Offline Reinforcement Learning for Mixture-of-Expert Dialogue Management

Anonymous Author(s)
Affiliation
Address
email

Abstract

Reinforcement learning (RL) has shown great promise for developing dialogue management (DM) agents that are non-myopic, conduct rich conversations, and maximize overall user satisfaction. Despite recent developments in RL and language models (LMs), using RL to power conversational chatbots remains challenging, in part because RL requires online exploration to learn effectively, whereas collecting novel human-bot interactions can be expensive and unsafe. This issue is exacerbated by the combinatorial action spaces facing these algorithms, as most LM agents generate responses at the word level. We develop a variety of RL algorithms, specialized to dialogue planning, that leverage recent Mixture-of-Expert Language Models (MoE-LMs)—models that capture diverse semantics, generate utterances reflecting different intents, and are amenable for multi-turn DM. By exploiting MoE-LM structure, our methods significantly reduce the size of the action space and improve the efficacy of RL-based DM. We evaluate our methods in open-domain dialogue to demonstrate their effectiveness w.r.t. the diversity of intent in generated utterances and overall DM performance.

1 Introduction

Natural Language Processing (NLP) has made significant strides in recent years, notably in the field of language generation. Thanks to advances in language modeling, particularly with the use of transformer Vaswani et al. (2017), NLP models can now generate human-like text that is often difficult to distinguish from text written by a person. However, despite these advancements, these models still fall short when it comes to having rich conversations. Current NLP models lack effective dialogue management, as these models are good at generating individual sentences, but struggle with maintaining coherent and engaging conversations. Whereas, most compelling conversations generally span numerous topics, are rather open-ended, and often have an underlying goal (e.g., customer success, task completion, recommendation). This requires dialogue agents to understand the context of the conversation and respond appropriately while maintaining the ability to achieve goals.

Reinforcement learning (RL) is a natural approach for optimizing a dialogue management agent's policy. Earlier work on RL-based dialogue systems relies on specific, hand-crafted semantic states (Levin and Pieraccini, 1997; Singh et al., 2002; Walker, 2000) or partially observable belief states (Williams and Young, 2007; Young et al., 2010), in which case the agent encodes conversations and chooses the best structured dialogue action at each turn. Applications include relational reasoning (Shah et al., 2018), task completion (Shi and Yu, 2018), and query fulfillment (Serban et al., 2017), whose action spaces are structured enough to be represented by hand-crafted features. To handle more complex dialogues, recent approaches use language models to extract semantic representations from conversation histories, treat these representations as dialogue states, and apply RL to learn a word-level generative DM agent (Jaques et al., 2019; Li et al., 2016, 2017; Shin et al., 2020).

However, unlike supervised learning approaches, where one can train imitation agents with offline conversation data, RL DM algorithms require online exploration to learn effectively. Unfortunately,

Submitted to 37th Conference on Neural Information Processing Systems (NeurIPS 2023). Do not distribute.

constant interactions with real users is often expensive and time-consuming. While one can potentially address the DM problem using *offline* RL, issues such as model exploitation leading to distribution shift on the state and action space, when training on static datasets are of paramount concern (Levine et al., 2020). Moreover, the myriad variation of language makes incorporating all possible conversation histories and bot utterances into the state and action spaces of an RL formulation of the DM problem impractical due to the combinatorics at play. As a result, naively applying RL to DM may result in poorly-performing agents that generate incomprehensible utterances (Zhao et al., 2019).

We tackle the issues above, related to the use of offline RL in DM systems, by leveraging recent advances in Mixture-of-Expert Language Models (MoE-LMs) (Chow et al., 2022). Specifically, we develop a suite of offline RL algorithms specialized in dialogue planning that exploit the structure of MoE-LMs. Our methods consist of three main components: **1)** a primitive LM which, using a probabilistic encoder and decoder, is capable of generating diverse semantic intents **1)** a primitive LM that uses a probabilistic encoder-decoder pair to generate sentences with diverse semantics and intents ; **2)** a number of *specialized* expert LMs, each of which generates utterances corresponding to a specific intent; and **3)** a compositional dialogue manager (DM) that, at each turn, given the encoded conversation history and a set of candidate utterance suggested by the experts, selects one candidate utterance for the DM agent to execute as a response to the conversation until now.

Our contributions to offline RL adapted for MoE-based DM agents are four-fold. First, we exploit the hierarchical structure of MoE-LMs, allowing our offline RL methods to work with a significantly smaller, finite action space, hence making the RL problem more tractable. Second, by leveraging pre-trained MoE-LMs—which generate sensible utterances—and offline RL *prior regularization*—which matches our DM's behaviors with that of the primitive LM—our RL algorithms focus on higher-level dialogue planning, and are more data-efficient than standard RL methods by allowing language fluency to be handled by the MoE-LMs. Third, by using the diverse semantic representations of MoE-LMs, our methods operate at the sentence embedding space and have much simpler critic and actor updates. This circumvents the word-level credit-assignment issue that is particularly challenging in long conversations (Saleh et al., 2020). Fourth, in contrast to the findings of Verma et al. (2022), where offline RL agents tend to lack utterance diversity (due to potential reward hacking and optimization of a single objective), our MoE-based DM agents are adept at generating utterances reflecting different intents by design.

We begin with a brief introduction of LMs, the MoE-LM architecture, and the use of MDPs in DM in Section 2. We then describe the pre-training procedure for MoE-LMs—which encode diverse semantics and generate fluent utterances capturing specific intents—in Section 3. We derive four state of the art (SOTA) offline RL algorithms for training MoE-LMs in Section 4, and three MoE-LM specialized offline RL algorithms in Section 5. Finally, in Section 6, we demonstrate the effectiveness of our algorithms in open-domain dialogues w.r.t. their ability to generate utterances with diverse intents and overall DM performance.

2 Preliminaries

Language Models (LMs) In this work, we employ seq2seq LMs Sutskever et al. (2014) to generate the next utterances in a dialogue. We assume access to a dataset of the form $\mathcal{D} = \{(\mathbf{X}^{(k)}, Y^{(k)})\}_{k=1}^{|\mathcal{D}|}$, where each $\mathbf{X}$ is an L-turn conversation history $\mathbf{X} = \{X_l\}_{l=0}^{L-1}$, wherein X_l is the utterance in a conversation at turn l, and Y is the next utterance. Let $N_\mathbf{X}$ be an upper-bound on the length (number of tokens) of each utterance X_l in $\mathbf{X}$[1]. The role of an LM is to predict the probability of the next utterance, Y, consisting of N tokens, conditioned on the conversation history, $\mathbf{X}$; i.e., $\Pr\big(Y = \{y_n\}_{n=1}^{N} \mid \mathbf{X}\big)$. In the transformer architecture (Wolf et al., 2019), a LM first encodes the conversation history $\mathbf{X}$ using an encoder Φ to a $(L \times N_\mathbf{X})$-length sequence of embeddings $\{(z_{l,0}, \ldots, z_{l,N_\mathbf{X}-1})\}_{l=0}^{L-1}$, where each $z_{l,n}$ is a vector in the latent space induced by the encoder Φ. For notational convenience, we concatenate these embeddings into a single embedding $z \in \mathcal{Z} \subseteq \mathbb{R}^d$ where d is the overall dimension of the latent space. The next utterance $\widehat{Y} = \{\widehat{y}_n\}_{n=1}^{N}$ is then sampled, token-by-token, from a decoder Ψ; i.e., $\widehat{Y} \sim \Psi\big(\cdot \mid z\big) := \prod_{n=1}^{N} \Psi\big(\widehat{y}_n \mid \widehat{y}_0, \ldots, \widehat{y}_{n-1}; z\big)$, where $\widehat{y}_0$ is a fixed initial (start-of-sentence) token (Chien and Kuo, 2019), and the latent state is denoted as $z = \Phi(\mathbf{X})$.

Markov Decision Processes (MDPs) have been used to model dialogue management problems in a variety of settings (Li et al., 2016; Asadi and Williams, 2016; Jaques et al., 2019). In such MDPs,

[1] If the actual utterance X_l has fewer tokens than $N_\mathbf{X}$, remaining spaces in the utterance will be padded by a specific token and masked.

denoted by $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, s_0, \gamma)$, the state space $\mathcal{S}$ represents the tokenized conversation history and the initial state $s_0 \in \mathcal{S}$ is the initial user's query. The action space $\mathcal{A}$ is the tokenized language space, with each action $a \in \mathcal{A}$ representing one possible next utterance of the agent. The transition kernel P models the distribution over the user's response to the action taken by the agent (bot) and current conversational context. Finally, the reward function r measures the user's satisfaction as a function of the conversation uptil the most recent step. In these MDPs, we can think of the LM as a policy that maps conversation histories to next utterances. The goal is to find a policy π^* with maximum expected discounted return, i.e., $\pi^* \in \arg\max_\pi J_\pi := \mathbb{E}[\sum_{k=0}^{\infty} \gamma^t r_t \,|\, P, s_0, \pi]$. Note that the size of the tokenized state and action spaces grow exponentially with the vocabulary size. This makes it intractable to solve MDPs of this type even for a medium-size vocabulary.

Mixture-of-expert Language Models (MoE-LMs). (Chow et al., 2022) recently demonstrated promising results using MoE-LMs to enrich a bot's utterances and improve DM (see Figure 1 for an architecture sketch). These results were achieved mainly due to (i) learning a language representation (called as *primitive discovery*) that captures different semantics, (ii) a machinery (*expert construction*) that embeds different intents into sub-models of this LM, so that they can behave appropriately when prompted, and (iii) a compositional dialogue manager module that comprehends the conversation and determines which response deems most appropriate.

For *primitive discovery*, one first learns a language model $\mathtt{LM}_0 = (\Phi, \mathcal{G}_0, \Psi)$ consisting of a *stochastic encoder* $\mathcal{G}_0 \circ \Phi$, which is composed of an encoder Φ that maps tokenized conversation histories $\mathbf{X}$ to a latent space $\mathcal{Z} \subseteq \mathbb{R}^d$ a Gaussian distribution $\mathcal{G}_0(z'|z) := \mathcal{N}\big(\mu_0(z), \sigma_0^2(z)\mathbf{I}_{d\times d}\big)$, and a decoder Ψ, which predicts the next utterance $\widehat{Y}_0$ (token-by-token) conditioned on the point z' sampled from the latent distribution $\Psi(\widehat{Y}_0|z')$, where $z' \sim \mathcal{G}_0(\cdot|z)$. Let $\mathtt{LM}_0(Y|\mathbf{X}) := \mathbb{E}_{z'\sim\mathcal{G}_0(\cdot|z), z=\Phi(\mathbf{X})}[\Psi(Y|z')]$ denote the *primitive*, which predicts the next utterance accurately and also has strong generalization in $\mathcal{Z}$ over a diverse set of possible utterances.

Given a primitive $\mathtt{LM}_0 = (\Phi, \mathcal{G}_0, \Psi)$, the algorithm learns m expert distributions $\{\mathcal{G}_i\}_{i=1}^m$, each defined as $\mathcal{G}_i(z'|z) = \mathcal{N}\big(\mu_i(z), \sigma_i^2(z)\mathbf{I}_{d\times d}\big)$, where each $\mathcal{G}_i$ corresponds to a personality and generates samples in specific parts of the latent space $\mathcal{Z}$. This results in m LMs, $\{\mathtt{LM}_i\}_{i=1}^m$, $\mathtt{LM}_i = (\Phi, \mathcal{G}_i, \Psi)$, each serving as an *expert* that generates one or more candidate next utterances $\widehat{Y}_i$ that are relevant to the conversation $\mathbf{X}$, and also compatible with its respective personality and intent. For dialogue management, the compositional DM μ takes as input the encoded conversation history $z = \Phi(\mathbf{X})$ and candidate action utterances generated by the experts $\{\widehat{Y}_i\}_{i=0}^m$, and selects one of them to execute, i.e., $Y \sim \mu(\cdot \mid z, \{\widehat{Y}_i\}_{i=0}^m)$. Given the state $s = \mathbf{X}$ and action $a = Y$, the MoE-LM policy that optimizes the DM MDP can be expressed as

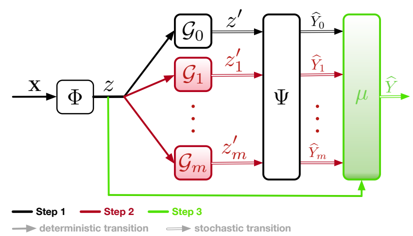


Figure 1: MoE-LM Architecture.

$$\pi_{\text{MoE}}(a|s) = \int_{\{\hat{a}_i, z'_i\}_{i=0}^m} \mu(a|\Phi(s), \{\hat{a}_i\}_{i=0}^m) \prod_{i=0}^m d\Psi(\hat{a}_i|z'_i) d\mathcal{G}_i(z'_i|\Phi(x)). \tag{1}$$

3 Warmstarting the MoE-LM

The MoE-LM approach reformulates the RL dialogue management problem with much smaller state and action spaces and focuses on optimizing the specific goal of the conversation task (as candidate utterances are separately optimized to follow particular bot-based characteristics/intents). Recall that the DM is a policy conditioned on both the latent state and the actions suggested by the experts. Before introducing the different RL methods for DM (Section 4 and 5), in the following we outline (i) the learning of diverse semantics (primitive LM) for conversation histories, which allows the agent to generate a wide variety of utterances, and (ii) the construction of specialized LMs (experts), which generate utterances of different intents.

Following from the primitive discovery procedure in Chow et al. (2022), the primitive LM, $\mathtt{LM}_0$, is learned by solving a KL-constrained optimization problem that aims at capturing diverse semantics:

$$\min_{(\Phi,\mathcal{G}_0,\Psi),\rho} \widehat{\mathbb{E}}_{z'\sim\rho(\cdot|z,Y), z=\Phi(\mathbf{X})}\big[-\log\Psi(Y|z')\big] \text{ s.t. } \widehat{\mathbb{E}}_{z=\Phi(\mathbf{X})}\big[\mathrm{KL}\big(\rho(z'|z,Y)||\mathcal{G}_0(z'|z)\big)\big] \le \epsilon_{\mathrm{KL}}, \tag{2}$$

where $\widehat{\mathbb{E}}$ is the empirical expectation over $(\mathbf{X}, Y)$ in the dataset $\mathcal{D}$, ρ is a distribution over the latent space conditioned on the encoded conversation history z and the target utterance Y, and ϵ_{KL} is a positive real-valued threshold. Using (2), we learn $\texttt{LM}_0 = (\Phi, \mathcal{G}_0, \Psi)$ by maximizing the log-likelihood of sentence Y for a context and latent generation, while enforcing consistency between the latent variable z' predicted by $\mathcal{G}_0(\cdot|z)$ and $\rho(\cdot|z, Y)$ via the KL constraint. The distribution $\rho(\cdot|z, Y)$ is a Gaussian $\mathcal{N}\big(\mu_\rho(z, \Phi_\rho(Y)), \sigma^2_\rho(z, \Phi_\rho(Y))\mathbf{I}_{d\times d}\big)$ in which Φ_ρ is a pre-trained encoder for the target utterance Y, and where the mean $\mu_\rho(\cdot,\cdot)$ and the variance $\sigma^2_\rho(\cdot,\cdot)$ are trainable models. In practice, we implement the KL constraint in (2) as a penalty weighted by a chosen coefficient.

To complete the MoE framework, one needs to train a set of experts $\texttt{LM}_i$, $\forall i \in \{1, \ldots, m\}$, with each generating candidate utterances of different intents. By viewing each expert as a distribution of particular behaviors in conversation data $\mathcal{D}$, we leverage the results in Chow et al. (2022) and adopt a universal encoder-decoder (Φ, Ψ) among all the experts. Therefore, each expert i is parameterized by an arbitrary latent distribution that samples certain regions of the latent space $\mathcal{Z}$. Let $\ell_i(\mathbf{X}, Y) \in \mathbb{R}$ be a real-valued label that *characterizes* the intent of expert $i \in \{1, \ldots, m\}$. We can think of $\ell_i(\mathbf{X}, Y)$ as score assigned to Y resembling how strongly Y exhibits the trait expert i is meant to represent. We train the latent distribution $\mathcal{G}_i(z)$ of expert i by solving the problem

$$\min_{\mathcal{G}_i} \; \widehat{\mathbb{E}}_{z'\sim\mathcal{G}_i(\cdot|z), z=\Phi(\mathbf{X}), Y\sim\Psi(\cdot|z')}[-\ell_i(\mathbf{X}, Y)]. \tag{3}$$

Each expert is learned via *reward-maximization*, where ℓ_i is treated like a reward signal w.r.t. expert i, wherein the expert tries to maximize that intent-aligned reward. Note that there is a correspondence of the above approach with contextual bandits (Chu et al., 2011), for which both the context and action spaces are latent space $\mathcal{Z}$, and the bandit policy is the latent distribution $\mathcal{G}_i$. The choice of greedy reward maximization is to encourage a particular behavior in the expert's immediate utterance rather than trying to control future utterances. Long-term dialogue planning is handled by the compositional dialogue manager. For example, with Gaussian experts $\mathcal{G}_i$, $i \in \{1, \ldots, m\}$, we can use the standard REINFORCE (Sutton et al., 1999a) algorithm where the model parameters (μ_i, σ_i) are updated in the following direction, where $\alpha > 0$ is the learning rate – $\alpha \cdot \mathbb{E}_{z'\sim\mathcal{G}_i(\cdot|z), Y\sim\Psi(\cdot|z')}[\ell_i(\mathbf{X}, Y) \cdot \nabla_{\{\mu_i,\sigma_i\}} \log \mathbb{P}_{\mathcal{G}_i}(z'|z)]$. To reduce the variance of these estimates, we can also adopt the baseline reduction technique in (Greensmith et al., 2004).

4 RL for Mixture-of-Expert DM

Offline RL, in which the policy must be learned from the collected conversations $\mathcal{D}$ (without further online interactions), potentially allows RL DM methods to leverage the abundance of offline conversational data for policy learning. Denote by $(\mathbf{X}, Y, X_+) \sim \mathcal{D}$ a tuple sampled from the offline conversation data $\mathcal{D}$, where X_+ is the follow-up user response, and where $s := \mathbf{X}$, $a := Y$, $r(X_+)$, $s_+ := (\mathbf{X}, Y, X_+)$ are the state, action, reward (w.r.t. the follow-up user response), and next state of the MDP, respectively. One standard offline RL algorithm is Q learning (Watkins and Dayan, 1992) which solves: $\min_Q \mathbb{E}_{(s,a,r,s_+)\sim\mathcal{D}}[(r + \gamma \max_{a_+} Q(s_+, a_+) - Q(s, a))^2]$.

However, with the large action space the inner maximization (also termed as greedification) $\max_{a_+} Q(s_+, a_+)$ is generally computationally intractable. Furthermore, since one cannot ensure that the optimal a^*_+ is sampled from the same action distribution as in the offline RL dataset (an issue worsened by the massive action set), such a co-variate shift in the sampling distribution can cause an overestimation bias of the Q estimate. To alleviate these issues, we propose to leverage the warm-started MoE LM (Sec. 3), where the diverse semantic representation and the expert LMs are learned separately. This is crucial to make our offline RL DM problem tractable as the language fluency is captured by the MoE-LM, while our RL-based DM focuses on higher-level planning strategies. In the following, we describe how this can be achieved via different offline RL algorithms.

Offline RL Methods for MoE LMs: One approach to address the aforementioned offline RL issues is *entropy regularization* (Haarnoja et al., 2018; Carta et al., 2021), which regularizes the greedification step to ensure the learned policy is either diverse enough or close to the behavior (data-generation) policy (e.g., with a Shannon entropy or a KL divergence between these policies). Recall that the primitive LM $(\Phi, \mathcal{G}_0, \Psi)$ models the utterance distribution in $\mathcal{D}$, and the state-action-reward-next-state tuple of the DM MDP (s, a, r, s_+). With the following latent states generated by the primitive LM: $z = \Phi(s)$, $z_a = \Phi((s, a))$, $z_+ = \Phi(s_+)$, we define the latent conversation data $\Phi(\mathcal{D})$ as a collection of (z, z_a, r, z_+) tuples. With Shannon-entropy regularization we can utilize the *soft actor critic* framework (Haarnoja et al., 2018) to develop RL updates for the *value function* $V(z)$, *state-action value function* $Q(z_a)$, and *latent generator* $\mathcal{G}(z'|z)$, which is initialized with the primitive latent expert $\mathcal{G}_0$ that minimizes the following losses:

$$L_Q = \mathbb{E}_{(z,z_a,r,z_+)\sim\Phi(\mathcal{D})}[(r+\gamma V_{\text{tar}}(z_+)-Q(z_a))^2] \tag{4}$$

$$L_V = \mathbb{E}_{z\sim\Phi(\mathcal{D}),(\hat{a},z')\sim\Psi\circ\mathcal{G}(.|z)}\left[Q_{\text{tar}}(z_{\hat{a}}) - \alpha\log\mathcal{G}(z'|z) - V(z)^2\right] \tag{5}$$

$$L_{\mathcal{G}} = \mathbb{E}_{z\sim\Phi(\mathcal{D}),(\hat{a},z')\sim\Psi\circ\mathcal{G}(.|z)}[Q(z_{\hat{a}}) - \alpha\log\mathcal{G}(z'|z)], \tag{6}$$

where the critic Q and V take any encoded conversation histories as input and predict the corresponding cumulative return; $\alpha > 0$ is the entropy temperature; $(V_{\text{tar}}, Q_{\text{tar}})$ are the target value networks; $z' \sim \mathcal{G}(.|z)$ is the latent sample generated by $\mathcal{G}$; $\hat{a} \sim \Psi(z')$ is the utterance sampled from $\Psi \circ \mathcal{G}$; and $z_{\hat{a}} = \Phi((\mathbf{X}, \hat{a}))$ is the corresponding latent state.

From a *hierarchical RL* viewpoint (Sutton et al., 1999b; Saleh et al., 2020), the latent generator behaves like a high-level policy, whose latent sample z' is used to generate a bot utterance via Ψ-decoding (with the primitive decoder Ψ acting as the low-level policy). Extending the above RL updates to the case of relative-entropy (KL) regularization can be straightforwardly done by replacing the term $\log\mathcal{G}(z'|z)$ with $\log(\mathcal{G}(z'|z)/\mathcal{G}_0(z'|z))$, since the primitive LM $(\Phi, \mathcal{G}_0, \Psi)$ approximates the behavior policy and the encoder-decoder pair (Φ, Ψ) is shared among the DMs.

Multiple techniques in value-function parameterization have been employed to tackle the overestimation bias. Fujimoto et al. (2018) proposed maintaining two Q functions, and a *dual Q* function chooses the minimum value between them to avoid overestimation. Jaques et al. (2019) applies dropout in the Q function to maintain an *ensemble* of Q values, and outputs the minimum value to avoid overestimation. By utilizing these methods within the MoE-LM framework, we can propose the following variants of offline RL algorithms: (i) **SAC**, which uses a dual Q function and actor-critic updates in (4) to (6), (ii) **EnsQ**, which uses an ensemble of Q functions and the same updates; and (iii) **KLC**, which uses an ensemble of Q functions and a latent KL-regularized actor-critic update.

Apart from the actor-critic approach that iteratively improves the value functions and the policy, recently Implicit Q Learning (IQL) (Kostrikov et al., 2021), a value-based offline RL algorithm, has shown success in tackling various problems, including task-oriented dialogue management (Snell et al., 2022). Within our MoE-LM framework, we propose the **IQL** DM algorithm, whose value function $V(z)$ minimizes the following loss: $L_V = \mathbb{E}_{(z,z_a)\sim\Phi(\mathcal{D})}[L_2^\tau(Q_{\text{tar}}(z_a) - V(z))]$ where L_2^τ is the expectile regression operator (Koenker and Hallock, 2001) of estimating the top-τ expectile statistics, and the Q function of IQL is updated identically to that of actor-critic in Eq. (4), which estimates $Q(z_a) \approx r + \gamma V(z_+)$ via a least-square loss (Bradtke and Barto, 1996). The V function estimates the top-τ quantile of the state-action $Q(z_a)$ random variable at every latent state z. When $\tau \to 1$ IQL updates converge to the optimal Q function $Q^*(z_a)$, i.e., $\mathbb{E}_{(z_a,r,z_+)\sim\Phi(\mathcal{D})}[(r + \gamma\max_b Q^*(z_{+,b}) - Q^*(z_a))^2] \to 0$, where $z_{+,b} = \Phi((\mathbf{X}, a, X_+, b))$ for any next-action utterance b. Intuitively, IQL leverages the generalization capacity of critic functions to estimate the value of the best action without directly querying the values of unseen actions. This makes it less conservative than most offline RL methods that either constrain the policy's actions to be in-distribution via behavior regularization (e.g., **SAC**, **EnsQ**, **KLC**).

Auto-regressive Decoding in Actor Critic: The actor-critic methods (SAC, EnsQ, KLC), to a certain extent, ameliorated the two issues in offline RL(The inner maximization is replaced with V function learning and covariate shift is controlled by policy entropy regularization.). However, implementing these methods (Eq. (5) to (6)) entails sampling utterances from the current policy, i.e., $\hat{a} \sim \Psi \circ \mathcal{G}$, which involves expensive auto-regressive LM decoding at every training update. To resolve this issue, one may empirically replace $\Psi \circ \mathcal{G}$ with a *teacher-forcing* variant (Toomarian and Bahren, 1995) $\Psi_{\text{TF}}(a) \circ \mathcal{G}$, which replaces auto-regressive decoding with a one-step generation from the bot utterance $a = Y$ in $\mathcal{D}$. This will further restrict the policy update of $\mathcal{G}$ to be close to the behavior policy. In contrast, since IQL does not perform explicit policy updates, it directly circumvents this expensive auto-regressive sampling operation of $\hat{a}$.

DM Construction in MoE-LMs: Recall that in an MoE-LM, the DM policy μ takes the encoded conversation history $z = \Phi(\mathbf{X})$, the $m+1$ candidate action utterances generated by the experts $\{\widehat{Y}_i\}_{i=0}^m$, and selects one of them to execute, i.e., $a \sim \mu(\cdot \mid z, \{\widehat{Y}_i\}_{i=0}^m)$. Given the Q function $Q(z_a)$ learned via any of the above offline RL algorithms, we extract the DM policy μ via softmax greedification over the finite set of MoE candidate utterances i.e., $\mu(a \mid z, \{\widehat{Y}_i\}_{i=0}^m) \propto \exp(\beta \cdot Q(z_a))$, where $\beta > 0$ is the policy temperature. This DM policy uses the Q function to score different candidate utterances and returns an utterance based on the likelihood of these scores.

5 Mixture-of-Expert Offline RL

In Sec. 4, we presented how state-of-the-art offline RL methods are adapted to the MoE framework, which can have limitations due to being agnostic to the model architecture. Recall that MoE dialogue management is a specialized hierarchial reinforcement learning (HRL) problem, which optimizes over a restricted class of DM policies defined by the convex hull of expert policy set (whose weights are defined by the DM policy μ). This problem is of great interest because it reduces the original RL DM problem, with a combinatorial action space, into one that has a much smaller finite action space. Leveraging the *mixture-of-policy* structure, in the following we develop offline RL algorithms that specifically target this HRL problem.

Stochastic-action IQL (SAIQL): Our first approach simply applies IQL to the discrete, stochastic set of candidate action utterances $\{\widehat{Y}_i\}_{i=0}^m$ as generated by the MoE experts. Equipped with the latent conversation data $\Phi(D) = \{(z, z_a, r, z_+)\}$ (see Sec. 4) and the latent expert policies $\{\mathcal{G}_i\}_{i=0}^m$ in the MoE-LM, we propose the following DM algorithm, whose value function $V(z)$ minimize the loss:

$$L_V = \frac{1}{m+1}\sum_{i=0}^{m}\mathbb{E}_{z,\hat{a}_i\sim\Psi\circ\mathcal{G}_i(\cdot|z)}[L_2^\tau(Q_{\text{tar}}(z_{\hat{a}_i}) - V(z))], \tag{7}$$

where $z_{\hat{a}_i} = \Phi((\mathbf{X}, \hat{a}_i))$ is the latent state that corresponds to the action utterance sampled from the i-th expert, L_2^τ is the expectile regression operator, and the Q function is updated based on Eq. (4). To incorporate the maximization over candidate utterances from the experts into IQL, we compute the expectile regression over the joint latent state and expert policy distributions.

However, unlike the standard IQL DM algorithm, which avoids autoregressive decoding for policy execution, SAIQL requires auto-regressive sampling of all $m+1$ candidate utterances. Suppose the augmented latent conversation data $\Phi(D)_{\text{SA}} = \{(z, z_a, r, z_+, \{z_{\widehat{Y}_i}\}_{i=0}^m)\}$ (which also includes the set of latent expert actions $\{z_{\widehat{Y}_i}\}_{i=0}^m$) is available. One straightforward way to circumvent this issue is by replacing the expectation over experts with the realized candidate utterances, i.e., by approximating the value function in SAIQL with its unbiased empirical average $\frac{1}{m+1}\sum_{i=0}^m \mathbb{E}_{(z,\{z_{\widehat{Y}_i}\}_{i=0}^m)\sim\Phi(\mathcal{D})_{\text{SA}}}[L_2^\tau(Q_{\text{tar}}(z_{\widehat{Y}_i}) - V(z))]$.

While having access to candidate utterances is not standard in IQL, it is necessary here to allow Q-Learning to exploit quantile regression over *realized* candidate utterances (an approach shown to be sound in Boutilier et al. (2018)). Therefore, we termed this method *stochastic action* IQL (**SAIQL**) to reflect the stochastic action sets used in IQL training. Once **SAIQL** converges, the DM policy is also constructed as a softmax of Q values applied to each candidate utterance.

The **MoE MDP** is defined as $\bar{\mathcal{M}} = (\bar{\mathcal{S}}, \bar{\mathcal{A}}, \bar{P}, \bar{r}, \bar{s}_0, \gamma)$, where the state space is the product of the learned latent space $\mathcal{Z}$ and the joint action space of the $m+1$ experts, i.e., $\bar{\mathcal{S}} = \mathcal{Z}\times\mathcal{A}^{m+1}$, the action space consists of the $m+1$ experts, i.e., $\bar{\mathcal{A}} = \{0, \ldots, m\}$, its initial state $\bar{s}_0$ is the encoding of the initial user's query and the utterances suggested by the experts in response to this query, the transition models both the user's responses and also the next experts' actions, and the reward is the same as in the original MDP. Since MoE-MDP has a finite number of actions, learning a policy λ is equivalent to solving a finite-action MDP: $\lambda^* \in \arg\max_\lambda J_\lambda := \mathbb{E}[\sum_{k=0}^\infty \gamma^t \bar{r}_t \,|\, \bar{P}, \bar{s}_0, \lambda]$.

Follow-the-Leading-Expert (FtLE): Banijamali et al. (2019) showed that the MoE-MDP problem is NP-hard but can be approximated by $\max_{\lambda\in\Delta^{m+1}}\sum_{i=0}^m \lambda(i)V^i(z) + \mathcal{U}(\bar{\mathcal{M}})$, where V^i is the value function of the i-th expert and $\mathcal{U}(\bar{\mathcal{M}}) > 0$ is a surrogate function that depends on the experts' stationary distributions. However, computing these distributions is generally intractable as the experts are LMs themselves. This motivates our heuristic **FtLE** algorithm, which ignores the second term, to train a set of expert critic functions, and picks the best action at each step. To efficiently parameterize the critic function, similarly to the architecture used in DQN (Mnih et al., 2013) for discrete-action RL, we define a $(m+1)$-headed critic function, where each head represents the value of following an expert's policy. We then modify the standard critic loss functions as follows, in order to train the multi-headed critic functions:

$$L_Q = \sum_{i=0}^{m}\mathbb{E}_{z,z_a,r,z_+}[(r + \gamma V_{\text{tar}}^i(z_+) - Q^i(z_a))^2], \quad L_V = \sum_{i=0}^{m}\mathbb{E}_{z,\hat{a}_i\sim\Psi\circ\mathcal{G}_i(\cdot|z)}[(Q_{\text{tar}}^i(z_{\hat{a}_i}) - V^i(z))^2], \tag{8}$$

where Q^i and V^i represent the critic-function head for expert i. To overcome the auto-regressive sampling issue in Eq. (8), we relabel the offline conversation data $\mathcal{D}$ by assigning action utterances to train the critic function(s) whose corresponding expert(s) most likely generate those utterances. Specifically, consider the following V-function loss

$$L_V = \sum_{i=0}^{m} \mathbb{E}_{z,z_a,Y}[\mathbf{1}_{i=i(z,Y)} \cdot (Q^i_{\text{tar}}(z_a) - V^i(z))^2], \tag{9}$$

where $\mathbf{1}_{i=i(z,z_a)}$ selects the expert based on the best log-likelihood $i(z,Y) := \arg\max_i \log \Psi(Y|z^{i,\prime})$, with $z^{i,\prime} \sim \mathcal{G}_i(.|z)$. After learning the critic functions, the **FtLE** DM policy can then constructed via $\mu(a \mid z, \{\widehat{Y}_i\}_{i=0}^m) \propto \exp(\beta Q^{i(z,a)}(z_a))$.

Value-based RL for MoE-MDP (MoE-VRL): Consider a $(m+1)$-headed value function Λ of the MoE-MDP, where each head represents the optimal value by choosing the corresponding expert's action. Applying standard DQN, this function can be learned by minimizing the following loss:

$$L_\Lambda = \mathbb{E}_{z,Y,r,z_+}[(r + \gamma \max_{i_+} \Lambda_{\text{tar}}(z_+, i_+) - \Lambda(z, i(z,Y)))^2], \tag{10}$$

where Λ_{tar} is the target-Λ network. For simpler exposition, we only use the partial MoE-MDP states of encoded conversations in the above DQN loss and omit the candidate action utterances. Extending to the full MoE-MDP state is straight-forward but is omitted for brevity. The inner maximization over i_+ can be computed explicitly because the MoE-MDP action space of expert indices is finite and small. Here, $i(z,Y)$ is the same index function that attributes utterance Y to the expert that most likely generates it, based on likelihood. With the optimal value function $\Lambda^*(z,i)$, the MoE-MDP policy picks the best expert $\lambda^*(z) := \arg\max_i \Lambda^*(z,i)$, and the DM policy can be constructed as $\mu(a \mid z, \{\widehat{Y}_i\}_{i=0}^m) \propto \exp(\beta Q^{\lambda^*(z)}(z_a))$, where $Q^{\lambda^*(z)}(z_a)$ is the critic of the optimal expert.

6 MoE-based DM Experiments

We evaluate our MoE-based offline RL algorithms on two open-domain benchmarks that are common in the RL-based dialogue management literature (Jaques et al., 2019). The first one is the Cornell Movie corpus (Danescu-Niculescu-Mizil and Lee, 2011), which consists of conversations between speakers in different movie. The second is the Reddit Casual (Ghandeharioun et al., 2019) conversations dataset, which is a subset of the Reddit corpus that only contains casual conversations.

Environment: We perform the experiment by having DM agents interact with a DialoGPT (Zhang et al., 2019) simulated-user environment. The task is to maximize user satisfaction, which is measured by the user's overall sentiment. To construct an immediate reward, we set $r(X_+) := \ell_{\text{sent}}(X_+)$, where $\ell_{\text{sent}}(X)$ is a RoBerTa-based sentiment classifier (Liao et al., 2021), which assigns a score from $[-1, 1]$ that is inversely proportional to the (negative) positive sentiment prediction probabilities.

We pre-train the MoE-LM with either the Cornell or Reddit dataset and construct 10 experts (i.e., $m = 9$, plus the primitive expert), each corresponding to an individual intent in open-ended dialogues, including "empathy", "optimism", "cheerfulness", "contentment", "dejection", "rage", "sorrow", "questioning", "exploration", etc. See Appendix B for details. The conversation lasts for a total of 5 turns (with $\gamma = 0.8$), where each turn entails a query/response from the user followed by an agent's utterance. During the agent's turn, each expert generates 5 candidate utterances thus resulting in a total of 50 candidate utterances. To evaluate the methods, we measure the return of the trajectory generated by different agents via $\mathbb{E}_{\mathbf{X}_0 \sim \mathcal{D}}[\sum_{i=0}^4 \gamma^i r(X_{i+1}) | Y_i \sim \text{LM}(.|\mathbf{X}_i), X_{i+1} \sim P_{\text{Dialog-GPT}}(.|\mathbf{X}_i, Y_i)]$

Evaluation: We employ two evaluation approaches, namely (i) a model-free approach that only utilizes the learned Q function to score candidate utterances, and where the DM policy selects the action utterance based on a softmax likelihood; and (ii) a model-based approach that uses the Value function (V) along with a learned next-user utterance model $P_{\text{user}}(X_+|z_Y)$, that optimizes the following loss: $L_{P_{\text{user}}} = \mathbb{E}_{(z_a,r)\sim\mathcal{D}, \hat{X}_+ \sim P_{\text{user}}(.|z_a)}[(r - r(\hat{X}_+))^2]$. We first approximate the Q function via $Q(z_a) \approx r(\hat{X}_+) + \gamma V(\hat{z}_+)$, where $\hat{X}_+$ denotes the next user utterance sampled from $P_{\text{user}}(.|z_a)$, then use that function to score candidate utterances, and, finally have the DM policy select the action utterance analogously. Human evaluation is also conducted on the DM performances of different offline RL agents. More details and results can be found in Appendix E and **??**.

Experiment 1: SOTA Offline RL with MoE-LMs: The goal of this experiment is to investigate the effectiveness of SOTA offline RL algorithms. In these experiments we only make use of the primitive language model $\texttt{LM}_0 = (\Phi, \mathcal{G}_0, \Psi)$ to generate sample utterances. To simulate previous works using single policy settings, we fine-tune the latent base distribution $\mathcal{G}_0$ for policy optimization

Table 1: SOTA offline RL methods.

Algo Name	Reddit Casual		Cornell	
	Model Free	Model Based	Model Free	Model Based
IQL	0.53 ± 0.47	$\mathbf{4.25 \pm 0.12}$	$\mathbf{-1.32 \pm 0.19}$	$\mathbf{1.47 \pm 0.15}$
SAC	$\mathbf{0.97 \pm 0.52}$	4.13 ± 0.21	-1.55 ± 0.19	0.36 ± 0.26
EnsQ	0.10 ± 0.40	4.06 ± 0.25	-1.51 ± 0.20	0.21 ± 0.21
KLC	0.31 ± 0.46	3.69 ± 0.37	-1.46 ± 0.21	-0.07 ± 0.25
BC	-0.65 ± 0.41		-2.18 ± 0.36	
Bandits	$\mathbf{4.3 \pm 0.16}$		1.3 ± 0.17	

Table 2: MoE specific offline RL methods.

Algo Name	Reddit Casual		Cornell	
	Model Free	Model Based	Model Free	Model Based
EXP 1*	0.97 ± 0.52	4.25 ± 0.12	-1.32 ± 0.19	1.47 ± 0.15
SAIQL	0.81 ± 0.42	$\mathbf{4.65 \pm 0.06}$	-1.34 ± 0.25	2.61 ± 0.24
FtLE	$\mathbf{1.14 \pm 0.49}$	4.59 ± 0.07	$\mathbf{-0.39 \pm 0.24}$	3.51 ± 0.19
MoE-VRL	0.72 ± 0.47	4.46 ± 0.10	-0.58 ± 0.24	$\mathbf{3.62 \pm 0.17}$

while keeping the encoder-decoder (Φ, Ψ) fixed. As mentioned in Sec. 4 we deploy the following offline RL algorithms to train the DM policy μ of MoE-LMs: (i) **SAC** (Haarnoja et al., 2018) with a dual Q function critic (Fujimoto et al., 2018); (ii) **EnsQ**, which utilizes an ensemble of Q functions (Jaques et al., 2019) with actor-critic; (iii) **KLC** (Saleh et al., 2020), which utilizes the dual Q function and applies KL regularization between the latent policy $\mathcal{G}$ and the primitive policy $\mathcal{G}_0$, i.e., $\mathbb{E}_{\mathcal{G}(\cdot|z)}[\log(\mathcal{G}(z'|z)/\mathcal{G}_0(z'|z))]$ in the actor-critic algorithm update [2]; (iv) **IQL** (Kostrikov et al., 2021), which adopts the idea from Q learning to estimate an optimal Q function in the MoE-LM latent space. To our knowledge, our work is among the first that uses IQL for open-domain dialogue management. These methods have been implemented in ways where the original idea has been preserved, making the comparison fair to the original works. With each learned Q function, the bot picks the final action by sampling from a softmax distribution of Q scores over all candidate utterances. To demonstrate the efficacy of offline RL methods, we also include results from Behavior Cloning (**BC**) as well as simple reward maximization (**Bandit**)(i.e, $\gamma = 0$) for comparisons.

Table 1 presents the results of our experiments with these methods in the open-dialogue system, where a 5-turn conversation was generated. The table displays the mean return over 100 conversations with their respective standard errors. Our experiments demonstrate that model-based evaluation can significantly improve dialogue management over the model-free counterpart, even with a next-user LM that is much simpler than the Dialog-GPT user. Among most model-based and model-free evaluations, we found that **IQL**, originally designed to tackle offline RL problems, outperforms other RL methods. This performance can be attributed to IQL's ability to (i) alleviate Q overestimation errors due to co-variate shifts; (ii) estimate the optimal values without being overly conservative w.r.t. the behavior policy, and (iii) avert the auto-regressive utterance sampling issues in training.

Interestingly, we also found that **KLC** and **EnsQ**, two standard methods in RL-based DM, struggled to achieve satisfactory performance in our experiments. This may be due to the fact that applying dropout (for ensemble Q) and KL regularization in the fixed MoE-LM latent space makes DM algorithms overly conservative. In contrast, **SAC** successfully learns a well-performing model-free DM policy but fails in the model-based regime, potentially demonstrating its instability in critic-function learning. **BC** also fails to provide any satisfactory performance on any of the domains and surprisingly, **Bandit** method or plain reward maximization did as well as **IQL**, pointing to the fact that maybe the offline RL methods being used or not exactly helping in planning at all.

Experiment 2: MoE-specific Offline RL: In this experiment, we explore the benefits of leveraging the MoE framework for training offline RL agents in open-domain conversational systems. Building upon the insights from our previous experiment (Experiment 1), we propose several modifications to standard Offline RL algorithms to take advantage of the MoE framework. As mentioned in Sec. 5, we developed the following MoE-specific offline RL algorithms for DM: (i) **SAIQL**, which extends IQL to incorporate the multiple candidate utterances generated by the experts; (ii) **FtLE**, which learns a DM policy to follow the best expert policy at each step (estimation of the experts' long-term values is done concurrently with a multi-headed critic architecture and data relabeling) and (iii) **MoE-VRL**, which learns an optimal meta-value function over the space of experts. Leveraging the MoE-MDP formulation, solving which leads to an optimal DM policy that provides the optimal sequences of expert policy switching. We aim to evaluate the potential of these MoE-specialized offline RL algorithms over off-the-shelf offline RL methods in DM.

Table 2 shows the return observed similar to ones displayed in table 1. The first row in the table displays the best performance across all methods from Experiment 1, for comparison. Our results demonstrate the efficacy of the proposed methods that utilize the structure of the MoE framework

[2]The RL DM approach in Jaques et al. (2019) which applies KL regularization at the word-level LM policy is not applicable to our case because our DM policy is defined in the latent space.

in dialogue management. All the methods that used all experts while training (**SAIQL**, **FtLE**, and **MoE-VRL**) outperformed the SOTA offline RL methods, indicating that an offline RL algorithm that takes the candidate utterances into account can generally improve dialogue planning. Moreover, making the RL algorithms attuned to the multiple-expert structure (i.e., **FtLE** and **MoE-VRL**) indeed results in even better DM performance, emphasizing the benefits of reformulating the DM MDP using the HRL paradigm, where the DM policy is optimized over a restricted class of finite-action policies. Also, we note that only MoE-aware offline RL methods were actually able to outperform simple per-step greedification (i.e. **Bandit**) which hints to the fact that they were actually able to plan ahead and perform long-term credit assignments to optimize return. Whereas all the standard offline RL methods failed to do that (Table 1). Using multiple critic functions to separately estimate the value of different experts also allows us to better understand their long-term utility (of the corresponding intents) and how they affect the conversation quality. Overall, these findings highlight the potential of the MoE-specific offline RL methods to improve dialogue management performance.

Experiment 3 aims to investigate the effectiveness of selecting different experts during dialogue management. To this end, we conduct a study where we measure the frequency with which different experts are selected throughout the conversation. Specifically, we demonstrate the diversity of intents in different offline RL algorithms in the model-based evaluation of the Cornell dataset. Given approximately 200 conversation turns, we measure the frequency of the expert agents when their utterances are selected and preset such frequency metric for the worst performing Offline RL method (**EnsQ**), a good performing method (**IQL**), and an MoE-specific RL algorithm (such as **MoE-VRL**). To visualize our findings, we plot a histogram of the frequencies on different experts being selected and calculate the KL divergence distance of this histogram and a uniform distribution over the experts. While we acknowledge that a uniform distribution may not be the optimal distribution of utterances, it provides a measure of how well the agents make use of different experts, along with their actual performance.

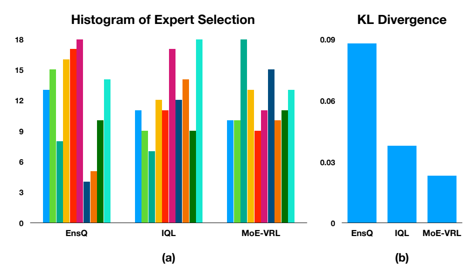


Figure 2: Experiment on the Cornell dataset with Model-based evaluation(a) Histogram of frequency of expert selection. (b) KL divergence against a uniform distribution

The results of Experiment 3 are shown in Figure 2, where we plot the frequency histogram of different expert agent utterances. We observe that the worst performing agent, **EnsQ**, has a highly skewed distribution of expert selections, with a few experts being heavily favored over others. This suggests that **EnsQ** is less diverse and does not effectively utilize the full range of expert knowledge available. On the other hand, both **IQL** and **MoE-VRL** exhibit a more balanced distribution of expert selection, with utterances chosen from multiple experts throughout the conversation; i.e., their frequency distributions are closer to a uniform distribution, with much lower KL divergence distance.

However, there is a clear performance gap between the two methods, with **MoE-VRL** significantly outperforming **IQL**. This highlights the importance of incorporating the MoE framework to better utilize the knowledge of different experts in dialogue planning, rather than relying on generating a diverse set of candidate utterances. Overall, these results suggest that encouraging diversity in intents and better utilizing expert knowledge in planning are essential to improve DM performance.

7 Concluding Remarks

By leveraging the recent advances of Mixture-of-Expert Language Models (MoE-LMs), we developed a suite of offline RL-based DM algorithms. Our methods significantly reduce the action space and improve the efficacy of DM. To understand how well our offline RL approaches generate diverse utterances and solve DM problems, we evaluated them on two open-domain dialogue tasks and compared them with SOTA offline RL baselines. Our results showed that by exploiting the MoE-LM structure, our specialized offline RL DM methods (i) improve the diversity of intents in bot utterances; (ii) have better sample efficiency; and (iii) yield better overall performance in both the model-based and model-free settings. Our work provides important insights on how to create scalable RL-based DM methods that train chatbots to achieve dialogue tasks and enhance user satisfaction. Future work includes fine-tuning the experts (i.e., low-level policies) with offline RL, learning the optimal semantic representation for hierarchical RL, preventing dialogue agents from generating harmful behaviors (e.g., by enforcing safety constraints in the RL algorithms), and evaluating our DM methods on more realistic problems, such as customer support, conversational recommendation, and persuasion.

References

Asadi, K. and Williams, J. (2016). Sample-efficient deep reinforcement learning for dialog control. *arXiv preprint arXiv:1612.06000*.

Bahl, L., Jelinek, F., and Mercer, R. (1983). A maximum likelihood approach to continuous speech recognition. *IEEE transactions on pattern analysis and machine intelligence*, (2):179–190.

Banijamali, E., Abbasi-Yadkori, Y., Ghavamzadeh, M., and Vlassis, N. (2019). Optimizing over a restricted policy class in mdps. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 3042–3050. PMLR.

Boutilier, C., Cohen, A., Hassidim, A., Mansour, Y., Meshi, O., Mladenov, M., and Schuurmans, D. (2018). Planning and learning with stochastic action sets. In *Proc. of the 27th International Joint Conf. on Artificial Intelligence*, pages 4674–4682.

Bradtke, S. J. and Barto, A. G. (1996). Linear least-squares algorithms for temporal difference learning. *Machine learning*, 22(1-3):33–57.

Carta, S., Ferreira, A., Podda, A. S., Recupero, D. R., and Sanna, A. (2021). Multi-dqn: An ensemble of deep q-learning agents for stock market forecasting. *Expert systems with applications*, 164:113820.

Chien, J. and Kuo, C. (2019). Markov recurrent neural network language model. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 807–813. IEEE.

Chow, Y., Tulepbergenov, A., Nachum, O., Ryu, M., Ghavamzadeh, M., and Boutilier, C. (2022). A mixture-of-expert approach to rl-based dialogue management. *CoRR*, abs/2206.00059.

Chu, W., Li, L., Reyzin, L., and Schapire, R. (2011). Contextual bandits with linear payoff functions. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 208–214. JMLR Workshop and Conference Proceedings.

Danescu-Niculescu-Mizil, C. and Lee, L. (2011). Chameleons in imagined conversations: A new approach to understanding coordination of linguistic style in dialogs. *arXiv preprint arXiv:1106.3077*.

Fujimoto, S., van Hoof, H., and Meger, D. (2018). Addressing function approximation error in actor-critic methods. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*.

Ghandeharioun, A., Shen, J., Jaques, N., Ferguson, C., Jones, N., Lapedriza, A., and Picard, R. (2019). Approximating interactive human evaluation with self-play for open-domain dialog systems. *Advances in Neural Information Processing Systems*, 32.

Greensmith, E., Bartlett, P., and Baxter, J. (2004). Variance reduction techniques for gradient estimates in reinforcement learning. *Journal of Machine Learning Research*, 5(9).

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. (2018). Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *ICML*.

Hurley, N. and Rickard, S. (2009). Comparing measures of sparsity. *IEEE Transactions on Information Theory*, 55(10):4723–4741.

Jaques, N., Ghandeharioun, A., Shen, J., Ferguson, C., Lapedriza, A., Jones, N., Gu, S., and Picard, R. (2019). Way off-policy batch deep reinforcement learning of implicit human preferences in dialog. *arXiv:1907.00456*.

Koenker, R. and Hallock, K. F. (2001). Quantile regression. *Journal of economic perspectives*, 15(4):143–156.

Kostrikov, I., Nair, A., and Levine, S. (2021). Offline reinforcement learning with implicit q-learning. *arXiv preprint arXiv:2110.06169*.

Levin, E. and Pieraccini, R. (1997). A stochastic model of computer-human interaction for learning dialogue strategies. In *Eurospeech*, volume 97, pages 1883–1886. Citeseer.

Levine, S., Kumar, A., Tucker, G., and Fu, J. (2020). Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*.

Li, J., Galley, M., Brockett, C., Gao, J., and Dolan, B. (2015). A diversity-promoting objective function for neural conversation models. *arXiv preprint arXiv:1510.03055*.

Li, J., Monroe, W., Ritter, A., Galley, M., Gao, J., and Jurafsky, D. (2016). Deep reinforcement learning for dialogue generation. *arXiv preprint arXiv:1606.01541*.

Li, J., Monroe, W., Shi, T., Jean, S., Ritter, A., and Jurafsky, D. (2017). Adversarial learning for neural dialogue generation. *arXiv preprint arXiv:1701.06547*.

Liao, W., Zeng, B., Yin, X., and Wei, P. (2021). An improved aspect-category sentiment analysis model for text sentiment analysis based on roberta. *Applied Intelligence*, 51(6):3522–3533.

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

Saleh, A., Jaques, N., Ghandeharioun, A., Shen, J., and Picard, R. (2020). Hierarchical reinforcement learning for open-domain dialog. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8741–8748.

Serban, I., Sankar, C., Germain, M., Zhang, S., Lin, Z., Subramanian, S., Kim, T., Pieper, M., Chandar, S., Ke, N., et al. (2017). A deep reinforcement learning chatbot. *arXiv preprint arXiv:1709.02349*.

Shah, P., Hakkani-Tur, D., Liu, B., and Tür, G. (2018). 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.

Shi, W. and Yu, Z. (2018). Sentiment adaptive end-to-end dialog systems. *arXiv preprint arXiv:1804.10731*.

Shin, J., Xu, P., Madotto, A., and Fung, P. (2020). Generating empathetic responses by looking ahead the user's sentiment. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7989–7993. IEEE.

Singh, S., Litman, D., Kearns, M., and Walker, M. (2002). Optimizing dialogue management with reinforcement learning: Experiments with the njfun system. *Journal of Artificial Intelligence Research*, 16:105–133.

Snell, C., Kostrikov, I., Su, Y., Yang, M., and Levine, S. (2022). Offline rl for natural language generation with implicit language q learning. *arXiv preprint arXiv:2206.11871*.

Sutskever, I., Vinyals, O., and Le, Q. (2014). Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27.

Sutton, R., McAllester, D., Singh, S., and Mansour, Y. (1999a). Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12.

Sutton, R. S., Precup, D., and Singh, S. (1999b). Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artif. Intell.*

Toomarian, N. and Bahren, J. (1995). Fast temporal neural learning using teacher forcing.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

Verma, S., Fu, J., Yang, M., and Levine, S. (2022). Chai: A chatbot ai for task-oriented dialogue with offline reinforcement learning. *arXiv preprint arXiv:2204.08426*.

Walker, M. (2000). An application of reinforcement learning to dialogue strategy selection in a spoken dialogue system for email. *Journal of Artificial Intelligence Research*, 12:387–416.

Watkins, C. J. C. H. and Dayan, P. (1992). Technical note q-learning. *Mach. Learn.*

Williams, J. and Young, S. (2007). Partially observable markov decision processes for spoken dialog systems. *Computer Speech & Language*, 21(2):393–422.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., and Brew, J. (2019). Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771.

Young, S., Gašić, M., Keizer, S., Mairesse, F., Schatzmann, J., Thomson, B., and Yu, K. (2010). The hidden information state model: A practical framework for pomdp-based spoken dialogue management. *Computer Speech & Language*, 24(2):150–174.

Zhang, Y., Sun, S., Galley, M., Chen, Y., Brockett, C., Gao, X., Gao, J., Liu, J., and Dolan, B. (2019). Dialogpt: Large-scale generative pre-training for conversational response generation. *arXiv preprint arXiv:1911.00536*.

Zhao, T., Xie, K., and Eskenazi, M. (2019). Rethinking action spaces for reinforcement learning in end-to-end dialog agents with latent variable models. *arXiv preprint arXiv:1902.08858*.

A Additional Results

A.1 Diversity over all agents and Datasets

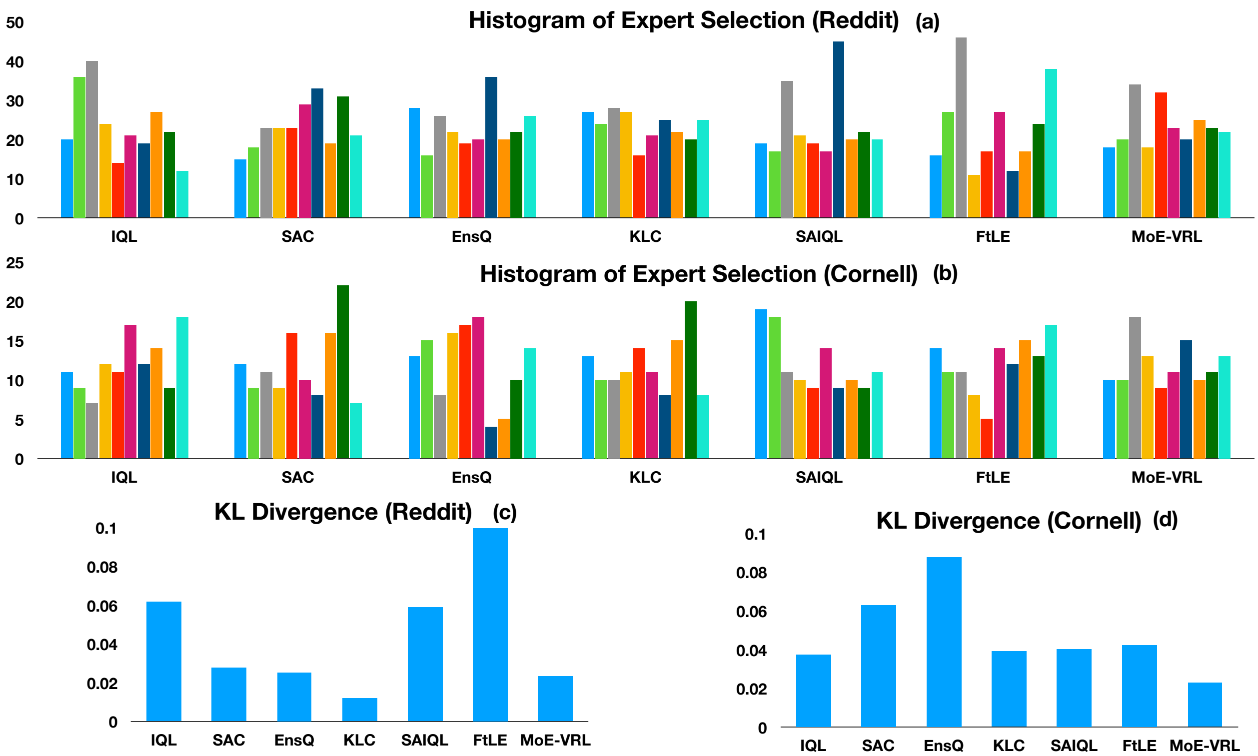


Figure 3: Diversity for all agents (a) Reddit with Model-based approximation, (b) Cornell with Model-based approximation, (c) and (d) depict the KL divergence of all agents w.r.t. to uniform distribution for Reddit and Cornell.

A.2 Different Metrics for MoE-LM's

To measure the quality of LMs learned in MoE-LM we measure the following three metrics, similar to Chow et al. (2022) for 25 generated utterances. **Diversity** : measured as 1 - Sparsity Hurley and Rickard (2009) of the singular values of the embedded utterances, **Gram- {1,2,3}** Li et al. (2015) : Ratio of unique {uni, bi, tri}-gram in generated utterances, and finally **Perplexity** Bahl et al. (1983).

Dataset	Diversity	Gram-1	Gram-2	Gram-3	Perplexity
Reddit	0.14 ± 0.05	0.35	0.77	0.90	38.81 ± 17.34
Cornell	0.12 ± 0.04	0.31	0.60	0.79	43.87 ± 28.81

Table 3: Diversity, Gram-{1,2,3}, and Perplexity of the MoE-LM primitive expert on Reddit Casual and Cornell

Dataset	Question	Exploration	Positive Sent.	Negative Sent.	Sent. Coherence	Joy	Optimism	Anger	Sadness
Reddit	0.95 ± 0.27	0.47 ± 0.21	3.29 ± 0.33	1.42 ± 0.38	0.51 ± 0.40	1.99 ± 0.38	1.25 ± 0.43	1.48 ± 0.39	2.01 ± 0.46
Cornell	1.58 ± 0.39	0.33 ± 0.17	3.55 ± 0.99	1.90 ± 0.5	0.69 ± 0.40	2.44 ± 0.71	2.11 ± 0.99	2.71 ± 0.69	3.45 ± 0.83

Table 4: Quality of Each Expert Trained on Reddit Casual and Cornell with respect to their trained label.

B Experimental Details

This section describes more details about our experimental setup to evaluate the algorithms.

B.1 Model parameters and Description

Language Model Description We make use of the **MoE-2** model as described in Chow et al. (2022) which is based on transformer Vaswani et al. (2017). This variant of MoE had shown diversity in its utterances while retaining semantic fluency with low perplexity. The model was not too large that it would become too costly to use it while training. We are repeating the details of the model over here for ease of the user, but the details remain the same from Chow et al. (2022).

Our MoE uses the simple transformer architecture, where the model parameters are summarized in Table 5:

Parameter	**Value**
Number of layers	2
Embedding hidden size	256
FFN inner hidden size	512
Attention heads	8
Key size	256
Value size	256
Dropout	0.1

Table 5: Simple Transformer Architecture

Latent distributions $\{\mathcal{G}_i\}$ are implemented as FFN that model mean and variance of the normal distribution. We use a target entropy of 1.0. The parameters for FFN are captured in Table 6 (note: FFN has a final layer without an activation).

$\{\mathcal{G}_i\}$ **FFN parameter**	**Value**
Number of layers	1
Activation	tanh
FFN Hidden Size	128

Table 6: $\{\mathcal{G}_i\}$ FFN architecture

B.2 Computational resources

Training and evaluation were run on 8 GPU instances with 32GB of RAM and a NVIDIA Tesla P100 graphics card. Training each experts takes around 2-3 days, and training each RL can take around 12 hours.

B.3 Dataset

Our models were developed using two conversational datasets, namely Reddit Casual and Cornell Movie. We obtained these datasets from the Neural Chat datasets of the MIT Media Lab, which is available at the following link: `https://affect.media.mit.edu/neural_chat/datasets`. These datasets comprise conversations between two speakers and each batch of training data consists of a subset of these conversations. The Reddit Casual dataset is approximately three times larger than the Cornell corpus.

B.4 Offline RL Training & Details

Table 7 summarizes the hyper-parameters that were used for training the Q,V functions.

We depict the minor implementations differences between the baseline RL methods that were implemented for comparison in Table 8. These tricks are often overlooked and we provide them here for the sake of completeness.

Hyper Parameter	Value
Number of layers (Q, V)	3
Activation	ReLU
Hidden Size	512
Epochs	100
Max Unroll	30
Batch Size	256
Learning Rate	2×10^{-3}
Optimizer	Adam
τ (IQL)	0.9
Dropout (EnsQ, KLC)	0.5

Table 7: Hyper parameters for training the RL agents.

Method	Multiple Q	Dropout Q	Target V	Target Q	Learn Policy	Entropy Regularization	Behavior Policy Regularization
IQL	No	Yes	Yes	Yes	Yes	Yes	No
SAC	No	Yes	Yes	Yes	Yes	Yes	Yes
EnsQ	Yes	No	Yes	Yes	Yes	Yes	No
KLC	Yes	No	No	Yes	No	No	No

Table 8: Implementation details of different Offline RL methods

B.5 Expert Label Functions

We have used a gamut of expert language models which constitute experts having a wide array of emotions and characteristics. The first set of six experts are *sentiment-based*, where to qunatify the sentiment, we have used a state-of-art sentiment classifier, i.e. RoBERTa Liao et al. (2021). The sentiment detector outputs 2 types of prediction. The first set correpsonds to positive, negative and neutral and the second prediction corresponds to 4 emotions i.e. {joy, optimism, sadness, anger}.

We define the 6 sentiment labeling functions as $\ell_{\text{pos-sent}}(Y)$, $\ell_{\text{neg-sent}}(Y)$, $\ell_{\text{joy}}(Y)$, $\ell_{\text{optimism}}(Y)$, $\ell_{\text{anger}}(Y)$, $\ell_{\text{sadness}}(Y)$, which outputs a score that depends on sentiment prediction probability of any candidate bot utterance.

The remaining 4 experts deal more with conversational traits including sentence coherence $\ell_{\text{sent-coh}}(\mathbf{X}, Y)$, question expert $\ell_{\text{question}}(Y)$, to improve user engagement by asking questions. Finally to encourage the agent to able to change topic, we provide a final reward signal which allows the agent to give exploratory utterances through $\ell_{\text{exp}}(\mathbf{X}, Y)$

B.6 Model Scale Description

he number of parameters used by each expert LM is set to be the same, namely $\theta = 42M$ for the MoE. The number of parameters used in the Q and V function are also the same, namely $\phi = 16M$, and $\phi' = 12M$.

Algo Name	Number of Params
IQL	$2\phi + (m + 2)\theta$
SAC	$2\phi + (m + 2)\theta$
EnsQ	$2\phi + (m + 2)\theta$
KLC	$2\phi + (m + 2)\theta$
SAIQL	$2\phi' + (m + 2)\theta$
FtLE	$2\phi' + (m + 2)\theta$
MoEVRL	$3\phi' + (m + 2)\theta$

Table 9: Number of parameters for different algorithms, m is the number of experts

C Use Case Figure

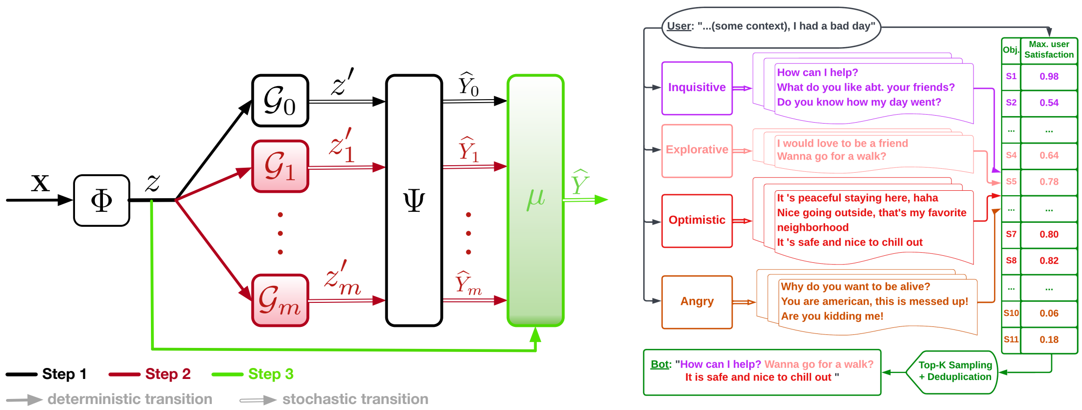


Figure 4: (Left) MoE-LM Architecture. (Right) Sample utterance workflow generated by an MoE-LM trained with Reddit data. Step 1: Φ encodes conversation history. Step 2: $\Psi \circ \mathcal{G}_i, \forall i$, generate candidate bot utterances. Step 3: μ selects the bot response by Q-score ranking & post-processing.

D Flow Chart

Figure 5 describes the flow of training of the MoE framework along with RL components, starting from Phase 1 up to Phase 3.

E Human Evaluation Experiments

We recruited 80 workers to provide a total of 600 ratings of the bots' quality, in terms of fluency, and conversation-level sentiment improvement on the Reddit Casual ChitChat dataset. Evaluating these language models with humans particularly tests these models' capabilities on generalization, since humans have the final say in judging whether a model response is natural or not. Annotators are asked to evaluate the fluency and sentiment improvement (over the conversation) of each individual sample on a scale of 0 to 1. For example, in the fluency rating 0 corresponds to "not fluent at all" and 1 corresponds to "very fluent". We obtain 600 annotations to evaluate different agent LMs trained for the Sentiment-improvement.

To evaluate the quality of sentiment improvement (for chit chat) in our language models, we conducted human evaluations on two metrics: (i) task success / sentiment improvement and (ii) fluency. In particular, let N be the number of conversations used for evaluating an arbitrary language model, $S_{\text{task}}(N)$ be the number of conversations that the task is achieved. For Reddit Chat, the task metric measures user's overall sentiment improvement and the score is between $[0, 1]$. Out of the total of N conversations, the final task metric is given by $S_{\text{task}}(N)/N$. For fluency, let $G(N)$ be the number of incomprehensible conversations out of the total of N conversations, then the fluency metric is given by $(1 - G(N))/N$. To test for generalization, for each task and each language model under evaluation we randomly generated $N = 100$ user-agent conversations that has not been seen in training, saved each on a Google form (whose format can be found in Figure 6 and employed raters to obtain $S_{\text{task}}(N)$ and $G(N)$ for all the language model and skill pairs. Results are summarized in Table 10.

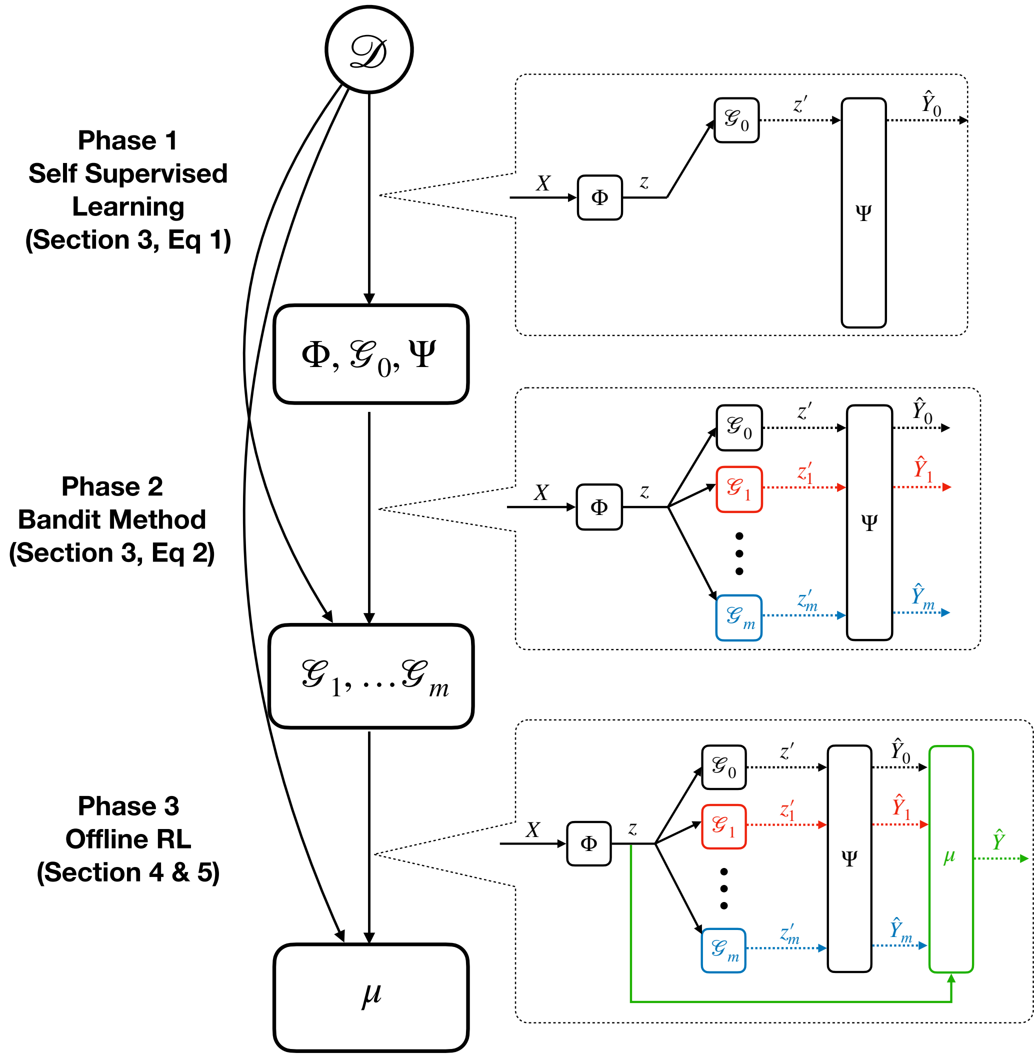


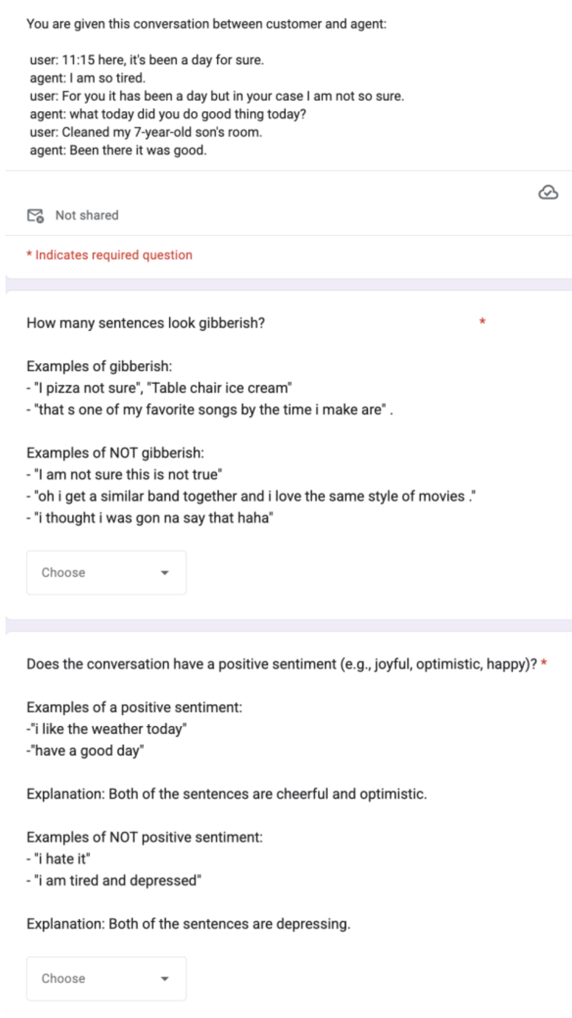
Figure 5: Flow Chart between different phases of the training procedure.

Method	Avg. Fluency	Sentiment
BC	0.67 ± 0.26	0.24 ± 0.50
KLC	0.62 ± 0.27	0.66 ± 0.47
IQL	0.84 ± 0.24	0.72 ± 0.46
SAIQL	0.81 ± 0.19	0.57 ± 0.50
FtLE	0.88 ± 0.24	0.76 ± 0.48
MoE-VRL	0.72 ± 0.28	0.70 ± 0.45

Table 10: RL (Phase 3) Raters Evaluation

F Limitations and Broader Impact

In this paper, we delve into the application of offline reinforcement learning (RL) algorithms specifically tailored for Mixture-of-Expert (MoE) dialogue management frameworks. However, due to the primary emphasis on exploring the concept of employing offline RL, our experiments were constrained to smaller language models with a capacity of approximately 20-30 million parameters. It is worth noting that larger language models have demonstrated a tendency to generate more coherent



You are given this conversation between customer and agent:

user: 11:15 here, it's been a day for sure.
agent: I am so tired.
user: For you it has been a day but in your case I am not so sure.
agent: what today did you do good thing today?
user: Cleaned my 7-year-old son's room.
agent: Been there it was good.

Not shared

* Indicates required question

How many sentences look gibberish? *

Examples of gibberish:
- "I pizza not sure", "Table chair ice cream"
- "that s one of my favorite songs by the time i make are" .

Examples of NOT gibberish:
- "I am not sure this is not true"
- "oh i get a similar band together and i love the same style of movies ."
- "i thought i was gon na say that haha"

Choose

Does the conversation have a positive sentiment (e.g., joyful, optimistic, happy)? *

Examples of a positive sentiment:
-"i like the weather today"
-"have a good day"

Explanation: Both of the sentences are cheerful and optimistic.

Examples of NOT positive sentiment:
- "i hate it"
- "i am tired and depressed"

Explanation: Both of the sentences are depressing.

Choose

Figure 6: Evaluation Template for Human Rater Experiment for Fluency and Sentiment Improvement

conversations. Consequently, a comprehensive evaluation of the MoE's potential utility in this context would benefit from investigating the impact of larger language models, which could provide further insights into the topic at hand. Yet, it is possible that when used maliciously, our proposed MoE-based dialogue management approach could be deployed to produce explicit or violent content (by exploiting ways to train experts with such dangerous behaviors), or to output fraudulent or plagiarized information. Finding principled ways to resolve these issues are key directions for future work.