Enhanced Whole Page Optimization via Mixed-Grained Reward Mechanism-Adapted Language Models

Anonymous ACL submission

Abstract

Optimizing the presentation of search and recommendation results is crucial to enhancing user experience and engagement. Whole Page Optimization (WPO) plays a pivotal role in this process, as it directly influences how information is surfaced to users. While Pre-trained Large Language Models (LLMs) have demonstrated remarkable capabilities in generating coherent and contextually relevant content, fine-tuning these models for complex tasks like WPO presents challenges. Specifically, the need for extensive human-annotated data to mitigate issues such as hallucinations and model instability can be prohibitively expensive, especially in large-scale systems that interact with millions of items daily. In this work, we address the challenge of fine-tuning LLMs for WPO by using user feedback as the supervision. Unlike manually labeled datasets, user feedback is inherently noisy and less precise. To overcome this, we propose a reward-based fine-tuning approach, PageLLM, which employs a mixed-grained reward mechanism that combines page-level and item-level rewards. The page-level reward evaluates the overall quality and coherence, while the item-level reward focuses on the accuracy and relevance of key recommendations. This dual-reward structure ensures that both the holistic presentation and the critical individual components are optimized. We validate PageLLM on both public and industrial datasets. PageLLM outperforms baselines and achieves a 0.44% GMV increase in an online A/B test with over 10 million users, demonstrating its real-world impact. The codes and data are available at this link.

1 Introduction

In the digital age, the presentation of search and recommendation results plays a pivotal role in shaping user experience and engagement (Wu et al., 2022; Bai et al., 2023). With the explosive growth of online information, Whole Page Optimization (WPO) has emerged as a critical task, aiming to surface the most relevant and diverse content in a cohesive and user-friendly manner (Wang et al., 2016; Ding et al., 2019). Recent advancements in Pre-trained Large Language Models (LLMs) have demonstrated remarkable capabilities in generating coherent and contextually relevant content (Zhao et al., 2023), offering a promising solution for addressing the challenges of WPO. However, applying these models to web-scale WPO tasks introduces significant complexities, particularly in balancing relevance, diversity, and the rank of items.

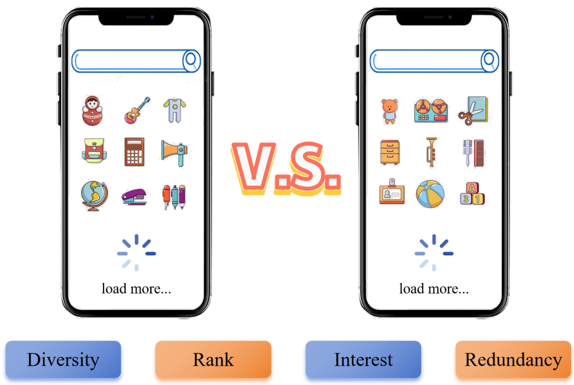

Figure 1: Which is better? Different ranking strategies lead to varying outcomes in diversity, interest alignment, redundancy, and ranking quality.

This research focuses on solving the web-scale WPO problem by leveraging the power of pre-trained LLMs to generate comprehensive and user-centric page presentations. Our goal is to optimize page layouts by considering multiple factors, including ranking (to ensure the most relevant items are prioritized), relevance (to align content with user intent), and diversity (to provide a rich and varied set of information). By achieving this, we aim to create a seamless and efficient user experience in search and recommendation scenarios, as illustrated in **Figure 1**.

Despite the potential of LLMs, applying them to WPO presents several challenges. First, fine-tuning these models for complex tasks typically requires extensive human-annotated data, which is costly and impractical for large-scale systems that interact with millions of items daily (Hadi et al., 2023). The lack of sufficient annotated data often leads to issues such as model hallucinations (generating factually inconsistent content) and instability (Zhang et al., 2023b). Additionally, user feedback, while abundant, is inherently noisy and less precise than manually labeled data, making traditional supervised fine-tuning methods difficult to apply directly.

Second, existing approaches often fail to account for the critical role of key items in determining overall page quality. For instance, in e-commerce, product images, and pricing information are pivotal in influencing user decisions. However, current page-level evaluation methods primarily focus on syntactic and semantic coherence, neglecting the impact of these key elements on user satisfaction. This oversight can result in suboptimal page presentations that fail to meet user expectations.

Our unique perspective is to leverage user feedback for RLHF fine-tuning and a mixed-grained reward mechanism to fine-tune pre-trained LLMs, optimizing both overall page coherence and key item effectiveness for web-scale WPO.

To address these challenges, we propose a reward-based fine-tuning framework that leverages user feedback to optimize pre-trained LLMs. Unlike traditional supervised methods, our approach constructs a golden item list for each user based on feedback (e.g., review scores), considering factors such as ranking, diversity, and redundancy. We then generate non-preferred lists that are inferior to the golden list in these aspects. Using these list pairs, we train a reward model and optimize the LLM through Reinforcement Learning from Human Feedback (RLHF). This method allows us to effectively utilize noisy user feedback, making the model more aligned with real-world user needs.

Furthermore, we introduce a mixed-grained reward mechanism that combines page-level and item-level (Xu et al., 2024a) rewards. The page-level reward evaluates the overall coherence and quality of the page, ensuring a smooth and logically consistent presentation. On the other hand, the item-level reward focuses on the accuracy and relevance of key recommendations, ensuring that critical elements are appropriately emphasized. This dual-reward structure enables a more nuanced optimization of WPO, balancing holistic page quality with the individual recommendation effectiveness.

We evaluated PageLLM on public and industrial datasets. In Amazon Review data sets, it surpasses baselines in key recommendation metrics and performs better in ranked, diversity, and redundancy metrics. In an industrial A/B test with more than 10 million users, it improves GMV by 0.44% and improves user engagement, proving its effectiveness in large-scale applications.

In summary, our main contributions are: **Reward-based fine-tuning framework.** We use user feedback to optimize pre-trained LLMs for WPO, addressing limitations of traditional supervised methods. **Mixed-grained reward mechanism.** We combine page-level and item-level rewards to enable more comprehensive and accurate page optimization. **Extensive evaluation and practical impact.** We demonstrate improvements in user engagement and satisfaction through A/B tests, providing a scalable and user-centric solution for WPO.

2 Related Work

Large Language Models (LLMs) have shown strong capabilities in NLP (Brown et al., 2020; Devlin et al., 2019) and have been applied in domains such as healthcare (Li et al., 2024; Wang et al., 2024a), education (Wang et al., 2022), creative writing (Franceschelli and Musolesi, 2024), and finance (Li et al., 2023a).

In recommender systems (Bai et al., 2024b,a; Cai et al., 2024; He et al., 2024), the integration of generative LLMs has enabled new modeling strategies. LlamaRec (Yue et al., 2023) and RecMind (Wang et al., 2024c) adopt sequential decision-making and self-inspiring algorithms for personalization. RecRec (Verma et al., 2023) and P5 (Geng et al., 2022) propose optimization-based and unified frameworks for diverse recommendation tasks. DOKE (Yao et al., 2023) incorporates domain-specific knowledge, while RLMRec (Ren et al., 2024) enhances graph-based modeling. RARS (Di Palma, 2023) combines retrieval and generative modules for sparse scenarios.

Prompt engineering techniques such as re-prompting and instruction tuning have improved LLM-based recommendations. ProLLM4Rec (Xu et al., 2024b) emphasizes model selection and prompt tuning. M6-REC (Cui et al., 2022) and PBNR (Li et al., 2023b) use personalized prompts to boost engagement and relevance.

Fine-tuning LLMs for recommendation has also gained traction. TALLRec (Bao et al., 2023) introduces dual-stage tuning for task-specific alignment. Flan-T5 (Kang et al., 2023) and InstructRec (Zhang et al., 2023a) demonstrate instruction-tuned effectiveness. RecLLM (Friedman et al., 2023) leverages conversational data, while DEALRec (Lin et al., 2024) applies pruning to improve efficiency. Further integration with user-item interaction modeling is explored in (Wang et al., 2024b). Recent advancements include Generative Recommenders (GRs) (Zhai et al., 2024), which reformulate recommendation tasks as sequential transduction problems using architectures like HSTU for large-scale, high-cardinality data.

Our work is also inspired by layout optimization for heterogeneous data (Gong et al., 2013) and multimedia-aware recommendation (Yi et al., 2022), both of which align with the WPO setting.

3 Problem Definition

Whole Page Optimization (WPO) in e-commerce search and recommendation aims to generate a ranked list of products that maximizes user satisfaction by optimizing presentation factors such as relevance, diversity, and redundancy.

We formulate WPO as a sequence generation task, where a pre-trained language model f_θ, parameterized by θ, generates an optimal product list $\pi' = f_\theta(q)$ for a given user query q, which encodes the user's historical item interactions. To align outputs with user preferences, we incorporate multi-grained supervision at both the sentence and token levels. Let q be a query and $\mathcal{I} = \{i_1, i_2, \ldots, i_N\}$ the set of available products. A product list is an ordered sequence $\pi = [i_{\pi_1}, i_{\pi_2}, \ldots, i_{\pi_K}]$, where K is the number of displayed items and $i_{\pi_k} \in \mathcal{I}$. The generation objective is:

$$\pi' = f_\theta(q) \tag{1}$$

The dataset $\mathcal{D}$ contains tuples $(q, \pi, y, \mathcal{F})$, where $y \in \{0, 1\}$ is a coarse-grained feedback for the list π, and $\mathcal{F}$ is a set of fine-grained feedback signals representing beneficial positional adjustments.

4 Dataset Generation

We construct a multi-granular dataset based on the Amazon Review corpus, designed to support both coarse-grained (page-level) and fine-grained (token-level) supervision signals for training. Each data instance includes a user query, a ground-truth ranked item list, and auxiliary feedback signals derived from user interactions. To facilitate reward modeling, we generate several types of paired item lists that reflect distinct optimization aspects—overall preference, ranking consistency, diversity, and redundancy. These pairs enable the construction of a unified reward function for reinforcement learning. Full details of dataset construction and supervision signal generation are provided in **Appendix A**.

5 PageLLM

Our framework (**Figure 2**) has three components: (1) supervised fine-tuning, (2) multi-grained reward modeling, and (3) policy optimization. These components work together to fine-tune a pre-trained LLM for WPO in recommender systems.

5.1 Supervised Fine-Tuning

To adapt the LLM for the recommendation task, we first perform supervised fine-tuning using a combination of user/item tokenization, meta-information pre-training, and ground truth fine-tuning.

5.1.1 User/Item Token

To enable the LLM to understand specific users and items, we create unique tokens to represent them (e.g. $user\_i$). These tokens are embedded into latent representation vectors, allowing the model to capture user preferences and item characteristics effectively. They are denoted as $\mathbf{e}_u = \text{Embedding}(u)$ for a user u and $\mathbf{e}_i = \text{Embedding}(i)$ for an item i.

5.1.2 Meta Information Pre-training

To shift the LLM's focus toward the WPO task, we pre-train the model using meta-information about users and items. This includes user profiles, item descriptions, and historical interactions. The prompt used in pre-training is shown in **Appendix B**. We design two pre-training tasks:

(1) rating prediction: The LLM predicts the user's rating for an item based on review text. The loss function is defined as:

$$\mathcal{L}_{\text{rating}} = \frac{1}{N} \sum_{(u,i,r)\in\mathcal{D}_{\text{rating}}} (r - f_\theta(u,i))^2, \tag{2}$$

where r is the ground truth, and $f_\theta(u,i)$ is the predicted rating.

(2) next token prediction: The LLM predicts the next token in the meta information prompts

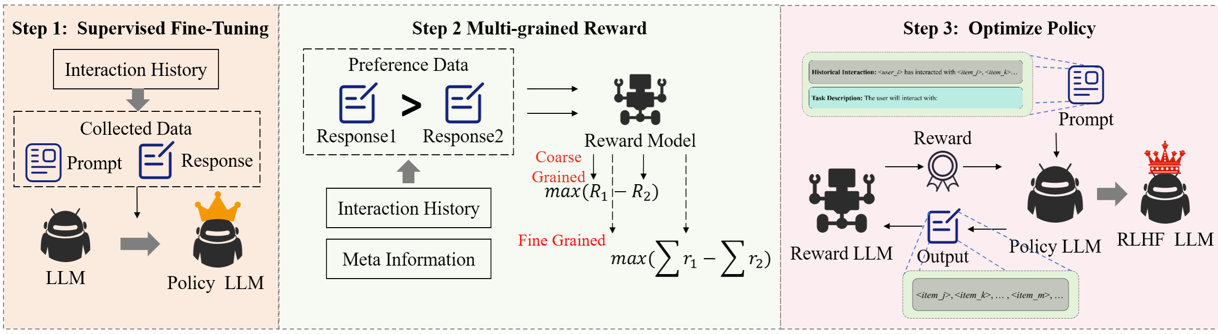

Figure 2: Overview of PageLLM. The framework incorporates mixed-grained rewards, combining both coarse-grained (page-level) and fine-grained (token-level) optimization.

of user/item background and interactions. The loss function is:

$$\mathcal{L}_{\text{next}} = -\sum_{t=1}^{T} \log p(w_t \mid w_{<t}; \theta), \tag{3}$$

where w_t is the t-th token in the sequence.

5.1.3 Ground Truth Fine-tuning

Then, the focus is shifted to recommendations. The LLM will predict a list of items. Here, we employ the ground truth dataset for Fine-tuning. After pre-training, we fine-tune the LLM using a ground truth dataset of user-item interactions. The prompt used in fine-tuning is also shown in **Appendix B**. The model is trained to generate a ranked list of items $\pi = [i_{\pi_1}, i_{\pi_2}, \ldots, i_{\pi_K}]$ for a given user u. The loss function is:

$$\mathcal{L}_{\text{rank}} = -\sum_{(u,\pi)\in\mathcal{D}_{\text{rank}}} \log p(\pi \mid u; \theta), \tag{4}$$

where π is the ground truth ranking.

5.2 Multi-grained Reward Function

To further optimize the LLM, we design a multi-grained reward function that provides both coarse-grained (page-level) and fine-grained (token-level) feedback. We use the preference pairs for RLHF training in **Section 4**. The training objective is to maximize $L = R(y_p) - R(y_l)$.

5.2.1 Coarse-Grained Reward

The coarse-grained reward evaluates the overall quality of the generated sequence π':

$$R_c(\pi') = g(\pi', u) \tag{5}$$

Here, $g(\pi', q)$ measures the alignment between the generated sequence π' with the user u.

5.2.2 Fine-Grained Reward

The fine-grained reward provides token-level supervision, which enables the model to learn from granular feedback and monitor the tiny differences. The generation process is formulated as a Markov Decision Process (MDP) with the tuple $\langle S, A, R, P, \gamma \rangle$:

- S: State space, with the initial state s_1 representing the input query q.
- A: Action space, where each action a_t corresponds to a token generated at time step t.
- R: Reward function, assigning a reward $r_t = r_\phi(s_t, a_t)$ to each token a_t in state s_t.
- P: State transition, defining the transition from s_t to s_{t+1} after generating the token a_t.
- γ: Discount factor, typically set to $\gamma = 1$ for this task.

The reward for the entire sequence $\pi' = \{a_1, a_2, \ldots, a_T\}$ is computed as the average of the token-level rewards:

$$R(\pi') = \frac{1}{T}\sum_{t=1}^{T} r_t \tag{6}$$

where T is the length of the sequence.

To train the token-level reward model, we utilize a loss function inspired by the Bradley-Terry model for preference modeling. Given two sequences π_i and π_j generated for the same query q, the preference probability is defined as:

$$p(\pi_i \succ \pi_j) = \sigma(R(\pi_i) - R(\pi_j)) \tag{7}$$

where σ is the sigmoid function. The loss function is then defined as the negative log-likelihood of the observed preferences:

$$L = -\mathbb{E}_{(\pi_i,\pi_j)\sim D}\left[\log\sigma\left(\frac{1}{T_i}\sum_{t=1}^{T_i} r_t^{(i)} - \frac{1}{T_j}\sum_{t=1}^{T_j} r_t^{(j)}\right)\right] \tag{8}$$

Here:

- D: Dataset of sequence pairs with preference annotations.
- T_i, T_j: Lengths of the sequences π_i and π_j, respectively.
- $r_t^{(i)}, r_t^{(j)}$: Token-level rewards for the t-th token in π_i and π_j.

This fine-grained reward framework provides precise token-level feedback, improving the alignment of generated sequences with user preferences.

5.3 RL from User Feedback

To further refine the model, we employ Reinforcement Learning from Human Feedback (RLHF). The objective is to maximize the expected cumulative reward:

$$J(\theta) = \mathbb{E}\pi' \sim f_\theta(u) \left[R(\pi') \right], \tag{9}$$

where $R(\pi')$ is the multi-grained reward function. We use the Proximal Policy Optimization (PPO) algorithm to update the model parameters:

$$\theta \leftarrow \theta + \eta \nabla_\theta J(\theta), \tag{10}$$

where η is the learning rate. The policy gradient is computed as:

$$\nabla_\theta J(\theta) = \mathbb{E}_{\pi' \sim f_\theta(u)} \left[\nabla_\theta \log f_\theta(\pi' \mid q) \cdot R(\pi') \right]. \tag{11}$$

5.4 Deployment

In the deployment phase, the fine-tuned LLM is integrated into the e-commerce platform to generate real-time recommendations. Given a user u, the model generates a ranked list of items π' as:

$$\pi' = \arg\max_\pi f_\theta(\pi \mid u). \tag{12}$$

To ensure scalability and efficiency, we deploy the model using a distributed inference framework, which partitions the computation across multiple GPUs. The inference latency is optimized by caching frequently accessed user/item embeddings and pre-computing meta-information.

6 Experiments

We evaluate PageLLM to answer the following research questions through a series of main and supplementary experiments:

- **RQ1**: What is the impact of PageLLM on the performance of whole-page optimization?
- **RQ2**: Does RLHF negatively affect recommendation quality?
- **RQ3**: Can the overall quality of the recommendations be positively evaluated using a comprehensive metric?
- **RQ4**: How do different LLMs influence the overall outcomes?
- **RQ5**: Can PageLLM perform well in industrial applications?

We also conduct cold-start studies (**Appendix F**), ablation studies (**Appendix G**), and the case study (**Appendix H**) to provide deeper insights.

6.1 Experiment Setup

We evaluate our method on seven categories of the Amazon Review dataset (McAuley and Yang, 2016). The parameters and the implementation of supervised fine-tuning and PPO RLHF are detailed in **Appendix D**. We implement our method using GPT-2 as the backbone model and run all experiments on the GPU server.

6.2 Main Results (RQ1)

Table 1 presents a comparative analysis of PageLLM with and without the reward mechanism on multiple datasets. It indicate that incorporating reinforcement learning with user feedback significantly improves the performance of recommendations. The metrics used in the experiments are detailed in **Appendix C**.

First, in terms of recommendation accuracy, PageLLM outperforms the baseline model across all datasets. Metrics such as Recall@20, Recall@40, and NDCG@100 show noticeable improvements, demonstrating that the reward mechanism effectively refines recommendation relevance. The most substantial gains in NDCG@100 are observed in the AM-Luxury, AM-Sports, and AM-Beauty datasets, with increases of 46.8%, 46.7%, and 40.8%, respectively. These findings highlight that RLHF optimizes the alignment between user preferences and recommended items, particularly in domains with more complex preference structures.

The ranked metrics (WAS, PWKT, WMRD, and DPA) remain largely stable across datasets. PageLLM achieves slight improvements in WAS and DPA, indicating better ranking alignment and accuracy, while PWKT and WMRD exhibit minimal changes, preserving ranking consistency and reducing ranking errors. These results suggest that RLHF enhances recommendation quality without disrupting the ranking structure or introducing bias.

In addition, both diversity and redundancy are improved. The ILD metric, which measures intra-

Table 1: Performance comparison of PageLLM with and without reward mechanisms across multiple datasets. The table evaluates recommendation accuracy, ranking quality, diversity, and redundancy.

Dataset	Model	Recommendation			Ranked				Diversity	Redundancy
		Recall@20 ↑	**Recall@40 ↑**	**NDCG@100 ↑**	**WAS ↑**	**PWKT ↑**	**WMRD ↓**	**DPA ↑**	**ILD ↑**	**Entropy ↑**
AM-Instruments	PageLLM	0.1698	0.2265	0.1919	0.0168	0.0003	0.0004	0.0165	-	-
	w/o Reward	0.1605	0.2097	0.1315	0.0157	0.0001	0.0007	0.0155	-	-
AM-Sports	PageLLM	0.0768	0.1283	0.0726	0.0156	0.0001	0.0003	0.0155	0.0418	0.0528
	w/o Reward	0.0722	0.1086	0.0495	0.0146	0.0000	0.0004	0.0132	0.0394	0.0498
AM-Luxury	PageLLM	0.3087	0.3445	0.3323	0.0160	0.0001	0.0005	0.0157	-	-
	w/o Reward	0.2910	0.3244	0.2263	0.0149	0.0001	0.0006	0.0137	-	-
AM-Beauty	PageLLM	0.1590	0.2177	0.1313	0.0156	0.0002	0.0006	0.0154	0.0412	0.0514
	w/o Reward	0.1435	0.1995	0.0932	0.0115	0.0001	0.0008	0.0103	0.0371	0.0463
AM-Food	PageLLM	0.1441	0.1677	0.1125	0.0165	0.0001	0.0004	0.0154	-	-
	w/o Reward	0.1398	0.1627	0.1019	0.0156	0.0000	0.0007	0.0146	-	-
AM-Scientific	PageLLM	0.1484	0.1908	0.1480	0.0157	0.0000	0.0001	0.0157	-	-
	w/o Reward	0.1468	0.1898	0.1071	0.0147	0.0000	0.0001	0.0145	-	-
AM-Toys	PageLLM	0.1349	0.1873	0.0971	0.0157	0.0001	0.0005	0.0155	0.0358	0.0482
	w/o Reward	0.1178	0.1781	0.0754	0.0147	0.0000	0.0006	0.0139	0.0355	0.0477

list diversity, increases in datasets, indicating a broader range of recommended items. Similarly, the Entropy metric, reflecting category balance, shows notable gains, reducing redundancy and promoting a more even category distribution. These improvements demonstrate that RLHF enhances recommendation diversity while maintaining ranking stability, contributing to more balanced and effective recommendations.

Analyzing performance across different datasets, it is evident that the impact of RLHF varies depending on the domain. The AM-Luxury and AM-Instruments datasets show the most substantial improvements, likely due to their nuanced user preferences. Meanwhile, datasets such as AM-Food and AM-Scientific exhibit smaller but consistent improvements, suggesting that the effect of RLHF is more pronounced in domains with inherently complex recommendation patterns. AM-Toys and AM-Sports also demonstrate moderate increases, indicating that reinforcement learning helps refine recommendations even in broader-interest categories.

Overall, these results confirm that RLHF contributes positively to recommendation quality, particularly in terms of accuracy and relevance, without significantly affecting ranking stability. Future work could explore how RLHF influences diversity and redundancy to provide a more holistic evaluation of whole-page optimization.

6.3 Recommendation Study (RQ2)

To investigate whether RLHF negatively impacts recommendation performance, we compare PageLLM with several baseline models (details in **Appendix E**). The results presented in **Table 2** demonstrate that PageLLM consistently achieves the highest performance across various datasets and metrics, indicating that RLHF does not degrade recommendation quality but rather enhances it.

Across most datasets, PageLLM achieves the highest Recall@20, Recall@40, and NDCG@100 scores. Notably, in the AM-Instruments dataset, PageLLM attains a NDCG@100 of 0.1919, significantly outperforming the second-best model (FDSA, 0.1080). Similarly, in the AM-Luxury dataset, PageLLM reaches an NDCG@100 of 0.3323, surpassing the best-performing baseline (FDSA, 0.2107) by a substantial margin. These results suggest that RLHF not only maintains but also improves the overall ranking quality of recommended items. This improvement in NDCG can be attributed to the reward mechanism aligning recommendations more closely with user preferences, ensuring that highly relevant items appear in top-ranked positions.

Further examining Recall@20 and Recall@40, PageLLM exhibits strong performance improvements. In AM-Sports, PageLLM achieves a Recall@40 of 0.1283, outperforming the best baseline (MD-CVAE, 0.1180). Likewise, in AM-Beauty, PageLLM attains a Recall@20 of 0.1590, surpassing the second-best baseline (SASRec, 0.1503). These consistent improvements across different datasets indicate that RLHF effectively optimizes recommendation relevance without introducing adverse effects.

Analyzing dataset-specific trends, PageLLM demonstrates the most significant advantage in AM-Luxury and AM-Instruments, likely due to the nuanced and highly personalized nature of user pref-

Table 2: Performance comparison of PageLLM and baseline models on the Amazon Review Dataset. The table reports Recall@20, Recall@40, and NDCG@100 across multiple domains, evaluating the effectiveness of different recommendation models.

Dataset	Metric	Multi-VAE	MD-CVAE	LightGCN	BERT4Rec	S^3Rec	UniSRec	FDSA	SASRec	GRU4Rec	RecMind	HSTU	PageLLM
AM-Instruments	Recall@20	0.1096	0.1398	0.1195	0.1183	0.1352	0.1684	0.1382	0.1483	0.1271	0.1315	0.1149	**0.1698**
	Recall@40	0.1628	0.1743	0.1575	0.1531	0.1767	0.2239	0.1787	0.1935	0.1660	0.1930	0.1428	**0.2265**
	NDCG@100	0.0735	0.1040	0.0985	0.0922	0.0894	0.1075	0.1080	0.0934	0.0998	0.1201	0.1083	**0.1919**
AM-Sports	Recall@20	0.0659	0.0714	0.0677	0.0521	0.0616	0.0714	0.0681	0.0541	0.0720	0.0614	0.0713	**0.0768**
	Recall@40	0.0975	0.1180	0.0973	0.0701	0.0813	0.1143	0.0866	0.0739	0.1086	0.1044	0.1094	**0.1283**
	NDCG@100	0.0446	0.0514	0.0475	0.0305	0.0438	0.0504	0.0475	0.0361	0.0498	0.0389	0.0238	**0.0726**
AM-Luxury	Recall@20	0.2306	0.2771	0.2514	0.2076	0.2241	**0.3091**	0.2759	0.2550	0.2126	0.2215	0.1879	0.3087
	Recall@40	0.2724	0.3206	0.3004	0.2404	0.2672	**0.3675**	0.3176	0.3008	0.2522	0.2898	0.2145	0.3445
	NDCG@100	0.1697	0.2064	0.1947	0.1617	0.1542	0.2010	0.2107	0.1965	0.1623	0.2017	0.1773	**0.3323**
AM-Beauty	Recall@20	0.1295	0.1472	0.1429	0.1126	0.1354	0.1462	0.1447	0.1503	0.0997	0.1445	0.0925	**0.1590**
	Recall@40	0.1720	0.2058	0.1967	0.1677	0.1789	0.1898	0.1875	0.2018	0.1528	0.1863	0.1137	**0.2177**
	NDCG@100	0.0835	0.0871	0.0890	0.0781	0.0867	0.0907	0.0834	0.0929	0.0749	0.0847	0.0633	**0.1313**
AM-Food	Recall@20	0.1062	0.1170	0.1149	0.1036	0.1157	0.1423	0.1099	0.1171	0.1140	0.0936	0.949	**0.1441**
	Recall@40	0.1317	0.1431	0.1385	0.1284	0.1456	0.1661	0.1317	0.1404	0.1389	0.1107	0.1218	**0.1677**
	NDCG@100	0.0727	0.0863	0.0853	0.0835	0.0926	0.1024	0.0904	0.0942	0.0910	0.0777	0.0672	**0.1125**
AM-Scientific	Recall@20	0.1069	0.1389	0.1385	0.0871	0.1089	**0.1492**	0.1188	0.1298	0.0849	0.0924	0.1089	0.1484
	Recall@40	0.1483	0.1842	0.1857	0.1160	0.1541	**0.1954**	0.1547	0.1776	0.1204	0.1246	0.1545	0.1908
	NDCG@100	0.0766	0.0872	0.0834	0.0606	0.0715	0.1056	0.0846	0.0864	0.0594	0.0749	0.0977	**0.1480**
AM-Toys	Recall@20	0.1076	0.1107	0.1096	0.0853	0.1064	0.1110	0.0972	0.0869	0.0657	0.1126	0.0986	**0.1349**
	Recall@40	0.1558	0.1678	0.1558	0.1375	0.1524	0.1457	0.1268	0.1146	0.0917	0.1564	0.1407	**0.1873**
	NDCG@100	0.0781	0.0812	0.0775	0.0532	0.0665	0.0638	0.0662	0.0525	0.0439	0.0584	0.0358	**0.0971**

erences in these domains. In contrast, for datasets such as AM-Toys and AM-Scientific, the performance gap between PageLLM and the baselines is narrower, suggesting that in more structured or less complex preference spaces, traditional methods still perform reasonably well. However, PageLLM remains the top performer, reinforcing the robustness of RLHF-based optimization.

Overall, the results indicate that RLHF does not negatively impact recommendation performance; instead, it enhances the accuracy and quality of recommendations across diverse datasets. By leveraging reinforcement learning to refine preference modeling, PageLLM achieves superior performance compared to state-of-the-art baselines, validating the effectiveness of RLHF in whole-page recommendation tasks.

6.4 LLM Judgement (RQ3)

To evaluate the overall quality of recommendations, we conduct a comparative analysis using Large Language Model (LLM) judgment based on the win rate metric. The win rate represents the proportion of cases where PageLLM-generated recommendations are preferred over the baseline model recommendations.

From **Figure 3**, it is evident that PageLLM achieves a significantly higher win rate compared to the baseline, as indicated by the dominant red bar in the visualization. The preference for PageLLM suggests that its recommendations align better with human evaluators' expectations in terms of relevance, diversity, and overall quality. Although a small fraction of cases favors the baseline (represented by the blue section), the overwhelming preference for PageLLM confirms the effectiveness of reinforcement learning with human feedback (RLHF) in refining recommendation quality.

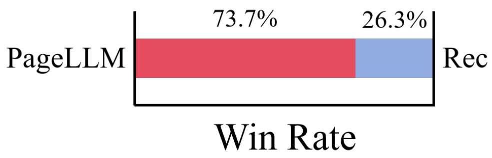

Figure 3: LLM-based preference judgment between PageLLM and baseline.

This result aligns with the findings from previous sections, where PageLLM demonstrated superior performance across multiple datasets and evaluation metrics. The LLM-based assessment further reinforces the claim that PageLLM enhances recommendation effectiveness, making it a more suitable model for whole-page optimization.

6.5 Base LLM Study (RQ4)

In our study, we initially implemented PageLLM using GPT-2 as the backbone model. However, given the rapid advancements in open-sourced LLMs, we want to identify the most suitable LLM backbone that achieves an optimal balance between performance and cost. To this end, we explored Llama 3.2, the latest model in the Llama family, which employs knowledge distillation techniques to deliver competitive performance with fewer parameters. Specifically, we implemented PageLLM using the Llama3.2-1B model and conducted a comprehensive comparison of performance and cost metrics, including training time and memory usage, against the GPT-2 backbone. The detailed results are presented in **Table 3**.

Table 3: GPT-2 vs. Llama3.2-1B: performance (Recall), training time, and memory usage (GPU).

Category	Metric	GPT-2	Llama3.2-1B
Performance	Recall@20	0.1698	0.1757
	Recall@40	0.2265	0.2414
Time	Pre-training	3h 22m 57s	73h 24m 46s
	Fine-tuning	18 s/epoch	1m 58s/epoch
Memory	GPU	8.94 GB (Full)	15.16 GB (LoRA)

From the results, it is evident that the Llama3.2-1B backbone, with its larger parameter size, delivers superior performance compared to GPT-2. However, this performance gain comes at a significant cost in terms of computational resources. The pre-training time for Llama3.2-1B is substantially higher. Similarly, fine-tuning Llama3.2-1B takes nearly 2 minutes per epoch, whereas GPT-2 completes an epoch in just 18 seconds. Moreover, the memory requirements for Llama3.2-1B are notably higher, even when employing Low-Rank Adaptation (LoRA) techniques to reduce memory usage.

In conclusion, while Llama3.2-1B demonstrates better performance, GPT-2 offers a more favorable performance-cost trade-off. GPT-2's significantly lower training time and memory requirements make it a more practical choice for scenarios where computational resources are constrained.

6.6 Industrial Dataset Experiment (RQ5)

In the online experiment, our aim is to evaluate how the proposed approach can improve search accuracy in the production environment. The online method utilizes the proposed approach to produce embeddings of listings which are appended as an additional feature to measure the WPO utility, which relieves the deployment requirement of GPU serving as it becomes model agnostic and can be fitted with traditional CPU serving in the process of online inference.

During the online test, we deploy the proposed algorithm globally to a commercial E-Commerce search engine as the treatment method and randomly assign 50% traffic to the treatment group. The online test has been running for over 1 week, and the total number of unique users is greater than 10 million. We focus on several key metrics:

- **GMV** refers to the total grand merchandise value.
- **CTR** refers to the average click-through rate of items exposed to both groups.
- **Avg Purchases** refers to the average number of purchases of users in both groups.
- **Session Failure Rate** refers to rate of sessions being abandoned by customers.
- **Session Purchase Rate** refers to the rate of sessions that end up with a customer purchase.

Table 4: Online A/B testing results with over 10 million unique users. We show the percentage of improvement on different metrics of the treatment group.

	GMV	CTR	Avg Purchases	Ses. Failure	Ses. Purchase
Treatment	↑ 0.44%**	↑ 0.14%**	↑ 1.01%**	↓ 0.08%	↑ 0.24%**

* ** indicates the statistical significance level of 0.01.

The online A/B testing results are shown in **Table 4**. All key metrics have been lifted in the treatment group, except that the Session Failure Rate was improved to be lower than the control group without being statistically significant. In particular, the key metric GMV has been significantly improved by 0.44% globally, while other up-funnel metrics such as average purchases and click-through rate have also been improved in a consistent manner, proving that the gain is trustworthy and it is not false positive.

7 Conclusion

Whole Page Optimization (WPO) is essential for improving user experience in search and recommendation systems, yet fine-tuning Large Language Models (LLMs) for this task is challenging due to costly annotations, model instability, and noisy user feedback. To address these issues, we propose PageLLM, a reward-based fine-tuning framework that leverages Reinforcement Learning from Human Feedback (RLHF) and a mixed-grained reward mechanism to optimize both page-level coherence and item-level relevance. By integrating real-world user feedback, PageLLM effectively enhances recommendation quality without relying on expensive human annotations. Extensive experiments on Amazon Review datasets and an industrial-scale A/B test with over 10 million users demonstrate its superiority over baselines, with a 0.44% increase in GMV and significant improvements in user engagement.

Limitations

While our approach demonstrates promising results across multiple datasets and settings, several limitations remain, which we outline here to guide future research directions: (1) Our evaluation is primarily conducted within single-domain settings, and the generalizability to cross-domain tasks has not been extensively explored. (2) The reward mechanism may be sensitive to noisy or implicit feedback signals, which can affect optimization quality. (3) The model assumes relatively stable user preferences and does not explicitly adapt to dynamic or rapidly changing behaviors. (4) While the proposed method shows robustness in cold-start simulations, further validation is needed for long-tail or rapidly evolving item pools. (5) Our current implementation focuses on textual inputs; incorporating multimodal signals such as images or structured metadata is a promising direction.

References

Haoyue Bai, Min Hou, Le Wu, Yonghui Yang, Kun Zhang, Richang Hong, and Meng Wang. 2023. Gorec: a generative cold-start recommendation framework. In *Proceedings of the 31st ACM international conference on multimedia*, pages 1004–1012.

Haoyue Bai, Min Hou, Le Wu, Yonghui Yang, Kun Zhang, Richang Hong, and Meng Wang. 2024a. Unified representation learning for discrete attribute enhanced completely cold-start recommendation. *IEEE Transactions on Big Data*.

Haoyue Bai, Le Wu, Min Hou, Miaomiao Cai, Zhuangzhuang He, Yuyang Zhou, Richang Hong, and Meng Wang. 2024b. Multimodality invariant learning for multimedia-based new item recommendation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 677–686.

Keqin Bao, Jizhi Zhang, Yang Zhang, Wenjie Wang, Fuli Feng, and Xiangnan He. 2023. Tallrec: An effective and efficient tuning framework to align large language model with recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems*, pages 1007–1014.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Miaomiao Cai, Min Hou, Lei Chen, Le Wu, Haoyue Bai, Yong Li, and Meng Wang. 2024. Mitigating recommendation biases via group-alignment and global-uniformity in representation learning. *ACM Transactions on Intelligent Systems and Technology*.

Zeyu Cui, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. 2022. M6-rec: Generative pretrained language models are open-ended recommender systems. *arXiv preprint arXiv:2205.08084*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Dario Di Palma. 2023. Retrieval-augmented recommender system: Enhancing recommender systems with large language models. In *Proceedings of the 17th ACM Conference on Recommender Systems*, pages 1369–1373.

Weicong Ding, Dinesh Govindaraj, and SVN Vishwanathan. 2019. Whole page optimization with global constraints. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3153–3161.

Giorgio Franceschelli and Mirco Musolesi. 2024. On the creativity of large language models. *AI & SOCIETY*, pages 1–11.

Luke Friedman, Sameer Ahuja, David Allen, Zhenning Tan, Hakim Sidahmed, Changbo Long, Jun Xie, Gabriel Schubiner, Ajay Patel, Harsh Lara, and 1 others. 2023. Leveraging large language models in conversational recommender systems. *arXiv preprint arXiv:2305.07961*.

Shijie Geng, Shuchang Liu, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. 2022. Recommendation as language processing (rlp): A unified pretrain, personalized prompt & predict paradigm (p5). In *Proceedings of the 16th ACM Conference on Recommender Systems*, pages 299–315.

Zhenhuan Gong and 1 others. 2013. Multi-level data layout optimization for heterogeneous access patterns.

Muhammad Usman Hadi, Rizwan Qureshi, Abbas Shah, Muhammad Irfan, Anas Zafar, Muhammad Bilal Shaikh, Naveed Akhtar, Jia Wu, Seyedali Mirjalili, and 1 others. 2023. A survey on large language models: Applications, challenges, limitations, and practical usage. *Authorea Preprints*.

Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 639–648.

Zhuangzhuang He, Yifan Wang, Yonghui Yang, Peijie Sun, Le Wu, Haoyue Bai, Jinqi Gong, Richang Hong, and Min Zhang. 2024. Double correction framework for denoising recommendation. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1062–1072.

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

Yupeng Hou, Shanlei Mu, Wayne Xin Zhao, Yaliang Li, Bolin Ding, and Ji-Rong Wen. 2022. 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.

Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*, pages 197–206. IEEE.

Wang-Cheng Kang, Jianmo Ni, Nikhil Mehta, Maheswaran Sathiamoorthy, Lichan Hong, Ed Chi, and Derek Zhiyuan Cheng. 2023. Do llms understand user preferences? evaluating llms on user rating prediction. *arXiv preprint arXiv:2305.06474*.

Haozhou Li, Qinke Peng, Xinyuan Wang, Xu Mou, and Yonghao Wang. 2023a. Sehf: A summary-enhanced hierarchical framework for financial report sentiment analysis. *IEEE Transactions on Computational Social Systems*.

Haozhou Li, Xinyuan Wang, Hongkai Du, Wentong Sun, and Qinke Peng. 2024. Sade: A speaker-aware dual encoding model based on diagbert for medical triage and pre-diagnosis. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 12712–12716. IEEE.

Xinyi Li, Yongfeng Zhang, and Edward C Malthouse. 2023b. Pbnr: Prompt-based news recommender system. *arXiv preprint arXiv:2304.07862*.

Dawen Liang, Rahul G Krishnan, Matthew D Hoffman, and Tony Jebara. 2018. Variational autoencoders for collaborative filtering. In *Proceedings of the 2018 world wide web conference*, pages 689–698.

Xinyu Lin, Wenjie Wang, Yongqi Li, Shuo Yang, Fuli Feng, Yinwei Wei, and Tat-Seng Chua. 2024. Data-efficient fine-tuning for llm-based recommendation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 365–374.

Julian McAuley and Alex Yang. 2016. Addressing complex and subjective product-related queries with customer reviews. In *Proceedings of the 25th International Conference on World Wide Web*, pages 625–635.

Xubin Ren, Wei Wei, Lianghao Xia, Lixin Su, Suqi Cheng, Junfeng Wang, Dawei Yin, and Chao Huang. 2024. Representation learning with large language models for recommendation. In *Proceedings of the ACM on Web Conference 2024*, pages 3464–3475.

Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pages 1441–1450.

Sahil Verma, Ashudeep Singh, Varich Boonsanong, John P Dickerson, and Chirag Shah. 2023. Recrec: Algorithmic recourse for recommender systems. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 4325–4329.

Xinyuan Wang, Haozhou Li, Dingfang Zheng, and Qinke Peng. 2024a. Lcmdc: Large-scale chinese medical dialogue corpora for automatic triage and medical consultation. *arXiv preprint arXiv:2410.03521*.

Xinyuan Wang, Qinke Peng, Xu Mou, Haozhou Li, and Ying Wang. 2022. A hierarchal bert structure for native speaker writing detection. In *2022 China Automation Congress (CAC)*, pages 3705–3710. IEEE.

Xinyuan Wang, Liang Wu, Liangjie Hong, Hao Liu, and Yanjie Fu. 2024b. Llm-enhanced user-item interactions: Leveraging edge information for optimized recommendations. *arXiv preprint arXiv:2402.09617*.

Yancheng Wang, Ziyan Jiang, Zheng Chen, Fan Yang, Yingxue Zhou, Eunah Cho, Xing Fan, Yanbin Lu, Xiaojiang Huang, and Yingzhen Yang. 2024c. RecMind: Large language model powered agent for recommendation. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 4351–4364, Mexico City, Mexico. Association for Computational Linguistics.

Yue Wang, Dawei Yin, Luo Jie, Pengyuan Wang, Makoto Yamada, Yi Chang, and Qiaozhu Mei. 2016. Beyond ranking: Optimizing whole-page presentation. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, pages 103–112.

Le Wu, Xiangnan He, Xiang Wang, Kun Zhang, and Meng Wang. 2022. A survey on accuracy-oriented neural recommendation: From collaborative filtering to information-rich recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 35(5):4425–4445.

Dehong Xu, Liang Qiu, Minseok Kim, Faisal Ladhak, and Jaeyoung Do. 2024a. Aligning large language models via fine-grained supervision. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 673–680, Bangkok, Thailand. Association for Computational Linguistics.

Lanling Xu, Junjie Zhang, Bingqian Li, Jinpeng Wang, Mingchen Cai, Wayne Xin Zhao, and Ji-Rong Wen. 2024b. Prompting large language models for recommender systems: A comprehensive framework and empirical analysis. *arXiv preprint arXiv:2401.04997*.

Jing Yao, Wei Xu, Jianxun Lian, Xiting Wang, Xiaoyuan Yi, and Xing Xie. 2023. Knowledge plugins: Enhancing large language models for domain-specific recommendations. *arXiv preprint arXiv:2311.10779*.

Gangman Yi, Donghoon Kim, and Neil Yen. 2022. Computational optimization and applications for heterogeneous multimedia data.

Zhenrui Yue, Sara Rabhi, Gabriel de Souza Pereira Moreira, Dong Wang, and Even Oldridge. 2023. Llamarec: Two-stage recommendation using large language models for ranking. *arXiv preprint arXiv:2311.02089*.

Jiaqi Zhai, Lucy Liao, Xing Liu, Yueming Wang, Rui Li, Xuan Cao, Leon Gao, Zhaojie Gong, Fangda Gu, Michael He, and 1 others. 2024. Actions speak louder than words: Trillion-parameter sequential transducers for generative recommendations. *arXiv preprint arXiv:2402.17152*.

Junjie Zhang, Ruobing Xie, Yupeng Hou, Wayne Xin Zhao, Leyu Lin, and Ji-Rong Wen. 2023a. Recommendation as instruction following: A large language model empowered recommendation approach. *arXiv preprint arXiv:2305.07001*.

Tingting Zhang, Pengpeng Zhao, Yanchi Liu, Victor S Sheng, Jiajie Xu, Deqing Wang, Guanfeng Liu, Xiaofang Zhou, and 1 others. 2019. Feature-level deeper self-attention network for sequential recommendation. In *IJCAI*, pages 4320–4326.

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, and 1 others. 2023b. Siren's song in the ai ocean: a survey on hallucination in large language models. *arXiv preprint arXiv:2309.01219*.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, and 1 others. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. 2020. S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization. In *Proceedings of the 29th ACM international conference on information & knowledge management*, pages 1893–1902.

Yaochen Zhu and Zhenzhong Chen. 2022. Mutually-regularized dual collaborative variational auto-encoder for recommendation systems. In *Proceedings of The ACM Web Conference 2022*, pages 2379–2387.

A Dataset Generation

For the WPO task, our goal is to train an LLM-based recommender system using user-item interaction data. This data will offer both page-level (coarse-grained) and token-level (fine-grained) supervision signals, which are crucial for the subsequent training and optimization of the model.

A.1 Dataset Construction

For the WPO task, we construct a dataset using the Amazon Review Dataset, providing both page-level (coarse-grained) and token-level (fine-grained) supervision signals. Let $\mathcal{U}$ and $\mathcal{I}$ denote the sets of users and items, respectively. For each user $u \in \mathcal{U}$, we construct an item list $\mathcal{I}_u$ as the target output.

Product rankings in e-commerce platforms are influenced by multiple factors, such as click-through rate (CTR) and conversion rate. When products have identical scores, their positions in the recommendation list π may be randomized, though minor positional shifts can significantly impact user engagement.

By analyzing these variations, we extract fine-grained feedback:

$$\mathcal{F} = \left\{(i, k, k') \mid \Delta E(i, k \to k') \neq 0\right\} \tag{13}$$

This dataset structure ensures PageLLM learns both high-level ranking preferences and subtle position-based optimizations.

A.2 Ground Truth

First, we collect the ground truth item lists, which are considered the optimal solutions. We take into account factors such as relationship, ranking, diversity, and redundancy when constructing these lists.

(1) User-Item Connection Graph: We generate a user-item connection table T, where each entry $(u, i, r_{ui}) \in T$, with $u \in \mathcal{U}$, $i \in \mathcal{I}$, and r_{ui} representing the rating score of user u for item i. **(2) Item Selection and Clustering**: We select items for which $r_{ui} > 3$ to represent a positive relationship. Using these scores, we cluster the items in the list $\mathcal{I}_u$ for each user u and rank them in descending order of the scores. Let $\mathcal{I}_u^+$ denote the set of selected items for user u. **(3) Input and Label Set Splitting**: We split the item list $\mathcal{I}_u$ into an input set $\mathcal{I}_u^{in}$ and a label set $\mathcal{I}_u^{out}$ based on coarse - grained ranking. We ensure that both the input and label sets contain different levels of rating scores.

(4) Fine-Grained Re-ranking and Redundancy Handling: For each user u in the label set $\mathcal{I}_u^{out}$, we re-rank the items within the same score group using fine-grained scores s_i. Let $\mathcal{U}_i$ be the set of users who have rated item i. Then, $s_i = \frac{1}{|\mathcal{U}_i|} \sum_{u \in \mathcal{U}_i} r_{ui}$. We also remove redundant items and prioritize diversity over fine-grained scores.

These ranked item lists are considered the ground truth for the WPO task, corresponding to the state with the lowest potential energy, which represents the best solution.

The input prompts include the user ID u and historical interactions (items with their corresponding scores), while the output is the item list $\mathcal{I}_u$ that takes into account relationship, ranking, diversity, and redundancy.

A.3 Paired Preference Data

A.3.1 Preference Pairs

Based on the ground truth item list $\mathcal{I}_u^{gt}$, we create preference pairs $(\mathcal{I}_u^{gt}, \mathcal{I}_u^{np})$ to evaluate the recommender quality. The preferred item list $\mathcal{I}_u^{gt}$ is the ground truth one, while the non-preferred item list $\mathcal{I}_u^{np}$ contains items with rating scores $r_{ui} < 3$. These pairs are used only for page-level (coarse-grained) supervision.

A.3.2 Ranked Pairs

Based on the ground truth item list $\mathcal{I}_u^{gt}$, we create ranked pairs $(\mathcal{I}_u^{gt}, \mathcal{I}_u^{r})$ to consider the ranking aspect. The preferred item list $\mathcal{I}_u^{gt}$ is the ground truth one, while the non-preferred item list $\mathcal{I}_u^{r}$ is obtained by switching items in $\mathcal{I}_u^{gt}$. This non-preferred list has a higher potential energy and is less stable. Besides page-level (coarse-grained) supervision, the switched items provide token-level (fine-grained) supervision.

A.3.3 Diversity Pairs

Based on the ground truth item list $\mathcal{I}_u^{gt}$, we create diversity pairs $(\mathcal{I}_u^{gt}, \mathcal{I}_u^{d})$ to consider the diversity aspect. The construction of these pairs is the same as that of the ranked pairs. These pairs also serve for both page-level (coarse-grained) and token-level (fine-grained) supervision.

A.3.4 Redundancy Pairs

Based on the ground truth item list $\mathcal{I}_u^{gt}$, we create redundancy pairs $(\mathcal{I}_u^{gt}, \mathcal{I}_u^{rd})$ to consider the redundancy aspect. The construction of these pairs is the same as that of the ranked pairs. These pairs also provide both page-level (coarse-grained) and token-level (fine-grained) supervision.

B Language Prompts

B.1 Pre-training Prompts

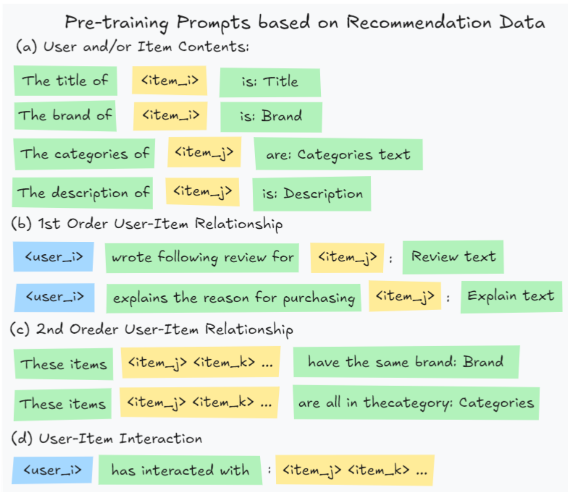

Figure 4: Pre-training prompt templates derived from recommendation data.

To enhance the model's understanding of recommendation semantics before fine-tuning, we design a set of structured pre-training prompts derived from user-item metadata and interaction logs. These prompts are categorized into four types, as illustrated in **Figure 4**:

User/Item Contents: Incorporates basic item attributes such as title, brand, category, and description to build item-aware representations.

1st Order User-Item Relationship: Captures explicit user feedback (e.g., reviews or explanations) associated with specific items.

2nd Order User-Item Relationship: Reflects co-occurrence patterns among items with shared attributes (e.g., same brand or category).

User-Item Interaction: Encodes historical interactions as token sequences for behavioral modeling.

These prompts are used for the pre-training objective to help the LLM develop task-relevant representations grounded in user and item semantics.

B.2 Fine-tuning Prompts

Personalized Predictive Prompts & Target: To enable the LLM to generate user-specific ranked item lists, we construct predictive prompts that condition on a user's past interactions. As illustrated in **Figure 5**, each input prompt includes a user token and a sequence of previously interacted items,

Figure 5: Fine-tuning prompt template for personalized ranking prediction.

followed by a masked target position. The model is trained to generate the next likely item (or list of items) that the user will interact with. This structure directly supports the learning objective defined in Equation (9), allowing the model to learn from explicit ranking supervision.

C Multi-Purpose Metric

We propose an evaluation framework that encompasses recommendation performance, ranking quality, diversity, and redundancy.

C.1 Recommendation Metric

Recall. Measures the ability of the recommendation system to cover items of interest to the user.

$$\text{Recall@K} = \frac{\text{Number of relevant items in top K}}{\text{Total number of relevant items}} \tag{14}$$

NDCG. Evaluates ranking quality, giving higher weight to items appearing earlier in the list.

$$\text{NDCG@K} = \frac{\text{DCG@K}}{\text{IDCG@K}} \tag{15}$$

$$\text{DCG@K} = \sum_{i=1}^{K} \frac{2^{rel_i} - 1}{\log_2(i+1)} \tag{16}$$

C.2 Ranked Metric

Weighted Alignment Score (WAS) : Evaluates alignment considering position importance.

$$\text{WAS} = \frac{1}{N} \sum_{i=1}^{N} w_i \cdot \max\left(0, 1 - \frac{|r_{\text{gen},i} - r_{\text{real},i}|}{\text{max\_shift}}\right) \tag{17}$$

Position-Weighted Kendall Tau (PWKT) : Measures ranking consistency with position-based weights.

$$\text{PWKT} = \frac{\sum_{i,j} w_{ij} \cdot \delta(i,j)}{\sum_{i,j} w_{ij}}, \quad w_{ij} = w_i \cdot w_j \tag{18}$$

Weighted Mean Rank Difference (WMRD) : Computes the weighted average of ranking differences.

$$\text{WMRD} = \frac{\sum_{i=1}^{N} w_i \cdot |r_{\text{gen},i} - r_{\text{real},i}|}{\sum_{i=1}^{N} w_i} \tag{19}$$

Discounted Positional Accuracy (DPA) : Evaluates ranking accuracy with logarithmic penalties.

$$\text{DPA} = \frac{\sum_{i=1}^{N} \frac{w_i}{1+\log_2(1+|r_{\text{gen},i} - r_{\text{real},i}|)}}{\sum_{i=1}^{N} w_i} \tag{20}$$

C.3 Diversity Metric

Intra-List Diversity (ILD) : Measures pairwise item similarity.

$$\text{ILD} = 1 - \frac{1}{|L|(|L|-1)} \sum_{i \neq j} \text{sim}(i,j) \tag{21}$$

C.4 Redundancy Metric

Entropy : Evaluates category balance.

$$H = -\sum_{i=1}^{N} p_i \log p_i \tag{22}$$

D Experimental Setup

D.1 Dataset Preprocessing

We use the Amazon Review dataset (McAuley and Yang, 2016) and select seven categories: Instruments, Sports, Luxury, Beauty, Food, Scientific, and Toys. We binarized the user-item interaction matrix by review scores. If the score is greater than 3, there is a connection between a user and an item. For each user in the dataset, we randomly select 80% of interactions for training, 10% for validation, and 10% for testing, with at least one sample selected in both the validation and test sets.

D.2 Implementation Details

We use GPT-2 as the backbone language model for PageLLM. The model has a token embedding dimension of 768 and supports a vocabulary of 50,257 natural language tokens. The maximum input sequence length is set to 1,024 tokens.

For fine-tuning with reinforcement learning, we adopt Proximal Policy Optimization (PPO). The model is optimized using a clipped surrogate objective with a clip range of 0.2. We set the learning rate to 1×10^{-5}, use a batch size of 64, and apply reward normalization to stabilize training. A KL-divergence penalty is added to constrain deviation from the pre-trained policy. Training is performed for 5 epochs on each dataset.

All experiments were conducted on Ubuntu 22.04.3 LTS OS, Intel(R) Core(TM) i9-13900KF CPU, with the framework of Python 3.11.5 and PyTorch 2.0.1. All data are computed on an NVIDIA GeForce RTX 4090 GPU, which features 24,576 MiB of memory with CUDA version 12.2.

E Baselines

We compare with the following baseline models:

- **Multi-VAE** (Liang et al., 2018): A variational autoencoder-based model for collaborative filtering with implicit feedback.
- **MD-CVAE** (Zhu and Chen, 2022): A mutually dependent conditional variational autoencoder designed for personalized recommendation.
- **LightGCN** (He et al., 2020): A simplified and efficient graph convolutional network for recommendation tasks that removes unnecessary components from GCNs.
- **BERT4Rec** (Sun et al., 2019): A sequential recommendation model using the BERT architecture to capture bidirectional context.
- S^3**Rec** (Zhou et al., 2020): A self-supervised learning framework that enhances sequential recommendation via multi-level data augmentation.
- **UniSRec** (Hou et al., 2022): A unified user representation model for sequential recommendation leveraging contrastive learning techniques.
- **FDSA** (Zhang et al., 2019): A feature distillation and self-attention based sequential recommendation model.
- **SASRec** (Kang and McAuley, 2018): A sequential recommendation model based on the self-attention mechanism from Transformer.
- **GRU4Rec** (Hidasi et al., 2015): A session-based recommendation model using GRU-based recurrent neural networks.
- **HSTU** (Zhai et al., 2024): Reformulate recommendation tasks as sequential transduction problems using architectures like HSTU for large-scale, high-cardinality data.
- **RecMind** (Wang et al., 2024c): Introduces a self-inspiring LLM agent capable of zero-shot personalized recommendations through external knowledge and tool usage.

F Cold-Start Study

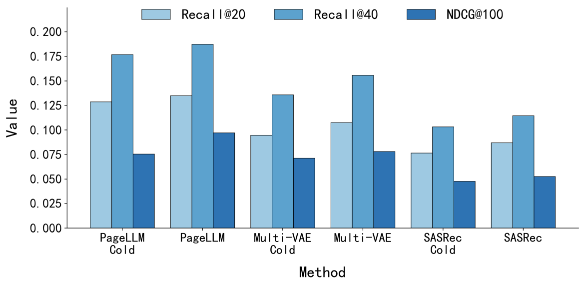

Figure 6: Performance comparison under cold-start setting on the AM-Toys dataset.

To assess robustness under limited data, we simulate a cold-start scenario by reducing the training data by half on the AM-Toys dataset. **Figure 6** compares the performance of PageLLM, Multi-VAE, and SASRec with and without cold-start constraints.

PageLLM shows a relatively small performance degradation, with Recall@20 and Recall@40 dropping by 6.2% and 5.6%, respectively, and NDCG@100 by 22.3%. In contrast, Multi-VAE and SASRec suffer larger relative drops across all metrics, especially under severe data sparsity.

These results suggest that PageLLM generalizes better in cold-start scenarios, likely benefiting from pretraining on interaction patterns and fine-grained reward signals. This highlights its potential for real-world applications where new users or items frequently emerge.

G Ablation Study

To evaluate the effectiveness of our mixed-grained reward mechanism, we conduct an ablation study on the AM-Toys dataset by comparing the full PageLLM model with its variants using only item-level rewards, only page-level rewards, and no reward supervision.

As shown in **Table 5**, removing either reward component leads to performance degradation across all metrics, indicating that both page-level and item-level signals contribute complementary information. In particular, using only item-level rewards results in a 15.2% drop in NDCG@100, while page-level only leads to a 17.8% decrease, highlighting the added value of joint optimization.

Furthermore, the full reward model also outperforms its variants in ranking alignment (WAS and DPA), and improves diversity (ILD) and category balance (Entropy). These results confirm that the

Table 5: Ablation study on the AM-Toys dataset to evaluate the impact of mixed-grained reward design. PageLLM is compared with its variants using only item-level or page-level reward signals.

Dataset	Model	Recommendation			Ranked				Diversity	Redundancy
		Recall@20 ↑	Recall@40 ↑	NDCG@100 ↑	WAS ↑	PWKT ↑	WMRD ↓	DPA ↑	ILD ↑	Entropy ↑
AM-Toys	PageLLM	0.1349	0.1873	0.0971	0.0157	0.0001	0.0005	0.0155	0.0358	0.0482
	Item-level	0.1236	0.1790	0.0823	0.0150	0.0001	0.0005	0.0145	0.0355	0.0478
	Page-level	0.1258	0.1804	0.0798	0.0150	0.0000	0.0006	0.0141	0.0355	0.0477
	w/o Reward	0.1178	0.1781	0.0754	0.0147	0.0000	0.0006	0.0139	0.0355	0.0477

mixed-grained reward mechanism facilitates more holistic and user-aligned page optimization.

H Case Study

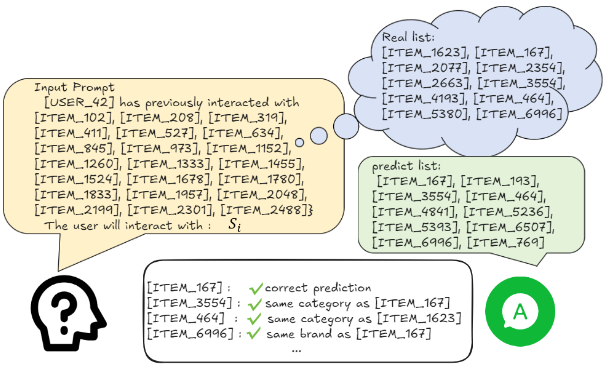

Figure 7: A case study showing PageLLM's prediction based on a user's historical interaction prompt. Predicted items align with ground-truth in both identity and semantic similarity.

To better illustrate the reasoning capability of PageLLM in personalized recommendation, we present a representative case in **Figure 7**. During inference, the model receives a prompt generated from the user's historical interactions, encoded using a predefined natural language template.

In this example, the input prompt lists 20 items previously interacted with by user `USER_42`. The model is asked to predict the next likely items that the user may be interested in. Among the predicted items, `ITEM_167` exactly matches the ground truth, while several other items such as `ITEM_3554`, `ITEM_464`, and `ITEM_6946` exhibit strong semantic and categorical alignment with the real list—either sharing the same category or brand.

This qualitative case highlights PageLLM's ability to generalize from historical patterns and make predictions that are not only accurate but also relevant. The model captures both explicit signals (i.e., exact matches) and implicit signals (e.g., semantic similarity), which aligns with the overall goal of whole-page optimization: surfacing diverse yet relevant content tailored to user preferences.