydoğan et al., 2017].

3.2 REINFORCEMENT LEARNING

We follow the standard protocol that formulates a RL environment as a Markov decision process (MDP), that is, $M = (\mathcal{S}, \mathcal{A}, \mathbb{P}, r, \gamma)$, where $\mathcal{S}$ is the state-space, $\mathcal{A}$ is the action space, $\mathbb{P} : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}'$ is the transition function, $r(s, a)$ is the reward function, and $\gamma \in [0, 1)$ is the discount factor. A policy is a distribution $\pi(a|s)$, which denotes the probability of taking action a_t conditioned on the current state s_t. The objective of the RL agent is to find a policy that maximizes the expected return $\mathbb{E}_\pi[\sum_{t=0}^{\infty} \gamma^t r_t]$. Everything in the negotiation scenario including the opponent is considered as the environment.

States. As negotiation domains varies significantly due to different structure (e.g., the issue number, the issue types, size of outcome space), states are necessarily described in a domain-independent way. Following the ideas presented in Sengupta et al. [2021], Wu et al. [2021], this work employs a similar approach by representing an outcome ω as $U_s(\omega)$ (U_s is the utility function of the negotiating agent). Specifically, two factors are taken into account. First, the timeline, which is relevant because negotiation fails if no agreement can be achieved before the deadline (T_{max}). Second, the offer trajectory, which is crucial because it has an strong impact on the agent's decision-making. Therefore, the state s at time t is defined as follows:

$$\begin{aligned} s_t = \Big(& \frac{t}{T_{\max}}, u_s(\omega_o^{t-3}), u_s(\omega_s^{t-3}), \\ & u_s(\omega_o^{t-2}), u_s(\omega_s^{t-2}), u_s(\omega_o^{t-1}), u_s(\omega_s^{t-1}) \Big) \end{aligned} \tag{2}$$

where $T_{\max}$ denotes the maximum number of rounds of a negotiation session, ω_o^{t-n} denotes the offer received from the opponent at step $t - n$, ω_s^{t-n} denotes the offer proposed by the DOREA agent, and u_s denotes the self utility function. Note that although more pairs of (ω_s^{t-n}, ω_o^{t-n}) (i.e., $n > 3$) could improve effectiveness of the agent at the cost of much more computational resources and time, the current choice already guarantees that the algorithm runs smoothly in practice and makes no significant differences compared to the case when $n = 5$ or 7 is adopted.

Actions. The set of actions at a given state consist of all possible target utility values in the range $[u_r, 1]$. So, the action at time t is defined as $a_t = u_s^t$ (where u_s^t denotes the utility of the next offer). To generate the offer corresponding to the utility value u_s^t, we define an inverse utility function $\mathcal{F}^{-1}$ that maps a real-valued number u to an outcome ω and selects the best possible outcome that maximizes the estimated opponent utility at the given utility. Formally, the inverse utility function is defined as

$$\mathcal{F}^{-1}\left(u_s^t\right) = \arg\max_{\omega} U_o^{'}(\omega) \tag{3}$$

where $U_o^{'}$ denotes the opponent's utility function estimated on the basis of issue frequency of the opponent's historical offers, following the approach of van Galen Last [2012].

Rewards. The agent is given a positive reward when an agreement is reached, and a punishment of -1 when no agreement can be settled before the deadline. The RL agent's acceptance strategy is simple, that is, if the opponent's offer is better than the intended next own offer, the agent then accepts it, otherwise rejects. Formally, the reward function is defined as follows:

$$r_{t+1}(s_t, a_t) = \begin{cases} U_s(\omega), & \text{if there is an agreement } \omega \\ -1, & \text{if no agreement reached by deadline} \\ 0. & \text{otherwise} \end{cases} \tag{4}$$

Actor-critic approaches can provide an effective way to optimize the RL objective. In the conventional actor-critic formalism [Barto et al., 1983, Sutton and Barto, 2018], an approximated Q-function Q_θ is learnt by minimizing the squared Bellman error (refereed to as policy evaluation), and optimizes the policy π_ϕ by maximizing the Q-function (referred as policy improvement). The Q-function $Q_\theta(s, a)$ is an estimation of how good is it to take action a at the state s. The above objectives are as follows:

$$\begin{aligned} Q(\theta) = \arg\min_{Q} \mathbb{E}_{(\mathbf{s},\mathbf{a},\mathbf{s}')\sim\mathcal{D}} \Big[& (Q(\mathbf{s}, \mathbf{a}) \\ & - \Big(r(\mathbf{s}, \mathbf{a}) + \gamma \mathbb{E}_{\mathbf{a}'\sim\pi_\phi(\mathbf{a}'|\mathbf{s}')} \left[Q_\theta\left(\mathbf{s}', \mathbf{a}'\right) \right] \Big)^2 \end{aligned} \tag{5}$$

$$\pi_\phi = \arg\max_{\pi} \mathbb{E}_{\mathbf{s}\sim\mathcal{D}} \left[\mathbb{E}_{\mathbf{a}\sim\pi_\phi(\mathbf{a}|\mathbf{s})} \left[Q_\theta(\mathbf{s}, \mathbf{a}) \right] \right] \tag{6}$$

where $\mathcal{D}$ can either be the replay buffer $\mathcal{B}$ generated by previous policy π_ϕ through online environment interactions, or a fixed dataset $\mathcal{D} = \{(s_t^i, a_t^i, s_{t+1}^i, r_t^i)\}_{i=1}^n$ as common in offline RL setting.

4 DOREA FRAMEWORK

The DOREA framework consists of two key components: an offline learning based strategy module and an strategy fine-tuning mechanism. Figure 1 provides an overview of the framework.

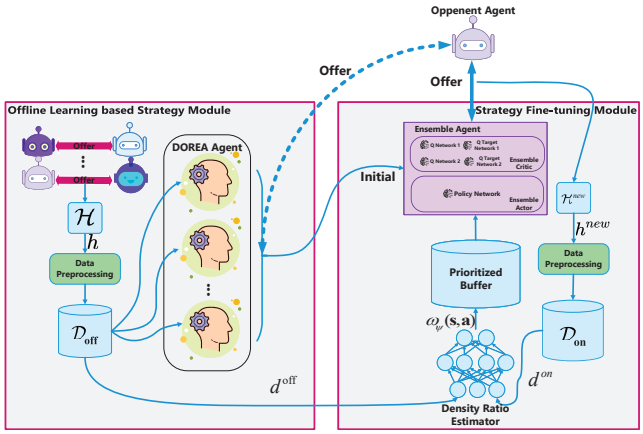

Figure 1: Overview of the proposed Deep Offline Reinforcement learning Negotiating Agent (DOREA) framework.

4.1 OFFLINE LEARNING BASED STRATEGY MODULE

The offline learning based strategy module comprises of two steps. First, a negotiation history denoted by $\mathcal{H}$ is collected. This history consists of previous negotiation traces between two parties, including the negotiation scenario, the exchanged offers between them, time stamp of each offer, both sides' preferences (utility functions), and the negotiation results (e.g., agreement/failure). A party follows a negotiation strategy, yet $\mathcal{H}$ can be obtained by a mixture of multiple strategies. These data h ($h \in \mathcal{H}$) are converted into transitions of RL (i.e., $h = (s, a, r, s')$) through a preprocessing procedure (e.g., mapping all offers to utility values r, generating corresponding action a and state s), and then saved as offline data $\mathcal{D}_{off}$.

Second, the module aims to learn an effective negotiation strategy from historical datasets $\mathcal{D}_{off}$ via offline RL. However, the negotiation datasets collected may be suboptimal (e.g., absent data or data having non-expert quality), the state and action space coverage is limited, and this may result in a distribution shift, that is, the offline RL-agent encounters online data $\mathcal{D}_{on}$ that have different state-action distribution from the offline data $\mathcal{D}_{off}$) — causing overestimation of the Q-value of out of distribution (OOD) action using classic off-policy RL algorithms. Consequently, the learned negotiation strategy might choose potentially inappropriate actions. Therefore, this framework employs Conservative Q-learning (CQL) [Kumar et al., 2020], which can reduce the harmful effect of a distribution shift by explicitly penalizing the Q-value of actions not available in offline dataset $\mathcal{D}_{off}$. CQL pessimistically evaluates the current policy and obtains the lower-bound of the real Q-function. It aims to training the Q-function by using the sum of standard temporal-difference (TD) error and the regularizer (see Eq. 9). This is achieved through minimizing the expectation of Q-value of action with overestimation on the sampling distribution, and maximizing the expectation of Q-value on the offline dataset. CQL can be instantiated as an actor-critic algorithm like SAC [Haarnoja et al., 2018]. SAC is an off-policy algorithm designed to optimize a stochastic policy, which objective is to both maximize the expected return and the entropy of the policy:

$$\pi_\phi = \underset{\pi}{argmax} \sum_{t=0}^{T} \mathbb{E}_{s,a\sim\pi} \gamma^t r_t(s,a) + \alpha \mathbb{H}(\pi(.|s)) \tag{7}$$

where $\mathbb{H}$ is the entropy and $\alpha > 0$ is the temperature parameter, γ is discount factor, and r_t is reward function at time-stamp t. The corresponding Q-function $Q^\pi(s,a)$ can be expressed as:

$$Q_\theta(s,a) = \underset{s,a\sim\pi}{\mathbb{E}} \left[\sum_{t=0}^{\infty} \gamma^t r(s,a) + \alpha \sum_{t=1}^{\infty} \gamma^t \mathbb{H}(\pi(\cdot \mid s)) \,|s,a \right] \tag{8}$$

Here, a variant of CQL – CQL($\mathcal{H}$) is chosen because it generally outperforms other variants [Kumar et al., 2020]. In order to more effectively mitigate the impact of distribution shift, multiple (N) pessimistic Q-functions are employed. Each policy evaluation steps $Q(\theta_i)$ ($i \in I$ and θ_i means the parameters for $i-th$ Q-function) minimize the following problem:

$$\begin{aligned} Q(\theta_i) = \min_Q \alpha \mathbb{E}_{\mathbf{s}\sim\mathcal{D}_{off}} & \underbrace{\left[\log \sum_{\mathbf{a}} \exp Q(\mathbf{s},\mathbf{a}) - \mathbb{E}_{\mathbf{a}\sim\hat{\pi}_\beta(\mathbf{a}|\mathbf{s})} [Q(\mathbf{s},\mathbf{a})] \right]}_{\text{CQL regularizer}} \\ & + \frac{1}{2} \underbrace{\mathbb{E}_{\mathbf{s},\mathbf{a},\mathbf{s}'\sim\mathcal{D}_{off}} \left[\left(Q_{\theta_i} - \boldsymbol{B}^{\pi_{\phi_i}} Q_{\bar{\theta}_i} \right)^2 \right]}_{\text{standard TD error}} \end{aligned} \tag{9}$$

where $\hat{\pi}_\beta(\mathbf{a}|\mathbf{s}) := \frac{\sum_{s,a\in\mathcal{D}_{off}} 1[s=s_0, a=a_0]}{\sum_{s\in\mathcal{D}_{off}} 1[s=s_0]}$ is the empirical behavior strategy, α is the trade-off factor, $\bar{\theta}_i$ is the delayed parameter, and $\boldsymbol{B}^{\pi_{\phi_i}}$ is the Bellman operator, which constitute the Bellman error with the third part of Eq. (9). Policy improvement step $\pi(\phi_i)$ is the same as SAC defined in Eq. (7). And the learned strategy is represented by an ensemble of the N CQL based Q-functions and policies that trained via update rules Eq. (7),(9) and expressed as $\{Q_{\theta_i}, \pi_{\phi_i}\}_{i=1}^N$, where θ_i and ϕ_i represent the parameters of the $i-th$ Q-function and policy, respectively. The corresponding Q-function and policy is described as follows:

$$\begin{aligned} Q_\theta &:= \frac{1}{N} \sum_{i=1}^{N} Q_{\theta_i}, \\ \pi_\phi(\cdot|s) &= \mathcal{N}\left(\frac{1}{N} \sum_{i=1}^{N} \mu_{\phi_i}(s), \frac{1}{N} \sum_{i=1}^{N} \left(\sigma^2_{\phi_i}(s) + \mu^2_{\phi_i}(s) \right) - \mu^2_\phi(s) \right) \end{aligned} \tag{10}$$

where $\theta := \{\theta_i\}_{i=1}^N$ and $\phi := \{\phi_i\}_{i=1}^N$.

4.2 STRATEGY FINE-TUNING MECHANISM

Having obtained a strategy via the offline learning based strategy module, the DOREA framework employs strategy fine-tuning to adjust its strategy when there is a change in the opponent's preferences or strategy in subsequent online

negotiation. To effectively adapt to changes in the opponent, inspired by the work of Lee et al. [2022], the strategy fine-tuning mechanism aims at safely utilizing online samples and mitigating the distribution shift more effectively.

Suffering from the distribution shift problem, a good initial offline strategy may be destroyed quickly using these online data directly with off-policy RL algorithms. It is thus necessary to utilize offline and online data effectively to fine-tune strategy. As such, a prioritized sampling scheme component called balanced experience reply is used. This component utilizes online data by sampling offline data related to the current policy. In this way, DOREA agent can implicitly recognize the change of an opponent's strategy or utility function without explicitly modelling the opponent.

The online negotiation data history is denoted as $\mathcal{H}^{new}$ and save them h^{new} ($h^{new} \in \mathcal{H}^{new}$) in $\mathcal{D}_{on}$ through the same pre-processing as in Section 4.1. The DOREA framework creates a prioritized buffer, which stores both the offline negotiation data $\mathcal{D}_{off}$ and the online data $\mathcal{D}_{on}$ respectively during fine-tuning. Then, the prioritized buffer sorts all available samples according to their $online\text{-}ness$. To measure $online\text{-}ness$ of samples, we use density ratio $\omega(s,a) := d^{on}(s,a)/d^{off}(s,a)$, a probability proportional to the density ratio between online samples and offline samples, where $d^{on}(s,a)$ and $d^{off}(s,a)$ denotes the distribution of state-action pairs in the online and offline buffer, respectively. DOREA estimates the density ratio by training a neural network $\omega_\psi(s,a)$ called density ratio estimator. The training procedure for the density ratio estimator $\omega_\psi(s,a)$ follows the approach of Sinha et al. [2022] and uses the variational representation of f-divergences ([Nguyen et al., 2007]). Let $f(y) := y \log \frac{2y}{y+1} + \log \frac{2}{y+1}$, and the Jensen-Shannon (JS) divergence is defined as $D_{JS}(P\|Q) = \int_{\mathcal{X}} f(dP(x)/dQ(x))dQ(x)$. Model ω_ψ is updated by maximizing the lower bound of the JS divergence:

$$\mathcal{L}^{\mathrm{DR}}(\psi) = \mathbb{E}_{x\sim P}\left[f'\left(w_\psi(x)\right)\right] - \mathbb{E}_{x\sim Q}\left[f^*\left(f'\left(w_\psi(x)\right)\right)\right] \tag{11}$$

where f^* is the convex conjugate of f. For the first term in Eq. (11), the expectation is estimated by sampling from $\mathcal{D}_{on}$, and the second is sampled from $\mathcal{D}_{off}$.

Additionally, we employ an ensemble agent whose parameters are initialized by $\{Q_{\theta_i}, \pi_{\phi_i}\}_{i=1}^N$ obtained in the offline learning module. θ and ϕ are updated via SAC update rules Eq. (5),(7), respectively, during strategy fine-tuning.

5 EXPERIMENTS

Three experiments are conducted in order to demonstrate the effectiveness of the DOREA framework. The first experiment explores the following three performance aspects: effectiveness of the negotiating agent strategy learned on the basis of previously collected offline data; impact of data collected by more advanced strategies on the performance of the DOREA agent; and performance of the learned strategy in comparison to the strategies used to collect the data. The second (third) experiment investigates whether the DOREA agent learned from offline datasets can also adapt to changes of its opponent's preferences (strategy) in subsequent online negotiations.

5.1 EXPERIMENTAL SETUP

Table 1: Statistics of all 18 domains in the experiments. The domains are classified into three groups according to outcome space (i.e., small, medium and large domains).

Domain	Outcome Space	Opposition	Number of Issues
NiceOrDie	3	0.840	1
Ultimatum	9	0.545	2
FiftyFifty2013	11	0.707	1
Laptop	27	0.160	3
Planes	27	0.164	3
DefensiveCharms	36	0.322	3
Coffee	112	0.447	3
Outfit	128	0.198	4
DogChoosing	270	0.051	5
Acquisition	384	0.117	5
HouseKeeping	384	0.272	5
Icecream	720	0.148	4
Animal	1152	0.110	5
Camera	3600	0.212	6
Lunch	3840	0.399	6
SmartPhone	12000	0.224	6
Kitchen	15625	0.057	6
Wholesaler	56700	0.308	7

In our experimental settings, each participant plays against all considered participants including itself (except when calculating the metric of utility against opponent as defined below) with given acting roles (e.g., "buyer" and "seller") in every domain for a number of repetitions. Moreover, in each repetition a pair of agents conduct negotiation twice where they exchange the order in which they start with bidding. The experiments consider the whole set of domains created for ANAC 2013. As shown in Table 1, these domains differ in their size of outcome space (i.e., the set of possible outcomes), ranging from 3 to 56700, in the opposition (i.e., the minimal Euclidean distance to the optimal outcome for both sides), ranging from 0.051 to 0.84, and in the number of negotiated issues, ranging from 1 to 7. Note that the choice of the ANAC 2013 domains is taken because 1) these domains cover a wide range of domain characteristics, 2) designers of all agents know these domains well and so none of these agents has a disadvantage, and 3) these domains are also adopted in other recent work [Sengupta et al., 2021, Wu et al., 2021, de Jonge, 2022] for comparability reasons. During each negotiation session, the reservation value for all domains is 0, and the maximum round per session is 1000. The repetition number is set to 300. For the implementation details of DOREA agent, the batch size is 256 and the size of both the offline and online reply buffer is set to 2e+6. The learning rates of the actor network and the critic network is 1e-4 and 3e-4, respectively. The discount factor in RL training is 0.99. DOREA agent is trained for 1e+6 timesteps.

Moreover, the CQL algorithm is based on the open SAC version [1], other parameter settings are identical to the setup of Kumar et al. [2020]. Following the suggestion of Lee et al. [2022], the ensemble size N is set to 5. More details can be found in the appendix.

The following metrics are considered in experiments:

(1) ***Domain utility***: the average utility of all agents $\in A$ (including self play) in domain $d \in D$, where A and D denote the set of agents and domains, respectively.

(2) ***Average utility***: the average utility acquired by an agent $a \in A$ when negotiating with every agent $b \in A$ (including a) across all domains D.

(3) ***Average utility against opponent*** a: the average utility obtained by other agents $b \in A \backslash a$ when negotiating against opponent a across all domains D.

5.2 INFLUENCE OF OFFLINE DATASET

To investigate whether a useful strategy can be learned through offline datasets and what the influence of offline datasets on the DOREA's performance is, two different datasets were collected. In both datasets, there is the same four opponents employing a different strategy from the four ANAC winner agents' strategies (winner strategies) – AgreeableAgent2018, PonpokoAgent, Caduceus and Atlas3 [2]. The first one (referred to as winner dataset) consists of the negotiation traces generated by four agents with each using one of the winner strategies, where each one plays against those opponents (except the opponent with the same strategy) in all 18 domains. The other dataset (referred to as random dataset) was built from negotiations between a simple random agent that uses a random bidding strategy and also accepts offers according to a probability distribution (random strategy) and the four opponents. Moreover, the negotiations between the random agent and the opponents were repeated three times in order to obtain an equal size of the winner dataset. Through training separately with the two different datasets, two negotiating agents referred to as DOREA-winner and DOREA-random can be acquired. Note that, as the experiment below aims at analyzing the influence of datasets on offline learning (corresponding to Sec. 4.1), the strategy fine-tuning mechanism is therefore disabled to avoid performance improvements achieved through this mechanism.

Figure 2 compares the performance of the two DOREA agents and two baselines (i.e., strategies used for collecting the two offline datasets) against the four opponents in the offline dataset. As depicted in the figure, DOREA-winner clearly achieved the best performance, whereas DOREA-random had a much lower utility against each of the four winners. Specifically, DOREA-winner led DOREA-random with a large margin of between 100.2% to 167.6% in the four cases, and it achieved a mean score of 0.85 against the four opponents, 129.1% higher than that of DOREA-random. The results indicate that a training of an DOREA agent with samples from advanced strategies than simple strategies can bring about a considerable performance improvement. This is because advanced strategies can exhibit more useful state-actions pairs leading to high rewards. Another valuable observation is that both DOREA-winner and DOREA-random managed to outperform the the strategies used for collecting the two offline datasets. More precisely, DOREA-winner exceeded the average performance of the four winner strategies in terms of average utility against opponent by 48.5%, and DOREA-random advanced the random agent by 33.2%. The DOREA framework's capability of solving a distribution shift (see Sec.4.1) may account for this success.

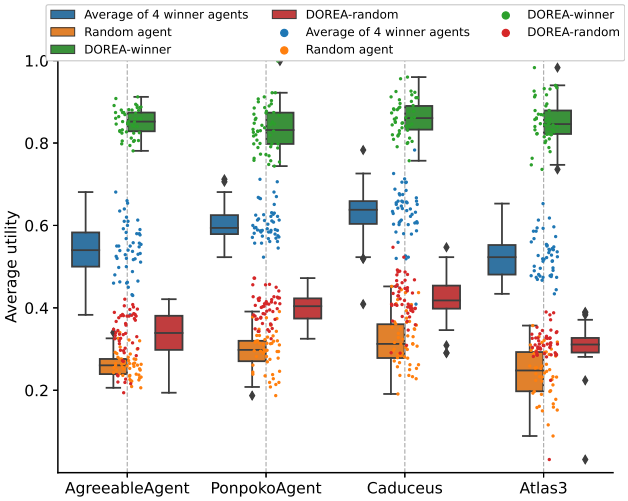

Figure 2: Box plots showing the utility against opponent (one of the four winner agents) for DOREA-winner, DOREA-random and two baselines (the random strategy and the average utility of the four winner agents). All results are obtained across all 18 domains. Points represent the utility of agreements reached when playing against each opponent, and farout is marked by the diamond symbol.

Next, to look closer into the performance of the DOREA agent, Figure 3 shows the tournament results for each of the 18 domains, where seven agents (including two DOREA agents, the random agent and four ANAC winner agents) are considered and the repetition was set to 100 for each domain to ensure statistical significance of the results. One can see that DOREA-winner was the most successful agent with a notable advantage, and ranked first in all domains except the NiceOrDie domain in which it ranked second. Moreover, it performed 38.0% higher than the average score of all agents across 18 domains, and outperformed the second best agent (Atlas3) by a margin of 24.2%. To sum up, the

[1] See https://github.com/vitchyr/rlkit.

[2] There were the ANAC winners in 2018, 2017, 2016 and 2015, respectively.

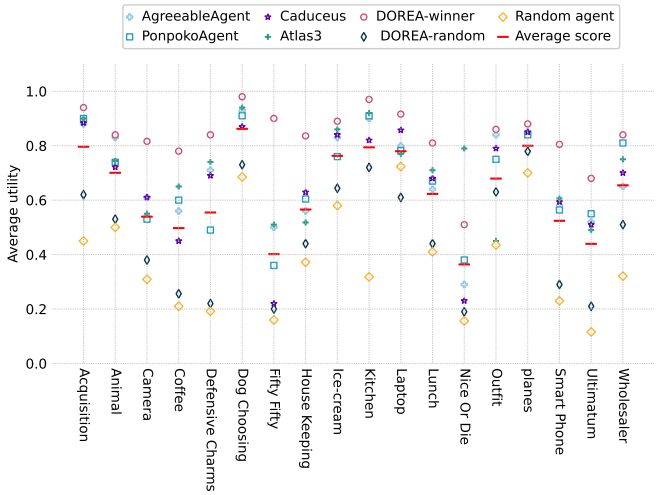

Figure 3: Domain utility of the two DOREA agents and ANAC winner agents in all 18 domains. The average score of all agents in each domain is marked as red solid line.

experimental results show that the DOREA framework is capable of learning an effective strategy from offline datasets and that the learned strategy was more performant than those strategies used for collecting the data.

5.3 PERFORMANCE OF DOREA WITH CHANGES IN THE OPPONENT PREFERENCES

As the opponent encountered in the offline dataset may change its preference profiles in subsequent online negotiations for many reasons that are hard to model. This experiment studies the performance of the DOREA framework against opponents with varying preferences. We assume that the opponent's preferences remain static for 250 sessions before being changed again. For simplicity, we also assume that the opponent keeps its strategy fixed when it changes preferences. 100 different sets of preference profiles of an opposing party are randomly generated for each domain (i.e., these preferences are different to that used in the offline dataset and are also different to each other). In particular, this experiment focuses on the offline-to-online performance against an opponent, that is, how well an agent can adapt to an opponent when it changes from the preferences shown in the offline dataset to some different preferences in subsequent online negotiations. The DOREA agent is trained with the winner dataset as described above. Three baselines are introduced for comparative evaluation – the DOREA agent without strategy fine-tuning (denoted as DOREA w/o sft), the RL-agent that employs SAC algorithm and learns from scratch online (denoted as SAC agent), and another SAC based agent initialized by the parameters of the DOREA agent (denoted as SAC-sft agent).

Illustrative examples of online negotiations against the four opponents in the Housekeeping domain are presented in Fig. 4, where the change in the opponent's preferences happens at session 0, and the results are averaged by negotiations where the opponent uses every one of the 100 different preferences. Quantitatively similar results have been obtained for the other domains, which are not reported here due to limited space. According to the figure, the DOREA agent clearly outperformed the baselines in terms of learning efficiency and final performance. Precisely, the DOREA agent achieved a stable performance around between 50 to 70 sessions, while both the SAC-sft and SAC agent reached it much slowly (approximately after 200 sessions). Besides, the DOREA agent obtained the highest average utility of 0.81, leading the SAC agent (i.e., learning from scratch) and the DOREA agent w/o sft (i.e., no fine-tuing) by a large margin. This shows the effectiveness of the strategy fine-tuning mechanism, which provides helpful offline data for the current negotiation and speeds up fine-tuning process, starting from pessimistic initialization.

Moreover, Table 2 summarizes the performance of the DOREA agent and the baselines in domains of small, medium and large size (refer to Table 1). Similar to the results observed above, the DOREA agent was still the best agent across the three classes of domains with an average utility of 0.837. It clearly achieved a better performance, leading the DOREA w/o sft by a margin of 28.8% on average. The SAC-sft agent, following DOREA agent, were ranked second in all three classes of domains. In sum, DOREA agent managed to outperform the baselines when competing against an opponent that changes its preferences in online negotiations.

Table 2: Average utility in three classes of domains, the bounds are based on the 95% confidence interval.

Domain	DOREA	DOREA w/o sft	SAC	SAC-sft
Small domain	$\mathbf{0.79}_{\pm 0.02}$	$0.57_{\pm 0.04}$	$0.61_{\pm 0.04}$	$0.74_{\pm 0.03}$
Medium domain	$\mathbf{0.88}_{\pm 0.03}$	$0.71_{\pm 0.05}$	$0.69_{\pm 0.06}$	$0.81_{\pm 0.04}$
Large domain	$\mathbf{0.84}_{\pm 0.03}$	$0.67_{\pm 0.03}$	$0.65_{\pm 0.40}$	$0.77_{\pm 0.04}$

5.4 PERFORMANCE OF DOREA WITH CHANGES IN THE OPPONENT STRATEGIES

This experiment investigates the performance of the DOREA agent against opponents who adopt a different strategy in subsequent online negotiations. Here we realistically assume that an opponent can use a new strategy that has not been seen during offline learning phase. As such, the opponent strategy pool includes the ANAC winner agents used in Sec. 5.2, that is, AgreeableAgent2018, PonpokoAgent, Caduceus and Atlas3. And the strategy pool considers the runner-ups of respective ANAC editions – Agent36, CaduceusDC16, YXAgent and ParsAgent as new strategies. Moreover, MiCRO [de Jonge, 2022], as a recently proposed effective negotiation strategy, is also included in the pool as well. The experiment settings here are similar to Sec.5.3

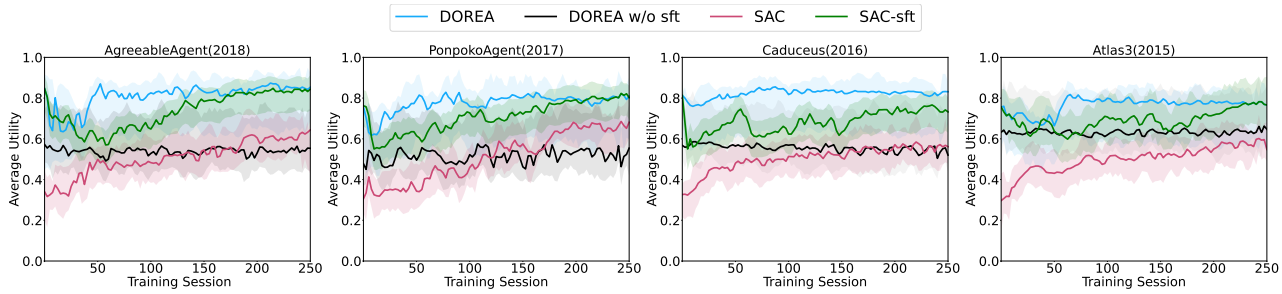

Figure 4: Four illustrative examples of fine-tuning performance of DOREA agent against four ANAC winner agents in the HouseKeeping domain. DOREA w/o sft represents the DOREA agent without strategy fine-tuning, SAC denotes the agent learning from scratch using SAC and SAC-sft denotes an SAC based agent initialized by the DOREA agent. The solid lines and shaded regions represent mean and standard deviation, respectively.

except that an opponent changes its strategy while its preferences are kept fixed.

Table 3: Comparison of DOREA agent with baselines against an opponent with a different strategy in online negotiations. All results are obtained across all 18 domains.

	Average util. against opponent	SAC-sft	DOREA w/o sft	DOREA
AgreeableAgent2018	$0.54_{\pm 0.06}$	$0.79_{\pm 0.06}$	$\mathbf{0.83}_{\pm 0.03}$	$0.82_{\pm 0.07}$
Agent36	$0.47_{\pm 0.04}$	$0.55_{\pm 0.04}$	$0.57_{\pm 0.04}$	$\mathbf{0.71}_{\pm 0.04}$
PonpokoAgent	$0.60_{\pm 0.04}$	$0.85_{\pm 0.02}$	$0.84_{\pm 0.05}$	$\mathbf{0.87}_{\pm 0.03}$
CaduceusDC16	$0.61_{\pm 0.02}$	$0.65_{\pm 0.04}$	$0.69_{\pm 0.05}$	$\mathbf{0.72}_{\pm 0.01}$
Caduceus	$0.62_{\pm 0.06}$	$0.84_{\pm 0.03}$	$0.86_{\pm 0.04}$	$\mathbf{0.88}_{\pm 0.07}$
YXAgent	$0.45_{\pm 0.08}$	$0.42_{\pm 0.07}$	$0.41_{\pm 0.02}$	$\mathbf{0.52}_{\pm 0.06}$
Atlas3	$0.52_{\pm 0.08}$	$0.81_{\pm 0.03}$	$0.85_{\pm 0.06}$	$\mathbf{0.87}_{\pm 0.01}$
ParsAgent	$0.53_{\pm 0.05}$	$0.61_{\pm 0.02}$	$0.65_{\pm 0.03}$	$\mathbf{0.73}_{\pm 0.05}$
MiCRO	$0.51_{\pm 0.02}$	$0.59_{\pm 0.05}$	$0.62_{\pm 0.05}$	$\mathbf{0.69}_{\pm 0.04}$

The results are given in Table 3, where the column "average utility against opponent" indicates the average utility against an agent (each row) achieved by all agents in the opponent strategy pool, and each entry of other columns means the average utility obtained by the column agent playing against the row agent. Some interesting observations follow from these outcomes. First, when encountering the four winner strategies that have been used during offline training, the performance of the DOREA agent w/o sft was better than the mean performance of the opponent strategies (see the second column of the table), 48.96% higher than the mean score of the opponents. However, this advantage in performance decreased about 13.95%, when the opponent switched to an unknown strategy. This indicates that overall the DOREA agent w/o sft was an effective strategy, but strategy adjustment was required when facing unknown strategies if a stronger performance is expected. Then, relying on the fine-tuning mechanism, the DOREA agent further improved performance, achieving on average an increase of 9.58% over its variant without fine-tuning. There was only one special case where the DOREA agent got a slightly worse utility (around 1.20%) than the DOREA agent w/o sft against AgreebleAgent2018. We suspect that in this case, the initial strategy was already good enough, making the DOREA agent end up with a similar performance level. The SAC-sft agent again lagged behind the DOREA agent with a considerable difference like results shown in Sec.5.3. These results validate that the strategy fine-tuning mechanism is effective for negotiations where the opponent changes its strategy.

6 CONCLUSION AND FUTURE WORK

This paper proposes a novel Deep Offline Reinforcement learning Negotiating Agent (DOREA) framework to learn strategy from previous negotiation datasets. The DOREA framework consists of two key components: the offline learning based strategy module and the strategy fine-tuning mechanism. The offline learning based strategy module leverages previously collected datasets to learn an effective negotiation strategy without interaction with opponents. Moreover, the strategy fine-tuning mechanism quickly fine-tunes the learned strategy via interactions and allows to adapt to changes of opponent preferences or strategies. Experimental results show that it is effective against a diverse set of state-of-the-art negotiating agents when exclusively using offline datasets, and is also capable of adapting to opponent preference or strategy changes.

We think the results clearly justify to invest further research efforts into this approach and open several new research avenues, among which we consider the following as most promising. First, as opponent modeling is another helpful way to improve the efficiency of negotiation, it's worthwhile investigating how to combine opponent modeling techniques with the proposed framework. Then, as the acceptance strategy also has impact on the performance of the learned strategy, it is very promising to explore the possibility to train the acceptance strategy instead of using the simple one used in the framework. A third important avenue we see is to enlarge the scope of the proposed framework to other negotiation forms such as concurrent negotiations and multi-lateral negotiations.

REFERENCES

R. Aydoğan, D. Festen, K. V. Hindriks, and C. M. Jonker. Alternating offers protocols for multilateral negotiation. In K. Fujita, Q. Bai, T. Ito, M. Zhang, R. Hadfi, F. Ren, and R. Aydoğan, editors, Modern Approaches to Agent-based Complex Automated Negotiation, page 153–167. Springer, 2017.

P. Bagga, N. Paoletti, B. Alrayes, and K. Stathis. A deep reinforcement learning approach to concurrent bilateral negotiation. In Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20, pages 297–303, 2020.

J. Bakker, A. Hammond, D. Bloembergen, and T. Baarslag. Rlboa: A modular reinforcement learning framework for autonomous negotiating agents. In Proceedings of the International Conference on Autonomous Agents and Multiagent Systems, pages 260–268, May 2019.

A. G. Barto, R. S. Sutton, and C. W. Anderson. Neuronlike adaptive elements that can solve difficult learning control problems. IEEE transactions on systems, man, and cybernetics, (5):834–846, 1983.

H.-C. H. Chang. Multi-issue negotiation with deep reinforcement learning. Knowledge-Based Systems, 211:106544, 2021. ISSN 0950-7051.

S. Chen, Q. Sun, Y. Heng, T. Yang, and J. Hao. Transfer learning based agent for automated negotiation. In Proceedings of the 22th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS '23. International Foundation for Autonomous Agents and Multiagent Systems, 2023.

X. Chen, A. Ghadirzadeh, T. Yu, J. Wang, Y. Gao, W. Li, L. Bin, C. Finn, and C. Zhang. Lapo: Latent-variable advantage-weighted policy optimization for offline reinforcement learning. In Advances in Neural Information Processing Systems, 2022.

D. de Jonge. An analysis of the linear bilateral ANAC domains using the micro benchmark strategy. In L. D. Raedt, editor, Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022, pages 223–229. ijcai.org, 2022. doi: 10.24963/ijcai.2022/32. URL https://doi.org/10.24963/ijcai.2022/32.

J. Devlin, M. Chang, K. Lee, and K. Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In J. Burstein, C. Doran, and T. Solorio, editors, Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers), pages 4171–4186. Association for Computational Linguistics, 2019. doi: 10.18653/v1/n19-1423. URL https://doi.org/10.18653/v1/n19-1423.

M. Fatemi, T. W. Killian, J. Subramanian, and M. Ghassemi. Medical dead-ends and learning to identify high-risk states and treatments. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, Advances in Neural Information Processing Systems, 2021. URL https://openreview.net/forum?id=4CRpaV4pYp.

S. S. Fatima, M. Wooldridge, and N. R. Jennings. An agenda-based framework for multi-issue negotiation. Artificial Intelligence, pages 1–45, 2004.

T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In International Conference on Machine Learning, pages 1861–1870. PMLR, 2018.

J. Hao, S. Song, H.-F. Leung, and Z. Ming. An efficient and robust negotiating strategy in bilateral negotiations over multiple items. Eng. Appl. Artif. Intell., 34:45–57, Sept. 2014. ISSN 0952-1976. doi: 10.1016/j.engappai.2014.05.008.

R. Higa, K. Fujita, T. Takahashi, T. Shimizu, and S. Nakadai. Reward-based negotiating agent strategies. In Proceedings of the 37th AAAI Conference on Artificial Intelligence, AAAI-23, pages 297–304, 2023.

N. R. Jennings, P. Faratin, A. R. Lomuscio, S. Parsons, C. Sierra, and M. Wooldridge. Automated negotiation: prospects, methods and challenges. International Journal of Group Decision and Negotiation, 10(2):199–215, 2001.

A. Kumar, A. Zhou, G. Tucker, and S. Levine. Conservative q-learning for offline reinforcement learning. Advances in Neural Information Processing Systems, 33:1179–1191, 2020.

S. Lee, Y. Seo, K. Lee, P. Abbeel, and J. Shin. Offline-to-online reinforcement learning via balanced replay and pessimistic q-ensemble. In A. Faust, D. Hsu, and G. Neumann, editors, Proceedings of the 5th Conference on Robot Learning, volume 164 of Proceedings of Machine Learning Research, pages 1702–1712. PMLR, 08–11 Nov 2022. URL https://proceedings.mlr.press/v164/lee22d.html.

S. Levine, A. Kumar, G. Tucker, and J. Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. CoRR, abs/2005.01643, 2020. URL https://arxiv.org/abs/2005.01643.

T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra. Continuous control with deep reinforcement learning. In 4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings, 2016.

X. Nguyen, M. J. Wainwright, and M. Jordan. Estimating divergence functionals and the likelihood ratio by penalized convex risk minimization. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, Advances in Neural Information Processing Systems, volume 20. Curran Associates, Inc., 2007. URL https://proceedings.neurips.cc/paper/2007/file/72da7fd6d1302c0a159f6436d01e9eb0-Paper.pdf.

R. F. Prudencio, M. R. O. A. Máximo, and E. L. Colombini. A survey on offline reinforcement learning: Taxonomy, review, and open problems. CoRR, abs/2203.01387, 2022. doi: 10.48550/arXiv.2203.01387. URL https://doi.org/10.48550/arXiv.2203.01387.

A. Sengupta, Y. Mohammad, and S. Nakadai. An autonomous negotiating agent framework with reinforcement learning based strategies and adaptive strategy switching mechanism. In Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS '21, pages 1163–1172, 2021.

A. Sengupta, S. Nakadai, and Y. Mohammad. Transfer learning based adaptive automated negotiating agent framework. In L. D. Raedt, editor, Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022, pages 468–474. ijcai.org, 2022. doi: 10.24963/ijcai.2022/67. URL https://doi.org/10.24963/ijcai.2022/67.

D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y. Chen, T. P. Lillicrap, F. Hui, L. Sifre, G. van den Driessche, T. Graepel, and D. Hassabis. Mastering the game of go without human knowledge. Nat., 550(7676):354–359, 2017. doi: 10.1038/nature24270. URL https://doi.org/10.1038/nature24270.

S. Sinha, J. Song, A. Garg, and S. Ermon. Experience replay with likelihood-free importance weights. In Learning for Dynamics and Control Conference, pages 110–123. PMLR, 2022.

R. S. Sutton and A. G. Barto. Reinforcement learning: An introduction. MIT press, 2018.

G. Tennenholtz and S. Mannor. Uncertainty estimation using riemannian model dynamics for offline reinforcement learning. In A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho, editors, Advances in Neural Information Processing Systems, 2022. URL https://openreview.net/forum?id=pGLFkjgVvVe.

N. van Galen Last. Agent smith: Opponent model estimation in bilateral multi-issue negotiation. In New trends in agent-based complex automated negotiations, pages 167–174. Springer, 2012.

L. Wu, S. Chen, X. Gao, Y. Zheng, and J. Hao. Detecting and learning against unknown opponents for automated negotiations. In D. N. Pham, T. Theeramunkong, G. Governatori, and F. Liu, editors, PRICAI 2021: Trends in Artificial Intelligence, pages 17–31, Cham, 2021. Springer International Publishing.

R. Yang, J. Chen, and K. Narasimhan. Improving dialog systems for negotiation with personality modeling. In ACL/IJCNLP, pages 681–693. Association for Computational Linguistics, 2021.

T. Yu, A. Kumar, Y. Chebotar, K. Hausman, S. Levine, and C. Finn. Conservative data sharing for multi-task offline reinforcement learning. Advances in Neural Information Processing Systems, 34:11501–11516, 2021.