

---

# Way Off-Policy Batch Deep Reinforcement Learning of Implicit Human Preferences in Dialog

---

Natasha Jaques, Asma Ghandeharioun, Judy Hanwen Shen, Craig Ferguson,  
Agata Lapedriza, Noah Jones, Shixiang Gu, Rosalind Picard

Department of Media Arts and Science  
Massachusetts Institute of Technology  
Cambridge, MA 02139  
jaquesn@mit.edu

## Abstract

Most deep reinforcement learning (RL) systems are not able to learn effectively from off-policy data, especially if they cannot explore online in the environment. This is a critical shortcoming for applying RL to real-world problems where collecting data is expensive, and models must be tested offline before being deployed to interact with the environment – *e.g.* systems that learn from human interaction. Thus, we develop a novel class of off-policy batch RL algorithms which use KL-control to penalize divergence from a pre-trained prior model of probable actions. This KL-constraint reduces extrapolation error, enabling effective offline learning, without exploration, from a fixed batch of data. We also use dropout-based uncertainty estimates to lower bound the target Q-values as a more efficient alternative to Double Q-Learning. This Way Off-Policy (WOP) algorithm on the problem of open-domain dialog generation; a challenging reinforcement learning problem with a 20,000 dimensional action space. WOP allows for the extraction of multiple different reward functions post-hoc from collected human interaction data, and can learn effectively from all of these. We test real-world generalization by deploying dialog models live to converse with humans in an open-domain setting, and demonstrate that WOP achieves significant improvements over state-of-the-art prior methods in batch deep RL.

## 1 Introduction

In order to scale deep reinforcement learning (RL) to safety-critical, real-world domains, two abilities are needed. First, since collecting real-world interaction data can be expensive and time-consuming, algorithms must be able to learn from off-policy data no matter how it was generated, or how little correlation between the data distribution and the current policy. Second, it is often necessary to carefully test a policy before deploying it to the real world; for example, to ensure its behavior is safe and appropriate for humans. Thus, the algorithm must be able to learn offline first, from a static batch of data, without the ability to explore.

This off-policy, *batch reinforcement learning* (BRL) setting represents a challenging RL problem. Most deep RL algorithms fail to learn from data that is not heavily correlated with the current policy [14]. Even models based on off-policy algorithms like *Q*-learning fail to learn in the offline, batch setting, when the model is not able to explore. If the batch data is not sufficient to cover the state-action space, BRL models can suffer from *extrapolation error*, learning unrealistic value estimates of state-action pairs not contained in the batch [14]. It can be impossible to correct for extrapolation error when there is a mismatch in the distribution of state-actions pairs in the batch data, and the distribution induced by the learned policy. For example, if the policy learns to select

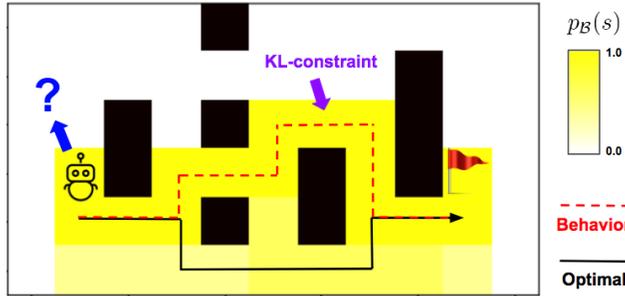


Figure 1: In this example batch RL problem, the robot’s goal is to travel the minimum distance around the black walls to get to the red flag. A trained behavior policy generated the batch data; the probability of each of the states appearing in the batch,  $p_B(s)$ , is in yellow (white locations are not contained in the batch). If the offline RL policy estimates the value of going *up* or *left* from the start position is high, it will have no way to refine this estimate using the batch data, or learn a good policy in this region of state space. The KL-constraint ensures that the RL policy will stay within the support of the batch data. However, the behavior policy is suboptimal, so using behavior cloning to directly imitate the batch data will result in suboptimal return. Instead, the KL-constrained model can learn to find the optimal policy, which is within the support of the batch.

actions which are not contained in the batch, it cannot learn a reasonable value function for those actions. Figure 1 illustrates this concept, where the batch only covers a subset of possible policies. Extrapolation error is particularly problematic in high-dimensional state and action spaces such as those inherent in language generation.

We propose to resolve these issues by leveraging a pre-trained language model. While training with RL, we penalize divergence from this prior model with different forms of KL-control. This technique ensures that the RL model learns a policy that stays close the state-action distribution of the batch, combating extrapolation error. We also propose using dropout to obtain uncertainty estimates of the target  $Q$ -values, and use this lower bound to alleviate overestimation bias. We benchmark against a discrete adaptation of Batch Constrained  $Q$ -learning (BCQ) [14], a recently proposed state-of-the-art BRL algorithm for continuous domains, and show that our Way Off-Policy algorithm achieves superior performance.

We apply these algorithms to a challenging, under-explored, real-world reinforcement learning problem: using implicitly expressed human reactions in chat to improve open-domain dialog systems. When a machine learning system interacts with humans, ideally we would like to learn about the humans’ preferences in order to improve its performance. Yet having humans manually indicate their preferences through explicit means like pressing a button (e.g. [7]) or submitting a feedback report, does not scale. Instead, we would like to be able to use humans’ implicit reactions, such as the sentiment they express, or the length of the conversation, in order to improve the policy. However, applying off-policy batch RL to language generation is challenging because the number of potential combinations of words and sentences leads to a combinatorial explosion in the size of the state space. The action space – the set of frequent vocabulary words in the English language – is 20,000-dimensional. This compounds extrapolation error, making BRL even more difficult. However, when learning from human interactions in the wild, it is crucial to be able to learn offline and test the policy before deploying it, lest it learn inappropriate behaviors (e.g. [22]).

To support this work, we developed an interactive online platform that allows humans to chat with deep neural network dialog models running on a GPU; the BRL models trained for this study are available live at <https://neural.chat/rl/>. Through this platform we collected human responses to a set of over 40 different dialog models over the course of several months. Using our Way Off-Policy algorithm, we are able to effectively learn from this batch of data, in spite of the fact that it was generated with a vastly different set of model architectures, which were trained on different datasets. Further, we use the batch to learn from many different reward functions designed post-hoc to extract implicit human preferences, something that is only possible with effective off-policy BRL.

In summary, the contributions of this paper are:

- A novel algorithm, Way Off-Policy learning, which is the first to propose using KL-control from a pre-trained prior model as a way to reduce extrapolation error in batch RL.

- Experiments showing the effectiveness of WOP for dialog generation above strong baselines based on prior work [14].
- A set of novel conversation rewards based on how human preferences are implicitly expressed in text. We are the first work to learn from implicit signals in conversation offline using batch RL.

## 2 Related Work

The approach we propose is based on KL-control, a branch of stochastic optimal control (SOC) [52] where the Kullback-Leibler (KL) divergence from some distribution is used to regularize an RL policy (e.g. [1, 28, 43, 55]). Well-known examples include Trust Region Policy Optimization (TRPO) [45], and use conservative, KL-regularized policy updates to restrict the RL algorithm to stay close to its own prior policy (e.g. [19, 27, 41, 43]). KL-control can also be applied to entropy maximization (e.g. [59, 39, 18, 12]). KL-control has also been used to improve transfer learning between maximum likelihood estimation (MLE) training on data, and training with RL [24]. To the best of our knowledge, our work is the first to propose KL-control to improve offline batch RL.

Other strategies to improve off-policy learning have been proposed, but differ from this work in key respect, either focusing on scenarios where the policy is able to explore and collect more data (e.g. [9, 44]); such as learning online from an outdated replay buffer (e.g. [38]), or performing off-policy policy evaluation (e.g. [10, 25, 42, 54]). In contrast, we learn a policy entirely offline, from a fixed batch of data, without the ability to explore. Other works that have tackled this problem using deep learning (e.g. [36, 17, 4, 29, 2, 14]) do not use KL-control. We propose using dropout to approximate model uncertainty of the target  $Q$ -network. Dropout can be used to estimate uncertainty in neural networks [15]. Different forms of uncertainty estimates have been used in RL (e.g. [26, 40]); for example, Bayesian uncertainty estimates have been proposed as an alternative to double DQN [3].

### 2.1 RL for language generation

Improving dialog systems with RL has largely been restricted to task-oriented dialog systems, which have a limited number of task-specific actions (e.g. [11, 16, 34, 35, 53]). These approaches may incorporate human input, usually through explicit, manual feedback (e.g. [48]), but sometimes with more implicit signals, such as the user interrupting the system or starting over [49]. Efforts to expand RL to the open-domain dialog setting, such as [31, 32, 33], are less numerous, and do not involve learning from human feedback. Even in the open-domain setting, authors may choose to use a highly restricted action space; for example, using RL to choose which dialog model to invoke [46].

Since the posting of the preprint of this paper, Ziegler and colleagues have used explicit human feedback to improve the summarization and text continuation performance of a large-scale language model [60]. Although they do not study dialog, batch RL, or implicit feedback (instead learning online from a trained model of explicit human feedback), they do make use of our proposal to penalize KL-divergence from a pre-trained language model, and find that this is important to achieving good performance. Although implicit signals such as sentiment [20] and conversation length [58] have been used in MLE systems, the idea of using such signals as a reward for RL is relatively unexplored. Shin and colleagues use on-policy learning in conjunction with a user-sentiment approximator to improve a seq2seq model [50], but are unable to learn directly from user feedback. To the best of our knowledge, we are the first to use batch RL to train open-domain dialog models on implicit cues gained from real human interactions.

## 3 Methods

We employ typical RL notation in which  $s_t$  represents the environment state at time  $t$ , the agent takes action  $a_t$  according to its policy  $\pi(a_t|s_t)$ , and receives a reward  $r(s_t, a_t)$ . The agent’s goal is to maximize reward over an episode trajectory  $\tau$ , with a discount factor of  $\gamma$  applied to future rewards.  $Q$ -learning methods learn an action-value estimate of the total expected discounted future reward,  $Q_\pi(a_t, s_t) = \mathbb{E}_\pi[\sum_{t'=t}^T \gamma^{t'-t} r(s_{t'}, a_{t'})]$ , through iterative updates based on the Bellman equation:

$$Q_{\theta_\pi}(s_t, a_t) = r(s_t, a_t) + \gamma \mathbb{E}_{s_{t+1} \sim p(\cdot|s_t, a_t)} [\max_{a_{t+1}} Q_{\theta_\pi}(s_{t+1}, a_{t+1})] \quad (1)$$

In deep  $Q$ -learning [37], a  $Q$ -network approximates  $Q_{\theta_\pi}(s_t, a_t)$  and drives the policy  $\pi$ . A second target  $Q$ -network approximates the expected reward from the next state,  $Q_{\theta_T}(s_{t+1}, a_{t+1})$  [56].

### 3.1 Batch RL and extrapolation error

In batch RL, we are given a fixed batch of data  $\mathcal{B}$ , and assume that no further interaction with the environment is possible. To train  $Q_{\theta_\pi}$ , we sample  $(s_t, a_t, r_t, s_{t+1}) \sim \mathcal{B}$ , and update the weights of the  $Q$ -network to approximate Eq. 1. Because  $Q$ -learning is an off-policy algorithm, in principle it should be able to learn from data collected by any behavior policy. However, extrapolation error can occur if the BRL policy learns to favour a state-action pair  $(s, a)$  that is unlikely, or not contained, in the batch data. In this case, the estimate  $Q(s', \pi(s'))$  can be arbitrarily bad [14]. Such errors can then accumulate through the Bellman backup operator [29]. Experiments from [14] show that extrapolation error can be highly detrimental to learning off-policy in BRL.

These problems are compounded by the fact that algorithms based on the Bellman operator are inherently optimistic in the face of uncertainty. When value estimates for some region of the state-action space are noisy (because too few experience samples have been used to refine them), the maximum operation in Eq. 1 will lead to an overestimation of expected future reward. In a normal RL setting, this overestimation bias drives the model to explore areas of the state-action space for which the value estimates have the highest variance, thus enabling it to refine them; in essence, creating a built-in drive to explore. However, in a batch setting where exploration is not possible, the model is instead driven to value parts of the state-action space for which it has little to no data to learn a good policy (see Figure 1).

### 3.2 Dropout for uncertainty estimation of Target $Q$ -values

The overestimation of  $Q$ -values in the BRL setting necessitates other methods for estimating the future reward via the Target  $Q$ -network. Clipped Double  $Q$ -learning [13] maintains two independent pairs of  $Q$ -networks, and takes the minimum of their estimates of future reward. This approach is computationally expensive and memory intensive. Instead, we leverage the fact that a network trained with dropout can be used to approximate a Bayesian uncertainty estimate of the output value [15]. Given the target  $Q$ -network  $Q_{\theta_T}$ , we compute  $Q(a_{t+1}, s_{t+1})$  using a Monte Carlo (MC) estimate of the lower-bound of  $Q_{\theta_T}(a_{t+1}, s_{t+1})$  by running  $M$  stochastic forward passes of the network, each with a new dropout mask  $d_i \sim q^W$ :

$$Q(a_{t+1}, s_{t+1}) = \min_{i=1 \dots M} [Q_{\theta_T}(a_{t+1}, s_{t+1}; d_i)] \quad (2)$$

Using the minimum operator penalizes high variance estimates and leads the algorithm to be pessimistic in the face of uncertainty, rather than optimistic. Such a bias will push the model to favour actions that lead to states well covered by the batch data.

### 3.3 Discrete Batch Constrained $Q$

Batch Constrained  $Q$ -learning (BCQ) [14] proposes to address the BRL problem by constraining the actions of the  $Q$ -network to be close to the data contained within the batch. This is accomplished by learning a generative model of the batch,  $G_w = p(a|s)$ , and sampling from this model during learning and inference. Because BCQ is designed for continuous action domains, it applies a learned perturbation model  $\xi(s, a; \Phi)$  which is allowed to alter the action within the range  $[-\Phi, \Phi]$ . BCQ learns  $Q$ -estimates that incorporate the perturbation model,  $Q_\theta(s, a + \xi(s, a; \Phi))$ . To act,  $n$  possible actions are sampled from the generative model,  $\{a_i \sim G_w(s)\}_{i=1}^n$ , perturbed, and the action with the maximum  $Q$ -value is selected, giving the BCQ policy:

$$\pi_{BCQ}(s) = \arg \max_{a_i + \xi(s, a_i; \Phi)} Q_\theta(s, a_i + \xi(s, a_i; \Phi)) \quad (3)$$

We focus on the scenario where a model of  $p(a|s)$  can be obtained through MLE training on data of known action sequences. This prior model provides a more robust estimate of  $p(a|s)$  than one learned from the batch data, assuming the size of the batch is small relative to unsupervised data related to the problem (i.e. when the batch comes from human interaction data). We propose an adaptation of BCQ to discrete action spaces ( $DBCQ$ ) which leverages such a strong pre-trained prior model as an improved version of  $G_w$ . Since BCQ relies on Double Clipped  $Q$ -learning [13], here we

use dropout-based uncertainty estimates as in Eq. 2. Because the action space is discrete, we do not use a perturbation model to modify actions, but instead define the DBCQ policy as:

$$\pi_{DBCQ}(s) = \arg \max_{a_i \sim p(a|s)} Q_{\theta_\pi}(s, a_i) \quad (4)$$

### 3.4 KL Control from pre-trained prior

Rather than simply sample from the prior, we would like the  $Q$ -learning algorithm to directly incorporate the prior into the policy. Thus, we use KL-control to penalize divergence between the prior  $p(a|s)$ , and the  $Q$ -network policy  $\pi_\theta$ , while still maximizing reward. Given a trajectory of actions,  $\tau = \{a_1, a_2, \dots, a_{t-1}\}$ , let  $q(\tau) = \prod_{t=1}^T \pi_\theta(a_t, s_t)$  be the policy of our  $Q$ -learning algorithm at the trajectory level. Similarly, let  $p(\tau) = \prod_{t=1}^T p(a_t|s_t)$  be the prior distribution over the trajectory, and  $r(\tau)$  be the rewards. We seek to maximize the following KL-regularized objective:

$$L(q) = \mathbb{E}_{q(\tau)}[r(\tau)]/c - D_{KL}[q(\tau)||p(\tau)] \quad (5)$$

Since  $D_{KL}[q||p] = \sum_x q(x)(\log q(x) - \log p(x))$ , we can see that this is equivalent to maximizing the following expected value function of the policy  $\pi_\theta$  at the action level:

$$Q^\pi(s_t, a_t) = \mathbb{E}_\pi[\sum_{t'=t}^T r(s_{t'}, a_{t'})/c + \log p(a_{t'}|s_{t'}) - \log \pi(a_{t'}|s_{t'})] \quad (6)$$

The two terms we have introduced in Eq. 6 have clear motivations. The  $p(a|s)$  term rewards the model for choosing actions that have high probability under the prior, biasing the model to state-action pairs that are realistic, and likely to be in the batch. The  $-\log \pi(a|s)$  term is analogous to entropy regularization. Maintaining diversity in the action space through entropy regularization is important for generative models like dialog systems, which are known to collapse to an uninteresting, small number of repeated samples [30]. Re-stating Eq. 6 as an entropy-regularized  $Q$ -function, we obtain:

$$Q(s_t, a_t) = \mathbb{E}_\pi[\sum_{t'=t}^T r(s_{t'}, a_{t'})/c + \log p(a_{t'}|s_{t'}) + \mathcal{H}(\cdot|s_{t'})] \quad (7)$$

Motivated by energy-based models of the form  $\pi(a_t|s_t) \propto \exp(-\mathcal{E}(s_t, a_t))$ , one can derive a soft version of the entropy-regularized  $Q$ -function that uses a Boltzmann distribution to estimate future reward [18]. We refer to it as a  $\Psi$ -function following previous work [24], which derived this function as a generalization of the  $\Psi$ -learning proposed by [43]. The optimal  $\Psi$ -function and policy are:

$$\Psi^*(s_t, a_t) = r(s_t, a_t)/c + \log p(a_t|s_t) + \gamma \log \sum_{a'} \exp(\Psi^*(s', a')) \quad (8)$$

$$\pi_\Psi^*(a_t|s_t) = \exp(\Psi^*(s_t, a_t)) \quad (9)$$

Because it avoids taking a hard max over noisy estimates,  $\Psi$ -learning leads to less overestimation of future reward [1, 18]. This leads to more stable temporal-difference updates and aids learning. Thus, it may be especially useful in the BRL setting for reducing optimism in the face of uncertainty. The Way Off-Policy (WOP) algorithm combines Monte Carlo (MC) target estimation,  $\Psi$ -learning, and KL-control from a pre-trained prior.

## 4 RL for open-domain dialog generation

Here, we tackle the problem of training an open-domain dialog model from human feedback. We consider human interaction to represent the ‘environment’. The response of a human to the bot’s utterance is used to compute a reward signal to train the model. The state is the conversation history, composed of a series of conversation turns or utterances,  $u_{1\dots t}$ , where each utterance is composed of vocabulary tokens. The model attempts to construct a response utterance  $u_{t+1}^\pi = [a_1, a_2, \dots, a_n]$  by iteratively choosing an action  $a_i$  as the next token. Applying RL to dialog generation is challenging due to the large state-action space. The number of tokens in the vocabulary of our pre-trained model is 20,000, making the action space very high-dimensional; this further compounds the problem of extrapolation error.

We trained over 40 dialog models with different architectures (e.g. [47]), on different datasets, generating models that varied significantly in the distribution of language they learned. We deployed these models to users via a web server that hosts neural network dialog models on GPU for fast, real-time inference: <https://neural.chat>. The code for the models and the server is available in open-source at [https://github.com/natashamjaques/neural\\_chat](https://github.com/natashamjaques/neural_chat). Using the server, we collected a batch of human interaction data containing 14232 pairs of user input and agent response. Because learning language online from humans on the internet can result in inappropriate behavior (see [22]), learning offline using BRL is imperative.

The batch data was used to train the RL models as described in Section 3. Here, we use a pre-trained language model to estimate  $p(a|s)$ . We also initialize the weights of the  $Q$ -network and target  $Q$ -network are from the pre-trained model, to combat extrapolation error. The trained RL models were then re-deployed to the web. We recruited 90 Mechanical Turk workers to provide a total of 718 7-point Likert scale ratings of the bots’ quality, fluency, diversity, contingency (relatedness), and empathy, after interacting with each bot for at least 3 turns. Participants also had the option to provide explicit feedback through upvoting or downvoting a particular utterance within the interface. We sum these manual votes to create an overall *votes* score. We note that using this platform to test our models “in the wild” with humans represents a more meaningful test of generalization than testing an RL model in the same limited (game) environment in which it was trained, since humans are not restricted in the text they can type as input to the model.

#### 4.1 Learning from implicit human preferences

We seek to improve a dialog model’s ability to engage in natural conversation with a human by learning from the signals implicit in the way that the human responds. Rather than having the human manually label good performance – which we show in this work does not scale – the agent should recognize informative cues within the user’s responses, like sentiment, and the amount of time they spend chatting. Essentially, we want to create an agent that is intrinsically motivated to produce positive reactions in its human conversation partner. We design several intrinsic reward functions based on the rich, interactive content of conversation, taking inspiration from the psychology of human conversation: 1) eliciting positive sentiment and transitions from negative to positive sentiment, due to the importance of emotion to creating a sense of understanding [6, 57]; 2) eliciting longer conversations and more words typed, since this is a signal of engagement [51, 58]; 3) eliciting laughter (counting the number of ‘ha’s in the user response), because of its importance in building solidarity [21]; 4) high semantic similarity (close distance in sentence embedding space [8]) between the human input and agent response, because paraphrasing and style matching are important in facilitating good conversation [23, 57]; and 5) asking questions, since this is an important active listening skill [5]. The total reward given to the agent is a combination of these, with details (and coefficients) given in the supplementary material. Note that the first 4 types of rewards depend on eliciting positive responses from a human user; we call these the implicit *human reward*. The 5th reward is easily exploitable by the agent itself. These rewards represent only an initial foray into designing good metrics of human enjoyment, and further experimentation will be needed to improve them.

## 5 Results

To compare models, we not only look at human users’ ratings and votes, but also consider the automatic signals detectable from the text itself. This implicit *human reward* metric aggregates the measures listed in items 1-4 in Section 4.1, and measures the ability to elicit positive responses from the human. Table 1 shows the results of the human evaluation, comparing WOP to ablations of itself, Batch Q, and DBCQ<sup>1</sup>. MC Target  $Q$  estimation leads to modest improvements in *votes* and *human reward*, but does not improve ratings. Using  $\Psi$ -learning improves all three. However, the most notable difference in performance comes from KL-control. The KL-control models show substantial gains over the baseline models across both ratings and human reward. We perform a one-way analysis of variance (ANOVA) comparing the KL-control models to the Batch  $Q$  baselines and

<sup>1</sup>We also compare the RL models to the prior, and see performance improvements in terms of the elicited human reward, but not in terms of the quality ratings. We believe this is because the rewards proposed here do not fully cover what it means to have a high quality conversation (as we will elucidate later), and hope that other researchers be able to use the techniques we propose to learn from improved rewards.

DBCQ on the total human rating score, and find that the KL-control models are significantly better,  $F(x) = 4.781, p < .05$ . This validates the hypothesis that KL-control with a strong, pre-trained prior can be used to improve batch RL.

Table 1: Interactive human evaluation of batch RL techniques. KL-control models strongly outperform other techniques. Ratings are Likert scale, votes and human reward are  $z$ -scores.

Model type	Quality	Fluent	Diverse	Related	Empathy	Total	Votes	Human reward
DBCQ	1.64 ± .29	1.87 ± .34	<b>3.13 ± .58</b>	1.84 ± .34	2.09 ± .38	10.58 ± 1.55	-.228	-.050
Batch Q	1.87 ± .30	2.36 ± .42	2.20 ± .41	1.91 ± .32	2.58 ± .47	11.91 ± 1.58	-.163	-.005
Batch Q + MC	1.85 ± .39	2.46 ± .44	2.46 ± .52	1.98 ± .39	2.34 ± .49	11.07 ± 1.82	-.068	.005
KL-control Q	<b>2.38 ± .39</b>	3.24 ± .47	3.42 ± .54	<b>2.38 ± .45</b>	2.56 ± .43	13.98 ± 1.81	.016	.004
KL-control $\Psi$ (WOP)	2.33 ± .41	<b>3.73 ± .53</b>	2.82 ± .50	2.31 ± .44	<b>3.47 ± .50</b>	<b>14.67 ± 1.82</b>	<b>.128</b>	<b>.061</b>

As shown in Figure 2, without KL-regularization the baseline RL models diverge quickly and continuously from the prior, losing information about realistic sequences. This figure also helps explain the poor performance of DBCQ in Table 1. The underlying  $Q$ -network in DBCQ does not directly integrate the prior. As  $Q$ -learning causes the model to diverge from the prior, the  $Q$ -estimates of language generated according to the prior become unrealistic, and Eq. 4 selects unrealistic actions. This results in highly ‘diverse’ (random) generated utterances. Note that since we operate in discrete action space, we could not include the perturbation model originally proposed by [14], which may be critical to achieving good performance with BCQ.

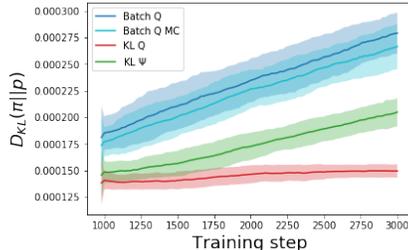


Figure 2: KL-divergence of the policy from the prior is lower with KL-control throughout training. Bands show  $\sigma$ .

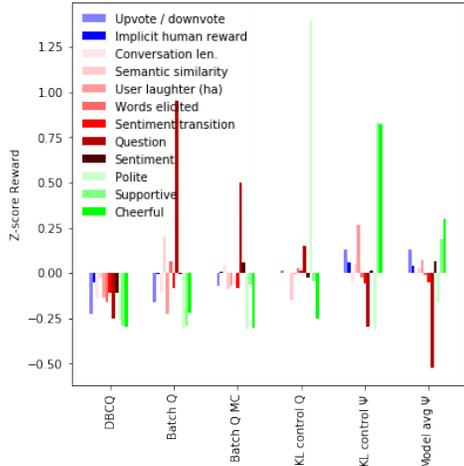


Figure 3: Z-scored metrics used for training (red), post-hoc evaluation (green). Traditional RL methods like Batch Q exploit simple action-based rewards, like asking questions. In contrast, KL-control methods shift their distribution towards polite, supportive, and cheerful conversation, allowing them to elicit higher *human reward* (blue).

contrast, KL-control models learn to rely more on realistic but polite, supportive, and cheerful dialog to elicit higher total *human reward*.

Table 3 presents the results of models trained with only a single reward function, ordered from lowest to highest quality. Notably, extracting multiple different reward functions post-hoc from a batch of data and training on these independently is only possible with an effective BRL model. Here all models are trained with KL-control,  $\Psi$ -learning, and MC targets. Investigating which rewards

The pre-trained prior may be especially important in a generative domain like dialog, where the true reward function is unknown, and so purely maximizing a heuristic may actually lead to lower quality conversations. Table 2 shows examples of conversations with a Batch  $Q$  and KL-control model. Because the Batch  $Q$  model has no incentive to stay close to realistic language, it learns to exploit the reward by asking a question and outputting the maximum number of tokens (30) every utterance. These sentences contain implausible phrases that do not represent realistic language (e.g. “*where did you say to me?*”). In contrast, the KL-control model uses fluent language, but shifts its distribution towards cheerful and polite speech, presumably because this is what led to positive human responses in the batch data.

In fact, we noticed that all models trained with the implicit human rewards described in Section 4.1 learned to use more cheerful and supportive language. Therefore, we create post-hoc metrics to measure this effect (see the supplementary material for details). Figure 3 shows how these metrics, as well as the implicit rewards, differ across models. Without KL-control, baseline methods like Batch  $Q$  exploit simple rewards like asking questions at the expense of realistic language, explaining their poor quality ratings. In contrast,

Table 2: Purely reward-maximizing methods like *Batch Q* trivially exploit the reward function by asking a question every turn, and using the maximum number of tokens in every sentence. In contrast, KL-control methods output plausible language by staying close to the prior, but shift to using polite, cheerful language to maximize implicit human rewards.

<p>[User]: hey, what's up?  [Batch Q]: where did you get??. that sounds so exciting. how do you work for you...  you want to talk to me?... you want  [User]: yes, i want to talk to you!  [Batch Q]: where did you say to me?... you want to talk to you about you...  you are so good for you... you are so good...  [User]: thanks, i appreciate that.  [Batch Q]: where did you get??. that sounds so exciting. how do you work for you...  you want to see you...</p>	<p>[User]: hey, what's up?  [KL-control]: i'm not sure, but i'm glad to see you, and  i'm so happy that you're not alone  [User]: oh thanks, that's very nice of you!  [KL-control]: i'm sorry, but i'm glad to see what you think.  [User]: don't be sorry, you did a good job.  [KL-control]: that's really nice to me.</p>
--	---

presented in Section 4.1 are most critical to achieving high-quality conversations with humans, we note that maximizing positive and minimizing negative sentiment in the user turns out to lead to the highest quality bot. This underscores the importance of affective signals as cues for good conversation. Bots trained on the manual upvotes and downvotes provided by users on the utterance level fail to achieve similarly high performance. Even though users were instructed to make use of the vote feature, the task is burdensome, and users did not vote frequently enough to provide a good training signal. This validates the hypothesis that *implicit* signals of human enjoyment (such as sentiment) are a more scalable way to learn from human preferences.

Table 3: Interactive human evaluation of WOP trained different reward functions.

Reward function	Quality	Fluent	Diverse	Related	Empathy	Total	Votes	Human reward
Conv. len.	2.20 ±.40	3.61 ±.53	3.02 ±.52	2.25 ±.46	2.48 ±.45	13.57 ±1.84	-.035	-.003
Semantic sim.	1.93 ±.34	3.50 ±.45	2.37 ±.45	2.11 ±.45	2.52 ±.48	12.43 ±1.75	-.020	.012
User laughter	1.96 ±.38	3.56 ±.48	2.33 ±.51	1.93 ±.42	3.20 ±.55	12.98 ±1.60	-.149	-.003
Words elicited	2.11 ±.32	3.96 ±.44	3.04 ±.45	2.04 ±.35	2.55 ±.46	13.70 ±1.44	.059	.024
Manual votes	2.14 ±.38	3.47 ±.45	2.91 ±.47	2.07 ±.39	2.42 ±.46	13.00 ±1.65	-.030	.010
Sent. trans.	2.02 ±.31	3.71 ±.49	2.98 ±.50	2.04 ±.42	2.84 ±.48	13.60 ±1.63	.031	.014
Question	2.29 ±.37	<b>4.31 ±.50</b>	<b>3.31 ±.52</b>	2.20 ±.40	2.60 ±.41	14.71 ±1.63	.057	.012
Sentiment	<b>2.47 ±.32</b>	4.05 ±.45	3.23 ±.46	<b>2.42 ±.39</b>	<b>3.23 ±.55</b>	<b>15.40 ±1.49</b>	<b>.085</b>	<b>.045</b>

## 6 Conclusion

This paper presents the Way Off-Policy (WOP) algorithm, which improves performance when learning off-policy without the possibility to explore – i.e. batch RL (BRL). We are the first to propose using KL-control from a strong prior model pre-trained on data as a way to avoid extrapolation and instability in BRL. Our results on traditional RL tasks demonstrate that our WOP algorithm provides performance improvements over state-of-the-art BRL techniques, and the results in dialog generation show that KL-control is critical to achieving good performance in this real-world, high-dimensional setting. In a generative domain such as dialog, the true reward function is not known, and trivially exploiting the rewards can actually lead to worse performance. Thus, KL-control may be particularly necessary to ensure samples remain realistic and close to the data distribution. We propose several reward functions that could allow an open-domain dialog generation model to learn from rich cues implicit in human interaction, where learning from expressed sentiment was most promising. While the proposed rewards are only a first step, we find that maximizing implicit rewards leads to better performance than relying on explicit feedback. We hope that the techniques presented here can improve learning with RL from offline data, making it easier to apply RL to safety-critical settings such as human interaction.

## References

- [1] Abbas Abdolmaleki, Jost Tobias Springenberg, Yuval Tassa, Remi Munos, Nicolas Heess, and Martin Riedmiller. Maximum a posteriori policy optimisation. *arXiv preprint arXiv:1806.06920*, 2018.
- [2] Rishabh Agarwal, Dale Schuurmans, and Mohammad Norouzi. Striving for simplicity in off-policy deep reinforcement learning. *arXiv preprint arXiv:1907.04543*, 2019.

- [3] Kamyar Azizzadenesheli, Emma Brunskill, and Animashree Anandkumar. Efficient exploration through bayesian deep q-networks. In *2018 Information Theory and Applications Workshop (ITA)*, pages 1–9. IEEE, 2018.
- [4] Aditya Bhatt, Max Argus, Artemij Amiranashvili, and Thomas Brox. Crossnorm: Normalization for off-policy td reinforcement learning. *arXiv preprint arXiv:1902.05605*, 2019.
- [5] Graham D Bodie, Kellie St. Cyr, Michelle Pence, Michael Rold, and James Honeycutt. Listening competence in initial interactions i: Distinguishing between what listening is and what listeners do. *International Journal of Listening*, 26(1):1–28, 2012.
- [6] Graham D Bodie, Andrea J Vickery, Kaitlin Cannava, and Susanne M Jones. The role of “active listening” in informal helping conversations: Impact on perceptions of listener helpfulness, sensitivity, and supportiveness and discloser emotional improvement. *Western Journal of Communication*, 79(2):151–173, 2015.
- [7] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, pages 4299–4307, 2017.
- [8] Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680, 2017.
- [9] Thomas Degris, Martha White, and Richard S Sutton. Off-policy actor-critic. In *Proceedings of the 29th International Conference on Machine Learning*, pages 179–186. Omnipress, 2012.
- [10] Mehrdad Farajtabar, Yinlam Chow, and Mohammad Ghavamzadeh. More robust doubly robust off-policy evaluation. In *International Conference on Machine Learning*, pages 1446–1455, 2018.
- [11] Mehdi Fatemi, Layla El Asri, Hannes Schulz, Jing He, and Kaheer Suleman. Policy networks with two-stage training for dialogue systems. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 101–110, 2016.
- [12] Roy Fox, Ari Pakman, and Naftali Tishby. Taming the noise in reinforcement learning via soft updates. In *Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence*, pages 202–211. AUAI Press, 2016.
- [13] Scott Fujimoto, Herke Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International Conference on Machine Learning*, pages 1582–1591, 2018.
- [14] Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. *arXiv preprint arXiv:1812.02900*, 2018.
- [15] Yariv Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059, 2016.
- [16] Milica Gašić, Filip Jurčićek, Blaise Thomson, Kai Yu, and Steve Young. On-line policy optimisation of spoken dialogue systems via live interaction with human subjects. In *2011 IEEE Workshop on Automatic Speech Recognition & Understanding*, pages 312–317. IEEE, 2011.
- [17] Carles Gelada and Marc G Bellemare. Off-policy deep reinforcement learning by bootstrapping the covariate shift. *arXiv preprint arXiv:1901.09455*, 2019.
- [18] Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. Reinforcement learning with deep energy-based policies. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1352–1361. JMLR. org, 2017.
- [19] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, pages 1856–1865, 2018.
- [20] Braden Hancock, Antoine Bordes, Pierre-Emmanuel Mazare, and Jason Weston. Learning from dialogue after deployment: Feed yourself, chatbot! *arXiv preprint arXiv:1901.05415*, 2019.
- [21] Jennifer Hay. Functions of humor in the conversations of men and women. *Journal of pragmatics*, 32(6):709–742, 2000.

- [22] Helena Horton. Microsoft deletes 'teen girl' ai after it became a hitler-loving sex robot within 24 hours. In *Telegraph UK*, 2016.
- [23] Molly E Ireland, Richard B Slatcher, Paul W Eastwick, Lauren E Scissors, Eli J Finkel, and James W Pennebaker. Language style matching predicts relationship initiation and stability. *Psychological science*, 22(1):39–44, 2011.
- [24] Natasha Jaques, Shixiang Gu, Dzmitry Bahdanau, José Miguel Hernández-Lobato, Richard E Turner, and Douglas Eck. Sequence tutor: Conservative fine-tuning of sequence generation models with kl-control. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1645–1654. JMLR. org, 2017.
- [25] Nan Jiang and Lihong Li. Doubly robust off-policy value evaluation for reinforcement learning. In *International Conference on Machine Learning*, pages 652–661, 2016.
- [26] Gregory Kahn, Adam Villafflor, Vitchyr Pong, Pieter Abbeel, and Sergey Levine. Uncertainty-aware reinforcement learning for collision avoidance. *arXiv preprint arXiv:1702.01182*, 2017.
- [27] Sham M Kakade. A natural policy gradient. In *Advances in neural information processing systems (NIPS)*, volume 14, pages 1531–1538, 2002.
- [28] Hilbert J Kappen, Vicenç Gómez, and Manfred Opper. Optimal control as a graphical model inference problem. *Machine learning*, 87(2):159–182, 2012.
- [29] Aviral Kumar, Justin Fu, George Tucker, and Sergey Levine. Stabilizing off-policy q-learning via bootstrapping error reduction. *arXiv preprint arXiv:1906.00949*, 2019.
- [30] Jiwei Li, Alexander H Miller, Sumit Chopra, Marc’ Aurelio Ranzato, and Jason Weston. Dialogue learning with human-in-the-loop. *arXiv preprint arXiv:1611.09823*, 2016.
- [31] Jiwei Li, Will Monroe, Alan Ritter, Dan Jurafsky, Michel Galley, and Jianfeng Gao. Deep reinforcement learning for dialogue generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202, 2016.
- [32] Jiwei Li, Will Monroe, Tianlin Shi, Sébastien Jean, Alan Ritter, and Dan Jurafsky. Adversarial learning for neural dialogue generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2157–2169, 2017.
- [33] Ziming Li, Julia Kiseleva, and Maarten de Rijke. Dialogue generation: From imitation learning to inverse reinforcement learning. *arXiv preprint arXiv:1812.03509*, 2018.
- [34] Bing Liu and Ian Lane. Iterative policy learning in end-to-end trainable task-oriented neural dialog models. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 482–489. IEEE, 2017.
- [35] Bing Liu, Gokhan Tür, Dilek Hakkani-Tür, Pararth Shah, and Larry Heck. Dialogue learning with human teaching and feedback in end-to-end trainable task-oriented dialogue systems. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2060–2069, 2018.
- [36] Yao Liu, Adith Swaminathan, Alekh Agarwal, and Emma Brunskill. Off-policy policy gradient with state distribution correction. *arXiv preprint arXiv:1904.08473*, 2019.
- [37] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.
- [38] Rémi Munos, Tom Stepleton, Anna Harutyunyan, and Marc Bellemare. Safe and efficient off-policy reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 1054–1062, 2016.
- [39] Ofir Nachum, Mohammad Norouzi, Kelvin Xu, and Dale Schuurmans. Bridging the gap between value and policy based reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 2775–2785, 2017.
- [40] Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. Deep exploration via bootstrapped dqn. In *Advances in neural information processing systems*, pages 4026–4034, 2016.
- [41] Jan Peters, Katharina Mülling, and Yasemin Altun. Relative entropy policy search. In *AAAI*, pages 1607–1612. Atlanta, 2010.

- [42] Doina Precup. Eligibility traces for off-policy policy evaluation. *Computer Science Department Faculty Publication Series*, page 80, 2000.
- [43] Konrad Rawlik, Marc Toussaint, and Sethu Vijayakumar. On stochastic optimal control and reinforcement learning by approximate inference. In *Robotics: science and systems*, 2012.
- [44] Martin Riedmiller. Neural fitted q iteration—first experiences with a data efficient neural reinforcement learning method. In *European Conference on Machine Learning*, pages 317–328. Springer, 2005.
- [45] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 1889–1897, 2015.
- [46] Iulian V Serban, Chinnadhurai Sankar, Mathieu Germain, Saizheng Zhang, Zhouhan Lin, Sandeep Subramanian, Taesup Kim, Michael Pieper, Sarath Chandar, Nan Rosemary Ke, et al. A deep reinforcement learning chatbot. *arXiv preprint arXiv:1709.02349*, 2017.
- [47] Iulian Vlad Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron Courville, and Yoshua Bengio. A hierarchical latent variable encoder-decoder model for generating dialogues. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.
- [48] Pararth Shah, Dilek Hakkani-Tur, Bing Liu, and Gokhan Tur. Bootstrapping a neural conversational agent with dialogue self-play, crowdsourcing and on-line reinforcement learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers)*, pages 41–51, 2018.
- [49] Weiyan Shi and Zhou Yu. Sentiment adaptive end-to-end dialog systems. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1509–1519, 2018.
- [50] Jamin Shin, Peng Xu, Andrea Madotto, and Pascale Fung. Happybot: Generating empathetic dialogue responses by improving user experience look-ahead. *arXiv preprint arXiv:1906.08487*, 2019.
- [51] Candace L Sidner, Cory D Kidd, Christopher Lee, and Neal Lesh. Where to look: a study of human-robot engagement. In *Proceedings of the 9th international conference on Intelligent user interfaces*, pages 78–84. ACM, 2004.
- [52] Robert F Stengel. *Stochastic optimal control*. John Wiley and Sons New York, New York, 1986.
- [53] Pei-Hao Su, Paweł Budzianowski, Stefan Ultes, Milica Gasic, and Steve Young. Sample-efficient actor-critic reinforcement learning with supervised data for dialogue management. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 147–157, 2017.
- [54] Philip Thomas and Emma Brunskill. Data-efficient off-policy policy evaluation for reinforcement learning. In *International Conference on Machine Learning*, pages 2139–2148, 2016.
- [55] Emanuel Todorov. Linearly-solvable markov decision problems. In *Advances in neural information processing systems (NIPS)*, pages 1369–1376, 2007.
- [56] Hado Van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.
- [57] Harry Weger Jr, Gina R Castle, and Melissa C Emmett. Active listening in peer interviews: The influence of message paraphrasing on perceptions of listening skill. *The Intl. Journal of Listening*, 24(1):34–49, 2010.
- [58] Li Zhou, Jianfeng Gao, Di Li, and Heung-Yeung Shum. The design and implementation of xiaoice, an empathetic social chatbot. *arXiv preprint arXiv:1812.08989*, 2018.
- [59] Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, and Anind K Dey. Maximum entropy inverse reinforcement learning. In *AAAI*, volume 8, pages 1433–1438. Chicago, IL, USA, 2008.
- [60] Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.