SCENARIO-AWARE MULTI-INTEREST RETRIEVAL WITH HIERARCHICAL LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Feed, search, and advertising capture different aspects of a user's interests, but their interaction histories differ sharply in frequency and timing. Combining these histories requires deciding both which events enter the model and how their signals are represented for retrieval. We present *R*ecommender *E*ngine for *D*iversified scenarios (*RED-Rec*), a hierarchical language-model framework for multi-scenario recommendation. A scenario-aware 2-D mixing policy combines per-scenario history quotas with blockwise temporal ordering. Learnable interest queries then extract multiple user representations from the mixed sequence. During training, an item-aware gate distributes supervision across queries; at serving time, each query retrieves candidates independently from precomputed item embeddings, and the results are merged by maximum similarity. This design supports cross-scenario interest modeling within a two-tower retrieval system. Controlled experiments compare the mixer and queries on a common backbone, and matched-budget comparisons improve over adapted multi-interest readouts. Public-benchmark and industrial results show gains in retrieval accuracy, with the largest cross-scenario gains on advertising. In a one-week online A/B test, *RED-Rec* increases advertiser value by 0.8864% and advertising spend by 0.3401%.

1 INTRODUCTION

A user may discover a topic in a feed, search for related information, and later interact with an advertisement. These actions are observed by different services but belong to the same evolving set of interests. Modeling them together can give a recommender useful evidence that is sparse within any one scenario. The challenge is to retain these differences while building user representations that support efficient retrieval (Figure 1). Simply concatenating the histories creates two practical difficulties. First, activity is uneven: frequent feed interactions can occupy most of a fixed-length sequence, while search and advertising contribute fewer events. Chronological ordering can also produce long runs from one scenario. Second, a single user vector must support retrieval for several concurrent interests, including interests represented by only a few interactions. We refer to these difficulties as *cross-scenario signal interference*. They motivate two design choices: allocate sequence capacity across scenarios, and provide multiple interest-specific readouts of the resulting history.

Recent work provides useful components for this design. HLLM (Chen et al., 2024) uses separate language models for item content and user histories (Chen et al., 2024); HyMiRec combines hierarchical sequence modeling with learnable interest queries (Zhou et al., 2025). Multi-scenario methods share information through domain-aware experts or adapters (Sheng et al., 2021; Zhang et al., 2024), while recent approaches address mixed search–recommendation sequences and two-tower matching (Pan et al., 2026; Wang et al., 2026). However, these lines of work do not jointly address how scenario imbalance affects both the construction of the interaction sequence and the allocation of representational capacity across user interests. Our focus is on allocating both input capacity and interest readouts across feed, search, and advertising, while keeping user encoding independent of candidate items at serving time.

We propose *RED-Rec*, a hierarchical two-tower framework with a multimodal Item LLM and a sequential User LLM. A scenario-aware 2-D mixing policy selects events under per-scenario quotas and optimizes block interleaving to balance temporal fidelity and scenario repetition. Scenario-specific learnable queries read the mixed history into multiple interest vectors. An item-aware hybrid

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

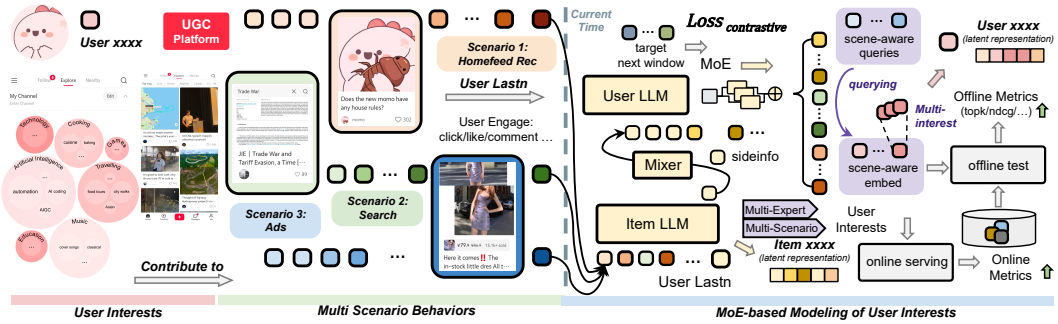


Figure 1: **Learning user interests from diverse behavioral contexts.** *Left:* User interests arise from multiple concurrent intents expressed through heterogeneous behaviors across feeds, search, and advertisements. *Middle:* These signals are organized into **extended multi-scenario behavioral sequences**, capturing long-term interest evolution. *Right:* The User LLM reads the mixed history through several learnable queries, producing interest vectors for candidate retrieval. Terminology is defined in Section A.

gate distributes training signals across these vectors, whereas inference uses parallel retrieval and maximum-score fusion over precomputed item embeddings. Joint representation learning followed by user-side refinement provides a shared item space for the queries.

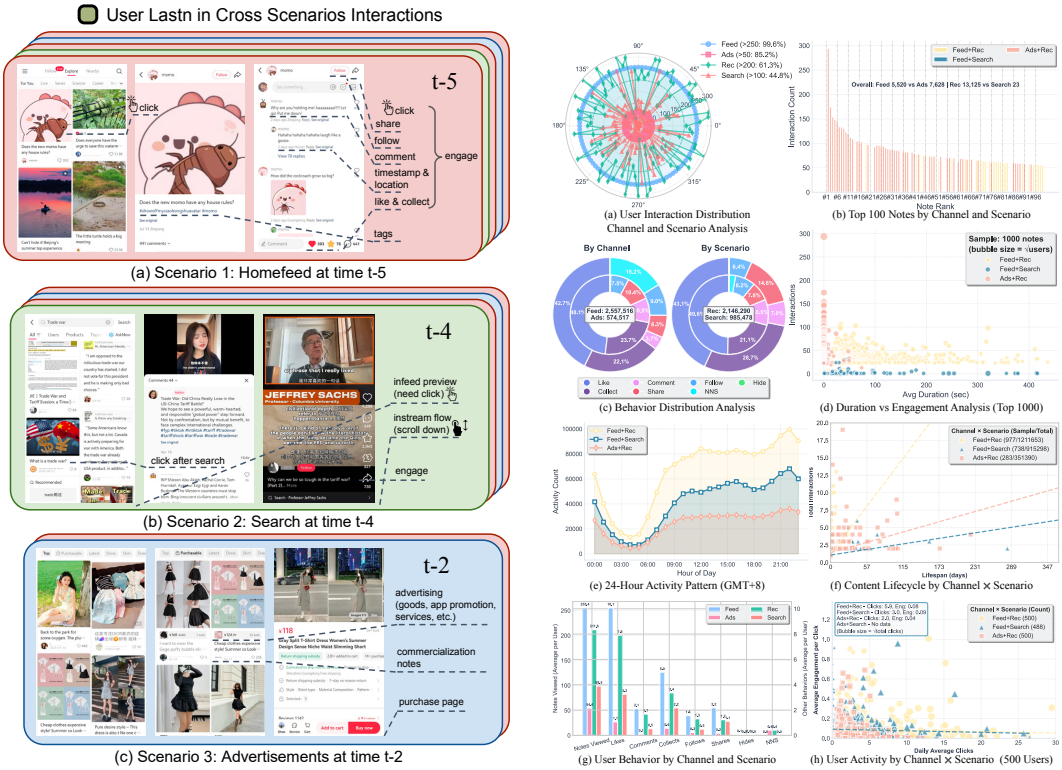
We evaluate the resulting design through single-scenario retrieval, mixed-scenario retrieval, component ablations, and an online A/B test. The experiments use public benchmarks and an industrial dataset covering feed, search, and advertising. On the industrial data, adding feed and search histories improves advertising retrieval, and multiple interest queries further increase accuracy. The online test measures gains in advertiser value and advertising spend alongside engagement metrics. Together, these results connect the sequence and readout design to a production retrieval setting with approximately 1.1 billion active items.

2 Related Work

Sequential and LLM-based recommendation. Sequential recommenders model interaction histories using recurrent networks (Hidasi et al., 2016) and self-attention (Kang & McAuley, 2018; Sun et al., 2019). HSTU scales sequence modeling for recommendation (Zhai et al., 2024), while HLLM uses language models to encode item content and user histories (Chen et al., 2024). Generative systems such as OneRec and MTGR explore item generation and large-scale recommendation pipelines (Deng et al., 2025; Han et al., 2025). *RED-Rec* uses language models as embedding encoders and retrieves candidates from precomputed item representations.

Multi-scenario recommendation. Multi-scenario methods learn shared and scenario-specific representations. STAR, AdaSparse, and APG adapt model parameters or structures to different domains (Sheng et al., 2021; Yang et al., 2022; Yan et al., 2022). M3oE separates shared, domain-specific, and task-specific information through multiple expert modules (Zhang et al., 2024). LLM4MSR derives scenario and user knowledge from language models (Wang et al., 2024), and LLM4CDSR uses language-model representations for cross-domain sequential recommendation (Liu et al., 2025). More recently, DSMOE addresses scenario imbalance in two-tower matching through adaptive projections and distillation (Wang et al., 2026). MIJSR extracts structural and semantic interests from mixed search and recommendation sequences (Pan et al., 2026). These methods motivate explicit treatment of scenario identity at sequence construction and representation learning.

Multi-interest and multimodal modeling. MIND and ComiRec represent users with several interest vectors for candidate retrieval (Li et al., 2019; Cen et al., 2020). HM4SR combines interactive and temporal expert modules to process multimodal sequences (Zhang et al., 2025). HyMiRec uses learnable queries and future-window matching within a hybrid LLM framework (Zhou et al., 2025). MuSeR combines temporal compression, multiple interest queries, and multimodal alignment for industrial retrieval (Fu et al., 2026). Our design brings scenario-specific history quotas and query allocation into the same hierarchical encoder, with item-aware training and candidate-independent user encoding at inference.



(a) Scenario 1: Homefeed at time t-5

(b) Scenario 2: Search at time t-4

(c) Scenario 3: Advertisements at time t-2

(a) Scenarios in our data.

(b) User engagement analytics dashboard.

Figure 2: **Overview of the dataset.** (a) shows the three scenarios covered; (b) summarizes the engagement analytics across 10k sampled users.

3 Multi-Scenario Interaction Data

Public datasets support different aspects of recommendation research: Amazon and MovieLens provide item interactions (Ni et al., 2019; Harper & Konstan, 2015), KuaiRand supports recommendation under randomized exposure (Gao et al., 2022), and Qilin records multimodal search and recommendation within app-level sessions (Chen et al., 2025). Our industrial dataset connects feed, search, and advertising histories for the same users, allowing us to study their contribution to retrieval within a shared item space.

The dataset contains homefeed browsing, search-driven interactions, and advertisement engagement from a UGC platform. It records clicks, likes, collections, shares, negative feedback, and viewing duration. The training dataset summarized in Table 7 contains 1.0 million users and 683.2 million actions. We retain users with at least 30 valid homefeed clicks and 5 valid advertisement clicks, where a valid click requires more than 5 seconds of viewing. This selection defines an active-user cohort with histories in both scenarios.

See dataset examples in Figure 2a; overall statistics in Figure 2b; additional details in Section C.

4 *RED-Rec*

Let $\mathcal{U}$ and $\mathcal{I}$ denote the user and item sets. Items include image–text posts, videos, and advertisements. For each user u, the chronological history is $S_u = ((i_j, a_j, s_j, t_j))_{j=1}^{T_u}$, where $i_j \in \mathcal{I}$ is the item, a_j is the action, $s_j \in \mathcal{S} = \{\text{homefeed}, \text{search}, \text{ads}\}$ is the scenario, and t_j is the timestamp.

Our objective is to learn a mapping $f_u(u, S_u) \rightarrow \mathbf{U}_u$ and $f_i(i) \rightarrow \mathbf{V}_i$ into a shared d-dimensional embedding space. For training, we sample a cut position c and a look-ahead horizon $\Delta \sim \text{Unif}(\{1, \ldots, W\})$, use the prefix $S_u^{\leqslant c}$ as input, and predict the target item $i^+ = i_{c+\Delta}$ (events after c are masked to avoid leakage). We optimize a unified next-item retrieval objective over $\mathcal{I}$, and report scenario-specific metrics using the held-out positive set for each target scenario.

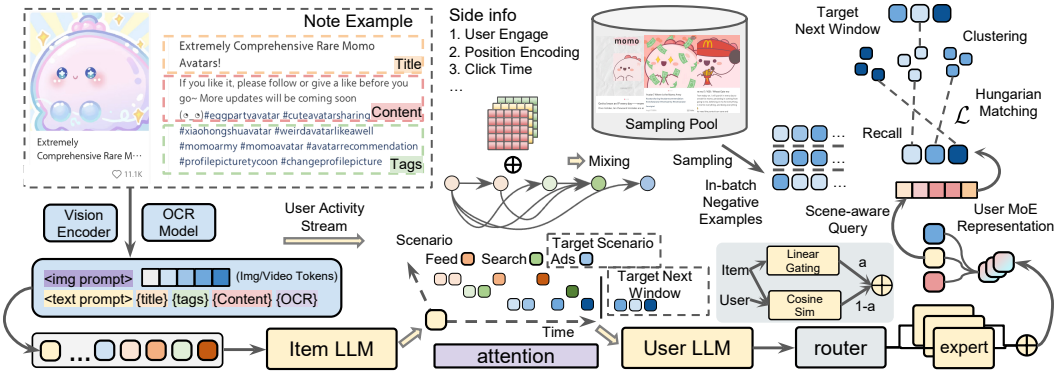


Figure 3: **Overview of *RED-Rec*.** An Item LLM encodes multimodal content, and a User LLM processes the mixed behavioral history. Scenario quotas and blockwise ordering construct the input sequence; scenario-specific queries read it into multiple interest vectors. Training uses item-aware gating, while inference retrieves candidates independently for each query and merges their scores. Joint alignment is followed by user-side refinement with a frozen item encoder.

Sequence construction and interest readout. The mixer controls which events enter the encoder and how they are ordered. The query module controls how the encoded history is read into retrieval vectors. We call these the temporal and semantic axes of the design, respectively. Scenario quotas reserve input capacity for sparse histories; multiple queries provide separate vectors for different interests. Section D relates these choices to the ablation results.

4.1 Hierarchical LLM-based Item Representation

We employ a multi-modal encoder E_{item} to generate item embeddings. For an item i, we concatenate its textual features $\mathbf{x}_i$ (titles, tags, OCR text) and visual features $\mathbf{v}_i$ (encoded via a ViT backbone). To align multimodal signals with the LLM's latent space, we introduce a learnable item embedding token $\mathbf{e}_{token} \in \mathbb{R}^d$, treated as a special prefix during the forward pass:

$$\mathbf{e}_i = \left[\mathrm{LLM}_{\text{item}}([\mathbf{e}_{token}; \mathrm{Text}_i; \mathrm{Vision}_i]; \theta_i)\right]_{\text{token}}. \tag{1}$$

Here $[\cdot]_{\text{token}}$ selects the output state of the pooling token. The Item LLM uses bidirectional attention over the pooling token and valid content tokens, allowing the prefix token to aggregate both text and visual features. The same Item LLM encodes content from all scenarios; scenario labels describe interactions and do not modify its content embedding. Expert-specific heads $P_e(\cdot)$ project $\mathbf{e}_i$ to $\{\mathbf{z}_{i,1}, \ldots, \mathbf{z}_{i,E}\}$ for matching against user interest vectors.

4.2 Sequential User Interest Modeling

Temporal Axis: Scenario-Aware 2-D Dense Mixing. We construct the User LLM input under a history budget L. Each scenario receives a quota n_s, with $\sum_s n_s = L$, and contributes up to n_s of its most recent events. We divide each scenario history into chronological blocks of at most b events, retaining within-block and within-scenario order. Let J be the total number of nonempty blocks, t_j the latest timestamp in block j, and s_j its scenario. We optimize over interleavings Π_b of the scenario-specific block sequences:

$$\min_{\pi \in \Pi_b} \underbrace{\frac{\sum_{k=1}^{J} |t_{\pi(k)} - t^{\uparrow}(k)|}{JT}}_{\text{normalized temporal distortion}} + \beta \underbrace{\frac{\sum_{k=2}^{J} \mathbb{I}[s_{\pi(k)} = s_{\pi(k-1)}]}{J-1}}_{\text{normalized block repetition}}, \tag{2}$$

where $t^{\uparrow}(k)$ is the k-th smallest block timestamp and $T = \max_j t_j - \min_j t_j$. We define temporal distortion as zero when $T = 0$, and repetition as zero when $J \leqslant 1$. Both terms lie in $[0, 1]$, making $\beta \geqslant 0$ dimensionless. Dynamic programming finds an exact minimum over Π_b (Section G.1). The penalty discourages consecutive blocks from one scenario but does not impose a hard bound on run length.

Each interaction is serialized with a learnable scenario token, enabling scenario-conditioned dynamics in a unified sequence appended with MoE queries. Since interactions are non-uniformly distributed in

time, we further employ a 2-D positional encoding $\mathbf{p}_j = \mathbf{PE}_{\text{seq}}(j) + \mathbf{PE}_{\text{gap}}(\Delta t_j)$, where $\mathbf{PE}_{\text{seq}}(j)$ is the absolute sequence index and $\mathbf{PE}_{\text{gap}}(\Delta t_j)$ is a learnable embedding for the elapsed time since the j-th event.

Semantic Axis: Query-based Mixture-of-Experts. We append learnable queries to the mixed history and use their output states as retrieval representations. Each scenario has its own query bank, allowing it to read different interests from the shared encoded history. These query-based experts operate at the readout; their routing weights assign training credit across interest vectors.

Expert Queries. For each target scenario s, we learn E vectors $\mathbf{Q}^{(s)} = [\mathbf{q}_1^{(s)}, \ldots, \mathbf{q}_E^{(s)}]$. They are input tokens that summarize the history, distinct from the query projections computed inside attention layers. During mixed-scenario training, we append the three query banks to the shared history embeddings $\hat{\mathbf{X}}_u$:

$$\begin{aligned}&[\mathbf{H}_u^{\text{hist}}, \mathbf{R}^{(\text{hf})}, \mathbf{R}^{(\text{sr})}, \mathbf{R}^{(\text{ads})}] \\ &\quad = E_{\text{user}}([\hat{\mathbf{X}}_u; \mathbf{Q}^{(\text{hf})}; \mathbf{Q}^{(\text{sr})}; \mathbf{Q}^{(\text{ads})}]; \theta_u, M).\end{aligned} \tag{3}$$

The attention mask M lets historical tokens attend bidirectionally within the history, but not to queries. Each query attends to the entire mixed history and to queries in its own scenario bank; different banks cannot attend to one another. This pattern is applied in every layer (Section G.1). Cross-scenario information is shared through the history and backbone, while each bank provides a scenario-specific readout.

Each query produces an interest vector $\mathbf{u}_e = P_e(\mathbf{R}_e^{(s)})$. A training target supervises only the bank for its scenario; specialization within that bank is learned through the retrieval objective. The projection P_e is shared between the item and user representations for expert e. We omit the scenario superscript below.

During training, the target item's representation determines the routing weight for expert e:

$$\begin{aligned}W(i,e) = \alpha \cdot \text{Softmax}(\mathbf{W}_g \mathbf{e}_i + \mathbf{b}_g)_e \\ + (1-\alpha) \cdot \frac{\exp(\cos(\mathbf{u}_e, \mathbf{z}_{i,e})/\tau)}{\sum_{e'=1}^{E} \exp(\cos(\mathbf{u}_{e'}, \mathbf{z}_{i,e'})/\tau)}.\end{aligned} \tag{4}$$

The first term gates on item content; the second weights experts by user–item similarity. Both distributions are normalized over the E experts of the target scenario.

At inference, we retain only the query bank for the requested scenario and preserve its training-time position IDs. Since neither the history nor that bank reads other banks, omitting them preserves the retained computation. The User LLM computes $\{\mathbf{u}_1, \ldots, \mathbf{u}_E\}$ once; each expert retrieves candidates using precomputed item embeddings, and the candidate sets are merged by $\text{score}(u,i) = \max_e \cos(\mathbf{u}_e, \mathbf{z}_{i,e})$. Item-conditioned gating is used only in training, so user encoding is shared across all candidates. This separation supports parallel approximate nearest-neighbor retrieval (Section G.2).

4.3 Two-stage Training and Optimization

Training consists of joint representation alignment followed by user-side refinement.

Stage I: Joint Representation Alignment We jointly optimize the item encoder E_{item}, user encoder E_{user}, and MoE experts to learn a shared embedding space.

Stage II: User-side Interest Refinement We freeze the item encoder (and item projection heads) and optimize only the user-side modules (including MoE queries), which provides a stationary item space for more stable expert specialization:

$$(\theta_u^*, \mathbf{Q}^*) = \arg\min_{\theta_u, \mathbf{Q}} \mathcal{L}_{\text{II}}(S_u;\ \text{frozen } \theta_i). \tag{5}$$

Freezing the item encoder in Stage II keeps the retrieval space fixed while the user encoder and queries adapt to it.

Objective. We use a weighted NCE loss with negatives sampled from the active item pool:

$$\mathcal{L}_{\text{NCE}} = \sum_{e=1}^{E} W(i,e) \cdot \ell_{\text{NCE}}(\mathbf{u}_e, \mathbf{z}_{i,e}), \tag{6}$$

Table 1: **Recommendation results for individual scenarios.** Baseline methods are evaluated separately on the homefeed and ads scenarios. In all metrics, higher scores reflect better performance. **Bold** indicates the best result per column.

	Homefeed				Ads			
Baselines ↑	HR/NDCG$_{10}$	HR/NDCG$_{100}$	HR/NDCG$_{1k}$	MRR$_{*100}$	HR/NDCG$_{10}$	HR/NDCG$_{100}$	HR/NDCG$_{1k}$	MRR$_{*100}$
SASRec	1.74/1.11	12.35/1.78	31.02/4.06	1.01	3.26/1.63	14.08/3.71	39.11/5.27	1.57
MoRec	1.78/1.25	12.48/2.23	31.98/**4.12**	1.21	3.47/1.67	13.98/**3.88**	38.27/4.89	1.78
HSTU	1.79/1.22	12.72/2.21	31.76/3.69	1.15	3.85/**1.70**	14.32/3.30	38.20/**5.38**	1.43
HLLM	1.88/1.15	13.24/2.21	32.57/4.01	1.23	4.21/1.21	14.27/3.37	39.21/4.48	1.39
DLRM-v3	1.63/1.03	11.33/2.01	28.96/3.72	1.13	3.54/1.21	15.27/3.22	35.39/4.27	1.67
RED-Rec	2.33/1.27	12.54/2.02	31.99/3.87	1.27	4.24/1.28	16.44/3.21	40.18/4.61	1.96
RED-Rec-mm	2.35/1.21	14.20/2.27	31.29/3.97	1.29	4.31/1.31	17.22/3.18	41.86/4.66	1.92
RED-Rec-MoE	**3.17/1.31**	**15.41/2.38**	**35.16**/4.11	**1.33**	**4.81**/1.69	**19.33**/3.73	**44.28**/5.27	**2.27**

where $W(i, e)$ is the routing weight and ℓ_{NCE} is the per-expert contrastive loss with a learnable temperature and negatives $\mathcal{N}$. In Stage II, we additionally cluster future-window items into interests, match their centroids to queries by Hungarian assignment, and apply matched-pair NCE with weight $\lambda_{\text{match}} = 0.1$ (Section G.1). We also use three regularizers:

$$\begin{aligned} \mathcal{L}_{\text{div}} &= \sum_{e \neq e'} \cos^2(\mathbf{q}_e, \mathbf{q}_{e'}), \\ \mathcal{L}_{\text{bal}} &= \text{Var}_e(\overline{W}_e) - \lambda \text{H}(\overline{\mathbf{W}}), \\ \mathcal{L}_{\text{spec}} &= \frac{1}{B} \sum_{e=1}^{E} \sum_{a=1}^{B} w_{ae}[1 - \cos(\widehat{\mathbf{z}}_{i_a,e}, \boldsymbol{\mu}_e)]. \end{aligned} \tag{7}$$

where $\overline{W}_e = \frac{1}{B} \sum_{a=1}^{B} W(i_a, e)$ is the batch-average routing weight, $\overline{\mathbf{W}} = (\overline{W}_1, \ldots, \overline{W}_E)$, $\text{H}(\cdot)$ denotes entropy, and $w_{ae} = W(i_a, e)$ is the soft assignment of the positive in training example a, conditioned on that example's user history. We use normalized item vectors $\widehat{\mathbf{z}}_{i_a,e} = \mathbf{z}_{i_a,e} / \|\mathbf{z}_{i_a,e}\|_2$ and weighted centroids

$$\boldsymbol{\mu}_e = \frac{\sum_{a=1}^{B} w_{ae} \widehat{\mathbf{z}}_{i_a,e}}{\sum_{a=1}^{B} w_{ae}}. \tag{8}$$

The sums are computed within each target scenario. We apply $\mathcal{L}_{\text{spec}}$ only in Stage II, with the item encoder and item projection heads frozen. Neither the soft routing weights nor the weighted centroids are detached, so gradients reach the user representations through the similarity-dependent assignments. Cosine normalization uses a small positive floor for near-zero norms. The diversity term discourages redundant queries; $\mathcal{L}_{\text{bal}}$ controls expert utilization; $\mathcal{L}_{\text{spec}}$ encourages angularly compact item assignments.

Additional implementation details are in Section G.

5 Experiments

We evaluate retrieval quality, the contributions of the mixer and queries under a common backbone, and their sensitivity to input allocation and model size. Section 5.3 gives controlled component and multi-interest comparisons; Section E examines attention, expert usage, and routing. Public-benchmark results, including five-seed estimates on Amazon Books, appear in Section B.

Setup. We evaluate on the public Amazon Books dataset (McAuley et al., 2015) and our industrial dataset (1M users, 10K test samples). Both item and user encoders are initialized from Qwen2.5-1.5B (Qwen Team, 2024), with CLIP ViT-B/16 as the vision encoder. We set look-ahead window W=10, default history length 128, and 5 MoE expert queries per scenario; the 2-D mixer uses per-scenario quotas $\{n_{\text{hf}}, n_{\text{ads}}, n_{\text{sr}}\}$=$\{64, 32, 32\}$ chosen from empirical statistics, the block size is $b = 4$ with mixing weight $\beta = 0.2$, and the hybrid gate uses α=0.5. Training takes ∼24h on 8×H100 (3 epochs, per-device batch 2, gradient accumulation 4). We compare against SASRec (Kang & McAuley, 2018), MoRec (Yuan et al., 2023), HSTU (Zhai et al., 2024), HLLM (Chen et al., 2024), M3oE (Zhang et al., 2024), LLM4CDSR (Liu et al., 2025), and DLRM-v3 (Meta, 2026), and report three variants: *RED-Rec* (base), *RED-Rec*-mm (multimodal), and *RED-Rec*-MoE. Standard ranking metrics are used (Section F); Amazon Books results are deferred to Section B.

5.1 SINGLE-SCENARIO

Table 1 reports retrieval results for Homefeed and Ads. The MoE variant has the highest HR at all three cutoffs in both scenarios. On Homefeed, HR@10 rises from 1.88 for HLLM to 3.17 for *RED-Rec*-MoE; on Ads, it rises from 4.21 to 4.81. MRR also improves in both scenarios. The NDCG results are more mixed, particularly on Ads, where HSTU and MoRec remain competitive. The clearest benefit is therefore in candidate coverage and early retrieval accuracy.

5.2 MULTI-SCENARIO

Table 2 evaluates mixed histories. For the base model, adding Homefeed to Ads raises Ads HR@10 from 4.24 to 4.36, and adding Search raises it further to 4.72. The corresponding MoE results are 4.81, 5.04, and 5.13. In the three-scenario setting, *RED-Rec*-MoE achieves the highest reported HR, NDCG, and MRR. These results show that feed and search histories provide useful signals for ads retrieval.

Table 2: **Recommendation results for mixed scenarios.** The target scenario is shown in parentheses. Bold marks the best value for each metric within each block, including ties.

Search + Homefeed (for Homefeed)				
Baselines ↑	HR/NDCG$_{10}$	HR/NDCG$_{100}$	HR/NDCG$_{1k}$	MRR$_{*100}$
SASRec	1.71/1.21	12.27/2.68	32.32/3.77	1.52
MoRec	1.79/1.30	13.92/2.99	33.01/3.98	1.53
LLM4CDSR	1.75/1.30	13.42/2.78	32.28/3.78	1.54
M3oE	1.62/1.21	11.37/3.01	29.77/4.01	1.52
HSTU	1.79/1.25	12.84/3.28	33.15/4.24	1.55
HLLM	1.88/1.34	13.50/3.19	33.06/4.20	**1.58**
DLRM-v3	1.64/1.18	11.35/3.02	30.89/3.98	1.48
RED-Rec	2.26/1.32	14.74/3.16	33.29/4.20	**1.58**
RED-Rec-MoE	**3.19/1.44**	**15.67/3.37**	**35.88/5.02**	**1.58**
Homefeed + Ads (for Ads)				
Baselines ↑	HR/NDCG$_{10}$	HR/NDCG$_{100}$	HR/NDCG$_{1k}$	MRR$_{*100}$
SASRec	3.72/1.24	16.18/3.08	38.94/4.72	1.94
MoRec	3.80/1.30	17.23/2.62	38.29/4.77	1.98
LLM4CDSR	3.88/1.27	18.24/3.02	38.92/4.66	1.92
M3oE	3.67/1.11	16.24/3.11	36.79/4.24	1.89
HSTU	3.89/1.28	16.95/3.15	40.12/4.81	2.01
HLLM	3.68/1.19	17.24/3.12	39.76/4.78	1.97
DLRM-v3	3.52/1.21	15.43/2.95	36.87/4.58	1.87
RED-Rec	4.36/1.31	18.32/3.27	42.61/5.02	2.11
RED-Rec-MoE	**5.04/1.47**	**19.77/3.70**	**48.96/5.03**	**2.46**
Homefeed + Search + Ads (for Ads)				
Baselines ↑	HR/NDCG$_{10}$	HR/NDCG$_{100}$	HR/NDCG$_{1k}$	MRR$_{*100}$
SASRec	3.67/1.21	14.31/2.39	38.21/4.33	1.94
MoRec	3.82/1.33	18.27/2.98	38.41/4.66	1.98
LLM4CDSR	3.82/1.26	18.02/2.88	38.02/4.24	1.84
M3oE	3.46/1.12	14.98/2.88	38.21/4.45	1.75
HSTU	3.92/1.31	17.21/3.19	40.14/4.81	2.11
HLLM	4.08/1.11	19.92/3.18	43.27/4.91	2.06
DLRM-v3	3.34/1.01	14.08/2.81	35.74/4.36	1.74
RED-Rec	4.72/1.33	18.33/3.22	42.89/4.97	1.94
RED-Rec-MoE	**5.13/1.48**	**20.42/3.88**	**48.67/5.11**	**2.58**

The effect varies by target scenario and cutoff. Adding Search to Homefeed raises the MoE model's HR@100 from 15.41 to 15.67 and HR@1000 from 35.16 to 35.88, while HR@10 changes from 3.17 to 3.19. For Ads, the gains from cross-scenario histories are larger. This pattern makes scenario-specific evaluation useful even when all histories are processed by one model.

5.3 CONTROLLED COMPARISONS

Mixer and interest queries. We compare the four combinations of sequence mixing and query-based MoE with Qwen2.5-1.5B, history length 128, and Homefeed + Search + Ads histories for Ads retrieval. The backbone, item features, and pretraining setup are held fixed. The no-mixer setting concatenates scenario histories without 2-D mixing; it differs from the global timestamp ordering in Table 3.

Panel b of Table 3 contains the controlled comparison. The full model has the highest value on all four controlled metrics. Adding queries to the mixer raises HR@10 from 4.72 to 5.13 and HR@100 from 18.33 to 20.42. Adding the mixer to the query model raises the same metrics from 4.51 to 5.13 and from 17.78 to 20.42. The interaction depends on the metric: the joint benefit is particularly pronounced at cutoff 100. Section D reports the corresponding interaction contrasts.

Controlling backbone scale and interest count. Table 4 compares the complete design with the backbone-only model at three Qwen2.5 sizes. The added components improve HR@10 at each size, with relative gains of 56.1%, 19.9%, and 24.8%. These within-size comparisons isolate the gain from adding the retrieval components more directly than comparisons across model families.

We also adapt MIND (Li et al., 2019) and ComiRec-SA (Cen et al., 2020) to the same hierarchical Qwen2.5-1.5B backbone. MIND applies dynamic routing with five capsules to projected User LLM states and uses label-aware attention during training. ComiRec-SA extracts five interests with self-

attention and selects the highest-scoring interest during training. All three models use maximum-score fusion at inference and are trained for 20,000 steps. Under this budget, *RED-Rec*-MoE improves HR@10 from 3.81 for ComiRec-SA to 4.61, a 21.0% relative gain. These shortened-run results are reported separately from the full-training results.

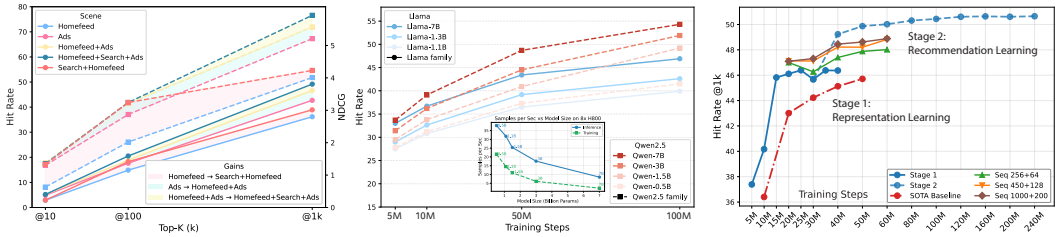


Figure 4: **Cross-scenario gains and model scaling.** (a) Performance gains achieved through unified modeling, where the shaded region represents the improvement; (b) Scaling behavior of model size with respect to click hit rate; (c) Performance gains achieved through two-stage pre-training.

Sensitivity and representation diagnostics. The full quota sweep shows that allocation matters: HR@10 is 4.92–5.13 for Homefeed:Ads:Search ratios between 4:1:1 and 1:1:1, but falls to 2.71 at 1:2:2. Changes in block size and the gate coefficient have smaller effects around the default settings. Attention and expert diagnostics on 10,000 test users complement the retrieval results: mixing retains more attention on sparse-scenario tokens, while the experts' retrieved sets have average pairwise Jaccard@100 of 0.18. Metric definitions and more ablations appear in Section E.

Table 3 varies history length, query count, pretraining, and the mixing policy. On Homefeed, increasing the history length from 32 to 128 raises HR@10 from 1.72 to 2.33; extending it to 288 leaves HR@10 unchanged. Increasing the number of expert queries from one to five raises HR@10 from 1.85 to 3.17. For mixed histories, 2-D mixing achieves HR@10 of 4.72, compared with 4.40 for timestamp-based 1-D mixing and 2.10 for global timestamp ordering. These ablations support allocating capacity across both input histories and interest readouts.

5.4 Module Ablation

Routing and Fusion. Hybrid gating with max fusion achieves HR@10 of 5.13, compared with 5.02 for similarity-only gating, 4.86 for content-only gating, and 4.94 for hard top-1 expert matching. Evaluating the same hybrid-supervision checkpoint and candidate pool, similarity-weighted soft fusion ($\tau_f = 0.1$) gives HR@10 of 5.01, versus 5.13 for max fusion (Table 11a). The comparison supports item-aware training combined with max-fused retrieval.

Qualitative Interpretation. The attention maps in Figure 5 reveal query-specific readout patterns, with different queries focusing on distinct portions of the behavioral history and thereby capturing multiple interests for retrieval. This specialization is also reflected quantitatively: across 10,000 test users, within-expert item

Table 3: **Ablations of *RED-Rec*.** (a) Homefeed ablations. History-length and pretraining experiments use the base model without MoE; pt denotes additional recommendation pretraining, while all variants are initialized from pretrained Qwen2.5 weights. The expert-count sweep uses the MoE variant with history length 128 and varies only the number of experts E. (b) The mixer sweep and controlled mixer × MoE comparison for Homefeed + Search + Ads → Ads. Bold marks the best value within each panel; dashes denote unreported cutoffs.

(a) History length, pretraining, and expert count.

Setting	HR/NDCG∗10	HR/NDCG∗100	HR/NDCG∗1k	MRR∗100
len=128, pt	2.90/1.48	14.89/2.02	**36.16**/4.01	1.30
len=288	2.33/1.42	12.48/1.83	32.46/3.88	1.27
len=128	2.33/1.27	12.54/2.02	31.99/3.87	1.27
len=64	2.08/0.71	11.32/1.94	28.67/3.92	1.29
len=32	1.72/0.61	9.48/1.76	25.47/3.64	1.29
len=128, $E = 1$	1.85/0.72	10.24/1.95	26.78/4.12	1.31
len=128, $E = 3$	2.01/0.83	11.86/1.88	29.45/**4.14**	1.31
len=128, $E = 5$	**3.17**/1.31	**15.41/2.38**	35.16/4.11	**1.33**

(b) Mixer policy and query-based MoE.

Mixer strategy	MoE	HR/NDCG∗10	HR/NDCG∗100	HR/NDCG∗1k	MRR∗100
By Timestamp	No	2.10/0.53	10.55/1.90	21.44/2.20	0.65
Naive Combination (A)	No	4.28/1.22	17.60/3.06	41.90/4.85	1.85
Naive Combination (B)	Yes	4.51/1.33	17.78/3.13	–	–
1D (on position)	No	4.31/1.23	17.65/3.08	41.95/4.87	1.86
1D (on timestamp)	No	4.40/1.25	17.80/3.10	42.20/4.90	1.88
2D-Mixing (C)	No	4.72/1.33	18.33/3.22	42.89/4.97	1.94
2D-Mixing (D)	Yes	**5.13/1.48**	**20.42/3.88**	**48.67/5.11**	**2.58**

cosine similarity is 0.71, compared with 0.29 between experts, while pairwise retrieval Jaccard@100 is only 0.18 (Section E.3). These results indicate that the experts form coherent yet differentiated retrieval spaces. The mixer further increases attention entropy and sparse-scenario coverage, with larger effects under long same-scenario runs (Section E.2), allowing infrequent behaviors to remain visible in the encoded history.

Model Scaling. We use Qwen2.5-1.5B (Qwen Team, 2024) as the default backbone and study the accuracy–throughput trade-off across model scales in Figure 4(b). Increasing model size generally improves hit rate at the cost of lower Sample per Second (SPS), exposing a direct trade-off between retrieval quality and serving capacity; the 1.5B model is therefore used as our deployment configuration. Importantly, the matched within-backbone comparisons in Table 4 show consistent gains from our components across the tested scales. Finally, Figure 4(c) isolates the training strategy and shows the benefit of the two-stage procedure: Stage I aligns the user and item encoders, while Stage II refines user representations against the resulting fixed item space.

Table 4: **Comparisons within a common backbone and training setting.** (a) Backbone-only versus *RED-Rec*-MoE at each Qwen2.5 size. (b) Five-interest readouts on Qwen2.5-1.5B after 20,000 steps. Both panels use history length 128 and Homefeed + Search + Ads → Ads.

(a) Backbone scale

Backbone	Size	Backbone-only HR@10	*RED-Rec* HR@10	Relative gain
Qwen2.5	0.5B	3.12	4.87	56.1%
Qwen2.5	1.5B	4.28	5.13	19.9%
Qwen2.5	7B	4.28	5.34	24.8%

(b) Multi-interest readout

Readout (20k steps)	HR@10	NDCG@10	HR@100	NDCG@100
MIND adaptation	3.58	1.08	16.12	2.78
ComiRec-SA adaptation	3.81	1.16	16.74	2.91
RED-Rec-MoE	**4.61**	**1.41**	**18.76**	**3.50**

5.5 Online Serving

We evaluated *RED-Rec* as a candidate generator for advertising in a one-week online A/B test, allocating 10% of traffic to treatment and 10% to control. The deployment environment contains approximately 1.1 billion active items and an active user population of approximately 160 million.

RED-Rec increased advertiser value by 0.8864% and advertising spend by 0.3401%, both with $p < 0.01$ (Table 5). More than 90% of its retrieved items were unique to this recall channel, adding candidates beyond those supplied by existing channels.

Table 5: **Online A/B Test Results.** * denotes statistical significance with $p < 0.01$.

Domain	Metric	Change (%)
Community	APP_LT (long-term active users)	-0.0015
	SAU_LT (search-related)	-0.0007
E-commerce	Overall Purchase UV	-0.0886
Live Streaming	Live Broadcast Engagement UV	-0.0694
Commercialization	Advertiser Value (*ADVV*)	**+0.8864***
	Advertising Spend	**+0.3401***

Purchase UV and live-broadcast engagement UV changed by -0.0886% and -0.0694%, respectively; the negative movements in Table 5 are not statistically significant ($p > 0.05$). Changes in candidate composition and downstream exposure allocation are possible explanations, alongside ordinary metric variation. More details are provided in Sections E.4 and G.2.

6 Conclusion

We presented *RED-Rec*, a hierarchical language-model framework for retrieval across feed, search, and advertising. Scenario-aware sequence mixing allocates history capacity across these contexts, and learnable queries produce multiple interest representations for retrieval. Item-aware training and maximum-similarity fusion allow the model to use candidate-specific supervision while retaining precomputed item embeddings at serving time. Within-backbone comparisons show gains over the backbone alone and adapted multi-interest readouts. The full mixer–query model performs best in the component comparison, with stronger interaction at the larger retrieval cutoff. The online test improves advertiser value and advertising spend. The results support jointly designing sequence construction and interest readout for multi-scenario retrieval.

AI USE STATEMENT

We used GPT-4.1 to generate natural-language rationales for the auxiliary experiment in Section H.2. AI assistance was also used for literature organization and language editing during manuscript preparation. The authors are responsible for the paper's content and conclusions.

ETHICS STATEMENT

We acknowledge the ICLR Code of Ethics. The dataset processing replaces user identifiers with hashes, shifts timestamps, and removes device and location metadata (Section C). Broader implications for privacy, exposure diversity, and advertising are discussed in Section I.

REPRODUCIBILITY STATEMENT

Model architecture and objectives are provided in Section 4, experimental settings and baselines in Section 5, and dataset processing, metrics, and implementation details in Sections C, G and G.2. Controlled comparisons, sensitivity sweeps, and diagnostics are documented in Sections E and 5.3; Table 6a reports results over five random seeds.

REFERENCES

Yukuo Cen, Jianwei Zhang, Xu Zou, Chang Zhou, Hongxia Yang, and Jie Tang. Controllable multi-interest framework for recommendation. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2020. 2, 7

Jia Chen, Qian Dong, Haitao Li, Xiaohui He, Yan Gao, Shaosheng Cao, Yi Wu, Ping Yang, Chen Xu, Yao Hu, et al. Qilin: A multimodal information retrieval dataset with app-level user sessions. *arXiv preprint arXiv:2503.00501*, 2025. 3

Junyi Chen, Lu Chi, Bingyue Peng, and Zehuan Yuan. Hllm: Enhancing sequential recommendations via hierarchical large language models for item and user modeling. *arXiv preprint arXiv:2409.12740*, 2024. 1, 2, 6

Jiaxin Deng, Shiyao Wang, Kuo Cai, Lejian Ren, Qigen Hu, Weifeng Ding, Qiang Luo, and Guorui Zhou. Onerec: Unifying retrieve and rank with generative recommender and iterative preference alignment. *arXiv preprint arXiv:2502.18965*, 2025. 2

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2021. 19

Yongkang Fu, Beining Bao, Yu Jiang, Xiangyu Zhao, Hongyang Wei, Guangxing Chen, Zuodong Yang, Shantao Li, Zonggang Wu, Yuqi Lu, Shouke Qin, Hanmeng Liu, and Maolin Wang. MuSeR: Scalable long-sequence recommendation with multi-interest modeling, 2026. URL https://arxiv.org/abs/2609.23677. 2

Chongming Gao, Shijun Li, Yuan Zhang, Jiawei Chen, Biao Li, Wenqiang Lei, Peng Jiang, and Xiangnan He. Kuairand: An unbiased sequential recommendation dataset with randomly exposed videos. In *ACM International Conference on Information and Knowledge Management*, 2022. 3

Ruidong Han, Bin Yin, Shangyu Chen, He Jiang, Fei Jiang, Xiang Li, Chi Ma, Mincong Huang, Xiaoguang Li, Chunzhen Jing, et al. Mtgr: Industrial-scale generative recommendation framework in meituan. *arXiv preprint arXiv:2505.18654*, 2025. 2

F Maxwell Harper and Joseph A Konstan. The movielens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems (TIIS)*, 5(4):1–19, 2015. 3

Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. Session-based recommendations with recurrent neural networks. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2016. 2

Wang-Cheng Kang and Julian McAuley. Self-attentive sequential recommendation. In *IEEE International Conference on Data Mining (ICDM)*, 2018. 2, 6

Chao Li, Zhiyuan Liu, Mengmeng Wu, Yuchi Xu, Huan Zhao, Pipei Huang, Guoliang Kang, Qiwei Chen, Wei Li, and Dik Lun Lee. Multi-interest network with dynamic routing for recommendation at tmall. In *ACM International Conference on Information and Knowledge Management*, 2019. 2, 7

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2023. 19

Qidong Liu, Xiangyu Zhao, Yejing Wang, Zijian Zhang, Howard Zhong, Chong Chen, Xiang Li, Wei Huang, and Feng Tian. Bridge the domains: Large language models enhanced cross-domain sequential recommendation. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1582–1592, 2025. 2, 6, 13

Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. Image-based recommendations on styles and substitutes. In *ACM SIGIR Conference on Research and Development in Information Retrieval*, 2015. 6, 13

Meta. Generative Recommenders: DLRM-v3 implementation. Software repository, 2026. URL https://github.com/meta-recsys/generative-recommenders. 6

Jianmo Ni, Jiacheng Li, and Julian McAuley. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Annual Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2019. 3

Xiangchen Pan, Wei Wei, Huakang Niu, and Zhicong Cheng. Multi interests for joint search-recommendation modeling, 2026. URL https://arxiv.org/abs/2608.10535. 1, 2

Qwen Team. Qwen2.5 technical report, 2024. URL https://arxiv.org/abs/2412.15115. 6, 9

Xiang-Rong Sheng, Liqin Zhao, Guorui Zhou, Xinyao Ding, Binding Dai, Qiang Luo, Siran Yang, Jingshan Lv, Chi Zhang, Hongbo Deng, et al. One model to serve all: Star topology adaptive recommender for multi-domain ctr prediction. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pp. 4104–4113, 2021. 1, 2

Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In *ACM International Conference on Information and Knowledge Management*, 2019. 2

Ruibing Wang, Shuhan Guo, Haotong Du, and Quanming Yao. Distillation-based scenario-adaptive mixture-of-experts for the matching stage of multi-scenario recommendation, 2026. URL https://arxiv.org/abs/2601.02368. 1, 2

Yuhao Wang, Yichao Wang, Zichuan Fu, Xiangyang Li, Wanyu Wang, Yuyang Ye, Xiangyu Zhao, Huifeng Guo, and Ruiming Tang. Llm4msr: An llm-enhanced paradigm for multi-scenario recommendation. In *ACM International Conference on Information and Knowledge Management*, 2024. 2

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 23

Bencheng Yan, Pengjie Wang, Kai Zhang, Feng Li, Hongbo Deng, Jian Xu, and Bo Zheng. Apg: Adaptive parameter generation network for click-through rate prediction. *Advances in Neural Information Processing Systems*, 35:24740–24752, 2022. 2

Xuanhua Yang, Xiaoyu Peng, Penghui Wei, Shaoguo Liu, Liang Wang, and Bo Zheng. Adasparse: Learning adaptively sparse structures for multi-domain click-through rate prediction. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pp. 4635–4639, 2022. 2

Zheng Yuan, Fajie Yuan, Yu Song, Youhua Li, Junchen Fu, Fei Yang, Yunzhu Pan, and Yongxin Ni. Where to go next for recommender systems? id-vs. modality-based recommender models revisited. In *ACM SIGIR Conference on Research and Development in Information Retrieval*, 2023. 6

Jiaqi Zhai, Lucy Liao, Xing Liu, Yueming Wang, Rui Li, Xuan Cao, Leon Gao, Zhaojie Gong, Fangda Gu, Jiayuan He, et al. Actions speak louder than words: Trillion-parameter sequential transducers for generative recommendations. In *Proceedings of International Conference on Machine Learning (ICML)*, 2024. 2, 6

Shengzhe Zhang, Liyi Chen, Dazhong Shen, Chao Wang, and Hui Xiong. Hierarchical time-aware mixture of experts for multi-modal sequential recommendation. In *Proceedings of the ACM on Web Conference 2025*, pp. 3672–3682, 2025. 2

Zijian Zhang, Shuchang Liu, Jiaao Yu, Qingpeng Cai, Xiangyu Zhao, Chunxu Zhang, Ziru Liu, Qidong Liu, Hongwei Zhao, Lantao Hu, et al. M3oe: Multi-domain multi-task mixture-of experts recommendation framework. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2024. 1, 2, 6

Jingyi Zhou, Cheng Chen, Kai Zuo, Manjie Xu, Zhendong Fu, Yibo Chen, Xu Tang, and Yao Hu. Hymirec: A hybrid multi-interest learning framework for llm-based sequential recommendation, 2025. URL https://arxiv.org/abs/2510.13738. 1, 2

A TERMS

We define the platform terminology used throughout the paper.

Homefeed refers to the main feed or landing page displayed to a user when they open a content platform or app. It typically consists of a personalized selection of items (such as posts, products, videos, etc.) recommended to the user based on their preferences and past behavior.

Internal Flow denotes the content consumption pattern within the single-column sliding or swiping through content (e.g., images, videos, or articles). Users engage with recommendations directly within this detailed view by navigating between related items or sliding to the next recommended content.

External Flow refers to the content consumption flow that occurs on the main feed of the platform, where users browse the list of recommended items presented to them upon opening the app. This process typically involves users scrolling vertically through the two-columns page.

Scenarios refer to distinct user interaction environments or channels within the platform, each characterized by unique user intents and behavioral patterns. In this paper, we focus on three core scenarios: **homefeed**, **ads**, and **search**. The homefeed scenario represents the primary personalized feed where users consume a diverse assortment of recommended content. The ads scenario corresponds to user engagement with sponsored or promotional content distributed throughout various parts of the platform. Although advertisement content can appear within the homefeed, we treat it as a separate scenario because it represents a different source and serves distinct business objectives. The search scenario involves users actively retrieving information or content by submitting queries.

last-n refers to the most recent 'n' items a user has interacted with on the platform. For example, 'last10' indicates the user's last 10 consumed items. This concept is commonly used to capture and analyze a user's most current interests or activity history.

Engage represents user interactions with content, such as clicks, likes, comments, shares, or dwell time. Engagement metrics are used to measure how users interact with recommended items and to assess the effectiveness of recommender systems.

B FURTHER EXPERIMENTS

We evaluate *RED-Rec* on Amazon Books (McAuley et al., 2015) and on the Clothing–Sports cross-domain setting used by LLM4CDSR (Liu et al., 2025). Results are shown in Tables 6a and 6b. Entries marked with an asterisk in Table 6b are taken from the LLM4CDSR paper; our runs follow the same split, candidate-pool construction, and negative-sampling protocol. The billion-parameter HLLM and Qwen2.5-1.5B models are of comparable scale, but not exactly parameter-matched. Within-backbone comparisons are reported separately in Table 4. For Amazon Books, each test example has one held-out target item. HR@K measures whether the target is retrieved within the top K, and NDCG@K discounts its rank. All cutoffs use the same candidate ranking.

On Amazon Books, the Qwen2.5 variant improves HR at all three cutoffs over HLLM, while HLLM retains the highest NDCG@200. In Clothing–Sports, LLM4CDSR has the strongest Clothing results, and the Qwen2.5 variant has the highest reported Sports HR@10 and NDCG@10. The results show that the relative gains depend on the domain and metric.

C DATASET

We compare the training dataset with public recommendation and search datasets in Table 7. The 300.6 million items refer to the offline training subset, not the platform-wide serving inventory. Production training uses the full platform data, and the serving environment contains approximately 1.1 billion active items.

A user record has the following structure:

Table 6: **Public benchmark results.** (a) Amazon Books; entries with $\pm$ report mean and standard deviation over five seeds, while MoRec and the LLaMA variant retain their point estimates. (b) Amazon Clothing–Sports; HR is on a $\times 100$ scale and NDCG is on $[0, 1]$. Bold marks the best reported mean in panel (a).

(a) Amazon Books.

Method	HR@10	NDCG@10	HR@50	NDCG@50	HR@200	NDCG@200
SASRec	3.06 ± 0.09	1.64 ± 0.05	7.54 ± 0.14	2.60 ± 0.08	14.31 ± 0.22	3.62 ± 0.11
MoRec (BERT)	3.21	1.82	8.21	3.11	18.31	4.48
HSTU-1B	4.78 ± 0.12	2.62 ± 0.07	10.82 ± 0.18	3.93 ± 0.10	19.08 ± 0.28	5.17 ± 0.13
DLRM-v3-1B	6.22 ± 0.13	3.46 ± 0.07	12.74 ± 0.21	5.12 ± 0.12	23.12 ± 0.31	6.87 ± 0.21
HLLM-1B	9.28 ± 0.16	5.36 ± 0.31	17.34 ± 0.25	7.41 ± 0.15	27.22 ± 0.36	$\mathbf{8.89} \pm 0.18$
RED-Rec-1.1B-LLaMA	9.21	5.13	18.63	7.03	28.72	8.41
RED-Rec-1.5B-Qwen2.5	$\mathbf{9.34} \pm 0.14$	$\mathbf{5.45} \pm 0.24$	$\mathbf{19.98} \pm 0.23$	$\mathbf{7.43} \pm 0.14$	$\mathbf{29.87} \pm 0.35$	8.88 ± 0.21

(b) Amazon Clothing–Sports cross-scenario setting.

Method	HR_{10} (Cloth)	$NDCG_{10}$ (Cloth)	HR_{10} (Sport)	$NDCG_{10}$ (Sport)
SASRec*	70.57	0.6543	59.64	0.4900
Bert4Rec*	65.31	0.5720	53.50	0.4514
AMID*	74.75	0.6814	63.77	0.5867
STAR	62.36	0.5243	51.22	0.4724
M2M	65.27	0.5362	52.08	0.4626
M3oE	72.40	0.7010	62.16	0.6208
LLM4CDSR*	80.18	0.7316	70.46	0.6312
RED-Rec-1.1B-LLaMA	76.28	0.6821	66.28	0.5824
RED-Rec-1.5B-Qwen2.5	78.25	0.7214	71.12	0.6318

Table 7: Comparison of public datasets and the industrial training dataset.

Property	**Amazon**	**JD Search**	**KuaiSAR**	**Qilin**	***RED-Rec*(dataset)**
Users	192.4k	173.8k	25.8k	15.5k	1.0m
Items	63.0k	12.9m	6.9m	2.0m	300.6m
Queries	3.2k	171.7k	453.7k	571.9k	search items
Actions	1.7m	26.7m	19.7m	2.5m	683.2m
Content	text/image	text	text/video	text/image/video	text/image/video
Scenario	Rec	Search	Search+Rec	Search+Rec	Search+Rec+Ads

Listing 1: Example of a user record in the training dataset.


```
{
  "user_id": "xxxx",
  "data": {
    "homefeed_item_lastn": [
      {
        "duration": 28,
        "is_click": 1,
        "is_click_profile": 0,
        "is_collect": 0,
        "is_comment": 0,
        "is_follow": 0,
        "is_hide": 0,
        "is_like": 0,
        "is_nns": 0,
        "is_pagetime": 1,
        "is_read_comment": 1,
        "is_share": 0,
        "is_videoend": 0,
        "item_id": "684a48440000000023014319",
        "page_key": 0,
        "timestamp": 1749771247,
        "type": "note"
      },
      {
        "duration": 17,
        "is_click": 1,
        "is_click_profile": 0,
        "is_collect": 0,
        ...
        "is_pagetime": 1,
        "is_read_comment": 0,
        "is_share": 0,
        "is_videoend": 0,
```

```
        "item_id": "684aa07200000000021003dbe",
        "page_key": 0,
        "timestamp": 1749732355,
        "type": "note"
      }
      // ...
    ],
    "ads_item_lastn": [ ... ],
    "search_item_lastn": [ ... ]
  }
}
```


where

- **user_id**: Unique identifier for the user, e.g., `xxxx`.
- **data**:
  - **homefeed_item_lastn**: An array of objects representing the last n items from the user's home feed. Each object contains:
    - `duration`: Viewing duration (in seconds).
    - `is_click`: Whether the item was clicked (1) or not (0).
    - `is_click_profile`: Whether the user's profile was clicked (1 or 0).
    - `is_collect`: Whether the item was collected or saved (1 or 0).
    - `is_comment`: Whether the item was commented on (1 or 0).
    - `is_follow`: Whether the user followed from this item (1 or 0).
    - `is_hide`: Whether the item was hidden (1 or 0).
    - `is_like`: Whether the item was liked (1 or 0).
    - `is_message`: Whether the author of the message was messaged (1 or 0).
    - `is_pagetime`: Whether the page time event was triggered (1 or 0).
    - `is_read_comment`: Whether comments were read (1 or 0).
    - `is_share`: Whether the item was shared (1 or 0).
    - `is_videoend`: Whether a video was watched until the end (1 or 0).
    - `item_id`: Identifier for the content item.
    - `page_key`: Page identifier.
    - `timestamp`: Timestamp of the interaction.
    - `type`: Type of item, e.g., `note`.
  - **ads_item_lastn**: Array of the last n interacted advertisement items (`item_id`, `duration`, etc.).
  - **search_item_lastn**: Array of the last n search items with similar structure.

We replace user identifiers with hashes, apply a consistent timestamp offset, and remove device information, location data, and detailed content descriptors. The retained fields describe the behavioral signals used for recommendation.

We focus exclusively on active platform users who demonstrate substantial engagement patterns: users must have at least 30 valid clicks in the homefeed scenario and 5 valid clicks in the advertisement scenario, where a click is considered valid only if the associated viewing duration exceeds 5 seconds.

D Component Effects and Their Interaction

The factorial comparison in Table 3b isolates the mixer and MoE under the same backbone and target scenario. We quantify their interaction as

$$I = (D - C) - (B - A), \tag{9}$$

where $B - A$ is the MoE gain without mixing and $D - C$ is the corresponding gain with mixing. Thus, $I > 0$ indicates that MoE provides a larger improvement when the mixer is enabled.

MoE improves all four metrics both with and without the mixer, but the gains are consistently larger when mixing is enabled. The interaction is most pronounced at cutoff 100: the MoE gain in HR@100 increases from 0.18 to 2.09 percentage points, yielding an interaction of 1.91 points; NDCG@100

Table 9: **Representation diagnostics on 10,000 test users.** (a) Attention entropy and sparse-token coverage before and after mixing. (b) Expert retrieval, routing, and representation statistics; item cosine measures cohesion rather than category-label purity.

(a) Attention by same-scenario run length.

	$H(a)$		R_{sparse}	
L_{run}	No mixer	Mixer	No mixer	Mixer
20	3.42	3.71	0.71	0.92
50	2.28	3.58	0.44	0.89
100	1.53	3.49	0.29	0.87

(b) Expert diagnostics.

Diagnostic	Value
Average pairwise retrieval Jaccard@100	0.18
Normalized routing entropy	0.42
Average within-expert item cosine	0.71
Average between-expert item cosine	0.29
Cross-seed routing Kendall's τ (3 seeds)	0.79
Maximum/minimum expert utilization	1.42

shows a similar pattern ($I = 0.59$). At cutoff 10, the interactions remain positive but smaller. Overall, the full model raises HR@10 from 4.28 to 5.13, corresponding to a 19.9% relative improvement. These results indicate complementary effects between the two components, particularly at broader retrieval cutoffs. The reported interactions are descriptive contrasts of point estimates rather than statistical tests of independence.

E Sensitivity and Representation Diagnostics

E.1 Hyperparameter Sensitivity

Unless specified otherwise, these sweeps use Qwen2.5-1.5B, history length 128, and Homefeed + Search + Ads histories for Ads retrieval. The default setting has quotas 64/32/32, block size $b = 4$, five queries per scenario, and gate coefficient $\alpha = 0.5$. Each sweep varies one setting.

Table 8: **Component gains and interaction.** Differences are in percentage points, computed from Table 3b.

Metric	$B - A$	$D - C$	I
HR@10	0.23	0.41	0.18
NDCG@10	0.11	0.15	0.04
HR@100	0.18	2.09	1.91
NDCG@100	0.07	0.66	0.59

The quota sweep in Table 10 shows the effect of allocating history capacity. Relative to 2:1:1, the ratios 4:1:1 and 1:1:1 reduce HR@10 by 4.1% and 1.8%, respectively. Allocating less capacity to Homefeed at 1:2:2 reduces it by 47.2%. The selected ratio is therefore a property of this data distribution, and substantial changes in scenario frequency call for retuning.

Block sizes 2 and 8 give HR@10 of 5.09 and 5.06, compared with 5.13 at $b = 4$, a maximum relative change of 1.4%. Varying α from 0.3 to 0.7 changes HR@10 by at most 2.1% relative to the default. Increasing the query count from five to seven gives no further gain. With a fixed total budget of 15 queries, balanced allocation across scenarios yields 5.13, compared with 5.04–5.06 for the tested skewed allocations and 4.91 for a shared query pool.

E.2 Attention and Same-Scenario Runs

We analyze attention on 10,000 test users. Let a_j be attention normalized over the n historical tokens. We measure attention entropy $H(a) = -\sum_{j=1}^{n} a_j \log a_j$ and the fraction of sparse-scenario tokens receiving more than uniform attention:

$$R_{\text{sparse}} = \frac{1}{|\mathcal{J}_{\text{sparse}}|} \sum_{j \in \mathcal{J}_{\text{sparse}}} \mathbb{I}[a_j > 1/n], \tag{10}$$

where $\mathcal{J}_{\text{sparse}}$ contains the Ads and Search history positions. Table 9a groups the analysis by the longest same-scenario run in the unmixed input.

Without mixing, entropy and sparse-token coverage decrease as runs become longer. With the mixer, both quantities vary less across the reported run lengths. At $L_{\text{run}} = 100$, sparse-token coverage increases from 0.29 to 0.87. This diagnostic connects the sequence construction to the distribution of attention over the history.

E.3 Expert Retrieval and Representation Statistics

Table 9b reports diagnostics on 10,000 test users. For expert retrieval sets $\mathcal{R}_e^{100}$, pairwise overlap is $|\mathcal{R}_e^{100} \cap \mathcal{R}_{e'}^{100}|/|\mathcal{R}_e^{100} \cup \mathcal{R}_{e'}^{100}|$, averaged over expert pairs and users. Normalized routing entropy divides the entropy over expert weights by $\log E$. We also compare item cosine similarity within and across expert assignments.

The retrieved sets overlap relatively little, while within-expert item similarity is higher than between-expert similarity. The utilization ratio indicates that the reported diversity is accompanied by usage of multiple experts. These statistics characterize the learned representation and retrieval behavior alongside the attention examples in Figure 5.

Table 10: **Sensitivity of Ads HR@10.** Quota ratios are ordered Homefeed:Ads:Search. The expert-count sweep uses the 2-D mixer throughout. Bold marks the best value within each sweep.

Parameter	Setting	HR@10
Scenario quota ratio	4:1:1	4.92
	2:1:1	**5.13**
	1:1:1	5.04
	1:2:2	2.71
Block size b	2	5.09
	4	**5.13**
	8	5.06
Queries per scenario E	1	4.72
	3	4.94
	5	**5.13**
	7	5.11
Gate coefficient α	0.3	5.02
	0.5	**5.13**
	0.7	5.08

E.4 Routing and Serving Diagnostics

Supervision and fusion. For the evaluation-time fusion comparison, we hold the hybrid-supervision checkpoint, user vectors, item embeddings, and approximately one-million-item candidate pool fixed. With $s_e(u,i) = \cos(\mathbf{u}_e, \mathbf{z}_{i,e})$, the two scores are

$$s_{\max}(u,i) = \max_e s_e(u,i), \qquad s_{\text{soft}}(u,i) = \sum_e \frac{\exp(s_e(u,i)/\tau_f)}{\sum_{e'} \exp(s_{e'}(u,i)/\tau_f)} s_e(u,i). \tag{11}$$

We set the fusion temperature to $\tau_f = 0.1$. This similarity-weighted soft fusion is distinct from the training-time hybrid router, which also uses an item-content gate. Timing is measured on an NVIDIA L20 with $E = 5$ and the same one-million-item candidate pool. User representations and item embeddings are precomputed. The measured interval includes five expert-wise cosine scoring passes, Max or Soft fusion, and Top-K selection; it excludes User LLM encoding, cache I/O, and ANN retrieval.

The hard-matching training variant selects the expert with the highest positive-item cosine similarity:

$$e^* = \arg\max_{e \in \{1,\dots,E\}} \cos(\mathbf{u}_e, \mathbf{z}_{i,e}), \qquad \mathcal{L}_{\text{NCE}}^{\text{hard}} = \ell_{\text{NCE}}(\mathbf{u}_{e^*}, \mathbf{z}_{i,e^*}). \tag{12}$$

The assignment is stop-gradient, while the selected expert receives gradients from its NCE loss. This replaces only the routing-weighted item-level NCE term; group-level Hungarian supervision, the regularizers, other training settings, and positive and negative sampling are unchanged from hybrid gating. The selection is performed within the target scenario's bank. This per-item hard top-1 assignment is distinct from the query–group Hungarian matching retained in both settings. Consequently, this comparison isolates item-level credit assignment, not the contribution of group-level matching.

Table 11a compares hybrid gating, similarity-only gating, content-only gating, and hard top-1 matching with max fusion, followed by hybrid gating with soft fusion. Hybrid gating with max fusion achieves the highest HR@10 and NDCG@10 among these settings. With max fusion, its HR@10 exceeds similarity-only, content-only, and hard matching by 0.11, 0.27, and 0.19 percentage points, respectively. All four settings have a measured p95 latency of 15.2 ms at the reported precision.

For hybrid supervision, max fusion yields HR@10 of 5.13 and NDCG@10 of 1.48, compared with 5.01 and 1.43 for soft fusion. Its measured p95 latency is 0.9 ms lower (15.2 versus 16.1 ms). These results favor hybrid supervision with max fusion in the measured setting; soft fusion does not provide a higher-accuracy reference here.

Table 11: **Routing and serving diagnostics.** (a) Controlled supervision and fusion comparison; HR and NDCG are on a $\times 100$ scale. Latency covers scoring, fusion, and Top-K on an L20 with $E = 5$ and one million candidates, excluding user encoding, cache I/O, and ANN retrieval. The hybrid-gate rows share a checkpoint, with $\tau_f = 0.1$ for soft fusion. (b) Offline-to-online HR@10 change by routing concentration; decreases are relative and do not form a same-candidate reranking ablation.

(a) Training supervision and evaluation fusion.

Training supervision	Fusion	HR@10	NDCG@10	Scoring + Top-K p95 (ms)
Hybrid gate	Max	**5.13**	**1.48**	**15.2**
Similarity only ($\alpha = 0$)	Max	5.02	1.44	**15.2**
Content only ($\alpha = 1$)	Max	4.86	1.38	**15.2**
Hard top-1 matching	Max	4.94	1.41	**15.2**
Hybrid gate	Soft	5.01	1.43	16.1

(b) Deployment change by routing concentration.

Top-1 routing mass	User share	Relative decrease
> 0.85	62%	2.1%
0.55–0.85	30%	4.3%
< 0.55	8%	5.8%

Score agreement and deployment diagnostics. Training uses item-aware expert weights, whereas serving merges the expert retrieval sets by maximum similarity. The reported score correlation between the two scoring rules is 0.94 on positive items, and Top-10 candidate Jaccard is 0.88. These measurements describe score agreement and candidate overlap.

The deployment comparison also reports relative HR@10 decreases by routing concentration (Table 11b). Users with more distributed routing have a larger measured decrease. This comparison spans offline evaluation and online retrieval, so its changes can reflect candidate availability and other serving conditions as well as the fusion rule.

F METRICS

We report Hit Ratio (HR), Normalized Discounted Cumulative Gain (NDCG), and Mean Reciprocal Rank (MRR). HR measures whether relevant items are retrieved; NDCG and MRR also account for their rank.

Hit Ratio (HR) Let $\mathcal{G}_u$ be the set of held-out relevant items for user u, and $\mathcal{R}_u^K$ the top-K ranked candidates. Hit Ratio uses a user-level any-hit definition:

$$\mathrm{HR@}K = \frac{1}{N}\sum_{u=1}^{N} \mathbb{I}(\mathcal{G}_u \cap \mathcal{R}_u^K \neq \varnothing). \tag{13}$$

Each user contributes at most one hit, irrespective of how many held-out items are retrieved. This differs from averaging recall over positive items when $|\mathcal{G}_u| > 1$; the two coincide for a single held-out positive.

HR@K measures retrieval success without distinguishing the positions of hits within the top-K.

Normalized Discounted Cumulative Gain (NDCG) Normalized Discounted Cumulative Gain (NDCG@K) extends HR@K by accounting for the position of relevant items, rewarding items that are ranked higher in the recommended list. For each test case, DCG is computed as:

$$\mathrm{DCG}_u\mathrm{@}K = \sum_{j=1}^{K} \frac{\mathbb{I}[\mathcal{R}_{u,j} \in \mathcal{G}_u]}{\log_2(j+1)}, \tag{14}$$

where $\mathcal{R}_{u,j}$ is the candidate at rank j. With binary relevance, $\mathrm{IDCG}_u\mathrm{@}K = \sum_{j=1}^{\min(K,|\mathcal{G}_u|)} 1/\log_2(j+1)$. Normalizing per user gives

$$\mathrm{NDCG@}K = \frac{1}{N}\sum_{u=1}^{N} \frac{\mathrm{DCG}_u\mathrm{@}K}{\mathrm{IDCG}_u\mathrm{@}K}. \tag{15}$$

NDCG rewards relevant items near the top of the list and normalizes for the number of relevant items.

Mean Reciprocal Rank (MRR) Mean Reciprocal Rank (MRR@K) evaluates how highly the first relevant item is ranked, and is defined as:

$$\mathrm{MRR@}K = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{\mathrm{rank}_i}, \tag{16}$$

where rank_i is the position of the first relevant item in the recommended list for user i, and set to infinity (i.e., reciprocal rank is 0) if no relevant items are found in the top-K. MRR@K emphasizes early precision, heavily rewarding algorithms that surface the relevant item at or near the top. Its sensitivity to the first relevant item's position makes it particularly apt for settings prioritizing immediate relevance (e.g., question answering, search).

Evaluation Protocols and Cutoff Values The industrial and Amazon Books tables report HR, NDCG, and MRR on a $\times 100$ scale. The Clothing–Sports table follows the source convention: HR is on a $\times 100$ scale and NDCG is on $[0, 1]$. Candidate sampling and temporal splitting are described in Section G.2.

G Implementation Details

G.1 Model Implementation

We provide additional implementation details of the proposed *RED-Rec*.

Item Encoder. We tokenize each item's title and text content and pass them to the Item LLM along with a learned embedding token. The token's output state gives the item representation, which is mapped into the retrieval space through projection heads.

We encode images with a ViT backbone (Dosovitskiy et al., 2021) and project its outputs to the Item LLM's input dimension, following the general vision–language alignment approach of LLaVA (Liu et al., 2023). The Item LLM then processes text and visual features jointly.

User Encoder. The User LLM processes item embeddings arranged by the scenario-aware mixer. Query output states are projected to interest vectors for retrieval. This representation allows different portions of the same history to contribute to different candidate sets.

Exact block interleaving. Let J_s be the number of blocks from scenario s and $t_{s,r}$ the endpoint timestamp of its r-th block. A state $F(\mathbf{r}, s)$ records the minimum objective after consuming r_s blocks from each scenario and ending in scenario s. With $k = \sum_v r_v$, the recurrence is

$$F(\mathbf{r}, s) = \frac{|t_{s,r_s} - t^{\uparrow}(k)|}{JT} + \min_v \left\{ F(\mathbf{r} - \mathbf{1}_s, v) + \frac{\beta\, \mathbb{I}[v = s]}{J - 1} \right\}. \tag{17}$$

Only states with $r_s > 0$ and reachable predecessors are considered. Initialize an empty state with cost zero and a sentinel last scenario that incurs no repetition cost; other states have infinite initial cost. Process states in increasing k, choose the least-cost terminal state $\min_s F((J_v)_v, s)$, and backtrack to recover the block order. The zero-span and single-block conventions of Equation (2) also apply to the recurrence. For an empty history, no block ordering is needed. Exactness follows because the consumed-block counts determine the next block and its output position, while the last scenario determines the repetition cost. The state space has $|\mathcal{S}| \prod_s (J_s + 1)$ entries and at most $|\mathcal{S}|$ predecessor choices per state.

Attention visibility and sequence budget. For the token groups $(H, Q_{\mathrm{hf}}, Q_{\mathrm{sr}}, Q_{\mathrm{ads}})$, the blockwise visibility matrix is

$$V = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}, \qquad M_{jk} = \begin{cases} 0, & \text{if token } j \text{ may attend to token } k, \\ -\infty, & \text{otherwise.} \end{cases} \tag{18}$$

Rows identify readers and columns identify the tokens they can read; a block value of one permits all valid token pairs in that block. We add M to the attention logits in every layer and mask padding keys. History representations do not depend on any query bank, preventing indirect communication between banks through the history. Queries within a bank are mutually visible, without a causal ordering among them.

The default budget of 128 counts historical interactions; scenario tokens and query tokens are additional. With $E = 5$, mixed-scenario training uses 15 query tokens, while a single-scenario serving request uses five. Position IDs for the retained history and query bank are unchanged when other banks are omitted. Thus, with stochastic training operations disabled, joint-bank and single-bank evaluation have the same retained computation, up to numerical differences.

All input interactions precede the prediction cutoff; future target items are used only in the loss and router, not inserted into the User LLM input. Bidirectional history attention therefore uses only information available at the cutoff. Supervision is applied to the query readouts for future targets, rather than as per-position next-item losses over a bidirectionally visible history.

Routing-weighted supervision. For a positive target i sampled from the future window and a negative set $\mathcal{N}$, the loss for expert e is

$$\ell_{\mathrm{NCE}}(\mathbf{u}_e, \mathbf{z}_{i,e}) = -\log \frac{\exp(\cos(\mathbf{u}_e, \mathbf{z}_{i,e})/\tau)}{\exp(\cos(\mathbf{u}_e, \mathbf{z}_{i,e})/\tau) + \sum_{j\in\mathcal{N}} \exp(\cos(\mathbf{u}_e, \mathbf{z}_{j,e})/\tau)}. \tag{19}$$

The hybrid gate weights these item-level losses across the E experts of the target scenario, not across all $3E$ queries. Within that bank, experts are supervised against the same target with different weights determined by item content and user–item similarity. Mixed-scenario batches aggregate these target-specific losses through the shared backbone. If several scenarios have valid targets for the same user and cutoff, their corresponding banks can be supervised together; banks without a target contribute no target-specific loss.

Group-level Hungarian supervision. Only in Stage II, we apply spherical k-means to frozen positive item embeddings in the future window ($W = 10$), separately for each target scenario. For n^+ positives, we use $K = \min(E, n^+)$ nonempty clusters ($E = 5$ by default), skipping this term when $n^+ = 0$. Let $\mathbf{c}_{e,k}$ denote cluster k's centroid in expert e's projected item space. Hungarian matching minimizes cosine distance over one-to-one assignments of the K groups to distinct queries:

$$C_{e,k} = 1 - \cos(\mathbf{u}_e, \mathbf{c}_{e,k}), \qquad \mathcal{M}^* = \arg\min_{\mathcal{M}} \sum_{(e,k)\in\mathcal{M}} C_{e,k}. \tag{20}$$

The assignment is stop-gradient; matched pairs are trained with the same NCE form and item negatives as the item-level objective:

$$\mathcal{L}_{\mathrm{match}} = \frac{1}{|\mathcal{M}^*|} \sum_{(e,k)\in\mathcal{M}^*} \ell_{\mathrm{NCE}}(\mathbf{u}_e, \mathbf{c}_{e,k}). \tag{21}$$

Thus, cosine distance selects the assignment and NCE trains the queries. These per-user future-window centroids are distinct from the batchwise soft centroids in $\mathcal{L}_{\mathrm{spec}}$; neither is inserted into the User LLM input.

Loss Balancing and Training Dynamics. The objectives for the two stages are

$$\begin{aligned} \mathcal{L}_{\mathrm{I}} &= \mathcal{L}_{\mathrm{NCE}} + \lambda_{\mathrm{div}}\mathcal{L}_{\mathrm{div}} + \lambda_{\mathrm{bal}}\mathcal{L}_{\mathrm{bal}}, \\ \mathcal{L}_{\mathrm{II}} &= \mathcal{L}_{\mathrm{I}} + 0.1\mathcal{L}_{\mathrm{match}} + \lambda_{\mathrm{spec}}\mathcal{L}_{\mathrm{spec}}. \end{aligned} \tag{22}$$

We use $\lambda_{\mathrm{bal}} \in [0.1, 0.5]$ and $\lambda_{\mathrm{spec}}, \lambda_{\mathrm{div}} \approx 0.01$, with a linear warmup of λ_{bal} over the first 5% of training steps. Stage I jointly aligns item and user encoders. Stage II freezes the item encoder and projection heads, adding group matching and specialization while retaining item-level soft-routing NCE.

G.2 Online Deployment Optimization

We describe item caching, user inference, and the offline evaluation protocol.

Item-side We precompute item embeddings and store them in a key-value (KV) database. Existing items are re-encoded every 7 days using 64 H800 GPUs; new items are indexed daily using 5 L20 GPUs. Updates run asynchronously. The most frequently clicked 20% of items account for 99% of user clicks, so caching their embeddings avoids repeated content encoding.

User-side For each request, we retrieve the embeddings of the user's recent interactions from the item cache and pass them to the User LLM. Requests are batched for GPU execution. For active users, cached user representations are updated as new behavior arrives. The user-side service uses 20 L20 GPUs.

Training uses $W(i, e)$ to distribute the contrastive loss across experts. At inference, each query searches the corresponding item index, and candidates are merged using $\max_e \cos(\mathbf{u}_e, \mathbf{z}_{i,e})$. The user representation is computed independently of the candidate items.

Users with little history. For users with fewer than approximately five recorded interactions, serving falls back to a popularity-based candidate pool with scenario-conditioned priors. This fallback is separate from the active-user evaluation cohort. Content-based item embeddings support scoring items without a learned ID embedding, but do not supply preferences for a user with no observed history.

Numerical Precision and Serving User-side inference uses bfloat16 arithmetic to reduce memory use, and cached item embeddings are stored to six decimal places. We use ONNX Runtime[1] to optimize the serving graph. Serving nodes are stateless and obtain item embeddings and user state from distributed caches and databases. Invalid items are replaced by padding so that the input structure remains consistent.

Benchmarking and Offline Evaluation We evaluate representations on recall tasks using a temporal split protocol. To avoid information leakage, the test set is independently sampled after the training period, i.e., all evaluation interactions occur strictly after the training data (no temporal crossover). We sample 10,000 test users uniformly from the qualified active-user cohort: at least 30 valid Homefeed clicks and 5 valid Ads clicks, with viewing duration above 5 seconds. Uniform sampling is within this cohort; the cohort itself selects users with histories in both scenarios.

For each evaluated user, the input history ends at a temporal cutoff, and subsequent interactions in the target scenario define the held-out positive set $\mathcal{G}_u$. We rank approximately one million sampled candidate items, drawn from the much larger platform inventory, rather than ranking the entire inventory. This evaluation pool is distinct from the source pool for training negatives. Candidates are drawn from items active during the evaluation period. HR@10 is a user-level any-hit percentage: 3.17 means that at least one held-out positive appears among the ten highest-ranked candidates for 3.17% of evaluated users. NDCG uses all held-out positives and MRR uses the first relevant rank, as defined in Section F.

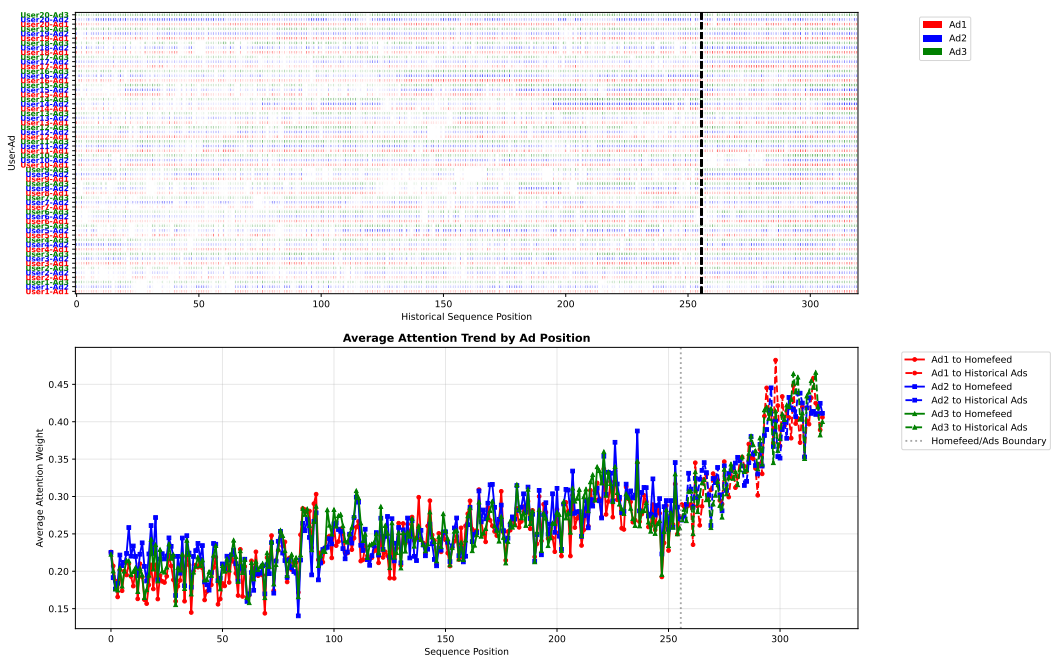
Since *RED-Rec* generates multiple interest embeddings $\{\mathbf{u}_1, \ldots, \mathbf{u}_E\}$ through its MoE architecture, we adopt a **multi-interest retrieval scoring strategy** during evaluation. For a candidate item i with its expert-specific embeddings $\{\mathbf{z}_{i,1}, \ldots, \mathbf{z}_{i,E}\}$, the final similarity score is defined as the maximum similarity across all experts:

$$\text{score}(u, i) = \max_{e \in \{1,\ldots,E\}} \cos(\mathbf{u}_e, \mathbf{z}_{i,e}). \tag{23}$$

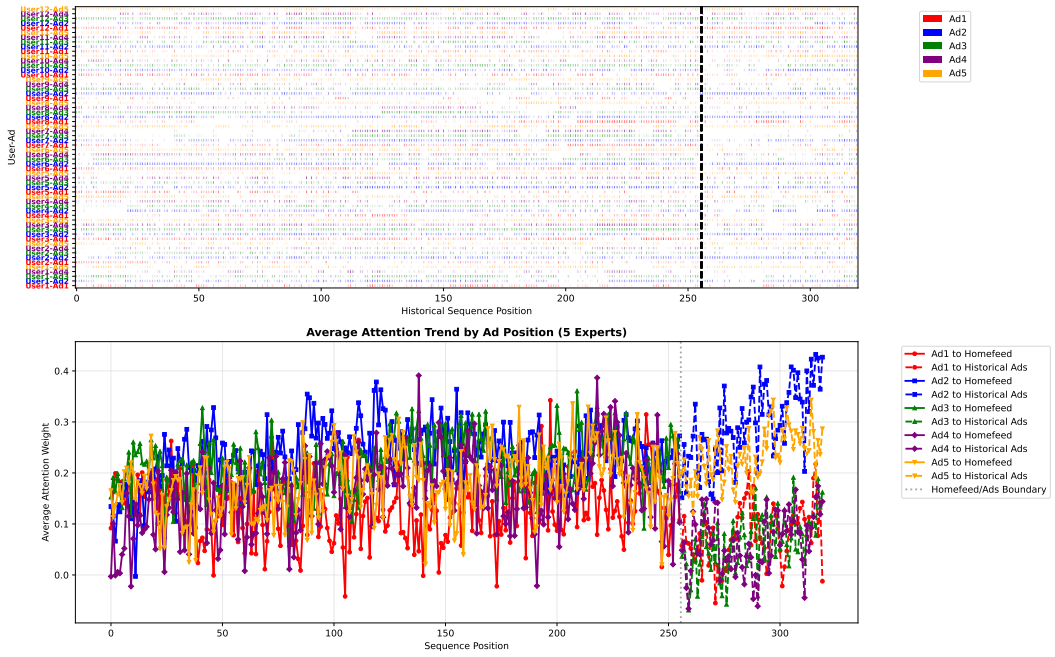
This scoring function simulates the real-world retrieval process where a target item only needs to match *one* of the user's potential interests to be recalled. Items in the candidate pool are ranked by this score to produce recommendations $\mathcal{R}_u$, and top-K results are evaluated using HR@K, NDCG@K, and MRR for $K \in \{10, 50, 100, 1000\}$.

Online Tradeoff The decreases in purchase UV and live-broadcast engagement UV are not statistically significant ($p > 0.05$), while advertiser value and advertising spend improve with $p < 0.01$ (Table 5). A change in recalled candidates can redistribute exposure through the downstream ranker; this is a possible explanation for non-target metric movements rather than a measured causal effect. The method has been deployed in production.

[1] https://github.com/microsoft/onnxruntime



(a) Without MoE.



(b) With query-based MoE (ours).

Figure 5: **Attention visualization of user encoding with and without MoE.** Without MoE, the User LLM tends to allocate most attention mass to a few most recent clicks, indicating short-term dominance. With MoE, different expert queries attend to different parts of the multi-scenario history.

H Additional Model Studies

H.1 Pretraining Validation

Table 12 compares pretrained Qwen encoders with variants that initialize either encoder from scratch or keep it frozen during training.

Table 12: Ablation results for different combinations of item and user LLMs and training strategies.

Scenario	Configuration	HR/NDCG$_{10}$	HR/NDCG$_{100}$	HR/NDCG$_{1k}$	MRR$_{*100}$
Homefeed	*RED-Rec* (2 * Qwen)	2.33/1.27	12.54/2.02	31.99/3.87	1.27
	Item LLM from scratch	0.00/0.00	0.00/0.00	0.03/0.01	0.00
	User LLM from scratch	0.00/0.00	0.03/0.01	1.32/0.21	0.01
	Item LLM frozen	1.27/0.36	5.51/0.77	11.37/1.02	0.37
	User LLM frozen	1.78/0.44	10.47/1.02	23.06/1.48	1.01
Homefeed + Ads	*RED-Rec* (2 * Qwen)	4.36/1.31	18.32/3.27	42.61/5.02	2.11
	Item LLM from scratch	0.00/0.00	0.00/0.00	0.08/0.04	0.01
	User LLM from scratch	0.00/0.00	0.00/0.00	1.01/0.07	0.03
	Item LLM frozen	1.49/0.41	9.49/1.31	19.29/1.52	0.76
	User LLM frozen	2.57/1.01	13.72/1.98	29.72/2.41	3.28
Homefeed + Ads	*RED-Rec* (2 * Qwen)	4.36/1.31	18.32/3.27	42.61/5.02	2.11
	RED-Rec-CoT (2 * Qwen)	4.46/1.35	18.78/3.60	44.61/5.01	2.15

The jointly trained pretrained encoders obtain the highest HR at all reported cutoffs in both settings. Training either encoder from scratch yields much lower HR, and freezing either encoder also reduces HR. The frozen-user variant has a higher MRR in Homefeed + Ads, showing that the ranking metrics need not move together.

H.2 CoT Validation

We study an auxiliary rationale-generation objective using Chain-of-Thought (CoT) targets (Wei et al., 2022). A language-model head predicts a rationale for an interaction conditioned on the user representation:

$$\mathcal{L}_{\text{CoT}} = -\sum_{u\in\mathcal{U}}\sum_{t=1}^{|S_u|}\sum_{\ell=1}^{L_t}\log p_\phi(r_{t,\ell} \mid r_{t,<\ell}, \mathbf{z}_{u,t}), \tag{24}$$

where $p_\phi()$ denotes the probability, computed by a learnable language model head parameterized by ϕ, of generating the ℓ-th token $r_{t,\ell}$ of the rationale conditioned on the previous tokens $r_{t,<\ell}$ and the contextualized user embedding $\mathbf{z}_{u,t}$ at interaction t. The overall training loss is then $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{NCE}} + \lambda_{\text{CoT}}\mathcal{L}_{\text{CoT}}$.

We use GPT-4.1 to generate the training rationales. One example is:

"The user browsed multiple articles related to Switzerland on the homepage, such as "Do you dare to guess how many days of sunshine in Switzerland?" and "What to wear for a trip to Switzerland next week?" This indicates a clear interest in Switzerland. While previously recommended ads included those related to travel, they were not specifically targeted at Switzerland.

Therefore, we recommend to the user the targeted ad "Personal tested and useful! The ultimate transportation ticket map tool for traveling in Switzerland!", as well as other ads related to traveling in Switzerland, such as "Countdown to opening! The four legendary theme parks of Fiesch First Mountain" and "Interlaken sledding premium tips — Save 400 RMB instantly".

The CoT variant improves HR@10 from 4.36 to 4.46 and HR@1000 from 42.61 to 44.61 in the Homefeed + Ads setting (Table 12). Rationale generation adds training cost, so this experiment is limited to a smaller-scale setting. The generated text provides an auxiliary training signal; the main retrieval model uses embedding similarity.

H.3 Interpretability via Attention Maps.

The attention maps in Figure 5 compare a standard readout with query-based readouts. The standard model concentrates on a few recent clicks, whereas the query outputs attend to different portions of the history. This visualization complements the query-count ablation by showing how several readouts distribute attention over the same input.

I Limitations and Broader Impact Statement

I.1 Limitations

The current design uses fixed scenario quotas and a fixed query budget. The quota sweep shows a substantial accuracy loss when history capacity is shifted away from Homefeed, motivating allocation that adapts to the activity distribution. The industrial evaluation also selects active users with feed and advertisement histories; users with little activity require separate evaluation. Larger backbones improve retrieval accuracy at a throughput cost, and rationale supervision adds training overhead. The expert diagnostics measure retrieval diversity and embedding cohesion; the semantic meaning of each query remains implicit. The offline-to-online comparison includes serving differences beyond the fusion rule; the fixed-candidate comparison isolates the latter.

I.2 Broader Impact Statement

RED-Rec combines behavioral evidence across services to improve retrieval. Such integration can also increase the influence of commercial objectives on content exposure. We therefore report engagement metrics alongside advertising outcomes (Table 5). Exposure diversity and the distribution of benefits across user groups remain relevant considerations for deployment.

Combining histories across scenarios increases the sensitivity of behavioral data. Our processing removes selected identifying fields, hashes user identifiers, and shifts timestamps (Section C). These measures reduce direct identification but do not by themselves eliminate the possibility of linking behavioral patterns to individuals.