MultiScale Contextual Bandits for Long Term Objectives

Richa Rastogi **Yuta Saito** **Thorsten Joachims**
Department of Computer Science
Cornell University

Abstract

The feedback that AI systems (e.g., recommender systems, chatbots) collect from user interactions is a crucial source of training data. While short-term feedback (e.g., clicks, engagement) is widely used for training, there is ample evidence that optimizing short-term feedback does not necessarily achieve the desired long-term objectives. Unfortunately, directly optimizing for long-term objectives is challenging, and we identify the disconnect in the timescales of short-term interventions (e.g., rankings) and the long-term feedback (e.g., user retention) as one of the key obstacles. To overcome this disconnect, we introduce the framework of MultiScale Policy Learning to contextually reconcile that AI systems need to act and optimize feedback at multiple interdependent timescales. Following a PAC-Bayes motivation, we show how the lower timescales with more plentiful data can provide a data-dependent hierarchical prior for faster learning at higher scales, where data is more scarce. As a result, the policies at all levels effectively optimize for the long-term. We instantiate the framework with **M**ulti**S**cale Off-Policy **B**andit **L**earning **(MSBL)** and demonstrate its effectiveness on three tasks relating to recommender and conversational systems.

1 Introduction

Many interactive AI systems (e.g., recommender systems, conversational systems) use abundantly collected short-term feedback for learning. However, it is well known that over-optimization of short-term feedback can adversely affect the long-term goals [14, 21, 26]. Similarly, challenges with competing time horizon objectives [40, 15], and issues like reward hacking [35, 29, 30] and user manipulation [18, 11] have been previously reported. For example, optimizing for engagement on social media platforms can lead to clickbait-y or toxic feeds. This neither reflects the user preferences nor the platform's goals of retaining users. We wish to design systems that optimize for long-term objectives, that are beneficial for various stakeholders in the system [1].

A key problem in achieving this goal is that the long-term feedback (e.g., user retention) is at a different timescale than the short-term interventions (e.g., rankings of recommended products) that are used to optimize them. For instance, optimizing rankings for clicks in a recommender system is relatively straightforward since clicks can be attributed to an individual ranking. Indeed, learning methods such as bandits that optimize for users' immediate response to recommendations such as clicks, likes, views, etc are widely used to learn ranking policies [19, 46]. Now, consider optimizing rankings for the feedback on users' subscription renewal, which is observed monthly. We note the disconnect between the timescale of rankings presented and the long-term objective of subscription renewal, making this a much harder problem than the previous one. While a Markov Decision Process can in principle model sequential dependencies to achieve long-term goals, in practice the large resulting state spaces, credit-assignment problems, and the sparsity of long-term feedback prohibit straightforward applications of reinforcement learning [23].

Workshop on Multi-Turn Interactions in LLMs at NeurIPS 2025.

To overcome this problem, our approach is to contextually reconcile the disconnect between short-term and long-term objectives by learning interventions and policies at multiple timescales. Consider a recommender system for a video streaming platform as depicted in Figure 1. At the lowest level, the system acts by providing rankings of recommended products. These rankings (short-term interventions) can be optimized for clicks (short-term reward). While clicks are an important signal for learning that is abundantly available at the lowest level of Figure 1, an unmitigated maximization of clicks is not necessarily aligned with higher-level goals. In particular, the platform may sacrifice some clicks if that leads to a higher weekly return rate. This metric is a more reliable indicator of user satisfaction, but it is available at a slower timescale. Finally, at the highest level, the platform ultimately cares about user retention and subscription renewal. This feedback lives at an even slower timescale, and it is thus more scarce but even more valuable. Similar multi-scale levels of feedback metrics also exist for other AI systems (e.g., tutoring chatbots that aim to achieve long-term learning outcomes), and additional settings are discussed in Appendix B.

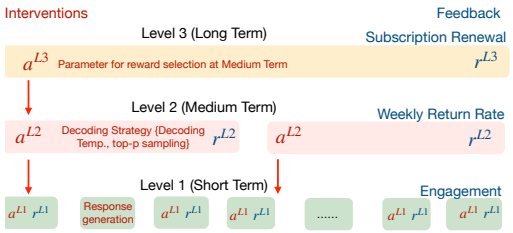


Figure 1: MultiScale feedback r with corresponding interventions a at each level $(L1, L2, L3)$. At the short-term level, engagement feedback (e.g., responses, clicks) is observed at the fastest timescale. At the next higher level, we observe feedback like the weekly return rate. And at an even higher level, subscription renewal is observed at the slowest timescale.

The main contributions of this work are as follows:

1. We introduce a new framework – **MultiScale Policy Learning –** that formalizes a multi-level approach for optimizing long-term objectives. The framework introduces a recursive construction of priors that are informed by data at the lower levels to speed up learning at the higher levels.
2. We demonstrate the practicality and generality of this framework by developing **MultiScale Off-Policy Bandit Learning (MSBL)** as a simple recursive algorithm for training multi-scale contextual bandit policies. It includes two widely applicable constructions for nested training of policies that enable the use of both abundant short-term and sparse long-term data for optimizing long-term outcomes.
3. We demonstrate the effectiveness of our approach empirically on three tasks ranging from recommender to conversational systems. Ablations show robustness of our method for optimizing the long term reward.

2 Related Works

Hierarchical Reinforcement Learning. Hierarchical RL (HRL) approaches, such as the options framework [37, 3] and feudal learning [7], learn to operate on different levels of temporal abstraction. HRL aims to accomplish complex tasks by dividing them into sub-goals with the higher level assigning sub-tasks to the next lower level, such as in robotic applications [31]. While our framework shares the idea of creating a hierarchy, the type of hierarchy is fundamentally different. In particular, our tasks do not have a subgoal decomposition as in HRL, where the discovery of the subgoals and their execution is critical and guides the micro policy. Instead, we exploit the hierarchy of feedback timescales to construct a hierarchical prior over the policy space to speed up learning for sparse long-term feedback. Our goal is to steer towards improving long-term outcomes, even at the expense of shorter-term rewards. This is fundamentally different from the macro actions (options) in HRL, which are abstractions over micro actions, and the goal is to combine these options as subroutines.

Long Term Optimization. There has been a growing interest in the study of recommender systems that go beyond optimizing engagement and clicks [25, 5, 23, 26, 6, 2]. Maystre et al. [23] provide an RL perspective to long-term learning in recommender systems and discuss challenges with credit attribution. Other approaches, such as [5] do not learn at the macro level and instead formulate long-term macro interventions that they aim to fulfill with minimum impact on short-term engagement metrics. These works operate on a single timescale and assume that the macro intervention is given and fixed. Our work can be viewed as learning the macro interventions themselves from long-term

feedback. In fact, these methods can be used for a single level-specific learning within our MultiScale framework, making them a special case of our approach.

Multi-Objective Optimization. Recent work has highlighted the importance of selecting weights for linear scalarization for multi-objective learning in recommender systems [27, 16, 42] and in text generation tasks [43]. A different line of work explores learning conditional policies as a single family of policies [9, 42, 43]. Our work leverages the idea of conditional policies and multi-objective learning from the perspective of optimizing for long-term outcomes. In doing so, we elevate single-stage learning to multiple levels with the macro level contextually selecting the objective to optimize at the micro level.

Additional related works are deferred to Appendix C.

3 Multi-Scale Policy Framework

We begin by providing a PAC-Bayesian motivation for why a hierarchical approach that exploits feedback across multiple scales can be substantially more data efficient. For simplicity of notation, we restrict to two levels – the micro level operating at the faster timescale $t1$ and the macro level operating at the slower timescale $t2$. As we will see later, the framework naturally extends to multiple scales of policy learning.

Our goal is to learn a policy $\pi(a|x)$ that selects an action a for a given context x. We assume that contexts are drawn i.i.d. from an unknown distribution $x \sim p(x)$, but we conjecture that our approach can be extended to stateful models as well. Our approach rests on the realization that we observe rewards at different timescales in many AI systems. For two levels, we observe a reward

$$r^{L1} \sim p(r^{L1}|x, a)$$

at the micro level for each action a and context x, corresponding to the short-term engagement (e.g., clicks). We also record rewards

$$r^{L2} \sim p(r^{L2}|(x_i, a_i), \dots (x_{i+T}, a_{i+T}))$$

after we have taken a sequence of actions. This corresponds to the long-term feedback (e.g., weekly returns, subscription renewal) which we would like to optimize, since it typically better reflects stakeholder objectives. This leads us to the following policy learning objective,

$$\pi^{L2*} \leftarrow \arg\max_{\pi \in \Pi} V^{L2}(\pi), \quad \text{where } V^{L2}(\pi) = \mathbb{E}_{x,a,r^{L2}}[r^{L2}]. \tag{1}$$

Unfortunately, for a large and complex policy space Π, finding the optimal policy π^* by simply replacing the expected reward $V^{L2}(\pi)$ with its empirical estimate $\hat{V}^{L2}(\pi)$ on some training data D^{L2} is typically intractable. The long-term reward is too infrequent for learning policies from scratch. As a result, existing approaches predominantly [44] optimize the more frequent reward signal r^{L1} at the micro level.

$$\pi^{L1*} \leftarrow \arg\max_{\pi \in \Pi} V^{L1}(\pi), \quad \text{where } V^{L1}(\pi) = \mathbb{E}_{x,a,r^{L1}}[r^{L1}]. \tag{2}$$

Note that even the policy π^{L1*} that perfectly optimizes the micro level reward can have substantially worse reward at the macro level, $V^{L2}(\pi^{L1*}) < V^{L2}(\pi^{L2*})$. However, π^{L1*} is typically much better than a random policy from Π. This raises the following question.

How do we exploit feedback at the micro level to learn faster at the macro level? To provide a theoretical motivation, we make the following PAC-Bayesian argument. PAC Bayes generalization bounds [20, 24] provide uniform convergence over all posterior distributions $Q(\pi)$ for any given prior distribution $P(\pi)$. Specifically, with probability $1 - \delta$,

$$\left|\mathbb{E}_{\pi \sim Q}[V^{L2}(\pi)] - \mathbb{E}_{\pi \sim Q}[\hat{V}^{L2}(\pi; D)]\right| \leq O\left(\sqrt{\frac{KL(Q||P) + ln(1/\delta)}{n}}\right). \tag{3}$$

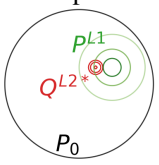
Since this bound holds for all Q, it also holds for any (approximately) optimal posterior Q^{L2*} that maximizes the macro level reward. For a discrete policy space, Q^{L2*} is the Dirac delta distribution centered on π^{L2*}. The bound states that this learning problem is 'easy' (i.e., requires a small number n of training examples) if the KL-divergence between Q^{L2*} and the prior P is small.

Can we improve the prior for the macro level with data from the micro level? While the optimal policy π^{L1*} at the micro level may be suboptimal at the macro level, a policy that is learned based on finite data at the micro level via $\hat{\pi}^{L1} \leftarrow \arg\max_{\pi \in \Pi} \hat{V}^{L1}(\pi)$ can provide useful prior information for learning at the macro level.

Consider the following illustrative example, where each policy $\pi(.|.,\theta) \in \Pi$ is defined via a parameter vector θ. We denote θ^{L1} as the parameters of policy $\hat{\pi}^{L1}$, and θ^{L2*} as those of the optimal macro policy π^{L2*}. For simplicity of demonstration, we define the target policy distribution as $Q^{L2*} = N(\theta^{L2*}, \Sigma^{L2})$, an uninformed prior distribution $P_0 = N(\theta_0, \Sigma_0)$ for some arbitrary θ_0, and an informed prior $P^{L1} = N(\theta^{L1}, \Sigma^{L1})$ centered at the learned micro policy $\hat{\pi}^{L1}$. The difference in training samples $n_0 - n^{L2}$ to get the same confidence interval for Q^{L2*} in Equation (3) is proportional to $KL(Q^{L2*}||P_0) - KL(Q^{L2*}||P^{L1})$. We show in Section E.1 that for an appropriately chosen Σ^{L1}, the improvement in required training samples by moving to the informed prior P^{L1} is at least

$$n_0 - n^{L2} \propto KL(Q^{L2*}||P_0) - KL(Q^{L2*}||P^{L1}) \in O(|\theta^{L2*} - \theta_0|_M - |\theta^{L2*} - \theta^{L1}|_M) \quad (4)$$

where $|\theta^{L2*} - \theta_0|_M$ is the squared Mahalanobis distance in the parameter space θ. This shows that the policy $\hat{\pi}^{L1} \sim P^{L1}$ learned at the micro level can provide training sample savings at the macro level, if $\hat{\pi}^{L1}$ and π^{L2*} are close compared to the uninformed prior P_0. In particular, if we can learn almost all parameters of π^{L2*} at the micro level, the distance $|\theta^{L2*} - \theta^{L1}|_M$ will be small, resulting in higher macro level sample savings. The figure on the right illustrates how an informed prior P^{L1} pulls the center of the prior P_0 closer to the parameters of $\pi^{L2*} \sim Q^{L2*}$.



To quantify the reduction in macro level training data, we construct a numerical example (detailed in Section E.1) with $\theta \in \mathbb{R}^{50}$, such that 49 parameters are learned well at the micro level and only 1 parameter needs to be adjusted at the macro level. This results in saving $\approx 98\%$ training samples[1], which means gathering enough macro level data may only take weeks instead of years.

4 Multi-Scale Policy Learning

In order to put the theoretical insights from the previous section into a practical policy learning algorithm, we introduce a factorization of contexts and policies at each level. We propose the policy factorization in a way that the micro level learns a large part of the parameter space, even if it is not aligned with the long term expected reward. This simplifies learning at macro level compared to learning from scratch.

Multi-Scale Contexts. The context $x \sim p(x)$ can be factorized as $p(x) \triangleq p(x^{L2}) \cdot p(x^{L1}|x^{L2})$. At the upper level, contexts $x^{L2} \sim p(x^{L2})$ arrive at timescale $t2$. An example of an upper-level context could be a user as described by demographic features or some long-term profile. For each such upper-level context, a sequence of lower-level contexts $x^{L1} \sim p(x^{L1}|x^{L2})$ is drawn conditionally on x^{L2}. Such lower-level contexts could be search queries, or chat requests.

Multi-Scale Policies. We consider the following factorization of policy space Π,

$$\Pi \triangleq \Pi^{L1} \cdot \Pi^{L2}, \quad (5)$$

where Π^{L1} consists of micro policies, that as we will see later, can provide a strong inductive bias for the long term optimal policy. In contrast, the macro level is only concerned with learning within the space of reasonable policies obtained after micro learning and has a much smaller policy space Π^{L2}.

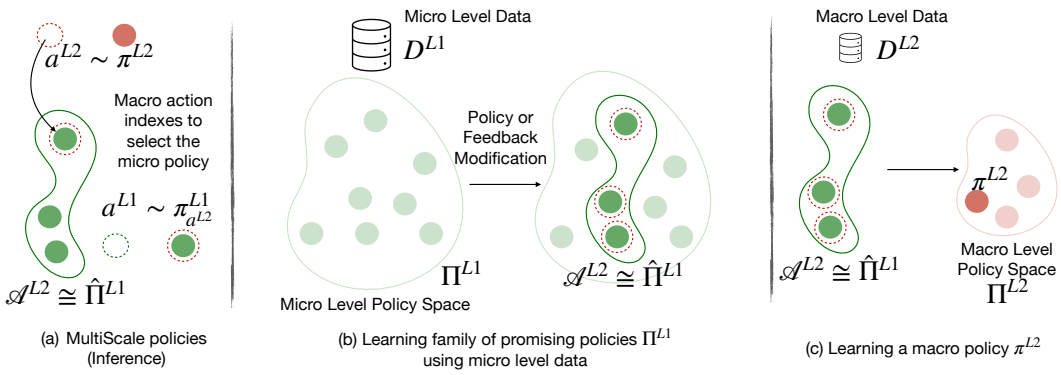
Specifically, for each upper-level context x^{L2}, the upper-level policy π^{L2} selects an action a^{L2} from action space $\mathcal{A}^{L2}$. Examples of a^{L2} are diversity boosts, aggressiveness of spam filtering, the decoding strategy of an LLM policy, etc.

$$a^{L2} \sim \pi^{L2}(a^{L2}|x^{L2}) \quad ; \quad \pi^{L2} \in \Pi^{L2}$$

Importantly, as shown in Figure 2 (a), the action a^{L2} corresponds to a lower-level policy $\pi^{L1}_{a^{L2}}$. This means that the action space $\mathcal{A}^{L2}$ at the upper level is isomorphic to a family of policies $\hat{\Pi}^{L1} = \{\hat{\pi}^{L1}_{a^{L2}} : a^{L2} \in \mathcal{A}^{L2}\}$, learned empirically at the lower level.

$$\mathcal{A}^{L2} \cong \hat{\Pi}^{L1}$$

[1]This calculation does not consider the investment of training samples n^{L1} for constructing P^{L1}. Generally, these n^{L1} samples are significantly cheaper than the macro samples n^{L2}, since they are T times more frequent.



(a) MultiScale policies (Inference)

(b) Learning family of promising policies Π^{L1} using micro level data

(c) Learning a macro policy π^{L2}

Figure 2: (a) At inference, a macro action indexes to select the particular micro policy from a family of micro policies. The macro action space $\mathcal{A}^{L2}$ is isomorphic to the family of policies $\hat{\Pi}^{L1}$ (b) Learning micro policies: Abundant micro-level data is used to learn promising policies $\hat{\Pi}^{L1}$ using policy or feedback modification (c) Macro-level data is used to learn a macro policy. For more than two levels, (b) and (c) are recursively called, narrowing down micro policy space/ macro action space.

The lower-level policy $\pi^{L1}_{a^{L2}}$ indexed by the chosen upper-level action a^{L2} will be executed for the subsequent lower-level contexts.

$$a^{L1} \sim \pi^{L1}_{a^{L2}}(a^{L1}|x^{L1})$$

The actions $a^{L1} \in \mathcal{A}^{L1}$ at the lower level are rankings, chat responses, or push notifications, just like in the conventional contextual bandit framework.

Multi-Scale Data. We collect contextual bandit feedback at both the upper and the lower level. At the lower level, we get the conventional data

$$D^{L1} = \{(x_i^{L1}, a_i^{L1}, r_i^{L1}, p_i^{L1})\}_{i=1}^{n^{L1}},$$

where we use the logging policy π_0^{L1} and record the propensity $p_i^{L1} = \pi_0^{L1}(a_i^{L1}|x_i^{L1})$ to enable off-policy learning. However, unlike for conventional bandit policies, we also get data for the upper-level, which includes the long-term rewards

$$D^{L2} = \{(x_i^{L2}, \pi^{L1}_{a_i^{L2}}, r_i^{L2}, p_i^{L2})\}_{i=1}^{n^{L2}} \quad \text{with propensity} \quad p_i^{L2} = \pi_0^{L2}(a_i^{L2}|x_i^{L2}).$$

We thus need to devise an algorithm for learning both $\hat{\pi}^{L2}$, as well as each of the policies in $\hat{\Pi}^{L1}$. We approach this problem by using the upper-level data D^{L2} for learning a^{L2}, and we use the lower-level data D^{L1} to learn a comparably small set of policies $\hat{\Pi}^{L1}$ that serve as actions for the upper-level. This provides the affordance to use the abundant feedback in D^{L1} for learning in the large action space $\mathcal{A}^{L1}$, thus narrowing down the set of potential actions for learning π^{L2} from the comparably scarce data in D^{L2}, as shown in Figure 2 (b) and (c).

4.1 Learning a Family of Micro Policies

The following proposes two options, which both define a comparably small $\hat{\Pi}^{L1}$ based on the more abundant data D^{L1} available at the lower level.

Policy Modification. In this procedure, we first learn a single policy π^{L1} from D^{L1} that we then modify to define the family $\hat{\Pi}^{L1}$. In particular, we first train $\hat{\pi}^{L1}$ to optimize the expected reward at the lower level according to Eq. (2). This single policy is then modified by each action a^{L2}, where each action takes the form of a function from $\Pi^{L1} \to \hat{\Pi}^{L1}$.

$$\hat{\pi}^{L1}_{a^{L2}} := a^{L2}(\hat{\pi}^{L1}) \tag{6}$$

In the example of text generation, a decoding strategy a^{L2} might modify the learned LLM policy for more varied response generations. Similarly, applying a boost a^{L2} to items of a particular type in a recommender system updates the ranking policy $\pi^{L1}_{a^{L2}}$ by ranking boosted items higher up, or items suspected to be click-bait lower. As a result, Eq. (6) defines policy update $\hat{\pi}^{L1}_{a^{L2}}$ for a given

intervention a^{L2} as a form of perturbing the short-term optimized policy $\hat{\pi}^{L1}$. While $\hat{\pi}^{L1}_{a^{L2}}$ could provide a lower expected short-term reward as compared to that from $\hat{\pi}^{L1}$, it can be more effective at optimizing the reward at the upper-level (e.g., fewer clicks by more aggressively pruning suspected click bait can lead to better weekly returns).

Feedback Modification. In this procedure, each upper-level action a^{L2} takes a form of a loss function that acts upon the observed feedback at the lower level when learning the lower-level policy. In this case we assume that the feedback r^{L1} is vectorial (e.g., clicks, likes, purchases, add-to-carts), and each a^{L2} is a different function (e.g., convex combination) for combining the feedback vector into a scalar loss. Then, for any given a^{L2}, we optimize

$$\pi^{L1}_{a^{L2}} := \arg\max_{\pi^{L1}\in\Pi^{L1}} \mathbb{E}_{p(x^{L1}),\pi^{L1}(a^{L1}|x^{L1}),p(r^{L1}|x^{L1},a^{L1})}[a^{L2}(r^{L1})] \tag{7}$$

to get a family of policies $\hat{\Pi}^{L1}$.

For a more efficient implementation that does not require us to explicitly enumerate the policies in $\hat{\Pi}^{L1}$, we include a^{L2} in the context and parameterize the reward $a^{L2}(r^{L1})$ during training π^{L1} for every $a^{L2} \in \mathcal{A}^{L2}$. In this way, we only learn a single policy that is parameterized by a^{L2} to represent all policies in $\hat{\Pi}^{L1}$. At inference, a^{L2} chosen from the learned macro policy is included in the context of the micro policy, selecting the particular $\pi^{L1}_{a^{L2}}$. As a result, while Eq. (7) in principle refers to a discrete set of policies, practically we only train one micro policy. For LLM policies, this can be implemented as described in [43].

Note again that the transformation of the rewards can lead to lower short-term reward on some primary metric (e.g., clicks), but that the upper-level policy now has a space of actions that can optimize the longer-term metric (e.g., user retention).

4.2 Learning the Macro Policy

Once the family of policies $\hat{\Pi}^{L1}$ that correspond to the upper-level actions $a^{L2} \in \mathcal{A}^{L2}$ is fixed, we can chose from a wide range of policy-learning methods that use the data in D^{L2} to optimize the expected upper-level reward $V^{L2}(\pi^{L2})$. In particular, we can use off-policy policy-gradient methods that optimize the inverse propensity weighted empirical average

$$\hat{\pi}^{L2} = \arg\max_{\pi^{L2}(.|.,\theta)} \frac{1}{n^{L2}} \sum_{i=1}^{n^{L2}} \frac{\pi^{L2}(a^{L2}_i|x^{L2}_i,\theta)}{p^{L2}_i} r^{L2}_i \tag{8}$$

as an estimate of the expected upper-level reward $V^{L2}(\pi^{L2})$. If the policy $\hat{\pi}^{L2}(.|.,\theta)$ is differentiable in its parameters θ, we can use stochastic-gradient descent for training [17]. As shown in Figure 2(c), we find the macro policy $\pi^{L2} \in \Pi^{L2}$ decoupled from learning a family of policies at the micro level.

4.3 MultiScale Bandit Learning Algorithm

We can now summarize our approach to learning a nested set of policies across multiple levels in Algorithm 1. The algorithm uses off-policy contextual bandits [10, 38, 33] at each level. Algorithm 1 is limited to two levels for conciseness of notation, but the full recursive procedure for an arbitrary number of levels is given in Appendix D.2. In the experiments, we will explore policy spaces with two and three levels.

Algorithm 1 MultiScale Training: Off-Policy Contextual Bandits

Procedure *PolicyLearning*(π^{L2}_0, π^{L1}_0)
- Collect Micro Logged dataset $D^{L1} := \{(x^{L1}_i, a^{L1}_i, r^{L1}_i, p^{L1}_i)\}_{i=1}^{n^{L1}} \sim \pi^{L1}_0$
- Learn Micro policies $\hat{\Pi}^{L1}$(Eq. (6) or (7) using D^{L1})
- Collect Macro Logged dataset $D^{L2} := \{(x^{L2}_j, a^{L2}_j, r^{L2}_j, p^{L2}_j)\}_{j=1}^{n^{L2}} \sim \pi^{L2}_0$
- Learn Macro Policy $\hat{\pi}^{L2} \leftarrow \arg\max_{\pi^{L2}} \hat{V}^{L2}(\pi^{L2}; D^{L2})$ (Eq. (8))
- **return** learned policies $\hat{\pi}^{L2}, \hat{\Pi}^{L1}$

The procedure requires the logging policies π^{L2}_0, π^{L1}_0 as input. We first collect logged bandit data D^{L1} and learn the micro policies either as policy or feedback modification to get a family of policies

$\hat{\Pi}^{L1} := \{\hat{\pi}^{L1}_{a^{L2}} : a^{L2} \in \mathcal{A}^{L2}\}$. As shown in Figure 2(b), the entire policy space Π^{L1} can be viewed as consisting of different sets of $\hat{\Pi}^{L1}$. With the learned policies $\hat{\Pi}^{L1}$, we now collect logged bandit data D^{L2} for the macro level. Note that the logged datasets D^{L1} and D^{L2} can be collected asynchronously, because during micro policy learning, we learn $\hat{\Pi}^{L1}$ for all $a^{L2} \in \mathcal{A}^{L2}$.

The algorithm and MSPL framework in general decomposes the policy space Π (Eq. (5)) into decoupled and independent learning at each level. This decoupled learning has the advantage that the policies can be updated asynchronously after deployment.

Overall, at each lower level $L(k-1)$, we use data $D^{L(k-1)}$ to learn the promising policies $\hat{\Pi}^{L(k-1)}$ that provide prior information and serve as the action space for the next upper level $L(k)$. While the training involves bottom-up learning of the policies, deployment involves top-down inference from learned policies at each of the levels. Figure 2 (a) shows the inference with actions chosen from the upper-most level policy, indexing the next lower level policy. The selection is conditional, where a^{L2} is first selected from the learned $\hat{\pi}^{L2}(.|x^{L2})$ and the lower level action a^{L1} is selected according to $\hat{\pi}^{L1}_{a^{L2}}(.|x^{L1})$. In this way, learning a family of policies enables adaptation for the micro level at inference time. We provide a formal inference algorithm in Section D.

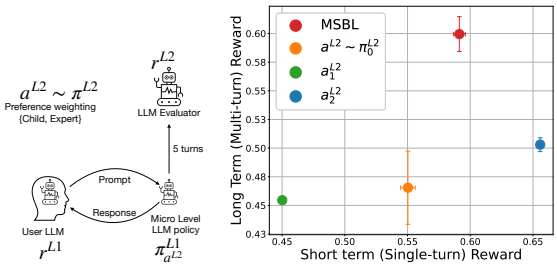
Since we use off-policy learning independently at each level, convergence for the contextual bandit learning at each level depends on the number of samples and the action space [10]. As a result, with a smaller macro action space $\mathcal{A}^{L2}$ compared to $\mathcal{A}^{L1}$, we can utilize the scarcer data at the upper level and get similar convergence as the micro policy that utilizes the more abundant data.

5 Experiments

We conduct experiments on three scenarios related to conversational systems (Anthropic Helpful Assistant [4] and a new simulator), and a conventional recommendation system (KuaiRand video streaming benchmark) for two and three timescale levels. We provide an additional experiment on a toy domain that is simple enough to make RL tractable in Section F.1, and as expected, we find that our approach is competitive. In the following experiments, we compare our approach against single-stage policies that cannot personalize at the macro level, a random baseline policy (denoted by $a^{L2} \sim \pi_0^{L2}$) that selects interventions uniformly, and an oracle skyline policy. We use subscript index to denote the static policies, e.g., a_1^{L2} is fixed action policy that always applies first macro intervention. The complete experiment setup, hyperparameters, and training details are provided in Section F.

5.1 Multi-turn Conversation

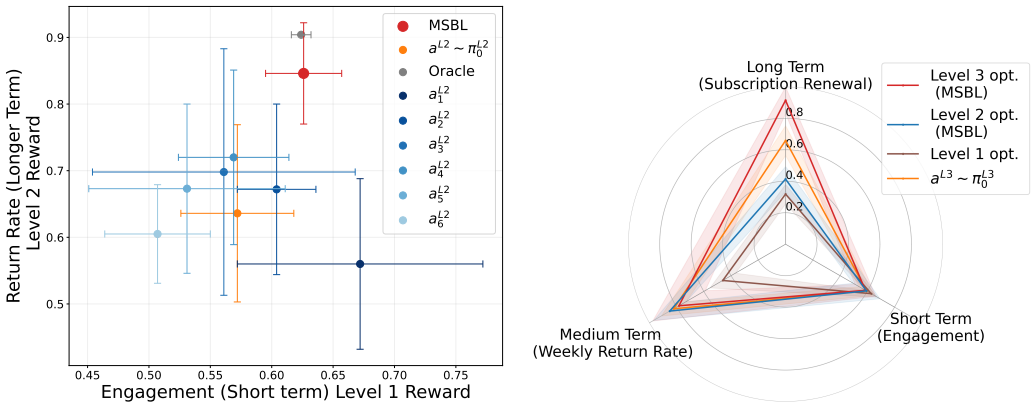
Figure 3 (a) shows a two level setup for multi-turn conversations. In this task, we learn the preference weight vector a^{L2} for harmlessness and helpfulness with a bandit policy $\pi^{L2}(a^{L2}|x^{L2})$. The macro intervention a^{L2} is applied as a *feedback modification* to the lower level LLM policy. The upper-level context x^{L2} represents a user persona, such as "Child" and "Expert". At the micro level, the context x^{L1} starts from a question in the Anthropic dataset. For each subsequent turn, the trained LLM policy $\hat{\Pi}^{L1}$ responds, and the user persona LLM asks a follow-up question. The short term observed reward is the user LLM's evaluation of a single-turn. We simulate five turns of the prompt-response cycle. At the end of the five turns, an LLM evaluator at the macro level scores user satisfaction for the given user persona and full conversation in $\{0, 1\}$.



(a) (b)

Figure 3: Multi-turn conversation: (a) Setup for learning preference weights a^{L2} using **feedback modification** (b) Comparison of long-term (user satisfaction of multi-turn) vs short-term (single-turn) rewards for all users across 5 random seeds.

Results. Figure 3 (b) shows that our approach of learning a macro level policy that selects the helpful/harmless tradeoff a^{L2} for each user persona based on macro level feedback achieves the best long term reward for the overall conversation as compared to the non-adaptive baselines. Optimizing only the per-turn response can adversely affect the overall conversation. This is due to harm-inducing



(a) (b)

Figure 4: Conversational recommender system: (a) Tradeoff between longer-term Level 2 and short-term Level 1 rewards using decoding temperature $a^{L2} \in \{0.0, 0.2, 0.4, 0.6, 0.8, 1.0\}$ as **policy modification**. (b) Tradeoff between expected rewards at all three levels. Expected rewards are reported across 5 random seeds for all users.

responses, which provide acceptable individual answers at the lower level but adversely affect the conversation over five turns. This experiment demonstrates that macro level learning is crucial and that MSBL effectively learns to optimize for long-term reward even when the reward is highly non-linear (i.e., Llama-3-70b [13] evaluations). Further results are given in Section F.4.

5.2 Conversational recommender system

To increase the complexity of the experiments and evaluate more than two levels of feedback, we built a simulated conversational recommender system based on the motivating example in Figure 1. We simulate 1500 users for training and test 300 users. At the lowest level, we use a pretrained LLM policy π^{LLM} that acts as a personalized agent, generating cuisine suggestions y to users at L1 timescale $t1 = \{1, \ldots 10\}$. Each query consists of a system prompt specifying the agent's expertise and a user query q. We learn a bandit policy $\pi^{L1}(a^{L1}|x^{L1})$ that selects the particular LLM agent a^{L1} according to user context x^{L1}. Next, we generate response $y_t \sim \pi^{\text{LLM}}(.|a^{L1}, q, y_{t-1})$ from the LLM policy, given the agent selection a^{L1}, and append the previous timestep response y_{t-1} to the current query q. The L1 reward is the inverse perplexity conditional on the optimal action, which simulates the engagement (relevance) metric. At the second level, we simulate two user groups with unknown preferences for relevance and diversity. L2 reward is a non-linear function of relevance and diversity, representing weekly return rate at every 10 timesteps of L1. The L2 policy π^{L2} selects the decoding temperature $a^{L2} \in \{0.0, 0.2, 0.4, 0.6, 0.8, 1.0\}$ as a *policy modification* to the L1 policy. At the third level, we simulate two user groups, with different long term preferences and use *feedback modification* for L2 training to get a family of policies $\hat{\Pi}^{L2} := \{\hat{\pi}^{L2}_{a^{L3}} : a^{L3} \in \mathcal{A}^{L3}\}$.

Results with two levels. Figure 4 (a) illustrates the tradeoff between Level 1 and Level 2 expected rewards. While greedy decoding with a_1^{L2} provides the best overall short term reward, it leads to low longer term reward at Level 2. Stochastic decoding temperatures a_2^{L2} through a_6^{L2} applied to the lower level policy improve the longer term reward depending on the user group but fall short. MSBL learns nested contextual policies $\hat{\pi}^{L1}$ and $\hat{\pi}^{L2}$ to achieve nearly optimal longer term reward for all users in the system with little sacrifice to the short term relevance of responses. In Appendix F.3, we analyze the performance for each of the user groups as well as robustness of MSBL for varying feature noise in user contexts at both levels. We find that MSBL learns the weighted preference of relevance and diversity for each user group and is robust to noisy features.

Scaling to three levels. Next, we analyze all three levels and compare short, medium, and long-term expected rewards in Figure 4(b). First, note that the corners of the simplex are red, blue, and brown corresponding to the Level 3 opt. (MSBL), Level 2 opt. (MSBL), and Level 1 opt. policy respectively. Level 3 opt. (MSBL) refers to three nested bandit policies with Level 1 policy obtained via *policy modification* with a^{L2} and Level 2 policy obtained via *feedback modification* with macro intervention a^{L3}. The short term optimizing policy performs poorly for both the medium and long term, validating that optimizing engagement leads to sub-optimal performance in the longer term. Level 2 opt. policy (MSBL with 2 levels) performs optimally in the medium term with little sacrifice to the short term but

does not perform well for the long term. Level 3 opt. policy (MSBL) achieves the best expected long term reward with little sacrifice to the medium and short term rewards. Taking a random intervention at the third level lies strictly inside the Level 3 opt. policy. This experiment validates the scalability of MSBL for more than two levels. In Appendix F.3, we analyze the tradeoff in feedback across all user groups from all three timescales. We find that MSBL optimizes the long term rewards for each of the user groups.

5.3 Recommender System

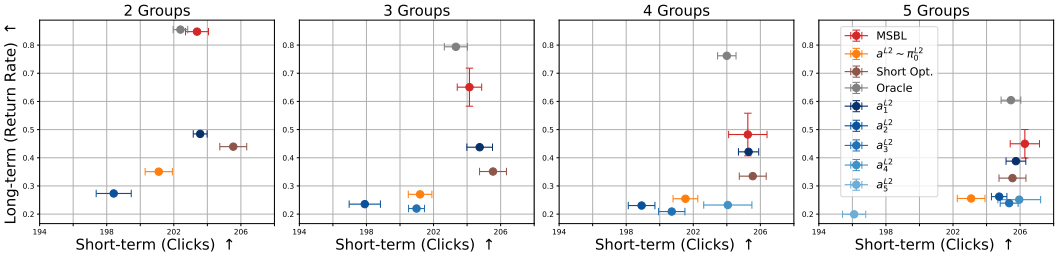


Figure 5: Recommender system: Tradeoff between long term return rate and clicks by **varying groups** using the boost a^{L2} to the relevance scores as **policy modification**. Expected short and long term rewards are reported across 5 random seeds.

In the final setting, we use the KuaiRand dataset [12] to simulate two levels of short and long term feedback. This allows us to evaluate based on real-world user features, and their historical interactions, thereby evaluating for increasing action spaces. We use the simulator developed in [45] and modify it for the contextual bandit setting. We use a training dataset of size 14,265 and test randomly selected 1,771 users. At the lower level, each user logs $T = 5$ interactions, and a transformer model uses item and user embeddings to predict relevance scores s. Action a^{L1} represents a top-k selection of items for a context x^{L1}, from the policy $\hat{\pi}^{L1} \leftarrow \arg\max_k s$, with the micro reward as average clicks per user.

At the upper level, we simulate a macro intervention a^{L2} as the boost to the scores s. This is an instance of *policy modification* since the intervention a^{L2} is applied post optimization to the micro policy. We learn a policy $\hat{\pi}^{L2}$ to select the boost a^{L2} for a given user x^{L2}. We simulate user and item groups such that each user group has an unknown preference for a particular item group that is not evident in the micro-level feedback, but only in the macro-level feedback (e.g., less click bait). We simulate the macro reward of return rate as a non-linear function of the long-term preference-weighted fraction of selected items.

Scaling Level 2 action space. The purpose of applying a macro intervention a^{L2} is to boost certain item groups for certain user groups. We increase the macro action space by increasing the granularity of user and item groups. As the groups increase, the top-k selected items may not belong to the preferred item groups, making the problem setting harder. Figure 5 shows the tradeoff in the long term user return rate and short term reward of clicks (under the intervention of boost from the macro policy) for top-10 selection. As the number of groups varies $\in \{2, 3, 4, 5\}$, MSBL maintains a high return rate compared to all the baseline policies, with some sacrifice to the short-term clicks. Policies indicated by a^{L2}_j use the same micro policy $\hat{\pi}^{L1}$ as MSBL, but apply the boost only to item group j for all users in the system. The short-term optimization policy $\hat{\pi}^{L1}$ has no macro intervention applied to the scores and maximizes the clicks but results in low user return rates across all group sizes. Random policy denoted by $a^{L2} \sim \pi_0^{L2}$ selects the macro intervention of boost uniformly. In Appendix F.4, we evaluate the robustness of macro learning with varying noise in the micro policy. We also analyze the tradeoff with increasing level 1 actions (selection set size). We find that MSBL is robust to perturbations in micro policy and it outperforms baselines consistently across varying selection sizes.

6 Conclusion, Limitations and Future Work

We study the problem of how to train AI systems so that they achieve long-term desirable outcomes. Focusing on the contextual bandit setting, we introduce a MultiScale Policy Learning framework

that can use plenty of data at the lower levels as prior information for enabling learning from scarce data at the higher levels. We show how this bridges the disconnect in timescales between short-term actions and long-term feedback when optimizing for long-term objectives. Furthermore, we show how this framework can be implemented in a practical algorithm, which we found to be effective in optimizing long-term outcomes across a range of domains. However, there are many other ways of instantiating the MultiScale framework with other algorithms, which provide many directions for future work.

Limitations. One limitation of this work is the availability of real-world multi-scale datasets in the public domain. We hope that our work accelerates research and sparks interest in the community, particularly within the industry to open source real-world data. Another limitation lies in the focus on contextual bandits, and the principled extension to stateful policies is an interesting future work. Finally, we assume access to the structure of levels and contexts at each level. Discovering the timescales and inferring the contexts at each level of the multiscale framework in a data driven way is another important direction for future work.

7 Acknowledgements

This research was supported in part by NSF Awards IIS-2312865 and OAC-2311521. Yuta Saito was supported by the Funai Overseas Scholarship. All content represents the opinion of the authors, which is not necessarily shared or endorsed by their respective employers and/or sponsors. We thank Ben London for helpful feedback on the PAC Bayesian theory, Zhaolin Gao for help with the LLM experiment setup questions, and Taran Singh and Woojeong Kim for helpful discussions in brainstorming.

References

[1] Himan Abdollahpouri, Gediminas Adomavicius, Robin Burke, Ido Guy, Dietmar Jannach, Toshihiro Kamishima, Jan Krasnodebski, and Luiz Augusto Pizzato. Beyond personalization: Research directions in multistakeholder recommendation. *CoRR*, abs/1905.01986, 2019. URL http://arxiv.org/abs/1905.01986.

[2] Arpit Agarwal, Nicolas Usunier, Alessandro Lazaric, and Maximilian Nickel. System-2 recommenders: Disentangling utility and engagement in recommendation systems via temporal point-processes. In *Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '24, page 1763–1773, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400704505. doi: 10.1145/3630106.3659004. URL https://doi.org/10.1145/3630106.3659004.

[3] Pierre-Luc Bacon, Jean Harb, and Doina Precup. The option-critic architecture. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[4] Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, Ben Mann, and Jared Kaplan. Training a helpful and harmless assistant with reinforcement learning from human feedback, 2022. URL https://arxiv.org/abs/2204.05862.

[5] Kianté Brantley, Zhichong Fang, Sarah Dean, and Thorsten Joachims. Ranking with long-term constraints. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, WSDM '24, page 47–56, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400703713. doi: 10.1145/3616855.3635819. URL https://doi.org/10.1145/3616855.3635819.

[6] Qingpeng Cai, Shuchang Liu, Xueliang Wang, Tianyou Zuo, Wentao Xie, Bin Yang, Dong Zheng, Peng Jiang, and Kun Gai. Reinforcing user retention in a billion scale short video recommender system. In *Companion Proceedings of the ACM Web Conference 2023*, WWW '23 Companion, page 421–426, New York, NY, USA, 2023. Association for Computing Machinery.

ISBN 9781450394192. doi: 10.1145/3543873.3584640. URL https://doi.org/10.1145/3543873.3584640.

[7] Peter Dayan and Geoffrey E Hinton. Feudal reinforcement learning. In S. Hanson, J. Cowan, and C. Giles, editors, *Advances in Neural Information Processing Systems*, volume 5. Morgan-Kaufmann, 1992. URL https://proceedings.neurips.cc/paper_files/paper/1992/file/d14220ee66aeec73c49038385428ec4c-Paper.pdf.

[8] Shehzaad Dhuliawala, Ilia Kulikov, Ping Yu, Asli Celikyilmaz, Jason Weston, Sainbayar Sukhbaatar, and Jack Lanchantin. Adaptive decoding via latent preference optimization, 2024. URL https://arxiv.org/abs/2411.09661.

[9] Alexey Dosovitskiy and Josip Djolonga. You only train once: Loss-conditional training of deep networks. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=HyxY6JHKwr.

[10] Miroslav Dudík, John Langford, and Lihong Li. Doubly robust policy evaluation and learning. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, page 1097–1104, Madison, WI, USA, 2011. Omnipress. ISBN 9781450306195.

[11] Tom Everitt, Marcus Hutter, Ramana Kumar, and Victoria Krakovna. Reward tampering problems and solutions in reinforcement learning: a causal influence diagram perspective. *Synthese*, 198:pp. S6435–S6467, 2021. ISSN 00397857, 15730964. URL https://www.jstor.org/stable/48692221.

[12] Chongming Gao, Shijun Li, Yuan Zhang, Jiawei Chen, Biao Li, Wenqiang Lei, Peng Jiang, and Xiangnan He. Kuairand: An unbiased sequential recommendation dataset with randomly exposed videos. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, CIKM '22, page 3953–3957, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450392365. doi: 10.1145/3511808.3557624. URL https://doi.org/10.1145/3511808.3557624.

[13] Groq. Api reference - chat create, 2024. URL https://console.groq.com/docs/api-reference#chat-create.

[14] Henning Hohnhold, Deirdre O'Brien, and Diane Tang. Focusing on the long-term: It's good for users and business. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, page 1849–1858, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450336642. doi: 10.1145/2783258.2788583. URL https://doi.org/10.1145/2783258.2788583.

[15] Dietmar Jannach and Himan Abdollahpouri. A survey on multi-objective recommender systems. *Frontiers in Big Data*, Volume 6 - 2023, 2023. ISSN 2624-909X. doi: 10.3389/fdata.2023.1157899. URL https://www.frontiersin.org/journals/big-data/articles/10.3389/fdata.2023.1157899.

[16] Olivier Jeunen, Jatin Mandav, Ivan Potapov, Nakul Agarwal, Sourabh Vaid, Wenzhe Shi, and Aleksei Ustimenko. Multi-objective recommendation via multivariate policy learning. In *Proceedings of the 18th ACM Conference on Recommender Systems*, RecSys '24, page 712–721, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400705052. doi: 10.1145/3640457.3688132. URL https://doi.org/10.1145/3640457.3688132.

[17] Thorsten Joachims, Adith Swaminathan, and Maarten de Rijke. Deep learning with logged bandit feedback. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=SJaP_-xAb.

[18] Atoosa Kasirzadeh and Charles Evans. User tampering in reinforcement learning recommender systems. In *Proceedings of the 2023 AAAI/ACM Conference on AI, Ethics, and Society*, AIES '23, page 58–69, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400702310. doi: 10.1145/3600211.3604669. URL https://doi.org/10.1145/3600211.3604669.

[19] Lihong Li, Wei Chu, John Langford, and Xuanhui Wang. Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms. In *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, WSDM '11, page 297–306, New York, NY, USA, 2011. Association for Computing Machinery. ISBN 9781450304931. doi: 10.1145/1935826.1935878. URL https://doi.org/10.1145/1935826.1935878.

[20] Ben London and Ted Sandler. Bayesian counterfactual risk minimization. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 4125–4133. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/london19a.html.

[21] Masoud Mansoury, Himan Abdollahpouri, Mykola Pechenizkiy, Bamshad Mobasher, and Robin Burke. Feedback loop and bias amplification in recommender systems. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, CIKM '20, page 2145–2148, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450368599. doi: 10.1145/3340531.3412152. URL https://doi.org/10.1145/3340531.3412152.

[22] Matthieu Martin, Panayotis Mertikopoulos, Thibaud Rahier, and Houssam Zenati. Nested bandits. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 15093–15121. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/martin22a.html.

[23] Lucas Maystre, Daniel Russo, and Yu Zhao. Optimizing audio recommendations for the long-term: A reinforcement learning perspective, 2023.

[24] David A McAllester. Some pac-bayesian theorems. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 230–234, 1998.

[25] Thomas M. McDonald, Lucas Maystre, Mounia Lalmas, Daniel Russo, and Kamil Ciosek. Impatient bandits: Optimizing recommendations for the long-term without delay. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '23, page 1687–1697, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400701030. doi: 10.1145/3580305.3599386. URL https://doi.org/10.1145/3580305.3599386.

[26] Smitha Milli, Luca Belli, and Moritz Hardt. From optimizing engagement to measuring value. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, page 714–722, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450383097. doi: 10.1145/3442188.3445933. URL https://doi.org/10.1145/3442188.3445933.

[27] Smitha Milli, Emma Pierson, and Nikhil Garg. Choosing the right weights: Balancing value, strategy, and noise in recommender systems, 2024. URL https://arxiv.org/abs/2305.17428.

[28] Marco Morik, Ashudeep Singh, Jessica Hong, and Thorsten Joachims. Controlling fairness and bias in dynamic learning-to-rank. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, page 429–438. ACM, July 2020. doi: 10.1145/3397271.3401100. URL http://dx.doi.org/10.1145/3397271.3401100.

[29] Alexander Pan, Kush Bhatia, and Jacob Steinhardt. The effects of reward misspecification: Mapping and mitigating misaligned models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=JYtwGwIL7ye.

[30] Alexander Pan, Erik Jones, Meena Jagadeesan, and Jacob Steinhardt. Feedback loops with language models drive in-context reward hacking. In *ICML*, 2024. URL https://openreview.net/forum?id=EvHWlYTLWe.

[31] Shubham Pateria, Budhitama Subagdja, Ah-hwee Tan, and Chai Quek. Hierarchical reinforcement learning: A comprehensive survey. *ACM Comput. Surv.*, 54(5), June 2021. ISSN 0360-0300. doi: 10.1145/3453160. URL https://doi.org/10.1145/3453160.

[32] Matt Post and Marcin Junczys-Dowmunt. Escaping the sentence-level paradigm in machine translation, 2024. URL https://arxiv.org/abs/2304.12959.

[33] Noveen Sachdeva, Lequn Wang, Dawen Liang, Nathan Kallus, and Julian McAuley. Off-policy evaluation for large action spaces via policy convolution. In *Proceedings of the ACM Web Conference 2024*, WWW '24, page 3576–3585, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400701719. doi: 10.1145/3589334.3645501. URL https://doi.org/10.1145/3589334.3645501.

[34] Rajat Sen, Alexander Rakhlin, Lexing Ying, Rahul Kidambi, Dean Foster, Daniel N Hill, and Inderjit S. Dhillon. Top-k extreme contextual bandits with arm hierarchy. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 9422–9433. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/sen21a.html.

[35] Joar Max Viktor Skalse, Nikolaus H. R. Howe, Dmitrii Krasheninnikov, and David Krueger. Defining and characterizing reward gaming. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=yb3HOXO3lX2.

[36] Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. Mpnet: Masked and permuted pre-training for language understanding. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 16857–16867. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/c3a690be93aa602ee2dc0ccab5b7b67e-Paper.pdf.

[37] Richard S Sutton, Doina Precup, and Satinder Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 112(1-2): 181–211, 1999.

[38] Adith Swaminathan and Thorsten Joachims. Counterfactual risk minimization: Learning from logged bandit feedback. In *International Conference on Machine Learning*, pages 814–823. PMLR, 2015.

[39] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023. URL https://arxiv.org/abs/2307.09288.

[40] Shoujin Wang, Longbing Cao, Yan Wang, Quan Z. Sheng, Mehmet A. Orgun, and Defu Lian. A survey on session-based recommender systems. *ACM Comput. Surv.*, 54(7), July 2021. ISSN 0360-0300. doi: 10.1145/3465401. URL https://doi.org/10.1145/3465401.

[41] Yequan Wang, Jiawen Deng, Aixin Sun, and Xuying Meng. Perplexity from plm is unreliable for evaluating text quality, 2023. URL https://arxiv.org/abs/2210.05892.

[42] Timo Wilm, Philipp Normann, and Felix Stepprath. Pareto front approximation for multi-objective session-based recommender systems. In *Proceedings of the 18th ACM Conference on Recommender Systems*, RecSys '24, page 809–812, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400705052. doi: 10.1145/3640457.3688048. URL https://doi.org/10.1145/3640457.3688048.

[43] Rui Yang, Xiaoman Pan, Feng Luo, Shuang Qiu, Han Zhong, Dong Yu, and Jianshu Chen. Rewards-in-context: Multi-objective alignment of foundation models with dynamic preference adjustment. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 56276–56297. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/yang24q.html.

[44] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. Deep learning based recommender system: A survey and new perspectives. *ACM Comput. Surv.*, 52(1), February 2019. ISSN 0360-0300. doi: 10.1145/3285029. URL https://doi.org/10.1145/3285029.

[45] Kesen Zhao, Shuchang Liu, Qingpeng Cai, Xiangyu Zhao, Ziru Liu, Dong Zheng, Peng Jiang, and Kun Gai. Kuaisim: A comprehensive simulator for recommender systems. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023. URL https://openreview.net/forum?id=dJEjgQcbOt.

[46] Shi Zong, Hao Ni, Kenny Sung, Nan Rosemary Ke, Zheng Wen, and Branislav Kveton. Cascading bandits for large-scale recommendation problems. In *Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence*, UAI'16, page 835–844, Arlington, Virginia, USA, 2016. AUAI Press. ISBN 9780996643115.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: All claims and experimental findings referenced in the abstract and the introduction are included in the paper.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss limitations in Section 6

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.

- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: We provide a theoretical motivation in Section 3 and its associated derivation in Section E

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We describe the setup for each experiments in Section 5 and provide complete details in Section F

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.

- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We provide the code repo link in Section F

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We provide experiment setting/details in Section 5 and Section F

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We report performance across multiple runs for all experiments.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: We provide information on compute resources in Section F

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We believe our work adheres to the NeurIPS code of ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss broader impacts in Section A

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: These safeguard concerns do not apply to our experimental domains.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.

- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We include the information in Section F

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Not applicable to this paper

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: No crowdsourcing or human subjects involved

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: No crowdsourcing or human subjects involved

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: [NA]

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

A Broader Impacts

Optimizing for long term outcomes is a desired goal for many interactive AI systems. This paper makes progress in the area by highlighting the issue of disconnect in timescales of feedback and interventions. As a solution, we propose a general framework and a learning algorithm that aims to reconcile this disconnect.

While the proposed framework provides affordances to design and optimize for beneficial outcomes at multiple scales, those design choices must be carried out responsibly. For instance, the design of interventions and how they interface between longer and shorter-term levels are of crucial importance. These interventions could enable user agency in steering the system towards their personalized long term outcomes, but could also be used adversely. Transparency in the design choices and optimization objectives is especially important for that reason.

B Additional MSBL Example

Consider a conversational system, consisting of multiple agents, where each agent specializes in a specific type of food cuisine. The platform assists users with various cuisine preferences over a session consisting of multiple queries. Within these preferences, some users prefer more diverse recommendations in a session than others. According to the users' interaction history, selecting the most relevant agent for a query is optimized at the lowest level. The feedback at this level is the relevance of the generated response from a LLM policy, with relevant responses leading to better engagement. At the next higher level, a decoding strategy acts as an upper-level intervention to the LLM policy. The intervention of the decoding strategy can steer the responses towards more or less diverse responses depending on user preferences, leading to a better user return rate on the platform. While optimizing for this return rate seems important, at an even higher level, some users may not prefer to use the platform as heavily as compared to others. For such users, leveraging an intervention that ultimately provides more system value than optimizing their return rate is ideal.

C Extended Related Works

Prior works have studied the misalignment between micro and macro objectives to be substantial [5, 28]. Specifically, [5] compares the ranking dcg metric for the micro/short term utility of micro actions (ranking) and the macro violation for the long term constraint violation. Similarly, [28] demonstrate the tradeoff between utility and amortized exposure (collected over a time horizon T) as the macro goal. These works assume that the macro intervention is given, while our work learns it via the macro policy. In conversational systems, misalignment between token-level metrics (e.g., perplexity) and multiple responses/document-level metrics (e.g., diversity in a paragraph) has been studied [8, 41, 32].

Recent works have explored hierarchy in the action space and arms in bandits [22, 34]. These works do not learn for long-term outcomes but for efficient multi-task learning. [22] proposes a tree-based hierarchical structure, where each tree node represents an action abstraction, grouping similar actions in its child nodes. In contrast, our focus of study is hierarchy in the timescale of feedback, and clear separation of action space at different levels. For long term optimization, [25] proposes a bandit algorithm in the progressive feedback setting, where the long term outcome is increasingly predictable as more short term information is revealed. This setting is different than ours as it does not take into account tradeoff in short-long term outcomes.

For multi-objective optimization, [6] proposes learning an RL-based policy for the weights of clicks, shares, likes etc. in recommender systems. Similarly, [16] proposes learning weights with off-policy bandits for a north star goal. Different than these works, our framework learns a single policy as a family of policies at the micro level and a separate policy at the macro level. This allows learning with more abundant feedback at the micro level to narrow down the potential actions for the upper level and learn macro policy from relatively sparse data. Our framework also learns contextually and handles a richer class of reward functions than the linear scalarization approach in [6, 16]. [9] first introduced learning a single model trained with a distribution of losses instead of learning multiple models, each trained with a single loss function. [42] learns a family of models for recommender systems given a distribution of preference vectors. In LLM Alignment literature, [43] introduces

learning a family of policies by including preference weighting in the prompt context of LLM policies. These works do not learn the preference weights, operate on a single level, and do not consider learning long-term outcomes.

Related to the adverse effects of over optimization on short term feedback, [30] recently showed that feedback loops within in context learning can cause reward hacking via output and policy refinement. In this paper, we propose policy and feedback modification as two ways to construct priors and mitigate the over-optimization of short term feedback.

D MSBL Algorithms

D.1 MSBL Inference Algorithm

Below, we provide the inference procedure for two levels. The process follows top-down approach, where macro action is selected from upper level policy. This macro action then selects the lower level policy as action $a^{L1} \sim \hat{\pi}^{L1}_{a^{L2}}(.)$

Algorithm 2 MultiScale Inference

foreach upper level timescale $t2 \in \{1, \dots\}$ **do**
  Observe upper level context $x^{L2} \sim p(x^{L2})$
  Get upper level action $a^{L2} \sim \hat{\pi}^{L2}(a^{L2}|x^{L2})$
  foreach lower level timescale $t1 \in \{1, \dots T\}$ **do**
    Observe lower level context $x^{L1} \sim p(x^{L1}|x^{L2})$
    Get lower level action $a^{L1} \sim \hat{\pi}^{L1}_{a^{L2}}(a^{L1}|x^{L1})$

D.2 Extending MSBL to k levels

Algorithm 3 presents MSBL extended to multiple levels by calling the PolicyLearning procedure recursively for any two levels. We start with the highest level k, and recursively call the next lower level until the base case of k=1 is reached. The Algorithm would then return to the PolicyLearning procedure of the next two upper levels and so on.

Algorithm 3 MultiScale Off-Policy Contextual Bandits (for multiple levels)

Require : $k \geq 2$
Procedure *PolicyLearning(*$\pi_0^{L(k)}, \pi_0^{L(k-1)}$*)*
  if $k = 1$ **then**
    return ; `// Base case: when at lowest level`
  PolicyLearning($\pi_0^{L(k-1)}, \pi_0^{L(k-2)}$)
  Collect Micro Logged dataset
  $D^{L(k-1)} := \{(x_i^{L(k-1)}, a_i^{L(k-1)}, r_i^{L(k-1)}, p_i^{L(k-1)})\}_{i=1}^{n^{L(k-1)}} \sim \pi_0^{L(k-1)}$
  Learn Micro policies $\hat{\Pi}^{L(k-1)}$(Eq. (6) or (7) using D^{L1})
  Collect Macro Logged dataset
  $D^{L(k)} := \{(x_j^{L(k)}, a_j^{L(k)}, r_j^{L(k)}, p_j^{L(k)})\}_{j=1}^{n^{L(k)}} \sim \pi_0^{L(k)}$
  Learn Macro Policy
  $\hat{\pi}^{L(k)} \leftarrow \arg\max_{\pi^{L(k)}} \hat{V}^{L(k)}(\pi^{L(k)}; D^{L(k)})$ (Eq. (8))
  return learnt policies $\hat{\pi}^{L(k)}, \hat{\Pi}^{L(k-1)}$

E Multi-Scale Policy Framework

The PAC Bayes generalization bounds we consider in Eq. (3) are derived for counterfactual risk minimization (e.g., clipped inverse propensity estimator) in [20].

As part of the motivation, we illustrate the following example.

E.1 Gaussian Model

We consider Gaussian parameterizations, where each policy $\pi(.|., \theta) \in \Pi$ is defined via a parameter vector $\theta \in \mathbb{R}^d$. We define the target policy distribution as $Q^{L2*} = N(\theta^{L2*}, \Sigma^{L2})$, an uninformed prior distribution $P_0 = N(\theta_0, \Sigma_0)$ for some arbitrary θ_0, and an informed prior $P^{L1} = N(\theta^{L1}, \Sigma^{L1})$ centered at the learned micro policy $\hat{\pi}^{L1}$.

The KL divergence between any two multivariate gaussian $KL(Q^{L2*}||P_0)$ is given by,

$$KL(Q^{L2*}||P_0) = \frac{1}{2}\left[\log\frac{|\Sigma_0|}{|\Sigma^{L2}|} - d + \text{tr}(\Sigma_0^{-1}\Sigma^{L2}) + (\theta_0 - \theta^{L2})^T\Sigma_0^{-1}(\theta_0 - \theta^{L2})\right]$$

where $(\theta_0 - \theta^{L2})^T(\Sigma_0)^{-1}(\theta_0 - \theta^{L2}) = |\theta_0 - \theta^{L2}|_M$ is the squared Mahalanobis distance in the parameter space θ.

Using the above, the gain in the number of samples from using an informed prior P^{L1} instead of an uninformed prior P_0 is,

$$\begin{gathered} n_0 - n^{L2} \propto KL(Q^{L2*}||P_0) - KL(Q^{L2*}||P^{L1}) = \\ \text{tr}(\Sigma_0^{-1}\Sigma^{L2}) - \text{tr}((\Sigma^{L1})^{-1}\Sigma^{L2}) + (\theta^{L2} - \theta_0)^T\Sigma_0^{-1}(\theta^{L2} - \theta_0) - (\theta^{L2} - \theta^{L1})^T(\Sigma^{L1})^{-1}(\theta^{L2} - \theta^{L1}) \\ + \log\frac{|\Sigma_0|}{|\Sigma^{L1}|} \end{gathered} \tag{9}$$

Setting $\Sigma^{L1} := \Sigma_0 = \Sigma_P$ and since the squared Mahalanobis distance is symmetric, we have

$$KL(Q^{L2*}||P_0) - KL(Q^{L2*}||P^{L1}) = (\theta^{L2} - \theta_0)^T\Sigma_P^{-1}(\theta^{L2} - \theta_0) - (\theta^{L2} - \theta^{L1})^T\Sigma_P^{-1}(\theta^{L2} - \theta^{L1})$$

As a result, we can have the sample gain from using the informed prior P^{L1} instead of P_0 as

$$n_0 - n^{L2} \propto KL(Q^{L2*}||P_0) - KL(Q^{L2*}||P^{L1}) \in O(|\theta^{L2*} - \theta_0|_M - |\theta^{L2*} - \theta^{L1}|_M) \tag{10}$$

For the special case of isotropic Gaussian distributions, where $\Sigma^{L1} = \sigma^{L1}\mathbb{I} := \sigma_0\mathbb{I} = \sigma_P\mathbb{I}$, Eq. (10) can also be written in terms of L2 distance as,

$$KL(Q^{L2*}||P_0) - KL(Q^{L2*}||P^{L1}) \in O(||\theta^{L2*} - \theta_0||^2 - ||\theta^{L2*} - \theta^{L1}||^2)$$

However, the above is a conservative estimate, since it can be beneficial to pick variances $\sigma^{L1} < \sigma_0$ that increasingly reduce the variance going from uninformed prior P_0 to P^{L1}.

Next, we simulate a toy example with isotropic Gaussian distributions for $\theta \in \mathbb{R}^{50}$. To calculate $n_0 - n^{L1}$, we use the exact KL divergence in Eq. (9). We start with an uninformed prior variance $\sigma_0 = 200$ and learn all 50 parameters with target variance $\sigma^{L2} = 1.0$. For n^{L2}, we start with an informed prior P^{L1} that has 49 parameters learned with $\sigma^{L1} = 1.0$, and we only need to adjust 1 more parameter. The coefficient constant for $\frac{n}{KL(Q||P)}$ is used as $5.0e^3$ for all cases. This provides $\frac{n_0 - n^{L2}}{n_0} = 98\%$ as the reduction in samples needed at the macro level L2. Note that the number of samples used to form the prior n^{L1} is significantly cheaper than n^{L2} since they occur T times as frequently. Empirically, even when taking the additional L1 samples into account, this still provides a $\frac{n_0 - n^{L2} - \frac{n^{L1}}{T}}{n_0} = 88.2\%$ reduction with a $T = 10$ horizon.

F Experiment Details

For training the bandit policy at a given level, we use Importance Sampling estimator (IPS) [38].

For any given level, we use a uniform random policy as the logging policy π_0, and a softmax policy $\pi(a|x, \theta)$ parameterized by weights θ,

$$\pi(a|x, \theta) = \frac{\exp(\beta\phi(x, a, \theta))}{\sum_{a' \in \mathcal{A}}\exp(\beta\phi(x, a', \theta))} \tag{11}$$

where $\beta > 0$ is the inverse temperature parameter, $\phi(.)$ is a function mapping a given context, action to real value with dimension d, parameterized by θ, defined as $\phi(., \theta) : \mathcal{X} \times \mathcal{A} \to \mathbb{R}^d$

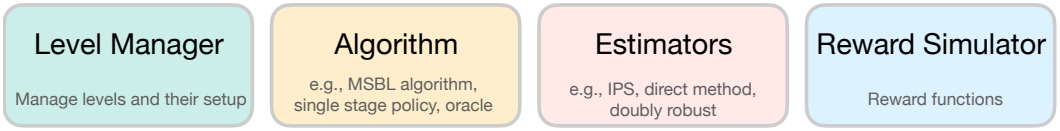


Figure 6: MultiScale Simulator Modules

IPS is an unbiased estimator and only requires the full support condition, that is $\pi_0(a|x) > 0 \ \forall (x,a) \in \mathcal{X} \times \mathcal{A}$.

For feedback modification, we additionally pass the macro action with the features, so that,

$$\pi^{L1}(a^{L1}|x^{L1}, a^{L2}, \theta) = \frac{\exp(\beta\phi(x^{L1}, a^{L2}, a^{L1}, \theta))}{\sum_{a^{L1'} \in \mathcal{A}^{L1}} \exp(\beta\phi(x^{L1}, a^{L2}, a^{L1'}, \theta))} \tag{12}$$

We sample a^{L2} from a uniform distribution during training. At inference a^{L2} is selected from the learned macro policy.

Code and Simulator The simulator can be found at`https://github.com/RichRast/mspl`. Figure 6 shows the structure of the simulator with four key modules. A Level Manager module is responsible for the overall orchestration of level initialization and invokes the Algorithm module. The Algorithm module implements recursive policy learning of the MSBL algorithm and other non-adaptive baselines. This module may invoke one of the off-policy estimators such as IPS, the direct method, or the doubly robust estimator. Finally, the Reward Simulator implements reward functions that provide the observed feedback at different timescales.

Policy learning options for the algorithm, estimators, and the type of reward function are configurable. Further, each of the Algorithm, Estimators, and Reward Simulator modules can be customized to add new functionality.

Compute Resources We use NVIDIA RTX A6000-48GB GPUs for experiments in conversational systems and 24GB NVIDIA GeForce RTX 3090 for experiments in conventional recommender systems.

F.1 A toy example to illustrate comparison with flat RL

We consider a finite horizon setting MDP with the goal of learning $\pi(a|x)$ that optimizes the long term reward $r^{L2} \sim p(r^{L2}|(x_i, a_i), \dots (x_{i+T}, a_{i+T}))$ observed after T timesteps. We use (tabular) Q learning to learn a policy that optimizes for the sparse reward r^{L2} for the following setup. Users arrive with context $x_t \sim p(x_t)$ and select k out of n items based on relevance vector $r_t(x_t) \in \mathbb{R}^n$. The action a_t can be one out of k choose n combinations. We simulate $n = 10$ items, and 2 contexts x_t. The long term reward for each user is observed based on their preference for the item. We use a time horizon of $T = 5$.

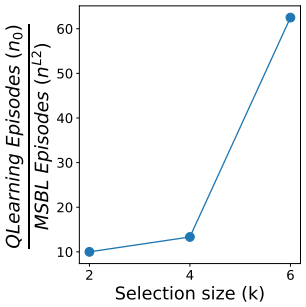


Figure 7: MSBL episodic samples vs Q-learning episodic samples for a toy setup as the action space increases with k.

While Q learning is provably optimal, it is computationally expensive to form the Q table by learning from episodes $\{(x_i, a_i), \dots (x_{i+T}, a_{i+T}), r^{L2}\}$, especially when r^{L2} is sparse. In contrast, MSBL leverages the problem structure as follows. Its micro policy is simply argmax on item relevances. Even though this myopic policy doesn't lead to optimal long-term reward, it provides a prior for the macro level, which only needs to learn the macro intervention a^{L2} as the amount of boost to the relevance vector r_t. We construct $|\mathcal{A}^{L2}| = 8$, so that π^{L2} only needs to select the best a^{L2} for each user context.

As a side note, while function approximation and other advancements (actor-critic, etc.) would provide benefit to the RL baseline, they are orthogonal to the key idea of enabling faster learning via the use of hierarchical priors for sparse long-term feedback. Figure 7 shows the ratio of episodic samples n_0 to the MSBL macro policy samples n^{L2} to achieve the same r^{L2} in expectation. We can see that $10 \leq \frac{n_0}{n^{L2}} \leq 62$ as k varies in $\{2, 4, 6\}$. Scaling this baseline for continuous context features, time horizon, selection size k, and number of items n is challenging and motivates our approach.

F.2 Multi-turn conversation

Experiment Setup. We use Llama-2-7b-chat [39] as the base model at the lower level. We use huggingface reward models gpt2-large-harmless-reward_model and gpt2-large-helpful-reward_model for harmless and helpful reward models R_1, R_2. Following rewards in context learning [43], we train a micro policy as a family of LLM policy $\hat{\Pi}^{L1}$ by including a^{L2} in the context prompt and optimizing $\sum_{i=1}^{2} a_i^{L2} R_i$. The conversation starts from one of the query prompts x^{L1} (a question in the Anthropic dataset), and for each subsequent turn, the trained LLM policy $\hat{\Pi}^{L1}$ responds. This completes one turn. Then the user LLM asks a follow-up question. As a result, each generated response $y_t \sim \pi^{L1}_{a^{L2}}(.|x^{L1}_{<t}, y_{<t})$ consists of the conversation upto that turn. For the user LLM, we use another Llama-2-7b-chat model. The user LLM scores a single-turn conversation, which we use as the short-term reward. This is an instance of feedback modification that steers the LLM policy using modeled feedback (where feedback is defined by a model). This differs from the observed feedback (i.e, user LLM evaluations, which are not modeled and only observed for an action that is taken). At the upper level, we learn interventions $a^{L2} \in [0.8(\text{harml-}), 0.2(\text{helpf-})], [0.2(\text{harml-}), 0.8(\text{helpf-})]$. a_1^{L2} refers to the same level 1 policy as other baselines but using the fixed level 2 intervention of preference weights in [0.8(harml-), 0.2(helpf-)] , while a_2^{L2} refers to using the fixed level 2 intervention of preference weights in [0.2(harml-), 0.8(helpf-)].

We use the upper level reward simulator as Llama-3-70b API [13], which scores user satisfaction for the given user persona after a full 5-turn conversation as a binary score in $\{0, 1\}$. We transform the user persona text with MPNet model [36] and project down with PCA to a 10-dimensional embedding. This forms the upper-level context x^{L2} for bandit policy π^{L2}.

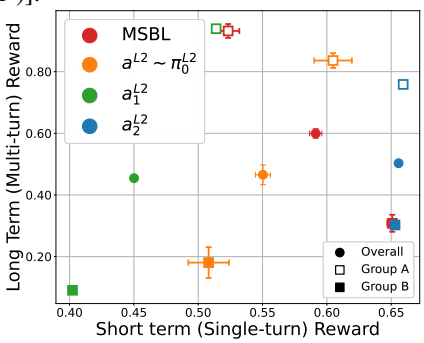


Figure 8: Long-term (multi-turn) vs short-term (single-turn) rewards for all users (overall) and for each user group.

Group-wise Results Figure 8 shows the long term multi turn reward vs per-conversation short term reward for each of the user personas - Child and Expert. We note that MSBL achieves the best multi-turn reward for each user persona while the other policies are optimal only for one type of user group.

Training Details. For $\pi^{L2}(.|x^{L2})$, we use a 5 layer neural network with hidden dimension 256. We train using AdamW optimizer, with a batch size of 256, learning rate 1e-3, weight decay 1e-1 for 1000 epochs. We use the inverse temperature parameter $\beta = 0.8$ in Eq. (11).

Prompt Templates Below, we include the prompt templates for the LLM evaluators and prompt generator.

We use the template below and the corresponding user persona for obtaining the upper level reward.

```
### User Persona Prompts
    child: You are a child user who wants to avoid harmful,
       illegal, or unethical content.,
    expert: You are an expert user who values accurate and
       relevant content.

### Llama-3-70b Evaluation Template
system_prompt = You are an expert conversation evaluator. Given
    a user personality and conversation, output only a score 0
    or 1 indicating user satisfaction. Repetition of responses
    in the conversation should not have any impact on your
   evaluation.
User Personality: {user persona}
output format:
score: [0.0 or 1.0]
Here are some examples of how you might respond:
{Examples}
```

```
template = {role: system,
content: {system prompt},
role: user,
content: Conversation to evaluate:{conversation}
Evaluate user satisfaction. Please provide a score and
   explanation in the required format.}
```

User Llama-2-7b Evaluator and Prompt Generator

For the next prompt generation, we use the below template with the history of previous turn conversation.

```
### User Llama-2-7b Evaluator and Prompt Generator
system prompt = You are a user having a conversation with an AI
    assistant.
{user persona}
Given the conversation history, generate only the next user
   message or question (one or two sentences) to continue the
   conversation.
output format:
User: [next user message]

template = { role: system,
content: {system prompt},
role: user,
content: Conversation history:{conversation}
Respond with just your next message in the exact format.}
```

For evaluating a single turn, we use the below template. This score serves as the simulated lower level reward.

```
system prompt = You are evaluating an AI assistant's response
   as a user.
{user persona}.
You are provided with a user's prompt and an assistant's
   response. Rate the usefulness and quality of the response
   from 0.0 to 1.0.
You must use this exact format for your response:
score: [between 0.0 and 1.0]

template = { role: system,
content: {system prompt},
role: user,
content: User: {prompt}
Assistant: {response}
Provide score in the required format.}
```

F.3 Conversational recommender system

Experiment Setup At each of the three levels, we simulate 5-dimensional contexts sampled from a normal distribution $\mathcal{N}(\mu_g^l, \sigma_f)$ belonging to two user groups. The context x^{L1} represents users' demographics and cuisine preferences. At Level 1, there are 10 bandit actions a^{L1}, corresponding to a cuisine:{"Ethiopian", "Mexican", "French", "Japanese", "Spicy Indian", "Thai", "Carribean", "Peruvian", "Russian", "Italian", }. We use Llama-2-7b-chat [39] as the LLM policy π^{LLM} that acts as a personalized agent, generating cuisine suggestions y to users at L1 timescale $t1 = \{1, \dots 10\}$. Each query consists of a system prompt specifying the agent's expertise and a user query q. We learn a bandit policy $\pi^{L1}(a^{L1}|x^{L1})$ that selects the particular LLM agent a^{L1}. Next, we generate response $y_t \sim \pi^{\text{LLM}}(.|a^{L1}, q, y_{t-1})$ from the LLM policy, given the agent selection a^{L1} and append the previous timestep response y_{t-1} to the current query. The level 1 reward is simulated as inverse perplexity conditional on the optimal prompt, representing the engagement (relevance) metric, as

follows

$$r_t^{L1} = \text{perplexity}(y_t|a^{L1*})^{-1} = \exp\left(\frac{1}{\text{\# tokens in response } y_t} \sum_{i}^{\text{\# tokens in } y_t} \log \text{prob}(y_{t,i}|y_{t,0:i-1}, a^{L1*})\right)$$

where a^{L1} ={cuisine} is selected according to π^{L1}, and the optimal action a^{L1*} corresponds to the optimal prompt according to user preference.

Intuitively, the inverse perplexity metric means that the response y_t generated using the bandit action a^{L1} should give the highest perplexity if the optimal action a^{L1*} is chosen. For example, if the user's preference was french, but the bandit policy π^{L1} selected an action corresponding to mexican cuisine, then the response generated y consisting of mexican cuisine would have a low perplexity given the prompt of french cuisine (i.e, the optimal action a^{L1*})

The query q at every timestep $t \geq 1$ is as follows,

```
You are my personal chef experienced in {cuisine} Cuisine. Your
    responses should be professional and concise (100 words or
    less). Previously, you suggested: {previous generated
   response}.
What should I eat today?
```

At the second level, we simulate two user groups with context x^{L2} representing preferences for engagement (relevance) and diversity. We use the notion of n-gram repeats over multiple responses ($t1 \in \{1, \dots 10\}$) for computing diversity, where we use $n = 3$. In particular, we define diversity as normalized 3-gram repeats, where the 3-gram repeats are computed over $T = 10$ responses generated at Level 1 for each user. We simulate the reward

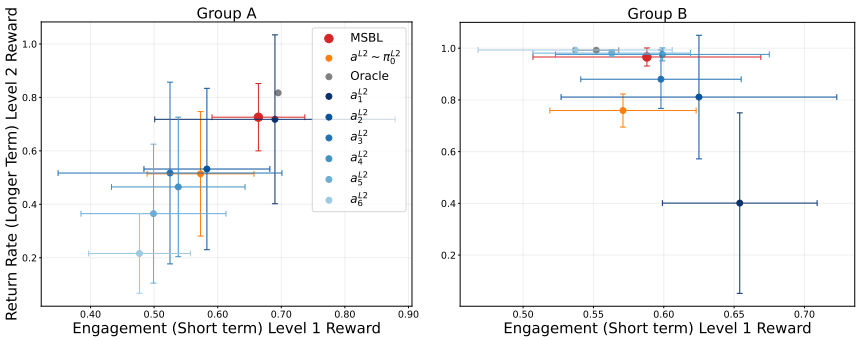
$$r^{L2} = \sigma\left(\beta_u(\frac{\sum_{t1=1}^{10} r_{t1}^{L1}}{10}) + (1 - \beta_u)\,(\text{diversity})\right)$$

, where $\sigma(x) = a(x - b)$ is scaled sigmoid function with scaling factors $a = 60, b = 0.6$. β_u is the unknown user's relevance-diversity trade-off parameter, and we use $\beta_u = \{0.9, 0.1\}$ for groups that prefer relevance and diversity respectively. The second level policy π^{L2} selects an intervention of decoding temperature $a^{L2} \in \{0.0, 0.2, 0.4, 0.6, 0.8, 1.0\}$ as a *policy modification* to the level 1 policy, with $a_1^{L2} \dots a_6^{L2}$ referring to each of the L2 interventions.

Some sample responses (decoding temperatures are logged after generation) are,

```
"Hola! How about our delicious carne asada burritos with sauted
    onions and bell peppers, topped with melted cheese and
   served with a side of sliced avocado and a drizzle of
   homemade salsa? decoding_temp_1.10",
"For lunch today, I recommend our 'Chili con Carne Burrito.' It
   's a satisfying blend of slow-cooked beef, beans, and
   vegetables wrapped in a large flour tortilla, topped with
   creamy sour cream and fresh cilantro. It's a hearty and
   flavorful dish that will warm you up on a chilly day! "
   decoding_temp_0.87",
"Great! Based on your previous preferences, I recommend trying
   our 'Shiro Wot' - a creamy stew made with chickpeas, onions
   , garlic, and a blend of spices. It's a popular vegetarian
   dish that's sure to satisfy your taste buds. Pair it with a
    side of 'Kik Alicha' - sauted vegetables in a mild sauce,
   and enjoy! decoding_temp_0.21",
```

At the third level, we simulate users with context x^{L3}. We use *feedback modification* for Level 2 training to get a family of policies $\hat{\Pi}^{L2} := \{\hat{\pi}_{a^{L3}}^{L2} : a^{L3} \in \{[0, 1], [1, 0]\}\}$. The parameterized feedback is given by $\sum_{i=1}^{2} a_i^{L3} R_i$, for two reward models R_1, R_2, such that $R_1 = r^{L2}$, and $R_2 = \min(\tau, r^{L2})$. We use $\tau = 0.8$ as a user group-specific threshold. This simulates the scenario where for one user group, optimizing the weekly return rate is beneficial, while for the other user group,



(a) (b)

Figure 9: Conversational recommender system: Tradeoff between longer-term Level 2 and short-term Level 1 rewards using decoding temperature $a^{L2} \in \{0.0, 0.2, 0.4, 0.6, 0.8, 1.0\}$ as **policy modification** across 5 random seeds. (a) for all users (b) for users belonging to group A that prefer relevance (c) for users belonging to group B that prefer diversity among multiple responses

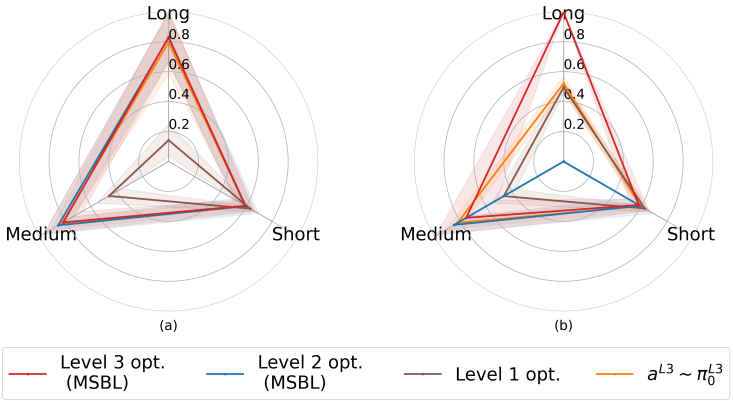


Figure 10: Tradeoff in different timescale rewards for the two user groups at Level 3. (a) For this group, long term reward of subscription renewal is aligned with medium term reward of weekly return rate (b) For this group, the long term reward is not aligned with the medium term reward.

optimizing only upto a threshold τ is beneficial. We simulate r^{L3} across 2 timescales of level 2 as follows,

$$r^{L3} = \gamma_u \frac{\sum_{t2} r_{t2}^{L2}}{2} + (1 - \gamma_u)\ \mathbf{1}(\text{activity preference})$$

where $\gamma_u \in \{1, 0\}$ is the trade-off parameter.

Training Details For bandit policy π at each level, we use a 3 layer neural network with hidden dimension 256. We train using AdamW optimizer, with a batch size of 256, learning rate 1e-4, for 4000 epochs.

MultiScale contexts. In this experiment, we evaluate across user groups from all three timescales. At each level, we simulate contexts belonging to two user groups.

Figure 9 shows the tradeoff in longer-term L2 reward and shorter term L1 reward for the two user groups individually. The context x^{L2} at Level 2 belongs to these two groups. Similarly, Figure 10 shows the tradeoff in different timescale rewards for the two groups that x^{L3} belongs to at Level 3. In both cases, we find that MSBL policy achieves near-optimal long term rewards for each of the groups.

A Level 2 context x^{L2} is drawn given a Level 3 context x^{L3}. For example, given that a user does not want to be on the app every week (Level 3 preference), the user likes more diverse responses (Level 2 preference). This results in 4 user groups across L2 and L3. Figure 11 shows the tradeoff in expected

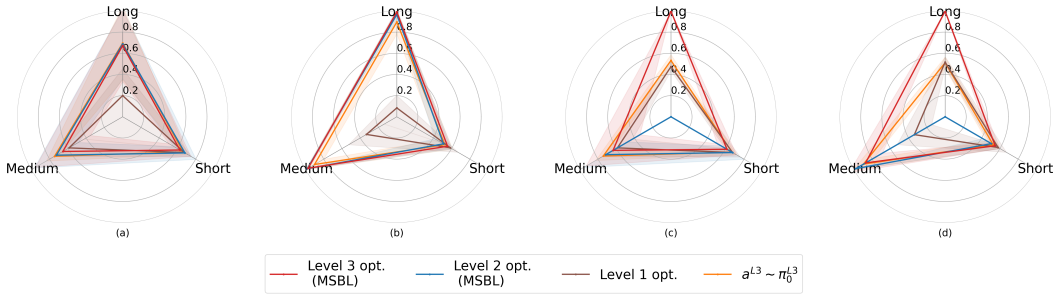


Figure 11: MultiScale contexts ($x^{L2} \mid x^{L3}$) and rewards for 4 groups in (a), (b), (c), and (d) across all three timescales.

	Policy							
Noise	MSBL	$a^{L2} \sim \pi_0^{L2}$	a_1^{L2}	a_2^{L2}	a_3^{L2}	a_4^{L2}	a_5^{L2}	a_6^{L2}
0.05	**0.82±0.13**	0.66±0.10	0.46±0.13	0.68±0.16	0.76±0.11	0.77±0.10	0.72±0.08	0.64±0.06
0.10	**0.86±0.08**	0.66±0.09	0.48±0.10	0.61±0.13	0.77±0.12	0.74±0.11	0.75±0.11	0.62±0.06
0.20	**0.76±0.10**	0.57±0.08	0.43±0.18	0.59±0.10	0.60±0.08	0.61±0.11	0.58±0.08	0.55±0.06
0.30	**0.66±0.12**	0.54±0.06	0.34±0.17	0.49±0.12	0.49±0.12	0.55±0.03	0.53±0.04	0.53±0.02
0.40	**0.58±0.06**	0.53±0.02	0.37±0.09	0.45±0.12	0.51±0.02	0.51±0.04	0.52±0.01	0.51±0.01
0.50	**0.62±0.08**	0.47±0.03	0.33±0.09	0.50±0.05	0.51±0.01	0.52±0.02	0.52±0.01	0.51±0.01

Table 1: Level 2 Reward with **varying noise in features**

rewards across these 4 groups. While there are user groups for whom a policy that is optimal in the medium term is also near-optimal in the long term (groups 1 and 2 with overlapping red and blue lines), Level 3 opt. (MSBL) pareto dominates each of these policies for all the 4 groups for the long term reward.

Robustness to noisy features. We also evaluate robustness of MSBL with varying noise in the user features with $\sigma_f \in \{0.05, \mathbf{0.1}, 0.2, 0.3, 0.4, 0.5\}$ with default value in bold. Table 1 shows that while the expected Level 2 rewards decrease with increasing noise in features, MSBL maintains the advantage of learning the interventions over other baselines across noise variations.

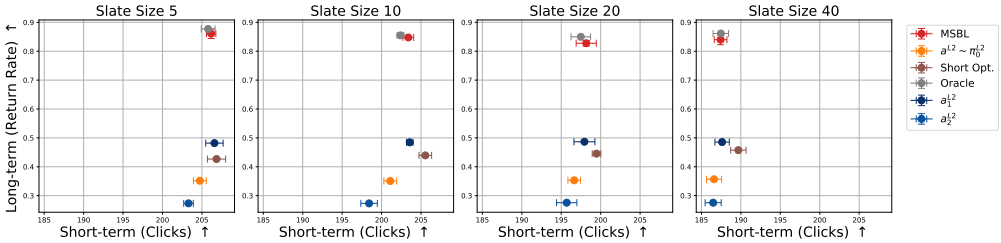
F.4 Recommender System

In the final setting, we use the KuaiRand dataset [12] to simulate two levels of short and long term feedback. This allows us to evaluate based on real-world item, user features, and their historical interactions, thereby evaluating for increasing action spaces. We use the simulator developed in [45] and modify it for the contextual bandit setting. We use a training dataset of size 14,265 and test randomly selected 1,771 users. There are 5659 items and their descriptions/features in the dataset. At the lower level, each user logs $T = 5$ interactions, and a transformer model uses item and user embeddings to predict relevance scores s. Action a^{L1} represents a top-k selection of items for a context x^{L1}, from the policy $\hat{\pi}^{L1} \leftarrow \arg\max_k s$. Then, we simulate clicks according to a Bernoulli distribution of the relevance scores of selected items and micro-level reward as average clicks per user.

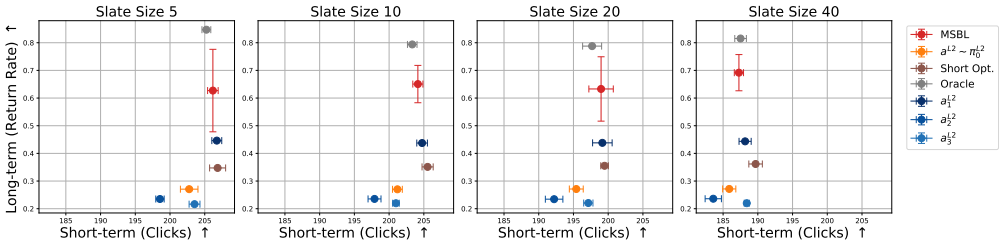
We form item groups based on the upload type of the video and user groups based on the users' activity level feature. At the upper level, each user group has an unknown preference $p_{u,i}$ for a particular item group. We use $p_{u,i} = 0.9$ for the preferred user and item group pair. The user retention probability p_r is simulated as this preference-weighted fraction of items selected, given by $p_r = \text{softmax}\big(\log(\sum_i p_{u,i} \frac{n_i}{\sum_i n_i})\big)$, where n_i is the number of items selected from group i over $t1 = \{1, \ldots, 5\}$ timesteps of the lower level. The reward r^{L2}, representing the return rate is simulated as the inverse of the return day to the app.

For training the lower level policy, which is a transformer model that uses input as the item and user embedding and predicts relevance scores, we use the user features and pre-processing following [45].

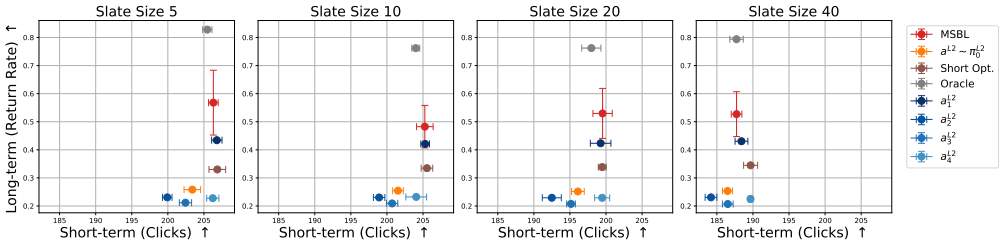
For the upper level bandit policy, we use 5 user features, namely 'uf_user_active_degree','uf_is_live_streamer','uf_is_video_author', 'uf_follow_user_num_range', 'uf_fans_user_num_range', with their definitions provided in [12].



(a) 2 groups



(b) 3 groups



(c) 4 groups

Figure 12: Recommender system: Tradeoff between long term return rate and clicks by **varying slate size** using the boost a^{L2} to the relevance scores as **policy modification** across 5 random seeds when the **number of groups vary** in (a) 2 groups (b) 3 groups and (c) 4 groups.

Training Details For bandit policy π^{L2} at upper level, we use an embedding module that embeds the raw user features into 16-dimensional vectors and a 4 layer neural network with hidden dimension 128. We train using AdamW optimizer, with a batch size of 128, learning rate 1e-4, for 4000 epochs.

Increasing Level 1 and Level 2 action space. Figure 12 shows the tradeoff with **varying groups** in $\{2, 3, 4\}$ and with varying Level 1 actions as the slate size top-k varies for $k \in \{5, 10, 20, 40\}$. We observe that the return rate for MSBL remains high across all sizes and for all groups. Since we report average clicks per user, there is a decrease in the short term (clicks) with increasing slate size. This experiment demonstrates that even with large micro action space $\mathcal{A}^{L1}$, MSBL can effectively leverage the interventions to drive the system toward the long-term reward.

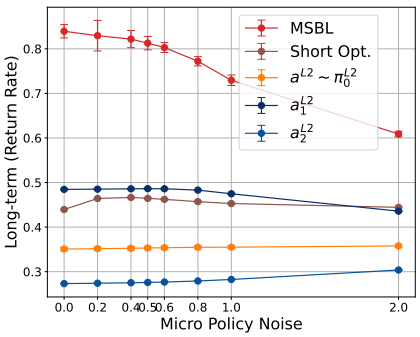


Figure 13: Long term return rate with **varying noise** σ_s in micro policy varies in $\{0.0, 0.2, 0.4, 0.6, 0.8, 1.0, 2.0\}$.

Robustness to errors in the micro policy. To evaluate the robustness with varying errors in the micro policy, we simulate a perturbed policy $\tilde{\pi}^{L1}_{a^{L2}} := \arg\max_k \tilde{s}$, where noisy relevance scores $\tilde{s}$ are sampled from $\mathcal{N}(s, \sigma_s^2)$, with $\sigma_s \in \{0.0, 0.2, 0.4, 0.6, 0.8, 1.0, 2.0\}$. Figure 13 shows long term reward as this noise varies for top-10 selection with 2 groups. We find that learning the macro policy is robust for fairly large noise σ_s and only begins to deteriorate for high values of σ_s. This experiment also demonstrates the advantage of learning micro policy to the long term return rate metric. In real-world deployment, the micro policy may be updated periodically, so this robustness to errors makes the nested bandit learning practically appealing.