Which is Better: A Selective Recommendation Correction Framework via Case-Conditioned Challenger Routing

Anonymous ACL submission

Abstract

Sequential recommendation models users' evolving preferences, and recent large language models have improved recommendation by enabling richer semantic understanding of user histories. However, existing LLM-based recommenders and recommendation agents often apply a uniform reasoning workflow to all cases, which can mis-handle mixed preferences and blur the roles of heterogeneous recommendation signals. We propose a selective recommendation correction framework that starts from a strong base recommender, routes only ambiguous cases to signal-specific challengers, and uses verification-controlled fusion to revise the current ranking conservatively. This design keeps easy cases lightweight while enabling deeper correction when the history or candidate set is genuinely uncertain. Experiments on four sequential recommendation benchmarks show that the framework outperforms strong aligned baselines, and further analyses indicate that case-conditioned routing is the main source of gains and that the effective correction depth varies across datasets.

1 Introduction

Recommendation systems are a cornerstone of modern online platforms, and sequential recommendation has been widely used to model users' evolving preferences from historical interactions (Hidasi et al., 2015; Kang and McAuley, 2018; Sun et al., 2019). Recently, large language models have opened a new direction for recommendation (Ji et al., 2024; Xu et al., 2024; Zhang et al., 2025a). Compared with traditional recommenders that mainly rely on ID-based interactions and latent embeddings, LLMs can exploit rich semantic information from item content, reason over long-range dependencies in user histories, and express recommendations in a more interpretable natural-language form. These capabilities make them especially appealing for alleviating long-standing weaknesses of classical recommenders in semantic understanding, cold-start generalization, and explainability (Xu et al., 2024; Zhang et al., 2025a).

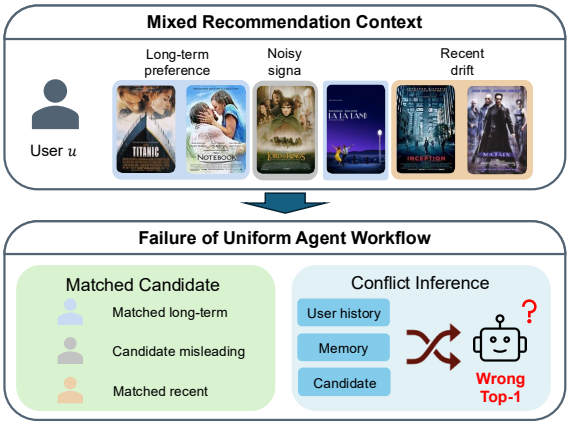

Figure 1: Motivation for selective recommendation correction. A mixed recommendation context may lead to the failure of uniform agent workflow.

Building on this progress, recent studies have moved beyond using LLMs merely as recommenders or generators, and have begun to treat them as recommendation agents (Wang et al., 2024a; Shi et al., 2024; Zhao et al., 2024; Huang et al., 2025c; Peng et al., 2025; Huang et al., 2025a). By endowing LLMs with planning, memory, and tool-use abilities, Rec Agents can interact with external recommenders, retrieve relevant evidence, and organize multi-step decision processes for more flexible recommendation behaviors (Wang et al., 2024a; Shi et al., 2024; Zhao et al., 2024; Huang et al., 2025c). This shift is attractive because it allows recommendation systems to go beyond one-shot scoring and instead perform structured reasoning over user context, external knowledge, and recommendation-specific signals.

However, as illustrated in Figure 1, this shift also exposes two recurring challenges for agentic recommendation. The first is **Mixed Preference Reasoning**. Unlike open-ended natural language

tasks, recommendation inputs often contain mixed, partial, and even competing interests within a single user history. Applying the same reasoning pattern to every sample can encourage the agent to force a single coherent interpretation onto inherently mixed preferences, which may lead to incorrect judgments about what the user currently wants (Chen et al., 2021; Jannach and Zanker, 2024; Lin et al., 2025). The second is **Heterogeneous Signal Integration**. In recommendation, the input context is often composed of multiple forms of data, and such multi-source information can vary substantially in semantics, reliability, and suitability across instances, making agent workflows either hard to exploit effectively or too expensive to analyze in depth (Hu et al., 2023; Xie et al., 2022). When these signals are processed in a coarse and uniform manner, their roles can easily become blurred: complementary evidence may be treated as redundant, while conflicting evidence may be fused too early, ultimately misleading the agent's preference inference (Li et al., 2026). These recurring issues motivate a Rec Agent that is adaptive in workflow and selective in how it leverages heterogeneous recommendation signals.

To address this need, we propose a case-conditioned challenger routing framework for selective recommendation correction in sequential recommendation. At a high level, we treat recommendation as *selective correction* on top of a strong base recommender, rather than forcing every instance through the same reasoning routine. To mitigate the **Mixed Preference Reasoning**, we organize recommendation instances into interpretable cases and allow the workflow to follow different intervention paths for different cases, which reduces the risk of over-reasoning easy cases and under-handling difficult ones. To address **Heterogeneous Signal Integration**, we further introduce signal-specific challengers and use verification-controlled fusion to regulate how complex evidence revises the current proposal, turning heterogeneous information from a source of noise into a useful complementary signal. In this way, the framework becomes both case-aware and source-aware, yielding more reliable top-ranked recommendations. Extensive experiments on sequential recommendation benchmarks show strong gains over aligned baselines and support the effectiveness of selective challenger routing under verification-controlled fusion. Our contributions are three-fold:

- First, we formulate agentic recommendation in sequential recommendation as a selective correction problem on top of a strong base recommender, and introduce case-conditioned challenger routing as the mechanism that decides when different instances require different correction paths.

- Second, we propose a source-aware correction workflow that employs signal-specific challengers together with verification-controlled fusion, so that heterogeneous recommendation signals can revise the current proposal without being collapsed into a single undifferentiated reasoning process.

- Third, experiments across sequential recommendation benchmarks show the effectiveness of the proposed framework and suggest that instance-wise specialist intervention is a promising direction for agentic recommendation. Our anonymized code is available at: `https://anonymous.4open.science/r/which-is-better-41E1/`

2 Related Work

Single-path recommenders from sequential recommendation to LLM4Rec. Sequential recommendation has largely been built on interaction IDs and latent factors, with models such as SASRec and BERT4Rec becoming representative paradigms for next-item prediction from behavior sequences (Kang and McAuley, 2018; Sun et al., 2019). Recent LLM4Rec studies extend this line along several routes, including prompt-based semantic enrichment (Lyu et al., 2024; Xu et al., 2024), hybrid alignment between behavioral and textual signals (Liao et al., 2024; Zhu et al., 2024; Chen et al., 2024), generative recommendation and retrieval (Geng et al., 2022; Rajput et al., 2023; Ji et al., 2024; Zheng et al., 2024; Yue et al., 2023; Kim et al., 2025; Yang et al., 2025), and broader taxonomies that summarize these transitions (Wu et al., 2024; Zhang et al., 2025a). P5 reformulates recommendation as a unified text-to-text problem (Geng et al., 2022), LC-Rec adapts LLMs by integrating collaborative semantics into generation (Zheng et al., 2024), HLLM strengthens sequential recommendation with hierarchical item-user modeling (Chen et al., 2024), and ItemRAG introduces item-based retrieval augmentation for LLM recommenders (Kim et al., 2025). The newer generative

branch, including stronger large-scale generative recommendation models and dedicated surveys, expands the semantic capabilities of this family (Yang et al., 2025). These approaches improve semantic grounding and broaden LLM applications in recommendations, from generating explanations to multi-step reasoning over user histories. However, most still follow a single ranking, generation, or retrieval path per instance, limiting selective intervention when user histories contain conflicting interests or multiple signals need different weighting. This restricts the model's ability to balance competing objectives in complex scenarios.

Intervention, reranking, and agentic control on top of a base recommender. A second nearby line studies how LLMs should intervene on top of an existing recommendation pipeline rather than replacing it wholesale. ToolRec frames recommendation through tool learning and explicit tool invocation (Zhao et al., 2024), learnable-planner approaches show that LLMs can act as recommendation planners rather than static scorers (Shi et al., 2024), and reranking-oriented work such as LLM4Rerank suggests that post-hoc corrective control can be effective when the base model already proposes reasonable candidates (Gao et al., 2025). This intervention view is especially relevant to our setting because we also operate on top of a strong base recommender, but our target is narrower: we focus on *selective correction*, namely deciding when a challenger path should activate, which challenger should intervene, and how challenger evidence should be fused back into the current proposal. In other words, we are not building the richest universal agent chain; we are studying recommendation-native correction under mixed preferences and heterogeneous signals.

General recommendation agents and memory-aware agentic recommenders. A growing line of work further casts recommendation as an agentic process. Early systems such as InteRecAgent and RecMind endow LLMs with memory, planning, reflection, and tool use to interact with external recommenders and organize multi-step decision making (Wang et al., 2024a; Huang et al., 2025c). Subsequent work broadens this direction through multi-agent collaboration, explicit tool orchestration, deliberate reasoning, and learned routing, as exemplified by Agent4Rec, MACRS, MACRec, STARec, AgentRec, and ChainRec (Zhang et al., 2024; Fang et al., 2024; Wang et al., 2024b; Wu et al., 2025; Ma et al., 2025; Li et al., 2026). Recent surveys and perspective papers consolidate these developments into the broader paradigm of agentic recommender systems (Peng et al., 2025; Huang et al., 2025a; Zhang et al., 2025b). Another nearby line studies stronger memory mechanisms inside agentic recommenders. MR.Rec combines memory and reasoning inside an LLM recommendation assistant (Huang et al., 2025b), while AMEM4Rec leverages cross-user similarity for memory evolution in agentic recommenders (Nguyen et al., 2026). These works demonstrate that recommendation agents benefit from richer memory structures, rather than relying solely on richer prompting techniques. However, their emphasis largely remains on general memory construction or broad workflow routing, without addressing fine-grained, signal-specific correction or targeted intervention. By contrast, our paper focuses on a narrower correction boundary: case-conditioned routing plus signal-specific challengers, with verification-controlled fusion carefully regulating how those challengers revise the current proposal.

3 Methods

Figure 2 provides an end-to-end overview of our selective recommendation correction framework. Starting from the user context and candidate set, an LLM-based base recommender first produces an initial proposal. As illustrated in the figure, this proposal is not revised uniformly. Instead, a memory-aware router summarizes the current instance through preference coherence, memory conflict, and candidate ambiguity, and decides whether the case should remain on a lightweight stable path or be routed to a sparse set of signal-specific challengers for deeper correction. These challengers revisit the proposal from complementary perspectives, including memory-sensitive, conflict/affinity, and candidate-sensitive views, while keeping heterogeneous evidence separate. A verification-controlled fusion module then assigns per-challenger correction strengths and integrates evidence that is compatible with the current recommendation context, yielding the refined ranking. In this way, the framework preserves the efficiency of near-identity refinement on stable cases, while enabling deeper, source-aware correction when user preference signals are mixed or ambiguous.

Viewed in this way, our framework does not regenerate recommendations from scratch; it selec-

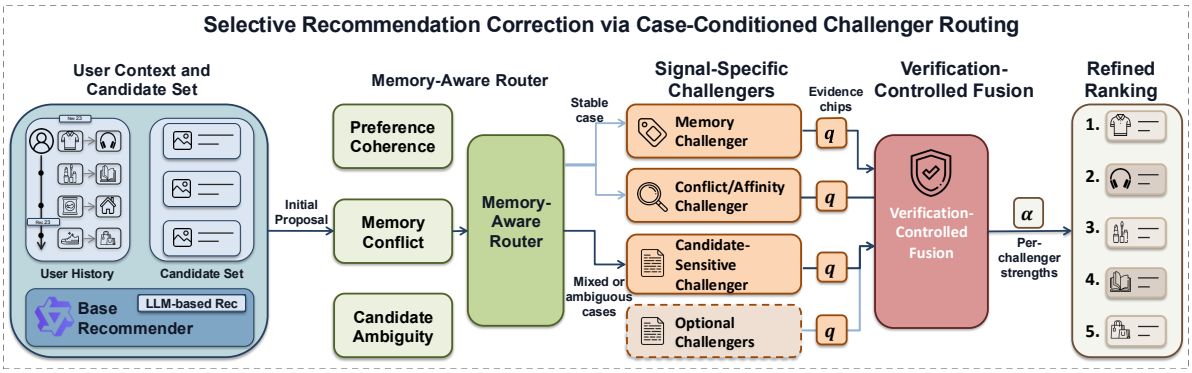

Figure 2: Overview of the proposed selective recommendation correction framework. An LLM-based base recommender first generates an initial proposal from the user history and candidate set. A memory-aware router then determines whether the current instance is stable or requires deeper intervention, activating only a sparse set of signal-specific challengers when necessary. Their evidence is moderated by verification-controlled fusion to produce the refined ranking, keeping stable cases lightweight while enabling deeper correction for ambiguous cases.

tively intervenes on top of a strong initializer. The remainder of this section formalizes the four stages highlighted in Figure 2: initial proposal, case-conditioned routing, signal-specific challenger intervention, and verification-controlled fusion.

In stable situations, this workflow preserves the efficiency of the initial proposal through near-identity refinement; in ambiguous or difficult cases, it applies deeper, source-aware corrections. Overall, the system remains fully both case-aware and source-aware, adapting the correction depth to the user state while keeping heterogeneous signals separate until verified.

3.1 Problem Setup, Notation, and Recommendation Intervention Setting

For a user u, let $H_u = [i_1, \ldots, i_T]$ denote the observed interaction history, and let $C_u = \{c_1, \ldots, c_N\}$ denote the candidate set considered at test time. A well-trained proposal initializer constructs an initial recommendation score over candidates:

$$s_u^{(0)}(c) = f_0(H_u, c), \qquad c \in C_u,$$

where f_0 denotes a well-trained base recommender that produces the initial user-aware proposal. The corresponding top-ranked prediction is

$$\hat{c}_u^{(0)} = \arg\max_{c \in C_u} s_u^{(0)}(c).$$

Rather than treating recommendation as a one-shot score assignment, we allow the proposal to enter different reflective fusion depths according to the current user state. Accordingly, our framework produces a refined score

$$s_u^*(c) = s_u^{(0)}(c) + \Delta_u(c),$$

and the final recommendation is

$$\hat{c}_u^* = \arg\max_{c \in C_u} s_u^*(c).$$

The central question is therefore not how to generate a new item from scratch, but how much challenger evidence should be fused into the current proposal under each recommendation context.

To keep the method rigorous but readable, we use only a small set of symbols throughout this section. We reserve H_u for user history, C_u for the candidate set, $s_u^{(0)}$ and s_u^* for the initial and final scores, M_u for the memory-aware routing state, $\mathcal{K}_u$ for the activated challenger set, $E_{u,k}$ for challenger evidence, and $\alpha_{u,k}$ for verification-controlled fusion strength. These symbols are reused consistently in the remainder of the section.

3.2 Case-Conditioned Challenger Routing

The first challenge identified in Section 1 is **Mixed Preference Reasoning**: a single interaction history may simultaneously contain persistent tastes, recent drifts, transient curiosities, and even conflicting evidence. Collapsing all of these into a single uniform reasoning pattern may lead to overly confident yet brittle recommendation decisions. We specifically address this issue through **case-conditioned challenger routing**, whose routing state is explicitly memory-aware.

Concretely, we derive a routing state

$$M_u = (\rho_u, \kappa_u, \eta_u),$$

where ρ_u summarizes preference concentration in the user's history, κ_u captures the intensity of memory conflict between supportive and contradictory

evidence, and η_u measures ambiguity among the current leading candidates. These three components are not intended as an exhaustive psychological model of preference, but as a compact operational state that tells the system whether the current instance resembles a stable recommendation regime or a mixed, conflict-sensitive one.

Based on M_u, the router predicts a routing decision

$$z_u = g(M_u),$$

and activates a sparse challenger set

$$\mathcal{K}_u = \mathcal{R}(z_u).$$

Intuitively, the router distinguishes between at least three representative regimes. In a *stable regime*, the user's recent and long-term signals are coherent, candidate ambiguity is low, and only a lightweight correction path is needed. In a *mixed-preference regime*, the history indicates competing interests or unresolved recency tension, so memory-sensitive challenger intervention becomes necessary. In a *hard-case regime*, the leading candidates are highly similar or near-positive lookalikes, so the workflow activates more cautious specialists to avoid premature suppression of promising items.

This routing mechanism is deliberately selective. It does not seek to make every sample look difficult, nor does it rely on a crude hard/easy threshold. Instead, it performs instance-adaptive challenger activation conditioned on the user's memory state. This is the key to making the framework recommendation-native: stable instances receive near-identity refinement, while difficult instances receive a more deliberate challenger path.

3.3 Signal-Specific Challengers

The second challenge in Section 1 is **Heterogeneous Signal Integration**. In our recommendation setting, the available evidence is naturally multi-source: user history, structured memory traces, conflict cues, candidate affinity, and other recommendation-side diagnostics do not always play the same semantic role. Instead of collapsing them into one shared prompt or one undifferentiated scorer, we organize them as a small family of **signal-specific challengers**.

For each activated challenger $k \in \mathcal{K}_u$, let $X_{u,k}$ denote the signal channel that the challenger is allowed to access. The challenger then produces an evidence packet

$$E_{u,k} = \left(\Delta_{u,k}, q_{u,k}\right) = \Psi_k\Big(H_u, C_u, X_{u,k}, s_u^{(0)}\Big),$$

where $\Delta_{u,k}(c)$ is a recommendation-specific revision cue over candidates and $q_{u,k}$ is the challenger's confidence in this evidence. Importantly, each challenger is source-aware: it reasons over its own channel before the framework decides how different signals should interact.

We instantiate this challenger layer with a small number of specialists rather than a sprawling agent society. Some challengers are *tool-style* specialists: they issue lightweight evidence calls over explicit recommendation signals such as conflict calibration, affinity checks, or candidate-level consistency diagnostics. Others are *agent-style* specialists: they revisit the current recommendation through deeper reflection over user memory and candidate semantics when the router detects a mixed-preference or hard-case state. In our current instantiation, this challenger family is operationalized primarily through memory-sensitive and candidate-sensitive paths, allowing different sources of recommendation evidence to challenge the current proposal from different perspectives.

This abstraction is important for two reasons. First, it allows simple yet useful signals to appear as tools instead of being overburdened with unnecessary agent complexity. Second, it allows deeper contextual reasoning to be introduced when recommendation ambiguity genuinely requires it. Both forms are consistently treated as challenger resources selected by the router, and both remain source-aware until verification time.

3.4 Verification-Controlled Fusion

Challenger evidence is only useful when it is fully compatible with the current recommendation context. For this reason, we do not fuse challenger outputs directly. Instead, we introduce **verification-controlled fusion**, which systematically evaluates how the collected evidence should contribute to the evolving recommendation proposal.

Given the routing state M_u, the initial score $s_u^{(0)}$, and the challenger evidence packets $\{E_{u,k}\}_{k\in\mathcal{K}_u}$, verification-controlled fusion assigns a fusion strength

$$\alpha_{u,k} = V\Big(M_u, E_{u,k}, s_u^{(0)}\Big) \in [0,1]$$

to each challenger. The final refined score is then computed as

$$s_u^*(c) = s_u^{(0)}(c) + \sum_{k\in\mathcal{K}_u} \alpha_{u,k}\,\Delta_{u,k}(c).$$

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

Dataset	#Users	#Items	#Interactions
Beauty	4,548	28,344	20,326
Movies&TV	20,000	164,203	157,346
Toys&Games	50,000	349,613	277,432
Yelp	50,000	149,659	488,995

This formulation makes the fusion explicitly adaptive. When challenger evidence is weak, redundant, or inconsistent with the current memory state, the corresponding $\alpha_{u,k}$ remains small and the final score stays close to an identity-level update. When the evidence is coherent and sufficiently informative, the framework applies a stronger revision to the current ranking.

The role of verification-controlled fusion is to provide a conservative control layer for selective correction. It is designed to keep low-quality challenger signals from disturbing already stable cases, to moderate conflicting challenger outputs, and to allow stronger revision only when the current user state justifies it. In practice, this is what turns a collection of source-aware challengers into a compatibility-aware correction process.

Finally, in implementation we calibrate the routing and verification behavior on validation data to maintain sparse activation and stable intervention behavior. This calibration does not change the overall formulation above; it simply ensures that lightweight cases remain lightweight, while difficult cases receive deeper challenger analysis when the recommendation context truly demands it.

4 Experiments

4.1 Experimental Setup

Datasets and Preprocessing. We evaluate on four sequential recommendation benchmarks: Amazon[1] Beauty, Movies&TV, Toys&Games, and Yelp[2]. The statistics reported here describe the processed datasets used in this paper rather than the raw corpora. For three 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 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. 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 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 HR@1 in recommendation tasks, the fraction of users whose held-out next item is ranked first), 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 current **Generative** rows 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 (Recommender AI 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 recommendation workflows.

Our Variants. The main comparison reports two variants of our framework, instantiated with Qwen3.5-2B[3]and Qwen3.5-4B[4] . The ablation and route analyses specifically focus on the Qwen3.5-4B variant, which consistently gives the strongest overall results in the main comparison. For methods that require an LLM backbone, we use a shared

[1] https://amazon-reviews-2023.github.io/

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

[3] https://huggingface.co/Qwen/Qwen3.5-2B

[4] https://huggingface.co/Qwen/Qwen3.5-4B

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

Model Class	Model	Beauty			Movies&TV			Toys&Games			Yelp		
		Pass@1	HR@5	NDCG@5	Pass@1	HR@5	NDCG@5	Pass@1	HR@5	NDCG@5	Pass@1	HR@5	NDCG@5
Traditional	SASRec	0.0922	0.1227	0.1091	0.0068	0.0267	0.0167	0.0041	0.0085	0.0064	0.0035	0.0121	0.0078
	BSARec	0.1182	0.1329	0.1266	0.0106	0.0161	0.0135	0.0086	0.0114	0.0101	0.0027	0.0091	0.0059
	FMLP-Rec	0.1192	0.1376	0.1292	0.0114	0.0236	0.0179	0.0086	0.0149	0.0120	0.0042	0.0115	0.0079
	Mamba4Rec	0.1094	0.1282	0.1197	0.0115	0.0228	0.0176	0.0081	0.0140	0.0113	0.0041	0.0094	0.0068
	HSTU	0.1199	0.1372	0.1294	0.0122	0.0195	0.0161	0.0079	0.0136	0.0103	0.0030	0.0114	0.0072
Generative	TIGER	0.1231	0.1385	0.1291	0.0119	0.0201	0.0182	0.0085	0.0150	0.0122	0.0086	0.0132	0.0108
	OneRec	0.1302	0.1698	0.1567	0.0201	0.0278	0.0241	0.0187	0.0249	0.0211	0.0103	0.0181	0.0149
LLM4Rec	TallRec	0.1234	0.1533	0.1400	0.0123	0.0286	0.0207	0.0121	0.0204	0.0164	0.0064	0.0082	0.0073
	HLLM	0.1462	0.1708	0.1487	0.0256	0.0448	0.0349	0.0179	0.0317	0.0280	0.0117	0.0195	0.0152
	ItemRAG	0.1306	0.1656	0.1467	0.0209	0.0454	0.0284	0.0148	0.0314	0.0235	0.0127	0.0176	0.0155
Agent4Rec	Rec Agent	0.1067	0.1345	0.1221	0.0096	0.0166	0.0133	0.0038	0.0131	0.0085	0.0028	0.0105	0.0067
	STARec	0.0240	0.0574	0.0411	0.0105	0.0190	0.0152	0.0076	0.0108	0.0094	0.0061	0.0135	0.0099
	AgentRec	0.0143	0.0350	0.0250	0.0104	0.0185	0.0149	0.0072	0.0104	0.0090	0.0044	0.0129	0.0088
	ToolRec	0.0055	0.0132	0.0094	0.0096	0.0166	0.0133	0.0042	0.0084	0.0064	0.0028	0.0105	0.0067
Ours	Qwen3.5-2B	0.1482	0.1715	0.1609	**0.0291**	0.0432	**0.0363**	0.0177	0.0222	0.0200	0.0129	0.0221	0.0184
	Qwen3.5-4B	**0.1669**	**0.1765**	**0.1721**	0.0221	**0.0486**	0.0358	**0.0208**	**0.0353**	**0.0283**	**0.0132**	**0.0289**	**0.0217**

Qwen3.5-4B backbone to reduce variance caused by foundation-model choice and to focus the comparison primarily on recommendation workflow design rather than backbone selection. Classical non-LLM recommenders retain their original architectures. We additionally report the Qwen3.5-2B variant as a compact sensitivity study to further validate the robustness of our framework.

4.2 Performance Comparison

Table 2 compares our framework against **Traditional**, **Generative**, **LLM4Rec**, and **Agent4Rec** baselines under a unified next-item prediction protocol. We report Pass@1 together with HR@5 and NDCG@5 metrics, so that the comparison more comprehensively reflects both top-ranked correction quality and overall shortlist ranking quality.

From the table, three main observations can clearly be drawn regarding the comparative performance of the methods.

(1) Our framework delivers the strongest overall performance in Table 2. Across the two backbone instantiations, it attains the best reported value in every metric column, with Qwen3.5-4B being the most consistently strong row overall. This pattern appears in both Pass@1 and cutoff-5 metrics, suggesting that the gains are not limited to a single notion of ranking quality.

(2) The strongest external competition in Table 2 comes from the semantically stronger **Generative** and **LLM4Rec** baselines, especially OneRec (Deng et al., 2025), HLLM (Chen et al., 2024), and ItemRAG (Kim et al., 2025), rather than from the traditional or generic agentic rows. Since all backbone-dependent methods in the aligned comparison use the same Qwen3.5-4B model, this result should be read primarily as a workflow-level comparison; even under this conservative shared-backbone setting, our framework remains strongest among the aligned agentic-style alternatives.

(3) The gains are generally stable but not identical across backbones. Qwen3.5-4B is the most reliable variant overall, while Qwen3.5-2B remains competitive and achieves the best Movies&TV Pass@1 and NDCG@5. This suggests that the framework does not depend on a single winning backbone choice and can remain effective in a more compact instantiation.

4.3 Ablation Study

Table 3 reports a standard incremental ablation on the two datasets with the cleanest aligned intermediate stages, namely Movies&TV and Toys&Games on the Qwen3.5-4B line. For readability, we abbreviate the cumulative settings as *Base*, *Base+Ctrl*, *Base+Ctrl+Route*, and *Base+Ctrl+Route+Verify*, where *Ctrl* denotes lightweight control, *Route* denotes case-conditioned challenger routing, and *Verify* denotes verification-controlled fusion.

The aligned ablation on Movies&TV and Toys&Games shows that the largest improvement appears when case-conditioned challenger routing is introduced on top of the base and lightweight-control path. In other words, the main benefit does not come from uniformly adding more processing stages, but from concentrating the gain in the routed

Table 3: Ablation study on the Qwen3.5-4B line. We progressively add lightweight control (*Ctrl*), case-conditioned challenger routing (*Route*), and verification-controlled fusion (*Verify*). 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	Movies&TV			Toys&Games		
	Pass@1	HR@5	NDCG@5	Pass@1	HR@5	NDCG@5
Base	0.01405	0.03300	0.02391	0.01332	0.02212	0.01796
Base+Ctrl	0.01240	0.03975	0.02584	0.01442	0.03138	0.02318
Base+Ctrl+Route	0.02095	0.04780	0.03487	0.02064	0.03484	0.02803
Base+Ctrl+Route+Verify	**0.02205**	**0.04865**	**0.03581**	**0.02076**	**0.03530**	**0.02829**

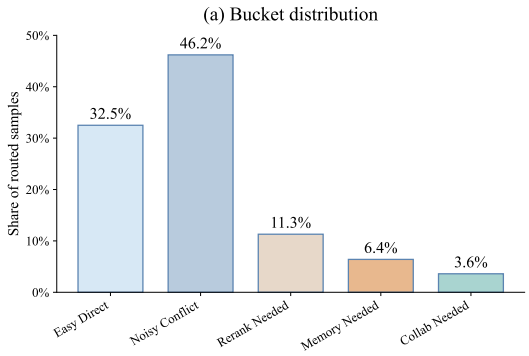

Figure 3: The routing analysis summarizes the aggregate distribution of routed traffic across different routing buckets, illustrating how often each routing choice is activated during the recommendation workflow.

correction stage. This is consistent with the paper's central claim that selective intervention, rather than a fixed deeper pipeline, is the main driver of the framework's overall gains.

Adding verification-controlled fusion after routing yields smaller but consistently positive gains in the ablation. We therefore interpret the Verify stage as a conservative refinement layer whose contribution is incremental relative to routing, rather than the primary source of improvement. Under the current aligned settings, most of the gain comes from routed correction, while verification-controlled fusion provides a modest final-stage benefit.

4.4 Route Analysis

We analyze routing on the Qwen3.5-4B on Movies&TV dataset by examining how samples are distributed across the predefined routing buckets in Figure 3. Here, `easy_direct` denotes straightforward cases that can usually be handled without substantial correction; `noisy_conflict` denotes cases with conflicting or noisy signals that require additional resolution; `rerank_needed` denotes cases where the candidate set is reasonable but the ranking still needs adjustment; `memory_needed` denotes cases that require richer contextual or historical information; and `collab_needed` denotes harder cases that benefit from more specialized multi-step or collaborative handling. As shown in the figure, most routed samples fall into `noisy_conflict` and `easy_direct`, while the more specialized buckets appear much less frequently. This pattern suggests that the routing mechanism is not allocating effort uniformly: it focuses primarily on direct cases and noisy preference conflicts, and invokes deeper interventions only for a smaller subset of harder samples. The distribution is consistent with our main finding that adaptive routing matches intervention depth to case difficulty, which helps improve recommendation quality without introducing unnecessary complexity for every individual instance.A detailed analysis of the routing behavior is provided in the Appendix A.2.

5 Conclusion

We study recommendation refinement as a selective correction problem rather than a uniform reasoning chain. Starting from a strong initial proposal, our framework uses case-conditioned challenger routing to trigger deeper intervention only when needed, and applies source-aware challengers with verification-controlled fusion to conservatively revise rankings under mixed or ambiguous contexts. Across traditional, generative, LLM4Rec, and agentic baselines, the framework is highly competitive under the shared protocol, with the strongest overall results achieved by the Qwen3.5-4B variant. Ablation results show that case-conditioned challenger routing is the main source of gains, while verification-controlled fusion serves as a conservative refinement layer. Route depth also varies across datasets, suggesting that recommendation agents benefit from selectively deciding when and how to intervene, rather than applying the same reasoning routine to every case.

Limitations

Our current study has two main limitations. First, because of limited computational resources, we have not yet extended the evaluation to larger-scale LLM backbones. While the present results already show that the proposed framework is effective in the current settings, broader validation on stronger backbones would help clarify its scalability and generality. Second, the current experiments do not yet cover more complex multimodal recommendation scenarios. As recommendation inputs in practice often combine textual, visual, and other heterogeneous signals, future work should further investigate how the framework behaves in richer multimodal environments.

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*.

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*.

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*.

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.

Wentao Shi, Xiangnan He, Yang Zhang, Chongming Gao, Xinyue Li, Jizhi Zhang, Qifan Wang, and Fuli Feng. 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*, 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.

Yancheng Wang, Ziyan Jiang, Zheng Chen, Fan Yang, Yingxue Zhou, Eunah Cho, Xing Fan, Yanbin Lu, Xiaojiang Huang, and Yingzhen Yang. 2024a. 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. 2024b. 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. 2025a. Cold-start recommendation towards the era of large language models (llms): A comprehensive survey and roadmap. *arXiv preprint arXiv:2501.01945*.

Yu Zhang, Shutong Qiao, Jiaqi Zhang, Tzu-Heng Lin, Chen Gao, and Yong Li. 2025b. A survey of large language model empowered agents for recommendation and search: Towards next-generation information retrieval. *arXiv preprint arXiv:2503.05659*.

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 Experiment Details

A.1.1 Evaluation Protocol

We follow a unified next-item prediction protocol on Beauty, Movies&TV, Toys&Games, and Yelp. 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 (i.e., HR@1, the fraction of users whose held-out next item is ranked first), together with HR@5 and NDCG@5, while the broader evaluation suite still uses standard cutoff-based HR/NDCG analysis.

A.1.2 Benchmark Layout

The main comparison table is organized around four baseline families: **Traditional**, **Generative**, **LLM4Rec**, and **Agent4Rec**, together with an **Ours** block. All reported rows follow the same evaluation protocol described in this paper. The Generative block includes TIGER and OneRec, and the LLM4Rec block includes TallRec, HLLM, and ItemRAG. The main comparison reports two variants of our method, based on Qwen3.5-2B and Qwen3.5-4B, and the supporting ablation and route analyses are reported on the Qwen3.5-4B variant.

A.1.3 Workflow Variant Naming

The ablation and route analyses use several short labels for compact presentation. In the ablation table, `Ctrl`, `Route`, and `Verify` denote lightweight control, case-conditioned challenger routing, and verification-controlled fusion. In the route figure, `Base` denotes the proposal-only path, `Fixed Fusion` denotes a conservative fixed-fusion path, `Adaptive Memory` denotes a memory-sensitive route, `Verifier` denotes a verified stopping path, `Bucket Specialist` denotes a bucket-specific specialist route, and `Predicted Main` denotes the primary routed challenger path.

A.1.4 Implementation Details

We implement the selective correction workflow as a collection of lightweight learned control modules on top of cached recommendation signals, rather than retraining the full recommendation backbone end-to-end for each intervention stage. Unless otherwise noted, learned control modules are trained with AdamW, learning rate 10^{-3}, weight decay 10^{-4}, dropout 0.1, random seed 42, and validation-driven early stopping.

For case-conditioned route correction, we use batch size 256, hidden dimension 64, up to 30 epochs, early-stop patience 6, `delta_scale` $= 0.75$, `route_temperature` $= 1.0$, and `selector_temperature` $= 1.0$, with a temperature-soft variant using `selector_temperature` $= 1.8$. For dynamic fusion-strength prediction, we use batch size 256, hidden dimension 128, up to 24 epochs, early-stop patience 5, a fixed control anchor $\alpha = 0.05$, and the evaluated grid $\{0.0, 0.02, 0.05, 0.08, 0.12, 0.15\}$. For source selection, we use batch size 256, hidden dimension 128, up to 30 epochs, early-stop patience 6, fixed $\alpha = 0.05$, `label_strategy=best_positive_proxy`, and `threshold_mode=bucket_margin`. For the workflow-margin verifier, we use batch size 256, hidden dimension 128, up to 30 epochs, early-stop patience 6, fixed $\alpha = 0.05$, margin loss, and `harm_penalty=4.0`. For the top-3 workflow challenger pool, we use batch size 512, hidden dimension 128, up to 30 epochs, early-stop patience 6, fixed $\alpha = 0.05$, and `top_k=3` under `workflow_only` pooling.

We did not perform an exhaustive continuous hyperparameter search. Instead, the current draft reports the small set of discrete configurations that survived branch-local validation and were then kept fixed for the corresponding test evaluations. All jobs were managed on a shared GPU cluster with at most four GPUs allocated concurrently.

A.1.5 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.

A.2 Route Buckets and Qualitative Analysis

A.2.1 Case Study of Routing Behavior

To make the routing behavior more interpretable, Figure 4 visualizes four representative archetypes for the non-residual buckets. The figure is intentionally schematic: it does not claim to reproduce a single observed user trace, but instead summarizes the characteristic routing logic behind each bucket name. `easy_direct` corresponds to a stable local regime where light verification is enough to preserve the original top-1. `noisy_conflict` highlights mixed short-term signals, where verification should suppress overreaction to a weak challenger. `rerank_needed` captures cases in which several candidates are plausible and a lightweight challenger only needs to repair the local order. `memory_needed` represents the long-range preference regime, where a deeper memory specialist becomes useful precisely because the shorter context is not sufficient. We omit the tiny `collab_needed` bucket from this figure because its support is currently too small to sustain a stable visual archetype.

A.2.2 Bucket Semantics

The routing analysis currently uses six bucket labels. `easy_direct` corresponds to cases whose current proposal is already locally stable and only needs light verification. `noisy_conflict` contains short-context conflicts where challenger evidence should be handled conservatively. `rerank_needed` captures cases with several plausible candidates that mainly require local order repair. `memory_needed` contains cases where longer-range preference evidence becomes more informative than the most recent actions. `collab_needed` is reserved for the rare cases that justify heavier multi-source intervention. Finally, `other` is a residual routed bucket used for traffic that does not cleanly fit the more interpretable non-residual categories.

A.2.3 Qualitative Route Archetypes

The route archetype figure in the main paper is intentionally schematic. It is not presented as a traced user-level case study or as a logged route-selection summary. Its role is to clarify the typical routing logic associated with the most interpretable non-residual buckets. In particular, it shows how the workflow keeps stable cases light, handles noisy conflicts conservatively, repairs small ranking mistakes through local challengers, and activates memory-aware revision only when shorter-context evidence is insufficient.

We omit the `collab_needed` bucket from the qualitative archetype figure because its current support is too small to sustain a stable visual template. The residual `other` bucket is also excluded from the qualitative figure because it functions as a broad catch-all category rather than a semantically clean case family.

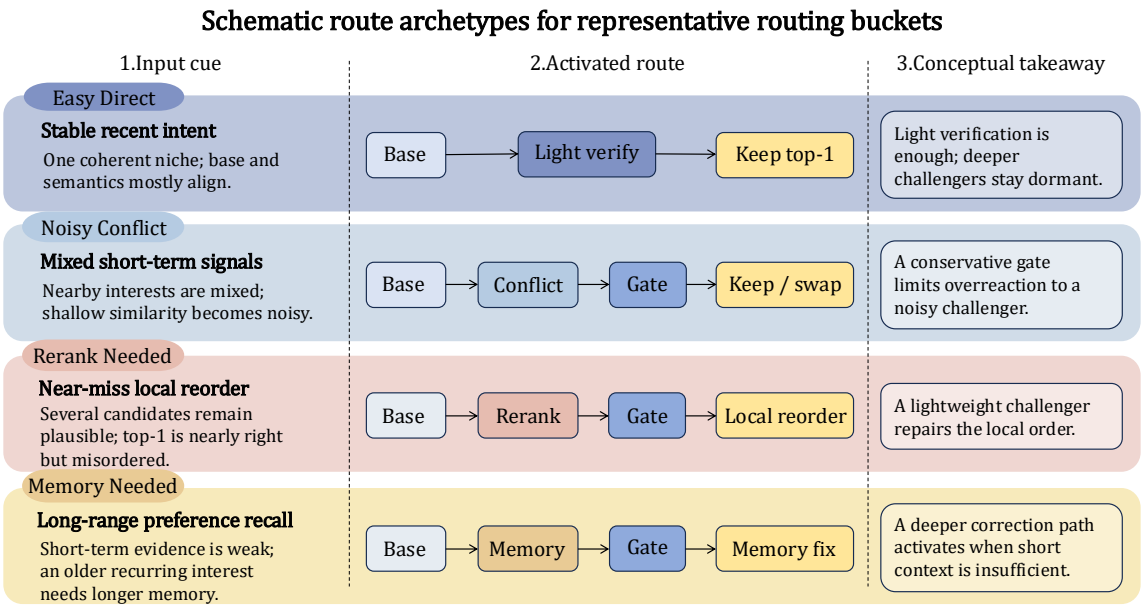

Figure 4: Schematic route archetypes used as an explanatory cartoon for the named non-residual buckets. The panels are not traced user-level cases and should be read only as conceptual illustrations of the corresponding correction patterns.

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.