Reward-on-the-Line Offline Reinforcement Learning for Conversational Agents

Anonymous Author(s)

ABSTRACT

Building conversational agents (CAs) that can converse with humans is a long-lasting artificial intelligence (AI) interest. A rapidly growing body of research and development has flourished to improve the capabilities and effectiveness of these agents. Recently, offline reinforcement learning (RL) has been applied to CAs to learn conversational strategies and policies directly from historical conversational data, without collecting live feedback from humans or simulators. However, a major technical challenge of offline RL is the extrapolation error caused by out-of-distribution (OOD) actions' evaluations, which can lead to over-optimistic policies. In this paper, we propose a novel offline RL method, reward-on-the-line, that generates OOD reward labels for conversational agents, effectively breaking the dependency between policy evaluation and value estimation, thus reducing the extrapolation errors and achieving superior performance on a legal domain conversational task. We have shown how to work on a publicly available dataset, with little pre-processing and domain knowledge required, and achieve strong performance. All codes will be made available after publication.

ACM Reference Format:
Anonymous Author(s). 2023. Reward-on-the-Line Offline Reinforcement Learning for Conversational Agents. In *CIKM '23: The 32nd ACM International Conference on Information and Knowledge Management, October 21-25, 2023, Birmingham, UK.* ACM, New York, NY, USA, 11 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Conversational agents (CAs) have become increasingly popular and indispensable in our daily life with the rapid advancement in Artificial Intelligence research. Reinforcement learning (RL) has been widely used in CAs for its extraordinary capability in learning conversational expressions and policies surpassing training data [33, 50]. The latest ChatGPT [33] agent is such an RL-based success. However, standard RL training is online, which requires live interactions with a user or a simulator for live feedback or rewards. This can be prohibitively expensive and time-consuming for small research groups who could not afford costly manual reward labeling as ChatGPT does. Moreover, conversations in professional domains, e.g. the Supreme Court legal conversations this paper studies, require reward labeling with professional knowledge, which further increases the labeling costs.

On the other hand, abundant historical conversational data is available in the form of recordings and logs. Offline RL [10, 21, 23, 42] is an alternative to the standard RL that can bypass the online training challenge by leveraging historical interactions without additional data collection. However, a major technical difficulty offline RL faces is extrapolation error. It is caused by distribution shifts between the behavior policy and the learned policy when rewards for out-of-distribution actions are no longer present. It can lead to serious overestimation of actions and appear in the context of conversations as dull responses (e.g., *"why is that? I mean, you know ..."*) and the imposter effect [9] when the agent sounds over-confident about inaccurate answers (e.g., *"the idea of eating glass actually has several unique benefits that make it worth considering as a dietary addition."*)

To address this issue, state-of-the-art offline RL methods employ techniques such as 1) policy regularization [12, 32], which regularizes the learned policy to be close to the behavior policy, for instance, by importance sampling (sampling more frequently from the action with a higher reward), or sample augmentation (generating additional data from the existing data to mitigate the distribution shift) [11]; 2) Conservative value estimation, which learns a conservative estimate of future returns to constrain the learned policy [20, 56, 61]; and 3) Ensemble dynamics for model-based methods [22, 36, 54]. However, these methods can be overdone and to be too pessimistic. More recent fixes [25, 55] were proposed to refine them to be less pessimistic, but either heavy-handed or overcomplicated. None of them address the cause of extrapolation errors directly and reduce the dependence between policy evaluation and action-value estimation. Therefore, extrapolation errors still exist, especially when the learning trajectory is long.

In this paper, we propose a novel offline RL method, *reward-on-the-line*, that generates reward labels for out-of-distribution actions when training conversational agents, offering an independent opinion to the original offline policy evaluation and effectively breaks the dependency between policy evaluation and value estimation. Our work is inspired by the recent findings in agreement-on-the-line [3] and accuracy-on-the-line [29]. They pointed out a strong linear correlation between labels in in-distribution (ID) data and in out-of-distribution (OOD) data. Our method leverages this understanding and uses an ensemble of neural networks to derive rewards for OOD actions generated during offline RL. In this work, we attempt to build a Virtual Justice in the U.S. Supreme Court who can work on the Appeal Court Cases as a judge. The

CIKM '23, October 21-25, 2023, Birmingham, UK

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

experiments are performed on publicly available legal conversation transcripts. The results show that our method performs superior to a few state-of-the-art offline RL agents.

Contributions of this paper include the following:

- It offers a working solution for a main obstacle in training RL conversational agents – the reward labeling challenge. With the novel reward-on-the-line method, this work makes use of unlabelled conversational history and generates new reward labels for offline Q-learning, with less erroneous value estimation.
- The resulting offline RL framework can be applied to any readily available historical conversation dataset, without the trouble of manual reward labeling and simulator construction;
- The work demonstrates conversational agents can be built offline for professional domains, including those highly-impactful, such as Supreme Court legal arguments.

2 RELATED WORK

2.1 Conversational Agents (CAs)

Conversational agents are computer programs designed to communicate with humans through voice or written conversations. Their functionalities range from answering simple factoid questions (e.g., Amazon Alexa) to open-ended discussion and active participation in creative projects (e.g., ChatGPT [33] and DALL-E [40]), and have been widely used in daily life and workspace. Many approaches leverage existing techniques in Information Retrieval (IR) and Natural Language Processing (NLP) and treat conversations as individual questions and answers (QA). They include 1) knowledge graph-driven question answering (KG-QA) [26, 27, 31, 59], which is trained to understand natural language questions and match them to appropriate queries to search against a knowledge graph; and 2) context-sensitive response retrieval [1, 37, 53, 60], which takes into account the context of a conversation (such as user profile and previous utterances) to select an appropriate response from a response collection.

Other approaches formulate CAs as sequential decision-making agents built upon Markov Decision Processes (MDPs) [6, 13, 46, 48, 58]. They include 3) classical dialogue systems consisting of a dialogue management and a natural language response generation (NLG) module; 4) supervised sequence-to-sequence methods [5, 15, 51] that trains language models to generate the next utterance given the current; and 5) RL-based methods [2, 47, 52, 62, 63].

The RL-based methods learn by interacting with an environment, usually the conversational partner or a simulator, and improve their conversation policies gradually. These agents are not constrained by the best performance in a training dataset. With properly-designed structures, RL algorithms can even exceed human performance. The most recent ChatGPT [33] leverages policy-based RL method, Proximal Policy Optimization (PPO) [8], and has demonstrated astounding performance. As a general-purpose large model, ChatGPT is trained on about 570GB data and 300 billion words obtained from books, web texts, Wikipedia, etc, and finally with months of training. However, ChatGPT is an online RL and relies on tremendous human efforts to label rewards for creating a reward model. This can be prohibitively expensive and time-consuming for most research labs that cannot afford such manual and resource costs; and may also cause ethical concerns, such as hiring low-paid labelers from Kenyan.[1]

2.2 Offline Reinforcement Learning (RL)

Offline RL [10, 21, 23, 42] is a type of RL where an agent learns from previously collected data without live interactions with an environment. It has shown promising results in Robotics [19, 34, 39, 45], NLP [17, 24], Manufacturing [64], Healthcare [44], Energy Management [28], and Finance [18]. All existing off-policy RL methods can be used as offline RL when substituting the replay buffer with historical interaction data. However, doing so creates a distribution shift between the behavior policy and the learned policy. Extrapolation errors can be caused by this distribution shift since instant feedback for OOD actions is no longer present and can lead to serious overestimation of actions.

To address this issue, many fixes have been proposed. For instance, policy regulation methods restrict the learned policy's divergence from the behavior policy. It can be done by importance sampling (sampling more frequently from the action with a higher reward) [12, 32] or sample augmentation (generating additional data from the existing data to mitigate the distribution shift) [11]. These sampling methods reduce the variance of policy estimates by sampling more actions with high rewards and sampling from larger datasets. Another family of offline RL is value-based [56, 61]. They add penalty terms to restrict the learned value functions. For instance, the conservative Q-learning (CQL) method learns a pessimistic estimate of the Q-value and optimizes the policy with it [20, 56, 61], so that the over-estimation issue can be mitigated. The agent takes fewer risky moves and improves the stability and robustness of the learned policy. Model-based offline RL methods leverage ensemble dynamics to obtain better predictive performance than any of the constituent learning algorithms alone for similar purposes.

The most similar piece of work to ours is perhaps Verma et al.'s CHAI [50]. It incorporates a pre-trained language model for utterance generation into an offline Q-learning framework. This combined approach allows the agent to utilize large amounts of offline conversational data. However, their method can only assign rewards for OOD actions by Q-function estimates, which is unstable and quickly accumulates extrapolation errors. In this paper, we propose a novel offline RL method for out-of-distribution reward generation. We leverage the mathematical correlation between ID and OOD data to derive appropriate reward labels for the out-of-distribution actions, which mitigate the shortcomings of CHAI significantly.

[1] https://time.com/6247678/openai-chatgpt-kenya-workers/.

Table 1: Example Courtroom Conversation.

Context: *A California trial court convicted Joshua Richter of burglary and murder. He exhausted his state court remedies and filed for habeas corpus relief in a California federal district court. Mr. Richter argued that he was denied effective assistance of counsel in violation of the Sixth Amendment. The district court denied the petition and was affirmed by the U.S. Court of Appeals for the Ninth Circuit.* **Question**: *Is a defense lawyer deficient for failing to consult blood evidence when planning strategy for trial?*

...

Justice: Did they explicitly say in every single case you have to consult an expert?

Attorney: No. They didn't say that.

Justice: Or did they say the circumstances of this case, given the nature of the issues, that consultation would have been effective?

Attorney: That is – that certainly is one reading of the Ninth Circuit's opinion. I submit that the more correct reading, so to speak, would be if the Court looks at the language that the Ninth Circuit uses in discussing this standard ...

Justice: We could take issue with the timing of that consultation. You would have no quarrel with saying it would have been ineffective for that counsel to have failed to confer with an expert, wouldn't you?

Attorney: No, I would disagree with that.

Justice: You would say, even if the expert were to – an expert would have given that kind of exculpatory information, that would not have been ineffective?

...

Justice: All right. Thank you, counsel, counsel. The case is submitted.

3 PROBLEM SETUP

This section presents the problem setup, including our conversational domain and task, the offline RL setting, and the issue of extrapolation error.

3.1 Virtual Justice in Courtroom Conversations

In this paper, we design and develop a conversational agent that works as a virtual Justice in the US Supreme Court for Appeal Court Cases. In an Appeal Court case, there is one chief Justice, zero or more associate Justices, the petitioner counsel(s) (also known as attorneys), and the respondent counsel(s). Usually, counsel representing the competing parties of a case each have 30 minutes in which to present their side to the Justices. The Justices may interrupt these presentations with comments and questions, leading to interactions between the Justices, the attorneys and, in some cases, the amici curiae, who are not a party to the case but nonetheless offer information that bears on the case to assist the Court. At the end of the court, the Justice will make a final decision on either approving or denying the appeal case. The entire courtroom conversation, including presentations, examinations, questioning and answering, is for the Justice to arrive at this final decision in a fair, well-informed, and just manner. Figure 1 illustrates the Supreme Court's Appeal Court's conversational structure.

We develop and evaluate our agent on the Supreme Court dataset from Oyez.org, a free law project from Cornell's Legal Information Institute. It's a multimedia archive and the most complete and authoritative source for all of the Court's audio since 1955. The audio archive contains more than 110 million

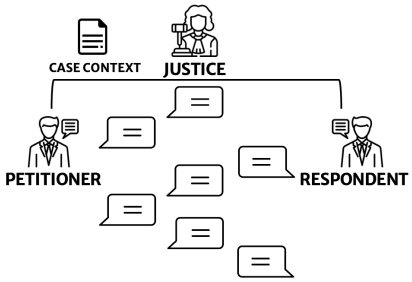

Figure 1: Appeal Court Conversation Structure.

words in more than 9,000 hours of audio-synchronized data in more than 7,000 cases, based on the court transcripts, to the sentence level. Oral arguments are, with rare exceptions, the first occasion in the processing of a case in which the Court meets face-to-face to consider the issues. For each case, we transcript the cases into natural language conversations between the above roles, e.g. the Justice and the counsels/attorneys.

Each case has two parts: context and transcript. The context includes the "Facts of the Case and Questions": Facts of the Case summarize the case's background and briefly introduce the opinion from both sides; Questions are provided by the judges and considered as the case's core conflict. The context provides essential information about a case, and we leverage them when developing the language model response generator. The transcript includes all the conversations in the oral argument of the case. Other metadata such as docket number, dates granted and argued, and name of petitioner and respondent are not relevant to our task and are skipped. Table 1 shows the context and example transcript for a case.

3.2 Setup as an Offline RL

We formulate the task of building a Supreme Court Virtual Justice using offline RL. The RL agent takes the role of the Justice and the environment is the attorneys and their utterances. The formulation builds on top of MDP and consists of a sequence of tuples of $\langle S, A, R \rangle$ that represent the state, action, and reward for a sequence of conversational decisions. In online RL, the agent can interact with the environment to obtain a trajectory τ of such tuples in real-time; while in offline RL, pre-collected interaction tuples are used instead. Below describes our RL settings for virtual Justice and implementations:

State S: containing four parts $s = (s_c, s_{type}, s_u, s_h)$, with s_c the context, i.e. the Case's background and Questions in the metadata, for an entire courtroom conversation, s_{type} the dialogue act type of the previous utterance (usually given by an attorney), s_u the previous utterance (usually given by an attorney), and s_h all the utterances (both attorney's and Justice's) in the conversation up to this point. Typical dialogue acts in legal conversations include *greeting, proposal, counter-proposal, examining, referencing, appointing, denial, and approval*, etc.

Action A: having two parts $a = (a_{type}, a_u)$, where a_{type} is the dialogue act type for the current utterance and a_u is the utterance. Note they are actions of the Justice's.

Reward R: Each Appeal Court case ends with a final decision from the Justice. It is a binary decision to either grant the appeal or deny it. We can extract this final decision via pattern-based information extraction from the conversation transcripts during training. In testing, we can set it as an input parameter to start the conversation. For a Justice agent, however, the main purpose is not to reach a particular decision, but to be able to reach one with high clarity and sufficient evidence. Therefore our rewards are metrics that measure the effectiveness of an utterance (by the Justice) to help dig out useful information for this final decision-making. In this paper, we utilize a set of implicit rewards as well as generate rewards for OOD actions (detailed in Section 4.3).

The implicit rewards r_{im} we use include *repetition* (a positive reward, if the attorney repeats what the agent said, which suggests the attorney agrees or accepts what the agent just said), *speech speed* (high talking speed can be a negative reward since it may indicate criticism or argument), *percentage of interjections* (e.g., expressions that are indicators of either agreement or disagreement with previous utterance from the agent), *on-topic relevance* (cosine similarity between SBert representation [41] of the utterance a_u to that of the context s_c), and *sentiment*.

Our intuition behind using *sentiment* as reward is that empathy in conversation leads to more agreeable conversations. If the sentiments of the Justice and the attorneys look close, which would indicate that the attorney gives a convincing argument. It indirectly shows that the Justice asked a good question and should receive a positive reward for her effective question. Thus, our sentiment-based reward measures the difference between the Justice's utterance a_u and the attorney's utterance s_u:

$$r_{sent}(a_u) = |sentiment(a_u) - sentiment(s_u)| \tag{1}$$

where $sentiment(.)$ assigns a sentiment score between [-1,+1] to an utterance using VADER, a sentiment analyzer [16].

Goal of Offline RL: Not different from standard RL, the goal of offline RL is to learn the best conversational policy π^* that maximizes the accumulative expected return,

$$\pi^* = \arg\max_{\pi} E_\tau[\sum_{t=1}^{T} \gamma^t r(s_t, a_t)], \tag{2}$$

where $\tau = \{\langle s_1, a_1, r_1\rangle, ...\langle s_t, a_t, r_t\rangle, ...\}$ is a conversation trajectory that can contain ID or OOD actions; γ is a discount factor.

3.3 Issue of Extrapolation Error

Deep Q-network (DQN) is a popular value-based RL method commonly used for discrete decision-makings, such as conversational agents. It can be made offline by filling its replay buffer with historical, instead of live, interactions. We can sample a batch of tuples, $\langle s, a, r, s'\rangle$, from the conversation histories and update the Q-value function network Q_{θ^*} by minimizing a modified Bellman loss:

$$Q_{\theta^*}(s,a) = \arg\min_{\theta} \sum_{(s,a)\in B} \Big(Q_\theta(s,a) - Q_{tar}(s,a)\Big)^2, \tag{3}$$

where s' is the next state, B is the replay buffer, which is a subset sampled from the whole trajectory $\{\langle S, A, R\rangle\}^{n_B}$. $Q_{tar}(s,a)$ is the target Q-network and can be calculated by the Bellman equation, $Q_{tar}(s,a) = R(s,a) + \gamma \mathbb{E}_{s'}[\max_a Q(s',a)]$.

Algorithm 1 Reward-on-the-line Offline RL for CAs

Input: Training Transcripts D, Q-networks Q^A and Q^B, Fine-tuned pre-trained Language model LM, Q-network sync interval η
Split D into D_{tran}, D_{val} and D_{OOD}, let $D_{ID} = \{D_{tran}, D_{val}\}$
Fit action agreements into a line:
$Agr(D_{OOD}) = k \cdot Agr(D_{ID}) + b$ (Eq. 11 to 13)
for $episode = 1, E$ **do**
  Sample a batch of transition tuples $(s_t, a_t, s_{t+1}))$ from D_{tran}
  Derive implicit reward r_t^{im} for each tuple (s_t, a_t, s_{t+1}) (Section 3.2)
  Store tuples $(s_t, a_t, s_{t+1}, r_t^{im})$ in replay buffer B
  for $t = 1, T$ **do**
    Sample a tuple of $(s_t, a_t, s_{t+1}, r_t^{im})$ from B
    for $i = 1, |\Omega|$ **do**
      Generate an OOD response a_{ti} for s_t using LM:
      $a_{ti} \sim LM(.|[s_{h,t}, s_{c,t}])$
      Obtain reward for a_{ti}:
      $r = \begin{cases} r_t^{im} & if\ a_{ti} == a_t, \\ \min(r_{OOD}(a_{ti}), \overline{R}(a_{ti})) & otherwise. \end{cases}$ (Eq. 14 and Eq. 4)
      Calculate CQL regularizer f^{cql}
      Update Q-network functions for Q^A and Q^B :
      $Q^A(s_t,a_t) \leftarrow Q^A(s_t,a_t) + \alpha(r + \gamma \max_a Q^B_{tar}(s_{t+1},a) + f^{cql} - Q^A_{tar}(s_t,a_t))$
      $Q^B(s_t,a_t) \leftarrow Q^B(s_t,a_t) + \alpha(r + \gamma \max_a Q^A_{tar}(s_{t+1},a) + f^{cql} - Q^B_{tar}(s_t,a_t))$
    end for
    Every η steps reset $Q^A_{tar} = Q^A$ and $Q^B_{tar} = Q^B$
  end for
end for
return $Q = Q^A_{tar} + \beta Q^B_{tar}$

A key invention in offline RL is that it makes use of the Q-value function to assign rewards to an OOD action a_{OOD}. Because the Q-value function predicts the overall expected reward of executing an action, which is somehow an indicator of reward, too. Suppose we have a reward function $\overline{R}$ based on the Q-function:

$$\overline{R}(s,a) \leftarrow \delta Q(s,a), \tag{4}$$

Then, Eq. 3 becomes:

$$Q_{\theta^*}(s,a) = \arg\min_{\theta} \sum_{(s,a)\in B} \Bigg(\underbrace{Q_\theta(s,a) - \Big(\delta Q_\theta(s,a)}_{\text{dependency}} + \gamma \mathbb{E}_{s'}[\max_{a'} Q(s',a')]\Big)\Bigg)^2, \tag{5}$$

which shows a dependence between Q-value estimation and policy evaluation, which can lead to severe extrapolation errors and worsen the over-estimation issue of Q-learning. Fig. **??** illustrates the dependency and the error.

4 PROPOSED METHOD

In this work, we propose to build the conversational agent with offline Q-learning. When collecting learning experiences with all possible actions for the next state, some actions will be OOD, i.e., these actions were never picked in that particular state by the behavior policy used to construct the training set. In such

circumstances, offline Q-learning relies on the current Q-value function's ability to extrapolate beyond the training data, and uses that to evaluate actions that are out-of-distribution, as in Eq. 4. It creates a dependence between value estimation and policy evaluation and results in extrapolation errors, as shown in Eq. 5. Since RL's maximization is greedy, these extrapolation errors worsen its bias toward an overestimation of actions. This overestimation issue can be very severe due to the high dimension in the NL utterances/actions.

Our approach leverages a few techniques to remedy the issue. They include conservative Q-learning, double Q-learning, action generation with fine-tuned domain-specific language models, and OOD reward generation. The overall framework is a double conservative Q-learning framework, which uses double Q-learning with penalized Q-function estimation. The new actions are generated by a GPT-2 language model [38] fine-tuned on the training dataset to stay close to the domain. A novel reward generation method, reward-on-the-line, is presented to break the dependence in extrapolation.

4.1 Double Conservative Q-Learning (DCQL)

Conservative Q-learning (CQL) [20] is used as the basic offline RL framework in this work. It explicitly penalizes the Q-value of actions not seen in the dataset and reduces over-estimation. On the original DQN, CQL adds an additional conservative regularizer to the Q-function estimation. The Q-Learning function with CQL is thus:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha(Q_{tar}(s_t, a_t) + f^{cql} - Q(s_t, a_t)), \quad (6)$$

where f^{cql} is the regularization term

$$f^{cql} = E_s \left[\log \sum_a \exp(Q(s_t, a_t)) - E_a(Q(s_t, a_t)) \right], \quad (7)$$

and Q_{tar} is the target network:

$$Q_{tar} = \overline{R}(s, a) + \gamma \mathbb{E}_{s'} \mathbb{E}_{a'} [\max_{a'} Q(s', a')] \quad (8)$$

where $\overline{R}(s, a)$ is $\delta Q(s, a)$ as in Eq. 4.

To further improve the model's stability, we propose using double Q-learning to continue reducing overestimation. It leverages two independent Q estimates Q^A and Q^B for value estimation and action selection separately:

$$\begin{cases} Q^A \leftarrow Q^A + \alpha(r + \gamma \max_{a'} Q^B_{tar}(s', a') + f^{cql} - Q^A) \\ Q^B \leftarrow Q^B + \alpha(r + \gamma \max_{a'} Q^A_{tar}(s', a') + f^{cql} - Q^B), \end{cases} \quad (9)$$

The overall Q-function estimation is $Q = Q^A + \beta Q^B$.

4.2 Domain-Specific Action Generation (AG)

When an offline RL agent proposes actions that are out-of-distribution, it is ideal that these OOD actions/utterances are still within the conversation domain and on-topic. In this paper, we propose to leverage pre-trained language models and fine-tune them to a specific language domain for more human-like, professional conversations.

Particularly, we fine-tune a GPT-2 language model [38] with the entire training dataset D. The language model LM outputs an action distribution over utterances. We then use the fine-turned language model to generate new actions given the conversation history s_h and the conversation's context s_c:

$$a_u \sim LM(.|[s_h, s_c]) \quad (10)$$

We use the language model to generate a set of candidate utterances $\Omega = \{a_u\}$ for the current state, where the size of Ω is set to 10. They are most likely to be out-distribution since they are generated, not from the training transcripts. They will also be combined with the a_{type}, which is generated by a discrete uniform distribution over all legal dialogue acts, to form the candidate action set.

This generation of candidate responses significantly increases the dialogue's variation and diversity. It can make the Justice's utterances, mostly questions to the counsel, to be more natural and human-like. It also decreases the possibility of encountering OOD actions since the actions are now made closer to the training dataset. However, many of them are still out-of-distribution and need to find their reward labels.

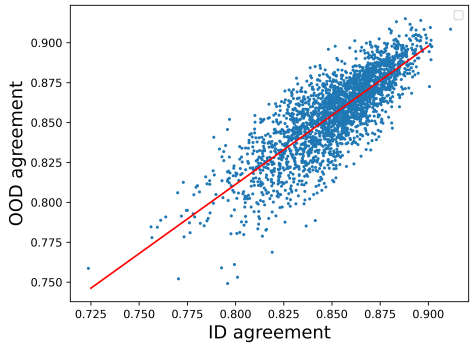

Figure 2: Linear relation between ID and OOD action selection agreements. The agreements are among similar Q-networks for in-distribution validation set vs. OOD set.

4.3 Reward-on-the-Line (ROL)

In this paper, we propose to generate reward labels for the OOD actions. It aims to break the dependency between policy evaluation and value function estimation (as shown in Eq. 4), thus reducing the extrapolation errors and yielding more accurate and reliable reward labels.

Recently, Miller et al.[29] found that a model's in-distribution (ID) accuracy has a strong linear correlation with its OOD accuracy. Further, Baek et al. [3] discovered that the agreement between the predictions of any two pairs of neural networks on OOD data also carry the same linear correlation with their agreement on ID data.

We did an experiment to validate the agreement-on-the-line phenomenon. To do so, we sampled a training dataset and a test dataset from the original courtroom dialogue dataset and use the training dataset to represent ID data and the test dataset OOD data. We trained 50 Q-networks with the same neural architecture but different and randomized parameter initializations and used the same training dataset with different data

reading orders to calculate the averaged agreements of selected actions between any pair of Q-networks on ID data and OOD data. The results are shown in Fig. 2, a sparse and scattered graph. However, we can still observe that these agreements fall into a line, which suggests that the Q-learning's action selection follows the phenomenon of agreement-on-the-line.

Inspired by this finding, we propose a reward-on-the-line method that utilizes the linear relationship between ID and OOD agreements for reward label generation. This way of obtaining rewards is independent of using the Q-function, thus can remedy the issue of accumulating extrapolation errors in value-based offline RL.

First, identify the linear relation. With random splittings of the entire dataset into D_{ID} and D_{OOD}, we aim to obtain a linear relation $Y_{OOD} = k \cdot Y_{ID} + b$ between the action selection agreements Y over D_{ID} and D_{OOD}. The steps are:

a) We measure the agreement of action selections among an ensemble of similar Q-networks. Let $\{Q\}^n$ denote the set of Q-networks trained on D_{train}, which is a training set partitioned from D_{ID}. Given any pair of models $Q_i, Q_j \in \{Q\}^n$ and any dataset D that can perform action selection with Q-learning, the expected agreement of action selection is:

$$Agr(D) = \frac{1}{|D|} \sum_{s,a \sim D} \mathbb{E}_{i,j}\{1 \; if \; \arg\max_a Q_{\theta_i}(s,a) == \arg\max_a Q_{\theta_j}(s,a)\} \tag{11}$$

Note that the calculation of the agreement does not require knowing the labels – it only cares about whether they share the same label, not the label itself. We then estimate this agreement quantity for data contained in D_{ID} and D_{OOD}, respectively:

$$Agr_{ID} = Agr(D_{val}), \quad Agr_{OOD} = Agr(D_{OOD}) \tag{12}$$

where D_{OOD} is the shifted distribution of interest and D_{val} is a validation set partitioned from D_{ID}, which allows us to estimate ID agreement.

b) We estimate the slope k and bias b by linear regression [43] using sklearn [35].

$$(k, b) \leftarrow \arg\min_{k,b} \sum (Agr_{OOD} - k \cdot Agr_{ID} - b)^2 \tag{13}$$

Second, obtain OOD reward. The steps are:

i) Based on the linear model (k, b) and the implicit rewards r_{im} (Section 3.2), we can derive OOD reward r_{OOD} for the OOD actions. These implicit rewards are real environmental feedback but are sparse and only available for some actions. We thus denote them as in-distribution rewards r_{ID} for those in-distribution actions/utterances that appear in the training trajectories. For a newly-generated action $a_u \in \Omega$, which is an OOD action at state s, its reward will be:

$$\forall a_u \in \Omega, \quad r_{OOD}(s, a_u) = k \cdot r_{ID}(s, a) + b \tag{14}$$

where r_{ID} is the implicit reward that can be obtained from the tuple $\langle s, a, r_{im} \rangle$ in the training trajectory for the same state s. Note that the tuple $\langle s, a, r_{im} \rangle$ already exists in the training trajectory. All OOD actions $a_u \in \Omega$ will share the same reward r_{OOD}. It is fine because what matters is the magnitude of the rewards and the overall optimization of an RL method is based on long-term accumulative rewards, not one immediate reward at a single time step. As long as it is a good estimation of a reward's magnitude, sharing rewards will not impact much on the policy that is learned eventually.

ii) Lastly, these OOD rewards calculated by reward-on-the-line can be used to compute the Q-target function in a way that is similar to Eq. 8:

$$Q_{tar} = r_{OOD} + \gamma \mathbb{E}_{s'} \mathbb{E}_{a'} [\max_a Q(s', a')], \tag{15}$$

Note that we no longer need to use $\overline{R}$ as in Eq. 4, which depends on previous Q-function estimations. This offers a solution to break the dependency between value estimation and policy evaluation. Algorithm 1 details the proposed method.

Table 2: Main Results: Effectiveness on In-distribution (ID) and out-of-distribution (OOD) Action Selection.

	In-distribution		Out-of-distribution	
	P@1 %	MRR	P@1 %	MRR
GPT-2 [38]	71.32	85.77	69.79	84.94
DDQN [49]	76.93	88.46	72.73	86.37
DQN [30]	75.15	87.51	72.21	86.24
PPO [8]	74.16	87.08	71.21	85.63
CHAI [50]	75.53	87.76	72.49	86.24
DCQL-AG-ROL (Ours)	**80.23**↑	**90.78**↑	**75.46**↑	**87.74**↑

4.4 Implementation Details

For the pre-trained language model, we use GPT2-medium from all the released versions. Mentions of names of the attorney and Justice in the transcripts are masked out with a special token to ease the burden to train the language model. To generate a candidate responding utterance, we concatenate the context of a case s_c and the current conversation history s_h and use them as the input of the language model. The language model outputs the next utterance; the process can be repeated to obtain multiple utterances.

Our Q-networks are feedforward neural networks that map states and actions to the Q-values. To transform the utterances into vectors for later training, we use our fine-tuned GPT-2 language model to embed the entire dialogue history and the current utterance. Then we concatenate state and action embeddings with the dialogue act type a_{type} to build a single vector to feed into a Q-network. Since we use double Q-learning in the work, we have more than one Q-network. Each Q-network is parameterized as a 2-layer feed-forward network with hidden sizes of 256 and ReLU nonlinearities. Reward-on-the-line is simple and efficient, with only $O(n)$ additional computations added to a baseline framework. We empirically set $\gamma = 0.99$ in Eq. 15 and $\alpha = 1, \beta = 1$ in Eq. 6 to Eq. 9.

5 EXPERIMENTS

We evaluate our approach using publicly available Supreme Court Appeal Court Cases as described in Section 3.1. It contains over 4GB of data and a total of 7,225 legal cases from the year 1955 to the year 2022. The dataset is divided into training,

Table 3: Ablation Study: Effectiveness on In-distribution (ID) and out-of-distribution (OOD) Action Selection.

	In-distribution		Out-of-distribution	
	P@1 %	MRR	P@1 %	MRR
DCQL-AG-ROL	**80.23**	**90.78**	**75.46**	**87.74**
CQL-AG-ROL	77.86	88.94	74.08(↓ 1.38)	87.00(↓ 0.74)
DCQL-AG	77.47	88.79	73.64(↓ 1.82)	86.81(↓ 0.93)
DCQL	74.97	87.45	71.23 (↓ 4.23)	85.63(↓ 2.11)

validation and test sets, with a ratio of 8:1:1. When calculating reward-on-the-line as in Section 4.3, we consider both the validation and training datasets as in-distribution and the test dataset as out-of-distribution and will use the namings ID and OOD to refer to them, respectively. Table 4 provides more details about dataset statistics.

Table 4: Dataset Statistics.

	min	average	max
# Turns per conversation	81	225	789
# Words per utterance	6	61	1,679
# Words per conversation	5,974	11,176	32,270
	approved		rejected
Case Decisions	460 (67%)		222 (33%)

5.1 Systems Under Comparison

We compare the proposed method, DCQL-AG-ROL (Section 4), with the following baselines. They include best-performing RL agents, such as DQN [30], DDQN [49], and PPO [8], which is the main RL framework behind ChatGPT [33]. They are made offline by sampling interactions from the training dataset. We also include an NL response generation agent that only uses the language model GPT-2 [38]. The last baseline agent is a state-of-the-art offline RL agent, CHAI [50]. It also uses Conservative Q-learning and language model but does not have double Q-learning and reward-on-the-line. It was reported on negotiation conversations and has been the SOTA.

These methods are all re-implemented by us and applied to the same courtroom conversation dataset. They share the same training trajectories for their training in this experiment. Among these agents, DQN, DDQN and PPO share the same 2-layer feedforward network structures as our Q-networks. They all use a learning rate of 3×10^{-4}, a batch size of 128, and a total of 5,000 training steps. For the pure GPT-2 language model and CHAI, they use the same GPT2-medium model as ours with the same hyperparameters for generating utterances, where $top_k = 50, top_p = 0.85$. We fine-tune the language models and train these agents on Nvidia 3090 GPU (24 GB memory). It lasts over 6 hours for the fine-tuning and over 2 hours for each agent's training on average.

5.2 Objective Evaluation

Evaluating conversations is challenging because human evaluation is indeed the best way to evaluate them. However, constrained by the costs of hiring human evaluators, not everyone can afford large-scale manual evaluations; without the scale, however, evaluations may not be conclusive. One form of automatic evaluation for conversations involves the challenge of creating simulators [4, 57], which may not be easily available for a historical conversation dataset as we use here.

In this paper, we manage to use the original historical conversation dataset as is and perform an objective evaluation. It allows us to compare our offline method against a range of existing CA and RL approaches. The idea is that for any state s_t, we evaluate an agent's policy by its ability to select the original action/utterance to the top rank when it is mixed with other ID or OOD actions. That is, we treat the utterances of the original actions in the training dataset as the gold standard. Although in theory, offline RL agents can generate better results than what is in a training dataset, we think it is fair to set the original actions as the gold standard for the purpose of evaluation due to the rather high performance of humans.

Since the action selection task can be viewed as a ranking task, we leverage retrieval-based effectiveness metrics: 1) Precision at Rank Position 1 [7]. $P@1 = \frac{\text{\# gold standard utterance @ rank 1}}{\text{\# total states}}$ 2) Mean Reciprocal Rank (MRR) [7]. $MRR = \frac{1}{\text{\# states}} \sum_{i=1}^{\text{\# states}} \frac{1}{rank_i}$. It is the mean of the inverse of the first occurrence of the gold standard (original) utterance in a list of actions/utterances for any state s_t. In this paper, we report both metrics and their effects on the in-distribution validation set (ID actions) and the OOD actions. Therefore, our metrics are ID P@1, OOD P@1, ID MMR and OOD MRR.

5.2.1 Effectiveness. Table 2 and Figure 3 show the main results on effectiveness for all baseline agents and our proposed method. As we can see in Table 2, our method outperforms all baseline systems, including the latest offline RL SOTA (i.e., CHAI), and the method behind ChatGPT (i.e., PPO) by a statistically significant gap. It outperforms CHAI on ID $P@1$ by 6% and OOD $P@1$ by 4% and outputforms the rest agents in both ID $P@1$ and OOD $P@1$ by a large 7-12% surplus.

Figure 3 reports OOD P@1 for all training steps. It reveals that our method consistently outperforms other baselines at all time steps and achieves 5% higher performance than the SOTA. It shows that the reward-on-the-line idea is highly effective in helping offline RL for building conversational agents.

Besides the use of reward-on-the-line, another reason why our method is better than CHAI could be that the latter was mainly designed for negotiation which is a goal-oriented conversational task with a more rigid structure than legal argumentation. It is worthwhile noting that tasks with clear goals are in general easier for RL methods. CraigslistBargain task [14] has a clear task goal (a bargain for a lower price). In this sense, our task of legal argumentation is a more challenging domain. The negotiation domain also has shorter utterances, shorter dialogues, and more concrete, and easy-to-understand rewards (e.g., price is a numerical reward). On the contrary, courtroom conversations have much longer dialogues (on average more than 100 dialogue turns) with much longer utterances (on average 200 words per utterance) and more abstract rewards (e.g., relevance and sentiment score). It

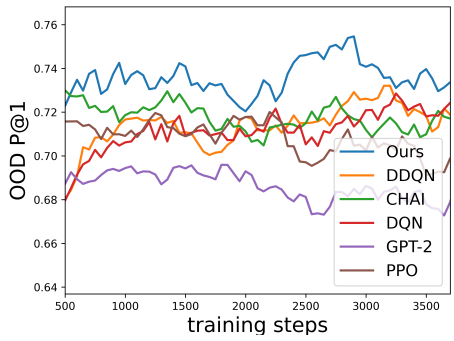

Figure 3: Main Results: Effectiveness OOD P@1.

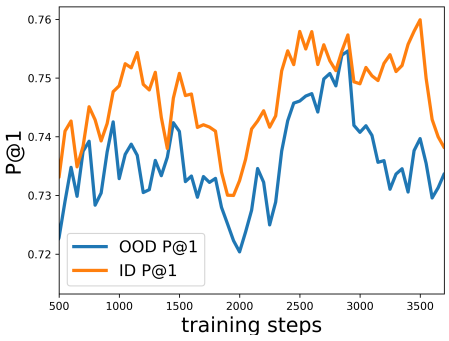

Figure 4: Proposed Method's ID P@1 vs. OOD P@1.

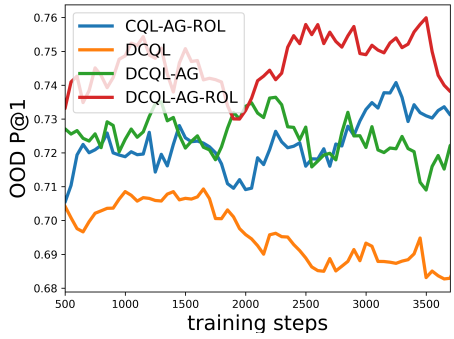

Figure 5: Ablation Study: OOD P@1.

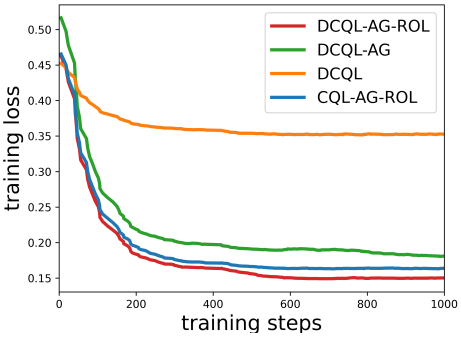

Figure 6: Ablation Study: Training Loss.

is encouraging that our model is able to better handle a more challenging conversation domain.

Figure 4 further examines how well the proposed method, handles the OOD actions. We observe that there is still a gap between ID and OOD performance. This is reasonable because due to distribution shifts, OOD performance can hardly be better than ID performance. However, around training step 3,000, OOD P@1 almost reaches ID P@1, which is encouraging. As the training steps increase, both P@1 scores drop after step 3,500; which may be caused by overfitting of the Q-networks.

5.2.2 Ablation Study. We conduct an ablation study to examine the effectiveness of different components in the proposed method. A few variations of our agent are tested. They are 1) DCQL-AG-ROL (the full model), 2) DCQL-AG (the full model without reward-on-the-line), 3) CQL-AG-ROL (the full model without Double Q-learning), and 4) DCQL (the full model without action generation and reward-on-the-line).

Table 3 and Figure 5 report the results. They show that the full DCQL-AG-ROL model is the best-performing model. Moreover, the reward-on-the-line (ROL) function is a component that adds the most improvement for methods using language models. The action generation with language models is crucial to enrich the utterance/action space and create human-like and relevant conversations. Figure 6 looks into the training loss in the training process and confirms the above observations.

5.3 Case Studies

Table 5 enumerates example utterances generated by our agent, PPO, and CHAI and the Justice's original utterances from the transcript. We can see that CHAI performs worse than ours. Although CHAI is fluent in grammar and logic, its response may not have anything meaningful value. This happens because, without ROL, CHAI's evaluation of the OOD actions can be problematic.

PPO seems to perform the worst. Sometimes, it even replies with totally unrelated utterances. We think the main cause is that the PPO is an online method and usually relies on live rewards given by human annotators (as in ChatGPT). Being offline greatly hurts its performance.

Our agent works well and can catch the same point as real justice, with fluent, coherent grammar. However, it can still generate redundant and/or meaningless utterances at times. E.g., our agent says 'If you're not going to take any steps ... you're going to have to take some steps'. It may be because the language model has not been adequately trained with courtroom cases. Given a conversation domain that possesses hundreds of rounds of dialogues with professional knowledge

Table 5: Example Virtual Justice Responses by Different Agents

Question: *Do the Department of Health and Human Services' guidelines satisfy the Religious Freedom Restoration Act's demanding test for overriding religious objections?*
...
Attorney: I think it would, because I think it would be very difficult to administer that kind of system if either you couldn't even know about the objection or you couldn't take any steps on the government's part to fill the spot. But I think what's critical
Justice: Well, then, isn't that the same thing here? If you don't know who can pay, or who – who's not eligible, or who's eligible to pay, how does this system work?
DCQL-AG-ROL: Well, I don't think it makes a difference. I mean, if you're not going to take any steps on the government's part to fill the spot, you're going to have to take some steps.
PPO: Well, I mean, I guess what I'm asking is, why would it survive?
CHAI: I'm not – I'm not sure I understand your answer.
Attorney: Well, two things, Your Honor. One, this is perhaps the unique government program where the government can provide an exemption without actually requiring somebody to opt out, because that's exactly what they do for the churches, for the integrated auxiliaries, and the religious orders
...
Attorney: Not quite, Your Honor. I think that the problem is that the government has to draw a definition that is coherent and that is rational. And I think the problem here is they've drawn a definition from the tax regime that doesn't comply when you carry it over to this regime. In the tax world, when the churches, when the universities, when the Little Sisters of the Poor file that informational tax return, they actually get the exact same exemption.
Justice: It's going to be very difficult for this Court to write an opinion which says that once you have a church organization, you have to treat a religious university the same. I just find that very difficult to write.
DCQL-AG-ROL: Well, I guess what I'm saying is, I don't know that that's responsive to Justice Ginsburg's question. I mean, it seems to me that you're saying that the government has to draw a line and that the line has to be between the churches and the Little Sisters of the Poor.
PPO: Well, I guess what I'm saying is...the Little Sisters of the Poor.
CHAI: Well, I guess what I'm saying is...the Little Sisters of the Poor.
Attorney: And, Your Honor, we're not suggesting that. What we are suggesting is that when the government has the same interest here that it has for all of the other employees in this country that don't get coverage from a – from an employer-based plan – and it's not just the religious employers.
...

and complex argumentation structures, it is likely to require more training data and training episodes for a well-performing language model than other chitchat tasks.

6 CONCLUSION

In this paper, we propose a novel offline RL method for conversational agents. Our approach leverages a few techniques to remedy the extrapolation error in offline RL. They include conservative Q-learning, double Q-learning, action generation with fine-tuned domain-specific language models, and OOD reward generation. The novel reward generation method, reward-on-the-line, helps reduce the influence of extrapolation error. Our model demonstrates a more stable performance and superior results when distribution shifts happen. The strong experimental results show that our method is highly effective for offline RL.

As an offline RL method, our approach does not require human labeling, which enables better usage of existing conversation recordings and transcript logs. We have shown how to work on a publicly available dataset, with little preprocessing and domain knowledge required – we only used a handful of implicit reward functions, e.g., on-topic relevance and sentiment, and achieved strong performance. In the future, we plan to explore model-based RL to learn and model the environmental dynamics, such as legal oral argument patterns. It will also be worthwhile to compare our reward generation method with traditional inverse RL methods for reward learning.

Our method promotes the construction of standalone, special-purpose conversational agents by making the process much easier with access to plenty of training datasets and no labeling requirements. These agents can be valuable alternatives to super-sized, know-it-all agents such as ChatGPT, offering unique opportunities for research and for addressing data privacy and information security problems.

REFERENCES

[1] Mohammad Aliannejadi, Hamed Zamani, Fabio Crestani, and W. Bruce Croft. 2019. Asking Clarifying Questions in Open-Domain Information-Seeking Conversations. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval* (Paris, France) (*SIGIR'19*). Association for Computing Machinery, New York, NY, USA, 475–484. https://doi.org/10.1145/3331184.3331265
[2] Layla El Asri, Jing He, and Kaheer Suleman. 2016. A Sequence-to-Sequence Model for User Simulation in Spoken Dialogue Systems. https://doi.org/10.48550/ARXIV.1607.00070
[3] Christina Baek, Yiding Jiang, Aditi Raghunathan, and Zico Kolter. 2022. Agreement-on-the-Line: Predicting the Performance of Neural Networks under Distribution Shift. https://doi.org/10.48550/ARXIV.2206.13089
[4] Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018. MultiWOZ – A Large-Scale Multi-Domain Wizard-of-Oz Dataset for Task-Oriented Dialogue Modelling. https://doi.org/10.48550/ARXIV.1810.00278
[5] Giovanni Campagna, Agata Foryciarz, Mehrad Moradshahi, and Monica Lam. 2020. Zero-Shot Transfer Learning with Synthesized Data for Multi-Domain Dialogue State Tracking. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, 122–132. https://doi.org/10.18653/v1/2020.acl-main.12

[6] Chieh-Yang Chen, Pei-Hsin Wang, Shih-Chieh Chang, Da-Cheng Juan, Wei Wei, and Jia-Yu Pan. 2020. AirConcierge: Generating Task-Oriented Dialogue via Efficient Large-Scale Knowledge Retrieval. In *Findings of the Association for Computational Linguistics: EMNLP 2020*. Association for Computational Linguistics, Online, 884–897. https://doi.org/10.18653/v1/2020.findings-emnlp.79

[7] Bruce Croft, Donald Metzler, and Trevor Strohman. 2009. *Search Engines: Information Retrieval in Practice* (1st ed.). Addison-Wesley Publishing Company, USA.

[8] Prafulla Dhariwal, Christopher Hesse, Oleg Klimov, Alex Nichol, Matthias Plappert, Alec Radford, John Schulman, Szymon Sidor, Yuhuai Wu, and Peter Zhokhov. 2017. OpenAI Baselines. https://github.com/openai/baselines.

[9] Emily Dinan, Samuel Humeau, Bharath Chintagunta, and Jason Weston. 2019. Build it Break it Fix it for Dialogue Safety: Robustness from Adversarial Human Attack. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, Hong Kong, China, 4537–4546. https://doi.org/10.18653/v1/D19-1461

[10] Damien Ernst, Pierre Geurts, and Louis Wehenkel. 2005. Tree-Based Batch Mode Reinforcement Learning. *Journal of Machine Learning Research* 6, 18 (2005), 503–556. http://jmlr.org/papers/v6/ernst05a.html

[11] Tom Everitt, Victoria Krakovna, Laurent Orseau, and Shane Legg. 2017. Reinforcement Learning with a Corrupted Reward Channel. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence* (Melbourne, Australia) *(IJCAI'17)*. AAAI Press, 4705–4713.

[12] Scott Fujimoto, David Meger, and Doina Precup. 2018. Off-Policy Deep Reinforcement Learning without Exploration. In *International Conference on Machine Learning*.

[13] Jianfeng Gao, Michel Galley, and Lihong Li. 2018. Neural Approaches to Conversational AI. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*. Association for Computational Linguistics, Melbourne, Australia, 2–7. https://doi.org/10.18653/v1/P18-5002

[14] He He, Derek Chen, Anusha Balakrishnan, and Percy Liang. 2018. Decoupling Strategy and Generation in Negotiation Dialogues. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Brussels, Belgium, 2333–2343. https://doi.org/10.18653/v1/D18-1256

[15] Ehsan Hosseini-Asl, Bryan McCann, Chien-Sheng Wu, Semih Yavuz, and Richard Socher. 2020. A Simple Language Model for Task-Oriented Dialogue. In *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (Eds.), Vol. 33. Curran Associates, Inc., 20179–20191. https://proceedings.neurips.cc/paper/2020/file/e946209592563be0f01c844ab2170f0c-Paper.pdf

[16] Clayton Hutto and Eric Gilbert. 2014. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Proceedings of the international AAAI conference on web and social media*, Vol. 8. 216–225.

[17] Natasha Jaques, Asma Ghandeharioun, Judy Hanwen Shen, Craig Ferguson, Agata Lapedriza, Noah Jones, Shixiang Gu, and Rosalind Picard. 2019. Way Off-Policy Batch Deep Reinforcement Learning of Implicit Human Preferences in Dialog. https://doi.org/10.48550/ARXIV.1907.00456

[18] Zhengyao Jiang, Dixing Xu, and Jinjun Liang. 2017. A deep reinforcement learning framework for the financial portfolio management problem. *arXiv preprint arXiv:1706.10059* (2017).

[19] Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, and Sergey Levine. 2018. QT-Opt: Scalable Deep Reinforcement Learning for Vision-Based Robotic Manipulation. https://doi.org/10.48550/ARXIV.1806.10293

[20] Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. 2020. Conservative Q-Learning for Offline Reinforcement Learning. *CoRR* abs/2006.04779 (2020). arXiv:2006.04779 https://arxiv.org/abs/2006.04779

[21] Sascha Lange, Thomas Gabel, and Martin Riedmiller. 2012. *Batch Reinforcement Learning*. Springer Berlin Heidelberg, Berlin, Heidelberg, 45–73. https://doi.org/10.1007/978-3-642-27645-3_2

[22] Kimin Lee, Michael Laskin, Aravind Srinivas, and Pieter Abbeel. 2021. SUNRISE: A Simple Unified Framework for Ensemble Learning in Deep Reinforcement Learning. In *Proceedings of the 38th International Conference on Machine Learning* (*Proceedings of Machine Learning Research, Vol. 139*), Marina Meila and Tong Zhang (Eds.). PMLR, 6131–6141. https://proceedings.mlr.press/v139/lee21g.html

[23] Sergey Levine, Aviral Kumar, G. Tucker, and Justin Fu. 2020. Offline Reinforcement Learning: Tutorial, Review, and Perspectives on Open Problems. *ArXiv* abs/2005.01643 (2020).

[24] Jiwei Li, Will Monroe, Alan Ritter, Michel Galley, Jianfeng Gao, and Dan Jurafsky. 2016. Deep reinforcement learning for dialogue generation. *arXiv preprint arXiv:1606.01541* (2016).

[25] Jiafei Lyu, Xiaoteng Ma, Xiu Li, and Zongqing Lu. 2022. Mildly Conservative Q-Learning for Offline Reinforcement Learning. In *Advances in Neural Information Processing Systems*, Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (Eds.). https://openreview.net/forum?id=VYYf6S67pQc

[26] Andrea Madotto, Samuel Cahyawijaya, Genta Indra Winata, Yan Xu, Zihan Liu, Zhaojiang Lin, and Pascale Fung. 2020. Learning Knowledge Bases with Parameters for Task-Oriented Dialogue Systems. In *Findings of the Association for Computational Linguistics: EMNLP 2020*. Association for Computational Linguistics, Online, 2372–2394. https://doi.org/10.18653/v1/2020.findings-emnlp.215

[27] Andrea Madotto, Chien-Sheng Wu, and Pascale Fung. 2018. Mem2Seq: Effectively Incorporating Knowledge Bases into End-to-End Task-Oriented Dialog Systems. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics* (*Volume 1: Long Papers*). Association for Computational Linguistics, Melbourne, Australia, 1468–1478. https://doi.org/10.18653/v1/P18-1136

[28] Brida V Mbuwir, Frederik Ruelens, Fred Spiessens, and Geert Deconinck. 2017. Battery energy management in a microgrid using batch reinforcement learning. *Energies* 10, 11 (2017), 1846.

[29] John Miller, Rohan Taori, Aditi Raghunathan, Shiori Sagawa, Pang Wei Koh, Vaishaal Shankar, Percy Liang, Yair Carmon, and Ludwig Schmidt. 2021. Accuracy on the Line: On the Strong Correlation Between Out-of-Distribution and In-Distribution Generalization. https://doi.org/10.48550/ARXIV.2107.04649

[30] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. 2013. Playing Atari with Deep Reinforcement Learning. (12 2013).

[31] Seungwhan Moon, Pararth Shah, Anuj Kumar, and Rajen Subba. 2019. OpenDialKG: Explainable Conversational Reasoning with Attention-based Walks over Knowledge Graphs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 845–854. https://doi.org/10.18653/v1/P19-1081

[32] Ashvin Nair, Bob McGrew, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. 2017. Overcoming Exploration in Reinforcement Learning with Demonstrations. https://doi.org/10.48550/ARXIV.1709.10089

[33] OpenAI. 2021. ChatGPT: Optimizing Language Models for Dialogue. https://openai.com/blog/chatgpt/

[34] Kishan Panaganti, Zaiyan Xu, Dileep Kalathil, and Mohammad Ghavamzadeh. 2022. Robust Reinforcement Learning using Offline Data. (08 2022). https://doi.org/10.48550/arXiv.2208.05129

[35] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.

[36] Robi Polikar. 2012. *Ensemble Learning*. Springer US, Boston, MA, 1–34. https://doi.org/10.1007/978-1-4419-9326-7_1

[37] Chen Qu, Liu Yang, Cen Chen, Minghui Qiu, W. Bruce Croft, and Mohit Iyyer. 2020. Open-Retrieval Conversational Question Answering. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (Virtual Event, China) *(SIGIR '20)*. Association for Computing Machinery, New York, NY, USA, 539–548. https://doi.org/10.1145/3397271.3401110

[38] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog* 1, 8 (2019), 9.

[39] Rafael Rafailov, Tianhe Yu, Aravind Rajeswaran, and Chelsea Finn. 2021. Offline Reinforcement Learning from Images with Latent Space Models. In *Proceedings of the 3rd Conference on Learning for Dynamics and Control* (*Proceedings of Machine Learning Research, Vol. 144*), Ali Jadbabaie, John Lygeros, George J. Pappas, Pablo A. Parrilo, Benjamin Recht, Claire J. Tomlin, and Melanie N. Zeilinger (Eds.). PMLR, 1154–1168. https://proceedings.mlr.press/v144/rafailov21a.html

[40] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. 2021. Zero-Shot Text-to-Image Generation. *CoRR* abs/2102.12092 (2021). arXiv:2102.12092 https://arxiv.org/abs/2102.12092

[41] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. https://doi.org/10.48550/ARXIV.1908.10084

[42] Martin Riedmiller. 2005. Neural Fitted Q Iteration – First Experiences with a Data Efficient Neural Reinforcement Learning Method. In *Machine Learning:*

ECML 2005, João Gama, Rui Camacho, Pavel B. Brazdil, Alípio Mário Jorge, and Luís Torgo (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 317–328.

[43] George AF Seber and Alan J Lee. 2003. *Linear regression analysis*. Vol. 330. John Wiley & Sons.

[44] Susan M Shortreed, Eric Laber, Daniel J Lizotte, T Scott Stroup, Joelle Pineau, and Susan A Murphy. 2011. Informing sequential clinical decision-making through reinforcement learning: an empirical study. *Machine learning* 84, 1 (2011), 109–136.

[45] Samarth Sinha, Ajay Mandlekar, and Animesh Garg. 2022. S4RL: Surprisingly Simple Self-Supervision for Offline Reinforcement Learning in Robotics. In *Proceedings of the 5th Conference on Robot Learning* (*Proceedings of Machine Learning Research, Vol. 164*), Aleksandra Faust, David Hsu, and Gerhard Neumann (Eds.). PMLR, 907–917. https://proceedings.mlr.press/v164/sinha22a.html

[46] Pei-Hao Su, David Vandyke, Milica Gaši´c, Dongho Kim, Nikola Mrkši´c, Tsung Hsien Wen, and Steve Young. 2015. Learning from Real Users: Rating Dialogue Success with Neural Networks for Reinforcement Learning in Spoken Dialogue Systems. https://doi.org/10.21437/Interspeech.2015-456

[47] Zhiwen Tang, Hrishikesh Kulkarni, and Grace Hui Yang. 2021. High-Quality Diversification for Task-Oriented Dialogue Systems. https://doi.org/10.48550/ARXIV.2106.00891

[48] David R. Traum. 1996. Book Reviews: Spoken Natural Language Dialogue Systems: A Practical Approach. *Computational Linguistics* 22, 3 (1996). https://aclanthology.org/J96-3008

[49] Hado van Hasselt, Arthur Guez, and David Silver. 2016. Deep Reinforcement Learning with Double Q-Learning. *Proceedings of the AAAI Conference on Artificial Intelligence* 30, 1 (Mar. 2016). https://doi.org/10.1609/aaai.v30i1.10295

[50] Siddharth Verma, Justin Fu, Mengjiao Yang, and Sergey Levine. 2022. CHAI: A CHatbot AI for Task-Oriented Dialogue with Offline Reinforcement Learning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics* (*NAACL*).

[51] Oriol Vinyals and Quoc Le. 2015. A Neural Conversational Model. https://doi.org/10.48550/ARXIV.1506.05869

[52] Jianhong Wang, Yuan Zhang, Tae-Kyun Kim, and Yunjie Gu. 2020. Modelling Hierarchical Structure between Dialogue Policy and Natural Language Generator with Option Framework for Task-oriented Dialogue System. *ArXiv* abs/2006.06814 (2020).

[53] Zhenduo Wang and Qingyao Ai. 2021. Controlling the Risk of Conversational Search via Reinforcement Learning. https://doi.org/10.48550/ARXIV.2101.06327

[54] Marco A. Wiering and Hado van Hasselt. 2008. Ensemble Algorithms in Reinforcement Learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B* (*Cybernetics*) 38, 4 (2008), 930–936. https://doi.org/10.1109/TSMCB.2008.920231

[55] Haoran Xu, Li Jiang, Jianxiong Li, and Xianyuan Zhan. 2022. A Policy-Guided Imitation Approach for Offline Reinforcement Learning. In *Advances in Neural Information Processing Systems*, Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (Eds.). https://openreview.net/forum?id=CKbqDtZnSc

[56] Haoran Xu, Xianyuan Zhan, and Xiangyu Zhu. 2022. Constraints Penalized Q-learning for Safe Offline Reinforcement Learning. *Proceedings of the AAAI Conference on Artificial Intelligence* 36, 8 (Jun. 2022), 8753–8760. https://doi.org/10.1609/aaai.v36i8.20855

[57] Grace Hui Yang, Zhiwen Tang, and Ian Soboroff. 2017. TREC '17 Dynamic Domain Track Overview. In *TREC '17*.

[58] Steve Young, Milica Gašić, Blaise Thomson, and Jason D. Williams. 2013. POMDP-Based Statistical Spoken Dialog Systems: A Review. *Proc. IEEE* 101, 5 (2013), 1160–1179. https://doi.org/10.1109/JPROC.2012.2225812

[59] Tom Young, Erik Cambria, Iti Chaturvedi, Minlie Huang, Hao Zhou, and Subham Biswas. 2017. Augmenting End-to-End Dialog Systems with Commonsense Knowledge. https://doi.org/10.48550/ARXIV.1709.05453

[60] Shi Yu, Jiahua Liu, Jingqin Yang, Chenyan Xiong, Paul Bennett, Jianfeng Gao, and Zhiyuan Liu. 2020. Few-Shot Generative Conversational Query Rewriting. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (Virtual Event, China) (*SIGIR '20*). Association for Computing Machinery, New York, NY, USA, 1933–1936. https://doi.org/10.1145/3397271.3401323

[61] Wenhao Zhan, Baihe Huang, Audrey Huang, Nan Jiang, and Jason Lee. 2022. Offline Reinforcement Learning with Realizability and Single-policy Concentrability. In *Proceedings of Thirty Fifth Conference on Learning Theory* (*Proceedings of Machine Learning Research, Vol. 178*), Po-Ling Loh and Maxim Raginsky (Eds.). PMLR, 2730–2775. https://proceedings.mlr.press/v178/zhan22a.html

[62] Yichi Zhang, Zhijian Ou, and Zhou Yu. 2019. Task-Oriented Dialog Systems that Consider Multiple Appropriate Responses under the Same Context. https://doi.org/10.48550/ARXIV.1911.10484

[63] Tiancheng Zhao, Kaige Xie, and Maxine Eskenazi. 2019. Rethinking action spaces for reinforcement learning in end-to-end dialog agents with latent variable models. *arXiv preprint arXiv:1902.08858* (2019).

[64] Tong Zhou, Haihua Zhu, Dunbing Tang, Changchun Liu, Qixiang Cai, Wei Shi, and Yong Gui. 2022. Reinforcement learning for online optimization of job-shop scheduling in a smart manufacturing factory. *Advances in Mechanical Engineering* 14, 3 (2022), 16878132221086120. https://doi.org/10.1177/16878132221086120 arXiv:https://doi.org/10.1177/16878132221086120