HiTower: Hierarchical Interest Modeling with Adaptive Gating for Personalized Recommendation

1st Nijia Mo, 2nd Bo Ren, 3rd Yongda Wei, 4th Hui Liu*
Shanghai University of International Business and Economics
School of Statistics and Data Science
Shanghai, China
21333003@suibe.edu.cn, 24349167@suibe.edu.cn, junystugh@gmail.com, liuh@suibe.edu.cn

***Abstract*—In recommendation systems, conventional approaches typically learn a fixed representation vector for each item. However, the value of an item is inherently condition-dependent; for example, the meaning and appeal of the same movie may differ significantly for teenagers and middle-aged viewers, just as the same product may hold different attractions for novice versus expert users. Existing solutions, such as incorporating attention mechanisms into item representations to embed user–item relevance, partially alleviate this issue. Yet, they are often constrained by flat modeling and struggle to dynamically adapt to the varying interests across different users.**

To address these issues, we propose a Hierarchical masked gating dual-tower model called HiTower. By introducing an additional concept pool and a dynamic, masked gating mechanism, our model can adaptively select a variable number of interest concepts based on user interaction sequences. These activated concepts are then used to enrich both the original sequence representations and candidate item embeddings. Furthermore, to alleviate the collapse effect of softmax in interest activation, we propose a layer-wise activation strategy that enables the model to attend to multiple focuses hierarchically. Experimental results on three real-world datasets demonstrate that HiTower consistently outperforms state-of-the-art baseline methods, validating the effectiveness and practical value of the proposed framework in user interest modeling.

I. Introduction

In recent years, deep learning-based recommendation algorithms have made significant strides in modeling user preferences based on their historical interaction sequences [5], [8], [10], [13], [34]. Most existing models primarily focus on capturing the correlation between user sequences and candidate items directly [3], [4], [11], [14], [25]. However, these methods ignore the different expressions of user interests under different attributes. For example, as shown in Figure 1, Amy clicked on movies {*Iron Man 1, Iron Man 2, The Avengers, Iron Man 3, The Avengers, The Avengers: Age of Ultron*}. If her interest focuses on the movie IP "Marvel", she might not click on *Oppenheimer*. However, if the user is a huge fan of Robert Downey Jr., maybe she would click on that animated film. Correlation modeling cannot explicitly identify user interests. In recommendation systems, understanding the representation of the same candidate item under different interests is important as it affects the recommendation performance [20], [23].

*Corresponding Author: Hui Liu.

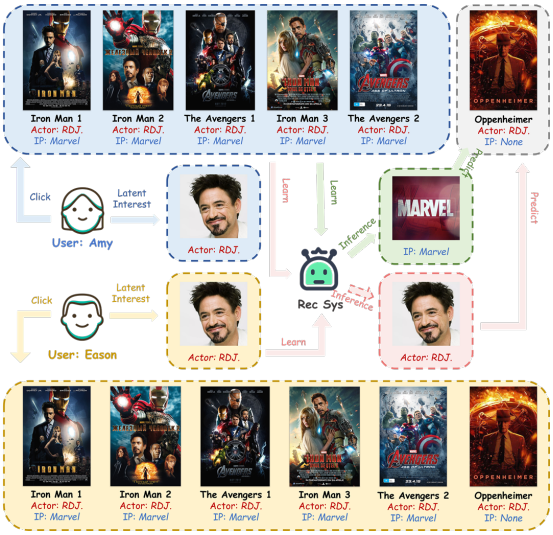

Fig. 1: Example of user behaviors. If Recsys were to directly learn Amy's interests, it might learn "Marvel" (Green), which doesn't match Amy's true interest "RDJ". However, if a concept sharing pool is added during the reasoning process, telling Recsys that there is a similar user, Eason, whose interest is "RDJ", then Recsys might be able to learn the correct interest (Pink).

Methods for learning interest representations of users can be primarily categorized into two types: user-independent learning approaches and shared concept pool-based approaches. The first category of methods [2], [16], [17] leverage the user's own behavioral sequence and employ self-attention or sequential modeling techniques to extract diverse interest representations. However, they lack an explicit cross-user knowledge sharing mechanism. Each user's interest representation is learned solely from their own history, without patterns from other users, as shown in green arrows on Figure 1.

The second category of methods [18], [26], introduces a shared learnable concept pool which is shared across the entire user population, with subsets of which being dynamically selected by attention or gating mechanisms for each user. This design enables the model to learn general interest patterns across users, achieve knowledge transfer among users, and provide interpretable interest concept labels.

These models have two drawbacks. First, they all use inner

integrals or softmax as the means of concept activation, and we will show that the model's predictive entropy increases according to the dimension of the input in Section III-B, which means model's predictions will become uncertain. Second, different users have different numbers of interests and preferences. Previous models set the number of interests to a fixed hyperparameter instead of personalizing them, thus failing to truly model user preferences.

Therefore, we designed a hierarchical activation structure that not only reduces prediction uncertainty but also allows for different numbers of activation preferences to be obtained for different users. Overall, we propose a novel recommendation framework named **HiTower**, a dual-tower framework for hierarchical interest activation. Our contributions are summarized as follows:

- We explicitly introduce the concepts to generate user preferences and candidate item representations, enabling the model to adaptively capture user interests under varying aspects.
- We propose a layer-wise hierarchical masked gating mechanism that dynamically selects relevant concept vectors from a global concept pool at each layer, which reduces the entropy of the models.
- We conduct extensive experiments on multiple benchmark datasets, showing that our approach outperforms multiple baselines. Further analysis demonstrates the effectiveness of our method in capturing user interests and improving recommendation performance.

II. Related Work

A. Models without Latent Preference

Some studies directly convert user sequences into preferences without implicit modeling, using models like DNNs [4], [11], [32], [35], [36] to capture user-item relationships via deep networks. RNNs and CNNs are widely adopted to capture sequential patterns [12], [21], [24]. However, these models often process historical sequences and items separately, limiting feature interaction and failing to capture interest diversity. Subsequent work [9], [27], [33] thus incorporates interaction features between sequences and items. Recently, attention-based [14], [25], [35], [36] models have addressed long-range dependencies and gained widespread adoption.

B. Models with Latent Preference

Some works learn latent user preferences. For example, early collaborative filtering and matrix factorization methods such as MF [15] and NeuMF [7] explicitly map users and items into a low-dimensional latent vector space to capture second-order feature interactions. With the rise of neural networks, a series of neural-based algorithms have been proposed [2], [16], [17], [29]–[31], which represent users' diverse interests through various activation mechanisms. For instance, MIND [16], ComiRec [2] and DGCF [29] leverage dynamic routing to perform clustering. These models extract latent variables endogenously, but they require assumptions on the prior distribution of the latent variables, and the high dimensionality of these latent vectors can make model training difficult.

The other works extract exogenous variables by explicitly introducing a concept pool and using dynamic routing to model users' multi-interest preferences [18], [26]. Such methods do not need to store multiple interests for each user explicitly, and only require a number of concepts much smaller than the total number of users, significantly reducing storage costs. With the advent of large language models, some works utilize their summarization capability to construct interpretable user preference profiles [19], [22]. For example, LACE [19] uses LLMs to extract concepts from user sequences, while URP [22] further verbalizes these concepts into natural language to obtain interpretable user profiles.

However, these models predefine the number of interests for each user and rely on various activation mechanisms. In reality, highly active users may generate far more interests than less active users. Therefore, a more dynamic and personalized interest activation method is needed.

III. Methodology

A. Task Formulation

Given a historical sequence embedding $X \in \mathbb{R}^{B\times N\times D}$ and a concept pool matrix $Z \in \mathbb{R}^{K\times D}$, where B is the batch size, N is the length of the sequence, D is the embedding dimension, and K is the number of concepts in the whole concept pool. Our goal is to predict the probability of interaction with candidate items $C \in \mathbb{R}^{B\times D}$. We explicitly introduce the learnable concept Z, and the recommendation task as shown in Equation 1, where y denotes the probability of click or like.

$$\begin{aligned} p(y|X,C) &= \sum_{z\in Z} p(y|X,C,z)\cdot p(z|X,C) \\ &= \sum_{z\in Z} p(y|X,C,z)\cdot \frac{p(C|z,X)p(z|X)}{p(C|X)} \end{aligned} \tag{1}$$

B. Hierarchical Activation

Since the latent variable z cannot be directly observed, the core task is to estimate $p(z \mid X)$. While softmax is widely used to model categorical distributions due to its differentiability and normalization, it tends to yield high-entropy outputs when the model is uncertain, dispersing probability mass evenly across multiple concepts and reflecting low confidence in identifying the truly activated one. Thus, directly relying on softmax-based activation introduces unnecessary uncertainty and may dilute the contribution of salient concepts, especially as the number of latent concepts grows.

Motivated by this observation, we propose a heuristic hierarchical activation mechanism shown in Algorithm 1 that progressively narrows down the candidate space and enforces low-entropy activation at each level. Instead of activating all concepts simultaneously, the proposed method decomposes the concept selection process into multiple stages, where coarse-grained concepts are first identified and then refined into fine-grained ones.

This hierarchical design effectively suppresses spurious activations and encourages confident concept selection. We

Algorithm 1 Hierarchical Masked Activation

Require: Input sequence $\mathbf{x} \in \mathbb{R}^{N\times d}$, concept set $\mathcal{Z}$ with C concepts, representing as $\mathbf{z} \in \mathbb{R}^{C\times d}$, number of groups G, threshold $\tau \in \mathbb{R}$
Ensure: Activation weight vectors $s = \{s^{(1)}||\dots||s^{(G)}\}$
Initialize global mask $\mathbf{M} \leftarrow [\text{False}]^C$
for $g = 1$ **to** G **do**
  Initialize score vector $s^{(g)} \leftarrow \mathbf{0}^C$
  Initialize candidate set $\mathcal{Z}_{\text{ac}}^{(g)} \leftarrow \emptyset$
  for $i = 1$ **to** C **do**
    if $M_i = \text{False}$ **then**
      Compute gating score: $s_i^{(g)} \leftarrow \sigma(\mathbf{x}, \mathbf{z}_i)$ {σ: similarity function}
    else
      $s_i^{(g)} \leftarrow -\infty$ {Already activated concepts are excluded}
    end if
  end for
  Determine $\mathcal{Z}_{\text{ac}}^{(g)}$: $\mathcal{Z}_{\text{ac}}^{(g)}$.add$\{i \mid s_i^{(g)} > \tau\}$
  Normalize scores over $\mathcal{Z}_{\text{ac}}^{(g)}$:

$$s^{(g)} \leftarrow \text{softmax}(s^{(g)}) \quad \forall i \in \mathcal{Z}_{\text{ac}}^{(g)}$$

  Update mask: $\mathbf{M}_i \leftarrow \text{True}$ for all $i \in \mathcal{Z}_{\text{ac}}^{(g)}$
  Get activated concepts: $\mathbf{z}_{\text{ac}}^{(g)} \leftarrow \{\mathbf{z}_i \mid i \in \mathcal{Z}_{\text{ac}}^{(g)}\}$
  Update $\mathbf{x}$: $\mathbf{x} \leftarrow \delta(\mathbf{x}, \mathbf{z}_{\text{ac}}^{(g)}, s^{(g)})$ {δ: combination function}
end for

return $s = \{s^{(1)}||\dots||s^{(G)}\}$

will prove it from the perspective of entropy. Let Z denote the concept embeddings with size $|\mathcal{Z}|$. The logit vector $\mathbf{z} = [z_1, \dots, z_{|\mathcal{Z}|}]^\top$ is computed via the inner product of Z and the fused input sequence vector $\mathbf{h}$, where each element z_i represents the alignment score between the input and the i-th concept embedding. We define the softmax distribution as Equation 2.

$$p_i^{\text{soft}}(\mathbf{z}) = \frac{e^{z_i}}{\sum_{j=1}^{|\mathcal{Z}|} e^{z_j}}, \quad i = 1, \dots, |\mathcal{Z}|. \tag{2}$$

We prove that the entropy of $\text{softmax}(\mathbf{z})$ is directly proportional to $\log|\mathcal{Z}|$, as equal to Lemma 1.

Lemma 1 (Entropy Scaling of Softmax Attention)**.** *The expected entropy scales logarithmically with $|\mathcal{Z}|$ is as Equation 3.*

$$\mathbb{E}\left[\mathcal{H}(p^{\text{soft}}(\mathbf{z}))\right] = \Theta(\log|\mathcal{Z}|). \tag{3}$$

Proof. We analyze the entropy of the softmax distribution via asymptotic expansion. We assume that $\{z_i\}_{i=1}^{|\mathcal{Z}|}$ are variables with bounded moments that satisfy exchangeability, under which a generalized law of large numbers holds.

Under these mild conditions, we have $\frac{1}{|\mathcal{Z}|}\sum_{j=1}^{|\mathcal{Z}|} e^{z_j} = \mathbb{E}[e^z] + \epsilon_z$, where ϵ_z is a random error term satisfying $\mathbb{E}[\epsilon_z] = 0$ and $\text{Var}(\epsilon_z) = O(1/|\mathcal{Z}|)$. Consequently, $\sum_{j=1}^{|\mathcal{Z}|} e^{z_j}$ could be denote as $|\mathcal{Z}|\big(\mathbb{E}[e^z] + \epsilon_z\big)$.

Thus, the entropy can be written following Equation 4.

$$\begin{aligned}
&\mathcal{H}(p^{\text{soft}}(\mathbf{z})) = -\sum_{i=1}^{|\mathcal{Z}|} p_i^{\text{soft}}(\mathbf{z}) \log p_i^{\text{soft}}(\mathbf{z}) \\
&= -\sum_{i=1}^{|\mathcal{Z}|} \frac{e^{z_i}}{|\mathcal{Z}|(\mathbb{E}[e^z] + \epsilon_z)} \log \frac{e^{z_i}}{|\mathcal{Z}|(\mathbb{E}[e^z] + \epsilon_z)} \\
&= -\frac{1}{|\mathcal{Z}|(\mathbb{E}[e^z] + \epsilon_z)} \sum_{i=1}^{|\mathcal{Z}|} e^{z_i}\Big[z_i - \log\big(|\mathcal{Z}|(\mathbb{E}[e^z] + \epsilon_z)\big)\Big] \\
&= \log|\mathcal{Z}| + \log\big(\mathbb{E}[e^z] + \epsilon_z\big) - \frac{\frac{1}{|\mathcal{Z}|}\sum_{i=1}^{|\mathcal{Z}|} e^{z_i} z_i}{\mathbb{E}[e^z] + \epsilon_z}.
\end{aligned} \tag{4}$$

Introducing a second error term $\delta_Z = \frac{1}{|\mathcal{Z}|}\sum_{i=1}^{|\mathcal{Z}|} e^{z_i} z_i - \mathbb{E}[e^z z]$, which also satisfies $\mathbb{E}[\delta_z] = 0$, we can further derive Equation 4 to obtain Equation 5.

$$\mathcal{H}(p^{\text{soft}}(\mathbf{z})) = \log|\mathcal{Z}| + \log\big(\mathbb{E}[e^z] + \epsilon_z\big) - \frac{\mathbb{E}[e^z z] + \delta_z}{\mathbb{E}[e^z] + \epsilon_z}. \tag{5}$$

Taking expectations and expanding to first order $O(\frac{1}{|\mathcal{Z}|})$ in the small fluctuations ϵ_z, δ_z, and $\log\mathbb{E}[e^z] - \frac{\mathbb{E}[e^z z]}{\mathbb{E}[e^z]}$ yielding a constant, we obtain $\mathbb{E}[\mathcal{H}(p^{\text{attn}}(\mathbf{z}))] = \Theta(\log|\mathcal{Z}|)$. □

Lemma 2 (Entropy of Hierarchical Gating)**.** *Partition the $|\mathcal{Z}|$ indices into G disjoint groups of sizes $|\mathcal{Z}|_1, \dots, |\mathcal{Z}|_G$ with $\sum_{g=1}^{G}|\mathcal{Z}|_g = |\mathcal{Z}|$. Let $\mathbf{z}^g$ denote the sub-vector corresponding to group g and define the local softmax distributions $p^g(\mathbf{z}^g) = \text{softmax}(\mathbf{z}^g), \quad g = 1, \dots, G$. Since $p^{\text{hier}}(\mathbf{z}) = \frac{1}{G}\big[p^1(\mathbf{z}^1)\|p^2(\mathbf{z}^2)\|\cdots\|p^G(\mathbf{z}^G)\big]$, its expected entropy satisfies the following.*

$$\mathbb{E}\big[\mathcal{H}(p^{\text{hier}}(\mathbf{z}))\big] = \log G + \frac{1}{G}\sum_{g=1}^{G}\Theta(\log|\mathcal{Z}|_g). \tag{6}$$

Proof. By the chain rule of entropy, the hierarchical distribution obeys Equation 7.

$$\mathcal{H}(p^{\text{hier}}(\mathbf{z})) = \log G + \frac{1}{G}\sum_{g=1}^{G}\mathcal{H}(p^g(\mathbf{z}^g)). \tag{7}$$

Taking expectations and applying Lemma 1 to each group gives the Equation 8,

$$\begin{aligned}
\mathbb{E}\big[\mathcal{H}(p^{\text{hier}}(\mathbf{z}))\big] &= \log G + \frac{1}{G}\sum_{g=1}^{G}\mathbb{E}\big[\mathcal{H}(p^g(\mathbf{z}^g))\big] \\
&= \log G + \frac{1}{G}\sum_{g=1}^{G}\Theta(\log|\mathcal{Z}|_g)
\end{aligned} \tag{8}$$

where the $\Theta(\cdot)$ notation captures the logarithmic growth with the group size $|\mathcal{Z}|_g$, exactly as in Lemma 1. □

Theorem 1 (Hierarchical Entropy Reduction)**.** *Under the same setting as Lemmas 1 and 2, the hierarchical gating mechanism reduces entropy compared to flat softmax in the following probabilistic sense, as shown in Equation 9.*

$$\Pr\left(\frac{1}{G}\sum_{g=1}^{G}\Theta(\log|\mathcal{Z}|_g) < \Theta(\log|\mathcal{Z}|)\right) \xrightarrow[G\to\infty]{} 1 \tag{9}$$

In particular, for any fixed $|\mathcal{Z}|$, as the number of groups G increases, the hierarchical entropy becomes strictly smaller than the flat softmax entropy with probability approaching 1.

Proof. Comparing the two entropy expressions by Jensen's inequality, the probability of The hierarchical activation yields a lower entropy than the vanilla softmax is shown in Equation 10.

$$\begin{aligned} &\Pr\left(\frac{1}{G}\sum_{g=1}^{G}\Theta(\log|\mathcal{Z}|_g) < \Theta(\log|\mathcal{Z}|)\right) \\ =&\Pr\left(\frac{1}{G}\sum_{g=1}^{G}\log|\mathcal{Z}|_g < \log|\mathcal{Z}|\right) \\ \geq&\Pr\left(\log\left(\frac{1}{G}\sum_{g=1}^{G}|\mathcal{Z}|_g\right) < \log|\mathcal{Z}|\right) \xrightarrow[G\to\infty]{} 1 \end{aligned} \tag{10}$$

As G grows, $\log(|\mathcal{Z}|/G)$ becomes strictly smaller than $\log|\mathcal{Z}|$ (since $G > 1$). Consequently, the gap between the left-hand side and the right-hand side widens, and the probability that the hierarchical entropy is strictly below the flat entropy converges to 1. This completes the proof. □

C. Model Architecture

The overall architecture adopts a dual-tower structure shown in Figure 2. The left tower processes the historical sequence X through a backbone network $f_\theta(\cdot)$, while the right tower encodes the candidate item C. A bridge between the towers is built by integrating concept vectors Z via a hierarchical masked gating mechanism, which adaptively selects relevant concepts at each layer to construct both the user representation and the candidate item representation. Their combination yields the final interaction probability.

D. Initialization

1) Concepts: We initialize the Z with orthogonal vectors to ensure diversity and reduce redundancy as $Z = \text{Orthogonal}(Z)$.

2) Global Memory: We initialize three global variables for each user in a batch: the activated concept vectors $Z_{\text{activated}} \in \mathbb{R}^{B\times C\times D}$, the activated scores $G_{\text{activated}} \in \mathbb{R}^{B\times C}$, and a binary mask $M \in \{0,1\}^{B\times C}$. All these variables are initialized to zero at the beginning of each training iteration.

$$Z_{\text{activated}} = \mathbf{0}, \quad G_{\text{activated}} = \mathbf{0}, \quad M = \mathbf{0} \tag{11}$$

E. User Preference

1) Dynamic Activation: First, To integrate the concept Z into the sequence modeling, we initialize a trainable gating parameter $W_x, W_z \in \mathbb{R}^{D\times D}$, then we map the input sequence $X^{(l-1)}$ and the concept Z to the same dimension followed 12.

$$h_b^{(l)} = \sum_{i=1}^{N} X_i^{(l-1)} W_x^{(l)}, \quad Z_{\text{proj}}^{(l)} = ZW_z^{(l)} \tag{12}$$

where $h_b^{(l)} \in \mathbb{R}^{1\times D}$ is the mapped sequence representation for the bth user in a batch, and $Z_{\text{proj}}^{(l)} \in \mathbb{R}^{C\times D}$ is the mapped concept representation. Then we compute the gating score $s_b^{(l)} \in \mathbb{R}^{1\times C}$ via inner product $s_b^{(l)} = h_b^{(l)} \cdot (Z_{\text{proj}}^{(l)})^T$.

Then, we select the concepts that score higher than a threshold τ, and set the remaining scores to $-\infty$ to prevent them from being activated, as shown in Equation 13. We set τ to 0 in our experiments.

$$s_{b,c}^{(l)} \leftarrow \begin{cases} s_{b,c}^{(l)} & \text{if } s_{b,c}^{(l)} > \tau \\ -\infty, & \text{if } s_{b,c}^{(l)} \leq \tau \end{cases} \tag{13}$$

2) Hierarchical Masked Gating: We propose a hierarchical masked gating mechanism that activates concepts layer by layer. Each layer means each group. We first initialize a global binary mask $M \in \mathbb{R}^{B\times C}$ with all zeros for each user in a batch, where 0 indicates that the concept is available for activation, and 1 indicates that the concept has been activated. The masked gating score $g_b^{(l)} \in \mathbb{R}^{1\times C}$ is computed as 14.

$$s_{b,c}^{(l)} \leftarrow \begin{cases} s_{b,c}^{(l)} & \text{if } m_{b,c} = 0 \\ -\infty, & \text{if } m_{b,c} = 1 \end{cases} \tag{14}$$

In this way, there are two kinds of masks to prevent the concept from being activated: (1) the concept score is lower than the threshold τ, and (2) the concept has been activated before. This ensures that each layer can only activate concepts that have not been activated in previous layers, effectively capturing the hierarchical relationship among concepts.

We apply softmax to obtain the gating weight $G_b^{(l)} \in \mathbb{R}^{1\times C}$ for the b-th user in a batch via $G_b^{(l)} = \text{softmax}(s_b^{(l)})$. If all the concepts have been activated before, the softmax score will be divided into $1/K$, which means that all concepts will be activated equally. To prevent this, we reset the score to all zeros, as shown in Equation 15. This ensures that the model can continuously adapt to new information and capture evolving user preferences over time.

$$G_b^{(l)} \leftarrow \begin{cases} G_b^{(l)} & \text{if } \sum_{c=1}^{C} m_{b,c} < C \\ 0 & \text{if } \sum_{c=1}^{C} m_{b,c} = C \end{cases} \tag{15}$$

3) Residual Connection: Finally, we get the chosen concept vector $Z_{b,\text{selected}}^{(l)}$ for the b-th user in a batch $Z_b^{(l)} = G_b^{(l)} \cdot Z$, where $Z_b^{(l)}$ represents the activated concept vector for the b-th user in a batch at layer l. For obtaining the remaining information for the next layer, a subtraction operation is applied to the input sequence $X^{(l-1)}$ and the activated concept vector $Z^{(l)}$, as shown as $X^{(l)} = X^{(l-1)} - \sum_{c=1}^{C} Z_c^{(l)}$.

4) Update Global Memory: After getting the activated concepts, we update the global memory for the next layer, as shown in Equation 16.

$$Z_{\text{activated}} \leftarrow Z_{\text{activated}} + Z^{(l)} \tag{16a}$$

$$G_{\text{activated}} \leftarrow G_{\text{activated}} + G^{(l)} \tag{16b}$$

$$M \leftarrow M \vee \text{Bool}(S^{(l)}) \tag{16c}$$

5) Candidate Representation: The l-th layered representation of candidate item $C^{(l)}$ is constructed as a linear combination of the activated concept $Z^{(l)}$ by $C^{(l)} = G^{(l)} \cdot Z^{(l)}$.

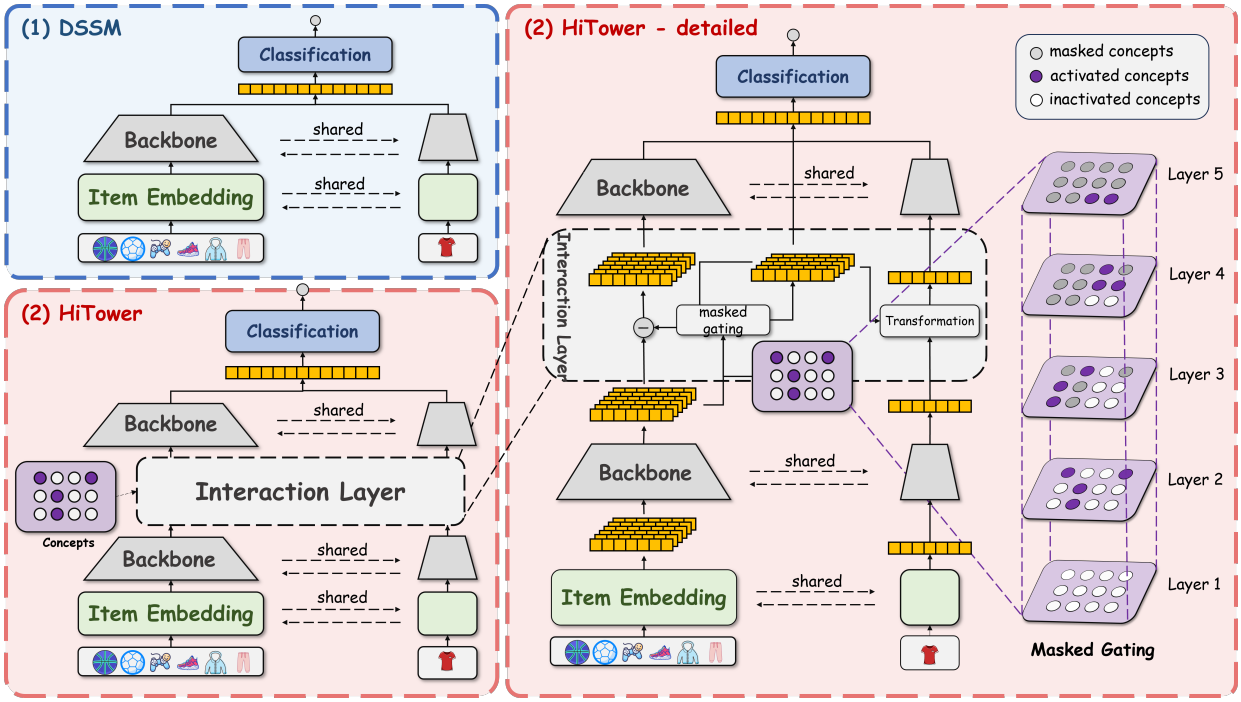

Fig. 2: The architecture of the proposed model. DSSM indicates the base dual semantic similarity matching structures.

F. Prediction

After processing through L layers of the backbone, we obtain dynamically activated concepts for each user in a batch. The final user representation $U \in \mathbb{R}^D$ is computed by the sum of the last layer's output sequence and the related activated concept vectors, as shown in Equation 17. The final candidate item representation C is $C^{(L)}$.

$$U = X^{(L)} + \frac{1}{L}(G_{\text{activated}})Z_{\text{activated}} \tag{17}$$

We concatenate the user representation U and the candidate item representation C, and pass them through a multi-layer perceptron (MLP) to predict the interaction probability $y \in \mathbb{R}^2$, as shown in Equation 18.

$$y = \text{MLP}([U; C]) \tag{18}$$

G. Loss Function

We use the binary cross-entropy loss as $\mathcal{L}_{\text{BCE}}$ to train the model. To make sure the Z is independent, we add a regularization term to the loss function to encourage orthogonality among the concept vectors as $\mathcal{L}_{\text{reg}} = ||ZZ^T - I||_F^2$, where I is the identity matrix, and $||\cdot||_F$ denotes the Frobenius norm. The final loss function is a combination of the binary cross-entropy loss and the regularization term, as $\mathcal{L} = \mathcal{L}_{\text{BCE}} + \mathcal{L}_{\text{reg}}$.

IV. Experiments

A. Datasets and Baselines

We evaluate our proposed method on three real-world datasets: MovieLens-1M [6], BookCrossing [37], and LastFM [1].

We compare our method with two categories of baselines: (1) latent preference models, and (2) DSSMs with various backbones. For preference modeling, we select **MIND** [16], **SINE** [26], and **MINER** [17]. For DSSM-based sequential recommenders, we implement **GRU** [9], **NextItNet** [33], **SASRec** [14], and **BERT** [25], [28]. Our method is combined with all above backbones, denoted as Ours-GRU, Ours-NextItNet, Ours-SAS, and Ours-BERT.

We do not include the latest large language models as baselines, since the LLM incorporates a lot of external knowledge, and the more parameters a model has, the better it performs. Our purpose is to verify the effectiveness of this activation framework.

B. Overall Performance

We use Hit Rate (HR), Normalized Discounted Cumulative Gain (NDCG), and Mean Reciprocal Rank (MRR) as the evaluation metrics.

From Table I, we can see that our proposed method significantly outperforms various baseline models on all three datasets, including MIND, SINE, MINER, and DSSMs with different architectures. On the MovieLens-1M dataset, all DSSM methods outperform the original multi-interest methods, and our method achieves the best performance when the backbone model is BERT. On the BookCrossing dataset, MIND performs slightly better than DSSM-GRU, DSSM-Next, and DSSM-BERT, mainly because the user historical behavior sequences in this dataset are relatively short, with an average of only two interactions per user. In such a short-sequence environment, MIND's multi-interest mechanism is more robust compared to GRU, Next, or Transformer, which rely on sequence length. However, our method still significantly outperforms all baselines, indicating that the introduced concepts can effectively decompose and supplement interests

TABLE I: Overall performance comparison of our proposed method with baseline models on three datasets.

Dataset	Model	NDCG@5	NDCG@10	MRR	HR@1	HR@5	HR@10
MovieLens-1M	MIND	0.3164±0.0009	0.3698±0.0011	0.3122±0.0007	0.1706±0.0003	0.4523±0.0018	0.6184±0.0026
	SINE	0.3195±0.0007	0.3715±0.0008	0.3138±0.0007	0.1727±0.0008	0.4581±0.0017	0.6197±0.0031
	MINER	0.3203±0.0010	0.3727±0.0009	0.3141±0.0009	0.1719±0.0008	0.4586±0.0021	0.6214±0.0037
	DSSM-GRU	0.4262±0.0019	0.4784±0.0007	0.4067±0.0018	0.2483±0.0007	0.5911±0.0035	0.7522±0.0037
	Ours-GRU	**0.4522±0.0024**	**0.5029±0.0012**	**0.4278±0.0023**	**0.2668±0.0010**	**0.6240±0.0041**	**0.7812±0.0043**
	DSSM-Next	0.4392±0.0026	0.4904±0.0032	0.4163±0.0021	0.2557±0.0011	0.6096±0.0035	0.7676±0.0039
	Ours-Next	**0.4561±0.0027**	**0.5063±0.0023**	**0.4320±0.0024**	**0.2703±0.0013**	**0.6265±0.0047**	**0.7817±0.0046**
	DSSM-SAS	0.4480±0.0022	0.4978±0.0022	0.4251±0.0019	0.2658±0.0011	0.6168±0.0039	0.7698±0.0039
	Ours-SAS	**0.4498±0.0019**	**0.5018±0.0020**	**0.4272±0.0014**	**0.2675±0.0013**	**0.6192±0.0021**	**0.7796±0.0040**
	DSSM-BERT	0.4333±0.0021	0.4845±0.0022	0.4145±0.0019	0.2612±0.0009	0.5946±0.0031	0.7522±0.0039
	Ours-BERT	**0.4619±0.0023**	**0.5124±0.0022**	**0.4387±0.0013**	**0.2786±0.0011**	**0.6293±0.0045**	**0.7855±0.0041**
BookCrossing	MIND	0.3332±0.0017	0.3597±0.0024	0.3238±0.0013	0.2171±0.0009	0.4376±0.0026	0.5192±0.0014
	SINE	0.3022±0.0022	0.3379±0.0023	0.2908±0.0016	0.1901±0.0011	0.4152±0.0027	0.4996±0.0029
	MINER	0.3141±0.0020	0.3425±0.0025	0.3045±0.0019	0.1929±0.0011	0.4233±0.0028	0.5106±0.0032
	DSSM-GRU	0.3296±0.0021	0.3538±0.0023	0.3245±0.0019	0.2289±0.0014	0.4177±0.0019	0.4925±0.0022
	Ours-GRU	**0.3408±0.0022**	**0.3652±0.0025**	**0.3254±0.0017**	**0.2092±0.0010**	**0.4555±0.0023**	**0.5305±0.0027**
	DSSM-Next	0.3394±0.0017	0.3652±0.0019	0.3320±0.0018	0.2281±0.0013	0.4370±0.0020	0.5164±0.0027
	Ours-Next	**0.3547±0.0020**	**0.3779±0.0021**	**0.3434±0.0018**	**0.2375±0.0014**	**0.4570±0.0025**	**0.5285±0.0029**
	DSSM-SAS	0.3506±0.0017	0.3736±0.0019	0.3395±0.0017	0.2324±0.0011	0.4529±0.0021	0.5236±0.0022
	Ours-SAS	**0.3515±0.0021**	**0.3750±0.0020**	**0.3398±0.0019**	**0.2325±0.0010**	**0.4557±0.0024**	**0.5279±0.0026**
	DSSM-BERT	0.3289±0.0018	0.3542±0.0017	0.3208±0.0021	0.2167±0.0016	0.4297±0.0025	0.5081±0.0027
	Ours-BERT	**0.3407±0.0020**	**0.3697±0.0023**	**0.3373±0.0017**	**0.2353±0.0009**	**0.4341±0.0021**	**0.5237±0.0025**
LastFM	MIND	0.4582±0.0023	0.4855±0.0021	0.4349±0.0023	0.3085±0.0017	0.5929±0.0029	0.6767±0.0035
	SINE	0.4142±0.0015	0.4360±0.0016	0.4104±0.0009	0.2744±0.0021	0.5836±0.0030	0.6543±0.0031
	MINER	0.4469±0.0016	0.3990±0.0018	0.4283±0.0014	0.2803±0.0026	0.5968±0.0028	0.6658±0.0033
	DSSM-GRU	0.4286±0.0019	0.4667±0.0028	0.4129±0.0026	0.2849±0.0017	0.5573±0.0039	0.6740±0.0024
	Ours-GRU	**0.4845±0.0011**	**0.5122±0.0029**	**0.4627±0.0024**	**0.3381±0.0014**	**0.6132±0.0028**	**0.6986±0.0033**
	DSSM-Next	0.4508±0.0019	0.4843±0.0020	0.4357±0.0023	0.3156±0.0017	0.5726±0.0028	0.6745±0.0037
	Ours-Next	**0.4798±0.0021**	**0.5087±0.0024**	**0.4587±0.0022**	**0.3337±0.0015**	**0.6071±0.0036**	**0.6964±0.0041**
	DSSM-SAS	0.4388±0.0021	0.4751±0.0017	0.4227±0.0020	0.2910±0.0019	0.5666±0.0026	0.6712±0.0039
	Ours-SAS	**0.4598±0.0021**	**0.4866±0.0020**	**0.4388±0.0022**	**0.3156±0.0016**	**0.5890±0.0027**	**0.6773±0.0038**
	DSSM-BERT	0.4459±0.0011	0.4772±0.0021	0.4221±0.0021	0.2893±0.0019	0.5797±0.0022	0.6838±0.0036
	Ours-BERT	**0.4470±0.0013**	**0.4817±0.0023**	**0.4280±0.0021**	**0.2959±0.0017**	**0.5885±0.0024**	**0.6871±0.0032**

even in extremely short sequences. On the LastFM dataset, MIND still outperforms most DSSM models, but our model achieves the best results, especially with the BERT backbone, where the performance improvement is more pronounced. Overall, our method performs excellently across all datasets and evaluation metrics, fully validating its effectiveness in capturing diverse user interests and enhancing recommendation performance.

C. Ablation Study

1) w/o Concepts: The result in Table I shows the effectiveness of the concept $K = 0$, which is DSSMs. We achieve significant improvements over the baselines across all datasets and backbones. This demonstrates that our proposed method effectively captures diverse user preferences and item characteristics, leading to improved recommendation performance.

2) Activation Mechanism: We evaluate the effectiveness of the dynamic gating mechanism by replacing it with 1) non-masked gating and 2) a softmax function on the MovieLens-1M and LastFM using BERT as the backbone. In addition, to evaluate the confidence level of recommendations, we introduce **Entropy@K**, which measures the uncertainty of the model's predictions in the top-K list.

TABLE II: Ablation study results on gating method over MovieLens-1M and LastFM datasets.

(a) MovieLens-1M

Metric	masked gating	gating	softmax
↑MRR	**0.4387**	0.4214	0.4217
↑NDCG@10	**0.5124**	0.4929	0.4947
↑HR@10	**0.7855**	0.7630	0.7688
↓Entropy@5	0.8188	**0.7875**	0.8326
↓Entropy@10	0.8561	**0.8380**	0.8631

(b) LastFM

Metric	masked gating	gating	softmax
↑MRR	**0.4280**	0.4203	0.4212
↑NDCG@10	**0.4817**	0.4800	0.4808
↑HR@10	**0.6871**	0.6792	0.6801
↓Entropy@5	**0.9201**	0.9254	0.9369
↓Entropy@10	**0.9297**	0.9298	0.9380

We observe that the masked gating mechanism consistently outperforms both the non-masked gating and softmax variants, highlighting the importance of avoiding redundant activations for capturing diverse user preferences. On sparse datasets like LastFM, however, softmax performs slightly better due to its smoother, more stable activations that prevent training instability when user interaction records are limited. Entropy analysis shows that both gating mechanisms yield more

confident activations than softmax, with non-masked gating exhibiting the lowest entropy on MovieLens-1M, likely due to its tendency to concentrate on dominant, head-effect items.

3) w/o Orthogonal Loss: The results are presented in Table III. We observe that removing the orthogonal loss leads to a consistent, albeit moderate, decline in all evaluation metrics. This result highlights the importance of enforcing diversity among concept vectors during training. The orthogonal constraint effectively prevents different concept embeddings from collapsing into similar directions in the latent space, thereby encouraging the model to capture multiple independent aspects of user interests.

TABLE III: Ablation study on the orthogonal loss over MovieLens-1M and LastFM datasets.

(a) MovieLens-1M

Metric	w/ orthogonal loss	w/o orthogonal loss
MRR	**0.4387**	0.4305
NDCG@10	**0.5124**	0.4998
HR@10	**0.7855**	0.7803

(b) LastFM

Metric	w/ orthogonal loss	w/o orthogonal loss
MRR	**0.4280**	0.4201
NDCG@10	**0.4817**	0.4797
HR@10	**0.6871**	0.6782

D. Hyperparameter Sensitivity

1) Concept Number: As presented in Figure 3, on MovieLens-1M, performance improves with more concept vectors C, peaking at $C = 32$ before declining. On sparser LastFM, optimal results occur at $C = 4$ or $C = 8$. This suggests the optimal C is tied to user interaction history richness: longer sequences benefit from more concepts to capture nuanced patterns, while excessive C risks redundancy and overfitting to noise; shorter sequences require fewer concepts for robust, generalized representations.

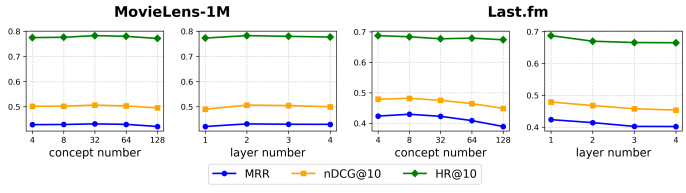

Fig. 3: Performance comparison across different numbers of concept vectors and layers.

2) Layer Number: As illustrated in Figure 3, The optimal model depth depends on dataset characteristics. Deeper models benefit MovieLens-1M's diverse user histories, while shallower architectures suit LastFM's sparse sequences to prevent overfitting. This highlights a trade-off that moderate depth enhances expressiveness, but excessive layers obscure salient patterns and harm generalization.

E. Case Study

We visualized the gating scores of 4 users on 32 concept vectors with 2 layers on the MovieLens-1M dataset. Masked gating and softmax are applied to the scores to get the gating weights. The results are shown in Figure 4.

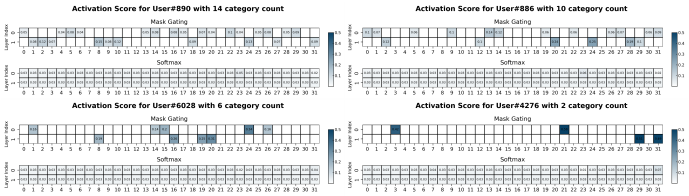

Fig. 4: Activation scores of four users.

This mechanism enables the model to selectively activate concept vectors based on individual interaction patterns, introducing sparsity that focuses on relevant concepts while suppressing noise. Unlike softmax, which smooths activations and weakens distinctiveness, masked gating adapts to varying preference richness: it activates more concepts for users with diverse histories such as User#890 and fewer for those with concentrated interests such as User#6028. This adaptability ensures robust interest representation across both sparse and dense scenarios.

V. Conclusion

In this paper, we propose a novel recommendation framework that incorporates concept vectors to enhance user and item representations. We introduced a hierarchical masked gating mechanism that selectively activates relevant concept vectors dynamically based on user behavior sequences. Our method effectively captures diverse user preferences and item characteristics, leading to improved recommendation performance.

References

[1] O. Celma, "Last.fm Music Recommendation Dataset (LastFM-360K / LastFM-1K)," 2010. [Online]. Available: https://zenodo.org/records/6090214

[2] Y. Cen, J. Zhang, X. Zou, C. Zhou, H. Yang, and J. Tang, "Controllable multi-interest framework for recommendation," *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:218685017

[3] Q. Chen, H. Zhao, W. Li, P. Huang, and W. Ou, "Behavior sequence transformer for e-commerce recommendation in alibaba," *CoRR*, vol. abs/1905.06874, 2019. [Online]. Available: http://arxiv.org/abs/1905.06874

[4] P. Covington, J. Adams, and E. Sargin, "Deep neural networks for youtube recommendations," in *Proceedings of the 10th ACM Conference on Recommender Systems*, ser. RecSys '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 191–198. [Online]. Available: https://doi.org/10.1145/2959100.2959190

[5] C. Gao, Y. Zheng, N. Li, Y. Li, Y. Qin, J. Piao, Y. Quan, J. Chang, D. Jin, X. He, and Y. Li, "A survey of graph neural networks for recommender systems: Challenges, methods, and directions," *ACM Trans. Recomm. Syst.*, vol. 1, no. 1, Mar. 2023. [Online]. Available: https://doi.org/10.1145/3568022

[6] F. M. Harper and J. A. Konstan, "The movielens datasets: History and context," *ACM Trans. Interact. Intell. Syst.*, vol. 5, no. 4, December 2015.

[7] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Proceedings of the 26th International Conference on World Wide Web*, ser. WWW '17. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee, 2017, p. 173–182. [Online]. Available: https://doi.org/10.1145/3038912.3052569

[8] Z. He, W. Liu, W. Guo, J. Qin, Y. Zhang, Y. Hu, and R. Tang, "A survey on user behavior modeling in recommender systems," in *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI 2023, 19th-25th August 2023, Macao, SAR, China*. ijcai.org, 2023, pp. 6656–6664. [Online]. Available: https://doi.org/10.24963/ijcai.2023/746

[9] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, "Session-based recommendations with recurrent neural networks," in *Proc. 4th Int. Conf. Learn. Represent. (ICLR)*, San Juan, Puerto Rico, 2016.

[10] J. Huang, J. Chen, J. Lin, J. Qin, Z. Feng, W. Zhang, and Y. Yu, "A comprehensive survey on retrieval methods in recommender systems," *ACM Trans. Inf. Syst.*, vol. 44, no. 1, Dec. 2025. [Online]. Available: https://doi.org/10.1145/3771925

[11] P.-S. Huang, X. He, J. Gao, L. Deng, A. Acero, and L. Heck, "Learning deep structured semantic models for web search using clickthrough data," in *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management*, ser. CIKM '13. New York, NY, USA: Association for Computing Machinery, 2013, p. 2333–2338. [Online]. Available: https://doi.org/10.1145/2505515.2505665

[12] T. Huang, Z. Zhang, and J. Zhang, "Fibinet: combining feature importance and bilinear feature interaction for click-through rate prediction," in *Proceedings of the 13th ACM Conference on Recommender Systems*, ser. RecSys '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 169–177. [Online]. Available: https://doi.org/10.1145/3298689.3347043

[13] D. Jannach and M. Zanker, "A survey on intent-aware recommender systems," *ACM Trans. Recomm. Syst.*, vol. 3, no. 2, Dec. 2024. [Online]. Available: https://doi.org/10.1145/3700890

[14] W.-C. Kang and J. McAuley, "Self-attentive sequential recommendation," in *2018 IEEE International Conference on Data Mining (ICDM)*, 2018, pp. 197–206.

[15] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.

[16] C. Li, Z. Liu, M. Wu, Y. Xu, H. Zhao, P. Huang, G. Kang, Q. Chen, W. Li, and D. L. Lee, "Multi-interest network with dynamic routing for recommendation at tmall," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, ser. CIKM '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 2615–2623. [Online]. Available: https://doi.org/10.1145/3357384.3357814

[17] J. Li, J. Zhu, Q. Bi, G. Cai, L. Shang, Z. Dong, X. Jiang, and Q. Liu, "MINER: Multi-interest matching network for news recommendation," in *Findings of the Association for Computational Linguistics: ACL 2022*, S. Muresan, P. Nakov, and A. Villavicencio, Eds. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 343–352. [Online]. Available: https://aclanthology.org/2022.findings-acl.29/

[18] Z. Liu, J. Lian, J. Yang, D. Lian, and X. Xie, "Octopus: Comprehensive and elastic user representation for the generation of recommendation candidates," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 289–298. [Online]. Available: https://doi.org/10.1145/3397271.3401088

[19] S. Mysore, M. Jasim, A. Mccallum, and H. Zamani, "Editable user profiles for controllable text recommendations," in *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '23. New York, NY, USA: Association for Computing Machinery, 2023, p. 993–1003. [Online]. Available: https://doi.org/10.1145/3539618.3591677

[20] T. Nguyen and A. Takasu, "NPE: neural personalized embedding for collaborative filtering," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, J. Lang, Ed. ijcai.org, 2018, pp. 1583–1589. [Online]. Available: https://doi.org/10.24963/ijcai.2018/219

[21] H. Palangi, L. Deng, Y. Shen, J. Gao, X. He, J. Chen, X. Song, and R. K. Ward, "Semantic modelling with long-short-term memory for information retrieval," *CoRR*, vol. abs/1412.6629, 2014. [Online]. Available: http://arxiv.org/abs/1412.6629

[22] J. Ramos, H. A. Rahmani, X. Wang, X. Fu, and A. Lipani, "Transparent and scrutable recommendations using natural language user profiles," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, L.-W. Ku, A. Martins, and V. Srikumar, Eds. Bangkok, Thailand: Association for Computational Linguistics, Aug. 2024, pp. 13 971–13 984. [Online]. Available: https://aclanthology.org/2024.acl-long.753/

[23] C. Ruan, A. Stewart, H. Li, R. Ye, D. Vengerov, and H. Wang, "Dynamic embedding-based retrieval for personalized item recommendations at instacart," in *Companion Proceedings of the ACM Web Conference 2023*, ser. WWW '23 Companion. New York, NY, USA: Association for Computing Machinery, 2023, p. 983–987. [Online]. Available: https://doi.org/10.1145/3543873.3587668

[24] Y. Shen, X. He, J. Gao, L. Deng, and G. Mesnil, "A latent semantic model with convolutional-pooling structure for information retrieval," in *CIKM*, November 2014.

[25] F. Sun *et al.*, "Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer," in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manage. (CIKM)*, Beijing, China, 2019, pp. 1441–1450.

[26] Q. Tan, J. Zhang, J. Yao, N. Liu, J. Zhou, H. Yang, and X. Hu, "Sparse-interest network for sequential recommendation," in *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, ser. WSDM '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 598–606. [Online]. Available: https://doi.org/10.1145/3437963.3441811

[27] J. Tang and K. Wang, "Personalized top-n sequential recommendation via convolutional sequence embedding," in *Proc. 11th ACM Int. Conf. Web Search Data Min. (WSDM)*, Marina Del Rey, CA, USA, 2018, pp. 565–573.

[28] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems (NIPS)*, Long Beach, CA, USA, 2017, pp. 5998–6008.

[29] X. Wang, H. Jin, A. Zhang, X. He, T. Xu, and T.-S. Chua, "Disentangled graph collaborative filtering," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 1001–1010. [Online]. Available: https://doi.org/10.1145/3397271.3401137

[30] Z. Xiao, L. Yang, W. Jiang, Y. Wei, Y. Hu, and H. Wang, "Deep multi-interest network for click-through rate prediction," in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, ser. CIKM '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 2265–2268. [Online]. Available: https://doi.org/10.1145/3340531.3412092

[31] Y. Xie, J. Gao, P. Zhou, Q. Ye, Y. Hua, J. B. Kim, F. Wu, and S. Kim, "Rethinking multi-interest learning for candidate matching in recommender systems," in *Proceedings of the 17th ACM Conference on Recommender Systems*, ser. RecSys '23. New York, NY, USA: Association for Computing Machinery, 2023, p. 283–293. [Online]. Available: https://doi.org/10.1145/3604915.3608766

[32] X. Yi, J. Yang, L. Hong, D. Z. Cheng, L. Heldt, A. Kumthekar, Z. Zhao, L. Wei, and E. Chi, "Sampling-bias-corrected neural modeling for large corpus item recommendations," in *Proceedings of the 13th ACM Conference on Recommender Systems*, ser. RecSys '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 269–277. [Online]. Available: https://doi.org/10.1145/3298689.3346996

[33] F. Yuan, A. Karatzoglou, I. Arapakis, J. M. Jose, and X. He, "A simple convolutional generative network for next item recommendation," in *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, ser. WSDM '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 582–590. [Online]. Available: https://doi.org/10.1145/3289600.3290975

[34] S. Zhang, L. Yao, A. Sun, and Y. Tay, "Deep learning based recommender system: A survey and new perspectives," *ACM Comput. Surv.*, vol. 52, no. 1, Feb. 2019. [Online]. Available: https://doi.org/10.1145/3285029

[35] G. Zhou, N. Mou, Y. Fan, Q. Pi, W. Bian, C. Zhou, X. Zhu, and K. Gai, "Deep interest evolution network for click-through rate prediction," in *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*, ser. AAAI'19/IAAI'19/EAAI'19. AAAI Press, 2019. [Online]. Available: https://doi.org/10.1609/aaai.v33i01.33015941

[36] G. Zhou, X. Zhu, C. Song, Y. Fan, H. Zhu, X. Ma, Y. Yan, J. Jin, H. Li, and K. Gai, "Deep interest network for click-through rate prediction," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, ser. KDD '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 1059–1068. [Online]. Available: https://doi.org/10.1145/3219819.3219823

[37] C.-N. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen, "Improving recommendation lists through topic diversification," in *Proceedings of the 14th international conference on World Wide Web (WWW '05)*. ACM, 2005, pp. 22–32.