Large Language Model-Enhanced RL for Diverse and Novel Recommendations

Jiin Woo[*]
Carnegie Mellon University
jiinw@andrew.cmu.edu

Alireza Bagheri Garakani
Amazon
alirezg@amazon.com

Tianchen Zhou
Amazon
tiancz@amazon.com

Zhishen Huang
Amazon
hzs@amazon.com

Yan Gao
Amazon
yanngao@amazon.com

Abstract

In recommendation systems, diversity and novelty are essential for capturing varied user preferences and encouraging exploration, yet many systems prioritize click relevance. While reinforcement learning (RL) has been explored to improve diversity, it often depends on random exploration that may not align with user interests. We propose LAAC (LLM-guided Adversarial Actor Critic), a novel method that leverages large language models (LLMs) as reference policies to suggest novel items, while training a lightweight policy to refine these suggestions using system-specific data. The method formulates training as a bilevel optimization between actor and critic networks, enabling the critic to selectively favor promising novel actions and the actor to improve its policy beyond LLM recommendations. To mitigate overestimation of unreliable LLM suggestions, we apply regularization that anchors critic values for unexplored items close to well-estimated dataset actions. Experiments on real-world datasets show that LAAC outperforms existing baselines in diversity, novelty, and accuracy, while remaining robust on imbalanced data—effectively integrating LLM knowledge without expensive fine-tuning.

1 Introduction

In recommendation systems (RS), diversity and novelty are crucial factors to address diverse user preferences and promote exploration of new interests. Many search and recommendation systems tend to prioritize click relevance, which often limits diversity, especially when data or user experiences are limited. Recognizing that reinforcement learning (RL) can offer personalized and dynamic recommendations by prioritizing long-term user satisfaction, several studies have investigated RL methods, enhancing diversity in recommendations [29, 24, 36] by encouraging random exploration or introducing explicit rewards for diversity. Such approaches may not consistently yield optimal results, as randomly selected items may not cater to users' needs.

A more effective approach is to leverage prior knowledge to explore novel items selectively. Large language models (LLMs), trained on enormous corpus of data, can be a reliable source of novel items beyond the system's limited user experiences. To utilize LLMs to enhance RS, some works have used LLMs as recommenders by generating recommendations directly from LLMs with prompts [15] or fine-tuning LLMs [3], but these approaches incur significant computational costs for inference. Meanwhile, recent research has suggested to use LLMs as an environment component to augment limited training data [32], reducing inference-time computational costs. This approach still requires

[*]Work done during internship at Amazon. Correspondence to: Jiin Woo <jiinw@andrew.cmu.edu>

2nd Workshop on Aligning Reinforcement Learning Experimentalists and Theorists (ARLET 2025).

significant computational resources for LLM training and risks losing the diverse user preferences learned during pretraining when models are retrained on specific datasets. Therefore, it is crucial to strategically integrate LLM suggestions into RL training without costly fine-tuning, to ensure fast adaptation to new data, scalability, and preservation of diverse user preferences learned during pretraining.

To address this issue, we investigate an RL method that uses LLM as a reference policy rather than retraining it. We can leverage its ability to suggest potentially appealing items and train a separate, lightweight policy to refine and align these recommendations using datasets collected from target systems. This approach enables practical deployment and personalization without modifying the LLM itself. However, using LLMs' suggestions without additional training can be risky, as not all recommendations of LLMs meet user expectations in the target systems, potentially compromising user satisfaction. Therefore, it's crucial to be selectively optimistic on LLMs' suggestions likely to yield high rewards, while also retaining popular items proven effective in datasets. To this end, we formulate the RL problem as a two-player game between a policy and a critic function [7, 4], where the policy seeks to learn actions that selectively improve upon the LLM's suggestions, while the critic learns to provide realistic value estimates that encourage promising novel recommendations from the LLM while remaining grounded in dataset observations. The adversarial dynamics between the policy and critic prevent both greedy exploitation of popular items in the datasets and blind optimism towards LLM recommendations, enabling the selective integration of novel items outside the dataset's coverage.

In this paper, we present an LLM-guided adversarial actor-critic training method (LAAC) that iteratively refines the actor and critic networks based on LLM recommendations and datasets, optimizing the adversarial objectives against one another. To prevent overestimation on unreliable LLM suggestions, we incorporate novel regularizations that ensure the critic values for unexplored items remain close to those for actions reliably estimated from the dataset. As a result, the algorithm develops a balanced policy that recommends both popular items in the dataset and novel items suggested by LLMs. We summarize our main contributions as follows:

- We propose LAAC, a novel adversarial actor-critic method that leverages LLMs as reference policies to guide exploration toward diverse and novel items, replacing uninformed exploration in existing diversity-promoting RL approaches with targeted LLM-guided exploration.
- We demonstrate that LAAC consistently outperforms both baselines across accuracy, diversity, and novelty metrics on real-world data, achieving superior diversity-accuracy trade-offs without requiring costly LLM fine-tuning.
- We provide comprehensive analysis of the method's regularization mechanisms, showing how grounding loss (α) and temporal difference loss (β) parameters control the balance between optimism for novel LLM suggestions and realism for dataset observations, along with robustness analysis on imbalanced data.

1.1 Related work

1.1.1 Reinforcement learning for diversity and novelty in recommendation systems

In RS, diversity and novelty are key factors for enhancing user experience and uncovering hidden preferences. Recent studies have explored RL algorithms to deliver more diverse and dynamic recommendations, focusing on long-term user satisfaction. To enhance diversity, [36] examined exploration-exploitation strategies in RL by randomly selecting item candidates from the vicinity of the currently recommended item. To maximize diversity in recommendation results, while preserving relevance, [24] integrated a determinantal point process model with the deep deterministic policy gradient algorithm. Lastly, [29] introduced a scalarized multi-objective RL algorithm that optimizes three key reward objectives: accuracy, diversity, and novelty.

1.1.2 Large language models for recommendation systems

LLMs [35, 23, 26], pre-trained on large natural language datasets, demonstrate improved transfer capabilities and are receiving growing interest in the field of recommendation systems [6, 22, 11]. To harness LLMs for recommendation tasks, some studies have suggested using LLMs as recommenders directly, generating suggestions with proper prompts in a zero-shot manner [15]. Additionally, to

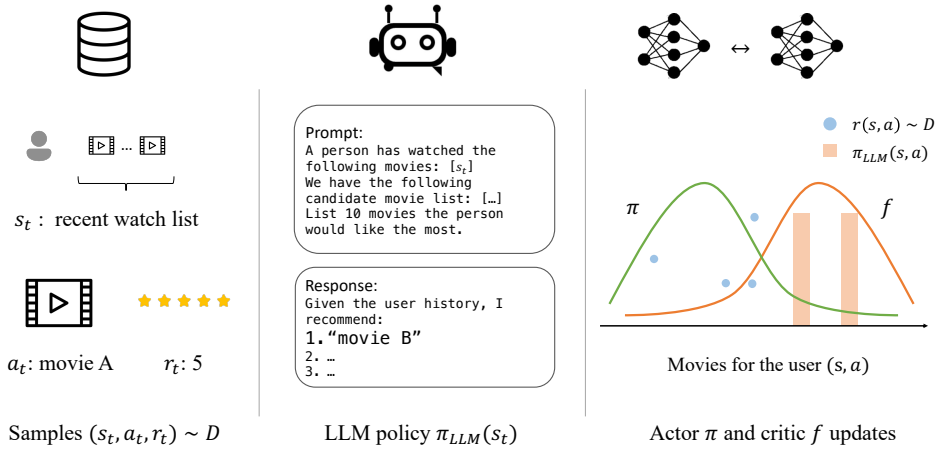


Figure 1: Overview of LAAC. At each training step t, transition samples are drawn from dataset $\mathcal{D}$ and the corresponding LLM policy π_{LLM} is constructed based on LLM recommendations. The critic f is updated to favor novel actions suggested by π_{LLM} over the policy π, and subsequently the policy π is refined using the updated critic f.

align LLMs with specific recommendation systems, other research has proposed retraining these models to serve as new recommenders via fine-tuning [26] or prompt-based tuning [23, 27]. Recent studies [17, 33] have explored training recommenders by analyzing user-item interactions alongside item features extracted from BERT [10], yielding promising results for cross-domain recommendations. Furthermore, [3] efficiently fine-tunes LLaMA-7B [31] using LoRA adapters [18] and an instruction prompt that incorporates item text descriptions to facilitate few-shot recommendations. However, these methods for employing LLMs as recommenders often entail significant time and computational costs during both training and inference. Recently, [32] has proposed a method that uses LLMs as an environment component within an RL framework, with the objective of augmenting the performance of existing recommenders, which reduces computational costs at inference time. Nevertheless, this approach still requires considerable computation for LLM training to function as an environmental simulator, and it risks losing the diverse user preferences learned during pre-training when LLMs are retrained on specific datasets.

2 Method

2.1 Preliminary

2.1.1 RL formulation

In RL, sequential RS can be framed as a Markov Decision Process (MDP), where an agent interacts with users by sequentially recommending items to maximize clicks or ratings. The MDP is characterized by $M = (\mathcal{S}, \mathcal{A}, P, R, \rho_0, \gamma)$, where $\mathcal{S}$ represents the user's state, $\mathcal{A}$ is a set of candidate action items, and the transition probability $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ defines $p(s'|s, a)$, indicating the likelihood of transitioning from state s to state s' when action a is taken. The reward function $r : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$ specifies that $r(s, a)$ gives a rating or click from users for item a in state s. ρ_0 denotes the initial state distribution and $\gamma \in [0, 1]$ is the discount factor. A policy π denotes an action-selection rule where $\pi(a|s)$ represents the probability of taking action a in state s.

For a given policy π, the state-action value function (i.e., *Q-function*) $Q^\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, which measures the expected discounted cumulative reward from an initial state-action pair (s, a), defined as

$$Q^\pi(s, a) \coloneqq r(s, a) + \mathbb{E}\left[\sum_{t=1}^{\infty} \gamma^t r(s_t, a_t) \,\middle|\, s_0 = s, a_0 = a\right].$$

Here, the expectation is taken with respect to the randomness of the trajectory $\{s_t, a_t, r_t\}_{t=0}^{\infty}$, sampled based on the transition probability (i.e., $s_{t+1} \sim P(\cdot|s_t, a_t)$) and the policy π (i.e., $a_t \sim \pi(\cdot|s_t)$) for

any $t \geq 0$. In this setting, the goal of the agent is to learn an optimal policy $\pi^\star$ that maximizes expected total rewards (ratings), i.e., $\pi^\star = \arg\max_\pi \mathbb{E}_{s\sim\rho_0, a\sim\pi(s)}\left[Q^\pi(s,a)\right]$.

2.1.2 Limited datasets and a reference policy

When the underlying MDP is unknown, the optimal policy can be trained from a dataset $\mathcal{D}$ composed of transition samples (s, a, r, s') collected from past user interactions, which provide the user response $r = r(s,a)$ and the user state transition $s' \sim P(\cdot|s,a)$ for an interacted item a at state s. However, the vast item space in RS makes exhaustive exploration impractical, leading the agent – trained on limited datasets – to focus on a narrow selection of popular items in datasets, which reduces diversity and overlooks potentially appealing yet unexplored items. One effective approach to address the limitation of datasets with insufficient coverage of good items is to utilize a *reference policy*. In many RL studies, leveraging baseline/reference policies or experts to regularize RL training has been shown to enhance sample efficiency and training stability [28, 34, 4, 19, 30]. Especially, when datasets lack sufficient observations to determine the optimal actions, these reference policies can offer safe alternatives. With the support of a reliable reference policy, agents can effectively learn novel actions beyond the dataset with minimal risk.

2.1.3 Robust policy improvement via adversarial optimization

A game-theoretic formulation for RL has been introduced in [7] to safely improve a behavior policy (the policy used to collect training data), formulated as a bilevel optimization problem:

$$\max_\pi g(\pi, f^\pi), \text{ s.t. } f^\pi \in \arg\min_f h(\pi, f),$$

where the two competing players are a policy π (action selector) and a critic f (value estimator), with g and h as their respective objective functions. They proposed an actor-critic algorithm (ATAC) that solves this optimization through adversarial training, where the policy and critic compete against each other. The policy π improves based on values predicted by the critic f^π, which deliberately provides pessimistic evaluations of π compared to the original behavior policy used for dataset collection. This adversarial training setup guarantees that the resulting policy provably outperforms the behavior policy across a wide range of hyperparameter choices, though it restricts the policy to only recommend actions that were observed in the training dataset. Recently, [4] extended this adversarial optimization framework to incorporate an arbitrary reference policy (which can be different from the behavior policy) in model-based RL settings. This extension enables developing an improved policy based on the reference policy, allowing actions beyond the available data support.

2.2 Adversarial optimization guided by LLMs

In this paper, we explore an RL approach that enhances the diversity of a policy through adversarial optimization [7, 4], using LLMs as a reference policy to suggest novel and potentially appealing items beyond the dataset, thereby improving diversity while avoiding excessive exploration. Given their proven effectiveness in recommendation tasks [9, 15], LLMs serve as a strong candidate for a reliable reference policy in RS. Here, we denote a reference policy generated from LLMs as π_{LLM}, where $\pi_{\text{LLM}}(a|s)$ indicates the probability of the LLM suggesting action a for state s. Our goal is to obtain a balanced policy that recommends a diverse mix of popular and novel items by carefully managing the trade-off between exploring new items from the LLM policy π_{LLM} and exploiting high-reward popular items from the dataset $\mathcal{D}$.

To integrate novel items from the LLM policy while retaining popular items from the dataset, a policy should be trained to select the novel items only when data is insufficient and optimal actions are uncertain, while prioritizing high-reward items when enough observations are available. To this end, inspired by the game-theoretic formulation of RL [7, 4], we formulate the problem as the following minimiax optimization:

$$\begin{aligned}
\hat{\pi} &= \arg\max_\pi E_{(s,a,r,s')\sim\mathcal{D}}\Big[f(s,\pi) - f(s,\pi_{\text{LLM}})\Big] \\
\text{s.t. } f &= \arg\min_f E_{(s,a,r,s')\sim\mathcal{D}}\Big[f(s,\pi) - f(s,\pi_{\text{LLM}})\Big] \\
&\qquad + \alpha\mathcal{E}_{\text{g}}(f,\pi_{\text{LLM}}) + \beta\mathcal{E}_{\text{td}}(f,\pi) \qquad (1)
\end{aligned}$$

Algorithm 1 LAAC (LLM-guided Adversarial Actor Critic)

Input: Dataset $\mathcal{D}$, discount factor $\gamma \in [0, 1]$, constants $\alpha,\ \beta \geq 0$, learning rates $\eta_{\text{critic}},\ \eta_{\text{actor}}$, candidate size n_c, response size n_r

Initialize critic networks $f_1,\ f_2$ and a actor network π.
for $i = 1, 2, \ldots, N$ **do**
  Sample minibatch $\mathcal{D}_{\text{mini}}$ from dataset $\mathcal{D}$.
  Initialize $\mathcal{L}(f, \pi), \mathcal{E}_{\text{g}}(f), \mathcal{E}_{\text{td}}(f) = 0$ for each $f \in \{f_1, f_2\}$.
  for $(s, a, r, s') \in \mathcal{D}_{\text{mini}}$ **do**
    Generate $\pi_{\text{LLM}}(s)$ from LLM responses.
    Compute losses for each critic $f \in \{f_1, f_2\}$
    $\mathcal{L}(f, \pi) += f(s, \pi) - f(s, \pi_{\text{LLM}})$
    $\mathcal{E}_{\text{g}}(f, \pi_{\text{LLM}}) += (f(s, a) - f(s, \pi_{\text{LLM}}))^2$
    $\mathcal{E}_{\text{td}}(f, \pi) += (f(s, a) - r - \gamma f(s', a'))^2$, where $a' \sim \pi(s')$
  end for
  Update critic networks $f \in \{f_1, f_2\}$
  $l_{\text{critic}}(f) \coloneqq \frac{1}{|\mathcal{D}_{\text{mini}}|}(\mathcal{L}(f, \pi) + \alpha \mathcal{E}_{\text{g}}(f, \pi_{\text{LLM}}) + \beta \mathcal{E}_{\text{td}}(f, \pi))$
  $f \leftarrow f - \eta_{\text{critic}} \nabla l_{\text{critic}}(f, \pi)$
  Update actor network π
  $l_{\text{actor}}(\pi) = -\frac{1}{|\mathcal{D}_{\text{mini}}|}\mathcal{L}(f_1, \pi)$
  $\pi \leftarrow \pi - \eta_{\text{actor}} \nabla l_{\text{actor}}$
end for

where $\alpha,\ \beta \geq 0$ are hyper parameters for the following losses:

$$\mathcal{E}_{\text{g}}(f, \pi_{\text{LLM}}) \coloneqq E_{(s,a,r,s') \sim \mathcal{D}}\Big[(f(s, a) - f(s, \pi_{\text{LLM}}))^2\Big]$$
$$\mathcal{E}_{\text{td}}(f, \pi) \coloneqq E_{(s,a,r,s') \sim \mathcal{D}}\Big[(f(s, a) - r - \gamma f(s', \pi))^2\Big].$$

Here, $f(s, \pi) = E_{a \sim \pi(\cdot|s)}[f(s, a)]$, where the critic function $f(s, a)$ represents the estimated expected reward for item a at state s, approximating the Q-function $Q(s, a)$. This optimization can be viewed as a two-player game, where one player refines a policy π in relation to the LLM policy π_{LLM} based on a given critic value f, while the other player adversarially updates the critic f to encourage optimism towards novel actions suggested by π_{LLM} over π. Ultimately, this process optimizes the policy π for the worst-case performance (f) inferred from both the dataset $\mathcal{D}$ and the LLM policy π_{LLM}. Notably, the optimization process only necessitates updating the policy π and the critic f, without the need for training of LLMs, which can be computationally intensive and time-consuming. In this manner, we can guide the policy to learn only effective actions from π_{LLM} consistent with the provided dataset $\mathcal{D}$, even if π_{LLM} it is not perfectly aligned with the target system.

Grounding the critic function via regularization. To guarantee compatibility with the target system without direct modification of π_{LLM}, we introduce the regularization losses $\mathcal{E}_{\text{g}}$ and $\mathcal{E}_{\text{td}}$ in (1).

- **Temporal difference loss ($\mathcal{E}_{\text{td}}$)**: The TD loss enforces Bellman consistency in the critic (f), ensuring it learns realistic values for in-sample actions aligning with the dataset $\mathcal{D}$. For frequently observed actions in $\mathcal{D}$, the loss strongly aligns the critic f with actual reward observations, while for actions rarely observed in $\mathcal{D}$, the loss has a smaller effect, allowing the critic f to maintain optimistic values.
- **Grounding loss ($\mathcal{E}_{\text{g}}$)**: Limiting critic values only for in-sample actions is insufficient, as it may still overestimate values for LLM-suggested items not seen in $\mathcal{D}$. To address this, we introduce a grounding loss that constrains $f(s, \pi_{\text{LLM}})$, the critic values of π_{LLM}, to stay close to those of in-sample actions. This reduces excessive optimism for unexplored actions and ensures a grounded evaluation based on the reliable values of popular actions.

With proper choices of α and β, the critic f learns to estimate values for novel items that are optimistic yet still grounded within the constraints of the training datasets. Meanwhile, the policy π evolves to generate recommendations that strike a balance between exploring novel items and exploiting popular ones. We will analyze the impact of α and β on performance in Section 3.

Dataset	Models	Accuracy						Reward			Diversity			Novelty		
		HR@5	HR@10	HR@20	NDCG@5	NDCG@10	NDCG@20	R@5	R@10	R@20	CV@10	CV@20	Entropy	NCV@10	NCV@20	NC@1
MovieLens	π_{Llama3}	0.0036	0.0064	0.0115	0.0022	0.0031	0.0044	55	99	179	0.9803	0.9803	2.2747	0.9692	0.9692	1,731
	π_{Claude3}	0.0061	0.0112	0.0161	0.0039	0.0055	0.0067	93	169	242	0.5318	0.5335	2.0651	0.2989	0.3018	209
	GRU4Rec	0.0401	0.0644	0.1026	0.0261	0.0339	0.0435	622	994	1,574	0.6773	0.7350	1.5207	0.3764	0.4782	219
	SMORL	0.0380	0.0620	0.0994	0.0243	0.0320	0.0414	587	954	1,523	0.6682	0.7310	-	0.3644	0.4727	191
	LAAC (Llama3)	**0.0458**	**0.0720**	**0.1104**	0.0303	0.0387	0.0484	711	**1,109**	1,687	**0.6899**	**0.7674**	8.0594	**0.4235**	**0.5464**	268
	LAAC (Claude3)	**0.0458**	**0.0720**	**0.1104**	**0.0305**	**0.0389**	**0.0486**	**712**	**1,109**	**1,690**	0.6877	0.7646	**8.0595**	0.4192	0.5410	**278**

Table 1: Performance comparison of baseline methods (GRU4Rec and SMORL) and LAAC evaluated on the MovieLens dataset across accuracy, reward, diversity, and novelty metrics. Best scores are highlighted in boldface, excluding the standalone LLM policies π_{Llama3} and π_{Claude3}. LAAC (Llama3) and LAAC (Claude3) represent LAAC trained with π_{Llama3} and π_{Claude3} reference policies, respectively.

2.3 LLM-guided adversarial actor critic (LAAC)

To solve the optimization problem (1), we present an LLM-guided Adversarial Actor Critic (LAAC), which constructs an LLM policy π_{LLM} from LLMs' responses and trains actor (π) and critic (f) networks to optimize their empirical losses computed based on batch datasets and π_{LLM}. The complete description of LAAC is provided in Algorithm 1.

2.3.1 Constructing LLM policy π_{LLM}

Obtaining the exact probability distribution of the LLM policy π_{LLM} over a large action space is challenging, and LLMs may suggest actions outside the predefined space. To address this, we extract recommendations $\mathcal{A}_r$ from the LLM by providing a prompt $p(s, \mathcal{A}_c, n_r)$, which includes a candidate action set $\mathcal{A}_c \subseteq \mathcal{A}$, the size of recommendations n_r, and the current state s. We then construct the LLM policy π_{LLM} as uniformly distributed over $\mathcal{A}_r$. To control prompt and response length, we randomly sample $\mathcal{A}_c$ from $\mathcal{A}$ and specify $n_r = |\mathcal{A}_r|$. We adopt a uniform policy over the list of items recommended by the LLM to ensure broader coverage of its suggestions, rather than disproportionately focusing on the top-ranked item. This design encourages exploration and aligns with our goal of promoting diversity and novelty in recommendations. The example of the prompts and LLM responses in movie recommendation scenarios is illustrated in Figure 1.

2.3.2 Adversarial actor critic training

We present a practical actor-critic approach that iteratively updates actor (π) and critic (f) networks to optimize the adversarial objectives in (1), where the loss is computed based on a sampled mini-batch dataset $\mathcal{D}_{\text{mini}} \subseteq \mathcal{D}$ and the LLM policy π_{LLM}. During the critic updates (Line 9 in Algorithm 1), the critic networks f are updated to minimize the adversarial loss $\mathcal{L}(f, \pi)$ added with the regularization terms $\mathcal{E}_{\text{g}}(f)$ and $\mathcal{E}_{\text{td}}(f)$. This increases the values of novel actions recommended by π_{LLM} in comparison to the actions of π. To avoid the deadly triad issue, we implement the double Q heuristic [7, 12, 13], using two critic networks (f_1 and f_2) for loss computation. Once critic networks are chosen, the actor network π is updated to maximize the adversarial loss $\mathcal{L}(f, \pi)$, in contrast to the critic updates, thereby improving the policy π in relation to the LLM policy π_{LLM} (Line 10 in Algorithm 1). After N iterations of the updates, the algorithm outputs the actor π and the critics f_1 and f_2.

3 Experiment

In this section, we evaluate the performance of LAAC for recommendation on real-world datasets.

3.1 Experimental settings

3.1.1 Datasets

We perform experiments using the real-world movie rating dataset MovieLens-1M [14], including 1 million ratings for 3,503 movies. Items with fewer than three interactions and users whose interaction length is smaller than three are removed from the datasets. From the datasets, we randomly sample ratings, organize them chronologically, and group them by user to create a sequence of interactions for each individual. For each user sequence, at each time step t, we define a state as $s_t = G(x_{t-5:t})$, where $x_{t-5:t}$ represents the five most recent items the user has watched prior to time step t, and G is an encoder for the sequential model that transforms the input sequence into a hidden state. Additionally, action a_t corresponds to an item ID rated by the user, and

Dataset	Models	Accuracy						Reward			Diversity			Novelty		
		HR@5	HR@10	HR@20	NDCG@5	NDCG@10	NDCG@20	R@5	R@10	R@20	CV@10	CV@20	Entropy	NCV@10	NCV@20	NC@1
MovieLens (skewed)	GRU4Rec	0.0387	0.0613	**0.0962**	0.0251	0.0324	0.0412	606	953	**1,485**	0.6460	0.6913	1.6640	0.3433	0.4117	202
	SMORL	0.0330	0.0535	0.0850	0.0215	0.0281	0.0360	515	830	1,314	0.6366	0.6900	-	0.3312	0.4107	169
	LAAC (Llama3)	**0.0399**	**0.0618**	0.0932	0.0267	**0.0337**	**0.0416**	**624**	**958**	1,438	**0.6722**	**0.7444**	8.0545	**0.4041**	**0.5147**	264
	LAAC (Claude3)	0.0397	0.0610	0.0929	**0.0268**	0.0336	**0.0416**	621	949	1,431	0.6672	0.7397	**8.1613**	0.3959	0.5062	**265**

Table 2: Performance comparison between baseline methods (GRU4Rec and SMORL) and LAAC on the skewed MovieLens dataset, which contains exclusively male user samples. All methods are evaluated on the MovieLens dataset with its original user distribution across metrics of accuracy, reward, diversity, and novelty. Boldface indicates best performance. LAAC (Llama3) and LAAC (Claude3) represent LAAC variants using π_{Llama3} and π_{Claude3} reference policies, respectively.

reward r_t is the rating given by the user for that item, which ranges from $[1, 5]$. Accordingly, we obtain 26,511 samples from 160 users in the MovieLens dataset, along with the corresponding suggestions generated from LLMs. Of these, 21,421 samples are used for training, while 5,090 samples are reserved for evaluation.

3.1.2 Baselines

- **LLM policy (π_{LLM}):** We establish the LLM policy as a baseline to understand the raw potential and limitations of LLM suggestions for diversity and accuracy. To ensure our evaluation is robust across different LLMs, we use two distinct LLMs to construct the LLM policy: LLama3-8B-Instruct [1] and Claude3 Haiku [2]. The LLM policy operates by generating responses to prompts that describe states in text format (as detailed in Section 2.3.1), then creating a uniform distribution over the items suggested in these responses. We denote the resulting baseline policies as π_{Llama3} and π_{Claude3} respectively.
- **GRU4Rec** [16]: GRU4Rec is a recurrent neural network model for RS, utilizing gated recurrent units (GRUs) [8] to encode sequences of user-item interactions. It is trained in a supervised manner using cross-entropy loss to predict the next item.
- **Scalarized multi-objective RL (SMORL)** [29]: SMORL is a multi-objective RL approach specifically designed to improve diversity and novelty in RS. It performs scalarized Q-learning for three different objectives – accuracy, diversity, and novelty – by employing distinct output layers for each of these objectives.

3.1.3 Implementation details.

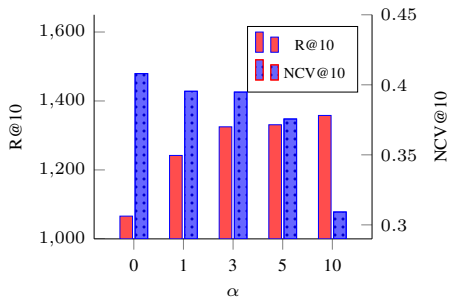
For all algorithms, except π_{LLM}, to generate state $s_t = G(x_{t-5:t})$, we use the five most recent items' embeddings $x_{t-5:t}$ and GRUs [8] to encode the input sequence. The item embedding size and hidden size are both set to 64 for all models. The learning rate is set as 0.005 for GRU4Rec, 0.01 for SMORL as suggested in [29], and $\eta_{\text{critic}} = 0.01$ and $\eta_{\text{actor}} = 0.001$ for LAAC to ensure the critic stably evaluates the policy [5, 25]. For the RL algorithms, we set the discount factor γ to 0.5 for SMORL as suggested in [29] and 0.99 for LAAC. For SMORL, we set equal weights for each reward objective for accuracy, novelty and diversity, i.e., $\boldsymbol{w} = [1, 1, 1]$. For LAAC, the default settings for the regularization coefficients are $\alpha = 1.0$ and $\beta = 1.0$. We use adaptive gradient descent algorithm ADAM [21] and train the models for 10,000 steps with a minibatch size of 128.

The LLM policy π_{LLM} is extracted directly from an LLM by prompting state information in text format. To generate $\pi_{\text{LLM}}(s_t)$ for a given state $s_t = x_{t-5:t}$, we provide the titles of five items the user has selected before t in a prompt to the LLM model. To limit prompt and response length, we provide 100 candidate items randomly sampled from the entire item set and specify $n_r = 10$ recommended items, i.e., $n_c = 100$ and $n_r = 10$. See Section 2.3.1 for details.

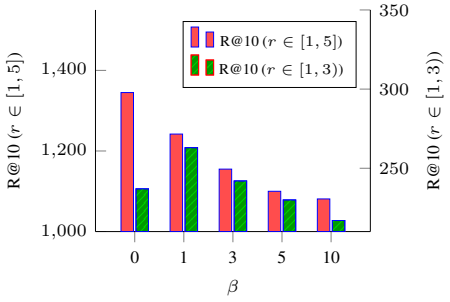
A key practical advantage of LAAC is its computational efficiency profile. During training, LLM calls can be batched and parallelized across training samples. Crucially, once training is complete, the learned policy π operates independently of the LLM, requiring only lightweight neural network inference comparable to traditional recommenders like GRU4Rec. This enables real-time recommendations with sub-millisecond latency, contrasting sharply with direct LLM recommendation approaches that require expensive LLM inference for each user query and can introduce latencies of hundreds of milliseconds to seconds, making them impractical for real-time applications.

3.1.4 Metrics

- **Accuracy/reward:** We consider two commonly used metrics for their top-k predictions, where $k = 5, 10$ and 20: hit ratio (HR@k) and normalized discounted cumulative gain (NDCG@k) [20]. HR@k evaluates whether the ground-truth item appears in the top-k positions of the recommendation list, while NDCG@k is a rank-sensitive metric that assigns higher scores to items positioned at the top of the recommendation list.



(a) Effect of α on reward and novelty



(b) Effect of β on reward

Figure 2: Performance analysis of LAAC (Llama3) on MovieLens dataset. **Left:** Evaluated rewards (R@10) and novelty (NCV@10) for varying $\alpha = 0, 1, 3, 5, 10$. Higher α improves reward but reduces novelty. **Right:** Evaluated rewards (R@10) for varying $\beta = 0, 1, 3, 5, 10$ on the full dataset ($r \in [1, 5]$) and the filtered dataset only consisting of samples with poor ratings ($r \in [1, 3)$). Too low β dcreases cumulative reward (R) when datasets consist of poor actions with low ratings.

To assess the long-term effect of algorithms' item recommendations in terms of high ratings, we measure cumulative reward (R@k), defined as the sum of user ratings for the top-k recommendations.

- **Diversity/novelty:** We measure item coverage (CV@k), novel item coverage (NCV@k), and the total count of novel items (NC@k). Here, novel items indicate items fall within the 50% tail in terms of popularity. More specifically, CV@k is a ratio of all items (less popular items) covered by all top-k recommendations of the test sequences, and NCV@k is the item coverage ratio only computed for novel items that fall within the 50% tail in terms of popularity. For GRU4Rec, π_{LLM}, and LAAC, learning stochastic policies, we measured the entropy of the policies to assess recommendation diversity, where higher entropy indicates more uniform probability distributions across items and thus greater potential for diverse recommendations.

All models were trained 50 times independently using different random seeds, and the final performance metrics were obtained by averaging the results across these trials.

3.2 Analysis

3.2.1 LLM Policy: diversity with misaligned relevance

We assess the performance of the LLM policies, π_{Llama3} and π_{Claude3}, directly derived from the responses of Llama3 and Claude3 without any additional training. As shown in Table 1, while π_{Llama3} exhibits notable diversity and novelty in its recommendations, their accuracy and cumulative rewards are low. Although π_{Claude3} shows relatively higher accuracy and rewards, it still underperforms compared to other baselines trained on the dataset. This outcome is expected, as the LLMs are not perfectly aligned with user preferences in the dataset, resulting in recommendations of lower quality that fail to effectively meet the needs of target users.

3.2.2 LAAC: diversity with aligned relevance

We evaluate LAAC against baseline methods, with SMORL serving as our diversity-focused RL baseline and GRU4Rec providing a strong sequential recommendation baseline. As presented in Table 1, both LAAC (Llama3) and LAAC (Claude3) consistently outperform all baselines across relevance and diversity/novelty metrics, demonstrating the effectiveness of incorporating LLM knowledge. The improvements over SMORL are particularly notable since both methods use identical GRU-based encoders, highlighting that LLM-guided exploration is more effective than scalarized multi-objective optimization for achieving diversity-accuracy trade-offs. While SMORL balances accuracy, diversity, and novelty through weighted objectives with relatively uninformed exploration, LAAC's LLM guidance provides more targeted exploration toward novel items likely to interest users. Remarkably, LAAC achieves both high accuracy and diversity without trade-offs, even though the LLM policies are not perfectly aligned with the target dataset, proving that unrefined LLM knowledge can be effectively leveraged while maintaining computational efficiency comparable to traditional recommenders.

3.2.3 The effect of LLM capabilities.

To validate that our algorithm's effectiveness generalizes across different language models, we evaluate LAAC trained with two distinct LLMs: LLama3-8B-Instruct [1] and Claude3 Haiku [2], corresponding to baseline policies π_{Llama3} and π_{Claude3} respectively. Our results demonstrate consistent performance improvements from

LAAC across both models. In terms of accuracy and reward, LAAC (Llama3) and LAAC (Claude3) achieve similar performance levels, despite π_{Claude3} initially outperforming π_{Llama3} on the target task. This suggests that the original LLM performance has minimal impact on our algorithm's final effectiveness, as the relevance alignment achieved through our training process sufficiently compensates for initial capability differences using the available dataset samples. Interestingly, for diversity and novelty metrics, LAAC (LLama3) outperforms LAAC (Claude3), which aligns with the relative strengths observed in the base LLM performance.

3.2.4 Robustness to skewed datasets

In RS, the distribution of users in training datasets may differ from that of target users in real applications, making robustness to distribution shifts crucial. To evaluate this, we create a skewed dataset with 14,883 ratings from 90 male users and train LAAC and baseline algorithms on it. As shown in Table 2, LAAC and GRU4Rec perform well in almost all metrics accuracy, diversity, and novelty, while SMORL's diversity performance declines significantly with the skewed dataset. These results demonstrate that LAAC's learned diversity and novelty handle bias from imbalanced data, generalizing well to different distributions.

3.2.5 The effect of grounding loss (α)

The grounding loss $\mathcal{E}_{\text{g}}$ constrains the critic values of novel actions suggested by the LLM to align with the values of in-sample actions. As shown in Figure 2, increasing α reduces novelty but improves accuracy. This occurs because large α restricts the optimism toward novel suggestions from the LLM, while promoting stable critic learning by ensuring that out-of-sample actions' values align with those of in-sample actions reliably learned with enough observations, which results in more accurate critic estimates.

3.2.6 The effect of TD loss (β)

The TD loss $\mathcal{E}_{\text{td}}$ regularizes the critic f by enforcing Bellman consistency to learn realistic values for in-sample actions. As shown in Figure 2, increasing β reduces accuracy, as it makes action estimates less optimistic. However, a very small β harms performance by making the policy blindly mimics actions observed in datasets or suggested by the LLM, ignoring actual rewards. We additionally tested our algorithm on a low-quality dataset (actions with ratings below 3) across different β values and found that rewards decrease with $\beta = 0$, demonstrating that too small β is detrimental, especially with poor-quality datasets.

4 Conclusion

We propose a novel RL method that enhances diversity and novelty in recommendations without compromising accuracy, by leveraging LLM-generated suggestions without the need for expensive LLM fine-tuning. Our approach learns a balanced policy capable of recommending both popular and novel items. Experiments on real-world datasets demonstrate that our method achieves significant gains in diversity, novelty, and accuracy, highlighting its effectiveness and practicality for real-world recommendation systems.

References

[1] AI@Meta. Llama 3 model card. 2024.

[2] Anthropic. The claude 3 model family: Opus, sonnet, haiku. 2024.

[3] Keqin Bao, Jizhi Zhang, Yang Zhang, Wenjie Wang, Fuli Feng, and Xiangnan He. Tallrec: An effective and efficient tuning framework to align large language model with recommendation. RecSys '23, page 1007–1014, 2023.

[4] Mohak Bhardwaj, Tengyang Xie, Byron Boots, Nan Jiang, and Ching-An Cheng. Adversarial model for offline reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, pages 1245–1269, 2023.

[5] Vivek S Borkar. Stochastic approximation with two time scales. *Systems & Control Letters*, 29(5):291–294, 1997.

[6] Jin Chen, Zheng Liu, Xu Huang, Chenwang Wu, Qi Liu, Gangwei Jiang, Yuanhao Pu, Yuxuan Lei, Xiaolong Chen, Xingmei Wang, et al. When large language models meet personalization: Perspectives of challenges and opportunities. *arXiv preprint arXiv:2307.16376*, 2023.

[7] Ching-An Cheng, Tengyang Xie, Nan Jiang, and Alekh Agarwal. Adversarially trained actor critic for offline reinforcement learning. In *Proceedings of the 39th International Conference on Machine Learning*, 2022.

[8] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014.

[9] Yashar Deldjoo, Zhankui He, Julian McAuley, Anton Korikov, Scott Sanner, Arnau Ramisa, René Vidal, Maheswaran Sathiamoorthy, Atoosa Kasirzadeh, and Silvia Milano. A review of modern recommender systems using generative models (gen-recsys). In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '24, page 6448–6458, New York, NY, USA, 2024. Association for Computing Machinery.

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[11] Wenqi Fan, Zihuai Zhao, Jiatong Li, Yunqing Liu, Xiaowei Mei, Yiqi Wang, Jiliang Tang, and Qing Li. Recommender systems in the era of large language models (llms). *arXiv preprint arXiv:2307.02046*, 2023.

[12] Scott Fujimoto, Herke Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International Conference on Machine Learning*, pages 1587–1596. PMLR, 2018.

[13] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. PMLR, 2018.

[14] F. Maxwell Harper and Joseph A. Konstan. The movielens datasets: History and context. *ACM Trans. Interact. Intell. Syst.*, 5(4), 2015.

[15] Zhankui He, Zhouhang Xie, Rahul Jha, Harald Steck, Dawen Liang, Yesu Feng, Bodhisattwa Prasad Majumder, Nathan Kallus, and Julian Mcauley. Large language models as zero-shot conversational recommenders. CIKM '23, 2023.

[16] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939*, 2015.

[17] Yupeng Hou, Shanlei Mu, Wayne Xin Zhao, Yaliang Li, Bolin Ding, and Ji-Rong Wen. Towards universal sequence representation learning for recommender systems. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 585–593, 2022.

[18] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

[19] Natasha Jaques, Judy Hanwen Shen, Asma Ghandeharioun, Craig Ferguson, Agata Lapedriza, Noah Jones, Shixiang Gu, and Rosalind Picard. Human-centric dialog training via offline reinforcement learning. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3985–4003, Online, November 2020. Association for Computational Linguistics.

[20] Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems (TOIS)*, 2002.

[21] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations*, 2015.

[22] Jianghao Lin, Xinyi Dai, Yunjia Xi, Weiwen Liu, Bo Chen, Xiangyang Li, Chenxu Zhu, Huifeng Guo, Yong Yu, Ruiming Tang, et al. How can recommender systems benefit from large language models: A survey. *arXiv preprint arXiv:2306.05817*, 2023.

[23] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35, 2023.

[24] Yong Liu, Zhiqi Shen, Yinan Zhang, and Lizhen Cui. Diversity-promoting deep reinforcement learning for interactive recommendation. ICCSE '21, page 132–139, 2022.

[25] Hamid Reza Maei, Csaba Szepesvari, Shalabh Bhatnagar, Doina Precup, David Silver, and Richard S Sutton. Convergent temporal-difference learning with arbitrary smooth function approximation. In *NIPS*, pages 1204–1212, 2009.

[26] Bonan Min, Hayley Ross, Elior Sulem, Amir Pouran Ben Veyseh, Thien Huu Nguyen, Oscar Sainz, Eneko Agirre, Ilana Heintz, and Dan Roth. Recent advances in natural language processing via large pre-trained language models: A survey. *ACM Computing Surveys*, 56(2):1–40, 2023.

[27] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

[28] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint ArXiv:1707.06347*, 2017.

[29] Dusan Stamenkovic, Alexandros Karatzoglou, Ioannis Arapakis, Xin Xin, and Kleomenis Katevas. Choosing the best of both worlds: Diverse and novel recommendations through multi-objective reinforcement learning. WSDM, 2022.

[30] Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 3008–3021. Curran Associates, Inc., 2020.

[31] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[32] Jie Wang, Alexandros Karatzoglou, Ioannis Arapakis, and Joemon M. Jose. Reinforcement learning-based recommender systems with large language models for state reward and action modeling. SIGIR '24, page 375–385, 2024.

[33] Jie Wang, Fajie Yuan, Mingyue Cheng, Joemon M Jose, Chenyun Yu, Beibei Kong, Xiangnan He, Zhijin Wang, Bo Hu, and Zang Li. Transrec: Learning transferable recommendation from mixture-of-modality feedback. *arXiv preprint arXiv:2206.06190*, 2022.

[34] Yifan Wu, George Tucker, and Ofir Nachum. Behavior regularized offline reinforcement learning. *arXiv preprint ArXiv:1911.11361*, 2019.

[35] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 2023.

[36] Guanjie Zheng, Fuzheng Zhang, Zihan Zheng, Yang Xiang, Nicholas Jing Yuan, Xing Xie, and Zhenhui Li. Drn: A deep reinforcement learning framework for news recommendation. In *Proceedings of the World Wide Web Conference*, 2018.