CCSM: Conflict-Calibrated Structured Memory Correction for Agentic Sequential Recommendation

Anonymous ACL submission

Abstract

Sequential recommendation is increasingly studied in agentic settings where large language models can reason over user history through memory, reflection, and tool use. However, more memory alone does not guarantee better recommendations: mixed interests, stale behaviors, and conflict-heavy negative evidence can distort preference inference, while hard, ambiguous cases can still damage the top-ranked recommendation. We study structured memory correction for agentic sequential recommendation and introduce CCSM, a method that combines conflict-calibrated structured memory with adaptive hard-case protection. Across all the datasets, CCSM consistently improves ranking quality against strong baselines, and the staged ablation shows an initial lift from memory calibration followed by a further benchmark-dependent gain from the full method. Ablations show an initial lift from memory calibration and a further, benchmark-dependent gain from the full method, while conflict-regime analysis and case studies indicate that the final refinement acts selectively when ambiguity remains after calibration.

1 Introduction

Sequential recommendation models users' evolving interests from behavior sequences (Hidasi et al., 2015; Kang and McAuley, 2018; Sun et al., 2019). Recently, large language models have opened a new direction for recommendation by bringing richer semantic understanding, longer-context reasoning, and more flexible natural-language interaction into the recommendation pipeline (Ji et al., 2024; Lyu et al., 2024; Liao et al., 2024; Yue et al., 2023; Zhu et al., 2024; Xu et al., 2024; Wu et al., 2024; Zhang et al., 2025). Building on this trend, recent studies have moved beyond using LLMs merely as recommenders or generators, and have begun to treat them as recommendation agents equipped with memory, reflection, and tool-use capabilities

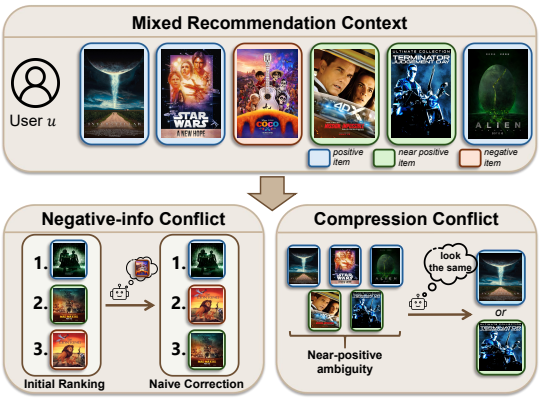

Figure 1: Motivation of CCSM. In mixed recommendation contexts, naive memory-based correction can be misled by negative signals and can also confuse similar candidate items, resulting in worse ranking results.

(Huang et al., 2025c; Wang et al., 2024b; Zhang et al., 2024; Shi et al., 2024b; Zhao et al., 2024; Fang et al., 2024; Wang et al., 2024c; Wu et al., 2025; Ma et al., 2025; Peng et al., 2025; Huang et al., 2025a). This shift makes recommendation increasingly rely on structured reasoning over user history, candidate evidence, and auxiliary signals.

Among these emerging capabilities, memory is especially important for sequential recommendation. A memory-enhanced recommendation agent can revisit fine-grained historical behaviors, recover long-range dependencies, and retain richer evidence than a plain top-K reranking pipeline, while recent memory-centered, hierarchical, and retrieval-augmented recommenders further suggest that structured memory can better support personalized reasoning than unstructured recall (Chen et al., 2024; Kim et al., 2025; Huang et al., 2025b; Nguyen et al., 2026). However, more history alone does not guarantee stable gains. Real user histories usually contain mixed interests, stale behaviors, noisy negatives, and conflict-heavy evidence that points in different directions. When such evidence

is aggregated too coarsely, additional memory can become counterproductive: the agent may overreact to isolated negative cues, compress multiple plausible interests and near-positive candidates into an overly coarse preference signal, or apply overly aggressive correction to candidates that are still compatible with the user's broader taste. As illustrated in Fig. 1, these failures often appear in two coupled forms: *negative-info conflict*, where conflict-heavy negative evidence is over-amplified during memory-based correction, and *compression conflict*, where coarse aggregation compresses multiple near-positive candidates into an overly similar representation. The central question is therefore not how to store more history, but how to organize memory evidence so that negative signals can make it to correct the ranking without suppressing plausible candidates.

These observations expose two recurring challenges for agentic recommendation with structured memory. The first is **Conflict-Calibrated Memory Reasoning**, which addresses the *negative-info conflict* in Fig. 1. In mixed user histories, memory-based correction can easily overreact to conflict-heavy negative cues and then pull the ranking away from the user's actual preference. Once this happens, additional memory no longer clarifies the decision and may instead distort it. This concern is consistent with earlier findings on passive negatives, false negatives, hard negatives, intent drift, and side-information fusion, all of which show that unreliable or over-weighted signals can mislead recommendation learning when they are handled too aggressively (Pan et al., 2023; Liu et al., 2023; Shi et al., 2024a; Liu et al., 2025; Chen et al., 2021; Jannach and Zanker, 2024; Lin et al., 2025; Hu et al., 2023; Xie et al., 2022). The second is **Top-Ranked Recommendation Fragility on Hard Cases**, which is closely related to the *compression conflict* in Fig. 1. Even when better memory improves overall ranking quality, the top-ranked recommendation can remain brittle on hard cases, especially when high-affinity candidates resemble previously rejected or ambiguous items. In these cases, a correction mechanism that is too uniform may suppress promising near-positive candidates together with truly irrelevant ones, hurting the most critical part of the recommendation list.

To address these challenges, we study CCSM, a structured memory correction method for agentic sequential recommendation. At its core, CCSM builds a *conflict-calibrated structured memory* module that explicitly organizes historical evidence and calibrates conflict-heavy negative signals before they dominate preference inference. This component is designed to make memory-based correction more stable under mixed and partially conflicting user histories. On top of it, CCSM further introduces an *affinity-aware hard-case protection* module that is invoked only when the current case is estimated to exhibit strong near-positive ambiguity. This adaptive module responds to hard-case uncertainty and provides targeted protection when aggressive correction is most likely to damage the top-ranked recommendation.

Across multiple sequential recommendation benchmarks, CCSM delivers stable ranking gains, while the staged ablation shows an initial lift from memory calibration and a further benchmark-dependent gain from the full method. Additional conflict-regime analysis and representative case studies then show that this last-stage refinement is selective: it mainly sharpens shortlist quality and can resolve representative ambiguous histories without acting like a uniform full-list rewrite. Our anonymized code is available at: `https://anonymous.4open.science/r/CCSM`

Based on these findings, our main contributions are threefold.

- We provide an in-depth analysis of structured memory correction in agentic sequential recommendation, and identify conflict calibration and adaptive hard-case protection as two key factors governing this method family.
- We introduce CCSM, which explicitly separates and calibrates conflict-heavy negative evidence through conflict-calibrated structured memory, thereby enabling more stable preference inference under mixed and partially conflicting user histories.
- Across multiple sequential recommendation benchmarks, together with conflict-regime analysis and representative case studies, we identify a staged improvement pattern: memory calibration provides an initial lift, and the full CCSM further improves recommendation quality through adaptive hard-case protection.

2 Related Work

From sequential recommendation to LLM4Rec. Sequential recommendation has long been built on interaction sequences, from early session-based

modeling to self-attentive and bidirectional transformer recommenders (Hidasi et al., 2015; Kang and McAuley, 2018; Sun et al., 2019). Recent LLM4Rec research extends this line along several routes. Early work mainly uses prompting, lightweight alignment, or semantic enrichment to adapt LLMs to recommendation (Lyu et al., 2024; Bao et al., 2023; Xu et al., 2024). Later studies show that LLMs can also act as rankers, rerankers, retrieval generators, or collaborative recommenders (Hou et al., 2024; Gao et al., 2025; Yue et al., 2023; Geng et al., 2022; Rajput et al., 2023; Ji et al., 2024; Zhu et al., 2024; Zheng et al., 2024). Other efforts strengthen user and item modeling through recommendation assistants, hierarchical modeling, retrieval augmentation, tokenization, or iterative preference alignment (Liao et al., 2024; Chen et al., 2024; Kim et al., 2025; Wang et al., 2024a; Deng et al., 2025). Recent surveys further summarize the broader rise of LLM-based and generative recommendation (Wu et al., 2024; Yang et al., 2025; Zhang et al., 2025). P5 reformulates recommendation as a text-to-text problem (Geng et al., 2022), LC-Rec integrates collaborative semantics into generation (Zheng et al., 2024), HLLM strengthens sequential recommendation through hierarchical item-user modeling (Chen et al., 2024), and ItemRAG introduces retrieval-augmented recommendation over item memories (Kim et al., 2025). At the same time, studies on false negatives, hard negatives, multi-interest behavior, intent variation, and adaptive signal fusion show that negative or heterogeneous evidence cannot be treated as uniformly reliable when modeling user preference (Pan et al., 2023; Liu et al., 2023; Shi et al., 2024a; Liu et al., 2025; Chen et al., 2021; Jannach and Zanker, 2024; Lin et al., 2025; Hu et al., 2023; Xie et al., 2022). These works establish the value of semantics, generation, ranking, and memory organization, but stop short of asking how memory should revise an already strong recommendation when the memory state itself is conflicted.

Agentic recommendation. A second line of research treats recommendation as an agentic process with memory, planning, tool use, and multi-agent collaboration (Huang et al., 2025c; Wang et al., 2024b; Zhang et al., 2024; Shi et al., 2024b; Zhao et al., 2024; Fang et al., 2024; Wang et al., 2024c; Wu et al., 2025; Ma et al., 2025; Li et al., 2026; Peng et al., 2025; Huang et al., 2025a). InteRecAgent and RecMind show that LLM agents can coordinate tool invocation and reflection for interactive recommendation (Huang et al., 2025c; Wang et al., 2024b). BLLP and ToolRec push this line toward long-term planning and explicit tool learning (Shi et al., 2024b; Zhao et al., 2024). MACRS, MACRec, STARec, AgentRec, and ChainRec further explore role decomposition, autonomous deliberate reasoning, multi-agent collaboration, and route-aware orchestration for recommendation workflows (Fang et al., 2024; Wang et al., 2024c; Wu et al., 2025; Ma et al., 2025; Li et al., 2026). Recent surveys consolidate these developments into the broader paradigm of agentic recommender systems (Peng et al., 2025; Huang et al., 2025a). Another nearby line of studies richer memory inside agentic recommenders, for example, through synergistic memory-reasoning assistants and cross-user memory evolution (Huang et al., 2025b; Nguyen et al., 2026). Most of these methods emphasize richer orchestration, broader interaction ability, or stronger general agent capabilities, while the earlier literature on false negatives, multi-interest behavior, and heterogeneous signal fusion mainly studies such difficulties from the perspective of learning or representation. Our work connects these two lines by focusing on a narrower inference-time question in agentic sequential recommendation: how structured memory should revise a current ranking when conflict-heavy evidence and hard, ambiguous cases interact around the top-ranked candidates.

3 Methods

3.1 Overview

Figure 2 shows the overall pipeline of CCSM. Given the user history and candidate set, a base recommender first produces an initial proposal shortlist. CCSM then builds a proposal-conditioned structured memory with supportive, conflict, and ambiguity compartments, applies an always-on conflict-calibrated correction, and further invokes affinity-aware protection only for ambiguity-prone cases. The following subsections present the problem setting, memory construction, stage-1 correction, stage-2 protection, and the final inference regime in order.

3.2 Problem Setup and Proposal-Conditioned Correction Setting

Let u denote a user, $\mathcal{H}_u = [i_1, i_2, \ldots, i_T]$ the observed interaction history, and $\mathcal{C}_u$ the candidate set for the current recommendation instance. Rather

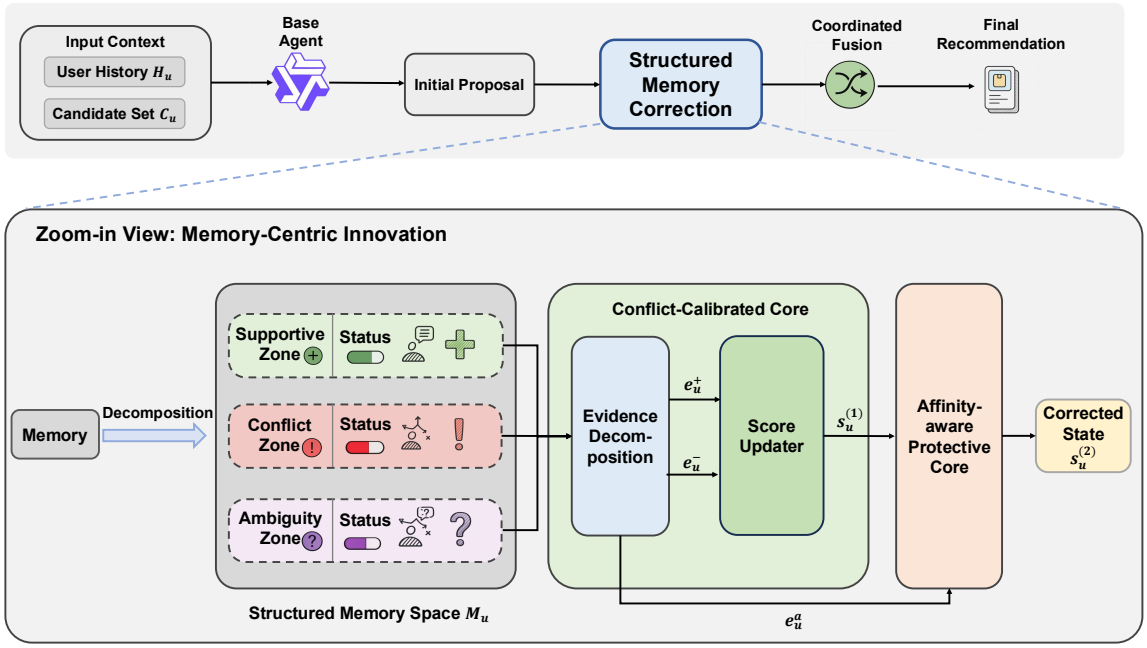

Figure 2: Framework of CCSM. A base recommender first produces an initial proposal. CCSM then organizes proposal-conditioned history into supportive, conflict, and ambiguity memory, performs conflict-calibrated correction, and selectively applies affinity-aware protection to obtain the final ranking.

than generating a recommendation from scratch, we assume a competent base recommender first produces a shortlist

$$B_u = [c_1, c_2, \ldots, c_K] \subseteq \mathcal{C}_u$$

together with base scores $s_u^{(0)}(c)$ for $c \in B_u$. Let $\pi_u^{(0)}$ denote the ranking induced by these base scores, and let π_u^* denote the final ranking after CCSM correction.

CCSM is therefore an inference-time correction layer on top of a strong initial proposal. Its purpose is not to replace the base recommendation logic, but to revise shortlist ordering when the current top-ranked candidates remain unstable under structured memory evidence. This intervention view is central to the method: the object being corrected is a shortlist-level ranking state rather than a full end-to-end recommendation generator.

We organize the user history into a proposal-conditioned structured memory bank

$$\mathcal{M}_u = (\mathcal{M}_u^+, \mathcal{M}_u^-, \mathcal{M}_u^a),$$

where $\mathcal{M}_u^+$ stores supportive traces, $\mathcal{M}_u^-$ stores conflict-heavy traces, and $\mathcal{M}_u^a$ stores affinity-sensitive ambiguity traces. The supportive compartment preserves evidence aligned with the shortlist, the conflict-heavy compartment records rejection-like or preference-tension evidence, and the ambiguity-sensitive compartment keeps high-affinity traces that may remain plausible even under mixed signals. CCSM then applies a compact two-stage pipeline: an always-on conflict-calibrated correction over the shortlist, followed by a selective protective refinement for ambiguity-prone candidates only.

3.3 Structured Memory Bank and Candidate-Level Evidence Encoding

The first technical step is to convert the history into typed memory objects that can be queried from the viewpoint of an individual candidate. Recent recommendation work suggests that structured memory and heterogeneous evidence roles are often more useful than flat recall alone (Huang et al., 2025b; Nguyen et al., 2026). Following this intuition, CCSM encodes historical interactions into a shared memory bank and allocates their contributions across the three typed compartments in $\mathcal{M}_u$ according to whether they explicitly reinforce the current shortlist, directly challenge it, or create high-affinity ambiguity.

For each candidate $c \in B_u$, CCSM reads the three compartments in a proposal-conditioned manner and forms

$$e_u^+(c), \quad e_u^-(c), \quad e_u^a(c),$$

where $e_u^+(c)$ summarizes supportive evidence, $e_u^-(c)$ summarizes conflict-heavy evidence, and $e_u^a(c)$ summarizes ambiguity-sensitive affinity cues. In addition to these candidate-level reads, CCSM keeps lightweight memory statistics that characterize how dense each evidence type is in the current user state, such as history length, positive density, dislike density, and affinity density. These descriptors are not treated as a standalone inference rule. Instead, they help contextualize whether local conflict should be interpreted as isolated or persistent relative to the broader memory state.

This design makes the later stages operational. Stage 1 no longer receives a vague memory summary, but a candidate-conditioned triple of supportive, conflict-heavy, and ambiguity-related evidence together with global density cues. The appendix later derives descriptive diagnostics from related quantities such as conflict overlap and affinity overlap, but those analysis-side heuristics should be read as post hoc summaries rather than as the deployed inference rule.

3.4 Conflict-Calibrated Memory Correction

The first CCSM stage is always on. Its goal is to revise the shortlist without letting a few locally strong rejection cues dominate the full decision. This concern is closely related to earlier observations on false negatives and hard-negative handling in sequential recommendation (Liu et al., 2023; Shi et al., 2024a), but here it appears as an inference-time correction problem over a candidate list rather than as a training-data issue alone.

Operationally, this stage can be understood through four functional roles: support/conflict evidence splitting, conflict-density estimation, calibration, and score correction. The evidence splitter converts the candidate-conditioned reads into a supportive response $p_u(c)$ and a conflict response $n_u(c)$. A conflict-density estimate then interprets whether the challenging evidence is isolated and noisy or repeated and coherent across the current memory state. Based on these signals, the stage forms a candidate-specific calibration strength $\lambda_u(c)$ that determines how much of the conflict channel should actually be allowed to modify the base score.

A useful score update view of stage 1 is

$$s_u^{(1)}(c) = F_{\text{corr}}(s_u^{(0)}(c), p_u(c), n_u(c), \lambda_u(c)),$$

where $F_{\text{corr}}(\cdot)$ denotes a calibrated score-update rule. The role of this update is inherently asymmetric: supportive evidence can further reinforce a plausible candidate, while conflict-heavy evidence is only admitted after careful calibration rather than acting as a direct veto. As a result, stage 1 stabilizes the shortlist under mixed histories and produces the memory-calibrated ranking

$$\pi_u^{(1)} = \text{sort}_{c \in B_u}(s_u^{(1)}(c)).$$

This stage is the main source of the broad lift observed in the ablation. It acts on every candidate in the shortlist, but its effect is controlled by the calibration signal, so aggressive revision is reserved for cases where conflict remains dense and consistent relative to the rest of the memory state.

3.5 Ambiguity-Aware Protective Refinement

A second difficulty remains after stage 1: some candidates are still near-positive and highly compatible with the user's broader taste, yet they overlap with conflict-heavy traces strongly enough to risk being over-suppressed. CCSM therefore places a second, selective controller on top of the memory-calibrated shortlist.

Operationally, this stage can be viewed through five functional roles: affinity reading, ambiguity estimation, protection decision, shortlist preservation, and conservative fusion. The affinity readout converts $e_u^a(c)$ into an affinity response $a_u(c)$ that measures how plausible the candidate remains under the broader user taste. An ambiguity signal $b_u(c)$ then combines this affinity response with the residual conflict left after stage 1. Intuitively, $b_u(c)$ becomes large only when two conditions hold simultaneously: the candidate still looks highly compatible, and the calibrated conflict signal has not fully resolved the tension around it.

This ambiguity signal determines whether candidate c should remain on the ordinary stage-1 path or receive an additional protective refinement. When such refinement is activated, CCSM distinguishes between cases where the evidence supports a decisive correction near the top of the ranking and cases where the safer behavior is to preserve the current top-ranked item and only refine the shortlist around it. An abstract score update view of this stage is

$$s_u^*(c) = F_{\text{prot}}(s_u^{(1)}(c), a_u(c), b_u(c)),$$

where $F_{\text{prot}}(\cdot)$ denotes a conservative fusion rule over the stage-1 score and the ambiguity-sensitive protection signal.

Table 1: Processed dataset statistics used in this paper after preprocessing and cohort construction.

Dataset	#Users	#Items	#Interactions
Toys&Games	50,000	349,613	277,432
Movies&TV	20,000	164,203	157,346
Yelp	50,000	149,659	488,995

This stage is therefore not a generic reranker. It is a protective controller activated only for ambiguity-prone candidates and can express two representative outcomes already reflected in the qualitative analyses: decisive top-ranked correction in some near-positive cases, or conservative shortlist refinement when the evidence does not justify overturning the current top-1.

3.6 Inference Regimes and Final Ranking

In practice, CCSM exhibits three inference behaviors. In ordinary cases, the method returns the stage-1 ranking $\pi_u^{(1)}$ directly. On some hard cases, the protective refinement can support a decisive correction near the top of the ranking. In other hard cases, the safer behavior is to keep the current top-ranked item in place and apply only a conservative shortlist refinement. CCSM finally sorts the fused scores $s_u^*(c)$ over the shortlist and outputs

$$\pi_u^* = \mathrm{sort}_{c \in B_u}(s_u^*(c)).$$

Because stage 2 acts only when ambiguity-sensitive evidence remains unresolved after calibration, CCSM remains selective rather than uniformly reflective. This behavior matches the empirical story developed later in the paper: the memory-calibration stage provides the broad initial improvement, while the final CCSM stage supplies a benchmark-dependent refinement whose effect is strongest when unresolved ambiguity remains after memory calibration.

4 Experiments

4.1 Experimental Setup

Datasets and Preprocessing. We evaluate on three sequential recommendation benchmarks: Amazon[1] Toys&Games, Amazon Movies&TV, and Yelp[2]. The statistics reported here describe the processed datasets used in this paper rather than the raw corpora. For the two Amazon datasets, we first construct a strict item universe by keeping items with valid metadata titles and sufficient interaction frequency, rebuild user sequences on that filtered universe, and then fix a reproducible user cohort for the paper's main experiments. For Yelp, we first keep users with at least three review interactions, build a strict business universe, filter out short sequences, and then construct a fixed reproduction cohort accordingly. As a result, the user counts reflect deliberate post-preprocessing cohort construction rather than the raw dataset sizes. Table 1 summarizes the resulting processed datasets used in our experiments. All methods are then consistently evaluated under the same next-item reranking protocol, where each held-out target is ranked against a fixed candidate pool.

Evaluation Metrics. We evaluate with standard cutoff-based HR and NDCG metrics. The main comparison reports Pass@1, which is equivalent to HR@1, together with HR@5 and NDCG@5, which capture top-ranked correction and shortlist quality. All comparative claims in this section are restricted to rows evaluated under the same held-out protocol.

Baselines. Traditional baselines include SASRec (Kang and McAuley, 2018), BSARec (Shin et al., 2024), FMLP-Rec (Zhou et al., 2022), Mamba4Rec (Liu et al., 2024), and HSTU (Zhai et al., 2024), covering strong sequential recommenders from transformer, MLP, state-space, and transductive-generation lines.

The **Generative** baselines in the main comparison are TIGER (Rajput et al., 2023) and OneRec (Deng et al., 2025), representing autoregressive retrieval-style recommendation and unified retrieve-and-rank generation.

LLM4Rec baselines include TallRec (Bao et al., 2023), HLLM (Chen et al., 2024), and ItemRAG (Kim et al., 2025), spanning lightweight alignment, hierarchical user-item modeling, and retrieval-augmented recommendation.

Agent4Rec baselines include Rec Agent (Huang et al., 2025c), STARec (Wu et al., 2025), AgentRec (Ma et al., 2025), and ToolRec (Zhao et al., 2024), covering tool-using, deliberate-reasoning, multi-agent, and tool-learning agentic recommendation workflows.

Our Variants. We denote the full method as *CCSM*. The ablation study additionally reports an intermediate *Memory Calibration* stage between

[1] https://amazon-reviews-2023.github.io/
[2] https://business.yelp.com/data/resources/open-dataset/

Table 2: Main performance comparison across three sequential recommendation datasets. Results are reported with Pass@1, HR@5, and NDCG@5. The best and second-best results in each metric column are highlighted in **bold** and underline, respectively.

Model Class	Model	Toys&Games			Movies&TV			Yelp		
		Pass@1	HR@5	NDCG@5	Pass@1	HR@5	NDCG@5	Pass@1	HR@5	NDCG@5
Traditional	SASRec	0.0041	0.0085	0.0064	0.0068	0.0267	0.0167	0.0035	0.0121	0.0078
	BSARec	0.0086	0.0114	0.0101	0.0106	0.0161	0.0135	0.0027	0.0091	0.0059
	FMLP-Rec	0.0086	0.0149	0.0120	0.0114	0.0236	0.0179	0.0042	0.0115	0.0079
	Mamba4Rec	0.0081	0.0140	0.0113	0.0115	0.0228	0.0176	0.0041	0.0094	0.0068
	HSTU	0.0079	0.0136	0.0103	0.0122	0.0195	0.0161	0.0030	0.0114	0.0072
Generative	TIGER	0.0085	0.0150	0.0122	0.0119	0.0201	0.0182	0.0086	0.0132	0.0108
	OneRec	0.0187	0.0249	0.0211	0.0201	0.0278	0.0241	0.0103	0.0181	0.0149
LLM4Rec	TallRec	0.0121	0.0204	0.0164	0.0123	0.0286	0.0207	0.0064	0.0082	0.0073
	HLLM	0.0179	0.0317	0.0280	0.0256	0.0448	0.0349	0.0117	0.0195	0.0152
	ItemRAG	0.0148	0.0314	0.0235	0.0209	0.0454	0.0284	0.0127	0.0176	0.0155
Agent4Rec	Rec Agent	0.0038	0.0131	0.0085	0.0096	0.0166	0.0133	0.0028	0.0105	0.0067
	STARec	0.0076	0.0108	0.0094	0.0105	0.0190	0.0152	0.0061	0.0135	0.0099
	AgentRec	0.0072	0.0104	0.0090	0.0104	0.0185	0.0149	0.0044	0.0129	0.0088
	ToolRec	0.0042	0.0084	0.0064	0.0096	0.0166	0.0133	0.0028	0.0105	0.0067
Ours	**CCSM**	**0.0209**	**0.0348**	**0.0281**	**0.0259**	**0.0544**	**0.0408**	**0.0131**	**0.0288**	**0.0216**

Table 3: Progressive ablation on the three reported benchmarks. Each row shows one stage of the method: starting from the baseline stage, activating memory calibration, and then completing CCSM. Metrics are reported with Pass@1, HR@5, and NDCG@5, and the best value in each dataset/metric block is highlighted in **bold**.

Setting	Toys&Games			Movies&TV			Yelp		
	Pass@1	**HR@5**	**NDCG@5**	**Pass@1**	**HR@5**	**NDCG@5**	**Pass@1**	**HR@5**	**NDCG@5**
Local Baseline	0.01210	0.02040	0.01640	0.01230	0.02860	0.02070	0.00640	0.00820	0.00730
Memory Calibration	0.01575	0.02667	0.02142	0.01353	0.03118	0.02263	0.00932	0.01766	0.01383
CCSM	**0.02090**	**0.03476**	**0.02806**	**0.02585**	**0.05440**	**0.04079**	**0.01308**	**0.02882**	**0.02160**

the baseline and the final method.

4.2 Performance Comparison

Table 2 compares CCSM against **Traditional**, **Generative**, **LLM4Rec**, and **Agent4Rec** baselines under the unified next-item prediction protocol.

Table 2 presents the main performance comparison across three sequential recommendation benchmarks. Overall, CCSM is the strongest reported method: it achieves the best result in every metric column on Toys&Games, Movies&TV, and Yelp under the shared next-item prediction protocol. The most competitive baselines are concentrated in the **Generative** and **LLM4Rec** groups, especially OneRec, HLLM, and ItemRAG, which consistently form the closest non-CCSM rows across different datasets and metrics. The **Agent4Rec** baselines also remain competitive, particularly on Yelp, where STARec gives the strongest baseline HR@5 and NDCG@5 within that block. Even so, CCSM maintains the overall best performance across all three benchmarks, suggesting that its structured memory correction yields a stronger and more consistent ranking advantage than the compared traditional, generative, LLM-based, and agent-based recommenders under the same evaluation setting. All methods are evaluated on the same Qwen3.5-4B backbone. As some agent-based baselines were not originally developed or tuned for this small model scale, their results should be interpreted under this unified setting rather than as their fully optimized performance on recommendations.

4.3 Ablation Study

Table 3 reports the progressive ablation on the three reported benchmarks, moving from the baseline stage to *Memory Calibration* and then to the full *CCSM* configuration. In this table, the baseline

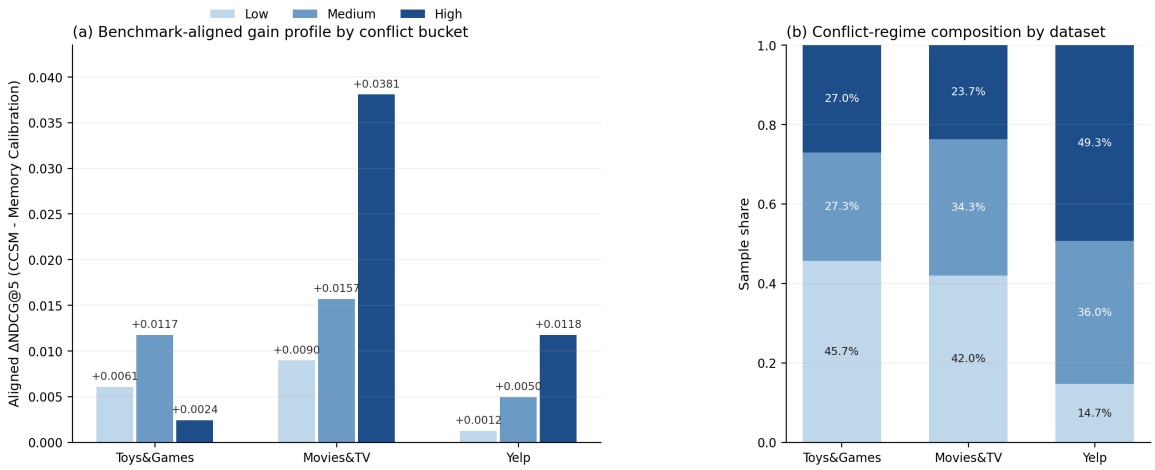

Figure 3: Conflict-regime analysis for the final CCSM refinement. Panel (a) reports a bucket-wise NDCG@5 gain profile whose weighted benchmark average matches the overall *Memory Calibration* → *CCSM* improvement in Table 3. Panel (b) shows the corresponding conflict-regime composition on each benchmark.

stage is aligned with the TallRec row in the main comparison, while *Memory Calibration* reports the corresponding intermediate stage before the final CCSM refinement.

The ablation shows a clear two-stage pattern. Across all three benchmarks, *Memory Calibration* provides the initial lift over the baseline, while the full *CCSM* configuration adds a second-stage gain whose size depends on the benchmark. That second-stage gain is clearest on Movies&TV, whereas Toys&Games and Yelp show smaller but still positive refinements. Read at the pattern level, this progression is consistent with the paper's staged design rationale: memory calibration contributes to the broad improvement, while the final model plays a more selective refinement role.

4.4 Memory Statistics Analysis

To examine where the final CCSM stage changes the recommendation, we derive a simple *memory conflict score* from the *Memory Calibration* outputs and split samples into low-, medium-, and high-conflict terciles accordingly. The exact definition is given in the appendix. This analysis allows us to inspect not only whether the second stage helps, but also under what conflict conditions its effect becomes most visible.

Figure 3(a) shows that the final CCSM refinement is selective rather than uniform. Movies&TV exhibits the clearest high-conflict signal, with the most visible second-stage gain concentrated in the high-conflict bucket. Toys&Games remains selective as well, but its gain is concentrated outside the highest-conflict bucket, indicating a different conflict profile. Yelp shows a smaller but still non-uniform pattern, with the benefit remaining more modest overall.

Figure 3(b) further contextualizes this benchmark-level variation by showing that the conflict composition itself changes across Toys&Games, Movies&TV, and Yelp. Taken together, the two panels suggest that the usefulness of the final CCSM stage depends on both where the gain is concentrated and how much mass each benchmark places in that regime. Appendix Figure 4 provides two representative examples that make this conflict-sensitive refinement more concrete and easier to understand.

5 Conclusion

We studied conflict-calibrated structured memory correction for agentic sequential recommendation. Rather than assuming that more memory automatically improves recommendations, CCSM organizes historical evidence into a structured memory state, calibrates conflict-heavy negative evidence before it affects revision of the current recommendation, and applies additional protection only when hard cases remain ambiguous after memory calibration. Overall, the results suggest that memory correction is most effective when treated as a structured and staged process: conflict calibration provides the primary source of improvement by stabilizing revision under mixed evidence, while selective hard-case refinement offers further gains when unresolved ambiguity still remains.

Limitations

Our study has two main limitations. First, the current evaluation is restricted to offline next-item recommendation benchmarks with textual item information. While this setting is appropriate for studying structured memory correction, it does not fully capture richer recommendation environments that involve multimodal item content, broader candidate universes, or interactive user feedback. Second, the present paper focuses on a single method family and a fixed evaluation protocol. Although the reported results and analyses consistently support the value of CCSM in this setting, broader validation on larger backbones and more diverse recommendation scenarios would help clarify how well the same conflict-calibrated design scales beyond the current benchmark suite.

References

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

Junyi Chen, Lu Chi, Bingyue Peng, and Zehuan Yuan. 2024. Hllm: Enhancing sequential recommendations via hierarchical large language models for item and user modeling. *arXiv preprint arXiv:2409.12740*.

Wanyu Chen, Pengjie Ren, Fei Cai, Fei Sun, and Maarten De Rijke. 2021. Multi-interest diversification for end-to-end sequential recommendation. *ACM Transactions on Information Systems (TOIS)*, 40(1):1–30.

Jiaxin Deng, Shiyao Wang, Kuo Cai, Lejian Ren, Qigen Hu, Weifeng Ding, Qiang Luo, and Guorui Zhou. 2025. Onerec: Unifying retrieve and rank with generative recommender and iterative preference alignment. *arXiv preprint arXiv:2502.18965*.

Jiabao Fang, Shen Gao, Pengjie Ren, Xiuying Chen, Suzan Verberne, and Zhaochun Ren. 2024. A multi-agent conversational recommender system. *arXiv preprint arXiv:2402.01135*.

Jingtong Gao, Bo Chen, Xiangyu Zhao, Weiwen Liu, Xiangyang Li, Yichao Wang, Wanyu Wang, Huifeng Guo, and Ruiming Tang. 2025. Llm4rerank: Llm-based auto-reranking framework for recommendations. In *Proceedings of the ACM on Web Conference 2025*, pages 228–239.

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

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

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

Hengchang Hu, Wei Guo, Yong Liu, and Min-Yen Kan. 2023. Adaptive multi-modalities fusion in sequential recommendation systems. In *Proceedings of the 32nd ACM international conference on information and knowledge management*, pages 843–853.

Chengkai Huang, Junda Wu, Yu Xia, Zixu Yu, Ruhan Wang, Tong Yu, Ruiyi Zhang, Ryan A Rossi, Branislav Kveton, Dongruo Zhou, and 1 others. 2025a. Towards agentic recommender systems in the era of multimodal large language models. *arXiv preprint arXiv:2503.16734*.

Jiani Huang, Xingchen Zou, Lianghao Xia, and Qing Li. 2025b. Mr. rec: Synergizing memory and reasoning for personalized recommendation assistant with llms. *arXiv preprint arXiv:2510.14629*.

Xu Huang, Jianxun Lian, Yuxuan Lei, Jing Yao, Defu Lian, and Xing Xie. 2025c. Recommender ai agent: Integrating large language models for interactive recommendations. *ACM Transactions on Information Systems*, 43(4):1–33.

Dietmar Jannach and Markus Zanker. 2024. A survey on intent-aware recommender systems. *ACM Transactions on Recommender Systems*, 3(2):1–32.

Jianchao Ji, Zelong Li, Shuyuan Xu, Wenyue Hua, Yingqiang Ge, Juntao Tan, and Yongfeng Zhang. 2024. Genrec: Large language model for generative recommendation. In *European Conference on Information Retrieval*, pages 494–502. Springer.

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

Sunwoo Kim, Geon Lee, Kyungho Kim, Jaemin Yoo, and Kijung Shin. 2025. Itemrag: Item-based retrieval-augmented generation for llm-based recommendation. *arXiv preprint arXiv:2511.15141*.

Fuchun Li, Qian Li, Xingyu Gao, Bocheng Pan, Yang Wu, Jun Zhang, Huan Yu, Jie Jiang, Jinsheng Xiao, and Hailong Shi. 2026. Chainrec: An agentic recommender learning to route tool chains for diverse and evolving interests. *arXiv preprint arXiv:2602.10490*.

Jiayi Liao, Sihang Li, Zhengyi Yang, Jiancan Wu, Yancheng Yuan, Xiang Wang, and Xiangnan He. 2024. Llara: Large language-recommendation assistant. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1785–1795.

Xiaolin Lin, Weike Pan, and Zhong Ming. 2025. Towards interest drift-driven user representation learning in sequential recommendation. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1541–1551.

Chengkai Liu, Jianghao Lin, Jianling Wang, Hanzhou Liu, and James Caverlee. 2024. Mamba4rec: Towards efficient sequential recommendation with selective state space models. *arXiv preprint arXiv:2403.03900*.

Xiaoyang Liu, Chong Liu, Pinzheng Wang, Rongqin Zheng, Lixin Zhang, Leyu Lin, Zhijun Chen, and Liangliang Fu. 2023. Ufnrec: utilizing false negative samples for sequential recommendation. In *Proceedings of the 2023 SIAM International Conference on Data Mining (SDM)*, pages 46–54. SIAM.

Xiaoyu Liu, Jiaxin Yuan, Yuhang Zhou, Jingling Li, Furong Huang, and Wei Ai. 2025. Csrec: Rethinking sequential recommendation from a causal perspective. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1562–1571.

Hanjia Lyu, Song Jiang, Hanqing Zeng, Yinglong Xia, Qifan Wang, Si Zhang, Ren Chen, Chris Leung, Jiajie Tang, and Jiebo Luo. 2024. Llm-rec: Personalized recommendation via prompting large language models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 583–612.

Bo Ma, Hang Li, ZeHua Hu, XiaoFan Gui, LuYao Liu, and Simon Lau. 2025. Agentrec: Next-generation llm-powered multi-agent collaborative recommendation with adaptive intelligence. *arXiv preprint arXiv:2510.01609*.

Minh-Duc Nguyen, Hai-Dang Kieu, and Dung D Le. 2026. Amem4rec: Leveraging cross-user similarity for memory evolution in agentic llm recommenders. *arXiv preprint arXiv:2602.08837*.

Yunzhu Pan, Chen Gao, Jianxin Chang, Yanan Niu, Yang Song, Kun Gai, Depeng Jin, and Yong Li. 2023. Understanding and modeling passive-negative feedback for short-video sequential recommendation. In *Proceedings of the 17th ACM conference on recommender systems*, pages 540–550.

Qiyao Peng, Hongtao Liu, Hua Huang, Qing Yang, and Minglai Shao. 2025. A survey on llm-powered agents for recommender systems. *arXiv preprint arXiv:2502.10050*.

Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan Hulikal Keshavan, Trung Vu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Tran, Jonah Samost, and 1 others. 2023. Recommender systems with generative retrieval. *Advances in Neural Information Processing Systems*, 36:10299–10315.

Kexin Shi, Jing Zhang, Linjiajie Fang, Wenjia Wang, and Bingyi Jing. 2024a. Enhanced bayesian personalized ranking for robust hard negative sampling in recommender systems. *arXiv preprint arXiv:2403.19276*.

Wentao Shi, Xiangnan He, Yang Zhang, Chongming Gao, Xinyue Li, Jizhi Zhang, Qifan Wang, and Fuli Feng. 2024b. 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*, pages 1893–1903.

Yehjin Shin, Jeongwhan Choi, Hyowon Wi, and Noseong Park. 2024. An attentive inductive bias for sequential recommendation beyond the self-attention. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pages 8984–8992.

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

Wenjie Wang, Honghui Bao, Xinyu Lin, Jizhi Zhang, Yongqi Li, Fuli Feng, See-Kiong Ng, and Tat-Seng Chua. 2024a. Learnable item tokenization for generative recommendation. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pages 2400–2409.

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

Zhefan Wang, Yuanqing Yu, Wendi Zheng, Weizhi Ma, and Min Zhang. 2024c. Macrec: A multi-agent collaboration framework for recommendation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2760–2764.

Chenghao Wu, Ruiyang Ren, Junjie Zhang, Ruirui Wang, Zhongrui Ma, Qi Ye, and Wayne Xin Zhao. 2025. Starec: An efficient agent framework for recommender systems via autonomous deliberate reasoning. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, pages 3355–3365.

Likang Wu, Zhi Zheng, Zhaopeng Qiu, Hao Wang, Hongchao Gu, Tingjia Shen, Chuan Qin, Chen Zhu, Hengshu Zhu, Qi Liu, and 1 others. 2024. A survey on large language models for recommendation. *World Wide Web*, 27(5):60.

Yueqi Xie, Peilin Zhou, and Sunghun Kim. 2022. Decoupled side information fusion for sequential recommendation. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval*, pages 1611–1621.

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

Zhen Yang, Haitao Lin, Ziji Zhang, and 1 others. 2025. Gr-llms: Recent advances in generative recommendation based on large language models. *arXiv preprint arXiv:2507.06507*.

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

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

An Zhang, Yuxin Chen, Leheng Sheng, Xiang Wang, and Tat-Seng Chua. 2024. On generative agents in recommendation. In *Proceedings of the 47th international ACM SIGIR conference on research and development in Information Retrieval*, pages 1807–1817.

Weizhi Zhang, Yuanchen Bei, Liangwei Yang, Henry Peng Zou, Peilin Zhou, Aiwei Liu, Yinghui Li, Hao Chen, Jianling Wang, Yu Wang, and 1 others. 2025. Cold-start recommendation towards the era of large language models (llms): A comprehensive survey and roadmap. *arXiv preprint arXiv:2501.01945*.

Yuyue Zhao, Jiancan Wu, Xiang Wang, Wei Tang, Dingxian Wang, and Maarten De Rijke. 2024. Let me do it for you: Towards llm empowered recommendation via tool learning. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1796–1806.

Bowen Zheng, Yupeng Hou, Hongyu Lu, Yu Chen, Wayne Xin Zhao, Ming Chen, and Ji-Rong Wen. 2024. Adapting large language models by integrating collaborative semantics for recommendation. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*, pages 1435–1448. IEEE.

Kun Zhou, Hui Yu, Wayne Xin Zhao, and Ji-Rong Wen. 2022. Filter-enhanced mlp is all you need for sequential recommendation. In *Proceedings of the ACM web conference 2022*, pages 2388–2399.

Yaochen Zhu, Liang Wu, Qi Guo, Liangjie Hong, and Jundong Li. 2024. Collaborative large language model for recommender systems. In *Proceedings of the ACM Web Conference 2024*, pages 3162–3172.

A Appendix

A.1 Additional Experimental Details

The main text reports Pass@1 together with HR@5 and NDCG@5. This appendix collects the supporting configuration details and qualitative evidence that help interpret the main benchmark results.

A.2 Shared Configuration

All reported benchmark experiments share the same high-level structure and are run under the same next-item prediction protocol. Across Toys&Games, Movies&TV, and Yelp, we keep the method fixed while swapping in dataset-specific candidate universes, item text, splits, and caches. The baseline stage, memory-calibration stage, and full CCSM stage, therefore, remain directly comparable within each benchmark.

To keep the three benchmark runs aligned, we also use the same starting training scale: batch size 256, learning rate 0.001, 20 epochs, early-stop patience 4, hidden dimension 64, and dropout 0.1. This shared configuration keeps the three benchmarks directly comparable under the same training scale.

A.3 Evaluation Scope

We use three public benchmark releases, namely Toys&Games, Movies&TV, and Yelp, and evaluate all methods under the same next-item prediction protocol. For each user, the observed interaction history forms the recommendation context, and the held-out target item is ranked against a fixed candidate set at test time. The main paper reports Pass@1 together with HR@5 and NDCG@5, and the supporting analyses reuse the same evaluation protocol so that benchmark-level comparison, ablation, conflict-regime analysis, and case-study discussion remain consistent with one another.

A.4 Memory-Statistics Heuristic

For the conflict-regime analysis in Section 4, we compute a *memory conflict score* from the *Memory Calibration* outputs:

$$s_u = 0.5\,\bar{c}_u + 0.3\,d_u + 0.2\,\bar{a}_u,$$

where $\bar{c}_u$ is the mean top-3 `negative_conflict_ratio`, d_u is the normalized dislike ratio `dislike_count/history_len`, and

Case 1 *Movies&TV* High conflict (0.173) **Memory Snapshot** `hist=2, pos=2, dislike=0, aff=4`	**Memory Calibration.** Top-1: Veronica Mars: Season 1 (2012); target Veronica Mars: Season 3 (2012) at rank 2. **CCSM.** Target Veronica Mars: Season 3 (2012) reaches rank 1; next competitor: Veronica Mars: Season 1 (2012). **Takeaway.** Same-franchise ambiguity is resolved: CCSM promotes the target from rank 2 to rank 1 ahead of a closely related Veronica Mars candidate. NDCG@5 change: +0.369.

Case 2 *Yelp* High conflict (0.182) **Memory Snapshot** `hist=9, pos=3, dislike=2, aff=3`	**Memory Calibration.** Top-1: Indian Rocks Beach (Indian Rocks Beach, FL) [...; target Indian Rocks Beach Nature Preserve (Indian Ro... at rank 5. **CCSM.** Top-1: Indian Rocks Beach (Indian Rocks Beach, FL) [...; target Indian Rocks Beach Nature Preserve (Indian Ro... at rank 4. **Takeaway.** Shortlist refinement stays conservative: CCSM keeps the same beach-focused top-1 while moving the target from rank 5 to rank 4. NDCG@5 change: +0.044.

Figure 4: Representative aligned memory cases under *Memory Calibration* and *CCSM*. The cases illustrate two different ways the final CCSM stage helps: a decisive top-ranked swap under same-franchise ambiguity and a conservative shortlist refinement that leaves the existing top-1 intact.

$\bar{a}_u$ is the mean top-3 `affinity_overlap`. We then sort the samples by s_u and split them into terciles, which define the low-, medium-, and high-conflict buckets used in Figure 3. The heuristic is intentionally lightweight: it is used only to partition samples for descriptive analysis, not to define a new optimization objective.

A.5 Representative Memory Cases

Figure 4 complements the bucketed statistics with two representative cases. The Movies&TV example mirrors the high-conflict regime where the final CCSM stage can change the top-ranked recommendation itself: the target *Veronica Mars: Season 3 (2012)* is promoted above a closely related same-franchise alternative. The Yelp example shows a different pattern: CCSM keeps the same beach-focused top-1 in place and only advances the target within the shortlist. Together, the two cases make the benchmark-level statistics easier to interpret.

A.6 Experimental Reporting Notes

Unless explicitly stated otherwise, all reported results are averaged over multiple independent runs under the same aligned evaluation protocol. The main tables report mean performance across repeated trials, and statistical significance is assessed with a significance level of 0.05, i.e., differences are considered significant when $p < 0.05$. Validation splits are used for model selection and early stopping, and all methods are compared under the same evaluation setting.

B Responsible Research Statement

B.1 Potential Risks

Although our study is limited to offline next-item recommendation benchmarks, selective correction systems still carry meaningful deployment risks. First, a correction workflow that repeatedly reinforces dominant signals from user history may amplify popularity bias or reduce exposure diversity, especially when stable high-frequency preferences are easier to verify than niche interests. Second, because the workflow reasons over behavioral traces and longer-range history, it may infer sensitive or transient user preferences that are not explicitly intended for profiling. Third, incorrect challenger activation or over-confident fusion may suppress exploratory items or over-correct borderline cases, producing rankings that appear more certain than the underlying evidence justifies. For these reasons, offline gains should not be treated as sufficient evidence for direct deployment without additional privacy, fairness, and intervention-auditing safeguards.

B.2 Data and Privacy Notes

We use public benchmark releases for sequential recommendation and do not collect new user data or new human annotations for this work. The experiments operate on benchmark-provided user/item identifiers, interaction histories, and item-side metadata released for research use. In our pipeline, we preserve anonymized identifiers, report only aggregate metrics in the paper, and do not release user-level traces as paper artifacts. At the same time, recommendation datasets are derived from user behavior and may still encode sensitive preferences or inherit noisy and potentially undesirable content from the original platforms. We therefore treat these benchmarks as potentially sensitive research artifacts and restrict our claims to offline benchmark evaluation rather than deployment safety.

B.3 Use of AI Assistants

AI assistants were used in a limited supporting role during writing and coding. In particular, they were used for wording refinement, structural editing, and small-scale coding assistance. All experimental design decisions, result checks, and final manuscript content were reviewed by the authors, who take responsibility for the correctness of the reported claims.