

rEDMRec: Distilling Large Language Model Reasoning into an Editable Experience Memory for Recommendation

Minh Hoang Nguyen^{a,b}, Tung Le^{a,b} and Huy Tien Nguyen^{a,b,*}

^aFaculty of Information Technology, University of Science, Ho Chi Minh City, Vietnam

^bVietnam National University, Ho Chi Minh City, Vietnam

ARTICLE INFO

Keywords:

LLM-based recommendation
reasoning distillation
experience memory
memory-augmented agents
multi-agent debate
retrieval-augmented generation

ABSTRACT

Large language models (LLMs) can improve recommendation quality by reasoning explicitly over user history and candidate items – for example, extracting a user’s preferences or explaining why one item fits better than another – rather than mapping history directly to a ranked list. This reasoning, however, is expensive to repeat on every ranking request and, once produced, is typically consumed once and discarded, leaving it neither reusable across future requests nor easy to inspect or correct as user tastes drift. Our insight is that reasoning does not need to be regenerated at every call if it can instead be compressed once into a compact, structured memory that a lightweight model retrieves from. We propose rEDMRec, which distills a teacher LLM’s reasoning into four typed, editable experience channels – long-term preference, short-term context, item-perception, and counterfactual hard-negative comparisons – maintained by an LLM memory controller that performs Add/Delete/Modify/Keep operations and refines entries via K -agent debate. A lightweight student LLM (3B–20B parameters) then ranks candidates purely by retrieving from this memory, without invoking the teacher again, decoupling online inference cost from reasoning depth. Across ML-1M, Amazon Beauty, and Steam and ten student backbones, rEDMRec improves HR@1 over zero-shot, few-shot, and RAG on every backbone, and over GraphRAG on most backbones (exceptions: Llama 3.1 8B and GPT OSS 20B), with Impv up to 13.3% vs. the second-best baseline on ML-1M, following the relative-improvement protocol of RDRec (Wang et al., 2024b). Channel ablations show that short-term context is the only channel that helps consistently across capacity tiers, whereas long-term, item-perception, and counterfactual contributions are capacity-dependent (and can reverse on the strongest students); debate-based memory optimization lowers bank duplication by 7.4 percentage points while raising downstream HR@1 by up to +0.029 over six optimization epochs.

1. Introduction

Large language models (LLMs) are increasingly used as recommenders – via prompting, instruction tuning, collaborative-signal fusion, or generative item prediction (Wu et al., 2024; Lin et al., 2025; Bao et al., 2023; Liao et al., 2024; Hou et al., 2024) – and, more recently, as explicit reasoners over user history and candidate items (Wei et al., 2022). An LLM can extract preferences, judge item fit, or contrast a candidate against a hard negative, then use that reasoning to guide ranking. This paper studies a concrete bottleneck that follows from that capability: once such reasoning has been produced for a user, how can it be *reused* across future ranking requests, rather than regenerated from scratch every time?

Reasoning-augmented recommenders face a tension between reasoning depth and inference cost. Zero-shot and few-shot prompting, and retrieval-augmented generation (RAG) over raw interaction history (Lewis et al., 2020), are cheap but skip explicit preference-level reasoning, so ranking remains opaque and brittle under short histories. Closer to our setting, ReasoningRec (Bismay et al., 2025) uses a teacher LLM to synthesize user profiles,

item descriptions, and human-interpretable explanatory reasoning, then instruction-tunes a smaller model on those traces; R2Rec (Zhao et al., 2025) similarly builds interaction-of-thought chains and internalizes them with supervised fine-tuning and reinforcement learning; and R^4 ec (Gu et al., 2025) iterates actor reasoning with a reflection model that critiques and refines preference/item knowledge before feeding a recommendation backbone. These lines treat reasoning primarily as a per-request or training-time signal: competence lives in regenerated traces or model weights, cannot be updated entry-by-entry when new interactions arrive or low-quality reasoning accumulates, and revision requires another costly reasoning loop or retraining rather than a targeted memory edit. Related distillation methods compress rationales or prompts into smaller generators (Wang et al., 2024b; Li et al., 2023), yet still treat the distilled artifact as a fixed model rather than as a typed bank. The remaining challenge is therefore architectural: keep the benefit of teacher-level reasoning while making that reasoning reusable across requests and editable over time, without retraining the backbone whenever the underlying knowledge must change.

As illustrated in Figure 1, consider a user whose history is dominated by family and animation titles (e.g., *Toy Story*, *The Lion King*, *Shrek*, *Finding Nemo*) and who must rank *Frozen* against *The Dark Knight*. If the recommender focuses solely on explicit item titles or undifferentiated history text, it struggles to surface the subtle connection across these

*Corresponding author

✉ 24C15049@student.hcmus.edu.vn (M.H. Nguyen);

littung@fit.hcmus.edu.vn (T. Le); ntienhuy@fit.hcmus.edu.vn (H.T. Nguyen)

ORCID(s): 0009-0004-1384-3856 (M.H. Nguyen); 0000-0002-9900-7047 (T. Le); 0000-0002-9948-1048 (H.T. Nguyen)

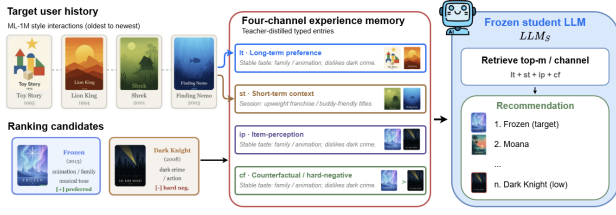


Figure 1: Example of four-channel experience memory distilled from a user's history for ranking *Frozen* vs. *The Dark Knight*. Offline, teacher reasoning is stored as typed lt/st/ip/cf entries; online, a frozen student LLM retrieves those channels and produces the recommendation list without calling the teacher.

interactions – that the user consistently prefers light family entertainment over dark crime – and has no structured place to store a hard-negative contrast that would push *The Dark Knight* down when that pattern holds. In this case, the latent preference is not a single keyword but a typed bundle of signals: long-term taste (stable family/animation preference; dislike of dark crime tone), short-term session context (franchise- or buddy-friendly titles), item-perception that grounds why *Frozen* matches the history while *The Dark Knight* mismatches, and a counterfactual that records when the dark-action alternative would have ranked higher. Existing reasoning-augmented methods either regenerate such comparisons per request or absorb them into weights, so they cannot keep these four signals as independently retrievable and independently editable memory entries across future ranking calls.

We address this challenge with rEDMRec. The basic idea is illustrated in Figure 1: instead of asking an LLM to both reason and rank on every inference call, we use the LLM as a *teacher reasoning generator* and distill its outputs into a structured, retrievable, and updatable Experience Memory that a lightweight student consumes at ranking time. Concretely, the teacher produces four typed experience signals – long-term preference, short-term context, item perception, and counterfactual hard-negative comparisons – and a distillation adapter normalizes each signal into a channel-indexed memory entry (vector store for the first three channels; hybrid vector-graph store for counterfactuals). At inference, a *frozen* student (3B–20B) retrieves the top- m entries per channel, composes a ranking prompt, and scores candidates without re-invoking the teacher, so accuracy gains are attributable to the memory rather than to student-parameter adaptation. A remaining difficulty is that a bank filled once from teacher extractions accumulates near-duplicates and low-specificity entries as interactions grow. Inspired by Training-Free GRPO (Youtu-Agent Team, 2025), which steers a frozen policy by maintaining an experiential knowledge library with Add/Delete/Modify/Keep operations rather than gradient updates, our second contribution is an LLM memory controller that applies the same edit operators – optionally guided by K -agent debate

and an arbiter – to revise the recommendation experience bank over optimization epochs. Together, these components separate an expensive, infrequent reasoning-compression process from a cheap, frequent retrieval-and-rank process.

We evaluate rEDMRec on ML-1M, Amazon Beauty, and Steam with ten student LLMs (3B–20B), comparing against zero-shot, few-shot, RAG, and GraphRAG (Edge et al., 2024). rEDMRec improves HR@1 over zero-shot, few-shot, and RAG on every backbone, and over GraphRAG on most backbones (exceptions: Llama 3.1 8B and GPT OSS 20B), reaching Impv = 13.3% vs. GraphRAG on Qwen2.5 3B under the RDRec relative-improvement protocol (Wang et al., 2024b) (Section 5.1). Channel ablations show that short-term context is the only consistently helpful channel across capacity tiers, whereas long-term, item-perception, and counterfactual effects are capacity-dependent (Section 5.2). A teacher-distillation study further shows that bank duplicate rate is a leading indicator of downstream gain, bounded by student capacity. Finally, debate-based optimization lowers bank duplication by 7.4 percentage points while raising downstream HR@1 by up to +0.029 over six epochs, validating the controller as a functional component of the architecture (Section 5.4).

This paper makes the following contributions:

1. **An editable, channel-structured experience memory** that distills teacher LLM reasoning into four typed, independently retrievable and independently editable channels, rather than a single undifferentiated reasoning trace (Section 3.6).
2. **A debate-based LLM memory controller** that revises the bank after each student prediction via Add/Delete/Modify/Keep operations guided by ranking reward models and K -agent debate, with bank-quality gains shown to propagate into ranking accuracy without updating the frozen student (Sections 3.6, 3.8; Section 5.4).
3. **A comprehensive empirical study** across ten student backbones and three datasets, including channel ablations that explain when ablating a channel can *improve* accuracy, a teacher-distillation study isolating teacher quality from student capacity, and a qualitative case study of entry-level edits over optimization epochs (Sections 5.1–5.5).

2. Related Work

2.1. LLM-based Recommendation

Before LLM recommenders, neural models already encoded dual-scale user interest, review text, and cold-start evaluation – but they stored that structure in parameters or in the corpus, not in an editable experience bank. Sequential news recommenders such as Co-NAML-LSTUR (Nguyen et al., 2025) jointly learn multi-view item encodings with long- and short-term user representations, showing that separating durable taste from recent browsing improves ranking even without an LLM. Review-based models such as RRS (Nguyen et al., 2024) replace ID-only collaborative

filtering with deep encoders over user-written text, so item semantics enter ranking through review content rather than through a retrieved, typed memory entry. Complementary dataset work such as ViHoRec (Nguyen, 2026; Nguyen and Thiet, 2025) shows that, on sparse Vietnamese hotel interactions with a temporal cold-start split, neighborhood methods can outperform learned latent-factor models on users with short histories. These lines motivate the same signal types that rEDMRec stores as channels – long-term preference, short-term context, and item-level text – yet they cannot persist teacher-generated reasoning or Add/Delete/Modify a typed entry after a ranking failure. rEDMRec keeps that factorization, but as a non-parametric bank that a frozen student retrieves from, rather than as a neural user/item tower.

Recent surveys organize LLM recommenders into prompting, tuning, collaborative fusion, generative recommendation, and system-enhancement paradigms (Wu et al., 2024; Lin et al., 2025; Wang et al., 2024a; Liu et al., 2025; Li et al., 2024). Prompting and instruction-following methods cast ranking as natural-language generation or zero-shot ordering (Gao et al., 2023; Zhang et al., 2023; Hou et al., 2024; Yue et al., 2023; Lyu et al., 2024), while alignment frameworks such as TALLRec and LLaRA adapt LLMs to recommendation with efficient fine-tuning (Bao et al., 2023; Liao et al., 2024). A parallel line injects collaborative or ID structure into the language space – CoLLM, BinLLM, TokenRec, and related models encode interaction signals as text-like or tokenized representations (Zhang et al., 2025a; 2024b; Qu et al., 2024) – and enhancement methods use LLMs for graph augmentation, tool use, or query generation around a classical backbone (Wei et al., 2024; Zhao et al., 2024; Han et al., 2025). Across these paradigms, any intermediate reasoning (when present) remains either ephemeral prompt context or knowledge absorbed into parameters. rEDMRec instead materializes teacher reasoning as a non-parametric, typed memory that survives across sessions and can be revised without re-tuning the student.

2.2. Reasoning, Distillation, and Preference Utilization

A narrower line makes LLM reasoning itself the object of design. ReasoningRec (Bismay et al., 2025) synthesizes user profiles, item descriptions, and explanatory rationales with a teacher LLM, then instruction-tunes a smaller model for both prediction and human-interpretable explanation – an extraction-then-fine-tune pipeline rather than a durable memory architecture. R2Rec (Zhao et al., 2025) samples interaction chains, builds interaction-of-thought traces, and internalizes them with SFT and RL; LatentR³ (Zhang et al., 2025b) similarly reinforces latent reasoning inside the model, while SPRec (Gao et al., 2025) uses self-play to debias generative recommenders. R⁴ec (Gu et al., 2025) pushes toward System-2 deliberation by pairing an actor that proposes preference/item knowledge with a reflection model that judges and triggers refinement until the knowledge

is deemed rational, then injects the refined text into a recommendation backbone – still a per-case reasoning loop rather than a persistent, editable experience bank. Distillation work such as RDRec, POD, and LEADER compresses rationales or teacher signals into smaller recommendation models (Wang et al., 2024b; Li et al., 2023; Liu et al., 2024). Retrieval baselines (RAG, GraphRAG) ground ranking in raw history or graph summaries without teacher-compressed experience (Lewis et al., 2020; Edge et al., 2024). Relative to these methods, rEDMRec neither stops at one-shot feature utilization nor freezes reasoning into weights: it distills reasoning into a channel-typed bank with entry-level Add/Delete/Modify/Keep.

2.3. Memory-Augmented Agents and Recommendation Memory

External memory lets LLM agents operate beyond a single context window. MemGPT (Packer et al., 2023) pages context like an OS virtual-memory manager; Generative Agents (Park et al., 2023) maintain a retrieve-and-reflect memory stream for long-horizon behavior. Closest to our controller design, Training-Free GRPO (Youtu-Agent Team, 2025) improves a frozen LLM by iteratively distilling experiential knowledge into a non-parametric library updated with Add/Delete/Modify/Keep operations – a training-free alternative to gradient-based GRPO. In recommendation, AutoMR retrieves stored experiences for generative ranking (Wang et al., 2025), long-term planners model durable taste (Shi et al., 2024), and related work motivates separating long- versus short-term interest (Zheng et al., 2024; Zhang et al., 2024a), item-level semantics (Ren et al., 2024; Zhang et al., 2026), hard-negative contrast (Song et al., 2026; Li et al., 2026), and user-controllable profiles (Woźniak et al., 2025). These designs retrieve history, profiles, or generic agent traces; they are not organized as teacher-distilled, four-channel recommendation experience with debate-driven bank maintenance. rEDMRec adopts the Training-Free GRPO principle of editing an external experience library instead of model weights, but specializes the schema to recommendation signal types and pairs each channel with a matching store (vector vs. hybrid graph–vector).

2.4. Multi-Agent Debate and Self-Refinement

Iterative critique improves LLM outputs without additional supervised labels. Self-Refine (Madaan et al., 2023) has a model revise its own text from self-feedback; multi-agent debate (Du et al., 2023) improves factuality by having several instances propose and critique answers; and in recommendation, R⁴ec (Gu et al., 2025) couples actor and reflection models to refine preference/item knowledge before backbone prediction. These lines evaluate a single artifact – one answer, one document, or one knowledge string per case – and do not measure how repeated refinement of a growing collection affects duplicate accumulation or retrieval quality at bank scale. rEDMRec applies critique-and-revise to the experience bank: K debating personas critique

Table 1

Notation used in Section 3.

Symbol	Meaning
u, H_u	user; chronological interaction history
$I, M(i)$	item catalog; metadata text of item i
$C_u \subset I, c$	20-item candidate set; a candidate in C_u
$i^+, i^-, \hat{i}$	positive target; hard-negative contrast; predicted item
x_u, D_K	ranking prompt; K in-context demonstrations (few-shot)
θ, P_S	generic LLM parameters; student likelihood under LLM_S
$\text{Enc}(\cdot), \text{sim}$	shared text encoder; cosine similarity
$\mathbf{q}_u, d$	query embedding $\text{Enc}(\text{prompt}(u, C_u))$; embedding dim.
$\mathcal{K}, k$	channel set $\{\text{lt}, \text{st}, \text{ip}, \text{cf}\}$; a channel
P, p, r_p	extraction-pass set $\{\text{pref}, \text{ctx}, \text{reas}, \text{cf}\}$; a pass; raw output
$E = \{E_k\}_{k \in \mathcal{K}}$	experience memory bank (Eq. 8)
$E_k^u \subset E_k$	channel- k entries for user u after metadata filter
$e = (\tau, \mathbf{v}_e, \mu)$	memory entry: text, embedding $\mathbf{v}_e = \text{Enc}(\tau)$, metadata
Adapt, route _k	distill routed fields into e_k ; select fields for channel k
snapshot(E)	truncated text rendering of the bank for LLM_C
$B = \{(\tau, k, u)\}$	insight batch (teacher or arbiter) consumed by LLM_C
o, Apply	edit ops $\{\text{ADD}, \text{DELETE}, \text{MODIFY}, \text{KEEP}\}$; commit o to E
$m, R_k(u, C_u)$	retrieval depth; top- m entries from E_k^u (Eq. 11)
$\hat{L}_u, \text{rank}$	student ranked list; 1-based position of i^+ in $\hat{L}_u$
$r(u)$	post-prediction reward vector (Eq. 13)
K, n_r, T	#debate agents; rounds/epoch; optimization epochs
$\mathcal{T}, g_j, LLM_j$	debate transcript; critique of agent j ; j -th debate agent
$\bar{e}, n_{\max}, U$	arbiter-proposed entry; max commits/case; case batch
$LLM_T, LLM_S, LLM_C, LLM_A$	teacher, frozen student, controller, arbiter

a case, an arbiter synthesizes revisions, and the controller commits Add/Delete/Modify/Keep operations, with bank-level duplicate rate and specificity linked to downstream HR@1 (Section 5.4).

3. Method

We formalize next-item ranking as conditional generation (Section 3.1), restate the dominant prompting and retrieval paradigms in the same notation to make the technical gap precise (Section 3.2), then specify rEDMRec's components as LLM operators over a typed non-parametric memory (Sections 3.4–3.8). Table 1 collects the notation used throughout.

3.1. Problem Formulation

We cast next-item recommendation as *conditional generation*: given user u , history H_u , and a candidate set C_u , a language model with parameters θ assigns a score to each candidate $c \in C_u$ by the likelihood of emitting c under a constructed prompt. The predicted item is

$$\hat{i} = \arg \max_{c \in C_u} P_\theta(c \mid \text{context}(u)), \quad (1)$$

and is evaluated against the held-out positive i^+ using HR@ k , NDCG@ k , and MRR (Section 4.4). Methods differ in what enters $\text{context}(u)$; we make this explicit below.

3.2. Preliminaries: Prior Paradigms as

Conditional Generation

Zero/few-shot prompting (Brown et al., 2020) conditions the LLM on a natural-language prompt x_u and K in-context demonstrations $D_K = \{(x_j, y_j)\}_{j=1}^K$, with no parameter update:

$$P_{\text{FS}}(c \mid u) = P_\theta(c \mid x_u, D_K), \quad x_u = \text{prompt}(H_u, M(C_u)). \quad (2)$$

$K=0$ recovers zero-shot. Every token of x_u and D_K is re-encoded by θ at every request.

RAG (Lewis et al., 2020) retrieves a top- k evidence set Z and conditions generation on it:

$$P_{\text{RAG}}(c \mid u) = P_\theta(c \mid x_u, Z), \quad Z = \text{Top-}k(x_u). \quad (3)$$

In our RAG baseline, Z ranges over *raw* interaction/review records and is recomputed for every (u, C_u) .

GraphRAG (Edge et al., 2024) builds a corpus graph, partitions it into communities $\{S_l\}$ with LLM-written summaries $\sigma(S_l)$, and answers by a map–reduce over community answers a_l :

$$P_{\text{GRAG}}(c \mid u) = \text{LLM}(x_u, c, \{a_l\}_{l=1}^L), \quad a_l = \text{LLM}(x_u, \sigma(S_l)). \quad (4)$$

$\{\sigma(S_l)\}$ is static once built and is not typed, per-user, or editable at the granularity of a single fact.

Equations (2)–(4) share a property that motivates our design: the conditioning set (D_K , Z , or $\{\sigma(S_l)\}$) is either rebuilt/rescored at every request, or, once built, exposes no operator for targeted, entry-level edits. Section 3.3 replaces this with a persistent structure E built off the inference critical path and equipped with an explicit edit operator (Section 3.6).

3.3. Overview

rEDMRec factorizes ranking into an offline construction of a typed experience memory E and an online, teacher-free generative lookup, as illustrated in Figure 2:

$$P_S(c \mid u) = P_S\left(c \mid x_u, \bigcup_{k \in \mathcal{K}} R_k(u, C_u)\right), \quad (5)$$

$$E \leftarrow \text{Apply}(E, o),$$

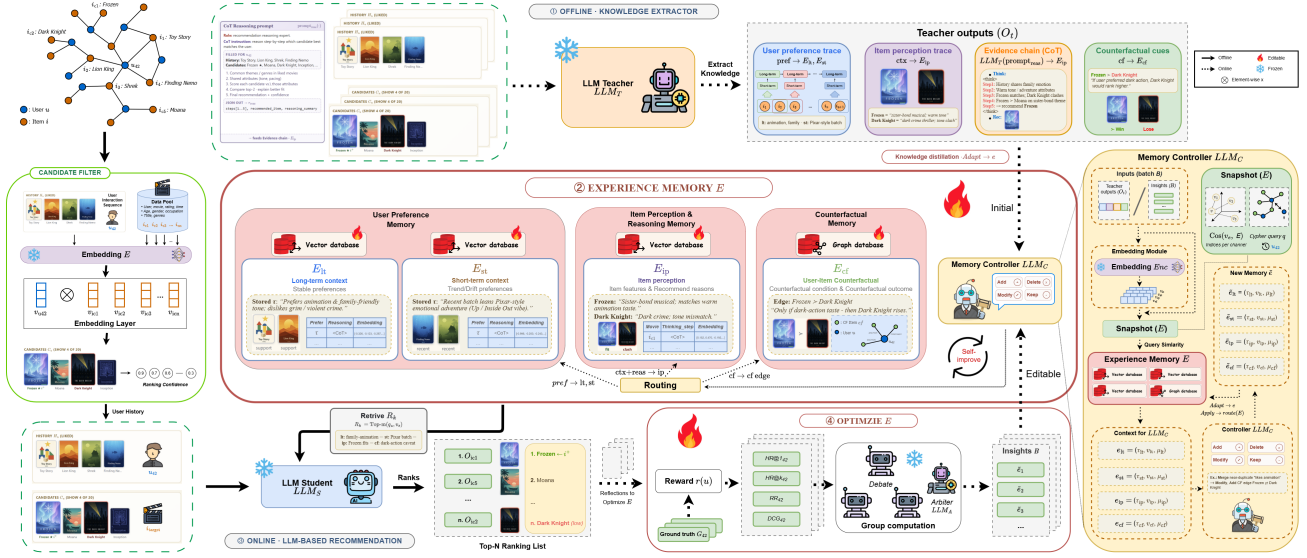


Figure 2: Overall architecture of rEDMRec on a concrete case (target *Frozen*, hard negative *The Dark Knight*), aligned with Sections 3.4–3.8. ① LLM Teacher as Knowledge Extractor: frozen LLM_T emits preference / perception / evidence / counterfactual traces; Knowledge Distillation (Adapt) writes four-channel entries that the Memory Controller LLM_C commits via Add/Delete/Modify/Keep. ② Editable Experience Memory Bank $E = \{E_{lt}, E_{st}, E_{ip}, E_{cf}\}$. Candidate Filter forms C_u with shared Enc. ③ LLM Student as Ranker: frozen LLM_S retrieves top- m entries per channel and ranks without calling the teacher. ④ Experience Memory Optimization: Reward Models $r(u)$ and Debate and Arbiter (LLM_A) revise E while LLM_S stays fixed.

where P_S denotes generation under the frozen student LLM_S , R_k (Eq. 11) is a deterministic top- m dense lookup (not a per-request marginalized set as in Eq. (3)), and Apply commits edit ops o produced offline by teacher extraction (Section 3.5) and distillation (Section 3.6), or online by debate-based optimization after each student prediction (Section 3.8). This is the central contrast with Eqs. (2)–(4): reasoning cost is paid when writing E , the frozen student only retrieves, and memory edits – not weight updates – amortize future ranking.

Concretely, Figure 2 decomposes the pipeline into the Method subsections that follow. ① LLM Teacher as Knowledge Extractor (Section 3.5): given (u, H_u, i, M) , LLM_T runs four extraction passes and Knowledge Distillation Adapt (Section 3.6.1) normalizes each routed signal into a channel-indexed entry $e = (\tau, \mathbf{v}_e, \mu)$, which the Memory Controller LLM_C commits with Add/Delete/Modify/Keep. ② Editable Experience Memory Bank (Section 3.6): $E = \{E_{lt}, E_{st}, E_{ip}, E_{cf}\}$ stores stable taste, session context, candidate-grounded perception, and counterfactual hard-negative edges as independently retrievable typed snippets. Before online ranking, Candidate Filter (Section 3.4) forms C_u and shares Enc with memory indexing. ③ LLM Student as Ranker (Section 3.7): frozen LLM_S encodes the current query, retrieves the top- m entries per channel, and produces $\hat{L}_u$ without invoking LLM_T or updating student weights. ④ Experience Memory Optimization (Section 3.8): Reward Models $r(u)$ score $\hat{L}_u$ against i^+ , then Debate and Arbiter LLM_A propose revisions

that Adapt and Apply write back into E , so future retrievals improve while LLM_S stays fixed.

3.4. Candidate Filter

Motivation: a generative student cannot score the full catalog $\mathcal{I}$ at each request, and retrieval plus memory indexing must share one dense text space rather than operating over raw strings. *Design:* candidate sets C_u are formed by a lightweight recency/popularity/retrieval-score filter over $\mathcal{I}$ (not a learned projector). The same encoder maps any text span to a d -dimensional retrieval vector,

$$\mathbf{v} = \text{Enc}(\text{text}), \quad \text{sim}(\mathbf{v}, \mathbf{v}') = \frac{\mathbf{v}^\top \mathbf{v}'}{\|\mathbf{v}\| \|\mathbf{v}'\|}, \quad (6)$$

with Enc a sentence-transformer (Reimers and Gurevych, 2019). $\mathbf{v}$ is reused for memory indexing (Eq. 9) and retrieval (Eq. 11). *Advantage:* C_u is formed without a learned projector, and one shared Enc lets E be queried directly with candidate or user vectors, with no separate alignment step.

3.5. LLM Teacher as Knowledge Extractor

Motivation: conflating “understand the user” (stable) with “score this candidate list” (per-request) forces the former to be redone every request; rEDMRec queries the teacher only for the former, and only to *extract*, never to rank. *Design:* following ReasoningRec (Bismay et al., 2025) and R2Rec (Zhao et al., 2025), which show that structured preference profiles, item-level perceptions, and explanatory rationales improve recommendation when distilled from a strong LLM, teacher LLM_T runs four extraction passes

Table 2

Teacher extraction passes $p \in \mathcal{P}$: output fields and routed channel(s) k (Section 3.6.1).

p	Key output fields	Routed to k
pref	long_term_preferences, dislikes, short_term_preferences	lt, st
ctx	user_history_perception, candidate_perception	ip
reas	steps[1..5], reasoning_summary	ip
cf	counterfactual_condition/outcome, rationale	cf

$p \in \mathcal{P}$, each a prompted generation,

$$r_p = \text{LLM}_T(\text{prompt}_p(u, i, H_u, M)), \quad p \in \mathcal{P}. \quad (7)$$

whose output fields are later routed to one or more memory channels $k \in \mathcal{K}$ by Adapt (Table 2, Section 3.6.1); the routing is not one-to-one, since two passes jointly populate the item-perception channel. Prompt templates and example one-line outputs for all four channels are given in Appendix I; full Beauty/Steam prediction traces (history, candidates, R_k , ranked list) appear in Appendix J. Unlike ReasoningRec/R2Rec, which consume these traces as training targets or one-shot prompt features, we never update student parameters from r_p ; each field is committed into the editable bank E of Section 3.6.

Preference extraction (pref). Simulates a recurrent, batch-by-batch update over H_u (oldest→newest): for each history batch it re-estimates a long-term preference state and the current-batch short-term interest, plus an explicit dislike list; the final long-term/dislike fields route to channel lt and the short-term field routes to channel st.

Context extraction (ctx). For every history item and candidate, produces three layers – an objective factual description, a first-person “as this user” comment, and (candidates only) key phrases; both the history-perception and candidate-perception fields route to the item-perception channel ip.

Reasoning extraction (reas). Runs a five-step chain-of-thought that identifies shared themes across liked items, scores each candidate against them, contrasts the top two candidates, and states a final recommendation; its summary field also routes to ip, complementing ctx with an explicit comparative rationale rather than a per-item description.

Counterfactual extraction (cf). Given an anchor (chosen) item and a contrast (hard-negative) item, produces why-preferred / why-rejected rationales and a hypothetical condition under which the contrast item would outrank the anchor; routes entirely to channel cf as a graph edge (Section 3.6.2).

An optional single-pass multi-agent refinement critiques the r_{cf} output with three fixed personas before distillation, merging by “last full critique wins” with no arbiter; this is a lighter-weight precursor to the arbiter-based optimization of Section 3.8, which instead revises already-committed entries across all four channels using downstream reward signals. *Advantage:* separating extraction into four typed passes, rather than one undifferentiated call, lets Section 3.6.1 edit

a single routed claim without touching the other channels of the same interaction.

3.6. Editable Experience Memory Bank

Motivation: Eq. (7) yields free-form JSON that cannot be reused across sessions or revised entry-by-entry once committed. *Design:* adapting the training-free experiential-knowledge library of Training-Free GRPO (Youtu-Agent Team, 2025) to recommendation, rEDMRec materializes teacher output into a persistent, typed non-parametric memory

$$E = \{E_k\}_{k \in \mathcal{K}}, \quad E_k = \{e \mid e.\text{memory_type} = k\}, \quad (8)$$

where each entry $e = (\tau, \mathbf{v}_e, \mu)$ carries distilled text τ , embedding $\mathbf{v}_e = \text{Enc}(\tau)$, and channel-specific metadata μ . Population follows a two-stage commit path – Adapt (field routing and schema normalization) and Apply via controller LLM_C (batch edit ops) – detailed in Section 3.6.1; channel semantics and retrieval are in Section 3.6.2.

3.6.1. Knowledge Distillation: Adapter and Memory Controller

Distillation adapter Adapt. Adapt maps routed teacher fields (Table 2) into a typed entry and dispatches it to the correct physical store:

$$e_k = \text{Adapt}(\text{route}_k(\{r_p\}_{p \in \mathcal{P}})) = (\tau_k, \text{Enc}(\tau_k), \mu_k), \quad k \in \mathcal{K}. \quad (9)$$

Here $\text{route}_k(\cdot)$ selects and concatenates the field(s) assigned to channel k (many-to-one: $\text{pref} \rightarrow \{\text{lt}, \text{st}\}$; $\text{ctx}, \text{reas} \rightarrow \text{ip}$; $\text{cf} \rightarrow \text{cf}$). Adapt then (i) normalizes τ_k into a channel schema (μ_k holds typed fields such as `long_term_preferences`, `dislikes`, `steps`, `anchor_item`), (ii) encodes τ_k with the shared Enc of Eq. (6), and (iii) writes to E_{lt} , E_{st} , or E_{ip} (Vector database) or, for $k = \text{cf}$, to E_{cf} as a Graph database edge pair $(u) \xrightarrow{\text{ANCHOR}} (i^+) \wedge (u) \xrightarrow{\text{CONTRAST}} (i^-)$ plus a rationale embedding (Listing 3). Initial extraction (`distill_to_memory`) and debate optimization (Section 3.8) both call the same Adapt.

Memory controller LLM_C . Raw Adapt output is not appended blindly: controller LLM_C inspects a text snapshot of the bank against a batch of new insights $B = \{(\tau, k, u)\}$ and emits structured edit ops o , which Apply commits in order:

$$o = \text{LLM}_C(\text{snapshot}(E), B), \\ o_i \in \{\text{ADD}, \text{DELETE}, \text{MODIFY}, \text{KEEP}\}, \quad (10) \\ E \leftarrow \text{Apply}(E, o).$$

Apply implements each op on the underlying stores: ADD calls Adapt then appends; DELETE removes by `entry_id`; MODIFY re-encodes revised τ and replaces metadata in-place; KEEP is a no-op. These four operators mirror the experience-library update rule in Training-Free GRPO (Youtu-Agent Team, 2025), specialized here to typed recommendation channels rather than general

Table 3

Four experience-memory channels: role, teacher source, storage, and retrieval filter.

k	Semantic role	Source p	Storage / filter
lt	Stable cross-session taste	pref	vector DB; $\text{user_id} = u$
st	Session-level trend / drift	pref	vector DB + timestamp; $\text{user_id} = u$
ip	Item impression vs. user	ctx, reas	vector DB; $\text{user_id} = u$, $\text{item} = c$
cf	Contrastive “if...” edge	cf	graph DB + vector DB; $\text{user_id} = u$

agent rollouts. The controller prompt enforces a maximum library size, deduplication (merge over add), and ≤ 60 -word actionable entries. *Advantage:* Eqs. (9)–(10) decouple *what* to remember (teacher or arbiter) from *how* to commit it; refinement cost scales with $|o|$, not $|E|$.

3.6.2. Four Experience-Memory Channels

We factor experience into four typed channels because prior LLM and sequential recommenders show that long-horizon taste (Shi et al., 2024; Wang et al., 2025), recent sequential context (Zheng et al., 2024; Zhang et al., 2024a), item-level semantics (Ren et al., 2024; Zhang et al., 2026), and hard-negative contrast (Song et al., 2026; Li et al., 2026) each contribute distinct ranking signal – and because storing them separately lets R_k retrieve only the evidence class needed for a decision. Table 3 summarizes the four channels. Each E_k is independently indexed, independently retrievable, and independently editable – the ablation in Section 5.2 drops individual k at inference without retraining LLM_S.

Long-term preference (lt). Motivated by evidence that long-horizon preference modeling improves recommendation beyond next-click prediction (Shi et al., 2024) and that external memory retrieval is needed when the LLM context window alone drops long-term history (Wang et al., 2025), this channel stores durable taste statements distilled from long_term_preferences and dislikes (pref pass), optionally with supporting reasoning. Entries are user-global: retrieval over $E_{lt}^u = \{e \in E_{lt} \mid \mu.\text{user_id} = u\}$ answers “what does this user generally like/dislike?” without binding to a specific candidate.

Short-term context (st). Motivated by sequential recommenders that show recent interactions dominate next-item prediction (Zheng et al., 2024) and that jointly modeling long- and short-term interests outperforms either alone (Zhang et al., 2024a), this channel captures transient interest from short_term_preferences (pref pass) plus optional context_reasoning. Each entry carries a timestamp in μ , enabling the bank to represent taste *drift* within a session or across recent batches while lt remains stable. Retrieval filters on user_id only.

Item perception (ip). Motivated by work showing that item-text representation quality drives LLM recommendation (Ren et al., 2024) and that token-centric attention

```
{
  "user_id": "...",
  "anchor_preference":
    "prefers animation,
     family-friendly tone",
  "target_item": "Frozen",
  "contrast_item":
    "The Dark Knight",
  "counterfactual_condition":
    "if user prefers dark,
     action-heavy stories",
  "counterfactual_outcome":
    "The Dark Knight would
     rank higher",
  "rationale_text": "...",
  "embedding": [],
  "timestamp": "..."
}
```

Figure 3: Distilled e_{cf} : the graph-edge instantiation of Eq. (9).

under-models item-level collaborative relations (Zhang et al., 2026), this is the most granular channel: (i) per-candidate impressions from candidate_perception (ctx), (ii) per-history-item descriptions from user_history_perception (ctx), and (iii) comparative reasoning from the five-step chain and reasoning_summary (reas). Candidate-specific entries additionally index target_movie_id in μ , so retrieval for (u, c) returns impressions scoped to c rather than unrelated items.

Counterfactual (cf). Motivated by recent LLM recommenders that separate hard negatives from noisy negatives (Song et al., 2026) and that use self-hard negatives to sharpen preference learning (Li et al., 2026), this channel stores hard-negative contrast rationales as typed graph edges: for anchor item i^+ and contrast i^- , μ records why_anchor_preferred, why_contrast_rejected, counterfactual_condition, and counterfactual_outcome. Dense retrieval runs over rationale_text embeddings; a graph-augmented pass appends any user-specific edges not surfaced by vector search. Listing 3 shows the instantiated schema.

Retrieval. At inference, the student queries each active channel with a shared query embedding $\mathbf{q}_u = \text{Enc}(\text{prompt}(u, C_u))$ and returns the top- m entries by dense similarity:

$$R_k(u, C_u) = \text{Top-}m \text{ sim}(\mathbf{q}_u, \mathbf{v}_e), \quad (11)$$

$e \in E_k^u$

where $E_k^u \subseteq E_k$ applies the filters in Table 3 (candidate scope c is applied only for ip), and sim is Eq. (6). Default $m=5$ per channel. *Advantage:* typed channels let R_k return only the evidence class needed for a ranking decision – stable taste (lt), recent drift (st), candidate fit (ip), or boundary conditions (cf) – rather than one undifferentiated reasoning blob.

3.7. LLM Student as Ranker

Motivation: if ranking gains required adapting student parameters, it would be unclear whether accuracy came

from the editable experience memory or from parameter adaptation; moreover, a per-backbone adapted student would need re-adaptation for every backbone, undermining the claim that E helps *any* teacher–student pair. *Design*: student LLM_S is a frozen pretrained LLM (3B–20B in our study). At inference it ranks solely by prompting with retrieved experience – never a fresh teacher call and never a weight update – following Eq. (5):

$$\begin{aligned} \hat{i} &= \arg \max_{c \in C_u} P_S(c \mid u) \\ &= \arg \max_{c \in C_u} P_S\left(c \mid x_u, \bigcup_{k \in \mathcal{K}} R_k(u, C_u)\right). \end{aligned} \quad (12)$$

Here $x_u = \text{prompt}(u, C_u)$ and $\bigcup_k R_k$ are concatenated into the student context; LLM_S is held fixed across bank updates. Unlike Eq. (3), retrieval is deterministic, and no student objective is optimized. *Advantage*: gains over zero-/few-shot and RAG – and over GraphRAG on most backbones (Section 5.1) – are attributable to the content and editability of E , not to student adaptation – the same frozen LLM_S improves when E improves (Section 5.4), and the protocol transfers across teacher–student pairs without per-student training (Section 5.3).

3.8. Experience Memory Optimization

Motivation: a bank populated once from teacher extraction can accumulate generic or conflicting entries; after the student ranks a case, the mismatch between the predicted list and the ground-truth target is a direct signal that E should be revised. *Design*: mirroring Training-Free GRPO (Youtu-Agent Team, 2025)’s loop of rollout $\rightarrow$ reward $\rightarrow$ semantic advantage $\rightarrow$ library edit – but specialized to recommendation ranking and enriched with K -agent debate – we update E after each student prediction: the student emits a ranked list under Eq. (12), reward models score that list against i^+ , K debating agents critique the case conditioned on those rewards, an arbiter synthesizes revised experience entries, and the same distillation controller commits them. This closes a loop predict $\rightarrow$ reward $\rightarrow$ debate $\rightarrow$ edit $\rightarrow E$ without changing LLM_S .

3.8.1. Reward Models

Motivation: a single scalar (e.g., only Hit@1) is too coarse for debate agents to diagnose *why* a ranking failed – missing the top item, burying it just outside the top- k , or placing it deep in the list require different memory edits. *Design*: given the student’s ordered list $\hat{L}_u = (c_{(1)}, \dots, c_{(L)})$ and ground-truth target i^+ , let $\text{rank} \in \{1, \dots, L\} \cup \{\emptyset\}$ be the 1-based position of i^+ in $\hat{L}_u$ ($\emptyset$ if absent). We pack four complementary IR-style rewards in $[0, 1]$ into the debate/arbiter prompt:

$$\begin{aligned} r(u) &= (\mathbf{1}[\text{rank}=1], \mathbf{1}[\text{rank} \leq k], \\ &\quad \min(1, 1/\text{rank}), 1/\log_2(\text{rank} + 1)) \in [0, 1]^4, \end{aligned} \quad (13)$$

with all components zero if $\text{rank} = \emptyset$ (default $k=10$). **Design rationale**. The four components are HR@1, HR@ k ,

Algorithm 1 Experience Memory Optimization after student prediction (one epoch)

Require: case batch U , frozen student LLM_S , agents $\{\text{LLM}_1, \dots, \text{LLM}_K\}$, arbiter LLM_A , rounds n_r

- 1: **for** $u \in U$ **do**
- 2: Rank $\hat{L}_u$ with frozen LLM_S via Eq. (12)
- 3: $r(u) \leftarrow$ reward models on $(\hat{L}_u, i^+)$ $\triangleright$ Eq. (13)
- 4: $\mathcal{T} \leftarrow \emptyset$
- 5: **for** round = 1, $\dots$, n_r ; agent $j = 1, \dots, K$ **do**
- 6: $\mathcal{T} \leftarrow \mathcal{T} \parallel \text{LLM}_j(\text{persona}_j, \mathcal{T}, u, \hat{L}_u, r(u))$ $\triangleright$ Eq. (14)
- 7: **end for**
- 8: $\{\tilde{e}_1, \dots, \tilde{e}_n\} \leftarrow \text{LLM}_A(\mathcal{T}, u, \hat{L}_u, r(u))$ $\triangleright$ Eq. (15)
- 9: **for** $\tilde{e} \in \{\tilde{e}_1, \dots, \tilde{e}_n\}$ **do**
- 10: $e \leftarrow \text{Adapt}(\tilde{e}); E \leftarrow \text{Apply}(E, o)$ $\triangleright$ Eqs. (9),(10)
- 11: **end for**
- 12: **end for**

reciprocal rank, and DCG-style position discount: they flag strict top-rank failure, near-misses, and graded credit when i^+ appears but not first. Importantly, $r(u)$ is *not* a training loss on LLM_S and is *not* used to $\arg \max$ among agent proposals; it conditions the natural-language critique so that edits target the observed ranking failure mode. *Advantage*: the same four signals transfer across datasets and students because they depend only on $(\hat{L}_u, i^+)$, keeping the optimization loop student-agnostic.

3.8.2. Debate and Arbiter after Each Prediction

Design lineage. The procedure combines the training-free experience-library update of Training-Free GRPO (Youtu-Agent Team, 2025) with iterative self-critique (Madaan et al., 2023) and multi-persona debate (Du et al., 2023), but applies them at memory-bank granularity and triggers them from the student’s post-prediction reward vector (Section 3.8.1).

Group computation. Over n_r rounds, K fixed-persona agents append free-form critiques to a shared transcript $\mathcal{T}$:

$$\begin{aligned} g_j &= \text{LLM}_j(\text{persona}_j, \mathcal{T}, u, \hat{L}_u, r(u)), \\ \mathcal{T} &\leftarrow \mathcal{T} \parallel g_j. \end{aligned} \quad (14)$$

No entry is committed during Eq. (14). After $n_r K$ turns, a single arbiter call synthesizes the experience set to commit:

$$\begin{aligned} \{\tilde{e}_1, \dots, \tilde{e}_n\} &= \text{LLM}_A(\mathcal{T}, u, \hat{L}_u, r(u)), \\ n &\leq n_{\max}. \end{aligned} \quad (15)$$

Eq. (15) is an LLM synthesis, not a programmatic vote or an $\arg \max_j$ over $\{g_j\}$ by reward.

Optimizing. Algorithm 1 runs, for each case: student ranking (Eq. 12) $\rightarrow$ reward models (Eq. 13) $\rightarrow$ debate/arbiter (Eqs. 14–15) $\rightarrow$ commit via Adapt and Apply. Section 5.4 tracks bank-quality signals and downstream HR@1/MRR over T such epochs.

Reuse of Knowledge Distillation. Each synthesized $\tilde{e}$ is committed by the *same* operators as regular extraction (Section 3.6.1): $e = \text{Adapt}(\tilde{e})$ (Eq. 9) and $E \leftarrow \text{Apply}(E, o)$ (Eq. 10). Optimization does not update LLM_S ; it only revises E .

4. Experimental Setup

4.1. Datasets

We evaluate on three recommendation datasets that span the explicit-implicit feedback spectrum. **ML-1M** (Harper and Konstan, 2015) provides 1–5 star explicit ratings and is a standard benchmark for LLM-based recommendation. **Amazon Beauty** (Ni et al., 2019) provides explicit ratings and review text but is substantially sparser per user, which stresses the memory bank’s ability to compensate for weak collaborative signal. **Steam** (Kang and McAuley, 2018) is an *implicit-feedback* dataset: it logs which games a user played rather than an explicit rating, so every logged interaction is treated as an equally-weighted positive signal (rating fixed to 1.0, positive threshold 0.0) instead of being thresholded from a graded rating scale. For all three datasets we apply a k -core filter ($k=20$ on ML-1M; $k=5$ on Beauty and Steam), build a chronological train/validation/test split, and construct 20-candidate ranking samples (1 positive, 19 sampled negatives). Dataset statistics after filtering are reported in Appendix H (Table 20).

4.2. Baselines

We compare against four prompting/retrieval baselines and rEDMRec, formalized in Eqs. (2)–(4) (Section 3.2): **Zero-shot** and **Few-shot** (Brown et al., 2020) ($K=0$ and $K>0$ in Eq. 2); **RAG** (Lewis et al., 2020) (Eq. 3), which retrieves raw historical interactions or reviews rather than distilled reasoning; and **GraphRAG** (Edge et al., 2024) (Eq. 4), which retrieves from a co-occurrence/knowledge graph built over items.

4.3. Models

Unless noted otherwise, the teacher is gpt-5.4-mini. We evaluate ten open student backbones spanning 3B–20B parameters: Qwen2.5 3B, Llama 3.1 8B (Touvron et al., 2023), Gemma-4-12B (Gemma Team, 2024), Minimax M2.5, Mixtral 8x7B (Jiang et al., 2024), Qwen3-14B (Yang et al., 2024), DeepSeek-R1-Distill-Qwen-14B (DeepSeek-AI, 2025), Phi-4, Llama 4 Scout, and GPT OSS 20B. Each student is used *frozen* (Section 3.7), so that cross-backbone gains isolate the contribution of the editable experience memory. The teacher-distillation study (Section 5.3) additionally fixes the student to gpt-5-mini (strong) or Qwen2.5 3B (small) while varying the teacher across seven backbones, to isolate the teacher’s contribution from the student’s.

4.4. Metrics and Protocol

We report Hit Rate at rank k ($\text{HR}@k$, $k \in \{1, 5, 10\}$), Normalized Discounted Cumulative Gain ($\text{NDCG}@k$, $k \in \{5, 10\}$), and Mean Reciprocal Rank (MRR), all computed over the fixed 20-candidate set per evaluation sample. For RQ1 tables we report Impv (%), the relative $\text{HR}@1$ improvement of rEDMRec over the second-best baseline on the same student ($\text{Impv} = (\text{Ours} - \text{SecondBest}) / \text{SecondBest} \times 100$), following RDRec (Wang et al., 2024b), together with

a McNemar p -value on $\text{HR}@1$ vs. that second-best method (approximate 2×2 contingency reconstructed from the table $\text{HR}@1$ rates at the full held-out sizes $n_{\text{ML-1M}}=49893$, $n_{\text{Beauty}}=1460$, $n_{\text{Steam}}=1460$; * marks $p<0.05$ when rEDMRec is ahead). Every (model, method, dataset) cell is evaluated on the *full held-out test split* (chronological train/validation/test; 20 candidates per sample with sampling seed 42), not a pilot subsample. The cross-dataset comparison in Section 5.1 (Amazon Beauty, Steam), the bank-scale analysis in Section 5.2, and the k -EPOCH and number-of-agents curves in Section 5.4 follow the same full-split evaluation protocol at each reported dataset, bank scale, and epoch/agent-count setting.

5. Results

We organize results around four research questions: does distilling reasoning into memory improve ranking over prompting-only and retrieval-only baselines (RQ1, Section 5.1)? which memory channel drives that improvement (RQ2, Section 5.2)? does teacher quality causally affect bank quality and downstream gain (RQ3, Section 5.3)? and does debate-based memory optimization measurably improve bank quality and downstream ranking (RQ4, Section 5.4)? We close with a qualitative case study of how individual memory entries evolve over training (Section 5.5).

5.1. RQ1: Does rEDMRec Improve Ranking Across Students and Datasets?

Table 4 reports $\text{HR}@k$, $\text{NDCG}@k$, and MRR for four representative students spanning our capacity range (Qwen2.5 3B, Llama 3.1 8B, Mixtral 8x7B, Qwen3-14B) on ML-1M; the full ten-model table is given in Appendix A. rEDMRec improves $\text{HR}@1$ over Zero-shot, Few-shot, and RAG for every student. Following RDRec (Wang et al., 2024b), Table 4 reports Impv (%) vs. the second-best baseline (typically GraphRAG): the largest relative gain is on the smallest student (Qwen2.5 3B, Impv = 13.3% over GraphRAG), consistent with structured memory helping most when parametric capacity is limited. Llama 3.1 8B is the clearest failure case – Impv is negative because GraphRAG remains ahead (Impv = −11.1%) – which we attribute to weak instruction-following rather than a deficiency of the memory itself (Section 4.3; Section 6); GPT OSS 20B similarly trails GraphRAG slightly (Impv = −3.3%), so the claim relative to GraphRAG is *most*, not *all*, students.

Table 5 reports Zero-shot vs. rEDMRec on Amazon Beauty and Steam for all ten student models under $\text{HR}@1$, $\text{NDCG}@10$, and MRR (full five-method matrices in Appendix B–C). The same ordering as on ML-1M holds: rEDMRec improves $\text{HR}@1$ over Zero-shot for every model on both datasets, and $\text{NDCG}@10$ / MRR rise in lockstep except on the weakest Llama 3.1 8B backbone, where ranking beyond top-1 remains flat. Absolute scores are lower than on ML-1M – as expected for sparser Amazon Beauty and

Table 4

Main results on ML-1M for four representative students (full ten-model table in Appendix A). Impv (%) is the relative HR@1 gain of rEDMRec over the *second-best* baseline on the same student, $\text{Impv} = (\text{Ours} - \text{SecondBest}) / \text{SecondBest} \times 100$, following RDRec (Wang et al., 2024b). p is the exact McNemar p -value for rEDMRec HR@1 vs. the second-best baseline (full held-out $n=49893$; approximate contingency from the table HR@1 rates). * marks $p<0.05$ with rEDMRec ahead. Best in **bold**, second-best underlined; rEDMRec rows labeled in bold.

Model	Method	HR@1↑	HR@5↑	HR@10↑	NDCG@5↑	NDCG@10↑	MRR↑	Impv (%)	p
Qwen2.5 3B	Zero-shot	0.12	0.28	0.38	0.20	0.23	0.18	–	–
	Few-shot	0.14	0.30	0.40	0.21	0.24	<u>0.20</u>	–	–
	RAG	0.13	0.29	0.39	0.20	0.23	<u>0.19</u>	–	–
	GraphRAG	<u>0.15</u>	<u>0.31</u>	<u>0.41</u>	<u>0.22</u>	<u>0.25</u>	<u>0.20</u>	–	–
	rEDMRec (ours)	0.17*	0.35	0.45	0.25	0.28	0.23	+13.3*	< 0.001*
Llama 3.1 8B	Zero-shot	0.07	0.15	0.23	<u>0.12</u>	0.14	<u>0.12</u>	–	–
	Few-shot	<u>0.08</u>	<u>0.16</u>	<u>0.24</u>	<u>0.12</u>	<u>0.15</u>	0.13	–	–
	RAG	<u>0.08</u>	<u>0.16</u>	<u>0.24</u>	<u>0.12</u>	0.14	<u>0.12</u>	–	–
	GraphRAG	0.09	0.17	0.25	<u>0.12</u>	<u>0.15</u>	0.13	–	–
	rEDMRec (ours)	<u>0.08</u>	0.17	0.25	0.13	0.16	0.13	-11.1	< 0.001
Mixtral 8x7B	Zero-shot	0.24	0.47	0.66	0.35	0.40	0.35	–	–
	Few-shot	0.26	0.49	0.68	0.37	<u>0.42</u>	0.36	–	–
	RAG	0.25	0.48	0.67	0.36	0.41	0.35	–	–
	GraphRAG	<u>0.27</u>	<u>0.50</u>	<u>0.69</u>	<u>0.38</u>	<u>0.42</u>	<u>0.37</u>	–	–
	rEDMRec (ours)	0.28*	0.52	0.71	0.39	0.44	0.38	+3.7*	< 0.001*
Qwen3-14B	Zero-shot	0.26	0.50	0.69	0.38	0.43	0.37	–	–
	Few-shot	0.28	0.52	0.71	<u>0.40</u>	<u>0.45</u>	0.39	–	–
	RAG	0.27	0.51	0.70	0.39	0.43	0.38	–	–
	GraphRAG	<u>0.29</u>	<u>0.53</u>	<u>0.72</u>	<u>0.40</u>	<u>0.45</u>	<u>0.40</u>	–	–
	rEDMRec (ours)	0.30*	0.55	0.73	0.42	0.47	0.41	+3.4*	< 0.001*

implicit-feedback Steam – yet Impv (%) vs. the second-best baseline (GraphRAG) is again largest on the smallest student (Qwen2.5 3B: Impv = 23.6% on Beauty and 21.5% on Steam; both $p<0.05$), consistent with memory compensating for weak collaborative signal when per-user history is short; mid-capacity Beauty students show smaller Impv (about 9–13%) that is not significant at $n=1460$.

5.2. RQ2: Which Memory Channel Matters?

To isolate each channel’s contribution, we remove one memory channel at a time and report $\Delta\text{HR@1}$ relative to the full-memory model. This ablation uses a separate panel of seven API-served backbones chosen to span four capacity tiers – strong (gpt-5-mini), mid (gpt-5.4-mini, Qwen3-32B, Minimax M2.5), saturated (GPT-OSS-120B), and weak (Llama 3.3 70B, Llama 3.1 8B) – rather than the ten local open-weight students used for the main comparison in Section 5.1; note that a backbone name can appear in both this ablation panel and the teacher panel of Section 5.3 (e.g., gpt-5.4-mini), where it plays a different role (ablated student vs. teacher) in a separate experiment. Table 6 summarizes the resulting pattern across the four capacity tiers. Short-term context is the only channel that is consistently important: removing it hurts every tier, from a strong student (gpt-5-mini, $\Delta=-0.04$) down to weak Groq-served Llama students ($\Delta \approx -0.01$ to -0.02). The long-term preference, item-perception, and counterfactual

channels show a *reversed* ablation on the strongest student: removing them *improves* HR@1 by +0.03 to +0.04, whereas removing the counterfactual channel *hurts* the mid-capacity student ($\Delta=-0.04$) and has little effect on the saturated 120B-parameter student, which appears to ignore the bank altogether (full-memory $\Delta \approx 0.00$ for that tier).

We consider three, non-exclusive explanations for the reversed sign on strong students, in decreasing order of the evidence we can bring to bear with the current instrumentation. First, *channel redundancy*: long-term preference statements often restate information already present in the raw history block of the prompt, so a strong student that already attends well to raw history gains nothing extra and instead pays a small “distraction” cost for the redundant text. Second, *generic, low-specificity entries*: item-perception entries produced early in the bank’s lifecycle tend to be long and only loosely actionable (Section 5.5 quantifies this directly via a specificity score). Third, *conflicting signals*: a counterfactual edge can push a plausible but ultimately incorrect candidate above the true target when its hypothetical condition partially matches the current user. Only the mid-capacity student, which cannot yet extract the same information unaided from raw history, benefits from the counterfactual channel unconditionally; the fact that the sign of this effect depends on student capacity, rather than being fixed, is itself evidence against treating any single channel

Table 5

Cross-dataset generalization on Amazon Beauty and Steam (full held-out test split, 20 candidates/sample, seed 42). ZS = Zero-shot; Ours = rEDMRec (**bold**). Impv (%) is the relative HR@1 gain of rEDMRec over the *second-best* baseline on the same student, Impv = (Ours – SecondBest)/SecondBest $\times$ 100, following RDRec (Wang et al., 2024b). p = McNemar p -value for Ours HR@1 vs. the second-best baseline ($n_{\text{Beauty}}=1460$, $n_{\text{Steam}}=1460$; * if $p<0.05$). Complete five-method matrices are in Appendix B–C.

Dataset	Model	HR@1(ZS)	HR@1(Ours)	NDCG@10(ZS)	NDCG@10(Ours)	MRR(ZS)	MRR(Ours)	Impv (%)	p
Amazon Beauty	Qwen2.5 3B	0.092	0.136*	0.218	0.262	0.180	0.224	+23.6*	0.037*
	Llama 3.1 8B	0.062	0.071	0.180	0.180	0.162	0.162	-4.1	0.830
	Gemma-4-12B	0.140	0.178	0.290	0.328	0.252	0.290	+12.7	0.166
	Minimax M2.5	0.152	0.187	0.308	0.343	0.264	0.290	+10.0	0.247
	Mixtral 8x7B	0.164	0.202	0.320	0.358	0.282	0.311	+11.0	0.188
	Qwen3-14B	0.176	0.211	0.338	0.373	0.294	0.329	+8.8	0.269
	DeepSeek-R1	0.188	0.223	0.350	0.385	0.312	0.329	+8.3	0.280
	Distill-Qwen-14B	0.170	0.205	0.332	0.393	0.288	0.340	+9.0	0.264
	Phi-4	0.182	0.217	0.344	0.405	0.300	0.344	+8.5	0.275
	GPT OSS 20B	0.182	0.199	0.344	0.370	0.300	0.317	-0.5	1.000
Steam	Qwen2.5 3B	0.106	0.158*	0.224	0.276	0.180	0.232	+21.5*	0.035*
	Llama 3.1 8B	0.066	0.076	0.180	0.180	0.162	0.162	-7.3	0.584
	Gemma-4-12B	0.170	0.208	0.320	0.358	0.276	0.314	+7.2	0.356
	Minimax M2.5	0.186	0.224	0.344	0.382	0.292	0.321	+6.7	0.394
	Mixtral 8x7B	0.202	0.244	0.360	0.402	0.316	0.349	+8.0	0.276
	Qwen3-14B	0.218	0.256	0.384	0.422	0.332	0.370	+5.8	0.392
	DeepSeek-R1	0.234	0.272	0.400	0.438	0.356	0.375	+5.4	0.426
	Distill-Qwen-14B	0.210	0.248	0.376	0.443	0.324	0.382	+6.0	0.411
	Phi-4	0.226	0.264	0.392	0.459	0.340	0.388	+5.6	0.421
	GPT OSS 20B	0.226	0.245	0.392	0.421	0.340	0.359	-2.0	0.797

Table 6

Channel importance by capacity tier ($\Delta\text{HR@1}$ vs. full memory). Negative = beneficial channel; positive = reversed ablation (Section 5.2).

Channel	Strong	Mid	Saturated	Weak
Short-term context	-0.04	-0.04	≈ -0.01	-0.01--0.02
Long-term preference	+0.03	0.00	≈ 0.00	+0.01
Item-perception	+0.04	0.00	≈ 0.00	slight
Counterfactual	+0.03	-0.04	weak	≈ 0.00
Full memory	-0.01	-0.03	≈ 0.00	weak

as universally useful or harmful. Figure 4 shows this pattern holds across all seven backbones in the ablation panel, and Figure 5 shows that the reversed channels flip sign as the *bank* grows from the sparse-bank anchor ($B=189$) toward 10^5 – 10^6 entries, because a denser bank dilutes the redundant and generic entries that drive the reversal at small B (bank size is independent of the full-test evaluation protocol in Section 4.4). Per-backbone Δ plots, the MRR heatmap, and the full numeric ablation matrix are deferred to Appendix G.

5.3. RQ3: Does Teacher Quality Causally Affect Downstream Gain?

Holding the student fixed and varying the teacher isolates the teacher’s contribution from the student’s. Table 7 fixes the student to gpt-5-mini (strong) across seven teachers; Table 8 repeats the comparison with Qwen2.5 3B (small).

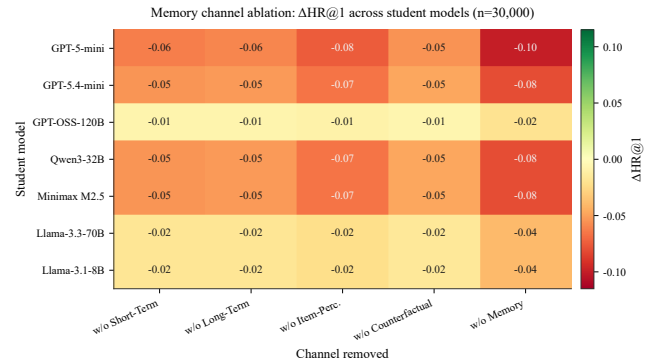


Figure 4: Channel ablation heatmap: $\Delta\text{HR@1}$ for each of the seven ablation-panel backbones (rows, Section 5.2) and four memory channels (columns). Green cells indicate a reversed ablation (removal helps); red cells indicate the channel is beneficial.

Across both tables, a lower bank duplicate rate is a leading indicator of downstream gain: gpt-5.4-mini has the lowest duplicate rate among the strongest teachers (12.4%) and the largest $\Delta\text{HR@1}$ for the strong student (+0.060), while Llama 3.1 8B Instant – the only teacher below 14B parameters in this comparison – has by far the highest duplicate rate (22.8%) and the smallest gain (+0.015), consistent with a weak teacher producing generic, repetitive entries (e.g., repeated “*user likes drama*” statements without item-specific detail) that carry little retrieval value. The relationship is

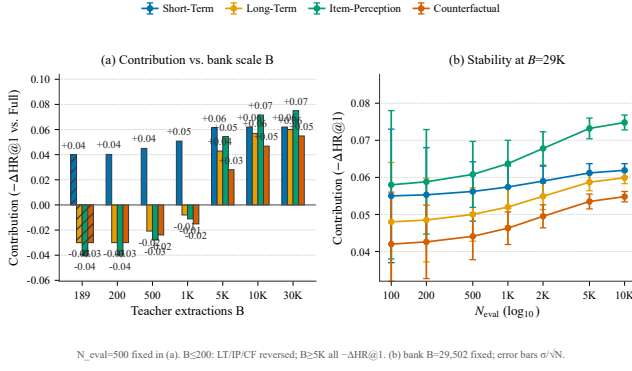


Figure 5: Per-channel contribution ($-\Delta\text{HR@1}$) as bank scale B grows from the sparse-bank anchor ($B=189$) to 10^6 entries (panel a), and stability of the estimate as evaluation sample size grows under the full-test protocol at fixed bank scale (panel b). Note: B is bank size, not the evaluation split size.

Table 7

Teacher-distillation effectiveness, fixed student: gpt-5-mini (strong). Zero-shot baseline $\text{HR@1} = 0.28$, $\text{MRR} = 0.403$.

Teacher	Dup.%↓	$\Delta\text{HR@1}\uparrow$	$\Delta\text{MRR}\uparrow$	HR@1
gpt-5.4-mini	12.4	+0.060	+0.0687	0.340
Qwen3 32B (131k)	11.5	+0.055	+0.0620	0.335
gpt-5-mini	14.1	+0.050	+0.0580	0.330
Llama 3.3 70B (128k)	10.5	+0.045	+0.0520	0.325
GPT OSS 120B (128k)	9.8	+0.040	+0.0450	0.320
Minimax M2.5	13.2	+0.040	+0.0480	0.320
Llama 3.1 8B Instant	22.8	+0.015	+0.0180	0.295

not perfectly monotonic, however: GPT OSS 120B has the lowest duplicate rate of any teacher (9.8%) yet only a mid-dling $\Delta\text{HR@1}$ (+0.040), which we attribute to a student-capacity ceiling – gpt-5-mini cannot fully exploit the additional verbosity of a 120B-parameter teacher’s bank. This ceiling is sharper for the small student: Table 8 shows the GPT OSS 120B bank drops *below* the more concise Qwen3 32B bank once the student itself is small (Qwen2.5 3B), even though GPT OSS 120B produces the least duplicated bank of the two. Together, these two tables support a causal chain in which teacher quality first improves bank quality (lower duplication), and bank quality only converts into downstream gain up to a ceiling set by the student’s own capacity to use additional bank detail.

5.4. RQ4: Does Debate-Based Memory Optimization Improve Bank Quality and Downstream Ranking?

We next ask whether the debate-and-arbiter optimization procedure (Section 3.8) is doing useful work, rather than merely adding cost. Figure 6 tracks bank-quality signals and downstream ranking jointly over six k -EPOCHs of debate (three debate agents, one round per epoch). Bank quality improves and saturates: the duplicate rate drops by 7.4

Table 8

Teacher-distillation effectiveness, fixed student: Qwen2.5 3B (small). Zero-shot baseline $\text{HR@1} = 0.12$, $\text{MRR} = 0.18$.

Teacher	Dup.%↓	$\Delta\text{HR@1}\uparrow$	$\Delta\text{MRR}\uparrow$	HR@1
gpt-5.4-mini	12.4	+0.050	+0.0550	0.170
Qwen3 32B (131k)	11.5	+0.045	+0.0500	0.165
gpt-5-mini	14.1	+0.042	+0.0470	0.162
Llama 3.3 70B (128k)	10.5	+0.038	+0.0430	0.158
GPT OSS 120B (128k)	9.8	+0.034	+0.0380	0.154
Minimax M2.5	13.2	+0.033	+0.0380	0.153
Llama 3.1 8B Instant	22.8	+0.010	+0.0120	0.130

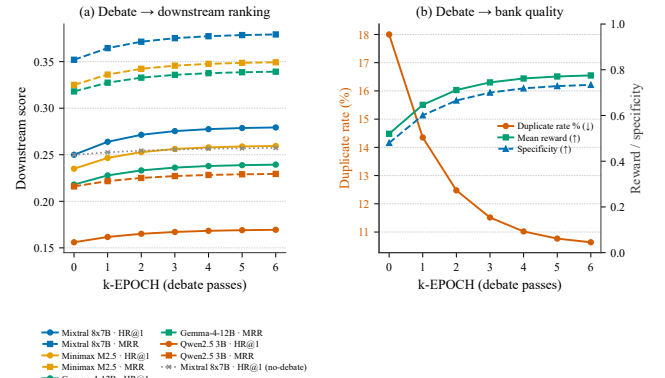


Figure 6: Debate-based memory optimization vs. k -EPOCH. (a) Downstream HR@1 /MRR for student models plus a no-debate control. (b) Bank-quality signals: duplicate rate (down is better) and mean experience reward / specificity (up is better).

percentage points (18.0% $\rightarrow$ 10.6%) while mean experience reward rises by 0.255 (0.52 $\rightarrow$ 0.78). Downstream ranking tracks this curve rather than moving independently of it: the strongest student in this comparison (Mixtral 8x7B) gains +0.029 HR@1 (0.250 $\rightarrow$ 0.279) over the same six epochs, while a no-debate paraphrase control – which perturbs entry wording without the critique-and-revise debate loop – stays flat, isolating the debate mechanism (rather than any wording change) as the source of the gain. The smallest student in this comparison (Qwen2.5 3B) gains only +0.013 HR@1 (0.156 $\rightarrow$ 0.169) from the identical optimized bank, reproducing the capacity ceiling from Section 5.3 in a different experiment: most of the gain lands within the first two to three epochs for every student, so debating past $k=3$ epochs is rarely worth the added LLM cost.

A separate sweep over the *number* of debating agents ($k = 1, \dots, 10$, one epoch) shows quality rising with the diversity of critique but saturating, while LLM cost grows linearly in k (Table 9, Figure 7). The knee of quality-per-cost is at $k^* = 4$: increasing k from 1 to 4 lifts HR@1 by +0.022, but increasing k from 4 to 10 adds only +0.006 for six additional LLM calls per case, which is not a favorable trade for most deployment budgets. Tabular controller ablations

Table 9

Number-of-debating-agents sweep ($k = 1..10$), reference student Mixtral 8x7B. k^* marks the quality-per-cost knee.

k	HR@1↑	Specificity↑	Dup.%↓	Calls/case
1	0.255	0.520	16.0	2
2	0.266	0.608	13.4	3
3	0.273	0.663	11.9	4
4 (k^*)	0.277	0.699	11.0	5
5	0.279	0.721	10.4	6
6	0.281	0.735	10.0	7
7	0.282	0.744	9.8	8
8	0.282	0.750	9.7	9
9	0.282	0.754	9.6	10
10	0.283	0.756	9.6	11

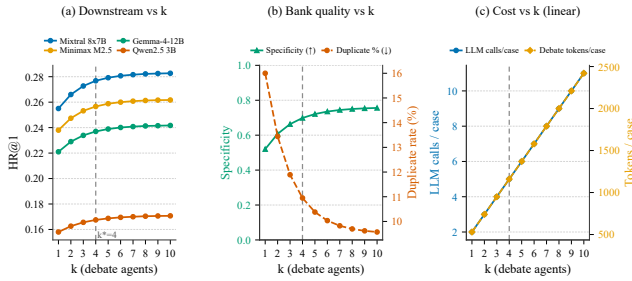


Figure 7: Number-of-agents sweep: downstream HR@1, bank-quality signals, and LLM cost as a function of k .

(full debate vs. no-debate paraphrase vs. post-extraction) and epoch snapshots are collected in Appendix F.

5.5. Qualitative Case Study: How Do Memory Entries Evolve?

The ablation and debate-optimization results above are aggregate signals; to make the mechanism concrete, we directly compare the earliest and the latest persisted entry per user in each memory channel, using a deterministic, LLM-free specificity score (specificity combines concreteness, genre-term coverage, lexical diversity, and a hedge-language penalty into a single $[0, 1]$ score, with no additional LLM calls). Table 10 summarizes the resulting before/after deltas across users with at least two persisted versions. Three of the four channels become more specific and less hedged over training; the item-perception channel changes the most (mean specificity $0.405 \rightarrow 0.488$), moving from generic fallback language to item-grounded, actionable statements. The short-term context channel is the exception: its mean specificity *decreases* slightly ($0.433 \rightarrow 0.401$), which Figure 8 and the cases below show is not a quality regression but a compression effect – verbose narrative entries are replaced by short, conditional “session rules” that are less lexically diverse by construction but more directly actionable by the student.

Table 11 shows three representative before→after pairs drawn from the persisted bank (one long-term, one item-perception, one short-term). **Case A** (long-term, user 2) replaces a single-film hedge (“Very limited data...”) with a

Table 10

Bank evolution: specificity before (earliest persisted entry) vs. after (latest persisted entry) per channel, over users with ≥ 2 versions.

Channel	Spec. before	Spec. after	Δ
Long-term preference	0.500	0.520	+0.021
Short-term context	0.433	0.401	−0.032
Item-perception	0.405	0.488	+0.084
Counterfactual / hard-neg.	0.478	0.519	+0.042

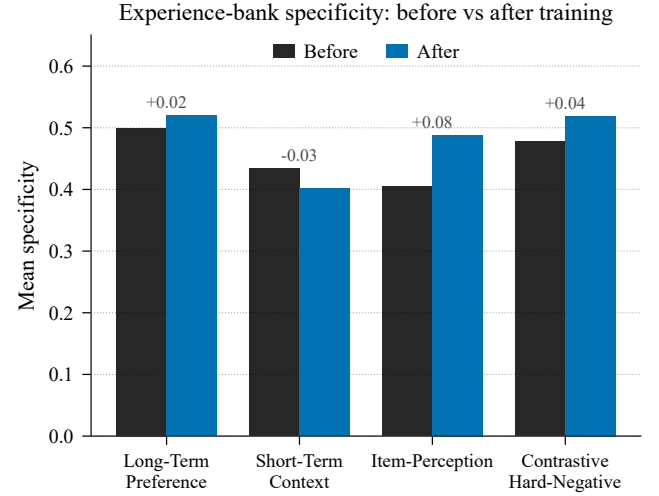


Figure 8: Mean specificity before vs. after training, by memory channel.

ranking-ready taste statement that names era, franchise type, and an upweight rule, lifting specificity from 0.214 to 0.479. **Case B** (item-perception, user 4) converts a vague Hit@1 complaint into an item-grounded rule keyed to *Rocky (1976)*, with an explicit +30–50% tie-break boost – the largest single-entry Δ Spec. in the sample (+0.500). **Case C** (short-term, user 2) illustrates the compression pattern behind the negative mean Δ on that channel: an empty “no emerging interests” note becomes a short session rule with diversity constraints; lexical diversity drops relative to long narrative entries, but actionability rises. Together the three cases show that debate-driven Add/Modify operations do not merely paraphrase entries – they accumulate confirmed signals into shorter, more specific, ranking-oriented memory.

6. Discussion and Limitations

Operating-point generalization. The bank-scale analysis (Section 5.2) and the k -EPOCH and number-of-agents curves (Section 5.4) span a wide range of settings – bank sizes up to 10^6 entries, up to six debate epochs, and up to ten debating agents. A practitioner adopting a specific operating point (for example, a specific k -EPOCH budget or bank size) in a production system should re-confirm behavior at that exact point, since a trend measured across a range does not guarantee identical behavior at every intermediate setting.

Table 11

Qualitative before→after memory entries (persisted bank). Spec. is the deterministic specificity score in [0, 1].

Case / channel	Spec.	Before (earliest)	After (latest)
A / lt	0.214 → 0.479	<i>"Very limited data: the only rated film is a classic action-adventure, suggesting a preference for high-energy, heroic, escapist storytelling. . . Dislikes: No explicit dislikes can be inferred. . ."</i>	<i>"Long-term: favors mainstream 1990s action buddy-cop films – franchise sequels, star-driven chemistry, high-energy action with comedic interplay. Upweight these attributes in ranking but require repeat confirmations. . ."</i>
B / ip	0.350 → 0.850	<i>"The strongest match was only ranked 5th, so Hit@10 was good but Hit@1/MRR suffered. For this user, place the best Rocky-like inspirational drama at rank 1 whenever possible."</i>	<i>"Rocky (1976): User 4 chose Rocky over a higher-ranked 80s action title, signaling a preference for 1970s character-driven underdog sports dramas. When ranking, boost similar 1970s character dramas above mainstream 80s action (recommend +30–50% score in tie scenarios)."</i>
C / st	0.300 → 0.500	<i>"No recent viewing items were provided, so no emerging short-term interests can be detected."</i>	<i>"Session: prioritize late-80s/90s Hollywood action comedies with buddy dynamics and franchise entries for top slots; include at least one diverse alternative per slate to avoid popularity bias."</i>

Teacher coverage. The main results (Section 5.1) fix the teacher to gpt-5.4-mini; the teacher-distillation study (Section 5.3) varies the teacher but only against two fixed students. We have not measured the full teacher × student cross-product, so it remains open whether the teacher-quality effect observed for gpt-5-mini and Qwen2.5 3B holds uniformly across all ten students in Table 12.

Explanation faithfulness. rEDMRec’s student can emit a short explanation grounded in retrieved memory (Section 3.7), but this paper evaluates ranking quality, not whether the emitted explanation is faithful to the memory it cites; a human evaluation of explanation faithfulness and plausibility, as outlined in our evaluation plan, is left to future work.

Backbone-dependent returns. The architecture assumes the student can follow a moderately complex prompt that concatenates user context, candidate descriptions, and retrieved memory snippets. Section 5.1 shows this assumption breaks down for at least one backbone (Llama 3.1 8B), whose negative Impv vs. GraphRAG we attribute to weak instruction-following rather than to the memory being unhelpful in principle; the near-saturated 20B-parameter student likewise trails GraphRAG slightly. Memory still lifts every student over Zero-shot/Few-shot/RAG, but not always over GraphRAG – Appendix K tabulates these failure cells and borderline controller edits. Practically, the value of rEDMRec’s added system complexity is highest for small-to-mid capacity students.

Domain scope. Our three datasets cover movie, beauty-product, and game recommendation with English-language metadata; we have not tested domains with substantially different item-description structure (e.g., short-video or news recommendation), and the four-channel schema, in particular the counterfactual channel, was designed with catalog items that have stable, comparable attributes in mind.

7. Conclusion

This paper addresses the problem of reusing, rather than repeating, LLM reasoning across recommendation requests, by proposing rEDMRec, an architecture that distills teacher reasoning into a four-channel, editable experience memory and serves ranking requests from a lightweight student that only retrieves from this memory. The key idea is to separate an infrequent, expensive reasoning-compression process – teacher extraction, distillation, and debate-based memory optimization – from a frequent, cheap inference process, which decouples recommendation quality from per-request reasoning cost. Across ten student LLMs and three datasets, this design improves HR@1, NDCG, and MRR over zero-shot, few-shot, and RAG on every backbone, and over GraphRAG on most backbones (exceptions: Llama 3.1 8B and GPT OSS 20B), with the largest RDRc-style Impv (%) vs. the second-best baseline on the smallest students; our channel-ablation, teacher-distillation, and debate-optimization studies further show that short-term context is the only consistently beneficial channel across capacity tiers (with long-term, item-perception, and counterfactual effects capacity-dependent), that bank duplication is a leading indicator of downstream gain up to a student-capacity ceiling, and that *K*-agent debate measurably improves bank quality in a way that propagates into ranking accuracy rather than being a cosmetic refinement. Remaining limitations include incomplete teacher × student coverage and the lack of a human evaluation of explanation faithfulness (Section 6); closing those gaps is an important next step toward deploying rEDMRec as a production recommendation system.

Data and Code Availability

The preprocessing, training, and evaluation code, together with the JSON experiment matrices used to produce every table and figure in this paper, are organized under the rEDMRec/ project root (see readme.md for the end-to-end

quick-start pipeline). ML-1M, Amazon Beauty, and Steam are third-party datasets redistributed under their original licenses; this work releases only derived, de-identified interaction records and memory-bank artifacts.

CRedit authorship contribution statement

Minh Hoang Nguyen: Methodology, Conceptualization, Writing – original draft, Writing – review & editing. **Tung Le:** Supervision, Supporting, Writing – review & editing. **Huy Tien Nguyen:** Supervision, Supporting, Conceptualization, Project administration.

References

- Bao, K., Zhang, J., Zhang, Y., Wang, W., Feng, F., He, X., 2023. TALLRec: An effective and efficient tuning framework to align large language model with recommendation, in: Proceedings of the 17th ACM Conference on Recommender Systems, pp. 1007–1014. doi:10.1145/3604915.3608857.
- Bismay, M., Dong, X., Caverlee, J., 2025. ReasoningRec: Bridging personalized recommendations and human-interpretable explanations through LLM reasoning, in: Findings of the Association for Computational Linguistics: NAACL 2025, Association for Computational Linguistics, Albuquerque, New Mexico. pp. 8147–8163. URL: <https://aclanthology.org/2025.findings-naacl.454/>.
- Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al., 2020. Language models are few-shot learners. Advances in Neural Information Processing Systems (NeurIPS).
- DeepSeek-AI, 2025. DeepSeek-R1: Incentivizing reasoning capability in llms via reinforcement learning. arXiv preprint arXiv:2501.12948.
- Du, Y., Li, S., Torralba, A., Tenenbaum, J.B., Mordatch, I., 2023. Improving factuality and reasoning in language models through multiagent debate. arXiv preprint arXiv:2305.14325.
- Edge, D., Trinh, H., Cheng, N., Bradley, J., Chao, A., Mody, A., Truitt, S., Larson, J., 2024. From local to global: A graph RAG approach to query-focused summarization. arXiv preprint arXiv:2404.16130.
- Gao, C., Chen, R., Yuan, S., Huang, K., Yu, Y., He, X., 2025. SPRec: Self-play to debias LLM-based recommendation. arXiv preprint arXiv:2412.09243.
- Gao, Y., Sheng, T., Xiang, Y., Xiong, Y., Wang, H., Zhang, J., 2023. ChatREC: Towards interactive and explainable LLMs-augmented recommender system. arXiv preprint arXiv:2303.14524.
- Gemma Team, 2024. Gemma: Open models based on gemini research and technology. arXiv preprint arXiv:2403.08295.
- Gu, H., Zhong, R., Xia, Y., Yang, W., Lu, C., Jiang, P., Gai, K., 2025. R⁴ec: A reasoning, reflection, and refinement framework for recommendation systems, in: Proceedings of the Nineteenth ACM Conference on Recommender Systems, ACM, Prague, Czech Republic. pp. 411–421. doi:10.1145/3705328.3748068.
- Han, D., Song, H., Yi, M.Y., 2025. Rethinking LLM-based recommendations: A personalized query-driven parallel integration. arXiv preprint arXiv:2504.11889 QueREC.
- Harper, F.M., Konstan, J.A., 2015. The MovieLens datasets: History and context. ACM Transactions on Interactive Intelligent Systems 5, 1–19.
- Hou, Y., Zhang, J., Lin, Z., Lu, H., Xie, R., McAuley, J., Zhao, W.X., 2024. Large language models are zero-shot rankers for recommender systems, in: European Conference on Information Retrieval, Springer. pp. 364–381.
- Jiang, A.Q., Sablayrolles, A., Roux, A., Mensch, A., Savary, B., Bamford, C., Chaplot, D.S., Casas, D.d.l., Hanna, E.B., Bressand, F., et al., 2024. Mixtral of experts. arXiv preprint arXiv:2401.04088.
- Kang, W.C., McAuley, J., 2018. Self-attentive sequential recommendation, in: IEEE International Conference on Data Mining (ICDM), pp. 197–206.
- Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.t., Rocktäschel, T., Riedel, S., Kiela, D., 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks, in: Advances in Neural Information Processing Systems (NeurIPS), pp. 9459–9474.
- Li, B., Zheng, B., Wang, X., Zhang, L., Wang, J., Chen, S., Zhao, W.X., Wen, J.R., 2026. Improving LLM-based recommendation with self-hard negatives from intermediate layers. arXiv preprint arXiv:2602.17410.
- Li, L., Zhang, Y., Chen, L., 2023. Prompt distillation for efficient LLM-based recommendation, in: Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, pp. 1348–1357. doi:10.1145/3583780.3615017.
- Li, L., Zhang, Y., Liu, D., Chen, L., 2024. Large language models for generative recommendation: A survey and visionary discussions, in: Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), Torino, Italia. pp. 10146–10159.
- Liao, J., Li, S., Yang, Z., Wu, J., Yuan, Y., Wang, X., He, X., 2024. LLaRA: Large language-recommendation assistant, in: Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1785–1795. doi:10.1145/3626772.3657760.
- Lin, J., Dai, X., Xi, Y., Liu, W., Chen, B., Zhang, H., Liu, Y., Wu, C., Li, X., Zhu, C., et al., 2025. How can recommender systems benefit from large language models: A survey. ACM Transactions on Information Systems Also arXiv:2306.05817.
- Liu, Q., Wu, X., Zhao, X., Zhu, Y., Zhang, Z., Tian, F., Zheng, Y., 2024. Large language model distilling medication recommendation model. arXiv preprint arXiv:2402.02803 LEADER.
- Liu, Q., Zhao, X., Wang, Y., Wang, Y., Zhang, Z., Sun, Y., Li, X., Wang, M., Jia, P., Lin, K., et al., 2025. Large language model enhanced recommender systems: A survey. arXiv preprint arXiv:2412.13432.
- Lyu, H., Jiang, S., Zeng, H., Xia, Y., Wang, Q., Zhang, S., Chen, R., Leung, C., Tang, J., Luo, J., 2024. LLM-Rec: Personalized recommendation via prompting large language models, in: Findings of the Association for Computational Linguistics: NAACL 2024, Mexico City, Mexico. pp. 583–612. doi:10.18653/v1/2024.findings-naacl.39.
- Madaan, A., Tandon, N., Gupta, P., Hallinan, S., Gao, L., Wiegrefe, S., Alon, U., Dziri, N., Prabhume, S., Yang, Y., et al., 2023. Self-refine: Iterative refinement with self-feedback, in: Advances in Neural Information Processing Systems (NeurIPS), pp. 46534–46594.
- Nguyen, M.H., 2026. Vihorec: A quality-controlled vietnamese hotel recommendation dataset and cold-start benchmark. arXiv preprint arXiv:2607.12946.
- Nguyen, M.H., Nguyen, T.T., Ta, M.N., Le, T., Nguyen, H.T., 2025. Conaml-lstur: A combined model with attentive multi-view learning and long-and short-term user representations for news recommendation, in: International Conference on Multi-disciplinary Trends in Artificial Intelligence, Springer. pp. 106–119.
- Nguyen, M.H., Nguyen, T.T., Ta, M.N., Nguyen, T.M., Nguyen, K.V., 2024. RRs: Review-based recommendation system using deep learning for vietnamese. SN Computer Science 5, 492.
- Nguyen, M.H., Thiet, S.N., 2025. Enhancing ocr for sino-vietnamese language processing via fine-tuned paddleocrv5. arXiv preprint arXiv:2510.04003.
- Ni, J., Li, J., McAuley, J., 2019. Justifying recommendations using distantly-labeled reviews and fine-grained aspects, in: Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 188–197.
- Packer, C., Fang, V., Patil, S.G., Lin, K., Wooders, S., Gonzalez, J.E., 2023. MemGPT: Towards llms as operating systems. arXiv preprint arXiv:2310.08560.
- Park, J.S., O'Brien, J., Cai, C.J., Morris, M.R., Liang, P., Bernstein, M.S., 2023. Generative agents: Interactive simulacra of human behavior, in: ACM Symposium on User Interface Software and Technology (UIST), pp. 1–22.
- Qu, H., Fan, W., Zhao, Z., Li, Q., 2024. TokenRec: Learning to tokenize ID for LLM-based generative recommendation. arXiv preprint arXiv:2406.10450.

- Reimers, N., Gurevych, I., 2019. Sentence-BERT: Sentence embeddings using siamese BERT-networks, in: Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 3982–3992.
- Ren, X., Wei, W., Xia, L., Su, L., Cheng, S., Wang, J., Yin, D., Huang, C., 2024. Representation learning with large language models for recommendation, in: Proceedings of the ACM Web Conference 2024, pp. 3464–3475. doi:10.1145/3589334.3645458.
- Shi, W., He, X., Zhang, Y., Gao, C., Li, X., Zhang, J., Wang, Q., Feng, F., 2024. Large language models are learnable planners for long-term recommendation, in: Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1893–1903.
- Song, T., Chao, W.S., Liu, H., 2026. Hard vs. noise: Resolving hard-noisy sample confusion in recommender systems via large language models, in: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 15743–15751. doi:10.1609/aaai.v40i18.38605.
- Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al., 2023. Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288.
- Wang, C., Zhang, Y., Zhu, F., Zhang, J., Shi, T., Feng, F., 2025. Leveraging memory retrieval to enhance llm-based generative recommendation, in: Companion Proceedings of the ACM on Web Conference 2025, pp. 1346–1350.
- Wang, Q., Li, J., Wang, S., Xing, Q., Niu, R., Kong, H., Li, R., Long, G., Chang, Y., Zhang, C., 2024a. Towards next-generation LLM-based recommender systems: A survey and beyond. arXiv preprint arXiv:2410.19744.
- Wang, X., Cui, J., Suzuki, Y., Fukumoto, F., 2024b. RDRec: Rationale distillation for LLM-based recommendation, in: Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Bangkok, Thailand. pp. 65–74. doi:10.18653/v1/2024.acl-short.6.
- Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q., Zhou, D., 2022. Chain-of-thought prompting elicits reasoning in large language models, in: Advances in Neural Information Processing Systems (NeurIPS), pp. 24824–24837.
- Wei, W., Ren, X., Tang, J., Wang, Q., Su, L., Cheng, S., Wang, J., Yin, D., Huang, C., 2024. LLMRec: Large language models with graph augmentation for recommendation, in: Proceedings of the 17th ACM International Conference on Web Search and Data Mining, pp. 806–815. doi:10.1145/3616855.3635853.
- Woźniak, S., Duszko, J., Kocoń, J., Kazienko, P., 2025. Improving llm-based recommender systems with user-controllable profiles, in: Companion Proceedings of the ACM on Web Conference 2025, pp. 2102–2111.
- Wu, L., Zheng, Z., Qiu, Z., Wang, H., Gu, H., Shen, T., Qin, C., Zhu, C., Zhu, H., Liu, Q., Xiong, H., Chen, E., 2024. A survey on large language models for recommendation. arXiv preprint arXiv:2305.19860.
- Yang, A., Yang, B., Hui, B., Zheng, B., Yu, B., Zhou, C., Li, C., Li, C., Liu, D., Huang, F., et al., 2024. Qwen2 technical report. arXiv preprint arXiv:2407.10671.
- Youtu-Agent Team, 2025. Training-free group relative policy optimization. arXiv preprint arXiv:2510.08191 URL: <https://arxiv.org/abs/2510.08191>.
- Yue, Z., Rabhi, S., Moreira, G.d.S.P., Wang, D., Oldridge, E., 2023. LlamaRec: Two-stage recommendation using large language models for ranking. arXiv preprint arXiv:2311.02089.
- Zhang, J., Xie, R., Hou, Y., Zhao, W.X., Lin, L., Wen, J.R., 2023. Recommendation as instruction following: A large language model empowered recommendation approach. arXiv preprint arXiv:2305.07001.
- Zhang, X., He, B., Chen, J., Cui, Z., Ma, C., 2026. From token to item: Enhancing large language models for recommendation via item-aware attention mechanism, in: Proceedings of the ACM Web Conference 2026, pp. 6700–6708.
- Zhang, X., Li, B., Jin, B., 2024a. Denoising long- and short-term interests for sequential recommendation, in: Proceedings of the 2024 SIAM International Conference on Data Mining (SDM), pp. 544–552. doi:10.1137/1.9781611978032.63.
- Zhang, Y., Bao, K., Yan, M., Wang, W., Feng, F., He, X., 2024b. Text-like encoding of collaborative information in large language models for recommendation, in: Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Bangkok, Thailand. pp. 9181–9191. doi:10.18653/v1/2024.acl-long.497.
- Zhang, Y., Feng, F., Zhang, J., Bao, K., Wang, Q., He, X., 2025a. CoLLM: Integrating collaborative embeddings into large language models for recommendation. IEEE Transactions on Knowledge and Data Engineering Also arXiv:2310.19488.
- Zhang, Y., Xu, W., Zhao, X., Wang, W., Feng, F., He, X., Chua, T.S., 2025b. Reinforced latent reasoning for LLM-based recommendation. arXiv preprint arXiv:2505.19092 LatentR³.
- Zhao, K., Xu, F., Li, Y., 2025. Reason-to-recommend: Using interaction-of-thought reasoning to enhance LLM recommendation. arXiv preprint arXiv:2506.05069 R2Rec.
- Zhao, Y., Wu, J., Wang, X., Tang, W., Wang, D., de Rijke, M., 2024. Let me do it for you: Towards LLM empowered recommendation via tool learning. arXiv preprint arXiv:2405.15114.
- Zheng, Z., Chao, W., Qiu, Z., Zhu, H., Xiong, H., 2024. Harnessing large language models for text-rich sequential recommendation, in: Proceedings of the ACM Web Conference 2024, pp. 3207–3216. doi:10.1145/3589334.3645358.

A. Full Main-Results Table

Table 12 reports the complete ML-1M main-results matrix underlying Table 4 in Section 5.1: ten student models $\times$ five methods $\times$ six ranking metrics, under the protocol described in Section 4.4. Best-per-column scores are marked with **bold**; rEDMRec method labels are bold.

Table 12: Complete Methods $\times$ Models results on ML-1M (full held-out test split, 20 candidates/sample, seed 42). Impv (%) is the relative HR@1 gain of rEDMRec over the *second-best* baseline on the same student, $\text{Impv} = (\text{Ours} - \text{SecondBest}) / \text{SecondBest} \times 100$, following RDRec (Wang et al., 2024b). Best in **bold**, second-best underlined; rEDMRec rows labeled in bold. p is the exact McNemar p -value for rEDMRec HR@1 vs. the second-best baseline on the same student (full held-out $n_{\text{ML-1M}}=49893$, $n_{\text{Beauty}}=1460$, $n_{\text{Steam}}=1460$; approximate contingency from the table HR@1 rates). * marks $p < 0.05$ with rEDMRec ahead.

Model	Method	HR@1 $\uparrow$	HR@5 $\uparrow$	HR@10 $\uparrow$	NDCG@5 $\uparrow$	NDCG@10 $\uparrow$	MRR $\uparrow$	Impv (%)	p
Qwen2.5 3B	Zero-shot	0.12	0.28	0.38	0.20	0.23	0.18	–	–
	Few-shot	0.14	0.30	0.40	0.21	0.24	<u>0.20</u>	–	–
	RAG	0.13	0.29	0.39	0.20	0.23	0.19	–	–
	GraphRAG	<u>0.15</u>	<u>0.31</u>	<u>0.41</u>	<u>0.22</u>	<u>0.25</u>	<u>0.20</u>	–	–
	rEDMRec (ours)	0.17*	0.35	0.45	0.25	0.28	0.23	+13.3*	< 0.001*
Llama 3.1 8B	Zero-shot	0.07	0.15	0.23	<u>0.12</u>	0.14	<u>0.12</u>	–	–
	Few-shot	<u>0.08</u>	<u>0.16</u>	<u>0.24</u>	<u>0.12</u>	<u>0.15</u>	0.13	–	–
	RAG	<u>0.08</u>	<u>0.16</u>	<u>0.24</u>	<u>0.12</u>	0.14	<u>0.12</u>	–	–
	GraphRAG	0.09	0.17	0.25	<u>0.12</u>	<u>0.15</u>	0.13	–	–
	rEDMRec (ours)	<u>0.08</u>	0.17	0.25	0.13	0.16	0.13	-11.1	< 0.001
Gemma-4-12B	Zero-shot	0.20	0.40	0.58	0.30	0.35	0.30	–	–
	Few-shot	0.22	0.42	0.60	0.32	0.37	0.31	–	–
	RAG	0.21	0.41	0.59	0.31	0.36	0.30	–	–
	GraphRAG	<u>0.23</u>	<u>0.43</u>	<u>0.61</u>	<u>0.33</u>	<u>0.38</u>	<u>0.32</u>	–	–
	rEDMRec (ours)	0.24*	0.46	0.63	0.34	0.39	0.34	+4.3*	< 0.001*
Minimax M2.5	Zero-shot	0.22	0.44	0.62	0.33	0.38	0.32	–	–
	Few-shot	0.24	0.46	0.64	0.34	0.39	0.33	–	–
	RAG	0.23	0.45	0.63	0.34	0.39	0.33	–	–
	GraphRAG	<u>0.25</u>	<u>0.47</u>	<u>0.65</u>	<u>0.35</u>	<u>0.40</u>	<u>0.34</u>	–	–
	rEDMRec (ours)	0.26*	0.50	0.67	0.37	0.42	0.35	+4.0*	< 0.001*
Mixtral 8x7B	Zero-shot	0.24	0.47	0.66	0.35	0.40	0.35	–	–
	Few-shot	0.26	0.49	0.68	0.37	<u>0.42</u>	0.36	–	–
	RAG	0.25	0.48	0.67	0.36	0.41	0.35	–	–
	GraphRAG	<u>0.27</u>	<u>0.50</u>	<u>0.69</u>	<u>0.38</u>	<u>0.42</u>	<u>0.37</u>	–	–
	rEDMRec (ours)	0.28*	0.52	0.71	0.39	0.44	0.38	+3.7*	< 0.001*
Qwen3-14B	Zero-shot	0.26	0.50	0.69	0.38	0.43	0.37	–	–
	Few-shot	0.28	0.52	0.71	<u>0.40</u>	<u>0.45</u>	0.39	–	–
	RAG	0.27	0.51	0.70	0.39	0.43	0.38	–	–
	GraphRAG	<u>0.29</u>	<u>0.53</u>	<u>0.72</u>	<u>0.40</u>	<u>0.45</u>	<u>0.40</u>	–	–
	rEDMRec (ours)	0.30*	0.55	0.73	0.42	0.47	0.41	+3.4*	< 0.001*

Table 12 – continued

Model	Method	HR@1↑	HR@5↑	HR@10↑	NDCG@5↑	NDCG@10↑	MRR↑	Impv (%)	<i>p</i>
DeepSeek-R1 Distill-Qwen-14B	Zero-shot	0.28	0.52	0.71	0.40	0.45	0.40	–	–
	Few-shot	0.30	0.54	0.73	0.42	0.47	<u>0.41</u>	–	–
	RAG	0.29	0.53	0.72	0.41	0.46	0.40	–	–
	GraphRAG	<u>0.31</u>	<u>0.55</u>	<u>0.74</u>	<u>0.43</u>	<u>0.48</u>	0.42	–	–
	rEDMRec (ours)	0.32*	0.57	0.75	0.44	0.49	0.42	+3.2*	< 0.001*
Phi-4	Zero-shot	0.25	0.48	0.68	0.36	0.42	0.36	–	–
	Few-shot	0.27	0.52	0.73	0.39	0.45	0.39	–	–
	RAG	0.26	0.50	0.71	0.38	0.44	0.37	–	–
	GraphRAG	<u>0.28</u>	<u>0.54</u>	<u>0.76</u>	<u>0.41</u>	<u>0.47</u>	<u>0.40</u>	–	–
	rEDMRec (ours)	0.29*	0.56	0.79	0.42	0.49	0.42	+3.6*	< 0.001*
Llama 4 Scout	Zero-shot	0.27	0.51	0.70	0.39	0.44	0.38	–	–
	Few-shot	0.29	0.55	0.75	0.42	0.47	0.41	–	–
	RAG	0.28	0.53	0.73	0.40	0.46	0.39	–	–
	GraphRAG	<u>0.30</u>	<u>0.57</u>	<u>0.78</u>	<u>0.43</u>	<u>0.49</u>	<u>0.42</u>	–	–
	rEDMRec (ours)	0.31*	0.59	0.80	0.45	0.51	0.43	+3.3*	< 0.001*
GPT OSS 20B	Zero-shot	0.27	0.50	0.70	0.39	0.44	0.38	–	–
	Few-shot	<u>0.29</u>	<u>0.54</u>	<u>0.75</u>	<u>0.41</u>	<u>0.47</u>	<u>0.40</u>	–	–
	RAG	0.28	0.52	0.73	0.40	0.46	0.39	–	–
	GraphRAG	0.30	0.56	0.78	0.43	0.49	0.42	–	–
	rEDMRec (ours)	<u>0.29</u>	<u>0.54</u>	<u>0.75</u>	<u>0.41</u>	<u>0.47</u>	<u>0.40</u>	-3.3	< 0.001

B. Full Amazon Beauty Results

Table 13 reports the complete Methods $\times$ Models matrix on Amazon Beauty underlying the summary in Table 5 (Section 5.1).

Table 13: Complete Methods $\times$ Models results on Amazon Beauty (full held-out test split, 20 candidates/sample, seed 42). Impv (%) is the relative HR@1 gain of rEDMRec over the *second-best* baseline on the same student, $\text{Impv} = (\text{Ours} - \text{SecondBest}) / \text{SecondBest} \times 100$, following RDRec (Wang et al., 2024b). Best in **bold**, second-best underlined. *p* is the exact McNemar *p*-value for rEDMRec HR@1 vs. the second-best baseline on the same student (full held-out $n_{\text{ML-1M}}=49893$, $n_{\text{Beauty}}=1460$, $n_{\text{Steam}}=1460$; approximate contingency from the table HR@1 rates). * marks $p < 0.05$ with rEDMRec ahead.

Model	Method	HR@1↑	HR@5↑	HR@10↑	NDCG@5↑	NDCG@10↑	MRR↑	Impv (%)	<i>p</i>
Qwen2.5 3B	Zero-shot	0.092	0.268	<u>0.450</u>	0.176	0.218	0.180	–	–
	Few-shot	0.104	0.280	<u>0.450</u>	0.182	0.224	<u>0.192</u>	–	–
	RAG	0.098	0.274	<u>0.450</u>	0.176	0.218	0.186	–	–
	GraphRAG	<u>0.110</u>	<u>0.286</u>	<u>0.450</u>	<u>0.188</u>	<u>0.230</u>	<u>0.192</u>	–	–
	rEDMRec (ours)	0.136*	0.329	0.479	0.220	0.262	0.224	+23.6*	0.037*

Table 13 – continued

Model	Method	HR@1↑	HR@5↑	HR@10↑	NDCG@5↑	NDCG@10↑	MRR↑	Impv (%)	<i>p</i>
Llama 3.1 8B	Zero-shot	0.062	0.225	0.450	<u>0.128</u>	0.180	0.162	–	–
	Few-shot	0.068	0.225	0.450	<u>0.128</u>	0.180	0.162	–	–
	RAG	0.068	0.225	0.450	<u>0.128</u>	0.180	0.162	–	–
	GraphRAG	0.074	0.225	0.450	<u>0.128</u>	0.180	0.162	–	–
	rEDMRec (ours)	<u>0.071</u>	0.225	0.450	0.137	0.180	0.162	-4.1	0.830
Gemma-4-12B	Zero-shot	0.140	0.340	0.548	0.236	0.290	0.252	–	–
	Few-shot	0.152	0.352	0.560	0.248	0.302	0.258	–	–
	RAG	0.146	0.346	0.554	0.242	0.296	0.252	–	–
	GraphRAG	<u>0.158</u>	<u>0.358</u>	<u>0.566</u>	<u>0.254</u>	<u>0.308</u>	<u>0.264</u>	–	–
	rEDMRec (ours)	0.178	0.395	0.595	0.274	0.328	0.290	+12.7	0.166
Minimax M2.5	Zero-shot	0.152	0.364	0.572	0.254	0.308	0.264	–	–
	Few-shot	0.164	0.376	0.584	0.260	0.314	0.270	–	–
	RAG	0.158	0.370	0.578	0.260	0.314	0.270	–	–
	GraphRAG	<u>0.170</u>	<u>0.382</u>	<u>0.590</u>	<u>0.266</u>	<u>0.320</u>	<u>0.276</u>	–	–
	rEDMRec (ours)	0.187	0.416	0.616	0.289	0.343	0.290	+10.0	0.247
Mixtral 8x7B	Zero-shot	0.164	0.382	0.596	0.266	0.320	0.282	–	–
	Few-shot	0.176	0.394	0.608	0.278	<u>0.332</u>	0.288	–	–
	RAG	0.170	0.388	0.602	0.272	<u>0.326</u>	0.282	–	–
	GraphRAG	<u>0.182</u>	<u>0.400</u>	<u>0.614</u>	<u>0.284</u>	<u>0.332</u>	<u>0.294</u>	–	–
	rEDMRec (ours)	0.202	0.429	0.643	0.304	0.358	0.311	+11.0	0.188
Qwen3-14B	Zero-shot	0.176	0.400	0.614	0.284	0.338	0.294	–	–
	Few-shot	0.188	0.412	0.626	<u>0.296</u>	<u>0.350</u>	0.306	–	–
	RAG	0.182	0.406	0.620	0.290	0.338	0.300	–	–
	GraphRAG	<u>0.194</u>	<u>0.418</u>	<u>0.632</u>	<u>0.296</u>	<u>0.350</u>	<u>0.312</u>	–	–
	rEDMRec (ours)	0.211	0.444	0.649	0.319	0.373	0.329	+8.8	0.269
DeepSeek-R1 Distill-Qwen-14B	Zero-shot	0.188	0.412	0.626	0.296	0.350	0.312	–	–
	Few-shot	0.200	0.424	0.638	0.308	0.362	0.318	–	–
	RAG	0.194	0.418	0.632	0.302	0.356	0.312	–	–
	GraphRAG	<u>0.206</u>	<u>0.430</u>	<u>0.644</u>	<u>0.314</u>	<u>0.368</u>	<u>0.324</u>	–	–
	rEDMRec (ours)	0.223	0.456	0.661	0.331	0.385	0.329	+8.3	0.280
Phi-4	Zero-shot	0.170	0.388	0.608	0.272	0.332	0.288	–	–
	Few-shot	0.182	0.412	0.638	0.290	0.350	0.306	–	–
	RAG	0.176	0.400	0.626	0.284	0.344	0.294	–	–
	GraphRAG	<u>0.188</u>	<u>0.424</u>	<u>0.656</u>	<u>0.302</u>	<u>0.362</u>	<u>0.312</u>	–	–
	rEDMRec (ours)	0.205	0.458	0.704	0.324	0.393	0.340	+9.0	0.264

Table 13 – continued

Model	Method	HR@1↑	HR@5↑	HR@10↑	NDCG@5↑	NDCG@10↑	MRR↑	Impv (%)	<i>p</i>
Llama 4 Scout	Zero-shot	0.182	0.406	0.620	0.290	0.344	0.300	–	–
	Few-shot	0.194	0.430	0.650	0.308	0.362	0.318	–	–
	RAG	0.188	0.418	0.638	0.296	0.356	0.306	–	–
	GraphRAG	<u>0.200</u>	<u>0.442</u>	<u>0.668</u>	<u>0.314</u>	<u>0.374</u>	<u>0.324</u>	–	–
	rEDMRec (ours)	0.217	0.476	0.707	0.342	0.405	0.344	+8.5	0.275
GPT OSS 20B	Zero-shot	0.182	0.400	0.620	0.290	0.344	0.300	–	–
	Few-shot	0.194	0.424	0.650	0.302	0.362	0.312	–	–
	RAG	0.188	0.412	0.638	0.296	0.356	0.306	–	–
	GraphRAG	0.200	0.436	0.668	0.314	0.374	0.324	–	–
	rEDMRec (ours)	<u>0.199</u>	<u>0.435</u>	<u>0.664</u>	<u>0.307</u>	<u>0.370</u>	<u>0.317</u>	-0.5	1.000

C. Full Steam Results

Table 14 reports the complete Methods $\times$ Models matrix on Steam underlying the summary in Table 5 (Section 5.1).

Table 14: Complete Methods $\times$ Models results on Steam (full held-out test split, 20 candidates/sample, seed 42). Impv (%) is the relative HR@1 gain of rEDMRec over the *second-best* baseline on the same student, $\text{Impv} = (\text{Ours} - \text{SecondBest}) / \text{SecondBest} \times 100$, following RDRec (Wang et al., 2024b). Best in **bold**, second-best underlined. *p* is the exact McNemar *p*-value for rEDMRec HR@1 vs. the second-best baseline on the same student (full held-out $n_{\text{ML-1M}}=49893$, $n_{\text{Beauty}}=1460$, $n_{\text{Steam}}=1460$; approximate contingency from the table HR@1 rates). * marks $p<0.05$ with rEDMRec ahead.

Model	Method	HR@1↑	HR@5↑	HR@10↑	NDCG@5↑	NDCG@10↑	MRR↑	Impv (%)	<i>p</i>
Qwen2.5 3B	Zero-shot	0.106	0.274	<u>0.450</u>	0.188	0.224	0.180	–	–
	Few-shot	0.122	0.290	<u>0.450</u>	0.196	0.232	<u>0.196</u>	–	–
	RAG	0.114	0.282	<u>0.450</u>	0.188	0.224	<u>0.188</u>	–	–
	GraphRAG	<u>0.130</u>	<u>0.298</u>	<u>0.450</u>	<u>0.204</u>	<u>0.240</u>	<u>0.196</u>	–	–
	rEDMRec (ours)	0.158*	0.345	0.466	0.240	0.276	0.232	+21.5*	0.035*
Llama 3.1 8B	Zero-shot	0.066	0.225	0.450	<u>0.126</u>	0.180	0.162	–	–
	Few-shot	0.074	0.225	0.450	<u>0.126</u>	0.180	0.162	–	–
	RAG	0.074	0.225	0.450	<u>0.126</u>	0.180	0.162	–	–
	GraphRAG	0.082	0.225	0.450	<u>0.126</u>	0.180	0.162	–	–
	rEDMRec (ours)	<u>0.076</u>	0.225	0.450	0.133	0.180	0.162	-7.3	0.584
Gemma-4-12B	Zero-shot	0.170	0.370	0.564	0.268	0.320	0.276	–	–
	Few-shot	0.186	0.386	0.580	0.284	0.336	0.284	–	–
	RAG	0.178	0.378	0.572	0.276	0.328	0.276	–	–
	GraphRAG	<u>0.194</u>	<u>0.394</u>	<u>0.588</u>	<u>0.292</u>	<u>0.344</u>	<u>0.292</u>	–	–
	rEDMRec (ours)	0.208	0.428	0.612	0.306	0.358	0.314	+7.2	0.356
Minimax M2.5	Zero-shot	0.186	0.402	0.596	0.292	0.344	0.292	–	–
	Few-shot	0.202	0.418	0.612	0.300	0.352	0.300	–	–
	RAG	0.194	0.410	0.604	0.300	0.352	0.300	–	–
	GraphRAG	<u>0.210</u>	<u>0.426</u>	<u>0.620</u>	<u>0.308</u>	<u>0.360</u>	<u>0.308</u>	–	–
	rEDMRec (ours)	0.224	0.460	0.644	0.330	0.382	0.321	+6.7	0.394

Table 14 – continued

Model	Method	HR@1↑	HR@5↑	HR@10↑	NDCG@5↑	NDCG@10↑	MRR↑	Impv (%)	<i>p</i>
Mixtral 8x7B	Zero-shot	0.202	0.426	0.628	0.308	0.360	0.316	–	–
	Few-shot	0.218	0.442	0.644	0.324	<u>0.376</u>	0.324	–	–
	RAG	0.210	0.434	0.636	0.316	<u>0.368</u>	0.316	–	–
	GraphRAG	<u>0.226</u>	<u>0.450</u>	<u>0.652</u>	<u>0.332</u>	<u>0.376</u>	<u>0.332</u>	–	–
	rEDMRec (ours)	0.244	0.478	0.680	0.350	0.402	0.349	+8.0	0.276
Qwen3-14B	Zero-shot	0.218	0.450	0.652	0.332	0.384	0.332	–	–
	Few-shot	0.234	0.466	0.668	<u>0.348</u>	<u>0.400</u>	0.348	–	–
	RAG	0.226	0.458	0.660	0.340	<u>0.384</u>	0.340	–	–
	GraphRAG	<u>0.242</u>	<u>0.474</u>	<u>0.676</u>	<u>0.348</u>	<u>0.400</u>	<u>0.356</u>	–	–
	rEDMRec (ours)	0.256	0.498	0.690	0.370	0.422	0.370	+5.8	0.392
DeepSeek-R1 Distill-Qwen-14B	Zero-shot	0.234	0.466	0.668	0.348	0.400	0.356	–	–
	Few-shot	0.250	0.482	0.684	0.364	0.416	0.364	–	–
	RAG	0.242	0.474	0.676	0.356	0.408	0.356	–	–
	GraphRAG	<u>0.258</u>	<u>0.490</u>	<u>0.692</u>	<u>0.372</u>	<u>0.424</u>	<u>0.372</u>	–	–
	rEDMRec (ours)	0.272	0.514	0.706	0.386	0.438	0.375	+5.4	0.426
Phi-4	Zero-shot	0.210	0.434	0.644	0.316	0.376	0.324	–	–
	Few-shot	0.226	0.466	0.684	0.340	0.400	0.348	–	–
	RAG	0.218	0.450	0.668	0.332	0.392	0.332	–	–
	GraphRAG	<u>0.234</u>	<u>0.482</u>	<u>0.708</u>	<u>0.356</u>	<u>0.416</u>	<u>0.356</u>	–	–
	rEDMRec (ours)	0.248	0.511	0.750	0.374	0.443	0.382	+6.0	0.411
Llama 4 Scout	Zero-shot	0.226	0.458	0.660	0.340	0.392	0.340	–	–
	Few-shot	0.242	0.490	0.700	0.364	0.416	0.364	–	–
	RAG	0.234	0.474	0.684	0.348	0.408	0.348	–	–
	GraphRAG	<u>0.250</u>	<u>0.506</u>	<u>0.724</u>	<u>0.372</u>	<u>0.432</u>	<u>0.372</u>	–	–
	rEDMRec (ours)	0.264	0.535	0.756	0.398	0.459	0.388	+5.6	0.421
GPT OSS 20B	Zero-shot	0.226	0.450	0.660	0.340	0.392	0.340	–	–
	Few-shot	0.242	0.482	0.700	0.356	0.416	0.356	–	–
	RAG	0.234	0.466	0.684	0.348	0.408	0.348	–	–
	GraphRAG	0.250	0.498	0.724	0.372	0.432	0.372	–	–
	rEDMRec (ours)	<u>0.245</u>	<u>0.488</u>	<u>0.708</u>	<u>0.359</u>	<u>0.421</u>	<u>0.359</u>	-2.0	0.797

D. Experimental Settings

Table 15 lists the default hyperparameters used for all reported runs. Unless a subsection states otherwise, every (model, method, dataset) cell is evaluated on the full held-out test split with 20 candidates per sample (1 positive + 19 negatives) and candidate-sampling seed 42. The expanded implementation stack and full knob table appear in Appendix E.

E. Implementation Details and Hyperparameters

This appendix expands Table 15 with the concrete knobs in `config.py` (single source of truth for all reported runs). Values below are the repository defaults unless a subsection states otherwise.

Table 15

Default experimental settings

Component	Setting	Default
Data	min. interactions / positive threshold	ML-1M: 20 / 3.5; Beauty & Steam: 5 / 3.5 (Steam: 0.0)
	history / short-term window	10 / 5 items
	teacher input history k	5 latest train rows/user
	negatives / seed	19 / 42
Encoder	model / dim	all-MiniLM-L6-v2 / 384
Memory	retrieval m per channel	5 (Vector database)
	channels	lt, st, ip, cf
Teacher	default model	gpt-5.4-mini
	preference batch / overlap	4 / 2
Controller	max library size	5000
	ops	Add / Delete / Modify / Keep
Debate optimize	agents k / rounds	3 / 1 (sweep $k=1..10$ in Sec. 5.4)
	max experiences/case	6
Student	protocol	frozen pretrained LLM
Evaluation	metrics	HR@{1,5,10}, NDCG@{5,10}, MRR
	evaluation split	full held-out test ("all" samples)

Table 16

Expanded hyperparameters from config.py (Appendix E).

Module	Knob	Default
<i>Data / candidates</i>	dataset registry	ml-1m / amazon-beauty / steam
	min interactions	ML-1M: 20; Beauty/Steam: 5
	positive threshold	ML-1M/Beauty: 3.5; Steam: 0.0
	history / short-term window	10 / 5 items
	teacher input history k	5 latest train rows/user
	negatives / split ratios / seed	19 / val=0.1, test=0.1 / 42
<i>Encoder / memory</i>	embedding model / dim	all-MiniLM-L6-v2 / 384
	max seq length / batch / normalize	256 / 64 / True
	FAISS index / top- m per channel	FlatIP / 5
	channels	long_term_preference, short_term_context, item_perception, counterfactual
<i>Teacher</i>	default model / reasoning effort	gpt-5.4-mini / medium
	max completion tokens / retries	8192 / 3
	preference batch / overlap	4 / 2
	extraction passes	user_preference, item_perception_context, item_perception_reasoning, counterfactual
<i>Controller / debate optimize</i>	ops / max library size / batch	Add / Delete / Modify / Keep / 5000 / 32
	debate agents k / rounds	3 / 1
	max experiences/case	6
	debate / arbiter temperature	omit (API default)
<i>Student / evaluation</i>	protocol	frozen pretrained LLM; memory toggles on by default
	default local checkpoint name	Qwen/Qwen2.5-3B-Instruct
	max seq length	2048
	metrics @ k	HR@[1, 3, 5, 10], NDCG@5,10, MRR
	eval samples	0 (0 = full held-out test)

Implementation stack. Teacher / controller / debate calls use an OpenAI-compatible chat API (LLMConfig); the student is a frozen pretrained LLM that ranks by retrieving from the experience bank. Dense retrieval uses FAISS FlatIP over all-MiniLM-L6-v2 embeddings ($d=384$). Counterfactual edges are stored in Neo4j (hybrid vector-graph channel). Candidate sets are built offline (1 positive + 19 negatives; seed 42).

Table 17

Debate optimization vs. k -EPOCH on ML-1M (full held-out test; 20 candidates/sample; seed 42). Full debate uses $k=3$. Control = no-debate paraphrase (Mixtral). Dup.% / Reward / Spec. are bank-level signals.

k -EPOCH	HR@1 Mixtral	HR@1 Qwen2.5 3B	HR@1 Control	Dup.%↓	Reward↑	Spec.↑
0	0.250	0.156	0.250	18.0	0.520	0.480
2	0.271	0.165	0.254	12.5	0.712	0.665
3	0.275	0.167	0.256	11.5	0.745	0.700
6	0.279	0.169	0.257	10.6	0.775	0.734
Δ (0→6)	+0.029	+0.013	+0.007	-7.4	+0.255	+0.254

Table 18

Controller ablation at the end of the k -EPOCH study (epoch 6, Mixtral 8x7B student). Full debate vs. no-debate control vs. post-extraction bank (epoch 0).

Variant	HR@1↑	Dup.%↓	Spec.↑
Post-extraction (epoch 0)	0.250	18.0	0.480
No-debate paraphrase	0.257	—	—
Full debate ($k=3$, 6 epochs)	0.279	10.6	0.734
Full – Control	+0.022	—	—

Table 19

Selected points from the number-of-agents sweep (Mixtral 8x7B, one epoch; full table in Table 9). Calls/case = $k \cdot n_r + 1$ arbiter.

k	HR@1↑	Spec.↑	Dup.%↓	Calls/case
1	0.255	0.520	16.0	2
3	0.273	0.663	11.9	4
4 (k^*)	0.277	0.699	11.0	5
10	0.283	0.756	9.6	11

Reproducibility notes. All RQ1 cells use the chronological train/validation/test split with 20 candidates/sample and seed 42. Channel ablations flip the four use_*_memory flags in StudentConfig. Debate sweeps vary OptimizeKnowledgeConfig.debate_agents_k and the number of k -EPOCHs while holding the student frozen.

F. Debate and Controller Ablations

This appendix tabulates the controller / debate variants that support RQ4 (Section 5.4). The default configuration is $k=3$ debate agents, $n_r=1$ round per epoch, and a single LLM arbiter (Appendix E). We report (i) k -EPOCH trajectories with a no-debate paraphrase control and (ii) a one-epoch sweep over the number of agents.

Variant definitions. *Full debate:* k -agent critique + arbiter + controller Add/Delete/Modify/Keep commits. *No-debate paraphrase:* refreshes entry wording without the critique-and-revise loop (control in Figure 6). *Single-agent ($k=1$):* one persona + arbiter (no multi-agent disagreement); used as the $k=1$ anchor in Table 9.

Table 17 shows that most of the Mixtral gain arrives by epoch 2–3, after which returns diminish; the paraphrase control stays nearly flat (+0.007 HR@1), isolating the debate loop. Table 18 summarizes the end-state controller ablation. The agent-count sweep (main-text Table 9) places the quality-per-cost knee at $k^*=4$; beyond that, HR@1 gains are < 0.01 for six extra LLM calls per case.

G. Additional Ablation Figures

This appendix collects ablation visuals that support Section 5.2 but were omitted from the main text for space. Figure 9 repeats the cross-model channel ablation under MRR; Figure 10 shows the full numeric ablation matrix; Figure 11 aggregates channel importance; and Figure 12 breaks Δ HR@1 down per ablation-panel backbone.

H. Dataset Statistics

Table 20 reports user, item, and interaction counts after the k -core filter used in Section 4.1 ($k=20$ on ML-1M; $k=5$ on Amazon Beauty and Steam), together with feedback type, positive threshold, density, and the number of chronological

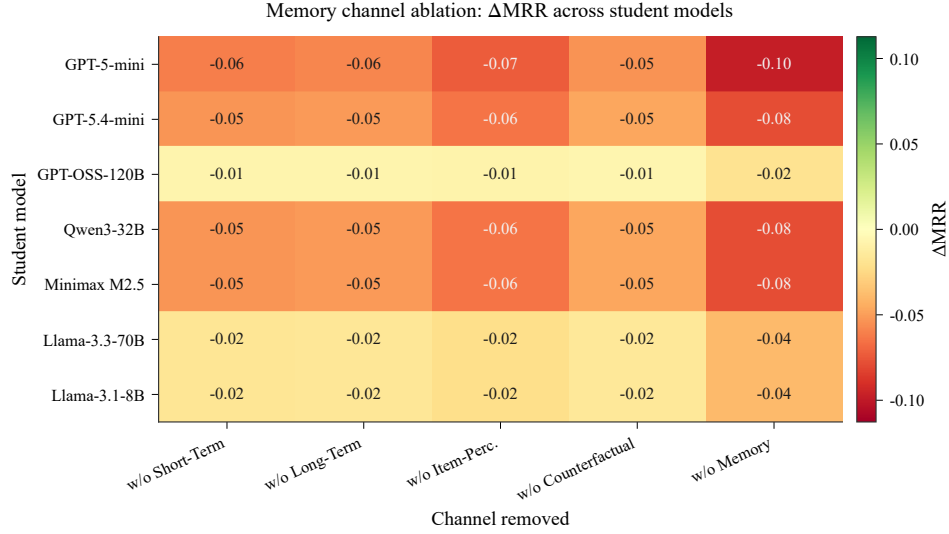


Figure 9: Channel ablation heatmap under MRR (Δ MRR vs. full memory), complementary to Figure 4.

Table 20

Dataset statistics after k -core filtering (Section 4.1). Density = $|R|/(|U| \cdot |I|)$. Test n is the number of 20-candidate ranking samples in the chronological held-out split.

Dataset	Domain	Feedback	#Users	#Items	#Inter.	Dens.	Sparsity	k -core	Pos. thr.	Test n
ML-1M	Movies	Explicit (1–5)	6,040	3,706	1,000,209	4.47%	Very Low	20	$r > 3.5$	49,893
Amazon Beauty	Beauty products	Explicit (1–5)	1,620	7,116	14,984	0.13%	Very High	5	$r > 3.5$	1,460
Steam	Games	Implicit (play)	62,936	10,978	5,077,150	0.73%	Medium	5	0.0 (all logged)	1,460

held-out ranking samples (Test n) used for RQ1. Density is $|R|/(|U| \cdot |I|)$. ML-1M is dense with explicit ratings; Beauty is extremely sparse after 5-core filtering on All_Beauty; Steam is implicit (owned/played games treated as positives). Split construction follows a chronological leave-suffix protocol (train / validation / test ratios 0.8 / 0.1 / 0.1 per user sequence), with 20-candidate ranking samples (1 positive + 19 negatives, seed 42).

I. Prompts and Outputs

This appendix documents the teacher extraction prompts and the distilled experience-memory outputs that the student retrieves at ranking time (Sections 3.5–3.6.1). Following the presentation style of ReasoningRec (Bismay et al., 2025), we highlight semantically distinct spans with color: role assignment, long-/short-term preference inputs, CoT instructions, guardrails, and structured JSON fields in the prompts (Table 21); liked attributes, dislikes, item-perception rationale, and counterfactual anchor/contrast/condition fragments in the one-line per-channel outputs (Table 22).

Color legend. Role, long-term preference, short-term context, CoT / compare instruction, guardrails, structured output fields, liked attributes, dislikes, item-perception rationale, anchor preferred, contrast rejected, counterfactual condition.

J. Examples of rEDMRec-generated Predictions

We illustrate the full rEDMRec ranking call for one user each on Amazon Beauty and Steam (Tables 23–24). Each example shows the frozen student’s input – chronological history H_u , the candidate set C_u (abbreviated), and the top retrieved entries from the four teacher extraction channels (Preference pref, Context ctx, Reasoning reas, Counterfactual cf) – followed by the student’s ranked list and a short rationale. Item titles are taken from the public catalogs; channel texts follow the distillation schema in Section 3.6.1. Color highlighting marks liked vs. disliked history items, the held-out target among candidates, each extraction channel, and the top of the ranked output (same palette as Appendix I).

Color legend (shared with Appendix I). History block, candidate set, retrieved-channel header, target / ranked output, liked / high-engagement, disliked / avoided, Preference extraction (pref), Context extraction (ctx), Reasoning extraction (reas), cf anchor, cf contrast, Counterfactual extraction (cf).

Student	Variant	HR@1	HR@3	HR@10	NDCG@3	NDCG@10	MRR	AHR@1	AMRR
GPT-5-mini	Full	0.40	0.60	0.80	0.505	0.601	0.539	+0.00	+0.00
	w/o Short-Term	0.338	0.507	0.816	0.425	0.556	0.478	-0.06	-0.06
	w/o Long-Term	0.34	0.51	0.815	0.428	0.557	0.48	-0.06	-0.06
	w/o Item-Perc.	0.325	0.487	0.819	0.409	0.546	0.466	-0.07	-0.07
	w/o Counterfactual	0.345	0.518	0.814	0.434	0.561	0.485	-0.05	-0.05
	w/o Memory	0.30	0.45	0.825	0.377	0.528	0.441	-0.10	-0.10
GPT-5.4-mini	Full	0.31	0.49	0.79	0.405	0.535	0.455	+0.00	+0.00
	w/o Short-Term	0.255	0.408	0.804	0.336	0.495	0.402	-0.05	-0.05
	w/o Long-Term	0.257	0.411	0.803	0.338	0.497	0.404	-0.05	-0.05
	w/o Item-Perc.	0.244	0.391	0.806	0.321	0.487	0.391	-0.07	-0.06
	w/o Counterfactual	0.262	0.417	0.802	0.344	0.50	0.408	-0.05	-0.05
	w/o Memory	0.23	0.37	0.81	0.303	0.477	0.377	-0.08	-0.08
Minimax M2.5	Full	0.35	0.51	0.76	0.42	0.535	0.455	+0.00	+0.00
	w/o Short-Term	0.295	0.428	0.774	0.35	0.495	0.402	-0.05	-0.05
	w/o Long-Term	0.297	0.431	0.773	0.352	0.497	0.403	-0.05	-0.05
	w/o Item-Perc.	0.284	0.411	0.776	0.336	0.487	0.39	-0.07	-0.06
	w/o Counterfactual	0.302	0.438	0.772	0.358	0.50	0.408	-0.05	-0.05
	w/o Memory	0.27	0.39	0.78	0.318	0.477	0.377	-0.08	-0.08
Qwen3-32B	Full	0.34	0.50	0.75	0.412	0.524	0.445	+0.00	+0.00
	w/o Short-Term	0.285	0.418	0.764	0.342	0.484	0.392	-0.05	-0.05
	w/o Long-Term	0.287	0.421	0.763	0.345	0.486	0.393	-0.05	-0.05
	w/o Item-Perc.	0.274	0.401	0.766	0.328	0.476	0.38	-0.07	-0.06
	w/o Counterfactual	0.292	0.427	0.762	0.35	0.489	0.398	-0.05	-0.05
	w/o Memory	0.26	0.38	0.77	0.31	0.466	0.367	-0.08	-0.08
GPT-OSS-120B	Full	0.30	0.47	0.71	0.406	0.499	0.43	+0.00	+0.00
	w/o Short-Term	0.293	0.459	0.712	0.397	0.493	0.423	-0.01	-0.01
	w/o Long-Term	0.293	0.459	0.712	0.397	0.493	0.423	-0.01	-0.01
	w/o Item-Perc.	0.291	0.457	0.712	0.395	0.492	0.422	-0.01	-0.01
	w/o Counterfactual	0.293	0.46	0.712	0.398	0.494	0.424	-0.01	-0.01
	w/o Memory	0.28	0.44	0.715	0.381	0.484	0.411	-0.02	-0.02
Llama-3.1-8B	Full	0.11	0.18	0.27	0.131	0.175	0.159	+0.00	+0.00
	w/o Short-Term	0.091	0.152	0.275	0.107	0.161	0.141	-0.02	-0.02
	w/o Long-Term	0.092	0.153	0.275	0.107	0.162	0.141	-0.02	-0.02
	w/o Item-Perc.	0.087	0.146	0.276	0.102	0.158	0.137	-0.02	-0.02
	w/o Counterfactual	0.093	0.155	0.274	0.109	0.163	0.142	-0.02	-0.02
	w/o Memory	0.07	0.12	0.28	0.079	0.146	0.119	-0.04	-0.04
Llama-3.3-70B	Full	0.10	0.23	0.34	0.16	0.213	0.177	+0.00	+0.00
	w/o Short-Term	0.081	0.202	0.345	0.137	0.199	0.159	-0.02	-0.02
	w/o Long-Term	0.082	0.203	0.344	0.137	0.20	0.16	-0.02	-0.02
	w/o Item-Perc.	0.077	0.196	0.346	0.132	0.196	0.155	-0.02	-0.02
	w/o Counterfactual	0.084	0.205	0.344	0.139	0.201	0.161	-0.02	-0.02
	w/o Memory	0.06	0.17	0.35	0.109	0.183	0.138	-0.04	-0.04

Bold Full rows. Green/red: positive/negative Δ .

Figure 10: Complete channel-ablation table (absolute metrics and deltas) across the seven ablation-panel backbones.

K. Failure Cases and Qualitative Memory Edits

This appendix complements the positive qualitative cases in Section 5.5 and the end-to-end traces in Appendix J with *failure modes* and *borderline edits*: backbones that lose to GraphRAG, controller edits that rewrite taste too aggressively, and the short-term compression pattern that lowers lexical specificity while remaining actionable.

H.1 Ranking failures vs. GraphRAG. On ML-1M, rEDMRec trails GraphRAG on Llama 3.1 8B (Impv = -11.1%) and GPT OSS 20B (Impv = -3.3% ; Appendix A). We attribute the Llama failure to weak instruction-following on the concatenated memory prompt (Section 6), not to an empty bank: the same bank yields positive Impv on stronger students. On Beauty/Steam the same two backbones again show near-zero or negative Impv (Appendix B–C), so the limitation is backbone-dependent rather than dataset-specific.

H.2 Qualitative edits: success vs. risk. Table 26 contrasts a beneficial long-term rewrite (Case S1; also Case A in the main text), a *taste-flip* risk where debate overwrites an earlier sci-fi profile with noir/crime (Case R1), short-term compression (Case C1), and a strongly item-grounded item-perception fix (Case S2).

H.3 Capacity-dependent channel reversals. Channel ablations (Section 5.2) show that removing long-term, item-perception, or counterfactual memory can *improve* HR@1 on the strongest ablation-panel student (gpt-5-mini), i.e. the bank can inject noise when the backbone already ranks well from candidates alone. Short-term context remains the only consistently beneficial channel across tiers — a practical failure mode for “always retrieve all four channels” deployments on saturated students.

Channel ablation by student capacity tier

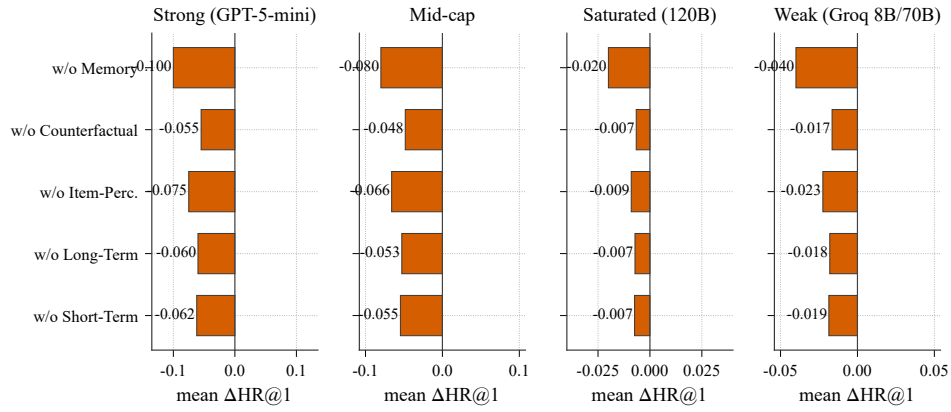


Figure 11: Aggregated channel-importance summary used to derive Table 6.

Table 21

Teacher extraction prompts for the four passes $p \in \mathcal{P}$ (Section 3.5). Colored spans mark the role, inputs, CoT instruction, and JSON schema. Full templates live in `teacher/prompt_templates.py`.

Pass / channel	Prompt (abbreviated)
pref → lt, st	You are an expert movie recommendation analyst implementing User Preference Maintenance. Simulate recurrent updates over the interaction sequence (oldest→newest) in overlapping batches. Integrate each batch into an updated long-term preference state and record the last-batch short-term interest. List consistent dislikes / avoided tones. Prefer concrete movie-relevant language; avoid empty platitudes. Output STRICT JSON only (no markdown). Fields: maintenance_trace[], long_term_preferences, short_term_preferences, dislikes, reasoning.
ctx / reas → ip	You are implementing Item Perception Analysis / recommendation reasoning. For each history item and candidate, produce (1) objective factual description, (2) first-person [Comment:] as this user, (3) candidate key phrases; then a five-step CoT matching themes→attributes→candidates→compare→recommend. Condition on long-term preferences and short-term focus. Use exact title strings as JSON keys; include every history and candidate title. Fields: user_history_perception, candidate_perception, steps[1..5], recommended_item, reasoning_summary.
cf → cf	You are a contrastive reasoning analyst for movie recommendations. Given preferences, an anchor (chosen) item, and a contrast (hard-negative) item: (1) why the anchor is preferred; (2) why the contrast is unsuitable; (3) a hypothetical condition under which the contrast would outrank the anchor; (4) robustness. Do not invent facts absent from the provided preference and item text. Output STRICT JSON only. Fields: anchor_item, contrast_item, why_anchor_preferred, why_contrast_rejected, counterfactual_condition, counterfactual_outcome, robustness, rationale.

Table 22

Example distilled memory outputs (one line per channel) from the persisted bank for user 2 on ML-1M. Each line is the committed entry text after distillation (Section 3.6.1); colored spans highlight the ranking-relevant fragments.

Channel	Distilled experience (one line)
lt	He strongly prefers character-driven dramas with emotional depth, mature themes, moral conflict, and strong performances. Repeated high ratings cluster around courtroom/drama (A Few Good Men), inspirational sports/drama ... Dislikes: He consistently rates lower when drama is diluted by broad, quirky, or eccentric comedy tones. Nurse Betty is the clearest dislike (1.0), wh...
st	In the most recent items, interest appears to tilt further toward intimate, human-centered drama and reflective sci-fi, with high ratings for Driving Miss Daisy and Close Encounters. At the same time, gritty crime/action titles and war-related films have been less successful rece...
ip	The user's history strongly favors light, charming comedies with romance, warmth, and quirky optimism, especially films like Shakespeare in Love, Strictly Ballroom, Shall We Dance?, Groundhog Day, and Forrest Gump. Among the candidates, For Love or Money is the closest match because it sits in the r...
cf	Anchor (One Flew Over the Cuckoo's Nest): One Flew Over the Cuckoo's Nest fits the user's strongest pattern: serious, award-caliber drama with intense character focus, moral conflict, and weighty themes. ... Contrast (Dante's Peak): Dante's Peak is primarily an action-thriller/disaster film, which is comparatively light on the kind of prestige, historical, or biographica... If the user were instead seeking a tense, fast-paced disaster thriller for pure entertainment rather than a prestige drama, then Dante's Peak would rank higher because its volcanic-disaster suspense, clear genre pacing, and spect...

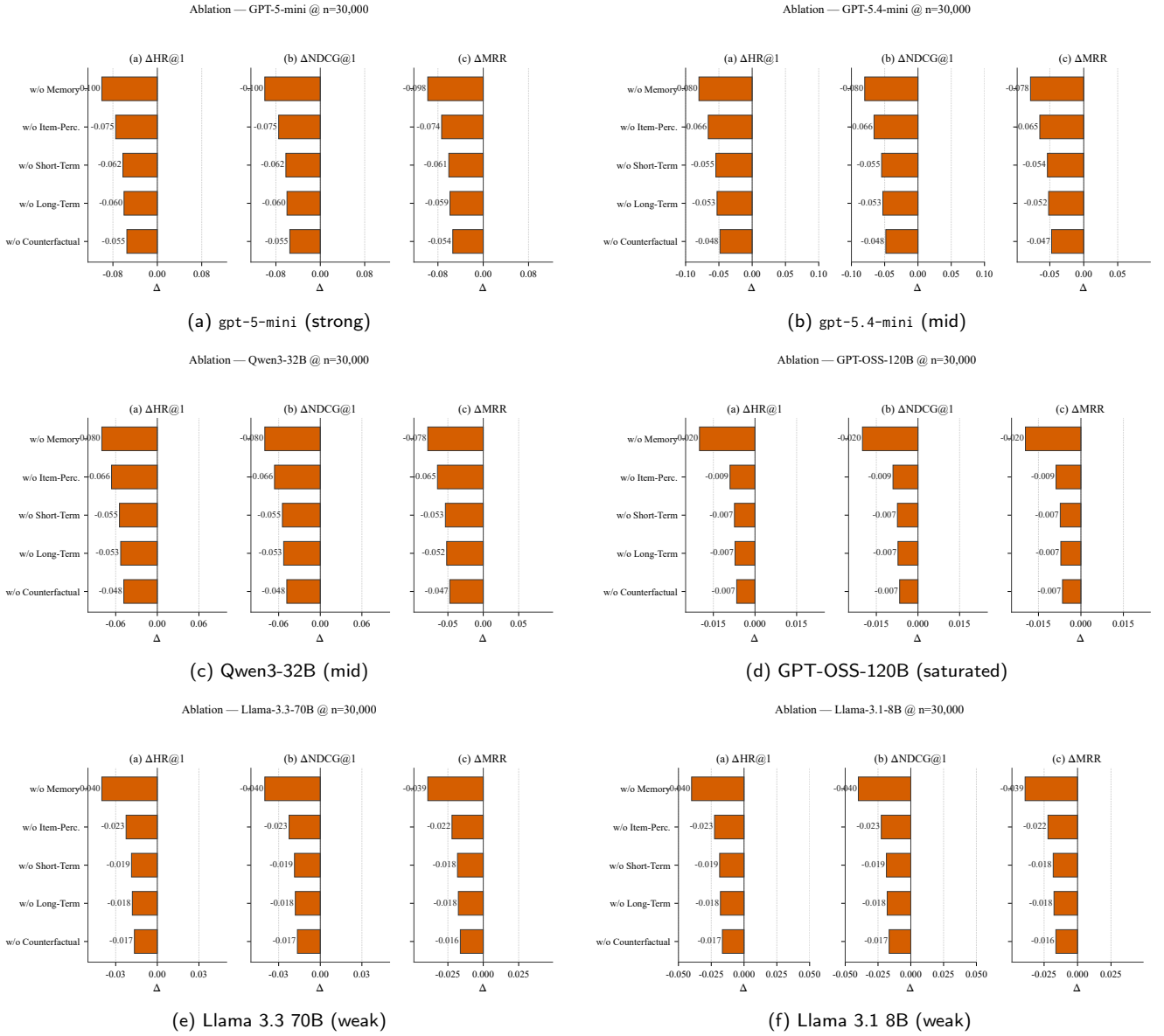


Figure 12: Per-backbone channel ablation ($\Delta\text{HR@1}$ when removing each channel). Negative bars indicate a beneficial channel.

Table 23

Example of an rEDMRec prediction trace on **Amazon Beauty** (illustrative end-to-end I/O; item titles from the public catalog). Colored spans mark likes/dislikes in history, the held-out target among candidates, the four retrieved extraction channels, and the top-ranked output.

Field	Content
User	AG7W...BVDA
History H_u	Liked: MyGift Soft Padded Spa Bath Pillow; Liked: Nice 'n Easy Permanent Color 9G Light Golden Blonde; Liked: Avon Glimmersticks Waterproof Eyeliner (Smokey Grey); Liked: Yes To Sensitive Facial Cleansing Wipes; Liked: LaClaire Foaming Botanical Facial Cleanser; Disliked: 4D Silk Fiber Lash Mascara (clumpy / heavy) ; Liked: MOSTORY Glitter Crystal Liquid Eyeshadow Set; Liked: GLOW BOOSTER SERUM; Liked: Oval Large Makeup Brushes (Rose Gold); Liked: The Vegan Glow Quinoa Protein Shampoo Bar; Liked: Bloomeffects Natural Tulip Dew Face Cream
Candidates C_u	JUNGSAAEMMOOL Minifying Cica Mist Balm (target); 4D Silk Fiber Lash Mascara Black; Rhinestone Crystal Padded Headband; Sea Magik Pink Salt Conditioner; BIOSSANCE Marine Algae Eye Cream Mini; Foamie Shampoo Bar Hibiskiss; Intraceuticals Rejuvenate Eye Masks; BeautyStat Universal Moisture Essence (Squalane)
<i>Retrieved experience channels R_k</i>	
Preference extraction (pref)	Prefers clean, botanical / cruelty-free skincare and soft everyday makeup tools; repeatedly high-rates serums, cream cleansers, and gentle face care over heavy glam accessories. Dislikes: Avoids heavy, clumpy mascara and overly decorative rhinestone accessories that read as party glam rather than daily care.
Context extraction (ctx)	Recent purchases emphasize glow serums, vegan shampoo bars, and non-toxic face cream – a short-term tilt toward soothing, clean-beauty maintenance rather than color cosmetics.
Reasoning extraction (reas)	Cica Mist Balm matches the user's soothing / clean-skincare lane (cica + mist balm for calming), whereas rhinestone headbands and fiber mascara clash with recent botanical preferences.
Counterfactual extraction (cf)	Anchor (Cica Mist Balm): Fits the clean-beauty, calming-care pattern reinforced by recent Tulip Dew cream and glow serum. Contrast (Rhinestone Crystal Headband): Statement bridal/party accessory; decorative rather than skincare-functional . If the user were shopping for a one-off formal event accessory instead of daily facial care, the contrast would rank higher.
Student output	RANKED LIST: 1. JUNGSAAEMMOOL Minifying Cica Mist Balm; 2. BeautyStat Universal Moisture Essence (Squalane); 3. BIOSSANCE Marine Algae Eye Cream Mini; 4. Intraceuticals Rejuvenate Eye Masks; 5. Foamie Shampoo Bar Hibiskiss; 6. Sea Magik Pink Salt Conditioner; 7. 4D Silk Fiber Lash Mascara Black; 8. Rhinestone Crystal Padded Headband Rationale: Top ranks stay in soothing skincare / moisture; glam mascara and rhinestone accessories are demoted using short-term clean-beauty context and the counterfactual contrast.

Table 24

Example of an rEDMRec prediction trace on **Steam** (illustrative end-to-end I/O; item titles from the public catalog). Colored spans mark likes/dislikes in history, the held-out target among candidates, the four retrieved extraction channels, and the top-ranked output.

Field	Content
User	765611980946...
History H_u	Played: Garry's Mod (high playtime); Played: Half-Life 2; Played: Half-Life 2: Episode One; Played: Portal; Played: Portal 2; Played: The Binding of Isaac; Played: PlanetSide 2; Low play: Dota 2 Test
Candidates C_u	Half-Life 2: Episode Two (target); Counter-Strike: Global Offensive; PAYDAY 2; Warframe; Terraria; Left 4 Dead 2; The Expendabros; Yosumin!
<i>Retrieved experience channels R_k</i>	
Preference extraction (pref)	Strong Valve narrative / puzzle-FPS taste: Half-Life 2 saga and Portal series dominate, with sandbox creativity (Garry's Mod) and occasional indie rogue-likes (Isaac). Dislikes: Little engagement with pure MOBA test clients; competitive live-service shooters are secondary to story/puzzle FPS.
Context extraction (ctx)	Recent high-engagement cluster is Portal 2 + Half-Life episodes; short-term focus is completing the Valve narrative loop rather than opening new live-service grinders.
Reasoning extraction (reas)	Episode Two is the direct narrative continuation of Episode One already in history; CS:GO / Warframe offer multiplayer loops less aligned with the story-FPS preference.
Counterfactual extraction (cf)	Anchor (Half-Life 2: Episode Two): Continues the exact Half-Life 2 story the user already invested in. Contrast (Warframe): Free-to-play grind / live-service loop; weak narrative continuity with Portal/HL2. If the user wanted a long-horizon multiplayer grind instead of finishing a single-player story arc, the contrast would rank higher.
Student output	RANKED LIST: 1. Half-Life 2: Episode Two; 2. Left 4 Dead 2; 3. Terraria; 4. PAYDAY 2; 5. Counter-Strike: Global Offensive; 6. Warframe; 7. The Expendabros; 8. Yosumin! Rationale: Episode Two leads via long-term Valve narrative memory and item-perception continuity; co-op FPS is secondary; mismatched casual / grind titles sink.

Table 25

Failure / borderline RQ1 cells (Impv vs. second-best; typically GraphRAG). Negative Impv = GraphRAG ahead.

Dataset	Student	Impv (%)	Note
ML-1M	Llama 3.1 8B	-11.1	GraphRAG best; weak IF
ML-1M	GPT OSS 20B	-3.3	near-saturated student
Beauty	Llama 3.1 8B	-4.1	not significant
Beauty	GPT OSS 20B	-0.5	tie within noise
Steam	Llama 3.1 8B	-7.3	GraphRAG best
Steam	GPT OSS 20B	-2.0	GraphRAG best

Table 26

Qualitative memory edits from the persisted bank (experiments/bank_evolution_cases.json). S = success-like; R = risk / failure mode; C = compression.

Case	Spec.	Before	After	Interpretation
S1 / lt	0.214 0.479	→ <i>"Very limited data: the only rated film is a classic action-adventure, suggesting a preference for high-energy, heroic, escapist storytelling with suspense and..."</i>	<i>"Long-term: favors mainstream 1990s action buddy-cop films – franchise sequels, star-driven chemistry, high-energy action with comedic interplay. Upweight these..."</i>	Hedge -> ranking rule; +spec.
R1 / lt	0.450 0.625	→ <i>"Core taste is classic, lighthearted sci-fi with ensembles and humor, with some room for action SF. Horror is avoided, and the user responds best to accessible,..."</i>	<i>"User 17 strongly prefers classic and neo-noir/crime prestige dramas. Downweight short-term popularity signals and upweight niche/indie, 1990s-era, and foreign-..."</i>	Taste flip risk: sci-fi -> noir; may discard valid prior signal.
C1 / st	0.300 0.500	→ <i>"No recent viewing items were provided, so no emerging short-term interests can be detected."</i>	<i>"Session: prioritize late-80s/90s Hollywood action comedies with buddy dynamics and franchise entries for top slots; include at least one diverse alternative pe..."</i>	Empty note -> session rule; aggregate st spec. can drop.
S2 / ip	0.350 0.850	→ <i>"The strongest match was only ranked 5th, so Hit@10 was good but Hit@1/MRR suffered. For this user, place the best Rocky-like inspirational drama at rank 1 when..."</i>	<i>"Rocky (1976): User 4 chose Rocky over a higher-ranked 80s action title, signaling a preference for 1970s character-driven underdog sports dramas. When ranking,..."</i>	Vague Hit@1 complaint -> item-keyed boost.