PREX-REC: PROVENANCE-AWARE EXPERIENCE TRANSFER FOR SEQUENTIAL RECOMMENDATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Reusing a recommendation experience across users requires deciding whether its guidance fits the target user's behavior. Textual similarity alone does not explicitly encode the source interactions associated with that guidance. We propose PREX-Rec, which pairs each natural-language experience with LLM-selected items from its source history and uses these associations for retrieval. A user–item–experience relational index connects target histories to experiences through shared items, weighted by evidence frequency and recency. An LLM screens retrieved experiences for applicability; independent experience-guided rankings are then fused with a base ranking. On the evaluated Amazon Games and Arts candidate-ranking tasks, PREX-Rec improves all reported metrics over five LLM-based baselines. Games NDCG@5 increases from the strongest baseline's 0.2918 to 0.3291. With independent integration fixed, item-based retrieval improves over the evaluated vector retriever; perturbing experience–item assignments within source histories lowers Games NDCG@5 to 0.3086. These results support the utility of experience-specific item associations within the evaluated retrieval-and-ranking framework.

1 INTRODUCTION

Sequential recommendation predicts a user's next interaction from an ordered history. Recurrent and Transformer models learn sequential patterns (Hidasi et al., 2016; Kang & McAuley, 2018; Sun et al., 2019), while LLM-based approaches exploit item text, prompting, and retrieved context (Geng et al., 2022; Li et al., 2023; Hou et al., 2024; Lin et al., 2024). Beyond reasoning about one case at a time, recommendation systems can reuse guidance distilled from previous decisions. MoRE combines intra-user reflection and cross-user collaborative signals (Qin et al., 2025); MemRec maintains a collaborative memory graph (Chen et al., 2026). These advances motivate a more specific question: which previously extracted guidance should be retrieved for a new recommendation context?

A central difficulty is that textual relevance need not imply behavioral applicability. For example, two experiences about strategy games may concern different patterns: a stable preference for a genre or a recent shift toward it. Their wording can be similar even when their conditions of use differ. Conversely, shared interactions can connect users whose useful guidance is expressed differently. We investigate whether linking each experience to selected items from its source history supplies a useful retrieval signal for cross-user reuse.

We call these associations *item-level behavioral provenance*. An experience comprises an applicability condition, actionable guidance, and selected source-history items. The items are observed interactions; their assignment as support is an LLM judgment, not a causal attribution. This distinction separates two decisions: a source-case check tests whether the guidance helps on the case that produced it, while target-side selection assesses whether its condition fits another user's observed history.

PREX-Rec operationalizes this idea through a user–item–experience relational index. Target histories connect to experiences through exact shared items, and inverse-frequency and recency weights rank these connections. The retrieved objects are reusable recommendation instructions, rather than candidate items or aggregate user profiles. The contribution is the coupling of these instructions to experience-specific source items and the use of that coupling for retrieval. This explicit rela-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

tion makes the retrieval mechanism inspectable and can be implemented with an item-to-experience inverted index.

The distinction matters when a source user exhibits several preferences. Attaching the entire history to every instruction gives those instructions the same behavioral retrieval key. Experience-specific subsets instead allow two instructions from the same user to be reached through different interactions. The text states *what to do and under which condition*; the linked items indicate *which observed source behaviors to match*. These are complementary roles: the links propose relevant guidance, while the condition must still be checked against the target sequence.

After retrieval, the LLM selects applicable experiences and applies each in a separate ranking call. Deterministic fusion combines these rankings with an experience-independent base ranking. Our evaluation distinguishes the complete system's performance from evidence about its components: retrieval comparisons test the item-based signal against a specified vector retriever, within-source evidence perturbation tests the assignment of items to experiences, and removal of selected cross-user experiences tests their contribution within the deployed workflow.

Our contributions are summarized as follows:

- We formulate evidence-grounded recommendation experience reuse by associating natural-language guidance with selected source-history items, making item-level provenance available for cross-user retrieval.
- We develop PREX-Rec, an explicit relational retrieval framework with source-case filtering, target-side applicability selection, and independent experience-guided ranking. These stages distinguish source-case usefulness from target-context applicability.
- We evaluate the framework on two domains against five LLM-based baselines, with focused checks on a third domain and a second backbone. Existing ablations support the utility of the full retrieval-and-integration configuration and its experience-specific item associations.

2 Related Work

2.1 Sequential and LLM-Based Recommendation

Sequential recommenders model ordered interactions with recurrent, attention-based, temporal, and self-supervised methods (Hidasi et al., 2016; Kang & McAuley, 2018; Sun et al., 2019; Li et al., 2020; Zhou et al., 2020; Xie et al., 2022). Language-based methods use textual representations, prompting, and instruction alignment (Geng et al., 2022; Li et al., 2023; Dai et al., 2023; Sanner et al., 2023; Bao et al., 2023; Lyu et al., 2024; Hou et al., 2024). Other approaches combine LLMs with collaborative information (Kim et al., 2024; Ren et al., 2024), retrieval (Lin et al., 2024; Kim et al., 2026), or agentic reasoning (Wang et al., 2024; Zhang et al., 2024; Wang et al., 2026). Our study concerns experience-augmented LLM candidate ranking, complementing work on learned sequential representations and full-catalog retrieval.

2.2 Experience Reuse and Collaborative Memory

Reflexion, ExpeL, and MetaReflection reuse feedback or guidance from previous trials (Shinn et al., 2023; Zhao et al., 2024; Gupta et al., 2024); persistent memory systems retain information across interactions (Zhong et al., 2024; Tan et al., 2025). In recommendation, MoRE uses explicit, implicit, and cross-user reflectors together with reflection selection (Qin et al., 2025). MemRec manages and synthesizes a collaborative user/item memory graph for downstream recommendation (Chen et al., 2026). Building on experience reuse and cross-user collaboration, PREX-Rec focuses on a more specific retrieval object: an individual reusable instruction paired with selected source-history items. Those pairs provide explicit item-to-experience connections for subsequent users.

2.3 Structured Experience Retrieval

HippoRAG uses relational structure for associative retrieval (Gutiérrez et al., 2024). GSEM organizes clinical experiences in a dual-layer graph with applicability-aware retrieval and feedback-based updates (Han et al., 2026). We adopt its multi-trajectory experience-construction paradigm.

Our recommendation-specific design grounds retrieval in observed item identities: a target history reaches an experience through shared source-history items. The experience pool and its evidence links remain fixed during evaluation, and no graph embeddings or learned message passing are used. This shifts the retrieval unit from a case or a user profile to an instruction with an explicit, experience-specific behavioral link.

3 Problem Setting and Experience Representation

Let $H_u = (i_1, \ldots, i_L)$ be user u's chronological observed history, $I_u = \{i : i \in H_u\}$ its distinct item set, and C_u a candidate set. The task is to output a ranked list $Y_u = (y_1, \ldots, y_K)$ of distinct candidates, placing the held-out next interaction higher.

An experience is $e = (x_e, I_e)$, where x_e contains natural-language guidance and its applicability condition, and $I_e \subseteq I_{s(e)}$ contains items selected from the observed source-case history of user $s(e)$. We use $\mathrm{Prov}(e) = I_e$ to denote *item-level* provenance. This records selected item identities, not the complete source trajectory or a verified causal explanation of the guidance. Source-user identity is retained as metadata.

Given a fixed pool $\mathcal{E}$ built from training cases, the system selects $\mathcal{E}_u \subseteq \mathcal{E}$ and outputs $Y_u = F(H_u, C_u, \mathcal{E}_u)$. Selected experiences may come from the target user's earlier cases or other users. The evaluated setting therefore concerns cross-user reuse within a shared pool, rather than generalization to users excluded from experience construction. The held-out target label is used for evaluation only.

4 PREX-REC: Provenance-Aware Experience Transfer

4.1 Framework Overview

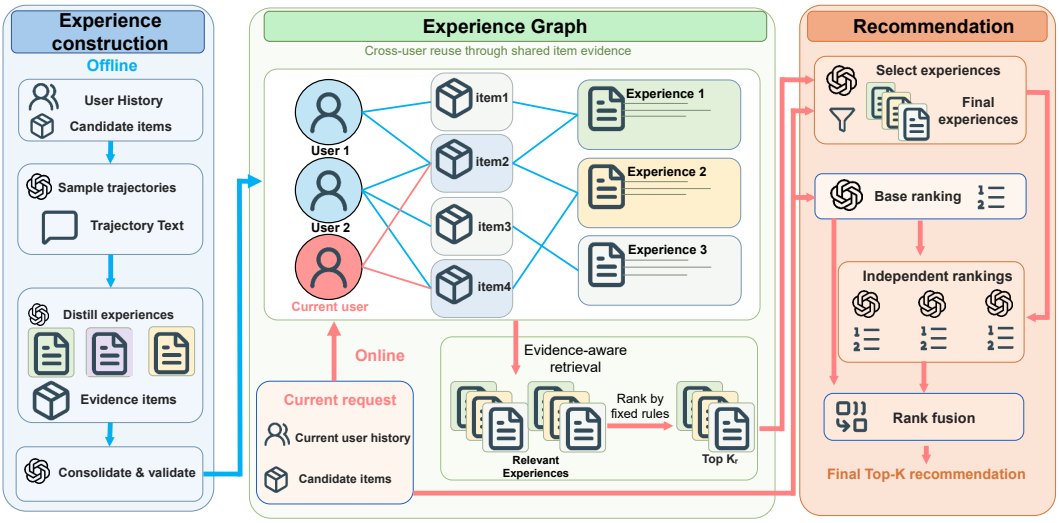


Figure 1: Overview of the proposed PREX-Rec framework. Blue components denote offline provenance-preserving experience graph construction, while red components denote online test-time recommendation.

PREX-Rec addresses experience transfer by preserving behavioral evidence during offline experience construction and exploiting it for online cross-user recommendation. Offline, experiences are extracted with their supporting evidence and organized into a provenance graph. Online, experiences are retrieved through shared evidence, filtered for applicability, and integrated through experience-guided ranking and fusion.

4.2 Experience Construction

We construct reusable experiences as evidence-grounded knowledge from historical training cases. Each experience follows the formulation defined in Section 3, where x_e is a natural-language recommendation rule and $I_e \subseteq I_u$ denotes its supporting evidence items. Following GSEM's multi-trajectory sampling paradigm (Han et al., 2026), we sample multiple recommendation attempts per case to capture successful and unsuccessful behaviors (Figure 2).

Trajectory sampling and experience extraction. For each historical user u, we sample T trajectories from H_u and candidate set C_u. Each trajectory produces a candidate ranking and reasoning trace:

$$(\tau_{u,t}, A_{u,t}) = f_{\text{traj}}(H_u, C_u). \tag{1}$$

Each ranking is evaluated against the ground-truth target. Successful trajectories provide useful recommendation patterns, while unsuccessful trajectories provide failure patterns and corrective guidance. The model then generates the experience description x_e and selects supporting evidence items I_e from the source history.

If successful trajectories exist for the same user, unsuccessful trajectories are compared with the best successful trajectory; otherwise, they are distilled independently. Each experience records a Condition, Guidance, Rationale, and Evidence; successful patterns yield Indications, while failure patterns yield Contraindications with corrective guidance. Source-case target labels are used offline to evaluate trajectories, but held-out test labels are not passed to the recommender. Evidence is restricted to observed source-history items. An item that reappears as a target may already occur in that history; its earlier occurrence remains eligible evidence.

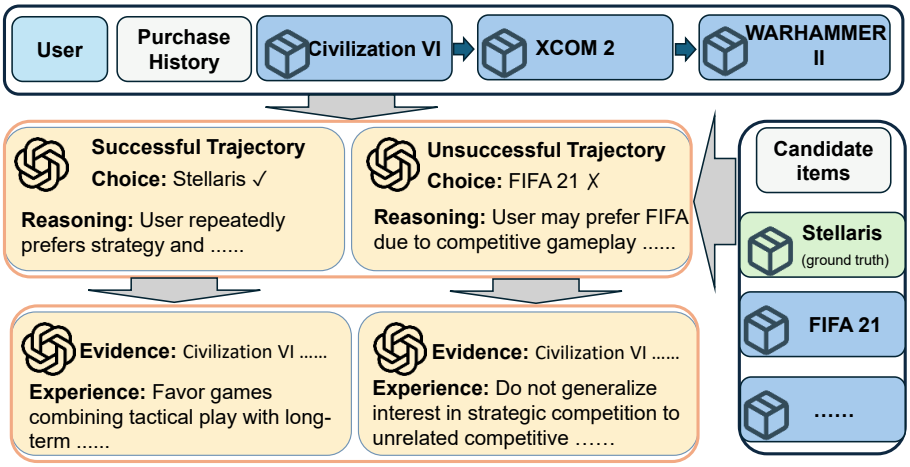


Figure 2: Experience construction from multiple trajectories and their supporting evidence items.

Consolidation and consistency filtering. Semantically redundant experiences from the same source user are consolidated. Each experience is further evaluated on its source case and retained only when its guided ranking satisfies the predefined success criterion without degrading the base recommendation. This check removes experiences that fail on their source case; it does not establish applicability to another user. Additional details are provided in Appendix C.1.

4.3 Provenance-Preserving Experience Graph Construction

To preserve the provenance defined in Section 3, we organize users, items, and experiences into a heterogeneous graph. User–item interactions and experience–item links preserve the behavioral evidence of each experience. Experiences can thus be transferred through shared evidence items:

$$\text{user} \rightarrow \text{item} \leftarrow \text{experience}.$$

Figure 3 illustrates this provenance structure. The graph is an explicit relational index; shared-item retrieval is also implementable with an inverted index. Source-user links record experience ownership, while the user–item–experience path supplies the retrieval score. No graph embeddings or graph neural networks are used.

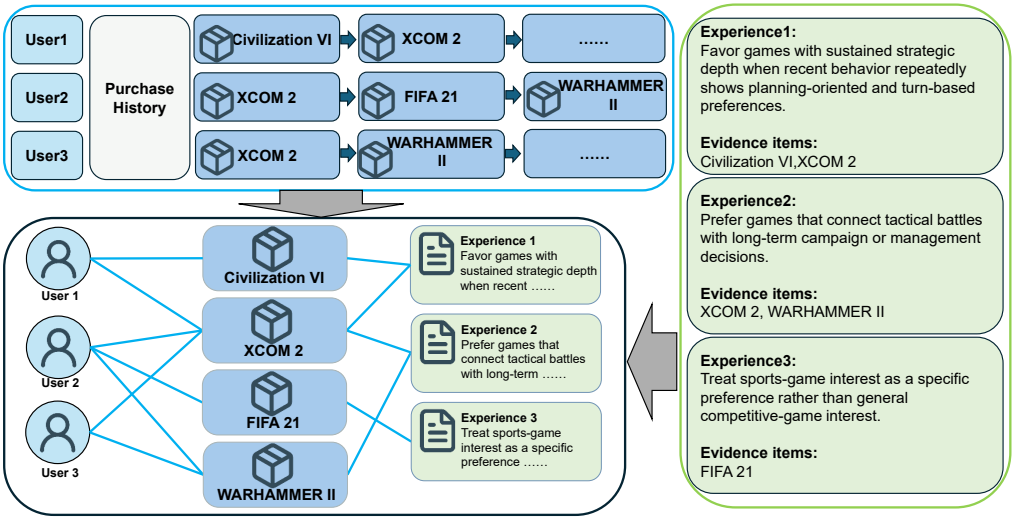


Figure 3: Illustration of the experience graph. User–item interactions and experience–item links preserve behavioral provenance for cross-user experience reuse.

Figure 3 illustrates why the evidence links belong to individual experiences. Guidance about sustained strategic play and guidance about tactical combat can arise from overlapping source histories but point to different item subsets. A target history can then reach either or both instructions according to its observed overlap. The graph stores this distinction explicitly; chronology and the natural-language condition are considered again during selection. The figure is a schematic illustration of the representation, not an additional empirical case study.

4.4 Evidence-Aware Graph Retrieval

For a target user, PREX-Rec uses shared evidence items as a retrieval heuristic for behavioral relevance. Specifically, it first identifies experiences whose evidence items overlap with the user's observed interaction history. Each shared item provides a direct structural connection between the user and experiences accumulated from previous cases.

The retrieval relevance is determined by three provenance-aware signals. First, experiences supported by more shared evidence items receive stronger relevance signals. Second, shared items are weighted by their specificity using inverse-frequency weighting (Salton & Buckley, 1988), since highly frequent evidence provides less discriminative information. Third, recency weighting emphasizes more recent interactions to capture evolving user preferences in sequential recommendation.

For distinct history items I_u and evidence items I_e, the retrieval score is

$$s_{\text{ret}}(u,e) = \frac{\sum_{i \in I_u \cap I_e} \left[1 + \log \frac{N_E+1}{df(i)+1}\right] \rho_u(i)}{\sqrt{|I_u||I_e|}}, \quad I_u \cap I_e \neq \varnothing. \tag{2}$$

Here N_E is the pool size, $df(i)$ counts experiences containing item i, and $\rho_u(i) \in [1,2]$ weights its last observed position from earliest to latest. The user-size factor is constant for that user's retrieval order; evidence-set normalization discounts broad item assignments.

The resulting score ranks candidate experiences, and the top K_r experiences are retained for subsequent applicability selection. The retrieval score is only used to identify candidate experiences

and does not directly affect final item ranking. Detailed weighting and normalization functions are provided in Appendix C.2.

4.5 Experience-Guided Recommendation

Retrieved experiences are not necessarily applicable to the current recommendation context. Therefore, PREX-Rec uses an LLM-based applicability selection step to identify useful experiences from the retrieved candidates.

We first obtain a base ranking from the user's history and candidate set. Each selected experience is then applied independently to generate an experience-guided ranking. This allows each experience to propose a separate, potentially selective ranking, without requiring the LLM to reconcile several instructions in one generation.

Selection returns up to three experience IDs and an operational note for each: an observed trajectory signal, a proposed ranking adjustment, and the candidate evidence needed to apply it. A Contraindication supplies corrective guidance rather than a failure pattern to imitate. Each selected instruction then generates its own candidate list using the same base ranking as a reference. This produces a common recommendation anchor while allowing different instructions to support different subsets of candidates.

For base list B_u and guided lists $R_{u,1}, \ldots, R_{u,m}$, fusion assigns candidate c the score

$$S_u(c) = \phi_{B_u}(c) + \sum_{j=1}^{m} \frac{\phi_{R_{u,j}}(c)}{\max(1, |R_{u,j}|)}, \qquad \phi_R(c) = \begin{cases} 2^{-(\mathrm{rank}_R(c)-1)}, & c \in R, \\ 0, & c \notin R. \end{cases} \tag{3}$$

Sorting these scores gives the final recommendation. Shorter guided lists receive greater per-item weight as a selectivity heuristic. The base ranking provides an experience-independent reference, while selected experiences contribute additional evidence according to their ranking positions and selectivity. If no experience is selected, PREX-Rec naturally reduces to the base recommendation model. The same fallback applies when no experience shares an observed item with the target history. Detailed fusion equations are provided in Appendix C.3.

5 Experiments

We compare PREX-Rec with five LLM-based recommenders on Games and Arts under GPT-4o-mini. The experiments address three questions: whether the complete framework improves candidate ranking; whether the retrieval signal and experience–item assignment matter within that framework; and whether selected cross-user experiences contribute to its predictions. Focused domain and backbone checks complement these component analyses in Appendix A.1 and Appendix A.2.

5.1 Experimental Setup

Datasets and protocol. We conduct experiments on the Arts and Games domains of the Amazon Review dataset (2018) (Ni et al., 2019), following the preprocessing and evaluation protocol of MoRE (Qin et al., 2025). For each test case, the model receives an observed history and a nominal pool of 50 candidates including one held-out target, and returns up to 10 items. In the saved Games data, 990 cases have 50 candidates and 10 have 49; all contain the target. Games uses 1,000 test users; each user's test history extends their training history with the training target. Experience texts and evidence links are built from training cases and frozen for evaluation; test nodes supply only observed history links for querying. We report mean Hit@k and NDCG@k, with one relevant item per case. Thus the protocol measures candidate ranking, not full-catalog recommendation.

Baselines. We compare with three categories of methods. MoRE (Qin et al., 2025) serves as the most related experience-based sequential recommendation baseline. LLM4RS (Dai et al., 2023) and LLMRank (Hou et al., 2024) represent general LLM-based recommendation methods. MemRec (Chen et al., 2026) and ItemRAG (Kim et al., 2026) represent recent memory and retrieval-augmented approaches. These baselines cover general LLM recommendation, experience-based recommendation, memory augmentation, and retrieval-based approaches.

Implementation details. All methods use GPT-4o-mini with item titles as textual input. For PREX-Rec, we sample $T = 3$ trajectories per user and retain experiences whose source-case guided ranking achieves NDCG@5 ≥ 0.5 without degrading the base ranking. We retrieve up to 10 experiences and select at most three. Trajectories are sampled at temperature 0.8; extraction and online ranking use 0.0. Games retains 917 experiences. Selection sees the most recent 20 history items, the candidate count, and retrieved experience fields; each guided ranking receives the history, candidate titles, selected guidance, its operational note, and the base ranking. Workflow call estimates appear in Appendix A.3.

5.2 Does Experience Reuse Improve Recommendation?

Table 1 reports the main recommendation results on Games and Arts.

Table 1: Main recommendation performance on Games and Arts.

Games

Method	Hit@1 ↑	Hit@3 ↑	Hit@5 ↑	NDCG@3 ↑	NDCG@5 ↑
LLM4RS (Dai et al., 2023)	0.1580	0.2670	0.3330	0.2211	0.2483
LLMRank (Hou et al., 2024)	0.1880	0.2880	0.3340	0.2469	0.2658
MoRE (Qin et al., 2025)	0.1380	0.2100	0.3880	0.1795	0.2521
MemRec (Chen et al., 2026)	0.1700	0.3120	0.4090	0.2520	0.2918
ItemRAG (Kim et al., 2026)	0.2070	0.3150	0.3670	0.2694	0.2909
PREX-Rec	**0.2220** *	**0.3460** *	**0.4330** *	**0.2932** *	**0.3291** *

Arts

Method	Hit@1 ↑	Hit@3 ↑	Hit@5 ↑	NDCG@3 ↑	NDCG@5 ↑
LLM4RS (Dai et al., 2023)	0.1390	0.2330	0.3010	0.1930	0.2210
LLMRank (Hou et al., 2024)	0.1620	0.2490	0.2960	0.2120	0.2310
MoRE (Qin et al., 2025)	0.1080	0.1750	0.3010	0.1450	0.1970
MemRec (Chen et al., 2026)	0.1990	0.3290	0.4130	0.2740	0.3080
ItemRAG (Kim et al., 2026)	0.2090	0.3280	0.3910	0.2769	0.3026
PREX-Rec	**0.2330** *	**0.3470** *	**0.4280** *	**0.2972** *	**0.3307** *

Bold black values denote the best result; light-gray cells mark the strongest baseline for the corresponding metric. * denotes a statistically significant improvement over the strongest baseline for the corresponding metric ($p < 0.05$, paired t-test).

PREX-Rec outperforms the five compared baselines in all reported metrics on both domains, with the paired-test significance indicated in Table 1. Games NDCG@5 increases from 0.2918 to 0.3291 over the strongest baseline (+12.8% relative), and Arts from 0.3080 to 0.3307 (+7.4%). These comparisons establish the performance of the complete system under this protocol; the following analyses examine retrieval, evidence assignment, and reuse separately. Appendix A.1 compares with MoRE on Musical Instruments, and Appendix A.2 compares experience augmentation with the base ranker under a second backbone.

5.3 Does Behavioral Provenance Matter?

We first compare item-based retrieval with a specified text-based retrieval baseline. For vector retrieval, user histories and experience descriptions are encoded using `all-MiniLM-L6-v2` (Reimers & Gurevych, 2019) and retrieved by cosine similarity. Joint integration applies selected experiences together; independent integration generates a separate ranking per experience. The same experience pool, base ranker, and selection limit are used. The actual number of selected experiences can differ across retrievers. Graph retrieval uses the evidence IDs, whereas the vector query matches history text to the abstract experience description. It therefore compares two retrieval signals, not two algorithms given identical information.

Table 2: Retrieval and integration on Games and Arts. The same four configurations expose the domain-dependent interaction between retrieval and integration. Graph + Independent corresponds to the full PREX-Rec configuration.

		Games				Arts			
Retrieval	Integration	H@3	H@5	N@3	N@5	H@3	H@5	N@3	N@5
Vector	Joint	0.2950	0.3560	0.2567	0.2815	0.3390	0.4150	0.2925	0.3238
Vector	Independent	0.3230	0.4080	0.2702	0.3049	0.3390	0.4060	0.2925	0.3203
Graph	Joint	0.2980	0.3780	0.2582	0.2910	0.3400	0.4170	0.2926	0.3244
Graph	Independent	**0.3460**	**0.4330**	**0.2932**	**0.3291**	**0.3470**	**0.4280**	**0.2972**	**0.3307**

H@k: Hit@k; N@k: NDCG@k. Bold marks the best value and gray shading marks the second-best value within each domain. No additional significance claims are made for the ablations.

With independent integration fixed, graph retrieval raises Games NDCG@5 from 0.3049 to 0.3291 (Table 2). With graph retrieval fixed, independent integration improves over joint integration from 0.2910 to 0.3291. Both choices therefore matter in this configuration. On Arts, graph retrieval improves NDCG@5 by 0.0104 with independent integration, while its gain under joint integration is only 0.0006, and independent integration lowers vector-retrieval NDCG@5 from 0.3238 to 0.3203. The strongest configuration combines the two components; their benefits are not uniform across settings. These are workflow comparisons, not an isolation of gains from additional LLM computation or an internal interference mechanism.

The abstraction used in experience extraction deliberately separates reusable guidance from concrete item names. Vector matching consequently tests retrieval from that abstract guidance, while graph matching restores access to source-item identities through I_e. The gain is consistent with the usefulness of retaining this additional behavioral information. The evidence-assignment perturbation below is the more targeted comparison: it keeps the experience text and source user fixed and changes which history items define the connections. Neither comparison establishes superiority over a text retriever supplied with evidence titles or a full-source-history item matcher.

Experience-specific evidence. To test the item assignments more directly, we replace each retained experience's evidence with the same number of randomly sampled items from its source history, preserving experience text and source identity. Retrieval weighting, applicability selection, and ranking procedures are unchanged. Table 3 shows a decrease in Games NDCG@5 from 0.3291 to 0.3086. However, 62.1% of original evidence items remain and 34.0% of experiences retain identical evidence sets. This is a partial perturbation, supporting the utility of the original assignments in this workflow; it does not establish causal faithfulness of the selected evidence. Appendix A.6 gives the overlap statistics. Base-reference and fusion diagnostics are in Appendix A.4.

Table 3: Effect of randomizing evidence-item assignment on Games. Each experience retains the same number of evidence items, sampled from its source user's history.

Setting	Hit@3 ↑	Hit@5 ↑	NDCG@3 ↑	NDCG@5 ↑
Full PREX-Rec	**0.3460**	**0.4330**	**0.2932**	**0.3291**
Randomized Evidence Items	0.3360	0.4040	0.2807	0.3086

5.4 Do Selected Cross-user Experiences Help?

We compare two variants on Games. *Self-only (selected)* keeps only same-user experiences from the full method's cached selection, without retrieving replacements. *Random Retrieval* samples 10 experiences from the full pool before applicability selection (seed 20260923). Both use the same base ranking and independent integration procedure.

Removing selected cross-user experiences lowers NDCG@5 from 0.3291 to 0.3017 (Table 4). It also reduces the number of applied experiences from 2,234 to 792, so the result measures their system-level contribution, not an effect isolated from experience quantity or inference budget. Random

Table 4: Selected cross-user removal and random retrieval on Games.

Setting	Hit@3	Hit@5	NDCG@3	NDCG@5
Random Retrieval	0.3220	0.3850	0.2626	0.2880
Self-only (selected)	0.3260	0.3860	0.2767	0.3017
Full PREX-Rec	**0.3460**	**0.4330**	**0.2932**	**0.3291**

retrieval selects 2,956 experiences yet reaches only 0.2880 NDCG@5; more selected experience alone does not ensure better ranking. The comparison therefore favors relevant selection over the indiscriminate addition of more experience-guided rankings. The shared-pool protocol remains distinct from a source-user-disjoint evaluation.

We further analyze selected experiences on the Games test set (Appendix A.5). Among 2,234 selected experiences, 64.55% originate from other users, and 72.9% of test users select at least one cross-user experience. This confirms that cross-user experiences are a substantial component of the recommendation process rather than rare fallback cases.

5.5 How Do Retrieval and Integration Contribute Together?

The existing controls separate several practical roles. Under the full graph configuration on Games, removing the base ranking from the guided prompt lowers NDCG@5 from 0.3291 to 0.3151; assigning every guided list weight 0.1 lowers it to 0.3186 (Appendices A.4 and C.3). The base reference and actual-length weighting therefore contribute alongside retrieval. In contrast, the joint configuration performs slightly better without the prompt reference (0.2952 versus 0.2910). This interaction supports evaluating an experience together with how its guidance is applied, rather than treating integration as a neutral final step.

The retrieval comparison also helps interpret inference volume: the vector-independent variant selects 2,988 experiences on Games, compared with 2,234 for the full graph configuration, yet has lower NDCG@5 (0.3049 versus 0.3291). Along with random retrieval's 2,956 selections and 0.2880 NDCG@5, this shows that the full configuration does not obtain its retrieval advantage simply by applying more experiences. These counts are not token- or latency-matched comparisons. The joint–independent comparison additionally changes the number and format of ranking calls, so it establishes a useful integration configuration rather than isolating the value of memory from generic test-time ensembling.

Focused domain and backbone checks complement these analyses. On Musical Instruments, PREX-Rec reaches 0.2224 NDCG@5 versus MoRE's 0.1575. On Games with Qwen3.5-Flash, it improves over the corresponding base ranker from 0.3073 to 0.3562 (Appendices A.1–A.2). These results extend the observed gains to another domain and backbone; the two main domains retain the full baseline comparison.

6 Conclusion

We present PREX-Rec, which links reusable recommendation guidance to selected items from its source history and retrieves it through shared item evidence. The full system outperforms five LLM-based baselines on Games and Arts. In existing Games controls, partial within-history evidence randomization lowers NDCG@5 from 0.3291 to 0.3086, while filtering the full method's selection to same-user experiences lowers it to 0.3017. These results support evidence-linked experience reuse in the evaluated shared-pool workflow. They do not establish superiority over full-history matching or compute-matched ranking ensembles. Because retrieval relies on shared observed items, its coverage may also be limited when target histories have little overlap with the available experience evidence. Future work can assess these alternatives and coverage limitations directly.

Reproducibility Statement

We provide the experimental setup and evaluation protocol in Section 5.1, additional experimental analyses in Appendix A, complete prompt templates in Appendix B, and the mathematical definitions of source-case consistency filtering, evidence-aware graph retrieval, and rank fusion in Appendix C.

AI Use Statement

Generative AI tools were used to assist with language polishing, manuscript formatting, bibliographic verification, and limited code modification and debugging during the preparation of this work. The research ideas, method design, experimental design, experimental results, and scientific conclusions were developed and verified by the authors. All AI-assisted outputs were reviewed by the authors, who take full responsibility for the content of this paper.

References

Keqin Bao, Jizhi Zhang, Yang Zhang, Wenjie Wang, Fuli Feng, and Xiangnan He. TALLRec: An effective and efficient tuning framework to align large language model with recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems*, pp. 1007–1014. ACM, 2023. doi: 10.1145/3604915.3608857. URL https://doi.org/10.1145/3604915.3608857.

Weixin Chen, Yuhan Zhao, Jingyuan Huang, Zihe Ye, Mingxuan Ju, Tong Zhao, Neil Shah, Li Chen, and Yongfeng Zhang. MemRec: Collaborative memory-augmented agentic recommender system. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 44515–44544. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.2061. URL https://aclanthology.org/2026.acl-long.2061/.

Sunhao Dai, Ninglu Shao, Haiyuan Zhao, Weijie Yu, Zihua Si, Chen Xu, Zhongxiang Sun, Xiao Zhang, and Jun Xu. Uncovering ChatGPT's capabilities in recommender systems. In *Proceedings of the 17th ACM Conference on Recommender Systems*, pp. 1126–1132. ACM, 2023. doi: 10.1145/3604915.3610646. URL https://doi.org/10.1145/3604915.3610646.

Shijie Geng, Shuchang Liu, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. Recommendation as language processing (RLP): A unified pretrain, personalized prompt & predict paradigm (P5). In *Proceedings of the Sixteenth ACM Conference on Recommender Systems*, pp. 299–315. ACM, 2022. doi: 10.1145/3523227.3546767. URL https://doi.org/10.1145/3523227.3546767.

Priyanshu Gupta, Shashank Kirtania, Ananya Singha, Sumit Gulwani, Arjun Radhakrishna, Gustavo Soares, and Sherry Shi. MetaReflection: Learning instructions for language agents using past reflections. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 8369–8385. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.emnlp-main.477. URL https://aclanthology.org/2024.emnlp-main.477/.

Bernal Jiménez Gutiérrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. HippoRAG: Neurobiologically inspired long-term memory for large language models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 59532–59569, 2024. doi: 10.52202/079017-1902. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/6ddc001d07ca4f319af96a3024f6dbd1-Abstract.html.

Xiao Han, Yuzheng Fan, Sendong Zhao, Haochun Wang, and Bing Qin. GSEM: Graph-based self-evolving memory for experience augmented clinical reasoning. *arXiv preprint arXiv:2603.22096*, 2026. doi: 10.48550/arXiv.2603.22096. URL https://arxiv.org/abs/2603.22096.

Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. Session-based recommendations with recurrent neural networks. In *The Fourth International Conference on Learning Representations*, 2016. URL https://openreview.net/forum?id=yoffK5KZSgQ.

Yupeng Hou, Junjie Zhang, Zihan Lin, Hongyu Lu, Ruobing Xie, Julian J. McAuley, and Wayne Xin Zhao. Large language models are zero-shot rankers for recommender systems. In *Advances in Information Retrieval: 46th European Conference on Information Retrieval, ECIR 2024, Glasgow, UK, March 24–28, 2024, Proceedings, Part II*, volume 14609 of *Lecture Notes in Computer Science*, pp. 364–381. Springer, 2024. doi: 10.1007/978-3-031-56060-6_24. URL https://doi.org/10.1007/978-3-031-56060-6_24.

Wang-Cheng Kang and Julian J. McAuley. Self-attentive sequential recommendation. In *2018 IEEE International Conference on Data Mining (ICDM)*, pp. 197–206. IEEE, 2018. doi: 10.1109/ICDM.2018.00035. URL https://doi.org/10.1109/ICDM.2018.00035.

Sein Kim, Hongseok Kang, Seungyoon Choi, Donghyun Kim, Minchul Yang, and Chanyoung Park. 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*, pp. 1395–1406. ACM, 2024. doi: 10.1145/3637528.3671931. URL https://doi.org/10.1145/3637528.3671931.

Sunwoo Kim, Geon Lee, Kyungho Kim, Jaemin Yoo, and Kijung Shin. ItemRAG: Item-based retrieval-augmented generation for LLM-based recommendation. In *Proceedings of the 49th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 3884–3888. ACM, 2026. doi: 10.1145/3805712.3809942. URL https://doi.org/10.1145/3805712.3809942.

Jiacheng Li, Yujie Wang, and Julian McAuley. Time interval aware self-attention for sequential recommendation. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, pp. 322–330. ACM, 2020. doi: 10.1145/3336191.3371786. URL https://doi.org/10.1145/3336191.3371786.

Jiacheng Li, Ming Wang, Jin Li, Jinmiao Fu, Xin Shen, Jingbo Shang, and Julian J. McAuley. Text is all you need: Learning language representations for sequential recommendation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 1258–1267. ACM, 2023. doi: 10.1145/3580305.3599519. URL https://doi.org/10.1145/3580305.3599519.

Jianghao Lin, Rong Shan, Chenxu Zhu, Kounianhua Du, Bo Chen, Shigang Quan, Ruiming Tang, Yong Yu, and Weinan Zhang. ReLLa: Retrieval-enhanced large language models for lifelong sequential behavior comprehension in recommendation. In *Proceedings of the ACM Web Conference 2024*, pp. 3497–3508. ACM, 2024. doi: 10.1145/3589334.3645467. URL https://doi.org/10.1145/3589334.3645467.

Hanjia Lyu, Song Jiang, Hanqing Zeng, Yinglong Xia, Qifan Wang, Si Zhang, Ren Chen, Chris Leung, Jiajie Tang, and Jiebo Luo. LLM-Rec: Personalized recommendation via prompting large language models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pp. 583–612. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-naacl.39. URL https://aclanthology.org/2024.findings-naacl.39/.

Jianmo Ni, Jiacheng Li, and Julian McAuley. 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)*, pp. 188–197, Hong Kong, China, 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1018. URL https://aclanthology.org/D19-1018/.

Weicong Qin, Yi Xu, Weijie Yu, Chenglei Shen, Xiao Zhang, Ming He, Jianping Fan, and Jun Xu. MoRE: A mixture of reflectors framework for large language model-based sequential recommendation. In *Proceedings of the Nineteenth ACM Conference on Recommender Systems*, pp. 299–308. ACM, 2025. doi: 10.1145/3705328.3748055. URL https://doi.org/10.1145/3705328.3748055.

Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. 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), pp. 3982–3992, Hong Kong, China, 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1410. URL https://aclanthology.org/D19-1410/.

Xubin Ren, Wei Wei, Lianghao Xia, Lixin Su, Suqi Cheng, Junfeng Wang, Dawei Yin, and Chao Huang. Representation learning with large language models for recommendation. In *Proceedings of the ACM Web Conference 2024*, pp. 3464–3475. ACM, 2024. doi: 10.1145/3589334.3645458. URL https://doi.org/10.1145/3589334.3645458.

Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5):513–523, 1988. doi: 10.1016/0306-4573(88)90021-0. URL https://doi.org/10.1016/0306-4573(88)90021-0.

Scott Sanner, Krisztian Balog, Filip Radlinski, Ben Wedin, and Lucas Dixon. Large language models are competitive near cold-start recommenders for language- and item-based preferences. In *Proceedings of the 17th ACM Conference on Recommender Systems*, pp. 890–896. ACM, 2023. doi: 10.1145/3604915.3608845. URL https://doi.org/10.1145/3604915.3608845.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, pp. 8634–8652, 2023. doi: 10.52202/075280-0377. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/1b44b878bb782e6954cd888628510e90-Abstract-Conference.html.

Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pp. 1441–1450. ACM, 2019. doi: 10.1145/3357384.3357895. URL https://doi.org/10.1145/3357384.3357895.

Zhen Tan, Jun Yan, I-Hung Hsu, Rujun Han, Zifeng Wang, Long Le, Yiwen Song, Yanfei Chen, Hamid Palangi, George Lee, Anand Rajan Iyer, Tianlong Chen, Huan Liu, Chen-Yu Lee, and Tomas Pfister. In prospect and retrospect: Reflective memory management for long-term personalized dialogue agents. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8416–8439. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.413. URL https://aclanthology.org/2025.acl-long.413/.

Tuo Wang, Meng Jian, Ge Shi, Lifang Wu, and Yashen Wang. MSR-Rec: Multi-step reasoning-enhanced LLM for sequential recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 15877–15885, 2026. doi: 10.1609/aaai.v40i19.38620. URL https://ojs.aaai.org/index.php/AAAI/article/view/38620.

Yancheng Wang, Ziyan Jiang, Zheng Chen, Fan Yang, Yingxue Zhou, Eunah Cho, Xing Fan, Yanbin Lu, Xiaojiang Huang, and Yingzhen Yang. RecMind: Large language model powered agent for recommendation. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pp. 4351–4364. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-naacl.271. URL https://aclanthology.org/2024.findings-naacl.271/.

Xu Xie, Fei Sun, Zhaoyang Liu, Shiwen Wu, Jinyang Gao, Jiandong Zhang, Bolin Ding, and Bin Cui. Contrastive learning for sequential recommendation. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pp. 1259–1273. IEEE, 2022. doi: 10.1109/ICDE53745.2022.00099. URL https://doi.org/10.1109/ICDE53745.2022.00099.

Junjie Zhang, Yupeng Hou, Ruobing Xie, Wenqi Sun, Julian J. McAuley, Wayne Xin Zhao, Leyu Lin, and Ji-Rong Wen. AgentCF: Collaborative learning with autonomous language agents for recommender systems. In *Proceedings of the ACM Web Conference 2024*, pp. 3679–3689. ACM, 2024. doi: 10.1145/3589334.3645537. URL https://doi.org/10.1145/3589334.3645537.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. ExpeL: LLM agents are experiential learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 19632–19642, 2024. doi: 10.1609/aaai.v38i17.29936. URL https://ojs.aaai.org/index.php/AAAI/article/view/29936.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. MemoryBank: Enhancing large language models with long-term memory. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 19724–19731, 2024. doi: 10.1609/aaai.v38i17.29946. URL https://ojs.aaai.org/index.php/AAAI/article/view/29946.

Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. S3-Rec: Self-supervised learning for sequential recommendation with mutual information maximization. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pp. 1893–1902. ACM, 2020. doi: 10.1145/3340531.3411954. URL https://doi.org/10.1145/3340531.3411954.

A Additional Experimental Details

A.1 Additional Domain Evaluation

To further examine whether the observed performance trend extends beyond the Games and Arts domains, we additionally evaluate PREX-Rec on the Musical Instruments domain of the Amazon Review dataset. Given the substantial cost of LLM-based evaluation, we do not repeat the full baseline suite used in the main experiments. Instead, we conduct a focused comparison with MoRE, the most directly related experience-based sequential recommendation method. All settings follow the same evaluation protocol and LLM backbone as in the main experiments.

Table 5: Additional evaluation on the Musical Instruments domain.

Method	Hit@3 ↑	Hit@5 ↑	NDCG@3 ↑	NDCG@5 ↑
MoRE	0.1400	0.2420	0.1160	0.1575
PREX-Rec	**0.2420**	**0.3120**	**0.1934**	**0.2224**

PREX-Rec outperforms MoRE on all reported metrics. This focused third-domain evaluation provides additional evidence that the improvements over the most directly related experience-based baseline also extend to a different recommendation domain.

A.2 Backbone Robustness

To examine whether the effectiveness of PREX-Rec depends on a particular LLM backbone, we additionally evaluate the method on Games using Qwen3.5-Flash in place of GPT-4o-mini. Given the substantial cost of repeating the complete baseline suite with an additional backbone, we focus on the effect of experience augmentation by comparing PREX-Rec with its corresponding base ranking under each backbone.

Table 6: Backbone robustness on the Games domain.

Backbone	Method	Hit@3 ↑	Hit@5 ↑	NDCG@3 ↑	NDCG@5 ↑
GPT-4o-mini	Base Only	0.2940	0.3440	0.2557	0.2763
GPT-4o-mini	PREX-Rec	**0.3460**	**0.4330**	**0.2932**	**0.3291**
Qwen3.5-Flash	Base Only	0.3270	0.3700	0.2895	0.3073
Qwen3.5-Flash	PREX-Rec	**0.3750**	**0.4720**	**0.3164**	**0.3562**

Under both backbones, experience augmentation consistently improves all reported metrics over the corresponding base ranking. With Qwen3.5-Flash, PREX-Rec improves Hit@3 from 0.3270 to 0.3750 and Hit@5 from 0.3700 to 0.4720, with corresponding gains in NDCG@3 and NDCG@5.

The consistent improvement under a second LLM backbone provides additional evidence that the benefit of the proposed experience-management framework is not specific to GPT-4o-mini.

A.3 Workflow-Level Model Usage

We report approximate workflow-level model invocation counts under the same 1,000-user evaluation scale. For PREX-Rec, the counts are derived from the implemented stage-level workflow, using the corresponding scripts and generated intermediate artifacts to account for the LLM-invoking operations in both offline experience construction and online recommendation. They are therefore code-supported workflow estimates derived from the implemented pipeline. Table 7 provides the detailed breakdown.

Table 7: Approximate stage-level model invocation breakdown of PREX-Rec for 1,000 users.

Stage	Script	Approx. Calls
Data preparation	`prepare_items.py`	0
Experience mining	`mine_experiences.py`	∼6,200
Source-case consistency filtering	`validate_experiences.py`	∼1,440
Graph construction	`build_graph.py`	0
Test-time base ranking	`run_base.py`	∼1,000
Test-time recommendation	`run_pipeline.py`	∼3,230
Total		**∼11.9K**

The dominant offline model usage comes from experience mining, which requires approximately 6,200 calls: about 3,000 for trajectory sampling, 3,000 for experience extraction, and 200 for within-user consolidation performed as part of the mining workflow. Source-case consistency filtering requires approximately 1,440 calls, including about 520 source-user base rankings and 920 experience-guided rankings. At test time, base ranking requires approximately 1,000 calls, while the full recommendation pipeline requires approximately 3,230 calls, including about 1,000 experience-selection calls and 2,230 independent experience-guided reranking calls. These estimates are derived directly from the implemented workflow and its intermediate outputs; small variations may occur across executions because of API retries and other runtime differences. Data preparation and graph construction are deterministic and require no LLM invocation.

For context, Table 8 compares the resulting overall workflow-level usage with two representative baselines. These numbers are intended as approximate end-to-end usage indicators rather than direct measures of token consumption, latency, or monetary cost.

Table 8: Approximate workflow-level model invocation counts for 1,000 users.

Method	Approx. Model Calls
PREX-Rec	∼11.9K
ItemRAG	∼17K
MoRE	∼34K

Overall, PREX-Rec has a lower approximate workflow-level model invocation count than the two representative baselines under this accounting. Despite the additional offline experience-construction stage, graph construction is deterministic and online retrieval introduces no additional LLM calls before experience selection.

A.4 Base Reference and Integration Diagnostics

Table 9 separates the base ranking's role as a prompt reference from its contribution during fusion. The independent variant improves from 0.3151 to 0.3291 NDCG@5 when the reference is provided. Joint integration instead declines from 0.2952 to 0.2910. A beneficial reference effect is therefore specific to the independent configuration in these results.

Table 9: Effect of providing the base ranking as a reference for experience-guided ranking.

Base Reference	Integration	Hit@3	Hit@5	NDCG@3	NDCG@5
–	Base Only	0.2940	0.3440	0.2557	0.2763
No	Joint	0.2980	0.3870	0.2584	0.2952
Yes	Joint	0.2980	0.3780	0.2582	0.2910
No	Independent	0.3360	0.4210	0.2797	0.3151
Yes	Independent	**0.3460**	**0.4330**	**0.2932**	**0.3291**

Table 10: Combined base-removal diagnostic on Games. The no-base variants also omit the prompt reference and request full-length guided rankings.

Base retained	Integration	Hit@3	Hit@5	NDCG@3	NDCG@5
No	Joint	0.2590	0.3300	0.2141	0.2434
No	Independent	0.2710	0.3450	0.2150	0.2457
Yes	Joint	0.2980	0.3780	0.2582	0.2910
Yes	Independent	**0.3460**	**0.4330**	**0.2932**	**0.3291**

Table 10 compares the full configurations with saved variants that omit the base reference, use full-length guided rankings, and omit the base contribution to fusion. The joint variant generates a ranking even with no selected experience; the independent variant falls back to candidate order in that case. Consequently, the lower scores describe a combined change in the recommendation workflow and do not isolate the effect of removing the base from fusion.

A.5 Cross-User Experience Usage Analysis

We further analyze the sources of the experiences selected by PREX-Rec on the Games test set. For each selected experience, we identify whether it was constructed from training cases of the current target user or from another user. Table 11 reports the complete statistics over the 1,000 test users.

Table 11: Source statistics of selected experiences on Games.

Statistic	Value
Number of test users	1,000
Total selected experiences	2,234
Average selected experiences per user	2.234
Experiences from current user	792
Experiences from other users	1,442
Current-user share	35.45%
Cross-user share	64.55%
Users selecting current-user experience	500 (50.0%)
Users selecting cross-user experience	729 (72.9%)
Unresolved experience IDs	0

Among the 2,234 selected experiences, 1,442 (64.55%) originate from users other than the current target user. Moreover, 729 of the 1,000 test users (72.9%) select at least one cross-user experience, compared with 500 users (50.0%) selecting at least one experience derived from their own training cases. Thus, cross-user experiences constitute the majority of selected experiences and are used for a substantial fraction of test users. Combined with the self-only ablation in Table 4, where retaining only same-user experiences from the cached selection lowers all reported metrics, these statistics support the practical system-level benefit of cross-user experience reuse, including broader access to relevant experiences.

A.6 Randomized Evidence-Item Assignment

The performance results are reported in Table 3 in the main text. This existing ablation replaces evidence items within each source user's history while retaining the experience text, source user, and evidence-set size. It uses one random seed, 20260925. The same retrieval and downstream recommendation procedures are then applied to the perturbed links. Table 12 quantifies how much of the original assignment remains.

Table 12: Overlap between original and randomized evidence assignments over all 917 retained experiences. Randomization uses seed 20260925.

Overlap statistic	Value
Identical ordered evidence lists	99 / 917 (10.8%)
Identical evidence sets (ignoring order)	312 / 917 (34.0%)
At least one shared evidence item	777 / 917 (84.7%)
Original evidence items retained	1,534 / 2,472 (62.1%)
Mean per-experience Jaccard similarity	0.5184

The perturbation preserves 1,534 of 2,472 original evidence items and leaves 312 of 917 evidence sets unchanged. Evidence order is immaterial to the set-based retrieval score; identical sets are therefore more informative than identical ordered lists. The lower scores show sensitivity to this partial within-history perturbation. They do not measure retrieval quality independently of the downstream ranker.

B Prompt Templates for Games

We provide paper-formatted versions of the prompt templates used in the Games experiments. Each boxed template corresponds to a single LLM invocation. Runtime inputs such as user histories, candidate items, trajectories, and retrieved experiences are represented by placeholders in braces. Where the implementation separates instructions and runtime inputs into different message roles, they are presented together here as one template. For readability, implementation-specific Markdown markers, line wrapping, and capitalization are normalized, while the task instructions, constraints, input fields, and output requirements are preserved. Graph construction and score fusion are deterministic procedures and therefore do not involve additional LLM prompts.

B.1 Base Ranking

The base recommender ranks candidates using only the observed interaction history and item titles. The same ranking later serves as the experience-independent reference for experience-guided recommendation.

Base Ranking Prompt

I've purchased the following products in the past in order:

{history}

Now there are {n_candidates} candidate products that I can consider to purchase next:

{candidates}

Use behavioral and recency signals inferable from the item titles to judge which product I am most likely to purchase next.
Rank the top 10 candidates from most to least likely to be purchased next, according to the given purchasing records. Use only the item titles shown above.

Required output.
Return only item IDs in this exact format:

```
RANKED LIST: [item_id1, item_id2, ...]
```

Do not explain. Do not output product titles or row numbers. Every item ID must come from the candidate list and appear at most once.

B.2 Trajectory Sampling

For each historical case, we sample multiple recommendation trajectories with temperature 0.8 and a maximum output budget of 512 tokens. The model first produces a natural-language reasoning trace and then ranks the candidate items.

Trajectory Sampling Prompt

You are a recommendation assistant. You must first reason about the user's behavioral trajectory, then rank the top 10 items.
Your reasoning should be concrete and grounded in the user's interaction history. It is the "trajectory" that will later be used to extract reusable recommendation experiences.

User interaction history (chronological, `[ID] title`)

{`interaction_history`}

Candidate items ({`candidate_count`} total)

{`candidate_list`}

Task.

1. Think step by step about the user's recent behavior, preferences, and shifts.
2. Then rank the top 10 items this user is most likely to choose next.

Required output.
Return only a JSON object with two keys:

```
{
  "reasoning":  "your reasoning trajectory",
  "ranking":  ["ID1", "ID2", ...]
}
```

B.3 Experience Extraction

Successful and unsuccessful trajectories are converted into *Indication* and *Contraindication* experiences, respectively. Each extracted experience contains an applicability condition, actionable guidance, a rationale, and supporting evidence items selected from the source user's observed interaction history.

B.3.1 Indication Extraction

For a successful trajectory, the model receives the source history, reasoning trajectory, successful ranking, and ground-truth item.

Indication Extraction Prompt

You extract a reusable positive recommendation experience from a successful ranking case.
A successful ranking means the model correctly captured the user's current behavioral direction and ranked the target item highly.

Analysis framework.

1. **Context:** What behavioral trajectory signal is present in the user's history?
2. **Actionable guidance:** What should a ranking do when this signal appears?
3. **Reasoning logic:** Why does this action capture the user's actual direction?

4. **Outcome benefit:** What ranking mistakes does this prevent, such as stale preference, surface category matching, or missing a recency shift?

User interaction history (chronological, `[ID] title`)

{interaction_history}

Recommender's reasoning trajectory

{trajectory}

Resulting successful ranking

{successful_ranking}

Item actually chosen by the user (context only; do not encode in the experience)

{ground_truth}

Extract one reusable Indication experience that explains what behavioral signal the ranking correctly recognized and how future rankings should act on it.

Evidence constraints.

- Evidence must contain at least one real user-history item ID that supports the experience.
- Select evidence IDs only from the `[ID]` labels in the provided interaction history.
- Do not use item titles, brands, franchise names, or free-form summaries as evidence.
- The same history item may appear in multiple experiences only when it genuinely supports each of them.

Abstraction constraints.

- Do not mention actual item titles, franchise names, brand names, platform names, specific game names, series names, or user IDs in the condition, guidance, rationale, or tags.
- Replace proper nouns with abstract role labels and describe behavioral patterns rather than concrete entities.
- Tags should describe behavioral signals and abstract categories.
- The condition must describe a trajectory direction rather than a static snapshot.
- Guidance must be actionable and conditional.
- The rationale must explain the decision logic rather than merely restating the guidance.

Required output.
Return only a JSON object with the following schema:

```
{
  "condition":
    "Applicability condition:  the behavioral trajectory signal
    that triggers this experience",
  "guidance":
    "Concrete ranking adjustment:  what to promote, what to
    demote, and what signal to verify",
  "rationale":
    "Why this adjustment improves ranking",
  "evidence":  ["item_id_1", "item_id_2", "..."],
  "tags":  ["short", "retrievable", "behavior", "tags"]
}
```

The source-case target is provided as offline feedback, and the prompt requests abstract guidance rather than copying that target. Evidence IDs are constrained to the observed source history. This

restriction concerns the information source: an earlier occurrence of an item remains eligible even if that item is later chosen again.

B.3.2 CONTRASTIVE CONTRAINDICATION EXTRACTION

For an unsuccessful trajectory, when a successful trajectory from the same source user is available, the two attempts are contrasted.

Contrastive Contraindication Extraction Prompt

You extract a reusable negative recommendation experience by comparing a failed ranking with a successful ranking.

Analysis framework.

1. **Decision comparison:** Compare how the failed and successful rankings treat the user's behavioral trajectory signals.
2. **Key divergence:** Identify the critical behavioral signal that the failed ranking overlooked or misweighted.
3. **Failure mechanism:** Explain how this incorrect emphasis leads to a wrong ranking.
4. **Consequence:** Identify what recommendation quality is lost, such as missing a recency shift or over-trusting stale signals.

Contraindication semantics.
The condition and rationale describe the failure risk, but the guidance must state the corrective action a future ranker should execute. Never write guidance that asks the ranker to repeat the failed behavior.

Failed user's interaction history (chronological, `[ID] title`)

{`failed_history`}

Failed reasoning trajectory

{`failed_trajectory`}

Failed top-10 ranking

{`failed_ranking`}

Comparable successful user's interaction history (chronological, `[ID] title`)

{`success_history`}

Successful reasoning trajectory

{`success_trajectory`}

Successful top-10 ranking

{`success_ranking`}

Failed user's actual chosen item (context only; do not encode)

{`ground_truth`}

Extract one reusable Contraindication experience: identify what behavioral trajectory signal the failed ranking missed and what a future ranking should do instead.

Evidence constraints.

- Evidence must contain at least one real history item ID.
- Select evidence IDs only from the `[ID]` labels in the failed user's interaction history.
- Do not use item titles, brands, franchise names, or free-form summaries as evidence.

Abstraction constraints.

- Do not mention actual item titles, franchise names, brand names, platform names, specific game names, series names, or user IDs in the condition, guidance, rationale, or tags.
- Replace proper nouns with abstract role labels and describe behavioral patterns rather than concrete entities.
- The condition must describe a trajectory direction.
- Guidance must be actionable and corrective.
- The rationale must explain the failure mechanism rather than merely restating the guidance.

Required output.
Return only a JSON object with the following schema:

```
{
  "condition":
    "Applicability condition: the behavioral trajectory signal
    that triggers this experience",
  "guidance": "Concrete corrective ranking adjustment",
  "rationale":
    "Why this adjustment avoids the failure mechanism",
  "evidence": ["item_id_1", "item_id_2", "..."],
  "tags": ["short", "retrievable", "behavior", "tags"]
}
```

B.3.3 Direct Contraindication Extraction

If no successful trajectory is available for the same user, the Contraindication is extracted directly from the failed trajectory.

Direct Contraindication Extraction Prompt

You extract a reusable negative recommendation experience from a failed ranking case when no comparable successful case is available.

Analysis framework.

1. **Error pattern:** What behavioral signal was ignored or misread?
2. **Recommendation context:** In what type of user trajectory does this error occur?
3. **Failure mechanism:** How does this mistake push the true target down?
4. **Consequence:** What recommendation signal is lost, such as recency, category shift, or preference transition?

Contraindication semantics.
The condition and rationale describe the failure risk, but the guidance must state the corrective action a future ranker should execute. Never write guidance that asks the ranker to repeat the failed behavior.

User interaction history (chronological, `[ID] title`)

{interaction_history}

Recommender's reasoning trajectory

{trajectory}

Failed top-10 ranking

{failed_ranking}

Item actually chosen by the user (context only; do not encode)

{ground_truth}

Extract one reusable Contraindication experience that explains the missed behavioral signal and the concrete ranking adjustment to make next time.

Evidence constraints.

- Evidence must contain at least one real history item ID.
- Select evidence IDs only from the `[ID]` labels in the provided interaction history.
- Do not use item titles, brands, franchise names, or free-form summaries as evidence.

Abstraction constraints.

- Do not mention actual item titles, franchise names, brand names, platform names, specific game names, series names, or user IDs in the condition, guidance, rationale, or tags.
- Replace proper nouns with abstract role labels and describe behavioral patterns rather than concrete entities.
- The condition must describe a trajectory direction.
- Guidance must be actionable and corrective.
- The rationale must explain the failure mechanism rather than merely restating the guidance.

Required output.
Return only a JSON object with the following schema:

```
{
  "condition":
    "Applicability condition: the behavioral trajectory signal
    that triggers this experience",
  "guidance": "Concrete corrective ranking adjustment",
  "rationale":
    "Why this adjustment avoids the failure mechanism",
  "evidence": ["item_id_1", "item_id_2", "..."],
  "tags": ["short", "retrievable", "behavior", "tags"]
}
```

B.4 Same-User Experience Consolidation

Indication and Contraindication experiences are consolidated separately within each source user. The curator identifies semantically redundant experiences and returns the IDs of the experiences to retain.

Same-User Experience Consolidation Prompt

You are a recommendation experience curator specializing in deduplication of reusable ranking-adjustment experiences. Your task is to identify and remove semantically similar or redundant experiences.

Deduplication criteria.

1. Two experiences are considered duplicates if:
   - their conditions describe semantically equivalent applicability contexts or behavioral trajectory signals;
   - their guidance conveys the same ranking strategy; and
   - they describe the same behavioral recommendation scenario.
2. When duplicates are found:
   - keep the version with clearer and more precise wording;
   - keep the version with more complete information; and
   - if both are equally good, keep the first one.

3. Preserve diversity:
   - keep experiences addressing different behavioral contexts;
   - keep experiences with distinct trajectory signals; and
   - keep experiences with non-overlapping applicability conditions.

Experience type

{experience_type}

Experiences to deduplicate ({count} total)

{experiences_json}

Required output.
Return only a valid JSON array containing the IDs of experiences to keep.
Copy IDs exactly from the input. Never invent or modify an ID. Do not return complete experience objects or rewritten content.
For example:

```
["exp_00001", "exp_00003"]
```

B.5 Source-Case Consistency Filtering

After same-user consolidation, retained experiences are checked on their source recommendation cases. This stage compares an experience-guided ranking against a source-case base ranking. The filtering criterion itself is given in Appendix C.1.

B.5.1 Source-Case Base Ranking

Source-Case Base Ranking Prompt

You are a recommendation assistant. Rank items from a candidate set.
Each candidate is shown as `[ID] title`. Output the IDs in ranked order.

Current recommendation scene

{scene_summary}

User interaction history (chronological)

{interaction_history}

Candidate items ({candidate_count} total)

{candidate_list}

Task.
Rank the top 10 items this user is most likely to choose next.

Required output.
Return only a JSON object with key `"ranking"`, containing exactly 10 item IDs:

```
{"ranking": ["ID1", "ID2", ...]}
```

B.5.2 Source-Case Experience-Guided Ranking

Source-Case Experience-Guided Ranking Prompt

You are a recommendation assistant. You will receive selected experiences labelled `[Indication]` or `[Contraindication]`.
An Indication describes a successful strategy. A Contraindication describes a failure risk, but its Guidance is a corrective instruction: follow that Guidance to avoid the failure mechanism. Never copy, reproduce, or reinforce the failed behavior.

Use the experience as an additional recommendation angle, not as a hard rule.

User interaction history (chronological, `[ID] title`)

{interaction_history}

Selected experiences

{experiences}

Candidate items ({candidate_count} total)

{candidate_list}

Task.
Rank the top 10 items this user is most likely to choose next.

Required output.
Return only a JSON object with key `"ranking"`, containing exactly 10 item IDs:
`{"ranking": ["ID1", "ID2", ...]}`

An experience passes source-case consistency filtering only when its guided ranking satisfies the predefined success criterion without underperforming the corresponding source-case base ranking.

B.6 Experience Selection

After graph retrieval, the LLM filters retrieved experiences according to their applicability to the current user's observed behavior. It may select zero to three complementary experiences.

Experience Selection Prompt

You select experiences that can make a concrete correction to a weak recommendation base. Candidates may be Indications or Contraindications.
An experience is applicable only when the target history contains direct evidence for its Condition. An Indication contributes its successful Guidance. A Contraindication contributes only its corrective Guidance; never imitate the failure behavior.
Select one to three complementary experiences when supported; select zero only when none has observable support. Do not select redundant experiences.
For each selected experience, write a concise operational note containing:

1. the observed trajectory signal;
2. the ranking adjustment to apply; and
3. what kind of candidate would count as confirming evidence.

Do not invent facts absent from item titles.

You are a video game recommendation expert. Match reusable experiences to the target user's chronological behavior.

Target user ID

{user_id}

User interaction history (chronological, `[ID] title`)

{interaction_history}

Candidate pool overview

{candidate_overview}

Candidate experiences (labelled `[Indication]` or `[Contraindication]`)

{candidate_experiences}

Prefer experiences whose Condition is visibly supported by the history and whose Guidance can distinguish candidates. Cover different behavioral angles. Select zero only if no candidate experience has direct support.

Required output.

```
{
  "selected_ids": ["experience_id", ...],
  "preference_notes": {
    "experience_id":
      "signal; adjustment; confirming candidate evidence"
  }
}
```

The operational note is subsequently supplied together with its corresponding experience during independent experience-guided ranking.

B.7 Single-Experience Guided Ranking

Each selected experience is applied independently to the candidate set. The base ranking is provided as a reference, while the selected experience defines the particular recommendation perspective used to revise it.

Single-Experience Guided Ranking Prompt

You are an experience-guided recommendation ranker. You will receive one selected experience labelled `[Indication]` or `[Contraindication]`.
For an Indication, apply its Guidance only when its Condition is supported by the target user's history. For a Contraindication, detect the failure risk and execute its corrective Guidance; never reproduce the failed behavior.
Ground every promoted candidate in both the user's observed trajectory and the experience's actionable guidance.
You are a video game recommendation expert. Produce a precise ranking correction from one selected experience.
Use behavioral and recency signals inferable from the item titles. Read the interaction history as an ordered sequence of behaviors such as franchise, genre, platform, play style, accessory versus game, and edition or region variants, and give recent interactions more weight when preferences shift.

Decision process.
Before answering, reason silently through the following steps:

1. Locate concrete support in the chronological history for the experience's Condition, using behavioral and recency signals.
2. Translate the experience's Guidance into candidate-level promotion and demotion criteria. For a Contraindication, apply the corrective action rather than the failure pattern.
3. Keep only candidates supported by both the user's history and this experience angle. Reject candidates supported only by generic popularity, superficial title overlap, or the weak base.
4. Use the weak base only as a fallback or tie-breaker. Do not copy it mechanically, but do not change an item merely to be different.
5. Rank the strongest evidence-backed corrections first and stop when the experience no longer distinguishes candidates reliably.

Target user ID

{user_id}

Angle-specific preference note

{preference_note}

Weak base ranking (fallback only)

{base_ranking}

Selected experience

{experiences}

User interaction history (chronological)

{interaction_history}

Candidate products

{candidate_list}

Required output.
Return item IDs only in this exact format:

```
RANKED LIST: [item_id1, item_id2, ...]
```

Return one to five unique candidates. Do not pad with weak or generic guesses. Every ID must come from the candidate list.

Allowing the model to stop after 1–5 candidates is what permits experience-guided rankings to have different lengths: an experience is not required to rank candidates that it does not meaningfully distinguish.

B.8 Deterministic Stages

Graph construction and final rank fusion do not invoke the LLM. Graph construction deterministically creates the user–item and experience–item relations used for retrieval, while final score fusion combines the base ranking and independently generated experience-guided rankings according to the rule in Appendix C.3.

C Additional Method Details

This section provides the mathematical details omitted from the main text for readability, including source-case consistency filtering, evidence-aware graph retrieval, and rank fusion.

C.1 Source-Case Consistency Filtering

After same-user consolidation, each experience is checked on the recommendation case from which it was derived. Let b_u denote the quality score of the base ranking for source user u, and let $v_{u,e}$ denote the score obtained when experience e is used to guide ranking on the same case. An experience is retained only if

$$v_{u,e} \geq \theta \qquad \text{and} \qquad v_{u,e} \geq b_u, \tag{4}$$

where θ is the predefined success threshold.

The first condition requires the experience-guided ranking to satisfy the same success criterion used for trajectory evaluation. The second ensures that the experience does not degrade the recommendation relative to the corresponding base ranking. This filtering step removes experiences that are ineffective even on their source cases rather than establishing cross-user transferability. The concrete ranking metric, cutoff, and threshold used in our experiments are specified in Section 5.1.

C.2 Detailed Graph Retrieval Function

For a target user u, let I_u denote the distinct items in the observed interaction history and let I_e denote the supporting evidence items of experience e. Only experiences satisfying

$$I_u \cap I_e \neq \varnothing \tag{5}$$

are considered eligible for graph retrieval.

The retrieval score combines shared behavioral evidence with evidence specificity and interaction recency. Let N_E denote the number of retained experiences and let $df(i)$ denote the number of experiences whose evidence sets contain item i. We assign each item the inverse-frequency weight

$$\mathrm{idf}(i) = \log\left(\frac{N_E + 1}{df(i) + 1}\right) + 1. \tag{6}$$

Items that occur as evidence for fewer experiences therefore contribute more strongly than items shared by many experiences.

Let $p_u(i)$ be the one-based position of the last occurrence of item i in the length-L observed history. We use $\rho_u(i) = 1 + (p_u(i) - 1)/(L - 1)$ for $L > 1$, and $\rho_u(i) = 1$ for $L = 1$. Repeated items contribute once, using their most recent position. These weights range from 1 to 2 without learned parameters.

The complete retrieval score is

$$s_{\mathrm{ret}}(u, e) = \frac{\sum_{i \in I_u \cap I_e} \mathrm{idf}(i)\rho_u(i)}{\sqrt{|I_u|\,|I_e|}}. \tag{7}$$

The numerator sums shared-item contributions. The $\sqrt{|I_e|}$ factor discounts larger evidence sets. For a fixed user, $\sqrt{|I_u|}$ is constant across experiences and therefore has no effect on their top-K_r order; it only rescales the scores supplied to selection. We do not claim this normalization removes history-length effects on retrieval. No eligible overlap yields an empty retrieval set and the base-ranking fallback.

Eligible experiences are ranked by $s_{\mathrm{ret}}(u, e)$, and the top K_r are passed to the subsequent applicability-selection stage. This score is used only for experience retrieval and does not directly contribute to the final candidate-ranking score.

C.3 Detailed Rank Fusion Rule

Let B_u denote the base ranking for user u, and let $R_{u,1}, \ldots, R_{u,m}$ denote the independently generated rankings from the m selected experiences. Define $R_{u,0} = B_u$ so that the base ranking and experience-guided rankings can be treated uniformly during fusion.

We use an exponentially decaying rank contribution

$$\phi(r) = 2^{-(r-1)}, \tag{8}$$

where r denotes the position of an item in a ranking. Candidates appearing near the top therefore receive substantially larger contributions.

The base ranking has unit weight $w_0 = 1$, while each guided ranking is weighted by its actual length:

$$w_j = \frac{1}{\max(1, |R_{u,j}|)}, \qquad j \geq 1. \tag{9}$$

The final fusion score of candidate c is

$$S_u(c) = \sum_{j=0}^{m} \mathbf{1}[c \in R_{u,j}]\, w_j\, \phi\big(\mathrm{rank}_{R_{u,j}}(c)\big)\,. \tag{10}$$

Thus, a candidate receives support from every ranking in which it appears, with stronger contributions from higher positions. Length normalization prevents an experience-guided ranking from receiving greater total influence simply because it contains more candidates. Guided prompts request selective outputs; shorter rankings are therefore assigned greater per-item weight as a design heuristic, rather than as a calibrated measure of confidence.

The final list sorts the union of returned candidates by $S_u(c)$ and keeps up to K items. Ties follow first occurrence: base-list order first, then guided-list order. Experiences are processed by decreasing

unweighted evidence overlap, with selection order breaking ties. Invalid and duplicate output IDs are discarded. Guided prompts request one to five items; the parser accepts at most ten valid items and fusion uses the actual returned length. If no applicable experience is selected, only the base ranking contributes and the procedure naturally reduces to the base recommendation.

Effect of length-aware weighting. We additionally examine whether allowing shorter guided rankings to express more concentrated support materially affects recommendation performance. We compare the full rule above with a controlled variant in which every experience-guided ranking is treated as having length 10 when computing its weight: $w_j = 1/10$ for $j \geq 1$, regardless of its actual length. Rank decay, the base-ranking weight, and all other components remain unchanged. This preserves the overall fusion procedure while removing the relative preference induced by differences in guided-ranking length.

Table 13: Effect of length-aware guided-ranking weighting on Games.

Setting	Hit@3 ↑	Hit@5 ↑	NDCG@3 ↑	NDCG@5 ↑
Fixed guided weight ($w_j = 0.1$)	0.3420	0.4200	0.2864	0.3186
Full PREX-Rec	**0.3460**	**0.4330**	**0.2932**	**0.3291**

The actual-length weighting has higher point estimates on all four metrics than fixed $w_j = 0.1$. This comparison changes both relative weights across guided lists and their overall influence against the unit-weight base list. It therefore supports this fusion choice within the tested system, without isolating selectivity calibration or proving that fusion is a minor source of the full system's gains.