MiRec: Multi-scale Interest Decomposition and Mixing for Sequential Recommendation

Anonymous authors
Paper under double-blind review

Abstract

Sequential recommendation predicts the next item a user will interact with by modeling the evolution of their interests from historical interactions. Multi-scale modeling captures user interests over different historical ranges but still faces two challenges. At the intra-scale level, stable and dynamic interests are entangled in holistic representations, making them difficult to model separately. At the cross-scale level, fusing entire representations does not adequately distinguish the mixing requirements of these two types of interests, limiting the targeted integration of complementary information. To address these challenges, we propose **MiRec**, a multi-scale interest decomposition and mixing model. Specifically, intra-scale interest decomposition extracts stable and dynamic interest representations through causal smoothing and residual computation. The cross-scale interest mixer then integrates representations of the same type from different scales through learnable subspace exchange. Finally, the updated representations are fused for next-item prediction. Experiments on four public datasets show that MiRec achieves relative improvements of 1.61%–5.81% in HR@10 and 2.72%–5.65% in NDCG@10 over the strongest baselines. Our code is available at `https://anonymous.4open.science/r/MiRec-6B25/`.

1 Introduction

Sequential recommendation (SR) provides personalized product and content recommendations on e-commerce and online video platforms (Tang et al., 2019; Singer et al., 2022). Early studies primarily focused on modeling dependencies among historical interactions (Liu et al., 2026). For example, GRU4Rec (Hidasi et al., 2016) encodes interaction sequences using recurrent neural networks, while SASRec (Kang & McAuley, 2018) employs self-attention to capture relationships among historical interactions. Multi-interest sequential recommendation methods, such as MIND (Li et al., 2019) and ComiRec (Cen et al., 2020), further learn multiple interest vectors to characterize the diversity of user preferences. However, long-term preferences, period-specific interests, and immediate needs often jointly influence a user's next choice (Zhang et al., 2025b). Capturing interest diversity alone does not adequately account for these differences across temporal scales.

To model user interests at different temporal scales, researchers have proposed various multi-scale sequential recommendation methods. For example, HyperHawkes (Peintner et al., 2025) uses Hawkes processes to model long-term interests and multi-level attention to capture short-term interests, combining their prediction scores for recommendation. MIMAR-SRec (Fang & Shi, 2026) constructs multi-granularity interest representations from subsequences of varying lengths and incorporates intent prototypes to learn user interests over different historical ranges. However, multi-scale sequential recommendation still faces two challenges: **intra-scale interest entanglement and insufficient cross-scale interest mixing**.

At the intra-scale level, stable and dynamic interests may coexist over the same historical range, and temporal partitioning alone does not explicitly separate them. Without interest decomposition, recurring stable interests may obscure dynamic interests, making it difficult to promptly capture changes in user needs. As illustrated in Figure 1(a), consider a family travel scenario in which a user has a longstanding interest in family trips, has recently been browsing seaside travel content, and subsequently begins searching for indoor activities on rainy days. For methods such as MIMAR-SRec (Fang & Shi, 2026), which do not explicitly distinguish stable and dynamic interests at the

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.

intra-scale level, the recurring seaside interest may obscure the emerging need for indoor activities, recommendations may remain biased toward family-friendly beach activities.

At the cross-scale level, even after separating the two types of interests, fusing scale representations only at the prediction stage does not explicitly model the relationships between same-type interests from different scales. Without such mixing, complementary information from different historical ranges cannot directly inform updates to each scale's interest representations, limiting a comprehensive characterization of the user's current needs. As shown in Figure 1(b), under the score summation strategy adopted by HyperHawkes (Peintner et al., 2025), long- and short-term information is combined at the output stage, without explicit representation-level mixing between the two temporal scales. When short-term scores dominate the ranking, ordinary indoor cafés may also rank highly, while the family travel context remains insufficiently utilized.

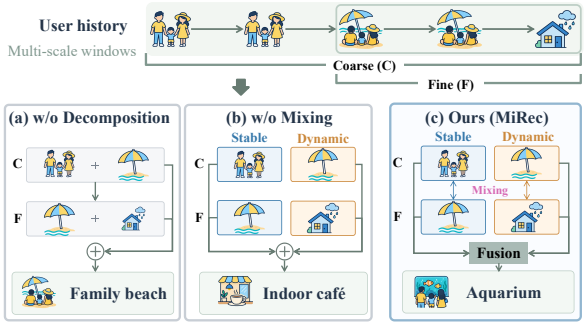


Figure 1: A recommendation example illustrating the motivation for MiRec. (a) Without decomposition, seaside interests may obscure indoor needs. (b) Without mixing, family preferences may be underused. (c) MiRec integrates family preferences with indoor needs. Recommendation outcomes are illustrative.

Motivated by these considerations, we propose **MiRec**, a multi-scale interest decomposition and mixing model. To address intra-scale interest entanglement, MiRec first constructs multi-scale interest representations using causal windows of varying lengths. At each scale, it extracts a stable component and the corresponding residual, forming stable and dynamic interest branches. To address insufficient cross-scale interest mixing, MiRec introduces a cross-scale interest mixer that partitions interest representations from adjacent scales into multiple subspaces along the feature dimension. It performs learnable subspace exchange and recombination separately in the two branches. Through progressive bidirectional propagation, stable interest representations move from coarse to fine scales, while dynamic interest representations move from fine to coarse scales. Finally, MiRec fuses the updated multi-scale interest representations with the base sequence representation for next-item prediction. By combining the stable preference for family travel with the dynamic need for indoor activities, MiRec can recommend a family-friendly aquarium, as illustrated in Figure 1(c). The main contributions are summarized as follows:

(1) We propose an intra-scale interest decomposition mechanism that separately models stable and dynamic interests at each temporal scale.

(2) We design a cross-scale interest mixer to learn subspace correspondences between representations of the same type across scales, enabling targeted integration of complementary information.

(3) Experiments on four public datasets validate the effectiveness of MiRec, and ablation studies verify the contributions of the decomposition and mixing modules.

2 Related Work

Traditional Sequential Recommendation. Sequential recommendation predicts a user's next interaction by modeling sequential dependencies in historical behaviors. Early deep sequential recommendation methods mainly focused on improving sequence encoding architectures. GRU4Rec (Hidasi et al., 2016) recursively models user behaviors with recurrent neural networks, while Caser (Tang & Wang, 2018) captures local sequential patterns through convolutional operations. With the success of Transformers in sequence modeling, SASRec (Kang & McAuley, 2018) introduced unidirectional self-attention to learn user representations by identifying historical behaviors relevant to the current prediction; BERT4Rec (Sun et al., 2019) further employed a bidirectional Transformer with a masked item prediction task to enhance contextual modeling. Subsequent stud-

ies have further improved sequential representations by incorporating temporal information, self-supervised learning, and more efficient attention architectures. These methods mainly focus on how to encode behavioral sequences more accurately, while the coexistence of multiple interests in user histories and their differences across temporal ranges are not their primary modeling targets.

Multi-interest Sequential Recommendation. Multi-interest sequential recommendation learns multiple latent interest representations from user behaviors to characterize users' coexisting preferences. MIND (Li et al., 2019) extracts multiple interest vectors from user histories through dynamic routing, while ComiRec (Cen et al., 2020) further models diverse user preferences through multi-interest representation learning. Subsequent studies have increasingly incorporated clustering and self-supervised learning to improve the quality of interest representations. For example, IO-CLRec (Wang et al., 2025) segments user behaviors into multiple subsequences and employs multi-level contrastive learning to capture fine-grained and diverse interests. InDiRec (Qu & Nobuhara, 2025) obtains latent interest signals through clustering and uses a conditional diffusion model to generate augmented views that preserve interest consistency with the original behaviors. Overall, these methods mainly focus on the diversity of interest content, with the core objective of discovering multiple latent interests and learning more accurate interest representations. MiRec further distinguishes stable and dynamic interests through intra-scale interest decomposition.

Multi-scale Sequential Recommendation. In addition to the diversity of interest content, user behaviors also exhibit clear differences across temporal scales. Some studies have begun to model user interests from short-term and long-term perspectives or across multiple temporal granularities. HyperHawkes (Peintner et al., 2025) separately models long-term recurring interests and fine-grained short-term interests, and fuses the two temporal signals for prediction. MIRRN (Xu et al., 2025) constructs target-aware, local-aware, and global-aware queries from behaviors observed over different temporal ranges to retrieve interests at different granularities, and further models relational information within the corresponding subsequences. MIMAR-SRec (Fang & Shi, 2026) partitions user histories into subsequences at different temporal granularities and learns multi-granularity interest representations through interest alignment. MoS (Lin et al., 2026), targeting interest shifts in long sequences, captures complementary information from multiple scales, including global behaviors, short-term behaviors, and theme-related behaviors. Although these methods construct, retrieve, or fuse representations over different historical ranges, stable and dynamic interests may still be entangled at the intra-scale level. MiRec combines intra-scale interest decomposition with cross-scale interest mixing, using a cross-scale interest mixer to exchange and recombine subspaces between adjacent scales separately in the stable and dynamic branches. Stable interest representations progressively propagate from coarse to fine scales, while dynamic interest representations propagate from fine to coarse scales.

3 METHOD

As shown in Figure 2, MiRec proceeds through multi-scale interest representation construction, intra-scale interest decomposition, cross-scale interest mixing, and fusion for next-item prediction.

3.1 PROBLEM DEFINITION

Sequential recommendation predicts the next item from a user's historical interactions. Let $\mathcal{V}$ denote the item set and $S_u = [v_1, \ldots, v_L]$ the chronologically ordered interaction sequence of user u, where $v_t \in \mathcal{V}$. Given the prefix $S_{u,1:t}$, the model learns a user representation,

$$\mathbf{o}_t = f_\theta(S_{u,1:t}), \tag{1}$$

here, $\mathbf{o}_t, \mathbf{e}_v \in \mathbb{R}^D$, $\mathbf{e}_v$ is an item embedding, and θ contains the learnable parameters. The prediction is $\hat{v}_{t+1} = \arg\max_{v \in \mathcal{V}} \mathbf{o}_t^\top \mathbf{e}_v$. We omit user and batch indices below. All temporal operations are causal and exclude padding positions. Appendix A summarizes the main notation.

3.2 MULTI-SCALE REPRESENTATION CONSTRUCTION

MiRec constructs multi-scale interest representations using causal windows of different lengths. First, item and positional embeddings form the base sequence representation $\mathbf{H} = [\mathbf{h}_1, \ldots, \mathbf{h}_L]^\top \in$

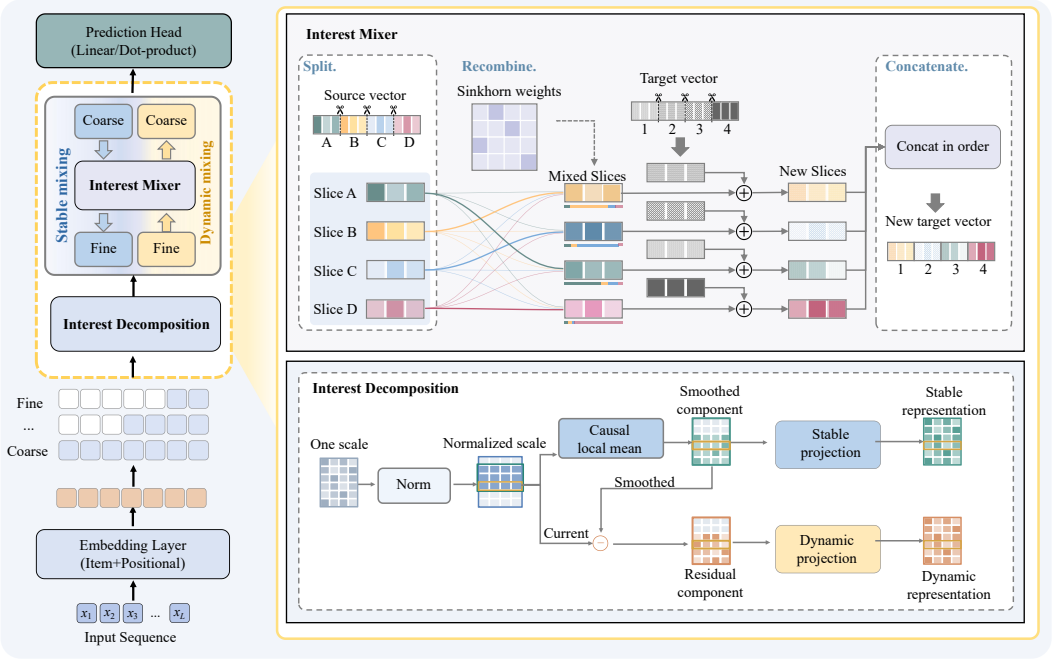


Figure 2: Overview of MiRec with intra-scale interest decomposition and the cross-scale interest mixer. Each Mixer consists of Split, Recombine, and Concatenate.

$\mathbb{R}^{L\times D}$,

$$\mathbf{h}_t = \mathrm{Dropout}\left(\sqrt{D}\,\mathbf{e}_{v_t} + \mathbf{p}_t\right), \tag{2}$$

where $\mathbf{p}_t \in \mathbb{R}^D$ is a learnable positional embedding. We then apply causal mean pooling with window sizes $\mathcal{W} = \{w_1, \ldots, w_M\}$ and add scale embeddings,

$$\mathbf{z}_t^{(m)} = \frac{1}{|\mathcal{I}_t(w_m)|} \sum_{j\in\mathcal{I}_t(w_m)} \mathbf{h}_j + \mathbf{a}_m, \tag{3}$$

here, $\mathbf{a}_m \in \mathbb{R}^D$ is a scale embedding, and $\mathcal{I}_t(w)$ contains the valid positions in $[\max(1, t-w+1), t]$. The windows satisfy $w_1 < \cdots < w_M$, defining scales from fine to coarse. Row t of $\mathbf{Z}^{(m)} \in \mathbb{R}^{L\times D}$ corresponds to $\mathbf{z}_t^{(m)}$. Stacking these representations along the scale dimension yields $\mathbf{Z} \in \mathbb{R}^{L\times M\times D}$.

3.3 Intra-scale Interest Decomposition

Intra-scale interest decomposition extracts stable and dynamic interest representations at each scale through causal smoothing, residual computation, and separate linear projections. First, each scale representation is normalized along the feature dimension:

$$\mathbf{x}_t^{(m)} = \mathrm{LN}_{\mathrm{dec}}\left(\mathbf{z}_t^{(m)}\right). \tag{4}$$

A causal window of length k then extracts the locally smoothed component:

$$\mathbf{T}_t^{(m)} = \frac{1}{|\mathcal{I}_t(k)|} \sum_{j\in\mathcal{I}_t(k)} \mathbf{x}_j^{(m)}. \tag{5}$$

The difference between the normalized representation and its smoothed component gives the residual, $\mathbf{R}_t^{(m)} = \mathbf{x}_t^{(m)} - \mathbf{T}_t^{(m)}$. Finally, separate linear projections produce the stable and dynamic interest representations,

$$\mathbf{s}_t^{(m)} = \mathbf{W}_{\mathrm{sta}}\mathbf{T}_t^{(m)} + \mathbf{b}_{\mathrm{sta}}, \tag{6}$$

$$\mathbf{d}_t^{(m)} = \mathbf{W}_{\text{dyn}}\mathbf{R}_t^{(m)} + \mathbf{b}_{\text{dyn}}, \tag{7}$$

here, $\mathbf{W}_{\text{sta}}, \mathbf{W}_{\text{dyn}} \in \mathbb{R}^{D\times D}$ and $\mathbf{b}_{\text{sta}}, \mathbf{b}_{\text{dyn}} \in \mathbb{R}^D$. Projection parameters are shared across scales within each mixing block, with weights and biases initialized to identity matrices and zeros. The window w_m determines the historical range represented by a scale, while k determines the temporal smoothing range within that scale.

3.4 Cross-scale Interest Mixer

The cross-scale interest mixer updates interest representations of the same type at adjacent scales through three steps: Split, Recombine, and Concatenate. The stable branch propagates from coarse to fine scales, while the dynamic branch propagates from fine to coarse scales. We omit the temporal index t in this subsection; all Mixer operations act at the same temporal position.

The two branches update their representations through successive Mixer operations. Starting from $\widehat{\mathbf{s}}^{(M)} = \mathbf{s}^{(M)}$, the stable branch updates scales in the order $m = M-1, \ldots, 1$,

$$\widehat{\mathbf{s}}^{(m)} = \text{Mixer}_{\text{sta}}^{(m)}\left(\widehat{\mathbf{s}}^{(m+1)}, \mathbf{s}^{(m)}\right). \tag{8}$$

Starting from $\widehat{\mathbf{d}}^{(1)} = \mathbf{d}^{(1)}$, the dynamic branch updates scales in the order $m = 2, \ldots, M$:

$$\widehat{\mathbf{d}}^{(m)} = \text{Mixer}_{\text{dyn}}^{(m-1)}\left(\widehat{\mathbf{d}}^{(m-1)}, \mathbf{d}^{(m)}\right). \tag{9}$$

Superscript m identifies the Mixer connecting scales m and $m+1$, and its inputs are the source and target representations, respectively. Each step uses the updated source representation. Different branches and connections have independent parameters. Each Mixer comprises three steps.

(1) Split. Split partitions the source and target representations into K contiguous feature subspaces. For $\mathbf{Z}^{\text{src}}, \mathbf{Z}^{\text{tar}} \in \mathbb{R}^D$, let $d = \lceil D/K \rceil$ be the subspace dimension and $D' = Kd$ the padded dimension. After zero-padding both representations to D' dimensions, we partition them into K subvectors $\mathbf{Z}_i^{\text{src}}$ and $\mathbf{Z}_i^{\text{tar}}$, each in $\mathbb{R}^d$.

(2) Recombine. Recombine combines source information using learnable subspace correspondences and injects it into the target subspaces. First, a learnable matrix $\mathbf{A} \in \mathbb{R}^{K\times K}$ is transformed into exchange weights, $\mathbf{E} = \text{Sinkhorn}_I(\mathbf{A}/\tau)$. Here, $\tau > 0$ is the temperature and I the number of iterations. Starting from the elementwise exponential $\exp(\mathbf{A}/\tau)$, Sinkhorn alternates row and column normalization to obtain an approximately doubly stochastic matrix. Entry E_{ij} weights the contribution of source subspace j to target subspace i, which is updated as

$$\widetilde{\mathbf{Z}}_i^{\text{tar}} = \mathbf{Z}_i^{\text{tar}} + \lambda \sum_{j=1}^{K} E_{ij}\mathbf{Z}_j^{\text{src}}. \tag{10}$$

The injection coefficient is fixed at $\lambda = 1/\sqrt{2}$. The exchange-weight matrix $\mathbf{E}$ of each Mixer is shared across users and temporal positions.

(3) Concatenate. Concatenate joins the updated target subspaces in their original order and removes padded dimensions to produce the Mixer output:

$$\widetilde{\mathbf{Z}}^{\text{tar}} = \text{Trunc}_D\left(\text{Concat}(\widetilde{\mathbf{Z}}_1^{\text{tar}}, \ldots, \widetilde{\mathbf{Z}}_K^{\text{tar}})\right). \tag{11}$$

Here, Trunc_D retains the first D dimensions. The complete operation is denoted by $\text{Mixer}(\mathbf{Z}^{\text{src}}, \mathbf{Z}^{\text{tar}}) = \widetilde{\mathbf{Z}}^{\text{tar}}$.

After both branches complete their Mixer updates, their changes are added back to the original scale representations. The branch increments are first combined:

$$\boldsymbol{\Delta}^{(m)} = \left(\widehat{\mathbf{s}}^{(m)} - \mathbf{s}^{(m)}\right) + \left(\widehat{\mathbf{d}}^{(m)} - \mathbf{d}^{(m)}\right). \tag{12}$$

A residual connection then updates the original representation: $\mathbf{u}^{(m)} = \mathbf{z}^{(m)} + \gamma\,\text{Dropout}\left(\boldsymbol{\Delta}^{(m)}\right)$, where γ is a learnable scalar initialized to 1. Finally, a scale-specific feed-forward network updates the features:

$$\mathbf{z}^{*(m)} = \mathbf{u}^{(m)} + \text{Dropout}\left(\text{FFN}_m\left(\text{LN}_{\text{ffn}}(\mathbf{u}^{(m)})\right)\right). \tag{13}$$

Each FFN_m contains two linear layers with GELU and dropout. Intra-scale interest decomposition, the cross-scale interest mixer, and the subsequent residual and feed-forward updates form one mixing block. With ℓ independently parameterized blocks, each block takes the preceding block's output as input. The final scale representation is $\mathbf{Z}^{*(m)} \in \mathbb{R}^{L\times D}$, whose row t corresponds to $\mathbf{z}_t^{*(m)}$.

3.5 Multi-scale Fusion and Prediction

MiRec aggregates the updated scale representations and fuses them with the base sequence representation for next-item prediction. First, the normalized scale representations are averaged:

$$\bar{\mathbf{z}}_t = \frac{1}{M}\sum_{m=1}^{M} \mathrm{LN}_{\mathrm{out}}\left(\mathbf{z}_t^{*(m)}\right). \tag{14}$$

A scalar gate is then computed to fuse the two representations: $g_t = \sigma(\mathrm{MLP}([\mathbf{h}_t \| \bar{\mathbf{z}}_t]))$. The final user representation is $\mathbf{o}_t = g_t\mathbf{h}_t + (1-g_t)\bar{\mathbf{z}}_t$. Here, $[\cdot\|\cdot]$ denotes concatenation and σ the sigmoid function. The MLP uses two linear mappings, $2D \to D \to 1$, with GELU between them, producing $g_t \in (0,1)$.

Item-matching scores determine the training loss. The ground-truth next item v_t^+ and a sampled negative item v_t^- receive scores $r_t^+ = \mathbf{o}_t^\top \mathbf{e}_{v_t^+}$ and $r_t^- = \mathbf{o}_t^\top \mathbf{e}_{v_t^-}$, respectively. The training objective is binary cross-entropy:

$$\mathcal{L} = -\frac{1}{|\Omega|}\sum_{t\in\Omega}\left[\log\sigma(r_t^+) + \log\big(1-\sigma(r_t^-)\big)\right], \tag{15}$$

where Ω contains the valid prediction positions. During inference, the representation at the last valid position is used to rank candidate items. A detailed time complexity analysis of MiRec is provided in Appendix B.

4 Experiments

4.1 Experimental Setup

Datasets. We evaluate MiRec on four public datasets: Beauty (McAuley et al., 2015), Kindle (McAuley et al., 2015), ML-1M (Harper & Konstan, 2015), and Video (McAuley et al., 2015). Beauty, Kindle, and Video are three category-specific subsets of the Amazon review dataset, containing timestamped user–item interaction records. ML-1M is a MovieLens dataset containing users' movie ratings and their associated timestamps.

Baselines. We compare MiRec with three groups of baselines: (1) general recommendation and basic sequence encoding methods: SASRec (Kang & McAuley, 2018) and LightGCN (He et al., 2020); (2) multi-interest and intent modeling methods: MIND (Li et al., 2019), ComiRec-SA (Cen et al., 2020), MGNM (Tian et al., 2022), TiMiRec (Wang et al., 2022), REMI (Xie et al., 2023), DisMIR (Du et al., 2024), IDURL (Lin et al., 2025), InDiRec (Qu & Nobuhara, 2025), IOCLRec (Wang et al., 2025), and MRFF (Zhang et al., 2025a); and (3) time-aware and multi-scale modeling methods: HM4SR (Zhang et al., 2025b), HyperHawkes (Peintner et al., 2025), MIRRN (Xu et al., 2025), MIMAR-SRec (Fang & Shi, 2026), and MoS (Lin et al., 2026).

Dataset details, experimental settings, evaluation metrics, and baseline descriptions are provided in Appendices C.1–C.4.

4.2 Overall Performance

Table 1 compares MiRec with 17 baselines on four datasets. MiRec achieves the best results on both metrics across all datasets, with relative improvements of 1.61%–5.81% in HR@10 and 2.72%–5.65% in NDCG@10 over the strongest baselines. These results demonstrate its consistent effectiveness. Average-rank comparisons and Nemenyi tests are provided in Appendix C.5.

Comparison with general recommendation and basic sequence encoding methods. SASRec (Kang & McAuley, 2018) outperforms LightGCN (He et al., 2020) on all four datasets. MiRec

Table 1: Overall performance on four datasets. The best and second-best results are **bold** and underlined, respectively. *Improv.* is the relative gain over the strongest baseline.

Method	Beauty		Kindle		ML-1M		Video	
	HR@10	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10
SASRec (2018)	0.4675	0.3192	0.6590	0.5154	0.7939	0.5514	0.7276	0.5292
LightGCN (2020)	0.3218	0.2209	0.3614	0.2728	0.4104	0.2335	0.4947	0.3258
MIND (2019)	0.4163	0.2726	0.6253	0.4705	0.6776	0.4056	0.6846	0.4679
ComiRec-SA (2020)	0.4433	0.2944	0.6085	0.4628	0.6742	0.4163	0.6517	0.4474
MGNM (2022)	0.4357	0.2907	0.6098	0.4668	0.6662	0.4036	0.6566	0.4612
TiMiRec (2022)	0.4622	0.3103	0.6322	0.4934	0.6925	0.4254	0.6787	0.4684
REMI (2023)	0.4205	0.2740	0.6120	0.4538	0.6258	0.3636	0.6263	0.4274
DisMIR (2024)	0.4570	0.3075	0.6402	0.4877	0.6881	0.4220	0.6863	0.4835
IDURL (2025)	0.3550	0.2045	0.3492	0.2225	0.6815	0.4418	0.4692	0.2788
InDiRec (2025)	0.4693	0.3239	0.6151	0.4559	0.7932	0.5542	0.7355	0.5322
IOCLRec (2025)	0.4646	0.3218	0.6522	0.5017	0.7055	0.4639	0.7350	0.5370
MRFF (2025)	0.4732	0.3276	0.6534	0.5092	0.7950	0.5509	0.7338	0.5302
HM4SR (2025)	0.3727	0.2399	0.5571	0.3895	0.7156	0.4670	0.6636	0.4540
HyperHawkes (2025)	0.3290	0.1983	0.5437	0.3809	0.6987	0.4608	0.5233	0.3245
MIRRN (2025)	0.4298	0.2873	0.6232	0.4902	0.7705	0.5223	0.6829	0.4776
MIMAR-SRec (2026)	0.4606	0.3172	0.6596	0.5165	0.7869	0.5391	0.7413	0.5377
MoS (2026)	0.4229	0.2909	0.6058	0.4760	0.7478	0.5182	0.6910	0.4928
MiRec	**0.5007**	**0.3461**	**0.6805**	**0.5441**	**0.8078**	**0.5693**	**0.7574**	**0.5565**
Improv.	+5.81%	+5.65%	+3.17%	+5.34%	+1.61%	+2.72%	+2.17%	+3.50%

further surpasses SASRec by 7.10% in HR@10 and 8.43% in NDCG@10 on Beauty, and by 4.10% and 5.16% on Video, respectively. Beyond encoding sequential dependencies, MiRec explicitly models interests over different temporal ranges through intra-scale interest decomposition and cross-scale interest mixing.

Comparison with multi-interest and intent modeling methods. MRFF (Zhang et al., 2025a) leads this category on both metrics of Beauty and on HR@10 of ML-1M, while InDiRec (Qu & Nobuhara, 2025) performs best on NDCG@10 of ML-1M. MiRec improves HR@10 and NDCG@10 over MRFF by 5.81% and 5.65% on Beauty, and exceeds the corresponding strongest baselines by 1.61% and 2.72% on ML-1M. These results support the effectiveness of modeling stable and dynamic interests across temporal scales.

Comparison with time-aware and multi-scale methods. MIMAR-SRec (Fang & Shi, 2026) is the strongest baseline in this category and among all baselines on Kindle and Video. MiRec surpasses it by 3.17% in HR@10 and 5.34% in NDCG@10 on Kindle, and by 2.17% and 3.50% on Video, respectively. These gains support MiRec's approach of decomposing stable and dynamic interests within each scale and mixing representations of the same type across scales. The individual contributions of decomposition and mixing are further examined in the ablation study.

4.3 Ablation Study

We evaluate four variants: (1) **w/o Decomp.** removes intra-scale interest decomposition and directly mixes interest representations across scales; (2) **w/o Mixing** retains decomposition but removes cross-scale interest mixing; (3) **Fixed Exch.** replaces learnable subspace correspondences with fixed ones; and (4) **Non-prog. Mix.** uses original rather than progressively updated source representations at each adjacent-scale update. Table 2 reports the results on Beauty and ML-1M. Complete results and additional ablation studies are provided in Appendix C.6.

Table 2: Ablation results on Beauty and ML-1M. Best results are in bold.

Variant	Beauty		ML-1M	
	HR@10	NDCG@10	HR@10	NDCG@10
MiRec	**0.5007**	**0.3461**	**0.8078**	**0.5693**
w/o Decomp.	0.4830	0.3372	0.7978	0.5502
w/o Mixing	0.4860	0.3377	0.7952	0.5612
Fixed Exch.	0.4858	0.3364	0.8000	0.5599
Non-prog. Mix.	0.4862	0.3388	0.8007	0.5636

Averaged across all four datasets, MiRec improves HR@10 and NDCG@10 by 2.04% and 2.60% over w/o Decomp., and by 2.30% and 2.49% over w/o Mixing, respectively. These gains support

separately modeling stable and dynamic interests within each scale and integrating complementary information across scales. MiRec also achieves average relative improvements of 1.80% in HR@10 and 2.13% in NDCG@10 over Fixed Exch., and 1.66% and 1.70% over Non-prog. Mix., respectively. These results support the benefits of learnable subspace exchange and progressive propagation in the Cross-scale Interest Mixer.

4.4 Visualizing Intra-scale Interest Decomposition.

To examine intra-scale interest decomposition, we visualize a user on ML-1M at the scale with window size 16. Auxiliary prediction heads map the pre-decomposition, stable, and dynamic representations to movie-genre distributions. Figure 3 shows how these distributions evolve with successive interactions, with darker colors indicating higher predicted genre weights.

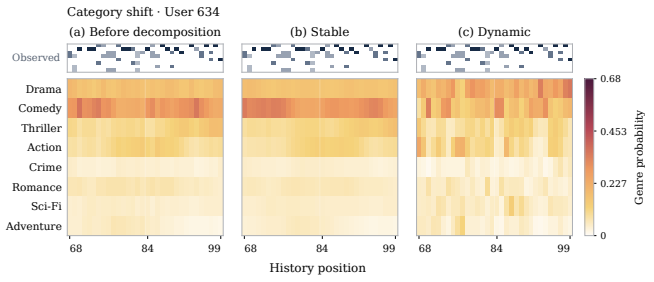


Figure 3: Intra-scale interest decomposition on ML-1M. Panels show auxiliary genre predictions from the pre-decomposition, stable, and dynamic representations. The top strips show the genres of the last observed item at each position, the lower heatmaps show predictions for the next item's genres.

The stable branch maintains a relatively consistent genre distribution, whereas the dynamic branch exhibits more pronounced local variation. For user 634, Drama interactions become more frequent toward the end of the displayed sequence, accompanied by an overall increase in the corresponding dynamic-branch weights. The mean Drama weight increases from 0.2031 in the first segment to 0.2679 in the last segment, while the stable-branch means remain similar at 0.1678 and 0.1656, respectively. These observations support the role of intra-scale interest decomposition in learning stable and dynamic interest representations with distinct variation patterns. Visualization details and additional cases are provided in Appendix C.7. A complementary analysis of 1,000 ML-1M users examines the temporal variation of both branches and their responses to behavioral novelty before and after subspace exchange within the cross-scale interest mixer (Appendix C.8).

4.5 Analysis of the Cross-scale Interest Mixer

Figure 4 visualizes the learned exchange-weight matrices between window sizes 4 and 16 on ML-1M, using seed 2026 and the checkpoint selected by validation NDCG@10. Rows and columns denote target and source subspaces, respectively. Additional visualizations for Beauty, Kindle, and Video, together with statistics across all adjacent window pairs and three random seeds, are provided in Appendix C.9. The stable $16 \rightarrow 4$ transfer concentrates most of its weight on diagonal entries while retaining nonzero weights for cross-subspace mixing. The dynamic $4 \rightarrow 16$ transfer allocates a larger share of its weight to off-diagonal entries. This difference is observed on ML-1M across seeds 2026, 2027, and 2028. Averaged over these three seeds, the diagonal mass fraction, defined as the sum of diagonal entries divided by the sum of all matrix entries, is 0.8578 for the stable branch and 0.5417 for the dynamic branch; the corresponding off-diagonal fractions are 0.1422

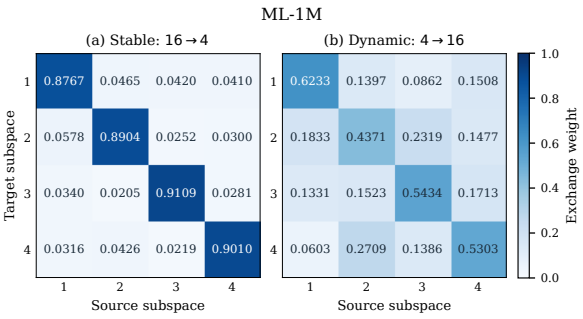


Figure 4: Learned subspace correspondences on ML-1M with seed 2026: (a) stable transfer ($16 \rightarrow 4$); (b) dynamic transfer ($4 \rightarrow 16$).

and 0.4583. These results show that the two branches learn different balances between same-index transfer and cross-subspace mixing at this window pair.

4.6 Hyperparameter Sensitivity Analysis

We vary the scale multiplier $r \in \{2, 3, 4, 5\}$ using four windows $\{1, r, r^2, r^3\}$, with other settings fixed. Figure 5 shows that $r = 4$ yields the best HR@10 and NDCG@10 on Beauty and ML-1M. On Beauty, it improves the two metrics by 3.05% and 2.67%, respectively, over $r = 2$. The consistent improvements from $r = 2$ to $r = 4$ suggest that a broader range of temporal contexts benefits interest modeling. However, both metrics decline when r increases to 5, indicating that further expanding the scale spacing does not necessarily improve recommendation performance. Results on the remaining datasets and sensitivity analyses of the other hyperparameters are provided in Appendix C.10.

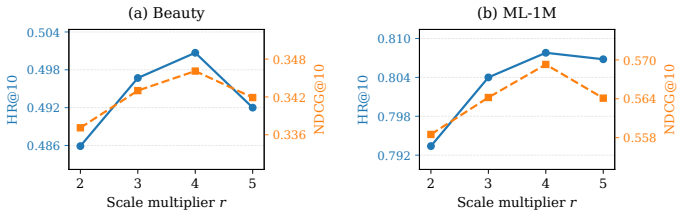


Figure 5: Sensitivity to the scale multiplier r on (a) Beauty and (b) ML-1M. Each panel reports HR@10 and NDCG@10.

4.7 Efficiency Analysis

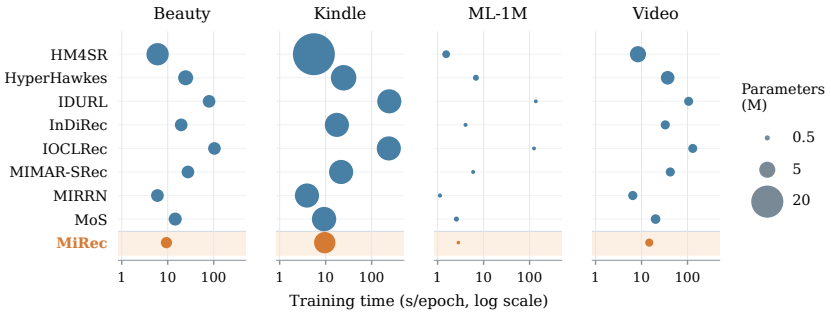


Figure 6: Training time and model size. The horizontal axis shows average training time per epoch over 200 epochs on a logarithmic scale. Circle area is proportional to parameter count, using the same size mapping across all panels. MiRec is highlighted in orange.

Figure 6 compares training time and parameter count. MiRec has the fewest parameters across all four datasets, reducing the parameter count by 10.64%–21.59% relative to IOCLRec. Causal averaging and smoothing introduce no learnable parameters, while small $K \times K$ subspace exchange matrices limit the parameter overhead of cross-scale interest mixing. These designs enable MiRec to achieve better recommendation performance with a smaller model. MiRec's average training time ranges from 2.83 to 14.72 seconds per epoch, corresponding to an $8.83\times$–$44.12\times$ speedup over IOCLRec. Prefix sums avoid repeated summation over overlapping windows, while subspace exchange runs in parallel across sequence positions, reducing computational overhead. Although HM4SR and MIRRN train faster, MiRec achieves better recommendation performance, offering a favorable balance between model size, training cost, and accuracy.

5 Conclusion

MiRec jointly models stable and dynamic interests through intra-scale interest decomposition and cross-scale interest mixing. Experiments on four public datasets demonstrate its effectiveness, while ablation studies show that both modules contribute to performance improvements. Future work will explore adaptive scale selection and decomposition mechanisms to organize historical information in ways that better reflect evolving user interests.

AI USE STATEMENT

In this work, we used generative AI tools for for grammar checking and language polishing of the manuscript.

ETHICS STATEMENT

This work studies sequential recommendation using publicly available benchmark datasets. We follow the original data usage guidelines and focus on improving recommendation accuracy without applications affecting individual rights or safety.

REPRODUCIBILITY STATEMENT

The main paper describes the overall framework of MiRec. Detailed descriptions of the model architecture and optimization objective are provided in Section 3. Appendix A summarizes the main notations, while Appendix C provides dataset descriptions, preprocessing procedures, experimental settings, evaluation protocols, and implementation details. Our code is available in an anonymous repository, with the link provided in the abstract.

REFERENCES

Yukuo Cen, Jianwei Zhang, Xu Zou, Chang Zhou, Hongxia Yang, and Jie Tang. Controllable multi-interest framework for recommendation. In *Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 2942–2951, 2020.

Yingpeng Du, Ziyan Wang, Zhu Sun, Yining Ma, Hongzhi Liu, and Jie Zhang. Disentangled multi-interest representation learning for sequential recommendation. In *Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining*, pp. 677–688, 2024.

Yangyi Fang and Haolin Shi. Multi-granularity intent modeling with adversarial robustness for sequential recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 14702–14710, 2026.

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

Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. LightGCN: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 639–648, 2020.

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

Wang-Cheng Kang and Julian McAuley. Self-attentive sequential recommendation. In *Proceedings of the 2018 IEEE International Conference on Data Mining*, pp. 197–206, 2018.

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 *Proceedings of the 28th ACM international conference on information and knowledge management*, pp. 2615–2623, 2019.

Xiao Lin, Zhicheng Tang, Weilin Cong, Mengyue Hang, Kai Wang, Yajuan Wang, Zhichen Zeng, Ting-Wei Li, Hyunsik Yoo, Zhining Liu, et al. Mixture of sequence: Theme-aware mixture-of-experts for long-sequence recommendation. In *Proceedings of the ACM Web Conference 2026*, pp. 6469–6480, 2026.

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

Yuchen Liu, Kunyu Ni, Zhongying Zhao, Guoqing Chao, and Yanwei Yu. S^2HyRec: Self-supervised hypergraph sequential recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 15394–15402, 2026.

Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 43–52, 2015.

Andreas Peintner, Amir Reza Mohammadi, Michael Müller, and Eva Zangerle. Hypergraph-based temporal modelling of repeated intent for sequential recommendation. In *Proceedings of the ACM on Web Conference 2025*, pp. 3809–3818, 2025.

Yuanpeng Qu and Hajime Nobuhara. Intent-aware diffusion with contrastive learning for sequential recommendation. In *Proceedings of the 48th international ACM SIGIR conference on research and development in information retrieval*, pp. 1552–1561, 2025.

Uriel Singer, Haggai Roitman, Yotam Eshel, Alexander Nus, Ido Guy, Or Levi, Idan Hasson, and Eliyahu Kiperwasser. Sequential modeling with multiple attributes for watchlist recommendation in e-commerce. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, pp. 937–946, 2022.

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 *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pp. 1441–1450, 2019.

Jiaxi Tang and Ke Wang. Personalized top-n sequential recommendation via convolutional sequence embedding. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pp. 565–573, 2018.

Jiaxi Tang, Francois Belletti, Sagar Jain, Minmin Chen, Alex Beutel, Can Xu, and Ed H. Chi. Towards neural mixture recommender for long range dependent user sequences. In *Proceedings of the The World Wide Web Conference*, pp. 1782–1793, 2019.

Yu Tian, Jianxin Chang, Yanan Niu, Yang Song, and Chenliang Li. When multi-level meets multi-interest: A multi-grained neural model for sequential recommendation. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval*, pp. 1632–1641, 2022.

Chenyang Wang, Zhefan Wang, Yankai Liu, Yang Ge, Weizhi Ma, Min Zhang, Yiqun Liu, Junlan Feng, Chao Deng, and Shaoping Ma. Target interest distillation for multi-interest recommendation. In *Proceedings of the 31st ACM international conference on information & knowledge management*, pp. 2007–2016, 2022.

Wuhong Wang, Jianhui Ma, Yuren Zhang, Kai Zhang, Junzhe Jiang, Yihui Yang, Yacong Zhou, and Zheng Zhang. Intent oriented contrastive learning for sequential recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 12748–12756, 2025.

Yueqi Xie, Jingqi Gao, Peilin Zhou, Qichen Ye, Yining Hua, Jae Boum Kim, Fangzhao Wu, and Sunghun Kim. Rethinking multi-interest learning for candidate matching in recommender systems. In *Proceedings of the 17th ACM conference on recommender systems*, pp. 283–293, 2023.

Xiang Xu, Hao Wang, Wei Guo, Luankang Zhang, Wanshan Yang, Runlong Yu, Yong Liu, Defu Lian, and Enhong Chen. Multi-granularity interest retrieval and refinement network for long-term user behavior modeling in ctr prediction. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, pp. 2745–2755, 2025.

Chunxu Zhang, Guodong Long, Hongkuan Guo, Zhaojie Liu, Guorui Zhou, Zijian Zhang, Yang Liu, and Bo Yang. Multifaceted user modeling in recommendation: A federated foundation models approach. In *Proceedings of the AAAI conference on artificial intelligence*, volume 39, pp. 13197–13205, 2025a.

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, 2025b.

APPENDIX

A NOTATION SUMMARY

Table 3 summarizes the main symbols used in this paper and their corresponding meanings.

Table 3: Summary of the main notations used in MiRec.

Symbol	Description
u	A user.
$\mathcal{V}$	The set of all items.
$S_u = [v_1, \ldots, v_L]$	Chronologically ordered interaction sequence of user u.
v_t	The item interacted with at position t.
L	Length of the interaction sequence.
D	Dimensionality of the hidden representation.
$\mathbf{H} = [\mathbf{h}_1, \ldots, \mathbf{h}_L]^\top$	Base sequence representation, where $\mathbf{h}_t \in \mathbb{R}^D$ is the representation at position t.
$\mathcal{W} = \{w_1, \ldots, w_M\}$	Set of causal window sizes for multi-scale representation construction.
M	Number of temporal scales.
w_m	Causal window size of the m-th temporal scale.
r	Scale multiplier used to construct temporal windows.
$\mathbf{Z}^{(m)}$	Interest representation at the m-th temporal scale.
$\mathbf{z}_t^{(m)}$	Interest representation at position t of scale m.
$\mathbf{x}_t^{(m)}$	Normalized representation of $\mathbf{z}_t^{(m)}$ used for intra-scale interest decomposition.
k	Decomposition window size used for causal smoothing.
$\mathcal{I}_t(w)$	Set of valid positions in the causal window of size w ending at position t.
$\mathbf{T}_t^{(m)}$	Causally smoothed component at position t of scale m.
$\mathbf{R}_t^{(m)}$	Residual component obtained by subtracting $\mathbf{T}_t^{(m)}$ from $\mathbf{x}_t^{(m)}$.
$\mathbf{s}_t^{(m)}$	Stable interest representation at position t of scale m.
$\mathbf{d}_t^{(m)}$	Dynamic interest representation at position t of scale m.
$\mathbf{W}_{\text{sta}}, \mathbf{b}_{\text{sta}}$	Learnable projection parameters for the stable interest branch.
$\mathbf{W}_{\text{dyn}}, \mathbf{b}_{\text{dyn}}$	Learnable projection parameters for the dynamic interest branch.
K	Number of feature subspaces in the cross-scale interest mixer.
d	Dimensionality of each feature subspace.
D'	Padded feature dimension, where $D' = Kd$.
$\mathbf{Z}^{\text{src}}, \mathbf{Z}^{\text{tar}}$	Source and target interest representations in a Mixer operation between adjacent scales.
$\mathbf{A} \in \mathbb{R}^{K \times K}$	Learnable subspace correspondence matrix before Sinkhorn normalization.
$\mathbf{E} \in \mathbb{R}^{K \times K}$	Exchange-weight matrix obtained by Sinkhorn normalization.
E_{ij}	Weight assigned to source subspace j when updating target subspace i.
τ	Exchange temperature used in Sinkhorn normalization.
I	Number of Sinkhorn normalization iterations.
λ	Coefficient controlling cross-scale information injection.
$\widetilde{\mathbf{Z}}^{\text{tar}}$	Target interest representation output by the cross-scale interest mixer.
γ	Learnable coefficient scaling the combined branch increment.
ℓ	Number of mixing blocks.
$\mathbf{Z}^{*(m)}$	Final interest representation of scale m after all mixing blocks.
$\bar{\mathbf{z}}_t$	Aggregated multi-scale representation at position t.
g_t	Gating weight balancing the base sequence and aggregated multi-scale representations.
$\mathbf{o}_t$	Final user representation at position t.
v_t^+, v_t^-	Positive next item and sampled negative item at position t, respectively.
r_t^+, r_t^-	Prediction scores for the positive and negative items, respectively.
Ω	Set of valid prediction positions used for training.
$\mathcal{L}$	Binary cross-entropy training objective.

B Time Complexity Analysis

We analyze the time complexity for a sequence of length L with hidden dimension D, treating the number of temporal scales M, mixing blocks ℓ, feature subspaces K, and Sinkhorn iterations I as fixed hyperparameters. The FFN expansion ratio is also fixed.

Multi-scale representation construction requires $\mathcal{O}(LD)$ time, as causal window means are computed using prefix sums. Within each mixing block, causal smoothing and residual computation also cost $\mathcal{O}(LD)$, while the linear projections producing stable and dynamic interest representations require $\mathcal{O}(LD^2)$.

The cross-scale interest mixer requires $\mathcal{O}(LD)$ time. With fixed K and I, computing the exchange-weight matrices has constant cost, and applying subspace exchange across all temporal positions costs $\mathcal{O}(LD)$. The scale-specific FFNs and final gating network each require $\mathcal{O}(LD^2)$, while scale aggregation costs $\mathcal{O}(LD)$. Therefore, the overall time complexity of sequence representation computation is

$$\mathcal{T}_{\text{enc}} = \mathcal{O}(LD^2). \tag{16}$$

During training, scoring one positive item and one sampled negative item at each valid position adds $\mathcal{O}(LD)$ time, leaving the forward complexity unchanged. During inference, scoring C candidate items adds $\mathcal{O}(CD)$ time. Thus, for a fixed hidden dimension, MiRec's representation computation scales linearly with sequence length.

C More Experimental Details

C.1 Dataset Details

We evaluate MiRec on four public datasets covering different recommendation domains and interaction densities. Beauty (McAuley et al., 2015) contains user interactions with beauty products on Amazon. Kindle (McAuley et al., 2015) contains user interactions with products in Amazon's Kindle Store, reflecting users' reading preferences. ML-1M (Harper & Konstan, 2015) is a movie-rating dataset collected by the MovieLens platform. Video (McAuley et al., 2015) contains user interactions with products in Amazon's Video Games category. Each user–item record in the preprocessed files is treated as an implicit positive interaction; the numerical rating values are not used by MiRec.

Table 4 summarizes the preprocessed datasets before training, validation, and test splitting. Density is calculated as $|\mathcal{R}|/(|\mathcal{U}|\,|\mathcal{I}|) \times 100\%$, where $\mathcal{R}$, $\mathcal{U}$, and $\mathcal{I}$ denote the interaction records, users, and items, respectively. The average sequence length is $|\mathcal{R}|/|\mathcal{U}|$, computed before truncation. The Amazon subsets are sparse, whereas ML-1M has a higher interaction density and longer user histories on average.

Table 4: Statistics of the preprocessed datasets before splitting. Avg. length denotes the average number of interactions per user before truncation.

Dataset	# Users	# Items	# Interactions	Density	Avg. length
Beauty	19,807	47,468	244,280	0.026%	12.33
Kindle	20,658	175,043	720,213	0.020%	34.86
ML-1M	6,040	3,416	999,611	4.845%	165.50
Video	31,013	23,715	287,107	0.039%	9.26

We adopt a leave-one-out strategy for data splitting. For each user with at least three interactions, the last interaction is reserved for testing, the penultimate interaction for validation, and the remaining interactions for training.

C.2 Additional Experimental Details

We implement MiRec using PyTorch 2.0.0 and conduct experiments on an Ubuntu 20.04 server equipped with one 32 GB vGPU and 62 GB of RAM. The software environment uses Python 3.8 and CUDA 11.8. Unless otherwise specified, the embedding dimension and maximum sequence

length are both set to 50, the batch size is 128, and the dropout rate is 0.5. We optimize the model using Adam with a learning rate of 0.001 for 200 epochs, evaluating after the first epoch and every ten epochs thereafter. For MiRec, we use four temporal windows, $\{1, 4, 16, 64\}$, one progressive mixing block, and a smoothing window of 3. The cross-scale exchange module uses four feature subspaces, with a Sinkhorn temperature of 0.5 and six normalization iterations.

To reduce the computational cost of full-catalog ranking over large item sets, we rank each ground-truth target item against sampled negative items and report HR@10 and NDCG@10. MiRec and all baseline methods use the same data splits and evaluation protocol.

C.3 Evaluation Metrics

HR@10. Hit Rate at 10 (HR@10) measures the proportion of users whose ground-truth target item appears among the top ten recommendations. It indicates whether the target item is successfully retrieved, without distinguishing its position within the top ten.

NDCG@10. Normalized Discounted Cumulative Gain at 10 (NDCG@10) accounts for both the presence and the position of the target item within the top ten recommendations. Higher-ranked targets receive larger gains through logarithmic discounting, while targets outside the top ten contribute zero gain.

For a single ground-truth target per user, let $\mathcal{U}$ denote the user set and r_u the one-based rank of user u's target item. The two metrics are defined as

$$\mathrm{HR@10} = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \mathbf{1}[r_u \leq 10], \tag{17}$$

$$\mathrm{NDCG@10} = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \frac{\mathbf{1}[r_u \leq 10]}{\log_2(r_u + 1)}, \tag{18}$$

where $\mathbf{1}[\cdot]$ is the indicator function. Both metrics range from 0 to 1, with higher values indicating better recommendation performance.

C.4 Baseline Methods

We group the baselines by their primary modeling focus into three categories: general recommendation and basic sequence encoding, multi-interest and intent modeling, and time-aware and multi-scale modeling. Within each category, methods are ordered by publication year, with ties resolved alphabetically. These categories may overlap technically; the descriptions below summarize the original designs of the methods.

General recommendation and basic sequence encoding.

- **SASRec** (Kang & McAuley, 2018) uses causal self-attention to identify historical interactions relevant to next-item prediction and learn the current sequential state. MiRec constructs multi-scale interest representations from the base sequence representation and separates stable and dynamic interests within each scale.
- **LightGCN** (He et al., 2020) learns collaborative filtering representations through simplified neighborhood propagation on the user–item bipartite graph. It primarily exploits interaction graph structure, whereas MiRec uses behavior order to model interest variations across multiple temporal scales and the relationships between them.

Multi-interest and intent modeling.

- **MIND** (Li et al., 2019) extracts multiple interest vectors through dynamic routing and uses label-aware attention to learn interests relevant to the target item. It focuses on discovering coexisting interests, whereas MiRec further distinguishes stable and dynamic interests within each temporal scale.

- **ComiRec-SA** (Cen et al., 2020) extracts multiple interest vectors using self-attention with positional embeddings and employs controllable aggregation to balance recommendation accuracy and diversity. It emphasizes coverage of different interest content, whereas MiRec models interest variations over different temporal ranges and complementary information across scales.
- **MGNM** (Tian et al., 2022) combines user-aware graph convolution with sequential capsule networks to extract multiple interests from different levels of item relationships. Its granularity primarily arises from graph propagation levels, whereas MiRec defines scales using temporal windows and performs intra-scale interest decomposition and progressive cross-scale interest mixing.
- **TiMiRec** (Wang et al., 2022) distills a target-interest predictor using similarities between the target item and individual interests as soft supervision, enabling context-dependent interest aggregation. It focuses on selecting currently relevant interests, whereas MiRec jointly models stable and dynamic interests through intra-scale interest decomposition and cross-scale interest mixing.
- **REMI** (Xie et al., 2023) uses interest-aware hard negative mining and routing regularization to address weak training signals and routing collapse in multi-interest learning. It primarily improves the training of multi-interest models, whereas MiRec introduces a representation structure that models stable and dynamic interests and their cross-scale relationships.
- **DisMIR** (Du et al., 2024) uses global item co-occurrence to guide item partitioning and refines interest disentanglement through Markov random fields and contrastive learning. It separates distinct semantic interests, whereas MiRec separates stable and dynamic interests within each temporal scale.
- **IDURL** (Lin et al., 2025) quantifies interest drift using item categories and learns user representations for different drift levels through representation disentanglement and drift-aware alignment. Its disentanglement targets interest drift levels, whereas MiRec extracts stable and dynamic interest representations within each temporal scale and models their cross-scale complementarity.
- **InDiRec** (Qu & Nobuhara, 2025) obtains latent intents through clustering and uses them to condition a diffusion model that generates semantically consistent augmented views for contrastive learning. It focuses on preserving intent consistency during augmentation, whereas MiRec extracts stable and dynamic interest representations at each scale and performs cross-scale interest mixing through learnable subspace exchange.
- **IOCLRec** (Wang et al., 2025) partitions historical interactions into subsequences representing intents at different stages and strengthens intent representations through fine-grained intent, single-intent, and multi-intent contrastive learning. It emphasizes intent-oriented augmentation and contrastive constraints, whereas MiRec performs intra-scale interest decomposition and uses the cross-scale interest mixer to progressively propagate stable and dynamic interest representations in opposite directions across scales.
- **MRFF** (Zhang et al., 2025a) combines group-level and user-specific feed-forward networks through group gating to learn multifaceted user preferences in a federated framework. It focuses on personalization at the user and group levels, whereas MiRec performs intra-scale interest decomposition and cross-scale interest mixing within an individual user's sequence.

Time-aware and multi-scale modeling.

- **HM4SR** (Zhang et al., 2025b) uses interactive and temporal mixtures of experts to select multimodal information and incorporate time intervals and absolute time into sequence modeling. Its hierarchy is organized around modalities and temporal conditions, whereas MiRec distinguishes stable and dynamic interests within each temporal scale and performs cross-scale interest mixing through subspace exchange.
- **HyperHawkes** (Peintner et al., 2025) extracts intents through hypergraphs and contrastive clustering, models long-term repeated consumption with Hawkes processes, and incorporates short-term interests learned by an attention network. It primarily combines long-

and short-term signals at the prediction-score level, whereas MiRec performs intra-scale interest decomposition and directional, progressive cross-scale interest mixing at the representation level.

- **MIRRN** (Xu et al., 2025) retrieves interest subsequences at multiple granularities using target-aware, local-aware, and global-aware queries, and refines interests through a Multi-Head Fourier Transformer and target attention. It focuses on candidate-specific interest retrieval and relationship modeling within subsequences, whereas MiRec focuses on intra-scale interest decomposition and cross-scale interest mixing through progressive subspace exchange between adjacent scales.
- **MIMAR-SRec** (Fang & Shi, 2026) constructs interest subsequences at different temporal granularities and learns their representations through semantic enhancement, intent-prototype and cross-user alignment, and intent-consistent adversarial augmentation. It emphasizes intent alignment and robust training, whereas MiRec further distinguishes stable and dynamic interests within each scale and progressively propagates their representations in opposite directions.
- **MoS** (Lin et al., 2026) uses theme-aware routing to extract behavioral subsequences and combines global, item, and window experts to model thematic information and temporal context at different granularities. It primarily relies on expert specialization and output fusion, whereas MiRec progressively propagates stable and dynamic interest representations in opposite directions through learnable subspace exchange.

C.5 Statistical Significance Analysis

We further evaluate MiRec's performance advantage through cross-dataset rank comparisons and statistical tests. We compute average ranks for 18 methods across four datasets, separately for HR@10 and NDCG@10; lower ranks indicate better performance. The Friedman test gives chi-square statistics of 55.9649 and 55.4035, with approximate p values of 4.79×10^{-6} and 5.90×10^{-6}, respectively. We then apply the Nemenyi post-hoc test for pairwise comparisons, using $\alpha = 0.05$ and a critical difference of $\mathrm{CD} = 13.1695$.

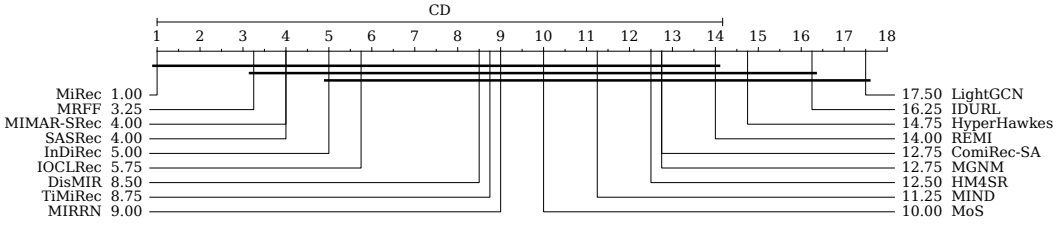


Figure 7: Critical Difference Diagram on HR@10

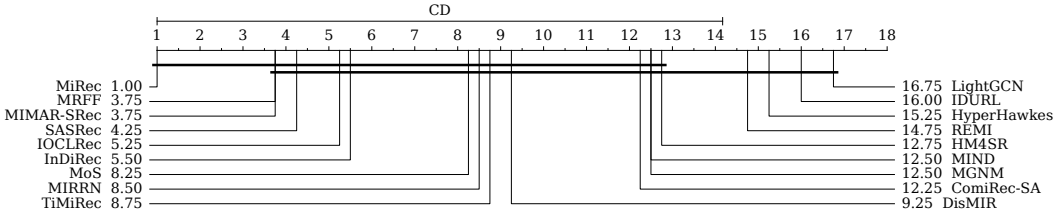


Figure 8: Critical Difference Diagram on NDCG@10

MiRec ranks first on both metrics across all four datasets, demonstrating a consistent recommendation advantage. As shown in Figures 7 and 8, MiRec achieves an average rank of 1.00 on both HR@10 and NDCG@10, leading all compared methods. On HR@10, MRFF ranks second with an average rank of 3.25, while MIMAR-SRec and SASRec both obtain 4.00. On NDCG@10, MRFF and MIMAR-SRec tie for second place at 3.75.

The Nemenyi test further supports MiRec's significant advantages over several baselines. MiRec significantly outperforms HyperHawkes, IDURL, and LightGCN on HR@10, and REMI, HyperHawkes, IDURL, and LightGCN on NDCG@10. Together with its best average ranks on both metrics, these results further support the effectiveness of MiRec and demonstrate its consistent performance advantage across the four datasets.

C.6 Additional Ablation Results

Table 5 reports the results on Kindle and Video for the four variants defined in Section 4.3. MiRec achieves the best HR@10 and NDCG@10 on both datasets, consistent with the results on Beauty and ML-1M. On Video, MiRec improves HR@10 and NDCG@10 over w/o Mixing by 3.01% and 4.31%, respectively, further supporting cross-scale interest mixing. Compared with Fixed Exch., the improvements are 1.72% and 2.39%; compared with Non-prog. Mix., they are 1.75% and 2.21%. These results further support learning subspace correspondences and progressively integrating information across scales.

Table 5: Ablation results on Kindle and Video. Best results are in bold.

Variant	Kindle		Video	
	HR@10	NDCG@10	HR@10	NDCG@10
MiRec	**0.6805**	**0.5441**	**0.7574**	**0.5565**
w/o Decomp.	0.6735	0.5350	0.7411	0.5424
w/o Mixing	0.6698	0.5350	0.7353	0.5335
Fixed Exch.	0.6707	0.5357	0.7446	0.5435
Non-prog. Mix.	0.6736	0.5365	0.7444	0.5445

Further Ablation Experiments. Building on the module-level ablations in the main text, we further examine decomposition operators, the two interest branches, and cross-scale mixing mechanisms. Table 6 summarizes the additional variants on four datasets.

Table 6: Additional ablation results on four datasets. The best and second-best mean results are in bold and underlined, respectively.

Variant	**Beauty**		**Kindle**		**ML-1M**		**Video**	
	HR@10	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10
DualProj	0.4844	0.3366	0.6711	0.5359	0.8003	0.5624	0.7443	0.5437
EMA-Decomp	0.4850	0.3382	0.6688	0.5356	0.8006	0.5632	0.7437	0.5437
Learnable-Decomp	0.4853	0.3380	0.6695	0.5355	0.8029	<u>0.5638</u>	0.7447	0.5447
w/o S-Mix	<u>0.4898</u>	0.3403	0.6693	0.5292	0.7989	0.5601	0.7468	<u>0.5465</u>
w/o D-Mix	0.4878	0.3397	0.6718	0.5372	0.8033	0.5626	<u>0.7469</u>	0.5451
Reversed Directions	0.4895	0.3403	0.6719	0.5365	0.7993	0.5629	0.7456	0.5439
Row Softmax	0.4888	0.3396	0.6698	0.5363	<u>0.8040</u>	<u>0.5638</u>	0.7447	0.5445
Attn-Mix	0.4882	<u>0.3404</u>	<u>0.6728</u>	0.5369	0.8013	0.5635	0.7463	0.5459
SharedProj	0.4877	0.3392	0.6690	0.5334	0.8010	0.5603	0.7446	0.5450
SharedEx	0.4874	0.3389	0.6709	0.5358	0.8034	0.5636	0.7448	0.5445
SharedFFN	0.4876	0.3388	0.6721	<u>0.5407</u>	0.7987	0.5603	0.7461	0.5459
MiRec	**0.5007**	**0.3461**	**0.6805**	**0.5441**	**0.8078**	**0.5693**	**0.7574**	**0.5565**

To test whether explicit decomposition offers benefits beyond a two-branch architecture, DualProj feeds the same undecomposed representation into two independent projections while retaining both mixing paths and the same parameter count. MiRec improves HR@10 and NDCG@10 over DualProj by 3.3579% and 2.8122% on Beauty, respectively, supporting the value of separating temporal components rather than simply introducing two projections. EMA-Decomp replaces the local mean with an exponential moving average, while Learnable-Decomp learns normalized local smoothing weights. Both retain residual extraction. Neither replacement improves on MiRec across the four datasets, and Learnable-Decomp does not consistently outperform EMA-Decomp in NDCG@10, suggesting that a more flexible smoothing operator alone is insufficient to improve recommendation.

The w/o S-Mix and w/o D-Mix variants remove the stable and dynamic mixing paths, respectively, to examine their individual contributions. MiRec outperforms both variants on all four datasets, supporting their joint use. Their relative contributions differ: on Kindle, retaining the stable path (w/o D-Mix) yields NDCG@10 of 0.5372, compared with 0.5292 when retaining the dynamic path (w/o S-Mix). On Video, the latter is better, reaching 0.5465 versus 0.5451. These differences show that the contributions of the two mixing paths vary by dataset.

Reversed Directions swaps the propagation directions of the stable and dynamic branches, while Row Softmax replaces Sinkhorn normalization with row-wise softmax. These variants test the assignment of propagation directions and the role of column balancing without changing parameter count. Both obtain lower results than MiRec across all four datasets. On ML-1M, their NDCG@10 values are 0.5629 and 0.5638, compared with 0.5693 for MiRec, supporting the proposed direction assignment and normalization within the Cross-scale Interest Mixer.

Attn-Mix replaces the exchange operator with content-dependent subspace attention to examine whether more flexible mixing provides benefits. It performs below MiRec on all four datasets, supporting the effectiveness of structured subspace exchange for cross-scale interest mixing.

Finally, SharedProj shares the stable and dynamic projections, SharedEx shares their exchange matrices for each adjacent scale pair, and SharedFFN shares the post-mixing transformation across scales. These variants test the need for branch- and scale-specific parameterization. All three obtain lower results than MiRec, although their relative rankings vary. For example, SharedFFN exceeds SharedEx in NDCG@10 on Kindle (0.5407 versus 0.5358), but falls below it on ML-1M (0.5603 versus 0.5636). The results favor retaining specialized transformations, while parameter sharing also reduces model capacity. Overall, the additional ablations support explicit within-scale decomposition and differentiated cross-scale mixing, while showing that alternative mixing operators can offer dataset-specific advantages.

C.7 Additional Visualizations of Interest Decomposition

Visualization Setup. With MiRec frozen, we train separate, identically structured linear heads on the three pre-exchange representations at scale 16, using user-disjoint splits for the heads. The target is the next item's normalized multi-label genre distribution. Behavior-selected cases span 32 positions, partitioned into 11, 11, and 10 positions. Eight genres share a linear color scale of $[0, 0.68]$, without probability renormalization.

Additional Cases. Figure 9 illustrates persistent preferences and a temporary category excursion. User 2101 consistently interacts with Drama movies; the stable branch maintains high mean Drama weights of 0.5517, 0.5599, and 0.5959 across the three segments. For user 4926, Comedy interactions increase in the middle segment and then decrease. The dynamic branch follows this local pattern, with mean Comedy weights rising from 0.1937 to 0.2558 before declining to 0.1866. These cases further illustrate the branches' different temporal behaviors. The auxiliary predictions describe genre tendencies in the representations, and the case-level observations apply to the displayed sequences.

C.8 Analysis of Interest Representations Before and After Cross-scale Mixing

This section examines whether cross-scale interest mixing further differentiates the roles of the two branches: maintaining smooth representation changes in the stable branch and producing more distinct responses to behavioral changes in the dynamic branch. We select 1,000 users from ML-1M and extract the stable and dynamic representations before and after exchange at the scale with window size 4 from the same trained MiRec model. We analyze both the magnitude of representation changes and their relationship with behavioral novelty.

The stable branch exhibits smoother changes after mixing. Before and after mixing, we separately compute the cosine distance between representations at consecutive interactions of the same user; a smaller distance indicates less variation as interactions progress. As shown in Figure 10(a), the stable branch's mean cosine distance decreases from 0.0420 to 0.0191, while that of the dynamic

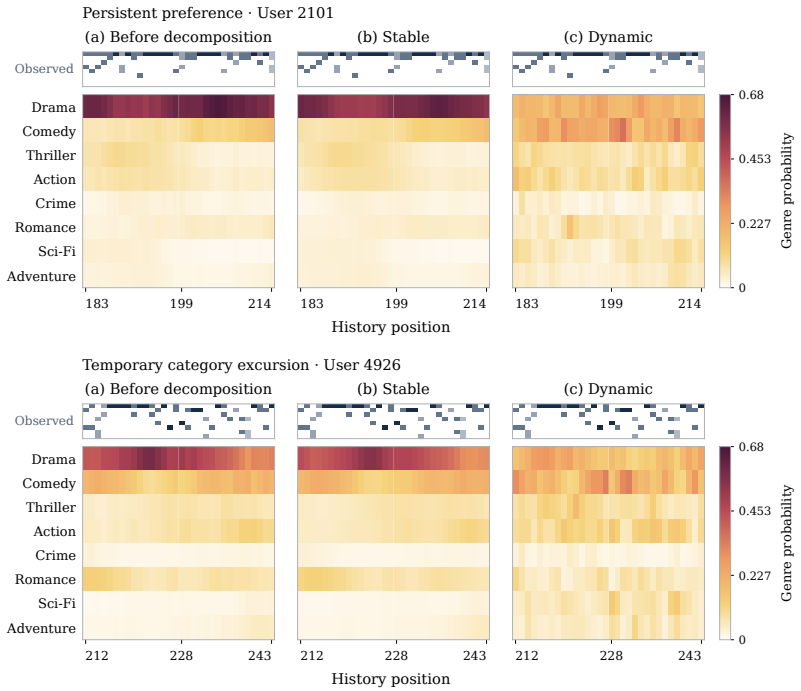


Figure 9: Additional decomposition cases on ML-1M. Top: persistent preferences of user 2101. Bottom: a temporary category excursion of user 4926. Each row shows auxiliary genre predictions before decomposition and from the stable and dynamic branches.

branch changes from 0.3610 to 0.3467. After mixing, the stable branch becomes smoother, while the dynamic branch retains greater variation.

The dynamic branch exhibits a more pronounced response contrast between different levels of behavioral novelty after mixing. We measure behavioral novelty using the Jensen–Shannon divergence between the current movie's genre distribution and the average genre distribution of the preceding eight movies. A larger value indicates a greater difference between the current interaction and recent behavior. For each user, we form high- and low-novelty groups using the top and bottom 25% of interactions ranked by novelty, compute the difference in their mean cosine distances, and then average these differences across users. A positive difference indicates greater representation changes for high-novelty interactions. As shown in Figure 10(b), mixing increases the dynamic branch's response contrast from 0.0238 to 0.0336, while reducing that of the stable branch from 0.0097 to 0.0040. After mixing, 66.3% of users have a positive dynamic response contrast. These results show that, at the examined scale, the stable branch becomes smoother and the dynamic branch exhibits a more distinct response to behavioral novelty, consistent with their intended roles.

Figure 11 presents user cases in which persistent viewing tendencies coexist with local interest changes. For example, user 873 repeatedly watches comedy and drama movies, briefly explores mystery content, and then returns to the recurring genres. User 2312 shifts from action and adventure movies to musicals, war movies, and dramas, exhibiting changes in focus across successive stages. These behaviors call for modeling both recurring interest patterns and changes introduced by recent interactions. To model these two types of information, MiRec first extracts relatively stable components and their residual variations at each scale, forming stable and dynamic branches. The cross-scale interest mixer then performs subspace exchange separately within each branch, integrating components of the same type from different historical ranges. In the post-mixing representations shown in Figure 16, the stable branch maintains small changes as interactions progress, whereas the dynamic branch exhibits more pronounced fluctuations. These cases illustrate the distinct variation patterns of the two representations after mixing, consistent with the intended roles of preserving persistent information in the stable branch and reflecting local changes in the dynamic branch. They also complement the quantitative comparison before and after mixing in Figure 10.

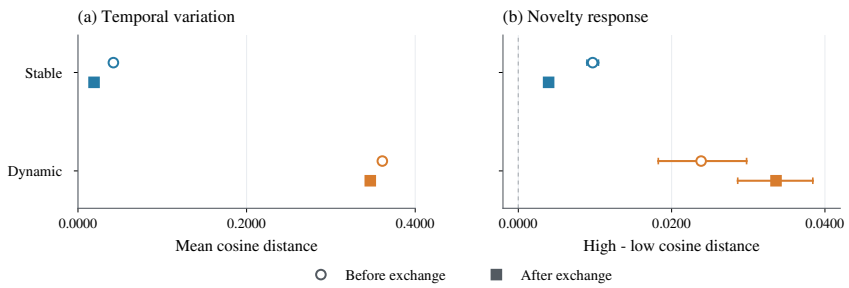


Figure 10: Analysis of branch representations before and after cross-scale exchange. (a) Mean cosine distance between representations at consecutive interactions. (b) Difference in mean cosine distance between high- and low-novelty groups; positive values indicate greater representation changes for high-novelty interactions.

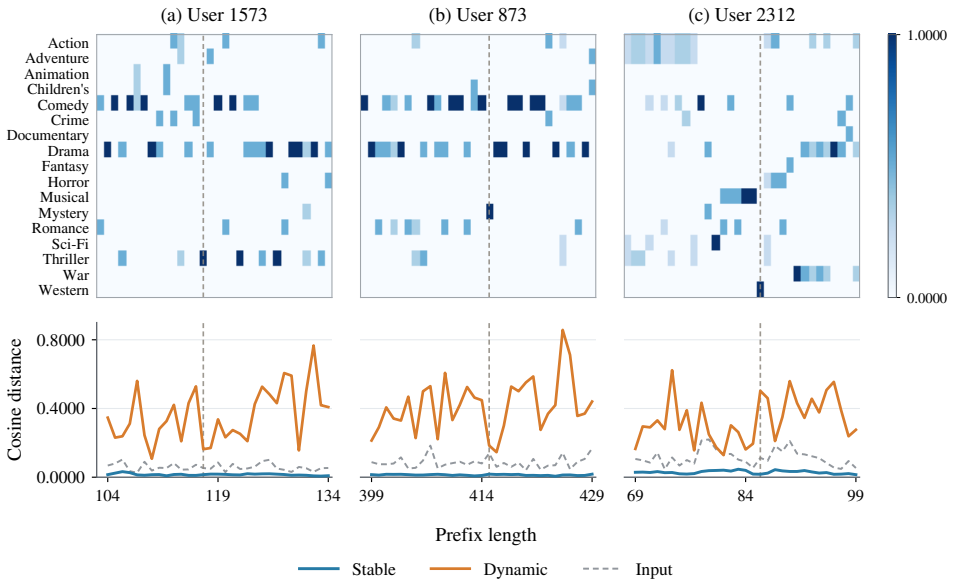


Figure 11: Individual trajectories on ML-1M at window size 4. Top: genre distributions of observed movies, with equal mass assigned to each movie's genres. Bottom: consecutive-step cosine distances for the stable and dynamic branches after exchange and for the normalized decomposition input. Vertical dashed lines mark the highest genre-novelty position in each displayed sequence. Cases are selected using behavioral variation without inspecting model responses.

C.9 Subspace Correspondences on Additional Datasets and Scales

Figure 12 shows the corresponding exchange matrices for Beauty, Kindle, and Video at the same window pair as the ML-1M visualization in Figure 4. All panels use seed 2026 and checkpoints selected by validation NDCG@10. The stable $16 \to 4$ transfers place more weight on same-index correspondences, whereas the dynamic $4 \to 16$ transfers allocate a larger fraction to off-diagonal entries. For this window pair, the dynamic branch has a larger off-diagonal mass fraction than the stable branch in each of the three seeds (2026, 2027, and 2028) on all four datasets.

Figure 13 summarizes all adjacent window pairs across three seeds. The diagonal mass fraction is computed within each run as the sum of diagonal entries divided by the sum of all matrix entries, and then averaged over seeds. We do not average matrix entries across independently trained models. The branch differences highlighted for the intermediate window pair in Figure 4 need not have the same magnitude or direction at other window pairs.

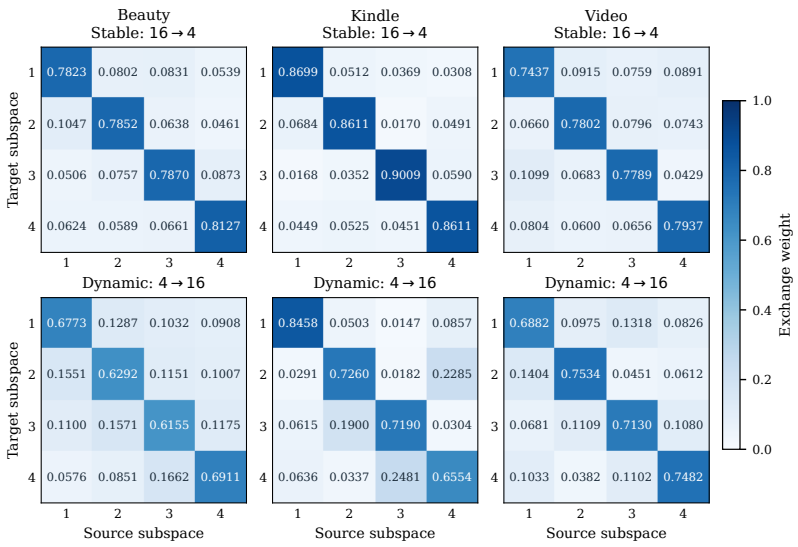


Figure 12: Learned subspace correspondences on Beauty, Kindle, and Video with seed 2026. The upper row shows stable transfers ($16 \to 4$), and the lower row shows dynamic transfers ($4 \to 16$). Rows and columns denote target and source subspaces, respectively. All panels use the same color scale, with values shown to four decimal places.

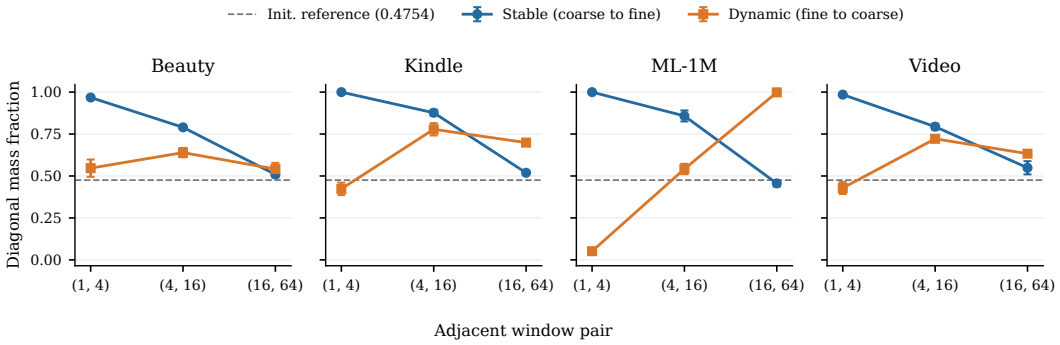


Figure 13: Diagonal mass fractions across all adjacent window pairs. Points and error bars denote the mean and sample standard deviation over seeds 2026, 2027, and 2028. Stable transfers proceed from the larger to the smaller window, while dynamic transfers proceed in the opposite direction. The dashed line is a reference reconstructed from the saved initialization settings, rather than a measured epoch-zero checkpoint. Window sizes denote configured values.

C.10 Additional Hyperparameter Sensitivity Analysis

This section examines MiRec's sensitivity to hyperparameters governing multi-scale construction, intra-scale decomposition, and cross-scale mixing. We vary the scale multiplier r, decomposition window k, number of subspaces K, mixer depth ℓ, exchange temperature τ, and number of Sinkhorn iterations I individually on Beauty, Kindle, ML-1M, and Video, while keeping the remaining settings fixed, and report HR@10 and NDCG@10.

Impact of Scale Multiplier r. A moderate scale multiplier performs best within the tested range. We use four scales with window sizes $\{1, r, r^2, r^3\}$ and evaluate $r \in \{2, 3, 4, 5\}$. As shown in Figure 14, $r = 4$ achieves the highest scores on both metrics across all four datasets. On Beauty, increasing r from 2 to 4 improves HR@10 from 0.4859 to 0.5007 and NDCG@10 from 0.3371 to 0.3461.

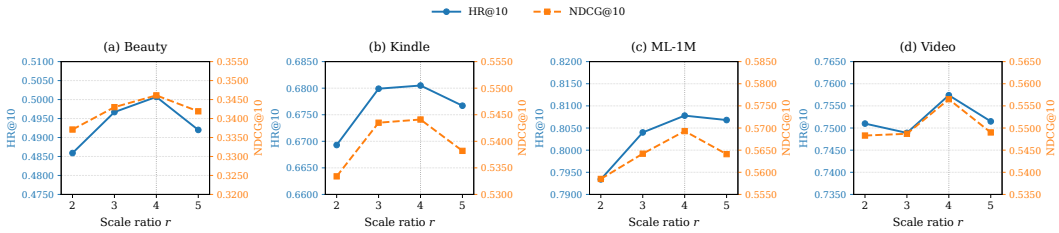


Figure 14: Sensitivity to the scale multiplier r. From left to right: Beauty, Kindle, ML-1M, and Video. Each panel reports HR@10 and NDCG@10.

Impact of Decomposition Window k. Smaller decomposition windows generally yield better results. We evaluate $k \in \{2, 3, 5, 7, 9\}$, with results shown in Figure 15. The setting $k = 3$ achieves the best results in seven of the eight dataset–metric combinations, including HR@10 on all four datasets. On Beauty, its NDCG@10 reaches 0.3461, only 0.0002 below the highest value obtained with $k = 2$. These results support the use of a compact window for intra-scale interest decomposition.

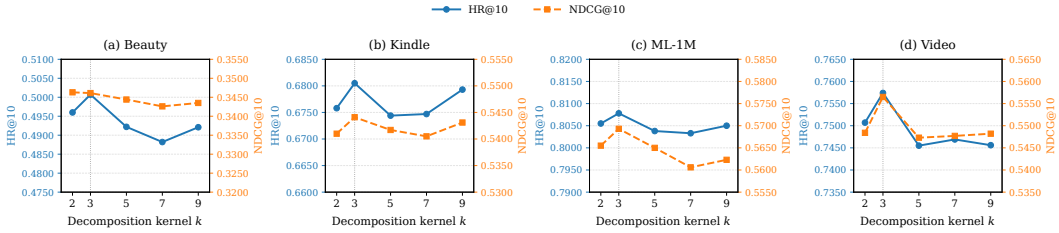


Figure 15: Sensitivity to the decomposition window k. From left to right: Beauty, Kindle, ML-1M, and Video. Each panel reports HR@10 and NDCG@10.

Impact of Number of Subspaces K. A moderate subspace partition performs better than an unpartitioned representation or finer partitions. We evaluate $K \in \{1, 2, 4, 5, 10\}$. As shown in Figure 16, $K = 4$ achieves the highest scores on both metrics across all four datasets. On Video, increasing K from 1 to 4 improves HR@10 from 0.7463 to 0.7574 and NDCG@10 from 0.5465 to 0.5565, supporting the effectiveness of subspace partitioning for cross-scale interest mixing.

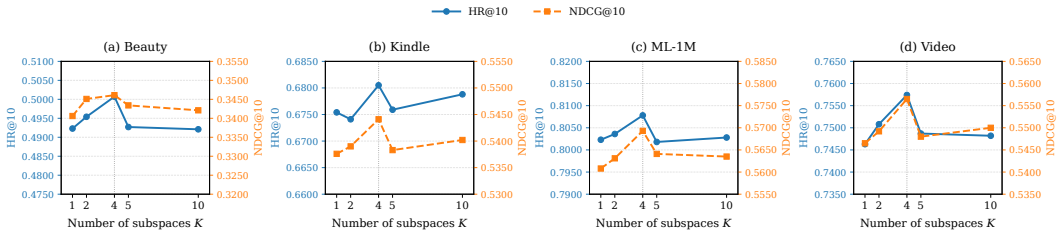


Figure 16: Sensitivity to the number of subspaces K. From left to right: Beauty, Kindle, ML-1M, and Video. Each panel reports HR@10 and NDCG@10.

Impact of Mixer Depth ℓ. A single mixer layer achieves the best results in seven of the eight dataset–metric combinations. We evaluate $\ell \in \{1, 2, 3, 4\}$, with results shown in Figure 9. Both metrics peak at $\ell = 1$ on Beauty, Kindle, and Video. On ML-1M, a single layer achieves the highest NDCG@10 of 0.5693, while its HR@10 of 0.8078 is close to the highest value of 0.8086 obtained with two layers. These results show that MiRec achieves strong recommendation performance with a compact, single-layer cross-scale interest mixer.

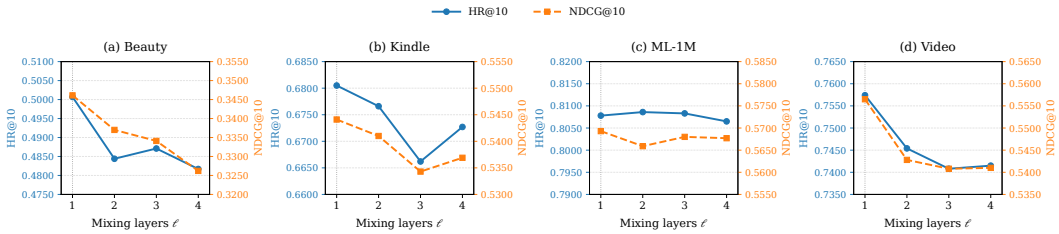


Figure 17: Sensitivity to the mixer depth ℓ. From left to right: Beauty, Kindle, ML-1M, and Video. Each panel reports HR@10 and NDCG@10.

Impact of Exchange Temperature τ. A moderate exchange temperature performs better on most dataset–metric combinations. We evaluate $\tau \in \{0.1, 0.3, 0.5, 0.7, 1.0\}$. As shown in Figure 18, the setting $\tau = 0.5$ achieves the best results in seven of the eight dataset–metric combinations. On Kindle, it achieves the highest NDCG@10 of 0.5441, compared with 0.5411 at $\tau = 1.0$, while its HR@10 is only 0.0003 below the highest observed value. These results support $\tau = 0.5$ as an effective setting across the four datasets.

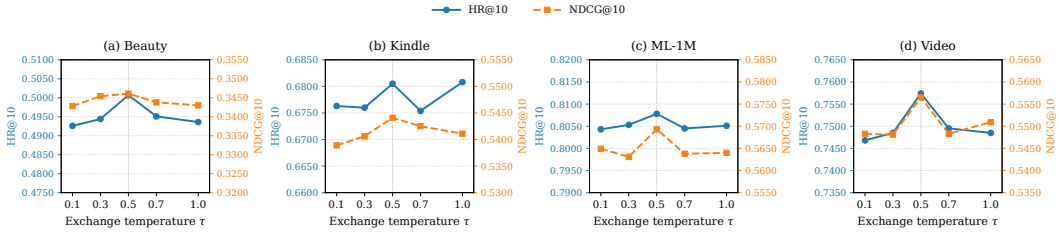


Figure 18: Sensitivity to the exchange temperature τ. From left to right: Beauty, Kindle, ML-1M, and Video. Each panel reports HR@10 and NDCG@10.

Impact of Number of Sinkhorn Iterations I. MiRec achieves its best recommendation performance with six Sinkhorn iterations. We evaluate $I \in \{1, 3, 6, 10\}$, with results shown in Figure 11. The setting $I = 6$ yields the highest HR@10 and NDCG@10 across all four datasets. On Kindle, it achieves 0.6805 and 0.5441, respectively, compared with 0.6744 and 0.5358 using ten iterations. These results show that the cross-scale interest mixer achieves its best tested performance with a modest number of normalization iterations.

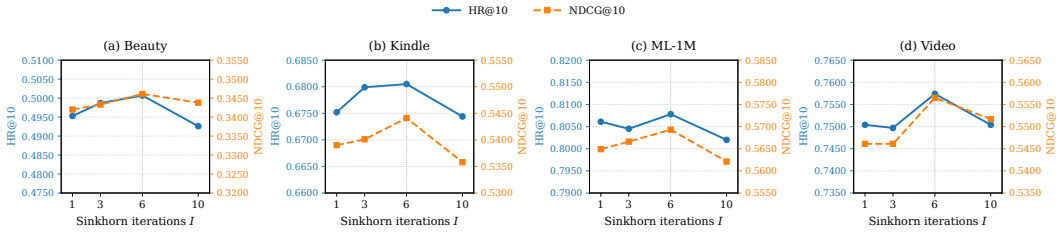


Figure 19: Sensitivity to the number of Sinkhorn iterations I. From left to right: Beauty, Kindle, ML-1M, and Video. Each panel reports HR@10 and NDCG@10.