MIDA: Multi-Interest Decoding with Adaptive Aggregation for Session-based Recommendation

Linjiang Guo, Nitin Bisht, Shiqing Wu, and Guandong Xu

© 2026 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

This is the author-accepted manuscript of an article published in *IEEE Transactions on Knowledge and Data Engineering (TKDE)*.

The final published version is available at IEEE Xplore.

DOI: 10.1109/TKDE.2026.3740693.

arXiv:2610.34192v1 [cs.IR] 25 Oct 2026

***Abstract*—Session-based recommendation (SBR) infers user interest from limited interaction sequences, yet current models show a consistent performance drop when longer histories are available. This degradation is associated with decoder designs that overemphasize recent items while underutilizing distant, yet semantically relevant, interactions, leading to poor modeling of long-range dependencies and multi-interest behaviors. To address this challenge, we introduce MIDA (Multi-Interest Decoding with Adaptive Aggregation for Session-based Recommendation), a lightweight and encoder-agnostic framework that provides a reformulation of the decoding stage. The central design principle is a *two-space decoupling*: clustering operates on raw item embeddings so that semantic groupings are not distorted by encoder-learned relational biases; encoder outputs are reserved for aggregation, where sequential context remains intact. Concretely, MIDA (i) employs differentiable multi-interest clustering on raw embeddings, preserving semantic structure that encoder representations would otherwise obscure, and (ii) fuses stability and plasticity branches through a gate conditioned on average cluster size, so that the inferred clustering structure directly governs aggregation behavior through a structural coupling across the two stages. Extensive experiments on multiple benchmark datasets demonstrate that MIDA consistently outperforms state-of-the-art methods, with meaningful accuracy improvements while remaining computationally practical.**

***Index Terms*—Session-Based Recommendations, Dynamic Aggregation, Multi-Interest, Session Representation**

I. Introduction

Recommendation systems (RS) [1]–[3] have become ubiquitous in modern online platforms, delivering personalized and timely content by learning from user behavior. In practice, persistent user profiles are often unavailable, or only interactions within limited time windows can be observed. Session-based Recommendation (SBR) [4]–[6] provides an effective solution: it infers user interest from the sequence of interactions in a single session and predicts the next item of interest, with particular relevance to e-commerce, news, and video streaming, where users browse anonymously or interact through temporary identifiers.

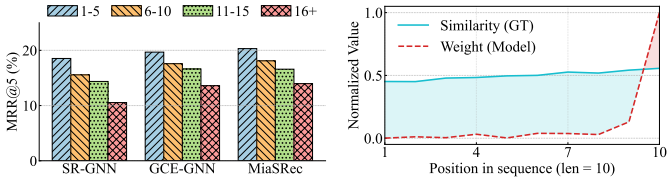


Fig. 1: Analysis of decoder limitations in existing session-based RS. *Top*: MRR@5 as a function of session length across state-of-the-art models, with longer sessions yielding lower accuracy despite containing richer information. *Bottom*: Attention weight distribution (red dashed) versus ground-truth item cosine similarity to the target (blue solid) in a length-10 session, providing evidence for a strong recency bias. Results are generated on the Diginetica dataset using an SR-GNN backbone; attention weights correspond to the single-layer decoder attention.

Recently, SBR has been advanced through graph-based [7], [8] and transformer-based architectures [9] that adopt an encoder-decoder framework, enabling more effective modeling of complex item dependencies. Prior research concentrates heavily on the encoder mechanism to capture both local and global relationships among items [10]–[12]. Related multi-intent methods further enhance session representations by constructing or selecting multiple intent representations within model-specific architectures [13]–[16]. The decoder mechanism, which ultimately determines how encoded information is transformed into user preference representations, has received comparatively little attention.

Some recent research utilizes pooling-based methods [13], [17], which compress all items into a fixed-length embedding. While simple and efficient, this approach inevitably discards sequential and item-level distinctions, becoming particularly harmful in long sessions where diverse user interests emerge. As a result, pooling-based decoders struggle to preserve the temporal information necessary for accurate next-item prediction. Another line of work focuses on point-wise approaches [12], which rely on attention or gating anchored on the last item to balance short-term and long-term intents. However, this design overweights recent interactions while underutilising earlier but semantically relevant ones, ultimately failing to capture the multi-interest nature of long sessions.

Linjiang Guo and Nitin Bisht contribute equally to this work.

L. Guo and N. Bisht are with the University of Technology Sydney, Sydney, Australia. E-mail: {linjiang.guo, nitin.bisht}@student.uts.edu.au.

S. Wu is with the City University of Macau, Macau SAR. E-mail: sqwu@cityu.edu.mo.

G. Xu is with the Education University of Hong Kong, Hong Kong SAR. E-mail: gdxu@eduhk.hk.

S. Wu and G. Xu are corresponding authors.

These limitations raise an important concern: do existing decoders truly exploit the richer historical context available in long sessions? To investigate, we conduct an empirical analysis on representative models, as shown in Fig. 1. Point-wise approaches assign disproportionate weight to the last few items while ignoring earlier ones; pooling-based approaches blur temporal signals across the sequence. Both weaknesses manifest in a consistent empirical pattern: when sessions are grouped by length, longer sessions produce lower accuracy across state-of-the-art models, despite containing richer historical information. Notably, this degradation is observed across architecturally distinct model families (GRU-based, graph-based, and Transformer-based) that differ in every encoder design dimension yet share the same decoder paradigm. This cross-architecture consistency cannot be attributed to any specific encoder design; rather, it implicates the shared decoder mechanism as the primary source of underperformance on longer sessions. To isolate the decoder's contribution, we adopt a controlled evaluation protocol (Section V): we hold each encoder fixed and vary only the decoder component across all baselines, enabling a direct assessment of decoder-side impact while factoring out encoder differences. The central research question is therefore: *How can we design a lightweight, encoder-agnostic decoder that effectively exploits diverse historical contexts to model user interests?*

To answer this, we introduce **MIDA** (Multi-Interest Decoding with Adaptive Aggregation for Session-based Recommendation) [1], a lightweight and encoder-agnostic framework that provides a reformulation of the decoding stage to tackle the above limitations by modeling both persistent and evolving aspects of user intent. The two empirical failure modes of existing decoders (pooling methods that conflate semantically distinct items and point-wise methods that discard all but the most recent interactions) translate into two concrete design challenges: (a) How can we capture diverse persistent interests that are semantically related but temporally distant? (b) How can we balance stability (persistent intent) with plasticity (evolving intent) in user behavior?

To address these challenges, MIDA employs *dynamic multi-interest clustering*, which groups semantically related items via differentiable similarity thresholds, thereby preserving long-range dependencies and capturing diverse user preferences that pooling-based methods collapse. It also introduces *adaptive aggregation*, which disentangles user intent into stability and plasticity and fuses them through a gating mechanism conditioned on both branch representations and semantic cohesion signals (i.e. average cluster size), enabling fine-grained adaptation to consistent and evolving behavioral signals. Together, these components yield a decoder that efficiently utilizes history information, retains temporal structure, and models the multi-interest nature of complex sessions. The framework is illustrated in Fig. 2.

The contributions of this work are as follows.

1) We identify the *decoder bottleneck* in SBR, namely the systematic tendency of existing decoders to underutilize long-range historical context, and provide a lightweight, encoder-agnostic reformulation of the decoding stage that directly addresses this limitation.
2) A differentiable clustering mechanism with regularisation is proposed for dynamic interest extraction, together with a stability–plasticity aggregation scheme that disentangles persistent and evolving user intents, emphasizing the *decoder-side perspective*.
3) The framework incorporates a two-stage decoder design in which semantic similarity computation is decoupled from relational representation: Dynamic Multi-Interest Clustering operates on raw embeddings to identify similar items across temporal distances, while Adaptive Aggregation uses encoder outputs to construct attention-based session representations that capture sequential dependencies. The decoder is encoder-agnostic: it replaces only the final aggregation step of any backbone without modifying encoder parameters.
4) Comprehensive experiments on multiple benchmark datasets show that MIDA outperforms strong baselines across diverse encoder architectures.

[1]Our code is available at: https://github.com/downw/MIDA_Rec

II. Related Work

a) Session-based recommendation.: Traditional SBR methods typically rely on heuristic or shallow models such as Markov chains to model item transitions [18]. Recent studies increasingly focus on deep neural architectures, including recurrent networks [19], [20], attention mechanisms [21], and graph neural networks [6], [7], [22], which effectively encode complex transition dependencies within sessions. Several works aim to integrate both local and global item relations [23], or leverage contrastive learning to enhance representation robustness [22]. Further explorations extend to Transformer-based frameworks [24], [25] and memory-augmented architectures [26], [27]. Side information has also emerged as a promising direction, including temporal signals [28], item descriptions [25], user behaviors [29], [30], and category information [31]. Across these studies, the encoder receives the bulk of attention, while the decoder remains largely fixed.

b) Multi-interest modeling.: There is growing interest in capturing multi-interest structures within sessions. One approach segments item embeddings into multiple parts, each representing a distinct interest [14]. Another treats items as multi-aspect entities mapped into multiple candidate spaces corresponding to different aspects [13], [15]. Dynamic interest adaptation mechanisms for automatically inferring the number of interests have also been explored [16]. MIDA is conceptually related to these lines of work, but differs in two important ways. First, it employs a dynamic, differentiable clustering mechanism based on within-session semantic similarity, grouping interests adaptively without a preset number or heavy overhead. Second, it applies multi-interest modeling at the *decoder stage* and further disentangles persistent from evolving intents, enabling richer utilisation of historical information.

c) Decoder methods in recommendation.: Decoder-based recommendation methods differ in how user preferences are

generated. Session-based decoders may form a user interest representation from the last interacted item [32], global session aggregation [33], or temporal exponential decay sum [34]. CTR prediction decoders score each item in a fixed candidate set via binary classifiers [35]. Sequential recommendation approaches such as LightGCN [36] bypass explicit decoding by computing relevance via neighborhood embedding aggregation. Recent work employs Deep Q-Network agents to optimize graph connectivity [37] or multi-scale rescaling of positional indices to mitigate the "lost-in-the-middle" problem [38]. Accurately capturing user interests is especially important because scoring every candidate item is impractical; imprecise interest representations can mislead the decoder [39], particularly when user intents shift during a session [40], [41]. MIDA addresses these challenges by reformulating the decoder as a dual-branch system grounded in within-session semantic structure.

Unlike existing methods that either treat the encoder and decoder jointly or focus on encoder redesign, MIDA treats the decoder as a first-class component while remaining encoder-agnostic. Specifically, MIDA replaces only the final aggregation module of a backbone (e.g. the attention-based pooling in SR-GNN/GCE-GNN or the output projection in GRU4Rec), preserving all encoder parameters and training schedules. This modular design enables straightforward integration across heterogeneous encoder families. The combination of differentiable threshold-based clustering (without a pre-specified number of interests), a stability and plasticity dual-query design, and a semantically conditioned gating mechanism distinguishes MIDA from multi-interest methods that operate at the encoder level and from decoder methods that rely on a single fixed aggregation rule. In particular, unlike AttenMixer [42], which introduces multi-level propagation layers that can conflict with rich encoders, MIDA's decoupled query construction explicitly separates semantic similarity (raw embeddings) from relational context (encoder outputs), avoiding gradient interference while retaining the benefits of both representations. The essential novelty of MIDA lies not in assembling independently existing modules but in a principled two-space decoupling: clustering operates on raw item embeddings so that semantic groupings are not distorted by the relational biases the encoder has learned; encoder outputs are reserved for aggregation, preserving the sequential dependencies the encoder provides. This is an architectural consequence of separating which items are related from how they should be aggregated given session context, a distinction no existing decoder draws explicitly.

III. The Proposed Framework

We propose **MIDA** (**M**ulti-**I**nterest and A**d**aptive **A**ggregation for Session-based Recommendation), a lightweight decoder framework that models both long-range, multi-interest dependencies and short-term preference shifts in SBR. An overview is illustrated in Fig. 2.

A. Notation

Table I summarises the key symbols used throughout this paper.

TABLE I: Key Notation

Symbol	Meaning
S	Session (ordered item sequence)
L	Session length ($\|S\|$)
d	Embedding dimension
τ	Similarity threshold (global learnable parameter)
k	Embedding-dimension subsampling stride
$\mathbf{h}_i \in \mathbb{R}^d$	Raw item embedding
$\mathbf{h}_i^{\text{enc}} \in \mathbb{R}^d$	Encoder output embedding
$\mathbf{M} \in \mathbb{R}^{L\times L}$	Pairwise cosine similarity matrix
$w_{ij} \in \mathbb{R}$	Soft cluster weight between items i and j
C_i	Cluster of item v_i
e	Average cluster size ($\bar{c} = \frac{1}{L}\sum_i \|C_i\|$)
$\tilde{\mathbf{h}}'_i \in \mathbb{R}^{\lceil d/k \rceil}$	Subsampled, ℓ_2-normalized embedding
$\hat{\mathbf{h}}_i \in \mathbb{R}^d$	Cluster-enhanced embedding of item v_i
$\mathbf{q}^{st} \in \mathbb{R}^d$	Stability query (sums over items $1,\dots,L-1$)
$\mathbf{q}^{pl} \in \mathbb{R}^d$	Plasticity query (anchored to last item v_L)
$\mathbf{h}_S \in \mathbb{R}^d$	Session-level representation

B. Problem Formulation

Given a session $S = [v_1, v_2, \dots, v_L]$, where each $v_i \in \mathcal{V}$ is the i-th interacted item and L is the session length, each item v_i is mapped to a raw embedding $\mathbf{h}_i \in \mathbb{R}^d$ that captures semantic information, where d is the embedding dimension. The encoder then produces a contextualised embedding $\mathbf{h}_i^{\text{enc}} \in \mathbb{R}^d$ that captures sequential dependencies. The task of SBR is to predict the next item v_{L+1} given the history S:

$$\begin{aligned} \mathbf{h}_S &= f_{\text{dec}}(\mathbf{h}^{\text{enc}}), \\ p(v \mid S) &= \frac{\exp(\mathbf{h}_S \cdot \mathbf{h}_v)}{\sum_{i\in\mathcal{V}} \exp(\mathbf{h}_S \cdot \mathbf{h}_i)}, \\ \hat{v}_{L+1} &= \arg\max_{v\in\mathcal{V}} p(v \mid S), \end{aligned} \tag{1}$$

where f_{dec} refers to the decoding function and $\mathbf{h}_S \in \mathbb{R}^d$ is the session-level representation.

C. Dynamic Multi-Interest Clustering

In existing pooling-based or point-wise decoding methods, $\mathbf{h}_S$ is either compressed into a single vector or overly focused on the last items, limiting the ability to recover interests that recur after long temporal gaps. outputs.

To uncover related yet temporally distant signals, we compute similarity using raw item embeddings rather than encoder outputs. This is because encoder outputs can mix item identity with backbone-specific relational and other signals (e.g. sequential dependencies or collaborative filtering signals), leading to conflation between real affinity with session context. Each raw embedding $\mathbf{h}_i \in \mathbb{R}^d$ is ℓ_2-normalized to remove magnitude effects:

$$\mathbf{h}'_i = \frac{\mathbf{h}_i}{\|\mathbf{h}_i\|_2}. \tag{2}$$

To improve computational efficiency, we apply a subsampling transform to $\mathbf{h}'_i$, producing $\tilde{\mathbf{h}}'_i$. Specifically, embeddings are sampled every k dimensions (e.g. dimensions $[1, 1{+}k, 1{+}2k, \dots]$). This stride is used only to form a lower-dimensional view for pairwise similarity estimation in the clustering module; it does not subsample session positions or truncate the encoded session representation used by the final

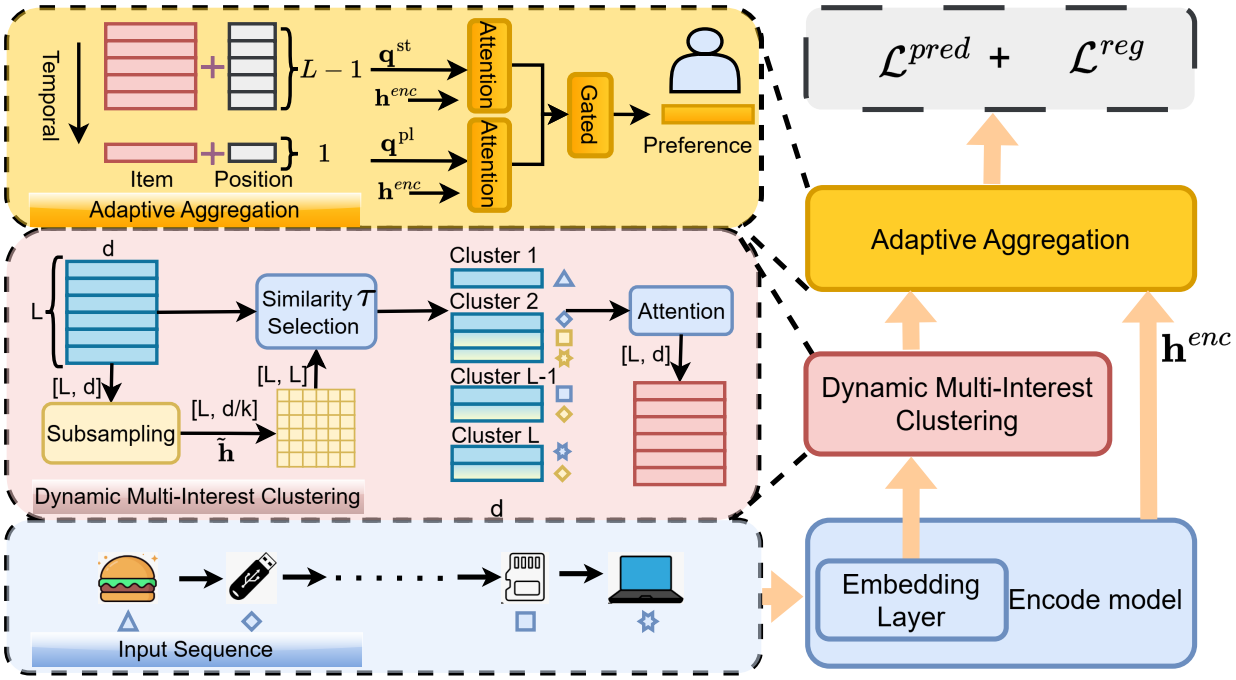


Fig. 2: Overview of the proposed MIDA framework. The model first extracts raw embeddings from the embedding layer, which are clustered by the Dynamic Multi-Interest Clustering module to retain semantically related yet temporally distant signals under a learned threshold τ. The Adaptive Aggregation module then decouples these signals into plasticity (evolving) and stability (persistent) intents, which interact with encoder outputs and are adaptively fused via a gating mechanism for final prediction.

prediction layer. We set $k = 4$ as a practical trade-off between clustering resolution and computational cost. Pairwise cosine similarities are then computed as

$$\mathbf{M}_{ij} = \tilde{\mathbf{h}}_i'^{\top} \tilde{\mathbf{h}}_j', \quad \mathbf{M} \in \mathbb{R}^{L \times L}, \tag{3}$$

where $1 \le i, j \le L$ index items within the same session. This similarity matrix serves as the basis for forming clusters of items that are semantically coherent even if they are far apart temporally.

a) Cluster definition.: For each item v_i, its cluster is defined as

$$C_i = \{\, j \mid \mathbf{M}_{ij} \ge \tau \,\}, \tag{4}$$

which includes all items whose cosine similarity to v_i exceeds the threshold τ. *No top-k truncation is applied.* Each item is included in its own cluster (i.e. $i \in C_i$), and cluster size is naturally bounded by the session length L.

b) Learnable threshold.: τ is a *global learnable parameter optimized via backpropagation*, shared across all sessions in the dataset. Since a hard threshold would make clustering non-differentiable, we introduce soft cluster weights $\tilde{w}_{ij} \in \mathbb{R}$:

$$\tilde{w}_{ij} = \max\left\{ \sigma_s\left(\frac{\mathbf{M}_{ij} - \tau}{\gamma} \right), \epsilon \right\}, \tag{5}$$

where $\sigma_s(\cdot)$ is the sigmoid function, γ is a temperature coefficient, and ϵ is a small constant that prevents weights from becoming numerically zero even for weakly related items.

c) Cluster-enhanced embeddings.: While clustering relates each item to its semantic neighborhood, it does not explicitly model how items within the same cluster interact. To capture such local co-occurrence patterns, we apply a shared position-aware attention over the weighted neighbors. Following the SASRec [43] design, each item representation is first augmented with a positional embedding:

$$\tilde{\mathbf{h}}_j = \mathbf{h}_j + \mathbf{p}_j', \tag{6}$$

where $\mathbf{p}_j' \in \mathbb{R}^d$ is the positional embedding for position j in the cluster. Attention is then computed over $\{\tilde{\mathbf{h}}_j : j \in C_i\}$ with the soft weights $\tilde{w}_{ij}$, and the output passes through a ReLU activation:

$$\hat{\mathbf{h}}_i = \mathrm{ReLU}\left(\sum_{j \in C_i} \frac{\tilde{w}_{ij}\, \tilde{\mathbf{h}}_j}{\sum_{j' \in C_i} \tilde{w}_{ij'}} \right), \quad \hat{\mathbf{h}}_i \in \mathbb{R}^d. \tag{7}$$

Semantically related signals that may be temporally distant in the session are thus preserved and made more salient, yielding a richer embedding space $\{\hat{\mathbf{h}}_i\}$ for subsequent decoding.

D. Adaptive Aggregation

A remaining challenge is how to balance user intents. If the decoder overemphasizes stability (persistent intents), it fails to adapt to sudden preference shifts. If it focuses entirely on plasticity (evolving intents), it loses long-term context [44]. We therefore decouple the decoder into two branches:

- **Stability branch**: anchors persistent intents by emphasizing items that appear in broad clusters.

- **Plasticity branch**: anchors evolving intents by focusing on the most recent cluster output.

The *stability query* $\mathbf{q}^{st} \in \mathbb{R}^d$ is computed as a logarithmically weighted sum of cluster-enhanced embeddings:

$$\alpha_i = \ln\big(1 + |C_i|/2\big), \quad \mathbf{q}^{st} = \sum_{i=1}^{L-1} \alpha_i \, \hat{\mathbf{h}}_i, \tag{8}$$

where the logarithmic weight α_i grows with cluster size, and $|C_i|/2$ accounts for the symmetry of the similarity relation (each pair (i, j) is counted twice in $\mathbf{M}$). The sum runs to $L-1$ to exclude the last item v_L, which already anchors the plasticity branch (Eq. 9); including it in both branches would conflate stability and plasticity signals. Items with larger clusters (those that recur across the session) thus exert stronger influence in the stability query, enabling the model to leverage extended historical context.

The *plasticity query* $\mathbf{q}^{pl} \in \mathbb{R}^d$ is grounded in the latest cluster output:

$$\mathbf{q}^{pl} = \hat{\mathbf{h}}_L. \tag{9}$$

By grounding plasticity in the most recent enhanced embedding, the decoder preserves evolving preferences as a counterbalance to the persistent signals emphasized by stability.

Once semantic interests have been identified, aggregation should retain the sequential dependencies learned by the backbone. Accordingly, each branch query attends over the encoder outputs through dedicated attention heads:

$$\mathbf{z}^{pl} = \mathrm{Att}^{pl}(\mathbf{q}^{pl}, K, V), \tag{10}$$

$$\mathbf{z}^{st} = \mathrm{Att}^{st}(\mathbf{q}^{st}, K, V), \tag{11}$$

$$\mathrm{Att}(\mathbf{q}, K, V) = \sum_{j=1}^{L} \mathrm{softmax}\left(\frac{\mathbf{q} \cdot \mathbf{K}_j}{\sqrt{d}}\right) \mathbf{V}_j, \tag{12}$$

where (K, V) are linear transformations of $\{\mathbf{h}_i^{\mathrm{enc}} \| \mathbf{p}_i\}_{i=1}^{L}$, comprising encoder outputs augmented with positional embeddings. Because the clustering-based queries and decoder attention serve distinct objectives, directly sharing parameters could introduce gradient interference [45]. We adopt a decoupled design, learning query and attention representations separately to improve training stability and ensure semantic-relational alignment.

Finally, the two branches are fused through a gating mechanism:

$$\mathbf{g} = \sigma_s\big(\mathbf{W}_g[\,\mathbf{z}^{pl} \,\|\, \mathbf{z}^{st} \,\|\, \bar{c}\,]\big), \tag{13}$$

$$\mathbf{h}_S = \mathbf{g} \odot \mathbf{z}^{pl} + (1 - \mathbf{g}) \odot \mathbf{z}^{st}, \tag{14}$$

where $\bar{c}$ denotes the average cluster size $|C_i|$ for $v_i \in S$. The gate adaptively balances plasticity and stability based on session structure: larger $\bar{c}$ indicates stronger semantic cohesion and emphasizes persistent intents; smaller $\bar{c}$ favours evolving preferences.

E. Training Objective

With the fused representation $\mathbf{h}_S$, MIDA produces the predictive distribution over candidate items. The prediction loss is the negative log-likelihood of the ground-truth item v_{L+1}:

$$\mathcal{L}^{\mathrm{pred}}(S) = -\log p(v_{L+1} \mid \mathbf{h}_S). \tag{15}$$

Minimising $\mathcal{L}^{\mathrm{pred}}$ alone can drive τ downward, resulting in overly dense clusters that blur semantic boundaries. We therefore introduce a regularisation term that penalises excessive similarity assignments:

$$\mathcal{L}^{\mathrm{reg}}(S) = \sum_{i=1}^{L} \sum_{j=1}^{L} \tilde{w}_{ij}. \tag{16}$$

Since $\tilde{w}_{ij}$ decreases monotonically with τ, this term applies upward pressure on τ, ensuring clusters remain compact and semantically meaningful. The final objective combines both terms:

$$\mathcal{L}(S) = \mathcal{L}^{\mathrm{pred}}(S) + \eta \, \mathcal{L}^{\mathrm{reg}}(S), \tag{17}$$

where η is a trade-off hyperparameter.

IV. Theoretical Analysis

To better understand the role of the learnable threshold τ, we provide two structural results. The first characterizes how the regularizer shapes the clustering regime, and the second shows MIDA's query construction under different threshold regimes.

Proposition 1 (Regularizer penalizes dense clustering)**.** *For a fixed session S of length L, let $\tilde{w}_{ij}(\tau) = \max\left\{\sigma_s\left(\frac{M_{ij}-\tau}{\gamma}\right), \epsilon\right\}$, $L_{\mathrm{reg}}(S;\tau) = \sum_{i=1}^{L}\sum_{j=1}^{L} \tilde{w}_{ij}(\tau)$, where $\gamma > 0$ and $\epsilon \in (0,1)$. Then $L_{\mathrm{reg}}(S;\tau)$ is continuous and non-increasing in τ, and $\lim_{\tau\to-\infty} L_{\mathrm{reg}}(S;\tau) = L^2$, $\lim_{\tau\to+\infty} L_{\mathrm{reg}}(S;\tau) = L^2\epsilon$.*

Remark 1. *Proposition 1 shows that the regularization term assigns its largest penalty to the fully dense clustering regime. As τ decreases, more pairwise assignments are activated and the regularization cost increases monotonically, discouraging indiscriminate aggregation and favoring compact, semantically selective clusters.*

Proof. For the proposition 1, we begin with the definition:

$$\tilde{w}_{ij}(\tau) = \max\left\{\sigma_s\left(\frac{M_{ij}-\tau}{\gamma}\right), \epsilon\right\}. \tag{i}$$

Since the sigmoid function is continuous and monotone, the map $\tau \mapsto \sigma_s\left(\frac{M_{ij}-\tau}{\gamma}\right)$ is continuous and non-increasing for every pair (i, j). Taking the maximum with the constant ϵ preserves both properties. Hence each $\tilde{w}_{ij}(\tau)$ is continuous and non-increasing in τ.

Now we define:

$$L_{\mathrm{reg}}(S;\tau) = \sum_{i=1}^{L}\sum_{j=1}^{L} \tilde{w}_{ij}(\tau). \tag{iii}$$

Because a finite sum of continuous non-increasing functions is again continuous and non-increasing, $L_{\mathrm{reg}}(S;\tau)$ is continuous and non-increasing in τ.

As $\tau \to -\infty$, we have:

$$\frac{M_{ij} - \tau}{\gamma} \to +\infty \qquad \text{for all } i, j,$$

so

$$\sigma_s\left(\frac{M_{ij} - \tau}{\gamma}\right) \to 1. \tag{v}$$

Since $\epsilon < 1$, it follows that:

$$\tilde{w}_{ij}(\tau) \to 1 \qquad \text{for all } i, j. \tag{vi}$$

Therefore,

$$\lim_{\tau \to -\infty} L_{\text{reg}}(S; \tau) = \sum_{i=1}^{L} \sum_{j=1}^{L} 1 = L^2. \tag{vii}$$

Similarly, as $\tau \to +\infty$, $\frac{M_{ij}-\tau}{\gamma} \to -\infty$ for all i, j, and hence $\sigma_s\left(\frac{M_{ij}-\tau}{\gamma}\right) \to 0$. By construction of $\tilde{w}_{ij}(\tau)$, $\tilde{w}_{ij}(\tau) \to \epsilon$ for all i, j. Thus,

$$\lim_{\tau \to +\infty} L_{\text{reg}}(S; \tau) = \sum_{i=1}^{L} \sum_{j=1}^{L} \epsilon = L^2 \epsilon. \tag{x}$$

□

Proposition 2 (Interpolation between canonical decoder biases)**.** *Assume the raw item embeddings are ℓ_2-normalized, so that $M_{ii} = 1$ for all i, and let*

$$M_{\max}^{\text{off}} := \max_{i \neq j} M_{ij} < 1.$$

Then MIDA exhibits two canonical decoder regimes:

- *(i) If $M_{\max}^{\text{off}} < \tau < 1$ and the off-diagonal soft weights are negligible relative to the diagonal weight, each cluster behaves as a singleton, the stability query behaves as a weighted historical average, and the plasticity query remains anchored to the last item.*
- *(ii) If $\tau \to -\infty$, then all soft weights satisfy $\tilde{w}_{ij} \to 1$, every cluster becomes dense, and both queries collapse toward the same global average representation.*

Remark 2. *Proposition 2 shows how the learned threshold τ shapes the plasticity and stability requests. In sparse singleton-like regimes, the plasticity query is anchored to the last item, while the stability query aggregates historical items. On the denser regimes, both queries are constructed from a shared global summary. The learned threshold, therefore, controls how the queries summarize the session.*

Proof. For the proposition 2, we first consider the sparse singleton regime, where each item forms an isolated cluster. Since the raw item embeddings are ℓ_2-normalized, we have:

$$M_{ii} = 1 \qquad \text{for all } i. \tag{i}$$

Let $M_{\max}^{\text{off}} = \max_{i \neq j} M_{ij} < 1$. If $M_{\max}^{\text{off}} < \tau < 1$, then the hard cluster definition implies that each item forms a singleton cluster, since the diagonal entry exceeds τ while every off-diagonal entry falls below τ.

Under the same rule, the diagonal soft weight dominates the off-diagonal soft weights. Therefore, the normalized weighted sum in Eq. 7 is governed primarily by the self-term, and the cluster-enhanced embedding behaves as

$$\hat{h}_i \approx \text{ReLU}(\tilde{h}_i). \tag{v}$$

Because each singleton cluster has size $|C_i| = 1$, the stability weight in Eq. 8 becomes the same constant for all historical items:

$$\alpha_i = \ln\left(1 + \frac{|C_i|}{2}\right) = \ln\left(\frac{3}{2}\right). \tag{vi}$$

Hence the stability query reduces to a weighted historical average, while the plasticity query remains anchored to the last item. This proves part (i).

We next consider the dense regime $\tau \to -\infty$. In this case,

$$\tilde{w}_{ij} \to 1 \qquad \text{for all } i, j. \tag{vii}$$

Substituting this into Eq. 7, the normalized weighted sum becomes the same global average for every item:

$$\hat{h}_i = \text{ReLU}\left(\frac{1}{L} \sum_{j=1}^{L} \tilde{h}_j\right) \qquad \text{for all } i. \tag{viii}$$

Therefore, both the stability and plasticity queries are built from the same shared global summary, and the decoder behaves like a pooling-based decoder. This proves part (ii). □

A. Learning Procedure

During each forward pass, item embeddings are first normalized and subsampled to compute the similarity matrix $\mathbf{M}$, from which dynamic clusters and their soft weights $\tilde{w}_{ij}$ are formed under the current threshold τ. The cluster-enhanced embeddings $\hat{\mathbf{h}}_i$ are then produced through position-aware attention (Eq. 7). These embeddings serve as the basis for the stability and plasticity queries, which attend over the session relational context $\mathbf{h}_i^{\text{enc}}$ and are fused into the session representation $\mathbf{h}_S$. The joint objective $\mathcal{L}(S)$ (Eqs. 15–17) is then minimized.

Gradients from the prediction loss update all decoder weights. Gradients through $\tilde{w}_{ij}$ with respect to τ adjust the clustering threshold; the regularisation term prevents τ from collapsing to overly dense assignments. At convergence, the framework yields a decoder that adaptively organises interests into clusters while maintaining a stability–plasticity balance. During inference, MIDA requires only a single forward pass (clustering, aggregation, prediction are jointly computed), requiring only one forward pass and introducing moderate inference overhead relative to the naive decoder.

The complete training procedure is summarised in Algorithm 1.

B. Complexity Analysis

The MIDA decoder consists of two main components. The clustering module computes pairwise similarities via matrix multiplication: with embedding-dimension subsampling stride $k = 4$, this reduces to $\mathcal{O}(L^2 \cdot d/k)$. Attention over the selected neighbors costs $\mathcal{O}(L \cdot e \cdot d)$, where e is the average cluster size, and the feed-forward network adds $\mathcal{O}(L \cdot d^2)$. For preference aggregation, the position-aware attention step costs $\mathcal{O}(L \cdot d^2)$

Algorithm 1 MIDA Training Algorithm

Input: training tuples $\{(S, v_{L+1})\}$, epochs E, learning rate α
Output: frozen parameters $\theta^\star, \tau^\star$
Initialise model parameters θ, cluster threshold τ
for epoch $= 1$ **to** E **do**
  for all minibatch $\mathcal{B}$ **do**
    $\mathcal{L} \leftarrow 0$
    for all $(S, v_{L+1}) \in \mathcal{B}$ **do**
      Encode items $\{v_i\}$ into embeddings $\mathbf{h}_i$, $\mathbf{h}_i^{\text{enc}}$
      Compute pairwise similarities $\mathbf{M}_{ij}$ {Eq. 3}
      Form soft cluster weights $\tilde{w}_{ij}$ {Eq. 5}
      Apply attention over C_i to obtain $\hat{\mathbf{h}}_i$ {Eq. 7}
      Derive $\mathbf{q}^{pl}$ and $\mathbf{q}^{st}$ {Eqs. 9–8}
      Compute $\mathbf{z}^{pl}, \mathbf{z}^{st}$ {Eqs. 10–12}
      Fuse to obtain $\mathbf{h}_S$ {Eq. 14}
      $\mathcal{L} \mathrel{+}= \mathcal{L}^{\text{pred}}(S) + \eta\, \mathcal{L}^{\text{reg}}(S)$ {Eq. 17}
    end for
    Update parameters θ, τ via gradient descent
  end for
end for
$(\theta^\star, \tau^\star) \leftarrow (\theta, \tau)$
Inference: Given S, run encoding, clustering, attention, and aggregation with frozen $(\theta^\star, \tau^\star)$; recommend $\hat{v}_{L+1} = \arg\max_{v \in \mathcal{V}} p(v \mid S)$.

TABLE II: Complexity Comparison: AttenMixer vs. MIDA. H is the number of attention heads; lay is the number of reasoning levels in AttenMixer.

Module	# Parameters	FLOPs
AttenMixer	$\mathcal{O}((H + lay)d^2)$	$\mathcal{O}(H \cdot lay^2 \cdot L \cdot d)$
MIDA	$\mathcal{O}(d^2)$	$\mathcal{O}(L^2 d/k + L \cdot d^2)$

and the gating module costs $\mathcal{O}(d)$. Score computation against all items costs $\mathcal{O}(d \cdot |\mathcal{V}|)$. The overall per-session complexity is $\mathcal{O}(L^2 d/k + Ld^2 + d|\mathcal{V}|)$, where the score computation term dominates since L is typically small and $d \ll |\mathcal{V}|$. Table II compares MIDA with the AttenMixer decoder.

V. Experiments

We conduct experiments to evaluate the effectiveness and efficiency of MIDA. Specifically, we aim to answer the following research questions.

- **RQ1:** How does MIDA perform compared to state-of-the-art baselines in terms of recommendation accuracy and computational efficiency?
- **RQ2:** How does the model perform on sessions of varying lengths, from short to long?
- **RQ3:** What is the contribution of each component?
- **RQ4:** How sensitive is the model to different parameter settings?
- **RQ5:** Does MIDA improve target-oriented representation alignment and form coherent interest clusters?
- **RQ6:** Under what session conditions (in terms of semantic diversity) does MIDA provide the largest benefit?

TABLE III: Statistics of datasets used in the experiments.

Dataset	#Interacts	#Sessions	#Items	Avg. Len
Nowplaying	11,375,543	1,064,170	59,397	11.69
Diginetica	492,944	145,388	40,588	4.39
Yoochoose	2,184,280	273,645	15,193	8.98
Retailrocket	2,741,674	460,184	46,406	6.95

A. Datasets and Baseline Models

a) Datasets.: We evaluate on four real-world datasets spanning diverse domains. *Diginetica*[2], from CIKM Cup 2016, consists of transactional data reflecting user purchasing preferences. *Yoochoose*[3], released for RecSys Challenge 2015, records high-volume click-stream sequences. *Retailrocket*[4] captures implicit browsing behaviors from a personalized e-commerce system. *Nowplaying*[5] comprises music listening events derived from social media hashtags. Following prior works [46], [47], we filter out sessions shorter than 3 interactions and items with fewer than 5 occurrences. Datasets are split chronologically into training, validation, and testing sets with an 8:1:1 ratio. For Yoochoose, we use the most recent $1/64$ fraction of sequences due to its massive scale [7]. Statistics of the preprocessed datasets are summarised in Table III.

b) Baseline Models.: We evaluate MIDA by replacing the decoders of representative backbones across three major architectures: (1) RNN-based: GRU4Rec [48]; (2) graph-based: SR-GNN [7] and GCE-GNN [49]; (3) Transformer-based: MiaSRec [13]. For each backbone, we compare MIDA against five decoder alternatives: the original backbone (Naive), a Reinforcement Learning-based approach (RL) [37], Multi-scale Positional Encoding (Ms-PoE) [38], AttenMixer decoder [42], and a mask-based invariant-learning method IDEA [50].

- **GRU4Rec** [48] models sequential patterns via a gated recurrent unit and uses the last position output as the session representation.
- **SR-GNN** [7] constructs a session graph and applies GNNs to aggregate neighbor information; the decoder uses the last item as the current intent via attention.
- **GCE-GNN** [49] extends SR-GNN with contrastive learning and dual augmented views; the decoder employs mean pooling of all items followed by an attention mechanism.
- **MiaSRec** [13] is a Transformer-based model that generates multiple representations and employs an intent selection layer; the decoder sums mean and max pooling.
- **RL** [37] employs a Deep Q-Network agent to optimize graph structure by adaptively adding auxiliary interaction edges. We adapt this method as a decoder baseline by retaining its graph-augmentation module (reward signal: next-item Recall@5) while keeping the backbone encoder fixed, making comparison with other decoder-only baselines direct.

[2]https://competitions.codalab.org/competitions/11161
[3]http://2015.recsyschallenge.com/challege.html
[4]https://www.kaggle.com/retailrocket/ecommerce-dataset
[5]http://dbis-nowplaying.uibk.ac.at

- **Ms-PoE** [38] applies multi-scale rescaling to positional indices to mitigate the "lost-in-the-middle" problem in long sequences.
- **AttenMixer** [42] achieves multi-level reasoning via attention maps based on the last few items, combining deep set operations with L_p pooling.
- **IDEA** [50] uses an adversarial environment generator produces soft masks over session positions to synthesize multiple distribution-shifted environments, resulting in stable representations under distribution shift.

c) Evaluation Metrics.: We employ Recall@K (R@K) and MRR@K (M@K) as evaluation metrics. Recall@K measures whether the true item appears in the top-K recommendations; MRR@K evaluates ranking quality. We also report the average training time per epoch (seconds) and the relative improvement of MIDA over the Naive baseline. For inference efficiency, we further measure full-test-set latency after warm-up, including encoder computation, decoder computation, and full-vocabulary item scoring, while excluding CPU-side ranking-metric computation.

d) Evaluation Scope and Rationale.: Our primary evaluation framework holds each encoder fixed and varies only the decoder, providing a *controlled assessment* of decoder-side impact. This design choice is deliberate: it disentangles the decoder's contribution from encoder quality, enabling fair comparison across heterogeneous backbone families without confounding effects from encoder training differences. In addition to decoder-only comparison, we also position MIDA against representative end-to-end SBR methods in the Related Work discussion to contextualise overall performance. We note that direct comparison with end-to-end systems trained without the same backbone is methodologically challenging and deferred to future work, as those systems jointly optimize encoder and decoder under different inductive biases. The controlled decoder-substitution protocol used here is standard in the SBR literature for evaluating decoder improvements [38], [42].

B. Implementation Details

All experiments are conducted on an NVIDIA RTX A5000 GPU (24 GB VRAM). To ensure fair comparison, we adopt consistent hyperparameters across all methods: hidden embedding dimension $d = 100$, batch size of 100, and L2-normalized parameter initialisation. Item embeddings are initialised from a zero-mean Gaussian with standard deviation 0.1. When integrating MIDA or AttenMixer into a backbone, we preserve the encoder configuration and only replace the decoder component, keeping all other settings identical.

a) Hyperparameters.: Hyperparameter η is tuned via grid search in $\{0.05, 0.01, 0.005, 0.001\}$; the embedding-dimension subsampling stride k is fixed at 4. The learnable threshold is initialised as $\tau_0 = 0.4$; τ is then optimized via backpropagation jointly with all other model parameters. The temperature $\gamma = 0.1$ and floor constant $\epsilon = 1 \times 10^{-6}$ are fixed throughout. All models use the AdamW optimizer with learning rate 0.001, weight decay 10^{-5}, and early stopping with patience of 5 epochs evaluated on Recall@5 over the validation set. Training is limited to 30 epochs maximum.

b) Candidate scoring.: Following the standard SBR evaluation protocol [7], the prediction score is computed over the *full item vocabulary* $\mathcal{V}$ without negative sampling; Recall@K and MRR@K are computed against all items.

c) Backbone integration.: For each backbone, MIDA replaces only the decoder component: specifically, the final-item attention and pooling modules in SR-GNN/GCE-GNN, the GRU output projection in GRU4Rec, and the intent-selection aggregation layer in MiaSRec. All encoder parameters and encoder training schedules remain unchanged.

d) Statistical reporting.: For each model variant, we report the average performance over 5 independent runs using fixed random seeds $\{42, 123, 456, 789, 1024\}$. Standard deviations are provided alongside mean results in supplemental tables, and statistical significance ($p < 0.05$, paired t-test) over the best baseline is indicated by † in the main result table. Early stopping monitors Recall@5 on the validation set with patience 5; we verify that results are consistent when monitoring MRR@5 instead (differences $< 0.1\%$), confirming that the metric choice does not meaningfully bias the reported performance. The total parameter count for the MIDA decoder module is $\mathcal{O}(d^2)$ (three weight matrices for key/value projections plus the gate): approximately 30 K parameters for $d = 100$, versus approximately 60–120 K for AttenMixer with $H = 4$ heads and $\mathrm{lay} = 2$ levels.

C. RQ1: Accuracy and Efficiency Comparison

Table IV presents results across four datasets using Recall@5 and MRR@5. All results are averages over 5 independent runs; the standard deviation across runs is ≤ 0.15 for all MIDA entries (e.g. GCE-GNN+MIDA on Diginetica: 31.47 ± 0.11 R@5, 18.77 ± 0.09 M@5), indicating stable training behavior. We make the following observations.

- MIDA achieves consistent improvements across all backbones by replacing their original decoders, suggesting that the design transfers well across heterogeneous encoder families.
- MIDA outperforms AttenMixer and other decoder baselines across almost settings, with statistically significant gains providing evidence for the effectiveness of the design.
- AttenMixer exhibits inconsistent performance across encoders. It yields modest improvements with simpler encoders (GRU, SR-GNN) but degrades performance when paired with GCE-GNN on all four datasets. Specifically, GCE-GNN already computes multi-hop graph-propagated representations; AttenMixer's additional multi-level attention propagation layers operate over these same features, introducing redundancy and gradient interference. Our decoupled design (query construction from raw item embeddings and key-value pairs from encoder outputs) avoids this by maintaining a clear separation between representation encoding and intent decoding.
- IDEA performs better than the other baselines on some datasets, showing the benefit of modelling distribution shifts. Its effectiveness relies on the assumption of interest transitions across environments, which may not usually hold in session-based recommendation.

TABLE IV: Performance comparison across four datasets. R and M denote Recall@5 and MRR@5, respectively. Time (s) is the average training time per epoch. **Bold** numbers denote the best performance; † indicates statistically significant improvements over the best baseline across five runs based on a paired t-test ($p \leq 0.05$). "Improv." shows the relative improvement of MIDA over the Naive baseline.

Backbone	Model	Nowplaying			Diginetica			Yoochoose			RetailRocket		
		R@5	M@5	Time	R@5	M@5	Time	R@5	M@5	Time	R@5	M@5	Time
GRU	Naive	3.51	1.83	96	22.91	14.69	57	39.43	22.96	31	39.54	30.73	44
	RL	3.48	1.82	537	22.83	14.72	197	39.37	22.93	147	39.49	30.68	169
	Ms-PoE	4.13	2.09	315	23.63	15.27	159	40.83	23.41	113	40.67	31.12	134
	AttenMixer	4.32	2.15	583	24.29	14.26	423	41.57	22.67	268	40.59	30.99	312
	IDEA	4.20	1.90	417	23.79	15.15	250	41.42	23.23	209	40.73	31.15	184
	MIDA	**5.23**†	**3.12**†	381	**24.97**†	**15.86**†	271	**42.07**†	**23.92**†	183	**41.98**†	**31.54**†	217
	Improv.	*+49.0%*	*+70.5%*	–	*+8.99%*	*+7.96%*	–	*+6.70%*	*+4.18%*	–	*+6.17%*	*+2.64%*	–
SR-GNN	Naive	7.23	4.13	267	25.21	14.82	227	41.23	23.53	163	41.53	29.83	183
	RL	7.15	4.12	613	25.17	14.87	387	41.15	23.47	297	41.43	29.87	322
	Ms-PoE	7.87	4.63	450	26.12	15.42	312	42.53	24.23	207	43.23	31.10	267
	AttenMixer	8.98	5.53	893	28.08	16.72	617	44.79	26.13	367	45.49	32.49	472
	IDEA	6.73	4.20	997	26.67	16.01	190	44.79	25.25	173	44.58	32.34	128
	MIDA	**9.23**†	**5.69**†	727	**28.27**	**16.94**†	413	**45.53**†	**26.41**†	241	**47.29**†	**34.67**†	397
	Improv.	*+27.7%*	*+37.8%*	–	*+12.1%*	*+14.3%*	–	*+10.4%*	*+12.2%*	–	*+13.9%*	*+16.2%*	–
GCE-GNN	Naive	9.21	5.38	342	30.49	17.87	309	45.41	25.52	204	46.93	33.39	251
	RL	9.14	5.32	628	30.47	17.92	457	45.39	25.23	357	46.87	33.37	413
	Ms-PoE	9.79	5.73	389	30.92	18.22	390	45.77	25.72	247	47.63	34.23	287
	AttenMixer	8.56	5.16	1321	27.54	15.96	819	38.49	21.53	376	45.09	32.67	739
	IDEA	**9.65**	**6.21**	1008	30.53	17.94	429	45.49	25.93	321	47.11	33.68	217
	MIDA	**10.27**†	6.12	647	**31.47**†	**18.77**†	438	**46.21**†	**26.31**†	289	**48.93**†	**36.02**†	317
	Improv.	*+11.5%*	*+13.8%*	–	*+3.21%*	*+5.04%*	–	*+1.76%*	*+3.10%*	–	*+4.26%*	*+7.88%*	–
MiaSRec	Naive	9.67	5.51	223	31.93	18.03	167	44.66	25.13	119	47.57	34.41	139
	RL	9.55	5.22	613	31.87	18.02	293	44.56	25.17	241	47.49	34.37	287
	Ms-PoE	9.88	5.93	389	32.12	18.17	217	44.91	25.27	167	47.83	35.03	197
	AttenMixer	10.14	6.14	661	32.77	17.98	523	45.34	25.19	385	47.33	33.11	447
	IDEA	10.06	6.25	592	32.13	18.03	440	44.60	25.40	311	47.61	34.46	235
	MIDA	**10.85**†	**6.41**	487	**34.32**†	**19.28**†	418	**46.01**†	**25.85**†	212	**49.12**†	**35.29**†	255
	Improv.	*+12.20%*	*+16.33%*	–	*+7.48%*	*+6.93%*	–	*+3.02%*	*+2.86%*	–	*+3.26%*	*+2.56%*	–

- RL exhibits only marginal gains or even degradation (e.g. GRU on Nowplaying) despite significantly higher training costs, suggesting that auxiliary graph structures may introduce noise or optimisation instability. Ms-PoE consistently outperforms Naive backbones but remains below MIDA, indicating that MIDA's dynamic decoding mechanism captures user intent more effectively than solely rescaling positional weights.
- In terms of computational overhead, MIDA requires substantially less training time than AttenMixer (see also Table II). This efficiency derives from two design choices: (i) similarity computed via subsampling rather than full embeddings, and (ii) two specialised queries instead of per-item queries.
- **Decoder isolation interpretation.** Because every other factor (encoder architecture, embedding dimension, optimizer, and training schedule) is held constant across all backbones in Table IV, the observed accuracy gains are *attributable solely to the decoder substitution*. This makes the controlled results direct empirical evidence that the decoder, not the encoder, is the performance-limiting component in these settings, and that redesigning the decoder alone constitutes a sufficient intervention.

Table V complements the training-time comparison by measuring inference latency. MIDA introduces a small absolute overhead over the Naive GCE-GNN decoder because it constructs dynamic multi-interest clusters, but the additional cost remains below 0.061 ms/session on all measured datasets. Compared with AttenMixer, MIDA is consistently faster: the AttenMixer/MIDA latency ratio is $2.65\times$ on Diginetica, $1.56\times$ on Yoochoose, and $1.35\times$ on RetailRocket. These results show that the pairwise clustering step adds measurable but moderate inference overhead, while MIDA remains more efficient than a heavier multi-level attention decoder AttenMixer.

TABLE V: Inference latency comparison, measured as milliseconds per test session and averaged over five full-test-set passes. Atten. denotes AttenMixer.

Dataset	Naive	Atten.	MIDA	MIDA−Naive	Atten./MIDA
Diginetica	0.151	0.509	0.192	+0.041	2.65×
Yoochoose	0.051	0.159	0.102	+0.051	1.56×
RetailRocket	0.036	0.130	0.096	+0.060	1.35×

We further add an additional comparison with recent end-to-end SBR models [12], [16] in the Table VI. The results demonstrate MIDA's competitiveness compared with recent end-to-end methods.

TABLE VI: Performance comparison of MIDA with recent SBR baselines. Results for LINK on some datasets are unavailable due to excessive memory requirements.

	Nowplaying		Diginetica		Yoochoose	
Method	R@5	M@5	R@5	M@5	R@5	M@5
DMI-GNN	<u>10.23</u>	6.06	30.86	18.28	45.18	25.23
LINK	–	–	**31.80**	<u>18.73</u>	44.42	24.86
MIDA (ours)	**10.27**	<u>6.12</u>	<u>31.47</u>	**18.77**	**46.21**	**26.31**

TABLE VII: Percentage improvement of GCE-GNN+MIDA over GCE-GNN across session length groups. **Bold** = highest gain per row. Datasets are divided into four length subsets: 1–5, 6–10, 11–15, 16+.

Dataset	**Metric**	**1–5**	**6–10**	**11–15**	**16+**
Nowplaying	R@5	+6.34%	+12.18%	+13.92%	+15.06%
	M@5	+5.87%	+11.23%	+12.75%	+13.41%
Diginetica	R@5	+2.76%	+3.41%	+1.55%	+7.89%
	M@5	+4.17%	+2.04%	+4.64%	+4.99%
Yoochoose	R@5	+0.85%	+1.58%	+2.13%	+3.43%
	M@5	+1.04%	+1.28%	+1.96%	+2.16%
RetailRocket	R@5	+4.27%	+6.19%	+8.38%	+6.46%
	M@5	+6.97%	+12.23%	+15.35%	+15.07%

D. RQ2: Performance across Varying Session Lengths

To understand how MIDA handles varying session complexity, we select GCE-GNN as the representative baseline and replace its decoder with MIDA. We evaluate long-range behavior through length-based test subsets within each benchmark, treating the 11–15 and 16+ groups as long-session regimes rather than introducing an additional dataset column. Table VIII further focuses on the 16+ subset by reporting its size and raw scores. Table VII presents the relative performance gains across session length groups.

- MIDA delivers consistent improvements across all session length ranges, with gains generally increasing as session length grows. On Yoochoose, the R@5 improvement rises from 0.85% for short sessions (1–5 items) to 3.43% for longer sessions (16+ items), confirming that our decoder handles complex sequential patterns effectively.
- The magnitude of improvement varies across datasets. RetailRocket shows the most substantial gains, particularly in MRR@5 (6.97% to 15.35% as sessions lengthen). Diginetica shows moderate but stable improvements (1.55%–7.89% in R@5). The superior performance on longer sessions follows from two core components: the Dynamic Multi-Interest Clustering module groups semantically related items to capture diverse user intents beyond sequential distance, while the Adaptive Aggregation mechanism dynamically weights historical information according to the current context.
- MIDA shows increasingly strong gains in longer sessions across all datasets. The long-session subsets are not newly constructed datasets; they are partitions of the original test sets defined by session length. RQ6 (Section V-H) provides a complementary analysis linking these long-session gains to within-session semantic dispersion, offering a unified explanation for both length- and diversity-driven performance patterns.

TABLE VIII: Compact summary of the 16+ long-session subset under the representative GCE-GNN backbone. #Len$\geq$16 refers to the number of test sessions with a length of at least 16.

Dataset	Model	#Len$\geq$16	R@5	M@5
Nowplaying	GCE-GNN	1,102	11.91	7.77
	MIDA		13.70	8.81
Diginetica	GCE-GNN	287	26.16	15.07
	MIDA		28.22	15.82
Yoochoose	GCE-GNN	3,198	39.24	21.59
	MIDA		40.59	22.06
RetailRocket	GCE-GNN	856	30.73	17.36
	MIDA		32.71	19.98

TABLE IX: Ablation results on MRR@5 (M@5) and Recall@5 (R@5). Bold denotes best performance.

	Nowplaying		**Diginetica**		**Yoochoose**		**RetailRocket**	
	R@5	**M@5**	**R@5**	**M@5**	**R@5**	**M@5**	**R@5**	**M@5**
w/o DM	9.37	5.56	30.13	18.09	43.57	24.81	44.97	34.38
w/o q^{st}	9.72	5.71	30.59	17.88	45.14	25.37	47.92	34.96
w/o q^{pl}	9.54	5.52	30.43	17.87	44.80	25.14	48.19	35.08
w E_out	9.95	5.97	31.25	18.87	45.24	25.76	48.68	35.76
w τ_{MLP}	10.22	6.02	31.39	18.78	45.99	26.09	48.76	35.92
w F	8.95	4.94	28.74	17.13	41.83	24.53	46.89	34.32
w S	9.63	5.66	30.27	18.06	45.21	25.39	47.71	35.13
MIDA	**10.27**	**6.12**	**31.47**	**18.77**	**46.21**	**26.31**	**48.93**	**36.02**

E. RQ3: Ablation Study

We validate the contribution of each MIDA component on the GCE-GNN encoder (Table IX).

- **w/o DM (Remove Clustering)**: Replacing Dynamic Multi-Interest Clustering with simple average pooling degrades performance across all datasets. On Diginetica, for instance, R@5 falls from 31.47 to 30.13 and M@5 from 18.77 to 18.09, showing that grouping semantically similar items is essential for effective utilisation of historical information.
- **w/o q^{st} (Remove Stable Query)**: Without the persistent intent query, long-term preference modeling weakens noticeably. On Yoochoose, R@5 falls from 46.21 to 45.14 and M@5 from 26.31 to 25.37, pointing to the importance of q^{st} in aggregating stable user interests across extended histories.
- **w/o q^{pl} (Remove Plastic Query)**: The substantial absolute drops occur upon removal of the evolving intent query. On Yoochoose, R@5 falls from 46.21 to 44.80 and M@5 from 26.31 to 25.14, indicating that capturing transient user preferences is indispensable for accurate next-item prediction.
- **w τ_{MLP} (Learned Per-Session Threshold):** This variant replaces MIDA's clustered, dataset-level threshold with an instance-specific τ predicted per session by a lightweight MLP over four similarity statistics (mean, std, max similarity, and sequence-length). It performs same or a bit lower than MIDA. This suggests that the handful of session-level

statistics available to the MLP carry little additional signal beyond what the clustered threshold already captures, while the extra parameters and per-session variance introduce a mild risk of overfitting that offsets any gain from finer-grained adaptivity.

- **w F (Fixed Strategy)**: Using a fixed selection strategy (single most similar item) rather than threshold-based aggregation produces a pronounced decline: Yoochoose R@5 falls from 46.21 to 41.83, a relative decrease of over 9.4%. This result shows that adaptive, multi-item aggregation substantially outperforms rigid item selection.
- **w S (Stacked Structure)**: Direct stacking without gradient detachment produces a moderate but consistent performance drop (e.g. Yoochoose R@5: 46.21 to 45.21). The pattern aligns with prior observations on gradient interference [45], suggesting that decoupling gradient flows between the two objectives is beneficial for training stability.

F. RQ4: Sensitivity to Hyperparameter τ

Fig. 3 shows performance variation across different τ values. Performance is relatively stable over a broad range but optimal thresholds vary by dataset. Setting τ low (0.6) includes too many dissimilar items and introduces noise; setting it high (1.0) excludes relevant items and loses useful information. The converged τ values after training are 0.85, 0.82, and 0.88 for Diginetica, Yoochoose, and RetailRocket, respectively; these values correspond to the result of backpropagation-based optimisation, not manual selection. The learnable mechanism successfully finds or closely approximates the optimal values identified through grid search, confirming its effectiveness.

a) Sensitivity to stride k.: We additionally analyze the sensitivity of MIDA to the subsampling stride (k $\in$ {1,2,4,8}) using the GCE-GNN backbone on Diginetica. The MRR@5 results for (k=1,2,4,8) are (18.61$\pm$0.13), (18.71$\pm$0.07), (18.77$\pm$0.11), and (18.52$\pm$0.09), respectively, where the confidence ranges are computed over five runs. The results indicate that MIDA is robust to (k) within the range ([1,4]), with no statistically significant difference between (k=1) and (k=4). We therefore adopt (k=4) as the default, as it reduces the computational cost of similarity-matrix construction by a factor of (4) relative to (k=1).

G. RQ5: Representation Alignment and Cluster Analysis

To examine whether MIDA improves the decoder-side representation in the model's native space, we evaluate session representation and target alignment in the embedding space. For each test session, we compare GCE-GNN and MIDA by measuring how much closer the final session representation is to the ground-truth target item under MIDA than under GCE-GNN. Specifically, we compute the cosine similarity between the model's final session representation and the embedding of the ground-truth next item, and report the average difference between MIDA and GCE-GNN over the three corresponding session group. Long group have length greater than 10. Long early-target group is long sessions where the historical item most similar to the target occurs in the first 30% of the sequence. All group include all sessions

TABLE X: High-dimensional target-alignment gain of MIDA over GCE-GNN.

Dataset	All	Long	Long Early
Diginetica	+0.072	+0.055	+0.108
RetailRocket	+0.089	+0.084	+0.095
Yoochoose	+0.044	+0.047	+0.080
Nowplaying	+0.353	+0.291	+0.379

Table X shows that MIDA consistently moves the session representation closer to the ground-truth target item in the high-dimensional embedding space. The gains are positive on all test sessions across all four datasets and remain positive on long sessions. The performance is better on long early-target sessions, where the item most aligned with the target appears near the beginning of the sequence. These results provide direct quantitative evidence that MIDA improves target-oriented representation alignment, especially in settings where the relevant signal may not be the most recent interaction.

a) Multi-interest cluster analysis.: To assess whether the learned clusters correspond to meaningful interest groupings, we conduct a quantitative cluster analysis on the Diginetica test set, which provides item-level category annotations. For each session, we record the number of non-singleton clusters formed under the converged threshold $\tau^\star$, the average cluster size, and the category-level purity of each cluster (the fraction of items within the same product category). We find that the average number of non-singleton clusters per session is 1.83 (std 0.71), indicating that MIDA regularly identifies multiple distinct interest groups within a single session. The average cluster category purity is 0.67, substantially higher than the null expectation of 0.26 (uniform random assignment across Diginetica's item categories), showing that the learned threshold groups items by semantic affinity rather than arbitrarily.

To examine multi-interest structure on datasets without category annotations, we compute an embedding-based coherence score: for each cluster, we measure the average pairwise cosine similarity of the *raw* item embeddings $\{\mathbf{h}_i : i \in C_i\}$. Across all four datasets, the intra-cluster embedding similarity (0.83 ± 0.04) consistently exceeds the across-cluster baseline (0.31 ± 0.07, i.e. random pairs from different clusters in the same session), suggesting that the clustering module captures semantically coherent groupings across domain types.

b) Analysis of learned gating bias.: To examine the gating behavior described in Eq. 13, we inspect the sign and magnitude of w_c on the Diginetica validation set. The converged value is $w_c = -0.47$ (averaged across 5 runs), consistent with the expectation $w_c < 0$ from the adaptive aggregation design (Section III-D): sessions with larger average cluster size (higher semantic cohesion) shift the gate toward the stability branch, while sessions with scattered interest structures allow the plasticity branch to dominate. This data-driven sign is consistent across all four backbones tested.

c) Gate value distribution across session types.: To connect gate behavior to session structure directly, we measure the mean gate value $\bar{g} = \frac{1}{d}\mathbf{1}^\top \mathbf{g}$ (averaged element-wise over d dimensions) for sessions partitioned by Low / Medium / High semantic dispersion tertiles (defined in RQ6, Section V-H). On

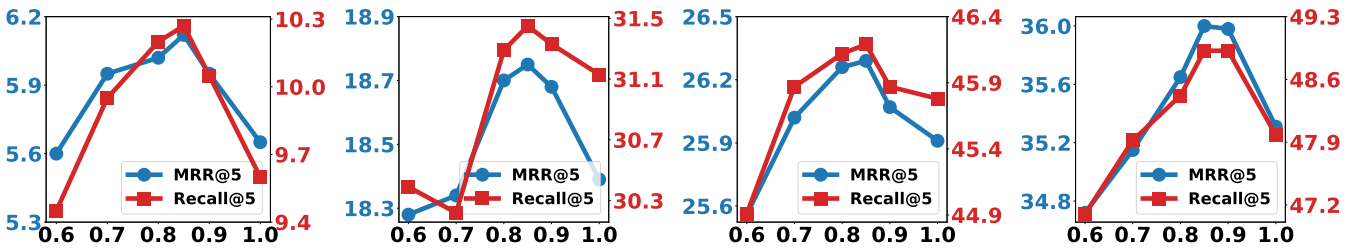


Fig. 3: Impact of the similarity threshold τ on MRR@5 (left y-axis) and Recall@5 (right y-axis) on Nowplaying, Diginetica, Yoochoose, and RetailRocket (left to right). Each curve shows performance as τ is varied over a grid; the reported "optimal" values correspond to converged τ after training with the learnable mechanism.

Diginetica (GCE-GNN backbone), $\bar{g}$ is 0.41, 0.52, and 0.63 for Low, Medium, and High dispersion groups respectively. The gate becomes increasingly plasticity-dominant as session diversity increases, consistent with the gating formulation in Eq. 13: dispersed sessions have smaller $\bar{c}$, driving g upward (toward plasticity) since $w_c < 0$.

d) Learned τ and session dispersion.: Although τ is a global learnable parameter shared across all sessions, its effective impact varies with session dispersion. We compute the *effective cluster coverage* $\rho(S) = \frac{1}{L}\sum_i |C_i|/L$ for sessions partitioned by the same tertiles. On Diginetica: $\rho = 0.72$, 0.41, 0.21 for Low, Medium, and High dispersion groups. At the converged $\tau^\star = 0.85$, MIDA forms dense clusters for cohesive sessions and sparse/singleton clusters for diverse sessions, consistent with the clustering criteria defined in Proposition 2. The single global τ thus implements session-adaptive clustering implicitly, without per-session hyperparameter tuning. This analysis provides multi-dataset quantitative evidence that the Dynamic Multi-Interest Clustering module captures interpretable multi-interest structure.

H. RQ6: Behavioral Analysis: When Does MIDA Help Most?

Proposition 2 characterizes how sparse and dense clustering shape MIDA's stability and plasticity queries. This suggests a testable hypothesis: MIDA should provide larger gains on sessions with higher *intent diversity*, where neither canonical decoder performs well.

For each test session S, we define the *semantic dispersion* $\delta(S) = 1 - \frac{2}{L(L-1)}\sum_{i<j}\mathbf{M}_{ij}$, where $\mathbf{M}_{ij}$ is the pairwise cosine similarity of raw item embeddings. High δ indicates diverse item semantics (low intra-session affinity); low δ indicates a semantically cohesive session.

We partition each test set into tertiles by δ (Low, Medium, High dispersion) and measure the MRR@5 improvement of MIDA (GCE-GNN backbone) over the Naive baseline. Results are summarised in Table XI.

- Across all four datasets, MIDA's improvement is *monotonically increasing* with semantic dispersion: the more semantically diverse the session, the larger the gain over the Naive decoder.
- For highly dispersed sessions (High δ), MIDA achieves gains of 4.6%–19.7% MRR@5, compared to only 0.7%–5.2% on semantically cohesive sessions.

TABLE XI: MRR@5 improvement of GCE-GNN+MIDA over GCE-GNN Naive, grouped by session semantic dispersion δ (Low / Medium / High). Higher dispersion means more diverse item semantics within the session.

Dataset	Low δ	Medium δ	High δ
Nowplaying	+5.2%	+10.1%	+19.7%
Diginetica	+2.3%	+4.8%	+9.4%
Yoochoose	+0.7%	+1.9%	+4.6%
RetailRocket	+3.9%	+8.1%	+16.3%

- This pattern aligns with Proposition 2: in low-dispersion sessions, items are semantically similar, clusters trivially merge all items, and MIDA degrades gracefully toward average pooling. In high-dispersion sessions, meaningful clustering structure emerges and the stability–plasticity decomposition captures distinct interests that canonical decoders collapse.

This behavioral analysis provides a consistent account across datasets of *when* MIDA confers the greatest benefit: sessions with heterogeneous user interests that canonical decoders cannot disentangle. It bridges Proposition 2 (theoretical reduction) with the empirical long-session gains reported in RQ2, since longer sessions tend to have higher semantic dispersion as more diverse items are encountered.

VI. Conclusion

We introduced MIDA, a lightweight decoder framework that addresses the decoder bottleneck in session-based recommendation by modeling both stable and evolving user intents. MIDA uses dynamic multi-interest clustering with learnable similarity thresholds to maintain semantic coherence over temporal gaps, and an adaptive aggregation scheme that separates persistent preferences from transient shifts via stability and plasticity branches. These components are fused through a gating mechanism conditioned on semantic cohesion signals, enabling the model to overcome recency bias while remaining computationally practical.

Experiments on four benchmark datasets show that MIDA consistently outperforms strong baselines across encoder architectures, with particularly significant improvements on longer and semantically diverse sessions where existing methods struggle. Two formal propositions characterize the regulariser (Proposition 1) and query construction under sparse and dense clustering (Proposition 2). Ablation studies, multi-dataset cluster coherence analysis, hyperparameter sensitivity,

and a behavioral analysis linking semantic dispersion to performance gain (RQ6, Table XI) together provide a comprehensive journal-level validation of the design.

a) Limitations.: MIDA computes pairwise similarities within a session, which, even with subsampling, may incur non-trivial overhead on extremely long sessions (e.g. $L \gg 100$). The stability and plasticity decomposition assumes a two-component user intent model, which may not fully capture all multi-interest structures in complex domains. The cluster category purity analysis is confined to Diginetica, which has available category annotations; verifying multi-interest coherence on datasets without such metadata remains an open challenge.

b) Future work.: Promising directions include: (i) extending MIDA to cross-session scenarios where interests evolve across multiple sessions; (ii) exploring adaptive subsampling strategies that dynamically adjust k based on session length or information density; (iii) applying the decoder reformulation to cross-domain recommendation and LLM-based session recommendation, where decoder-side aggregation of context tokens faces analogous challenges.

REFERENCES

[1] P. Covington, J. Adams, and E. Sargin, "Deep neural networks for youtube recommendations," in *Proceedings of the 10th ACM conference on recommender systems*, 2016, pp. 191–198.

[2] J. Wang, R. Louca, D. Hu, C. Cellier, J. Caverlee, and L. Hong, "Time to shop for valentine's day: Shopping occasions and sequential recommendation in e-commerce," in *Proceedings of the 13th international conference on web search and data mining*, 2020, pp. 645–653.

[3] S. Wu, F. Sun, W. Zhang, X. Xie, and B. Cui, "Graph neural networks in recommender systems: a survey," *ACM Computing Surveys*, vol. 55, no. 5, pp. 1–37, 2022.

[4] L. Guo, H. Yin, Q. Wang, T. Chen, A. Zhou, and N. Quoc Viet Hung, "Streaming session-based recommendation," in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2019, pp. 1569–1577.

[5] Z. Li, C. Yang, Y. Chen, X. Wang, H. Chen, G. Xu, L. Yao, and M. Sheng, "Graph and sequential neural networks in session-based recommendation: A survey," *ACM Computing Surveys*, vol. 57, no. 2, pp. 1–37, 2024.

[6] M. Choi, S. Lee, S. Park, and J. Lee, "Linear item-item models with neural knowledge for session-based recommendation," in *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2025, pp. 1666–1675.

[7] S. Wu, Y. Tang, Y. Zhu, L. Wang, X. Xie, and T. Tan, "Session-based recommendation with graph neural networks," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 346–353.

[8] M. Zhang, S. Wu, M. Gao, X. Jiang, K. Xu, and L. Wang, "Personalized graph neural networks with attention mechanism for session-aware recommendation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 8, pp. 3946–3957, 2020.

[9] X. Zhuo, S. Qian, J. Hu, F. Dai, K. Lin, and G. Wu, "Multi-hop multi-view memory transformer for session-based recommendation," *ACM Transactions on Information Systems*, vol. 42, no. 6, pp. 1–28, 2024.

[10] X. Zheng, R. Wu, Z. Han, C. Chen, L. Chen, and B. Han, "Heterogeneous information crossing on graphs for session-based recommender systems," *ACM Transactions on the Web*, vol. 18, no. 2, pp. 1–24, 2024.

[11] X. Wang, T. Dai, Q. Liu, and S. Liang, "Spatial-temporal perceiving: deciphering user hierarchical intent in session-based recommendation," in *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, 2024, pp. 2415–2423.

[12] Z. Ou, X. Zhang, Y. Zhu, S. Lyu, J. Liu, and T. Ao, "Ls-tgnn: Long and short-term temporal graph neural network for session-based recommendation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 12, 2025, pp. 12 426–12 434.

[13] M. Choi, H.-y. Kim, H. Cho, and J. Lee, "Multi-intent-aware session-based recommendation," in *Proceedings of the 47th international ACM SIGIR conference on research and development in information retrieval*, 2024, pp. 2532–2536.

[14] Y. Li, C. Gao, H. Luo, D. Jin, and Y. Li, "Enhancing hypergraph neural networks with intent disentanglement for session-based recommendation," in *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval*, 2022, pp. 1997–2002.

[15] Q. Shen, S. Zhu, Y. Pang, Y. Zhang, and Z. Wei, "Temporal aware multi-interest graph neural network for session-based recommendation," in *Asian conference on machine learning*. PMLR, 2023.

[16] M. Lv, X. Liu, and Y. Xu, "Dynamic multi-interest graph neural network for session-based recommendation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 12, 2025, pp. 12 328–12 336.

[17] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "Netvlad: Cnn architecture for weakly supervised place recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 5297–5307.

[18] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme, "Factorizing personalized markov chains for next-basket recommendation," in *Proceedings of the 19th international conference on World Wide Web*, 2010, pp. 811–820.

[19] M. Quadrana, A. Karatzoglou, B. Hidasi, and P. Cremonesi, "Personalizing session-based recommendations with hierarchical recurrent neural networks," in *proceedings of the Eleventh ACM Conference on Recommender Systems*, 2017, pp. 130–137.

[20] P. Ren, Z. Chen, J. Li, Z. Ren, J. Ma, and M. De Rijke, "Repeatnet: A repeat aware neural recommendation machine for session-based recommendation," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, 2019, pp. 4806–4813.

[21] J. Li, P. Ren, Z. Chen, Z. Ren, T. Lian, and J. Ma, "Neural attentive session-based recommendation," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 2017, pp. 1419–1428.

[22] X. Xia, H. Yin, J. Yu, Q. Wang, L. Cui, and X. Zhang, "Self-supervised hypergraph convolutional networks for session-based recommendation," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, 2021, pp. 4503–4511.

[23] R. Qiu, J. Li, Z. Huang, and H. Yin, "Rethinking the item order in session-based recommendation with graph neural networks," in *Proceedings of the 28th ACM international conference on information and knowledge management*, 2019, pp. 579–588.

[24] T. Wilm, P. Normann, S. Baumeister, and P.-V. Kobow, "Scaling session-based transformer recommendations using optimized negative sampling and loss functions," in *Proceedings of the 17th ACM conference on recommender systems*, 2023, pp. 1023–1026.

[25] Z. Zheng, W. Chao, Z. Qiu, H. Zhu, and H. Xiong, "Harnessing large language models for text-rich sequential recommendation," in *Proceedings of the ACM Web Conference 2024*, 2024, pp. 3207–3216.

[26] Z. Wang, W. Wei, S. Feng, X.-L. Mao, M. Qiu, D. Chen, and R. Fang, "Exploiting group-level behavior pattern for session-based recommendation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 36, pp. 152–166, 2023.

[27] M. Li, A. Vardasbi, A. Yates, and M. De Rijke, "Repetition and exploration in sequential recommendation," in *Proceedings of the 46th international ACM SIGIR Conference on Research and Development in Information Retrieval*, 2023, pp. 2532–2541.

[28] Z. Wan, X. Liu, B. Wang, J. Qiu, B. Li, T. Guo, G. Chen, and Y. Wang, "Spatio-temporal contrastive learning-enhanced gnns for session-based recommendation," *ACM Transactions on Information Systems*, vol. 42, no. 2, pp. 1–26, 2023.

[29] Y. Yang, C. Huang, L. Xia, Y. Liang, Y. Yu, and C. Li, "Multi-behavior hypergraph-enhanced transformer for sequential recommendation," in *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, 2022, pp. 2263–2274.

[30] Y. Han, H. Wang, K. Wang, L. Wu, Z. Li, W. Guo, Y. Liu, D. Lian, and E. Chen, "Efficient noise-decoupling for multi-behavior sequential recommendation," in *Proceedings of the ACM Web Conference 2024*, 2024, pp. 3297–3306.

[31] Y. Wang, A. Javari, J. Balaji, W. Shalaby, T. Derr, and X. Cui, "Knowledge graph-based session recommendation with session-adaptive propagation," in *Companion Proceedings of the ACM Web Conference 2024*, 2024, pp. 264–273.

[32] Z. Pan, F. Cai, W. Chen, H. Chen, and M. De Rijke, "Star graph neural networks for session-based recommendation," in *Proceedings of*

the 29th ACM international conference on information & knowledge management, 2020, pp. 1195–1204.
[33] Y. Zheng, S. Liu, Z. Li, and S. Wu, "Dgtn: Dual-channel graph transition network for session-based recommendation," in *2020 international conference on data mining workshops (ICDMW)*. IEEE, 2020, pp. 236–242.
[34] Z.-H. Deng, C.-D. Wang, L. Huang, J.-H. Lai, and P. S. Yu, "G 3 sr: Global graph guided session-based recommendation," *IEEE transactions on neural networks and learning systems*, vol. 34, no. 12, pp. 9671–9684, 2022.
[35] Z. Si, L. Guan, Z. Sun, X. Zang, J. Lu, Y. Hui, X. Cao, Z. Yang, Y. Zheng, D. Leng *et al.*, "Twin v2: Scaling ultra-long user behavior sequence modeling for enhanced ctr prediction at kuaishou," in *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, 2024, pp. 4890–4897.
[36] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. 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*, 2020, pp. 639–648.
[37] H. Zeng, Z. He, Z. Yue, J. McAuley, and D. Wang, "Fair sequential recommendation without user demographics," in *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2024, pp. 395–404.
[38] Z. Zhang, R. Chen, S. Liu, Z. Yao, O. Ruwase, B. Chen, X. Wu, Z. Wang *et al.*, "Found in the middle: How language models use long contexts better via plug-and-play positional encoding," *Advances in Neural Information Processing Systems*, vol. 37, pp. 60 755–60 775, 2024.
[39] A. Klenitskiy, A. Volodkevich, A. Pembek, and A. Vasilev, "Does it look sequential? an analysis of datasets for evaluation of sequential recommendations," in *Proceedings of the 18th ACM Conference on Recommender Systems*, 2024, pp. 1067–1072.
[40] L. Wang, X. Zhang, H. Su, and J. Zhu, "A comprehensive survey of continual learning: Theory, method and application," *IEEE transactions on pattern analysis and machine intelligence*, vol. 46, no. 8, pp. 5362–5383, 2024.
[41] H. Yoo, S. Kang, R. Qiu, C. Xu, F. Wang, and H. Tong, "Embracing plasticity: Balancing stability and plasticity in continual recommender systems," in *Proceedings of the 48th International ACM SIGIR conference on research and development in Information Retrieval*, 2025, pp. 2092–2101.
[42] P. Zhang, J. Guo, C. Li, Y. Xie, J. B. Kim, Y. Zhang, X. Xie, H. Wang, and S. Kim, "Efficiently leveraging multi-level user intent for session-based recommendation via atten-mixer network," in *Proceedings of the sixteenth ACM international conference on web search and data mining*, 2023, pp. 168–176.
[43] W. Kang and J. J. McAuley, "Self-attentive sequential recommendation," in *IEEE International Conference on Data Mining, ICDM 2018, Singapore, November 17-20, 2018*. IEEE Computer Society, 2018, pp. 197–206.
[44] M. Quadrana, P. Cremonesi, and D. Jannach, "Sequence-aware recommender systems," *ACM computing surveys (CSUR)*, vol. 51, no. 4, pp. 1–36, 2018.
[45] N. Feng, J. Pan, J. Wu, B. Chen, X. Wang, Q. Li, X. Hu, J. Jiang, and M. Long, "Long-sequence recommendation models need decoupled embeddings," *arXiv preprint arXiv:2410.02604*, 2024.
[46] D. Yu, Q. Li, H. Yin, and G. Xu, "Causality-guided graph learning for session-based recommendation," in *Proceedings of the 32nd ACM international conference on information and knowledge management*, 2023, pp. 3083–3093.
[47] Q. Zhang, X. Zhou, X. Zhang, X. Yang, B. Wang, and X. Yi, "Unified empirical evaluation and comparison of session-based recommendation algorithms," *ACM Computing Surveys*, vol. 57, no. 10, pp. 1–39, 2025.
[48] B. Hidasi, "Session-based recommendations with recurrent neural networks," in *The International Conference on Learning Representations (ICLR)*, 2016.
[49] Z. Wang, W. Wei, G. Cong, X.-L. Li, X.-L. Mao, and M. Qiu, "Global context enhanced graph neural networks for session-based recommendation," in *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*, 2020, pp. 169–178.
[50] Y. Liao, Y. Yang, M. Hou, L. Wu, H. Xu, and H. Liu, "Mitigating distribution shifts in sequential recommendation: An invariance perspective," in *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2025, pp. 1603–1613.



Linjiang Guo is currently pursuing the Ph.D. degree in the School of Computer Science at the University of Technology Sydney, under the joint supervision of Prof. Guandong Xu and Dr. Shiqing Wu. He previously received his master's degree from the University of Chinese Academy of Sciences. His research interests include temporal recommendation, sequential recommendation, and machine unlearning.



Nitin Bisht is a PhD student in the School of Computer Science at the University of Technology Sydney, Australia. He received his Master's degree from the Indian Institute of Technology Kharagpur. His research interests include machine learning, statistical modeling, and recommendation systems, with a particular focus on developing reliable, fair, and efficient algorithms for large-scale user modeling and data-driven decision applications.



Shiqing Wu is an Assistant Professor in the Faculty of Data Science at City University of Macau, Macau SAR, China. Before that, he was a Postdoctoral Research Fellow in the School of Computer Science at the University of Technology Sydney, Australia, from 2022 to 2024. He received his PhD in Computer Science from the University of Tasmania, Australia, in 2022, and a joint B.Sc. in Computer Science from Auckland University of Technology, New Zealand, and China Jiliang University, China, in 2016. His research interests include agent-based modeling, social influence analysis, recommender systems, and reinforcement learning. He has published over 30 research papers in prestigious venues, including SIGIR, ICDM, IJCAI, AAMAS, TKDE, and TOIS. He is an Editorial Board Member of Information Processing & Management.



Guandong Xu (Senior Member, IEEE) is a Chair Professor of Artificial Intelligence at the Education University of Hong Kong (EdUHK). Before joining EdUHK, he was a full Professor in Data Science at the University of Technology Sydney. His research interests include data science, recommender systems, user modeling, and social computing. He has had 300+ publications in leading journals and conferences. He is the editor-in-chief of the Human-Centric Intelligent Systems and the assistant editor-in-chief of the World Wide Web Journal. He was elevated as a fellow of the Institute of Engineering and Technology (IET) and the Australian Computer Society (ACS) in 2022 and 2021, respectively.