HORIZON: A Benchmark for in-the-wild User Behavior Modelling

Arnav Goel[♠*] **Pranjal A. Chitale**[♡] **Bhawna Paliwal**[◇*]
Bishal Santra[♡] **Amit Sharma**[♡†]

[♠]Carnegie Mellon University [♡]Microsoft Research India
[◇]University of California, Berkeley

arnavgoe@cs.cmu.edu, amshar@microsoft.com

microsoft/horizon-benchmark

Abstract

User behavior in the real world is diverse, cross-domain, and spans long time horizons. Existing user modeling benchmarks however remain narrow, focusing mainly on short sessions and next-item prediction within a single domain. Such limitations hinder progress toward robust and generalizable user models. We present HORIZON a new benchmark that reformulates user modeling along three axes i.e. dataset, task, and evaluation. Built from a large-scale, cross-domain reformulation of Amazon Reviews, HORIZON covers 54M users and 35M items, enabling both pretraining and realistic evaluation of models in heterogeneous environments. Unlike prior benchmarks, it challenges models to generalize across domains, users, and time, moving beyond standard missing-positive prediction in the same domain. We propose new tasks and evaluation setups that better reflect real-world deployment scenarios. These include temporal generalization, sequence-length variation, and modeling unseen users, with metrics designed to assess general user behavior understanding rather than isolated next-item prediction. We benchmark popular sequential recommendation architectures alongside LLM-based baselines that leverage long-term interaction histories. Our results highlight the gap between current methods and the demands of real-world user modeling, while establishing HORIZON as a foundation for research on temporally robust, cross-domain, and general-purpose user models.

1 Introduction

Personalization is at the core of modern digital platforms, driving user engagement in domains such as e-commerce, streaming, and social networks by tailoring content and services to individual preferences. Early personalization methods relied mainly on static user representations and focused on prediction of next-item within single-domain datasets such as MovieLens (Harper and Konstan, 2015a) and Amazon Reviews (Ni et al., 2019). However, contemporary user behavior is inherently multi-faceted, spanning diverse content types and multiple platforms, and reflecting complex, evolving preferences and latent interests that are not adequately captured by static models (Zhou et al., 2024; Treves et al., 2025; Hou et al., 2022b; Lin et al., 2019). To address these challenges, recent work has increasingly framed personalization as a *sequential recommendation* problem, aiming to model long-term dependencies and dynamic user intent from interaction histories (Kang and McAuley, 2018; Sun et al., 2019; Hou et al., 2022b).

Despite significant advances in modeling world knowledge (Yue et al., 2023; Wang et al., 2024; Goel et al., 2023) and semantic reasoning driven by transformers and pretrained large language models (LLMs) (Anand et al., 2023; Kapuriya et al., 2025; Li et al., 2025), prior work in personalization largely remains constrained to *single-domain* recommendation settings. Such formulations fail to leverage cross-domain signals and instead learn fragmented, domain-specific representations of users, capturing only a partial view of their preferences (Kang and McAuley, 2018; Sun et al., 2019; Hou et al., 2022b, 2024). Even when recent datasets, such as Amazon-Reviews 2023 (Hou et al., 2024), incorporate interactions across multiple domains, evaluation protocols and task formulations continue to treat each domain independently. This disconnect highlights a critical gap between real-world user behavior and existing benchmarking practices, motivating the need for a unified framework that can evaluate and advance truly cross-domain personalization.

Existing benchmarks predominantly focus on single-domain, next-item prediction, inadvertently encouraging models to exploit item–item similar-

[*]Work done while at Microsoft Research India
[†]Corresponding author

Findings of the Association for Computational Linguistics: ACL 2026, pages 30060–30078
July 2-7, 2026

Attribute	PF	Amz-M2	MIND	Amz-Reviews	HORIZON
No. of users	N/A	N/A	1M	54.51M	**54.51M**
Avg User History Length	N/A	4.2	N/A	3.86	**9.07**
No. of items	N/A	1.42M	0.16M	34.52M	**34.52M**
No. of interactions	N/A	16.78M	24.15M	485.89M	**485.89M**
Cross-domain	✓	✓	×	×	✓
Diversity	✓	×	×	×	✓
Interaction Timestamps	–	×	×	✓	✓
Open-Source	×	✓	✓	✓	✓

Table 1: Comparison of existing Sequential Recommendation Benchmarks with HORIZON. (PF refers to PinnerFormer, Amz-M2 refers to Amazon M2, Amz-Reviews is the Amazon Reviews dataset.)

ities rather than develop a holistic understanding of user preferences (Rendle et al., 2020). While recent efforts such as PinnerFormer (Pancha et al., 2022) and USE (Zhou et al., 2024) highlight the importance of richer user modeling, they rely on private datasets, limiting reproducibility and leaving a critical gap in open benchmarking standards. Moreover, real-world recommendation scenarios often unfold over extended time horizons. For instance, user behaviors in e-commerce require models to reason over multi-year interaction histories and anticipate long-term intent. Such capabilities are essential for applications like proactive recommendation and inventory planning. However, current benchmarks, constrained to short-term next-item prediction within limited temporal windows, fail to evaluate whether models can capture long-range dependencies and evolving user preferences across extended time spans.

Taken together, these limitations point to a fundamental gap in current evaluation practices: existing benchmarks do not adequately measure whether models can generalize across domains, reason over long temporal horizons, or capture deeper semantic structure in user behavior. In particular, three key challenges remain underexplored: **(i) cross-domain generalization**, where models must transfer knowledge across diverse content domains and platforms; **(ii) long-range temporal generalization**, requiring anticipation of user intent far beyond the immediate training window; and **(iii) semantic understanding**, involving the ability to uncover latent, non-obvious relationships within user interaction histories.

To address these challenges, we introduce HORIZON, a fully open-source, large-scale benchmark for evaluating sequential recommendation models under realistic cross-domain and long-horizon settings. Unlike prior benchmarks that rely on ratio-based temporal splits and short-term evaluation, HORIZON disentangles key generalization axes by explicitly distinguishing between *seen* and *unseen* users, as well as temporally *close* versus *distant* scenarios relative to the training distribution. This structured evaluation reveals insights obscured by existing paradigms: we observe substantial variation in model performance across temporal and user generalization regimes, showing that strong in-distribution results do not reliably translate to real-world robustness. For instance, BERT4Rec (Sun et al., 2019), while state-of-the-art in standard settings, degrades significantly for out-of-distribution users yet remains competitive in long-range temporal extrapolation (e.g., Recall@50 and Recall@100). Moreover, we find that LLMs do not consistently outperform specialized architectures on user behavior modeling tasks.

In summary, our contributions are as follows:

1. We introduce HORIZON, the first fully open-source benchmark for evaluating sequential recommendation models across *cross-domain* and *long-horizon* personalization settings.
2. We propose a unified evaluation framework that disentangles key generalization axes i.e. *seen vs. unseen users* and *seen vs. unseen timeframes*, enabling more faithful and fine-grained assessment of real-world performance.
3. We provide a comprehensive empirical study revealing previously unobserved trade-offs in modern recommendation models, including discrepancies between in-distribution accuracy, temporal generalization, and out-of-distribution user adaptation, as well as the limitations of LLMs for user modeling.

2 Related Work

Sequential Recommendation Datasets: Sequential recommendation research has primarily re-

lied on a small number of established benchmarks that, while influential, offer limited coverage of realistic user behavior. MovieLens (Harper and Konstan, 2015b) remains one of the most widely used datasets, providing temporally ordered movie ratings; however, it is relatively small in scale and strictly single-domain. Even its largest variant, MovieLens-25M, contains only 25 million interactions and reflects a narrow media-focused item space, limiting its suitability for evaluating large-scale or cross-domain user modeling.

Several other commonly used datasets share similar constraints. Yelp, which contains approximately 1.2 million users and 5 million reviews of local businesses,[1] and Gowalla (Cho et al., 2011), a location-based dataset with 197K users and 6.4M check-ins collected between 2009-2012, are both single-domain and exhibit weak sequential structure. Recent analysis shows that shuffling test-time sequences in such datasets has minimal impact on the performance, suggesting limited temporal dependency (Klenitskiy et al., 2024). Similarly, the Steam dataset (Sobkowicz and Stokowiec, 2016), while capturing richer behavioral signals such as purchases and playtime, remains confined to the gaming domain and is modest in scale, with fewer than 8 million interactions.

The Amazon Reviews dataset (Ni et al., 2019; Hou et al., 2024) provides substantially larger coverage across many product categories, but is typically evaluated by partitioning categories into isolated recommendation tasks. This segmentation prevents models from capturing cross-category transitions that naturally arise in real user behavior and often leads to severe sparsity within individual categories. As a result, existing benchmarks fall short of supporting large-scale, cross-domain, and temporally grounded evaluation of sequential user modeling.

Towards Large-scale and Long Horizon User Modeling: To address the limitations of traditional benchmarks, several datasets have sought to capture richer and broader user behaviors. MIND (Wu et al., 2020a) provides large-scale news consumption logs with approximately one million users and rich textual features, but is restricted to a single domain and short user histories spanning only two weeks, limiting its suitability for long-horizon or cross-domain evaluation. Amazon-M2 (Jin et al., 2023a) extends coverage to multilingual and cross-locale e-commerce interactions across six regions, but is primarily designed for session-based recommendation rather than modeling long-term user behavior over extended time spans.

More recent efforts approach the required scale and temporal depth but remain inaccessible. The Pinterest dataset used in PinnerFormer (Pancha et al., 2022) contains billions of multimodal interactions across multiple years, and USE (Zhou et al., 2024) includes diverse behavioral sequences from Snapchat users. However, both datasets are proprietary, limiting reproducibility and preventing their adoption as public benchmarks for large-scale, long-horizon user modeling.

3 HORIZON Benchmark

Benchmark Description: User modeling and sequential recommendation aim to predict a user's future interactions based on their past behavior. Formally, for a user u, we observe a sequence of interactions over time $\mathcal{H}_u = [i_1, i_2, \ldots, i_t]$, where i_t denotes the item interacted with at time t. The objective is to estimate the likelihood of the next interaction i_{t+1} or future next interactions over some time period T *i.e.* $\hat{i}_{t+1,..,T} = (i_{t+1}, i_{t+2}, ...i_T)$, given the user's historical context:

$$\hat{i}_{t+1} = \arg\max_{i \in \mathcal{I}} \Pr(i \mid \mathcal{H}_u),$$

where $\mathcal{I}$ denotes the candidate item set. This formulation underpins several established benchmarks such as MIND, M2, and Amazon Reviews (Wu et al., 2020b; Jin et al., 2023b; Hou et al., 2024). As noted in Section 2, the Amazon Reviews dataset has become a widely used resource for training and evaluating sequential recommenders. However, it segregates user interactions by product categories, making it domain-specific and thus limiting its ability to capture holistic user preferences. In the real world, users engage with a variety of domains, and isolating interactions to a single domain introduces artificial boundaries, resulting in incomplete modeling of cross-domain behaviors and potentially spurious patterns causing incorrect user modeling.

To address this limitation, we introduce HORIZON, a large-scale benchmark designed to support cross-domain user modeling and sequential recommendation. HORIZON is constructed by refactoring and consolidating the Amazon Reviews 2023 dataset (Hou et al., 2024), merging interactions

[1] https://business.yelp.com/data/resources/open-dataset/

across all available categories to create unified, realistic user histories. The resulting benchmark comprises of 53.5 million users and 34.5 million unique items, enabling rigorous evaluation of models under settings that better reflect real-world recommendation scenarios. [2]

3.1 Comparison with Existing Benchmarks

Table Section 1 provides a comparative analysis of our dataset against existing sequential recommendation benchmarks. While proprietary datasets like PinnerFormer (Pancha et al., 2022) offer scale and diversity, they remain inaccessible to the broader research community. Public datasets such as Amazon-M2 (Jin et al., 2023a) provide cross-domain capabilities but lack temporal depth due to these being restricted to session-based interactions rather than long-term user modeling. The MIND dataset (Wu et al., 2020a), despite its million-user scale, covers only two weeks of user history, severely limiting its utility for long-horizon recommendation research (Klenitskiy et al., 2024). Similarly, the Amazon Reviews dataset (Ni et al., 2019; Hou et al., 2024) provides timestamps, but artificially segments interactions into isolated domains. In contrast, HORIZON uniquely combines cross-domain coverage, interaction diversity, and comprehensive temporal information, enabling more realistic evaluation of sequential recommendation systems across extended time horizons.

4 Task Formulations

4.1 Traditional Evaluation Setups

Drawbacks of Traditional Evaluation Setups Traditional evaluation methodologies in recommendation systems have primarily focused on **in-distribution settings**, where the training, validation, and test splits are derived from the same user interaction trace. This leads to substantial overlap in distributional characteristics across splits, limiting the capacity to evaluate generalization or robustness in real-world scenarios (McElfresh et al., 2022). Two widely adopted paradigms for this evaluation are *Leave-One-Out* and *Ratio-Based* evaluations.

Leave-One-Out. For a user history sequence with n events, the $(n-1)^{\text{th}}$ interaction is held out for validation, the n^{th} for testing, and the preceding $(n-2)$ interactions form the training set. This approach has been widely deployed for evaluating user modeling architectures in recent years (Sun, 2023) but can often leak future interactions into training data, violating the temporal order of real-world scenarios. This leads to inflated performance metrics that don't reflect practical deployment conditions (Meng et al., 2020; Ji et al., 2023).

Ratio-Based. Here, the user sequence is split into training, validation, and test segments based on interaction timestamps, such that the resulting partition approximately adheres to a predefined ratio—typically 8:1:1. While this method introduces some variability in sequence lengths and improves over the deterministic nature of Leave-One-Out, it fails to evaluate out-of-distribution generalization to unseen users. Moreover, it can result in temporal leakage, where interactions in the training set of one user may overlap in time with test interactions of another (Meng et al., 2020).

To conclude, both paradigms fail to evaluate generalization under distribution shift, as validation and test sets largely resemble the training distribution. This undermines the assessment of model robustness and overlooks the temporal evolution of user preferences, which central to real-world user behavior modeling. To address these shortcomings, we propose a temporally grounded evaluation protocol that enforces a fixed time-based cutoff and includes held-out users to explicitly test for extrapolation and out-of-distribution generalization to unseen users.

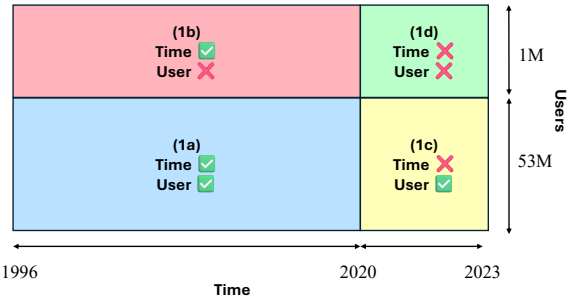

Figure 1: Proposed evaluation splits on the HORIZON benchmark for Task 1.

4.2 Proposed Task Formulations on HORIZON

To address the aforementioned limitations, we introduce a multi-axis evaluation protocol on the HORIZON benchmark that disentangles generalization across time and user identity. This design enables a controlled analysis of each factor's impact.

[2] Detailed plots and stats for the benchmark are added to the appendix A

We instantiate this framework through three tasks, beginning with traditional next-item recommendation, followed by two LLM-based user modeling tasks motivated by their growing industrial relevance.

Task 1 — Next Item Recommendation. In the traditional next-item recommendation setting, given a user history $\mathcal{H}_u = [i_1, i_2, \ldots, i_t]$, the goal is to recommend the next likely item i_{t+1}. To move beyond static evaluation protocols, we adopt a *global temporal cut-off* τ, using interactions before τ for training and those after for evaluation. This preserves temporal order and induces realistic distribution shifts, following prior work that advocates temporally grounded evaluation in recommender systems (Meng et al., 2020).

Crucially, rather than treating temporal generalization in isolation, we explicitly factor evaluation along two orthogonal dimensions: *temporal position* (before vs. after τ) and *user visibility* (users seen vs. unseen during training). This yields four complementary evaluation settings that systematically disentangle in-distribution performance, temporal extrapolation, user-level generalization, and their combination (illustrated in Figure 1). Together, these settings provide a fine-grained view of model behavior under progressively harder and more realistic distribution shifts, while using a single, fixed training protocol[3]:

(1a) **In-Distribution, Temporally Aligned Evaluation (Leave-One-Out).** For users included in training, we hold out their final interaction *before* the global cut-off τ for testing, using the preceding interactions for training. This mirrors the standard Leave-One-Out setup, but restricted to a temporally consistent subset. It evaluates short-context prediction under a distribution closely matched with training, and constitutes the **only training setup** we propose.

(1b) **In-Distribution, Temporal Extrapolation Evaluation (All-Post-τ).** Using the model trained in 1a, we evaluate on the full sequence of interactions occurring *after* the temporal cut-off τ for the same users. Ground-truth items are incrementally revealed during evaluation, enabling assessment of temporal generalization and user preference evolution.

(1c) **OOD-User, Temporally Aligned Evaluation (Leave-One-Out).** For held-out users unseen during training, we perform Leave-One-Out evaluation on interactions before the temporal cut-off τ. This setting assesses generalization to new users under temporal conditions aligned with the training distribution.

(1d) **OOD-User, Temporal Extrapolation Evaluation (All-Post-τ).** In this most challenging setting, the model predicts all interactions following the temporal cut-off τ for held-out users unseen during training. With short, temporally recent histories as input, this task evaluates generalization across both user identity and time.

Task 2 - LLM-Based Next Item Recommendation: Large language models (LLMs), extensively pre-trained on web-scale corpora, are well-suited to model semantic patterns in text. As demonstrated in Figure 2, in this task, we treat the LLM as a user behavior encoder that reformulates a given user history into a diverse set of ten search queries $Q = \{q_1, \ldots, q_{10}\}$, intended to capture various aspects of user intent and preference.[4] These queries are mapped, along with catalog items $i_j \in \mathcal{I}$, into a shared embedding space using a pre-trained item encoder.

An approximate nearest neighbor (ANN) index is constructed over catalog item embeddings $\{\mathbf{i}_j\}$, and top-K candidates are retrieved for each query embedding $\mathbf{q}_k$. These are merged to form a final set of K recommendations $\hat{I}$. We evaluate retrieval quality using Precision@K and Recall@K, based on cosine similarity with the ground truth item.

Beyond ranking metrics, this task uniquely probes the LLM's ability to generate high-quality, interpretable queries that reflect the underlying user behavior—serving as a semantic bridge between user history and candidate retrieval.

Task 3 - LLM-Based Long-Horizon User Modeling. Traditional sequential recommendation typically focuses on predicting the immediate next item in a user's interaction sequence. However, user modeling requires capturing longer-term behavior patterns that unfold over extended time windows (Zhou et al., 2024; Pancha et al., 2022). Motivated by this, we propose the long-horizon modeling task on the HORIZON benchmark, leveraging the availability of longer and cross-domain user histories.

[3] Analysis of Distribution Shifts in Appendix B, C

[4] Prompts in Appendix F

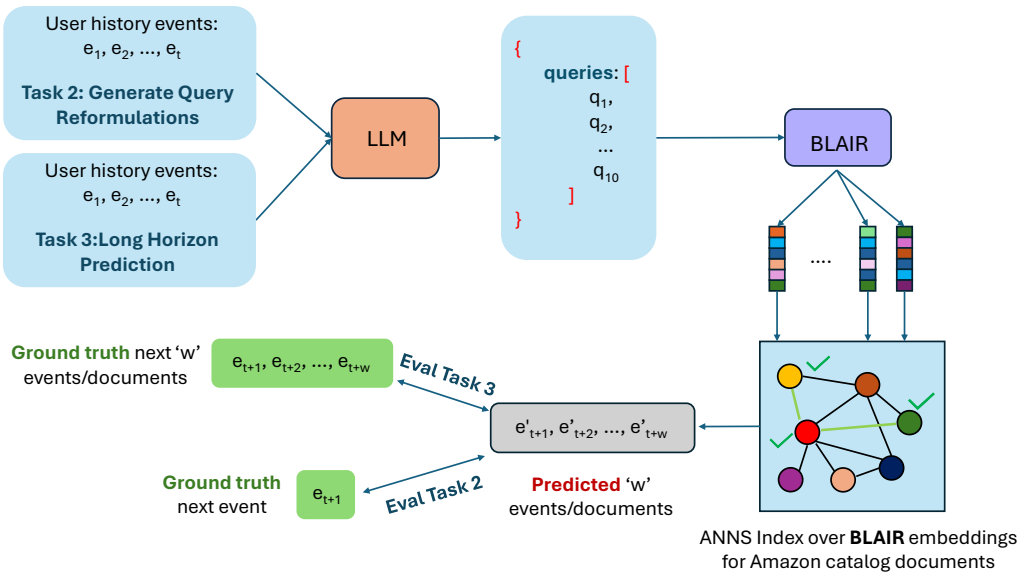

Figure 2: Pipeline Detailing the LLM Generation, Retrieval and Evaluation Process Proposed for Tasks 2 and 3.

Given a user's interaction history prior to a temporal cut-off, the LLM is tasked with generating natural language descriptions of the next 10 items the user is likely to engage with. These descriptions represent a high-level summary of future behavior across a wider horizon. Using the same retrieval pipeline as Task 2, each generated description is embedded and used to retrieve matching items from the catalog.

Evaluation is performed using standard retrieval metrics (e.g., Recall@K, Precision@K) by comparing the retrieved items with ground truth future interactions. This task assesses the LLM's ability to model long-term user intent and will be an essential evaluation for assessing their ability to model user behaviors.

5 Experimental Setup

Task 1 Setup: We adopt a temporal cut-off of $\tau = 2020$ to define the training window. From the full dataset of ∼54M users, we randomly sample 1M users who have any post-τ interactions as our out-of-distribution (OOD) user set, and treat the remaining 53M as the in-distribution (IND) pool. From this IND pool, 1M users are sampled to construct the test set for sub-task (1c). Due to computational constraints, we train all models on a 100K user subset of the IND set, and evaluate on 25K users each for sub-task (1d) (IND extrapolation) and sub-task (1c) (OOD prediction). [5] For all baselines, we use the RECBOLE framework (Zhao et al., 2021, 2022), which offers standardized implementations and reproducible pipelines for recommendation models. The following popular item-ID-based baselines are included:

GRU4REC (Hidasi et al., 2016) employs a recurrent architecture with gated recurrent units to capture sequential dependencies in user histories. SASREC (Kang and McAuley, 2018) adopts a transformer-based architecture with self-attention mechanisms to model user behavior sequences. BERT4REC (Sun et al., 2019) utilizes a bidirectional transformer encoder trained with a Cloze-style objective to leverage full-sequence context. CORE (Hou et al., 2022a) formulates session representations as weighted linear combinations of item embeddings, aligning both session and item vectors in a shared latent space.

While these methods are typically evaluated using either ratio-based or leave-one-out strategies, we retrain and evaluate them under the temporally grounded evaluation protocol described in the previous section. All models are trained with standardized hyperparameters and evaluated on our four evaluation settings using **MRR**, **Recall@K**, and **NDCG@K** for $K = \{10, 50, 100\}$.

Task 2 and 3 Setup: For Tasks 2 and 3, we use the held-out out-of-distribution (OOD) test set comprising 1M users as our evaluation benchmark. We primarily focus on evaluating the zero-shot capabilities of LLMs for modeling user behavior, as effective training paradigms for LLMs in recom-

[5] Detailed stats highlighting difference between the splits is added to the Appendix.

Table 2: In-Distribution User - Temporally Aligned Evaluation (N=NDCG, M=MRR, R=Recall)

Baseline	N			M			R		
	10	50	100	10	50	100	10	50	100
CORE	8.5	8.7	8.7	7.25	7.30	7.30	12.1	13.0	13.4
SASRec	25.2	27.4	27.9	22.5	22.9	23.0	**34.1**	**43.8**	**46.6**
BERT4Rec	**26.4**	**27.8**	**28.2**	**23.9**	**24.2**	**24.3**	33.9	40.4	42.9
GRU4Rec	0.08	0.12	0.14	0.07	0.07	0.08	0.14	0.31	0.43

Table 3: OOD User - Temporally Aligned Evaluation (N=NDCG, M=MRR, R=Recall)

Baseline	N			M			R		
	10	50	100	10	50	100	10	50	100
CORE	5.9	6.8	7.2	4.19	4.39	4.43	11.1	15.4	17.9
SASRec	**17.8**	**19.2**	**19.6**	**15.2**	**15.5**	**15.5**	**26.2**	**32.2**	34.6
BERT4Rec	11.8	14.4	15.2	9.96	10.50	10.58	17.8	29.5	**34.7**
GRU4Rec	0.01	0.01	0.02	0.004	0.004	0.005	0.01	0.03	0.08

Table 4: In-Distribution User - Temporal Extrapolation Evaluation (N=NDCG, M=MRR, R=Recall)

Baseline	N			M			R		
	10	50	100	10	50	100	10	50	100
CORE	0.09	0.47	0.75	0.04	0.11	0.13	0.26	2.10	3.78
SASRec	**2.9**	**3.6**	3.9	**1.88**	**2.03**	**2.05**	**6.2**	9.4	11.0
BERT4Rec	1.1	3.2	**4.0**	0.56	0.99	1.10	2.8	**12.8**	**17.8**
GRU4Rec	0.01	0.01	0.02	0.004	0.005	0.01	0.01	0.03	0.08

Table 5: OOD User - Temporal Extrapolation Evaluation (N=NDCG, M=MRR, R=Recall)

Baseline	N			M			R		
	10	50	100	10	50	100	10	50	100
CORE	0.10	0.53	0.82	0.04	0.12	0.15	0.32	2.33	4.13
SASRec	**3.1**	**3.9**	**4.1**	**2.01**	**2.17**	**2.19**	**6.7**	9.9	11.6
BERT4Rec	1.1	3.4	4.3	0.55	1.02	1.10	2.8	**13.7**	**18.9**
GRU4Rec	0.01	0.01	0.02	0.004	0.004	0.005	0.01	0.04	0.07

mendation settings remain an open research problem and present unique challenges in our context given the extremely long-tailed item distribution. Nevertheless, we include standard fine-tuning baselines (PEFT and full fine-tuning) for completeness.

To ensure the integrity of comparisons, we avoid merging the in-distribution (IND) test set used in Task 1 with this OOD evaluation pool, as we use the same set for over fine-tuning baselines. We evaluate three recent and publicly available language models up to 9B parameter scale: LLAMA-3.1-8B (Grattafiori et al., 2024), QWEN3-8B (Yang et al., 2025), and GEMMA2-9B (Team et al., 2024). All models are queried in a zero-shot manner using a standardized prompt for each task.

For encoding the items and queries, we use the pre-trained BLAIR item encoder (Hou et al., 2024) as it is pre-trained on the Amazon-Reviews items and the FAISS library (Douze et al., 2024) for creating the ANN-based vector databases to perform retrieval. As we do not perform ranking across queries, we compute standard retrieval metrics i.e. RECALL@K and PRECISION@K for $K = 10, 50, 100$ to assess the effectiveness of the generated outputs in retrieving relevant items. [6]

6 Results & Discussion

6.1 Benchmarking traditional ID-based baselines

Tables 2 to 5 demonstrate the performance of traditional ID-based baselines across both In-Distribution as well as of OOD user settings across both temporal setups.

Challenging Nature of the Task Our task formulation is substantially more challenging than prior work such as (Hou et al., 2024). Unlike settings that train and evaluate on narrow category-wise domains and subsets (eg: Beauty), we train on the full distribution of user activity spanning diverse product categories, aiming to predict the next plausible item in a multi-domain environment. While non-attention-based models like GRU4REC (Hidasi et al., 2016) have shown strong results in simpler setups (Betello et al., 2025), we find that they struggle in our broader and more complex setting. In contrast, attention-based models such as BERT4REC, SASREC perform markedly better, underscoring the importance of flexible context modeling.

Is traditional in-distribution leave-one-out evaluation sufficient? Standard evaluation protocols in recommendation typically adopt an in-distribution leave-one-out setting, where a user's next interaction—already seen during training—is held out as the target. As shown in Table 2, models achieve strong performance under this setup. However, in our more realistic out-of-distribution (OOD) evaluation, where test users are entirely unseen during training, we observe **a significant performance drop across all methods**. This highlights that in-distribution protocols may overestimate generalization and motivates the need for OOD-based evaluation to better assess true model robustness.

Is model performance stable on OOD users? Despite the degradation observed in OOD evalu-

[6] Detailed hyperparameter settings, prompts and implementation details are provided in the Appendix.

Table 6: LLM-based Query Reformulation

Model	Recall			Precision		
	@10	@50	@100	@10	@50	@100
LLAMA-3.1-8B	1.62	2.37	2.84	0.20	0.23	0.22
Qwen3-8B	**2.06**	**2.95**	**3.50**	**0.25**	**0.28**	**0.28**
Gemma2-9B	1.45	2.26	2.66	0.16	0.21	0.19

Table 7: Long-Horizon User Modeling

Model	Recall			Precision		
	@10	@50	@100	@10	@50	@100
LLAMA-3.1-8B	1.26	6.52	13.25	0.51	0.52	0.53
Qwen3-8B	**1.51**	**7.78**	**15.75**	**0.63**	**0.65**	**0.66**
Gemma2-9B	0.98	5.07	10.39	0.42	0.43	0.44

ation (Table 3), attention-based models retain relatively high accuracy. This indicates that shared behavioral patterns across users; especially within the same temporal window can still be leveraged, consistent with collaborative filtering principles. While OOD evaluation poses a harder challenge, these results suggest that robust sequence modeling can partially bridge the generalization gap.

Impact of temporal distributional shifts As shown in Tables 4 and 5, temporal shifts result in significant performance degradation across all models, irrespective of whether the user was seen during training. Notably, models generalize better to unseen users from the same time period than to the same users across different time periods. We attribute this to the reliance on ID-based representations, which lack semantic grounding. Consequently, models struggle to adapt to new items, such as those from emerging brands, due to an absence of similarity encoding with previously seen items. This highlights the need for incorporating textual or semantic features to enhance robustness.

6.2 Benchmarking LLM-based Query Reformulation for Recommendation

Table 6 summarizes the evaluation of three prominent LLMs i.e. LLAMA-3.1-8B, Qwen3-8B, and Gemma2-9B on their ability to reformulate user queries for item retrieval. The results indicate modest performance in both Recall and Precision, with improvements as the number of recommendations increases from 10 to 100. This trend suggests that LLMs capture some relevant items within larger candidate sets, reflecting a partial understanding of broader user intent. Nonetheless, absolute Recall values remain relatively low, indicating challenges in consistently retrieving a substantial portion of truly relevant items. Among the models, Qwen3-8B consistently outperforms LLAMA-3.1-8B and Gemma2-9B.

To further evaluate query quality, we measured semantic similarity between reformulated queries and ground-truth items using BLAIR embeddings. The average cosine similarity scores are approximately 0.73 for Qwen3-8B, 0.72 for Gemma2-9B, and 0.71 for LLAMA-3.1-8B. These moderate similarities indicate that, while the queries capture reasonable semantic relatedness, there remains scope to generate better quality reformulations.

We also evaluate reasoning models like Qwen3-235B in both reasoning and non-reasoning modes to understand if **Large Reasoning Models** can be an effective solution. We observe a **Recall@100 of 2.96** in Reasoning mode and **3.4** (Non-Reasoning mode), which is comparable in performance with the Qwen3-8B model, suggesting model scaling or reasoning may not be very effective currently.

Lastly, we conducted **LLM fine-tuning experiments** using parameter-efficient (LoRA) (Hu et al., 2022) and full fine-tuning approaches with LLaMA-3.1-8B and Qwen3-8B models. Fine-tuned models are comparable to the zero-shot setting on next-item recommendation (best of 10 performance) (see Tables 6 and 10), with the zero-shot approach being more scalable for growing catalogs. We elaborate on these results in Appendix E.

6.3 Benchmarking Results on Long-Horizon settings

Table 7 shows how LLMs perform on the challenging task of long-horizon user modeling, where models generate high-level summaries of future interests that are mapped to catalog items. While Recall improves with higher k, indicating some relevance capture over long sequences, Precision remains low, reflecting a high rate of irrelevant predictions. Among the models, Qwen3-8B consistently outperforms LLAMA-3.1-8B and Gemma2-9B. When comparing with the Query Reformulation task (Table 6), it's important to note that long-horizon evaluation benefits from multiple ground-truth targets, unlike the single-reference setup in query reformulation. This may overstate long-horizon performance, even though it requires modeling deeper preference evolution. Furthermore, prior work often relaxes strict ordering in long-horizon evaluation (Pancha et al., 2022; Zhou et al., 2024), further complicating direct metric comparisons. Thus, evaluations across these tasks should be interpreted

with care, given their differing objectives.

7 Conclusion

User behavior modeling is central to modern personalized systems, yet existing benchmarks and evaluation protocols fall short of capturing the generalization challenges faced in real-world deployments. In this work, we introduced HORIZON, a benchmark designed to evaluate sequential recommendation models under realistic long-horizon, temporal, and user-level distribution shifts.

HORIZON defines five evaluation settings that systematically disentangle in-distribution performance, temporal extrapolation, and generalization to unseen users. Through extensive experiments across multiple model families, we demonstrate that model performance varies substantially across these settings, exposing limitations that are obscured by conventional evaluation practices.

Our findings highlight the need for evaluation frameworks that move beyond short-horizon next-item accuracy and explicitly measure robustness under realistic generalization scenarios. By providing a large-scale, cross-domain benchmark and a principled evaluation protocol, HORIZON aims to support the development of more robust user modeling methods for modern personalized platforms.

8 Limitations

Despite the comprehensive nature of our benchmark, we acknowledge several limitations in our current work. Our benchmark is currently restricted to English-only data, representing an important opportunity for future extension to multilingual contexts. Additionally, the dataset is limited to textual modalities, though expanding to multimodal data would significantly enhance its applicability for multimodal user modeling tasks that integrate visual / auditory information. For our experiments using the RecBole framework, we were limited to using only a subset of the complete dataset for training and evaluation due to computational resource constraints. Lastly, the scope of our experimentation is restricted to the e-commerce setting. However, our evaluation framework is extensible to any arbitrary domain.

9 Acknowledgements

We sincerely thank Yashoteja Prabhu (Microsoft Research) and Medha Hira (Carnegie Mellon University) for their insightful discussions and generous review of this draft, which helped us organize our ideas more effectively and improve its overall readability.

References

Avinash Anand, Arnav Goel, Medha Hira, Snehal Buldeo, Jatin Kumar, Astha Verma, Rushali Gupta, and Rajiv Ratn Shah. 2023. Sciphyrag - retrieval augmentation tonbsp;improve llms onnbsp;physics q a. In *Big Data and Artificial Intelligence: 11th International Conference, BDA 2023, Delhi, India, December 7–9, 2023, Proceedings*, page 50–63, Berlin, Heidelberg. Springer-Verlag.

Filippo Betello, Antonio Purificato, Federico Siciliano, Giovanni Trappolini, Andrea Bacciu, Nicola Tonellotto, and Fabrizio Silvestri. 2025. A reproducible analysis of sequential recommender systems. *IEEE Access*, 13:5762–5772.

Eunjoon Cho, Seth A. Myers, and Jure Leskovec. 2011. Friendship and mobility: user movement in location-based social networks. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11, page 1082–1090, New York, NY, USA. Association for Computing Machinery.

Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. 2024. The faiss library. *arXiv preprint arXiv:2401.08281*.

Arnav Goel, Medha Hira, Avinash Anand, Siddhesh Bangar, and Rajiv Ratn Shah. 2023. Advancements in scientific controllable text generation methods. *Preprint*, arXiv:2307.05538.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

F Maxwell Harper and Joseph A Konstan. 2015a. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)*, 5(4):1–19.

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

Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based recommendations with recurrent neural networks. *Preprint*, arXiv:1511.06939.

Yupeng Hou, Binbin Hu, Zhiqiang Zhang, and Wayne Xin Zhao. 2022a. Core: Simple and effective session-based recommendation within consistent

representation space. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '22, page 1796–1801, New York, NY, USA. Association for Computing Machinery.

Yupeng Hou, Jiacheng Li, Zhankui He, An Yan, Xiusi Chen, and Julian McAuley. 2024. Bridging language and items for retrieval and recommendation. *arXiv preprint arXiv:2403.03952*.

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

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

Yitong Ji, Aixin Sun, Jie Zhang, and Chenliang Li. 2023. A critical study on data leakage in recommender system offline evaluation. *ACM Transactions on Information Systems*, 41(3):1–27.

Wei Jin, Haitao Mao, Zheng Li, Haoming Jiang, Chen Luo, Hongzhi Wen, Haoyu Han, Hanqing Lu, Zhengyang Wang, Ruirui Li, Zhen Li, Monica Cheng, Rahul Goutam, Haiyang Zhang, Karthik Subbian, Suhang Wang, Yizhou Sun, Jiliang Tang, Bing Yin, and Xianfeng Tang. 2023a. Amazon-m2: a multilingual multi-locale shopping session dataset for recommendation and text generation. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

Wei Jin, Haitao Mao, Zheng Li, Haoming Jiang, Chen Luo, Hongzhi Wen, Haoyu Han, Hanqing Lu, Zhengyang Wang, Ruirui Li, and 1 others. 2023b. Amazon-m2: A multilingual multi-locale shopping session dataset for recommendation and text generation. *Advances in Neural Information Processing Systems*, 36:8006–8026.

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

Janak Kapuriya, Anwar Shaikh, Arnav Goel, Medha Hira, Apoorv Singh, Jay Saraf, Sanjana, Vaibhav Nauriyal, Avinash Anand, Zhengkui Wang, and Rajiv Ratn Shah. 2025. Enhancing scientific visual question answering via vision-caption aware supervised fine-tuning. In *Proceedings of the 2nd International Workshop on Large Vision - Language Model Learning and Applications*, LAVA '25, page 13–30, New York, NY, USA. Association for Computing Machinery.

Anton Klenitskiy, Anna Volodkevich, Anton Pembek, and Alexey Vasilev. 2024. Does it look sequential? an analysis of datasets for evaluation of sequential recommendations. In *Proceedings of the 18th ACM Conference on Recommender Systems*, RecSys '24, page 1067–1072, New York, NY, USA. Association for Computing Machinery.

Huihan Li, Arnav Goel, Keyu He, and Xiang Ren. 2025. Attributing culture-conditioned generations to pretraining corpora. *Preprint*, arXiv:2412.20760.

Tzu-Heng Lin, Chen Gao, and Yong Li. 2019. Cross: Cross-platform recommendation for social e-commerce. In *Proceedings of the 42nd International ACM SIGIR conference on research and development in information retrieval*, pages 515–524.

Duncan McElfresh, Sujay Khandagale, Jonathan Valverde, John Dickerson, and Colin White. 2022. On the generalizability and predictability of recommender systems. *Advances in Neural Information Processing Systems*, 35:4416–4432.

Zaiqiao Meng, Richard McCreadie, Craig Macdonald, and Iadh Ounis. 2020. Exploring data splitting strategies for the evaluation of recommendation models. In *Proceedings of the 14th ACM Conference on Recommender Systems*, RecSys '20, page 681–686, New York, NY, USA. Association for Computing Machinery.

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.

Nikil Pancha, Andrew Zhai, Jure Leskovec, and Charles Rosenberg. 2022. Pinnerformer: Sequence modeling for user representation at pinterest. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '22, page 3702–3712, New York, NY, USA. Association for Computing Machinery.

Steffen Rendle, Walid Krichene, Li Zhang, and John Anderson. 2020. Neural collaborative filtering vs. matrix factorization revisited. In *Proceedings of the 14th ACM Conference on Recommender Systems*, pages 240–248.

Antoni Sobkowicz and Wojciech Stokowiec. 2016. Steam review dataset - new, large scale sentiment dataset.

Aixin Sun. 2023. Take a fresh look at recommender systems from an evaluation standpoint. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2629–2638.

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*, CIKM '19, page 1441–1450, New York, NY, USA. Association for Computing Machinery.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, and 179 others. 2024. Gemma 2: Improving open language models at a practical size. *Preprint*, arXiv:2408.00118.

Ben Treves, Emiliano De Cristofaro, Yue Dong, and Michalis Faloutsos. 2025. Viki: Systematic cross-platform profile inference of online users. *arXiv preprint arXiv:2503.14772*.

Yuling Wang, Changxin Tian, Binbin Hu, Yanhua Yu, Ziqi Liu, Zhiqiang Zhang, Jun Zhou, Liang Pang, and Xiao Wang. 2024. Can small language models be good reasoners for sequential recommendation? In *Proceedings of the ACM Web Conference 2024*, pages 3876–3887.

Fangzhao Wu, Ying Qiao, Jiun-Hung Chen, Chuhan Wu, Tao Qi, Jianxun Lian, Danyang Liu, Xing Xie, Jianfeng Gao, Winnie Wu, and Ming Zhou. 2020a. MIND: A large-scale dataset for news recommendation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3597–3606, Online. Association for Computational Linguistics.

Fangzhao Wu, Ying Qiao, Jiun-Hung Chen, Chuhan Wu, Tao Qi, Jianxun Lian, Danyang Liu, Xing Xie, Jianfeng Gao, Winnie Wu, and 1 others. 2020b. Mind: A large-scale dataset for news recommendation. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 3597–3606.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

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*.

Wayne Xin Zhao, Yupeng Hou, Xingyu Pan, Chen Yang, Zeyu Zhang, Zihan Lin, Jingsen Zhang, Shuqing Bian, Jiakai Tang, Wenqi Sun, Yushuo Chen, Lanling Xu, Gaowei Zhang, Zhen Tian, Changxin Tian, Shanlei Mu, Xinyan Fan, Xu Chen, and Ji-Rong Wen. 2022. Recbole 2.0: Towards a more up-to-date recommendation library. *arXiv preprint arXiv:2206.07351*.

Wayne Xin Zhao, Shanlei Mu, Yupeng Hou, Zihan Lin, Yushuo Chen, Xingyu Pan, Kaiyuan Li, Yujie Lu, Hui Wang, Changxin Tian, Yingqian Min, Zhichao Feng, Xinyan Fan, Xu Chen, Pengfei Wang, Wendi Ji, Yaliang Li, Xiaoling Wang, and Ji-Rong Wen. 2021. Recbole: Towards a unified, comprehensive and efficient framework for recommendation algorithms. In *CIKM*, pages 4653–4664. ACM.

Zhihan Zhou, Qixiang Fang, Leonardo Neves, Francesco Barbieri, Yozen Liu, Han Liu, Maarten W Bos, and Ron Dotsch. 2024. Use: Dynamic user modeling with stateful sequence models. *arXiv preprint arXiv:2403.13344*.

A HORIZON Statistics and Plots

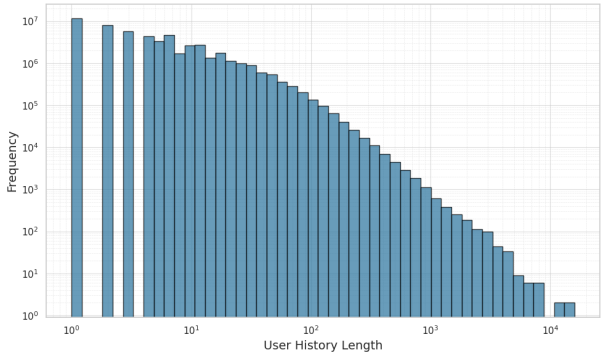

Figure 3: Histogram Depicting the Frequency Distribution of User History Lengths in HORIZON. The presence of ultra-long user histories highlights the need for architectures capable of modeling long-range sequential dependencies.

The HORIZON benchmark is curated by reformulating the widely-used Amazon Reviews 2023 dataset (Hou et al., 2024), merging all 33 categories into unified user histories to enable robust long-term, cross-domain user modeling. This section provides an in-depth statistical analysis of the dataset through visualizations and derived insights. **Scale and Diversity:** The benchmark comprises approximately **53.5M users and 34.5M unique items**, amounting to nearly 486M interaction records. This scale is significantly larger than prior public benchmarks and captures highly diverse behavioral patterns. With the unified formulation, user histories naturally span multiple product categories—introducing heterogeneous context that is both semantically diverse and temporally rich. This setting reflects real-world personalization chal-

lenges more faithfully than isolated category-based modeling.

User History Lengths: Figure 3 illustrates a long-tailed distribution of user history lengths in HORIZON. While a large portion of users exhibit short interaction sequences, there exists a substantial number with extremely long histories—extending beyond 1,000 timestamps for tens of thousands of users. This highlights the need for models capable of handling long-range dependencies and memory-efficient representations. Traditional sequence models struggle in this regime due to vanishing gradients and computational bottlenecks, motivating the exploration of transformer-based or memory-augmented architectures for this benchmark.

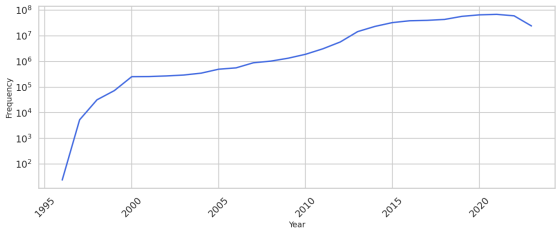

Figure 4: Line Plot Depicting the Temporal Distribution of User Histories in HORIZON. The balanced volume before and after 2020 makes it suitable for temporal extrapolation tasks.

Temporal Structure and Generalization: The temporal distribution of interactions (Figure 4) reveals a sharp rise in user activity post-2010, peaking around 2020. Crucially, nearly half the interactions occur after the 2020 temporal cut-off used in our evaluation framework. Specifically, the average number of timestamps before 2020 is 4.99, while it remains comparable after 2020 at 4.09. This temporal balance ensures that both training and test splits are adequately rich, setting up a robust testbed for extrapolative evaluation and temporal generalization. As models are evaluated on unseen user interactions post-2020, they are challenged to infer future behavior patterns from past, potentially outdated, preferences—mirroring real-world drift in user intent.

Product Distribution: Figure 5 plots the frequency distribution of product IDs, which exhibits a pronounced long-tail trend. A small fraction of items dominate interactions, while the majority are sparsely interacted with. This reflects typical e-commerce dynamics but poses unique challenges for recommender systems: most prior models are biased toward frequent items. The high item cardinality (34M) and sparse tail necessitate models that generalize well to rarely seen or previously unseen products. Incorporating textual features or content-based augmentations could be beneficial in this context, especially under cold-start settings.

Benchmark Design Implications: The three key observations from these plots underscore the difficulty of the HORIZON benchmark:

1. **Long Histories:** Users with thousands of interaction points require models that capture dependencies over extended horizons and adapt across evolving interests.

2. **Temporal Drift:** A significant portion of test data lies beyond the training horizon (post-2020), enforcing extrapolation beyond the training distribution and testing robustness to temporal shifts.

3. **Item Sparsity:** The skewed product frequency implies that many test items are low-frequency or unseen, further intensifying the generalization challenge.

Taken together, HORIZON enables a comprehensive stress test of user behavior models across multiple axes—scale, history length, temporal generalization, and sparsity. Its unified multi-category formulation fosters the development of general-purpose, temporally robust, and cross-domain recommendation architectures.

B Task 1 Splits and Out-of-Distribution Analysis

In our proposed Task 1 setup, the user population is explicitly partitioned into two cohorts to rigorously test generalization: *in-distribution (IND)* users observed during training, and *out-of-distribution (OOD)* users who are entirely held out. The fixed temporal cutoff at $\tau = 2020$ allows us to decouple user generalization from temporal extrapolation. Below, we elaborate on the statistical and structural distinctions between these cohorts, which underline the difficulty of the proposed evaluation.

Temporal Shift and Behavioral Drift: As visualized in Figure 4, a significant volume of user activity in the dataset occurs post-2020. By construction, OOD users are sampled from this post-2020 pool, whereas IND users have interactions both before and after the temporal boundary. This creates a

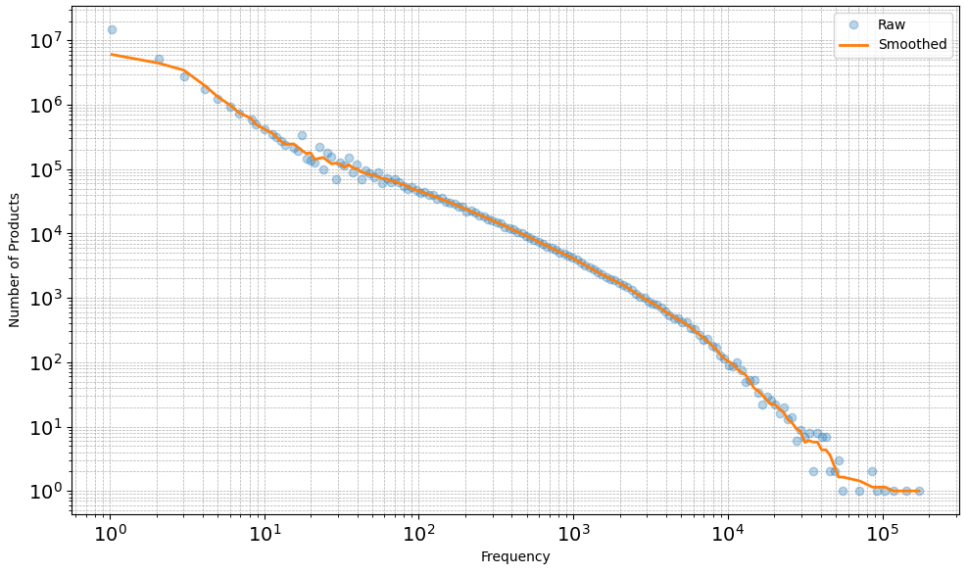

Figure 5: Frequency Distribution of Products in the HORIZON Benchmark. The power-law structure reflects extreme item sparsity, with most items having very few interactions.

natural distributional shift: the OOD cohort is inherently more recent and behaviorally different, reflecting newer products, evolving user preferences, and potentially different session structures. Hence, even under temporally aligned evaluation (Subtask 1c), the OOD test set exhibits non-trivial variance from the training distribution.

Semantic Divergence via Topic Modeling. To investigate the semantic distinctiveness between in-distribution (IND) and out-of-distribution (OOD) user groups, we apply Latent Dirichlet Allocation (LDA) to model topics from user review histories, treating each user as a document composed of concatenated item descriptions and metadata. The resulting topic distributions uncover meaningful divergence in user interests.

Both groups engage with broad product themes (e.g., books, electronics, fashion), yet OOD users demonstrate stronger focus on niche and emergent categories. For example, OOD-specific topics include terms like *"telescope," "kite," "bjj," "freemason,"* and musical instruments such as *"guitar," "ukulele," "pedal"*, suggesting a shift toward specialized or subcultural interests.

In contrast, IND topics reflect more mainstream and diversified engagement, including wellness supplements (e.g., *"nootropic," "creatine," "arginine"*) and general home goods. To quantify these patterns, we compute entropy and dominance over user topic distributions. OOD users show significantly lower entropy (mean = 1.18 vs. 1.28) and higher topic dominance (mean = 0.51 vs. 0.48), indicating more focused topical preferences. A t-SNE projection of user topic vectors reveals clear separation between IND and OOD clusters.

Additionally, the average KL divergence from IND to OOD topic distributions exceeds 0.8, reinforcing the semantic shift. These findings suggest that OOD generalization reflects not just temporal drift but substantive thematic changes in user behavior and product engagement.

C Cross-domain Statistics and Distribution Shifts

Unified cross-domain user histories. Unlike category-segmented or short-timespan variants of Amazon Reviews, HORIZON constructs fully unified user interaction sequences spanning all 35M items across diverse product domains. This design removes artificial boundaries between categories, allowing models to observe and learn from naturally interleaved behaviors (Section 3.1, Table 1). In practice, real-world users frequently transition between domains—e.g., electronics, books, and household items—within a single behavioral trajectory. By preserving this structure, HORIZON enables the study of cross-domain preference transfer, long-range dependencies, and heterogeneous context modeling at unprecedented scale. Such evaluation is fundamentally infeasible in existing benchmarks such as MovieLens, MIND, or session-based datasets, which are either restricted to a single domain or limited to short temporal windows

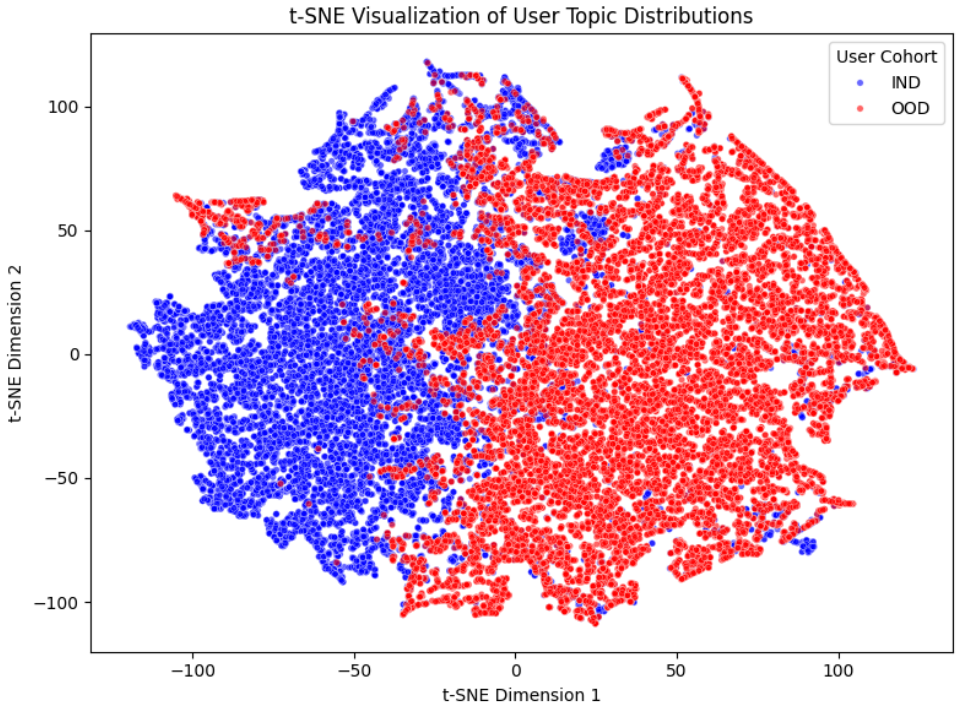

Figure 6: t-SNE depicting the distinct user topic distributions in the in-distribution and OOD users.

that truncate long-term behavioral signals.

Cross-domain user behavior under distribution shift. A defining characteristic of HORIZON is the prevalence of multi-domain user activity. On average, users interact with approximately six distinct domains, and over 90% of users exhibit cross-domain behavior. This diversity becomes even more pronounced in the out-of-distribution (OOD) split, which is explicitly constructed to reflect more complex and heterogeneous user patterns. In particular, only 1% of users in the OOD split remain single-domain, compared to 11% in the training set. This shift indicates that models must generalize from relatively simpler, less diverse training trajectories to significantly richer and more entangled behavioral patterns at test time. As a result, HORIZON naturally induces a distribution shift not only in item space but also in user behavior complexity, requiring models to capture transferable representations across domains rather than relying on narrow, domain-specific signals.

Item novelty, sparsity, and temporal dynamics. The benchmark further introduces realistic challenges through a combination of item sparsity, temporal spread, and catalog evolution. OOD users tend to have sparse yet long-horizon interaction histories, with 63% exhibiting between 6 and 20 interactions distributed across multiple years. This creates a setting where models must reason over temporally distant signals while operating under limited per-user data. Additionally, item exposure in the OOD split reflects significant novelty: 33% of test interactions involve rare or tail items, nearly doubling the 19% observed during training. Only 60% of candidate items overlap with the training catalog, indicating substantial item turnover and the presence of previously unseen or underrepresented items at inference time. Together, these factors simulate realistic recommendation environments where user preferences evolve, item distributions shift, and long-tail discovery becomes critical—conditions that are largely absent in prior benchmarks such as MIND, Amazon-M2, and category-partitioned Amazon Reviews.

D Hyperparameters and Implementation Details

D.1 RecBole Experiments - Task 1

All models in Task 1 were trained using the RecBole framework (Zhao et al., 2021, 2022) with a consistent configuration to ensure a fair comparison. The common training hyperparameters were selected based on prior literature and empirical tuning on a held-out validation set. These include a small learning rate of 2×10^{-5} to stabilize optimiza-

Table 8: Model-specific hyperparameter configurations

Parameter	BERT4Rec	GRU4Rec	SASRec	CORE
Hidden/Embedding size	256	256	256	256
Number of layers	3	3	3	3
Attention heads	4	-	4	4
Dropout probability	0.15	0.15	0.15	0.15
Batch size	8192	8192	4096	4096
Loss function	BPR	BPR	CE	CE
Mask ratio	0.2	-	-	-

tion over long sequences, a maximum of 10 epochs for training, and early stopping with a patience of 10 epochs to prevent overfitting. To ensure reproducibility across all experimental runs, we fixed the random seed to 2025.

Training Hyperparameters. All models were trained with the following consistent configuration

- **Learning rate**: 2×10^{-5}
- **Maximum epochs**: 10
- **Early stopping patience**: 10
- **Random seed**: 2025
- **Maximum sequence length**: 100
- **Validation metric**: MRR@10
- **Evaluation cutoffs**: $k \in \{10, 20, 50, 100\}$
- **Test negative samples**: 100

To support uniform evaluation across models, we truncated all user interaction sequences to a maximum of 100 items and used mean reciprocal rank at cutoff 10 (MRR@10) as the primary validation metric. During testing, we sampled 100 negative items for each user-item query to simulate realistic top-k recommendation settings and report metrics at various cutoffs (k).

Model-Specific Hyperparameters Each model was configured using a 256-dimensional embedding and three layers to capture higher-order dependencies. Attention-based models (BERT4Rec, SASRec, and CORE) used 4 attention heads to balance modeling capacity and memory cost. A dropout rate of 0.15 was applied to all models for regularization. Batch sizes were tuned based on GPU memory availability and empirical training stability: 8192 for BERT4Rec and GRU4Rec, and 4096 for SASRec and CORE due to their higher per-batch memory footprint. These are further detailed in Table 8.

Architecture Details: Given below are the architectural details about the RecBole baselines which we have employed in our study on the HORIZON benchmark:

- **BERT4Rec**: It leverages bidirectional Transformers to model sequence-wide context and predicts masked items using a masked language modeling (MLM) objective, with a mask ratio set to 0.2.
- **GRU4Rec**: GRU4Rec uses gated recurrent units (GRUs) to model sequential dependencies.
- **SASRec**: SASRec is built on unidirectional self-attention layers, enabling it to capture short- and long-term dependencies without recurrence.
- **CORE**: CORE integrates self-attention with collaborative filtering signals, enhancing personalization through a hybrid architecture

Loss Function Configuration. Given below are the possible loss function configurations available in RecBole for training sequential recommendation models:

- **BPR models** (BERT4Rec, GRU4Rec, SASRec): Bayesian Personalized Ranking with negative sampling during training
- **CE models** (CORE): Cross-entropy loss without negative sampling during training

Models trained with BPR loss (BERT4Rec, GRU4Rec, SASRec) rely on dynamic negative sampling and optimize the ranking of positive over negative interactions. In contrast, CORE optimizes a classification objective using cross-entropy loss computed over the full softmax distribution.

Table 9: Hyperparameters used for different models.

Hyperparameter	LLAMA-3.1-8B	QWEN3-8B	GEMMA2-9B
Batch Size	512	512	256
Temperature	0.7	0.7	0.7
Top-P	0.8	0.8	0.8
Top-K	20	20	20
Max-Tokens (Task 2)	220	220	220
Max-Tokens (Task 3)	350	350	350

Execution Details. All experiments were conducted using a high-performance compute cluster equipped with 4 NVIDIA A100 GPUs (80GB VRAM each). We employed PyTorch's automatic mixed precision (AMP) to accelerate training and reduce memory usage. Training time per epoch varied with architectural complexity: GRU4Rec, being lightweight, completed one epoch in approximately 0.75 hours, while BERT4Rec, with its attention-heavy encoder and MLM objective, required around 1.25 hours per epoch. Multi-GPU training was implemented using the NCCL backend for synchronized distributed training. All hyperparameters and implementation choices were fixed across all splits to ensure experimental consistency and comparability.

D.2 Task 2 and 3 Experiments

LLM Inference Setup. We adopt a consistent inference pipeline for both Task 2 (LLM-based Next Product Recommendation via Query Reformulation) and Task 3 (LLM-based Long-Horizon User Modeling), as described in Section 5 and illustrated in Figure 2. All models are prompted in a zero-shot setting, without any fine-tuning or retrieval augmentation, to evaluate their general-purpose reasoning capabilities over long user histories and contexts.

We utilize three state-of-the-art, instruction-tuned open-source LLMs: LLAMA-3.1-8B (Grattafiori et al., 2024), QWEN3-8B (Yang et al., 2025), and GEMMA2-9B (Team et al., 2024). These models were selected for their strong instruction-following capabilities and competitive performance on public benchmarks.

Table Table 9 summarizes the decoding hyperparameters used. The temperature was fixed at 0.7 across all models to balance determinism and diversity in outputs. We set Top-P and Top-K sampling parameters based on model-specific best practices to control generation randomness. The maximum token limits were adjusted per task: 220 tokens for Task 2 (shorter search queries), and 350 tokens for Task 3 (longer next-item descriptions). Batch sizes were selected based on each model's memory footprint and throughput on A100 GPUs, with the larger GEMMA2-9B model using a smaller batch size.

Execution Details. All inference was run using the vLLM engine on a compute cluster with 4× NVIDIA A100 40GB GPUs. The full test set consists of 1 million users, with each user processed independently in batched decoding mode. End-to-end inference across all models required approximately 5 days due to the volume of input prompts and the autoregressive nature of generation.

To support reproducibility and accessibility, we will release all evaluation code, prompt templates, and precomputed predictions on smaller held-out test splits post-acceptance. These subsets will enable low-resource experimentation on the same evaluation protocol without requiring access to large-scale GPU compute.

Generating Query and Item Embeddings using BLAIR. To encode the item catalog and predicted queries, we leverage the BLAIR item encoder (Hou et al., 2024), a RoBERTa-based model pretrained on Amazon review titles. We use the `hyp1231/blair-roberta-base` checkpoint via the HuggingFace Transformers library [7], and tokenize each product title with a maximum sequence length of 512 tokens. Embeddings are obtained by extracting the `[CLS]` token representation from the final hidden layer, followed by ℓ_2 normalization to facilitate cosine similarity-based retrieval. To scale embedding computation across a large number of titles, we utilize the `Accelerate` library with mixed-precision inference (fp16) and distributed processing across

[7] https://huggingface.co/hyp1231/blair-roberta-base

Table 10: Comparison of Fine-tuned LLMs for Next-Item Prediction

Setting	Recall@K (%)			Precision@K (%)		
	FFT (LLaMA3)	LoRA (LLaMA3)	LoRA (Qwen3)	FFT (LLaMA3)	LoRA (LLaMA3)	LoRA (Qwen3)
In-Domain Temporal Extrapolation (Task 1c)						
K=10	1.45	**1.65**	1.38	0.98	**1.29**	0.90
K=50	1.67	**1.82**	1.60	0.97	**1.28**	0.90
K=100	2.02	**2.09**	1.93	0.97	**1.28**	0.89
Out-of-Domain Temporal Extrapolation (Task 1d)						
K=10	**1.24**	0.71	1.18	**0.82**	0.42	0.77
K=50	**1.41**	0.84	1.37	**0.81**	0.42	0.77
K=100	**1.71**	1.07	1.67	**0.80**	0.42	0.76

multiple GPUs, achieving efficient batch-wise encoding with a batch size of 4096. We shard the workload across processes and later merge the outputs to form a single embedding matrix for the catalog and prediction sets.

Retrieval and Indexing using FAISS. For approximate nearest neighbor (ANN) search, we employ the FAISS library (Douze et al., 2024), which implements the Hierarchical Navigable Small World (HNSW) graph-based indexing algorithm. We build a HNSW index on the catalog embeddings using cosine similarity as the distance metric. The key hyperparameters used during index construction include: `M=64`, which controls the number of bi-directional links created for each new node and influences index accuracy and memory usage; and `efConstruction=256`, which sets the dynamic list size for the graph during construction and affects indexing time and final recall quality. At query time, we use `efSearch=256` to control the breadth of the search and balance between latency and retrieval performance. These values were selected based on a grid search over the validation set to optimize top-k recall, where $k = 10$, while ensuring sub-millisecond retrieval latency per query on a modern GPU setup.

This setup enables scalable, low-latency nearest neighbor search over millions of product titles, while maintaining semantic alignment between predicted queries and candidate items.

E LLM-Finetuning Results

What is the overall performance of fine-tuned LLMs? Table 10 reports the performance of fine-tuned LLMs for next-item prediction under temporal extrapolation, evaluated on both in-domain (Task 1c) and out-of-domain users (Task 1d). Across all configurations, performance remains uniformly low. In the in-domain setting, Recall@10 ranges between 1.38% and 1.65%, increasing only marginally to a maximum of 2.09% at Recall@100 for LoRA fine-tuned LLaMA-3.1-8B. Precision follows a similarly flat trend, peaking at 1.29%. Notably, increasing the cutoff from $K = 10$ to $K = 100$ yields only modest gains (e.g., 1.65% $\rightarrow$ 2.09% for LoRA LLaMA-3.1-8B), indicating that relevant items are not substantially concentrated even within larger candidate sets.

Across training strategies, parameter-efficient fine-tuning (LoRA) provides slight but consistent improvements over full fine-tuning (FFT) for LLaMA-3.1-8B in the in-domain setting, though the absolute gains remain small (e.g., +0.07 at Recall@100). Qwen3-8B exhibits comparable but slightly weaker performance overall. These results suggest that neither scaling parameter updates nor switching architectures meaningfully improves next-item prediction under this formulation.

How does performance change under distribution shift? Performance degrades consistently in the out-of-domain setting (Task 1d), where models must generalize to unseen users with more diverse and complex interaction patterns. Recall@10 drops from 1.65% (LoRA LLaMA-3.1-8B, in-domain) to 0.71% under OOD conditions, and Recall@100 decreases from 2.09% to 1.07%. A similar degradation is observed across all models, though the extent varies: FFT LLaMA-3.1-8B retains relatively stronger performance (1.71% at Recall@100), while LoRA-based adaptations exhibit larger drops. Precision mirrors this behavior, remaining below 0.82% across all configurations and dropping sharply for LoRA LLaMA-3.1-8B (0.42% across all K).

Another notable pattern is the reduced sensitivity to K in the OOD setting. For example,

LoRA LLaMA-3.1-8B increases only from 0.71% to 1.07% when moving from Recall@10 to Recall@100, compared to a larger relative gain in-domain. This suggests that under distribution shift, relevant items are not only harder to rank highly but also less likely to appear even in extended candidate lists. Overall, these results highlight the difficulty of extrapolating across both temporal shifts and user distributions when models are trained to generate a single next item from a large candidate space.

Do fine-tuned LLMs outperform simple baselines? Despite the additional training, fine-tuned models achieve performance comparable to our zero-shot retrieval-based baseline (Table 6). The retrieval approach generates ten semantically diverse queries per user without any task-specific optimization, yet matches the effectiveness of fine-tuned LLMs across both in-domain and OOD settings. Given the extremely low absolute recall values observed after fine-tuning, this parity is particularly striking. It suggests that instruction fine-tuning, when framed as single-item generation over extremely large and long-tailed item spaces, provides limited practical benefit. The consistency of this observation across architectures (LLaMA-3.1-8B and Qwen3-8B) and adaptation strategies further indicates that the limitation is not model-specific.

Why do fine-tuned LLMs struggle in this setting? A key limitation lies in the training formulation. Unlike discriminative sequential recommenders, which leverage contrastive supervision and large-scale negative sampling to structure the item space, instruction-tuned LLMs operate under a purely generative objective. This provides no explicit signal to distinguish among millions of plausible candidates or to calibrate ranked outputs. The weak scaling of recall with increasing K, together with uniformly low precision, indicates that models fail to meaningfully organize the candidate space, often producing semantically plausible but misaligned predictions. Consequently, fine-tuning does not fully exploit the semantic and representational strengths of LLMs in long-tailed recommendation settings.

What directions could address these limitations? These findings motivate alternative training paradigms that better align language modeling with recommendation objectives. Promising directions include contrastive or hybrid objectives with explicit negative sampling, as well as approaches that align model vocabularies with item identifiers for more direct reasoning over discrete item spaces. Additionally, multi-candidate generation or retrieval-augmented decoding may help bridge generative modeling with ranking-based evaluation, improving effectiveness under distribution shift.

F Prompts

F.1 Task 2: Query-Based Next-Item Recommendation.

As outlined in Section 4.2, this task evaluates an LLM's ability to generate personalized search queries from a user's Amazon product history. The prompt asks the model to produce 10 queries balancing relevance with serendipity. These queries act as soft proxies for next-item prediction and reveal how well the model generalizes user intent. The setup is zero-shot, requiring the model to function as a semantic encoder-decoder without fine-tuning.

```
PROMPT FOR TASK 2 - LLM-Based Next Item Recommendation:

You are an expert at turning a user's Amazon product
      history into personalized search queries.

History: I1 <SEP> I2 <SEP> ..... <In>
This was the users Amazon product history.

Your task is to generate a set of 10 personalized search
      queries that reflect the user's interests and
      preferences.
Try to balance diversity and serendipity with relevancy
      to the user history. These queries will be used to
      recommend the next product to the user.

Out of these 10 queries:
4 queries should be directly related to the user's
      history;
3 queries should be tangentially related;
3 queries should be completely unrelated but interesting.

Process:
1. Think of a guideline explaining what intents or
      aspects you observed in the user history which
      helped you formulate these queries. You don't need
      to specify which is which.
2. Then, generate exactly 10 search queries balancing
      core interests with a bit of serendipity.

## Output Format
Provide the response only as a JSON object with one field
      : (do not generate anything else)

{
  "queries": [
    "query1",
    "query2",
    "...",
    "query10"
  ]
}
```

F.2 Task 3

As described in Section 4.2, the following is the prompt for Task 3: Long-Horizon User Model-

ing using Large Language Models (LLMs). This task is designed to evaluate a model's ability to understand and extrapolate from a user's product history over time. The prompt guides the LLM to generate forward-looking, autoregressive item descriptions based on prior purchases, simulating realistic recommendation scenarios. Specifically, it instructs the model to infer underlying user preferences and behavioral patterns, and to generate coherent, temporally ordered predictions that balance relevance and serendipity. The prompt is framed in a zero-shot setting, encouraging the LLM to reason sequentially without access to explicit training examples.

```
PROMPT for Task 3 - LLM-Based Long-Horizon User Modeling:

You are an expert at predicting the next products a user
      may want based on their Amazon product history.

History: I1 <SEP> I2 <SEP> ..... <In>
This was the user's Amazon product history with exact
      product titles (NOT descriptions).

Your task is to generate descriptions for the next 10
      items the user is most likely to be interested in.
      Provide concise, onesentence descriptions that
      capture the essence of each potential item. These
      will guide recommendation generation.

Try to model the sequences in the user history and
      provide a mix of relevant and serendipitous items
      trying to capture the user's interests, intents and
       changes in behavior. Use the first item
      description to guide your next timestep's item
      description generation in an autoregressive manner.

Process:
1. Think of a guideline explaining the patterns or
      preferences you observed in the user history that
      informed your item descriptions.
2. Provide exactly 10 next-item descriptions balancing
      relevance and serendipity generated one after the
      other in temporal order.

## Output Format
Provide the response only as a JSON object with one field
      : (do not generate anything else)

{
  "item_descriptions_timewise": [
    "item_description_time_step1",
    "item_description_time_step2",
    "...",
    "item_description_time_step10"
  ]
}
```