ArCHer: Training Language Model Agents via Hierarchical Multi-Turn RL

Yifei Zhou [1] **Andrea Zanette** [1] **Jiayi Pan** [1] **Sergey Levine** [1] **Aviral Kumar** [2]

Abstract

A use case of large language models (LLMs) is in goal-directed decision-making tasks (or "agent" tasks), where an LLM needs to make intelligent decisions over a multi-turn interaction to accomplish a task. Reinforcement learning (RL) provides a general paradigm to address such agent tasks, but current RL methods for LLMs largely focus on optimizing single-turn rewards (e.g., PPO for RLHF). By construction, most single-turn RL methods cannot endow LLMs with the ability to perform credit assignment, or reason about their past actions, in multiple turns. How can we design effective and efficient multi-turn RL algorithms for LLMs? In this paper, we develop a framework for building multi-turn RL algorithms for fine-tuning LLMs, that preserves the flexibility of existing single-turn RL methods for LLMs, while accommodating multiple turns, long horizons, and delayed rewards. Our framework builds a hierarchical RL approach and runs two RL algorithms in parallel: a high-level off-policy value-based RL algorithm to aggregate reward over utterances, and a low-level policy gradient RL algorithm that utilizes this high-level value function to train a token policy within each turn. Our hierarchical framework, **A**cto**r**-**C**ritic Framework with a **H**i**e**rarchical Structu**r**e (**ArCHer**), can also give rise to other RL methods. Empirically, we find that ArCHer significantly improves efficiency and performance on agent tasks, attaining a sample efficiency of about **100x** over existing methods, while also improving with larger model capacity (upto the 7 billion scale).

1. Introduction

Owing to their generalist knowledge, large language models (LLMs) have a tremendous potential to address a wide variety of decision-making or "agent" problems that can be expressed in text or natural language, from writing code (Yang et al., 2023a; Li et al., 2022; Lin et al., 2018), navigating the web (Zhou et al., 2023a; Yao et al., 2023a), and using tools (Schick et al., 2023), all the way to interacting with humans (Ghosal et al., 2022; Verma et al., 2022; Jaques et al., 2020). In order to succeed in these domains, an LLM needs to make a *sequence* of intelligent decisions over multiple turns instead of generating the most *probable* text completion at each step that is trained to do in pre-training.

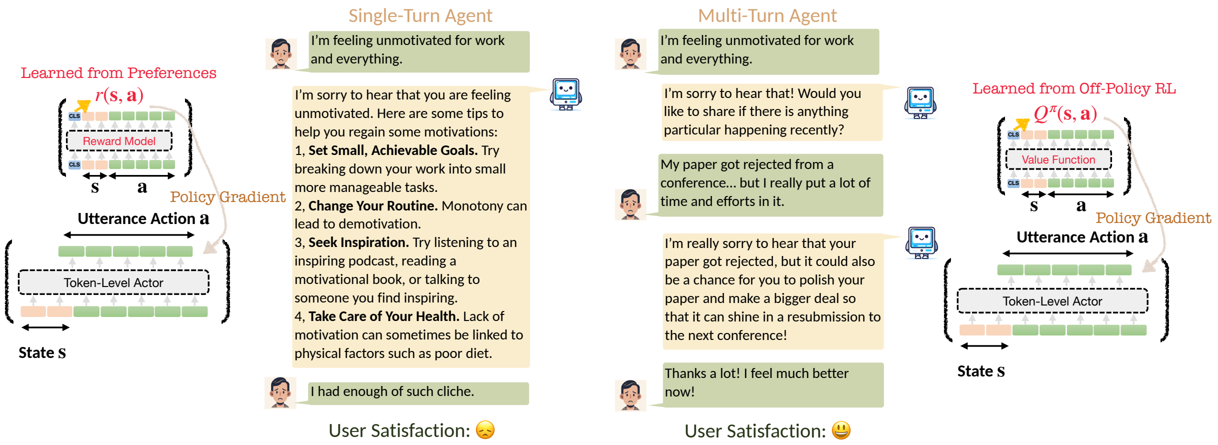
Despite clear use cases in multi-turn agent problems, most methods for eliciting goal-directed behavior from LLMs often rely on myopic objectives that either attempt to mimic successful demonstrations at each step (Zeng et al., 2023; Chen et al., 2023), or optimize for single-turn preferences (Touvron et al., 2023; Ouyang et al., 2022; Bai et al., 2022). Policies trained via single-turn methods often fail to perform effective credit assignment (e.g., they fail to identify good actions that may lead to long-term future performance despite appearing sub-optimal just at a given step) and do not endow policies with information-seeking behavior, which is important in agent problems (e.g., when dealing with a new tool). Therefore, we consider the problem of building multi-turn RL approaches that are able to directly maximize long-term objective of interest (e.g., customer satisfaction at the end of a multi-turn conversation with an LLM assistant), formulated via a scalar reward function.

Unlike single-step RL (Touvron et al., 2023; Ouyang et al., 2022; Bai et al., 2022), training LLMs via multi-turn RL presents a number of unique challenges. First of all, multi-turn RL would require online interaction with external sources such as humans or web servers, which can be slow and expensive. Due to this, on-policy methods such as PPO (Schulman et al., 2017) quickly become impractical owing to their inability to reuse data from past interaction. While off-policy methods circumvent this problem, these methods present other challenges: as the number of tokens increase with multiple turns (often substantially so, due to the bias of LLMs towards producing long responses), token-level methods (Snell et al., 2023; Jaques et al., 2020) that consider individual tokens as actions need to now propagate reward signal over extremely long horizons. This results in extremely slow learning speeds over long horizons, with additional error compounding. For example, token-level ILQL (Snell et al., 2023) takes more than 10 days to converge on a task in our experiments, while filtered behavior cloning takes less than a day. To address long horizon issues,

---

[1] University of California, Berkeley [2] Google Deepmind. Correspondence to: Yifei Zhou <yifei_zhou@berkeley.edu>, Aviral Kumar <aviralkumar@google.com>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).



I'm feeling unmotivated for work and everything.

I'm sorry to hear that you are feeling unmotivated. Here are some tips to help you regain some motivations:
1, **Set Small, Achievable Goals.** Try breaking down your work into small more manageable tasks.
2, **Change Your Routine.** Monotony can lead to demotivation.
3, **Seek Inspiration.** Try listening to an inspiring podcast, reading a motivational book, or talking to someone you find inspiring.
4, **Take Care of Your Health.** Lack of motivation can sometimes be linked to physical factors such as poor diet.

I had enough of such cliche.

I'm feeling unmotivated for work and everything.

I'm sorry to hear that! Would you like to share if there is anything particular happening recently?

My paper got rejected from a conference... but I really put a lot of time and efforts in it.

I'm really sorry to hear that your paper got rejected, but it could also be a chance for you to polish your paper and make a bigger deal so that it can shine in a resubmission to the next conference!

Thanks a lot! I feel much better now!

Figure 1: ***Single-turn RL vs multi-turn RL for LLMs (ours).*** In particular, the single-turn RL agent seeks to resolve the request within a single turn and hence ends up providing as much information as possible in its response. On the contrary, the multi-turn RL agent can execute information-gathering actions and address requests in a targeted manner over turns. While current single-turn RL methods for LLMs abstractly use some form of policy gradients computed using a reward model to train the LLM policy, the method proposed in our paper extends this paradigm to the multi-turn setting by now replacing the reward model with a learned value function, which is trained with off-policy reinforcement learning.

in principle, one can treat the entire utterance for each turn as an action (Verma et al., 2022; Jang et al., 2022), but this comes at the cost of introducing an enormous, variable-size action space, presenting a challenge for off-policy methods based on temporal-difference (TD) learning that require maximization over the action at each time step. This necessitates a multi-turn RL framework that attains the right balance in terms of the aforementioned challenges.

In this paper, we devise a framework for building multi-turn RL algorithms that attains this kind of a sweet spot. Our key insight is that a **hierarchical approach** for RL with language models that addresses the challenges with both on-policy and off-policy RL approaches as outlined above. Specifically, our framework prescribes an off-policy temporal difference learning method for training an utterance-level value function at the high level, and any on-policy policy gradient algorithm for optimizing the token generation at each turn of the interaction at the low level, treating the high-level value function as the terminal reward for that turn. Unlike on-policy methods, this allows for sample reuse and faster convergence, while avoiding Bellman backups over individual tokens or maximization over enormous action spaces, as the high-level critic is trained at a coarser time-scale, on tokens produced by the actor. In addition, it also directly inherits implementation details from existing token-level RL algorithms developed for single-turn RL with preferences, for training the policy. This way we are able to obtain the best of both utterance-based and token-based, and off-policy and on-policy approaches for training LLMs.

Our main contribution is a framework for developing hierarchical RL approaches for LLMs, that we call: **A**ctor-**C**ritic framework with a **Hie**rarchical Structu**r**e (or **ArCHer** in short). We study several concrete algorithmic instantations derived from the ArCHer framework by conducting experiments on a range of language "agent" tasks with active data collection (i.e., the "online" setting). We find that algorithms derived from ArCHer are 100x more sample efficient than on-policy methods such as PPO, and converge to a better performance than off-policy methods. Moreover, our methods are easy to build on existing single-turn RL methods and scale to different transformer architectures and more parameters (we show effectiveness of our approach up to the 7 billion scale), directly enabling plug-and-play choices of RL algorithms and models. [1]

2. Related Work

Single-turn RL for LLMs. Most prior works that use RL for LLMs study problems where the language model must produce a single decision, with no further steps of interaction with an external environment (e.g., the "single-turn" preference optimization setting (Casper et al., 2023; Christiano et al., 2023; Ziegler et al., 2019)). Typical algorithms used for this sort of single-turn RL include policy-gradient methods such as PPO (Ouyang et al., 2022; OpenAI et al., 2023; Ramamurthy et al., 2023), A2C (Glaese et al., 2022), offline optimization methods such as DPO (Rafailov et al., 2023), and filtered supervised learning approaches (Yuan et al., 2023; Korbak et al., 2023; Gulcehre et al., 2023; Dong et al., 2023). Despite promising results, there remain many important agent problems that cannot be solved in a single-turn setting. Many of these problems require the agent to explicitly take the steps to gather information before making a decision, such as asking for personalized preferences before making a travel plan (Hong et al., 2023) or initially attempting to read the help manual of the shell in a linux terminal, before carrying out the requested tasks (Liu et al.,

---

[1] The project page is https://yifeizhou02.github.io/archer.io/ and code can be found at https://github.com/YifeiZhou02/ArCHer.

2023). Single-step approaches cannot learn such nuanced strategies, necessitating multi-turn RL for training LLMs.

Training language agents without RL. Motivated by few-shot learning and reasoning abilities of LLMs, prior works also utilize LLMs for sequential decision-making via prompt engineering. For example, ReAct (Yao et al., 2023b) and Reflexion (Shinn et al., 2023) prompt the LLM to "think" and analyze past failures before executing the next action. Voyager (Wang et al., 2023) prompts the LLM agent to develop and refine a curriculum and action library based on environment input. However, without updating the parameters of the LLM, the effectiveness of these methods is inherently limited by the intrinsic capabilities obtained from pre-training (Zeng et al., 2023; Chen et al., 2023). Even state-of-the-art models such as GPT-4 with in-context learning can perform very sub-optimally in out-of-distribution settings (Liu et al., 2023; Yang et al., 2023b) without updating the model. To improve over pre-trained capabilities, another line of work finetunes LLMs with successful trajectories (generated manually or by rolling out a strong pre-trained LLM) (Schick et al., 2023; Zeng et al., 2023; Chen et al., 2023). However, manual labels and tool call annotations are be expensive to obtain. Moreover, it would be prohibitively expensive for automated approaches to stumble upon successful rollouts will as the task horizon increases (Liu et al., 2023; Abdulhai et al., 2023). Therefore, in this paper, we sidestep these problems by directly maximizing the objective of interest via RL.

Multi-turn RL for LLMs. While many prior works directly use off-the-shelf policy-gradient methods, such as PPO (Schulman et al., 2017; Szot et al., 2023; Yao et al., 2023a) and REINFORCE (Sutton et al., 1999; Williams, 2004; Ranzato et al., 2015; Wu & Hu, 2018; Paulus et al., 2017) to train LMs, these methods can become sample inefficient in multi-step settings that require interaction with an external environment (Verma et al., 2022; Jang et al., 2022). To address such sample complexity issues, off-policy and offline value-based methods learn from existing static data (Snell et al., 2023; Jaques et al., 2020; Verma et al., 2022; Jang et al., 2022). However, existing off-policy methods for multi-turn language tasks either **(1)** consider a single token as an action (i.e., "token-level") (Snell et al., 2023; Jaques et al., 2020) and must deal with long horizons, or **(2)** consider an utterance as a single action, but utilize multiple candidate utterances from a frozen pre-trained LLM for maximization in the Bellman backup (Verma et al., 2022; Jang et al., 2022), reducing the pace of policy improvement, as we also observe. Our approach addresses both limitations.

Hierarchical RL. Within the field of hierarchical RL (outside of LLMs), there are several algorithmic approaches. These approaches can be grouped into methods based on learning options (e.g., Option-critic (Bacon et al., 2016), Semi-MDP for options (Sutton, 1998), DADS (Sharma et al., 2019)) and hierarchical goal-conditioned RL (e.g., HIQL (Park et al., 2024), HIRO (Nachum et al., 2018), Feudal networks (Vezhnevets et al., 2017), and LEAP (Nasiriany et al., 2019)), which can be regarded as a specific instantiation of the options framework for the class of goal-reaching problems. Hierarchical goal-conditioned RL approaches train the high-level policy to propose intermediate states or waypoints, or more generally, latent task embeddings for the low-level to solve.

3. Actor-Critic Framework with a Hierarchical Structure (ArCHer)

Existing RL methods that consider an individual token as an action suffer from very long horizons over multiple turns. Utterance-level RL methods avoid this challenge, but now they must tractably maximize over a coherent set of tokens within an utterance. To address these issues, we will develop a class of hierarchical RL methods, ArCHer, that bypass these challenges by running two RL algorithms, in parallel.Of course, RL algorithms running at two levels in this hierarchy do share information, but this is done in a way to still prevent challenges with previous methods. We start by describing how multi-turn language generation can be posed as a hierarchical Markov decision process (MDP), followed by building RL methods in this hierarchical MDP.

3.1. Language Generation as a Hierarchical MDP

To build multi-turn approaches, we will first present a formulation of language generation as acting in a hierarchical MDP. Our construction defines a *high-level* MDP, and a *low-level* MDP embedded inside the high-level MDP. For both MDPs, states are a variable-length sequence of tokens. For the high-level MDP, an action is defined as a sequence of tokens. An action in the low-level MDP is a single token, such that executing a sequence of actions in the low-level MDP corresponds to a single action in the high-level MDP. Formally, each state s_t in the high-level MDP consists of an interaction history between the LLM and the external environment, and each action a_t in this MDP is a variable-length sequence of tokens. The low-level MDP models the generation of the high-level action produced by the agent *within* a single high-level action, where each low-level action a_t^h is an individual token (i.e., the h-th token in the t-th high-level action). A state in this low-level MDP is obtained by concatenating a high-level state s_c consisting of the interaction history until this turn and $a_t^{1:h-1}$, the history of individual action tokens produced within the current turn before step h. The next state is obtained by concatenating the current action to this token history. Figure 2 shows a visualization of our notations for the hierarchical MDP.

Policy optimization in the high-level MDP aims to maximize task reward, whereas a policy in the low-level MDP attempts to find a sequence of up to L tokens $a_t^{1:L}$ that maximizes reward equal to the value function of the high-level MDP $Q^\pi(s, a_t^{1:L})$, provided at the end of the low-level rollout. A rollout in the low-level MDP ends as soon as the policy ends up choosing to produce an "EOS" token.

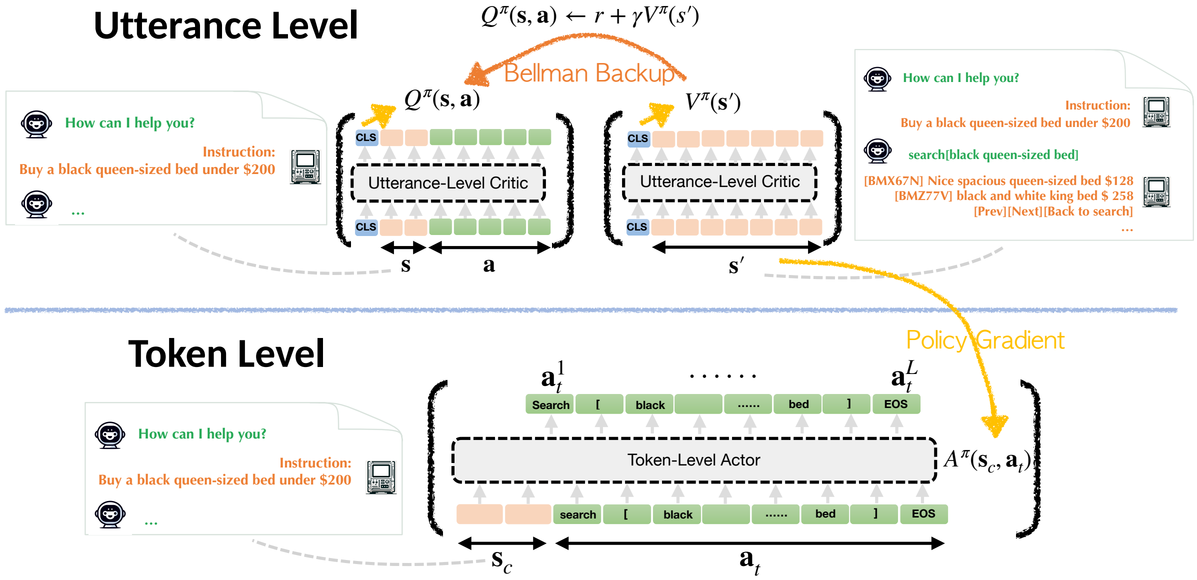

Utterance Level
$Q^\pi(\mathbf{s},\mathbf{a}) \leftarrow r + \gamma V^\pi(s')$
Bellman Backup
$Q^\pi(\mathbf{s},\mathbf{a})$
$V^\pi(\mathbf{s}')$
How can I help you?
Instruction:
Buy a black queen-sized bed under $200
...
CLS
Utterance-Level Critic
s
a
s′
search[black queen-sized bed]
[BMX67N] Nice spacious queen-sized bed $128
[BMZ77V] black and white king bed $ 258
[Prev][Next][Back to search]
Policy Gradient
Token Level
$\mathbf{a}_t^1$
$\mathbf{a}_t^L$
Search [black ...... bed] EOS
Token-Level Actor
$A^\pi(\mathbf{s}_c, \mathbf{a}_t)$
search [black ...... bed] EOS
$\mathbf{s}_c$
$\mathbf{a}_t$

Figure 2: ***Schematic of the practical instantiation of Actor-Critic Framework with a Hierarchical Structure (ArCHer).*** Our algorithm operates both at the utterance level and the token level. At the utterance level, our algorithm learns a Q-function, via Bellman bootstrapping with TD errors. At the token level, the policy is learned by maximizing the advantage function induced by the utterance-level Q-function using a policy gradient approach, where this advantage estimate is provided as a reward at the end of the sequence of tokens appearing within the utterance.

A concrete and natural instantiation of this hierarchical framework is when each turn or an "utterance" corresponds to a single time step in the high-level MDP, and each token within a turn is an action in the low-level MDP. In other words, this construction chooses to use the utterance-level MDP (Figure 2) at the high level and the token-level MDP at the low level. For example, in the context of a web agent, the state would be the interaction of web pages visited so far and a high-level action would be an utterance, e.g., "search[queen-sized bed, black]". A candidate reward function would be +1 if the correct item can be bought and 0 otherwise. The dynamics would involve the web engine reacting to an action. Each token of an utterance (e.g., "search[queen-sized bed, black]") would be an individual action in the embedded token-level MDP, for example, an action would be individual tokens "search", "[", "queen".

3.2. Preliminaries: Reinforcement Learning Definitions

To build our approach, we first provide a few standard definitions. The Q function of a policy π is the long-term reward obtained by executing a certain action at the current step, followed by executing π thereafter: $Q^\pi(s_h, a_h) = \mathbb{E}_\pi\left[\sum_{t=0}^{\infty} \gamma^t r(s_{h+t}, a_{h+t})\right]$. The value function $V^\pi(s_h)$ is given by taking an expectation of the Q-value, $Q^\pi(s_h, a_h)$, under actions a_h sampled from the policy π. The advantage $A^\pi(s_h, a_h)$ of a state-action pair is the difference between its Q-value and the value of the state under the policy: $A^\pi(s_h, a_h) = Q^\pi(s_h, a_h) - V^\pi(s_h)$. We will denote the value function in the low-level MDP as $\widetilde{V}(s_c, a_h^{1:i-1})$.

3.3. RL Algorithms in the Hierarchical Language MDP

This hierarchical MDP provides flexibility in designing multi-turn RL algorithms: we could use any choice of RL algorithm for the high or the low level. That said, note that only the high level requires interaction with a (non-differentiable) environment, while the low level optimizes against the high-level value function, and therefore trains entirely "in silico," without any interaction with an environment. Therefore, the requirements on these methods are different: the high-level algorithm should be highly sample efficient, while the low-level algorithm should enable easy policy optimization. Hence, a convenient choice is to use TD learning at the high level, while using policy gradient, akin to single-turn preference optimization approaches (Ouyang et al., 2022; Bai et al., 2022) at the low level.

3.4. A Practical Instantiation of ArCHer for Sample-Efficient Online RL

For deriving a concrete practical algorithm, we will utilize the natural hierarchical MDP induced in multi-turn language interaction: the utterance-level MDP at the high level and the embedded token-level MDP at low level, as discussed at the end of Section 3.1. In this setting, our approach would train an utterance-level critic with TD backups and a token-level policy with policy gradients.

High-level utterance critic. Following the practice in prior RL algorithms (Snell et al., 2023), we train two LLM models at the high-level, one to represent the utterance-level Q-function $Q^\pi_\theta(s, a)$, and one to represent the utterance-level value function, $V^\pi_\psi(s)$. The Q-model is trained on Bellman targets computed from a delayed copy of the value-model. And the value model, in turn, is trained to approximate the expected value of the Q-model on token sequences (i.e., utterances) obtained by sampling autoregressively from the low-level policy, π_ϕ. Due to the off-policy nature of this training process, we store and train on data from all previous online interactions $\mathcal{D} = \{s_i, a_i, r_i, s'_i\}_{i=1}^N$. The objective for training the Q-model and the value-model are given by:

$$J_Q(\theta) = \mathbb{E}_{s,a,r,s'\sim\mathcal{D}}\left[(Q_\theta(s,a) - r - \gamma V_{\bar{\psi}}(s'))^2\right]. \quad (1)$$

$$J_V(\psi) = \mathbb{E}_{s\sim\mathcal{D}}\left[\mathbb{E}_{a\sim\pi_\phi(\cdot|s)}\left[(V_\psi(s) - Q_{\bar{\theta}}(s,a))^2\right]\right]. \quad (2)$$

We estimate Equation 2 by sampling a batch of n observations $\{s_i\}_{i=1}^n$, followed by auto-regressively sampling token sequences from the actor $\{a_i^L\}_{i=1}^n$. The delayed target models $Q_{\bar{\theta}}$ and $V_{\bar{\psi}}$ are updated towards their current counterparts with Polyak averaging (Haarnoja et al., 2018).

Low-level token actor. At the low-level, we train the token-level actor $\pi_\phi(\cdot|s_c, a_t^{1:h})$ via an on-policy policy gradient approach to find a sequence of tokens that maximizes the prediction of the Q-model. To reduce variance, we use advantage values derived from the Q-model as the terminal reward. This subroutine for training the actor generalizes single-turn RL methods from RLHF, except that the terminal reward is now an estimate of the multi-turn advantage instead of a reward model. Concretely, we update the token-level policy with the policy gradient computed via REINFORCE (Williams, 2004):

$$J_\phi(\pi) = \mathbb{E}_{s_c\sim\mathcal{D}, a_t^{1:L}\sim\pi(\cdot|s_c)}\left[\sum_{i=1}^{L} A(s_c, a_t^{1:L}) \log \pi_\phi(a_t^i|s_c, a_t^{1:i-1})\right]. \quad (3)$$

3.5. Other Practical Instantiations of ArCHer

The instantiation of ArCHer described in Section 3.4 is simple and ready-to-use, but the flexibility of the ArCHer framework also enables it to incorporate other components that prior RL works have found to be successful. We describe two variants in the two paragraphs to follow. The first improves the sample efficiency while interacting with the external environment and the second enables ArCHer to learn entirely from a dataset of pre-collected experience.

Improvements to token-level policy gradient. In applications where the horizon of the token-level MDP is long (i.e. each utterance has many tokens), despite the use of advantage values (instead of Q-model predictions directly), the REINFORCE estimator corresponding to Equation 3 can struggle to improve the policy due to a high variance in token-level reward (Schulman et al., 2015). This variance can be reduced by introducing a baseline value function, $\widetilde{V}_\eta(\widetilde{\pi})$ parameterized by η, in the token-level MDP. For simplicity, we opt to train this token-level baseline via supervised regression onto Monte-Carlo return estimates (with a discount factor of 1.0) in the token-level MDP as shown in Equation 4, though Bellman backups can also be employed in the low-level MDP to estimate it:

$$J_\eta(\widetilde{V}) = \mathbb{E}_{s_c\sim\mathcal{D}, a_t^{1:L}\sim\pi(\cdot|s_c)}\left[\sum_{i=1}^{L}\left(A(s_c, a_t^{1:L}) - \widetilde{V}_\eta(s_c, a_t^{1:i-1})\right)^2\right]. \quad (4)$$

Incorporating this token-level baseline, the new objective for the actor given by Equation 5:

$$J_\phi(\pi) = \mathbb{E}_{s_c\sim\mathcal{D}, a_t^{1:L}\sim\pi(\cdot|s_c)}$$
$$\left[\sum_{i=1}^{L}\left(A(s_c, a_t^{1:L}) - \widetilde{V}_\eta(s_c, a_t^{1:i-1})\right) \cdot \log \pi_\phi(a_t^i|s_c, a_t^{1:i-1})\right]. \quad (5)$$

Offline RL training with ArCHer. ArCHer can also learn from a dataset of pre-collected experience without any online interaction. A distinguishing aspect of the offline setting is that improving the policy normally results in selecting out-of-distribution actions (Kumar et al., 2019), whose Q values are difficult to estimate accurately given only the offline dataset. As a result, directly optimizing the Q values, as in the online setting, often results in severe overestimation, divergence, and poor policy performance. This suggests that we need to adopt different objective functions in this offline setting. One concrete instantiation is to utilize the implicit Q-learning (IQL) (Kostrikov et al., 2021) algorithm for obtaining backup targets for the utterance-level critic restricted to in-support actions, and the AWR (Peng et al., 2019) algorithm for imposing a penalty on deviating far away from the data on the actor. While our offline RL experiments utilize these design choices, one could also utilize other techniques such as explicitly regularizing the critic's predictions (Kumar et al., 2020) or imposing a behavioral cloning loss on the actor (Fujimoto & Gu, 2021).

The IQL loss aims to derive a version of the TD error that aims to inherit characteristics of the Bellman optimality operator but without performing an explicit maximization over the actions, by instead regressing Q-functions towards a higher expectile of possible target values at the next state. For a given expectile parameter $\tau \in [0.5, 1)$, the IQL loss is given by the following:

$$J_\psi^{\text{IQL}}(V) = \mathbb{E}_{s\sim\mathcal{D}}[\mathbb{E}_{a\sim\pi_\phi(\cdot|s)}[L_2^\tau(V_\psi(s) - Q_{\bar{\theta}}(s,a))]], \quad (6)$$

where $L_2^\tau(u) = |\tau - \mathbf{1}\{u < 0\}|u^2$. See the paper (Kostrikov et al., 2021) for more details.

The policy extracted by AWR (Peng et al., 2019) trades off between searching for high-return policies and imitating the policy that generated the dataset, by minimizing the loss

$$J_\phi(\pi) = -\mathbb{E}_{(s_c, a_t^{1:L})\sim\mathcal{D}} \left[\exp\left(\beta \cdot A(s_c, a_t^{1:L})\right) \cdot \sum_{i=1}^{L} \log \pi_\phi(a_t^i | s_c, a_t^{1:i-1})\right]. \quad (7)$$

The tradeoff is controlled by a positive, user-defined scalar value β. Low values for β encourage imitating the policy that generated the dataset. Large values of β lead to more aggressive maximization of rewards, but potentially at the cost of stability (Peng et al., 2019). The results for the offline alternative is shown in Appendix D.1.

4. Theoretical Analysis

We will present empirical results showing the effectiveness of ArCHer in Section 5, but in this section we will first highlight some theoretical and conceptual benefits of our hierarchical design. An important difference between ArCHer and prior token-level RL algorithms such as ILQL (Snell et al., 2023) is the high-level critic. Thus, we aim to understand the impact of estimation errors in this high-level critic on the token-level policy in contrast with the impact of estimation errors in a token-level critic. While our proof techniques can be easily applied to general high-level and low-level MDPs, we focus on analyzing the specific contrast of utterance and token-level critics to be concrete.

Conditions for convergence. To start, we show that a hierarchical RL design prescribed by ArCHer requires substantially weaker conditions for algorithm convergence compared to off-policy token-level methods. These conditions pertain to **(1)** the capacity of the function class representing the critic (i.e., Bellman completeness (Song et al., 2023; Zhou et al., 2023b; Zanette, 2023; Xie et al., 2021)), and **(2)** the coverage of off-policy data as measured by the density ratio (Zhan et al., 2022; Foster et al., 2021), following the practice standard in RL theory. For **(1)**, we show in Lemma 1 that satisfying the Bellman completeness (Song et al., 2023; Zhou et al., 2023b; Zanette, 2023; Xie et al., 2021) condition imposes weaker requirements on the function class used to model the critic at the utterance-level (as is the case with ArCHer) as opposed to the token level. Intuitively, this is because a function class representing the token-level critic must exhibit flexibility to realize arbitrary functions at the next token, which would require higher capacity compared to a utterance-level critic that only needs to be able to realize arbitrary functions at the coarser time-scale of utterances.

For **(2)**, we show in Lemma 2 that the density ratio condition (Zhan et al., 2022; Foster et al., 2021) imposes identical requirements on the coverage of the offline data for token-level and utterance-level critic, despite a larger space of possible utterances (i.e., actions in the higher-level MDP). Intuitively, this is because the a given offline dataset induces the same trajectory distribution at both the utterance and token levels. In other words, if a trajectory is covered by the offline data at the utterance level, it is also covered by the offline data at the token level, and vice versa.

Statistical error analysis. With the convergence conditions discussed above, we are able to establish an analysis of the statistical error incurred in estimating the advantage estimates using the utterance-level and token-level critics. Intuitively, this means that an utterance-level critic provides a much more correct signal for improving the policy. We state the conclusions of the theorem below, and refer interested readers to Appendix G for a formal proof.

Theorem 1 (Main Theorem; Informal)**.** *For an utterance-level MDP with discount factor γ, where L is the maximum length of each utterance, suppose utterance-level Assumption 1 and 2 holds, let f be the final Q-function returned by fitted policy evaluation formalized in Algorithm 2 at the utterance level, f yields a suboptimality gap of*

$$\mathbb{E}_{s,a\sim d^\pi}\left[\left((\bar{f}(s,a) - \mathbb{E}_{a'\sim\pi(\cdot|s)}[\bar{f}(s,a)]) - A^\pi(s,a)\right)^2\right]$$
$$\leq \frac{1}{\gamma L^{1/2}}\mathcal{O}\left(\frac{1}{(1-\gamma)(1-\gamma^{1/L})L^{1/2}}\left(\epsilon_{stat} + \sqrt{\epsilon_{stat}}\right)\right).$$

For an equivalent token-level MDP with discount factor $\gamma^{1/L}$, suppose token-level Assumption 1 and 2 holds, let f be the final Q function returned by Fitted Policy Evaluation formalized in Algorithm 2 at the token level, f yields a suboptimality gap of

$$\mathbb{E}_{s,a\sim\widetilde{d}^\pi}\left[\left((\bar{f}(s,a) - \mathbb{E}_{a'\sim\pi(\cdot|s)}[\bar{f}(s,a)]) - \widetilde{A}^\pi(s,a)\right)^2\right]$$
$$\leq \mathcal{O}\left(\frac{1}{(1-\gamma)(1-\gamma^{1/L})L^{1/2}}\left(\epsilon_{stat} + \sqrt{\epsilon_{stat}}\right)\right),$$

where ϵ_{stat} is the statistical error, proportional to $N^{-1/2}$ (N is the number of utterance-level transitions).

Informally, Theorem 1 shows that the error in estimating advantages using the token-level critic is $\gamma\sqrt{L}$ larger than the the utterance-level critic (in the worst case), where L is the maximum number of tokens in each utterance, due to error accumulation. In practice, a common choice for γ is greater than 0.95 while L can be as large as 64 tokens, resulting in $\gamma L^{1/2} >> 1$. Therefore, we have not only shown that a hierarchical design requires weaker conditions for convergence, but it also enjoys improved guarantees on the statistical error, resulting in a more accurate estimate of policy gradient for improving the policy in the worst case.

5. Experiments

The goal of our experiments is to evaluate the efficacy of hierarchical RL algorithms derived from ArCHer. Specifically, we aim to answer the following questions: **(1)** Is ArCHer able to achieve better sample complexity and performance than prior on-policy and off-policy RL methods for LLMs? **(2)** Does the TD-learning design for the utterance-level critic enable an effective use of off-policy data? **(3)** How does the performance of ArCHer scale with larger base models (such as Mistral 7B (Jiang et al., 2023))? **(4)** How do different practical algorithms derived from our ArCHer framework compare? To answer these questions, we will present an extensive empirical evaluation of ArCHer and several prior methods on a suite of environments encompassing natural language games, navigation problems posed in natural language, and interaction with the web.

5.1. Tasks and Environments

To stress-test ArCHer, we need environments and task setups that satisfy several desiderata. First, the chosen tasks must require strategic multi-step planning and reasoning under delayed rewards, and cannot be solved in one turn. We also want these tasks to require LLMs to generate coherent natural language and keep the task realistics.

Next, we want these tasks to be solvable by models of upto 7 billion parameter scale, which corresponds to the upper end of our computational budget. Finally, the chosen tasks should support fast and reliable evaluations, for reproducibility and benchmarking. Most existing LLM agent tasks such as those which require interacting with terminals, operating systems, and databases (Yang et al., 2023a; Liu et al., 2023) require larger base models (typically larger than 7B) for obtaining non-trivial success rates and can often be solved in a single step. This makes these tasks unfavorable for fast iteration within our compute budget. Other dialogue and tutoring tasks (Hong et al., 2023; Verma et al., 2022; Zhu et al., 2020; Lee et al., 2019; Budzianowski et al., 2018) require either costly user studies or evaluate using metrics that do not directly represent task performance, making them unfavorable for stress-testing our approach. Therefore, we utilize a different set of tasks for our evaluations.

Concretely, we consider the following tasks: **(1) Detective Game** (Hausknecht et al., 2019), an interactive text fiction game where the agent must generate a sequence of natural language actions (e.g.,“take paper”, “eat apple”) based on environment feedback. A reward is given if the agent reaches some milestones towards finishing the game, where the end goal is to successfully find the murderer in a murder mystery; **(2)Twenty Questions** (Abdulhai et al., 2023), a dialogue task where the agent plays the role of a guesser trying to guess a hidden word from a list of 157 words within twenty yes/no questions. The oracle answers the questions with “Yes.”, “No.”, or “Invalid Question.” The oracle is simulated with a “flan-t5-small” (Chung et al., 2022) model trained with supervised fine-tuning on the dataset provided by Abdulhai et al. (2023). Upon guessing the correct word, the agent receives a reward of 0 and the environment terminates. Otherwise, a reward of -1 is provided at each time step. We also study a variation of this task with a list of only 10 possible underlying words, that we call **Twenty Questions Subset**. This variant challenges the algorithms to tailor a very specific strategy when there is a shortcut in the task; **(3) Guess My City** (Abdulhai et al., 2023), a similar multi-turn task where the agent attempts to guess the name of a hidden city from a list of 100 cities within twenty questions. A crucial difference between the Guess My City task and the Twenty Questions task is that the guesser is now allowed to ask any question and can observe free-form responses (which are not necessarily “Yes” or “No”); **(4) WebShop** (Yao et al., 2023a), a tool-use task where the agent is instructed to buy an item from a shopping server. A dense reward between 0 and 1 is provided based on the similarity of the item purchased and the item requested. See Appendix A for more details. These tasks require planning over long horizons, allow non-trivial success rates within the 7 billion parameter scale, and come equipped with reproducible and task-directed evaluation metrics.

5.2. Comparisons and Baseline Approaches

We compare our method to a number of alternative RL approaches. For token-level methods, we consider token-level **PPO** (Schulman et al., 2017) due to its state-of-the-art performance with LLMs (Ouyang et al., 2022; Ramamurthy et al., 2022). For each iteration, **PPO** collects new on-policy data by rolling out the current actor and uses these data to estimate the policy gradient. Perhaps most importantly, data collected by previous policies is simply discarded for later updates. We use the existing PPO implementation by Abdulhai et al. (2023). We also implemented **token-level DQN** (Mnih et al., 2013), but were unable to get it to attain non-zero task performance and we did not find any prior work evaluating this method so omit it from our results.

We also considered a non-RL approach based on filtered behavioral cloning (BC), denoted as **Filtered BC**. As prior work (Abdulhai et al., 2023; Snell et al., 2023) shows, perhaps surprisingly, this simple baseline often attains competitive or better performance than RL approaches for LLMs, implying that outperforming filtered BC is a hallmark of proper functioning of the “RL component” in an approach. Our implementation of filtered BC maintains a buffer of recent rollouts and trains the actor on top 10% rollouts.

For utterance-level methods, we compare with a state-of-the-art utterance-level RL method, **CHAI** (Verma et al., 2022). CHAI was designed for offline RL specifically. To extend it to learn from online rollouts, we simply replace the pessimistic loss function (i.e., a conservative Q-learning (Kumar et al., 2020) loss) in this approach with a standard TD-learning loss function on data sampled from the off-policy replay buffer, identical to the one used in ArCHer. That said, the key difference is that CHAI utilizes a frozen actor obtained by behavioral cloning (or supervised fine-tuning) on the replay buffer, whereas ArCHer optimizes the actor as well. Each time when an action needs to be sampled, k utterances are sampled from the frozen actor. The utterance-level critic in CHAI ranks these k utterances and chooses the utterance with the highest Q value. A larger value of k would likely lead to better performance, but is also com-

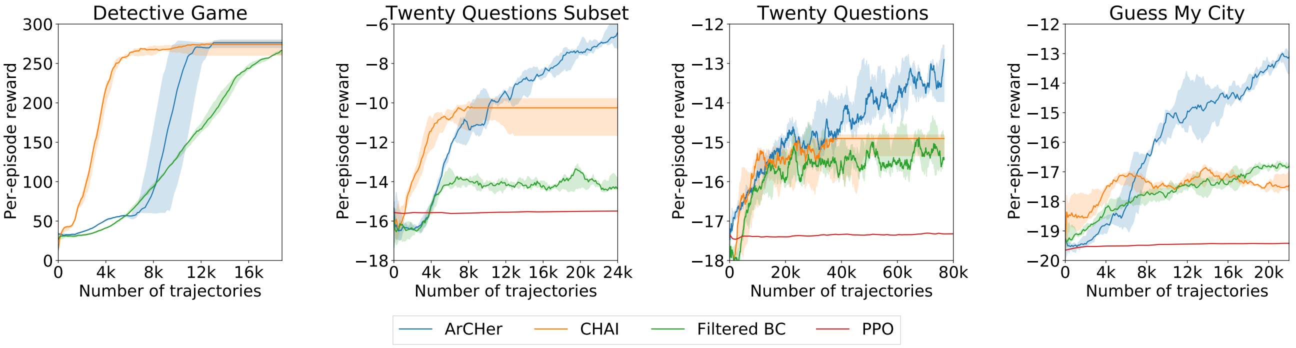


Figure 3: ***Online RL results*** comparing ArCHer and other approaches on four tasks. We plot the median performance of each method across three seeds. Observe that ArCHer steadily improves the policy, outperforming all other methods on three tasks and matching the best prior approach on the simple Detective Game task. While PPO appears to not be learning, by zooming into the learning curve in Figure 7, we find that PPO still gradually improves but at a very slow speed.

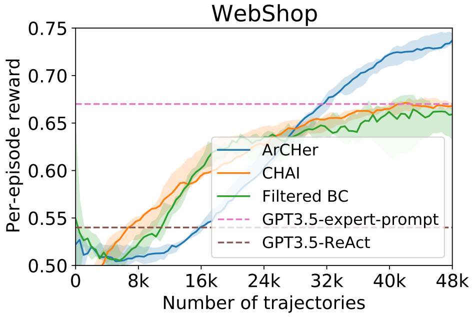


Figure 4: ***Webshop results.*** Observe that fine-tuning a GPT2 base model with ArCHer outperforms prior approaches, filtered BC and CHAI, and is the only approach to outperform GPT 3.5 equipped with several effective prompting strategies.

putationally expensive for this method. To obtain a sweet spot, we instantiate CHAI with $k = 5$ to effectively balance computational overhead and performance (**note that $k = 5$ already results in a runtime of about 4 times longer than our approach for this prior method**; going beyond would be prohibitive for our computational resources). Model architectures, learning rates, and other algorithm-agnostic details are kept identical between all prior methods and ArCHer (see Section A.1).

Finally, akin to single-turn RL fine-tuning of LLMs, we initialize the token-level policy for all methods with a policy checkpoint obtained by running supervised instruction tuning on sub-optimal data for the task (see Appendix A for how this sub-optimal data is generated). Uniformly across all methods, this initialization enables effective exploration at the beginning of RL fine-tuning.

5.3. Results: Sample Efficiency in the Online Setting

Figure 3 and 4 show the comparison between ArCHer with other methods across the five tasks. We also provide some example rollouts of ArCHer for each environment in Appendix E. Overall, we found that ArCHer converges to a much better performance than all other prior methods on the four harder tasks that require identifying hidden information or present diverse initial states (i.e., Twenty Questions Subset, Twenty Questions, Guess My City, and WebShop). In fact, on WebShop, online RL training of GPT2 base model via ArCHer outperforms several effective prompting strategies (i.e., an expert-written prompt and ReAct (Yao et al., 2023b)) applied on top of GPT-3.5, a strong LLM.

First, we found that token-level PPO fails to achieve performance competitive with all other off-policy methods using the same amount of data. This is perhaps unsurprising, as PPO is an on-policy method, and therefore, cannot effectively reuse samples from past iterations. In particular, on the **Twenty Questions** task, we observed that PPO could only stably improve when provided with at least 1024 on-policy rollouts for each gradient step, likely because of high gradient variance. This observation corroborates the finding of Abdulhai et al. (2023), suggesting that online PPO is less practical for this task. Quantitatively, we find that while it takes more than 100k samples for PPO to attain an average return just higher than -17 (see Figure 7 in the appendix), ArCHer attains this reward with fewer than 1000 samples, implying at least a **100x boost** in sample efficiency.

While filtered BC generally converges very quickly, the resulting policy often performs suboptimally and does not improve with more data in Figure 3. On the other hand, ArCHer enjoys steady policy improvement as it collects more samples. Finally, we observed that while CHAI improved at a faster than ArCHer initially, it often converged to a worse final performance. We suspect this is because the critic in CHAI is directly used to rerank samples from a frozen behavior policy, which only enables a narrow margin for policy improvement. On the contrary, ArCHer needs an initial learning phase to reduce critic estimation error, after which it can improve steadily.

5.4. Ablation Study: Importance of Off-Policy Data

In Figure 5 (b), we investigate the importance of off-policy data by varying the size of replay buffer on the **Guess My City** task. A smaller buffer means that updates rely on repeatedly sampling on-policy data. In our experiments, we observed that using a replay buffer containing only the most

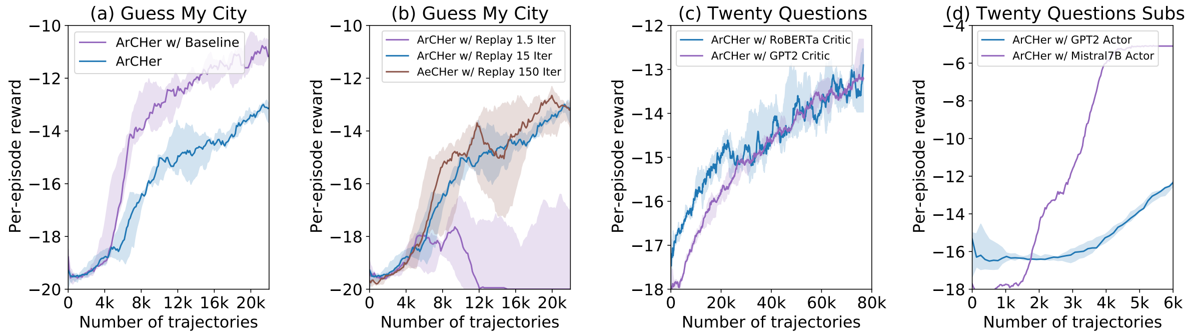


Figure 5: **(a)** Ablation study of the token-level baseline on Guess My City. **(b)** Ablation study of the importance of off-policy data by varying the size of the replay buffer on Guess My City. **(c)** Ablation study of changing the base model for critic from encoder-only RoBERTa to autoregressive decoder-only GPT2 on Twenty Questions. **(d)** Ablation study of scaling the base model for the actor from GPT2 to Mistral7B on Twenty Questions Subset.

recent 48 rollouts resulted in unstable learning, likely due to overfitting on limited data, which has been observed in standard RL problems outside of LLMs (Nikishin et al., 2022). On the other hand, larger buffers are more stable. However, increasing the size of the buffer beyond a certain point is benign, resulting in no meaningful changes to performance. Overall, this means that making use of off-policy data can improve the stability and performance of practical methods.

5.5. Ablation Study: Alternate Base Models for the High-Level Critic in ArCHer

In Figure 5 (c), we changed the architecture for the critic model from an encoder-only RoBERTa (Liu et al., 2019) to an autoregressive decoder-only GPT2 (Radford et al., 2019), where we took the embedding of the last "[EOS]" token as the embedding of the utterances. Observe that although ArCHer w/ RoBERTa critic learns a bit faster in the beginning, learning curves for both of these critic models behave identically past a certain number of initial samples. Therefore, ArCHer can also use decoder-only transformer models, with no loss in performance.

5.6. Ablation Study: Scaling the Base Model from 100M to 7B Parameters

In Figure 5 (d), we replaced the 100 million parameter GPT-2 model used to represent the token-level actor in ArCHer with a 7 billion parameter Mistral model (Jiang et al., 2023). When using this 7B model, we did not need to apply supervised fine-tuning since the open-source checkpoint already attained non-trivial rewards on **Twenty Questions Subset** when evaluated zero-shot. Observe in Figure 5 (d), that ArCHer with this Mistral7B actor learns to solve the task much faster than ArCHer with a GPT2 Actor. This indicates that our ArCHer framework can scale well with LLMs with more parameters. More broadly, due to similarities between the token-level actor update in ArCHer and single-turn RL fine-tuning for LLMs in RLHF, we would expect performance to exhibit similar benefits from scaling the model size for the policy (Gao et al., 2023).

5.7. Alternate Practical Algorithms

In Figure 5 (a), we compare the performance of ArCHer with and without a token-level baseline on the **Guess My City** task (Equation 5). This task requires the utterances of the agent in each turn to be longer and more diverse than other tasks. Observe that incorporating this token-level baseline in ArCHer outperforms standard ArCHer by a large margin, supporting our hypothesis that the introduction of the token-level baseline can effectively reduce the variance of the vanilla policy gradient while updating the token-level actor (especially when the utterances are long and diverse). That said, this improvement requires paying an extra overhead associated with training $\tilde{V}_\eta$, which might not be necessary when each utterance is short.

To summarize, our experiments show that ArCHer can be used to derive multi-turn RL methods that lead to substantial improvements to sample efficiency of LLM training, benefit from offline and off-policy experience as well as improvements to RL algorithms, and scale with model capacity.

6. Discussion and Conclusion

We proposed a novel Actor-Critic Framework with a Hierarchical Structure (ArCHer) for multi-turn LLM agent tasks. By running two RL algorithms simultaneously, one at the high level (i.e., utterances in our practical method) and one at the low level (i.e., tokens), ArCHer reduces task horizons while enjoying the ability to retain a compact token-level action space at the low level. Empirically, we observed that ArCHer significantly outperforms prior RL methods for LLMs, on a range of online RL tasks and scales favorably with more capable base models and other design improvements. Deriving and evaluating novel practical algorithms from the hierarchical framework of ArCHer is also an interesting avenue for future work with the potential to greatly improve task performance in multi-turn problems. Our evaluations also focus entirely on tasks with computational rewards, and our method still requires thousands of interactions, so an important direction is to make these methods perform well with only hundreds of interactions.

Impact Statement

Our work focuses on RL methods for training LLMs. This has a variety of applications, and some of these applications have complex and unpredictable societal implications: LLM agents that can improve from interaction can be used for many highly desirable problems, such as personalization, improving efficiency of AI assistants, and more quickly resolving customer service issues. They can also be used in settings where care must be taken, such as marketing, where the ability to optimize for persuasiveness may lead to either unintended or unethical behavior. Finally, RL algorithms for LLMs can also be used maliciously, for example to enable agents to more effectively perform illegal acts. As RL methods for LLMs improve, we expect this will also increase the importance of research on detecting RL-enabled LLM agents in interactions, and also research on methods to mitigate unintended harmful behavior of such agents, in order to address the potential societal challenges of these technologies.

Acknowledgements

We thank Seohong Park, Mitsuhiko Nakamoto, Joey Hong, Yi Su, and members of the RAIL lab at UC Berkeley for valuable feedback on a previous version of this paper. YZ and AZ thank Marwa Abdulhai, Isadora White, and Charlie Snell for help with debugging and setting up the LMRL benchmark. AK thanks Charlie Snell, George Tucker, and Young Geng for informative discussions and perspectives. This work is supported by NSF IIS-2246811 and ONR N00014-21-1-2838, TRC TPU credit donations from Google Cloud, and compute credits from the Center for AI Safety (CAIS). AZ is supported by a postdoctoral fellowship from the Simons Institute for the Theory of Computing.

Author Contributions

This work was done at UC Berkeley. Yifei Zhou led the project. He wrote the prototype of ArCHer, iterated on refining and improving it, implemented the baselines for online comparisons, and different ablation in the online experiments. He also took a lead in writing the paper. Andrea Zanette led the offline experiments of the project. He developed the offline variant of ArCHer with IQL and AWR, and helped editing the paper. Jiayi Pan set up the environment of webshop for our use and implemented prompting baselines. He also refactored the code and helped set up data distributed parallel training for ArCHer and baselines. Sergey Levine advised the project, provided inputs for prototyping the method, and helped editing the paper. Aviral Kumar proposed the project idea, advised the project, helped the prototyping of ArCHer, and substantially edited the paper.

References

Abdulhai, M., White, I., Snell, C., Sun, C., Hong, J., Zhai, Y., Xu, K., and Levine, S. Lmrl gym: Benchmarks for multi-turn reinforcement learning with language models, 2023.

Bacon, P., Harb, J., and Precup, D. The option-critic architecture. *CoRR*, abs/1609.05140, 2016. URL http://arxiv.org/abs/1609.05140.

Bai, Y., Jones, A., Ndousse, K., Askell, A., Chen, A., DasSarma, N., Drain, D., Fort, S., Ganguli, D., Henighan, T., Joseph, N., Kadavath, S., Kernion, J., Conerly, T., El-Showk, S., Elhage, N., Hatfield-Dodds, Z., Hernandez, D., Hume, T., Johnston, S., Kravec, S., Lovitt, L., Nanda, N., Olsson, C., Amodei, D., Brown, T., Clark, J., McCandlish, S., Olah, C., Mann, B., and Kaplan, J. Training a helpful and harmless assistant with reinforcement learning from human feedback, 2022.

Budzianowski, P., Wen, T., Tseng, B., Casanueva, I., Ultes, S., Ramadan, O., and Gasic, M. Multiwoz - A large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. *CoRR*, abs/1810.00278, 2018. URL http://arxiv.org/abs/1810.00278.

Casper, S., Davies, X., Shi, C., Gilbert, T. K., Scheurer, J., Rando, J., Freedman, R., Korbak, T., Lindner, D., Freire, P., Wang, T., Marks, S., Segerie, C.-R., Carroll, M., Peng, A., Christoffersen, P., Damani, M., Slocum, S., Anwar, U., Siththaranjan, A., Nadeau, M., Michaud, E. J., Pfau, J., Krasheninnikov, D., Chen, X., Langosco, L., Hase, P., Bıyık, E., Dragan, A., Krueger, D., Sadigh, D., and Hadfield-Menell, D. Open problems and fundamental limitations of reinforcement learning from human feedback, 2023.

Chen, B., Shu, C., Shareghi, E., Collier, N., Narasimhan, K., and Yao, S. Fireact: Toward language agent fine-tuning. *ArXiv*, abs/2310.05915, 2023. URL https://api.semanticscholar.org/CorpusID:263829338.

Christiano, P., Leike, J., Brown, T. B., Martic, M., Legg, S., and Amodei, D. Deep reinforcement learning from human preferences, 2023.

Chung, H. W., Hou, L., Longpre, S., Zoph, B., Tay, Y., Fedus, W., Li, Y., Wang, X., Dehghani, M., Brahma, S., Webson, A., Gu, S. S., Dai, Z., Suzgun, M., Chen, X., Chowdhery, A., Castro-Ros, A., Pellat, M., Robinson, K., Valter, D., Narang, S., Mishra, G., Yu, A., Zhao, V., Huang, Y., Dai, A., Yu, H., Petrov, S., Chi, E. H., Dean, J., Devlin, J., Roberts, A., Zhou, D., Le, Q. V., and Wei, J. Scaling instruction-finetuned language models, 2022.

Dong, H., Xiong, W., Goyal, D., Zhang, Y., Chow, W., Pan, R., Diao, S., Zhang, J., Shum, K., and Zhang, T. Raft: Reward ranked finetuning for generative foundation model alignment, 2023.

Foster, D. J., Krishnamurthy, A., Simchi-Levi, D., and Xu, Y. Offline reinforcement learning: Fundamental barriers for value function approximation. *CoRR*, abs/2111.10919, 2021. URL https://arxiv.org/abs/2111.10919.

Fujimoto, S. and Gu, S. S. A minimalist approach to offline reinforcement learning. In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021.

Gao, L., Schulman, J., and Hilton, J. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning*, pp. 10835–10866. PMLR, 2023.

Ghosal, D., Shen, S., Majumder, N., Mihalcea, R., and Poria, S. Cicero: A dataset for contextualized commonsense inference in dialogues. In *Annual Meeting of the Association for Computational Linguistics*, 2022. URL https://api.semanticscholar.org/CorpusID:247762111.

Glaese, A., McAleese, N., Trebacz, M., Aslanides, J., Firoiu, V., Ewalds, T., Rauh, M., Weidinger, L., Chadwick, M., Thacker, P., Campbell-Gillingham, L., Uesato, J., Huang, P.-S., Comanescu, R., Yang, F., See, A., Dathathri, S., Greig, R., Chen, C., Fritz, D., Elias, J. S., Green, R., Mokrá, S., Fernando, N., Wu, B., Foley, R., Young, S., Gabriel, I., Isaac, W., Mellor, J., Hassabis, D., Kavukcuoglu, K., Hendricks, L. A., and Irving, G. Improving alignment of dialogue agents via targeted human judgements, 2022.

Gulcehre, C., Paine, T. L., Srinivasan, S., Konyushkova, K., Weerts, L., Sharma, A., Siddhant, A., Ahern, A., Wang, M., Gu, C., et al. Reinforced self-training (rest) for language modeling. *arXiv preprint arXiv:2308.08998*, 2023.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *CoRR*, abs/1801.01290, 2018. URL http://arxiv.org/abs/1801.01290.

Hausknecht, M. J., Ammanabrolu, P., Côté, M., and Yuan, X. Interactive fiction games: A colossal adventure. *CoRR*, abs/1909.05398, 2019. URL http://arxiv.org/abs/1909.05398.

Hong, J., Levine, S., and Dragan, A. Zero-shot goal-directed dialogue via rl on imagined conversations, 2023.

Jang, Y., Lee, J., and Kim, K.-E. GPT-critic: Offline reinforcement learning for end-to-end task-oriented dialogue systems. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=qaxhBG1UUaS.

Jaques, N., Shen, J. H., Ghandeharioun, A., Ferguson, C., Lapedriza, À., Jones, N., Gu, S. S., and Picard, R. W. Human-centric dialog training via offline reinforcement learning. *CoRR*, abs/2010.05848, 2020. URL https://arxiv.org/abs/2010.05848.

Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., de las Casas, D., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., Lavaud, L. R., Lachaux, M.-A., Stock, P., Scao, T. L., Lavril, T., Wang, T., Lacroix, T., and Sayed, W. E. Mistral 7b, 2023.

Korbak, T., Shi, K., Chen, A., Bhalerao, R., Buckley, C. L., Phang, J., Bowman, S. R., and Perez, E. Pretraining language models with human preferences, 2023.

Kostrikov, I., Nair, A., and Levine, S. Offline reinforcement learning with implicit q-learning, 2021.

Kumar, A., Fu, J., Soh, M., Tucker, G., and Levine, S. Stabilizing off-policy q-learning via bootstrapping error reduction. *Advances in Neural Information Processing Systems*, 32, 2019.

Kumar, A., Zhou, A., Tucker, G., and Levine, S. Conservative q-learning for offline reinforcement learning. *CoRR*, abs/2006.04779, 2020. URL https://arxiv.org/abs/2006.04779.

Lee, S., Zhu, Q., Takanobu, R., Li, X., Zhang, Y., Zhang, Z., Li, J., Peng, B., Li, X., Huang, M., and Gao, J. Convlab: Multi-domain end-to-end dialog system platform. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019.

Li, Y., Choi, D. H., Chung, J., Kushman, N., Schrittwieser, J., Leblond, R., Tom, Eccles, Keeling, J., Gimeno, F., Lago, A. D., Hubert, T., Choy, P., de, C., d'Autume, M., Babuschkin, I., Chen, X., Huang, P.-S., Welbl, J., Gowal, S., Alexey, Cherepanov, Molloy, J., Mankowitz, D. J., Robson, E. S., Kohli, P., de, N., Freitas, Kavukcuoglu, K., and Vinyals, O. Competition-level code generation with alphacode. *Science*, 378:1092 – 1097, 2022. URL https://api.semanticscholar.org/CorpusID:246527904.

Lin, X. V., Wang, C., Zettlemoyer, L., and Ernst, M. D. Nl2bash: A corpus and semantic parser for natural language interface to the linux operating system. *CoRR*, abs/1802.08979, 2018. URL http://arxiv.org/abs/1802.08979.

Liu, X., Yu, H., Zhang, H., Xu, Y., Lei, X., Lai, H., Gu, Y., Ding, H., Men, K., Yang, K., Zhang, S., Deng, X., Zeng, A., Du, Z., Zhang, C., Shen, S., Zhang, T., Su, Y., Sun, H., Huang, M., Dong, Y., and Tang, J. Agentbench: Evaluating llms as agents, 2023.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019. URL http://arxiv.org/abs/1907.11692.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. A. Playing atari with deep reinforcement learning. *CoRR*, abs/1312.5602, 2013. URL http://arxiv.org/abs/1312.5602.

Nachum, O., Gu, S., Lee, H., and Levine, S. Data-efficient hierarchical reinforcement learning. *CoRR*, abs/1805.08296, 2018. URL http://arxiv.org/abs/1805.08296.

Nasiriany, S., Pong, V. H., Lin, S., and Levine, S. Planning with goal-conditioned policies. *CoRR*, abs/1911.08453, 2019. URL http://arxiv.org/abs/1911.08453.

Nikishin, E., Schwarzer, M., D'Oro, P., Bacon, P.-L., and Courville, A. The primacy bias in deep reinforcement learning. In *International conference on machine learning*, pp. 16828–16847. PMLR, 2022.

OpenAI, :, Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., Avila, R., Babuschkin, I., Balaji, S., Balcom, V., Baltescu, P., Bao, H., Bavarian, M., Belgum, J., Bello, I., Berdine, J., Bernadett-Shapiro, G., Berner, C., Bogdonoff, L., Boiko, O., Boyd, M., Brakman, A.-L., Brockman, G., Brooks, T., Brundage, M., Button, K., Cai, T., Campbell, R., Cann, A., Carey, B., Carlson, C., Carmichael, R., Chan, B., Chang, C., Chantzis, F., Chen, D., Chen, S., Chen, R., Chen, J., Chen, M., Chess, B., Cho, C., Chu, C., Chung, H. W., Cummings, D., Currier, J., Dai, Y., Decareaux, C., Degry, T., Deutsch, N., Deville, D., Dhar, A., Dohan, D., Dowling, S., Dunning, S., Ecoffet, A., Eleti, A., Eloundou, T., Farhi, D., Fedus, L., Felix, N., Fishman, S. P., Forte, J., Fulford, I., Gao, L., Georges, E., Gibson, C., Goel, V., Gogineni, T., Goh, G., Gontijo-Lopes, R., Gordon, J., Grafstein, M., Gray, S., Greene, R., Gross, J., Gu, S. S., Guo, Y., Hallacy, C., Han, J., Harris, J., He, Y., Heaton, M., Heidecke, J., Hesse, C., Hickey, A., Hickey, W., Hoeschele, P., Houghton, B., Hsu, K., Hu, S., Hu, X., Huizinga, J., Jain, S., Jain, S., Jang, J., Jiang, A., Jiang, R., Jin, H., Jin, D., Jomoto, S., Jonn, B., Jun, H., Kaftan, T., Łukasz Kaiser, Kamali, A., Kanitscheider, I., Keskar, N. S., Khan, T., Kilpatrick, L., Kim, J. W., Kim, C., Kim, Y., Kirchner, H., Kiros, J., Knight, M., Kokotajlo, D., Łukasz Kondraciuk, Kondrich, A., Konstantinidis, A., Kosic, K., Krueger, G., Kuo, V., Lampe, M., Lan, I., Lee, T., Leike, J., Leung, J., Levy, D., Li, C. M., Lim, R., Lin, M., Lin, S., Litwin, M., Lopez, T., Lowe, R., Lue, P., Makanju, A., Malfacini, K., Manning, S., Markov, T., Markovski, Y., Martin, B., Mayer, K., Mayne, A., McGrew, B., McKinney, S. M., McLeavey, C., McMillan, P., McNeil, J., Medina, D., Mehta, A., Menick, J., Metz, L., Mishchenko, A., Mishkin, P., Monaco, V., Morikawa, E., Mossing, D., Mu, T., Murati, M., Murk, O., Mély, D., Nair, A., Nakano, R., Nayak, R., Neelakantan, A., Ngo, R., Noh, H., Ouyang, L., O'Keefe, C., Pachocki, J., Paino, A., Palermo, J., Pantuliano, A., Parascandolo, G., Parish, J., Parparita, E., Passos, A., Pavlov, M., Peng, A., Perelman, A., de Avila Belbute Peres, F., Petrov, M., de Oliveira Pinto, H. P., Michael, Pokorny, Pokrass, M., Pong, V., Powell, T., Power, A., Power, B., Proehl, E., Puri, R., Radford, A., Rae, J., Ramesh, A., Raymond, C., Real, F., Rimbach, K., Ross, C., Rotsted, B., Roussez, H., Ryder, N., Saltarelli, M., Sanders, T., Santurkar, S., Sastry, G., Schmidt, H., Schnurr, D., Schulman, J., Selsam, D., Sheppard, K., Sherbakov, T., Shieh, J., Shoker, S., Shyam, P., Sidor, S., Sigler, E., Simens, M., Sitkin, J., Slama, K., Sohl, I., Sokolowsky, B., Song, Y., Staudacher, N., Such, F. P., Summers, N., Sutskever, I., Tang, J., Tezak, N., Thompson, M., Tillet, P., Tootoonchian, A., Tseng, E., Tuggle, P., Turley, N., Tworek, J., Uribe, J. F. C., Vallone, A., Vijayvergiya, A., Voss, C., Wainwright, C., Wang, J. J., Wang, A., Wang, B., Ward, J., Wei, J., Weinmann, C., Welihinda, A., Welinder, P., Weng, J., Weng, L., Wiethoff, M., Willner, D., Winter, C., Wolrich, S., Wong, H., Workman, L., Wu, S., Wu, J., Wu, M., Xiao, K., Xu, T., Yoo, S., Yu, K., Yuan, Q., Zaremba, W., Zellers, R., Zhang, C., Zhang, M., Zhao, S., Zheng, T., Zhuang, J., Zhuk, W., and Zoph, B. Gpt-4 technical report, 2023.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L. E., Simens, M., Askell, A., Welinder, P., Christiano, P. F., Leike, J., and Lowe, R. J. Training language models to follow instructions with human feedback. *ArXiv*, abs/2203.02155, 2022. URL https://api.semanticscholar.org/CorpusID:246426909.

Park, S., Ghosh, D., Eysenbach, B., and Levine, S. Hiql: Offline goal-conditioned rl with latent states as actions, 2024.

Paulus, R., Xiong, C., and Socher, R. A deep reinforced model for abstractive summarization, 2017.

Peng, X. B., Kumar, A., Zhang, G., and Levine, S. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *CoRR*, abs/1910.00177, 2019. URL http://arxiv.org/abs/1910.00177.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. Language models are unsupervised multitask learners. 2019. URL https://api.semanticscholar.org/CorpusID:160025533.

Rafailov, R., Sharma, A., Mitchell, E., Ermon, S., Manning, C. D., and Finn, C. Direct preference optimization: Your language model is secretly a reward model, 2023.

Ramamurthy, R., Ammanabrolu, P., Brantley, K., Hessel, J., Sifa, R., Bauckhage, C., Hajishirzi, H., and Choi, Y. Is reinforcement learning (not) for natural language processing?: Benchmarks, baselines, and building blocks for natural language policy optimization. 2022. URL https://arxiv.org/abs/2210.01241.

Ramamurthy, R., Ammanabrolu, P., Brantley, K., Hessel, J., Sifa, R., Bauckhage, C., Hajishirzi, H., and Choi, Y. Is reinforcement learning (not) for natural language processing: Benchmarks, baselines, and building blocks for natural language policy optimization, 2023.

Ranzato, M., Chopra, S., Auli, M., and Zaremba, W. Sequence level training with recurrent neural networks, 2015.

Schick, T., Dwivedi-Yu, J., Dessì, R., Raileanu, R., Lomeli, M., Zettlemoyer, L., Cancedda, N., and Scialom, T. Toolformer: Language models can teach themselves to use tools, 2023.

Schulman, J., Moritz, P., Levine, S., Jordan, M. I., and Abbeel, P. High-dimensional continuous control using generalized advantage estimation. *CoRR*, abs/1506.02438, 2015. URL https://api.semanticscholar.org/CorpusID:3075448.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017. URL http://arxiv.org/abs/1707.06347.

Sharma, A., Gu, S., Levine, S., Kumar, V., and Hausman, K. Dynamics-aware unsupervised discovery of skills. *CoRR*, abs/1907.01657, 2019. URL http://arxiv.org/abs/1907.01657.

Shinn, N., Cassano, F., Labash, B., Gopinath, A., Narasimhan, K., and Yao, S. Reflexion: Language agents with verbal reinforcement learning. 2023. URL https://api.semanticscholar.org/CorpusID:258833055.

Snell, C., Kostrikov, I., Su, Y., Yang, M., and Levine, S. Offline rl for natural language generation with implicit language q learning, 2023.

Song, Y., Zhou, Y., Sekhari, A., Bagnell, J. A., Krishnamurthy, A., and Sun, W. Hybrid rl: Using both offline and online data can make rl efficient, 2023.

Sutton, R. S. Between mdps and semi-mdps : Learning , planning , and representing knowledge at multiple temporal scales. 1998. URL https://api.semanticscholar.org/CorpusID:2191003.

Sutton, R. S., McAllester, D., Singh, S., and Mansour, Y. Policy gradient methods for reinforcement learning with function approximation. In Solla, S., Leen, T., and Müller, K. (eds.), *Advances in Neural Information Processing Systems*, volume 12. MIT Press, 1999. URL https://proceedings.neurips.cc/paper_files/paper/1999/file/464d828b85b0bed98e80ade0a5c43b0f-Paper.pdf.

Szot, A., Schwarzer, M., Agrawal, H., Mazoure, B., Talbott, W., Metcalf, K., Mackraz, N., Hjelm, D., and Toshev, A. Large language models as generalizable policies for embodied tasks, 2023.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., Bikel, D., Blecher, L., Ferrer, C. C., Chen, M., Cucurull, G., Esiobu, D., Fernandes, J., Fu, J., Fu, W., Fuller, B., Gao, C., Goswami, V., Goyal, N., Hartshorn, A., Hosseini, S., Hou, R., Inan, H., Kardas, M., Kerkez, V., Khabsa, M., Kloumann, I., Korenev, A., Koura, P. S., Lachaux, M.-A., Lavril, T., Lee, J., Liskovich, D., Lu, Y., Mao, Y., Martinet, X., Mihaylov, T., Mishra, P., Molybog, I., Nie, Y., Poulton, A., Reizenstein, J., Rungta, R., Saladi, K., Schelten, A., Silva, R., Smith, E. M., Subramanian, R., Tan, X. E., Tang, B., Taylor, R., Williams, A., Kuan, J. X., Xu, P., Yan, Z., Zarov, I., Zhang, Y., Fan, A., Kambadur, M., Narang, S., Rodriguez, A., Stojnic, R., Edunov, S., and Scialom, T. Llama 2: Open foundation and fine-tuned chat models, 2023.

van Hasselt, H., Guez, A., and Silver, D. Deep reinforcement learning with double q-learning. *CoRR*, abs/1509.06461, 2015. URL http://arxiv.org/abs/1509.06461.

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

Vezhnevets, A. S., Osindero, S., Schaul, T., Heess, N., Jaderberg, M., Silver, D., and Kavukcuoglu, K. Feudal networks for hierarchical reinforcement learning. *CoRR*, abs/1703.01161, 2017. URL http://arxiv.org/abs/1703.01161.

Wang, G., Xie, Y., Jiang, Y., Mandlekar, A., Xiao, C., Zhu, Y., Fan, L. J., and Anandkumar, A. Voyager: An open-ended embodied agent with large language models. *ArXiv*, abs/2305.16291, 2023. URL https://api.semanticscholar.org/CorpusID:258887849.

Williams, R. J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256, 2004. URL https://api.semanticscholar.org/CorpusID:19115634.

Wu, Y. and Hu, B. Learning to extract coherent summary via deep reinforcement learning, 2018.

Xie, T., Cheng, C., Jiang, N., Mineiro, P., and Agarwal, A. Bellman-consistent pessimism for offline reinforcement learning. *CoRR*, abs/2106.06926, 2021. URL https://arxiv.org/abs/2106.06926.

Yang, J., Prabhakar, A., Narasimhan, K., and Yao, S. Intercode: Standardizing and benchmarking interactive coding with execution feedback, 2023a.

Yang, K., Swope, A. M., Gu, A., Chalamala, R., Song, P., Yu, S., Godil, S., Prenger, R., and Anandkumar, A. LeanDojo: Theorem Proving with Retrieval-Augmented Language Models. *arXiv preprint arXiv:2306.15626*, 2023b.

Yao, S., Chen, H., Yang, J., and Narasimhan, K. Webshop: Towards scalable real-world web interaction with grounded language agents, 2023a.

Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., and Cao, Y. React: Synergizing reasoning and acting in language models, 2023b.

Yuan, Z., Yuan, H., Tan, C., Wang, W., Huang, S., and Huang, F. Rrhf: Rank responses to align language models with human feedback without tears, 2023.

Zanette, A. When is realizability sufficient for off-policy reinforcement learning?, 2023.

Zeng, A., Liu, M., Lu, R., Wang, B., Liu, X., Dong, Y., and Tang, J. Agenttuning: Enabling generalized agent abilities for llms, 2023.

Zhan, W., Huang, B., Huang, A., Jiang, N., and Lee, J. D. Offline reinforcement learning with realizability and single-policy concentrability, 2022.

Zhou, S., Xu, F. F., Zhu, H., Zhou, X., Lo, R., Sridhar, A., Cheng, X., Bisk, Y., Fried, D., Alon, U., and Neubig, G. Webarena: A realistic web environment for building autonomous agents. *ArXiv*, abs/2307.13854, 2023a. URL https://api.semanticscholar.org/CorpusID:260164780.

Zhou, Y., Sekhari, A., Song, Y., and Sun, W. Offline data enhanced on-policy policy gradient with provable guarantees, 2023b.

Zhu, Q., Zhang, Z., Fang, Y., Li, X., Takanobu, R., Li, J., Peng, B., Gao, J., Zhu, X., and Huang, M. Convlab-2: An open-source toolkit for building, evaluating, and diagnosing dialogue systems. *CoRR*, abs/2002.04793, 2020. URL https://arxiv.org/abs/2002.04793.

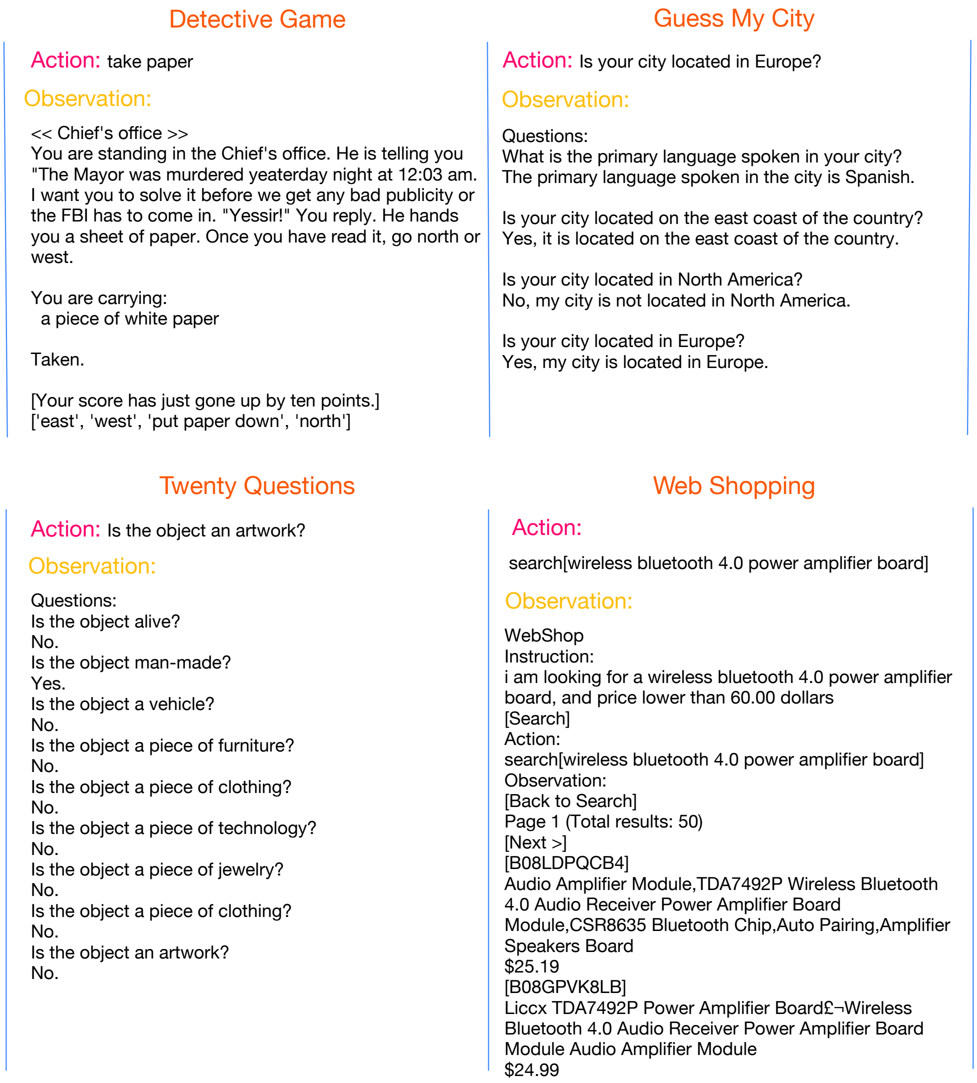
Ziegler, D. M., Stiennon, N., Wu, J., Brown, T. B., Radford, A., Amodei, D., Christiano, P. F., and Irving, G. Finetuning language models from human preferences. *CoRR*, abs/1909.08593, 2019. URL http://arxiv.org/abs/1909.08593.

Appendices

A. Environment and Dataset Details

In this section, we provide more details on the environments and datasets that we used in our experiments. Example actions and observations for each environment are shown in Figure 6.

Detective Game (Hausknecht et al., 2019). In this game, the agent plays the role of a detective looking into a case where the Mayor got murdered. At each time step, the agent generates a free-form text where the game engine parses the text and determines the next state at each time step. The game engine provides a feedback of "Illegal Action." if the generated text is an illegal action or cannot be correctly parsed. The optimal policy takes 51 steps to solve and reaches a maximum reward of 360. The game timeouts and terminates after 60 steps (including steps where illegal actions are generated). The observation at each time step includes the current surroundings, items carried, environment feedback for the outcome of the last action, and a list of available actions. The Supervised Fine-Tuning (SFT) dataset for this environment consists of 1000 trajectories of an agent picking a random action from the list of available actions at each timestep.



Detective Game

Action: take paper

Observation:

<< Chief's office >>
You are standing in the Chief's office. He is telling you "The Mayor was murdered yeaterday night at 12:03 am. I want you to solve it before we get any bad publicity or the FBI has to come in. "Yessir!" You reply. He hands you a sheet of paper. Once you have read it, go north or west.

You are carrying:
a piece of white paper

Taken.

[Your score has just gone up by ten points.]
['east', 'west', 'put paper down', 'north']

Guess My City

Action: Is your city located in Europe?

Observation:

Questions:
What is the primary language spoken in your city?
The primary language spoken in the city is Spanish.

Is your city located on the east coast of the country?
Yes, it is located on the east coast of the country.

Is your city located in North America?
No, my city is not located in North America.

Is your city located in Europe?
Yes, my city is located in Europe.

Twenty Questions

Action: Is the object an artwork?

Observation:

Questions:
Is the object alive?
No.
Is the object man-made?
Yes.
Is the object a vehicle?
No.
Is the object a piece of furniture?
No.
Is the object a piece of clothing?
No.
Is the object a piece of technology?
No.
Is the object a piece of jewelry?
No.
Is the object a piece of clothing?
No.
Is the object an artwork?
No.

Web Shopping

Action:

search[wireless bluetooth 4.0 power amplifier board]

Observation:

WebShop
Instruction:
i am looking for a wireless bluetooth 4.0 power amplifier board, and price lower than 60.00 dollars
[Search]
Action:
search[wireless bluetooth 4.0 power amplifier board]
Observation:
[Back to Search]
Page 1 (Total results: 50)
[Next >]
[B08LDPQCB4]
Audio Amplifier Module,TDA7492P Wireless Bluetooth 4.0 Audio Receiver Power Amplifier Board Module,CSR8635 Bluetooth Chip,Auto Pairing,Amplifier Speakers Board
$25.19
[B08GPVK8LB]
Liccx TDA7492P Power Amplifier Board£¬Wireless Bluetooth 4.0 Audio Receiver Power Amplifier Board Module Audio Amplifier Module
$24.99

Figure 6: Example actions and observations for each environment that we study in this paper.

Twenty Questions and Twenty Questions Subset (Abdulhai et al., 2023). In this environment, for each episode, a random word is chosen from a list of 157 words of household items such as "basketball", "apple", and "car". The word is held hidden from the agent and the agent is tasked to guess the hidden word within 20 questions. The questions are limited to yes/no questions and the answers from the oracle are limited to "Yes.", "No.", and "Invalid Question.". As opposed to using "flan-t5-xl"(Chung et al., 2022) as the oracle (Abdulhai et al., 2023), we train a 'flan-t5-small" to simulate the oracle with the same data and use it for our online experiments due to computational constraints. The agent gets a reward of 0 if it guesses the correct word and the episode terminates. Otherwise, the agent gets a reward of -1 for each question it raises. This reward structure results in a minimum reward of -20 if the agent does not guess the corret word with twenty questions and a maximum reward of 0 if the agent guesses the correct word with the first question although it is very unlikely. We use the official offline dataset provided by Abdulhai et al. (2023) with 100K simulated episodes. Our SFT checkpoints for online experiments for both Twenty Questions and Twenty Questions Subset are also trained with this dataset. Twenty Questions Subset keeps everything else the same except that it uses a subset of 10 hidden words in the word list. Since the offline dataset and the SFT checkpoint for online experiments are based on the entire Twenty Questions, Twenty Questions Subset challenges different algorithms with a significant distribution shift and requires the agent to come up with an entirely different strategy from behavior cloning.

Guess My City (Abdulhai et al., 2023). This environment is a similar dialogue task to Twenty Questions. For each episode, a random city is chosen from a list of 100 cities in the world. The city is held hidden from the agent and the agent is tasked to guess the name of the city within 20 questions. Both the questions and answers are free-form except that the answers are not allowed to contain the name of the city. As opposed to using "flan-t5-xl"(Chung et al., 2022) as the oracle (Abdulhai et al., 2023), we train a 'flan-t5-small" to simulate the oracle with the same data and use it for our online experiments due to computational constraints. We found in our online experiments that the agent can easily learn to "exploit" the oracle by tricking it to directly output the name of the city. Therefore, we simply replace the answer with a hardcoded template "I cannot answer that question." if the name of the city is found in the output of the oracle language model to reduce reward hacking. The reward structure is the same as Twenty Questions. We use the official offline dataset provided by Abdulhai et al. (2023) with 100K simulated episodes.

Web Shopping (Yao et al., 2023a). This environment challenges the ability of the agents to interact with external tools. For each episode, a random instruction requesting a specific item is chosen and shown to the agent. The agent needs to make use of a simplified web shopping server to make the purchase. Every successful purchase is consisted of searching the keywords in the search engine, selecting an item from searched results, clicking on features and attributes for the item, and finally making the purchase. Following ReAct (Yao et al., 2023b), the agent can choose to take a "think" action before taking any actual actions such as "search" and "click". An observation consists of the instruction and the history of visited webpages (described in text) and actions. The reward is a scalar between 0 and 1 depending on the similarity of the purchased item with the requested item. For example, a partial reward will be given if the agent purchases a black king-sized bed while a black queen-sized bed is requested. The episode timeouts after 10 interaction steps and a reward of 0 is issued. Our main online environments use a subset of 100 instructions from index 2000 to 2100 for a fast evaluation. We collect the offline dataset using the instructions from index 0 to 1000 with GPT-3 text-davinci-002 with prompts from ReAct's official implementation.

A.1. Framework Summary and Practical Implementation Details

Pseudocode. The algorithms derived from the ArCHer framework so far are summarized in Algorithm 1. These algorithms can operate in either offline or online mode (Line 4), and can utilize a variety of objectives for training the utterance-level Q- and V-models (Lines 9-15) as well the token-level policy (Line 20). Optionally, a token-level baseline value function may also be utilized (Line 17).

Implementation details. In our main experiments, we use a GPT-2 (Radford et al., 2019) architecture for parameterizing the policy, and a RoBERTa-base model (Liu et al., 2019) with a linear layer on top of the embeddings corresponding to the "[CLS]" token for obtaining the critic's predictions. To address the issue of overestimation of Q-values, we also employ the double Q-learning trick (van Hasselt et al., 2015) and train two copies of Q- and V-models, $\{Q_1, V_1\}$ and $\{Q_2, V_2\}$, independently. The advantage value is calculated by using a minimum over Q_1, Q_2 and V_1, V_2.

To save computation and memory costs, Q_1, Q_2, V_1, V_2 share the same language model encoder backbone with separate MLP heads. The parameters of the token-level actor are independent from the critic. When utilized, the token-level value baseline is parameterized by a separate GPT2 architecture with a MLP layer on top of the hidden states for each token. Additional details and hyperparameters for our approach are provided in Appendix F.

Algorithm 1 ArCHer: Practical Framework

```
Initialize parameters $\phi, \psi, \theta, \bar{\theta}$, (Optionally) $\eta$
Initialize replay buffer $\mathcal{D}$ (optionally from an offline dataset).
for each iteration do
    ## Data Collection.                                                          ▷ [only online mode]
    for each environment step do
        Execute $a_t \sim \pi_\phi(\cdot|s_t)$ , obtain the next state $s_{t+1}$, add to buffer $\mathcal{D}$.
    end for
    for each critic step do
        ## Update utterance-level Q and V functions by target function bootstrapping.
        $\theta \leftarrow \theta - \nabla J_\theta(Q)$                                      ▷ Equation 1
        $\psi \leftarrow \psi - \nabla J_\psi(V)$                                      ▷ Equation 2 or 6
        ## Update target Q and V functions.
        $\bar{\theta} \leftarrow (1-\tau)\bar{\theta} + \tau\theta$
        $\bar{\psi} \leftarrow (1-\tau)\bar{\psi} + \tau\psi$
    end for
    ## Update token-level baseline by MC regression.
    for each baseline step do
        $\eta \leftarrow \eta - \nabla J_\eta(\widetilde{V})$                              ▷ (Optionally), Equation 4
    end for
    ## Update token-level actor with utterance-level critic.
    for each actor step do
        $\phi \leftarrow \phi - \nabla J_\phi(\pi)$                                  ▷ Equation 3, 5, or 7
    end for
end for
```

B. Offline Algorithm and Practical Considerations

Our offline algorithm is a hierarchical version of the IQL algorithm (Kostrikov et al., 2021). Specifically, the critic leverages IQL (Eq. (6)) while the actor update is based on AWR (Equation 7).

These choices for the actor and for the critic update identify two key hyperparameters, the expectile value τ (defined in Equation 6 and 7) and the temperature β, whose effect is described in the respective sections. These hyper-parameters are already present in the original IQL algorithm (Kostrikov et al., 2021), and they have a similar interepretation here. By choosing τ and β appropriately, the algorithm identifies a policy whose performance should be between the optimal one and the one that generated the dataset. (In general, recovering the optimal policy by just using a dataset may not be possible as the dataset may not contain information about an optimal policy).

The offline algorithm shares most of the ingredients with its online counterpart, such as the double critic, target networks, soft updates, and value function heads. However, some unique features inherited from IQL allow to considerably simplify several algorithmic choices.

- The actor and the critic no longer need to be synchronized by using a certain update ratio. This is because the critic update defined in Equation 6 is independent of the actor's current policy, and so the two can be updated with any desired frequency without introducing instabilities.

- It is not necessary to pre-train the policy with a behavioural cloning objective, because such objective is already included in the actor's loss function in Equation 7.

- The warmup steps for the critic are also not necessary, because the initially small advantage function has a neglegible effect in the AWR loss.

C. Additional Baseline Details

C.1. Performance of PPO

In Figure 7, we provide a zoom-in of the learning curves of PPO for Twenty Questions, Twenty Questions Subset, and Guess My City. We observed that PPO does improve over the SFT checkpoint, especially in the more simple task Twenty Questions Subset. However, as PPO is unable to reuse past off-policy data, we need to collect at least 1024 trajectories of on-policy data for each PPO update, as shown in Appendix F. This observation is consistent with Abdulhai et al. (2023).

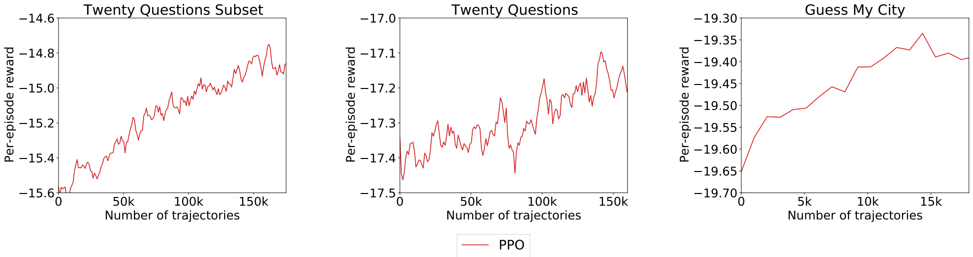


Figure 7: A zoom-in of learning curves of PPO. PPO gradually improves despite its worse sample complexity compared to other off-policy methods.

C.2. Additional Reproduction Details for WebShop Experiment

For our WebShop experiment, we utilized the environment and the few-shot prompting baselines from ReAct (Yao et al., 2023b). ReAct introduces two prompting strategies: the ReAct method and the Act-only method, which we denote as *ReAct* and *expert-prompt* in Figure 4. The ReAct method additionally allows the agent to articulate its reasoning before making an action, whereas the Act-only method does not. We use the original prompts in ReAct's implementation without any modifications, and to ensure that our paper is self-contained, we also include these prompts in Figure 9 and 8.

We assessed these algorithms on the webshop index from 2000 to 2100 to establish the LLM-based few-shot baselines in Figure 4. Notably, by early January 2024, OpenAI had deprecated the GPT-3 text-davinci-002 model used in the original ReAct study. Consequently, we switched to the gpt-3.5-turbo-instruct model. While the expert-prompt approach yielded performance comparable to the best performing text-davinci-002 based method, the ReAct method underperformed significantly with the newer model. This discrepancy, also observed and replicated in other tasks independetly by other researchers [2], is faithfully depicted in Figure 4.

---

[2] For more details, see the Github Issue on ReAct's repository: `https://github.com/ysymyth/ReAct`

ReAct Prompt
Webshop
Instruction:
i would like a 3 ounce bottle of bright citrus deodorant for sensitive skin, and price lower than 50.00 dollars
[Search]
Action: search[3 ounce bright citrus deodorant sensitive skin]
Observation:
[Back to Search]
Page 1 (Total results: 50)
[Next >]
[B078GWRC1J]
Bright Citrus Deodorant by Earth Mama — Natural and Safe for Sensitive Skin, Pregnancy and Breastfeeding, Contains Organic Calendula 3-Ounce
$10.99
[B078GTKVXY]
Ginger Fresh Deodorant by Earth Mama — Natural and Safe for Sensitive Skin, Pregnancy and Breastfeeding, Contains Organic Calendula 3-Ounce
$10.99
[B08KBVJ4XN]
Barrel and Oak - Aluminum-Free Deodorant, Deodorant for Men, Essential Oil-Based Scent, 24-Hour Odor Protection, Cedar & Patchouli Blend, Gentle on Sensitive Skin (Mountain Sage, 2.7 oz, 2-Pack)
$15.95
Action: think[B078GWRC1J and B078GTKVXY are bright citrus deodorant less then 50 dollars. I can check B078GWRC1J first.]
Observation: OK.
Action: click[B078GWRC1J]
Observation:
[Back to Search]
[< Prev]
scent [assorted scents][bright citrus][calming lavender][ginger fresh][simply non-scents]
size [travel set (4-pack)][3 ounce (pack of 1)][3-ounce (2-pack)]
Bright Citrus Deodorant by Earth Mama — Natural and Safe for Sensitive Skin, Pregnancy and Breastfeeding, Contains Organic Calendula 3-Ounce
Price: $10.99
Rating: N.A.
[Description]
[Features]
[Reviews]
[Buy Now]
Action: think[For 3 ounce bottle of bright citrus deodorant for sensitive skin, the item has options 'bright citrus' and '3 ounce (pack of 1)' and seems good to buy.]
Observation: OK.
Action: click[bright citrus]
Observation: You have clicked bright citrus.
Action: click[3 ounce (pack of 1)]
Observation: You have clicked 3 ounce (pack of 1).
Action: click[Buy Now]

Figure 8: Prompts used by the act-only (expert) baseline.

Act-only (expert) Prompt

```
Instruction:
i would like a 3 ounce bottle of bright citrus deodorant for sensitive skin, and price lower than 50.00 dollars
[Search]
Action: search[3 ounce bright citrus deodorant sensitive skin]
Observation:
[Back to Search]
Page 1 (Total results: 50)
[Next >]
[B078GWRC1J]
Bright Citrus Deodorant by Earth Mama — Natural and Safe for Sensitive Skin, Pregnancy and Breastfeeding, Contains Organic Calendula 3-Ounce
$10.99
[B078GTKVXY]
Ginger Fresh Deodorant by Earth Mama — Natural and Safe for Sensitive Skin, Pregnancy and Breastfeeding, Contains Organic Calendula 3-Ounce
$10.99
[B08KBVJ4XN]
Barrel and Oak - Aluminum-Free Deodorant, Deodorant for Men, Essential Oil-Based Scent, 24-Hour Odor Protection, Cedar & Patchouli Blend, Gentle on Sensitive Skin (Mountain Sage, 2.7 oz, 2-Pack)
$15.95
Action: click[B078GWRC1J]
Observation:
[Back to Search]
[< Prev]
scent [assorted scents][bright citrus][calming lavender][ginger fresh][simply non-scents]
size [travel set (4-pack)][3 ounce (pack of 1)][3-ounce (2-pack)]
Bright Citrus Deodorant by Earth Mama — Natural and Safe for Sensitive Skin, Pregnancy and Breastfeeding, Contains Organic Calendula 3-Ounce
Price: $10.99
Rating: N.A.
[Description]
[Features]
[Reviews]
[Buy Now]
Action: click[bright citrus]
Observation: You have clicked bright citrus.
Action: click[3 ounce (pack of 1)]
Observation: You have clicked 3 ounce (pack of 1).
Action: click[Buy Now]
```

Figure 9: Prompts used by the act-only (expert) baseline.

D. Additional Experimental Results

D.1. Offline ArCHer with IQL and AWR.

We now present a preliminary study of ArCHer in the offline setting, when learning from a static dataset from past environment interactions. Due to computational constraints, we were not able to perform extensive comparisons with the state-of-the-art algorithms; rather, we investigated the effect of several design choices in order to investigate the effect of various design choices in the offline setting, including IQL and AWR losses described in Section 3.5. We also incorporate a baseline, **BC**, which performs (unfiltered) imitation learning on the offline dataset. Finally, we also ran filtered BC, which only imitates the best trajectories in the offline dataset.

In Table 1, we also evaluate several other design choices in the offline setting. Directly borrowing the REINFORCE objective from the online setting (Equation 3) results in a quick collapse of performance due to the lack of any regularization to

	Twenty Questions (Return)
ArCHer (IQL + AWR)	$\mathbf{-14.1}$
ArCHer (IQL + REINFORCE)	-20
ArCHer (IQL + REINFORCE + BC)	-15.3
ArCHer (SARSA + AWR)	-14.5
Filtered BC	-15.4
BC	-16.8

Table 1: **Variants of ArCHer in the offline RL setting.** The performance of different approaches is evaluated by running 1280 trajectories across 5 random seeds for ArCHer and the Filtered BC approach. Observe that handling out-of-distribution actions by instantiating ArCHer with the IQL and AWR objectives works best in the offline setting.

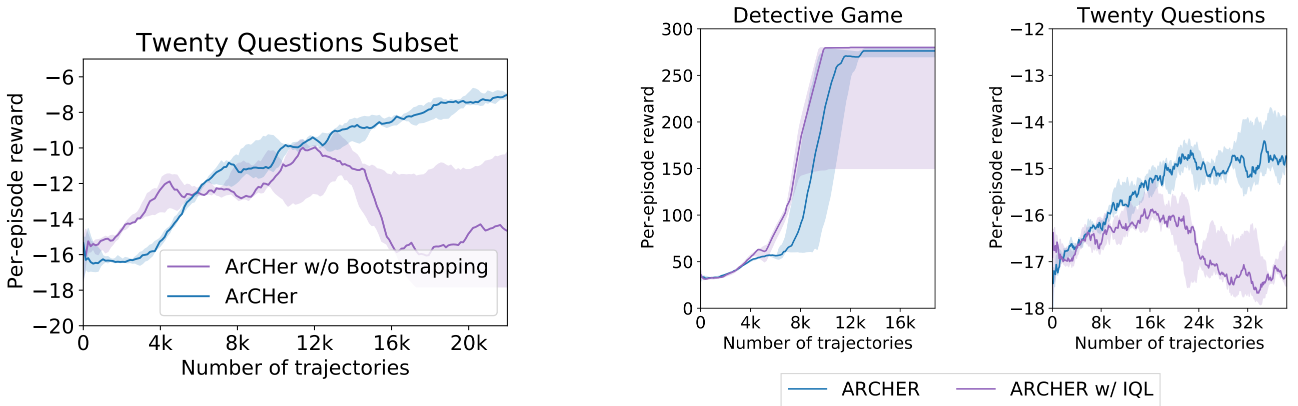
prevent out-of-distribution actions, as is well known in the offline RL problem setting outside of LLMs (Kumar et al., 2019). Combining Equation 3 with an imitation learning loss stabilizes learning and results in a performance improvement, but still underperforms advantage-weighted regression (AWR) (Peng et al., 2019). Finally, we replaced the IQL in-sample expectile backup with a SARSA backup, where the utterance present in the offline dataset at the next turn is used to compute the Bellman target, i.e., no implicit or explicit maximization over target values is utilized, and the value function is trained to represent the long-term Q-values of the data collection policy. Observe that this variant did not offer the same level of policy improvement as using IQL to train the critic in this setting. This highlights the importance of maximization over actions to calculate Bellman targets in the offline setting. Finally, we also find that instantiations of ArCHer that use IQL and SARSA in conjunction with AWR, both outperform the naïve BC and filtered BC, further higlighting the importance of dynamic programming to train the critic.

D.2. TD-Learning v.s. MC Regression

To validate whether TD-learning plays an important role in ArCHer, we carried out an ablation study where we replaced TD-learning in ArCHer with MC regression for critic updates. To make sure that data in the replay buffer are generated by similar policies, we use a smaller replay buffer that contains trajectories collected by three latest policies. The ablation results are shown in Figure 10a. We observed that MC regression may learn faster in the beginning as the information propagates faster than per step TD learning, but it fails to learn reliably over then entire training process. This ablation result shows the importance of TD-learning to effectively make use of off-policy data.

D.3. Online IQL Critic Loss

In our ablation study of using IQL critic loss in the online setting, we set $\tau = 0.9$ to encourage more risk-seeking for better explorations. As we can see from Figure 10b, the use of IQL critic can indeed accelerate explorations in simple tasks such as Detective Game. However, a naive instantiation of IQL fails to provide an unbiased estimate for the policy gradient resulting in potential instabilities in harder tasks such as Twenty Questions.



(a) Ablation study of replacing ArCHer critic update with MC Regression. As data in the replay buffer is generated by different policies, MC regression fails to improve reliably.

(b) Ablation study of replacing ArCHer critic update with an IQL loss. Using an IQL critic loss can sometimes accelerate training but introduce more instability.

E. Reward Hacking

To understand whether our agent has really learnt to behave more strategically with reinforcement learning instead of exploiting the environment, we present sample trajectories for Twenty Questions Subset, Twenty Questions, Guess My City, and WebShop in Figure 11, 12, 13, 14. Note that for detective game, there is only a limited number of legal actions and all illegal actions will lead to making no progress in the game, resulting in a lower rewards. For Twenty Questions Subset and Twenty Questions, we observed that the successful dialogues tend to be mostly natural and strategic while some unsuccessful dialogues tend to contain more meaningless repetitions. For Guess My City, we observed that the agent learnt to ask the predominant language of the city which is one of the most informative questions in the SFT distribution we considered in Appendix A. However, as there are some vulnerabilities and hallucinations of the imperfect oracle model for generating free-form answers, some of the conversations might diverge to meaningless repetitions in the end. For WebShop, as the environment forces the format of legal actions for the web server to process, the trajectories tend to be natural and realistic.

Twenty Questions Subset
Questions:
Is it a living thing? No.
Is it a man-made? Yes.
Is it a machine? No.
Is it a tool? No.
Is it a piece of clothing? No.
Is it a piece of furniture? Yes.
Is it a chair? Yes.

Figure 11: Example environment interactions of ArCHer for Twenty Questions Subset at the end of training. It can be seen that in this environment, ArCHer learnt a near-optimal binary search strategy.

Twenty Questions

Questions:
Is it a living thing? Yes.
Is the object a mammal? No.
Is the object a plant? Yes.
Is the object edible? Yes.
Is the object a vegetable? No.
Is the object a fruit? Yes.
Is the object a citrus fruit? No.
Is the object a berry? Yes.
Is the object a strawberry? No.
Is the object a strawberry? No.
Is the object a blueberry? No.
Is the object a rasepeberry? No.
Is the object a currant? No.
Is the object a grape? Yes.

Questions:
Is the object a living thing? No.
Is the object man-made? Yes.
Is the object a vehicle? No.
Is the object a piece of jewelry? No.
Is the object a musical instrument? Yes.
Is the object a stringed instrument? No.
Is the object a wind instrument? Yes.
Is the object a trumpet? No.
Is the object a trombone? No.
Is the object a flute? No.
Is the object a saxophone? No.
Is the object a clarinet? No.
Is the object a clarinet? No.
Is the object a clarinet? No.
Is the object a clarinet? No.
Is the object a clarinet? No.
Is the object a clarinet? No.
Is the object a clarinet? No.
Is the object a clarinet? No.
Is the object a clarinet? No.

Figure 12: Example environment interactions of ArCHer for Twenty Questions (one successful and one unsuccessful trajectory) at the end of training. In this environment, the conversation is smooth and natural for successful trajectories. However, for unsuccessful trajectories, the agent may sometimes keep repeating the same question.

Guess My City
Questions:
What is the predominant language spoken:
The predominant language spoken is Mandarin Chinese.
Is the city you are from Shanghai?
No, I am not from Shanghai.
Is the city you are from Taipei,Is the city you are from TaipeiIs the city you are from Tis aipeiIs the city you are from Tai
Yes

Questions:
What is the predominant language spoken:
English
Is the city you are from London?
No, I am not from London.
Is the city you are from LondonIs the city you are from London Is the city you are from London
Yes, my city is from London.
Is the city you are from LondonIs the city you are from London Is the city you are from London
Yes, my city is from London.
Is the city you are from LondonIs the city you are from London Is the city you are from London
Yes, my city is from London.
Is the city you are from LondonIs the city you are from London Is the city you are from London
Yes, my city is from London.
Is the city you are from LondonIs the city you are from London Is the city you are from London
Yes, my city is from London.
......

Figure 13: Example environment interactions of ArCHer for Guess My City at the end of training. The agent learns to ask the question for the predominant language, which is one of the most informative questions from the offline SFT distribution. However, as the oracle model provides free-form responses can be out of distribution, the agent is more prone to collapsing in out-of-distribution states.

WebShop
Observation:
WebShop
Instruction:
i would like a citrus yao conditioner made with natural ingredients, and price lower than 50.00 dollars
[Search]
Action:
search[citrus yao conditioner natural ingredients]
Observation:
[Back to Search]
Page 1 (Total results: 50)
[Next >]
[B099KWBV49]
VIORI Terrace Garden Conditioner Bar - Handcrafted with Longsheng Rice Water & Natural Ingredients - Sulfate-free, Paraben-free, Cruelty-free, Phthalate-free, pH balanced 100% Vegan, Zero-Waste
$16.25
[B09KQZ9GTK]
VIORI Hidden Waterfall Shampoo & Conditioner Bar Set WITHOUT Bamboo Holder - Handcrafted with Longsheng Rice Water & Natural Ingredients - Sulfate-free, Paraben-free, Phthalate-free, 100% Vegan
$29.25
Action:
click[B099KWBV49]
Observation:
[Back to Search]
[< Prev]
style [bamboo bar holder][citrus yao][hidden waterfall]
size [bamboo bar holder][conditioner][shampoo][shampoo + conditioner set]
VIORI Terrace Garden Conditioner Bar - Handcrafted with Longsheng Rice Water & Natural Ingredients - Sulfate-free, Paraben-free, Cruelty-free, Phthalate-free, pH balanced 100% Vegan, Zero-Waste
Price: $16.25
Rating: N.A.
[Description]
[Features]
[Reviews]
[Attributes]
[Buy Now]
Action:
click[citrus yao]
Observation:
You have clicked citrus yao.
Action:
click[Buy Now]
Observation:
Your score (min 0.0, max 1.0): 0.75

Figure 14: Example environment interactions of ArCHer for WebShop at the end of training. It can be seen that in this environment, the behavior of ArCHer is very natural.

F. Hyperparameters

Table 2 shows hyperparameters for ArCHer and other baselines for all environments. As shown in the table, most hyperparameters are held the same across all environment except that a smaller actor learning rate is used for Twenty Questions and a larger number of rollout trajectories is used for Web Shopping. This shows that ArCHer is relatively insensitive to selection of hyperparameters and does not require too much hyperparameter tuning to work in different environments.

Table 2: Hyperparameters for All Experiments

		Detective Game	Twenty Questions Subset	Twenty Questions	Guess My City	Web Shopping
SFT	actor lr	2e-4	2e-4	2e-4	2e-4	2e-4
	batch size	32	32	32	32	32
Filtered BC	actor lr	3e-4	3e-4	3e-5	3e-4	3e-4
	batch size	256	256	256	256	256
	rollout trajectories	32	32	32	32	128
	replay buffer size	10000	10000	10000	10000	10000
	filtering percentage	0.1	0.1	0.1	0.1	0.1
	actor updates per iteration	10	10	10	10	10
CHAI	actor lr	3e-4	3e-4	3e-5	3e-4	3e-4
	critic lr	6e-4	6e-4	6e-4	6e-4	6e-4
	batch size	256	256	256	256	256
	rollout trajectories	128	128	128	128	512
	replay buffer size	10000	10000	10000	10000	10000
	critic updates per iteration	50	50	50	50	50
	discount	0.98	0.95	0.95	0.95	0.9
	polyak alpha	0.9	0.9	0.9	0.9	0.9
ArCHer	actor lr	3e-4	3e-4	3e-5	3e-4	3e-4
	critic lr	6e-4	6e-4	6e-4	6e-4	6e-4
	batch size	256	256	256	256	256
	rollout trajectories	128	128	128	128	512
	replay buffer size	10000	10000	10000	10000	10000
	critic updates per iteration	50	50	50	50	50
	discount	0.98	0.95	0.95	0.95	0.9
	polyak alpha	0.9	0.9	0.9	0.9	0.9
	actor updates per iteration	3	3	3	3	3
	warm up iters with no actor update	10	10	20	10	20
ArCHer w/ Baseline	actor lr	\	\	\	3e-4	\
	critic lr	\	\	\	6e-4	\
	batch size	\	\	\	256	\
	rollout trajectories	\	\	\	128	\
	replay buffer size	\	\	\	10000	\
	critic updates per iteration	\	\	\	50	\
	discount	\	\	\	0.95	\
	actor updates per iteration	\	\	\	3	\
	baseline updates per iteration	\	\	\	60	\
	warm up iters with no actor update	\	\	\	10	\
	polyak alpha	\	\	\	0.9	\
PPO	actor lr	\	6e-6	6e-6	6e-4	\
	batch size	\	1024	1024	1024	\
	rollout trajectories	\	2048	2048	1024	\
	PPO epochs	\	10	20	4	\
	discount	\	0.95	0.95	0.95	\
	GAE lambda	\	0.95	0.95	0.95	\
	clip range	\	0.2	0.2	0.2	\

Table 3: Hyperparameters for ArCHer and baseline methods for all experiments.

G. Proof of Main Theorem

G.1. Equivalent Utterance and Token Level MDPs

We consider the groundtruth utterance-level discounted infinite horizon MDP $\mathcal{M} = \{\mathcal{S}, \mathcal{A}, \gamma, r, \mu_0, P\}$ as defined in Section 3.1. An equivalent token-level infinite horizon MDP can be constructed with $\widetilde{\mathcal{M}} = \{\widetilde{\mathcal{S}}, \widetilde{\mathcal{A}}, \gamma^{1/L}, r, \mu_0, \widetilde{P}\}$, where L is the length of each utterance, $\widetilde{\mathcal{A}}$ contains all individual tokens with a special padding token (such that each utterance is padded to the same length L), $\widetilde{\mathcal{S}} \in \mathcal{S} \times \widetilde{\mathcal{A}}^L$ contains both the state in the utterance-level MDP and the partial sequence of utterance that has already been generated, $\gamma^{1/L}$ is the equivalent discount factor. Note that this definition of token-level MDP is not the same as the definition of token-level MDP in the hierarchical language MDP defined for ArCHer in Section 3.1. The token-level MDP for ArCHer is only embedded in one particular utterance and the only reward that it receives is at the end of the utterance from the utterance-level critic while the equivalent token-level MDP spans multiple utterances and receives rewards from the external environment at the end of each utterance. This construction of token-level MDP is "equivalent" to the utterance-level MDP in the sense that for any autoregressive policy π that generates an utterance token by token, we have:

$$\forall s \in \mathcal{S}, V^\pi(s) = \widetilde{V}^\pi(s),$$

where V^π and $\widetilde{V}^\pi$ are value functions of π in the utterance-level MDP $\mathcal{M}$ and the token-level MDP $\widetilde{\mathcal{M}}$ respectively. We use $\widetilde{\pi}$ for the same utterance-level policy π when it generates one token at a time.

As usual, for any MDP $\mathcal{M}$, we define $d^\pi \in \Delta(\mathcal{S} \times \mathcal{A})$ as the average state-action occupancy measure of policy π such that:

$$d^\pi(s,a) = (1-\gamma)(\mu_0(s)\pi(a|s) + \sum_{t=1}^{\infty} \gamma^t P^\pi(s_t = s, a_t = a))$$

We denote $V^\pi = \mathbb{E}_{s_0 \sim \mu_0} V^\pi(s_0)$ as the expected total discounted reward of π. We denote $\mathcal{T}^\pi$ as the Bellman operator associated with π, i.e., given a function $f \in \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$, we have

$$\mathcal{T}^\pi f(s,a) = r(s,a) + \gamma \mathbb{E}_{s' \sim P(s,a), a' \sim \pi(s')}[f(s',a')].$$

Similar definitions can be made in the token-level MDP $\widetilde{\mathcal{M}}$ for $\widetilde{d}^\pi$, $\widetilde{V}^\pi$, and $\widetilde{\mathcal{T}}^\pi$. We also define $\widetilde{A}^\pi$ as the advantage function in the token level.

G.2. Fitted Policy Evaluation Subroutine

In this section, we present our theoretical subroutine for fitting the critic in Algorithm 2. On a high level, it just repeats finding the Q function that minimizes the bellman error with respect to the Q function in the last iteration, and returns the average of all Q functions in the end. Both critic fitting in the token-level MDP $\widetilde{\mathcal{M}}$ or in the utterance-level MDP $\mathcal{M}$ follows from the same subroutine with the same function class $\mathcal{F}$ that map the space of seuqnces of tokens to real values. This theoretical algorithm is simply a more fomalized version of the critic update in Algorithm 1.

Algorithm 2 Fitted **P**olicy **E**valuation (**FPE**)

```
Require: Policy $\pi$, function class $\mathcal{F}$, number of iterations $K$, weight $\lambda$
Require: $K$ independent datasets $\mathcal{D}_{1:K} = \{(s,a,r,s')\}$ of $M$ many samples each from the same offline distribution $\nu$.
Initialize $f_0 \in \mathcal{F}$.
for $k = 1, \dots, K$ do
    Solve the square loss regression problem to compute:
        $f_k \leftarrow \mathop{\mathrm{argmin}}_{f \in \mathcal{F}} \widehat{\mathbb{E}}_{\mathcal{D}_k}[(f(s,a) - r - \gamma f_{k-1}(s', \pi(s')))^2]$    (8)
end for
Return $\bar{f} = \frac{1}{K}\sum_{k=1}^{K} f_k$.
```

G.3. Assumptions

We present the two important assumptions that we use for analyzing FPE subroutine, and both assumptions share the same definition for utterance-level MDP $\mathcal{M}$ and token-level MDP $\widetilde{\mathcal{M}}$.

Assumption 1 (Bellman Completeness (Song et al., 2023; Zhou et al., 2023b; Zanette, 2023; Xie et al., 2021)). *We say that $\mathcal{F}$ is Bellman Complete for some policy π, if for all $f \in \mathcal{F}$, there exists a $f' \in \mathcal{F}$ such that $\|f'(s,a) - \mathcal{T}^\pi f(s,a)\|_\infty = 0$.*

Assumption 2 (Density Ratio Coefficient (Zhan et al., 2022; Foster et al., 2021)). *Given the offline distribution ν, for any policy π, we define the density ratio coefficient as*

$$C_{\nu,\pi} := \max_{s,a} \frac{d^\pi(s,a)}{\nu(s,a)}.$$

The following two lemmas compare the utterance-level assumptions and token-level assumptions.

Lemma 1. *For any stationary policy π, token-level Bellman Completeness for $\widetilde{\mathcal{M}} \implies$ utterance-level Bellman Completeness for $\mathcal{M}$*

Proof. $\forall s \in \mathcal{S}, a \in \mathcal{A}$ being state and action in the utterance-level, the utterance a can be decomposed into L tokens in the action space of the token-level MDP:

$$a = \widetilde{a}_{1:L}, \widetilde{a}_i \in \widetilde{\mathcal{A}}, i = 1, \dots, L.$$

Therefore,

$$\begin{aligned}
&\min_{f' \in \mathcal{F}} |f'(s,a) - \mathcal{T}^\pi f'(s,a)| \\
=& \min_{f_1,\dots,f_L \in \mathcal{F}} |f_1(s,a) - \widetilde{\mathcal{T}}^\pi f_2(s,a) + r(s,a) + \gamma^{1/L} \mathbb{E}_{s' \sim P(\cdot|s,a), \widetilde{a}_1 \sim \widetilde{\pi}(\cdot|s')} f_2(s', \widetilde{a}_1) \\
&- \gamma^{1/L} \mathbb{E}_{s' \sim P(\cdot|s,a), \widetilde{a}_1 \sim \widetilde{\pi}(\cdot|s')} \widetilde{\mathcal{T}}^\pi f_3(s', \widetilde{a}_1) + \cdots + \gamma^{(L-1)/L} \mathbb{E}_{s' \sim P(\cdot|s,a), \widetilde{a}_{1:L-1} \sim \widetilde{\pi}(\cdot|s')} f_L(s', \widetilde{a}_{1:L-1}) \\
&- r(s,a) - \gamma^{(L-1)/L} \mathbb{E}_{s' \sim P(\cdot|s,a), \widetilde{a}_{1:L-1} \sim \widetilde{\pi}(\cdot|s')} \widetilde{\mathcal{T}}^\pi f(s', \widetilde{a}_{1:L-1})| \\
\leq& \min_{f_1,\dots,f_L \in \mathcal{F}} |f_1(s,a) - \widetilde{\mathcal{T}}^\pi f_2(s,a)| \sum_{i=2}^{L} \gamma^{(i-1)/L} \mathbb{E}_{s' \sim P(\cdot|s,a), \widetilde{a}_{1:i-1} \sim \widetilde{\pi}(\cdot|s'))} |f^i(s, \widetilde{a}_{1:i-1}) - \widetilde{\mathcal{T}}^\pi((s, \widetilde{a}_{1:i-1})| \\
\leq& 0,
\end{aligned}$$

where the last inequality follows from the assumption of token-level Bellman Complete. $\square$

The next lemma assumes that the offline distribution ν is the same for both token-level MDP and utterance-level MDP, i.e. the token-level transitions are derived by splitting utterance-level transitions. We use $\widetilde{\nu}$ to denote the token-level offline distribution created in this way.

Lemma 2. *For any stationary policy π, and any offline distribution ν, we have Density Ratio Coefficient for token-level $\widetilde{\mathcal{M}}$ = Density Ratio Coefficient for utterance-level $\mathcal{M}$.*

$$C_{\nu,\pi} = C_{\widetilde{\nu},\pi}$$

Proof. The proof is constructed by noticing the fact that each token-level state consists of not only the utterance-level state but also all the tokens that have been generated in the current utterance:

$$\begin{aligned}
&\max_{\widetilde{s} \in \widetilde{\mathcal{S}}, \widetilde{a} \in \widetilde{\mathcal{A}}} \frac{\widetilde{d}^\pi(\widetilde{s}, \widetilde{a})}{\widetilde{\nu}(\widetilde{s}, \widetilde{a})} \\
=& \max_{s \in \mathcal{S}, \widetilde{a}_{1:i-1} \in \widetilde{\mathcal{A}}, i \in [1,L]} \frac{\widetilde{d}^\pi(s, \widetilde{a}_{1:i-1}, \widetilde{a}_i)}{\widetilde{\nu}(s, \widetilde{a}_{1:i-1}, \widetilde{a}_i)} \\
=& \max_{s \in \mathcal{S}, \widetilde{a}_{1:L} \in \widetilde{\mathcal{A}}} \frac{\widetilde{d}^\pi(s, \widetilde{a}_{1:L})}{\widetilde{\nu}(s, \widetilde{a}_{1:L})} \\
=& \max_{s \in \mathcal{S}, a \in \mathcal{A}} \frac{d^\pi(s,a)}{\nu(s,a)},
\end{aligned}$$

where the first equation follows by regrouping states and actions as each state in the token-level state space $\widetilde{s} \in \widetilde{\mathcal{S}} = \mathcal{S} \times \widetilde{\mathcal{A}}^L$, and the second equation holds because $\max_b \frac{p(a,b)}{q(a,b)} \geq \frac{p(a)}{q(a)}$. $\square$

G.4. Proof of Main Theorem

Now we are ready to analyze the sample complexity of Fitted Policy Evaluation in Algoritm 2, the proof of our main theorem makes use of several technical lemmas from Section G.5.

Lemma 3 (Guarantees of FPE)**.** *For the algorithm described in Algorithm 2 with K independent datasets $\mathcal{D}_{1:K} = \{(s,a,r,s')\}$ such that the effective number of samples $N = MK$, if the function class satisfies $\max_{f\in\mathcal{F},s\in\mathcal{S},a\in\mathcal{A}}|f(s,a)| < R$, then with probability at least $1-\delta$ the output value function satisfies:*

$$\mathbb{E}_{s,a\sim d^\pi}[(\bar{f}(s,a) - \mathcal{T}^\pi \bar{f}(s,a))^2] \leq C_{\nu,\pi}\left(\frac{4(R+1)^2}{K} + \frac{256R^2\log(2|\mathcal{F}|K/\delta)}{M}\right).$$

Proof. By applying Lemma 6, to each iteration $1,\dots,K$, and apply a union bound over all iterations, we have that:

$$\begin{aligned}\forall k, M\mathbb{E}_{s,a\sim\nu}[(f_k(s,a) - \mathcal{T}^\pi f_{k-1}(s,a))^2] &\leq 3M\inf_{f\in\mathcal{F}}\mathbb{E}_{s,a\sim\nu}[(f(s,a) - \mathcal{T}^\pi f_{k-1}(s,a))^2] + 256R^2\log(2|\mathcal{F}|K/\delta)\\ \forall k, \mathbb{E}_{s,a\sim\nu}[(f_k(s,a) - \mathcal{T}^\pi f_{k-1}(s,a))^2] &\leq \frac{256R^2\log(2|\mathcal{F}|K/\delta)}{M}, \qquad (9)\end{aligned}$$

where the second line holds by Assumption 1. To combine the guarantees that we have from each round to the guarantee of the returned value function $\bar{f}$:

$$\begin{aligned}&\mathbb{E}_{s,a\sim\nu}[(\bar{f}(s,a) - \mathcal{T}^\pi\bar{f}(s,a))^2]\\ =&\mathbb{E}_{s,a\sim\nu}\left[\left(\frac{1}{K}(f_1(s,a) - \mathcal{T}^\pi f_K(s,a) + \sum_{k=2}^{K} f_k(s,a) - \mathcal{T}^\pi f_{k-1}(s,a))\right)^2\right]\\ \leq&\frac{1}{K}\mathbb{E}_{s,a\sim\nu}\left[(f_1(s,a) - \mathcal{T}^\pi f_K(s,a))^2 + \sum_{k=2}^{K}(f_k(s,a) - \mathcal{T}^\pi f_{k-1}(s,a))^2\right]\\ \leq&\frac{1}{K}\left[4(R+1)^2 + (K-1)\frac{256R^2\log(2|\mathcal{F}|K/\delta)}{M}\right]\\ \leq&\frac{4(R+1)^2}{K} + \frac{256R^2\log(2|\mathcal{F}|K/\delta)}{M}, \qquad (10)\end{aligned}$$

where the first inequality follows by Jensen's Inequality and the second inequality follows by plugging in Line 9. Then, we can plug in Assumption 2 to conclude the proof.

$$\begin{aligned}\mathbb{E}_{s,a\sim d^\pi}[(\bar{f}(s,a) - \mathcal{T}^\pi\bar{f}(s,a))^2] \leq& C_{\nu,\pi}\mathbb{E}_{s,a\sim\nu}[(\bar{f}(s,a) - \mathcal{T}^\pi\bar{f}(s,a))^2]\\ &\leq C_{\nu,\pi}(\frac{4(R+1)^2}{K} + \frac{256R^2\log(2|\mathcal{F}|K/\delta)}{M}).\\ &\leq 64C_{\nu,\pi}R(R+1)\sqrt{\frac{\log(2|\mathcal{F}|K/\delta)}{KM}}\\ &= 64C_{\nu,\pi}R(R+1)\sqrt{\frac{\log(2|\mathcal{F}|K/\delta)}{N}}\\ &:= \epsilon_{stat} \qquad (11)\end{aligned}$$

□

Theorem 1 (Main Theorem)**.** *For an utterance-level MDP with discount factor γ^L, where L is the maximum length of each utterance, suppose utterance-level Assumption 1 and 2 holds, let f be the final Q function returned by Fitted Policy Evaluation formalized in Algorithm 2 at the utterance level, yields a suboptimality gap of*

$$\begin{aligned}&\mathbb{E}_{s,a\sim d^\pi}[((\bar{f}(s,a) - \mathbb{E}_{a'\sim\pi(\cdot|s)}[\bar{f}(s,a)]) - A^\pi(s,a))^2]\\ \leq&\frac{4}{1-\gamma}(\epsilon_{stat} + 2(R+1)\sqrt{\epsilon_{stat}})\\ \leq&\frac{4}{\gamma L(1-\gamma^{1/L})}(\epsilon_{stat} + 2(\frac{1}{1-\gamma}+1)\sqrt{\epsilon_{stat}}).\end{aligned}$$

For an equivalent token-level MDP with discount factor $\gamma^{1/L}$, suppose token-level Assumption 1 and 2 holds, let f be the final Q function returned by Fitted Policy Evaluation formalized in Algorithm 2 at the token level, yields a suboptimality gap of

$$\begin{aligned}
&\mathbb{E}_{s,a\sim d^\pi}[((\bar{f}(s,a) - \mathbb{E}_{a'\sim\pi(\cdot|s)}[\bar{f}(s,a)]) - \widetilde{A}^\pi(s,a))^2]\\
\leq &\frac{4}{(1-\gamma^{1/L})L^{1/2}}(\epsilon_{stat} + 2(\frac{1}{1-\gamma} + 1)\sqrt{\epsilon_{stat}}L^{1/4}),
\end{aligned}$$

where ϵ_{stat} is the statistical error defined in Line 11 proportional to $N^{-1/2}$ the number of utterance-level transitions. This error term is defined the same for both utterance-level MDP and token-level MDP.

Proof. First, we start with analyzing utterance-level FPE with effective number of samples, we start by bounding the errors of the Q functions:

$$\begin{aligned}
&\mathbb{E}_{s,a\sim d^\pi}(\bar{f}(s,a) - Q^\pi(s,a))^2\\
=&\mathbb{E}_{s,a\sim d^\pi}(\bar{f}(s,a) - \mathcal{T}^\pi\bar{f}(s,a) + \mathcal{T}^\pi\bar{f}(s,a) - Q^\pi(s,a))^2\\
=&\mathbb{E}_{s,a\sim d^\pi}[(\bar{f}(s,a) - \mathcal{T}^\pi\bar{f}(s,a))^2 + 2(\bar{f}(s,a) - \mathcal{T}^\pi\bar{f}(s,a))(\mathcal{T}^\pi\bar{f}(s,a) - Q^\pi(s,a)) + (\mathcal{T}^\pi\bar{f}(s,a) - Q^\pi(s,a))^2]\\
\leq&\mathbb{E}_{s,a\sim d^\pi}[(\bar{f}(s,a) - \mathcal{T}^\pi\bar{f}(s,a))^2] + 2\sqrt{\mathbb{E}_{s,a\sim d^\pi}(\bar{f}(s,a) - \mathcal{T}^\pi\bar{f}(s,a))^2(\mathcal{T}^\pi\bar{f}(s,a) - Q^\pi(s,a))^2}\\
&+ \mathbb{E}_{s,a\sim d^\pi}[(\mathcal{T}^\pi\bar{f}(s,a) - Q^\pi(s,a))^2]\\
\leq&\mathbb{E}_{s,a\sim d^\pi}[(\bar{f}(s,a) - \mathcal{T}^\pi\bar{f}(s,a))^2] + 2(R+1)\sqrt{\mathbb{E}_{s,a\sim d^\pi}(\bar{f}(s,a) - \mathcal{T}^\pi\bar{f}(s,a))^2} + \mathbb{E}_{s,a\sim d^\pi}[(\mathcal{T}^\pi\bar{f}(s,a) - Q^\pi(s,a))^2]\\
\leq&\mathbb{E}_{s,a\sim d^\pi}[(\bar{f}(s,a) - \mathcal{T}^\pi\bar{f}(s,a))^2] + 2(R+1)\sqrt{\mathbb{E}_{s,a\sim d^\pi}(\bar{f}(s,a) - \mathcal{T}^\pi\bar{f}(s,a))^2}\\
&+ \gamma^2\mathbb{E}_{s,a\sim d^\pi,s'\sim P(\cdot|s,a),a'\sim\pi(\cdot|s')}[(\bar{f}(s',a') - Q^\pi(s',a'))^2]\\
\leq&\mathbb{E}_{s,a\sim d^\pi}[(\bar{f}(s,a) - \mathcal{T}^\pi\bar{f}(s,a))^2] + 2(R+1)\sqrt{\mathbb{E}_{s,a\sim d^\pi}(\bar{f}(s,a) - \mathcal{T}^\pi\bar{f}(s,a))^2} + \gamma\mathbb{E}_{s,a\sim d^\pi}(\bar{f}(s,a) - Q^\pi(s,a))^2,
\end{aligned}$$

where the first inequality follows from Jensen's inequality, and the second inequality follows from $\max_{s,a}\max\{|\mathcal{T}^\pi\bar{f}(s,a)|, |Q^\pi(s,a)|\} \leq R+1$, the third inequality follows again from Jensen's inequality,and the fourth inequality follows from Lemma 5. Finally, we get:

$$\begin{aligned}
&\mathbb{E}_{s,a\sim d^\pi}(\bar{f}(s,a) - Q^\pi(s,a))^2\\
\leq&\frac{1}{1-\gamma}(\mathbb{E}_{s,a\sim d^\pi}[(\bar{f}(s,a) - \mathcal{T}^\pi\bar{f}(s,a))^2] + 2(R+1)\sqrt{\mathbb{E}_{s,a\sim d^\pi}(\bar{f}(s,a) - \mathcal{T}^\pi\bar{f}(s,a))^2})\\
=&\frac{1}{1-\gamma}(\epsilon_{stat} + 2(R+1)\sqrt{\epsilon_{stat}}),
\end{aligned}$$

where ϵ_{stat} is defined in Line 11. Finally, we can directly bound the errors for the advantages:

$$\begin{aligned}
&\mathbb{E}_{s,a\sim d^\pi}[((\bar{f}(s,a) - \mathbb{E}_{a'\sim\pi(\cdot|s)}\bar{f}(s,a)) - A^\pi(s,a))^2]\\
=&\mathbb{E}_{s,a\sim d^\pi}[((\bar{f}(s,a) - \mathbb{E}_{a'\sim\pi(\cdot|s)}\bar{f}(s,a)) - (Q^\pi(s,a) - \mathbb{E}_{a'\sim\pi(\cdot|s)Q^\pi(s,a)}))^2]\\
\leq&2\mathbb{E}_{s,a\sim d^\pi}[(\bar{f}(s,a) - Q^\pi(s,a))^2] + 2\mathbb{E}_{s\sim d^\pi}[(\mathbb{E}_{a'\sim\pi(\cdot|s)}[\bar{f}(s,a') - Q^\pi(s,a')])^2]\\
\leq&4\mathbb{E}_{s,a\sim d^\pi}[(\bar{f}(s,a) - Q^\pi(s,a))^2]\\
\leq&\frac{4}{1-\gamma}(\epsilon_{stat} + 2(R+1)\sqrt{\epsilon_{stat}}),\\
\leq&\frac{4}{(1-\gamma^{1/L})\gamma L}(\epsilon_{stat} + 2(R+1)\sqrt{\epsilon_{stat}})\\
=&\frac{4}{(1-\gamma^{1/L})\gamma L}(\epsilon_{stat} + 2(\frac{1}{1-\gamma} + 1)\sqrt{\epsilon_{stat}})\\
&,
\end{aligned}$$

where the second inequality follows from Jensen's inequality, and the fourth inequality follows from Lemma 4. A similar analysis can be done with token-level FPE, noticing that the effective number of samples for token-level FPE is NL because each utterance transition can be splitted into L token transitions:

$$\begin{aligned}\mathbb{E}_{s,a\sim\widetilde{d}^\pi}[((\bar{f}(s,a) - \mathbb{E}_{a'\sim\pi(\cdot|s)}\bar{f}(s,a)) - \widetilde{A}^\pi(s,a))^2] \le & \frac{4}{1-\gamma}(\epsilon_{stat}L^{-1/2} + 2(R+1)\sqrt{\epsilon_{stat}}L^{-1/4}). \\ \le & \frac{4}{(1-\gamma)L^{1/2}}(\epsilon_{stat} + 2(\frac{1}{1-\gamma}+1)\sqrt{\epsilon_{stat}}L^{1/4}).\end{aligned}$$

□

G.5. Technical Lemmas

First, we would like to prove an interesting lemma that allows us to compare the different discount factors for token level $\gamma^{1/L}$ and utterance level γ.

Lemma 4. *With the discount factor $\gamma \in (0,1)$, and L being a positive number, we have that:*

$$\begin{aligned}\frac{1}{1-\gamma} =& \frac{1}{(1-\gamma^{1/L})(1+\gamma^{1/L}+\gamma^{2/L}+\cdots+\gamma^{(L-1)/L})} \\ \le& \frac{1}{L\gamma(1-\gamma^{1/L})}\end{aligned}$$

Below are some common technical lemmas useful for reinforcement learning.

Lemma 5 ((Zhou et al., 2023b, Lemma 2))**.** *For any policy π, and non-negative function $g(s,a)$, we have:*

$$\mathbb{E}_{\bar{s},\bar{a}\sim d^\pi}\mathbb{E}_{s\sim P(\cdot|\bar{s},\bar{a}),a\sim\pi(a|s)}[g(s,a)] \le \frac{1}{\gamma}\mathbb{E}_{s,a\sim d^\pi}[g(s,a)].$$

where μ_0 denotes the initial state distribution (which is the same for all policies π).

Proof. Recall that $\lim_{h\to\infty}\gamma^h = 0$. We start by noting that:

$$\begin{aligned}d^\pi(s,a) &= (1-\gamma)(\mu_0(s,a) + \gamma d_1^\pi(s,a) + \gamma^2 d_2^\pi(s,a) + \dots) \qquad (12)\\ &\ge \gamma(1-\gamma)\left(\sum_{\bar{s},\bar{a}}\mu_0(\bar{s},\bar{a})P(s|\bar{s},\bar{a})\pi(a|s) + \gamma\sum_{\bar{s},\bar{a}}d_1^\pi(\bar{s},\bar{a})P(s|\bar{s},\bar{a})\pi(a|s) + \dots\right)\\ &= \gamma(1-\gamma)\sum_{\bar{s},\bar{a}}\left(\mu_0(\bar{s},\bar{a}) + \gamma d_1^\pi(\bar{s},\bar{a}) + \dots\right)P(s|\bar{s},\bar{a})\pi(a|s)\\ &= \gamma\sum_{\bar{s},\bar{a}}d^\pi(\bar{s},\bar{a})P(s|\bar{s},\bar{a})\pi(a|s) \qquad (13)\\ &= \gamma\mathbb{E}_{\bar{s},\bar{a}\sim d^\pi}[P(s|\bar{s},\bar{a})\pi(a|s)],\end{aligned}$$

where Line 13 follows by plugging in the relation in Line 12 for $\bar{s},\bar{a}$. The above implies that for any function $g \ge 0$,

$$\sum_{s,a}d^\pi(s,a)g(s,a) \ge \sum_{s,a}\gamma\mathbb{E}_{\bar{s},\bar{a}\sim d^\pi}[P(s|\bar{s},\bar{a})\pi(a|s)g(s,a)],$$

which implies that

$$\mathbb{E}_{\bar{s},\bar{a}\sim d^\pi}\mathbb{E}_{s\sim P(\cdot|\bar{s},\bar{a}),a\sim\pi(a|s)}[g(s,a)] \le \frac{1}{\gamma}\mathbb{E}_{s,a\sim d^\pi}[g(s,a)].$$

□

Lemma 6 (Least squares generalization bound, (Song et al., 2023, Lemma 3))**.** *Let $R > 0$, $\delta \in (0,1)$, and consider a sequential function estimation setting with an instance space $\mathcal{X}$ and target space $\mathcal{Y}$. Let $\mathcal{H} : \mathcal{X} \mapsto [-R,R]$ be a class* *of real valued functions. Let* $\mathcal{D} = \{(x_1, y_1), \dots, (x_T, y_T)\}$ *be a dataset of* T *points where* $x_t \sim \rho_t := \rho_t(x_{1:t-1}, y_{1:t-1})$*, and* y_t *is sampled via the conditional probability* $p_t(x_t)$ *(which could be adversarially chosen). Additionally, suppose that* $\max_t |y_t| \leq R$ *and* $\max_h \max_x |h(x)| \leq R$*. Then, the least square solution* $\widehat{h} \leftarrow \operatorname{argmin}_{h \in \mathcal{H}} \sum_{t=1}^{T} (h(x_t) - y_t)^2$ *satisfies*

$$\sum_{t=1}^{T} \mathbb{E}_{x \sim \rho_t, y \sim p_t(x)} \left[(\widehat{h}(x) - y)^2 \right] \leq 3 \inf_{h \in \mathcal{H}} \sum_{t=1}^{T} \mathbb{E}_{x \sim \rho_t, y \sim p_t(x)} \left[(h(x) - y)^2 \right] + 256 R^2 \log(2|\mathcal{H}|/\delta)$$

with probability at least $1 - \delta$*.*