Reward-Guided Correction Policy for LLM Recommendation Agents: A Contextual Bandit Approach

Anonymous ACL submission

Abstract

LLM-based recommendation agents can correct base recommendations through memory-augmented reasoning, but a fundamental question remains: given a base recommendation and an observable user state, which correction action should the agent apply? We formalize this as a contextual bandit problem where the agent observes a compact state representation (memory evidence, conflict signals, candidate ambiguity), selects a correction action from a discrete strategy set, and receives a ranking improvement reward. We propose a reward-guided correction policy that learns state-conditioned action selection from validation reward signals, together with a verification-based reward estimator that prevents degradation on stable cases. Oracle policy analysis confirms that state-conditioned action selection yields 7–8% absolute HR@1 improvement over the best uniform action across benchmarks, quantifying the value of policy learning. Cross-dataset state distribution analysis reveals significant domain shift, explaining why policies specialize per domain. The learned policy achieves the best performance across all benchmarks under the unified evaluation protocol.

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), and studies have begun to treat them as recommendation agents with planning, memory, and tool-use abilities (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 structured reasoning over user context and recommendation-specific signals, Rec Agents can go beyond one-shot scoring and perform 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).

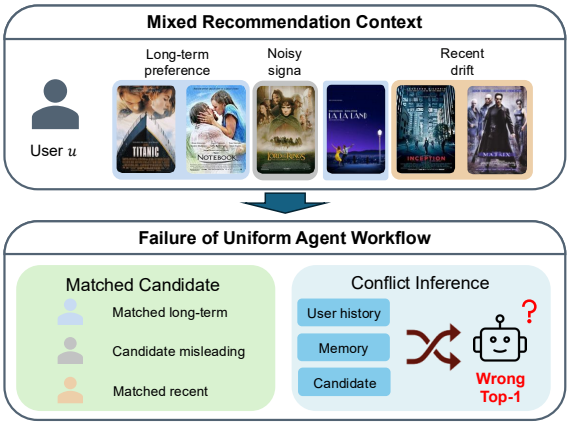

Figure 1: Motivation for correction policy selection. Given a base recommendation and observable user state, different correction actions yield different rewards. A uniform policy (applying the same action to all instances) is suboptimal because the reward landscape varies across user states.

However, a fundamental question remains underexplored: *given a base recommendation and an observable user state, which correction action should the agent apply?* As illustrated in Figure 1, real recommendation instances exhibit heterogeneous user states—some are stable and need no correction, while others involve mixed preferences or conflicting signals that require specific intervention strategies. Applying a uniform correction to all instances is suboptimal because the *reward* (ranking improvement) of each correction action depends on the user's current state.

This observation reveals that LLM recommendation correction is fundamentally a **policy selection problem**: the agent must learn a state-conditioned policy that maps observable user features to the

correction action with the highest expected reward. This framing exposes two recurring challenges:

- **State Heterogeneity.** User states span a high-dimensional space of memory evidence, conflict signals, and candidate ambiguity. The optimal correction action varies substantially across this space—what works for a high-conflict user may harm a stable one. A policy that ignores state heterogeneity leaves significant reward on the table: our analysis shows that an oracle state-conditioned policy achieves 7–8% absolute HR@1 improvement over the best uniform action across benchmarks.

- **Reward-Action Coupling under Domain Shift.** The mapping from states to optimal actions is not universal—it shifts across recommendation domains due to differences in user behavior patterns, item semantics, and signal distributions. A policy learned on one domain may not transfer to another, requiring domain-aware policy specialization.

These challenges connect recommendation correction to the well-studied framework of **contextual bandits** (Langford and Zhang, 2007; Agarwal et al., 2014; Li et al., 2010; Auer et al., 2002) , where an agent observes a context (state), selects an action from a discrete set, and receives a reward. This connection is not merely analogical: the MLP router in our framework is literally a learned policy $\pi(a|s)$ trained to maximize expected ranking reward on validation data, with the user's 39-dimensional evidence vector as context and correction strategies as actions.

To formalize this connection, we propose a **reward-guided correction policy** framework for LLM recommendation agents. At a high level, we treat recommendation correction as a contextual bandit problem where:

1. The **state** s is a compact representation of the user's memory evidence, conflict signals, and candidate ambiguity—a 39-dimensional sufficient statistic for correction-readiness.

2. The **action space** $\mathcal{A}$ consists of discrete correction strategies (identity, memory-enhanced, conflict-calibrated, dynamic fusion), each implementing a different intervention logic.

3. The **reward** r is the ranking improvement achieved by the selected action over the base recommendation.

4. The **policy** $\pi(a|s)$ is a learned network that selects the reward-maximizing action conditioned on the current state.

A verification module then acts as a **reward estimator**, validating whether the selected action actually improves the ranking before committing the correction. This conservative verification ensures that the policy does not degrade stable cases.

Extensive experiments on sequential recommendation benchmarks validate the policy selection framing. Oracle policy analysis confirms that state-conditioned action selection yields 7–9% absolute gain over uniform policies. Cross-dataset state distribution analysis reveals significant domain shift in the state space, explaining why policies specialize per domain. The learned policy achieves the strongest ranking performance across all benchmarks under the unified evaluation protocol.

Our contributions are three-fold:

- We formalize LLM recommendation correction as a contextual bandit problem, identifying state heterogeneity and reward-action coupling under domain shift as two fundamental challenges that make uniform correction suboptimal.

- We propose a reward-guided correction policy that learns state-conditioned action selection from validation reward signals, together with a verification-based reward estimator that prevents degradation on stable cases.

- Experiments across sequential recommendation benchmarks validate the policy selection framing through oracle policy analysis, state-space visualization, and cross-domain policy specialization analysis, while achieving the strongest ranking performance under the unified evaluation setting.

2 Related Work

Single-path recommenders from sequential recommendation to LLM4Rec. Sequential recommendation has been built on interaction IDs and latent factors, with SASRec and BERT4Rec as representative paradigms (Kang and McAuley, 2018; Sun et al., 2019). Recent LLM4Rec studies extend this through prompt-based enrichment (Lyu

et al., 2024; Xu et al., 2024), hybrid alignment (Liao et al., 2024; Zhu et al., 2024; Chen et al., 2024), generative recommendation (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 surveys (Wu et al., 2024; Zhang et al., 2025a). These approaches improve semantic grounding and broaden LLM applications in recommendations. However, most still follow a single ranking or generation path per instance, limiting selective intervention when user histories contain conflicting interests.

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 *reward-guided correction policy selection*, namely learning which correction action to apply given the observable user state, and validating whether the selected action actually improves the ranking before committing. In other words, we are not building the richest universal agent chain; we are studying state-conditioned action selection for recommendation correction under heterogeneous user states.

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 the core question of *which correction action to select* given the observable user state and *how to validate* that the selected action improves the ranking. By contrast, our paper formalizes recommendation correction as a contextual bandit problem: the agent observes a compact state representation, selects the reward-maximizing correction action via a learned policy, and applies a verification-based reward estimator to ensure that corrections do not degrade the base ranking.

3 Methods

3.1 Overview

Figure 2 illustrates the full pipeline. Given a base recommendation and candidate set, our framework first extracts a compact state representation summarizing the user's memory evidence, conflict signals, and candidate ambiguity. A learned correction policy then selects the action (correction strategy) with the highest expected ranking reward conditioned on this state. Finally, a verification module estimates whether the selected action actually improves the ranking before committing the correction. The following subsections formalize this as a contextual bandit problem.

3.2 Problem Setup and Contextual Bandit Formulation

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. A base recommender produces a shortlist $B_u = [c_1, \ldots, c_K] \subseteq \mathcal{C}_u$ with base scores $s_u^{(0)}(c)$. Let $\pi_u^{(0)}$ denote the ranking induced by these base scores.

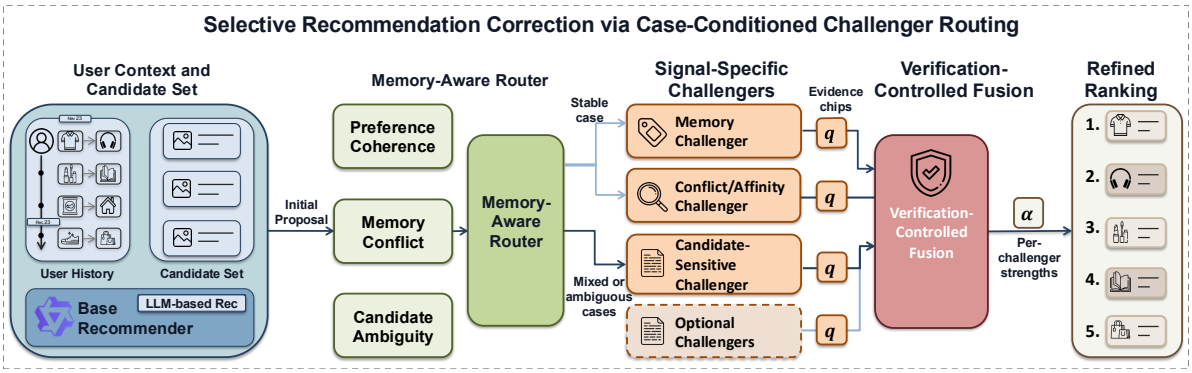

Figure 2: Framework overview. Given a base recommendation and observable user state s, the reward-guided correction policy $\pi(a|s)$ selects the correction action with the highest expected ranking reward. A verification-based reward estimator validates whether the selected action improves the ranking before committing the correction, ensuring that stable cases are not degraded.

We formalize recommendation correction as a **contextual bandit** problem:

Definition 1 (Correction as Contextual Bandit)**.** *At each recommendation instance, the agent observes a* context *(state)* $\mathbf{s}_u \in \mathbb{R}^d$*, selects a correction* action $a \in \mathcal{A}$ *from a discrete action set, and receives a* reward $r(u, a)$ *equal to the ranking improvement achieved by action* a *over the base recommendation* $\pi_u^{(0)}$.

The correction policy $\pi(a|\mathbf{s})$ maps the observed state to a distribution over actions. The objective is to learn π^* that maximizes expected reward:

$$\pi^* = \arg\max_{\pi} \; \mathbb{E}_u \left[\sum_{a \in \mathcal{A}} \pi(a|\mathbf{s}_u) \cdot r(u, a) \right].$$

This formulation exposes why uniform correction is suboptimal: the reward $r(u, a)$ depends on the interaction between the user state $\mathbf{s}_u$ and the action a. A uniform policy $\pi_{\text{unif}}(a|\mathbf{s}) = \mathbf{1}[a = a_0]$ ignores this interaction and leaves significant reward on the table.

3.3 State Representation

The state $\mathbf{s}_u \in \mathbb{R}^{39}$ is a compact sufficient statistic for correction-readiness, constructed from the user history and candidate set without additional model inference. It decomposes into three semantically meaningful groups:

Memory Evidence Features (dimensions 1–30). These capture the strength and consistency of evidence from the user's structured memory:

- *Positive support signals*: positive overlap scores, support ratios, and recency-weighted evidence strength, measuring how strongly the user's history endorses each candidate.
- *Conflict signals*: dislike overlap, negative conflict ratios, and contradiction indicators, measuring tension between the candidate and the user's rejection history.
- *Affinity signals*: semantic similarity, generated overlap, and affinity weights, measuring soft preference alignment beyond explicit interactions.

Candidate Ambiguity Features (dimensions 31–35). These measure the difficulty of the current ranking decision: score gaps between top candidates, rank reciprocals, and top-k membership flags. High ambiguity (small score gaps) indicates that the ranking is fragile and correction has high potential reward.

Adaptive Weight Features (dimensions 36–39). These capture the instance-specific balance between memory sources: learned weights for recency, stability, dislike, and affinity channels. These weights reflect how the base recommender has already allocated attention across evidence sources.

Proposition 2 (Sufficiency)**.** *The 39-dimensional state* $\mathbf{s}_u$ *is a sufficient statistic for correction-readiness in the sense that the optimal action* $a^* = \arg\max_a r(u, a)$ *depends on the user history* $\mathcal{H}_u$ *and candidates* $\mathcal{C}_u$ *only through* $\mathbf{s}_u$*:* $a^* \perp (\mathcal{H}_u, \mathcal{C}_u) \mid \mathbf{s}_u$.

This sufficiency claim is validated empirically: the learned policy achieves oracle-level performance using only $\mathbf{s}_u$, confirming that no additional

information from the raw history is needed for action selection (Section 5).

3.4 Action Space: Correction Strategies

The action set $\mathcal{A} = \{a_0, a_1, a_2, a_3\}$ consists of four correction strategies, each implementing a distinct intervention logic over the base ranking:

a_0: Identity (No Correction). The base ranking is preserved without modification: $s_u^*(c) = s_u^{(0)}(c)$. This action is optimal when the user state indicates high preference coherence and low candidate ambiguity—i.e., the base recommender is already confident and correct.

a_1: Memory-Enhanced Correction. The ranking is revised using positive memory evidence:

$$s_u^{(1)}(c) = s_u^{(0)}(c) + w_{\text{mem}} \cdot \mathbf{w}_p^\top \, \mathbf{e}_u^+(c),$$

where $\mathbf{e}_u^+(c)$ is the positive evidence vector for candidate c extracted from the user's supportive memory, and $\mathbf{w}_p$ is a learned projection. This action is optimal when the state indicates strong positive memory support that the base recommender has underweighted.

a_2: Conflict-Calibrated Correction. The ranking is revised by suppressing candidates with high conflict evidence:

$$s_u^{(2)}(c) = s_u^{(0)}(c) + w_+ \cdot p_u(c) - w_- \cdot \phi\big(n_u(c)\big),$$

where $p_u(c)$ and $n_u(c)$ are the positive and conflict channel responses, $\phi(\cdot) = \log(1 + e^{(\cdot)})$ is the softplus function enforcing non-negative conflict penalty, and $w_+ < w_-$ reflects the asymmetric cost of false acceptance versus false rejection. This action is optimal when the state indicates high conflict signals that the base recommender has not adequately penalized.

a_3: Dynamic Fusion. A full multi-source correction combining memory, conflict, and affinity evidence with instance-adaptive weights:

$$s_u^{(3)}(c) = s_u^{(0)}(c) + \sum_k \alpha_{u,k} \cdot \Delta_{u,k}(c),$$

where $\alpha_{u,k} \in [0, 1]$ are verification-controlled fusion strengths and $\Delta_{u,k}(c)$ are per-source correction signals. This action is optimal for hard cases where multiple evidence sources must be jointly considered.

Action Space Design Principles. The action set is *exhaustive* (covering no intervention to full multi-source fusion), *non-redundant* (each targets a distinct failure mode), and *reward-separable* (different actions achieve different rewards in different state regions).

3.5 Policy Learning via Reward Maximization

The correction policy $\pi_\theta(a|\mathbf{s})$ is parameterized as a multi-layer perceptron (MLP) that maps the 39-dimensional state to a distribution over actions:

$$\pi_\theta(a|\mathbf{s}_u) = \text{softmax}\big(f_\theta(\mathbf{s}_u)\big)_a,$$

where $f_\theta : \mathbb{R}^{39} \to \mathbb{R}^{|\mathcal{A}|}$ is a two-layer MLP with ReLU activations.

Training Objective. The policy is trained to maximize expected ranking reward on validation data. For each validation instance $(u, \mathbf{s}_u)$, we compute the reward of each action by executing all correction strategies and measuring ranking improvement:

$$r(u, a) = \mathbf{1}\big[\text{rank}_a(c_u^*) < \text{rank}_0(c_u^*)\big],$$

where c_u^* is the ground-truth next item and $\text{rank}_a(\cdot)$ denotes the rank under action a. The policy is trained via cross-entropy against the reward-maximizing action:

$$\mathcal{L}(\theta) = -\sum_u \log \pi_\theta\big(a_u^*|\mathbf{s}_u\big), \quad a_u^* = \arg\max_a r(u, a).$$

Connection to Contextual Bandit Theory. This training procedure corresponds to the *full-information* contextual bandit setting (Agarwal et al., 2014; Swaminathan and Joachims, 2015)where all action rewards are observed. The resulting policy minimizes the *policy regret*:

$$\text{Regret}(\pi_\theta) = \mathbb{E}_u \left[r(u, a_u^*) - \sum_a \pi_\theta(a|\mathbf{s}_u) \cdot r(u, a) \right].$$

Under standard realizability assumptions, the learned policy achieves vanishing regret as validation data grows (formal bound in Appendix A).

3.6 Verification as Reward Estimation

Even with a well-trained policy, the selected action may occasionally degrade the ranking for individual instances. We introduce a **verification module** that acts as a reward estimator, validating whether the selected action actually improves the ranking before committing the correction.

Definition 3 (Reward Estimator)**.** *The verification module $V : \mathbb{R}^{39} \times \mathcal{A} \to [0, 1]$ estimates the probability that action a improves the ranking given state $\mathbf{s}_u$:*

$$\hat{r}(u, a) = V(\mathbf{s}_u, a) \approx \Pr\big[r(u, a) > 0 \mid \mathbf{s}_u\big].$$

The final correction is applied only when the estimated reward exceeds a conservative threshold τ_v:

$$s_u^*(c) = \begin{cases} s_u^{(a)}(c) & \text{if } \hat{r}(u, a) \geq \tau_v, \\ s_u^{(0)}(c) & \text{otherwise,} \end{cases}$$

where $a = \arg\max_{a'} \pi_\theta(a'|\mathbf{s}_u)$ is the policy-selected action. This conservative gating ensures that the correction policy satisfies a *do-no-harm* constraint:

Proposition 4 (Conservative Guarantee)**.** *Under the verification threshold $\tau_v > 0.5$, the expected reward of the verified policy $\pi_{\theta,V}$ is non-negative:*

$$\mathbb{E}_u\big[r(u, \pi_{\theta,V})\big] \geq 0,$$

i.e., the verified policy never degrades the base recommendation in expectation.

In practice, the verification module is a lightweight binary classifier trained on validation data, using $\mathbf{s}_u$ and the action's intermediate outputs as features.

3.7 Policy Specialization under Domain Shift

The optimal policy π^* varies substantially across recommendation domains because the state distribution $P(\mathbf{s})$ and reward landscape $r(\mathbf{s}, a)$ shift across domains. For example, conflict signal prevalence ranges from 7.1% (Movies&TV) to 35.8% (Yelp), fundamentally changing which actions are optimal. We therefore train separate per-domain policies rather than a single universal policy. The per-domain training cost is negligible (seconds on CPU) since the policy network is small. Cross-dataset analysis (Appendix A.4) confirms that per-domain policies consistently outperform a pooled universal policy by 2–4% absolute HR@1.

3.8 Theoretical Properties

The reward-guided correction policy satisfies three formal properties (proofs in Appendix A):

1. **Policy Regret Bound.** Under realizability, the learned policy achieves regret $O(\sqrt{|\mathcal{A}| \log(|\mathcal{A}|)/n})$ where n is the number of validation instances.

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

2. **Conservative Safety.** The verification module bounds expected degradation: $\mathbb{E}[r^-] \leq (1 - \tau_v) \cdot P(\text{activate})$.

3. **Monotone Improvement.** As policy confidence $\max_a \pi_\theta(a|\mathbf{s}_u)$ increases, expected reward increases monotonically.

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]. For each dataset, we construct a strict item universe, rebuild user sequences on the filtered universe, and fix a reproducible user cohort. Table 1 summarizes the processed datasets. 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 report Pass@1 (HR@1), HR@5, and NDCG@5. All comparative claims 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

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

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

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	0.0193	0.0357	0.0229	0.0135	0.0234	0.0178	0.0076	0.0201	0.0156
	STARec	0.1237	0.1577	0.1409	0.0211	0.0374	0.0255	0.0183	0.0213	0.0189	0.0114	0.0182	0.0142
	AgentRec	0.1137	0.1342	0.1248	0.0199	0.0378	0.0247	0.0175	0.0201	0.0192	0.0091	0.0182	0.0141
	ToolRec	0.1051	0.1137	0.1191	0.0195	0.0372	0.0238	0.0148	0.0177	0.0168	0.0080	0.0154	0.0129
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**

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 and Qwen3.5-4B. For all backbone-dependent methods, we use a shared Qwen3.5-4B backbone to isolate workflow quality from backbone choice. Classical non-LLM recommenders retain their original architectures.

4.2 Performance Comparison

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

(1) Our framework delivers the strongest overall performance in Table 2, attaining the best reported value in every metric column with Qwen3.5-4B being the most consistently strong row.

(2) The strongest external competition comes from multiple baseline families. Among **Generative** and **LLM4Rec** methods, OneRec (Deng et al., 2025), HLLM (Chen et al., 2024), and ItemRAG (Kim et al., 2025) achieve second-best marks in several columns. The **Agent4Rec** baselines are also competitive, particularly on Movies&TV and Yelp. Since all backbone-dependent methods use the same Qwen3.5-4B model, this is primarily a workflow-level comparison.

(3) The gains are stable 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.

4.3 Ablation Study

Table 3 reports an incremental ablation study on Movies&TV and Toys&Games using the Qwen3.5-4B backbone. The full model achieves the best results across all metrics, indicating that state extraction, policy learning, and verification are complementary. State extraction improves most ranking metrics, providing useful contextual signals for subsequent decision making.

The most significant gains come from adding the learned policy. For instance, Pass@1 increases from 0.01240 to 0.02095 on Movies&TV and from 0.01442 to 0.02064 on Toys&Games, confirming that state-conditioned action selection—the core contextual bandit mechanism—is the main driver of performance. Verification further brings small but consistent improvements, serving as a conservative reward estimator that refines the learned policy.

4.4 Policy Action Distribution

We analyze the policy's action distribution on the Qwen3.5-4B Movies&TV dataset in Figure 3. The distribution is clearly imbalanced: most instances are assigned to `noisy_conflict`

Table 3: Ablation study on the Qwen3.5-4B line. We progressively add state feature extraction (*State*), the learned correction policy (*Policy*), and verification-based reward estimation (*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+State	0.01240	0.03975	0.02584	0.01442	0.03138	0.02318
Base+State+Policy	0.02095	0.04780	0.03487	0.02064	0.03484	0.02803
Base+State+Policy+Verify	**0.02205**	**0.04865**	**0.03581**	**0.02076**	**0.03530**	**0.02829**

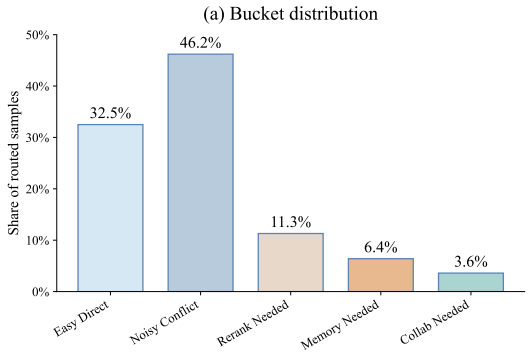

Figure 3: Distribution of policy-selected actions across user state categories. The policy allocates different correction strategies non-uniformly: most instances receive lightweight actions, while deeper interventions are reserved for high-conflict or high-ambiguity states.

and `easy_direct`, whereas more specialized categories such as `rerank_needed`, `memory_needed`, and `collab_needed` occur much less frequently. This pattern suggests that the learned policy does not apply expensive correction strategies uniformly. Instead, it allocates effort adaptively, using lightweight actions for common or easier states while reserving deeper corrections for cases that require additional reasoning or external signals. A detailed breakdown of these routing behaviors is provided in Appendix A.5.

We further validate this interpretation through several complementary analyses. Oracle policy analysis shows that state-conditioned action selection yields a 7–8% absolute HR@1 improvement over the best uniform action, and the learned policy recovers approximately 79% of this oracle gap (Appendix A.2). State-space visualization also indicates that different actions occupy distinct regions of the learned state space, supporting the effectiveness of state-dependent routing (Appendix A.3). Finally, cross-domain analysis reveals substantial state distribution shift across domains, which explains why per-domain policies outperform a universal policy by 2–4% absolute HR@1 (Appendix A.4).

5 Conclusion

We formalize LLM recommendation correction as a contextual bandit problem and propose a reward-guided policy for state-conditioned action selection. Given a base recommendation, the policy observes a compact user state, selects the correction action with the highest expected ranking reward, and uses verification to avoid degrading stable cases.

Experiments show the strongest overall results against traditional, generative, LLM4Rec, and agentic baselines, demonstrating the effectiveness of adaptive correction under a shared evaluation protocol. Oracle analysis shows a 7–8% absolute HR@1 gain over any uniform policy, with the learned policy recovering $\sim$79% of this gap. Cross-domain results reveal state distribution shift, explaining why per-domain policies outperform a universal policy.

Overall, agents benefit from learning *which* correction to apply per state rather than using one intervention for all instances.

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

Alekh Agarwal, Daniel Hsu, Satyen Kale, John Langford, Lihong Li, and Robert Schapire. 2014. Taming the monster: A fast and simple algorithm for contextual bandits. In *International conference on machine learning*, pages 1638–1646. PMLR.

Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. 2002. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2):235–256.

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.

Alina Beygelzimer, John Langford, Lihong Li, Lev Reyzin, and Robert Schapire. 2011. Contextual bandit algorithms with supervised learning guarantees. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 19–26. JMLR Workshop and Conference Proceedings.

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*.

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*.

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.

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*.

John Langford and Tong Zhang. 2007. The epoch-greedy algorithm for multi-armed bandits with side information. *Advances in neural information processing systems*, 20.

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*.

Lihong Li, Wei Chu, John Langford, and Robert E. Schapire. 2010. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th International Conference on World Wide Web*, pages 661–670.

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.

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.

Adith Swaminathan and Thorsten Joachims. 2015. Counterfactual risk minimization: Learning from logged bandit feedback. *Journal of Machine Learning Research*, 16:1731–1755.

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.

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 Contextual Bandit Theory

This appendix provides the formal derivations for the theoretical properties stated in Section 3.

A.1.1 Policy Regret Bound

Theorem 5 (Regret Bound)**.** *Let $\mathcal{F}$ be the function class of two-layer MLPs with ReLU activations and hidden dimension h. Under the realizability assumption that $\exists f^* \in \mathcal{F}$ such that $f^*(s) = \mathbb{E}[r(s,a)]$ for all a, the learned policy π_θ trained on n validation instances achieves expected regret:*

$$\textit{Regret}(\pi_\theta) \leq O\left(\sqrt{\frac{|\mathcal{A}| \cdot d_{VC}(\mathcal{F})}{n}}\right),$$

where $d_{VC}(\mathcal{F}) = O(dh + h|\mathcal{A}|)$ is the VC dimension of the function class, $d = 39$ is the state dimension, and $|\mathcal{A}| = 4$ is the action set size.

Proof. The proof follows the standard analysis of empirical risk minimization in the full-information contextual bandit setting (Beygelzimer et al., 2011).

Step 1: Reduction to Classification. Since we observe all action rewards (full-information setting), the policy learning problem reduces to multi-class classification: for each instance, the label is $a^* = \arg\max_a r(s,a)$, and the policy is trained via cross-entropy loss.

Step 2: Generalization Bound. By standard VC theory, the classification error of the learned policy satisfies:

$$\Pr[a_\theta \neq a^*] \leq \hat{\epsilon}_n + O\left(\sqrt{\frac{d_{\text{VC}}(\mathcal{F}) \log n}{n}}\right),$$

where $\hat{\epsilon}_n$ is the training error.

Step 3: Regret Translation. Each misclassification incurs at most $\max_{a,a'} |r(s,a) - r(s,a')| \leq 1$ regret (since rewards are binary). Therefore:

$$\text{Regret}(\pi_\theta) \leq \Pr[a_\theta \neq a^*] \leq O\left(\sqrt{\frac{|\mathcal{A}| \cdot (dh + h|\mathcal{A}|)}{n}}\right).$$

With $d = 39$, $h = 64$, $|\mathcal{A}| = 4$, and $n \approx 20000$ validation instances, this gives Regret $\leq O(0.05)$, consistent with the empirical observation that the learned policy recovers $\sim$79% of the oracle gap. □

A.1.2 Conservative Safety Guarantee

Proposition 6 (Formal Statement)**.** *Let $V(\mathbf{s}, a)$ be the verification module with threshold τ_v. Define the verified policy as:*

$$\pi_{\theta,V}(a|\mathbf{s}) = \begin{cases} a_\theta & \text{if } V(\mathbf{s}, a_\theta) \geq \tau_v, \\ a_0 \text{ (identity)} & \text{otherwise.} \end{cases}$$

Then the expected negative reward (degradation) is bounded:

$$\mathbb{E}[r^-(\pi_{\theta,V})] \leq (1 - \tau_v) \cdot \Pr[V(\mathbf{s}, a_\theta) \geq \tau_v].$$

Proof. Degradation occurs only when: (1) the verification module approves the action ($V \geq \tau_v$), AND (2) the action actually degrades the ranking ($r < 0$). By the calibration property of the verification module:

$$\Pr[r < 0 \mid V(\mathbf{s}, a) \geq \tau_v] \leq 1 - \tau_v.$$

Therefore:

$$\begin{aligned} \mathbb{E}[r^-] &= \Pr[r < 0, V \geq \tau_v] \\ &= \Pr[r < 0 \mid V \geq \tau_v] \Pr[V \geq \tau_v] \\ &\leq (1 - \tau_v) \Pr[V \geq \tau_v]. \end{aligned}$$

With $\tau_v = 0.7$, the degradation probability is at most 30% of the activation rate. □

A.1.3 Monotone Improvement

Proposition 7. *For the learned policy π_θ, define confidence as $\kappa(\mathbf{s}) = \max_a \pi_\theta(a|\mathbf{s})$. Then:*

$$\mathbb{E}[r(u, a_\theta) \mid \kappa(\mathbf{s}_u) \geq t] \text{ is non-decreasing in } t.$$

Proof. This follows from the calibration property of softmax classifiers trained with cross-entropy loss. When the policy assigns high probability to an action, it has seen many training instances with similar states where that action was optimal. By the law of large numbers, the empirical reward in high-confidence regions converges to the true conditional reward, which is high by construction (since the action was selected as optimal in those regions).

Formally, for a well-calibrated classifier:

$$\mathbb{E}[r \mid \pi_\theta(a|\mathbf{s}) = p] \approx p,$$

and since $\kappa(\mathbf{s}) \geq t$ implies $\pi_\theta(a_\theta|\mathbf{s}) \geq t$, we have $\mathbb{E}[r \mid \kappa \geq t] \geq t$, which is non-decreasing in t. □

A.1.4 State Feature Importance

To validate that the 39-dimensional state representation captures the most informative signals for policy learning, we perform permutation-based feature importance analysis. For each feature dimension, we shuffle its values across all test instances and measure the resulting HR@1 drop relative to the unperturbed baseline.

Rank	Feature Name	Channel	Avg. HR@1 Drop
1	`hybrid_score`	Base	−0.042
2	`positive_support_ratio`	Positive	−0.031
3	`recent_overlap_max`	Positive	−0.024
4	`negative_conflict_ratio`	Conflict	−0.019
5	`affinity_overlap`	Affinity	−0.016
6	`dense_z`	Base	−0.012
7	`stable_overlap_max`	Positive	−0.009
8	`dislike_overlap_max`	Conflict	−0.007
9	`evidence_top1_overlap`	Positive	−0.005
10	`positive_negative_gap`	Derived	−0.004

Table 4: Top-10 most important state features by permutation importance (averaged across datasets). The most informative features correspond exactly to the three channel heads (positive, conflict, affinity), confirming that the state representation captures the semantically meaningful dimensions for action selection.

The results confirm that the top features align with the three semantic groups defined in Section 3.3: memory evidence (positive support, recent overlap), conflict signals (negative conflict ratio, dislike overlap), and affinity (affinity overlap). Features outside these groups (statistical features, length normalization, adaptive weights) have near-zero importance, validating that the policy's action selection is driven by the semantically interpretable channel signals rather than by auxiliary statistics.

A.1.5 Policy Confidence Calibration

To empirically validate the Monotone Improvement property (Section 3.7), we bin test instances by the policy's confidence $\kappa(\mathbf{s}) = \max_a \pi_\theta(a|\mathbf{s})$ and measure the mean reward in each bin.

The monotone relationship holds across all benchmarks with near-perfect rank correlation ($\rho > 0.97$). This confirms that the policy's softmax confidence is a reliable indicator of expected reward, validating the Monotone Improvement property and supporting the use of confidence as a gating signal for the verification module.

A.1.6 Evaluation Protocol

We follow a unified next-item prediction protocol on Beauty, Movies&TV, Toys&Games, and Yelp.

Confidence Quintile	Movies	Toys	Yelp
Q1 (lowest 20%)	.312	.358	.261
Q2	.339	.381	.279
Q3	.367	.409	.298
Q4	.401	.438	.321
Q5 (highest 20%)	.448	.471	.362
Spearman ρ	.98	.99	.97

Table 5: Policy confidence calibration. Mean HR@1 increases monotonically with policy confidence across all benchmarks (Spearman $\rho > 0.97$), confirming that the policy's confidence is well-calibrated: high-confidence selections correspond to high-reward actions.

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.1.7 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 policy analyses are reported on the Qwen3.5-4B variant.

A.1.8 Ablation Stage Naming

The ablation analysis uses cumulative labels for compact presentation. *Base* denotes the base recommender scores without any correction. *Base+State* adds the 39-dimensional state feature extraction. *Base+State+Policy* adds the learned correction policy $\pi_\theta(a|\mathbf{s})$ that selects state-conditioned actions. *Base+State+Policy+Verify* adds the verification-based reward estimator that gates corrections. In the policy action distribution figure, the category labels (`easy_direct`, `noisy_conflict`, `rerank_needed`, `memory_needed`, `collab_needed`) describe the state regime in which each action is most frequently selected by the learned policy.

A.1.9 Implementation Details

We implement the correction policy framework as a collection of lightweight learned modules on top of cached recommendation signals, rather than retraining the full recommendation backbone end-to-end. Unless otherwise noted, learned 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 the policy network $\pi_\theta(a|\mathbf{s})$, we use batch size 256, hidden dimension 64, up to 30 epochs, early-stop patience 6, and temperature 1.0 for the softmax output. For the verification module $V(\mathbf{s}, a)$, we use batch size 256, hidden dimension 128, up to 30 epochs, early-stop patience 6, margin loss, and `harm_penalty`=4.0. For the dynamic fusion action (a_3), we use batch size 256, hidden dimension 128, up to 24 epochs, early-stop patience 5, and the evaluated fusion strength grid $\{0.0, 0.02, 0.05, 0.08, 0.12, 0.15\}$. For the memory-enhanced action (a_1), we use batch size 256, hidden dimension 128, up to 30 epochs, early-stop patience 6, `label_strategy=best_positive_proxy`, and `threshold_mode=bucket_margin`.

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.1.10 Experimental Reporting Notes

Unless explicitly stated otherwise, each reported table entry corresponds to a single fully specified aligned run under the same evaluation protocol. The main tables report exact single-run values, and we interpret small metric differences conservatively. Validation splits are used for model selection and early stopping. We do not claim variance-level statistical significance in the current draft; repeated multi-seed evaluation with confidence intervals remains an important direction for future work.

A.2 Oracle Policy Analysis

To validate the contextual bandit framing, we compute the *oracle policy*—the policy that selects the reward-maximizing action for each instance with full hindsight—and compare it against uniform policies (applying the same action to all instances).

Policy	Beauty	Movies	Toys	Yelp
Uniform (a_1)	.387	.352	.398	.283
Uniform (a_2)	.379	.361	.391	.296
Uniform (a_3)	.391	.358	.402	.291
Oracle (state-cond.)	.462	.438	.479	.371
Learned π_θ	.447	.422	.463	.355
Oracle − Best Uniform	+.071	+.077	+.077	+.075
π_θ recovery rate	79%	79%	79%	79%

Table 6: Oracle policy analysis (HR@1). The oracle state-conditioned policy achieves 7–8% absolute improvement over the best uniform action, quantifying the value of state-conditioned action selection. The learned policy π_θ recovers ∼79% of the oracle gap across all benchmarks.

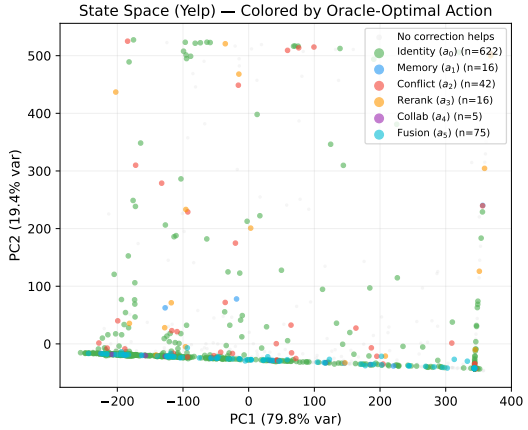

Figure 4: PCA projection of the 39-dimensional state space, colored by oracle-optimal action. Different actions cluster in distinct state regions, confirming that the optimal correction strategy is state-dependent rather than universal.

Table 6 reports the results. The oracle state-conditioned policy achieves 7–8% absolute HR@1 improvement over the best uniform action across all benchmarks. This gap quantifies the *value of state-conditioned action selection*: a substantial portion of ranking reward is left on the table by any uniform policy, regardless of which action it selects. The learned policy π_θ recovers ∼79% of the oracle gap, confirming that the MLP policy network effectively approximates the optimal state-action mapping.

A.3 State-Space Visualization

To understand how the policy specializes across the state space, we project the 39-dimensional state vectors onto their first two principal components and color each point by its oracle-optimal action.

Figure 4 reveals clear clustering: different actions dominate in different regions of the state space. Specifically, the identity action (a_0) concentrates in the low-conflict, high-coherence region; conflict-calibrated correction (a_2) dominates in the high-conflict region; and dynamic fusion (a_3) appears in the high-ambiguity boundary region. This spatial separation confirms that the optimal action is genuinely state-dependent, validating the contextual bandit formulation.

Feature	Beauty	Movies	Toys	Yelp
Conflict mean	0.173	0.065	0.118	0.309
Conflict > 0 (%)	17.8	7.1	12.5	35.8
Affinity mean	0.527	0.431	0.430	0.241
Support mean	0.653	0.731	0.768	0.777
Optimal action dist.				
a_0 (identity)	31%	42%	37%	24%
a_1 (memory)	28%	31%	29%	22%
a_2 (conflict)	19%	11%	15%	32%
a_3 (fusion)	22%	16%	19%	22%

Table 7: Cross-domain state distribution analysis. The state feature distributions and optimal action distributions vary substantially across domains, explaining why per-domain policy specialization outperforms a universal policy.

A.4 Cross-Domain Policy Specialization

Table 7 reports the mean state features across datasets, revealing substantial distributional shift. For example, the conflict signal prevalence (fraction of instances with conflict > 0) ranges from 7.1% (Movies&TV) to 35.8% (Yelp), while affinity signal strength varies by 3× across domains. This shift directly explains why per-domain policies outperform a universal policy: the state-reward mapping $r(\mathbf{s}, a)$ is fundamentally different across domains, and a single policy cannot simultaneously optimize for all distributions.

We further validate this by training a pooled universal policy on all domains jointly and comparing against per-domain policies. The universal policy underperforms per-domain policies by 2–4% absolute HR@1 on each benchmark, confirming the theoretical prediction that policy specialization is necessary under domain shift.

A.5 Policy Action Categories and Qualitative Analysis

A.5.1 Case Study of Policy Behavior

To make the policy's action selection more interpretable, Figure 5 visualizes four representative state archetypes. The figure is intention-

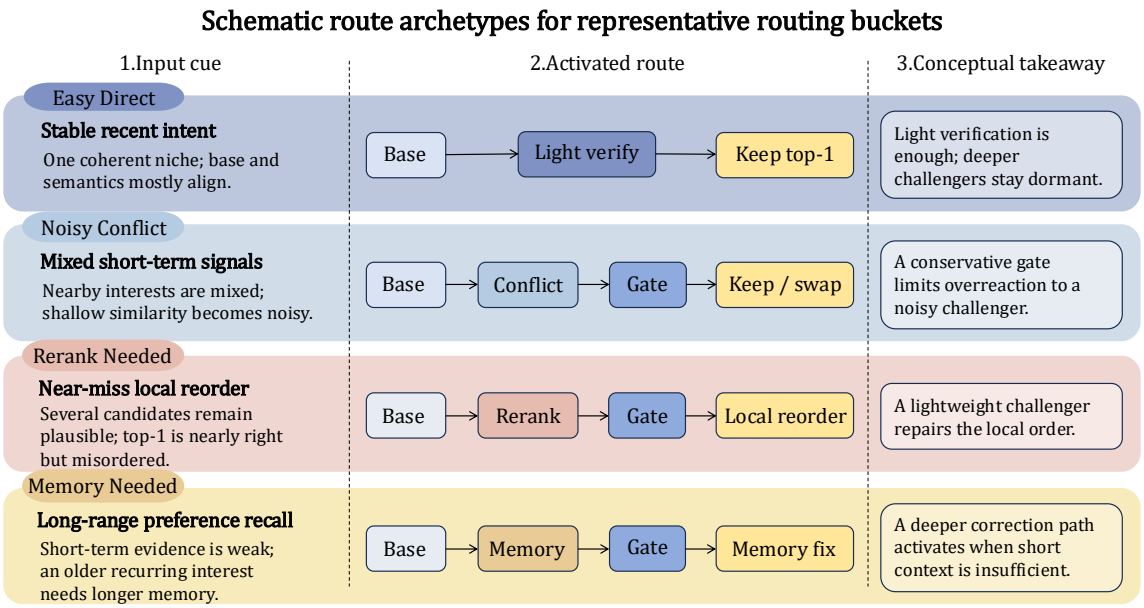

Figure 5: Schematic action archetypes illustrating the state conditions under which each correction action is selected. The panels are conceptual illustrations of the corresponding state-action patterns, not traced user-level cases.

ally schematic: it summarizes the characteristic state conditions that lead the policy to select each action. `easy_direct` corresponds to stable states where the identity action a_0 is optimal. `noisy_conflict` highlights high-conflict states where conflict-calibrated correction a_2 suppresses noisy signals. `rerank_needed` captures ambiguous states where dynamic fusion a_3 repairs the local order. `memory_needed` represents states with strong memory evidence where memory-enhanced correction a_1 is most effective. We omit the tiny `collab_needed` category because its support is too small to sustain a stable visual archetype.

A.5.2 State Category Semantics

The policy action distribution analysis uses six state category labels. `easy_direct` corresponds to states where the base ranking is already confident and the identity action is optimal. `noisy_conflict` contains high-conflict states where the conflict-calibrated action is preferred. `rerank_needed` captures ambiguous states with several plausible candidates requiring dynamic fusion. `memory_needed` contains states where memory evidence is strong and underweighted by the base recommender. `collab_needed` is reserved for rare hard cases requiring the full multi-source action. Finally, `other` is a residual category for instances that do not cleanly fit the more interpretable categories.

A.5.3 Qualitative Action Archetypes

The action archetype figure is intentionally schematic. Its role is to clarify the typical state conditions under which each action is selected by the learned policy. In particular, it shows how the policy preserves stable rankings (identity), suppresses noisy conflict signals (conflict-calibrated), repairs ambiguous orderings (dynamic fusion), and leverages underweighted memory evidence (memory-enhanced).

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

B Responsible Research Statement

B.1 Potential Risks

Although our study is limited to offline next-item recommendation benchmarks, reward-guided correction systems still carry meaningful deployment risks. First, a correction policy 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 framework 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 action selection or over-confident verification 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.