# Disentangling Past-Future Modeling in Sequential Recommendation via Dual Networks

**Hengyu Zhang***
zhang-hy21@mails.tsinghua.edu.cn
Tsinghua Shenzhen International
Graduate School, Tsinghua University
Shenzhen, China

**Enming Yuan***
yem19@mails.tsinghua.edu.cn
Institute for Interdisciplinary
Information Sciences,Tsinghua
University
Beijing, China

**Wei Guo**
guowei67@huawei.com
Huawei Noah's Ark Lab
Shenzhen, China

**Zhicheng He**
hezhicheng9@huawei.com
Huawei Noah's Ark Lab
Shenzhen, China

**Jiarui Qin**
qinjr@icloud.com
Shanghai Jiao Tong University
Shanghai, China

**Huifeng Guo**
huifeng.guo@huawei.com
Huawei Noah's Ark Lab
Shenzhen, China

**Bo Chen**
chenbo116@huawei.com
Huawei Noah's Ark Lab
Shenzhen, China

**Xiu Li**[†]
li.xiu@sz.tsinghua.edu.cn
Tsinghua Shenzhen International
Graduate School, Tsinghua University
Shenzhen, China

**Ruiming Tang**[†]
tangruiming@huawei.com
Huawei Noah's Ark Lab
Shenzhen, China

## ABSTRACT

Sequential recommendation (SR) plays an important role in personalized recommender systems because it captures dynamic and diverse preferences from users' real-time increasing behaviors. Unlike the standard autoregressive training strategy, future data (also available during training) has been used to facilitate model training as it provides richer signals about user's current interests and can be used to improve the recommendation quality. However, these methods suffer from a severe training-inference gap, i.e., both past and future contexts are modeled by the same encoder when training, while only historical behaviors are available during inference. This discrepancy leads to potential performance degradation. To alleviate the training-inference gap, we propose a new framework DualRec, which achieves past-future disentanglement and past-future mutual enhancement by a novel dual network. Specifically, a dual network structure is exploited to model the past and future context separately. And a bi-directional knowledge transferring mechanism enhances the knowledge learnt by the dual network. Extensive experiments on four real-world datasets demonstrate the superiority of our approach over baseline methods. Besides, we demonstrate the compatibility of DualRec by instantiating using RNN, Transformer, and filter-MLP as backbones. Further empirical analysis verifies the high utility of modeling future contexts under our DualRec framework.

---
*Work done when they were research interns at Huawei Noah's Ark Lab, and both authors contributed equally to this research.
[†]Corresponding author.

## CCS CONCEPTS

• **Information systems** → **Recommender systems**.

## KEYWORDS

Sequential recommendation; training-inference gap; dual network

## 1 INTRODUCTION

Recommender systems have been widely deployed in online service platforms, ranging from online advertising and retailing [3, 7, 32] to music and video recommendation [4, 5, 30]. Generally, users' interests are dynamic and evolve over time, which are depicted by users' sequential interactions. Therefore, it leads to Sequential Recommendation (SR) that models the sequential characteristics of users' behaviors and provides more precise and customized services. To make accurate predictions, it's essential to learn effective representation for users based on the historical interactions they have engaged with. Over the years, great efforts have been devoted, and different model architectures are proposed for sequential recommendation, including Recurrent Neural Network (RNN) [10], Convolutional Neural Networks (CNN) [29], Self-Attention Network (SAN)[14], and Graph Neural Networks (GNN) [23, 33].

Typically, the sequential recommendation problem is formulated as a *next-item-prediction* problem, also known as autoregressive model (Figure 1a), which predicts the next item a user will interact with based on her historical behaviors [10, 14, 29]. It is a straightforward modeling choice for sequential data. However, in
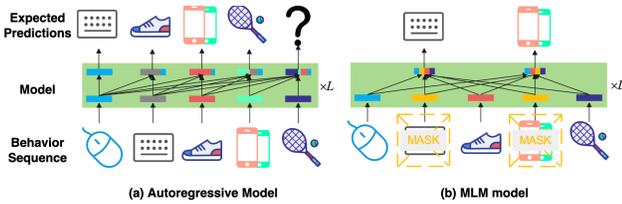
**Figure 1: Illustration of common sequential recommendation models.**

the sequential recommendation, the autoregressive schema could weaken the model's expressiveness. Because in practice, the sequential dependencies of user behaviors may not strictly hold. For example, after purchasing an iPad, a user may click on Apple pencil, iPad case, and headphones. But it is likely that the user clicked on these three products at random. Then simply modeling it in a compulsory sequential order loses some overall contextual information. Because future data (interactions that occur after the target interaction) also provide rich collaborative information to assist the model training. Therefore, it's reasonable to leverage the future data to train better sequential recommendation models.

Recently, researchers have proven that leveraging both past and future contextual information during training will significantly boost recommendation performances compared to the autoregressive models [28, 36]. For example, inspired by the advances in the field of natural language processing (NLP), BERT4Rec [28] employs a masked language model (MLM) training objective which predicts masked items based on both historical and future behavior records during training (Figure 1b). BERT4Rec significantly improves recommendation performances compared to its unidirectional autoregressive counterpart SASRec [14].

Despite the richer contextual information brought by training with future interaction data, simply adopting MLM objectives for SR can introduce a severe **training-inference gap**. Specifically, at training time, the MLM model predicts masked items with both past and future interactions as context, which can be illustrated as $P(i|\mathbf{x}_{past}, \mathbf{x}_{future})$. While at inference, only past behaviors are available available for prediction, i.e., $P(i|\mathbf{x}_{past}, \text{NULL})$. The discrepancy of context between training and inference can bias the model during inference and lead to potential performance degradation.

To exploit richer contextual information from the future while alleviating the potential training-inference gap. The following model desiderata should be met:

- **Past-future disentanglement**: The training-inference gap in existing methods is caused by the use of a single encoder predictor that entangles past and future contextual information, thus messing with inference. Instead, the future data should be modeled in a separate way without explicitly interfering with modeling historical interaction data. If both disentangled encoders get well-trained, the absence of future information will not degrade the performance of the past information encoder. By this means, we can use only past behaviors for inference, with a minimal gap between training and inference.
- **Past-future mutual enhancement**: Users' interests captured by past and future behaviors are closely related and complementary. Simply separating past and future modeling processes

hinders leveraging knowledge learned from each other. To better exploit future data, an elegant way is to have the disentangled past-future modeling process mutually enhance each other.

In this article, we propose a framework for better utilization of past and future information in sequential recommendation, named **DualRec**. To alleviate the training-inference gap, DualRec adopts a dual network structure. For a target interaction, past and future contextual behaviors are modeled by two encoders, respectively. The two encoders perform dual tasks, i.e. the past encoder performs next-item prediction (primal) while the future encoder performs previous-item-prediction (dual). Future information is decoupled from the modeling of past information in this way. During inference, only the past encoder is used to make predictions, thus avoiding the training-inference gap. Secondly, dual network enhance each other through a multi-scale knowledge transferring. Specifically, the internal representations of two networks are constraint to alignment based on the assumption that users' interests captured by past and future behaviors are closely related and complementary. Finally, as a general framework, DualRec can be instantiated using different backbone models, including RNN, Transformer and filter-based MLP.

To summarize, our contributions are as follows:

- We highlight the training-inference gap existed in sequential recommendation models when leveraging future data. To handle this problem, we propose a novel framework DualRec that achieves the disentanglement and mutual enhancement of past-future modeling.
- DualRec explicitly decouples the past information and future information modeling into two separate encoders, thus alleviating the training-inference gap, and further using a past-future knowledge transferring to learn an enhanced representation.
- We conduct comprehensive experiments on four public datasets. Experimental results demonstrate the effectiveness of our proposed DualRec as compared with several baseline models. Further analysis illustrates its compatibility.

## 2 RELATED WORK

Sequential recommenders are designed to model the sequential dynamics in user behaviors. Early efforts leverage the Markov Chain (MC) assumption [25] and model item-item transition to predict user's next action based on the last visited items. Recently, different neural network-based models have been applied, including Recurrent Neural Networks (RNNs) [20], Convolutional Neural Networks (CNNs) [17], Attention Networks [31] and Graph Neural Networks [16]. GRU4Rec [10] is a pioneering work that employs RNN to capture the dynamic characteristic of user behaviors. Hidasi and Karatzoglou further extends GRU4Rec with enhanced ranking functions as well as effective sampling strategies. Another line of research is based on CNN. Caser [29] treats the embedding matrix of items as a 2D image and models user behavior sequences with convolution. The main advantage of CNN-based models is that they are much easier to be parallelized on GPUs compared with RNN-based models. Self-Attention Mechanism and the Transformer architectures [31] are applied to sequential recommendation and proved to be advantageous in discovering user behavior patterns due to the adaptive weight learning and better integration of long-range

dependencies, such as SASRec [14], BERT4Rec [28] and S3-Rec [38]. More recently, Graph Neural Networks have been explored to encode the global structure of user interactions and capture complex item transitions. SRGNN [33] and GCSAN [35] model sequence or session as graph structure, and use graph neural networks and attention mechanism to capture the rich dependencies. Inspired by work related to self-supervised learning, S3-Rec [38] and CL4Rec [34] used a contrastive learning approach to pre-train the sequential recommendation model. Furthermore, CLEA [22] uses a contrastive learning approach to automatically filter out items from the user sequence that are more irrelevant to the target.

Among these sequential recommendation works, some bidirectional Transformer-based methods adopting MLM objectives, such as BERT4Rec[28], attempt to utilize both the past and future contextual information. However, these works suffer from a severe training-inference gap as analyzed in Section 1.

## 3 METHOD

In this section, we first define the sequential recommendation problem (Section 3.1). Then, we elaborate on the technical details of our proposed DualRec framework. The DualRec framework is shown in Figure 2 (b). We first introduce the base encoder (Section 3.2), which is a Transformer-based backbone. Then to utilize future context while alleviating the training-inference gap, we present the dual-network structure (Section 3.3) that models past and future behaviors through two encoders, respectively. Furthermore, the bi-directional information transferring mechanism (Section 3.4) is adopted to make the dual networks enhance each other. The details in model training and inference are presented in Section 3.5. Further, we discuss the complexity and compatibility of DualRec (Section 3.6). Essentially, DualRec can work in cooperation with most existing sequential recommendation models.

### 3.1 Problem Formulation

Sequential recommendation learns to predict users' next behavior from their historical behavior sequences. Given a set of users $\mathcal{U} = \{u_1, u_2, ..., u_{|\mathcal{U}|}\}$, and a set of items $\mathcal{I} = \{i_1, i_2, ..., i_{|\mathcal{I}|}\}$, let $\mathcal{S}^{(u)} = [i_1^{(u)}, i_2^{(u)}, ..., i_{T_u}^{(u)}]$ denote the chronologically sorted behavior sequence of user $u \in \mathcal{U}$, where $i_t^{(u)}$ is $t$-th interacted item of user $u$, and $T_u = |\mathcal{S}^{(u)}|$ is the length of behavior sequence. A sequential recommendation model predicts the next item $i_{T_u+1}^{(u)}$ that user $u$ will interact with based on the behavior history $\mathcal{S}^{(u)}$, which can be formulated as:

$$p(i_{T_u+1}^{(u)} = i^{(c)}|\mathcal{S}^{(u)}) = \text{SeqRecModel}(\mathcal{S}^{(u)}, i^{(c)}), \quad (1)$$

where $i^{(c)}$ is the candidate item, and SeqRecModel$(\cdot, \cdot)$ is a sequential recommendation model.

### 3.2 Base Encoder

For simplicity, we illustrate with the standard Transformer as the base encoder of the dual network structure, which is widely used in many sequential recommendation methods [14, 28, 38] and has been proven to be advantageous in discovering sequential patterns due to the adaptive weight learning. Notably, our proposed framework can also work with other backbone architectures, including RNNs

and CNNs. We will evaluate the performance of our framework with different backbones in the experimental section.

*3.2.1 Embedding Layer.* The input user behavior sequence $S^{(u)}$ is first transformed into a fixed-length sequence $s = (i_1, i_2, ..., i_n)$ (For simplicity, we omit the superscript $u$), where $n$ is the predefined maximum length. Specifically, if the original sequence length is greater than $n$, we keep the most recent $n$ actions. If the original sequence length is less than $n$, special [padding] tokens are padded to the left of the sequence as dummy past interactions.

**Item Embedding:** For all items given in $\mathcal{I}$, we create an item embedding matrix $\mathbf{E}^{\mathcal{I}} \in \mathbb{R}^{|\mathcal{I}| \times d}$ where $d$ is the embedding size. User behavior sequence $s = (i_1, i_2, ..., i_n)$ is embedded as:

$$\mathbf{X}^{(0)} = (\mathbf{e}_1, \cdots, \mathbf{e}_n), \ \mathbf{e}_k = \text{LookUp}(i_k, \mathbf{E}^{\mathcal{I}}), \quad (2)$$

where LookUp$(\cdot, \cdot)$ retrieves an item embedding from the embedding matrix, and always gets a constant zero vector for the padding item.

**Relative Positional Embedding:** Transformer doesn't contain any recurrent or convolutional operation and is not sensitive to positional information. Therefore, we need to inject additional positional embedding. We adopt the relative positional embedding as it is more robust than absolute positional embedding [27]. When calculating the attention weight of the $j$-th item given the $i$-th item as query, the relative positional embedding is calculated as

$$\mathbf{p}(i, j) = \text{LookUp}(\text{Dist}(i, j), \mathbf{E}^{\mathcal{P}}), \quad (3)$$

where $\text{Dist}(i, j) \in [-n+1, n-1]$ is the relative distance between two items. $\mathbf{E}^{\mathcal{P}} \in \mathbb{R}^{(2n-1) \times h}$ is the positional embedding where $h$ is the number of heads in the self-attention mechanism. The relative positional matrix $\mathbf{p}$ is added to the attention matrix to inject additional relative positional information.

*3.2.2 Transformer Layer.* After the embedding layer, the input $\mathbf{X} \in \mathbb{R}^{n \times d}$ are fed to multi-head self-attention (MSA) blocks for capturing the relations among different items in this sequence. The computation for each head can be formulated as follows:

$$\mathbf{head}_i = softmax\left(\frac{(\mathbf{XW}_i^Q) \cdot (\mathbf{XW}_i^K)^\top}{\sqrt{d/h}}\right)(\mathbf{XW}_i^V), \quad (4)$$

where $\mathbf{W}_i^Q \in \mathbb{R}^{d \times \frac{d}{h}}$, $\mathbf{W}_i^K \in \mathbb{R}^{d \times \frac{d}{h}}$ and $\mathbf{W}_i^V \in \mathbb{R}^{d \times \frac{d}{h}}$ are the query, key, and value projection matrices, respectively, $(\cdot)$ is the matrix multiplication operator, $i$ indicates a specific head, $h$ is the total number of heads and $\sqrt{d/h}$ is the scaling factor.

MSA performs the above self-attention operation $h$ times in parallel, then combines the outputs of each head together, and linearly projects them to get the final output:

$$\text{MSA}(\mathbf{X}) = \text{Concat}(\mathbf{head}_i, \mathbf{head}_2, ..., \mathbf{head}_h)\mathbf{W}^o, \quad (5)$$

where $\mathbf{W}^o \in \mathbb{R}^{d \times d}$ is the corresponding transformation matrix.

To introduce non-linearity and perform feature transformation between MSA layers, point-wise feed-forward (PFF) layer is used:

$$\text{PFF}(\mathbf{X}) = \text{FC}(\sigma(\text{FC}(\mathbf{X}))), \text{FC}(\mathbf{X}) = \mathbf{XW} + \mathbf{b}, \quad (6)$$

where $\mathbf{W} \in \mathbb{R}^{d \times d}$ and $\mathbf{b} \in \mathbb{R}^{1 \times d}$ are the weights and bias of a fully-connected (FC) layer, and $\sigma(\cdot)$ is the non-linear activation function such as ReLU [6].

Finally, residual connections [8] and layer normalizations [2] are employed to connect the MSA and PFF modules to get the Transformer layer:

$$\mathbf{H}^{(l)} = \text{LayerNorm}\left(\mathbf{X}^{(l-1)} + \text{MSA}(\mathbf{X}^{(l-1)})\right),$$
$$\mathbf{X}^{(l)} = \text{LayerNorm}\left(\mathbf{H}^{(l)} + \text{PFF}(\mathbf{H}^{(l)})\right), \tag{7}$$

where $\mathbf{H}^{(l)}$ is the intermediate representation at layer $l$ and $\mathbf{X}^{(l)}$ is the hidden representation at layer $l$.

## 3.3 Dual Network

To exploit future information while alleviating the potential training-inference gap, we present the dual network as shown in Figure 2(b). In the dual network model, two encoders (denoted as past encoder and future encoder, respectively) are instantiated using the base encoder, through which the past and future modeling are disentangled explicitly. For a target item, the two encoders perform dual tasks, i.e., the past encoder performs next-item-prediction (primal) while the future encoder performs previous-item-prediction (dual). In this way, the future data is modeled in a separate way without explicitly interfering with the modeling of historical interaction data. Hence no significant training-inference gap is introduced.

Next, we elaborate on the details of how we implement the dual network model, including a shared embedding layer connecting the two encoders and the dual self-attention mechanism that achieves the dual tasks.

### 3.3.1 Shared Embedding Layer.
Embeddings play a vital role in sequential recommendation and determine the effectiveness of modeling associations between items. Here, to improve the effectiveness of embedding learning, we share the parameters of the embedding layer in the past and future encoders. Hence, the embeddings are learned by receiving knowledge from both past and future.

### 3.3.2 Dual Self-Attention.
In the dual task setting, each encoder is a unidirectional autoregressive model, where the past encoder learns the past-to-future autoregressive function while the future encoder learns the opposite future-to-past autoregressive function. To this end, we employ the causal attention mask to ensure causality [14, 31]. As illustrate in Figure 2 (a), we modify the attention by disabling attention links between $\mathbf{Q}_i$ and $\mathbf{K}_j (j > i)$ for the past encoder. Likewise, we disable attention links between $\mathbf{Q}_i$ and $\mathbf{K}_j (j < i)$ for the future encoder.

## 3.4 Bi-directional Information Transferring

The dual network presented above computes representations for items with past and future information independently, which overlooks the interaction between the two encoders. However, users' preferences captured by past and future encoders are closely related and complementary. Therefore, the modeling of past and future behaviors can be mutually enhanced. Furthermore, in practice, user interests are multi-scale, as there are often both stable long-term interests and dynamic short-term interests. Therefore, how to effectively model user interest at different scales and transfer useful knowledge to assist the learning of each encoder remains a challenging problem. In this section, two encoders in the dual network are further endowed with the ability to extract users' multi-scale

interests and transfer related and complementary knowledge to enhance each other.

### 3.4.1 Multi-scale Interest Extraction.
User's interests can be divided into local interests, global interests and interests at other scales [37]. Existing methods only consider short- and long-term interests in the past behaviors [13, 40]. Here, users' multi-scale interests from both historical and future behaviors are considered. To this end, we propose a multi-scale scheme that extracts users' interest representations at different temporal scales. The multi-scale scheme can cooperate with the multi-head self-attention mechanism by adding multi-scale masks at different heads, thereby aggregating information with different receptive fields. Specifically, we design heuristic rules to assign a window size to each head based on the head id $i$ and sequence length $n$:

$$WS(i) = \begin{cases} i + 1 & \text{if } i \leq \frac{h}{2} \\ \frac{h}{2} + \left\lceil \frac{\exp(i - \frac{h}{2})}{\exp \frac{h}{2}} \cdot \left(n - \frac{h}{2}\right) \right\rceil & \text{if } i > \frac{h}{2} \end{cases}, \quad i = 1 \cdots h \tag{8}$$

where $WS(i)$ is the window size assign function for head $i$. To comprehensively capture users' interests at multi-scale level, we set $h/2$ heads with linearly increasing local windows and $h/2$ heads with exponentially increasing global windows, respectively.

For a specific head with a window size of $\sigma$, we rewrite the multi-head self-attention mechanism to adapt to the multi-scale scheme:

$$head_{i,j}(\mathbf{X}, \sigma) = softmax\left(\frac{(\mathbf{X}\mathbf{W}_i^Q)_j \cdot S_j\left(\mathbf{X}\mathbf{W}_i^K, \sigma\right)^{\top}}{\sqrt{d/h}}\right) \cdot S_j\left(\mathbf{X}\mathbf{W}_i^V, \sigma\right),$$
$$S_j(\mathbf{X}, \sigma) = \left[\mathbf{X}_{j-\sigma}, \cdots, \mathbf{X}_j\right], \tag{9}$$

where $head_{i,j}$ indicates the $i$-th attention head vector in $j$-th position, and $\mathbf{X}_j$ means the $j$-th vector in the matrix $\mathbf{X}$. The function $S_j(\mathbf{X}, \sigma)$ extracts the context of $\mathbf{X}_j$ in the range $\sigma$, where $\sigma > 0$ in past encoders and $\sigma < 0$ in future encoders. In Figure 3, an illustration of the multi-scale self-attention mechanism is shown for $\sigma = 2, 3, 4$.

In the subsequent compatibility analysis, for the backbone of the non-transformer architecture, we used multi-scale input to implement multi-scale interest extraction.

### 3.4.2 Interest-level Knowledge Transferring.
After extracting users' multi-scale interests from both past and future behaviors, we perform past-future knowledge transferring to make the two encoders enhance each other. We borrow the idea of knowledge distillation [12, 26] and introduce a regularization term that encourages the representation distributions of two encoders to be consistent with each other. In this way, the related and complementary user interests captured by the two encoders are enhanced.

The goal is to regularize the model by minimizing the bidirectional Kullback-Leibler (KL) divergence between the past and future encoders' output distributions at multi-scale (Figure 3). The regularization loss is calculated as:
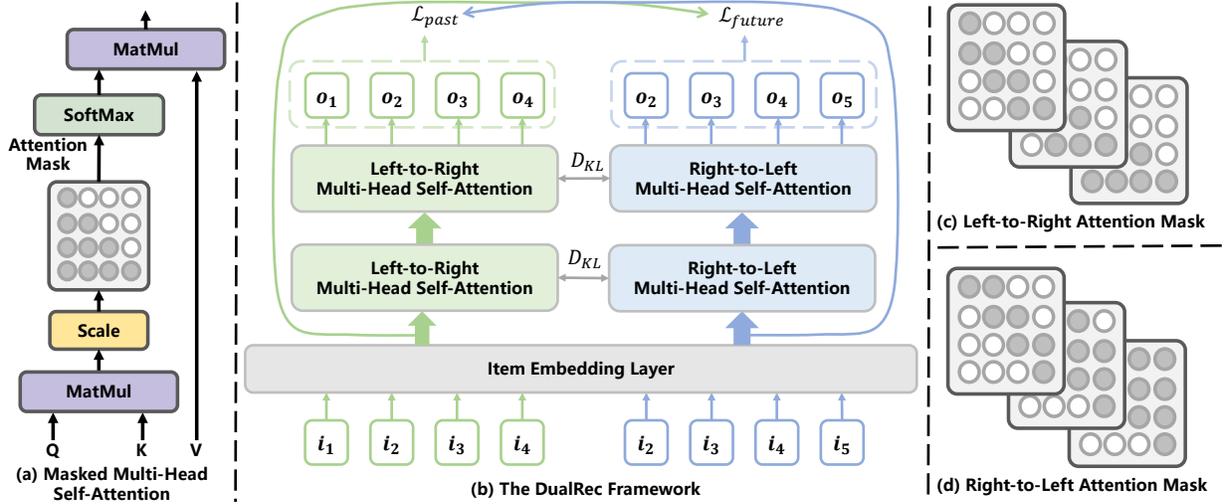
**Figure 2: Model architecture of the proposed DualRec framework. (a) Illustration of masked multi-head self-attention. The shaded nodes are visible. (b) Overall structure of the dual network, with the past encoder on the left and the future encoder on the right. Past and future encoders are associated by the shared embedding layer and bi-directional information transferring using KL divergences. (c) and (d) are the attention masks from left to right in the past encoder and from right to left in the future encoder, respectively, and the time windows corresponding to each attention head are different, i.e., multi-scale.**

$$\mathcal{L}_{reg} = \sum_{i=1}^{h} \frac{1}{2} \left( D_{KL}(\mathbf{head}_i^p \| \mathbf{head}_i^f) + D_{KL}(\mathbf{head}_i^f \| \mathbf{head}_i^p) \right), \tag{10}$$

where $\mathbf{head}_i^p$ and $\mathbf{head}_i^f$ are output distributions at head $i$ of the past encoder and the future encoder, respectively. And $D_{KL}(p\|q)$ is the KL divergence between two distributions:

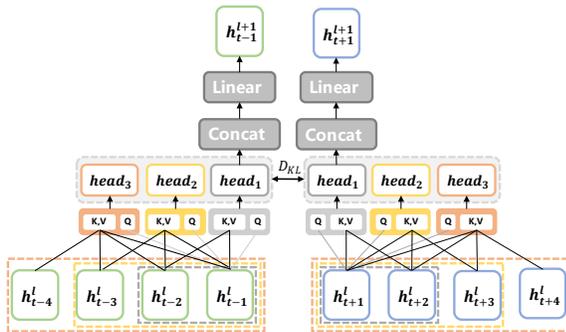$$D_{KL}(p\|q) = \sum_i p_i \log \frac{p_i}{q_i}, \tag{11}$$



**Figure 3: Illustration of interest-level knowledge transferring between past and future information.**

## 3.5 Model Training and Inference

*3.5.1 Model Training.* The two encoders in the dual network extract information of previously consumed items and future consumed items, respectively. Correspondingly, when make prediction for the $t$-th behavior, we use the output $\mathbf{o}_{t-1}^p$ of the past encoder

and $\mathbf{o}_{t+1}^f$ of the future encoder. Specifically, we estimate the probability of each candidate item according to the inner product based similarity:

$$\mathbf{s}_t^p = \mathbf{o}_{t-1}^p \cdot \mathbf{E}^{\mathcal{I}\top}, \quad \mathbf{s}_t^f = \mathbf{o}_{t+1}^f \cdot \mathbf{E}^{\mathcal{I}\top}, \tag{12}$$

where $\mathbf{E}^{\mathcal{I}} \in \mathbb{R}^{|\mathcal{I}| \times d}$ is the embedding matrix for all candidate items which reuses the input item embedding matrix to alleviate overfitting and reduce the model size. Then we apply a softmax function to get the output distribution over candidate items:

$$\hat{\mathbf{y}}_t^p = softmax(\mathbf{s}_t^p), \quad \hat{\mathbf{y}}_t^f = softmax(\mathbf{s}_t^f), \tag{13}$$

where $\hat{\mathbf{y}}_t^p, \hat{\mathbf{y}}_t^f \in \mathbb{R}^{|\mathcal{I}|}$ denotes the estimated probabilities of each item to appear at position $t$ using past and future behavior information, respectively.

We employ the cross-entropy function to evaluate the prediction loss. The final loss function is a combination of cross-entropy loss of two encoders as well as the regularization term:

$$\mathcal{L} = \alpha \mathcal{L}_{past} + (1 - \alpha) \mathcal{L}_{future} + \beta \mathcal{L}_{reg}$$
$$= -\alpha \sum_{t=2}^{n-1} \text{OneHot}(i_t^\star) \log \hat{\mathbf{y}}_t^p - (1 - \alpha) \sum_{t=2}^{n-1} \text{OneHot}(i_t^\star) \log \hat{\mathbf{y}}_t^f + \beta \mathcal{L}_{reg}, \tag{14}$$

where $i_t^\star$ is the ground truth item appears at position $t$, $\alpha \in [0, 1]$ and $\beta > 0$ are two hyperparameters that control the contribution of dual tasks and the regularization term, respectively.

*3.5.2 Model Inference.* During inference, only the past encoder is used to make predictions based on users' historical interactions. Since the input context of the past encoder is consistent between training and inference. This will alleviate the training-inference gap problem in existing methods.

## 3.6 Model Discussion

*3.6.1 Complexity Analysis.* This subsection analyzes the time and space complexity of our proposed dual network and bi-directional information transferring. The training time and space complexity of the dual network is determined by the backbone network used, i.e., twice the time and space complexity of the backbone. While the inference time complexity of the dual network is the same as that of the backbone. For the Transformer-based backbone we use, both time and space complexity per layer of the dual network is $O(n^2 d)$. Moreover, the time complexity of bi-directional information transferring is $O(h)$ as the multi-scale interest number is $h$; while its space complexity is $O(1)$ because it's a parameter-free operation. Since $h << n^2$, the time complexity of bi-directional information transferring can be ignored. So the total time complexity of our proposed DualRec is approximate twice the complexity of the backbone, which is $O(n^2 d)$ for our Transformer-based backbone.

*3.6.2 Compatibility Analysis.* The two core components of our proposed DualRec, namely the dual network and bi-directional information transferring, can be applied seamlessly to other sequential recommendation backbones, such as GRU4Rec, SASRec, FMLP-Rec and *etc.* The dual network ensures past-future disentanglement while introducing future information in the training process; while bi-directional information transferring enables the mutual enhancement of past and future knowledge based on the dual network. These two components are generalized well to the sequential recommendation methods, which is demonstrated by the extensive compatibility analysis experiments conducted in Section 4.4. Notably, for the non-self-attention methods, multi-scale interest is captured by the multi-scale inputs.

## 4 EXPERIMENTS

We conduct extensive experiments on four real-world datasets to answer the following research questions:

- **(RQ1)** How does DualRec perform compared with state-of-the-art SR models?
- **(RQ2)** How do different components affect the performance of DualRec respectively?
- **(RQ3)** Can DualRec be compatible with mainstream SR models?
- **(RQ4)** How do different hyperparameters affect DualRec?
- **(RQ5)** How is future information used in DualRec?

## 4.1 Experimental Setting

*4.1.1 Datasets.* To comprehensively investigate the performance of DualRec, we conduct experiments on four large-scale real-world recommendation datasets:

- **Amazon Beauty, Sports, and Toys**[1]: this series of datasets was crawled from relevant product reviews on Amazon.com [19], which are widely used to evaluate recommendation performance [14, 38, 39]. We select three categories: "Beauty", "Sports and Outdoors", and "Toys and Games" for our experiments.
- **Yelp**[2]: this is a dataset for the business recommendation, and we only use the user transaction records after *January 1st, 2019* for our experiments.

---

[1]http://jmcauley.ucsd.edu/data/amazon/
[2]https://www.yelp.com/dataset

For a fair comparison, we strictly follow the dataset settings of FMLP-Rec [39]. We use the same pre-processed datasets and negative samples with FMLP-Rec. Concretely, for all datasets, the user transaction records are grouped by userIDs and sorted by the transaction timestamps ascendingly to form the user behavioral sequence. Moreover, the inactive users and unpopular items that have only fewer than five transaction records are filtered, following [11, 25]. Table 1 summarizes the statistics of the four datasets.

**Table 1: Dataset Statistics**

| Dataset | # Users | # Items | # Interactions | Density |
|---------|---------|---------|----------------|---------|
| Beauty | 22,363 | 12,101 | 198,502 | 0.07% |
| Sports | 25,598 | 18,357 | 296,337 | 0.05% |
| Toys | 19,412 | 11,924 | 167,597 | 0.07% |
| Yelp | 30,431 | 20,033 | 316,354 | 0.05% |

*4.1.2 Evaluation Metrics.* We evaluate models in terms of the top-K recommendation performance with three metrics, i.e., the Hit Ratio (HR@K), the Normalized Discounted Cumulative Gain (NDCG@K), and Mean Reciprocal Rank (MRR), which are widely used in related works [14, 28, 39]. HR@K is a recall-based metric that measures the average proportion of right items in the top-K recommendation lists. NDCG@K evaluates the ranking quality of the top-K recommendation lists in a position-wise manner. Specifically, we report the evaluation results on HR@{1,5,10}, NDCG@{5,10} and MRR, where we omit the NDCG@1 metric as it is equivalent to HR@1. To make a fair comparison, we use the same negative samples as the FMLP-Rec [39]. For any ground-truth item, the 99 negative samples are selected randomly from the items that the user has not interacted with.

*4.1.3 Comparison Baselines.* To comprehensively demonstrate the effectiveness of the proposed DualRec framework, we compare it with eleven state-of-the-art recommendation baselines from different lines of research as listed below:

- Sequential recommendation models. (1) **Caser** [29] captures short-term dynamic patterns of user activity through convolution. (2) **GRU4Rec** [10] uses the GRU to model sequences of user behavior. (3) **HGN** [18] captures long-term and short-term user interest through a hierarchical gating mechanism. (4) **RepeatNet** [24] is based on an encoder-decoder architecture to select items from a user's transaction history records for repeat recommendations. (5) **SASRec** [14] uses a causal transformer-based structure to predict the next item with the multi-head self-attention mechanism. (6) **BERT4Rec** [28] uses a bidirectional transformer model to train through a masked item prediction task. (7) **FMLP-Rec** [39] is a all-MLP model that replaces the self-attention mechanism in the transformer structure with a learnable filter-enhanced block.
- Graph-based models. (8) **SRGNN** [33] uses graph attention networks to model session data as a session graph. (9) **GCSAN** [35] combines graph neural networks with the self-attention mechanism for session recommendation.

**Table 2: Performance comparison using different methods on four popular sequential datasets. The best performance and the second-best performance methods are denoted in bold and underlined fonts respectively. The "*" mark denotes the statistical significance (p < 0.05) of comparing DualRec with the strongest baseline results and the "Improv." column represents the relative improvement of DualRec over the strongest baseline.**

| Datasets | Metric | GRU4Rec | Caser | HGN | RepeatNet | CLEA | SASRec | S3-Rec$_{MIP}$ | BERT4Rec | SRGNN | GCSAN | FMLP-Rec | DualRec | Improv. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Beauty | HR@1 | 0.1519 | 0.1337 | 0.1683 | 0.1578 | 0.1325 | 0.1907 | 0.1678 | 0.1531 | 0.1729 | 0.1973 | 0.2051 | **0.2289**$^*$ | +11.60% |
| | HR@5 | 0.3612 | 0.3032 | 0.3544 | 0.3268 | 0.3305 | 0.4036 | 0.3710 | 0.3640 | 0.3518 | 0.3678 | 0.4103 | **0.4241**$^*$ | +3.24% |
| | NDCG5 | 0.2608 | 0.2219 | 0.2656 | 0.2455 | 0.2353 | 0.3022 | 0.2735 | 0.2622 | 0.2660 | 0.2864 | 0.3133 | **0.3320**$^*$ | +5.97% |
| | HR@10 | 0.4657 | 0.3942 | 0.4503 | 0.4205 | 0.4426 | 0.5043 | 0.4749 | 0.4739 | 0.4484 | 0.4542 | 0.5070 | **0.5190**$^*$ | +2.37% |
| | NDCG@10 | 0.2944 | 0.2512 | 0.2965 | 0.2757 | 0.2715 | 0.3358 | 0.3069 | 0.2975 | 0.2971 | 0.3143 | 0.3443 | **0.3626**$^*$ | +5.32% |
| | MRR | 0.2593 | 0.2263 | 0.2669 | 0.2498 | 0.2376 | 0.2990 | 0.2731 | 0.2614 | 0.2686 | 0.2882 | 0.3102 | **0.3302**$^*$ | +6.45% |
| Sports | HR@1 | 0.1366 | 0.1135 | 0.1428 | 0.1334 | 0.1114 | 0.1676 | 0.1107 | 0.1255 | 0.1419 | 0.1669 | 0.1722 | **0.1947**$^*$ | +13.07% |
| | HR@5 | 0.3552 | 0.2866 | 0.3349 | 0.3162 | 0.3041 | 0.3919 | 0.3141 | 0.3375 | 0.3367 | 0.3588 | 0.3886 | **0.4127**$^*$ | +5.31% |
| | NDCG5 | 0.2487 | 0.2020 | 0.2420 | 0.2274 | 0.2096 | 0.2823 | 0.2143 | 0.2341 | 0.2418 | 0.2658 | 0.2839 | **0.3080**$^*$ | +8.49% |
| | HR@10 | 0.4853 | 0.4014 | 0.4551 | 0.4324 | 0.4274 | 0.5169 | 0.4491 | 0.4772 | 0.4545 | 0.4737 | 0.5098 | **0.5383**$^*$ | +4.14% |
| | NDCG@10 | 0.2907 | 0.2390 | 0.2806 | 0.2649 | 0.2493 | 0.3244 | 0.2578 | 0.2775 | 0.2799 | 0.3029 | 0.3231 | **0.3485**$^*$ | +7.43% |
| | MRR | 0.2493 | 0.2100 | 0.2469 | 0.2334 | 0.2156 | 0.2838 | 0.2203 | 0.2378 | 0.2461 | 0.2691 | 0.2830 | **0.3078**$^*$ | +8.47% |
| Toys | HR@1 | 0.1303 | 0.1114 | 0.1504 | 0.1333 | 0.1104 | 0.1760 | 0.1825 | 0.1262 | 0.1600 | 0.1996 | 0.2003 | **0.2268**$^*$ | +13.23% |
| | HR@5 | 0.3526 | 0.2614 | 0.3276 | 0.3001 | 0.3055 | 0.3975 | 0.3892 | 0.3344 | 0.3389 | 0.3613 | 0.4010 | **0.4152**$^*$ | +3.54% |
| | NDCG5 | 0.2444 | 0.1885 | 0.2423 | 0.2192 | 0.2102 | 0.2907 | 0.2903 | 0.2327 | 0.2528 | 0.2836 | 0.3055 | **0.3253**$^*$ | +6.48% |
| | HR@10 | 0.4691 | 0.3540 | 0.4211 | 0.4015 | 0.4207 | 0.5034 | 0.4935 | 0.4493 | 0.4413 | 0.4509 | 0.4977 | **0.5145**$^*$ | +2.21% |
| | NDCG@10 | 0.2820 | 0.2183 | 0.2724 | 0.2517 | 0.2473 | 0.3271 | 0.3239 | 0.2698 | 0.2857 | 0.3125 | 0.3367 | **0.3573**$^*$ | +6.12% |
| | MRR | 0.2424 | 0.1967 | 0.2454 | 0.2253 | 0.2138 | 0.2877 | 0.2890 | 0.2338 | 0.2566 | 0.2871 | 0.3034 | **0.3256**$^*$ | +7.32% |
| Yelp | HR@1 | 0.1970 | 0.2188 | 0.2428 | 0.2341 | 0.2102 | 0.2327 | 0.2250 | 0.2405 | 0.2176 | 0.2493 | 0.2625 | **0.2893**$^*$ | +10.21% |
| | HR@5 | 0.5788 | 0.5111 | 0.5768 | 0.5357 | 0.5707 | 0.5949 | 0.5978 | 0.5976 | 0.5442 | 0.5725 | 0.6246 | **0.6328**$^*$ | +1.32% |
| | NDCG5 | 0.3933 | 0.3696 | 0.4162 | 0.3894 | 0.3955 | 0.4198 | 0.4171 | 0.4252 | 0.3860 | 0.4162 | 0.4507 | **0.4681**$^*$ | +3.93% |
| | HR@10 | 0.5788 | 0.6661 | 0.7411 | 0.6897 | 0.7473 | 0.7722 | 0.7764 | 0.7597 | 0.7096 | 0.7371 | 0.7699 | **0.7896**$^*$ | +2.25% |
| | NDCG@10 | 0.4511 | 0.4198 | 0.4695 | 0.4393 | 0.4527 | 0.4790 | 0.4751 | 0.4778 | 0.4395 | 0.4696 | 0.4981 | **0.5190**$^*$ | +4.20% |
| | MRR | 0.3684 | 0.3595 | 0.3998 | 0.3769 | 0.3751 | 0.3994 | 0.3938 | 0.4026 | 0.3711 | 0.4006 | 0.4236 | **0.4466**$^*$ | +5.43% |

- Contrastive learning augmented models. (10) **S3-Rec** [38] designed four pretraining tasks based on maximum mutual information to pretrain the self-attention-based sequential recommendation model. We use only the Masked Item Prediction tasks without using attribute information for pretraining in our work. (11) **CLEA** [22] uses a contrastive learning approach to automatically extract items relevant to the target item to denoise the user behavior sequence.

*4.1.4 Implementation Details.* We implement GRU4Rec, SASRec, S3-Rec, and FMLP-Rec using PyTorch [21], and set hyperparameters according to the suggested values of their original work. For other methods (Caser, HGN, RepeatNet, CLEA, BERT4Rec, SRGNN and GCSAN), we follow the results of the reimplementations in FMLP-Rec [39] and S3-Rec [38], as the datasets and evaluation metrics used in these works are strictly consistent with ours.

Our proposed DualRec is also implemented in PyTorch [21]. For a fair comparison, we set the maximum sequence length $n = 50$ and the dimensionality of the item embedding $d = 64$, which is consistent with [38, 39]. There are 2 Transformer layers and 8 heads per layer. Dropout is used to regularize the training process, and the dropout ratio is set to 0.5 for each dataset. We train with Adam optimizer [15] with a learning rate of 0.001 and a weight decay of 1e-4. And the batch size is set as 256 for training, validation, and testing.

## 4.2 Overall Performances (RQ1)

The overall performance of the proposed DualRec model and baselines are presented in Table 2. Based on the results, we summarize the following observations:

- First, the Transformer-based methods (i.e., SASRec, S3-Rec and BERT4Rec) and the GNN-based methods (i.e., SRGNN and GCSAN) achieve better results than RNN-based and CNN-based models generally. For Transformer-based methods, it can be attributed to the self-attention mechanism that better captures long-range dependencies in user interaction sequences. For GNN-based methods, the great performance is because GNNs can better capture complex interactions of items than the conventional sequential recommendation method.

- Second, the methods using the masked item prediction (MIP) task for training (i.e., BERT4Rec) or pretraining (i.e., S3-Rec) do not yield better results than SASRec. These works are improvements on SASRec with the addition of MIP tasks to introduce future information, which should result in better performance than SASRec. But the strong training-inference gap existing in MIP tasks (as described in Section 1) severely affects performance gains and may even cause performance degradation.

- Finally, DualRec consistently achieves the best performances on all four datasets under all six evaluation metrics, demonstrating the superiority of our proposed method. It gains 12.03% HR@1, 2.74% HR@10, 5.77% NDCG@10, and 6.92% MRR improvements

**Table 3: Ablation study results of different consisting components in DualRec.**

| Methods | Beauty | | Sports | | Toys | | Yelp | |
|---|---|---|---|---|---|---|---|---|
| | HR@5 | NDCG@5 | HR@5 | NDCG@5 | HR@5 | NDCG@5 | HR@5 | NDCG@5 |
| DualRec | **0.4241** | **0.3320** | **0.4127** | **0.3080** | **0.4152** | **0.3253** | **0.6328** | **0.4681** |
| DualRec w/o DE | 0.4154 | 0.3251 | 0.4037 | 0.3016 | 0.4098 | 0.3202 | 0.6273 | 0.4610 |
| DualRec w/o BIT | 0.4235 | 0.3311 | 0.4102 | 0.3055 | 0.4122 | 0.3246 | 0.6284 | 0.4658 |
| DualRec w/o RPE | 0.4216 | 0.3306 | 0.4093 | 0.3044 | 0.413 | 0.3234 | 0.6320 | 0.4677 |
| DualRec w/o LN | 0.4086 | 0.3075 | 0.4097 | 0.3031 | 0.3987 | 0.2963 | 0.6321 | 0.4653 |
| DualRec w/o RC | 0.3653 | 0.2642 | 0.3720 | 0.2632 | 0.3388 | 0.2360 | 0.5751 | 0.4052 |

**Table 4: Compatibility Analysis with Different Backbones on four sequential datasets. Prefix "Dual" indicates the corresponding dual network model of backbone, and "BIT" means bi-directional information transferring.**

| Datasets Model | Beauty | | | Sports | | | Toys | | | Yelp | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | HR@5 | NDCG@5 | MRR | HR@5 | NDCG@5 | MRR | HR@5 | NDCG@5 | MRR | HR@5 | NDCG@5 | MRR |
| SASRec | 0.4036 | 0.3022 | 0.2990 | 0.3919 | 0.2823 | 0.2838 | 0.3975 | 0.2907 | 0.2877 | 0.5949 | 0.4198 | 0.3994 |
| DualSASRec | 0.4178 | 0.3147 | 0.3108 | 0.4055 | 0.2946 | 0.2936 | 0.4068 | 0.3038 | 0.3009 | 0.6198 | 0.4415 | 0.4183 |
| DualSASRec+BIT | **0.4223** | **0.3199** | **0.3153** | **0.4111** | **0.2985** | **0.2961** | **0.4111** | **0.3084** | **0.3050** | **0.6199** | **0.4457** | **0.4239** |
| GRU4Rec | 0.3612 | 0.2608 | 0.2593 | 0.3552 | 0.2487 | 0.2493 | 0.3526 | 0.2444 | 0.2424 | 0.5788 | 0.3933 | 0.3684 |
| DualGRU4Rec | 0.3779 | 0.2726 | 0.2696 | 0.3681 | 0.2578 | 0.2575 | 0.3532 | 0.2454 | 0.2443 | 0.5869 | 0.4036 | 0.3803 |
| DualGRU4Rec+BIT | **0.3866** | **0.2826** | **0.2798** | **0.3768** | **0.2638** | **0.2624** | **0.3679** | **0.2567** | **0.2535** | **0.5988** | **0.4141** | **0.3888** |
| FMLP-Rec | 0.4103 | 0.3133 | 0.3102 | 0.3886 | 0.2839 | 0.2830 | 0.4010 | 0.3055 | 0.3034 | 0.6246 | 0.4507 | 0.4236 |
| DualFMLP-Rec | 0.4172 | 0.3200 | 0.3170 | 0.3937 | 0.2889 | 0.2878 | 0.4041 | 0.3084 | 0.3063 | **0.6351** | **0.4708** | **0.4454** |
| DualFMLP-Rec+BIT | **0.4210** | **0.3240** | **0.3208** | **0.4002** | **0.2940** | **0.2914** | **0.4067** | **0.3109** | **0.3087** | 0.6343 | 0.4668 | 0.4407 |

on average against the most competitive baselines. The excellent performance of DualRec demonstrates the great utility of introducing future information into the training process. The dual network structure in DualRec effectively avoids the training-inference gap in the MIP task by disentangling past and future information. In Section 4.4, further extensive compatibility analysis experiments on other backbones are carried out to verify the high utility of our proposed framework for modeling future contexts.

## 4.3 Ablation Study (RQ2)

As introduced in Section 3, the backbone architecture of DualRec is constructed with two main components, i.e., the dual encoders (DE) and the bi-directional information transferring (BIT) component. To verify the effectiveness of both components, we conduct the ablation study by removing either component from the DualRec architecture, which results in two sub-models:

- *DualRec w/o DE*: Without dual encoders (DE) means, the future encoder is removed. So we utilize the past and future information to generate the corresponding representations on the same encoder with the left-to-right and right-to-right attention masks, respectively. Then the past and future representations generated by the same encoder perform bidirectional information transferring between themselves.
- *DualRec w/o BIT*: Without bi-directional information transferring, the shared embedding layer becomes the only associated component between the two encoders in the dual network.

Moreover, we also discuss other relevant components related to Transformer-based backbone structures, including relative positional embedding (*DualRec w/o RPE*), layer normalization (*DualRec w/o LN*), and the residual connection (*DualRec w/o RC*).

The comparative results between DualRec and the sub-models are shown in Table 3, where only the HR@10 and NDCG@10 scores are reported due to space limitation, from which we draw the following conclusions:

- It can be observed that removing either component leads to performance degradation, proving that all components of the proposed model are effective and necessary. Both the introduction of future information and the bi-directional information transferring positively contribute to the performance. What is more, the removal of residual connections in backbones (*DualRec w/o RC*) leads to a network degradation problem that severely impairs the performance.
- The dual network model plays a more critical role in introducing future information into the training process, with bi-directional information transferring playing a relatively supportive role.

## 4.4 Compatibility Analysis (RQ3)

DualRec is not only a specific model, but also a general framework that can be applied to most existing SR models. To verify our claim, we replace the backbone with other representative SR models and evaluate the resulting models by comparing them with the original backbones. The chosen SR models include Transformer-based SASRec, RNN-based GRU4Rec and Filter-based FMLP-Rec, and the comparison results are shown in Table 4.

As we can see, for each backbone model, using the dual network to introduce future information into the training process achieves

a consistent performance improvement on all four datasets. This performance improvement is due to the shared embedding layer learning knowledge from both past and future behaviors, while past-future disentanglement avoids severe training-inference gaps.

Based on the dual network, further enhancing the interaction between the past and future information using bi-directional information transferring can also facilitate improvement on the next item prediction training task. It is because information transferring between past and future promotes mutual learning and useful knowledge enhancement between the two encoders, thus improving the performance of the method. Furthermore, it is obvious that bi-directional knowledge transferring on DualFMLP-Rec causes performance degradation. This may be caused by the loss of original information along with the exchange of knowledge between the past and the future, and therefore the extent of knowledge transferring should not be too high which is also demonstrated by Figure 7.

The above results demonstrate the broad generality and compatibility of DualRec on different backbone architectures, which can provide significant performance improvements to the backbone model.
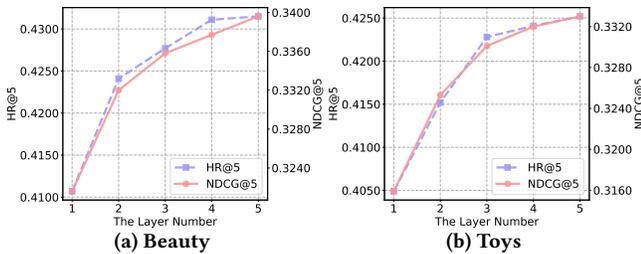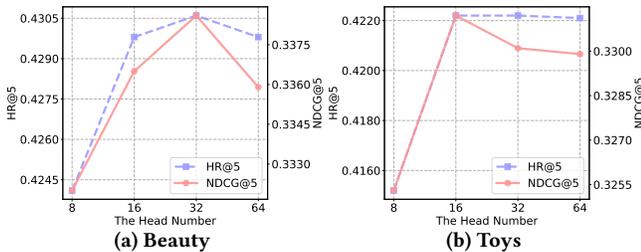


**Figure 4: Performances with different layer number.**



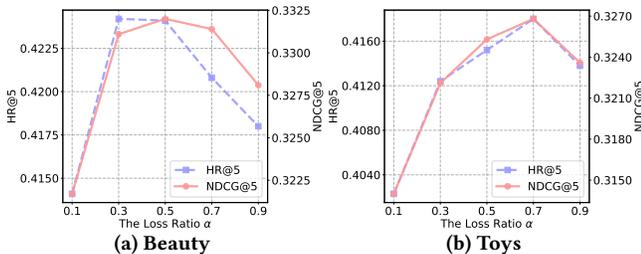**Figure 5: Performances with different attention head number.**



**Figure 6: Performances with different loss ratio of dual network model.**
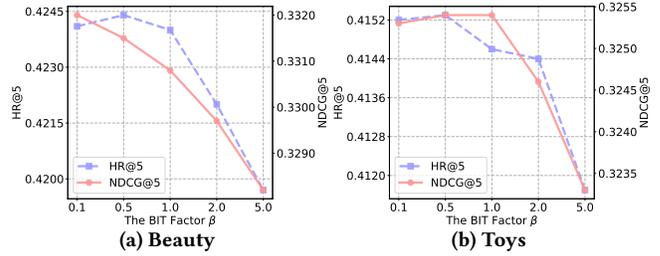


**Figure 7: Performances with different factor of bi-directional information transferring loss.**

## 4.5 Hyperparameter Analysis (RQ4)

To investigate the impact of different hyperparameters on DualRec, we perform experiments on Beauty and Toys with different configurations of key hyperparameters, including (1) the number of layers $L$, (2) the number of self-attention heads $h$, (3) the loss ratio $\alpha$ of the dual network, and (4) the regularization factor $\beta$ of bi-directional information transferring. We keep all other hyperparameters fixed when investigating the hyperparameter of interest.

- **The number of layers $L$ and the number of attention heads $h$.** The number of layers and heads together determine the capacity of the base encoder, thus affecting the performance of the sequential representation learning. Figure 4 and 5 show the performance with different number of layers $L$ and attention heads $h$, respectively. As $L$ or $h$ increases, i.e., the capacity of the model increases, the performance of the model also improves. Nevertheless, as they grow to a certain point, model performance tends to stabilize. The reason is that model with a large capacity tends to overfit noise in data, while model with a small capacity is not sufficient to capture user interest well.

- **The loss ratio $\alpha$ of the dual network.** The loss ratio of the dual network defines the weight of the prediction task loss for the past encoder and the future encoder. It controls the ratio of past and future information utilized in the dual network model. Overall, the best results are obtained when the losses in the two encoders of the dual network model are relatively balanced, i.e., when the loss ratio is around 0.5, as shown in Figure 6. Because either too small or too large $\alpha$ will cause the past or future encoder to be undertrained. Therefore, keeping the weights of training loss approximately equal between the two encoders is beneficial for stable training.

- **The factor of bi-directional information transferring loss $\beta$.** The factor $\beta$ determines the extent of bi-directional information transferring, i.e., the coefficient of the regular term. Figure 7 shows that the factor $\beta$ should not be set too high, which would affect the performance of the model. This performance degradation may result from excessive $\beta$ leading to loss of original information during bidirectional information transferring. At the same time, if the factor $\beta$ is set too small, knowledge transfer will not work effectively, so $\beta$ can be set to a moderate value, such as 0.5 in the Figure 7.

## 4.6 Future Information Utilization (RQ5)

We delve deep into how future information is used in DualRec. First, as sequential dependencies of user behaviors may not be strictly held, we can train SR models in different ways, including (1)

**Table 5: The performance of Past-Only Model, Future-Only Model, and DualRec on four datasets.**

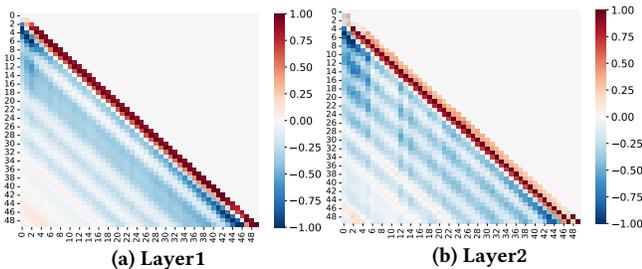| Datasets | Metric | Past-Only | Future-Only | DualRec |
|---|---|---|---|---|
| Beauty | HR@5 | 0.4166 | 0.3833 | **0.4235** |
| | NDCG@5 | 0.3272 | 0.2912 | **0.3311** |
| Sports | HR@5 | 0.4061 | 0.3786 | **0.4102** |
| | NDCG@5 | 0.3027 | 0.2766 | **0.3055** |
| Toys | HR@5 | 0.4085 | 0.3601 | **0.4122** |
| | NDCG@5 | 0.3212 | 0.2761 | **0.3246** |
| Yelp | HR@5 | 0.6276 | 0.6215 | **0.6284** |
| | NDCG@5 | 0.4628 | 0.4576 | **0.4658** |



**(a) Layer1**  **(b) Layer2**

**Figure 8: Heat maps of the average difference in attention score per layer between the past encoder in the DualRec and the Past-Only Model on the Yelp test dataset (red means the average attention scores in the DualRec are higher than those in the Past-Only Model; and blue indicates the opposite case). The coordinates in the figure indicate the sequence index.**

**Past-Only Model**: standard next-item-prediction with only past behaviors, (2) **Future-Only Model**: previous-item prediction with only future behaviors, and (3) **DualRec**: dual tasks with both past and future behaviors. But when testing, we follow the standard setup of predicting users' next interactions based on their historical interactions. Table 5 shows the results of these three settings. Not surprisingly, due to the extreme case of training-inference gap, training with only future data performs worse than training with past data. But it is acceptable, for example, the performance only drops by 8.0% compared to using only past information in terms of HR@5 on the Beauty dataset (0.3833 and 0.4166, respectively). It suggests that the knowledge learned in the future data is beneficial for the next-item-prediction task. And this is why combining past and future data yields better results.

We further explore the impact of the future information. We compare the past encoder in DualRec (DualRec-Past for simplicity) and the Past-Only Model, since they are structurally identical and differ only in training settings (past encoder in DualRec can leverage future information). Specifically, we calculate the average difference in attention scores between the two models on the Yelp testing dataset. The heat maps are shown in Figure 8, where red means the average attention scores in DualRec-Past are higher than those in the Past-Only Model, and blue indicates the opposite case. Noticed that we set the number of attention head as 1 for simplicity. From Figure 8, the attention scores of DualRec-Past are significantly higher around the diagonal than those of Past-Only

Model (the diagonal line of the heat map appears red). That can attribute to the dual tasks introducing future information, where the self-attention mechanism focuses more on items that describe users' current preference, i.e., short-term items. The widely recognized Markov property [25] in sequential recommendation indicates that short-range history plays a more important role than long-range history in capturing users' real-time interest. So with the help of future information, DualRec-Past can better focus on the more valuable short-range history, resulting in significant performance improvements.

## 5 CONCLUSIONS

In this paper, we present the DualRec framework to introduce future data into the training process while alleviating the potential training-inference gap. Specifically, a dual network is proposed to achieve past-future disentanglement, hence avoiding training-inference gap. Further, bi-directional information transferring is devised to make better use of both past and future context knowledge. Extensive experiments on four real-world datasets demonstrate the superior effectiveness and wide compatibility of the DualRec framework. In future work, we will further explore adopting our DualRec framework to the multi-behavior sequential recommendation.

## ACKNOWLEDGMENTS

## REFERENCES

[1] 2020. MindSpore. https://www.mindspore.cn
[2] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450* (2016).
[3] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, Rohan Anil, Zakaria Haque, Lichan Hong, Vihan Jain, Xiaobing Liu, and Hemal Shah. 2016. Wide & Deep Learning for Recommender Systems. In *Proc. Workshop Deep Learning for Recommender Systems*.
[4] James Davidson, Benjamin Liebald, Junning Liu, Palash Nandy, Taylor Van Vleet, Ullas Gargi, Sujoy Gupta, Yu He, Mike Lambert, Blake Livingston, et al. 2010. The YouTube video recommendation system. In *Proceedings of the fourth ACM conference on Recommender systems*. 293–296.
[5] Yashar Deldjoo, Mehdi Elahi, Paolo Cremonesi, Franca Garzotto, Pietro Piazzolla, and Massimo Quadrana. 2016. Content-based video recommendation system based on stylistic visual features. *Journal on Data Semantics* 5, 2 (2016), 99–113.
[6] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 315–323.
[7] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: A Factorization-Machine based Neural Network for CTR Prediction. In *IJCAI*.
[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.
[9] Balázs Hidasi and Alexandros Karatzoglou. 2018. Recurrent neural networks with top-k gains for session-based recommendations. In *Proceedings of the 27th ACM international conference on information and knowledge management*. 843–852.
[10] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2015. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939* (2015).
[11] Balázs Hidasi, Massimo Quadrana, Alexandros Karatzoglou, and Domonkos Tikk. 2016. Parallel Recurrent Neural Network Architectures for Feature-rich Session-based Recommendations. In *Acm Conference on Recommender Systems*.

[12] Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531* 2, 7 (2015).

[13] Linmei Hu, Chen Li, Chuan Shi, Cheng Yang, and Chao Shao. 2020. Graph neural news recommendation with long-term and short-term interest modeling. *Information Processing & Management* 57, 2 (2020), 102142.

[14] Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *2018 IEEE International Conference on Data Mining (ICDM)*. IEEE, 197–206.

[15] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[16] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).

[17] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems* 25 (2012), 1097–1105.

[18] Chen Ma, Peng Kang, and Xue Liu. 2019. Hierarchical gating networks for sequential recommendation. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 825–833.

[19] Julian J. McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel. 2015. Image-Based Recommendations on Styles and Substitutes. In *Proceedings of SIGIR*. ACM, 43–52.

[20] Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model.. In *Interspeech*, Vol. 2. Makuhari, 1045–1048.

[21] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch. (2017).

[22] Yuqi Qin, Pengfei Wang, and Chenliang Li. 2021. The world is binary: Contrastive learning for denoising next basket recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 859–868.

[23] Ruihong Qiu, Jingjing Li, Zi Huang, and Hongzhi Yin. 2019. 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*. 579–588.

[24] Pengjie Ren, Zhumin Chen, Jing Li, Zhaochun Ren, Jun Ma, and Maarten De Rijke. 2019. Repeatnet: A repeat aware neural recommendation machine for session-based recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 4806–4813.

[25] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing personalized Markov chains for next-basket recommendation. In *Proceedings of the 19th International Conference on World Wide Web, WWW 2010, Raleigh, North Carolina, USA, April 26-30, 2010*.

[26] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108* (2019).

[27] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-Attention with Relative Position Representations. In *Proceedings of NAACL-HLT*. 464–468.

[28] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management*. 1441–1450.

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

[30] Aäron Van Den Oord, Sander Dieleman, and Benjamin Schrauwen. 2013. Deep content-based music recommendation. In *Neural Information Processing Systems Conference (NIPS 2013)*, Vol. 26. Neural Information Processing Systems Foundation (NIPS).

[31] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in NeurIPS 2017*. 5998–6008.

[32] Jizhe Wang, Pipei Huang, Huan Zhao, Zhibo Zhang, Binqiang Zhao, and Dik Lun Lee. 2018. Billion-scale commodity embedding for e-commerce recommendation in alibaba. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 839–848.

[33] Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. 2019. Session-based recommendation with graph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 346–353.

[34] Xu Xie, Fei Sun, Zhaoyang Liu, Shiwen Wu, Jinyang Gao, Bolin Ding, and Bin Cui. 2020. Contrastive learning for sequential recommendation. *arXiv preprint arXiv:2010.14395* (2020).

[35] Chengfeng Xu, Pengpeng Zhao, Yanchi Liu, Victor S Sheng, Jiajie Xu, Fuzhen Zhuang, Junhua Fang, and Xiaofang Zhou. 2019. Graph Contextualized Self-Attention Network for Session-based Recommendation.. In *IJCAI*, Vol. 19. 3940–3946.

[36] Fajie Yuan, Xiangnan He, Haochuan Jiang, Guibing Guo, Jian Xiong, Zhezhao Xu, and Yilin Xiong. 2020. Future data helps training: Modeling future contexts for session-based recommendation. In *Proceedings of The Web Conference 2020*. 303–313.

[37] Guorui Zhou, Na Mou, Ying Fan, Qi Pi, Weijie Bian, Chang Zhou, Xiaoqiang Zhu, and Kun Gai. 2019. Deep interest evolution network for click-through rate prediction. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 5941–5948.

[38] Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. 2020. S3-Rec: Self-Supervised Learning for Sequential Recommendation with Mutual Information Maximization. In *Proceedings of CIKM 2020*. 1893–1902.

[39] Kun Zhou, Hui Yu, Wayne Xin Zhao, and Ji-Rong Wen. 2022. Filter-enhanced MLP is All You Need for Sequential Recommendation. *arXiv preprint arXiv:2202.13556* (2022).

[40] Jing Zhu, Yanan Xu, and Yanmin Zhu. 2020. Modeling Long-Term and Short-Term Interests with Parallel Attentions for Session-Based Recommendation. In *International Conference on Database Systems for Advanced Applications*. Springer, 654–669.