LLM-based User Profile Management for Recommender System

Anonymous ACL submission

Abstract

The rapid advancement of Large Language Models (LLMs) has opened new opportunities in recommender systems by enabling zero-shot recommendation without conventional training. Despite their potential, most existing works rely solely on users' purchase histories, leaving significant room for improvement by incorporating user-generated textual data, such as reviews and product descriptions. Addressing this gap, we propose PURE, a novel LLM-based recommendation framework that builds and maintains evolving user profiles by systematically extracting and summarizing key information from user reviews. PURE consists of three core components: a Review Extractor for identifying user preferences and key product features, a Profile Updater for refining and updating user profiles, and a Recommender for generating personalized recommendations using the most current profile. To evaluate PURE, we introduce a continuous sequential recommendation task that reflects real-world scenarios by adding reviews over time and updating predictions incrementally. Our experimental results on Amazon datasets demonstrate that PURE outperforms existing LLM-based methods, effectively leveraging long-term user information while managing token limitations.

1 Introduction

The rapid advancement of Large Language Models (LLMs) (Touvron et al., 2023; Dubey et al., 2024; Achiam et al., 2023; Team et al., 2024) has significantly impacted various domains, such as text summarization (Lewis et al., 2020a) and search (Karpukhin et al., 2020). Recent studies leverage LLMs in recommender systems for their human-like reasoning and external knowledge integration through in-context learning (Brown et al., 2020) and retrieval-augmented generation (Lewis et al., 2020b). As such, LLMs exhibit the potential to be used as *zero-shot* recommendation models without conventional training, which traditionally relies on explicit user-item interactions and training data (He et al., 2017; Kang and McAuley, 2018; He et al., 2020).

Despite the advanced capability of LLMs, most recent works (Hou et al., 2024; Wei et al., 2024; Ren et al., 2024; He et al., 2023; Zhai et al., 2023) rely solely on users' past purchase history (*i.e.,* list of purchased items). This leaves significant room for further improvement by incorporating additional user-generated textual information, such as user reviews and product descriptions, which have yet to be fully leveraged. In other words, they still fail to fully leverage various text data due to their inability to retain and process the increasing contextual information as users continue to make purchases, leading to longer recommendation sessions. This issue is primarily attributed to the *omission* of the context, either due to the information loss within the LLM's memory (Liu et al., 2024) or the memory capacity by the token limit (Li et al., 2024; Ding et al., 2024). Thus, extracting key features from a user's diverse textual sources is essential, as demonstrated in MemoryBank (Zhong et al., 2024), a framework that enhances LLMs with *long-term* memory by summarizing key information from conversations and updating user profiles.

Building on this foundation, we take the first step in extending LLMs' long-term memory beyond conversations in MemoryBank, adapting it to the evolving dynamics of recommendation systems. We propose PURE, a novel LLM-based **P**rofile **U**pdate for **RE**commender that constructs user profile by integrating users' purchase history and reviews, which naturally expand as the recommendation sessions progress. Designed specifically for recommendation in Fig. 1, PURE systematically extracts user preferences, dislikes, and key features from reviews and integrates them into structured user profiles. Specifically, PURE consists of three main components: *"Review Extractor"*, which ana-

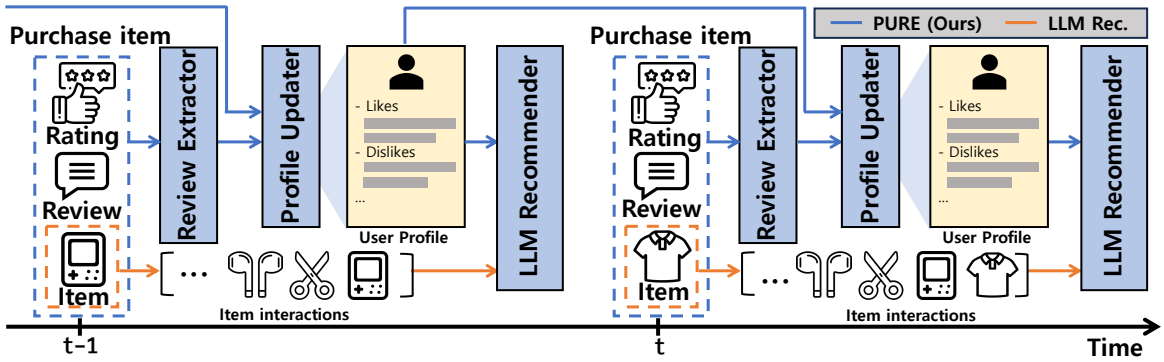


Figure 1: **Overall system of PURE.** PURE incorporates reviews, ratings, and item interactions, whereas LLM Recommender handles only item interactions. By using the "*Review Extractor*" to identify key information and the "*Profile Updater*" to refine the user profile, PURE addresses scalability issue (*i.e.,* growth of input token size).

lyzes user reviews to identify and extract user preferences, dislikes, and preferred product features, referred to as "key features", offering a comprehensive view of user interests and purchase-driving attributes; *"Profile Updater"*, which refines newly extracted representations by eliminating redundancies and resolves conflicts with the existing user profile, ensuring a compact and coherent user profile; and *"Recommender"*, which utilizes the most up-to-date user profile for recommendation task.

Our main contributions are as follows: (1) We propose PURE, a novel framework that systematically extracts, summarizes, and stores key information from user reviews, optimizing LLM memory management for the recommendation. (2) We validate the effectiveness of PURE by introducing a more realistic sequential recommendation setting, where reviews are incrementally added over time, allowing the model to update user profiles and predict the next purchase continuously. This setup more accurately reflects real-world recommendation scenarios compared to prior works, which assume all past purchases are provided at once, ignoring the evolving nature of user preferences. (3) We empirically show that PURE surpasses existing LLM-based recommendation methods on Amazon data, demonstrating its effectiveness in leveraging lengthy purchase history and user reviews.

2 Related Works

Recommendation Setup. Conventional sequential recommendation methods (Wang et al., 2019; Kang and McAuley, 2018; Sun et al., 2019; Hidasi and Karatzoglou, 2018; Kim et al., 2024) followed a one-shot prediction setup, where user history is split: the last item as the test set, the second-to-last as validation, and the rest for training. These models predict a single target item, failing to capture evolving user behavior. Tallrec (Bao et al., 2023) framed the task as binary classification to predict whether an item should be recommended.

LLM-based Recommendation. Tallrec (Bao et al., 2023) proposed the parameter efficient fine-tuning (PEFT) method in recommendation system, and A-LLMRec (Kim et al., 2024) proposed to fine-tune the embedding model for LLM to leverage the collaborative knowledge. In contrast, LLM elicited responses and extracted multiple representations from the conversation without extra training (Wang and Lim, 2023). Moreover, the authors (Dai et al., 2023) showed the potential of ChatGPT for reranking the candidates. InstructRec (Zhang et al., 2023) designed the instruction to recognize the users' intention and preference from context.

3 Method

3.1 Problem Formulation

In our recommender system, we consider the user u dataset as follow: $\mathfrak{D}_u = \{\mathbf{R}_u, \mathbf{I}_u\}$, where $\mathbf{R}_u = \{r_u^1, \cdots, r_u^{k_u}\}$ represents the historical reviews, $\mathbf{I}_u = \{i_u^1, \cdots, i_u^{k_u}\}$ denotes the corresponding purchased items, and k_u is the total number of purchased items from user u. Leveraging the user's dataset $\mathfrak{D}_u$, we aim to predict the next purchased item $i_u^{k_u+1}$ from a candidate set $\mathcal{C}_u^{k_u+1}$, which contains the ground-truth item.

One-shot Sequential Recommendation. It predicts a single next item based on a static history of user interactions up to timestep $k_u - 1$. Given the dataset $\mathfrak{D}_u$, the model observes $\mathfrak{D}_u^{k_u-1} = \{\mathbf{R}_u^{k_u-1}, \mathbf{I}_u^{k_u-1}\}$ and predicts the last item $i_u^{k_u}$ from the candidate set $\mathcal{C}_u^{k_u}$. This focuses on a one-time prediction without considering future timesteps.

Continuous Sequential Recommendation. This setup predicts the next item at every timestep

Algorithm 1: PURE

Input: Review extractor $\mathcal{E}(\cdot)$, User profile updater $\mathcal{U}(\cdot)$, Recommender $\mathcal{R}(\cdot)$, Dataset $\mathfrak{D}_u = \{\mathbf{R}_u, \mathbf{I}_u\}$ for user u, User profile $\mathbf{P}_u^t$, next purchase candidates $\mathcal{C}_u^{t+1}$, timestep t

Extract representations from reviews

$\tilde{l_u^t}, \tilde{d_u^t}, \tilde{f_u^t} = \mathcal{E}(r_u^t)$

$\hat{l_u^t} = l_u^{t-1} \cup \tilde{l_u^t}$ ▷ List of items user likes

$\hat{d_u^t} = d_u^{t-1} \cup \tilde{d_u^t}$ ▷ List of items user dislikes

$\hat{f_u^t} = f_u^{t-1} \cup \tilde{f_u^t}$ ▷ List of user's key features

Update user profile after redundancy removal

$l_u^t, d_u^t, f_u^t = \mathcal{U}(\hat{l_u^t}, \hat{d_u^t}, \hat{f_u^t})$

$\mathbf{P}_u^t = \{l_u^t, d_u^t, f_u^t\}$

Recommend next purchase item

$\text{pred} = \mathcal{R}(\mathbf{P}_u^t, \mathbf{I}_u^t, \mathcal{C}_u^{t+1})$

Output: pred

$(4 \leq t \leq k_u - 1)$, making it a multi-step prediction task. At each timestep t, the model observes the updated interaction history $\mathfrak{D}_u^t = \{\mathbf{R}_u^t, \mathbf{I}_u^t\}$ and predicts the next item i_u^{t+1} from the candidate set $\mathcal{C}_u^{t+1}$. This multi-step prediction process effectively captures temporal dependencies and allows continuous updates of user preferences, making it more aligned with real-world scenarios.

3.2 PURE: Profile Update for REcommender

In this section, we introduce PURE, novel framework that manages the user profile $\mathbf{P}_u$ from user reviews $\mathbf{R_u}$ and predict the next item with user profile. Algorithm 1 can be divided into three steps (See Appendix A for prompt template).

STEP 1: Extract User Representation.
We begin by providing the LLM with raw inputs, including user reviews $\mathbf{R_u}$ and product names $\mathbf{I_u}$. The LLM extracts $\tilde{l_u^t}$(items the user likes), $\tilde{d_u^t}$(items the user dislikes), and $\tilde{f_u^t}$(key user features) from the incoming review as user representation.

STEP 2: Update User Profile.
After the extraction in STEP 1, the extracted representation <$\tilde{l_u^t}$, $\tilde{d_u^t}$, $\tilde{f_u^t}$> concatenates with previous user profile $\mathbf{P}_u^{t-1} = \{l_u^{t-1}, d_u^{t-1}, f_u^{t-1}\}$. However, this faces a scalability issue as the number of reviews increases. Thus, leveraging the previous profile, we use an LLM to remove redundant and conflicting content from the extracted representation, yielding a more compact and up-to-date user profile $\mathbf{P}_u^t$ after concatenation.

STEP 3: Recommend Next Purhcase Item.
Recommender $\mathcal{R}$ reranks the given candidate item list to predict the user's next purchase by leveraging the updated profile $\mathbf{P}_u^t$ and purchased items $\mathbf{I}_u$.

4 Experiment

Datasets. For a thorough evaluation, we utilize two datasets from the Amazon collection (Ni et al., 2019): Video Games and Movies & TV. To ensure a comprehensive analysis, we intentionally select datasets with diverse statistical properties, particularly in terms of the number of items (See Appendix B for details).

Baselines. (Hou et al., 2024) is the recommendation method that utilizes pre-trained LLMs without additional training or fine-tuning, making it a suitable baseline. It describes three approaches for LLM-based recommendation: Sequential, Recency, and in-context learning (ICL). We compare our method with all three approaches and demonstrate the superiority of PURE when these techniques were applied to our framework, further highlighting its effectiveness. (See Appendix C.1 for details.)

Evaluation Setting. To assess the performance of PURE, we adopt a continuous sequential recommendation task. Note that NDCG scores are first aggregated per user across multiple recommendation sessions and then across all users, reflecting the continuous nature of our setup.

Implementation Details. The prediction process is framed as a classification task where the model selects one item from candidate set $\mathcal{C}_u$. Each candidate set consists of 19 randomly selected non-interacted items and a ground truth item. We adopt Llama-3.2-3B-Instruct (Touvron et al., 2023) as the backbone model for all the experiments.

4.1 Experimental Results

Impact of Review Extractor. Tab. 1 compares PURE with (1) three baselines solely based on purchased items; (2) modified baselines, marked with †, that additionally utilize users' raw reviews. The results reveal that baselines that simply combine item interactions with raw reviews show inconsistent performance improvements. In contrast, PURE, which leverages the review extractor and profile updater, significantly outperforms all baselines. This demonstrates that processing reviews at three levels, i.e., like, dislike, and key features, is essential for enhancing performance.

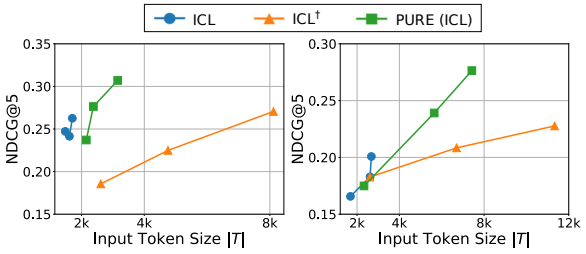
Component-wise Study. Tab. 2 shows the ablation study of PURE, where we analyze the impact of reviews (using or not using) and the effect of components (enabling or disabling the review extractor and profile updater). The use of reviews

Data	Method	Games				Movies			
		N@1	N@5	N@10	N@20	N@1	N@5	N@10	N@20
items	Sequential	10.75	18.25	23.13	28.97	9.99	15.92	20.17	26.94
	Recency	15.34	24.31	28.82	34.24	12.17	17.75	22.18	28.19
	ICL	14.28	26.57	30.51	35.72	12.03	19.56	23.36	29.91
items + reviews	Sequential†	11.14	19.95	24.97	32.00	8.05	13.11	17.72	25.57
	Recency†	12.19	23.64	28.37	35.35	8.54	15.78	21.31	29.21
	ICL†	15.11	26.34	31.25	37.39	12.24	22.10	27.31	34.52
	PURE (Sequential)	15.06	25.71	31.08	38.28	12.59	21.33	25.96	32.21
	PURE (Recency)	**18.18**	28.90	33.91	40.69	13.85	21.99	26.53	33.37
	PURE (ICL)	16.62	**29.81**	**35.60**	**42.00**	**15.80**	**26.32**	**32.03**	**38.93**

Table 1: **Comparison PURE with Baselines.** We evaluate performance under two data settings: using only item interactions and using item interactions augmented with reviews. † indicates customized baselines where review data is naively incorporated into the original prompt templates designed for item interactions only (see Appendix C.2).

Method	Data		Components			Games					Movies				
	items	reviews	Rec.	Ext.	Upd.	N@1	N@5	N@10	N@20	$\|T\|$	N@1	N@5	N@10	N@20	$\|T\|$
Sequential	✓		✓			10.75	18.25	23.13	28.97	245.52	9.99	15.92	20.17	26.94	243.89
	✓	✓	✓			11.14	19.95	24.97	32.00	29165.17	8.05	13.11	17.72	25.57	60429.80
	✓	✓	✓	✓		16.09	26.94	32.35	40.08	486.49	13.05	21.38	26.11	32.62	459.69
	✓	✓	✓	✓	✓	15.06	25.71	31.08	38.28	415.01	12.59	21.33	25.96	32.21	384.87
Recency	✓		✓			15.34	24.31	28.82	34.24	253.31	12.17	17.75	22.18	28.19	249.64
	✓	✓	✓			12.19	23.64	28.37	35.35	29235.16	8.54	15.78	21.31	29.21	60509.43
	✓	✓	✓	✓		20.85	31.36	36.51	43.19	602.13	16.00	24.81	29.66	36.98	565.13
	✓	✓	✓	✓	✓	18.18	28.90	33.91	40.69	485.85	13.85	21.99	26.53	33.37	458.60
ICL	✓		✓			14.28	26.57	30.51	35.72	268.40	12.03	19.56	23.36	29.91	261.58
	✓	✓	✓			15.11	26.34	31.25	37.39	29388.72	12.24	22.10	27.31	34.52	60800.61
	✓	✓	✓	✓		19.60	32.96	38.21	44.97	803.60	16.05	27.25	33.11	40.15	867.36
	✓	✓	✓	✓	✓	16.62	29.81	35.60	42.00	592.48	15.80	26.32	32.03	38.93	634.02

Table 2: **Component-wise study of PURE.** Each configuration varies which data sources (items, reviews) and which PURE components are used (Rec. = Recommendation, Ext. = Extractor, Upd. = Updater), as indicated by ✓. We report N@k scores ($k \in \{1, 5, 10, 20\}$) and average of input token size (|T|) for Recommender.



(a) Video Games (b) Movies and TV

Figure 2: **Trade-off between NDCG and token size.**

bring high performance gains only when accompanied by Review Extractor (Ext.). This is due to the sharp increase in input tokens (see the $|T|$ column of the 2nd and 3rd rows of each method) as the user continues purchases.

Notably, the best recommendation performance is achieved when Profile Updater (Upd.) is disabled (see the 3rd and 4th rows for each method). That is well-formed context by Review Extractor can bring higher gains when simply concatenated. However, it may face a challenge, as the number of purchases grows, leading to significant computational overhead. Thus, we use the Profile Updater (Upd.) to maintain compact user profiles, reducing input token size by 15–20% with only a slight 1–3% performance drop. This trade-off underscores the importance of using Profile Updater for long-term recommendations.

Trade-off Analysis. We categorize users into three groups based on the total cumulative review token count per user, as the criterion: 0–500 (short), 500–1000 (middle), and 1000–2000 (long) tokens. Fig. 2 presents the trade-off between recommendation performance and input token length of the three models including PURE.

PURE achieves the best trade-off, showing the steepest NDCG increase compared to other methods as input token size grows. Therefore, this demonstrates that PURE accurately distills key information from long reviews, while achieving high efficiency by minimizing input token growth without information loss, even for long-group users.

5 Conclusion

We present PURE, a novel framework for LLM-based recommendation that builds and maintains evolving user profiles by systematically extracting and summarizing user representations from reviews. By introducing a continuous sequential recommendation task, we demonstrated how updating user profiles improves recommendation quality while addressing token limitation challenges.

6 Limitations

A notable limitation of our approach is the tendency of the LLM to exhibit hallucination by occasionally recommending items beyond the predefined candidate set, even when explicitly instructed to select from it. This phenomenon underscores the inherent difficulty in imposing strict constraints within LLM-based recommendation models while maintaining flexibility and accuracy. Also, our study was constrained by the inability to utilize datasets containing a larger number of user reviews, which may have provided richer context.

7 Potential Risks

A potential risk associated with our approach is the possibility that user reviews may contain personal information, making data management and privacy protection critical concerns. Ensuring secure handling and anonymization of such data is essential to prevent breaches of user privacy.

8 Ethical Statement

This study used Amazon datasets which is publicly available. The dataset does not contain any personal identifiable information (PII), ensuring user privacy and ethical compliance. To ensure fair and accurate evaluation, we customized the baseline models by incorporating both item interactions and user reviews, enabling a more balanced comparison with our proposed approach. Lastly, our research aims to advance the development of recommendation systems while avoiding potential negative impacts such as bias or misuse.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *ArXiv:2303.08774*.

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 B Brown, Benjamin Mann, Nick Ryder, et al. 2020. Language models are few-shot learners. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 1877–1901.

Sunhao Dai, Ninglu Shao, Haiyuan Zhao, Weijie Yu, Zihua Si, Chen Xu, Zhongxiang Sun, Xiao Zhang, and Jun Xu. 2023. Uncovering chatgpt's capabilities in recommender systems. In *Proceedings of the 17th ACM Conference on Recommender Systems*, pages 1126–1132.

Yiran Ding, Li Lyna Zhang, Chengruidong Zhang, Yuanyuan Xu, Ning Shang, Jiahang Xu, Fan Yang, and Mao Yang. 2024. Longrope: Extending llm context window beyond 2 million tokens. *ArXiv:2402.13753*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *ArXiv:2407.21783*.

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 international ACM SIGIR conference on research and development in Information Retrieval*, pages 639–648.

Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of International Conference on World Wide Web (WWW)*, pages 173–182.

Zhankui He, Zhouhang Xie, Rahul Jha, Harald Steck, Dawen Liang, Yesu Feng, Bodhisattwa Prasad Majumder, Nathan Kallus, and Julian McAuley. 2023. Large language models as zero-shot conversational recommenders. In *Proceedings of the ACM International Conference on Information and Knowledge Management (CIKM)*, pages 720–730.

Balázs Hidasi and Alexandros Karatzoglou. 2018. Recurrent neural networks with top-k gains for session-based recommendations. In *Proceedings of the 27th ACM international conference on information and knowledge management*, pages 843–852.

Yupeng Hou, Junjie Zhang, Zihan Lin, Hongyu Lu, Ruobing Xie, Julian McAuley, and Wayne Xin Zhao. 2024. Large language models are zero-shot rankers for recommender systems. In *Proceedings of European Conference on Information Retrieval*, pages 364–381. Springer.

Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *Proceedings of International Conference on Data Mining (ICDM)*, pages 197–206. IEEE.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781.

Sein Kim, Hongseok Kang, Seungyoon Choi, Donghyun Kim, Minchul Yang, and Chanyoung Park. 2024.

Large language models meet collaborative filtering: An efficient all-round llm-based recommender system. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1395–1406.

Mark Lewis, Yinhan Liu, et al. 2020a. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020b. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 9459–9474.

Xinyi Li, Sai Wang, Siqi Zeng, Yu Wu, and Yi Yang. 2024. A survey on llm-based multi-agent systems: workflow, infrastructure, and challenges. *Vicinagearth*, 1(1):9.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pages 188–197.

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*, 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.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. 2024. Gemma 2: Improving open language models at a practical size. *ArXiv:2408.00118*.

Hugo Touvron, Thibaut Lavril, et al. 2023. Llama: Open and efficient foundation language models. *ArXiv:2302.13971*.

Lei Wang and Ee-Peng Lim. 2023. Zero-shot next-item recommendation using large pretrained language models. *ArXiv:2304.03153*.

Shoujin Wang, Liang Hu, Yan Wang, Longbing Cao, Quan Z Sheng, and Mehmet Orgun. 2019. Sequential recommender systems: Challenges, progress and prospects. In *Proceedings of International Joint Conference on Artificial Intelligence Organization (IJCAI)*, pages 6332–6338.

Wei Wei, Xubin Ren, Jiabin Tang, Qinyong Wang, Lixin Su, Suqi Cheng, Junfeng Wang, Dawei Yin, and Chao Huang. 2024. Llmrec: Large language models with graph augmentation for recommendation. In *Proceedings of the ACM International Conference on Web Search and Data Mining*, pages 806–815.

Jianyang Zhai, Xiawu Zheng, Chang-Dong Wang, Hui Li, and Yonghong Tian. 2023. Knowledge prompt-tuning for sequential recommendation. In *Proceedings of ACM International Conference on Multimedia*, pages 6451–6461.

Junjie Zhang, Ruobing Xie, Yupeng Hou, Xin Zhao, Leyu Lin, and Ji-Rong Wen. 2023. Recommendation as instruction following: A large language model empowered recommendation approach. *ACM Transactions on Information Systems*.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. 2024. Memorybank: Enhancing large language models with long-term memory. In *Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI)*, volume 38, pages 19724–19731.

-Supplementary Material-

LLM-based User Profile Management for Recommender System

A Prompt Template

A.1 Extractor $\mathcal{E}$

The extractor $\mathcal{E}$ aims to extract the user representations from reviews. Here is the prompt template.

Prompt template for Extractor $\mathcal{E}$

```
I purchased the following products and left
reviews in chronological order: {input_reviews}
Analyze user's likes/dislikes/key features by
referring to their reviews.
```

A.2 Profile Updater $\mathcal{U}$

The purpose of the profile updater $\mathcal{U}$ is to remove the redundant information in the user profile. As such, the prompt template is designed as below:

Prompt template for User Profile Updater $\mathcal{U}$

```
You are given a list: {list}
Update this list by removing redundant or
overlapping information. Note that crucial
information should be preserved.
```

A.3 Recommender $\mathcal{R}$

Due to utilizing both item interactions and user profile, prompt can be constituted of various components. Below one is the prompt template of the recommender.

Prompt template for Recommender $\mathcal{R}$

```
Positive aspects: {likes}
Negative aspects: {dislikes}
Key Features: {key_features}
Based on these inputs, rank the {candidate_list}
from 1 to 20 by evaluating their likelihood of
being purchased.
```

B Dataset

Amazon Review Dataset (Ni et al., 2019) contains product reviews and metadata from Amazon, including 142.8 million reviews spanning May 1996 – July 2014. Specifically, this dataset includes reviews (ratings, text, helpfulness votes), product metadata (descriptions, category information, price, brand, and image features), and links (also viewed/also bought graphs). Among them, we selected two domain datasets (Video Games and Movies & TV), and we utilized ASIN, product name, rating, and review for each data and sort the reviews chronologically for each user. Here are the specific descriptions for each dataset.

Video Games. We select about 15K users and 37K items. Following existing studies (Kang and McAuley, 2018), we removed users and items with fewer than 10 interactions.

Movies and TV. We select about 98K users and 126K items, removing users and items with fewer than 10 interactions as in the Video Games dataset.

C Baselines

C.1 User-Item interactions

In our experimental setup, the LLM is tasked with predicting the item that a user is likely to purchase at time step t. We utilize **user-item interactions** up to time step (t-1) in chronological order and constructed a candidate list consisting of one ground-truth item and 19 non-interacted items as input. Here, time step t refers to the period starting from the user's 4th purchase up to their final purchase k.

Sequential. We provide the LLM with instructions, supplying only the user-item interactions and the candidate list. The LLM was then tasked with ranking the items in the candidate list based on the likelihood of being purchased at time step t.

Recency-Focused. In the *sequential* prompt above, we add an instruction to emphasize the most recently purchased item, specifically the item bought at time step (t-1). The additional prompt is as follows: *"Note that my most recently purchased item is* ***{recent item}****."*

In-Context Learning. Unlike the previous *sequential* and *recency-focused* prompts, this approach utilize **user-item interactions** only up to time step (t-2) and recently purchased item which is bought at time step (t-1) as input. The additional prompt is as follows: *"I've purchased the following products:* ***{user-item interactions}****, then you should recommend* ***{recent item}*** *to me and now that I've bought* ***{recent item}****."*

C.2 User-Item interactions & User Reviews

In this setup, we extend **user-item interactions** to include both interactions and **user reviews**. Based

on Appendix C.1, the † present the results when both **user-item interactions** and **user reviews** are used as input.

D Qualitative Results

To validate the effectiveness of each component of PURE, we summarized the qualitative results in Tab. 3, which illustrates the entire input/output process for both the baselines and PURE in the sequential recommendation task. We can observe that the Review Extractor first removes irrelevant or uninformative content for the given reviews, while the Profile Updater reduces redundancy and overlapping information in the user profile. As such, we can conclude that PURE reduces the input token size of the recommender system while retaining essential information, making it more memory-efficient and potentially improving overall performance.

Method	Type	Contents
Baselines	*Recommender* **Input**	I've purchased the following products in chronological order: {**user-item interactions & reviews**} Then if I ask you to recommend a new product to me according to the given purchasing history, you should recommend **{recent item}** and now that I've just purchased **{recent item}**. There are 20 candidate products that I can consider to purchase next: **{20 candidate items}** Please rank these 20 products by measuring the possibilities that I would like to purchase next most, according to the given purchasing records. Please think step by step. Please show me your ranking results with order numbers. Split your output with line break. You MUST rank the given candidate product. You cannot generate products that are not in the given candidate list. No other description is needed.
	Recommender **Output**	[20 ordered items]
PURE **(Ours)**	*Review Extractor* **Input**	I purchased the following products in chronological order: **{user-item interactions & reviews}** Then if I ask you to recommend a new product to me according to the given purchasing history, you should recommend **{recent item}** and now I've just purchased **{recent item}**. And I left review: **{recent item review}** Your task is to analyze user's purchasing behavior and extract user's likes, dislikes and key features from the input review. Response only likes/dislikes/key features in descriptive form. Please prioritize the most recent item **{recent item}** when analyzing likes/dislikes/key features. Split likes, dislikes, and key features and response in same format.
	Review Extractor **Output**	**Likes**: {['*Long gameplay experience(50-60 hours), *Responsive controls, *Fantastic storyline, *Challenging puzzles, *Emotional resonance (e.g.remorse), *Ability to gain new posers by killing enemies', '*Humor and fun in games, *References to the simpsons franchise, *Variety of playable characters (Marge, Lisa, Apu, Bart, and Homer), *Ability to drive or walk depending on preference, *Great voice acting from the cast members, *Presence of key locations from the Simpsons universe (Kwik-E-Mart, Power Plant, Church, etc.), *Cool vehicle designs and stats, *Fantastic game overall']} **Dislikes**: {['*No pause time when selecting a weapon, making the player vulnerable, *Inventory management can be inconvenient, requiring the player to switch to the inventory screen to user gadgets', '*Boring story, *Not funny, *Awful weapons, *Unresponsive controls, *Terrible graphics, *Worse gameplay']} **Key Features**: {['*No in-game loading, *Fighting mechanics, *Soul-hunger gameplay mechanic, *Ability to cover up face to hide disfigured jaw', '*New camera system (Devil May Cry position), *Redone fighting mechanics, *Playable as both Raziel and Kain, *Puzzles with a challenging but fun diffculty level']}
	Profile Updater **Input**	You are given a list: **{list of likes/dislikes/key features}** You have to update this list by removing redundant or overlapping information. Note that crucial information should be preserved. Please response only a list. No other description is needed.
	Profile Updater **Output**	**Likes**: {['*Long Gameplay experience (50-60 hours), *Challenging puzzles, *Emotional resonance (e.g.remorse), *Ability to gain new powers by killing enemies', '*Variety of playable characters, *Ability to drive or walk depending on preference, *Presence of key locations from the Simpsons universe, *Great voice acting, *Cool vehicle designs and stats']} **Dislikes**: {['*No pause time when selecting a weapon, making the player vulnerable, *Inventory management can be inconvenient', '*Unresponsive controls, *Terrible graphics, *Worse gameplay']} **Key Features**: {['*Fighting mechanics, *Soul-hunger gameplay mechanic, *Ability to cover up face', '*New camera system, *Redone fighting mechanics, *Playable as both Raziel and Kain, *Puzzles']}
	Recommender **Input**	**This is positive aspects from purchase history**: {['*Long Gameplay experience (50-60 hours), *Challenging puzzles, *Emotional resonance (e.g.remorse), *Ability to gain new powers by killing enemies', '*Variety of playable characters, *Ability to drive or walk depending on preference, *Presence of key locations from the Simpsons universe, *Great voice acting, *Cool vehicle designs and stats']} **This is negative aspects from purchase history**: {['*No pause time when selecting a weapon, making the player vulnerable, *Inventory management can be inconvenient', '*Unresponsive controls, *Terrible graphics, *Worse gameplay']} **This is key features of products**: {['*Fighting mechanics, *Soul-hunger gameplay mechanic, *Ability to cover up face', '*New camera system, *Redone fighting mechanics, *Playable as both Raziel and Kain, *Puzzles']} Based on these inputs, your task is to rank 20 candidate products by evaluating their likelihood of being purchased. Now there are 20 candidate products that I consider to purchase next. Note that there is no specific order for these candidate items. Please rank the **{20 candidate items}** from 1 to 20. Your task is to rank these products based on the likelihood of purchase. You cannot generate products that are not in the given candidate list. No other description is needed.
	Recommender **Output**	{[20 ordered items]}

Table 3: **Qualitative Results: Baselines vs PURE.** Note that green-highlighted boxes indicate portions removed due to redundancy or overlapping information, while yellow-highlighted boxes represent summarized content where unnecessary modifiers or examples were omitted for conciseness.