# Adaptive Submodular Policy Optimization

**Anonymous authors**
Paper under double-blind review

**Keywords:** submodularity, adaptive submodularity, policy gradients

## Summary

We propose KL-regularized policy optimization for adaptive submodular maximization. Adaptive submodularity is a framework for decision-making under uncertainty with submodular rewards. The benefit of policy optimization is that we can learn controllers for large action spaces that can utilize state-of-the-art large language model (LLM) priors. The benefit of submodularity are more efficient policy gradient updates because the gradient associated with an action only affects its immediate gain. When the reward model is correctly specified, we prove that our policies monotonically improve as the regularization diminishes and converge to the optimal greedy policy. Our experiments show major gains in statistical efficiency, in both synthetic problems and LLMs.

## Contribution(s)

1. We propose KL-regularized policy optimization for adaptive submodular maximization.
   **Context:** There are prior works on gradient-based optimization of submodular (not adaptive) functions. See Paragraph 2 in Section 6. There are prior works on policy gradients in more general settings. See Paragraphs 1 and 3 in Section 6.
2. We derive more efficient policy gradient estimators than in more general settings, with $O(n)$ terms as opposing to $O(n^2)$, where $n$ is the horizon.
   **Context:** None
3. We prove that our policy converges to the optimal greedy policy for adaptive submodular maximization as the regularization diminishes (Theorem 1).
   **Context:** None
4. We prove that our policies monotonically improve over reference policies used for their regularization as the regularization diminishes (Theorem 4).
   **Context:** None
5. We demonstrate the efficiency of new policy gradient estimators empirically, on both synthetic problems and LLMs (Section 5).
   **Context:** None

# Adaptive Submodular Policy Optimization

**Anonymous authors**
Paper under double-blind review

## Abstract

We propose KL-regularized policy optimization for adaptive submodular maximization. Adaptive submodularity is a framework for decision-making under uncertainty with submodular rewards. The benefit of policy optimization is that we can learn controllers for large action spaces that can utilize state-of-the-art large language model (LLM) priors. The benefit of submodularity are more efficient policy gradient updates because the gradient associated with an action only affects its immediate gain. When the reward model is correctly specified, we prove that our policies monotonically improve as the regularization diminishes and converge to the optimal greedy policy. Our experiments show major gains in statistical efficiency, in both synthetic problems and LLMs.

## 1 Introduction

Many real-world problems have *diminishing returns*. The number of influenced people in a social network increases sublinearly with the number of influencers (Kempe et al., 2003). The information gain due to adding a sensor decreases if other sensors have already been placed at similar locations (Krause et al., 2008). Recommending similar content to already recommended content does not increase engagement (Yue & Guestrin, 2011; Hiranandani et al., 2019). The property of diminishing returns, known as *submodularity*, allows for efficient optimization. Specifically, a greedy algorithm for maximizing submodular functions in $n$ steps is $(1 - 1/e)$-optimal (Nemhauser et al., 1978).

We study adaptive decision making with submodular functions. *Adaptive submodularity* (Golovin & Krause, 2011) is a generalization of submodularity where the expected gain in reward after taking an action, in expectation over its observation, is a submodular function. One application of adaptive submodularity is preference elicitation (Gabillon et al., 2013), which is a special case of question-answering games (Dasgupta, 2005; Karbasi et al., 2012). These problems are submodular because the information gain due to asking a question diminishes with more previously asked questions. A greedy algorithm for adaptive submodular maximization in $n$ steps, which takes the action with the highest expected gain conditioned on the history, is $(1 - 1/e)$-optimal (Golovin & Krause, 2011).

The goal of this work is to bring together submodular and policy optimization, to their mutual benefit. In particular, *policy gradients* (Williams, 1992) arose as a versatile tool for reinforcement learning (Sutton & Barto, 1998) and are behind the recent advances in learning *large language models (LLMs)* (Schulman et al., 2015; 2017; Ouyang et al., 2022). The benefit of casting submodular maximization as policy learning is that we can learn controllers for large action spaces, of all responses of the LLM. The benefit of submodularity in optimization are more efficient policy gradient updates, because the gradient associated with an action only affects its immediate gain. This is in contrast to more general recent frameworks, such as submodular reinforcement learning (Prajapat et al., 2024).

We make the following contributions:

1. We propose KL-regularized policy optimization for adaptive submodularity (Section 3). The benefit of formulating adaptive submodular maximization in this way is that we can learn controllers for large action spaces that can leverage state-of-the-art pre-trained policies, such as LLMs. Our

main contribution to policy optimization are more efficient policy gradient updates, because the gradient associated with an action only affects its immediate gain.

2. We analyze our policies and prove two claims. First, we show that our policy converges to the optimal greedy policy for adaptive submodular maximization as the regularization diminishes. Second, we show that our policies monotonically improve over reference policies used for their regularization as the regularization diminishes. The main contribution in our analysis is bringing together techniques for analyzing KL-regularized policies and adaptive submodular maximization. This requires generalization of existing concepts of near-optimal adaptive submodular policies to stochastic policies, for instance.
3. We empirically evaluate our policies for adaptive submodular maximization. They can be learned more efficiently than using a vanilla policy gradient and are applicable to LLMs.

## 2 Background

We start with introducing our notation. Random variables are capitalized, except for Greek letters like $\theta$. We denote the marginal and conditional probabilities under probability measure $p$ by $p(X = x)$ and $p(X = x \mid Y = y)$, respectively. When the random variables are clear from context, we write $p(x)$ and $p(x \mid y)$. For a positive integer $n$, we define $[n] = \{1, \dots, n\}$. The indicator function is $\mathbb{1}\{\cdot\}$. The $i$-th entry of vector $v$ is $v_i$. If the vector is already indexed, such as $v_j$, we write $v_{j,i}$.

We introduce our multi-step optimization notation next. An *agent* interacts with the environment for $n$ steps. To simplify exposition, we assume that $n$ is fixed. The agent initially observes a *context* $x \in \mathcal{X}$, where $\mathcal{X}$ is the space of contexts. The context is a side information that could define the problem instance, for example. In step $t \in [n]$, the agent takes an *action* $a_t$ and *observes* $y_t$. The difference between actions and observations is that the agent controls the actions. The observations depend on actions but are provided by the environment. The *history* of $n$ actions and their observations is a set $h_n = \{(a_t, y_t)\}_{t \in [n]}$. We denote by $r(x, h_n) \geq 0$ the *reward* associated with context $x$ and history $h_n$. The probability that action $a$ is taken in context $x$ and history $h_{t-1}$ is $\pi(a \mid x, h_{t-1}; \theta)$, and is parameterized by $\theta \in \Theta$. We call $\theta$ a *policy* and $\Theta$ the space of policy parameters. The action and observation in step $t$ are generated as $a_t \sim \pi(\cdot \mid x, h_{t-1}; \theta)$ and $y_t \sim p(\cdot \mid x, h_{t-1}, a_t)$, respectively. Since the order of the observations in the history does not matter, our setting is less general than classic reinforcement learning (Sutton & Barto, 1998) but more general than a bandit (Lattimore & Szepesvari, 2019), because both $a_t$ and $y_t$ depend on the history.

The probability of history $h_n$ in context $x$ under policy $\theta$ factors as

$$\pi(h_n \mid x; \theta) = \prod_{t=1}^{n} p(y_t \mid x, h_{t-1}, a_t)\, \pi(a_t \mid x, h_{t-1}; \theta)\,. \tag{1}$$

This follows from the chain rule and our modeling assumptions. The value of policy $\theta$ is

$$V(\theta) = \mathbb{E}_{x,\, h_n \sim \pi(\cdot \mid x; \theta)} \left[ r(x, h_n) \right] ,$$

where $x \sim \mathcal{D}$ is drawn from a distribution of contexts $\mathcal{D}$. The optimal policy and its value are

$$\theta^* = \arg\max_{\theta \in \Theta} V(\theta)\,, \quad V^* = \max_{\theta \in \Theta} V(\theta)\,, \tag{2}$$

respectively. The question-answering game in Section 1 can be formulated in our notation as follows. The questions are actions, the answers are observations, and the reward is the fraction of objects that the user does not think about, based on the questions and their answers in the history.

### 2.1 Adaptive Submodularity

Adaptive submodularity (Golovin & Krause, 2011) is a framework for sequential decision making under uncertainty with diminishing returns. Under this assumption, a near-optimal policy is greedy conditioned on the history and thus can be computed efficiently.

*Adaptive submodularity* is formally defined as follows. Let

$$\Delta(a \mid x, h_{t-1}) = \mathbb{E}_{y \sim p(\cdot \mid x, h_{t-1}, a)} \left[ r(x, h_{t-1} + \{(a, y)\}) \right] - r(x, h_{t-1}) \tag{3}$$

be the *expected gain* in reward after taking action $a$ in context $x$ and history $h_{t-1}$. We make two assumptions. First, the expected gain is *non-negative*; $\Delta(a \mid x, h_{t-1}) \geq 0$ holds for any context $x$, history $h_{t-1}$, and action $a$. Second, the expected gain is *submodular*,

$$\Delta(a \mid x, h_{t-1}) \geq \Delta(a \mid x, h_{t-1} + \{(a', y')\})$$

holds for any context $x$, history $h_{t-1}$, actions $a$ and $a'$, and observation $y'$. These assumptions are analogous to those in classic submodularity (Nemhauser et al., 1978), except that the ground set are actions and the assumptions are in expectation over the observations of the actions. Similarly to the classic setting, they imply efficiency. Specifically, let

$$\pi_g(a \mid x, h_{t-1}) = \mathbb{1}\left\{ a = \arg\max_{a'} \Delta(a' \mid x, h_{t-1}) \right\} \tag{4}$$

be the greedy policy with respect to $\Delta(a \mid x, h_{t-1})$. Then its expected value is at least $(1 - 1/e)V^*$ (Golovin & Krause, 2011), where $V^*$ is defined in (2).

### 2.2 KL-Regularized Policy Optimization

One limitation of solving adaptive submodular problems as in (4) is that the maximization is difficult when the action space is large or infinite, such as in LLMs (Brown et al., 2020; Wei et al., 2022). This motivates our work on solving (4) as a controller learning problem. Learning of controllers for large action spaces is at the center of *reinforcement learning from human feedback (RLHF)* (Christiano et al., 2017). Specifically, once a reward model is learned, the policy is optimized to maximize the expected reward under the reward model using *proximal policy optimization (PPO)* (Schulman et al., 2017). Specifically, the objective is

$$\mathcal{L}_{\text{PPO}}(\theta) = \mathbb{E}_{x \sim \mathcal{D},\, a \sim \pi(\cdot \mid x; \theta)} \left[ r(x, a) - \beta \log \frac{\pi(a \mid x; \theta)}{\pi_0(a \mid x)} \right], \tag{5}$$

where $x$ is a prompt sampled from a dataset of prompts $\mathcal{D}$, $a$ is its response, and $\pi(a \mid x; \theta)$ is the probability of generating response $a$ to prompt $x$ by policy $\theta$. The first term is the expected reward for response $a$ to prompt $x$. The second term penalizes for deviations of the optimized policy from a *reference policy* $\pi_0$, usually obtained by supervised fine-tuning (Mangrulkar et al., 2022; Hu et al., 2022). The parameter $\beta \geq 0$ trades off the two terms. In adaptive submodularity (Section 2.1), the prompt $x$ and its response $a$ are the history and action, respectively.

PPO is a popular policy-learning framework with two benefits. First, it is suitable for large action spaces. Specifically, once the policy is learned, the best action is just sampled from it. Second, the prior information can be integrated through the reference policy. While PPO has been popularized by RLHF, we note that the idea of KL-regularized policies goes to Schulman et al. (2015), where it was used to motivate trust-region policy optimization; and to Todorov (2006), where it was proposed and analyzed in the context of Markov decision processes (Puterman, 1994).

## 3 Algorithm

We bring together adaptive submodular maximization and KL-regularized policy optimization. This has two benefits. First, we extend adaptive submodular maximization to large action spaces and learning from pre-trained reference policies. Second, KL-regularized policy optimization can be done more efficiently by leveraging adaptive submodularity.

**Algorithm 1** KL-PO

1: **Input:** Learning rate schedule $(\alpha_i)_{i \in \mathbb{N}}$
2: Initialize $\theta$ and $i \leftarrow 1$
3: **while** not convergence **do**
4: Simulate $h_n \sim \pi(\cdot \mid x; \theta)$
5: $\theta \leftarrow \theta + \alpha_i \sum_{t=1}^{n} (f_t(\theta) - \beta) \sum_{\ell=1}^{t} \nabla \log \pi(a_\ell \mid x, h_{\ell-1}; \theta)$
6: $i \leftarrow i + 1$
7: **Output:** Learned policy $\theta$

**Algorithm 2** KL-SubPO

1: **Input:** Learning rate schedule $(\alpha_i)_{i \in \mathbb{N}}$
2: Initialize $\theta$ and $i \leftarrow 1$
3: **while** not convergence **do**
4: Simulate $h_n \sim \pi(\cdot \mid x; \theta)$
5: $\theta \leftarrow \theta + \alpha_i \sum_{t=1}^{n} (f_t(\theta) - \beta) \times \nabla \log \pi(a_t \mid x, h_{t-1}; \theta)$
6: $i \leftarrow i + 1$
7: **Output:** Learned policy $\theta$

### 3.1 Classic Policy Optimization

To understand the benefit of our method, we first introduce a classic $n$-step KL-regularized policy optimization. When actions in (5) are replaced with histories, we immediately obtain

$$\mathcal{L}_{\text{KL-PO}}(\theta, \beta) = \mathbb{E}_\theta \left[ r(x, h_n) - \beta \sum_{t=1}^{n} \log \frac{\pi(h_n \mid x; \theta)}{\pi_0(h_n \mid x)} \right] ,$$

where $\mathbb{E}_\theta\left[\cdot\right] = \mathbb{E}_{x \sim \mathcal{D},\, h_n \sim \pi(\cdot \mid x; \theta)}\left[\cdot\right]$. The problem of policy optimization is to maximize $\mathcal{L}_{\text{KL-PO}}(\theta, \beta)$ with respect to $\theta$. We call this algorithm KL-PO and present it in Algorithm 1. The main challenge is that the gradient of $\mathcal{L}_{\text{KL-PO}}(\theta, \beta)$ has $O(n^2)$ terms. To see this, we first note that

$$\mathbb{E}_\theta\left[r(x, h_n)\right] = \sum_{t=1}^{n} \mathbb{E}_{\theta, t}\left[\Delta(a_t \mid x, h_{t-1})\right] ,$$

where $\mathbb{E}_{\theta,t}\left[\cdot\right] = \mathbb{E}_{x \sim \mathcal{D},\, h_{t-1} \sim \pi(\cdot \mid x; \theta), a_t \sim \pi(\cdot \mid x, h_{t-1}; \theta)}\left[\cdot\right]$. This follows from the factorization of $\pi(h_n \mid x; \theta)$ in (1) and the definition of $\Delta(a_t \mid x, h_{t-1})$ in (3). Therefore, the $n$-step objective is

$$\mathcal{L}_{\text{KL-PO}}(\theta, \beta) = \sum_{t=1}^{n} \mathbb{E}_{\theta, t}\left[f_t(\theta)\right] , \tag{6}$$

where

$$f_t(\theta) = \Delta(a_t \mid x, h_{t-1}) - \beta \log \frac{\pi(a_t \mid x, h_{t-1}; \theta)}{\pi_0(a_t \mid x, h_{t-1})} .$$

Using basic rules of differentiation and the score identity (Aleksandrov et al., 1968), we obtain

$$\nabla \mathbb{E}_{\theta,t}\left[f_t(\theta)\right] = \mathbb{E}_{\theta,t}\left[ (f_t(\theta) - \beta) \sum_{\ell=1}^{t} \nabla \log \pi(a_\ell \mid x, h_{\ell-1}; \theta) \right] . \tag{7}$$

Therefore, the policy gradient (Williams, 1992) of (6) involves $n(n+1)/2$ terms. This leads to an $O(n^2)$ variance in the empirical estimate in KL-PO (line 5). The dependence on prior actions arises because they all impact the gain in step $t$. This motivated many prior works on variance reduction of policy gradients (Sutton et al., 2000; Baxter et al., 2001; Baxter & Bartlett, 2001; Munos, 2006).

### 3.2 Adaptive Submodular Policy Optimization

The key idea in our algorithm is to replace the empirical gradient estimate in KL-PO (line 5), which involves $\sum_{\ell=1}^{t} \nabla \log \pi(a_\ell \mid x, h_{\ell-1}; \theta)$, with $\nabla \log \pi(a_t \mid x, h_{t-1}; \theta)$. An informal justification for this choice is that for any content $x$ and history $h_{t-1}$, a near-optimal policy in (4) only maximizes the immediate gain conditioned on $x$ and $h_{t-1}$.

Mathematically, this change can be viewed as follows. Suppose that (6) is replaced with

$$\mathcal{L}_{\text{KL-SUBPO}}(\theta, \beta) = \sum_{t=1}^{n} \mathbb{E}_{\theta,\theta_h,t}\left[f_t(\theta)\right] , \tag{8}$$

where $\mathbb{E}_{\theta,\theta_h,t}\left[\cdot\right] = \mathbb{E}_{x\sim\mathcal{D},\, h_{t-1}\sim\pi(\cdot|x;\theta_h), a_t\sim\pi(\cdot|x,h_{t-1};\theta)}\left[\cdot\right]$ and $\theta_h$ is a history-generating policy that is independent of $\theta$. Then, using basic rules of differentiation and the score identity (Aleksandrov et al., 1968), we obtain

$$\nabla\mathbb{E}_{\theta,\theta_h,t}\left[f_t(\theta)\right] = \mathbb{E}_{\theta,\theta_h,t}\left[(f_t(\theta) - \beta)\nabla \log \pi(a_t \mid x, h_{t-1}; \theta)\right] . \tag{9}$$

This gradient differs from (7) because we do not differentiate with respect to $\theta_h$. The result is a major gain in efficiency, due to replacing $t$ terms in $\nabla\mathbb{E}_{\theta,t}\left[f_t(\theta)\right]$ by a single one.

We call the resulting algorithm `KL-SubPO` and present it in Algorithm 2. Although (9) has fewer terms than (7), the objective (8) needs to be properly justified and we do that in Section 4. Specifically, we prove that when the problem is adaptive submodular, the maximization of (8) yields near-optimal greedy policies for any history-generating policy $\theta_h$. The learned policies monotonically improve over reference policies $\pi_0$ as $\beta \to 0$ when the reward model is correctly specified.

## 4 Analysis

We make the following assumptions. First, we analyze an idealized variant of `KL-SubPO`, which is formulated as a maximization of (8). Second, we assume that the optimal solution to (8) is realizable and identifiable. Finally, we assume that the reward model is known.

We start with the observation that

$$\mathbb{E}_{\theta,\theta_h,t}\left[f_t(\theta)\right] = \mathbb{E}_{x\sim\mathcal{D},\, h_{t-1}\sim\pi(\cdot|x;\theta_h)}\left[\mathbb{E}_{a_t\sim\pi(\cdot|x,h_{t-1};\theta)}\left[f_t(\theta) \,|\, x, h_{t-1}\right]\right] ,$$

The inner expectation has the same algebraic form as (5). Thus, for any context $x$ and history $h_{t-1}$, the maximizer has a closed form (Todorov, 2006) of

$$\pi(a \mid x, h_{t-1}; \theta) = \frac{1}{Z(x, h_{t-1})}\pi_0(a \mid x, h_{t-1}) \exp\left[\frac{1}{\beta}\Delta(a \mid x, h_{t-1})\right] , \tag{10}$$

where $Z(x, h_{t-1})$ is the normalizer. This allows us to analyze the properties of the optimal policy irrespective of $\theta_h$. In the following, we first show that as $\beta \to 0$, the policy converges to the optimal greedy policy. Then we introduce $\gamma$-approximate policies to analyze the non-asymptotic behavior of `KL-SubPO`.

**Theorem 1.** *Let $\hat{\theta}(\beta) = \arg\max_\theta \mathcal{L}_{\text{KL-SUBPO}}(\theta, \beta)$. Let $\Delta(a \mid x, h_{t-1})$ be the expected gain of taking action $a$ in context $x$ and history $h_{t-1}$, as defined in (3). Let $\pi_g$ be the near-optimal greedy policy in (4). Then, if the best greedy action is unique, for any $x$, $h_{t-1}$, and $a$,*

$$\lim_{\beta\to 0} \pi(a \mid x, h_{t-1}; \hat{\theta}(\beta)) = \pi_g(a \mid x, h_{t-1}) .$$

*Proof Sketch.* When $\beta = 0$, the KL regularizer in (8) vanishes and our policy ends up maximizing $\Delta(a \mid x, h_{t-1})$, which is exactly the greedy policy in (4). See Appendix A for details. □

This result confirms that as the KL regularization diminishes, our policy becomes the greedy policy that maximizes the expected marginal gain. Now we analyze the non-asymptotic behavior through the novel concept of $\gamma$-approximate greedy policies.

### 4.1 $\gamma$-Approximate Greedy Policies

Traditional greedy policies take actions that maximize the expected marginal gain. The solutions to (8) do that only approximately. Therefore, we extend the notion of the marginal gain from individual actions to entire policies. For a policy $\theta$, the expected marginal gain is

$$\Delta(\theta \mid x, h_{t-1}) = \mathbb{E}_{a\sim\pi(\cdot|x,h_{t-1};\theta)}\left[\Delta(a \mid x, h_{t-1})\right] . \tag{11}$$

**Definition 2** ($\gamma$-Approximate Greedy Policy)**.** *For $\gamma \geq 1$, a policy $\theta$ is $\gamma$-approximate greedy if for all contexts $x$ and histories $h_{t-1}$,*

$$\Delta(\theta \mid x, h_{t-1}) \geq \frac{1}{\gamma} \max_{a'} \Delta(a' \mid x, h_{t-1})\,. \tag{12}$$

Our notion of $\gamma$-approximate greedy policies is inspired by but distinct from the approximate greedy policies in Golovin & Krause (2011). While they require every action in the policy's support to be approximately optimal, we only require the approximate optimality of the *expected gain* with respect to a fixed policy. This relaxation is better suited for our setting because we learn stochastic policies, which have large action spaces and are regularized by pre-trained LLM priors using KL.

**Theorem 3** (Performance of $\gamma$-Approximate Greedy Policies)**.** *Let $\theta$ be a $\gamma$-approximate greedy policy and $V^*$ be the expected value of the optimal $n$-step policy. Under the assumptions in Section 2.1,*

$$V^* - V(\theta) \leq (1 - 1/e^{1/\gamma})V^*\,. \tag{13}$$

*Proof Sketch.* The proof follows a standard submodularity argument. We define the optimality gap $X_t = V^* - r(x, h_t)$ and show that $\mathbb{E}[X_t]$ decreases exponentially at rate $1/(\gamma n)$. See Appendix A for details. □

This result generalizes the classic $(1 - 1/e)$-approximation guarantee to approximate greedy policies, with the approximation factor that degrades smoothly with $\gamma$. When $\gamma = 1$, we recover the classic guarantee of the exact greedy policy.

### 4.2 Improvement Guarantees

Having characterized the performance of $\gamma$-approximate greedy policies generally, we now establish how `KL-SubPO` produces improved policies.

**Theorem 4** (Policy Improvement)**.** *Let the reference policy $\pi_0$ in (8) be $\gamma$-approximate greedy. Let $\hat{\theta}(\beta) = \arg\max_\theta \mathcal{L}_{\text{KL-SUBPO}}(\theta, \beta)$ be the optimal solution and $\hat{\pi}(\cdot \mid \cdot) = \pi(\cdot \mid \cdot; \hat{\theta}(\beta))$. Then there exists $\gamma' \in [1, \gamma]$ such that*

$$V^* - r(\hat{\pi} \mid x, h_{t-1}) \leq \left(1 - \frac{1}{\gamma' n}\right)(V^* - r(x, h_{t-1})) \tag{14}$$

*where $r(\hat{\pi} \mid x, h_{t-1}) = \mathbb{E}_{a \sim \hat{\pi}}[r(x, h_{t-1} \cup \{(a, y)\})]$, holds for all contexts $x$ and histories $h_{t-1}$. Furthermore:*

1. *$\hat{\pi}$ is a $(1 - 1/e^{1/\gamma'})$-optimal policy.*
2. *$\gamma'$ decreases monotonically with the regularization parameter $\beta$.*

This theorem establishes two important properties of our `KL-SubPO` policies. First, they improve a $\gamma$-approximate greedy reference policy to a policy with an approximation factor $\gamma' \leq \gamma$. Second, the regularization parameter $\beta$ affects this improvement: a stronger regularization (larger $\beta$) leads to more conservative improvements, while a weaker regularization makes the policy more greedy. The core insight behind this result is the closed-form solution in (10), which indicates monotonicity. We formalize and prove it properly.

## 5 Experiments

We conduct three experiments. The first two experiments are synthetic and the last one is on LLMs.

### 5.1 Linear Maximization

In the first experiment, we study $n$-step maximization of a linear function with $K$ unknown parameters. The function is represented by a vector $w \in \mathbb{R}^K$ where $w_k = (k/K)^2$. The actions are the standard

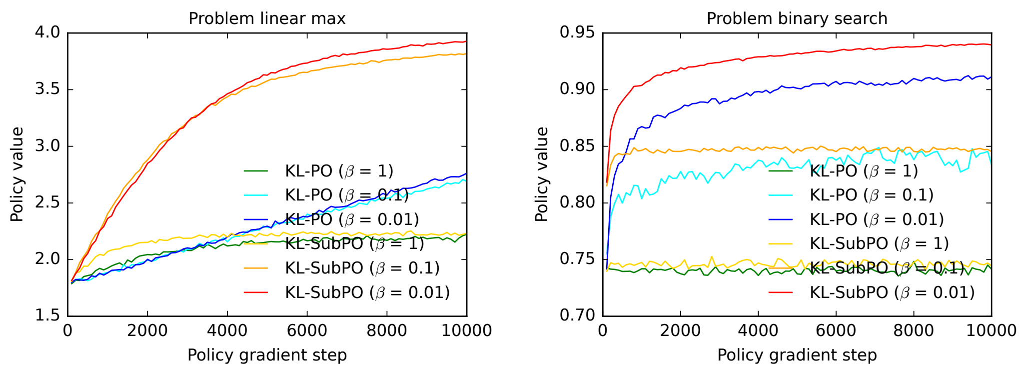


Figure 1: Experiments on the linear maximization problem in Section 5.1 and the binary search problem in Section 5.2.

basis in $\mathbb{R}^K$, $\mathcal{A} = \{e_i\}_{i=1}^K$. The non-zero entry of an action indicates the revealed entry of $w$. The reward is the sum of the revealed entries $r(x, h_t) = \sum_{\ell=1}^t a_\ell^\top w$. The policy is parameterized as $\pi(a \mid x, h_t; \theta) \propto \exp[\phi(h_t, a)^\top \theta]$, where $\phi(h_t, a)$ is the feature vector for history $h_t$ and action $a$. The feature vector for action $e_i$ is a zero vector if the action was taken before and $e_i$ otherwise. Formally, for any $e_i \in \mathcal{A}$ and $k \in [K]$, $\phi_k(h_t, e_i) = e_{i,k} \prod_{\ell=1}^t (1 - a_{\ell,k})$. We set $K = 20$ and the horizon is $n = 5$. The optimal policy selects the 5 highest entries of $w$ and its value is 4.07. We experiment with $\beta \in \{0.01, 0.1, 1.0\}$ to show a range of operating modes of `KL-SubPO`. The reference policy $\pi_0$ is uniform. All policies are optimized by Adam (Kingma & Ba, 2015).

Our results are reported in Figure 1a. We observe three main trends. First, `KL-SubPO` outperforms `KL-PO` for all $\beta$. This shows that `KL-SubPO` is generally more efficient than `KL-PO`. Second, `KL-SubPO` policies improve as $\beta \to 0$ when the reward model is correctly specified (Section 4). Finally, the `KL-SubPO` policy at $\beta = 0.01$ is near optimal.

### 5.2 Binary Search

In the second experiment, we have a binary search problem over $[K]$. A random integer $k_*$ is chosen from $[K]$ and our goal is to identify it. The actions are all possible halving questions on $[K]$. More specifically, $\mathcal{A} = \{q_i\}_{i=1}^{K-1}$, where $q_i \in \{0,1\}^K$ is a vector whose first $i$ entries are ones and the rest are zeros. When the agent takes action $q_i$ in step $t$, the observation is $y_t = q_{i,k_*}$. Simply put, the answer is "yes" if $k_* \leq i$ and "no" otherwise. The reward is the fraction of eliminated integers in $[K]$, that cannot be $k_*$ based on the answers thus far,

$$r(x, h_t) = \frac{1}{K} \sum_{k=1}^K \prod_{\ell=1}^t y_\ell (1 - a_{\ell,k}) + (1 - y_\ell) a_{\ell,k} \,.$$

The policy is parameterized as in Section 5.1. The feature vector for action $q_i$ is an outer product of the state $s_t$, which indicates the remaining integers, and $q_i$, $\phi(h_t, q_i) = \mathrm{vec}(s_t^\top q_i)$. The state is

$$s_{t,k} = \mathbb{1}\left\{ \sum_{\ell=1}^t y_\ell a_{\ell,k} + (1 - y_\ell)(1 - a_{\ell,k}) = t \right\} .$$

We set $K = 32$ and the horizon is $n = 5$. The optimal policy is binary search and its value is 0.97. We experiment with the same policies as in Section 5.1.

Our results are reported in Figure 1b. We observe three main trends. First, `KL-SubPO` performs comparably to `KL-PO` when $\beta$ is high and both policies perform poorly. Second, `KL-SubPO` policies improve as $\beta \to 0$ when the reward model is correctly specified (Section 4). Finally, the `KL-SubPO` policy at $\beta = 0.01$ is near optimal.

### 5.3 Twenty Questions

The last experiment is a 20Q game (Karbasi et al., 2012) with 20 animals. The agent is an LLM. It is optimized against a user represented by an LLM. The reward is the fraction of eliminated animals. The horizon is $n = 6$ questions. The experimental setup is described in detail in Appendix B. We conduct another experiment, where the animals are replaced with Amazon products, in Appendix C.

We let the agent interact with the user and generate a dataset of 200 trajectories of length $n = 6$. The reward of the original LLM is $0.817 \pm 0.006$. We standardize trajectory rewards to zero mean and unit variance, and learn a policy by KL-PO. Its reward is $0.815 \pm 0.006$ and the policy does not improve over the baseline. When the trajectory rewards are clipped at 0, the reward is $0.833 \pm 0.005$ ($2\%$ improvement over the baseline). We also standardize per-step gains to zero mean and unit variance, and learn a policy by KL-SubPO. Its reward is $0.829 \pm 0.006$ ($1.5\%$ improvement over the baseline). When the per-step gains are clipped at 0, the reward is $0.876 \pm 0.004$ ($7\%$ improvement over the baseline). We conclude that KL-SubPO outperforms KL-PO in both settings, irrespective of the rewards being clipped or not.

## 6 Related Work

Our work can be viewed as a special case of submodular reinforcement learning (Prajapat et al., 2024). This is because adaptive submodular functions are set functions, as opposing to functions of sequences of states and actions in Prajapat et al. (2024). These additional properties allow us to derive policy gradients that do not have a quadratic number of terms in the horizon $n$, unlike in Prajapat et al. (2024). The limitations of adaptive submodularity have been noted before and therefore it was extended, for instance to functions of sequences (Mitrovic et al., 2019).

Gradient-based optimization of submodular functions has also been explored before. For instance, Hassani et al. (2017) showed that stochastic projected gradient methods can provide strong approximation guarantees for maximizing continuous submodular functions with convex constraints. Bai et al. (2018) optimized deep submodular functions by gradient ascent. Our paper is the first work on gradient-based optimization of adaptive submodular functions.

Policy gradients were proposed by Williams (1992) and build on the score identity of Aleksandrov et al. (1968). It is well known that policy gradients have a high variance and therefore many variance reduction techniques have been proposed (Sutton et al., 2000; Baxter et al., 2001; Baxter & Bartlett, 2001; Munos, 2006; Kveton et al., 2020). Our contribution to these works is a policy gradient that does not have a quadratic number of terms in the horizon $n$.

## 7 Conclusions

We propose KL-regularized policy optimization for adaptive submodular maximization, a framework for decision-making under uncertainty with submodular rewards. The submodularity allows for more efficient policy gradients than in more general settings. The KL-regularization allows for learning policies for large or infinite action spaces that utilize state-of-the-art LLM priors.

Our analysis makes several simplifying assumptions, which allow us to study the problem more cleanly. First, we analyze an idealized variant of KL-SubPO, which is formulated as a maximization of (8). Second, we assume that the optimal solution to (8) is realizable and identifiable. Finally, we assume that the reward model is known. This is rarely the case in practice and the model has to be estimated. We will address these limitations in our future work.

## References

V. M. Aleksandrov, V. I. Sysoyev, and V. V. Shemeneva. Stochastic optimization. *Engineering Cybernetics*, 5:11–16, 1968.

Wenruo Bai, William Stafford Noble, and Jeff Bilmes. Submodular maximization via gradient ascent: The case of deep submodular functions. In *Advances in Neural Information Processing Systems 31*, 2018.

Jonathan Baxter and Peter Bartlett. Infinite-horizon policy-gradient estimation. *Journal of Artificial Intelligence Research*, 15:319–350, 2001.

Jonathan Baxter, Peter Bartlett, and Lex Weaver. Experiments with infinite-horizon, policy-gradient estimation. *Journal of Artificial Intelligence Research*, 15:351–381, 2001.

Tom Brown et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33*, 2020.

Paul Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems 30*, 2017.

Sanjoy Dasgupta. Analysis of a greedy active learning strategy. In *Advances in Neural Information Processing Systems 17*, pp. 337–344, 2005.

Victor Gabillon, Branislav Kveton, Zheng Wen, Brian Eriksson, and S. Muthukrishnan. Adaptive submodular maximization in bandit setting. In *Advances in Neural Information Processing Systems 26*, pp. 2697–2705, 2013.

Daniel Golovin and Andreas Krause. Adaptive submodularity: Theory and applications in active learning and stochastic optimization. *Journal of Artificial Intelligence Research*, 42:427–486, 2011.

Hamed Hassani, Mahdi Soltanolkotabi, and Amin Karbasi. Gradient methods for submodular maximization. In *Advances in Neural Information Processing Systems 30*, 2017.

Gaurush Hiranandani, Harvineet Singh, Prakhar Gupta, Iftikhar Ahamath Burhanuddin, Zheng Wen, and Branislav Kveton. Cascading linear submodular bandits: Accounting for position bias and diversity in online learning to rank. In *Proceedings of the 35th Conference on Uncertainty in Artificial Intelligence*, 2019.

Edward Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *Proceedings of the 10th International Conference on Learning Representations*, 2022.

Amin Karbasi, Stratis Ioannidis, and Laurent Massoulie. Hot or not: Interactive content search using comparisons. In *2012 Information Theory and Applications Workshop*, pp. 291–297, 2012.

David Kempe, Jon Kleinberg, and Eva Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 137–146, 2003.

Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations*, 2015.

Andreas Krause, Ajit Paul Singh, and Carlos Guestrin. Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research*, 9:235–284, 2008.

Branislav Kveton, Martin Mladenov, Chih-Wei Hsu, Manzil Zaheer, Csaba Szepesvari, and Craig Boutilier. Differentiable meta-learning in contextual bandits. *CoRR*, abs/2006.05094, 2020. URL http://arxiv.org/abs/2006.05094.

Tor Lattimore and Csaba Szepesvari. *Bandit Algorithms*. Cambridge University Press, 2019.

Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, and Benjamin Bossan. Peft: State-of-the-art parameter-efficient fine-tuning methods. https://github.com/huggingface/peft, 2022.

Marko Mitrovic, Ehsan Kazemi, Moran Feldman, Andreas Krause, and Amin Karbasi. Adaptive sequence submodularity. In *Advances in Neural Information Processing Systems 32*, 2019.

Remi Munos. Geometric variance reduction in Markov chains: Application to value function and gradient estimation. *Journal of Machine Learning Research*, 7:413–427, 2006.

G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions - I. *Mathematical Programming*, 14(1):265–294, 1978.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems 35*, 2022.

Manish Prajapat, Mojmir Mutny, Melanie Zeilinger, and Andreas Krause. Submodular reinforcement learning. In *Proceedings of the 12th International Conference on Learning Representations*, 2024.

Martin Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, New York, NY, 1994.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *Proceedings of the 32nd International Conference on Machine Learning*, pp. 1889–1897, 2015.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017. URL https://arxiv.org/abs/1707.06347.

Richard Sutton and Andrew Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 1998.

Richard Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems 12*, pp. 1057–1063, 2000.

Emanuel Todorov. Linearly-solvable Markov decision problems. In *Advances in Neural Information Processing Systems 19*, 2006.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew Dai, and Quoc Le. Finetuned language models are zero-shot learners. In *Proceedings of the 10th International Conference on Learning Representations*, 2022.

Ronald Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3-4):229–256, 1992.

Yisong Yue and Carlos Guestrin. Linear submodular bandits and their application to diversified retrieval. In *Advances in Neural Information Processing Systems 24*, pp. 2483–2491, 2011.

# A Proofs

*Proof of Theorem 1.* This is trivial. When $\beta = 0$, there is the KL-term vanishes from $\mathcal{L}_{\texttt{KL-SubPO}}$. So the optimal policy is the one that maximizes $\Delta(a|x, h_{t-1})$ at every $x, h_{t-1}$. This is exactly what greedy policy does. □

**Lemma 5** (Value Upper Bound)**.** *Let $\pi(| \ x, h_{t-1}, \theta)$ be a $\gamma$-approximate greedy policy and $V^*$ be the expected reward of the optimal $n$-step policy. Then for all contexts $x$ and histories $h_{t-1}$:*

$$V^* \le r\big(x, h_{t-1}\big) + \gamma n \Delta\big(\theta \mid x, h_{t-1}\big),$$

*Proof.* The proof is based on the the usual submodular "each step can't help more than the first step" argument. Let $\pi^*$ be an optimal $n$ steps policy. Then

$$\begin{aligned} V^* - r\big(x, h_{t-1}\big) &\le \mathbb{E}_{h_n \sim \pi*}\left[r(x, h_{t-1} + h_n)\right] - r(x, h_{t-1}) \\ &= \sum_{k=1}^{n} \mathbb{E}\Big[\Delta\big(a_k^* \mid x, h_{k-1} + h_{t-1}\big)\Big] \end{aligned}$$

where $h_{k-1}$ is the history after $k-1$ steps under $\pi^*$ and $a_k^* \sim \pi^*(\cdot \mid x, h_{k-1})$. By *adaptive submodularity*, each incremental gain satisfies

$$\begin{aligned} \Delta\big(a_k^* \mid x, h_{k-1} + h_{t-1}\big) &\le \Delta\big(a_k^* \mid x, h_{t-1}\big) \\ &\le \max_{a'} \Delta(a' \mid x, h_{t-1}) \\ &\le \gamma \Delta(\theta \mid x, h_{t-1}). \end{aligned}$$

Summing over $n$ steps gives

$$V^* - r\big(x, h_{t-1}\big) \le \gamma n \Delta\big(\theta \mid x, h_{t-1}\big).$$

□

**Lemma 6** (One-step Gap Reduction)**.** *Under adaptive submodularity and for any $\gamma$-approximate greedy policy $\pi$, the expected reduction in the optimality gap after one step satisfies:*

$$\mathbb{E}[X_t] \le (1 - 1/(\gamma n))\mathbb{E}[X_{t-1}]. \tag{15}$$

*Proof.* For any realized history $h_t$, and any policy $\pi$ we define the expected one-step reward as:

$$\begin{aligned} r(\pi \mid x, h_t) &:= r(x, h_t) + \mathbb{E}_{a \sim \pi(\cdot|x, h_t; \theta)}[\Delta(a \mid x, h_t)] \qquad (16) \\ &= \mathbb{E}_{a \sim \pi(\cdot|x, h_t; \theta)}[r(x, h_t \cup \{(a, y)\})] \qquad (17) \end{aligned}$$

where the second equality follows from the definition of $\Delta(a \mid x, h_t)$ in (3). By Lemma 5 adaptive submodularity implies:

$$V^* \le r(x, h_{t-1}) + \gamma n \Delta(\pi \mid x, h_{t-1}) \tag{18}$$

This inequality captures the key property that the remaining value after history $h_{t-1}$ is bounded by $\gamma n$ times the one-step gain.

Expanding using the definition of $r(\pi \mid x, h_{t-1})$:

$$\begin{aligned} V^* &\le r(x, h_{t-1}) + \gamma n \Delta(\pi \mid x, h_{t-1}) \qquad (19) \\ &= r(x, h_{t-1}) \qquad (20) \\ &+ \gamma n(r(\pi \mid x, h_{t-1}) - V^* + V^* - r(x, h_{t-1})) \qquad (21) \end{aligned}$$

Rearranging terms gives:

$$V^* - r(\pi \mid x, h_{t-1}) \leq (1 - \frac{1}{\gamma n})(V^* - r(x, h_{t-1})) \tag{22}$$

Note that this holds for every history. Therefore, the result follows by noting that $X_t = V^* - r(x, H_t)$ and taking expectations. □

*Proof of Theorem 3 (Performance of $\gamma$-Approximate Greedy Policies).* Let $H_t$ denote the (random) history after $t$ actions of policy $\pi$. Define the gap random variables $X_t = V^* - r(x, H_t)$, which measure how far we are from optimality after $t$ steps. By Lemma 6 we have that $\mathbb{E}[X_i]$ decreases exponentially:

$$\mathbb{E}[X_t] \leq (1 - 1/(\gamma n))\mathbb{E}\left[X_{t-1}\right]. \tag{23}$$

Iterating this inequality from $t = 1$ to $n$:

$$\mathbb{E}[X_n] \leq (1 - 1/(\gamma n))^n \mathbb{E}[X_0] \tag{24}$$

Since $X_0 = V^* - r(\pi \mid x, H_0)$ where $H_0$ is the empty history, and $\mathbb{E}[X_0] = V^* - V(\theta)$:

$$V^* - V(\theta) \leq (1 - 1/(\gamma n))^n V^* \leq e^{-1/\gamma} V^*. \tag{25}$$

When $\gamma = 1$, we recover the classical $(1 - 1/e)$-approximation of the exact greedy policy. □

**Lemma 7.** *Let $p(x)$ be a probability distribution, and let $g(x)$ be a real valued function. Define $\mathbb{E}_p[g(x)] = \int p(x)\, g(x)\, dx$. Now define a new distribution $p'(x)$ by reweighting $p(x)$ with the factor $e^{g(x)}$: $p'(x) = \frac{p(x)\, e^{g(x)}}{Z}$, where $Z = \mathbb{E}_p[e^{g(x)}] = \int p(x)\, e^{g(x)}\, dx$.*

*Then,*

$$\mathbb{E}_{p'}[g(x)] \quad \geq \quad \mathbb{E}_p[g(x)]$$

*Proof.* We want to show

$$\frac{1}{Z}\, \mathbb{E}_p[e^{g(x)} g(x)] \geq \mathbb{E}_p[g(x)].$$

Equivalently,

$$\mathbb{E}_p[\, e^{g(x)}\, g(x)\,] \quad \geq \quad Z\, \mathbb{E}_p[g(x)] \; = \; \mathbb{E}_p[\, e^{g(x)}\,]\, \mathbb{E}_p[g(x)].$$

Thus it suffices to show

$$\mathbb{E}_p[\, e^{g(x)}\, g(x)\,] \quad \geq \quad \mathbb{E}_p[\, e^{g(x)}\,]\, \mathbb{E}_p[g(x)].$$

Let $Y = g(x)$ be a real-valued random variable under $p$. We claim

$$\mathbb{E}[\, e^Y\, Y\,] \quad \geq \quad \mathbb{E}[\, e^Y\,]\, \mathbb{E}[Y].$$

Rewrite this as

$$\mathbb{E}\big[e^Y (Y - \mathbb{E}[Y])\big] \; = \; \mathrm{Cov}\big(e^Y, Y\big) \; \geq \; 0.$$

But $\mathrm{Cov}\big(e^Y, Y\big) \geq 0$ holds because $e^Y$ is a strictly increasing function of $Y$. By a standard result (e.g., Chebyshev's sum inequality), an increasing function of a random variable is positively correlated with that variable. □

*Proof of Theorem 6 (Policy Improvement).* To establish the theorem, it suffices to show that for all contexts $x$ and histories $h_{t-1}$:

$$\Delta(\hat{\pi}|x, h_{t-1}) \geq \Delta(\pi_0|x, h_{t-1}) \tag{26}$$

This improvement in expected marginal gain directly implies the desired approximation bounds.

| | | | | |
|---|---|---|---|---|
| Dog | Cat | Elephant | Lion | Tiger |
| Giraffe | Panda | Kangaroo | Horse | Penguin |
| Dolphin | Koala | Zebra | Wolf | Shark |
| Eagle | Cheetah | Bear | Monkey | Snake |

Figure 2: Animals in the 20Q game.

From the optimality conditions of `KL-SubPO` in (10), we know that:

$$\hat{\pi}(a|x, h_{t-1}) = \frac{1}{Z(x, h_{t-1})}\pi_0(a|x, h_{t-1}) \exp\left(\frac{1}{\beta}\Delta(a|x, h_{t-1})\right), \tag{27}$$

where $Z(x, h_{t-1})$ is the normalization factor:

$$Z(x, h_{t-1}) = \sum_{a' \in \mathcal{A}} \pi_0(a'|x, h_{t-1}) \exp\left(\frac{1}{\beta}\Delta(a'|x, h_{t-1})\right). \tag{28}$$

Fix any context-history pair $(x, h_{t-1})$. Let $p(a) = \pi_0(a|x, h_{t-1})$ and define $g(a) = \frac{1}{\beta}\Delta(a|x, h_{t-1})$. Then $\hat{\pi}$ can be written as:

$$p'(a) = \frac{p(a)\exp(g(a))}{\sum_{a'} p(a')\exp(g(a'))} \tag{29}$$

By Lemma 7, we have:

$$\mathbb{E}_{a\sim p'}[g(a)] \geq \mathbb{E}_{a\sim p}[g(a)] \tag{30}$$

which directly implies the desired improvement property.

For $\beta_2 < \beta_1$, we can express $\pi(\cdot|\cdot;\hat{\theta}(\beta_2))$ as a reweighting of $\pi(\cdot|\cdot;\hat{\theta}(\beta_1))$:

$$\begin{aligned} \pi(a|x, h_{t-1}; \hat{\theta}(\beta_2)) = &\frac{1}{\hat{Z}(x, h_{t-1})}\hat{\pi}(a|x, h_{t-1}) \\ &\times \exp\left(\frac{1}{\delta}\Delta(a|x, h_{t-1})\right), \end{aligned}$$

where $\delta = 1/\beta_2 - 1/\beta_1$. Applying our previous result twice yields:

$$\begin{aligned} \Delta(\pi(\cdot|\cdot;\hat{\theta}(\beta_2))|x, h_{t-1}) &\geq \Delta(\pi(\cdot|\cdot;\hat{\theta}(\beta_1))|x, h_{t-1}) \\ &\geq \Delta(\pi_0(\cdot|\cdot)|x, h_{t-1}). \end{aligned}$$

This establishes the monotonicity of $\gamma'$ with respect to $\beta$. □

# B Twenty Questions Experiment

The last experiment is a 20Q game (Karbasi et al., 2012) with 20 animals. The agent is represented by an LLM and it is optimized against a user, which is also represented by an LLM. The animals are listed in Figure 2 and the horizon of the game is $n = 6$.

Both the agent and user are implemented using Llama-3.1-8B. The role of the agent is

*You try to guess an animal. Respond with up to 6 words.*

The question of the agent is generated using prompt

*Ask a question.*

It is conditioned on the history of the conversation. The role of the user is

| Question | Answer | Reward |
|---|---|---|
| Does it live on land? | Yes | 0.100 |
| Does it have four legs? | Yes | 0.200 |
| Does it have a tail? | Yes | 0.200 |
| Does it primarily eat plants? | No | 0.600 |
| Does it have sharp claws? | Yes | 0.600 |
| Is it a carnivorous mammal? | Yes | 0.600 |

Figure 3: One 20Q game between the user and agent. The target animal is dog.

```
Bluetooth Speaker   Phone Charger   Air Fryer   Yoga Mat
Water Bottle    Ring Doorbell   Echo Dot    Wireless Earbuds
Protein Powder  LED Strip Lights    Portable Power Bank
Coffee Maker    Weighted Blanket Desk Lamp  Wireless Mouse
Reusable Straws    Robot Vacuum     Shower Curtain
Cast Iron Skillet Kindle Paperwhite
```

Figure 4: Products in the Amazon selection game.

*Answer with Yes or No. No period.*

The answer of the user is generated by prompt

*You think of [animal]. You are asked: [question]*

where [animal] is replaced by the target animal name from Figure 2 and [question] is replaced by the last question of the agent. The reward is the fraction of eliminated animals. The animal is eliminated if at least one property of the animal disagrees with at least one answer of the user. One conversation between the user and agent is shown in Figure 3.

## C Amazon Product Selection Experiment

The last experiment is a product selection game on a set of 20 Amazon products. The agent is tasked with narrowing down to a specific product by asking yes/no questions. The products are listed in Figure 4 and the horizon of the game is $n = 4$.

Both the agent and user are implemented using Llama-3.1-8B. The agent is provided with the system message:

*You are playing a 20 Questions game to guess an Amazon product from this list: [list of products]. Ask clear yes/no questions to efficiently narrow down the possibilities. Keep questions concise (ideally under 10 words). The user will only respond with Yes or No.*

The question of the agent is generated using prompt:

*Ask a question.*

It is conditioned on the history of the conversation. The user's response is generated by prompt:

*You think of [product]. You are asked: [question]*

where [product] is replaced by the target product name from Figure 4 and [question] is replaced by the last question of the agent. The reward is the fraction of eliminated products. A product is eliminated if its response to a question differs from the target product's response to the same question. This reward calculation creates a natural submodular structure as questions eliminate overlapping subsets of products.

| Question | Answer | Reward |
|---|---|---|
| Is the product electronic? | Yes | 0.350 |
| Can the product be held in your hand? | Yes | 0.550 |
| Does the product plug into a wall outlet? | No | 0.850 |
| Does the product require charging? | Yes | 0.850 |

Figure 5: Example run of the Amazon product selection game.

The baseline model achieves a reward of $0.837 \pm 0.005$. We compare this with various configurations of our methods: (1) `KL-SubPO` with standardized trajectory rewards achieves $0.841 \pm 0.004$; (2) `KL-SubPO` with clipped rewards from below at 0 achieves $0.858 \pm 0.004$ ; (3) `KL-PO` with standardized per-step gains achieves $0.828 \pm 0.005$; (4) `KL-PO` with clipped rewards from below at 0 achieves $0.847 \pm 0.004$.